ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

"Diseño de tarjetas para el desarrollo de aplicaciones con dsPIC"

TESIS DE GRADO

Previa la obtención del Título de:

INGENIERO EN ELECTRICIDAD

ESPECIALIDAD EN ELECTRÓNICA Y AUTOMATIZACIÓN INDUSTRIAL

Presentado por:

Orlando Xavier Barcia Rosero

Roger Alfredo Calle Chóez

GUAYAQUIL – ECUADOR

AÑO 2008

AGRADECIMIENTO

A Dios.

A los amigos.

A todas las personas que contribuyeron en el desarrollo de este trabajo.

A todos quienes apuestan por el desarrollo tecnológico en Ecuador.

DEDICATORIA

A Dios que siempre nos ha acompañado, siendo su amor la fuente de energía para alcanzar nuestras metas.

A nuestros padres, por su comprensión y ayuda incondicional, quienes siempre nos inculcaron perseverancia con valores éticos, permitiéndonos iniciar nuestra vida profesional, y a nuestros hermanos que siempre han apoyado nuestras decisiones y retos.

TRIBUNAL DE GRADUACIÓN

Ing. Holger Cevallos U. Presidente del Tribunal	Ing. Carlos Valdivieso A. Director de Tesis
Ing. Efrén Herrera Miembro Principal	Ing. Carlos Salazar Miembro Principal

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de este trabajo, nos corresponde exclusivamente; y el patrimonio intelectual del mismo a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”.

(Reglamento de exámenes y títulos profesionales de la ESPOL)

Orlando Xavier Barcia Rosero

Roger Alfredo Calle Chóez

RESUMEN

El presente proyecto de tópico tiene como finalidad el diseño y construcción de tarjetas para el desarrollo de diferentes aplicaciones con dsPIC. La implementación de las dos tarjetas se ha realizado sobre placa de fibra de vidrio a doble capa y el dsPIC30F4011 es la unidad de control implementada en dos tarjetas.

La tarjeta de entrenamiento tiene implementado diversos periféricos y salidas por conectores que podrán ser usados para diferentes aplicaciones y serán controladas según el código del programa desarrollado por el usuario y que ha sido grabado en el dsPIC30F4011 de la respectiva tarjeta logrando operar los periféricos, también es posible realizar sobre la misma tarjeta el manejo de tres tipos de motores (servomotor, de pasos, DC); con respecto a la tarjeta para aplicaciones básicas viene provista de conectores apropiados para salidas de las señales de control para diferentes aplicaciones.

Para poder trabajar con la tarjeta de entrenamiento primero se debe configurar adecuadamente los puentes de habilitación para los periféricos que requiera usarse, lo anterior es muy ventajoso porque posibilita la implementación de varios periféricos en la tarjeta con una misma señal del controlador, en la tarjeta de aplicaciones básicas los periféricos deben ser implementados externamente y las señales del dsPIC llegan a través de los conectores disponibles en la misma.

En ambas tarjetas se ha tratado de aprovechar la mayor parte de los recursos del dsPIC con la finalidad de disponer de un modelo para entrenamiento muy útil y que contribuya con futuras investigaciones basadas en este tipo de controladores.

ÍNDICE GENERAL

ABREVIATURAS

ADCONx	Registro de control x del bloque analógico digital.
ADCHS	Registro de selección de canal de entrada.
ADPCFG	Puerto de configuración de registros.
ADCSSL	Registro de selección para el análisis.
AGU	Unidad generadora de direcciones.
ALU	Unidad lógica aritmética.
ANSI	Instituto estadounidense de estándares nacionales.
ASCII	Código estadounidense estándar para el intercambio de información.
BCD	Decimal codificado en binario.
Bit	Unidad de medida de información equivalente a la elección entre dos posibilidades igualmente probables.
Born. Exter	Bornera externa.
CAD	Diseño asistido por computadora.
CANT	Cantidad.
CF	Costos fijos.
CM	Costo de material.
CODEC	Codificador decodificador.
CPU	Unidad central de proceso.
CT	Costos totales.
CV	Costos variables.
DC	Corriente directa.
DSC	Controlador digital de señales.
DSP	Procesador digital de señales.
dsPIC	Dispositivo con soporte para procesamiento de señales.
H	Horas.
HEX	Fichero en hexagecimal de un programa para la carga.
ICD2	Depurador en circuito.

ICx	Pin de entrada del módulo de captura del canal x.
IDE	Entorno integrado de desarrollo.
INT	Interrupción.
IT	Ingresos totales.
I2C	Protocolo de comunicación entre integrados.
kHz	Kilo hertz.
kOhm	Kilo ohmio.
LCD	Pantalla de cristal líquido.
Led	Diodo emisor de luz.
MCU	Microcontrolador producido por Microchip.
MHz	Mega hertz.
MSPS	Millones de muestras por segundo
NI	National Instrument.
OCx	Canal x del módulo de comparación de salida.
Ohm	Ohmio.
PC	Computador personal.
PCB	Tarjeta con circuito impreso.
PDCx	Registro asociado al ciclo de trabajo de la señal PWM.
PGC	Pin de envío de pulsos de reloj (ICD2).
PGD	Pin de envío serial de datos (ICD2).
PIC	Microcontrolador de Microchip.
PID	Proporcional, integral y derivativo.
PRx	Registro para período del temporizador x.
PVT	Precio de venta total.
PWM	Modulación de ancho de pulso.
PWM xL	Señal PWM en bajo del canal x.
PWM xH	Señal PWM en alto del canal x.
Qx	Transistor x tipo mosfet de canal N.
Rx	Señal de recepción del módulo de comunicación serial.
RBx	Pin x del puerto B del dsPIC.
RCx	Pin x del puerto C del dsPIC.
RDx	Pin x del puerto D del dsPIC.
REx	Pin x del puerto D del dsPIC.
RFx	Pin x del puerto F del dsPIC.
RISC	Conjunto de instrucciones reducidas.
RPM	Revoluciones por minuto.
RS-232	Estándar recomendado 232 para comunicación serial.
SAR	Registros de Aproximación Sucesiva.
SCADA	Control para supervisión y adquisición de datos.
SCK	Reloj de comunicación serial (SPI).
SCL	Reloj serial sincrónico (I2C).
SDA	Dato seriales asincrónico (I2C).
SDI	Entrada serial de datos (SPI).
SDO	Salida serial de datos (SPI).

SPI	Protocolo de interfase serial entre periféricos.
Temp	Temperatura.
THD	Montaje a través de agujeros.
TMRx	Registro de contador para temporizador x.
TxCON	Registro de control del temporizador x.
Tx	Señal de transmisión del módulo de comunicación serial.
TxIE	Bits de habilitación de interrupción para temporizador x.
TxIF	Bits de bandera de interrupción para temporizador x.
TxIP	Bits de prioridad de interrupción para temporizador x.
UART	Transmisión serial asincrónica.
UAII	Ingresos de la utilidad.
UART	Unidad de transmisión recepción serial asincrónica.
UNE	Unión de normas europeas.
UPS	Sistemas de alimentación ininterrumpidas.
USB	Bus serial universal.
UxTX	Pin de transmisión del módulo UARTx.
UxRX	Pin de recepción del módulo UARTx.
U1ATx	Pin de transmisión del módulo UART1 alternativo.
U1ARx	Pin de recepción del módulo UART1 alternativo.
U2Tx	Pin de transmisión del módulo UART2.
U2Rx	Pin de recepción del módulo UART2.
VDC	Voltaje de alimentación continuo.
VI	Instrumento virtual.
Visualiz Segx	Segmento x del visualizador de 7 segmentos.
Visualiz Punto	Punto del visualizador de 7 segmentos.

SIMBOLOGÍA

A/D	Analógico a digital.
AC	Corriente alterna.
A	Amperio.
Ci	Capacitancia intrínseca.
cm	Centímetro.
ºC	Grado Celsius.
DB9	Conector de nueve pines para comunicación serial.
E/S	Entradas y salidas.
Li	Inductancia intrínseca.
uF	Micro Faradios.
uH	Micro Henrios.
pF	Pico Faradios.
Vcc	Voltaje de control.
Vin	Voltaje de entrada.
Vmotor	Voltaje de alimentación para motor DC.
Vo	Voltaje de salida.
V	Voltio.
ω_0	Frecuencia de resonancia.
x2	Modo para medición de posición con encoder.
x4	Modo para medición de posición con encoder .
Z	Impedancia.
$	Dólares.

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

El presente documento corresponde a un proyecto final del tópico de graduación “Plataforma para el desarrollo de proyectos con dsPICs y visión robótica con LabVIEW”, este proyecto consiste en el “Diseño de tarjetas para el desarrollo de aplicaciones con dsPIC”.

El primer capítulo describe el alcance de las especificaciones técnicas del sistema a implementarse el cual esta constituido por tres tarjetas de entrenamiento, las características del dispositivo de control, y el software utilizado para la implementación del proyecto.

El capítulo dos explica las consideraciones para el diseño de las tarjetas entrenamiento, también hace referencia a la distribución de los diferentes periféricos implementados sobre la tarjeta de entrenamiento, para finalizar este capítulo se describe el fundamento teórico sobre el diseño de placas de circuito impreso indicando los parámetros requeridos para el cálculo del ancho de las pistas de cobre para la fabricación de las tarjetas.

El tercer capítulo expone los detalles de construcción de las placas de circuito impreso, además enseña la correcta configuración de los puentes que habilitan a los periféricos implementados y los tipos de comunicación disponibles sobre la tarjeta de entrenamiento con dsPIC30F4011, en este capítulo también se realiza un análisis económico de los costos de producción de la tarjeta de entrenamiento.

El capítulo cuatro presenta ejemplos para el manejo de los periféricos, motores, y de comunicación serial, que podrán implementarse en las tarjetas.

En el capitulo cinco se presenta las pruebas realizadas para el manejo de un motor de corriente continua y su comportamiento cuando se realiza un control PI.

Para finalizar la descripción del documento, se expondrán las conclusiones y recomendaciones, adquiridas durante la implementación del proyecto.

CAPÍTULO 1

1. GENERALIDADES Y ARQUITECTURA DE LOS DSPIC

1.1 Antecedentes

El estudio de las tarjetas desarrolladoras para sistemas basados en microcontroladores es parte fundamental de la formación de los estudiosos de la materia, debido a las grandes ventajas que ofrecen, permitiendo el trabajo con múltiples periféricos e interfases de forma fácil y sencilla.

El problema que se plantea en este proyecto final de tópico es el de encontrar un entorno de desarrollo, específicamente tarjetas que permitan el estudio de circuitos e interfases con dsPICs. Existen múltiples soluciones para este problema pero cada fabricante pone énfasis en sus propias características con el objeto de lograr inducir al cliente en el uso de sus desarrolladores que funcionan con ciertos dispositivos y generalmente utilizando programas propietarios. El objetivo de este proyecto es desarrollar una tarjeta principal que permita el empleo de programas de uso común que pueden obtenerse con facilidad y evitar los hardwares y softwares propietarios que limitan el desarrollo amplio y sin fronteras de aplicaciones. Una tarjeta a diseñarse es la *tarjeta de aplicaciones con dsPIC30F4011*, está permite acoplar otros módulos con aplicaciones específicas, tal como es el caso del *manejador de motores DC*, que se utilizarán en la Plataforma de Desarrollo.

La tarjeta desarrolladora a implementarse tiene como objeto principal servir de soporte técnico y experimentación al diseñador de proyectos en donde principalmente se necesita velocidad de procesamiento y precisión en los resultados obtenidos, tomando en cuenta que estos dispositivos microcontroladores dsPIC, facilitan el manejo de estos parámetros teniendo como plataforma de programación los mismos entornos usados para la programación de los microcontroladores MCU comunes.

En el mercado existen tarjetas de desarrollo que proporcionan prestaciones básicas que no permiten utilizar la mayoría de funciones que ofrece el dsPIC como elemento de desarrollo. Considerando que para la ejecución de proyectos el diseñador muchas veces se ve en la necesidad añadir sobre la marcha nuevas prestaciones al prototipo, el mismo que debe de contar con una plataforma que le permita disponer de una amplia gama de funciones, es decir contar con una herramienta de apoyo que satisfaga las principales necesidades del diseñador. Siendo muy importante que esta herramienta nos permita manejar los campos de las aplicaciones más comunes pertinentes a la arquitectura de los dsPICs las cuales muy difícilmente podrían implementarse con los MCUs comunes y en ciertos casos sería una tarea prácticamente imposible.

Aprender a manejar y aplicar los microcontroladores sólo se consigue desarrollando prácticamente diseños reales, aún más cuando no se cuenta con simuladores disponibles como en los microcontroladores clásicos, esto es prioritario en la concepción de un proyecto en laboratorio. Por lo tanto la tarjeta de desarrollo se convertirá en una herramienta invalorable para el diseñador con dsPICs.

1.2 Especificaciones técnicas del sistema a implementarse

1.2.1 Características técnicas de la tarjeta

Este proyecto consiste en el diseño e implementación de dos módulos de entrenamiento basado en dsPIC, y una tarjeta manejadora de motores DC.

A continuación se describe en forma general los beneficios de los módulos de entrenamiento basados en dsPIC, en la ***tarjeta de entrenamiento con dsPIC30F4011*** se encuentran implementados periféricos de forma bien diferenciada para fines educativos, adicionalmente una ***tarjeta básica para aplicaciones con dsPIC30F4011***, útil para desarrollo y control de nuevas aplicaciones como el manejo de cuatro motores de corriente continua los cuales podrán mover un sistemas de bandas transportadoras. También es posible usar las salidas externas disponibles en los módulos como elemento de partida para el desarrollo de otras aplicaciones como el manejo de sensores que podrán ser conectados de manera externa al módulo, previo su respectivo acondicionamiento, para tal propósito el módulo cuenta con salidas destinados para esta función.

Para acceder de manera fácil al sistema desarrollado los módulos soportan aplicaciones tipo SCADA, las cuales podrán ser desarrolladas en software destinados para este propósito como LabVIEW. ***En la tarjeta de entrenamiento con dsPIC30F4011***, se podrán usar este tipo de aplicaciones como el manejo de los motores de paso, corriente continua, y servomotores; además del uso de sensores implementados en la misma. El módulo de entrenamiento cuenta con varios puertos de comunicación, uno de esos es el puerto serial con estándar RS-232, es el puerto de comunicación usado entre el computador que ejecuta la aplicación SCADA y la tarjeta de entrenamiento.

Con respecto a la ***tarjeta manejadora de motores DC***, es un módulo que puede ser acoplado a cualquiera de las dos tarjetas anteriormente descritos.

1.2.2 Entradas y salidas de los módulos

La tarjeta básica para aplicaciones con dsPIC30F4011, dispone de borneras para conexiones de periféricos externos, estas borneras están asociadas directamente con los pines del dsPIC en mención, de tal forma poder usarlas como entradas o salidas, permitiendo que el usuario configure por software los pines del dsPIC ya sea como entradas digitales o analógicas, o como salidas digitales.

En la tarjeta de entrenamiento con dsPIC30F4011, también se debe configurar los pines del dsPIC, pero se debe de considerar que existen periféricos conectados a través de puentes e interruptores a los pines de la unidad de control, considerando esto se debe elegir de acuerdo al periférico usado cuando un pin es configurado como entrada, salida o ambos en algunos casos. En la siguiente tabla se hace un bosquejo de la distribución de los periféricos conectados a los pines del dsPIC.

Tabla 1.1 Distribución de pines de los periféricos de la tarjeta de entrenamiento con dspic30f4011

PIN	PUERTO	PERIFÉRICO 1	2	3	
2	RB0	E/S Externa	Diodo Led 1	Teclado Matricial	–
3	RB1	E/S Externa	Diodo Led 2		–
4	RB2	Relé	Interruptor1		SPI-SS
5	RB3	Encoder	Interruptor2		–
6	RB4		Interruptor3		–
7	RB5		Interruptor4		–
8	RB6	Sensor Temp.	Potenciometro2		–
9	RB7	Potenciometro1	Botón2		–
10	RB8	Led	–	–	–
15	RC13	LCD	Visualiz SegA	Barra de led1	U1ATx
16	RC14		Visualiz SegB	Barra de led2	U1ARx
23	RD0		Visualiz SegC	Barra de led3	–
18	RD1		Visualiz SegD	Barra de led4	–
22	RD2		Visualiz SegE	Barra de led5	–
19	RD3		Visualiz SegF	Barra de led6	–
38	RE0	E/S Externa	Visualiz SegG	Barra de led7	–
37	RE1	E/S Externa	Visualiz Punto	Barra de led8	–
36	RE2	PWM 2L	E/S Externa	–	–
35	RE3	PWM 2H	E/S Externa	–	–
34	RE4	PWM 3L	E/S Externa	–	–
33	RE5	PWM 3H	E/S Externa	–	–
17	RE8	Pulsador INT	E/S Externa	–	–
30	RF0	PS/2	E/S Externa	–	–
29	RF1		E/S Externa	–	–
26	RF2	Habilitador Visualiz1 / ICD2-PGC / I2C-SDA / SPI-SDI / ES Externa			
25	RF3	Habilitador Visualiz2 / ICD2-PGD / I2C-SCL / SPI-SDO / ES Externa			
28	RF4	U2Rx	–	–	–
27	RF5	U2Tx	–	–	–
24	RF6	Bocina	Botón1	SPI-SCK	IO Externa

-- : No disponible

FUENTE: *Personal*

Se debe de recordar que la unidad de control es el dsPIC30F4011 y como tal maneja señales digitales tanto en la entrada como en la salida, por lo cual su nivel voltaje en operación normal es de 5 VDC máximo o de 0 VDC como valor mínimo. Siendo estos valores interpretados como salidas o entradas lógicas que toman los valores de uno y cero lógico respectivamente. Respecto a las señales analógicas el dsPIC solo dispone de entradas analógicas, dicha entrada es interpretada y codificada en el interior del dsPIC a una conversión digital de 10 ó 12 bits dentro de un determinado rango que dependerá de los voltajes de referencia.

1.2.3 Fuentes de alimentación

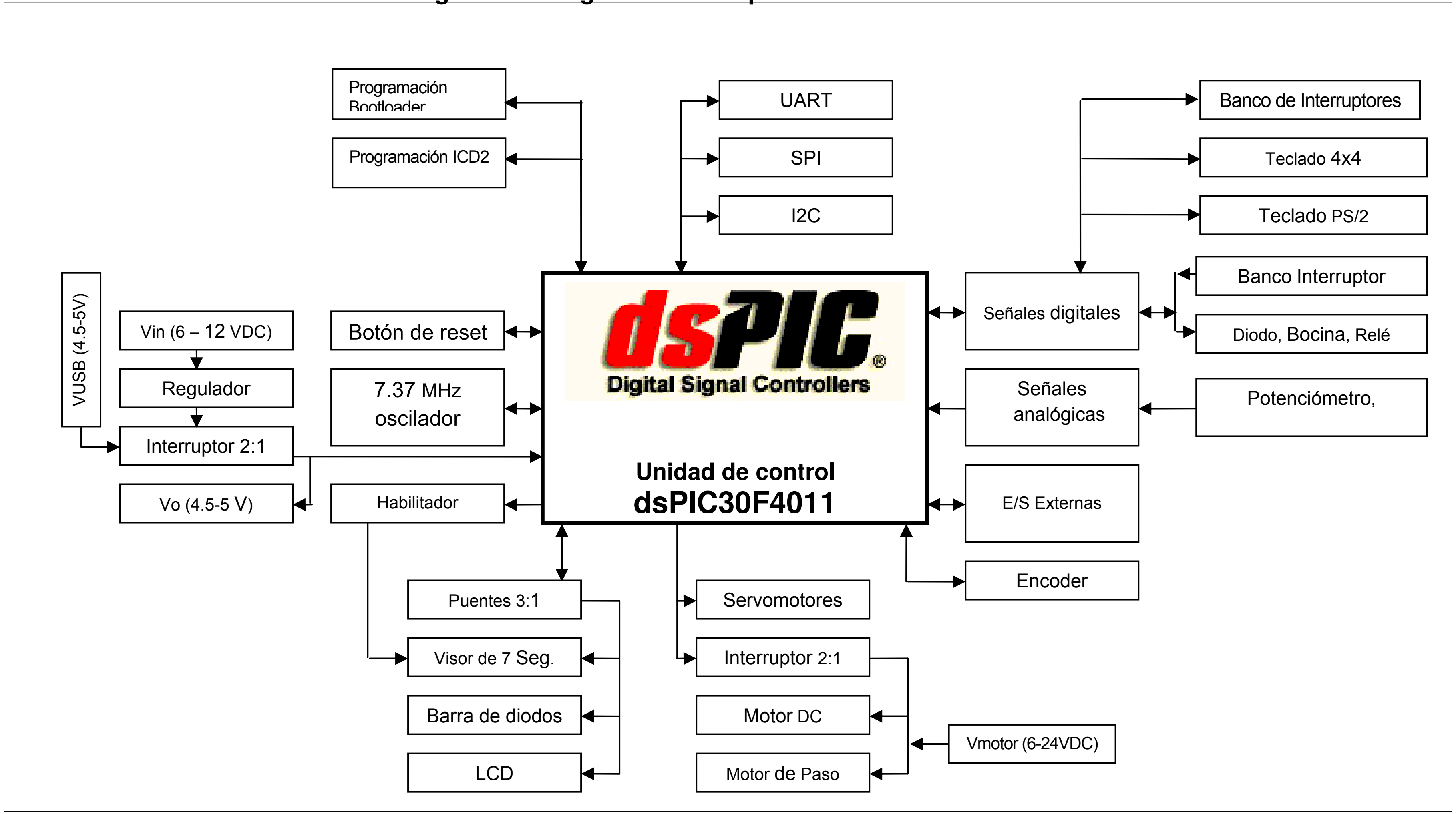
Se requieren dos niveles de tensión de voltaje continuo, uno para la unidad de control y adicionalmente otro que suministre tensión al bloque de motores. Respecto a la alimentación del circuito de control, esta se obtiene de una fuente DC con capacidad de entregar de 4.5 a 5.5V y hasta 1 amperio en la salida, los integrados usados para este propósito son el puente rectificador W06M y el regulador de voltaje LM7805 que nos permite obtener los parámetros de alimentación requeridos para el circuito de control y periféricos disponibles que manejan estos niveles de tensión en la tarjeta.

En cuanto al suministro de energía para los motores esta se la obtiene de una fuente externa que alimenta con un voltaje de 6 a 24V, hasta un máximo de 5 amperios.

1.2.4 Diagrama de bloques

El diagrama mostrado a continuación detalla la distribución de los bloques que forman parte del proyecto, como bloque central se encuentra implementado el dsPIC30F4011, este se encarga de las funciones de control de los periféricos que se encuentran implementados en la tarjeta de entrenamiento, este microcontrolador también esta encargado de la comunicación con la PC.

Figura 1.1 Diagrama de bloques del módulo de entrenamiento



FUENTE: *Personal*

1.3 Revisión histórica de los PIC y dsPIC

1.3.1 Generalidades de los microcontroladores

El mundo de los microcontroladores está basado en la tecnología de los circuitos integrados, permitiendo almacenar miles de transistores dentro de un solo chip. Esto fue un prerrequisito para la producción de los microprocesadores. Las primeras computadoras fueron construidas agregando componentes externos como lo son: memorias, puertos de salida y entrada, contadores de tiempo, entre otros. A diferencia de los microprocesadores el microcontrolador es un circuito integrado programable que contiene todos los componentes de un computador, empleado para controlar un determinado proceso, su reducido tamaño facilita su instalación en el elemento a gobernar, esto se denomina controlador embebido. Siendo un computador dedicado, solo gobierna una aplicación determinada programada en su memoria, sus entradas y salidas soportan conexiones de sensores y actuadores.

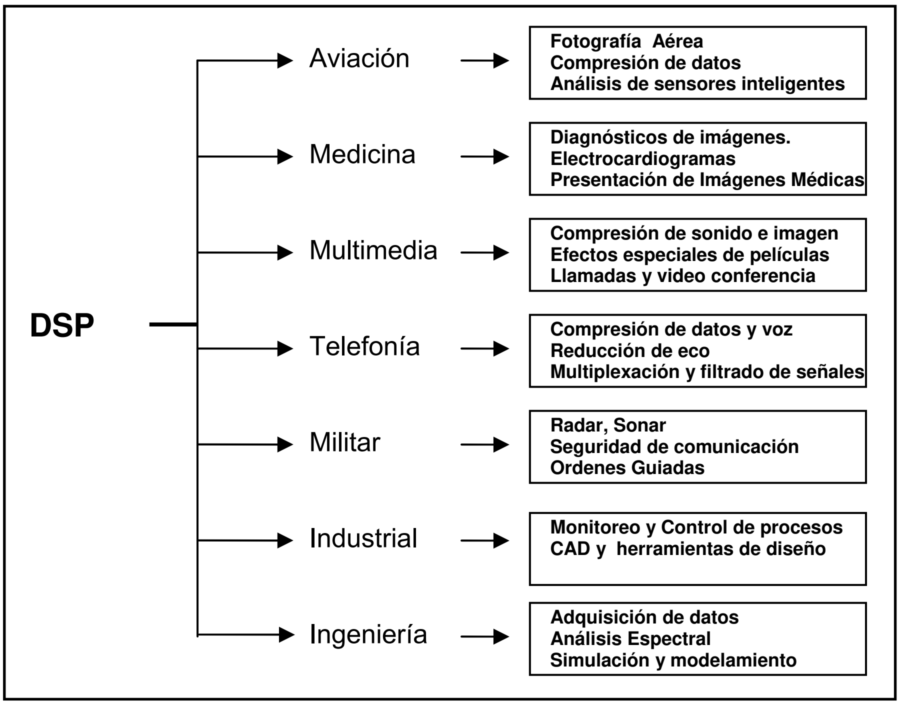
En la actualidad los microcontroladores han solucionado un gran número de problemas como en el hogar, industrias, y en diversas áreas de aplicación.

Es notable la gran aceptación en el mercado por parte del microcontrolador, pero con el avance vertiginoso de la tecnología han surgido nuevas áreas de aplicación existiendo problemas donde su eficiencia en precisión y velocidad se ve limitada, siendo estos dos parámetros críticos al momento de su implementación, a pesar de que el PIC es considerado como un computador completo.

1.3.2 ¿Por qué el dsPIC?

Los DSP o Procesadores Digitales de Señales surgieron cuando la tecnología permitió su fabricación y las aplicaciones en el manejo de señales lo requirieron. Los DSP son procesadores digitales cuyo diseño ha sido enfocado para soportar las especificaciones del tratamiento de señales, que por su implicación en campos tecnológicos más demandados recientemente, supone disponer de un dispositivo programable con recursos físicos y lógicos precisos para las exigencias de dichas áreas, como la telefonía móvil, la electromedicina, aplicaciones robóticas, el procesamiento del sonido y la imagen, internet, el control de motores, y la instrumentación.

Figura 1.2 Aplicaciones DSP



FUENTE: *Instituto Tecnológico de Puebla, México*

Los microprocesadores digitales y microcontroladores de 8,16 y 32 bits no tienen la capacidad de procesar eficientemente las tareas que el procesamiento digital de señales exige, procediéndose a reforzar sus arquitecturas, se amplió el repertorio de instrucciones y se les integró con numerosos periféricos complementarios para dar lugar a los dsPIC.

La compañía *Microchip Technology Inc.,* constituye uno de los fabricantes que en los últimos años en la comercialización de los microcontroladores de 8 bits, actualmente tiene a la disposición de sus clientes unos nuevos componentes llamados dsPIC, que les permite introducirse en las aplicaciones contemporáneas del proceso digital de señales de forma sencilla basada en arquitecturas y repertorios de instrucciones conocidos.

El dsPIC es un controlador digital de señales (DSC) que esta basado en las características más potentes de los microcontroladores de 16 bits y que incorpora las principales prestaciones de los DSP, lo que facilita enormemente la transición entre los diseños clásicos y los más complejos y propios del procesamiento digital de señales.

1.3.3 Familias de dsPIC

Microchip ha dividido los diferentes modelos de la serie dsPIC30F en tres familias caracterizadas por su área de aplicación, nombradas a continuación:

Familia de dsPIC de propósito general

La familia de propósito general dsPIC30F utilizada en aplicaciones embebidas que requieren un MCU de 16-bits. Además, las variantes con interfases para CODEC están especialmente indicadas para aplicaciones de audio. La tabla mostrada a continuación detalla las características de propósito general de la familia dsPIC30F:

Tabla 1.2 Tipos dsPIC de propósito general

DISPOSITIVO	P i N	MEMORIA FLASH		SRAM	EEPROM	ENTRADAS MODO CAPTURA
		BYTES	INSTRUCC			
dsPIC30F3014	40/44	24kB	8kB	2048 B	1024 B	2
dsPIC30F4013	40/44	48kB	16kB	2048 B	1024 B	4
dsPIC30F5011	64	66kB	22kB	4096 B	1024 B	8
dsPIC30F6011	64	132kB	44kB	6144 B	2048 B	8
dsPIC30F6012	64	144kB	48kB	8192 B	4096 B	8
dsPIC30F5013	80	66kB	22kB	4096 B	1024 B	8
dsPIC30F6013	80	132kB	44kB	6144 B	2048 B	8
dsPIC30F6014	80	144kB	48kB	8192 B	4096 B	8

DISPOSITIVO	SALIDA MODO COMPARADOR	INTERFAZ CODEC	A/D 12-BIT 100Ks	UART	SPI	I2C	CAN	LINEAS E/S
dsPIC30F3014	2		13	2	1	1		30
dsPIC30F4013	4	AC97,1	13	2	1	1	1	30
dsPIC30F5011	8	AC97,1	16	2	2	1	2	52
dsPIC30F6011	8		16	2	2	1	2	52
dsPIC30F6012	8	AC97,1	16	2	2	1	2	52
dsPIC30F5013	8	AC97,1	16	2	2	1	2	68
dsPIC30F6013	8		16	2	2	1	2	68
dsPIC30F6014	8	AC97,1	16	2	2	1	2	68

FUENTE: *Hoja de datos dsPIC30F*

Familias dsPIC para el control de motores

La familia de los dsPC30F reúne siete modelos para soportar aplicaciones de control de motores tanto de corriente continua, de inducción monofásica y trifásica, pudiendo también ser utilizados en sistemas de alimentación ininterrumpidas (UPS), fuentes de alimentación conmutadas, inversores. Siendo adecuado en aplicaciones donde el algoritmo para el control de motores es de elevada complejidad.

Tabla 1.3 Tipos dsPIC para control de motores

DISPOSITIVO	PINES	MEMORIA FLASH		SRAM	EEPROM	ENTRADA MODO CAPTURA	SALIDA MODO COMPARA
		BYTE	INSTRUCC				
dsPIC30F2010	28	12kB	4kB	512 B	1024 B	4	2
dsPIC30F3010	28	24kB	8kB	1024 B	1024 B	4	2
dsPIC30F4012	28	48kB	16kB	2048 B	1024 B	4	2
dsPIC30F3011	40/44	24kB	8kB	1024 B	1024 B	4	4
dsPIC30F4011	40/44	48kB	16kB	2048 B	1024 B	4	4
dsPIC30F5015	64	66kB	22kB	2048 B	1024 B	4	4
dsPIC30F6010	80	44kB	48kB	8192 B	1024 B	8	8

DISPOSITIVO	CONTROL MOTORES	A/D 12-BIT 500Ksps	CODIFICADOR CUADRATURA	UART	SPI	I2C	CAN	LINEAS E/S
dsPIC30F2010	6 canales	6 canales	Sí	1	1	1		20
dsPIC30F3010	6 canales	6 canales	Sí	1	1	1		20
dsPIC30F4012	6 canales	6 canales	Sí	1	1	1	1	20
dsPIC30F3011	6 canales	9 canales	Sí	2	1	1		30
dsPIC30F4011	6 canales	9 canales	Sí	2	1	1	1	30
dsPIC30F5015	6 canales	16 canales	Sí	1	2	1	1	52
dsPIC30F6010	8 canales	16 canales	Sí	2	2	1	2	68

FUENTE: *Hoja de datos dsPIC30F*

La familia de los dsPIC30 dispone de dos subfamilias para el control de motores, una es de gama baja que incorpora módulos PWM de seis salidas, y otra de gama superior con 8 salidas. La siguiente tabla muestra las características de los módulos PWM de las dos subfamilias:

Tabla 1.4 Características de los tipos de módulos PWM de dsPIC

CARACTERÍSTICAS	PWM 6 SALIDAS (gama baja)	PWM 8 SALIDAS (gama alta)
Pines E/S	6	8
Generadores PWM	3	4
Pines de fallo	1	2
Generadores de tiempo muerto	1	2

FUENTE: *Hoja de datos dsPIC30F*

Familia de dsPIC para el control de sensores

La familia de sensores dsPIC30F tiene características diseñadas para soportar aplicaciones embebidas de altas prestaciones y de bajo costos. Los encapsulados de 18- y 28-pines están diseñados para aplicaciones críticas en espacio.

Tabla 1.5 Tipos dsPIC para control de sensores

DISPOSITIVO	PINES	MEMORIA FLASH		SRAM	EEPROM	ENTRADAS MODO CAPTURA
		BYTES	INSTRUCC			
dsPIC30F2011	18	12kB	4kB	1024 B		2
dsPIC30F3012	18	24kB	8kB	2048 B	1024 B	2
dsPIC30F2012	28	12kB	4kB	1024 B		2
dsPIC30F3013	28	24kB	8kB	2048 B	1024 B	2

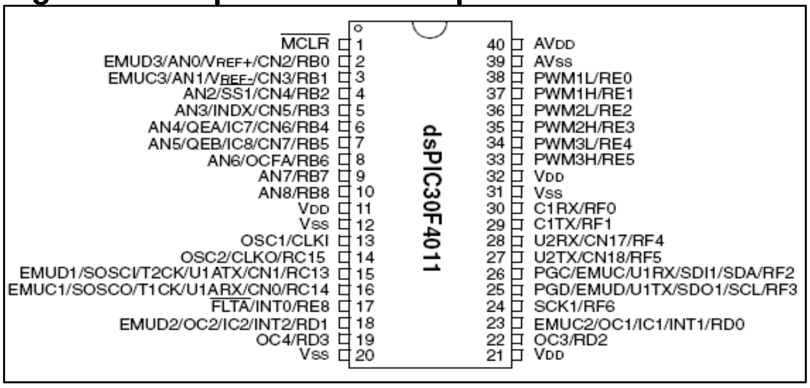
DISPOSITIVO	SALIDAS MODO COMPARADOR	INTERFAZ CODEC A/D 12-BIT 100Ksps	UART	SPI	I2C	LINEAS E/S
dsPIC30F3011	2	8 canales	1	1	1	20
dsPIC30F3012	2	8 canales	1	1	1	20
dsPIC30F2012	2	10 canales	1	1	1	20
dsPIC30F3013	2	10 canales	2	1	1	20

FUENTE: *Hoja de datos dsPIC30F*

1.4 Estructura de los dispositivos

El principal componente del sistema de control es el microcontrolador dsPIC30F4011, el cual tiene un núcleo RISC con arquitectura Harvard modificada de 16 bits. Su estructura se puede separar en tres partes: microprocesador, integración del sistema y periféricos.

Figura 1.3 Disposición de los pines del dsPIC30F4011



FUENTE: *Hoja de datos Microcontrolador 30F4011*

La CPU tiene palabras de instrucción de 24 bits con un campo variable para el código de operación. Tiene además 16 registros de trabajo que junto con instrucciones tipo MCU y DSP realizables en un ciclo de instrucción proveen una gran rapidez y complejidad de cálculo.

La memoria de datos puede contener 32 mil palabras o 64 mil bytes que se separan en dos bloques cada uno con su propia AGU (generadores de direcciones). La arquitectura DSP contiene un multiplicador de 17 bits, una ALU, dos acumuladores de 40 bits, un registro de desplazamiento bidireccional de 40 bits, todo esto permite realizar un procesamiento de señales en tiempo real de manera óptima.

El dsPIC30F4011 forma parte de la familia dsPIC30F para el control de motores en su arquitectura tiene potentes unidades internas que favorecen esta función entre las que se encuentran: seis canales de PWM con salidas independientes o complementarias con una base de tiempos dedicada, posibilidad de controlar la polaridad de las salidas, generación de tiempos muertos, etc. También se dispone de módulo de detección de señales de encoder en cuadratura con un contador interno de 16 bits.

Tabla 1.6 Características del dsPIC30F4011

CARACTERÍSTICAS	CAPACIDAD
Frecuencia de operación	EC a 40 MHz
Memoria de Instrucción (Bytes)	16 K
Memoria de Programa FLASH	48 K
Memoria de Datos RAM (Bytes)	2 K
Memoria EEPROM (Bytes)	1 K
Fuentes de Interrupción	30
Puertos de Entrada/Salida	5 (Puerto B,C,D,E,F)
Instrucciones Básicas	83
Temporizadores de 16 bits	5
Adicionalmente timers de 32 bits	2
Módulos de PWM	6
Comunicaciones Seriales	SPI,I2C,UART,CAN
Canales de Conversión Analógica Digital de 10 bits	9
Módulo Encoder de Cuadratura de 16 bits	1
MOTOR DSP	
➢ Multiplicador rápido 17 X 17 bits. ➢ Registro de desplazamiento de 40 bits. ➢ Sumador restador de 40 bits. ➢ Dos registros acumuladores de 40 bits. ➢ Modos de operación: redondeo y saturación lógica. ➢ Todas las instrucciones son de un solo ciclo.	

FUENTE: *Hoja de datos dsPIC30F4011*

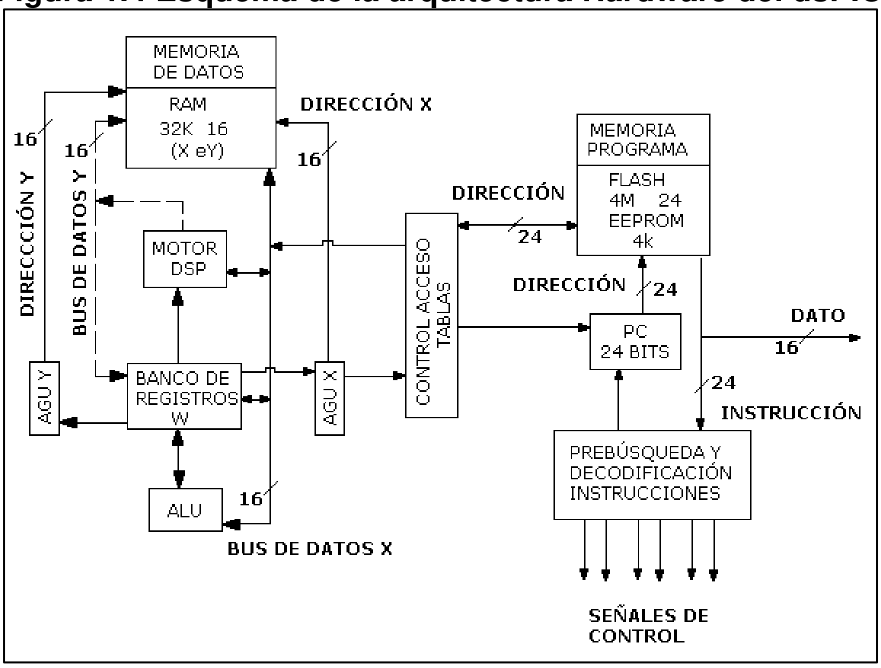
Todos los procesadores de los dsPIC incorporan en su procesador las siguientes características:

a) Arquitectura Hardware: posee dos memorias una para datos y otra para instrucciones.
b) Procesador RISC: posee un conjunto de instrucciones optimizadas para soportar el lenguaje C.

c) Cauce segmentado: permitiendo elevar el rendimiento del procesador, consiguiendo así que una instrucción empiece a ejecutarse antes de que hayan terminado las anteriores y, por tanto, que haya varias instrucciones procesándose simultáneamente.

d) Integración de Recursos propios de DSP: el aporte más considerable es el motor DSP, facilitando la resolución de operaciones matemáticas complejas, en los algoritmos para el rápido procesamiento de señales.

Figura 1.4 Esquema de la arquitectura Hardware del dsPIC



FUENTE: *Microcontroladores Avanzados, México Autor José María Angulo*

Se observa en el diagrama la división de dos grandes bloques de memoria, la memoria de programa FLASH direccionada por el contador de programa de PC, el segundo bloque formado por la memoria RAM de datos subdividida en dos partes X y Y, controlados por los generadores de direcciones AGU X y AGU Y. También existe un Banco de registros de 16 bits que alimenta al motor DSP.

A continuación se revisan las características más representativas de cada uno de los elementos del microcontrolador dsPIC, los puertos de entrada y salidas, además de los periféricos complementarios al procesador:

1.4.1 Motor DSP

El motor DSP es un bloque del hardware con recursos matemáticos que proporcionan al procesador la posibilidad de realizar operaciones de gran velocidad, que generalmente se emplean en el tratamiento se señales digitales. El motor DSP el cual recibe los datos desde un arreglo de registros (W), los mismos que direccionan el bloque ALU del microcontrolador. Aunque no es posible un manejo simultáneo de instrucciones para la ALU y el motor DSP, el dsPIC puede resolver esto mediante el manejo adecuado del juego de instrucciones.

1.4.2 Temporizadores

Los temporizadores incorporados en los dsPIC30F son de 16 bits, cuyos registros asociados son TMRx (contador), PRx (período), y TxCON (registro de control). Además los bits de interrupciones asociados como TxIE (habilitación de interrupción), TxIF (flag de interrupción), TxIP (3 bits de prioridad de la interrupción).

Existen tres clasificaciones de tipos de temporizadores A, B y C de 16 bits, algunos se pueden encadenar y formar 32 bits. Los de tipo A (timer1) puede funcionar con el oscilador de bajo consumo de 32kHz, de forma asíncrona con un reloj externo. El timer2 y 4 son de tipo B. mientras que el timer3 y 5 corresponden al de tipo C. Únicamente los del tipo B y C se pueden concatenar para formar los temporizadores de 32 bits.

1.4.3 Módulo de captura de entrada.

El módulo de captura de entrada del dsPIC30F4011 tiene cuatro entradas para la captura IC1, IC2, IC7, IC8, siendo muy usado en aplicaciones donde se necesite medir frecuencias o pulsos de periodos de tiempo, también como una fuente adicional de interrupción externa, el funcionamiento de este módulo se basa en poner en marcha y leer el valor de un temporizador cuando sucede un evento determinado en un pin de entrada, el tiempo de lectura es controlado por los timer2 o timer3. Las características operacionales principales son:

i. Modo simple de captura de eventos.
ii. Modo de elección del timer2 y timer3.
iii. Interrupción provocada por evento de captura de entrada.

1.4.4 Módulo de comparación de salidas.

El módulo de comparación salida dispone de 4 canales, pudiendo ser usados en aplicaciones donde se requieran generación de señales digitales con ancho de pulso variable, en la actualidad se encuentran en desarrollo aplicaciones que permitan mejorar los factores de potencia por medio del uso de este módulo. Entre las características operación del módulo de comparación del dsPIC30F4011 se encuentran: posibilidad de usar temporizador 2 ó 3, modo de comparación simple, modo de comparación doble, modulación por ancho de pulso.

1.4.5 Conversor Analógico Digital de 10 bits

El módulo A/D esta constituido por 9 canales de entrada, con convertidor de 10 bits analógico-digital (el dsPIC30F4011 no dispone conversión con 12 bit), y una velocidad de conversión de 500 Kbps, basado en arquitectura SAR (Registros de Aproximación Sucesiva).

Las entradas analógicas interiormente se encuentran multiplexadas y conectadas a amplificadores, las salidas de los amplificadores son conectadas como entradas del bloque convertidor A/D, este último es el encargado de generar nuestro resultado final. El bloque convertidor A/D, posee seis registros de 16 bits: tres registros de control (ADCONx), un registro de selección de canal de entrada (ADCHS), puerto de configuración de registros (ADPCFG), y un registro de selección para el análisis (ADCSSL).

1.4.6 Módulos de comunicación UART, SPI, I2C, CAN.

Para poder comunicarse con otros dispositivos, además de los puertos de E/S los microcontroladores disponen de una serie de módulos de comunicaciones que le permiten conectarse a distintos buses o canales de comunicaciones estándar: UART, SPI, I2C, y CAN.

El módulo de Transmisión Recepción Universal Asincrónico (UART), es el módulo serial básico disponible en la familia de los DsPIC30F, UART es sistema dúplex asincrónico que puede comunicarse con dispositivos periféricos así como interfaces RS-232, y RS-485. Las principales características del módulo UART se describen en la siguiente tabla:

Tabla 1.7 Características del módulo UART del dsPIC30F4011

CARACTERÍSTICAS	CAPACIDAD
Bidireccional a través de UxTX y UxRX	8 ó 9 bits de datos
Opciones de parida y no paridad (8avo bit)	1 bit
Bit de parada	1 ó 2 bits
Generador Baud Rate	preescalar 16 bits
Transmisión de datos en buffer	4-word
Recepción de datos en buffer	4-word
Separación de instrucciones Rx y Tx	
Detección de error de paridad	

FUENTE: *Hoja de datos dsPIC30F4011*

Otra opción de comunicación es la interfaz serial de periféricos (SPI), es un módulo de comunicación serial sincrónico entre dispositivos periféricos como otros microcontroladores, registros, manejador de visualizadores, etc. La cantidad de módulos de SPI depende del modelo del dsPIC, el modelo dsPIC30F4011 tiene un módulo SPI.

Adicionalmente se dispone un bus de comunicaciones serial sincrónica I2C (Interconexión de circuitos integrados). Utiliza únicamente dos hilos para la transferencia de información entre los elementos que se acoplan al bus que son: línea bidireccional de datos SDA (dato serial), y reloj de sincronización SCL (Reloj Serial).

El módulo I2C del DsPIC 30F4011 provee un hardware que permite trabajar en modo Esclavo y Multi-Master, con interface de 16 Bits. Este módulo posee las siguientes características:

a) El módulo I2C, soporta operaciones de Maestro y Esclavo.
b) El modo I2C Esclavo, soporta direcciones de 7 y 10 bits.
c) El modo I2C Maestro, soporta direcciones de 7 y 10 bits.
d) El puerto I2C permite transferencia bi-direccional entre Maestro y Esclavo.

El módulo de comunicación CAN, es una interfaz serial útil para permitir comunicaciones múltiples con otros módulos CAN o microcontroladores dentro de ambientes ruidosos, el protocolo CAN 2.0A/B con especificaciones definidas por BOSCH, esta conformado de un protocolo para motores y un buffer mensaje/control, maneja motores mediante funciones para recibir y transmitir mensajes. Durante la transmisión los mensajes son cargados a un registro de datos la información que se va a intercambiar se descompone en mensajes, a los cuales se les asigna un identificador y se encapsulan en tramas para su transmisión. Cada mensaje tiene un identificador único dentro de la red, con el cual los nodos deciden aceptar o no dicho mensaje, se puede verificar el estado de los datos y errores en la comunicación mediante los registros destinados para estos propósitos.

1.4.7 Módulo encoder de cuadratura

Este módulo nos ofrece una interfaz para manejo de un encoder incremental, para obtener datos que determinan la posición de un mecanismo rotatorio. Entre las características del módulo se encuentran:

a) Tres canales de entrada; dos son señales de fase y un pulso para referencia de inicio de giro.
b) Contador de posición incremental y decremental de 16 bits.
c) Medición de posición en modo x2 y x4.
d) Filtro digital programable para ruido.
e) Modo Timer/Counter de 16 bits.

1.4.8 Módulo control de motor PWM

El módulo PWM facilita la generación y sincronización de modulación de ancho de pulso, en el dsPIC30F4011 permite dos tipos de modulación de ancho de pulso estas se encuentran agrupadas en modulación PWM de uso común, y el PWM para el control de motores, para el primer tipo utiliza canales de comparación de cuatro salidas PWM denotados por OCx, y el siguiente dispone canales PWMxL y PWMxH, una característica particulares de este tipo de modulación es el poder trabajar de manera independiente o complementaria.

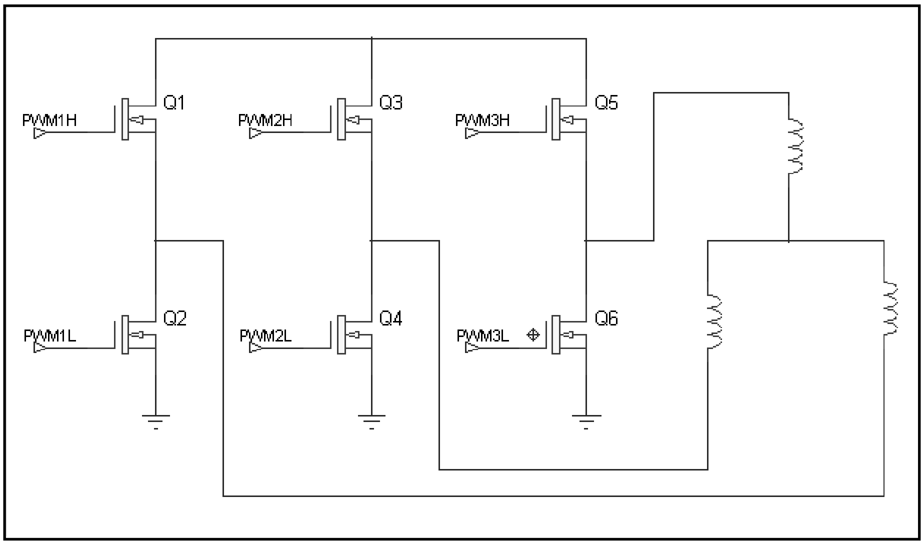
El ciclo de trabajo del PWM tiene tres registros de funciones especiales de 16 bits, usados para especificar el valor del ancho de pulso (PDCx), dicho registro esta asociado con los generadores PWMxL y PWMxH. Este módulo permite obtener señales PWM para el control de motores a partir de formas de ondas de dos tipos: triangulares y rampa, además permite actualizar continuamente el valor del registro de trabajo. Opcionalmente se dispone de una salida PWM Override, esta permite al usuario manejar las salidas PWM de manera manual e independiente los estados lógicos de las mismas.

