



# ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

Titulación:

INGENIERO TÉCNICO INDUSTRIAL ELÉCTRICO

Título del proyecto:

CONTROL POR FRECUENCIA DE UN CONVERTIDOR  
TRIFÁSICO

Sergio Sóñora Mariño

Jesús López Taberna

Pamplona, 26 de Julio de 2012

*El presente PFC es un esfuerzo en el cual, directa o indirectamente, participaron varias personas leyendo, opinando, corrigiendo, teniéndome paciencia, dando ánimo, acompañando en los momentos de crisis así como en los momentos de felicidad.*

*Agradezco a mi tutor Jesús López por haber confiado en mi persona, por la paciencia y por la dirección de este trabajo. A Alberto Equiza por los sabios consejos, el apoyo y el ánimo que me brindó.*

*A mi madre y a mi hermana que de forma incondicional, entendieron mis ausencias y mis malos momentos. A mi padre, que a pesar de la distancia siempre estuvo atento para saber cómo iba mi proceso. A ti Judith, que desde un principio hasta el día de hoy sigues dándome ánimo para terminar con este proyecto.*

*Gracias a todos.*



# Índice general

Índice de figuras	VII
Índice de tablas	VII
<b>1. Introducción</b>	<b>2</b>
<b>2. Módulo de potencia</b>	<b>6</b>
<b>3. Microcontrolador</b>	<b>14</b>
3.1. Características y elección del microcontrolador . . . . .	15
3.2. El Oscilador . . . . .	19
3.3. Conversor analógico-digital (ADC) . . . . .	20
3.4. Módulo PWM . . . . .	24
3.5. Módulo QEI . . . . .	31
<b>4. Placa de Control</b>	<b>36</b>
4.1. dsPIC33FJ256MC710A . . . . .	36
4.2. Lógica de encendido y apagado . . . . .	41
4.2.1. Pulsadores . . . . .	46
4.2.2. Diodos LED . . . . .	48
4.3. Señales PWM . . . . .	49
4.4. Acondicionamiento de señales . . . . .	51
4.4.1. Acondicionado de tensión . . . . .	52
4.4.2. Acondicionado de corriente . . . . .	54

4.4.3. Acondicionamiento del encoder . . . . .	58
4.4.4. Acondicionamiento señales de estado . . . . .	60
4.5. Protección de sobrecorriente . . . . .	61
4.6. Protección de sobretensión . . . . .	67
4.7. Charge Input Signal (CIS) . . . . .	68
4.8. Display LCD alfanumérico . . . . .	70
4.9. Fuente de alimentación . . . . .	73
<b>5. Montaje final y ensayos</b>	<b>76</b>
5.1. Resultados . . . . .	80
<b>6. Conclusiones y líneas futuras</b>	<b>84</b>
<b>Apéndices</b>	<b>90</b>
<b>A. Bits de configuración</b>	<b>92</b>
A.1. Registros de configuración . . . . .	92
A.2. Ejemplo de configuración . . . . .	100
<b>B. Configuración del oscilador</b>	<b>102</b>
B.1. Registros de configuración . . . . .	102
<b>C. Configuración del módulo ADC</b>	<b>104</b>
C.1. Registros de configuración . . . . .	104
C.2. Ejemplo de Inicialización . . . . .	108
<b>D. Configuración del módulo PWM</b>	<b>110</b>
D.1. Registros de configuración . . . . .	110
D.2. Ejemplo de Inicialización . . . . .	114
<b>E. Configuración del módulo QEI</b>	<b>116</b>
E.1. Registros de configuración . . . . .	116
E.2. Ejemplo de Inicialización . . . . .	119

<b>F. Configuración del Timer1 y Timer2</b>	<b>120</b>
F.1. Registros de configuración . . . . .	120
F.2. Ejemplo de Inicialización . . . . .	122
<b>G. Programa</b>	<b>124</b>
G.1. main.c . . . . .	124
G.2. arranque.h . . . . .	136
G.3. Librería LCD . . . . .	140
<b>Anexos</b>	<b>148</b>

# Índice de figuras

2.1. Esquema de potencia del Semistack SKS 22F . . . . .	7
2.2. Circuito para la medida de la tensión $V_{DC}$ . . . . .	8
2.3. Esquema eléctrico del módulo de potencia . . . . .	10
3.1. Diagrama de pines del dsPIC33FJ256MC710A . . . . .	17
3.2. Arquitectura Harvard . . . . .	18
3.3. Diagrama de bloques del PLL . . . . .	20
3.4. Diagrama de tiempos del modo <i>Center-Aligned</i> . . . . .	25
3.5. Tiempos muertos . . . . .	26
3.6. PWM generada a partir de la tabla del seno . . . . .	27
3.7. Escalado de los ciclos de trabajo . . . . .	28
3.8. Diagrama de tiempos del módulo QEI . . . . .	31
3.9. Diagrama de tiempos del filtro . . . . .	32
3.10. Diagrama de tiempos del QEI en modo x4 . . . . .	33
4.1. Entradas y salidas del microcontrolador . . . . .	37
4.2. Circuito oscilador . . . . .	37
4.3. Circuito para la programación del dsPIC . . . . .	38
4.4. Circuito de adaptación de las señales RS-232 . . . . .	39
4.5. Conector SPI . . . . .	40
4.6. Circuito lógico para la detección del tipo de fallo de los drivers	42
4.7. Circuito lógico que da lugar al apagado del convertidor. . . . .	43
4.8. Biestable utilizado para el encendido y apagado del convertidor	45
4.9. Configuración del pulsador P_SGIRO . . . . .	46

4.10. Configuración de los pulsadores P_MARCHA y P_PARO . . .	47
4.11. Configuración del pulsador P_RESET . . . . .	48
4.12. Circuito de visualización de diodos LED . . . . .	49
4.13. Circuito de acondicionamiento de las señales PWM . . . . .	50
4.14. Circuito de acondicionamiento de la señal PWM de la rama de disipación . . . . .	51
4.15. Circuito de acondicionamiento y filtrado de la tensión de bus .	54
4.16. $R_M$ y divisor de tensión . . . . .	56
4.17. Circuito de acondicionamiento y filtrado de la corriente . . . .	58
4.18. Circuito de acondicionamiento y filtrado del encoder . . . . .	60
4.19. Circuito de acondicionamiento de la señal SOBREV . . . . .	61
4.20. Comparador básico en montaje no inversor e inversor . . . . .	62
4.21. Circuito utilizado para generar REF_MAX . . . . .	63
4.22. Circuito utilizado para generar REF_MIN . . . . .	64
4.23. Etapa de comparación de la protección de sobrecorriente de la fase U . . . . .	66
4.24. Circuito para la protección de sobretensión . . . . .	67
4.25. Circuito de acondicionamiento del CIS . . . . .	69
4.26. Esquema de conexiones del LCD . . . . .	72
4.27. Fuente de alimentación Emerson LPT45 . . . . .	73
4.28. Circuito regulador de tensión de 3.3V . . . . .	74
5.1. Conjunto placa de control, Semistack y fuente de alimentación. Vista lateral. . . . .	76
5.2. Conjunto placa de control, Semistack y fuente de alimentación. Vista superior. . . . .	77
5.3. Detalle display LCD . . . . .	77
5.4. Detalle placa de control . . . . .	78
5.5. Detalle diodos LED y display LCD . . . . .	79
5.6. Señal de salida a 50Hz . . . . .	81
5.7. Vista de la corriente frente a la tensión a 50Hz . . . . .	82
5.8. Tiempo muerto de $3\mu s$ . . . . .	82

# Indice de tablas

2.1. Características más importantes del módulo de potencia . . .	8
2.2. Asignación de pines del conector CN1 . . . . .	11
2.3. Asignación de pines del conector CN2 . . . . .	11
2.4. Asignación de pines del conector CN3 . . . . .	12
3.1. Microcontroladores descartados . . . . .	16
4.1. Tabla de verdad de las señales de estado de los drivers . . . .	42
4.2. Tabla de verdad del biestable tipo D . . . . .	45
4.3. Asignación de pines del encoder . . . . .	59
4.4. Asignación de pines del LCD . . . . .	71







# 1. Introducción

Este capítulo introduce los conceptos básicos desarrollados en el proyecto así como los objetivos prioritarios durante su elaboración.

El presente documento se entrega como memoria del Proyecto Final de Carrera de los estudios de Ingeniería Técnica Industrial especialidad en Electricidad, trabajo realizado en el departamento de Ingeniería Eléctrica y Electrónica, perteneciente a la Universidad Pública de Navarra.

Desde el departamento se han ido desarrollando diferentes convertidores para su utilización en accionamientos eléctricos, todos apoyándose en el control analógico. Esta vez hemos querido romper con la línea establecida hasta ahora y diseñar un nuevo modelo de convertidor más acorde a lo que vemos hoy día en el ámbito industrial. Un convertidor innovador en el sentido que incorpora el uso de un microcontrolador como elemento central para realizar la estrategia de control del accionamiento.

Además, se pretende que sea una plataforma que sirva para el desarrollo de futuros PFCs para los estudiantes de las nuevas ingenierías de grado, permitiendo de este modo desarrollar y motivar el aprendizaje en el uso y programación de los microcontroladores.

Una de las grandes ventajas de utilizar un microcontrolador radica en la programación de éste, con lo cual, sin tener que realizar cambios en el

hardware, podemos modificar los parámetros establecidos con anterioridad e incluso toda la estrategia de control por otra que nos convenga más según la aplicación.

El convertidor realizado en este proyecto está orientado para ser utilizado en un accionamiento eléctrico, ya que fue concebido para controlar la velocidad de una máquina de corriente alterna (MCA). De todos modos, al disponer de una parte programable (el microcontrolador) podemos utilizarlo no solo en máquinas síncronas o asíncronas, sino también en máquinas de corriente continua o incluso en cualquier otra situación donde necesitemos un control total sobre la tensión, corriente y la forma de onda requerida por la aplicación.

Conviene comentar también las ventajas de los accionamientos en alterna frente a los de continua. Históricamente los accionamientos de continua han sido muy utilizados ya que se podían controlar por separado par-velocidad y flujo, lo que los hacía sumamente sencillos de controlar frente a los de corriente alterna. Pero el gran desarrollo de la electrónica de potencia (mejora de semiconductores, más fiables, menores pérdidas, etc) y la digital (microcontroladores, DSPs, FPGAs...) junto con las ventajas de los motores síncronos y asíncronos (más robustos y de menor coste) han llevado al desarrollo sin precedentes de los accionamientos para MCA, dejando en un segundo plano a los de continua.

La estrategia de control utilizada será el llamado control escalar o  $V/f$ , este control nos permite, mediante una variación de la tensión y frecuencia de la señal con la que se alimenta el motor en una proporción constante, variar la velocidad de éste manteniendo constante el flujo del entrehierro, con un aumento de la tensión a bajas velocidades para evitar efectos resistivos propios de la máquina.

La técnica utilizada para conseguir implementar el control  $V/f$  es la conmutación SPWM (modulación del ancho de pulso sinusoidal). Más

adelante se explica en detalle en que consiste y como se consigue. Recordar que la modulación la realiza el microcontrolador, por lo que podríamos modificar la estrategia y técnica mediante una reprogramación.

Las técnicas de control se pueden agrupar en dos tipos, las de lazo abierto y las de lazo cerrado. El control escalar o V/f se encuentra dentro de las de lazo abierto, ya que no necesita de una realimentación para controlar la velocidad de la máquina, por lo que acaba siendo un método más fácil de implementar.

Aun así, el convertidor ha sido diseñado para que pueda trabajar también en lazo cerrado, y haya una realimentación que ayude a comparar en todo momento la referencia a seguir con la magnitud medida. De este modo se pueden realizar técnicas de control que tengan en cuenta la velocidad de la máquina, posición y corriente, obteniendo una mayor precisión.

El objetivo final del presente PFC es que sirva como plataforma para el desarrollo de nuevas estrategias de control en futuros PFCs, para ello, nos hemos centrado en el diseño de la placa de control siguiendo las siguientes líneas:

- Dotar a la placa de unas protecciones robustas, redundando estas, por lo que aparte de la protección que se puede programar en el micro hemos decidido incluir unas protecciones analógicas que protegen el equipo a expensas de lo que decida el micro, haciéndolo más robusto ante posibles sobrecargas, sobretensiones, etc.
- Incluir el mayor número de sensores y modos de comunicación entre el microcontrolador y el exterior, dotando a la placa de entradas para encoder óptico, sensores de corriente y tensión, comunicación serie RS-232, comunicación SPI, puerto para programación, e incluso un display alfanumérico para mostrar información en tiempo real.



## 2. Módulo de potencia

En este capítulo se describe el módulo de potencia, punto de partida a la hora de diseñar la placa de control. Se muestran las características más importantes así como la asignación de cada conector.

La etapa de potencia está formada por un módulo del fabricante Semikron, concretamente el SEMISTACK SKS 22F B6U+E1CIF+B6CI. En el *Anexo IX: Hojas de características* se puede encontrar la documentación del módulo.

El módulo consta de los siguientes elementos:

- Puente rectificador trifásico
- Batería de condensadores
- Tres células de conmutación formadas por 2 IGBTs cada una
- Rama de disipación formada por IGBT más diodo en serie
- Driver para cada célula de conmutación y otro para la rama de disipación
- Radiador de aluminio con ventilación forzada
- Dos sensores de tipo Hall para medidas de corriente
- Resistencia de precarga con su relé de desconexión

En la figura 2.1 se puede observar la configuración de los distintos elementos que integran el módulo de potencia.

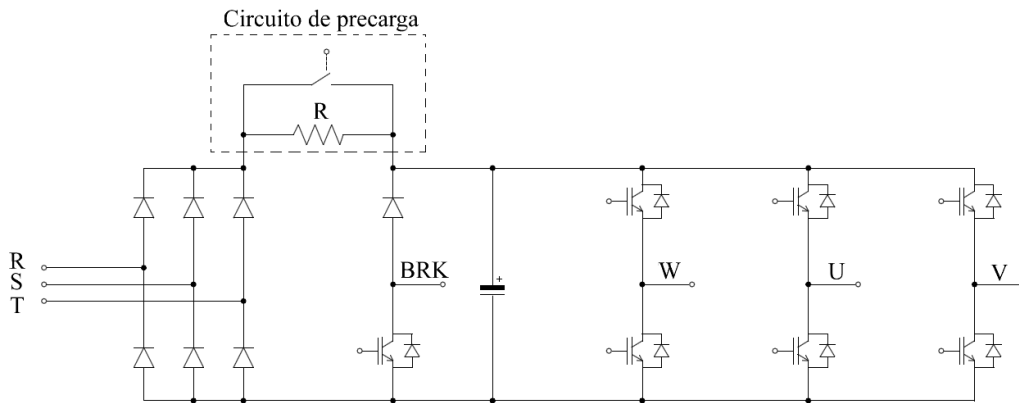


Figura 2.1: Esquema de potencia del Semistack SKS 22F

El módulo está diseñado de tal forma que tan solo necesita de las señales de control para poder funcionar. En la tabla 2.1 se detallan las características más importantes y que se han de tener en cuenta a la hora de diseñar la etapa de control.

## Medida de tensión

En el Semistack no se incluye un sensor para la medida de tensión del bus de continua, algo necesario para utilizar la rama de disipación. Se opta por añadir una célula LEM para el sensado de la tensión  $V_{DC}$ . En la figura 2.2 se puede ver el circuito utilizado.

	Condiciones	Valores	Unidades
Vred	Tensión de alimentación del convertidor	380	V
Irms max Tamb = 35°C	Sin sobrecarga y a 10 kHz de Fconmutación	22	A
	Sobrecarga del 150 %, durante 60s cada 10min	27	A
	Sobrecarga del 200 %, durante 10s cada 10min	30	A
Ceqvl	Capacidad del banco de condensadores	1360/800	$\mu\text{F}/\text{V}$
Tds %	Tiempo de descargar del banco de condensadores	240	s
Vdc max	Máxima tensión que se puede aplicar al bus de continua	750	V
Tamb	Rango admisible de temperatura ambiente	-20..+55	°C
Cooling	Alimentación DC máxima ventiladores	24	V
	Consumo de corriente por ventilador	110	mA
Pérdidas	Trabajando a Pmax y Tamb = 35 °C	300	W
	Eficiencia	97	%

Tabla 2.1: Características más importantes del módulo de potencia

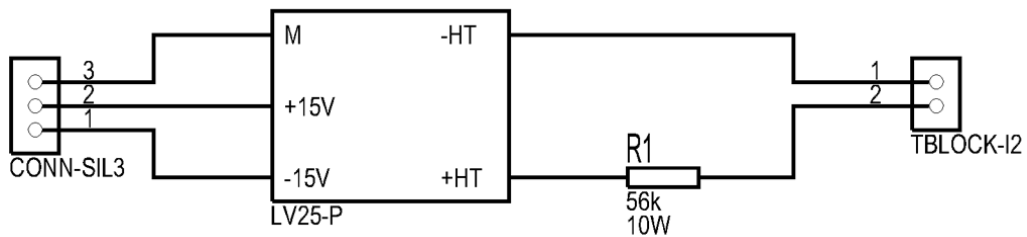


Figura 2.2: Circuito para la medida de la tensión  $V_{DC}$

Para medir la tensión del bus de continua se necesita hacer pasar una corriente por la célula LEM, esto se consigue mediante una resistencia de  $56\text{k}\Omega$  en serie con el positivo del bus. De este modo, a la tensión de funcionamiento ( $535\text{V}$ ) circula una corriente de  $9.55\text{mA}$  por el primario de la LEM, lo que se traduce en una corriente de  $23.8\text{mA}$  en el secundario del sensor al aplicarle una relación de 2.5 veces.

La máxima tensión que va medir la LEM se establece en  $670\text{V}$ . Para fijar este nivel de tensión se ha tenido en cuenta la tensión máxima del convertidor, que es de  $750\text{V}$  y también el margen necesario para la rama de disipación. Se

puede pasar a calcular la máxima corriente que puede llegar a suministrar la LEM antes de que se desconecte el convertidor a través de las ecuaciones 2.1 y 2.2.

$$I_{p_{max}} = \frac{V_{DC_{max}}}{R1} \quad (2.1)$$

$$I_{s_{max}} = I_{p_{max}} \cdot 2,5 \quad (2.2)$$

$$I_{s_{max}} = \frac{670}{56000} 2,5 = 29,91mA$$

A partir del cálculo se elige una resistencia de 10W y 5% de tolerancia, además de un bajo coeficiente térmico. Se intenta conseguir la mínima tolerancia, ya que el error en la medida debe ser lo más pequeño posible.

### Medida de corriente

El módulo de potencia incluye dos células LEM LAH 25-NP para la medida de la corriente de las fases U y V. Por lo tanto no se necesita hacer modificaciones en este aspecto, tan salo la asignación de pines.

### Drivers

A la hora de diseñar la placa de control se ha tenido muy en cuenta el modo de funcionamiento de los drivers incluidos en el módulo. Por ejemplo, algo que se ha observado durante la fase de diseño ha sido la necesidad de alimentar todos los drivers simultáneamente, ya que si solo se alimenta uno éste no trabaja.

También se han analizado las protecciones internas de los drivers, de este modo se ha implementado un control donde se puede distinguir entre un cortocircuito en una rama o un fallo por baja alimentación.

Los drivers disponen de sus propios tiempos muertos generados por



hardware y configurados en  $3\mu s$ . Por lo tanto, no se necesita volver a generarlos por software.

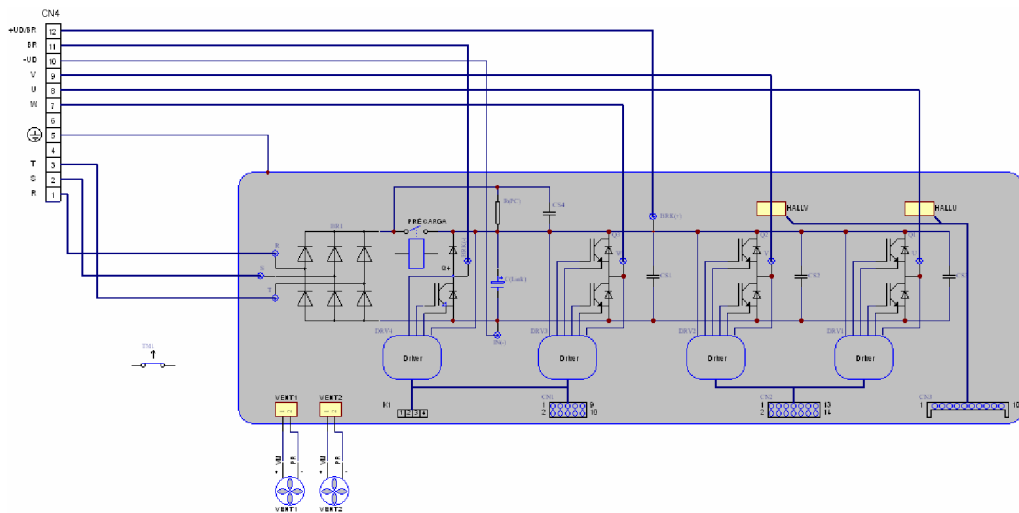


Figura 2.3: Esquema eléctrico del módulo de potencia

## Entradas y salidas

A continuación se detallan todas las entradas y salidas de las que dispone el módulo de potencia. A través de estas conexiones será por donde se comunique la placa de control.

En el conector CN1 se encuentran los pines de alimentación del driver de la fase W y los del driver utilizado en la rama de disipación. El resto de pines se utilizan para las órdenes de encendido/apagado y para las señales de estado del driver. En la tabla 2.2 se recogen las conexiones correspondientes al conector CN1.

El conector CN2 contiene los pines utilizados para la alimentación del driver de la fase U y del de la fase V. Además se encuentran los pines utilizados para recibir las órdenes de encendido/apagado y los pines de las

Pin	Símbolo	Descripción	Valores	Unidades
1	TOP W	Entrada High Fase W	0/15	V
2	ER W	Salida Error Fase W	0/15	V
3	BOT W	Entrada Low Fase W	0/15	V
4	GND	Masa	0	V
5	Vin (BRK)	Entrada Señal Freno	0/15	V
6	ER BRK	Salida Error Freno	0/15	V
7	+Vs	Alimentación Driver	15	V
8	+Vs	Alimentación Driver	15	V
9	GND	Masa	0	V
10	GND	Masa	0	V

Tabla 2.2: Asignación de pines del conector CN1

señales de estado de los drivers. En la tabla 2.3 se muestran la asignación de pines correspondientes al conector CN2.

Pin	Símbolo	Descripción	Valores	Unidades
1	TOP U	Entrada High Fase U	0/15	V
2	ER U	Salida Error Fase U	0/15	V
3	BOT U	Entrada Low Fase U	0/15	V
4	GND	Masa	0	V
5	TOP V	Entrada High Fase V	0/15	V
6	ER V	Salida Error Fase V	0/15	V
7	BOT V	Entrada Low Fase V	0/15	V
8	GND	Masa	0	V
9	+Vs	Alimentación Driver	15	V
10	+Vs	Alimentación Driver	15	V
11	GND	Masa	0	V
12	GND	Masa	0	V
13		NC		
14		NC		

Tabla 2.3: Asignación de pines del conector CN2

La tabla 2.4 muestra la asignación de pines del conector CN3 utilizados para el sensado de la corriente de las fases U y V. Este conector también se utiliza para la alimentación de ambos sensores.

Pin	Símbolo	Descripción	Valores	Unidades
1	+15V	Alimentación Positiva	15	V
2	-15V	Alimentación Negativa	-15	V
3	GND	Masa	0	V
4	HALL U	Salida Sensor Fase U		
5	HALL V	Salida Sensor Fase V		
6		NC		
7	GND	Masa	0	V
8	+15V	Alimentación Positiva	15	V
9	-15V	Alimentación Negativa	-15	V
10		NC		

Tabla 2.4: Asignación de pines del conector CN3

Existe un cuarto conector llamado K1 utilizado para alimentar los ventiladores de disipación y los relés que cortocircuitan las resistencias de precarga. Tan solo utiliza dos pines, el CIS (pin 2) y CIS\_GND (pin 3), que se conectarán a la placa de control.

Para las conexiones de potencia se ha diseñado un panel con conectores de seguridad para su utilización en el laboratorio. En los anexos aparece un plano del panel donde se puede ver la asignación y distribución de cada conector.



## 3. Microcontrolador

Este capítulo hace referencia a la parte digital del circuito, y en él se definen las características del microcontrolador seleccionado así como los requisitos que se deben de tener en cuenta a la hora de elegirlo. Se realizará una descripción de cada módulo junto con un pequeño ejemplo ilustrativo.

Un microcontrolador, DSP o dsPIC tienen distintos modos de funcionamiento y distintos módulos que pueden ser activados o no a elección del usuario. A su vez, los módulos que incorpora se configuran según las necesidades de cada aplicación.

Tal configuración se realiza mediante registros. En los registros se escriben *unos* y *ceros*, dependiendo de lo que se escriba y en que registro se escriba el dispositivo activará o no los módulos que trabajarán según la configuración elegida.

Para saber que valores poner en los registros, los fabricantes ponen a disposición el *DataSheet*, que explica con detalle los módulos del microcontrolador y los registros de configuración de cada módulo. Leer este documento es crítico, ya que se explican con detalle las características del dispositivo, lo cual permite ver si sirve para la aplicación a desarrollar.

La comunicación con el microcontrolador se realiza mediante las patillas de éste, de hecho la típica aplicación básica que consiste en imprimir por pantalla *Hello World!*, en microcontroladores lo primero y más básico es encender un LED conocido como *Blinking LED*. Cada patilla se puede configurar como entrada o salida, si se configura como salida, activará físicamente esa patilla con un nivel alto o uno bajo. Al configurarlo como entrada, se realizará una conversión analógica/digital del valor de tensión en la patilla y se guardará en un registro el cual podrá ser leído por un programa.

Hay distintos tipos de microcontroladores, algunos son de propósito general y otros están pensados para controlar motores. Por ejemplo, si se desea usar un solo motor en lazo abierto quizás con uno de propósito general sería suficiente sin embargo, para controlar distintos motores y en lazo cerrado sería interesante utilizar aquellos especializados en ello. Es importante saber elegir el dispositivo que mejor se adapta a las necesidades de la aplicación.

### 3.1. Características y elección del microcontrolador

Para la elección del microcontrolador se deben tener en cuenta los requisitos demandados por el módulo de potencia explicado en el capítulo anterior, así como entradas y salidas suficientes, velocidad de procesamiento, cantidad de periféricos y cuales de ellos dedicados a funciones específicas para el control de motores.

En la tabla 3.1 se lista una serie de microcontroladores que se han ido analizando y que no han cumplido los requisitos exigidos.

Microcontrolador descartado	Motivo por el que se descarta
Microchip PIC16F72	Uno de los microcontroladores más utilizados y fácil de programar pero que nos queda insuficiente tanto en prestaciones como en número de entradas y salidas
Microchip PIC18F4431	Se trata de un micro de 8-bits con módulo PWM y con suficientes entradas y salidas, pero que a la hora de realizar un control más complejo, donde se necesita de velocidad de procesamiento, se queda por debajo de lo exigido.
Texas Instruments TMS320LF2403A	Cuenta con pines de I/O insuficientes.
Texas Instruments TMS320LF2406A	De características muy similares al micro seleccionado tanto en número de pines, periféricos como prestaciones. Se descarta por el hecho de tener más conocimientos en el software de programación de los micros de Microchip.

Tabla 3.1: Microcontroladores descartados

Finalmente se ha seleccionado el **dsPIC33FJ256MC710A** de Microchip, se trata de un DSP de 16-bits y 100 pines, con altas prestaciones, tanto por cantidad de periféricos como por velocidad de procesamiento, entre cuyas características destacan:

- Gran velocidad, hasta 40 Millones de Instrucciones por Segundo (MIPS)
- Bajo consumo, alimentación a 3.3 V
- Opciones de reloj: externo, RC interno, uso de PLL, etc.
- Hasta 256 Kbytes de memoria flash de programa y 30 Kbytes de SDRAM
- 85 pines digitales programables I/O

Interrupciones:

- 67 fuentes de interrupción

- 5 interrupciones externas
- 7 niveles de prioridad programables

Periféricos:

- Módulo PWM de hasta 8 canales
- Módulo Quadrature Encoder Interface (QEI) para uso de encoder
- Módulos de comunicaciones: I<sup>2</sup>C<sup>TM</sup>, SPI, UART y CAN
- Módulos Analógico-Digital (ADC) de 10-bits o 12-bits de resolución
- Acceso directo a memoria (DMA)
- 9 Timers de 16-bits o posibilidad de 4 de 32-bits

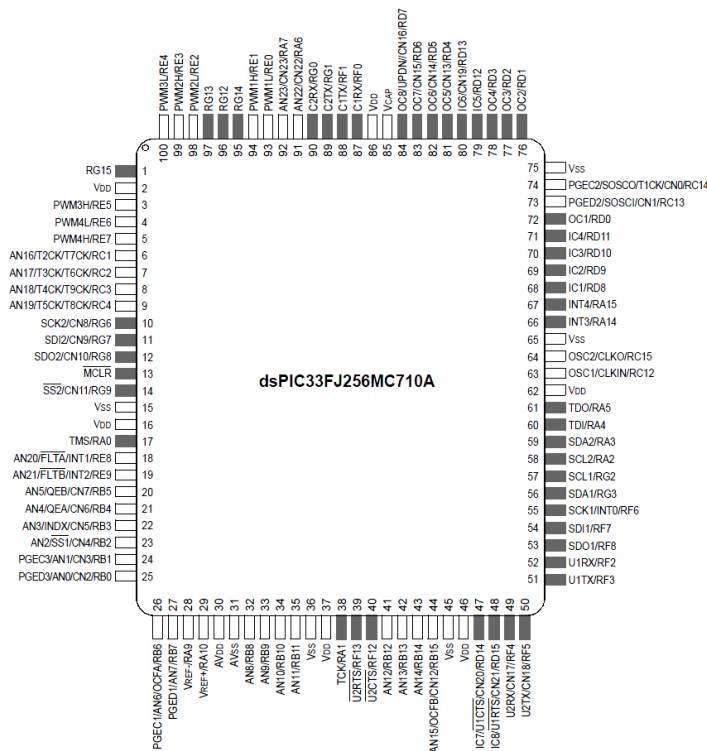


Figura 3.1: Diagrama de pines del dsPIC33FJ256MC710A



Básicamente un dsPIC es un potente microcontrolador de 16-bits al que se le han añadido las principales características de los DSP. Poseen todos los recursos de los mejores microcontroladores embebidos junto con las principales características de los DSP.

Ofrecen todo lo que se puede esperar de los microcontroladores: velocidad, potencia, manejo flexible de interrupciones, un amplio campo de funciones periféricas analógicas y digitales, opciones de reloj, simulación en tiempo real, etc.

Son capaces de procesar señales digitales rápidamente, por ejemplo trabajar con audio y video en tiempo real, siendo muy útiles en aplicaciones en las que no se permitan retrasos.

Los dsPIC, al igual que otros DSP usan la arquitectura Harvard que se caracteriza por la separación entre zonas de memoria y datos.

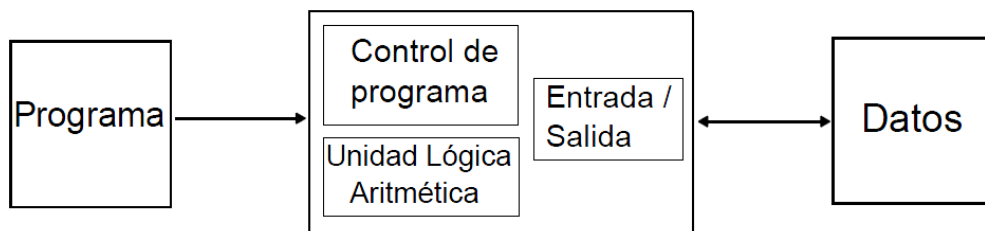


Figura 3.2: Arquitectura Harvard

La empresa que fabrica estos dispositivos pone al alcance de forma gratuita el entorno de desarrollo MPLAB IDE, esta herramienta proporciona un entorno sencillo y potente. Gestiona todas las partes en las que se descompone un proyecto, proporciona un completo editor y un potente depurador a nivel del lenguaje fuente, en nuestro caso C.

El integrado posee distintos módulos que permiten el control de motores, uno de ellos el módulo PWM que se encarga de generar las señales de

encendido y apagado de los distintos IGBTs. Otro módulo utilizado es el QEI, encargado de interactuar con el encoder obteniendo de este datos de la posición, velocidad e incluso aceleración de la máquina.

La configuración de los distintos módulos así como la del oscilador se definen a continuación.

## 3.2. El Oscilador

El sistema oscilador del dsPIC33FJ256MT710A dispone de 9 opciones distintas para conseguir la frecuencia de trabajo deseada. Se va a trabajar a una frecuencia de 40 MIPS que se consigue a partir de un cristal de cuarzo externo resonando a 8 MHz. Por lo tanto, se configura el micro en modo XT con PLL (Phase Locked Loop)<sup>1</sup> para aumentar la frecuencia proporcionada por el cristal hasta los 80 MHz.

La frecuencia de operación ( $F_{CY}$ ) viene dada por la siguiente ecuación:

$$F_{CY} = \frac{F_{OSC}}{2} \quad (3.1)$$

Donde  $F_{OSC}$  se determina a partir de la configuración del PLL:

$$F_{OSC} = F_{IN} \cdot \frac{M}{N1 \cdot N2} \quad (3.2)$$

---

<sup>1</sup>Diagrama de bloques en la figura 3.3

Para conseguir una  $F_{CY}$  de 40 MIPS tenemos que ajustar los valores de M, N1 y N2:

$$M = 40$$

$$N1 = 2$$

$$N2 = 2$$

$$F_{CY} = \frac{F_{OSC}}{2} = \frac{1}{2} \cdot \frac{8000000 \cdot 40}{2 \cdot 2} = 40 \text{ MIPS} \quad (3.3)$$

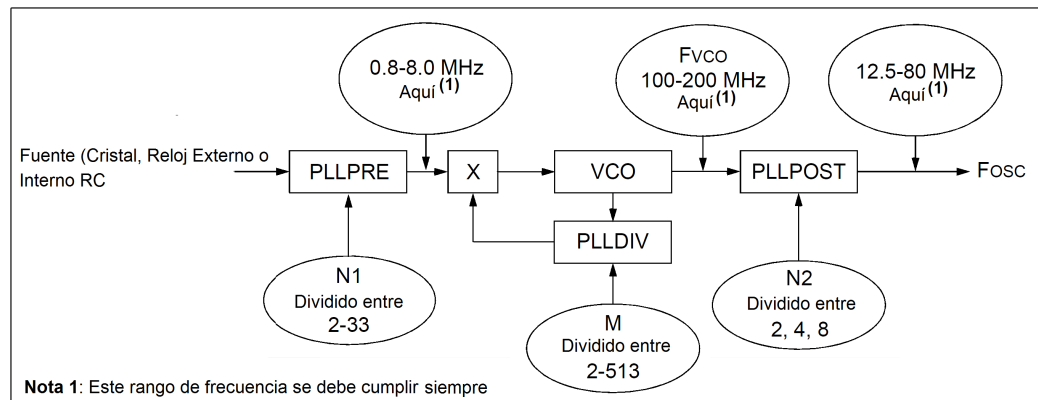


Figura 3.3: Diagrama de bloques del PLL

La configuración de cada registro del oscilador la encontraremos en el *Apéndice B.1: Registros de configuración*.

### 3.3. Conversor analógico-digital (ADC)

La conversión analógica-digital consiste en transformar señales analógicas en señales digitales, para poder procesarlas de forma más sencilla. Las señales

digitales se pueden procesar en computadores, lo cual implica la posibilidad de aplicar filtros para mejorar la señal o eliminar el ruido.

Convertir una señal analógica en digital consiste en obtener un valor de la señal para cada instante de tiempo, a esto se le llama muestreo, y al tiempo entre la toma de muestras, periodo de muestreo.

El dsPIC33FJ256MC710A dispone de dos módulos ADC de hasta 32 entradas analógicas repartidas en 4 canales. Cada módulo puede ser configurado como ADC de 10-bits y muestrear 4 canales simultáneamente o ADC de 12-bits y muestrear tan solo un canal pero a mayor resolución.

