OSCILOSCOPIO DIGITAL CON PANTALLA TÁCTIL

Tesis de Grado en Ingeniería Electrónica

Tomás Migone - 49002

Alejandro Luebs - 49081

Javier Velazquez Traut - 49104

Pablo Carranza Vélez - 49270

Profesores de Tesis: Ing. Claudio Marcelo Muñoz
Ing. Ricardo Alejandro Pingitore

AGRADECIMIENTOS

Este proyecto fue un gran desafío, y para enfrentarlo fue fundamental la ayuda que recibimos de distintas personas. Por eso, queremos agradecer sus colaboraciones:

A Roxana Saint-Nom, que desde su actividad pedagógica nos inculcó la mentalidad de que no hay metas que no se puedan alcanzar.

A Daniel Jacoby y Pablo Cosutta, que con su aporte de conocimientos y experiencia nos ayudaron a encontrar soluciones a algunas de las incógnitas más complejas del desarrollo.

A Jorge Cáceres, cuya ayuda fue indispensable en el diseño mecánico y el ensamblaje de los circuitos impresos.

A nuestro compañero Juan Matus, que siempre nos brindó consejos prácticos de implementación.

A nuestros familiares y amigos, por el apoyo durante estos intensos años de carrera.

RESUMEN

La presente tesis consiste en el proceso de diseño de un osciloscopio digital con pantalla táctil, desde su concepción hasta la validación de un prototipo funcional.

Se busca llegar a un producto capaz de competir en el mercado y satisfacer las necesidades de los usuarios. Para esto, en primera instancia se analiza dichas necesidades mediante una encuesta, a partir de la cual se identifican los requerimientos y, con la técnica de la Casa de Calidad, se definen las especificaciones.

Posteriormente se estudia la factibilidad del producto y luego se procede a su ingeniería. Para el diseño, se atraviesa campos variados de la ingeniería electrónica, pues entran en juego el diseño analógico y digital, el tratamiento de señales en tiempo real, la programación de FPGA, el desarrollo sobre el *kernel* de Linux y la programación y diseño de una interfaz gráfica. La conjunción de todas estas técnicas y actividades presenta, sin duda, un gran desafío.

La ingeniería se aplica a la construcción de un prototipo en el que se integran todas las áreas mencionadas, y culmina con la validación y análisis del funcionamiento de dicho prototipo, donde se estudia la calidad y confiabilidad del producto y se evalúa el resultado del proceso de diseño.

ÍNDICE

ÍNDICE 4
I. INTRODUCCIÓN 8
1.1 Reseña Histórica 8
1.2 Glosario de Términos 11
1.3 Justificación del proyecto 13
II. OBJETIVOS 16
2.1 Finalidad del Proyecto 16
2.2 Planteamiento del Problema a Resolver 16
III. DEFINICIÓN DE PRODUCTO 18
3.1 Requerimientos 18
3.1.1 Construcción de la Casa de Calidad 19
3.2 Especificaciones 22
3.2.1 Especificaciones de Hardware 22
3.2.2 Especificaciones de Software 24
IV. ANÁLISIS DE FACTIBILIDAD 26
4.1 Factibilidad Tecnológica 26
4.1.1 Propuesta de alternativa de diseño 26
4.1.2 Elección de una solución 31
4.1.3 DFMEA 33
4.2 Factibilidad de Tiempos 34
4.2.1 Planificación - Pert y Montecarlo 34
4.3 Factibilidad Legal y Responsabilidad Civil 41
4.4 Factibilidad Económica 42
4.4.1 Análisis del mercado 42
4.4.2 Canales comerciales 43

4.4.3 Marketing 44
4.4.4 Presupuesto de marketing y ventas 44
4.4.5 Organización de la producción 45
4.4.6 Costos Fijos 45
4.4.7 Costo Inicial 46
4.4.8 Costo Variable 46
V. INGENIERÍA DE HARDWARE 50
5.1 Circuito de Adaptación de Señal 51
5.1.1 Impedancia de entrada y acoplamiento 52
5.1.2 Generación de offset 54
5.1.3 Resta de offset 56
5.1.4 Ganancia 57
5.1.5 Acondicionamiento para ADC 58
5.1.6 Alimentación 59
5.1.7 Plan de Pruebas 60
5.2 ADC 60
5.3 FPGA 63
5.3.1 Bloque de ADC Control 64
5.3.2 Bloque de Acondicionamiento 64
5.3.3 Bloque de *Trigger* 65
5.3.4 Bloque de Almacenamiento 67
5.3.5 Bloque de Comunicación y Control 69
5.3.6 Plan de Pruebas 71
5.4 Microprocesador 72
5.4.1 Plan de Pruebas 75
5.5 Interfaz LCD y touch 77
5.5.1 Plan de pruebas 78
VI. INGENIERÍA DE SOFTWARE 79
6.1 Kernel Linux 81
6.1.1 Definición de arquitectura 82
6.1.2 Bus I2C – Driver de touch 86
6.1.3 Bus SPI 87
6.1.4 Display LCD 88
6.1.5 Driver de FPGA – “Wahoo” 89
6.2 Aplicación de usuario: *Softscope* 92

6.2.1 *Softscope*: diseño e implementación 92
6.2.2 *Softscope*: Interfaz gráfica 96
6.2.3 *Softscope*: Plan de prueba y depuración 100
VII. CONSTRUCCIÓN DEL PROTOTIPO 102
7.1 Definición de los módulos 102
7.2 Diseño de los circuitos impresos 106
7.2.1 KrakenBoard 106
7.2.2 LobsterBoard 108
7.2.3 HCI Board 111
7.3 Diseño mecánico 113
VIII. VALIDACIÓN DEL PROTOTIPO 117
8.1 Validación del sistema 117
8.2 Plan y protocolos especiales de medición 117
8.3 Mediciones 119
8.4 Análisis de los resultados 121
IX. ESTUDIOS DE CONFIABILIDAD 123
9.1 Estudios de confiabilidad de Hardware 123
9.1.1 Resistencias 123
9.1.2 Capacitores 124
9.1.3 Inductores 125
9.1.4 Diodos 126
9.1.5 Circuitos Integrados 126
9.1.6 Conectores BNC 127
9.1.7 Conectores Pines 128
9.1.8 Sistema completo 129
9.2. Confiabilidad de software 130
X. CONCLUSIONES 132
XI. ANEXO 136
A. Código de software 136

B. Adaptación de impedancias en comunicación ADC-FPGA 136

C. Cálculo de los elementos de la placa de entrada 139

D. Listado de partes 140

E. Conexiones entre placas 142

XII. BIBLIOGRAFÍA 146

I. INTRODUCCIÓN

El osciloscopio es una herramienta tecnológica moderna fundamental para el análisis de eventos temporales manifestados en formas de onda. Una gran variedad de fenómenos físicos se manifiestan mediante ondas y oscilaciones. Desde el movimiento oceánico hasta una explosión, pasando por la luz y el sonido; la naturaleza misma se compone en gran medida de distintas ondas. Mediante la utilización de una vasta cantidad de sensores electrónicos, el osciloscopio permite a los científicos e ingenieros modernos el estudio de las formas de onda más relevantes para la humanidad. A su vez, la tecnología electrónica se ha introducido fuertemente en la vida cotidiana de las personas. Por lo tanto, se hace indispensable para un ingeniero que pretende realizar mantenimiento o desarrollo de nueva tecnología contar con un osciloscopio acorde a sus necesidades. Es por esto que la investigación e innovación aplicada a este instrumento de medición es motivo de gran entusiasmo por grandes empresas de tecnología en todo el mundo.

1.1 Reseña Histórica

Un osciloscopio es un instrumento de medición electrónico que permite observar la forma de onda de señales eléctricas a lo largo del tiempo. La información se presenta por lo general en un gráfico de dos dimensiones donde el eje horizontal representa al tiempo y el vertical la diferencia de potencial eléctrico observada.

Tradicionalmente, el osciloscopio surgió como una herramienta para analizar características temporales básicas de una señal como su amplitud, frecuencia y forma de onda. La aparición de nuevas tecnologías, en particular la posibilidad de realizar un procesamiento digital de la información en tiempo real, permitió incorporar nuevas funcionalidades al osciloscopio. Un equipo moderno cuenta con una gran variedad de funciones de medición avanzadas tanto en el dominio del tiempo como en el de la frecuencia. Ejemplos de ello son la posibilidad de medir el *rise time* de una señal cuadrada o realizar un análisis espectral utilizando una FFT.

El osciloscopio es utilizado en diversos campos desde la industria de las telecomunicaciones hasta la medicina. Es muy utilizado en investigación y trabajo de laboratorio; sin embargo, su aplicación más importante es en el desarrollo y mantenimiento de equipos electrónicos. Esto hace que la utilización de un osciloscopio sea necesaria para el desarrollo de prácticamente cualquier producto moderno más allá de su campo de aplicación.

Los primeros métodos para generar la imagen de una forma de onda fueron basados en la utilización de galvanómetros. Los primeros "oscilogramas" requerían de intervención humana para su generación, un técnico especializado tomaba nota de las mediciones realizadas por el galvanómetro para luego obtener la imágen correspondiente. Un prototipo de Joules Joubert en la década de 1880 introdujo un mecanismo semi-automático que facilitaba la tarea; usando un mecanismo de rotación paso a paso se lograba una escala ajustable con la cual el técnico podía tomar las mediciones a intervalos regulares. Años después se comenzó a utilizar el galvanómetro para mover un lápiz a lo largo de un rollo de papel que avanzaba continuamente. Estos equipos contaban con un grave problema: debido a que estaban basados en componentes mecánicos el tiempo de reacción del sistema era relativamente lento y no era capaz de seguir señales rápidas (altas frecuencias). Esto implica que la forma de onda resultante en las imágenes para estos casos no era la original sino una forma de onda promedio. La siguiente mejora sería el llamado "oscilograma de bobina móvil" por parte de William Duddell. Este prototipo subsanó la problemática de los dispositivos con galvanómetro reemplazándolo por un galvanómetro especial que utilizaba, en lugar de una aguja, un espejo móvil para reflejar un haz de luz. La forma de onda de la señal de interés se registraba tomando fotos del haz o simplemente filmando su progresión. A pesar de que la solución aún era del tipo mecánica, el tiempo de respuesta de este tipo de galvanómetros era considerablemente mejor por lo que mejoró la precisión del instrumento. No fue sino hasta la aparición de los osciloscopios basados en tubos de rayos catódicos (CRT) que este tipo de instrumentos comenzaron a ser de gran utilidad. A mediados de 1920 los monitores CRT comenzaron a ser de uso comercial; sin embargo, su irrupción como tecnología predominante en el mercado de los osciloscopios se debió a la invención del sistema de disparo (*trigger*). En 1946, Howard Vollum y Jack Murdock (futuros fundadores de Tektronix, empresa líder en manufactura de osciloscopios) desarrollaron los primeros osciloscopios CRT con sistema de *trigger*. La tecnología CRT permitía que el instrumento responda adecuadamente frente a señales de rápida variación, mientras que el *triggering* permitía ver las formas de onda de manera estacionaria y legible en la pantalla (a pesar de ser de rápida variación). Esto sentó las bases de lo que sería toda la tecnología de osciloscopios analógicos.

El siguiente avance significativo fue la incorporación de la electrónica digital en osciloscopios. A partir de la década de 1980 comenzaron a surgir los primeros osciloscopios digitales. Su principio de funcionamiento es similar al de los analógicos. Los DSO (por *Digital Storage Oscilloscope*) utilizan un conversor analógico a digital en conjunto con memorias para almacenar una representación digital de la forma de onda de la señal de entrada. Esto permite una mayor

flexibilidad a la hora del análisis y presentación de la información. Una gran ventaja que presentan los equipos digitales es que permiten observar eventos que suceden antes del *triggereo*, dado que se almacenan todas las muestras de forma continua. Esto permite analizar eventos intermitentes u esporádicos. A continuación (fig. 1.1) se muestra un esquema genérico de un DSO:

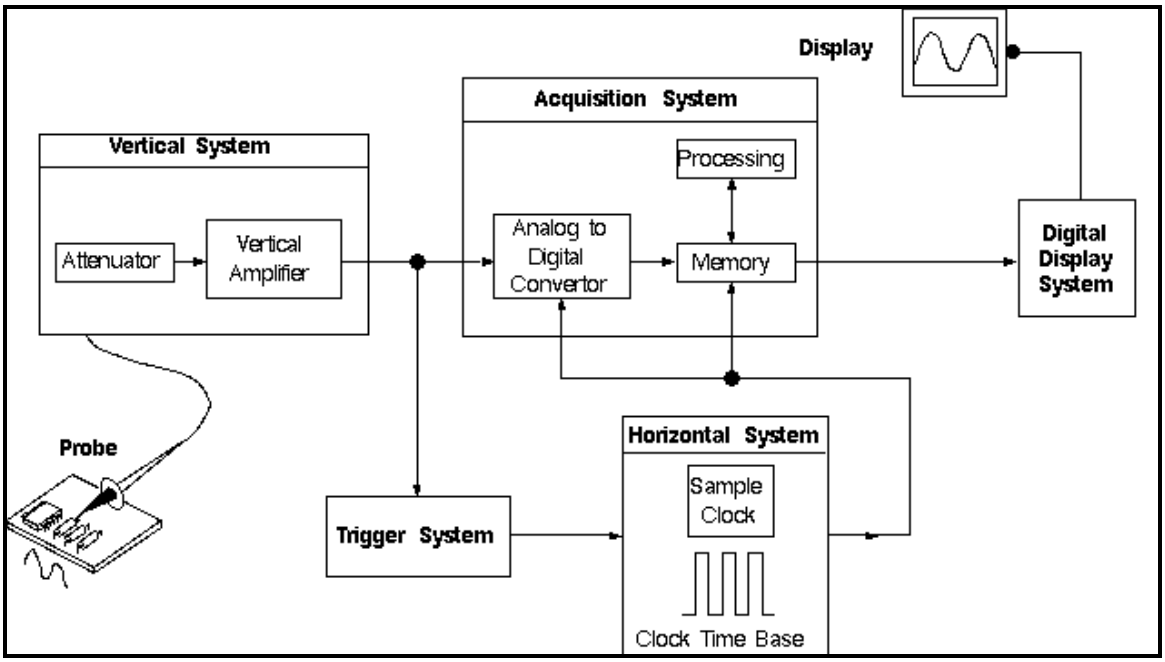


Fig. 1.1: Esquéma genérico de un Digital Storage Oscilloscope[1]

El sistema consiste de tres bloques fundamentales: Sistema Vertical, Sistema Horizontal y Sistema de Adquisición. El Sistema Vertical tiene como función adecuar la señal de entrada de acuerdo a los requerimientos del Sistema de Adquisición; en líneas generales, se filtra la señal y se le da una ganancia. Señales de muy pequeña amplitud serán amplificadas mientras que señales de gran amplitud son atenuadas, permitiendo un mayor rango de operación en cuanto a la tensión de entrada. Controla el eje vertical de la pantalla del osciloscopio. El Sistema de Adquisición consiste en un conversor analógico a digital, memoria y una unidad de procesamiento. En esta etapa es donde se puede realizar cualquier tipo de tratamiento a la señal. El conversor AD es

[1] Elements of Electronic Instrumentation and Measurements - David A. Bell, 2º Edición, Prentice Hall, 1994.

controlado por una señal proveniente del Sistema Horizontal que le indica en que momentos debe tomar una nueva muestra. El Sistema Horizontal controla el eje horizontal de la pantalla del osciloscopio. A su vez, el Sistema Horizontal es controlado por el sistema de *trigger* que se encarga de mantener sincronizado al Sistema Horizontal con la señal para obtener una imagen estable en la pantalla. Por último, los datos son mostrados en una pantalla digital.

A pesar de que hoy en día los osciloscopios son de uso masivo a lo largo del mundo, hay ocasiones en las cuales no resulta la herramienta más cómoda para trabajar. Por ejemplo, si se está trabajando en una planta industrial y se debe reparar maquinaria de gran tamaño un osciloscopio tradicional podría no ser la solución más cómoda. Situaciones como ésta en las cuales no es posible llevar el objeto de la medición a un laboratorio derivan en la necesidad de contar con instrumentos portátiles. En la última década, impulsados por el creciente avance tecnológico, fueron surgiendo los primeros modelos de osciloscopios portátiles.

1.2 Glosario de Términos

4-5 wire: Tecnología de pantalla táctil cuya salida involucra un conector con 4 - 5 cables.

amplificador de ganancia programable: Amplificador cuya ganancia puede ser controlada mediante a señales digitales.

ancho de banda: Rango de frecuencias de operación de un sistema.

application notes: Documentación sobre casos de aplicación concretos para componentes electrónicos provista por el fabricante.

Assembly: Lenguaje de programación de bajo nivel.

backlight: Panel de alimentación de una pantalla LCD. Regula la intensidad de brillo.

bootlets: Pequeños programas que conforman el bootloader.

bootloader: Programa que se encarga de cargar el sistema operativo en memoria para su ejecución.

buffer: Unidad de almacenamiento temporario en memoria.

buildroot: Herramienta open source utilizada para generar un kernel y sistema de archivos a medida.

buses: Denominación para las conexiones entre conectores o bloques de una arquitectura.

CMOS: Tecnología de fabricación de transistores.

conversor AD: Circuito integrado que traduce señales eléctricas del dominio analógico al digital.

D/A: Circuito integrado que traduce señales eléctricas del dominio digital al analógico.

Decimación: Reducción de la frecuencia de muestreo.

drivers: Programa que controla un determinado periférico o porcion de hardware.

DSP: micro controlador optimizado para el procesamiento de señales digitales.

duty-cycle: Porcentaje del período de una señal donde el valor de la misma se encuentra por encima de su valor medio.

DXP Altium Protel: Programa utilizado para el desarrollo de circuitos impresos.

encapsulados BGA: Packaging estándar de un circuito integrado.

FPGA: Circuito integrado reconfigurable al cual se le programan las conexiones entre sus componentes (compuertas lógicas, memoria, etc).

frecuencia de corte: Frecuencia para la cual la amplitud de salida de un sistema es aproximadamente el 70% de la amplitud de entrada.

front-end : Sección de un programa visible al usuario.

fuentes de switching: Fuente de tensión regulada por transistores en conmutación.

glitches: Cambio de estado espurio en el valor de un bit o una señal digital.

header: Denominación que se le da a la agrupación de varios pines/conectores de un PCB.

high-end: Dicese de un equipo de alta gama.

Hold-Off: Tiempo mínimo que ocurre entre la detección de dos eventos de disparo de trigger en el osciloscopio.

Impedancia de entrada: Impedancia equivalente que se observa desde la entrada de un dispositivo.

in-house: Dicese de un desarrollo o equipo propio manufacturado manualmente.

kernel: Programa central en un sistema operativo. Administra los pedidos del software sobre los dispositivos I/O y otros recursos.

layout: Disposición de los componentes y conexiones en un PCB.

LCD: Pantalla de cristál líquido.

LED: Dispositivo semiconductor emisor de luz.

Matlab: Programa de simulación matemática basado en operaciones matriciales.

mid-end: Dicese de un equipo de gama media.

Montecarlo: Herramienta matemática utilizada en administración y planificación de proyectos.

Msps: Cantidad de muestras por segundo expresadas en millones.

multitouch: Capacidad de un panel táctil de reconocer multiples puntos de contacto en simultaneo.

OLED: Dispositivo orgánico emisor de luz.

open source: Es una filosofía que promueve el acceso libre y universal al diseño de un producto.

PCB: Circuito impreso sobre una placa típicamente de cobre.

Pert: Herramienta estadística utilizada en administración y planificación de proyectos.

pico a pico: Diferencia de tensión entre el valor máximo y mínimo de una señal.

precio FOB: Precio de un equipo considerando todos los impuestos hasta la aduana del país de salida.

PWM: Técnica en la cual se controla el ciclo de trabajo de una señal periódica para controlar la cantidad de energía que se transfiere.

QFP: Packaging estándar de un circuito integrado de fácil manipulacion.

Quartus II: Programa para desarrollo en FPGA.

RAM: Memoria volátil utilizada para almacenamiento de datos temporales.

rango dinámico: Margen que existe en un sistema electrónico entre el valor de tensión máximo admitido y el nivel de ruido presente.

rechazo al modo común: Medida del nivel de la salida de un amplificador operacional cuando sus entradas son iguales.

relays: Llave magnética operada eléctricamente.

resistencia de encendido: Resistencia que presenta una llave en su estado encendido. Idealmente debe ser cero.

resistencia de pull-up: Resistencia utilizada para forzar un valor de encendido sobre una pata de un circuito integrado.

ripple: Oscilaciones indeseadas en una señal de continua.

rise time: Tiempo que tarda una señal eléctrica en alcanzar el 90% de su valor partiendo desde el 10%.

RMS: Tensión de continua equivalente que disipa la misma cantidad de potencia.

RoHS: Directiva de la Unión Europea para restringir el uso de ciertas sustancias contaminantes o peligrosas en la fabricación de equipos electrónicos.

SPI: Interface sincrónica de comunicacion serie utilizada extensivamente en desarrollos con microcontroladores.

tensión media: Valor medio de la tensión de una señal.

TFT: Tecnología de fabricación de pantallas LCD.

trigger: sistema de sincronización temporal para osciloscopios basado en cambios en la señal de entrada

uC: Abreviación para micro controlador.

1.3 Justificación del proyecto

Para satisfacer el gran aumento del desarrollo y mantenimiento tecnológico requerido en la actualidad, la cantidad de técnicos, ingenieros y científicos que requieren utilizar un osciloscopio se encuentra en constante crecimiento. Así como aumenta la cantidad de usuarios, se diversifican las áreas de aplicación y la situación de medición en la que se utiliza el osciloscopio. El proyecto pretende desarrollar un Osciloscopio Digital Portátil para satisfacer las necesidades de un técnico, ingeniero o científico que no sólo tenga exigencias respecto al tamaño o facilidad de uso, sino que también esté interesado en pagar un precio acorde a las prestaciones que se le brindan. A su vez, el mercado interesado en un producto como el propuesto tiene una gran proyección fuera de las áreas académica y profesional, dado que la cantidad de *hobbistas* que se introducen en el mundo tecnológico crece día a día.

Se pretende que el producto final tenga un tamaño que responda a la necesidad de portabilidad tal y como lo hacen los artefactos tecnológicos modernos que las personas suelen llevar consigo. En un principio, se podría pensar que dicha

prestación no es de relevancia, siendo que no se espera que un osciloscopio sea transportado junto a su operador constantemente. Sin embargo, dicha posibilidad es de gran interés para profesionales que deben realizar mediciones fuera de un laboratorio. Los técnicos que reparan instrumental electrónico, por ejemplo, a la hora de realizar trabajos de campo o reparaciones a domicilio, sacarían gran provecho de la portabilidad del osciloscopio. Incluso en el ámbito universitario, disponer de un instrumento de medición que se pueda trasladar a distintas locaciones de un laboratorio sería de gran utilidad. Esta prestación modifica el paradigma al que nos debemos enfrentar cuando deseamos analizar, diseñar o reparar un instrumental electrónico; en lugar de ser el artefacto quien deba ser trasladado al laboratorio para ser analizado (lo que necesariamente modifica las condiciones para dicho artefacto y por lo tanto los resultados de la medición), será el instrumental de medición el que se traslade hasta el lugar necesario. Un ejemplo en el que se hace indispensable la portabilidad del equipo surge de la situación en la que, a través de un sensor electrónico, se requiera medir una magnitud no eléctrica cuya preservación sólo se garantiza si la medición es realizada en un lugar específico.

La portabilidad del dispositivo no sólo se garantiza con peso y dimensiones reducidas, sino que además es esencial que el equipo posea una autonomía que tenga en cuenta la situación de uso más común del mismo. Para ello, el equipo se complementará con una batería recargable cuya duración permita que el usuario no tenga que interrumpir sus mediciones. Cabe destacar que dicha variable de duración infiere fuertemente en los precios de las baterías y éstas suelen ser uno de los elementos de mayor costo para la manufactura de un producto electrónico.

El elemento más destacado e innovador del osciloscopio que se diseñará es la interfaz de usuario utilizando una pantalla táctil. El uso de este tipo de interactividad entre el usuario y el equipo se encuentra en constante crecimiento y las prestaciones que brinda dicha tecnología se han ampliado en gran medida. Por otro lado, los costos que implica incorporar una pantalla táctil se han reducido fuertemente hasta el punto en que se hace viable competir en precio con un producto de interfaz tradicional. Por este motivo, se considera que la incorporación de un interfaz táctil como principal método de interacción será efectiva y le dará un valor agregado al producto que será bien valorizado por los interesados en el mismo.
Este producto podría ocupar un nicho del mercado bien determinado pero amplio. Este mercado es estable y con certezas de crecimiento dado que el osciloscopio se utiliza como instrumento de medición de artefactos tecnológicos desde el principio de la electrónica y se espera que siga cumpliendo ese rol. Tal y como se verá más adelante, la tecnología utilizada en un producto como el

propuesto permite diseñar un prototipo en un laboratorio universitario con costos similares a los de un producto final de producción masiva. De esta manera, se podrá comparar al Osciloscopio Digital Portátil con respecto a la competencia y se podrá realizar un análisis del impacto que tendría dicho producto en el mercado actual.

II. OBJETIVOS

2.1 Finalidad del Proyecto

El proyecto tiene como finalidad lograr un osciloscopio digital portátil con pantalla táctil que sea competente en el mercado y que tenga potencial de producción en grandes cantidades para su venta y comercialización. Se desea lograr un producto que sea sencillo de manipular por parte del usuario, con fácil conexionado, posibilidad de selección de diferentes rangos de tensión, con lectura clara y precisa de los valores que se desean medir, con comandos que resulten sencillos de utilizar, y finalmente que sea confiable, brindando niveles de seguridad adecuados. Principalmente, se establecerán las especificaciones de producto de forma precisa para garantizar que las mediciones realizadas con el mismo sean fehacientes y reproducibles. Los potenciales clientes de dicho producto versan desde los técnicos de reparación de instrumentos tecnológicos hasta los científicos que se vinculen con la electrónica, pasando por laboratorios universitarios y *hobbistas*. El precio estará apuntado a un sector intermedio, pues será un producto superior a los osciloscopios de la competencia de gama baja (entre U$S 100 y U$S 300) pero de usos más acotados que los de las marcas más prestigiosas (cuyos precios superan los U$S 1200 y llegan hasta U$S 5000).

2.2 Planteamiento del Problema a Resolver

El proyecto atraviesa campos diversos de la electrónica dado que se compone de una sección de electrónica analógica y otra sección de electrónica digital. Por este motivo, es esencial subdividir el mismo en etapas para analizarlas individualmente y luego desarrollar su interconexión.

En primera instancia, dado que se trata de un instrumento de medición, es indispensable que se hagan análisis exhaustivos de la precisión del instrumento. Para ello se deben analizar todas las secciones de forma individual y luego analizarlas en conjunto. Los resultados de dichos análisis darán una medida de la calidad del producto final y por lo tanto serán procedimientos críticos para evaluar la competencia del mismo y tomar conclusiones relevantes del trabajo realizado.

Una de las ventajas más importantes de los osciloscopios digitales es que proveen la posibilidad de extraer parámetros de mayor complejidad de la señal

que está siendo analizada. Para la extracción de dichos parámetros, se debe realizar un procesamiento digital de las muestras existentes. Es crítico que este procesamiento se realice en tiempos acotados y acordes con el resto de las restricciones temporales del sistema. De allí surge una relación de compromiso entre los parámetros que se le ofrecen al usuario y el ancho de banda de señales que pueden ser analizadas con el osciloscopio. En este sentido, se debe realizar un análisis exhaustivo de los procedimientos matemáticos más eficientes para extraer los parámetros requeridos. También se debe adecuar el hardware utilizado para este fin.

Uno de los problemas que requiere mayor atención es el desarrollo de una interfaz intuitiva mediante el uso de una pantalla táctil. La gran mayoría de los osciloscopios en la actualidad no utilizan una interfaz táctil como método principal de interactividad. Por el contrario, se suelen utilizar botones y perillas que incluso suelen tener una configuración estándar. Esto ayuda a que los osciloscopios de distintos modelos y fabricantes puedan ser utilizados por los mismos usuarios. El cambio tan drástico de interfaz genera un gran desafío para el diseñador dado que la mayoría de los usuarios no estarán habituados a un osciloscopio con menor cantidad de botones y perillas.

Es importante garantizar que la utilización de energía del producto sea la más eficiente posible. Esta es la principal medida ecológica que se debe tomar. La batería del mismo no debe estar sobredimensionada y con la intención de aumentar la autonomía del dispositivo, se deberá pensar en distintos modos de consumo de energía que inhabiliten o habiliten servicios y que sean óptimos en función de los requerimientos del usuario.

Por último, un problema que abarca todas las etapas de diseño, son las limitaciones de precio y tiempo que establecen para el desarrollo del prototipo. En todo momento se debe resolver la relación de compromiso entre el tiempo invertido en el análisis de un problema y el dinero invertido en la solución del mismo. La selección de componentes electrónicos debe ser adecuada para no invertir dinero excesivo. A su vez, se debe tomar especial consideración en los tiempos que se requieren en la compra de productos a empresas internacionales, cuyas variables de tiempo suelen tener rangos muy variables. Para lidiar con esta clase de problemas es importante realizar un planeamiento adecuado de tareas, teniendo en cuenta la posible simultaneidad de las mismas y los posibles percances que cada una pueda sufrir.

III. DEFINICIÓN DE PRODUCTO

Definir un producto electrónico requiere un análisis de los requerimientos del cliente para definir las especificaciones del diseño. Ahora bien, al tratarse de un producto cuyos clientes tienen cierta capacitación técnica, los requerimientos y las especificaciones están muy fuertemente relacionados (pues, por ejemplo, cierto cliente puede tener un requerimiento de frecuencia de muestreo específico). Por otro lado, el mercado está dado en buena medida por estudiantes de Ingeniería Electrónica (o carreras afines) como los mismos realizadores del proyecto. Es por eso que parte de la determinación de los requerimientos se puede derivar de los conocimientos propios sobre el producto. De todas maneras, para ampliar el estudio de mercado y obtener una definición más clara del producto se llevó a cabo una encuesta entre estudiantes y profesionales de electrónica.

3.1 Requerimientos

A partir de estos factores se identificaron los siguientes requerimientos:

Prestaciones en cuanto a las formas de medir: **Esto se refiere a la variedad de funciones de medición (*Quick Measures*, modos de *trigger*, etc.). La versatilidad del producto estará fuertemente relacionada a qué tantas opciones tenga el usuario a la hora de medir y qué tan satisfactoriamente funcionen dichas opciones.**

Prestaciones en cuanto a señales de entrada que puede medir: **La variedad de aplicaciones del osciloscopio implica que las señales de entrada que deberá poder recibir y procesar es amplia. En tanto mayor sea la variedad de señales que puedan ingresar, más se amplía el mercado del producto pues habrá más clientes que puedan ser satisfechos por las especificaciones.**

Conectividad: **De los resultados de la encuesta se observa que a la mayoría de los usuarios le interesa poder complementar la medición en el osciloscopio con el procesamiento en una PC. Es por esto que se vuelve un requerimiento el poder conectar, de alguna manera, el osciloscopio con una PC. Esto no requiere ser en tiempo real (aunque esto sería beneficioso), y puede ser cubierto con una conexión USB o con la capacidad de manejar un pendrive o algún otro dispositivo de almacenamiento.**

Portabilidad: **A un 82% de los encuestados le interesaría contar con un osciloscopio portátil, o lo consideraría una opción al elegir un instrumento. El proyecto en sí se centra en producir un osciloscopio *portátil*, pero hay una variedad de factores (tamaño, peso, etc.) que entran en juego al determinar qué tan portátil es el producto.**

Usabilidad: **La comodidad de la interfaz fue señalada como un punto clave por la mayor parte de los encuestados (un 48% le dio el puntaje máximo), por lo que se requiere que el osciloscopio sea *usable*, es decir, agradable para el usuario en presentación e interfaz.**

3.1.1 Construcción de la Casa de Calidad

Luego de definir los requerimientos es necesario definir de qué manera impactan éstos en las especificaciones del producto. Para esto se elabora una Casa de Calidad, que permite visualizar las relaciones entre los requerimientos y las especificaciones, las importancias de cada ítem y evaluar la performance esperada en comparación con competidores.

A partir de lo arrojado por la encuesta y los requerimientos, se llegó a la conclusión de que las especificaciones relevantes para este análisis son las siguientes:

- **Autonomía de la batería**
- **Peso**
- **Dimensiones**
- **Carga USB**
- **Soporte de tarjeta SD, pendrive o conexión USB a PC**
- **Funciones de medición (*Quick Measures*)**
- **FFT y funciones matemáticas**
- **Cursores**
- **Cantidad de canales**
- **Frecuencia de muestreo**
- **Ancho de banda**

- Rango de entrada permitido

- Resolución mínima de tensión

- Resolución y tamaño de pantalla

- Interfaz *touchscreen*

- Profundidad de memoria de canal

Los productos evaluados como competencia son los siguientes osciloscopios portátiles:

E'go DSO203: **Producto de origen chino que se comercializa en Internet. Es *open source*, es decir que su diseño está liberado de manera que cualquiera puede conseguir los esquemáticos, el software, etc. Es un producto de bajo precio (aproximadamente U$S200) y buen funcionamiento en general, pero la modalidad de fabricación y venta da pocas garantías sobre el efectivo cumplimiento de las especificaciones.**

Velleman HPS50: **Velleman es una compañía multinacional de origen Belga, con 36 años de experiencia en el desarrollo de electrónica. El HPS50 es, quizás, el competidor más cercano al producto que se busca; pero se espera que pueda superarse en las prestaciones. El precio del HPS50 es de U$S290 aproximadamente.**

Agilent U1602B: **Producto portátil de la marca Agilent (derivada de Hewlett-Packard y generalmente reconocida, junto a Tektronix, como líder en el mercado mundial de osciloscopios). Es un instrumento de alta tecnología y calidad, y su precio (U$S1360) está claramente por encima del objetivo de este proyecto, pero permite evaluar un ejemplo de especificaciones altamente deseables pero muy exigentes.**

Se muestra a continuación la Casa de Calidad producida, con el análisis de requerimientos y especificaciones y la comparación con la competencia.

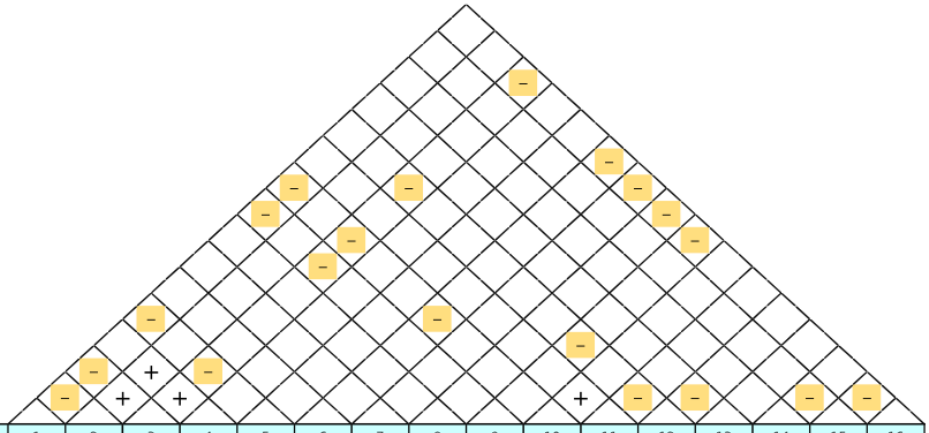
QFD: Casa de Calidad

Proyecto: Osciloscopio digital portátil con pantalla táctil
Revisión: 1
Fecha: 24/9/2011

Correlaciones
Positiva +
Negativa –
Sin Correlación

Relaciones
Fuerte ●
Moderada ○
Débil ▽

Dirección de mejora
Maximizar ▲
Target ◇
Minimizar ▼



Fila #	Gráfico de peso	Peso relativo	Importancia para el cliente	Relación máxima	Requerimientos del cliente \ Especificaciones	Autonomía de la batería	Peso	Dimensiones	Carga USB	SDCard / Pendrive / USB host	Funciones de medición	FFT/Math	Cursores	Cantidad de canales	Frecuencia de muestreo	Ancho de Banda	Rango de entrada permitido	Resolución mínima de tensión	Resolución / tamaño de pantalla	Interfaz touchscreen	Memoria de canal	Fila #
					Columna #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
					Dirección de mejora	▲	▼	▼	◇	◇	◇	◇	◇	▲	▲	▲	▲	▼	▲	◇	▲	
1	▮▮▮▮▮▮▮▮▮▮	22%	8	9	Conectividad					●												1
2	▮▮▮▮▮▮▮▮	19%	7	9	Prestaciones s/ formas de medir						●	○	○	○	○	▽	▽	▽			▽	2
3	▮▮▮▮▮▮▮▮	19%	7	9	Prestaciones s/ señales de entrada									○	●	●	●	○			○	3
4	▮▮▮▮▮▮▮	17%	6	9	Portabilidad	○	○	●	○													4
5	▮▮▮▮▮▮▮▮▮▮	22%	8	9	Usabilidad			○											○	●		5
					Target	4 horas	1 Kg	20x15x5	Sí	SDCard	Vpp, Vavg, freq, period, Vrms, RT, FT, DC, phase, ratio	FFT, suma y resta	Sí	2	50Msps	10 MHz	300Vrms	5mV/div	480x270	Sí	4 k puntos	
					Relación máxima	3	3	9	3	9	9	3	3	3	9	9	9	3	3	9	3	
					Rating de importancia técnica	50.0	50.0	216.7	50.0	200.0	175.0	58.3	58.3	116.7	233.3	194.4	194.4	77.8	66.7	200.0	77.8	
					Peso relativo	2%	2%	11%	2%	10%	9%	3%	3%	6%	12%	10%	10%	4%	3%	10%	4%	
					Gráfico de peso	▮	▮	▮▮▮	▮	▮▮▮	▮▮▮	▮	▮	▮▮	▮▮▮	▮▮▮	▮▮▮	▮▮	▮▮	▮▮▮	▮	
					Competencia #1: E'go DSO203	No especificado	80g + batería (total 250g)	9,8x6x1,45	Sí	USB a PC (pendrive interno)	Max, min, Vpp, Vdc, RMS	No	Sí	2 + 2 digitales	72Msps	30MHz mínimo	80Vpp	20mV/div	400x240px, 3''	No	4096 puntos	U$S 200
					Competencia #2: Velleman HPS50	5 horas	450g + baterías	22x10,5x3,5	No, línea	USB a PC	RMS, Vp	No	Sí	1	40Msps	5 - 12MHz	100Vp	5mV/div	192x112px	No	256 puntos	U$S 290
					Competencia #3: Agilent U1602B	4 horas	1,5 Kg	24,1x13,8x6,6	No, línea	USB client	Todas (Vpp, max, min,avg, freq, RT, etc)	FFT, suma y resta	Sí	2	200Msps	20MHz	300Vrms	5mV/div	320x240px, 4.5''	No	125 k puntos	U$S 1360
					Columna #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	

Fig 3.1: Casa de calidad del proyecto

Al analizar la Casa de Calidad se observa que las especificaciones más importantes son las siguientes, y en este orden:

- **Frecuencia de muestreo**
- **Dimensiones**
- **Ancho de banda**
- **Rango de entrada permitido**
- **Interfaz *touchscreen***
- **Soporte de tarjeta SD**
- **Funciones de medición**

Estas características son las que afectan en mayor medida a los requerimientos más importantes. Cabe notar que puntualmente la frecuencia de muestreo y el ancho de banda son, por lo general, el factor crítico a la hora de elegir un osciloscopio, pues suelen ser los limitantes más fuertes para la variedad de señales que se pueden medir. Lamentablemente, el ancho de banda es también uno de los objetivos más difíciles de cumplir, pues aumentarlo requiere un diseño cuidadoso de la placa y el uso de componentes de alta calidad.

Por otro lado, las dimensiones afectan muy fuertemente a la portabilidad y, en menor medida, a la usabilidad. Este factor está, sin embargo, muy limitado por la tecnología de fabricación utilizada (componentes, cantidad de capas, etc.) por lo que es de esperar que sea difícil superar a la competencia.