1.5 Campos de aplicación

1.5.1 Generalidades

Los módulos implementados cuentan con un potente microcontrolador desarrollado específicamente para el control de motores entre las más interesantes aplicaciones se encuentran el control de motores de inducción trifásico, síncronos de reluctancia variable, corriente continua, por lo cual la tarjeta dispone de salidas externas (ver tabla 1.1), las cuales podrán ser usadas para la conexión de módulos de control externo de los motores antes mencionados. Un diseño básico del control de un motor de trifásico AC se muestra en la figura 1.5

Figura 1.5 PWM modo complementario controlando motor trifásico AC



FUENTE: *Microcontroladores Avanzados, México Autor José María Angulo*

Para el control de motores se hará uso de la integración de las siguientes aplicaciones desarrolladas, una esta basada en el manejo de un encoder de cuadratura muy útil para conocer parámetros de rotación del motor, y la otra aplicación esta basada en el control de motores mediante modulación PWM.

1.5.2 Codificador de encoder de cuadratura

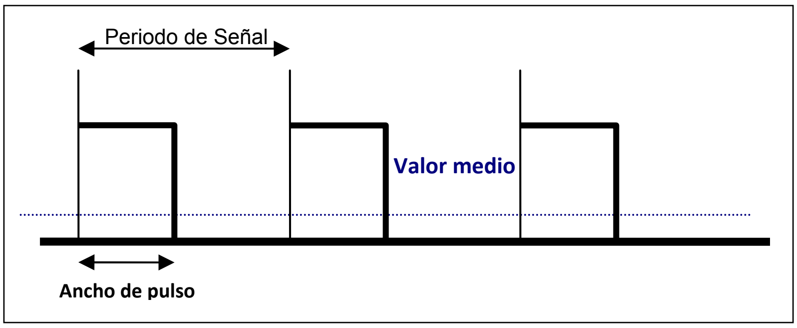
Un encoder es un sensor electro-opto-mecánico que unido a un eje, proporciona información de la posición angular. Su fin, es actuar como un dispositivo de realimentación en sistemas de control integrado, el encoder de tipo incremental es muy utilizado para la detección de la posición y la velocidad de movimientos rotativos como en los motores. Este transductor esta formado por una rueda ranurada en su periférica, conectada al eje del motor y un módulo detector/emisor formados por diodos leds y fotodiodos, los cuales reciben el haz luminoso del led cuando entre el emisor y el receptor se cruza una ranura del disco, estas señales llamadas fase A, fase B, y una señal adicional llamada pulso Z, nos dan la información que permite calcular la dirección del giro del eje y su velocidad. La frecuencia de fase A, fase B depende de la velocidad de giro, y el sentido se identifica observando el signo de la diferencia de fases, es decir cuando la fase A adelanta la fase B, entonces la dirección o giro del motor es positiva (hacia delante), si la fase A esta atrasada de la fase B, entonces la dirección o giro del motor es negativa (hacia atrás), mientras que la señal pulso Z indica un pulso por revolución. La principal limitación de este tipo de encoder es que después de un corte de energía, la posición absoluta es desconocida.

1.5.3 Control de motores DC mediante PWM

El movimiento de motor eléctrico se consigue mediante la variación continua de un campo eléctrico o magnético, dependiendo del tipo de motor, para obtener así un campo rotatorio, esta variación continua se consigue alimentando adecuadamente los embobinados del rotor o del estator, según el tipo de motor, y variando dicha corriente de alimentación se consigue variar la velocidad del motor, un método utilizado para lograr este propósito es mediante PWM.

PWM son las siglas de “pulse width modulation” o “modulación por anchura de pulsos”, el dsPIC incorpora esta función hardware que se puede utilizar para conseguir una salida analógica a partir de un a partir de la variación del valor eficaz una señal digital, esto se consigue modulando la anchura de pulso de un tren de onda cuadrada, (ver figura 1.6).

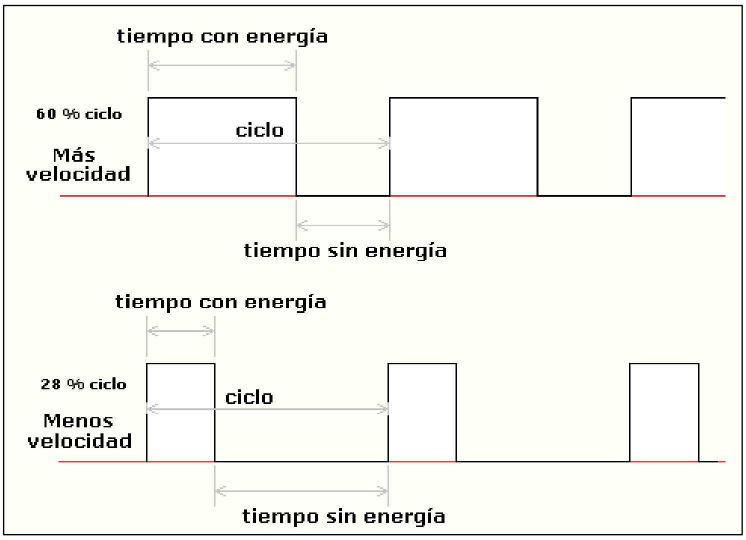
Figura 1.6 Valor eficaz de potencia entregada al motor



FUENTE: *Personal*

Para un pulso más ancho el valor eficaz de la señal es mayor que para un pulso más estrecho, por lo tanto variando la anchura del tren de pulsos de la señal digital se puede conseguir una señal cuyo valor eficaz (valor medio) varíe de forma deseada, es decir la regulación de motores DC está basada en el control de la alimentación entregada al motor. Mediante la modulación PWM se controla la corriente de alimentación del motor, la energía que recibe el motor será proporcional a la relación entre la parte alta (habilita corriente) y baja (cero corriente) del ciclo de la onda cuadrada, es decir, variando la anchura de pulso de la señal, varía la potencia entregada al motor. Controlando esta relación se logra variar la velocidad del motor de una manera bastante aceptable.

Figura 1.7 Esquema del ciclo de trabajo del PWM



FUENTE: *Personal*

1.6 Entornos de desarrollo

1.6.1 Elección del compilador

En el mercado existen muchas alternativas para la programación de los microcontroladores, pudiendo contar con herramientas de desarrollo y librerías compatibles con la arquitectura de los dsPIC.

Empresas como Microchip, Hitech PICc, CCS, Mikroelectrónica están entre las compañías fabricantes más conocidos presentando cada una su gama de compiladores, en este proyecto se escogió como compilador mikroBasic un producto de Microelectrónica basado en lenguaje BASIC para microcontroladores PIC y los dsPIC, que esta conquistando cada vez a más programadores y que lleva tiempo en el mercado.

La empresa Mikroelectrónica distribuye una serie de compiladores para microcontroladores, entre los que se destacan el mikroC y mikroBasic. La características más destacadas de estos compiladores es la inclusión de un IDE (entorno de desarrollo integrado o en inglés Integrated Development Environment) que hace muy cómoda la programación, ya que resalta la sintaxis del lenguaje, proporciona un acceso muy rápido a la excelente ayuda incluida, estadísticas sobre el uso de recursos del microcontrolador, y muchas ventajas adicionales.

Un recurso que nos ofrece Mikroelectrónica es que nos permite descargar una versión gratuita del compilador, que a pesar de estar limitado en la generación de código a 2Kb., es más que suficiente para muchos proyectos, y sobre todo, sirve perfectamente para que podamos ejecutar nuestro proyecto de tópico. Probablemente mikroBasic sea el entorno que soporta más modelos de micros y además dispone de un enorme grupo de librerías, divididas en comunicaciones RS-232, RS-485 e

I2C; teclados PS/2, conexiones USB, interfaz para pantalla de cristal líquido, etc.

1.6.2 Entorno del compilador elegido

Respecto de la organización interna del programa, debemos saber que es necesario que las partes que componen el programa (funciones, rutinas, etc.) sigan ciertas reglas en su escritura. No es necesario que todas estén presentes. Las secciones que son obligatorias son “program” y el bloque “main-end”. Las demás opciones, solo las usaremos si las necesitamos. A continuación se muestra la organización de un programa realizado en mikroBasic:

```
Program  <Nombre del programa>
```

```
Incluye <Nombre de librerias>

'Declaraciones
'(globales, disponibles en todo el programa):

'Simbolos
 symbol ...


'Variables
 dim ...


'Constantes
 const ...


'Procedimientos
 sub procedure nombre_del_procedimiento(...)
 ...
 end sub


'Funciones
 sub function nombre_de_la_funcion(...)

 <declaraciones locales>

 end sub
```

```
'Programa principal:
 main: 'Aquí escribimos nuestro código
 end.
```

Las instrucciones más elementales de este programa son:

Comentarios: el código usando el apostrofe **“ ' ”.** Estos pueden ir solos en una línea, o a continuación de una instrucción. Su utilidad es poder realizar una descripción detallada de lo que realiza determinadas instrucciones, por ejemplo:

```
ADPCFG=$FFFF          'Digitales RB15 a RB0
```

Identificadores: Son los nombres de nuestras variables, procedimientos, funciones, constantes, etc. En mikroBasic los nombres de identificadores pueden contener letras desde la “a” hasta “z” y desde la “A” hasta “Z”, el guión bajo (“_”) y los dígitos del “0” al “9”, estas no son “case-sensitive”, es decir que “Total”, “total” y “TOTAL” son nombres de la misma variable, y podemos usar cualquiera de ellos por igual. El primer caracter no puede ser un digito, además, los nombres de los identificadores no permiten palabras reservadas, por ejemplo

Tabla 1.8 Declaración de identificadores en MikroBasic

Correctos	Incorrectos	
Temperatura1	1Temperatura	'Comienza con un digito
aux_12	Aux.12	'Contiene un punto
rele_on	_on	'Comienza con un guión bajo
Humedad	Humedad	% 'Contiene un carácter invalido (%)

Variables: Las variables son objetos cuyo valor puede cambiar con la ejecución del programa. Cada variable tiene un nombre único, que se ajuste a las reglas definidas mas arriba para los identificadores. Deben declararse con **dim** antes de ser usadas, y su ámbito de validez varía de acuerdo a la sección en que se declaran. La sintaxis de **dim** es:

dim lista de variables **as** tipo

La lista de tipos disponibles incluyen: Byte / chart (8 bits, con valores de 0 a 255), short (8 bits, con valores de -127 a 128), Word (2 bytes, o 16 bits, con valores de 0 a 65535), integer (16 bits, con valores de -32768 a 32767), longword / dword (32 bits, 0 .. 4294967295), Longint (32bits con valores desde -2147483648 a 2147483647), y float (32 bits con valores de ±1.17549435082 x10-38 a ±6.80564774407 x1038. Algunos ejemplos validos de **dim** son:

dim i, j, k **as** byte

dim contar, temp **as** word

dim cadena **as** longint[100]

En el anexo B de describe la instalación y creación de un nuevo proyecto en mikroBasic.

1.7 Adquisición de datos por LabVIEW

1.7.1 Generalidades de LabVIEW

La plataforma de LabVIEW ("Laboratory Virtual Instrument Engineering Workbench") es una aplicación para el desarrollo de programas empleando un lenguaje de programación gráfico (G) creando programas en forma de diagramas de bloques. Los programas compilados en este tipo de ambiente son llamados instrumentos virtuales (VI) porque en su apariencia y operación es similar a instrumento físico real. Un VI se compone de dos partes, ordenadas en pantallas diferentes, uno es el tablero frontal ("front panel") y el otro el diagrama de bloques ("block diagram"). El tablero frontal, es la parte interactiva del programa con el usuario final, personalizado utilizando controles e indicadores tales como: perillas, botones, indicadores, pantallas gráficas y dispositivos de entrada. Después de construir la interfaz de usuario, se debe programar el diagrama de bloques, usando los VI incorporados en las librerías LabVIEW, siendo este diagrama de bloques el código fuente del panel frontal.

1.7.2 Principios de la comunicación con LabVIEW

LabVIEW posee herramientas para controlar la mayor parte de los protocolos de comunicación usados en las computadoras, como el

TCP, UDP, serial, paralelo, IrDA, Bluetooth o SMTP, su capacidad para realizar transmisión de datos (distancia, voltaje y corriente de salida) está limitada por las características propias del protocolo de comunicación usado, del tipo de puerto y del dispositivo de salida conectado directamente a la computadora. Los datos pueden introducirse usando el ratón o el teclado de la computadora en el tablero frontal y los resultados del proceso pueden ser vistos en la pantalla de la computadora.

1.7.3 Comunicación y adquisición de datos con RS-232

La tarjeta desarrolladora nos permite mediante comunicación serial la interacción entre LabVIEW y datos provenientes del exterior, este tipo de instrumentación virtual aumenta la productividad y disminuye los costos para aplicaciones de medición dentro del entorno industrial, siendo posible utilizar un VI en aplicaciones donde se requiere del procesamiento de entradas, como visión, comunicaciones con “hardware” industrial pudiendo controlar y monitorear estos procesos, con ello es posible realizar adquisición de datos, analizar mediciones y generar reportes de los resultados de sus aplicaciones por medio de la creación de archivos, además es posible establecer comunicación con dispositivos externos para control conectados directamente al computador.

CAPÍTULO 2

2. DISEÑO DE LA TARJETA DESARROLLADORA

2.1 Consideración de diseño

En este capítulo se describirá la estructura de la tarjeta desarrolladora, los periféricos implementados en la misma, los tipos de motores que se han usado para el desarrollo de prácticas basadas en el control en lazo abierto mediante señales provenientes del dsPIC para la regulación de velocidad y giro en motores de corriente continua, de paso, y servomotores, implementados en la tarjeta desarrolladora. Se debe también recalcar que no solamente se puede trabajar con motores de corriente continua sino que además la tarjeta dispone de salidas externas en las cuales se puede incorporar un módulo que permita trabajar con motores trifásico de corriente alterna, esta una de las posibles aplicaciones a desarrollarse a futuro.

Respecto a la selección del microcontrolador a usarse en el desarrollo de la tarjeta entrenadora se ha elegido el dsPIC30F4011 de gama media, su encapsulado exterior luce como cualquier microcontrolador de 40 pines, sin embargo en su interior la historia es totalmente diferente, este potente dispositivo que posee millones de circuitos eléctricos completamente estructurados ofreciéndonos: velocidad, potencia, manejo flexible de interrupciones, un amplio manejo de funciones periféricas, además de las prestaciones del DSP, esta diseñado específicamente para soportar aplicaciones de control de motores, alcanzando un nivel de rendimiento muy cercano al de los microcontroladores de 32 bits y a los DSP, siendo su precio muy ventajoso, constituyéndose en un nuevo comienzo del control embebido.

2.2 Especificaciones de hardware

2.2.1 Unidad de programación

El microcontrolador seleccionado además de ofrecer velocidad de procesamiento y precisión de resultados, tiene la alternativa de poder usar un programa de carga conocido como bootloader que ocupa cierta posición de la memoria de programa y cuyo único objetivo es el de permitir la carga de programas ejecutables. De tal forma que un programador externo se utilizará sólo una vez para cargar el bootloader, luego de esto el dsPIC es ubicado nuevamente en la tarjeta y el método de programación del microcontrolador es a través del puerto de comunicación serial (UART1 alternativo), quien de allí en adelante se encargará de las tareas de programación, este método es una de las alternativas para la programación del dsPIC usado en las tarjetas.

El bootloader utilizado con el dsPIC30F4011 se ubica en las últimas posiciones de la memoria flash de programa del microcontrolador y reposiciona los vectores de interrupción mediante punteros. Arranca cuando alimentamos el procesador y espera un comando por un puerto de comunicación si no lo recibe continúa con la ejecución normal de nuestro programa. Si lo recibe comienza a recibir el programa del usuario a través del puerto de comunicación y a grabarlo en la memoria flash de programa del microcontrolador de la tarjeta. Esta característica es especialmente importante en un laboratorio educativo donde se tendría acceso directo a la disponibilidad de programación sin necesidad de extraer el microcontrolador de la tarjeta desarrolladora a un módulo dedicado únicamente a la programación, brindándole al diseñador más seguridad y facilidad al momento de cambios en el programa, además de ser un ahorro económico al momento de elegir la unidad programadora del dsPIC, esto es un típico problema en los laboratorios de desarrollo porque muchas veces no se cuenta con los programadores suficientes y necesarios donde un mismo dispositivo dsPIC debe ser grabado una y otra vez para la realización de pruebas, por no disponer de un software que simule el proyecto que se realiza, el proceso de carga por medio de bootloader se explica en el anexo A. Sin embargo de presentarse la necesidad de realizar la programación del microcontrolador haciendo uso del programador externo Pickit2 se ha implementado en la tarjeta de entrenamiento con dsPIC30F4011 la interfaz ICD2 quedando disponible un conector RJ12 (6 pines) para su respectiva conexión.

2.2.2 Fuentes de entradas analógicas y digitales

La tarjeta dispone de entradas digitales y analógicas, estas entradas comparten determinados pines de un mismo puerto del dsPIC que puede ser configurado como digital o analógico. Dado que comparten la misma entrada, es necesario disponer de algún mecanismo que permita elegir una entrada o la otra, o incluso ninguna en el caso de utilizar señales externas. Para ello se han utilizado puentes que permiten elegir las tres posibilidades: analógico, digital y desconectado. Hay 3 fuentes internas de señales analógicas: dos de ellas son potenciómetros que permiten seleccionar cualquier tensión entre 0 y 5 Voltios. La última fuente analógica disponible viene de un sensor de temperatura, que entrega una salida lineal con la temperatura. Con relación a las fuentes digitales de entradas se han implementado mediante 4 interruptores, y 3 micropulsadores.

2.2.3 Barra de 10 diodos emisores de luz

Se ha elegido una barra de diodos emisores de luz (leds) para reducir el espacio en la tarjeta, la barra ofrece 10 salidas digitales, y esta conectado un diodo por cada bit de un puerto.

2.2.4 Visualizador múltiple de 7 segmentos

Se dispone de cuatro visualizadores de 7 segmentos, existen varias posibilidades de control de este tipo de visualizador múltiple, pero se ha optado por la multiplexación, ya que permite introducir conceptos como la realización de interrupciones de refresco, durante la ejecución normal de un programa.

2.2.5 Teclado matricial y minidin PS/2

Es importante disponer de entradas por medio de teclados, el de tipo matricial esta normalmente codificado ubicando las teclas en una matriz de filas y columnas. Todas las filas y columnas son muestreadas periódicamente por un controlador a una frecuencia suficientemente alta para poder percibir cualquier cambio en el teclado.

Una de las alternativas que ofrece la tarjetade entrenamiento es el manejo de teclado matricial, que ofrece 2*n teclas por cada n bits. También se encuentra disponible una entrada PS/2, en la cual se tiene la posibilidad de conectar un teclado del mismo tipo.

El teclado 4x4 correspondiente a 16 teclas (de igual manera se puede conectar un teclado 4x3), está conectado a un puerto del microcontrolador. Los 4 bits más significativos serán conectados a las filas, y los 4 menos significativos a las columnas. En tanto el teclado PS/2 únicamente utiliza dos pines del microcontrolador, este dispone de 104 teclas para PC es un dispositivo razonablemente inteligente. Esto quiere decir que él mismo se encarga de la parte hardware del rastreo de teclas, de evitar los insufribles rebotes y de transmitirlos, mediante una trama perfectamente establecida, de los resultados de nuestras manipulaciones sobre sus teclas. Al presionar una tecla, se produce un contacto eléctrico entre la fila y la columna, las señales eléctricas son transmitidas a un control interno del teclado, que envía un código (BCD, ASCII o Unicode) al dsPIC describiendo el caracter que corresponde a dicha tecla.

2.2.6 Pantalla de cristal líquido

El último paso hacia un visualizador de mediana complejidad lo constituye la pantalla de cristal líquido. Hay otras pantallas más sofisticadas, pero hay pocas que sean tan versátiles, estándares, económicas, y fáciles de usar. La pantalla de la tarjeta es de 16X2 y está conectada a los puertos del microcontrolador, distribuyéndose entre ellos el bus de datos y las señales de control.

2.2.7 Motores de uso común

Existen gran cantidad de métodos para producir el movimiento físico de un mecanismo. Sistemas neumáticos que trabajan a partir de aire presurizado, los sistemas hidráulicos que utilizan fluidos a presión y los sistemas eléctricos que utilizan motores para convertir la energía eléctrica en energía mecánica (movimiento). Los motores son máquinas capaces de transformar la energía almacenada en combustibles, baterías u otras fuentes, en energía mecánica capaz de realizar un trabajo. En este proyecto final de tópico se usará motores de uso frecuente en aplicaciones para microcontroladores: DC de imán permanente, con bobinado en serie, el motor de pasos, y servomotor.

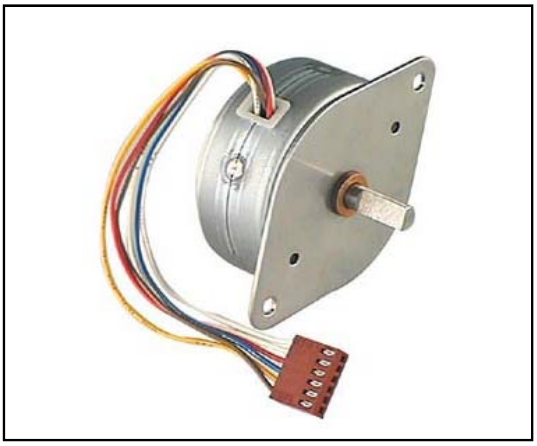
El motor DC o de corriente continua de pequeño tamaño, son ideales para nuestro caso; no sólo por su reducido costo, sino también, por su razonable eficiencia, versatilidad y gran aplicabilidad a los procesos industriales. Los motores DC, son ampliamente utilizados en aplicaciones tales como la robótica, mecatrónica y en sistemas de control. Estos motores serán los encargados del movimiento y control de dirección de las bandas transportadoras usadas en él tópico, por lo que es necesario controlar la velocidad y el sentido de giro.

Figura 2.1 Motor DC

FUENTE: *Foto personal*

El motor de pasos llamado también stepper motor, es un motor alimentado con tensión continua (DC) y convierte los pulsos eléctricos aplicados en sus bobinas en movimientos rotacionales discretos. Pueden girar continuamente o sólo un ángulo llamado paso. Debido a esto resultan muy precisos y confiables, se emplean comúnmente en aplicaciones donde el posicionamiento mecánico resulta ser muy importante. Son ideales cuando se tiene que girar un eje y detenerlo en cierta posición con una precisión de hasta centésimas de milímetros. En ésta posición puede permanecer bloqueado, hasta recibir una nueva orden de giro. Una de sus principales ventajas es que no necesitan de sensores para determinar la posición del eje, ya que se conoce la posición por la secuencia de pulsos que se han enviado a las bobinas.

Figura 2.2 Motor de paso a paso bipolar

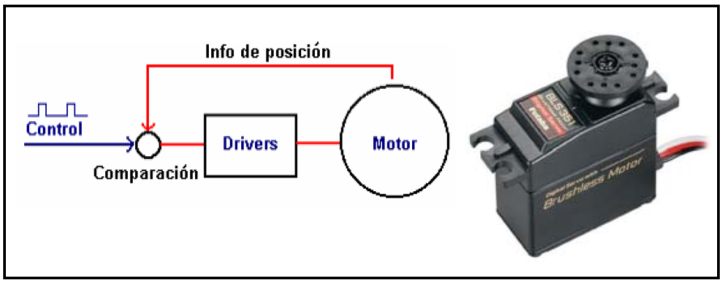


FUENTE: *Foto personal*

Un servomotor, es un motor de corriente continua (DC) especializado, el cual, puede ser comandado para que gire a una posición determinada. Tiene un circuito de "control" construido en su interior que se realimenta en todo momento con la posición de su rotor. De ésta forma puede llevar a cabo una auto-regulación. En lugar de rotar continuamente como los motores comunes de DC, un servomotor puede ser posicionado enviando las señales apropiadas al servo a un punto específico y éste quedarse fijo en esa posición.

Los servos poseen un conector para 3 cables: Uno de alimentación, uno de tierra y el cable de control. En el servomotor FUTABA dispone de tres alambres son de color rojo, negro y blanco, dos de los cuales son alimentación (rojo y negro), que puede ir desde 4.8V a 6.0V, en este caso se trabaja con un voltaje de 5V, el tercer alambre (blanco) es de control conectado al dsPIC el mismo que genera una señal necesaria para el funcionamiento del servo.

Figura 2.3 Servo motor y su control interno de lazo cerrado



FUENTE: *Foto personal*

2.3 Diseño de controladores para motores

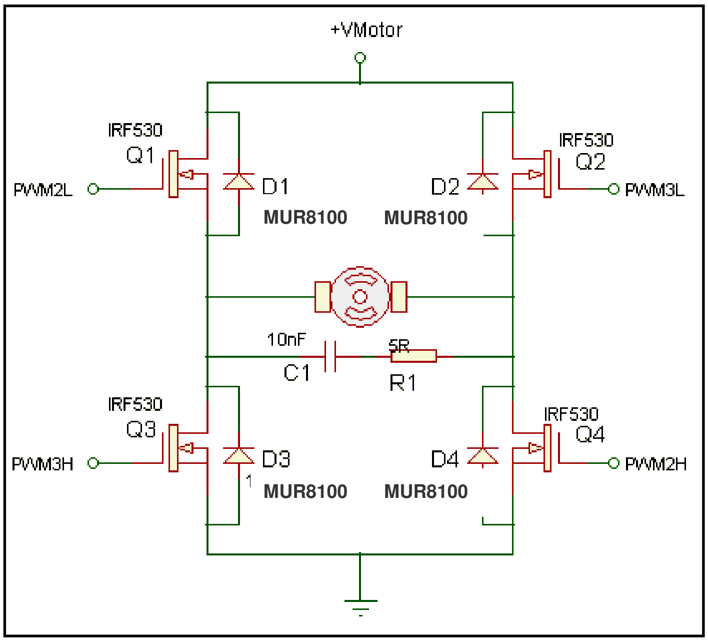
2.3.1 Manejador de motores

Los manejadores para motores son usados en sistemas que manejan pequeñas cantidades de vatios hasta cientos de kilovatios, existen varios métodos para regular la velocidad de los motores eléctricos: regular tensión, intercalando resistencias, modulación de la tensión de alimentación mediante el procedimiento de modulación por anchura de pulsos (PWM), en motores de corriente alterna se puede actuar sobre la frecuencia. En este proyecto de tópico se ha usado la modulación por anchura de pulsos (PWM) a través de un puente “H”, debido a su rendimiento y control dinámico que muchas veces superan a los manejadores comunes para motores dc.

2.3.2 Estrategias para el manejo de motor DC

El circuito de control para el manejo del motor DC está basado en un arreglo de un puente “H" constituido por cuatro transistores Q1, Q2, Q3, y Q4 de tipo mosfet canal N controlados por el dsPIC30F4011 implementado en la tarjeta, los transistores trabajan en pares. Los transistores que realizan la conmutación son habilitados por la compuerta, a través de una señal PWM que ha sido configurada en software para que esta señal que entrega el dsPIC trabaje en modo independiente es decir, los pines PWM2L y PWM2H del dsPIC, envían la misma señal a los transistores Q1 y Q4, mientras que los transistores Q2 y Q3 permanecen deshabilitados, haciendo que el motor gire en sentido horario a una determinada velocidad correspondiente al voltaje eficaz suministrado al motor, (ver figura 1.6). De manera análoga al procedimiento anterior se puede hacer girar el motor en sentido antihorario habilitando los transistores Q2 y Q3, mientras los transistores Q1 y Q4 permanecen deshabilitados, en la figura 2.4 se ilustra el circuito para el manejo de un motor DC.

Figura 2.4 Esquema para manejo de motor DC

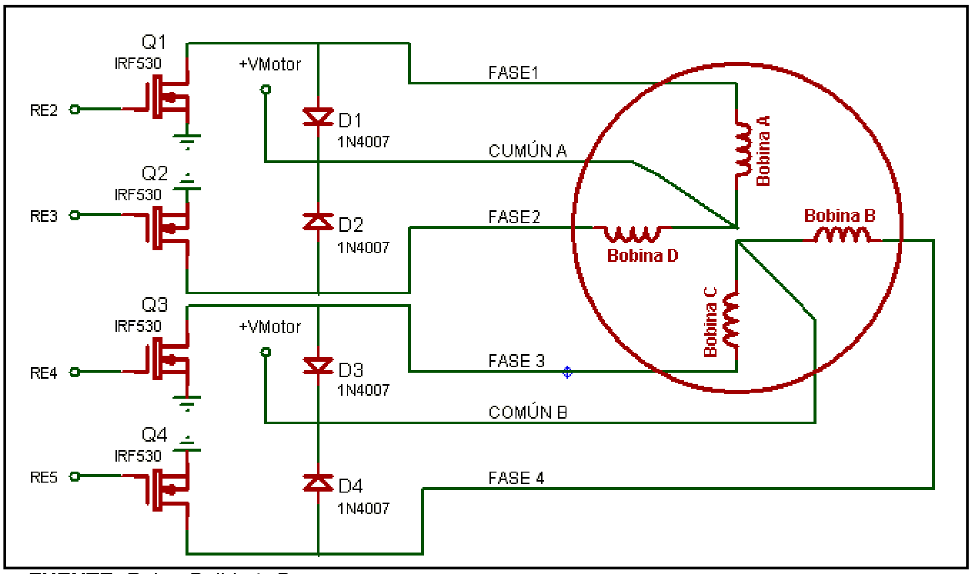


FUENTE: *Robot Builder's Bonanza*

2.3.3 Estrategias para el manejo de motor de paso

Para controlar el movimiento del motor de paso unipolar empleado en el proyecto, se ha realizado un circuito de control basado en transistores mosfet canal N, que será el responsable de convertir las señales de avance de un paso y sentido de giro en la necesaria secuencia de energización de los bobinados ubicados en el estator del motor, logrando producir el avance por pasos del motor, si se invierte el orden de esta secuencia, se logra que el motor gire en sentido opuesto, si los pulsos de alimentación no se proveen en el orden correcto, el motor no se moverá apropiadamente. La conmutación de los transistores es controlada externamente por el dsPIC30F4011, en la siguiente figura se presenta el esquema utilizado para el motor de paso.

Figura 2.5 Esquema para manejo de motor de paso



FUENTE: *Robot Builder's Bonanza*

Como se mencionó al principio es necesario que el motor reciban una determinada secuencia de energización de sus bobinados, existiendo tres posibilidades para realizarlo: secuencia normal, secuencia de bajo consumo de corriente, secuencia de medio paso.

La tabla 2.1 muestra la secuencia normal, siendo esta la más usada y normalmente recomendada por el fabricante, debido a que siempre están encendidas dos bobinas se obtiene un alto torque de paso y de retención.

Tabla 2.1 Secuencia normal de encendido de bobinas

PASO	Bobina A	Bobina B	Bobina C	Bobina D
1	Encendido	Encendido	Apagado	Apagado
2	Apagado	Encendido	Encendido	Apagado
3	Apagado	Apagado	Encendido	Encendido
4	Encendido	Apagado	Apagado	Encendido

***FUENTE**: Personal*

En la siguiente secuencia mostrada en la tabla 2.2 se activa una bobina a la vez, consumiendo menos corriente, aunque también se obtiene un menor torque de paso y de retención.

Tabla 2.2 Secuencia de bajo consumo encendido de bobina

PASO	Bobina A	Bobina B	Bobina C	Bobina D
1	Encendido	Apagado	Apagado	Apagado
2	Apagado	Encendido	Apagado	Apagado
3	Apagado	Apagado	Encendido	Apagado
4	Apagado	Apagado	Apagado	Encendido

***FUENTE**: Personal*

Por último se puede usar la secuencia de medio paso en la cual se activan las bobinas de tal forma se obtiene un movimiento igual a la mitad del paso real, para ello se activan primero dos bobinas y luego solo una y así sucesivamente. A diferencia de las secuencias anteriores que únicamente ofrecen cuatro pasos, esta consta de ocho pasos como se muestra en la tabla 2.3

Tabla 2.3 Secuencia de medio paso de encendido de bobinas

PASO	Bobina A	Bobina B	Bobina C	Bobina D
1	Encendido	Apagado	Apagado	Apagado
2	Encendido	Encendido	Apagado	Apagado
3	Apagado	Encendido	Apagado	Apagado
4	Apagado	Encendido	Encendido	Apagado
5	Apagado	Apagado	Encendido	Apagado
6	Apagado	Apagado	Encendido	Encendido
7	Apagado	Apagado	Apagado	Encendido
8	Encendido	Apagado	Apagado	Encendido

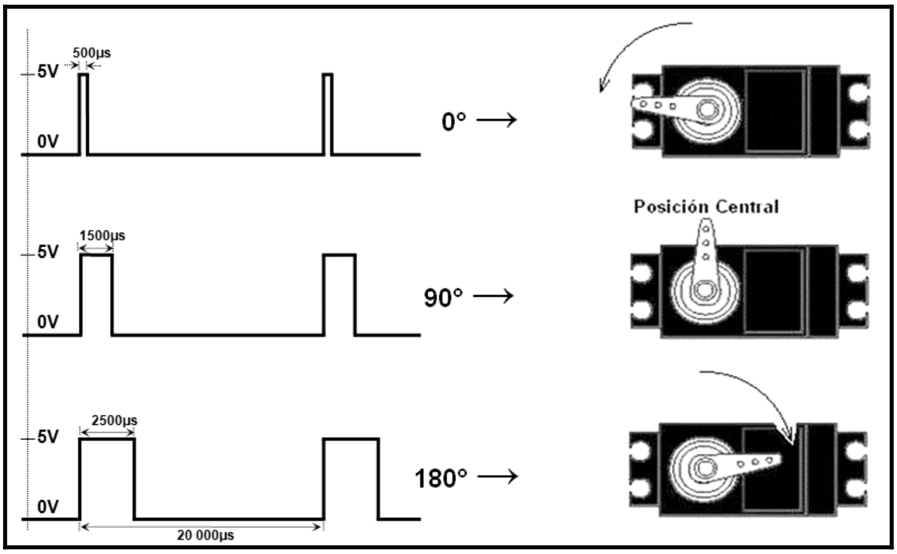
***FUENTE**: Personal*

2.3.4 Estrategias para el manejo del servomotor

Realizar el control de movimiento del servomotor usado en este proyecto resulta una tarea que no requiere de un manejador adicional, porque en su interior se encuentra el diseño electrónico necesario para manejar este tipo de motores, es decir el control de su posición se lo hace enviando un pulso continuo de onda cuadrada proveniente del dsPIC al motor a través del cable de control, el tiempo que permanece en alto este pulso indica la posición en que el servomotor se ubicara, en la tabla 2.4 se indica los tiempos en alto y en bajo que debe de tener la señal cuadrada para que el servomotor se mueva a una determinada posición. Estos motores trabajan a una frecuencia constante de 50 Hz es decir a un periodo de 20000us, por ejemplo si queremos posicionar al servomotor a la posición central (90º), debemos enviar un pulso de onda cuadrada de 1500 us en alto y de 18500 us en bajo, esta señal debe ser periódica para mantener dicha posición.

Si el ancho de pulso disminuye el movimiento del servomotor será antihorario, por ejemplo si el pulso en alto es de 500 us el servomotor se posicionara en cero grados como se lo muestra en la figura 2.6, pero en cambio si el ancho de pulso aumenta el movimiento será horario.

Figura 2.6 Señales de pulso de onda cuadrada y posición del servomotor



***FUENTE**: http:// robots-argentina.com.ar*

Tabla 2.4 Tiempos de la señal de la onda cuadrada para el posicionamiento del servomotor

GRADOS	TIEMPOS DEL PULSO (us)	
	EN ALTO	EN BAJO
0	500	19500
5	556	19444
10	611	19389
15	667	19333
20	722	19278
25	778	19222
30	833	19167
35	889	19111
40	944	19056
45	1000	19000
50	1056	18944
55	1111	18889
60	1167	18833
65	1222	18778
70	1278	18722
75	1333	18667
80	1389	18611
85	1444	18556
90	1500	18500
95	1556	18444
100	1611	18389
105	1667	18333
110	1722	18278
115	1778	18222
120	1833	18167
125	1889	18111
130	1944	18056
135	2000	18000
140	2056	17944
145	2111	17889
150	2167	17833
155	2222	17778
160	2278	17722
165	2333	17667
170	2389	17611
175	2444	17556
180	2500	17500

FUENTE*: Personal*

2.3.5 Elemento de conmutación

El interruptor de potencia usado en la construcción del manejador de la tarjeta de entrenamiento con dsPIC30F4011 es el transistor mosfet IRF530 y el IRF3710 para la tarjeta complementaria manejadora de motores DC, estos conmutadores electrónicos tienen tres terminales identificadas como: compuerta, drenador y surtidor, son dispositivos controlados por voltaje aplicado entre compuerta y surtidor, la corriente que circula por la compuerta es casi nula debido a que esta unida mediante un aislante por lo que tendrá una impedancia muy grande, sin embargo durante las conmutaciones si existirá circulación de corriente por la compuerta. La corriente que entrega el transistor mosfet depende del voltaje aplicado entre compuerta y el surtidor, y no de la corriente que circula por la compuerta como ocurre con los transistores bipolares. Debido a sus características de construcción los mosfet nos permiten que las conmutaciones se realicen en tiempos muy cortos.

2.3.6 Redes snubber RC

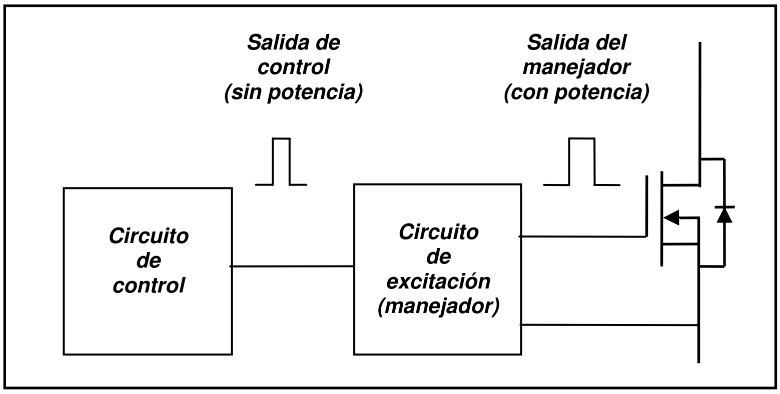
Las redes snubber son circuitos que absorben energía usados para eliminar picos de voltajes causados por las inductancias de los circuitos cuando estos conmutan, sean mecánicos o con un transistor. El objetivo de esta red es proporcionar un camino alterno para que la corriente fluya a través de él. Existen funciones importantes que realiza este circuito entre las cuales se encuentran: crear un camino alterno durante la conmutación para mantener un modo seguro de operación para el transistor, remover la energía del transistor durante la conmutación y disiparla en una resistencia para reducir la temperatura de la juntura.

En cuanto a la selección del capacitor y resistencia de la red se debe elegir una resistencia no inductiva evitando aquellas que estén formadas por espiras, una buena elección es un resistor compuesto de carbón, una resistencia de este tipo es ideal para el circuito. Se debe dimensionar el capacitor para soportar los altos picos de corriente en el snubber, para valores de capacitancia arriba de 0.1uF considerar capacitores de mica, para valores altos de capacitancia considerar tipos polipropileno. En análisis de las dimensiones de la resistencia y capacitor de la red snubber se explica en el anexo D.

2.3.7 Etapa de acoplamiento

Un manejador es un sistema capaz de gobernar eficazmente la conducción y no conducción del interruptor de potencia (mosfet IRF530,3710) partiendo de las ordenes que llegan del circuito de control (dsPIC30F4011), debido a que los circuitos de potencia y los circuitos de control no manejan los mismos niveles de tensión y corriente, será necesario utilizar una etapa de acoplamiento de voltajes, corrientes e impedancias, para ello se utilizó optoacopladores para que el microcontrolador y el mosfet puedan interactuar.

Figura 2.7 Esquema del circuito de acoplamiento para el manejador del motor



Fuente: *Personal*

2.4 Esquemáticos

En la figura 2.8 tenemos el esquemático de la tarjeta básica para aplicaciones con dsPIC, que tiene incorporado borneras externas para acceder a los puertos del dsPIC con la finalidad de facilitar el manejo de señales para el control de nuevas aplicaciones. Respecto a los esquemáticos de la tarjeta de entrenamiento con dsPIC30F4011, se muestran desde la figura 2.9 hasta la figura 2.15 con la distribución de los distintos periféricos y sus conexiones con el microcontrolador, estos esquemáticos describen los diferentes módulos que se encuentran integrados en la tarjeta de entrenamiento como son: la fuentes para el circuito de control y alimentación de los motores, la unidad de control de la tarjeta, conectores externos de la tarjeta para acceso al puerto de la unidad de control, el esquemático de la unidad de comunicaciones básicas implementada en la tarjeta, la unidad de visualización, la unidad de señales digitales y analógicas, finalmente se mostrará la unidad para el manejo de los motores aplicando una técnica de troceador de voltaje que consiste en mantener una diagonal permanentemente abierta, y proporcionando un ciclo de trabajo a los otros dos interruptores (abriéndolos y cerrándolos a la vez), de esta manera se podría cambiar velocidad del motor, y dependiendo de la diagonal utilizada para el troceado de voltaje se cambiaria sentido de giro.

La figura 2.16 muestra el esquemático del manejador de motores DC que permite usar cualquier técnica de troceo clase E, como por ejemplo modulación de ancho de pulso y conmutación de voltaje bipolar, unipolar, entre otras.

