



**UNIVERSIDAD POLITÉCNICA DE MADRID**

**Escuela Universitaria de Ingeniería Técnica de  
Telecomunicación**

Proyecto Fin de Carrera

**Implementación de un sistema  
multicanal de medida de  
temperatura con termopares**

Autora: Silvia Callejas González

Tutor: Mariano Ruiz González

Madrid, 2013



If you need something  
to walk forward,  
just put your headphones in  
and press play.



# Índice

---

Índice.....	v
Índice de figuras.....	ix
Índice de tablas.....	xi
Agradecimientos.....	xiii
Resumen.....	xv
Abstract.....	xvii
1 Introducción.....	1
2 Requisitos del sistema.....	3
2.1 Introducción.....	3
2.2 Especificaciones del diseño.....	4
2.3 Objetivos de aprendizaje.....	5
2.1 Entorno de trabajo.....	6
3 Descripción del sensor de temperatura.....	9
3.1 Introducción.....	9
3.2 Termoelectricidad.....	10
3.3 Termopares. El efecto Seebeck.....	10
3.4 Tipos de Termopares.....	13
3.5 Sensor Utilizado: Tipo J.....	16
4 Descripción del convertor.....	19
4.1 Introducción.....	19
4.2 Convertor de Termopar a Digital con Compensación de la Unión Fría.....	20
4.2.1 Descripción Hardware.....	20
4.2.2 Descripción Software.....	23
5 Sistema de control y procesado: FPGA.....	27
5.1 Introducción.....	27
5.2 FPGA.....	28
5.3 Kit de desarrollo.....	29
5.4 Diagrama de bloques.....	30
6 Arquitectura del sistema completo.....	33
6.1 Introducción.....	33
6.2 Visión general del sistema completo.....	34

7	Bloque I: Conversión.....	37
7.1	Introducción.....	37
7.2	Diagrama de bloques.....	38
7.3	Esquemático.....	39
7.3.1	Alimentación.....	39
7.3.2	Conectores de termopar y protecciones.....	39
7.3.3	Circuito de conexión del MAX31855.....	40
7.4	Diseño.....	41
7.4.1	Diseño PCB.....	41
7.4.1	Fabricación del PCB.....	42
7.4.2	Montaje de los componentes.....	43
7.4.3	Costes.....	45
8	Bloque II: Procesado digital.....	47
8.1	Introducción.....	47
8.2	Periféricos utilizados.....	48
8.3	Configuración de la FPGA: XPS.....	48
8.3.1	Bloques VHDL y drivers implementados en la plataforma XPS.....	49
8.3.1	Restricciones del sistema y pines de entrada salida.....	53
8.4	Programación del procesador: SDK.....	55
9	Bloque III: Procesado de datos en el PC.....	59
9.1	Introducción.....	59
9.2	Captura y procesado de la trama mediante LabVIEW.....	60
9.3	Interfaz de usuario LabVIEW.....	65
9.3.1	Interfaz de usuario para visualización de medidas.....	65
9.3.2	Interfaz de usuario para visualización y captura de medidas.....	66
10	Proceso de calibración.....	67
10.1	Introducción.....	67
10.2	Calibrador del sistema.....	68
10.3	Cálculos de ajuste y calibración mediante Excel.....	69
	Primer método de calibración.....	70
	Segundo método de calibración.....	73
10.4	Verificación de los cálculos de calibración.....	75
11	Conclusiones y posibles mejoras.....	77
11.1	Conclusiones.....	77

11.2	Posibles mejoras .....	78
12	Apéndices.....	79
12.1	Esquemático completo del Bloque I .....	81
12.2	Fotolitos para la fabricación de la placa acondicionadora del Bloque I .....	82
12.3	Lista de componentes necesarios. MPL (Master Part List).....	83
12.4	IPCores de XPS .....	85
12.5	Diagrama de bloques de XPS .....	86
12.6	Restricciones del usuario UCF (User Constraints File).....	87
12.7	Código de programa en C del PowerPC440.....	88
12.8	Código del archivo MHS.....	96
12.9	Código del archivo MSS.....	99
12.10	Código del archivo *.UCF .....	101
12.11	Código LabVIEW .....	102
	Panel frontal básico de usuario .....	104
	Panel frontal de calibración .....	104
12.12	Cálculos y gráficas en Excel de los cuatro canales.....	105
	Canal 1 .....	105
	Canal 2 .....	106
	Canal3 .....	107
	Canal4 .....	108
	Bibliografía.....	109





# Índice de figuras

---

Figura 1. Diagrama del desarrollo. ....	3
Figura 2. Diagrama de bloques del sistema completo. ....	5
Figura 3. Termopar tipo J. ....	9
Figura 4. Efecto Seebeck: Al aplicar calor a una unión de las uniones de dos materiales diferentes, se crea una diferencia de tensión proporcional a la temperatura aplicada.....	10
Figura 5. Unión fría creada al conectar un instrumento de media en los bornes del termopar, con ello se crean tensiones parásitas que deben ser compensadas.....	12
Figura 6. Modelo de compensación de la unión fría mediante región isotérmica. Se crean dos nuevas uniones de las mismas características (J2 y J4) de este modo se anulan las tensiones parásitas (VJ2 y VJ4). La compensación de la tensión parásita VJ3 se debe hacer mediante software o hardware. ....	12
Figura 7. Tensión de termopar frente a temperatura según los diferentes tipos de termopares.....	16
Figura 8. Encapsulado MAX31855.....	19
Figura 9. Diagrama de bloques. MAX31855, incluye un amplificador operacional, detección del estado de la conexión del termopar y un bloque especial para la compensación de la unión fría así como un ADC para hacer la conversión de termopar a digital.....	20
Figura 10. Circuito de aplicación típico del MAX31855, a la entrada se conecta el termopar, la comunicación SPI se gestionará en este caso con el procesador de la FPGA.....	21
Figura 11. Pinout del MAX31855. Encapsulado tipo SOP de 8 pines. ....	23
Figura 12. Diagrama del protocolo de la comunicación SPI. ....	23
Figura 13. Diagrama de tiempos de la comunicación SPI. ....	24
Figura 14. Vista simplificada de una FPGA. ....	27
Figura 15. Esquema de la arquitectura interna de una FPGA. ....	28
Figura 16. Placa de desarrollo Virtex-5 ML507. ....	29
Figura 17. Diagrama de bloques de la plataforma de evaluación FPGA Virtex-5 ML507. ....	31
Figura 18. Diagrama de bloques global .....	33
Figura 19. Sistema completo. ....	34
Figura 20. Placa acondicionadora de señal (Bottom) .....	37
Figura 21. Bloque de conversión. ....	38
Figura 22. Conectores de alimentación, jumper de selección y condensadores de desacoplo para los circuitos integrados. ....	39
Figura 23. Conector de termopar y circuito de protección.....	40
Figura 24. Circuito de conexión del MAX31855. ....	40
Figura 25. Diseño PCB en OrCAD Layout.....	41

Figura 26. Insoladora donde se colocará la placa con el fotolito. ....	42
Figura 27. Equipo de revelado y atacado con calentadores para los líquidos y generador de burbujas que aceleran el proceso de fabricación. ....	43
Figura 28. Placa acondicionadora finalizada. Detalle soldadura. ....	44
Figura 29. Placa acondicionadora finalizada. ....	44
Figura 30. FPGA Virtex 5. ....	47
Figura 31. Diagrama de bloques implementados en la plataforma de desarrollo Virtex-5 .....	48
Figura 32. Bus de interfaces implementadas en la plataforma de desarrollo XPS. ....	50
Figura 33. Diagrama de bloques de los IP Cores implementados mediante la plataforma de desarrollo XPS. ....	51
Figura 34. Diagrama de flujo del programa principal del procesador PowerPC440 ...	56
Figura 35. Trama de temperaturas para comunicación RS-232 .....	57
Figura 36. Logo del programa LabVIEW de National Instruments. ....	59
Figura 37. Código LabVIEW para lectura escritura de puerto serie. ....	61
Figura 38. Inicio del procesado de la trama leída. ....	61
Figura 39. Procesado de la trama, código completo. ....	62
Figura 40. Procesado de la trama, comprobación del estado de la conexión. Byte de control. ....	63
Figura 41. Procesado de la trama. Procesado de temperaturas con valor decimal. ....	63
Figura 42. Procesado de trama de medida. Índice de canal medido. ....	64
Figura 43. Procesado de trama de medidas. Guardado de datos en array. ....	64
Figura 44. Interfaz gráfica de usuario en LabVIEW. ....	65
Figura 45. Interfaz gráfica de usuario en LabVIEW. ....	66
Figura 46. Calibrador FLUKE 725. ....	67
Figura 47. Sistema completo con el calibrador utilizado. ....	68
Figura 48. Representación gráfica de los errores de ganancia y offset. ....	69
Figura 49. Error de medida tras la calibración por el primer método. ....	72
Figura 50. Comparación del valor medido respecto al ideal con el primer método de calibración. ....	72
Figura 51. Error de medida tras la calibración por el segundo método. ....	74
Figura 52. Comparación de la medida real frente a la ideal con el segundo método de calibración. ....	74

# Índice de tablas

---

Tabla 1. Comparativa de los tipos de de termopares más comunes según material y rango de temperatura. ....	15
Tabla 2. Coeficientes para la calibración de termopar tipo J en función del rango de temperatura. ....	17
Tabla 3. Condiciones de trabajo del MAX31855 recomendados por el fabricante.....	21
Tabla 4. Características térmicas y tiempos de conversión del MAX31855 en su versión para termopar tipo J. ....	22
Tabla 5. Características de los tiempos en la comunicación SPI.....	24
Tabla 6. Descripción del Mapa de Memoria. Indica la posición de los bits y la información de la trama.....	25
Tabla 7. Formato de la temperatura en la trama de comunicación serie. Izquierda: temperatura del termopar, derecha: temperatura de la unión fría.....	25
Tabla 8. Costes de fabricación del PCB. ....	45
Tabla 9. Resumen de recursos utilizados en la FPGA.....	52
Tabla 10. Localización de los LEDs en la FPGA.....	53
Tabla 11. Localización del conector de expansión J6 en la FPGA. ....	54
Tabla 12. Cálculos del primer método de calibración realizados con Excel. ....	71
Tabla 13. Cálculos del segundo método de calibración realizados con Excel. ....	73



# Agradecimientos

---

Qué mejor broche para cerrar este ciclo que agradecer a todos los que han estado conmigo en los últimos años de estudio, tanto en los libre acceso y en la biblioteca estudiando y terminando prácticas hasta las tantas, como en los ratos libres de descanso merecido para desconectar.

Gracias también en especial a Mariano, mi tutor, por ayudarme y orientarme en todo el desarrollo, y por preocuparse siempre de que sus alumnos estemos bien y hagamos nuestro trabajo aprendiendo el máximo posible.

Por supuesto, y rozando la originalidad, quiero agradecer a mis padres todo lo que han apostado por mí, y los ánimos y los “acuéstate que ya es tarde, mañana sigues” y un largo etcétera. Gracias por estar ahí, por toda vuestra ayuda y aguantar el trabajo que os damos.

Mi hermana se merece un apartado especial en exclusiva en esta página, Rosi, tú siempre me lees y sabes antes que nadie lo que me pasa, gracias por tus consejos y por todos esos pasos que me has ahorrado cuando tú te los has tenido que trabajar.

Muchas gracias PANDA por estar ahí siempre, pase lo que pase y sea cuando sea. Nos fundamos hace casi diez años y ya somos diez, seguiremos sumando y creciendo, viajando y riendo...y jugando, por supuesto.

Gracias también a toda la gente de AETEL, vosotros sois los que más me habéis visto estos meses siempre acompañada del portátil y sin otro tema de conversación casi que no fuese sobre el proyecto. Os meto caña pero sabéis que es desde el cariño.

¡¡Mangueros!! Los veranos se hacen más cortos, pero aprovechables al 100%, mínimo una semana al año no se perdona, esperemos que pueda ser más.

Néstor, gracias por hacerme este camino más fácil. Te debo una, o dos, o mil. Rober, tú has estado ahí desde el primer día, y por muy lejos que te vayas no te libras de mí, ya lo sabes. Barras, mucho ánimo que esto acaba. Alberto y Guti, esos libres accesos marcarán época. Extre, ¿me traerás esas cerezas como premio por haber llegado hasta aquí? Tania, gracias por tus minidetallazos que alegran el día y por recordarme que el túnel acaba sin que te des cuenta de que has llegado al final. Jana, mi compi en el extranjero, al final presentamos el mismo año ¿dónde nos vamos para celebrarlo?

Al gañán de turno le quiero dar las gracias por aguantarme y por todo lo que me ha ayudado, aunque diga que no. Richi, gracias por las risas cuando no había manera de encontrarlas.

Quiero agradecer a Juanjo, Javi, Carlos, Luis y Chechu por la gran oportunidad y toda la ayuda que me han dado a lo largo de estos meses.



# Resumen

---

Este proyecto fin de carrera tiene como finalidad el diseño e implementación de un sistema multicanal de medida de temperaturas con termopares con procesado digital.

Se ha realizado un prototipo de cuatro canales con conexión de termopar, que es el tipo de sensor utilizado para realizar dichas medidas. La tensión generada por el termopar es procesada mediante un conversor de termopar a digital con salida en interfaz modo serie o SPI (Serial Peripheral Interface). El control de dicha comunicación se realiza por medio de un Array de Puertas Lógicas Programables o FPGA (Field Programmable Gate Array), en concreto se ha utilizado una plataforma de desarrollo modelo Virtex-5 de la empresa Xilinx. Esta tarjeta se ha programado también para el procesado software y la posterior comunicación serie con el PC, el cual consta de una interfaz de usuario donde se muestran los resultados de las medidas en tiempo real.

El proyecto ha sido desarrollado en colaboración con una empresa privada dedicada principalmente al diseño electrónico. La finalidad de este prototipo es el estudio de una actualización del bloque de medida para el control de las curvas de temperatura de un equipo de reparación aeronáutica.

En esta memoria se describe el proceso realizado para el desarrollo del prototipo, incluye la presentación de los estudios realizados y la información necesaria para llevar a cabo el diseño, la fabricación y la programación de los diferentes bloques que componen el sistema.





# Abstract

---

The aim of this project is to implement a multichannel temperature measurement system with digital processing, using thermocouples.

A four-channel prototype with thermocouple connection has been built. The thermocouple voltage is converted to digital line using a Thermocouple-to-Digital Converter with a Serial Peripheral Interface (SPI) output. The master which controls this communication is embedded in a Field Programmable Gate Array (FPGA), specifically the Xilinx Virtex-5 model. This FPGA also has the code for software temperature processing and the prototype to PC serial communication embedded. The PC user interface displays the measurement results in real time.

This project has been developed at a private electronics design company. The company wants to study an update to change the analogue temperature controller equipment to a digital one. So this prototype studies a digital version of the temperature measurement block.

The processes accomplished for the prototype development are detailed in the next pages of this document. It includes the studies and information needed to develop the design, manufacturing process and programming of the blocks which integrate with the global system.



# 1 Introducción

---

Este documento está dividido en 12 capítulos en los cuales se explica el estudio, desarrollo, diseño e implementación del proyecto realizado. Se detallan también los elementos empleados en cada bloque tanto sensores, conversores, sistemas digitales, programas utilizados, como los estudios realizados y descripciones necesarias para comprender el funcionamiento y manejo del prototipo. A continuación se presenta un breve resumen del contenido de los siguientes capítulos.

Capítulo 2. Requisitos del sistema. Se presentan las especificaciones iniciales, así como las especificaciones propuestas para el diseño del sistema. Se incluyen también los objetivos de aprendizaje, y el entorno de trabajo con las herramientas necesarias para llevar a cabo la fabricación e implementación del prototipo.

Capítulo 3. Descripción del sensor de temperatura. Se presenta una breve introducción a la termoelectricidad en lo referente a termopares. También se incluye una descripción de los tipos de termopares que se encuentran en el mercado, y se define el tipo de termopar utilizado en este proyecto.

Capítulo 4. Descripción del convertidor. En este capítulo se hace un estudio de la hoja de características del convertidor empleado, el MAX 31855 del fabricante Maxim, mostrando desde el diagrama de bloques interno hasta el formato de la trama de salida en su comunicación de interfaz serie (SPI), pasando por los tiempos de comunicación, los valores de tensión de alimentación, frecuencia de trabajo y el pinout del componente.

Capítulo 5. Sistema de control y procesado: FPGA. Aquí se explica brevemente cómo es una FPGA internamente, sus inicios y evolución. También se presenta el kit

de desarrollo Virtex-5, que es el que se emplea en este proyecto, junto con las herramientas de desarrollo necesarias.

Capítulo 6. Arquitectura del sistema completo. Para facilitar la comprensión de la descripción de los bloques que componen este proyecto, en este capítulo se muestra el sistema completo diferenciando los bloques por los que está compuesto e indicando el tipo de comunicación utilizado entre cada uno de ellos.

Capítulo 7. Bloque I: Conversión. Presentación del esquemático de la placa fabricada, incluyendo los componentes empleados y el funcionamiento de la placa. Así como una descripción del diseño mediante la herramienta OrCAD y fabricación de la misma por método manual en laboratorio.

Capítulo 8. Bloque II: Procesado digital. Se describen los periféricos de la tarjeta Virtex-5 que se emplean en este proyecto así como su configuración, la cual se realiza mediante la herramienta de desarrollo XPS (Xilinx Platform Studio), junto con la descripción de cada periférico en lenguaje hardware VHDL. También se incluye en este capítulo la programación del procesador embebido en la FPGA, mediante la herramienta de desarrollo software SDK (Software Development Kit)

Capítulo 9. Software: Procesado de medidas y comunicación con el PC. Aquí se incluye por una parte el software de procesado de medidas que ejecutará el microcontrolador Power PC 440, y por otra la aplicación diseñada en LabVIEW para el control de la comunicación mediante puerto serie y visualización de resultados en el PC.