Si se opta por la opción de muestrear varios canales simultáneamente se debe utilizar el modo de acceso directo a memoria (DMA), ya que el módulo ADC almacena los datos muestreados en un buffer que tendremos que leer a través de la DMA.

En el ejemplo que hemos realizado para leer las dos corrientes, tensión del bus de continua y el valor del potenciómetro se ha elegido la configuración de 12-bits, leyendo una entrada tras otra, ya que el tiempo requerido para el muestreo y la conversión de cada señal es ínfimo y de este modo conseguimos mayor resolución en la medida, además de no necesitar configurar el módulo DMA.

En el Apéndice B se encuentran recogidos los registros de configuración e inicialización del módulo ADC.

El siguiente ejemplo muestra el uso del módulo ADC, se ejecuta al finalizar cada muestreo-conversión mediante la activación del flag de interrupción del ADC.

```
// Interrupción por ADC

void __attribute__((__interrupt__)) _ADC1Interrupt (void
){

switch(entrada){
// Leo una entrada analógica por cada interrupción

case 0:
    i_med_u = ADC1BUF0;
    // Almaceno el valor de la corriente fase u
    iu = (((i_med_u+73)*3.3)/4095)-1.65)/0.122;
    // Escalo corriente y compenso error
    if (iu_max < iu) iu_max = iu;
    // Me quedo con el valor pico
    iu_rms = iu_max/sqrt(2);
    // Calculo el valor eficaz
    iuf_rms = iu_rms*0.005+iuf_rms*0.995;
    // Filtro la corriente
    iu_max = iu_max * 0.999;
    // Voy disminuyendo el valor máximo almacenado
    AD1CON1bits.ADON = 0;
    // Paro el módulo ADC
    AD1CSSL = 0x0002;
    // Cambio el canal a muestrear a AN1
    AD1CON1bits.ADON = 1;
    // Inicio el módulo ADC
    break;

case 1:
    i_med_v = ADC1BUF0;
    // Almaceno el valor de la corriente fase v
    iv = (((i_med_v+12)*3.3)/4095)-1.65)/0.122;
```

```
// Escalo corriente y compenso error
if (iv_max < iv) iv_max = iv;
iv_rms = iv_max/sqrt(2);
ivf_rms = iv_rms*0.005+ivf_rms*0.995;
iv_max = iv_max * 0.999;
AD1CON1bits.ADON = 0;
// Paro el módulo ADC
AD1CSSL = 0x0004;
// Cambio el canal a muestrear a AN2
AD1CON1bits.ADON = 1;
// Inicio el módulo ADC
break;

case 2:
vdc_med = ADC1BUF0;
// Almaceno el valor de la tensión de bus
vdc = ((vdc_med-50) * 0.1640625);
// Escalo y compenso el error
vdcf = vdc*0.01+vdcf*0.99;
// Filtro la tensión de bus
AD1CON1bits.ADON = 0;
// Paro el módulo ADC
AD1CSSL = 0x0100;
// Cambio el canal a muestrear a AN8
AD1CON1bits.ADON = 1;
// Inicio el módulo ADC
break;

case 3:
pot_med = ADC1BUF0;
// Almaceno el valor del potenciómetro
pot_medf = pot_med*0.01+pot_medf*0.99;
// Filtro el potenciómetro
AD1CON1bits.ADON = 0;
// Paro el módulo ADC
```

```
        AD1CSSL = 0x0001;
        // Cambio el canal a muestrear a AN0
        AD1CON1bits.ADON = 1;
        // Inicio el módulo ADC
        break;
default:
        break;
}

        entrada++;
        // Una vez leídos los 4 canales vuelvo a
        // seleccionar el canal AN0
        if (entrada>3) entrada=0;

IFS0bits.AD1IF = 0;
// Borro el flag de interrupción
return;
}
```

Cada vez que salta la interrupción se escalan los valores muestreados, se corrige el pequeño error que puede existir y se aplica un filtrado digital. Finalmente se almacena en una variable que usaremos posteriormente en otras partes del programa principal.

Por cada entrada de la interrupción se va incrementado el valor del contador entrada, de este modo podemos leer el canal seleccionado en la interrupción anterior y cambiarlo para cuando salte la siguiente.

## 3.4. Módulo PWM

El módulo PWM se encarga de generar una señal con forma de pulso periódico. El período y el ciclo de trabajo de los pulsos son programables. El

ciclo de trabajo se va actualizando en función al valor obtenido de la tabla del seno.

Se utiliza el modo *Center-Aligned* del módulo PWM, de este modo se encienden simultáneamente tres de los seis IGBTs, uno de cada rama, reduciendo considerablemente los ruidos producidos por las conmutaciones.

La Figura 3.4 muestra el diagrama de tiempos del modo *Center-Aligned*.

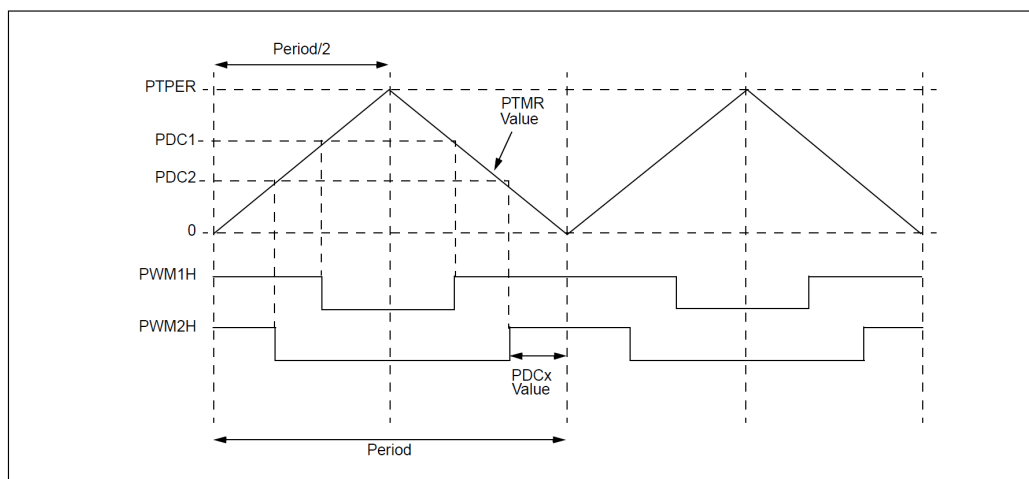


Figura 3.4: Diagrama de tiempos del modo *Center-Aligned*

El módulo PWM también nos permite establecer tiempos muertos entre cada una de las dos señales complementadas, tal y como se puede observar en la Figura 3.5, de todos modos en nuestro caso no lo vamos a necesitar debido a que los propios drivers generan dichos tiempos muertos por hardware.



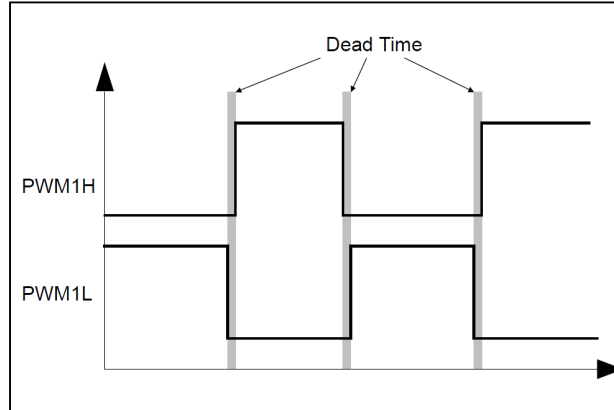


Figura 3.5: Tiempos muertos

La frecuencia de la PWM se establece en 20 kHz para evitar que se escuchen ruidos derivados de la conmutación, debido a que a partir de esta frecuencia se hace imperceptible para el oído humano.

La ecuación 3.4 se utiliza para determinar el periodo que se debe introducir en el registro PTPER a partir de la frecuencia de conmutación elegida, en este caso 20 kHz.

$$PTPER = \frac{F_{CY}}{F_{PWM}} - 1 \quad (3.4)$$

Siendo  $F_{CY}$  la frecuencia de operación (40 MIPS) y  $F_{PWM}$  (20 kHz) la frecuencia de conmutación. La configuración de este y de otros registros utilizados para inicializar el módulo PWM se encuentran recogidos en el Apéndice D.

Para generar las 3 salidas trifásicas se utiliza una tabla de 64 valores, que contiene los valores del seno entre -1 y 1 tabulados en un registro de 16-bits. Por lo tanto por cada valor decimal del seno le corresponde un valor entero contenido en la tabla.

La variable *Angulo* se utiliza como puntero de 16-bits de la tabla de valores del seno, donde 0x0000 representa un ángulo de 0°, y 0xFFFF representa un ángulo de 359.99°. Por cada periodo de la PWM (50  $\mu$ s a 20 kHz) se genera una interrupción que incrementa el valor de *angulo* según *\_DELTA*. El valor de *\_DELTA* determina como de rápido se desplaza el código a través de la tabla del seno, obteniendo como resultado la frecuencia de modulación.

La figura 3.6 muestra las 6 salidas PWM generadas por el módulo PWM a partir de los valores de la tabla del seno, se puede observar de forma superpuesta la tensión media que ven las bobinas del motor a partir de las señales PWM.

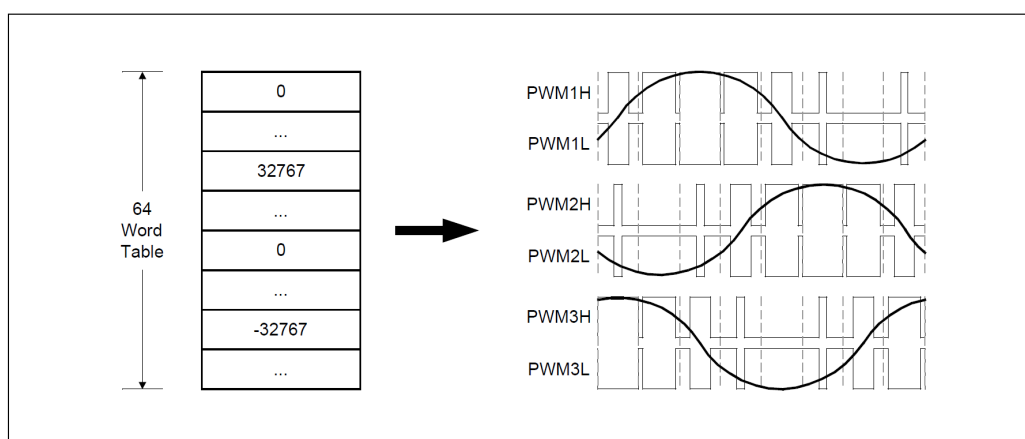


Figura 3.6: PWM generada a partir de la tabla del seno

El valor de *\_DELTA* se obtiene a partir de la expresión 3.5. Una vez calculado el valor de *Angulo*, se le suma un offset de 0x5555 equivalente a 120°, y otro de 0xA444 equivalente a 240°, para conseguir desfase las dos fases restantes.

$$_DELTA = 2^{16} \cdot \frac{frec(Hz)}{F_{PWM}} \quad (3.5)$$

Siendo  $frec$  la frecuencia deseada y FPWM la frecuencia de conmutación.

Posteriormente, los 3 punteros de 16-bits se desplazan hacia la derecha 10 bits para quedarnos con los 6 bits más significativos. Solo necesitamos los 6 más significativos debido a que la tabla del seno la forman 64 valores, para otros tamaños de la tabla necesitaríamos diferente cantidad de bits.

Una vez obtenido el valor de la tabla necesitamos escalarlo, inicialmente se escala el valor de la amplitud y posteriormente el de la frecuencia de modulación. Debido a que los valores obtenidos de la tabla están comprendidos entre 0x7FFF y 0x8000, y los del ciclo de trabajo entre 0 y 2 veces PTPER, se hace necesario añadirle al resultado el valor de PTPER.

En la figura 3.7 se puede observar lo explicado.

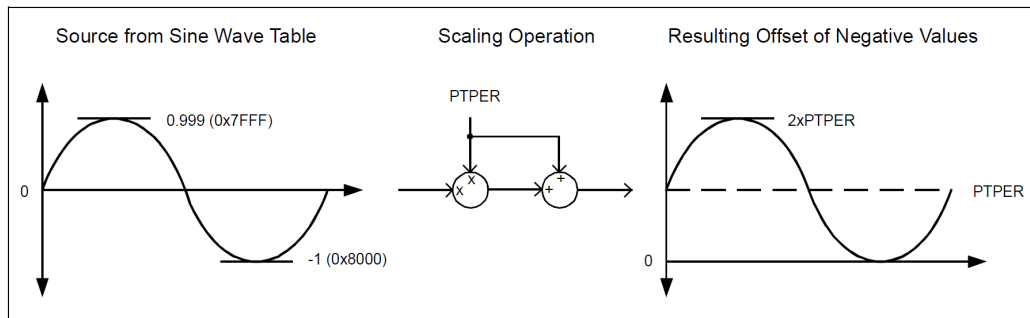


Figura 3.7: Escalado de los ciclos de trabajo

El siguiente fragmento de código muestra como se actualizan los 3 ciclos de trabajo a partir del valor obtenido de la tabla del seno.

```
// Tabla con 64 valores de la función seno escalados a
// números enteros de 16-bits

const int TablaSeno [64] = {
    0,3212,6393,9512,12539,15446,18204,20787,23170,
    25329,27245,28898,30273,31356,32137,32609,32767,
    32609,32137,31356,30273,28898,27245,25329,23170,
    20787,18204,15446,12539,9512,6393,3212,0,-3212,
    -6393,-9512,-12539,-15446,-18204,-20787,-23170,
    -25329,-27245,-28898,-30273,-31356,-32137,-32609,
    -32767,-32609,-32137,-31356,-30273,-28898,-27245,
    -25329,-23170,-20787,-18204,-15446,-12539,-9512,
    -6393,-3212};

// Interrupcion por PWM

void __attribute__((__interrupt__))_PWMInterrupt(void){

    PWMCON2bits.UDIS = 1;
    // Inactivo las salidas PWM
    Angulo += _DELTA;
    // Voy aumentando el ángulo según _DELTA

    //Actualizo CicloTrabajo 1
    ValorSeno = TablaSeno[Angulo>>10];
    // Obtengo el valor del seno
    TPWM = PTPER*2;
    ValorSeno = ((long)ValorSeno*(long)(_TENSION))
        >>16;
    ValorSeno = ((long)ValorSeno*(long)TPWM)>>16;
    // Elimino los valores negativos del ciclo de
    // trabajo
```

```
PDC1 = ValorSeno + PTPER;

//Actualizo CicloTrabajo 2
ValorSeno = TablaSeno[(Angulo + _120_GRADOS)
    >>10];
// Obtengo el valor del seno
TPWM = PTPER*2;
ValorSeno = ((long)ValorSeno*(long)(_TENSION))
    >>16;
ValorSeno = ((long)ValorSeno*(long)TPWM)>>16;
// Elimino los valores negativos del ciclo de
    trabajo
PDC2 = ValorSeno + PTPER;

//Actualizo CicloTrabajo 3
ValorSeno = TablaSeno[(Angulo + _240_GRADOS)
    >>10];
//Obtengo el valor del seno
TPWM = PTPER*2;
ValorSeno = ((long)ValorSeno*(long)(_TENSION))
    >>16;
ValorSeno = ((long)ValorSeno*(long)TPWM)>>16;
// Elimino los valores negativos del ciclo de
    trabajo
PDC3 = ValorSeno + PTPER;

PWMCON2bits.UDIS = 0;
// Vuelvo activar salidas PWM
IFS3bits.PWMIF = 0;
// Borro el flag de interrupción
}
```

### 3.5. Módulo QEI

El módulo QEI proporciona una interfaz para encoders incrementales que permite obtener datos sobre la posición del eje. Los encoders incrementales o de cuadratura detectan la posición de sistemas en movimiento de rotación, y a partir de la posición podemos calcular su velocidad angular.

Un típico encoder de cuadratura incluye una rueda ranurada unida al eje del motor y un módulo basado en optoacopladores que detecta las ranuras de la rueda. La información se emite normalmente por tres o seis canales de salida, Fase A (QEA), Fase B (QEB) e Index (INDX), en el caso de seis canales, tres de ellos son complementarios a los otros tres, por lo que finalmente al módulo QEI tan solo le llegan tres. Las señales proporcionadas son digitales.

Los canales Fase A y Fase B tienen una relación única. Si la Fase A va por delante de la Fase B, la dirección del motor se considera positiva o hacia delante. Si la Fase B va por delante de la Fase A, se considera negativa o hacia atrás. El pulso que incrementa el Index ocurre una vez por revolución y es usado como referencia para indicar la posición absoluta. Se observa un diagrama de tiempos relativo a éstas tres señales en la figura 3.8

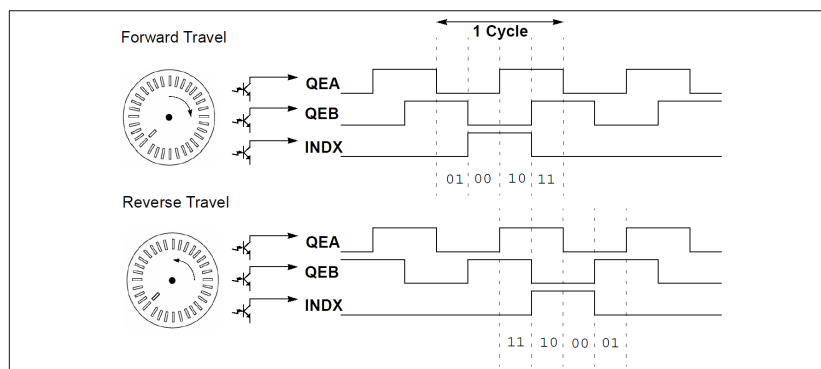


Figura 3.8: Diagrama de tiempos del módulo QEI

## Filtrado digital

El módulo QEI incluye la opción de aplicar un filtrado digital a las señales procedentes del encoder para evitar posibles interferencias. El diagrama de tiempos de la figura 3.9 muestra como las señales de entrada son filtradas.

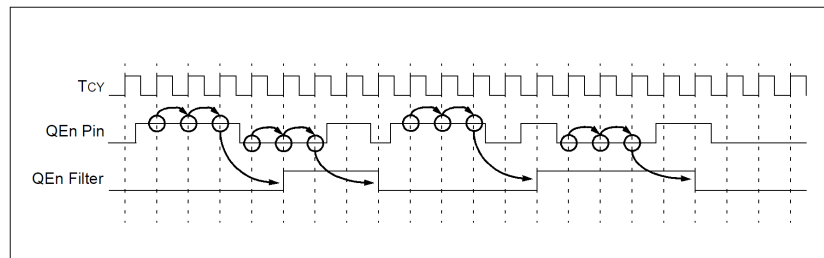


Figura 3.9: Diagrama de tiempos del filtro

Para determinar los valores del filtro se debe tener en cuenta el ancho mínimo de los pulsos provenientes del encoder, los cuales se determinan a partir de la velocidad máxima del motor y de la resolución del encoder. La ecuación 3.6 muestra el cálculo del ancho mínimo de los pulsos.

$$PULSO\_MIN = \frac{30}{MAX\_RPM \cdot PULSOS\_ENCODER} \quad (3.6)$$

Se debe configurar el filtro para que rechace cualquier pulso menor al valor obtenido en la ecuación 3.6. Para ello debemos calcular el periodo de filtrado tal como se muestra en la ecuación 3.7 que posteriormente será utilizado para configurar el registro QECK. El registro QECK no admite cualquier valor, siempre se escogerá uno inferior al obtenido a partir de la ecuación. En el Apéndice E se explica la configuración de cada registro.

$$FILTER\_DIV = \frac{FCY \cdot PULSO\_MIN}{3} \quad (3.7)$$

El módulo QEI permite aumentar hasta 4 veces la resolución del encoder, lo consigue incrementando o decrementando el registro POSCNT por cada flanco de subida y de bajada de las señales QEA y QEB. La figura 3.10 muestra el diagrama de tiempos correspondiente a la configuración x4.

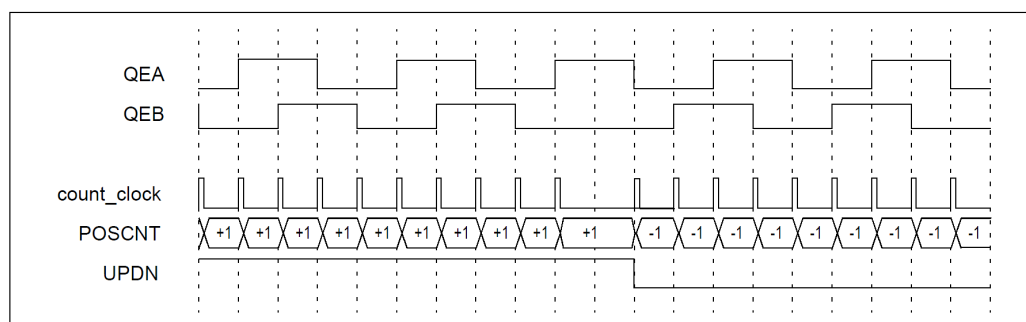


Figura 3.10: Diagrama de tiempos del QEI en modo x4

### Reseteo del contador

El registro POSCNT se puede resetear cada vez que se recibe un pulso por el pin Index. Algunos encoders no disponen de esta salida para obtener referencias absolutas de posición, en estos casos se puede utilizar el registro MAXCNT e inicializarlo en el valor que deseemos para resetear POSCNT cuando alcance dicho valor. En la aplicación programada utilizamos el reset por pulso de Index ya que el encoder si dispone de esta salida.

### Cálculo de la velocidad angular

Para determinar la velocidad angular se necesita conocer el número de pulsos generados en un intervalo de tiempo, este intervalo se establece mediante el uso de un Timer. El tiempo que tarda en saltar la interrupción debe ser menor al tiempo requerido para dar media vuelta a máxima velocidad, según la ecuación 3.8



$$PERIODO\_INTERRUPCION = \frac{60}{2} \cdot MAX\_SPEED\_RPM \quad (3.8)$$

Una vez conocido el intervalo de tiempo se debe configurar el Timer para que genere una interrupción. En el Apéndice F se muestra un ejemplo de inicialización del Timer así como la configuración de cada registro.

### Ejemplo de código

```
#define MAX_RPM 2400
// Velocidad máxima del motor [RPM]
#define MAX_CNT_PER_REV (1024 * 4 - 1)
// Número de pasos por vuelta (pasos del encoder * modo
// x4 - 1)
#define MAXSPEED (unsigned int)(((unsigned long)
    MAX_CNT_PER_REV*2048)/256)
// Velocidad máxima escalada a enteros de 16-bits
#define HALFMXSPEED (MAXSPEED>>1)
// La mitad de MAXSPEED

// Interrupción por Timer 2 para el cálculo de la
// velocidad angular

void __attribute__((__interrupt__, auto_psv))
    _T2Interrupt(void) {

    POSCNTcopy = (int)POSCNT;
    // Guardo el valor del contador en una variable
    // Si gira hacia la izquierda le restamos al valor de
    // una vuelta completa el valor del contador para que
    // nos quede positivo
```

```
if (!(QEICON & 0x0800)) POSCNTcopy = 65536 - POSCNTcopy;

AngPos[1] = AngPos[0];
AngPos[0] = (unsigned int)(((unsigned long)POSCNTcopy *
    2048)/256);
// Escalo el valor de POSCNTcopy a una variable de 16-
// bits y lo guardo en AngPos[0]
// 0 <= POSCNTcopy <= 4095
// 0 = AngPos <= 32760

Speed = AngPos[0] - AngPos[1];
// Diferencia de pulsos entre la interrupción actual y
// la anterior
if (Speed >= 0){
    if (Speed >= (HALFMAXSPEED))
        Speed = Speed - MAXSPEED;
}
else{
    if (Speed < -(HALFMAXSPEED))
        Speed = Speed + MAXSPEED;
}
Speed *= 8;
Speedf = 0.02*Speed+0.98*Speedf;
// Filtro para la medida de velocidad

IFS0bits.T2IF = 0;
// Borro el flag de interrupción
return;
}
```

## 4. Placa de Control

Una vez analizada la etapa de potencia y comentado las posibilidades que ofrecen los microcontroladores, se pasa a explicar el diseño de la placa de control que servirá como plataforma para el desarrollo de futuros proyectos en el ámbito de los accionamientos eléctricos.

### 4.1. dsPIC33FJ256MC710A

El microcontrolador dsPIC33FJ256MC710A de Microchip podría decirse que es el cerebro de la placa de control. En el *Capítulo 3: Microcontrolador* se han explicado sus características más importantes así como los módulos utilizados en este PFC.

Para tener una idea global se explica brevemente a través del gráfico de la figura 4.1 las señales de entrada y de salida que llegan al microcontrolador. En rojo se representan las señales analógicas y en azul las digitales. Las señales verdes son digitales del tipo entrada-salida.

Para ver una relación detallada de cada pin del microcontrolador se puede recurrir al *Esquema 01: Microcontrolador dsPIC* del *Anexo: Esquemas de la Placa de Control*.

Debido a la cantidad de señales que llegan y salen del micro, se elige colocarlo en una posición central dentro de la placa. Viene montado en módulo llamado *Plug-In Module* (PIM) que permite utilizar un zócalo donde se inserta el micro. De este modo, se puede quitar y poner cuando se desee.

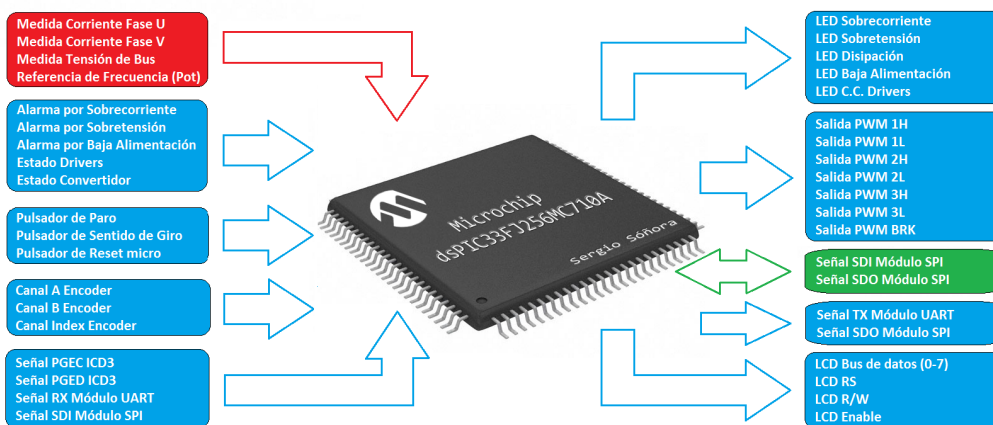


Figura 4.1: Entradas y salidas del microcontrolador

El microcontrolador necesita de una fuente de reloj externa para conseguir su frecuencia de trabajo. Este reloj recibe el nombre de oscilador, utiliza un cristal de cuarzo que vibra u oscila a una frecuencia fija, en este caso 8 MHz.

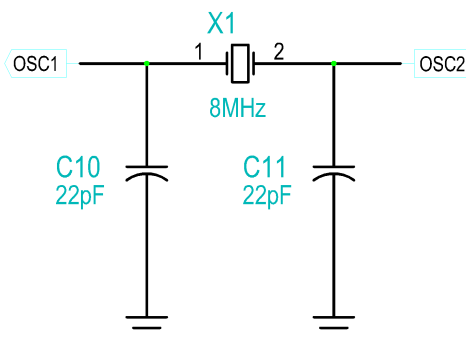


Figura 4.2: Circuito oscilador

En la figura 4.2 se muestra el circuito oscilador, además del cristal se colocan dos condensadores cerámicos de 22pF, tal y como indica la hoja de características del microcontrolador.

### Puerto de programación

A la placa de control se le ha dotado de un puerto para poder programar el dsPIC. Se utiliza el MPLAB ICD3 para realizar la programación sin necesidad de quitar el micro de su zócalo. Además de programarlo mediante el ICD3, se puede hacer uso del modo *debugger*, de este modo se puede probar el código paso a paso para depurarlo, facilitando mucho la tarea al programador.

En la figura 4.3 aparece el circuito utilizado para realizar la programación. El conector escogido para esta aplicación es un RJ11, el clásico conector telefónico.

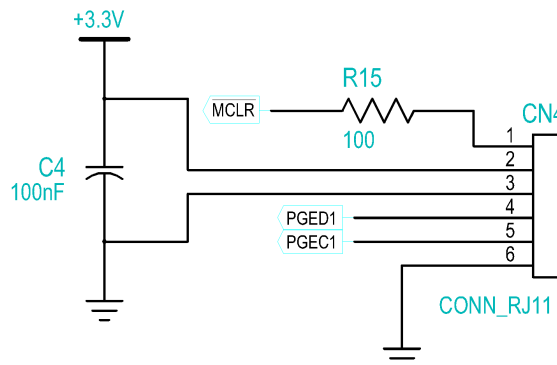


Figura 4.3: Circuito para la programación del dsPIC

### Comunicación Serial RS-232

El microcontrolador dispone de un módulo UART para comunicación serie RS-232. De este modo se puede enviar y recibir información hacia o desde un ordenador, por lo tanto se ha pensado en incluir un circuito para hacer

uso de esta interfaz. En la figura 4.4 se puede ver el esquema del circuito implementado.

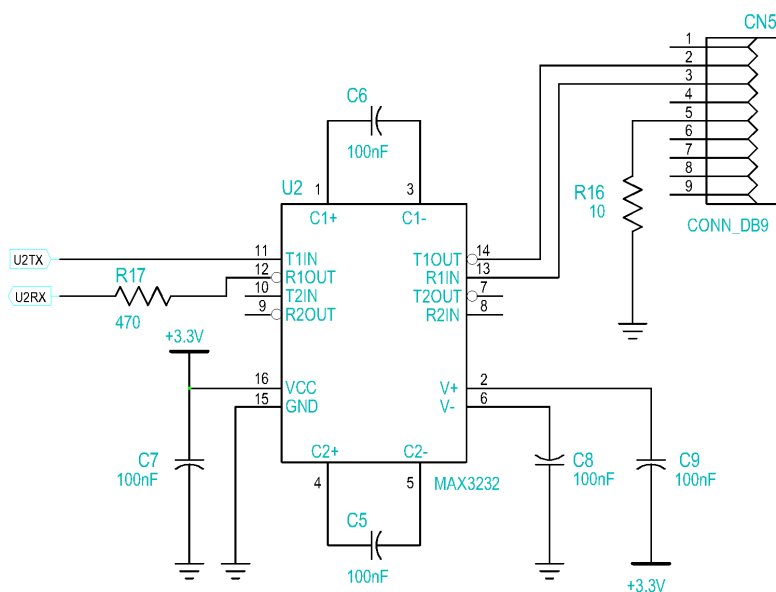


Figura 4.4: Circuito de adaptación de las señales RS-232

Se utiliza el circuito integrado MAX3232 de Maxim que convierte las señales del puerto serie RS-232 a señales compatibles con los niveles lógicos del microcontrolador. El MAX3232 sirve como interfaz de transmisión y recepción para las señales RX y TX.

El circuito integrado dispone de salidas para manejar los niveles de tensión del estándar RS-232 (aprox.  $\pm 7.5V$ ) generadas a partir de su tensión de alimentación (3.3V). Tan solo necesita de cuatro condensadores cerámicos de 100nF, más otro a la entrada de alimentación, para trabajar con el puerto RS-232. En el programa actual del microcontrolador no se hace uso de esta interfaz, pero queda a disposición para su uso en el desarrollo de futuros proyectos utilizando la placa de control diseñada en este PFC.

## Comunicación SPI

El SPI es un bus full duplex, síncrono y serial desarrollado por Motorola. También puede encontrarse con el nombre de Microwire, propiedad de National SemiConductors.

Al ser un bus serial su número de hilos es reducido en comparación a los buses paralelos. Si bien estos últimos son más rápidos, la eficiencia y simplicidad que un bus serial puede ofrecer hace que esta diferencia de velocidad sea menos importante.

Se deja abierta la posibilidad de comunicarse con el microcontrolador a través del bus SPI. De este modo, es posible enviar y recibir información con otro microcontrolador o módulo de expansión dotado con este mismo bus.

Por ejemplo, sería posible conectar un módulo bluetooth a través del SPI y controlar el convertidor mediante un Smartphone o teléfono móvil, recibiendo información en tiempo real.

Para poder utilizarlo se incluye un conector macho de 4 pines en la placa de control que permite el acceso a las cuatro señales utilizadas en la comunicación SPI, tal y como se muestra en la figura 4.5.

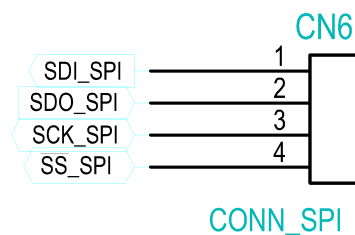


Figura 4.5: Conector SPI

## 4.2. Lógica de encendido y apagado

Para encender y apagar el convertidor existen dos señales digitales comandadas por dos pulsadores, uno de MARCHA para el encendido y otro PARO para el apagado. Si MARCHA=1 el convertidor se enciende y si PARO=1 el convertidor se apaga. Sin embargo existen otros mecanismos que apagan el convertidor ante cualquier alarma o fallo, desactivando las salidas PWM.

En cuanto se conecta el convertidor a la red este no se enciende de forma automática aunque el micro si lo haga. Los drivers tienen una protección por baja alimentación que genera una señal de error mientras la alimentación no supera cierto umbral, de este modo nunca arrancará el convertidor sin dar la orden de encendido mediante el pulsador, ya que la tensión de alimentación tarda unos milisegundos en superar el umbral de protección.

Los mecanismos que pueden apagar el convertidor ante una alarma son las protecciones de **sobrecorriente**, **sobretensión** y las señales de **estado de los drivers**. Mediante las señales de estado de los drivers es posible detectar un fallo por baja alimentación, ya que estos drivers disponen de un circuito de monitorización de la tensión de alimentación. Por lo tanto, si existe un error generalizado que afecta a todos los drivers de forma simultánea, se entiende que se trata de un fallo por baja alimentación y no por un cortocircuito en los IGBTs.

La figura 4.6 muestra el circuito utilizado para detectar el tipo de error del driver. Las salidas de error de los drivers son de colector abierto, se les debe de colocar una resistencia en configuración *Pull-Up* a +15V. Si existe un error, la señal proveniente de los drivers se pone a GND, en condiciones normales se tiene un 1 a través de las resistencias *Pull-Up*.



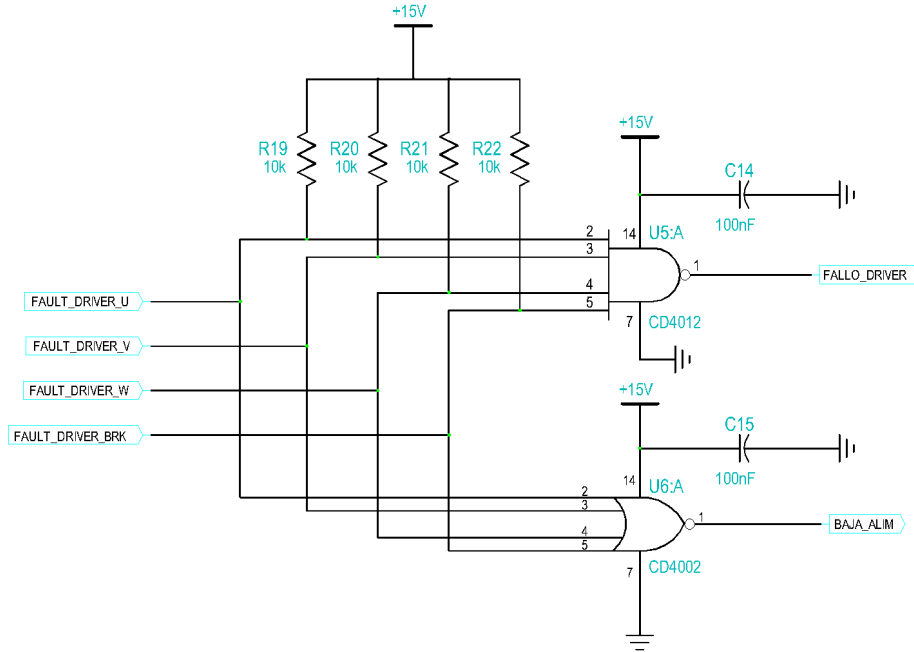


Figura 4.6: Circuito lógico para la detección del tipo de fallo de los drivers

Para detectar el tipo de error, se lleva cada señal a una puerta NOR cuádruple y a otra NAND también cuádruple. De este modo, si cualquiera de las señales de error del driver se pone a cero, en la salida de la puerta NAND se obtiene un “uno”, y en la salida de la puerta NOR tan solo se obtiene un “uno” si todas las señales muestran un “cero”. Explicado así puede resultar lioso, en la tabla 4.1 se muestra la tabla de verdad a partir de las señales lógicas de estado de drivers.