Figura 2.8 Esquemático de la tarjeta básica para aplicaciones con dsPIC30F4011

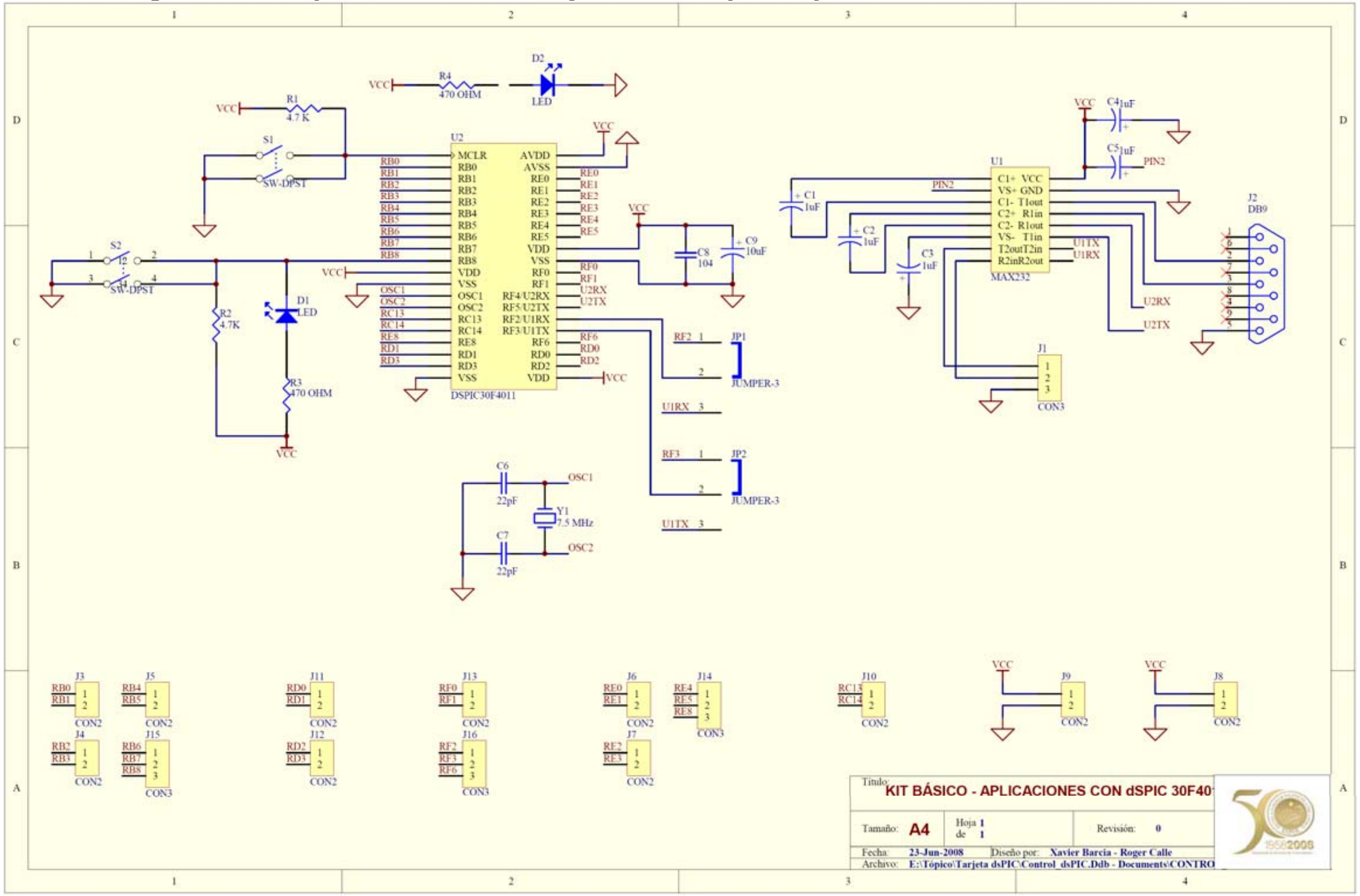


Figura 2.9 Esquemático de fuentes para el circuito de control y alimentación de los motores

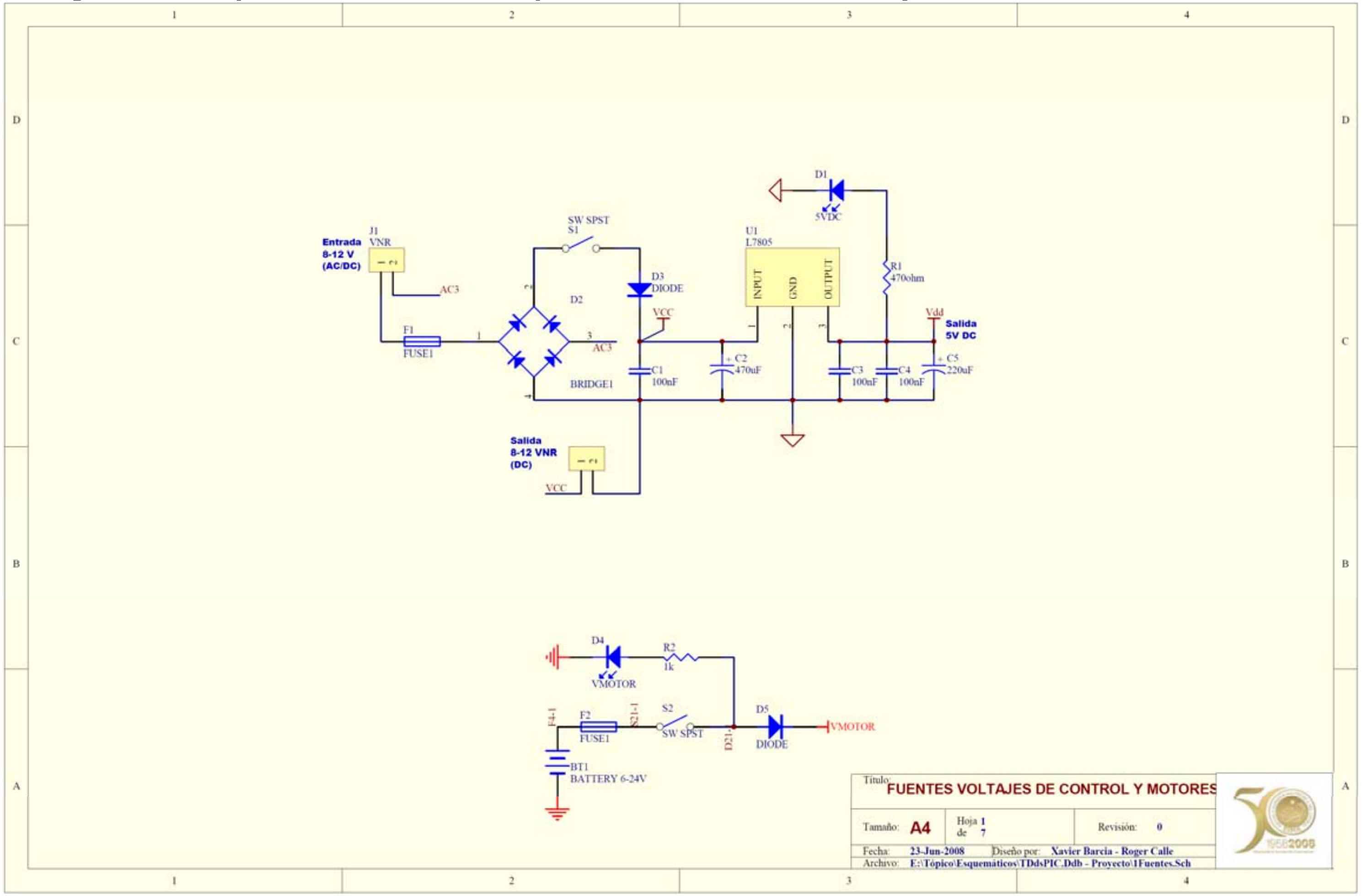


Figura 2.10 Esquemático de la unidad de control de la tarjeta

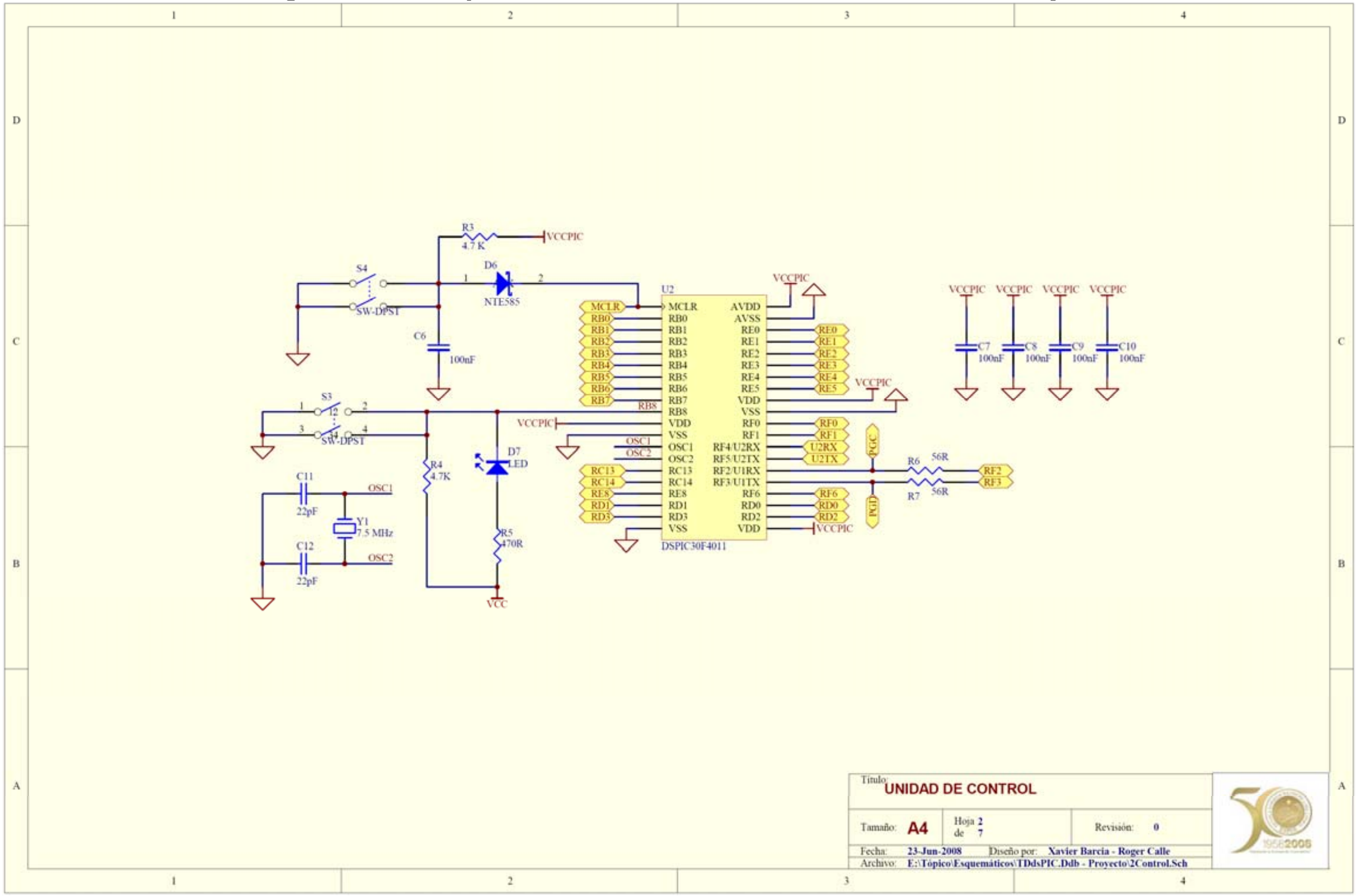


Figura 2.11 Esquemático de conectores externos de la tarjeta para acceso al puerto

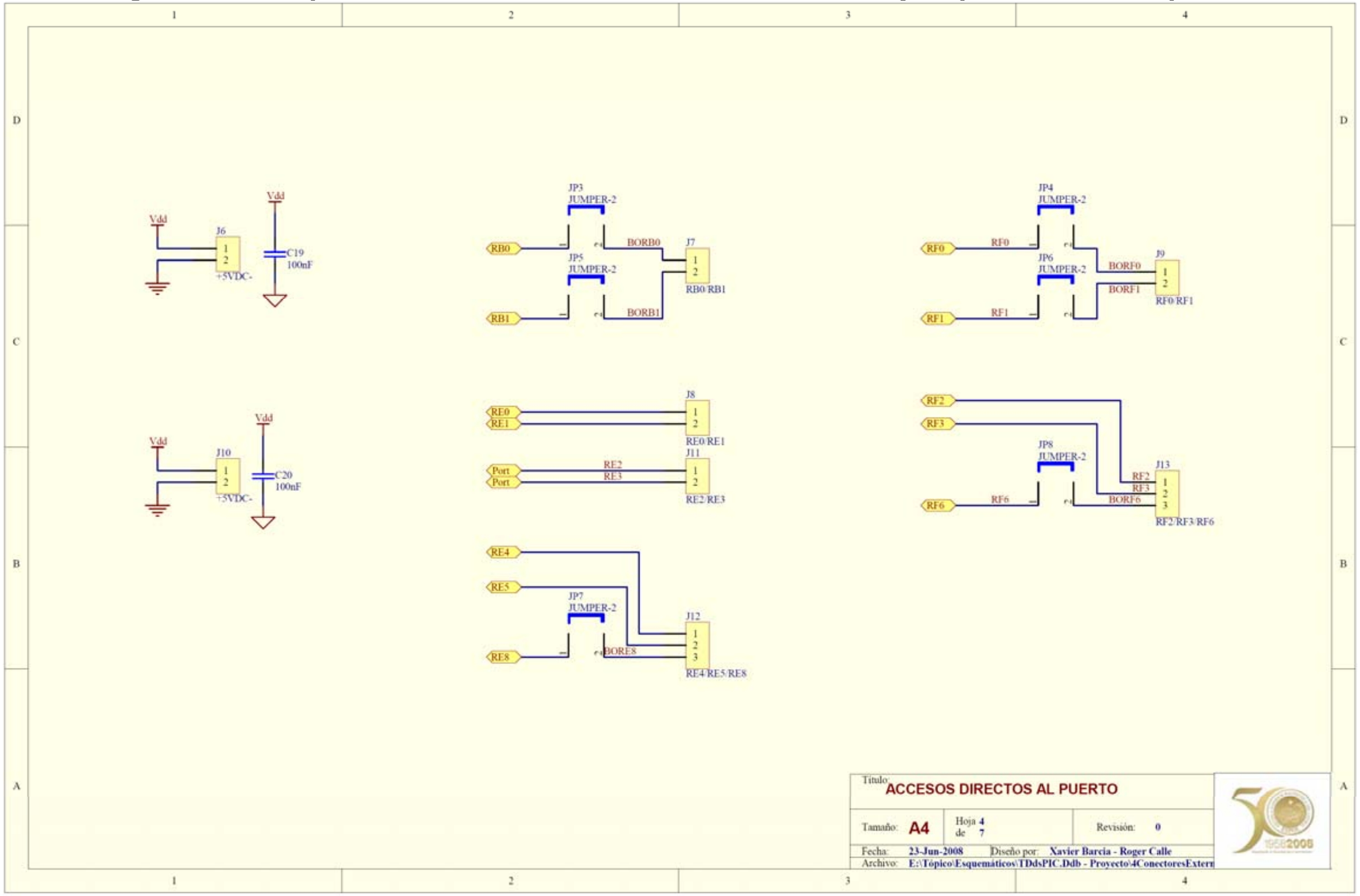


Figura 2.12 Esquemático de la unidad de comunicación

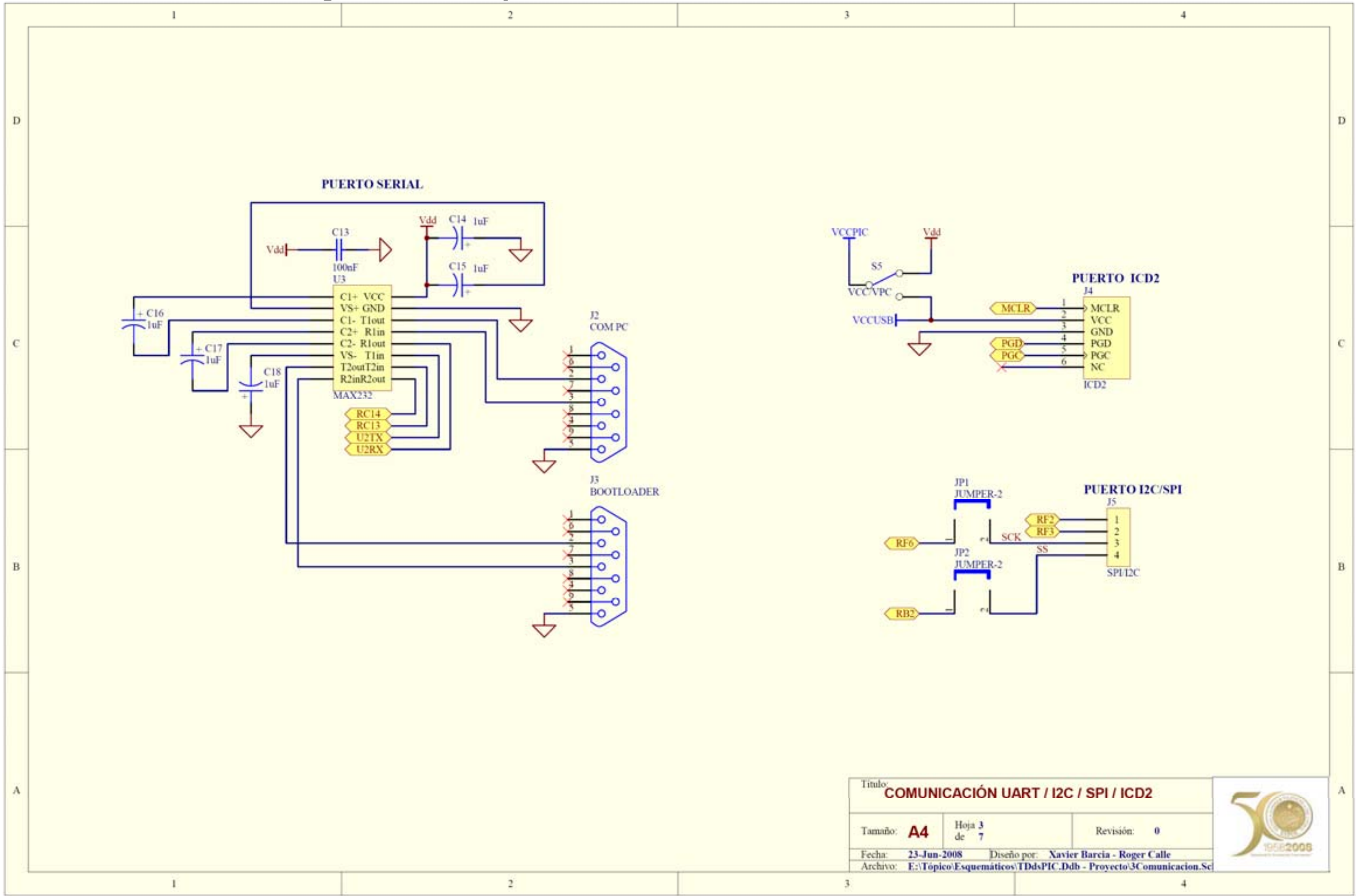


Figura 2.13 Esquemático de unidad de visualización

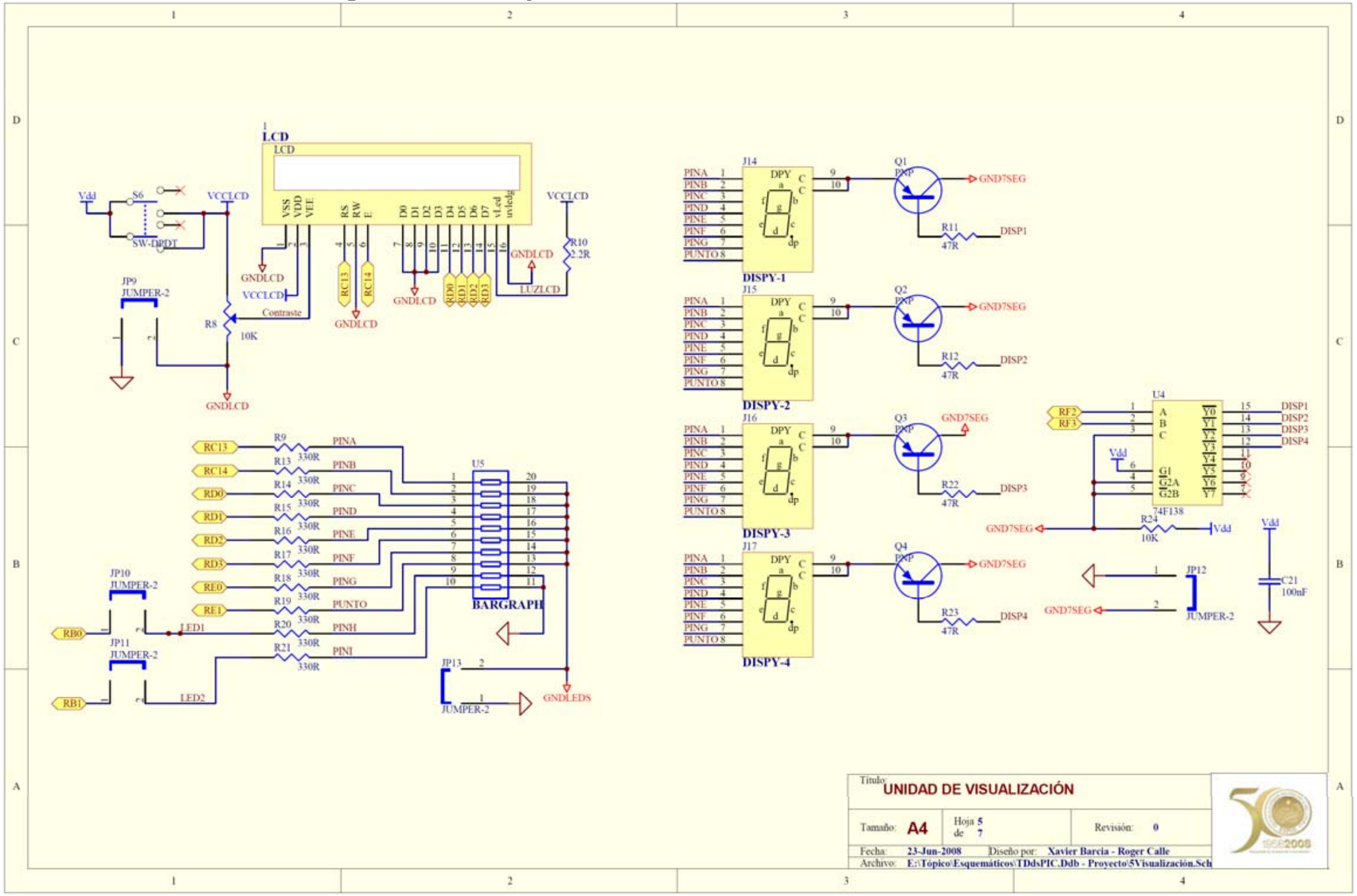


Figura 2.14 Esquemático de la unidad de señales digitales y analógicas

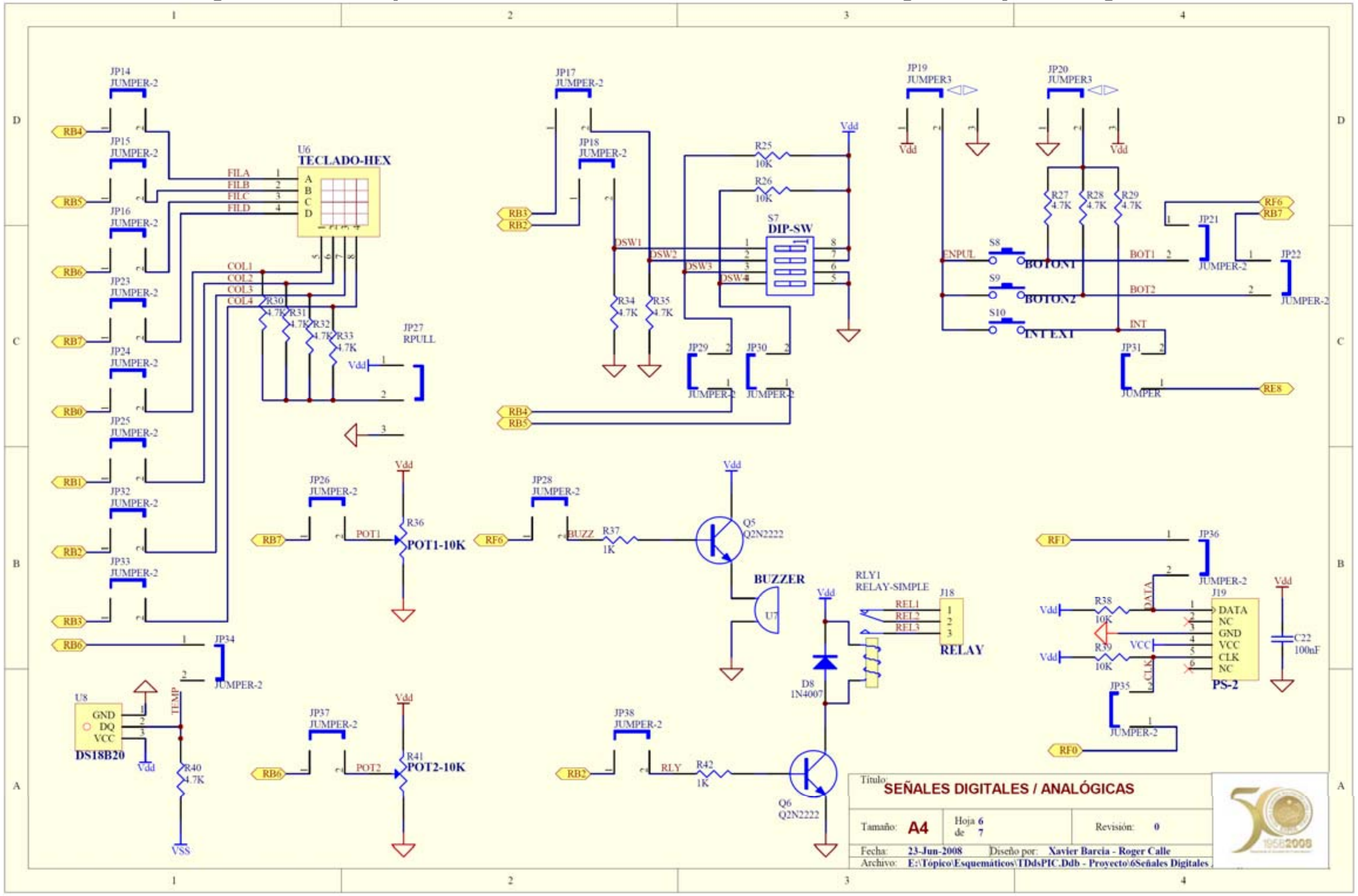


Figura 2.15 Esquemático de la unidad de manejo de motores

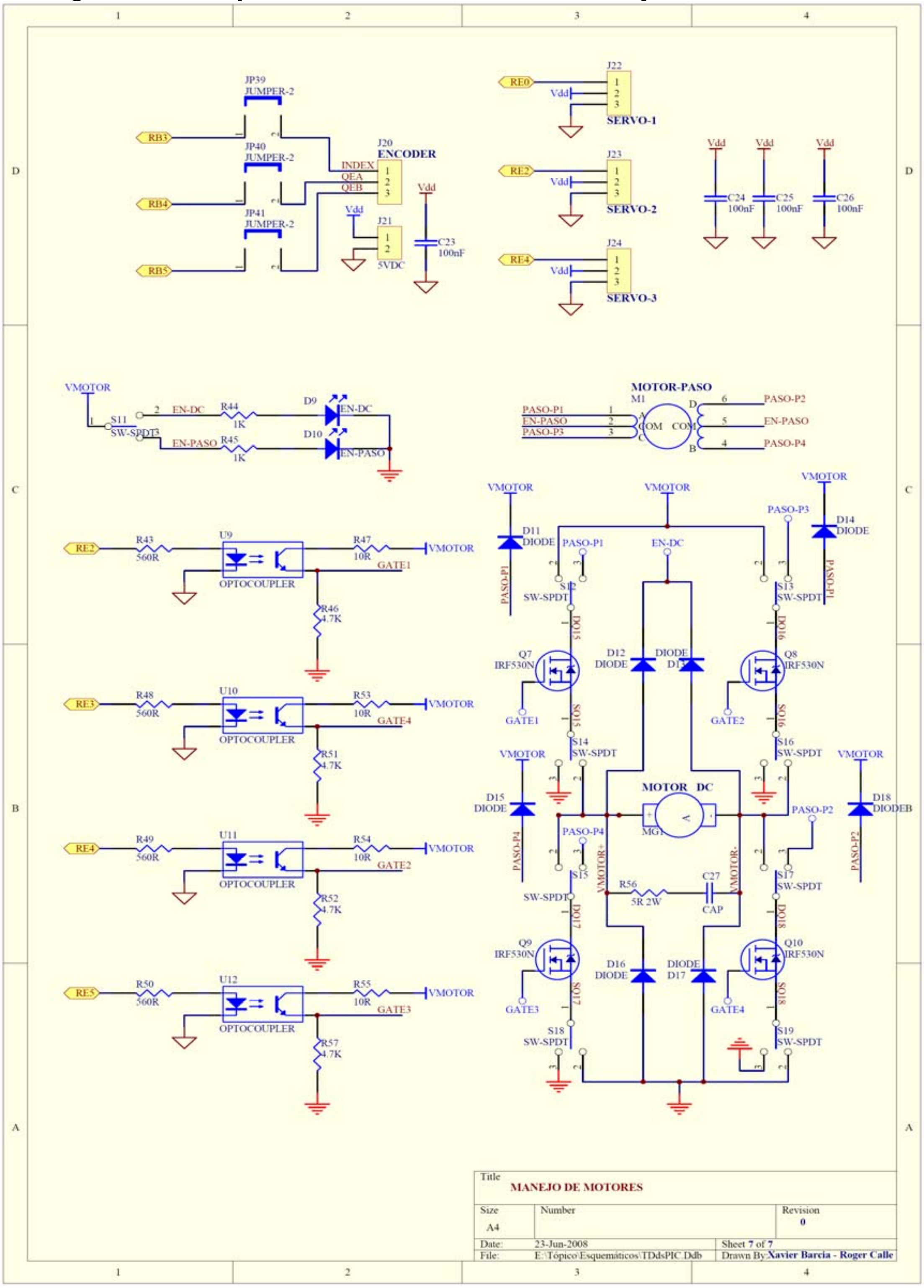
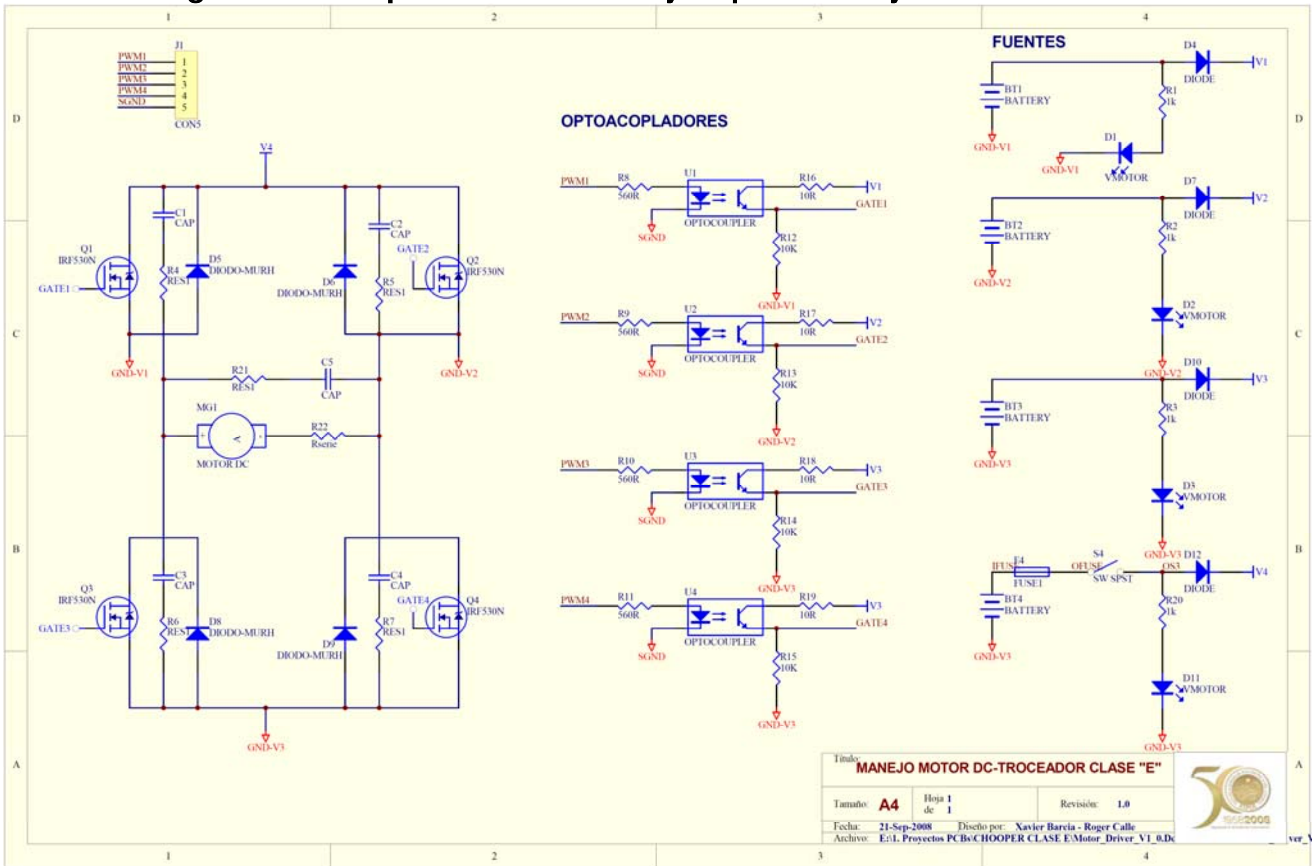


Figura 2.16 Esquemático de la tarjeta para manejo de motores DC



2.5 Diseño del PCB usando herramientas CAD

2.5.1 Antecedentes de diseño de PCB

El tradicional trazado de pistas de cobre se realiza empleando una hoja cuadriculada transparente de guía, sobre la cual se colocaba una cinta opaca con la que se trazan las pistas, el trabajo se lo realiza al doble del tamaño natural y luego se reduce por medios fotográficos, de esta forma cualquier error cometido en el trazado queda también reducido, durante la elaboración del trazado de las pistas es muy común usar múltiples hojas transparentes para las diferentes capas del trabajo, como para el ruteado de las pistas de cobre, las referencia de los componentes, los puntos de taladrado, entre otros; este proceso hace difícil las correcciones posteriores al diseño.

Actualmente la tecnología nos ofrece programas de diseño asistidos por computadora para la concepción de este tipo de trabajos, a diferencia de los métodos manuales de diseño, estas herramientas nos permiten distribuir e interconectar los componentes además de ir realizando pruebas y mejorando el diseño, estos programas almacenan información relacionada con el diseño, facilitando la edición para su mejoramiento o como punto de partida de un nuevo proyecto, para este proyecto se escogió el programa PROTEL99 como herramienta de diseño del PCB de las tarjetas de los módulos a implementarse.

2.5.2 Descripción de PROTEL99 para diseño de placas PCB

El programa PROTEL99 se trata de un sistema de diseño completo para el entorno Windows, que proporciona un conjunto de herramientas formadas por un gestor de documentos integrado, un capturador de esquemáticos, un simulador analógico basado en SPICE, un simulador digital basado en CUPL (lenguaje similar a ORCAD/PLD), una herramienta de diseño de placas de circuito impreso (PCB) y muchas otras funciones que nos facilitan el diseño del PCB.

En este proyecto de final de tópico únicamente se utilizó la herramienta de diseño de placas de circuito impreso, la parte más importante de esta herramienta de diseño es la librería de componentes que puede ser sencilla o compleja de acuerdo a las necesidades del diseñador, para obtener el PCB de las placas PROTEL99 hace uso de sus herramientas integradas como el capturador de esquemáticos con herramientas de diseño de PCB, el cual tiene enlaces que definen como un símbolo del esquemático se relaciona con un símbolo de PCB, esta herramienta realiza el procesamiento del esquemático generando un fichero de netlist en el que se encuentra toda la información de los componentes que serán usados para crear el PCB por medio del programa de ruteado. Después de crear el fichero netlist se debe definir el tamaño de los exteriores de la placa y los agujeros de montaje necesarios en el PCB, según los requerimientos del circuito eléctrico y electrónico, luego se procede a examinar el netlist, este proceso analiza los componentes utilizados de las librerías y los coloca uno encima de otros en el origen del dibujo posteriormente se procede a la correcta ubicación de los componentes, cuando se mueve un componente a una zona libre de la placa, se observa las líneas que lo unen a otros componentes, estas líneas sirven para encontrar la disposición más adecuada con respecto a los demás componentes. Finalizada la adecuada disposición de los componentes sobre el área de la placa se procede a realizar el ruteado automático donde será ubicado el cobre que conectara los diferentes componentes.

2.5.3 Cálculo del ancho de la pista de una placa de circuito impreso

El cálculo del ancho que debe tener una determinada pista de un circuito impreso esta basado en la aplicación del estándar general para la fabricación de circuitos impresos ANSI-ICP 2221 desarrollado por la IPC (Association connecting electronics industries).

Para el cálculo de una determinada pista se requiere conocer tres parámetros relacionados con la pista: la corriente máxima que soportara la pista será representada en amperios, el incremento máximo de temperatura arriba de la temperatura ambiente que soportara la pista expresado en grados Celsius, el grosor de la pista es decir la altura de la pista respecto a la base de la placa y será expresado en onzas por pie cuadrado. En las tarjetas desarrolladas en el proyecto se ha considerado 1 amperio como corrientes máximas permisibles que circularán sin dañar las pistas de control, 5 amperios en la pistas de fuerza para el caso de la tarjeta de entrenamiento con dsPIC, y un máximo de 6.5 amperios en las pistas de la tarjeta manejadora de motores DC, también se consideró en el diseño de todas las tarjetas que no se generará un incremento de temperatura mayor a 10 °C con respecto a la temperatura ambiente, por último el grosor de las pistas en las placas será de 1 onza/pie^2 ó 35 micras. En el anexo C, se encuentran las ecuaciones utilizadas para determinar los resultados mostrados en las siguientes tablas.

Tabla 2.5 Parámetros de las pistas de control

DATOS	Corriente	1	A
	Espesor del cobre	1	onza/pie^2
	Temperatura máxima	35	°C
	Temperatura ambiente	25	°C
ANCHO PISTA	Externa	0,2489	mm
	Interna	1,0126	mm

Tabla 2.6 Parámetros de las pistas de fuerza

DATOS	Corriente	5	6,5	A
	Espesor del cobre	1	1	onza/pie^2
	Temperatura máxima	35	35	°C
	Temperatura ambiente	25	25	°C
ANCHO PISTA	Externa	2,7182	4,01	mm
	Interna	9,0476	12,92	mm

2.5.4 Circuito impreso de las tarjetas

Los circuitos impresos obtenidos están diseñados a doble cara, es decir las pistas de cobre estarán distribuidas tanto en la parte de arriba como de abajo de la tarjeta, en las siguientes figuras se muestran los PCBs para la tarjeta básica para aplicaciones con dsPIC y de la tarjeta de entrenamiento con dsPIC30F4011.

Figura 2.17 Diagrama del PCB de la tarjeta básica para aplicaciones con dsPIC

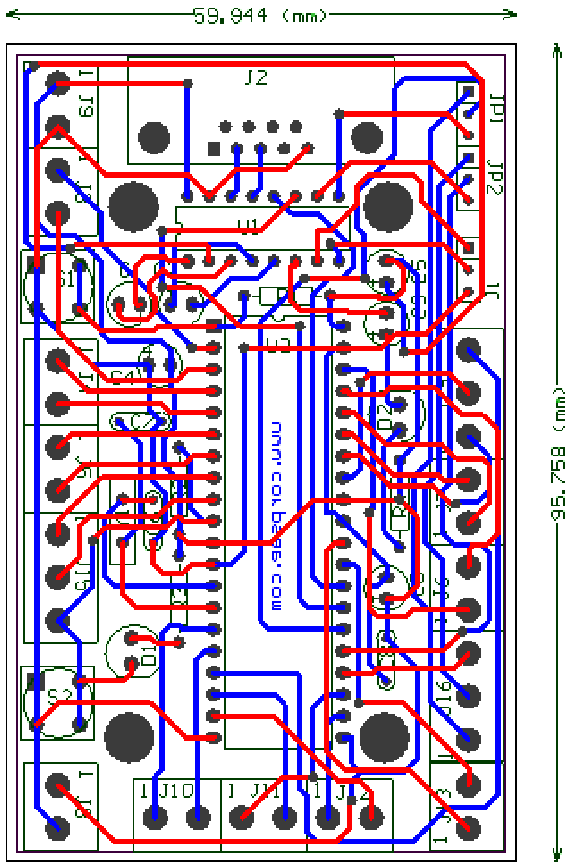


Figura 2.18 Diagrama del PCB de la tarjeta de entrenamiento con dsPIC

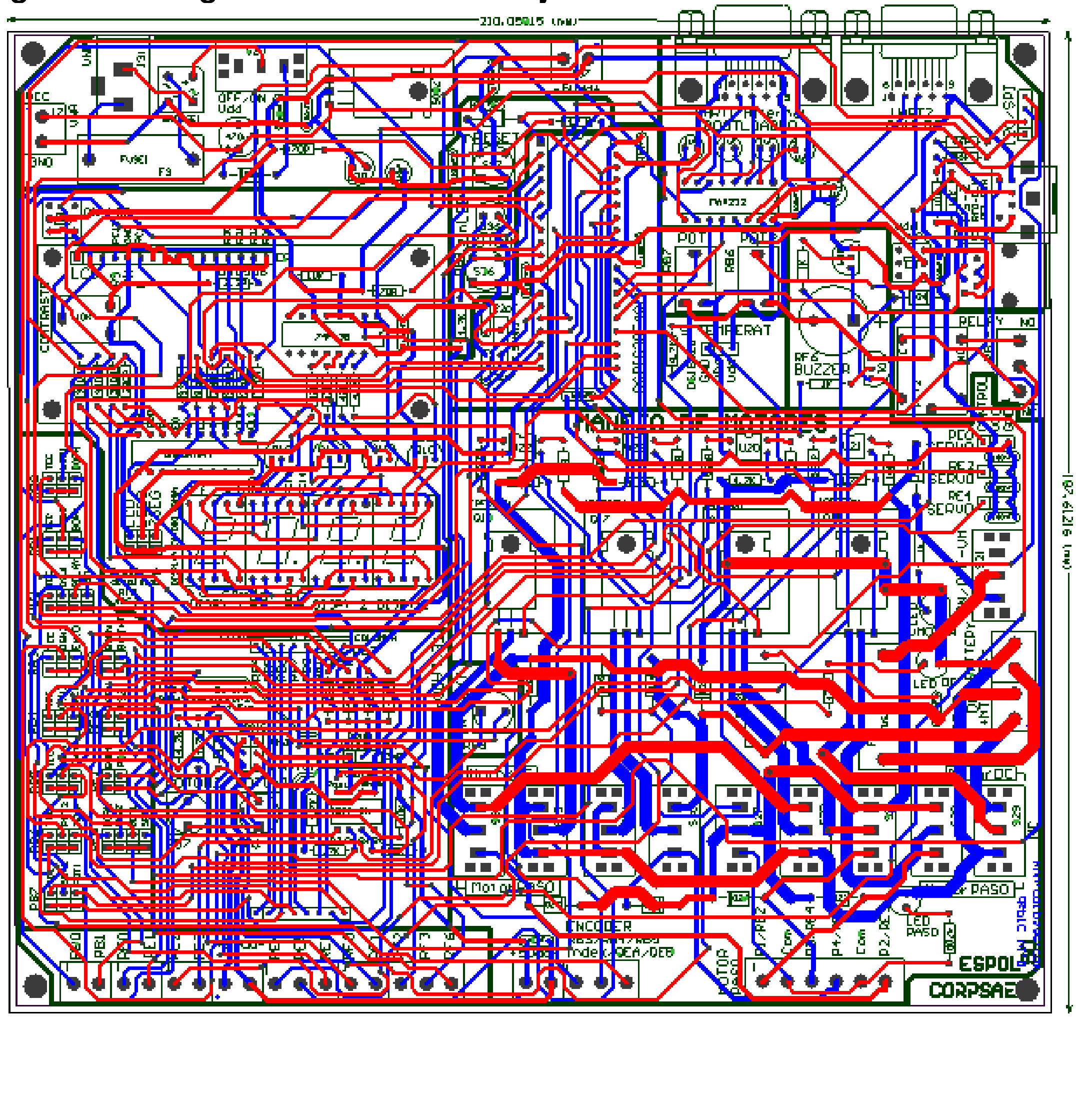
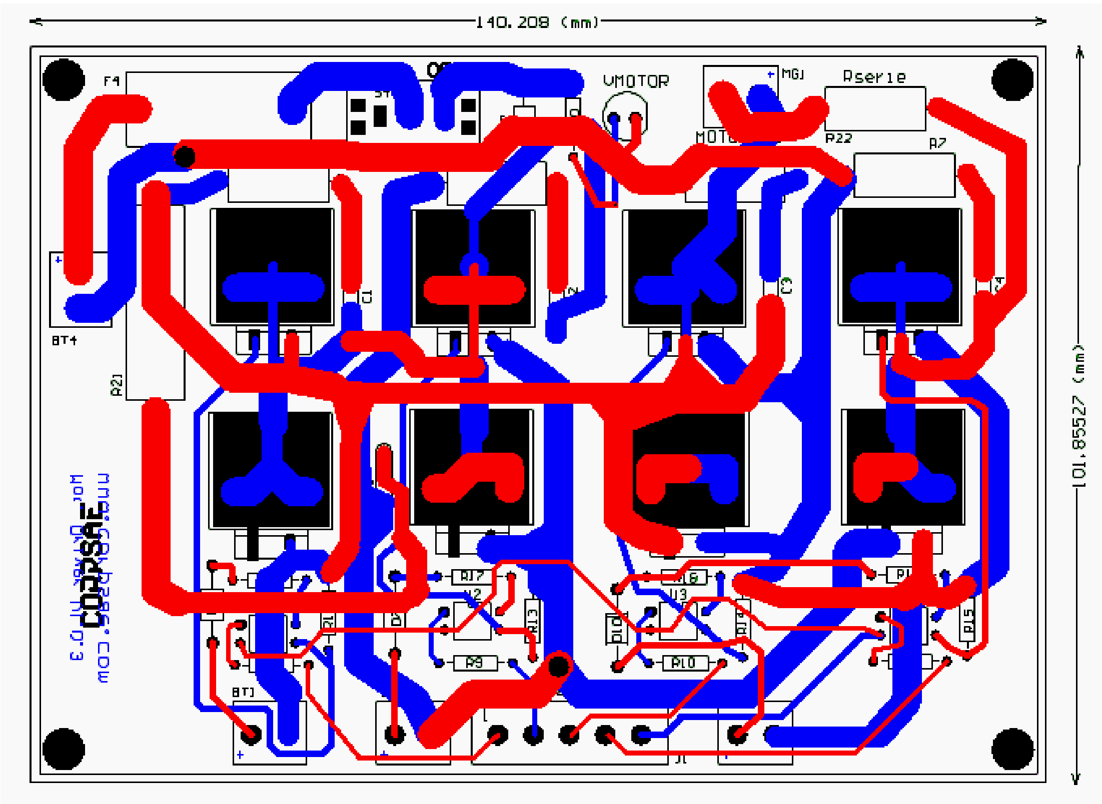


Figura 2.19 Diagrama del PCB del manejador de motores



CAPÍTULO 3

3. IMPLEMENTACIÓN DE LAS TARJETAS PARA EL ENTORNO DE LA PLATAFORMA DE DESARROLLO

3.1 Consideraciones generales

Como se mencionó en los capítulos anteriores el entorno del desarrollo de los programas para la tarjeta se realizará a través del compilador IDE de mikroBasic para dsPIC y su programación podrá ser realizada por medio del Pickit2 o con el bootloader. Respecto a la comunicación es a través del estándar RS-232 y su interfaz será implementada con la versión estudiantil de LabVIEW, para la ejecución es prioritario disponer de una plataforma de desarrollo que soporte los entornos antes mencionados, para suplir esta necesidad se escogió el sistema operativo Windows XP de Microsoft.

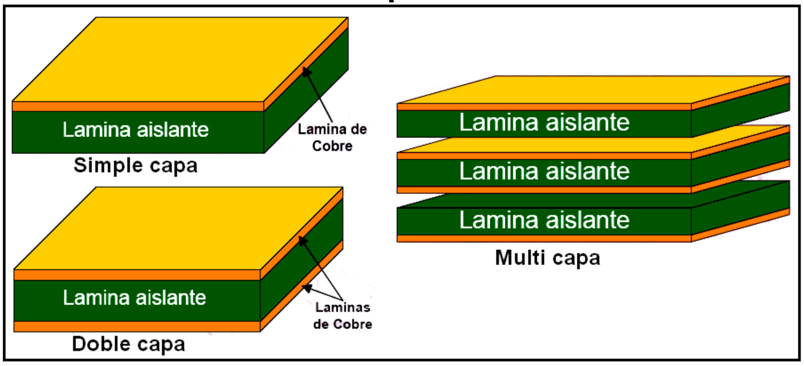
En cuanto a las consideraciones de construcción de las tarjetas, cada una quedó montada sobre una placa de fibra de vidrio de doble capa de cobre, agujeros metalizados, recubrimiento antisoldante, y serigrafiado sobre las tarjetas, en ambas se utilizó criterios de las normas UNE (europeas), y ANSI (Instituto nacional de estándar americano), esto asegura un correcto funcionamiento de las mismas con sus respectivas limitaciones mencionadas en los anteriores capítulos, a continuación se menciona los pasos más elementales del proceso de construcción para ambas tarjetas.

3.2 Detalles de construcción

3.2.1 Base aislante de la placa de circuito impreso

Los niveles de fabricación de los circuitos impresos depende de la cantidad de láminas de cobre que se depositen sobre un determinado número de bases aislantes, de esta forma se podrá disponer de placas de circuito impreso de simple, doble y múltiples capas (ver figura 3.1), en Ecuador el mercado de fabricación de placas de circuito impreso nos ofrece únicamente la opción de construcción a nivel de simple y doble capa.

Figura 3.1 Niveles de fabricación de placas de circuito impreso



***FUENTE**: Personal*

Las tarjetas de circuito impreso utilizadas en este proyecto final de tópico son de doble capa constituidas de una base aislante de fibra de vidrio, en ambas capas se coloca una fina lámina de cobre sobre la cual se hace el trazado de las pistas conductoras que realizan las conexiones entre los distintos componentes de la tarjeta, el espesor de la base aislante pueden ser de 0.2, 0.5, 0.7, 0.8, 1.0, 1.2, 1.5, 1.6, 2.0, 2.4, 3.2, ó 6.4 mm. (UNE 20-621-84/3), el espesor de la capa conductora puede ser de 1, 2, 3 onza/pie^2, es decir 35, 70, 105 micras. En la siguiente tabla se muestra las dimensiones, espesor de la base aislante y capa conductora del circuito impreso de las tarjetas implementadas.

Tabla 3.1 Dimensiones del PCB de las tarjetas

	ELEMENTO	DIMENSIÓN
Espesor	Base aislante del PCB	1,5 mm
	Capa conductora	1onz/pie^2
Tarjeta básica	Largo de la placa	95,76 mm
	Ancho de la placa	59,94 mm
Tarjeta de entrenamiento	Largo de la placa	197,61 mm
	Ancho de la placa	210,06 mm
Manejador de motores DC	Largo de la placa	101,86 mm
	Ancho de placa	140,21 mm

3.2.2 Diámetro y metalizado de los agujeros del PCB

El tamaño de los agujeros depende del componente que se vaya a insertar en la tarjeta, la tabla 3.2 muestra los diámetros nominal y mínimo para los agujeros tipo metalizados recomendados de acuerdo a la norma UNE 20-621-84/3.

Tabla 3.2 Diámetros para agujeros metalizados

DIÁMETRO NOMINAL (mm)	DIÁMETRO MÍNIMO (mm)
0,4	0,35
0,5	0,45
0,6	0,55
0,8	0,75
0,9	0,85
1,0	0,9
1,3	1,2
1,6	1,5
2,0	1,9

FUENTE: *Norma UNE 20-621-84/3*

Las tarjetas obedecen a un diseño a doble capa es decir que existirán pistas de conexiones en ambos lados de la tarjeta, siendo necesario que exista continuidad entre algunas pistas de la capa superior e inferior, por tal motivo se ha optado por la metalización de los agujeros de la tarjeta. Las etapas de este proceso consisten en depositar cobre por medio de electrólisis en las perforaciones con la finalidad de que estas adquieran conductividad, luego adquieren una segunda capa de estaño para hacerlas mas resistentes y proteger el cobre de alguna oxidación y brindar una mejor soldabilidad de los componentes.

Cada agujero metalizado de la tarjeta tiene asociado un pad de soldadura cuyo tamaño depende del diámetro del agujero metalizado para la inserción del componente y de las exigencias de la soldadura, en la tabla 3.3 se muestran algunos diámetros de agujeros y su respectivo pad recomendado por la norma UNE20-552-75 .

Tabla 3.3 Diámetros del PAD de soldadura del PCB

DIÁMETRO NOMINAL AGUJERO (mm)	DIÁMETRO MÍNIMO DEL PAD (mm)
0,6	1,8
0,8	2,3
1,0	2,5
1,3	2,8
1,6	3,1
2,0	3,5

FUENTE: *Norma UNE 20-555-75*

3.2.3 Protección con máscara antisoldante

La máscara antisoldante es una capa de laca horneable de alta resistencia mezclada con un tinte, la cual tiene la función de proteger las pistas de cortocircuitos, evitar la corrosión y el sulfatado de las pistas por efectos químicos o ambientales, y permite que al soldar la soldadura se extienda únicamente en el punto donde esta el agujero en el cual entra el pin del componente dando una forma redonda y uniforme a el punto de soldadura, evitando que la suelda se disperse por las pistas. La aplicación de la laca de alta resistencia es a través de una máscara de antisoldante que evita que los pads de soldadura queden cubiertos por el aislante obteniéndose una resolución de 0.3mm, quedando protegidas las caras superior e inferior de las placas del circuito impreso.

3.2.4 Técnica de serigrafía sobre las tarjetas

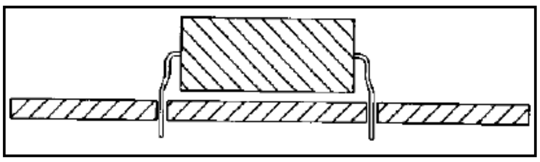
La serigrafía es una técnica que combina procedimientos fotográficos que junto a tintas especiales permite imprimir sobre el circuito impreso los símbolos, nombres e inscripciones, configuración de los interruptores referentes a los componentes de la tarjeta, esta impresión puede realizarse en la capa inferior o superior de la misma.

Las tarjetas de circuito impreso vienen con el serigrafiado sobre la capa superior donde se indican diferentes distintivos que serán de mucha utilidad al momento del ensamblaje, configuración y programación de los diferentes periféricos, entre los distintivos se encontrará nombres o capacidades de elementos electrónicos, puentes y señales provenientes de la unidad de control.

3.2.5 Montaje de los componentes sobre la tarjeta de circuito impreso

La forma de conectar los componentes con las capas conductoras es a través de agujeros metalizados y su respectivo pad de soldadura, este tipo de montaje de los componentes sobre la tarjeta se denomina THD (las siglas del inglés through hole device), es decir los pines de los componentes se acoplan a los agujeros metalizados para su posterior soldado con la placa de circuito impreso. (Ver figura 3.2)

Figura 3.2 Montaje THD sobre el circuito impreso



***FUENTE**: Personal*

En las tablas mostradas a continuación se detalla la referencia y la correspondiente descripción de cada componente usado en la tarjeta de desarrollo y aplicaciones con dsPIC30F4011.

