



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**

UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERIAS INDUSTRIALES

Grado en Ingeniería Electrónica Industrial y Automática

**Programación del microcontrolador
TMS320F28379D desde Matlab/Simulink.
Aplicación al control de convertidores
electrónicos de potencia**

Autor:

Izquierdo Buznego, Pablo

Fecha:

17/06/21

Tutor(es):

**Herrero de Lucas, Luis Carlos
Tecnología Electrónica**

RESUMEN

Este trabajo pretende presentar las funciones de control que puede realizar un microcontrolador, como el TMS320F28379D, integrado en la tarjeta de desarrollo LAUNCHXL-F28379D. Para ello se analizarán los distintos bloques, características y componentes electrónicos de la misma, y posteriormente se programará desde Matlab/Simulink procurando así desarrollar un entorno mucho más intuitivo para observar con claridad el procesamiento de las señales. Además, se profundizará en su aplicación para el control de convertidores electrónicos de potencia, centrándonos principalmente en el control de convertidores CC/CA (inversores).

ABSTRACT

This paper aims to present the control functions that a microcontroller such as the TMS320F28379D integrated in the LAUNCHXL-F28379D development card can perform. For this the different blocks, characteristics and electronic components of the same will be analyzed, and later it will be programmed from Matlab / Simulink thus trying develop a much more intuitive environment to clearly observe signal processing. In addition, its application for the control of power electronic converters will be deepened, focusing mainly on the control of DC / AC converters (inverters).

PALABRAS CLAVE

TMS320F28379D, LAUNCHXL-F28379D, Microcontrolador, Programación, Matlab, Simulink, Control de convertidores de Potencia, Inversor, PWM,

KEYWORDS

TMS320F28379D, LAUNCHXL-F28379D, Microcontroller, Programming, Matlab, Simulink, Control of power systems, Inverter, PWM

ÍNDICE

RESUMEN.....	3
PALABRAS CLAVE	3
ÍNDICE.....	5
ÍNDICE DE FIGURAS.....	7
ÍNDICE DE TABLAS.....	11
1. INTRODUCCIÓN Y OBJETIVOS	13
2. DESARROLLO DE TARJETA LAUNCHXL-F28379D Y PRINCIPIOS DE PROGRAMACIÓN EN SIMULINK.	17
2.1 ALIMENTACIÓN	17
2.1.1 Tensiones internas	17
2.1.2 Tensiones externas	23
2.1.3 Planos de PCB analógicos	28
2.2 PROCESADOR	30
2.2.1 Características de la CPU.....	32
2.2.2 Memorias.	33
2.2.3 Módulos externos para mejora del rendimiento.....	40
2.2.4 Comunicación. Modulo IPC.....	44
2.3 MÓDULO PWM.....	49
2.3.1 Control PWM	49
2.3.2 Módulo ePWM en la tarjeta LAUNCHXL-F28379D	50
2.3.3 Aplicación mediante Simulink	59
2.4 CONVERTIDORES ADC Y DAC.....	66
2.4.1 Convertidor ADC	67
2.4.2 Convertidor DAC	70
2.4.3 Aplicación mediante Simulink	71
2.5 OTROS BLOQUES EN SIMULINK	74
2.5.1 Salidas y entradas digitales GPIO	74
2.5.2 Interfaces de comunicación	75
2.5.3 Módulos eQEP y eCAP.....	78
3. APLICACIÓN EN ELECTRÓNICA DE POTENCIA.....	81
3.1 ELECTRÓNICA DE POTENCIA. EL INVERSOR	81
3.1.1 Semiconductores de potencia.....	81
3.1.2 Convertidores de potencia	83
3.1.3 El inversor	90
3.2 CONTROL DEL INVERSOR	95
3.2.1 Control por ancho de pulso. SPWM.....	96
3.2.2 Control vectorial. Modulación SVPWM.....	97

4. DESARROLLO PRÁCTICO DEL TFG	113
4.1 ANALISIS DE OBJETIVOS	113
4.2 APLICACIÓN FINAL. ALGORITMO DE CONTROL DQ	116
4.2.1 Algoritmo de control dq. Objetivos	116
4.2.2 Desarrollo en Simulink.....	118
4.2.3 Resultados teóricos.....	121
4.3 IMPLEMENTACIÓN REAL.....	126
4.3.1 Elementos físicos necesarios	126
4.3.2 Conexión de la tarjeta LAUNCHXL-F28379D con el inversor.....	137
4.3.3 Implementación final en la tarjeta	141
5. CONCLUSIONES Y FUTURO DEL PROYECTO	145
BIBLIOGRAFÍA.....	147

ÍNDICE DE FIGURAS

Figura 1. Bloques funcionales de un microcontrolador	13
Figura 2. Sistema de control con lazo cerrado	15
Figura 3. Aplicaciones para el control en tiempo real de un microcontrolador de la familia C2000	15
Figura 4. Estructura del procesador con sus 100 pin PZP	18
Figura 5. Pines de alimentación del procesador	19
Figura 6. Disposición de pines del chip TPS62080ADSGT	20
Figura 7. Esquema de chip reductor 3.3V-1.2V	21
Figura 8. Filtrado de Vddosc y Vdda	22
Figura 9. Filtrado de Vddio	22
Figura 10. Localización de diferentes procesamientos de tensiones externas en la tarjeta	23
Figura 11. Disposición de salidas del puerto USB	24
Figura 12. Esquema de chip reductor 5V-3.3V	25
Figura 13. Pines de alimentación J16 y J10	25
Figura 14. Esquema eléctrico del convertidor elevador	26
Figura 15. Disposición de pines del chip LMR62421	26
Figura 16. Esquema de convertidor elevador en la tarjeta F283792	27
Figura 17. Divisor resistivo para obtener Vo	27
Figura 18. Layout de Vdd	29
Figura 19. Layout de GND	29
Figura 20. Arquitectura del procesador C28x	31
Figura 21. Arquitectura de la memoria RAM	35
Figura 22. Sistema de acceso a memoria Flash	36
Figura 23. Mapa de memoria de programa desde CCS	36
Figura 24. Mapa de memoria de datos desde CCS	37
Figura 25. Password Match Flow (PMF)	39
Figura 26. Esquema de comunicación con el módulo DMA	40
Figura 27. Direct Memory Access (DMA)	41
Figura 28. Bloque CLA en Simulink	43
Figura 29. Parámetros del bloque CLA	44
Figura 30. Arquitectura del módulo IPC	46
Figura 31. Bloque IPC Transmit	46
Figura 32. Bloque IPC Receive	46
Figura 33. Simulación IPC_CPU1	47
Figura 34. Simulación IPC_CPU2	48
Figura 35. Resultado final de la simulación en IPC_CPU2	48
Figura 36. Ejemplo de generación de señal PWM	50
Figura 37. Esquema de módulo ePWM	51
Figura 38. Diagrama de bloques del módulo ePWM	53
Figura 39. Formas de onda generadas en el submódulo TB	54

Figura 40. Generación de eventos en el submódulo CC desde modo de contador down-count.....	55
Figura 41. Generación de eventos en el submódulo CC desde modo de contador up-down-count.....	55
Figura 42. Ejemplo de implementación de las dos salidas PWM	56
Figura 43. Señales PWM resultantes al pasar por el submódulo PC	58
Figura 44. Bloque ePWM en Simulink	60
Figura 45. Pantalla general de programación del bloque ePWM en Simulink.....	60
Figura 46. Ejemplo de implementación en Simulink del módulo ePWM	64
Figura 47. Señal PWM con referencia fija obtenida a partir de la tarjeta LAUNCHXL-F28379D.....	65
Figura 48. Señal PWM con referencia variable, obtenida a partir de la tarjeta LAUNCHXL-F28379D, cuando el ciclo de servicio se aproxima a 100%.....	65
Figura 49. Señal PWM con referencia variable, obtenida a partir de la tarjeta LAUNCHXL-F28379D, cuando el ciclo de servicio se aproxima a 0%.....	66
Figura 50. Esquema de procesamiento de la señal	67
Figura 51. Diagrama de bloques del módulo ADC	67
Figura 52. Circuito de muestreo y retención	68
Figura 53. Salida analógica del convertidor S/H	68
Figura 54. Diagrama de bloques del módulo DAC	71
Figura 55. Bloque ADC en Simulink	72
Figura 56. Ventana de configuración del bloque ADC en Simulink	73
Figura 57. Bloque DAC en Simulink	73
Figura 58. Bloques de programación de entradas y salidas digitales en Simulink..	74
Figura 59. Bloques de programación de la interfaz de comunicación SPI en Simulink	75
Figura 60. Bloques de programación de la interfaz de comunicación UART en Simulink.....	76
Figura 61. Bloques de programación del bus de transmisión eCAN en Simulink. ...	77
Figura 62. Bloques de programación del módulo I2C en Simulink.	78
Figura 63. Bloque de programación del módulo eCAP en Simulink.....	79
Figura 64. Bloque de programación del módulo eQEP en Simulink.....	80
Figura 65. Estructura de un IGBT	83
Figura 66. Esquema de tipos de convertidores de potencia.....	84
Figura 67. Convertidor reductor	85
Figura 68. Convertidor elevador	85
Figura 69. Convertidor reductor-elevador.....	85
Figura 70. Regulador de CA total	86
Figura 71. Señal de salida del regulador total	86
Figura 72. Regulador de CA diferencial	87
Figura 73. Señal de salida de regulador diferencial.....	87
Figura 74. Topología y tensión de salida de un rectificador no controlado simple trifásico.....	88

Figura 75. Topología y tensión de salida de un rectificador no controlado serie trifásico.....	89
Figura 76. Topología y tensión de salida de un rectificador controlado simple trifásico con ángulo de disparo de 45°	89
Figura 77. Topología y tensión de salida de un rectificador controlado serie trifásico con ángulo de disparo de 90°.....	90
Figura 78. Estructura de un inversor	90
Figura 79. Topología y tensión de salida de inversor monofásico en semipunte...92	
Figura 80. Topología y tensión de salida de inversor monofásico en puente.....	93
Figura 81. Topología y tensión de salida de inversor monofásico en puente.....	93
Figura 82. Diagrama de bode de ganancia de un filtro paso-bajo.....	94
Figura 83. Distribución de tensiones de fase de un sistema trifásico en el plano bidimensional α - β	98
Figura 84. Evolución del vector resultante de un sistema trifásico en tres instantes de tiempo.	99
Figura 85. Relación entre ejes a-b-c y α - β	100
Figura 86. Resultado de la transformada de Clarke.....	101
Figura 87. Relación entre ejes α - β y d-q	101
<i>Figura 88. Estados de conmutación de un inversor trifásico</i>	103
Figura 89. Hexágono Vectorial.....	104
Figura 90. Hexágono vectorial con tensión de referencia V_{ref}	105
Figura 91. Esquema en Simulink con el objetivo de generar un control SVPWM..	108
Figura 92. Referencia de sistema trifásico en plano α - β	109
Figura 93. Valores de salida de pulsos generados por control SVPWM.....	110
Figura 94. Tensión de fase V_a a la salida del inversor comparada con la referencia inicial	111
Figura 95. Tensiones de fase a la salida del filtro comparadas con el sistema trifásico de referencia	112
Figura 96. Esquema de objetivo inicial del proyecto	113
Figura 97. Bloque ePWM	114
Figura 98. Esquema de segunda opción de desarrollo del proyecto.....	114
Figura 99. Diagrama de bloques de solución final para el desarrollo práctico del proyecto.....	115
Figura 100. Algoritmo de control del Inversor	118
Figura 101. Simulación de la etapa de potencia en Simulink	119
Figura 102. Simulación de inversor trifásico en Simulink.....	119
Figura 103. Simulación de transformadas en Simulink	120
Figura 104. Algoritmo dq con control por ancho de pulso en Simulink	120
Figura 105. Algoritmo dq con control vectorial en Simulink	121
Figura 106. Salida de corriente en simulación con control de ancho de pulso y con control vectorial.	122
Figura 107. Corriente y tensión, potencia activa y reactiva, y potencia instantánea al inyectar $I_{act}= 200A$ e $I_{react}= 0 A$	124

Figura 108. Corriente y tensión, potencia activa y reactiva, y potencia aparente al inyectar $I_{act}= 0$ A e $I_{react}= 200$ A.....	124
Figura 109. Corriente y tensión, potencia activa y reactiva, y potencia aparente al inyectar $I_{act}= 200$ A e $I_{react}= 200$ A.....	125
Figura 110. Fuente de alimentación conmutada de alta potencia del fabricante Elektro-automatik.	127
Figura 111. Esquema de generación de tensión a partir de un panel fotovoltaico	127
Figura 112. Simulador de red REGATRON Serie TopCon	128
Figura 113. Transductor de corriente de la familia de fabricantes LEM.....	129
Figura 114. Transductor de tensión de la familia de fabricantes LEM	130
Figura 115. Circuito inversor	131
Figura 116. Circuito sumador no inversor	131
Figura 117. Esquema de acondicionador de señal óptimo.....	132
Figura 118. Acondicionador de señal SIRIUS HV	133
Figura 119. Inversor Freesun HEMK.....	134
Figura 120. Inversor EDPS Power Stage.....	134
Figura 121. Inversor TIDA-010025	135
Figura 122. Filtro LCL	136
Figura 123. Filtro comercial LCL del fabricante Hilkar	136
Figura 124. Localización de conectores importantes sobre la PCB del inversor... ..	138
Figura 125. Módulo TIDA-00199.....	139
Figura 126. Esquema de funcionamiento general del programa en la tarjeta LAUNCHXL- F28379D.....	141
Figura 127. Programa final de control del sistema en Simulink.....	142
Figura 128. Parámetros del bloque ADC	143

ÍNDICE DE TABLAS

Tabla 1. Operaciones matemáticas que se añaden con la unidad TMU.....	33
Tabla 2. Gestión de bloque Global Shared en memoria RAM.....	34
Tabla 3. Registros de comando.....	44
Tabla 4. Tabla de estados del bloque IPC Receive en Simulink.....	47
Tabla 5. Tabla de eventos correspondiente al ejemplo de la Figura 42.....	57
Tabla 6. Posibles valores de salida PWM según registros TZCLT.....	58
<i>Tabla 7. Interpretación de salida digital para resolución de 12 bits.....</i>	<i>69</i>
Tabla 8. Interpretación de salida digital para resolución de 16 bits.....	70
<i>Tabla 9. Tabla descriptiva de tiristores y MOSFET.....</i>	<i>82</i>
Tabla 10. Estados de tensiones de fase de un inversor trifásico.....	104
Tabla 11. Vector y sector de referencia en función de la proyección α y β	107
Tabla 12. Valores de rangos analógicos permitidos en el convertidor ADC de la tarjeta LAUNCHXL-F28379D.....	130
Tabla 13. Características técnicas de los elementos que van a formar el desarrollo práctico del proyecto.....	137
Tabla 14. Relación entre pines macho del módulo controlador TIDA-010025CB y pines hembra J1 y J3 de la tarjeta LAUNCHXL-F28379D.....	140
Tabla 15. Relación entre pines macho del módulo controlador TIDA-010025CB y pines hembra J4 y J6 de la tarjeta LAUNCHXL-F28379D.....	140

1. Introducción y objetivos

Antes de comenzar a exponer realmente el grueso de este trabajo, vamos a responder, en forma de introducción, algunas de las preguntas que cualquiera se puede hacer al tratar un tema como es el de los microcontroladores. Servirá como base para comprender mucho mejor los temas que más adelante se abordarán:

I. ¿Qué es un microcontrolador?

Estrictamente hablando podríamos definirlo como “circuito integrado capaz de procesar unas determinadas instrucciones grabadas en su memoria”[1]. Esta definición se queda muy corta si no profundizamos en la forma en la que se procesan esas instrucciones, hacia quien van dirigidas o sencillamente que utilidad le podemos dar. Para ello abrimos la siguiente pregunta

II. ¿De qué se compone un microcontrolador?

Cuando hablamos de un microcontrolador es necesario no solo conocer la función que realiza, (indicado en la pregunta I) si no como la realiza. Para ello se pueden clasificar los componentes que lo forman en distintos bloques funcionales, que en conjunto realizan la función principal del sistema. Estos bloques son los siguientes:

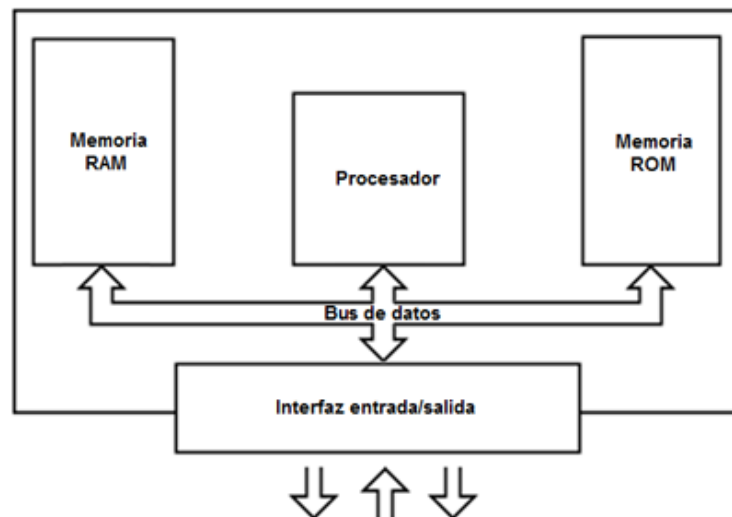


Figura 1. Bloques funcionales de un microcontrolador

- **Procesador:** Bloque principal del microcontrolador, que se encarga de interpretar las instrucciones recibidas mediante operaciones básicas
- **Memoria:** Bloque que se encarga de gestionar y guardar los programas y los datos necesarios para el correcto procesamiento de las instrucciones en el procesador.

Llegados a este punto sería importante recalcar el tipo de arquitectura más común dentro de la familia de los microcontroladores, y se trata de la arquitectura *Harvard*, que, a diferencia de su predecesora en este campo, la arquitectura *Von Neumann*,

dispone de dos memorias independientes, una para instrucciones, y otra para datos con buses independientes para cada una, que facilitan la concurrencia y la velocidad del procesador. Estas memorias pueden ser de diferentes tipos como RAM, ROM, Flash, EPROM, OTP... todas ellas se diferencian entre sí por dos características fundamentales: la volatilidad y la programación de las mismas (lectura y escritura). Normalmente y como luego veremos, estas suelen ir integradas en el propio chip. [3]
[4]

- **Periféricos E/S:** Este bloque es fundamental para la interacción del procesador con el entorno exterior a través de los puertos de entrada y salida, sobre todo para aplicaciones de control, como a las que nos vamos a enfrentar en este caso con la tarjeta F28379D. Sus principales funciones podrían ser las siguientes: Leer señales analógicas o digitales, habilitar o deshabilitar salidas, sacar mediante un puerto de salida una señal analógica o digital. Podríamos clasificarlos según su función en tres tipos:

- I. *Periféricos de entrada/salida:* Para el intercambio de datos directamente con el procesador.
- II. *Periféricos seriales:* Para la transformación de mensajes digitales paralelos en tramas, I2C, CAN, SPI, UART...).
- III. *Periféricos analógicos:* Para la conversión de señales ADC o DAC. [5]

- **Otros bloques:** En este último apartado podemos añadir bloques también importantes como son: El oscilador para generar una señal de reloj (necesaria para el control de las memorias o de los periféricos), la alimentación (con la que se aísla al procesador de tensiones de entrada que puedan ser elevadas) o elementos variados como el botones (como reset o posición inicial), jumps para conectar o desconectar partes circuitos o LEDs para comprobar el estado del microcontrolador [3]

III. *¿En que se basa un sistema de control en tiempo real?*

Como ya se ha ido comentando la principal función de la tarjeta que vamos a tratar es la de actuar como control ante cualquier sistema, eso quiere decir que, para distintas entradas, el sistema debe reaccionar con unas salidas que cambien de acuerdo con una forma deseada.

Dentro de un sistema de control podemos observar dos tipos: El sistema de control con lazo cerrado o realimentación o el sistema de control de lazo abierto. El que nos va a interesar más para las aplicaciones que buscamos va a ser el sistema de lazo cerrado que indicamos en la *Figura 2*, donde más concretamente nos encargaremos de la programación del controlador.

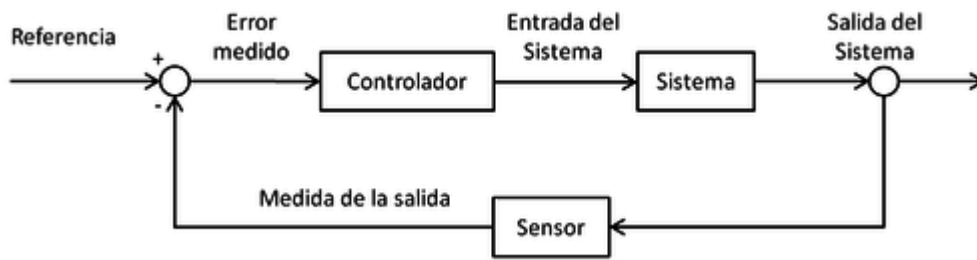


Figura 2. Sistema de control con lazo cerrado

Un sistema de control en tiempo real coincide con la descripción que ya hemos dado, pero se diferencia en la importancia que se le da a minimizar el tiempo entre la medida del sensor, el procesamiento del controlador y la aplicación al sistema. [6]

IV. ¿Qué aplicaciones le podemos dar?

Las aplicaciones que se le pueden dar a un microcontrolador son cada vez mayores y cada vez más aparatos necesitan de uno, ya sea por la continua automatización de los procesos, por el desarrollo de nuevas tecnologías (drones, e-bikes, patinetes...) o por el auge de nuevas fuentes de energía renovables donde es fundamental el control de la electrónica de potencia integrada. El propio fabricante de la tarjeta nos facilita un esquema como el de la Figura 3 donde se indican los diferentes campos de aplicación del microcontrolador [6].

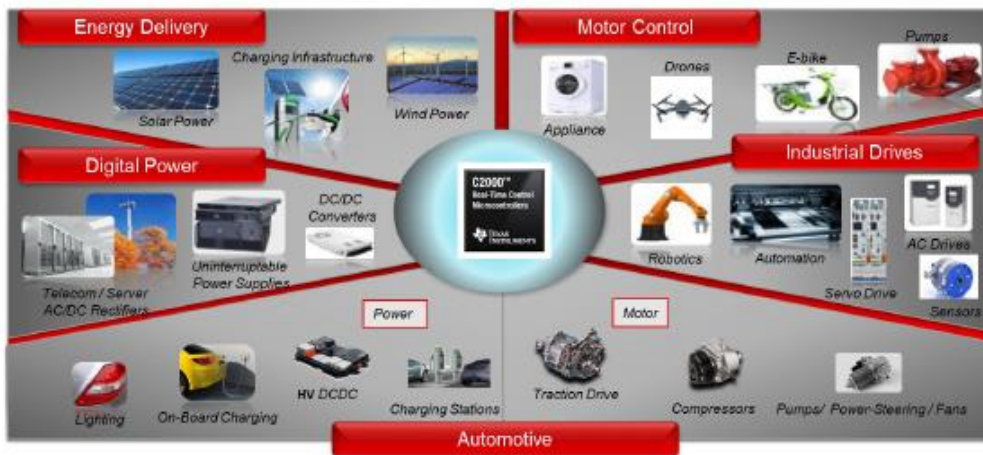


Figura 3. Aplicaciones para el control en tiempo real de un microcontrolador de la familia C2000

En definitiva, el campo de la programación de microcontroladores está en auge, y cada vez es más necesario comprender como funcionan para programarlos ante cualquier aplicación.

Este trabajo servirá para poner el foco en uno de ellos, el TMS320F28379D de la familia C2000 de Texas Instrument (empresa puntera en el desarrollo e implementación de microchips). Fijaremos principalmente cuatro objetivos en el desarrollo del TFG:

- Describir el hardware del microcontrolador.
- Comprender y facilitar su programación a través de Matlab/Simulink.
- Analizar sus aplicaciones en la electrónica de potencia.
- Desarrollar en la práctica alguna aplicación de control.

2. Desarrollo de tarjeta LAUNCHXL-F28379D y principios de programación en Simulink.

Para comenzar con el desarrollo del proyecto vamos a dedicar las siguientes páginas al análisis de los componentes de la tarjeta LAUNCHXL-F28379D. Nos fijaremos principalmente en cuatro bloques funcionales de la placa: La alimentación, el procesador, el módulo PWM, y los módulos ADC y DAC. En el apartado de alimentación observaremos las diferentes tomas de tensión que tiene la tarjeta y como se procesan para dar lugar a diferentes valores de alimentación. En el siguiente punto, analizaremos la función del procesador en la tarjeta y sus características más importantes, también ahondaremos en los diferentes módulos que mejoran su rendimiento y su implementación en Simulink. En los dos siguientes apartados trataremos de analizar dos módulos con gran relevancia para nuestro proyecto: el módulo PWM, y los módulos ADC-DAC. Trataremos de explicar su función en la tarjeta y también su programación desde la interfaz Simulink. Por último, analizaremos el resto de los módulos programables en Simulink de la tarjeta LAUNCHXL-F28379D dando una visión general de cada uno de ellos.

2.1 ALIMENTACIÓN

Todo microcontrolador debe poseer en su estructura alguna clase de circuito electrónico para adaptar cualquier señal de alimentación externa a la alimentación del propio circuito. En este apartado vamos a descifrar cuales son las tensiones externas que se pueden conectar a la tarjeta, y que procesos se llevan a cabo para convertirlas en los distintos valores de potenciales que necesita el procesador.

Vamos a analizar en primer lugar las tensiones internas que va a necesitar el procesador.

2.1.1 Tensiones internas

Para conocer la tensión que se puede aplicar al procesador vamos a observar en el datasheet de la tarjeta cuales de las 100 patillas de las que se compone se corresponden con entradas de alimentación. En la *Figura 4* podemos ver las diferentes entradas al procesador.

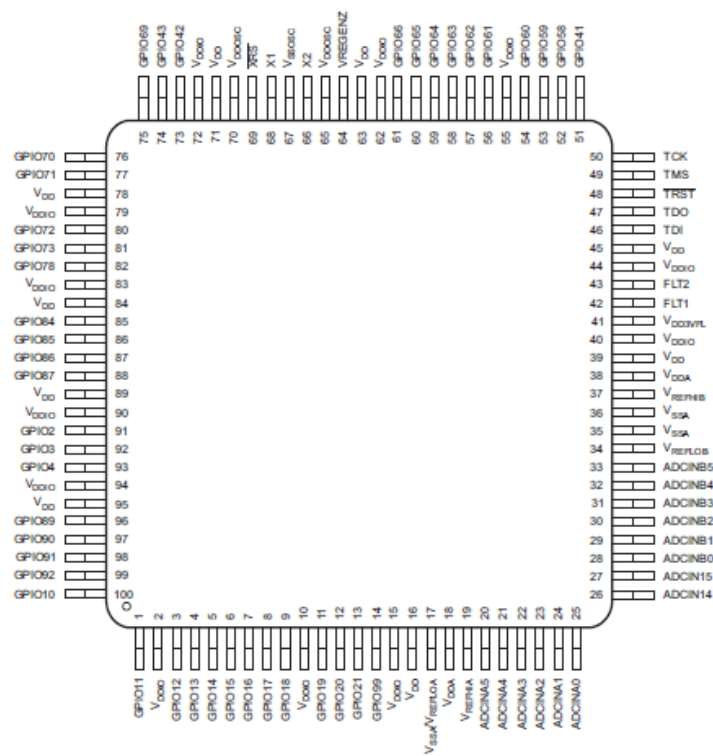


Figura 4. Estructura del procesador con sus 100 pin PZP¹

Con un simple vistazo podemos resaltar, como en la Figura 5, las siguientes diferentes entradas de alimentación, desechando otras como las GPIO, las entradas al convertidor ADC, o las entradas de reloj.

¹ La Figura 4, muestra únicamente 100 de los 176 pines que tiene el procesador que se corresponden con el tipo PZP de diseño de las patillas. Se decide mostrar esta disposición con el único fin de facilitar la lectura de las distintas entradas

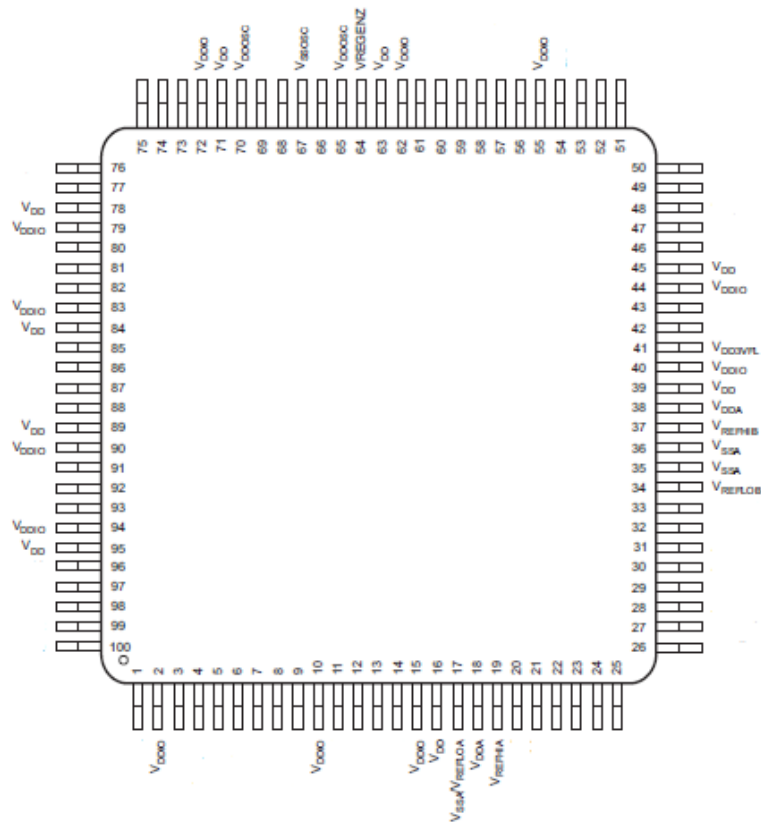


Figura 5. Pines de alimentación del procesador

Podemos clasificar las entradas de tensión en entradas de alimentación y en entradas para referencias del convertidor ADC. Para cada entrada el fabricante recomienda añadir un condensador para el desacoplamiento, en función del uso del pin su valor será diferente. Las entradas de alimentación son las siguientes:

- V_{DDIO}: Pines de alimentación para entradas y salidas de electrónica digital. 3,3V. Condensador acoplado de 0,1uF como mínimo que dependerá de la regulación de la tensión de salida
- V_{DD}: Pines de alimentación para electrónica digital. 1,2V. Condensador acoplado de 20uF.
- V_{DDA}: Pines de alimentación para electrónica analógica. 3,3V. Condensador acoplado de mínimo 2,2uF.
- V_{SS}: Pin de masa. Soldado al plano de tierra de la tarjeta.
- V_{SSA}: Pin de masa analógica.
- V_{DD3VFL}: Pines de alimentación para memoria flash. 3,3V. Condensador acoplado de 0,1uF.

- Vddosc: Pines de alimentación para el oscilador. 3,3V. Condensador acoplado de 0,1uF.
- Vssosc: Pin de tierra para alimentación del oscilador.

Las entradas de referencias para los convertidores ADC son las siguientes²:

- Vrefloa, Vreflob: Referencia inferior para los convertidores ADCA y ADCB que se corresponderá con la salida de tierra.
- Vrefhia, Vrefhib: Referencia superior para los convertidores ADCA y ADCB. Condensador acoplado de 1uF para resolución de 12 bits, y de 22uF para resolución de 16 bits [7].

Como hemos ido analizando podemos confirmar que la tensión nominal va a ser 3,3V para todas las entradas menos para las de carácter digital, Vdd, que admiten un máximo de 1,2V. Así pues, va a ser necesario añadir un **chip reductor de tensión** para obtener el valor de 1,2V.

En la tarjeta y gracias a la hoja de características podemos observar cómo se implanta esta conversión a partir del chip TPS62080ADSGT que vamos a analizar a continuación.

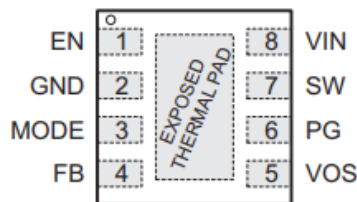


Figura 6. Disposición de pines del chip TPS62080ADSGT

Como podemos observar en la *Figura 6*, el chip dispone de 8 pines:

El pin 1 y el pin 2 son el de habilitación y de toma a masa, el pin 3 habilita el modo ahorro (alta eficiencia con poca corriente, 2 mA), el pin 4 es el que se usa para la realimentación, el pin 5, es el de entrada de la realimentación de la tensión de salida, el pin 6 es el que comprueba que se mantengan los límites de la tensión de salida, el pin 7 es el chip que proporciona la tensión de salida y que debe de estar siempre conectado a la terminal de un inductor y el pin 8 se corresponde con la entrada de tensión

² Existen hasta cuatro convertidores ADC, así que se pueden considerar también las entradas Vrefloc, Vreflod, Vrefhic, Vrefhid

El chip puede aceptar una tensión de entrada desde 2,3V hasta 6V, y una tensión de salida desde 0,5 hasta 4V.

El funcionamiento del convertidor está basado en el control DCS que contiene como principal ventaja la entrada en el modo ahorro al detectar una bajada de corriente sustancial. Sin modo ahorro la topología de control se corresponde con la de un PWM.

La forma de ajustar la tensión de salida es con un divisor de tensión formado por dos resistencias entre el pin 4 (FB) y el 7 (SW), de esta manera la tensión de salida se obtiene de la siguiente forma:

$$V_{out} = V_{fb} * \frac{R2 + R1}{R2} \quad (1)$$

Vfb tiene siempre el valor de salida de 0,45V así que es fácil conocer cuáles pueden ser los valores de R2 y de R1 para que la tensión de salida sea 1,2V, en este caso los valores son 64,9KΩ y 39,2KΩ para R1 y R2 respectivamente. Podemos comprobar como efectivamente la tensión de salida se corresponde con 1,2V.

El circuito en el que se implementa el chip es el que se indica en la *Figura 7*:

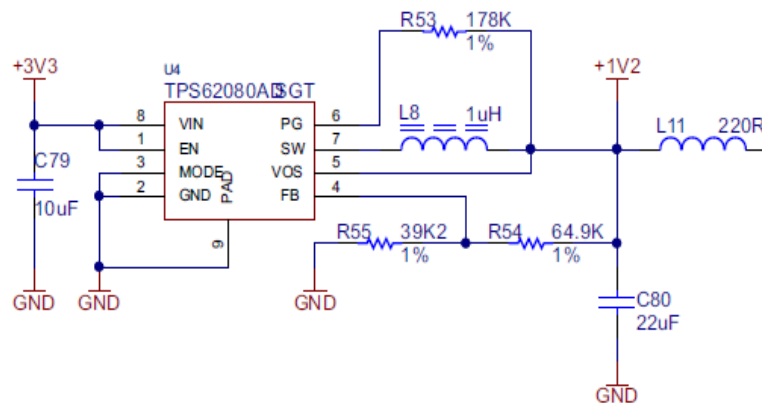


Figura 7. Esquema de chip reductor 3.3V-1.2V

A la alimentación de 3,3 V se conectan los pines de tensión de entrada y de habilitación, a masa se conecta el pin GND y el de modo que deshabilita el modo ahorro, el pin 6 de comprobación se conecta a la tensión de salida mediante una resistencia³. Entre los pines FB (de realimentación) y SW (de tensión de salida) se conecta el divisor de tensión y por último el pin 5 también actúa como realimentación de salida.

³ Resistencia de pull-up que permitirá al pin entrar en modo de alta impedancia cuando el servicio esté deshabilitado

Observamos también como son necesarias tanto dos inductancias para el filtrado de la señal de salida, como dos condensadores para el correcto acoplamiento de la señal

Por último, para acabar con este apartado antes de entrar con las tensiones externas, vamos a mostrar en la *Figura 8* y en la *Figura 9*, cómo se filtran las entradas de tensión de 3,3V antes de llegar al pin correspondiente de alimentación, a través de una bancada de condensadores:

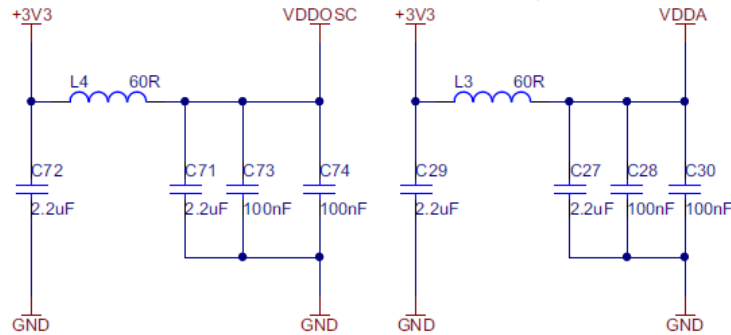


Figura 8. Filtrado de Vddosc y Vdda

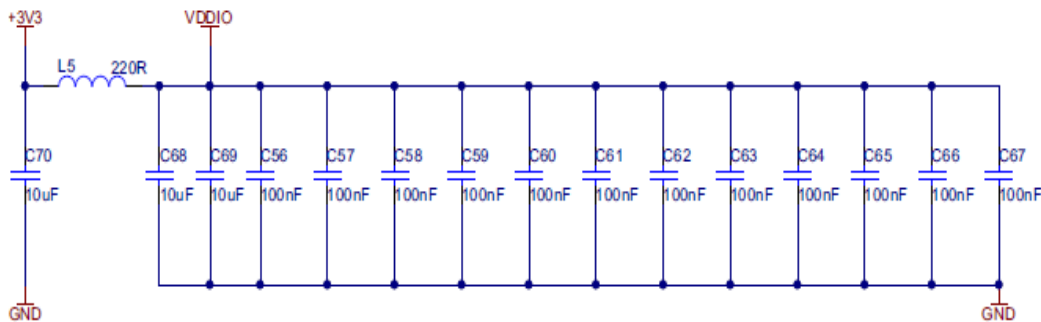


Figura 9. Filtrado de Vddio

Con las dos imágenes podemos concluir que, efectivamente, para la entrada de tensión de 3,3V, Vddio usa condensadores de 10uF (por encima del mínimo de 0,1uF) y Vddosc y Vdda, utilizan condensadores de 2,2uF (por encima del 0,1uF y el 2,2uF fijados como mínimo).

Pendientes nos quedan de analizar la obtención de las tensiones de referencia para los convertidores ADC (Vrefhia, Vrefhib...), pero hemos decidido añadirlo en el apartado de tensiones externas pues para su obtención es necesaria una alimentación externa de 5V [7] [8] [9].

2.1.2 Tensiones externas

Como hemos avanzado al final del apartado anterior no todos los procesos que se llevan a cabo en la tarjeta se bastan con la alimentación de 3,3V, es necesaria para muchos de ellos una alimentación superior, de 5V.

Para obtener cualquiera de las dos siempre es necesaria una referencia externa, un periférico que sea capaz de alimentar la placa desde una fuente de alimentación exterior. En este caso vamos a tener tres posibles formas de procesamiento de la tensión de entrada:

- Procesamiento desde puerto USB.
- Procesamiento desde fuentes externas de 5V o 3,3V.
- Procesamiento interno a través de convertidores elevadores.

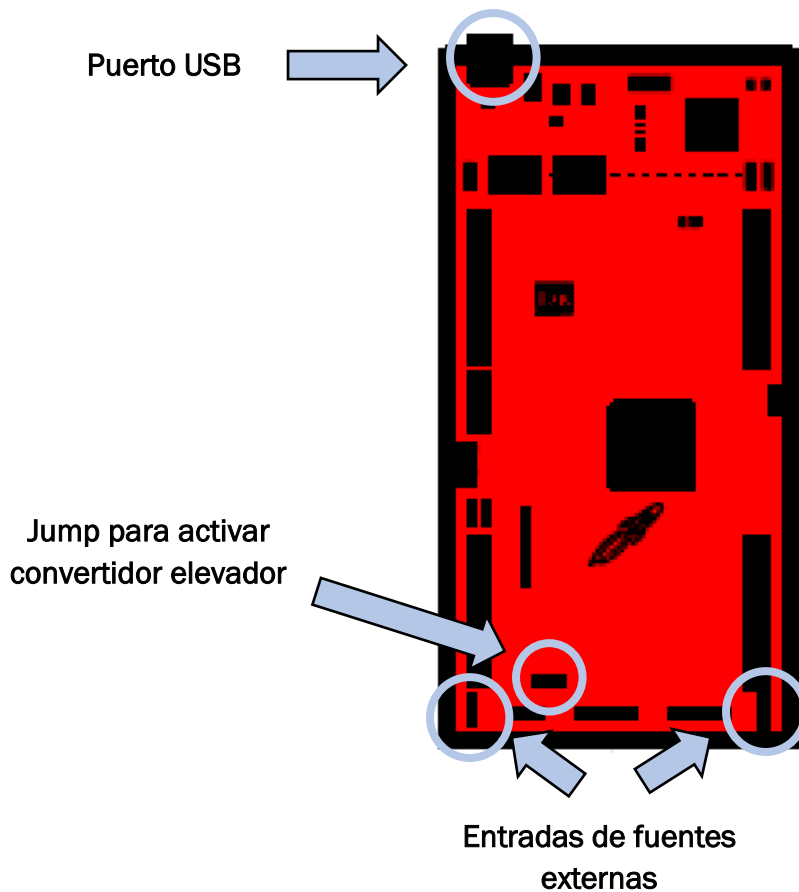


Figura 10. Localización de diferentes procesamientos de tensiones externas en la tarjeta

2.1.2.1 Procesamiento desde USB

Para abordar el procesamiento desde el puerto USB debemos tener en cuenta un punto importante sobre su implantación y no es otro que el aislamiento, la posibilidad de aislar la entrada desde el puerto de datos hacia la placa. Por este afán de conseguir un buen aislamiento, la propia placa dispone de tres Jumper, o interruptores manuales de control de paso de la corriente: JP1, JP2, y JP3. JP1 lleva la alimentación de 3,3V a la placa, JP2 lleva la tierra, y JP3 lleva la alimentación de 5V.

La tensión de referencia que ofrece el puerto USB, (USBVCC) se considera de 5V, como se puede observar en la *Figura 11*, por lo que podemos deducir que va a ser necesario recurrir de nuevo al reductor de tensión que ya hemos comentado en el apartado anterior.

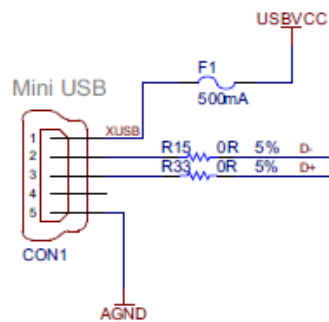


Figura 11. Disposición de salidas del puerto USB

En este caso el chip se denomina TPS62162DSGT, aunque se considera de la misma familia que el ya comentado TPS62080ADSGT, por lo que nos ahorramos la descripción de los pines de entrada y salida, pero no su esquema de funcionamiento ya que en este caso tiene un ligero cambio.

Como se puede apreciar en la *Figura 12*, en este caso no existe divisor de tensión en la realimentación y el pin 5 está conectado a tierra, esta configuración es la que indica que la tensión de salida no se puede regular, la tensión va a ser fija, y su valor está programado en la fabricación del chip [8] [9].

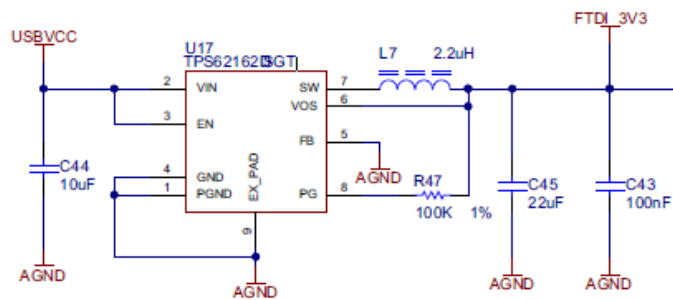


Figura 12. Esquema de chip reductor 5V-3.3V

2.1.2.2 Procesamiento desde fuentes externas

Otra forma de alimentación externa la podemos encontrar a partir de las entradas J10 o J16, de 5V o 3,3V respectivamente. Estas entradas aparecen en la parte inferior de la tarjeta en forma de 3 pines macho, como se señala en la Figura 10. Un pin se corresponde con el de alimentación y los otros dos con masa como se indica en la Figura 13:

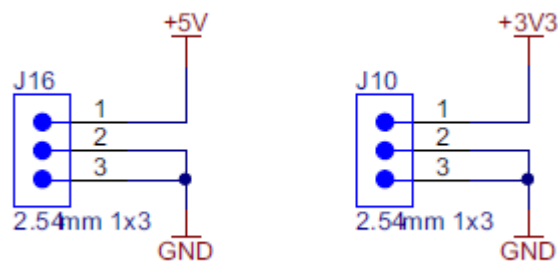


Figura 13. Pines de alimentación J16 y J10

Además de J10 y de J16, también existe la posibilidad de alimentar a la tarjeta desde lo que se denomina el **BoosterPack**, la sucesión de pines que se conectan directamente al microprocesador, y que van a servir en su mayoría para generar las señales de entrada y las de salida. La alimentación se realiza a través de los pines 1 para 3.3V, 20 para masa, 21 para 5V y 22 para masa.