DRIVER_U	DRIVER_V	DRIVER_W	DRIVER_BRK	FALLO_DRIVER	BAJA_ALIM
0	0	0	0	0	0
X	X	X	1	1	0
X	X	1	X	1	0
X	1	X	X	1	0
1	X	X	X	1	0
1	1	1	1	1	1

Tabla 4.1: Tabla de verdad de las señales de estado de los drivers

Donde  $X$  es condición indiferente. En las señales de fallo de driver un “cero” es error mientras que un “uno” indica que no hay alarmas en ese driver.

Se ve claramente como la señal de baja alimentación solo se produce en el caso que fallen todos los drivers. Estas señales se acondicionan posteriormente y se llevan al microcontrolador. La señal de FALLO\_DRIVER se conduce también a una puerta OR, que en caso de existir error, reseteará el biestable tipo  $D$  causando el apagado del convertidor sin intervención del micro y lo mantendrá en ese estado.

En la figura 4.7 se puede ver todas las señales que llegan a la puerta OR 4072, si cualquiera de ellas presenta un nivel alto la puerta generará un “uno” a su salida reseteando el biestable tipo  $D$ . D7 y D8 evitan un cortocircuito entre las salidas de las dos puertas, se utiliza un diodo 1N4148 de propósito general.

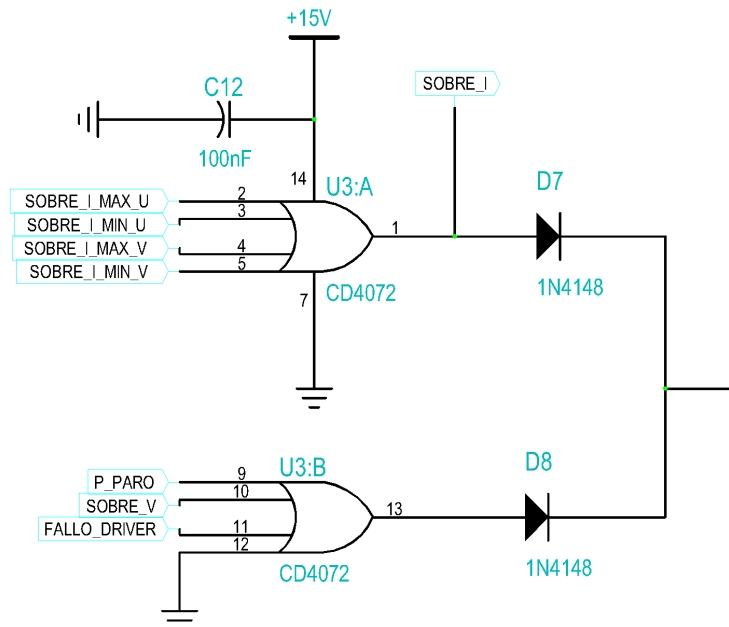


Figura 4.7: Circuito lógico que da lugar al apagado del convertidor.

Como se ha comentado anteriormente, todas las señales de error confluyen en el integrado 4013 formado por dos biestables o *flip-flop* tipo *D*. Las salidas de las puertas lógicas 4072 están conectadas a uno de los dos biestables del integrado, de tal forma que un nivel alto resetea dicho biestable provocando el apagado del convertidor y manteniéndolo en ese estado.

La configuración del biestable debe cumplir los siguientes requisitos:

- En el momento de conectar el convertidor a la red no se debe visualizar el encendido del convertidor, ni por su puesto encenderlo.
- Una vez conectado, y pulsando el botón de MARCHA, las señales de conmutación del micro deben llegar a los drivers y visualizar el estado de encendido a través del LED colocado para tal propósito, siempre y cuando no exista algún error activado.
- Si se activa cualquier error o se pulsa el botón de PARO, el convertidor debe apagarse desconectando las señales PWM de los drivers.
- Si se pulsa MARCHA mientras existe un error, o al mismo tiempo se pulsa PARO, el convertidor debe mantenerse apagado.
- Si se pulsa MARCHA en ausencia de errores, el convertidor vuelve a funcionar.

Los requisitos anteriores se pueden ver reflejados en la tabla de verdad 4.2. Para conseguir la desconexión de las señales PWM de los drivers se utilizará el integrado 4066, el cual se explica en el apartado 4.3: *Señales PWM*.

ENTRADAS				SALIDAS	
CLOCK (CLK)	DATA (D)	RESET (R)	SET (S)	Q	$\bar{Q}$
$\frac{\_}{\_}$	1	0	0	1	0
$\frac{\_}{\_}$	X	0	0	Q	$\bar{Q}$
X	X	1	0	0	1

Tabla 4.2: Tabla de verdad del biestable tipo D

Donde  $X$  representa la condición de indiferencia.

Los errores obtenidos a la salida de las dos puertas OR se llevan al RESET del biestable. La resistencia R18 asegura un “cero” lógico ante un estado bajo, además de asegurar el bloqueo de los diodos cuando entran en conducción. La señal P\_MARCHA enviada por el pulsador de MARCHA se manda a CLK.

$Q$  corresponde con la señal de ENCENDIDO que será la encargada de conmutar las señales PWM hacia los drivers y de iluminar un LED que indica el encendido del convertidor.  $\bar{Q}$  es la señal de APAGADO, se envía al microcontrolador para que este esté informado del estado del convertidor. En la figura 4.8 se muestra la configuración descrita del biestable 4013.

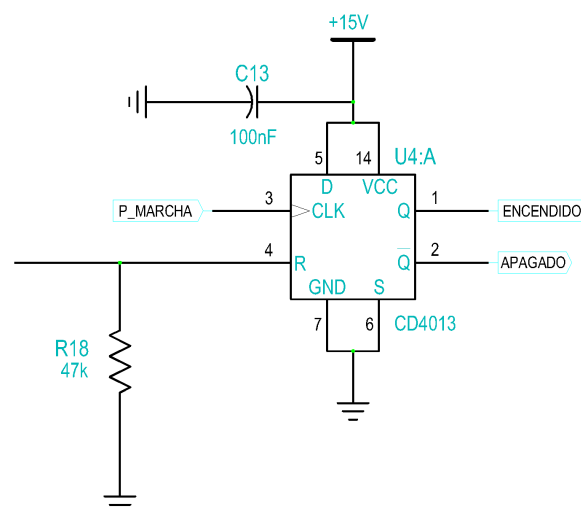


Figura 4.8: Biestable utilizado para el encendido y apagado del convertidor

### 4.2.1. Pulsadores

El pulsador P\_MARCHA enciende el convertidor siempre y cuando no existan errores en el sistema, mientras que el pulsador P\_PARO apaga el convertidor después de un funcionamiento normal, es decir, si antes no se ha apagado por la activación de algún error. El pulsador P\_SGIRO se utiliza para informar al microcontrolador que se desea invertir el sentido de giro actual. La figura 4.9 muestra la configuración de P\_SGIRO

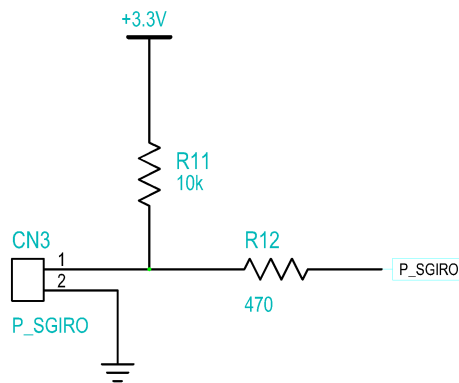


Figura 4.9: Configuración del pulsador P\_SGIRO

Se trata de un pulsador normalmente abierto y conectado a +3.3V a través de R11 y R12, por lo tanto su accionamiento genera un “cero” lógico que recibe el microcontrolador para proceder al cambio del sentido de giro. Al ser un pulsador únicamente conectado al micro su función puede ser modificada por software si se desea.

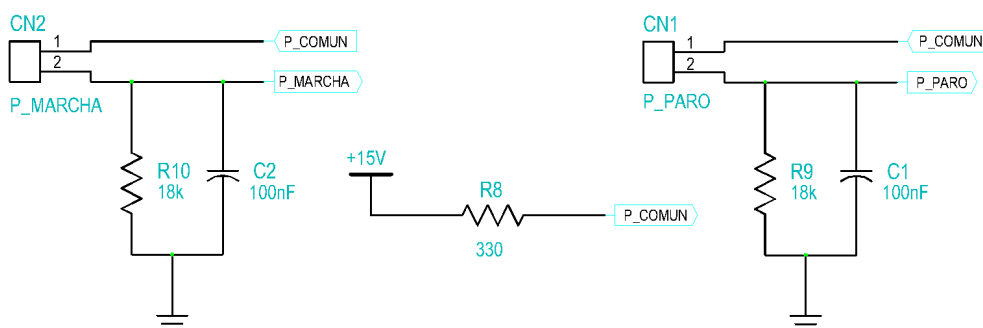


Figura 4.10: Configuración de los pulsadores P\_MARCHA y P\_PARO

La figura 4.10 muestra la configuración de los pulsadores P\_MARCHA y P\_PARO. Ambos pulsadores están normalmente abiertos y su accionamiento genera un “uno” lógico.

Para evitar rebotes en su accionamiento se ha diseñado el circuito de primer orden formado por C2, R8 y R10 para el caso del pulsador P\_MARCHA, análogo al de P\_PARO. Cuando el pulsador se abre C2 se descarga por R10 hasta 0V para volverse a cargar en un posterior accionamiento.

Cuando el pulsador se acciona se pretende que C2 se cargue rápidamente. Para ello, se desprecia la influencia de R10 en la carga, y se dimensiona R8. La señal de P\_PARO se acondiciona mediante un optoacoplador para ser enviada al microcontrolador además de a la puerta OR de la figura 4.7.

Además de los pulsadores indicados anteriormente, se cuenta con otro pulsador colocado directamente sobre la placa en montaje superficial. Se trata del pulsador P\_RESET, este pulsador manda una señal de reset al microcontrolador lo que provoca su reinicio. Si el microcontrolador se queda “colgado” por algún motivo, se puede hacer uso de este pulsador para volver a inicializarlo. En la figura 4.11 se muestra la configuración del pulsador.

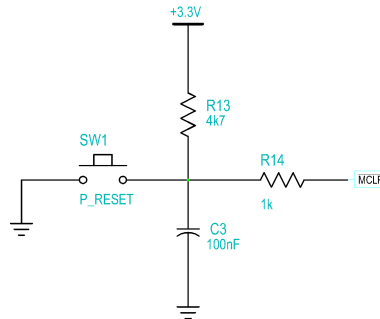


Figura 4.11: Configuración del pulsador P\_RESET

El pulsador permanece normalmente abierto, la señal  $\overline{MCLR}$  se envía al microcontrolador por una entrada del mismo diseñada para tal propósito. Se trata de una señal negada, por lo tanto un cero lógico causa un reset. En el estado de reposo la señal  $\overline{MCLR}$  está conectada a través de R13 y R14 a +3.3V

#### 4.2.2. Diodos LED

Se utilizan 6 diodos LED para poder visualizar los diferentes estados del convertidor. Cinco de ellos se utilizan para mostrar el estado de las protecciones y un sexto para visualizar el estado del convertidor.

Los cinco diodos LED encargados de visualizar las protecciones están directamente conectados a cinco salidas del microcontrolador, Se permite de este modo configurar libremente su encendido por software. Se utilizan diodos rojos de 5mm.

El diodo encargado de visualizar el encendido o apagado del convertidor no está conectado al microcontrolador, está conectado a la señal ENCENDIDO, por lo que el micro no tiene ningún control sobre él. Se utiliza un diodo verde también de 5mm.

En la figura 4.12 se muestra el circuito de visualización formado por los 6 diodos LED. Cada diodo está conectado en serie con una resistencia que limita su corriente a un valor de aproximadamente 3mA. La resistencia de D1 es mucho mayor debido a la tensión de la señal ENCENDIDO que es de 15V, mientras que las otras señales son de 3.3V. Se trata de diodos de bastante luminosidad, por lo que con poca corriente lucen muy bien.

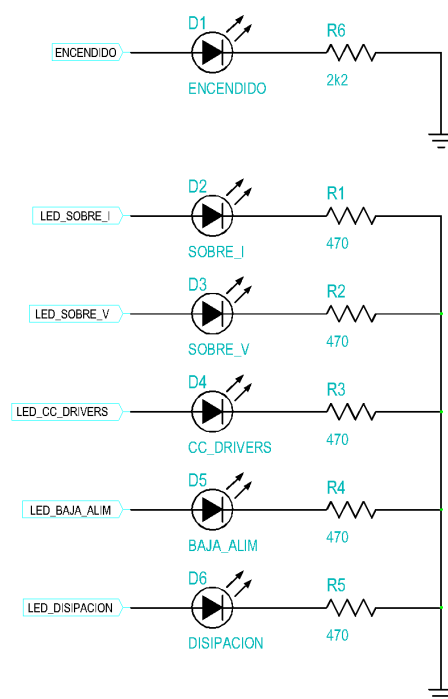


Figura 4.12: Circuito de visualización de diodos LED

### 4.3. Señales PWM

Las salidas PWM del microcontrolador se deben de acondicionar a los niveles de tensión de los drivers (0/15V). Para ello se utiliza el integrado SN7407, se trata de buffer no inversor con salida a colector abierto. En la figura 4.13 se puede ver el circuito utilizado.



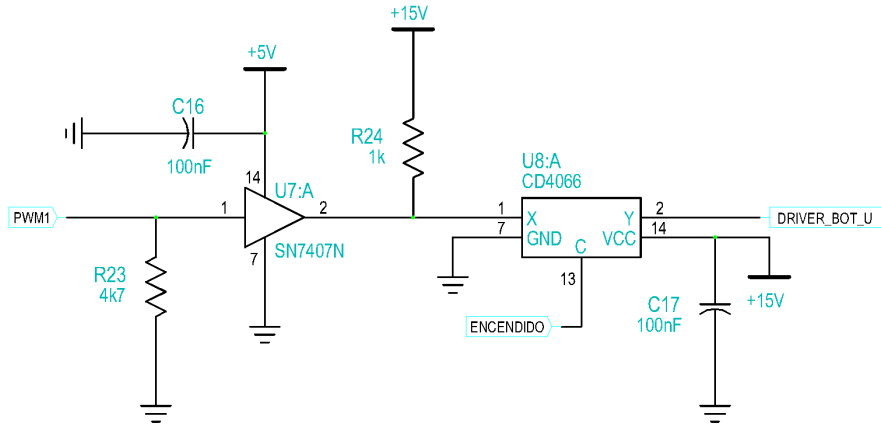


Figura 4.13: Circuito de acondicionamiento de las señales PWM

Mediante la resistencia de *Pull-Down* R23 se asegura el estado lógico de 0V, debido que la salida del buffer es de colector abierto necesita de una resistencia de *Pull-Up* (R24) conectada a +15V. El valor de R24 debe asegurar una caída de tensión menor a 1V, ya que sino el driver no recibiría los niveles lógicos adecuados. De este modo, se consiguen obtener niveles de tensión en funcionamiento de entre 0V y 14.5V, más que aceptables para poder distinguir un nivel alto de uno bajo.

El circuito integrado 4066 permite la desconexión de la señal PWM de los drivers cuando la señal ENCENDIDO está a 0V. Se trata de un conmutador digital de alta velocidad, para comandarlo se utilizaba la señal ENCENDIDO del biestable 4013. De este modo, siempre que ENCENDIDO esté a un nivel alto las señales PWM llegan a los drivers y los IGBTs empiezan a conmutar.

Se utiliza esta misma configuración para todas las salidas excepto para la rama de disipación. En este caso no se incluye el integrado 4066, debido a que no interesa dejar de disipar energía aunque se apague el convertidor. De este modo, en el caso que se esté inyectando energía en el bus de continua y salte la protección de sobretensión, el convertidor se apagará pero la rama

de disipación seguirá funcionando hasta que la tensión de bus vuelva a sus valores nominales.

En la figura 4.14 se puede ver la configuración adoptada para la señal de PWM correspondiente a la rama de disipación.

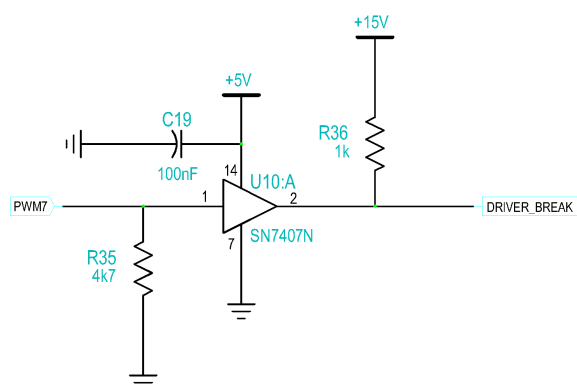


Figura 4.14: Circuito de acondicionamiento de la señal PWM de la rama de disipación

Se han utilizado resistencias de 250mW y de una tolerancia del 5 %, todos los circuitos integrados llevan un condensador de 100nF, colocado entre la entrada de alimentación y GND, como estabilizador de tensión.

## 4.4. Acondicionamiento de señales

La tensión de trabajo del microcontrolador seleccionado es de 3.3V, lo que obliga a acondicionar todas las señales a este nivel de tensión. El dsPIC admite valores de hasta 5V en alguno de sus pines<sup>1</sup>, pero tan solo como entradas digitales, por lo tanto se opta por acondicionar todas las tensiones a 3.3V.

<sup>1</sup>Véase *Figura 3.1: Diagrama de pines del dsPIC33FJ256MC710A*

Debido al bajo nivel de tensión requerido, las tensiones obtenidas por las LEM de tensión y corriente no necesitan ser amplificadas, ya que la corriente que circula por las resistencias de medida provoca una caída de tensión dentro del rango requerido.

El encoder genera pulsos de niveles comprendidos entre los 0V y los 5V, además está diseñado bajo el estándar RS-422 de señales diferenciales, con lo cual se necesita un circuito para acondicionar dichas señales, filtrando posibles ruidos y adaptando los niveles de tensión a los requeridos por el microcontrolador.

El uso de tensiones tan pequeñas hace necesario recurrir a componentes de baja tolerancia (del orden del 1 % para resistencias) y con bajo coeficiente térmico. Aún así es muy probable que haya pequeños errores en las medidas, por lo que posteriormente se deben corregir por código mediante offsets o/y filtros digitales.

#### 4.4.1. Acondicionado de tensión

La tensión del bus de continua se mide mediante la LEM LV 25-P, instalada en una placa en uno de los laterales del módulo de potencia, el circuito utilizado para realizar la medida de tensión puede verse en el *Capítulo 2. Módulo de potencia*.

Al circuito de control le llega una corriente proporcional a la tensión medida en el bus. La tensión máxima que puede medirse en el bus de continua se establece en 670V, tensión que coincide con el valor al que se activa la protección de sobretensión. Por encima de 670V no tiene sentido seguir tomando medidas ya que la protección habrá desconectado el convertidor.

A este nivel de tensión se obtiene una corriente en el secundario de la LEM de 29.91mA según la *ecuación 2.2 del Módulo de potencia*.

Para conseguir una tensión escalada la corriente anterior se hace circular por una resistencia  $R_M$  de valor conocido. Los valores vienen limitados en función de la alimentación de la LEM y de la corriente que circula por su primario. Para el caso descrito el intervalo de valores de  $R_M$  aparecen reflejados en la ecuación 4.1.

$$\left. \begin{array}{l} V_{cc} = \pm 15V \\ I_p < 14mA \end{array} \right\} \Rightarrow \left. \begin{array}{l} R_{Mmin} = 100\Omega \\ R_{Mmax} = 190\Omega \end{array} \right\} \quad (4.1)$$

Se necesita un valor de tensión  $V_M \leq 3.3V$  para que esté comprendido dentro de los niveles aceptados por el microcontrolador. A partir de la ecuación 4.2 se obtiene el valor de  $R_M$  en función del criterio de tensión establecido.

$$R_M \leq \frac{V_M}{I_S} = \frac{3,3V}{29,91mA} = 110,33\Omega \quad (4.2)$$

Para conseguir los  $110\Omega$  se colocan dos resistencias en paralelo de  $220\Omega$  cada una y  $250mW$ , con una tolerancia del 1%. Se procurará que sean de un bajo coeficiente térmico. De este modo se obtiene una tensión  $V_M = 3.29V$  para  $670V$  de tensión en el bus de continua.

Debido al ruido electromagnético derivado de las altas frecuencias de conmutación de los IGBTs, se hace estrictamente necesario realizar un filtrado de la señal proveniente del sensor mediante un filtro paso-bajo.

Esto se consigue añadiendo un condensador en paralelo con las resistencias anteriormente calculadas. Para ello se fija la frecuencia de corte del filtro igual o menor a una década por debajo de la de conmutación. De este modo, a partir de la ecuación 4.3, se obtiene el valor de la capacidad del condensador.

$$\omega_c \leq \frac{2\pi \cdot f_{con}}{10} \quad (4.3)$$

$$\frac{1}{R_M \cdot C_M} \leq \frac{2\pi \cdot f_{con}}{10}$$

$$C_M \geq \frac{10}{2\pi \cdot 110 \cdot 20k} = 723nF$$

Combinando tres capacidades en paralelo de 270nF se consigue un valor de  $C_M = 810nF$ . Se eligen condensadores de tipo cerámico o de poliéster.

La figura 4.15 muestra el esquema del circuito de acondicionamiento y filtrado de la medida de tensión del bus de continua.

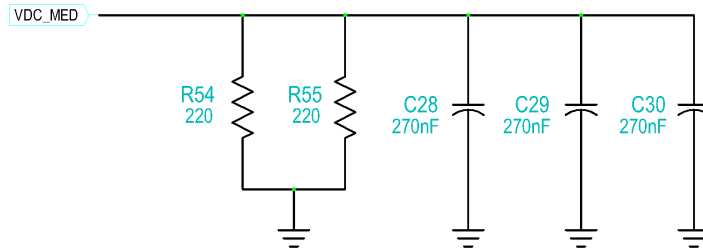


Figura 4.15: Circuito de acondicionamiento y filtrado de la tensión de bus

#### 4.4.2. Acondicionado de corriente

El módulo de potencia de Semikron incluye dos sensores de corriente del tipo LEM LAH-25 NP sobre las fases U y V que van hacia el motor. El valor de corriente de la tercera fase (W) no se necesita sensar, ya que se puede obtener fácilmente a partir de las otras dos aplicando las leyes de Kirchhoff.

Se necesita conocer el valor máximo que puede alcanzar la corriente antes que salte la protección, debido a que se han establecido unas protecciones por hardware, y se quiere conseguir la mayor resolución posible en la medida. Se tienen dos valores máximos, por un lado, el máximo de la protección analógica, y por otro lado, el máximo que puede leer el microcontrolador.

El valor máximo de corriente que puede leer el micro se sitúa entorno a los  $13.5 A_{pico}$ , valor más cercano a la corriente nominal. El máximo de la protección de sobrecorriente se fija en los  $17 A_{pico}$ , de este modo, se obtiene una mejor resolución dentro del rango nominal de trabajo del convertidor y se protege el equipo ante un pico elevado de corriente.

También hay que tener en cuenta que el microcontrolador no acepta tensiones negativas en sus pines, y como la célula LEM no es más que un transformador de corriente acaba reproduciendo de forma escalada la onda alterna del primario en su secundario. Esta onda se mueve entre valores negativos y positivos, teniendo su centro en 0, por lo que se hace indispensable aplicar un offset positivo de  $1.65V$  para centrarla en ese valor y que no supere los  $3.3V$  ni baje de los  $0V$  protegiendo así el microcontrolador.

En primer lugar, se calcula la resistencia  $R_M$  sobre la que se realizará la medida de tensión. La corriente máxima que se quiere escalar a los niveles aceptados por el dsPIC es de  $13.5 A_{pico}$  en el primario, lo que según la ecuación 4.4 son  $13.5mA$  por el secundario.

$$I_S = I_P \cdot a = 13,5 \cdot 0,001 = 13,5mA \quad (4.4)$$

Como se trata de una señal alterna, su valor oscilará entre  $-13.5mA$  y  $+13.5mA$ , por lo tanto es necesario aplicarle un offset positivo de  $13.5mA$ . De este modo, la corriente oscilará entre  $0$  y  $27mA$ . Una vez determinado el valor máximo de corriente se puede calcular  $R_M$  mediante la ecuación 4.5

$$R_M \leq \frac{V_M}{I_S} = \frac{3,3V}{27mA} = 122,22\Omega \quad (4.5)$$

Para conseguir el offset de  $13.5 mA$  se colocan dos resistencias de  $150\Omega$  cada una formando un divisor de tensión, una resistencia se conecta a  $+3.3 V$  y la otra a GND obteniendo un valor medio de  $1.65 V$ . Para conseguir un

valor equivalente de  $R_M = 122\Omega$  se coloca una tercera resistencia de  $47\Omega$  en serie con el divisor de tensión tal y como se muestra en la figura 4.16.

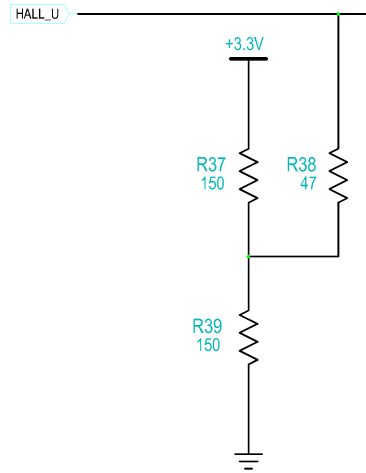


Figura 4.16:  $R_M$  y divisor de tensión

Se puede observar que el valor de  $R_M = 122\Omega$  se encuentra dentro del intervalo de valores permitido por el fabricante:

$$\left. \begin{array}{l} V_{cc} = \pm 15V \\ I_p[A_{t_{RMS}}] \end{array} \right\} \Rightarrow \left. \begin{array}{l} R_{Mmin} = 67\Omega \\ R_{Mmax} = 263\Omega \end{array} \right\} \quad (4.6)$$

Con esta configuración se consigue escalar la corriente a valores soportados por el microcontrolador, de todos modos se debe tener en cuenta que en ocasiones se sobrepasará el intervalo marcado anteriormente, de hecho, se ha fijado la protección analógica en  $17 A_{pico}$ , lo que corresponde a una tensión escalada de  $3.721V$  para los  $+17 A_{pico}$  y de  $-0.427V$  para los  $17 A_{pico}$  negativos.

La solución propuesta para evitar dañar el micro cuando esto suceda consiste en colocar dos diodos Schottky, uno con el cátodo a  $+3.3V$  y otro con el ánodo a GND. De esta forma cuando la tensión supere  $3.3V$  o sea

inferior a 0V el diodo entrará en conducción, llegando al micro 3.3V más la caída de tensión del diodo ó 0V menos la caída de tensión del diodo.

Al utilizar diodos Schottky la caída de tensión pasa de 0.6V de un diodo típico a tan solo 0.3V, que entra dentro de lo aceptable por el microcontrolador sin dañarlo. Tan solo resta colocar una resistencia de 4.7kΩ en serie entre la señal que va hacía la protección y la que va hacía el microcontrolador.

Todas las resistencias serán de 250mW y, las empleadas para conseguir el valor de  $R_M$ , de como mucho el 1% de tolerancia con un bajo coeficiente térmico. R40 no tiene un valor crítico, por lo que con una tolerancia del 5% es suficiente.

Debido al ruido electromagnético derivado de las altas frecuencias de conmutación de los IGBTs, se hace estrictamente necesario realizar un filtrado de la señal proveniente del sensor mediante un filtro paso-bajo.

Esto se consigue añadiendo un condensador en paralelo con las resistencias anteriormente calculadas. Para ello, se fija la frecuencia de corte del filtro igual o menor a una década por debajo de la de conmutación. De este modo, a partir de la ecuación 4.7 se obtiene el valor de la capacidad del condensador.

$$\omega_c \leq \frac{2\pi \cdot f_{con}}{10} \quad (4.7)$$

$$\frac{1}{R_M \cdot C_M} \leq \frac{2\pi \cdot f_{con}}{10}$$

$$C_M \geq \frac{10}{2\pi \cdot 122 \cdot 20k} = 652nF$$

Combinando dos capacidades en paralelo de 330nF se consigue un valor de  $C_M = 660nF$ . Se eligen condensadores de tipo cerámico o de poliéster.



En la figura 4.17 se muestra el circuito completo de acondicionamiento y filtrado de la corriente de la fase U, para la fase V se realizará exactamente el mismo circuito.

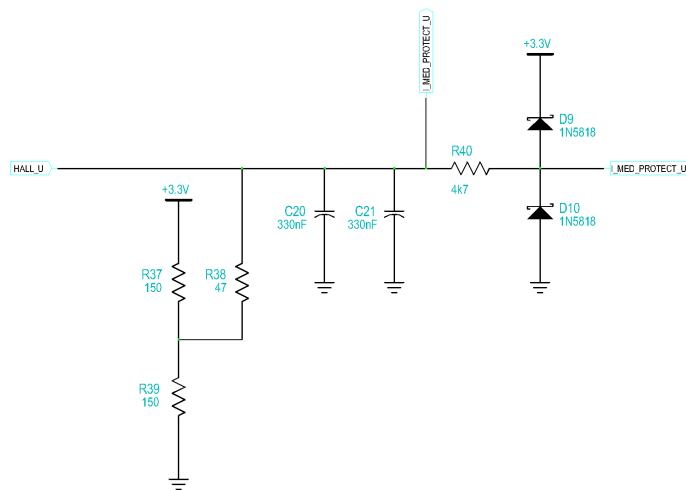


Figura 4.17: Circuito de acondicionamiento y filtrado de la corriente

### 4.4.3. Acondicionamiento del encoder

Las señales generadas por el encoder están en el rango de los 0V y 5V. Además, se tiene un par de señales complementadas por cada canal, tal y como dicta el estándar RS-422. Se necesita obtener una señal por cada canal en los niveles aceptados por el microcontrolador, entre 0V y 3.3V.

En la tabla 4.3 se muestra la asignación de las señales de cada pin proveniente del encoder por el puerto RJ45.

Pin	Símbolo	Función
1	A	Canal A (0/5V)
2	V <sub>cc</sub>	Alimentación Positiva (+5V)
3	B	Canal B (0/5V)
4	$\bar{A}$	Canal $\bar{A}$ (0/5V)
5	INDX	Canal INDX (0/5V)
6	$\bar{B}$	Canal $\bar{B}$ (0/5V)
7	GND	Alimentación Negativa (GND)
8	$\overline{INDX}$	Canal $\overline{INDX}$ (0/5V)

Tabla 4.3: Asignación de pines del encoder

Para el acondicionamiento de las señales se ha optado por utilizar un circuito utilizado por algunos fabricantes de encoders. En las características del circuito se comenta que tiene un alto grado de rechazo a ruidos y que puede funcionar a frecuencias elevadas, del orden de 50 kHz.

La salida del comparador diferencial (AD820) se lleva a un optoacoplador, consiguiendo aislamiento eléctrico entre el encoder y el microcontrolador.

Una cosa que hay que tener muy en cuenta son los tiempos de encendido y apagado del optoacoplador, ya que utilizando un encoder de 1024 pasos y a una velocidad de 2000 RPMs se obtiene una frecuencia a la entrada del opto de aproximadamente 35 kHz.

A la hora de realizar el circuito se ha probado a utilizar un optoacoplador 4N25 obteniendo muy malos resultados debido a los tiempos de encendido y apagado. Posteriormente se ha sustituido por un H11L1 con unos tiempos muy inferiores y salida lógica *Schmitt Trigger* en vez de fototransistor, obteniendo excelentes resultados. Además, al ser *Schmitt Trigger* permite filtrar todavía más cualquier ruido de alta frecuencia que se haya colado a la entrada del optoacoplador.

El circuito finalmente adoptado se puede ver en la figura 4.18

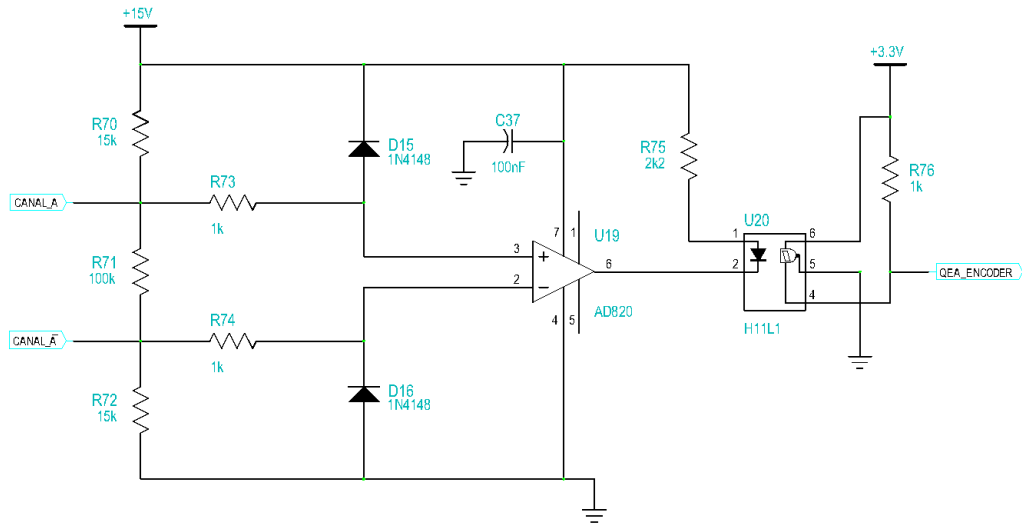


Figura 4.18: Circuito de acondicionamiento y filtrado del encoder

Utilizaremos el mismo circuito tanto para el Canal A, el Canal B como el Index.

#### 4.4.4. Acondicionamiento señales de estado

Todas las señales de los estados en que se encuentran las protecciones se necesitan llevar al microcontrolador. De este modo, se consigue mantener informado al micro de cualquier alarma que se haya detectado por hardware.

Se trabaja con una lógica de entre 0V y 15V, por lo tanto se debe acondicionar estas señales de estado a la lógica del microcontrolador, entre 0V y 3.3V. Para ello se utilizan optoacopladores 4N25 con sus resistencias limitadoras de corriente. En la figura 4.19 se puede ver el circuito empleado para acondicionar el estado de la señal de SOBRETENSION.

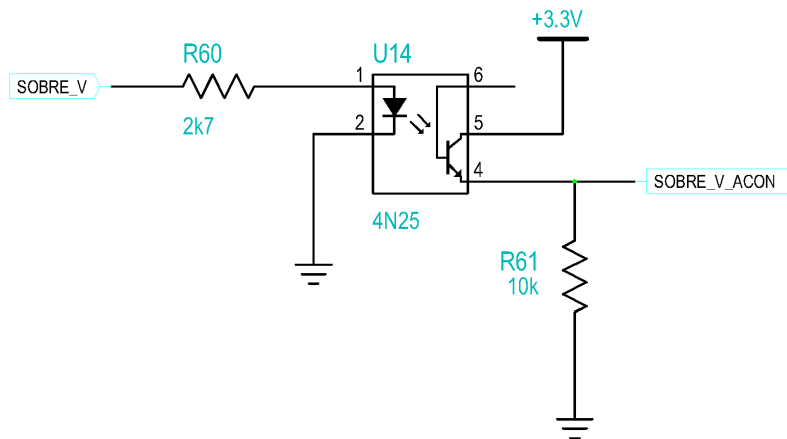


Figura 4.19: Circuito de acondicionamiento de la señal SOBRE\_V

Con R60 se asegura tener la corriente de entrada al optoacoplador limitada entorno a 6mA. Se utiliza una resistencia de 10kΩ colocada a GND para asegurar el nivel lógico de 0V, también conocida como *Pull-Down*. De este modo, mientras el LED del optoacoplador permanezca apagado la entrada del microcontrolador se mantendrá a 0V a través de R61, en el momento que se encienda el LED el fototransistor entrará en saturación llevando la referencia de +3.3V a la entrada del micro.

