PLATAFORMA DE DESARROLLO DE CONTROL PARA INYECCIÓN A LA RED ELÉCTRICA.

Martín Edgardo Basavilbaso

Este trabajo Final fue presentado al Departamento de Ingeniería Electrónica de la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata el 16 de 08 de 2022, como requisito para la obtención del título de

Ingeniero en Electrónica

Director: Dr. Ing. Sergio A. González

Co-Director: Dr. Ing. Jonatan R. Fischer

RINFI

REPOSITORIO INSTITUCIONAL

INTEMA | FACULTAD DE INGENIERÍA | UNMDP

RINFI se desarrolla en forma conjunta entre el INTEMA y la Biblioteca de la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata.

Tiene como objetivo recopilar, organizar, gestionar, difundir y preservar documentos digitales en Ingeniería, Ciencia y Tecnología de Materiales y Ciencias Afines.

A través del Acceso Abierto, se pretende aumentar la visibilidad y el impacto de los resultados de la investigación, asumiendo las políticas y cumpliendo con los protocolos y estándares internacionales para la interoperabilidad entre repositorios


Esta obra está bajo una Licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional.

PLATAFORMA DE DESARROLLO DE CONTROL PARA INYECCIÓN A LA RED ELÉCTRICA.

Martín Edgardo Basavilbaso

Este trabajo Final fue presentado al Departamento de Ingeniería Electrónica de la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata el 16 de 08 de 2022, como requisito para la obtención del título de

Ingeniero en Electrónica

Director: Dr. Ing. Sergio A. González

Co-Director: Dr. Ing. Jonatan R. Fischer

Índice general - contenidos:

1. Agradecimientos:

A mi familia por darme la oportunidad de estudiar y apoyarme en todos mis proyectos, a mis amigos por bancarme siempre, y a todas las personas que me acompañaron a lo largo de estos años.

Martín Basavilbaso

2. Introducción:

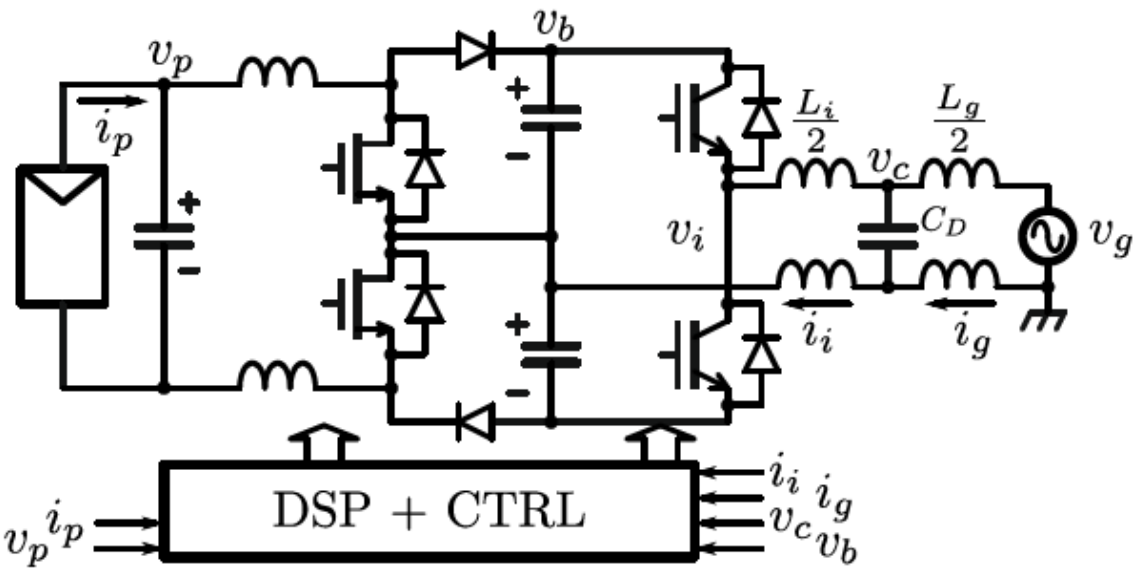
El presente trabajo se enfoca en la puesta en marcha de una plataforma de control que permite la implementación de diferentes convertidores de potencia. Esta plataforma integra en un solo sistema las funcionalidades para ello y pone a disposición del LIC (Laboratorio de Instrumentación y Control) un dispositivo que sirve para realizar investigación, desarrollo y ensayo de procesadores de potencia. Dicho trabajo se puede dividir en dos partes (una parte de Hardware y otra de Firmware) que resuelven distintas problemáticas para el correcto funcionamiento del sistema como:

- Adquisición y acondicionamiento de señales de interés
- Protecciones del sistema en caso de fallas
- Integración, Interconexión y alimentación de los componentes antes mencionados
- Interfaces con el resto de los componentes (sensores, drivers de potencia, etc.)
- Algoritmo de control y generación de PWMs
- Sincronismo
- Implementación del Controlador
- Puesta en marcha y testeo en funcionamiento

Puesto que una de las líneas de investigación del LIC es la de energías renovables e inyección de energía (Control de inyección de corriente en sistemas de generación distribuida), se consensuo con los directores del proyecto el comisionamiento y puesta en marcha de un inversor de potencia para inyectar a la red eléctrica a modo de demostrar el funcionamiento íntegro y la performance del sistema, y continuar uno de los proyectos de investigación actual del laboratorio.

El inversor de potencia opera procesando una fuente de potencia DC generando mediante la conmutación de llaves de potencia una señal AC que en el caso de la aplicación antes mencionada, es una señal sinusoidal, la cual debe ser inyectada de forma sincrónica (monitoreando las fases de línea) a la red para operar de manera correcta. Para lograr dicha señal sinusoidal, se conmutan las llaves en base a una referencia generada por firmware mediante un esquema tipo PWM (pulse width modulation), donde se modifica el valor medio visto a la salida del dispositivo en base al tiempo de encendido y apagado de dichas llaves. Luego, esta señal es filtrada por un filtro LCL y así se obtiene la forma de onda adecuada para esta aplicación. Para que dicha forma de onda sea aceptable la corriente inyectada a la red es monitoreada y realimentada al sistema, el cual debe cumplir con una serie de indicadores de performance a modo de que esta sea aceptable.

El diagrama en bloques del sistema en cuestión es el siguiente:



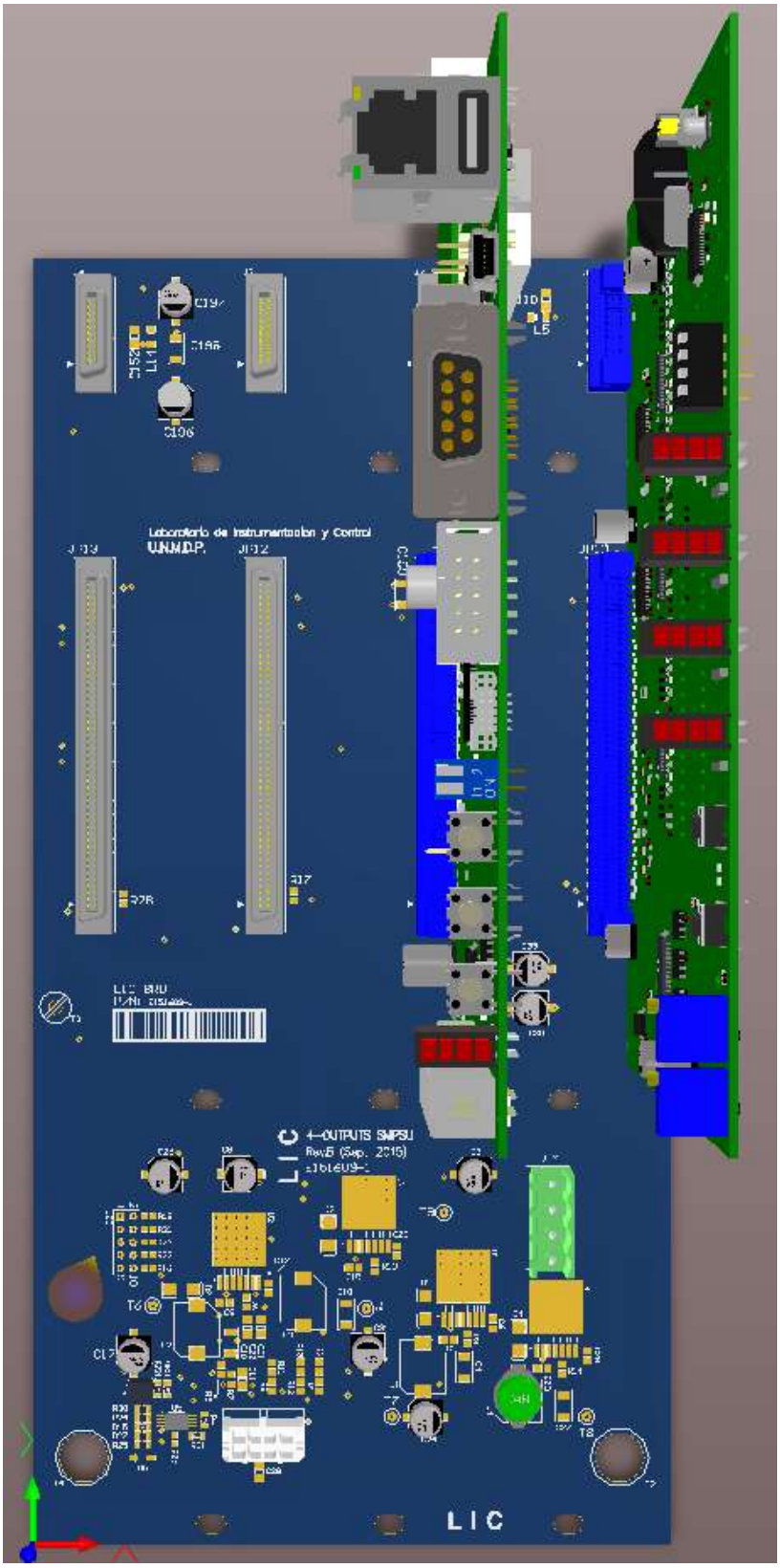
Sistema de inyección a Red eléctrica (para una rama).

Como características del inversor de potencia implementado, cabe destacar que este es del tipo controlador por corriente, su frecuencia de conmutación es de $10\ KHz$, el sistema posee una frecuencia de adquisición de magnitudes de $30\ KHz$ (a modo de logar un sobre-muestreo), posee una tensión de continua en el DC Bus de aproximadamente $600\ V$. Las llaves utilizadas poseen una tensión máxima de bloqueo de $1200\ V$, y pueden manejar una corriente de hasta $50\ A$**.** Además, debe tenerse en cuenta un tiempo muerto mínimo de $2\ \mu s$, y los inductores de acople utilizados entre el inverter y la red eléctrica son de $1.7\ y\ 1.4\ mHy$ respectivamente. Estos parámetros de diseño son tenidos en cuenta en la plataforma que controla al sistema.

Desde el punto de vista del Hardware, el sistema consiste de una serie de PCBs (printed circuit boards, o circuitos impresos) los cuales poseen tanto los componentes programables como discretos necesarios para permitir la implementación de las funcionalidades antes mencionadas. En particular se destaca la utilización de microcontroladores de 32 bits ARM M4 como corazón del proyecto en cuestión, como también así de CPLDs (complex programmable logic devices, dispositivos de lógica programable) para lograr algunas funciones y permitir mayor flexibilidad y una interfaz más sencillas entre los distintos subsistemas del assembly.

Dicho hardware fue desarrollado en conjunto con integrantes del LIC, utilizando tanto conocimientos previamente obtenidos a lo largo de la carrera como así también documentación técnica disponible para los ICs utilizados, y una serie de distintos software que facilitaron y permitieron la realización de distintas etapas del desarrollo, como por ejemplo el Altium Designer para el diseño físico de los PCBs, Filter Pro para el diseño del filtro de entrada en la etapa de adquisición, SPICE para simulaciones de subsistemas en diferentes etapas del proyecto, ISE para la síntesis del hardware en base al HDL, entre otros.

La arquitectura general del hardware del sistema ya se encontraba diseñada previamente puesto que esta es una continuación de una de las líneas de investigación desarrollada por el LIC donde participaron los directores del proyecto junto a otros estudiantes.



Render 3D del diseño y arquitectura inicial del sistema.

Las funciones mencionadas previamente se implementaron de la siguiente manera:

- **DigitalControlBoard:** En este PCB se realizó la función de dirigir, controlar y generar las señales y tiempos requeridos para el funcionamiento del sistema, el algoritmo de control que calcula la acción de control a aplicar sobre la planta en base a las mediciones obtenidas en AnalogFrontEnd, la generación de PWMs a partir del cómputo de dicho algoritmo, el sincronismo respecto a la red eléctrica y partes de las protecciones del sistema en caso de fallas.

- **AnalogFrontEnd:** En este, se realizó el acondicionamiento y la adquisición propiamente dicha de las magnitudes de interés (tensiones de red, corrientes inyectadas, tensión de DC bus) como así también se implementó la otra parte de las protecciones del sistema en caso de falla (aquí en particular, las protecciones de hardware).

- **InterfaceConn:** En este circuito impreso, se sintetiza la interfaz que permite interconectar los PWMs generados por la DigitalControlBoard con los drivers de las llaves de potencia, teniendo en cuenta la adaptación de niveles de tensión adecuada para poder controlar los diferentes tipos de drivers de potencia disponibles.

- **AnalogFrontEndConn:** En este circuito, se presenta la interfaz entre los sensores de las magnitudes y circuitos de aislación para la adquisición de dichas magnitudes, como así también la adaptación entre conectores incompatibles y la expansión a nuevos conectores.

- **BaseBoard:** Finalmente, este PCB interconecta los circuitos previamente mencionados y proporciona la alimentación correspondiente para el correcto funcionamiento del assembly.

Conjunto de PCBs que componen la plataforma de control implementada en el LIC.

Desde el punto de vista del Firmware, se trabajó para lograr que el hardware disponible en los PCBs funcione de manera adecuada y cumpla con las funciones antes mencionadas de forma eficaz. Para ello, se discutió con los directores del proyecto posibles implementaciones de código y funciones de firmware necesarias para obtener una plataforma con los resultados esperados, además de utilizar los módulos de hardware dedicados disponibles en los microcontroladores para lograr un desarrollo más simple y eficiente. Se determinó que el firmware posea una estructura modular o de librería donde los distintos componentes sean independientes y se puedan integrar entre si ya sea en el código principal, o mediante la utilización de colas y comandos en el RTOS (real time operating system, o sistema operativo de tiempo real).

Esta implementación cuenta con varios niveles y/o jerarquías dependiendo de qué tan crítica o sensible sea la función en particular desde el punto de vista del tiempo o la operatividad del sistema. Dicha jerarquía puede ser representada en dos grupos básicos:

- **Críticos**: Implementados de forma directa (a bajo nivel), se ejecutan independientemente del resto de los procesos o estados del sistema a partir de interrupciones de módulos de hardware o en base a los estados de una FSM (máquina de estados finita), tales como la lectura de las magnitudes adquiridas, el sincronismo respecto a la red eléctrica, la generación de los PWMs, referencias o las protecciones en caso de una falla del sistema.

- **Otros**: Implementados a más alto nivel, utilizando de base un sistema operativo de tiempo real. En este grupo se encuentra la HMI (interfaz usuario maquina) de tipo línea de comandos, donde se ingresan parámetros y/o configuraciones al sistema, además de la comunicación entre procesos de alto nivel para la implementación a futuro de por ejemplo, un monitoreo remoto.

El firmware fue desarrollado principalmente en el IDE (entorno de desarrollo integrado) Eclipse en lenguaje C, utilizando diferentes plug-ins que permiten y facilitan el desarrollo de código programable y utilizable por microcontroladores de la familia ARM Cortex.

Cabe destacar que cada etapa o subsistema tanto desde el punto de vista del hardware o firmware del proyecto fue testeado y validado experimentalmente en el laboratorio y el desarrollo del proyecto se fue dando de manera incremental e integrando las diferentes funcionalidades para lograr que el conjunto funcione de forma correcta.

Por último, se realizó el testeo del sistema completo a modo de obtener su performance y la evaluación de distintos algoritmos de control implementados para la inyección a red eléctrica.

3. Conceptos Teóricos:

En este capítulo se realizó una descripción sobre distintas cuestiones teóricas, desde el funcionamiento básico del sistema que se desea implementar, los diferentes aspectos a describir y las funcionalidades que vale la pena destacar en este proyecto para así darle un marco teórico.

Inversor de fuente de tensión:

El inversor de potencia es un sistema procesador de potencia DC/AC, es decir, un dispositivo que permite la conversión de corriente continua provista por una determinada fuente en corriente alterna. Estos pueden clasificarse en dos tipos, los VSI (voltage source inverter o inversor con fuente de tensión) y los CSI (inversor con fuente de corriente). En este caso, el inversor implementado es del tipo VSI.

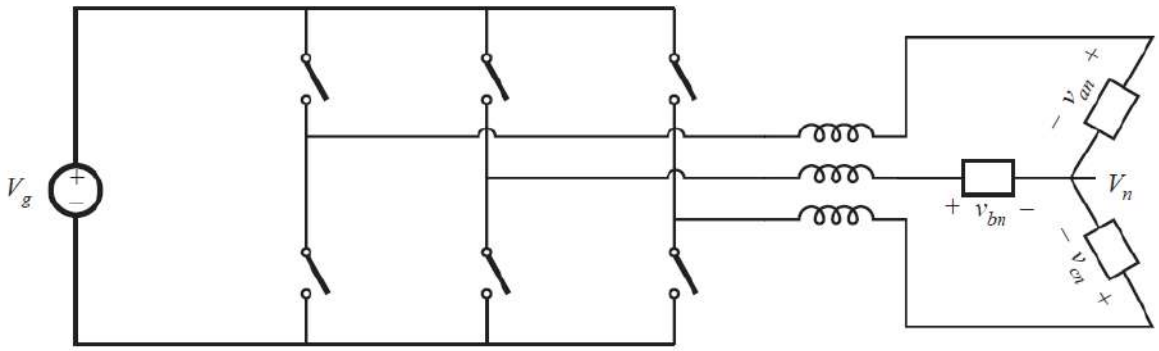


Diagrama genérico de inversor tipo VSI trifásico siendo V_g la fuente DC y las salidas de los inductores la carga trifásica.

Los inversores de tipo VSI disponen de una estructura básica dada por un DC Bus, que provee la entrada al sistema, ya sea por una fuente DC o baterías, y una cantidad de ramas conformadas por dos llaves de potencia que conmutan para generar la corriente alterna que se entrega a la carga. En el caso implementado, el inversor puede ser monofásico o trifásico, lo cual se ve reflejado en que este conmuta una o tres ramas.

Como condición adicional, el estado de las llaves es tal que dos llaves de una misma rama no pueden estar cerradas al mismo tiempo. Esto es debido a que si ambas llaves estuvieran conduciendo, se produciría un cortocircuito en la fuente de DC con el consiguiente daño que se generaría sobre los componentes del sistema.

Ya que la aplicación de este sistema será la de inyección a la red eléctrica, se desean formas de onda sinusoidales a la salida de cada rama. Para ello se utilizara algún método de modulación PWM para generar en base a la conmutación de las llaves dichas formas de onda. En este caso, se aplicó una referencia sinusoidal a los canales de PWM que comandan la conmutación de las llaves, y al pasar por los inductores de acople a la salida, se ven filtrados y se obtiene una señal sinusoidal con un ripple asociado a esta, dado principalmente por la frecuencia de conmutación.

Este sistema trabaja en modo conmutado, por lo que se debe tener en cuenta que las formas de onda generadas tendrán un contenido armónico y una distorsión asociada a la salida. Dicha distorsión dependerá entre otras cosas, de la frecuencia de conmutación, los tiempos muertos insertados para evitar la condición de cortocircuito, el valor del inductor de salida y/o comportamiento del filtro LCL, y la estrategia de control implementada a la hora de inyectar a la red.

Los índices de performance de un inversor están dados por la potencia que este puede inyectar a la red eléctrica, la eficiencia del sistema, y la THD (total harmonic distortion, o distorsión armónica total) que ingresa a la red.

Modelado de la planta para la formulación de controlador:

Se puede considerar a la planta de este sistema, tal que su transferencia sea la superposición de las transconductancias $G_u(s)$ dada por la tensión promedio del inversor, v_i, y $G_w(s)$, debida a la tensión de red, v_g, tomando como salida la corriente inyectada a la red, I_g. Que para el caso de un filtro inductivo simple L, teniendo en cuenta que la impedancia del filtro de acople es $Z_i + Z_g = R_s + s\,L_s = Z_s$

Siendo: $R_s = (R_i + R_g)$, $L_s = L_i + L_g$, y $Z_c \to \infty$

de lo cual se obtiene:

$$G_u(s) = \frac{i_g}{v_i}|v_g = 0 = \frac{1}{s\,L_s + R_s}$$

$$G_w(s) = \frac{i_g}{v_g}|v_i = 0 = \frac{-1}{s\,L_s + R_s}$$

Y que para el caso del filtro LCL, las impedancias son:

$Z_i = R_i + s\,L_i$, $Z_g = R_g + s\,L_g$, $Z_c = R_d + \frac{1}{sC_d}$

De las cuales se obtienen las expresiones aproximadas:

$$G_u(s) \approx \frac{1}{s\,L_s + R_s}\frac{\omega_0^2(1 + sC_dR_d)}{(s^2 + 2\xi_0\omega_0 s + \omega_0^2)}$$

$$G_w(s) \approx \frac{-1}{s\,L_s + R_s}\frac{\omega_0^2(s^2 + 2\xi_1\omega_1 s + \omega_1^2)}{\omega_1^2(s^2 + 2\xi_0\omega_0 s + \omega_0^2)}$$

Donde $2\xi_0\omega_0 = \frac{R_d}{L_p}$, $\omega_0^2 = \frac{1}{L_pC_d}$, $2\xi_1\omega_1 = \frac{(R_d+R_i)}{L_i}$, $\omega_1^2 = \frac{1}{L_iC_d}$

Siendo L_p el paralelo de las inductancias L_i y L_g. En ambos casos se puede observar que el comportamiento del filtro de acople es inductivo en las bajas frecuencias, y esta dominado por la impedancia serie Z_s. La introducción de la rama capacitiva del filtro, introduce términos resonantes, que dependiendo de la selección de componentes, pueden estar dentro del ancho de banda de control y afectar la estabilidad y THD del mismo.

En los casos mencionados, se puede describir al filtro utilizando el modelo de estados:

$$\dot{x}(t) = \boldsymbol{A}x(t) + \boldsymbol{B}u(t) + \boldsymbol{E}w(t)$$

$$y(t) = \boldsymbol{C}x(t)$$

Siendo $\dot{x}(t)$ el vector de estados ($\dot{x}(t) = i_g$ para el caso del inductor, y $\dot{x}(t) = \left[i_i, i_g, v_c\right]^T$ en el caso del filtro LCL), $y(t)$ como la salida a controlar, $u(t) = v_i$ la entrada del control y $w(t) = v_g$ la entrada de perturbación del sistema. Este modelo de estados puede modificarse de manera simple para considerar además la dinámica del sensor de corriente y filtro anti-aliasing asociado a su implementación. La utilización del modelo de variables de estado permite una descripción completa del sistema y permite el diseño del controlador utilizando las mismas herramientas matemáticas para los casos antes mencionados como así también con el agregado de dinámica que conlleva la realización del sistema.

Estrategias de control implementadas:

Se implementó una estrategia de control de tipo RPCC (robust predictive current control, control de corriente predictivo robusto) con un compensador adicional RP (harmonic resonant o armónico resonante) propuesta por los directores, puesto que este era un tema de investigación que se estaba llevando a cabo en ese momento, y también se implementó un control proporcional tipo P y un RPCC a modo de comparar la performance y sacar conclusiones respecto al controlador digital propuesto.

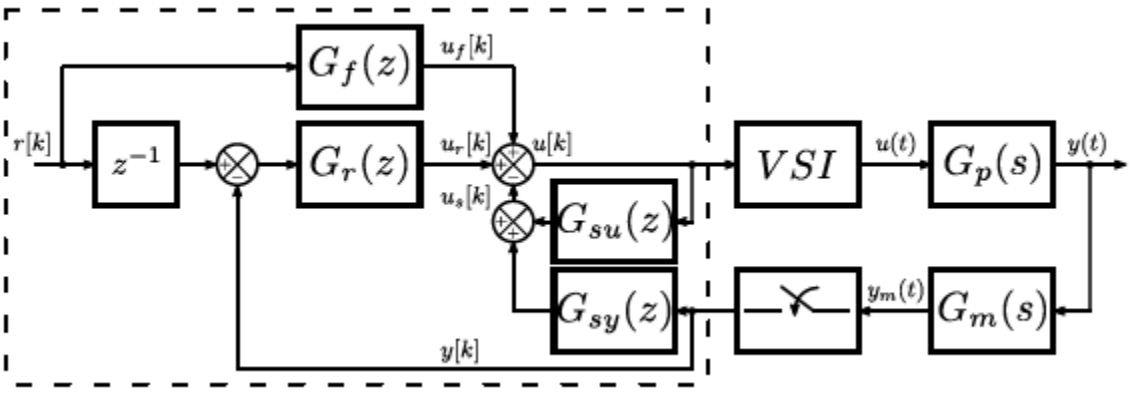


Diagrama en bloques del controlador.

Como el control RPCC se basa en el promediado del sistema en tiempo discreto, se necesita discretizar el modelo continuo mediante el método de retenedor de orden cero, y luego incorporar una variable de estados adicional para tener en cuenta el retardo existente entre la adquisición, cálculo de algoritmo y actualización del modulador PWM.

Al discretizar la planta, esta se puede representar por el siguiente modelo de estados:

$$x_p[k+1] = G_p x_p[k] + \boldsymbol{H}_p u[k] + \boldsymbol{N}_p w[k]$$
$$y_p[k] = \boldsymbol{C}_p x_p[k]$$

Considerando que la entrada de perturbación es cero ($w[k] = 0$) y el retardo es de una muestra, se puede ampliar el vector de estados a $x[k] = [x_p[k], y_p[k]]^T$, de forma que el modelo se representa como:

$$x_p[k+1] = \begin{bmatrix} \boldsymbol{G}_p & 0 \\ \boldsymbol{C}_p & 0 \end{bmatrix} x_p[k] + \begin{bmatrix} \boldsymbol{H}_p \\ 0 \end{bmatrix} u[k]$$
$$y_p[k] = [0 \quad 1] x[k]$$

$$x_p[k+1] = \boldsymbol{G}_s x_p[k] + \boldsymbol{H}_s u[k]$$
$$y_p[k] = \boldsymbol{C}_s x[k]$$

Haciendo $u[k] = -\boldsymbol{K_s} x[k] + K_f r[k]$ donde K_f ajusta la ganancia en continua y $r[k]$ es la referencia de corriente, se pueden localizar los polos a lazo cerrado de la planta en una ubicación arbitraria. Para el diseño de $\boldsymbol{K_s}$ se debe considerar que no se debe mover el polo adicional debido al retardo, con lo cual siempre se cumple que

$$\boldsymbol{K_s} = [K_p, 0]$$

Para compensar el retardo a la implementación del control, es necesario estimar las variables de estado de la planta para el próximo intervalo de muestreo k. Para ello, se usa un observador de predicción de Luenberger y se introduce un retardo en la señal $u[k]$, tal que:

$$x_p[k+1] = \boldsymbol{G}_0 x_p[k] + \boldsymbol{H}_p u[k-1] - \boldsymbol{H}_p w[k] + \boldsymbol{L}_p y[k]$$

Siendo $\boldsymbol{G}_0 = (\boldsymbol{G}_p - \boldsymbol{L}_p \boldsymbol{C}_p)$. La estima futura del vector de estados para realimentación, es decir $u[k] = -\boldsymbol{K_p} x_p[k+1] + K_f r[k]$. Las ganancias del observador, $\boldsymbol{L}_p$ deben ser elegidas por el diseñador de forma de ubicar los autovalores de $\boldsymbol{G}_0$ dentro del círculo unitario.

Puesto que la tensión de red es una entrada de perturbación al sistema que es medida para generar el sincronismo en la corriente generada con la frecuencia de red, esta puede ser usada para cancelar su efecto en la corriente de salida. Para ello, se debe estimar el valor promedio de tensión durante el próximo intervalo de muestreo, y aplicarlo sumando dicha estima a la tensión que aplica el inversor al sistema. Para sistemas monofásicos, esta es:

$$v_g[k+1] = \frac{5}{2} v_g[k] - \frac{3}{2} x_p[k-1]$$

Dicha cancelación es muy buena en el caso del filtro L, pero en el caso del filtro LCL el efecto de esta cancelación se degrada a medida que la impedancia Z_c disminuye.

El control planteado hasta aquí, permite controlar la corriente gracias a la reubicación de polos y una precisa cancelación feedforward de la red, que es la principal perturbación del sistema. A medida que la tensión de red presenta mayor THD, la estima de tensión pierde precisión y se degrada el rechazo. Para el filtro LCL, generar una cancelación precisa, aun sin distorsión armónica, puede llegar a ser un desafío. Otras perturbaciones que no se han modelado, por ejemplo los tiempos muertos de las llaves del inverter, provocan distorsión adicional en la corriente. Por lo tanto se plantea complementar al control descripto anteriormente con un compensador que provea rechazo a las frecuencias armónicas de la red.

Para sistemas monofásicos, un compensador PR tiene polos ubicados en el círculo unitario del plano z, es decir $|z| = 1$. El compensador para una armónica particular se puede expresar como:

$$\frac{U_h(z)}{E(z)} = \frac{-k_{h1} - k_{h2} z^{-1}}{1 - 2\rho \cos(h\omega T_s) z^{-1} + \rho^2 z^{-2}}$$

Donde h es un número entero que indica la frecuencia armónica respecto de la frecuencia nominal $\omega = 2\pi 50Hz$ y $\rho \leq 1$ es un factor que desplaza levemente dentro del círculo unitario a los polos y provee estabilidad numérica. Por lo tanto, el modelo de estados de cada compensador es:

$$v_h[k+1] = \boldsymbol{G}_{rh} v_h[k] + \boldsymbol{H}_{rh}(r[k+1] - y[k])$$

$$u_h[k] = \boldsymbol{K}_{rh} v_h[k]$$

Donde $\boldsymbol{G}_{rh} = \begin{bmatrix} 0 & 1 \\ -\rho^2 & 2\rho\cos(h\omega T_s) \end{bmatrix}$, $\boldsymbol{H}_{rh} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, y $\boldsymbol{K}_{rh} = \begin{bmatrix} k_{h1} & k_{h2} \end{bmatrix}$.

También se puede introducir un integrador $(h = 0)$ en el compensador y en este caso, las matrices son $\boldsymbol{G}_{r0} = 1, \boldsymbol{H}_{r0} = 1\ y\ \boldsymbol{K}_{r0} = k_0$. A partir de esto, el modelo de estados completo del compensador resonante es:

$$\boldsymbol{G}_r = \begin{bmatrix} \boldsymbol{G}_{r0} & 0 & \cdots & 0 \\ \vdots & \boldsymbol{G}_{r1} & \ddots & \vdots \\ 0 & 0 & \ldots & 0 \\ 0 & 0 & \cdots & \boldsymbol{G}_{rn} \end{bmatrix}, \boldsymbol{H}_r = \begin{bmatrix} \boldsymbol{H}_{r0} \\ \boldsymbol{H}_{r1} \\ \vdots \\ \boldsymbol{H}_{rn} \end{bmatrix}$$

Siendo la matriz de ganancias del compensador:

$$\boldsymbol{K}_{rh} = [\boldsymbol{K}_{r0} \quad \boldsymbol{K}_{r1} \quad \cdots \quad \boldsymbol{K}_{rn}]$$

Al considerar todos los componentes del control a lazo cerrado, se obtiene el siguiente modelo de estados:

$$z[k+1] = (\boldsymbol{G}_e - \boldsymbol{H}_e \boldsymbol{K}_e)z[k] + \boldsymbol{H}_f r[k+1]$$

$$y[k] = \boldsymbol{C}_e z[k]$$

Donde:

$$\boldsymbol{G}_e = \begin{bmatrix} \boldsymbol{G}_s & 0 \\ -\boldsymbol{H}_r \boldsymbol{C}_s \boldsymbol{G}_s & \boldsymbol{G}_r \end{bmatrix}$$

$$\boldsymbol{G}_e = \begin{bmatrix} \boldsymbol{H}_s \\ -\boldsymbol{H}_r \boldsymbol{C}_s \boldsymbol{H}_s \end{bmatrix}$$

$$\boldsymbol{G}_e = [\boldsymbol{K}_s \quad -\boldsymbol{K}_r]$$

$$\boldsymbol{G}_e = \begin{bmatrix} K_f \boldsymbol{H}_s \\ K_f \boldsymbol{H}_r \boldsymbol{C}_s \boldsymbol{H}_s + \boldsymbol{H}_r \end{bmatrix}$$

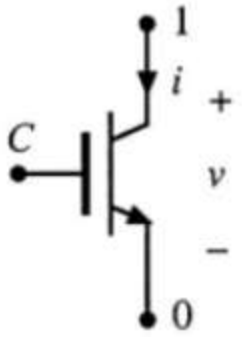
$$\boldsymbol{G}_e = [\boldsymbol{C}_s \quad 0]$$

Generador de Referencia:

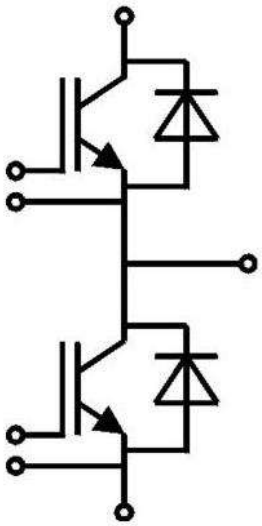
Para generar una referencia en sincronía con la tensión de red, se emplea un PLL (Phase Locked Loop o lazo de enganche de fase) digital de frecuencia de muestreo variable, el cual posee la ventaja de que el producto $\omega T_s = N$ se mantiene constante a medida que la frecuencia de red ω varia, siendo N la cantidad de periodos de muestreo dentro de un periodo de red. Cuando esto sucede, los coeficientes de los compensadores armónicos son constantes y no necesitan ser ajustados si ocurre una variación en la frecuencia, de forma que se puede lograr un alto rechazo a las frecuencias armónicas de la red en todo momento.

Realización de llaves de potencia:

Las llaves de potencia deben ser bidireccionales para permitir el paso de corriente en ambos sentidos para este tipo de aplicación. Dichas llaves son realizadas en la práctica con semiconductores de potencia. En este caso en particular se utilizaron módulos IGBTs (insulated-gate bipolar transistor o transistor bipolar de puerta aislada).



Símbolo de IGBT.



Símbolo de módulo IGBT utilizado.

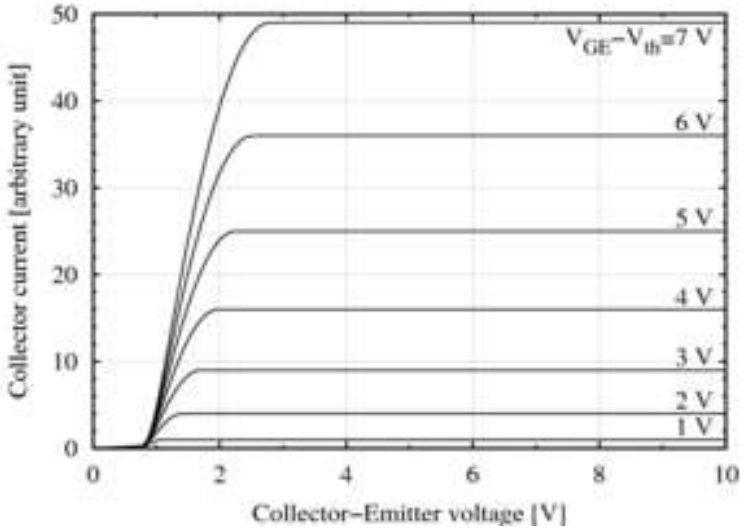


Diagrama de IGBT de tecnología tipo trench y característica $i - v$.

Puesto que su característica $i - v$ solo permite que este conduzca en un cuadrante, en los módulos utilizados ya viene integrado un diodo en anti-paralelo por cada IGBT que logra permitir el paso de corriente en ambos sentidos. Además de esto, los módulos ya sintetizan una rama completa del inversor con acceso a sus pines de control y alimentación.

Dichos dispositivos trabajan en un rango de conmutación típico de $3 - 30\ KHz$ con un tiempo de conmutación de alrededor del microsegundo y se utilizan en aplicaciones de entre $500\ a\ 2\ kV$ y para potencias de $1 - 1000\ kW$.

Por lo expresado anteriormente, y teniendo en cuenta las características del sistema que se desea realizar, estos módulos IGBT resultan adecuados para utilizarse en el proyecto en cuestión.

Como este es un dispositivo semiconductor, el cual no tiene características ideales, tendrá asociada cierta pérdida y baja en la eficiencia del sistema determinada principalmente por la caída de tensión al conducir.

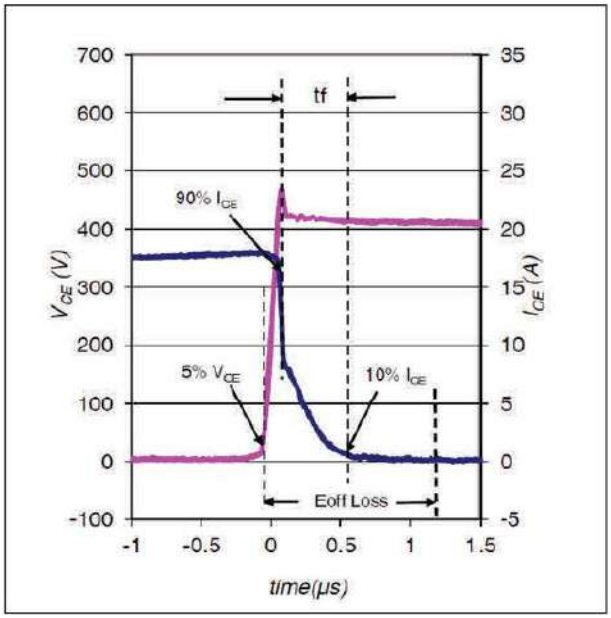
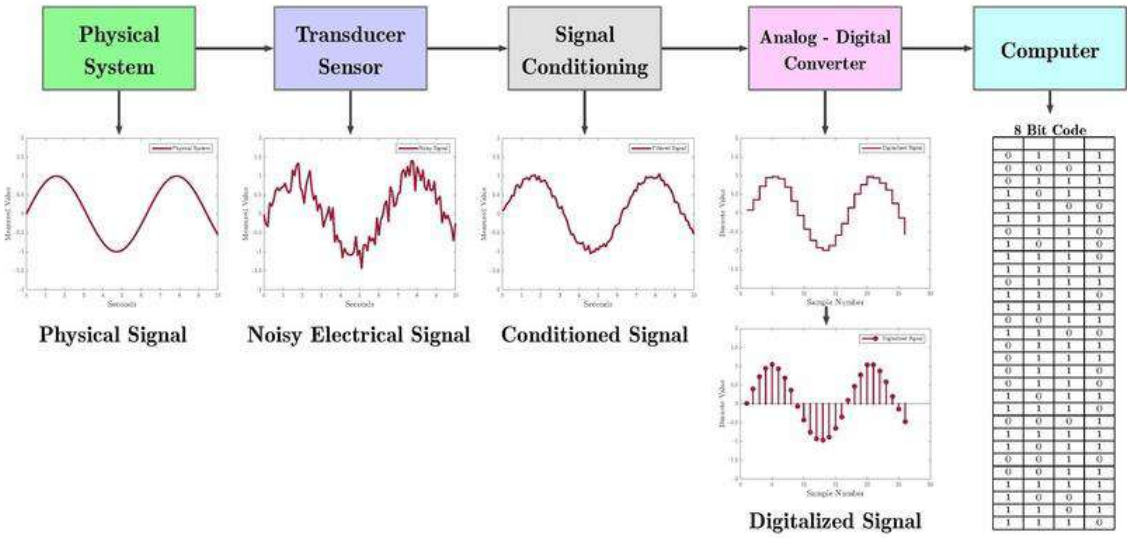


Diagrama de tiempo de apagado de IGBT genérico.

Además, como se mencionó anteriormente, puesto que su tiempo de conmutación es del orden del microsegundo, será necesaria también una espera comparable entre la conmutación de cada una de las llaves que componen una rama para evitar la condición de cortocircuito, la cual se resolverá añadiendo a la secuencia de conmutación PWM cierto tiempo de espera, llamado tiempo muerto.