3.2 Especificaciones

A partir de la Casa de Calidad se determinan las siguientes especificaciones:

3.2.1 Especificaciones de Hardware

- Autonomía de la batería: **Debe ser de al menos 4 horas para permitir que el equipo se verdaderamente portátil con características similares a las de la competencia.**

- Peso: **No debe superar 1kg, de esta manera será fácil de transportar y sostener en la mano. Está muy afectado por el tamaño de la batería y los componentes utilizados.**

- Dimensiones: **Se establece un máximo de 20cm x 15cm x 5cm. De esta manera queda muy similar al tamaño del Agilent. No obstante, si el diseño de la placa no se ve afectado es posible (y deseable) que este valor se reduzca. El límite inferior es el tamaño de la pantalla.**

- Carga USB: **Esto implica utilizar un conector mini USB estándar y utilizar circuitos adaptadores de tensión para regular la carga de la batería. Permite que la carga se pueda realizar en la mayor variedad de condiciones, independizándose de normas de enchufes internacionales y de la necesidad de agregar una fuente para adaptar tensión de línea.**

- Soporte de tarjeta SD: **El hardware debe contar con un slot para tarjetas Secure Digital y el software debe soportar el protocolo especificado por SanDisk (ya sea con protocolo SPI o con el protocolo propietario de SD).**

- Funciones de medición: **Se debe poder medir automáticamente tensión media, pico a pico y RMS, frecuencia, período, rise-time, fall-time, *duty-cycle*, fase y razón de tensiones entre canales. De esta manera se llega a prestaciones similares a las del DSO203, claramente superiores a las del Velleman y ligeramente peores que las del Agilent. Es una especificación de software, pero es posible implementar parte en hardware (FPGA).**

- FFT y funciones matemáticas: **Al agregar FFT y suma y resta de canales se agrega una funcionalidad de software deseable, superando al DSO203 y al HPS50, con costo bajo por ser puro software o desarrollo en FPGA.**

- Cursores: **En eje X e Y, permiten realizar mediciones manuales.**

- Cantidad de canales: **Se realizará un osciloscopio de 2 canales. Según la encuesta, el 82% considera que es suficiente. Agregar canales agrega costo casi proporcionalmente pero no agrega demasiado desarrollo, por lo que es una mejora a pensar para futuros productos derivados.**

- Frecuencia de muestreo: **50 Msps es un valor tecnológicamente factible (depende principalmente del conversor AD a utilizar), que ubica el producto en superioridad al Velleman y en el mismo orden que los otros competidores. Se estudiará aumentarla si no implica un aumento considerable del costo, pues mejoraría enormemente la calidad del producto.**

- Ancho de banda: **Se establece un mínimo de 10MHz, considerando la dificultad de hacerlo más grande. No obstante, y de la misma manera que con la frecuencia de muestreo, un valor más alto siempre será beneficioso en tanto no aumente los costos considerablemente.**

- Rango de entrada permitido: **Se fija una tensión máxima de entrada de 300Vrms. Esto quiere decir que el producto debe poder medir sin problemas señales de ese nivel.**

- Resolución mínima de tensión: **5mV/división. Esto implica que la resolución del conversor AD debe permitir que discernir menos que eso. De todas maneras, la** resolución del AD**, especificación derivada de ésta, no será menor a** 8 bits **para que el osciloscopio funcione aceptablemente en esta escala.**

- Resolución y tamaño de pantalla: **Se utilizará una pantalla de 4 pulgadas o 4.3 pulgadas y 480x270 píxeles. Si las dimensiones de la placa y el costo lo permiten, se evaluará aumentarla a 7 pulgadas, pues esto mejoraría la usabilidad. Nótese que la resolución elegida supera a la de todos los competidores evaluados, pues lo central de este diseño (en cuanto a lo novedoso) es la comodidad de la interfaz.**

- Interfaz *touchscreen*: **Se usará una pantalla táctil resistiva. La operación del osciloscopio se hará enteramente a través de esta interfaz, salvo por el encendido y apagado.**

- Memoria de canal: **Debe poder almacenar al menos 4000 puntos por canal.**

Se definieron, entonces, especificaciones tecnológicamente factibles y competitivas de acuerdo a lo existente en el mercado, que permiten cubrir los requerimientos. Se estima que estas especificaciones se pueden cumplir con un costo **en componentes menor a U$S 300.**

3.2.2 Especificaciones de Software

- Funciones de medición: **Se implementarán las funciones de medición típicas para un osciloscopio: Tensión pico a pico, tensión media, tensión RMS, relación entre tensiones entre los dos canales, frecuencia, período, *rise time*, *fall time*, *duty cycle*, diferencia de fase entre los dos canales.**

- Herramientas matemáticas: **Se implementarán las siguientes herramientas matemáticas: FFT, suma y resta entre los dos canales.**

- Cursores: **Se implementará la funcionalidad de cursores en el osciloscopio que permitirán realizar mediciones específicas en el eje temporal y el eje de amplitudes.**

IV. ANÁLISIS DE FACTIBILIDAD

4.1 Factibilidad Tecnológica

4.1.1 Propuesta de alternativa de diseño

Para el análisis de factibilidad tecnológica se dividió al sistema en sus bloques fundamentales. La siguiente figura muestra cuales son y cómo se relacionan entre sí:

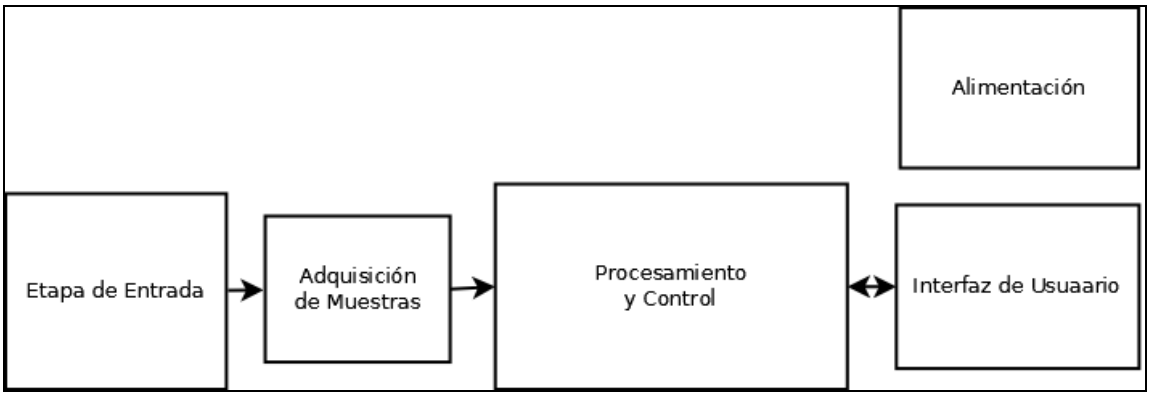


Fig. 4.1: Esquema en bloques del osciloscopio

Se consideraron distintas formas de implementación para cada uno de los bloques especificados. A continuación se detalla el análisis de alternativas de diseño.

Bloque Procesamiento y Control:

El bloque de procesamiento es una de las unidades más importantes del producto. Dentro de sus funciones se encuentran las siguientes: recibir las muestras de la señal provenientes del conversor A/D; controlar la interfaz gráfica (pantalla LCD) así como también la pantalla táctil; replicar el funcionamiento del osciloscopio, es decir llevar a cabo las tareas de *triggering* y control del sistema horizontal del mismo; y por último debe encargarse del procesamiento de la información, llevar a cabo las diversas funciones del osciloscopio (por ejemplo *average* o cualquier función matemática). Dadas estas tareas, surge una serie de

especificaciones mínimas que este bloque debe de cumplir para llevarlas a cabo y son las siguientes:

1. **Unidad de procesamiento con capacidad suficiente**
2. **Periférico de comunicación serie o paralelo para interfacear con el conversor A/D**
3. **Periférico controlador de LCD con soporte TFT**
4. **Periférico controlador de pantalla táctil 4-5 wire (recomendable)**

Como respuesta a este problema surgen varias alternativas que involucran distintos tipos de hardware (microcontrolador, FPGA, DSP) cada una con un enfoque diferente. A continuación se presentan las opciones más apropiadas para analizar en este caso:

Microcontrolador y FPGA (Field Programmable Gate Array): **La idea de esta opción s utilizar una FPGA como intermediario entre el conversor y el microcontrolador. La FPGA se encarga en este esquema de realizar dos de las tareas básicas del osciloscopio: el *trigger*, y el manejo del sistema horizontal. El uC en tanto, se encarga del procesamiento y control de la interfaz.**
La FPGA permite diseñar *hardware* específico a medida de los requerimientos del usuario. Esta solución permite implementar las funcionalidades básicas del osciloscopio de manera natural y directa, sin embargo no es la mejor alternativa cuando se necesitan módulos grandes como periféricos de comunicación o controladores de *hardware*. Si bien existen soluciones comerciales pre-hechas para este problema, hay alternativas que resultan más sencillas, como por ejemplo agregar al esquema el uso de un microcontrolador que complemente el trabajo de la FPGA. En la actualidad, los uC cuentan con una amplia variedad de periféricos integrados que facilitan el control y comunicación con *hardware* externo (conversores, módulos de comunicación, controladores, etc). Por lo tanto, un uC es una opción más robusta para cumplir las necesidades de procesamiento y control de interfaz en esta aplicación. Otra ventaja es la flexibilidad que brinda el uso de un uC, ya que por lo general trabajan con lenguajes de programación de nivel intermedio (como C). Esto facilita enormemente el desarrollo y mantenimiento de la aplicación, así como también resulta sencillo agregar nuevas funcionalidades a la misma (nuevas funciones matemáticas o de análisis de la señal que se está midiendo). De esta forma, se puede usar entonces una FPGA de tamaño reducido, con un funcionamiento específico, y un microcontrolador *mid-end* complementario que simplifique el control de la interfaz y el desarrollo del software/algoritmos de procesamiento.

Como desventaja de esta alternativa surge el hecho de que se requieren dos circuitos integrados para cumplir con la funcionalidad de este bloque. Esto requiere una especial atención y cuidado a la hora del diseño en PCB del circuito, y sobretodo del ensamblado de las partes. Si se logra un buen diseño esto no representaría un problema significativo.

Microcontrolador: **Otra opción podría ser utilizar únicamente un microcontrolador. Naturalmente, el microcontrolador necesario debe ser superior al del caso anterior (en cuanto a capacidad de procesamiento y periféricos disponibles) ya que debe realizar una mayor cantidad de tareas. Existen uC *high-end* con prestaciones más que suficientes que las necesarias en este caso, sin embargo el precio es un factor importante a considerar.**

Microcontrolador embebido en FPGA: **Se puede implementar un procesador dentro de una FPGA. Esto tiene las ventajas de la primera alternativa descripta sumadoa a que se usaría, en este caso, sólo un circuito integrado para el procesamiento. No obstante, el precio de la FPGA podría aumentar sensiblemente. Por otro lado, los requerimientos de memoria RAM pueden llegar a ser demasiado altos, lo cual aumentaría aún más el costo de la FPGA o implicaría la necesidad de agregar un integrado de memoria externa, lo cual complicaría el diseño y agregaría costo.**

DSP (Digital Signal Processor): **Otra opción a considerar es trabajar con un DSP. Los**
DSP son procesadores optimizados para trabajar con procesamiento de señales en tiempo real. Usualmente son más potentes y costosas que un uC típico pero por lo general no tienen la gran variedad de periféricos que sí tiene un microcontrolador. Son la solución óptima para tratar con problemas de procesamiento de señales en tiempo real, sin embargo se valen en gran medida de la eficiencia del lenguaje de programación utilizado. Es decir que para utilizar adecuadamente todo el potencial de un DSP es necesario programar en lenguaje de bajo nivel, *Assembly* típicamente.

Bloque de Adquisición de Muestras:

Este bloque consta principalmente de un conversor analógico-digital (conversor A/D) quien es el encargado de tomar la señal analógica de entrada (que ya fue previamente acondicionada en la etapa de entrada) y transformarla en una señal digital que pueda ser trabajada por el bloque de procesamiento. De acuerdo a las especificaciones que se plantearon para el desarrollo de este producto (ver 3.2 Especificaciones) el conversor A/D debe cumplir con las siguientes características:

1. **Frecuencia de muestreo de 50 Msps**

2. **Resolución de al menos 8 bits**

Nuevamente existe más de una alternativa, las opciones consideradas se describen a continuación:

- *Integrado a unidad de procesamiento:* **La mayoría de los microcontroladores y DSP cuentan con un conversor A/D, D/A integrado. Las prestaciones de estos conversores dependen fuertemente de las del procesador que los acompaña. Para la mayor parte de los problemas los conversores integrados a un uC son suficientes en cuanto a resolución y frecuencia de muestreo, pero en este caso, al buscar una tasa de 50Msps, se requeriría uno de altísima calidad. La gran ventaja de este tipo de conversores es que no requiere un desarrollo extra en cuanto a diseño de PCB se refiere dado que forma parte del chip del uC/DSP.**

- *Conversor externo:* **Los conversores externos son circuitos integrados con la función específica de actuar como conversor A/D, D/A. Existe una gran variedad en cuanto a prestaciones se refiere, superando ampliamente en características de frecuencia de muestreo por ejemplo a los integrados a un uC. Un factor a considerar es el hecho de que la evolución histórica demostró que la tecnología de conversores avanza más rápido que la tecnología de los microcontroladores. Es decir que un conversor quedará obsoleto antes de que lo haga la unidad de procesamiento (por obsoleto se entiende que existirá un producto con mejores prestaciones y menor precio). Esto es una ventaja de los conversores externos, ya que son más fáciles de reemplazar. La principal desventaja es que se debe diseñar cuidadosamente el PCB para que el ruido no afecte al conversor, caso contrario se podría hasta llegar a perder algunos de los bits menos significativos.**

Bloque Alimentación:

Uno de los requerimientos esenciales del producto es la portabilidad del mismo, por lo que la elección de la batería del equipo es crítica. El osciloscopio tendrá una batería recargable y contará con un cargador aparte conectado a la red eléctrica. Los principales tipos de baterías recargables disponibles en el mercado son:

- *Baterías de Plomo (Pb):* **Son las baterías de mayor antigüedad en el mercado en parte por su elevada vida útil y su alta relación potencia entregada - peso (potencia específica). Sin embargo, están hechas de compuestos tóxicos y**

contaminantes que no solo pueden traer problemas para el medio ambiente sino para el ser humano en caso de exposición prolongada.

- *Baterías de Níquel (NiCd-NiMh):* **Una tecnología más reciente son las baterías de Níquel. En la actualidad, una gran porción del mercado utiliza este tipo de soluciones dado a que son más amigables con el medioambiente aunque son inferiores a las baterías de plomo en lo que se refiere a la potencia específica. Una desventaja importante de estas baterías es el complejo método de carga, por corriente constante, que requiere disponer de una fuente de corriente y un sensor de corriente. El proceso de carga aumenta la temperatura de la batería considerablemente y puede dañar su vida útil si no se realiza correctamente.**

- *Baterías de Ion-Litio (Li-Ion):* **La última alternativa analizada son las baterías basadas en iones de litio. Es la tecnología preponderante hoy en día, abarcando casi un 70% del mercado. La gran mayoría de los dispositivos móviles (celulares, reproductores de música, computadoras portátiles, etc) usan baterías de litio. Poseen en general una potencia específica superior incluso a las baterías de plomo con un nivel de contaminación inferior. La gran ventaja de esta tecnología sin embargo es el método de carga, la carga por tensión constante es un proceso más sencillo que la carga por corriente. En contrapartida, las baterías de litio tienen una vida útil no tan prolongada como las de plomo.**

Bloque Interfaz de Usuario (LCD+touchscreen):

Uno de los requerimientos principales del producto resultó ser la usabilidad del mismo. La comodidad de la interfaz fue señalada como un punto clave por los encuestados. Dado que la interfaz del producto será exclusivamente a través de la pantalla táctil resulta de gran importancia la elección tanto de la pantalla LCD como de la tecnología *touchscreen* que se utilizará. Existen varias alternativas en el mercado incluyendo la posibilidad de adquirir LCDs con *touchscreen* integrado.

- *Pantalla LCD*

En el caso de las pantallas LCD hay una gran variedad de opciones disponibles, aunque las principales alternativas son LCDs basados en TFT, LED o OLED. En el mercado actual, las pantallas TFT están ampliamente difundidas, y aceptadas como una tecnología accesible económicamente y de buenas prestaciones, incluso algunos microcontroladores cuentan con hardware dedicado para su

control. Por otro lado, es cierto que las pantallas OLED (LED) son superiores en cuanto a calidad se refiere. Como desventaja sin embargo, las pantallas OLED por ser una tecnología más reciente suelen ser más caras, la cantidad de fabricantes es menor, y la disponibilidad en cuanto a tamaños y variedad en general es menor.

- *Touchscreen*

Existen dos tecnologías preponderantes en el mercado en la actualidad y son los paneles resistivos y los capacitivos. Las diferencias entre ambos son constructivas y de funcionamiento. A la hora de usarlos sin embargo cada uno tiene características especiales. Los paneles capacitivos funcionan únicamente con objetos que sean conductores eléctricos capaces de modificar la capacitancia de la pantalla. Los paneles resistivos por otro lado simplemente requieren de un objeto que ejerza presión sobre el panel. Constructivamente los paneles capacitivos son más complejos, por ello suelen tener un precio considerablemente mayor a los resistivos. En cuanto a la interfaz eléctrica, los resistivos son más sencillos, típicamente se controlan con tan solo 4 pines. A pesar de contar con estas ventajas, los paneles resistivos tienen la gran desventaja de que por hardware no soportan *multitouch*, es decir que solo pueden trabajar con un punto de la pantalla a la vez. Existen algunas soluciones por *software* que buscan remediar esta situación pero su performance es inferior a la de los paneles capacitivos. Definitivamente si se desea dar soporte para gestos *multitouch* se deberá utilizar paneles capacitivos, pero se deberá considerar la opción de los resistivos por una cuestión de costo.

4.1.2 Elección de una solución

Bloque Procesamiento

La elección de la unidad de procesamiento resulta la más compleja de analizar dado que las alternativas son muy dispares entre sí y no es sencilla la comparación. Finalmente, se optó por la combinación microcontrolador y FPGA ya que brinda la mayor flexibilidad posible en el desarrollo, sobretodo pensando en el mantenimiento y mejoras a futuro del sistema. Se trabajará con integrados *mid-end* ya establecidos en el mercado, esto nos da dos ventajas. Por un lado, los precios son más accesibles; por el otro, al ser productos más masivos (en contraposición con un integrado muy especializado) se tendrá seguramente una mejor documentación y soporte.

Bloque Conversor A/D

En cuanto a la elección de la alternativa óptima de diseño para el bloque del conversor A/D se decidió trabajar con un conversor externo. Los motivos son varios, principalmente debido a las prestaciones que se necesitan del conversor. Como se estableció en el apartado *3.2 Especificaciones* el conversor debe ser capaz de muestrear a 50 MSPS con al menos 8 bits de resolución. Los uC que tienen un conversor con esas especificaciones son por lo general los denominados *high-end*, los uC más avanzados del mercado. El elevado precio de estos circuitos integrados no se justifica para utilizar su conversor A/D existiendo la posibilidad de contar con un conversor externo de precio relativamente bajo. Por otro lado, trabajar en un esquema de conversor e unidad de procesamiento por separado le brinda al sistema más flexibilidad a la hora de reemplazar un componente.

Bloque Alimentación

A la hora de elegir una alternativa para el bloque de alimentación se priorizó el método de carga de la batería. Se decidió trabajar con una batería Ion-Litio. El hecho de que se carguen por tensión constante simplifica el diseño significativamente. Por otro lado esta tecnología provee de la mayor potencia específica, lo cual en un ambiente donde el espacio es reducido implica una mayor capacidad en el mismo volumen. Otro factor importante que se consideró es el impacto ambiental, por ello se usarán baterías que acatan la directiva RoHS. Por último, el alto grado de inserción en el mercado de las baterías de litio hace que los precios sean en general más accesibles que el resto de las tecnologías.

Bloque Interfaz de Usuario

Se decidió trabajar con una pantalla LCD basada en TFT y un panel *touchscreen* resistivo. La decisión de la pantalla resulta sencilla si se la analiza desde la perspectiva del diseño del equipo. Se prefirió trabajar con un producto que no es el estado del arte pero está mundialmente aceptado y cuenta con una gran cantidad de soporte. Ejemplo de esto es la posibilidad de usar un uC con controladora TFT integrada, situación no tan frecuente con tecnología OLED. Otro factor decisivo es el precio y la disponibilidad de los productos, donde la tecnología TFT tiene un amplio margen por sobre la OLED. El hecho de que TFT no sea la tecnología más avanzada hoy en día no implica un riesgo mayor ya que la gran mayoría de los productos en el mercado siguen utilizándola. Factores similares fueron los que llevaron a la decisión de usar paneles *touchscreen* resistivos por sobre los capacitivos. En este caso se priorizó la simplicidad del diseño y el costo, los paneles resistivos proveen las funcionalidades necesarias

para el producto que se busca desarrollar siempre teniendo en mente a la usabilidad del mismo.

4.1.3 DFMEA

Como herramienta para asegurar la calidad del producto, minimizar las futuras fallas y garantizar la factibilidad tecnológica del producto, se realiza un DFMEA, que consiste en una enumeración de las posibles fallas y un análisis de su severidad, detectabilidad y probabilidad de ocurrencia. De esta manera se obtienen indicadores cuantitativos (basados en la experiencia de los integrantes del equipo) de la prioridad de cada modo de falla, y se plantean cuidados especiales a tener en el diseño para minimizar el riesgo.

Pos.	BLOQUE / FUNCIÓN	MODO POTENCIAL DE FALLA	CAUSA POTENCIAL	EFECTOS POTENCIALES	METODO DE DETECCIÓN	SEV	OCC	DET	RPN	Acciones recomendadas
1	**Alimentación**	Baja tensión	Batería baja	Sistema inoperable	Brownout detector	6	2	4	48	Indicador de batería baja
2		Polaridad incorrecta	Error en la colocación de la batería	Falla destructiva	Revisión de la conexión	8	2	3	48	Conector polarizado - circuito de protección
3		Tensión nula	Conexiones deficientes	Sistema inoperable	Medición con multímetro	6	2	2	24	Cuidado en la soldadura - materiales de calidad
4		Sobretensión	Falla de reguladores, sobretensión desde el cargador	Falla destructiva		8	4	7	224	Circuito de protección
5	**Procesamiento y control**	uC/FPGA inoperables	Ripple de tensión	Sistema inoperable		6	2	6	72	Regulación de tensión, desacople
6		Puerto quemado	Corriente excesiva	Falla de periféricos	Display inoperable, medición errónea	6	4	6	144	Margen de las corrientes de salida dentro de las especificaciones del uC
7		uC/FPGA inoperables	Error de programación	Sistema inoperable	Debuggeo, testeo	6	5	4	120	Debuggeo intenso, control de fallas por software

8	**Etapa de entrada**	Sobretensión	Entrada de tensión superior a la especificación	Falla destructiva, peligro para el usuario		9	4	7	252	Circuito de protección, documentación para usuario
9		Alinealidad, errores de medición	Falla de componentes de filtros, atenuadores, amplificadores	Errores de medición	Testeo de mediciones	6	3	5	90	Testeo intenso, componentes de calidad
10		Entrada nula	Falla de conector	Medición nula	Testeo de mediciones	6	3	3	54	Conector de calidad
11	**Conversor A/D**	Entrada quemada	Entrada de tensión superior a la especificación	Sistema inoperable	Medición de prueba	6	3	5	90	Circuito de protección, documentación para usuario
12		Medición errónea	Falla en el ruteo, ruido o problemas de propagación	Errores de medición	Testeo de mediciones	6	3	3	54	Consideraciones en el ruteo de pistas de alta frecuencia
13	**Display/Touchscreen**	Backlight inoperable	Falla de alimentación	Equipo inutilizable	Medición con multímetro	7	3	3	63	Componentes de calidad, precaución en la soldadura
14		No muestra lo que debe	Error de programación	Equipo inutilizable	Debuggeo, testeo	6	5	4	120	Debuggeo intenso, control de fallas por software
15		Touchscreen inoperable	Error de conexión	Equipo inutilizable	Revisión de la conexión	6	3	4	72	Conector polarizado
16	**PCB**	Falla de comunicación entre bloques	Errores de ruteo, soldaduras frías	Sistema inoperable	Testeo con osciloscopio, mediciones con multímetro	6	5	4	120	Testeo intenso, consideraciones en el ruteo, componentes de calidad
17	**Gabinete**	Rotura	Golpe, caída	Falla destructiva	A simple vista	7	4	1	28	Materiales de calidad, diseño mecánico robusto
18		Falla de aislación	Caída o sobretensión	Riesgo para el usuario		8	4	7	224	Diseño mecánico seguro

Fig 4.2: DFMEA del proyecto

4.2 Factibilidad de Tiempos

4.2.1 Planificación - Pert y Montecarlo

Con el objetivo de planificar adecuadamente el desarrollo del proyecto se dividió al mismo en una lista de tareas, por bloques. Se armó un diagrama de Pert considerando las precedencias de las tareas, sus duraciones optimas, medias y pesimistas. La duración de cada tarea se consideró una variable aleatoria, obteniéndose la duración estimada total del proyecto como la suma de todas las variables. La estimación estará acompañada de un porcentaje de riesgo para hacer más realista la estimación. A continuación se adjunta el diagrama de Pert

del proyecto, especificándose las tareas a realizar en cada aspecto junto con su duración esperada.

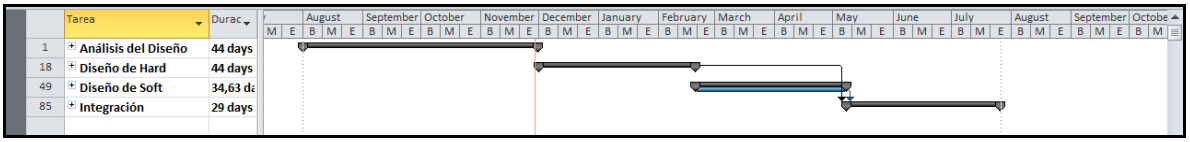


Fig 4.3: Esquéma general del diagrama de Pert

El primer diagrama muestra la división por etapas realizada. Las siguientes imágenes muestran las tareas a realizar en cada etapa.

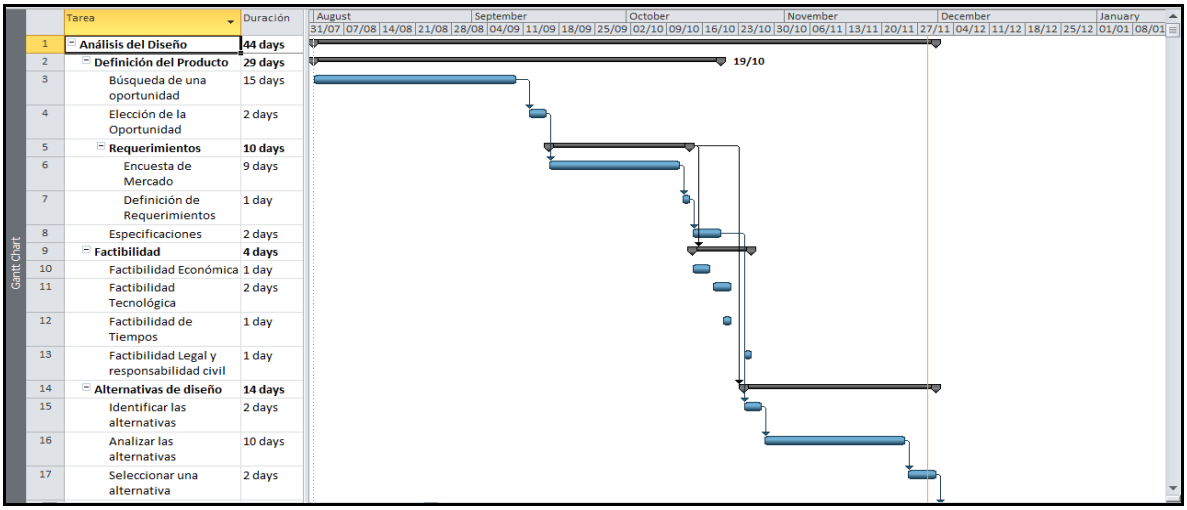


Fig 4.4: Pert - Detalle de 'Análisis del Diseño'

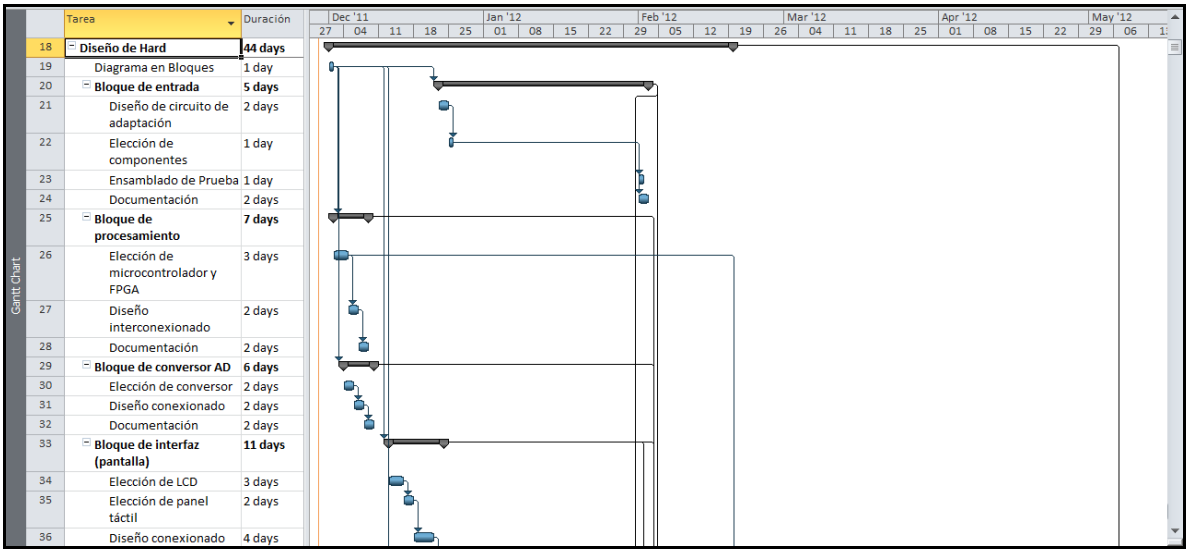


Fig 4.5: Pert - Detalle de ' Diseño de Hard' 1

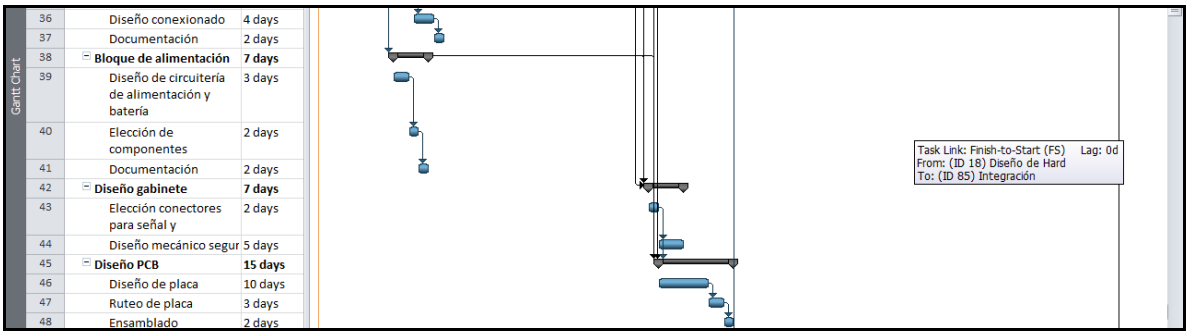


Fig 4.6: Pert - Detalle de 'Análisis del Diseño' 2

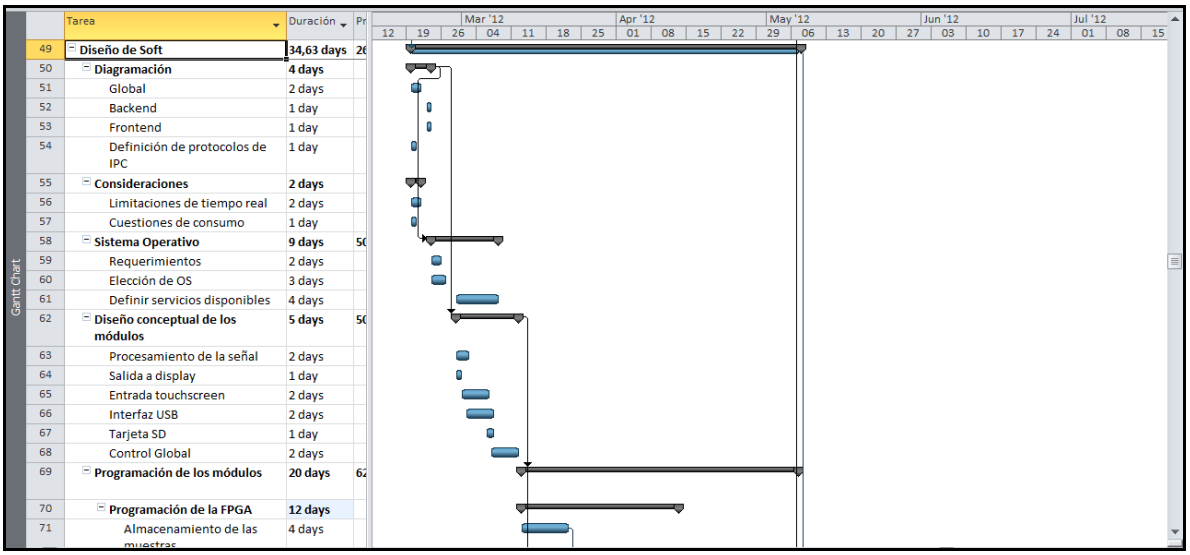


Fig 4.7: Pert - Detalle de 'Diseño de Soft' 1

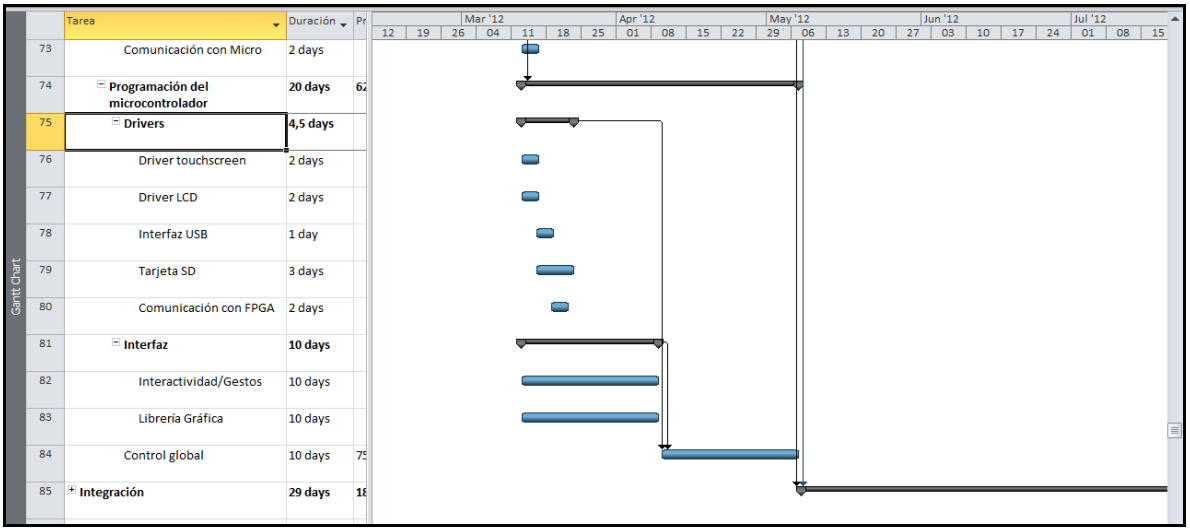


Fig 4.8: Pert - Detalle de 'Diseño de Soft' 2

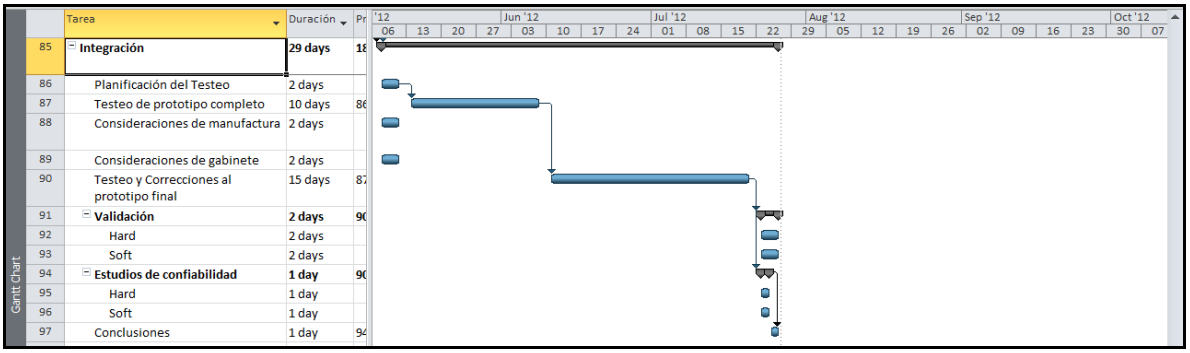


Fig 4.9: Pert - Detalle de 'Diseño de Soft' 3

Hay que destacar sobre el diagrama que la duración de las tareas está especificada en horas hombre. Se consideraron 4 horas hombre diarias para la mayor parte del proyecto. Para los meses de diciembre y febrero se consideraron 8 horas hombre diarias. El mes de Enero se dejo libre.

De acuerdo al diagrama realizado, se estima la finalización del proyecto para fines de Julio del 2012 basándose en la duración media de cada tarea. Usando una simulación de Montecarlo se puede obtener un análisis más complejo, con resultados más precisos y con un cierto margen de riesgo o probabilidad acumulada. Para ello es necesario determinar lo que se denomina camino crítico del proyecto.

A partir del diagrama anterior se determinaron las tareas que forman el camino crítico. A cada tarea se le asignó además de su tiempo medio, un tiempo óptimo y un tiempo pesimista. La siguiente tabla muestra cuales son dichas tareas:

ID	Tarea a realizar	Tiempo óptimo [días]	Tiempo medio [días]	Tiempo pesimista [días]	Muestra Pert [días]
1	**Análisis del Diseño**	**31**	**44**	**67**	
2	Definición del Producto	20	29	44	29.50267642
9	Factibilidad	1	1	3	1.001370066
14	Alternativas de diseño	10	14	20	12.05855042
18	**Diseño de Hardware**	**32**	**44**	**62**	
19	Diagrama en bloques	1	1	3	1.168632177
20	Bloque de entrada	3	5	7	5.474969608

25		Bloque de procesamiento	6	7	12	7.277423846
33		Bloque de interfaz	9	11	15	11.35787864
42		Diseño de gabinete	2	2	5	3.175558526
45		Diseño de PCB	11	15	20	15.44191007
49	**Diseño de Software**		**25**	**34**	**47**	
50		Diagramas	3	4	5	4.32342928
58		Sistema Operativo	3	5	7	5.798379528
62		Diseño de los modulos	4	5	10	4.207167712
74		Programacion uC	15	20	25	17.21145924
85	**Integración**		**23**	**29**	**39**	
86		Planificación del Testeo	1	2	3	1.174592809
87		Testeo del prototípo	8	10	13	10.25264979
90		Correcciones y retesteo	12	15	17	15.89669404
91		Validación	1	1	3	1.013432212
94		Confiabilidad	1	1	3	1.267052688
Tiempo Total [días]			**111**	**151**	**215**	**148**

Utilizando estos valores como referencia se realizó un análisis de Montecarlo. Los resultados se observan a continuación.