Se utilizará el mismo circuito para el resto de señales de estado, teniendo en cuenta que la corriente de entrada al optoacoplador debe de ser de al menos 6mA para asegurar su encendido.

## 4.5. Protección de sobrecorriente

La protección de sobrecorriente permite apagar el convertidor si se supera un valor límite de corriente en cualquiera de las fases U o V. El valor límite al que se activa esta protección es de  $\pm 17 A_{\text{pico}}$ . El circuito está formado por:

- Cuatro comparadores formados por el integrado LM339 que analizan en todo momento si la corriente se sale de los límites de protección.
- Dos circuitos formados por el LM385 para generar las dos referencias que marcan los límites, tanto superior como inferior, a los que se activa la protección.
- Una puerta lógica OR cuádruple<sup>2</sup>.

La alimentación del comparador LM339 se realiza entre +15V y GND, debido a que a la salida tan solo se necesitan valores lógicos entre 15V y 0V.

La configuración que se utiliza es la de un comparador básico tal y como se ve en la figura 4.20.

Se utiliza esta configuración ya que el circuito biestable posee una memoria que mantiene su salida a nivel alto hasta que no se genere un RESET, aunque en el comparador su salida se ponga a nivel bajo. Esto es importante por el ruido generado, que podría influir en el comparador.

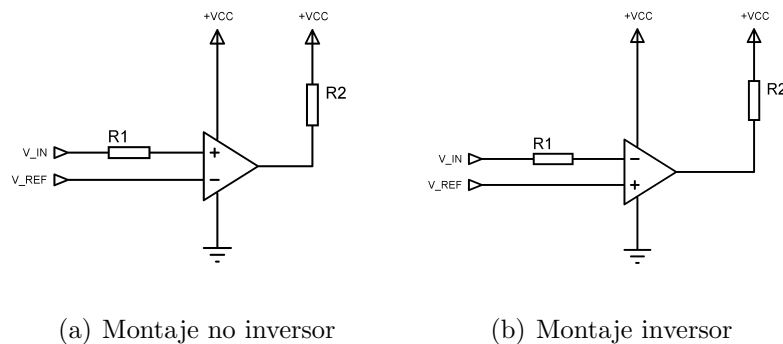


Figura 4.20: Comparador básico en montaje no inversor e inversor

<sup>2</sup>Se explica en el apartado 4.4: *Lógica del convertidor* (Ver página 43)

Las salidas de los comparadores son del tipo colector abierto. Es decir, su nivel alto se puede fijar a cualquier valor de tensión. Para el valor alto se utiliza el valor de alimentación, +15V y se conecta a la salida del comparador con una resistencia de *Pull-Up*. En las figuras 4.20 (a) y 4.20 (b) se puede observar lo explicado.

### Referencias de tensión

Para generar las dos referencias de tensión, tanto para el límite de los  $17 A_{pico}$  como para el de  $-17 A_{pico}$ , que equivalen a 3.72V y -0.42V respectivamente, se utilizan los circuitos mostrados en las figuras 4.21 y 4.22.

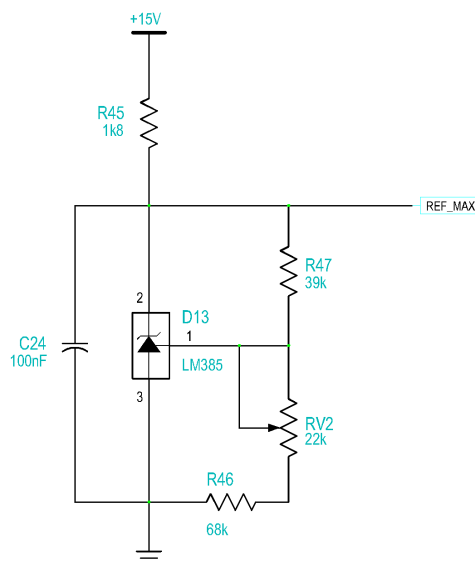


Figura 4.21: Circuito utilizado para generar REF\_MAX

Mediante el uso del integrado LM385 se consigue obtener una referencia de tensión mucho más estable que utilizando un divisor de tensión. Es decir, aunque la tensión de alimentación varíe entre  $\pm 5V$ , la tensión de referencia se mantiene constante.

En el datasheet del integrado se puede encontrar un ejemplo de aplicación del que se ha partido para obtener el circuito implementado. El potenciómetro RV2 permite ajustar la referencia deseada entre 3.4V y 4.1V, estos valores se obtienen de la ecuación 4.8

$$REF\_MAX = 1,24 \cdot \left( \frac{R46 + RV2}{R47} + 1 \right) \quad (4.8)$$

Para obtener la referencia negativa de -0.42V se utiliza el circuito de la figura 4.22, muy similar al utilizado para la referencia positiva. En este caso se debe alimentar el circuito con una tensión negativa de -15V.

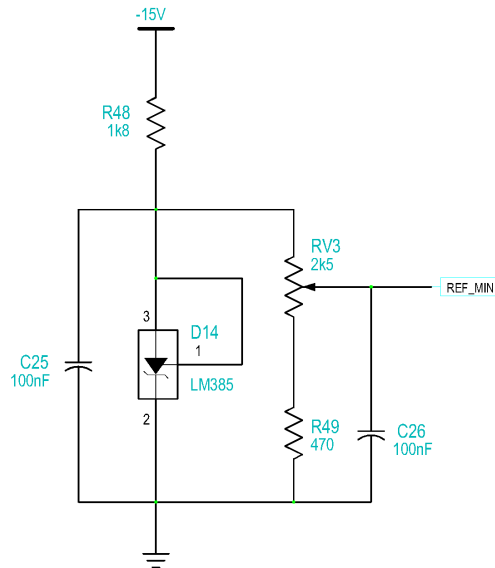


Figura 4.22: Circuito utilizado para generar  $REF\_MIN$

Partimos de 1.24V, mínima referencia que se puede obtener del LM385, y posteriormente utilizamos un divisor de tensión formado por RV3 y R49 para conseguir la referencia deseada. Al utilizar una alimentación negativa se debe colocar el integrado de forma invertida, suministrando a su salida -1.24V respecto a GND.

El potenciómetro RV3 permite ajustar la referencia deseada entre -1.24V y -0.29V, estos valores se obtienen de la ecuación 4.9

$$REF\_MIN = -1,24 \cdot \frac{R49}{R49 + RV3} \quad (4.9)$$

Los condensadores C24, C25 y C26 se utilizan para filtrar cualquier ruido de alta frecuencia que pueda aparecer en las alimentaciones de los integrados. Su valor se estable en 100nF.

### **Circuitos comparadores**

Una vez generadas las dos referencias de tensión se llevan al operacional LM239, para ser comparadas con el valor de corriente de la fase correspondiente medido y acondicionado.

Tanto para la fase U como para la fase V se utiliza el circuito de la figura 4.23, en el que se emplean dos comparadores del integrado LM239, uno en modo no inversor y otro en modo inversor.



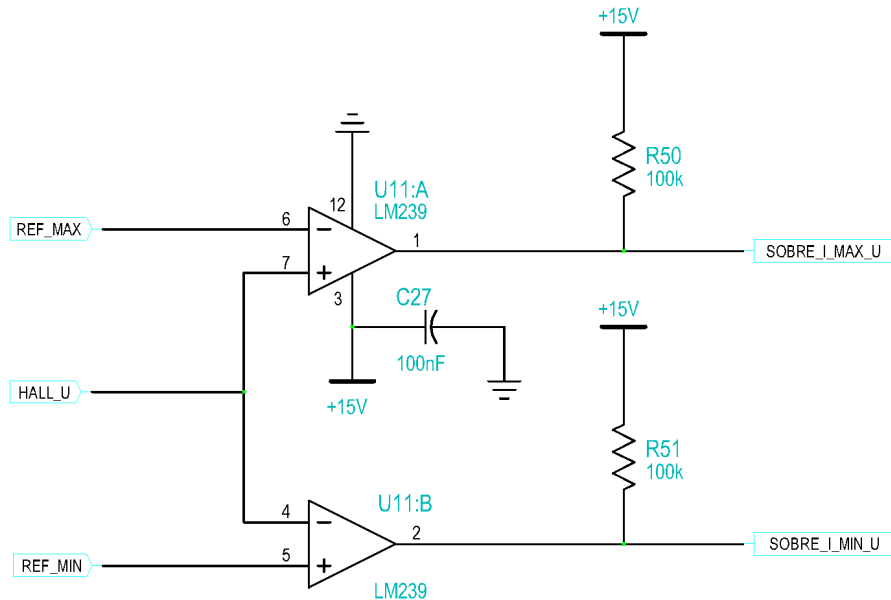


Figura 4.23: Etapa de comparación de la protección de sobrecorriente de la fase U

La salida del comparador se lleva a la puerta OR a través de una resistencia de *Pull-Up* conectada a +15V. La puerta lógica es de tecnología CMOS, por lo que puede trabajar con corrientes muy pequeñas, esto hace que el valor de la resistencia R50 sea de 100kΩ.

El comparador se alimenta en modo fuente simple, es decir, Vcc se pone a +15V y Vdd a GND. De este modo, ante un nivel alto se obtienen 15V a través de la resistencia R50, y ante un nivel bajo el comparador lleva la salida a GND ó 0V.

## 4.6. Protección de sobretensión

La protección de sobretensión apaga el convertidor si la tensión del bus de continua supera el límite establecido, fijado en 670V. El circuito de protección lo constituye uno de los dos comparadores del integrado LM293. Se realiza un montaje básico de comparación en modo no inversor tal y como se describe en la figura 4.20.

Se ha seleccionado una tensión de referencia de 3.3V, que corresponde con el valor máximo que se va a medir, 670V, valor al que se debe activar la protección. En la figura 4.24 se puede ver el circuito adoptado para realizar la protección de sobretensión.

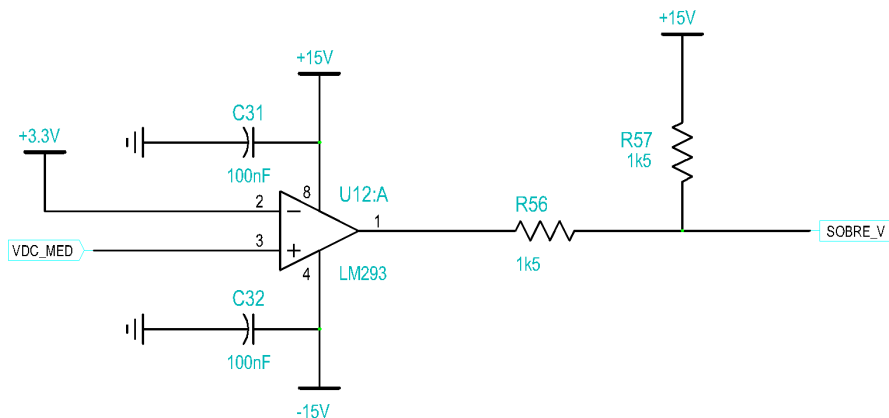


Figura 4.24: Circuito para la protección de sobretensión

La salida del comparador se lleva a una puerta lógica y también al optoacoplador encargado de acondicionar la señal de estado que posteriormente llega al microcontrolador, a través de la resistencia R57 de *Pull-Up* conectada a +15V.

Debido a la corriente necesaria para el buen funcionamiento del optoacoplador, se ve necesario utilizar una resistencia de *Pull-Up* de bajo

valor, esto provoca que la corriente de retorno por el comparador durante el estado bajo sea bastante grande, por eso se incluye la resistencia R56, limitando a 10mA.

Los condensadores C31 y C32 se utilizan como filtros estabilizadores de la tensión de alimentación del integrado, con una capacidad de 100nF. Las dos resistencias utilizadas son de 250mW y de una tolerancia del 1%.

## **4.7. Charge Input Signal (CIS)**

El convertidor necesita de una señal para cortocircuitar, mediante dos relés, la resistencia de precarga una vez cargados los condensadores del bus de continua. Se trata de una señal de alimentación de 24V, necesaria para conmutar los relés y arrancar los ventiladores para la refrigeración forzada.

La señal CIS se activará siempre y cuando la señal ENCENDIDO esté a uno, de este modo se asegura que siempre que el convertidor esté conmutando las resistencias de precarga estarán desactivadas y los ventiladores funcionando.

Se hace así para evitar la dependencia con el microcontrolador, al no ser necesario este para activar o desactivar la señal CIS. De este modo, se asegura que ante un problema en el software no queden activadas las resistencias mientras estén los IGBTs conmutando, lo que produciría una tremenda disipación de energía en las resistencias llevándolas a su destrucción.

Para conseguir una tensión cercana a los 24V se recurre a alimentar la señal CIS entre -15V y +5V a partir de la fuente de alimentación, obteniendo una tensión de 20V. Se comprueba que con esta tensión es suficiente para el correcto funcionamiento de los ventiladores y de los relés.

Un optoacoplador 4N25 es el encargado de excitar la puerta de un MOSFET IRF530 al estar activa la señal de ENCENDIDO, haciéndolo entrar en saturación. La resistencia R91 limita la corriente a unos 6mA para iluminar el LED interno del optoacoplador. Se utiliza un divisor de tensión para obtener un valor de 10V entre puerta-emisor, suficiente para hacer conmutar el MOSFET.

Debido que se hacen conmutar circuitos inductivos, como son las bobinas de los relés, se hace necesario incluir un diodo 1N4148 entre los dos terminales del CIS. De este modo se deja un camino para que circulen las corrientes al producirse la conmutación.

Mediante el optoacoplador se consigue, además del acondicionado, un aislamiento entre la señal ENCENDIDO de la placa de control y la alimentación de 20V del CIS. En la figura 4.25 se puede ver el circuito diseñado para tal fin.

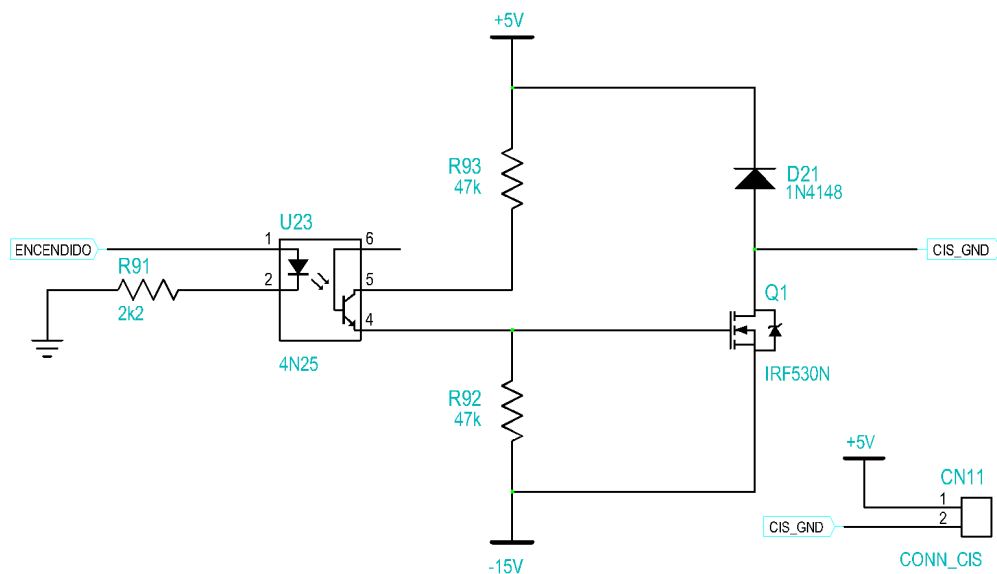


Figura 4.25: Circuito de acondicionamiento del CIS

## 4.8. Display LCD alfanumérico

Un display alfanumérico de matriz de puntos es un dispositivo de interfaz humana formado por una pantalla de cristal líquido o LCD (Liquid Crystal Display) sobre la que se pueden mostrar mensajes formados por distintos caracteres: letras, números, símbolos, etc. Se encuentran en distintos formatos, por ejemplo, 2x8, 2x16, 4x20... El primer dígito indica el número de filas del display y el segundo el número de columnas.

Para la placa de control del presente proyecto se ha elegido uno de 4 filas y 20 columnas retroiluminado, con el que se tiene suficiente espacio para mostrar todas las variables de forma simultánea.

Estos dispositivos vienen gobernados por un microcontrolador, que normalmente va incorporado sobre la misma placa de circuito impreso que soporta el LCD. Es muy habitual encontrarse con el controlador HD44780 de Hitachi, el mismo que utiliza el display seleccionado para este proyecto. El controlador se encarga de gestionar el display líquido: polarizar los puntos de la pantalla, generar los caracteres, desplazar la pantalla, mostrar el cursor, etc. El usuario se despreocupa de todos estos problemas y simplemente necesita conocer una serie de comandos o instrucciones de alto nivel que le permitirán mostrar mensajes sobre la pantalla de forma sencilla.

Los caracteres que se pueden representar están formados por una matriz de puntos que vienen almacenados en memoria ROM dentro del propio controlador. El fabricante reserva una pequeña zona de memoria RAM donde se pueden definir algunos caracteres especiales.

El número de pines de un display alfanumérico es normalmente de 14 (en este caso al ser retroiluminado es de 16). En la tabla 4.4 se muestra la asignación de las señales de cada pin. Hay tres tipos de señales en el LCD: de alimentación, de control y de datos.

Pin	Símbolo	Función
1	Vss	Alimentación negativa (GND)
2	Vdd	Alimentación positiva (+3.3V)
3	NC	No conectado
4	RS	Selección de registro (Register Select) RS = 1: Registro de Datos RS = 0: Registro de Instrucciones
5	R/W	Lectura / Escritura (Read / Write) R/W = 1: Lectura R/W = 0: Escritura
6	E	Habilitación del display (Enable) E = 1: Habilitado E = 0: Deshabilitado
7-14	DB[0..7]	Pines del bus de datos bidireccional
15	LED+	Alimentación retroiluminación (+3.3V)
16	LED-	Alimentación retroiluminación (GND)

Tabla 4.4: Asignación de pines del LCD

### Señales de alimentación

Son tres señales correspondientes a los pines 1, 2 y 3. El pin 1 corresponde con GND, el 2 a la alimentación positiva y el 3 normalmente corresponde al ajuste del contraste. En el LCD seleccionado para la placa de control el pin 3 no está conectado debido que al ser un LCD retroiluminado no necesita ajuste de contraste, por el contrario dispone de dos pines más de alimentación, el 15 y el 16. El pin 15 corresponde con el ánodo del LED de retroiluminación y el 16 con el cátodo.

## Señales de control

Los pines 4, 5, y 6 forman las señales de control del LCD. El pin 4 (RS) sirve para seleccionar el registro de datos (DR) o el de instrucciones (IR), poniendo RS=1 o RS=0, respectivamente. El pin 5 permite leer (R/W=1) o escribir (R/W=0) en el módulo LCD tanto datos como instrucciones. Y por último, el pin 6 (E) permite habilitar, con E=1, o deshabilitar el display (E=0).

## Señales de datos

Por último, los pines del 7 al 14 forman un bus de datos bidireccional de 8 bits (DB0 - DB7) por donde se pueden escribir datos e instrucciones y se puede leer el estado del display (si está o no ocupado, la posición actual del cursor, etc.)

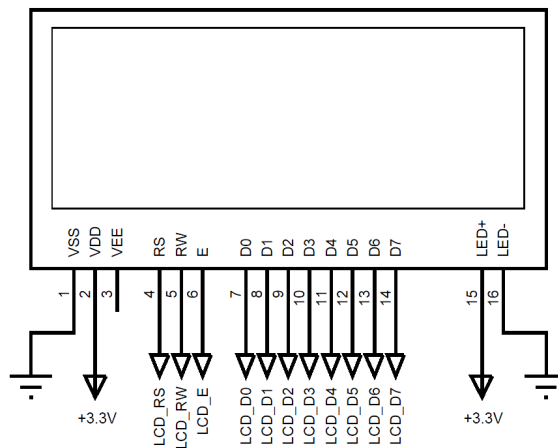


Figura 4.26: Esquema de conexiones del LCD

La figura 4.26 muestra un esquema del módulo LCD utilizado con todos sus pines y asignaciones que se han tenido en cuenta a la hora de diseñar la placa de control.

En el *Apéndice G.3 Librería LCD* aparece recogido el código necesario para comunicarse con el controlador HD44780 del módulo LCD mediante el dsPIC. Se necesita de una librería que contenga todas las funciones necesarias para facilitar el manejo del LCD desde el microcontrolador.

## 4.9. Fuente de alimentación

La fuente de alimentación se ha seleccionado en base a las tensiones necesarias por los componentes de la placa de control y sus consumos. Tanto los sensores de corriente como de tensión necesitan una alimentación de  $\pm 15V$ , los drivers y algunos integrados se alimentan a 15V y otros a 5V, además se necesita de una tensión de 3.3V para la alimentación del microcontrolador.

En base a estas necesidades se ha seleccionado una fuente de 40W del fabricante EMERSON, concretamente la LPT45. Se puede consultar su hoja de características en el *Anexo: Hojas de características*. En la figura 4.27 se observa el esquema de las salidas de alimentación.

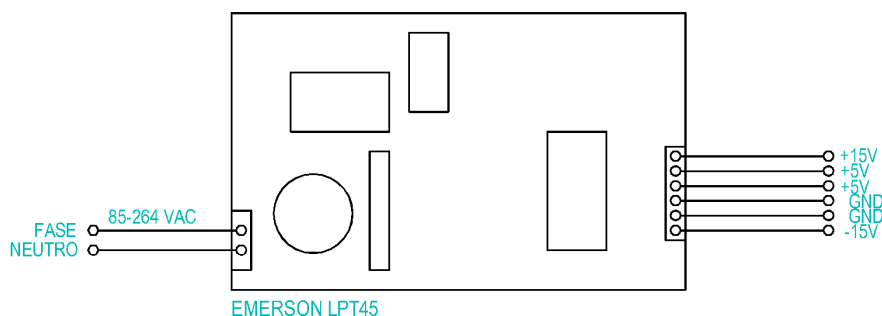


Figura 4.27: Fuente de alimentación Emerson LPT45



Los 3.3V se obtienen a partir de la línea de 5V mediante un regulador de tensión. Se selecciona el LM1117-3.3 que cumple con los requisitos exigidos sin necesidad de utilizar un radiador para disipar el calor, ya que los consumos del microcontrolador son muy pequeños. En la figura 4.28 se muestra el circuito utilizado para regular los 3.3V.

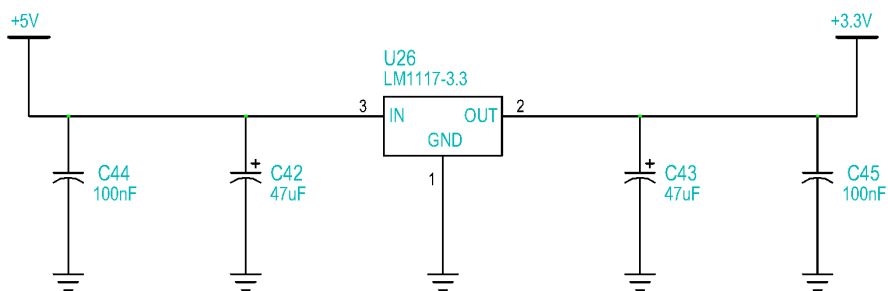


Figura 4.28: Circuito regulador de tensión de 3.3V

Para conseguir una tensión lo más estable posible y con el mínimo rizado se utilizan dos condensadores electrolíticos de  $47\mu\text{F}$  y dos cerámicos o de poliéster de  $100\text{nF}$ .

Cabe mencionar que la fuente de alimentación necesita de unos consumos mínimos para poder trabajar de forma eficaz, estos consumos aparecen en su hoja de características y se cumplen en todo momento.

El microcontrolador dsPIC33FJ256MC710A requiere de unos condensadores de  $100\text{nF}$  en cada entrada de alimentación, se diseña la placa de control con la posibilidad de colocarlos. En el montaje final se prescinde de ellos debido a que el micro se encuentra colocado en un módulo donde ya se incluyen estos condensadores.



## 5. Montaje final y ensayos

Una vez finalizado tanto la parte de programación como el dimensionado de la placa de control, se procede a realizar el montaje final. Para conseguir un montaje robusto se utiliza el propio módulo de Semikron como estructura de fijación.

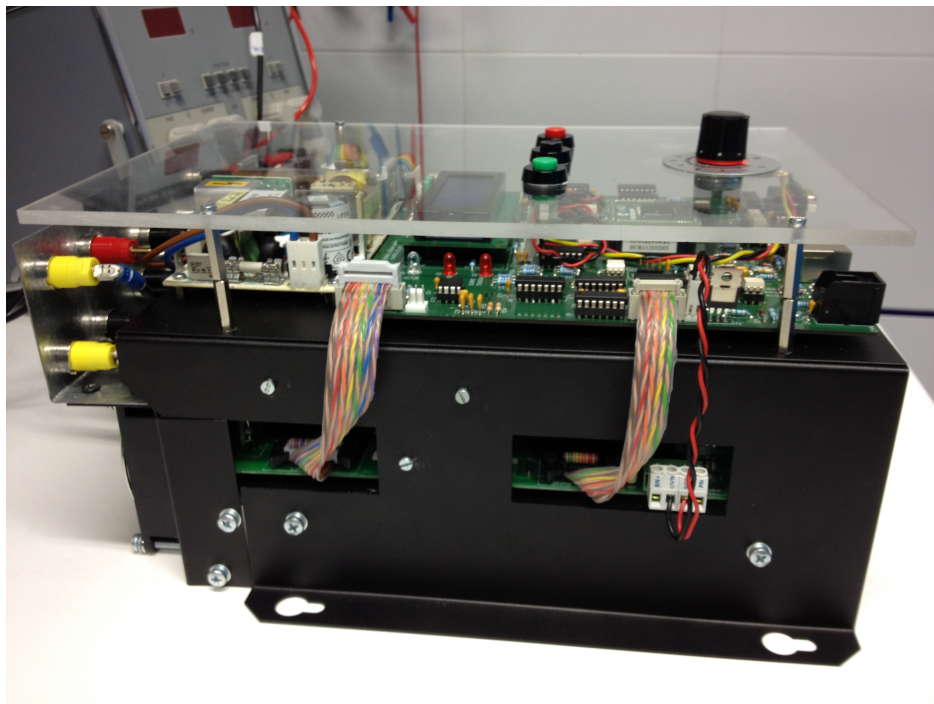


Figura 5.1: Conjunto placa de control, Semistack y fuente de alimentación. Vista lateral.

El montaje es muy sencillo, ya que la placa de control encaja a la perfección sobre el módulo de potencia. Junto a ésta se coloca la fuente de alimentación, y posteriormente, la tapa de metacrilato que contiene los botones y el potenciómetro quedando un conjunto muy compacto.



Figura 5.2: Conjunto placa de control, Semistack y fuente de alimentación. Vista superior.

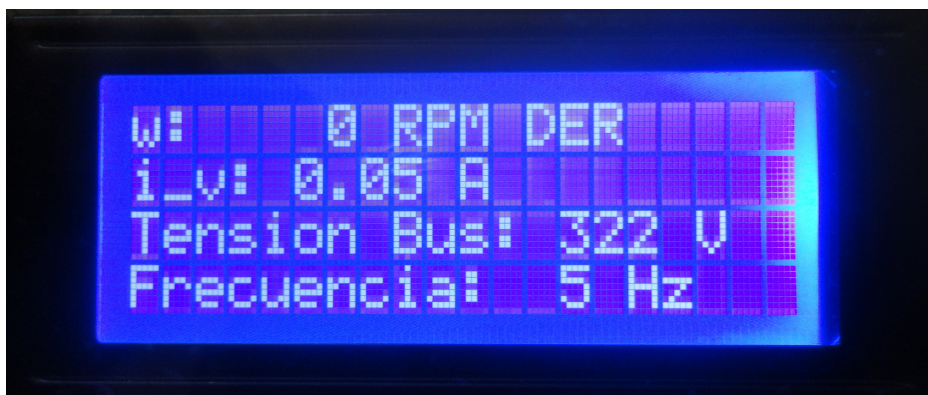
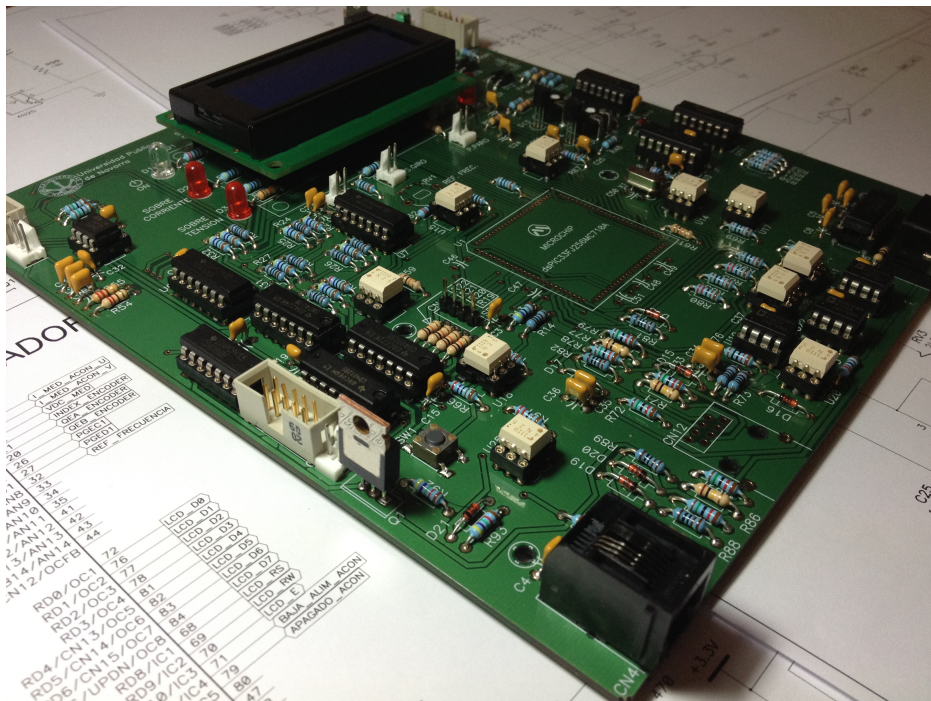


Figura 5.3: Detalle display LCD

A continuación, se muestran unas imágenes de la placa de control donde se puede ver claramente la posición central del microcontrolador con todos los optoacopladores a su alrededor. En ella todavía no está colocado el zócalo con el micro. Se observa también el display LCD y los diodos LEDs para la visualización de los estados del convertidor. En la figura 5.2 se puede ver el pulsador SMD utilizado para resetear el microcontrolador.





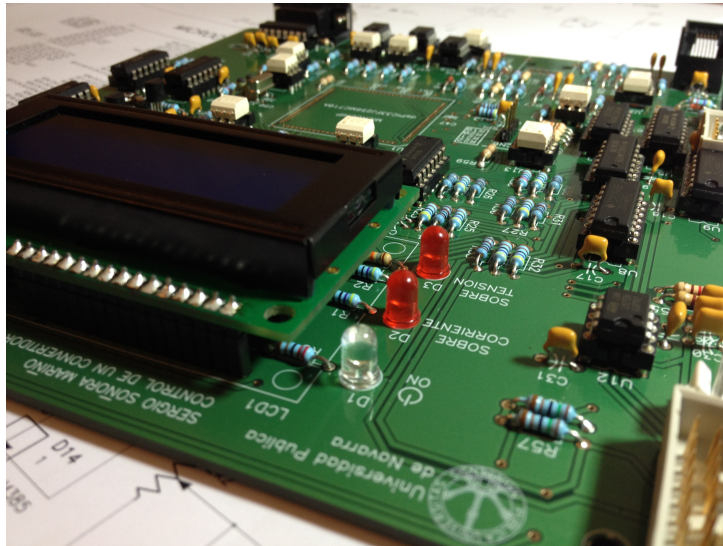


Figura 5.5: Detalle diodos LED y display LCD

Para la realización del rutado de la placa de control se utilizó el software *PCAD versión 2006*, posteriormente una empresa dedicada a la fabricación de PCBs a través de los ficheros de PCAD se encargó de producirla.

El montaje de los componentes se realizó con sumo cuidado debido al pequeño tamaño de estos, sobre todo la parte del microcontrolador, donde los pines están a tan solo 1.27mm unos de otros. También se han realizado pruebas de continuidad con la ayuda de un polímetro para descartar posibles cortocircuitos o falsos contactos.

Como se puede observar en las figuras, la placa de control dispone de varios conectores por donde se comunica con el módulo de potencia. La mayor parte de las conexiones están realizadas mediante cables planos trenzados de varios hilos, en concreto las conexiones con los drivers y los sensores de corriente.

El display LCD dispone de un zócalo donde se encaja, lo que facilita su extracción si fuese necesario.

## 5.1. Resultados

Una vez realizado el montaje práctico se procede a comprobar el funcionamiento de los diferentes módulos que componen la placa de control.

Se comprueba la fuente de alimentación para verificar que todas las tensiones son correctas. Posteriormente se comprueban los mecanismos de encendido y apagado del convertidor. Una vez comprobado que todo va bien, se programa el microcontrolador con programas sencillos y se observa si responde como se espera.

Antes de conectar los drivers se comprueban las salidas PWM del microcontrolador, si son complementadas, si existen o no tiempos muertos, etc. Una vez comprobado se prueba a fijar una frecuencia de referencia mediante el potenciómetro, posteriormente se analizan los pulsos con el osciloscopio para ver si corresponden.

Comprobados los módulos por separado se procede a realizar ensayos del equipo en conjunto.

Inicialmente se alimenta el bus de continua con una fuente de tensión continua variable para realizar una prueba de conmutación a muy baja potencia. Se conecta una carga y se procede a ejecutar el programa del módulo PWM. Se pulsa el pulsador P\_MARCHA, el diodo LED indica el encendido del convertidor y los ventiladores de disipación comienzan a girar, mediante el osciloscopio se pueden ver las conmutaciones de los IGBTs. Posteriormente se pulsa P\_PARO, el LED de encendido se apaga y los ventiladores dejan de funcionar.

Se comprueba el programa de medida de las corrientes, así como de la tensión de bus. Se observa un error en la medida menor al 5 % para la tensión de bus que se puede corregir directamente en el programa.

Una vez verificado lo anterior se pone en funcionamiento el convertidor en modo monofásico, con una tensión de bus de 315V, y se conecta un motor síncrono para probar el control V/f. A medida que se gira el potenciómetro se observa como el motor responde perfectamente a la rampa de aceleración solicitada. Realizando una inversión de fases por software se puede cambiar el sentido de giro.

Tan solo falta comprobar el encoder, se acopla el conector RJ45 y se procede a medir la velocidad a través del módulo QEI del microcontrolador. En el display se puede ver las RPMs, y como se asemejan a las calculadas a partir de la frecuencia y número de polos del motor.

A continuación se muestran las gráficas capturadas durante las pruebas V/f.