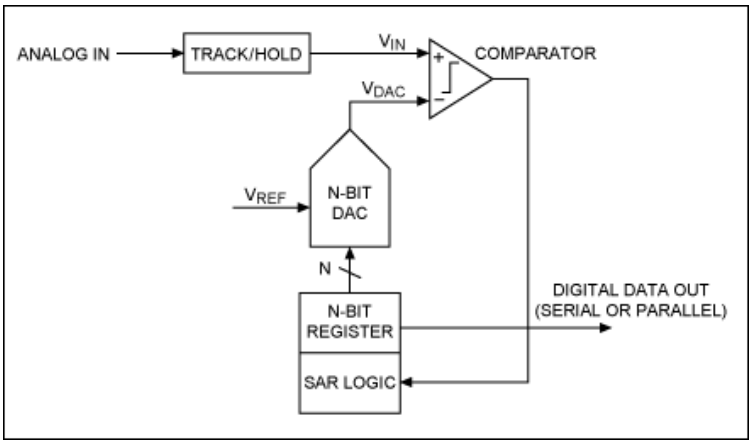
ADCs y adquisición de magnitudes:

Los ADC o conversores analógico-digital son circuitos electrónicos cuyo objetivo es la toma de información de una magnitud analógica de tiempo continuo como las que se encuentran en el mundo real y su conversión a una señal digital discreta tomando muestras con un ancho de banda limitado. Como características principales de un ADC se puede mencionar el ancho de banda de este y su resolución, y por lo tanto su SNR o relación señal ruido. Otros factores a tener en cuenta son la velocidad de muestreo, velocidad de conversión, tipo de conversor, linealidad, entre otras.

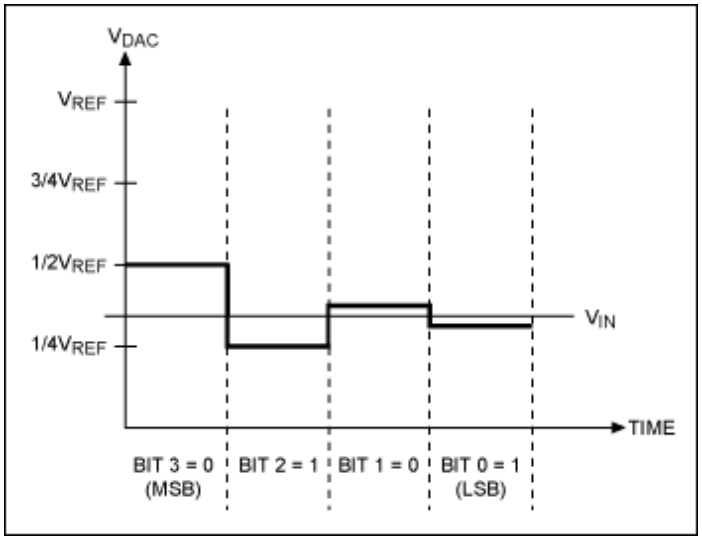


Cadena de adquisición digital genérica de una señal.

El tipo de ADC utilizado para este sistema es el SAR (registro de aproximaciones sucesivas) el cual funciona en base a un algoritmo de búsqueda que converge en el valor de la señal de entrada, haciendo comparaciones desde el MSB (bit más significativo) al LSB (bit menos significativo) del conversor.



Arquitectura simplificada de un ADC tipo SAR.



Principio de operación básico del SAR.

Como se mencionó anteriormente, el sistema desarrollado es un sistema de control realimentado y por ende es necesario disponer de información para poder implementar la realimentación y computar una acción de control. Puesto que el sistema de control esta implementado digitalmente, se requiere la conversión de las magnitudes analógicas de interés (en este caso, las tensiones y corrientes de salida y la tensión de entrada) y su ingreso al algoritmo de control para ser utilizado por este.

Además de esto, es necesario mapear los valores posibles que entrega el conversor de tal forma que estos representen de forma útil las señales que se desean medir. Otras dificultades a superar son que la magnitud que se desea medir no sea compatible con la entrada de un ADC, el cual generalmente espera un voltaje de entrada acotado, o que se desee que estas magnitudes no estén acopladas de manera directa al circuito electrónico de conversión.

Para ello, se utilizaran una serie de dispositivos transductores o sensores, que entregan una tensión o corriente proporcional a la magnitud que se desea medir utilizando diferentes principios físicos y dependiendo de la aplicación en particular se emplearan dispositivos que aíslen las diferentes etapas del sistema o que acondicionen las magnitudes de tal forma que estas sean compatibles con las entradas de los conversores utilizados. Las características a tener en cuenta en los sensores son su sensibilidad, linealidad, resolución y ancho de banda.

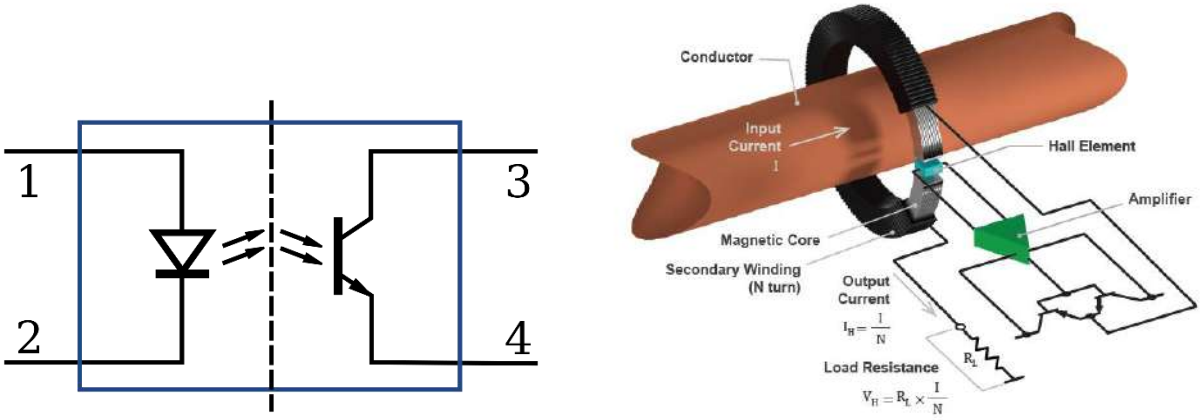


Diagrama de opto-acoplador y sensor de efecto hall tipo closed-loop.

En este caso particular, se utilizaron divisores resistivos y opto acopladores para la toma de valores de tensión y aislación galvánica entre la etapa de potencia y la de medición, y para las mediciones de corriente se utilizaron sensores de tipo efecto hall a lazo cerrado. Dichos sensores generan una corriente secundaria proporcional a la corriente primaria de la etapa de potencia, que luego al utilizar un shunt de valor conveniente, proveen de una tensión que indica la corriente primaria, con la ventaja de proveer de aislación eléctrica entre las corrientes primaria y secundaria sin necesidad de elementos adicionales.

Las magnitudes entregadas por los sensores generalmente son acondicionadas para su compatibilidad con el conversor analógico-digital utilizado en el sistema en cuestión, tanto desde el punto de vista de los valores admisibles de tensión de entrada de estos como así también desde el punto de vista del filtrado y ancho de banda de las señales que ellos entregan para lograr una adquisición funcional. Esto generalmente se logra implementando una etapa de amplificación y filtrado de señal antes de la adquisición propiamente dicha.

Una opción muy utilizada en este caso, es la de un filtro activo implementado en base a amplificadores operacionales, que sintetice un filtro y/o amplificador según corresponda en cada caso. En este caso en particular se utilizó un filtro activo implementado a partir de amplificadores operacionales. Con esto, se completa la cadena de un sistema de adquisición digital, que en este caso, se utiliza para la realimentación del sistema de control en cuestión.

4. Diseño e implementación de Hardware del Proyecto:

El objetivo de este capítulo es el de detallar el diseño desde el punto de vista del hardware utilizado para realizar la plataforma en cuestión, como así también las diferentes validaciones, cambios, y testeos realizados para la puesta en marcha del sistema.

El hardware consiste de una serie de PCBs, algunos previamente diseñados por los directores del proyecto y/o alumnos anteriores, ya sea en forma genérica o completa, y otros desarrollados por quien escribe, los cuales sintetizan todos los aspectos de control y funcionamiento de un convertidor de potencia en cuanto a señales de sincronismo y control, procesamiento, adquisición, etc., que permiten su interfaz respecto a los subsistemas de potencia ya disponibles en un panel de testeo con el que cuenta el LIC.

Como se mencionó anteriormente, la plataforma cuenta con cinco PCBs, los cuales realizan funciones específicas y se interconectan entre sí por un backplane que aporta la interconexión de los distintos PCBs como así también la alimentación de la electrónica utilizada.

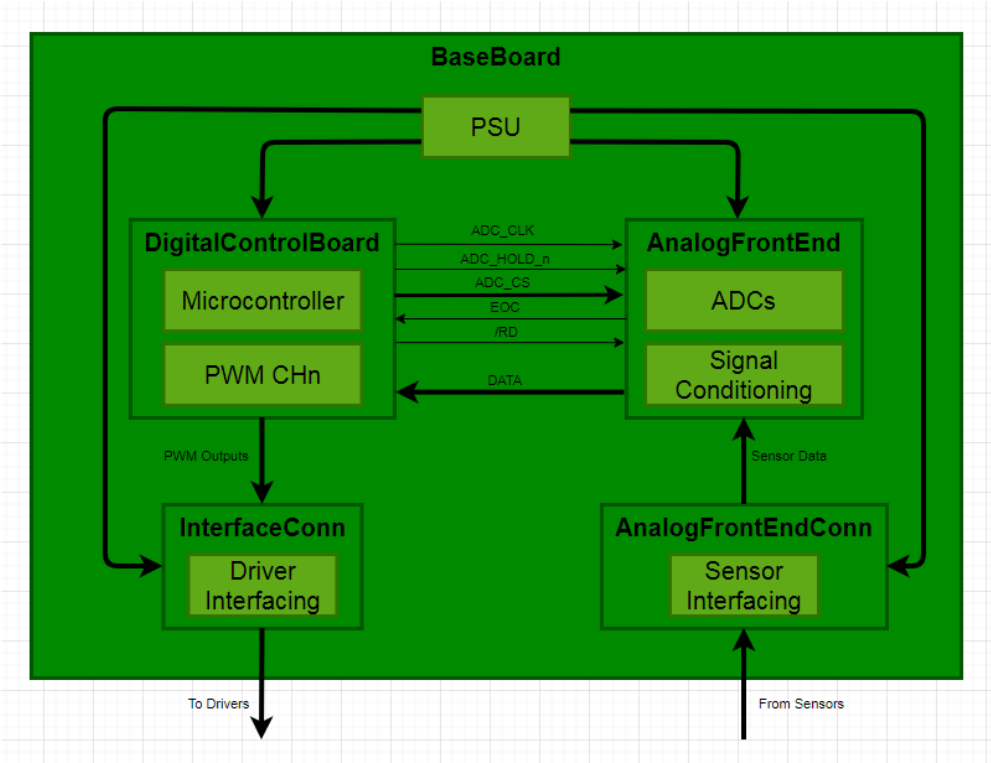


Diagrama de interconexión básica entre PCBs del proyecto.

Dichos PCBs fueron diseñados a nivel circuito impreso en el EDA (automatización de diseño electrónico) Altium Designer, utilizando el conjunto de herramientas y librerías que provee el software como así también generando una librería de componentes personalizados cuando se lo requiera.

Se procedió a popular cada uno de los PCBs con los componentes correspondientes a medida que el laboratorio fue proveyendo de estos en distintas compras y etapas del desarrollo. Inicialmente se comenzó por las placas **AnalogFrontEnd** y **DigitalControlBoard** para testear distintos sub-sistemas que podían desarrollar sus funciones o podían testearse funcionalmente de manera independiente, y utilizando fuentes de alimentación externas provistas por el laboratorio.

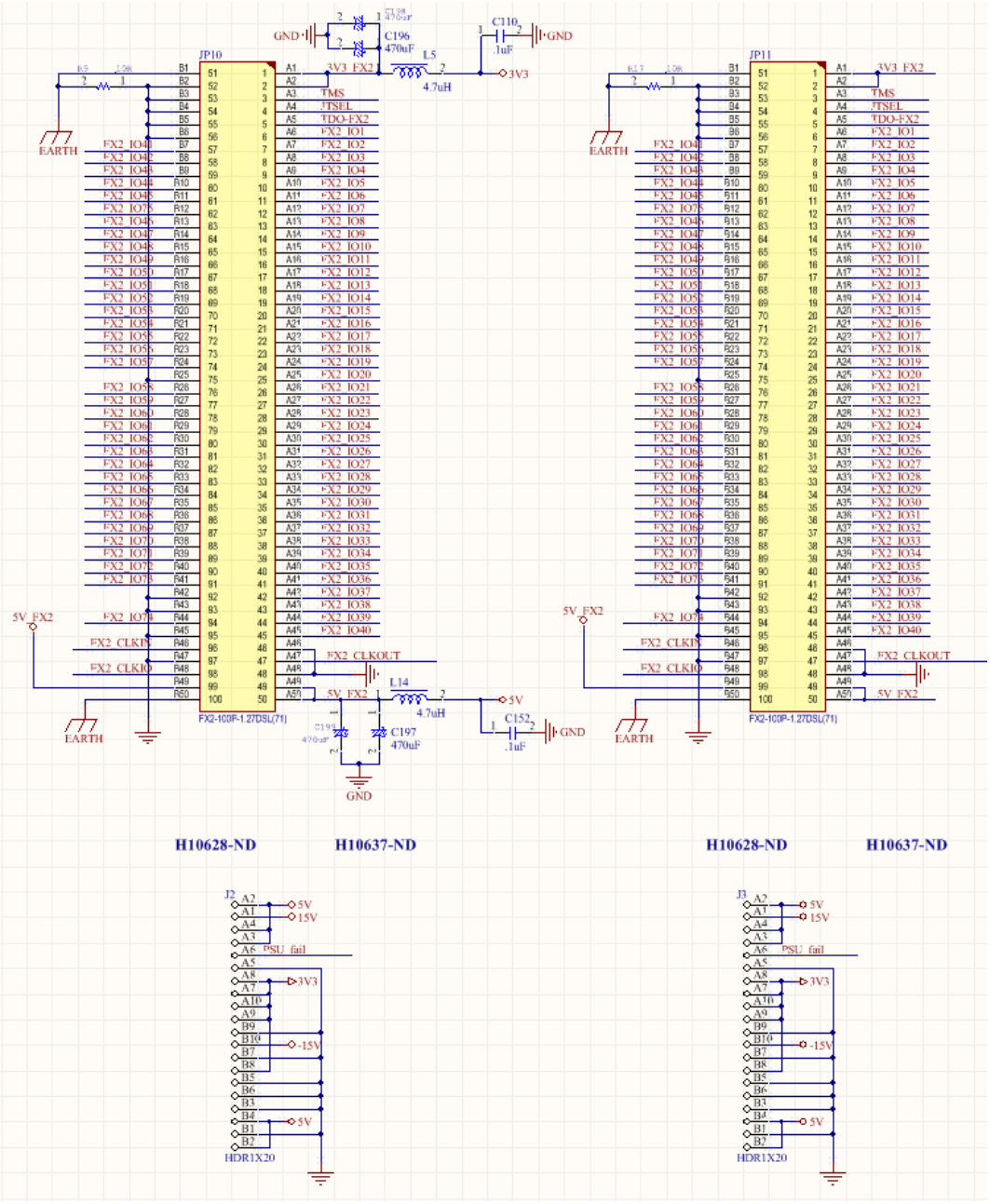
Luego se montó el PCB **BaseBoard** en cuanto se empezaron a testear sistemas a mayor nivel y el intercambio de señales entre PCBs, y por último se armaron las interfaces respecto a los componentes del sistema externos a la plataforma de control tales como los drivers controlados por la **InterfaceConn** y la **AnalogFrontEndConn** que permite la interconexión con las etapas de medición y de potencia ya montadas sobre el panel de testeo del laboratorio.

A medida que se fueron desarrollando las distintas etapas del proyecto y se fue pasando del diseño teórico al prototipo funcional, fueron surgiendo una serie de problemáticas a resolver, desde errores de diseño de arrastre originados por falta de testeo previo y/o mal ruteo, falta de información o esquemáticos erróneos, hasta necesidades de adaptar o modificar ciertos sub-sistemas para lograr el funcionamiento íntegro de todo el conjunto.

Además, también fue necesario testear y validar las etapas críticas y funciones del prototipo para asegurar el funcionamiento del sistema y poder verificar que este sea utilizado para su propósito de investigación y testeo de algoritmos de control por el laboratorio de control y cumpla con el objetivo de inyectar energía a la red eléctrica.

BaseBoard:

El PCB llamado **BaseBoard** es el backplane o PCB que hace de soporte para los circuitos impresos que realizan las tareas enumeradas anteriormente. Este cuenta con conectores HIROSE-FX2, por donde pasan todas las señales de interés entre los PCBs del assembly para el funcionamiento del sistema.



Parte del esquemático de conexión entre PCB **BaseBoard** y otros PCBs del proyecto.

Se encontraron errores de diseño en el ruteo del circuito esquemático original, lo cual generaba una incompatibilidad en la comunicación entre ciertos slots del **BaseBoard** debido a un desfasaje entre las señales que iban entre los conectores Hirose FX-2 por lo que se realizó una modificación para subsanar la situación y permitir una comunicación adecuada entre las señales de los distintos PCBs montados sobre el backplane. Dicha modificación generó que las posiciones de los distintos PCBs queden fijadas en los distintos slots según un orden determinado para poder lograr que las señales se transmitan entre ellos correctamente.

En dicho PCB, además se generan las tensiones de alimentación necesarias para suplir al resto de la electrónica del sistema. Esto se genera a partir de una tensión de $24\ V$ tanto de continua como de alterna (en este caso pasan por un rectificador integrado en el backplane para lograr la tensión continua) que se utiliza de entrada para cuatro fuentes switching implementadas en base al IC (circuito integrado) LM22678 y la circuitería auxiliar que este requiere para funcionar. Este presenta la función de regulador step-down conmutado, donde la tensión de salida es dependiente de un divisor resistivo utilizado para realimentar al IC.

8.2 Typical Applications

8.2.1 Typical Buck Regulator Application

Figure 8-2 shows an example of converting an input voltage range of 5.5 V to 42 V, to an output of 3.3 V at 5 A.

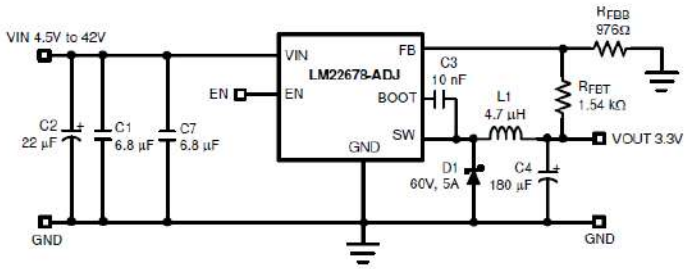


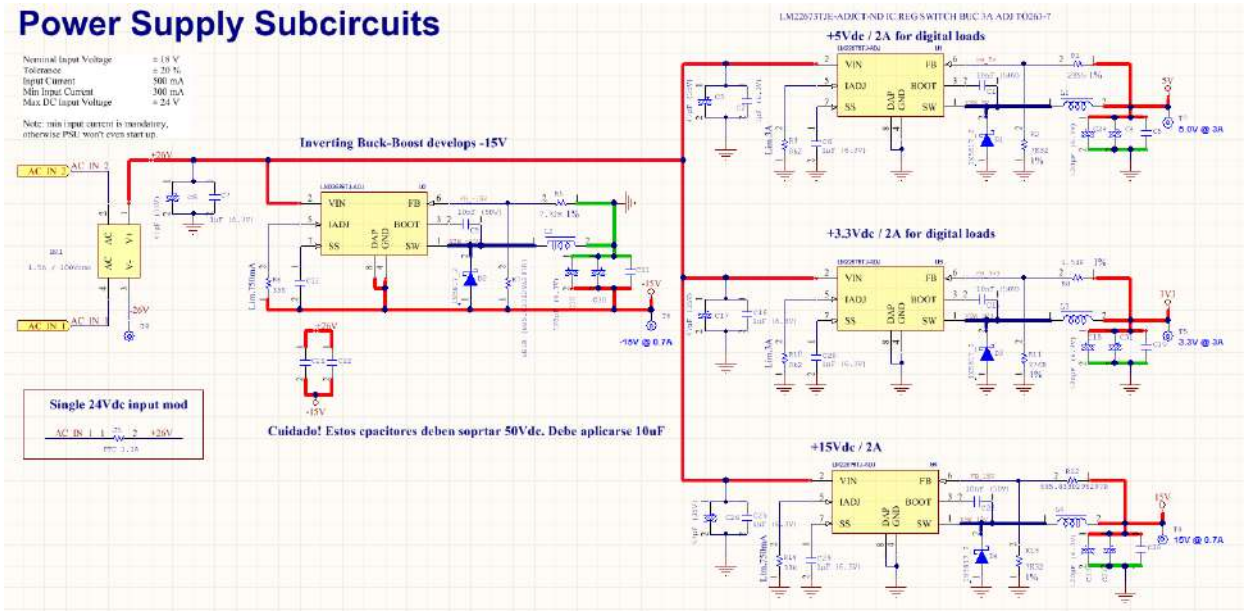
Figure 8-2. Typical Buck Regulator Application

8.2.1.1 Design Requirements

DESIGN PARAMETERS	EXAMPLE VALUE
Driver Supply Voltage (VIN)	4.5 to 42 V
Output Voltage (VOUT)	3.3 V
R_{FBT}	Calculated based on R_{FBB} and V_{REF} of 1.285 V.
R_{FBB}	1 kΩ to 10 kΩ
I_{OUT}	5 A

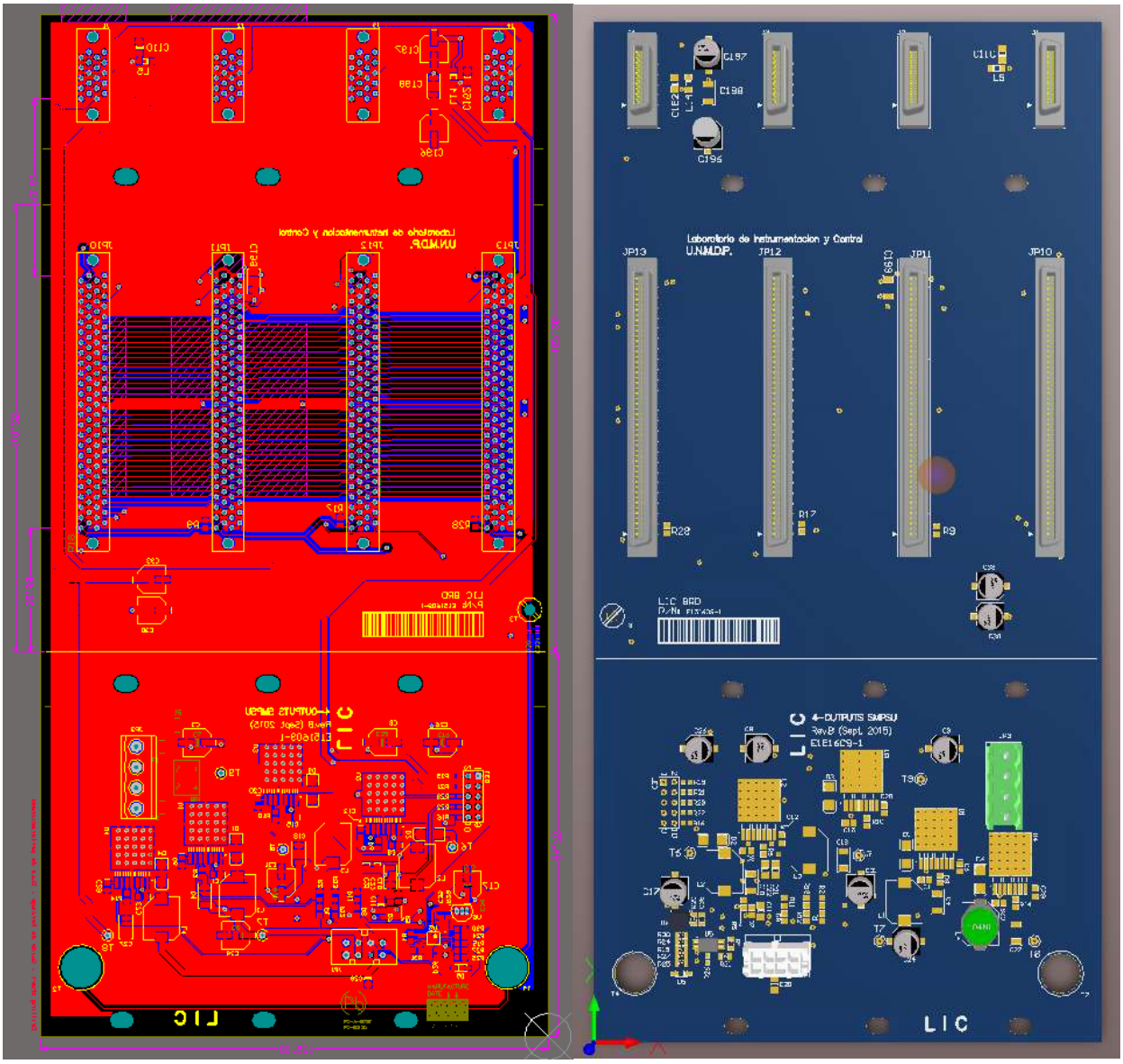
Diagrama utilizado para el diseño de las fuentes de alimentación en base al LM22678.

En base a la información provista en la application note (nota de aplicación) del IC, se diseña el circuito y se seleccionan los componentes discretos auxiliares de manera conveniente para lograr las tensiones requeridas por el resto de la electrónica del assembly. Dicho esquema es replicado en cada una de las fuentes de alimentación para lograr las tensiones de $+5\,V;\ +3.3\,V;\ +15\,V;\ -15\,V$ de forma simple y compacta.



Circuito esquemático de fuentes en PCB **BaseBoard** para lograr las tensiones antes mencionadas.

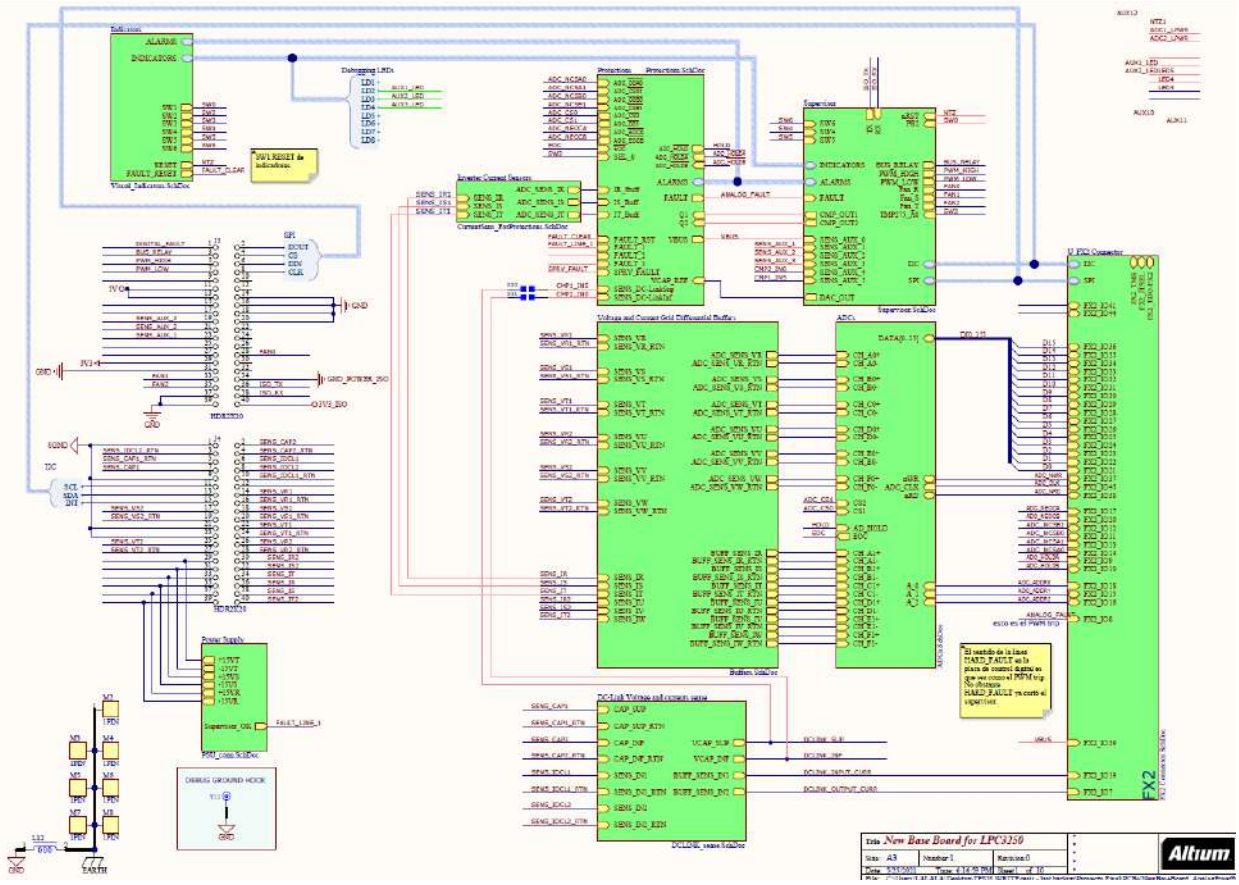
Se realizó un re-work sobre una de las fuentes por un error en el diseño original en el esquemático que no permitía generar uno de los rails de alimentación correspondiente de manera adecuada. Esta falla se resolvió luego del re-ruteo correspondiente y el corte del trace correspondiente. Por último, cada fuente se verifico experimentalmente con carga para asegurar que no hubiera una caída que pudiera generar alguna falla en etapas posteriores del proyecto.



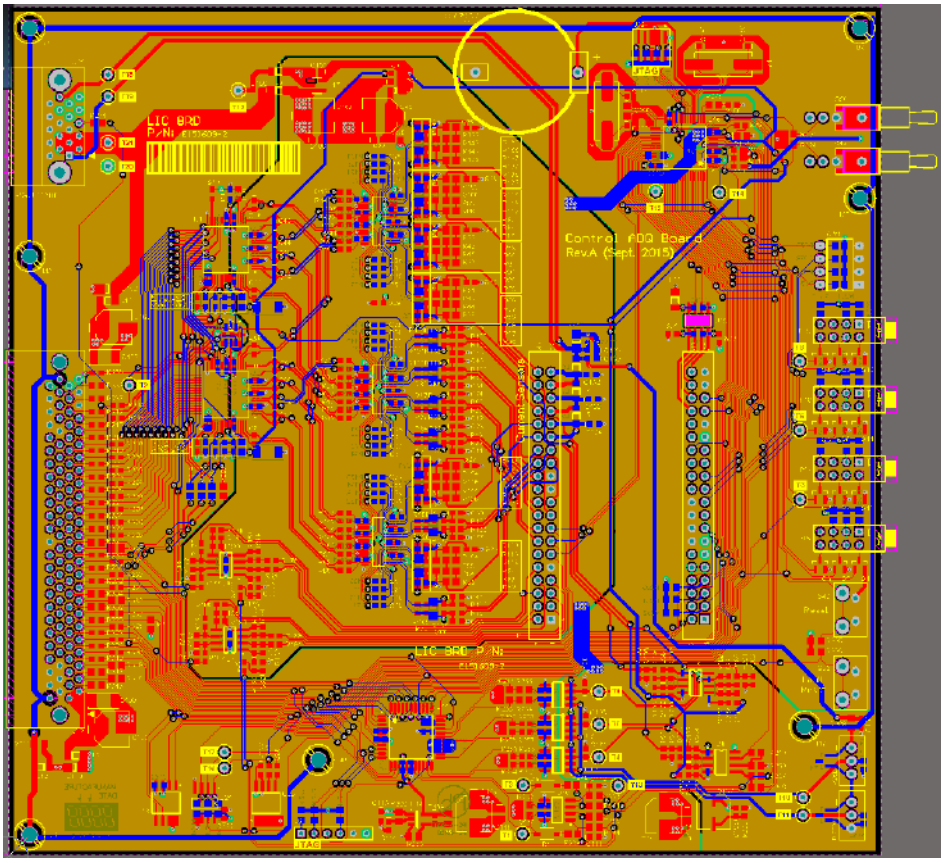
Layout de PCB **BaseBoard** y su render 3D en Altium Designer.

AnalogFrontEnd:

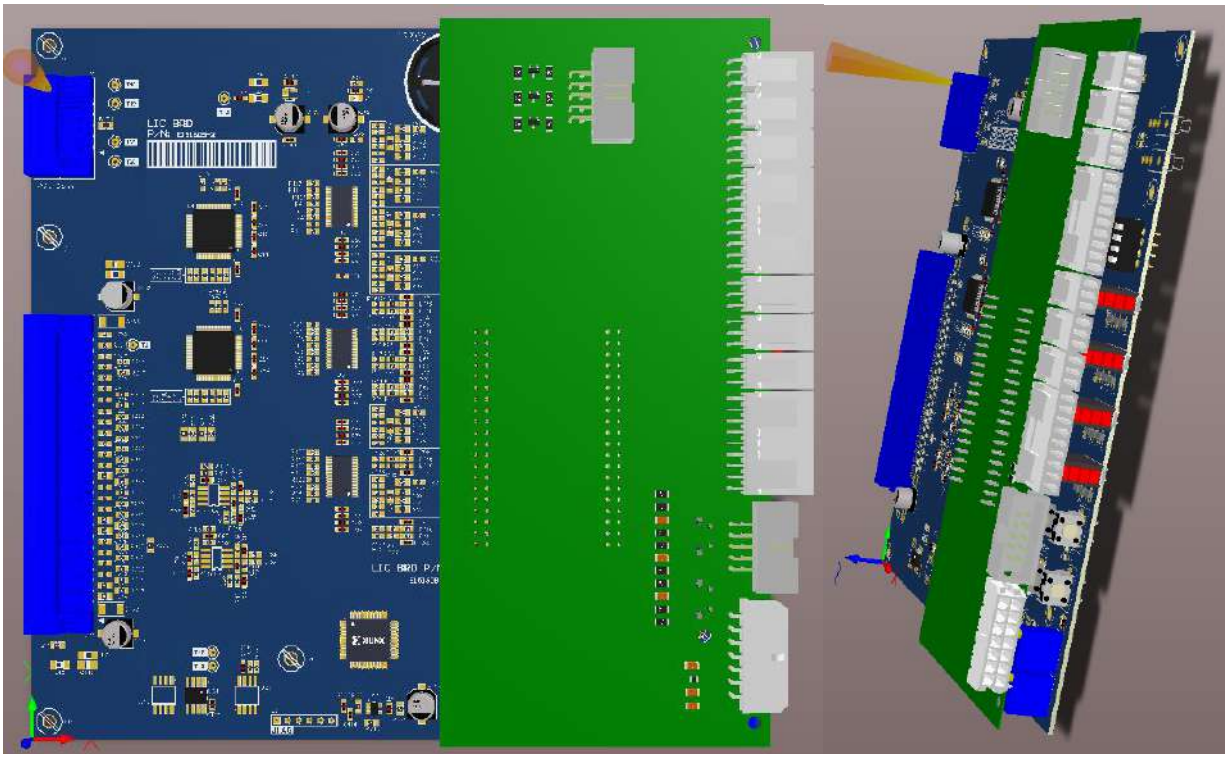
El PCB **AnalogFrontEnd** contiene el hardware utilizado para el acondicionamiento y la adquisición de las magnitudes de interés para el convertidor previamente mencionadas, además de la electrónica para sintetizar las protecciones de hardware en caso de una falla del sistema. Cabe destacar que en el diseño general del assembly está contemplada la posibilidad de expansión de la cantidad de magnitudes a medir agregando una segunda **AnalogFrontEnd** en el caso de que la topología o el diseño del convertidor así lo requieran.



Esquema general de PCB **AnalogFrontEnd**.



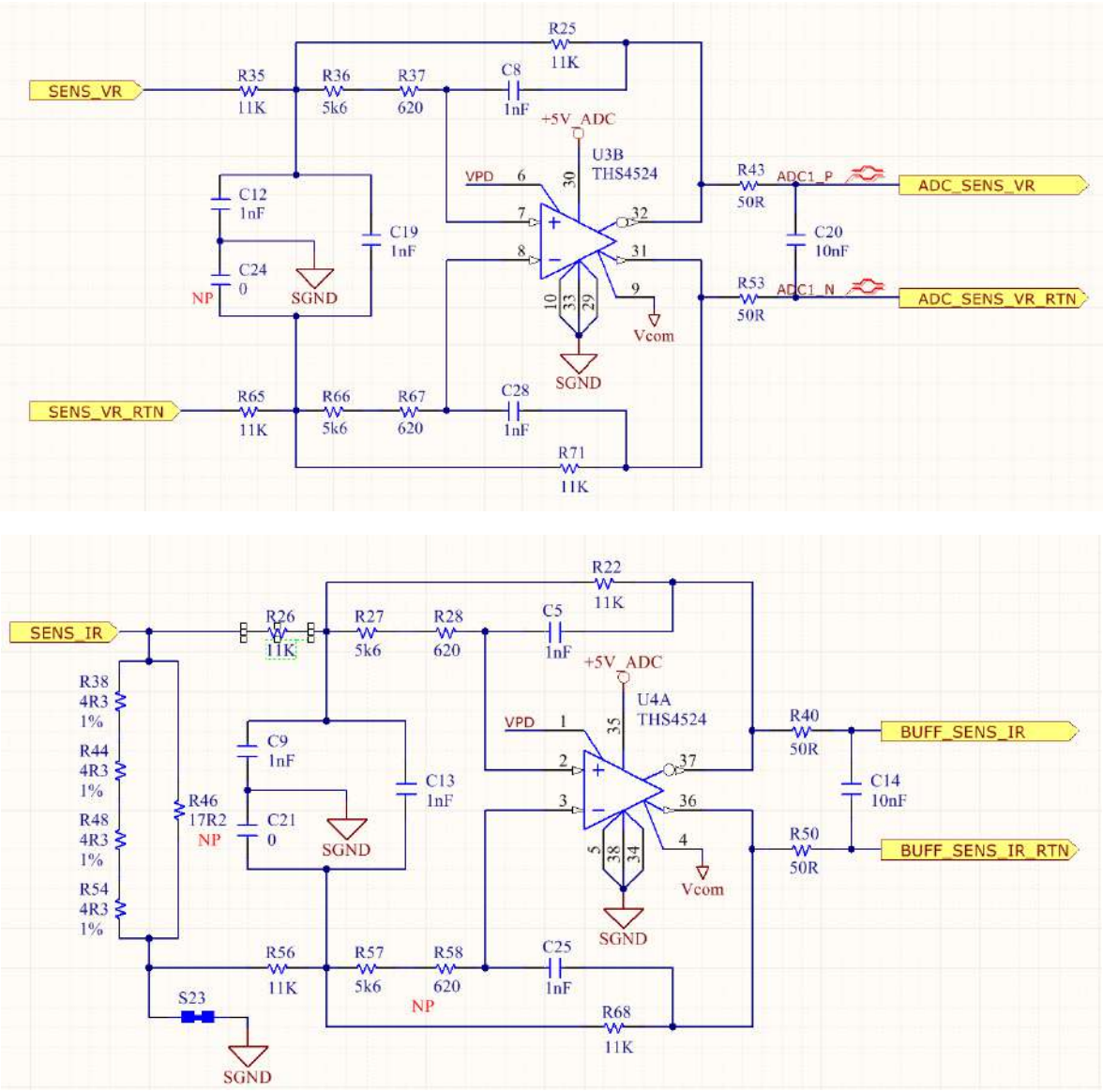
Layout de PCB **AnalogFrontEnd**.



Render de PCB **AnalogFrontEnd** con **AnalogFrontEndConn** montada piggyback.

Acondicionamiento de señales:

Este PCB cuenta con doce canales de adquisición diferenciales (preparados para recibir seis tensiones y seis corrientes). Dichas magnitudes pasan en primera instancia por una etapa de acondicionamiento y filtrado diferencial implementada en base al IC THS4524. Dicho integrado posee cuatro amplificadores operacionales rail-to-rail completamente diferenciales que son utilizados para sintetizar un filtro Butterworth o de máxima planicidad utilizando la topología MFB (multiple feedback o realimentación multiple). Estos filtros se utilizan para anti-aliasing y disminución del ruido en la magnitud medida para una correcta adquisición en la etapa posterior además de adaptar las señales tanto en nivel de continua como en salida diferencial para compatibilizarla con la etapa siguiente.



Circuito esquemático con topología MFB para medición de Tensión y Corriente.

Este diseño se realizó teniendo en cuenta las consideraciones descriptas en la nota de aplicación Fully-Differential Amplifiers de Texas Instruments, de las que se puede obtener un análisis sobre el ruido en la medición, el rechazo mayor a CMRR, la disminución de la distorsión de armónicos de segundo orden y demás parámetros del filtro.