Tabla 3.4 Resistencias y potenciómetros utilizados en la tarjeta de entrenamiento con dsPIC30F4011

REFERENCIA	DESCRIPCIÓN
R1	Resistencia 470R 1/4W, para led D1 indicador de energización de la fuente de control
R2	Resistencia 1K 1/4W, para led D4 indicador de energización de fuente de motores
R3	Resistencia 4,7K 1/4W, para botón de nReseteo
R5	Resistencia 470R 1/4W, para led D7
R6, R7	2 Resistencia 56R 1/4W, para protección de programación por ICD2
R8	Potenciómetro 10K 1/4W, para contraste de pantalla de cristal líquido
R9, R13,R14, R15, R16, R17, R18, R19, R20, R21	10 Resistencias 330R 1/4W (naranja-naranja-marón), para barra de leds
R10	Resistencia 2.2R 1/4W, para pantalla de cristal líquido
R11, R12, R22, R23	2 Resistencias 47R 1/4W (amarillo-violeta-negro),para Q1, Q2, Q3, y Q4 -2N3906
R24	Resistencia 10K 1/4W, para 74LS138
R25, R26	2 Resistencia 10K 1/4W, para la dos ultimas posiciones del banco de interruptores configurados en lógica negativa
R4, R27, R28, R29	4 Resistencia 4,7K 1/4W, para nBotón, Botón1, Botón2, Botón int-ext
R30, R31, R32, R33	4 Resistencia 4,7K 1/4W, para teclado hexadecimal
R34, R35	2 Resistencia 4,7K 1/4W, para la dos primeras posiciones del banco de interruptores configurados en lógica positiva
R36, R41	Potenciómetro tipo araña de 10K
R37, R42	Resistencia 1K 1/4W, para Q5 y Q6 -2N2222
R38, R39	2 Resistencia 10K 1/4W, para conector PS/2
R40	Resistencia 4,7K 1/4W, para DS18B20 pin2
R43, R48, R49, R50	4 Resistencia 560R 1/4W, para los PC817 U09, U10, U11, U12
R44	Resistencia 1K 1/4W, para LED D9 indicador de habilitación de bornera de conexión del motor DC
R45	Resistencia 1K 1/4W, para LED D10 indicador de habilitación de bornera de conexión del motor PASO
R46, R51, R52, R57	4 Resistencia 4,7K 1/4W, para los PC817 U09, U10, U11, U12
R47, R53, R54, R55	4 Resistencia 10R 1/4W, para PC817 U09, U10, U11, U12
R56	Resistencia 5R 2W

FUENTE: *Personal*

Tabla 3.5 Diodos utilizados en la tarjeta de entrenamiento con dsPIC30F4011

REFERENCIA	DESCRIPCIÓN
D1	Led indicador de energización de fuente de alimentación de control
D3	Diodo 1N4007 de protección para fuente de alimentación de control
D4	Led indicador de energización de fuente de alimentación de motores
D5	Diodo 1N4007 de protección para fuente de alimentación de motores
D6	Diodo schotty para protección del dsPIC30F4011 durante programación por ICD2
D7	Diodo LED en lógica negativa conectado a pin10 del dsPIC30F4011
D8	Diodo para protección de relé J18
D9	Led indicador EN-DC de energización del motor DC
D10	Led indicador EN-PASO de energización del motor de paso
D11	Diodo de paso libre para bobina A del motor de paso
D12	Diodo D12 para protección de Q7-IRF530
D13	Diodo D13 para protección de Q8-IRF530
D14	Diodo de paso libre para bobina C del motor de paso
D15	Diodo de paso libre para bobina B del motor de paso
D16	Diodo D16 para protección de Q9-IRF530
D17	Diodo D17 para protección de Q10-IRF530
D18	Diodo de paso libre para bobina D del motor de paso

FUENTE: *Personal*

Tabla 3.6 Capacitores utilizados en la tarjeta de entrenamiento con dsPIC30F4011

REFERENCIA	DESCRIPCIÓN
C1	Capacitor de desacoplo 100nF para el regulador LM7805
C2	Capacitor electrolítico 470uFpara el regulador LM7805
C3	Capacitor de desacoplo 100nF para el regulador LM7805
C4	Capacitor de desacoplo 100nF para el regulador LM7805
C5	Capacitor electrolítico de 220uF para el regulador LM7805
C6	Capacitor 100nF de estabilización del funcionamiento para el dsPIC30F4011
C7	Capacitor de desacoplo 100nF para el dsPIC30F4011
C8	Capacitor de desacoplo 100nF para el dsPIC30F4011
C9	Capacitor de desacoplo 100nF para el dsPIC30F4011
C10	Capacitor de desacoplo 100nF para el dsPIC30F4011
C11	Capacitor de 22pF para cristal de cuarzo
C12	Capacitor de 22pF para cristal de cuarzo
C13	Capacitor de desacoplo 100nF para el MAX232
C14	Capacitor electrolítico de 1uF para el MAX232
C15	Capacitor electrolítico de 1uF para el MAX232
C16	Capacitor electrolítico de 1uF para el MAX232
C17	Capacitor electrolítico de 1uF para el MAX232
C18	Capacitor electrolítico de 1uF para el MAX232
C19	Capacitor de desacoplo 100nF para bornera externa J6
C20	Capacitor de desacoplo 100nF para bornera externa J10
C21	Capacitor de desacoplo 100nF para 74LS138
C22	Capacitor de desacoplo 100nF para PS/2
C23	Capacitor de desacoplo 100nF para encoder
C24	Capacitor de desacoplo 100nF para servomotor 1
C25	Capacitor de desacoplo 100nF para servomotor 2
C26	Capacitor de desacoplo 100nF para servomotor 3
C27	Capacitor de ?? RED SNUBBER

FUENTE: *Personal*

Tabla 3.7 Conectores, y borneras utilizados en la tarjeta de entrenamiento con dsPIC30F4011

REFERENCIA	DESCRIPCIÓN
J0	Bornera de salida de voltaje no regulado 8-12 VDC
J1	Adaptador de 8-12 V AC/DC
J2	Conector DB9 hembra para COM PC
J3	Conector DB9 hembra para bootloader
J4	Conector RJ12 para ICD2
J5	Conector polarizado para I2C/SPI
J6	Bornera de salida de voltaje regulado 5 VDC
J7	Bornera de conexión externa I/O para RB0 y RB1
J8	Bornera de conexión externa I/O para RE0 y RE1
J9	Bornera de conexión externa I/O para RF0 y RF1
J10	Bornera de salida de voltaje regulado 5 VDC
J11	Bornera de conexión externa I/O para RE2 y RE3
J12	Bornera de conexión externa I/O para RE4, RE5, y RE8
J13	Bornera de conexión externa I/O para RF2, RF3, y RF6
J18	Bornera de salida por relé
J19	Conector PS/2 para teclado de PC
J20	Bornera para conexión del encoder Señal Z, Fase A, y Fase B
J21	Bornera de salida de voltaje regulado 5 VDC
J22	Conector polarizado para servomotor 1
J23	Conector polarizado para servomotor 2
J24	Conector polarizado para servomotor 3
U6	Conector simple tipo cabecera hembra x8 para teclado hexadecimal
U8	Conector simple hembra maquinado x 3 para sensor de temperatura DS18B20
LCD1	Conector simple tipo cabecera hembra x18 para LCD
Y1	Zócalo simple hembra maquinado x 3 para cristal de cuarzo
M1	Bornera para conexión del motor de paso
MG1	Bornera para conexión del motor DC
BT1	Bornera de alimentación para motores 6-24 VDC
JP1, JP2, JP3, JP4, JP5, JP6, JP7, JP8, JP9, JP10 JP11, JP12, JP13, JP14, JP15, JP16, JP17, JP18, JP21, JP22, JP23, JP24, JP25, JP26, JP28, JP29, JP30, JP31, JP32, JP33, JP34, JP35, JP36, JP37, JP38, JP39, JP40, JP41	38 Conector simple macho x 2 pines
JP19, JP20, JP27,	3 Conector simple macho x 2 pines
JP1 a JP41	16 puentes*

* Para saber la descripción del respectivo puente refiérase a conectores y señales

FUENTE: *Personal*

Tabla 3.8 Integrados, zócalos y otros dispositivos utilizados en la tarjeta de entrenamiento con dsPIC30F4011

REFERENCIA	DESCRIPCIÓN
J14, J15, J16, J17	Zócalo maquinado de 20 pines para los 4 visualizadores 7 de segmentos
	4 visualizadores de 7 segmentos cátodo común
U2	Zócalo de 40 pines para dsPIC30F4011
	dsPIC30F4011
U3	Zócalo de 16 pines para MAX232
	MAX232
U4	Zócalo de 16 pines para decoder 74LS138
	74LS138
F1	Porta fusible para la fuente de control
	Fusible de 1A
F2	Porta fusible para la fuente de motores
	Fusible de 5A
Y1	Cristal de cuarzo de 7,3728MHz conectado entre pines 13 y 14 del daPIC30F4011
U8	Sensor de temperatura DS18B20
LCD1	Pantalla de cristal líquido 16x2
U5	Zócalo de 20 pines para barra de diodos emisores de luz (leds)
	Barra de diodos emisores de luz x 10
U6	Teclado matricial
U7	Bocina 5 VDC
RLY1	Relé simple de 5 VDC
F1	Fusibles de 1amperio
F2	Fusibles de 5 amperios
Q1	Transistor bipolar PNP 3906 para habilitar visualizador1
Q2	Transistor bipolar PNP 3906 para habilitar visualizador2
Q3	Transistor bipolar PNP 3906 para habilitar visualizador3
Q4	Transistor bipolar PNP 3906 para habilitar visualizador4
Q5	Transistor bipolar NPN 2N2222 para bocina
Q6	Transistor bipolar NPN 2N2222 para relé simple
Q7	Transistor mosfet IRF530 para manejo de motor DC y de paso
Q8	Transistor mosfet IRF530 para manejo de motor DC y de paso
Q9	Transistor mosfet IRF530 para manejo de motor DC y de paso
Q10	Transistor mosfet IRF530 para manejo de motor DC y de paso
U9	Optoacoplador de 1 canal PC817 para manejo de motor DC y de paso
U10	Optoacoplador de 1 canal PC817 para manejo de motor DC y de paso
U11	Optoacoplador de 1 canal PC817 para manejo de motor DC y de paso
U12	Optoacoplador de 1 canal PC817 para manejo de motor DC y de paso
D2	Puente rectificador 1 amperio para fuente de control
U1	Regulador de voltaje a 5 VDC para fuente de control

FUENTE: *Personal*

Tabla 3.9 Interruptores, botones y teclados utilizados en la tarjeta de entrenamiento con dsPIC30F4011

REFERENCIA	DESCRIPCIÓN
S1	Interruptor de 2 posiciones para ON/OFF de fuente de alimentación de control
S2	Interruptor de 2 posiciones para energización de los motores
S7	Banco de Interruptores x 4
S11	Interruptor de 3 posiciones para habilitación de motor DC/NC/PASO
S12	Interruptor de 3 posiciones para habilitación de motor DC/NC/PASO
S13	Interruptor de 3 posiciones para habilitación de motor DC/NC/PASO
S14	Interruptor de 3 posiciones para habilitación de motor DC/NC/PASO
S15	Interruptor de 3 posiciones para habilitación de motor DC/NC/PASO
S16	Interruptor de 3 posiciones para habilitación de motor DC/NC/PASO
S17	Interruptor de 3 posiciones para habilitación de motor DC/NC/PASO
S18	Interruptor de 3 posiciones para habilitación de motor DC/NC/PASO
S19	Interruptor de 3 posiciones para habilitación de motor DC/NC/PASO
S3	Pulsador para nReset
S4	Pulsador con lógica negativa
S5	Pulsador/Interruptor para alimentación del dsPIC durante la programación usando la fuente de control o del Pickit (USB)
S6	Pulsador/Interruptor para energizar pantalla de cristal líquido
S8	Pulsador con lógica negativa o positiva
S9	Pulsador con lógica negativa o positiva
S10	Pulsador con lógica negativa o positiva

FUENTE: *Personal*

3.3 Señales y conectores

Para empezar a utilizar la tarjeta de entrenamiento con dsPIC30F4011 es conveniente colocar todos los interruptores y puentes de acuerdo con la aplicación requerida, las siguientes tablas muestran los ajustes previos para utilizar las diferentes aplicaciones implementadas en la tarjeta.

La alimentación para la fuente de control, se realiza a través de un conector cuyo voltaje de entrada es 8-12 V AC/DC, luego se debe colocar en encendido el interruptor para la energización obteniéndose finalmente un voltaje regulado de 4.5-5.5 VDC y un suministro de hasta máximo 1 amperio de corriente que energizarán a la unidad de control de la tarjeta (ver tabla 3.10).

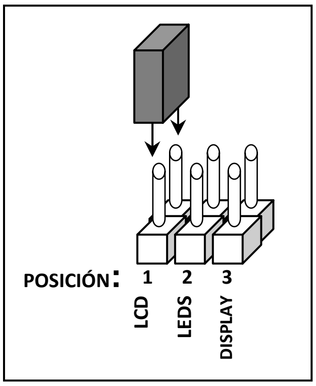
Tabla 3.10 Alimentación de la fuente de control

CONECTOR	ENTRADA	
Conector de alimentación para fuente de control ⊖—⊖—⊕	8-12 V AC/DC	
CONECTOR/INTERRUPTOR	**SALIDA**	
Interruptor de energización	APAGADO	ENCENDIDO
Bornera de VNR	0 VDC	8-12 VDC
Bornera de VR	0 VDC	4,5 - 5,5 VDC

FUENTE: *Personal*

Los periféricos de visualización quedarán habilitados de acuerdo a la ubicación del puente de este módulo, para la posición 1 se habilita la pantalla de cristal líquido, en la posición 2 se habilitan los ocho primeros leds de la barra, y en la posición 3 quedan habilitados los visualizadores de siete segmentos, es decir solo quedara conectado en una de las tres posiciones habilitando su respectivo periférico como se ilustra la en figura 3.3, de igual forma en la tabla 3.11 se indica las posiciones en que se ubica el puente.

Figura 3.3 Esquema de conexión del puente para periféricos de visualización



FUENTE: *Personal*

Tabla 3.11 Posición del puente para periféricos de visualización

PERIFÉRICO	POSICIÓN DEL PUENTE 1-2		
	1	2	3
Pantalla de cristal líquido	Habilitado	Deshabilitado	Deshabilitado
Barra de leds x 8	Deshabilitado	Habilitado	Deshabilitado
Visualizador 7SEG	Deshabilitado	Deshabilitado	Habilitado

FUENTE: *Personal*

Cuando se encuentre habilitado la pantalla de cristal líquido adicionalmente está disponible un interruptor para el encendido o apagado y un potenciómetro para variar el contraste de la pantalla.

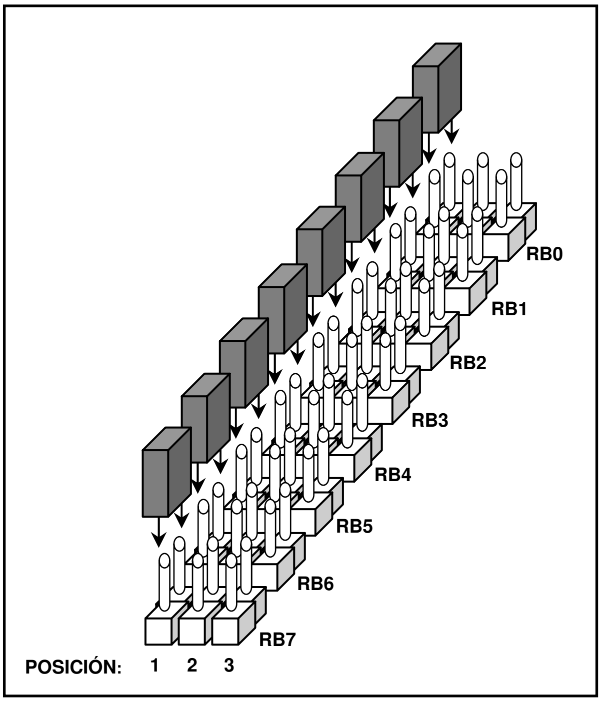
En cuanto a los periféricos que trabajan con el puerto B, se utilizan ocho puentes para habilitar las diferentes combinaciones de los dispositivos que trabajan con ese puerto del dsPIC30F4011 según su posición (tabla 3.12), para la ***posición 1*** todo el puerto B manejará un teclado matricial (ver figura 3.4), mientras que en la ***posición 2*** el puerto B maneja a cuatro periféricos, leds desde RB0 a RB1, banco de interruptores desde RB2 a RB5, potenciómetro para RB6, pulsador para RB7, finalmente si los puente se ubican en la ***posición 3*** se habilitará las señales RB0 y RB1 en borneras externas, el relé simple para RB2, las señales del encoder para RB3,RB4,RB5, el sensor de temperatura para RB6, potenciómetro para RB7. También es posible realizar combinaciones de las ubicaciones de los puentes para el manejo de los periféricos siempre que esto sea factible para su respectivo funcionamiento.

Tabla 3.12 Posiciones del puente para habilitación del periférico en el puerto RB

Pin	Puerto	Posición de los Puentes		
		1	2	3
2	RB0	TECLADO MAT	Diodo led 1	Bornera externa RB0
3	RB1	TECLADO MAT	Diodo led 2	Bornera externa RB1
4	RB2	TECLADO MAT	BancoInterrup1	Relé
5	RB3	TECLADO MAT	BancoInterrup2	Encoder Señal Z
6	RB4	TECLADO MAT	BancoInterrup3	Encoder Fase A
7	RB5	TECLADO MAT	BancoInterrup4	Encoder Fase B
8	RB6	TECLADO MAT	Potenciómetro 2	Sensor Temperatura
9	RB7	TECLADO MAT	Botón pulsador 2	Potenciómetro 1

FUENTE: *Personal*

Figura 3.4 Esquema de conexión de los puentes para periféricos del puerto RB

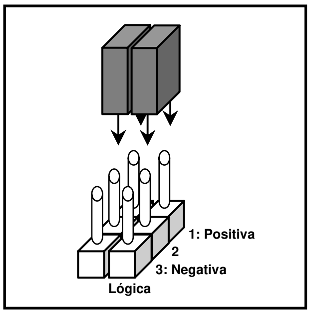


FUENTE: *Personal*

Las resistencias del teclado matricial son manejadas por un puente de dos posiciones que las dejará conectadas a Vcc o Gnd.

La tarjeta de entrenamiento también dispone de tres pulsadores botón uno, dos, e interrupción externa, configurables en lógica positiva o negativa mediante dos puentes de dos posiciones (figura 3.5), para la configuración de una lógica se debe ubicar ambos puentes en la posición 1-2 ó 2-3 como se muestra en la tabla 3.13

Figura 3.5 Esquema de conexión de los puentes para pulsadores con lógica configurable



FUENTE: *Personal*

Tabla 3.13 Posición de los puentes para pulsadores con lógica configurable

PERIFÉRICO	Puente -3	
	Posición 1-2	**Posición 2-3**
Teclado matricial	Resistencia pull up a Vcc	Resistencia pull up a GND
Botón 1	Lógica positiva*	Lógica negativa*
Botón 2		
Botón INT EXT		

*:Se configuran dos puente al mismo tiempo

FUENTE: *Personal*

Adicionalmente se encuentran implementados dos pulsadores *nBoton* y *nReseteo*, ambos con lógica negativa, el primero comparte el puerto RB8 del dsPIC con un diodo emisor de luz, obteniéndose dos periféricos adicionales, mientras que el segundo pulsador sirve como señal de reseteo para el dspic30F4011.

La tabla 3.14 muestra donde debe colocarse los puentes para la habilitar los periféricos conectados a RE8, RF0, RF1, y RF6 del dsPIC30F4011, habrá un puente en cada puerto habilitando únicamente a un periférico, por ejemplo para tener las cuatro borneras externas se debe colocar los puentes en la posición1, pero para habilitar el resto de periféricos se debe de reubicar el puente(s), en la posición del periférico requerido.

Tabla 3.14 Posiciones del puente para habilitación del periférico en el puerto RE8,RF0,RF1, y F6

Pin	Puerto	Posición del puente-2			
		1	2	3	4
17	RE8	Born. Exter. RE8	Botón Interrup.	--	--
30	RF0	Born. Exter. RF0	PS/2 - Reloj	--	--
29	RF1	Born. Exter. RF1	PS/2 - Dato	--	--
24	RF6	Born. Exter. RF6	Bocina	Botón puls. 1	SPI – SCK

-- : No disponible
FUENTE: *Personal*

Para el manejo de los motores de paso y de corriente continua es necesario conectar una fuente de alimentación externa en la bornera destinada para este propósito, luego se habilita el suministro de energía mediante el interruptor de alimentación de dos posiciones encendido/apagado, el encendido de un led indicará cuando el interruptor está habilitado y el suministro de energía es correcto, posteriormente se debe habilitar la configuración del motor que se utilizará, para esto se cuenta con un banco interruptores de tres posiciones, seleccionando entre las configuraciones para el manejo de motor dc, ningún motor conectado, o para manejar un motor de paso (tabla 3.15), la conexión del motor seleccionado se realizará a través de borneras externas implementadas para estos motores.

Tabla 3.15 Configuración para el manejo de los motores DC y paso

Habilitador	Entradas		Salidas
	Posición	Acción	
Bornera de conexión de fuente de motores 6-24V (Batería +DC-)	1	Borne DC(+)	--
	2	Borne DC(-)	--
Interruptor de alimentación (encendido/apagado – VM)	1	encendido de motores	Led encendido (VM)
	2	apagado de motores	Led apagado (VM)
Banco de Interruptores de fuerza S11 a S19	1	Motor DC	Led encendido-DC
	2	NC	Led apagado
	3	Motor Paso	Led encendido-Paso

-- : No disponible

FUENTE: *Personal*

En la tabla 3.16 se indica la señal correspondiente a cada punto de conexión de las borneras implementadas para conectar un encoder, motor DC, motor de paso, y las borneras destinadas para la conexión de hasta tres servomotores.

Tabla 3.16 Conectores para encoder, motores dc, paso, y servo motores

CONECTOR		MOTOR	
		Cable	Alimentación
Encoder J20, J21	1	Vcc	5VDC
	2	Gnd	
	3	Señal Z	
	4	Fase A	
	5	Fase B	
Bornera del motor DC MG1 (MOTOR DC +MT-)	1	M(+)	6-24 VDC
	2	M(-)	
Bornera del motor de paso M1 (MOTOR-PASO)	1	Fase 1	6-24 VDC
	2	Común A	
	3	Fase 3	
	4	Fase 4	
	5	Común B	
	6	Fase 2	
Conectores de servo motores J22, J23, J24 (SERVO x 3)	1	Control	5VDC
	2	Vcc	
	3	Gnd	

FUENTE: *Personal*

En la tarjeta de entrenamiento existen otros conectores adicionales que tienen vinculados su respectivo periférico y/o función, por ejemplo existen dos conectores DB9 hembra de los cuales uno esta destinado para la carga del programa de bootloader al dsPIC, y el otro conector servirá como puerto de comunicación entre el computador y el microcontrolador, también existen conectores para manejo de otros periféricos como teclados tipo PS/2, dispositivos para comunicación SPI, entre otros como se muestra en la tabla 3.17

Tabla 3.17 Conexión de elementos adicionales

CONECTOR	FUNCIÓN
DB9 hembra	Bootloader
DB9 hembra	COM PC
Conector Polarizado x4	I2C / SPI
RJ12	ICD2
PS/2	Teclado PC
Conector hembra x3	Sensor de Temperatura
Bornera	Relé
Sin conector	Bocina

FUENTE: *Personal*

3.4 Interfases de comunicación disponible

Se dispone de diferentes interfases de comunicación siendo estos protocolos reglas fundamentales para permitir el flujo de información entre distintos periféricos conectados a las tarjetas y el computador para el intercambio de datos. Los protocolos de la tarjeta básica para aplicaciones con dsPIC30F4011, pueden ser implementados de manera externa a través de las borneras disponibles para este propósito, en cuanto a la tarjeta de entrenamiento con dsPIC30F4011 los protocolos disponibles se describirán a continuación:

Comunicación serial RS-232, es importante que una tarjeta de desarrollo disponga de puertos de comunicaciones y transmisión de datos. Se ha incorporado una interfaz para realizar la comunicación serial mediante el protocolo RS-232, para lo cual se ha hecho uso del UART2 del dsPIC, teniendo como elemento final un conector DB9 hembra pudiendo ser conectado a la mayoría de ordenadores personales.

La tarjetas de entrenamiento dispone de dos tipos de comunicación SSP (puerto serial sincrónico), muy útil para comunicación con otros microcontroladores, dispositivos o periféricos, por ejemplo: EEPROM serie (almacenamiento de datos no volátiles), registros de desplazamiento (expansión de entradas y/o salidas), manejadores de visualizadores (reducción de conexiones), conversores A/D (digitalización externa de señales), entre otros.

Los modos de operación posibles del módulo SSP son: SPI (interfase serial de periféricos) interfase de periféricos seriales, e I2C (entre circuitos integrados).

La comunicación SPI permite la transmisión y recepción sincrónica simultánea de datos seriales entre periféricos, debiendo existir un dispositivo maestro (genera la señal de reloj) y uno ó varios esclavos (reciben la señal de reloj) mediante la conexión de 3 pines asociados al puerto SPI, descritos a continuación: salida serial de datos (SDO-dato serial de salida), entrada serial de datos (SDI- dato serial de entrada), y el reloj de comunicación serial (SCK-reloj serial), de manera adicional hay una cuarta señal que se puede utilizar cuando el microcontrolador se configura como dispositivo esclavo (SS-selección de esclavo).

El protocolo de bus I2C permite la transmisión de datos entre varios circuitos integrados, usando tan solo dos hilos, descritos a continuación: reloj serial sincrónico (SCL-synchronous serial clock) y la señal de datos seriales (SDA-synchronous serial data).

La simplicidad y facilidad de uso son sólo algunas de las razones para incluirlo en sistemas empotrados. La mayoría de microcontroladores modernos, disponen de interfases I2C y SPI.

3.5 Análisis de costos

El siguiente análisis estima el costo económico del proyecto, para lo cual se han dividido los gastos en varias etapas mostradas a continuación: recursos por hardware, por software, por componentes de fabricación del circuito impreso, y por recursos humanos. El total de los costos es la suma de todos los costos anteriores mas el costo del producto una vez puesto en el mercado.

3.5.1 Costos de recursos por hardware

Son los recursos necesarios para la concepción final del proyecto pero estos no forman parte de él. Entre los recursos de hardware se encuentra un ordenador personal con las siguientes características mostradas en la tabla 3.18

Tabla 3.18 Características del ordenador personal

DESCRIPCIÓN
Procesador Core 2 Duo de 3GHz
Memoria RAM de 2 Gb
Disco duro de 250 Gb
Monitor panorámico de 20" LCD
Modem para Internet por cable

FUENTE: *Personal*

Adicionalmente se necesita como hardware un kit de soldadura para el montaje de componentes sobre el circuito impreso, multímetro digital, un programador para dsPIC, entre otros, en la tabla 3.19 se detalla los costos de estos recursos.

El ordenador personal tiene un valor de 1000 dólares, pero será amortizado a un periodo de 4 años, entonces el valor del computador antes de la amortización sería de 20.83 \$/mes, para el análisis de la amortización se ha considerado 8 horas diarias de trabajo y suponiendo que el mes cuenta con 20 días laborables se obtiene 160 h/mes que corresponden a las horas de uso del computador durante un mes, respecto al tiempo demandado por proyecto por parte del ingeniero es de 450 horas de trabajo, finalmente se tendrá el siguiente gasto mensual amortizado del computador de 58.59 dólares (ver tabla 3.20).

$$\text{Amortización del computador} = \frac{20.83\ \$/\text{mes}}{160\ \text{h/mes}} * 450\ \text{h} = \ \$\,58.59$$

Tabla 3.19 Costo por recursos de hardware

DESCRIPCIÓN	CANT	VALOR
PC personal Core 2 Duo de 3GHz	1	\$ 58,59*
PicKit2 - Grabador de dsPIC	1	\$ 60,00
Kit Soldador de precisión Back & Decker	1	\$ 35,00
Multímetro digital	1	\$ 20,00
Pinza extracción de integrados	1	2,80
Tableros de pruebas medianos	3	\$ 25,00
TOTAL		\$201,39

FUENTE: *Personal*

Tabla 3.20 Costos de amortización del computador

DESCRIPCIÓN	VALOR	UNIDAD
PC personal Core 2 Duo de 3GHz	1000,00	dólares
Amortización del PC en 4 años	48	meses
Costo por mes del PC	20,83	$/mes
Horas de uso del computador	160	h/mes
Horas de trabajo del ingeniero por proyecto	450	h
Amortización del computador	58,59	dólares

FUENTE: *Personal*

3.5.2 Costos de recursos por software

Los recursos por software usados para el proyecto están conformados por el sistema operativo Windows XP Profesional SP3, el programa de diseño de placas de circuitos impresos Altium Designer 6.0, el software mikroBasic para dsPIC 30/33, para el desarrollo de aplicaciones con dsPIC, el programador USB Pickit2 v2.50 (ver tabla 3.21).

Tabla 3.21 Costo por recursos de software

DESCRIPCIÓN	CANT	PVT
Microsoft Windows XP Profesional	1	$ 0,00
PicKit2 v2.50 – Software(incluido en Kit)	1	$ 0,00
mikroBasic para dsPIC 30/33	1	$ 149,00
Altium Designer 6.0	1	$ 187,50
TOTAL		$ 336,50

FUENTE: *Personal*

En cuanto a la herramienta de diseño de placas de circuito impreso tiene un valor de 8000 dólares, pero al igual que el computador será amortizado en un periodo de 4 años, entonces el valor del software antes de la amortización sería de 166.67 $/mes, para el análisis de la amortización se ha considerado 160 horas laborables en un mes, respecto al tiempo demandado por diseño por parte del ingeniero es de 180 horas de trabajo, finalmente se tendrá el siguiente gasto mensual amortizado del computador de 187.50 dólares (ver tabla 3.22).

$$\text{Amortización de herramienta de diseño} = \frac{166.67\ \$/\text{mes}}{160\ \text{h/mes}} * 180\ \text{h} = \ \$187.50$$

Tabla 3.22 Costos de amortización de Altium designer

DESCRIPCIÓN	VALOR	UNIDAD
Altium Designer 6.0	1000.00	dólares
Amortización del Altium Designer en 4 años	48	meses
Costo por mes del Altium Designer	166.67	$/mes
Horas de uso del PC	160	h/mes
Horas de trabajo del ingeniero por diseño	180	h
Amortización del Altium Designer 6.0	187.50	dólares

FUENTE: *Personal*

3.5.3 Costos por recursos humanos

Los costos por recursos humanos corresponden a los sueldos hipotéticos correspondientes al ingeniero a cargo de investigaciones para diseño, experimentación, e implementación de nuevos proyectos este sería de 550 $/mes; mientras que al técnico encargado del montaje, comprobación y prueba de las tarjetas sería de 250 $/mes, para este análisis se considera que un mes tiene 160 horas hábiles entonces se obtiene los salarios por hora, para mayor detalle refiérase a las siguientes tablas.

Tabla 3.23 Salarios por hora del personal

DESCRIPCIÓN	CANT
Ingeniero	3,44 $/h
Técnico	1,56 $/h

FUENTE: *Personal*

Tabla 3.24 Costo por tarea del ingeniero

TAREA	HORAS	COSTO
Análisis técnico	40	137,50
Diseño	100	343,75
Implementación	150	515,63
Experimentación	80	275,00
TOTAL	370	1271,88

FUENTE: *Personal*

Tabla 3.25 Costo por tarea del técnico

TAREA	HORAS	COSTO
Montaje	12	18,75
Comprobación y prueba de la tarjeta	5	7,81
TOTAL	17	26,56

FUENTE: *Personal*

Tabla 3.26 Costo por recursos humanos

TAREA	HORAS	COSTO
Ingeniero	1	1271,88
Técnicos	1	26,56
TOTAL	2	1298,44

FUENTE: *Personal*

3.5.4 Costo total del proyecto

En la siguiente tabla se muestra los valores de los gastos correspondientes a cada etapa de realización del proyecto como se mencionó anteriormente.

Tabla 3.27 Costo total del proyecto

TAREA	COSTO
Costo de recurso por hardware	210,51
Costo de recurso por software	253,17
Costo de componentes y fabricación para la tarjeta principal	289,19
Costo de componentes y fabricación para la tarjeta secundaria	37,79
Costo de recursos humanos	1298,44
TOTAL	2089,09

FUENTE: *Personal*

El presupuesto total para el desarrollo de los prototipos asciende a la cantidad de dos mil ochenta y nueve 09/100 dólares.

3.5.5 Estudio del costo de la tarjeta de entrenamiento con dsPIC30F4011 en el mercado.

Es muy importante conocer un valor referencial para la comercialización del kit de entrenamiento, dado este motivo es importante identificar los diferentes tipos de gastos generados para la producción del producto, para determinar el costo de cada unidad se deben considerar los siguientes parámetros: costos fijos (CF), costos variables (CV), costos totales (CT), ingresos totales (IT), utilidad (U), y el punto de equilibrio.

Se debe de tener en cuenta que los costos fijos son independientes del número de unidades vendidas como los gastos por recursos de software y hardware, mientras que aquellos gastos que infieren de acuerdo a la producción demandada se llaman costos variables por ejemplo gastos en materiales, y costos por recursos humanos formarían parte de este tipo de costo. El costo total es la suma de los costos fijos con los costos variables.

$$CT = CF + CV = CF + CV_{UNIDAD} * N^{o}\ UNIDADES$$

Los ingresos totales son obtenidos del número de unidades vendidas.

$$IT = PU * N^{\circ} UNIDADES - CT$$

El punto de equilibrio se define como el nivel de ventas que cubre todos los costos de operación fijo y variables, es decir el nivel en el cual las utilidad es cero, la ecuación para la utilidad antes de impuestos e intereses se describe a continuación.

$$UAII = PU * N^{\circ} UNIDADES - CF - CV_{UNIDAD} * N^{\circ} UNIDADES$$

Igualando la anterior ecuación a cero se obtiene la siguiente ecuación para el cálculo el punto de equilibrio:

$$N^{\circ} UNIDADES = \frac{CF}{PU - CV}$$

Los costos de los materiales para la tarjeta de entrenamiento disminuyen de acuerdo a la cantidad de elementos comprados, como el dsPIC, sensor de temperatura, transistores, entre otros componentes de la tarjeta, entre los materiales no esta incluido la pantalla de cristal líquido, motores, y encoder.

Tabla 3.28 Costo de materiales para implementación de la tarjeta de entrenamiento según las unidades producidas

UNIDADES PROD.	COSTO DE MATERIAL
1-5	173,44
6-25	170,18
26-50	123,12
51-100	110,67
101-500	106,42
501-1000	101,16
1001-5000	97,28
5001-10000	91,37
10000+	88,36

FUENTE: *http://www.electroavil-es.com/*

Basándose en los datos de la tabla anterior se aproximará un valor referencial de 250 dólares para la producción de 1 a 25 unidades, pero si aumenta la demanda de la tarjeta de entrenamiento implicaría que el valor de venta disminuya a una cantidad que cubra los costos totales de producción y obtener una utilidad considerable para el inversionista. Se debe considerar que realizando una comparación económica entre las tarjetas de entrenamiento disponibles en el exterior estas tienen un valor aproximado incluidos costos de transporte e impuestos de 230 dólares (precio en Ecuador), en cuanto al aspecto técnico la tarjeta de entrenamiento con dsPIC30F4011 dispone de una mayor variedad de funcionalidades en relación con las tarjetas ofertadas en el mercado exterior.

La tabla 3.28 muestra los costos en materiales de acuerdo a la cantidad de unidades producidas, el costo de cada unidad disminuye a medida que la producción aumenta con el objetivo de incentivar su compra y obtener más beneficios. En cuanto al punto de equilibrio este es la cantidad mínima que se debe de producirse para vender a un determinado precio y cubrir los costos totales de producción es decir cuando la utilidad es igual a cero.

Por citar el caso donde la producción asciende a 1001 unidades los costos referentes a esta producción permiten que el precio de venta al público sea de 150 dólares, siendo su punto de equilibrio de 40 unidades producidas y vendidas, obteniendo un ingreso aproximado de 150 mil dólares cuando toda la producción halla sido vendida, con una utilidad de 51 mil dólares. Siendo el valor de venta a este nivel de producción un precio competitivo en el mercado interno de tarjetas de entrenamiento (ver tabla 3.29).

Tabla 3.29 Costo por el número de unidades producidas

UNIDADES	CF	CM	CV	CT	PRECIO UNITARIO	INGRESO TOTAL	UAII	PUNTO DE EQUILIBRIO Unidades
1-5	1762,11	173,44	173,44	1935,55	250,00	250,00	-1685,55	23,02
6-25	1762,11	170,18	1021,08	2783,20	250,00	1500,00	-1283,20	22,08
26-50	1762,11	123,12	3201,02	4963,13	220,00	5720,00	756,87	18,19
51-100	1762,11	110,67	5644,00	7406,12	180,00	9180,00	1773,88	25,42
101-200	1762,11	106,42	10748,26	12510,37	180,00	18180,00	5669,63	23,95
201-500	1762,11	106,42	21390,10	23152,22	175,00	35175,00	12022,78	25,69
501-1.000	1762,11	101,16	50679,71	52441,82	175,00	87675,00	35233,18	23,86
1.001-5.000	1762,11	97,28	97377,45	99139,56	150,00	150150,00	51010,44	33,42
5.001-10.000	1762,11	91,37	456958,15	458720,27	150,00	750150,00	291429,73	30,06
10.001-100.000	1762,11	88,36	883654,89	885417,01	150,00	1500150,00	614732,99	28,59
100.001-1.000.000	1762,11	88,36	8835753,73	8837515,84	150,00	15000150,00	6162634,16	28,59
1.000.000+	1762,11	88,36	88356653,73	88358415,84	150,00	150000000,00	61641584,16	28,59

Fuente: *Personal*

CAPÍTULO 4

4. PROGRAMACIÓN Y APLICACIONES

4.1 Ejemplos básicos para el uso de la tarjeta de desarrollo con dsPICs

En este capítulo se describe varios programas desarrollados que serán aplicaciones fundamentales muy útiles para empezar con el manejo de algunos de los periféricos implementados en las tarjetas, como se mencionó en capítulos anteriores el software para el desarrollo de las aplicaciones es mikroBasic en su versión estudiantil que puede ser descargada gratuitamente desde su sitio en internet de Mikroelectronica con la limitación respecto al tamaño del programa, pero será suficiente para el desarrollo de los programas para la tarjeta de entrenamiento, mientras que para la programación del dsPIC30F4011 se hará uso del programador para dsPIC (PicKit2), o alternativamente podrá usarse el bootloader desarrollado por la compañía Ingenia, este paquete es de libre uso y dedicado a la familia de los dsPIC30F, el cual permitirá descargar en el dsPIC los programas desarrollados en mikroBasic a través del puerto serial (Ver anexo A).

Entre las aplicaciones básicas desarrolladas se encuentran el manejo de visualizadores de siete segmentos, una bocina, el uso de una pantalla cristal líquido 16x2, un teclado matricial, conversión analógica a digital de un potenciómetro, entre otras aplicaciones.

El programa descrito a continuación se refiere a la conversión analógica a digital que será mostrada a través de una pantalla de cristal líquido.

```
'============================================================
'| TÍTULO     : CONVERSIÓN ADC                                |
'| AUTORES    : O.XAVIER BARCIA R. -- ROGER A. CALLE C.       |
'| ESCUELA    : ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL      |
'| FECHA      : Julio/2008                                    |
'| VERSIÓN    : 1.0                                           |
'============================================================
'|CONFIGURACIONES DE LOS PUENTES                              |
'| - LCD                                                      |
'| - Potenciómetro uno en RB7                                 |
'|ACCIONAMIENTO                                               |
'| - Encender el interruptor de alimentación del control      |
'| - Setear el potenciómetro en una determinada posición      |
'============================================================
program ADC
'========================================================
'DECLARACIONES GLOBALES DISPONIBLE EN TODO EL PROGRAMA
'========================================================
'Declaracion de Symbols
'------------------------------------------------
symbol DELAY_medS = Delay_ms(500)     'Delay de 500 ms
symbol DELAY1S = Delay_ms(1000)     'Delay de 500 ms

'Declaracion de Variables
'------------------------------------------------
dim i as byte
```

```
'=========================================================
'P R O G R A M A    P R I N C I P A L
'=========================================================
main:
  'Variables locales
  dim adcRes as word
  dim txt,txthex as string[20]
  dim ADCVolt as float
  ADPCFG=$FF7F      'Todas Digitales, excepto RB7 como
                '   analógica
  TRISB = $80       'Se configura el puerto B como salida
  TRISC = 0         '   TRISB.7  = 1  seteado como entrada
  TRISD = 0         '   necesario para trabajar como ADC
  TRISE = 0
  TRISF = 0
  PORTB.8 = 1        'Salidas del puerto B8 en alto
  DELAY_medS         'Delay_ms(500)
  Lcd_Init(PORTD, 3,2,1,0, PORTC, 13,12,14)
  Lcd_Cmd(LCD_CURSOR_OFF)
  Lcd_Out(1, 2, "Parpadeando LED!")
  FOR i = 1 to 4
    Lcd_Out(2, 2, ".  ")
    PORTB.8 = %0
    Delay_ms(100)        'Espera para la inicialización
    Lcd_Out(2, 2, ".. ")  '       del teclado
    PORTB.8 = %1
    Delay_ms(100)        'Espera para la inicialización
    Lcd_Out(2, 2, "...")  '       del teclado
    Delay_ms(100)
  NEXT i
  Lcd_Cmd(LCD_CLEAR)
  Lcd_Out(1, 1, "ADC ==>")

  while true
    adcRes = Adc_Read(7)  'Valores  max=0x03FF = 5 Voltios
                     'min=0 voltios
    Delay_ms(50)
    ADCVolt= adcRes * 5/1023

    'sub function FloatToStr(dim fnum as real, dim byref
    '                        str as string[20]) as byte
    '0 if conversion was successful
    FloatToStr(ADCVolt, txt)
    WordToHex(adcRes, txthex)
    Lcd_Out(1, 9,"Hex:")
    Lcd_Out_Cp(txthex)
    Lcd_Out(2, 1,"V=")
    Lcd_Out_Cp(txt)
    Lcd_Out_Cp("[V]                 ")
    delay1s
  wend
end.
```

El siguiente programa describe la rutina para el uso del teclado PS/2, de tal manera que el valor de las teclas pulsadas se muestran en la pantalla de cristal líquido, para escribir en la segunda fila de esta pantalla se debe presionar la tecla enter.

```
'=============================================================
'| TÍTULO     : USO DEL TECLADO PS/2                          |
'| AUTORES    : O.XAVIER BARCIA R. -- ROGER A. CALLE C.       |
'| ESCUELA    : ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL      |
'| FECHA      : Julio/2008                                    |
'| VERSIÓN    : 1.0                                           |
'=============================================================
'|CONFIGURACIONES                                             |
'| - Configurar puestes para habilitar el Encoder             |
'| - Conexión del encodes y motor en sus respectivas          |
'|   borneras                                                 |
'| - Configurar los interruptores en posición MOTOR DC        |
'| - Conectar el cable serial del pc a la tarjeta             |
'|ACCIONAMIENTO                                               |
'| - Encender el interruptor de alimentación del motor        |
'|   y la del control.                                        |
'| - Ejecutar la interfaz de LabVIEW, ingresando los          |
'|   parámetros de velocidad en RPM, Kp, Ki y presionar el    |
'|   botón de sentido de giro del motor                       |
'=============================================================
program TecladoPS2_LCD
'=========================================================
'DECLARACIONES GLOBALES DISPONIBLE EN TODO EL PROGRAMA
'=========================================================
'dim ? as short     'short    : 8bit ->   -127...128
'dim ? as word      'word     :16bit ->      0...65535
'dim ? as integer   'integer  :16bit -> -32768...32767
'longword,dword,longint
'dim ? as           'float    :32bit -> ±1,1754943x10^38
                                 ...±6,8056477x10^38
dim i as byte       'bytes,char: 8bit->  0...255
dim sw as byte
dim  Key_capture, Key_valor, Key_especial, pressed  as  word

'=========================================================
' DECLARACIONES DE PROCEDIMIENTOS Y FUNCIONES
'=========================================================
sub procedure mostrar_caracter(dim Key_valor_p as word)
     if (Key_valor_p=13) then
       if sw=0 then
        Lcd_Out(2, 1, "")
        sw=1
```

```
      else
      Lcd_Cmd(LCD_CLEAR)
      sw=0
     end if
   end if
  if (Key_valor_p<>13) then
    Lcd_Chr_Cp(Key_valor_p)
  end if
end sub


'==========================================================
' DECLARACIONES DE PROCEDIMIENTOS Y FUNCIONES
'==========================================================
main:
  '=> ADPCFG configura los pines del puerto como entrada
  '   analógicas o entradas/salidas digitales. Ejemplo:
  '   $00FF=%0000 0000 1111 1111
  '   ADPCFG=$00FF Analógico Rb15..RB8 Digital RB7...8;

  'Configuraciones e inicializaciones
  ADPCFG=$FFFF        'Digitales RB15...RB0
  Key_valor   = 0
  Key_especial= 0
  Pressed = 0
  sw=0
   'Ps2_Config(port,clkpin,datapin)
   Ps2_Config(PORTF, 0, 1)
   'Ps2_Init(PORTC)
   Delay_ms(500)    'Espera inicialización de teclado

   TRISB = 0        'Se configura el puerto B como salida
   PORTB = 0         '%11111111 Salidas del puerto B en bajo
   PORTB.8 = 1
   for i = 1 to 3
     PORTB.0 = 1    'Se invierten la salidas del PORTB si
                  'estaban en alto se ponen en bajo y
                  'viceversa
     delay_ms(500)
     PORTB.0 = 0
     delay_ms(500)
   next i
   PORTB.7 = 0
   'Lcd_Init( byrport, db3,db2,db1,db0,ctrl_port,
   '          rs,ctrl_rw,enable)

   Lcd_Init(PORTD, 3,2,1,0, PORTC, 13,12,14)

   'Lcd_Out(1, 3, "Hello!")
   'Lcd_Cmd(LCD_CURSOR_OFF)
   'Lcd_Cmd(LCD_FIRST_ROW)
   'Lcd_Out(2, 6, "Elektronika")
```

```
    'Delay_ms(1000)
    'Lcd_Cmd(LCD_CLEAR)
    'Lcd_Out(1, 6, "Elektronika")
    'Press Enter to continue:
while(1)
  if (Ps2_Key_Read(Key_valor, Key_especial, pressed))
      then Key_capture=Key_valor
   while (Ps2_Key_Read(Key_valor, Key_especial, pressed)=0)
        PORTB.7 = 1
   wend
      mostrar_caracter(Key_capture)
      PORTB.8 = 0
      delay_ms(100) '100ms
  end if
    PORTB.8 = 1
    Delay_ms(1) ' debounce
   wend

end.
```

4.2 Desarrollo de programas para el control de motores con dsPICs

Como fue explicado en capítulos anteriores las características de diseño del dsPIC utilizado en las tarjetas están orientadas para el manejo de motores, para esto se ha desarrollado un programa que permitirá controlar la dirección del giro de un motor de corriente continua de imán permanente, el programa también incluye un control de velocidad de lazo cerrado, la realimentación de la velocidad del motor proviene de un encoder incremental acoplado al eje del motor las señales provenientes del sensor están conectados en las borneras implementadas para este propósito.

Para la implementación del controlador se partió de la ecuación continua del controlador PID en el dominio del tiempo, se halló la ecuación del PID discreto para su programación en el dsPIC30F4011 de la siguiente manera:

$$u(t) = K_c * e(t) + \frac{K_c}{T_i}\int_0^t e(t)dt + K_C T_d \frac{de(t)}{dt} + u_0$$

Para discretizar esta ecuación, se aproxima los términos de la integral y la derivada a valores que sean manipulables en el tiempo discreto.

$$\frac{\partial e(t)}{\partial t} \approx \frac{e(t) - e(t-1)}{T_S}$$

$$\int_0^t e(t)\partial t \approx T_s \sum_0^t e(i)$$

De esta forma el algoritmo del controlador PID discreto es:

$$u(t) = \mathrm{K_C} * e(t) + \frac{K_C T_S}{T_i}\sum_{i=0}^{t} e(i) + \frac{K_c T_d\left[e(t) - e(t-1)\right]}{T_S} + u_0$$

$$u(t) = \mathrm{K_p} * e(t) + \mathrm{K_i}\sum_{i=0}^{t} e(i) + \mathrm{K_d}\left[e(t) - e(t-1)\right] + u_0$$

Para controlar la velocidad del motor se ha implementado a partir de la ecuación anterior un control en modo proporcional el cual ha sido desarrollado y grabado en el dsPIC, el comportamiento del sistema para un modo proporcional se expresa en la siguiente ecuación.

$$u(t) = \mathrm{K_p} * e(t) = \mathrm{K_p}\left[y(t) - R(t)\right]$$

Donde *Kp* es la constante de proporcionalidad, *y(t)* es la velocidad actual del sistema, y *R(t)* es la velocidad de referencia del sistema, estas últimas medidas revoluciones por minuto (RPM).