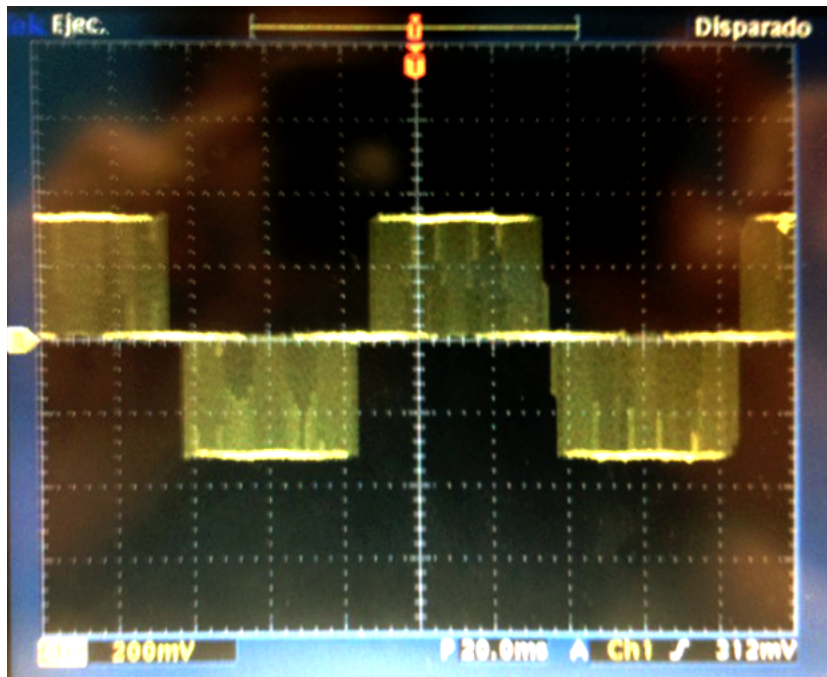


Figura 5.6: Señal de salida a 50Hz



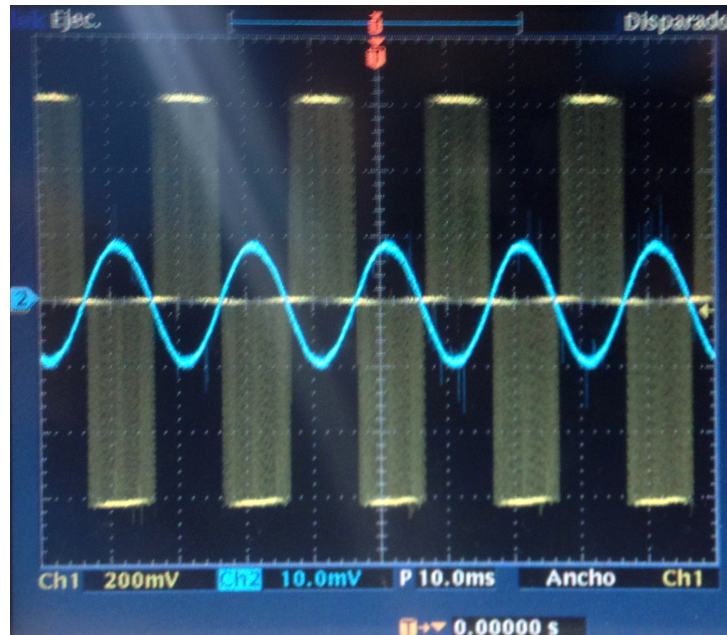


Figura 5.7: Vista de la corriente frente a la tensión a 50Hz

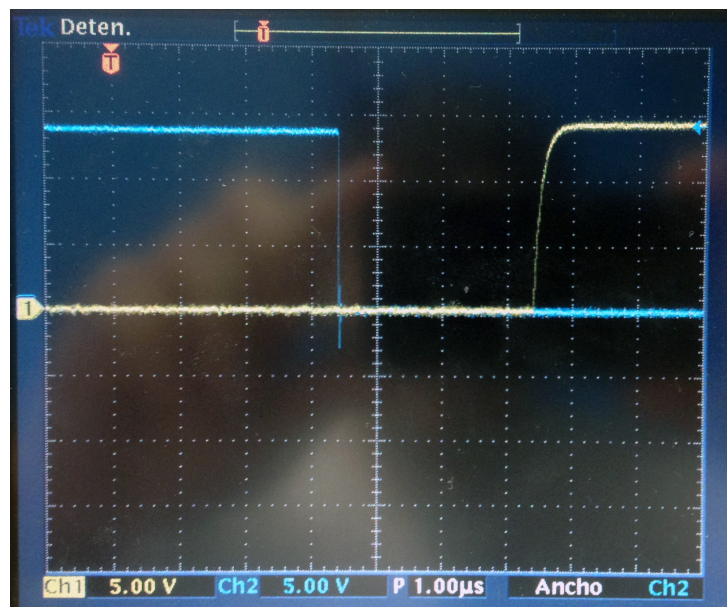


Figura 5.8: Tiempo muerto de  $3\mu s$



## 6. Conclusiones y líneas futuras

Después de haber realizado las etapas de diseño, montaje y ensayos del convertidor se han sacado las siguientes conclusiones.

En cuanto al montaje, se ha obtenido un equipo compacto y robusto. La carcasa de metacrilato protege y a la vez permite visualizar todos los elementos de la placa de control.

Dispone de numerosas entradas y salidas para poder realizar diversos tipos de controles, tanto en lazo cerrado como en abierto. El display LCD permite visualizar gran cantidad de datos a tiempo real.

El hecho de utilizar un microcontrolador como elemento central proporciona numerosas ventajas frente a los accionamientos analógicos no programables. Se espera que en futuros proyectos se aproveche todavía más esta particularidad.

En definitiva, la placa diseñada cumple perfectamente con el propósito de servir de plataforma de desarrollo de nuevos proyectos gracias a la versatilidad de los microcontroladores.

## **Fallos en la placa de control inicial**

Como todo proyecto que nace directamente de la etapa de diseño, es muy inusual que todo funcione a la primera, y en este caso no podía ser de otra manera. En las siguientes líneas se describen los problemas encontrados así como la forma de solucionarlos.

En el circuito de protección de sobrecorriente la referencia de tensión negativa estaba conectada a +15V cuando debe estar a -15V.

En el circuito de protección de sobretensión, las patas 2 y 3 del comparador LM293 aparecen intercambiadas en la PCB.

Otro detalle a tener en cuenta son los dos zéner que suministran una tensión de referencia de 1.24V, aparecen girados 180° en la PCB, por lo que se deben montar completamente al revés.

Uno de las partes que ha dado mayores problemas ha sido el circuito de acondicionamiento del encoder, el problema radicaba en los tiempos de conmutación de los optoacopladores 4N25, que no son lo suficientemente rápidos. Esto se debe a que el motor girando a una velocidad de 2000 RPM genera pulsos a una frecuencia de aproximadamente 35kHz que el optoacoplador no consigue seguir. Fue solucionado cambiando estos optoacopladores por unos mucho más rápidos.

Otra dato a tener en cuanto son los ruidos, en un primer momento se conectó el motor con cable sin apantallar, lo que provocaba que se indujese mucho ruido en los cables del encoder, dificultando la medida de este. Posteriormente se colocó cable apantallado mejorando notablemente el problema de las interferencias electromagnéticas.

Una vez subsanado esto, el control del convertidor se puede decir que ha cumplido todas las expectativas, quedando preparado para la programación de nuevos controles.

### **Líneas futuras.**

Algunas posibles mejoras pueden ser la inclusión de un segundo microcontrolador que disponga de módulo USB y se comunique con el dsPIC mediante el bus SPI. De este modo se puede dotar a la placa de una conexión USB para transmitir y recibir información a mayor velocidad que la conexión RS-232.

Otra opción sería la de incluir un módulo bluetooth con comunicación SPI o Serial para poder diseñar alguna aplicación móvil donde mostrar información proveniente del convertidor, y también poder enviar instrucciones desde un Smartphone.

Pero esto ya queda a elección de quién se decida a seguir desarrollando aplicaciones haciendo uso de la placa diseñada en este PFC.



# Bibliografía

- [1] Microchip, *Datasheet of dsPIC33FJ256MC710A Microcontroller*  
<http://ww1.microchip.com/downloads/en/DeviceDoc/70594d.pdf>
- [2] Microchip, *16-bit MCU and DSC Programmer's Reference Manual*  
<http://ww1.microchip.com/downloads/en/DeviceDoc/70157D.pdf>
- [3] Microchip, *Application Note AN983*  
<http://ww1.microchip.com/downloads/en/AppNotes/ACInductionMotor00984a.pdf>
- [4] Microchip, *Driving an ACIM with the dsPIC DSC MCPWM Module*  
<http://ww1.microchip.com/downloads/en/DeviceDoc/93004A.pdf>
- [5] Microchip, *Measuring Speed and Position with the QEI Module*  
<http://ww1.microchip.com/downloads/en/devicedoc/93002A.pdf>
- [6] Microchip, *Explorer 16 Development Board User's Guide*  
<http://ww1.microchip.com/downloads/en/DeviceDoc/Explorer16UserGuide51589a.pdf>
- [7] PFC de Miguel Ángel Chica Sánchez,  
*Puente en H con control de corriente para una máquina de corriente continua*





# Apéndices



# A. Bits de configuración

Los bits de configuración, llamados también colectivamente "fuses" permiten reconfigurar ciertos aspectos generales del funcionamiento del microcontrolador, estos bits de configuración se guardan en una localización específica de la memoria de programa del PIC y se graban al mismo momento que el código de programación.

## A.1. Registros de configuración

### FBS

Este registro permite definir zonas con código protegido en la memoria de arranque.

U	U	U	U	U	U	U	U
—	—	—	—	—	—	—	—
bit 23				bit 16			

U	U	U	U	U	U	U	U
—	—	—	—	—	—	—	—
bit 15				bit 8			

R/P	R/P	U	U	R/P	R/P	R/P	R/P
RBS<1:0>		—	—	BSS<2:0>			BWRP
bit 7				bit 0			

## Bits de configuración

Es importante la configuración de los siguientes bits:

**RBS** (bits 7-6): Mediante estos bits se puede seleccionar una zona de memoria RAM de arranque donde almacenar código protegido. No se define ninguna zona. (0b11)

**BSS** (bits 3-1): Mediante estos bits se puede seleccionar una zona de la memoria Flash de arranque donde almacenar código protegido. No se define ninguna zona (0b011)

**BWRP** (bit 0): Mediante estos bits se puede activar o no la protección de escritura de la zona de memoria segura reservada en la Flash de arranque. Se desactiva la protección de escritura. (0b1)

## FSS

En este registro se pueden definir zonas de memoria con código protegido.

U	U	U	U	U	U	U	U
—	—	—	—	—	—	—	—
bit 23						bit 16	
U	U	U	U	U	U	U	U
—	—	—	—	—	—	—	—
bit 15						bit 8	
R/P	R/P	U	U	R/P	R/P	R/P	R/P
RSS<1:0>		—	—	SSS<2:0>		SWRP	
bit 7						bit 0	

Se configuran los siguientes bits:

**RSS** (Bits 7-6): Mediante estos bits se puede seleccionar una zona de memoria RAM donde almacenar código protegido. No se define zona. (0b11)

**SSS** (Bits 3-1): Mediante estos bits se puede seleccionar una zona de la memoria Flash donde almacenar código protegido. No se define zona. (0b011)

**SWRP** (Bit 0): Mediante estos bits se puede activar o no la protección de escritura de la zona de memoria segura reservada en la Flash. Se desactiva la protección de escritura. (0b1)

## FOSCEL

En este registro se elige la fuente del oscilador y también el modo de arranque del oscilador.

U	U	U	U	U	U	U	U
—	—	—	—	—	—	—	—
bit 23				bit 16			
U	U	U	U	U	U	U	U
—	—	—	—	—	—	—	—
bit 15				bit 8			
R/P	U	U	U	U	R/P	R/P	R/P
IESO	—	—	—	—	FNOSC<2:0>		
bit 7				bit 0			

Es necesaria la configuración de los siguientes bits:

**IESO** (Bit 7): Mediante estos bits se selecciona el modo de arranque del oscilador. En este caso, el dsPIC arranca con el oscilador FRC interno y posteriormente pasa al seleccionado en FNOSC. (0b1)

**FNOSC** (Bits 2-0): Mediante estos bits se selecciona la fuente del oscilador. Se elige oscilador externo con multiplicador PLL. (0b011)

## FOSC

Este registro permite seleccionar el tipo de oscilador externo así como la función del pin OSC2.

U	U	U	U	U	U	U	U
—	—	—	—	—	—	—	—
bit 23				bit 16			
U	U	U	U	U	U	U	U
—	—	—	—	—	—	—	—
bit 15				bit 8			
R/P	R/P	R/P	U	U	R/P	R/P	R/P
FCKSM<1:0>		IOL1WAY	—	—	OSCIOFNC	POSCMD<1:0>	
bit 7				bit 0			

Se configuran los siguientes bits:

**FCKSM** (Bits 7-6): Mediante estos bits se configura el *Clock Switching* y el *Fail-Safe Clock Monitor*. Se desactivan ambas opciones. (0b10)

**OSCIOFNC** (Bit 2): Permite utilizar el pin OSC2 como pin de entrada/-salida o como pin del oscilador externo. Se asigna el pin OSC2 al oscilador externo. (0b1)

**POSCMD** (Bits 1-0): Mediante estos bits seleccionamos el modo de operación del oscilador primario en función a su frecuencia. Para un cristal de 8 MHz se debe seleccionar el modo XT. (0b01)

## FWDT

Permite la activación del *WatchDog Timer* o “Perro Guardián”. El objetivo del *WatchDog* es prevenir que el microcontrolador se quede indefinidamente bloqueado o “colgado” debido a errores en la lógica del software.

U	U	U	U	U	U	U	U
—	—	—	—	—	—	—	—
bit 23						bit 16	
U	U	U	U	U	U	U	U
—	—	—	—	—	—	—	—
bit 15						bit 8	
R/P	R/P	R/P	U	U	R/P	R/P	R/P
FWDTEN	WINDIS	—	WDTPRE		WDTPOST<3:0>		
bit 7						bit 0	

Se configuran los siguientes bits:

**FWDTEN** (Bit 7): Bit para la activación o desactivación del WatchDog Timer. Se desactiva esta opción. (0b0)

## FGS

Permiten activar las protecciones de escritura e incluso de lectura del código de programa.

U	U	U	U	U	U	U	U
—	—	—	—	—	—	—	—
bit 23						bit 16	
U	U	U	U	U	U	U	U
—	—	—	—	—	—	—	—
bit 15						bit 8	
U	U	U	U	U	R/P	R/P	R/P
—	—	—	—	—	GSS<1:0>		GWRP
bit 7						bit 0	

*Bits de configuración*

Se configuran los siguientes bits:

**GSS** (Bits 2-1): Mediante estos bits se selecciona el nivel de protección. Se puede seleccionar alta protección, estándar e incluso ningún tipo de protección, tal y como se hace en este caso, por tanto, queda desactivada la protección del código. (0b11)

**GWRP** (Bit 0): Con este bit se selecciona la protección de escritura del código de programa. Se deshabilita para la realización del PFC. (0b1)

**FPOR**

Este registro permite establecer la configuración de los pines PWM.

U	U	U	U	U	U	U	U
—	—	—	—	—	—	—	—
bit 23						bit 16	

U	U	U	U	U	U	U	U
—	—	—	—	—	—	—	—
bit 15						bit 8	

R/P	R/P	R/P	R/P	R/P	R/P	R/P	R/P
PWMPIN	HPOL	LPOL	ALT12C	BOREN	FPWRT<2:0>		
bit 7						bit 0	

Es importante la configuración de los siguientes bits:

**PWMPIN** (Bit 7): Permite seleccionar quien controla los pines del módulo PWM después de un reinicio. Se selecciona que sean controlados por los registros de puerto. (0b1)

**HPOL** (Bit 6): Permite seleccionar el modo de funcionamiento de los pines PWM-HIGH entre activos en alto o activos en bajo. Activando este bit los pines se activan ante un nivel alto y siempre arrancan en estado bajo. (0b1)

**LPOL** (Bit 5): Permite seleccionar el modo de funcionamiento de los pines PWM-LOW entre activos en alto o activos en bajo. Activando este bit los



pinos se activan ante un nivel alto y siempre arrancan en estado bajo. (0b1)

**FPWRT** (Bits 2-0): Permite establecer un tiempo de retardo al encendido ante un reinicio. Se selecciona un tiempo de 128 ms. (0b111)



## A.2. Ejemplo de configuración

```
_FBS (RBS_NO_RAM & BSS_NO_FLASH & BWRP_WRPROTECT_OFF);  
    // No se definen zonas protegidas en la memoria  
    RAM de arranque  
    // No se definen zonas protegidas en la memoria  
    Flash de arranque  
    // Protección de escritura en memoria de  
    arranque desactivada  
  
_FSS (RSS_NO_RAM & SSS_NO_FLASH & SWRP_WRPROTECT_OFF);  
    // No se definen zonas protegidas en la memoria  
    RAM  
    // No se definen zonas protegidas en la memoria  
    Flash  
    // Protección de escritura de programa  
    desactivada  
  
_FOSCSEL (IESO_ON & FNOOSC_PRIPLL);  
    // El dsPIC arranca con el oscilador interno  
    // y posteriormente pasa al externo  
    // Activado oscilador externo con PLL  
  
_FOSC (FCKSM_CSDCMD & OSCIOFNC_OFF & POSCMD_XT);  
    // Clock switch & Monitoring desactivado  
    // Pin OSC2 asignado al oscilador  
    // Oscilador en modo XT (Cristal entre 3 y 10  
    MHz)  
  
_FWDT (FWDTEN_OFF);  
    // Watchdog Timer desactivado
```

```
_FGS (GSS_OFF & GCP_OFF & GWRP_OFF);  
    // Protección de código desactivada  
    // Protección de escritura desactivada  
  
_FPOR (PWMPIN_ON & HPOL_ON & LPOL_ON & FPWRT_PWR128);  
    // Los pines PWM se controlan por registros de  
    // puerto  
    // PWM High & Low activos en alta  
    // FPOR Power On reset 128ms
```

## B. Configuración del oscilador

En este apartado se explican los registros utilizados para configurar el módulo oscilador según las especificaciones consideradas en el *Capítulo 3.2: Módulo oscilador*. Aparte de estos registros, ya se han configurado y explicado previamente otros relacionados con el módulo en el *Apéndice A: Bits de configuración*.

### B.1. Registros de configuración

#### CLKDIV

En este registro se configuran los divisores N1 y N2 para configurar el módulo PLL según los requisitos deseados.

R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0
ROI	DOZE<2:0>		DOZEN <sup>(1)</sup>		FRCDIV<2:0>		
bit 15							bit 8
R/W-0	R/W-1	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PLLPOST<1:0>		—	PLLPRE<4:0>				
bit 7							bit 0

En ese registro se configuran los siguientes bits:

**PLLPOST** (bits 7-6): Corresponde con el valor del divisor postcaler PLL

denominado N2. Tal y como hemos visto anteriormente se establece un  $N2 = 2$ . (0b00) **PLLPRE** (bits 0-4): Corresponde con el valor del divisor prescaler PLL denominado N1. Para conseguir los 40MIPS se establece un valor de  $N1 = 2$ . (0b00)

## PLLFBD

En este registro se configura el factor M (2..513), por el que se multiplica la entrada PLL.

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	PLLDIV<8>
bit 15							bit 8
R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0
PLLDIV<7:0>							
bit 7							bit 0

Se configuran los siguientes bits:

PLLDIV (bits 8-0): Selección del factor de multiplicación M. Se elige  $M=40$ . (0b0010 0110)

# C. Configuración del módulo ADC

## C.1. Registros de configuración

A continuación se describen los distintos registros que se han de configurar para poner en funcionamiento la conversión analógica/digital.

### AD1CON1

Permite activar o desactivar el módulo ADC, seleccionar entre el modo de 10-bits o 12-bits, configurar el formato de datos de salida y utilizar o no el modo autoconversión. En el modo de 10-bits se puede seleccionar usar un canal, dos o cuatro canales y no todos los pines del microcontrolador pueden usarse para cualquier canal, además en este modo se pueden muestrear hasta 4 canales de forma simultánea, eso sí, requiere utilizar el módulo DMA para leer el buffer de datos.

R/W-0	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
ADON	—	ADSIDL	ADDMABM	—	AD12B	FORM<1:0>	
bit 15							bit 8
R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0, HC,HS	R/C-0, HC, HS
SSRC<2:0>			—	SIMSAM	ASAM	SAMP	DONE
bit 7							bit 0

Es imprescindible la configuración de los siguientes bits:

**ADON** (bit 15): Bit para activar o desactivar el modulo ADC. Para la configuración del módulo debe estar desactivado, a cero. (0b0)

**AD12B** (bit 10): Selección del modo de muestreo de 12-bits. Si se activa este modo no se pueden muestrear varias entradas de forma simultánea. De todos modos, se elige esta configuración ya que ofrece una mayor resolución y no muestrearemos señales de forma simultánea. (0b1)

**FORM** (bits 9-8): Configura el formato de los datos de salida. Seleccionamos el modo entero. (0b00)

**SSRC** (bit 7-5): *Sample Clock Source Select* bits. Permite seleccionar el origen de los pulsos que se toman como referencia para el muestreo, y una vez finalizado, se inicia la conversión. Se elige como referencia el contador interno. (0b111)

**ASAM** (bit 2): Si se pone a uno, el muestreo empieza inmediatamente después de la última conversión, a cero, el muestreo comienza cuando el bit SAMP esté a uno. (0b1)

**SAMP** (bit 1): Bit para la activación del muestreo, si ASAM=1, este bit se activa automáticamente por hardware (auto-conversion). (0b0)

## AD1CON2

En este registro se pueden seleccionar las tensiones de referencia VREFH y VREFL para las conversiones A/D, ya sea mediante valores externos conectados a los pines o usando los valores internos del microcontrolador AVDD y AVSS.

R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	
VCFG<2:0>			—	—	CSCNA	CHPS<1:0>		
bit 15								bit 8
R-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
BUFS	—	SMPI<3:0>				BUFM	ALTS	
bit 7								bit 0



Los bits a configurar son los siguientes:

**VCFG** (bits 15-13): Selección de la tensión de referencia en la conversión. En este caso, AVDD = 3.3V y AVSS = 0V (0b000)

**CSCNA** (bit 10): Escaneo de las entradas del canal 0. Para ello el bit debe estar a uno. (0b1)

**BUFS** (bit 7): Bit de lectura sobre el estado del llenado del buffer. Si está a cero el ADC está llenando la primera mitad del buffer, si está a uno, el ADC está llenando la segunda mitad del buffer.

### AD1CON3

En este registro se configura el reloj de conversión analógica para que coincida la velocidad de muestreo de los datos con el reloj del procesador.

R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADRC	—	—	SAMC<4:0> <sup>(1)</sup>				
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADCS<7:0> <sup>(2)</sup>							
bit 7							bit 0

Se configuran los siguientes bits del registro AD1CON3:

**ADRC** (bit 15): Permite seleccionar entre reloj interno RC o reloj del sistema, seleccionamos el reloj RC interno. (0b1)

**SAMC** (bits 12-8): Periodo de tiempo para realizar los muestreos. Se selecciona un periodo de tiempo de 14 TAD. (0b01110)

### AD1CSSL

Este registro permite elegir las entradas a escanear y muestrear por el módulo ADC. Si se elige el modo de muestreo de 12-bits las entradas serán

## Configuración del módulo ADC

escaneadas una tras otra, nunca de forma simultánea.

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CSS15	CSS14	CSS13	CSS12	CSS11	CSS10	CSS9	CSS8
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CSS7	CSS6	CSS5	CSS4	CSS3	CSS2	CSS1	CSS0
bit 7							bit 0

Se configuran los siguientes bits:

**CSSx** (bits 15-0): Se seleccionan las entradas a escanear. Inicialmente se selecciona la entrada AN0, y una vez realizada la conversión se selecciona la siguiente entrada. (0b0000000000000001)

### AD1PCFGL

Mediante este registro se especifica si una entrada es del tipo analógico o digital. Cuando se configura un pin como entrada analógica, también hay que configurarlo como entrada usando el registro TRIS.

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCFG15	PCFG14	PCFG13	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

Se configuran los siguientes bits:

**PCFGx** (bits 15-0): Con PCFGx=0 configuramos el pin x como entrada analógica. Configuramos los pines 0, 1, 2 y 8 como entradas analógicas.

## C.2. Ejemplo de Inicialización

```
AD1PCFGLbits.PCFG0 = 0; // Configuro AN0 (I fase U)
    como entrada analógica
AD1PCFGLbits.PCFG1 = 0; // Configuro AN1 (I fase V)
    como entrada analógica
AD1PCFGLbits.PCFG2 = 0; // Configuro AN2 (Tensión de
    bus) como entrada analógica
AD1PCFGLbits.PCFG8 = 0; // Configuro AN8 (Potenciómetro
    ) como entrada analógica

AD1CON1 = 0x04E4; // Finalizado el muestreo se
    inicia la conversión, 12-bits de resolución y datos
    de salida de tipo entero
AD1CON1bits.AD12B = 1; // 12-bits de resolución de
    muestreo
AD1CON2 = 0x0400; // Se produce una interrupción
    después de cada muestreo, escaneo el Canal 0,
    referencia entre Vcc y Vss
AD1CON3 = 0x8E00; // Tiempo de muestreo 14TAD,
    uso reloj interno RC como base de tiempos
AD1CSSL = 0x0001; // De forma inicial se
    selecciona AN0 como entrada a escanear.

IFS0bits.AD1IF = 0; // Borro el flag de
    interrupción
IEC0bits.AD1IE = 1; // Activo interrupciones por
    ADC

AD1CON1bits.ADON = 1; // Una vez configurado activo
    el módulo ADC
```



# D. Configuración del módulo PWM

## D.1. Registros de configuración

El microcontrolador posee distintos registros con distintos bits para controlar el modo de operación del módulo PWM. La información de todos los registros se encuentra en la hoja de características del microcontrolador. Se explicarán los utilizados, el resto, al no modificarse, quedarán en su valor por defecto (cero), y por tanto, en el modo de funcionamiento correspondiente al cero.

### P1TCON

En el registro P1TCON se selecciona el modo de funcionamiento y se controla el reloj del PWM.

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
PTEN	—	PTSIDL	—	—	—	—	—
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTOPS<3:0>			PTCKPS<1:0>		PTMOD<1:0>		
bit 7							bit 0

Se configuran los siguientes bits:

**PTEN** (bit 15): Enciende el módulo PWM (0b1)

**PTMOD** (bits 1-0): Selecciona el modo de funcionamiento del módulo PWM. Seleccionamos el modo Continuous Up/Down Count para generar las salidas PWM en el modo Center-Aligned explicado en el punto 3.4, pág. 25 (0b10)

## PTPER

En el registro P1TPER se escribe el valor del periodo de la PWM.

## PWM1CON1

En el registro PWM1CON1 se seleccionan los pines por los cuales saldrá el PWM, se pueden elegir dos pines y hacer que funcionen en modo independiente o complementario.

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	PMOD4	PMOD3	PMOD2	PMOD1
bit 15							bit 8
R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
PEN4H <sup>(1)</sup>	PEN3H <sup>(1)</sup>	PEN2H <sup>(1)</sup>	PEN1H <sup>(1)</sup>	PEN4L <sup>(1)</sup>	PEN3L <sup>(1)</sup>	PEN2L <sup>(1)</sup>	PEN1L <sup>(1)</sup>
bit 7							bit 0

Se configuran los siguientes bits:

**PMOD** (bits 11-8): Activa las salidas del módulo PWM en modo complementario (0b0000)

**PEN4H:PEN1H** (bits 7-4): Si los bits (altos) están a “1” habilitan los pines PWM como salidas de la PWM, si por el contrario, están a cero se configuran como pines de entrada/salida (0b0111)

**PEN4L:PEN1L** (bits 3-0): Se configura de la misma manera que el anterior, estos bits constituyen los bits más bajos (0b0111)

## PWM1CON2

Aun que este registro tiene varias opciones, en este caso será utilizado únicamente para la actualización del ciclo de trabajo.

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	SEVOPS<3:0>			
bit 15				bit 8			

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	IUE	OSYNC	UDIS
bit 7				bit 0			

Se configuran los siguientes bits:

**UDIS** (bit 0): Si el bit está a “1” desactiva la actualización del ciclo de trabajo (0b1), Se inicializa el código con dicho bit desactivado.

## P1DTCON1

En el registro P1DTCON1 se configura el periodo de reloj del tiempo muerto.

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DTBPS<1:0>		DTB<5:0>					
bit 15				bit 8			

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DTAPS<1:0>		DTA<5:0>					
bit 7				bit 0			

Se configuran los siguientes bits:

**DTAPS** (bits 7-6): Selección del prescaler para el tiempo muerto. Seleccionamos un prescaler de 2 (0b01).

**DTA** (bits 5-0): Asigna el valor del tiempo muerto, que será multiplicado por FCY/prescaler. Por tanto, se obtiene un tiempo muerto de 3 $\mu$ s (0b011110).

## PDC1, PDC2, PDC3

En el registro PDC1, PDC2 y PDC3 se escribe el valor, en 16 bits, del ciclo de trabajo de la salida PWM para cada fase.

## P1OVDCON

En el registro P1OVDCON se seleccionan los pines PWM que serán controlados por el módulo PWM.

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
POVD4H	POVD4L	POVD3H	POVD3L	POVD2H	POVD2L	POVD1H	POVD1L
bit 15						bit 8	

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
POUT4H	POUT4L	POUT3H	POUT3L	POUT2H	POUT2L	POUT1H	POUT1L
bit 7						bit 0	

Se configuran los siguientes bits:

**POVDxH: POVDxL** (bits 15-8): Si los bits están a uno los pines PWM los controla el módulo PWM, si por el contrario están a cero, serán activados o no en función del valor de los bits POUTxH Y POUTxL. (0b0011 1111)

**POUTxH: POUTxL** (bits 7-0): Si este bit está a uno la salida correspondiente PWM estará activa, si está a cero será desactivada. El estado de estos bits no tiene relevancia alguna si los bits de POVD están a uno. (0b0000 0000)



## D.2. Ejemplo de Inicialización

```
P10VDCON = 0x3F00;
P1DTCON1 = 0x5E;           // 3uS de tiempo muerto
PWM1CON1 = 0x77;           // Activo salidas PWM en
    modo complementario
PWM1CON2 = 1;               // Actualizaciones del
    ciclo de trabajo desactivadas al inicio

PDC1 = PTPER;               // Cargo el valor del
    periodo en cada ciclo de trabajo
PDC2 = PTPER;               // equivalente a tener
    un Duty Cycle = 0 al estar
PDC3 = PTPER;               // configurado el modo
    Center-Aligned

IFS3bits.PWMIF = 0;         // Borro el flag de
    interrupción
IPC14bits.PWMIP = 7;        // Máxima prioridad para la
    interrupción por PWM
IEC3bits.PWMIE = 1;         // Activo interrupciones
    por PWM

P1TCONbits.PTMOD = 2;       // Selecciono modo Center-
    Aligned
P1TCONbits.PTEN = 1;        // Activo el módulo PWM
```



# E. Configuración del módulo QEI

## E.1. Registros de configuración

El módulo QEI tiene cuatro registros accesibles para el usuario. Para poner en marcha este módulo, se deberán configurar estos registros adecuadamente según las necesidades.

### QEI1CON

El QEI Control Register, controla las operaciones QEI y proporciona flags de estado del modulo.

R/W-0	U-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
CNTERR	—	QEISIDL	INDEX	UPDN	QEIM<2:0>		
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SWPAB	PCDOUT	TQGATE	TQCKPS<1:0>	POSRES	TQCS	UPDN_SRC <sup>(1)</sup>	
bit 7							bit 0

Se configuran los siguientes bits:

**CNTERR** (bit 15): Indica si ha ocurrido un error en el Status. Se puede reiniciar poniendo el bit a cero. (0b0)

**QEISIDL** (bit 13): Permite mantener o no funcionando el módulo QEI

## Configuración del módulo QEI

aunque el micro entre en modo espera o Idle. Si el bit está a cero el módulo QEI continuará trabajando aunque entremos en modo espera. (0b0)

**QEIM** (bits 10-8): Permite seleccionar el modo de trabajo del módulo QEI. Se selecciona el modo contador x4 con reset por pulso de Index. Si se pone a cero se apaga el modulo. (0b110)

**SWPAB** (bit 7): Permite intercambiar entre si las entradas QEA y QEB. Si este bit se encuentra a uno se produce un intercambio entre dichos canales, si por el contrario está a cero no se intercambian. (0b0)

**PCDOUT** (bit 6): *Position Counter Direction State Output Enable* bit. Al habilitar este bit obtendremos un estado del sentido de giro en el pin RD7 (0b0)

**POSRES** (bit 2): *Position Counter Reset Enable* bit. Si está a uno, un pulso en Index resetea POSCNT, si está a cero se desactiva esta función. (0b1)

## DFLTCON

El Digital Filter Control Register, controla las operaciones del filtro digital del módulo QEI.

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	IMV<2:0>		CEID
bit 15							bit 8
R/W-0	R/W-0		U-0	U-0	U-0	U-0	
QEOUT	QECK<2:0>		—	—	—	—	
bit 7							bit 0

Se configuran los siguientes bits:

**CEID** (bit 8): *Count Error Interrupt Disable* bit. Si dicho bit está a uno se desactivan las interrupciones por errores, si está a cero, se activan. (0b1)

**QEOUT** (bit 7): *Digital Filter Output Enable* bit. Si se encuentra a uno

las salidas digitales del filtro están activas, si se encuentra a cero, las salidas están deshabilitadas. (0b1)

**QUECK** (bits 6-4): *Digital Filter Clock Divide Select* bits. Selección del periodo de filtrado. En este caso se adopta un filtrado de 1:16 FCY. (0b011)

### **POSCNT**

El Position Count Register, permite la lectura y escritura del contador de posición de 16 bits. Cuando este registro llega a su valor máximo, vuelve a valer cero y sigue incrementándose. Esto puede provocar problemas a la hora de calcular el valor de pulsos transcurrido en un intervalo de tiempo, al hacer la resta entre el valor del inicio y del final del intervalo el registro puede reiniciarse provocando un resta errónea desde el punto de vista del algoritmo de control.

## E.2. Ejemplo de Inicialización

```
QEICONbits.QEIM = 0;           // Apago el módulo QEI
    para configurarlo
QEICONbits.CNTERR = 0;        // Borro cualquier error
    previo
QEICONbits.QEISIDL = 0;       // Sigue contando aunque
    se entre en modo espera
QEICONbits.SWPAB = 0;         // Canal A y canal B no
    intercambiados
QEICONbits.PCDOUT = 0;        // Desactivado el uso
    del pin RD7 como indicador de sentido de giro.
QEICONbits.POSRES = 1;        // Un pulso en INDEX
    hace un reset
DFLTCNbits.CEID = 1;          // Desactivo
    interrupciones por errores
DFLTCNbits.QEOUT = 1;         // Activo el uso de
    filtros digitales
DFLTCNbits.QECK =0b011;       // Aplico un periodo de
    filtrado de 1:16 veces FCY
POSCNT = 0;                   // Pongo el
    valor del contador a cero
QEICONbits.QEIM = 0b110;      // Modo resolución x4 y
    reset por pulso en INDEX
```

# F. Configuración del Timer1 y Timer2

## F.1. Registros de configuración

A continuación se explican los registros de configuración del Timer 1, análogos al resto de Timers.

### T1CON

Mediante este registro se selecciona el período del Timer, así como el prescaler y la fuente de pulsos de reloj. También permite activar o desactivar el Timer.

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON	—	TSIDL	—	—	—	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	U-0
—	TGATE	TCKPS<1:0>		—	TSYNC	TCS	—
bit 7							bit 0

Se configuran los siguientes bits:

**TON** (bit 15): Bit para activar o desactivar el Timer. Para la configuración debe estar desactivado, a cero.

## Configuración del Timer1 y Timer2

**TS1DL** (bit 13): Si el bit 13 está a uno el Timer 1 no funciona mientras el micro está en modo espera, si por el contrario, está a 0, el Timer continúa trabajando durante el modo espera. (0b0)

**TCKPS** (bits 5-4): Selección del prescaler del Timer 1. Se selecciona un prescaler de 1:256 FCY. (0b11)

**TCS** (bit 1): Selección del reloj del Timer 1. Si está puesto a uno se utiliza el pin T1CK como entrada de reloj, si está a cero la base de tiempos se toma del reloj interno (FCY). (0b0)

**PR1**: En este registro se carga el valor del periodo del Timer. La ecuación F.1 muestra como calcular el valor de PR1 a partir del periodo de tiempo que queremos cargar en el Timer.

$$PR1 = \frac{FCY \cdot T}{TCKPS} \quad (F.1)$$

Siendo T el período deseado.