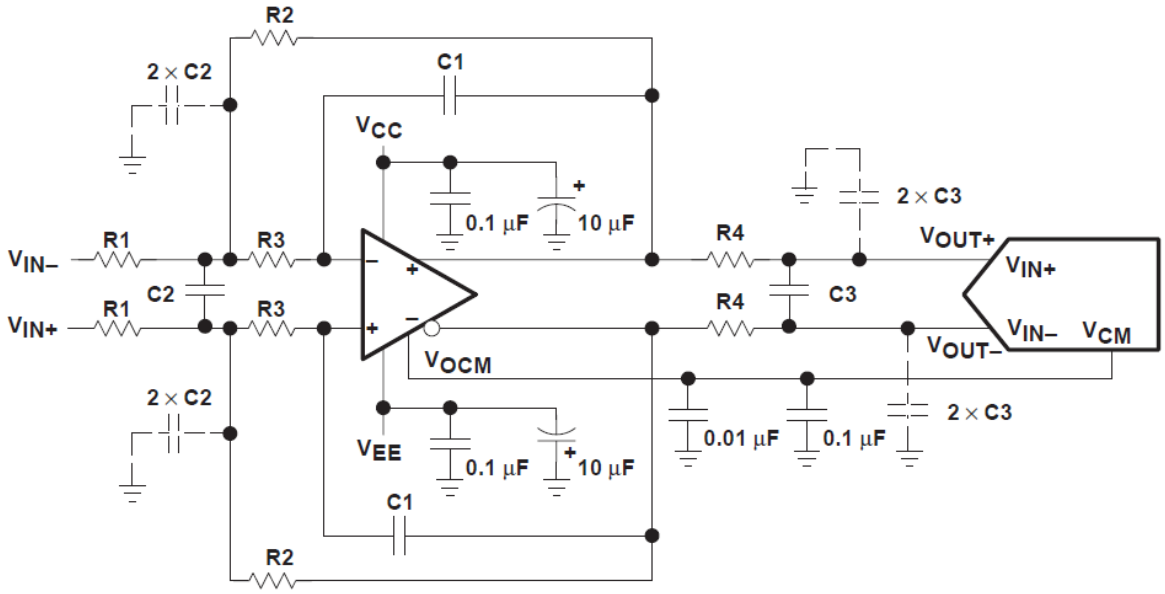


Figure 28. Third-Order Low-Pass Filter Driving an ADC

The transfer function for this filter circuit is:

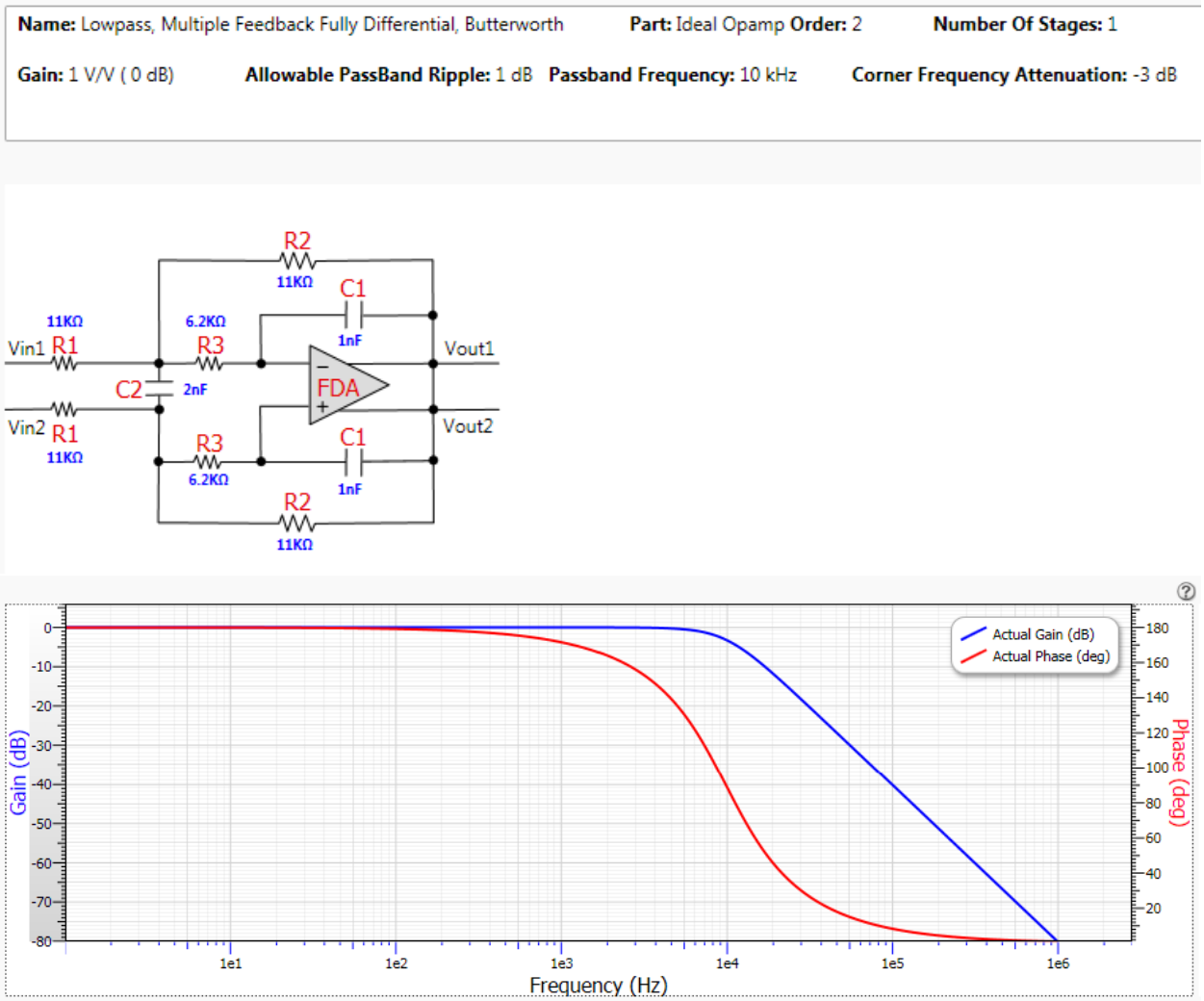
$$\frac{Vod}{Vid} = \left(\frac{K}{-\left(\frac{f}{FSF \times fc}\right)^2 + \frac{1}{Q}\,\frac{jf}{FSF \times fc} + 1} \right) \times \left(\frac{1}{1 + j2\pi f \times 2 \times R4C3} \right),$$

where $K = \frac{R2}{R1}$, $FSF \times fc = \frac{1}{2\pi\sqrt{2 \times R2R3C1C2}}$, and $Q = \frac{\sqrt{2 \times R2R3C1C2}}{R3C1 + R2C1 + KR3C1}$.

Extracto de nota de aplicación sobre amplificadores operacionales diferenciales, con análisis de función transferencia para topología MFB.

Este se sintetizo a partir del software Filter Pro de Texas Instrument. Luego de seleccionar la topología y tipo de filtro a utilizar. Dicho software permite observar y evaluar la respuesta en frecuencia de dicha topología y tipo de filtro a utilizar en base a valores de diseño seleccionados y teniendo en cuenta la tolerancia y valor de los componentes pasivos utilizados para la posterior realización. Además, se hicieron coincidir los valores de los componentes pasivos, midiendo de forma manual cada uno de ellos para disminuir el offset que introduce un desajuste entre estos para este tipo de topología. El diseño fue realizado teniendo en cuenta una frecuencia de corte de aproximadamente $10\ KHz$ y máxima planicidad en la banda de paso del filtro.

Por último, la terminación de salida se diseña teniendo en cuenta que el polo que esta agrega al sistema no afecte al filtrado previamente mencionado y se acople de manera adecuada al adquisidor, para este caso se diseña alrededor de los $100\ KHz$.



Diseño, valores de componentes y respuesta del filtro obtenido en base a los parámetros y consideraciones indicadas en banda de paso de interés.

Los amplificadores mapean las magnitudes de interés medidas por los PCBs provistos por el LIC donde se encuentran los sensores y etapas de aislación. En el caso de las tensiones, esto se realiza a partir de divisores resistivos opto-acoplados, los cuales al pasar por la etapa de acondicionamiento de señal quedan en el rango de $0\ a\ 5\ V$ de tensión. Para las mediciones de corriente, los transductores utilizados son los LEM LA 55-P/SP1, sensores de efecto Hall ya aislados por su misma construcción, los cuales se conectan de forma directa a la etapa de acondicionamiento de señal y donde solo debe ajustarse la ganancia de la medición mediante resistores shunt en la entrada diferencial, los cuales deben lograr una excursión de tensión correspondiente a la corriente que se desea medir. Para este caso, se desea que quede mapeado en el intervalo de $\pm\ 2.5\ V$ respecto a tierra para una corriente máxima de 40 A, de lo cual se deduce que $R_{shunt} \cong 17.2\ \Omega$.

Puesto que un error de ruteo en la etapa de salida de los amplificadores de los canales de tensión invertía la polaridad de las mediciones a ser adquiridas por el ADC, hubo que buscar una forma de salvar este problema. Se optó por una corrección vía firmware de los valores adquiridos puesto que al estar intercambiada la polaridad en la salida de algunos amplificadores, se obtenía un valor adquirido con la fase contraria a la real pero con una magnitud correcta.

Toda la etapa descripta anteriormente fue ensayada y validada de manera experimental primero utilizando un banco de testeo comprendido por un generador de señales, osciloscopio, los circuitos de amplificación previamente descriptos y alimentados por la fuente del PCB BaseBoard y una referencia de $+2.5\ V$ y luego también con la aplicación funcionando y el sistema completo. En dicho testeo se pudo convalidar el correcto funcionamiento de cada uno de los canales de acondicionamiento de señal, la excursión de los amplificadores operacionales y también si la respuesta en frecuencia respondía al diseño teórico del filtro. Los resultados obtenidos fueron los esperados y solo se calibro de manera fina las ganancias vía firmware para mapear y equiparar las mediciones de cada canal en particular.

Adquisición:

La etapa posterior a la de acondicionamiento de señal es la de adquisición propiamente dicha, implementada a partir de un ADC. Esta se realizó a partir del IC ADS8365 el cual posee entradas diferenciales compatibles con la etapa anterior. Este ADC del tipo SAR (succesive approximation register, registro de aproximaciones sucesivas) dispone de seis canales de $16\ bits\ y\ 250\ KSa/s$ y una interfaz paralela de alta velocidad para la lectura y almacenamiento de las magnitudes adquiridas. En este diseño, se distribuyeron tres tensiones y tres corrientes asociado cada par tensión-corriente, a una fase determinada en cada uno de los circuitos integrados disponibles en el PCB. Estos ADCs adquieren las señales de interés a una frecuencia de $30\ KHz$ impuesta en el diseño del inversor, dirigidos por las señales de control generadas en el PCB **DigitalControlBoard**.

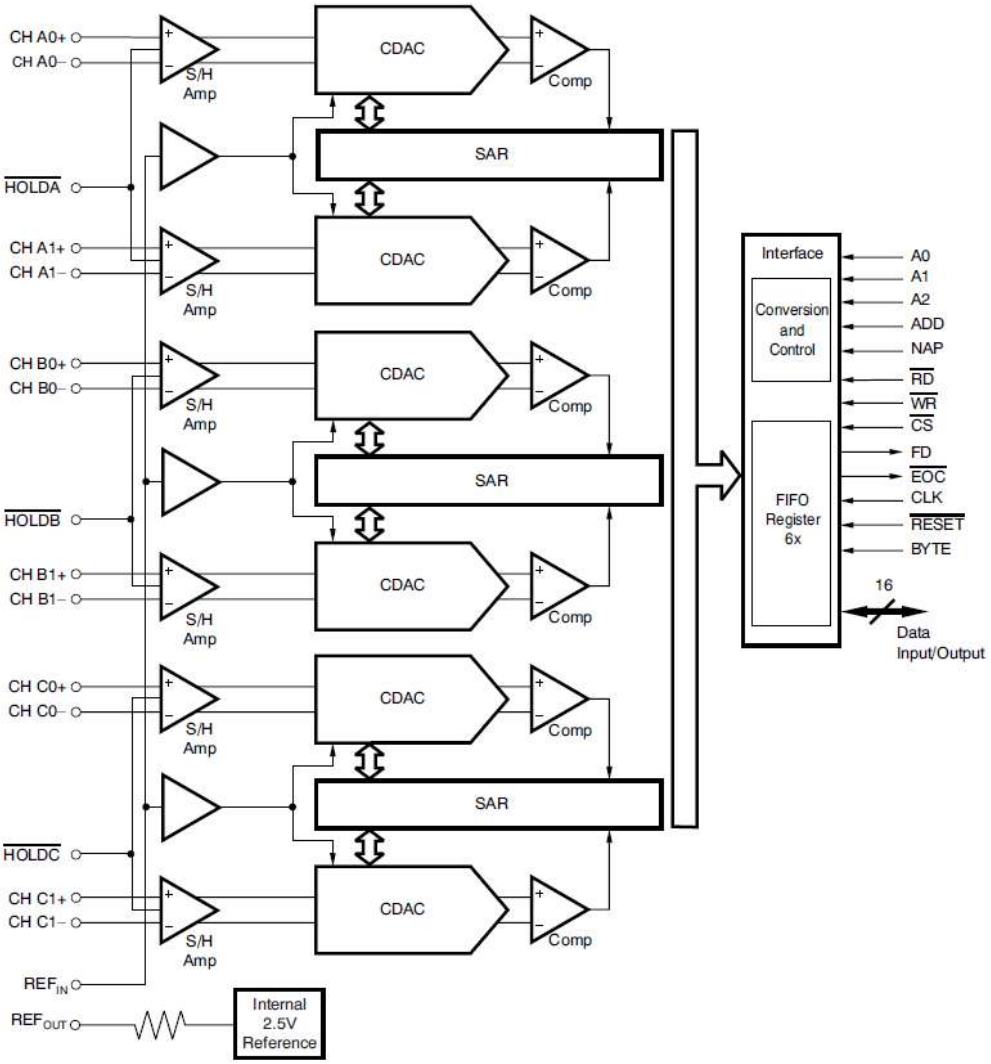
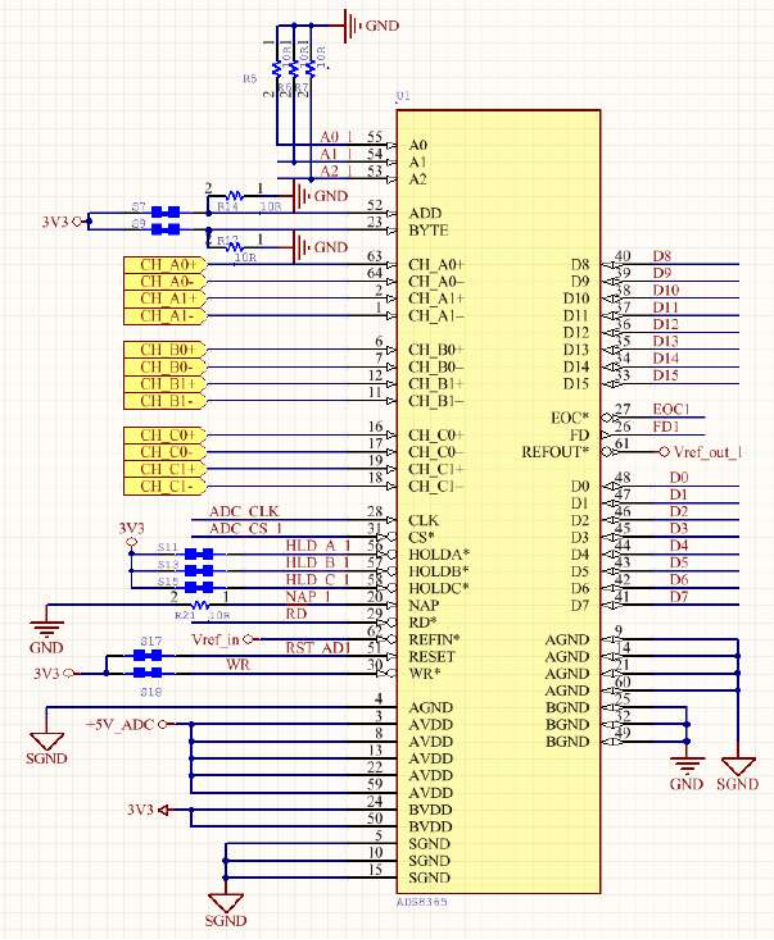
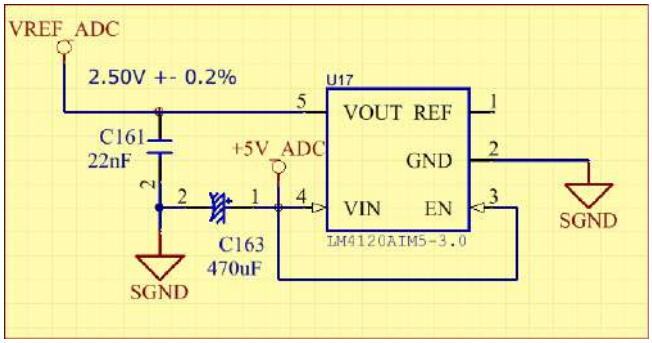


Diagrama en bloques de ADS8365.



Extracto de circuito esquemático, etapa de adquisición

Estos ADCs requieren de una referencia de $2.5\,V$ para funcionar correctamente. Puesto que los adquisidores utilizados cuentan con una referencia interna, pero esta solo posee una corriente de cortocircuito de $1.25\,mA$ la cual es insuficiente para utilizarse de forma segura, se dispuso de la utilización de una referencia de tensión generada a partir del IC LM4120AIM5-3.0 el cual puede generar una tensión de $2.5\,V\ \pm 0.2\,\%$ al utilizarse junto a su circuitería auxiliar detallada en su hoja de datos. Dicho valor fue verificado posteriormente en la implementación y cumplía con la especificación dada.



Circuito esquemático de referencia de tensión para adquisición.

Los ADS8365 disponen de una serie de requerimientos en cuanto a las señales, y tiempos de hold y conversión que deben ser respetados. Cabe destacar que por un problema de hardware en el módulo que se pensó utilizar originalmente para resolver el acceso, lectura y comunicación entre el adquisidor y el microcontrolador que maneja la operación del sistema, y a modo de no perder velocidad en el clock del sistema con su correspondiente perdida de performance (el módulo FlexBus utilizado para dicha función posee un clock que está relacionado de manera proporcional al system clock y este puede ser dividido hasta un cierto punto) se emplean los ADCs con un clock superior al máximo especificado de $5\ MHz$ en la hoja de datos del circuito integrado y en cambio estos funcionan a $7.5\ MHz$. El funcionamiento a esta frecuencia superior a la máxima especificada en la hoja de datos, se verifico de forma exhaustiva en la validación y testeo experimental del sistema para comprobar que sea factible utilizar los ICs de esta manera. En el caso del resto de los tiempos especificados en la hoja de datos, todos ellos se respetan a modo de asegurar el correcto funcionamiento del sistema.

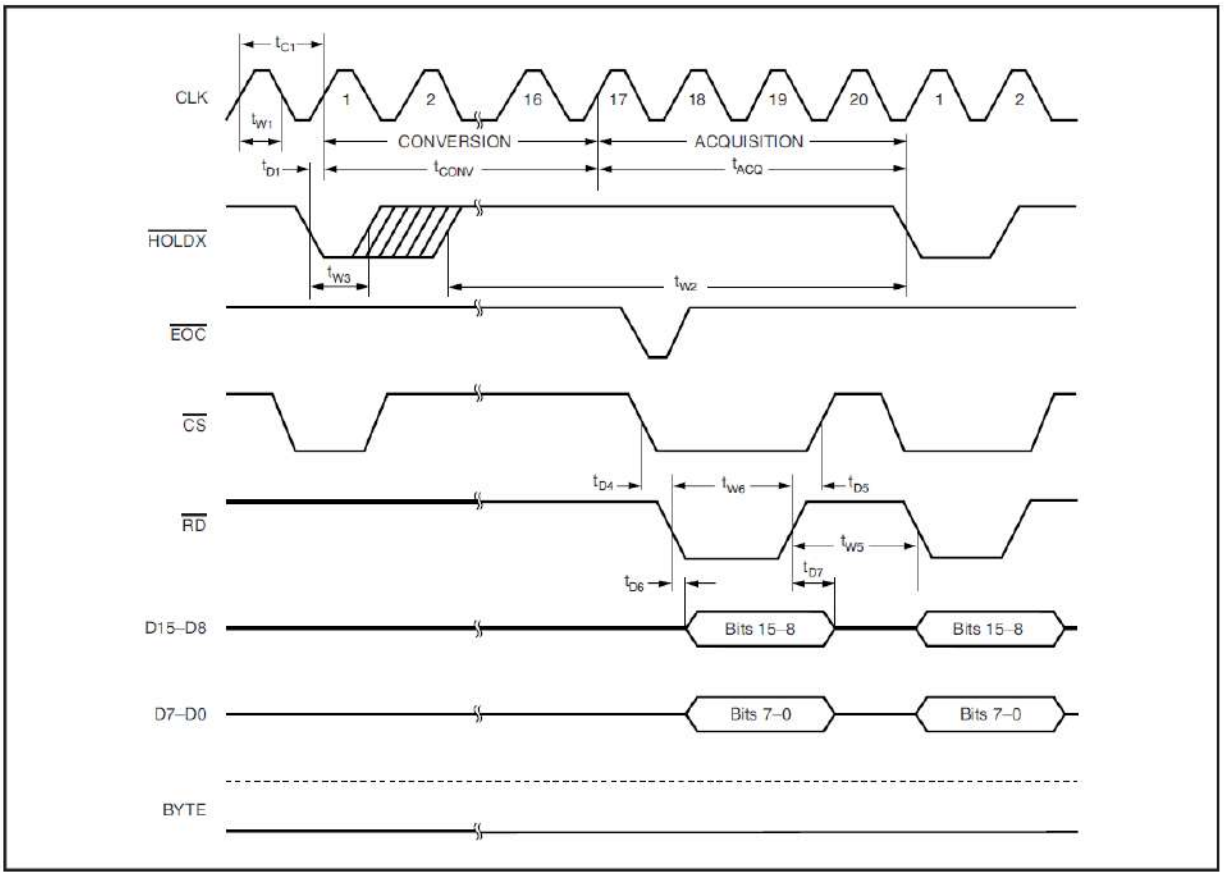


Diagrama de tiempos de ADS8365 para un ciclo de conversión.

Donde se puede observar que una operación de adquisición está compuesta por una etapa de **HOLD/CONVERSIÓN** indicada por una señal de inicio de conversión, seguido de tiempo de procesamiento del circuito integrado en sí mismo, seguido de un aviso de que la conversión ha terminado correctamente y se pueden leer datos validos con la señal de EOC, y luego una etapa de **LECTURA** de datos donde se direcciona cada circuito integrado para leer un dato en particular que representa la muestra de la magnitud deseada en un instante de tiempo determinado. Las magnitudes adquiridas son transmitidas en forma de datos (words, de 16 bits) y leídas a través del bus paralelo ($D15..D0$) que posee el circuito integrado.

Las señales de control del circuito integrado son las siguientes:

- $\overline{RESET}$: Reset asincrónico utilizada para inicializar el sistema de forma determinística en el tiempo.

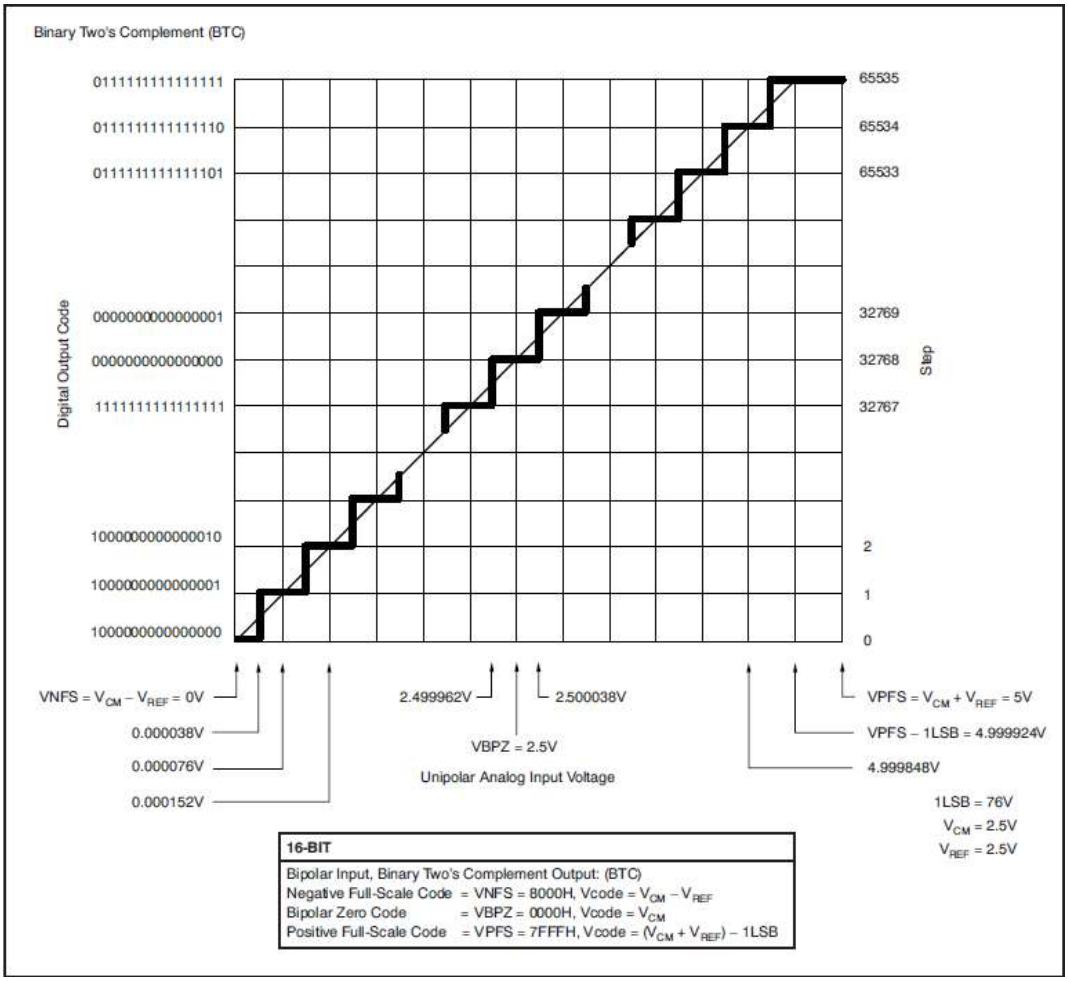
- $\overline{HOLDx}$ (A, B, C): Las cuales indican el momento de un nuevo comienzo de conversión de un par de canales del ADC. Luego del assert de esta señal de forma externa, se ignoran nuevos pulsos de HOLDx hasta que se completa la conversión.

- $\overline{EOC}$: Indica cuando la conversión ha sido completada por el IC y pueden leerse datos validos desde el adquisidor.

- $\overline{CSn}$: Chip Select, indica que se está accediendo a un circuito integrado en particular, utilizada en el inicio de conversión y en el proceso de lectura de datos.

- $ADDRESSn$ $(A0..A2)$: conjunto de señales que indican el direccionamiento de los canales para un circuito integrado específico. En el caso de la implementación realizada, se optó por utilizar un direccionamiento de tipo cíclico (cycle mode), donde la lectura de todos los canales se realiza de modo secuencial.

- $\overline{RD}$: señal de lectura utilizada en forma de strobe para indicar que se está ejecutando una operación de lectura de datos sobre el IC.



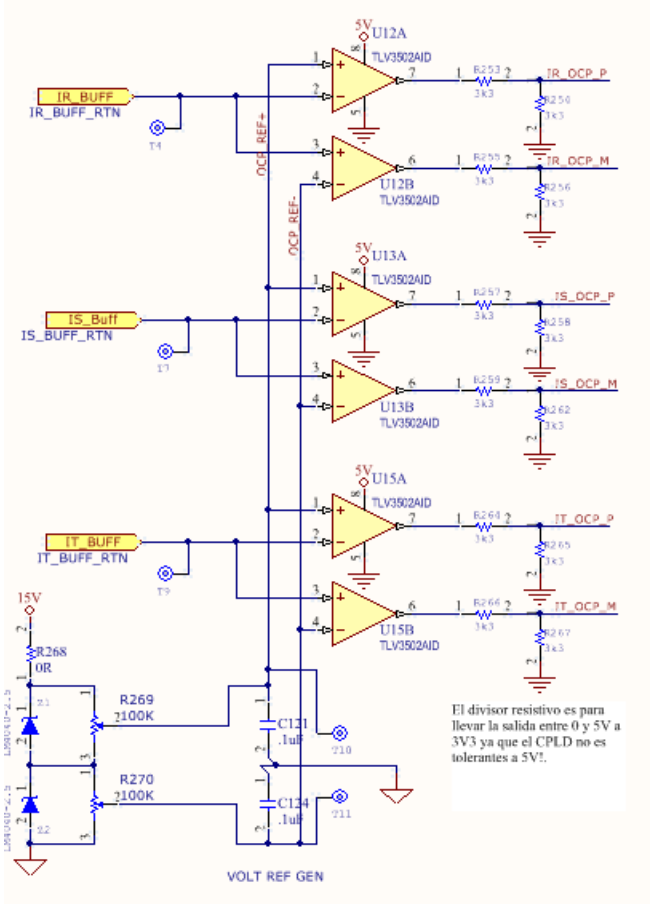
Característica de conversión ideal (single-ended; $V_{CM} = chXX;\ V_{REF} = 2.5\ V$).

Aquí se puede observar la secuencie de bits esperada para mediciones en particular, las cuales fueron utilizadas de patrón para la validación de los canales de adquisición ingresando con valores predeterminados o señales conocidas a la entrada de estos y así poder testear todo el conjunto de adquisición. Entre otras cosas se atacaron las entradas de adquisición tanto con valores fijos de tensión como con señales sinusoidales y se tomaron vectores de muestras para después graficarlos en MATLAB y observar si las señales fueron medidas exitosamente o se encontró alguna distorsión o problema de escala en ellas. También a partir de estas pruebas se pudo testear si el módulo FlexBus estaba funcionando correctamente en la lectura de datos.

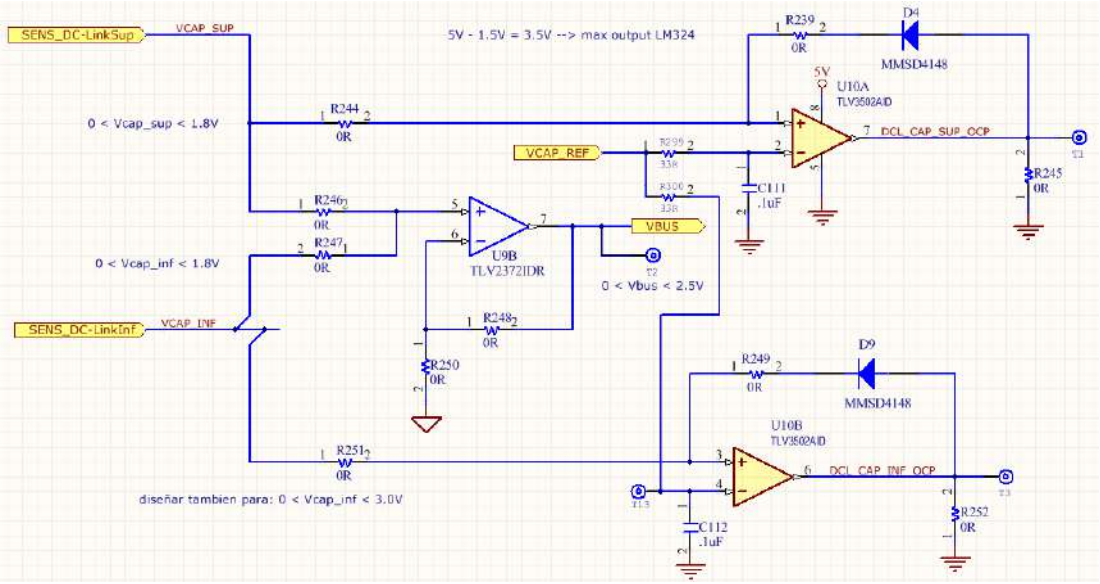
La etapa de adquisición se testeo y valido de forma experimental tanto en banco de prueba como funcionando como sub-sistema del inversor de potencia.

Protecciones de hardware e interfaz entre control y adquisición:

Como parte fundamental del sistema, es necesaria la implementación de una serie de protecciones, las cuales evitan que en caso de una falla en el assembly, esta produzca una falla catastrófica o roturas en componentes de costo elevado, particularmente en la etapa de potencia. Para esto, hay una serie de protecciones implementadas directamente sobre el hardware, utilizando cierta electrónica que accione e interrumpa las funciones del sistema y toda la etapa de potencia quede inactiva. Estas protecciones se implementaron en base a los comparadores TLV3502AID los cuales disponen de un umbral ajustable con un trimmer y son atacados por las magnitudes de potencia que indican una falla tanto por exceso de corriente en algunas de las fases inyectadas (I_R, I_S, I_T) como por un exceso de tensión en el DC bus del convertidor (DCB_{OV}). En caso de excederse el umbral calibrado para cada magnitud, se genera un vuelco de la salida, que se propaga a través de una lógica programable que funciona como interfaz, y esta inhabilita todas las funciones de salida del sistema.



Circuito esquemático de protecciones de corriente por hardware implementadas con comparadores en **AnalogFrontEnd**.



Circuito esquemático de protección de sobre-tensión implementado por hardware.

6.7 Switching Characteristics

At T_A = 25°C and V_S = 2.7 V to 5.5 V, unless otherwise noted.

PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
$T_{(pd)}$	Propagation delay time(1)(2)	ΔV_{IN} = 100 mV, Overdrive = 20 mV	At T_A = 25°C		4.5	6.4	ns
			At T_A = −40°C to +125°C			7	ns
		ΔV_{IN} = 100 mV, Overdrive = 5 mV	At T_A = 25°C		7.5	10	ns
			At T_A = −40°C to +125°C			12	ns
$\Delta t_{(SKEW)}$	Propagation delay skew(3)	ΔV_{IN} = 100 mV, overdrive = 20 mV			0.5		ns
f_{MAX}	Maximum toggle frequency	Overdrive = 50 mV, V_S = 5 V			80		MHz
t_R	Rise time(4)				1.5		ns
t_F	Fall time(4)				1.5		ns

6.8 Typical Characteristics

At T_A = 25°C, V_S = 5 V, and input overdrive = 100 mV, unless otherwise noted.

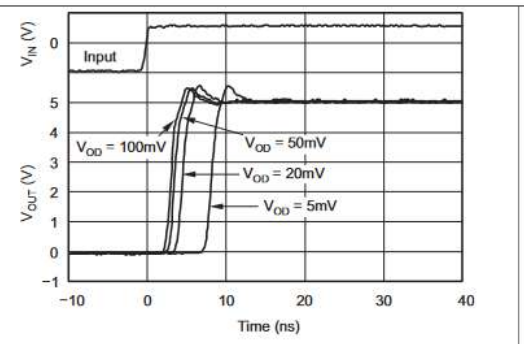


Figure 1. Output Response for Various Overdrive Voltages (Rising)

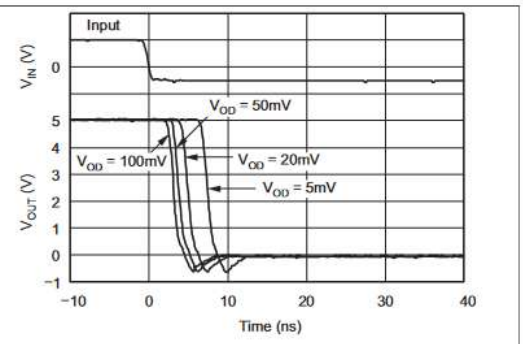


Figure 2. Output Response for Various Overdrive Voltages (Falling)

Tiempos de respuesta y características de tiempo del comparador TLV3502AID.

El PCB dispone de un CPLD de la familia CoolRunner II, fabricado por Xilinx, el XC2C64A-7VQG44C**,** dicho dispositivo de lógica programable es muy versátil puesto que este contiene celdas que permiten la implementación de diferentes funciones o interfaces mediante el uso de un lenguaje de descripción de hardware y cuenta con una gran cantidad de GPIOs (entradas-salidas de propósito general).

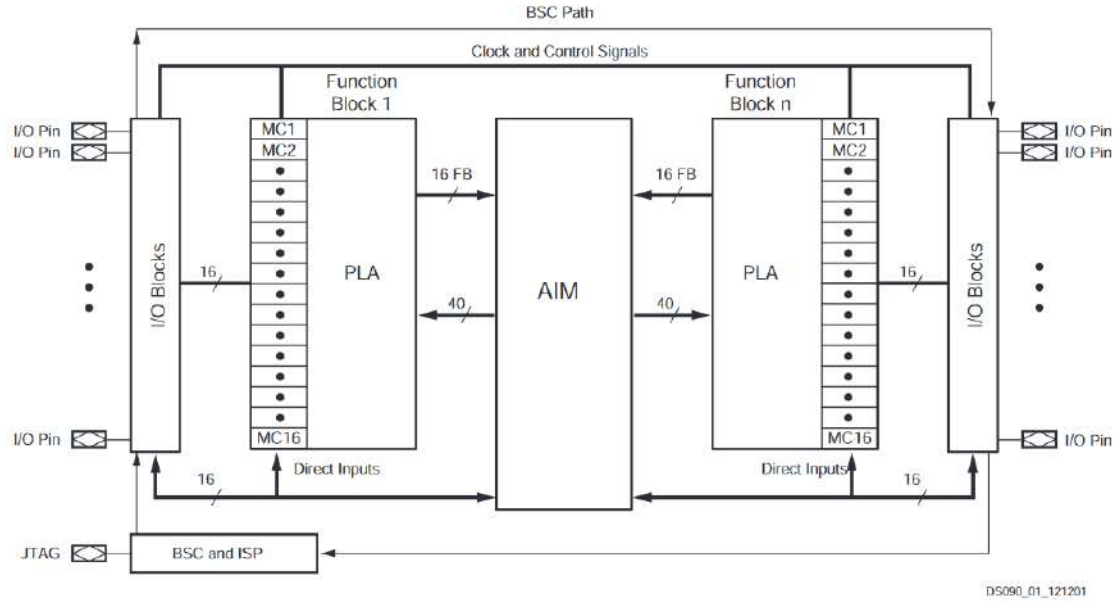


Diagrama en bloques de arquitectura de CPLDs CoolRunner-II.

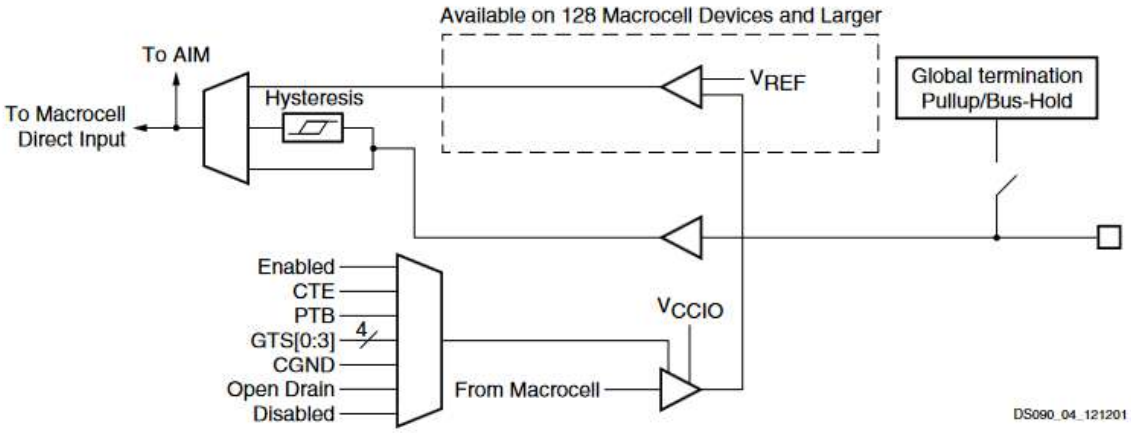


Diagrama en bloques de I/O en CPLDs utilizados.

En este caso, se utilizó VHDL (VHSIC Hardware Description Language) para realizar la descripción de hardware deseada, y el desarrollo se realizó en la plataforma ISE, también propietaria de Xilinx, la cual permite realizar el HDL (descripción de hardware), como así también generar un esquema RTL (Register Transfer Level, una representación de alto nivel del circuito descripto) y observar el hardware sintetizado, asignar el pinout en el IC, observar el real-estate disponible y ocupado por el diseño, y también cargar el HDL al CPLD correspondiente vía JTAG entre otras funcionalidades.

Además de tener la ventaja de poder implementar diferentes funciones de forma concurrente en el mismo dispositivo en vez de utilizar lógica discreta. Dicho HDL describe el funcionamiento y la lógica del circuito que se desea sintetizar para implementar las funciones requeridas, lo cual también permite testear distintas configuraciones lógicas sin modificar el diseño desde el punto de vista del PCB.