Es muy importante para utilizar esta configuración que la sección de procesamiento de la señal USB este bien aislada del resto de la placa, esto quiere decir, que bajo ningún concepto se pueden conectar los jumpers JP1, JP2 o JP3 ya que se produciría algún fallo eléctrico al detectarse más de una fuente de alimentación en la tarjeta. El USB puede estar conectado, ya que el aislamiento va a servir para transmitir las señales desde el PC, pero en ningún caso la alimentación.

También es lógico que si no se puede conectar ninguna fuente de tensión cuando tenemos J16 o J10 activos, el jumper JP6 tampoco debe estar habilitado, pues es el que enciende el chip elevador que alimenta a la placa con 5V [9].

2.1.2.3 Procesamiento interno

Para terminar con el procesamiento de las señales de tensión externas tratamos de nuevo en este apartado, la implantación de chips reguladores de tensión en la tarjeta. En los epígrafes anteriores ya hemos estudiado dos chips reductores, el TPS62080ADSGT (3V-1,2V) y el TPS62162DSGT (5V-3.3V).

En este caso nos encontramos con la necesidad de dotar al sistema de una alimentación de 5V cuando solo poseemos alimentación inferior: 3,3V. Para ello el fabricante ha decidido incluir un convertidor boost (elevador) en vez de un chip como en los casos en los que se reduce la tensión. En la *Figura 14* se muestra el esquema básico de un convertidor elevador.

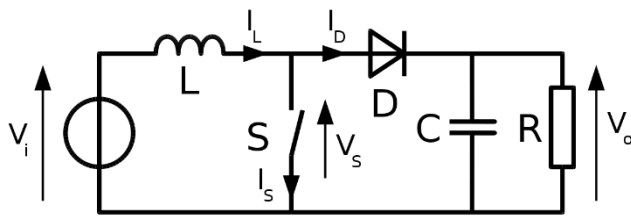


Figura 14. Esquema eléctrico del convertidor elevador

Lo que se trata de controlar en un convertidor de este tipo es el ciclo de servicio D que representa el tiempo en el que el semiconductor (representado como interruptor S) va a estar cerrado (T_{on}) con respecto del periodo total (T).

Para calcular la tensión de salida (V_o) se utilizará la siguiente ecuación, que relaciona V_o con D y la tensión de entrada V_i [10]:

$$V_o = V_i * \frac{1}{1-D} \quad (2)$$

A continuación, vamos a comentar la implantación del convertidor elevador en la tarjeta F28379D. Para el control de los pulsos se utiliza el chip LMR62421 que tiene la disposición de patillas que se indica en la *Figura 15*:

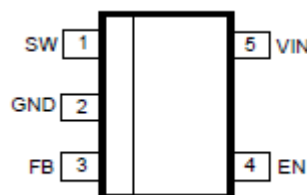


Figura 15. Disposición de pines del chip LMR62421

La función de cada pin se puede deducir fácilmente. El pin 5, es por donde llega el voltaje a convertir, el pin 2 irá conectado a masa, el pin 4 se corresponde con el de habilitación, el pin 3 es el encargado de sacar la tensión de salida y el pin 1 es también un pin de salida de tensión conectado directamente al transistor NMOS, que debe ir conectado al inductor y al diodo.

Podemos observar la implementación del convertidor en la tarjeta en la *Figura 16*.

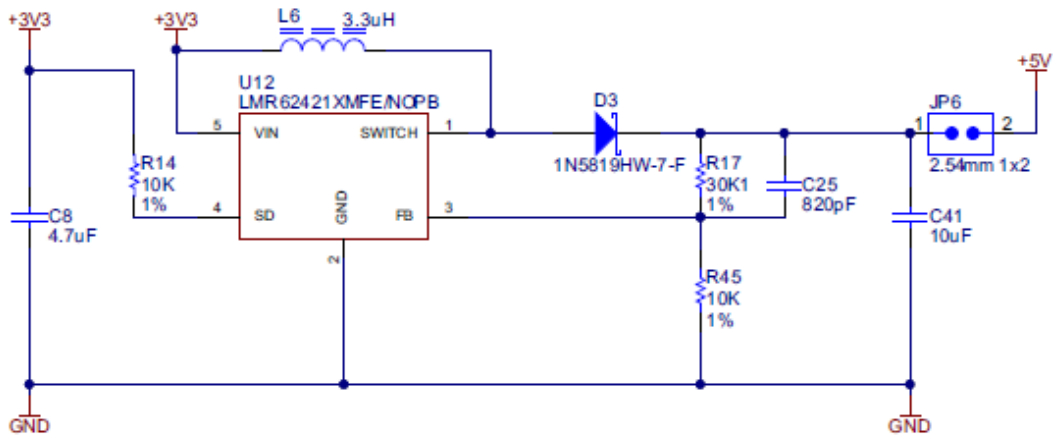


Figura 16. Esquema de convertidor elevador en la tarjeta F283792

Para la tensión de salida, el fabricante indica que esta se puede fijar a partir de un divisor resistivo que se sitúa a la salida de la tensión de salida del pin 3 V_{FB} como se puede apreciar en la *Figura 16*, y que detalladamente tiene la forma que se observa a continuación en la *Figura 17*.

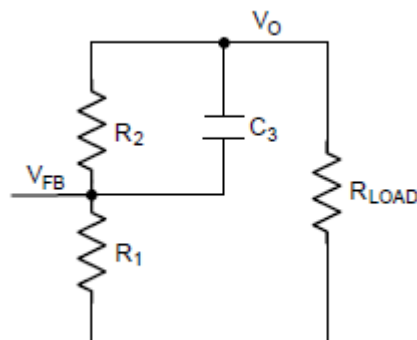


Figura 17. Divisor resistivo para obtener V_o

Observamos que en paralelo con la resistencia R_2 existe un condensador C_3 , la utilidad de este elemento no es otra que añadir una compensación externa al valor de salida para ganar estabilidad. Basándonos en el divisor resistivo, la forma de calcular la salida es:

$$V_0 = V_{FB} * \frac{R2+R1}{R1} \quad (3)$$

El fabricante aconseja un valor para R1 de 10KΩ, y la tensión FB del pin 3 es de 1,255V según indica la hoja de datos [11] así que conociendo la tensión de salida deseada (5V) podemos conocer el valor de R2 despejándolo de la ecuación.

El valor de R2 es 29,84KΩ por lo que se utilizará una resistencia normalizada de 30KΩ.

Volviendo a la *Figura 16* vamos a dar dos apuntes más relacionados con la carga inductiva L6 y los condensadores C8 y C41.

Para el cálculo del valor de la L6, se tiene en cuenta el máximo rizado que se desea de la corriente que es inversamente proporcional a su valor.

El condensador de entrada, C8, tiene la función de evitar picos en la tensión de entrada durante las transiciones, y el condensador de salida, C41, actuará como filtro para reducir el rizado de la tensión de salida.

Por último, no podemos olvidar mencionar el Jump JP6, que como se puede apreciar en la *Figura 16* si este no está conectado, es imposible que el convertidor funcione [9][11].

2.1.3 Planos de PCB analógicos

Para acabar con el apartado de análisis de la alimentación de la tarjeta hemos decidido añadir como último epígrafe el análisis de las PCB que nos aporta el fabricante, para analizar la disposición de las pistas de alimentación que existen en la placa.

El plano de PCB (*Printed Circuited Board*), es el diseño que se realiza sobre una placa para imprimir en ella las pistas conductoras que van a conectar eléctricamente los elementos. El material del que se suelen componer estas pistas es el cobre, y para conseguirlo es necesario hacer pasar a la placa por una serie de procesos como son exposiciones a luz ultravioleta para serigrafiar el circuito, y ataque a la placa con ácidos para eliminar restos de material conductor por la superficie de la placa.

Estos planos se diseñan mediante programas informáticos, en los que se generan los distintos layout (planos) de los que se va a componer el circuito impreso de la placa.

Existen varios tipos de layout en los que la principal diferencia entre ellos es el ancho de las pistas que los componen. Cuanto más ancha sea la pista menos resistencia ofrece al paso de la señal.

Para el diseño de las pistas que llevan tensión, hay que tener en cuenta que el grueso tiene que ser superior que para las pistas que llevan señales digitales, pues es muy importante que el valor de la tensión no se pierda en la resistencia que ofrecen las propias pistas.

A continuación, en la *Figura 18* podemos observar el layout de la tensión Vdd, que como hemos indicado en el apartado 2.1.1 Tensiones internas, tiene un carácter especial y es que su valor de tensión es de 1,2V, valor que no se puede aportar desde una referencia externa y para el que es necesario un circuito reductor de tensión desde 3.3V.

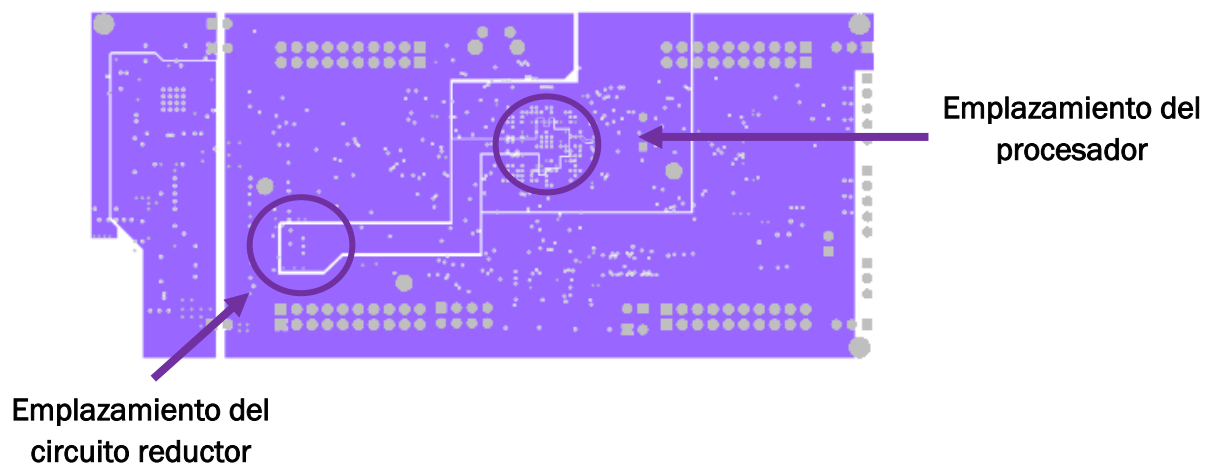


Figura 18. Layout de Vdd

En la figura se puede apreciar como la pista tiene un grueso bastante considerable y el recorrido que realiza está relacionado con lo que ya hemos comentado: La alimentación de los pines del procesador con una tensión reducida de 1,2 V.

Otro layout interesante de analizar es el GND, que se muestra a continuación, en la *Figura 19. Layout de GND*.

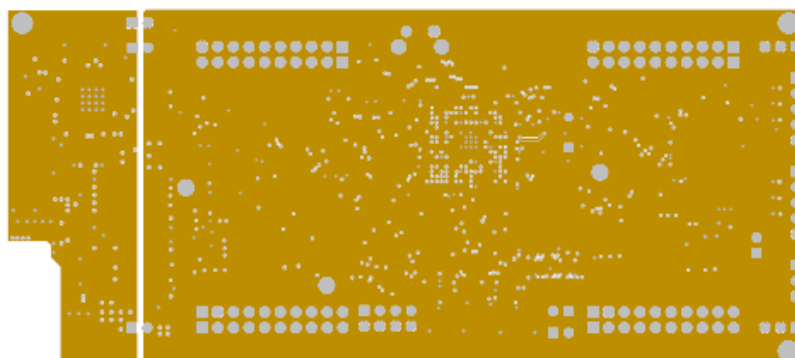


Figura 19. Layout de GND

En este se puede observar cómo no existen pistas si no que son diferentes puntos los que se dan, la explicación no es otra que la existencia de un plano de masa. Cada punto de los que se señalan indica una conexión con este plano y por consiguiente con GND [9].

2.2 PROCESADOR

En este nuevo punto nos vamos a centrar en el elemento más importante de los que se compone la tarjeta, el que lee las señales, y ejecuta las instrucciones para procesar una salida: el procesador.

En el caso de nuestra tarjeta el procesador pertenece a la familia de los C28x de Texas Instruments, que tienen diversas características que los diferencian de los demás. Algunas de ellas son las siguientes:

- Procesador de 32 bits, de punto fijo.
- Arquitectura computacional RISC⁴.
- Arquitectura Harvard modificada que permite ejecutar la lectura de instrucciones o de datos en paralelo con su escritura.
- Ejecución de única instrucción por ciclo.
- Seis buses separados para direcciones y datos.

Además de estas características el procesador está compuesto por muchos más elementos que lo completan y que podrían resumirse en la *Figura 20*, que es un recorte del esquema funcional del microprocesador, donde se puede observar su arquitectura básica.

Una de las características que más destaca a simple vista y por las que se reconoce este procesador es por la aparición de dos CPU, por lo cual se le añade siempre a la descripción el término de *dual-Core*. La razón de contar con dos procesadores es más simple de lo que parece, es una forma de mejorar y optimizar el control añadiendo tareas que se ejecuten en paralelo. Esta idea, en un principio, puede parecer compleja, sobre todo en el campo de las comunicaciones entre ambas CPU, pero para ello se dispone de un bloque común entre ellas que mejora el rendimiento y disminuye la complejidad, este es el bloque IPC. Comentaremos sus características más adelante en el apartado 2.2.4 Comunicación. Modulo IPC.

⁴ Arquitectura computacional caracterizada por instrucciones de tamaño fijo y formato reducido. Además, solo las instrucciones de carga y almacenamiento pueden acceder a la memoria de datos

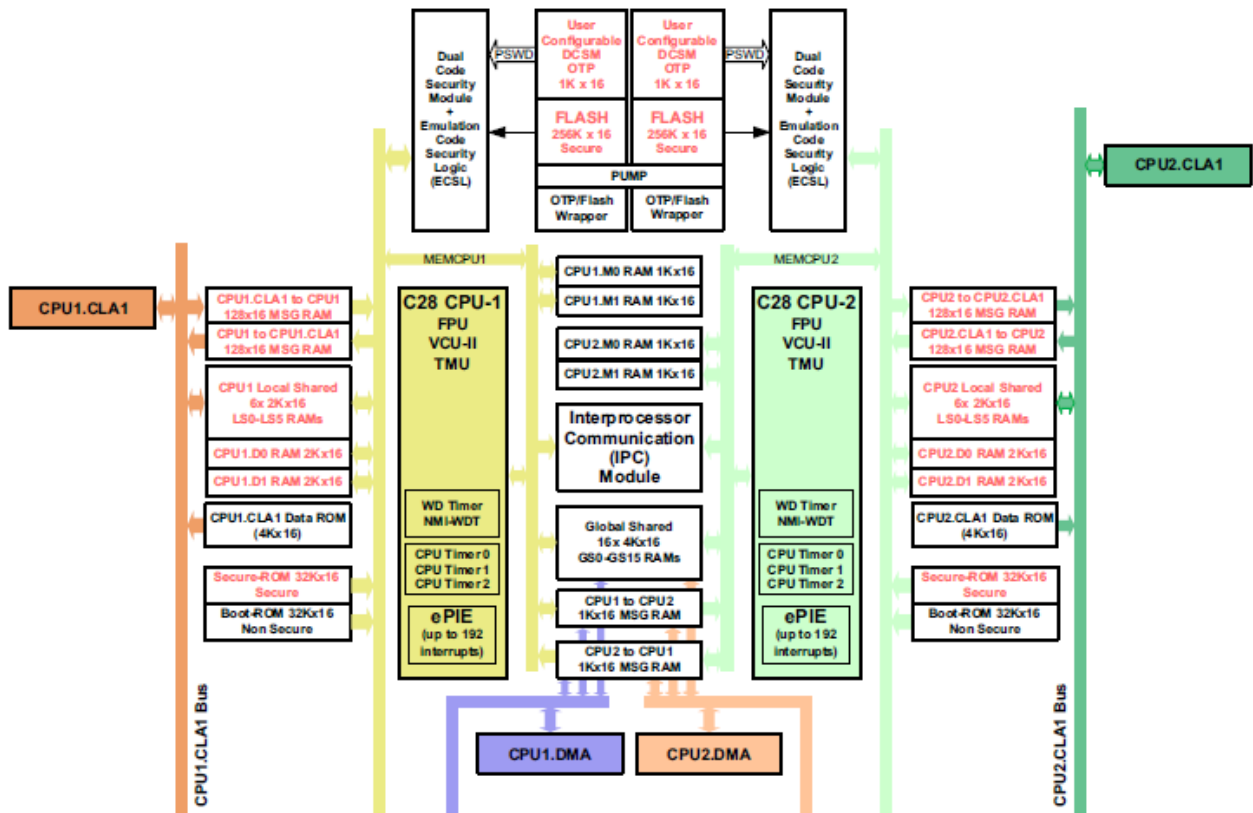


Figura 20. Arquitectura del procesador C28x

Cada procesador, también, va a tener acceso a su propia memoria RAM y FLASH, además del acceso a los datos compartidos en un bloque de la memoria RAM global. El estudio de estos elementos se extiende en el apartado 2.2.2 Memorias.

Añadido a estos elementos que consideramos básicos aparecen otros módulos, independientes para cada CPU, fundamentales para la mejora del rendimiento global del procesador, estos son, el módulo CLA “Control Law Accelerator” (para ejecutar operaciones matemáticas complejas), y el módulo DMA “Direct Memory Access” (para el acceso a memoria). Trataremos en profundidad sus características en el apartado 2.2.3 Módulos externos para mejora del rendimiento.

2.2.1 Características de la CPU

En este primer apartado nos centraremos en analizar las tres principales características de cada CPU y que tienen la función de mejorar las prestaciones iniciales del procesador estas son las siguientes:

FPU (Floating-Point Unit)

Esta unidad consigue habilitar la CPU para el procesamiento de señales codificadas como punto flotante.

Como ya comentamos en las líneas anteriores, los procesadores de la familia C28x solo procesan señales en punto fijo, para así mejorar el rendimiento y evitar complejidad, pero esta extensión FPU añade un set de registros e instrucciones en este formato con el fin de integrarlos en la CPU.

VCU (Viterbi, Complex Math, and CRC Unit)

La función de esta extensión es la de añadir un listado de registros e instrucciones con los que se acelerarán las comunicaciones basadas en algoritmos. También mejora el rendimiento de las FFTs (transformadas rápidas de Fourier), indispensables para realizar el control de cualquier sistema.

Con la ayuda de esta unidad el procesador puede soportar una variedad mayor de tipos de comunicaciones como puede ser la tecnología PLC [12].

TMU (Trigonometric Math Unit)

La unidad TMU es la que se va a encargar de añadir al procesador un conjunto de instrucciones con las que pueda ejecutar operaciones matemáticas complejas, tanto aritméticas como trigonométricas.

Esta unidad, lo mismo que la FPU, añade un soporte para procesar las operaciones matemáticas con punto flotante [12][13].

En la *Tabla 1* podemos observar las operaciones que, gracias a esta extensión, el procesador va a poder ejecutar:

Tabla 1. Operaciones matemáticas que se añaden con la unidad TMU

INSTRUCTIONS	C EQUIVALENT OPERATION	PIPELINE CYCLES
MPY2PIF32 RaH,RbH	$a = b * 2\pi$	2/3
DIV2PIF32 RaH,RbH	$a = b / 2\pi$	2/3
DIVF32 RaH,RbH,RcH	$a = b/c$	5
SQRTF32 RaH,RbH	$a = \text{sqrt}(b)$	5
SINPUF32 RaH,RbH	$a = \sin(b*2\pi)$	4
COSPUF32 RaH,RbH	$a = \cos(b*2\pi)$	4
ATANPUF32 RaH,RbH	$a = \text{atan}(b)/2\pi$	4
QUADF32 RaH,RbH,RcH,RdH	Operation to assist in calculating ATANPU2	5

2.2.2 Memorias.

Un elemento fundamental del procesador es la memoria, y como ya hemos comentado en la introducción, al tratarse de arquitectura Harvard, se van a diferenciar al menos dos memorias, una para datos y otra para instrucciones, en este caso la memoria RAM y la Flash.

Cada CPU, posee una estructura o mapa de memoria que señala y define los distintos bloques de la memoria a los que se puede acceder. Esta estructura se muestra en la Figura 20 y a partir de ella vamos a ir analizando las distintas memorias.

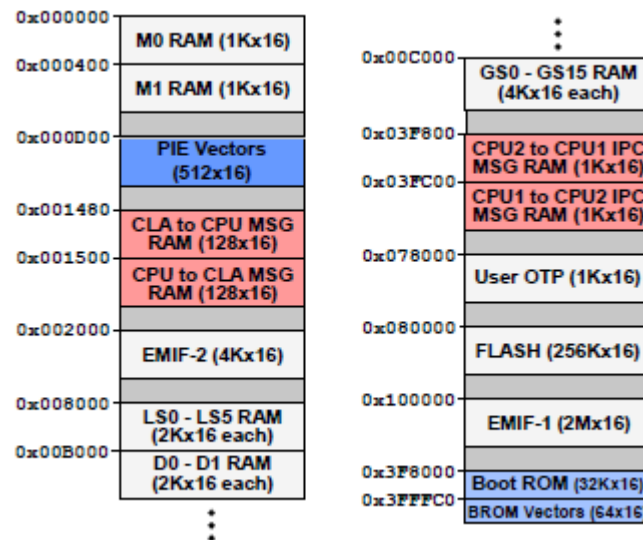


Figura 20. Mapa de memoria del procesador C28x

2.2.2.1 Memoria RAM

Cada CPU, va a tener dedicada a la memoria RAM cuatro bloques **M0**, **M1**, **D0** y **D1**, estos bloques son de acceso restringido solo para cada CPU, no se van a compartir ni con ninguna otra memoria ni con ningún módulo. Para asegurar esto se cuenta con la protección ECC (error correcting-code).

M0 y M1, tienen el tamaño de 1Kx16 palabras cada una, D0 y D1, tienen capacidad de hasta 2Kx16 palabras.

Se cuenta también con seis bloques compartidos de cada CPU con el módulo CLA, estos son desde **LS0** hasta **LS5**, cada uno tiene una capacidad de almacenamiento de 2Kx16 palabras.

El mayor número de bloques se encuentra en los que son compartidos con la otra CPU (global shared), y con el módulo DMA, que van desde **GS0** a **GS15**. Cada bloque cuenta con una capacidad de 4Kx16 palabras.

Gracias al registro GSxMSEL, se puede gestionar el acceso a memoria de parte de cada una de las CPU y su respectivo bloque DMA. En el caso de que la CPU1 reserve un bloque, entonces ella y su módulo DMA tendrán acceso total (búsqueda lectura y escritura), mientras que la CPU2 y su respectivo DMA solo tendrán permiso de lectura, al inverso ocurre exactamente lo mismo. En la *Tabla 2* puede observar la gestión de estos bloques GSx.

Por último, para terminar con el análisis de la memoria RAM, nos falta analizar los bloques de mensajes que tienen como función compartir datos entre las dos CPU o entre el bloque CLA y cualquier CPU.

Tabla 2. Gestión de bloque Global Shared en memoria RAM

GSxMSEL	CPU1	CPU1	CPU1	CPU1.DMA	CPU1.DMA	CPU2	CPU2	CPU2	CPU2.DMA	CPU2.DMA
	Fetch	Read	Write	Read	Write	Fetch	Read	Write	Read	Write
0	Yes	Yes	Yes	Yes	Yes	No	Yes	No	Yes	No
1	No	Yes	No	Yes	No	Yes	Yes	Yes	Yes	Yes

El bloque de mensajes con CPU se subdivide a su vez en el bloque **CPU2 to CPU1 IPC MSG RAM** para compartir datos desde CPU2 hacia CPU1 y el bloque **CPU1 to CPU2 IPC MSG RAM** que funciona a la inversa. Como se puede comprobar la principal función de estos bloques es la intercomunicación entre CPUs por ello se les añade IPC en la descripción (Interprocessor Communication). El tamaño de cada bloque es de 1Kx16 palabras.

El otro bloque de mensajes para la comunicación con el bloque CLA se divide de la misma manera en dos bloques con funciones inversas. El bloque **CLA to CPU MSG RAM** y el bloque **CPU to CLA MSG RAM**. Cada uno tiene un tamaño de 1Kx16 palabras.

Podemos observar en la *Figura 21* un esquema que nos va a servir para entender la arquitectura de la memoria RAM, con los bloques que acabamos de explicar y la relación que tienen con las CPU, las DMA y las CLA [13] [14].

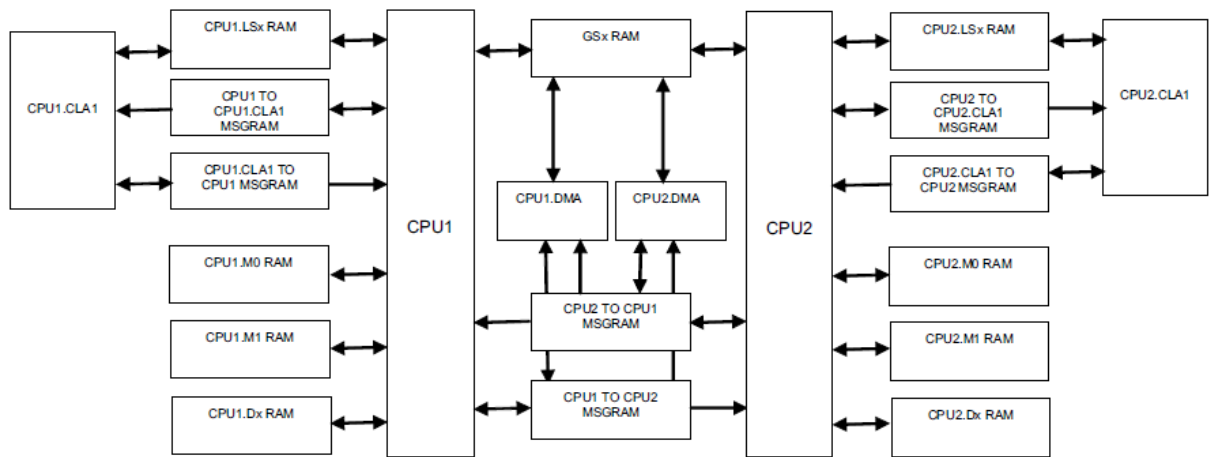


Figura 21. Arquitectura de la memoria RAM

2.2.2.2 Memoria Flash

La memoria Flash es una memoria programable que puede ser modificada las veces que sean necesarias. En este caso tiene la función principal de guardar el código de los programas, pero también es capaz de guardar datos estáticos.

Cada CPU tiene asignado un banco de memoria con las siguientes características:

- Tamaño máximo de 256Kx16 palabras por banco.
- Cada banco posee un controlador denominado FMC (Flash Module Controller).
- Protección ECC para la corrección y detección de un error.
- Protección SECDED para la detección de doble error.

CPU 1 o CPU 2 interactúan con la memoria a partir del controlador FMC, que va a gestionar el acceso a la lectura de programa de la memoria FLASH basado en un sistema de turnos. Este sistema se observa a continuación en la *Figura 22*.

El funcionamiento básicamente consiste en el control de un semáforo que da acceso a una bomba (pump) que dirige la búsqueda del programa al banco de memoria correspondiente. Tanto la CPU 1 como la CPU 2 comparten la bomba de redirección.

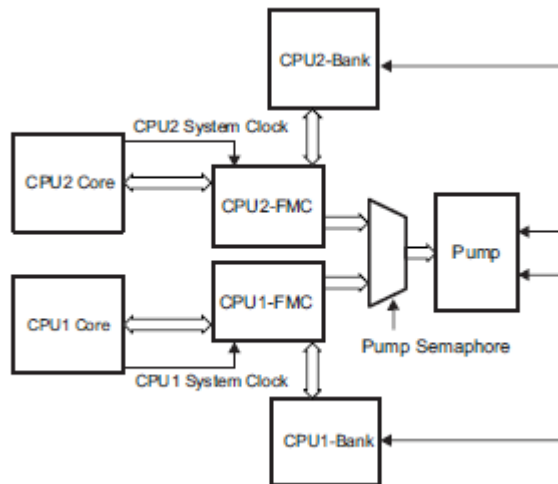


Figura 22. Sistema de acceso a memoria Flash

Al conectar la tarjeta LAUNCHXL-F28379D al ordenador mediante el uso del programa Coder Composer Studio (CCS) podemos visualizar la gestión de la memoria del procesador mediante un código que ya viene predeterminado de fábrica. En la Figura 23 y en la Figura 24 lo podemos observar:

```

MEMORY
{
PAGE 0 : /* Program Memory */
        /* Memory (RAM/FLASH) blocks can be moved to PAGE1 for data allocation */
        /* BEGIN is used for the "boot to Flash" bootloader mode */

BEGIN          : origin = 0x080000, length = 0x000002
RAMM0          : origin = 0x000122, length = 0x0002DE
RAMD0          : origin = 0x00B000, length = 0x000800
RAMLS0         : origin = 0x008000, length = 0x000800
RAMLS1         : origin = 0x008800, length = 0x000800
RAMLS2         : origin = 0x009000, length = 0x000800
RAMLS3         : origin = 0x009800, length = 0x000800
RAMLS4         : origin = 0x00A000, length = 0x000800
RAMGS14        : origin = 0x01A000, length = 0x001000
RAMGS15        : origin = 0x01B000, length = 0x001000
RESET          : origin = 0x3FFFC0, length = 0x000002

/* Flash sectors */
FLASHA        : origin = 0x080002, length = 0x001FFE /* on-chip Flash */
FLASHB        : origin = 0x082000, length = 0x002000 /* on-chip Flash */
FLASHC        : origin = 0x084000, length = 0x002000 /* on-chip Flash */
FLASHD        : origin = 0x086000, length = 0x002000 /* on-chip Flash */
FLASHE        : origin = 0x088000, length = 0x008000 /* on-chip Flash */
FLASHF        : origin = 0x090000, length = 0x008000 /* on-chip Flash */
FLASHG        : origin = 0x098000, length = 0x008000 /* on-chip Flash */
FLASHH        : origin = 0x0A0000, length = 0x008000 /* on-chip Flash */
FLASHI        : origin = 0x0A8000, length = 0x008000 /* on-chip Flash */
FLASHJ        : origin = 0x0B0000, length = 0x008000 /* on-chip Flash */
FLASHK        : origin = 0x0B8000, length = 0x002000 /* on-chip Flash */
FLASHL        : origin = 0x0BA000, length = 0x002000 /* on-chip Flash */
FLASHM        : origin = 0x0BC000, length = 0x002000 /* on-chip Flash */
FLASHN        : origin = 0x0BE000, length = 0x002000 /* on-chip Flash */

```

Figura 23. Mapa de memoria de programa desde CCS

```

PAGE 1 : /* Data Memory */
        /* Memory (RAM/FLASH) blocks can be moved to PAGE0 for program allocation */

BOOT_RSVD      : origin = 0x000002, length = 0x000120    /* Part of M0, BOOT rom will use this for stack */
RAMM1          : origin = 0x000400, length = 0x000400    /* on-chip RAM block M1 */
RAMD1          : origin = 0x00B800, length = 0x000800

RAMLS5         : origin = 0x00A800, length = 0x000800

RAMGS0         : origin = 0x00C000, length = 0x001000
RAMGS1         : origin = 0x00D000, length = 0x001000
RAMGS2         : origin = 0x00E000, length = 0x001000
RAMGS3         : origin = 0x00F000, length = 0x001000
RAMGS4         : origin = 0x010000, length = 0x001000
RAMGS5         : origin = 0x011000, length = 0x001000
RAMGS6         : origin = 0x012000, length = 0x001000
RAMGS7         : origin = 0x013000, length = 0x001000
RAMGS8         : origin = 0x014000, length = 0x001000
RAMGS9         : origin = 0x015000, length = 0x001000
RAMGS10        : origin = 0x016000, length = 0x001000
RAMGS11        : origin = 0x017000, length = 0x001000
RAMGS12        : origin = 0x018000, length = 0x001000
RAMGS13        : origin = 0x019000, length = 0x001000

CPU2TOCPU1RAM : origin = 0x03F800, length = 0x000400
CPU1TOCPU2RAM : origin = 0x03FC00, length = 0x000400
}

```

Figura 24. Mapa de memoria de datos desde CCS

Podemos ver todos los bloques que hemos ido comentando en las anteriores capturas donde además se indica la dirección y la longitud de estos.

Lo primero que llama la atención es cómo los bloques de memoria M, D y LS se añaden en la página 0 junto al contenido de la memoria de programa como son los distintos sectores de la memoria FLASH, esto indica que algún programa es posible que se pueda almacenar en la memoria RAM en alguno de estos bloques.

Existe la opción de mover los bloques de la página 0 a la 1 o viceversa con el fin de colocarlos como memoria de programa o de datos respectivamente

También se observan otras peculiaridades como los bloques BEGIN o RESET (bloques de inicio o de reinicio de la memoria FLASH), los 14 sectores de la memoria FLASH de distintitos tamaños, o el curioso caso de los bloques RAMGS14 y RAMGS15 que se utilizan para el almacenamiento de programa en la RAM como ya hemos indicado antes [13] [14].

2.2.2.3 Memoria OTP

Memoria que solo se puede programar una vez, la memoria OTP tiene el mismo espacio y mapa de memoria que la memoria Flash. Esta memoria también tiene la peculiaridad de que no puede ser borrada.

Dentro de la memoria OTP se pueden distinguir dos bloques el bloque TI, y el bloque USER cada uno de un tamaño de 1Kx16 palabras:

Bloque TI-OTP

Contiene distintos datos de calibración de los convertidores ADC o DAC y de los osciladores internos, además de distintas configuraciones para poder borrar o programar operaciones la memoria FLASH. Este bloque solo se puede leer nada de escribir ni borrar.

Bloque USER-OTP

Bloque que contiene dentro principalmente distintos parámetros programables relacionados con la seguridad de la información en el procesador. Estos pueden ser, por ejemplo, las contraseñas de bloques de memoria protegidos, o la localización de las contraseñas que se dan por defecto al adquirir la tarjeta. Este bloque si que puede ser programado y por ello comparte la bomba de redirección (pump) con la memoria Flash,

La programación de las dos memorias se realiza desde el módulo DCSM que asegura la protección de los datos. A continuación, vamos a comentar sus peculiaridades [13] [14].

2.2.2.4 Módulo DCSM

El módulo DCSM o módulo de seguridad para código dual, es una característica de seguridad que incorpora el procesador para prevenir intrusiones o duplicados de memorias protegidas, por parte de personas desautorizadas a través del puerto JTAG o periféricos externos. Cada CPU contiene protección DCSM y esta a su vez dos zonas de seguridad para cada CPU, zona 1 y zona 2. Cada zona posee el mismo mecanismo de seguridad, y se puede seleccionar que bloques queremos incluir en cada zona.

Cada zona posee sus propios recursos de seguridad, para configurarlos se reserva un contenido en el bloque USER OTP de la memoria OTP donde se guardan.

Para comprender el funcionamiento de este módulo mejor tomamos el ejemplo de la memoria RAM donde los bloques Dx y LSx es necesario que sean seguros para no permitir ni la lectura ni la edición de los datos que contienen. Estos bloques se pueden localizar en cualquiera de las dos zonas siempre y cuando se configure la localización de su GRABRAM en el bloque USER OTP.

Cada zona posee también su propia contraseña de 128 bits (o cuatro palabras de 32 bits) que también será gestionada por la memoria OTP.

A continuación, en la *Figura 25*, vamos a indicar el diagrama de una secuencia denominada PMF (Password Match Flow) que a través de la simulación de lecturas de las localizaciones de cuatro contraseñas escribe su resultado en los registros CSMKEY con el fin de iniciar los registros de seguridad y deshabilitarlos para tener acceso a ellos y probar la efectividad de las contraseñas.

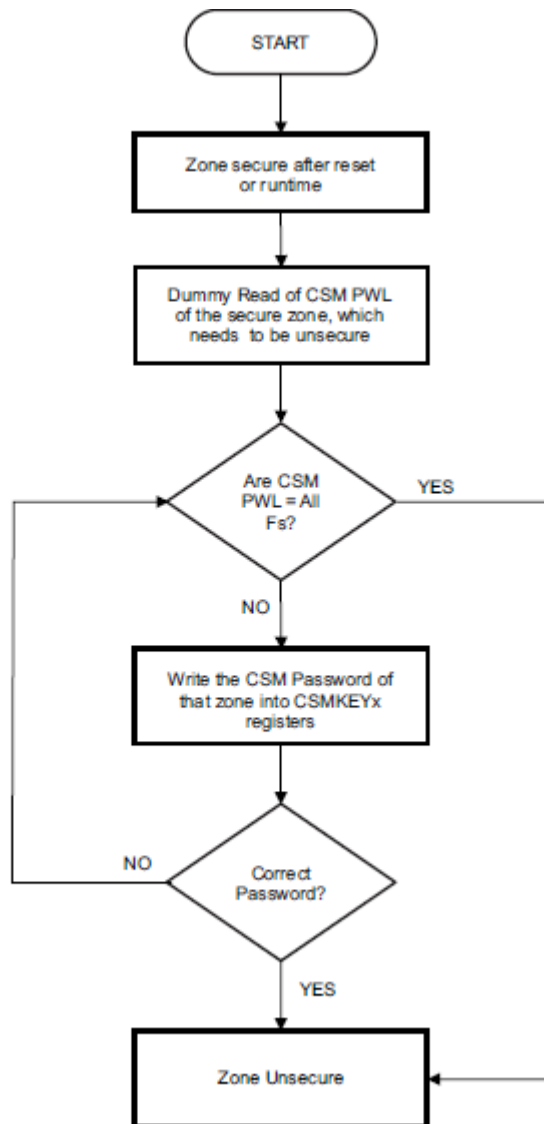


Figura 25. Password Match Flow (PMF)

Esta es la única forma de leer contenido bloqueado por el módulo DCSM, como por ejemplo el del bloque USER OPT que contiene la configuración de las zonas de seguridad [14].

2.2.3 Módulos externos para mejora del rendimiento

En el apartado que comenzamos a continuación vamos a analizar dos módulos físicos que se añaden al procesador con el fin de mejorar su rendimiento añadiendo mejoras a la comunicación entre CPUs y al acceso a las memorias, estos módulos son el DMA y el CLA:

2.2.3.1 DMA

El módulo de acceso directo a memoria (Direct Memory Access) proporciona un método de transferencia de datos entre periféricos y/o memorias sin intervención de la CPU. Cada CPU va a poseer uno de estos módulos que les van a ser muy útiles cuando la cantidad de información a transferir (a la memoria o entre periféricos) es demasiada y el procesador tiene otras tareas pendientes que de esta forma se podrán ejecutar en paralelo a la transferencia de datos.

Hay que tener en cuenta que la memoria, la CPU y el módulo DMA comparten bus por ello es fundamental que exista una comunicación ajena a este entre ellos para gestionar su uso, no pueden acceder simultáneamente. A través de este contacto la CPU va a configurar los puertos que desea que gestione DMA, y esta va a recibir la información de disponibilidad desde los periféricos. En la *Figura 26* se detalla un esquema de la comunicación que existe entre diferentes elementos de la tarjeta y el módulo DMA, donde con “handshake” nos referimos a los canales de comunicación que ya hemos comentado [15].

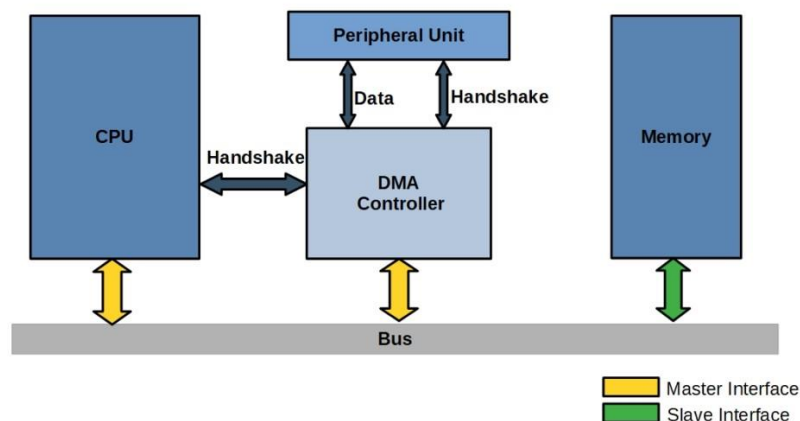


Figura 26. Esquema de comunicación con el módulo DMA

El módulo posee seis canales con su correspondiente interrupción PIE⁵ como podemos apreciar en la *Figura 27*. Desde cada uno de ellos puede llegar una señal de interrupción y en ese caso es cuando el DMA va a ejecutar su ciclo. Anteriormente mediante la línea de comunicación con la CPU (handshake), esta va a indicar a la DMA cuales son las direcciones de los registros de origen de los datos y de destino. El ciclo de funcionamiento del módulo DMA es el siguiente:

1. El puerto de origen lanza una interrupción al obtener un dato a través del PIE.
2. El módulo detecta nuevo dato disponible en el registro de lectura del puerto mediante la comunicación con este (handshake).
3. Lectura de registro del puerto por parte del DMA.
4. Almacenamiento de la lectura en el buffer FIFO del módulo DMA.
5. El controlador del módulo comprueba que el bus de transporte de datos no está ocupado y actúa dependiendo del modo de uso (explicado en la siguiente página).
6. Transferencia desde el buffer a la dirección destino indicada por la CPU.
7. Si hay más datos que transferir se incrementa la dirección de memoria y se vuelve al paso 2, si no se finaliza el ciclo [13] [15].

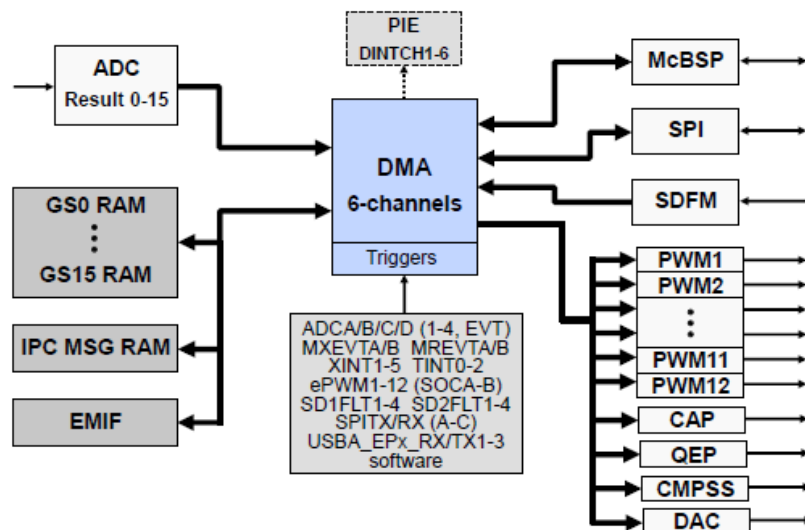


Figura 27. Direct Memory Access (DMA)

⁵ Peripheral Interrupt Expansion. Es básicamente, un bloque de expansión de interrupciones en forma de multiplexor. Como al procesador solo pueden llegar un máximo de 16 interrupciones (INT1-INT14, RTOSINT y DLOGINT), este sirve para ampliar este rango y arbitrar más. [45]

Según el uso de memoria que se realice por parte del módulo pueden existir tres modos de uso:

- **Modo Cycle Stealing:** Después del uso de memoria para transmitir una cantidad dada de bits la DMA devuelve el acceso a la CPU. Misma prioridad de CPU y DMA.
- **Modo Burst:** El módulo manda una interrupción a la CPU que la mantiene en espera hasta que se realice por completo la transferencia de datos a la memoria. Más prioridad DMA sobre CPU.
- **Modo Transparente:** El DMA solo podrá acceder a la memoria cuando la CPU no la esté utilizando. Más prioridad de CPU sobre DMA[16].

2.2.3.2 CLA

El bloque CLA es básicamente un acelerador de hardware de 32 bits en punto flotante que ejecuta complejos algoritmos matemáticos en paralelo a la CPU. Posee también el acceso directo a periféricos de control y comunicaciones que permite evitar la sobrecarga del procesador.

Un problema muy importante al que se enfrentan siempre los sistemas de control en tiempo real es el exceso de **latencia**, producido por la sucesión de retardos en las tareas, que conducen a una ejecución lenta del control. Con el módulo CLA este problema se va a minimizar considerablemente gracias al acceso directo a periféricos sin pasar por la CPU.

La principal peculiaridad que posee cualquier CLA es la aparición de ocho nuevos bloques denominados “tareas” que van a ser como una especie de interrupciones pero que no están gestionadas desde la CPU. Cada tarea puede ser programada independientemente de las demás desde un código externo y puede ser capaz de ser activada por una cantidad específica de periféricos sin la intervención de la CPU.