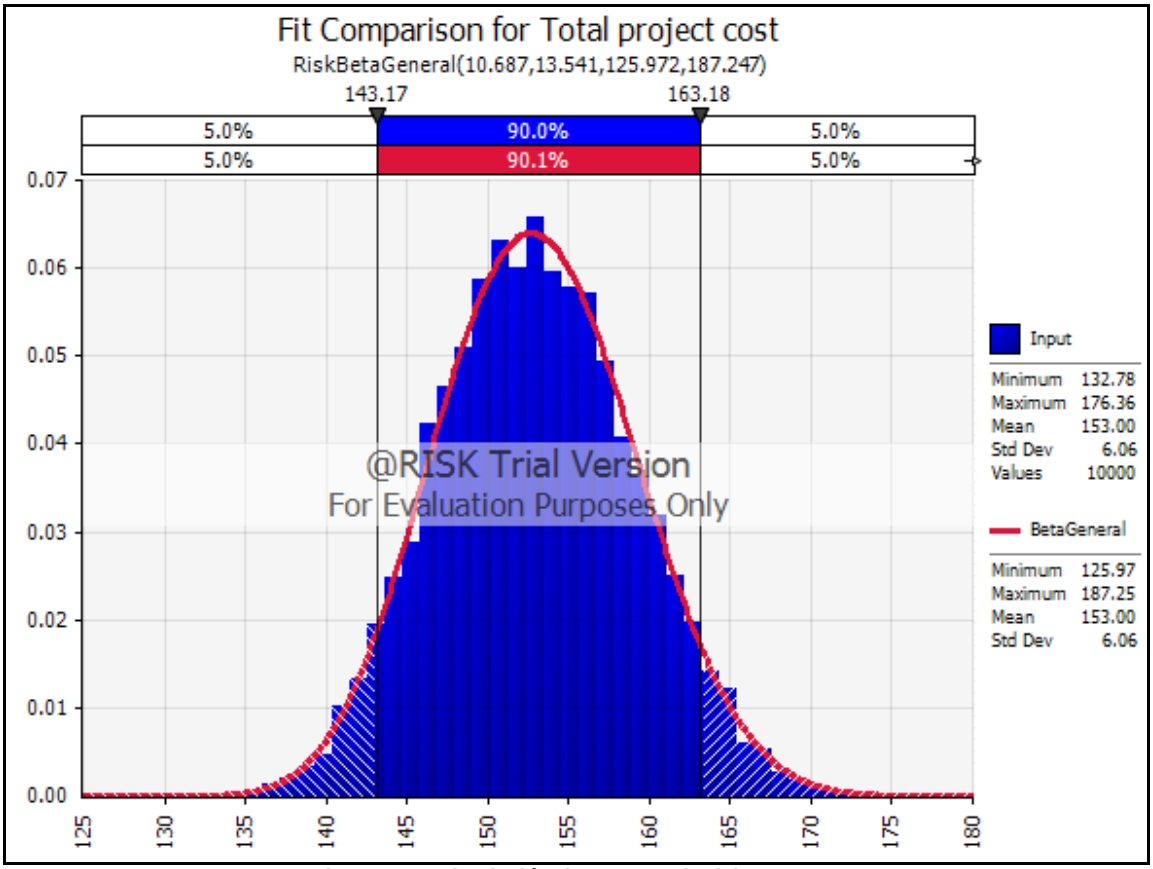


Figura 4.10: Simulación de Montecarlo del proyecto

La Figura 4.10 muestra la distribución de probabilidades de la variable aleatoria que representa la duración del proyecto. Como ejemplo, se ajustó la curva con una distribución Beta. Esta gráfica permite determinar la fecha de finalización del proyecto con un determinado riesgo. Para ello se construyó la curva de probabilidad acumulada a partir del histograma calculado en la simulación.

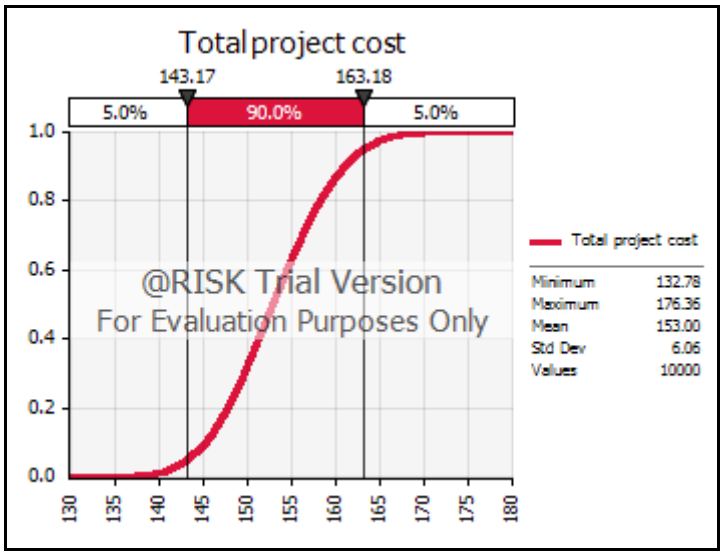


Figura 4.11: Probabilidad Acumulada de la duración del proyecto

La curva de probabilidad acumulada da la probabilidad de que el proyecto sea completado antes de una cierta cantidad de días. Los valores críticos se resumen en la tabla a continuación.

Duración en días	Fecha Estimada	Probabilidad Acumulada
140	11/7/2012	1%
143.26	16/7/2012	5%
145	18/7/2012	7%
150	25/7/2012	30%
151	26/7/2012	32%
155	2/8/2012	60%
160	9/8/2012	85%
163.32	14/8/2012	95%

A partir del análisis de Montecarlo, se puede afirmar que el proyecto concluirá antes del día 14 de Agosto de 2012 con un 95% de probabilidad.

4.3 Factibilidad Legal y Responsabilidad Civil

La validación legal del producto desarrollado es un factor que no debe dejarse de lado si se pretende entrar a un mercado. Existen dos componentes importantes que deben ser consideradas: por un lado las certificaciones propias del producto que sean requeridas para su comercialización; por el otro la adquisición de licencias del software que se utilizará para el desarrollo del producto.

El software que se utilizará dependerá fuertemente de los componentes elegidos para el bloque de procesamiento. Tanto la FPGA como el microcontrolador requieren *software* específico para su programación. Los fabricantes proveen del *software* necesario ya sea con licencia paga o como una alternativa libre y gratuita. Esto puede incluso ser un factor que determine la elección final de algunos componentes. Por otro lado, se requiere de *software* de desarrollo de circuitos impresos (en este caso se trabajará con el DXP Altium Protel) así como también de simulación matemática (Matlab). Estos productos no son de licencia gratuita por lo que deberán ser adquiridos. Los costos asociados serán puestos de manifiesto en el análisis de factibilidad económica.

En cuanto a las certificaciones necesarias para el producto se destacan dos categorías: seguridad eléctrica y compatibilidad electromagnética. Hay que destacar que estas certificaciones son dependientes de la región en la que se desea comercializar el producto.

Las normas de seguridad eléctrica contemplan las protecciones contra descargas, corrientes de fuga, filtración de líquidos, condiciones de falla, etc. En la Argentina, la resolución 92 del año 1998 regula todo lo que se consideran equipos electrónicos de baja tensión. El Instituto Argentino de Normalización y Certificación (IRAM) es quien brinda el sello de conformidad y asegura el cumplimiento de la resolución 92. La certificación de un producto bajo los estándares IRAM tiene validez internacional bajo la normativa ISO, y permite la comercialización del producto tanto dentro como fuera del país. Los ensayos necesarios para obtener la acreditación IRAM/ISO pueden realizarse en nuestro país por el Instituto Argentino de Ensayos y Validaciones (IADEV).

Por último, hay que destacar la certificación de compatibilidad electromagnética, que es un certificado sobre emisión y susceptibilidad electromagnética. Este certificado asegura que el equipo no corre riesgo de malfuncionamiento por influencia de la radiación exterior, y que a su vez las emisiones del mismo equipo no son perjudiciales tanto para la salud humana como para la interferencia con otros equipos. En la Argentina, los ensayos de compatibilidad electromagnética son realizados por el Instituto Nacional de Tecnología Industrial (INTI).

Un último tópico que se quiere analizar es el de la responsabilidad social, en particular en lo relacionado al impacto medioambiental del producto desarrollado. La creciente preocupación de la sociedad por el medioambiente hace que dentro del diseño de un producto se deba trabajar cuidadosamente en los aspectos relacionados al mismo. Hoy en día, lograr un producto con características *eco-friendly* puede representar un valor agregado importante a los ojos del mercado y el consumidor, incluso puede resultar un factor determinante a la hora de elegir entre dos productos de especificaciones similares. La directiva más importante en relación al cuidado medioambiental es la propuesta por la Unión Europea, RoHS, que restringe la utilización de algunas sustancias peligrosas para el medioambiente en aparatos eléctricos y electrónicos. Conseguir una certificación RoHS para un equipo propio puede resultar costoso tanto en tiempo como en dinero. Es por eso que con el objetivo de lograr un producto lo más *eco-friendly* posible se buscará trabajar con componentes RoHS.

4.4 Factibilidad Económica

El desarrollo de este producto puede llevar a un emprendimiento profesional de diseño y producción de instrumentos de medición. Es menester, entonces, analizar las variables económicas en juego para determinar la rentabilidad del producto. Para esto, se analizará el mercado en cuestión y luego se estimarán los resultados económicos esperados.

4.4.1 Análisis del mercado

El mercado mundial de instrumentos de medición crece en conjunto con el desarrollo tecnológico, pues concierne a las herramientas necesarias para dicho desarrollo. Cualquier desarrollo electrónico requiere contar con un osciloscopio y, según el campo de aplicación, puede requerir además de generadores de señales, analizadores de espectro, analizadores de impedancia, etc.

Analizaremos, para este caso, el mercado de osciloscopios, pues es el que con más claridad permite mostrar los órdenes de magnitud involucrados, además de que este es el primer producto que lanzaremos.

Observando el análisis de un estudio de Frost & Sullivan, se aprecia que el mercado de osciloscopios es uno consolidado y maduro, pero que atraviesa en este momento una crisis y presenta perspectivas de crecimiento. De ingresos de 1269,7 millones de dólares en el 2006 se cayó a 987,7 con la crisis del 2009, pero

se proyecta que llegará a 1513,4 en el 2014. Se trata, entonces, de un crecimiento del orden del 50% para los próximos tres años.

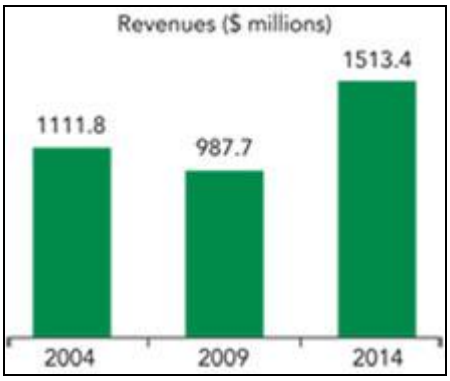


Fig 4.12: Ingresos en el mercado de osciloscopios

Ahora bien, nuestra estrategia atacará primero a los mercados argentino y latinoamericano. Esto implica entrar como uno de los primeros *players* en la región, donde antes sólo trabajaban empresas de otros continentes. Un estudio de Global Industry Analysts muestra que hay 90 *players* en el mercado mundial de osciloscopios, pero ninguno es de Latinoamérica.

El mercado argentino es pequeño pero en crecimiento. Un estudio de la Secretaría de Estado de Ciencia, Tecnología e Innovación Productiva del 2006 muestra que el gasto total en "Equipamiento y Rodados" (que involucra muchos factores además de instrumentos de medición) para actividades de innovación tecnológica fue de 234 millones de pesos, con un crecimiento del orden del 25% anual en los cinco años anteriores. Esta cifra incluye una gran variedad de equipos, pero permite estimar que la tasa de crecimiento sería similar.

Un mercado latinoamericano importante para atacar es el brasileño: el gasto total en desarrollo tecnológico fue en el 2005 unas 5,3 veces más grande que el argentino, y creció en más de un 100% desde entonces.

4.4.2 Canales comerciales

El principal canal de venta serán los distribuidores de electrónica. Esto nos permitirá entrar rápidamente en mercados internacionales y sostener un nivel de ventas considerable con una producción correctamente dimensionada.

A este canal se suma el de una página *web* para venta directa *online*. Esto permitirá atacar a *hobbistas* y estudiantes, es decir, el lado *business to consumer* del negocio.

El tercer canal de ventas es el contacto directo con universidades. No obstante, este canal no se atacará con el objetivo principal de generar ingresos sino desde un punto de vista estratégico: el uso de nuestro instrumental en laboratorios universitarios permitirá que la próxima generación de ingenieros tenga el conocimiento y el hábito de uso de nuestros productos, ayudando fuertemente al posicionamiento de la marca. Por este motivo, se trabajará en este segmento con precios promocionales y fuertes descuentos y beneficios (posiblemente trabajando a pérdida), como parte del presupuesto de marketing.

4.4.3 Marketing

Al tratarse de un segmento de mercado bien definido, el marketing será muy focalizado. Se contratará publicidad en publicaciones específicas de electrónica (publicaciones de la IEEE, revistas de *hobbismo* electrónico, foros, etc.) y se hará presencia en congresos y ferias de electrónica, procesamiento de señales, sistemas embebidos, y otras áreas de la industria.

El marketing a través de universidades será fundamental, como ya se ha expresado, pues permite un posicionamiento de marca fuerte a mediano/largo plazo. Esto se atacará a través de la venta directa con descuentos pero también con *roadshows* y *sponsoreo* de actividades (por ejemplo, la feria de electrónica del ITBA y eventos similares en otras universidades locales y extranjeras).

4.4.4 Presupuesto de marketing y ventas

El aspecto central en cuanto a marketing y ventas es el contacto con distribuidores, por lo que buena parte del presupuesto estará destinado a las misiones para reunirse con potenciales distribuidores (además de contactar especialistas de comercio exterior).

Las actividades presenciales como *roadshows*, congresos y ferias son de costo relativamente bajo en comparación con la publicidad en publicaciones, además de permitir un *conversion rate* más alto, si bien la publicidad permite llegar más rápido a más potenciales clientes y mercados y ayuda con más claridad a la percepción del posicionamiento de marca. Se destinará, por lo tanto, un 60% del presupuesto de marketing a las actividades presenciales y el 40% restante a publicidad. Se estima como óptimo, en este caso, destinar entre un 30 y un 40% de la inversión inicial a éstas áreas. Porcentajes más bajos serían también

aceptables, pero harían que el escalamiento de las ventas sea más lento, llevando a un tiempo de retorno de inversión más largo.

4.4.5 Organización de la producción

Considerando que el mercado de osciloscopios es altamente competitivo a nivel mundial, se debe minimizar el costo de fabricación del equipo. El esquema más eficiente, en este sentido, y considerando la posibilidad de un escalamiento rápido de la producción, es el de tercerización en China. Se cuenta allí con una amplia oferta de empresas de fabricación y ensamblado. Por otro lado, este esquema implica la cercanía de proveedores de componentes, de manera que se minimizan los costos de transporte de materia prima. Asimismo, se reduce la inversión inicial y los costos fijos, pues no es necesaria una fábrica propia para realizar el ensamblado.

En el largo plazo se puede pensar en una fábrica propia, pero esto sólo sería rentable contando con varios productos y una producción a mayor escala.

Sí se realizarán *in-house* los prototipos correspondientes y las primeras preseries de producción, para la realización de pruebas piloto y ensayos.

4.4.6 Costos Fijos

Al analizar los costos fijos se debe tener en cuenta que no son exclusivos de este producto, ya que, si se planea llevar a cabo un emprendimiento, se desarrollarán otros productos relacionados (es decir, otros modelos de osciloscopio y otros instrumentos de medición). No obstante, resulta necesario estudiar la composición de estos costos para determinar la viabilidad del producto. No consideraremos los costos iniciales que constituyen activos fijos (principalmente instrumental de laboratorio) pues se amortizarán más allá del ciclo de vida de este producto.

Tampoco consideraremos los costos de sueldos de desarrollo, pues sólo se harán efectivos para productos futuros.

Concepto	**Costo**
Sueldos	4000 U$D

Oficinas	1000 U$D
Contaduría y Asesoramiento Legal	1000 U$D
Marketing y Venta	2000 U$D

Por lo tanto, contamos con un costo fijo de aproximadamente U$D 7000 mensuales.

4.4.7 Costo Inicial

Luego del desarrollo electrónico, se deberá incurrir en costos iniciales para lanzar el producto al mercado.

Concepto	**Costo**
Diseño de gabinete	3000 U$D
Licencias de software	2000 U$D
Matrices (de gabinete y placa)	3500 U$D
Certificaciones	3000 U$D

4.4.8 Costo Variable

El costo para cada producto proviene de los componentes, la fabricación de la placa y el gabinete, el ensamblado y *packaging*. Los costos de componentes se estiman a partir de evaluar, para los involucrados según la alternativa de diseño

elegida, precios promedios de distintos componentes de las prestaciones necesarias.

Concepto	**Costo**
Componentes	160 U$D
Gabinete	20 U$D
Fabricación placa	8 U$D
Ensamblado	20 U$D
Packaging	5 U$D

Esto nos da un costo variable de aproximadamente U$D 213 por osciloscopio.

4.4.9 Precio de venta

Considerando los competidores evaluados, y los otros productos disponibles en el mercado, se estima que el producto podrá ubicarse en un precio de aproximadamente U$D 500 (sin contar distribución y transporte, es decir, precio FOB para el distribuidor). Esto ubica al producto en un punto medio entre los osciloscopios de alta gama (*Agilent* y *Tektronix*) de especificaciones similares, y los de gama baja, lo cual es razonable dadas las especificaciones y la calidad del osciloscopio.

4.4.10 Ventas y ciclo de vida

Se estima que el producto tendrá un ciclo de vida de aproximadamente 2 años. Esta es una estimación conservadora, pero que considera el rápido avance de la tecnología en el mercado e incita a la empresa que venda este producto a desarrollar mejores versiones para entonces.

Estimar las ventas es, por supuesto, una tarea difícil. Estimamos que, contactando a distribuidores en toda Latinoamérica, se puede llegar a superar las 800 unidades a lo largo de toda la vida del producto. Esto implica tomar un 0,02% del mercado mundial, lo cual es una predicción razonable considerando que es un mercado con sólo 90 *players* en el mundo. Modelamos dichas ventas de acuerdo al siguiente gráfico:

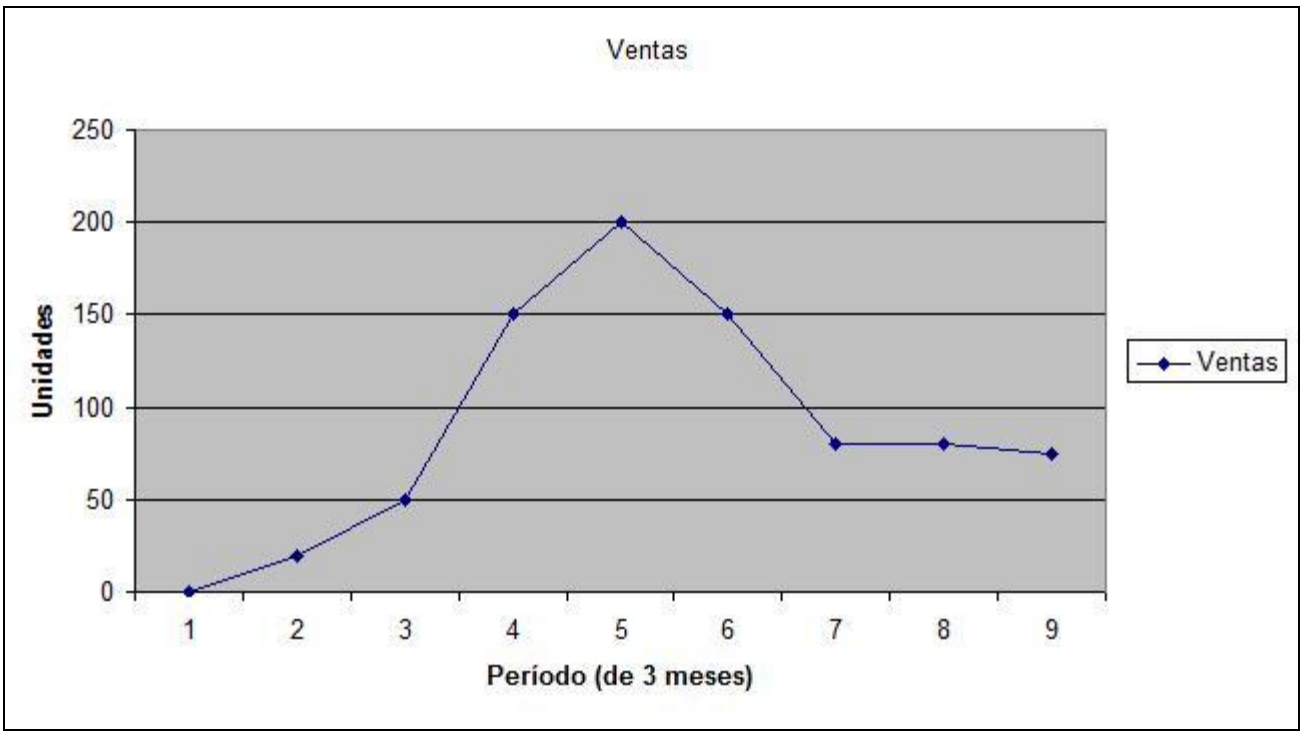


Fig 4.13: Modelado de ventas esperadas

Este modelo presenta un crecimiento acentuado a medida que se posiciona la marca en el mercado a lo largo del primer año, un decaimiento en el tercer semestre y un sostenimiento a niveles relativamente bajos para los últimos períodos a medida que concluye el ciclo de vida.

4.4.11 Resultados económicos esperados

A partir de los parámetros ya especificados, se obtiene el siguiente gráfico para el flujo por período y el flujo acumulado:

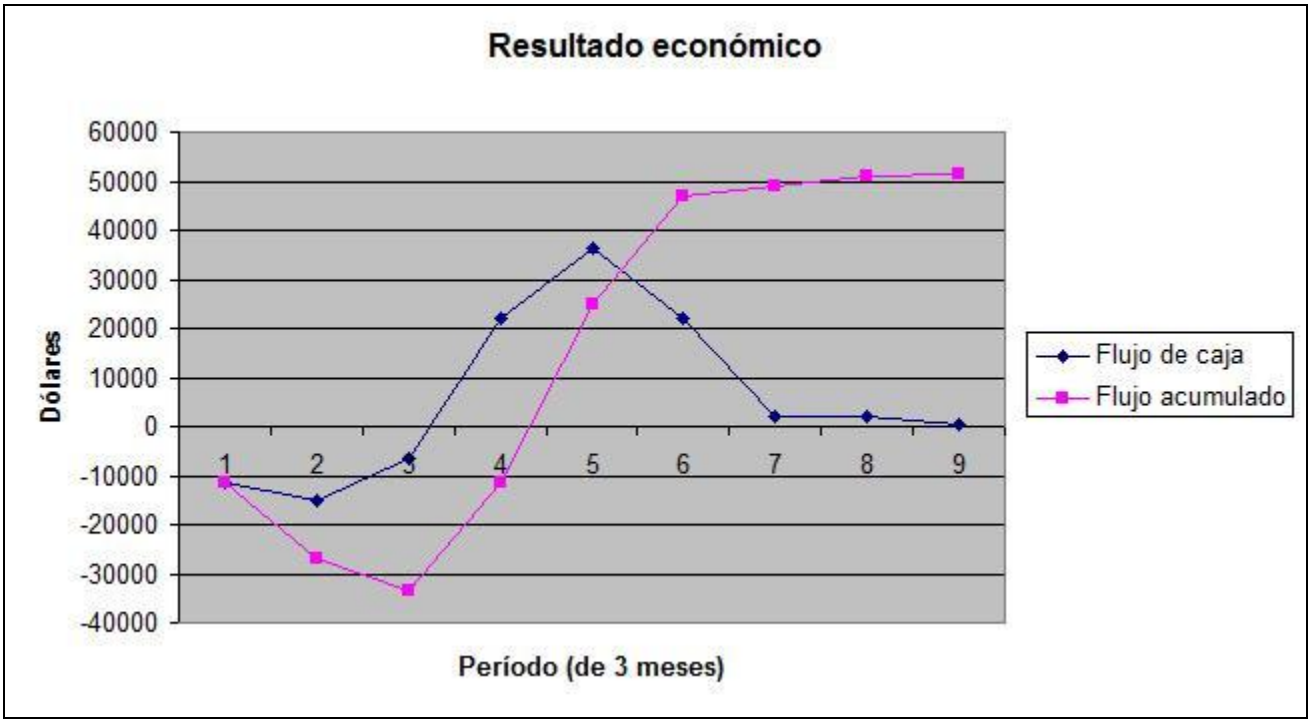


Fig 4.14: Resultado económico esperado

El gráfico muestra que el tiempo de retorno de inversión es de aproximadamente un año, y que el mayor déficit económico es de unos U$D 33400.

Finalmente, se calcula, usando una tasa de corte de 5% trimestral, un Valor Actual Neto de U$D37416 y una Tasa Interna de Retorno (es decir, la tasa de interés que llevaría el VAN a 0) de 33%.
Estos últimos resultados muestran que el proyecto es económicamente factible y probablemente atractivo para inversores.

V. INGENIERÍA DE HARDWARE

Según el Análisis de Factibilidad realizado en la sección 4 y la resolución de las Alternativas de Diseño expuesta previamente se construyó el siguiente diagrama general con las definiciones del Hardware a utilizar:

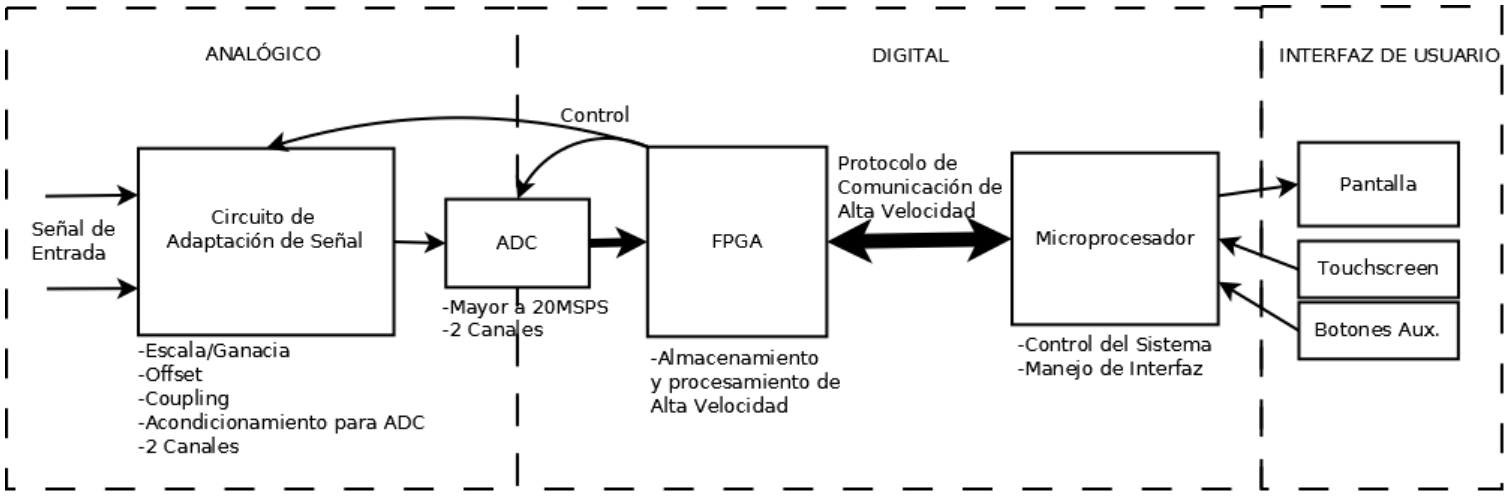


Fig. 5.1: Esquema de Hardware del Osciloscopio

El flujo de datos por canal se comporta de la siguiente manera: la señal analógica de entrada atraviesa un circuito de adaptación cuyos parámetros están directamente vinculados con los niveles de señal que el usuario desee ver. Dicha adaptación es necesaria para que en el bloque posterior, el convertidor Analógico-Digital (ADC) realice el muestreo de la señal. El intervalo constante de muestreo también tiene vínculo directo con la escala temporal que el usuario haya seleccionado. La FPGA recibe una a una las muestras obtenidas por el ADC, las procesa y envía a un *buffer* interno de almacenamiento. En una segunda instancia, el envío de datos desde la FPGA hacia el microprocesador se realiza por bloques de muestras. De esta forma, la primera se encarga de realizar el procesamiento de los datos en tiempo real (esto es, muestra a muestra, como por ejemplo la detección de la condición de *trigger*) y el microprocesador se ocupa del procesamiento de los datos por bloques relativos a las muestras en pantalla. Precisamente en este aspecto radica la ventaja de trabajar con una FPGA además de un microprocesador. Este último, además es el encargado de proveer una interfaz de usuario a través de los controladores de pantalla, *touchscreen* y una mínima cantidad de botones físicos auxiliares.

Se tomó como criterio que las señales de control que modifican el estado del Circuito de Adaptación de Señal y el ADC provengan de la FPGA, logrando así

aislar en mayor medida los bloques que intervienen en la adquisición de muestras inicial y el microprocesador quien se encarga del *front-end* del dispositivo.

5.1 Circuito de Adaptación de Señal

En el proceso de diseño se puso mayor énfasis en los bloques de los cuales depende la calidad del producto final. Evidentemente, todas y cada una de las partes son esenciales para que el sistema general funcione correctamente, pero eso no quita que el rendimiento total sea más sensible al diseño de algunos elementos particulares. Para poder focalizar el esfuerzo, primero se debe definir cuál es el criterio para medir la calidad del producto. Como es sabido, la calidad es el grado de cumplimiento de las especificaciones. Al tratarse de un osciloscopio, es decir un instrumento de medición, la cualidad más importante es poder medir de forma fiel y sin interferencias la señal de entrada. Aceptando este criterio, el bloque al que debe prestarse mayor atención a la hora del diseño es la etapa de entrada.

Lo primero que se debe definir son las funciones de este bloque. De forma general es la adaptación de la señal de entrada para poder ser digitalizada tomando en cuenta los parámetros establecidos por el usuario. Entrando más en detalle sobre las funciones de la etapa de entrada, se pueden mencionar las siguientes:

- **Impedancia de entrada**
- **Acoplamiento (de continua y de alterna)**
- ***Offset***
- **Ganancia (dependiendo de la escala)**
- **Acondicionamiento para ADC**

La siguiente pregunta es el orden en que estas etapas deben encontrarse en el recorrido de la señal. Durante el diseño se evaluaron distintas posibilidades, llegando finalmente al siguiente diagrama en bloques para la etapa de entrada.

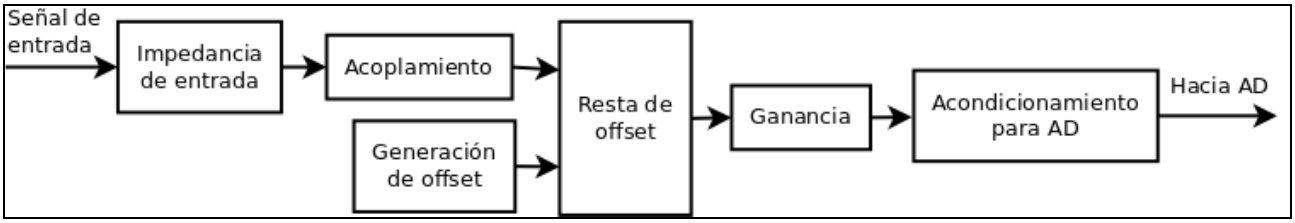


Fig. 5.2: Esquema de Bloques del Circuito de Adaptación de Señal

Es evidente que el primer bloque después de la punta del osciloscopio debe ser la impedancia de entrada y que el último antes del ADC debe ser el acondicionamiento para este. Lo que a primera vista no parece tan trivial es el orden de las tres etapas del medio. Cabe mencionar que el acoplamiento debe estar antes del offset en el camino de la señal, ya que de la otra forma, en caso de estar en acoplamiento de alterna, la continua que se le agrega a la señal en la etapa de offset será filtrada por el acoplamiento. Finalmente, lo que queda por definir es qué debe ir primero: la etapa de *offset* o la de ganancia. En una primera inspección uno podría pensar que el primer proceso debe ser aplicar la ganancia, logrando así normalizar la señal respecto a la escala y minimizando el rango dinámico de los niveles de *offset* que son necesarios generar internamente. Sin embargo, se decidió intercambiar este orden, teniendo en cuenta el incremento en el rango dinámico de los niveles de continua de la etapa de offset que esto implica. Esta decisión se debe a que esta es la única forma en que se logra ver con mayor detalle una sección reducida de una señal con amplitud excedente a la amplitud de tensión presente en pantalla. Si el orden fuese inverso la señal primero sería amplificada, con la saturación que esto conlleva, y luego se le sumaría el offset a la señal ya saturada. La decisión de diseño permite que primero se agregue el offset deseado a la señal, centrando la misma en la sección de niveles de tensión requeridos para que luego sea amplificada y que la saturación no forme parte de los niveles de señal que se visualizan en pantalla.

Teniendo en cuenta el orden de los bloques mencionado anteriormente, se diseñó cada etapa en detalle velando por la integridad de la señal en todo el camino.

5.1.1 Impedancia de entrada y acoplamiento

En el primer acercamiento al diseño de la etapa de entrada, se pensó a la impedancia de entrada y al acoplamiento como dos bloques completamente independientes. Luego se llegó a la conclusión que al fusionar estas etapas se puede aprovechar mejor las cualidades de cada una sin la necesidad de agregar componentes externos. Esto se pondrá en evidencia más adelante en esta sección. Primero se analizan las funcionalidades por separado y luego se presenta la solución implementada.

La impedancia de entrada de los osciloscopios debe ser alta y acotada. Debe ser alta para no cargar el circuito a medir y no atenuar innecesariamente la señal, con la perdida de rango dinámico que esto representa. Debe ser acotada para

saber en qué medida se dan estos dos fenómenos y poder controlarlos con el diseño de las puntas.

Para decidir qué valor de impedancia de entrada se va a utilizar, se valorizó la versatilidad del producto con respecto a las puntas de osciloscopio mayormente ofrecidas en el mercado. Se llegó a la conclusión de que un valor estándar, como 1 MΩ en paralelo con 20 picofaradios, es la mejor opción. Es decir que el circuito de esta etapa es el siguiente.

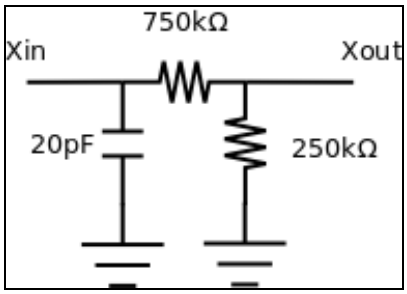


Fig. 5.3: Circuito esquemático de la impedancia de entrada

Como se ve en la figura anterior, si hacia *Xout* hay alta impedancia, la impedancia de entrada del sistema esta impuesta únicamente por esta etapa.

Además de imponer la impedancia de entrada, esta etapa atenúa la señal de entrada 4 veces. Esto se hace de forma intencional, ya que por las especificaciones, la mayor tensión de entrada es de *300Vrms*, es decir *424Vp*, que si se utiliza una punta *x10* son *42.4Vp* en la entrada del osciloscopio. Si se atenúa 4 veces se obtiene una señal de *10.6Vp*, que es menor, tomando una guarda razonable, a las *+12V* y *-12V* que se van a utilizar para alimentar toda la etapa de entrada.

En cualquier osciloscopio comercial existen dos formas de acoplar la señal: en continua o en alterna. En acoplamiento continuo, la señal atraviesa esta etapa sin sufrir modificaciones. En alterna, la componente de continua que presenta la señal de entrada es filtrada completamente, dejando pasar únicamente las componentes de la señal de mayor frecuencia.

Para lograr este fin, la señal debe pasar por un cable o por un capacitor en serie. En el segundo caso, si la siguiente etapa presenta una alta impedancia de entrada, la señal puede perder su referencia introduciendo ruido innecesariamente. Por otro lado, si se agrega una resistencia a tierra, esta modifica la impedancia de entrada y esto no es aceptable. Finalmente, lo que se decidió fue dividir la resistencia a masa de la etapa de impedancia de entrada en

dos del doble en paralelo y agregar entre ellas la etapa de acoplamiento. En conclusión, el circuito que se obtiene es el siguiente:

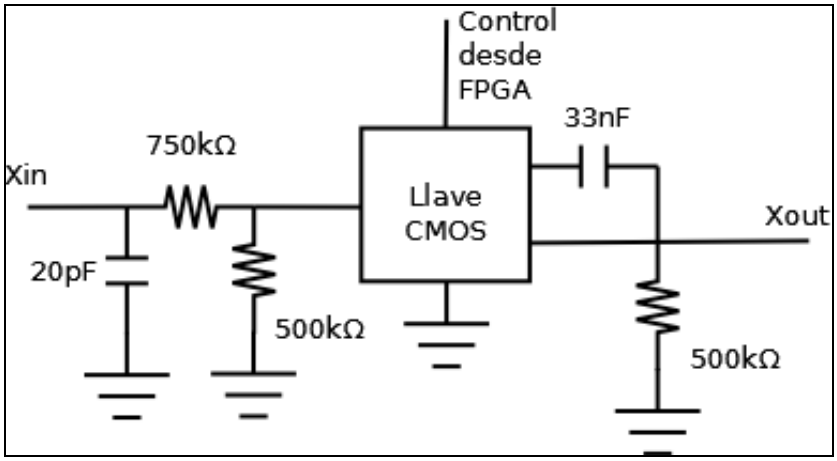


Fig. 5.4: Circuito esquemático de la impedancia de entrada y el acoplamiento

Como se ve en la figura anterior, esta etapa únicamente consiste en una llave *CMOS* de gran ancho de banda. En un caso la señal pasa limpia al otro lado, mientras que en el otro la misma es filtrada al pasar por el capacitor. El capacitor se tomó de 33*nF*, ya que de esta forma se obtiene una frecuencia de corte de 9.65Hz, la que normalmente es 10Hz en los osciloscopios comerciales. La decisión del acople deseado se controla desde la FPGA, ya que el acoplamiento depende de lo que decida el usuario y debe poder cambiarse de forma dinámica. Se decidió utilizar una llave *CMOS* en vez de un *relay* o *photorelay* ya que se consideró prioritario garantizar un ancho de banda con margen que no distorsione la señal. A su vez, en esta etapa no se requiere un manejo de corrientes exigentes y frente al manejo de los *relays*, las llaves CMOS se controlan de forma más simple (los relés requieren circuito de protección frente a disparos de corriente de la bobina). La llave *CMOS* que se utilizó es la MAX4659 que cuenta con un ancho de banda de 250MHz y una resistencia de encendido de sólo 25Ω.

5.1.2 Generación de offset

Por ahora siempre se especificó que se debe poder restar un *offset* dinámicamente variable a la señal, sin embargo nunca se aclaró como generar este nivel de continua. Como el sistema es controlado de forma digital, no es trivial obtener el *offset* deseado. Este bloque es el que se encarga de esta tarea.

Se llegó a la conclusión de que la manera más sencilla, salvando que se deba incorporar un convertidor adicional, es utilizar un PWM para este fin.

Las dos problemáticas principales de este método surgen de cómo obtener un PWM con niveles y rango dinámico deseados y cómo este debe ser filtrado para obtener el nivel de continua. El circuito que se implementó para solucionar estas problemáticas es el siguiente:

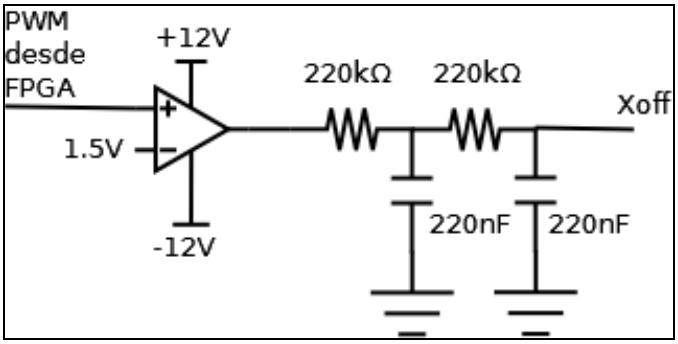


Fig. 5.5: Circuito esquemático del bloque de generación de offset

Para empezar, cabe aclarar que el control del PWM lo realiza la FPGA, ya que depende del nivel de continua que el usuario desee y debe poder cambiarse de forma dinámica. Teniendo en cuenta que las escalas del osciloscopio van desde 5mV/div hasta 10V/div y sabiendo que para cuando se ajuste el offset la señal se encuentra atenuada 4 veces se propuso que el PWM debe ir desde -12V a +12V y debe estar discretizado cada 150µV. A partir de estas premisas se calculó el rango dinámico del PWM (aproximadamente 110dB) y se llegó a la conclusión que se puede codificar con 18 bits. Sabiendo que la FPGA va a trabajar con aproximadamente 50MHz y que el PWM se implementa con un contador de 18bits, la frecuencia del PWM será 190Hz. Para obtener el nivel deseado del PWM se utilizó un comparador con una de las entradas con una referencia de tensión en el punto medio de la salida digital de la FPGA que es de +3.3V. A su vez, este comparador se alimenta con una tensión bipolar de 12V.