Capítulo 10. Proceso de calibración. En este capítulo se describe el método utilizado para el ajuste de las medidas, incluyendo las ecuaciones que se deben aplicar a dichas medidas. Del mismo modo, se presentan los resultados obtenidos así como el proceso de verificación de los mismos.

Capítulo 11. Conclusiones y posibles mejoras. Aquí se detallan las conclusiones finales tras el proceso de diseño y desarrollo del proyecto completo. También se detallan posibles mejoras para versiones futuras del sistema.

Capítulo 12. Apéndices. Recoge los documentos necesarios para fabricar la placa de conversión, el código ejecutado por el procesador y los bloques de la aplicación software para el PC, así como las hojas de cálculo de calibración.

## 2 Requisitos del sistema

---

### 2.1 Introducción

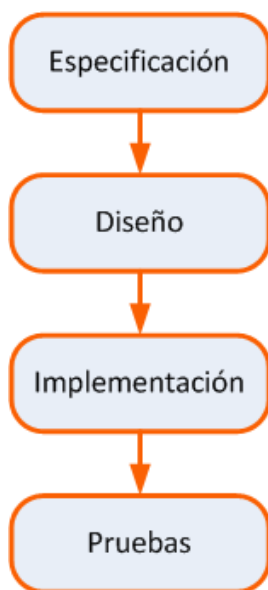


Figura 1. Diagrama del desarrollo.

En este proyecto se desarrolla el prototipo de un sistema de medida de temperaturas de cuatro canales. El diseño está orientado a formar parte del control de temperatura de un equipo industrial de reparación aeronáutica.

Con este diseño se pretende sustituir el sistema analógico actual, por uno digital de modo que se consigue un gran ahorro en tamaño y componentes empleados. Los sistemas digitales tienen muchas ventajas frente a los analógicos, por ejemplo, se consigue mayor inmunidad al ruido en las transmisiones evitando así las distorsiones que se producen en los sistemas analógicos; requieren menos ajustes hardware puesto que la mayoría se hacen mediante software, lo que se traduce también en un menor coste en actualizaciones; la integración de sistemas resulta mucho más sencilla y más fácil de estandarizar. Como resultado, el sistema digital es un sistema mucho más flexible con un menor coste hardware.

En este capítulo se muestran las especificaciones del sistema, tanto los requisitos iniciales como la especificación final propuesta para el diseño. Del mismo modo, se presentan los objetivos de aprendizaje y el entorno de desarrollo con las herramientas necesarias para implementar el prototipo, tanto la parte hardware como la parte software, y las herramientas necesarias para realizar las pruebas de funcionamiento del sistema. Las diferentes fases del proceso de desarrollo del sistema se pueden ver en el diagrama de la figura 1.

## 2.2 Especificaciones del diseño

El prototipo a desarrollar debe cumplir unas especificaciones determinadas. Las especificaciones básicas requieren que el prototipo conste de cuatro canales de medida con conexión de termopar tipo J, y un error que no supere  $\pm 1^{\circ}\text{C}$ . El procesado de las medidas se realizará de forma digital, mediante una placa de Array de Puertas Lógicas Programable FPGA (del inglés Field Programmable Gate Array). Concretamente se dispone del modelo Virtex-5 en su versión ML507 del fabricante Xilinx. El motivo del uso de esta placa es el interés en el aprendizaje de la configuración y programación de la misma, para en un futuro poder desarrollar aplicaciones más complejas en sistemas embebidos.

Tras estudiar las especificaciones básicas, se ha realizado una propuesta de especificación más detallada, en la que se añade el tipo de conversión y comunicaciones que se van a realizar.

- El prototipo constará de cuatro canales de medida.
- El sensor de medida utilizado será termopar Tipo J.
- La conversión de termopar a digital se realizará con el MAX31855 del fabricante Maxim.
- El procesado y las comunicaciones serán gestionadas mediante la FPGA modelo Virtex-5 FXT FPGA ML507 del fabricante Xilinx.
- La interfaz de usuario y la programación de los diferentes elementos se realizarán en un PC con sistema operativo Windows en la versión necesaria para cada aplicación.

A continuación, en la figura 2, se puede ver el diagrama de los bloques que componen el prototipo desarrollado. El diseño consta de dos bloques hardware diferenciados. El primero se encarga de la conversión a trama digital de las temperaturas medidas, las cuales llegan al segundo bloque mediante una comunicación serie, desde este bloque se controlan las comunicaciones y se procesan dichas medidas, una vez procesadas se envían mediante puerto serie al PC. Este, a su vez, es el último bloque del sistema, el bloque software donde se implementa la interfaz de usuario para la visualización de las medidas en tiempo real y el calibrado del sistema.

El funcionamiento de los diferentes bloques se explica en los siguientes capítulos, así como los elementos empleados y los tipos de comunicación que se realizan entre cada módulo.

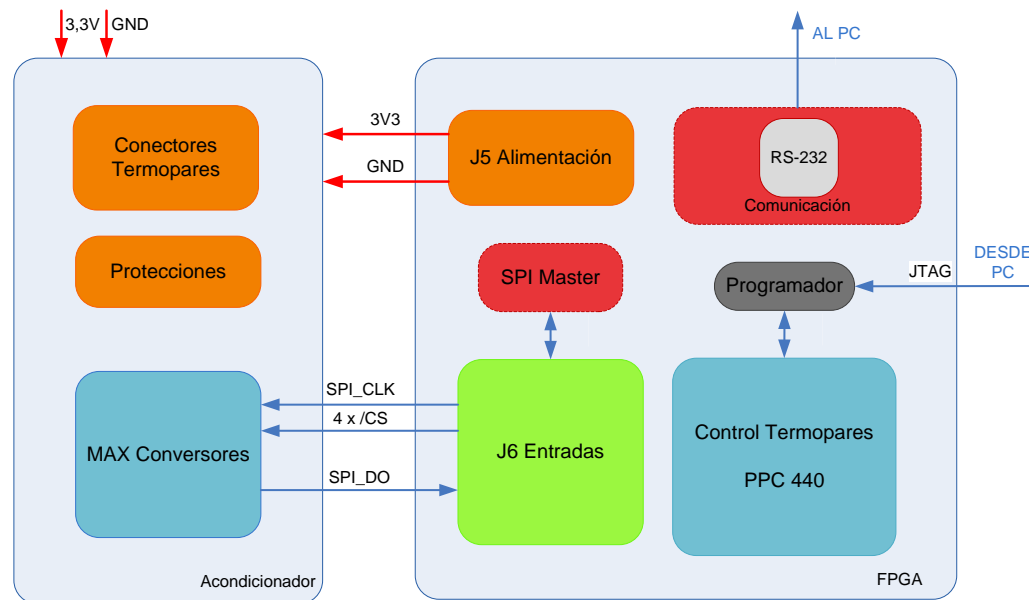


Figura 2. Diagrama de bloques del sistema completo.

## 2.3 Objetivos de aprendizaje

El diseño de este proyecto aúna conocimientos de diversos ámbitos de la electrónica. Para la parte de desarrollo hardware, se realiza el diseño de esquemáticos y rutado de placas así como su fabricación y soldadura. Por tratarse de un sistema con procesado digital mediante FPGA, el desarrollo software cobra gran importancia y requiere conocimientos de modelado en lenguaje hardware VHDL así como programación en lenguaje C. Para ello se utiliza el kit de desarrollo de Xilinx modelo Virtex-5 ML507 que se presenta en detalle en el capítulo 5.

El uso de este kit requiere también aprender a manejar el software de desarrollo asociado, el "Embedded Design Kit" (EDK). Se trata de una plataforma de desarrollo de sistemas embebidos muy potente que contiene dos programas. El primero, llamado "Xilinx Platform Studio" (XPS), permite programar la FPGA mediante los "Intellectual Property Cores" (IP Cores) que son bloques de código de descripción hardware o VHDL. En segundo lugar se dispone de la herramienta de

desarrollo software llamada “Software Development Kit” (SDK), desde donde se programa el procesador embebido en la FPGA.

### 2.1 Entorno de trabajo

Parte de las herramientas software utilizadas en este proyecto requieren un sistema operativo específico para su correcto funcionamiento, concretamente el kit de desarrollo de Xilinx EDK en su versión 11.1, compatible con Windows XP. Se dispone de un portátil con Windows 7 como sistema operativo, por tanto, para solventar el inconveniente del EDK, se utiliza dicho portátil con una máquina virtual con Windows XP para asegurar la compatibilidad con todo el paquete de herramientas de Xilinx.

Tal como se ha dicho, este proyecto está orientado al uso de un sistema basado en FPGA con el uso del kit de evaluación de la tarjeta Virtex-5 ML507 de Xilinx, así como su entorno de desarrollo EDK con los programas XPS para el diseño hardware y SDK para diseño software de dicha tarjeta.

El diseño de la placa acondicionadora se realiza de forma manual, empleando el programa OrCAD. Para el diseño del esquemático del circuito se emplea OrCAD Capture, una vez finalizado el diseño del esquemático, se enlaza con la herramienta OrCAD Layout para proceder con el diseño del PCB.

Para que la presentación de resultados sea más visual por parte del usuario, se ha implementado una aplicación en LabVIEW mediante la que se controla la adquisición de datos y el modo de visualización y guardado de los mismos.

Para los cálculos de calibración del sistema se emplea el programa Excel, en el que se cargan los datos guardados previamente por la aplicación LabVIEW. Esta herramienta permite hacer cálculos de un modo rápido y sencillo, así como visualizar los resultados de forma gráfica para poder comprobar el correcto funcionamiento del sistema final.

Para la calibración final se utiliza un calibrador profesional marca FLUKE que permite simular temperaturas de termopar tipo J con una resolución de 0,1°C. Permite también generar un perfil de temperatura para comprobar la respuesta del sistema.



El calibrador se utiliza con el sistema completo, de este modo se pueden hacer las medidas necesarias, con las que posteriormente se realizan los cálculos de ajuste pertinentes para la calibración y posterior verificación del funcionamiento del sistema completo.

Las curvas de temperaturas medidas y cálculos necesarios para hallar los parámetros de calibración del sistema se realizan mediante hojas de cálculo de Excel.



## 3 Descripción del sensor de temperatura

---

### 3.1 Introducción



Figura 3. Termopar tipo J.

La medida de temperatura ha sido muy empleada a lo largo de los años, desde los primeros termómetros de presión hasta los sensores más precisos de la actualidad. En muchos campos de ingeniería se necesita gran precisión en la medida de la temperatura, lo que ha provocado la aparición de muchos tipos de sensores diferentes.

Dependiendo de la finalidad del equipo, no es tan importante la sensibilidad del sensor como su robustez y su rango de temperaturas. Uno de los requisitos de este proyecto es el uso de termopares como sensor de medida de temperatura, este tipo de sensor se explica con más detalle en el apartado 3.4 de este capítulo. Dada su robustez, su sencillez de uso y su bajo coste, los termopares se emplean en muchos instrumentos de medida. Sin embargo, el comportamiento de los termopares puede suponer un problema para la estabilidad del sistema, debido a la falta de linealidad de su comportamiento, puesto que su respuesta tiene una desviación polinómica para las diferentes temperaturas dentro del rango de medida, al tratarse éste prototipo de un sistema con procesado digital, la no linealidad se puede compensar mediante aproximación por software, como se verá más adelante.

## 3.2 Termoelectricidad

La termoelectricidad es la parte de la física termodinámica que explica las relaciones entre el flujo de calor y las corrientes eléctricas en un material. Esta parte de la ciencia da explicación a efectos como la generación de calor en la unión de dos conductores distintos cuando circula una corriente a través de ellos o el efecto contrario, es decir, la generación de una corriente eléctrica en un conductor cuando está sometido a un gradiente de temperatura. El estudio de estos y otros efectos eléctricos relacionados con la temperatura son, en su conjunto, lo que se hace llamar termoelectricidad.

## 3.3 Termopares. El efecto Seebeck

El primer efecto termoeléctrico descubierto fue el efecto Seebeck. El físico Thomas Johann Seebeck realizó experimentos en busca de una relación entre el calor y la electricidad. En 1921 descubrió que, soldando en lazo dos alambres de materiales diferentes (cobre y bismuto en su experimento) como en la figura 4, y aplicando calor a uno de ellos mientras el otro se mantenía frío, se generaba un campo magnético entre ambos, y si dicho calor era aplicado a la unión de los dos metales se generaba una corriente eléctrica a través de ellos.

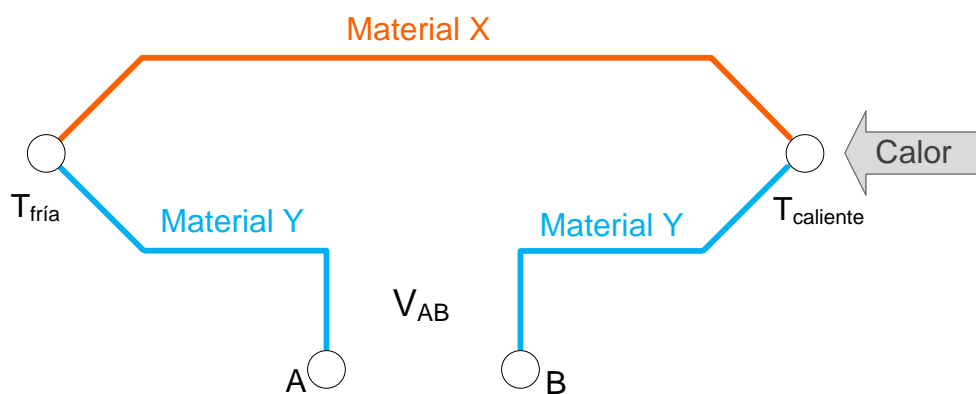


Figura 4. Efecto Seebeck: Al aplicar calor a una unión de las uniones de dos materiales diferentes, se crea una diferencia de tensión proporcional a la temperatura aplicada.

Del modelo para el estudio del efecto Seebeck, visto en la figura 4, se extrae la relación entre la temperatura de ambas uniones y la tensión que se crea entre los bornes A y B, que se presenta en la ecuación 1:

$$V_{AB} = \alpha_{xy} \cdot (T_{caliente} - T_{fría}) \quad (Ec.1)$$

*Ecuación 1. Del modelo del Efecto Seebeck se deduce la tensión diferencial proporcional a la temperatura de la unión caliente de dos hilos de materiales diferentes.*

En la ecuación 1 se tiene:

- $V_{AB}$  : Tensión de salida en voltios (V)
- $\alpha$ : Coeficiente de Seebeck entre los dos materiales, x e y, expresado en voltios por grado Kelvin (V/°K)
- $T_{caliente}$   $T_{fría}$  : Temperaturas caliente y fría de las uniones expresadas en grados Kelvin (°K)

De la ecuación 1 se obtiene la definición del Coeficiente Seebeck, que se expresa como la variación de la tensión en función de la temperatura:

$$\alpha_{xy} = \frac{\partial V_{AB}}{\partial T} \quad (Ec.2)$$

*Ecuación 2. Definición del Coeficiente Seebeck, variación de la tensión en función de la temperatura aplicada.*

Se deduce de la definición del efecto Seebeck, expresada en la ecuación 2, que no se puede medir directamente la tensión en bornes de los materiales ya que al conectar el voltímetro a éste, se crean dos nuevas uniones en el circuito. Estas dos nuevas uniones, generalmente de cobre, y la temperatura en su unión con los extremos del termopar hacen que varíe la tensión, por lo que se introduce un error en la medida realizada.

En la figura 5 se observa que debido a la conexión del termopar, la tensión a medir (V2), es función de la temperatura (V1) y de las tensiones parásitas debidas a las uniones J2 y J3. Por tanto es necesario conocer dichas tensiones parásitas, este efecto se conoce como “unión fría” y debe ser compensado para obtener una medida fiable.

### 3. Descripción del sensor de temperatura

Para ello, se debe saber la relación entre los materiales y la temperatura a la que están expuestos.

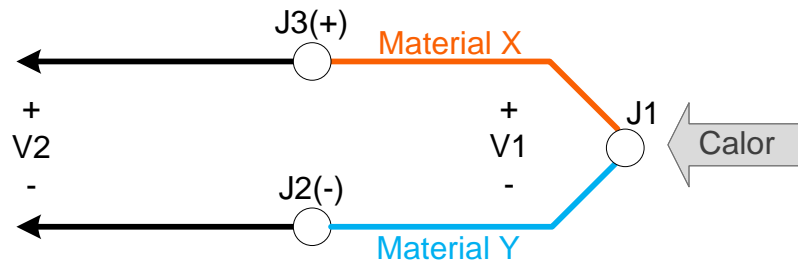


Figura 5. Unión fría creada al conectar un instrumento de medida en los bornes del termopar, con ello se crean tensiones parásitas que deben ser compensadas.

Un modo de conocer las tensiones parásitas es la inserción de un cable intermedio en una de las uniones, este cable debe ser del mismo material que el del hilo contrario, estas nuevas uniones deben estar contenidas en un ambiente isotérmico de temperatura conocida, tal como se muestra en la figura 6.

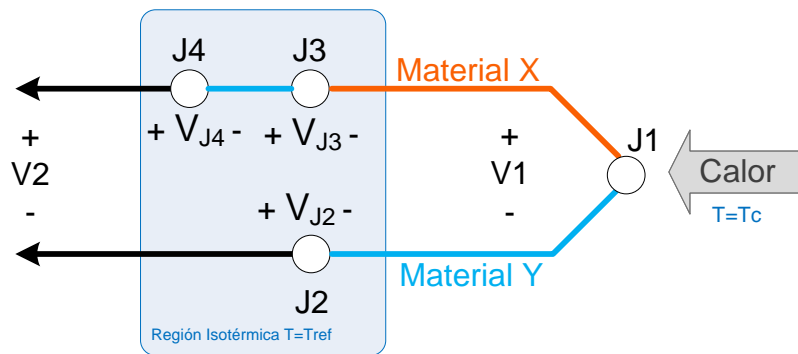


Figura 6. Modelo de compensación de la unión fría mediante región isotérmica. Se crean dos nuevas uniones de las mismas características (J2 y J4) de este modo se anulan las tensiones parásitas ( $V_{J2}$  y  $V_{J4}$ ). La compensación de la tensión parásita  $V_{J3}$  se debe hacer mediante software o hardware.