En la *Figura 29* podemos observar la estructura básica de este módulo, donde cobran gran importancia las interrupciones mediante tareas (task).

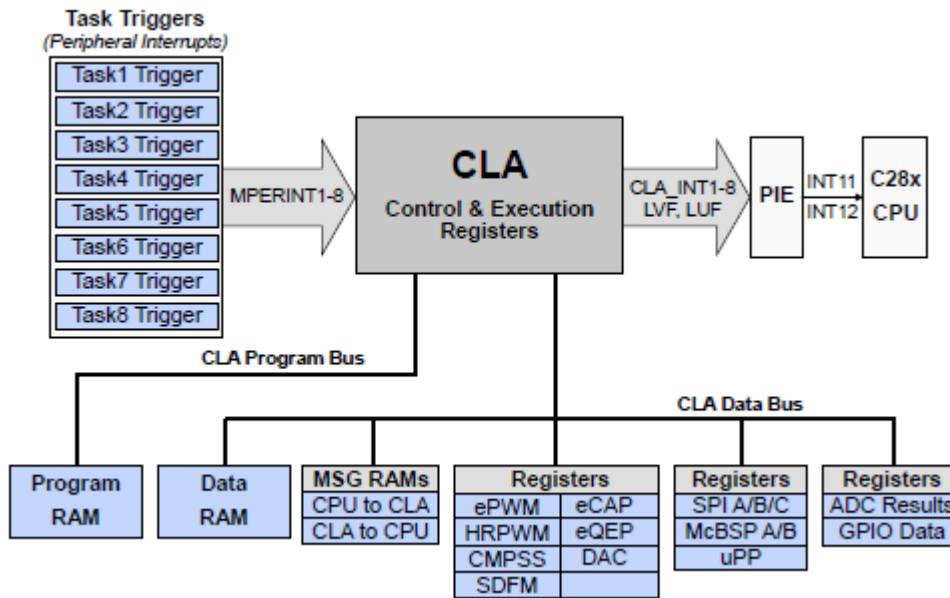


Figura 29. Módulo CLA

Para la programación de las tareas podemos utilizar múltiples opciones dado que las vamos a tratar como interrupciones.

En **Simulink** al añadir el add-on de Texas Instruments, existe un bloque que permite la configuración de las tareas y la opción de gestionar la interrupción desde algún puerto o desde un programa.

El bloque como se muestra en la *Figura 28* se llama CLA Task y posee una única salida que es hacia dónde va a ser dirigida la acción desencadenada por la tarea.

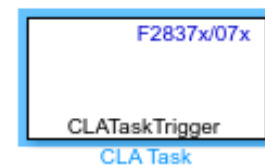


Figura 28. Bloque CLA en Simulink

Dentro de los parámetros que podemos editar está el número de la tarea que se va a programar, desde donde se va a solicitar la interrupción y la duración total de esta. El bloque de edición de parámetros es el siguiente:

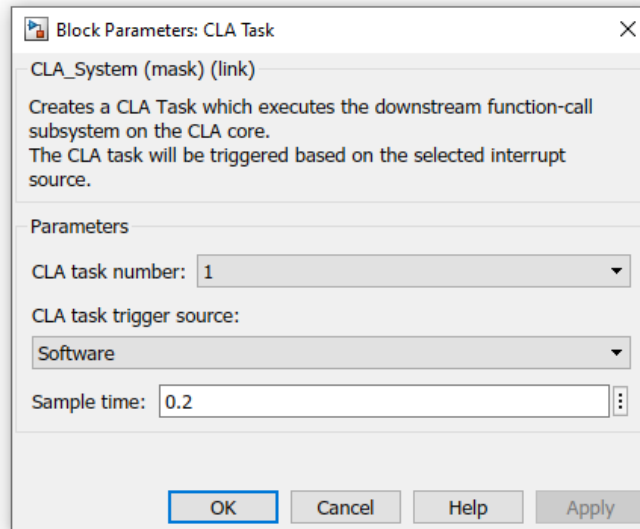


Figura 29. Parámetros del bloque CLA

Algunas de las opciones desde donde se puede gestionar la interrupción son: el Software (código en C/C++), el puerto ADC, el puerto PWM, puerto CAP...[12][13][14]

2.2.4 Comunicación. Módulo IPC

El módulo IPC (Inter-Processor Communications), es el módulo que va a gestionar la comunicación entre las dos CPU. Este módulo se va a encargar de la gestión de los bloques de memoria “CPU1 to CPU2” y “CPU2 to CPU1” entre otras cosas ya que va a disponer del uso de 32 señales de evento en forma de flags o interrupciones.

La forma de intercambiar mensajes entre CPUs se basa en la existencia de ocho registros de comando por cada CPU, cuatro para recibir, y cuatro para mandar mensajes, mayoritariamente complejos.

Para mandar mensajes existen tres registros de lectura/escritura y uno de lectura solo, para recibirlos existen tres registros de lectura y uno de lectura/escritura. En la *Tabla 3* se pueden observar los ocho registros con sus posibles opciones de edición:

Tabla 3. Registros de comando

Local Register Name	Local CPU	Remote CPU	Remote Register Name
IPSENDCOM	R/W	R	IPCRECVCOM
IPSENDADDR	R/W	R	IPCRECVADDR
IPSENDATA	R/W	R	IPCRECVDATA
IPCREMOTEREPLY	R	R/W	IPCLOCALREPLY

Existen otros bloques de este módulo que componen el protocolo de comunicación algunos de los más importantes son los siguientes:

- **IPCSET:** Registro formado por 32 banderas (flags) a las que la CPU indica su intención de comenzar una comunicación. Las cuatro primeras indican una interrupción en la CPU remota, los 28 siguientes no.
- **IPCACK:** Registro formado por 32 banderas (flags) que reciben la señal de la otra CPU para indicar que la comunicación ha sido realizada correctamente.
- **IPCCLR:** Registro de 32 bits que sirve para cancelar cualquier evento desde la CPU que lo ha inicializado. La CPU que recibe la comunicación puede cancelarla desde el registro IPCACK.
- **IPCFLG:** Registro de lectura por parte de la CPU que inicia la comunicación. Gestiona el estado de las banderas en cada etapa de la comunicación.
- **IPCSTS:** Registro de lectura por parte de la CPU que recibe la comunicación. Colecciona un conjunto de banderas que sirven para conocer el estado de la comunicación des de la CPU.

En una comunicación, IPCSET coloca la bandera correspondiente de IPCSTS y de IPCFLG a 1, con esto la CPU a la que es destinado el mensaje, tras leer IPCSTS, sabe que tiene que leer los registros de comunicación IPCRECV. La CPU que manda el mensaje gracias a la bandera activa en IPCFLG, sabe que puede escribir el mensaje en los registros de comunicación IPCSEND. Cuando termina la comunicación, la CPU receptora, indica gracias a IPCACK que todo está correcto, y las banderas de IPCFLG y IPCSTS se vuelven a poner a 0 indicando a la CPU emisora que la comunicación a terminado.

En la *Figura 30* se puede observar la arquitectura del módulo IPC [14].

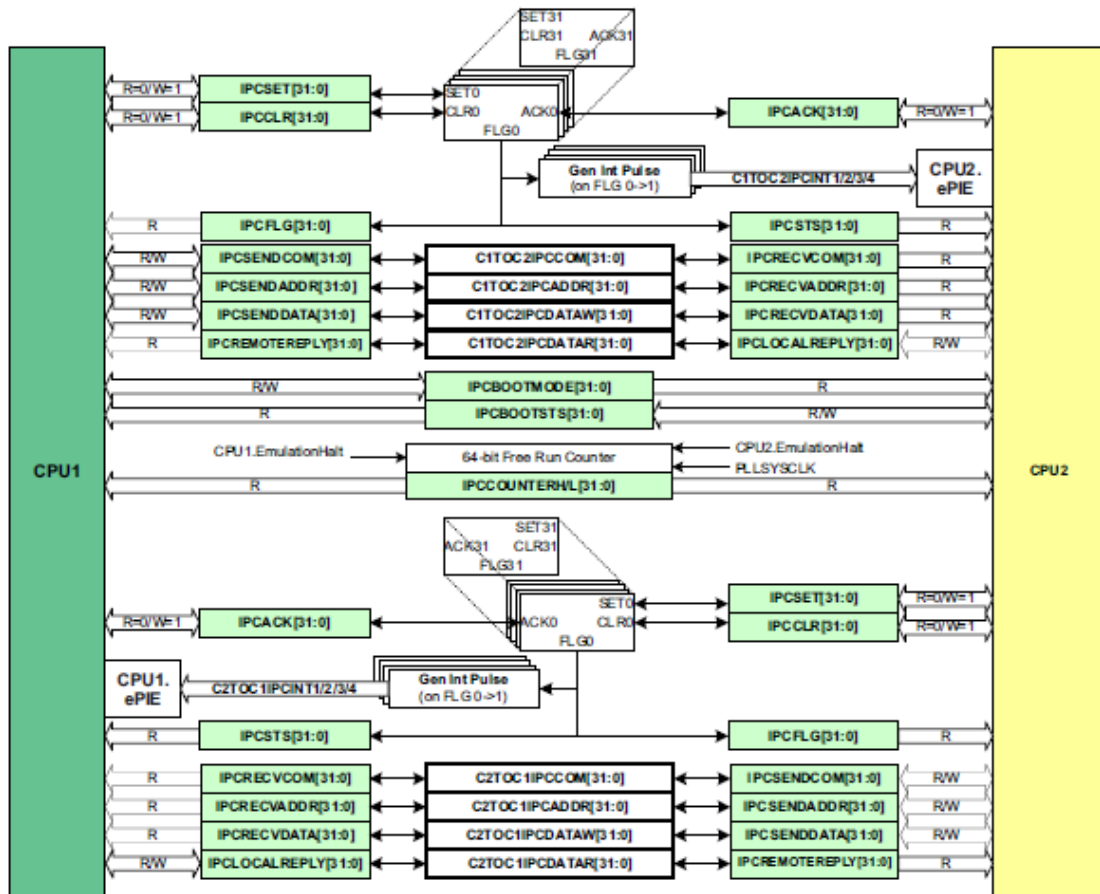


Figura 30. Arquitectura del módulo IPC

Para comprobar la función de este módulo, en la tarjeta LAUNCHXL-F28379D realizamos una sencilla simulación en Simulink gracias a los dos bloques que disponemos referentes al módulo IPC. Estos son los que se indican en la Figura 31 y en la Figura 32:

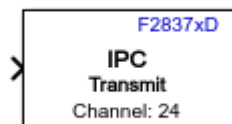


Figura 31. Bloque IPC Transmit



Figura 32. Bloque IPC Receive

El bloque IPC Transmit se encarga de enviar la señal desde la CPU origen, el único parámetro que podemos seleccionar es el número de canal por el que se desea mandar la información. Solo tiene una única entrada por donde llega el dato a enviar a la CPU.

El bloque IPC Receive, gestiona la llegada de la señal a la otra CPU, a través de la salida Out muestra su valor, y a través de la salida Status muestra el estado de la comunicación en función de distintos valores que se muestran en la *Tabla 4*.

Tabla 4. Tabla de estados del bloque IPC Receive en Simulink

Estado	Estado IPC
0	No hay errores
1	Dato no disponible
2	El tipo de dato no coincide
4	El tamaño del dato no coincide
6	El tipo o el tamaño del dato no coincide

Los parámetros que se pueden modificar en este bloque son: el número del canal (tiene que coincidir con el del bloque IPC Transmit), el tipo de dato que se recibe (single, int8, uint8, boolean...), el tamaño del dato, y el tiempo de muestreo.

En la simulación realizamos vamos a tratar de enviar un entero, tipo uint32 desde la CPU1 a la CPU2. El programa nos exige ejecutar primero la transmisión y luego la lectura, así que creamos dos simulaciones la primera “IPC_CPU1” se ejecuta desde la CPU1 a través del canal 24 como se indica en la *Figura 33*.

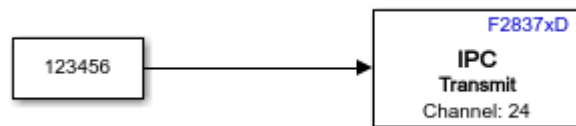


Figura 33. Simulación IPC_CPU1

En la otra simulación en “IPC_CPU2”, se recibe el dato en la CPU2, indicando que se trata de un dato uint32 con tamaño 1 y un tiempo de muestreo de 0.5 segundos. El diagrama de bloques es el de la *Figura 34*.

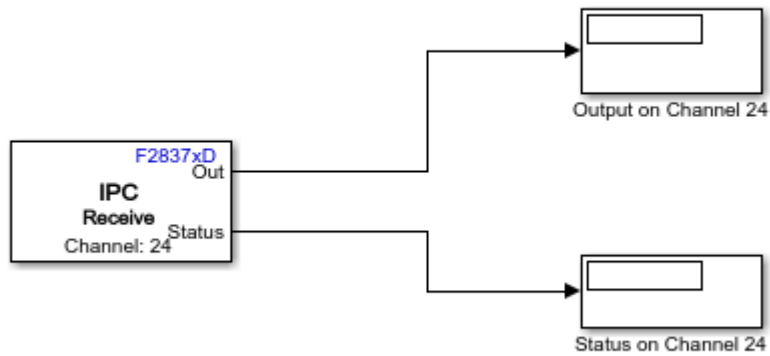


Figura 34. Simulación IPC_CPU2

Antes de ejecutar las simulaciones en la tarjeta tenemos que ajustar algunos parámetros de conexión con el Hardware en Simulink, como son el tipo de tarjeta, el puerto por el que se conecta y la CPU que vamos a utilizar (distinta en cada caso)

La simulación en CPU1 se ejecuta primero y se implementa en el hardware, en cambio la simulación en CPU2, se ejecuta y también lee los resultados. Siguiendo estos pasos en la Figura 35 comprobamos que la comunicación se ha dado correctamente.

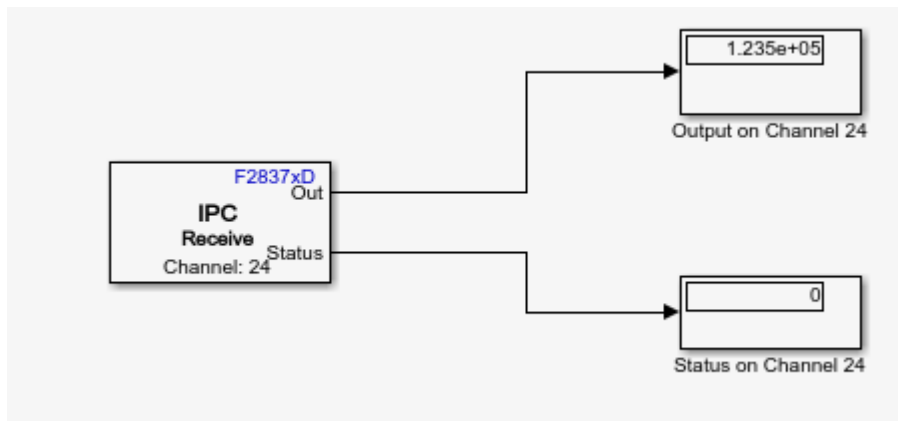


Figura 35. Resultado final de la simulación en IPC_CPU2

2.3 MÓDULO PWM

En este tercer apartado de análisis de la tarjeta vamos a centrar nuestro estudio en el módulo PWM, fundamental para entender el propósito final de este trabajo que no es otro que la aplicación del controlador en electrónica de potencia. Para ello vamos a dividir este apartado en diferentes bloques. En el primero explicaremos los conceptos básicos del control PWM, en el segundo describiremos los periféricos ePWM que incluye la tarjeta, y finalmente mostraremos la aplicación del módulo mediante Simulink

2.3.1 Control PWM

La modulación por ancho de pulso (Pulse Width Modulation) es una técnica de control muy utilizada para elementos de potencia como pueden ser los inversores, aunque no solamente con ellos también es muy útil para el control de velocidad de un motor o el control de una iluminación LED.

Esta técnica consiste básicamente en modificar el ciclo de trabajo D de cualquier señal periódica que cumple la función de “interruptor”, cuanto más pequeño sea el ciclo de trabajo D, más tiempo está el “interruptor” abierto y la potencia que se inyecte al sistema será menor, justo al contrario que con un ciclo D mayor cercano a 100% donde inyecta al sistema toda la potencia.

El módulo PWM posee distintas partes: dos entradas, una salida y un comparador. Las dos entradas son distintas y cada una tiene una función:

- **Entrada portadora:** Va a transmitir la señal una señal triangular en forma normalmente de dientes de sierra para determinar la frecuencia de conmutación de la señal de salida PWM.
- **Entrada moduladora:** Esta entrada se va a encargar de transmitir la señal que va a establecer el ciclo de trabajo de la salida PWM, así como la frecuencia del armónico fundamental. Esta entrada puede tratarse de una señal senoidal, continua, cuadrada...

Estas dos entradas van a llegar a un comparador de señales que observando el valor en cada instante de las dos señales establecerá la salida en valor alto o en valor bajo, según la condición que se establezca entre las dos posibles:

- Valor de portadora \geq Valor de moduladora.
- Valor de moduladora \leq Valor de portadora.

El comparador se puede tratar de un amplificador operacional como es el caso del ejemplo de la *Figura 36* donde se muestra una modulación a partir de una señal V_c en forma de rampa y una señal V_{ramp} en forma de dientes de sierra.

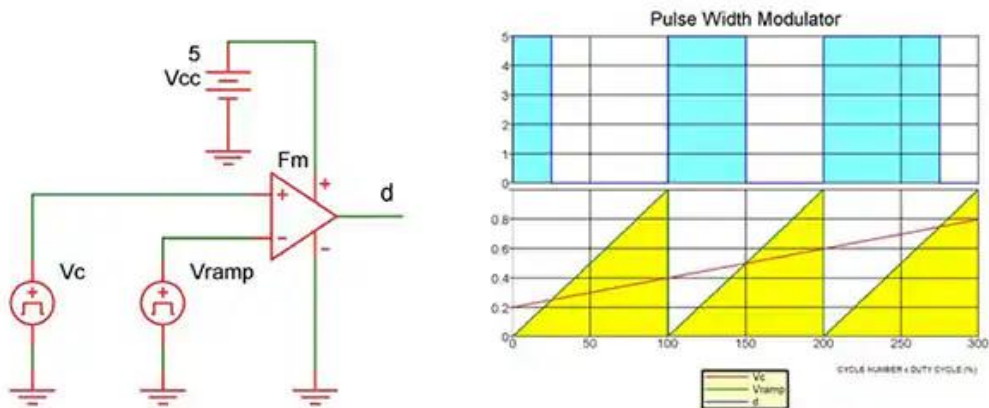


Figura 36. Ejemplo de generación de señal PWM

En este caso la señal V_c está conectada a la entrada positiva del amplificador por lo que siempre que la rampa tenga un valor mayor que el diente de sierra la señal de salida coincide con la tensión de saturación del amplificador en positivo a 5V. En el caso contrario satura a negativo a 0V. V_{ramp} fija la frecuencia de la señal así que es la **portadora** y V_c fija el ciclo de servicio D así que se corresponde con la **señal moduladora**.

Muchas son las ventajas de este control, entre ellas que se reduce el ruido, se ahorra energía o que se simplifican los cálculos. Entre los inconvenientes el más importante se encuentra en el problema común en toda la electrónica de potencia y no es otro que la aparición de armónicos que generan ruido en el sistema [17].

Mas adelante, en el apartado 3.2 CONTROL DEL INVERSOR, añadimos más información acerca de este control en inversores.

2.3.2 Módulo ePWM en la tarjeta LAUNCHXL-F28379D

El control PWM en la tarjeta LAUNCHXL-F28379D está representado mediante un módulo denominado ePWM compuesto por dos salidas: PWMxA y PWMxB. Dentro de la tarjeta existen hasta 12 módulos diferentes que siguen la misma estructura que vamos a explicar en este apartado.

En la *Figura 37* se puede observar el esquema base que vamos a seguir para explicar la estructura del módulo ePWM, donde podemos diferenciar entre entradas y salidas y la estructura del propio módulo con ocho diferentes submódulos.

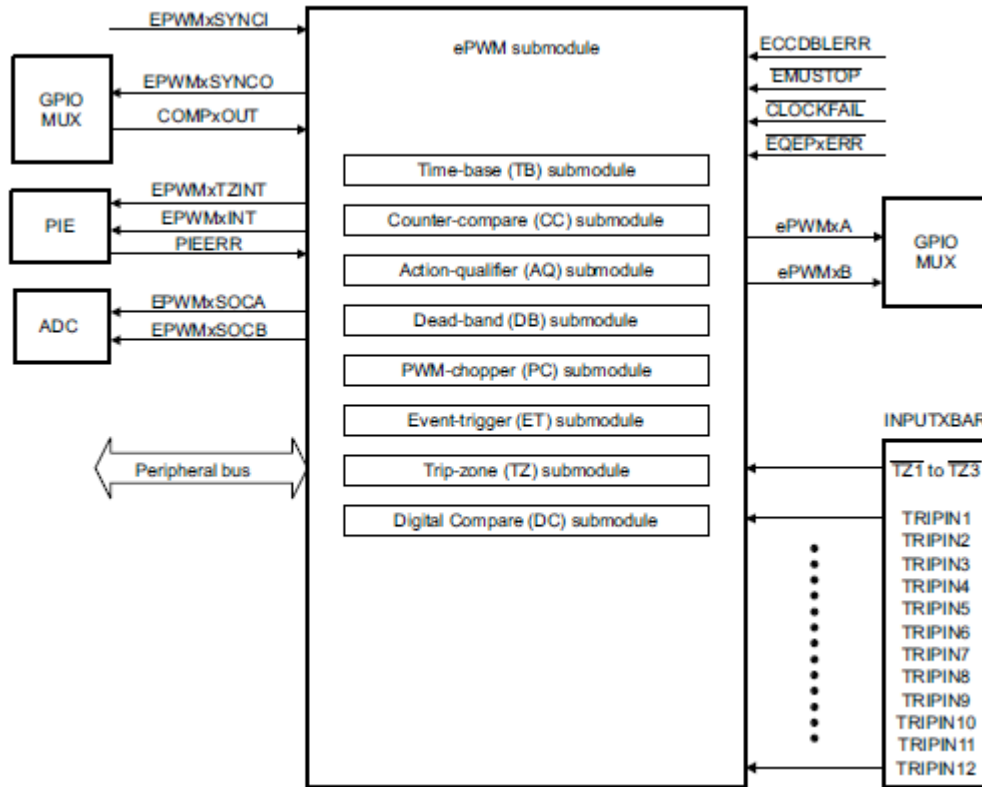


Figura 37. Esquema de módulo ePWM

2.3.2.1 Entradas y salidas de ePWM

Dentro de las diferentes salidas y entradas que más interesan dentro del módulo se encuentran las siguientes:

- Señal de salida PWM

Como ya hemos comentado, una de las peculiaridades de esta tarjeta, en cuanto al PWM, era la existencia de dos salidas que pueden utilizarse en forma de tres distintas configuraciones:

- Como dos salidas independientes con un único modo de operación.
- Como dos salidas independientes con dos modos de operación simétricos.
- Como una única salida con dos modos de operación asimétricos.

Las dos salidas se pueden hacer disponibles para el usuario a través de los periféricos GPIO (Entrada/Salida de propósito general) que se asignan en el bloque GPIO MUX

- Señales para la sincronización

Como para cualquier módulo dentro de un sistema digital síncrono es necesario que todos los procesos estén sincronizados en base a una señal de reloj. A partir de la entrada INPUTXBAR6 se puede configurar la opción de ignorar o usar la señal de reloj para ejecutar un proceso de forma asíncrona o síncrona respectivamente.

La señal de reloj externa se utiliza para sincronizar un grupo de tres módulos ePWM con esta misma señal, por lo que se conecta esta señal de reloj a través de la entrada EPWMxSYNCl del primer módulo, y este se encarga de transmitirla a los demás módulos de su grupo. A través de GPIO MUX se envía la señal de reloj de salida EPWMxSYNCO a cualquier puerto GPIO para que el usuario la pueda leer.

- Señales dirigidas al submódulo Trip Zone

Estas son las señales que van a alertar al módulo del fallo de otro componente externo. Van a ir desde TZ1 hasta TZ6. Las señales TZ1, TZ2 y TZ3 se consideran entradas asíncronas y el fallo no está condicionado por la señal de reloj si no que llega desde algún puerto GPIO y se detecta a partir del módulo X-BAR⁶. Las entradas TZ4, TZ5 y TZ6 detectan el fallo condicionado por el reloj (síncrono) desde el módulo eQEP (entrada EQEPxERR), desde el propio sistema generador de señales de reloj (entrada CLOCKFAIL), o desde la CPU (entrada EMUSTOP) respectivamente.

- Otras señales

Otras señales con algo menos de relevancia que podemos indicar son las siguientes:

- Señales para el convertidor ADC.

Dos salidas del módulo ePWM: EPWMxSOCA y EPWMxSOCB, son las que se encargan de enviar un impulso al convertidor analógico-digital para dar comienzo a una conversión.

- Señal de salida del comparador.

Salida del módulo comparador de ePWM. Este debe ser alimentado desde el módulo X-BAR por cualquiera de sus 12 entradas TRIP que serán comparadas en el submódulo "Trip Zone" dando lugar a un valor digital en la salida COMPxOUT.

- Bus periférico.

Este bus de 32 bits es el que se va a encargar de escribir en los registros de 16 y 32 bits del módulo ePWM [14].

⁶ Bloque que gestiona e identifica la dirección de las entradas o salidas de los periféricos GPIO

2.3.2.2 Estructura del módulo ePWM

Como hemos podido observar en la *Figura 37* el módulo ePWM no se compone únicamente de salidas y entradas también estructura su funcionamiento interno en 7 submódulos: time-base, counter-compare, action-qualifier, dead-band generator, PWM chopper, trip-zone, digital-compare y event trigger. En este apartado trataremos de dar una visión general de cada uno de ellos siguiendo también la estructura más detallada de la *Figura 38*.

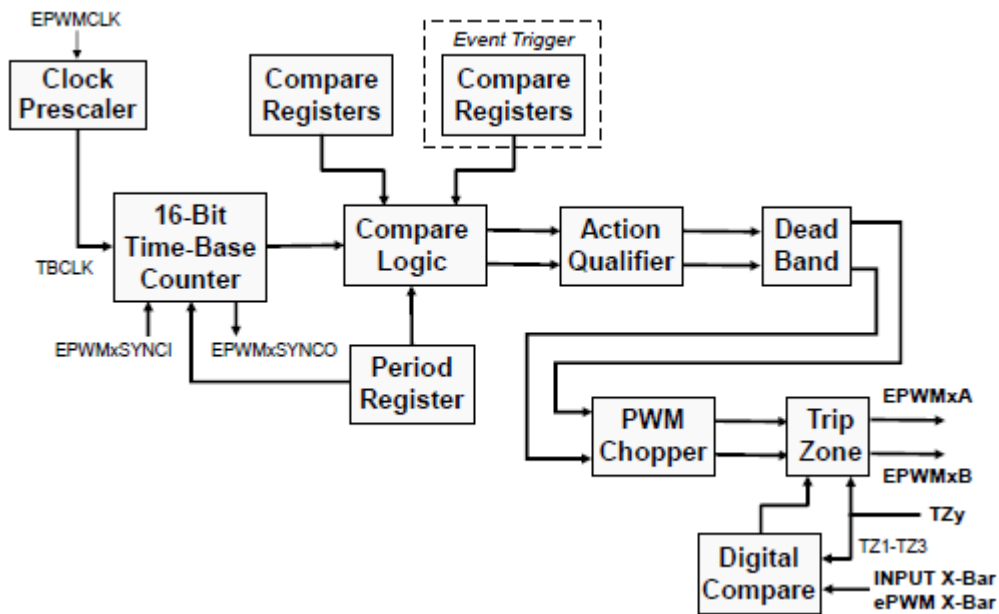


Figura 38. Diagrama de bloques del módulo ePWM

I. Time Base (TB)

El primer submódulo al que nos enfrentamos consiste en un contador de 16 bits. Su principal labor va a ser la de generar una forma de onda que luego se dirigirá al comparador. Para ello TB cuenta con dos elementos indispensables: el registro del periodo (Period Register en la *Figura 38*), y el reloj pre-escalado TBCLK a partir de la señal de reloj del ePWM EPWMCLOCK.

Con estos dos elementos el primer submódulo va a ser capaz de generar una primera forma de onda, configurando en primer lugar el número de periodos que se quieren contar en el registro de periodo, y luego el modo de funcionamiento del contador entre los tres posibles:

- Up-count. El contador empieza en cero y va creciendo hasta detenerse en el periodo señalado por el registro volviendo a reanudar el ciclo desde cero.
- Down-count. El contador comienza con el valor del periodo y decrece hasta detenerse cuando alcanza el cero. En ese momento se reinicia el contador y vuelve a decrementar desde el valor del periodo.
- Up-down-count. El contador comienza incrementado su valor desde cero y cuando llega al máximo valor señalado en el registro, el contador comienza a decrecer hasta llegar de nuevo a cero y comenzar el ciclo de nuevo.

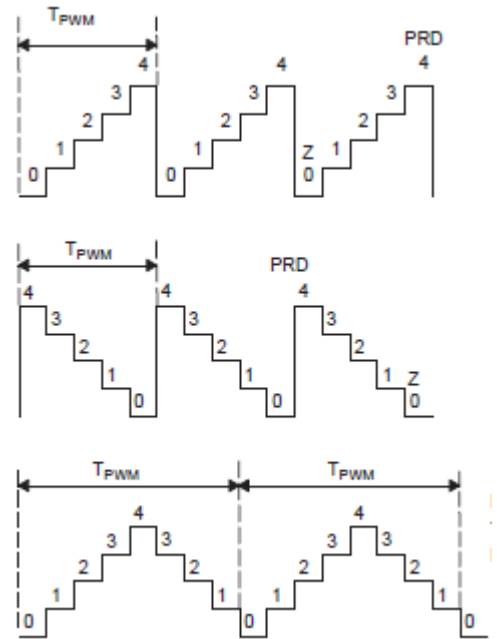


Figura 39. Formas de onda generadas en el submódulo TB

En la *Figura 39* podemos ver un ejemplo de los tres tipos de forma de onda que el contador puede generar, cuando el periodo fijado en el registro es de 4 segundos. De esta forma estamos creando la señal portadora que será la encargada de fijar la frecuencia del PWM como se explica en el apartado 2.3.1 Control PWM [13] [14].

II. Counter Compare (CC)

El propósito de este bloque es comparar la señal de salida del contador con el valor de cuatro registros comparadores: CMPA, CMPB, CMPC y CMPD. Con esto se generan cuatro eventos como señales de salida independientes que indican si el valor del contador es el mismo que el valor del registro comparador correspondiente.

Para el modo up-count y el down-count el evento solo se produce una vez por ciclo generando un pulso bastante asimétrico. Para el modo up-down-count son dos veces las que se sucede la señal creando un pulso simétrico. En la *Figura 40* y en la *Figura 41* indicamos en dos ejemplos de los pulsos que se generan cuando el contador está en modo down-count y en modo up-down-count. También nos sirve este ejemplo para indicar la labor de la entrada EPWMxSYNCl que al llegarle un pulso externo genera el valor en el contador que indica el registro TBPHS produciéndose una asimetría en la onda.

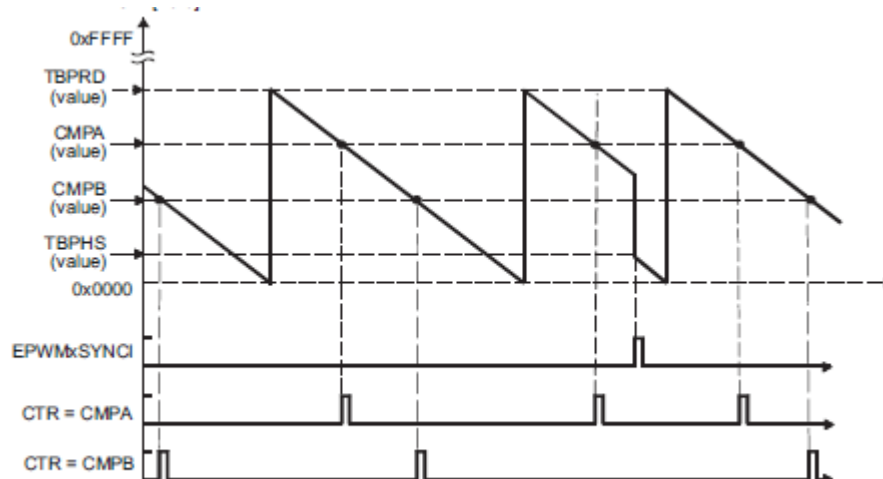


Figura 40. Generación de eventos en el submódulo CC desde modo de contador down-count

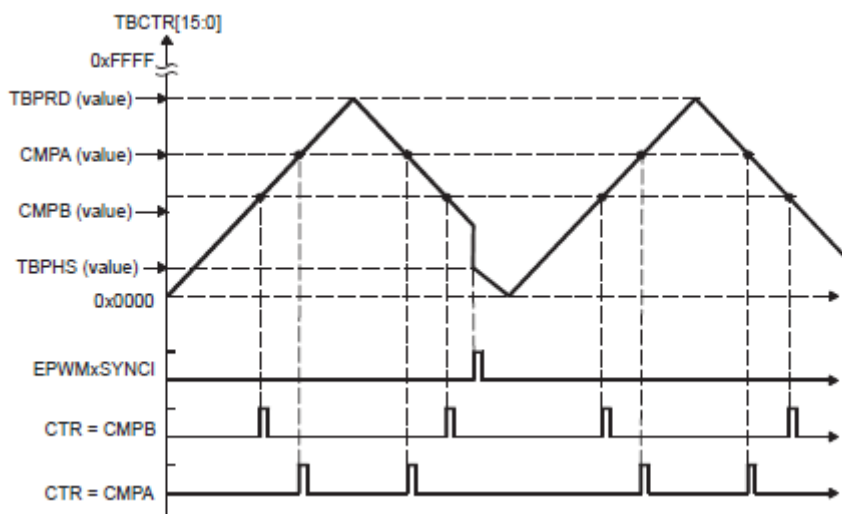


Figura 41. Generación de eventos en el submódulo CC desde modo de contador up-down-count

Viendo los ejemplos de las anteriores figuras se observa lo que antes ya hemos indicado, la simetría de aparición de los eventos cuando el contador está en modo up-down-count, y la asimetría generada en los eventos del modo down-count [13] [14].

III. Action Qualifier (AQ)

Podríamos decir que el submódulo AQ es el más importante del proceso de generación de la señal PWM, ya que va a ser el encargado de generar el primer prototipo de forma de onda de salida en EPWMxA y en EPWMxB a partir de cualquiera de los siguientes eventos:

- TBCTR=TBPRD: Contador igual al periodo máximo programado.
- TBCTR=0x000: Contador igual a cero.
- TBCTR=CMPA: Contador igual al valor del registro comparador A.
- TBCTR=CMPB: Contador igual al valor del registro comparador B.
- Aparición de eventos por entradas T1 y T2: Causados por interrupciones en el comparador, la señal de sincronismo o en el recorrido de la señal.

A partir de la aparición de estas señales, AQ va a controlar el comportamiento de las salidas calificando el valor de estas y también por la dirección del contador (si está creciendo o decreciendo).

Las acciones que se puede realizar en la salida son: **Poner nivel alto, poner nivel bajo, conmutar de nivel o no hacer nada.** Es importante indicar, antes de generar la salida PWM, la acción que le corresponde a cada valor de las distintas entradas. Para ello se va a generar siempre una tabla de eventos como la que indicamos en la *Tabla 5* a propósito del ejemplo de la *Figura 42*.

Cada salida puede funcionar independientemente de la otra, como en el caso que vamos a mostrar en la *Figura 42* donde se trata de generar dos formas de onda asimétricas a partir del modo up-count del contador.

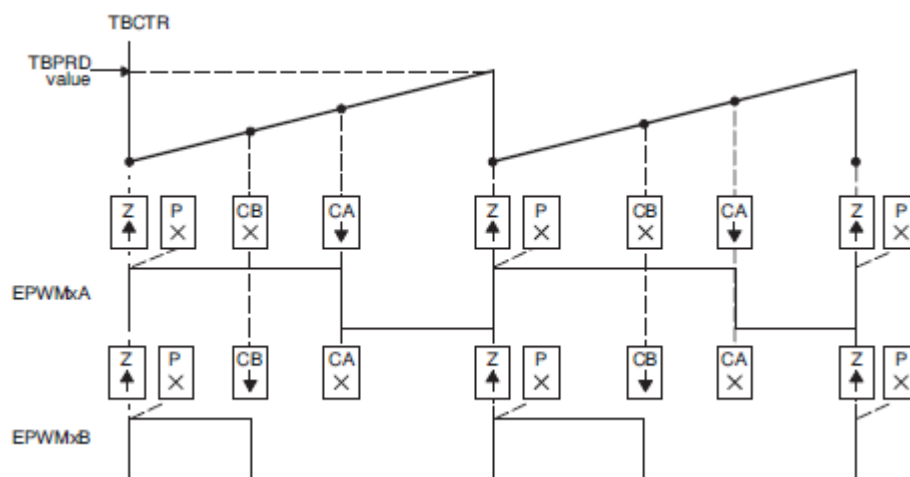


Figura 42. Ejemplo de implementación de las dos salidas PWM

Como podemos observar, en el ejemplo se han predefinido las acciones para cuando se den las condiciones que indica cada entrada. En la tabla de eventos de la *Tabla 5* señalamos estas instrucciones [13] [14].

Tabla 5. Tabla de eventos correspondiente al ejemplo de la Figura 42

Valor del contador	Salida del PWM	
	EPWMxA	EPWMxB
Igual a 0	Nivel Alto	
Igual al periodo	No hace nada	
Igual al valor del registro CMPA	Nivel Bajo	No hacer nada
Igual al valor del registro CMPB	No hacer nada	Nivel Bajo

Desde el siguiente submódulo que vamos a analizar hasta el último, se consideran algunas modificaciones a la forma de onda que obtenemos desde el bloque AQ.

IV. Dead Band Generator (DB)

Primer submódulo “modificador” de las señales PWM, que se va a encargar de gestionar los retardos, generados por la aparición de la banda muerta o dead-band⁷, que se pueden dar en la salida PWM, con el fin de no alterar la conmutación de los elementos de potencia (los más afectados por los retardos). Para ello este bloque va a poder programar retardos de caída (FED) o retardos de subida (RED), complementarlos, y añadirlos a la señal (con un complejo sistema de conmutadores) para anular todos aquellos retardos que se hayan podido generar [13] [14].

V. PWM Chopper (PC)

Este submódulo es necesario sobre todo si las salidas PWM van a ser utilizadas para controlar un elemento conmutador de potencia que necesita una frecuencia mucho mayor de pulso para funcionar. Por ello, la principal función de este bloque es el aumento de la frecuencia del pulso cuando está en nivel alto generando para este

⁷ Con dead-band nos estamos refiriendo al conjunto de valores de la salida PWM que no son ni altos ni bajos, dan como salida el valor intermedio (siempre cercano a 0), que además de generar retardos en la señal, pueden perjudicar a los elementos de potencia como son los transistores MOSFET a los que generaría un cortocircuito.

una cantidad importante de conmutaciones. En la *Figura 43* se aprecia mejor esta función tan importante de “dividir” el pulso.

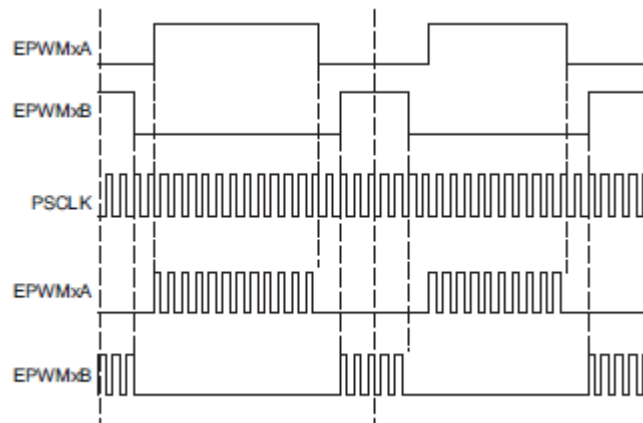


Figura 43. Señales PWM resultantes al pasar por el submódulo PC

Es importante indicar que el reloj PSCLK es una multiplicación de frecuencia del reloj inicial del sistema EPWMCLOCK, y que, además, también es posible en este bloque añadir un pulso inicial con una anchura más grande para asegurar ese nivel alto inicial [13] [14].

VI. Trip Zone (TZ)

El submódulo TZ, del que ya hemos comentado algo en el apartado 2.3.2.1 Entradas y salidas de ePWM, es el encargado de gestionar las interrupciones que se puedan dar en el sistema a través de: algún puerto GPIO, entradas TZ1-TZ3, o por algún fallo interno del sistema, entradas EQEPxERR (fallo en módulo eQEP), CLOCKFAIL (fallo en el reloj) o EMUSTOP (fallo en el procesador).

El bloque se compone de distintos registros denominados TZCTL que son programados para hacer frente a cada interrupción e indican el estado de la salida PWM cuando esta se dé. En la *Tabla 6* se muestran algunas de las posibles acciones [13] [14].

Tabla 6. Posibles valores de salida PWM según registros TZCLT

TZCTL Register bit-field Settings	EPWMxA and/or EPWMxB	Comment
0,0	High-Impedance	Tripped
0,1	Force to High State	Tripped
1,0	Force to Low State	Tripped
1,1	No Change	Do Nothing. No change is made to the output.

VII. Digital Compare (DC)

El submódulo DC se basa en generar eventos a partir de comparar señales externas al módulo ePWM que dan lugar a cambios en las señales de salida PWM. Estas señales externas provienen generalmente de algún pin GPIO, o de algún periférico interno redirigida su señal a través del bloque ePWM X-BAR. Cuando una o varias de las señales seleccionadas por el bloque cambia su valor a bajo o alto es cuando se genera el evento que se dirige a los submódulos TZ, TB o ET con el fin de poder ejecutar alguna de las siguientes acciones:

- Generar una interrupción.
- Activar el comienzo de una conversión ADC.
- Forzar un evento.
- Generar un evento de sincronización para el contador TBCTR [13] [14].

VIII. Event Trigger (ET)

Finalmente nos queda indicar el funcionamiento del ultimo bloque del módulo ePWM, que, aunque no se encuentra en la sucesión de submódulos indicada en la *Figura 38*, es fundamental para la comunicación del módulo ePWM con el resto de los elementos de la tarjeta, a través de la indicación de las interrupciones que se han generado.

Son el bloque PIE (manejo de interrupciones de la tarjeta) y el convertidor ADC los que forman esta red de comunicación del resto de la tarjeta con el submódulo ET. Este va a generar eventos que puede dar lugar a una interrupción en el sistema, o el comienzo de una conversión ADC. Estos eventos tienen su origen en los submódulos TB, CC y DC cuando suceden eventos como: contador igual a cero, contador igual al periodo máximo, contador igual a valor del registro CMPA, CMPB, CMPC o CMPD o señal desde bloque DC.

ET también tiene la opción de gestionar cada cuanto genera un nuevo evento de salida, por ello tiene la opción de generarlos por cada evento de entrada nuevo que llega, por cada dos o por cada cincuenta o más [13] [14].

2.3.3 Aplicación mediante Simulink

Una vez analizadas las entradas y la estructura del módulo ePWM llegamos al último apartado del capítulo con el fin de explicar la programación del módulo mediante la herramienta Simulink y señalar también algunos ejemplos de su uso.

Como ya hemos ido comentando, el objeto de la programación en Simulink es el de facilitar la programación, convirtiendo el código que nos podemos encontrar para la programación de cualquier sistema, en herramientas más visuales que nos aporta Simulink como son los bloques programables.

Ya hemos tratado en los apartados 2.2.3.2 CLA y 2.2.4 Comunicación. Modulo IPC, algunas de estas herramientas que añade el add-on de Texas Instrument como son el bloque CLA para generar interrupciones, o los bloques de comunicación IPC entre CPUs. En este apartado vamos a analizar la posibilidad de generar ondas PWM a través de la programación del bloque ePWM disponible para nuestro modelo de tarjeta LAUNCHXL-F28379D y que tiene la forma que se presenta en la *Figura 44*.

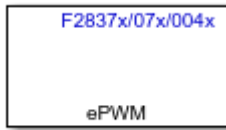


Figura 44. Bloque ePWM en Simulink

Lo primero que puede llamar la atención del bloque ePWM es la ausencia de entradas y salidas del mismo, pero hay que tener en cuenta que esto no es más que el bloque inicial, sin programar valores, por lo que podemos confirmar la presencia de entradas, si las programamos, en la pantalla de configuración del bloque que a continuación vamos a analizar a partir de la captura de la *Figura 45*.