Luego, para filtrar el PWM se utiliza un filtro de segundo orden. Esto se decidió considerando que si fuese de primer orden la frecuencia de corte del filtro debería estar aproximadamente 5 décadas debajo de la frecuencia del PWM (para asegurar una atenuación del primer armónico de 100dB) y esto impondría una respuesta muy lenta para cambios en el nivel de continua. Por otro lado, utilizando un filtro de segundo orden la frecuencia de corte solo debe estar por debajo de los 5.79Hz, lo cual ya proporciona una velocidad de respuesta aceptable. Una alternativa para implementar este filtro es una celda activa pasabajos de segundo orden. Sin embargo, tomando provecho de que no se necesita una baja impedancia de salida (ya que hacia *Xoff* hay alta impedancia) y

tampoco una banda pasante plana, se puede implementar usando dos filtros pasivos RC.

5.1.3 Resta de offset

Una vez generado el *offset* deseado, es necesario poder sustraerlo de la señal. Además, hay que tener en cuenta que esta es la primera etapa activa que se encuentra en el camino de la señal y que por lo tanto es elemental para fijar la figura de ruido del sistema completo. Teniendo esto en cuenta se decidió implementar este bloque con un amplificador de instrumentación, tomando provecho de su gran inmunidad al ruido debido a su rechazo al modo común. Se evaluó la posibilidad de diseñar e implementar este amplificador diferencial de forma discreta para poder tener mayor flexibilidad y control sobre el mismo. Sin embargo, se llegó a la conclusión de que utilizar uno integrado asegura una mayor inmunidad al ruido y menor distorsión, ya que se minimizan las influencias de componentes parásitos indeseados.

El siguiente paso en el diseño es elegir el amplificador de instrumentación a utilizar. Se decidió trabajar con el *AD8253* de *Analog Devices*, ya que cumple con todos los requerimientos: posee gran ancho de banda (*10Mhz* con ganancia unitaria), alto rechazo de modo común (100dB), bajo ruido (45 nV/√H), consume potencia reducida (4.5mA) y permite ser alimentado con ±12V.

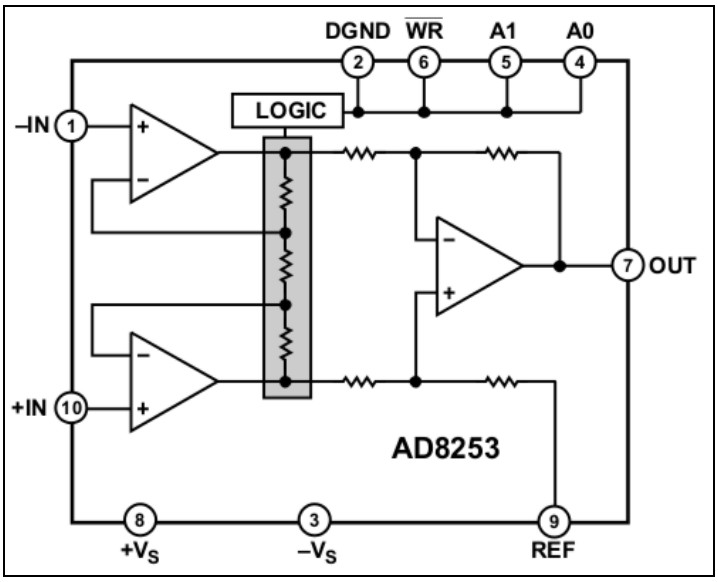


Fig. 5.6: Esquema del Amplificador de Instrumentación AD8253 (hoja de datos del fabricante)

Este amplificador es de ganancia programable, sin embargo se decidió mantener una ganancia unitaria constante y delegar la amplificación a otra etapa, ya que de otra manera no se cumple con la especificación del ancho de banda (el producto Ganancia-Ancho de Banda resulta un limitante).

5.1.4 Ganancia

La ganancia que se aplica a la señal depende de la escala que seleccionó el usuario y debe ser variable dinámicamente. Por este motivo se debe poder controlar la ganancia del amplificador de esta etapa desde la FPGA. Por otro lado, debido a que las escalas son discretas entre 5mV/div y 10V/div, sólo se necesitan 11 ganancias distintas. La primera idea fue implementarlo multiplexando los caminos que recorre la señal, utilizando distintos valores de resistencias en cada uno de ellos. Sin embargo, este método es lento, introduce ruido y mayor distorsión y depende en gran medida de la precisión de las resistencias. Por lo tanto se llegó a la conclusión de que resulta conveniente utilizar un amplificador de ganancia programable (*PGA*).

Como el osciloscopio debe tener un ancho de banda de 10MHz para todas las escalas, es necesario que para todas las ganancias (que en esta etapa van desde -20dB a 44dB) se conserve xesta especificación. Esto significa que si se quisiese utilizar un único amplificador, este debería tener un producto ganancia ancho de banda demasiado grande, lo cual es difícil de conseguir y tiene gran impacto sobre el precio. Es por eso que se decidió implementar esta etapa con 2 PGA en cascada. Esto, además de reducir la exigencia sobre cada uno de los mismos, aumenta la flexibilidad de la ganancia total.

Finalmente, resta decidir con qué PGA se va a trabajar. Se eligió el amplificador *THS7002* **de *Texas Instruments*, ya que cumple con todos los requisitos: ganancia programable (de -22dB a 20dB con pasos de 6dB), gran ancho de banda (70MHz), bajo ruido (500 nV/√Hz en el peor de los casos), posibilidad de alimentación con ±12V, protección de saturación de salida y posibilidad de apagar el amplificador para ahorrar energía. Además, estos integrados son duales, es decir que cada uno contiene dos PGA, lo que reduce la cantidad de integrados a utilizar y las conexiones necesarias.**

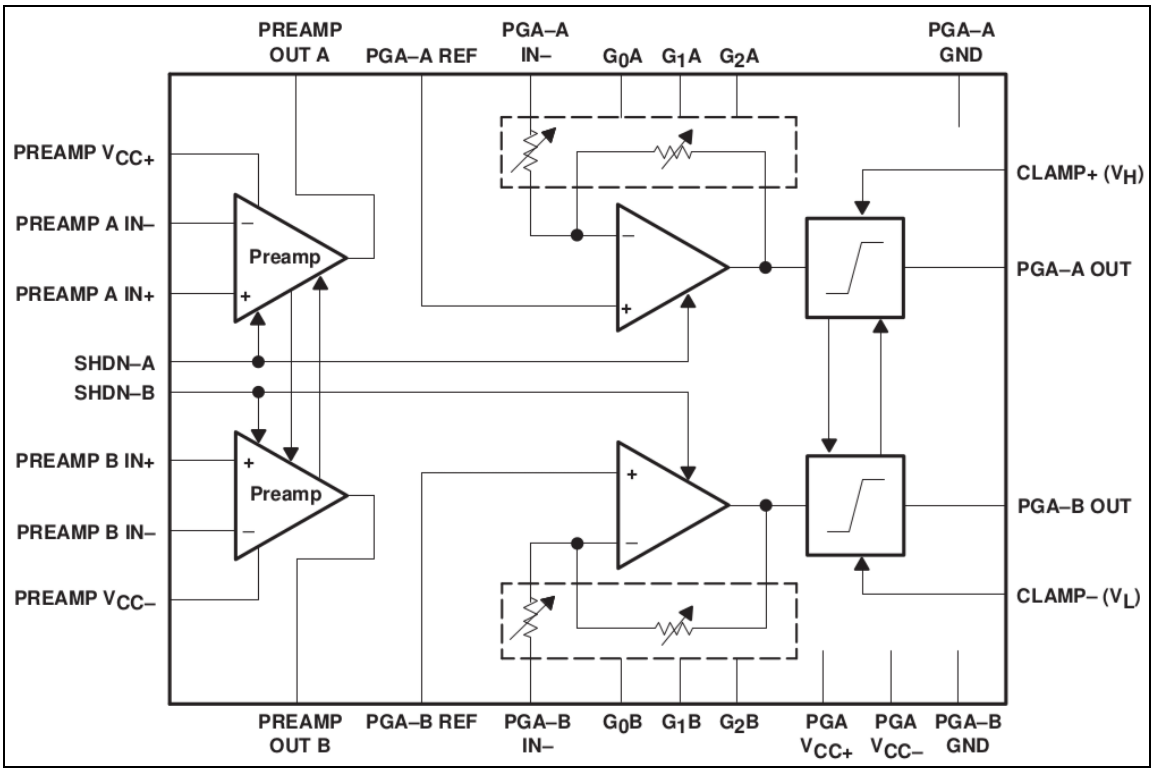


Fig. 5.7: Esquema del PGA THS7002 (hoja de datos del fabricante)

Utilizando 2 PGA en cascada es posible obtener todas las ganancias necesarias de manera casi exacta. De todas formas, más adelante se demostrará que el ajuste fino de dichas ganancias se realiza en la FPGA. Esto no reduce el rango dinámico ya que, como se explicará posteriormente, se utilizan 10 bits para la conversión analógica digital de los cuales solo 8 bits son significativos.

5.1.5 Acondicionamiento para ADC

En el primer bloque de la etapa de entrada la señal fue atenuada 4 veces, asegurando que se encuentre entre -10.6V y +10.6V. Luego de agregarle el nivel de continua deseado, la etapa de ganancia dependiendo de la escala seleccionada, coloca la señal entre -1V y +1V, ya que el AD elegido tiene un rango máximo de 2Vpp. Es decir, que hasta ahora la señal fue bipolar para aprovechar al máximo el rango de los amplificadores de las etapas anteriores. Sin embargo, la señal en esta instancia no es compatible con la entrada diferencial del ADC. A su vez, los rangos de tensión admitidos por dicho componente para cada entrada diferencial se encuentran entre +0.5V y +1.5V (un total de 1Vpp por entrada). Por lo tanto, es necesario realizar una última adaptación de la señal. Esto se realizó

utilizando amplificadores diferenciales recomendados por el fabricante del ADC que se utilizará:

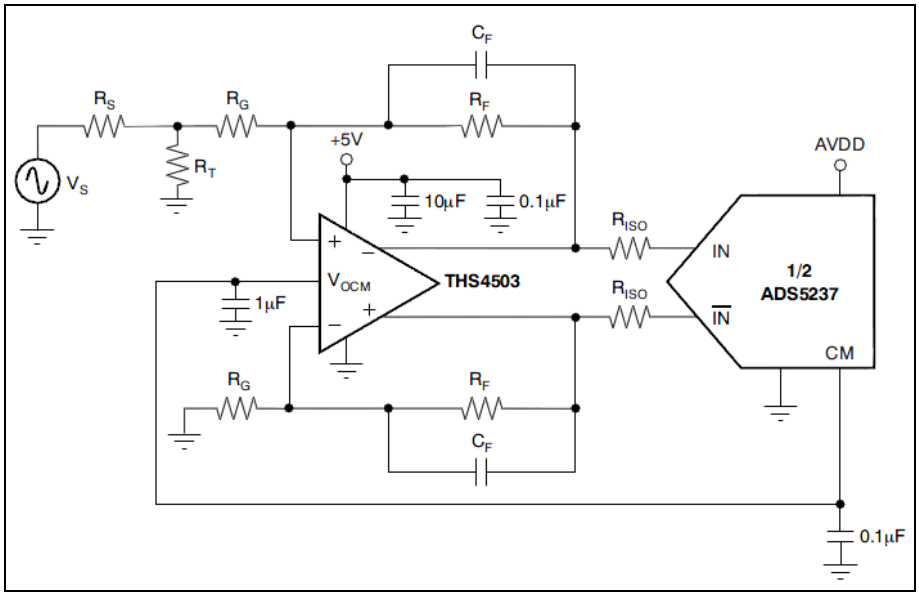


Fig. 5.8: *Application Circuit* del amplificador diferencial THS4502 (hoja de datos del fabricante)

El amplificador cumple las dos funciones principales mencionadas anteriormente. Por un lado realiza una nivelación de la señal agregándole un valor de continua de +1.5V. Este valor de referencia proviene del mismo ADC tal y como se indica en el diagrama. Por otro lado, transforma la señal a diferencial logrando que cada salida tenga una amplitud de 1Vpp y que la resta de ellas, al ser complementarias, recupere los 2Vpp de la señal amplificada. El integrado que se utilizará es el THS4502, que más allá de cumplir con las funcionalidades descriptas, posee la posibilidad de entrar en modo de ahorro de energía.

5.1.6 Alimentación

Todo el circuito de adaptación de la señal se alimenta con +12V y -12V, ya que debe manipular señales analógicas de diferentes amplitudes y niveles de continua. Como la etapa de entrada es la única que presenta estas características, también es la única que necesita estos niveles de alimentación. Por esta razón es necesario agregar al diseño de esta etapa fuentes capaces de adquirir las tensiones deseadas a partir de 3.3V.

Para esta tarea se utilizaron fuentes de *switching* integradas debido a su gran eficiencia. Se eligieron las fuentes MAX765 y MAX734 de Maxim para generar los -12V y +12V respectivamente, ya que son fuentes pequeñas que entregan

suficiente potencia como para alimentar toda la etapa de entrada. Los circuitos de alimentación se tomaron de las hojas de datos de estas.

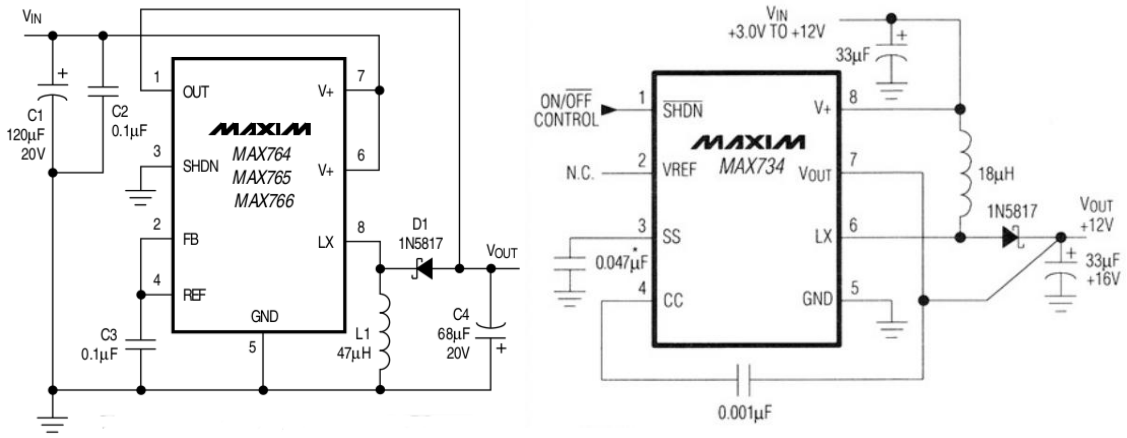


Fig. 5.9: Esquema de las fuentes MAX765 y MAX734 (hoja de datos del fabricante)

Con el objetivo de mitigar el efecto del *ripple* de tensión que es propio de las fuentes *switching*, se tomó la consideración de hacer coincidir la corriente nominal de las mismas con el consumo de corriente esperado de la placa de entrada.

5.1.7 Plan de Pruebas

Para la comprobación del funcionamiento del Circuito de Adaptación de Señal en primera instancia se realizarán simulaciones con modelos de los componentes que comprueben el funcionamiento del mismo. En segunda instancia, se realizará una placa de prueba con el pedido de componentes correspondiente para comprobar de forma práctica el funcionamiento de uno de los canales. Gracias a que el diseño realizado se conforma principalmente de circuitos integrados, las distintas secciones de esta etapa analógica poseen de forma individual sus hojas de datos y recomendaciones de ruteo y conexionado. Esto permite la comprobación del sistema más ágil y de forma conjunta.

5.2 ADC

El proceso de elección del convertidor del bloque ADC se reduce al análisis de especificaciones identificando aquellos ítems que determinan las características con las que el mismo debe cumplir:

- Frecuencia de muestreo mayor o igual a 50 Msps: **la velocidad de muestreo del convertidor es uno de los factores más cruciales en cuanto a la variación de precio de los mismos por lo que dicho valor no podrá ser muy holgado.**

- Resolución mayor o igual a 8 bits: **Este es el segundo factor crítico en cuanto a la variación de precio dado que junto con la frecuencia de muestreo infiere directamente en la tecnología y arquitectura del convertidor.**

- Ancho de banda: **esta característica debe ser coherente con la frecuencia de muestreo para que la señal no sufra distorsión.**

- 2 Canales: **es de gran utilidad hallar un mismo circuito integrado que contemple los dos canales para simplificar la comunicación y reducir el espacio necesario.**

- Consumo: **para contribuir con la reducción de consumo del sistema en su totalidad es de gran utilidad que el convertidor posea la opción de reducir su consumo si alguno o ambos canales se encuentran sin utilización.**

En función de la velocidad requerida de muestreo y la elección de la FPGA como receptor de muestras, se concluye que la salida del ADC debe ser en paralela **tanto para cada uno de los canales como para los bits de dichos canales. Esto se define en parte porque una comunicación serie conlleva una frecuencia necesariamente más elevada y por otro lado porque la FPGA utilizada provee de una cantidad de pines de entrada coherentes con las necesidades para establecer dicha comunicación.**

Para concluir con la elección de un convertidor fueron grandes determinantes ciertos aspectos que no tienen que ver con las especificaciones, sino más bien con la factibilidad de realización de las pruebas del mismo. En este sentido se valoró: la disponibilidad de stock, la documentación provista por el fabricante, el precio de envío y la posibilidad de conseguir muestras gratuitas del mismo para utilizar en la etapa de desarrollo. Por estos factores se acudió a la empresa *Texas Instruments* y se seleccionó el convertidor *ADS5237* **con las siguientes características de relevancia:**

	ADS5237
Resolución (*bits*)	10
Frecuencia de Muestreo máx. (*MSPS*)	65
Canales de Entrada	2
SNR (*dB*): Relación señal a ruido a 5MHz	61.7

SFDR (*dB*): Rango dinámico libre de ruido espurio a 5MHz	
Consumo Típico (*mW*)	330
Interface de Salida de Datos	Paralela CMOS
Interface de comunicación de configuración	Serial SPI
Alimentación Analógica Mín. (*V*)	3.0
Alimentación Analógica Máx. (*V*)	3.6
Alimentación Digital Mín. (*V*)	3.0
Alimentación Digital Máx. (*V*)	3.6
Arquitectura	Pipeline
INL Máx. (*+/-bit menos significativo*): Máxima no-linealidad Integrada	1
SINAD (*dB*): Relación señal a ruido y distorsión a 5MHz	61.6
ENOB (*bits*): Número efectivo de bits	9.9
Rango de Entrada (*Vpp*)	2
Tipo de referencia	Interna y Externa
DNL Máx. (*+/- bit menos significativo*): Máxima no-linealidad diferencial	0.5
Ancho de Banda de Entrada (*MHz*)	300
Pines/Package	64 TQFP
Precio Aproximado por unidad (*U$S*) en 1000 unidades	8.65

La resolución del convertidor seleccionado excede los 8 bits de manera tal que el ENOB también se ubique cómodamente por arriba de dicha resolución. Esto nos permite realizar correcciones necesarias de ganancia sobre las muestras e incluso poder además despreciar la distorsión y el ruido introducido por este componente. A su vez, la configuración de *hardware* de esta sección de muestreo no resulta un limitante para el aumento de la cantidad de niveles a mostrar en pantalla (lo que sólo tendría sentido si se incrementa la resolución de pixeles de la misma) si así se deseara proceder con un producto a futuro.

Como se verá más adelante, el plan de pruebas del ADC se realizará en conjunto con el plan de pruebas de la FPGA.

5.3 FPGA

La FPGA cumple dos funciones principales: provee el *hardware* necesario para el sistema de *trigger* o disparo del osciloscopio e incorpora el sistema de almacenamiento de las muestras recogidas. Como se observa en el diagrama del sistema completo, la FPGA es una etapa intermedia entre el bloque del microprocesador y el convertidor ADC por lo que es la encargada de realizar el procesamiento de los datos en tiempo real (muestra a muestra).

En cuanto a la elección de la FPGA a utilizarse, se decidió trabajar a partir de la placa de desarrollo EVM CYCLONE provista por la cátedra de Laboratorio de Microprocesadores. El principal motivo de esta elección fue el hecho de que se cuenta con experiencia previa de trabajo en este entorno. La placa EVM utiliza una FPGA *Cyclone I* (*EP1C6T144C6*). Para un diseño final propio resultaría de interés fabricar un PCB que incluya tanto el *Cyclone*, el ADC y la etapa de entrada de la señal.

El esquema interno realizado en la FPGA se muestra a continuación:

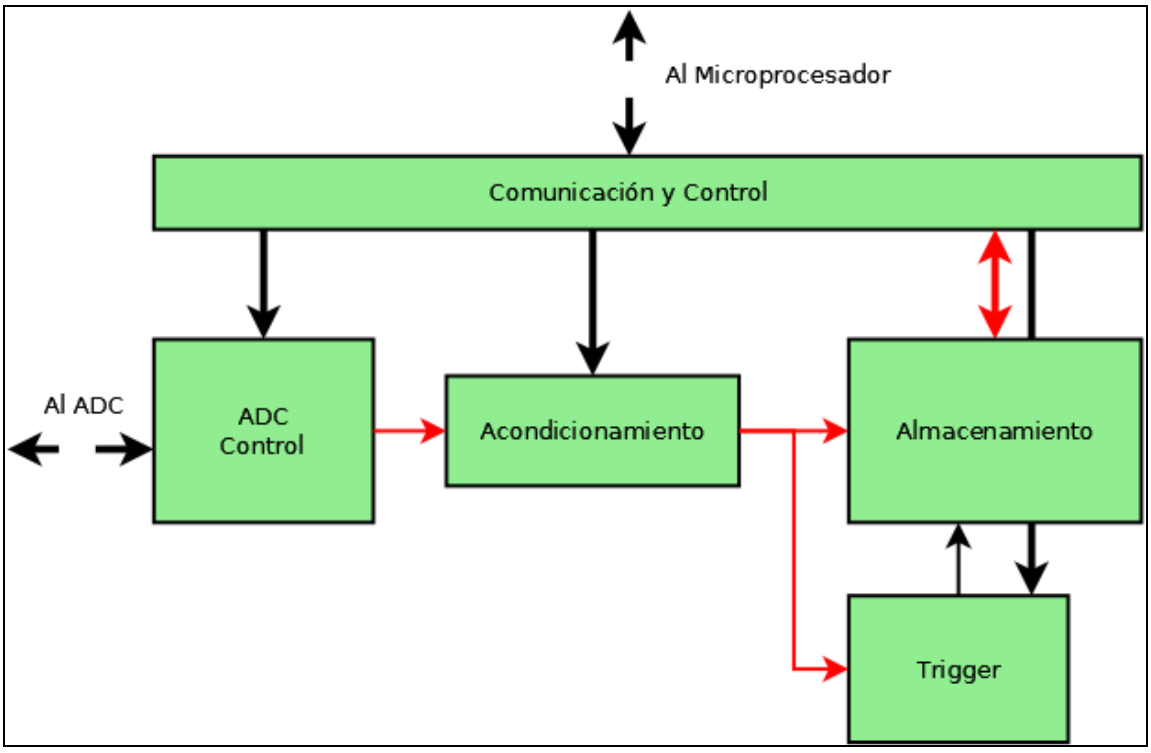


Fig. 5.10: Esquema por bloques de la FPGA

5.3.1 Bloque de ADC Control

Como se ve, tenemos un bloque que realiza la interface con el convertidor AD. Este bloque cumple dos funciones. Por un lado, toma la muestra de salida del ADC de 10 bits en paralelo y la envía a los bloques de *trigge* y almacenamiento. Por el otro lado, interactúa con el convertidor AD a través de una comunicación serie, enviando comandos de control que permiten variar los parámetros de la conversión en tiempo real. Los comandos realizan lo siguiente:

-Inicialización del ADC.

-Cambio del estado de muestreo de los canales.

-Fijar la frecuencia de muestreo del ADC.

Se debe mencionar que la ejecución de estos mensajes se encuentra controlada a su vez por el bloque de Comunicación y Control de la FPGA.

5.3.2 Bloque de Acondicionamiento

Previo al almacenamiento de las muestras, este bloque las siguientes 3 tareas fundamentales:

- Decimación: **La frecuencia de muestreo del ADC se encuentra fijada en 50MSPS por lo que este bloque se encarga de resolver la frecuencia de muestreo aparente según la escala de tiempo que se quiere visualizar.**

- Promedio de muestras: **Para no descartar las muestras excedentes luego de la decimación, se implementó un promedio de las mismas para mejorar la resolución aparente del osciloscopio. El tamaño de este promedio es variable de acuerdo a la frecuencia de muestreo y dado que la implementación de divisores con números arbitrarios en compuertas lógicas requiere un gran excedente de recursos, se tomó la consideración de restringir dicho tamaño a la potencia de 2 más cercana.**

- Corrección de ganancia: **Dado que los amplificadores utilizados en la etapa de entrada no logran introducir la ganancia deseada con precisión, en esta etapa se realiza la corrección fina de todas las ganancias deseadas.**

5.3.3 Bloque de *Trigger*

En el bloque de *trigger* se encuentra el hardware responsable del disparo de la base temporal del osciloscopio. Los modos de disparo del osciloscopio son 3:

- Modo Normal**: este modo almacena muestras hasta hallar un cruce (con las opciones de flanco positivo y/o flanco negativo) por el nivel de señal especificado por el usuario. El esquema que debe seguir este bloque puede ser representado por una máquina de estados, como se mostrará más adelante.**

- Modo Automático**: este modo es básicamente una ejecución de disparo forzada cuando no se cumple con la condición esperada en el Modo Normal. La implementación realizada responde a dicho funcionamiento dado que el bloque no cumple con la condición de cruce, la FPGA ignora el nivel de disparo seleccionado por el usuario y se encarga de completar la totalidad de la profundidad de memoria y solo se detiene cuando lo ha logrado. Todas las muestras almacenadas son enviadas al Microprocesador cuando este las solicita.**

- Modo Roll**: este caso tampoco depende de un nivel seleccionado por el usuario sino que todas las muestras recibidas, almacenadas en tiempo real, son enviadas hacia la pantalla. Se debe tomar en consideración que para la realización de este modo, la frecuencia de muestreo es limitada fuertemente dado que el refresco de pantalla en coherencia con la**

velocidad de respuesta de la visión humana, no permite que se analice en tiempo real muestras de una frecuencia de muestreo elevada. En el Modo Roll, la escala temporal podrá representar como mínimo 1 segundo en la totalidad de la pantalla.

Además de atender a estos 3 modos de disparo, el bloque de *Trigger* posee las siguientes opciones:

- **Posibilidad de disparo con cualquiera de los *2 canales***
- **Posibilidad de disparo con *flanco ascendente* o *flanco descendente*.**
- ***Tiempo de Hold-Off*: implementado con un contador variable que inhabilita el cumplimiento de la condición de disparo por un determinado tiempo.**
- ***High-Frequency Reject*: implementado con un filtro IIR pasabajos constante que puede ser tomado como entrada en la comparación con el nivel de disparo.**
- ***Noise Reject*: filtro pasaaltos implementado con un filtro IIR.**

A continuación se muestra una primera implementación en la FPGA del sistema de *trigger*. La implementación está basada en una máquina de estados sencilla y un comparador. A partir de las señales TEQ (indica que la señal de entrada es igual al umbral de *trigger*), y TLT (indica que la señal es mayor al umbral), la máquina de estados produce a la salida un pulso cuando se cumple la condición de disparo (es decir, para disparo con flanco positivo, la señal de entrada venia por debajo del umbral y lo cruza).

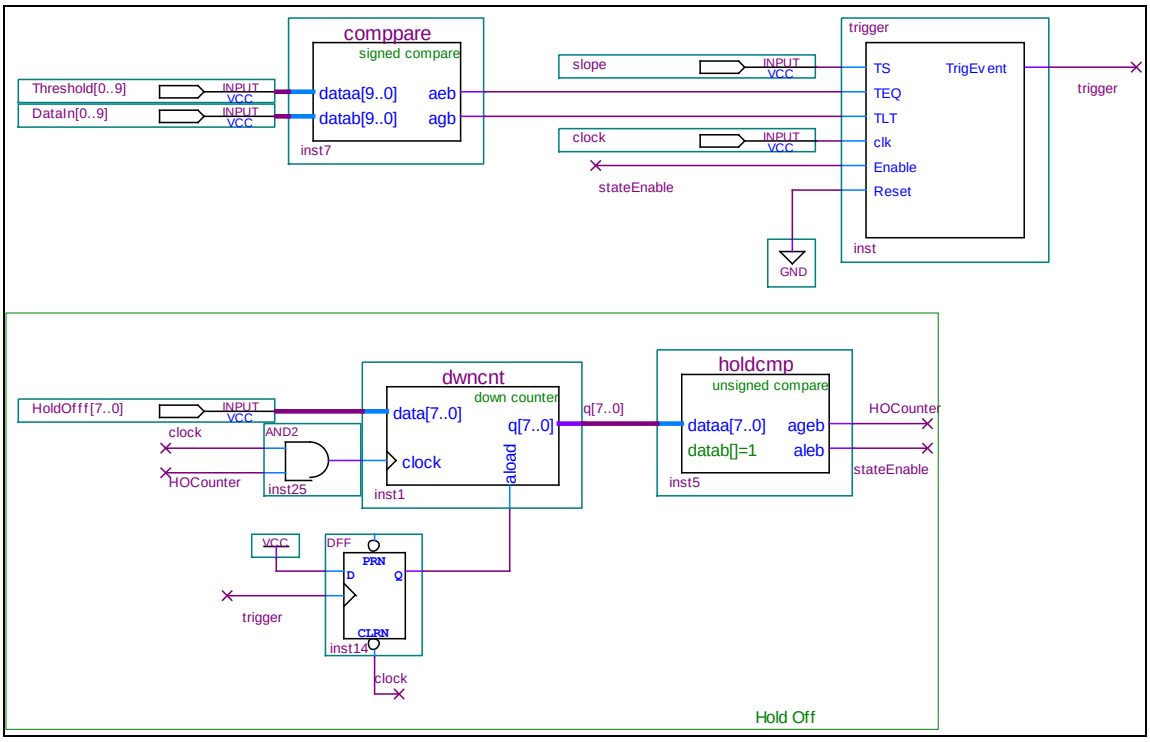


Fig 5.11: Implementación del trigger en FPGA

Se observa también la implementación del sistema de *hold-off*. Cada vez que ocurre un evento de disparo de *trigger* se pone en marcha un contador. Mientras este sea distinto de cero, el sistema de *hold-off* inhabilita a la máquina de estados. Es decir, mientras que inhibe la aparición de nuevos eventos, a pesar de que se cumpla la condición de disparo.

5.3.4 Bloque de Almacenamiento

El bloque de almacenamiento recoge las muestras que ingresan a la FPGA, y las coloca en el *buffer* de salida para transmitirlas al microprocesador. La implementación tiene como componente principal un *buffer* circular de 4096 muestras por canal (acorde a las especificaciones del producto). Sin embargo, no es trivial analizar la dependencia del estado de almacenamiento según los distintos modos de disparo. En el Modo Roll, las muestras son almacenadas continuamente y es el Microprocesador quién se encarga de solicitarlas de forma ininterrumpida. La FPGA sólo debe tener memoria de cuál fue la última muestra enviada para así completar el envío siguiente con las muestras nuevas. Por el contrario, tanto en el Modo Normal como en el Modo Automático el procedimiento de este bloque se encuentra representado por el siguiente esquema:

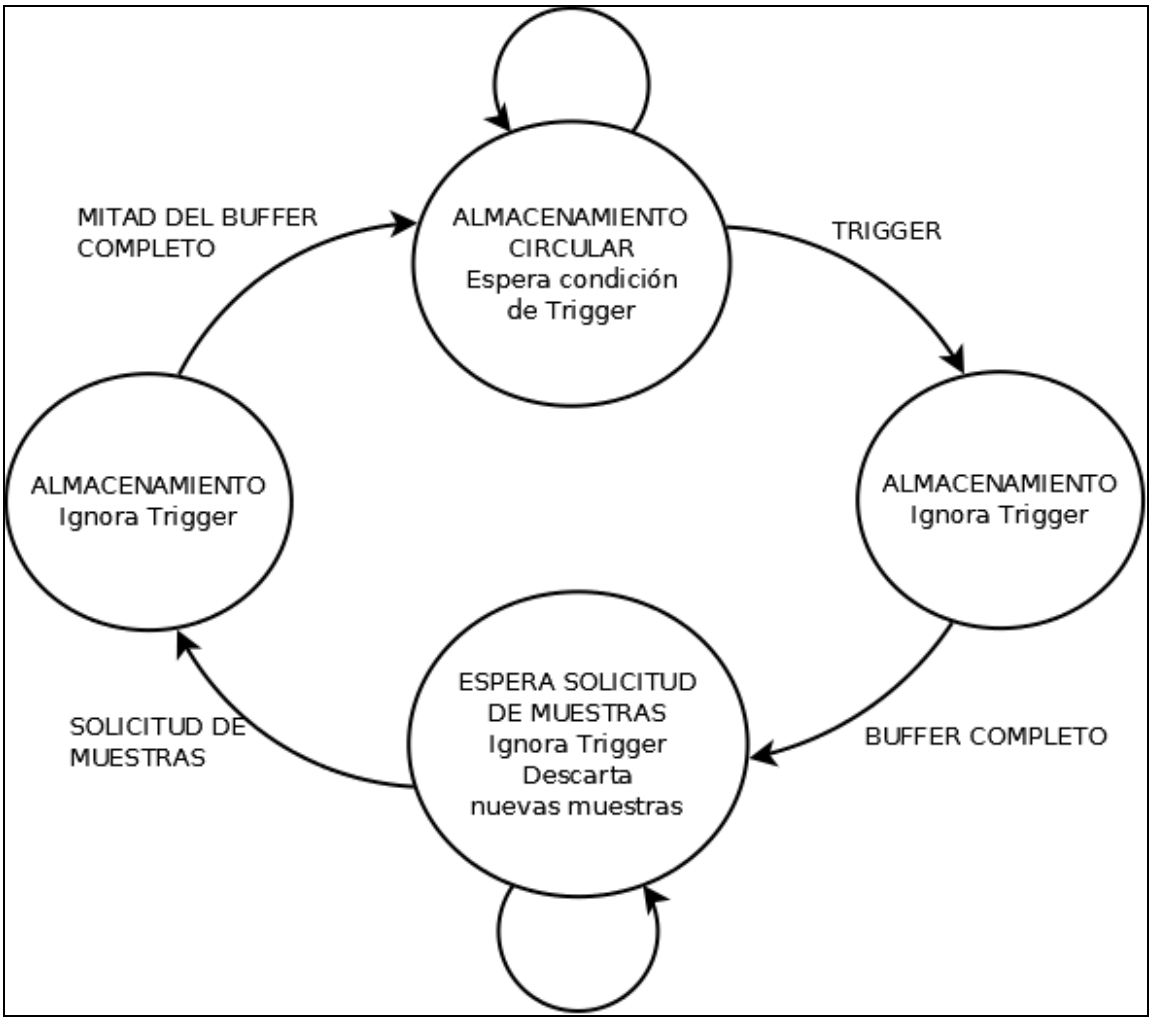


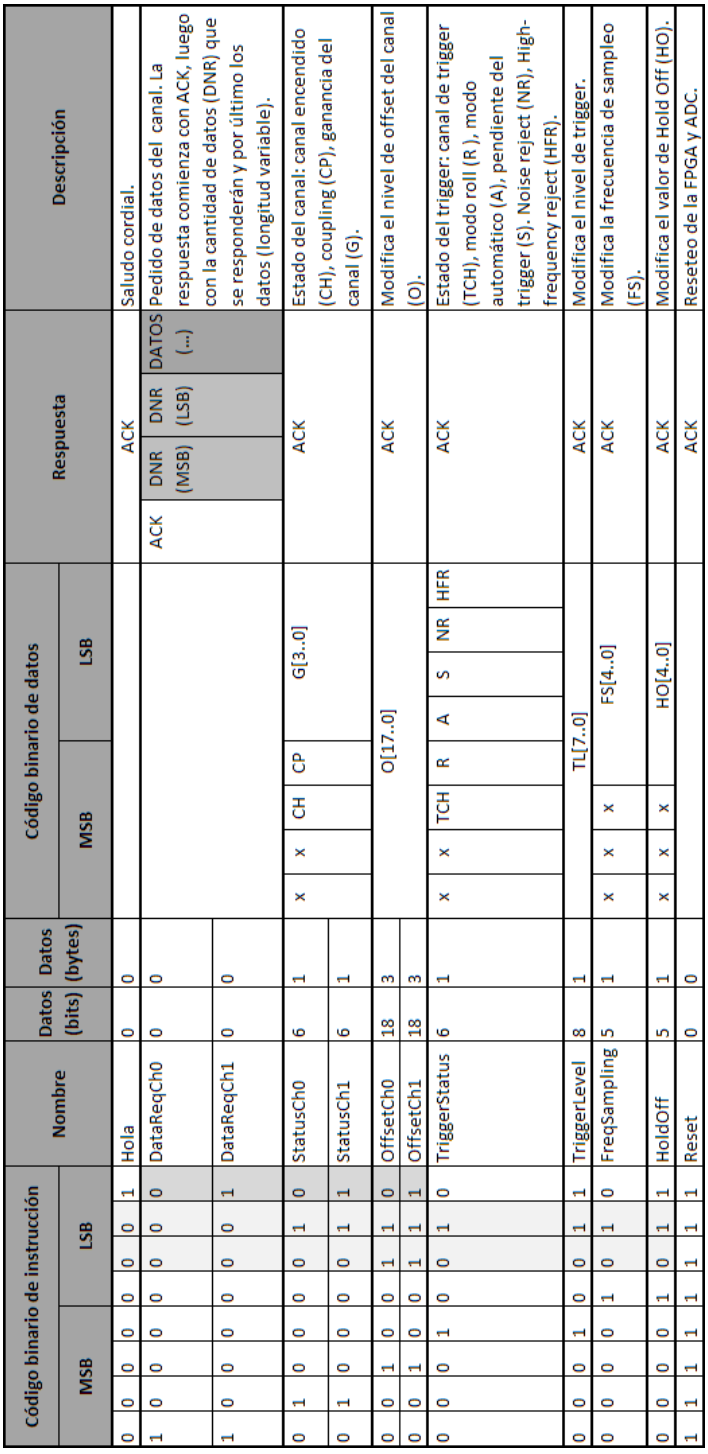
Fig. 5.12: Máquina de estados del Bloque de Almacenamiento de la FPGA

Si partimos del estado de Almacenamiento Circular, el bloque se encontraría esperando una condición de disparo. Una vez que la obtiene, solo restará llenar el restante %50 del *buffer* para lograr que la cantidad de muestras antes y después del disparo sea simétrica. Este estado concluye al completar el *buffer* y se pasa al estado de espera, en donde las muestras son ignoradas. Esta situación de descarte de muestras es necesaria dado que, como fue mencionado anteriormente, el refresco de pantalla puede no ser lo suficientemente rápido para actualizar las muestras actuales con el mismo ritmo con el que se cumple la condición de disparo. Una vez que el Microprocesador solicita las muestras, es necesario seguir ignorado la condición de *trigger*. Esto se debe a que toda la profundidad del *buffer* será enviada al Microprocesador para luego ser desplegada en pantalla y es necesario que al menos la mitad del buffer se encuentre completo con nuevas muestras para poder atender los nuevos disparos y completar el restante %50.

5.3.5 Bloque de Comunicación y Control

Por último se encuentra el bloque de comunicación y control. Este bloque recibe las instrucciones desde el microprocesador y es el encargado de controlar el funcionamiento del resto de la FPGA. Cualquier variación de parámetros propuesta por el usuario es interpretada por el microprocesador que envía la instrucción correspondiente a la FPGA. El bloque de comunicación y control recibe la instrucción, la interpreta e internamente se comunica con el Bloque de ADC, el Bloque de Almacenamiento, el Bloque de Acondicionamiento y el Bloque de *Trigger* para realizar los cambios necesarios. Para dicho fin, se creó un set de instrucciones que emplean la FPGA y el Microprocesador.

La comunicación entre los dos dispositivos se implementó mediante el periférico SPI del microprocesador, desarrollando la lógica de decodificación necesaria en la FPGA. El periférico permite realizar pedidos de forma asincrónica, pero tiene el envío de bits de forma sincronizada lo cual es necesario para cumplir con los requerimientos de alta velocidad que son necesarios en esta comunicación. A su vez, para llevar a cabo el diseño del set de instrucciones se debió afrontar el desafío de garantizar la implementación eficaz de su decodificación en la FPGA. Para lograr una longitud específica de datos para cada instrucción, se utilizaron los bits 1 y 2 de cada código binario para indicar la cantidad de datos. De esta misma forma, para indicar el canal de la instrucción, se utilizó el bit 0. De forma similar, fue posible implementar una longitud variable de datos de respuesta en el pedido de datos de cada canal. Se estableció que en los 2 primeros *bytes* de la respuesta de dicho pedido, se comunicaran la cantidad de *bytes* que serían enviados posteriormente. Esto posibilita que según el modo de muestreo del osciloscopio, la cantidad de datos de respuesta sean variables.