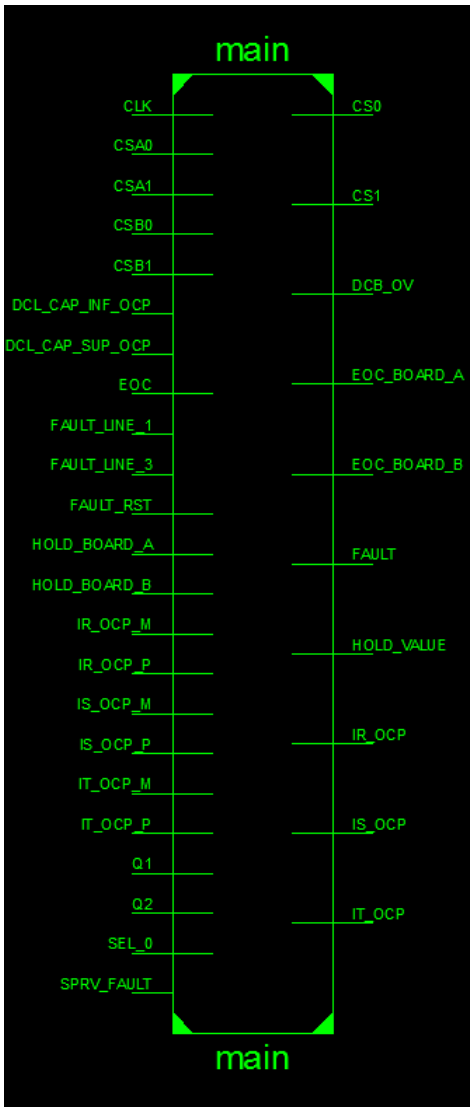
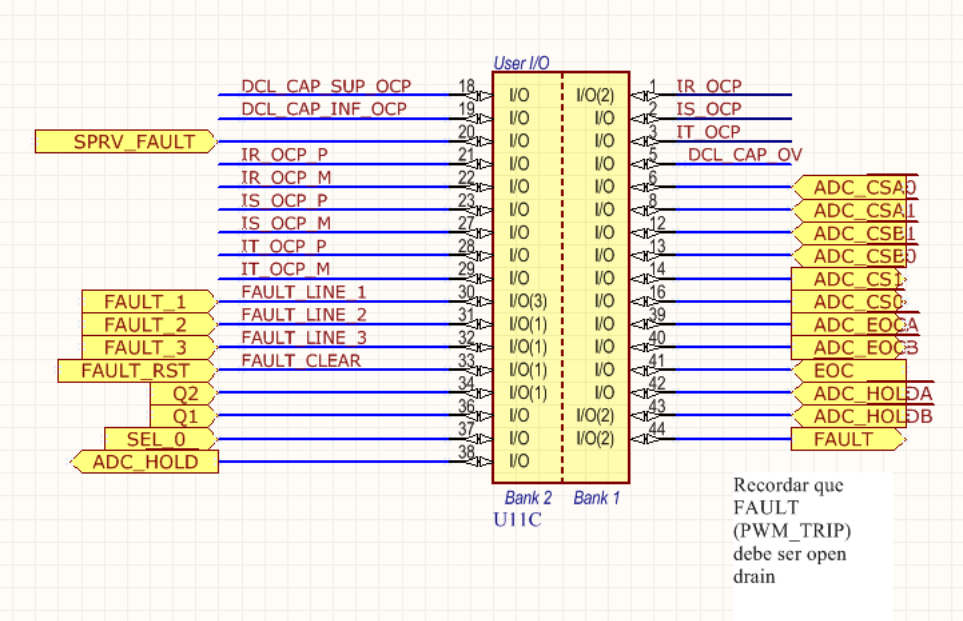


Diagrama de bloque principal RTL para CPLD de **AnalogFrontEnd**.



Circuito esquemático con señales que pasan por CPLD en **AnalogFrontEnd**.

En este caso, el dispositivo se utilizó como lógica de interfaz entre el PCB **DigitalControlBoard** que controla el funcionamiento del sistema y el hardware dedicado para la adquisición en **AnalogFrontEnd**, de modo que las señales de control generadas tanto en **DigitalControlBoard** como en los adquisidores son monitoreadas y retransmitidas a los componentes correspondientes.

Otra de las funciones de este dispositivo, es la de identificar de forma correcta las señales a los dos posibles PCBs de adquisición cuando corresponda (en caso de que se cuente con una expansión de canales de adquisición con un segundo PCB **AnalogFrontEnd**). Esta identificación dada por la señal $BOARD\_ID$ es controlada por un DIP switch que posee cada PCB **AnalogFrontEnd**.

Este dispositivo tiene la función adicional de ser utilizado como interfaz y supervisión de las protecciones de hardware, puesto que monitorea una serie de comparadores antes mencionados, los cuales en caso de cambiar de estado a causa de una falla, causan que se genere una acción de $FAULT$, la que se transmitirá por la señal dedicada $HARD\_FAULT$ hacia **DigitalControlBoard** (en particular, esta es recibida por el módulo de hardware integrado en el microcontrolador que se utiliza para controlar los PWMs y pone en estado inactivo todos los canales, lo cual abre todas las llave de potencia del sistema). Dicha acción de falla debe ser reseteada por la señal $FAULT\_CLEAR$ que ingresa a partir del accionar de un botón de reset, monitoreado y procesado por un anti rebote implementado en el mismo CPLD.

Para implementar, este se realizó mediante un re-work y se le ingreso al CPLD la señal de clock del FlexBus a modo de disponer de una referencia desde el punto de vista sincrónico y se agregó al HDL original del CPLD el código de debounce.

En caso de una falla detectada por el monitoreo de los comparadores, también se encenderá una indicación visual que permite identificar cuál de las protecciones genero la falla en base a una situación anómala, la cual puede ser causada por una sobre corriente inyectada en una de las fases o una sobretensión en el bus de continua del convertidor.

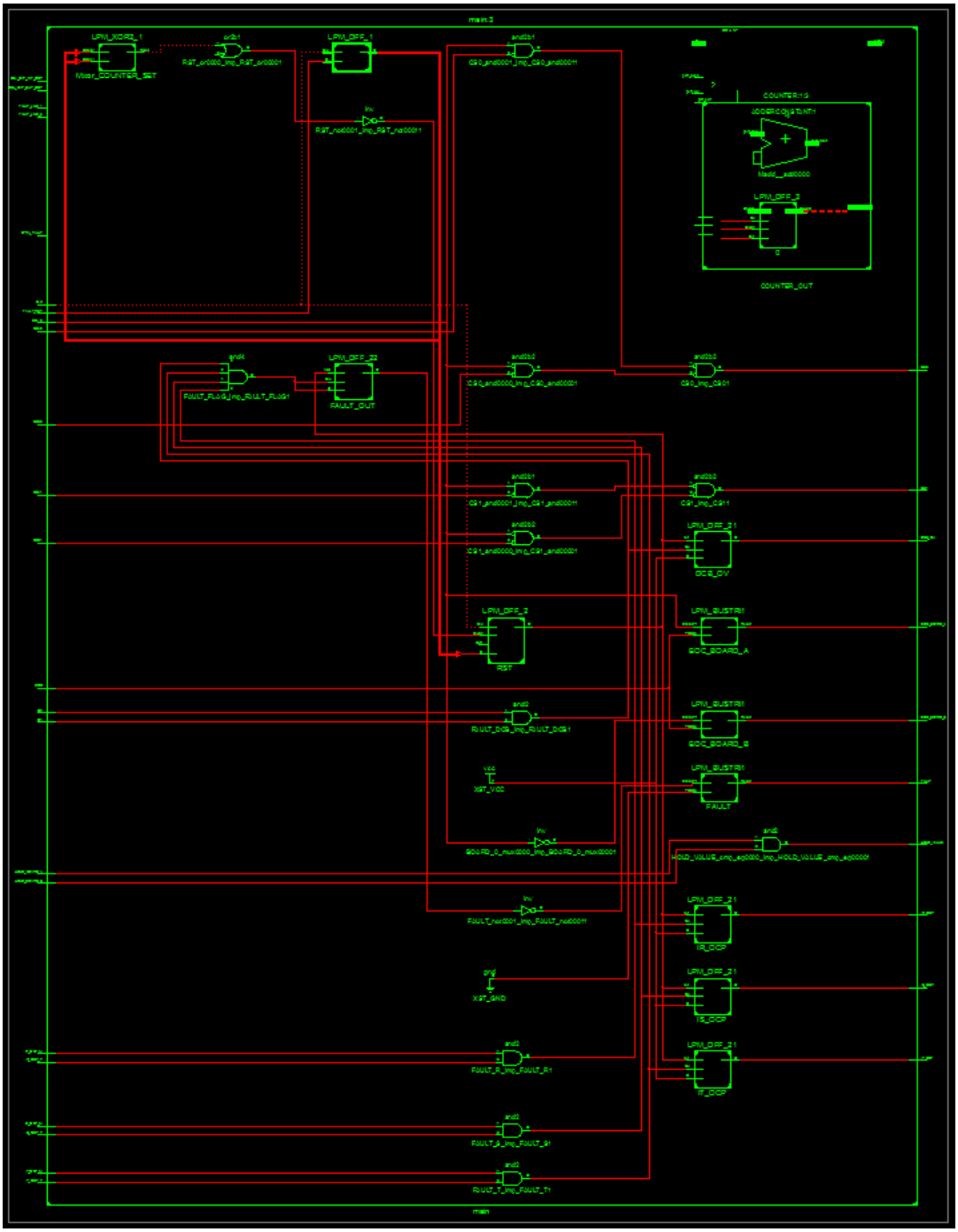
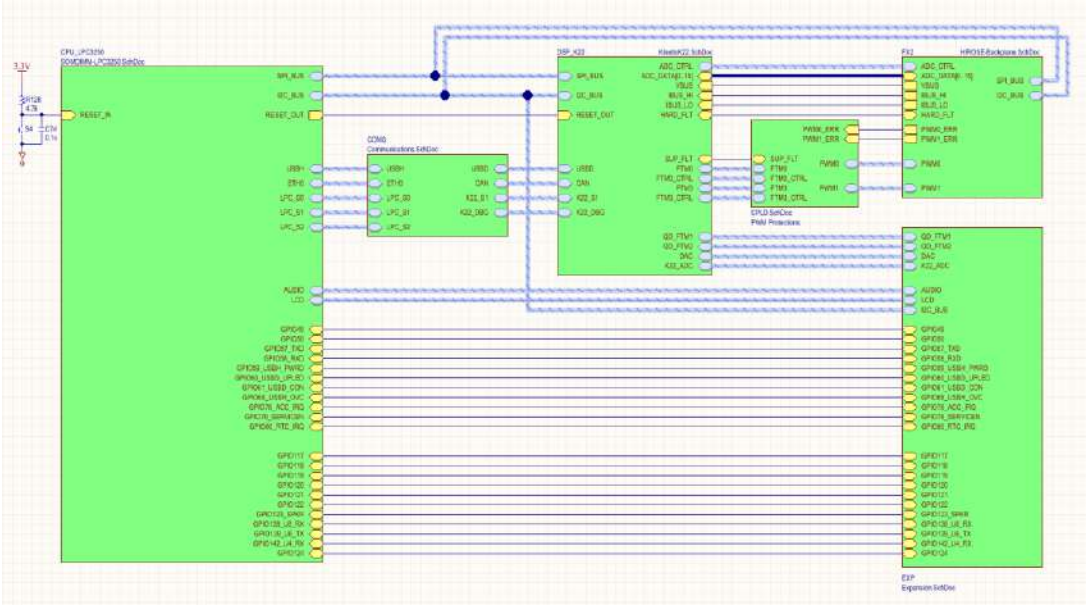


Diagrama RTL completo de CPLD en **AnalogFrontEnd** con todas las funciones implementadas, generado por software ISE.

DigitalControlBoard:

El PCB **DigitalControlBoard** es el que dispone del hardware que se utiliza para lograr las funciones principales y algoritmos de control, la generación de señales de sincronización, de canales de PWM, sincronismo respecto a la red eléctrica, control de adquisición, el hardware necesario para expandir a futuro las funcionalidades disponibles y la comunicación e interfaces al exterior, entre otras cosas.

Este PCB posee como componente principal el microcontrolador ARM Cortex-M4, el cual contiene módulos de hardware integrados en su arquitectura interna, los cuales son de gran utilidad para concretar las funciones antes mencionadas, además de su gran poder de procesamiento y la gran cantidad de documentación, librerías y ejemplos útiles para el desarrollo de proyectos tales como este.



Esquema general de PCB **DigitalControlBoard**.

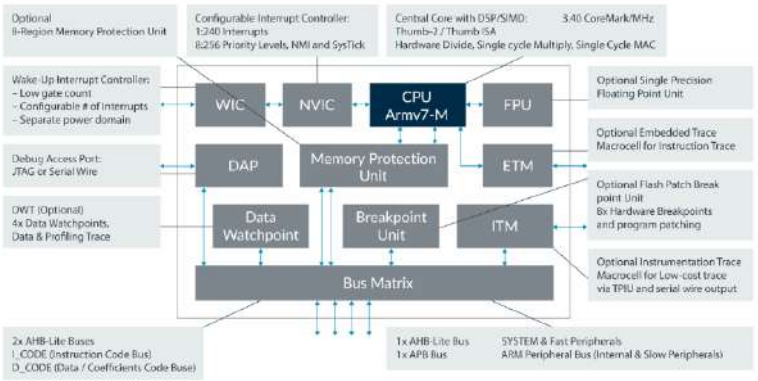
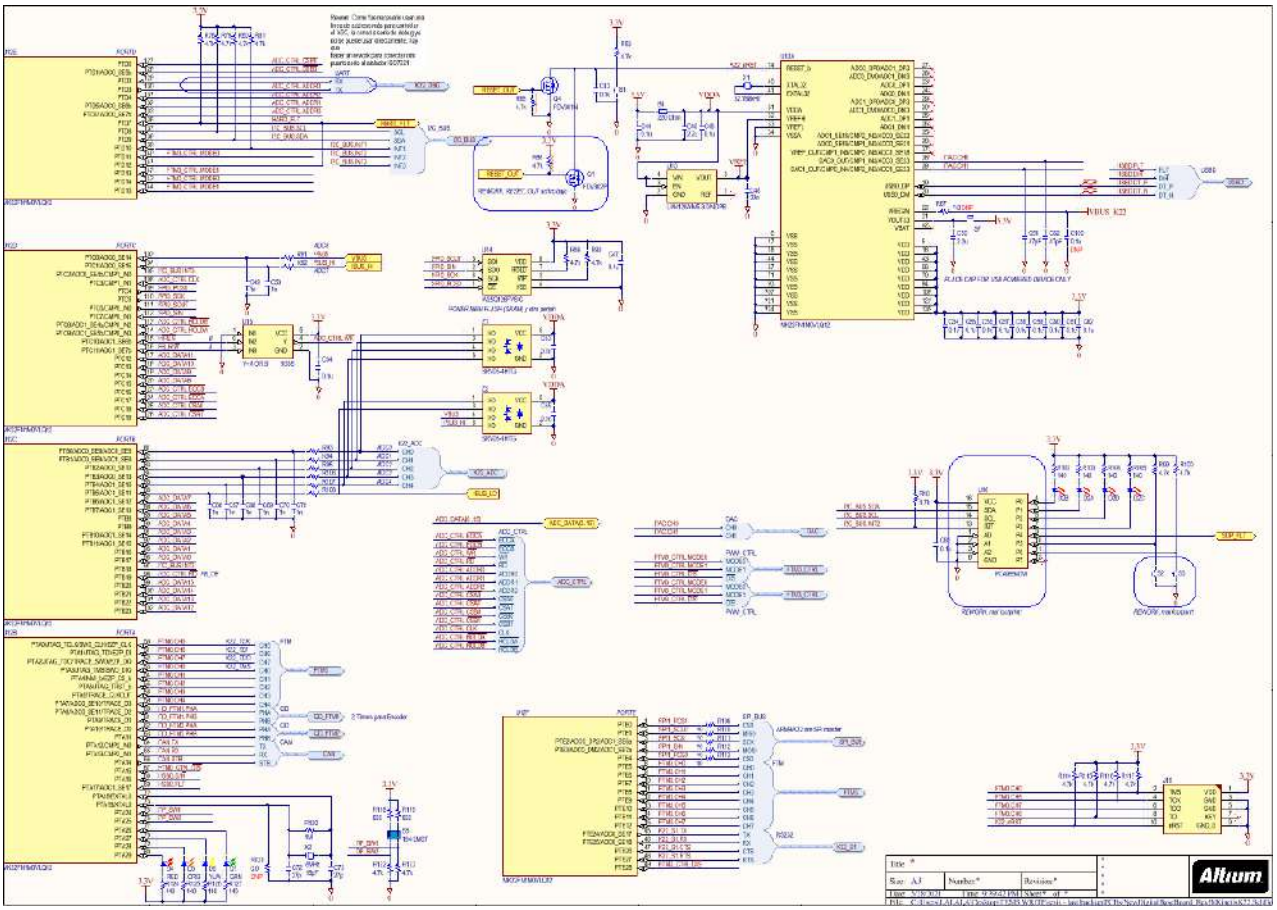
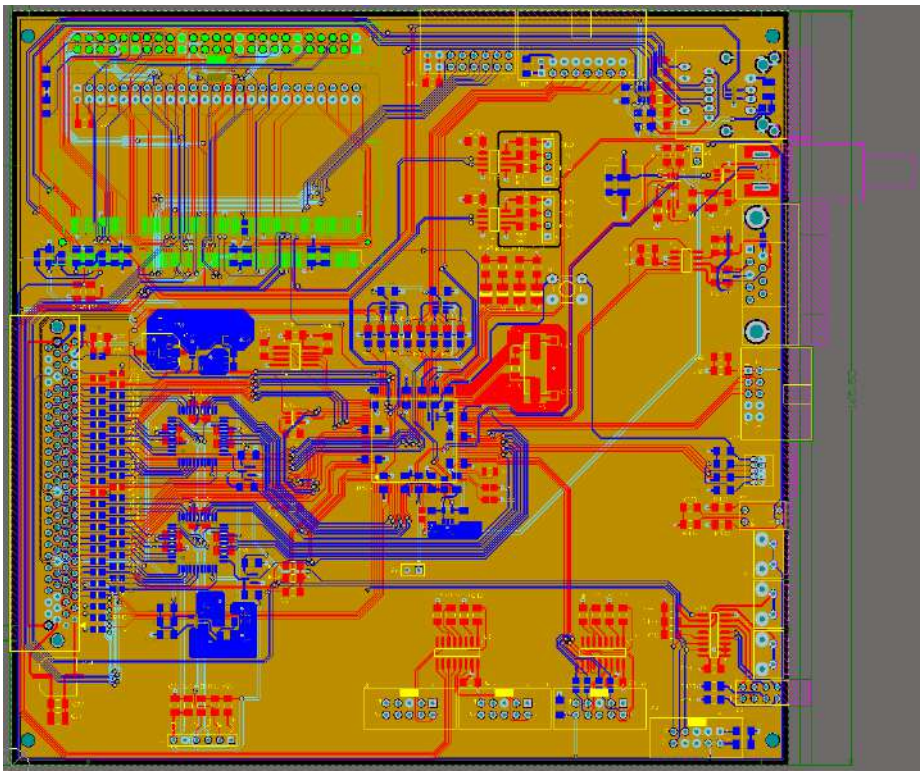


Diagrama en bloques de arquitectura ARM Cortex-M4.

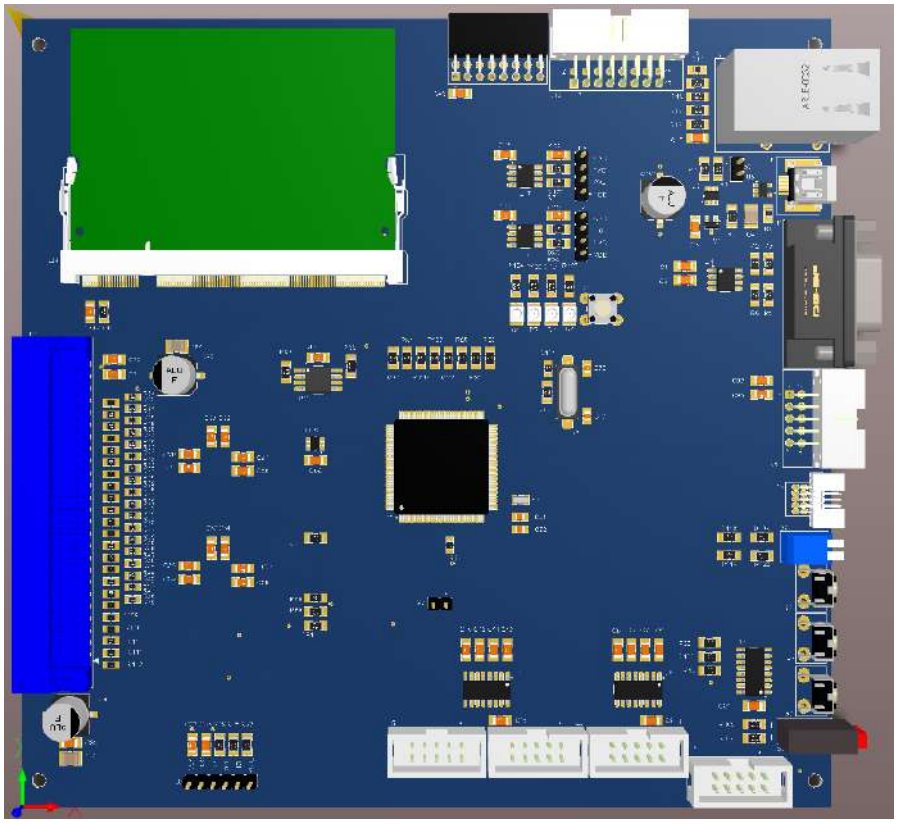
El microcontrolador de 32 bits Cortex-M4 K22P144M120SF5RM es el circuito integrado que realiza las funciones principales del sistema. Este funciona a partir de un clock de $120\ MHz$, posee una amplia capacidad de procesamiento incluyendo FPU (unidad de punto flotante) y capacidad de DSP (procesamiento digital de señales), una cantidad de pines I/O (entradas-salidas) y diferentes módulos de hardware integrados que realizan funciones específicas muy útiles para implementar la plataforma de control, y así el convertidor de potencia, tales como un bus de lectura paralela, un módulo diseñado específicamente para el control y generación de señales PWM, un módulo de Delay programable que genera interrupciones, ADCs y DACs integrados, etc.



Circuito esquemático de las principales funciones y señales en **DigitalControlBoard**.



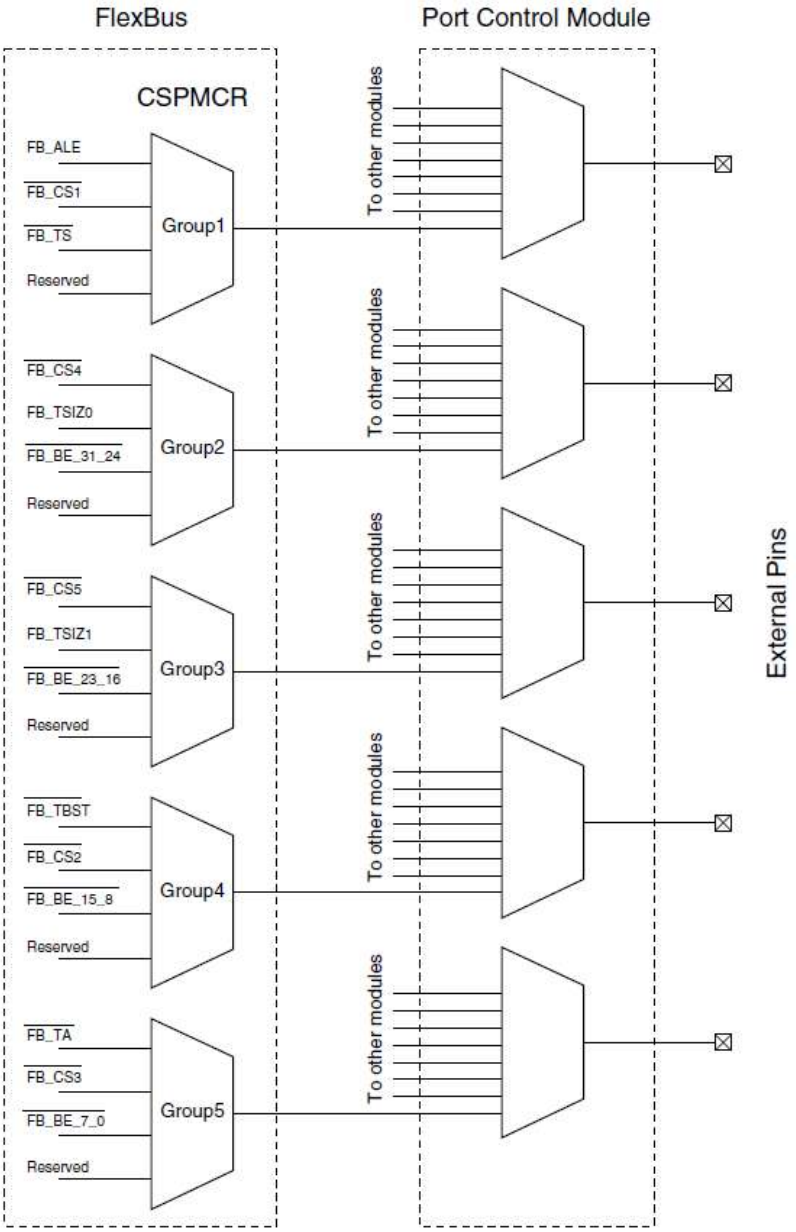
Layout de PCB **DigitalControlBoard.**



Render 3D en Altium Designer de PCB **DigitalControlBoard**.

FlexBus (External Bus Interface):

Uno de los módulos utilizados en la realización del sistema es el FlexBus. Dicho modulo es una interfaz de bus externa multifunción, la cual provee de conexión a periféricos externos con un bus paralelo, capaz de ser conectada a otros dispositivos esclavo sin circuitería adicional. Este posee un tamaño de puerto configurable de 8, 16, o 32 bits multiplexado o no multiplexado para direcciones y datos con 6 señales de chip-select independientes. Permite transferencias de 8 a 32 bits en formato burst y tiempos de direccionamiento programables.



Señales de control y multiplexado del módulo FlexBus.

Las señales involucradas en este módulo son las siguientes:

- $FB\_A31 - FB\_A0$: Bus de direccionamiento. Cuando el FlexBus se utiliza en modo no-multiplexado, este funciona como bus de direccionamiento, en caso contrario, este bus queda sin utilizar.
- $FB\_D31 - FB\_D0$: Bus de datos. En modo multiplexado este será el bus de datos y direcciones, caso contrario solo funcionara como bus de datos. La cantidad de bytes que llevan los datos estará determinada por el tamaño del bus. En modo multiplexado se utilizaran los bits libres para transmitir la direcciones o se hará en dos etapas, una primer etapa de direccionamiento y luego una etapa de datos. En este caso se utilizara en 16 bits de forma no multiplexada.
- $\overline{FB\_CS5} - \overline{FB\_CS0}$: Chip-Selects. Indican cuál de los periféricos externos se selecciona. Cuando una dirección de transferencia está comprendida dentro del espacio de direcciones de periférico, un CS en particular se habilita. Esto se define en CSAR [BA] y CSMR [BAM].
- $\overline{FB\_BE\_31 \ldots 0}$: Byte-Enable. Indica que los datos deben ser guardados o transmitidos a un byte específico del bus de datos. CSCR [BEM] determina si estas señales son habilitadas en lecturas y escrituras o solo en escritura.
- $\overline{FB\_OE}$: Output-Enable. Enviado hacia el periférico para habilitar una operación de lectura. Esta señal es habilitada durante los accesos de lectura solo cuando el CS coincide con la dirección decodificada actualmente.
- $FB\_R/\overline{W}$: Read/Write. Indica si el bus está operando en una operación de lectura o escritura.
- $\overline{FB\_TS}$: Transfer Start. Indica que el IC comenzó una transacción de bus y que la dirección y los atributos son válidos.
- FB_ALE: Address Latch Enable. Indica cuando la dirección está siendo transmitida en el bus de direcciones.
- $FB\_TSIZ1 - FB\_TSIZ0$: Tamaño de transferencia. Indica junto a la señal $\overline{FB\_TBST}$ el tamaño de la transferencia de datos de la operación del bus actual. La interfaz soporta transferencias de 8 a 32 bits y permite accesos a puertos de datos de 8 a 32 bits.
- $\overline{FB\_TBST}$: Transfer Burst. Indica que una transferencia de tipo burst está en progreso de transmisión por el IC. Una transferencia de tipo burst puede ser de 2 a 16 datos dependiendo de FB_TSIZ1 – FB_TSIZ0 y el tamaño del puerto.
- $\overline{FB\_TA}$: Transfer acknowledge. Indica que la transferencia de datos externa ha sido completada, cuando $\overline{FB\_TA}$ es habilitada durante una transferencia de lectura, el FlexBus latchea los datos y termina la transferencia. Cuando $\overline{FB\_TA}$ es habilitada durante una transferencia de escritura, la transferencia es finalizada.
- FB_CLK: Clock Output. Es el clock generado por el FlexBus para los periféricos utilizados. Su frecuencia depende del clock del sistema y la configuración del bus.

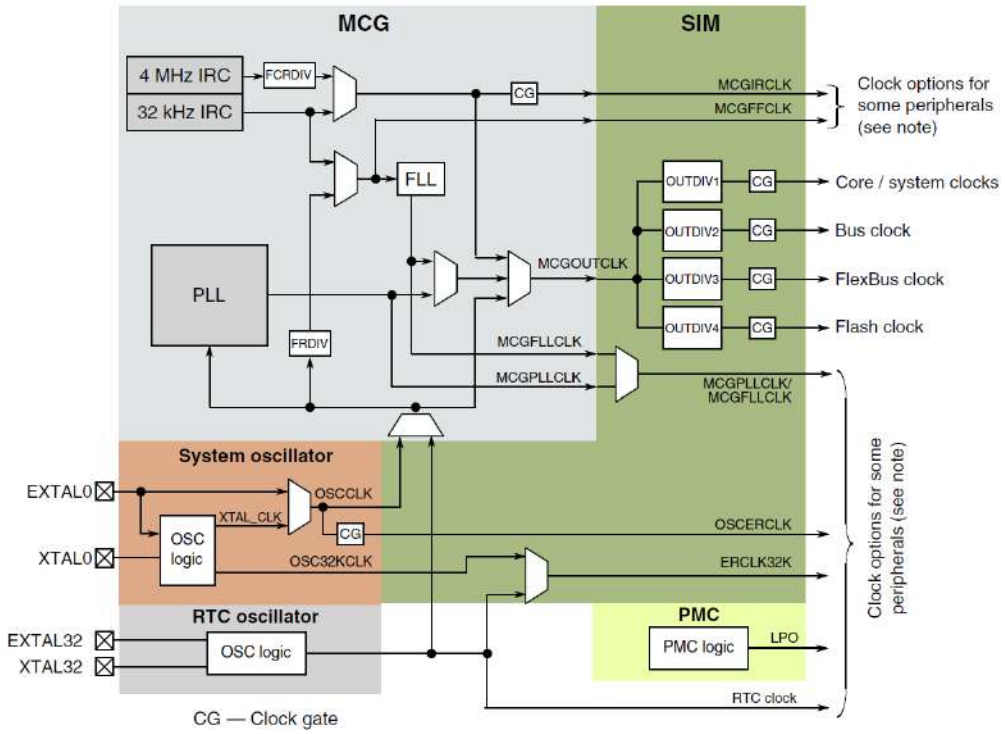


Diagrama de clock del IC ARM Cortex-M4 (incluyendo clock del FlexBus).

Como se mencionó anteriormente, se utilizó un clock de $7.5\ MHz$ mayor al soportado según el datasheet del ADS8365 (máximo $5\ MHz$), puesto que el clock del FlexBus depende del clock del sistema de $120\ MHz$ el cual se modifica con un divisor como se muestra en el gráfico, pero dicho divisor, al testearse experimentalmente, demostró no funcionar correctamente para lograr la frecuencia de $5\ MHz$ para la máxima tasa de muestreo posible con el IC de adquisición del que se dispuso en el diseño. Por otro lado, se podría haber bajado el clock del sistema a $80\ MHz$ para lograr que con el divisor se entregaran $5\ MHz$ en el clock del FlexBus, pero esto generaría una pérdida de poder de procesamiento en el resto del sistema, por lo que se optó por testear la adquisición con el clock mencionado anteriormente de $7.5\ MHz$.

Esta condición se testeo exhaustivamente tanto en banco de prueba como utilizando al sistema de manera experimental, y se logró que este funcione de manera correcta, por lo que se pudo solucionar dicha problemática sin la consiguiente pérdida de performance.

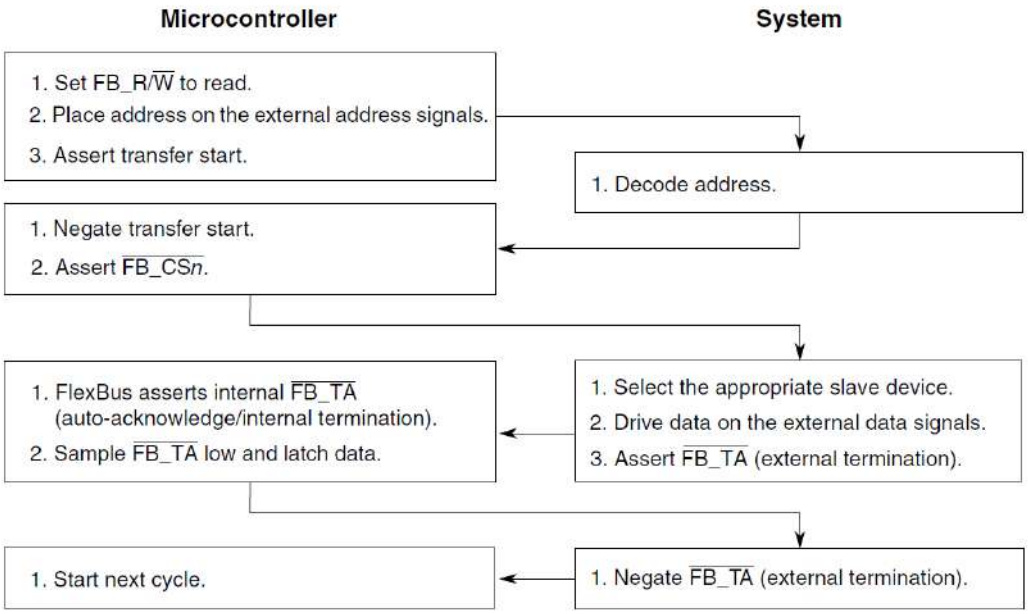
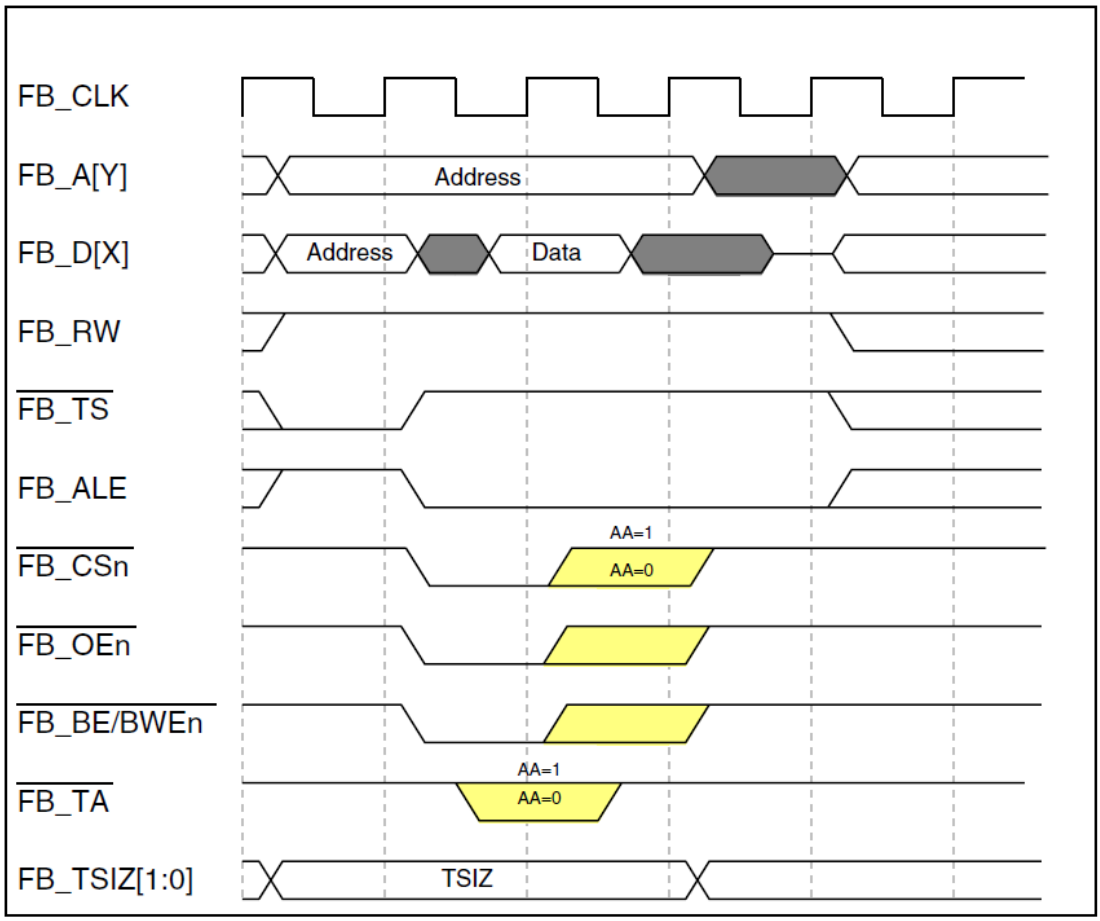
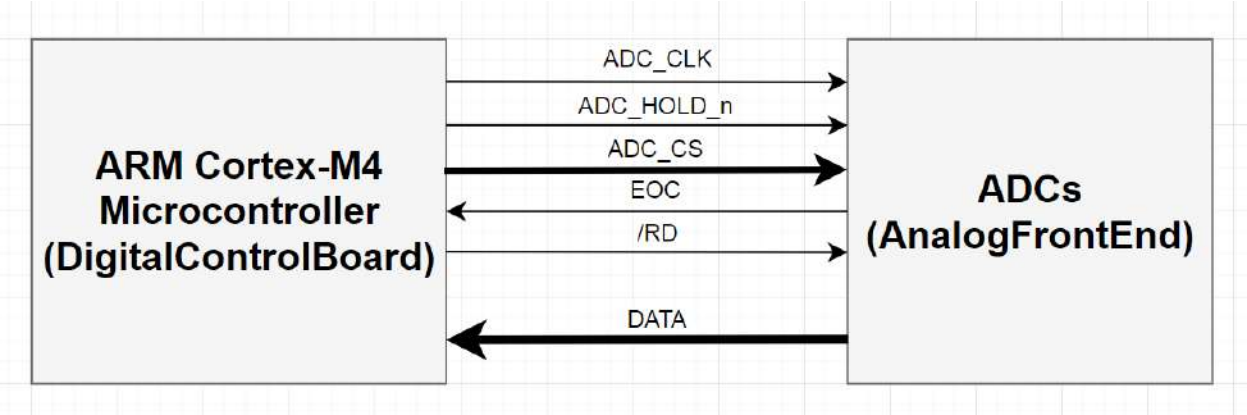


Diagrama de flujo para ciclo de lectura de FlexBus.



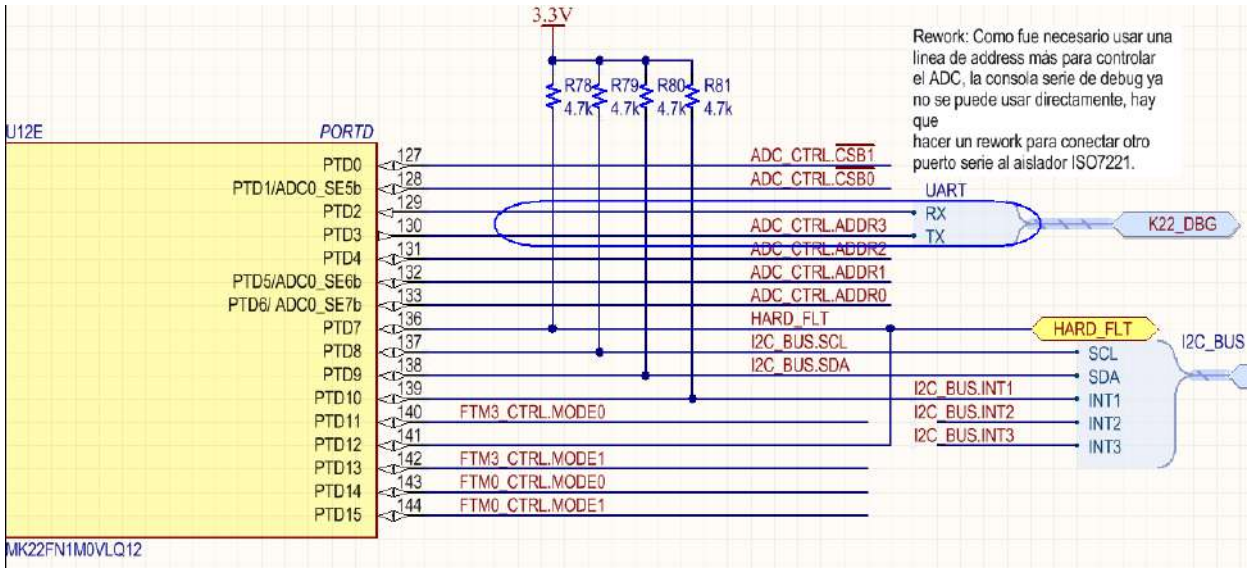
Ciclo de lectura básico del FlexBus con señales de control especificadas.