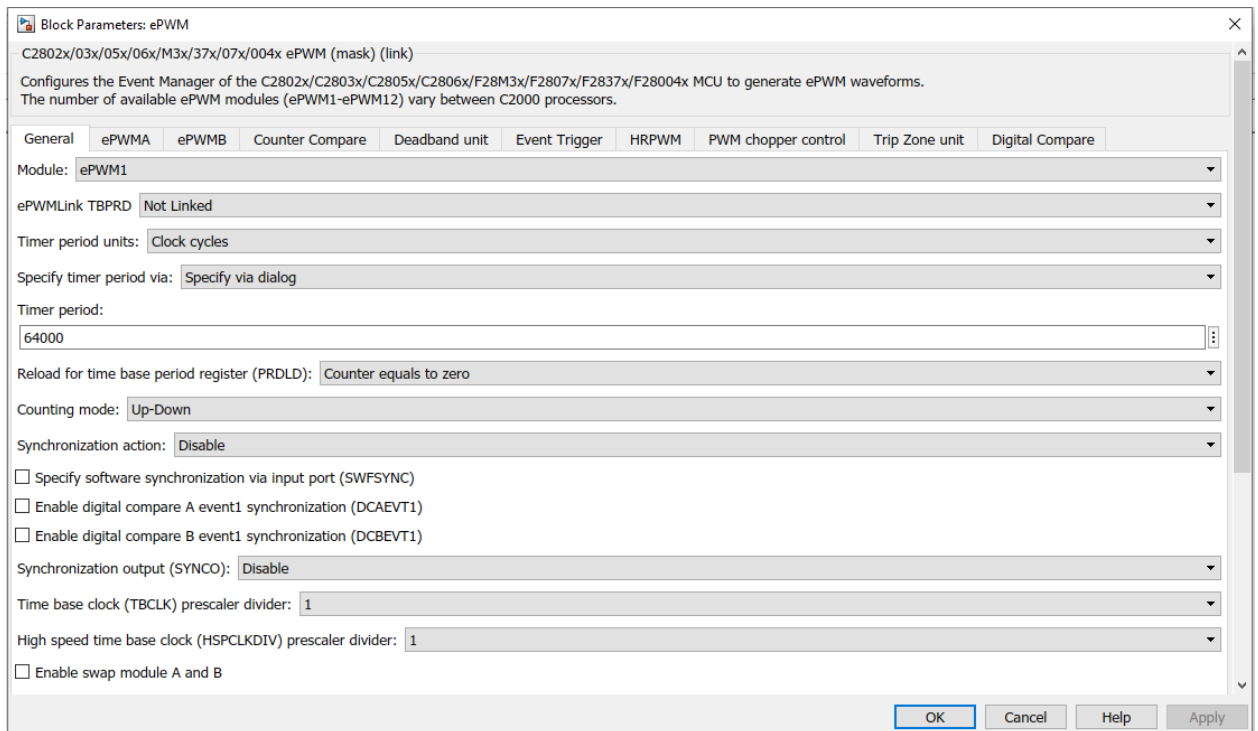


Figura 45. Pantalla general de programación del bloque ePWM en Simulink

Lo primero que se puede resaltar al observar la imagen es la presencia de un número importante de pestañas. Desde una pantalla general, hasta las distintas pestañas correspondientes con la programación de hasta seis de los ocho submódulos del PWM. Otras de las pestañas nos dan la opción de programar las salidas ePWMA y ePWMB o el bloque de salida en alta resolución HRPWM. En las siguientes líneas se tratará de dar una imagen general de la configuración de las pantallas más importantes:

2.3.3.1 Pestaña General

Pestaña en la que se seleccionan los principales parámetros del módulo ePWM haciendo especial hincapié en el contador y en el sincronismo.

◦ **Parámetro de selección de módulo**

Module: Selecciona el módulo PWM que queremos programar, existen hasta 12 opciones.

◦ **Parámetros del contador**

- *ePWMLink TBPRD:* Permite seleccionar el contador utilizado por otro de los 12 módulos. Si se selecciona la opción *Not Linked* se abre la opción de programar el contador bajo los siguientes parámetros.

- *Timer period units:* Indica la unidad de tiempo en la que se va a contar. Segundos o ciclos de reloj.

- *Specify timer period via:* Indica la vía por la que se va a fijar el periodo del contador. Esta puede ser externa (mediante una entrada) o interna mediante un cuadro de diálogo.

- *Timer period/initial period:* Indica, según se haya señalado el periodo máximo del contador (vía interna) o el periodo inicial (vía externa).

- *Reload for time base period register (PRDLD):* Indica que evento se va a utilizar para actualizar el registro de periodo con un nuevo valor. Cuando se llega a cero, o en cada instante.

- *Counting mode:* Indica el modo de uso del contador. Up-down counter, Up counter o Down counter.

◦ **Parámetros de sincronización**

- *Synchronization action:* Indica la fuente desde la cual se quiere obtener la compensación de la fase del reloj de sincronismo. Por defecto esta desactivado.

- *Specify software synchronization via input port:* Da la opción de poder lograr un correcto sincronismo entre todos los bloques del módulo ePWM.

- *Enable digital compare A/B event synchronization:* Da la opción de poder sincronizar el reloj de este módulo con el de otro cercano a través de un comparador.

- *Synchronization output:* Indica la opción de poder generar una salida del bloque a partir del bloque TB cuando se den ciertos valores en el contador.

◦ **Otros parámetros**

- *Time base clock prescaler divider*: Indica la división que queremos establecer de frecuencia del reloj del módulo en referencia a la frecuencia predeterminada del reloj de la tarjeta. Los valores son las potencias de dos hasta 128.

- *High speed time base clock prescaler divider*: Indica la división de frecuencia que se va a utilizar en el módulo de salida de alta resolución HRPWM.

- *Enable swap module A and B*: Da la opción de intercambiar el valor de las salidas ePWMA por ePWMB y viceversa.

2.3.3.2 Pestañas ePWMA y ePWMB

Pestaña para configurar la programación del submódulo AQ que a partir de la señal del contador va a dar lugar a la salida PWM.

◦ **Parámetros de control de acción PWM**

- *Enable ePWM#x*: Da la opción de activar la salida ePWM A y/o B del módulo correspondiente seleccionado en la pantalla general.

- *Action when...:* Conjunto posibles situaciones del contador (=ZERO, =PRD, =CMPA, =CMPB) para las cuales se pueden programar cuatro posibles respuestas: Set, Clear, Do nothing o Clear. Es la forma de llevar a cabo el diseño de las tablas de eventos como la explicada en la *Tabla 5*.

◦ **Parámetros de respuesta a evento externo**

- *Compare value reload condition*: Indica cuando y si se debe llevar a cabo la actualización de eventos externos. Con el contador a cero, con el contador en el periodo, en los dos casos o nunca.

- *Add continuous software force input port*: Da la opción de crear un puerto de entrada (SFA) desde el que se habiliten eventos como poner a 0 o a 1 la señal PWM.

- *Continuous software force logic*: Indica, si no se habilita la entrada SFA, cómo reacciona el módulo ante una señal externa del Software.

- *Reload condition for software force*: Indica las condiciones a las que vuelve la salida PWM después de terminar la interrupción por el evento. Puede ser a cero, al valor del periodo, a cualquiera de las dos o al valor inmediato.

2.3.3.3 Pestaña Counter Compare

Pestaña para configurar el valor de los distintos registros comparadores que actúan sobre el contador. Se repite la misma estructura para los cuatro posibles registros CMPA, CMPB, CMPC y CMPD.

- *ePWMLink CMPx*: Indica si se va a utilizar el valor del comparador programado de otro módulo (1-12) o se programa el del mismo (opción Not Linked).
- *CMPx units*: Indica las unidades que se utilizan para seleccionar el valor del registro con respecto a los del contador. Pueden ser unidades de ciclo de reloj (valores del reloj), o unidades de porcentaje de periodo realizado.
- *Specify CMPx via*: Indica la vía de llegada del valor del registro comparador, puede ser mediante dialogo (valor fijo) o mediante una referencia externa para la que se habilita una entrada.
- *CMPx value/initial value*: Ventana de dialogo en la que se indica el valor del comparador si se indica mediante diálogo, o el valor inicial si es mediante una referencia externa.
- *Reload for compare x Register*: Indica el momento en el que el periodo puede ser reseteado.

2.3.3.4 Otras pestañas

Como ya hemos comentado a partir del submódulo AQ, los bloques que lo suceden sirven para producir mejoras en la salida PWM. Son mejoras optativas por lo que las pestañas que corresponden a la programación de estos bloques son totalmente opcionales para cada salida. El bloque más programable es el TZ en el que se tienen que indicar que señales generan las interrupciones [18].

2.3.3.5 Ejemplo de implantación del módulo

Para comprobar realmente el funcionamiento de este módulo, desde Simulink, vamos a desarrollar un pequeño ejemplo que implantaremos en la tarjeta LAUNCHXL-F28379D.

Se trata de un generador de señal PWM en dos canales diferentes, pero a través del mismo módulo, estos serán ePWM1A y ePWM1B. El objetivo es comprobar que se genera la señal en los puertos GPIO0 y GPIO1, que se corresponden, a su vez, con la salida de los dos canales del primer módulo PWM de la tarjeta.

Vamos a probar la generación de la señal PWM de dos formas diferentes, a través de una señal continua, que nos indica el valor del ciclo de servicio de la señal generada, y a través de una onda senoidal. El esquema del ejemplo en Simulink es el de la *Figura 46*, donde podemos probar con una referencia constante de valor 75 y con una referencia variable en forma senoidal con una amplitud máxima de 50V y una frecuencia de 50Hz.

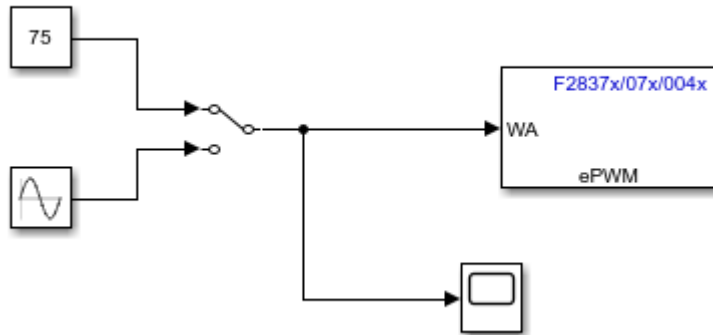


Figura 46. Ejemplo de implementación en Simulink del módulo ePWM

Los aspectos más importantes programados en el bloque ePWM son:

- ❖ El contador: En modo Up con un periodo de 5000 ciclos de reloj.
- ❖ Señal externa: Es fundamental indicar que el valor del registro CMPA proviene de una fuente externa y se va a dar en unidades de porcentaje.
- ❖ La tabla de eventos: Señal a 1 con el contador a 0 y señal a 0 cuando coincide el valor del contador con el de la señal externa.
- ❖ Módulo deadband: Módulo que es aconsejable programar cuando se obtienen dos señales de un mismo módulo. En este caso se indica aquí que serán señales complementarias las de los dos canales con una banda muerta de valor 100 ciclos de reloj.

El resultado de la implementación de este ejemplo en la tarjeta LAUNCHXL-F28379D lo obtenemos a través de un osciloscopio digital, y es el que se indica en la *Figura 47* para la referencia fija.

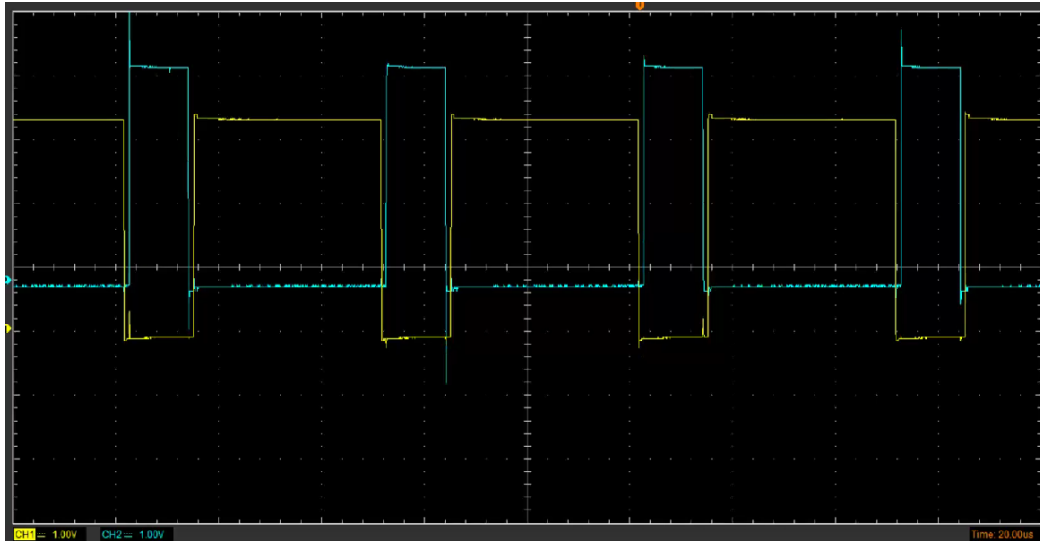


Figura 47. Señal PWM con referencia fija obtenida a partir de la tarjeta LAUNCHXL-F28379D

Como podemos observar, el ciclo de servicio se acerca al valor de 75% que habíamos programado para el canal 1 cambiando su valor a 0 cuando coincide con el del registro CMPA y volviendo a 0 cuando el contador interno se resetea. El canal 2 cumple con la función de ser el complementario, respetando las bandas muertas.

Para representar la señal PWM con referencia variable, no podemos capturar su totalidad en una imagen pues está en continuo cambio dependiendo directamente del valor de la señal senoidal, por lo tanto, hemos tomado dos capturas en diferentes instantes que se representan en la *Figura 48* y en la *Figura 49*.

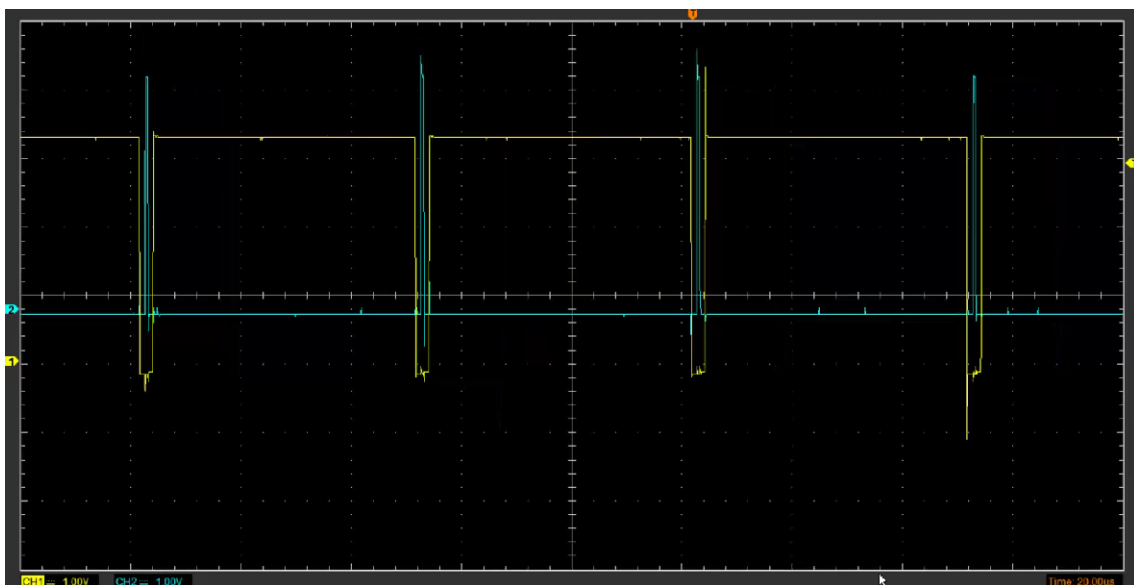


Figura 48. Señal PWM con referencia variable, obtenida a partir de la tarjeta LAUNCHXL-F28379D, cuando el ciclo de servicio se aproxima a 100%

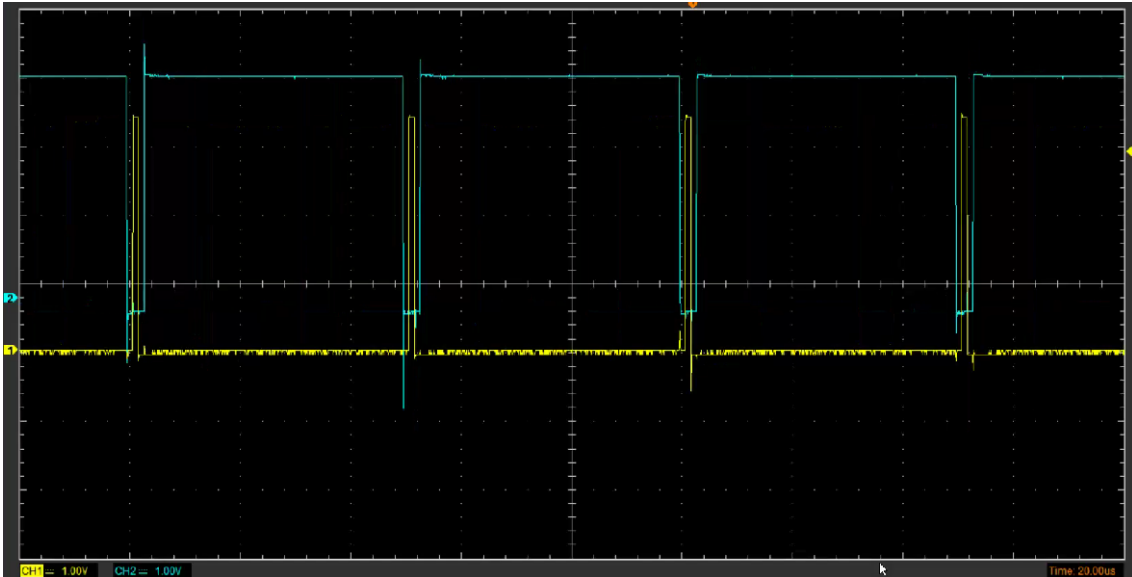


Figura 49. Señal PWM con referencia variable, obtenida a partir de la tarjeta LAUNCHXL-F28379D, cuando el ciclo de servicio se aproxima a 0%

Las dos imágenes, como podemos ver, indican dos de los estados por los que pasan las señales PWM obtenidas, y es que su valor de ciclo de servicio cambia desde 0 hasta 100 siguiendo la referencia variable de la señal senoidal que cambia en cada instante. La frecuencia de esta señal PWM, como ya hemos comentado, va a ser la misma que la señal variable (moduladora), que en este caso son 50 Hz.

Esta aplicación del módulo PWM es muy útil y sencilla de implementar en cualquier tarjeta de desarrollo, y especialmente para el diseño de un sistema de control por ancho de pulso SPWM que comentaremos en el apartado 3.2.1 Control por ancho de pulso. SPWM

2.4 CONVERTIDORES ADC Y DAC

En cualquier sistema de control es fundamental el correcto procesamiento de las señales externas, su naturaleza es comúnmente analógica, al provenir de un sensor, y la del procesador es digital, así que para poder gestionar estas señales va a ser necesario un convertidor Analógico-Digital (de ahora en adelante ADC), y cuando son estas son procesadas mediante el controlador, es necesario devolverlas a su estado analógico inicial a través del convertidor opuesto Digital-Analógico (a partir de ahora DAC). La *Figura 50* muestra un esquema de este proceso de adquisición, procesamiento, y distribución de la señal.

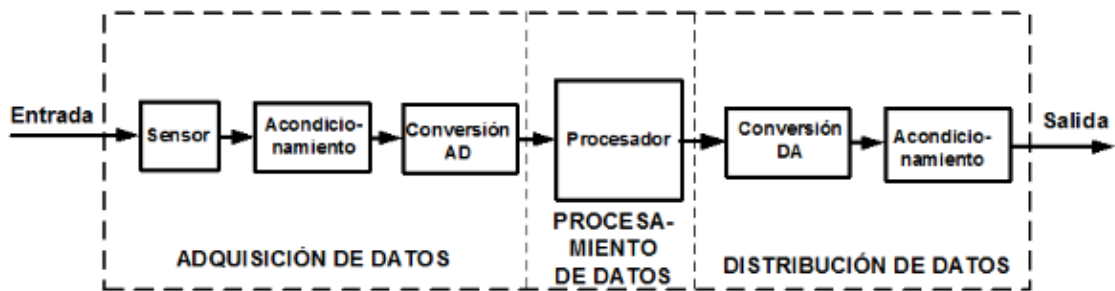


Figura 50. Esquema de procesamiento de la señal

A continuación, vamos a analizar en los siguientes apartados los módulos ADC, DAC de la tarjeta LAUNCHXL-F28379D y su programación a través de Simulink.

2.4.1 Convertidor ADC

La tarjeta LAUNCHXL-F28379D incluye cuatro módulos ADC independientes y el convertidor es de tipo SAR (Successive Approximation Register) con una resolución que se puede seleccionar entre 16 y 12 bits. El módulo está compuesto básicamente por cuatro partes diferenciadas: El **circuito de entrada**, que se encarga de muestrear la señal analógica, el **circuito convertidor**, que genera la señal digital de salida, el módulo y los registros **SOC**, que gestionan el inicio de la conversión, y el módulo **EOC**, que gestiona las interrupciones. En la *Figura 51* mostramos un esquema que señala las partes en las que se divide el módulo ADC.

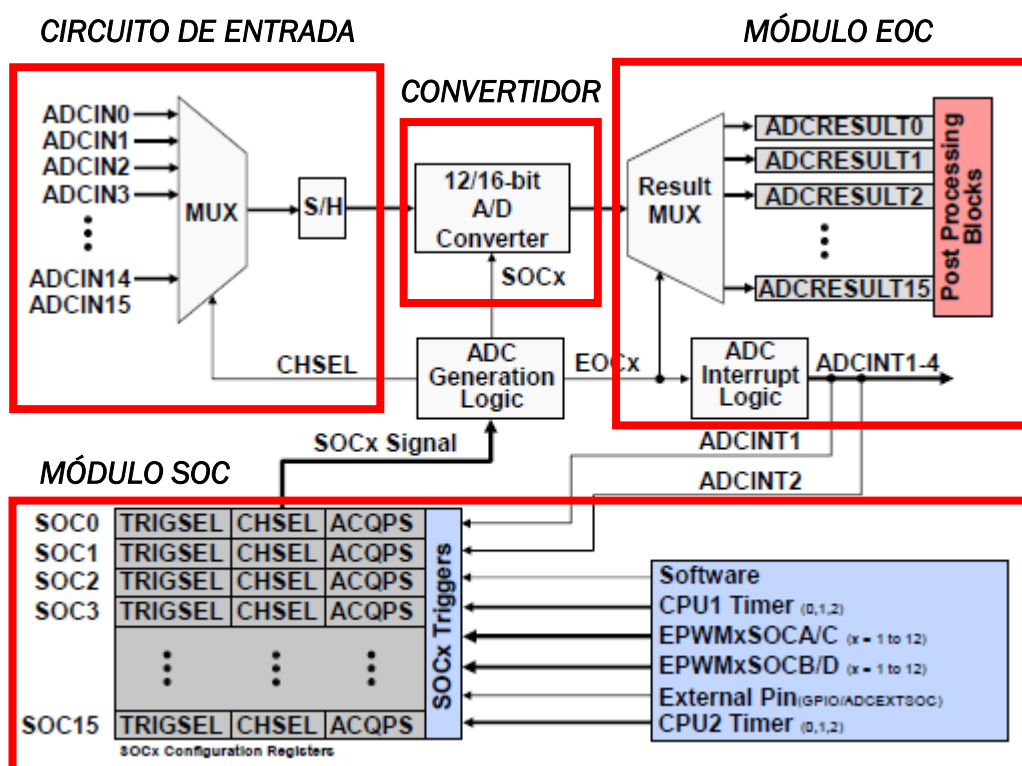


Figura 51. Diagrama de bloques del módulo ADC

En los siguientes apartados vamos a analizar brevemente el contenido de cada parte del módulo

2.4.1.1 Circuito de entrada

El circuito de entrada está compuesto por dos circuitos analógicos: El multiplexor de entrada y el circuito S/H (sample-and-hold).

- **Multiplexor:** Mediante la señal CHSEL que proviene del registro correspondiente SOC se selecciona la señal de entrada ADCIN que se va a procesar.

- **Circuito S/H:** Es el circuito que va a realizar las labores de muestreo y retención, una labor fundamental en los convertidores ADC que precisan que la señal a convertir permanezca constante durante un tiempo determinado. El circuito analógico es realmente sencillo. En la *Figura 52* podemos apreciar que se basa en un interruptor que controla el periodo de muestreo (semiconductor de potencia) y en un condensador, que mantiene la señal constante cuando el interruptor se abre. La señal de salida, también analógica podría tener la forma del ejemplo de la *Figura 53* [19].

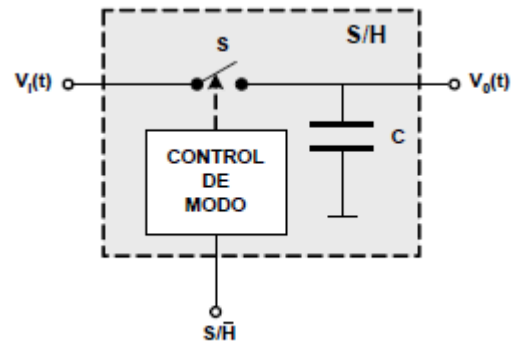


Figura 52. Circuito de muestreo y retención

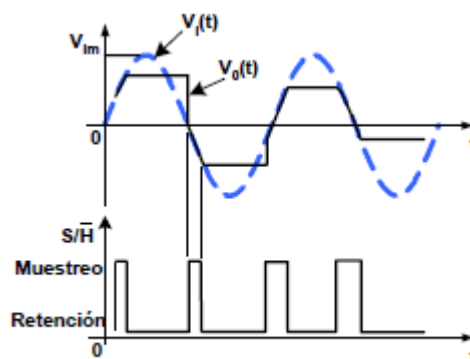


Figura 53. Salida analógica del convertidor S/H

2.4.1.2 Circuito convertidor

Parte fundamental del convertidor que se dedica transformar las señales analógicas muestreadas y retenidas, en una muestra de señales binarias que indican al procesador los valores de la señal de entrada.

La resolución del convertidor se puede seleccionar entre 12 y 16 bits, pero esta decisión acarrea dos diferencias en el funcionamiento del módulo:

■ **Rendimiento:** Cuanta más resolución tengamos en el convertidor, más espacio va a necesitar la memoria y más complejas van a ser las instrucciones para el procesador, por ello es normal que una mayor resolución conlleve una eficiencia menor que si la resolución es más baja, donde el procesador va a funcionar más rápido. Mientras que para la resolución de 16 bits el rendimiento es de 1.1 MSPS⁸ pudiendo llegar al máximo de 4.4 MSPS, para la de 12 bits el rendimiento es de 3.5 MSPS alcanzando el máximo de 14 MSPS.

■ **Modo de entrada de la señal:** La segunda diferencia al decantarnos por una resolución de 16 o de 12 bits es fundamental, pues afecta al modo de entrada de la señal. Mientras que para una resolución de 16 bits se necesitan dos señales diferenciales ADCINxP y ADCINxN para una resolución menor de 12 bits solo es necesaria una señal de entrada ADCINx.

La ventaja principal del convertidor de 16 bits es que la señal de entrada al convertidor es la diferencia entre las señales diferenciales, por lo que también se van a eliminar ruidos que puedan interferir, su principal inconveniente es la complejidad que deriva en un rendimiento más lento. Sin embargo, una resolución de 12 bits ahorra en complejidad y la convierte en más eficiente, pero la aparición de ruidos añadido a la reducción de resolución puede producir en una señal digital menos exacta.

El resultado de la conversión va a depender directamente de la resolución y es que con 16 bits podremos alcanzar un valor máximo de 2^{16} o lo que es lo mismo 65535 unidades. Con 12 bits el valor máximo se reduce hasta 4095. La *Tabla 7* y la *Tabla 8* nos van a servir para interpretar los resultados digitales de la conversión:

Tabla 7. Interpretación de salida digital para resolución de 12 bits.

	Analog Input	Digital Result
Single-Ended	when $ADCINy \leq VREFLO$	$ADCRESULTx = 0$
	when $VREFLO < ADCINy < VREFHI$	$ADCRESULTx = 4096 \left(\frac{ADCINy - VREFLO}{VREFHI - VREFLO} \right)$
	when $ADCINy \geq VREFHI$	$ADCRESULTx = 4095$
Differential	Invalid Mode	Invalid Mode

⁸ MSPS: Millones de instrucciones por segundo

Tabla 8. Interpretación de salida digital para resolución de 16 bits.

	Analog Input	Digital Result
Single-Ended	Invalid Mode	Invalid Mode
Differential	when $ADCINyP - ADCINyN \leq -VREFHI$	$ADCRESULTx = 0$
	when $-VREFHI < ADCINyP - ADCINyN \leq VREFHI$	$ADCRESULTx = 65536 \left(\frac{ADCINyP - ADCINyN + VREFHI}{2 VREFHI} \right)$
	when $ADCINyP - ADCINyN \geq VREFHI$	$ADCRESULTx = 65535$

Como vemos es fundamental para la lectura de los datos, tener en cuenta los valores de referencia de tensión de Vrefhi y de Vreflo.

La referencia de tensión Vreflo coincide normalmente con Vssa, cercana a 0 V, mientras que para Vrefhi el fabricante recomienda un valor entre 2,4V y Vdda⁹ (3,3V).

2.4.1.3 Módulo Start of conversion (SOC)

El inicio de la conversión está directamente dirigido por una serie de 16 registros que componen el módulo SOC y cada registro SOCx está configurado para especificar:

- TRIGSEL: La naturaleza del disparo que da la orden de comenzar la conversión. Son cuatro los posibles orígenes: Pines externos, módulo ePWM, software o desde alguna de las dos CPU.
- CHSEL: Canal seleccionado para convertir (ADCINx).
- ACQPS: Duración del muestreo de la señal, basada en la cantidad de tiempo que el condensador puede mantener la señal constante sin salir de un error indicado (0.5LSB).

2.4.1.4 Módulo End of conversion (EOC)

Por último, no es menos reseñable la labor del módulo EOC, el cual se encarga de crear las interrupciones generadas por el convertidor. En total son cuatro las señales que dispone desde ADCINT1 hasta ADCINT4 y estas se pueden dar o bien al inicio de la conversión o un ciclo de reloj después de obtener el resultado de la misma. Los canales ADCINT1 y ADCINT2 pueden ser también la fuente de disparo que indique el inicio de una nueva conversión [7][13][14].

2.4.2 Convertidor DAC

El otro convertidor que vamos a estudiar es el que se encarga de transformar un código en formato digital a analógico. En este caso, son cuatro los convertidores que se encuentran en la tarjeta LAUNCHXL-F28379D, se trata de convertidores independientes con una resolución fija de 12 bits.

⁹ Véase el apartado 2.1.1 Tensiones internas para más información

Cada módulo DAC está compuesto también por un buffer de salida analógico que controla la tensión de salida DACx. Si se deshabilita el buffer, una resistencia pull-down se encargará de fijar esta tensión en el pin correspondiente.

Otra característica importante del convertidor es la posibilidad de sincronizar la conexión con eventos PWM a través de una de las señales EPWMxSYNCPER que se pueden seleccionar con el registro SYNCSEL. En la *Figura 54* podemos ver el diagrama de bloques del módulo.

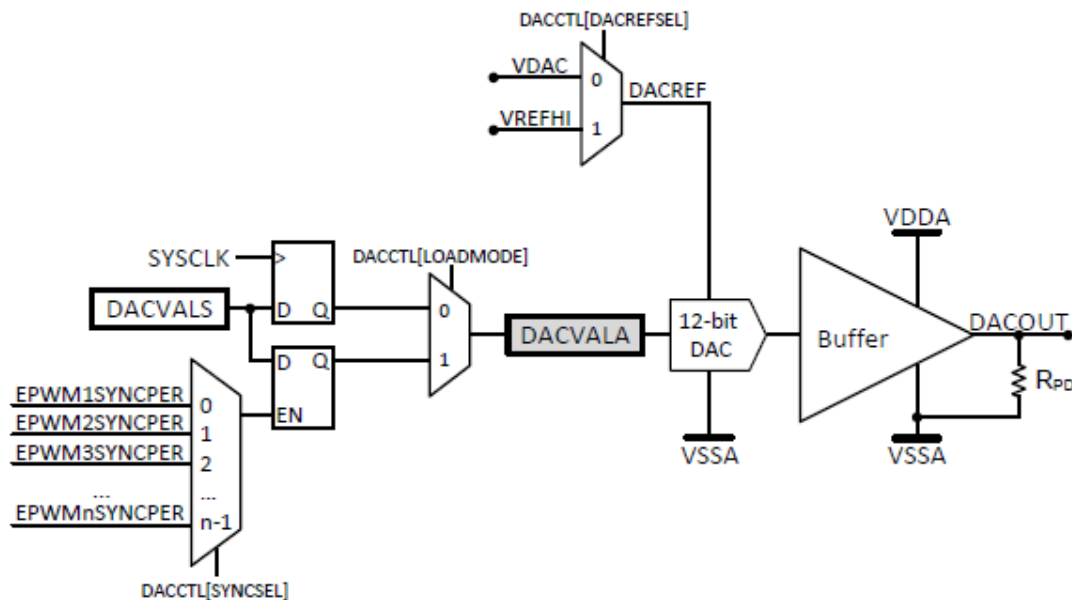


Figura 54. Diagrama de bloques del módulo DAC

Como se puede apreciar, otra parte fundamental del convertidor es la referencia de tensión DACREF que se puede seleccionar a través del registro DACREFSEL entre Vdac y Vrefhi.

Los dos registros DACVAL tienen una labor muy importante también. DACVALS es el registro solo de escritura que desemboca en el registro DACVALA de solo lectura¹⁰ que controla a su vez el valor digital que pasa al convertidor DAC. [7][14]

El valor ideal de tensión de salida puede ser calculado mediante la siguiente ecuación:

2.4.3 Aplicación mediante Simulink

Como hemos hecho en el caso del módulo ePWM, para terminar el apartado de convertidores, vamos a analizar su programación mediante Simulink, la novedosa interfaz de programación que queremos emplear en este trabajo.

¹⁰ Solo si se selecciona el modo de escritura inmediato y no mediante un evento de sincronización PWM, mediante el registro LOADMODE

Para comenzar analizaremos el bloque ADC que tiene la forma que se indica en la *Figura 55*. Es un bloque configurable de una sola salida en el que podemos programar los siguientes parámetros relacionados con el bloque SOC y EOC:

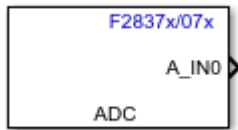


Figura 55. Bloque ADC en Simulink

- ▶ **Módulo ADC:** Elegir entre el módulo A, B, C o el D.
- ▶ **Resolución del módulo:** 12 o 16 bits (única entrada o entrada diferencial).
- ▶ **Número de registro de disparo SOC:** Desde SOC0 hasta SOC15.

▶ **Periodo de adquisición de la señal:** Valor dado en número de ciclos del reloj del sistema SYSCLK.

▶ **Fuente del disparo en SOCx:** Puede ser desde el software, desde el módulo PWM, desde la CPU, o desde un pin externo (módulo X-BAR).

▶ **Posibilidad de disparo desde interrupciones del propio convertidor:** Desde ADCINT1, ADCINT2, o desde ninguna.

▶ **Periodo de muestreo de la señal:** Intervalo de tiempo entre muestras dado en segundos.

▶ **Tipo de datos de la salida:** Datos en forma de double, single, int8, uint8, int16, uint16, int32, o uint32.

▶ **Interrupciones basadas en bloque EOC:** Posibilidad de crear una interrupción tras la conversión a través de cualquiera de los cuatro canales habilitados para ello: desde ADCINT1 hasta ADCINT4.

En la *Figura 56* indicamos la ventana de configuración donde se pueden seleccionar los parámetros que hemos indicado anteriormente.

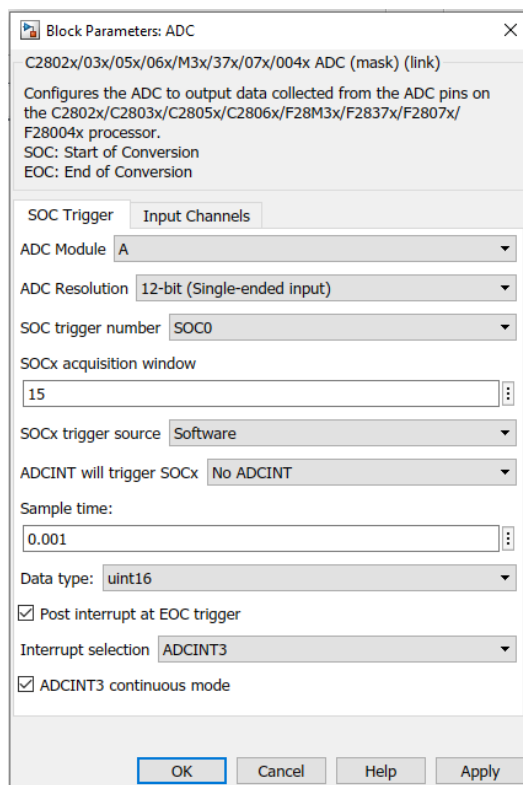


Figura 56. Ventana de configuración del bloque ADC en Simulink

Como se puede apreciar también existe una ventana de configuración de los canales de entrada “Input Channels”, aquí se selecciona que canal desde ADCIN0 hasta ADCIN15 queremos que se encargue de importar la señal analógica. Si la resolución es de 16 bits de deben seleccionar dos canales contiguos, 8 posibilidades [18].

Tras analizar los parámetros programables del convertidor ADC, pasamos ahora a analizar el bloque DAC que como vemos en la Figura 57 posee solo una entrada digital para convertirla en analógica.



Figura 57. Bloque DAC en Simulink

Solo hay dos parámetros seleccionables en este bloque. El canal de conversión (A, B o C), y la posibilidad de saturar la salida si se alcanza un valor superior a 4095 (2^{12}).

Pero además no solo son importantes los parámetros, el fabricante también indica que las salidas DACA, DACB O DACC no se pueden usar si están siendo utilizadas por el convertidor ADC las entradas ADCINA0, ADCINA1 o ADCINB1 respectivamente, pues en la tarjeta se corresponden con los mismos pines. También se indica que el tipo de valor de entrada al bloque puede ser double, float, int, o uint [18].

Para conocer cómo funciona el bloque ADC en una simulación dedicamos unas líneas para explicar su implementación el apartado 4. Desarrollo práctico del TFG.

2.5 OTROS BLOQUES EN SIMULINK

Tras analizar a lo largo de todo este apartado algunos de los módulos más importantes de la tarjeta LAUNCHXL-F28379D, y su implementación en Simulink, vamos a terminar dedicándole este punto al resto de bloques programables que nos aporta la interfaz de programación y que también se incluyen físicamente en la tarjeta. Estos son: las salidas y entradas digitales GPIO, las interfaces de comunicación SPI, UART, CAN e I2C y los módulos eQEP y eCAP.

2.5.1 Salidas y entradas digitales GPIO

Las salidas y entradas de propósito general (GPIO), son una parte fundamental para el desarrollo de cualquier aplicación. Son todos aquellos pines incluidos en la tarjeta que pueden recibir o transmitir información hacia elementos externos controlables.

Para que estos pines se identifiquen como entrada o como salida, es crucial la labor del módulo X-BAR, ya nombrado en este proyecto, y es que este bloque va a ser el encargado de gestionar y direccionar todos y cada uno de los puertos de propósito general.

Este módulo este compuesto por un bloque **Input X-BAR** para procesar las señales de entrada, un bloque **Output X-BAR** que identifica las señales de salida y las asigna un puerto GPIO, y un bloque **ePWM X-BAR** que gestiona las señales digitales que van dirigidas al módulo ePWM.

Para programar desde Simulink el módulo X-BAR, e indicarle cuales son los puertos que queremos tratar como salidas y cuales, como entradas, su interfaz nos aporta los siguientes bloques programables que podemos ver en la *Figura 58*.



Figura 58. Bloques de programación de entradas y salidas digitales en Simulink

A través del bloque **GPIO DI** podemos seleccionar que puerto GPIO queremos leer indicando al módulo X-BAR que se trata de una entrada digital. A través de Simulink podríamos programar el procesado de esta señal posteriormente.

El bloque **GPIO DO** va a ser útil para programar un puerto GPIO como salida digital. Con esto, podríamos mandar las señales procesadas, de nuevo, a un GPIO de salida de la tarjeta LAUNCHXL-F28379D.

El gran problema de estos bloques programables es que solo vamos a ser capaces de leer y enviar a la tarjeta señales digitales. Para procesar una señal analógica tendremos que hacer uso de los bloques ADC y DAC que ya hemos analizado en el apartado 2.4 CONVERTIDORES ADC Y DAC.

2.5.2 Interfaces de comunicación

Como en cualquier microcontrolador, la transmisión de datos y la comunicación con elementos externos, para adquirirlos, es fundamental. Para ello existen diferentes interfaces y protocolos de comunicación programables que se incluyen físicamente en la tarjeta. La interfaz más conocida de comunicación es la USB (universal serial bus) que debido a sus características es muy utilizada en transmisión de datos entre equipos de cualquier clase.

En la tarjeta LAUNCHXL-F28379D se incorpora también un puerto USB, pero en este caso nos interesa centrarnos más en las interfaces programables de Simulink como son la SPI y la UART para comunicaciones digitales, y el bus CAN como alternativa programable al puerto USB. La tarjeta también aporta el módulo I2C para gestionar la comunicación entre elementos internos de la misma.

2.5.2.1 Interfaz SPI

El bus SPI (Serial Peripheral Interface) es una interfaz de comunicación para transferencia síncrona en serie de bits.

Se compone de cuatro líneas de transmisión que se corresponden con: el reloj (SPICLK), el dato entrante (SPISIMO), el dato saliente (SPISIMI), y la habilitación (SPISTE). Puede actuar como esclavo recibiendo información, o como maestro enviándola.

La tarjeta LAUNCHXL-F28379D incluye cuatro módulos SPI, soporta un máximo de 16 bits por envío, y recibe y transmite utilizando colas FIFO (primero en entrar, primero en salir).

En cuanto a su programación, Simulink ofrece los tres tipos de bloques que vemos en la *Figura 59*.

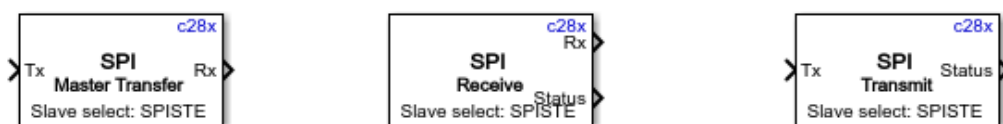


Figura 59. Bloques de programación de la interfaz de comunicación SPI en Simulink

El bloque **SPI Master Transfer** se encarga de programar la transferencia de datos desde un servicio maestro hacia un servicio esclavo, el bloque **SPI Receive** lee los datos que recibe desde otro servicio, a través del bus SPI, para procesarlos en el programa, y el bloque **SPI Transmit** envía los datos procesados en dirección al bus SPI.

En cada bloque se pueden programar los mismos parámetros, entre ellos: el módulo SPI (A, B, C o D), el tamaño de bits por mensaje, o la polaridad del reloj.

La principal aplicación de este servicio es para la conexión del microcontrolador con periféricos externos o con otros controladores con esta tecnología, ya que destaca en su velocidad de transmisión, su fácil implementación y su flexibilidad [14].

2.5.2.2 Interfaz UART

El bus de comunicación UART (Universal Asynchronous Receiver-Transmitter) o también denominado SCI (Serial Communication Interface), se trata de una interfaz de comunicación para transferencia asíncrona en serie de bits.

La diferencia con el bus SPI es que no necesita el reloj para recibir o enviar datos, aunque siendo una interfaz serie, como la interfaz SPI, no es necesario más que un cable para la transmisión desde cualquiera de los cuatro módulos que posee.

Al no contar con un reloj, la interfaz UART cuenta solo con dos canales de transmisión: el de dato entrante (SCIRxD) y el de dato saliente (SCITxD). Al ser una interfaz asíncrona tampoco hay clasificación de módulos como maestro o como esclavo.

Para su desarrollo en Simulink, la herramienta nos ofrece los dos bloques que mostramos en la *Figura 60*.



Figura 60. Bloques de programación de la interfaz de comunicación UART en Simulink

El bloque **SCI RCV** se encarga de procesar la señal que recibe desde el puerto SCI para que se pueda representar en Simulink, y el bloque **SCI XMT** tiene la función de enviar cualquier dato procesado en la herramienta hacia el puerto SCI de salida de la tarjeta LAUNCHXL-F28379D.

La principal aplicación de este servicio es para la conexión del microcontrolador con el ordenador. Entre sus ventajas se encuentra las grandes distancias que puede recorrer la transmisión, aunque a costa de esto pierda velocidad [14].

2.5.2.3 Bus CAN

El bus CAN (Controller Area Network) es una interfaz de comunicación para transferencia de mensajes en entornos distribuidos, como puede ser un control en tiempo real.