## F.2. Ejemplo de Inicialización

### TIMER 1

```
TMR1 = 0; // Se resetea el Timer 1
T1CONbits.TON = 0; // Apagado del Timer 1
T1CONbits.TSIDL = 0; // El Timer 1 sigue trabajando
    en modo espera
T1CONbits.TCS = 0; // Se selecciona FCY como base
    de tiempos
T1CONbits.TCKPS = 3; // Prescaler de 256
PR1 = 15625; // 100ms de periodo

IFS0bits.T1IF = 0; // Borro flag de interrupción
IPC0bits.T1IP = 4; // Prioridad 4
IEC0bits.T1IE = 1; // Se activan interrupciones por
    Timer 1

T1CONbits.TON = 1; // Se enciende el Timer 1
```

### TIMER 2

```
TMR2 = 0; // Se resetea el Timer 2
T2CONbits.TON = 0; // Apagado del Timer 2
T2CONbits.TSIDL = 0; // El Timer 2 sigue trabajando
    en modo espera
T2CONbits.TCS = 0; // Se selecciona FCY como base
    de tiempos
T2CONbits.TCKPS = 1; // Prescaler de 8
PR2 = 15625; // 3.125ms de periodo

IFS0bits.T2IF = 0; // Borro flag de interrupción
IPC1bits.T2IP = 7; // Prioridad 5
```

## *Configuración del Timer1 y Timer2*

---

```
IEC0bits.T2IE = 1;      // Se activan interrupciones por  
    Timer 2  
  
T2CONbits.TON = 1;     // Se enciende Timer 2
```

# G. Programa

## G.1. main.c

```
#include "p33fj256mc710.h"
#include "config.h"
#include "PWM_Init.h"
#include "ADC_Init.h"
#include "QEI_Init.h"
#include "Init_TMR1.h"
#include "Init_TMR2.h"
#include "arranque.h"
#include "lcd_explorer.h"
#include <libpic30.h>
#include <stdio.h>
#include <math.h>

#define FCY      40000000
// Frecuencia de operación 40 [MIPS]
#define FPWM     20000
// Frecuencia de conmutación [Hz]
#define Vbase    230
// Tensión base (valor fase eficaz) [V]
```

```
#define Vmax      230
// Tensión máxima que puedo dar del bus (valor fase
// eficaz) [V]
#define Fmin      5
// Frecuencia mínima [Hz]
#define Fmax      80
// Frecuencia máxima [Hz]
#define Fbase     50
// Frecuencia base de trabajo [Hz]

#define _120_GRADOS      0x5555
#define _240_GRADOS      0xAAAA

#define MAX_RPM  2400
// Velocidad máxima del motor [RPM]
#define MAX_CNT_PER_REV (1024 * 4 - 1)
// Número de pasos por vuelta (pasos encoder * x4 - 1)
#define MAXSPEED (unsigned int)(((unsigned long)
    MAX_CNT_PER_REV*2048)/256)
// Velocidad máxima escalada a enteros de 16-bits
#define HALFMXSPEED (MAXSPEED>>1)
// La mitad de la velocidad máxima en 16-bits

#define PSGIRO          PORTFbits.RF6
// Pulsador de cambio de sentido de giro
#define APAGADO          PORTDbits.RD12
// Estado del convertidor

// Tabla con 64 valores de la función seno escalados a
// números enteros

const int TablaSeno [64] = {
    0,3212,6393,9512,12539,15446,18204,20787,23170,
```

```
25329,27245,28898,30273,31356,32137,32609,32767,
32609,32137,31356,30273,28898,27245,25329,23170,
20787,18204,15446,12539,9512,6393,3212,0,-3212,
-6393,-9512,-12539,-15446,-18204,-20787,-23170,
-25329,-27245,-28898,-30273,-31356,-32137,-32609,
-32767,-32609,-32137,-31356,-30273,-28898,-27245,
-25329,-23170,-20787,-18204,-15446,-12539,-9512,
-6393,-3212};

// Variables
int ValorSeno, TPWM, tension, frec, vdc, entrada=0,
    AngPos[2] = {0,0}, POSCNTcopy=0, Speed, rpms;
long _TENSION, i_med_u, i_med_v, vdc_med, pot_med;
unsigned int _DELTA, Angulo=0;
float cte, vdcf, pot_medf, iu, iuf, iv, ivf, iw, ia, ib,
    iabs, iu_max=0, iu_rms, iuf_rms, iv_rms, iv_max=0,
    ivf_rms, Speedf;
char rpm_lcd[20], imedv_lcd[20], vdc_lcd[20], frec_lcd
    [20], dir=0, dir_prev=0, sentido[5];

// Definición de funciones
void __attribute__((__interrupt__, no_auto_psv))
    _ADC1Interrupt(void);
void __attribute__((__interrupt__, auto_psv))
    _T1Interrupt(void);
void __attribute__((__interrupt__, auto_psv))
    _T2Interrupt(void);
void __attribute__((__interrupt__, auto_psv))
    _PWMInterrupt(void);
void arranque(void);
void Init_TMR1(void);
void Init_TMR2(void);
void PWMInit(void);
```

```
void ADCInit(void);
void QEIInit(void);

// Interrupción por ADC
void __attribute__((__interrupt__)) _ADC1Interrupt (void
){

switch(entrada){          // Leo una entrada analógica
                        // por cada interrupción

case 0:
    i_med_u = ADC1BUF0;
    // Almaceno el valor de la corriente fase u
    iu = (((i_med_u+73)*3.3)/4095)-1.65)/0.122;
    // Escalo corriente y compenso error
    if (iu_max < iu) iu_max = iu;
    // Me quedo con el valor pico
    iu_rms = iu_max/sqrt(2);
    // Calculo el valor eficaz
    iuf_rms = iu_rms*0.005+iuf_rms*0.995;
    // Filtro la corriente
    iu_max = iu_max * 0.999;
    // Voy disminuyendo el valor máximo almacenado
    AD1CON1bits.ADON = 0;
    // Paro el módulo ADC
    AD1CSSL = 0x0002;
    // Cambio el canal a muestrear a AN1
    AD1CON1bits.ADON = 1;
    // Inicio el módulo ADC
    break;

case 1:
    i_med_v = ADC1BUF0;
```

```
// Almaceno el valor de la corriente fase v
iv = (((i_med_v+12)*3.3)/4095)-1.65)/0.122;
// Escalo corriente y compenso error
if (iv_max < iv) iv_max = iv;
iv_rms = iv_max/sqrt(2);
ivf_rms = iv_rms*0.005+ivf_rms*0.995;
iv_max = iv_max * 0.999;
AD1CON1bits.ADON = 0;
// Paro el módulo ADC
AD1CSSL = 0x0004;
// Cambio el canal a muestrear a AN2
AD1CON1bits.ADON = 1;
// Inicio el módulo ADC
break;

case 2:
vdc_med = ADC1BUF0;
// Almaceno el valor de la tensión de bus
vdc = ((vdc_med-50) * 0.1640625);
// Escalo y compenso el error
vdcf = vdc*0.01+vdcf*0.99;
// Filtro la tensión de bus
AD1CON1bits.ADON = 0;
// Paro el módulo ADC
AD1CSSL = 0x0100;
// Cambio el canal a muestrear a AN8
AD1CON1bits.ADON = 1;
// Inicio el módulo ADC
break;

case 3:
pot_med = ADC1BUF0;
// Almaceno el valor del potenciómetro
```

```
        pot_medf = pot_med*0.01+pot_medf*0.99;
        // Filtro el potenciometro
        AD1CON1bits.ADON = 0;
        // Paro el módulo ADC
        AD1CSSL = 0x0001;
        // Cambio el canal a muestrear a AN0
        AD1CON1bits.ADON = 1;
        // Inicio el módulo ADC
        break;
default:
        break;
}

        entrada++;
        if (entrada>3) entrada=0;
        // Para volver al escanear el canal AN0

IFS0bits.AD1IF = 0;
// Borro el flag de interrupción
return;
}

// Interrupción por TIMER1
void __attribute__((__interrupt__, auto_psv))
    _T1Interrupt(void) {

if (!PSGIRO && APAGADO && !dir_prev) dir = !dir;
// Cambio sentido de giro
if (dir)
        sprintf(sentido,"IZQ ");
else
        sprintf(sentido,"DER ");
dir_prev = !PSGIRO;
```



```
// Almaceno el sentido de giro anterior

frec = Fmin + (((Fmax - Fmin) * pot_medf)/4095);
// Escalo la frecuencia entre Fmin-Fmax según valor pot
cte = Vbase / Fbase;
// Calculo la constante V/f
tension = cte*frec;
if (tension > Vmax) tension = Vmax;

_TENSION = tension * 65536 / Vmax;
// Escalo el valor de la tensión
_DELTA = (unsigned int)(frec * 65536 / FPWM);
// Escalo el valor de la frecuencia

rpms = (int)(((long)Speedf * MAX_RPM)/ MAXSPEED);

sprintf(rpm_lcd,"%s %4u %s %s", "w:",rpms,"RPM",sentido)
;
// Paso la variable rpms y sentido a string
sprintf(imedv_lcd,"%s %4.2f %s", "i_v:",ivf_rms,"A");
// Paso la variable ivf_rms a string
sprintf(vdc_lcd,"%s %3.0f %s", "Tension Bus:",vdcf,"V");
// Paso la variable vdcf a string
sprintf(frec_lcd,"%s %2u %s", "Frecuencia:",frec,"Hz");
// Paso la variable frec a string

gotoxy_lcd(1,1);
puts_lcd(rpm_lcd);
gotoxy_lcd(1,2);
puts_lcd(imedv_lcd);
gotoxy_lcd(1,3);
puts_lcd(vdc_lcd);
gotoxy_lcd(1,4);
```

```
puts_lcd(frec_lcd);

IFS0bits.T1IF = 0;
// Borro el flag de interrupción
}

// Interrupción por TIMER2 para el cálculo de la
// velocidad angular
void __attribute__((__interrupt__, auto_psv))
_T2Interrupt(void) {

POSCNTcopy = (int)POSCNT;
if (!(QEICON & 0x0800)) POSCNTcopy = 65536 - POSCNTcopy;

AngPos[1] = AngPos[0];
AngPos[0] = (unsigned int)((((unsigned long)POSCNTcopy *
2048)/256);
// 0 <= POSCNT <= 4095 to 0 <= AngPos <= 32760

Speed = AngPos[0] - AngPos[1];
if (Speed >= 0){
    if (Speed >= (HALFMAXSPEED))
        Speed = Speed - MAXSPEED;
}
else{
    if (Speed < -(HALFMAXSPEED))
        Speed = Speed + MAXSPEED;
}
Speed *= 8;
Speedf = 0.02*Speed+0.98*Speedf;
// Filtro la medida de la velocidad

IFS0bits.T2IF = 0;
```

```
// Borro el flag de interrupción
return;
}

// Interrupcion por PWM
void __attribute__((__interrupt__)) _PWMInterrupt(void){

    PWMCON2bits.UDIS = 1;
    // Inactivo las salidas PWM
    Angulo += _DELTA;
    // Voy aumentando el ángulo según _DELTA

    //Actualizo CicloTrabajo 1
    if (!dir)
    // Para cambio de sentido de giro
        ValorSeno = TablaSeno[Angulo >> 10];
    // Obtengo el valor del seno
    else
        ValorSeno = TablaSeno[(Angulo +
            _120_GRADOS) >> 10];

    TPWM = PTPER*2;
    ValorSeno = ((long)ValorSeno*(long)(_TENSION))
        >> 16;
    ValorSeno = ((long)ValorSeno*(long)TPWM) >> 16;
    PDC1 = ValorSeno + PTPER;
    // Elimino los valores negativos del ciclo de
    trabajo

    //Actualizo CicloTrabajo 2
    if (!dir)
    // Para cambio de sentido de giro
```

```
        ValorSeno = TablaSeno[(Angulo +
            _120_GRADOS) >> 10];
    else
        ValorSeno = TablaSeno[Angulo >> 10];
        // Obtengo el valor del seno

    TPWM = PTPER*2;
    ValorSeno = ((long)ValorSeno*(long)(_TENSION))
        >> 16;
    ValorSeno = ((long)ValorSeno*(long)TPWM) >> 16;
    PDC2 = ValorSeno + PTPER;
    // Elimino los valores negativos del ciclo de
        trabajo

    //Actualizo CicloTrabajo 3
    ValorSeno = TablaSeno[(Angulo + _240_GRADOS) >>
        10];
    //Obtengo el valor del seno
    TPWM = PTPER*2;
    ValorSeno = ((long)ValorSeno*(long)(_TENSION))
        >> 16;
    ValorSeno = ((long)ValorSeno*(long)TPWM) >> 16;
    PDC3 = ValorSeno + PTPER;
    // Elimino los valores negativos del ciclo de
        trabajo

    PWMCON2bits.UDIS = 0;
        // Reactivo salidas PWM
    IFS3bits.PWMIF = 0;
    // Borro el flag de interrupción
}
```

```
// PROGRAMA PRINCIPAL
int main (void){
//FCY = 1/2(XTAL*M/N1*N2) = 1/2*(8000000*40/2*2)= 40
    MIPS
    PLLFBDbits.PLLDIV = 0x26;
// Configuro PLLDIV(M) = 40
    CLKDIVbits.PLLPOST = 0b00;
// Configuro PLLPOST(N2) = 2
    PTPER = (FCY/FPWM - 1 ) >> 1;
// Calculo el periodo de la PWM

    TRISDbits.TRISD12 = 1;
// Configurado como entrada
    TRISFbits.TRISF6 = 1;
// Configurado como entrada

    arranque();
// Inicio el LCD
    ADCInit();
// Inicio el módulo ADC
    PWMInit();
// Inicio el módulo PWM
    QEIInit();
// Inicio el módulo QEI
    Init_TMR1();
// Inicio el Timer1
    Init_TMR2();
// Inicio el Timer2

    while (1);      //Bucle infinito
    return 0;
}
```



## G.2. arranque.h

En este archivo se encuentran las instrucciones para la inicialización del display LCD.

```
#include "p33FJ256MC710.h"
#include "lcd_explorer.h"
#define FCY 40000000
#include <libpic30.h>

unsigned int i;

void arranque(void){

    Init_LCD();
    home_clr();
    TRISA = 0xFF00;
    i=0;

    while(i<2){
        LATAbits.LATA0 = 1;
        LATAbits.LATA1 = 0;
        __delay_ms(150);
        LATAbits.LATA0 = 0;
        LATAbits.LATA1 = 1;
        __delay_ms(150);
        LATAbits.LATA0 = 0;
        LATAbits.LATA2 = 1;
        __delay_ms(150);
        LATAbits.LATA1 = 0;
        LATAbits.LATA3 = 1;
        __delay_ms(150);
    }
}
```

```
LATABits.LATA2 = 0;
LATABits.LATA4 = 1;
__delay_ms(150);
LATABits.LATA3 = 0;
LATABits.LATA4 = 1;
__delay_ms(150);
LATABits.LATA4 = 0;
LATABits.LATA3 = 1;
LATABits.LATA2 = 1;
__delay_ms(150);
LATABits.LATA3 = 0;
LATABits.LATA1 = 1;
__delay_ms(150);
LATABits.LATA2 = 0;
LATABits.LATA0 = 1;
__delay_ms(150);
i=i++;
}

while(i<8){
LATA = 0x0000;
LATA = 0x00FF;
__delay_ms(60);
LATA = 0x0000;
__delay_ms(60);
i=i++;
}

Init_LCD();
home_clr();

gotoxy_lcd(1,2);
puts_lcd(" PFC Realizado por:");
```



```
gotoxy_lcd(1,3);  
puts_lcd(" Sergio Sonora");  
__delay_s(2);  
home_clr();  
}  
  
//gotoxy_lcd(1,1);  
//puts_lcd("Velocidad: 2000 RPM");  
//gotoxy_lcd(1,4);  
//puts_lcd("I: 10 A      P: 3.5 kW");
```



## G.3. Librería LCD

lcd.h

### Prototipos de funciones LCD

```
void Init_LCD( void );  
// inicializa el lcd  
unsigned int busy_lcd( void );  
// espera al lcd  
void lcd_cmd( char cmd );  
// escribe comando en el lcd  
void lcd_data( char data );  
// escribe datos en el lcd  
void putc_lcd ( unsigned int data );  
// escribe datos en el lcd admitiendo borrado y salto de  
// linea  
void puts_lcd ( char *data );  
// escribe strings en el lcd  
void gotoxy_lcd ( unsigned int x, unsigned int y);  
// nos desplaza a una posicion x-y
```

### Prototipos de comandos LCD

```
#define cursor_right() lcd_cmd( 0x14 )  
#define cursor_left() lcd_cmd( 0x10 )  
#define display_shift() lcd_cmd( 0x1C )  
#define home_clr() lcd_cmd( 0x01 )  
#define home_it() lcd_cmd( 0x02 )  
#define line_2() lcd_cmd( 0xC0 )
```

## lcd.c

En este apartado se muestra la Librería LCD utilizada.

```
#include "p33FJ256MC710.h"
#include "lcd_explorer.h"
#define FCY 4000000
#include <libpic30.h>

// Control signal data pins
#define RW LATDbits.LATD9
// LCD R/W signal
#define RS LATDbits.LATD8
// LCD RS signal
#define E LATDbits.LATD10
// LCD E signal

// Control signal pin direction
#define RW_TRIS TRISDbits.TRISD9
#define RS_TRIS TRISDbits.TRISD8
#define E_TRIS TRISDbits.TRISD10

// Data signals and pin direction
#define DATA LATD // Port for LCD data
#define DATAPORT PORTD
#define TRISDATA TRISD // I/O setup for data Port

/***** LCD SUBROUTINE *****/

void Init_LCD( void )
// initialize LCD display
```

```
{
// 15mS delay after Vdd reaches nnVdc before proceeding
// with LCD initialization
// not always required and is based on system Vdd rise
// rate
__delay_ms(15);
// 15ms delay

/* set initial states for the data and control pins */
LATD &= 0xFF00;
RW = 0;
// R/W state set low
RS = 0;
// RS state set low
E = 0;
// E state set low

/* set data and control pins to outputs */
TRISD &= 0xFF00;
RW_TRIS = 0;
// RW pin set as output
RS_TRIS = 0;
// RS pin set as output
E_TRIS = 0;
// E pin set as output

/* 1st LCD initialization sequence */
DATA &= 0xFF00;
DATA |= 0x0038;
E = 1;
Nop();
Nop();
Nop();
```

```

E = 0;
// toggle E signal
__delay_ms(5);
// 5ms delay

/* 2nd LCD initialization sequence */
DATA &= 0xFF00;
DATA |= 0x0038;
E = 1;
Nop();
Nop();
Nop();
E = 0;
// toggle E signal
__delay_us(200);
// 200uS delay

/* 3rd LCD initialization sequence */
DATA &= 0xFF00;
DATA |= 0x0038;
E = 1;
Nop();
Nop();
Nop();
E = 0;
// toggle E signal
__delay_us(200);
// 200uS delay

lcd_cmd( 0x38 );
// function set
lcd_cmd( 0x0C );
// Display on/off control, cursor blink off (0x0C)

```

```
    lcd_cmd( 0x06 );
    // entry mode set (0x06)
}

void lcd_cmd( char cmd )
// subroutine for lcd commands
{
    // TRISD &= 0xFF00;
    // ensure RD0 - RD7 are outputs
    DATA &= 0xFF00;
    // prepare RD0 - RD7
    DATA |= cmd;
    // command byte to lcd
    RW = 0;
    // ensure RW is 0
    RS = 0;
    E = 1;
    // toggle E line
    Nop();
    Nop();
    Nop();
    E = 0;
    __delay_ms(5);
    // 5ms delay
}

void lcd_data( char data )
// subroutine for lcd data
{
    // TRISD &= 0xFF00;
    // ensure RD0 - RD7 are outputs
```

```

RW = 0;
// ensure RW is 0
RS = 1;
// assert register select to 1
DATA &= 0xFF00;
// prepare RD0 - RD7
DATA |= data;
// data byte to lcd
E = 1;
Nop();
Nop();
Nop();
E = 0;
// toggle E signal
RS = 0;
// negate register select to 0
__delay_us(400);
// 400uS delay
}

unsigned int busy_lcd( void )
// esperamos a que este preparado el lcd
{
    __delay_us(100);
    // Retardo de 100us
    return 0;
}

void gotoxy_lcd( unsigned int x, unsigned int y )
// nos desplaza a una posicion x-y
{
    unsigned int direccion;
    switch(y){

```



```
        case 1: direccion = 0x00; break;
        case 2: direccion = 0x40; break;
        case 3: direccion = 0x14; break;
        case 4: direccion = 0x54; break;
        default: direccion = 0x00; break;
    }

    direccion += x - 1;
    while( busy_lcd() );
    lcd_cmd( 0x80 | direccion );
}

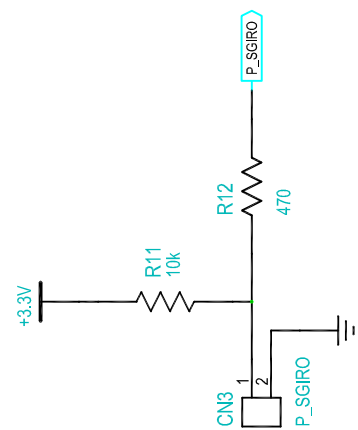
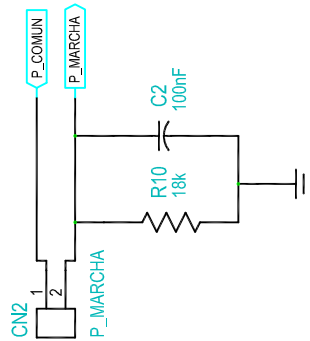
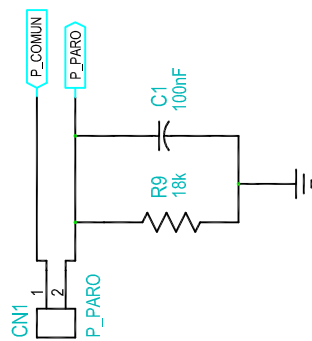
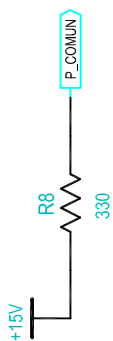
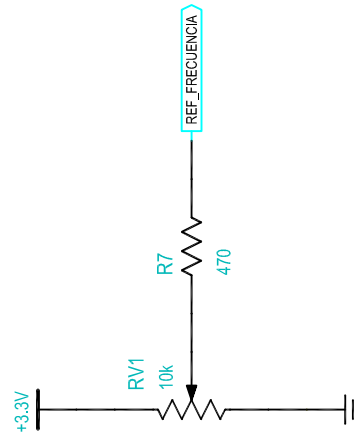
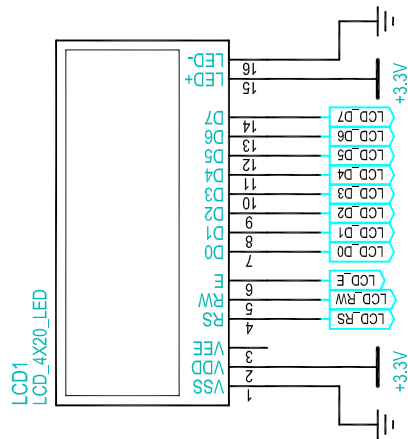
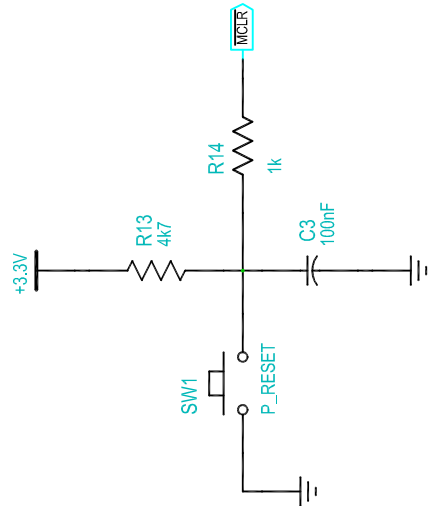
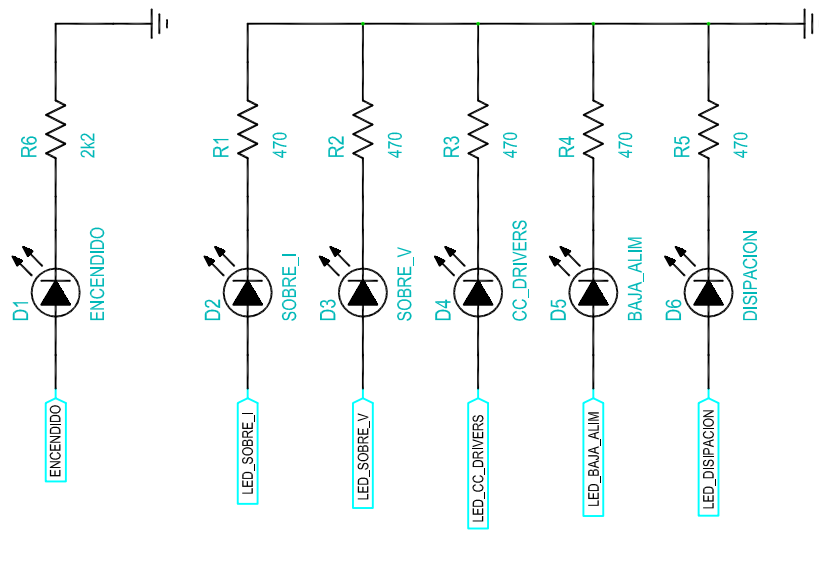
void putc_lcd( unsigned int Data )
// usar con \f para borrar lcd o \n para salto de linea
{
    switch(Data){
        case '\f':
            lcd_cmd(0x01);
            __delay_ms(2);
            break;
        case '\n':
            gotoxy_lcd(1,2);
            break;
        default:
            lcd_data(Data);
    }
}

void puts_lcd( char *buffer ){
// permite enviar strings al lcd

    while(*buffer != '\0')    {
        lcd_data(*buffer++);
    }
}
```



# Anexos



Universidad Pública  
de Navarra  
Nafarroako  
Unibertsitate Publikoa

**E.T.S.I.I.T.**  
**INGENIERO  
TECNICO INDUSTRIAL E.**

DEPARTAMENTO:  
**DEPARTAMENTO DE  
INGENIERIA ELECTRICA  
Y ELECTRONICA**

PROYECTO:  
**CONTROL DE UN CONVERTIDOR TRIFÁSICO**

REALIZADO:  
**Soñora Mariño, Sergio**

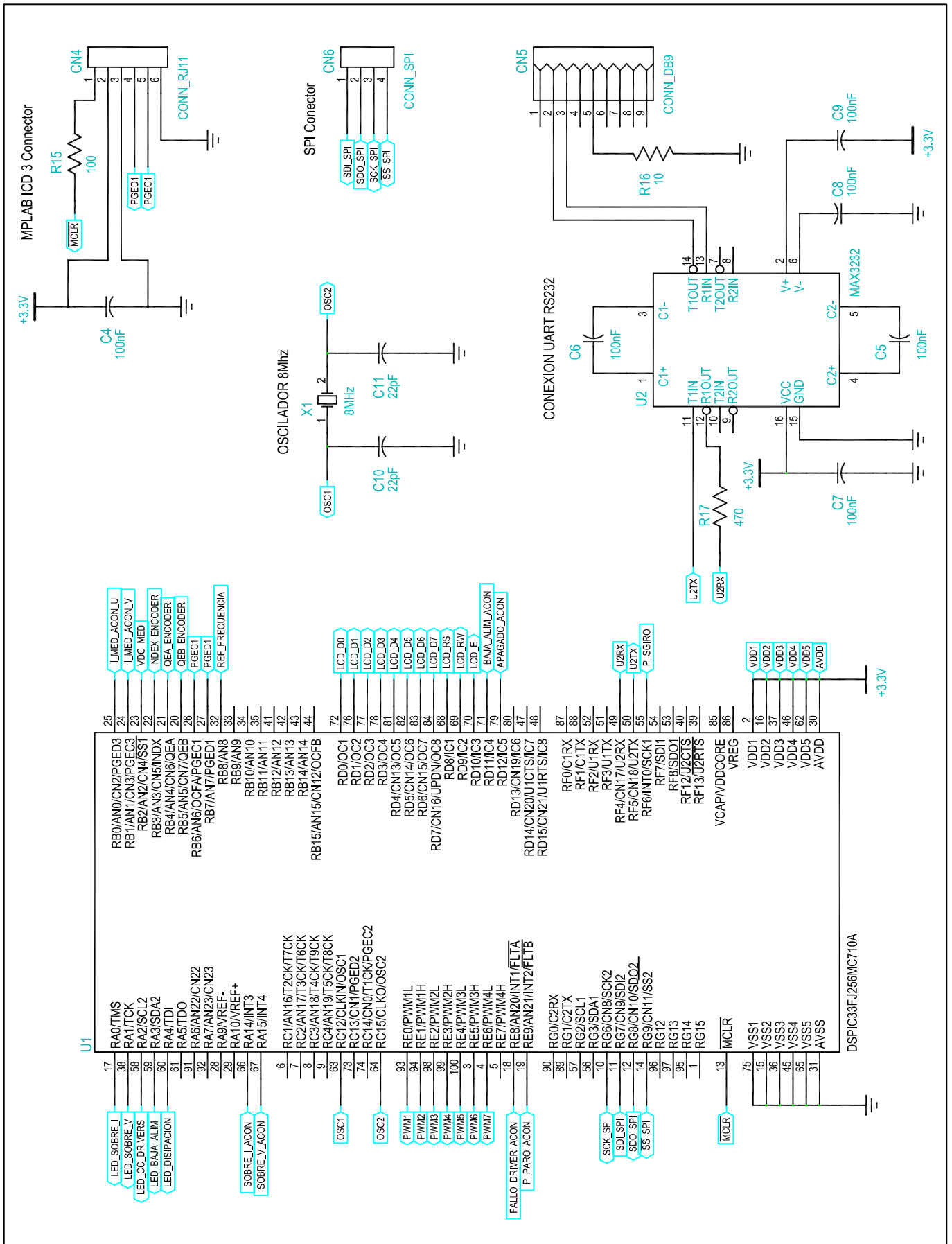


**PANEL DE CONTROL**

FECHA:  
**7/2012**

ESCALA:  
**S/E**

Nº PLANO:  
**01**



Universidad Pública  
de Navarra  
Nafarroako  
Unibertsitate Publikoa

**E.T.S.I.I.T.**  
**INGENIERO  
TECNICO INDUSTRIAL E.**

DEPARTAMENTO:  
**DEPARTAMENTO DE  
INGENIERIA ELECTRICA  
Y ELECTRONICA**

PROYECTO:  
**CONTROL DE UN CONVERTIDOR TRIFÁSICO**

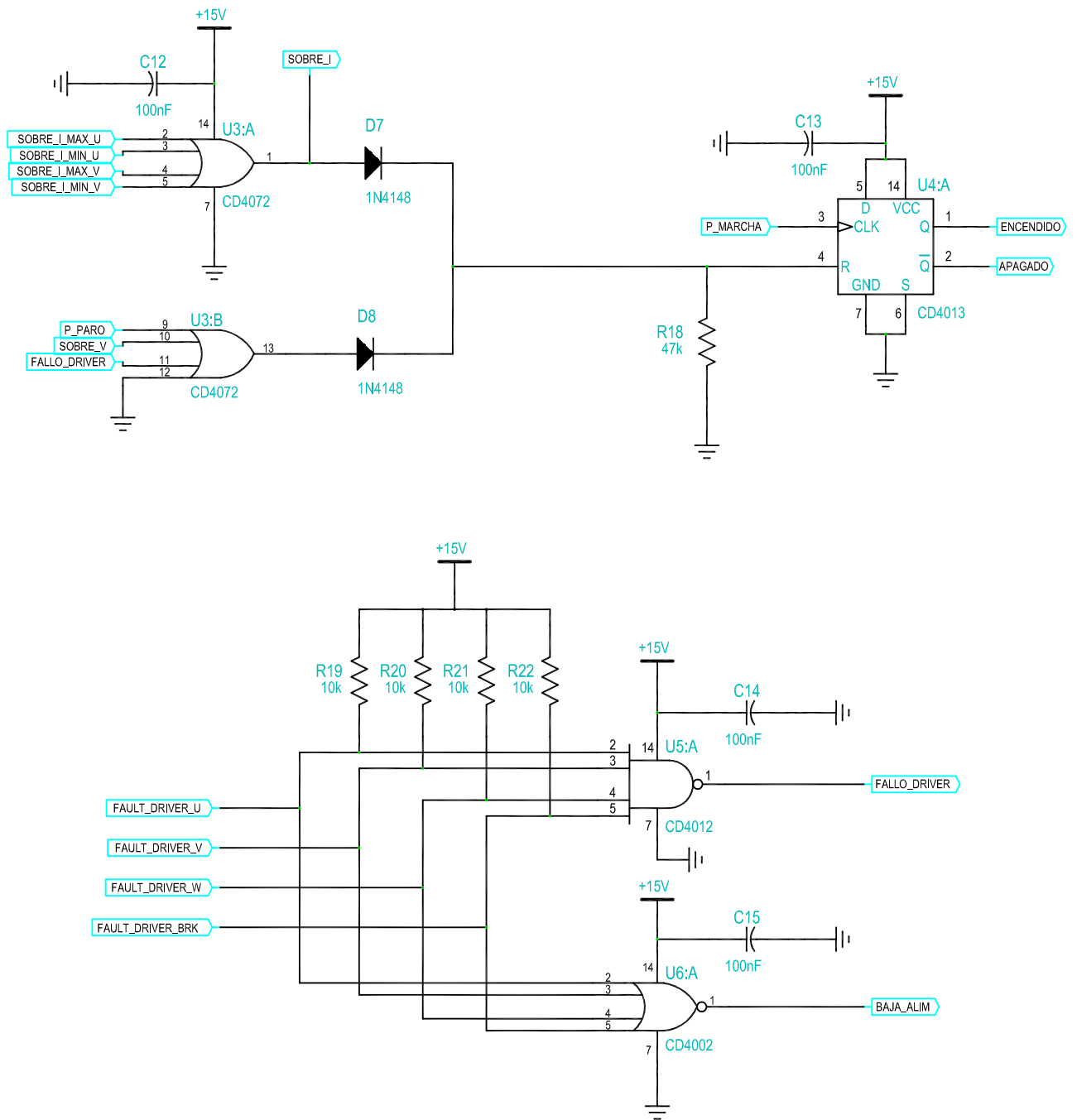
REALIZADO:  
**Soñora Mariño, Sergio**

PLANO:  
**MICROCONTROLADOR dsPIC**

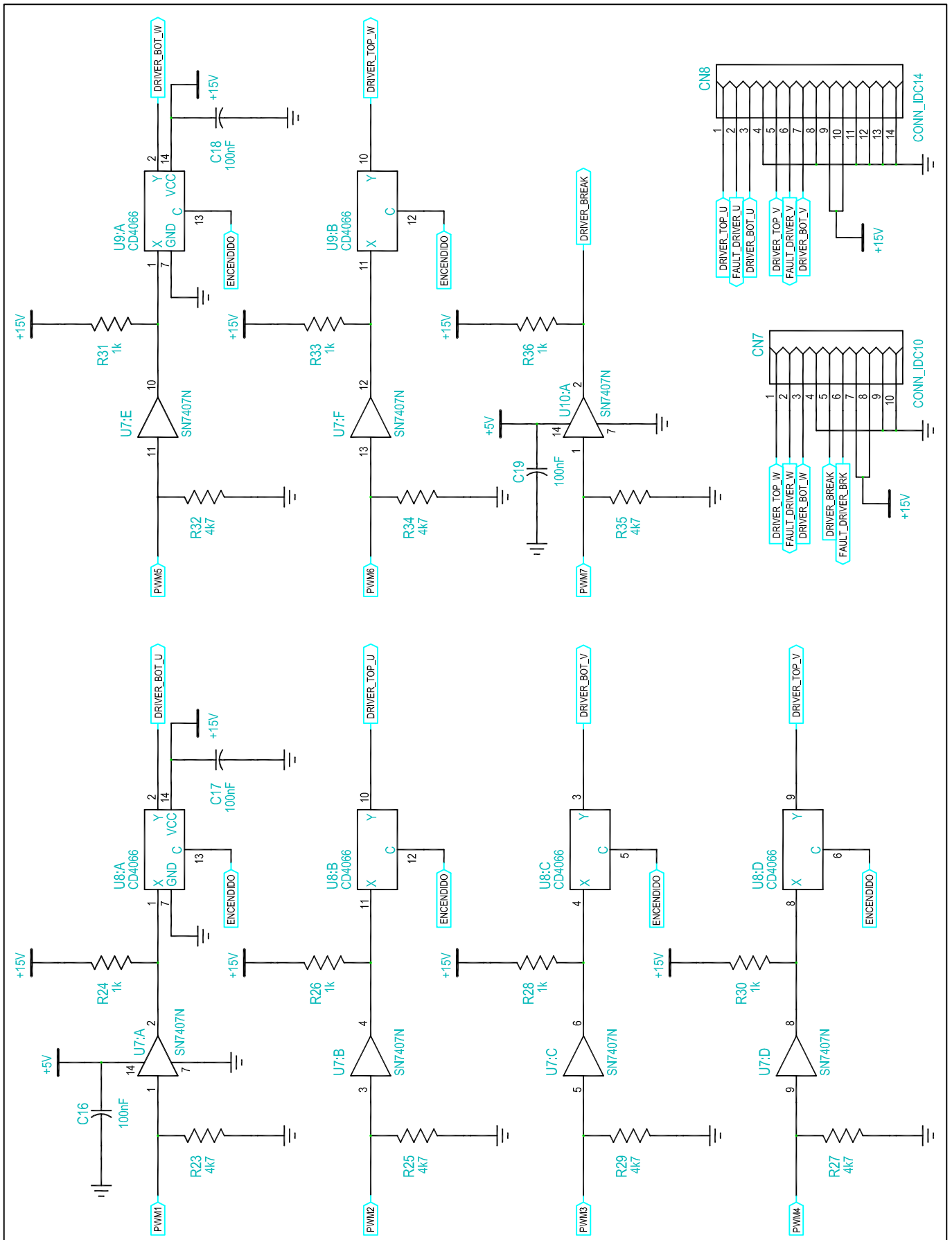
FECHA:  
**7/2012**

ESCALA:  
**S/E**

Nº PLANO:  
**02**



 <p>Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa</p>	<p><b>E.T.S.I.I.T.</b></p>	DEPARTAMENTO:		
	<p><b>INGENIERO TECNICO INDUSTRIAL E.</b></p>	<p><b>DEPARTAMENTO DE INGENIERIA ELECTRICA Y ELECTRONICA</b></p>		
PROYECTO: <p style="text-align: center;"><b>CONTROL DE UN CONVERTIDOR TRIFÁSICO</b></p>		REALIZADO: <p style="text-align: center;"><b>Soñora Mariño, Sergio</b></p>		
PLANO: <p style="text-align: center;"><b>LÓGICA ENCENDIDO/APAGADO</b></p>		FECHA: <p style="text-align: center;"><b>7/2012</b></p>	ESCALA: <p style="text-align: center;"><b>S/E</b></p>	Nº PLANO: <p style="text-align: center;"><b>03</b></p>