El giro del motor es controlado a través de los pulsadores que dispone la tarjeta de aplicaciones básicas con dsPIC30F4011, el valor de referencia de la velocidad es establecido a través de un potenciómetro dicho valor puede fijarse para obtener una velocidad entre 0 y 90 RPM, finalmente se podrá ingresar perturbaciones para comprobar la respuesta del sistema. El programa descrito a continuación está encargado de gobernar el proceso anteriormente mencionado.

```
'============================================================
'| TÍTULO      : CONTROL PI PARA MOTOR DC USANDO LabVIEW       |
'| AUTORES     : O.XAVIER BARCIA R. -- ROGER A. CALLE C.       |
'| ESCUELA     : ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL      |
'| DESCRIPCIÓN:                                                |
'| FECHA       : Septiembre /2008                              |
'| VERSIÓN     : 1.0                                           |
'============================================================
'|CONFIGURACIONES                                              |
'| - Configurar puentes para habilitar el encoder              |
'| - Conexión del encoder y motor en sus respectivas           |
'|   borneras                                                  |
'| - Cofigurar los interruptores en posición del MOTOR DC      |
'| - Conectar el cable serial del pc a la tarjeta              |
'|ACCIONAMIENTO                                                |
'| - Encender el interruptor de alimentación del motor         |
'|   y la del control.                                         |
'| - Ejecutar la interfaz de LabVIEW, ingresando los           |
'|   parámetros de velocidad en RPM, Kp, Ki y presionar el     |
'|   botón de sentido de giro del motor                        |
'============================================================
```

```
'===========================================================
program Encoder_MotorDC
'===========================================================
symbol boton_derecha  = PORTF.6
symbol boton_izquierda= PORTB.7 'PORTE.0
symbol boton_parar  = PORTE.8
symbol Luz = PORTB.8
symbol Encender = 0
symbol Apagar = 1
symbol Frenando = Estado.0   'Frenando
symbol Detenido = Estado.1   'Detenido
symbol Izquierda = Estado.2  'Girando izquierda
symbol Derecha = Estado.3    'Girando derecha
symbol DatoListo = Estado.4  'Dato listo para enviar

'La multiplicación por 100 es para que no quede valores < 1
'RPM=V*60/1024, sabiendo que en 1024 pasos se dan en 0.5 seg.

'Muestreo a 500ms
symbol Desp2RPM = 5.859375   '60/1024*100 = 60/(2048/2)*100
'Muestreo a 250ms
'symbol Desp2RPM = 11.718750   '60/512*100 = 60/(2048/4)* 100
'Muestreo a 200ms
'symbol Desp2RPM = 14.648438   '60/409.6*100 = 60/(2048/5)*100
'Muestreo a 25ms
'symbol Desp2RPM = 117.1875    '60/409.6*100 = 60/(2048/5)*100
'symbol Kp = 2
symbol VelocidadRefrescar = 250
symbol TiempoEspera = 10000

dim Estado as word
dim i, duty_50,duty_100,iDuty,ADCporcentaje as word
dim intADC as integer
dim txt,txthex,txtint,txtfloat as string[20]
dim vuelta,ADCRes,ADCword,iDuty_word as word
dim desplazamiento, posicion_anterior as word
dim RPMRef,iDuty_float,RPM as float
dim et, ut as float
dim yt as longint
dim VelocidadRefrescarContador as word
'Variables para el manejo del puerto serial
dim RecibirCaracter as word
dim utAux as longint

sub procedure Detener()
   Frenando = 0
   Izquierda = 0
   Derecha = 0
   Detenido = 1
   Pwm_Mc_Set_Duty(0, 3)
   Pwm_Mc_Set_Duty(0, 2)
end sub
```

```
sub procedure CalcularVelocidadActual()
dim posAux as word
dim xRPM as float
dim xTMR4 as word
   xTMR4 = TMR4
   'enceramos el timer
   TMR4 = 0
   'calculamos desplazamiento
   posAux = POSCNT
   if Izquierda = 1 then
     desplazamiento = posicion_anterior - posAux
     Detenido = 0
   end if
   if Derecha = 1 then
     desplazamiento = posAux - posicion_anterior
     Detenido = 0
   end if
   if Frenando = 1 then
     if desplazamiento = 0 then
        Detener()
     end if
   end if
   posicion_anterior = posAux
   'Calculamos el RPM basandose en el TMR4
   'En medio segundo tenemos del TMR4 57598 pasos
   'RPM=(x vueltas/t minutos)=
   '     (60*2*57598/2048)*(desplazamiento/xTMR4)
   RPM = (3374.94140625*desplazamiento) / xTMR4
  'Enviamos los RPM
   xRPM = RPM * 100000
   yt = xRPM
end sub
sub procedure Timer4Int org $3E
   CalcularVelocidadActual()
   IFS1.5 = 0  'Limpiamos la interrupción
end sub

sub procedure Controlar()
dim xUno, xDos, xTres, xCuatro as word
dim xRetardo as word
  CalcularVelocidadActual()
 'Enviamos un nuevo valor de y(t)
 'Uart2_Write_Char($FF)
  Uart2_Write_Char(Highest(yt))
  Uart2_Write_Char(Higher(yt))
  Uart2_Write_Char(Hi(yt))
  Uart2_Write_Char(Lo(yt))
  xRetardo = 0
  while (Uart2_Data_Ready() = 0)
       xRetardo = xRetardo + 1
       if xRetardo > TiempoEspera then
          exit
```

```
      end if
  wend
'Esperamos el siguiente caracter
  xUno = Uart2_Read_Char()
  xRetardo = 0
  while (Uart2_Data_Ready() = 0)
      xRetardo = xRetardo + 1
      if xRetardo > TiempoEspera then
        exit
      end if
  wend  'Esperamos el siguiente caracter
  xDos = Uart2_Read_Char()
  xRetardo = 0
  while (Uart2_Data_Ready() = 0)
      xRetardo = xRetardo + 1
      if xRetardo > TiempoEspera then
        exit
      end if
  wend  'Esperamos el siguiente caracter
  xTres = Uart2_Read_Char()
  xRetardo = 0
  while (Uart2_Data_Ready() = 0)
      xRetardo = xRetardo + 1
      if xRetardo > TiempoEspera then
        exit
      end if
  wend  'Esperamos el siguiente caracter

  xCuatro = Uart2_Read_Char()

  'Unimos los segmentos del word
  utAux = xUno
  utAux = utAux << 8
  utAux = utAux or xDos
  utAux = utAux << 8
  utAux = utAux or xTres
  utAux = utAux << 8
  utAux = utAux or xCuatro

  ut = utAux / 100000
  'Mostramos la corrección recibida
  FloatToStr(ut, txtfloat)
  Lcd_Out(2, 9,"ut :    ")
  Lcd_Out(2,13,txtfloat)
  'control proporcional
  'et = RPMRef - RPM     'Viene desde LabVIEW
  'ut = Kp * et          'Viene desde LabVIEW
  if Detenido = 0 then
    iDuty_word = iDuty_word + ut
    if Izquierda = 1 then
      Pwm_Mc_Set_Duty(iDuty_word, 3)
    end if
    if Derecha = 1 then
```

```
      Pwm_Mc_Set_Duty(iDuty_word, 2)
    end if
  end if
end sub

main:
  'DESABILITACIÓN DE INTERRUPCIONES
  IEC0 = 0        'Registro de control de habilitación de
                'interrupciones(IECx)
  IEC1 = 0        'Desabilitación de interrupciones
  IEC2 = 0
  'INICIALIZACION DE VARIABLES
  Estado = 0
  desplazamiento = 0
  ut = 0
  Derecha  =0
  Izquierda=0
  Detenido = 1

  'CONFIGURACIONES DE PUERTOS
  'ADPCFG=$FFBF, todas digitales, excepto RB6 como analógica
   ADPCFG=$FFFF
  TRISB  = $00F8       'C0 ≡ %010000000  'Salida: Led B8
                   'Entrada: potenciometro B6,
                    'IZQUIERDA B7
 'TRISE  = %100000000  'Entrada: botonera PARO E8
 'TRISF  = %001000000  'Entrada: botonera F6 DERECHA

 'INICIALIZACIÓN DEL PUERTO SERIAL
  Uart2_Init(19200)
  PORTB.8 = 1
  Delay_ms(100)

  'CONFIGURACIÓN E INICIALIZACIÓN DEL ENCODER
  POSCNT = 0
  'QEICONbits.QEISIDL = 0  'Continuar en modo idle (0)
   QEICON.13 = 0
  'QEICON.QEIM = 0b100     'Resolución 4x, modo de Reset
                        'por MAXCNT
   QEICON.10 = 1
   QEICON.9  = 1
   QEICON.8  = 0
  'QEICONbits.SWPAB = 0    'Phase-A y Phase-B pines
                        'originales
   QEICON.7 = 0
  'QEICONbits.PCDOUT = 1   'Activado el pin UPDN para
           'informar el sentido de giro del encoder
   QEICON.6  = 1
  'QEICON.TQGATE = 0        'Timer gate apagado
   QEICON.5 = 0
  'QEICONbits.TQCKPS = 0     'Prescaler 1:1
   QEICON.4= 0
   QEICON.3= 0
```

```
'QEICONbits.POSRES = 0     'Un pulso en INDEX no hace un
                          'reset en POSCNT
 QEICON.2 = 0
'QEICONbits.TQCS =0        'Usamos clock interno para el
                          'timer  Tcy
 QEICON.1 =0
'QEICONbits.UDSRC=1        'Phase-B indica dirección
 QEICON.0=1
 MAXCNT=1000
 DFLTCON = 0x0000
'ACTIVACIÓN DE INTERRUPCIÓN
posicion_anterior = 0
TMR4 = 0
PR4 = 57599     'Muestreo de 500ms
'PR4 = 28799    'Muestreo de 250ms
'PR4 = 23039    'Muestreo de 200ms
'PR4 = 2880     'Muestreo de 25ms
IFS1.5 = 0      'Limpiamos la interrupción
IEC1.5 = 1      'Activación de interrupción para Timer4
T4CON = $8030   'Habilitamos Timer4 y Prescaler 1:256
'T4CON = $8000  'Habilitamos Timer4 y Prescaler 1:1
'T4CON = $8010  'Habilitamos Timer4 y Prescaler 1:8
'T4CON = $8020  'Habilitamos Timer4 y Prescaler 1:64

'CONFIGURACIÓN E INICIALIZACIÓN DE LCD
Lcd_Init(PORTD, 3,2,1,0, PORTC, 13,12,14)
Lcd_Cmd(LCD_CURSOR_OFF)

'INICIALIZACIÓN FRECUENCIA DEL MOTOR A 50Hz
'sub function Pwm_Mc_Init(dim freq_hz,
                        'pair_output_mode,
                        'enable_output_x,
     'clock_prescale_output_postscale as word) as word

'Pwm_Mc_Init retorna 50% del ciclo de trabajo
'f=3200 Hz y un presalador de 1:64 obtenemos =>
                       'duty_50= CicloTrabajo_50%=9215
duty_50 = Pwm_Mc_Init(3200,1,%01100110,0)
duty_100 =duty_50*2
PWMCON1=%111101100110 'Configuración de pwm en modo
                     'independiente y habilitación de
                     'PWM3H,PWM2H,PWM3L,y PWM2L
PTCON.3=1            'Configuración de prescaladores
PTCON.2=1
Pwm_Mc_Set_Duty(0, 2) 'Pwm_Mc_Set_Duty(dim duty,
                     'channel as word)
Pwm_Mc_Set_Duty(0, 3)
Pwm_Mc_Start()       'Inicia el modulo PWM

'INDICADOR DE FUNCIONAMIENTO DEL dsPIC
FOR i = 1 to 3
  PORTB.8 = 0
  Delay_ms(100)
```

```
    PORTB.8 = 1
    Delay_ms(100)
  NEXT i

  iDuty_word = 0
  while true
      'Comandos que recibe la tarjeta
    if Uart2_Data_Ready() = 1 then
       RecibirCaracter = Uart2_Read_Char()
       select case RecibirCaracter
       case "I"    'Comando Izquierda
           Izquierda = 1
       case "D"    'Comando Derecha
           Derecha = 1
       case "P"    'Comando parar
           Detener()
       case "Y"    'Comando reportar salida
           Controlar()
       case "N"
           Uart2_Write_Text("N15")
       end select
    end if
'ciclo de refrescado
  VelocidadRefrescarContador = VelocidadRefrescarContador + 1
    if VelocidadRefrescarContador > VelocidadRefrescar then
       'Mostramos el duty calculado
       'FloatToStr(RPMRef, txtfloat)
       'Lcd_Out(2, 9,"RPM:    ")
       'Lcd_Out(2,13,txtfloat)
       'Mostramos el DutyActual

        wordToStr(iDuty_word, txtfloat)
        Lcd_Out(1, 2,txtfloat)

       'Posicion actual
        wordToStr(POSCNT, txtfloat)
        Lcd_Out(1,8,txtfloat)
       'Muestra la velocidad real en RPM
        FloatToStr(RPM, txtfloat)
        txtfloat[5] = 0
        Lcd_Out(2, 1,txtfloat)
       'Indicamos la direccion
        if (QEICON and $0800)  then
          Lcd_Out(1, 1,"D")
        else
           Lcd_Out(1, 1,"I")
        end if
     end if

  wend
end.
```

4.3 Manejo de motores de paso utilizados en la plataforma de desarrollo

La tarjeta de entrenamiento con dsPIC30F4011, puede ser configurado para el manejo de motores de paso, para lo cual se ha desarrollado el programa que permitirá manejar esta clase de motor.

```
'==========================================================
'| TÍTULO     : MANEJO DE MOTOR DE PASO UNIPOLAR            |
'| AUTORES    : O.XAVIER BARCIA R. -- ROGER A. CALLE C.     |
'| ESCUELA    : ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL    |
'| FECHA      : Julio/2008                                  |
'| VERSIÓN    : 1.0                                         |
'==========================================================
'|CONFIGURACIONES                                          |
'| - No es necesario configura los puentes para en motor    |
'| - Configurar los puentes para el teclado matricial y el  |
'|   LCD                                                    |
'|ACCIONAMIENTO                                             |
'| - Encender el interruptor de alimentación del motor      |
'|   y la del control.                                      |
'| - Las teclas del 1 al 9 en el teclado matricial          |
'|   se usa para configurar la velocidad del motor          |
'| - La tecla asterisco y numeral es para el giro del       |
'|   motor hacia la izquierda o derecha respectivamente     |
'| - El pulsador de interrupción es usado para el paro      |
'|   del motor                                              |
'==========================================================
Program MotorPaso
'======================================================
'DECLARACIONES GLOBALES DISPONIBLE EN TODO EL PROGRAMA
'======================================================
'Declaración de Symbols
'------------------------------------------------
symbol nLED   = PORTB.8
symbol Phase1 = PORTE.2
symbol Phase2 = PORTE.3
symbol Phase3 = PORTE.4
symbol Phase4 = PORTE.5
symbol MotorPaso = PORTE

'Declaración de Variables
'------------------------------------------------
dim SW_Derecha,SW_Izquierda  as byte
dim SW_Velocidad,SW_Stop as byte
dim kp,Velocidad as word
```

```
dim kps as string[2]
dim kpss as string[5]
dim i as word
dim cpaso as integer
dim cnt as longint

'Declaración de Constantes
'-----------------------------------------------
const Encender as byte = 1
const Apagar   as byte = 0
const Npaso as byte =8
'Secuencia de MEDIO PASO 1   2   3   4   5   6   7   8
const paso as word [8]=
      ($0004,$000C,$0008,$0018,$0010,$0030,$0020,$0024)

'=========================================================
' DECLARACIONES DE PROCEDIMIENTOS Y FUNCIONES
'=========================================================
sub procedure nLed_blink ()
  for i = 1 to 2
    nLED = 0
    delay_ms(500)
    nLED = 1
    delay_ms(500)
  next i
end sub

sub procedure Config_ExtInt0 ()
 INTCON2=$FFFE  'INTCON2<0>=INT0EP=0, Habilitación por
                              'flanco positivo
 IEC0=$0001     'IEC0 <0>  =INTOIE=1, Habilitación de
                         'interrupción externa 0
 IPC0=$7        'INTOIP<2:0>=$111 Prioridad alta para INT0
end sub

sub procedure ExternalInt0 org $94  'direccionar interrupción
 IFS0 = IFS0 and $FFFE          'Clear IFS0<0>=INT0IF
 nLED = 0  'not PORTB.8
 delay_ms(500)
 nLED = 1
 SW_Stop=1
 Keypad_Init(PORTB)
 Delay_ms(100)
 'Lcd_Out(1, 13, "Stop")
 Lcd_Out(1, 1, "VEL:   ESTD:Stop")
 'Lcd_Out(1, 5, " ")
end sub
```

```
sub function keyPulse_string (dim kpf as word) as string[2]
    select case kpf
      case  1 result = "1"
      case  2 result = "2"
      case  3 result = "3"
      case  4 result = "A"
      case  5 result = "4"
      case  6 result = "5"
      case  7 result = "6"
      case  8 result = "B"
      case  9 result = "7"
      case 10 result = "8"
      case 11 result = "9"
      case 12 result = "C"
      case 13 result = "*"
      case 14 result = "0"
      case 15 result = "#"
      case 16 result = "D"
    end select
end sub

sub procedure Velocidad_motor (dim Veloc as word)
    select case veloc
      case  1 delay_ms(800)
      case  2 delay_ms(200)
      case  3 delay_ms(50)
      case  5 delay_us(12500)
      case  6 delay_us(3125)
      case  7 delay_us(881)
      case  9 delay_us(100)
      case 10 delay_us(25)
      case 11 delay_us(1)
    end select
end sub

sub procedure Giro_Derecha ()
 cpaso=(cpaso+1) mod  Npaso
 MotorPaso=paso[cpaso]
 inc(cnt)
 WordToStr(cnt, kpss)
 Lcd_Out(2, 8, kpss)
 Velocidad_motor(Velocidad)
end sub

sub procedure Giro_Izquierda()
 cpaso= (cpaso-1) mod  Npaso
 if cpaso = -1 then cpaso=7 end if
 MotorPaso=paso[cpaso]
 dec(cnt)
 WordToStr(cnt, kpss)
 Lcd_Out(2, 8, kpss)
```

```
 Velocidad_motor(Velocidad)
end sub

'=========================================================
'P R O G R A M A   P R I N C I P A L
'=========================================================
main:
  ADPCFG= $FFFF   'Todas Digitales
  TRISB = $000F   'Salida: nLed RB8   Entrada/salida:Teclado Hex <RB7:RB0>
  TRISE = $0100   'Entrada: botón Int_Ext de parada
  TRISF = $0040   'Entrada: botón de giro_derecha RF6
  delay_ms(50)
  nLED = 1
  nLed_blink()
  Delay_ms(50)
  Config_ExtInt0()
  Delay_ms(50)
  Keypad_Init(PORTB)
  Delay_ms(100)
  Lcd_Init(PORTD, 3,2,1,0, PORTC, 13,12,14)
  Lcd_Cmd(LCD_CURSOR_OFF)
  SW_Derecha  =0
  SW_Izquierda=0
  SW_Velocidad=0
  SW_Stop=1
  Keypad_Init(PORTB)
  Delay_ms(100)
  Lcd_Out(1, 2, "MOTOR DE PASO")
  Lcd_Out(2, 5, "UNIPOLAR")
  delay_ms(2000)
  Lcd_Cmd(LCD_CLEAR)
  Lcd_Out(1, 1, "VEL:   ESTD:Stop")
  Lcd_Out(2, 1, "#Pasos:")
  kp = 0
  cnt=0
  cpaso=0
  while true

    if SW_Stop=1  then
      SW_Stop=0
      while (kp=0) or (kp=4) or (kp=8) or (kp>=12)
        kp = Keypad_Key_Click()
        delay_us(100)
      wend
      'WordToStr(kp, kps)
      kps= keyPulse_string(kp)
      Lcd_Out(1, 5, kps)
      Velocidad= kp
      kp=0
      while (kp=0) or (kp<=12) or (kp=16)
        kp = Keypad_Key_Click()
        delay_us(100)
      wend
```

```
   select case kp
    case  13
       SW_Izquierda = 1
       SW_Derecha   = 0
       Lcd_Out(1, 13, " Izq")
    case  14
       SW_pasoXpaso = 1
    case  15
       SW_Derecha   = 1
       SW_Izquierda = 0
       Lcd_Out(1, 13, " Der")
   end select
  SW_Stop=0
 end if

 if (SW_Derecha=1) and (SW_Stop = 0) then
   Giro_Derecha()
 end if
 if (SW_Izquierda=1)and (SW_Stop = 0)  then
   Giro_Izquierda()
 end if
 if cnt=0 then
   SW_Stop = 1
   Lcd_Out(1, 1, "VEL:   ESTD:Stop")
 end if
wend
end.
```

4.4 Manejo de servo motores

Para el manejo de servomotores la tarjeta de entrenamiento con dsPIC30F4011 dispone de pines para su conexión, a continuación se muestra el programa desarrollado para ubicar en un determinado ángulo este tipo de motor.

```
'==============================================================
'| TÍTULO     : MANEJO DE SERVO MOTOR                          |
'| AUTORES    : O.XAVIER BARCIA R. -- ROGER A. CALLE C.        |
'| ESCUELA    : ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL       |
'| DESCRIPCIÓN:                                                |
'| FECHA      : Julio/2008                                     |
'| VERSIÓN    : 1.0                                            |
'==============================================================
'|CONFIGURACIONES DE LOS PUENTES                               |
'| - LCD                                                       |
'| - Teclado 4x3                                               |
'| - Conector del servo motor RE0                              |
'|   (no necesita confurar puente)                             |
'|ACCIONAMIENTO                                                |
'| - Digitar tres dígitos a través del teclado debe estar      |
'|   entre 0 a 180 grados seguidamente presionar el botón      |
'|   asterisco para ejecutar la acción, si necesitamos         |
'|   cambiar posición debemos presionar la tecla numeral       |
'==============================================================
program ServoMotor
'=========================================================
'DECLARACIONES GLOBALES DISPONIBLE EN TODO EL PROGRAMA
'=========================================================
'Declaracion de Symbols
'-----------------------------------------------
symbol nLED   = PORTB.8

'Declaracion de Variables
'-----------------------------------------------
dim i, duty_50 as word
dim kp as word
dim kpi as integer
dim kpf,Direccion_f as float
dim kpf_MSB,kpf_ISB,kpf_LSB  as float
dim kps_t  as string[3]
dim txt as string[3]
dim grado_f as float
dim kps as string[2]
```

```
'==========================================================
' DECLARACIONES DE PROCEDIMIENTOS Y FUNCIONES
'==========================================================
sub procedure nLed_blink ()
  for i = 1 to 2
    nLED = 0
    delay_ms(500)
    nLED = 1
    delay_ms(500)
  next i
end sub

sub procedure init_fpwm_50Hz()
  'sub function Pwm_Mc_Init(dim freq_hz,
         'pair_output_mode, enable_output_x,
         'clock_prescale_output_postscale as word) as word

  'Pwm_Mc_Init retorna 50% del ciclo de trabajo
  'f=3200 Hz y un presalador de 1:64 obtenemos =>
                    'duty_50= CicloTrabajo_50%=9215
  duty_50 = Pwm_Mc_Init(3200,1, %00000001,$0)
  PTCON.3=1
  PTCON.2=1
  Pwm_Mc_Start()
  PORTE=0
end sub

sub procedure keyPulse (dim kp_pulse as word)
    select case kp_pulse
      case  1  kpf=1
        kps = "1"
      case  2 kpf=2
        kps = "2"
      case  3 kpf=3
        kps = "3"
      case  4 kps = "A"
      case  5 kpf=4
        kps = "4"
      case  6 kpf=5
        kps = "5"
      case  7 kpf=6
        kps = "6"
      case  8 kps = "B"
      case  9 kpf=7
        kps = "7"
      case 10 kpf=8
        kps = "8"
      case 11 kpf=9
        kps = "9"
      case 12 kps = "C"
      case 13 kps = "*"
      case 14 kpf=0
        kps = "0"
```

```
    case 15 kps = "#"
    case 16 kps = "D"
  end select
end sub

'==========================================================
'P R O G R A M A   P R I N C I P A L
'==========================================================
main:
 ADPCFG= $FFFF   'Todas Digitales RB
 TRISB = $000F   'Salida: nLed RB8
                'Entrada/salida:Teclado Hex <RB7:RB0>
 delay_ms(50)
 nLED = 1
 nLed_blink()
 Delay_ms(50)
 Keypad_Init(PORTB)
 Delay_ms(100)
 PORTE=0
 Lcd_Init(PORTD, 3,2,1,0, PORTC, 13,12,14)
 Lcd_Cmd(LCD_CURSOR_OFF)
 init_fpwm_50Hz()

 Lcd_Out(1, 3,"SERVO  MOTOR" )
 delay_ms(2000)
 Lcd_Out(1, 1, "S.MOTOR EST:Stop" )
 Lcd_Out(2, 1, "Gr:   Dty:      " )
 kp = 0
 kps_t="   "
 while true
  kp=0
  while (kp=0) or (kp=4) or (kp=8) or (kp=12) or (kp=13) or (kp>=15)
   kp = Keypad_Key_Click()
   delay_us(100)
  wend
  keyPulse(kp)
  kpf_MSB=kpf
  Lcd_Out(2, 4, kps)
  kp=0
  kpi=0
  while (kp=0) or (kp=4) or (kp=8) or (kp=12) or (kp=13)
      or (kp>=15)
   kp = Keypad_Key_Click()
   delay_us(100)
  wend
  keyPulse(kp)
  kpf_ISB=kpf
  Lcd_Out(2, 5, kps)
  kp=0
  kpi=0

  while (kp=0) or (kp=4) or (kp=8) or (kp=12) or (kp=13)
   or (kp>=15)
```

```
    kp = Keypad_Key_Click()
    delay_us(100)
  wend
  keyPulse(kp)
  kpf_LSB=kpf
  Lcd_Out(2, 6, kps)
  kp=0
  kpi=0
  while (kp=0) or (kp<=12) or (kp=14) or (kp=16)
    kp = Keypad_Key_Click()
    delay_us(100)
  wend
  if kp=13 then
    Lcd_Out(1, 13, "Actv")
    grado_f= (100*kpf_MSB) + (10*kpf_ISB) + (kpf_LSB)
    Direccion_f= 500 + 1600 * grado_f/180
    FloatToStr(Direccion_f, txt)
    Lcd_Out(2, 11, txt)
    Lcd_Out(2, 15,"us")
    Pwm_Mc_Set_Duty(Direccion_f, 1)
  end if
  while (kp=0) or (kp<=14) or (kp=16)
    kp = Keypad_Key_Click()
    delay_us(100)
  wend
  Lcd_Out(1, 1, "S.MOTOR EST:Camb" )
  Lcd_Out(2, 1, "Gr:   Dty:      " )
 wend
end.
```

4.5 Ejemplos de comunicación

Para realizar aplicaciones con comunicación se escogió el tipo serial de tal manera que haciendo uso del conector DB9 disponible en las tarjetas, se desarrollo una aplicación con el programa LabVIEW, con la finalidad de realizar una interfaz que permita gobernar un motor de corriente continua conectado a los bornes de la tarjeta, la cual se comportará como un módulo de adquisición de datos, debido a que la tarjeta intercambiará datos con el computador, a través del protocolo de comunicación serial RS-232.

La aplicación desarrollada permite enviar órdenes para la velocidad, direccionamiento, y el detenido del motor, mientras que el dsPIC de la tarjeta contiene el programa que se encarga de ejecutar las órdenes recibidas desde la aplicación.

Figura 4.1 Panel frontal para interfaz del control de motor

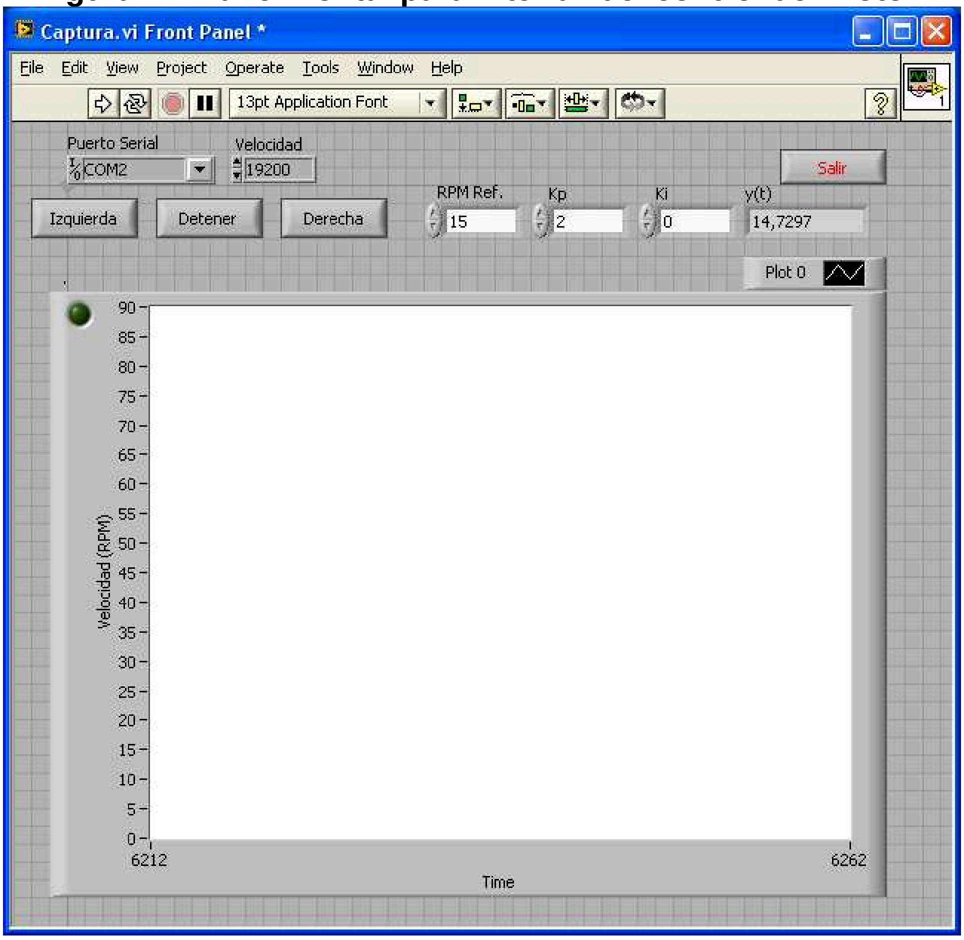
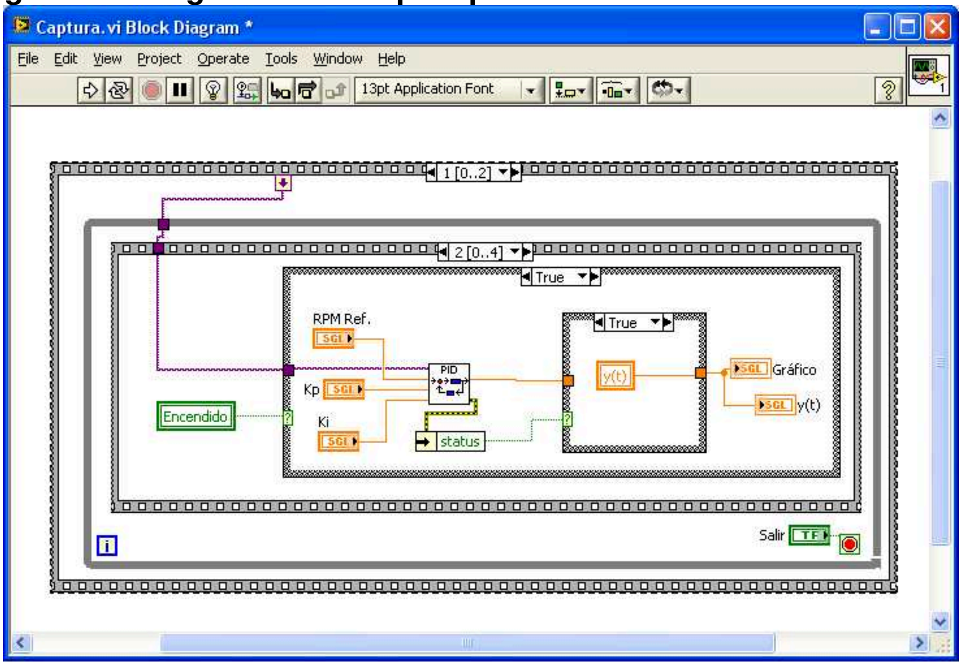


Figura 4.2 Diagrama de bloques para interfaz del control de motor



En el anexo E se encontrará los diagramas de flujo de cada programa presentado durante el desarrollo de este capítulo, en cada diagrama se bosquejara el programa principal y las subrutinas utilizadas por el mismo.

CAPÍTULO 5

5. DATOS EXPERIMENTALES

5.1 Pruebas realizadas

5.1.1 Operación del ciclo de trabajo de los motores

Para comprobar los parámetros de funcionamiento del puente manejador para motores se probó la operación de los mismos con diferente modulación por ancho de pulso y con dos tipos de motores de corriente continua uno de imán permanente y otro de bobinado en serie. En las pruebas realizadas se verificó el ciclo de trabajo en los motores, el voltaje máximo, mínimo y promedio que están manejando, así como la frecuencia de operación del mismo, y la curva de modulación del voltaje que recibe el motor.

La modulación de ancho de pulso fue realizada utilizando ***conmutación de voltaje bipolar*** de los transistores, donde los pares de mosfet Q1-Q4 y Q2-Q3, trabajan como un par de interruptores; la conmutación de cada pareja es simultáneamente realizada de manera complementaria, es decir que mientras un par de mosfet recibe una determinada modulación el otro par recibe el complemento de señal antes mencionada.

La frecuencia de las señales de conmutación es generada por el dsPIC, el cual internamente genera una onda triangular para realizar una comparación con el voltaje de control. Cuando $V_{control} > V_{onda\ triangular}$, entonces los mosfet Q1-Q4 son encendidos, mientras que los transistores Q2-Q3 son apagados, caso contrario se tiene un comportamiento de encendido para Q2-Q3 y apagado en Q1-Q4.

5.1.2 Comportamiento del voltaje y corriente de los motores

Para realizar esta prueba en ambos motores se fijó a un ciclo de trabajo de 32%, y una frecuencia de operación de 200 hertz para la obtención de la curva de corriente se usó una resistencia prueba de 1 ohm en serie con el motor, mientras que la curva de voltaje fue obtenida directamente entre las terminales del motor. Otros parámetros obtenidos de esta prueba son los valores máximo, mínimo y promedio para la corriente y voltaje en los motores.

5.1.3 Operación del motor DC usando controlador PI

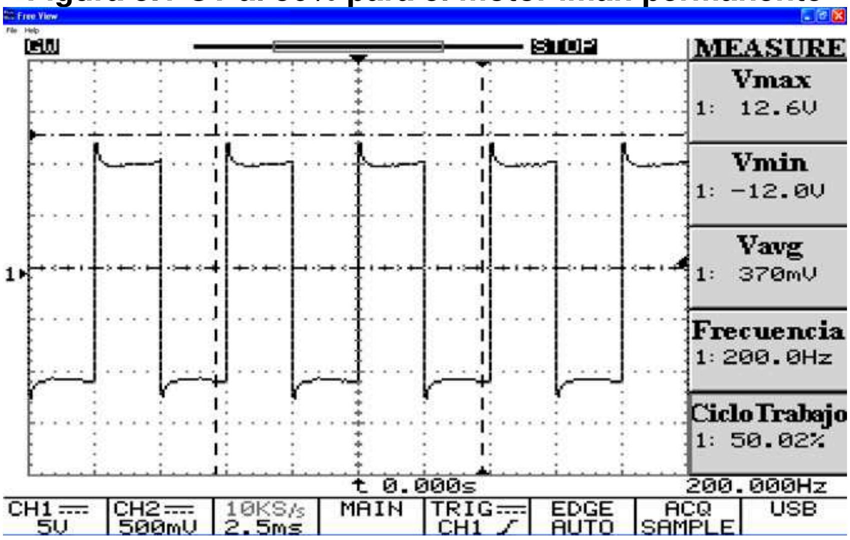
Esta prueba tuvo como objetivo mejorar la respuesta de estabilización de un motor de corriente continua, para observar e interactuar el comportamiento de la respuesta del motor se desarrollo una aplicación en LabVIEW, permitiendo variar las constantes del controlador PI, con la finalidad de escoger la mejor respuesta, para obtener un correcto funcionamiento del control implementado se ha integrado al sistema un sensor de desplazamiento que como se menciono anteriormente es el encoder, el controlador que gobierna a nuestro actuador es el dsPIC30F4011, finalmente el manejador de motores conformado por los transistores mosfet constituyen el actuador que modificará al sistema según sea el caso.

5.2 Datos obtenidos

5.2.1 Curvas de operación del ciclo de trabajo de los motores

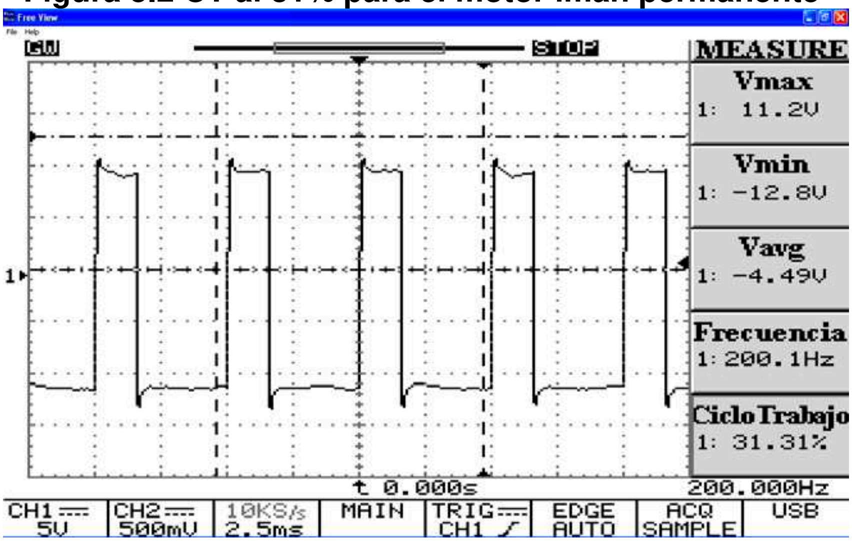
En las siguientes figuras se presenta las curvas obtenidas para diferentes ciclos de trabajo.

Figura 5.1 CT al 50% para el motor imán permanente



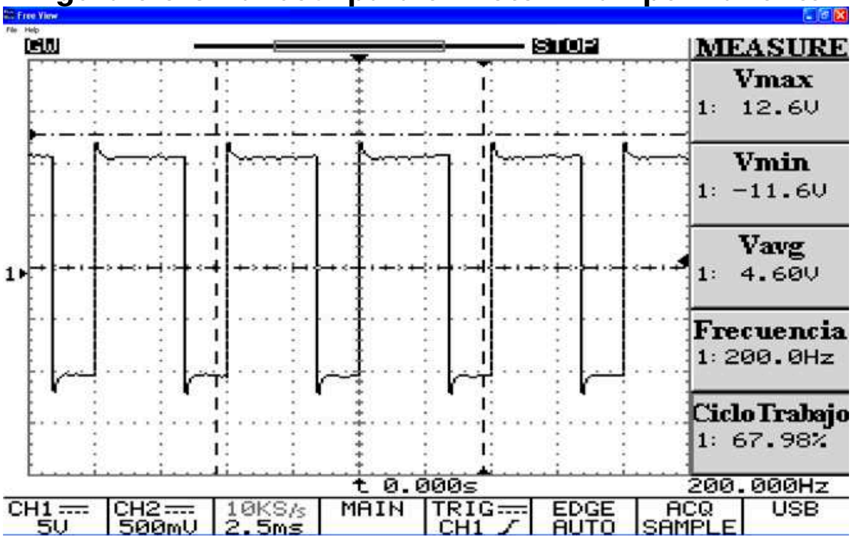
FUENTE: *Personal*

Figura 5.2 CT al 31% para el motor imán permanente



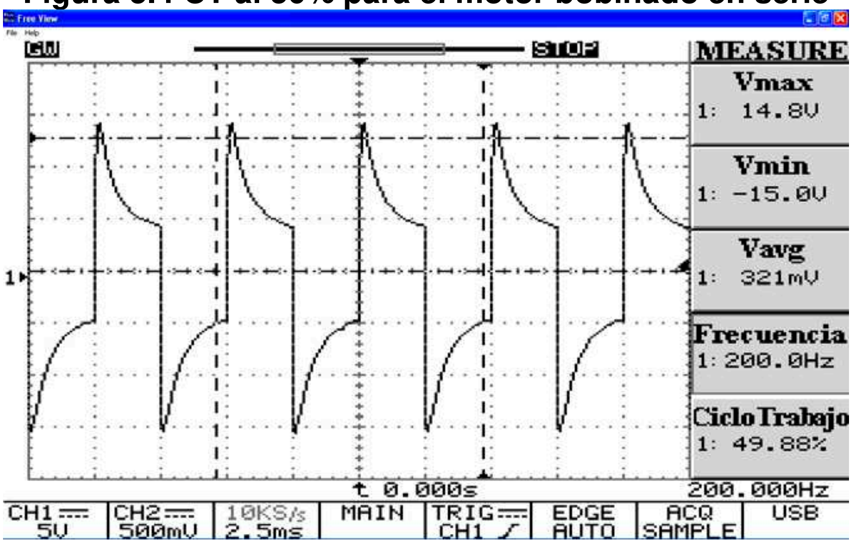
FUENTE: *Personal*

Figura 5.3 CT al 68% para el motor imán permanente



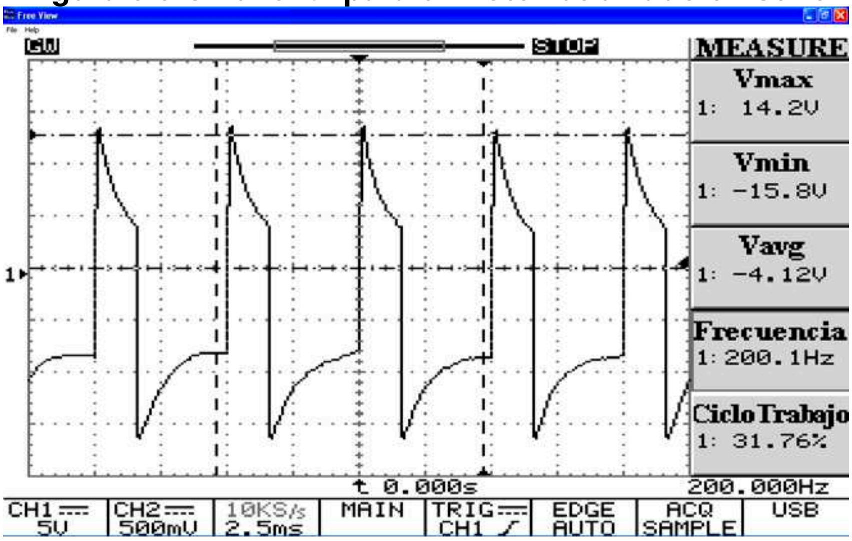
FUENTE: *Personal*

Figura 5.4 CT al 50% para el motor bobinado en serie



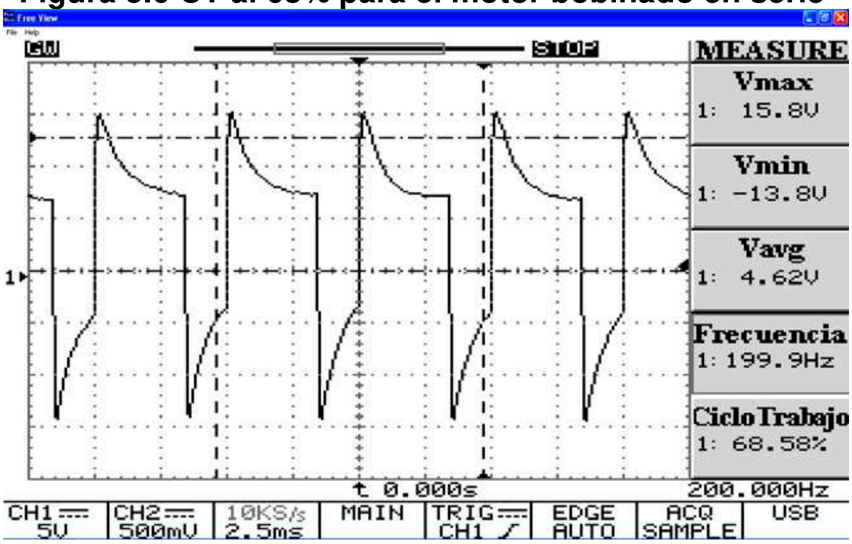
FUENTE: *Personal*

Figura 5.5 CT al 31% para el motor bobinado en serie



FUENTE: *Personal*

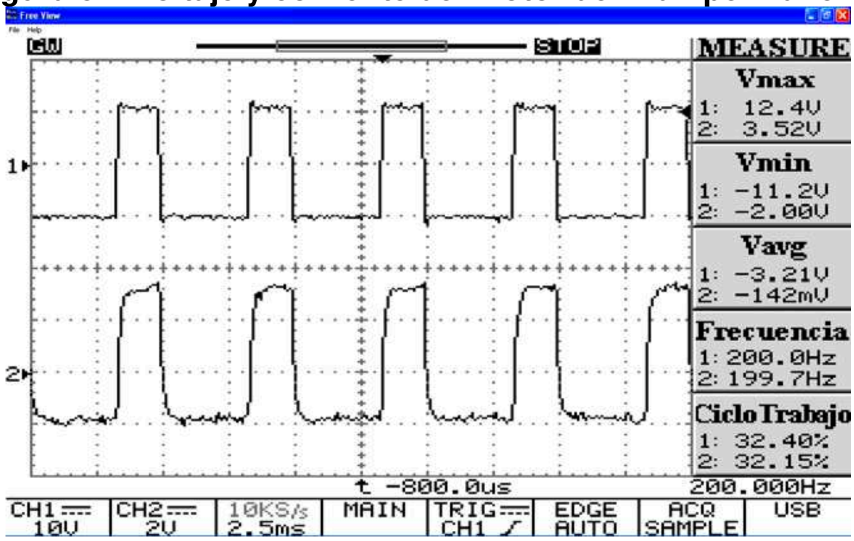
Figura 5.6 CT al 68% para el motor bobinado en serie



FUENTE: *Personal*

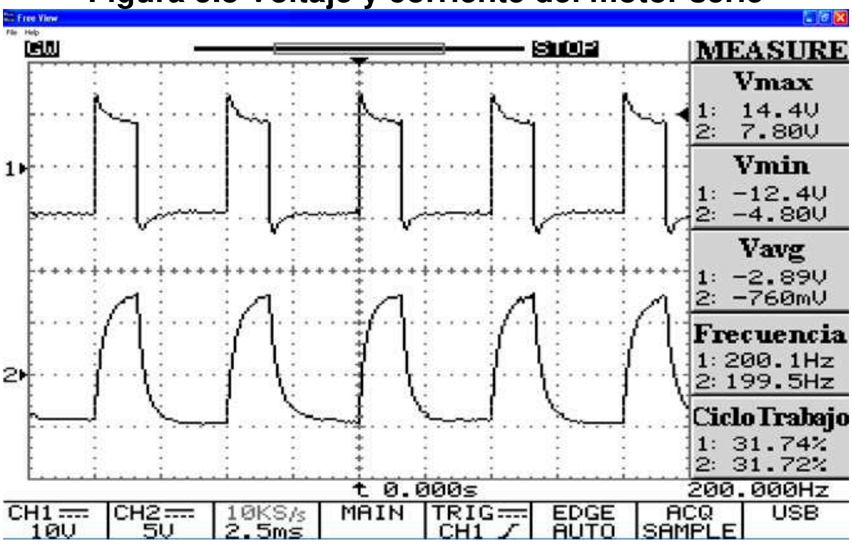
5.2.2 Curvas de comportamiento del voltaje y corriente de los motores

Figura 5.7 Voltaje y corriente del motor de imán permanente



FUENTE: *Personal*

Figura 5.8 Voltaje y corriente del motor serie



FUENTE: *Personal*

5.2.3 Curvas de operación del motor DC usando controlador PI

Para la obtención de estas curvas se hizo uso de la aplicación desarrollada y explicada la sección 5.1.3, en las siguientes figuras se muestran diferentes respuestas del motor, las cuales obedecen al PI que realiza el control de acuerdo a las constantes seleccionadas.