Este bus se diferencia de los dos anteriores al desarrollar un innovador protocolo de comunicación denominado también CAN. A partir de su implementación se pueden alcanzar velocidades de transmisión de hasta 1Mbps.

Su módulo de interfaz tiene acceso directo a la CPU a través de la memoria RAM, por ello, en la tarjeta LAUNCHXL-F28379D, los pines CAN no comparten espacio con los pines GPIO. La implementación de este protocolo exige esta independencia.

Son dos los canales necesarios (más la toma a tierra) para poner en marcha la transmisión a través de este bus: El canal CAN-H y el canal CAN-L. La transmisión se lleva a cabo cuando existe una diferencia de potencial entre ellos. El bloque CAN-transceiver que incorpora la tarjeta va a ser el encargado de descifrar el contenido del mensaje.

Para la programación en Simulink de este novedoso protocolo se nos aportan los dos tipos de bloques que podemos ver en la *Figura 61*.



Figura 61. Bloques de programación del bus de transmisión eCAN en Simulink.

El bloque **eCAN RCV**, se va a encargar de recibir los mensajes desde los pines del bus y con ello va a emitir la llamada a una función que incorpora, así como los datos de salida en el formato que se haya programado.

El bloque **eCAN XMT** transmite el mensaje que se ha procesado y lo devuelve al bus CAN.

Este tipo de bus es ampliamente conocido en aplicaciones industriales como puede ser la automatización de una planta o el control de un robot. Entre sus ventajas destacan la inmunidad a interferencias, la reducción de cableado y la habilidad de autodiagnóstico y reparación de errores [7] [14].

2.5.2.4 Módulo I2C

La última interfaz de comunicación que vamos a analizar es la del módulo I2C. La función llevada a cabo por este módulo es fundamental para la comunicación interna entre las diferentes partes de la tarjeta, y no es otra que la de proveer a la tarjeta una interfaz de comunicación con los periféricos.

Su diseño se basa en la forma de un bus maestro-esclavo, donde el módulo enlaza el puerto que quiere transferir y que actuará como maestro y el puerto que debe reaccionar que actuará como esclavo.

Esta labor se lleva a cabo a partir de dos líneas de comunicación: La SDA (Serial Data) y la SCL (Serial Clock). El canal SDA se encarga de transferir los datos y el canal SCL de aportar el reloj para sincronizar la transmisión.

En Simulink se puede programar este módulo a través de los dos bloques de la *Figura 62*



Figura 62. Bloques de programación del módulo I2C en Simulink.

Como en anteriores ocasiones, en este caso, también es el bloque **I2C RCV** el encargado de leer los datos que transmite el módulo I2C y el bloque **I2C XMT** el responsable de devolver la información procesada al canal de transmisión.

Entre las ventajas que supone contar con este módulo en la tarjeta están el bajo coste y la sencillez de implementación, entre sus inconvenientes se encuentran su lentitud y su sensibilidad a interferencias [14].

2.5.3 Módulos eQEP y eCAP

Para terminar de analizar los bloques programables en Simulink vamos a ver en este apartado las características y programación de dos módulos característicos de la tarjeta LAUNCHXL-F28379D: Los módulos eCAP y eQEP.

2.5.3.1 Módulo eCAP

Este módulo tiene la función de ofrecer captura mejorada de datos (Enhanced Capture) a la entrada de señales externas. Es preciso cuando se requiere una exacta sincronización de eventos con la tarjeta.

El módulo está compuesto por diferentes bloques independientes por los que pasa la señal capturada para mejorar su resolución. Estos son alguno de ellos: pin de

captura, contador de 32 bits, cuatro registros de 32 bits para capturar los datos recogidos, control de estos registros...

Las entradas a este módulo se pueden conectar a través de cualquier puerto de propósito general (GPIO). Cuando no se utiliza este modo de captura (eCAP), el módulo puede utilizarse como un canal de salida PWM en su modo APWM.

En Simulink podemos programar el módulo a partir del bloque eCAP de la *Figura 63*. A partir de este bloque se puede obtener la señal procesada y optimizada por el mismo, señalando también porque pin se obtiene la señal, si se va a utilizar en modo APWM, o que tipo de control se ejerce sobre los datos (continuo o único).



Figura 63. Bloque de programación del módulo eCAP en Simulink

Algunas de las aplicaciones que se le pueden dar a este módulo tienen que ver con la velocidad de lectura de sensores para obtener información ampliada de estos valores [7] [14].

2.5.3.2 Módulo eQEP

El otro módulo característico que se puede programar a través de Simulink es el que se encarga de interactuar directamente con los encoders (direccionales o rotatorios) que obtienen la posición, la dirección y la velocidad del elemento que se esté controlando.

Para todo esto, el bloque cuenta con cinco bloques funcionales para procesar la señal:

- Quadrature Capture Unit (QCAP): Para capturar la dirección y la rotación de medidas obtenidas a baja velocidad.
- Position Counter/Control Unit (PCCU): Para obtener la posición del elemento.
- Quadrature Decoder Unit (QDU): Para decodificar la dirección y la rotación de las medidas obtenidas a través del encoder.
- Unit Time Base for speed and frequency measurement (UTIME): Para fijar las unidades de tiempo que va a medir la velocidad y la frecuencia.
- Watchdog timer for detecting stalls (QWDOG): Contador para interrumpir la media del sensor si se da algún evento externo.

El bloque eQEP en Simulink es el que utilizaremos para gestionar este módulo. Como se puede ver en la *Figura 64*, este contiene una salida para representar la posición, la velocidad y la dirección del elemento capturado por los pines del bloque QEP.



Figura 64. Bloque de programación del módulo eQEP en Simulink

Es posible también, configurar algunos parámetros del módulo QEP a través del menú interno del bloque, donde se puede editar en cada pestaña los diferentes parámetros de los bloques funcionales que hemos ido comentado.

La aplicación principal para la que está diseñada este módulo, en la tarjeta LAUNCHXL-F28379D, es el control sobre sistemas robotizados, donde los sensores basados en encoders son cada vez más utilizados y es necesario procesar estas señales, para ejecutar algunos de los movimientos más precisos del robot [7] [14].

3. Aplicación en electrónica de potencia

Como comentamos en la introducción, este trabajo no solo tiene el fin de presentar la tarjeta LAUNCHXL-F28379D y su desarrollo en la interfaz de Simulink, si no que una parte fundamental es la de conocer el uso que se le puede dar en aplicaciones como el control de dispositivos de electrónica de potencia.

Vamos a utilizar este apartado fundamentalmente para introducir las nociones básicas de la electrónica de potencia, con el fin de llegar a conocer y analizar el funcionamiento y control de un inversor para su posible implementación a través de la tarjeta LAUNCHXL-F28379D.

3.1 ELECTRÓNICA DE POTENCIA. EL INVERSOR

Con electrónica de potencia nos referimos principalmente a todos aquellos dispositivos electrónicos controlables que son capaces de integrarse en una red de media-alta tensión. De esta manera es posible controlar algunos valores eléctricos como son la tensión, la corriente, o la potencia entregada a cualquier dispositivo conectado normalmente a una red de alimentación externa.

Por todo esto una de las principales aplicaciones de la electrónica de potencia va a ser la transformación de la energía, entre valores de corriente continua, entre valores de corriente alterna o entre valores de continua y alterna o viceversa. Estos van a ser los convertidores de potencia que veremos posteriormente. El control de cualquier convertidor se lleva a cabo mediante los semiconductores de potencia.

3.1.1 Semiconductores de potencia

Para el control de una gran cantidad de potencia en un circuito electrónico necesitamos elementos seguros y eficaces que permitan manejar valores normalmente de gran magnitud. Estos son los denominados semiconductores de potencia.

Los semiconductores van a funcionar en dos estados, ON y OFF y el tipo de control usado para este cambio va a permitir realizar una primera clasificación, entre semiconductores controlados por corriente (tiristor o GTO¹¹) o por tensión (transistor MOSFET).

Otra clasificación que se puede dar es por la frecuencia de conmutación máxima permitida, donde los transistores MOSFET pueden funcionar en un máximo de hasta 100kHz y son los tiristores los que menos frecuencia aceptan.

Por último, se pueden clasificar los semiconductores por potencia máxima permitida, aquí son los tiristores los que soportan hasta 10^4 kVA de potencia, y los transistores los que menos potencia permiten controlar.

¹¹ Gate turn-off. Tiristor de apagado por puerta

Como no son ideales, es muy importante también conocer cuáles son las pérdidas de potencia que se pueden dar en cada uno de ellos. Estas pueden ser principalmente dos:

- *Perdidas por conducción*

Perdidas de potencia debidas a la caída de potencial cuando conduce la corriente durante t_{cond} . La forma de obtenerlas es la siguiente:

$$P_{COND} = V * I * \frac{t_{cond}}{T} \quad (4)$$


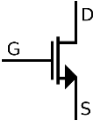
- *Perdidas por conmutación*

Perdidas de potencia debidas a la conmutación entre dos estados de tensión, influidas por la frecuencia de esta y la energía transmitida tanto para el cambio a ON como para OFF, como se indica en la siguiente ecuación:

$$P_{CONM} = (E_{ON} + E_{OFF}) * f \quad (5)$$

En el caso de los tiristores, las pérdidas en conducción son muy pequeñas, al contrario que las pérdidas en conmutación, que debido a su lentitud se vuelven elevadas. Para los MOSFET, existe gran disipación de potencia en el estado de conducción, y poca en el estado de conmutación, pues la velocidad de conmutación es mucho más alta. En la *Tabla 9* hemos hecho un resumen de las características más importantes de estos dos semiconductores.

Tabla 9. Tabla descriptiva de tiristores y MOSFET

<u>TIRISTORES</u>	<u>MOSFET</u>
	
<ul style="list-style-type: none"> • <i>Alta potencia</i> • <i>Baja frecuencia</i> 	<ul style="list-style-type: none"> • <i>Baja potencia</i> • <i>Alta frecuencia</i>
<ul style="list-style-type: none"> • <i>Perdidas de potencia en conmutación</i> 	<ul style="list-style-type: none"> • <i>Perdidas de potencia en conducción</i>

Conocidas las diferencias entre tiristores y transistores MOSFET, podemos confirmar que ninguno de los dos es ideal, siempre va a haber pérdidas de potencia, y las aplicaciones no pueden ser las mismas para los dos.

Con el fin de englobar todos los rasgos de los semiconductores y minimizar las pérdidas, ha aparecido recientemente un nuevo tipo de semiconductor de potencia, el transistor IGBT, que posee unas características intermedias (sin trabajar a muy

altas frecuencias ni potencias) que lo postulan como el más utilizado en los convertidores.

El IGBT une en un solo encapsulado las peculiaridades de los transistores bipolares BJT y las de los unipolares MOSFET. Como los BJT poseen (como los tiristores) menores pérdidas en conducción se utilizará su estructura de colector-emisor para la etapa de transmisión de potencia, mientras que los MOSFET que tienen menos pérdidas en conmutación aportan su estructura de puerta que se utilizará para el control. La *Figura 65* nos permite visualizar el esquema de lo que sería la estructura interna de un IGBT.

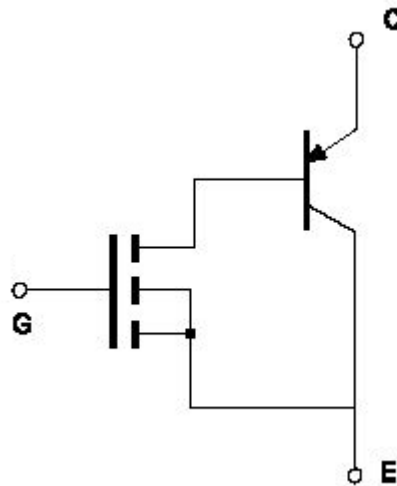


Figura 65. Estructura de un IGBT

Como resultado obtenemos un semiconductor eficiente, sin demasiadas pérdidas, que no requiere de corriente en la base para mantenerse en conducción pero que posee muy buenas prestaciones en esta etapa. Es el semiconductor ideal para trabajar con convertidores de potencia. En el siguiente apartado vamos a ver los tipos de conversiones que hay [20].

3.1.2 Convertidores de potencia

Como ya hemos ido comentando, los semiconductores de potencia van a ser parte fundamental en el control de los convertidores de potencia y son la unidad más pequeña programable, así que vamos a contar con ellos en todos los tipos de conversión. Para entender mejor la función que desempeñan en cada una de las topologías, analizaremos sus detalles en este apartado.

Como podemos ver en el esquema de la *Figura 66* existen cuatro tipos de convertidores, dos que trabajan con el mismo tipo de tensión, y otros dos que se encargan de transformarla. Nos vamos a centrar en analizar el funcionamiento, y los tipos de cada topología, centrandó especial atención en la labor de los

semiconductores. Para el inversor veremos más profundamente sus pormenores en el apartado 3.1.3 *El inversor*.

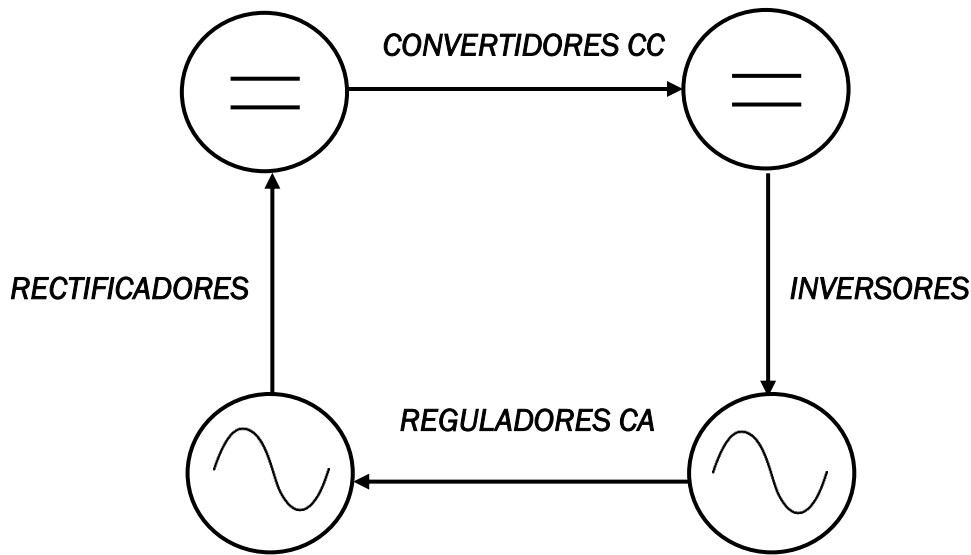


Figura 66. Esquema de tipos de convertidores de potencia

3.1.2.1 Convertidores CC/CC

Los convertidores de CC/CC son como una especie de transformador donde se especifica la tensión a la entrada y a la salida del mismo se da el mismo tipo de tensión, pero con un valor diferente: mayor o menor.

Podríamos incluir este tipo de convertidores dentro del grupo de fuentes de alimentación conmutada sin transformador, y es que esa es una de las ventajas de su uso, la ausencia de transformadores que pueden interferir en la señal. A cambio, los convertidores de continua van a estar compuestos siempre por los mismos elementos: condensador, autoinducción, diodo, resistencia de carga y semiconductor de potencia.

El funcionamiento es sencillo y parecido en todas las posibles aplicaciones. Se trata de almacenar y entregar energía temporalmente hacia una salida con voltaje mayor o menor. El almacenamiento se produce en el condensador (tensión) y en la autoinducción (corriente), y la entrega lineal de energía a la resistencia de carga se asegura mediante el diodo.

El semiconductor hace posible la transición temporal de potencia. Al ajustar el ciclo de servicio D^{12} , se controla la cantidad de energía transmitida a la carga y por lo tanto la tensión de salida.

¹² El ciclo de servicio es la relación entre el tiempo en el que el interruptor está apagado y el periodo total de funcionamiento.

Las tres posibles configuraciones que dan lugar a las diferentes aplicaciones del convertidor son las siguientes [10]:

- Convertidor reductor (Buck)

La tensión de salida menor que la de entrada

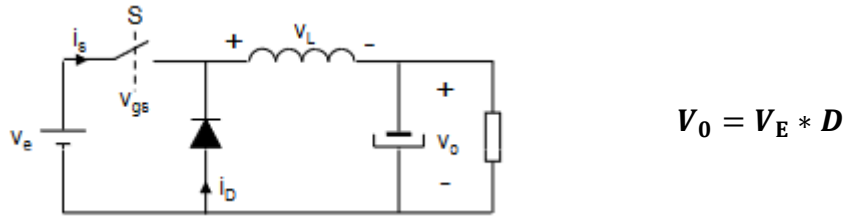


Figura 67. Convertidor reductor

- Convertidor elevador (Boost)

La tensión de salida mayor que la de entrada

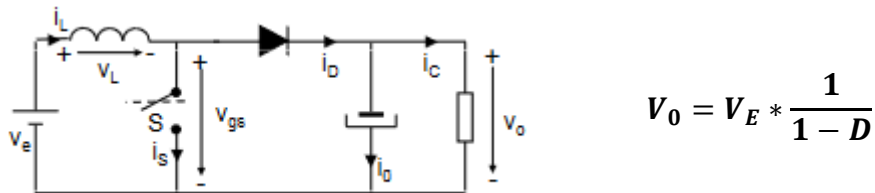


Figura 68. Convertidor elevador

- Convertidor reductor-elevador (Buck-boost)

La tensión de salida va a ser mayor o menor que la de entrada en función del valor del ciclo de trabajo [10].

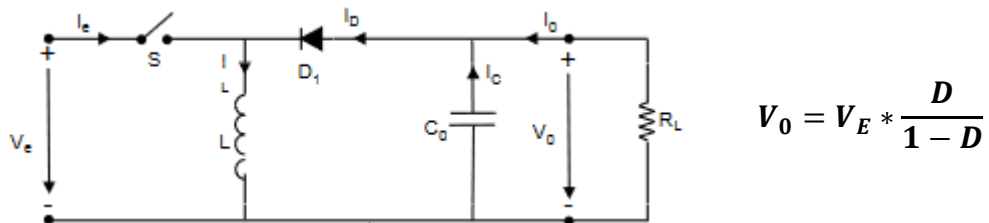


Figura 69. Convertidor reductor-elevador

3.1.2.2 Reguladores CA

Los reguladores de corriente alterna tienen la misma función que los convertidores de continua, cambiar el valor de la amplitud de la corriente sin alterar su forma. Esta regulación se puede realizar también mediante un convertidor de CC/CC acompañado de un rectificador CA/CC y un inversor CC/CA a la entrada y a la salida respectivamente, pero el uso de estos reguladores incluye la ventaja de reducir la conversión a una única etapa.

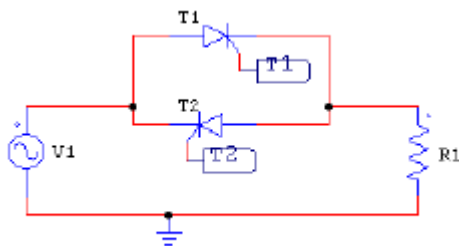
Entre los inconvenientes de estos reguladores se encuentra el gran contenido en armónicos de la tensión de salida que conlleva a añadir siempre un filtro en la salida, y una frecuencia no siempre igual que la de la onda de entrada con además algo de desfase.

El funcionamiento de estos reguladores va a tener su base en el uso de tiristores como elementos conmutadores de potencia y es que parten con la ventaja de no necesitar un bloqueo forzado (tiristores GTI) por el paso natural por cero de la intensidad, así que solo es programado un disparo, el inicial.

Existen fundamentalmente dos tipos de reguladores en función del número de señales usadas [21]:

- Regulador Total

Una sola onda de entrada con dos tiristores para el control bidireccional de la señal.



Mayor número de armónicos, pero mayor variedad de amplitud a la salida

Figura 70. Regulador de CA total

La señal de salida tiene la siguiente forma:

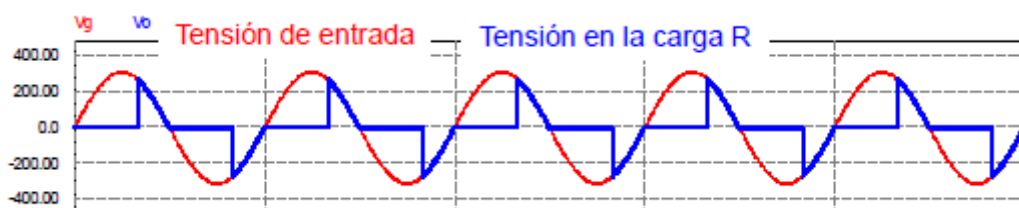
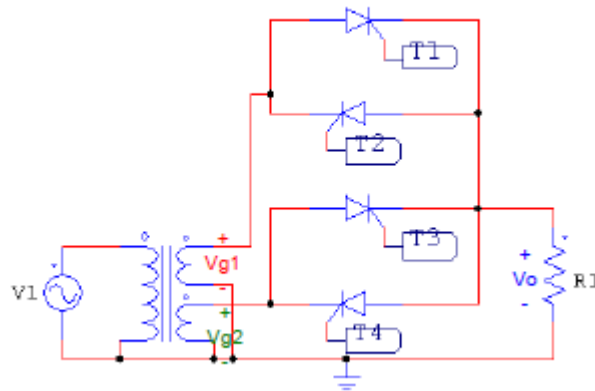


Figura 71. Señal de salida del regulador total

- Regulador diferencial

Dos señales de onda a la entrada generadas mediante un transformador con dos pares de tiristores para cada señal.



Menor número de armónicos, y menos variedad de amplitud a la salida

Figura 72. Regulador de CA diferencial

La señal de salida es la siguiente [21]:

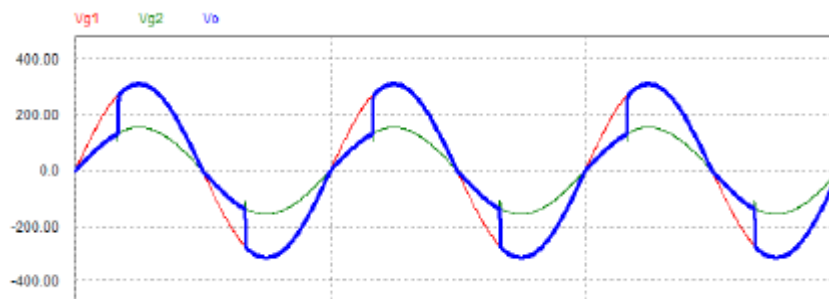


Figura 73. Señal de salida de regulador diferencial

3.1.2.3 El rectificador

Los rectificadores son los primeros convertidores que vamos a analizar que cambian el tipo de tensión, de CA a CC. La tensión de salida en este caso va a estar formada por diferentes tramos de la tensión de entrada que se corresponden normalmente con las cúpulas de las ondas de la que va a ser una tensión de entrada polifásica.

Los semiconductores más usados para el control de estos convertidores van a ser como para los reguladores, diodo o tiristores, que con corriente alterna tienen la ventaja de no necesitar bloqueo forzado. El uso de tiristores va a permitir un control mucho más estricto que si se usan diodos, por el simple hecho de poder programar el ángulo de disparo que influye directamente en la forma de onda.

Es importante también para el control de la tensión de salida conocer el número de fases que tenemos a la entrada, esto es el número M , que afecta directamente a la forma de onda de la tensión rectificada. Cuantas más fases en la entrada, más cúpulas tenemos en la entrada y más rectificada va a estar la señal.

Otra forma de clasificar al rectificador es según la topología que se utilice, esto es, topología simple con neutro común, o topología serie, con montaje en puente. Un rectificador simple solo va a tener una vía de rectificación¹³ y por lo tanto la mitad de los semiconductores que un rectificador en puente, con doble vía y el doble de semiconductores.

Con todo esto ya podemos hacer una clasificación general de los rectificadores que podemos encontrar [22]:

1. Rectificador no controlado simple

Uso de diodos para la rectificación de cada fase que comparten un solo neutro.

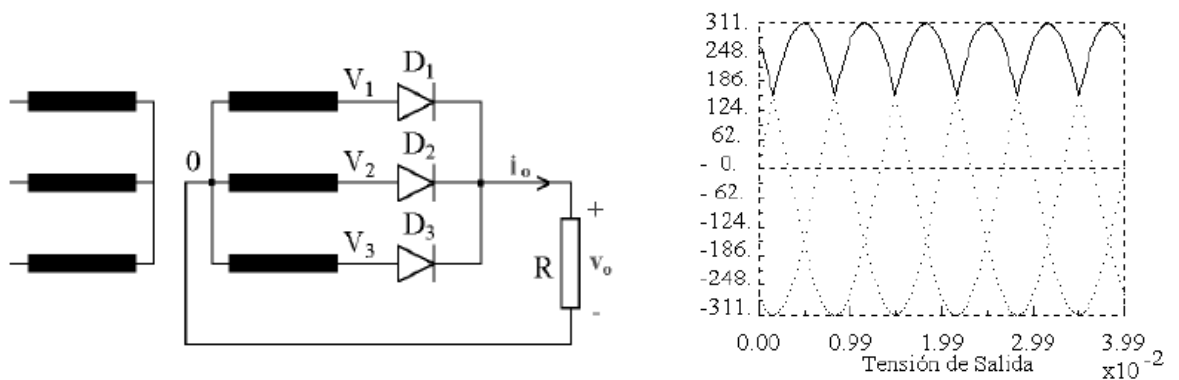


Figura 74. Topología y tensión de salida de un rectificador no controlado simple trifásico

La salida toma cada cúpula de onda para generar una señal continua

2. Rectificador no controlado serie

Uso de diodos para la rectificación de cada fase dispuestos en tantas ramas como fases tenga la tensión de entrada

¹³ Un único valor de tensión rectificada a la salida

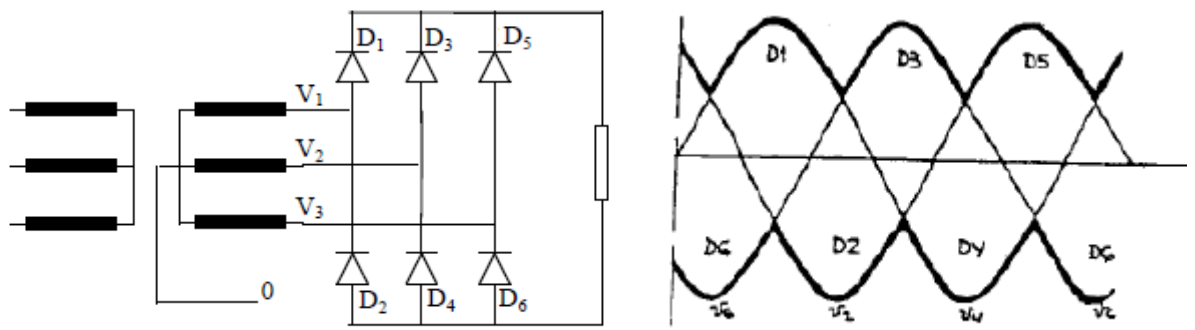


Figura 75. Topología y tensión de salida de un rectificador no controlado serie trifásico

La salida toma dos valores, uno positivo y otro negativo, el controlado por los diodos superiores, y el de los inferiores.

3. Rectificador controlado simple

Uso de tiristores para la rectificación con la opción de controlar la tensión de salida mediante el ángulo de disparo además de por el número de fases.

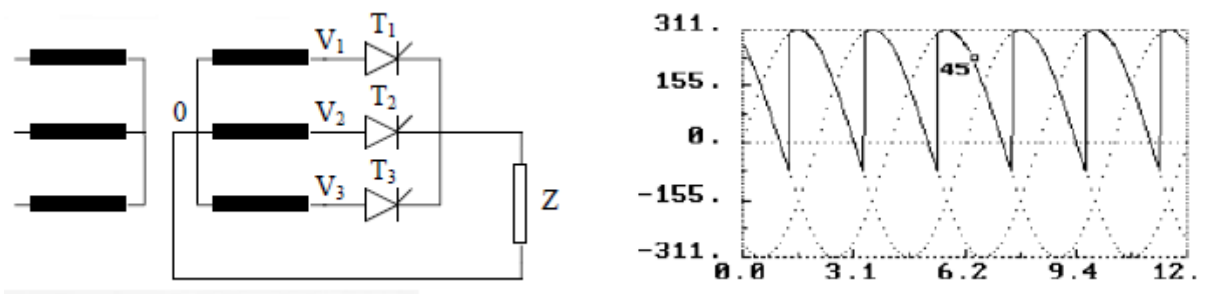


Figura 76. Topología y tensión de salida de un rectificador controlado simple trifásico con ángulo de disparo de 45°

Con este tipo de control, al añadir un parámetro controlable más, vamos a ser capaces de gestionar un mayor rango de valores para la tensión continua de salida.

4. Rectificador controlado serie

Uso de tiristores para la rectificación, con control mediante ángulo de disparo y número de fases para dar lugar a un único valor de tensión, a diferencia del rectificador no controlado.

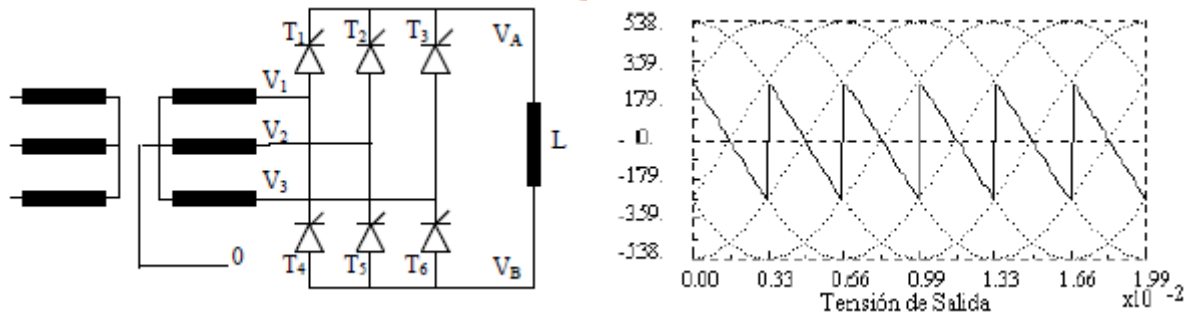


Figura 77. Topología y tensión de salida de un rectificador controlado serie trifásico con ángulo de disparo de 90°

La principal diferencia con la tipología simple es la influencia de la carga en la tensión final, con una carga inductiva el resultado es el mismo, pero con la carga resistiva se anulan las tensiones negativas [22].

Tras el análisis de los semiconductores más importantes y de su aplicación en tres de los cuatro convertidores de potencia que existen, es el momento de entrar de lleno en la aplicación que más nos interesa, y que trataremos de controlar desde la tarjeta LAUNCHXL-F28379D: El inversor.

3.1.3 El inversor

El inversor, o convertidor CC/CA, se va a encargar, gracias a la conmutación de los polos de potencia (semiconductores) de generar una tensión de salida alterna controlada en fase, amplitud y frecuencia, a partir de una señal de entrada continua.

Por la gran cantidad de aplicaciones que tiene como alimentación, actuador, o generador de fuentes de energía renovable, es la topología más conocida y utilizada en el ámbito industrial.

Aunque nos centraremos principalmente en el control del inversor, es importante analizar las diferentes partes y el funcionamiento del inversor en primer lugar. En la Figura 78 vemos un primer esquema de la estructura básica del inversor.

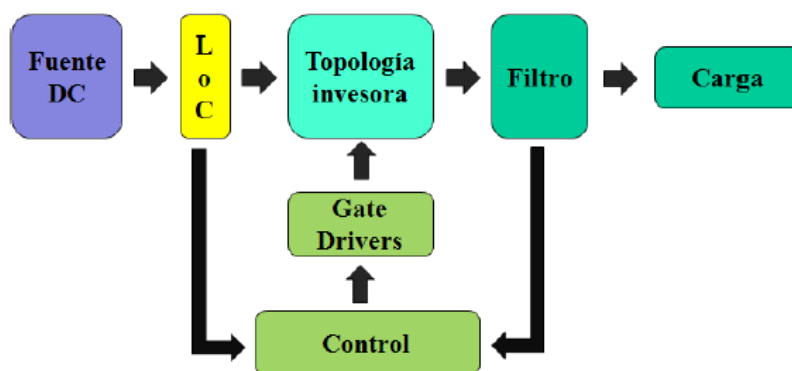


Figura 78. Estructura de un inversor

Como podemos observar, se podrían diferenciar tres partes fundamentales en la estructura entre la entrada continua y la salida alterna:

- **Topología inversora:** Etapa de potencia que se puede presentar en forma de tres diferentes topologías: Semipuente, puente y trifásica. Analizaremos las tres en el apartado 3.1.3.2 Tipos de inversores.
- **Filtro:** Etapa de salida del convertidor para obtener la salida senoidal. Análisis en el apartado 3.1.3.3 Filtro.
- **Control:** Etapa de control que se encarga de recoger los valores de salida del filtro y compararla con unos valores predeterminados, que darán como resultado una secuencia de pulsos que leerán los polos de potencia de la etapa de potencia. En el apartado 3.2 CONTROL DEL INVERSOR , analizaremos las diferentes técnicas de control que se pueden aplicar en un inversor, y más adelante presentaremos el algoritmo de control que finalmente utilizaremos.

3.1.3.1 Funcionamiento

En general todos los tipos de inversores tienen el mismo procedimiento para conseguir generar una señal alterna desde una continua.

La clave de este proceso es conseguir dirigir la corriente a través de la carga en dos direcciones, la positiva y la negativa. Un cambio de flujo que sea controlable.

La manera más simple de conseguir esto es utilizando interruptores (polos de potencia) que estén emparejados dos a dos, con dos interruptores por rama como máximo, lo que puede dar lugar a inversores de una rama con dos interruptores, dos ramas con cuatro interruptores o tres ramas con seis interruptores.

Es muy importante emparejar a uno de los dos interruptores de cada rama con la carga y con otro interruptor de una rama diferente¹⁴, si la hay. Esto quiere decir que estos dos interruptores estarán controlados por la misma señal y conmutaran al mismo tiempo, produciendo un flujo de tensión (positivo o negativo) en la carga.

3.1.3.2 Tipos de inversores

Como ya hemos comentado todos los tipos de inversores tienen un punto en común: No puede haber más de dos polos por rama. Pero las diferencias principales en este caso estriban en: El **número de fases** que se quieren dar en la salida (monofásico o trifásico) que afecta directamente al número de ramas, y en la **topología de la etapa de potencia** (tipo puente o semipuente) válidas solo para salidas monofásicas, que

¹⁴ Es muy importante que no conmuten a la vez los dos interruptores de una misma rama, pues esto conduce a cortocircuitos en la carga.

se diferencian principalmente por la conexión de la fuente de tensión de entrada. Para el caso de semipunte hay conexión con la carga en el punto medio de las dos tensiones de entrada, para el caso de puente no hay conexión directa entre la carga y la tensión de salida.

- Inversor monofásico en semipunte

Dos fuentes de tensión continua iguales en la entrada, y con una sola fase de salida resultado de una sola rama con dos interruptores. Como se indica en la *Figura 79*.

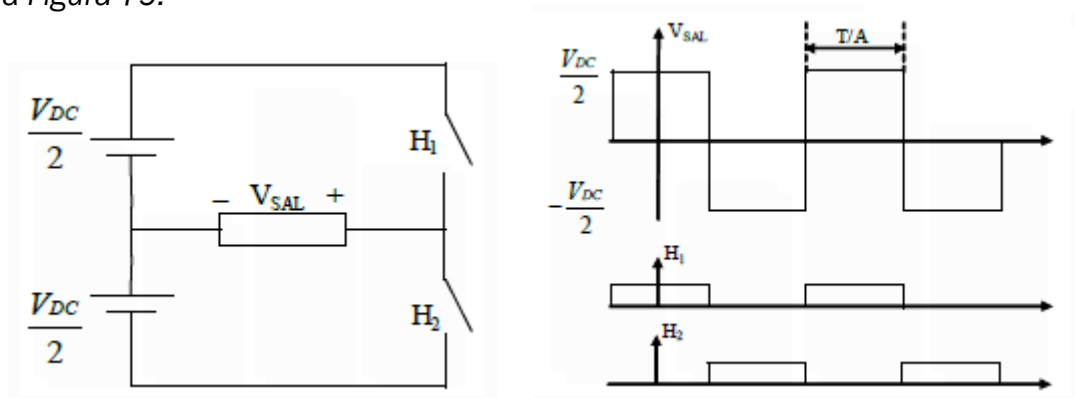


Figura 79. Topología y tensión de salida de inversor monofásico en semipunte

En este caso la salida tiene solo dos valores de tensión que se generan justo cuando uno de los dos interruptores está cerrado y el otro abierto. La amplitud se corresponde con la del voltaje de entrada.

- Inversor monofásico en puente

Dos tensiones de entrada, con una fase en la salida, pero con dos ramas para generarla que implican dos pares de polos en cada rama. En la *Figura 80*, podemos apreciar la tensión en V_a y en V_b , que da como resultado la tensión final. V_a es controlada por los interruptores H_1 y H_2 , y V_b por H_3 y H_4 .

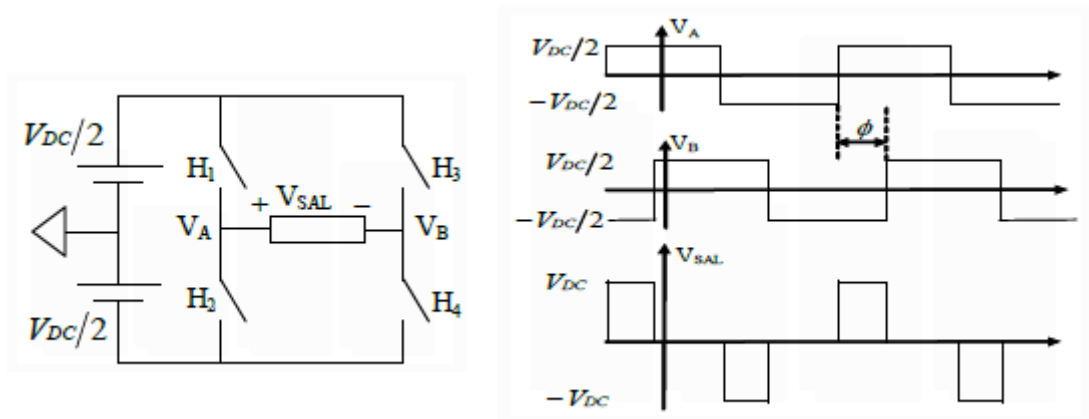


Figura 80. Topología y tensión de salida de inversor monofásico en puente

En este caso, al poder controlar el desfase ϕ entre señales de control, controlamos también la amplitud de la tensión de salida siendo máxima para un ϕ de 180° que tendría la forma final de la Figura 79.

- Inversor trifásico

Dos tensiones continuas e iguales a la entrada y tres fases a la salida, una por cada rama y conectadas en estrella con los correspondientes dos polos de potencia. Para este caso, como se representa en la Figura 81, tenemos dos posibles tensiones en la salida, las de línea, o las de fase, las tensiones de fase son las que se dan entre los polos, las tensiones de línea, la diferencia entre dos de fase.

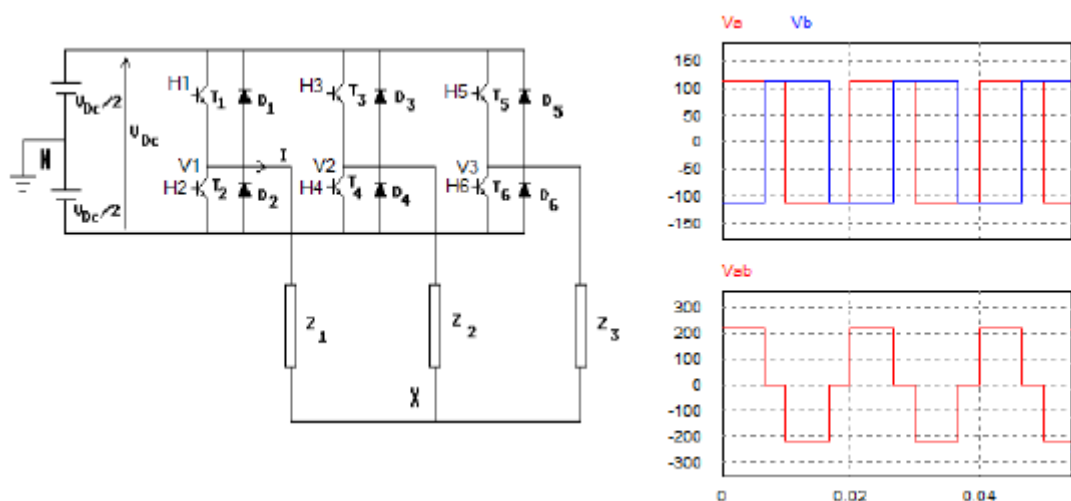


Figura 81. Topología y tensión de salida de inversor monofásico en puente

En el inversor trifásico no hay forma de controlar la salida de ninguna manera, el ángulo entre fases es siempre fijo para un sistema equilibrado, y la amplitud de cada nivel es siempre constante para eliminar componente continua [23].

Para conocer más detalles del control de las distintas topologías emplazamos al lector al apartado 3.2 CONTROL DEL INVERSOR sobre técnicas de control más conocidas de los inversores.

3.1.3.3 Filtro

Por último y para terminar este apartado sobre el funcionamiento general de un inversor, nos vamos a centrar en la última parte del sistema: el filtro.

Los filtros son elementos activos, y necesarios en cualquier circuito eléctrico conectado a la red, su función principal es eliminar el contenido armónico de la tensión y dejar pasar solo la frecuencia fundamental, lo que se denomina un filtro paso-bajo.

Principalmente están compuestos por una serie de elementos reactivos como son las inductancias y los condensadores. El más conocido es el filtro LC, aunque para aplicaciones industriales donde se requiere poco peso se utilizan filtros LCL, como el que analizaremos a la hora de proponer una solución real al proyecto en el apartado 4.3.1 Elementos físicos necesarios.

Para seleccionar el valor de los elementos reactivos necesitamos conocer primero el diagrama de Bode¹⁵ de un filtro paso-bajo LC que es el que se muestra en la *Figura 82*.

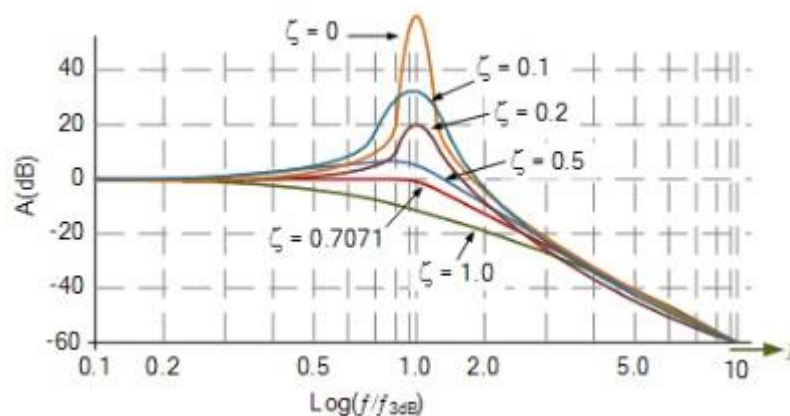


Figura 82. Diagrama de bode de ganancia de un filtro paso-bajo

El coeficiente de amortiguamiento ξ que se ve en la figura es el número que multiplica, en el denominador de una función de transferencia de segundo orden, al segundo término de s . Para el caso del filtro LC la función de transferencia es la siguiente:

¹⁵ Representación gráfica de la ganancia de un sistema en función de la frecuencia

$$\frac{1}{1+LCs^2} \quad (6)$$

Como no tiene termino en s podemos decir que ξ va a ser cero, y por lo tanto se puede dar a determinadas frecuencias un repunte considerable de ganancia.

Como vemos también el coeficiente de s^2 se corresponde con el producto de L por C, como este coeficiente tiene el valor de $(1/w_0)^2$ y s en el dominio de la frecuencia es w podemos confirmar el desarrollo de la siguiente ecuación:

$$\left(\frac{w}{w_0}\right)^2 = LCw^2 \rightarrow w_0 = \frac{1}{\sqrt{LC}} \quad (7)$$

Donde w_0 es la frecuencia de corte a partir de la cual se disminuye la ganancia total del sistema. Al conocer la ganancia que queremos atenuar podemos deducir la frecuencia de corte y con ello el valor de L y de C [23]

3.2 CONTROL DEL INVERSOR

A diferencia del primer apartado en este vamos a centrar la exposición en un único convertidor de potencia: El inversor, concretamente en su control, tras haber descrito la etapa de potencia y el filtro final en el apartado 3.1.3 El inversor.

Analizaremos dos de las técnicas de control más conocidas, pero prestando especial atención al control vectorial, pues es la técnica que desarrollaremos en la fase práctica del proyecto.

Los dos tipos de control están basados en la técnica de modulación de ancho de pulso PWM, considerando esta la mejor opción para el control de inversores por su sencillez para implementarse, y por la cantidad de aplicaciones que puede llevar a cabo. Para conocer algo más sobre esta técnica véase el apartado 2.3.1 Control PWM

Para **sistemas monofásicos** en topología puente o semipuente son muy interesantes las técnicas de modulación por ancho de pulso SPWM, sobre todo por la facilidad de implementación, aunque eso conduzca a unas elevadas pérdidas por conmutación debido a la aparición de armónicos por el número elevado de conmutaciones.