Universidad Pública  
de Navarra  
Nafarroako  
Unibertsitate Publikoa

**E.T.S.I.I.T.**  
**INGENIERO  
TECNICO INDUSTRIAL E.**

DEPARTAMENTO:  
**DEPARTAMENTO DE  
INGENIERIA ELECTRICA  
Y ELECTRONICA**

PROYECTO:  
**CONTROL DE UN CONVERTIDOR TRIFÁSICO**

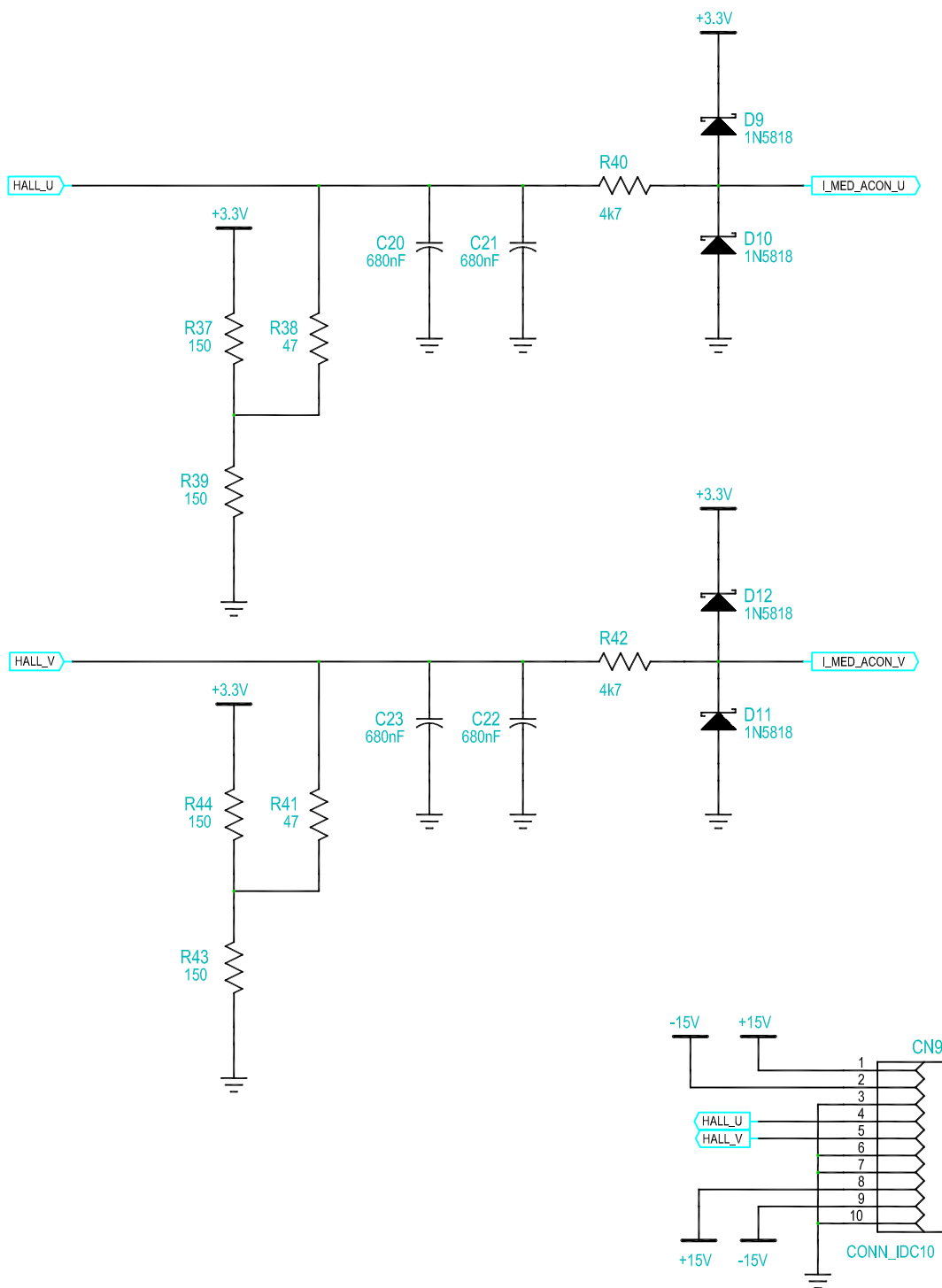
REALIZADO:  
**Soñora Mariño, Sergio**

PLANO:  
**ACONDICIONAMIENTO SEÑALES PWM**

FECHA:  
**7/2012**

ESCALA:  
**S/E**

Nº PLANO:  
**04**



Universidad Pública  
de Navarra  
Nafarroako  
Unibertsitate Publikoa

**E.T.S.I.I.T.**

**INGENIERO  
TECNICO INDUSTRIAL E.**

DEPARTAMENTO:

**DEPARTAMENTO DE  
INGENIERIA ELECTRICA  
Y ELECTRONICA**

PROYECTO:

**CONTROL DE UN CONVERTIDOR TRIFÁSICO**

REALIZADO:

**Soñora Mariño, Sergio**

PLANO:

**ACONDICIONAMIENTO DE CORRIENTE**

FECHA:

**7/2012**

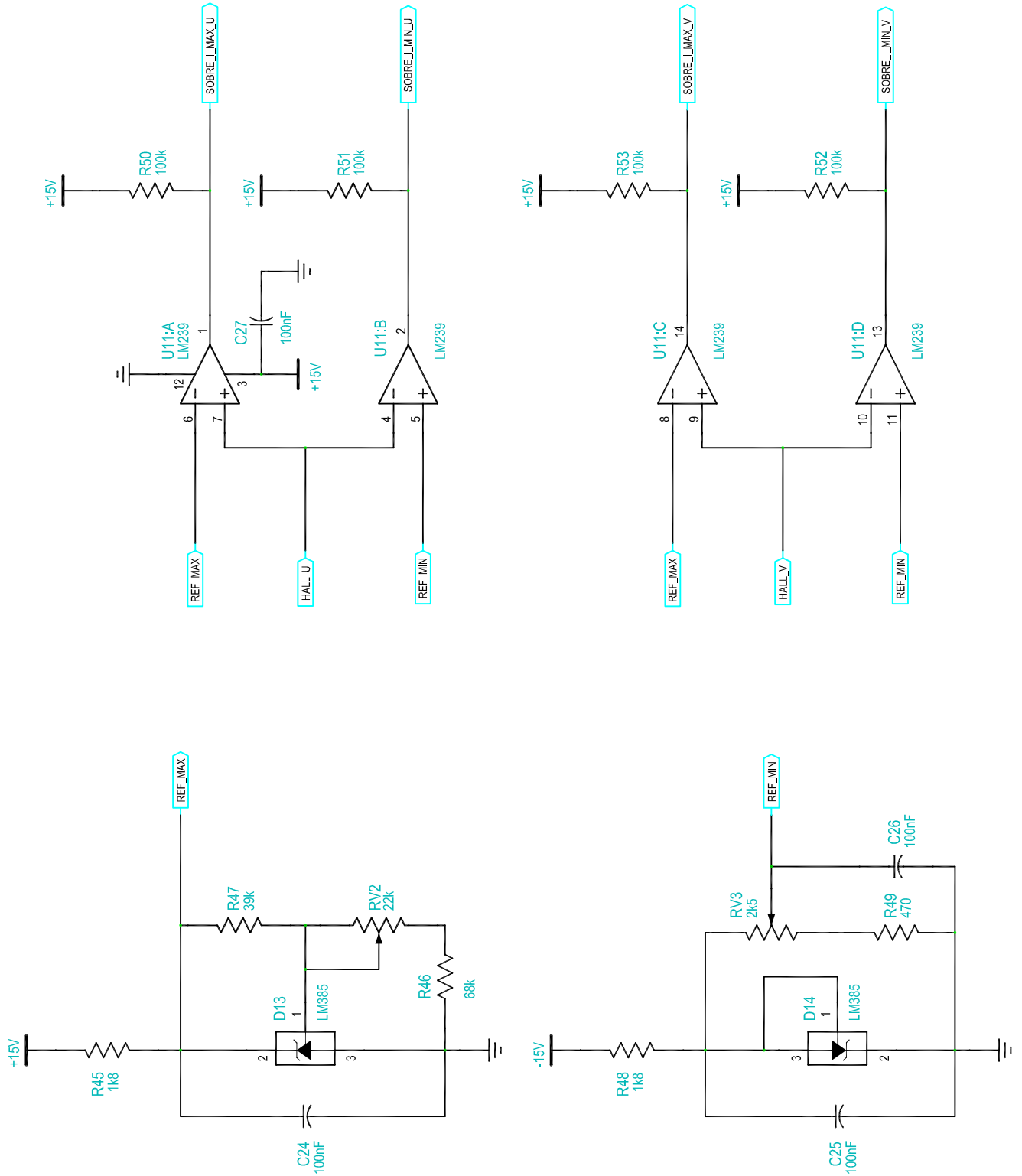
ESCALA:

**S/E**

Nº PLANO:

**05**





Universidad Pública  
de Navarra  
Nafarroako  
Unibertsitate Publikoa

**E.T.S.I.I.T.**  
**INGENIERO  
TECNICO INDUSTRIAL E.**

DEPARTAMENTO:  
**DEPARTAMENTO DE  
INGENIERIA ELECTRICA  
Y ELECTRONICA**

PROYECTO:

**CONTROL DE UN CONVERTIDOR TRIFÁSICO**

REALIZADO:

**Soñora Mariño, Sergio**

PLANO:

**PROTECCIÓN SOBRECORRIENTE**

FECHA:

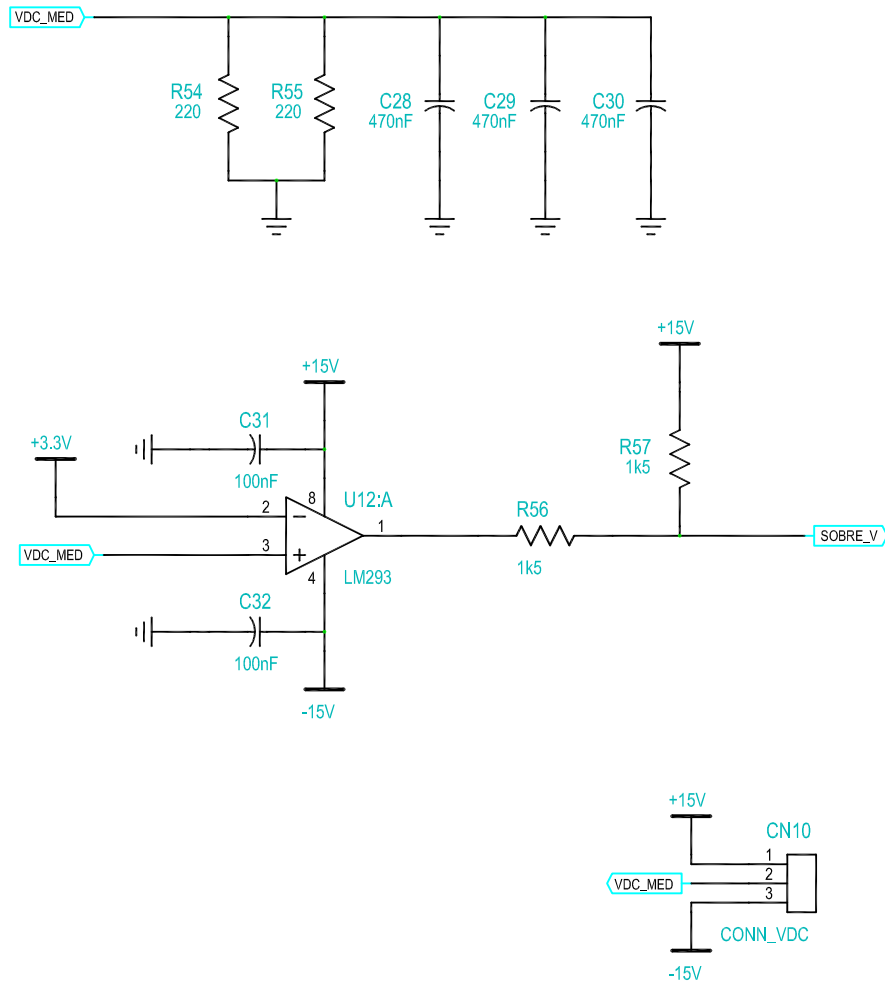
**7/2012**

ESCALA:

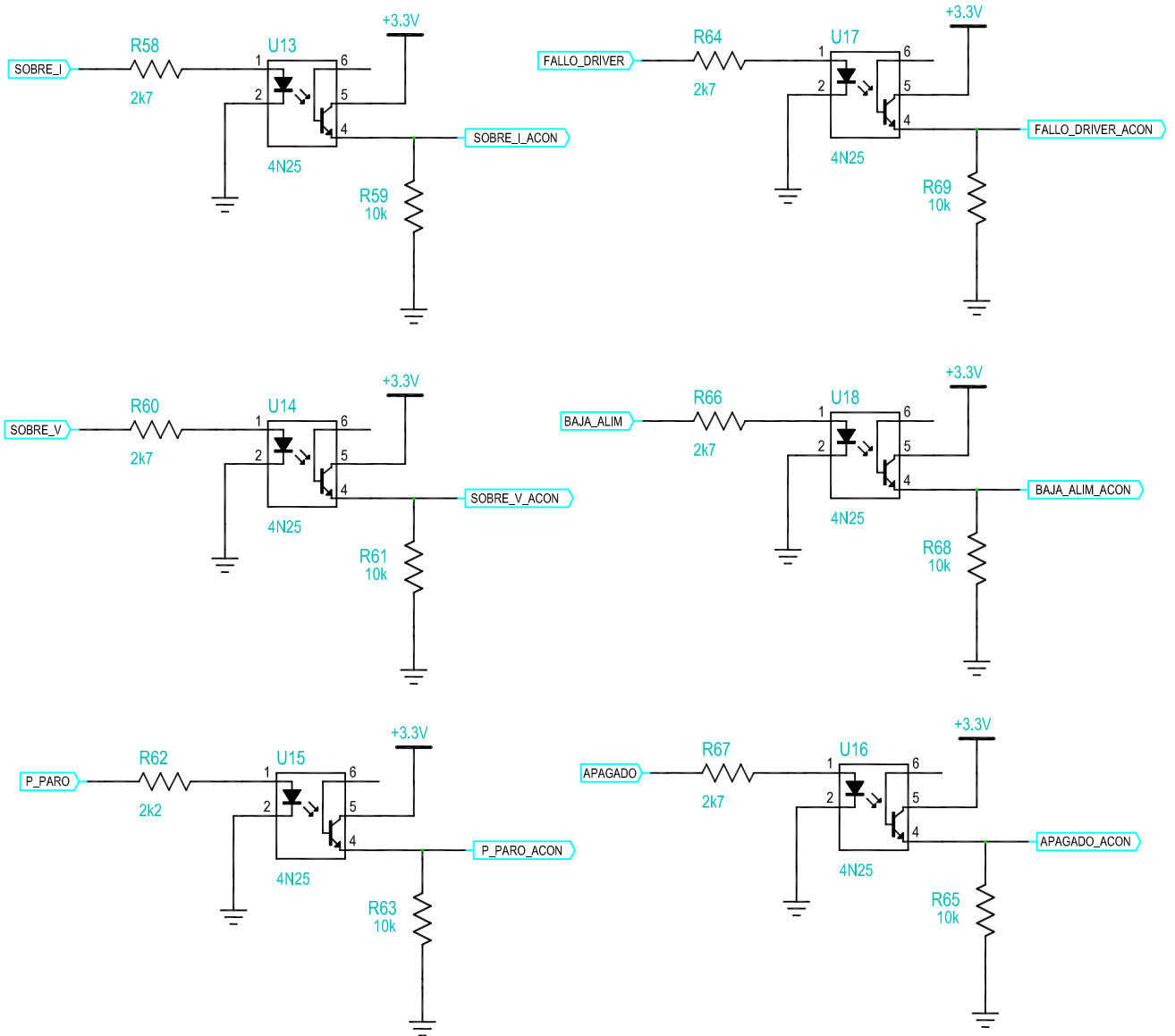
**S/E**

Nº PLANO:

**06**



 <p>Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa</p>	<p><b>E.T.S.I.I.T.</b></p>	DEPARTAMENTO:		
	<p><b>INGENIERO TECNICO INDUSTRIAL E.</b></p>	<p><b>DEPARTAMENTO DE INGENIERIA ELECTRICA Y ELECTRONICA</b></p>		
PROYECTO:	<p><b>CONTROL DE UN CONVERTIDOR TRIFÁSICO</b></p>		REALIZADO:	<p><b>Soñora Mariño, Sergio</b></p>
PLANO:	<p><b>PROTECCIÓN SOBRETENSIÓN</b></p>		FECHA:	<p><b>7/2012</b></p>
			ESCALA:	<p><b>S/E</b></p>
			Nº PLANO:	<p><b>07</b></p>



Universidad Pública  
de Navarra  
Nafarroako  
Unibertsitate Publikoa

**E.T.S.I.I.T.**  
**INGENIERO  
TECNICO INDUSTRIAL E.**

DEPARTAMENTO:

**DEPARTAMENTO DE  
INGENIERIA ELECTRICA  
Y ELECTRONICA**

PROYECTO:

**CONTROL DE UN CONVERTIDOR TRIFÁSICO**

REALIZADO:

**Soñora Mariño, Sergio**

PLANO:

**ACONDICIONAMIENTO SEÑALES DE ESTADO**

FECHA:

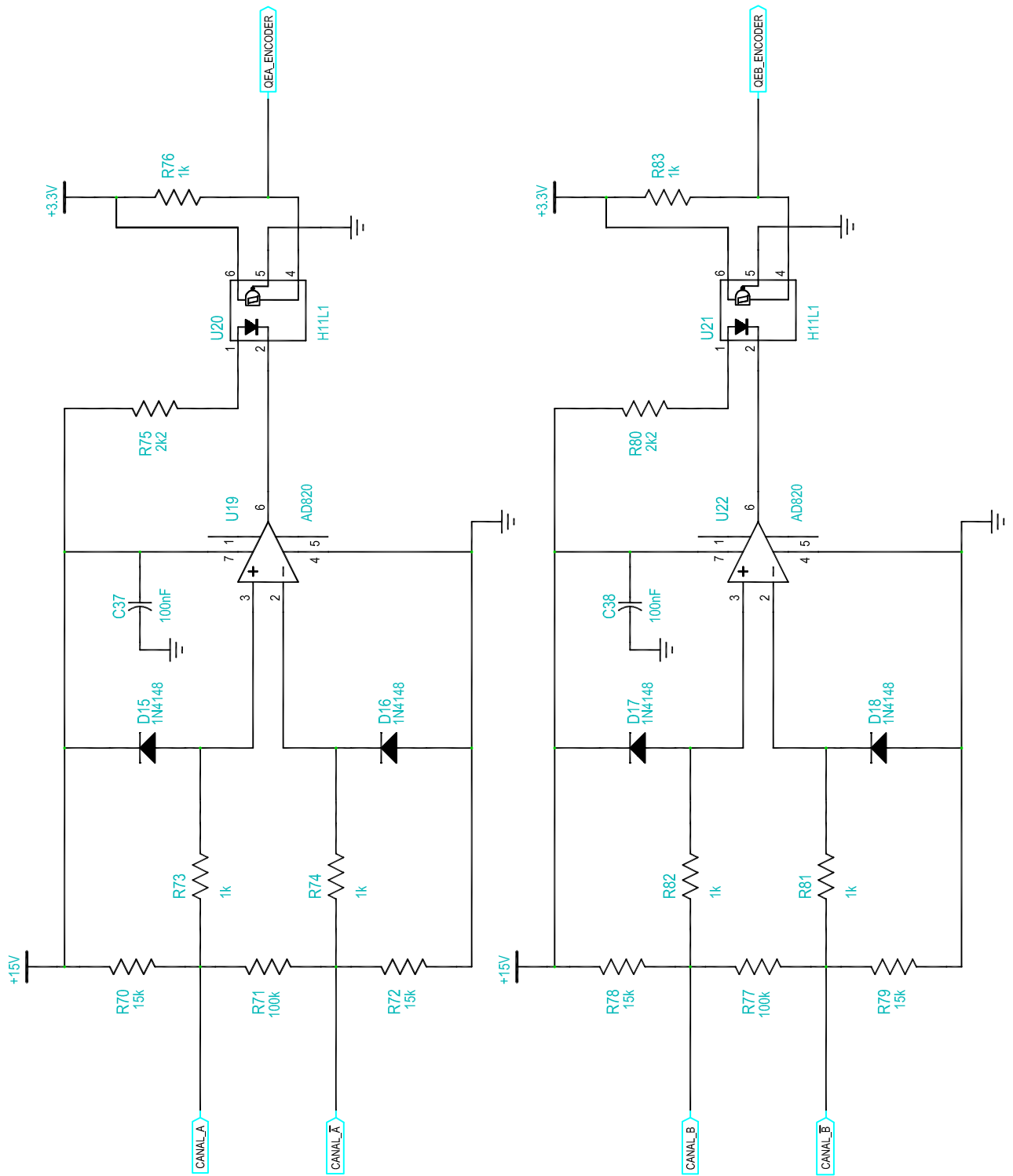
**7/2012**

ESCALA:

**S/E**

Nº PLANO:

**08**



Universidad Pública  
de Navarra  
Nafarroako  
Unibertsitate Publikoa

**E.T.S.I.I.T.**  
**INGENIERO  
TECNICO INDUSTRIAL E.**

DEPARTAMENTO:  
**DEPARTAMENTO DE  
INGENIERIA ELECTRICA  
Y ELECTRONICA**

PROYECTO:

**CONTROL DE UN CONVERTIDOR TRIFÁSICO**

REALIZADO:

**Soñora Mariño, Sergio**

PLANO:

**ACONDICIONAMIENTO ENCODER**

FECHA:

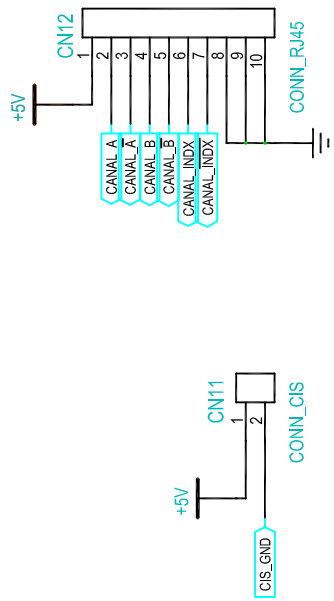
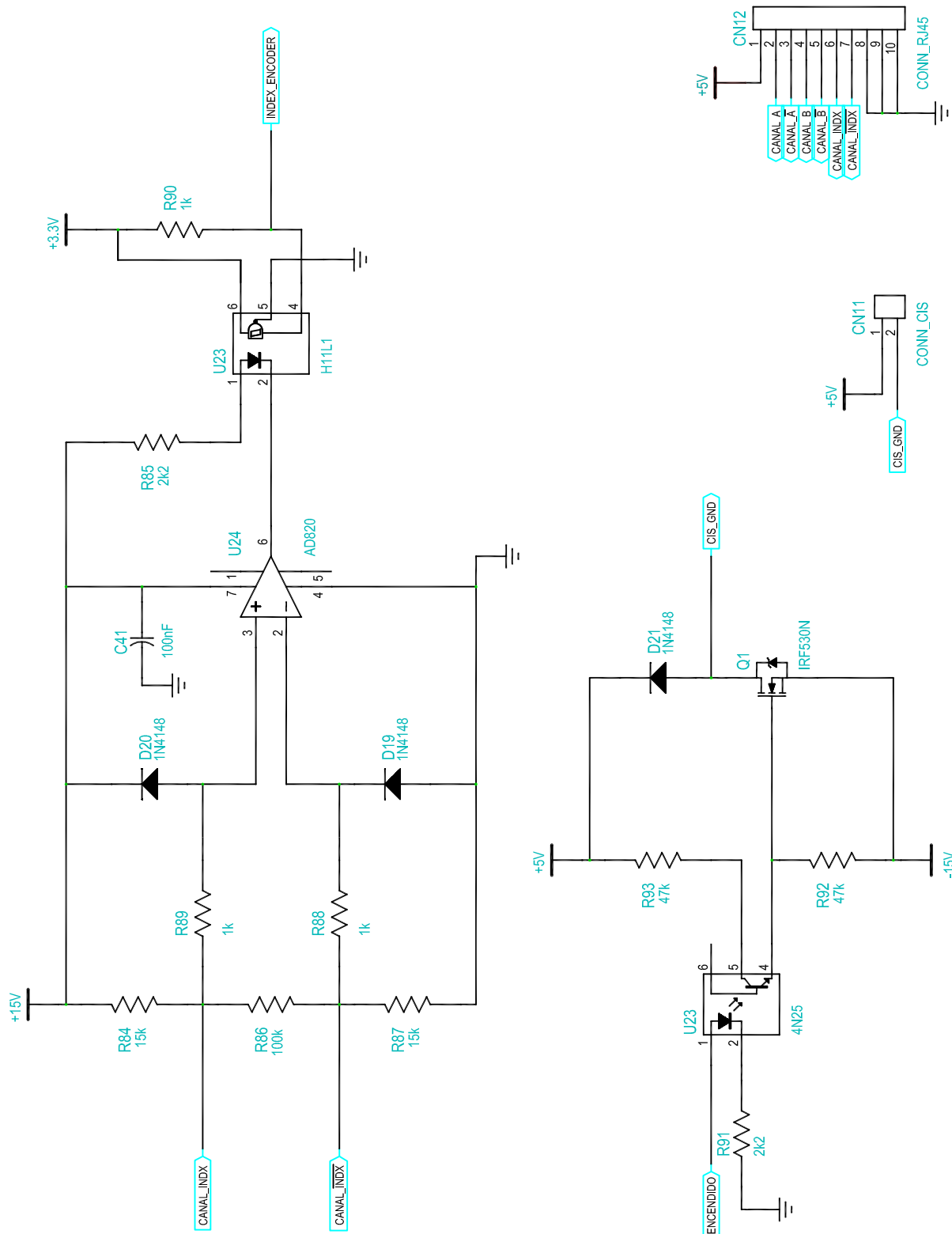
**7/2012**

ESCALA:

**S/E**

Nº PLANO:

**09**



Universidad Pública  
de Navarra  
Nafarroako  
Unibertsitate Publikoa

**E.T.S.I.I.T.**  
**INGENIERO  
TECNICO INDUSTRIAL E.**

DEPARTAMENTO:  
**DEPARTAMENTO DE  
INGENIERIA ELECTRICA  
Y ELECTRONICA**

PROYECTO:  
**CONTROL DE UN CONVERTIDOR TRIFÁSICO**

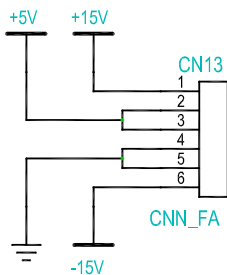
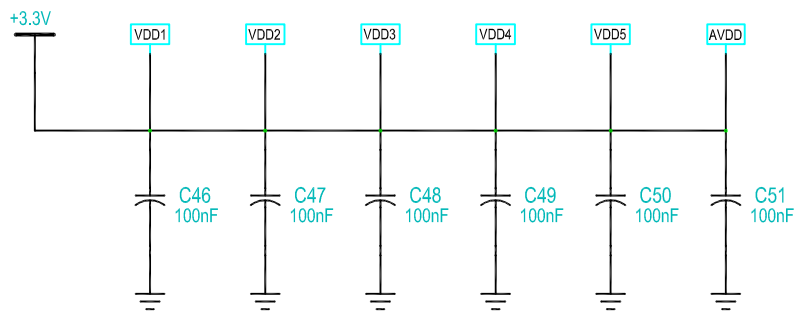
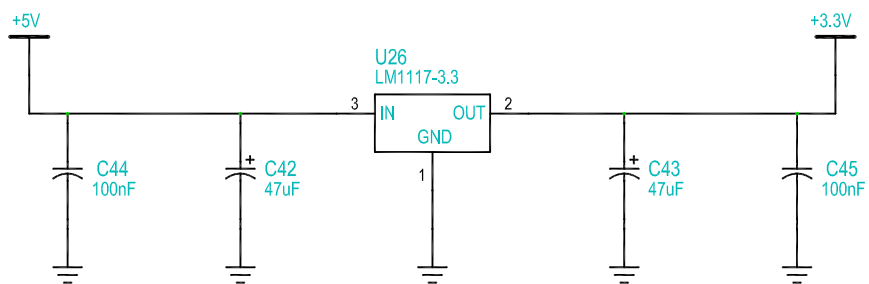
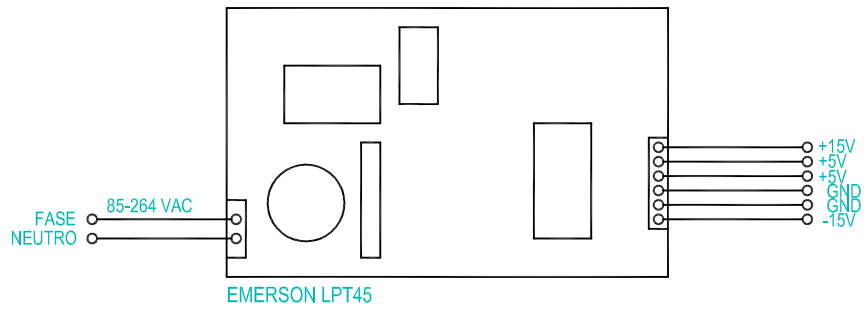
REALIZADO:  
**Soñora Mariño, Sergio**

PLANO:  
**ACONDICIONAMIENTO ENCODER Y CIS**

FECHA:  
**7/2012**

ESCALA:  
**S/E**

Nº PLANO:  
**10**



Universidad Pública  
de Navarra  
Nafarroako  
Unibertsitate Publikoa

**E.T.S.I.I.T.**  
**INGENIERO  
TECNICO INDUSTRIAL E.**

DEPARTAMENTO:  
**DEPARTAMENTO DE  
INGENIERIA ELECTRICA  
Y ELECTRONICA**

PROYECTO:  
**CONTROL DE UN CONVERTIDOR TRIFÁSICO**

REALIZADO:  
**Soñora Mariño, Sergio**

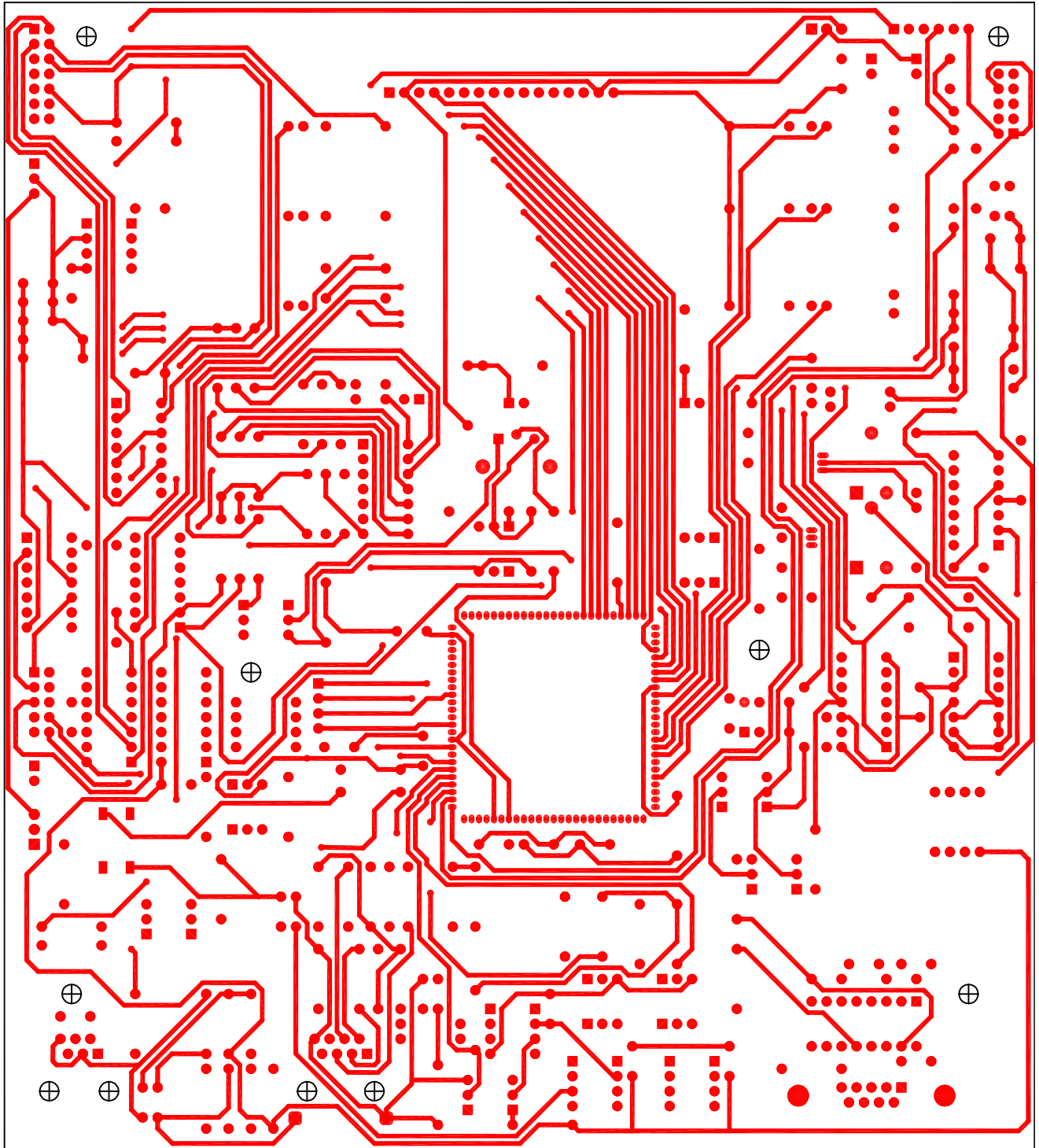


**FUENTE DE ALIMENTACIÓN**

FECHA:  
**7/2012**

ESCALA:  
**S/E**

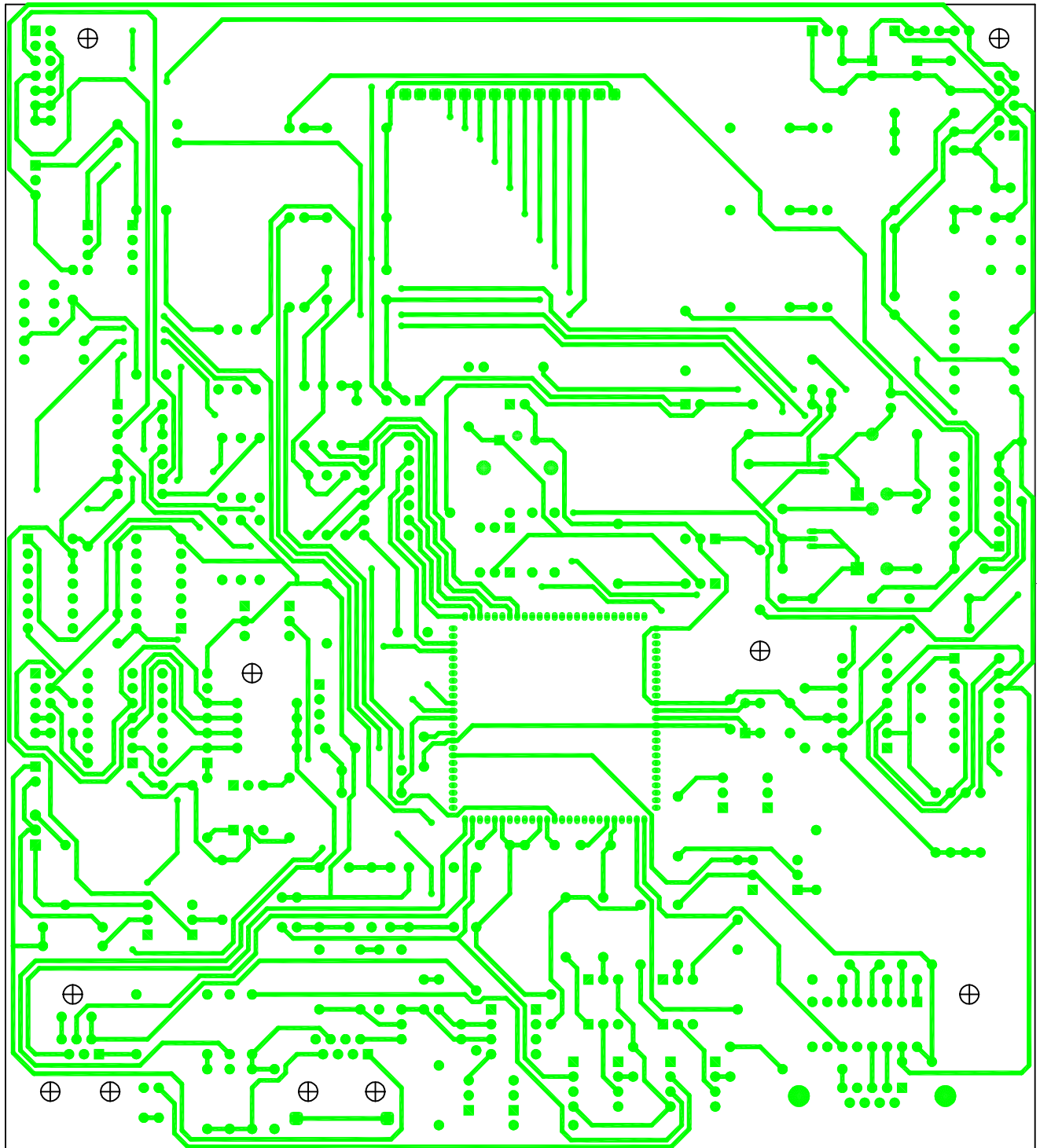
Nº PLANO:  
**11**



Observaciones:


No se ha incluido el plano de tierra en la capa para una visualización mas clara.  
 Para una mejor referencia consultar PCB en el software PCAD-2006.