Código binario de instrucción								Nombre	Datos (bits)	Datos (bytes)	Código binario de datos									Respuesta			Descripción
MSB							LSB				MSB								LSB	ACK	DNR (MSB)	DNR (LSB)	DATOS (...)	
0	0	0	0	0	0	0	1	Hola	0	0										ACK				Saludo cordial.
1	0	0	0	0	0	0	0	DataReqCh0	0	0										ACK				Pedido de datos del canal. La respuesta comienza con ACK, luego con la cantidad de datos (DNR) que se responderán y por último los datos (longitud variable).
1	0	0	0	0	0	0	1	DataReqCh1	0	0														
0	1	0	0	0	0	1	0	StatusCh0	6	1	x	x	CH	CP	G[3..0]					ACK				Estado del canal: canal encendido (CH), coupling (CP), ganancia del canal (G).
0	1	0	0	0	0	1	1	StatusCh1	6	1														
0	0	1	0	0	1	1	0	OffsetCh0	18	3	O[17..0]									ACK				Modifica el nivel de offset del canal (O).
0	0	1	0	0	1	1	1	OffsetCh1	18	3														
0	0	0	1	0	0	1	0	TriggerStatus	6	1	x	x	TCH	R	A	S	NR	HFR		ACK				Estado del trigger: canal de trigger (TCH), modo roll (R), modo automático (A), pendiente del trigger (S). Noise reject (NR), High-frequency reject (HFR).
0	0	1	0	0	1	1	1	TriggerLevel	8	1	TL[7..0]									ACK				Modifica el nivel de trigger.
0	0	0	1	0	1	0	0	FreqSampling	5	1	x	x	x	FS[4..0]						ACK				Modifica la frecuencia de sampleo (FS).
0	0	0	1	0	1	1	1	HoldOff	5	1	x	x	x	HO[4..0]						ACK				Modifica el valor de Hold Off (HO).
1	1	1	1	1	1	1	1	Reset	0	0										ACK				Reseteo de la FPGA y ADC.

Fig. 5.13: Máquina de estados del Bloque de Almacenamiento de la FPGA

5.3.6 Plan de Pruebas

En primera instancia, las pruebas que se llevan a cabo en paralelo con el desarrollo de los módulos se realizan en una simulación. El programa que brinda dicha funcionalidad es el *Quartus II*[2] de *Altera*. Cabe destacar que la simulación en esta clase de entornos requiere que previamente se realice la compilación y síntesis de las secciones tal y como serían sintetizadas para ser programadas en el dispositivo físico real. Además, para llevar a cabo la simulación, el entorno realiza una emulación de los componentes físicos utilizando modelo matemático de cada uno de ellos, lo que posibilita tener en cuenta los tiempos de retardo y *glitches* esperados de los componentes.

Para cada Bloque descripto anteriormente se realizaron simulaciones con excitaciones de prueba. En la etapa de prueba se exponen las formas de onda de ciertas señales que no serán utilizadas en las versiones finales. Habiendo comprobado la funcionalidad de los bloques individuales, la integración de los mismos se llevó a cabo en el siguiente orden:

- **Comunicación y Control + Almacenamiento**
- **Comunicación y Control + Almacenamiento + *Trigger***
- **Comunicación y Control + Almacenamiento + *Trigger* + Acondicionamiento**
- **Comunicación y Control + Almacenamiento + *Trigger* + Acondicionamiento + ADC**

Una vez que las simulaciones de los bloques fueron comprobadas y corregidas, se procedió con la programación de la FPGA. Utilizar una placa de desarrollo para la FPGA *Cyclone I* es de gran utilidad para realizar las pruebas dado que existen dos métodos de programación de la misma. El método utilizado durante las pruebas es la programación a RAM, memoria volátil que no tiene permanencia en el apagado y encendido del dispositivo. Solo las versiones finales de los bloques se programan en la memoria Flash del dispositivo y tienen permanencia en el mismo. Dado que la FPGA integra el control del ADC, el plan de pruebas comprende las pruebas sobre el convertidor para verificar la comunicación entre estos bloques y el almacenamiento correcto de muestras.

---

[2] http://www.altera.com/products/software/quartus-ii/web-edition

5.4 Microprocesador

El desarrollo del sistema de procesamiento central al dispositivo representa un gran desafío, ya que involucra frecuencias altas y circuitos integrados de gran complejidad. El sistema debe ser capaz de procesar las muestras provenientes del osciloscopio, aplicar escalamientos, transformaciones y funciones matemáticas sobre estos datos, en paralelo con el manejo de la interfaz de usuario. Una interfaz con *display* LCD gráfico requiere altas capacidades de cálculo, por lo que resultaría ineficiente el uso de un microcontrolador común y corriente. Por otro lado, teniendo en cuenta el desarrollo de software, es conveniente contar con una plataforma que permita usar un sistema operativo completo como Linux, tanto por la facilidad del desarrollo como por el resultado estético y la consecuente experiencia del usuario.

Ahora bien, el desarrollo de una plataforma capaz de correr Linux suele implicar el uso de integrados con encapsulados BGA (*ball-grid arrays*), los cuales presentan un desafío en cuanto a la manufacturabilidad, pues requiere una mayor inversión en equipos para la soldadura, en especial para la fabricación de prototipos. Por lo tanto, seleccionamos un *Applications Processor* que tiene las especificaciones necesarias y es, además, uno de los pocos que se consiguen con encapsulado QFP (*quad flatpack*). En versiones futuras del dispositivo se podría considerar el uso de integrados BGA para obtener una mejor performance, pero es necesario establecer un nivel tecnológico de manufactura que permita soldarlos, lo cual sólo se justificaría luego del éxito comercial de la primera versión y con altos volúmenes de producción esperados.

El procesador elegido es el *i.MX233* de *Freescale Semiconductor*. A continuación se muestran sus especificaciones de acuerdo al fabricante[3]:

- ***ARM926EJ-S™, 454 MHz maximum speed***
- ***16 KB I/D Cache***
- ***Power Management Unit (PMU)***
- ***Mixed Signal Audio***
- ***LCD Controller***

---

[3] http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=i.MX233

- ***HS Device/Host + PHY***
- ***3 UART***
- ***2 SSP***
- ***I2C Master***
- ***6 Channel 12-bit Low Resolution ADC (LRADC)***
- ***Serial Audio IF x2***
- ***SPDIF***
- ***External memory interface:***
    - ***1.8V mDDR, DDR1 w/2.5V regulator***
    - ***Hardware BCH (up to 20-bit correction) and RS ECC8***
    - ***NAND Flash support for SLC/MLC and managed NAND***
- ***This product is included in Freescale's product longevity program, with assured supply for a minimum of 10 years after launch***

Como se observa, las prestaciones del dispositivo permiten desarrollar una gran variedad de productos. La velocidad de *clock* de 454MHz implica un desafío de diseño, por las implicancias de consideraciones de *layout* y cuestiones de compatibilidad electromagnética.

Una gran ventaja de este procesador es la documentación disponible: Freescale provee esquemáticos de referencia y *application notes* que facilitan el diseño. El procesador está pensado para trabajar con Linux, y de hecho el fabricante también provee un *kernel* y *drivers* específicos para esta plataforma.

Por otro lado, el costo es sorprendentemente bajo: ronda los 8 dólares para compra de unidades individuales, lo cual lo equipara en precio a integrados de mucho menores prestaciones.

Se desarrollará, en primera instancia, una placa de desarrollo con el procesador y memoria RAM (64MB) y Flash SPI (32MB). Estas memorias son más que suficientes para permitir el funcionamiento adecuado de un *kernel* de Linux minimalista que permita manejar el dispositivo. Por otro lado, la placa en cuestión es versátil pues se podría utilizar para desarrollar una gran variedad de

dispositivos (*handhelds*, reproductores de audio, *tablets*), por lo que podríamos considerar que es un producto derivado del desarrollo**. Es decir que se podría comercializar por separado como herramienta de desarrollo o utilizar para nuevos productos de la empresa.**

A continuación (fig. 5.14) se muestra el esquemático de dicha placa (en su segunda versión), a la que se le ha dado el nombre de fantasía "LobsterBoard". El esquemático también se agrega en mayor tamaño en el anexo.

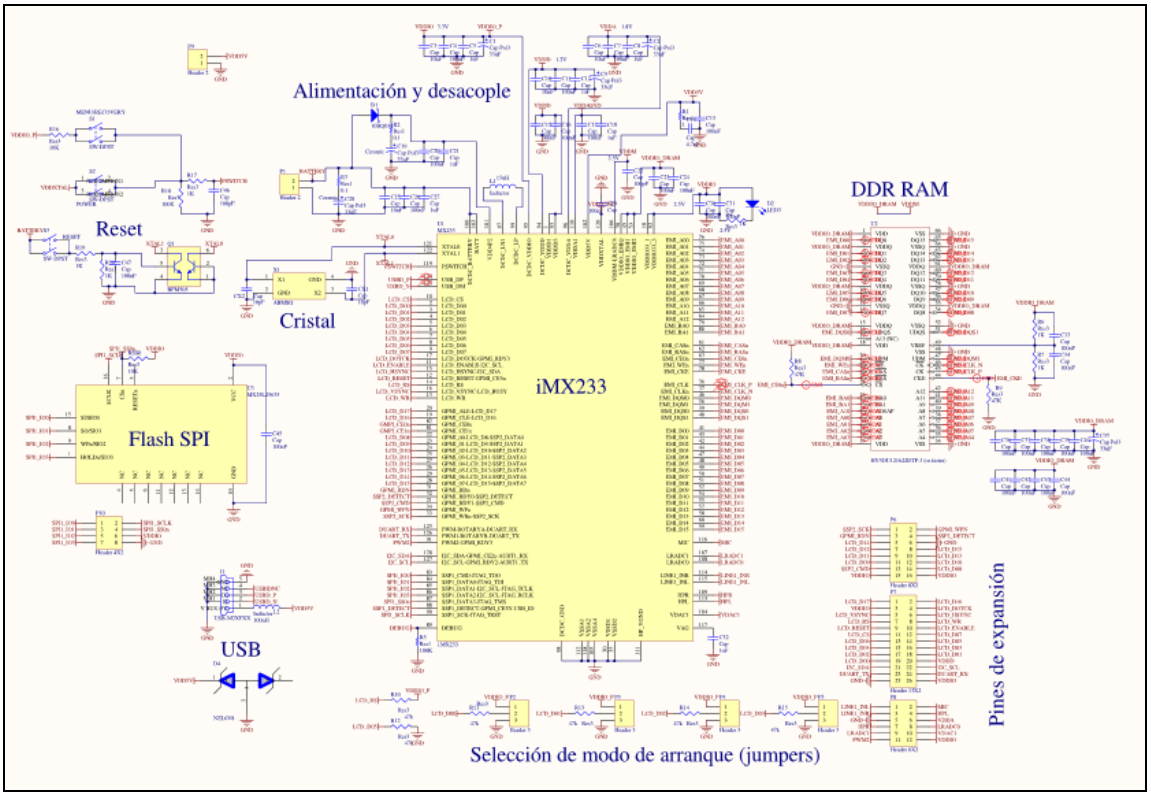


Fig. 5.14: Circuito esquemático de "LobsterBoard v2.0" del microprocesador iMX233

El esquemático se desarrolló usando como base, principalmente, el diseño de referencia de Freescale. También se aprovechó el aporte de proyectos *open-source* basados en este procesador: el *Olinuxino* de Olimex[4] y el *AndroidStamp* de la Universidad Nacional de Colombia[5]. Cuenta con una DDR RAM de 2.5V a

[4] http://www.olimex.com/Products/OLinuXino/

[5] http://linuxencaja.net/wiki/AndroidStamp

133MHz, conectada a la interfaz de expansión de memoria del procesador. La Flash SPI se conecta al SPI1 del microprocesador, dejando el SPI2 para la conexión de los demás periféricos, en particular la placa de adquisición de datos del osciloscopio y la tarjeta SD. Se incluyó capacitores de los valores y en la cantidad propuesta por Freescale para desacoplar las alimentaciones y evitar problemas de ruido.

Todos los puertos del procesador restantes se conectaron a pines de expansión, usando un *header* para los pines del SPI2, otro para los pines del LCD y la interfaz I2C (que se utilizará para el controlador de touchscreen) y otro para todos los restantes. Cada *header* cuenta por lo menos con una alimentación de 3.3V y un GND, de manera de poder conectar y alimentar una *daughterboard* de expansión.

Para la alimentación de la placa se aprovecha una ventaja del i.MX233, y es que cuenta con un sistema de *Power Management* avanzado, diseñado para alimentarse por 5V de USB y con una batería de ión-litio de 3.7V. El procesador es capaz de controlar la carga de la batería y cuenta con reguladores y fuentes *switching* que dan salidas de 1.2, 1.8, 2.5 y 3.3 Volts para alimentar todos los periféricos. La fuente *switching* utiliza una única bobina de 15uH que se observa en la parte superior del procesador en el esquemático.

El resultado de este diseño es prácticamente una *computer-on-board* con sólo tres circuitos integrados.

5.4.1 Plan de Pruebas

El sistema de procesamiento desarrollado implica un cuidadoso diseño digital y el proceso de prueba está muy atado al desarrollo de software (el *kernel* de Linux y el correspondiente *bootloader*, sobre lo cual se desarrolla en la sección de Software). El plan de pruebas debe llevar a levantar el sistema Linux completo sobre la plataforma.

Al ser un proceso largo y sujeto a la incertidumbre del *testeo* (una falla que implique necesidad de rediseño podría atrasar considerablemente el desarrollo), se decidió utilizar un *OLinuXino* como solución provisoria en el prototipo para avanzar en el resto del sistema (*software* e interacción con la etapa de entrada y el bloque de interfaz). El diseño final del equipo apunta al uso de la LobsterBoard.

Ahora bien, una vez que el circuito impreso está fabricado y soldado, el plan de pruebas es el siguiente:

- **Alimentar desde USB. Comprobar las tensiones de salida del regulador. Esto indica el correcto funcionamiento del sistema de *Power Management* del procesador.**

- **Comprobar que el procesador es detectado como dispositivo USB por una PC. Esto valida el diseño de las pistas de USB, que al ser de alta velocidad podrían presentar problemas electromagnéticos.**

- **Cargar por USB un programa de prueba para la RAM externa. Este programa es, básicamente, un pequeño ciclo de lectura escritura, y que imprimiría por el puerto serie el posible error de diferencia entre valor escrito y valor leído.**

- **Utilizar una tarjeta SD para correr Linux. Esto implica tener un *kernel* preparado para correr desde una SD. Se valida, entonces, tanto *software* como *hardware* en este paso. La tarjeta SD se conecta al SPI2 con una placa que consiste sólo en el conector, un capacitor de desacople y una resistencia de *pull-up*, como se muestra en el esquemático de la figura 5.15.**

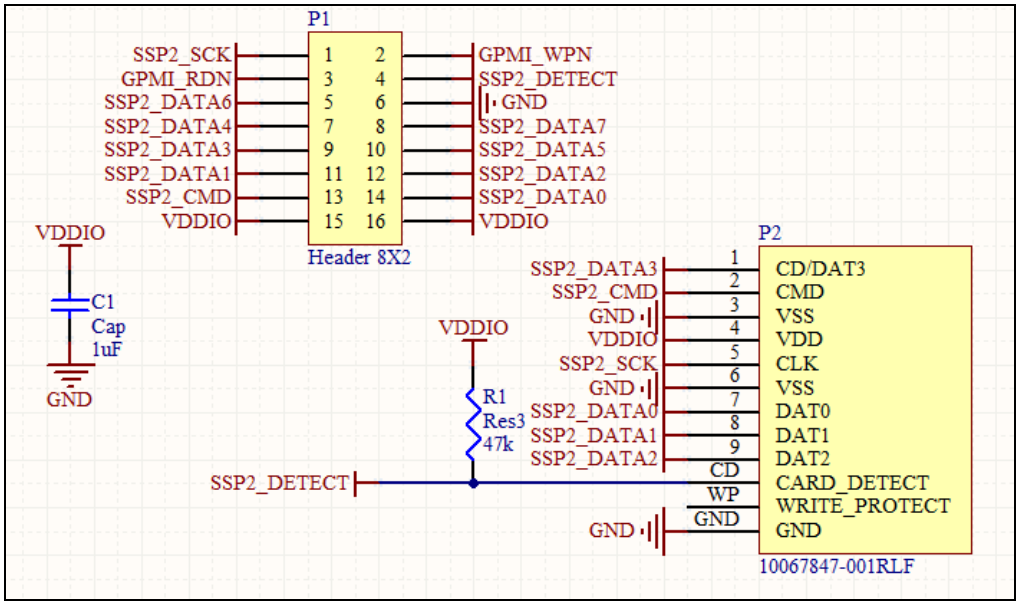


Fig. 5.15: Esquemático del conector para tarjeta SD.

- **Una vez que Linux corre desde la SD, se debe cargar a la memoria *Flash* SPI el *kernel* y resto del sistema operativo (este sería el *kernel* definitivo, preparado para ejecutarse desde la *Flash*). Esto es a su vez la prueba del último componente (dicha memoria).**

5.5 Interfaz LCD y touch

El bloque de interfaz de usuario consiste en un *display* LCD con pantalla táctil y un controlador para dicha pantalla táctil. El controlador muestrea las líneas provenientes de la pantalla, hace una conversión analógico/digital y las envía al procesador por algún protocolo.

Se decidió utilizar el controlador TSC2007 de *Texas Instruments*, que es de bajo costo y cuenta con *drivers* para Linux. Este controlador utiliza el protocolo I2C.

En cuanto al *display*, se consiguió un modelo de 5.6 pulgadas de un proveedor chino, Shenzhen Control, que cuenta con una interfaz compatible con la del procesador, es de muy bajo costo (46 dólares por 1 unidad) y tiene las prestaciones necesarias. La hoja de datos del *display*, cuyo modelo es AT056TN53, se adjunta como anexo.

Se diseñó una placa de interfaz entre el procesador y el *display*. Esta placa cuenta con los conectores necesarios, el controlador de *touch* y una fuente *switching* para poder alimentar el *backlight* tanto desde los 5V del USB o cargador como desde los 3.7V de la batería.

La alimentación de dicha fuente se obtiene de la batería y pasa a los 5V cuándo estos están presentes mediante un MOSFET que conmuta automáticamente.

A continuación (fig. 5.16) se muestra el esquemático de dicha placa de interfaz, también adjuntada con mayor resolución en el anexo.

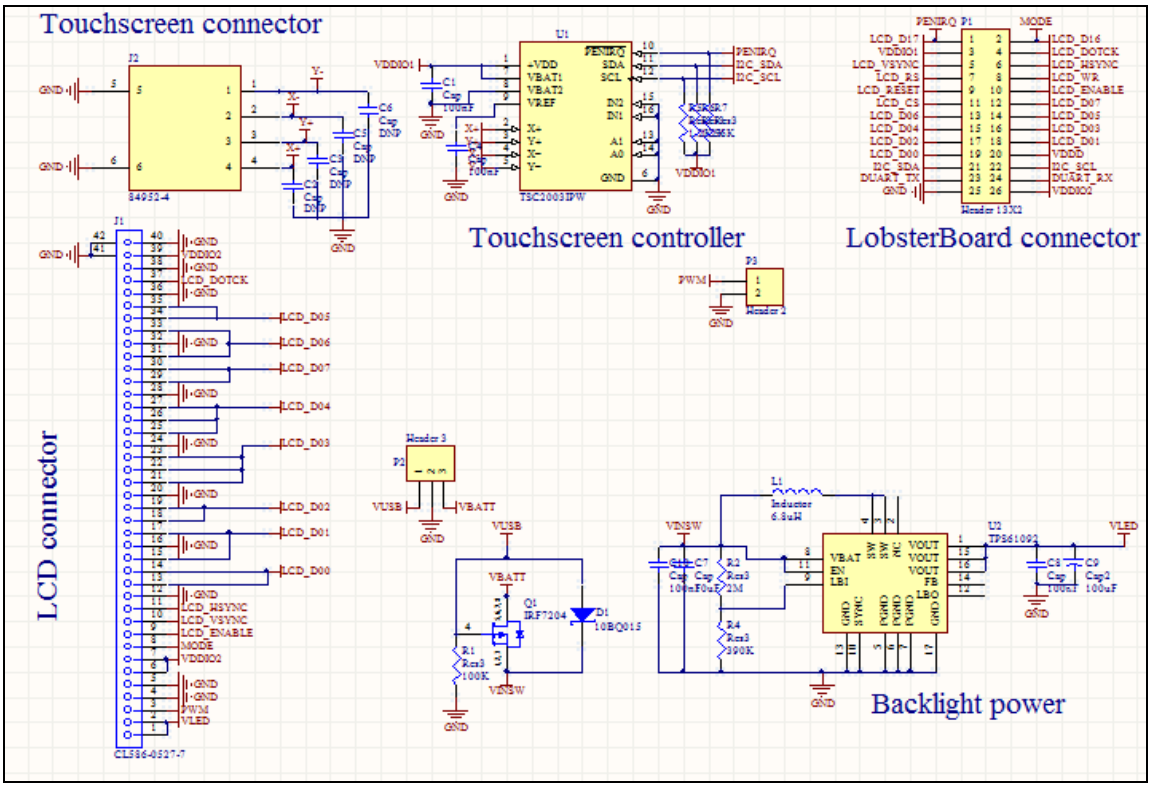


Fig. 5.16: Esquemático de la placa de interfaz procesador - display.

5.5.1 Plan de pruebas

Para probar este sistema, se debe primero alimentar y comprobar que la salida de la fuente sea 5V, tanto con batería como con alimentación USB. De esa manera se comprueba que la fuente y el circuito del MOSFET funcionan. Luego se lo debe conectar al sistema del procesador (con los drivers correspondientes) y al *display*, y evaluar que se puede mostrar imágenes en el *display* y recibir mediciones del controlador de *touch*.

VI. INGENIERÍA DE SOFTWARE

El desarrollo de software necesario para el osciloscopio tiene dos bloques fundamentales: el *kernel* de Linux con los drivers específicos de la plataforma de *hardware* desarrollada, y la aplicación de osciloscopio propiamente dicha. Esto corresponde con la habitual separación entre *user-space* ("el espacio del usuario") y *kernel-space* ("el espacio del *kernel*") del *software* de Linux. La figura 6.1 (ver próxima página) muestra un diagrama de bloques del sistema completo, donde se observa esta separación.

Para que el desarrollo sea exitoso, los drivers deben manejar los periféricos correspondientes cumpliendo los requerimientos de tiempo real, y la aplicación debe hacer un uso eficiente de los mismos implementando la interfaz y lógica deseadas.

Al estar trabajando sobre una plataforma de *hardware* nueva (ya que la distribución de los periféricos es distinta a la de placas existentes como el Olinuxino) se debió modificar buena parte de la sección dependiente de hardware del *kernel*. También se debió generar un *rootfs* (es decir, el sistema de archivos con los ejecutables y librerías fundamentales para el sistema operativo) específico para la memoria de almacenamiento limitada con la que se cuenta.

Por otro lado, la aplicación del osciloscopio (llamada "*softscope*") corre como una aplicación normal en *user-space*, y se configura el sistema operativo para que se ejecute en el arranque del sistema.

En las siguientes subsecciones se explicará en detalle las características de cada uno de los subsistemas mencionados.

J

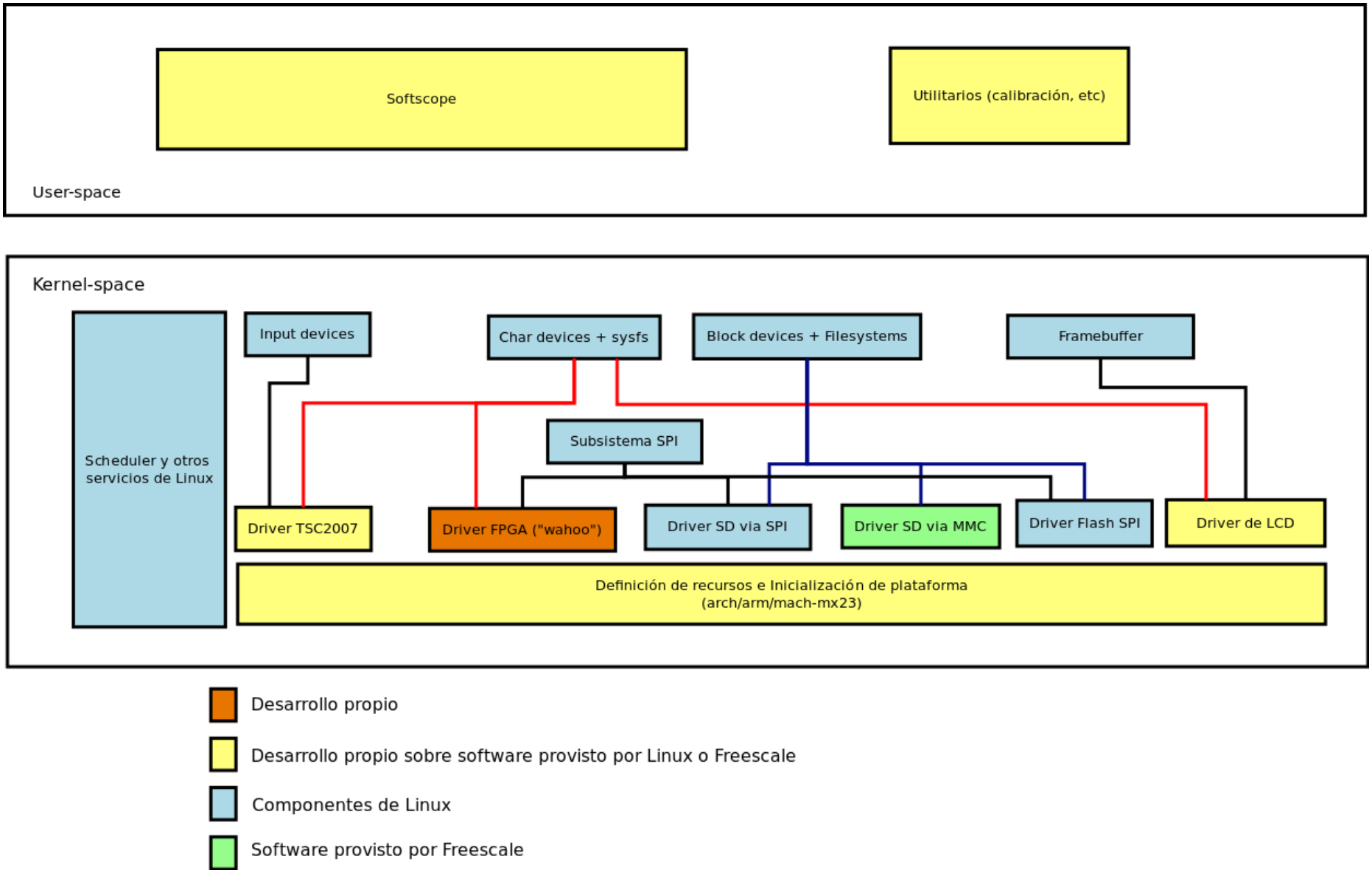


Fig. 6.1: Esquemático de la placa de interfaz procesador - display.

6.1 Kernel Linux

Para agregar la funcionalidad necesaria al *kernel*, se partió de una versión provista por Freescale en un repositorio de *git*[6]. Se trata de la versión 2.6.35 del *kernel* oficial de Linux al que le han agregado los archivos específicos de la plataforma i.MX233. Sin embargo, ese *kernel* está diseñado para la *Evaluation Board* de Freescale, por lo que para usarlo en el Olinuxino se deben aplicar algunos parches provistos por Olimex[7].

Este repositorio fue copiado a uno nuevo, propio, en el que se agregaron los cambios del proyecto (luego de aplicar los parches mencionados). Se puede consultar el repositorio en su página de *Github*[8].

Por otro lado, el *kernel* de Linux se carga utilizando un *bootloader* de Freescale que consiste en pequeños programas llamados *bootlets*. Estos se obtienen en el sitio de Freescale y se les deben aplicar algunos parches para adaptarlos al Olinuxino, y luego a la LobsterBoard.

Además, como se mencionó al comienzo de esta sección, el *kernel* necesita un *rootfs* para funcionar. El *rootfs* es el sistema de archivos, ya sea en la memoria *Flash* SPI o en la tarjeta SD, que contiene los programas y configuraciones necesarios para el sistema[9]. Esto se realizó utilizando la herramienta *buildroot*[10], que automatiza la creación de directorios y permite seleccionar qué paquetes de programas incluir.

El desarrollo dentro del *kernel* tiene dos partes importantes:

- Generar una definición de los distintos periféricos de *hardware* presentes en el sistema, y configurar el *kernel* de acuerdo a esa definición.

---

[6] http://git.freescale.com/git/cgit.cgi/imx/linux-2.6-imx.git/

[7] https://github.com/OLIMEX/archlinuxarm-olinuxino/tree/master/olinuxino/kernel26-olinuxino

[8] https://github.com/pcarranzav/linux-for-lobster

[9] http://www.tldp.org/HOWTO/Bootdisk-HOWTO/buildroot.html

[10] http://buildroot.uclibc.org/

- Desarrollar un *driver* para el circuito que se ha implementado en la FPGA.

La primera de estas dos tareas es, en definitiva, "portar" Linux la plataforma de *hardware* con la que se está trabajando. La segunda es producir el *software* que brinde una capa de abstracción sobre *hardware* nuevo y original, trabajando con el protocolo SPI descripto en la sección 5.3.5.

6.1.1 Definición de arquitectura

El punto de partida para portar Linux suele ser una plataforma similar; en este caso, se parte de los archivos propios del *Olinuxino*, ya que este fue usado en gran parte del desarrollo y tiene periféricos en común con la *LobsterBoard* desarrollada.

Para comenzar, entonces, se necesita definir el *hardware* que se conecta al procesador y con qué *buses* y protocolos. La realidad es que hay tres definiciones de *hardware* distintas: una es el Olinuxino con el *hardware* adicional (utilizado para el desarrollo del osciloscopio), otra es la LobsterBoard con el *kernel* en la tarjeta SD, y otra (la versión final) es la LobsterBoard con el *kernel* en la memoria *Flash* SPI. Las figuras 6.2, 6.3 y 6.4 ilustran estas distintas plataformas con sus periféricos más relevantes.

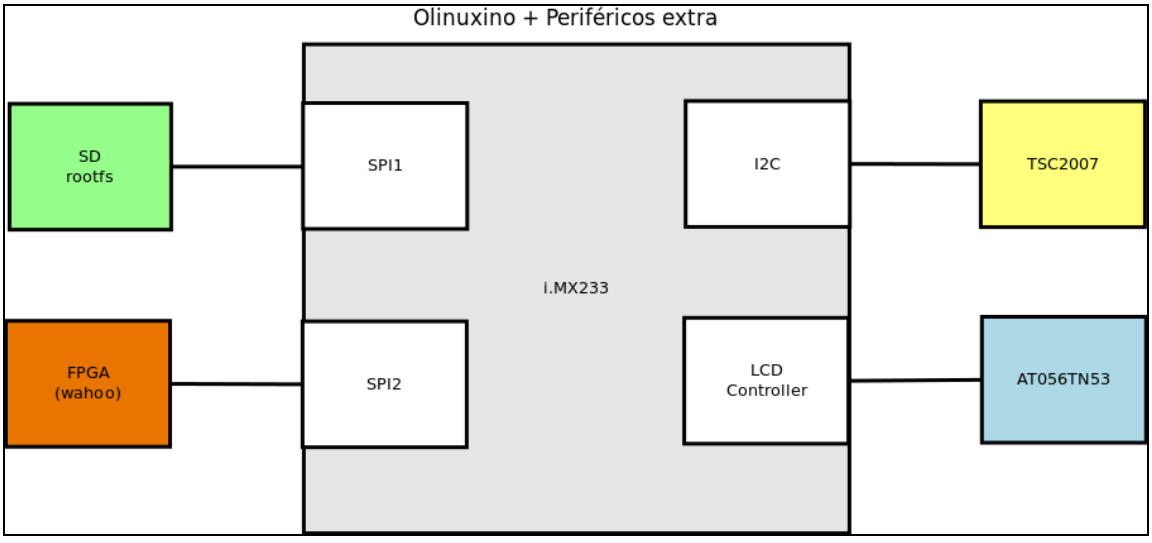


Fig. 6.2: Estructura de hardware del Olinuxino con periféricos adicionales

El Olinuxino, en su versión original, sólo cuenta con la SD en el SPI1 del procesador. A eso se le ha agregado el controlador de *touch*, el *display* LCD y la FPGA en el SPI2.

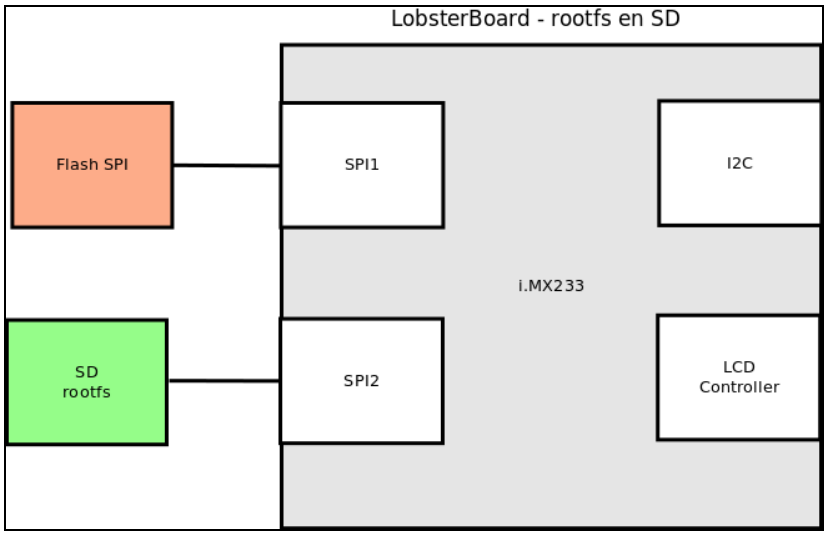


Fig. 6.3: Estructura de hardware de la LobsterBoard con sistema operativo en la tarjeta SD

En la LobsterBoard esa estructura cambia: ahora la SD se encuentra en el SPI2 y el SPI1 tiene la memoria *Flash*. Los otros periféricos no son necesarios para esta etapa, pues sus únicos objetivos son *testear* la LobsterBoard y grabar el último *kernel* en la *Flash*.

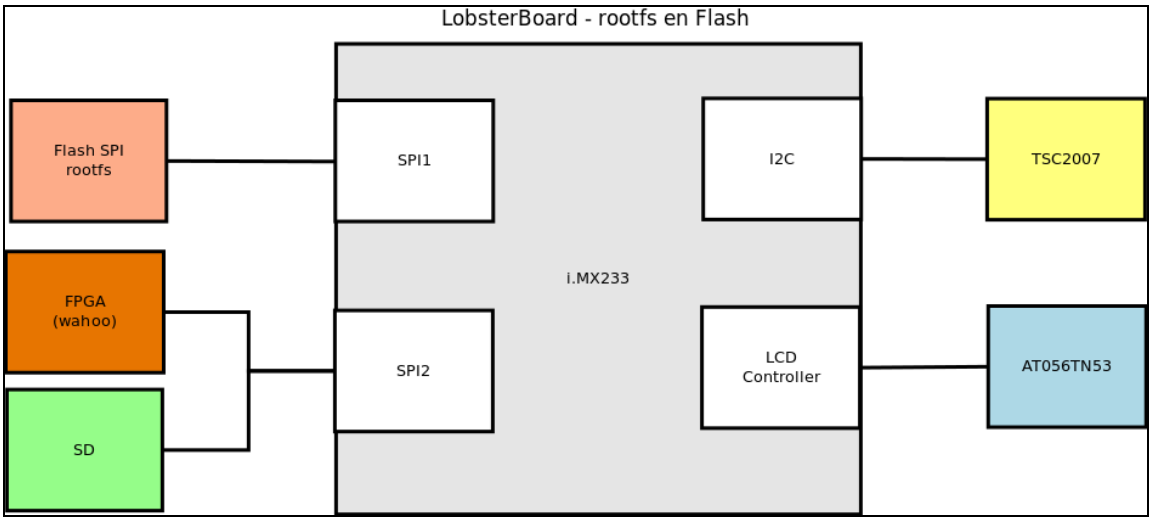


Fig. 6.4: Estructura final de hardware de la LobsterBoard, con sistema operativo en Flash SPI

La versión final del *kernel* en la LobsterBoard tiene el sistema operativo en la memoria *flash* en SPI1, y el SPI2 está compartido entre la tarjeta SD y la FPGA. Nuevamente se encuentran el LCD y el controlador de *touch*.

Las definiciones que dependen de la plataforma se encuentran concentradas en un directorio del *kernel* llamado *arch.* Los archivos para el i.MX233 se encuentran en la carpeta *arch/arm/mach-mx23.* En este directorio están los archivos para todas las plataformas basadas en este procesador; fundamentalmente, la definición de arquitectura para el Olinuxino está en los archivos *imx23_olinuxino.c* e *imx23_olinuxino_pins.c*. El primero tiene las funciones de inicialización del procesador y sus *buses* y periféricos internos, el segundo tiene las definiciones de la asignación de pines (ya que estos se pueden multiplexar entre varios periféricos).

Además, el archivo *device.c* contiene las definiciones de todos los recursos correspondientes a cada periférico, es decir, en qué zonas de memoria se encuentran los registros de cada uno, qué canales de DMA y qué interrupciones utilizan.

Para adaptar el *kernel* a la arquitectura de la LobsterBoard, se crearon nuevos archivos *imx23_lobster.c* e *imx23_lobster_pins.c*, partiendo de la base de los archivos del Olinuxino.

Se está definiendo una nueva plataforma, por lo que se debe definir la estructura de la "máquina" (*machine*). Esto se realiza con el macro MACHINE_START, como se muestra a continuación:

```
MACHINE_START(IMX233_LOBSTER, "iMX233 LobsterBoard")
	.phys_io	= 0x80000000,
	.io_pg_offst	= ((0xf0000000) >> 18) &
0xfffc,
	.boot_params	= 0x40000100,
	.fixup		= fixup_board,
	.map_io			= mx23_map_io,
	.init_irq	= mx23_irq_init,
	.init_machine = imx233_lobster_init_machine,
	.timer		= &mx23_timer.timer,
MACHINE_END
```

El macro define una estructura y la inicializa con algunos valores, como por ejemplo la dirección de comienzo de la zona de registros, *phys_io*.

Otros campos son punteros a distintas funciones: mapeo de memoria, *timer* y, fundamentalmente, inicialización (*init_machine*). Este esquema de estructura con punteros a función es muy habitual en el *kernel* de Linux; de esa manera se

generalizan funcionalidades y cada implementación específica "registra" qué puede hacer y con qué función.