Se puede observar que las uniones J2 y J4 son del mismo material, por lo que tendrán el mismo comportamiento frente a la temperatura de la unión J1 lo que hace que sus tensiones parásitas ( $V_{J2}$  y  $V_{J4}$ ) se anulen. Por otro lado, la unión J3 tiene las mismas características que la unión J1, pero se encuentran a temperaturas diferentes. De esta manera se obtiene la siguiente relación:

$$V_2 = V_1(T_c) + V_{J3}(T_{ref}) \quad (Ec.3)$$

Ecuación 3 Tensión resultante de la compensación de unión fría mediante región isotérmica.

La compensación de la tensión parásita  $V_{J3}$  se realiza mediante hardware o software dependiendo del tipo de diseño. La compensación hardware se realiza mediante una fuente de tensión contraria a la tensión  $V_{J3}$  de modo que ésta queda anulada y no afecta a la salida, la corrección hardware resulta costosa tanto en diseño como en materiales, por lo que en la actualidad se emplea muy poco. La compensación software requiere un procesador y un sensor de temperatura. Se procede midiendo la temperatura de la región isotérmica, que es la temperatura a la que está expuesta la unión J3; el procesador se encarga de hacer el cálculo de la tensión parásita y de restarlo a la salida, de modo que el valor de salida corresponde a la tensión resultante de la temperatura aplicada a la unión caliente. La compensación software tiene un diseño mucho más sencillo y menos costoso, resultando más fiable y más fácil de calibrar en caso de que se detecten desviaciones en la medida realizada.

### 3.4 Tipos de Termopares

El efecto Seebeck se ha utilizado para la fabricación de sensores de temperatura (también llamados termopares). Tal como se ha detallado en el apartado anterior, manteniendo una de las dos soldaduras entre los dos materiales a una temperatura controlada y relativamente fría y utilizando el otro punto de unión de los materiales como detector de la temperatura de interés, se crea una diferencia de potencial  $V_2$ , vista en la figura 7, directamente relacionada con la diferencia de temperatura entre las uniones de los dos materiales.

El termopar es el sensor de temperatura más utilizado comercialmente ya que se puede emplear para medir amplios intervalos de temperatura, son muy robustos, tienen una buena exactitud (si no se requiere la medida de variaciones muy pequeñas de temperatura), una rápida respuesta y un bajo coste.

Existen varios modelos de termopares dependiendo de los materiales que componen sus uniones como pueden ser: Cromo-Aluminio, Cromo-Constantan, Hierro-Constantan, Platino-Rodio, etc. A continuación se hace una breve descripción de los tipos más comunes, donde se detalla el rango de temperaturas y la sensibilidad media de cada tipo.

### 3. Descripción del sensor de temperatura

---

- Tipo K (Cromo (Ni-Cr) Chromel / Aluminio (aleación de Ni -Al) Alumel): con una amplia variedad de aplicaciones, está disponible a un bajo costo y en una variedad de sondas. Tienen un rango de temperatura de  $-200^{\circ}\text{C}$  a  $+1.372^{\circ}\text{C}$  y una sensibilidad media de  $41\mu\text{V}/^{\circ}\text{C}$  aproximadamente. Posee buena resistencia a la oxidación.
- Tipo E (Cromo / Constantan (aleación de Cu-Ni)): tienen una sensibilidad superior a los demás tipos, gracias a ello, son ideales para el uso en bajas temperaturas, en el ámbito criogénico. Tienen una sensibilidad media de  $68\mu\text{V}/^{\circ}\text{C}$ .
- Tipo J (Hierro / Constantan): debido a su limitado rango, el tipo J es menos popular que el K. El tipo J no puede usarse a temperaturas superiores a  $760^{\circ}\text{C}$  ya que una abrupta transformación magnética causa una descalibración permanente. Tienen un rango de  $-40^{\circ}\text{C}$  a  $+750^{\circ}\text{C}$  y una sensibilidad media de  $52\mu\text{V}/^{\circ}\text{C}$ . Es afectado por la corrosión.
- Tipo N (Nicrosil (Ni-Cr-Si / Nisil (Ni-Si)): es adecuado para mediciones de alta temperatura gracias a su elevada estabilidad y resistencia a la oxidación de altas temperaturas, y no necesita del platino utilizado en los tipos B, R y S que son más caros.

Por otro lado, los termopares tipo B, R y S son los más estables, pero debido a su baja sensibilidad media de unos  $10\mu\text{V}/^{\circ}\text{C}$ , generalmente son usados para medir altas temperaturas (por encima de los  $1000^{\circ}\text{C}$ ).

- Tipo B (Platino (Pt)-Rodio (Rh)): son adecuados para la medición de altas temperaturas superiores a  $1.800^{\circ}\text{C}$ . El tipo B por lo general presentan el mismo resultado a  $0^{\circ}\text{C}$  y  $42^{\circ}\text{C}$  debido a su curva de comportamiento.
- Tipo R (Platino (Pt)-Rodio (Rh)): adecuados para la medición de temperaturas de hasta  $1.300^{\circ}\text{C}$ . Su baja sensibilidad media ( $10\mu\text{V}/^{\circ}\text{C}$ ) y su elevado precio hacen que sea menos utilizado que otros tipos.
- Tipo S (Platino / Rodio): ideales para mediciones de altas temperaturas hasta los  $1.300^{\circ}\text{C}$ , pero su baja sensibilidad ( $10\mu\text{V}/^{\circ}\text{C}$ ) y su elevado precio lo



convierten en un instrumento no adecuado para el uso general. Debido a su elevada estabilidad, el tipo S es utilizado para la calibración universal del punto de fusión del oro (1064,43°C).

En la tabla 1 se puede observar una clasificación de los tipos de termopar más comunes, incluyendo el material de sus uniones, así como el código de colores por el que se diferencian y el rango máximo de temperatura en el que se emplean.


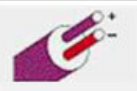
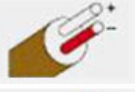



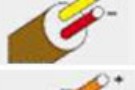
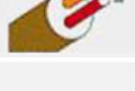


Código ANSI	Combinación de la aleación		Código de color		Rango Máximo de Temperatura	
	+	-	Termopar	Extensión	Termopar	Extensión
<b>E</b>	Cromel Níquel-Cromo Ni-Cr	Constantan Cobre-Níquel Cu-Ni			-270 a 1000°C	0 a 200°C
<b>J</b>	Hierro Fe	Constantan Cobre-Níquel Cu-Ni			-40 a 750°C	0 a 200°C
<b>T</b>	Cobre Cu	Constantan Cobre-Níquel Cu-Ni			-270 a 400°C	-60 a 100°C
<b>K</b>	Cromel Níquel-Cromo Ni-Cr	Alumel Níquel-Aluminio Ni-Al			-270 a 1372°C	0 a 200°C
<b>N</b>	Nicrosil Níquel-Cromo-Silicio Ni-Cr-Si	Nisil Níquel-Silicio-Magnesio Ni-Si-Mg			-270 a 1300°C	0 a 200°C
<b>R</b>	Platino-13%Rodio Pr-13%Rh	Platino Pt	No establecido		-60 a 1768°C	0 a 150°C
<b>S</b>	Platino-10%Rodio Pr-10%Rh	Platino Pt	No establecido		-60 a 1768°C	0 a 150°C
<b>B</b>	Platino-30%Rodio Pr-30%Rh	Platino-6%Rodio Pr-6%Rh	No establecido		0 a 1820°C	0 a 100°C

Tabla 1. Comparativa de los tipos de de termopares más comunes según material y rango de temperatura.

Como se puede ver en la figura 7, para los diferentes tipos de termopar existen diferentes gráficos de comportamiento. Se puede observar que a partir de 100°C la linealidad prácticamente se conserva. A temperaturas más bajas la aproximación de las medidas respecto a la temperatura real es polinómica.

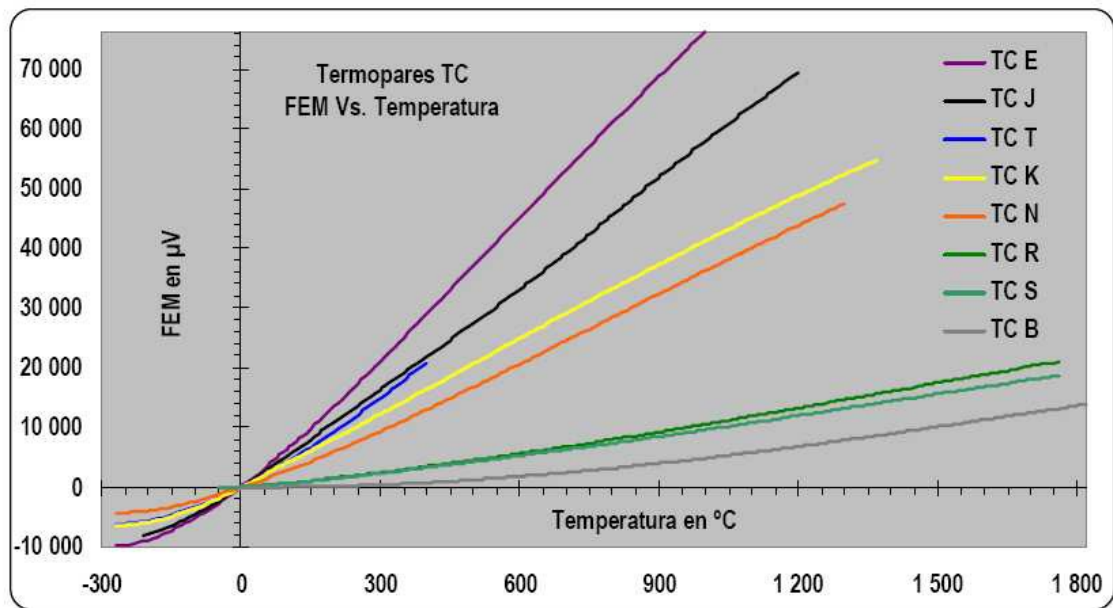


Figura 7. Tensión de termopar frente a temperatura según los diferentes tipos de termopares.

La mayoría de controladores de temperatura comerciales utilizan la compensación por software para realizar la medición de temperatura en un dispositivo termopar. Así mismo, también realizan los cálculos necesarios para corregir la no linealidad de estos dispositivos a bajas temperaturas.

### 3.5 Sensor Utilizado: Tipo J

Uno de los requisitos de este proyecto es el empleo de termopares tipo J. Tal como se ha observado en la tabla 1, este tipo de termopar cubre el rango de temperaturas que se pretende medir, que será de 0 a 200°C. Tiene una sensibilidad media de unos 52µV/°C.

A la hora de calibrar el sistema, se deberá tener en cuenta que esta sensibilidad no es lineal en todo el rango de medida sino que varía según la temperatura que se esté midiendo, existen unos coeficientes para el cálculo de calibración, como se puede ver en la tabla 2, estos coeficientes se aplican tal como se indica en la ecuación 4:

$$E_{0j} = b_0 + b_1T_j + b_2T_{2j} + \dots + b_nT_{nj}; \text{ donde } T_j \text{ en } ^\circ\text{C y } E_{0j} \text{ en mV} . \quad (\text{Ec. 4})$$

Ecuación 4. Calibración de termopar tipo J por coeficientes en función de la temperatura.

	$-210^\circ\text{C} \leq T \leq 760^\circ\text{C}$	$760^\circ\text{C} \leq T \leq 1200^\circ\text{C}$
$b_0 =$	0	$2,9645625681 \cdot 10^2$
$b_1 =$	$5,0381187185 \cdot 10^{-2}$	-1,4976127790
$b_2 =$	$3,0475386930 \cdot 10^{-5}$	$3,1787103294 \cdot 10^{-3}$
$b_3 =$	$-8,5681065720 \cdot 10^{-8}$	$-3,1847686701 \cdot 10^{-6}$
$b_4 =$	$1,3228195295 \cdot 10^{-10}$	$1,5720819004 \cdot 10^{-9}$
$b_5 =$	$-1,7052958337 \cdot 10^{-13}$	$-3,0691369056 \cdot 10^{-13}$
$b_6 =$	$2,0948090690 \cdot 10^{-16}$	0
$b_7 =$	$-1,2538395226 \cdot 10^{-19}$	0
$b_8 =$	$1,5631725697 \cdot 10^{-23}$	0

Tabla 2. Coeficientes para la calibración de termopar tipo J en función del rango de temperatura.

El conversor de termopar a trama digital que se utiliza en este proyecto, realiza tanto la compensación de la unión fría como la calibración de las temperaturas. El conversor transforma dichas medidas en su valor decimal en  $^\circ\text{C}$  correspondiente. Se debe tener en cuenta tanto el tipo de calibración que realiza el conversor, como el resultado de la compensación interna de la unión fría. Estos puntos se estudian a lo largo de esta memoria, se podrá comprobar que el conversor asume un comportamiento lineal en los termopares, y que para mejorar la compensación de la unión fría, el chip se deberá colocar lo más cerca posible del conector de termopar y en un ambiente lo más estable posible, evitando corrientes de aire.



## 4 Descripción del conversor

---

### 4.1 Introducción



Figura 8. Encapsulado MAX31855.

Para convertir la tensión generada en el termopar a un valor digital correspondiente a la temperatura medida, en rasgos generales se deberá emplear un amplificador operacional, puesto que el rango de tensiones del termopar es muy pequeño, también sería necesario el uso de un conversor analógico digital o ADC (Analog to Digital Converter), así como un sensor de temperatura para poder hacer la compensación de la unión fría tal como se explica en el apartado anterior.

Los avances en miniaturización y el crecimiento en la oferta de componentes que integran varias funcionalidades en un solo chip, nos permiten encontrar elementos como el que se utiliza en este proyecto. Se trata de un conversor de termopar a digital que incluye compensación de la unión fría.

A continuación se detallan los aspectos más importantes a tener en cuenta en el diseño de la placa acondicionadora y al mismo tiempo se describe el conversor empleado, incluyendo la información más relevante de la hoja de especificaciones facilitada por el fabricante.

## 4.2 Conversor de Termopar a Digital con Compensación de la Unión Fría

Para la conversión de termopar a digital se utiliza el “Cold-Junction Compensated Thermocouple-to-Digital Converter”(en español: Conversor de Termopar a Digital con Compensación de Unión Fría) cuya referencia es MAX31855 del fabricante Maxim, concretamente se trata del MAX31855JASA+ que es el correspondiente al termopar tipo J.

### 4.2.1 Descripción Hardware

Este componente reúne los elementos necesarios para hacer la conversión de termopar a digital ya comentados en el apartado 4.1 de este capítulo. En la figura 9 se puede observar la composición interna del mismo, que consta de una entrada para termopar que se chequea para conocer el estado de la conexión, seguida de un amplificador operacional. Dispone de un bloque especial dedicado a la compensación de la unión fría, que incluye un sensor que realiza la medición de la temperatura ambiente y le aplica la sensibilidad del termopar utilizado, por esta razón es importante tener en cuenta que el tipo de termopar deben coincidir con el encapsulado del MAX utilizado. La salida del amplificador ya compensada es enviada a un ADC con control digital, quedando a la salida del MAX las tres señales digitales que forma la comunicación SPI, éstas son SCK que es el reloj que sincroniza la comunicación, SO es la salida de datos digital por donde se enviará la trama hacia la FPGA, y /CS es la señal de entrada de selección que se activará desde la FPGA.

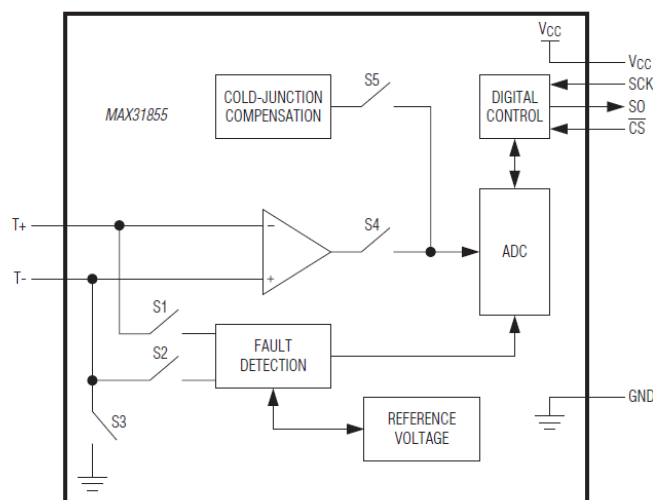


Figura 9. Diagrama de bloques. MAX31855, incluye un amplificador operacional, detección del estado de la conexión del termopar y un bloque especial para la compensación de la unión fría, así como un ADC para hacer la conversión de termopar a digital

En la figura 10 se puede ver el circuito típico de conexión del MAX31855 recomendado por el fabricante, donde se especifica la entrada del termopar indicando el positivo y el negativo. Se observa también, a modo de ejemplo, la conexión de la salida hacia un microcontrolador, pero en este proyecto el procesado digital se realiza mediante el procesador PowerPC de la tarjeta Virtex-5.