Para **sistemas trifásicos** es preferible el otro tipo de modulación, la modulación de ancho de pulso por vectores espaciales SVPWM. Con esta técnica se logran reducir los armónicos y por tanto las perdidas en conmutación.

3.2.1 Control por ancho de pulso. SPWM

La base del control por ancho de pulso para cualquier inversor es la **función existencial**. Esta ecuación va a dar como resultado siempre o un 1 o un 0, o cerramos el interruptor o lo mantenemos abierto. Y gracias a parámetros como A, w o α , podremos configurar: el ancho, la frecuencia de conmutación y el ángulo de retardo del pulso. La forma de esta función es la siguiente:

$$H_1 = \frac{1}{A} + \frac{2}{\pi} \sum_{n=1}^{\infty} \left[\frac{\text{sen}\left(\frac{n\pi}{A}\right)}{n} \right] \cos n(\omega t - \alpha) \quad (8)$$

Para generar la función existencial existen dos posibles formas que se distinguen únicamente por el cambio del ancho del pulso A:

1. Pulso único por semiciclo: El valor de A va a ser siempre el mismo.
2. Pulso múltiple por semiciclo: Se aprovechará el cambio del valor de A en cada ciclo para optimizar el valor de la tensión de salida.

3.2.1.1 Pulso único

En este caso se mantiene un único pulso por semiciclo debido al invariable valor del ciclo de servicio. Además, siempre va a haber una razón para mantener constante el valor de A. ya sea para anular la componente continua (topología monofásica en semipunto y trifásica) o para anular los armónicos pares (topología monofásica en puente).

Es el tipo de control más sencillo que nos podemos encontrar, ya que no se necesita más que un generador de onda cuadrada para controlar el inversor, el valor de la frecuencia de conmutación de los polos no afecta al resultado de tensión en la salida y el ángulo de desfase entre dos señales diferentes es siempre aconsejable de 180°.

En definitiva, control fácil de implementar, que no necesita ningún parámetro de entrada, o lo que es lo mismo realimentación del propio circuito. Su uso va a condicionar siempre la **aparición de un convertidor CC/CC**, anterior al inversor, que va a controlar el valor de la tensión de pico de salida de alterna ya que ningún parámetro interno va a ser controlable.

3.2.1.2 Pulso múltiple

El objeto de este otro tipo de control es el de añadir nuevos parámetros controlables dentro de la función existencial, a parte de los ya comentados, para conseguir así más flexibilidad a la hora controlar la frecuencia o la tensión de la señal alterna de salida.

La forma más fácil de lograr todo esto es generando una nueva función existencial, con ancho de pulso variable en cada semiciclo, a partir de otras dos señales: La portadora y la moduladora¹⁶. Con esto vamos a tener dos nuevos índices para controlar:

✚ La modulación de la amplitud: x o m_a

Es la relación entre la tensión máxima de la moduladora y la de la portadora. Normalmente el valor de tensión máxima de la portadora es 1 así que este parámetro coincide con el valor de la tensión de pico de la moduladora, que va a determinar el valor de la amplitud de la tensión de salida.

✚ La modulación de la frecuencia: K o m_f

Es la relación entre la frecuencia de la moduladora (w) y la de la portadora (w_s). Va a ser un valor mayor que uno ya que w va a ser siempre mayor que w_s . Este parámetro determinará las frecuencias del espectro armónico.

Si el índice x sobrepasa el valor de 1 tendremos la ventaja de una tensión en la salida mayor que el valor en continua, pero a consecuencia de esto aparecerán más armónicos a frecuencias más pequeñas.

Para K tampoco es recomendable un valor encima de 21 ya que las dos señales de control no estarían en fase y aparecerían subarmónicos en el espectro.

En cuanto a los ángulos de desfase, solo los tendremos en cuenta en la topología monofásica en puente, donde vamos a tener en cuenta los valores de desfase entre funciones de modulación ϕ_1 y funciones existenciales ϕ_2 .

Con un valor de desfase entre funciones moduladoras de 180° , conseguiremos maximizar el valor de la amplitud.

En cuanto al desfase entre funciones existenciales si este es 0° la salida se va a modular en tres niveles y se anulan además los armónicos impares, este control es conocido por **SPWM Unipolar**. Si el desfase es de 180° la modulación es entre dos niveles y los armónicos impares no se anulan, este control se denomina **SPWM Bipolar** [23][24].

3.2.2 Control vectorial. Modulación SVPWM

Para hacer frente a aplicaciones industriales, donde la salida del inversor tiene que ser trifásica, estudiaremos en este apartado la técnica de modulación por ancho de pulso del vector espacio, o lo que es lo mismo, el control vectorial, utilizado muy frecuentemente para sistemas trifásicos.

¹⁶ Este tema ya ha sido tratado en el apartado 2.3.1 Control PWM. Para explicar el control PWM en este punto nos centraremos en su aplicación a los diferentes tipos de inversores

El objetivo de este control es realizar una serie de transformaciones en el sistema trifásico de salida de la etapa de potencia, con el fin conseguir dos señales continuas y constantes con el tiempo para poder así compararlas con referencias constantes, y con esto ajustar el ciclo de servicio del inversor a través de la modulación SVPWM.

Para lograr todo este proceso necesitamos conocer las herramientas que permiten transformar las tres señales senoidales en dos señales continuas. Por ello en los siguientes apartados estudiaremos la representación como un solo vector del sistema trifásico, y como a partir de este vector logramos, primero un sistema con dos señales senoidales en el plano α - β a través de la **transformada de Clarke** y después dos señales constantes en el plano d-q a partir de la **transformada de Park**.

Por último, analizaremos también la modulación SVPWM que a través del vector resultante del control del sistema trifásico marcará una referencia que indica al inversor los polos que deben conmutar.

3.2.3.1 Representación vectorial de sistemas trifásicos

Cualquier sistema trifásico puede representarse en un plano bidimensional, representando las componentes de tensión, desfasadas entre sí 120° siempre que el sistema esté equilibrado.

En un sistema trifásico de **cuatro hilos**, donde se toman las tensiones de las tres fases: V_a , V_b y V_c y el neutro, la representación vectorial en un plano α - β es la que se indica en la *Figura 83*:

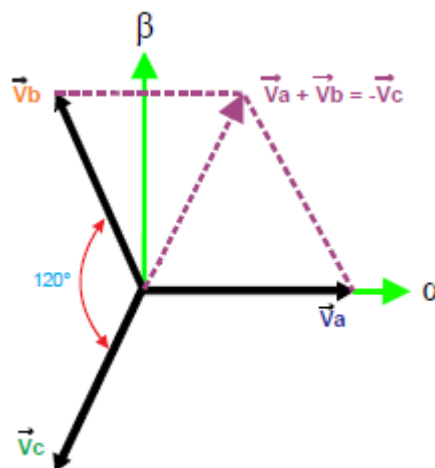


Figura 83. Distribución de tensiones de fase de un sistema trifásico en el plano bidimensional α - β

En la figura podemos observar también como al ser un sistema equilibrado se da siempre la relación:

$$V_A + V_B + V_C = 0 \quad (9)$$

Por lo tanto, el sistema trifásico se puede reducir a un solo vector V_R que opera a una magnitud de voltaje constante V y a una fase ϕ variable con el tiempo. Al ser la suma vectorial en cada instante de tiempo, este vector va a tener también una evolución temporal como la respuesta senoidal, representada por un giro alrededor del origen a una frecuencia ω , como se puede observar en la *Figura 84*, donde se representan tres instantes de tiempo de un sistema trifásico.

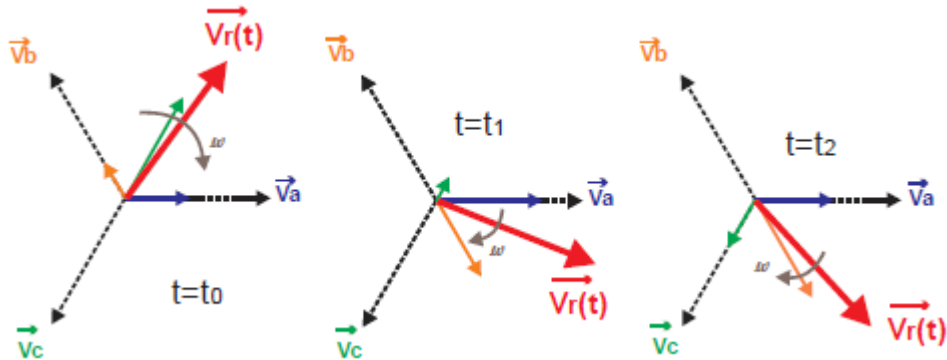


Figura 84. Evolución del vector resultante de un sistema trifásico en tres instantes de tiempo.

Por otro lado, la expresión matemática del vector V_R es la siguiente:

$$\vec{V}_r(t) = \frac{2}{3} (\vec{V}_a + \vec{V}_b e^{j120^\circ} + \vec{V}_c e^{j240^\circ}) \quad (10)$$

[25][26][27]

3.2.3.2 Transformada de Clarke

Esta transformada, que simplifica el análisis de un sistema trifásico, permite pasar de este tipo de sistema con coordenadas a , b y c a un sistema bifásico con dos ejes estáticos α y β . La relación entre estos dos sistemas la podemos observar en la *Figura 85*.

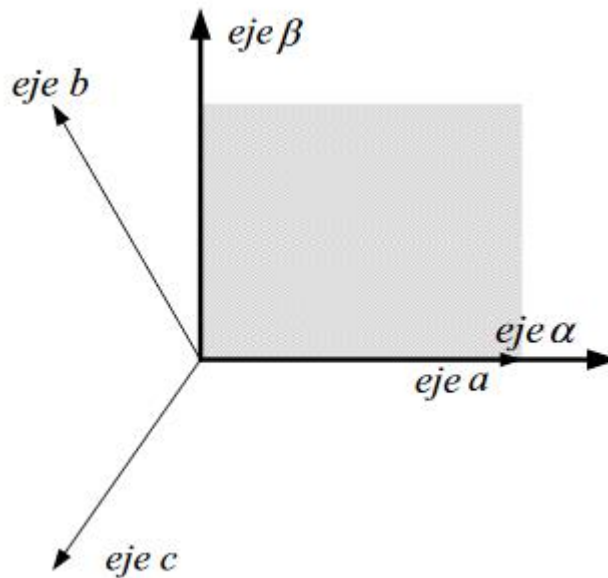


Figura 85. Relación entre ejes a-b-c y α - β

Sabiendo que el vector en el sistema a-b-c va a tener la forma de una matriz de dimensiones 3×1 y el vector que queremos en el plano α - β debe tener unas dimensiones de 2×1 podemos conocer las dimensiones de la matriz de transformación que serán de 2×3 .

Para cualquier vector con coordenadas a, b, c sus valores en el eje α y en el eje β son:

$$\alpha = a + b * (-\text{sen}(30)) + c * (-\text{sen}(30)) \quad (11)$$

$$\beta = a * \cos(90) + b * \cos(30) + c * (-\cos(30)) \quad (12)$$

Sabiendo también que el factor de escala aplicado a un vector en un sistema trifásico para que coincida el módulo en el sistema bifásico es de $2/3$, la matriz de transformación es la siguiente:

$$[T_{\alpha\beta}] = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \quad (13)$$

A partir de esta matriz de transformación siempre que tengamos un vector en el plano trifásico podemos pasarlo al plano bidimensional al multiplicar sus valores por los de la transformada de Clarke.

En el caso del sistema trifásico de tensiones convertiremos las señales senoidales de V_a , V_b y V_c , desfasadas 120° , en dos señales V_α y V_β desfasadas 90° como se puede ver en la Figura 86 [28].

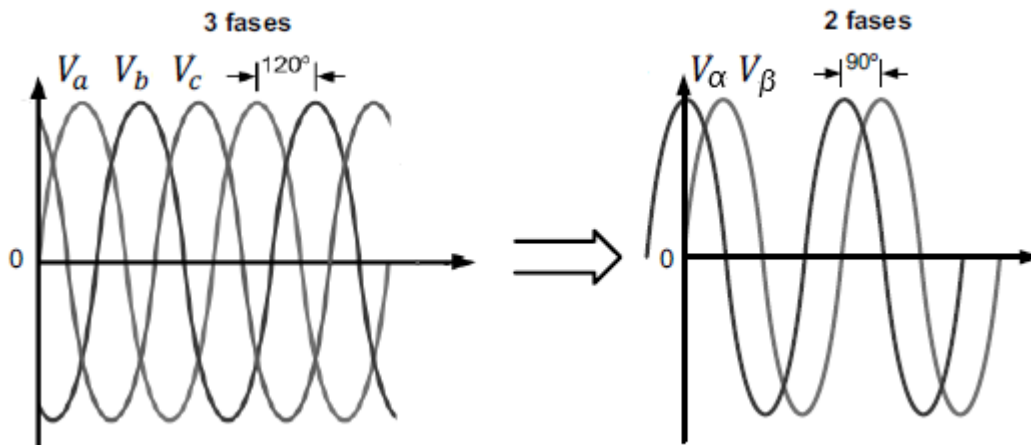


Figura 86. Resultado de la transformada de Clarke

3.2.3.3 Transformada de Park

Una vez que tenemos las señales en un plano dimensional nos hemos ahorrado ya la aparición de tres ejes y el desfase de 120° . El siguiente paso es conseguir transformar las dos señales senoidales, dependientes del tiempo, en dos señales constantes y estáticas. Para conseguir esto vamos a pasar del sistema estático de dos ejes α - β a un sistema en movimiento, dinámico, que gire a la velocidad angular ωt de las señales, con dos ejes d y q .

En la *Figura 87* se ve más clara la relación entre sistemas, si los ejes d - q giran a la misma velocidad angular que el sistema trifásico inicial, podremos considerar los valores en dq constantes.

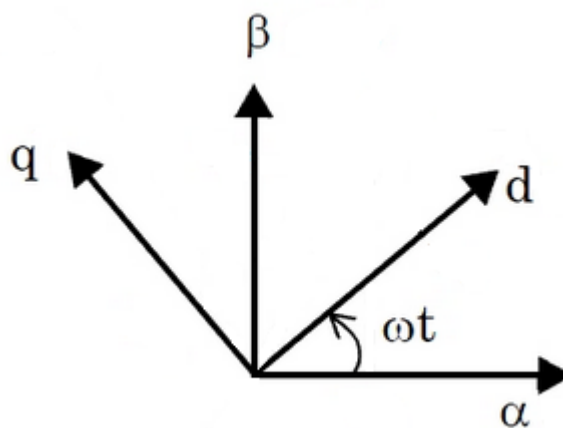


Figura 87. Relación entre ejes α - β y d - q

Al tener dos sistemas bifásicos también podemos deducir que la matriz de transformación tendrá unas dimensiones de 2x2. Las relaciones entre coordenadas de los ejes para esta transformada vienen en función de la frecuencia de giro w :

$$d = \alpha * \cos(wt) + \beta * \text{sen}(wt) \quad (14)$$

$$q = \beta * \cos(wt) - \alpha * \text{sen}(wt) \quad (15)$$

En este caso no hay que tener en cuenta ningún factor de escala así que la matriz de transformación es la siguiente:

$$[T_{dq}] = \begin{bmatrix} \cos(wt) & \text{sen}(wt) \\ -\text{sen}(wt) & \cos(wt) \end{bmatrix} \quad (16)$$

Con esta transformada podemos cambiar de un sistema dinámico a un sistema estático que lo que va a generar son dos señales constantes en los ejes d y q [28].

3.2.3.4 Modulación SVPWM

En apartados anteriores, como en el 3.2.1 Control por ancho de pulso. SPWM, hemos analizado la modulación por ancho de pulso SPWM, en la que, a partir de la función existencial, variábamos algunos parámetros como el ancho de pulso, el ángulo de desfase, o la frecuencia de conmutación para cambiar la forma de onda final en alterna. Como también hemos comentado, esta modulación produce un gran espectro armónico y por lo tanto pérdidas en conmutación, es solo recomendable para **sistemas monofásicos**.

Con el fin de reducir las pérdidas en conmutación y ofrecer mejoras en el bus de corriente continua, se utiliza la modulación de ancho de pulso por vectores espaciales SVPWM para **sistemas trifásicos**, que emplea el vector representativo del sistema, V_R , para generar un lazo de control cerrado de alta eficiencia que reduce el espectro armónico. Este sistema de modulación se basa en los estados de conmutación del inversor trifásico.

Como ya vimos en el apartado 3.1.3.2 Tipos de inversores, este tipo de inversor está formado por tres ramas, cada una con dos interruptores que deben cumplir siempre dos condiciones:

- 1) En cada rama no podemos tener dos polos con el mismo estado.
- 2) Siempre tiene que haber un semiconductor conduciendo por cada rama.

Si partimos de que cada rama tiene dos estados que varían entre 0 y 1 (estado de un polo, el otro será al contrario) podemos concluir que existen hasta ocho posibles combinaciones de funcionamiento del inversor, como se muestran en la *Figura 88*.

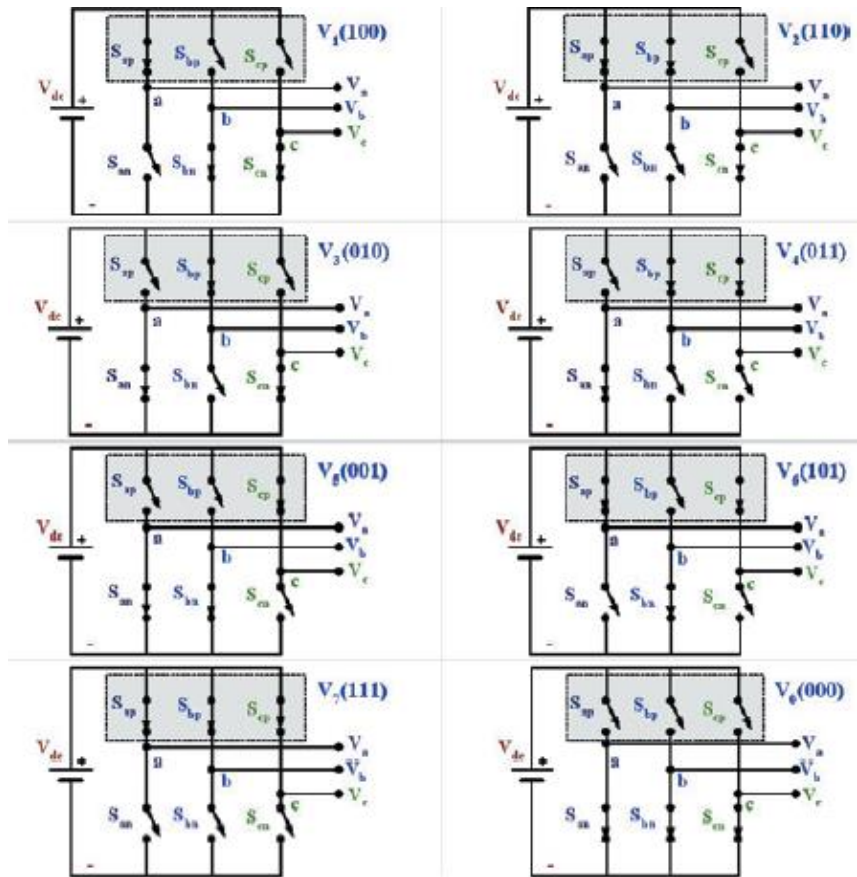


Figura 88. Estados de conmutación de un inversor trifásico

De los ocho posibles estados del inversor, nos podemos dar cuenta de que el V_7 y el V_0 son vectores nulos que aun así cumplen con 1) y 2).

El resto de los estados cambian sus valores de tensiones de fase: V_a , V_b y V_c , entre V_{dc} , $-V_{dc}$ y 0 de la forma que indica la *Tabla 10*.

Tabla 10. Estados de tensiones de fase de un inversor trifásico

Vector	Estados interruptores	V _a	V _b	V _c	Expresión SVV
V0	(000)	0	0	0	$\vec{V}_0 = 0V$
V1	(100)	+V _{dc}	0	-V _{dc}	$\vec{V}_1 = \frac{2}{\sqrt{3}}V_{dc} \angle 0^\circ$
V2	(110)	0	+V _{dc}	-V _{dc}	$\vec{V}_2 = \frac{2}{\sqrt{3}}V_{dc} \angle -60^\circ$
V3	(010)	-V _{dc}	+V _{dc}	0	$\vec{V}_3 = \frac{2}{\sqrt{3}}V_{dc} \angle -120^\circ$
V4	(011)	-V _{dc}	0	+V _{dc}	$\vec{V}_4 = \frac{2}{\sqrt{3}}V_{dc} \angle -180^\circ$
V5	(001)	0	-V _{dc}	+V _{dc}	$\vec{V}_5 = \frac{2}{\sqrt{3}}V_{dc} \angle -240^\circ$
V6	(101)	+V _{dc}	-V _{dc}	0	$\vec{V}_6 = \frac{2}{\sqrt{3}}V_{dc} \angle -300^\circ$
V7	(111)	0	0	0	$\vec{V}_7 = 0V$

Representando cada uno de estos estados como un vector, obtenemos seis vectores con el mismo módulo, pero con una diferencia de ángulo entre vector de 60°. En la última columna de la *Tabla 10* podemos observar estos valores del vector que le corresponden a cada estado.

Si representamos los seis vectores con su módulo y su ángulo en los ejes estáticos α y β , vamos a obtener el conocido como hexágono vectorial, donde cada vértice se corresponde con cada uno de los seis vectores con módulo y fase. En la *Figura 89* podemos apreciar mejor este hexágono.

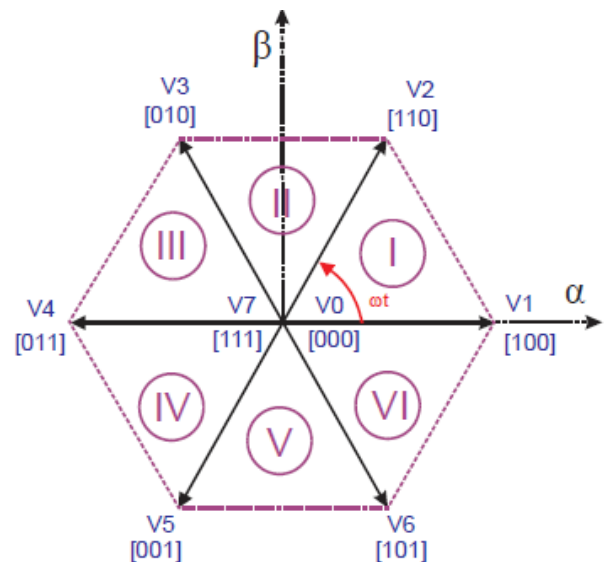


Figura 89. Hexágono Vectorial

En este tipo de representaciones siempre se indica un vector inicial que se corresponda con valor 0 en el eje β y positivo en el eje α . Para este caso es el vector V1 ya que se aplica corriente continua en la primera fase.

V7 y V0 no tienen módulo ni fase así que se corresponden con la coordenada 0 del sistema de ejes.

Como podemos ver, con una sucesión de combinaciones de estados de polos en el inversor vamos a ser capaces de generar una sucesión de vectores unitarios para un sistema trifásico como son V1, V2, V3... El resultado van a ser tres formas de onda

cuadradas de salida para cada una de las tres tensiones de línea del sistema trifásico con un valor máximo de V_{dc} , mínimo de $-V_{dc}$ e intermedio en 0. Cada una de ellas va a estar también desfasada de las otras 120° . Aplicar esto sería aplicar una modulación SPWM.

Para hacer uso de la técnica de modulación SVPWM necesitaremos utilizar un sistema trifásico de referencia, V_{ref} , para generar el sistema trifásico deseado en la salida. Este vector es dinámico, ya que es la representación activa del sistema trifásico de referencia, por lo tanto, su proyección en el plano bidimensional α - β es de un vector que gira a una velocidad angular indicada, en sentido horario o antihorario como se ve en la *Figura 90*.

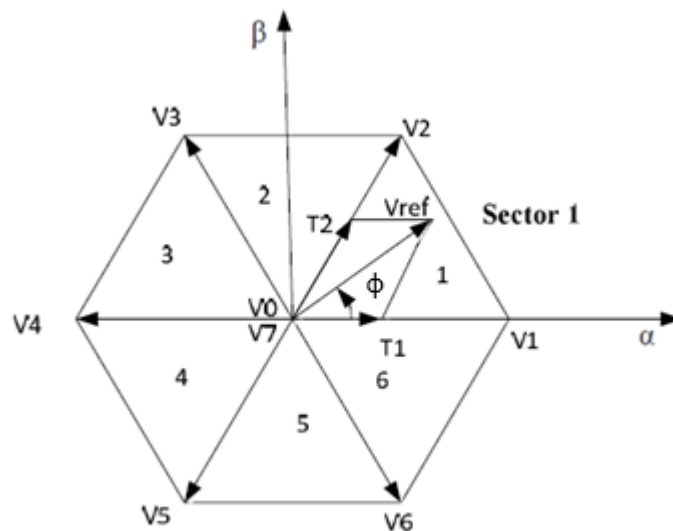


Figura 90. Hexágono vectorial con tensión de referencia V_{ref} .

El objetivo del control SVPWM, es activar el estado en los polos del inversor ($V_1, V_2, V_3...$) que se corresponda con el estado del vector referencia V_{ref} en cada instante, para ello es fundamental conocer el sector en el que se encuentra la referencia para seleccionar una de las cuatro posibles secuencias de conmutación que se encuentran dentro del mismo: V_0, V_7, V_x y V_{x+1} , donde x es el número del sector.

La selección se lleva a cabo a través de dos parámetros fundamentales:

La **proyección en α y en β** del vector referencia. Conseguir el valor de la proyección en cada momento del vector referencia es fundamental para conocer en qué sector se encuentra a cada instante.

Una proyección negativa en α indicaría una posición del vector entre los sectores 2, 3, 4 o 5, una proyección positiva una posición en los sectores 1, 2, 5 o 6. Con la proyección en β reduciríamos la búsqueda del sector pues el signo negativo indicaría un sector entre 4, 5 o 6, y el positivo un sector entre 1, 2 o 3.

Como lo que nos interesa dentro de cada sector es conocer que vector de conmutación asignar al inversor, va a ser necesario calcular otro parámetro más: **el ángulo ϕ** . En cada momento este ángulo cambia con respecto al eje α que vamos a considerar como referencia de 0° . Para conocer su valor es necesario utilizar la siguiente ecuación, que se corresponde con la del arco tangente, a partir de las proyecciones del vector referencia en β y en α .

$$\alpha = \tan^{-1} \left(\frac{V_\beta}{V_\alpha} \right) \quad (17)$$

La función inversa del arco tangente solo da como resultados ángulos entre 90° y -90° en función de los signos de las proyecciones, que pueden tener hasta un máximo de cuatro combinaciones:

- **α y β positivos:** En este caso el ángulo se mueve entre 0 y 90° y los sectores entre el 1 y el 2.
- **α negativo y β positivo:** Nos encontraríamos entre los sectores 2 o 3 y el ángulo cambiaría desde -90° hasta 0° (siempre que nos movamos en sentido antihorario).
- **α y β negativos:** Para dos proyecciones negativas el ángulo vuelve a ser positivo entre 0° y 90° , aunque en este caso se encontraría entre los sectores 4 y 5.
- **α positivo y β negativo:** Por último, el ángulo vuelve a ser negativo entre -90° y 0° y nos encontraríamos entre los sectores 5 y 6.

Con todo esto se puede dar lugar a la *Tabla 11*, en la que, conociendo el valor de la proyección del vector referencia y en función del ángulo (entre 90° y -90°) podemos conocer el sector en el que se encuentra el vector referencia, pero no solo eso, sino que para un mismo sector podemos hacer una selección del vector de conmutación correspondiente en función del valor del ángulo ϕ : mayor o menor de $\pm 30^\circ$ para los sectores 1, 3, 4 y 6 o 90° para los sectores 2 y 5.

Tabla 11. Vector y sector de referencia en función de la proyección α y β

Sector	Vector	Angulo ϕ	Proyección α	Proyección β
1	1	$0^\circ - 30^\circ$	+	+
	2	$30^\circ - 60^\circ$	+	+
2	2	$60^\circ - 90^\circ$	+	+
	3	$-90^\circ / -60^\circ$	-	+
3	3	$-60^\circ / -30^\circ$	-	+
	4	$-30^\circ - 0^\circ$	-	+
4	4	$0^\circ - 30^\circ$	-	-
	5	$30^\circ - 60^\circ$	-	-
5	5	$60^\circ - 90^\circ$	-	-
	6	$-90^\circ / -60^\circ$	+	-
6	6	$-60^\circ / -30^\circ$	+	-
	1	$-30^\circ - 0^\circ$	+	-

A partir de esta tabla podemos ver cómo, solo conociendo la proyección en todo momento del vector de referencia dinámico en el eje α y en el eje β , podemos conocer el vector que le corresponde activar al inversor en cada momento, siempre y cuando no queramos seleccionar un vector nulo.

Seleccionar un vector nulo como V0 o V7 dentro de un sector implicaría reducir el tiempo de aplicación de los vectores y por lo tanto tener la ventaja de aumentar la amplitud de la tensión de salida. No es de extrañar que entre dos vectores de conmutación entre el 1 y el 6 se añada un vector nulo para mejorar la amplitud de salida.

Cuantos más estados existan, más fiel será la tensión final de salida a la tensión de referencia, más senoidal, y por lo tanto más posibilidad habrá de reducir el espectro armónico, pero también más pérdidas por conmutación se van a producir y más complejos serán los cálculos, pues en un mismo espacio de tiempo van a haber más vectores y van a conmutar más polos [25][26][29][30].

Para conocer el funcionamiento de este control y sus principales ventajas hemos acudido de nuevo a la herramienta Simulink (aunque en este caso sin conexión a la tarjeta) donde podemos demostrar cómo se generan estos pulsos a partir del control SVPWM para dar lugar a una señal trifásica en la salida que tiene que seguir la referencia que marca el inversor. El circuito que utilizamos para simular este comportamiento es el de la *Figura 91*:

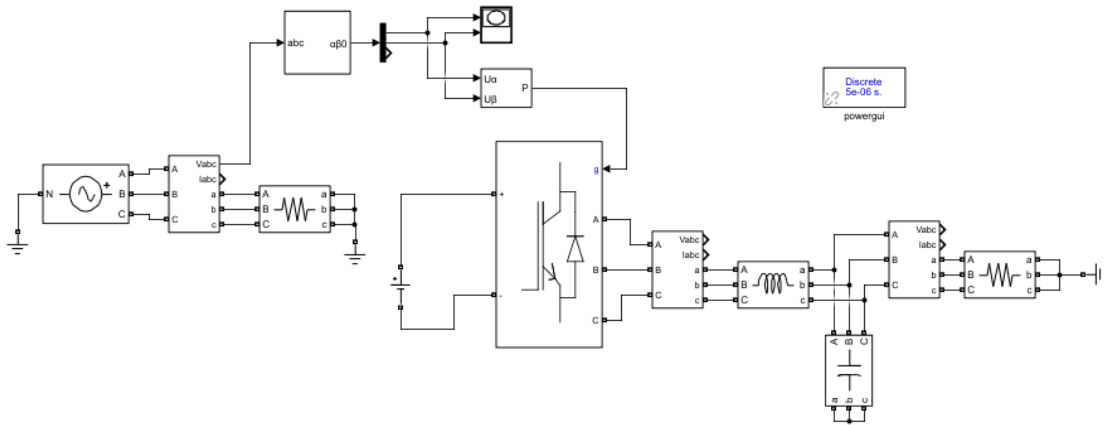


Figura 91. Esquema en Simulink con el objetivo de generar un control SVPWM

Con el esquema que implementamos en Simulink, lo que tratamos de hacer es generar la misma señal trifásica a través de un inversor que la que se da a partir de una fuente ideal¹⁷. Para ello se van a tomar las tensiones de fase y se va a realizar la transformada de Clarke sobre ellas, para obtener las dos señales ya estudiadas V_α y V_β . A partir de estas dos y gracias al bloque “SVPWM Generator” se generan los seis pulsos que servirán para controlar los polos del inversor para generar el sistema trifásico final.

Como hemos visto, el objetivo del control vectorial es seguir la referencia del vector resultante del sistema trifásico inicial a través de la generación de distintos valores de tensión en el inversor (vectores). Esto es en lo que nos vamos a centrar al analizar el resultado de la simulación con un tiempo de 0.02 segundos, que se corresponde con el periodo de la onda.

¹⁷ Los parámetros de esta fuente son frecuencia 50Hz y valor de tensión eficaz (V_{rms}) 220V

En primer lugar, tras terminar de simular el circuito, nos vamos a fijar en la referencia en el plano α - β que deja tras de sí el vector resultante del sistema trifásico inicial.

Como hemos estudiado en páginas anteriores y como se ve en la *Figura 92*, el vector referencia dibuja un círculo donde su radio se corresponde con el valor de pico de las tensiones de fase que en este caso es $179,5 \text{ V}^{18}$.

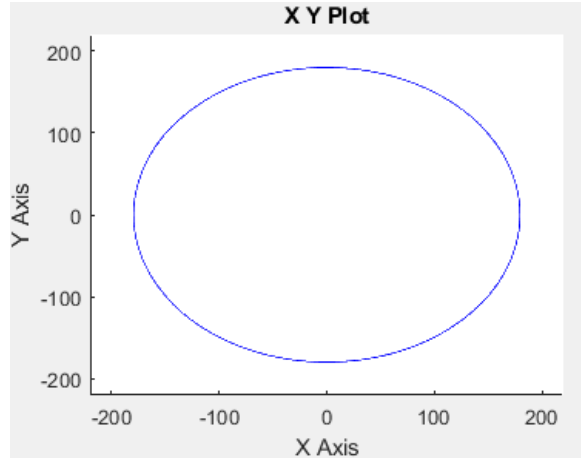


Figura 92. Referencia de sistema trifásico en plano α - β

En segundo lugar, nos vamos a fijar en el resultado de los pulsos generados tras el bloque "SVPWM Generator", sobre todo en los tres valores dados para cada una de las tres ramas del inversor (sin contar con la señal opuesta dirigida al polo de la misma rama) que son los que podemos vamos a analizar a partir de la *Figura 93*.

¹⁸ Resultado de $V_{rms}/0.707$ para obtener el valor de pico de la tensión de línea dividido a su vez por $\sqrt{3}$ para obtener este valor en la tensión de fase que es la que estamos estudiando.

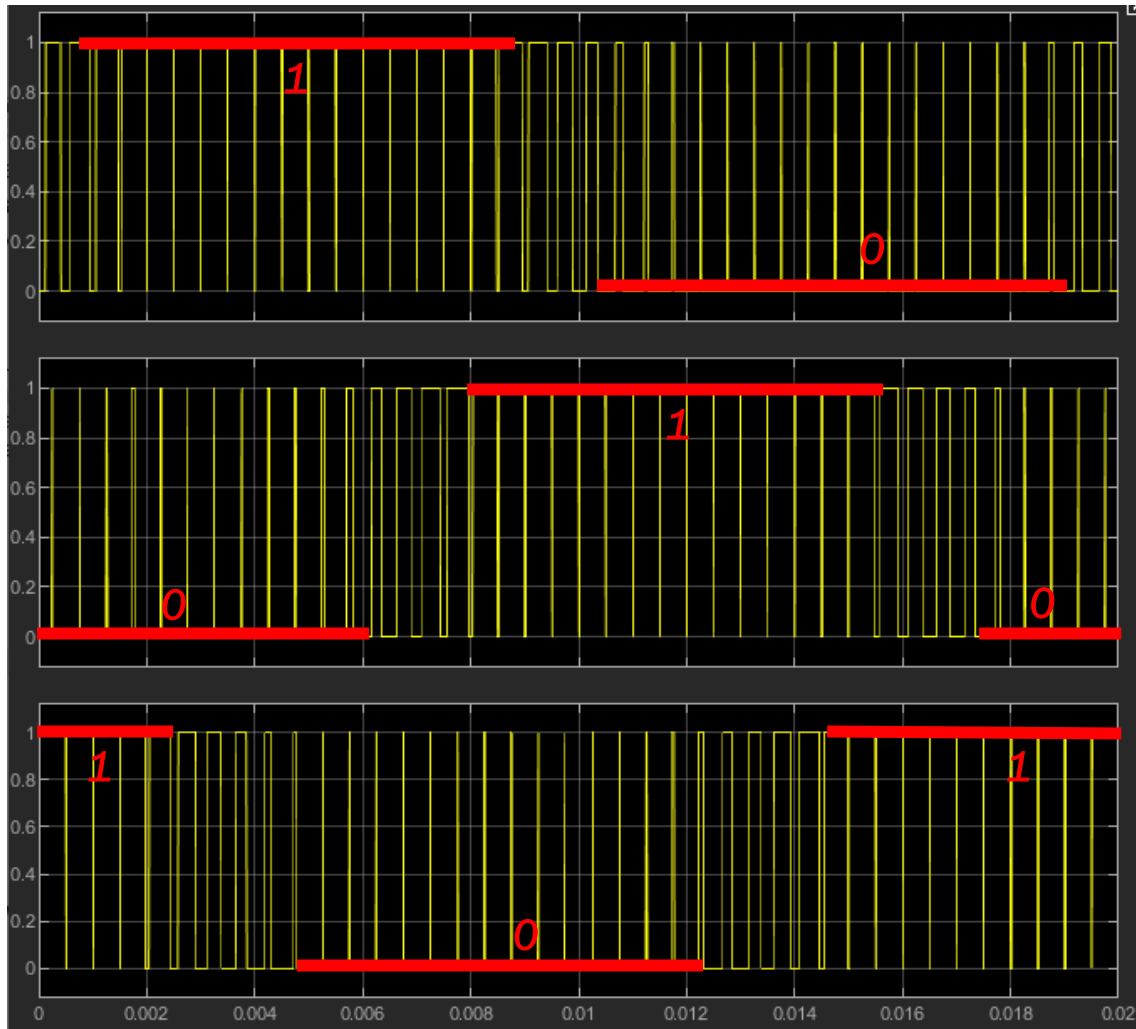


Figura 93. Valores de salida de pulsos generados por control SVPWM

Como podemos observar, en este periodo de tiempo de 0.02 segundos, las tres señales de control pasan por diferentes estados que generan unas tensiones de salida en cada rama del inversor, como ya apreciamos en el hexágono vectorial de la *Figura 89*. Siguiendo el hexágono de esta figura y comparándolo con los resultados de la simulación, podemos confirmar que la primera tensión que se genera en el inversor es V6 (101) y a partir de ahí se generan todas las demás en sentido antihorario: V1 (100), V2 (110), V3 (010), V4 (011) y V5 (001).

Algo también significativo en este caso es la sucesión de estados nulos (V0 o V7) que podemos observar intercalados entre los valores de salida reales. Podemos deducir que su aparición se debe al muestreo que lleva a cabo el PWM. En este caso la frecuencia de muestreo se corresponde con 2000 Hz o lo que es lo mismo 0,0005 segundos entre muestras, por lo que es lógico que cuando transcurra este tiempo se genere un estado nulo mientras se muestrea el siguiente valor correspondiente.

El último punto para analizar de esta simulación es la forma final de las corrientes y de las tensiones. En primer lugar, vamos a visualizar la salida de la tensión de fase V_a a la salida del inversor, antes de pasar por ningún filtro. En la misma gráfica, vamos a mostrar también la tensión V_a inicial para comprobar cómo se sigue la referencia al utilizar la modulación del vector espacio para controlar los polos del inversor. El resultado es el de la *Figura 94*.

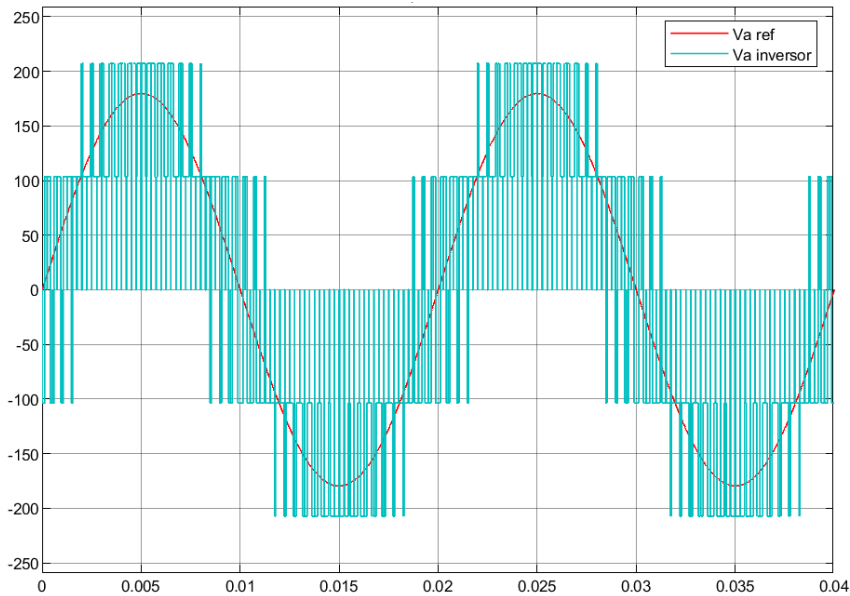


Figura 94. Tensión de fase V_a a la salida del inversor comparada con la referencia inicial

Como se puede ver en la imagen, y ya se podía suponer, la tensión a la salida del inversor no está filtrada, pero esto nos permite comprobar también como es la forma de onda generada por el inversor trifásico, donde se pueden apreciar tres niveles de tensión generados al tratarse de tres ramas las que son controladas. También podemos comprobar como la referencia de tensión inicial se sigue correctamente al usar la modulación SVPWM.

Al añadir un filtro LC a la salida del inversor vamos a reducir el espectro armónico considerablemente, acercándonos más a la forma de onda de referencia. En la *Figura 95* podemos visualizar todas las tensiones de fase del sistema trifásico a la salida del filtro comparadas con las de la referencia inicial.

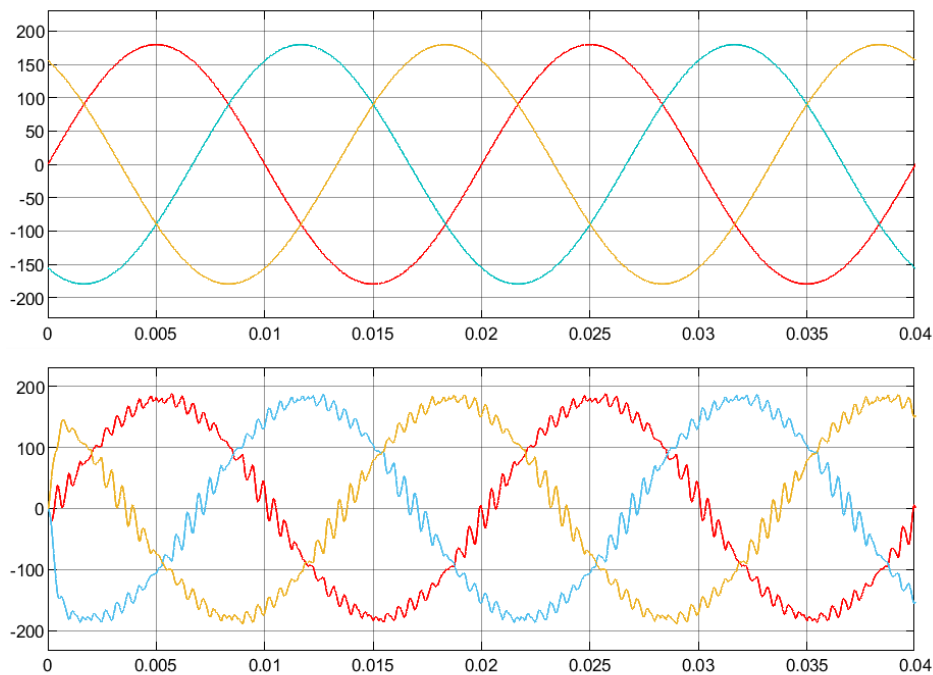


Figura 95. Tensiones de fase a la salida del filtro comparadas con el sistema trifásico de referencia

Es evidente, tras observar la figura, que el objetivo de la simulación se ha cumplido. Las tensiones filtradas a la salida del inversor siguen a la referencia inicial correctamente al hacer uso del control vectorial. Es cierto que existe cierto ruido y eso puede generar algunas interferencias en el posterior procesamiento de la señal, pero el resultado global se puede considerar satisfactorio.

En cuanto a las corrientes, podríamos decir lo mismo que hemos comentado para las tensiones, pues al tener una carga resistiva ideal con un valor unitario a la salida, la forma de onda para la tensión va a ser la misma que para la corriente.

Podemos concluir entonces, para terminar el apartado, que el control vectorial es la forma óptima de controlar un inversor trifásico. La sencillez de implementación, la reducción de armónicos y el óptimo seguimiento de la referencia son, entre otras, sus principales ventajas.