El *kernel* de Linux se compila según distintas "configuraciones". Esto es lo que permite que sea multiplataforma: según las opciones que se seleccionen en la configuración, varía desde qué archivos se compilan hasta qué compilador se utiliza. Las opciones de configuración se agregan en un archivo llamado *Kconfig*, y luego se agregan los nuevos archivos al *makefile* condicionalmente:

```
obj-$(CONFIG_MACH_IMX233_LOBSTER) += imx233_lobster.o imx233_lobster_pins.o
```

Las opciones de configuración son, además, constantes de preprocesador que están disponibles al momento de compilar, por lo que se puede realizar compilación condicional. Por ejemplo, el siguiente código muestra la función de inicialización de la plataforma, donde el SPI y el I2C se inicializan sólo si el módulo correspondiente fue incluido en la configuración:

```
static void __init imx233_lobster_device_init(void)
{
	imx233_lobster_init_adc();

	#if defined(CONFIG_TOUCHSCREEN_TSC2007)
	i2c_device_init();
	#endif
	#if defined(CONFIG_SPI_MXS)
	spi_device_init();
	#endif
}
```

En *imx23_lobster_pins.c* se define la multiplexación de pines. Esto también se realiza con estructuras, como se muestra en el siguiente ejemplo:

```
#if (!defined(CONFIG_SPI_MXS_SSP2)) && (defined(CONFIG_SPI_MXS) || \
defined(CONFIG_SPI_MXS_MODULE))
static struct pin_desc lobster_spi1_pins[] = {
	{
	 .name = "SSP1_DATA0",
	 .id   = PINID_SSP1_DATA0,
	 .fun  = PIN_FUN1,
	 .strength    = PAD_4MA,
	 .voltage     = PAD_3_3V,
	 .drive       = 1,
	 },
	{
	 .name = "SSP1_DATA3",
	 .id   = PINID_SSP1_DATA3,
	 .fun  = PIN_FUN1,
	 .strength    = PAD_4MA,
	 .voltage     = PAD_3_3V,
	 .drive       = 1,
	 },
	{
	 .name = "SSP1_CMD",
	 .id   = PINID_SSP1_CMD,
	 .fun  = PIN_FUN1,
	 .strength    = PAD_4MA,
	 .voltage     = PAD_3_3V,
```

Este arreglo contiene las estructuras para los pines del SPI1. Estos se registran en la inicialización del módulo de SPI, en la siguiente función:

```
int mxs_spi_enc_pin_init(void)
{
	#if (defined(CONFIG_SPI_MXS) || defined(CONFIG_SPI_MXS_MODULE))
	mxs_request_pins(lobster_spi1_pins, ARRAY_SIZE(lobster_spi1_pins));
	#endif
	return 0;
}
```

6.1.2 Bus I2C – Driver de touch

Para soportar los nuevos periféricos (es decir, los que no están en el Olinuxino original), se agregan estructuras para identificar el *hardware* conectado en el *bus* I2C, por ejemplo:

```
static struct tsc2007_platform_data tsc2007_info = {
	.model              = 2007,
	.x_plate_ohms       = 530,
	.get_pendown_state = tsc2007_get_pendown_state,
	.init_platform_hw = lobster_tsc2007_pin_init,
	.exit_platform_hw = lobster_tsc2007_pin_release,
};

static struct i2c_board_info lobster_i2c_devices[] = { {
		I2C_BOARD_INFO("tsc2007", 0x48),
		.platform_data      = &tsc2007_info,
		/* irq number is run-time assigned */
	},
};
```

Luego, en la incialización se registra esta estructura:

```
static void i2c_device_init(void)
{
	// Define a GPIO pin for the TSC2007 interrupt
	lobster_i2c_devices[0].irq = gpio_to_irq(TSC2007_IRQGPIO);
	// Register I2C board info
	i2c_register_board_info(0, lobster_i2c_devices, ARRAY_SIZE(lobster_i2c_devices));
}
```

De esta manera se le da al driver del TSC2007, que es multiplataforma, la información específica de esta implementación: el modelo de controlador (*.model*), la resistencia de la pantalla táctil (*x_plate_ohms*, que se utiliza para hacer el cálculo de presión), y punteros a las funciones para utilizar el *hardware* específico, que en este caso es un pin que realiza una interrupción cuando el *touch* es presionado.

Por otro lado, se realizó una pequeña modificación en el *driver* del controlador, siguiendo una recomendación de la hoja de datos. Cuando se pide una medición de la posición del *touch*, se debe activar la alimentación de la línea del eje X unos milisegundos antes de pedir los datos; de esa manera, el filtro de entrada del conversor analógico digital se puede asentar (pues tiene un transitorio) y la medición es más confiable.

Este *driver* se encuentra en el archivo *drivers/input/touchscreen/tsc2007.c*. Utiliza la interfaz estándar de Linux para dispositivos de entrada, creando archivos en la carpeta */dev/input* del sistema de archivos desde los cuales se pueden leer los eventos de *touch*.

6.1.3 Bus SPI

El mismo procedimiento que para I2C se sigue para los dispositivos en el *bus* SPI. No obstante, el uso de este bus requiere más trabajo, ya que en la implementación de Freescale sólo se había considerado el uso del SPI1, ya sea en modo SPI o en modo MMC (para manejar una SD). Esta situación es más compleja, ya que se utilizarán los dos *buses*, y en uno de ellos habrá una SD y un dispositivo SPI.

La solución es definir distintas opciones de configuración en *device.c*, donde se asignan los recursos del SPI1 y SPI2 ya sea al *driver* de SPI o al de MMC. Por otro lado, para controlar la tarjeta SD con el mismo *bus* que la FPGA, se debe no utilizar el driver de MMC sino el de SPI, y manejar la tarjeta en modo SPI (es decir, como un dispositivo común y sin el protocolo especial de Secure Digital). Es por esto que, en la figura 6.1 mostrada al comienzo de esta sección, se presentan dos *drivers* de SD, uno via MMC (usando el controlador nativo del procesador, en

el Olinuxino y en el primer *kernel* de LobsterBoard) y otro vía SPI (para la versión final).

6.1.4 Display LCD

Freescale provee un *driver* para el controlador de LCD que trae el procesador. Sin embargo, este *driver* está pensado para dos modelos de *display* que funcionan en modos gráficos diferentes a los del AT056. Por lo tanto, fue necesario realizarle algunas modificaciones.

El driver se encuentra en la carpeta *drivers/video/mxs*, con algunas funciones *inline* en el header *arch/arm/mach-mx23/include/mach/lcdif.h*.

Para agregar soporte para el AT056, se creó un archivo *lcd_at56.c* a partir de uno similar, *lcd_lms430.c*.

Este archivo provee una estructura que define las características del LCD, como se muestra a continuación:

```
static struct mxs_platform_fb_entry fb_entry =
{
        .name = "at56",
        .x_res = 640,
        .y_res = 480,
        .bpp = 8,
        .cycle_time_ns = 40,
        .lcd_type = MXS_LCD_PANEL_DOTCLK,
        .init_panel = init_panel,
        .release_panel = release_panel,
        .blank_panel = blank_panel,
        .run_panel = mxs_lcdif_run,
        .stop_panel = mxs_lcdif_stop,
        .pan_display = mxs_lcdif_pan_display,
        .bl_data = &bl_data,
};
```

En esta estructura se definen los parámetros de resolución, *timing*, protocolo de comunicación y bits por píxel. Además, se proveen punteros a las funciones propias de este *display*. Como ejemplo, se muestra a continuación la función de inicialización, que escribe a los registros del controlador los valores apropiados para la configuración de este panel:

```
static int init_panel(struct device *dev, dma_addr_t phys, int memsize, struct mxs_platform_fb_entry *pentry)
{
        int ret = 0;
        lcd_clk = clk_get(NULL, "lcdif");
        if (IS_ERR(lcd_clk)) {
                ret = PTR_ERR(lcd_clk);
                return ret;
        }
        ret = clk_enable(lcd_clk);
        if (ret) {
                clk_put(lcd_clk);
                return ret;
        }
```

Las constantes que se pasan como parámetros a la función *setup_dotclock_panel_8bit* proveen los valores de *timing* según la hoja de datos del *display*.

Esta función también tuvo que ser desarrollada, ya que el driver no estaba pensado para utilizar un bus de sólo 8 bits de datos. Además, se debió modificar el *driver* en varias secciones para que soporte este ancho de bus.

Una vez realizados estos cambios, la funcionalidad del driver permite usar el *display* con la interfaz estándar de *framebuffer* de Linux, con la que el usuario puede *mapear* una zona de memoria, a partir del archivo */dev/fb0*, a los píxeles del *display*.

6.1.5 Driver de FPGA – "Wahoo"

Se desarrolló un *driver* específico para el circuito programado en la FPGA (denominado con el nombre "Wahoo" que trabaja sobre la interfaz de SPI del *kernel*. Este se puede encontrar, como corresponde según su tipo de dispositivo, en *drivers/spi/wahoo.c.*

El *driver* implementa el protocolo descrito en la sección 5.3.5. Como se suele trabajar en el *kernel*, la funcionalidad se expone al usuario mediante archivos (que no se escriben a la memoria no-volátil sino que están en un sistema de archivos virtual); se eligió trabajar con el sistema *Sysfs*, donde cada archivo es un "atributo" del dispositivo que puede ser leído o escrito. Se define, entonces,

una función de lectura y una función de escritura para cada atributo, y el *kernel* las llamará cada vez que el usuario lea o escriba el archivo.

Los atributos se corresponden con las distintas funcionalidades del osciloscopio: el *trigger*, la configuración de los canales, los datos provenientes de cada canal y la frecuencia de muestreo. Por otro lado, se implementó un atributo de "*reset*" para permitir establecer la configuración *default.*

En cada escritura de los archivos de configuración se pasa una estructura que define el estado que se desea aplicar. Estas estructuras son las siguientes:

- Estructura de configuración de canal

```
struct chan {
	char id;		// Channel id (0: A, 1: B)
	char on;		// 1 if the channel is enabled, 0 otherwise
	unsigned char gain;	// Gain for the channel, setting vertical scale
	char coupling;		// DC or AC coupling
	unsigned int offset;// Vertical offset
};
```

- Estructura de configuración de trigger

```
struct trig {
	char channel;		// Channel for edge detection (0: A, 1: B)
	char slope;		// Trigger on rising (1) or falling (0) edge
	char aut;		// Auto trigger on (1) or off (0)
	char roll;		// Roll mode
	char noise;		// Noise reject
	char hf;		// High Frequency reject
	char holdOff;		// Hold-Off value
	char level;		// Trigger level value
};
```

A continuación se muestra como ejemplo la función de escritura de configuración de frecuencia de muestreo:

```
static ssize_t wahoo_sample_rate_store(struct device *dev,
		struct device_attribute *attr,
		const char *buf, size_t len)
{
	struct wahoo_state *st = dev_get_drvdata(dev);
	unsigned char ops[3];
	int ret;
	// Check the parameter is only 1 byte
	if (len != 1)
		return -EINVAL;
	// Lock mutex to prevent conflicts
	mutex_lock(&st->lock);

	//Send CHANNEL_CONF command.
	ops[0] = (unsigned char) (buf[0] & SAMPLE_RATE_MASK);
	ret = wahoo_send_command(st,CMD_SAMPLE_RATE,ops,1);
	// Check if command was properly sent
	if(ret<0)
```

Estás funciones se crean y se definen como atributos usando el siguiente macro:

```
static DEVICE_ATTR(sampleRate, S_IWUSR |S_IRUGO, wahoo_sample_rate_show, wahoo_sample_rate_store);
```

De esa manera, al registrar este atributo en la inicialización del driver, se crea un archivo "*sampleRate*" en un subdirectorio de */sys*, cuya escritura dispara la función antes citada.

El uso de la interfaz de SPI se vislumbra en la función para enviar un comando al dispositivo:

```
// Send a command to Wahoo. Returns the one-byte response or a negative
// error.
static int wahoo_send_command(struct wahoo_state *st, unsigned char
cmd, unsigned char ops[], int nops)
{
	unsigned char *buf = st->tx_buff;
	int i;
	int ret;
	unsigned char val=0;
	buf[0] = cmd;
	for(i=0;i<nops;i++)
		buf[i+1] = ops[i];

	// Use SPI to send the command and read the response
	ret =  spi_write_then_read(st->us, buf, nops+1, &val, 1);
	//dev_err(&st->us->dev, "Received %x\n",val);

	if(ret<0)
	{
		dev_err(&st->us->dev, " Error: SPI failed");
		return ret;
	}
	else
	{
		if(val!=(unsigned char)1)
			dev_err(&st->us->dev,
"Error: Received %x\n instead of 1",val);
		return val;
	}
}
```

Aquí se observa una de las bondades de Linux: La interfaz de SPI es genérica, por lo que el driver se podría portar a cualquier plataforma con SPI sin cambiar ni una línea de código.

6.2 Aplicación de usuario: *Softscope*

6.2.1 *Softscope*: diseño e implementación

Sobre el *kernel* del sistema operativo descrito anteriormente se programó la aplicación principal del osciloscopio llamada *softscope*. La función de dicha aplicación es la de graficar la interfaz, responder adecuadamente a las interacciones que realice el usuario, y realizar el procesamiento y presentación de los datos.

La aplicación está escrita en C++ y se ejecuta directamente sobre el sistema operativo (Linux). El manejo fino del hardware y los periféricos es realizado a través de los drivers del sistema operativo, funcionando *softscope* como un árbitro o controlador de los mismos.

Una de las bondades del uso de una plataforma con Linux es la posibilidad de utilizar cualquier lenguaje de programación que cuente con un compilador para la arquitectura de trabajo. Se decidió utilizar C++ ya que permite un manejo a bajo nivel de los periféricos y a la vez brinda la posibilidad de estructurar la aplicación en clases y trabajar con un paradigma de programación orientado a objetos. Este enfoque permite desacoplar de manera fácil e intuitiva las distintas funciones de la aplicación facilitando el desarrollo.

La aplicación consiste en más de 5000 líneas de código por lo que por practicidad se brinda simplemente una breve descripción su funcionamiento. Para ello se presenta un diagrama de clases donde se pueden ver las clases principales del programa, la interacción entre ellas y su relación con los drivers de Linux; así como también se muestran los métodos más importantes de cada una de ellas.

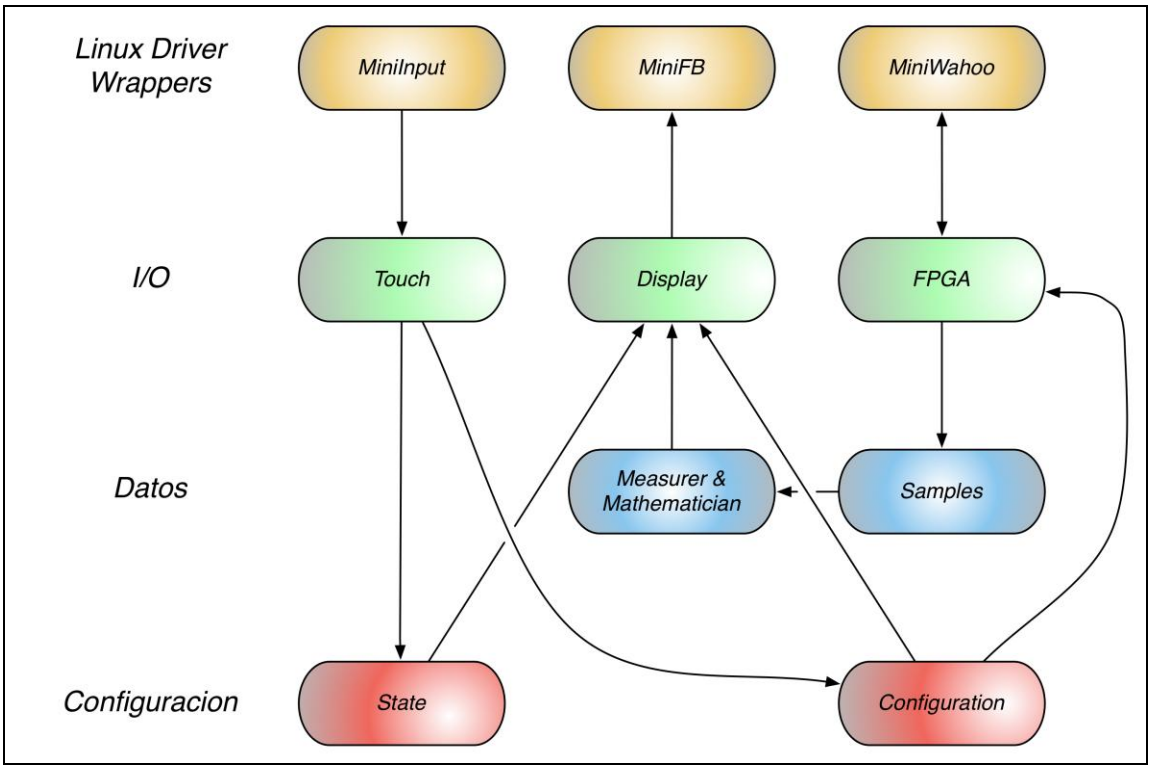


Fig. 6.5: Diagrama de clases de la aplicación softscope

Como se observa en el diagrama, las principales clases pueden dividirse en cuatro categorías:

- **Linux Driver Wrappers:** Estas clases funcionan como una capa de abstracción sobre los *drivers* del sistema operativo. Permite que el resto de la aplicación acceda a dichos *drivers* de manera transparente, es decir sin una referencia específica a la arquitectura:

  - **MiniInput:** Crea un nuevo *thread* (proceso que corre en simultaneo en el procesador) que está constantemente atendiendo las entradas del driver estándar de *mouse* de Linux (utilizado para controlar el dispositivo táctil ya que opera de la misma forma);

  - **MiniFB:** Provee funciones del estilo de una librería gráfica sencilla para dibujar una interfaz gráfica. Escribe directamente sobre el driver estándar de *framebuffer* de Linux;

  - **MiniWahoo:** Provee funciones de comunicación con la FPGA utilizando el driver Wahoo desarrollado para Linux;

- **I/O (Entrada/Salida):** Agrupa todas las clases que involucran entrada y salida de datos al sistema, ya sea por acción del usuario o por interacción con otro bloque de *hardware* del sistema:

  - **Touch:** Interpreta los eventos que se generaron por acción del usuario y actúa en consecuencia modificando la configuración del sistema;
    - *void getInput(void)*: Obtiene las coordenadas del punto de la pantalla accionado por el usuario, determina que botón fue presionado y dispara la función correspondiente.

  - **Display:** Dibuja la pantalla a partir de la información de configuración del sistema y las muestras de la señal;
    - *void print(void)*: Refresca la pantalla periódicamente.

  - **FPGA:** Es la clase encargada de mantener la comunicación con la FPGA basándose en la configuración del sistema;
    - *void fetchSamples(void)*: Envía el comando de pedido de muestras, lo envía a la FPGA, y espera hasta recibir las muestras.

- **Configuración:** Almacena datos que representan configuraciones del sistema, ya sean propias del osciloscopio, de la interfaz gráfica o internas de la aplicación:

  - **State:** Almacena el estado y los parámetros de configuración de la interfaz de la aplicación (botones presionados, opciones seleccionadas, etc);

  - **Configuration:** Almacena los parámetros propios del osciloscopio (por ejemplo la escala vertical, el modo de *trigger*, el *coupling* de cada canal, etc);

- **Datos:** Agrupa las clases que almacenan y manipulan los datos correspondientes a las señales que están siendo medidas por el osciloscopio.

  - **Samples:** Almacena las últimas muestras que se obtuvieron de la FPGA;

  - **Mathematician:** Realiza las operaciones matemáticas seleccionadas sobre las muestras de la señal (FFT, diferencia entre ambas señales);
    - *vector getSamples(void)*: Obtiene las muestras ya procesadas de la función matemática seleccionada.

- **Measurer:** Realiza el cálculo de las distintas mediciones sobre la señal (por ejemplo la tensión RMS, la frecuencia, la relación entre señales, etc);
  - *string getMeasure(void)*: Obtiene el valor en texto de la medición seleccionada para graficarlo sobre la pantalla.

Todas las clases aquí definidas son del tipo *singleton*, es decir que la rutina *main()* de la aplicación las instancia una única vez. Luego de crear estos objetos, el *main()* ingresa en un lazo donde repite indefinidamente las llamadas a las funciones principales de los objetos de la capa de Entrada/Salida:

```
int main(int argc, char** argv)
{
	// object instantiation
	Configuration configuration;
	State state;
	Samples samples;
	Touch touch(configuration, state);
	FPGA fpga(configuration, samples);
	Display display(configuration, state, samples);

	// main loop
	while(true)
	{
		touch.getInput();
		fpga.fetchSamples();
		display.print();
	}
	return 0;
}
```

El tiempo que demora el micro controlador en ejecutar un ciclo entero de esta secuencia determina la taza de refresco de la pantalla. Por este motivo, se prestó especial atención durante la etapa de desarrollo en la optimización del código de estas tres funciones.

Un ejemplo claro de esto se da con la implementación que se realizó de un tipo de datos de punto fijo basado en la utilización de números enteros. Por defecto, C++ utiliza datos del tipo *float* para almacenar números fraccionarios. El micro controlador utilizado en nuestro caso (i.MX233) no tiene soporte de punto flotante, por lo que las operaciones entre variables del tipo *float* no pueden ser resueltas por *hardware* y lo hacen por *software*, resultando en un proceso muy poco eficiente. Para sortear esta dificultad se creó en C++ un nuevo tipo de datos que permite el manejo de números fraccionarios utilizando enteros. Para esto se aprovechó una característica del lenguaje C++ que se denomina "sobrecarga de operadores". A modo de ejemplo se muestra a continuación la sobrecarga del operador multiplicación:

```
class fix
{
   private:
       long long value;

   public:
       // Se omite el resto de los operadores para mejor lectura del código.
       fix& operator*=(const fix& other);
};


fix& fix::operator*=(const fix& other)
{
       // Realiza la operación *= entre dos variables del tipo fix utilizando
       // números del tipo entero.
       // value es un entero que representa el valor de punto fijo.
       // FRACTION es la cantidad de bits después de la coma.
       value >>= (FRACTION / 4);
       value *= (other.value >> (FRACTION / 4));
       value >>= (FRACTION / 2);
       return *this;
}

// Caso de uso
fix a = 1.5;
fix b = 3.7;
fix c = a * b; // la multiplicación se realiza utilizando el operador sobrecargado
```

6.2.2 *Softscope*: Interfaz gráfica

A continuación se dará una detallada descripción de la interfaz gráfica del producto. Durante el desarrollo de la misma se trabajó con la comodidad y usabilidad como los factores de mayor importancia. Con ese espíritu se intentó aprovechar la totalidad de la pantalla a la hora de requerir acción del usuario (por ejemplo usando botones grandes, difíciles de errar); y mantener el diseño lo más sencillo posible (por ejemplo usando colores como código del canal que está seleccionado).

La aplicación contiene una única pantalla desde la cual se puede acceder a toda la funcionalidad del osciloscopio, y es la siguiente:

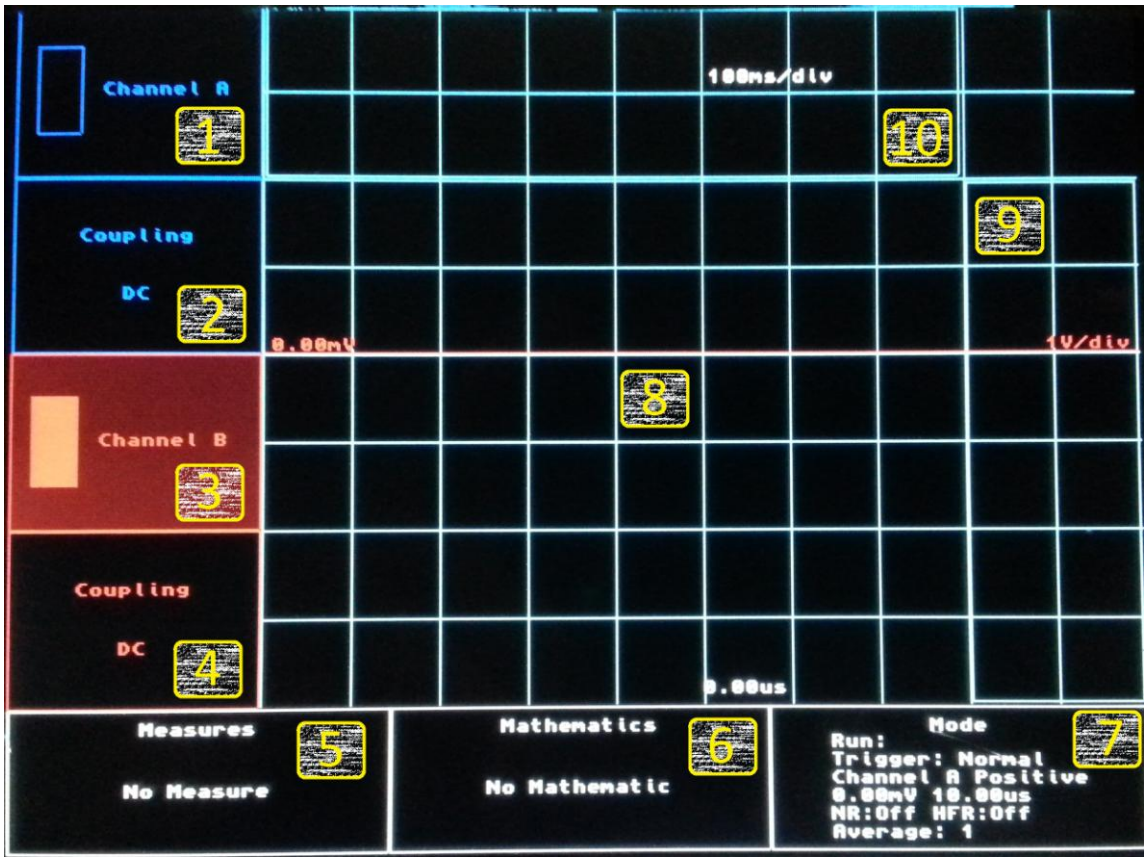


Fig. 6.6: Pantalla principal del osciloscopio - Modo datos

Por comodidad se categorizó la interfaz en tres secciones diferentes. La funcionalidad y comportamiento de los principales elementos de la interfaz es:

- **Sector de 'Selección de canales':** Agrupa los botones que operan sobre cada uno de los canales de manera individual;
  - **1 / 3 (Canal A/B):** Cada canal tiene un pequeño botón de *On/Off* que permite elegir cuál se desea visualizar en pantalla. Además, se puede seleccionar cual de los dos canales es el activo, para las mediciones y operaciones matemáticas, presionando el botón sobre el texto que indica el nombre. Un indicador visual (color del botón resaltado) informa rápidamente cual es la selección actual.
  - **2 / 4 (Coupling A/B):** Estos botones permiten cambiar el tipo de *coupling* deseado para cada canal (AC ó DC).

- **Sector de 'Funcionalidad':** Agrupa los botones que brindan la funcionalidad principal del osciloscopio (mediciones, operaciones matemáticas, configuraciones del *trigger*, etc). Cada uno de los botones en este sector despliega un menú con opciones sobre el sector de **'Datos/Opciones'**. El menú será diferente para cada caso pero funcionalmente se comportan de la misma forma;

- **5 (Measures):** Despliega el menú asociado a las mediciones disponibles en el osciloscopio (frecuencia, valor medio, valor pico, etc).
- **6 (Mathematics):** Despliega el menú asociado a las operaciones matemáticas que puede realizar el osciloscopio (FFT, diferencia entre canales, etc).
- **7 (Scope):** Despliega el menú asociado a las configuraciones propias del osciloscopio (modo de operacion, *trigger*, etc).

- **Sector de 'Datos/Opciones':** Sección principal del osciloscopio que presenta dos modos de operación: modo datos (mostrado en la figura 6.6, abarca los puntos 8, 9 y 10); y modo opciones (mostrado en la figura 6.7);

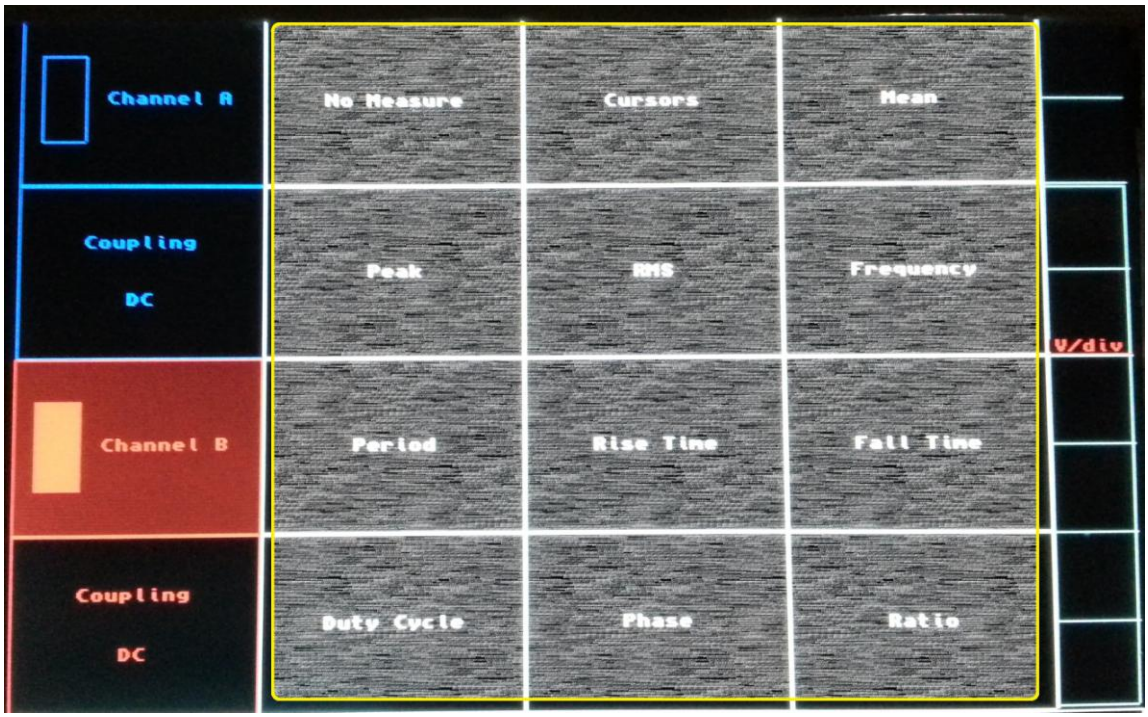


Fig. 6.6: Pantalla principal del osciloscopio - Modo Opciones (Opciones para Measures)

- **Modo datos:** Modo principal de operación del osciloscopio, muestra de datos sobre la pantalla;
  - **8 (Grilla):** Sección donde se muestran las señales medidas. Permite ajustar el *delay* y el *offset* de las mismas arrastrando sobre el eje X, y el eje Y respectivamente.
  - **9 (Escala de tiempo):** Este slider permite ajustar la escala de tiempo del osciloscopio arrastrando sobre el eje X. El valor permitido va desde un mínimo de 1 useg/división hasta un máximo de 1 seg/división. Los valores intermedios son los típicos para un osciloscopio: 1, 2, 5 para cada uno de los multiplicadores.

- **10 (Escala de amplitud):** Este *slider* permite ajustar la escala de amplitud del osciloscopio arrastrando sobre el eje Y. El valor permitido va desde un mínimo de 10 mV/división hasta un máximo de 10 V/división. Los valores intermedios son los típicos para un osciloscopio: 1, 2, 5 para cada uno de los multiplicadores.

- **Modo opciones:** Modo que permite ajustar opciones del osciloscopio o elegir tipo de medición a realizar. Despliega hasta 12 botones en pantalla (mostrados en la figura 6.7 para la funcionalidad de *Measures*) . A continuación se muestran en detalle las opciones disponibles en función de la funcionalidad seleccionada:

Funcionalidad	***Opciones***	***Descripción***
Measures	No measure	Desactiva la medición actual.
	Cursors	Mide las diferencias en los dos ejes entre dos puntos en la pantalla. El usuario debe elige el primer punto sobre la pantalla y arrastrar hasta el segundo.
	Mean	Mide el valor medio del canal seleccionado.
	Peak	Mide el valor pico del canal seleccionado.
	RMS	Mide el valor RMS del canal seleccionado.
	Frecuency	Mide la frecuencia del canal seleccionado.
	Period	Mide el periodo del canal seleccionado.
	Rise time	Mide el tiempo de rise del canal seleccionado.
	Fall time	Mide el tiempo de fall del canal seleccionado.
	Duty cycle	Mide el duty cycle del canal seleccionado.
	Phase	Mide la diferencia de fase entre los dos canales.
	Ratio	Mide la relación de amplitudes pico entre los dos canales.

Funcionalidad	***Opciones***	***Descripción***
Mathematics	No mathematics	Desactiva la operación matemática actual.
	Difference	Calcula la señal diferencia entre los dos canales y la muestra en pantalla.
	FFT	Calcula la FFT del canal seleccionado y la muestra en pantalla.

Funcionalidad	*Opciones*		*Descripción*
Scope	Mode	Run	Configura al osciloscopio en modo 'Run'.
		Stop	Configura al osciloscopio en modo 'Stop'.
		Single	Configura al osciloscopio en modo 'Single'.
		Roll	Configura al osciloscopio en modo 'Roll'.
	Trigger mode	Normal	Configura el trigger para disparo en modo 'Normal'.
		Automatic	Configura el trigger para disparo en modo 'Automatic'.
	Trigger channel	Channel A	Configura el trigger para disparo con el Canal A.
		Channel B	Configura el trigger para disparo con el Canal B.
	Trigger slope	Positive	Configura el trigger para disparo con pendiente positiva.
		Negative	Configura el trigger para disparo con pendiente negativa.
	Trigger noise reject	On	Activa la funcionalidad de 'Noise Reject' en el trigger.
		Off	Desactiva la funcionalidad de 'Noise Reject' en el trigger.
	Trigger HF reject	On	Activa la funcionalidad de 'High Frequeny Reject' en el trigger.
		Off	Desactiva la funcionalidad de 'High Frequency Reject' en el trigger.
	Trigger level / Holdoff	-	Configura el nivel de disparo del trigger y el holdoff. El usuario debe ajustar los valores de manera manual tocando la pantalla y arrastrando (eje X: holdoff - eje Y: trigger level).
	Average	1 - 1024	Configura la función de promedio que se realiza sobre los datos en pantalla. Los valores disponibles son todas las potencias de dos entre 1 y 1024.

6.2.3 *Softscope*: Plan de prueba y depuración

Para la programación de la aplicación, como para el proyecto en general, se utilizó la metodología de desarrollo ágil. Es decir que se trabajó siempre sobre un producto en funcionamiento para una detección temprana de problemas y una mayor productividad. De esta manera se fueron depurando los inconvenientes de cada funcionalidad a medida que se la fue incorporando al producto final.

Utilizar Linux facilita considerablemente el trabajo con este tipo de metodologías. Una de sus grandes ventajas radica en el hecho de que es muy sencillo migrar una aplicación de una

arquitectura a otra, siempre y cuando se cuente con los *drivers* necesarios. Esto permite concentrar el fuerte del desarrollo en un ambiente controlado y conocido (como puede ser una PC de escritorio con menos variables para ajustar), depurar en profundidad el código y su funcionalidad, y migrar a la plataforma final una vez que se sepa que la aplicación es confiable.

El desarrollo se inició en una PC de escritorio corriendo la conocida distribución de Linux llamada Ubuntu con entradas y salidas simuladas. Esto brinda, además de lo ya mencionado, una ventaja importante y es que el tiempo de compilación en una PC con un procesador moderno es despreciable, mientras que en el i.MX233 puede ser de varios minutos.

La segunda etapa del desarrollo consistió en migrar el *software* a la plataforma de desarrollo (OLinuXino i.MX233 / LobsterBoard) luego de que la aplicación esté completamente funcional. En este caso se utilizaron *drivers* estándar de Linux para mouse y video compuesto para capturar los eventos de entrada y presentar información en lugar de las simulaciones que se hicieron en la etapa anterior.

Finalmente se reemplazan los *drivers* de entrada y salida por los desarrollados para la arquitectura específica (de controlador táctil y *display* LCD).

En cada etapa del desarrollo se realizan pruebas extensivas de la interfaz. Dado que la aplicación presenta una única pantalla, resulta sencillo repasar todos los menús y opciones buscando posibles errores.

Una vez finalizado el desarrollo se vuelve a repasar el código con vistas a garantizar su calidad. Se recorre cada archivo de código chequeando las definiciones de clases, constantes, estructuras y métodos de uso interno. Algunas de los controles que se realizan son por ejemplo: la coherencia de los argumentos con el código en cada función (que no se reciban parámetros que no son usados); que no haya código definido en archivos de tipo *header (.hpp)*; que no se haga uso de números arbitrarios; que no haya una mala administración de la memoria pudiendo ocasionar una falla por falta de la misma (*memory leak)*; en el caso de que la función retorne parámetros se revisa que para cada condición el valor de retorno esté garantizado; que no se devuelvan valores aleatorios o basura de memoria en ninguna circunstancia.

VII. CONSTRUCCIÓN DEL PROTOTIPO

Para llevar a cabo la construcción del prototipo se procedió con el pedido de componentes y módulos necesarios para luego concretar la manufactura de las placas.

7.1 Definición de los módulos

Los módulos del prototipo han sido seleccionados tomando en consideración la funcionalidad independizada de los mismos y su correspondiente puesta a prueba de forma individual. A continuación se detallan los módulos y las placas resultantes de cada uno de ellos:

- **Módulo de Etapa de Entrada y ADC**
  - KrakenBoard**: Esta placa recibe la señal de entrada y la condiciona para poder ser muestreada. El último de sus componentes es el conversor analógico digital por lo que la misma se compone mayoritariamente de señales analógicas. La KrakenBoard es un diseño propio; el PCB fue fabricado y ensamblado manualmente en la universidad.**
- **Módulo de FPGA**
  - EVM Cyclone**: Esta es la placa de evaluación de la FPGA *Cyclone I* que contiene todos los pines de prueba expuestos en 4 cabezales de 30 pines cada uno. A su vez, esta placa posee el conector *JTAG* y *ASMI* que permiten programar la FPGA. Es un diseño original del Prof. Daniel Jacoby.**
- **Módulo de Microprocesador**

  El modulo de microprocesador requiere de una aclaración extra. Como se comentó en el inciso correspondiente a la Ingeniería de Hardware, el desarrollo del equipo se realizó utilizando la placa de evaluación OLinuXino del procesador i.MX233. En paralelo se trabajo en el desarrollo de una versión *stripped-down* del OLinuXino llamada LobsterBoard, es decir una versión que tenga solo los componentes mínimos necesarios para el funcionamiento en el equipo. Esto se pensó como un componente

para el diseño final del equipo y no como parte del prototipo inicial. Si bien se lograron resultados satisfactorios con LobsterBoard, por falta de tiempo se decidió posponer su integración al sistema en reemplazo del OLinuXino. En el estado de desarrollo actual, la LobsterBoard pasó todas las pruebas y se ha ejecutado Linux portado sobre la misma, pero queda pendiente su integración al resto del sistema.

- LobsterBoard**: La placa del microprocesador i.MX233 junto con la memoria RAM y la memoria Flash SPI como fue especificado en la ingeniería de detalle. Es un diseño propio y sus detalles se especifican en el próximo inciso. La fabricación del PCB fue delegada a la empresa china PCB Cart[11], mientras que el ensamble fue hecho en la universidad.**

- OLinuXino[12]**: La computadora *single board* distribuida por Olimex con diseño de hardware y software de código abierto (*Open Source*) de bajo costo. Se utiliza la versión con el microprocesador i.MX233 llamada iMx233-OLinuXino-MAXI[13]. Esta placa presenta puertos y periféricos de propósito general configurables por lo que fue utilizada como placa de desarrollo del i.MX233.**

- **Módulo de Interfaces**

  - HCI Board**: Dado que el controlador *touchscreen* no viene integrado en el iMX233, se utilizó un integrado con su conector en la HCI Board. A su vez, el conexionado del *display* también se encuentra en esta placa. La HCIBoard es un diseño propio; el PCB fue fabricado y ensamblado manualmente en la facultad.**

- **Display y Touchscreen**

  - **La pantalla y el *touchscreen* utilizados en el prototipo condice con las especificaciones definidas previamente:**

---

[11] www.pcbcart.com

[12] http://www.olimex.com/Products/OLinuXino

[13] http://www.olimex.com/Products/OLinuXino/iMX233/iMX233-OLinuXino-MAXI/open-source-hardware

- **Tamaño: 5,6 pulgadas**
- **Driver TFT**
- **Resolución: 640x480 RGB**
- **Touchscreen: Resistivo**
- **Modelo: AT056TN53**
- **Marca: INNOLUX Display Corporation**
- **Fabricante: Shenzhen Control Electronics**

- **Gabinete**
  - **El gabinete utilizado para la construcción del prototipo se halla en el orden de dimensiones establecidas en las especificaciones con ciertos márgenes que facilitan las pruebas iniciales. En el inciso de diseño mecánico se detalla la construcción del mismo.**
- **Puntas de Osciloscopio**
  - **Se utilizaron puntas de osciloscopio con ancho de banda no limitante de 100MHz que se encontraban disponibles en el laboratorio donde se realizó la confección del prototipo.**
- **Batería**
  - **El consumo estimado máximo del sistema con todos sus componentes es de 1500mA (ver Anexo), por lo que se realizó una investigación de modelos de baterías con capacidad suficiente para cumplir con la especificación de 4 horas de autonomía. Para realizar las pruebas con el prototipo, se tomó en cuenta contar con un proveedor cercano y no sufrir percances en los tiempos de entrega. Finalmente, se confeccionó el prototipo con una batería de ión litio de 3.7V y 5800mAh *UltraFire*.**
- **Transformador**
  - **Dado el consumo estimado de 1500mA máximo y la característica del microprocesador i.MX233 que permite realizar la carga de la batería mientras está siendo utilizado, se dimensionó el transformador de tal**

manera que la potencia entregada por el mismo pueda tanto alimentar a todo el dispositivo como tener un excedente para cargar la batería. El transformador utilizado es de 5V y 3000mA.