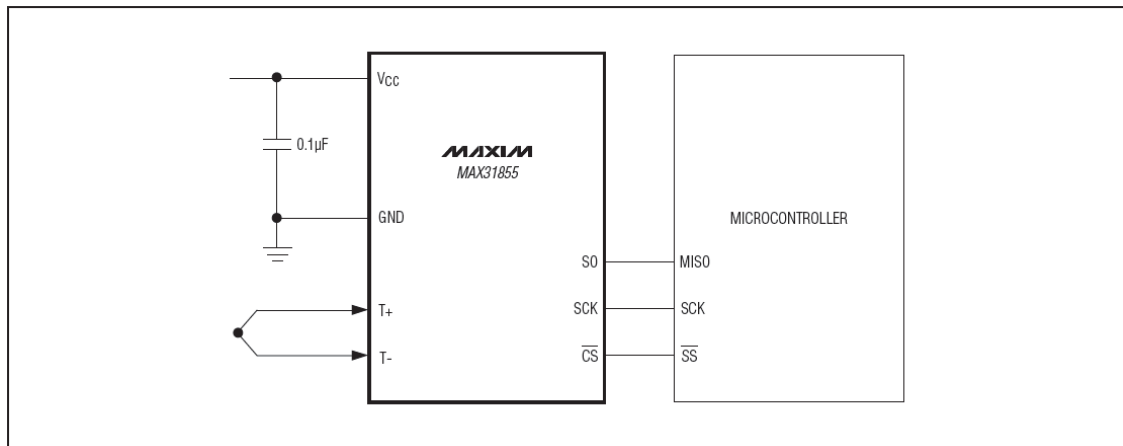


Figura 10. Circuito de aplicación típico del MAX31855, a la entrada se conecta el termopar, la comunicación SPI se gestionará en este caso con el procesador de la FPGA.

Las condiciones de trabajo recomendadas por el fabricante se resumen en la tabla 3. La tensión de alimentación típica es de 3,3V que es un valor común en la alimentación de circuitos digitales, esta alimentación se podrá obtener de la FPGA ya que dispone de pines de salida de alimentación a 3,3V. Se debe tener en cuenta también que se respeta el rango de tensión de los niveles lógicos,  $V_{IL}$  para nivel bajo y  $V_{IH}$  para el nivel alto.

#### PARÁMETROS DE FUNCIONAMIENTO

( $T_A = -40^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$ , unless otherwise noted.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Power-Supply Voltage	$V_{CC}$	(Note 2)	3.0	3.3	3.6	V
Input Logic 0	$V_{IL}$		-0.3		+0.8	V
Input Logic 1	$V_{IH}$		2.1		$V_{CC} + 0.3$	V

Tabla 3. Condiciones de trabajo del MAX31855 recomendados por el fabricante.

En la tabla 4 se pueden ver las características térmicas del MAX31855 en su uso con termopares tipo J, de la primera fila de esta tabla se deduce que el conversor asume una sensibilidad del termopar constante, con un valor de  $57,953\mu\text{V}/^{\circ}\text{C}$ , por lo que la compensación no tendrá en cuenta la no linealidad en la medida del termopar.

#### 4. Descripción del conversor

Se puede observar también la resolución de la conversión de temperatura del termopar, que es de  $0,25^{\circ}\text{C}$ , y la resolución de la temperatura medida en la unión fría, que es de  $0,0625^{\circ}\text{C}$ . Es importante tener en cuenta estos valores a la hora de procesar las medidas en el procesador de la FPGA

#### CARACTERÍSTICAS TÉRMICAS

( $3.0\text{V} \leq V_{\text{CC}} \leq 3.6\text{V}$ ,  $T_{\text{A}} = -40^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$ , unless otherwise noted.) (Note 4)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
MAX31855J Thermocouple Temperature Gain and Offset Error ( $57.953\mu\text{V}/^{\circ}\text{C}$ nominal sensitivity) (Note 4)		$T_{\text{THERMOCOUPLE}} = -210^{\circ}\text{C}$ to $+750^{\circ}\text{C}$ , $T_{\text{A}} = -20^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ (Note 3)	-2		+2	$^{\circ}\text{C}$
		$T_{\text{THERMOCOUPLE}} = -210^{\circ}\text{C}$ to $+1200^{\circ}\text{C}$ , $T_{\text{A}} = -40^{\circ}\text{C}$ to $+125^{\circ}\text{C}$ (Note 3)	-4		+4	
Thermocouple Temperature Data Resolution				0.25		$^{\circ}\text{C}$
Internal Cold-Junction Temperature Error		$T_{\text{A}} = -20^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ (Note 3)	-2		+2	$^{\circ}\text{C}$
		$T_{\text{A}} = -40^{\circ}\text{C}$ to $+125^{\circ}\text{C}$ (Note 3)	-3		+3	
Cold-Junction Temperature Data Resolution		$T_{\text{A}} = -40^{\circ}\text{C}$ to $+125^{\circ}\text{C}$		0.0625		$^{\circ}\text{C}$
Temperature Conversion Time (Thermocouple, Cold Junction, Fault Detection)	$t_{\text{CONV}}$	(Note 5)		70	100	ms
Thermocouple Conversion Power-Up Time	$t_{\text{CONV\_PU}}$	(Note 6)	200			ms

Tabla 4. Características térmicas y tiempos de conversión del MAX31855 en su versión para termopar tipo J.

El error introducido por el conversor para la temperatura del termopar es de  $\pm 2^{\circ}\text{C}$ , en el rango de temperaturas de  $-210^{\circ}\text{C}$  a  $750^{\circ}\text{C}$ , que es el rango en el que trabajará el prototipo. Para unas condiciones de trabajo con temperatura ambiente entre  $-20^{\circ}\text{C}$  y  $85^{\circ}\text{C}$ , el error introducido en la medida de la temperatura de la unión fría puede llegar también a los  $\pm 2^{\circ}\text{C}$ .

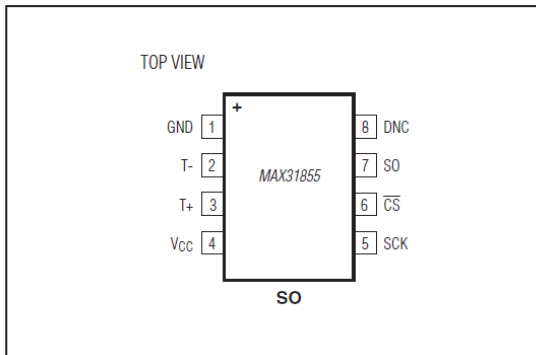
Tanto el error introducido por el conversor, como el error en la conversión debido a la no linealidad de la medida real del termopar, deberán ser minimizados mediante calibración software.

Puesto que la aplicación del prototipo no requiere una velocidad de transmisión muy elevada, los tiempos de conversión se respetarán sin problemas ya que el tiempo entre medidas será mucho mayor de 100ms. El tiempo de inicialización también se respetará, puesto que la inicialización del resto de elementos del sistema es mayor que el tiempo de inicialización del MAX31855 de 200ms.

En la figura 11 se encuentra la descripción de pines del MAX31855, así como su disposición en el encapsulado de ocho pines de montaje superficial tipo SOP, que es el encapsulado utilizado en este diseño.



## PINOUT



PIN	NAME	FUNCTION
1	GND	Ground
2	T-	Thermocouple Input. See Table 1. Do not connect to GND.
3	T+	Thermocouple Input. See Table 1.
4	V <sub>CC</sub>	Power-Supply Voltage
5	SCK	Serial-Clock Input
6	$\overline{CS}$	Active-Low Chip Select. Set $\overline{CS}$ low to enable the serial interface.
7	SO	Serial-Data Output
8	DNC	Do Not Connect

Figura 11. Pinout del MAX31855. Encapsulado tipo SOP de 8 pines.

### 4.2.2 Descripción Software

En toda comunicación digital es necesario conocer la configuración de la señal, En la figura 12 y 13 se muestran los diagramas del protocolo y los tiempos a tener en cuenta en la comunicación serie del sistema entre el bloque de conversión y el de procesado digital, de estas imágenes se obtiene la información del flanco activo de reloj, de subida en este caso, así como el orden de envío de los bits de datos. En el diagrama del protocolo se observa que el bit de menor peso D0 es el primero que se envía, por lo que se sabe que en recepción, en la FPGA, el primer bit recibido será el menos significativo o LSB. En el diagrama de tiempos aparecen indicados los tiempos mínimos que se deben respetar para que la comunicación se realice de forma correcta. Estos tiempos se detallan en la tabla 5, de donde se obtiene la información necesaria para configurar la comunicación desde la FPGA, en esta tabla, el fabricante indica que la frecuencia de reloj deberá ser como máximo de 5MHz.

#### PROTOCOLO DE INTERFAZ SERIE

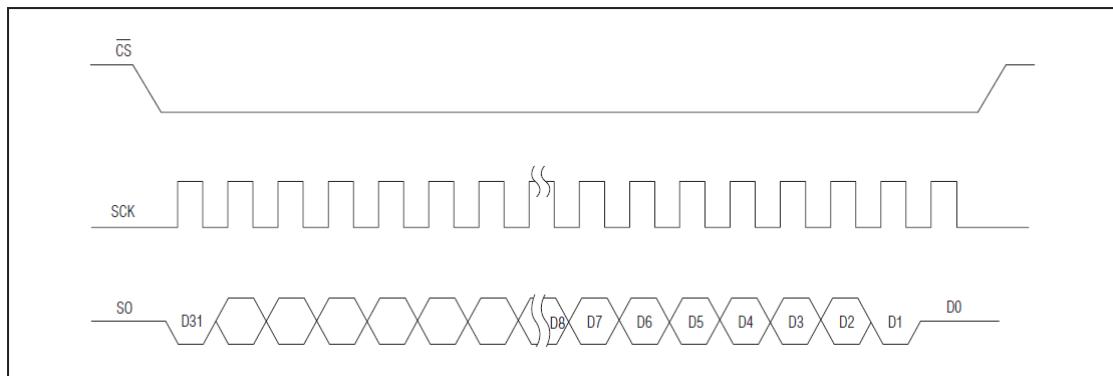


Figura 12. Diagrama del protocolo de la comunicación SPI.

**TEMPORIZACIÓN DE INTERFAZ SERIE**

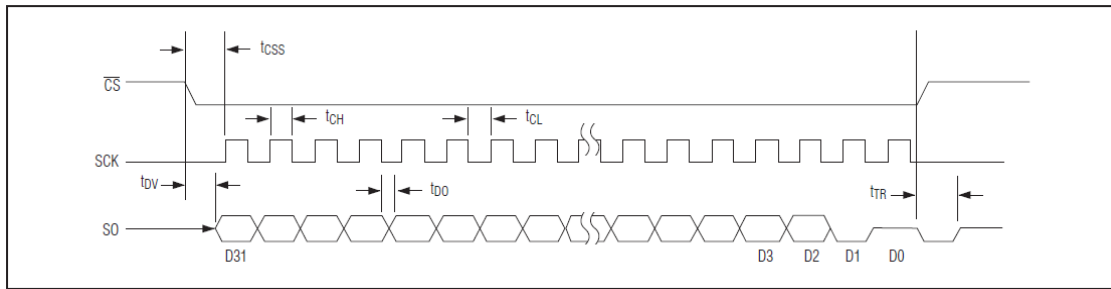


Figura 13. Diagrama de tiempos de la comunicación SPI.

**CARACTERÍSTICAS DE TIEMPOS COMUNICACIÓN SPI**

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Input Leakage Current	$I_{LEAK}$	(Note 7)	-1		+1	$\mu A$
Input Capacitance	$C_{IN}$			8		pF
Serial-Clock Frequency	$f_{SCL}$				5	MHz
SCK Pulse-High Width	$t_{CH}$		100			ns
SCK Pulse-Low Width	$t_{CL}$		100			ns
SCK Rise and Fall Time					200	ns
CS Fall to SCK Rise	$t_{CSS}$		100			ns
SCK to CS Hold			100			ns
CS Fall to Output Enable	$t_{DV}$				100	ns
CS Rise to Output Disable	$t_{TR}$				40	ns
SCK Fall to Output Data Valid	$t_{DO}$				40	ns
CS Inactive Time		(Note 3)	200			ns

Tabla 5. Características de los tiempos en la comunicación SPI.

En el diagrama de la figura 12 se ha observado el protocolo de la trama de comunicación SPI. La descripción del contenido de esta trama aparece en la tabla 6, Los tres bits de menor peso D0, D1, D2 informan sobre el estado de la conexión del termopar, si está cortocircuitado a masa, a alimentación o si no está conectado. Los siguiente doce bits D[15..4] contienen el valor de la temperatura medida en la unión fría, con el bit D16 se puede chequear si cualquiera de los tres bits más bajos está activado, es decir, si hay algún fallo en la conexión del termopar. Los catorce bits de mayor peso D[31..18] contienen el valor correspondiente a la medida del termopar, con una resolución de 0,25°C, como se ha explicado en las características generales del convertor.

## MAPA DE MEMORIA

BIT	NAME	DESCRIPTION
D[31:18]	14-Bit Thermocouple Temperature Data	These bits contain the signed 14-bit thermocouple temperature value. See <a href="#">Table 4</a> .
D17	Reserved	This bit always reads 0.
D16	Fault	This bit reads at 1 when any of the SCV, SCG, or OC faults are active. Default value is 0.
D[15:4]	12-Bit Internal Temperature Data	These bits contain the signed 12-bit value of the reference junction temperature. See <a href="#">Table 5</a> .
D3	Reserved	This bit always reads 0.
D2	SCV Fault	This bit is a 1 when the thermocouple is short-circuited to $V_{CC}$ . Default value is 0.
D1	SCG Fault	This bit is a 1 when the thermocouple is short-circuited to GND. Default value is 0.
D0	OC Fault	This bit is a 1 when the thermocouple is open (no connections). Default value is 0.

Tabla 6. Descripción del Mapa de Memoria. Indica la posición de los bits y la información de la trama.

Para poder procesar las temperaturas contenidas en la trama serie, es necesario conocer el formato de los datos. Esta información se encuentra detallada en los ejemplos de la tabla 7. A la izquierda se encuentran ejemplos con el formato de la trama, en binario, correspondiente al valor de temperatura del termopar y a la derecha, el correspondiente a la temperatura de la unión fría. Como ya se indicaba en la tabla 4, la resolución del termopar es de  $0.25^{\circ}\text{C}$ , las dos cifras decimales vienen dadas por los dos últimos bits, sin embargo, para la temperatura de la unión, la resolución de  $0,625^{\circ}\text{C}$  y estos tres decimales vendrán dados por los cuatro últimos bits.

Es importante estudiar estas tablas con detalle para encontrar el mejor modo de procesar la información una vez que se reciba en la FPGA para su posterior calibración.

## TEMPERATURA TERMOPAR

TEMPERATURE (°C)	DIGITAL OUTPUT (D[31:18])
+1600.00	0110 0100 0000 00
+1000.00	0011 1110 1000 00
+100.75	0000 0110 0100 11
+25.00	0000 0001 1001 00
0.00	0000 0000 0000 00
-0.25	1111 1111 1111 11
-1.00	1111 1111 1111 00
-250.00	1111 0000 0110 00

## TEMPERATURA UNIÓN FRÍA

TEMPERATURE (°C)	DIGITAL OUTPUT (D[15:4])
+127.0000	0111 1111 0000
+100.5625	0110 0100 1001
+25.0000	0001 1001 0000
0.0000	0000 0000 0000
-0.0625	1111 1111 1111
-1.0000	1111 1111 0000
-20.0000	1110 1100 0000
-55.0000	1100 1001 0000

Tabla 7. Formato de la temperatura en la trama de comunicación serie. Izquierda: temperatura del termopar, derecha: Temperatura de la unión fría.



# 5 Sistema de control y procesado: FPGA

---

## 5.1 Introducción

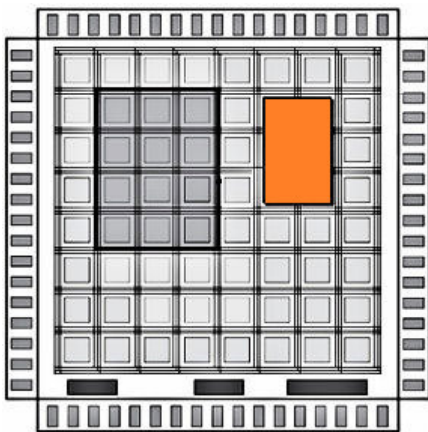


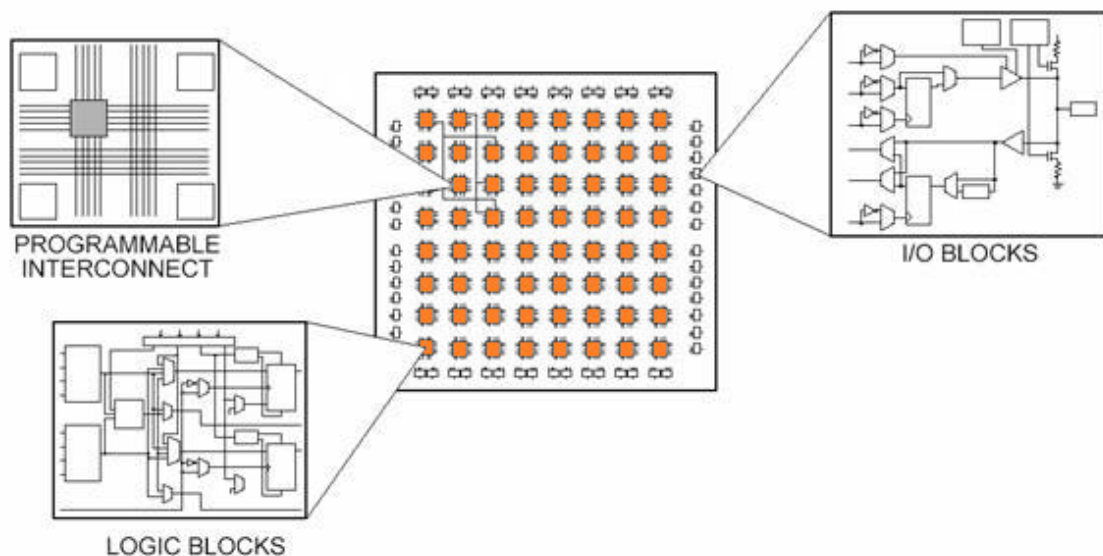
Figura 14. Vista simplificada de una FPGA.