El FlexBus se utilizó como interfaz de bus paralelo para hacer las operaciones de lectura de datos de todos los canales de adquisición implementados en **AnalogFrontEnd** con los ADS8365 de manera eficiente y robusta.



Esquema de conexión entre microcontrolador y adquisidores.

Cuando se testeo experimentalmente el FlexBus, se encontró que había un conflicto en el diseño original entre este y la UART (universal asynchronous receiver-transmitter, comunicación) utilizada. Específicamente entre las señales de direccionamiento del FlexBus y las señales de Tx y Rx de la UART, originado por un error en cuanto a los pines utilizados para ambas funcionalidades en el microcontrolador. Para remediar este problema se optó por realizar una serie de re-works en el PCB y un cambio en la configuración del FlexBus. Desde el lado del FlexBus, se optó por una lectura de modo secuencial y por otro lado, se re-ruteo de modo de utilizar la UART4 hacia la UART2.



Extracto de circuito esquemático donde se ve el problema.

FlexTimer:

Otro módulo de hardware del que dispone el microcontrolador el cual vale la pena destacar por la utilización en este proyecto es el módulo FlexTimer (FTM). Dicho modulo es un timer multicanal que soporta captura de entrada, comparación de salida y generación de señales PWM para el control de motores y aplicaciones de potencia. Este posee además, hardware para la inserción de tiempos muertos, entradas de control de fallas, funcionalidades de disparo, inicialización, control de polaridad, forzado de salida, y enmascarado, entre otras.

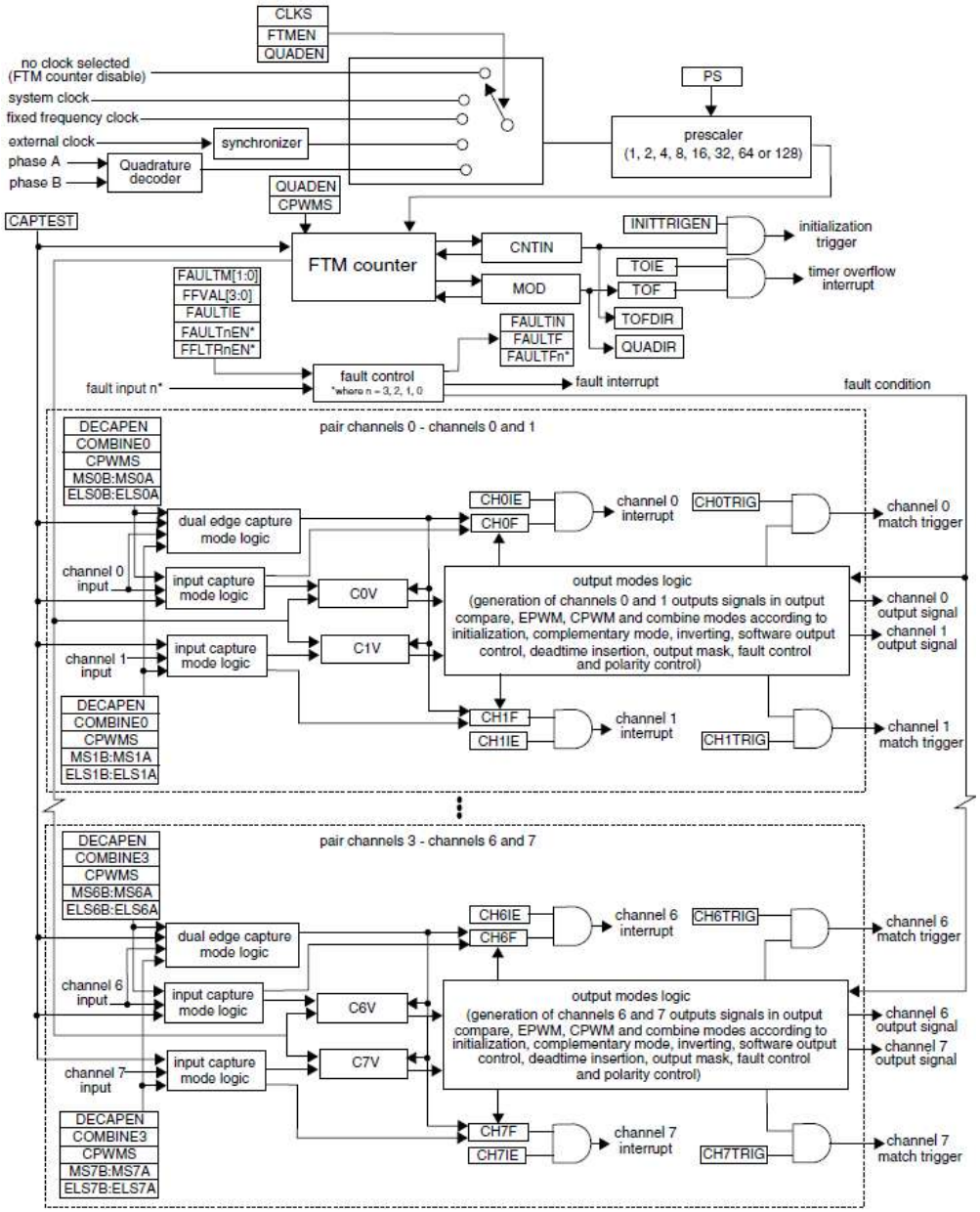


Diagrama en bloques del FTM (FlexTimer).

Las señales utilizadas por el módulo FTM son las siguientes:

- $EXTCLK$: External Clock. Entrada de señal utilizada como clock para el contador de 16 bits del FTM. Seleccionable en el campo $CLKS\ [1:0]$ en el registro SC. Esta no debe exceder ¼ de la frecuencia del clock del sistema. La selección del prescaler del contador FTM y su configuración son utilizados si se selecciona un clock externo.

- CHn: Canal (n) del FTM. Donde n puede ser $0 - 7$. Cada uno de estos canales puede ser configurado para operar como entrada o salida. La dirección asociada a cada canal, se selecciona de acuerdo al modo asignado para este canal.

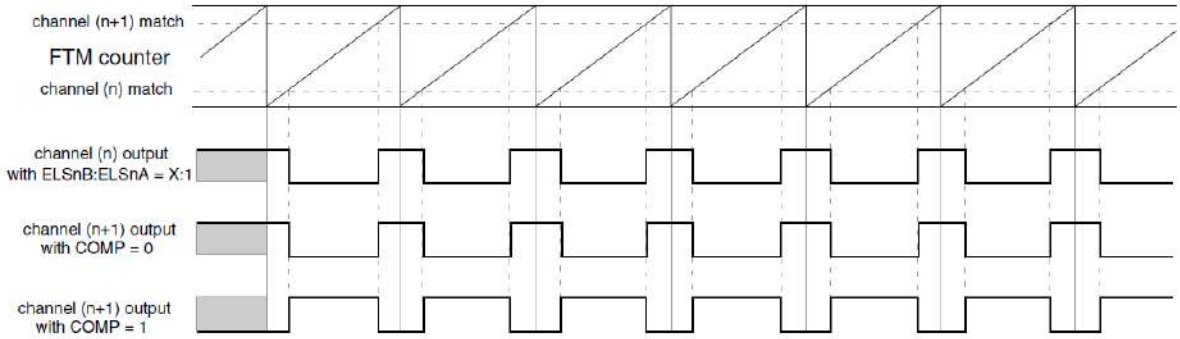
- $FAULTj$: Entrada (j) de falla. Donde j puede ser $0 - 3$. Las señales de entrada de falla pueden ser utilizadas para controlar los estados de salida de los canales CHn. Si se detecta una falla, la señal $FAULTj$ es habilitada y la salida del canal es forzada a un estado seguro. El comportamiento de la lógica de fallas está definido por los campos $FAULTm\ [1:0]$ en el registro MODE y los bits FAULTEN están definidos para cada par de canales. Puesto que hay varias entradas $FAULTj$, cada una de estas es activada por el bit $FAULTjEN$ en el registro FLTCTRL.

- PHA: Fase A del decodificador de cuadratura, utilizada si se selecciona modo Quadrature Decoder.

- PHB: Fase B del decodificador de cuadratura, utilizada si se selecciona modo Quadrature Decoder.

Este módulo fue utilizado para la generación de las señales de PWM que conmutan las llaves del convertidor, la inserción de los tiempos muertos acorde a las llaves utilizadas, y el bloqueo y puesta en estado seguro de las llaves en caso de falla por sobre-corriente o sobre-tensión.

Las señales de PWM generadas son de tipo complementarias, combinadas, y con la llave en estado activo para pulsos activo-bajo. Es decir, un nivel bajo en el canal (n) y nivel alto en el canal $(n+1)$. En modo complementario, el canal $(n+1)$ es el inverso de la salida de canal (n).

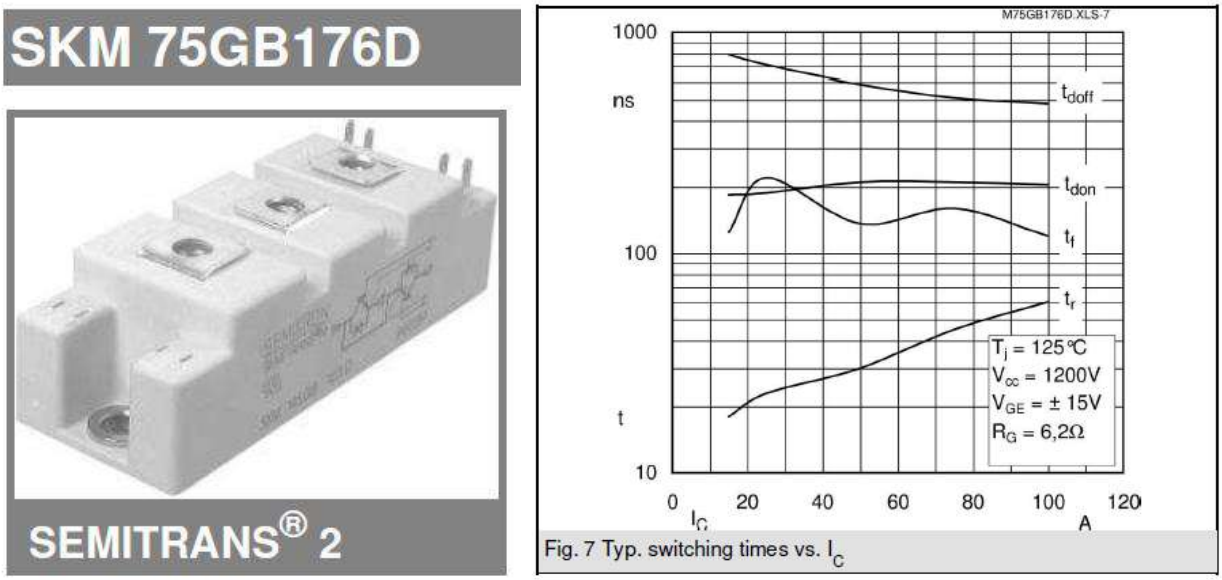


Salida de canales (n) y $(n+1)$ para la configuración utilizada en modo complementario.

Este módulo permite la inserción de tiempos muertos. Estos son tiempos de seguridad para evitar una condición donde ambas llaves de potencia estén en estado activo al mismo tiempo transitoriamente debido al periodo que le toma a la llave pasar de estado activo a inactivo. Dicho tiempo es generado por fenómenos constructivos (generados a partir del tiempo que tardan los portadores minoritarios para recombinarse en el material utilizado para la fabricación de la llave). Si dichos tiempos no fueran implementados en las señales de PWM, se generaría una condición de corto-circuito en el convertidor y se dañarían los componentes de la etapa de potencia.

Las llaves utilizadas en este sistema son las que ya estaban implementadas en el panel de testeo del laboratorio, con drivers asociados a IGBTs SKM75GB176D que poseen ciertas características tales como la tensión de bloqueo y corriente de conducción antes mencionadas (sobredimensionadas por criterio de diseño).

Otra de las características fundamentales de las llaves como fue mencionado anteriormente es el tiempo de conmutación, y en base a estos tiempos de conmutación es que se decide por los tiempos muertos a insertar en los PWMs.



IGBTs utilizados y su característica de conmutación.

Para evitar esto, se dimensionan los tiempos muertos de modo tal que estos sean mayores al tiempo de conmutación de las llaves, pero sean de una duración lo más corta posible para lograr una forma de onda con la menor distorsión posible. En este caso, la inserción de retardo de tiempo muerto en el FTM asegura que dos canales complementarios (n) y $(n+1)$ no puedan estar en estado activo al mismo tiempo si estos se dimensionan de manera correcta.

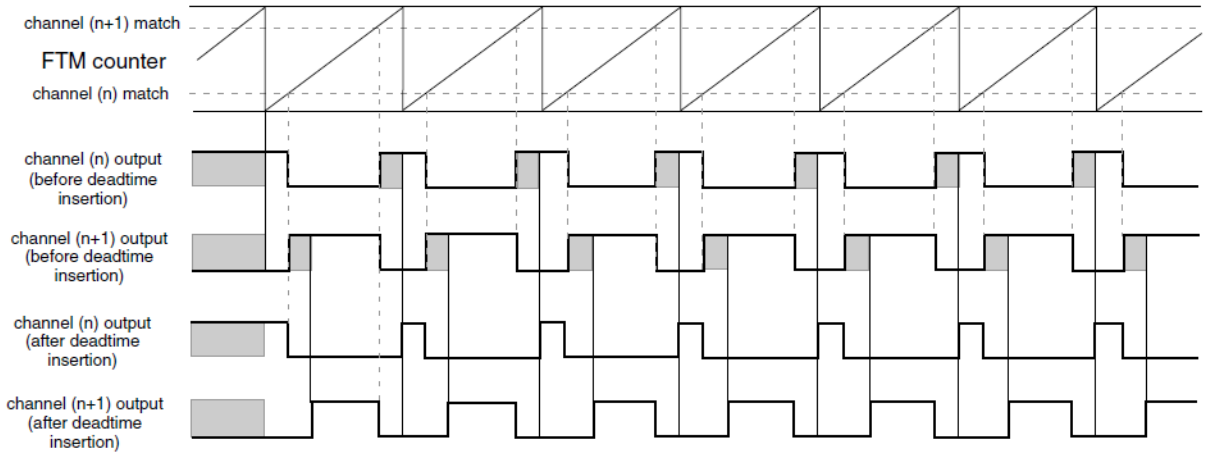


Diagrama de inserción de tiempos muertos en canales (n) y $(n+1)$ valido para modo complementario y la configuración utilizada.

Este módulo cuenta con la posibilidad de controlar la polaridad de las señales de PWM generadas (en este caso, el diseño es con polaridad activo bajo por el diseño de los drivers), y de seleccionar un estado inicial para las llaves. Para garantizar un inicio seguro, ambas llaves se inicializan en estado alto (es decir, en su estado inactivo).

Otra de las características útiles de este módulo integrado en el microcontrolador, es la de control de fallas, que monitorea una serie de señales de entrada, y en caso de que se detecte una condición de falla, las salidas de PWM son forzadas hacia valores seguros, es decir, que el canal (n) toma el valor de POL(n) y el canal $(n+1)$ toma el valor de POL$(n+1)$, además de generarse una interrupción de falla en el módulo, el cual permite tanto un clearing automático como manual de esta condición.

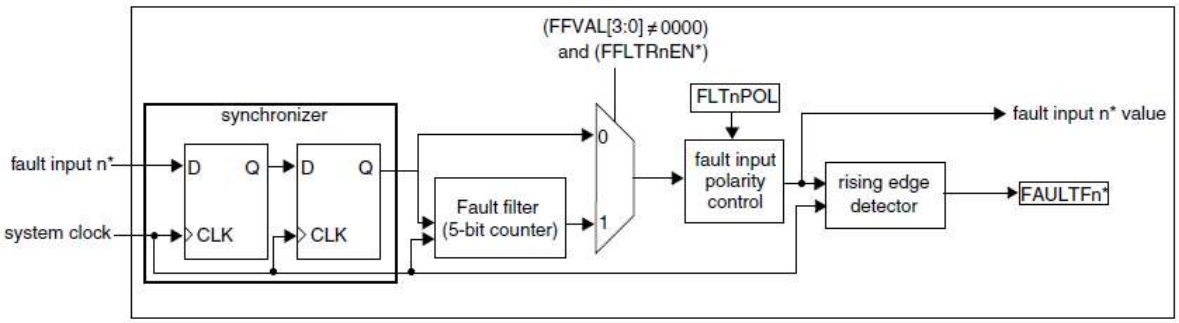


Diagrama de bloques de Fault input n control.

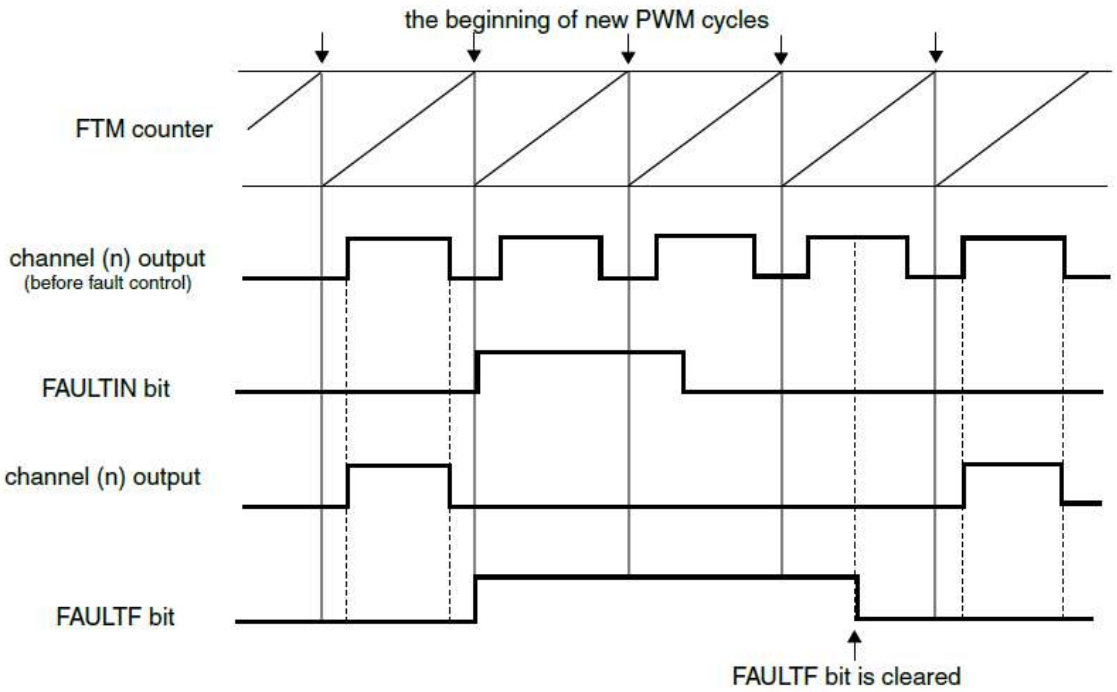


Diagrama de Fault clearing manual utilizado en este proyecto.

Estas y otras características del módulo son las que definen el valor de las salidas PWM. Dichas configuraciones tienen prioridad entre sí, como puede observarse en el siguiente gráfico:

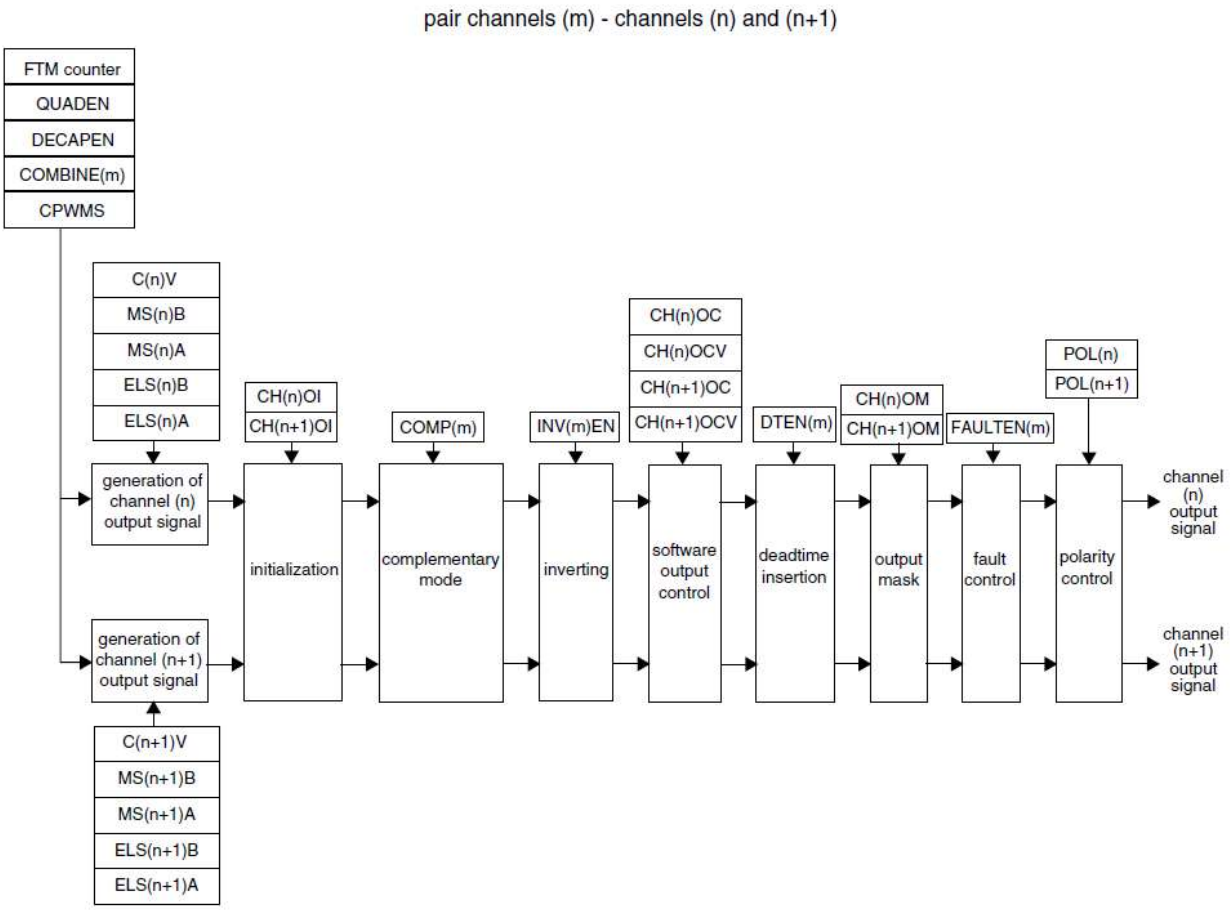


Diagrama de orden de prioridad para las configuraciones posibles en el FTM.

Dicho modulo es el que dicta la temporización del control y adquisición, ocurriendo la interrupción de este cada $100\ \mu s$ (lo cual equivale a una frecuencia de conmutación de $10\ KHz$). En cuanto a la adquisición, esta es temporizada por dos módulos diferentes, siendo este (FTM) el que marca el comienzo de un ciclo de PWMs y adquisición.

Programmable Delay Block:

Como se mencionó anteriormente, la adquisición de las magnitudes de interés ocurre a $30\ KHz$, es decir, al triple de frecuencia de conmutación del convertidor. Esto se debe a que se desea hacer un sobre-muestreo de las señales de interés, para luego procesarlas posteriormente y obtener una medición de mejores características. Para lograr esto de forma sincrónica, se utilizó otro módulo de hardware llamado PDB (programmable Delay block), el cual permite la programación de un retardo para utilizar como interrupción interna o externa o como disparo para hardware (en este caso, para los adquisidores). Esto permite que la temporización entre las conversiones de ADC sea precisas en el tiempo. Dicho modulo fue utilizado para disparar dos ciclos de adquisición adicionales además del primer ciclo en la interrupción del FTM, y así lograr la implementación del over-sampling (sobre-muestreo) para el filtrado digital.

Dicho modulo posee diferentes características, entre ellas:

- Modo one-shot o modo continuo
- Salida de trigger para ADCs y hasta 8 salidas pre-trigger para seleccionar como disparo de ADC por canal de PDB.
- Posee registro de retardo de 16 bits.
- Triggers externos opcionales.

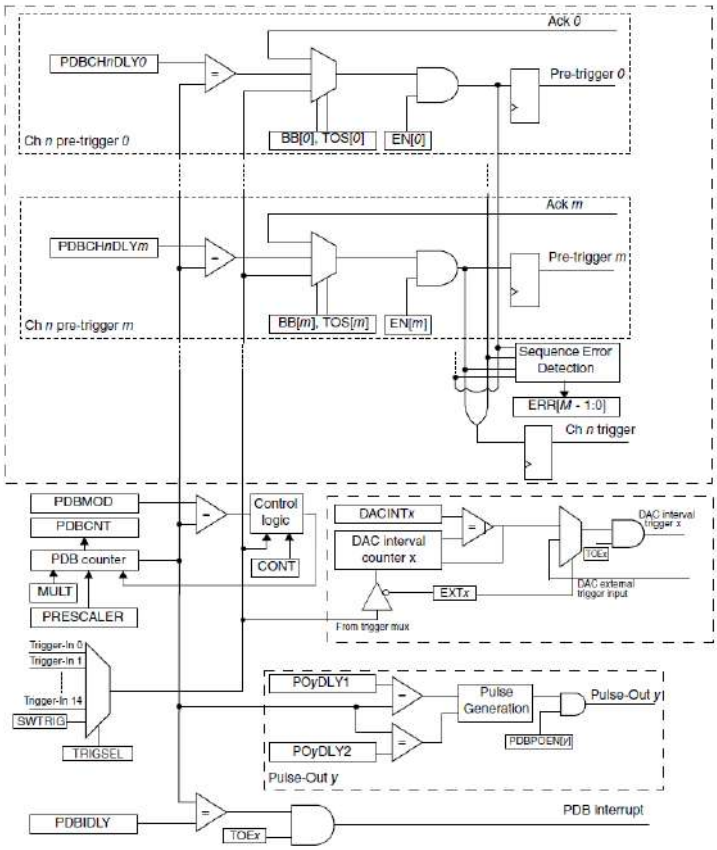


Diagrama en bloques del módulo PDB.

El funcionamiento de este módulo se da en base a un contador interno el cual se compara contra diferentes valores, y genera un trigger de acuerdo a su configuración (en este caso, el trigger maestro será el del FTM3), y a través de este módulo se generaran dos señales más que dispararan nuevos ciclos de adquisición y lectura de magnitudes.

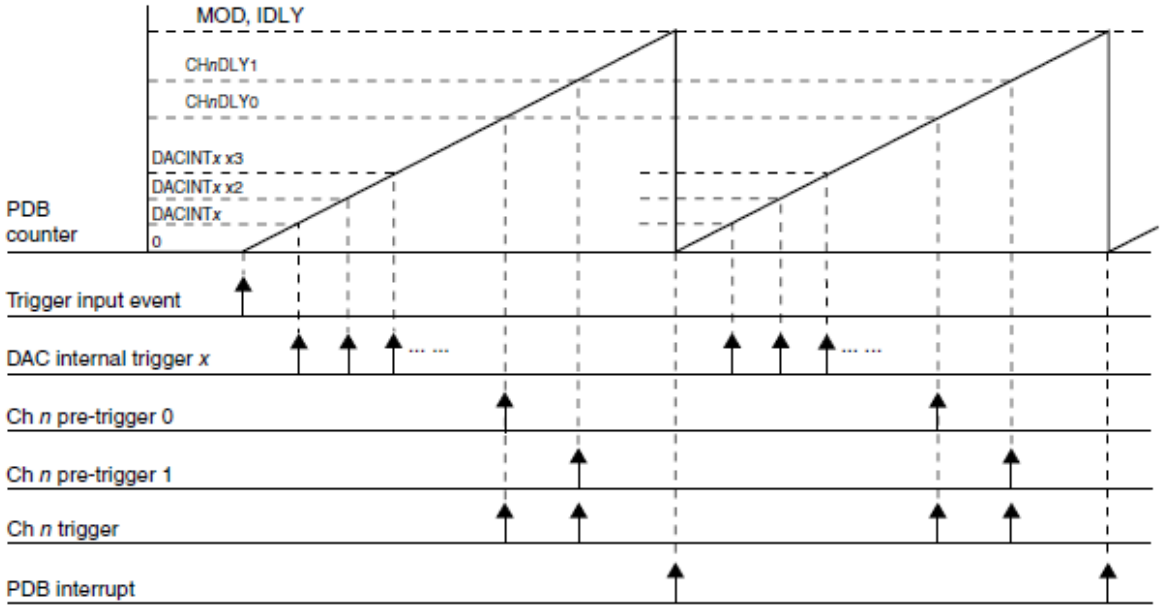


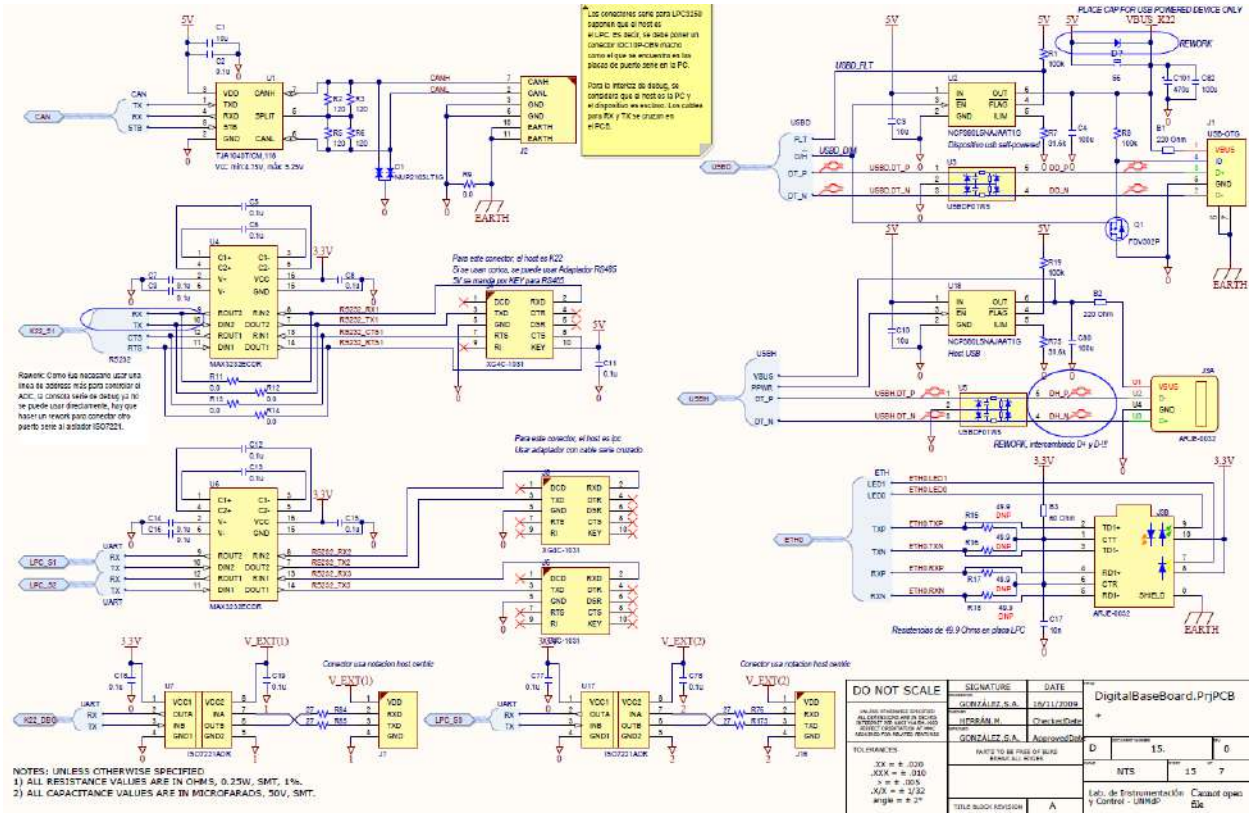
Figure 37-57. PDB ADC triggers and DAC interval triggers use case

Ejemplo de disparos generados por la cuenta del módulo PDB.

Dichas interrupciones fueron validadas experimentalmente en banco de prueba para comprobar que efectivamente disparan nuevas adquisiciones de magnitudes de forma correcta en los intervalos de tiempo esperados.

Comunicación y expansión:

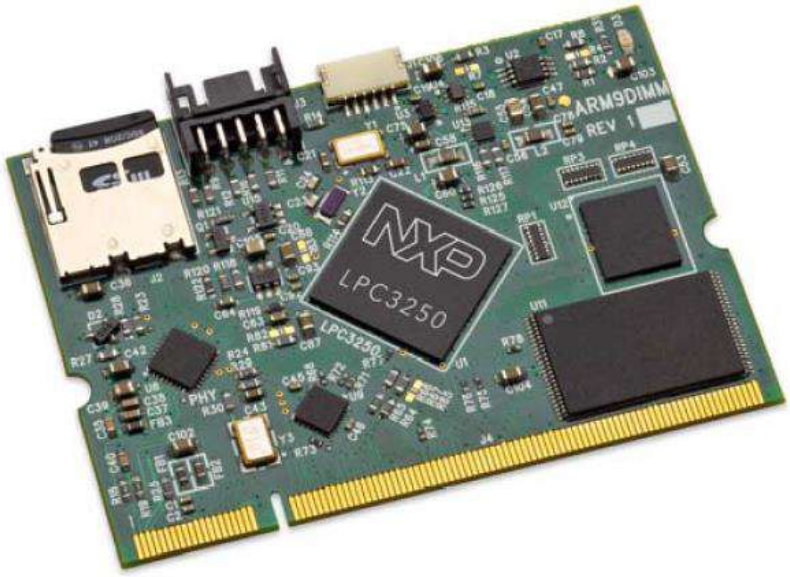
Además de contener todos los módulos de hardware antes mencionados que permiten la implementación de la plataforma de control, el ARM M4 de la **DigitalControlBoard** es el que nuclea las interfaces de comunicación con el exterior, entre ellas la interfaz serie, I2C, JTAG, CAN, SPI, USB y Ethernet. Dichas interfaces de comunicación están ya implementadas en el caso del serie o I2C, o en el caso de las demás, ya tenido en cuenta en el hardware y dejado listo para una futura expansión y agregado de funcionalidades adicionales al diseño del sistema.



Circuito esquemático general de comunicaciones y expansiones disponibles en PCB **DigitalControlBoard**.

La comunicación que se utilizó principalmente en las etapas del desarrollo del proyecto fue la comunicación serie, a través de la cual se ingresan diferentes comandos y parámetros hacia la plataforma, se eligen diferentes modos de testeo y se inicia o detiene la etapa de salida, y se informa del estado actual de este o alguno de sus parámetros seleccionados mediante una CLI (command line interface, o interfaz de comando).

Otra de las posibilidades de expansión que presenta este proyecto, dejada para una futura expansión es la del System on Module LPC3250 de NXP, el cual permitiría la posibilidad de implementar funcionalidades de más alto nivel y ejecutar Linux en dicha plataforma. Dicho sub-sistema puede verse como un PCB que se conecte a través del standard SOMDIMM a la **DigitalControlBoard**.



System on Module LPC3250.

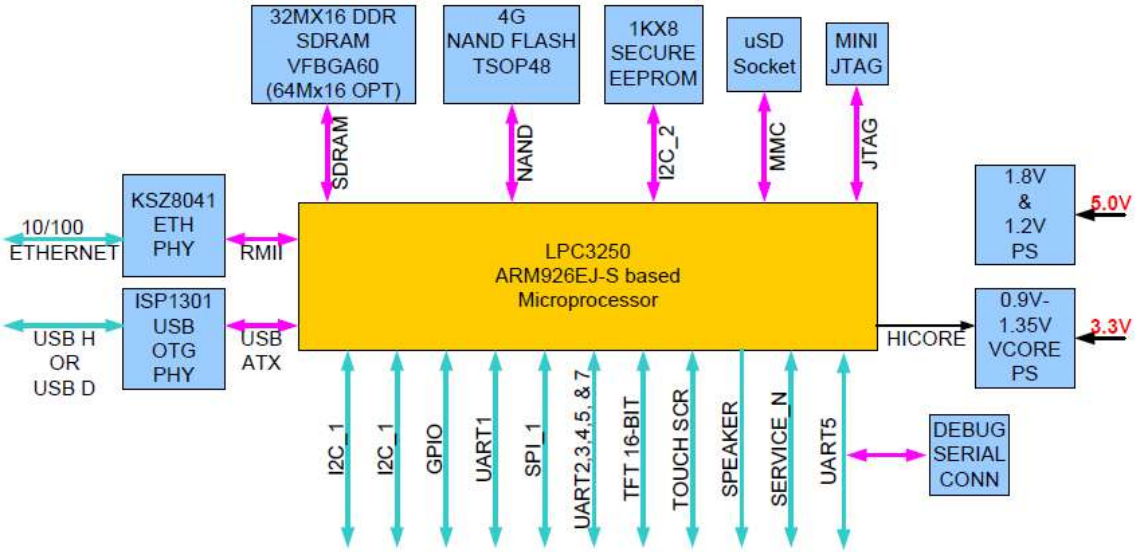


Diagrama en bloques de LPC3250.

Dicho modulo está pensado a futuro para agregar capacidades de monitoreo remoto de la plataforma y de la generación de una interfaz web para controlar y/o monitorear parámetros del inversor de potencia que este ejecutando sus funciones.

Para lograr un incremento de canales de PWM disponibles en la **DigitalControlBoard,** se utilizaron además otros dos CPLDs XC2C64A-7VQG44C como los utilizados en **AnalogFrontEnd**, de modo de duplicar los canales disponibles generando un complemento para un canal (n) determinado mediante la descripción de hardware de una serie de buffers tri-state como se puede observar a continuación:

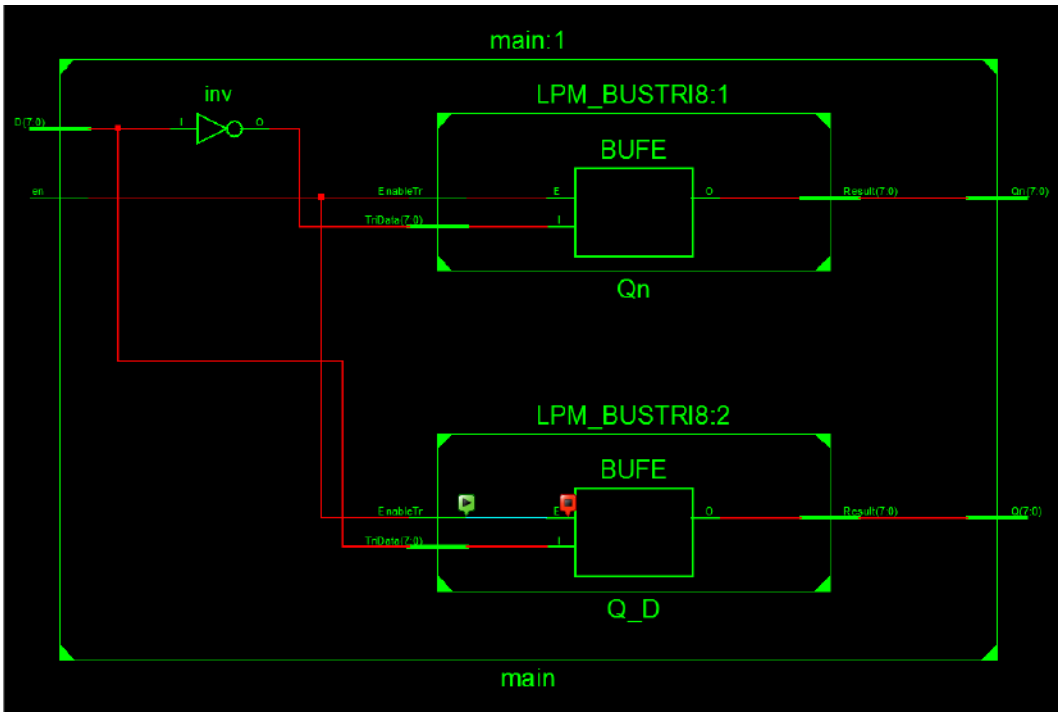


Diagrama RTL de CPLDs en **DigitalControlBoard** para expansión de canales PWM.