El esquema a continuación representa los módulos mencionados y su vínculo dentro del sistema:

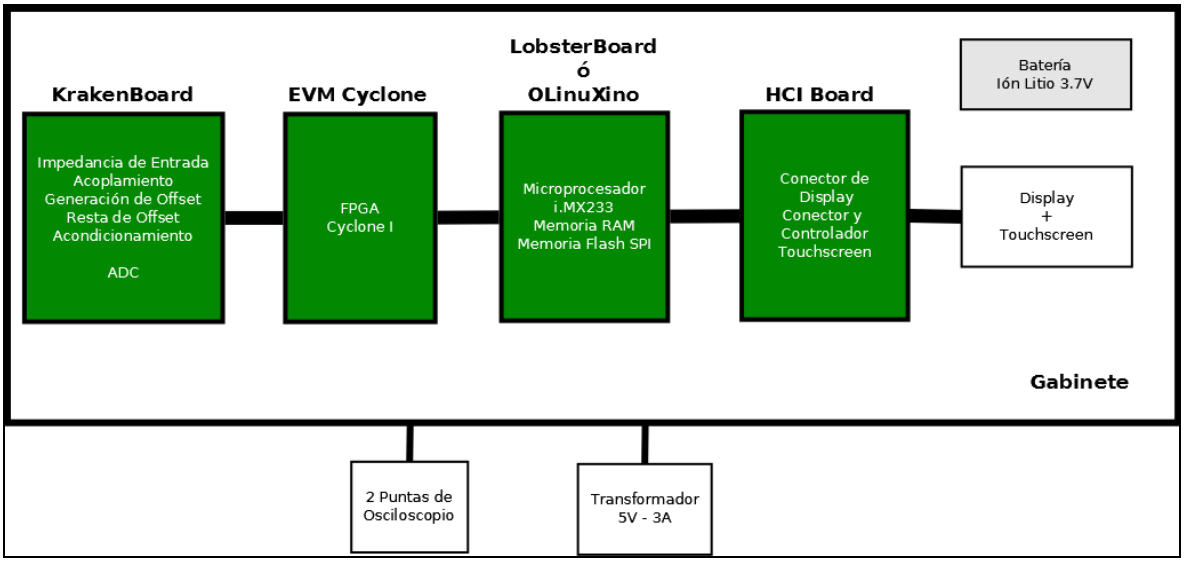


Fig 7.1: Módulos físicos del sistema

7.2 Diseño de los circuitos impresos

7.2.1 KrakenBoard

La consideración de diseño más incidente en el proceso de desarrollo de esta placa fue garantizar la estabilidad de la señal de entrada en el camino hacia el muestreo. Bajo esta premisa, se tomaron los siguientes recaudos:

- **La señal no atraviesa vías de una capa hacia la otra con el objetivo de no introducir capacidades parásitas propias de los cambios de medio.**
- **La disposición de los componentes es la más próxima posible para impedir caminos inductivos parásitos (esto con la limitación de armado de la misma que exige ciertas distancias mínimas).**
- **El ancho de las pistas es el máximo posible a partir de los tamaños limitantes del *footprint* de los integrados utilizados.**
- **Planos de masa recubren gran parte de una de la capa inferior de la placa.**

- **Para facilitar la etapa de depuración, el camino de la señal y de alimentación de los componentes de cada canal se encuentran independientes.**

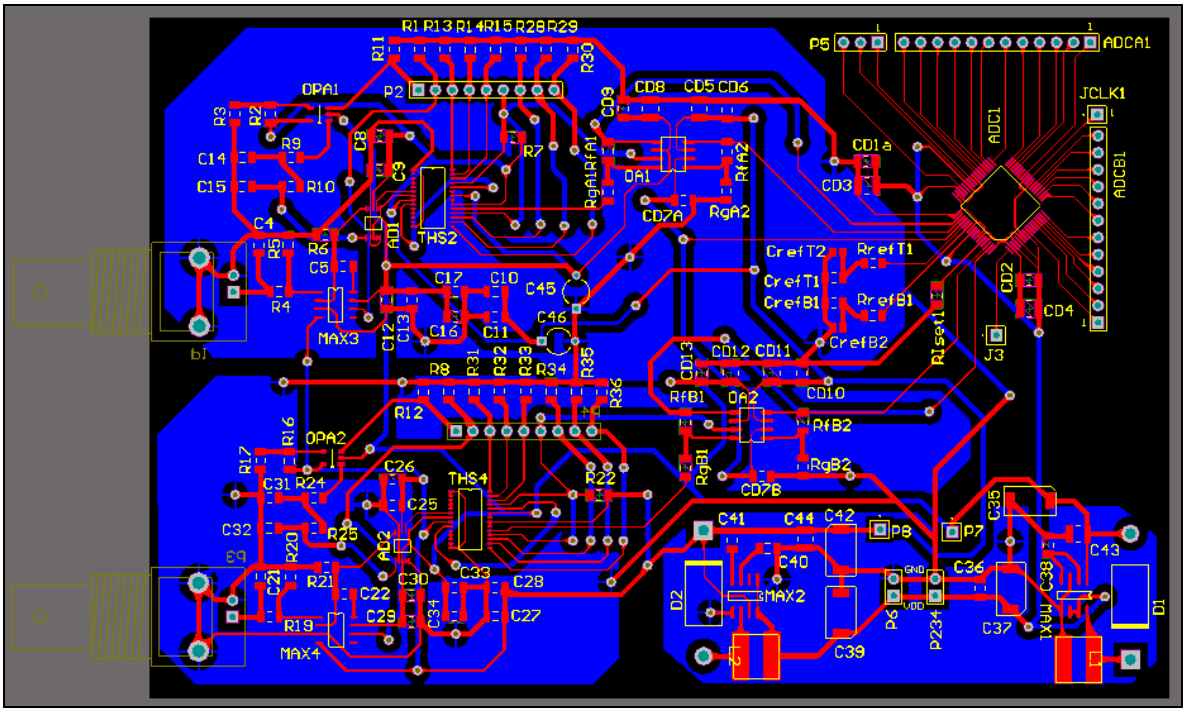


Fig 7.2: Esquemático del PCB de la KrakenBoard

Como se muestra en el siguiente diagrama, los dos canales ocupan espacios separados de la placa. Cabe destacar, que los planos de masa se encuentra individualizados para cada canal, para la sección de alimentación y que no interfieren en la sección del convertidor analógico digital.

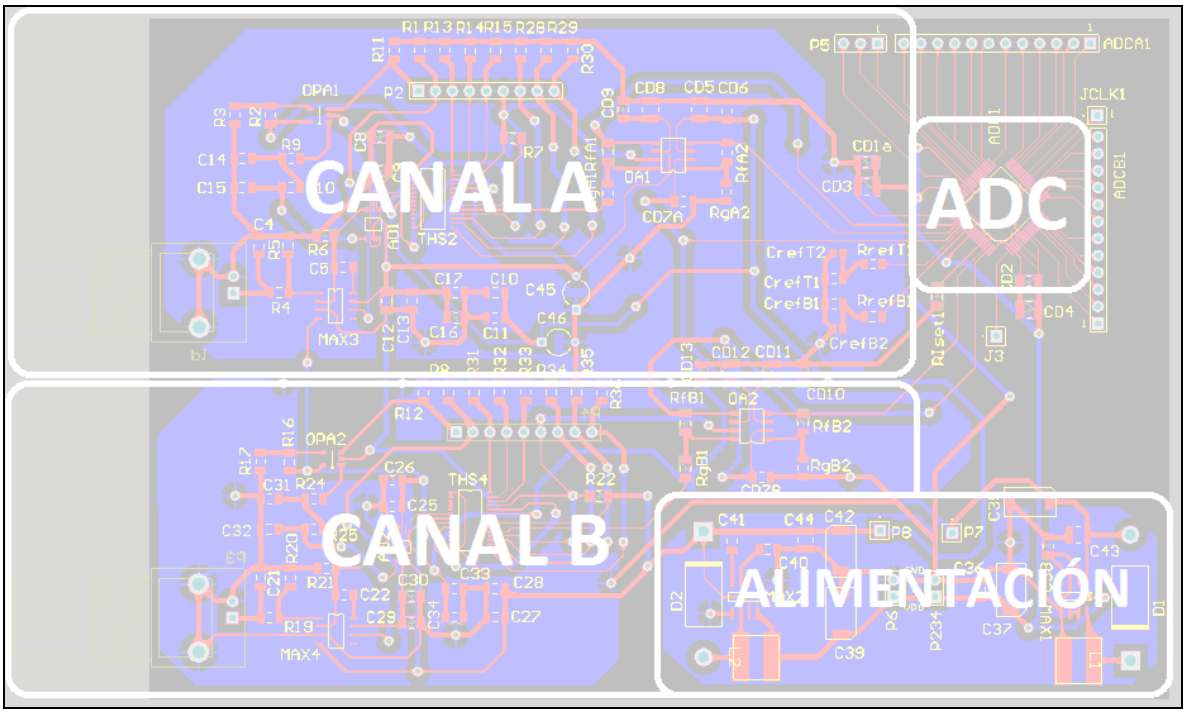


Fig. 7.3: Esquemático del PCB de la KrakenBoard - Delimitación de zonas

Por último, se menciona que todas las interfaces de salida y entrada de la placa se encuentran separadas por sección pero agrupadas para lograr emprolijar el cableado de cada una de ellas.

Al momento de la integración con la FPGA, fue necesario agregar resistencias en serie para adaptar la impedancia en la comunicación entre el ADC y la FPGA. El método para calcular estas resistencias se detalla en el anexo.

7.2.2 LobsterBoard

A continuación (fig. 7.4) se muestra el diseño del PCB correspondiente a la placa del procesador i.MX233 denominada “LobsterBoard”, en su segunda versión:

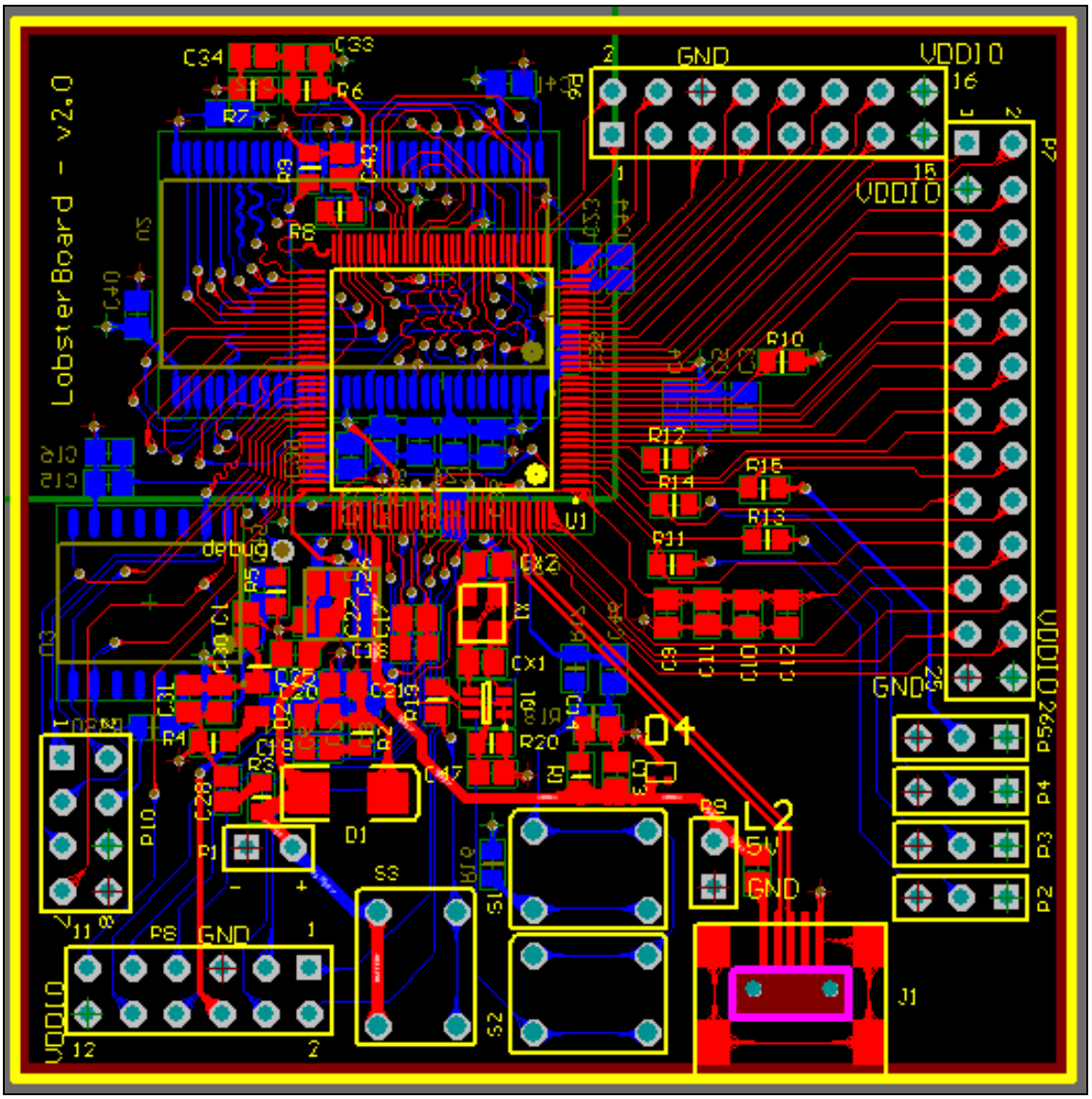


Fig. 7.4: PCB de "LobsterBoard v2.0" del microprocesador iMX233

Se trata de un PCB de 60mm x 60mm, diseñado en cuatro capas**, que contiene el procesador i.MX233, memoria DDR RAM y memoria Flash SPI.. Se trata de la segunda versión pues la primera, realizada en dos capas, no superó las pruebas de comunicación de RAM y comunicación USB por problemas de *timing* e impedancia de pistas de alta frecuencia.**

La figura 7.5 muestra las secuencia de capas de la placa:

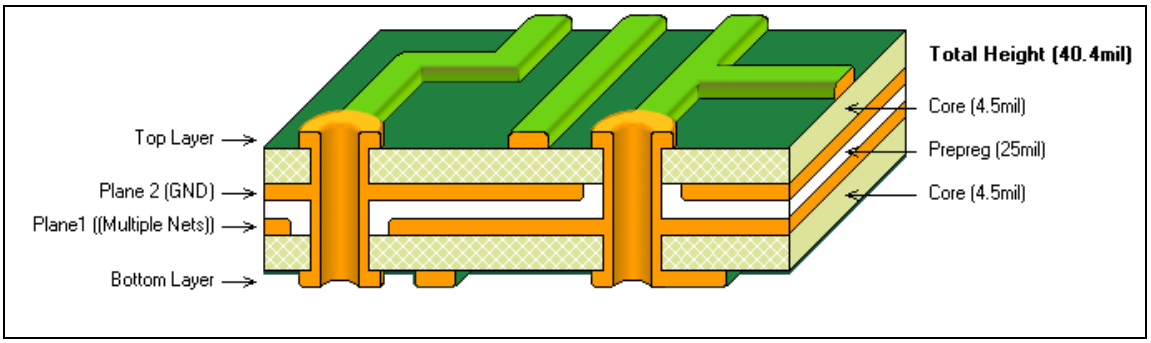


Fig. 7.5: Diagrama de capas de la placa LobsterBoard v2.0

Esta configuración de capas se eligió por múltiples motivos. Por un lado, la presencia de un plano de masa simplifica el trazado de pistas y permite un comportamiento electromagnético favorable, ya que todas las señales tienen la posibilidad de realizar un camino de retorno óptimo (la corriente se distribuye por el plano de la mejor manera posible). Por otro lado, hay un plano de alimentación que está partido en dos secciones (esa separación se observa en las líneas verdes en la figura 7.4). Una sección tiene la tensión de alimentación de la RAM, cuya correcta provisión es fundamental para el funcionamiento del circuito, y la otra tiene, convenientemente, los 3.3V de salida que se utilizan para alimentar periféricos. Finalmente, la distancia reducida entre la capa superior y el plano de masa (4.5 mils) facilita el diseño de la impedancia de las pistas de USB. El espesor del dieléctrico central (25 mils) se definió por recomendación del fabricante, para que el espesor total de la placa sea de aproximadamente 1mm.

El *layout* respeta, en general, la disposición natural que surge de los pines del procesador. Se observa el procesador en la parte superior izquierda, y la memoria RAM en el mismo lugar pero en la otra faz. Este *layout* está inspirado en la placa *open-source* del mismo procesador llamada *AndroidStamp*[14], mencionada en la descripción del esquemático (sección 5.4).

La Flash SPI se encuentra abajo y a la izquierda del procesador, y en la faz opuesta. De esta manera se minimiza la distancia promedio entre los pines de ambos integrados.

La conexión entre el procesador y las memorias se realizó con largos de pista ecualizados para que haya una diferencia de no más de 500mil entre las pistas

[14] http://linuxencaja.net/wiki/AndroidStamp

del mismo bloque. Esta diferencia es menor al 1% de la longitud de onda de las señales de 133MHz en el material FR-4 de la placa, lo cual asegura que no habrá diferencias de *timing* entre las pistas de conexión de memoria.

El cristal se conectó de la manera más independiente posible al resto de los componentes, pues es una parte crítica en cuanto a interferencia electromagnética.

Los desacoples se repartieron por las distintas zonas para minimizar la distancia a los distintos componentes. Se combinan capacitores electrolíticos y cerámicos para un mejor comportamiento electromagnético.

El conector USB se ubicó en el sector inferior derecho, con distancia entre sus pistas (que son de alta velocidad) y el resto de las conexiones. Siguiendo la recomendación de Freescale, se seleccionó un ancho de pista tal que, considerando la distancia del plano de masa, la impedancia diferencial entre las líneas de datos sea de aproximadamente 90 ohms.

El prototipo se envió a fabricar en material FR-4 de siguiendo el esquema de capas antes descripto. Las pistas más largas son de aproximadamente 1000mils, por lo que la relación ancho/largo más desfavorable ronda 1/150, lo cual es un valor aceptable para consideraciones de compatibilidad electromagnética[15].

7.2.3 HCI Board

Esta placa posee la interfaz de conexionado entre el *display* y el módulo del microprocesador. Esta interfaz también realiza una ganancia de corriente para alimentar el *backlight* de la pantalla mediante el integrado TPS61092 de Texas Instruments. Dado que las señales de manejo del *display* son de alta frecuencia (orden de los 10MHz), se tomó el criterio de colocar planos de masa tanto en la capa superior como en la capa inferior de la placa y vincularlos a través de vías en puntos clave.

El *HCI Board* también posee la interfaz de conexionado entre el *touchscreen* y el controlador de *touchscreen* que luego será enviado al módulo del

---

[15] http://www.cec.cubaindustria.cu/contenido/jornadaVII/3_3.pdf

microprocesador mediante I2C. El integrado utilizado en este caso es el TSC2003 que es específico para dispositivos *touchscreen* de 4 cables (*4-wire*).

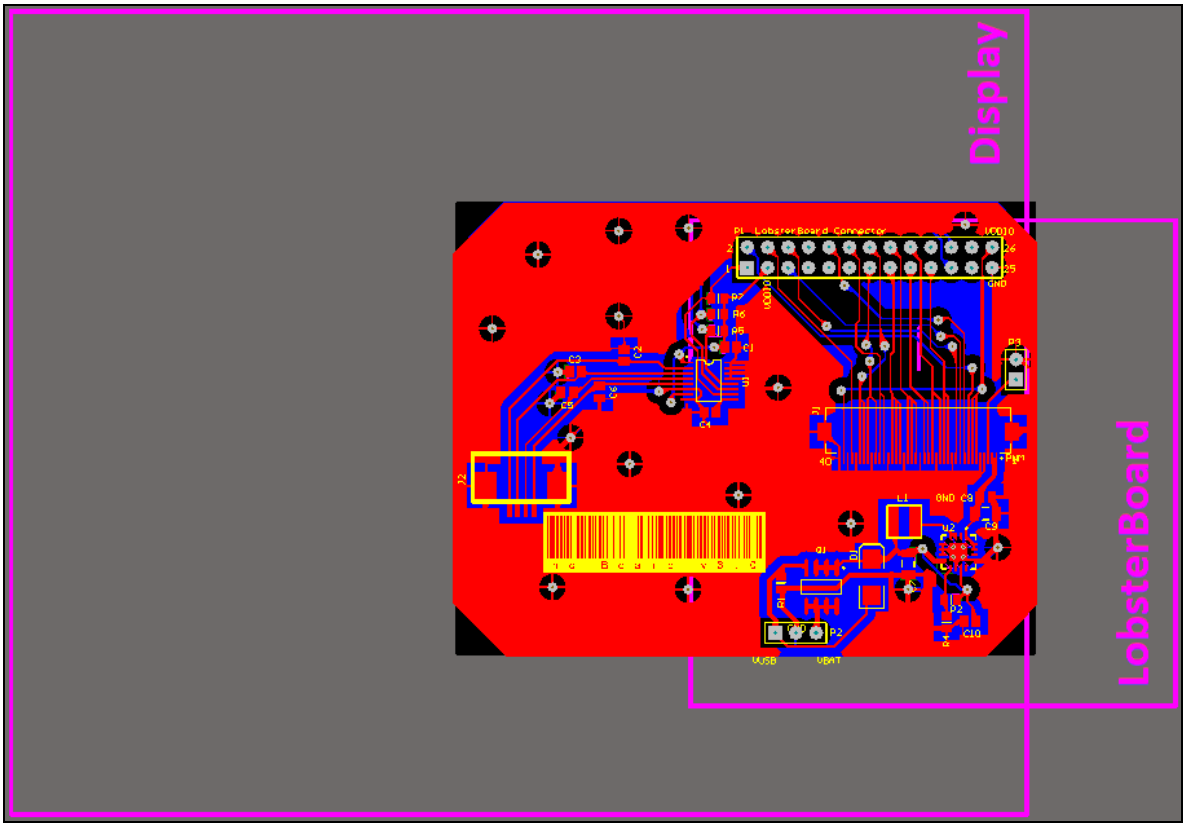


Fig. 7.6: Disposición física de la LobsterBoard (OLinuXino), el display y la HCI Board.

Cabe mencionar que los recuadros remarcados con violeta representan la disposición física de la placa *LobsterBoard (OLinuXino)* y el *display*. El cabezal de 26 pines que figura en la parte superior fue diseñado para coincidir con el cabezal de la placa *LobsterBoard* lo que permite anexar el HCI Board fácilmente a la placa del microprocesador. A su vez, la disposición de los elementos de la placa logra no demandar más espacio dentro del gabinete de lo que ya demandan el display y el módulo del microprocesador.

7.3 Diseño mecánico

La principal consideración a la hora del diseño mecánico es la seguridad del operador. Debido a que el equipo se va a usar con las manos y puede ser conectado a tensiones de línea es importante que quién lo utiliza esté eléctricamente aislado de los circuitos que residen en su interior. Típicamente se utilizan dos materiales para cumplir con este propósito, plástico o madera.

Se decidió utilizar un gabinete plástico estándar por una cuestión de costos y porque el plástico resulta más sencillo de manipular que la madera (perforaciones para empotrar el *display*, la llave de encendido, los conectores, el *led* indicador, etc). El gabinete elegido es el PB100 del fabricante *Chillemi*; sus dimensiones son 253x176x102mm (largo x ancho x alto). La decisión de usar un modelo estándar y no un diseño a medida radica en que al ser un prototipo (que va a sufrir modificaciones antes de llegar al diseño final) no se justifica el costo que representa el diseño y la producción de la matriz del mismo.

Cabe aclarar que las dimensiones de este gabinete se encuentran por fuera de las especificaciones (200x150x50mm). El prototipo del equipo consiste actualmente en cuatro placas y una pantalla de siete pulgadas, lo que hace que el espacio necesario para fijar y conectar todos los elementos entre sí sea mayor a lo esperado. Una de los principales cambios que se esperan para el diseño final del equipo es integrar las placas en una, de manera de reducir su cantidad y con ello el espacio requerid y la cantidad de conectores. Con esto en vista, es de esperarse que se cumpla con la especificación sin problemas.

En cuanto al peso del equipo, se fijó una especificación de 1 kilogramo máximo de peso, valor que se cumple de manera holgada.

A continuación se muestra una imagen del prototipo construido:

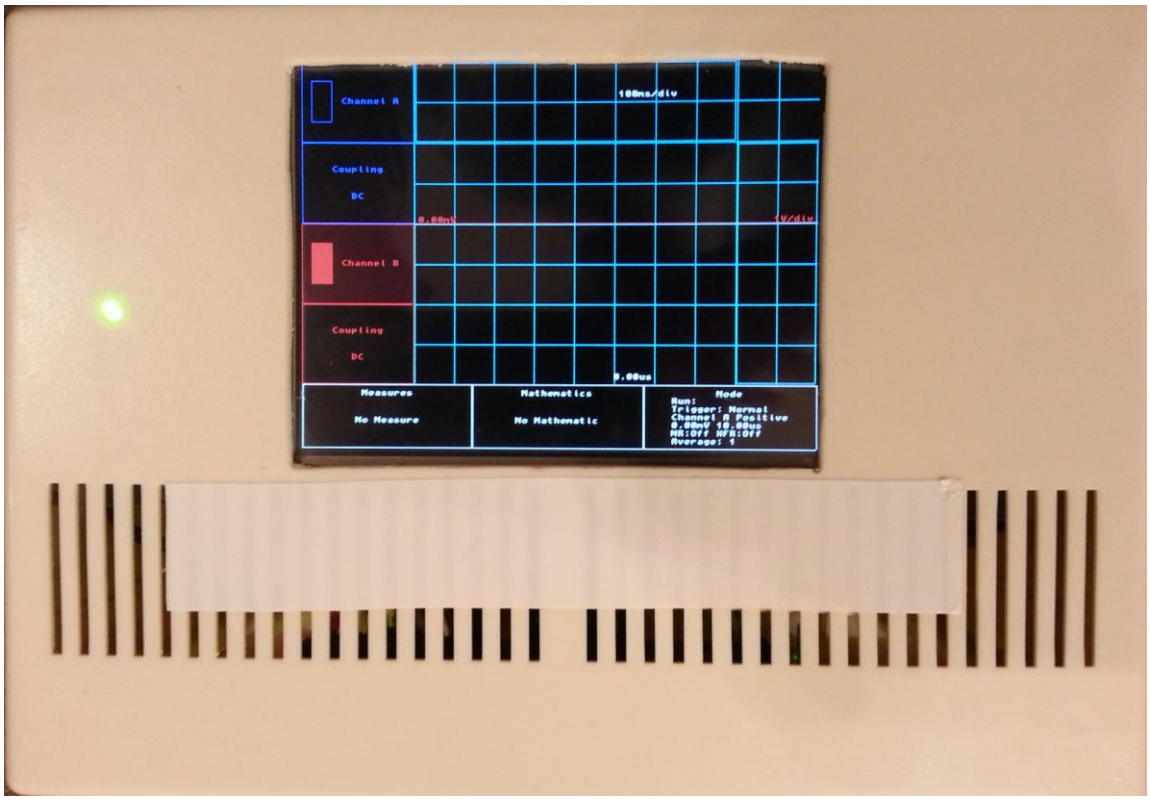


Fig. 7.7: Aspecto exterior del prototipo construido

El proceso de construcción requirió adaptar el gabinete de acuerdo a las necesidades del equipo. Primero se realizó el calado de la tapa superior del gabinete para la pantalla, con la precaución de que el marco sea menor a su tamaño efectivo para que esté correctamente asegurado. Se utilizó una estructura de acrílico montada en el interior del gabinete para fijar la pantalla y evitar que se mueva al ser presionada. Junto al panel LCD, se perforó el plástico para colocar el LED indicador de encendido del equipo. En los paneles laterales se realizaron sendas perforaciones para exponer por un lado los conectores BNC correspondientes a las entradas, y por el otro el conector de alimentación y la llave de encendido. Dado que el gabinete está construido enteramente de un plástico maleable resultó sencillo realizar estas modificaciones.

Fig. 7.8: Conectores BNC para entrada de las señales (arriba); conector para alimentación y botón de encendido (abajo)

Las placas fueron fijadas a los distintos paneles del gabinete utilizando tornillos y porta-placas adhesivos. La disposición de las distintas placas fue elegida en base a la cercanía de las conexiones. Se prestó especial atención a la mantenibilidad a la hora de la construcción, particularmente se buscó minimizar la cantidad de cables que conectan la base con la tapa con el objetivo de facilitar la apertura del mismo. A continuación se describe en detalle la disposición de cada placa:

Tapa superior del gabinete: El *display LCD* está asegurado a la tapa superior mediante a una estructura de acrílico que impide su movilización. Atornillado a la tapa superior está también la placa *HCI Board* que se conecta al *display LCD* mediante a un bus flexible tipo FPC. A su vez, la *HCI Board* utiliza un *bus* plano pin-a-pin para conectarse con la placa de desarrollo olinuxino que se encuentra fijada a la tapa superior mediante a porta-placas adhesivos. El diseño final reemplaza la placa olinuxino por la *LobsterBoard* y permite una conexión directa entre las dos placas sin necesidad de un *bus* intermedio.

Base del gabinete: Fijado a la base del gabinete mediante a dos tornillos se encuentra la *KrakenBoard*, que expone los conectores BNC a través de los orificios creados en el panel lateral.

Panel trasero: Por último, la placa de desarrollo *EVM Cyclone* se asegura al panel trasero mediante al uso de porta-placas adhesivos.

Los conectores utilizados fueron *buses* planos hembra-hembra ya que permiten la flexibilidad y el orden necesario. Los detalles de conexión se encuentran documentados en el Anexo.

VIII. VALIDACIÓN DEL PROTOTIPO

8.1 Validación del sistema

Cada uno de estos procesos de validación se llevaron a cabo para cada uno de los canales del osciloscopio. Se asume que los errores que introducen los equipos generadores de onda son despreciables frente a los errores del osciloscopio, por lo tanto el error medido coincidirá con el error del sistema.

8.2 Plan y protocolos especiales de medición

Para llevar a cabo la validación se definieron diversas señales de prueba (A - G) y junto a ellas se diseñaron casos de prueba que pongan en evidencia las especificaciones del equipo.

Las señales de prueba se presentan a continuación (abarcan distintas frecuencias y amplitudes):

Señal de prueba	Tipo	Frecuencia	Amplitud
A	Continua	-	2 V
B	Sinusoidal	0.5 Hz	2 Vpp
C	Sinusoidal	50 kHz	2 Vpp
D	Sinusoidal	10 MHz	2 Vpp
E	Sinusoidal	1 kHz	20 Vpp
F	Sinusoidal	1 kHz	20 mVpp
G	Continua	-	0 V

Los casos de prueba que se utilizaron son los siguientes:

1.A - Validación de coupling *****

> **Colocar la señal G con escala vertical de 1V/div y prender el *coupling* AC del canal. Es de esperarse que el resultado obtenido no sea de 0 V sino**

que exista un pequeño *offset* de continua producto de la etapa analógica (amplificadores operacionales, etc). Esta medición nos permite determinar el nivel de *offset* que presenta el sistema por diseño.

1.B - Validación de coupling *****

Colocar la señal A con escala vertical de 1V/div y prender el *coupling* AC del canal. Verificar que la medición de valor Medio de la señal utilizando las herramientas de medición provistas por el osciloscopio figura como "0V".

***Escala vertical de 1V/div, configuración de *trigger* automático y escala temporal de 500µseg/div.**

2.A - Validación de escala temporal ******

Colocar la señal B con escala temporal mínima de 1 seg/div y proceder con la medición del periodo de la señal utilizando las herramientas de medición provistas por el osciloscopio. Debería figurar "2 seg" en pantalla con cierto margen de error admisible.

2.B - Validación de escala temporal ******

A continuación, colocar la señal C con escala temporal máxima de 1µseg/div y realizar la medición de periodo. El periodo debería ser "20µseg".

3.A - Validación del ancho de banda******

Colocar la señal B con escala temporal mínima de 1 seg/div y proceder con la medición del valor RMS de la señal. Debería figurar "0.707V" en pantalla.

3.B - Validación del ancho de banda******

Luego colocar la señal D y modificar la escala temporal a 1µs/div y comprobar que la indicación de valor RMS es la misma que la anterior "0.71 V" atenuados 3dB, es decir "0.502 V" (la caída de 3 dB se debe a que en 10 MHz, la frecuencia de la señal D, se encuentra la frecuencia de corte del osciloscopio).

****Coupling* AC, configuración de *trigger* normal y escala vertical de 500mV/div.**

4 - Validación del rango de entrada permitido ***

Colocar la señal E con escala vertical de 10V/div y verificar que la señal presente en pantalla no se encuentra distorsionada ni recortada. Como complemento de esta validación se puede utilizar la herramienta matemática de FFT del osciloscopio para verificar que no haya presencia significativa de armónicos en la señal.

5.A - Validación de escala vertical ***

Colocar la señal E con escala vertical de 10V/div y verificar que la medición de RMS es de "7.07V".

5.B - Validación de escala vertical ***

Colocar la señal G con escala vertical de 5mV/div y medir el valor medio de la señal. El resultado será el piso de ruido del sistema y determinará cuál es la escala vertical mínima que devuelve mediciones válidas.

***Coupling* AC, configuración de *trigger* automático y escala temporal de 500µseg/div.**

8.3 Mediciones

A continuación se muestran las mediciones realizadas para cada uno de los casos de prueba planteados en el apartado 8.2:

Caso	Valor esperado	Canal A: Valor medido	Canal A: Error porcentual	Canal B: Valor medido	Canal B: Error porcentual
1.A	0 V	-62 mV	**62 mV***	31 mV	**31 mV***
1.B	-62 mV / 31 mV	- 42 mV	**20 mV***	62 mV	**31 mV***
2.A	2 seg	1.983 seg	**0.83 %**	1.983 seg	**0.83 %**
2.B	20 µseg	20.16 µseg	**0.83 %**	20.16 µseg	**0.83 %**
3.A	0.707 V	0.701 V	**0.85 %**	0.68 V	**3.82 %**

3.B	0.502 V	0.516 V	**2.78 %**	0.554 V	**10.3 %**
5.A	7.07 V	7.31 V	**3.39 %**	7.42 V	**4.95 %**
5.B	-	39 mV	**-**	24 mV	**-**

***Los valores indicados corresponden al error absoluto y no porcentual.**

La medición número 4 se ejemplifica con la siguiente captura de pantalla del osciloscopio:

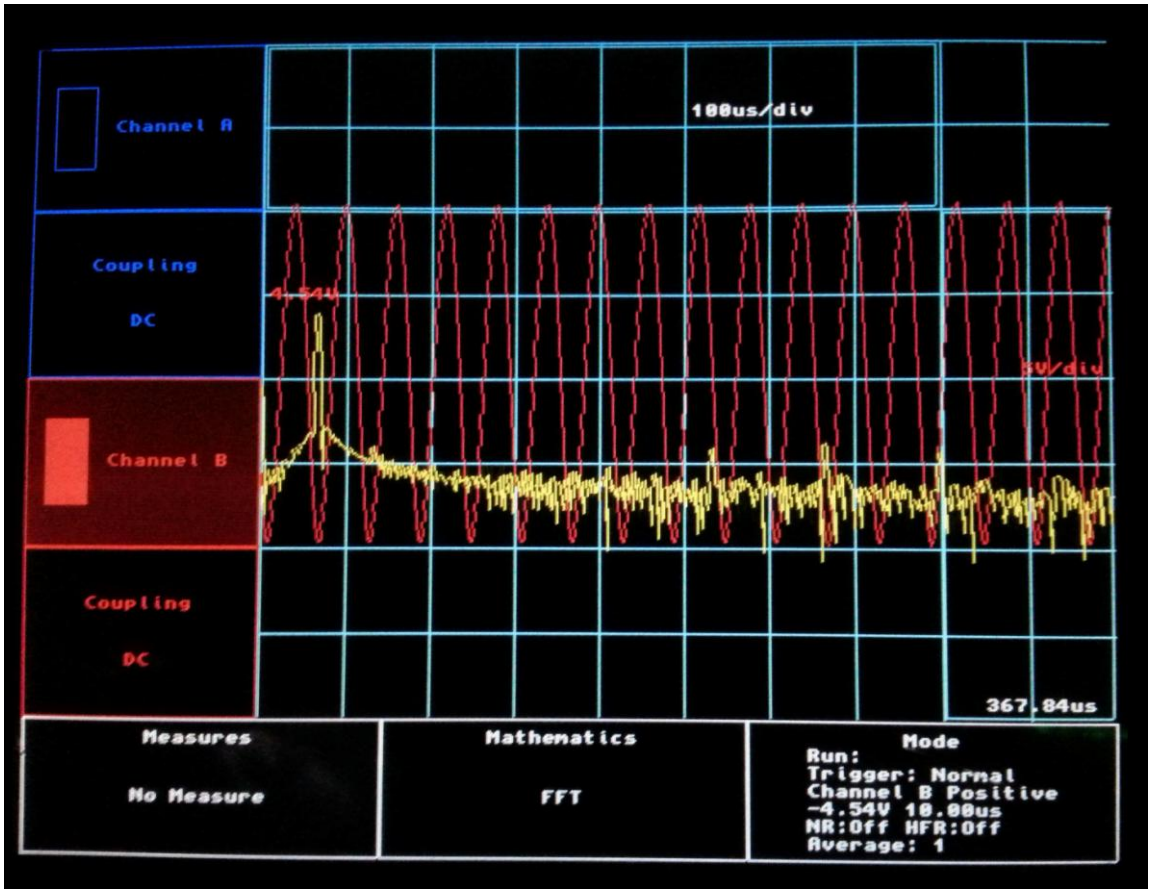


Fig 8.1: Captura de pantalla del caso de prueba 4

8.4 Análisis de los resultados

Las pruebas realizadas sirven tanto para validar el sistema como para caracterizar algunos de parámetros del mismo como por ejemplo el piso de ruido y el nivel de continua.

A partir del caso de prueba 5.B **se obtiene el piso de ruido del osciloscopio para el ambiente de laboratorio donde se tomaron todas las mediciones. Este ruido es inducido al sistema desde el exterior (interferencia electromagnética, ruido de línea, etc.). Los valores obtenidos de 39mV y 24mV para ambos canales están dentro de lo esperado incluso para un osciloscopio de primera línea en el mismo ambiente de medición.**

El nivel de continua del sistema es aquél *offset* de continua que se presenta a la salida del sistema producto de los dispositivos semiconductores (transistores, operacionales, etc.) en la etapa de entrada especialmente. Con el caso de prueba 1.A **se obtienen los valores para ambos canales.**

El resto de los casos de prueba validan las características temporales y de amplitud del sistema. En los casos 2.A **y** 2.B **se verifica el error del sistema en la escala temporal. Como se puede ver en los resultados el error obtenido en este caso es menor al 1% para ambos canales.**

En los casos 3.A**,** 3.B **y** 5.A **se valida al sistema en la escala de amplitudes. Lo primero que se observa en este caso es que el canal B presenta un error mayor al canal A, esto probablemente sea debido a diferencias en los componentes utilizados, ya sea por variación dentro del rango de tolerancia admitido o por imperfecciones en los integrados. Otro punto interesante de analizar es lo que sucede con las mediciones** 3.B**. La señal de entrada utilizada en este caso es de 10 MHz. Por especificación, los componentes de la placa de entrada (*KrakenBoard*) están limitados a un ancho de banda de 10MHz, y algunos a 20MHz. Teóricamente si hubiera una frecuencia de corte bien definida en 10 MHz, con un único polo, la atenuación sufrida en esta frecuencia sería de 3dB. En la práctica, la presencia de polos en frecuencias más altas como 20 MHz hace que la caída no sea exactamente de 3 dB. Esto explica los altos errores encontrados en las mediciones** 3.B**.**

Otro punto de interés con respecto a la validación del sistema contra señales de alta amplitud es la distorsión que se genera. La medición 4 **busca retratar esto. La imagen 8.1 muestra los resultados de dicha medición. Se observa una señal de 20 Vpp de amplitud y su respectiva FFT. Como puede verse, el espectro**

obtenido carece de armónicos más allá del principal. La distorsión que pueda introducir el sistema es despreciable.

Analizando los resultados de los casos de prueba se puede concluir que el sistema se comporta de la forma esperada dentro de rangos de error razonables; la mayor discrepancia ocurriendo en el extremo superior de frecuencias.

IX. ESTUDIOS DE CONFIABILIDAD

9.1 Estudios de confiabilidad de Hardware

Para cualquier producto que se desea desarrollar es de suma importancia poder caracterizar la confiabilidad del mismo. Para ello existen deferentes parámetros, particularmente es de interés el tiempo medio entre fallas (MTBF). Como únicamente se desarrolló un prototipo y no se cuenta con una población de muestra, no se puede estimar este tiempo con un método estadístico. En este caso, la única opción es predecir el tiempo medio entre fallas a partir de las tasas de fallas de sus componentes. Estos valores se predicen de expresiones cuyos factores dependen de las condiciones de operación y se obtienen, por ejemplo, del manual del departamento de defensa de Estados Unidos, MIL-HDBK-217F. Esto es válido únicamente debido a la suposición que el sistema se encuentra en estado estacionario, es decir que la tasa de fallas es constante.