El uso de FPGAs en diseños electrónicos está cada vez más extendido debido en parte a la bajada de precios y a los avances en las técnicas de fabricación de placas electrónicas. Las FPGAs resultan idóneas para prototipado por la gran flexibilidad que tienen en cuanto a diseño, así como la facilidad de reprogramación, lo que supone una gran ventaja a la hora de hacer pruebas en un sistema. En la actualidad hay modelos que incorporan un procesador embebido, representado en naranja en la figura 14, de modo que la eficiencia y velocidad de procesado aumenta respecto a los modelos en los que el procesador es simulado mediante la lógica interna de la FPGA.

El sistema de control de comunicaciones y procesado de señales de este proyecto se implementa mediante el kit de desarrollo de Xilinx modelo Virtex-5 FXT ML507 basado en la FPGA modelo XC5VFX70T-2FFG1136C, la cual tiene embebido un procesador PowerPC440. En este capítulo se presentan las nociones básicas de FPGAs y las herramientas proporcionadas por el fabricante para el desarrollo de los sistemas embebidos.

## 5.2 FPGA

Una FPGA (Field Programmable Gate Array) es un dispositivo semiconductor formado por un conjunto de bloques lógicos programables y recursos de interconexión también programables. Los bloques lógicos pueden ser programados para duplicar la funcionalidad de puertas lógicas, tales como AND, OR, XOR, NOT o funciones combinatoriales más complejas, tales como decodificadores o simples funciones matemáticas. En muchas FPGA, estos componentes lógicos programables también incluyen elementos de memoria, los cuales pueden ser simples flip-flops o bloques de memoria más complejos. El código se programa directamente en el hardware, de modo que éste puede adoptar casi cualquier arquitectura. Las herramientas de síntesis determinan la forma de configurar todos los elementos que componen la FPGA para que el funcionamiento sea el descrito por el programador. En la figura 15 se puede ver de forma esquemática la arquitectura de una FPGA, en naranja se representan los bloques lógicos, y en los cuatro extremos los bloques de entrada salida (bloques I/O), dichos bloques están interconectados por una red reconfigurable de pistas.



Los últimos avances en esta tecnología se centran en conseguir un “Sistema programable en un único chip”, para ello se combinan los bloques lógicos e interconexiones tradicionales de las FPGAs, con microprocesadores y periféricos relacionados. La tendencia es embeber éstos en el chip de la FPGA de modo que se combinan los soft-cores (bloques lógicos e interconexiones) con hard-cores, es decir el chip de silicio que compone el procesador embebido, tal como se ha visto en la figura 14, representado en color naranja.

### 5.3 Kit de desarrollo

Como ya se ha comentado, este proyecto se basa en la FPGA Virtex-5 modelo XC5VFX70T-2FFFG1136C de Xilinx, ésta FPGA contiene un procesador PowerPC440 embebido. El kit de evaluación que se utiliza para el procesado digital y control de comunicaciones del prototipo es el Virtex-5 ML507 que se muestra en la figura 16, incluye una gran diversidad de periféricos a disposición del diseñador. Cada periférico deberá ir asociado a un código diseñado en lenguaje de descripción hardware (VHDL), dichos bloques son los llamados IPCore (Bloque de Propiedad Intelectual). Contienen parte de la información necesaria para la configuración interna de la FPGA, para completar el diseño se deben describir, también en lenguaje de descripción hardware, las restricciones que tiene el diseño, como por ejemplo señales de reloj, señales que requieren un retardo específico, y demás elementos críticos. Estos programas se implementan en la FPGA por medio de sus herramientas de desarrollo asociadas.

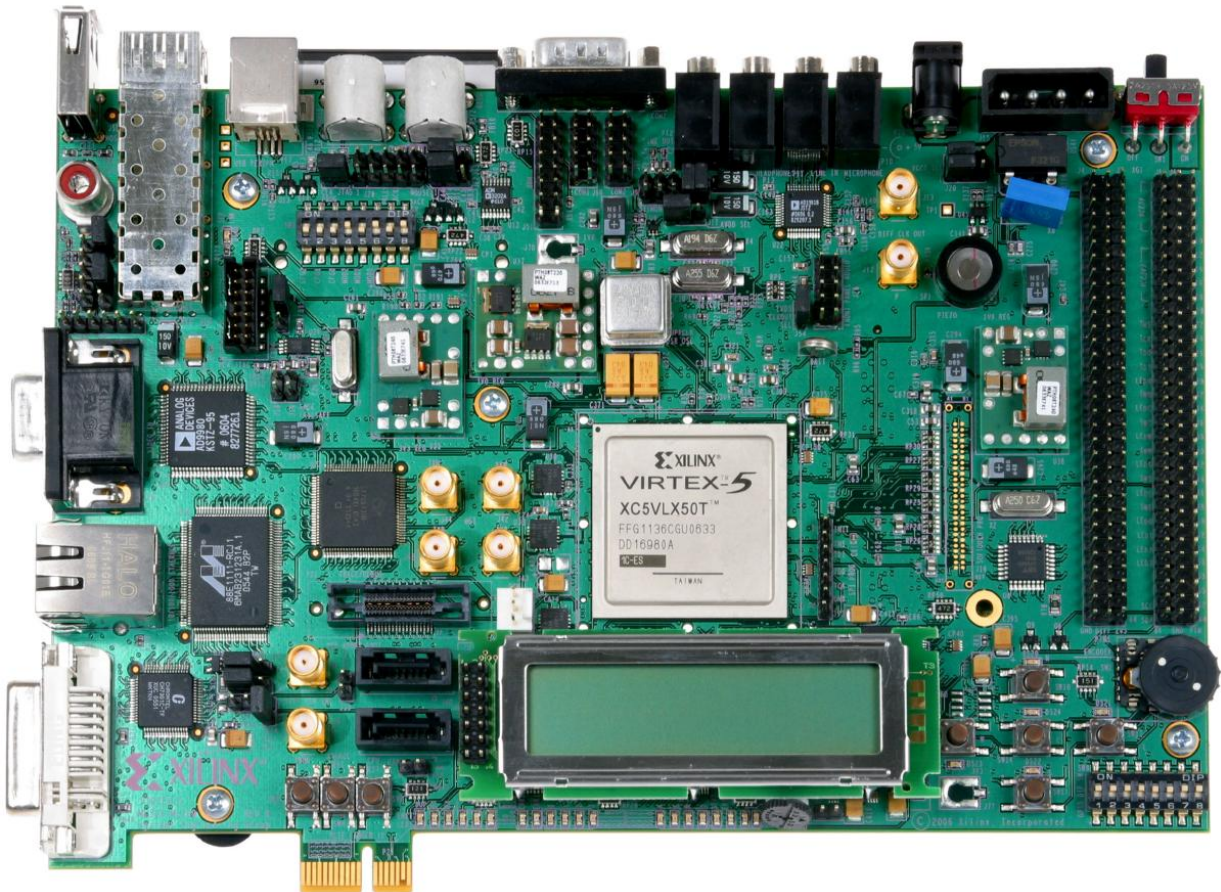


Figura 16. Placa de desarrollo Virtex-5 ML507.

## 5.4 Diagrama de bloques

La placa de desarrollo Virtex-5 FXT ML507 contiene, como ya se ha visto, el modelo de FPGA XC5VFX70T-2FFG1136C del fabricante Xilinx. Además de numerosos periféricos disponibles para su uso mediante la configuración de la FPGA, como pueden ser entradas de audio, vídeo, USB, memorias, pulsadores, LEDs, pines de entrada salida de propósito general (GPIO), etc. A continuación se describen los elementos utilizados en el diseño.

- **Microprocesador:** Es un PowerPc 440 embebido en la *FPGA* en forma de bloque y que se configura mediante las herramientas de Xilinx.
- **Comunicaciones externas:** Se refiere a todos los elementos que permiten la comunicación de la tarjeta con otro dispositivo externo.
  - **Serie:** Para la comunicación serie se utiliza un IPCore que implementa una UART en la FPGA.
  - **SPI:** Utilizado para controlar la comunicación SPI si se configura como maestro.
  - **JTAG:** La conexión JTAG permite configurar la PROM donde se almacena la configuración de la FPGA y depurar el PowerPc 440.
- **Memorias:** El bloque de memorias está formado por tres tipos de memorias:
  - **DRAM:** Se dispone de 512Mbyte de memoria DDR2 SDRAM de 333MHz, distribuida en dos módulos de 256 Mbytes (modelo MT47H128M16HG-3 IT de Micron).
  - **Flash:** Se dispone 128 Mbyte de memoria de tipo NOR (modelo PC28F00AP30TF de Numonyx).
  - **PROM:** Tiene 32 Mbyte y es el modelo XCF32P de Xilinx.
- **Entrada salida de propósito general (GPIO):** Agrupa diferentes elementos de entrada/salida de propósito general:
  - **LEDs:** Se dispone de ocho diodos LED.
  - **Conector de expansión:** El conector J6 de la placa, se puede configurar tanto en modo diferencial como single. Se utiliza para la conexión de la comunicación SPI.



En el diagrama de bloques de la figura 17 aparecen representados los periféricos disponibles en la tarjeta Virtex-5 ML507 incluyendo los que se utilizan en este proyecto, descritos anteriormente.

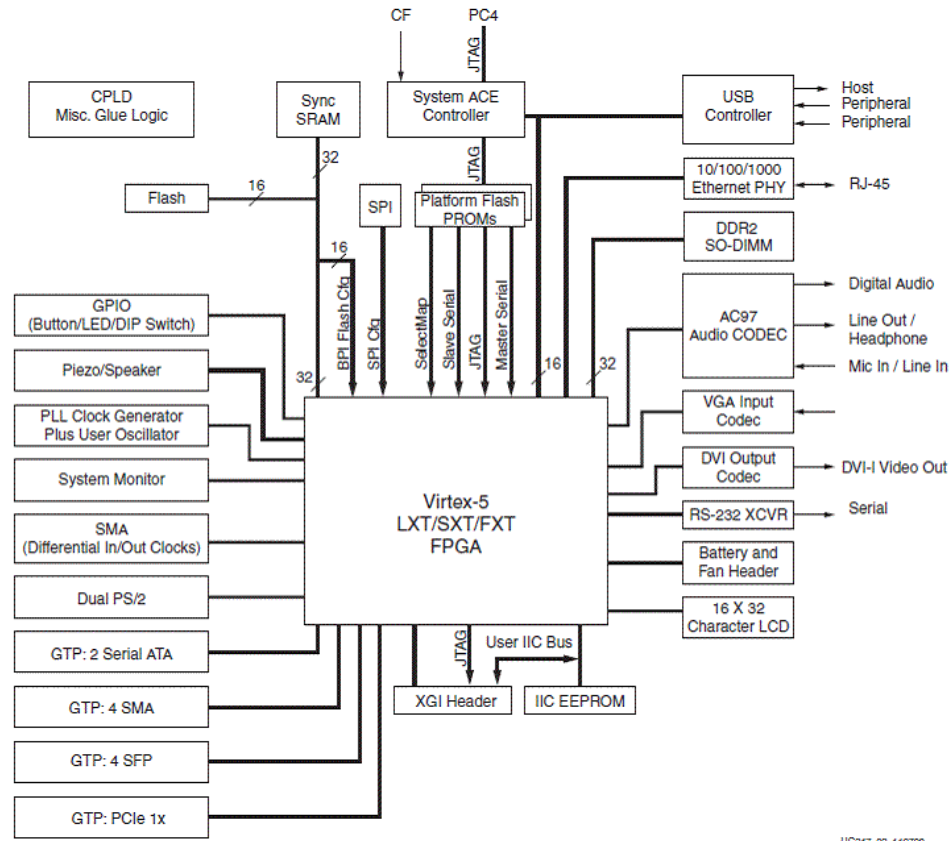


Figura 17. Diagrama de bloques de la plataforma de evaluación FPGA Virtex-5 ML507.

Xilinx proporciona un entorno de desarrollo llamado Embedded Development Kit (EDK) formado por dos herramientas muy potentes, una es la plataforma destinada a la implementación hardware de la FPGA, llamada Xilinx Platform Studio (XPS), que incluye los IP Cores facilitados por el fabricante, así como todo el entorno necesario para realizar la configuración de los dispositivos que se vayan a utilizar. Y la otra, el Software Development Kit (SDK) destinado a la programación del software para el procesador del sistema, ya sea simulado (Microblaze) o embebido (PowerPC). Este entorno se describe en más detalle en el capítulo 8



# 6 Arquitectura del sistema completo

---

## 6.1 Introducción



Figura 18. Diagrama de bloques global

En este capítulo se presentan los bloques que componen el sistema final para poder tener una primera aproximación del funcionamiento global. Se describen los tres bloques en los que se ha dividido el mismo. Cada bloque atiende a la naturaleza de los elementos que lo componen, simplificando así su estudio y presentación. En primer lugar se tiene el sensor de temperatura y su conversión a digital para la posterior comunicación, dichos elementos componen el primer bloque, en este documento también se hace referencia a dicho bloque como “placa acondicionadora de señal”. A continuación se tiene la gestión de las comunicaciones y el procesado de la temperatura medida, de esto se encarga la FPGA, en la que estará implementado el módulo de control de la comunicación SPI con la placa acondicionadora, el de comunicación serie con el PC así como el código para el procesado de temperaturas y gestión de comunicaciones. Por último, la parte de software, en el PC se implementa la interfaz de usuario y se realizan los cálculos de calibración. Todo el software de programación y configuración de la FPGA se realiza desde el PC tal como se explica en el capítulo 8.

En los capítulos anteriores se ha presentado el estudio realizado sobre el funcionamiento del sensor de temperatura, del conversor de termopar a digital y de la FPGA. En los siguientes capítulos se describen los diferentes bloques que se presentan en este capítulo, profundizando más en el diseño y el funcionamiento de cada bloque.

## 6.2 Visión general del sistema completo

El sistema final, tal como se observa en la figura 19, está compuesto por los bloques descritos en el apartado anterior, figura 18, que son la placa acondicionadora que compone el bloque I, y que incluye el hardware necesario para la conversión analógico-digital. La FPGA tiene dos funciones principales, la primera es hacer de puente entre la placa y el PC, a su entrada controla la comunicación SPI con la placa acondicionadora y a su salida la comunicación serie con el PC, y la segunda es realizar la calibración y procesamiento de la trama SPI. Recibe la trama que contiene las medidas realizadas por el MAX31855, aplica la calibración a dichas medidas y conforma una nueva trama que envía por puerto serie al PC. El tercer y último bloque, a la derecha en la imagen, está formado por el PC con las aplicaciones necesarias para la programación de la FPGA con el paquete de desarrollo software EDK de Xilinx.

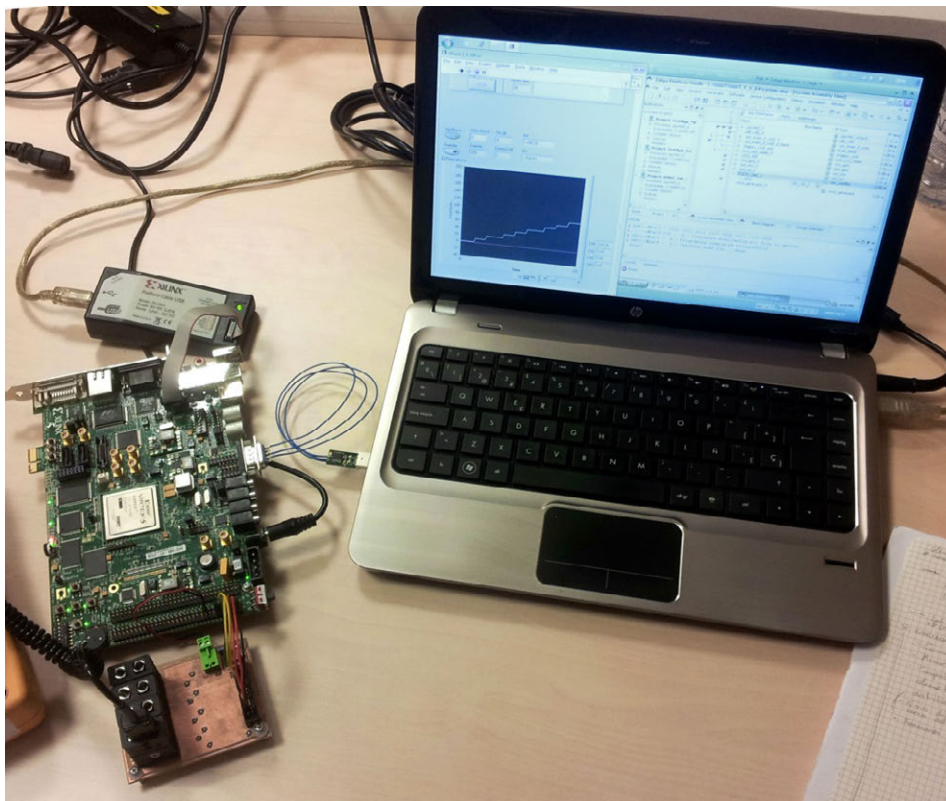


Figura 19. Sistema completo.

El diseño e implementación del primer bloque se presenta en más detalle en el capítulo 7 de esta memoria, se explica el proceso de diseño del esquemático y el posterior diseño de la placa acondicionadora así como el proceso de fabricación de la misma.

El capítulo 8 está dedicado al segundo bloque, se explica el funcionamiento de la FPGA. Se presentan los bloques disponibles en la placa de desarrollo Virtex-5, profundizando en los bloques empleados en este proyecto, explicando la configuración de cada bloque, las restricciones que debe cumplir el diseño hardware y el diagrama de flujo del código de programación en lenguaje C para el procesador, que realizará el ajuste de las medidas de temperatura y gestionará las comunicaciones con el resto de elementos del sistema. En dicho capítulo se explican también las herramientas de desarrollo y programación de la plataforma Virtex-5.

En el capítulo 9 se engloba el software de calibración, mediante hoja de cálculo de Excel y el software de la aplicación de interfaz de usuario generada en LabVIEW, en dos versiones, una destinada al usuario para la visualización de temperaturas y otra que incluye opciones de medida para el calibrado.