4. Desarrollo práctico del TFG

Tras estudiar el funcionamiento de la tarjeta LAUNCHXL-F28379D, los principios de programación de la herramienta en Simulink, y el estudio de una aplicación en electrónica de potencia basada en el control de un Inversor, es el momento de unir todos estos conocimientos para una aplicación real.

En este apartado vamos a explicar, en primer lugar, los objetivos iniciales de la parte práctica y como hemos tenido que desechar algunas ideas por la inviabilidad de las mismas hasta dar con la solución final. Después, nos vamos a centrar en explicar el objetivo de la aplicación y el algoritmo de control que utilizaremos. A partir de aquí obtendremos los resultados ideales en Simulink.

Por último, analizaremos como llevar el algoritmo de control a la práctica a través de la tarjeta LAUNCHXL-F28379D, que elementos electrónicos necesitaríamos para ejecutarlo y cómo van a ser las conexiones entre la tarjeta y el inversor para cumplir con las especificaciones de cada fabricante.

4.1 ANALISIS DE OBJETIVOS

Como hemos comentado en el apartado 2. *Desarrollo de tarjeta LAUNCHXL-F28379D y principios de programación en Simulink.*, una de las aplicaciones más interesantes de la tarjeta LAUNCHXL-F28379D es la posibilidad de utilizarla como generador de señales PWM, que, como hemos visto también en el apartado 3. *Aplicación en electrónica de potencia*, es una parte fundamental del control del inversor. En la *Figura 96* podemos observar mediante un esquema la idea del control inicial propuesta.

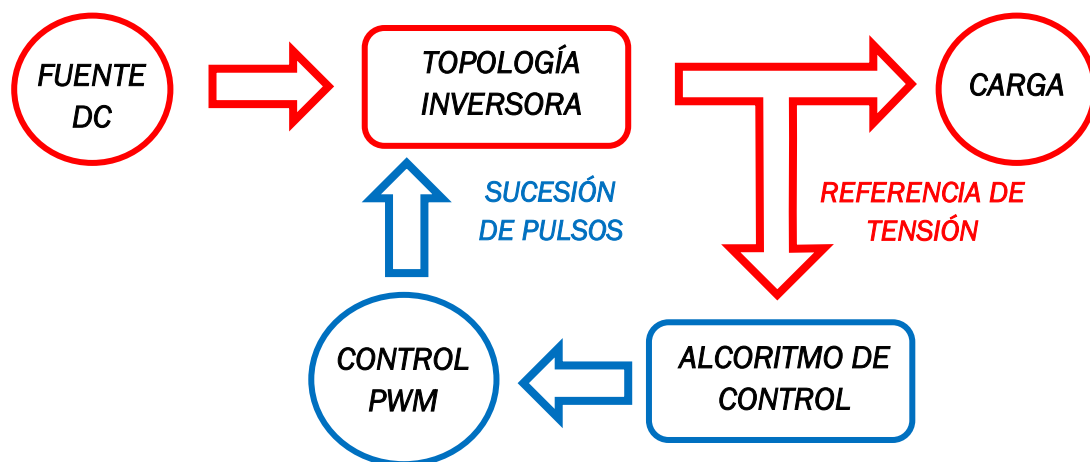


Figura 96. Esquema de objetivo inicial del proyecto

El objetivo sería poder generar los elementos en rojo a partir de la herramienta Simulink (evitando así la adquisición y puesta en marcha de los mismos) y a través de un puerto USB conectar el programa con la tarjeta (en color azul) donde este se

desarrollaría, se generarían los pulsos PWM a partir de las referencias de tensión y se mandarían al programa para cerrar el lazo de control en Simulink.

Llevar a cabo esta idea hubiera supuesto una nueva forma de simulación de cualquier circuito de potencia, que ahorraría la necesidad de poseer ningún elemento activo de para comprobar la efectividad del circuito. Únicamente serían necesarios tres elementos: Un ordenador, un cable USB y la tarjeta LAUNCHXL-F28379D.

El problema de poner en marcha esta primera idea fue la inviabilidad de poder mandar la secuencia de pulsos generada en la tarjeta hacia Simulink de nuevo y es que el bloque ePWM, que podemos ver en la *Figura 97*, permite programar pulsos a partir de una referencia, pero no es capaz de mandarlos de nuevo a Matlab. Tampoco existe ninguna opción dentro de los bloques de la tarjeta en Simulink que permita mandar la información desde un pin PWM a través de ningún puerto.



Figura 97. Bloque ePWM

Ante la imposibilidad de continuar por este camino, se decide cambiar el rumbo del proyecto y barajar otras opciones de desarrollo.

La segunda opción que proponemos consiste en generar pulsos PWM, sin devolverlos a la tarjeta, con el fin de conocer su comportamiento, con realimentación interna en el propio Simulink de estos mismos pulsos para no romper el lazo de control. El diagrama de bloques que vemos en la *Figura 98* muestra el desarrollo teórico de esta segunda opción de la misma manera que lo hicimos con la primera en la *Figura 96*.

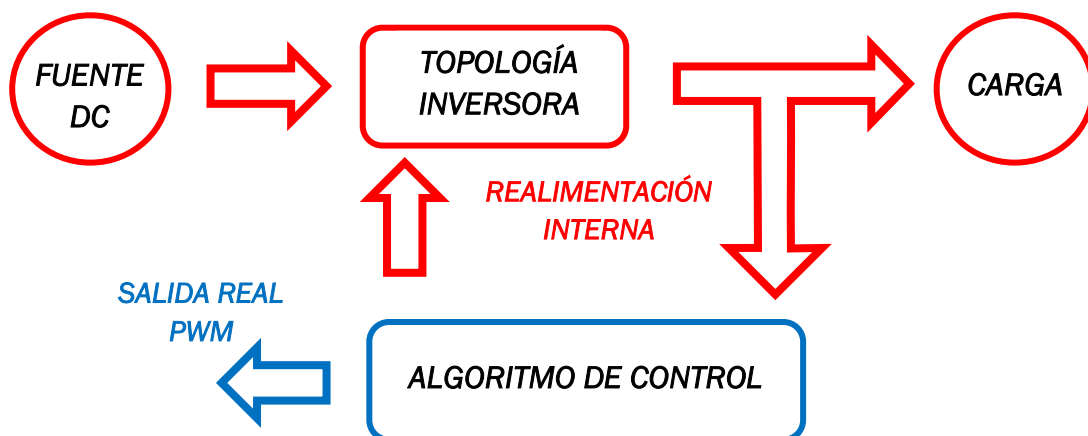


Figura 98. Esquema de segunda opción de desarrollo del proyecto

Esta alternativa nos permitiría desarrollar el proyecto de forma que lográramos obtener la referencia PWM al mismo tiempo que ejecutamos toda la parte de electrónica de potencia en el ordenador. El problema de llevar a cabo esta segunda opción erradicó en la imposibilidad de ejecutar en Simulink elementos de salida

binaria como son los polos de los inversores, que estaban controlados por la propia salida PWM realimentada en Simulink.

Este nuevo descubrimiento sirvió para darnos cuenta de que no es viable controlar un inversor ideal, simulado en un ordenador, a partir de una tarjeta externa. Es necesario disponer de él físicamente.

Con esto llegamos a la tercera y última opción que barajamos. Para desarrollar esta alternativa, al contrario que con las dos anteriores, tendríamos que hacer uso de un sistema completo de potencia en el que poder integrar el inversor. El proyecto se podría llevar a cabo con los elementos necesarios e **integrando en la tarjeta el algoritmo de control**, la salida PWM y la entrada de las referencias de tensión y corriente a partir del convertidor Analógico-Digital que lleva integrado la tarjeta (véase apartado 2.4.1 *Convertidor ADC*). El diagrama de bloques de esta solución es el que mostramos en la *Figura 99*.

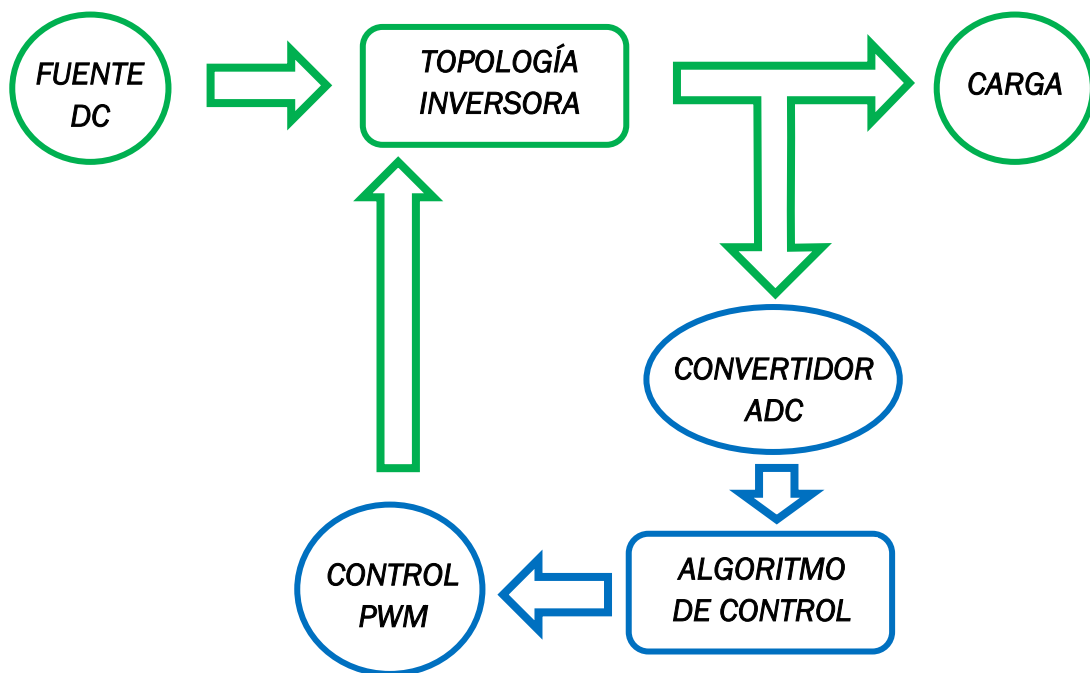


Figura 99. Diagrama de bloques de solución final para el desarrollo práctico del proyecto

En este caso podemos ver un cambio en los colores del diagrama con respecto a los otros casos, y es que en la *Figura 96* y en la *Figura 98*, utilizábamos el color rojo para señalar los elementos implementados en Simulink, mientras que en el diagrama de la *Figura 99* utilizamos el color verde para estos mismos elementos, ya que en este caso su implementación va a ser física. El color azul sigue señalando los procesos que se llevarían a cabo en la tarjeta LAUNCH-F28379D. Se puede apreciar que aquí hemos añadido el convertidor Analógico-Digital, que se incluye en la tarjeta, para procesar las señales de referencia que va a leer el algoritmo de control.

Conocida la opción final más viable, y la labor que va a desarrollar la tarjeta en esta, nos fijamos cumplir los siguientes objetivos:

- Explicar y simular la finalidad de la aplicación a través del algoritmo de control, que desarrollaremos en Simulink, para controlar el inversor.
- Conocer los resultados teóricos del control y comparar los dos tipos de modulación PWM posibles.
- Conocer los dispositivos necesarios para desarrollar el proyecto en un laboratorio industrial.
- Conocer las conexiones reales entre la tarjeta LAUNCHXL-F28379D, el inversor, y el resto de los elementos para llevar a cabo una implementación real de la aplicación cumpliendo las especificaciones que marca cada fabricante.

Para llevar a cabo todos los objetivos comenzaremos explicando, en el segundo apartado, la aplicación que le vamos a dar a la tarjeta, como la desarrollamos a través del algoritmo dq, y cuáles son los resultados teóricos en Simulink. En el tercer apartado veremos los elementos necesarios para llevar a cabo el proyecto, las especificaciones técnicas y como conectarlos entre ellos. Por último, indicaremos también el programa final desarrollado en Simulink que la tarjeta LAUNCHXL-F28379D llevará a cabo.

4.2 APLICACIÓN FINAL. ALGORITMO DE CONTROL DQ

En este apartado vamos a centrarnos en el papel de la tarjeta de desarrollo LAUNCHXL-F283792 en el proyecto. Para ello, vamos a explicar en primer lugar, el algoritmo de control dq que vamos a programar en la tarjeta, y cuál es su propia finalidad. Después veremos su desarrollo en Simulink y, por último, vamos a analizar los resultados teóricos del mismo, simulando la solución final en Simulink.

4.2.1 Algoritmo de control dq. Objetivos

Existen variedad de algoritmos para el control de un inversor que se han ido desarrollando por distintos fabricantes de microcontroladores como el que estamos analizando en este proyecto. El objetivo general de todos ellos es diseñar una serie de ecuaciones, que van transformando las señales resultantes de comparar la referencia inicial con las referencias externas, para dar lugar a una sucesión de pulsos que controlan los polos del inversor y por lo tanto las tensiones y corrientes de salida.

En este proyecto vamos a utilizar el algoritmo de control dq. Este algoritmo es ampliamente utilizado en aplicaciones de gestión de fuentes de energía alternativa ya sea eólica o fotovoltaica, y es que **el objetivo final es el control de la inyección de**

potencia tanto activa como reactiva hacia la red. Esta es la aplicación final que le vamos a dar a nuestra tarjeta.

Para poner en marcha el algoritmo dq es fundamental, por un lado, tener una referencia constante tanto de la potencia activa como de la reactiva que queremos inyectar (valor asignado por el usuario), y por otro lado la potencia real que se está inyectando, que tiene que ser comparada con la referencia.

Para obtener la cantidad de potencia activa y reactiva que se está inyectando es muy útil proyectar el vector referencia de corriente labc en los ejes α - β ya que es muy significativo por la relación de la corriente con la carga¹⁹.

Con la transformada de Clarke del vector de referencia labc, el valor I_α se correspondería con la potencia activa y el de I_β con la potencia reactiva. Estos valores son variables con el tiempo y cambian a cada instante, por lo que, para poder compararlo con una referencia constante es necesario utilizar la transformada de Park para dar lugar a dos referencias invariables con el tiempo: I_q (Reactiva) e I_d (Activa). Con esto obtenemos dos **secuencias de control paralelas** una para el control de cada tipo de potencia. Estas referencias I_d y I_q son las que se van a comparar con la referencia constante de corriente activa (I_{act}) y reactiva (I_{react}) que deseamos inyectar a la red.

Una vez que la diferencia entre ellas es tratada por un controlador PI, aplicamos la *Teoría del marco de referencia síncrono* para obtener las ecuaciones del algoritmo dq, que nos permiten calcular el valor de las nuevas I_d^* e I_q^* a partir de la realimentación.

$$\bullet I_d^* = R \cdot I_d + L \cdot dI_d/dt - \omega \cdot L \cdot I_q \quad (18)$$

$$\bullet I_q^* = R \cdot I_q + L \cdot dI_q/dt + \omega \cdot L \cdot I_d \quad (19)$$

Conociendo I_d^* e I_q^* y aplicando la transformada inversa de Clarke y de Park podemos conocer el nuevo vector de referencia labc que puede ser tratado por los dos tipos de generadores PWM que conocemos, tanto el vectorial como el de ancho de pulso.

El paso por el generador PWM nos va a dar como resultado los pulsos que vamos a necesitar para controlar los polos del inversor y así cerrar también la realimentación del control.

En la *Figura 100* podemos observar un diagrama de bloques en el que esta resumido todo el algoritmo de control que vamos a implementar en la tarjeta con el fin de

¹⁹ El desfase que tiene la onda de intensidad con respecto a la tensión viene dado en función de la naturaleza de la carga y es el que origina la energía reactiva.

controlar la potencia reactiva y la reactiva inyectadas a la red a través de las referencias i_{act} e i_{react} [31][32][33].

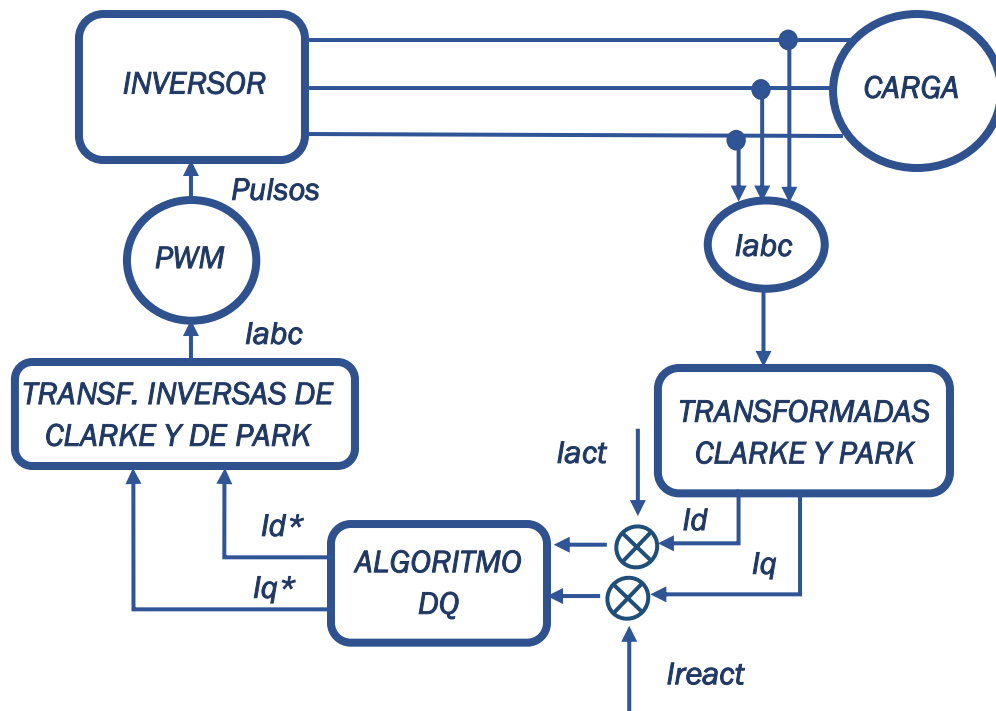


Figura 100. Algoritmo de control del Inversor

4.2.2 Desarrollo en Simulink

Una vez que conocemos el algoritmo de control que queremos implementar es el momento de generar toda la solución final explicada en la Figura 99 a través de la herramienta Simulink para obtener los resultados teóricos de la misma y el correcto funcionamiento del algoritmo de control. Para ello hemos dividido el esquema de la simulación en tres fases: Etapa de potencia, transformadas y algoritmo dq.

4.2.2.1 Etapa de potencia

En esta primera parte hemos simulado el comportamiento de un inversor trifásico, del filtro y de la carga que lo acompaña.

Como podemos ver en la Figura 101, hemos generado el inversor a partir de un subsistema (el contenido lo reflejamos en la Figura 102) que conecta fuente de tensión de continua, los pulsos generados por el PWM, y las tres fases generadas a la salida por cada rama del inversor. Añadimos también el filtro que se corresponde con un LCL adecuado para la gran cantidad de potencia, pero difícil de implementar en la realidad por su tamaño. También medimos los parámetros de corriente que nos serán útiles para la etapa de control. El sistema termina en una simulación de la conexión a la red.

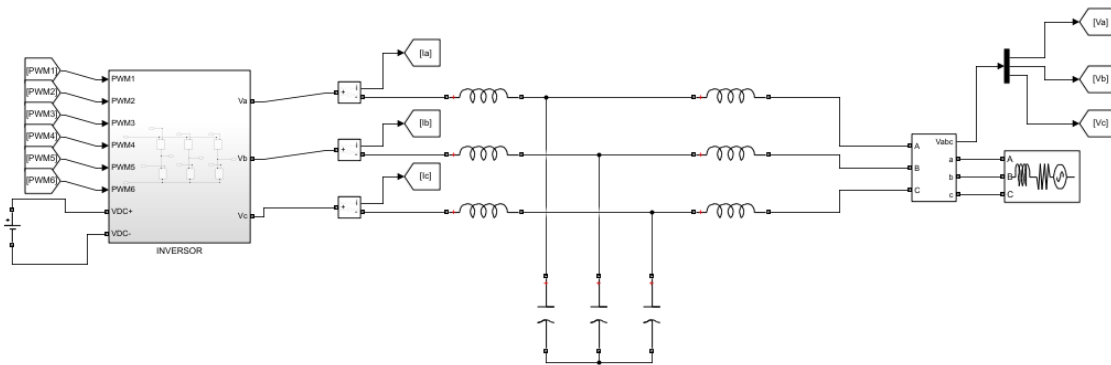


Figura 101. Simulación de la etapa de potencia en Simulink

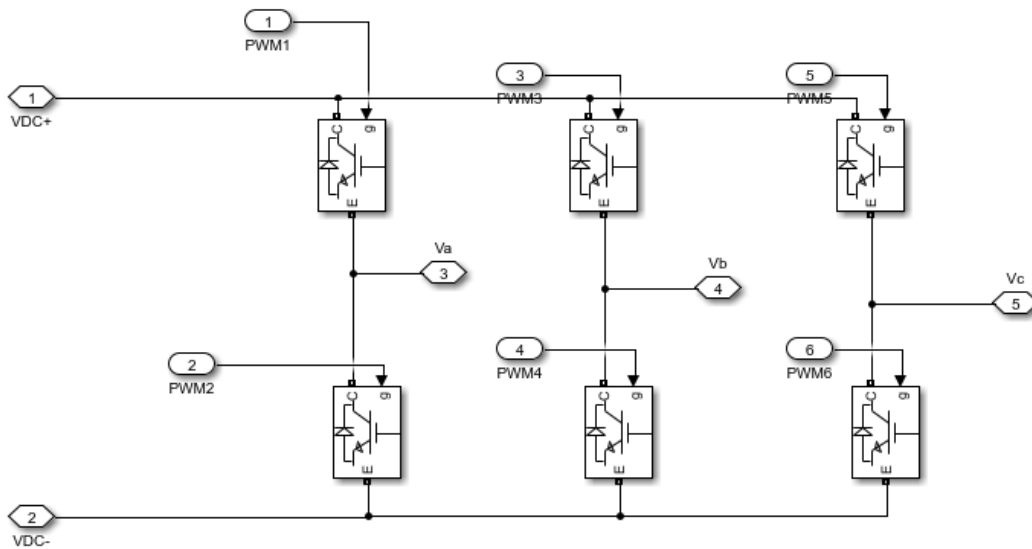


Figura 102. Simulación de inversor trifásico en Simulink

4.2.2.2 Transformadas

Esta etapa podría considerarse como la primera parte de la etapa de control, ya que, gracias a ella vamos a convertir la salida trifásica de corriente en una referencia constante que procesa el algoritmo de control. El esquema es el que se indica en la Figura 103.

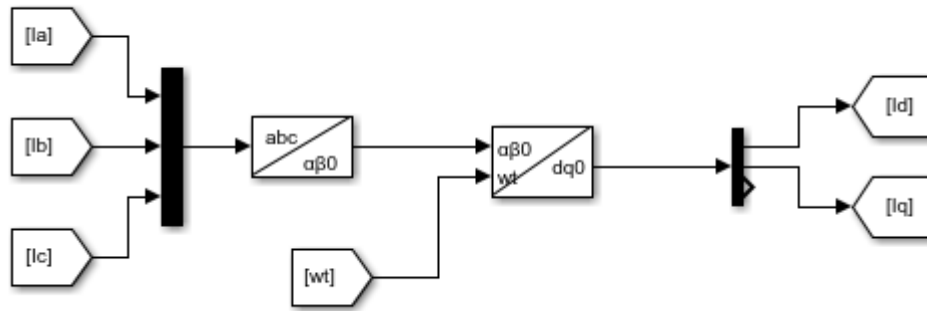


Figura 103. Simulación de transformadas en Simulink

Como podemos ver, hacemos uso de las transformadas de Clarke y de Park para obtener las referencias de I_d e I_q a partir de las referencias trifásicas de la etapa de potencia. Para la transformada de Park, además, necesitamos la velocidad angular ωt , que obtenemos a través de otra simulación secundaria a partir de las referencias de tensión V_a , V_b y V_c .

4.2.2.3 Algoritmo dq

Por último, y para terminar con la etapa de control y cerrar la realimentación del circuito, implementamos el algoritmo de control dq que utilizamos para obtener las nuevas referencias de tensión I_d^* y I_q^* .

A partir de ahí y aplicando las transformadas inversas, obtenemos la nueva referencia labc que se transformará en 6 salidas PWM a través del control por ancho de pulso o vectorial. En el caso del primer esquema, el de la Figura 104, utilizamos la modulación por ancho de pulso que obtiene la salida PWM a partir de la comparación de las señales de tensión con una señal triangular.

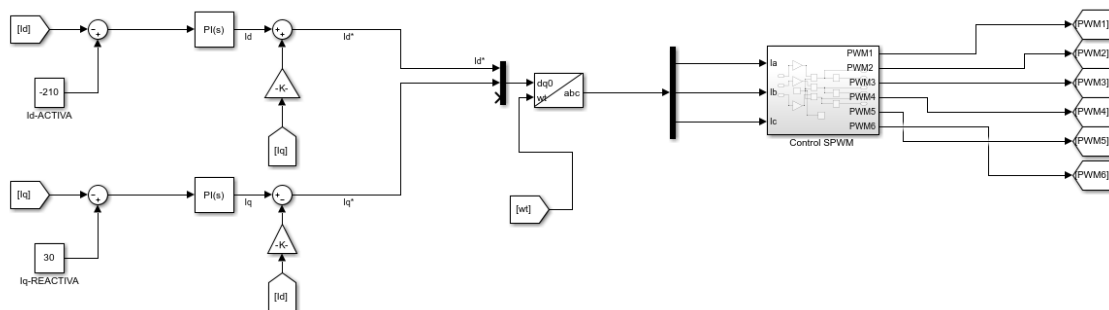


Figura 104. Algoritmo dq con control por ancho de pulso en Simulink

En cuanto al algoritmo, y tal y como avanzamos en el diagrama de la *Figura 100*, las entradas son I_d e I_q para dar lugar en la salida a una nueva referencia I_d^* e I_q^* . La diferencia principal de este algoritmo con el algoritmo inicial, de las ecuaciones del algoritmo dq, es que no existe el término que multiplica R ni el que multiplica L pues no contamos con ningún elemento activo en esta parte de la simulación.

Si quisiéramos implementar el control vectorial (SVPWM) sería tan fácil como coger la referencia a partir de la transformada de Clarke utilizando el bloque ya comentado en apartados anteriores de “SVPWM Generator (2-Level)” que obtiene directamente los seis pulsos de salida como se indica en la *Figura 105*:

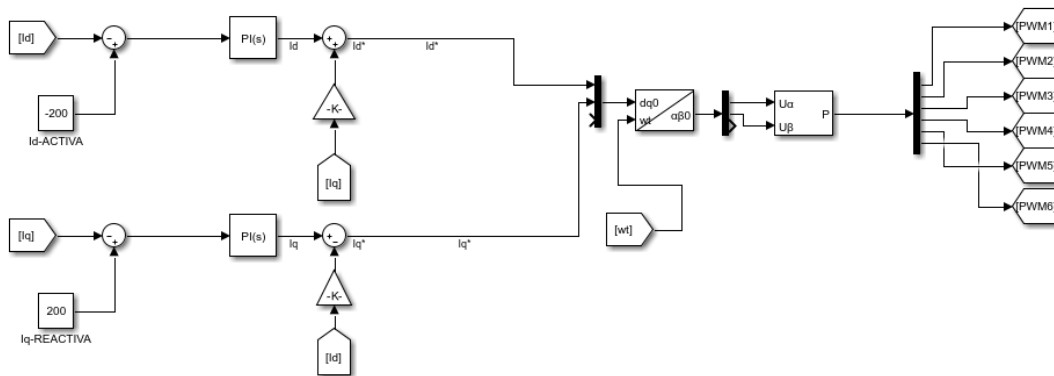


Figura 105. Algoritmo dq con control vectorial en Simulink

Los seis valores resultantes PWM se envían directamente a la entrada del inversor cerrando de esta forma la realimentación del control.

4.2.3 Resultados teóricos

Para terminar con este apartado de implementación del algoritmo de control en Simulink, vamos a dedicar este último punto mostrando los resultados de tensiones y corrientes que se inyectan a la red, ya sea a partir de un control por ancho de pulso (SPWM) o a partir de un control vectorial (SVPWM).

Como ya comentamos en el apartado 3.2 CONTROL DEL INVERSOR, el uso del control vectorial es deseable en sistemas trifásicos, o de gestión de altas potencias, por la simple razón de evitar muchas pérdidas en conmutación, gracias a la considerable reducción de los armónicos que se lleva a cabo en comparación con el control SPWM. Pues bien, esto mismo se puede observar en la *Figura 106*. Salida de corriente en simulación con control de ancho de pulso y con control vectorial. cuando comparamos la salida de la corriente en cada una de las dos simulaciones:

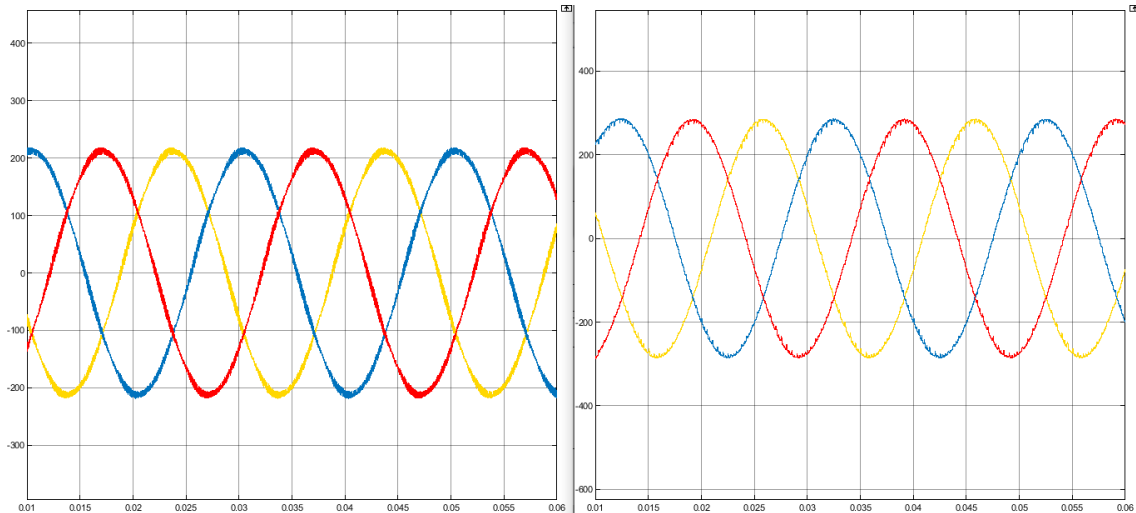


Figura 106. Salida de corriente en simulación con control de ancho de pulso y con control vectorial.

Aunque los valores de pico y la corriente no cambian, es muy notable la reducción del rizado al usar el control vectorial. En cuanto a la tensión no se aprecia apenas diferencia entre los dos casos y es que es a la corriente a la que más afecta el cambio de pulsos en los polos del inversor.

Una vez analizadas las diferencias entre los dos tipos de control pasamos a comprobar el funcionamiento correcto del algoritmo de control. Para ello cambiaremos tanto la referencia de i_{act} como la de i_{react} que van a dar lugar a una inyección mayor o menor de potencia activa o reactiva a la red.

Antes de comenzar con el análisis en Simulink, vamos a hacer un breve repaso a los conceptos de potencia activa, reactiva, y potencia instantánea:

- **Potencia instantánea:** Resultado de la multiplicación de la onda de tensión por la onda de corriente en cada momento, **si estas dos están en fase**, la potencia aparente será una onda con el doble de frecuencia de la tensión que, además, nunca toma valores negativos. Es la potencia que realmente absorbe la carga, la potencia activa y la reactiva serán resultados de la descomposición de esta misma.
- **Potencia reactiva:** Componente de la potencia instantánea que aparece cuando existe desfase entre las ondas de tensión y corriente debido a la existencia de cargas capacitivas o inductivas. Su valor máximo se situaría a 90° (carga inductiva pura) o -90° (carga capacitiva pura) de la onda de potencia aparente. No genera trabajo útil al ser siempre su componente continua 0.

- **Potencia activa:** Componente de la potencia instantánea en fase con la tensión, generada por los elementos resistivos. Un mayor desfase entre tensión y corriente (aparecen elementos reactivos) la hace desaparecer progresivamente, y un mayor valor de la carga resistiva la va a hacer crecer. Esta potencia es la que genera el trabajo útil, ya que su componente continua cambia desde su máximo con desfase 0° (carga resistiva pura) hasta un valor nulo con cargas capacitivas o inductivas puras.

Como vemos la aparición o desaparición de potencia activa o reactiva depende directamente de la carga, que afecta al desfase entre tensión y corriente.

En el caso que queremos implementar **no podemos controlar los valores de la carga, pero si los de la corriente**, por eso, a partir de la componente I_{react} controlamos el desfase de la corriente con la tensión, que genera potencia reactiva, y a partir de la componente I_{act} controlamos la amplitud de la corriente, que genera potencia activa.

En la *Figura 107* mostramos el primer caso que vamos a analizar, en el que **inyectamos solo activa**, con un valor de I_d de 200 A. Como vemos, al no inyectar reactiva, la tensión y la corriente (primera gráfica) están en fase, además el valor de pico de la corriente se corresponde con el de la referencia I_d de 200 A. La potencia instantánea (tercera gráfica) es siempre positiva, que es señal de inyección únicamente de potencia activa a la red.

La segunda gráfica muestra los valores de potencia activa y reactiva que se inyectan a la red, y que obtenemos gracias al bloque "Power (3ph, Instantaneous)" de Simulink. Este valor se corresponde con el del módulo del vector que representan: P y Q (activa y reactiva).

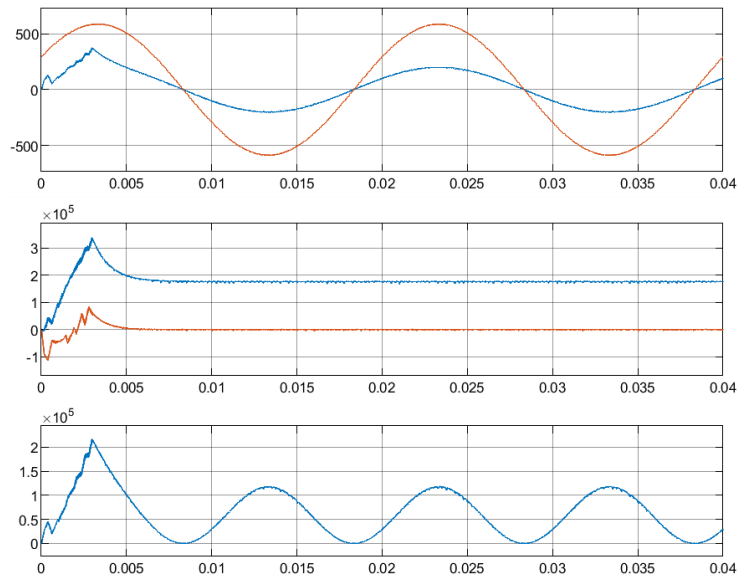


Figura 107. Corriente y tensión, potencia activa y reactiva, y potencia instantánea al inyectar $I_{act}= 200A$ e $I_{react}= 0 A$

Otro caso que se puede dar es el de **inyectar únicamente potencia reactiva** a través de la componente I_{react} . En la *Figura 108* podemos ver las consecuencias de esta inyección a través del desfase entre tensión y corriente, que podría considerarse como de 90° con un valor en la amplitud como el de I_{react} de 200 A, y a través de la forma de onda de la potencia instantánea, donde podemos ver que durante un periodo se toman los mismos valores positivos que negativos, lo que indica un trabajo útil nulo y una potencia media de 0 W entregados. Esta aplicación solo sería útil para compensar la energía reactiva en el punto de la red donde se inyecta.

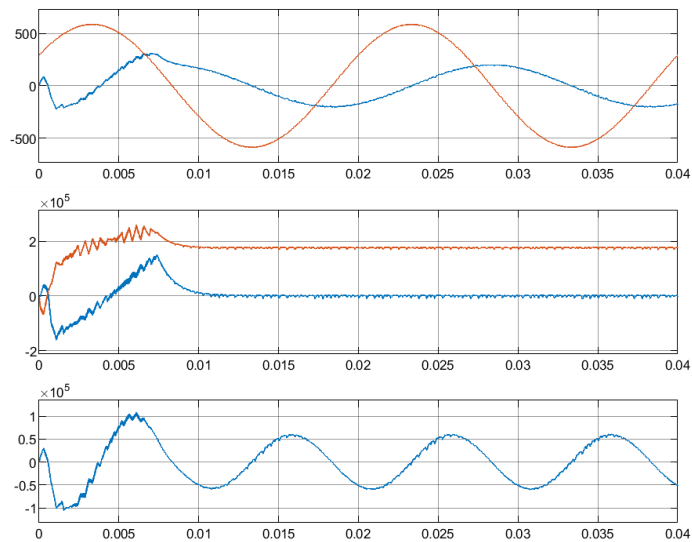


Figura 108. Corriente y tensión, potencia activa y reactiva, y potencia aparente al inyectar $I_{act}= 0 A$ e $I_{react}= 200 A$

El último caso que vamos a analizar es el de **inyección tanto de potencia activa como reactiva** a la red. Para ello nos valemos del mismo valor tanto para I_{act} como para I_{react} de 200 A, que sería lo mismo que conectar una carga RL con la misma impedancia en cada elemento. Vemos en la *Figura 109* como la tensión y la corriente siguen lejos de estar en fase pero que esta inyección de componente I_{react} no permite tampoco un desfase tan grande como el de 90° que hemos visto antes. Los valores de activa y reactiva son los mismos, pero la potencia instantánea se mantiene lejos de un valor medio igual a cero, aunque ya recoge valores negativos.

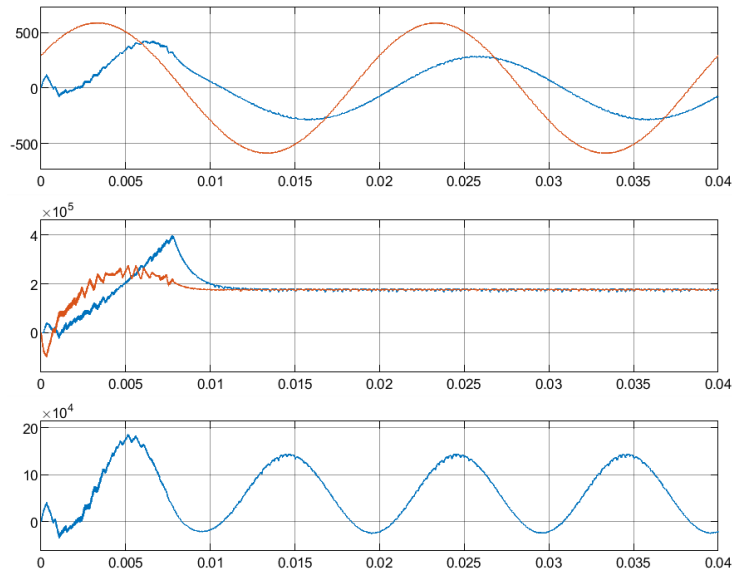


Figura 109. Corriente y tensión, potencia activa y reactiva, y potencia aparente al inyectar $I_{act}= 200$ A e $I_{react}= 200$ A

Si tuviéramos el sistema conectado a una carga y se dan valores menores que cero en la potencia instantánea, se diría que la carga está cediendo potencia, mientras que, si estos son positivos, la carga absorbe potencia. Con una carga completamente inductiva o capacitiva, esta cedería en un mismo periodo la misma potencia que absorbe, por eso se dice que no existe ningún trabajo útil en ese caso.

4.3 IMPLEMENTACIÓN REAL

Así como al empezar el apartado 4.2 APLICACIÓN FINAL. ALGORITMO DE CONTROL DQ, indicábamos que íbamos a centrarnos en el papel de la tarjeta LAUNCHXL-F28379 en el proyecto, en este apartado vamos a estudiar el resto de los elementos que van a acompañar a la tarjeta, sus características técnicas y como conectarlos para desarrollar la aplicación. Por último, mostraremos también el programa final en Simulink que la tarjeta deberá procesar para llevar a cabo correctamente la implementación real del proyecto.

Es necesario aclarar aquí que todos los elementos que se analizan en este apartado no han sido adquiridos, la idea es dar una visión general del desarrollo práctico más viable del proyecto. La adquisición y puesta en marcha de todos estos elementos podría ocupar perfectamente el desarrollo de otro proyecto a continuación de este.

4.3.1 Elementos físicos necesarios

En primer lugar, vamos a analizar en este punto los diferentes elementos que vamos a necesitar para llevar a cabo el desarrollo real del proyecto.

El esquema de la *Figura 99* nos servirá de principal apoyo, pues trataremos de buscar en el mercado los componentes que se correspondan con los elementos indicados en verde. Pero no todos los aparatos que necesitamos se muestran en el esquema, también será necesario analizar otros elementos como los sensores y acondicionadores de señal que son igual de importantes pero que no tienen cabida en la *Figura 99*.

4.3.2.1 Fuente DC y carga

Dos de los elementos más importantes de cualquier aplicación con inversores son los que van a fijar el comportamiento del mismo externamente, sin ser afectados apenas por el lazo de control. Estos son: La fuente de tensión de continua y la carga externa. A continuación, nos vamos a fijar en las funciones de cada uno dentro del proyecto, y analizaremos que tipo de artículo en el mercado nos puede dar las mejores prestaciones.

- Fuente DC

Como ya conocemos, para el correcto funcionamiento del inversor es necesario poseer una fuente de tensión constante para alimentar el sistema de polos de potencia del inversor. Podemos obtener esta referencia de dos formas diferentes:

1. Fuente de alimentación continua

Este es un elemento fácil de encontrar en todos los laboratorios. Tiene las **ventajas** de poder manejar el nivel de tensión deseado fácilmente, y de su sencillez para implementarlo, pues no ocupa un espacio excesivo, es fácilmente transportable y no demasiado caro. Pero también tiene sus

inconvenientes, y es que tiene que estar conectado siempre a una fuente de tensión alterna que tendrá que aportar valores mayores cuanto mayor sea la tensión en continua que queramos generar.

En la *Figura 110* podemos ver el modelo de una fuente de alimentación conmutada de alta potencia del fabricante **Elektro-Automatik**, que incorpora una señal continua de salida regulable entre 0V y 1500V, siempre y cuando la alimentación en alterna se encuentre en un rango entre 340 y 460V. La fuente está controlada por un microcontrolador de alto rendimiento y cuenta con display digital y puertos adicionales USB y Ethernet [34].



Figura 110. Fuente de alimentación conmutada de alta potencia del fabricante Elektro-automatik.

2. Generador a partir de fuentes de energía externas

Otra opción para obtener tensión continua es a partir de fuentes de energía externas, donde la más conocida por ser a su vez la más limpia y barata, es la energía solar, que transforma la luz, que recibe un panel fotovoltaico, en tensión continua que se puede procesar.

Para llevar a cabo esta generación de tensión, son necesarios algunos elementos, además del panel solar. Hace falta también un banco de baterías y un regulador de carga, como mínimo, no es sencilla su implementación, como podemos observar en la *Figura 111* y eso es **su mayor desventaja** para el desarrollo en un laboratorio.



Figura 111. Esquema de generación de tensión a partir de un panel fotovoltaico

En conclusión, la mejor opción para generar un valor de tensión constante en continua es a partir de una fuente de alimentación conmutada de alta potencia. Es un elemento robusto, no demasiado grande, que cumple suficientemente con las expectativas marcadas. La generación de continua a partir de paneles solares es la opción más rentable, pero para un uso industrial, donde se pueda hacer negocio con los excedentes vertidos a la red. Para un uso didáctico no es la opción más conveniente.

- Carga

Otro elemento fundamental para el correcto desarrollo del proyecto es la carga final, hacia donde se dirige la tensión y la corriente generadas. Como ya comentamos en el apartado 4.2 APLICACIÓN FINAL. A, el objetivo final del proyecto es el del control de inyección de potencia a la red, por lo tanto, nuestra carga no puede ser otra que la red eléctrica.

En un laboratorio es difícil conectar un sistema directamente a la red, por lo tanto, existe la posibilidad de simular esta conexión a partir de elementos como el REGATRON Serie TopCon (véase la *Figura 112*) del fabricante **ACS** que funciona como un simulador de alta eficiencia de la red a partir de la tecnología doble inversor multinivel. Otros parámetros destacados son un rango de potencia de simulación desde 50kVA hasta 700kVA y una tensión máxima de entrada de hasta 280V. Además, este simulador incluye un novedoso software de simulación de la red: GridSim [35].



Figura 112. Simulador de red REGATRON Serie TopCon

4.3.2.2 Sensores y acondicionadores

Los sensores son elementos fundamentales en aplicaciones de electrónica de potencia, por ello es muy importante conocer las principales características del sensor que vayamos a utilizar.