Figura 5.9 Respuesta del motor a inadecuados parámetros de kp y ki

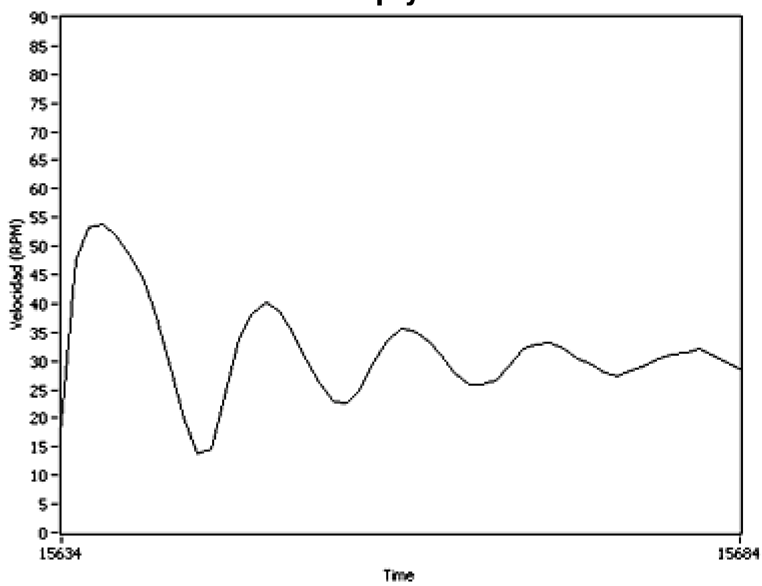


Figura 5.10 Respuesta del motor al controlador kp=10

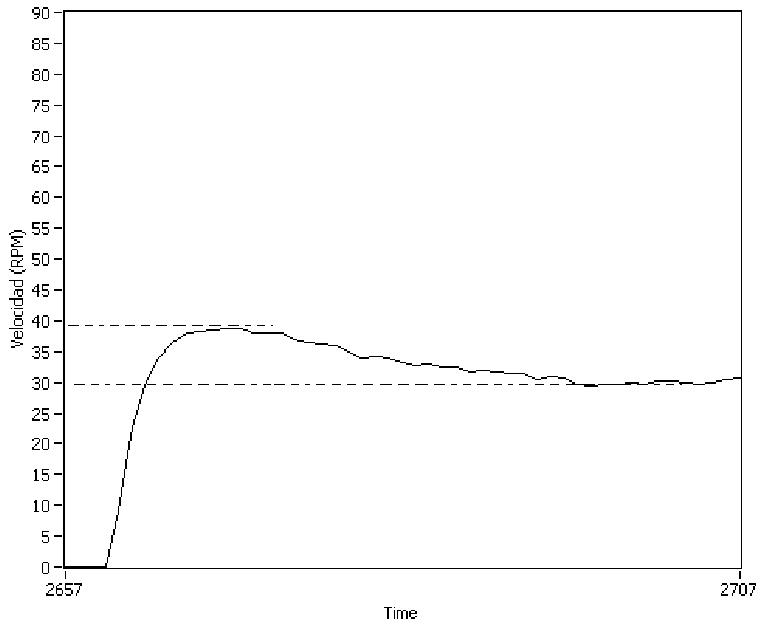
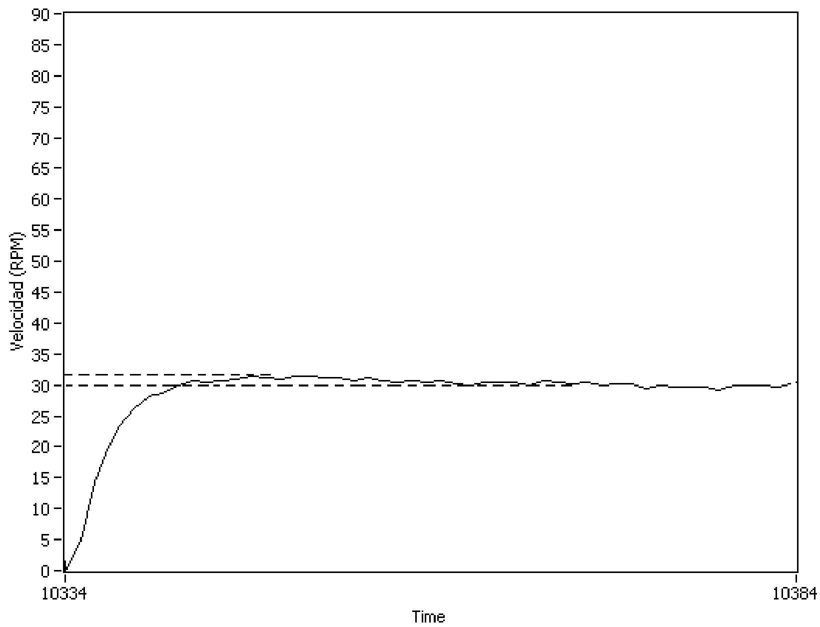


Figura 5.11 Respuesta del motor al controlador kp=2 y ki=5



5.3 Análisis de resultados

5.3.1 Análisis de operación del ciclo de trabajo y voltaje en los motores

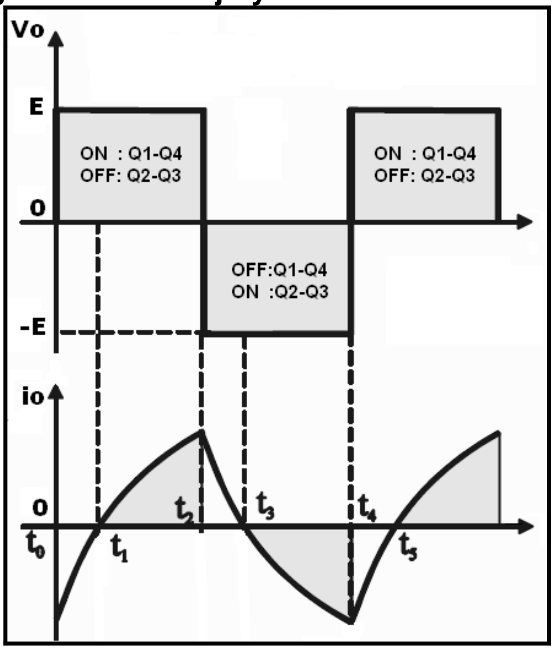
El proceso de esta prueba se explicó en la sección 5.1.1 y los resultados se muestran en las figuras 5.1 al 5.6, en las cuales se observa la curva de la señal de voltaje modulada que llega al motor. Se puede observar cuando el motor recibe un ciclo de trabajo del 50% este se puede considerar detenido con un voltaje promedio de 370mV (ver figura 5.1 y 5.4), también se observó que el giro del motor esta determinado por el porcentaje de ciclo de trabajo, si se tiene un porcentaje mayor al 50% se obtendrá un giro horario con un voltaje promedio positivo, mientras que para un porcentaje menor al 50% se tendrá un giro antihorario con un voltaje promedio negativo.

5.3.2 Análisis de comportamiento de la corriente en los motores

El proceso de esta prueba se explicó en la sección 5.1.2 y los resultados se muestran en las figuras 5.7 y 5.8, estas figuras muestran las curvas de las señales de voltaje en el canal 1 del osciloscopio, mientras que el canal 2 se captura la señal de corriente.

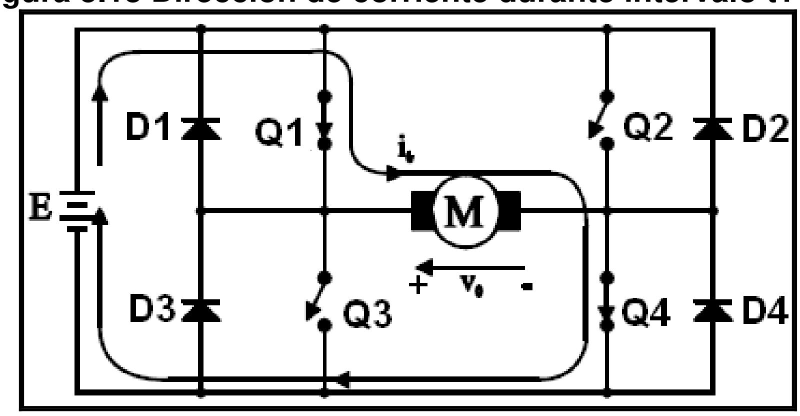
Durante esta prueba se verificó el comportamiento de la corriente en el puente manejador de motores, cuya respuesta se divide en cuatro etapas como se muestra en la figura 5.12, durante el intervalo de tiempo t_1 a t_2 los transistores Q1-Q4 están encendidos y la corriente fluye a través de ellos creciendo positivamente hasta llegar a su máximo valor (figura 5.13), en ese instante de tiempo t_2 los transistores Q1-Q4 se apagan y se encienden Q2-Q3, consecuentemente la corriente decrecerá y su polaridad será la misma fluyendo a través de los diodos D2-D3 hasta llegar a cero en el tiempo t_3 (figura 5.14), luego en el intervalo de t_3-t_4 los transistores Q2-Q3 continúan encendidos pero la corriente que fluye a través de ellos crece negativamente con polaridad contraria los anteriores intervalos (t_1-t_2 y t_2-t_3) hasta llegar a su máximo valor en el tiempo t_4 (figura 5.15), en este instante los transistores Q2-Q3 se apagan y se encienden Q1-Q4 subsecuentemente la corriente decrecerá y su polaridad será la misma fluyendo a través de los diodos D1-D4 hasta llegar a cero en el tiempo t_5 (figura 5.16), estas etapas se repiten sucesivamente desde t_1 a t_5 obteniéndose la curvas mostradas en la sección 5.2.2 .

Figura 5.12 Voltaje y corriente en el motor



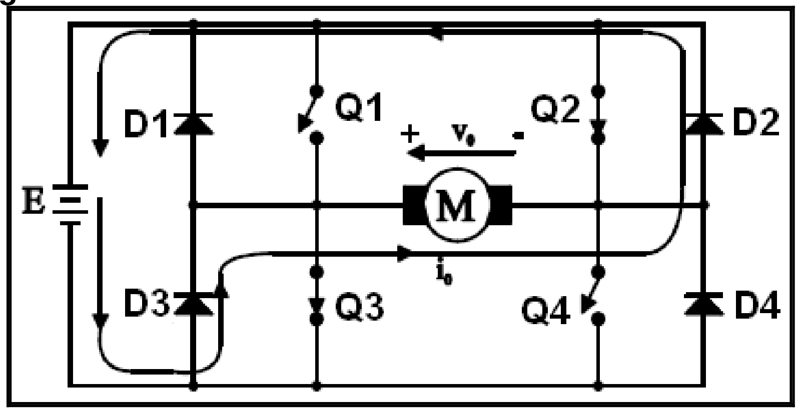
FUENTE: *Personal*

Figura 5.13 Dirección de corriente durante intervalo t1-t2



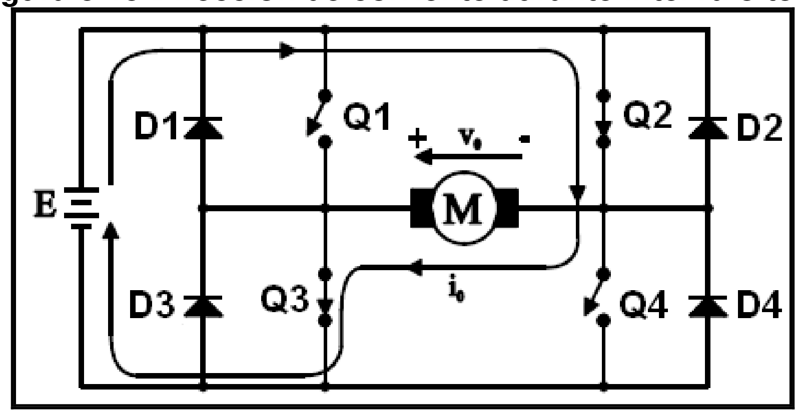
FUENTE: *Personal*

Figura 5.14 Dirección de corriente durante intervalo t2-t3



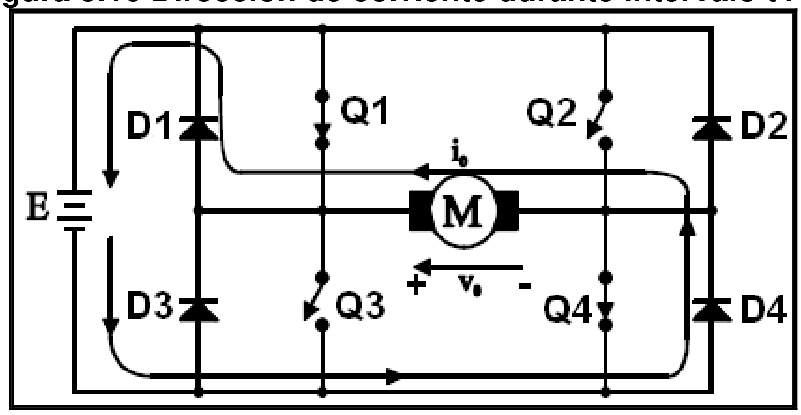
FUENTE: *Personal*

Figura 5.15 Dirección de corriente durante intervalo t3-t4



FUENTE: *Personal*

Figura 5.16 Dirección de corriente durante intervalo t4-t5



FUENTE: Personal

5.3.3 Análisis operación del motor DC usando controlador PI

Los resultados de esta prueba se muestran en la sección 5.2.3, estas figuras muestran como la respuesta del motor depende del tipo de controlador que se halla seleccionado, es decir mucho influirán las constantes elegidas de tal manera que el controlador PI se volverá más proporcional o más integral según los valores que registren estas constantes, para el caso de la figura 5.9, las constantes seleccionadas le atribuyen al sistema una respuesta poco adecuada para el correcto funcionamiento del motor presentado crestas y valles demasiados pronunciados para alcanzar la estabilidad.

En la figura 5.10 se presenta un controlador totalmente proporcional este tiene una constante de proporcionalidad igual a diez, aunque para esta respuesta las oscilaciones disminuyen y la estabilidad se produjo en un periodo más corto, aún se produce una sobreoscilación que dependerá de la constante de proporcionalidad seleccionada, este modo de control esta basado en la multiplicación del error producido por la diferencia del valor medido y esperado de velocidad, con la constante de proporcionalidad, normalmente este error es permanente pero con este tipo control ese error se mantiene en rangos tolerables, se debe procurar escoger valores de la constante que no produzcan sobreoscilaciones que generen niveles mayores al 30% del valor de velocidad esperado.

La figura 5.11, muestra la respuesta del motor de corriente continua para un control PI, en donde las constantes de proporcionalidad e integración toman los valores de dos y cinco respectivamente, ocasionando que el error en estado estacionario producido por el control proporcional disminuya hasta considerarlo despreciable, esto se debe a que este error es integrado, este resultado es multiplicado por la constante de integración para que posteriormente la respuesta integral se sume con la proporcional, originando una curva con error de estado estacionario despreciable como se muestra en la figura mencionada.

CONCLUSIONES Y

RECOMENDACIONES

Conclusiones

El desarrollo de este proyecto se ha aportado con una solución de diseño y análisis de producción de tarjetas de entrenamiento para dsPIC, haciendo conocer que aquí en Ecuador se puede diseñar equipos que superen los requerimientos de las tarjetas de entrenamiento ofrecidos en el exterior y resultando más económicos que en el extranjero, comprobándose que se puede hacer ingeniería en el país.

El dsPIC30F4011 utilizado como unidad de control destaca la capacidad de manejar un encoder incremental, la posibilidad de realizar modulación por ancho de pulso para motores, y la facilidad para la configuración de la frecuencia de operación de un motor, respecto a mikroBasic constituye la herramienta de desarrollo con un lenguaje que permite acceder de manera amigable a los recursos que ofrece el controlador, por lo anteriormente expuesto estas herramientas ofrecen una solución de altas prestaciones y facilidad de programación.

El desarrollo de este proyecto ha logrado concebir una tarjeta de entrenamiento en donde se aprovechó la mayoría de los recursos que ofrece el dsPIC30F4011, para lograr este objetivo se utilizó puentes de habilitación para acceder a los diferentes dispositivos implementados en uno o varios pines logrando optimizar la implementación de hardware sobre la tarjeta.

La interfaz LabVIEW permitió realizar de manera eficaz la interacción con las tarjetas implementadas pudiendo cambiar en línea los parámetros del controlador, de esta manera se logró realizar un sistema de control de lazo cerrado usando un controlador PI.

La utilización de controladores en PI, depende mucho de las características del sistema que a será controlado, esto exige entender correctamente el funcionamiento del sistema. Como se verificó en las pruebas realizadas un control completamente proporcional nos garantiza estabilidad sin embargo produce un error de estado estacionario originando un sobrenivel, mientras que al agregar un control integral este error se corrige hasta alcanzar la estabilidad.

RECOMENDACIONES

Para incrementar la vida útil de la tarjeta de entrenamiento se debe procurar no exceder las limitaciones máximas de corriente en las pistas de control y fuerza como se especificó en las tablas 2.5 y 2.6, si opera con valores superiores por un largo tiempo no se garantiza la correcta funcionalidad del sistema o podría causar daños irreversibles, por consiguiente se debe verificar las características eléctricas del motor que será conectado en las borneras de la tarjeta.

Motivar el desarrollo de proyectos que integren nuevas tendencias tecnológicas para encontrar soluciones a las mismas, como es el caso de este proyecto que involucra un dsPIC de altas prestaciones y facilidad de programación como unidad de control.

Se recomienda el uso de LabVIEW como una herramienta que permita interactuar con la tarjeta de entrenamiento a manera de tarjeta de adquisición de datos, para la implementación de un sistema SCADA.

ANEXOS

ANEXO A

Procedimiento para carga de programa de usuario a través de bootloader

Para el desarrollo de ejemplos básicos podría usarse mikroBasic y en caso de no disponer de un programador se puede hacer uso de la interfaz para bootloader de Ingenia, permitiendo de esta manera la utilización de los periféricos implementados la tarjeta de entrenamiento con dsPIC30F4011, a continuación se explica es procedimiento para la carga de un programa de usuario:

Interfaz de Ingenia bootloader

El paquete de Ingenia para el bootloader esta conformado por un firmware de código abierto y una aplicación de interfaz gráfica para el usuario. Una vez instalado la interfaz gráfica en el computador, podrá arrancase esta aplicación la cual requerirá la configuración del puerto de comunicación donde se conectara el dsPIC y la velocidad de transferencia de datos entre el dsPIC y el puerto seleccionado.

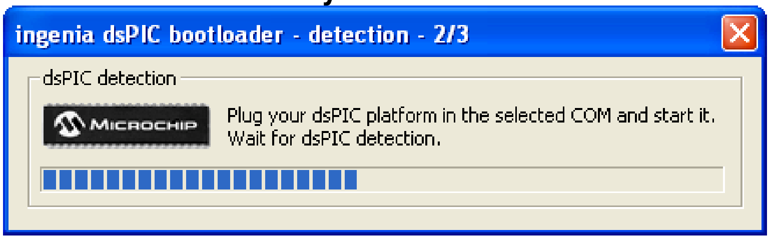
Configuración del puerto de comunicación

Una vez configurados los parámetros para la comunicación, la aplicación solicitará que se apague la alimentación de la tarjeta de entrenamiento con la finalidad de sincronizar el bootloader grabado en el dsPIC y la aplicación de interfaz gráfica para el usuario.

Arrancando la interfase gráfica de Ingenia dsPIC

Por ultimo la aplicación comenzará con el proceso de detección del dsPIC30F4011.

Detección del dsPIC y del firmware del bootloader



Iniciada la detección la aplicación solicitará la conexión del dsPIC a los puertos de comunicación seleccionados, una vez realizada exitosamente la sincronización y detección del dsPIC se mostrará un mensaje con el tipo de dsPIC detectado y el firmware del bootloader que se encuentra grabado en el dispositivo.

Posteriormente a la detección la aplicación mostrará la ventana de dialogo que permitirá cargar un nuevo programa asociando la selección con su correspondiente fichero HEX, quedando disponible para iniciar la escritura del programa sobre las primeras posiciones de memoria flash del dsPIC30F4011.

Carga del programa mediante bootloader

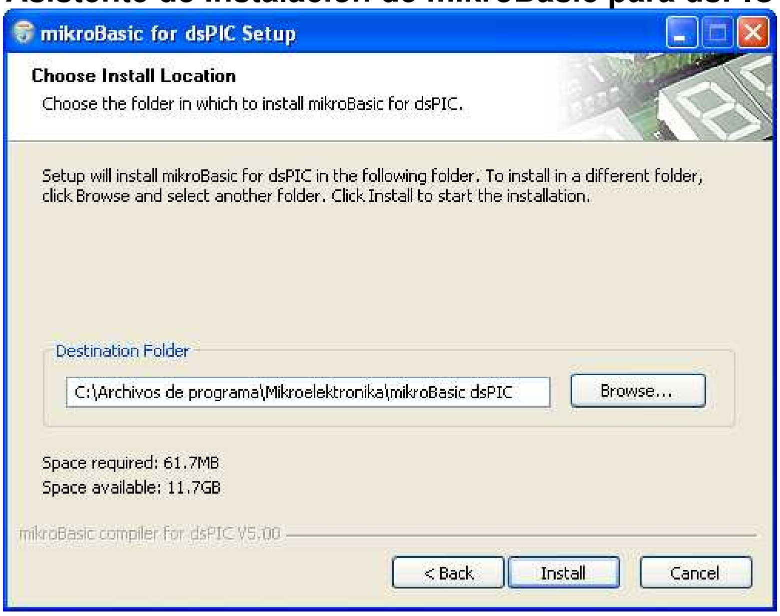
ANEXO B

Instalación y creación de proyectos con mikroBasic

Compilador para dsPIC30/33

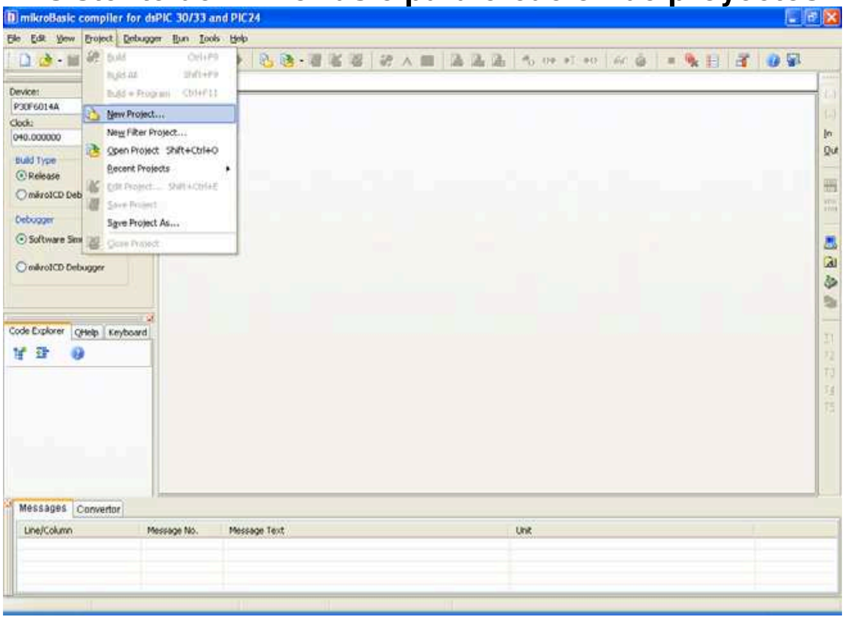
La instalación del programa de desarrollo se realizará con la ayuda de su asistente, la siguiente figura muestra la ventana de selección del directorio que por defecto establece, para posteriormente iniciar este proceso.

Asistente de instalación de mikroBasic para dsPIC



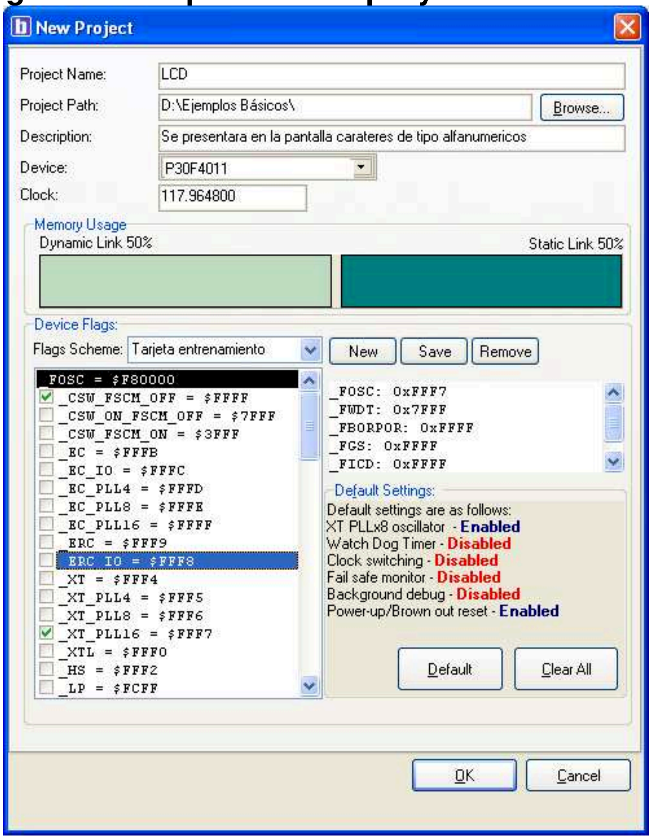
Luego de la instalación de mikroBasic se podrá desarrollar aplicaciones para el manejo de los periféricos disponibles en la tarjeta de entrenamiento. Para comenzar la descripción de un nuevo programa se debe crear un proyecto, para ello se utilizará el asistente disponible para este propósito.

Asistente de mikroBasic para creación de proyectos



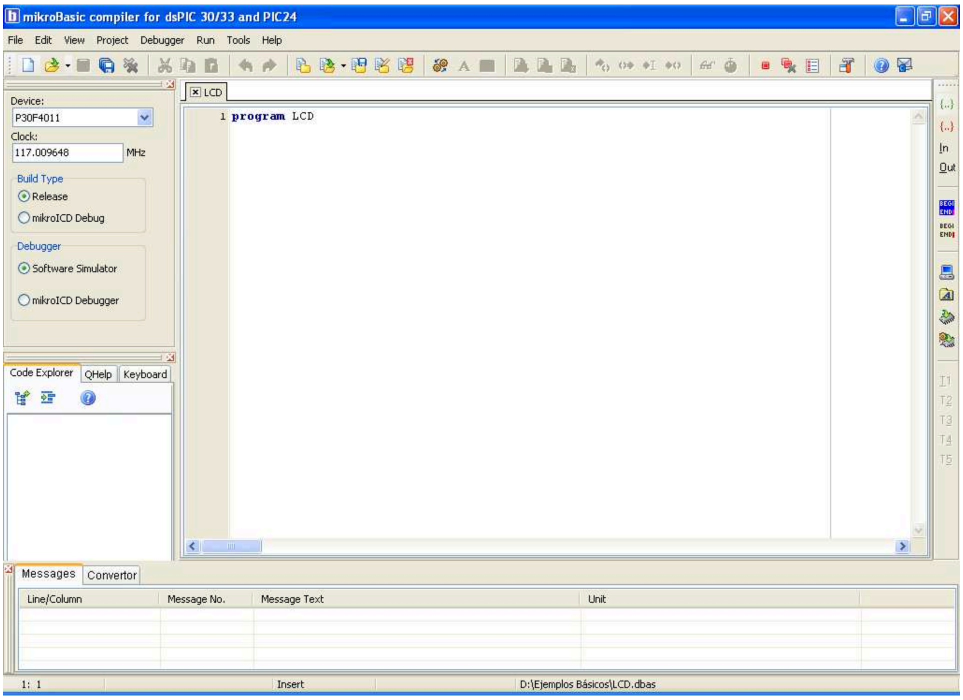
El asistente para creación de proyectos ayudará a seleccionar el dsPIC utilizado además de la configuración de los parámetros de funcionamiento del dispositivo seleccionado, se debe de considerar que la frecuencia de funcionamiento del dsPIC30f4011 será de 117.9648 MHz, puesto que se dispone de un cristal de 7.3728 MHz configurado con un multiplicador de 16 en las tarjeta de entrenamiento.

Configuraciones para nuevo proyecto en mikroBasic



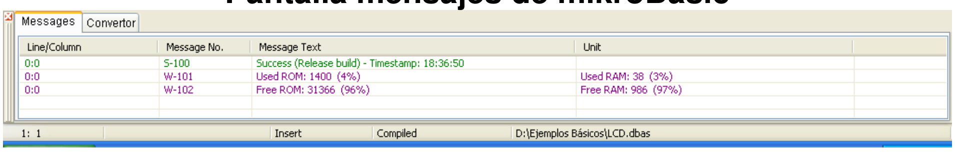
Luego de realizar las configuraciones del proyecto inmediatamente aparecerá el editor de texto, la primera instrucción del programa es ubicada automáticamente esta incluye el nombre del proyecto, seguidamente después de esta instrucción se procederá a realizar la descripción del programa a ser utilizado en la tarjeta de entrenamiento, esta programación esta basada en lenguaje Basic, tanto su entorno y compilador son desarrollados por Mikroelectrónica.

Pantalla del editor de texto de mikroBasic



Una vez desarrollado el programa se deberá ejecutar la compilación del mismo para ello se deberá desplegar el menú proyecto y pulsar la opción compilar todo, si no existen errores en la ventana de mensajes aparecerá como se muestra en la siguiente figura, en caso contrario en la misma ventana se enlistara los tipos de errores cometidos

Pantalla mensajes de mikroBasic



ANEXO C

Ecuaciones y parámetros necesarios para dimensionar el ancho de pista de cobre

Una vez seleccionados los parámetros de corriente máxima, el incremento de temperatura sobre la temperatura ambiente, y el grosor de la pista, se procederá al cálculo del ancho de la pista según las siguientes ecuaciones de acuerdo a la norma ANSI-ICP2221, previamente se requiere el dato del área:

$$Area = \left(\frac{I}{\left(K_1 * \Delta T^{K2} \right)} \right)^{1/K3} = \left[mils^2 \right]$$

donde:

I simboliza la corriente máxima que circulara por la pista.

K1 es una constante definida por el estándar aplicado que vale 0,0150 cuando nuestra pista es interna (placas con más de dos capas) y 0,0647 cuando es externa.

K2 es otra constante que vale 0,5453 cuando la pista es interna y 0,4281 cuando la pista es externa.

K3 es una constante que vale 0,7349 cuando la pista es interna y 0,6732 cuando es externa.

El ancho de pista resultante se expresa en mils al cuadrado. Un mils es la milésima parte de una pulgada, que es la unidad estandarizada en el diseño de placas de circuitos impreso. El área debe expresarse en mils al cuadrado y el grosor (L) en onzas por píe cuadrado.

$$Ancho = \frac{Area}{L * 1{,}378} = \frac{\left[mils^2 \right]}{\left[onz/pie^2 \right]} = \left[mils \right]$$

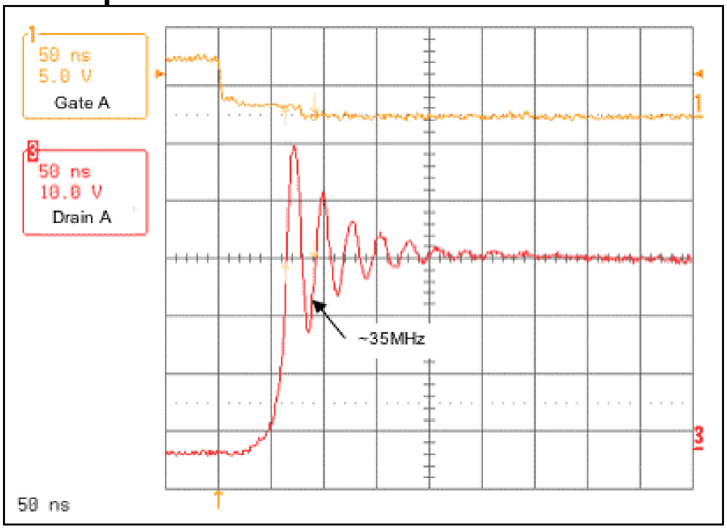
Las tablas 2.5 y 2.6 del capítulo 2 resumen el resultado de los parámetros de funcionamiento y el ancho de pistas para las señales de control y de fuerza en las tarjetas de circuito impreso implementadas en las tarjetas de entrenamiento.

ANEXO D

Procedimiento para dimensionar la red snubber

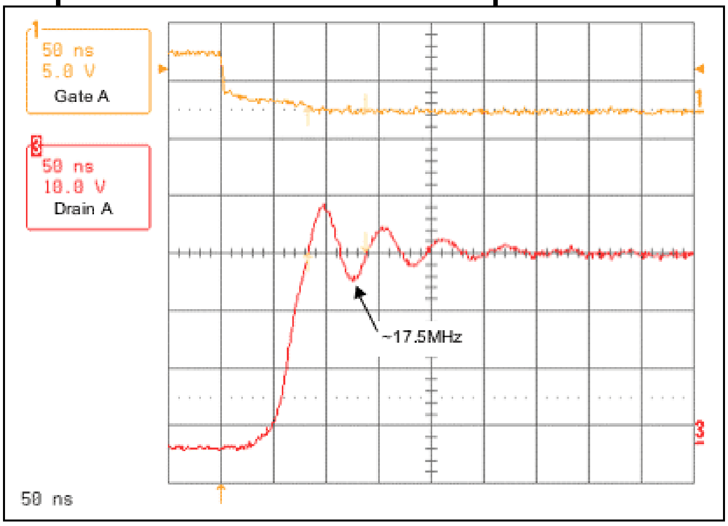
Para un diseño óptimo de la red snubber se debe realizar el análisis del circuito en corriente alterna y determinar los valores de capacitancia (Ci) e inductancia (Li) intrínsecos del circuito, dichos parámetros se obtendrán del análisis experimental, y haciendo uso de las ecuaciones que describen la respuesta de oscilación durante la conmutación de los mosfet. Para obtener la capacitancia intrínseca primero se requirió conocer frecuencia inicial de oscilación del sistema, este parámetro será obtenido de manera experimental haciendo uso del osciloscopio, en la siguiente gráfica se muestra la forma de onda obtenida:

Respuesta de resonancia sin red Snubber



De la onda anterior se requiere que la frecuencia de oscilación se reduzca en un factor de dos, para lo cual se utilizó un capacitor de mica cuyo valor inicial fue de 100pF conectado entre drenador y surtidor, el valor final de capacitancia que logró reducir la frecuencia de oscilación a la mitad fue 330 pF que representara la capacitancia parasita total (ver figura 2.5).

Respuesta de resonancia con capacitor snubber



La capacitancia intrínseca del mosfet será 1/3 de la capacitancia parasita total, la inductancia intrínseca se calculará utilizando la ecuación de frecuencia de resonancia ω0:

$$\omega_0 = 2\pi f_i = \frac{1}{\sqrt{L_i C_i}}$$

Usando la anterior ecuación obtenemos el valor de inductancia intrínseca del mosfet:

$$L_i = \frac{1}{C_i\left(2\pi f_i\right)^2} = \frac{1}{110\mathrm{pF}*\left(2\pi*35\mathrm{MHz}\right)^2} = 0.1879\,uH$$

Calculamos la impedancia característica del sistema:

$$Z = \sqrt{\frac{L_i}{C_i}} = \sqrt{\frac{0.1879\ uH}{110\ pF}} = 41.34\ \mathrm{ohms}$$

Se debe de elegir un valor de resistencia snubber que no sea mayor al valor de la impedancia característica del sistema para que la mayor parte de la corriente circule por la red snubber durante la conmutación y el valor de capacitancia snubber será un factor de cuatro a diez veces la capacitancia intrínseca. A partir de las consideraciones mencionadas se procedió a elegir un valor de resistencia de 39 ohmios y una capacitancia de 1000pF, como parámetros de la red snubber. Una vez dimensionados los elementos de la red snubber se calculó la potencia máxima disipada por la misma mostrándose a continuación:

$$P_{Rs} = C_S V_0^{\ 2}\ f_S = 1000\ pF*\left(100\ V\right)^2 * 50\ kHz = 0.5\,W$$

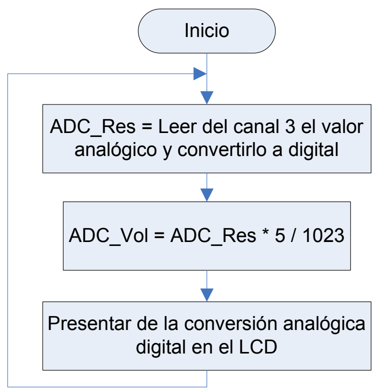
La potencia de la red snubber fue calculada para una velocidad de conmutación del mosfet de 50kHz y voltaje pico máximo de 100V.

ANEXO E

Diagramas de flujo de las aplicaciones

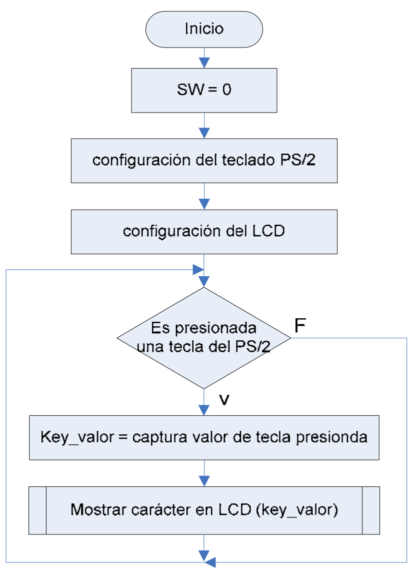
E.1 Manejo de Conversión ADC

Programa principal

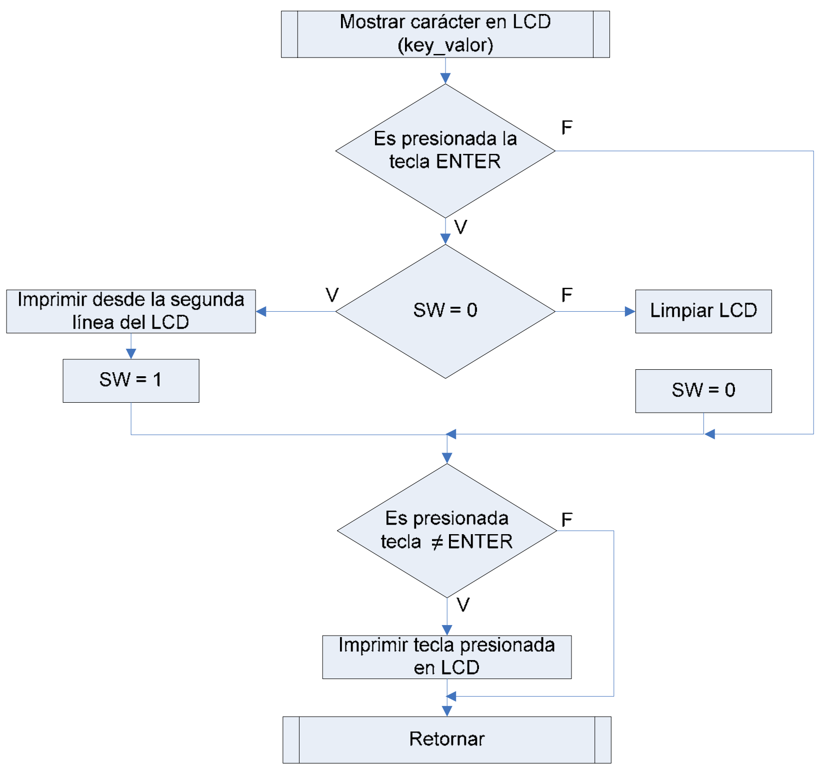


E.2 Manejo de Teclado PS2_LCD

Programa principal

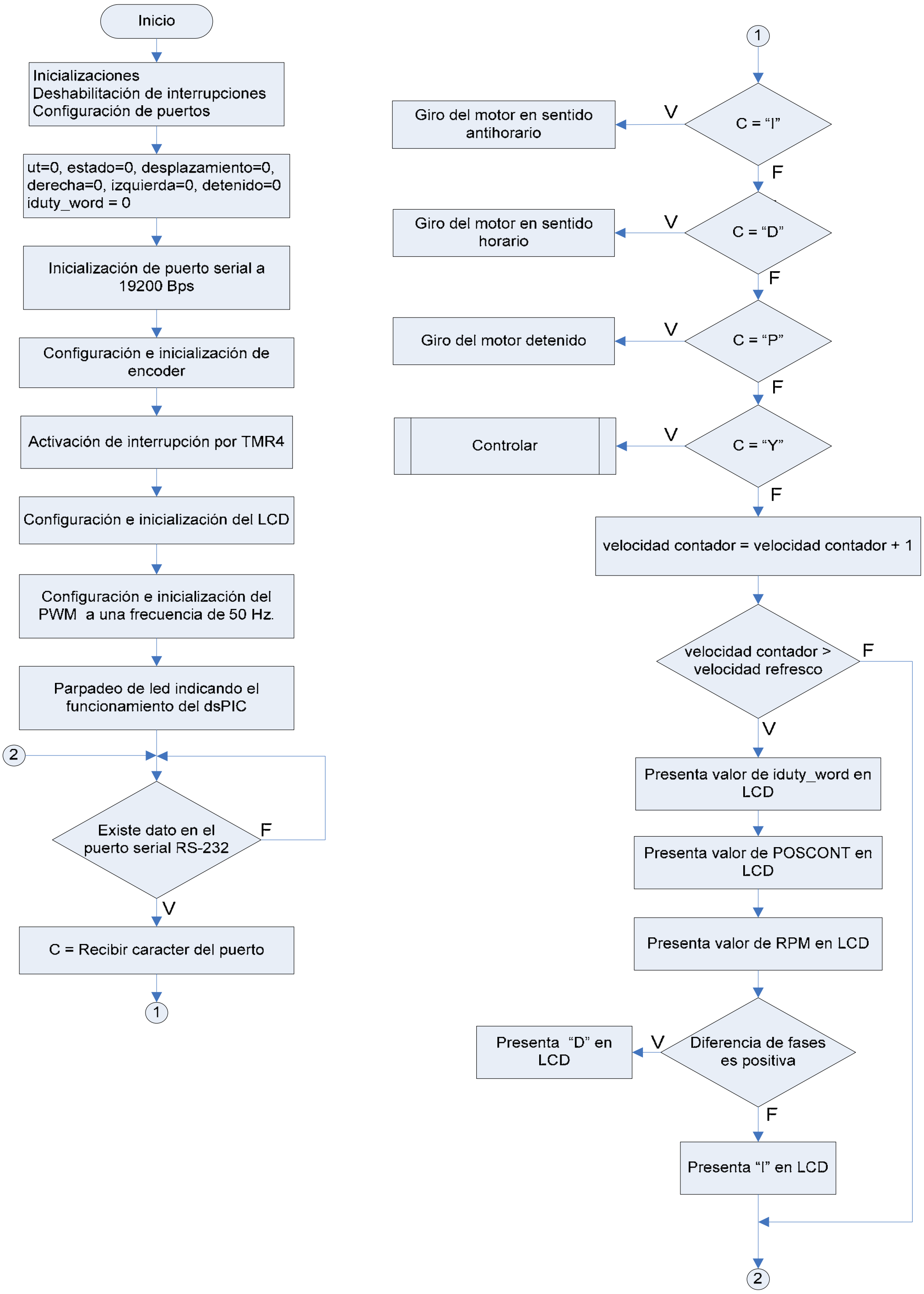


Subrutina

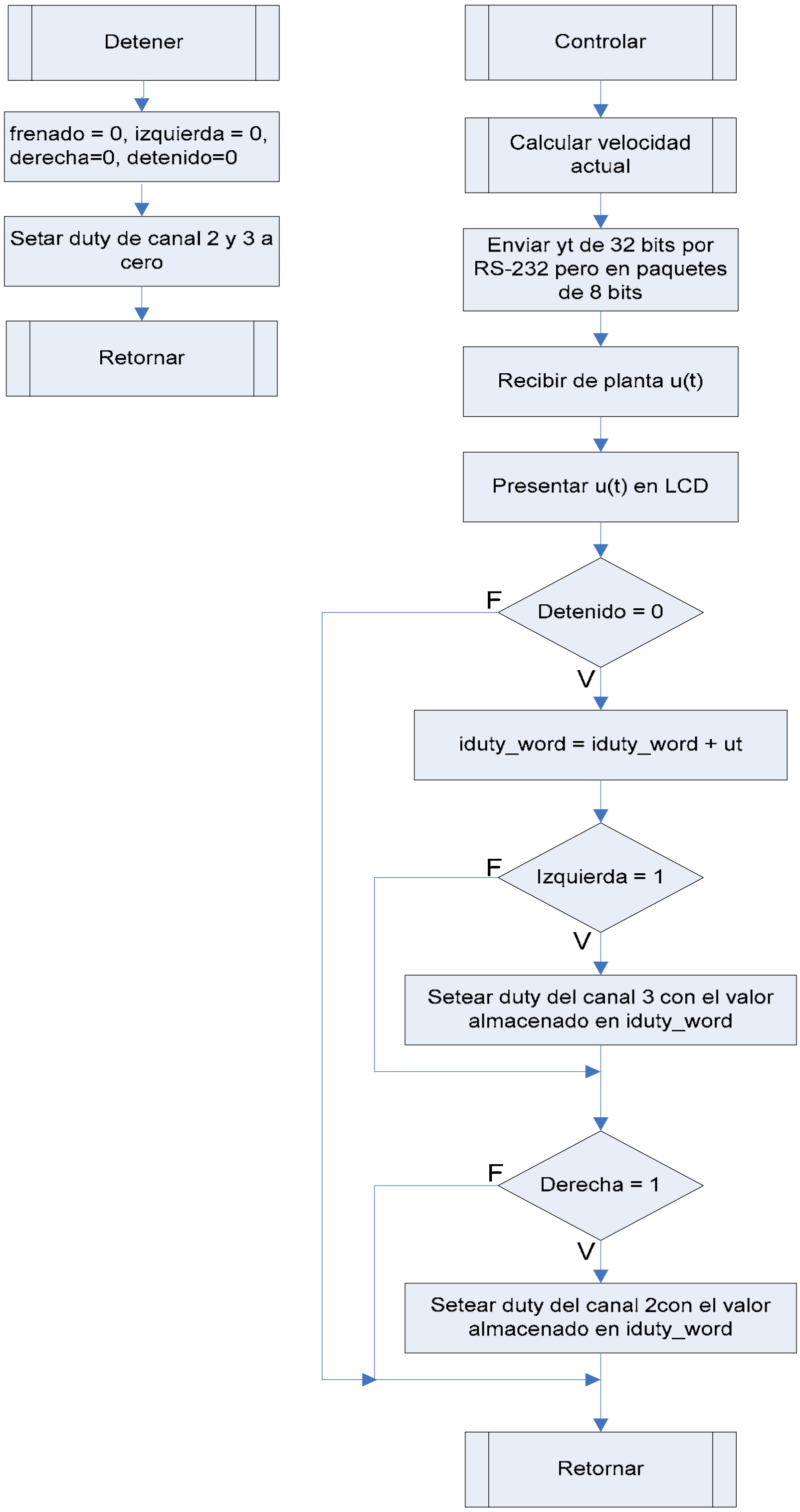

Mostrar carácter en LCD (key_valor)
Es presionada la tecla ENTER
F
V
SW = 0
V
F
Imprimir desde la segunda línea del LCD
Limpiar LCD
SW = 1
SW = 0
Es presionada tecla ≠ ENTER
F
V
Imprimir tecla presionada en LCD
Retornar

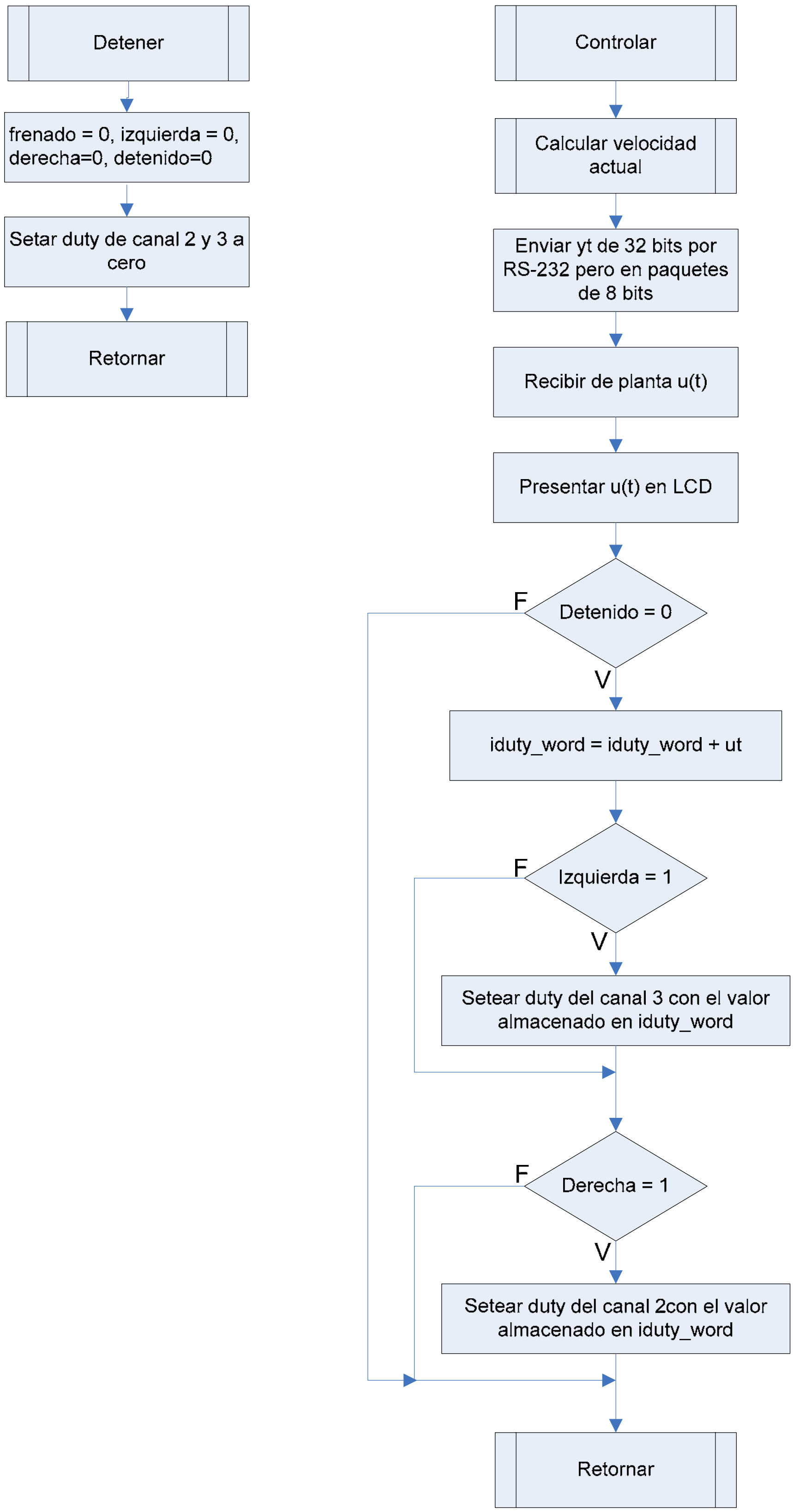
E.3 Control motor DC con dsPIC30F4011 y LabVIEW