## 7 . Bloque I: Conversión

---

### 7.1 Introducción

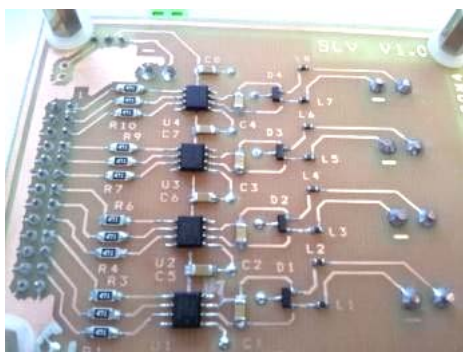


Figura 20. Placa acondicionadora de señal (Bottom)

En los capítulos 3 y 4 se presenta un estudio de las características de los elementos principales de este bloque.

A la entrada se encuentran los conectores para los cuatro sensores de temperatura, estos conectores son específicos para el tipo de termopar utilizado.

La conversión de termopar a digital se realiza mediante el uso del MAX31855, éste a su vez proporciona la salida del

bloque de conversión en modo SPI para la comunicación con la FPGA.

La trama digital que se envía contiene información sobre el canal leído, la temperatura de la unión fría, la temperatura externa medida por el termopar de dicho canal, así como el estado de la conexión del termopar. Todos estos aspectos se detallan a continuación.

## 7.2 Diagrama de bloques

En la figura 21 se ve el diagrama correspondiente a este bloque, formado por los conectores de termopar, el bloque de protecciones que incluye las protecciones necesarias para la conexión del termopar con el convertidor, en azul se muestra el bloque de convertidores MAX y sus líneas de comunicación SPI.

Más adelante, en este capítulo, se describe el circuito de la placa acondicionadora, que contiene los elementos del bloque de conversión. Se detalla el esquemático parte por parte, así como el diseño de la placa PCB, el esquemático se realiza mediante la herramienta OrCAD Capture y el diseño se realiza con la herramienta OrCAD Layout. La placa se ha fabricado en un laboratorio por el método de serigrafía, insolado, revelado y atacado mediante ácidos siguiendo el proceso que se explica en el apartado 7.4.1.

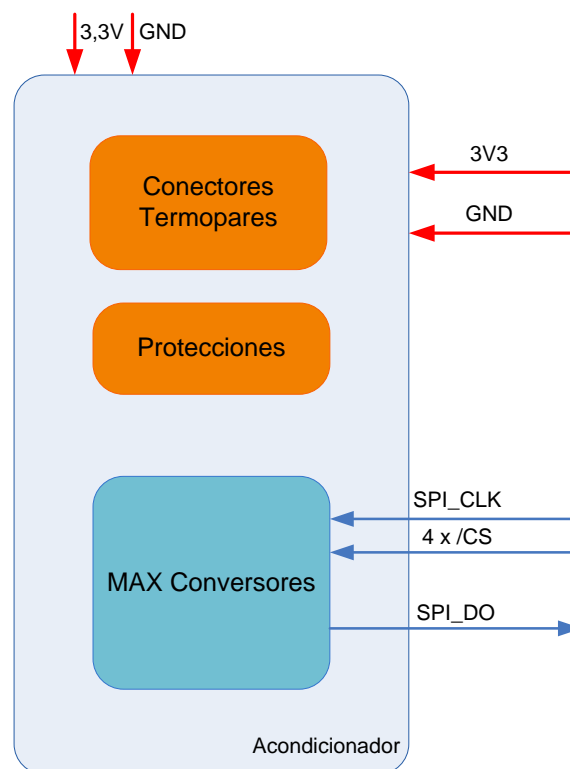


Figura 21. Bloque de conversión.



## 7.3 Esquemático

### 7.3.1 Alimentación

Los termopares no requieren de un circuito de alimentación específico, por tanto, la alimentación a tener en cuenta es la que necesita el MAX para su correcto funcionamiento. En las hojas de características del componente, el fabricante recomienda una alimentación típica de 3,3V. El circuito ha sido diseñado con dos posibles entradas de alimentación: una directa, desde los pines dedicados de alimentación digital a 3,3V de la FPGA y otra más indicada para alimentar desde una fuente de alimentación externa. Dispone de un jumper de selección con dos posiciones según la alimentación empleada.

En la figura 22 se pueden ver los dos conectores de alimentación comentados, el jumper de selección para las dos alimentaciones posibles y cuatro condensadores de desacoplo, que se deberán colocar cada uno lo más cerca posible a la entrada de alimentación de cada circuito integrado, es decir, de cada MAX, por lo que es necesarios al menos un condensador de desacoplo para cada canal de medida.

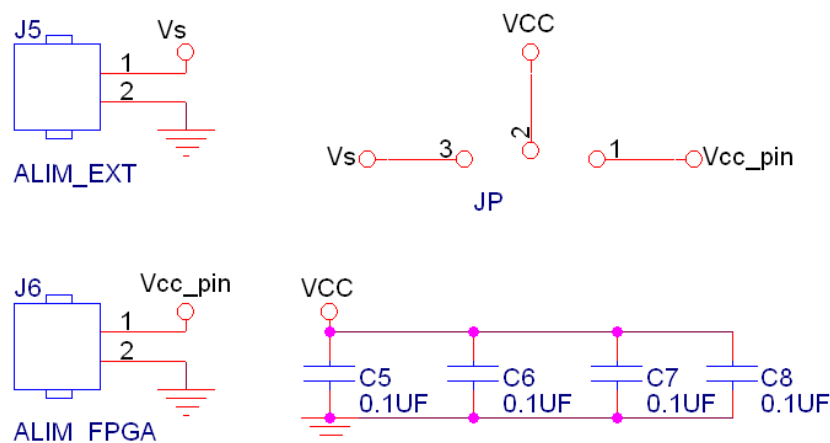


Figura 22. Conectores de alimentación, jumper de selección y condensadores de desacoplo para los circuitos integrados.

### 7.3.2 Conectores de termopar y protecciones

Los termopares requieren un conector de termopar estándar, éste a su vez va conectado a los terminales T+ y T- del convertor. La entrada al convertor debe ser protegida tal como se ve en la figura 23. Como supresor de ruido de alta frecuencia se

emplean dos ferritas, L1 y L2, colocadas cada una en una pista de conexión entre el conector de termopar y el MAX. Entre ambas conexiones se incorpora un diodo supresor de tensión, D1, en concreto el NUP2105, este diodo protege la conexión de forma bidireccional, mejorando el comportamiento del sistema frente a posible radiación electromagnética (EMI). Finalmente se conecta un condensador cerámico, C1, que funciona como condensador de desacoplo.

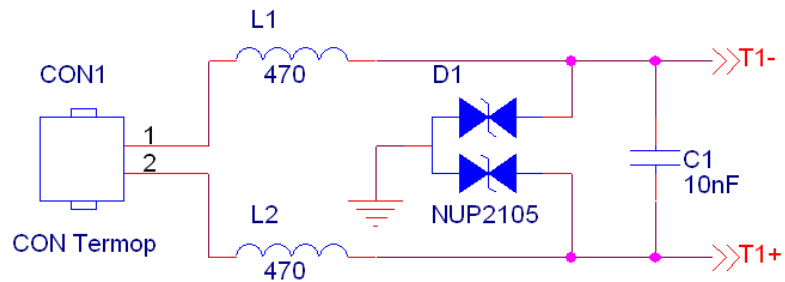


Figura 23. Conector de termopar y circuito de protección.

### 7.3.3 Circuito de conexión del MAX31855

En la figura 24 aparece el circuito del MAX31855. Tiene como entrada la tensión diferencial de T+ y T- proveniente del termopar una vez protegida, y como salida las tres líneas digitales para conexión SPI:

- SO es la señal de datos de salida hacia la FPGA.
- SCK es la señal de reloj a 5MHz tal como se aconseja en las hojas del fabricante.
- /CS es la señal de chip select a nivel bajo, para elegir de cuál de los 4 canales se hará la lectura.

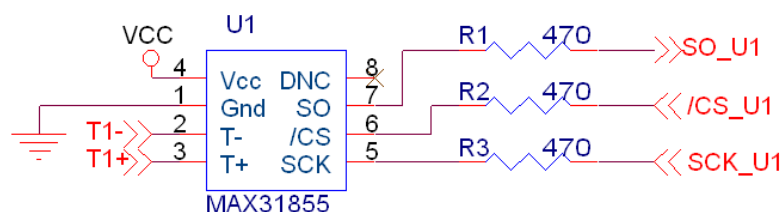


Figura 24. Circuito de conexión del MAX31855.

## 7.4 Diseño

### 7.4.1 Diseño PCB

Una vez cerrado el esquemático, se procede al diseño del PCB. Este diseño ha sido realizado con OrCAD Layout. Se ha optado por un diseño híbrido de doble cara, con componentes tanto de montaje superficial como de inserción, haciendo que todas las soldaduras se encuentren en la capa bottom, de modo que los termopares se conectarán por la capa superior o top, donde se tendrá acceso también a los conectores de alimentación y comunicación. Una vez decidida la mecánica del prototipo (es decir, la orientación de los componentes críticos como conectores de entrada, salida y alimentación), se procede al posicionamiento de todos los componentes, verificando que las huellas se correspondan con el encapsulado de los componentes que se van a emplear. Tras esto se procede al rutado manual del PCB.

En la figura 25 se presenta el estado final del diseño del PCB, se muestran ambas capas tal como aparecen en la ventana de diseño. Se puede apreciar que la disposición de los componentes de los cuatro canales coincide, el diseño se ha realizado de este modo para evitar en la medida de lo posible que el hardware pueda introducir diferencias en el comportamiento de los canales debidas a diferencias en la disposición de los componentes.

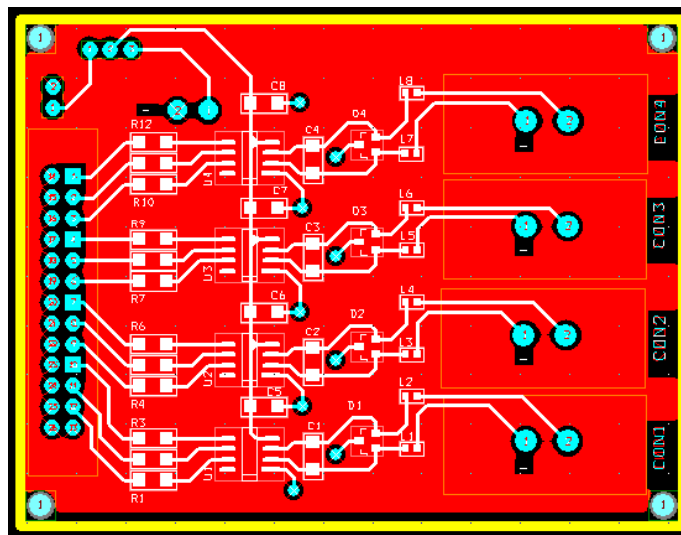


Figura 25. Diseño PCB en OrCAD Layout.

### 7.4.1 Fabricación del PCB

La tarjeta acondicionadora de señal se ha realizado de forma manual en una placa fotosensible de doble cara, los elementos necesarios para llevar a cabo el proceso son los siguientes:

- Placa fotosensible de doble cara.
- Fitolito con el diseño impreso en tinta negra lo más opaco posible.
- Insoladora.
- Líquido de revelado (Sosa cáustica y agua a partes iguales).
- Líquido de atacado (Cloruro férrico y agua).

El primer paso para realizar el diseño descrito en el apartado anterior, ha sido imprimir el diseño en un fitolito que hace las veces de negativo al a hora de insolar la placa, es decir, aplicar luz en ambas capas de la placa para que su barniz fotosensible se sensibilice en las zonas que no quedan protegidas por la tinta del fitolito. La insoladora utilizada se puede observar en la figura 26.



Figura 26. Insoladora donde se colocará la placa con el fitolito.

A continuación, la placa ya sensibilizada se debe introducir en el líquido de revelado, de modo que el barniz sensibilizado queda eliminado y se descubrirá el dibujo del diseño, este paso finaliza introduciendo la placa en agua. En algunos casos es necesario repasar la placa con cuidado, empleando un pincel o con un dedo para

eliminar posibles restos de barniz fotosensible. Una vez hecho esto, como tercer y último paso, se introduce la placa en el líquido de atacado, de modo que el cobre no protegido queda eliminado, dando como resultado la placa de fibra de vidrio (FR4) con las pistas y huellas de los componentes en cobre. Los líquidos se encuentran en una cubeta como la mostrada en la figura 27, dispone de tres espacios, correspondientes al líquido de revelado, agua y líquido de atacado. Este tipo de cubetas tienen calentadores para los líquidos y circuito generador de burbujas para acelerar el proceso de fabricación.



*Figura 27. Equipo de revelado y atacado con calentadores para los líquidos y generador de burbujas que aceleran el proceso de fabricación.*

#### **7.4.2 Montaje de los componentes**

Una vez fabricado el PCB, se han realizado los taladros para los componentes de inserción, utilizando el diámetro de broca adecuado para cada caso según el tamaño del pin del componente. Y por último, se ha procedido a soldar todos los componentes también por el método manual, utilizando una estación de soldadura con soldador de punta fina, en la figura 28 se muestra un detalle de la soldadura de los componentes de montaje superficial de la placa. El resultado de la placa acondicionadora una vez finalizado el montaje se puede ver en la figura 29.

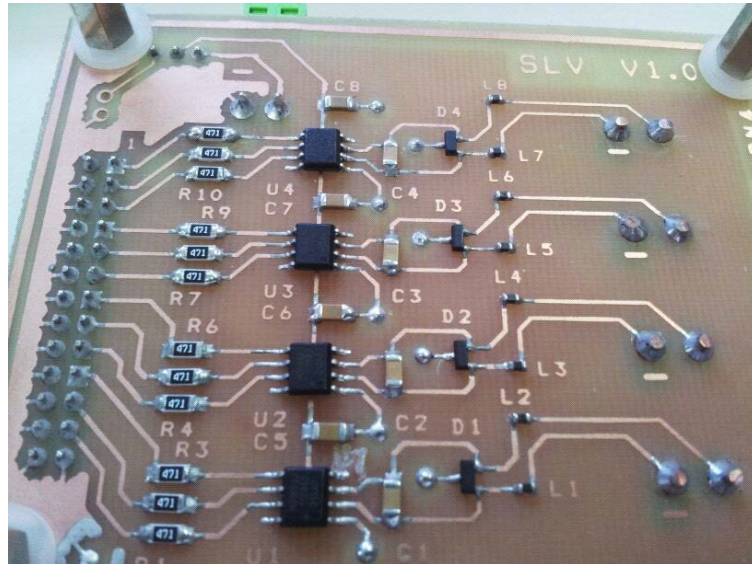


Figura 28. Placa acondicionadora finalizada. Detalle de soldaduras.

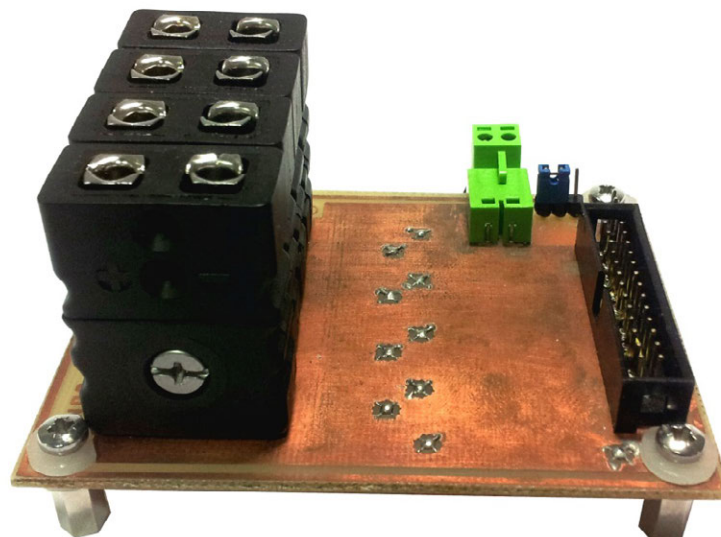


Figura 29. Placa acondicionadora finalizada..

### 7.4.3 Costes

El coste del prototipo es en su mayor parte debido a los componentes empleados, a continuación se muestra la tabla de costes de componentes, tabla 8. En los apéndices se incluye la lista de referencias de componentes MPL (Master Part List) completa con referencia del fabricante, distribuidor, referencia del distribuidor y precio.

Cantidad	Descripción	Precio	Total
1	Conector 32 pin	3,00 €	3,00 €
1	Conector alimentación 2 pines extraíble	1,00 €	1,00 €
1	Jumper 2 pin	0,30 €	0,30 €
1	Jumper 3 pin	0,30 €	0,30 €
4	Conector termopar tipo J	3,13 €	12,52 €
8	Ferrite bead SMD 0603 470	0,30 €	2,40 €
12	Resistencia SMD 0805 470R 1/4W	0,02 €	0,26 €
4	Convertor de termopar a digital MAX31855JASA+	5,68 €	22,72 €
4	Condensador SMD 0805 0,1uF	0,05 €	0,19 €
1	Condensador SMD 0805 10nF	0,16 €	0,16 €
4	Diodo NUP2105 SOT-23	0,43 €	1,72 €
			<b>44,58 €</b>

Tabla 8. Costes de fabricación del PCB.





## 8 . Bloque II: Procesado digital

---

### 8.1 Introducción



Figura 30. FPGA Virtex 5.

El procesado digital de este proyecto se realiza mediante una plataforma de desarrollo de Xilinx, en concreto la tarjeta de desarrollo basada en FPGA modelo Virtex-5, en su versión ML507. Este kit incluye la FPGA modelo XC5VFX70T-2FFG1136C.

Para la programación y configuración de la tarjeta, es necesario emplear una serie de herramientas de desarrollo específicas que son facilitadas por el fabricante.