Estos CPLDs son alimentados por los ICs [TPS73201DBVR]. Estos son reguladores que permiten suplir a los CPLDs con las tensiones necesarias de $+3.3\,V$ y $+1.8\,V$.

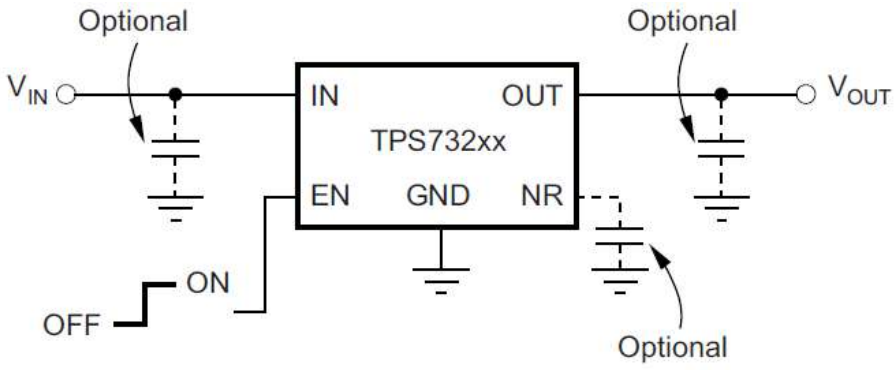


Diagrama de aplicacion tipica de reguladores TPS73201.

Puesto que ocurrió un error de ruteo en el diseño original por un footprint (huella) erróneo, se procedió a realizar un re-work que permita alimentar de forma correcta a los CPLDs. Luego, se testeo que estos funcionen correctamente con los cambios realizados sobre el PCB.

Se encontró además un problema de ruteo entre los CPLDs y el bus de PWMs (un desplazamiento entre lo diagramado en el circuito esquemático y los pines), por lo que se realizó un cambio en el HDL para reconfigurar los pines de los CPLDs y hacerlos coincidir sin realizar un re-work a nivel circuito impreso.

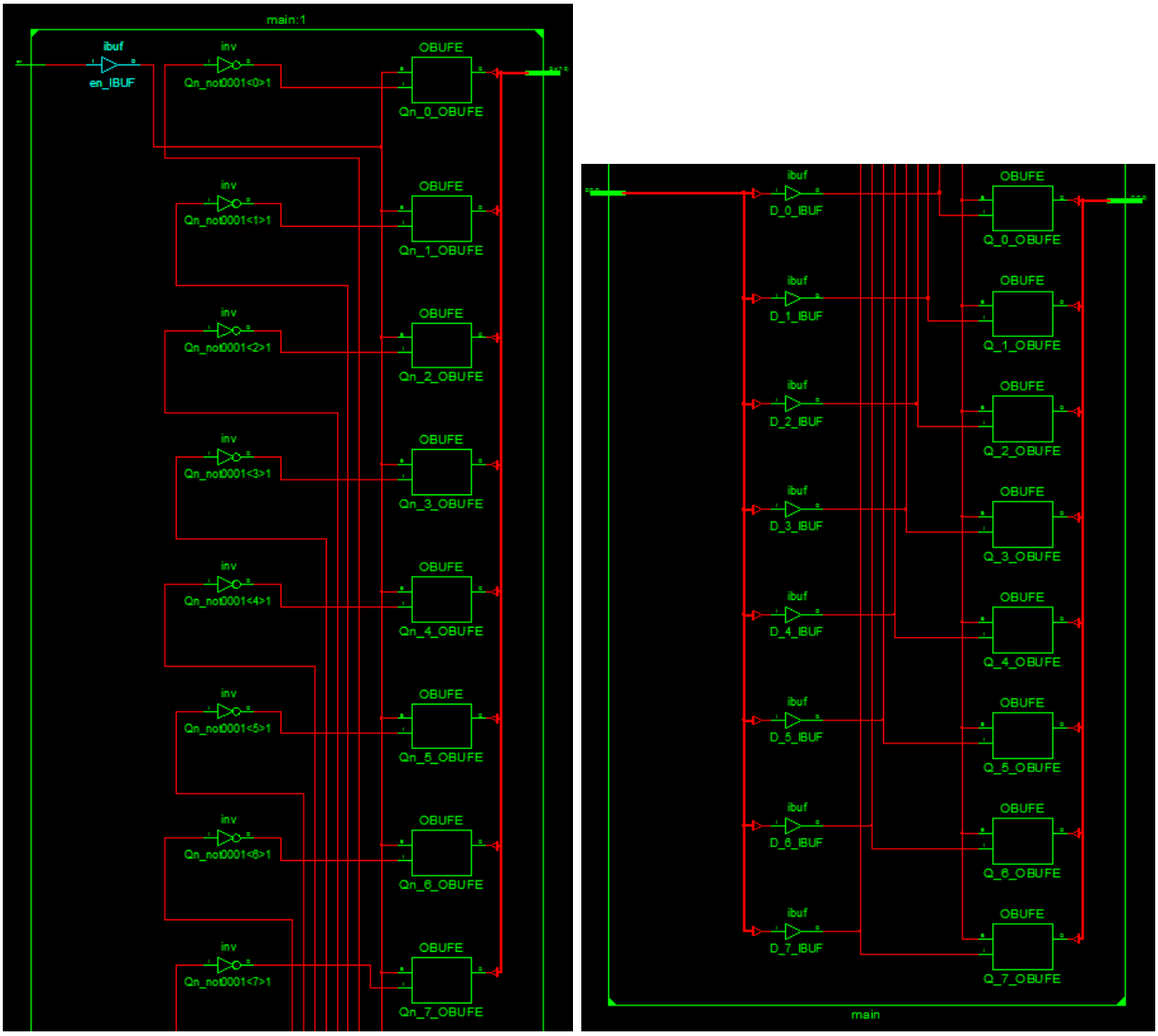
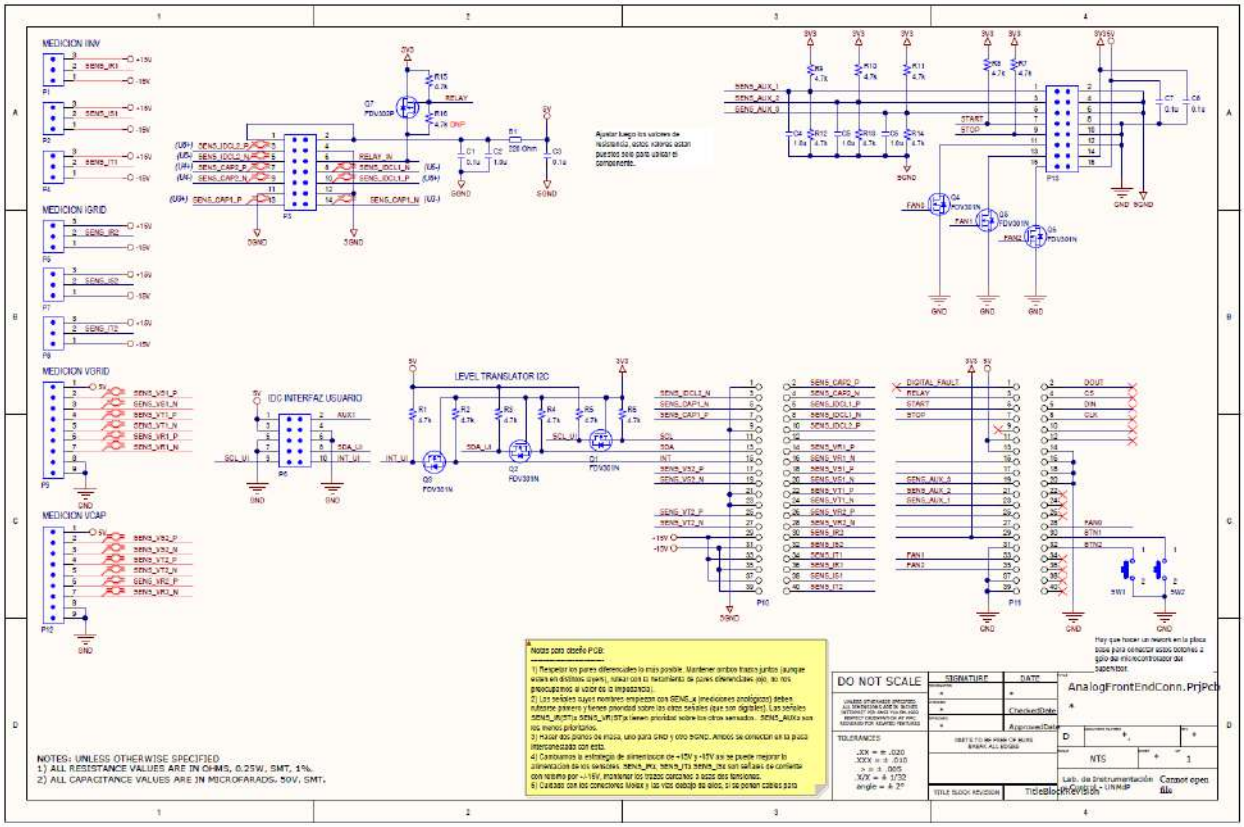


Diagrama de hardware sintetizado a partir de descripción en VHDL.

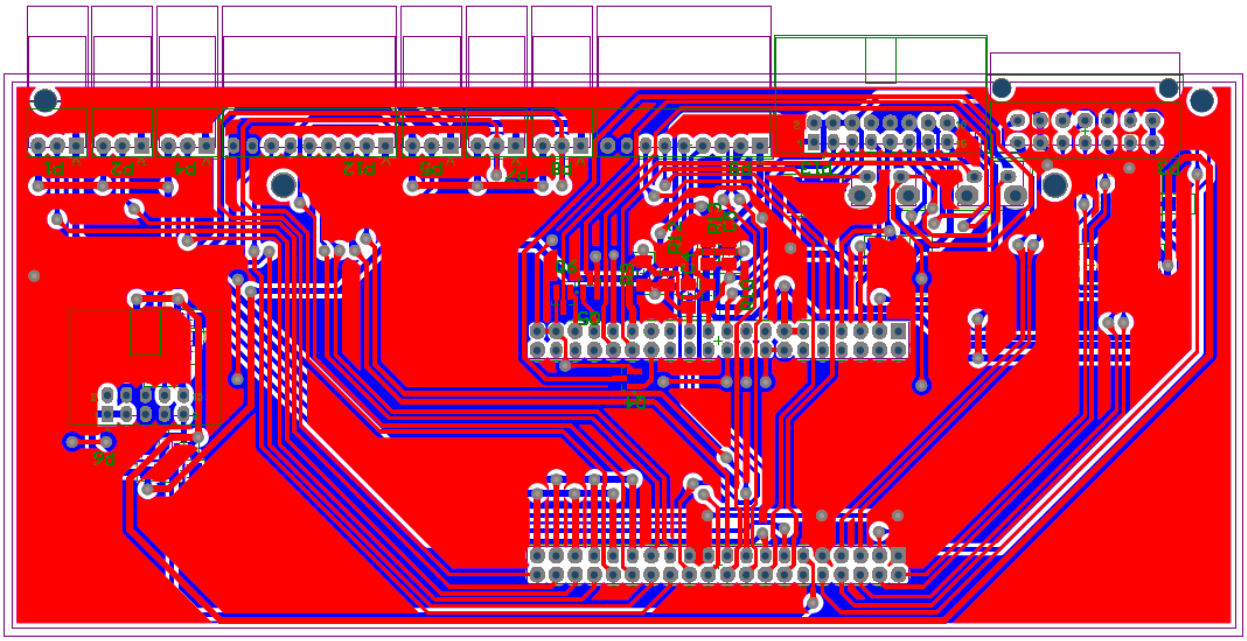
AnalogFrontEndConn:

Este PCB se realizó a modo de interfaz entre la electrónica de medición de magnitudes del PCB **AnalogFrontEnd** descripto anteriormente y los sensores propiamente dichos montados sobre el panel de testeo del LIC. Dicho PCB fue desarrollado teniendo en cuenta que iba a estar montado físicamente sobre la placa **AnalogFrontEnd** e interconectada a través de una serie de pines IDC (dicha placa está montada de manera piggyback) compatibilizando y mejorando las conexiones en el panel, las cuales estaban resueltas anteriormente con cables que generaban falsos contactos en el sistema. Además de lo mencionado anteriormente, el PCB permite suplir de alimentación a las etapas de medición de corriente para su correcto funcionamiento.

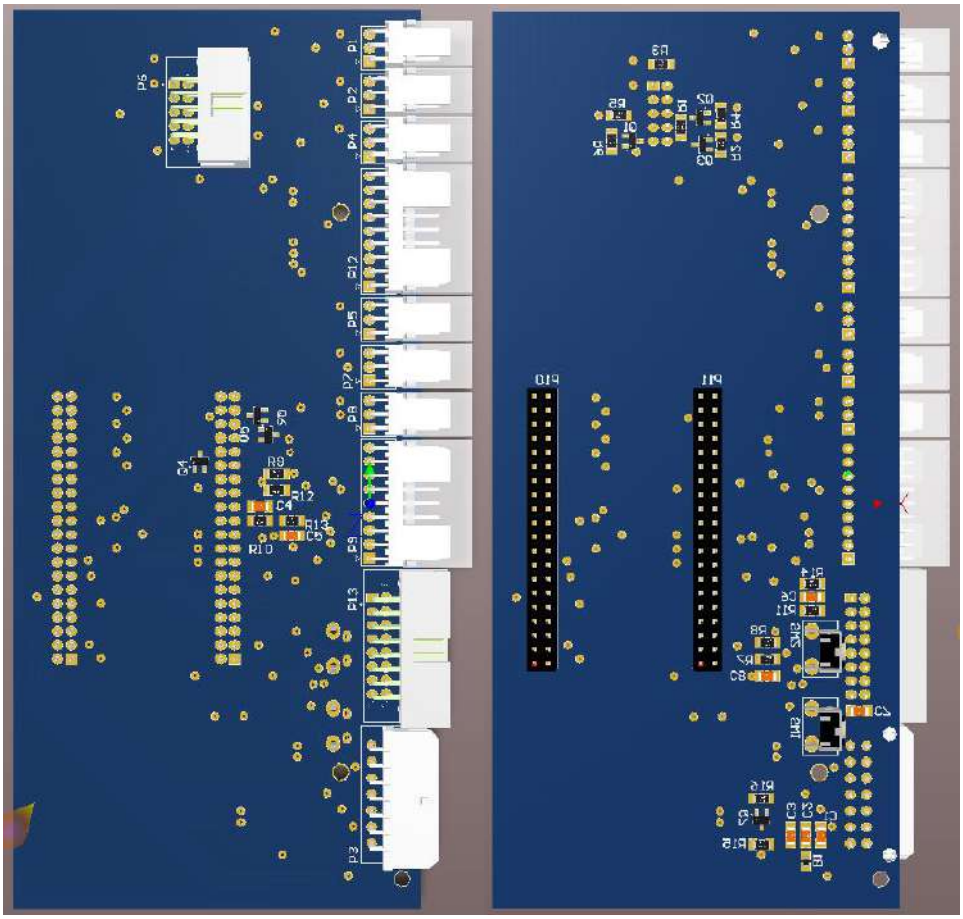


Circuito esquemático de PCB **AnalogFrontEndConn**.

Se encontró un problema generado por el cable que interconecta la lectura del DC Bus con el PCB, el cual generaba una caída y falsos contactos, por lo que este fue reemplazado por uno nuevo.



Layout de PCB **AnalogFrontEndConn** lado superior.



Render de PCB **AnalogFrontEndConn**.

InterfaceConn:

El PCB **InterfaceConn** se diseñó y realizó a modo de interfaz entre los canales de salida del PCB **DigitalControlBoard** y los PCBs que comandan y contienen a las llaves de potencia que generan las formas de onda trifásica que se inyectan en la red. Este PCB se conecta de manera directa al backplane mediante conectores HIROSE FX-2 que distribuyen las señales de salida de los canales PWMs y suplen la alimentación necesaria para los integrados CD4504BM96 que se utilizan para adaptar los niveles de señal necesarios para comandar las llaves de manera adecuada, puesto que las salidas de la **DigitalControlBoard** son de $+3.3\ V$ y los drivers esperan valores de $+15\ V$ o $+5\ V$ dependiendo de los utilizados en el panel de testeo del laboratorio, además de dejar disponibles conectores robustos para la interconexión entre las llaves antes mencionadas ubicadas en el panel de testeo del LIC y la plataforma de control.

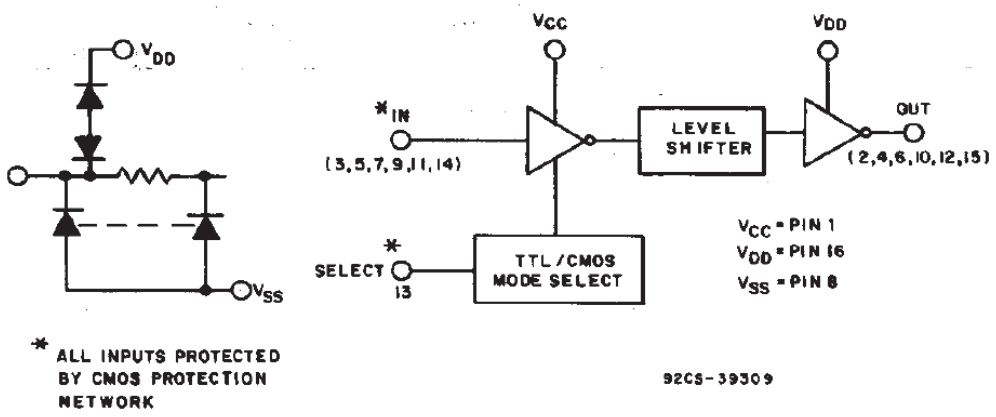


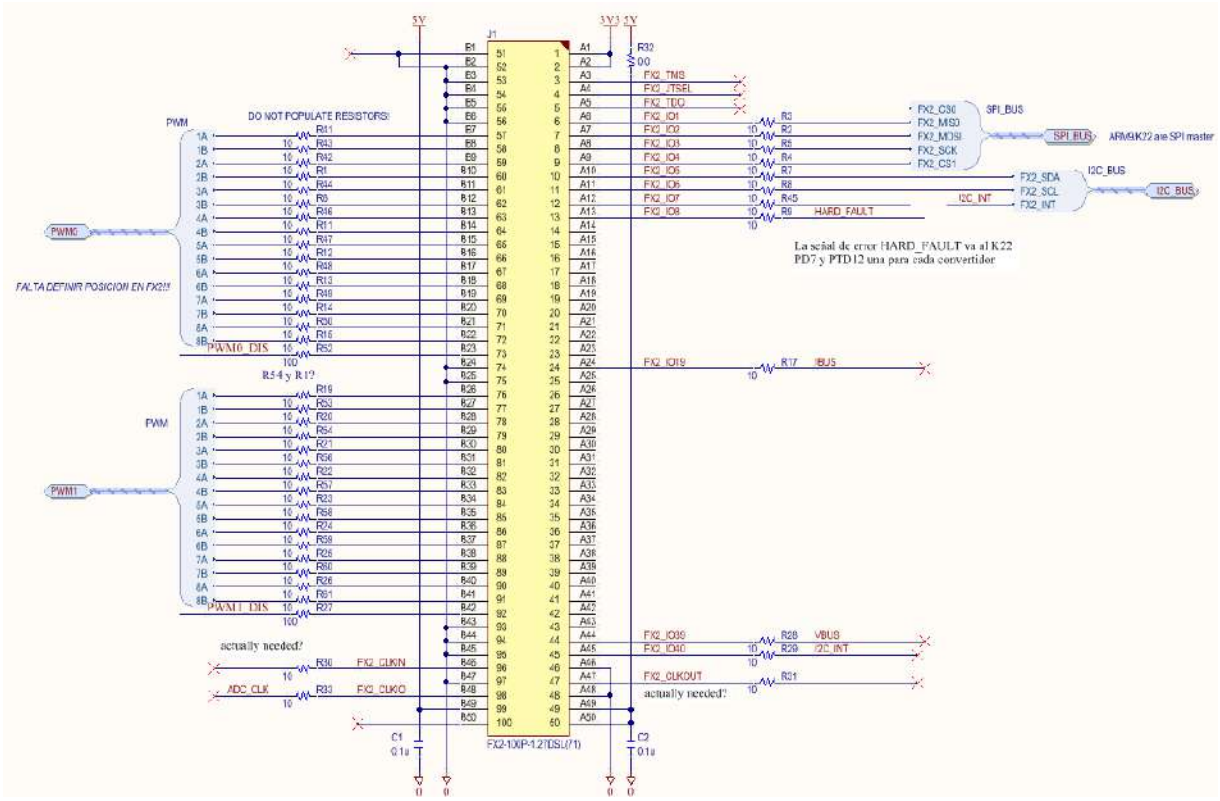
Fig. 1 - Functional diagram for CD4504B.

MAXIMUM RATINGS, ***Absolute-Maximum Values:***

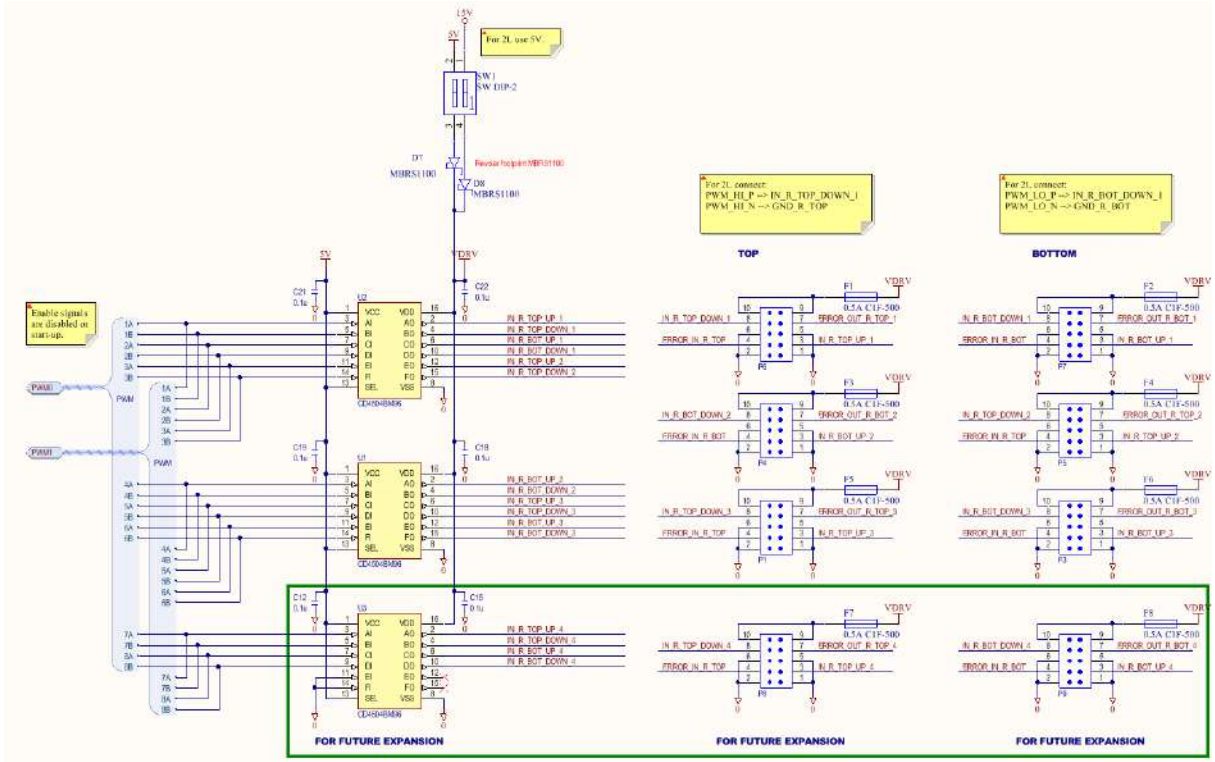
DC SUPPLY-VOLTAGE RANGE, (V_{DD})
(Voltages referenced to V_{SS} Terminal) ..... −0.5V to +20V
INPUT VOLTAGE RANGE, ALL INPUTS ..... −0.5V to V_{CC} +0.5V
DC INPUT CURRENT, ANY ONE INPUT ..... ±10mA

Diagrama en bloques del IC CD4504 y características de interés.

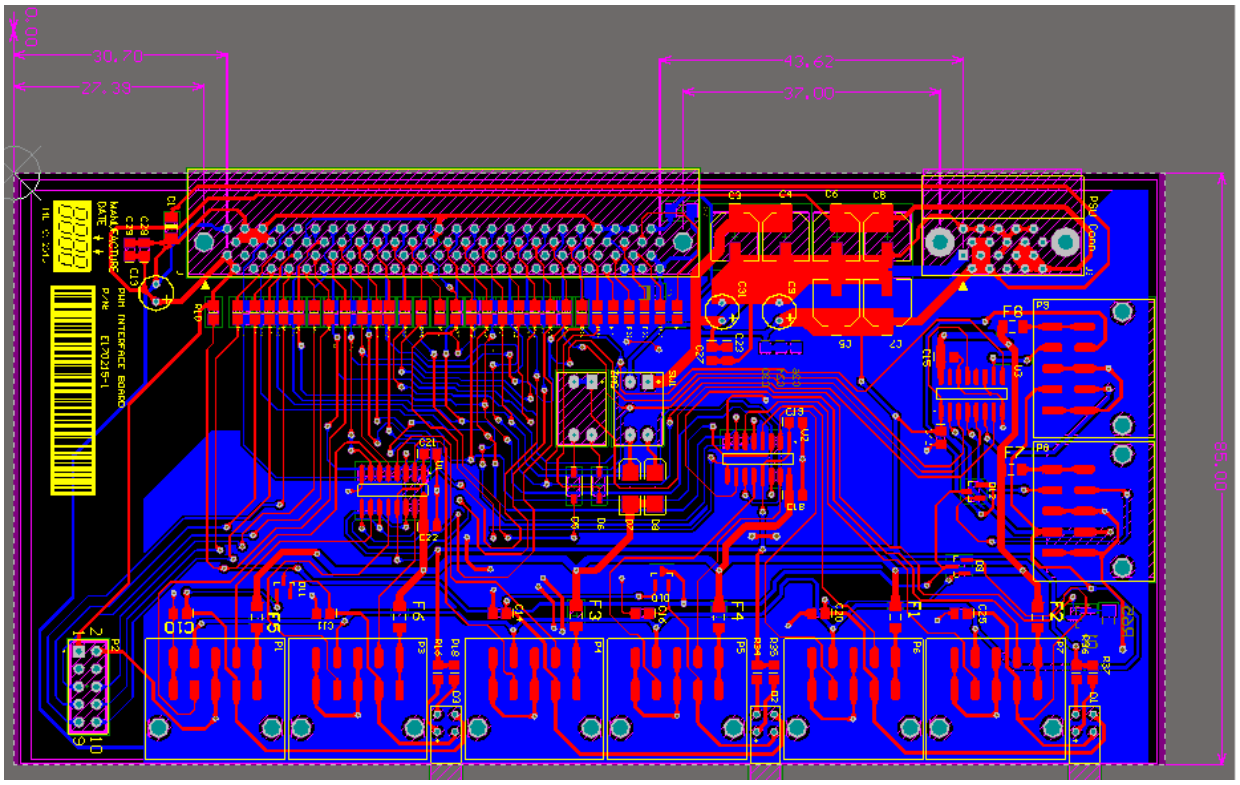
Se resolvió un problema de diseño debido a unos pull-ups de $1\ k\Omega$ que consumían demasiada corriente y generaban fallas aleatorias reemplazándolos por otros de mayor impedancia ($3.3\ k\Omega$).



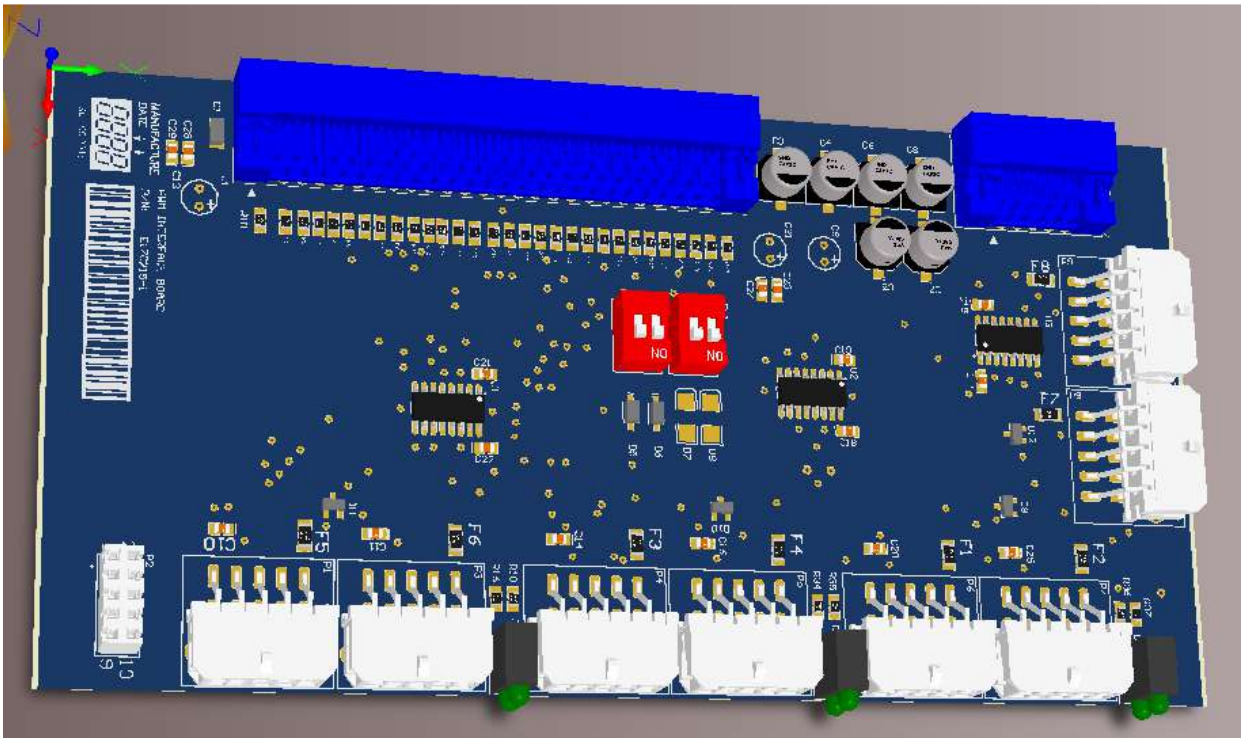
Circuito esquemático de PCB **InterfaceConn**.



Circuito esquemático de PCB **InterfaceConn**.



Layout de PCB **InterfaceConn lado superior**.



Render de PCB **InterfaceConn** en Altium Designer.

5. Diseño de Firmware de la plataforma:

En el siguiente capítulo, se describió el desarrollo, implementación y debugging del firmware de este proyecto. Dicho firmware se implementó sobre el microcontrolador Cortex-M4 K22P144M120SF5RM con todas sus ventajas y características antes mencionadas, además de poder utilizar como base o punto inicial para el desarrollo distintos ejemplos y demos provistos por el fabricante del microcontrolador.

Para esto se empezó por la elección de un IDE que permita programar dicho microcontrolador. Se optó por utilizar la sugerencia de los directores y se desarrolló en el entorno Eclipse versión Mars.2 para C/C++ al cual se le agregaron una serie de plug-ins llamados Eclipse Embedded CDT disponibles al momento de la realización práctica de este proyecto, que permiten desarrollar y debuggear código para los microcontroladores ARM en conjunto con la interfaz J-Link de SEGGER y el GDB Debugger (Toolchain de GNU).

Para el desarrollo práctico del proyecto, en particular para llevar un control de versiones y para el testeo de distintos módulos o funciones que debieran ser integrados al código principal, tanto desarrollados por quien escribe, como provistos por los directores, se optó por utilizar Git (plataforma open source que permite lo mencionado anteriormente).

Puesto que este proyecto es para un sistema embebido, muchas de las funciones desarrolladas en firmware pudieron ser testeadas y/o debuggeadas monitoreando magnitudes físicas, utilizando módulos ya integrados en el microcontrolador, o midiendo con osciloscopio digital a modo de verificar que el sistema funciona correctamente y se comanda al hardware de la manera esperada. En casos donde esto no fue posible o conveniente, también se debuggeo sobre el mismo código, exportando vectores de datos, utilizando breakpoints, etc.

Al desarrollar el firmware del proyecto, se tuvo en cuenta que a medida que se agregaban funcionalidades y se implementaban controles más complejos o matemáticamente intensivos, se iba a utilizar más poder de procesamiento y recursos, por lo cual hubo funciones o partes del código que fueron siendo depuradas, mejoradas u optimizadas para permitir que se puedan expandir aún más las funcionalidades del proyecto a futuro tanto por los directores que realizan investigación utilizando la plataforma como herramienta, como también para el desarrollo de futuros proyectos en base a este, y además que el proyecto sea factible y ejecute las funciones de manera adecuada con el hardware utilizado en el diseño.

Estructura del firmware:

En primera instancia se comenzó por testear de forma básica que el microcontrolador respondiera de forma esperada en el PCB **DigitalControlBoard**, inicializando y testeando GPIOs, probando algunas funcionalidades básicas, y debuggeando con las herramientas antes mencionadas sobre el main.

Una vez que se comprobó que este respondía de la forma esperada, se optó por definir la estructura del código. Para esto, se habló con los directores y se adoptó que esta sea en formato de librería con diferentes archivos .c, de modo que se puedan ir agregando funciones que tengan un código independiente o auto-contenido. A su vez, el código o las funciones se pueden clasificar en base a qué tan críticas son desde el punto de vista de su proceso en el tiempo, con funciones que disponen de una prioridad máxima (por ejemplo, las protecciones de hardware, que en caso de que no se ejecuten inmediatamente podrían ocasionar fallas graves o roturas) y otras de una prioridad inferior.

Teniendo en cuenta estas consideraciones, se realizó una implementación en dos niveles, donde se tienen las funcionalidades críticas corriendo a bajo nivel de manera directa sobre el programa principal llamando a determinadas funciones de la librería, y luego ciertas funcionalidades de menor demanda de prioridad, las cuales se implementaron en el FreeRTOS (RTOS o sistema operativo en tiempo real open source) o se dejó el framework ya funcionando para que estas se implementen a futuro como funciones dentro del mismo RTOS.

A nivel RTOS, se integró el kernel al código del proyecto y se generó una CLI (interfaz de líneas de comando) mediante una comunicación serie con una UART para poder ingresar los comandos previamente definidos con los que se puede interactuar con el sistema, pasar parámetros mediante queues, seleccionar opciones o funciones, como así también mostrar información de este, y se dejó listo el marco donde se puedan agregar o expandir las funciones disponibles.

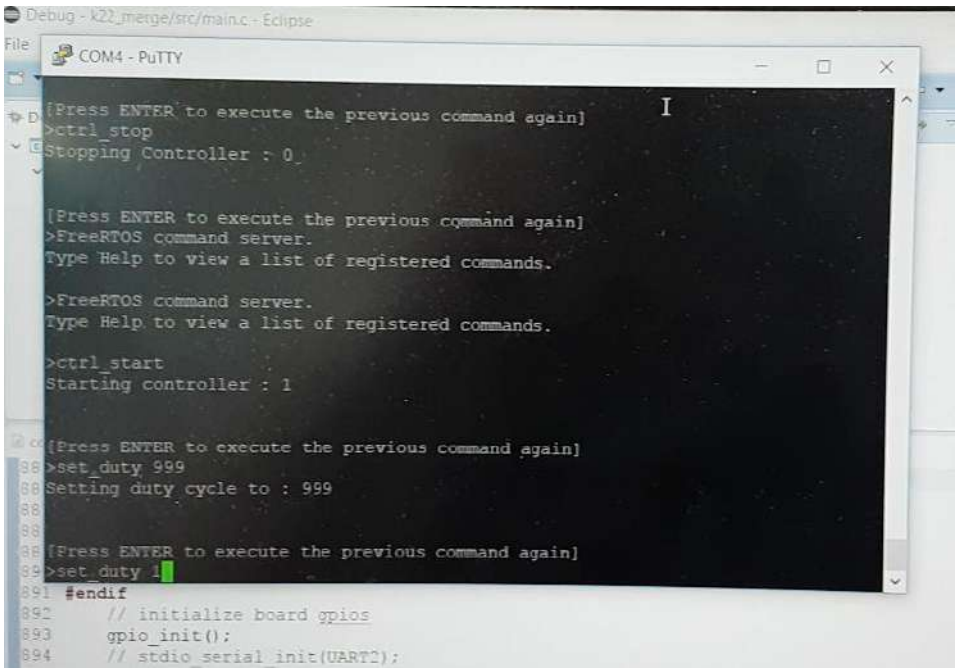


Imagen de CLI vía comunicación serie.

A continuación se da una lista de algunos de los comandos implementados que puede interpretar el código que gestiona el control a partir de la CLI:

- $Ref\_x\ (n)$: Asigna la corriente de alguna de las fases (r, s, t) al valor n.
- $Ref\_all\ (n)$: Asigna la corriente de todas las fases al valor n.
- $Synch\_type\ (b)$: Selecciona entre la sincronización de una fase y de tres fases (0: monofásico; 1: trifásico).
- $Ctrl\_sel\ (n)$: Selecciona el algoritmo de control entre los implementados en el código fuente a utilizar para calcular la acción de control que comanda las llaves de potencia (control p, control rpcc, control pr_rpcc actualmente implementados).
- $Ctrl\_type\ (n)$: Selecciona diferentes opciones dentro de las implementaciones de cada algoritmo de control.
- $Manual$: Selecciona un modo de testeo para las salidas, permite elegir un valor de ciclo de trabajo determinado (para debugging, etc.).
- $Duty\ (n)$: Ejecuta un duty cycle fijo determinado por el usuario.

- $Start$: Habilita las salidas y empieza a ejecutarse el algoritmo de control seleccionado.

- $Stop$: Detiene la ejecución del algoritmo de control elegido o del modo manual.

- $Reset$: Resetea al estado de $IDLE$ la plataforma, cera el estado del algoritmo de control seleccionado. Si existiera una condición de $FAULT$ espera que se accione el reset de hardware por seguridad.

A su vez, la parte del código de bajo nivel referida al control fue implementada dentro de una FSM en base al análisis de los posibles estados de la plataforma de control y la lectura de una serie de eventos definidos, que corre sobre la interrupción del módulo FTM antes descripto.

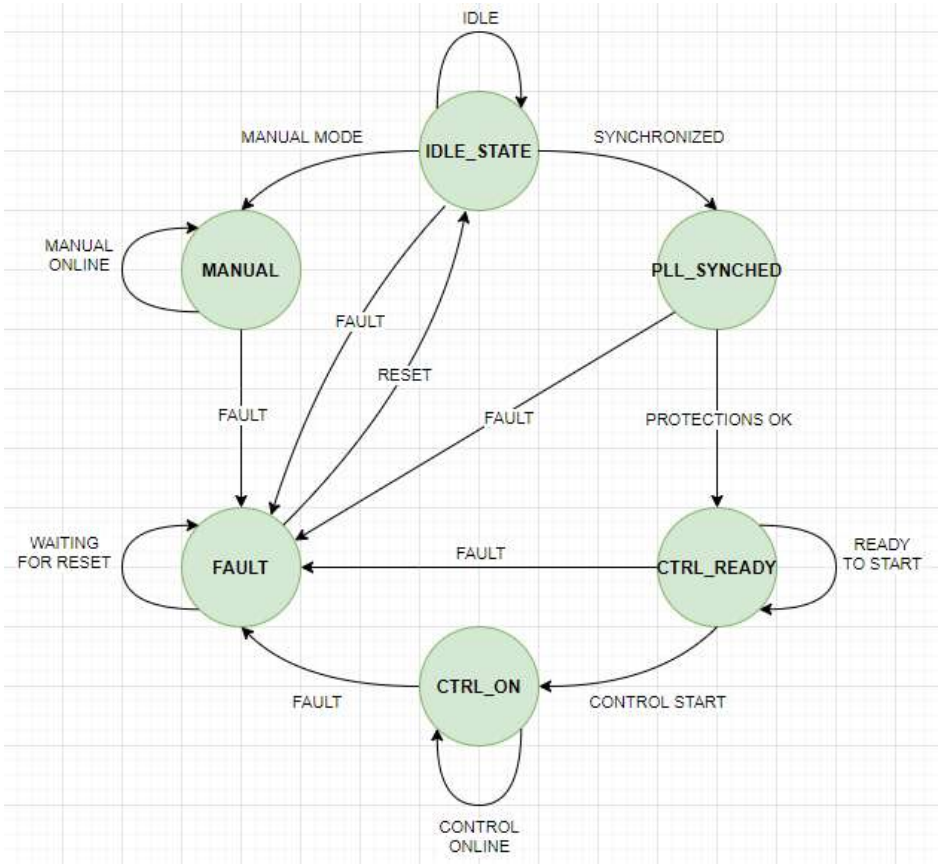


Diagrama de estados del sistema.