Programa principal del dsPIC30F4011

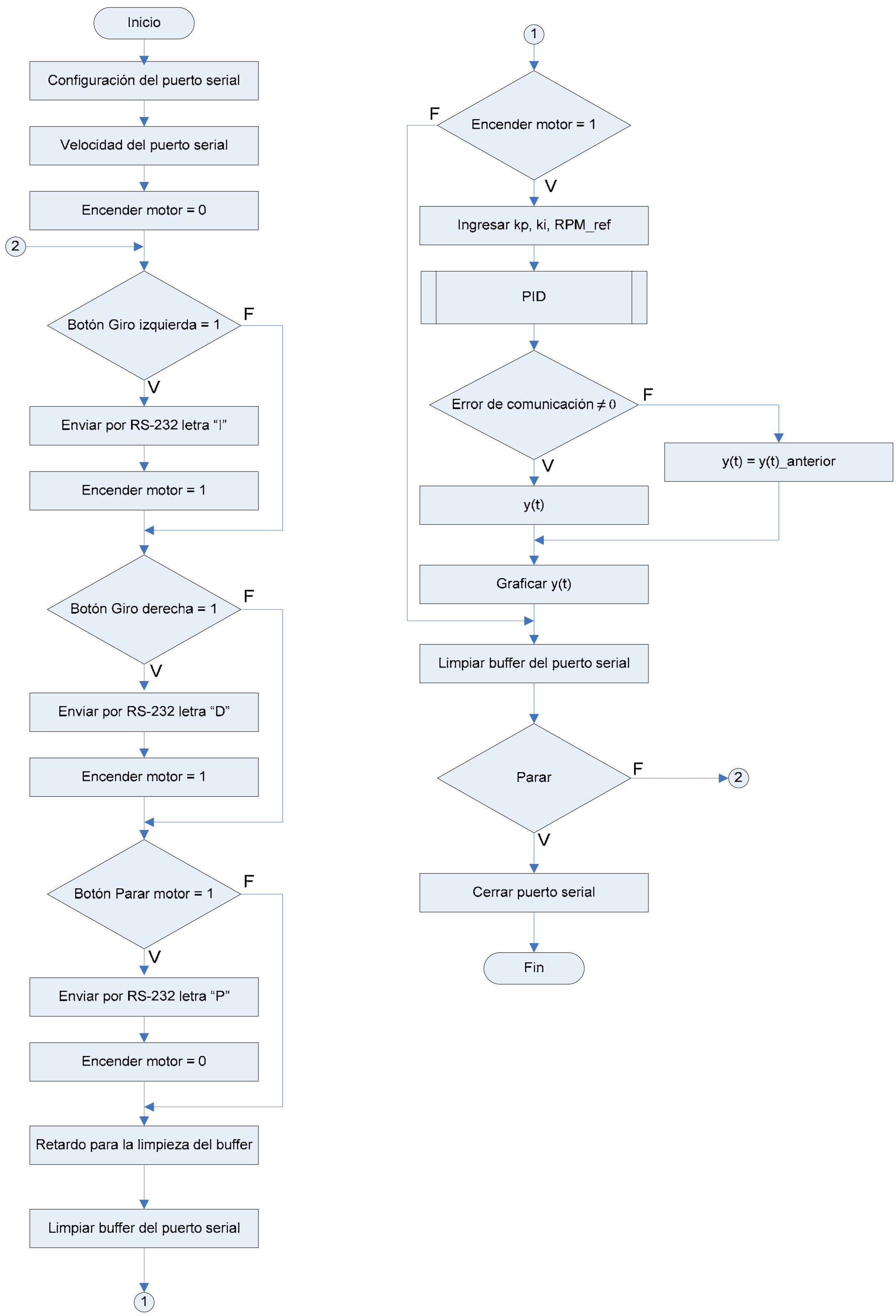


Subrutinas utilizadas en el dsPIC30F4011

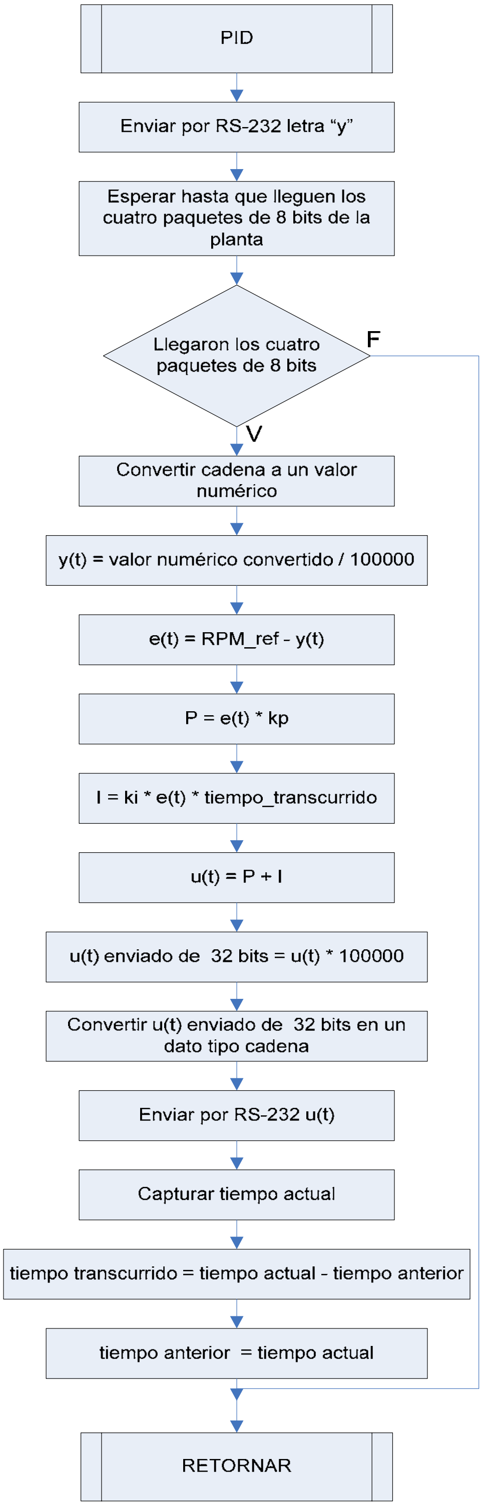



Detener
frenado = 0, izquierda = 0, derecha=0, detenido=0
Setar duty de canal 2 y 3 a cero
Retornar
Controlar
Calcular velocidad actual
Enviar yt de 32 bits por RS-232 pero en paquetes de 8 bits
Recibir de planta u(t)
Presentar u(t) en LCD
Detenido = 0
F
V
iduty_word = iduty_word + ut
Izquierda = 1
F
V
Setear duty del canal 3 con el valor almacenado en iduty_word
Derecha = 1
F
V
Setear duty del canal 2con el valor almacenado en iduty_word
Retornar

Programa principal de la interfaz LabVIEW

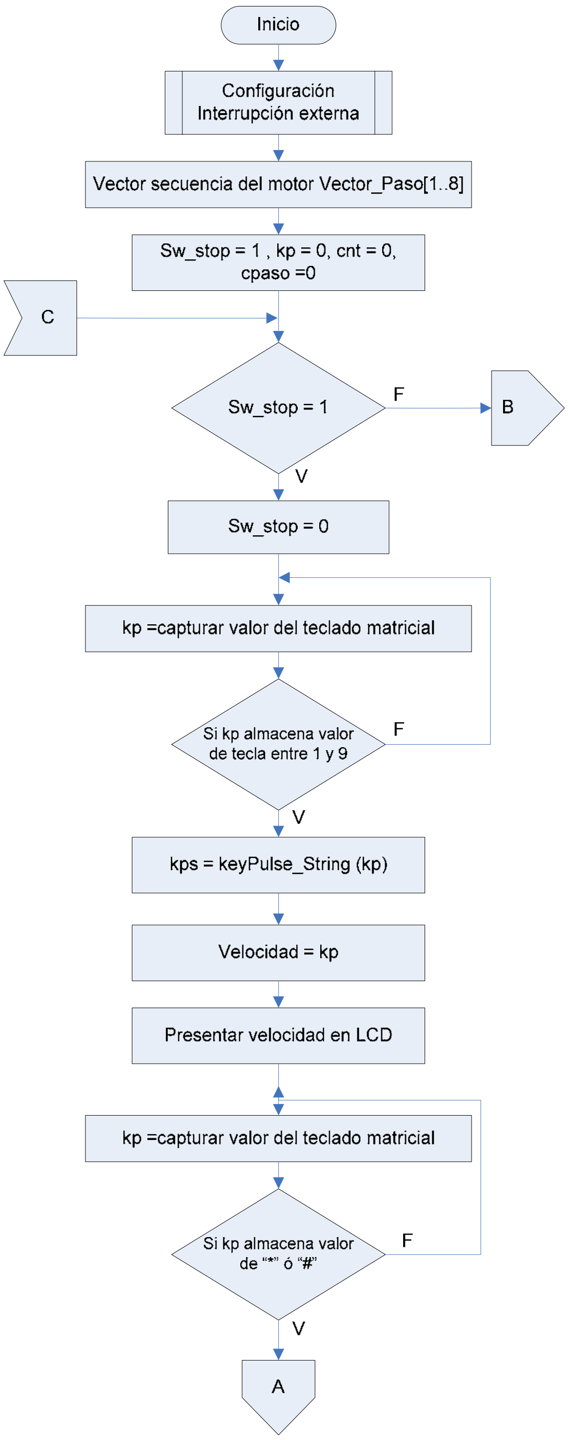


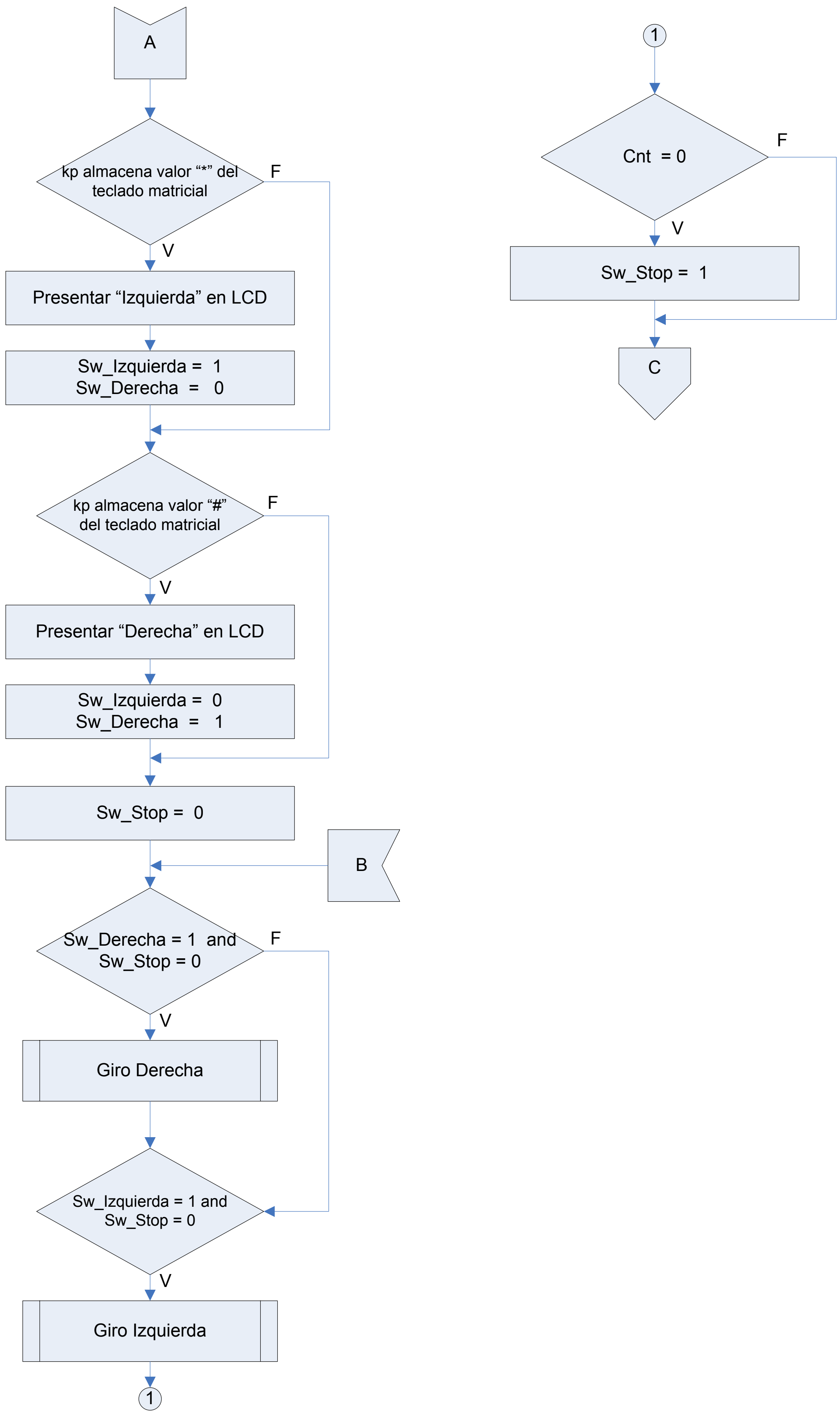
Subrutinas utilizadas en la interfaz LabVIEW



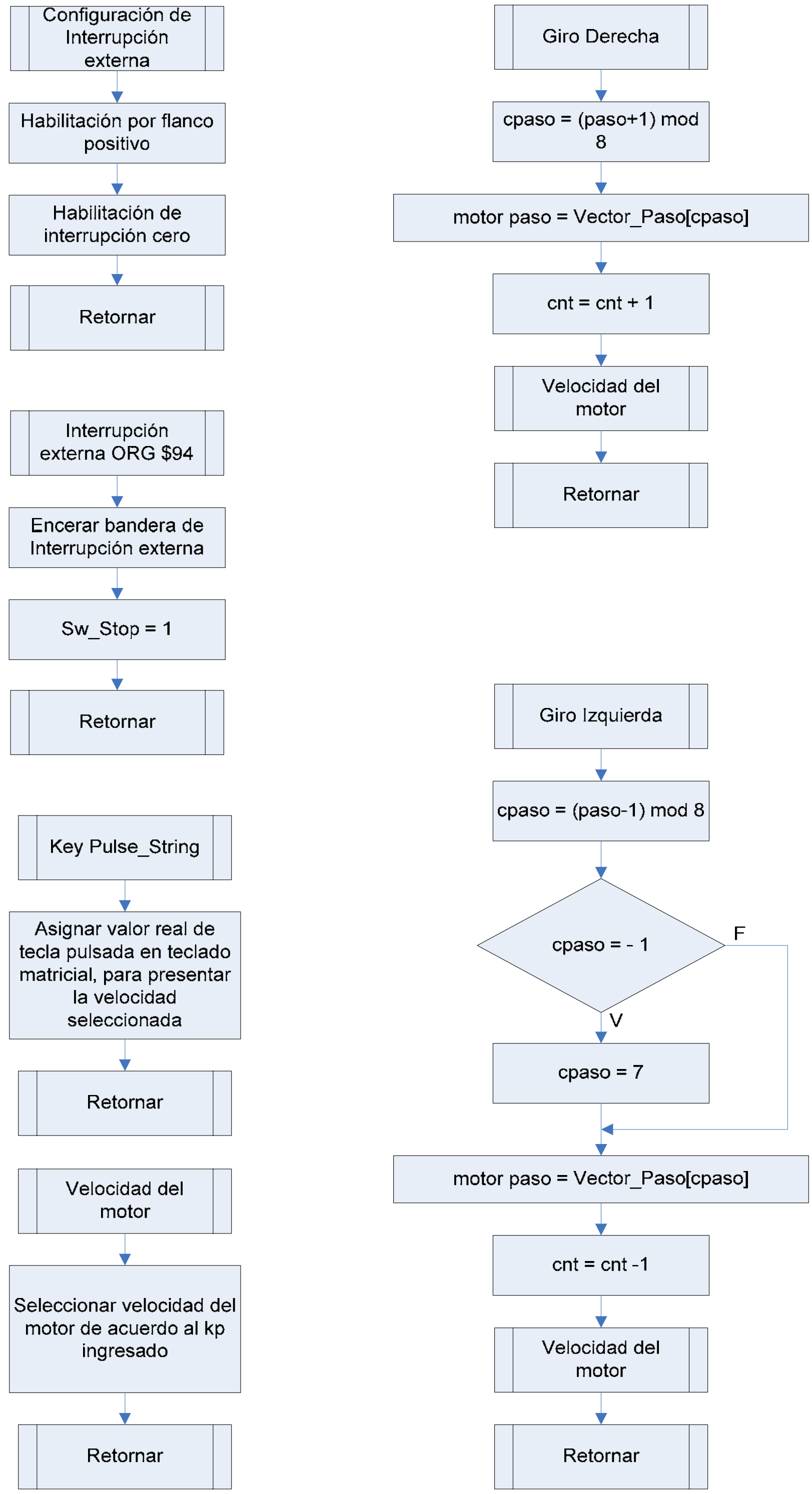
E.4 Manejo de motor de pasos unipolar

Programa principal



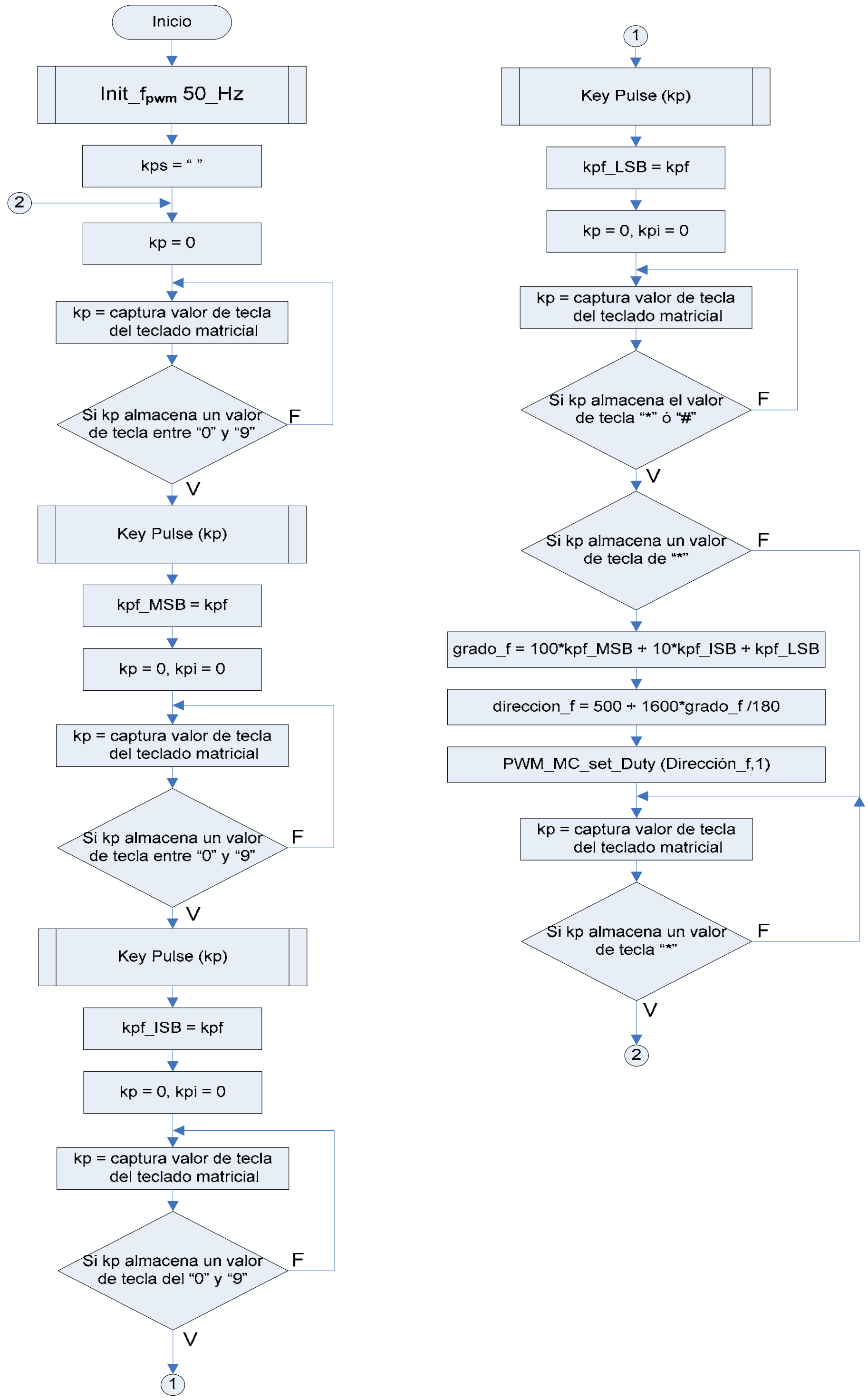

A
kp almacena valor “*” del teclado matricial
F
V
Presentar “Izquierda” en LCD
Sw_Izquierda = 1
Sw_Derecha = 0
kp almacena valor “#” del teclado matricial
F
V
Presentar “Derecha” en LCD
Sw_Izquierda = 0
Sw_Derecha = 1
Sw_Stop = 0
B
Sw_Derecha = 1 and Sw_Stop = 0
F
V
Giro Derecha
Sw_Izquierda = 1 and Sw_Stop = 0
V
Giro Izquierda
1
1
Cnt = 0
F
V
Sw_Stop = 1
C

Subrutinas

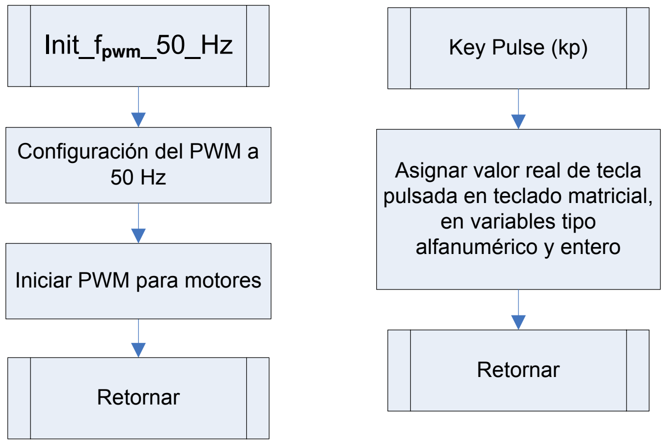

Configuración de Interrupción externa
Habilitación por flanco positivo
Habilitación de interrupción cero
Retornar
Interrupción externa ORG $94
Encerar bandera de Interrupción externa
Sw_Stop = 1
Retornar
Key Pulse_String
Asignar valor real de tecla pulsada en teclado matricial, para presentar la velocidad seleccionada
Retornar
Velocidad del motor
Seleccionar velocidad del motor de acuerdo al kp ingresado
Retornar
Giro Derecha
cpaso = (paso+1) mod 8
motor paso = Vector_Paso[cpaso]
cnt = cnt + 1
Velocidad del motor
Retornar
Giro Izquierda
cpaso = (paso-1) mod 8
cpaso = - 1
F
V
cpaso = 7
motor paso = Vector_Paso[cpaso]
cnt = cnt -1
Velocidad del motor
Retornar

E.5 Manejo de motor de servomotor

Programa principal



Subrutinas

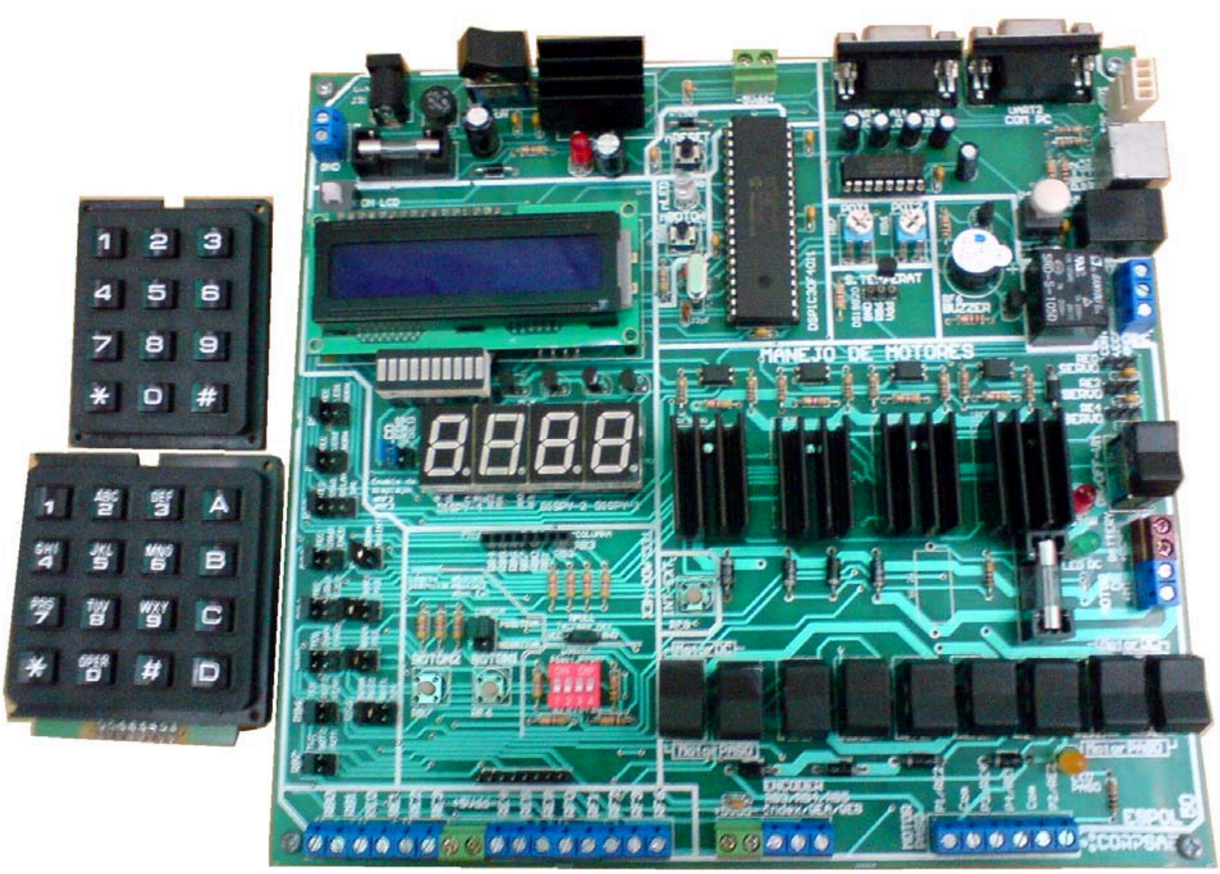


$Init\_f_{pwm}\_50\_Hz$
Configuración del PWM a 50 Hz
Iniciar PWM para motores
Retornar
Key Pulse (kp)
Asignar valor real de tecla pulsada en teclado matricial, en variables tipo alfanumérico y entero
Retornar

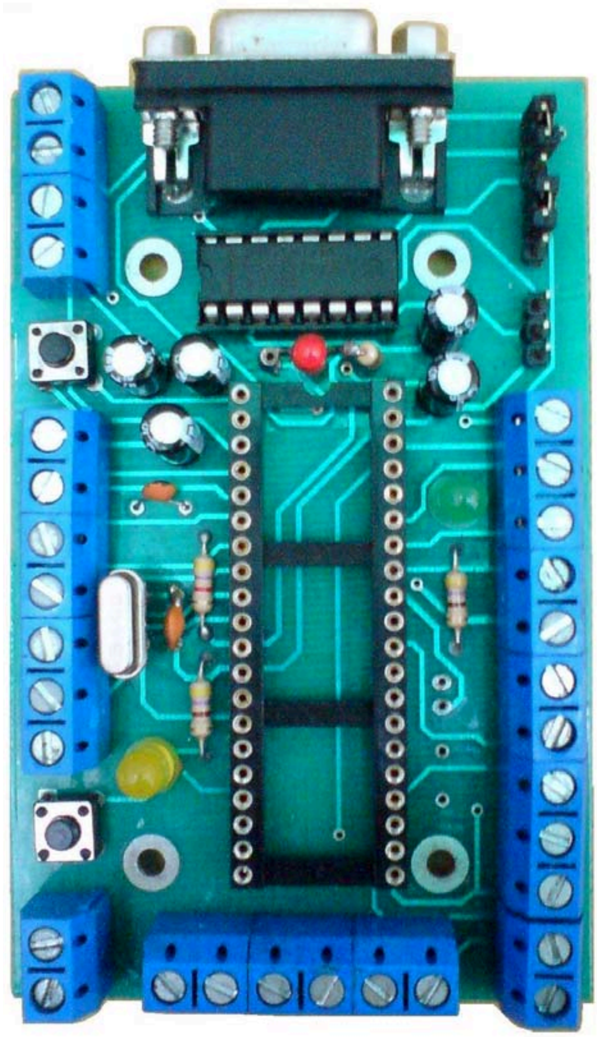
ANEXO F

Fotografías de las tarjetas de entrenamiento

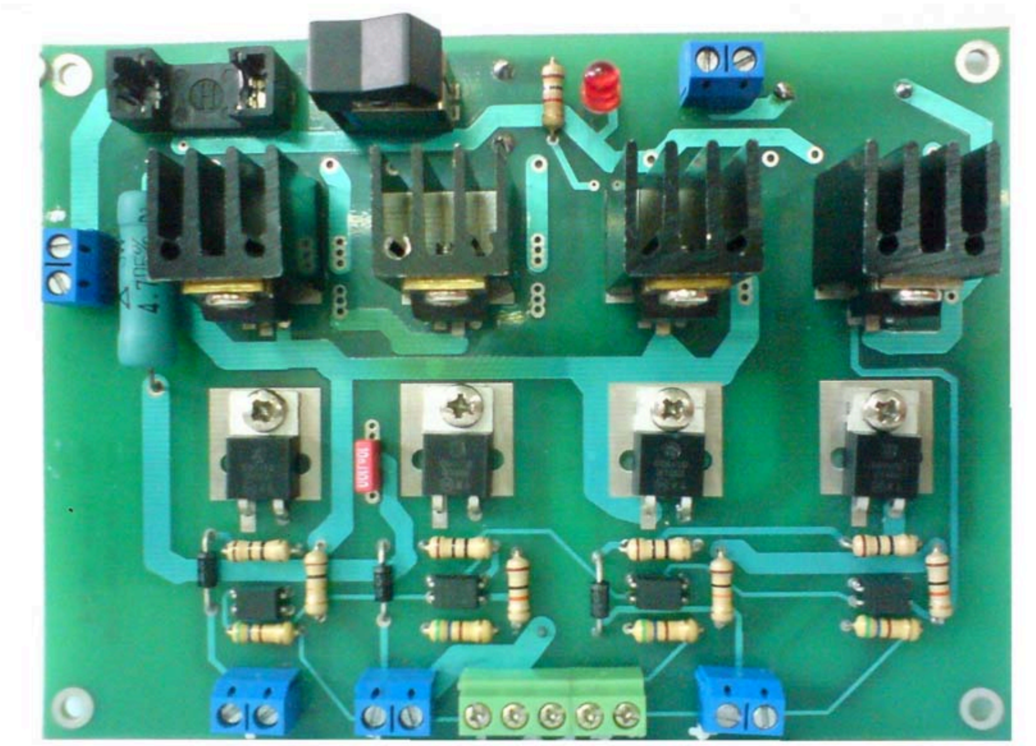
Tarjeta de entrenamiento con dsPIC30F4011



Tarjeta básica para aplicaciones con dsPIC30F4011



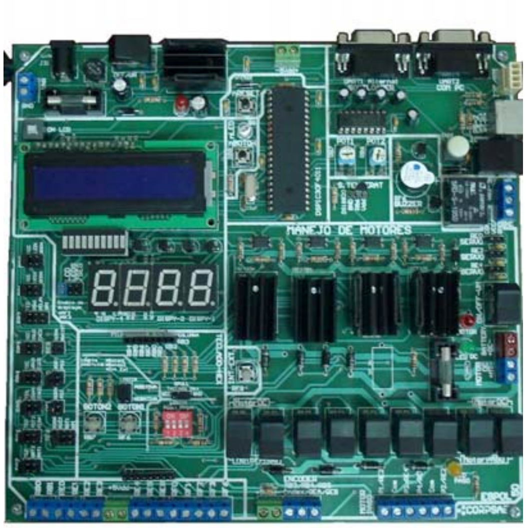
Tarjeta manejadora de motores DC

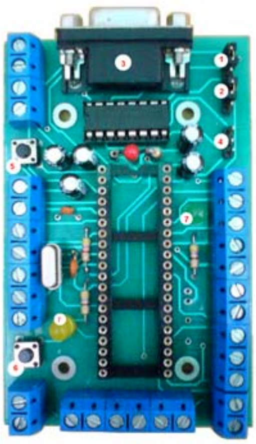


ANEXO G

Equipo de entrenamiento basado en el procesador dsPIC30F4011

MANUAL DE USUARIO V1.0





Xavier Orlando Barcia Rosero

Roger Alfredo Calle Chóez

FACULTAD DE INGENIERIA EN ELECRICIDAD Y COMPUTACIÓN

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

1.- INTRODUCCIÓN AL DOCUMENTO.

El presente documento describe las funcionalidades de las tarjetas denominadas *tarjeta de aplicaciones básicas*, *tarjeta de entrenamiento con dsPIC30F4011*, y *tarjeta manejadora de motores DC*. Estas constituyen el proyecto final de tópico "***Diseño de tarjetas para el desarrollo de aplicaciones con dsPIC***".

Se comienza con una descripción general de las posibilidades de las tarjetas y el ámbito de aplicación de las mismas para pasar a la descripción de su estructura hardware interna, los subsistemas incluidos y su interconexión.

También se explica detalladamente la forma de acceder a cada uno de los sistemas de la *tarjeta de entrenamiento con dsPIC30F4011*, entre los cuales destacan los periféricos implementados para el manejo de tres tipos de motores: servomotor de modelismo, motor de pasos, y DC. Para los dos últimos se implementó una configuración que permite utilizar los mismos elementos de manejo previa selección de uno de los motores.

2.- DESCRIPCIÓN DE LAS TARJETAS.

Las tarjetas proporcionadas están diseñadas para ser utilizadas como una plataforma para el desarrollo de diferentes aplicaciones con dsPIC. Debido a su reducido tamaño y amplia funcionalidad es especialmente útil para su utilización en Talleres o Seminarios de inicialización al diseño de controles minimizando los problemas del diseño hardware.

El sistema está basado en el Microcontrolador dsPIC30F4011 que posee 48Kbytes de memoria FLASH y 2K de RAM que puede ser programado en diversos compiladores como CCs, C30, mikrobasic entre otros. Además dispone de un potente sistema de temporización con entrada de captura, salidas de comparación y generación de PWM a través de 6 canales dedicados al control de motores, 2 UARTs, 1 módulo de bus CAN, 9 canales de ADC de 10 bits, posibilidad de comunicación por bus I2C, y otros subsistemas.

3.- TARJETA DE APLICACIONES BÁSICAS CON dsPIC30F4011.

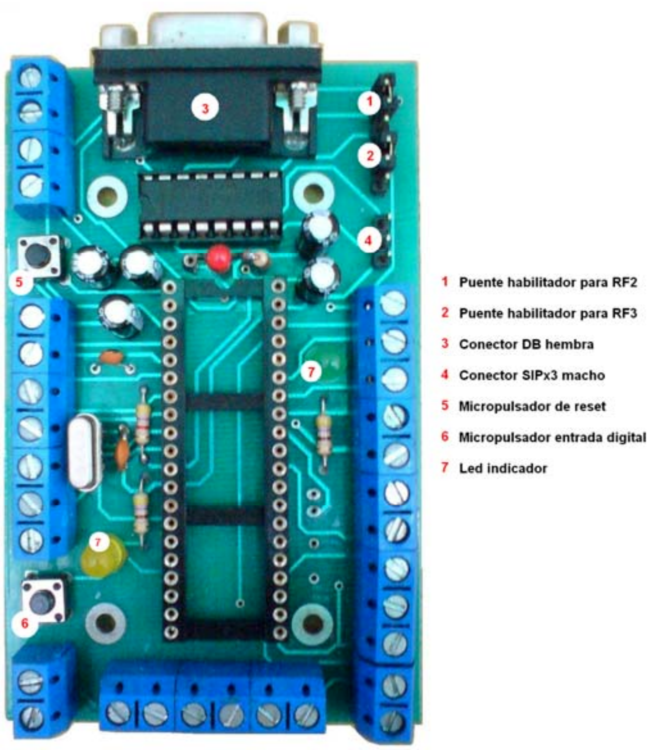
Esta tarjeta está pensada para ser usada como un sistema para el control de dispositivos. Incluye un conector DB9 hembra, y otro conector del tipo espadin. Ambos para la comunicación tipo serial a través de RS-232. Los pines no utilizados del dsPIC (I/O digitales, entradas analógicas y buses de comunicación) son accesibles por medio de borneras para expansión hardware y la conexión de otros tipos de sensores.

Las características de la tarjeta de aplicaciones básicas son las siguientes:

- Procesador dsPIC30F4011 de la familia dsPIC de Microchip.
- 1 cristal externo de 7.3728MHz.
- 2 diodos leds implementados para salida de señales digitales.
- 1 micropulsador que sirve como señal de reinicialización.
- 1 micropulsador que sirve como entrada digital.
- 1 puente de selección para la salida señal RF2 por bornera o como U2Rx por conector espadín para comunicación UART.
- 1 puente de selección para la salida señal RF3 por bornera o como U2Tx por conector espadín para comunicación UART.
- 2 UART con adaptador de niveles para RS232.
- 1 conector DB9 de comunicación vía UART1.
- 1 conector espadín de comunicación vía UART2.
- 2 borneras disponibles para conectar la fuente de alimentación del sistema.
- 12 borneras azules para la expansión hardware*.
- Capacidad de soportar 1A en las pistas.
- Tamaño 60x96 mm.

* En la tarjeta del Kit estos componentes no estarán incluidos.

Figura 3.1 Tarjeta de aplicaciones básicas



4.- TARJETA DE ENTRENAMIENTO CON dsPIC30F4011

La tarjeta de entrenamiento con dsPIC30F4011, es una tarjeta que posibilita la exploración de la mayoría de los recursos del dsPIC. Debido al hardware implementado sobre la tarjeta, el usuario se concentra únicamente en el desarrollo del software.

La tarjeta tiene las siguientes características:

- Procesador dsPIC30F4011 de la familia dsPIC de Microchip.
- 1 cristal externo de 7.3728MHz.
- 1 Barra con 8 leds configurables por el usuario.
- 4 microinterruptores, y 3 micropulsadores que sirven como entradas digitales de selección para el usuario.
- 2 potenciómetros conectados como entradas analógicas utilizables por el usuario.
- 1 sensor digital de temperatura configurables por el usuario.

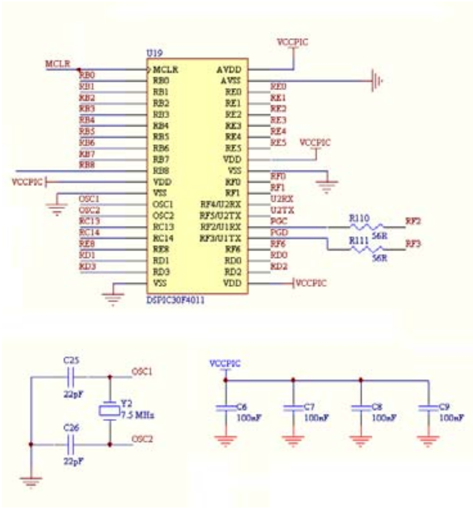
- 1 Relé electromagnético configurable por el usuario.
- 2 UART con adaptador de niveles para RS232*.
- 1 conector para teclado PS/2 configurable por el usuario.
- 1 conector para teclado matricial configurable por el usuario.
- 1 conector para una pantalla alfanumérica LCD configurables por el usuario.
- 1 conector para cuatro visualizadores de siete segmentos del tipo cátodo común.
- 1 conector polarizado para la comunicación con dispositivos I2C ó SPI.
- 1 conector RJ12 de 6 patitas para programación del dsPIC a través de ICD2.
- 3 conectores tipo espadín para conectar hasta tres servomotores.
- 1 bornera x 6 para conectar un motor de pasos unipolar.
- 1 bornera para conectar la fuente de alimentación para motor DC y de pasos.
- 1 bornera para conectar un motor de DC.
- 1 bornera x5 de entrada para el encoder utilizado en la medición de la velocidad del motor DC.
- 3 borneras para la salida de voltaje regulado a 5 voltios.
- 7 borneras auxiliares disponibles para futuras aplicaciones del usuario que canalizan algunos pines del dsPIC, 4 de ellos sirven para conectarse con la tarjeta de potencia manejadora de motor DC.
- Capacidad de soportar 1A en las pistas de control y 5A en las pista de fuerza.
- Voltaje máximo de entrada para los motores de 24VDC.
- Tamaño 198x210 mm.

* En la tarjeta los motores y encoder no serán incluidos.

4.1.- Microcontrolador.

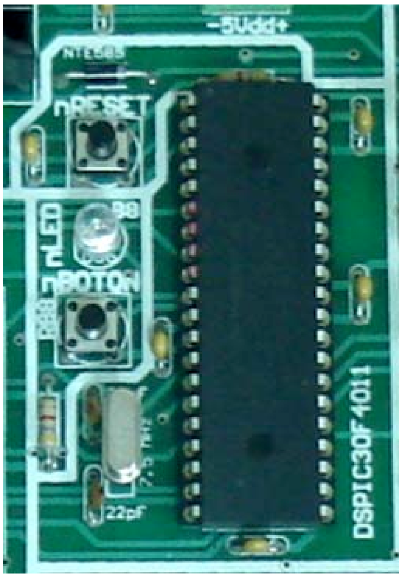
El microcontrolador dsPIC30F4011 dispone de todos los recursos necesarios para controlar entre otros dispositivos a un motor. Su característica más importante es el procesador de señal (DSP). El dispositivo está pensado originalmente para realizar operaciones de procesamiento digital de señal de forma rápida y eficiente. Por ello incorporan instrucciones de ensamblador que permiten realizar operaciones de “multiplicación y suma” en una sola instrucción.

Figura 4.1 Esquema del control



El microcontrolador, utiliza una tensión de alimentación de 5 voltios y un circuito de reloj externo con un cristal de 7.3728 MHZ. También dispone de un circuito de reinicialización formado por un pulsador.

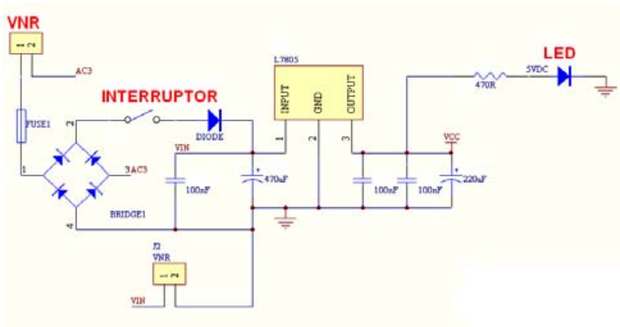
Figura 4.2. dsPIC30F4011 sobre la tarjeta



4.2.- Sistema de alimentación.

El sistema de alimentación es el encargado de fijar la tensión de funcionamiento del sistema de control. La tarjeta trabaja a una tensión de alimentación de 5 V, mediante el regulador LM7805/5V, y un fusible de 1 amperio. En la figura puede observarse el esquema del subsistema de alimentación.

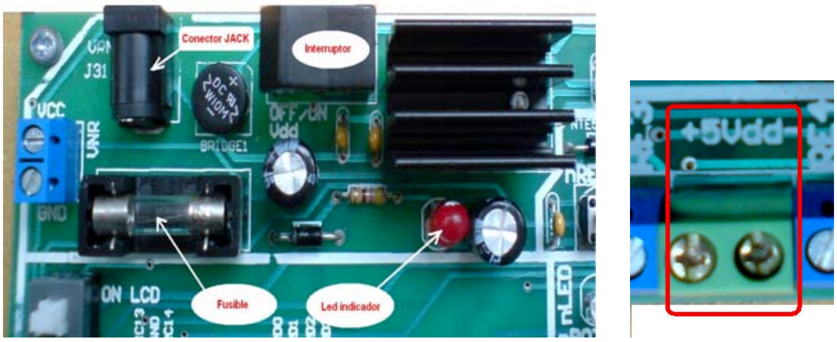
Figura 4.3 Diseño de fuente de control



Para activar la fuente es necesario:

- Conectar una fuente no regulada externa en el conector JACK etiquetado con 'VNR'.
- Activar el interruptor etiquetado por 'Vdd'.
- Inmediatamente se encenderá un led rojo que indica el correcto funcionamiento.

Figura 4.4. Sistema fuente y salida por bornera

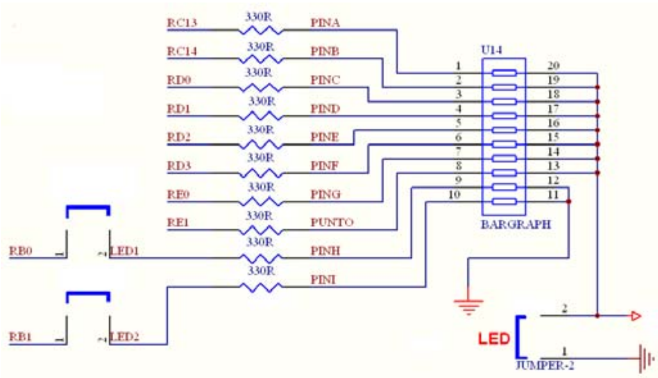


Como se puede verificar, en la tarjeta se encuentran distribuidas tres borneras de color verde con la salida del voltaje regulado a 5 VDC. También puede identificarse las borneras a través de la etiqueta '+5Vdd-'.

4.3.- Barra de leds.

La tarjeta de control incluye una barra de 10 leds de bajo consumo conectados a través de su puente de habilitación a los pines RC13, RC14,RD0, RD1, RD2, RD3, RE0, y RE1.

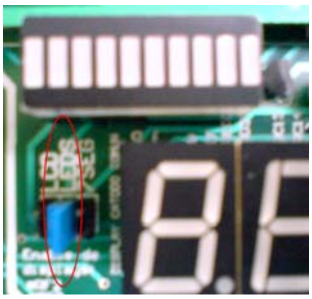
Figura 4.5. Esquema de barra de leds



Para acceder a este periférico será necesario:

Colocar el puente habilitador del módulo de visualización en la posición donde se encuentra la etiqueta *'LEDS'.*

Figura 4.6. Habilitador para barra de leds



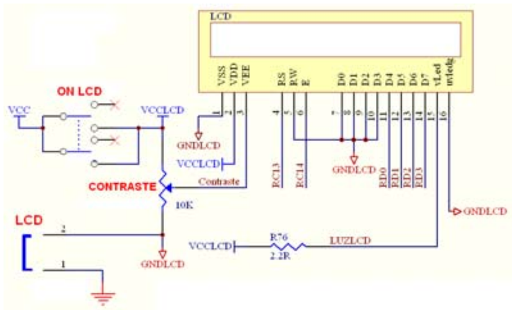
Para encender cada led se debe poner un nivel alto ("1"), en los pines antes mencionados.

NOTA: El acceso a los dos últimos leds es a través de los puentes de habilitación de las señales RB0 y RB1.

4.4.- Control de pantalla de cristal líquido.

La tarjeta incorpora un conector tipo hembra para una pantalla LCD alfanumérica de 2x16. El circuito de control se puede ver en la siguiente figura.

Figura 4.7. Esquema del LCD

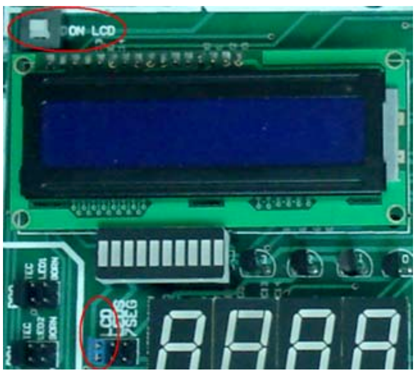


La señal RW esta conectada directamente a GND, con lo cual se escoge el modo de escritura sobre el LCD, la señal RS para selección de registro se conecta a RC13 y la señal de habilitación E al pin RC14. El control de este se realizará por los pines RD0 a RD3.