El “Embedded Design Kit” (EDK), se trata de una plataforma de desarrollo de sistemas embebidos muy potente. Se utilizan dos programas. El primero, llamado “Xilinx Platform Studio” (XPS) permite programar la FPGA mediante los “Intellectual Property Cores” (IPCoers) que son bloques de código VHDL, estos bloques pueden ser desarrollados bien por el fabricante o por el usuario. Se eligen según la configuración de los periféricos del kit de desarrollo que se vaya a utilizar. En segundo lugar se dispone de la herramienta de desarrollo software llamada “Software Development Kit” (SDK), desde donde se programa el procesador PowerPC440.

## 8.2 Periféricos utilizados

A continuación, en la figura 31, se presenta una breve descripción de los bloques utilizados en este proyecto, que se desarrollará en mayor profundidad a lo largo del capítulo. En primer lugar, se puede observar el bloque del conector J5, que contiene los pines de salida dedicados a alimentación de la placa Virtes-5, se destinan dos de estos pines a la alimentación de la placa acondicionadora, uno para 3,3V y el otro para masa. El conector J6 tiene pines de entrada salida de propósito general (GPIO), se utilizan seis pines para la comunicación SPI con la placa acondicionadora, cuatro de ellos destinados a la señal de selección del canal (chip select), otro para la señal de reloj y el último para la señal de datos de entrada a la FPGA. La comunicación con el PC se realiza mediante el puerto serie RS-232 que deberá tener la misma configuración en origen y destino. Y finalmente se puede ver el bloque correspondiente al PowerPC que se programa en lenguaje C mediante el conector de programación.

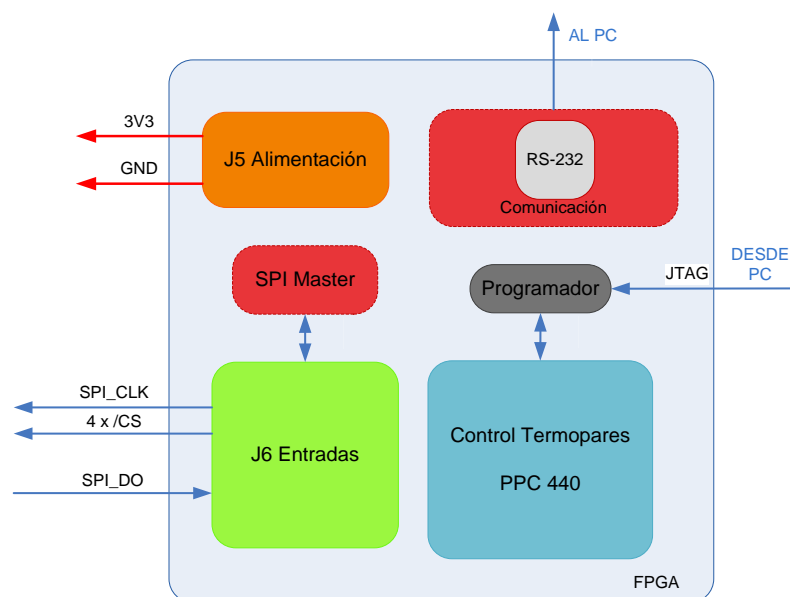


Figura 31. Diagrama de bloques implementados en la plataforma de desarrollo Virtex-5

## 8.3 Configuración de la FPGA: XPS

Para implementar tanto la comunicación SPI entre la placa acondicionadora y la FPGA como la comunicación serie (RS-232) entre FPGA y PC se necesitan los bloques diseñados en lenguaje hardware HDL (Hardware Design Language)

correspondientes a dichas comunicaciones. En este proyecto se han utilizado los bloques de propiedad intelectual o IP Cores (Intellectual Property Core) facilitados por el fabricante.

Para el correcto funcionamiento del sistema es necesario declarar algunas restricciones, así como asignar los pines de la FPGA que se quieren utilizar para determinadas señales. Se deben utilizar los drivers adecuados, también facilitados por el fabricante, junto con sus correspondientes hojas de características y funciones de programación para el manejo del driver.

### 8.3.1 Bloques VHDL y drivers implementados en la plataforma XPS

A continuación se presenta una breve descripción de los bloques IP Core utilizados en el diseño, en la figura 32 se pueden ver los IP Cores utilizados y su bus de interfaces, tal como aparece en el "System Assembly View" del programa XPS. A continuación se describe brevemente la funcionalidad de cada bloque empleado.

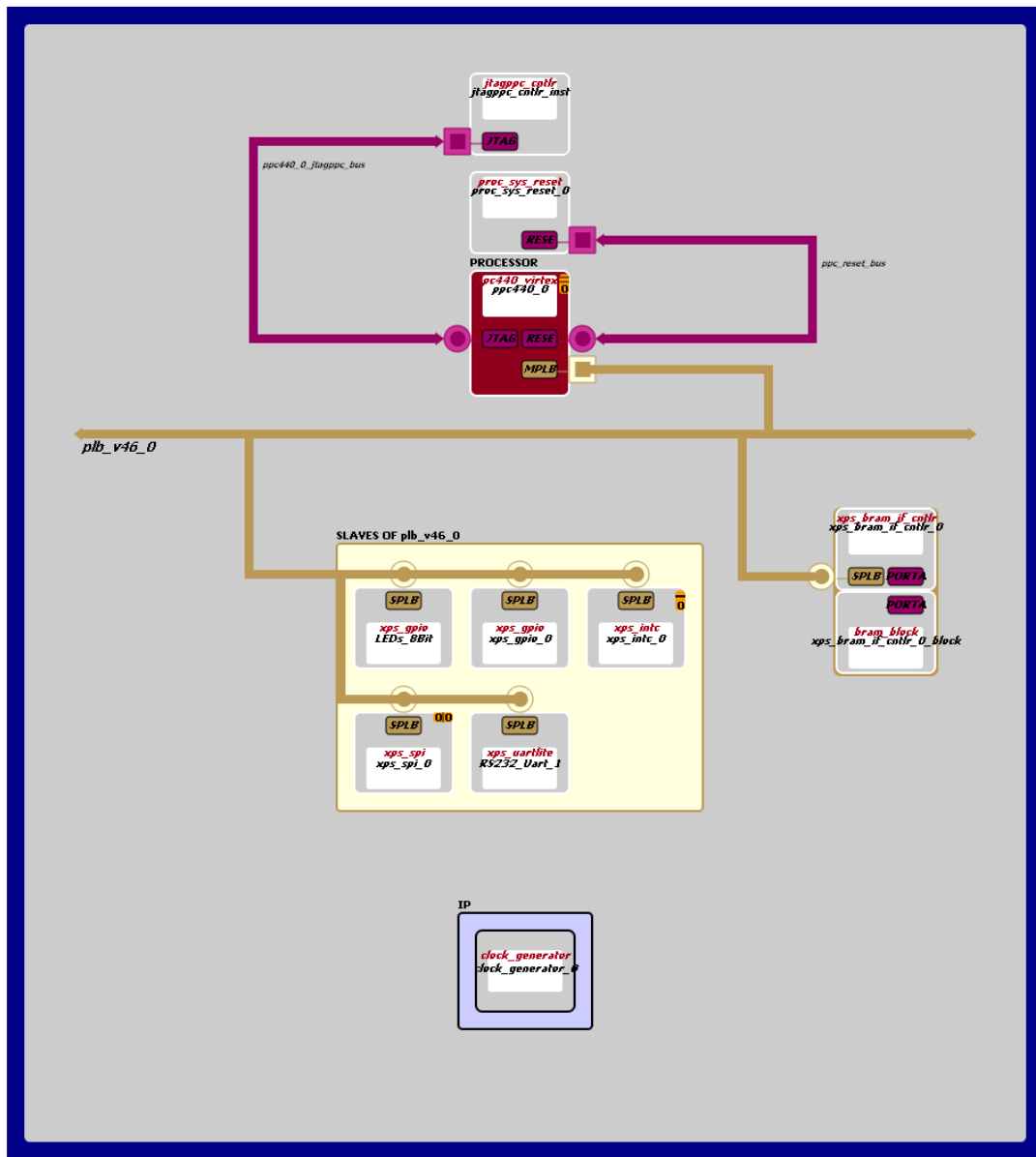
- **Ppc440\_0:** Es el IP Core correspondiente al procesador embebido en la FPGA, se utiliza en su versión 1.01.a
- **P1b\_v46:** Correspondiente al bus principal del sistema, al que se conectarán todos los periféricos que actúen como maestros o esclavos en el diseño. La versión utilizada es la 1.04.a.
- **Xps\_bram\_if\_cntrl\_0:** Es el bloque controlador de la memoria BRAM de la FPGA, donde se almacena el código del programa que ejecuta el procesador. Aparece en su versión 1.00.b
- **Jtagpc\_cntrl\_inst:** Utilizado en su versión 2.01.c este IP Core permite el control de la conexión JTAG con el programador de Xilinx, para poder configurar y programar la FPGA.
- **Xps\_spi\_0:** Serial Peripheral Interface. Se ha utilizado el Driver en su versión IP 2.00.a de Xilinx. Ya que permite modo sondeo, es decir, modo de uso sin interrupciones, que no son necesarias para el tipo de comunicación de este proyecto, puesto que se lee de un canal cada vez, sin realizar ninguna otra función al mismo tiempo. Para poder sincronizar esta comunicación, se declara una restricción en el archivo de restricciones del usuario UCF (User Constraints File), de modo que el diseño interno asegure un reloj de 5MHz, tal como sugiere el fabricante

- **Xps\_gpio\_0:** GPIO se refiere a los pines de entrada/salida de propósito general, se utilizan para hacer la conexión SPI con la placa acondicionadora. La versión IP es la 2.00.a
- **RS232\_Uart\_1:** Bloque para conexión por puerto serie con el PC, se utiliza la versión 1.01.a
- **Clock\_generator\_0:** Este bloque está programado para generar el reloj del sistema.

Name	Bus Name	IP Type	IP Version	IP Classification
ppc440_0		★ ppc440_virtex5	1.01.a	Processor
MPLB	plb_v46_0			
SPLB0	No Connection			
SPLB1	No Connection			
PPC440MC	ppc440_0_PPC440MC			
MDCR	No Connection			
SDCR	No Connection			
MFCB	No Connection			
MFCM	ppc440_0_MFCM			
JTAGPPC	ppc440_0_jtagppc_bus			
RESETPPC	ppc_reset_bus			
plb_v46_0		★ plb_v46	1.04.a	PLBV46 Bus
xps_bram_if_cntlr_0	plb_v46_0	★ xps_bram_if_cntlr	1.00.b	Memory Controller
SPLB	plb_v46_0			
PORTA	xps_bram_if_cntlr_0_PORTA			
xps_bram_if_cntlr_0_block		★ bram_block	1.00.a	Memory
PORTA	xps_bram_if_cntlr_0_port			
PORTB	No Connection			
jtagppc_cntlr_inst	jtagppc_cntlr_inst_JTAGPPC0	★ jtagppc_cntlr	2.01.c	Peripheral
proc_sys_reset_0		★ proc_sys_reset	2.00.a	Peripheral
RESETPPC0	proc_sys_reset_0_RESETPPC0			
LEDS_BBK		★ xps_gpio	2.00.a	Peripheral
SPLB	plb_v46_0			
xps_gpio_0		★ xps_gpio	2.00.a	Peripheral
SPLB	plb_v46_0			
xps_spi_0		★ xps_spi	2.01.b	Peripheral
SPLB	plb_v46_0			
RS232_Uart_1		★ xps_uartlite	1.01.a	Peripheral
SPLB	plb_v46_0			
clock_generator_0		★ clock_generator	3.02.a	IP

Figura 32. Bus de interfaces implementadas en la plataforma de desarrollo XPS

El diagrama de bloques de la configuración realizada se puede observar en la figura 33. Se muestran en morado las conexiones P2P (peer to peer), es decir, las conexiones internas entre bloques. La de la izquierda corresponde al bus de programación del procesador, el bloque de programación **Jtagpc\_cntlr\_inst** recibe las señales desde el conector de programación de la placa Virtex-5 y actúa como maestro en la programación del procesador mediante el bus **ppc\_440\_jtag\_ppc\_bus**. A la derecha, también en morado, se tiene el bus de reset llamado **ppc\_reset\_bus**, el control de este bus se realiza mediante el bloque **proc\_sys\_reset\_0** que actúa como maestro, y al otro extremo, como esclavo se conecta el procesador. En color marrón, se representa el Bus PLB o Processor Local Bus (Bus Local del Procesador), el procesador es el maestro de este bus, y tiene como esclavos, por una parte, todos los IP de los dispositivos que se utilizan, y por otra el IP de la memoria BRAM.



SPECS	
EDK VERSION	11.5
ARCH	virtex5
PART	xc5vfx70ff1136-1
GENERATED	Tue Apr 23 20:28:20 2013

KEY			
SYMBOLS			
	<b>Bus connections</b>	<b>External Ports</b>	<b>Interrupts</b>
			<small>X = Controller ID Y = Interrupt Priority</small>
COLORS			

Figura 33. Diagrama de bloques de los IPCores implementados mediante la plataforma de desarrollo XPS.

A continuación se puede ver el resumen del uso de recursos de la FPGA realizado con la configuración presentada anteriormente, en la tabla 9 aparece el nombre del recurso, los elementos utilizados y el porcentaje de uso respecto al total disponible.

Device Utilization Summary				<a href="#">[-]</a>
Slice Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Registers	1,101	44,800	2%	
Number used as Flip Flops	1,101			
Number of Slice LUTs	942	44,800	2%	
Number used as logic	902	44,800	2%	
Number using O6 output only	874			
Number using O5 output only	4			
Number using O5 and O6	24			
Number used as Memory	37	13,120	1%	
Number used as Shift Register	37			
Number using O6 output only	37			
Number used as exclusive route-thru	3			
Number of route-thrus	10			
Number using O6 output only	7			
Number using O5 output only	3			
Number of occupied Slices	834	11,200	7%	
Number of LUT Flip Flop pairs used	1,606			
Number with an unused Flip Flop	505	1,606	31%	
Number with an unused LUT	664	1,606	41%	
Number of fully used LUT-FF pairs	437	1,606	27%	
Number of unique control sets	194			
Number of slice register sites lost to control set restrictions	434	44,800	1%	
Number of bonded <a href="#">IOBs</a>	27	640	4%	
Number of LOCed IOBs	17	27	62%	
IOB Flip Flops	24			
Number of BlockRAM/FIFO	32	148	21%	
Number using BlockRAM only	32			
Number of 36k BlockRAM used	32			
<b>Total Memory used (KB)</b>	<b>1,152</b>	<b>5,328</b>	<b>21%</b>	
Number of BUFG/BUFGCTRLs	4	32	12%	
Number used as BUFGs	4			
Number of PLL_ADVs	1	6	16%	
Number of PPC440s	1	1	100%	
Average Fanout of Non-Clock Nets	3.82			

Tabla 9. Resumen de recursos utilizados en la FPGA.

### 8.3.1 Restricciones del sistema y pines de entrada/salida

Para asegurar el correcto funcionamiento de los bloques implementados en la FPGA es necesario describir algunas restricciones mediante las que se indican a la FPGA los requisitos que deben cumplir las señales más críticas. Para ello se utiliza el archivo \*.UCF en la plataforma de desarrollo XPS.

El archivo \*.UCF es el que contiene todas las restricciones (Constraints) y conexiones declaradas por el usuario (Nets), es un archivo de programación con un lenguaje específico en el que se declaran los requisitos que se deben tener en cuenta para la configuración interna de la FPGA.

Para cada Net declarada se debe indicar la conexión interna con la FPGA. Para elegir los pines correctamente se debe consultar la documentación del fabricante, en este caso la guía de usuario de Xilinx "ML505/ML507 Evaluation Platform User Guide 347". Este archivo describe todos los dispositivos disponibles en el kit de evaluación y contiene las tablas de correspondencia entre los periféricos y los pines de la FPGA.

A continuación se muestran las restricciones y nets que ha sido necesario añadir al archivo de restricciones de usuario. Se indica la configuración de pines de la FPGA realizada para cada periférico utilizado, mostrando también, si procede, un extracto de la tabla correspondiente al documento de la guía de usuario de Xilinx.

#### LEDs

Para el uso de los LEDs se ha debido configurar la localización de los pines internos de la FPGA correspondientes a los LEDs de la tarjeta Virtex-5. Dicha localización se encuentra en la tabla 10, extraída de la guía de usuario.

Reference Designator	Label/Definition	Color	FPGA Pin	Buffered
DS17	GPIO LED 0	Green	H18	Yes
DS16	GPIO LED 1	Green	L18	Yes
DS15	GPIO LED 2	Green	G15	Yes
DS14	GPIO LED 3	Green	AD26	No
DS13	GPIO LED 4	Green	G16	Yes
DS12	GPIO LED 5	Green	AD25	No
DS11	GPIO LED 6	Green	AD24	No
DS10	GPIO LED 7	Green	AE24	No

Tabla 10. Localización de los LEDs en la FPGA.

Las líneas a añadir en el archivo de restricciones son las siguientes:

```
Net fpga_0_LEDs_8Bit_GPIO_IO_pin<0> LOC = AE24 | IOSTANDARD=LVCMOS18 | PULLDOWN |
  SLEW=SLOW | DRIVE=2;
Net fpga_0_LEDs_8Bit_GPIO_IO_pin<1> LOC = AD24 | IOSTANDARD=LVCMOS18 | PULLDOWN |
  SLEW=SLOW | DRIVE=2;
Net fpga_0_LEDs_8Bit_GPIO_IO_pin<2> LOC = AD25 | IOSTANDARD=LVCMOS18 | PULLDOWN |
  SLEW=SLOW | DRIVE=2;
Net fpga_0_LEDs_8Bit_GPIO_IO_pin<3> LOC = G16 | IOSTANDARD=LVCMOS25 | PULLDOWN |
  SLEW=SLOW | DRIVE=2;
Net fpga_0_LEDs_8Bit_GPIO_IO_pin<4> LOC = AD26 | IOSTANDARD=LVCMOS18 | PULLDOWN |
  SLEW=SLOW | DRIVE=2;
Net fpga_0_LEDs_8Bit_GPIO_IO_pin<5> LOC = G15 | IOSTANDARD=LVCMOS25 | PULLDOWN |
  SLEW=SLOW | DRIVE=2;
Net fpga_0_LEDs_8Bit_GPIO_IO_pin<6> LOC = L18 | IOSTANDARD=LVCMOS25 | PULLDOWN |
  SLEW=SLOW | DRIVE=2;
Net fpga_0_LEDs_8Bit_GPIO_IO_pin<7> LOC = H18 | IOSTANDARD=LVCMOS25 | PULLDOWN |
  SLEW=SLOW | DRIVE=2;
```

### Comunicación SPI

Los pines de entrada/salida destinados a la comunicación SPI con la placa acondicionadora se encuentran en el conector de expansión J6 de la tarjeta Virtex-5, en la tabla 11 se puede ver la correspondencia interna con los pines de la FPGA que se han debido declarar en el archivo de restricciones.