Dicha FSM contiene los siguientes estados:

- $\boldsymbol{IDLE\_STATE}$: Estado Idle o de reposo, es el estado inicial de la plataforma de control al inicializarse o luego de un reset, deja todas las salidas en estado inactivo y resetea los valores del algoritmo de control.

- $\boldsymbol{PLL\_SYNCHED}$: Estado con PLL sincronizado, donde se comprueba si la función de PLL implementada por firmware está funcionando correctamente, lo cual es un requerimiento para que la inyección de corriente a la red eléctrica funcione de manera adecuada.

- $\boldsymbol{CTRL\_READY}$: Estado Ready, donde se comprueba además que las tensiones del DC bus y de las fases están dentro de los límites impuestos por software. Este sería el estado donde la plataforma está en condiciones de comandar las llaves de la etapa de potencia e inyectar corriente a la red.

- $\boldsymbol{CTRL\_ON}$: Estado ON, en el cual se está ejecutando el algoritmo de control seleccionado y comandando la etapa de potencia mediante el módulo FTM.

- $\boldsymbol{FAULT}$: Estado de Fault o falla, dado por la detección de algún problema, en este estado se inhabilita y deja en estado inactivo todas las salidas, se resetean los parámetros del algoritmo de control que estaba corriendo y se cambian los valores de las banderas de control de los estados. Se puede salir de este estado accionando el reset de hardware del sistema.

- $\boldsymbol{MANUAL}$: En este estado de testeo, se accede directamente a comandos que permiten testear la salida de los canales PWM o de las llaves de potencia indicando con comandos entre otras cosas el duty cycle, solo se puede acceder desde un comando en la CLI.

El cambio entre estados se da en base a la función **`READ_EVENT`** la cual monitorea una serie de flags y señales, tales como si el sistema se sincronizó, si se acciono el reset del sistema, y si las tensiones y corrientes están dentro de los limites definidos por firmware, menores a los que definen las protecciones de hardware mencionadas en el capítulo anterior (que también dispararían el estado de $\boldsymbol{FAULT}$), y sirven como redundancia en caso de que no se dispararan por algún motivo o fallaran.

Implementación de funcionalidades en el sistema:

Los módulos descriptos anteriormente en el hardware del ARM Cortex-M4 fueron configurados y utilizados en el firmware a modo de obtener las funcionalidades requeridas. En el caso del FlexBus y el PDB, estos fueron utilizados para implementar la lectura de datos y la interrupción para la adquisición respectivamente, y la interrupción maestra a partir de la cual el sistema ejecuta la toma de valores por el control y la implementación de la acción de control mediante PWMs y control de fallas se hizo a partir del módulo FTM.

Para la implementación de la adquisición, como fue mencionado, es necesaria la señalización de inicio, un tiempo de procesamiento para obtener las mediciones correspondientes por el hardware, y posteriormente una etapa de lectura para disponer de estas mediciones en formato digital en el microcontrolador.

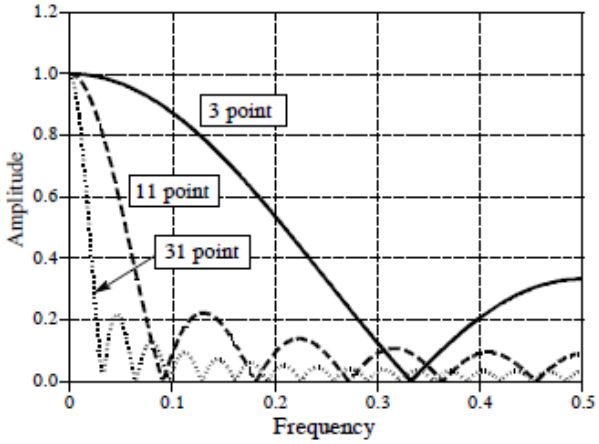
Dado que el inversor funciona a una frecuencia de conmutación de $10\ KHz$, si se utilizara solo la interrupción provista por el módulo FTM, la adquisición también ocurriría a esta frecuencia. Como es deseable implementar un filtrado adicional vía firmware para el cual se requieren más muestras dentro de cada periodo de conmutación, se utilizó el módulo PDB, el cual genera interrupciones a partir de un tiempo pre-programado.

A partir de esto, se implementó la ejecución de tres ciclos de adquisición por cada periodo de FTM. Cada interrupción generada por el FTM provoca que se ejecuten tres interrupciones a partir del PDB. Cada una de ellas inicia el proceso de adquisición y la posterior lectura secuencial de cada canal de adquisición por el módulo FlexBus.

Luego de la toma de las muestras, se realiza un filtrado digital y se obtienen las lecturas filtradas para su utilización por el algoritmo de control. La implementación del filtrado digital es del tipo moving average y se realizó a partir de un cálculo sobre las muestras mencionadas, de las cuales se obtiene un valor promedio para cada periodo. Este filtrado elimina principalmente ruido de alta frecuencia en el que puede estar inmersa la magnitud que se desea medir.

$$y[i] = \frac{1}{M} \sum_{j=0}^{M-1} x[i+j]$$

Ecuación del filtro moving average implementado, donde $y[\,]$ es la señal de salida, $x[\,]$ es la señal de entrada, y M es la cantidad de muestras tomada para el promediado.



Respuesta en frecuencia del filtro moving average.

A continuación, se describen las funciones realizadas para la implementación de la etapa de adquisición mediante la utilización de los módulos de hardware correspondientes:

- **`void FLEXBUS_INIT (void):`** Función de inicialización y configuración del FlexBus. Este se configura para funcionar a 7.5 MHz, en modo de 16 bits de direccionamiento y lectura de datos con los Chip Select correspondientes.

- **`void PDB_INIT_FUN (void):`** Función de inicialización y configuración del PDB. Configurado para ejecutar la interrupción de inicio a partir del FTM, en modo continuo, y con un módulo de 2000 (el módulo de periodo del FTM es de 6000, por lo que se obtienen tres interrupciones para disparar una adquisición en cada una de ellas).

- **`void PDB_IRQHandler (void):`** Función de servicio de interrupción del PDB. Ejecuta la función **`ACQ_START`** que inicializa un nuevo ciclo de adquisición y toma de datos.

- **`void ACQ_IRQHandler (void):`** Función de servicio de interrupción de adquisición. Detecta el fin de conversión, a partir del cual pueden leerse datos válidos con la función **`ACQ_READ.`**

- **`void ACQ_START (void):`** Función de inicio de ciclo de conversión.

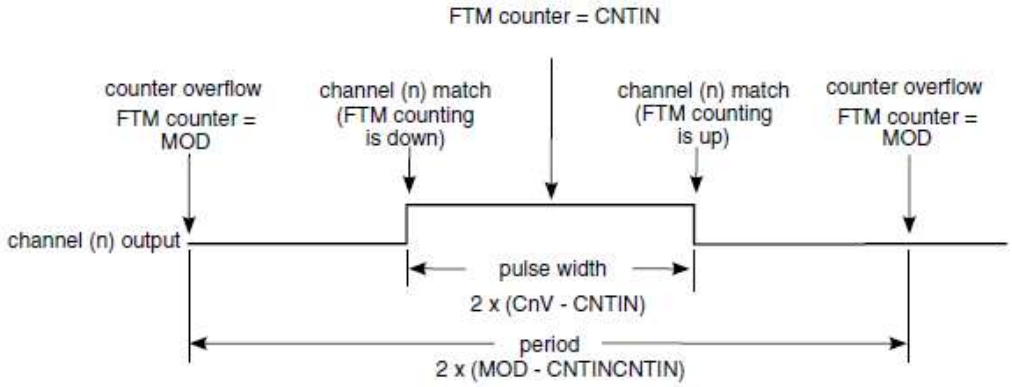
- **`void ACQ_READ_n (void):`** Función de lectura de datos en PCB **AnalogFrontEnd** A o B, a partir del FlexBus utilizando la máscara correspondiente y almacenando los datos obtenidos en vectores para su posterior procesamiento, además de restar el offset inherente de cada canal (dichos offset fueron obtenidos a partir del testeo del sistema).

- **`void ACQ_FILTER (void):`** Función de filtrado digital del tipo moving average implementada a partir de las muestras tomadas en cada ciclo de conmutación

Otro de los módulos implementados en las funcionalidades del sistema es el FTM, el cual se utilizó para sintetizar el control, en particular este se encarga del control de los canales de PWM e interviene de diferentes maneras teniendo en cuenta la máquina de estados descripta anteriormente. La configuración del módulo depende tanto de parámetros de diseño tales como la frecuencia de conmutación de las llaves (que define un periodo de PWM) como así también la lógica de los drivers de las llaves de potencia ya implementados en el panel de test del laboratorio entre otras cosas.

En este caso la configuración es del tipo complementaria como se comentó en la descripción del módulo de hardware, es decir, los canales de PWM (n) y $(n+1)$ tienen un valor inverso entre sí, pero comienzan todos en estado inactivo.

Además la lógica es activo bajo y el esquema de PWMs es center-aligned tal que:



Esquema de PWM “center-aligned”.

El módulo de cada periodo Ts del PWM de modo que:

$$Ts = \frac{BUS\_CLK}{INV\_PWM\_FREQ} = 6000$$

Siendo $BUS_{CLK} = \frac{SYY\_CLK}{2} = 60\ MHz$ e $INV\_PWM\_FREQ\ =\ 10\ KHz$

Se configuró el valor de tiempo muerto insertado en los canales de PWM en $2\ \mu s$. Este se configuró a partir de un valor inicial y se ensayó experimentalmente teniendo en cuenta la hoja de datos de las llaves de potencia.

También se habilito la opción de detección de fallas para la inhabilitación de las salidas de PWM en caso de que ocurra un problema en el sistema, en caso de que ocurriera un problema todas las salidas quedan en estado alto (sin conducción).

Para la utilización del módulo, se definieron funciones para su inicialización y configuración como así también la implementación de funciones que controlan las salidas PWM a partir de dicho módulo, Entre ellas:

- `void CTRL_INIT (void):` Función de inicialización de la etapa de control, esta inicializa el módulo FTM como así también otros módulos integrados en el microcontrolador para su uso en debugging o testeo experimental.

- `void INV_INIT (void):` Función de inicialización del FTM propiamente dicha, habilita el clock y GPIOs de este, como así también su configuración inicial y el IRQ del módulo.

- `void ACQ_VALUES (void):` Función donde se toman los valores obtenidos por la adquisición luego de ser filtrados digitalmente para su utilización por el algoritmo de control seleccionado. Además, en esta función se realiza el ajuste fino de ganancias para cada canal de adquisición para mapear el valor y que represente la magnitud a escala real. Por último, esta función también contiene el término de corrección de balance del DC Bus (medido experimentalmente).

Además de las funciones antes mencionadas se debieron integrar una serie de funcionalidades para la implementación del inversor de potencia, entre ellas la generación de señales de referencia, la sincronización del sistema a la red eléctrica, y los algoritmos de control que se desean testear en el sistema para verificar su performance de forma experimental utilizando la plataforma desarrollada.

La generación de las referencias y la sincronización se implementó en base a una librería de firmware ya implementada previamente proporcionada por los directores, que se integró al proyecto en cuestión. Dicho código se integró en una serie de funciones, entre ellas:

- `void GEN_REF (void):` Función realizada para generar las referencias sinusoidales en base a la librería provista, modifica los vectores de referencia utilizados por el control.

- `void vsf_pll_step (const float vg_ab[], bool tph):` Función utilizada para la sincronización del sistema a partir de un PLL realizado vía firmware. A partir de esta, se obtiene un nuevo periodo Ts utilizado como maestro por todo el sistema (adquisición, FTM, algoritmo de control, etc.) como así también una bandera que indica si el sistema esta propiamente sincronizado utilizado por la FSM.

Para la función de sincronización en particular se tuvo que trabajar en un problema en el código original del PLL, que generaba jitter y drift, lo que no permitiría la correcta sincronización del sistema a la red eléctrica. Este problema fue detectado realizando pruebas en la plataforma y se solucionó modificando el código y agregando un Filtro de Goertzel para lograr el objetivo de sincronizarse a la red sin incertidumbres en la frecuencia y fase.

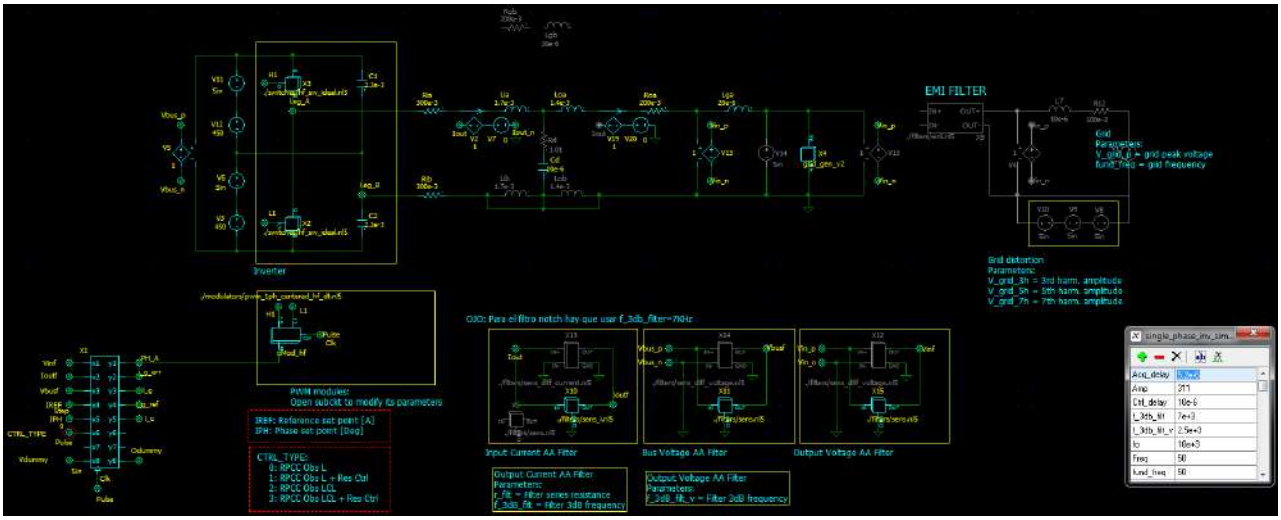
Puesto que el objetivo de este proyecto es la realización de una plataforma que permita implementar y evaluar diferentes algoritmos de control para el inversor de potencia, tanto desde el punto de vista del procesamiento requerido, como así también de comparar ciertos indicadores de performance obtenidos en forma experimental al utilizar el sistema, se realizó la integración de diferentes controles digitales.

Dichos controles, provistos por los directores, se integraron como archivos de código fuente y funciones, que como se mencionó anteriormente, pueden ser seleccionados en la CLI con las funciones $Ctrl\_sel\ (n)$ y $Ctrl\_type\ (n)$ para ver las diferentes respuestas del sistema a cada uno de ellos y sus variantes de implementación.

Entre ellos se encuentran los siguientes algoritmos de control:

- $Control\_p$: Implementación de control proporcional como una referencia de base para la evaluación respecto a otros algoritmos de control y verificación del funcionamiento del sistema.
- $Control\_rpcc$: Implementación de control de corriente predictivo robusto a modo de testear y verificar una opción válida para la implementación del sistema.
- $Control\_pr\_rpcc$: Implementación de control de corriente predictivo robusto con un compensador adicional PR que agrega términos resonantes y logran un mayor rechazo a la frecuencia de red y sus armónicos más importantes.

Estas estrategias de control fueron simuladas en conjunto con los directores del proyecto en NL5 para verificar el funcionamiento del algoritmo de control a integrar al sistema, y predecir y evaluar la respuesta que estos tendrían en el sistema real con las características de la planta ya tenidas en cuenta.



Simulación de algoritmos de control implementados (monofásica)

También se implementaron otras funciones auxiliares y de debug en el sistema. Entre ellas se encuentran las funciones de inicialización para utilizar los ADCs y DACs integrados en el microcontrolador, como así también la inicialización de la comunicación serie vía UART, un heartbeat que muestra si ocurrió una falla y el sistema pasó al estado de FAULT con indicadores visuales de LEDs, la inicialización del scheduler y el registro de los comandos implementados en la CLI para que el RTOS los pueda leer en la comunicación serie, entre otras cosas.

Testeo, optimización y debugging:

Otro de los puntos donde se hizo hincapié en este proyecto desde el punto de vista del firmware, fue en la utilización de los recursos del microcontrolador.

Puesto que por un lado los algoritmos de control a implementar para su evaluación y testeo son intensivos desde el punto de vista del procesamiento como así también está la posibilidad de una futura expansión de funcionalidades que también requerirían tiempo libre de ejecución, a medida que se fueron realizando las distintas etapas del firmware estas se fueron analizando y reescribiendo a modo de lograr menores tiempos de cómputo, un código más compacto y una mayor performance del sistema.

Para ello, se realizaron testeos experimentales en base a diferentes versiones de código y midiendo los tiempos de procesamiento en banco de prueba con osciloscopio utilizando diferentes modos de testeo en base a breakpoints y conmutando pines a modo de banderas de hardware para poder realizar mediciones sobre los distintos tiempos de proceso críticos del sistema, tales como el tiempo de adquisición (principalmente la lectura de todos los canales) y el tiempo de procesamiento para diferentes implementaciones de controles con más o menos cantidad de términos a computar por el sistema.

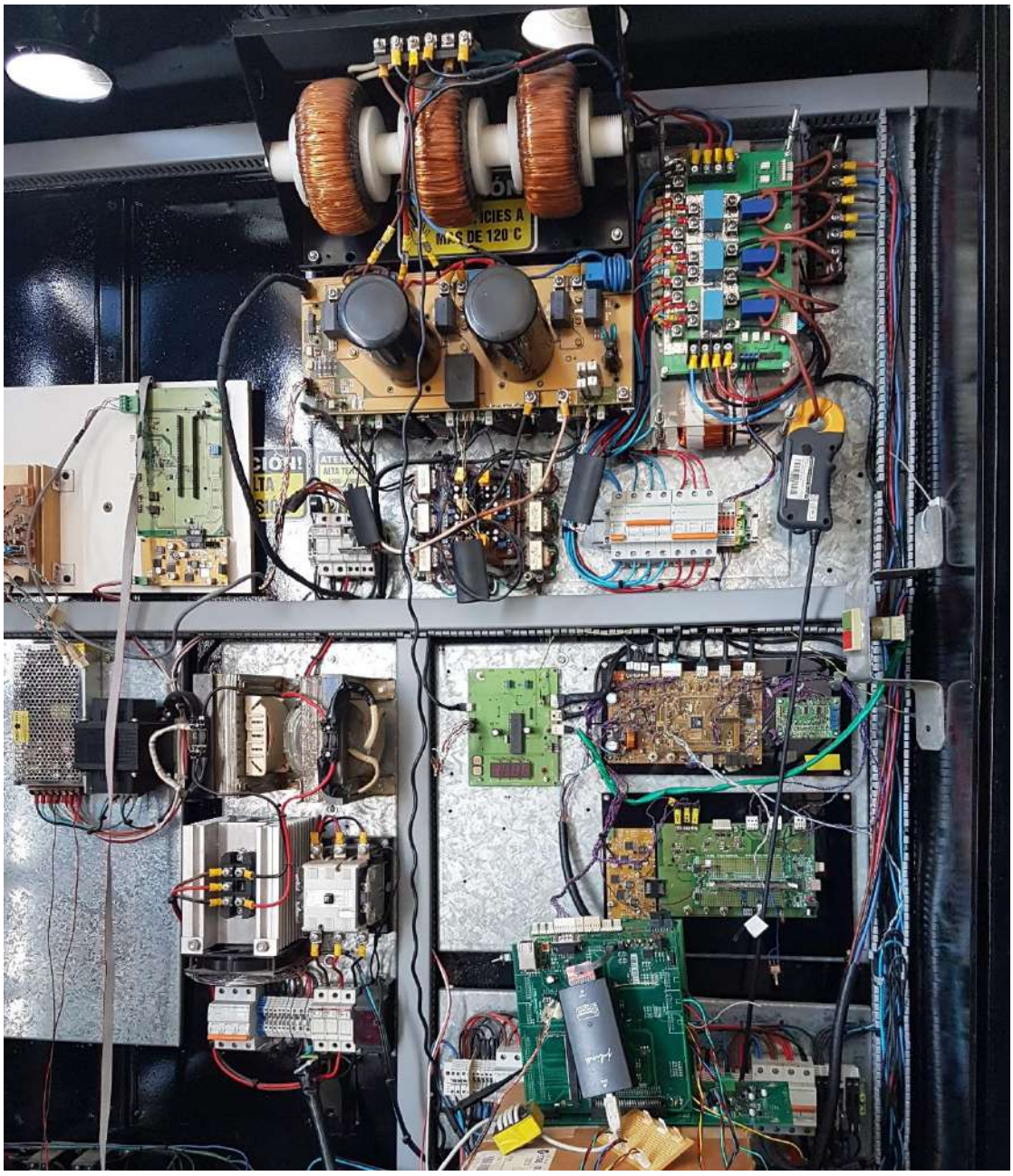
Una de las estrategias utilizadas fue la opción de reemplazar la utilización de funciones pre-definidas para el set y clear de bits que consumen tiempo adicional por la utilización de bit-banding, una feature de uso opcional de los ARM Cortex-M4 que permite realizar estas operaciones de manera más eficiente. También se utilizó bit-banding en las operaciones de lectura de los canales del ADC puesto que permiten un acceso más rápido a memoria.

Además de la utilización del bit-banding antes mencionado, se realizó una implementación donde ciertas funciones que demandan gran tiempo de procesamiento son copiadas a la RAM del sistema, lo cual genera un incremento de performance. Esto se definió como un atributo para las funciones que se copian a la memoria con el atributo **`_ramfunc_`** y se utilizó en particular para las implementaciones de los algoritmos de control y la función de lectura del FlexBus.

Por último, se realizó una configuración del compilador para optimizar las tiempos de procesamiento y se mantuvieron versiones del firmware de debug para el testeo inicial de funcionalidades y una versión optimizada, la cual tiene tiempos de ejecución menores y más recursos de procesamiento disponibles.

6. Resultados y Conclusiones:

En este capítulo se plasmaron los resultados obtenidos en base a los testeos y validaciones tanto en las distintas etapas de desarrollo como del sistema completo en funcionamiento, y la respuesta del sistema en base a los distintos algoritmos de control implementados y su performance.

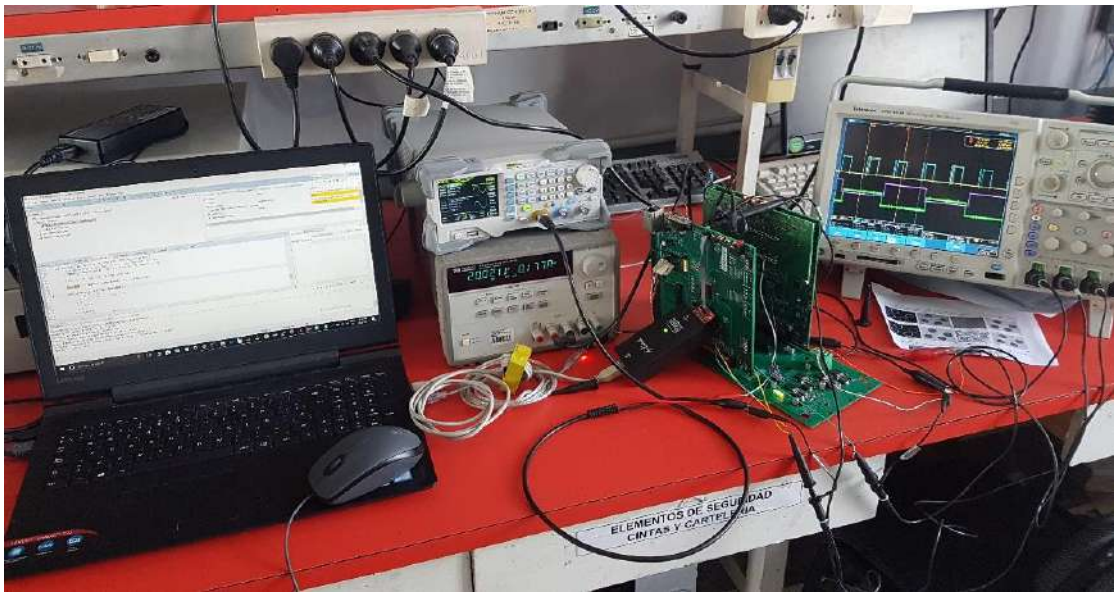


Sistema completo en etapa de testeo sobre panel del LIC.

Testeo de etapas y funciones del sistema:

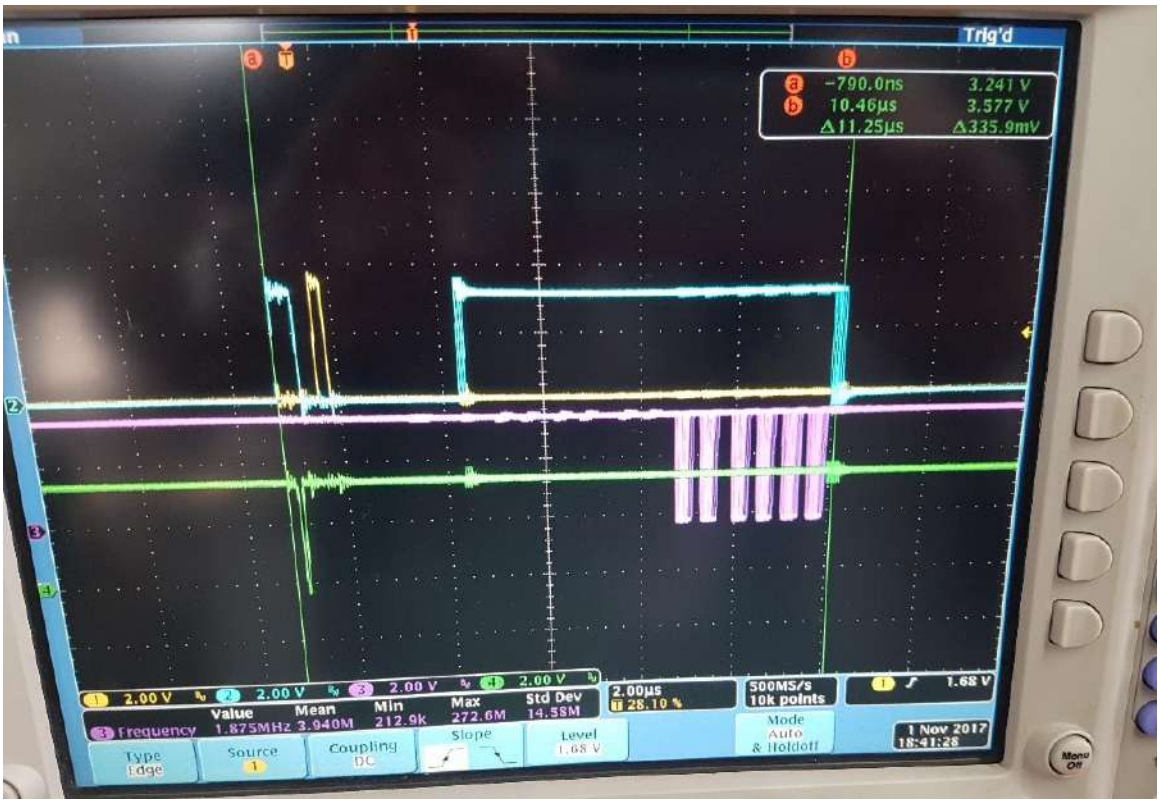
Validación y testeo experimental del sistema de adquisición y canales PWM:

La etapa de adquisición se testeo y valido de forma experimental tanto en banco de prueba como funcionando como sub-sistema del inversor de potencia. A continuación se puede observar el banco de prueba utilizado para dicho testeo:

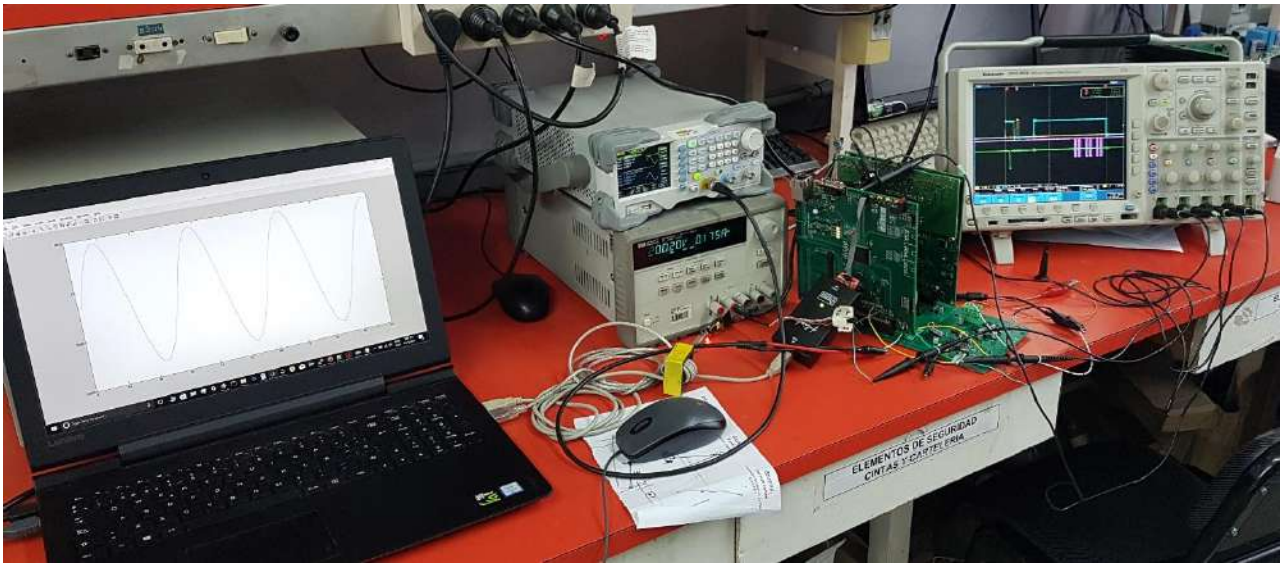


Banco de prueba para testeo y validación del sistema.

Validación de tiempos de adquisición, se puede observar en la imagen que ocurre la toma de tres muestras por periodo de conmutación y la etapa de conversión y lectura.



Señalización de ciclo de adquisición, se pueden observar las etapas de inicio de conversión y de lectura de datos en celeste, con la lectura propiamente dicha por el FlexBus en violeta.



Banco de prueba de adquisición, inyectado de señal de testeo sinusoidal y gráfico de vector de muestras adquiridas en MatLab.

Como parte del testeo del sistema, se inyectan señales sinusoidales a cada canal de adquisición y estas son medidas por la etapa de adquisición descripta anteriormente. A partir de las tres muestras obtenidas por periodo de conmutación, se realiza el filtrado y su escalado correspondiente dependiendo si es una magnitud de voltaje o de corriente y se guarda en vectores auxiliares para luego ser exportadas a MatLab y graficarse, a modo de verificar que la etapa adquiera sin distorsión o pérdida de datos.

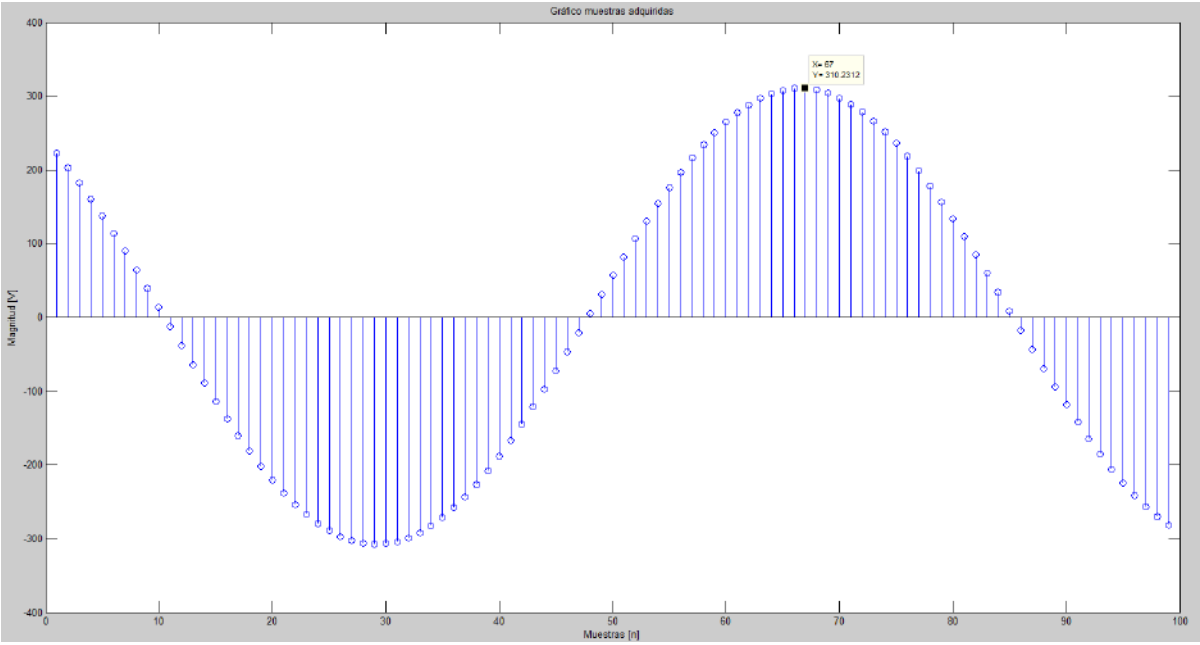
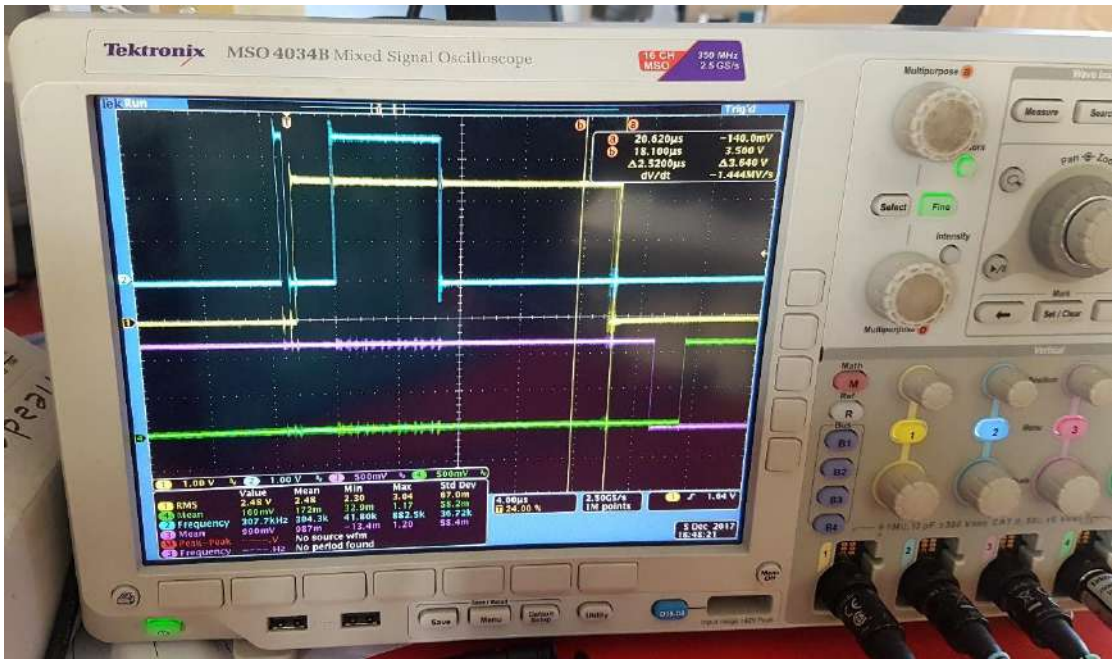


Gráfico de vector de muestras de un canal de adquisición (en este caso de tensión).

Otra de las funciones a verificar es la funcionalidad de los canales de PWM realizados a partir del FTM. Aquí se verifica que los canales tengan las características esperadas a partir de su configuración y el agregado correspondiente de los tiempos muertos a los canales de PWM.



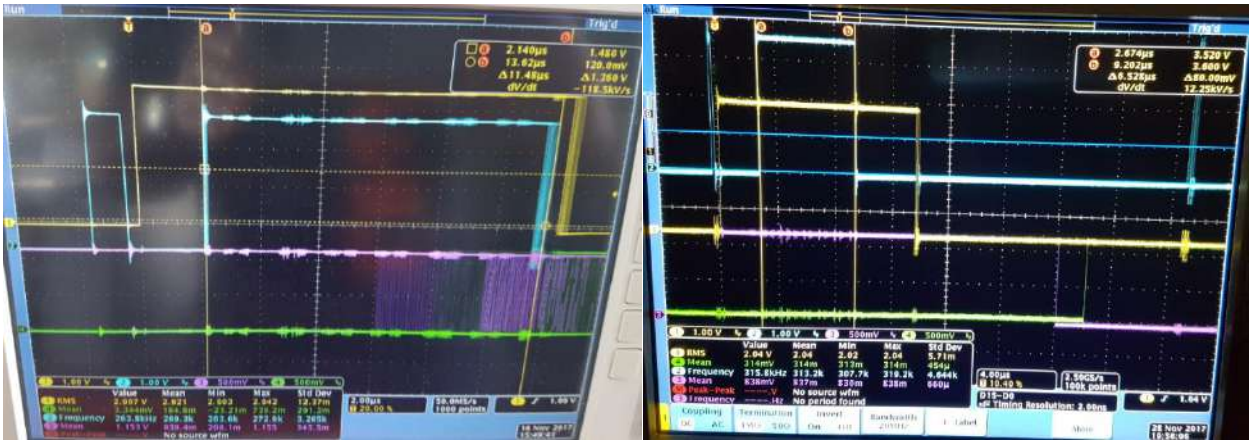
En verde y violeta un canal de PWM determinado, la brecha entre los cambios de estado de ambas señales se debe a los tiempos muertos insertados ($2\ \mu s$).

Característica de conmutación de IGBTs medida con osciloscopio, brecha debido a la inserción de tiempos muertos.

Resultados de las optimizaciones aplicadas sobre el firmware:

En estas capturas, se pueden apreciar los tiempos de ejecución antes y después de aplicar las optimizaciones mencionadas anteriormente en el proceso de adquisición (para 12 canales funcionando, equivalente a un PCB **AnalogFrontEnd**):

Comparativa experimental de optimización de tiempo de procesamiento en etapa de hold de adquisición.

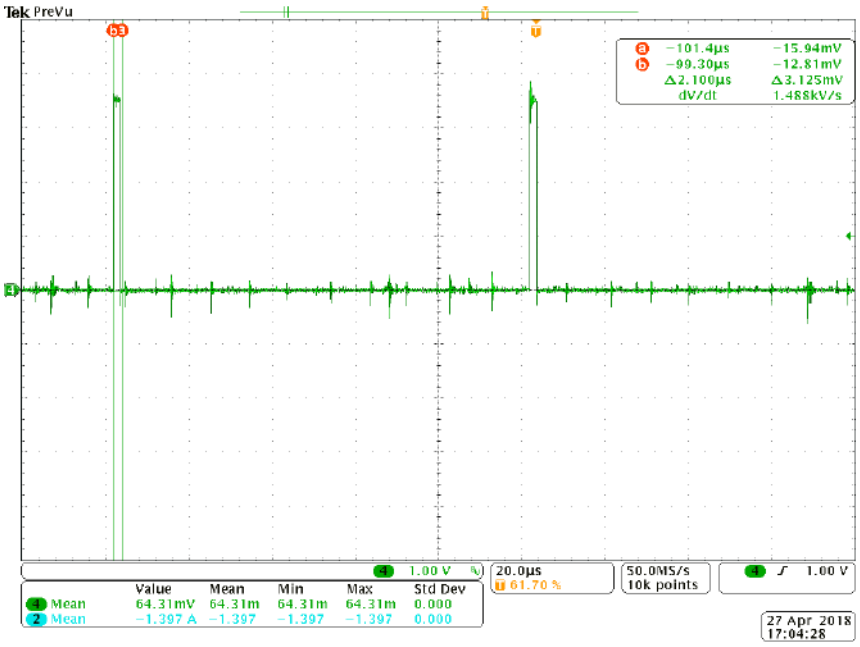


Comparativa experimental de tiempo de ejecución de lectura de datos de adquisición por FlexBus sin optimizaciones y con código optimizado.