Para acceder a este periférico será debe:

- Colocar el puente habilitador del módulo de visualización en la etiqueta 'LCD'.
- Presionar el botón de encendido del 'ON LCD'.
- Ajustar la luminosidad mediante el potenciómetro 'CONTRASTE' de 10K.

Figura 4.8. Habilitadores de LCD



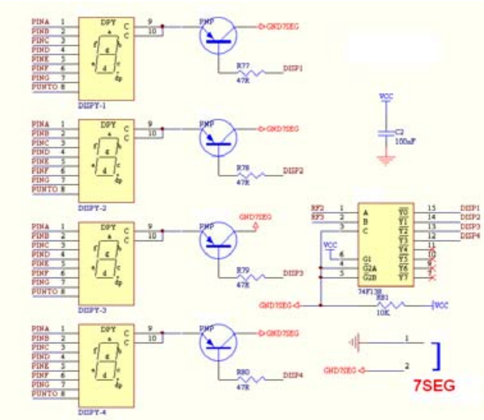
NOTA: El potenciómetro de contraste se encuentra debajo de la pantalla.

4.5.- Visualizadores de 7 segmentos.

Se trata de 4 displays de 7 segmentos conectados a un decodificador (74LS138), para realizar la codificación este elemento toma en su entrada dos señales del puerto F (RF2-RF3), y otra señal de viene del puente (7SEG), finalmente cuatro señales de salida del codificador constituyen una señal de habilitación para cada display.

La visualización de un número de 4 cifras se realiza mediante un barrido continuo.

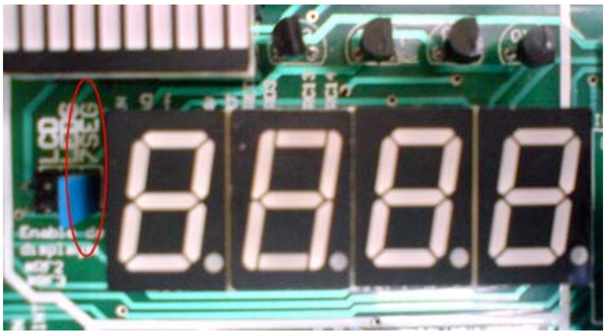
Figura 4.9. Esquema de displays



Para poder trabajar los cuatro visualizadores se debe:

- Ubicar el puente habilitador del módulo de visualización en la etiqueta '7SEG'.
- Colocar en entrada del decodificador que provienen de RF3 y RF2 la cifra '00' para activar el display menos significativo. A continuación se pone cifra '01', y así sucesivamente hasta completar las cuatro cifras, volviendo otra vez a la cifra menos significativa.
- Finalmente se podrá visualizar números de hasta 4 cifras.

Figura 4.10. Habilitador de displays



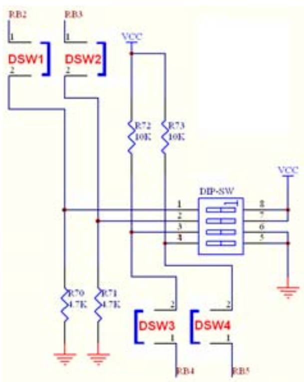
Es conveniente esperar unos pocos milisegundos entre cambios de cifras para que los displays puedan encender.

NOTA: El decodificador para los displays se encuentra debajo de la pantalla LCD.

4.6.- Micro-interruptores.

Los microinterruptores están conectados a los pines RB2, EB3, RB4, y RB5.

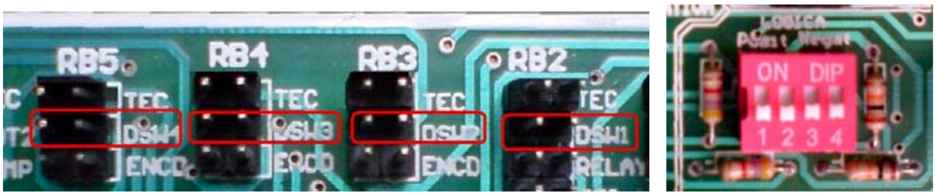
Figura 4.11. Esquema de micro-interruptores



Para utilizar estos cuatro interruptores se debe:

- Configurar sus puentes de habilitación, en las posiciones identificadas por ‘DWS1, DSW2, DSW3, DSW4’.

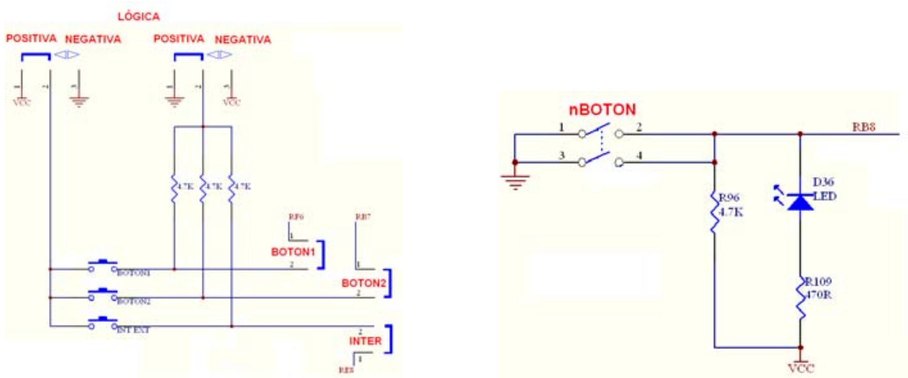
Figura 4.12. Micro-interruptores y sus habilitadores



4.7.- Micro-pulsadores.

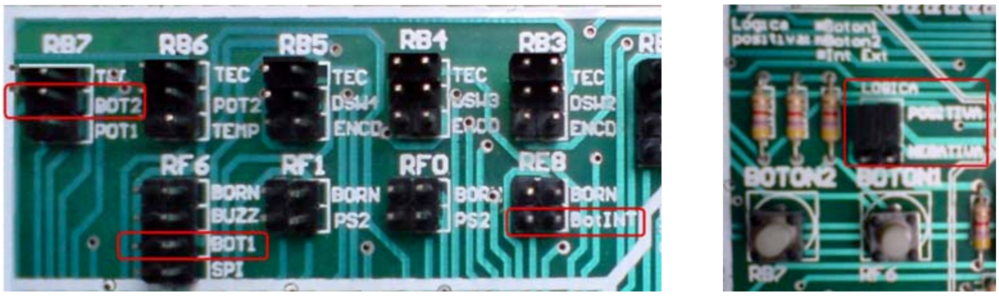
La tarjeta de entrenamiento con dsPIC30F4011 provee al usuario de cuatro micro-pulsadores que pueden usarse como entradas digitales para el controlador. Se debe mencionar que existe un pulsador adicional que sirve como botón de reinicialización del sistema. Estos elementos pueden ser identificados sobre la tarjeta a través de sus nombres ‘BOTON1, BOTON2, INT EXT, nBOTON, y nRESET’.

Figura 4.13. Esquema de micro-pulsadores



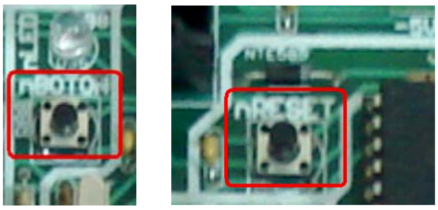
Los tres primeros micro-pulsadores se puede acceder a través de sus puentes de habilitación que deben ser colocado en donde se encuentre la etiquetas ‘BOT1, BOT2, e INTER’ respectivamente, estas pueden ser configuradas en lógica positiva o negativa haciendo uso de 2 jumper disponibles para esta función ambos deben ubicarse en la posición indicada por la etiqueta ‘POSITIVA’ o ‘NEGATIVA’, para que sean utilizables en la lógica requerida.

Figura 4.14. Micro-pulsadores y sus habilitadores



El pulsador nBOTON, es una entrada digital, con una configuración en lógica negativa y se encuentra conectado directamente al pin RB8 del dsPIC.

Figura 4.15. Micro-pulsador de reset y lógica negativa



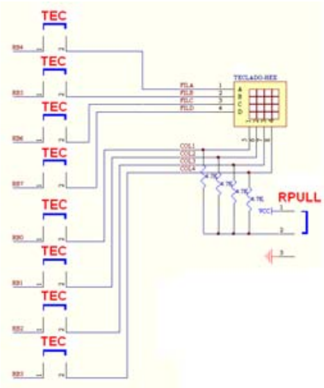
El pulsador nRESET, es una entrada que sirve para la reinicialización del sistema y se encuentra conectado directamente al pin 1del dsPIC.

4.8.- Teclado matricial.

Uno de los elementos de entrada más interesante y útil es el teclado. En este caso se ha previsto a la tarjeta de un teclado matricial de 12 teclas (4x3). El teclado matricial es interesante porque permite tener un mayor número de teclas con un menor número de cables.

Un teclado matricial 4x4 tiene 4 columnas y 4 filas de hilos, de manera que cuando se pulsa una tecla se pone en contacto una columna con una fila; sabiendo qué fila y qué columna se han unido es posible saber cuál de las 16 teclas ha sido pulsada. Las columnas del teclado se encuentran conectadas a los puertos ‘RB0,RB1,RB2,RB3’ y las filas distribuidas en ‘RB4,RB5,RB6,RB7’. Si se utiliza un teclado que no cumpla las condiciones anteriores no se asegura su funcionalidad.

Figura 4.16. Esquema teclado matricial

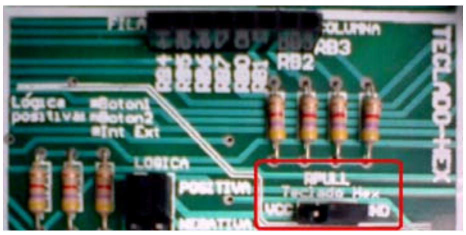


Para acceder a este periférico se debe:

- Configurar los puentes de habilitación desde RB0 hasta RB7, estos habilitadores deben ubicarse en la posición indicada por las etiquetas ‘TEC’ para cada puente.
- También se dispone de resistencia de pull-up las cuales pueden configurarse en alto o en bajo. Para esto existe un puente que debe ubicarse en posición ‘VCC’ o ‘GND’ indicada por la etiqueta ‘RPULL’ según el requerimiento.

Figura 4.17. Habilitadores de teclado matricial

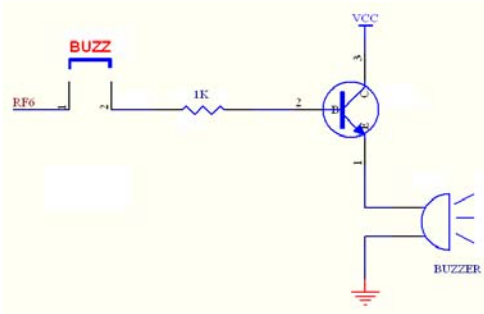
Figura 4.18. Habilitador de pull-up



4.9.- Salida digital por buzzer.

La tarjeta viene provista de buzzer o bocina, este elemento ofrece una señal de salida de audio. Esto posibilita el desarrollo de aplicaciones en donde supervisar visualmente alguna señal digital sea dificultoso para el usuario.

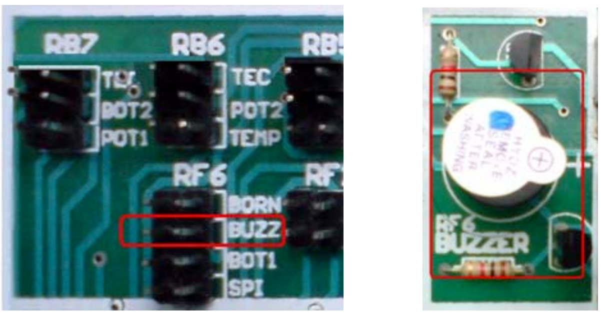
Figura 4.19. Esquema del buzzer



Para utilizar este periférico se debe:

- Colocar el puente de habilitación de RF6, en la posición indicada por la etiqueta 'BUZZ'.

Figura 4.20. Buzzer y sus habilitadores

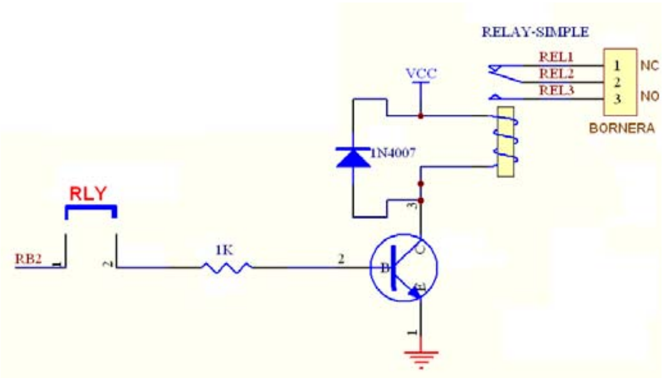


El sonido emitido por el buzzer se escuchará cada vez que llegue a través de RF6 la señal de 1 lógico, ocasionando que el transistor trabaje en modo de saturación. Otra alternativa podría ser el envío de un tren de pulsos a una determinada frecuencia para escuchar diferentes tonalidades sonoras.

4.10.- Relé electrónico.

La bobina del réle se activa cada vez que llegue a través de RB2 la señal de 1 lógico, ocasionando que el transistor asociado trabaje en modo de saturación. Una vez activada la bobina se produce el cambio de estado del los contactos del relé.

Figura 4.21. Esquema del relé



Para acceder a, este interruptor electrónico se debe:

- Colocar el puente de habilitación de RB2, en la posición indicada por la etiqueta 'RELAY'.

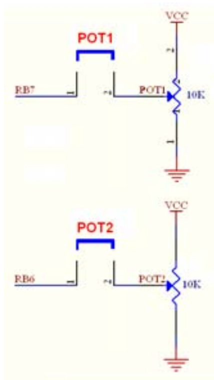
Figura 4.22. Salidas del relé y sus habilitadores

Las salidas de los contactos del relé se encuentran disponibles en una bornera vinculada con este periférico.

4.11.- Potenciómetros.

La tarjeta de entrenamiento cuenta con dos potenciómetros de 10K ajustables de 0 hasta 5 V que permiten evaluar y simular variables analógicas. El esquema eléctrico se muestra en la siguiente figura.

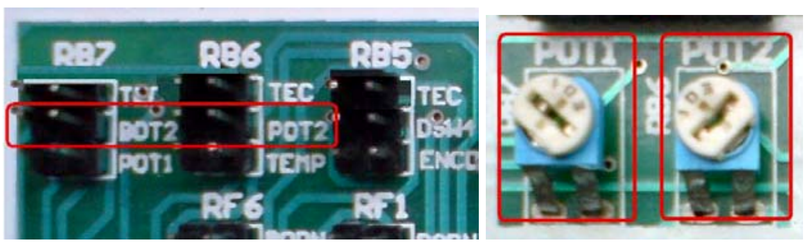
Figura 4.23. Esquema de los potenciómetros



Para acceder a estos dispositivos se debe:

Colocar el puente de habilitación de RB7, y RB6 en la posición indicada por las etiquetas 'POT1' para el potenciómetro 1 y 'POT2' para potenciometro2 según se aprecia en la serigrafía de tarjeta.

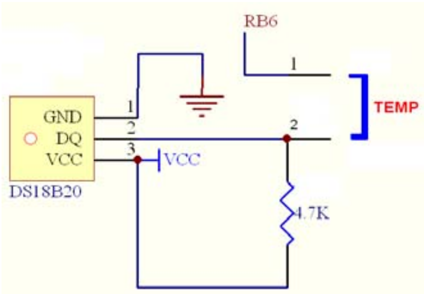
Figura 4.24. Potenciómetros y sus habilitadores



4.12.- Sensor de temperatura.

Existe un conector de tres pines sobre el cual se encuentra implementado un sensor digital de temperatura (DS18B20), usado para medir temperatura entre un rango -55ºC hasta 125 ºC, con una precisión de ± 0.5ºC. El conector implementado posibilita reemplazar el sensor DS18B20, en caso de requerir utilizar sensor de temperatura diferente. En la siguiente figura anterior se muestra el esquema de conexión utilizado.

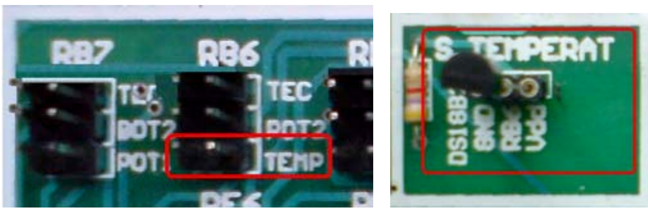
Figura 4.25. Esquema del sensor de temperatura



Para acceder al sensor de temperatura se debe:

Ubicar el puente habilitador de la señal RB6 en la posición indicada por la etiqueta 'TEMP'.

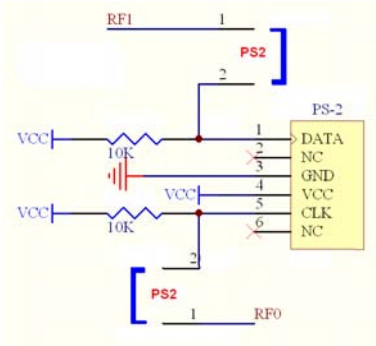
Figura 4.26. Sensor y sus habilitadores



4.13.- Conector para comunicación PS/2.

Existe un conector PS/2 que posibilita la conexión directa entre la tarjeta y un dispositivo que utiliza este tipo de comunicación como un teclado o un mouse de PC. El esquema implementado se muestra a continuación.

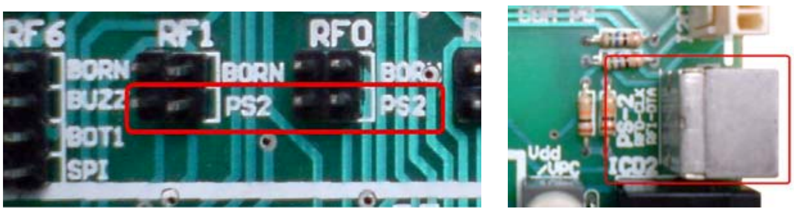
Figura 4.27. Esquema para PS/2



Para disponer de este conector se debe:

Colocar los puentes que habilitadores de las señales RF0 y RF1 en la posición indicada por las etiquetas 'PS2' para ambas señales.

Figura 4.28. Conector PS/2 y sus habilitadores

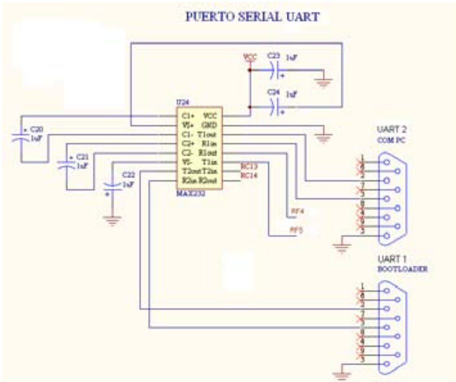


4.14.- Conectores para comunicación UART.

La tarjeta tiene la posibilidad de conectarse con un ordenador o con otro dispositivo mediante una comunicación serie asíncrona que sigue la norma RS-232. La comunicación entre dos dispositivos utilizando una comunicación serie asíncrona se puede conseguir con sólo tres hilos: transmisión (TX), recepción (RX) y GND. El pin de transmisión de un dispositivo debe ir conectado al de recepción del otro dispositivo y viceversa.

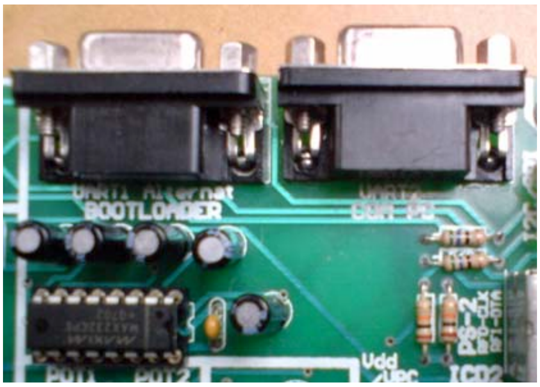
En el ordenador será necesario seleccionar el puerto serie correspondiente (COM1 o COM2) con la velocidad de transmisión deseada. La tarjeta dispone de 2 unidades UART, con su respectivo conector DB9.

Figura 4.29. Esquema para UART



Para el acceso a estos conectores no se requiere configuración de puentes habilitadores por defecto estos se encuentran disponibles, debido a que se encuentran conectados directamente a los pines RC13, RC14, RF4, y RF5 del dsPIC30F4011.

Figura 4.30. Conectores DB9 y el MAX232

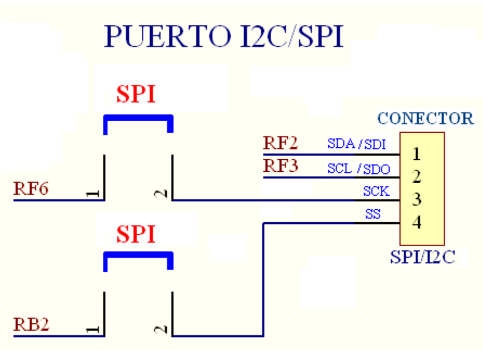


4.15.- Conectores para comunicación I2C/SPI.

La tarjeta de entrenamiento con dsPIC30F4011, tiene otras posibilidades de comunicación serial con otros dispositivos se trata de los buses SPI e I2C. La interfaz SPI es un bus que emplea tres líneas, adicionalmente utiliza una señal de selección de dispositivo (modo maestro-esclavo).

El bus de I2C requiere únicamente dos hilos para la transferencia de información entre los elementos que se acoplan al bus.

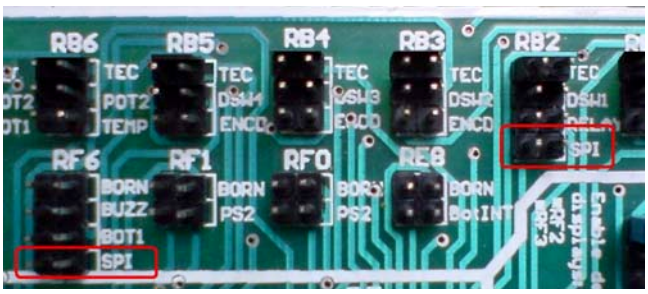
Figura 4.31. Esquema para I2C/SPI



Para disponer de las señales del bus SPI (SS, SDI, SDO, SCK), deben configurarse como se indica a continuación:

- SDI, se accede directamente al pin RF2 sin ningún puente de habilitación.
- SDO, se accede directamente al pin RF3 sin ningún puente de habilitación.
- SCK, se accede ubicando el puente habilitador de la señal RF6 en la posición indicada por la etiqueta 'SPI'.
- SS, se accede ubicando el puente habilitador de la señal RB2 en la posición indicada por la etiqueta 'SPI'.

Figura 4.32. Habilitadores para SPI

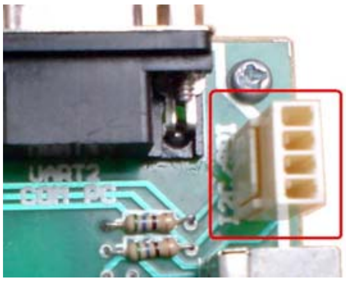


Para disponer de las señales del bus I2C (SDA, SCL), deben configurarse como se indica a continuación:

- SDA, se accede directamente al pin RF2 sin ningún puente de habilitación.
- SDO, se accede directamente al pin RF3 sin ningún puente de habilitación.

Para utilizar estos tipos de comunicaciones se tiene disponible un conector polarizado de cuatro pines, el usuario debe elegir la interfaz que requiera, configurarla, y conectar el dispositivo.

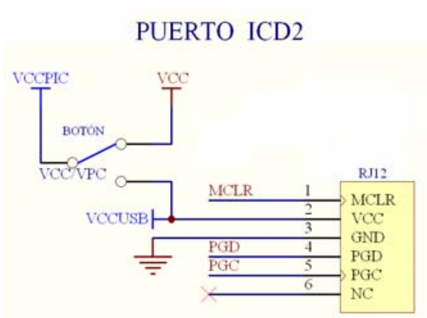
Figura 4.33. Conector para I2C o SPI



4.16.- Conectores para comunicación ICD2.

Para la programación en circuito del dsPIC30F4011 se incorporo a la tarjeta de entrenamiento un conector RJ12. A continuación se muestra la interfaz electrónica implementada.

Figura 4.34. Esquema ICD2



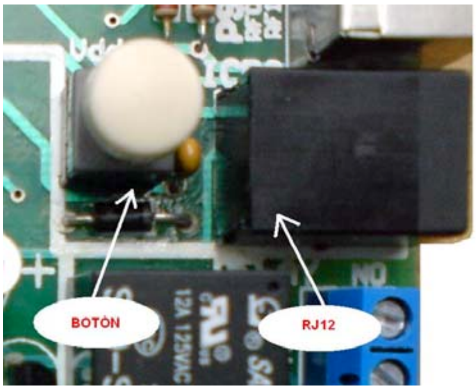
Para disponer de las señales de este puerto de comunicación (PGD, PGC), deben configurarse como se indica a continuación:

- MCLR, se accede directamente al pin 1 del dsPIC, sin ningún puente de habilitación.
- VCC, se accede a través del pulsador de selección de la fuente entre el computador ó la tarjeta.
- GND, se accede directamente a la referencia del sistema.
- PGC, se accede directamente al pin RF2 sin ningún puente de habilitación.
- PGD, se accede directamente al pin RF3 sin ningún puente de habilitación.

Se recuerda que para la programación del dsPIC este debe estar energizado, para cumplir el requisito anterior se dispone de energía proveniente desde la fuente de alimentación de la tarjeta ó externamente desde la fuente USB de un computador, para la selección de la alimentación se implementó un pulsador con retención de señal; estando presionado selecciona la fuente de computador, caso contrario se seleccionara la fuente de la tarjeta.

Es importante procurar que nunca se mantener ambas fuentes activadas al mismo tiempo, en caso de hacerlo el dsPIC no responderá a la ejecución de la aplicación ó a la programación.

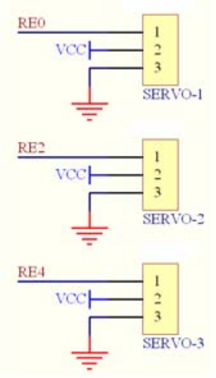
Figura 4.35. RJ12 y su selector de alimentación

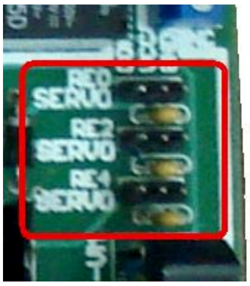


4.17.- Control de servomotores.

Se incorpora tres conectores tipo espadín (3 pines), para el manejo de tres servomotores de modelismo, alimentados desde la fuente de la tarjeta, y controlados a partir de las señales RE0, RE2, RE4, provenientes directamente del dsPIC.

Figura 4.36. Esquema y conectores para servos





4.18.- Control de motores de paso y DC.

Se incorpora un puente configurable para seleccionar entre manejo de un motor de pasos o un motor DC, con un sistema de aislamiento del control con la parte fuerza a través de cuatro optoacopladores (PC817).

Las señales RE2, RE3, RE4, RE5, manejan a los optoacopladores los cuales a su vez controlan la parte de fuerza formada por cuatro transistores mosfet de canal N (IRF530) conectados en una configuración en puente H.

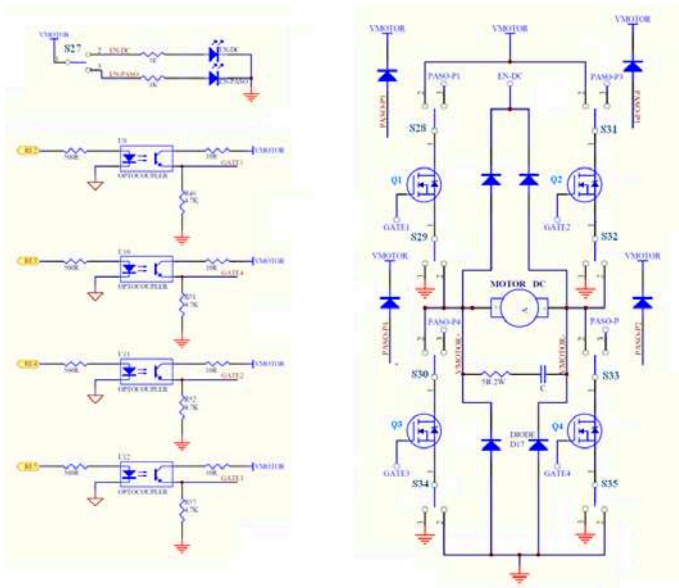
Para la selección entre el motor de pasos y el DC se dispone de un banco de nueve interruptores de tres posiciones (S27, S28, S29, S30, S31, S32, S33, S34, S35), que seleccionaran la configuración de uno de los dos motores o de ninguno.

Cuando todos los interruptores se encuentran presionados hacia arriba indica la selección de la configuración del motor de pasos.

Cuando todos los interruptores se encuentran presionados hacia abajo indica la selección de la configuración del motor de DC.

Cuando todos los interruptores se encuentran presionados hacia en la mitad indica que ninguna configuración de los motores.

Figura 4.37. Esquema para motor de pasos y DC



Para utilizar un motor de pasos con la tarjeta se debe:

- Presionar los 9 interruptores de selección de motor, tal que queden levantados del lado de la etiqueta ‘MotorPASO’.
- Conectar a la tarjeta una fuente DC externa de hasta 24 VDC, para la conexión de la fuente esta implementada una bornera roja de 2 bornes, identificable con la etiqueta ‘BATTERY DC’.
- Conectar el motor de pasos con la tarjeta, para la conexión del motor esta implementado un grupo de borneras azules, identificables con la etiqueta ‘MOTOR PASO’. Cada borne se encuentra etiquetado con su correspondiente señal para el control de este tipo de motores.
- Activar la fuente de control con el interruptor etiquetado con ‘Vdd’, el interruptor debe quedar levantado de lado indicado por ‘ON’. Inmediatamente se debe encender el led indicador de color rojo (fig. 5).
- Activar la fuente del motor con el interruptor etiquetado con ‘Vdd’, el interruptor debe quedar levantado de lado indicado por ‘ON’. Inmediatamente se deben encender los dos leds. Uno de ellos color rojo que indica la correcta energización del sistema, otro de color amarillo indicando la correcta configuración para el manejo del motor de pasos.
- Finalmente todo estará listo para que el usuario comience con el desarrollo de sus aplicaciones.

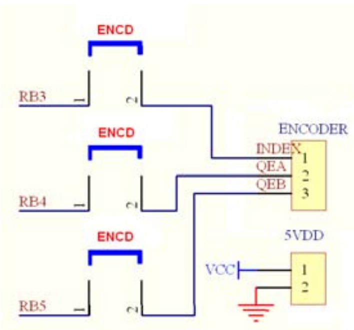
Para utilizar un motor de DC con la tarjeta se debe:

- Presionar los 9 interruptores de selección de motor, tal que queden levantados del lado de la etiqueta ‘MotorDC’.
- Conectar a la tarjeta una fuente DC externa de hasta 24 VDC, para la conexión de la fuente esta implementada una bornera roja de 2 bornes, identificable con la etiqueta ‘BATTERY DC’.
- Conectar el motor DC con la tarjeta, para la conexión del motor esta implementado una de bornera azul, identificables con la etiqueta ‘MOTOR DC’.
- Activar la fuente de control con el interruptor etiquetado con ‘Vdd’, el interruptor debe quedar levantado de lado indicado por ‘ON’. Inmediatamente se debe encender el led indicador de color rojo (fig. 5).
- Activar la fuente del motor con el interruptor etiquetado con ‘Vdd’, el interruptor debe quedar levantado de lado indicado por ‘ON’. Inmediatamente se deben encender los dos leds. Uno de ellos color rojo que indica la correcta energización del sistema, otro de color verde indicando la correcta configuración para el manejo del motor DC.
- Finalmente todo estará listo para que el usuario comience con el desarrollo de sus aplicaciones.

4.19.- Manejo del encoder.

Para poder realimentar el manejo de un motor DC, se necesita un sistema que nos diga a que velocidad nos movemos para poder compararla con la que deseamos y así poder corregir el error. Esta velocidad será medida mediante los pulsos suministrados por un encoder, la cual nos suministrara pulsos para la medida de la velocidad y el sentido de giro del motor.

Figura 4.38. Esquema para encoder

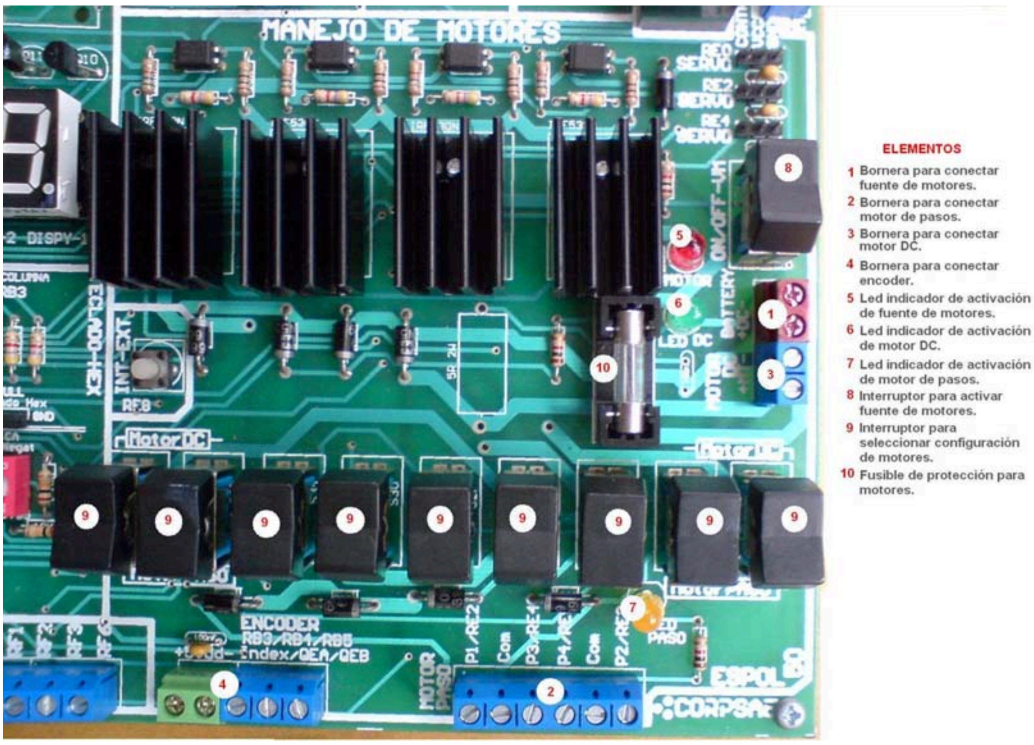


Para trabajar con las señales de control del encoder se debe:

- Configurar los puentes que habilitan las señales RB3, RB4, y RB5.
- Cada puente debe ubicarse en las etiquetas 'ENCD', quedando disponibles las señales INDEX, QEA, y QEB respectivamente.

Para la conexión de un encoder incremental con salida en cuadratura a la tarjeta de entrenamiento con dsPIC30F4011, esta tarjeta dispone de una bornera de color azul de tres bornes para las señales de control, también hay otra bornera de color verde con dos bornes para la energizar el sensor. Estas dos borneras se encuentran grupadas sobre la tarjeta y etiquetadas con el nombre 'ENCODER'. Adicionalmente en cada borne tiene marcado su correspondiente señal.

Figura 4.39. Manejador para motores de pasos y DC



Existe un grupo de borneras que cumplen la función de llevar a sus bornes 14 señales del dsPIC30F4011 las cuales son: RB0, RB1, RE0, RE1, RE2, RE3, RE4, RE5, RE8, RF0, FR1, RF2, RF3, RF6. Cada borne tiene una etiqueta para identificar la señal que tiene asociada.

- B0, ubicar el puente habilitador de la señal RB0 en la posición indicada por la etiqueta 'BORN'.

- RB1, se accede ubicando el puente habilitador de la señal RB1 en la posición indicada por la etiqueta 'BORN'.
- RE0, se accede directamente al pin de la señal RE0 sin ningún puente de habilitación.
- RE1, se accede directamente al pin de la señal RE1 sin ningún puente de habilitación.
- RE2, se accede directamente al pin de la señal RE2 sin ningún puente de habilitación.
- RE3, se accede directamente al pin de la señal RE3 sin ningún puente de habilitación.
- RE4, se accede directamente al pin de la señal RE4 sin ningún puente de habilitación.
- RE5, se accede directamente al pin de la señal RE4 sin ningún puente de habilitación.
- RE8, se accede ubicando el puente habilitador de la señal RE8 en la posición indicada por la etiqueta 'BORN'.
- RF0, ubicar el puente habilitador de la señal RF0 en la posición indicada por la etiqueta 'BORN'.
- RF1, ubicar el puente habilitador de la señal RF1 en la posición indicada por la etiqueta 'BORN'.
- RF6, ubicar el puente habilitador de la señal RF6 en la posición indicada por la etiqueta 'BORN'.

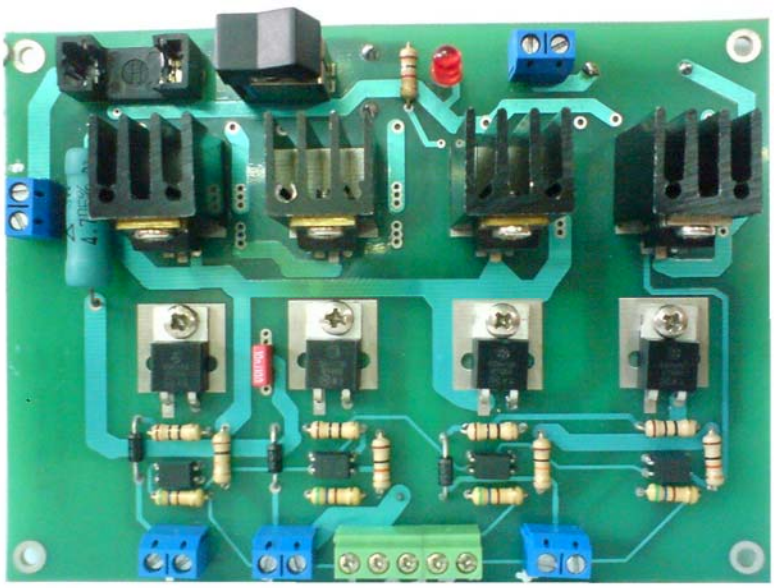
Figura 04.40. Habilitadores de salida por bornera

Figura 4.41. Bornera de salida de señales

5.- TARJETA MANEJADORA DE MOTORES DC.

En esta tarjeta se encuentran todos los recursos de potencia para controlar motores y otros dispositivos externos como contactores.

Figura 5.1. Tarjeta para motor DC



Se pueden destacar las siguientes características de la tarjeta:

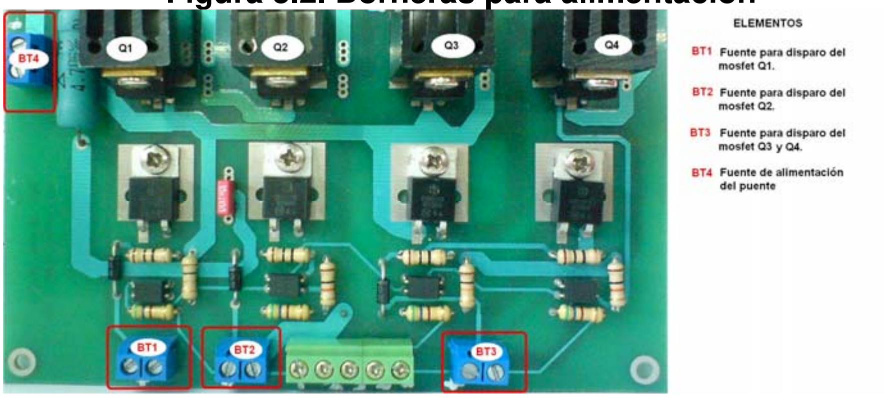
- 1 puente en H realizado con parejas de transistores unipolares discretos (MOSFET) capaces de soportar hasta 14A de corriente en régimen permanente y hasta 56A de pico. Se puede controlar un motor DC en velocidad y sentido.
- 1 led indicador del funcionamiento del sistema.
- Capacidad de soportar 1A en las pistas de control y 6.5A en las pista de fuerza.
- Tamaño 102x140 mm.

Esta tarjeta de fuerza irá acoplada con una de la *tarjeta de aplicaciones básicas*, o con *tarjeta de entrenamiento con dsPIC30F4011*, la cual trabajará como el control.

5.1.- Sistema de alimentación del manejador.

La tarjeta tiene implementado un juego de cuatro borneras (BT1,BT2,BT3,BT4), para la alimentación sistema de disparos de los mosfet canal N (Q1, Q2, Q3, Q4), además tiene un sistema de aislamiento por optoacopladores entre la tarjeta de fuerza y la de control.

Figura 5.2. Borneras para alimentación

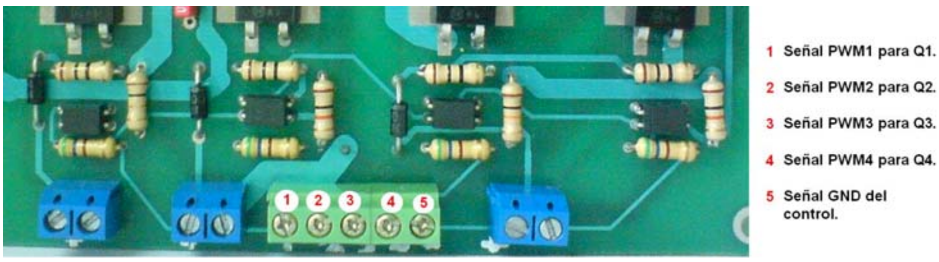


Debido a que BT4 y BT3 tienen la misma referencia de GND, se puede conectar la misma fuente a esas borneras.

5.2.- Sistema de control.

La tarjeta tiene implementado un juego de borneras para las señales de control provenientes de la tarjeta de aplicaciones básicas, o con tarjeta de entrenamiento con dsPIC30F4011.

Figura 5.3. Borneras para el sistema de control

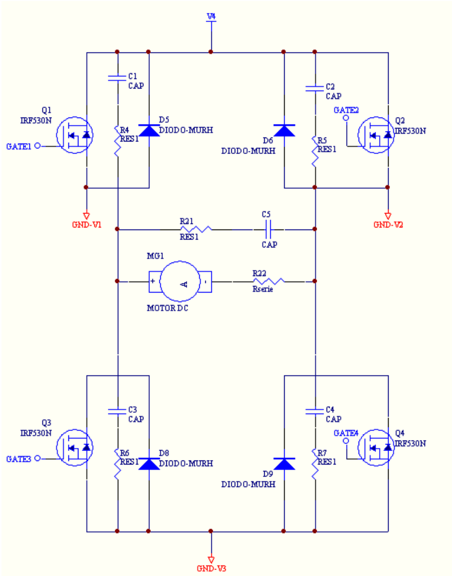


5.2.- Puente en H.

La tarjeta dispone de 1 puente en H, formado por parejas de transistores unipolares mosfet IRF530. Esta tarjeta permite usar cualquier técnica de troceo clase E, como por ejemplo modulación de ancho de pulso y conmutación de voltaje bipolar, unipolar, entre otras.

El esquema eléctrico del puente se puede ver en siguiente figura.

Figura 5.4. Puente H del la tarjeta



Para utilizar esta tarjeta manejadora de motores DC se debe:

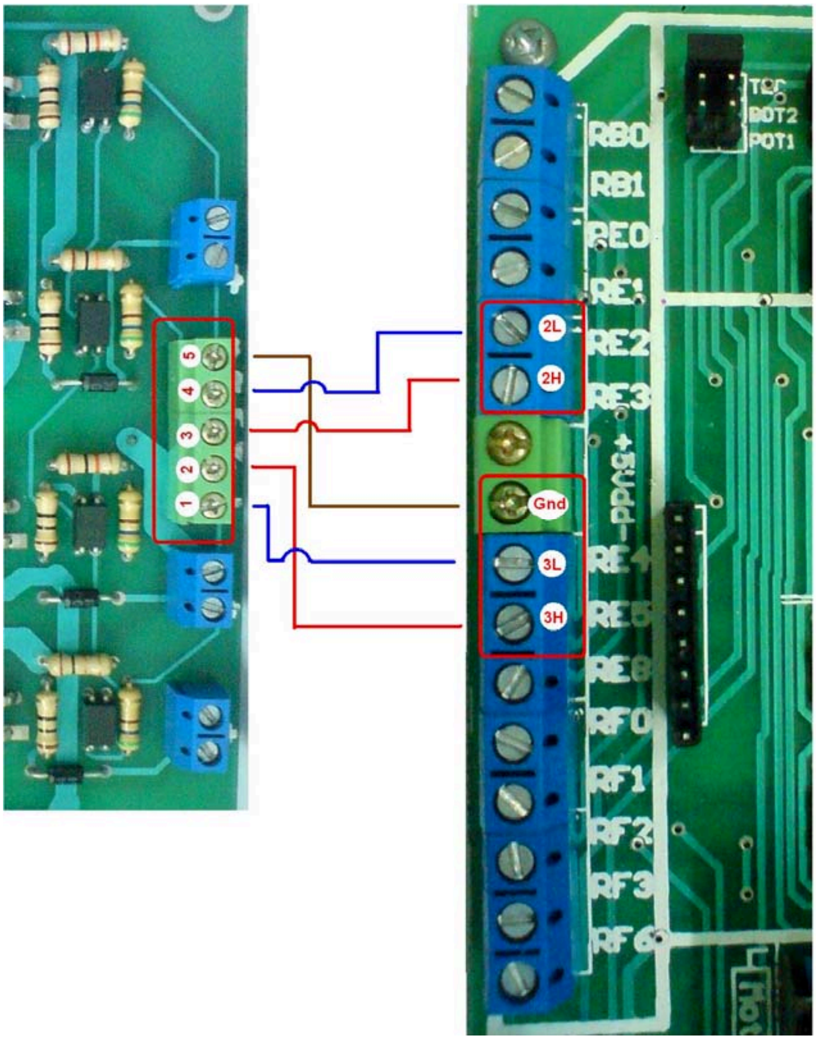
- Escoger la tarjeta control que se usará como interfaz.
- Identificar en la tarjeta de control las borneras que canalizan las señales para el manejo del módulo PWM.*
- Identificar en la tarjeta de fuerza las borneras de control y asociarlas con las del paso anterior.
- Conectar las tarjetas (fuerza y control) a través de las borneras, utilizar cable. **
- Conectar las fuentes de la tarjeta de fuerza en cada uno de sus borneras.
- Presionar el interruptor para energizar el sistema de fuerza.

- Inmediatamente se deberá encender un led rojo indicador de que la tarjeta esta lista para realizar las pruebas.

*Tanto la *tarjeta de aplicaciones básicas*, como la *tarjeta de entrenamiento con dsPIC30F4011*, tienen la salida de cada una de estas señales por bornera (PWM1L/RE0, PWM1H/RE1, PWM2L/RE2, PWM2H/RE3, PWM3L/RE4, PWM3H/RE5).

**En la figura 5.5 se muestra las conexiones entre la *tarjeta de entrenamiento con dsPIC30F4011* con la tarjeta manejadora de motores DC.

Figura 5.5. Conexión entre tarjetas



BIBLIOGRAFÍA

1. ANGULO USATEGUI JOSÉ, Microcontroladores avanzados dsPIC 1º Edición; Thomson; Madrid-España, 2006; pp 3-495.
2. DEWAN B. S., Power semiconductor, Jhon Wiley & Sons, Toronto-Canada, 2006; pp 282-352.
3. GITMAN J. LAWRENCE, Fundamentos de administración financiera 3º Edición, HARLA, Mexico, 1986; pp 149-154.
4. MALONEY J. TIMOTHY, Electrónica industrial moderna 3º Edición; Pretince Hall; Naulcalpan de Juarez-Mexico, 2007; pp 526-530.
5. ‘McCOMB GORDON, The Robot Builder's Bonanza 2º Edición, McGrill-Hill, Estados Unidos, 2001; pp 253-318.
6. Guía de usuario para aplicación de bootloader http://www.ingenia-cat.com/
7. Guía para elaborar una Tesis de Grado http://www.cib.espol.edu.ec
8. Implementación de controles PID http://www.automatas.org/hardware/teoria_pid.htm
9. Herramientas y estándares para diseño de circuitos impresos http://www.ipc.org/ContentPage.aspx?pageid=4.6

10. Hoja del fabricante para las especificaciones de los integrados utilizados en la implementación de las tarjetas.

http://www.datasheetcatalog.com

11. Hojas de fabricante de especificación del dsPIC30F y dsPIC30F4011

http://www.microchip.com

12. Información general sobre dsPIC

http://isa.uniovi.es/~fernando/Programa_II_EUITI.htm

13. Learn LabVIEW DSP in 3 Hours, Getting Started with LabVIEW

http://www.ni.com

14. Manual de mikroBasic para dsPIC

http://www.mikroe.com

15. Proyecto para un Controlador PID

http://perso.wanadoo.es/chyryes/circuitos/pid.htm

16. Precio de componentes electrónicos

http://www.electroavil-es.com/

17. Tutorial de diseño de circuitos impreso con PROTEL99

http://www.uhu.es/manuel_sanchez/tutoriales/protel99/protel99.html

18. Tutorial sobre Comunicación Serial

http://digital.ni.com/worldwide/

19. Reglas para diseño electrónico

http://www.monografias.com/trabajos35/dibujo-electronico/dibujo-electronico.shtml