Figura 113. Transductor de corriente de la familia de fabricantes LEM

En este caso nos hemos decantado por un tipo en especial, los transductores de corrientes basados en el efecto Hall, que se venden en el mercado por parte del fabricante **LEM**. Este tipo de transductores se diferencian del resto al utilizar el efecto Hall para medir corrientes, y eso es lo que los hace tan útiles pues no ocupan apenas espacio y permiten conocer el valor de la corriente al ser atravesados por el conductor que la lleva.

La principal desventaja de este tipo de sensores es su necesidad de estar conectados siempre a una tensión continua (que varía dependiendo del modelo).

Para el modelo de la *Figura 113* (IT-205-S) podemos conocer gracias a su data-sheet que tiene un rango de medida máximo de corriente en alterna de hasta ± 849 A, tiene la posibilidad de medir en continua también, y la tensión de referencia continua es 15,75V como máximo [36].

Los sensores de tensión, también de la familia LEM, no utilizan el efecto Hall, y necesitan la referencia de una corriente proporcional a la tensión medida que se obtendría a partir de una resistencia.

Es el caso del transductor de tensión LV-25 P (véase *Figura 114*) que acepta un valor máximo de corriente de entrada de 10mA, y un valor máximo de la tensión de salida de 500V. El principal problema de este sensor es que es necesario montarlo en una PCB, divergiendo esto con la comodidad que nos aportaba el sensor de corriente, aun así este transductor se convierte en nuestra primera opción a la hora de seleccionar en el mercado [37].



Figura 114. Transductor de tensión de la familia de fabricantes LEM

Otro problema que surge al intentar implementar estos sensores es la conexión con la tarjeta LAUNCHXL-F28379D, y es que, como nos señala el fabricante y hemos podido ver en el apartado 2.1 ALIMENTACIÓN, el rango de tensión máxima de entrada (V_{refhi}) para el convertidor ADC va desde 2.4V hasta el valor máximo de V_{DDA} que es 3.3V. Es la Tabla 12 la que nos indica estos valores:

Tabla 12. Valores de rangos analógicos permitidos en el convertidor ADC de la tarjeta LAUNCHXL-F28379D.

	MIN	TYP	MAX	UNIT
ADCCLK (derived from PERx.SYSCLK)	5		50	MHz
Sample window duration (set by ACQPS and PERx.SYSCLK) ⁽¹⁾	320			ns
V_{REFHI}	2.4	2.5 or 3.0	V_{DDA}	V
V_{REFLO}	V_{SSA}	0	V_{SSA}	V
$V_{REFHI} - V_{REFLO}$	2.4		V_{DDA}	V
ADC input conversion range	V_{REFLO}		V_{REFHI}	V
ADC input signal common mode voltage ⁽²⁾⁽³⁾	$V_{REFCM} - 50$	V_{REFCM}	$V_{REFCM} + 50$	mV

Fijándonos entonces en la tabla, el rango de entrada del convertidor se podría considerar entre 3V y 0V. Como podemos deducir, no podemos conectar a la tarjeta directamente la salida del transductor de tensión de un valor máximo de $\pm 500V$. Va a ser necesario hacer uso de un **acondicionador de señal** antes de conectar a la tarjeta el transductor de tensión.

Estos elementos acondicionadores se venden en el mercado con el objetivo de proporcionar al convertidor ADC: adaptación del rango y protección. Para ello se componen de tres diferentes etapas:

- **Etapa de normalización del rango:** Esta primera etapa va a ser la encargada de reducir los valores del rango de tensión iniciales a unos mucho más asequibles. Para ello, es necesario disponer de un circuito de amplificación (basado en Amplificadores Operacionales) que nos permita añadir una ganancia menor que 0 a la señal. Con estas condiciones solo podemos hacer uso de un circuito inversor²⁰ como el de la Figura 115:

²⁰ El circuito no inversor no permite ganancias menores de 1

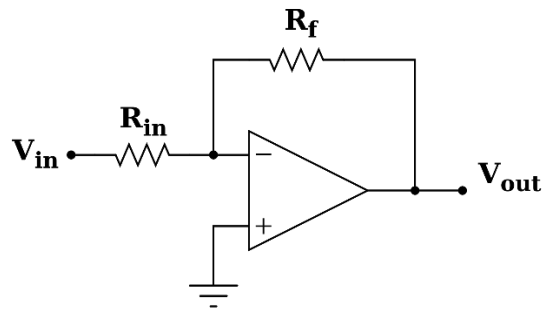


Figura 115. Circuito inversor

Donde la tensión de salida depende de la relación entre la resistencia R_f y R_{in} ya que se cumple que:

$$V_{out} = -V_{in} * \frac{R_f}{R_{in}} \quad (20)$$

Por lo tanto, con un valor de R_f inferior al valor de R_{in} podremos reducir el valor de la señal el rango que veamos conveniente. Tras este circuito sería necesario añadir otro circuito no inversor con ganancia 1 para devolver a la señal reducida su signo inicial.

En nuestro caso, nos vamos a encontrar con tensiones de $\pm 500V$ a la entrada del amplificador, que tendremos que reducir a un rango de $\pm 1,5V$, por lo tanto, necesitaremos una ganancia de 0,003.

- **Etapa de adaptación:** En esta etapa, adaptamos el rango reducido de entrada al rango que corresponda para la aplicación que vamos a dar a la señal, en este caso, entrada de un convertidor Analógico-Digital, por lo tanto, necesitaríamos una salida variable entre 0 y 3V. Para lograr esta adaptación la mejor solución es utilizar un circuito sumador como el de la *Figura 116* en el que añadiremos la tensión máxima de referencia a la salida reducida de la anterior etapa.

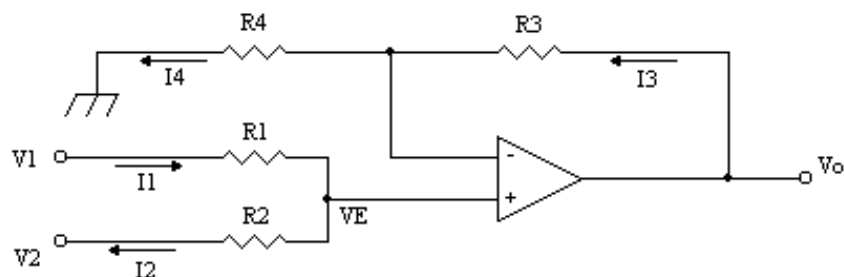


Figura 116. Circuito sumador no inversor

De esta forma, siguiendo la siguiente ecuación y equilibrando los valores de las resistencias, podemos obtener una salida que varíe entre 0 y el rango máximo de tensión aceptada por el convertidor ADC.

$$V_0 = \frac{R_3}{R_1} * V_1 + \frac{R_3}{R_2} * V_2 \quad (21)$$

Para nuestro caso, al sumarse el valor de referencia máximo aceptable de 3V, a la tensión variable entre $\pm 1,5V$ obtenemos el rango final de 0-3V

- **Etapa de protección:** Última etapa mediante la cual nos aseguramos de que la tensión que llega al convertidor ADC, no sobrepase los valores nominales que nos da el fabricante. Este proceso se lleva a cabo mediante un solo elemento electrónico: El diodo Zener.

El diodo Zener tiene la peculiaridad de conducir cuando esta polarizado, pero de polarizarse en inversa en cuanto la tensión en sus bornes es mayor a una predeterminada. Para nuestro caso es interesante fijar esta tensión en un valor cercano a 3.3V, el valor máximo de tensión permitido a la entrada del convertidor ADC.

El esquema final del acondicionador sería el siguiente:

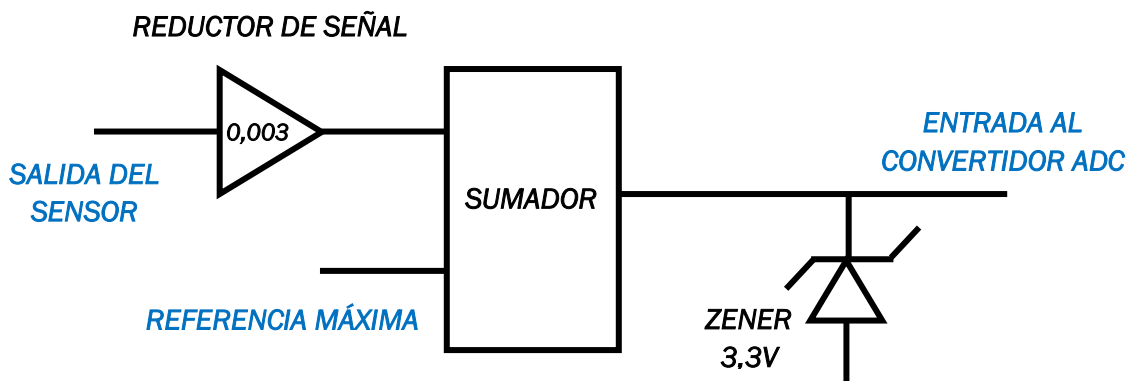


Figura 117. Esquema de acondicionador de señal óptimo

Si buscamos un acondicionador de señal algo más comercial, el mercado nos aporta dos buenas opciones que son los acondicionadores SIRIUS HV (Figura 118) o los SIRIUS HS-HV del fabricante dewesoft. Son elementos mucho más robustos y pesados pero que añaden etapas de filtrado o aislamiento además de aportar conectores adecuados para la aplicación que deseemos llevar a cabo [38].



Figura 118. Acondicionador de señal SIRIUS HV

En conclusión, adquirir un acondicionador comercial implica añadir otro elemento pesado a nuestro esquema, contando ya con la fuente de tensión y el elemento de carga, por lo tanto, vamos a optar por diseñar nuestro propio acondicionador en una placa PCB siguiendo las etapas señaladas anteriormente.

También puede darse el caso en el que el inversor incluya su propio acondicionador de señal. Si esto ocurriera, suprimiríamos el diseño y nos adaptaríamos a las condiciones que aportan las salidas de tensión y corriente del inversor.

4.3.2.3 Inversor y filtro

El inversor es el elemento principal de este proyecto, sin él sería imposible llevarlo a cabo, y gracias a él vamos a controlar el valor de la potencia que inyectamos a la red. El funcionamiento de este elemento ya lo hemos explicado en profundidad en el apartado 3.1.3 El inversor, para este apartado, nos vamos a centrar en sus características técnicas y en las del filtro que lo va a acompañar:

- Inversor

Al conocerse una variedad de gamas de inversores en el mercado, nos hemos centrado únicamente en tres.

La **primera opción** que nos llama la atención, sobre todo por su cercanía geográfica, es la solución de **Powers Electronics**. Esta empresa Valenciana ofrece sobretodo inversores destinados para el autoconsumo industrial, por lo tanto, con un rango de potencias muy grande que alcanza hasta los 3.8 MW. El diseño del inversor se caracteriza por su robustez, que asegura reducir los costes de instalación. El acceso a la etapa de potencia es sencillo y aseguran remplazo rápido de elementos averiados.

Como es obvio, tras observar sus dimensiones en la *Figura 119*, su implementación en un laboratorio se nos antoja inviable. Es una buena solución para granjas solares, pero lejos del alcance de nuestro proyecto [39].



Figura 119. Inversor Freesun HEMK

La **segunda posibilidad** que presentamos como opción viable de implementación, es el prototipo EDPS Power Stage que nos ofrece el fabricante **Trenz Electronic**. Al contrario que la oferta de Power Electronics, en este caso, el inversor se caracteriza por su pequeño tamaño, lo que lo hace de primeras mucho más accesible. En la *Figura 120*, podemos ver una foto del inversor donde si nos fijamos, nos damos cuenta de que se incluye el transductor de tensión LV-25 P que hemos comentado en el apartado 4.3.2.2 Sensores y acondicionadores, lo que nos señala que no tenemos que hacernos con él. Aun así, sí que va a ser necesario desarrollar un acondicionador de señal externo al inversor ya que en la PCB no se incluye.



Figura 120. Inversor EDPS Power Stage

En cuanto a características técnicas nos permite utilizar un rango de tensiones de entrada de hasta 48V y 30A de corriente y nos proporciona además la medida de estas mismas con el sensor LV-25 que mide al menos dos fases.

El pequeño rango de tensiones que nos ofrece este inversor, y el hecho de que esté diseñado especialmente para el control de motores, nos conduce a desechar este diseño, a pesar del tamaño óptimo que ofrece [40].

Por último, llegamos a la tercera y **última opción** que vamos a valorar, y se trata del inversor TIDA-010025 de la familia de **Texas Instrument**. El hecho de pertenecer a la misma línea de fabricantes que la tarjeta controladora, LANCHXL-F28379D, es la primera ventaja que nos ofrece este inversor, pero no es la única. Su rango de tensión continua de entrada nos permite valores desde 200V a 800V y, además, nos ofrece un sistema sofisticado de adaptación de la señal que integra la recogida de valores de tensión y corriente en la propia PCB.

En la *Figura 121* podemos ver como para la implementación del inversor no es necesaria solo una placa electrónica, en total son tres las que se van a utilizar: La controladora LAUNCHXL-F28379D que contiene el programa de control, la etapa inversora basada en transistores IGBT, que se alimenta de las tensión continua de entrada para dar lugar a las tres fases de alterna en la salida, y el módulo controlador que procesa las señales PWM enviadas desde el control y acondiciona las señales medidas por la etapa de potencia que comunicar al control [41].



Figura 121. Inversor TIDA-010025

En conclusión, tras revisar diversas opciones en el mercado, nos quedaríamos con el inversor TIDA-010025 que ofrece Texas Instruments, no solo por la comodidad de trabajar con el mismo fabricante que el de la tarjeta de control, si no que por la gran accesibilidad que aporta, por el rango aceptable de tensiones y corrientes que podemos generar y por su tamaño no excesivamente grande.

- Filtro

El filtro más deseable entre todos los tipos que existen, para llevar a cabo el acoplamiento a la red de una salida trifásica, es el filtro LCL.

El filtro LCL, tiene como ventaja respecto al filtro L, la reducción de la impedancia total del sistema que afecta directamente al peso de la bobina, al cobre y al paquete magnético, que disminuyen considerablemente. Otra de las virtudes de este filtro es la capacidad de conseguir un filtrado mejor que en el caso inductivo, (incluso con una inductancia equivalente menor que en ese caso) y es que la

frecuencia de resonancia²¹ va a ser más pequeña en el filtro LCL y empieza a frecuencias menores la atenuación de la señal, logrando así atenuar a la misma frecuencia que un filtro L mucha más magnitud.

En la *Figura 122* podemos ver el esquema eléctrico de un filtro LCL, donde las impedancias de las bobinas se asocian en paralelo.

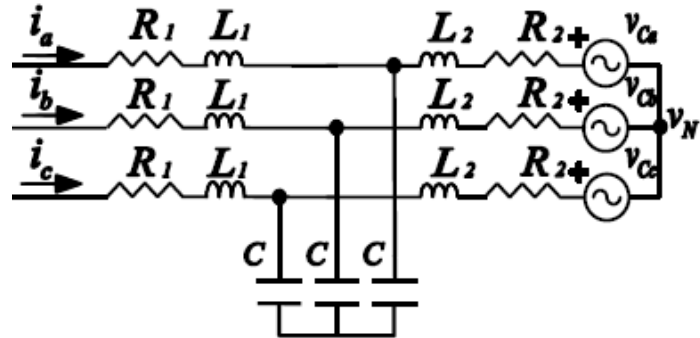


Figura 122. Filtro LCL

Es un filtro muy adecuado para aplicaciones de inyección a la red desde un inversor, pues produce baja distorsión de la corriente en la salida y produce poca potencia reactiva. La principal desventaja es su tamaño, ya que, aun habiendo reducido su peso necesitaríamos bastante espacio para albergarlo. [42][43]

En el mercado vamos a destacar el filtro LCL que nos ofrece el fabricante **Hilkar** ya que nos aseguran facilidad en la instalación y rentabilidad a largo plazo. Además, añaden unas características técnicas que nos convienen enormemente, como son la tensión nominal de hasta 690V, y la corriente de hasta 4250A. En la *Figura 123* podemos ver su aspecto [44].



Figura 123. Filtro comercial LCL del fabricante Hilkar

²¹ Frecuencia para la cual el filtro funciona como corto circuito y a partir de ella se atenúan las ganancias

4.3.2 Conexión de la tarjeta LAUNCHXL-F28379D con el inversor

Como hemos ido viendo en el apartado anterior, ya tenemos en líneas generales, los aparatos necesarios para poner en marcha nuestro proyecto. En este punto vamos a recordar las características técnicas de cada uno y como podría realizarse su conexión.

Para comenzar, hemos reunido en la *Tabla 13* las características técnicas de cada producto que hemos ido seleccionando en el apartado anterior.

Tabla 13. Características técnicas de los elementos que van a formar el desarrollo práctico del proyecto.

ELEMENTO	CARACTERÍSTICAS TÉCNICAS	
	ALIMENTACIÓN	SALIDA
Fuente de alimentación de Elektro-Automatik	340-360 Vac	0-1500 Vdc, 30A, 15kW
Inversor TIDA-010025 de Texas Instrument	Inversor: 200-800 Vdc IGBT: 15Vdc/-8Vdc	200-480 Vac, 25A
Tarjeta de desarrollo LAUNCHXL-F28379D	ADC: 0-3.3 V	PWM: 0-3.3 V
Filtro LCL de Hilkar	0-690 Vac nominal, 1A-4250A, 630kW	
Simulador de red REGATRON Serie TopCon	360-440 Vac, 85A, 0-50kVA	

Para cada uno de los elementos, podemos observar cuales son los rangos de tensión necesarios a la entrada, para la alimentación y cuales se pueden dar a la salida. Estos valores son fundamentales para definir si los elementos son compatibles entre sí en la implementación final del proyecto.

Si comenzamos desde la parte final de la aplicación es fácil deducir que no podemos sobrepasar la generación de 440 V en alterna para alimentar el simulador de red, este valor va a ser la referencia máxima para el filtro y sobre todo para la salida del inversor.

El filtro es compatible con el simulador, pues es capaz de procesar señales de hasta 690 V, pero con el inversor hemos de tener más cuidado pues no depende solo del rango de salida (que en este caso es muy aceptable), si no que de la tensión continua que aportemos a la entrada.

Un valor ideal para asegurarnos no llegar a ese valor máximo de 440 Vac podría ser el de la mitad de esta tensión en continua esto es: 220 Vdc. Como observamos, no tendríamos ningún problema en obtener esta señal a partir de la fuente de alta potencia, que aporta un rango desde 0 hasta 1500 Vdc.

A continuación, en los siguientes apartados, vamos a analizar las conexiones más complejas que nos vamos a encontrar: las conexiones del inversor y las de la tarjeta LAUNCHXL-F28379D.

4.3.2.2 Conexión del inversor

Como hemos estudiado en apartados anteriores, el inversor va a ser el elemento central de la aplicación, y como mínimo necesita una tensión continua de entrada para transformarla en una señal alterna.

En el caso del inversor TIDA-010025, que vamos a implementar, no es solo necesaria esta señal continua, necesitamos disponer además de una tensión de alimentación externa de 24V. Esta tensión será útil para alimentar entre otros elementos: los transductores de señal, los IGBT, los elementos de aislamiento, del módulo controlador, la tarjeta LAUNCHXL-F28379D...

La entrada de la tensión continua hacia el inversor se produce a partir de las borneras J2 y J5, equipadas con dos terminales cada una. Los dos terminales externos se corresponden con los valores +DC y -DC, los dos terminales internos son utilizados para añadir un condensador de protección ante sobretensiones. En la *Figura 124* se indica su localización sobre la PCB.

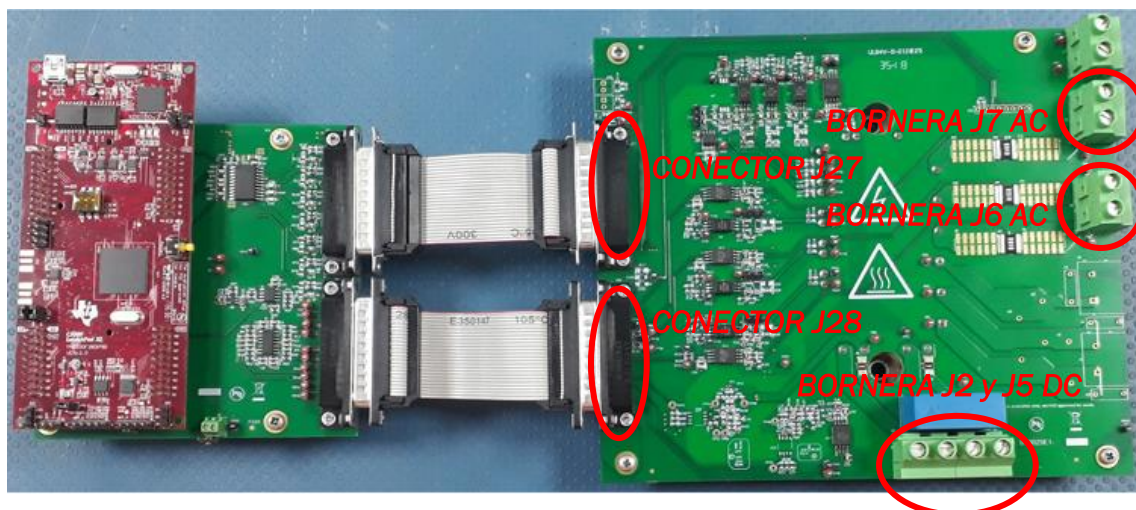


Figura 124. Localización de conectores importantes sobre la PCB del inversor

El conector J27 va a ser el encargado de transmitir las seis señales PWM, generadas en la tarjeta LAUNCHXL-F28379D y procesadas por el módulo controlador. El conector J28 transmite desde el inversor la realimentación de corrientes hacia la tarjeta.

La última conexión que podemos analizar sobre la *Figura 124* es la salida de tensión trifásica que se produce también a través de dos borneras: J6 para las fases U y V, y J7 para la fase W y la tierra. Sus valores máximos de salida en alterna son de 480V.

La llegada de la tensión a la placa de 24V, para alimentar a los IGBT, se produce a través de otro módulo externo de aislamiento: El módulo TIDA-00199. Como podemos ver en la *Figura 125*, la placa que lo contiene no es excesivamente grande y cumple, básicamente, con la función de convertir la tensión continua de 24V en cuatro señales +15V/-8V que van a alimentar los 6 IGBT del inversor a través de tres terminales cada uno.

La conexión con el inversor se realiza a través de cuatro cables trenzados que se conectan a los terminales correspondientes para cada par de IGBT [41].



Figura 125. Módulo TIDA-00199

4.3.2.2 Conexión de la tarjeta LAUNCHXL-F28379D

Como ya hemos avanzado en el punto 4.2 APLICACIÓN FINAL. A , el control del inversor va a tener su base en la tarjeta LAUNCHXL-F28379D a través del algoritmo de control. Este algoritmo, como señalamos en la *Figura 100*, necesita, como entrada, la realimentación de señales de corriente de la etapa de potencia, para dar como salida los pulsos PWM que se devuelven al inversor para el control de los polos.

Para llevar a cabo todas estas conexiones, el fabricante aporta, junto al módulo inversor, el módulo controlador integrado: **TIDA-010025CB**. Este módulo va a tener la función principal de servir como interfaz de comunicación entre el inversor y la tarjeta LAUNCHXL-F28379D.

Si analizamos la forma de este circuito integrado, lo primero que nos puede llamar la atención es la aparición de cuatro terminales macho (J1, J2, J3 y J4) con 20 pines cada una que van a hacer la función de conexión con las terminales hembra de la parte inferior de la tarjeta LAUNCHXL-F28379D.

Para conocer que pin debemos programar en la tarjeta para que esta recoja la realimentación de tensiones y corrientes por las entradas ADC, el fabricante nos ofrece unas tablas donde nos indica el pin que debemos usar. La *Tabla 14* es la que nos ofrece que pines debemos programar para recibir las corrientes de las tres fases acondicionadas (U, V, W) a través del convertidor analógico digital ADC.

Tabla 14. Relación entre pines macho del módulo controlador TIDA-010025CB y pines hembra J1 y J3 de la tarjeta LAUNCHXL-F28379D.

TIDA-010025 FUNCTION	LaunchPad™ FUNCTION USED	J1 PINS	J3 PINS	LaunchPad™ FUNCTION USED	TIDA-010025 FUNCTION
+3.3 V	3.3 V	1	21		
		2	22	GND	GND
		3	23	ADCIN14	V_DC_LINK
		4	24		
		5	25		
		6	26		
		7	27	ADCINC2	I_U_PHASE
		8	28	ADCINB2	I_V_PHASE
		9	29	ADCINA2	I_W_PHASE
		10	30		

Como se puede apreciar, van a ser los pines 27, 28 y 29 que se corresponden con la segunda entrada de los bloques ADC A, B, y C, los que van a recibir las corrientes de fase U, V, y W. El pin 1 y el 22 reciben la alimentación, y el pin 23 recibe a partir de la entrada ADCIN14 la tensión continua que se le aplica al inversor.

Para la salida de las señales PWM, a partir de la *Tabla 15* se nos indica que van a ser los pines desde el 35 hasta el 40 los que pueden mandar las señales de control, hacia cada fase, donde es necesaria la salida PWM positiva y la negativa.

Tabla 15. Relación entre pines macho del módulo controlador TIDA-010025CB y pines hembra J4 y J6 de la tarjeta LAUNCHXL-F28379D.

TIDA-010025 FUNCTION	LaunchPad™ FUNCTION USED	J4 PINS	J2 PINS	LaunchPad™ FUNCTION USED	TIDA-010025 FUNCTION
uC_PWM_UP	EPWM1A	40	20		GND
uC_PWM_UN	EPWM1B	39	19		
uC_PWM_VP	EPWM2A	38	18		
uC_PWM_VN	EPWM2B	37	17		
uC_PWM_WP	EPWM3A	36	16		
uC_PWM_WN	EPWM3B	35	15		
		34	14		
		33	13	GPIO124	OVERLOAD_FAULT
		32	12	GPIO125	Fault_Control
		31	11		

El módulo controlador TIDA-010025CB, se va a encargar también de acondicionar tanto las señales de salida PWM, que se dirigen hacia el inversor, como las de realimentación, que recibe de los sensores de corriente, para el correcto procesamiento por parte de la tarjeta, que solo va a aceptar y enviar señales con rangos entre 0 y 3.3 V [41].

4.3.3 Implementación final en la tarjeta

El último punto para conocer la implementación real del proyecto es el desarrollo del programa de control en la tarjeta LAUNCHXL-F28379D.

Como ya conocemos, las señales trifásicas que toma la realimentación son muy elevadas, y el módulo controlador se encarga de reducirlas para poder así transferirlas a la tarjeta de control. Pero no solo este paso es necesario. Como señalamos en la *Figura 50*, para procesar la señal, después de acondicionarla, es necesaria una conversión Analógico-Digital, por lo tanto, el esquema de nuestro programa en Simulink, debe contener esa conversión necesaria antes de iniciar el algoritmo de control.

Por último, se generarán las señales PWM, a través de las salidas digitales de la tarjeta. Esto mismo es lo que queremos representar en el siguiente esquema de la *Figura 126*:

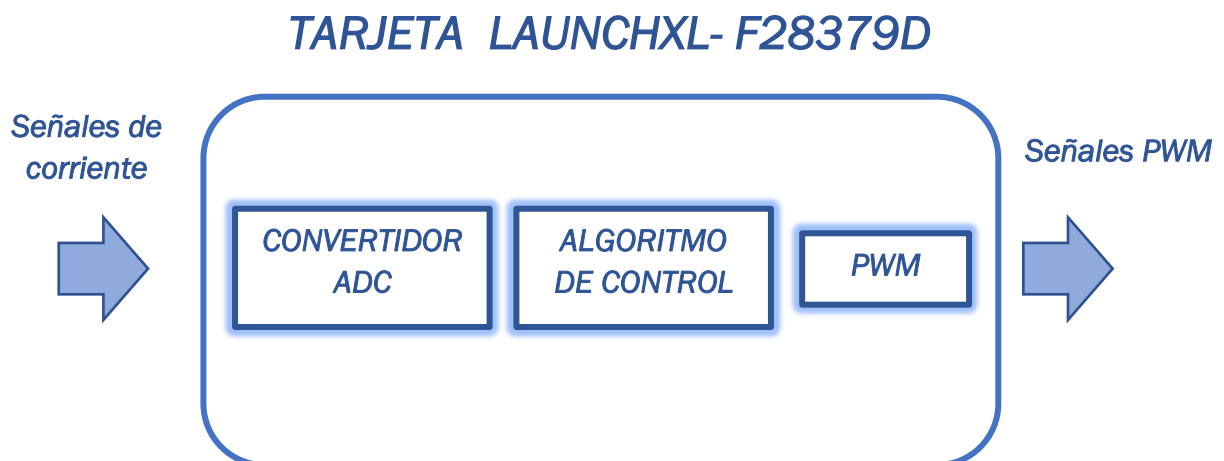


Figura 126. Esquema de funcionamiento general del programa en la tarjeta LAUNCHXL-F28379D

En Simulink, nuestra interfaz programadora, debemos incluir entonces todos estos niveles. La implementación del convertidor ADC ya la hemos estudiado en el apartado 2.4.1 Convertidor ADC, por lo tanto, nos centraremos en la configuración de los parámetros del bloque. El posterior procesamiento de la señal a través del algoritmo dq, es también conocido al explicarse en el apartado 4.2 APLICACIÓN FINAL. ALGORITMO DE CONTROL DQ. Y el último bloque, el de control mediante PWM, ha sido comentado en el apartado 2.3 MÓDULO PWM, donde se tratan los pormenores del módulo PWM de la tarjeta y en el 3.2 CONTROL DEL INVERSOR, donde se decide utilizar el control vectorial para desarrollar la aplicación práctica del proyecto.

El programa final en Simulink, tiene entonces la forma de la *Figura 127*

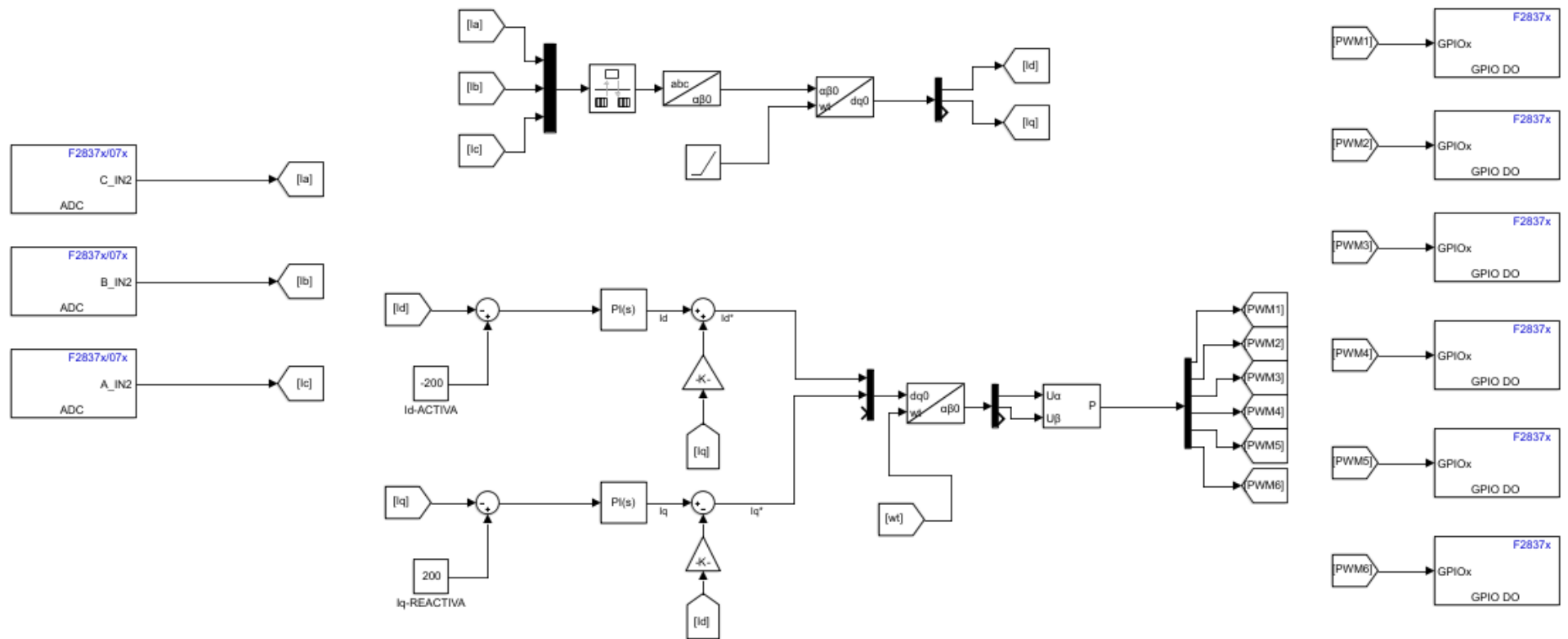


Figura 127. Programa final de control del sistema en Simulink

Como podemos observar, las señales de corriente I_a , I_b e I_c , se obtienen a partir del bloque ADC, a través de las entradas que nos indicaba la *Tabla 14* para la conexión con el módulo controlador. Los parámetros seleccionados para llevar a cabo la conversión son los siguientes:

ADC Module: A

ADC Resolution: 12-bit (Single-ended input)

SOC trigger number: SOC0

SOCx acquisition window: 15

SOCx trigger source: Software

ADCINT will trigger SOCx: No ADCINT

Sample time: 0.001

Data type: double

Post interrupt at EOC trigger

Figura 128. Parámetros del bloque ADC

La configuración de parámetros de la captura se corresponde con la de la gestión de la corriente W , para nosotros I_c , a través de la entrada ADCIN2 del módulo de conversión A. En la otra ventana de selección del bloque indicamos que el canal de conversión será el segundo como indica el fabricante del inversor.

Para la resolución digital, nos decantamos por 12 bits, pues vamos a utilizar una única entrada (no diferencial) para obtener los valores de corriente.

La fuente del disparo se produce al iniciar el programa desde el software, y los datos de salida van a ser de tipo double (int16 es el que está predeterminado), para evitar errores a la salida de las transformadas de Park y de Clarke.

En cuanto al algoritmo de control, solo hemos cambiado dos detalles con respecto a la simulación realizada en el apartado 4.2 APLICACIÓN FINAL. ALGORITMO DE CONTROL DQ. Uno es la aparición del **bloque Rate Transition**, antes de aplicar la transformada de Clarke. La razón de esta novedad es el hecho de tener que contar en este caso con la transferencia de datos, en el programa, desde la tarjeta. Este bloque tiene la función de asegurar la integridad y el valor de los datos recogidos.

El otro cambio es la desaparición de la **referencia ωt** para la transformada de Park, que es sustituida por una señal en forma de rampa (que genera los mismos valores), ya que el circuito generador de la velocidad angular necesitaría leer las tensiones de cada fase, cosa que no aporta el módulo controlador ya que no se incluyen sensores de tensión en el inversor.

Por último, tras aplicar el control vectorial (que ya comentamos, es el más efectivo en aplicaciones trifásicas), enviamos las salidas PWM de vuelta al inversor a través de los puertos de salida digital GPIO 0, 1, 2, 3, 4 y 5, que se corresponden con los pines PWM1A, PWM1B, PWM2A, PWM2B, PWM3A y PWM3B que indicaba el fabricante del inversor a través de la *Tabla 15*.

5. Conclusiones y futuro del proyecto

Una vez que llegamos al final del proyecto es el momento de analizar con objetividad los puntos logrados y de proponer continuidad al trabajo realizado.

En el primer apartado de la memoria proponíamos 4 objetivos fundamentales para cumplir a lo largo del desarrollo del proyecto, que hemos ido desarrollando a través de los diferentes títulos.

El primer objetivo consistía en **describir el hardware** del controlador y el segundo en **conocer su programación** en la interfaz programadora Simulink. A estos dos retos nos enfrentamos en el segundo apartado del proyecto: *2. Desarrollo de tarjeta LAUNCHXL-F28379D y principios de programación en Simulink*. Analizamos en diferentes partes la alimentación, el procesador y los módulos PWM, ADC y DAC de la tarjeta F28379D, y a su vez, dimos soluciones en Simulink para la programación, tanto de los módulos como de algunas partes del procesador como el módulo IPC, para familiarizarnos con la herramienta gráfica.

Con el conocimiento de la tarjeta y de la interfaz desarrolladora del control nos centramos en el tercer objetivo, el de conocer qué clase de **aplicación en electrónica de potencia** le daríamos al proyecto. Así, en el apartado 3. *Aplicación en electrónica de potencia*, introdujimos al lector en los distintos aspectos de este campo de la electrónica haciendo especial hincapié en el que se iba a programar posteriormente: El inversor. Analizamos como se controla este elemento de potencia a través de la tecnología PWM y presentamos también los dos tipos de control que podríamos llevar a cabo, prestando especial atención al control vectorial que es el que finalmente implantamos.

Por último, y para cumplir el cuarto objetivo, nos centramos en el **desarrollo práctico del proyecto**. En este punto hemos presentado los objetivos iniciales de implementación y como la idea fue cambiando con el desarrollo del proyecto. A partir de ahí dimos una aplicación real al proyecto a través de la implementación del algoritmo dq en Simulink para inyectar potencia a la red eléctrica.

En este apartado dimos también una idea general del desarrollo ideal del proyecto en un laboratorio, aportando los elementos óptimos y el programa perfectamente desarrollado en Simulink, para llevar a cabo la idea de implementación real.

El futuro del proyecto pasa por preguntarse de nuevo por las soluciones que se puedan dar a la idea inicial de desarrollar todos los elementos físicos de potencia en la herramienta Simulink, quizá con alguna nueva tecnología como es el Hardware on the loop o con otro software alejado de Matlab. Mientras, solo se puede llevar a cabo la implementación real mediante la adquisición de estos elementos físicos y su montaje en un laboratorio.

Bibliografía

- [1] G. Gunther, W. Bettina, and Vienna University of Technology, *Introduction to microcontrollers*. Viena, 2007.
- [2] N. Goilav, *Arduino - Aprender a desarrollar para crear objetos inteligentes*, Ediciones. 2016.
- [3] A. Valverde Villarán, “Capítulo 2. Microcontroladores: Características y aplicaciones generales,” *Bibl. la Univ. Sevilla*, pp. 1–13, 2013, [Online]. Available: <http://bibing.us.es/proyectos/abreproy/11301/fichero/Memoria%252FCapitulo+2.pdf+>.
- [4] A. Ardións, “profesionalreview.com.” <https://www.profesionalreview.com/2017/01/18/memoria-ram-rom-diferencias/>.
- [5] R. E-Marmolejo, “hetpro-store.com,” [Online]. Available: <https://hetpro-store.com/TUTORIALES/microcontrolador/https://hetpro-store.com/TUTORIALES/microcontrolador/>.
- [6] M. Pate, “The Essential Guide for Developing With C2000™ Real-Time Microcontrollers,” 2020. [Online]. Available: www.ti.com.
- [7] Texas Instruments, “TMS320F2837xD Dual-Core Microcontrollers,” 2020.
- [8] Texas Instruments, “TPS6208x 1.2-A High-Efficiency, Step-Down Converter With DCS-Control™ and Snooze Mode 1,” 2016.
- [9] Texas Instruments, “LAUNCHXL-F28379D Overview,” 2019.
- [10] U. C. I. de M. Grupo de Sistemas Electrónicos de Potencia, “Conversión CC/CC.”
- [11] Texas Instruments, “LMR62014 SIMPLE SWITCHER® 20Vout, 1.4A Step-Up Voltage Regulator in SOT-23,” no. April, 2013.
- [12] K. W. Schachter and C. Technical, “Enhancing the Computational Performance of the C2000™ Microcontroller Family,” no. April, pp. 1–11, 2020.
- [13] Texas Instruments, “The TMS320F2837xD Architecture : Achieving a New Level of High Performance,” 2016.
- [14] D. M. Huffman, L. H. Tiffany, G. Knaphus, and R. A. Healy, “Technical References,” *Mushrooms Other Fungi Midcont. United States*, no. September, pp. 357–360, 2018, doi: 10.2307/j.ctt20ks0sv.18.
- [15] Open4Tech, “Direct Memory Access (DMA) in Embedded Systems.” .

- [16] J. Roca, “¿Qué es, cómo funciona una unidad DMA y cuál es su utilidad?,” 2020. <https://hardzone.es/reportajes/que-es/unidades-dma/>.
- [17] M. A. Navas, “PWM: Qué es y para que sirve en los ventiladores.” <https://www.profesionalreview.com/2019/10/16/pwm-que-es-y-para-que-sirve/>.
- [18] Matlab, “F2837xD Blocks.” https://es.mathworks.com/help/supportpkg/texasinstrumentsc2000/f2807x-blocks.html?s_tid=CRUX_topnav.
- [19] J. M. Ruiz Gonzalez, “Tema 5 .- Sistemas de adquisición y procesamiento de datos. 1.”
- [20] L. C. Herrero de Lucas and F. Martinez Rodrigo, “Semiconductores de potencia,” *Univ. Valladolid*, p. 38.
- [21] U. C. I. de M. Grupo de Sistemas Electrónicos de Potencia, “Conversión CA/CA.”
- [22] L. C. Herrero de Lucas, “Convertidores CA/CC,” no. 452.
- [23] L. C. Herrero de Lucas, “Convertidores CC/CA,” no. 452.
- [24] J. D. González, “Inversores Pwm,” pp. 1–26, 1998.
- [25] S. Seguí Chilet, R. Masot Peris, S. Orts Grau, and F. J. Gimeno Sales, “Control vectorial de inversores trifásicos 1,” *Rev. española electrónica*, no. 582, pp. 66–72, 2003.
- [26] C. E. Noriega, “Control Control Vectorial de Corriente del Inversor Trifásico con Carga RL,” 2012.
- [27] G. Díaz, C. Peraza, F. Gonzalez-longatt, and C. Villanueva, “Control de inversores en sistemas de generación distribuida,” *Rev. Ing. UC*, vol. 12, no. 3, pp. 29–39, 2005.
- [28] D. Ron, A. Barba, “Contrucción E Implementación Del Sistema De Control Vectorial De Orientación De Campo Directo (DFOC) Para Un Motor Trifásico De Inducción,” pp. 1–193, 2006.
- [29] D. P. Dorantes and M. H. Ángeles, “Análisis de las técnicas de modulación SPWM y SVPWM en control de inversores trifásicos de dos y tres niveles,” vol. 1, no. May, pp. 587–592, 2016.
- [30] S. Seguí Chilet, R. Masot Peris, S. Orts Grau, and F. J. Gimeno Sales, “Control vectorial de inversores trifásicos 2,” *Rev. española electrónica*, no. 582, pp. 66–72, 2003.
- [31] P. Nieto Montero, “Análisis del modelo y control de inversores en fuentes de energía fotovoltaica,” 2010, [Online]. Available: <http://www.bvsspa.es/papi/ezproxy.php?url=http://search.ebscohost.com/login.aspx?direct=true&db=edsbas&AN=edsbas.7D70C3FE&lang=es&site=e>

ds-live&scope=site.

- [32] T. Sathiyarayanan and S. Mishra, "Synchronous reference frame theory based model predictive control for grid connected photovoltaic systems," *IFAC-PapersOnLine*, vol. 49, no. 1, pp. 766–771, 2016, doi: 10.1016/j.ifacol.2016.03.149.
- [33] L. Llamas Binaburo, "¿Qué es la energía reactiva?," 2013, [Online]. Available: <https://www.luisllamas.es/que-es-la-energia-reactiva/>.
- [34] P. Hochleistungs-dc-netzgeräte, "EA-PSI 9000 3U 3.3 kW - 15 kW," 2018.
- [35] ACS, "Grid Simulator –full digital, full 4-quadrant, full regenerative."
- [36] LEM, "Current Transducer IT 205-S ULTRASTAB," 2010.
- [37] LEM Components, "Voltage Transducer LV 25-P I PN = 10 mA V PN = 10 .. 500 V Electrical data," 1997.
- [38] Dewesoft, "Dewesoft Training Portal."
- [39] powerelectronics, "Freesun HEMK." <https://power-electronics.com/es/productos/productos-solar/hemk/>.
- [40] Trenz-electronic, "EDPS Power Stage." <https://shop.trenz-electronic.de/en/TEC0053-04-EDPS-Power-Stage?number=TEC0053-04>.
- [41] Texas Instruments, "Three-phase inverter reference design for 200 – 480 VAC drives with opto-emulated input gate drivers," no. November 2018, pp. 1–39, 2019.
- [42] A. Villa Manrique, "Estudio del filtro LCL aplicado a inversores fotovoltaicos," pp. 9–32, 2011.
- [43] C. L. Trujillo Rodríguez, D. V. De La Fuente, E. Figueres Amorós, G. Garcerá Sanfeliú, and J. Guacaneme Moreno, "Diseño, modelado e implementación de inversor conectado a la red eléctrica a partir de fuentes renovables," *Rev. Tecnura*, vol. 16, no. 32, p. 12, 2012, doi: 10.14483/udistrital.jour.tecnura.2012.2.a01.
- [44] Hilkar, "Filtros LCL Hilkar." [Online]. Available: <http://www.hilkar.com/es/filtroslcl.html>.
- [45] T. Instrument, "System Control and Interrupts," 2009.