El prototipo consta de cuatro placas. Dos de ellas son placas de desarrollo del microcontrolador y de la FPGA, cuyas tasas de fallas no se encuentran detalladas por el fabricante. Sin embargo debido a la calidad de sus componentes, sus métodos de fabricación y carácter digital permite suponer que son despreciables frente a las otras dos. Dentro de las placas restantes es esperable que la *KrakenBoard* tenga la tasa de fallas significativa debido a su mayor complejidad y el estrés eléctrico que sufre. Por este motivo se decidió predecir el tiempo medio entre fallas del sistema a partir del de la placa de entrada.

Se debe destacar que, al no haber trabajado con redundancias, todos los componentes se encuentran en serie en el diagrama de confiabilidad. A continuación se presentan los cálculos con las tasas de fallas de los componentes.

9.1.1 Resistencias

La confiabilidad de las resistencias puede ser aproximada utilizando los siguientes coeficientes:

- **λ_b: Tasa de fallas básica.**

- **π_T: Factor de carga por temperatura.**
- **π_P: Factor de carga por factor de potencia.**
- **π_S: Factor de carga por desgaste de potencia.**
- **π_Q: Factor de carga por calidad.**
- **π_E: Factor de carga por ambiente.**

La tasa de fallas λ_p entonces se calcula:

$$\lambda_p = \lambda_b \, \pi_T \, \pi_P \, \pi_S \, \pi_Q \, \pi_E$$

Tipo	Cantidad	λ_b	π_T	π_P	π_S	π_Q	π_E	λ_p
Resistencia	41	0.0037	1.1	0.17	0.79	10	16	3.58570256

9.1.2 Capacitores

La confiabilidad de los capacitores puede ser aproximada utilizando los siguientes coeficientes:

- **λ_b: Tasa de fallas básica.**
- **π_T: Factor de carga por temperatura.**
- **π_C: Factor de carga por capacidad.**
- **π_V: Factor de carga por estrés de voltaje.**
- **π_Q: Factor de carga por calidad.**
- **π_E: Factor de carga por ambiente.**

La tasa de fallas λ_p entonces se calcula:

$$\lambda_p = \lambda_b \, \pi_T \, \pi_C \, \pi_V \, \pi_Q \, \pi_E$$

Capacitor	Cantidad	λ_b	π_T	Π_C	Π_V	π_Q	π_E	λ_p
10pF	4	0.00099	1.3	0.35	1	10	20	0.36036
20nF	3	0.00099	1.3	0.66	1	10	20	0.509652
100nF	20	0.00099	1.3	0.81	1	10	20	4.16988
1uF	4	0.0004	1.1	1	1	10	20	0.352
10uF	17	0.0004	1..1	1.3	1	10	20	1.9448
100uF	3	0.004	1.1	1.6	1	10	20	0.4224

9.1.3 Inductores

La confiabilidad de los inductores puede ser aproximada utilizando los siguientes coeficientes:

- **λ_b: Tasa de fallas básica.**
- **π_T: Factor de carga por temperatura.**
- **π_Q: Factor de carga por calidad.**
- **π_E: Factor de carga por ambiente.**

La tasa de fallas λ_p entonces se calcula:

$$\lambda_p = \lambda_b \, \pi_T \, \pi_Q \, \pi_E$$

Tipo	Cantidad	λ_b	π_T	Π_Q	Π_E	λ_p
Inductor	2	0.00003	1.1	3	12	0.002376

9.1.4 Diodos

La confiabilidad de los diodos puede ser aproximada utilizando los siguientes coeficientes:

- **λ_b: Tasa de fallas básica.**
- **π_T: Factor de carga por temperatura.**
- **π_C: Factor de carga por factor contacto.**
- **π_S: Factor de carga por desgaste de potencia.**
- **π_Q: Factor de carga por calidad.**
- **π_E: Factor de carga por ambiente.**

La tasa de fallas λ_p entonces se calcula:

$$\lambda_p = \lambda_b \, \pi_T \, \pi_C \, \pi_S \, \pi_Q \, \pi_E$$

Tipo	Cantidad	λ_b	π_T	Π_C	Π_S	π_Q	π_E	λ_p
Diodo	2	0.003	1.2	0.42	1	5.5	9	0.149688

9.1.5 Circuitos Integrados

La confiabilidad de los circuitos integrados puede ser aproximada utilizando los siguientes coeficientes:

- **C_1: Tasa de fallas básica.**
- **π_T: Factor de carga por temperatura.**
- **C_1: Factor de carga por factor contacto.**
- **π_Q: Factor de carga por calidad.**

- **π_E: Factor de carga por ambiente.**
- **π_L: Factor de carga por aprendizaje.**

La tasa de fallas λ_p entonces se calcula:

$$\lambda_p = (C_1\,\pi_T + C_2\,\pi_E)\,\pi_Q\,\pi_L$$

Integrado	Tipo	C_1	π_T	C_2	Π_E	π_Q	Π_L	λ_p
Llave Analógica	2	0.01	0.15	0.0034	4	2	1.5	0.0453
Amp Operacional	2	0.01	0.15	0.0025	4	2	1.5	0.0345
Amp. Instrumentación	2	0.02	0.71	0.0043	4	2	1.5	0.0942
Amp. Programable	2	0.04	2.8	0.013	4	2	1.5	0.492
Amp. Diferencial	2	0.02	0.71	0.0034	4	2	1.5	0.0834
ADC	1	0.04	0.71	0.032	4	2	1.5	0.4692
Fuentes Switching	2	0.01	0.71	0.0034	4	2	1.5	0.0621

9.1.6 Conectores BNC

La confiabilidad de los conectores puede ser aproximada utilizando los siguientes coeficientes:

- **λ_b: Tasa de fallas básica.**
- **π_T: Factor de carga por temperatura.**
- **π_K: Factor de carga por conexión/desconexión.**
- **π_Q: Factor de carga por calidad.**
- **π_E: Factor de carga por ambiente.**

La tasa de fallas λ_p entonces se calcula:

$$\lambda_p = \lambda_b \, \pi_T \, \pi_K \, \pi_Q \, \pi_E$$

Conector	Tipo	λ_b	π_T	Π_K	Π_Q	Π_E	λ_p
BNC	2	0.001	1.1	4	2	8	0.1408

9.1.7 Conectores Pines

La confiabilidad de los conectores puede ser aproximada utilizando los siguientes coeficientes:

- **λ_b: Tasa de fallas básica.**
- **π_A: Factor de carga por activos.**
- **π_Q: Factor de carga por calidad.**
- **π_E: Factor de carga por ambiente.**

La tasa de fallas λ_p entonces se calcula:

$$\lambda_p = \lambda_b \, \pi_A \, \pi_Q \, \pi_E$$

Conector	Tipo	λ_b	Π_A	Π_Q	Π_E	λ_p

Pines	1	0.00064	5.9	1	14	0.052864

9.1.8 Sistema completo

Habiendo calculado el valor de confiabilidad de todos los componentes en serie, se puede resolver que la confiabilidad resultante será:

$$\lambda_{p\ TOTAL} = \sum_{i} \lambda_p^i = 12.9712226$$

MTBF (horas)	**77093.7354**
MTBF (años)	8.8006

A partir de la tabla anterior se puede predecir un tiempo medio entre fallas de aproximadamente 8,8 años. Teniendo en cuenta la velocidad en la que avanza la electrónica hoy en día y lo rápido que se vuelven obsoletos estos productos, contar con un tiempo mayor a 8 años es más que suficiente. Además esto asegura que la cantidad de osciloscopios que fallan dentro del primer año, una posible garantía del producto, es mínima. Cabe destacar que, a pesar de haber despreciado el efecto de las placas de desarrollo y la *HCI Board*, se trata de un análisis del peor caso, ya que se utilizó la norma militar MIL-HDBK-217F. Esto permite pensar que el producto final podría tener un tiempo medio entre fallas aún mayor.

Del estudio de confiabilidad realizado también se desprende que las resistencias y los capacitores son la mayor causa de fallas del sistema, principalmente debido a la gran cantidad. La manera más sencilla de reducir la tasa de fallas es utilizando componentes de mayor calidad. Pero esto aumenta de forma drástica el costo, y por lo tanto el precio del producto. Por lo tanto, y teniendo en cuenta que un tiempo medio entre fallas de 8 años es más que razonable, se decidió utilizar componentes comerciales comunes.

9.2. Confiabilidad de software

Definimos a la confiabilidad como un atributo que mide el grado en que un producto opera sin fallas bajo condiciones establecidas por un periodo de tiempo determinado. La confiabilidad de software como atributo cuantitativo debe ser evaluada en base a la cantidad de errores que arroja el software del sistema completo a lo largo una determinada cantidad de tiempo. Este análisis no es posible en el esquema de desarrollo ágil que fue llevado a cabo, dado que las fallas encontradas en el software fueron atacadas a medida que iban surgiendo. Al mismo tiempo, la depuración del software de forma modularizada permitió que el conjunto de componentes para conformar el sistema completo no tuviera una falla de errores considerable.

Por un lado, la tasa de fallas de hardware tiene un valor esperado que aumenta en el tiempo debido al inevitable malgasto de los componentes.

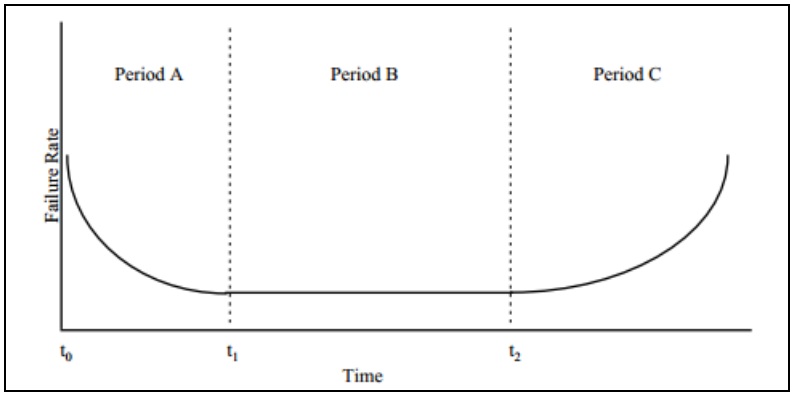


Fig. 9.1: Esquema de confiabilidad de Hardware[16]

Sin embargo, el gráfico a continuación demuestra que la tasa de errores de *software* tiene a mantenerse constante una vez que se han realizado todos los cambios requeridos.

[16] "ELECTRONIC RELIABILITY DESIGN HANDBOOK" Military Handbook; MIL-HDBK-338B; 1 October 1998.

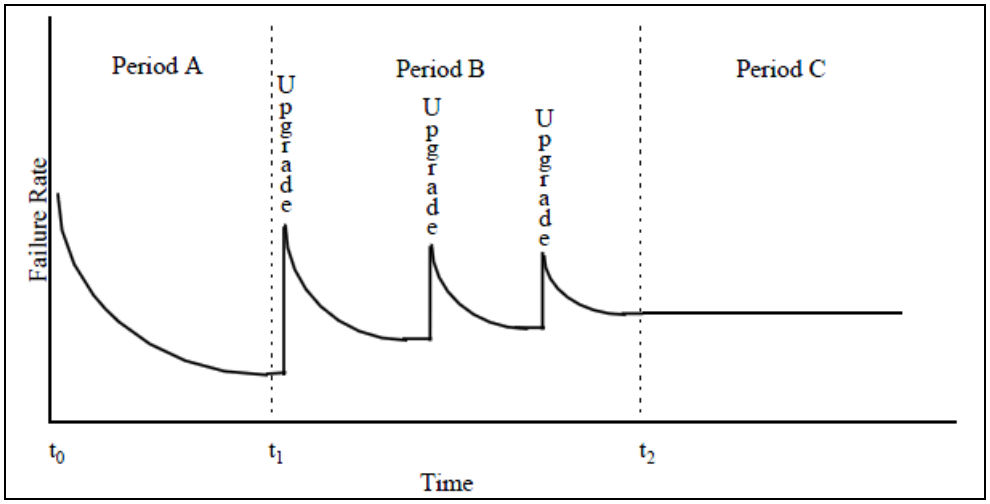


Fig. 9.2: Esquema de confiabilidad de Software

Es válido asumir que la metodología ágil de desarrollo que se llevó a cabo nos traslada al periodo C en el que el valor esperado de la tasa de errores debiera ser constante.

Por último, la confiabilidad de *software* como un atributo cualitativo no hace una gran inferencia en la calidad del producto dado que, de encontrarse alguna falla de este tipo, es coherente asumir que lo único que debe tomar como medida el usuario para corregir el funcionamiento es apagar y prender el dispositivo. Por lo tanto, el análisis de confiabilidad de *software* no genera el suficiente valor agregado frente al costo que implica realizar el mismo.

X. CONCLUSIONES

Llegada esta instancia del desarrollo, corresponde evaluar los objetivos planteados al comienzo de este informe y hasta qué punto han sido alcanzados. En primer lugar, se puede afirmar que el objetivo central ha sido cumplido: se cuenta con un osciloscopio digital de pantalla táctil que se ajusta a las especificaciones planteadas. Tanto el ancho de banda como la frecuencia de muestreo y el rango de entrada, dos de los factores más importantes, están dentro de los rangos requeridos.

No obstante, el desarrollo ciertamente tiene debilidades. Al analizar la evolución del proyecto, es notorio el contraste entre el tiempo estimado al comienzo para su transcurso, y el tiempo real que requirió. Esto se debió principalmente a dos factores: por un lado, la estimación de horas/hombre disponibles por día, sujeta a las vicisitudes de la vida diaria y la inserción en el mundo laboral y, por otro lado, la incertidumbre ante la necesidad de enfrentar desafíos técnicos como los mencionados, que requerían aprender sobre la marcha y, por lo tanto, encontrar dificultades inesperadas. No se pudo, por ejemplo, estimar la curva de aprendizaje necesaria para modificar el kernel de Linux, ni la complejidad resultante de las altas frecuencias a la hora de integrar el ADC con la FPGA. Por otro lado, los integrantes del grupo comenzaron a trabajar durante el transcurso del proyecto, lo cual disminuyó la cantidad de horas/hombre disponibles por día.

Por otro lado, se puede notar algunos faltantes en el prototipo que quedarían como pendientes a agregar en una última etapa, correspondiente al paso intermedio entre el prototipo y la versión final de producción. Principalmente, queda pendiente la integración de la LobsterBoard, por más que es un proceso que ya está delineado. Esto implica, también, que el prototipo no cuenta con la posibilidad de grabar mediciones en una tarjeta SD, ya que el Olinuxino la utiliza para albergar el sistema operativo. A su vez, la utilización de la placa de Olimex imposibilitó la presentación del prototipo alimentado con batería. Esto se debe en primera instancia a que el diseño de la misma debe ser intervenido para trabajar con batería de 3.7V, a pesar de que el microprocesador iMX233 posee el soporte necesario para hacerlo. Por otro lado, la presencia irremovible de ciertos componentes anexos al microprocesador, tal y como el módulo de comunicación LAN, aumentan el consumo del Olinuxino en gran medida, imposibilitando que los reguladores de esta última den abasto para suplir a los módulos propios del

diseño del osciloscopio, tal y como la pantalla o la FPGA. Cabe mencionar nuevamente que el diseño de la LobsterBoard no sólo contempla la alimentación a través de una batería ión litio, sino que además posee un consumo significativamente menor al Olinuxino dado que no posee componentes que no sean de utilidad para el funcionamiento del osciloscopio.

Además, no se ha cumplido con las dimensiones especificadas para el prototipo, ya que se usa una placa de desarrollo para la FPGA con cierto espacio sobrante, y que el cableado requiere otro espacio adicional. Esto se resolvería integrando la etapa de entrada y ADC con la FPGA en una sola placa, lo cual a su vez mejoraría la confiabilidad de la comunicación entre estos subsistemas. Faltaría, a su vez, la creación de un gabinete con un diseño más sofisticado, más apto para el mercado, pero eso escapa al alcance de este proyecto, pues para un resultado aceptable (dada la tendencia del mercado a poner foco en este aspecto) sería indispensable colaborar con un diseñador industrial o, incluso, un equipo de diseño.

Por último, algunas especificaciones quedaron incumplidas por imposibilidades tecnológicas, que el equipo no pudo vislumbrar al comienzo del desarrollo. Por un lado, la carga por USB resultó imposible dado el requerimiento de potencia combinado de los amplificadores de entrada y el sistema de *display*. Esto se resuelve con un cargador, sin afectar considerablemente la percepción de valor del producto, ya que la mayoría de los dispositivos portátiles cuentan con este tipo de accesorios. Por otro lado, la escala mínima horizontal esperada no se pudo alcanzar, ya que la combinación entre la resolución de pantalla y la frecuencia de muestreo máxima generaría la necesidad de interpolar muestras, lo cual carece de sentido ante la aplicación del producto, que es mostrar las señales de la manera más fiel posible. Dada la frecuencia de muestreo, que es el requerimiento más prioritario, la escala horizontal alcanzada es la mínima posible manteniendo este requerimiento.

Más allá de los objetivos puntuales, el presente informe muestra el proceso de ingeniería de un producto electrónico, desde la idea original hasta la validación de un prototipo en funcionamiento.

Es central, en este proceso, el enfoque ingenieril: el desarrollo se lleva a cabo de manera sistemática, con una metodología específica y pasos razonados y definidos. Cada etapa del proceso contribuye a asegurar el éxito y garantizar la calidad y confiabilidad del producto.

Es pertinente notar el uso de una metodología *agile*, en particular para el software, que permite acelerar el proceso de desarrollo y adaptarse a los cambios que se producen, inevitablemente, al evolucionar el proyecto.

He aquí, entonces, un proyecto final de Ingeniería, en el que se han aplicado conocimientos adquiridos a lo largo de toda la carrera.

Por un lado, se realizó un desarrollo de hardware analógico y digital. Del lado analógico, se observa la implementación de filtros, amplificadores y otros acondicionadores de señal. Del lado digital, la aplicación de conocimientos se afirma en la creación de un circuito de procesamiento en lógica programable (dentro de la FPGA) y en un circuito impreso (en el caso de la *LobsterBoard*). Asimismo, se han aplicado los conocimientos de procesamiento de señales, también en los dominios analógico y digital. Desde estos conocimientos se pudo implementar el hardware y software que aseguraran el correcto tratamiento de las señales medidas.

Por otro lado, el desarrollo de software implicó un uso intenso de los conceptos aprendidos sobre programación, para lograr una estructuración adecuada del código y un sistema mantenible y confiable. El paradigma de la programación orientada a objetos, por ejemplo, fue fundamental para el desarrollo de la aplicación de Linux sobre la que funciona el dispositivo.

Más allá de los conocimientos particulares, de las técnicas y conceptos específicos sobre los que se construye este proyecto, es importante notar la *actitud* que se obtiene de la carrera, que permitió enfrentar este proyecto cuando, en el origen, varios de sus componentes eran incógnitos. Esta actitud es la de "aprender haciendo", y la de enfrentar desafíos con un enfoque hermenéutico, buscando y encontrando soluciones a medida que aparecen los problemas.

Esto es destacado, en buena medida, porque así como se aplicó una variedad de conocimientos de la carrera, muchas de las técnicas aplicadas en este desarrollo escapan a (o dan un paso más allá de) lo aprendido en las materias de electrónica.

En primer lugar, el desarrollo sobre Linux embebido era desconocido para los integrantes del grupo, y el *kernel* y su API conforman un gran universo de código, que requirió un proceso de aprendizaje lento. En este proceso, se debió aplicar la actitud mencionada para identificar los componentes del sistema, comprenderlos y luego poder modificarlos.

De la misma manera, los trabajos prácticos de la carrera no habían requerido desarrollar circuitos de la complejidad del osciloscopio. Del lado analógico, se trabajó con una serie de etapas de amplificación en relativamente altas frecuencias, por lo que la selección de componentes fue crucial y requirió sumergirse en las hojas de datos y documentación para encontrar la mejor manera de acondicionar las señales. Del lado digital, se desarrolló una computadora considerablemente más compleja que las placas de desarrollo con las que se había trabajado en las materias relacionadas, que involucraba comunicaciones a altas frecuencias y que fue el primer acercamiento de estos desarrolladores al diseño de circuitos multicapa. En la FPGA, se debió implementar un sistema de procesamiento, almacenamiento y comunicación confiable y efectivo a altas velocidades.

Finalmente, la construcción del prototipo requirió tener en cuenta consideraciones prácticas y mecánicas con las que ningún integrante del grupo estaba familiarizado. Es posible que este sea uno de los aspectos en los que más se nota la falta de experiencia.

En síntesis, este desarrollo consistió en la producción de un sistema sumamente complejo, con una variedad de subsistemas altamente disímiles; llevarlo a cabo fue un desafío lleno de aprendizaje.

Al observar el proceso de desarrollo en su conjunto, con el trabajo que implica, y el sistema complejo y funcional resultante, se puede sostener que el proyecto ha sido satisfactorio. Las dificultades, incógnitas, los errores, el aprendizaje y los logros, vistos a la luz de un producto que cubre una necesidad, generan una realización que es muy característica de esta disciplina, en la que el conocimiento es aplicado en pos de la mejora constante y la efectividad creativa. Se trata, en definitiva, de la sensación de haber llegado, mediante el aprendizaje y la creación, a un nuevo lugar desde donde continuar aprendiendo y creando.

XI. ANEXO

A. Código de software

Debido a la longitud del código generado en este proyecto se decidió proveerlo en formato digital junto a la entrega en CD. De todas formas se puede consultar el mismo a través sus repositorios online en *github*: https://github.com/spairal/softscope (código de la aplicación); https://github.com/spairal/linux-for-lobster (*kernel* de Linux).

B. Adaptación de impedancias en comunicación ADC-FPGA

Al conectar la FPGA con el ADC, se comprobó que la comunicación provocaba errores de bits debido a una desadaptación de impedancias, ya que se trata de señales cuadradas de 50MHz pasando por cables de 30 cm de largo.

Dado que los cables no son tan largos como para necesitar modelarlos como líneas de transmisión (ya que la longitud de onda es mucho mayor), se puede modelar como una resistencia en serie con una inductancia, que termina en la capacitancia parásita de la placa de FPGA.

Si el cable es de cobre, de 30cm de largo y de aproximadamente 1 mm de diámetro, su inductancia ronda los 380nH[17] y su resistencia de unos 0.16 ohms[18].

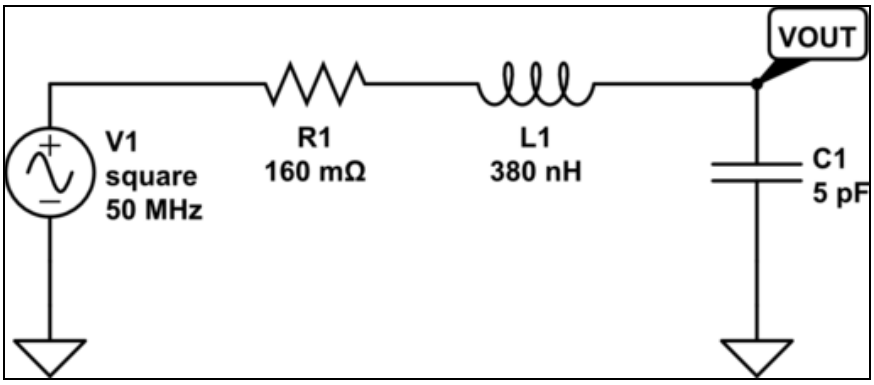
La capacitancia parásita se suele estimar alrededor de 5pF. Eso nos lleva al siguiente circuito (simulado en CircuitLab[19]):

---

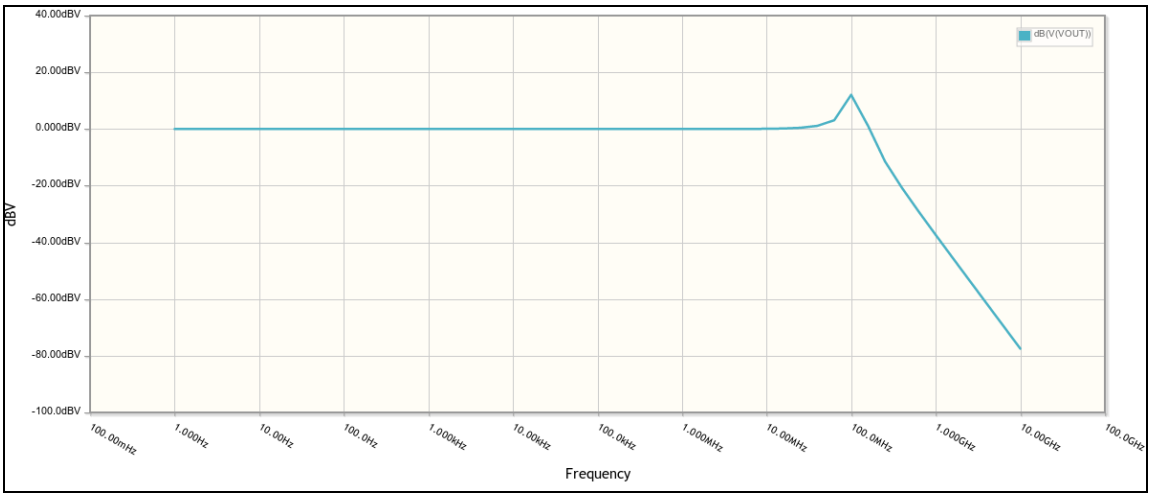
[17] http://www.consultrsr.com/resources/eis/induct5.htm

[18] http://hyperphysics.phy-astr.gsu.edu/hbase/electric/resis.html

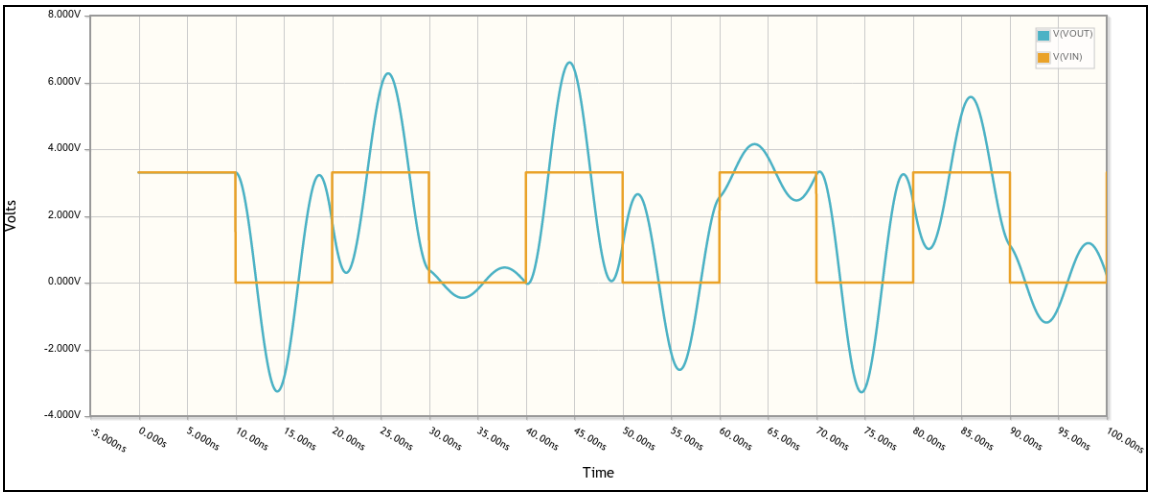
[19] http://www.circuitlab.com



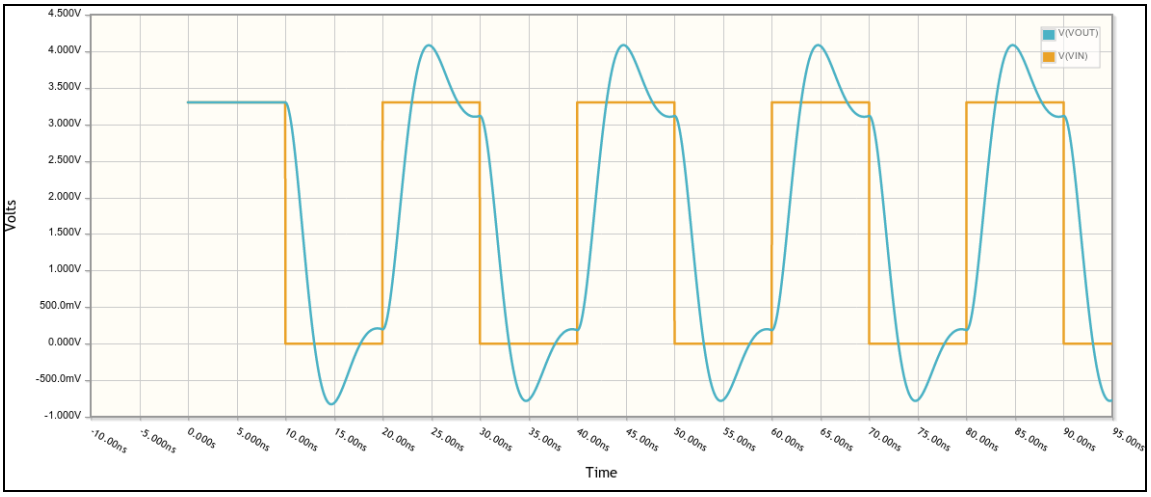
La respuesta en frecuencia de este circuito tiene un sobrepico, como se observa en el siguiente gráfico simulado:



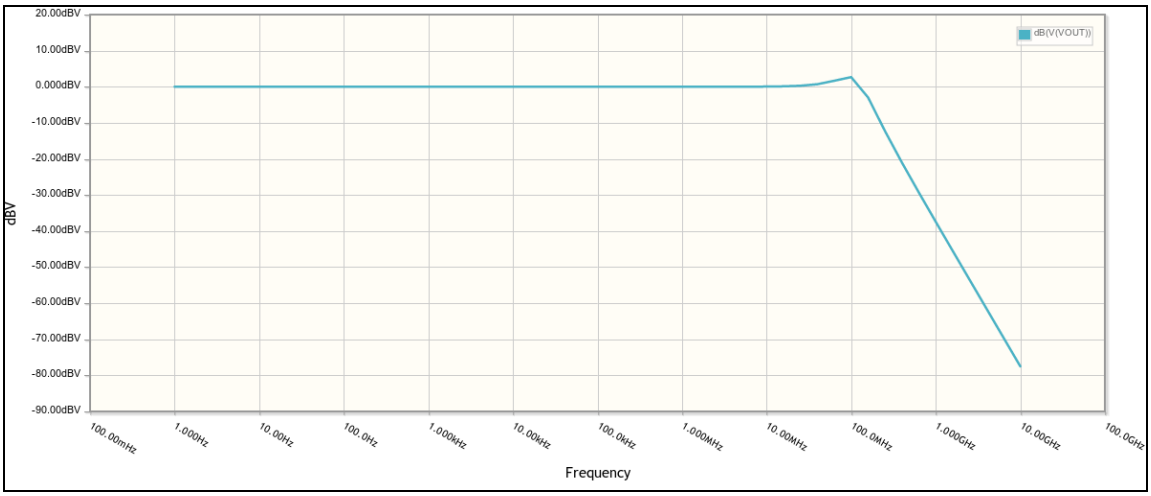
Esto corresponde a una respuesta sub amortiguada, que puede producir *ringing*:



Esto inevitablemente genera errores de bit. Si agregamos una resistencia en serie de 220 ohms, la respuesta mejora considerablemente, ya que los polos tienen menor Q:



Este valor de resistencia se eligió de manera empírica sobre las simulaciones. Un valor demasiado alto mueve los polos hacia la izquierda, filtrando la señal como pasabajos. Por eso se eligió un valor relativamente bajo, dejando margen para errores en la estimación de los componentes parásitos. El diagrama de Bode resultante es el siguiente:



C. Cálculo de los elementos de la placa de entrada

Cálculo del capacitor de acoplamiento

Se estudiará el caso en el que se utiliza un acoplamiento de alterna, es decir que el capacitor de acoplamiento se encuentra en el camino de la señal. La transferencia completa de la etapa de entrada y el acoplamiento en este caso es la siguiente:

$$H(s) = \frac{R_2^2 \cdot C \cdot s}{(2 \cdot R_1 + R_2) \cdot R_2 \cdot C \cdot s + R_1 + R_2}$$

Donde R_1 vale $750k\Omega$, R_2 vale $500k\Omega$ y C es la capacitancia que se pretende calcular. De ahí se desprende que la frecuencia de corte obedece a la siguiente expresión:

$$f = \frac{R_1 + R_2}{2 \cdot \pi \cdot (2 \cdot R_1 + R_2) \cdot R_2 \cdot C}$$

Si esta frecuencia se desea que sea 10Hz, la expresión para calcular el capacitor es la siguiente:

$$C = \frac{R_1 + R_2}{2 \cdot \pi \cdot (2 \cdot R_1 + R_2) \cdot R_2 \cdot f} = 33\text{nF}$$

Cálculo del filtro del generador de offset

Asumiendo que las resistencias y capacitancias de ambas etapas son equivalentes, la transferencia del filtro del generador de offset es la siguiente:

$$H(s) = \frac{1}{R^2 \cdot C^2 \cdot s^2 + 3 \cdot R \cdot C \cdot s + 1}$$

Donde R es la resistencia y C la capacitancia de cada etapa del filtro. A partir de la expresión anterior se desprende la siguiente fórmula para la frecuencia de corte media.

$$f = \frac{1}{2 \cdot \pi \cdot R \cdot C}$$

Adicionalmente se impone una relación sobre los valores de R y C para que ninguno sea demasiado alto. Si se busca una frecuencia de corte de aproximadamente 5Hz, se puede deducir de las expresiones anteriores valores que se deben utilizar una resistencia de aproximadamente $220k\Omega$ y una capacidad de aproximadamente 220nF.

D. Listado de partes

A continuación se listan los principales componentes de cada una de las placas de interés en el sistema.

Listado de partes de la *KrakenBoard:*

Nombre	Descripción	Cantidad
1N5817	Diodo	2
AD8253	Amplificador de instrumentación	2
ADS5237	Conversor A/D	1
MAX734	Fuente *switching* (12 v)	1
MAX765	Fuente *switching* (-12 v)	1
MAX4659	Llave analógica	2
OPA170	Amplificador operacional	2
THS4130	Amplificador diferencial	2
THS7002	Amplificador programable	2

Listado de partes importantes de la HCI Board:

Nombre	Descripción	Cantidad
TPS61092	Fuente switching tipo Boost – salida 5V	1
IRF7204	MOSFET canal P	1
10BQ015	Diodo	1

FH12-40S-0.5H	Conector de cable FPC para LCD	1
84952-4	Conector de cable FPC para touch	1
TSC2007	Controlador de touch resistivo I2C	1

Listado de partes importantes de la LobsterBoard:

Nombre	Descripción	Cantidad
MCIMX233CAG4C	Procesador Freescale i.MX233	1
HY5DU121622DTP-J	Memoria DDR RAM 512Mbits	1
MX25L25635e	Memoria SPI Flash 256Mbits	1
ABM8G	Cristal 24MHz 10pF	1
NZL6V8	Diodo Zener	1
BFM505	Transistor doble NPN	1
10BQ015	Diodo	1

E. Conexiones entre placas

Conexiones entre EVM Cyclone y *KrakenBoard*:

EVM Cyclone		KrakenBoard		
#	Header	#	Header	Name
	J5 – B4 – 11	1	Canal A	Offset
	J5 – B4 – 12	2	Canal A	Coupling
	J5 – B4 – 13	3	Canal A	G0A
	J5 – B4 – 17	4	Canal A	G1A
	J5 – B4 – 18	5	Canal A	G2A
	J5 – B4 – 19	6	Canal A	G2B
	J5 – B4 – 20	7	Canal A	G1B
	J5 – B4 – 23	8	Canal A	G0B
	J2 – B2 – 4	1	Canal B	Offset
	J2 – B2 – 8	2	Canal B	Coupling
	J2 – B2 – 12	3	Canal B	G0A
	J2 – B2 – 16	4	Canal B	G1A
	J2 – B2 – 18	5	Canal B	G2A
	J2 – B2 – 20	6	Canal B	G2B
	J2 – B2 – 26	7	Canal B	G1B
	J2 – B2 – 28	8	Canal B	G0B
	J5 – B4 – 1	Pin suelto	ADC	SEL
	J5 – B4 – 2	1	ADCSERIAL	SCK
	J5 – B4 – 3	2	ADCSERIAL	SEN
	J5 – B4 – 4	3	ADCSERIAL	SDATA
	J5 – B4 – 5	1	ADC CLK	CLK
	J2 – B2 - 29	1	ADC A	DV
	J2 – B2 – 27	2	ADC A	D0

J2 – B2 – 25	3	ADC A	D1
J2 – B2 - 23	4	ADC A	D2
J2 – B2 – 17	5	ADC A	D3
J2 – B2 – 13	6	ADC A	D4
J2 – B2 – 11	7	ADC A	D5
J2 – B2 - 9	8	ADC A	D6
J2 – B2 – 5	9	ADC A	D7
J2 – B2 – 3	10	ADC A	D8
J2 – B2 - 1	11	ADC A	D9
J3 – B3 - 11	12	ADC A	OR
J3 – B3 – 13	1	ADC B	DV
J3 – B3 – 19	2	ADC B	D0
J3 – B3 – 21	3	ADC B	D1
J3 – B3 – 23	4	ADC B	D2
J3 – B3 – 25	5	ADC B	D3
J3 – B3 – 27	6	ADC B	D4
J3 – B3 - 29	7	ADC B	D5
J3 – B3 – 12	8	ADC B	D6
J3 – B3 – 20	9	ADC B	D7
J3 – B3 – 24	10	ADC B	D8
J3 – B3 – 26	11	ADC B	D9
J3 – B3 - 28	12	ADC B	OR

Conexiones entre *HCI Board* y olinuxino:

HCI Board		olinuxino		
#	Nombre	#	Header	Nombre
1	LCD_D17 (TSC_IRQ)	3	GPIO	LCD_D16

3	VDDIO1 (TSC_PWR)	1	UEXT	3.3V
4	LCD_DOTCK	6	GPIO	DOTCLK
5	LCD_VSYNC	8	GPIO	VSYNC
6	LCD_HSYNC	10	GPIO	HSYNC
12	LCD_D07	21	GPIO	LCD_D07
13	LCD_D06	23	GPIO	LCD_D06
14	LCD_D05	25	GPIO	LCD_D05
15	LCD_D04	27	GPIO	LCD_D04
16	LCD_D03	29	GPIO	LCD_D03
17	LCD_D02	31	GPIO	LCD_D02
18	LCD_D01	33	GPIO	LCD_D01
19	LCD_D00	35	GPIO	LCD_D00
21	I2C_SDA	4	UEXT	AUART1_RXD
22	I2C_SCL	3	UEXT	AUART1_TXD
25	GND	40	GPIO	GND
26	VDDIO2 (LCD_PWR)	38	GPIO	3.3V
1_PWM	PWM	22	GPIO	PWM2

Conexiones entre olinuxino y *EVM Cyclone*:

olinuxino			EVM Cyclone		
#	Header	Nombre	#	Header	Nombre
7	UEXT	SSP2_MISO	27	J5	MISO
8	UEXT	SSP2_MOSI	28	J5	MOSI
9	UEXT	SSP2_SCK	25	J5	Serial clock
10	UEXT	UEXT_CS	29	J5	Slave select

XII. BIBLIOGRAFÍA

- **Proakis, Manolakis. "Digital signal processing: Principles, algorithms and applications". Editorial Pearson. 2006.**
- **Sedra, Smith. "Microelectronic circuits". Editorial Oxford. 2009**
- **Mohan. "Power electronics". Editorial Wiley. 2002.**
- **Franco. "Design with operational amplifiers". Editorial McGraw Hill. 2001.**
- **Floyd. "Digital fundamentals". Editorial Prentice Hall. 2008.**
- **Bell. "Electronic instrumentation and measurements". Editorial Oxford. 2007.**
- **Tektronix. "XYZs of oscilloscopes". http://ecee.colorado.edu/~mcclurel/txyzscopes.pdf**
- **Hallinan. "Embedded Linux premier". Editorial Prentice Hall. 2006.**
- **Free Electrons. "Cursos sobre Linux embebido". www.free-electrons.com**
- **Hojas de datos de componentes electrónicos provistas por los fabricantes.**