 Universidad Pública de Navarra <i>Nafarroako</i> <i>Unibertsitate Publikoa</i>	<b>E.T.S.I.I.T.</b>	DEPARTAMENTO: <b>DEPARTAMENTO DE INGENIERIA ELECTRICA Y ELECTRONICA</b>		
	<b>INGENIERO TECNICO INDUSTRIAL E.</b>			
PROYECTO: <b>CONTROL DE UN CONVERTIDOR TRIFÁSICO</b>		REALIZADO: <b>Soñora Mariño, Sergio</b>		
PLANO: <b>PLACA DE CONTROL: PCB CAPA TOP</b>		FECHA: <b>7/2012</b>	ESCALA: <b>1:1</b>	Nº PLANO: <b>12</b>



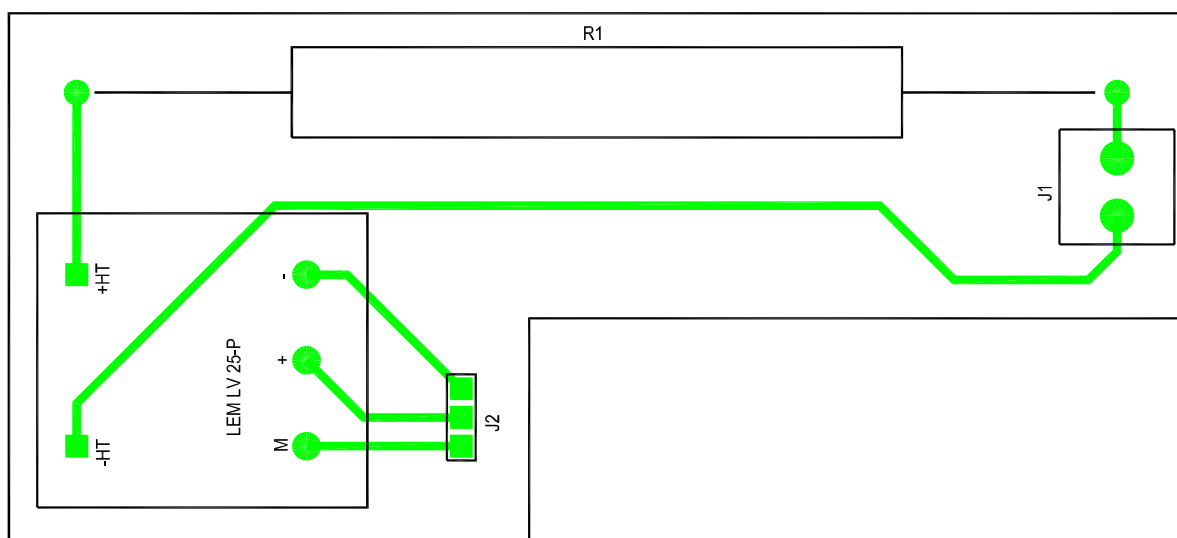
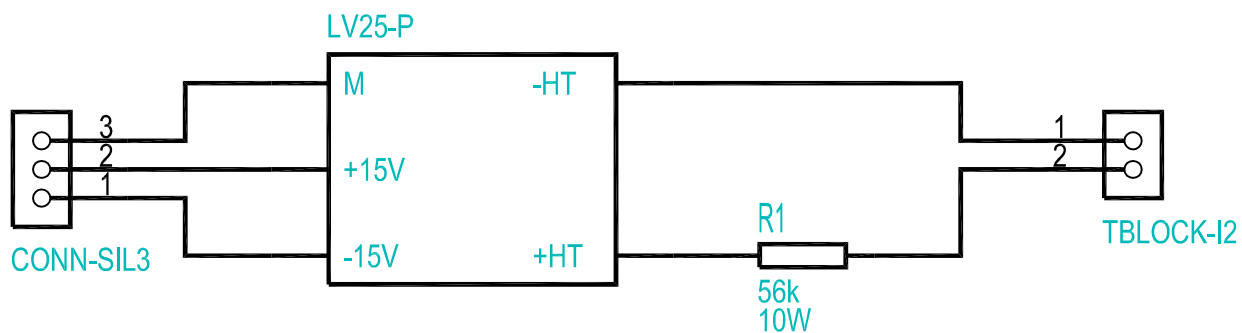
Observaciones:

No se ha incluido el plano de tierra en la capa para una visualización mas clara.  
 Para una mejor referencia consultar PCB en el software PCAD-2006.

 Universidad Pública de Navarra <i>Nafarroako</i> <i>Unibertsitate Publikoa</i>	<b>E.T.S.I.I.T.</b>	DEPARTAMENTO: <b>DEPARTAMENTO DE          INGENIERIA ELECTRICA          Y ELECTRONICA</b>		
	<b>INGENIERO          TECNICO INDUSTRIAL E.</b>	REALIZADO: <b>Soñora Mariño, Sergio</b>		
PROYECTO: <b>CONTROL DE UN CONVERTIDOR TRIFÁSICO</b>		FECHA: <b>7/2012</b>	ESCALA: <b>1:1</b>	Nº PLANO: <b>13</b>
PLANO: <b>PLACA DE CONTROL: PCB CAPA BOTTOM</b>				







Universidad Pública  
de Navarra  
Nafarroako  
Unibertsitate Publikoa

**E.T.S.I.I.T.**

**INGENIERO  
TECNICO INDUSTRIAL E.**

DEPARTAMENTO:

**DEPARTAMENTO DE  
INGENIERIA ELECTRICA  
Y ELECTRONICA**

PROYECTO:

**CONTROL DE UN CONVERTIDOR TRIFÁSICO**

REALIZADO:

**Soñora Mariño, Sergio**

PLANO:

**PLACA DE MEDIDA DE TENSION V<sub>DC</sub>: ESQUEMA Y PCB**

FECHA:

**7/2012**

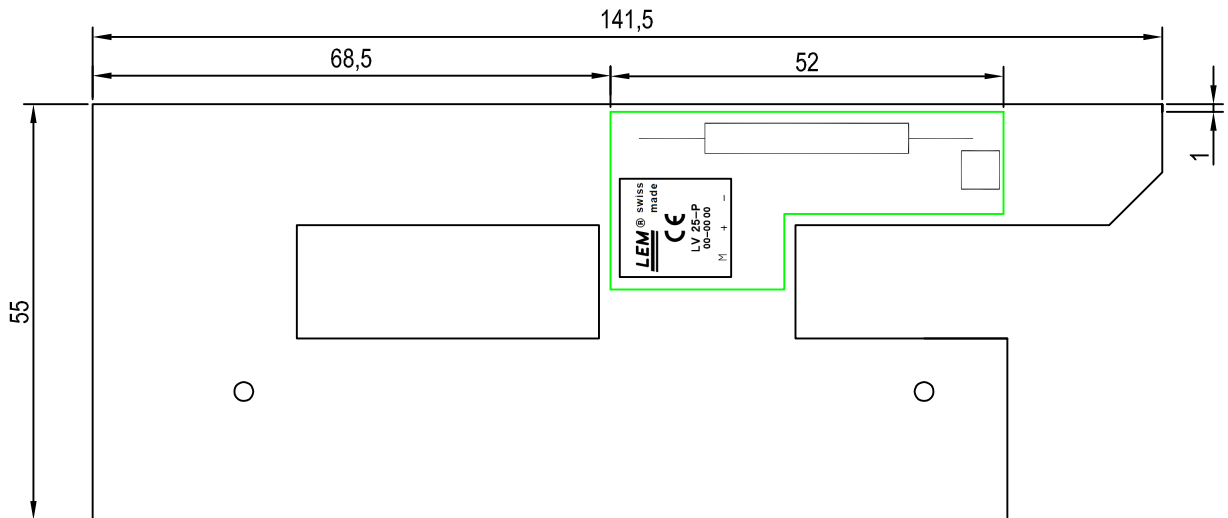
ESCALA:

**S/E**

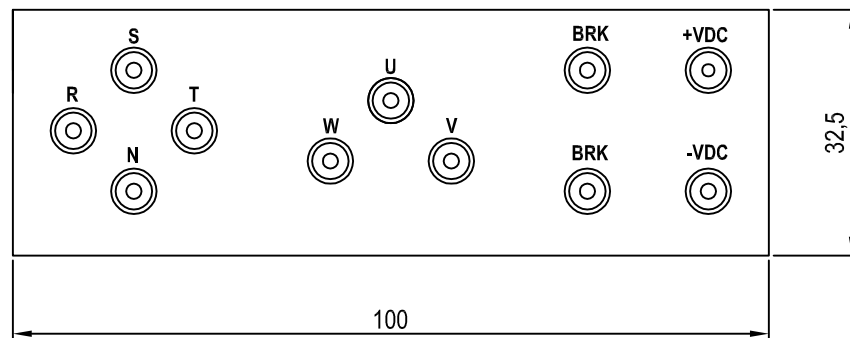
Nº PLANO:

**15**

Detalle de la situación de la placa para la medida de tensión del bus de continua en el módulo de potencia



Detalle de la placa con los bornes de conexiones añadidos al módulo de potencia



Universidad Pública  
de Navarra  
Nafarroako  
Unibertsitate Publikoa

**E.T.S.I.I.T.**

**INGENIERO  
TECNICO INDUSTRIAL E.**

DEPARTAMENTO:

**DEPARTAMENTO DE  
INGENIERIA ELECTRICA  
Y ELECTRONICA**

PROYECTO:

**CONTROL DE UN CONVERTIDOR TRIFÁSICO**

REALIZADO:

**Soñora Mariño, Sergio**

PLANO:

**SITUACIÓN PLACA VDC Y BORNES DE CONEXIONES**

FECHA:

**7/2012**

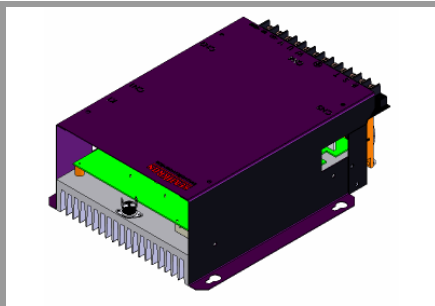
ESCALA:

**1:2**

Nº PLANO:

**16**

# SEMISTACK - IGBT



SEMITOP Stack 1)

## Three-phase inverter

SKS 22F B6U+E1CIF+B6CI 13 V12

SK 30 GB 128

SK 30 GAL 123

SK 95 D 12

P 35/260F

SKHI 20opA

Preliminary Data

## Features

- Compact design
- Hall Effect Current Sensor
- Circuit for soft charge the capacitors
- IGBT Braking chopper
- Vce monitoring

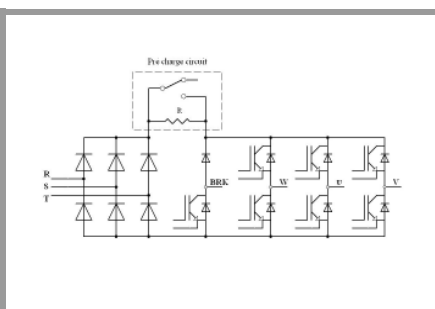
## Typical Applications

- AC Motor Control
- Elevator
- Industrial

1) Photo non- contractual

Circuit	$I_{rms}$	$V_{ac}$ ( $V_{dc}$ )		Types
B6CI	22	380	750	SKS 22F B6U+E1CIF+B6CI 13 V12
		-	-	

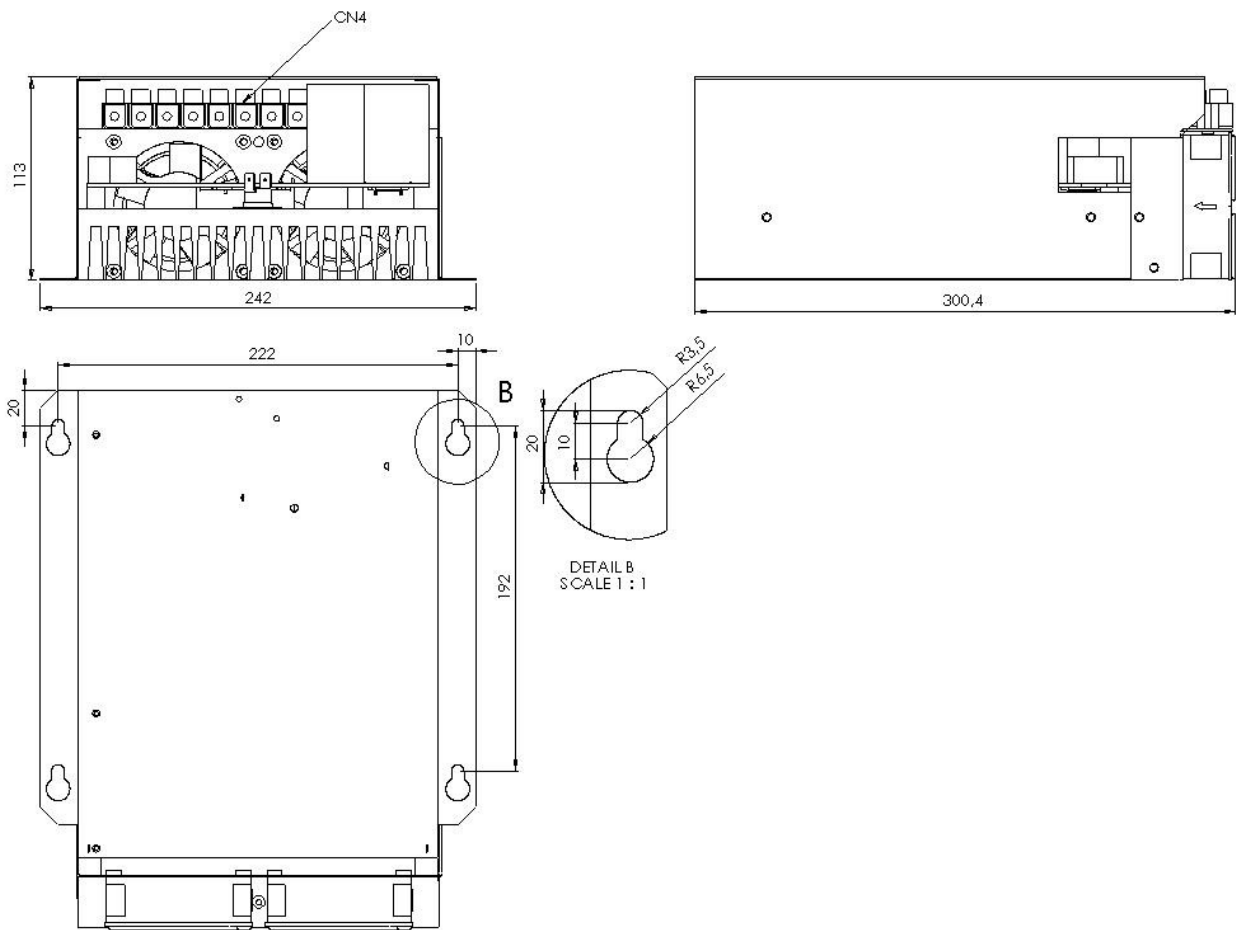
Symbol	Conditions	Values	Units
$I_{rms}$ max	No overload; 10 kHz	22	A
$T_{amb} = 35$ °C	150% overload, 60s every 10min ( $I_{ov}/I_N$ )	27/18	A
	200% overload, 10s every 10min ( $I_{ov}/I_N$ )	30/15	A
$V_{ce}$ max		1200	V
$f_{sw}$ max	Absolute maximum switching frequency	15	kHz
$f_{sw}$ maxCsl	Advise maximum switching frequency	6	kHz
C	Type EPCOS B43303A0687	680/400	$\mu$ F/V
$C_{eqvl}$	Equivalent capacitor bank	1360/800	$\mu$ F/V
$T_{ds\%}$	Discharge time of the capacitor bank	240	s
$V_{DC}$ max	Max DC voltage applied to capacitor bank	750	V
Rectifier		380	$V_{ac}$
$V_{net}$ max	Max network voltage (line side)	-20%/+15%	
$T_{vj}$	Junction temperature for continous operation	-40...+125	°C
$T_{stg}$	without requirement of reforming of capacitors	-20...+40	°C
$T_{amb}$		-20...+55	°C
$V_{isol}$	60Hz/1min	2500	V
w	Aprox. total weight	5.5	Kg
Cooling	Fan, DC power supply	24	V
	Current Consumption (per fan)	0.11	A
	Required air flow (per fan)	42.5	$m^3/h$
Losses	B6CI , Converter at $P_{max}$ , $T_{amb} = 35$ °C	300	W
	Efficiency	97	%
Current sensor	Hall-type LEM LAH 25-NP		
Thermal trip	normally closed	71	°C
Others components	Relay Metaltex J1NAC3		
Options			
Tests	Functional Test Short Circuit Test Visual Inspection		
	-		



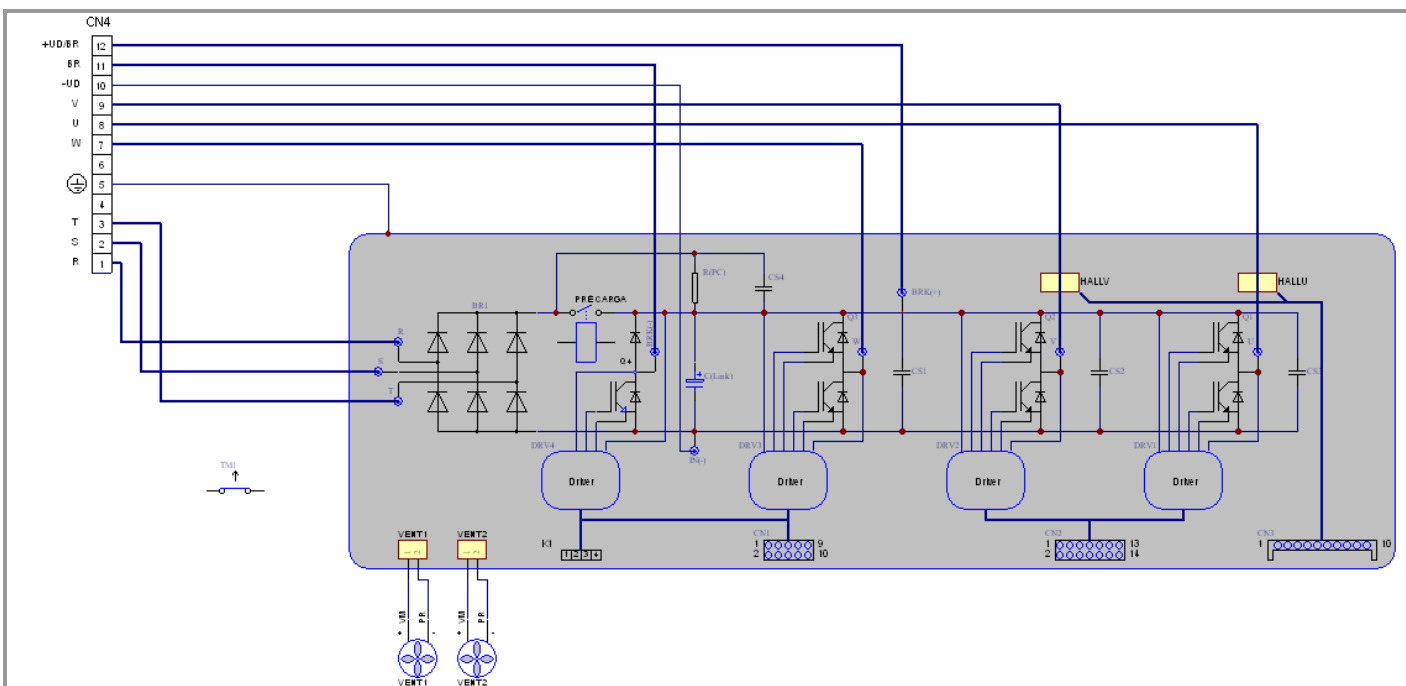
B6U+E1CIF+B6CI

# SEMISTACK - IGBT

Dimensions in mm



Stack design may vary depending upon the version. Please contact SEMIKRON for further details



Electrical Data

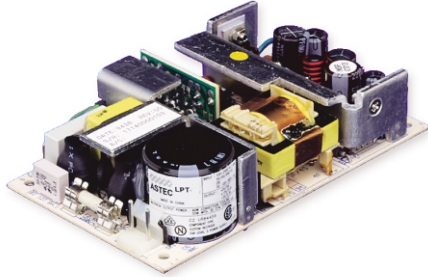
# SEMISTACK - IGBT

Connectors						
Connector Pin	Symbol	Description	Values			Units
			min.	typical	max.	
CN1:1	TOP W	Top phase W input signal		0/15 (CMOS)		V
CN1:2	ER W	Vce phase W error output signal		0/15 (CMOS)		V
CN1:3	BOT W	Bot phase W input signal		0/15 (CMOS)		V
CN1:4	GND	Ground		0		V
CN1:5	Vin(BRK)	Break input signal		0/15 (CMOS)		V
CN1:6	ER BRK	Vce Break error output signal		0/15 (CMOS)		V
CN1:7	+Vs	Supply voltage	14,0		15,6	V
CN1:8	+Vs	Supply voltage	14,0		15,6	V
CN1:9	GND	Ground		0		V
CN1:10	GND	Ground		0		V
CN2:1	TOP U	Top phase U input signal		0/15 (CMOS)		V
CN2:2	ER U	Vce phase U error output signal		0/15 (CMOS)		V
CN2:3	BOT U	Bot phase U input signal		0/15 (CMOS)		V
CN2:4	GND	Ground		0		V
CN2:5	TOP V	Top phase V input signal		0/15 (CMOS)		V
CN2:6	ER V	Vce phase V error output signal		0/15 (CMOS)		V
CN2:7	BOT V	Bot phase V input signal		0/15 (CMOS)		V
CN2:8	GND	Ground		0		V
CN2:9	+Vs	Supply voltage	14,0	15	15,6	V
CN2:10	+Vs	Supply voltage	14,0	15	15,6	V
CN2:11	GND	Ground		0		V
CN2:12	GND	Ground		0		V
CN2:13		NC				
CN2:14		NC				
CN3:1	+15V	Supply Voltage (positive)	14,5	15	15,5	V
CN3:2	-15V	Supply Voltage (negative)	-14,5	-15	-15,5	V
CN3:3	GND	Ground		0		V
CN3:4	HALL U	Output Hall phase U				
CN3:5	HALL V	Output Hall phase V				
CN3:6		NC				
CN3:7	GND	Ground		0		V
CN3:8	+15V	Supply Voltage (positive)	14,5	15	15,5	V
CN3:9	-15V	Supply Voltage (negative)	-14,5	-15	-15,5	V
CN3:10						
CN4:1	R	Input Phase R	176	220	253	V
CN4:2	S	Input Phase S	176	220	253	V
CN4:3	T	Input Phase T	176	220	253	V
CN4:4	(+)	Positive FAN Power Supply		24		V
CN4:5	(-)	Reference FAN Power Supply				
CN4:6		NC				
CN4:7	U	Output U Inverter Phase		220	253	V
CN4:8	V	Output V Inverter Phase		220	253	V
CN4:9	W	Output W Inverter Phase		220	253	V
CN4:10	-UD	DC Link Negative				
CN4:11	BR	Break Resistor Input				
CN4:12	+UD/BR	DC Link Positive Reference and Break Resistor Input				
K1:1		NC				
K1:2	CIS	Charge Input Signal	0	0/18	26,5	V
K1:3	GND	Ground		0		V
K1:4		NC				

This technical information specifies semiconductor devices but promises no characteristics. No warranty or guarantee expressed or implied is made regarding delivery, performance or suitability.

# 40 Watts LPT40 Series

**Total Power:** 40-55 Watts  
**Input Voltage:** 85-264 VAC  
 120-300 VDC  
**# of Outputs:** Triple



## Electrical Specs

### Input

Input range	85-264 VAC 120-300 VDC
Frequency	47-440 Hz
Inrush current	<18 A peak @ 115 VAC; <36 A peak @ 230 VAC, cold start @ 25°C
Input current	1 A max. (RMS) @ 115 VAC
Efficiency	70% typical at full load (60% for LPT41)
EMI filter	FCC Class B conducted CISPR 22 Class B conducted EN55022 Class B conducted VDE 0878 PT3 Class B conducted
Safety ground leakage current	<0.5 mA @ 50/60 Hz, 264 VAC input

### Output

Maximum power	40 W for convection, (LPT41, 25W); 55 W with 30 CFM forced air (LPT41, 41W)
Adjustment range	-5, +10% minimum
Cross regulation	±2% on output 1; ±5% on outputs 2, 3
Hold-up time	20 ms @ 40 W load, 115 VAC nominal line
Overload protection	Short circuit protection on all outputs. Case overload protected @ 110-145% above peak rating
Overvoltage protection	5.7 to 6.7 VDC on main output LPT41: 3.6 to 4.6 VDC
Remote sense	Compensates for 0.5 V lead drop min. Will operate without remote sense connected. Reverse connection protected.

## Special Features

- Universal input
- 3" x 5" footprint
- Remote sense on main output
- Built-in EMI filter
- Low output ripple
- Adjustable main output
- Overvoltage protection
- Overload protection
- 110 KHz switching frequency
- LPX40 enclosure kit available

## Environmental

Operating temperature: 0° to 50°C ambient;  
derate each output at 2.5% per degree from 50° to 70°C

Electromagnetic susceptibility: designed to meet IEC 801, -2, -3, -4, -5, -6, Level 3

Humidity: Operating; non-condensing  
5% to 95%

Vibration: Three orthogonal axes, sweep at 1 oct/min, 5 min. dwell at four major resonances 0.75 G peak 5 Hz to 500 Hz, operational

Storage temperature: -40° to 85°C

Temperature coefficient: ±.04% per °C

MTBF demonstrated: >550,000 hours at full load and 25°C ambient condition

## Safety

<b>VDE</b>	0805/EN60950 (IEC950)	11774-3336-1241 (LC# 84936)
<b>UL</b>	UL1950	E132002
<b>CSA</b>	CSA 22.2-234 Level 3	LR53982C
<b>NEMKO</b>	EN 60950/EMKO-TUE (74-sec) 203	P94100374
<b>CB</b>	Certificate and report	1118,1122,1123,1124
<b>CE</b>	Mark (LVD)	

rev 12.01.03

AMERICAS

5810 Van Allen Way  
Carlsbad, CA 92008  
Telephone: 760-930-4600  
Facsimile: 760-930-0698

EUROPE

Astec House, Waterfront Business Park  
Merry Hill, Dudley  
West Midlands, DY5 1LX, UK  
Telephone: 44 (1384) 842-211  
Facsimile: 44(1384) 843-355

ASIA

Units 2111-2116, Level 21  
Tower1, Metroplaza  
223, Hing Fong Road  
Fwai Fong, New Territories  
Hong Kong  
Telephone: 852-2437-9662  
Facsimile: 852-2402-4426



## Ordering Information

Model Number	Output Voltage	Minimum Load	Maximum Load with Convection Cooling	Maximum Load with 30 CFM Forced Air	Peak Load1	Regulation2	Ripple P/P (PARD)3
LPT41	+3.3 V	0.5 A	4 A	7 A	7 A	±2%	50 mV
	+5 V	0	1.5 A	2.0 A	2.5 A	±2%	50 mV
	+12 V	0	0.5 A	0.7 A	0.7 A	±5%	120 mV
LPT42	+5 V	0.4 A	4 A	5 A	7 A	±2%	50 mV
	+12 V	0.2 A	2 A	2.5 A	4 A	±5%	120 mV
	-12 V	0 A	0.5 A	0.7 A	1 A	±5%	120 mV
LPT43	+5 V	0.5 A	6 A	8 A	9 A	±2%	50 mV
	+12 V	0 A	0.5 A	0.7 A	1 A	±5%	120 mV
	-12 V	0 A	0.5 A	0.7 A	1 A	±5%	120 mV
LPT44	+5 V	0.4 A	4 A	5 A	7 A	±2%	50 mV
	+12 V	0.2 A	2 A	2.5 A	4 A	±5%	120 mV
	-5 V	0 A	0.5 A	0.7 A	1 A	±5%	50 mV
LPT45	+5 V	0.4 A	4 A	5 A	7 A	±2%	50 mV
	+15 V	0.2 A	2 A	2.5 A	3 A	±5%	150 mV
	-15 V	0 A	0.5 A	0.7 A	1 A	±5%	150 mV
LPT46	+5 V	0.4 A	4 A	5 A	6 A	±2%	50 mV
	+24 V	0.1 A	1.0 A	1.5 A	2 A	±7%	240 mV
	+12 V	0 A	0.5 A	0.7 A	1 A	±5%	120 mV
LPT47	+5 V	0.4 A	4 A	5 A	6 A	±2%	50 mV
	+24 V	0.2 A	1.0 A	1.5 A	2 A	±7%	240 mV
	-12 V	0 A	0.5 A	0.7 A	1 A	±5%	120 mV

1. Peak current lasting <30 seconds with a maximum 10% duty cycle.
2. At 25°C including initial tolerance, line voltage, load currents and output voltages adjusted to factory settings.
3. Peak-to-peak with 20 MHz bandwidth and 10 µF in parallel with a 0.1 µF capacitor at rated line voltage and load ranges.
4. Minimum Loads are required.

## Pin Assignments

Connector	LPT41	LPT42	LPT43	LPT44	LPT45	LPT46	LPT47
SK1-1	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral
SK1-3	Line	Line	Line	Line	Line	Line	Line
SK2-1	5 V	+12 V	+12 V	+12 V	+15 V	+24 V	+24 V
SK2-2	3.3 V	+5 V	+5 V	+5 V	+5 V	+5 V	+5 V
SK2-3	3.3 V	+5 V	+5 V	+5 V	+5 V	+5 V	+5 V
SK2-4	Common	Common	Common	Common	Common	Common	Common
SK2-5	Common	Common	Common	Common	Common	Common	Common
SK2-6	12 V	-12 V	-12 V	-5 V	-15 V	+12 V	-12 V
SK201-1	+Sense	+Sense	+Sense	+Sense	+Sense	+Sense	+Sense
SK201-2	-Sense	-Sense	-Sense	-Sense	-Sense	-Sense	-Sense

### Mating Connectors

AC Input: Molex 09-50-8031 (USA)  
09-91-0300 (UK)  
PINS: 08-58-0111

DC Outputs: Molex 09-50-8061 (USA)  
09-91-0600 (UK)  
PINS: 08-58-0111

Remote Sense: Molex 22-01-2025  
PINS: 08-50-0114

Astec Connector Kit #70-841-006, includes all of the above.

### Notes:

1. Specifications subject to change without notice.
2. All dimensions in inches (mm), tolerance is ±.02".
3. Mounting holes M1 and M2 should be grounded for EMI purposes.
4. Mounting hole M1 is safety ground connection.
5. Specifications are for convection rating at factory settings at 115 VAC input, 25°C unless otherwise stated.
6. Warranty: 1 year
7. Weight: 0.5 lb. / 0.23 kg