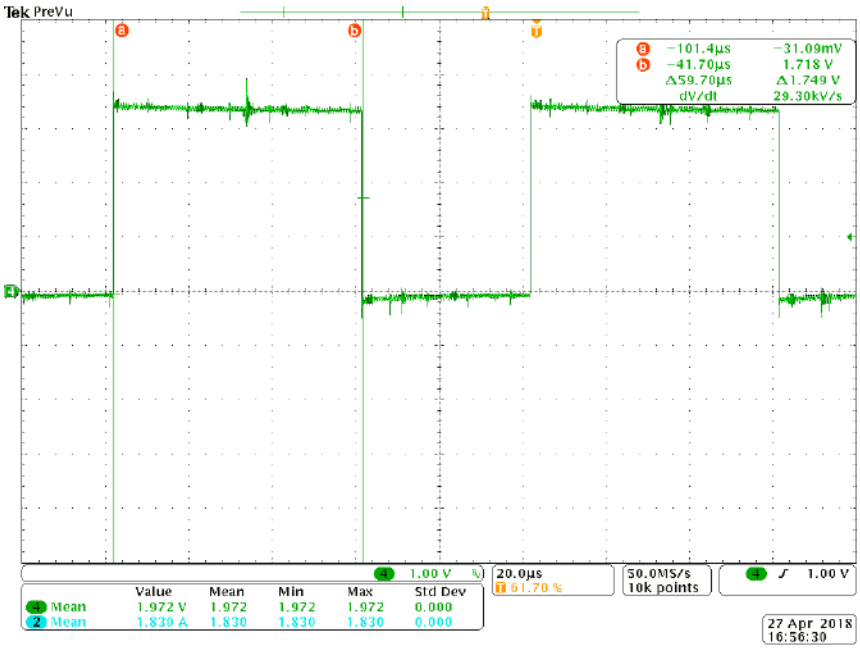
De estas mediciones obtenidas, se puede inferir que los tiempos totales de ejecución son de $37.8\ \mu s$ versus $20.1\ \mu s$ (sin optimizar versus optimizado) para el sistema en idle (solo adquiriendo) en el periodo de conmutación de $100\ \mu s$ del sistema.

Medición y comparativa de tiempos de ejecución en algoritmos de control:

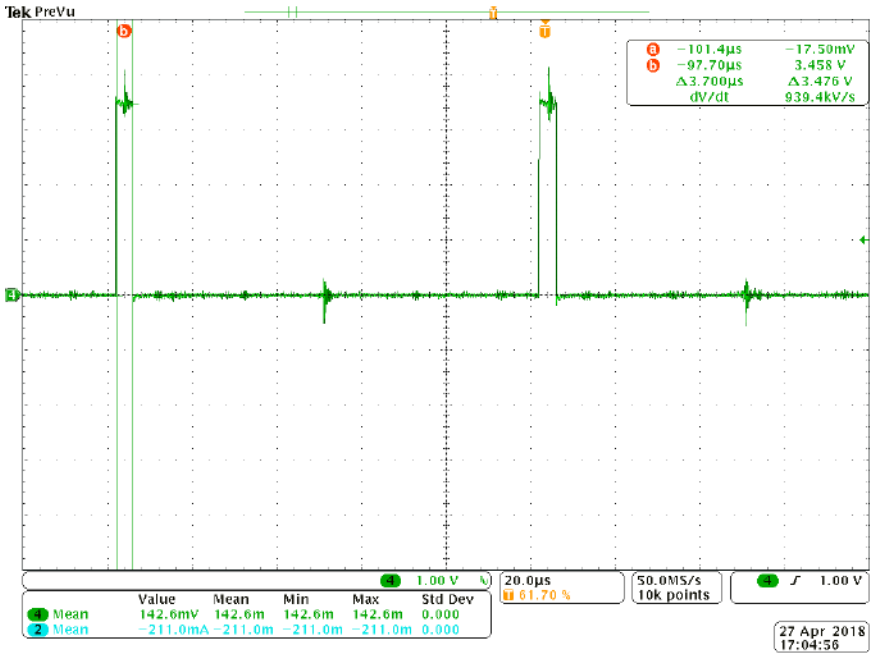
A continuación, se pueden apreciar capturas del osciloscopio tomando los tiempos de ejecución de los algoritmos de control implementados en diferentes condiciones tales como con y sin optimización, y con una cantidad mayor o menor de términos resonantes, lo cual se verá reflejado en un mayor o menor tiempo de ejecución.



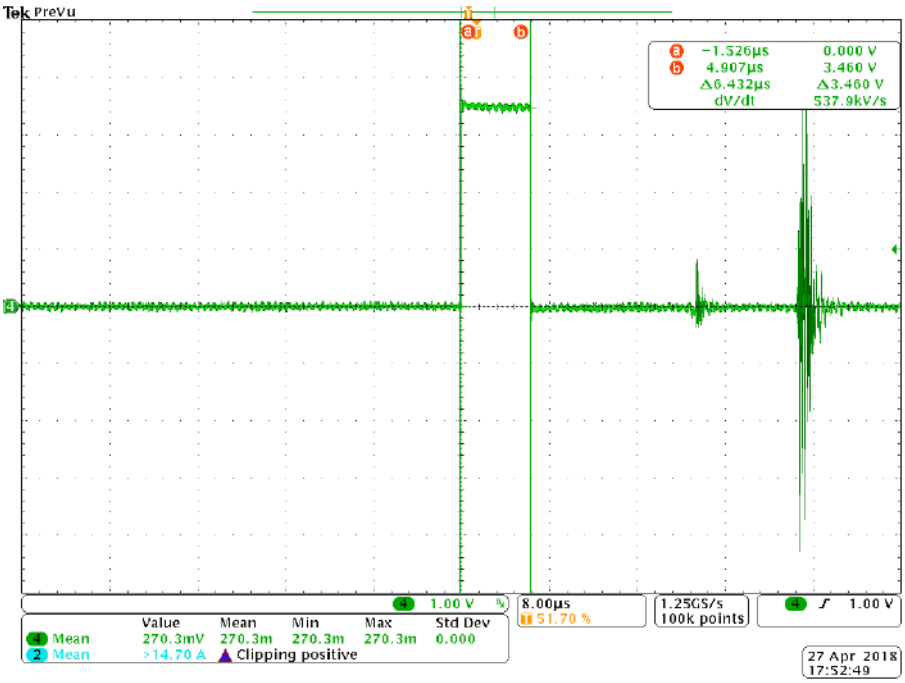
Medición de tiempo de ejecución de control RPCC optimizado.



Medición de tiempo de ejecución de control resonante RP-RPCC (15 términos) sin optimizar.



Medición de tiempo de ejecución de control RP-RPCC (8 términos) optimizado.



Medición de tiempo de ejecución de control RP-RPCC (15 términos) optimizado.

A su vez, aquí se puede observar que los tiempos de ejecución del algoritmo de control también dependen de su implementación, nivel de optimización y que tan computacionalmente intensivos sean.

En este sistema el control más demandante implementado actualmente, sin optimización consumiría alrededor de $60\ \mu s$ versus los $6.5\ \mu s$ de su implementación optimizada. De aquí surge que el trabajo sobre la optimización de las funciones es importante para la implementación de código intensivo si se cuenta con recursos limitados, y que de esta manera se logra implementar controles más sofisticados en un sistema determinado.

Además, al liberar tiempo de ejecución total del sistema, esto permite que se puedan implementar funcionalidades adicionales o estas puedan ser expandidas sin que existan problemas de realización.

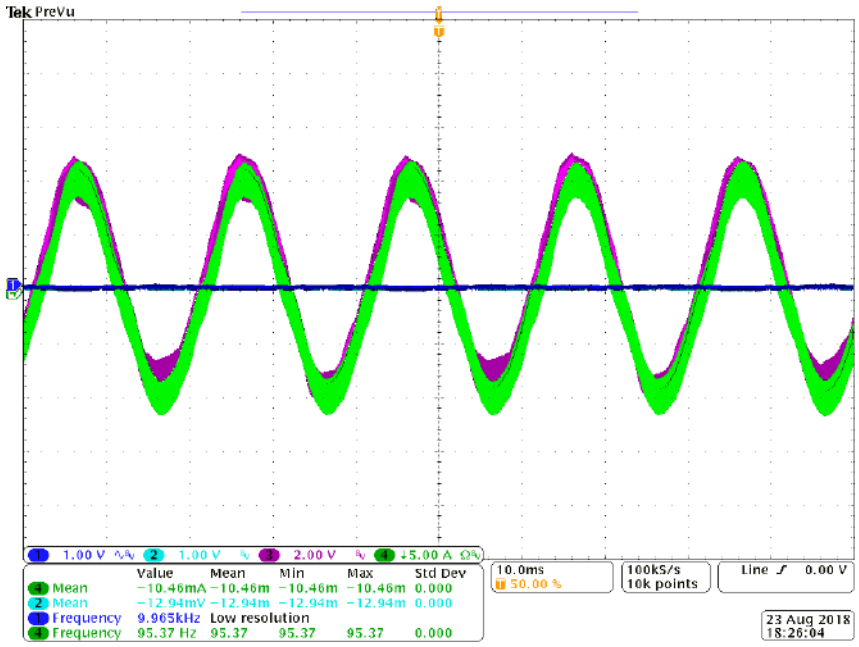
Testeo completo del sistema en condiciones de funcionamiento:

Por último, se testeó el proyecto completo en condiciones reales con carga, conectado a la red y comandando las llaves con alta tensión, y se verifico su correcto funcionamiento. Además se realizó un análisis de las formas de onda de salida inyectadas a la red y sus indicadores de performance.

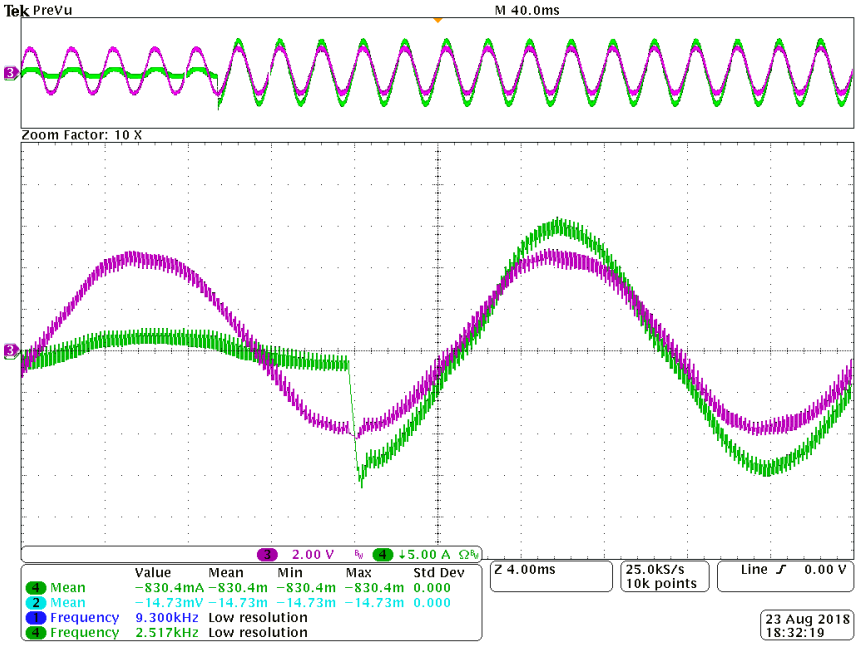
Sistema completo inyectando a la red $10\ KW$ junto a instrumentos de medición y CLI para comandar sistema de control.

Testeo de control RPCC inyectando a la red eléctrica:

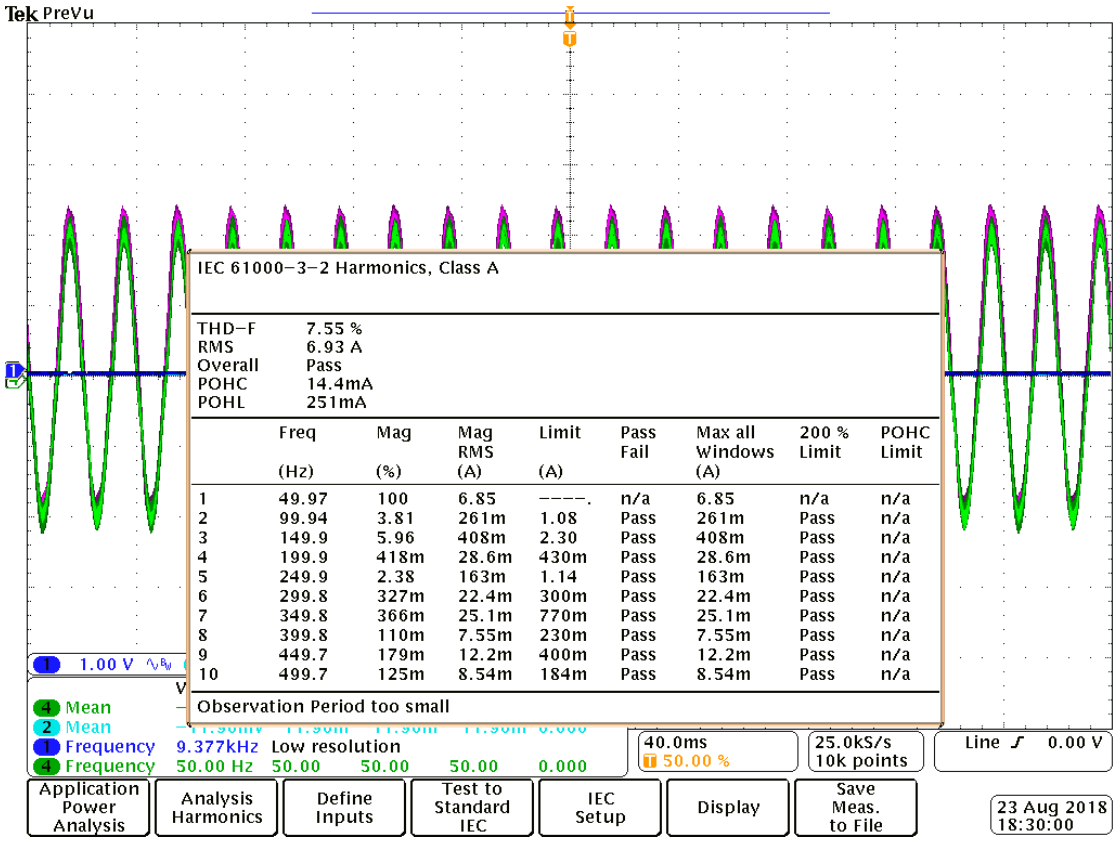
En las siguientes capturas de osciloscopio puede observarse la performance del algoritmo de control RPCC implementado, su forma de onda, respuesta a una variación de la referencia de corriente y su transitorio que se desea inyectar, y el análisis de THD correspondiente:



Captura de inyección de $10\ A$ de corriente a la red eléctrica con control RPCC.



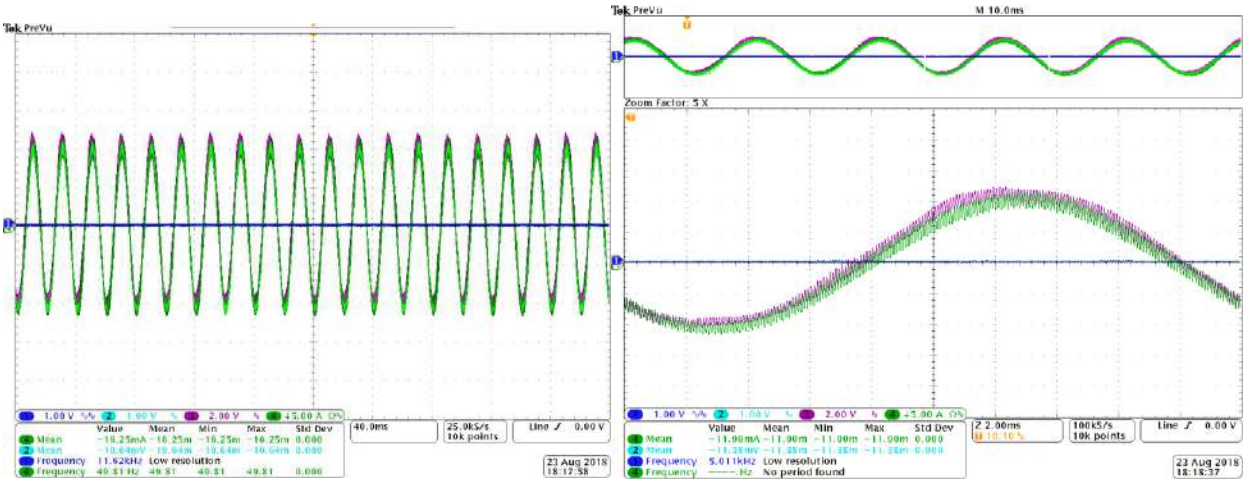
Respuesta transitoria en cambio de referencia $2\ a\ 10\ A$ de control RPCC.



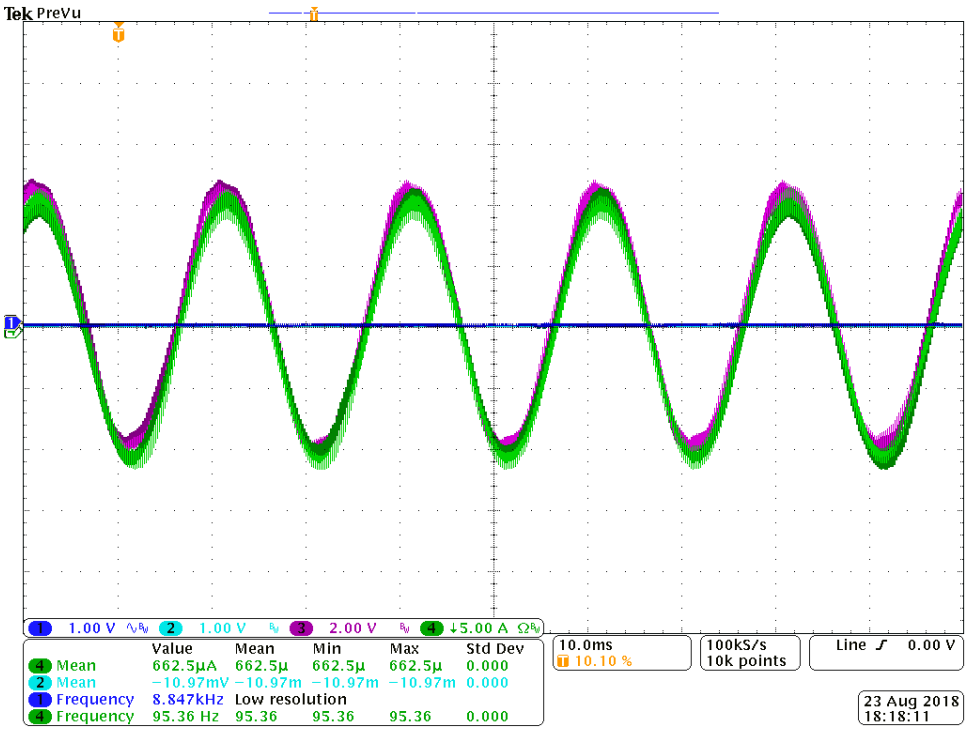
Análisis de THD de control RPCC generado por osciloscopio sobre corriente inyectada a la red (10 A).

Testeo de control con términos resonantes inyectando a la red eléctrica:

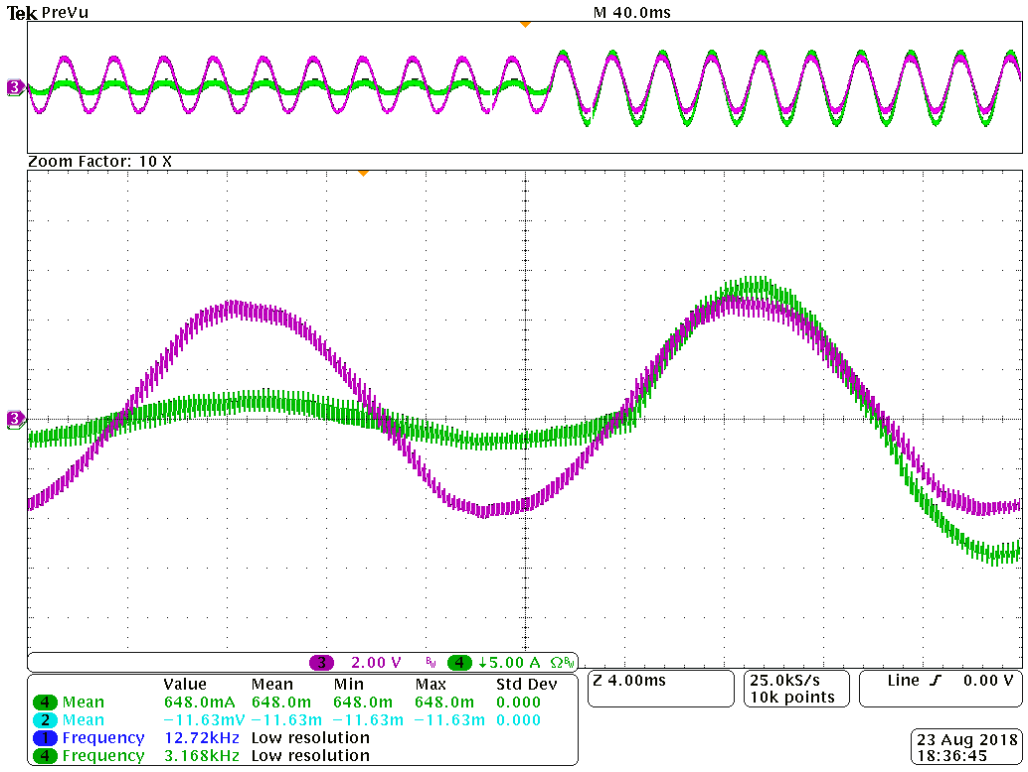
En las siguientes capturas de osciloscopio puede observarse la performance del algoritmo de control resonante implementado, comparativas con otro de menor cantidad de términos, su forma de onda, respuesta a una variación de la referencia de corriente y su transitorio que se desea inyectar, y el análisis de THD correspondiente:



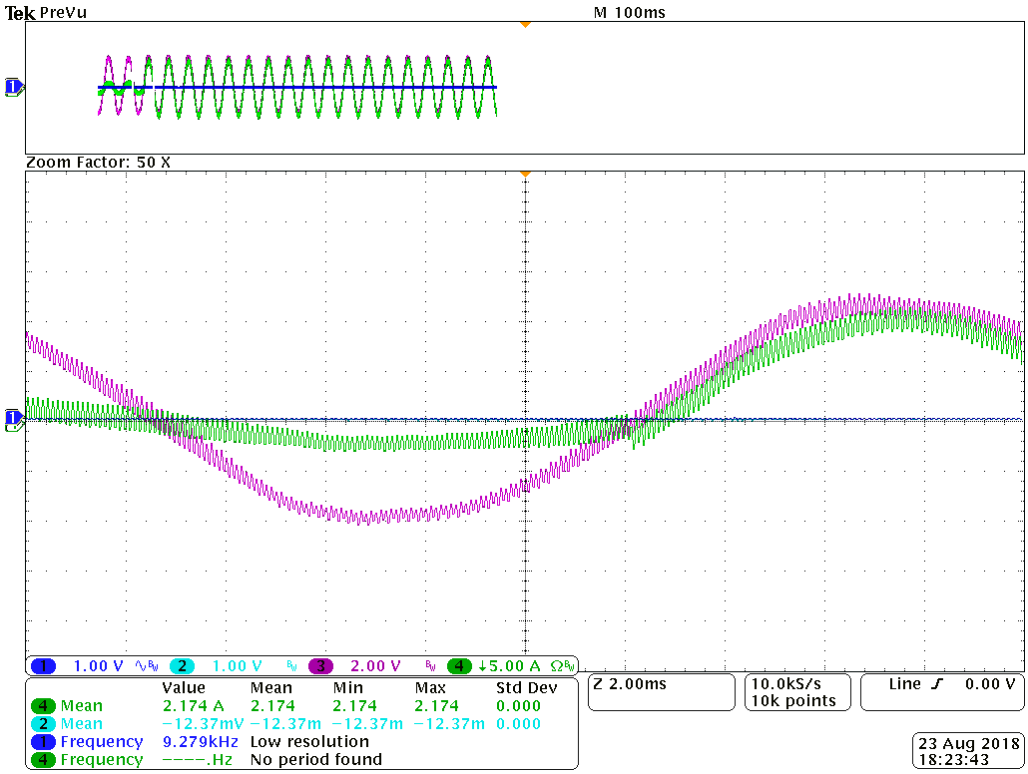
Forma de onda y ampliación de esta para inyección de 10 A a la red eléctrica con algoritmo de control resonante.



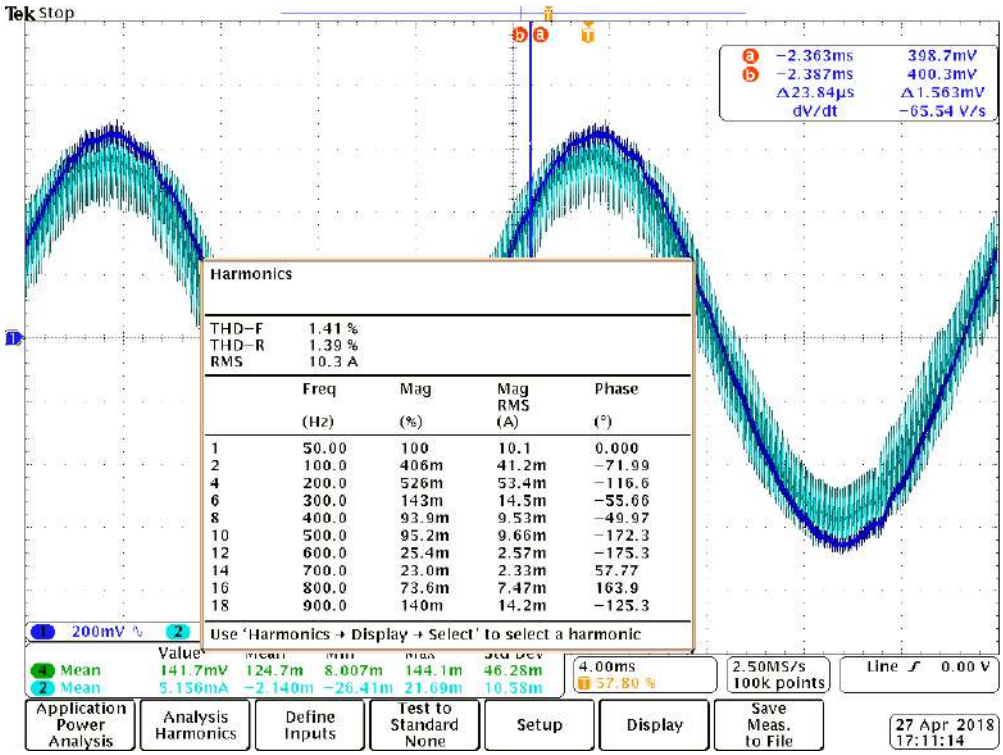
Forma de onda de inyección de 10 A a la red eléctrica con algoritmo de control RP-RPCC.



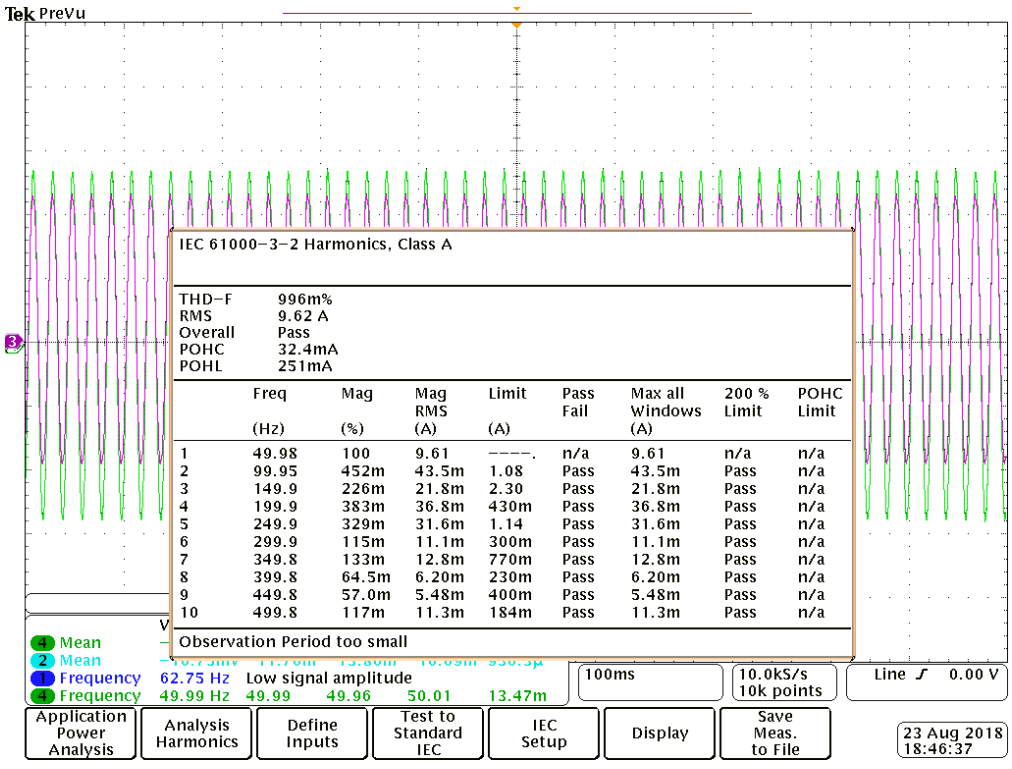
Respuesta transitoria de control resonante con cambio de referencia de $2\ a\ 10\ A$.



Ampliación de transitorio de $2\ a\ 10\ A$ con algoritmo de control RP-RPCC.



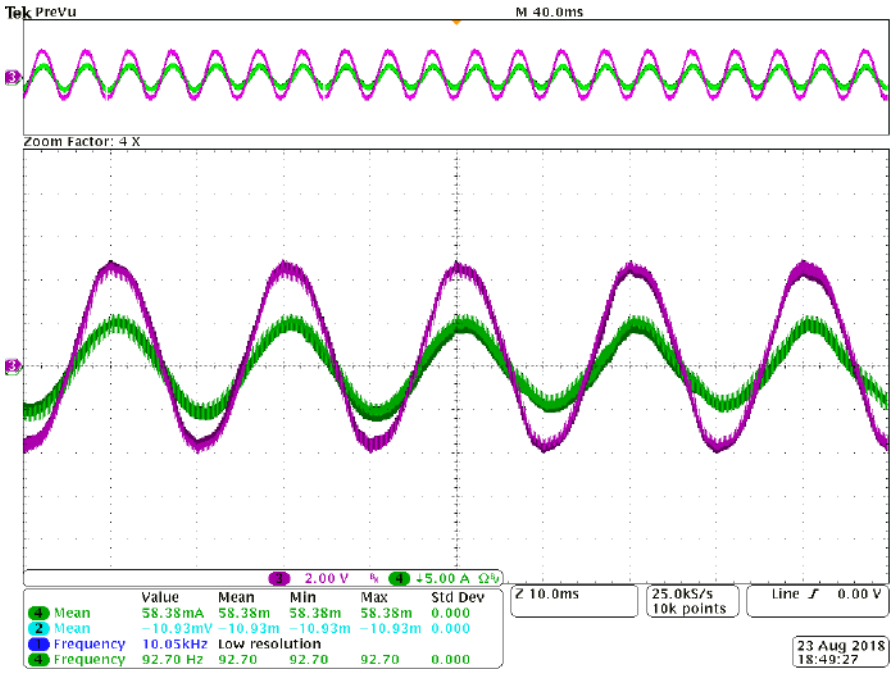
Análisis de THD de control resonante (8 términos) generado por osciloscopio sobre corriente inyectada a la red (10 A).



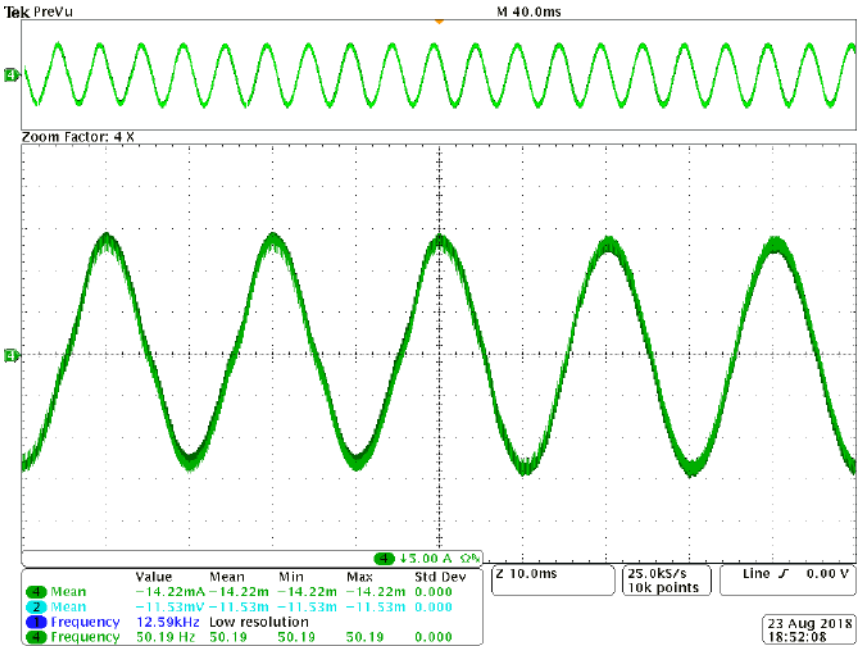
Análisis de THD de control RP-RPCC con una implementación posterior (15 términos) generado por osciloscopio sobre corriente inyectada a la red (10 A).

Selección y cambio “on the fly” entre algoritmos de control:

En las siguientes capturas del osciloscopio puede observarse el cambio entre algoritmos de control online vía CLI mientras estos funcionaban inyectando corriente a la red eléctrica:



Control RP-RPCC a control RPCC (5 A).



Control RPCC a RP-RPCC (13 A).

Conclusiones:

Se cumplieron los objetivos planteados por el proyecto final, tanto desde el punto de vista del laboratorio y los directores del proyecto, interesado en obtener una plataforma para el desarrollo y testeo de estrategias de control implementada en el panel de pruebas del LIC para realizar investigación, como así también desde el punto de vista del alumno, que integró y aplicó conocimientos previamente obtenidos a lo largo de la carrera junto a nuevas temáticas propias de este proyecto en sí, aplicadas para el desarrollo, la puesta en marcha y el testeo de este sistema.

Como se presentó anteriormente en este texto, el sistema presenta una plataforma de buena performance preparada para la investigación y desarrollo sobre aplicaciones de potencia, y la posibilidad de continuar expandiendo sus funcionalidades implementadas utilizando lo desarrollado de base para futuras mejoras y pruebas.

Entre los conocimientos aplicados a lo largo del desarrollo de este proyecto se puede destacar el aprendizaje o la profundización de este sobre un proyecto de envergadura considerable, que contiene muchas de las incumbencias de un ingeniero electrónico, como el desarrollo de PCBs, el control digital, los convertidores de potencia, el proceso de adquisición y procesamiento de señales, el trabajo sobre sistemas embebidos tanto en microcontroladores como en dispositivos de lógica programable, su desarrollo de hardware y firmware correspondiente, entre otros temas de gran interés para un profesional de esta carrera.

Por último, este proyecto sirvió la función de concluir mis estudios de grado para la obtención del título de ingeniero electrónico, profesión a la cual ya me dedico actualmente hace varios años y que me genera gran satisfacción personal y aporta un gran valor a la comunidad.

7. Apéndices:

Descripción de hardware de CPLDs **AnalogFrontEnd**

Descripción de hardware de PWMInterface en **DigitalControlBoard**

Extractos de código **DigitalControlBoard (ADC_EXT, CONTROL/INVERTER, CLI...)**:

Proyecto **BaseBoard** en Altium Designer

Proyecto **DigitalControlBoard** en Altium Designer

Proyecto **AnalogFrontEnd** en Altium Designer

Proyecto **AnalogFrontEndConn** en Altium Designer

Proyecto **InterfaceConn** en Altium Designer

8. Bibliografía y referencias:

Sistemas de Control en Tiempo Discreto – Katsuhiko Ogata

Fundamentals of Power Electronics – Robert W. Erickson, Dragan Maksimović

Aplicaciones Industriales de Electrónica de Potencia – Universidad Nacional de Mar del Plata, Laboratorio de Instrumentación y Control

Power Electronics Handbook – Muhammad H. Rashid

Robust Predictive Current Control with Harmonic Compensators for Grid-Connected VSI – Jonatan R. Fischer, Juan F. Martinez, Marcos G. Judewicz, Noelia I. Echeverría, Sergio A. Gonzalez, LIC, ICyTE, UNMdP/CONICET Mar del Plata

Datasheet IGBTs, SKM75GB176D:

http://www.gl-igbt.com/uploadfile/files/20130116152612867O.pdf

Applying IGBT and diode dies, Application Note 5SYA 2059-04:

https://search.abb.com/library/Download.aspx?DocumentID=5SYA2059&LanguageCode=en&DocumentPartId=&Action=Launch

Datasheet LM22678 Step-Down Voltage Regulator:

https://www.ti.com/lit/ds/snvs585m/snvs585m.pdf

Datasheet THS4524 Fully Differential Amplifier:

https://www.ti.com/lit/ds/symlink/ths4524-ep.pdf

Application Report, Fully-Differential Amplifiers:

https://www.ti.com/lit/an/sloa054e/sloa054e.pdf

FilterPro Desktop Texas Instruments:

https://www.ti.com/lit/an/slyt113/slyt113.pdf

Datasheet LEM LA 55-P/SP1 Current Transducer:

https://www.lem.com/sites/default/files/products_datasheets/la_55-p_sp1_e.pdf

Datasheet ADS8365 SAR ADS:

https://www.ti.com/lit/ds/symlink/ads8365.pdf

Tutorial of Succesive Approximation Registers (SAR) and Flash ADCs:

https://www.maximintegrated.com/en/design/technical-documents/tutorials/1/1080.html

Datasheet LM4120AIM5-3.0 Low Dropout Voltage Reference:

https://www.ti.com/lit/ds/symlink/lm4120.pdf

Datasheet XC2C64A CoolRunner-II CPLD:

https://www.xilinx.com/support/documentation/data_sheets/ds311.pdf

CoolRunner II CPLD Family:

https://www.xilinx.com/support/documentation/data_sheets/ds090.pdf

Free Range VHDL – Bryan Mealy, Fabrizio Tappero:

https://freerangefactory.org/

Datasheet TLV3502AID Rail-to-Rail, High-Speed Comparator:

https://www.ti.com/lit/ds/symlink/tlv3502.pdf

Datasheet TPS73201, Voltage Regulator:

https://www.ti.com/lit/ds/symlink/tps732.pdf

Reference Manual, K22 Microcontroller Sub-Family:

https://www.nxp.com/files-static/32bit/doc/ref_manual/K22P144M120SF5RM.pdf

Datasheet CD4504B CMOS Voltage-Level Shifter:

https://www.ti.com/lit/ds/symlink/cd4504b.pdf

ISE, entorno para síntesis y análisis de diseños HDL:

https://www.xilinx.com/products/design-tools/ise-design-suite.html

SEGGER J-LINK, debugger JTAG/SWD:

https://www.segger.com/products/debug-probes/j-link/models/j-link-base/

Altium Designer, EDA para desarrollo de PCBs:

https://www.altium.com/altium-designer/

Eclipse Mars.2, entorno de desarrollo para C/C++:

https://www.eclipse.org/

Eclipse Embedded CDT, plug-ins para desarrollo en eclipse de ARM:

https://gnu-mcu-eclipse.github.io/

Git, control de versiones distribuido:

https://git-scm.com/

BitBucket, donde se realizó el control de versiones y el desarrollo colaborativo:

https://bitbucket.org/

The C Programming Language – Brian W. Kernighan, Dennis M. Ritchie

FreeRTOS, sistema operativo de tiempo real open source:

https://www.freertos.org/

FreeRTOS Reference manual, API Functions and Configuration Options:

https://www.freertos.org/fr-content-src/uploads/2018/07/FreeRTOS_Reference_Manual_V10.0.0.pdf

FSM whitepaper, implementación de código con estructura de máquina de estados:

https://www2.keil.com/docs/default-source/default-document-library/software-based-finite-state-machine-(fsm)-with-general-purpose-processorsf4f837f788736c26abc1ff00001d2c02.pdf?sfvrsn=0

FlexBus for Kinetis Microcontrollers, application note:

https://www.nxp.com/docs/en/application-note/AN4393.pdf

The Scientist and Engineer's Guide to Digital Signal Processing, Moving Average Filters:

https://www.dspguide.com/ch15.htm

Implementation of a novel synchronization method using sliding Goertzel DFT:

https://www.researchgate.net/publication/224305580_Implementation_of_a_novel_synchronization_method_using_Sliding_Goertzel_DFT

NL5, Circuit Simulator:

https://sidelinesoft.com/nl5/index.php?page=index&lang=es

Bit-banding, ARM Cortex-M4 Reference Manual:

https://developer.arm.com/documentation/ddi0439/b/Programmers-Model/Bit-banding

Using the GNU Compiler Collection (GCC), options that control optimization:

https://gcc.gnu.org/onlinedocs/gcc/Optimize-Options.html