J6 Pin	Schematic Net Name	FPGA Pin
2	HDR1_2	H33
4	HDR1_4	F34
6	HDR1_6	H34
8	HDR1_8	G33
10	HDR1_10	G32
12	HDR1_12	H32
14	HDR1_14	J32
16	HDR1_16	J34

Tabla 11. Localización del conector de expansión J6 en la FPGA.

Las líneas a añadir en el archivo de restricciones son las siguientes:

- Los pines de reloj y datos:

```
Net xps_spi_0_MISO LOC = H33 | IOSTANDARD=LVCMOS33; #pinJ6 2 S0
Net xps_spi_0_SCK LOC = H32 | IOSTANDARD=LVCMOS33; #pinJ6 12 CLK
```

- Los pines de chip select:

```
Net xps_spi_0_SS_0_pin<0> LOC = F34 | IOSTANDARD=LVCMOS33; #pinJ6 4 /CS1
Net xps_spi_0_SS_0_pin<1> LOC = H34 | IOSTANDARD=LVCMOS33; #pinJ6 6 /CS2
Net xps_spi_0_SS_0_pin<2> LOC = G33 | IOSTANDARD=LVCMOS33; #pinJ6 8 /CS3
Net xps_spi_0_SS_0_pin<3> LOC = G32 | IOSTANDARD=LVCMOS33; #pinJ6 10 /CS4
```



Para configurar la frecuencia de reloj de la comunicación SPI se ha consultado el documento de especificaciones del driver empleado, en él se indica que se debe añadir la siguiente línea de restricciones de tiempo al archivo de restricciones, con lo que queda especificado el tiempo entre flancos de los flip-flop. Con ésto el diseño interno de la FPGA asegura que la conexión con el pin del conector J6 correspondiente al reloj SCK tendrá una frecuencia de 5MHz, tal como indica el fabricante del MAX31855.

```
NET "clk_100_0000MHzPLL0_ADJUST" TNM_NET = "splb_clk";  
TIMESPEC "TS_splb_clk" = PERIOD "splb_clk" 200 ns HIGH 50%; #reloj MAX31855 max 5MHz
```

## 8.4 Programación del procesador: SDK

El procesador PowerPC440 funciona en modo standalone, es decir, no utiliza ningún sistema operativo, sino que se programa en lenguaje C. El procesador controla la comunicación de entrada/salida implementada en la FPGA, así mismo, se encarga de realizar el ajuste de las medidas que recibe de la placa acondicionadora mediante la comunicación SPI, y da formato a la trama de salida que envía por puerto serie. Cada IPCore implementado en la FPGA tiene sus propias funciones declaradas en los archivos \*.h correspondientes. Dichos archivos se deben incluir en el código, para controlar la comunicación SPI se incluyen "xspi.h" y "xspi\_l.h" y para la comunicación serie se incluyen "xuartlite.h" y "xuartlite\_l.h". Se debe tener en cuenta la versión del driver empleado, puesto que, tanto las opciones o modos de comportamiento como las funciones incluidas pueden variar mucho de una versión a otra.

La figura 34 se representa el diagrama de flujo que resume el código principal ejecutado por el procesador. Éste inicializa el sistema, para continuar con el bucle de lectura del canal SPI correspondiente, si la medida es válida procederá a su ajuste para un posterior envío hacia el PC de la medida ajustada, si la medida no es válida, la trama enviada al PC incluirá el tipo de fallo del canal y fijará los valores de medida a un valor por defecto (0,00).

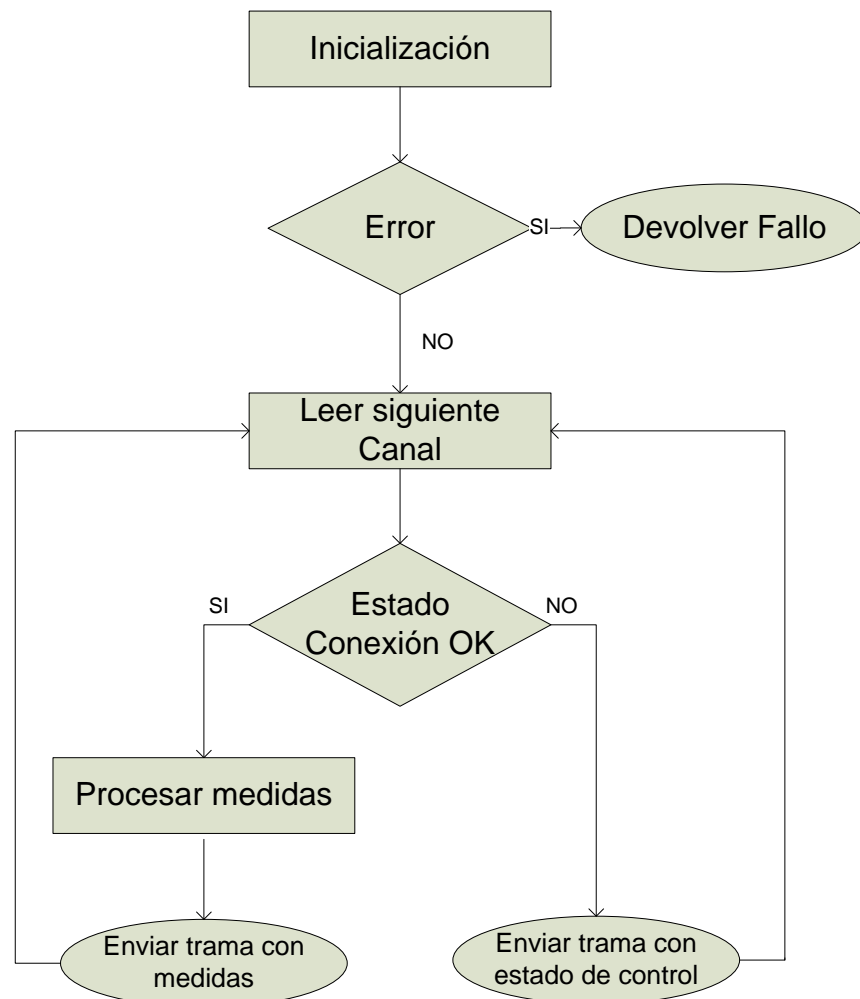


Figura 34. Diagrama de flujo del programa principal del procesador PowerPC440

En primer lugar se inicializan los dispositivos para la comunicación, mediante las funciones de inicialización contenidas en sus archivos \*.h. La comunicación serie con el PC se realiza mediante la UART que no requiere inicialización. Sin embargo, la comunicación SPI debe ser inicializada indicando el dispositivo que se va a utilizar.

Una vez inicializada la comunicación de forma correcta, se procede a la lectura de los canales de forma continua. Se lee el buffer SPI de entrada y se almacena. Tras la lectura, se comprueba el bit de estado del canal, para saber si hay o no un termopar conectado y que no está cortocircuitado a masa ni a alimentación. A cada uno de los posibles estados de conexión del termopar se le asigna un valor, y según se sabrá el tipo de incidencia. Se emplea el valor 0 para indicar que la conexión es correcta, valor 1 si no hay ningún termopar conectado, valor 2 si hay un cortocircuito a masa y valor 3 si se trata de un cortocircuito con la alimentación

Si la conexión es correcta, es decir el valor de estado es 0, se procesa la medida recibida aplicando el offset y la ganancia correspondientes al canal y al valor de medida. El valor ya ajustado se guarda en una estructura para posteriormente crear la trama a enviar. La estructura guarda los valores de signo, parte entera y parte decimal.

Si la conexión es incorrecta, se procede a generar una trama con estado de control, en la que se indica el número de incidencia y los valores de las medidas se fijan a un valor conocido (0,00).

Tal como se muestra en la figura 35, el formato de la trama que llega al PC por puerto serie contiene el canal medido, la temperatura de la unión, la temperatura del termopar y por último un delimitador de fin de trama. Los datos está separados por el delimitador '\t'. Los cuatro byte correspondientes a la temperatura contienen en primer lugar, el signo de la misma, a continuación la parte entera representada con tres dígitos y dos dígitos para el valor decimal. La trama completa tiene un tamaño de 19 bytes, éste será el tamaño a comprobar para que la trama sea considerada válida cuando se procesa en LabVIEW.

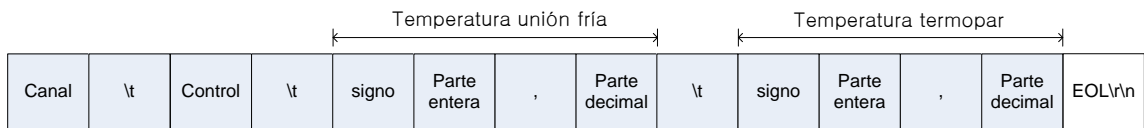


Figura 35. Trama de temperaturas para comunicación RS-232



## 9 . Bloque III: Procesado de datos en el PC

---

### 9.1 Introducción



Figura 36. Logo del programa LabVIEW de National Instruments.

Para realizar la aplicación de usuario en el PC se ha utilizado el entorno de desarrollo LabVIEW (acrónimo de Laboratory Virtual Instrumentation Engineering Workbench). Se trata de un entorno ideado para su empleo en ingeniería y ciencia. Utiliza un lenguaje de programación visual gráfico y una alta capacidad de integración de hardware, lo que la convierte en una herramienta muy útil y flexible a la hora de diseñar aplicaciones de sistemas de medida y

control. Dispone de multitud de bloques prediseñados, de modo que la programación de los Instrumentos Virtuales (VI), que es como se denomina cada programa desarrollado, resulta muy rápida. Cada VI consta de dos partes diferenciadas, una llamada Front Panel (Panel Frontal) en la que se compone la interfaz de usuario, donde se pueden incluir botones de interacción, displays, gráficas, imágenes... y la segunda parte, el Block Diagram (Diagrama de Bloques) incluye el entorno de programación donde se pueden encontrar todos los bloques funcionales aplicables, clasificados por categorías según su función, ya sea numérica, de programación, uso de strings, control de tiempos...

## 9.2 Captura y procesado de la trama mediante LabVIEW

Se ha programado una aplicación en LabVIEW con la que se pueden capturar las temperaturas medidas, en este caso se han realizado medidas con un incremento de 10°C en un intervalo de 0°C a 200°C. La trama recibida por puerto serie desde la FPGA se procesa mediante el programa de LabVIEW creado, y se guarda en un archivo compatible con Excel donde se realizarán los cálculos necesarios para la calibración. También se visualiza la gráfica de temperatura de los cuatro canales en tiempo real. La captura del dato se realiza mediante un botón de guardado, con esto se asegura que el dato sea estable y que solo se guarde un valor por medida.

A continuación se expone una explicación del funcionamiento del código de la interfaz de usuario en LabVIEW. Para proceder con la explicación, se ha dividido el código en bloques según su funcionalidad. El código completo se puede encontrar en los apéndices.

El primer bloque presentado en la figura 37 corresponde a la funcionalidad de lectura escritura del puerto serie. A la izquierda, se configuran las opciones del puerto serie según el baud rate, el número de bits, la paridad, el número de bits de parada y si hay o no control de flujo, esta configuración debe ser la misma que la del bloque de comunicación serie de la FPGA, dicha configuración se realiza dentro de una estructura de tipo secuencia "Flat Sequence Structure", este tipo de estructura no finaliza hasta que no se ha terminado de completar todo el código que contiene, y no comienza hasta que no llega información a todas sus entradas. Los siguientes bloques dispuestos en paralelo son, el de escritura (en la parte superior) que en este caso no se utiliza, pero queda para posibles aplicaciones futuras, y el inferior, que es el de lectura de puerto serie, en el comentario (en amarillo) se informa del formato de trama que se recibe, es decir, la trama a procesar por el resto del código. La lectura se realiza en un bucle "while" (cuadro gris sólido) cada 50ms mientras no se presione el botón de Stop. Mediante una estructura tipo "Case" se comprueba si hay bytes (en azul) disponibles para lectura, en caso de que haya, caso "True" tal como aparece en la imagen, se procede a leer del puerto, se concatena la lectura de los bytes leídos y se indexa al string (en rosa) de lectura, siempre que se hayan leído al menos 80 caracteres, asegurando así que el tamaño leído sea una trama completa. Por último, se tiene otra estructura tipo secuencia, que una vez finalizado todo el proceso, se encarga de la comprobación de errores de transmisión y recepción y compone la tabla con las información procesada, esta tabla se guarda en formato \*.csv de modo que es compatible con Excel.

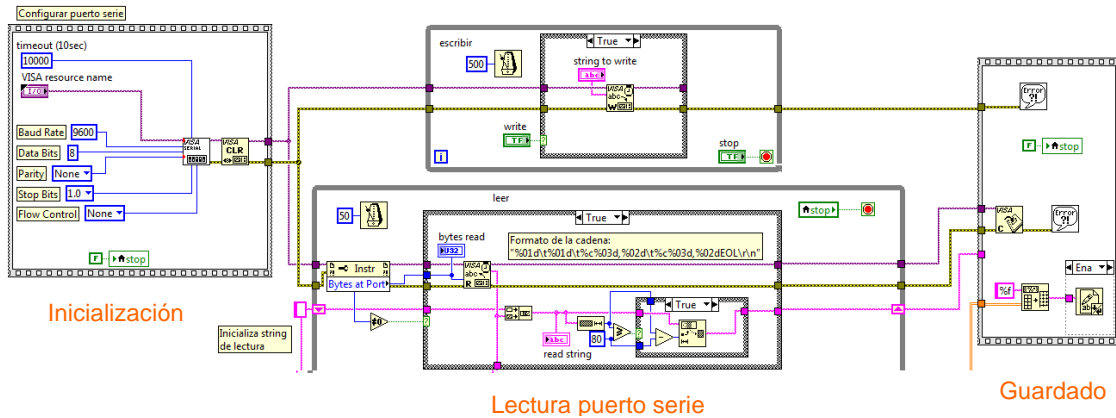


Figura 37. Código LabVIEW para lectura escritura de puerto serie.

Tras la lectura de la trama, se inicia el procesado de la misma, tal como se observa en la figura 38, se inicializa un array de tipo single con cuatro posiciones para almacenar la información procesada del canal leído, una para el número del canal, otra para el control, y las dos restantes para la temperatura de la unión y la temperatura medida del termopar. En esta figura se aprecia mejor la inicialización del string de lectura al que se hacía referencia en el párrafo anterior. Dentro de una estructura “while”, mientras no se pare la ejecución, se irán recibiendo las tramas leídas separadas por un delimitador, la división de tramas se realiza mediante la búsqueda de “tokens” o delimitadores, utilizando la función “Match Pattern”, en este caso el delimitador a buscar es ‘EOL\r\n’ si delante del delimitador se encuentran 19 bytes, la trama es válida y se procesa, el resto de la cadena se indexa para el siguiente procesado.

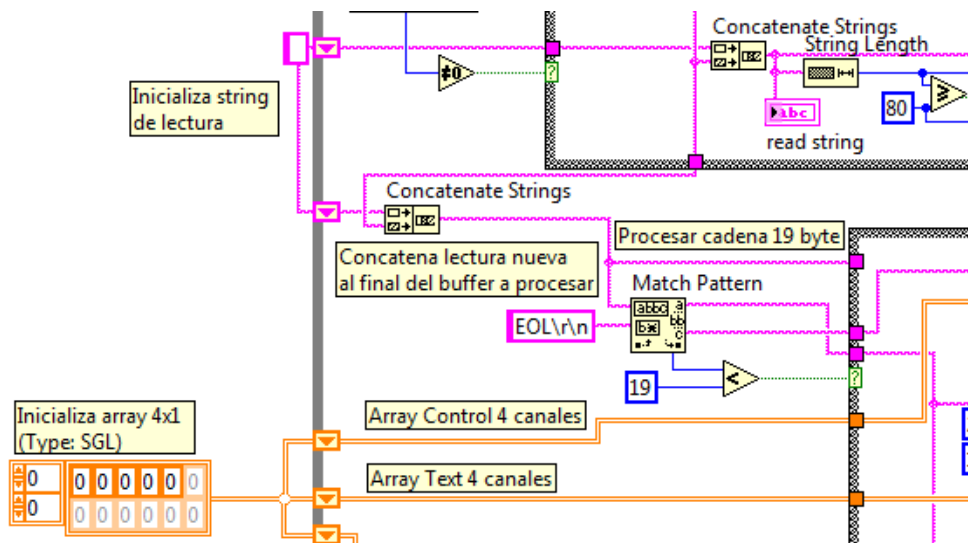


Figura 38. Inicio del procesado de la trama leída.

El procesado de la trama se basa en el tamaño que ocupa cada medida, utilizando la herramienta "String Subset" se pueden separar substrings de la trama original conociendo su tamaño y su posición de inicio dentro de la trama. En primer lugar se encuentra el byte de control que informa del estado de la conexión del termopar, si no hay error de conexión se procesarán las temperaturas y se guardarán en el array. El código completo de procesado de trama se puede observar en la figura 39 que se desglosa en las figuras 40 y 41 para facilitar su comprensión.

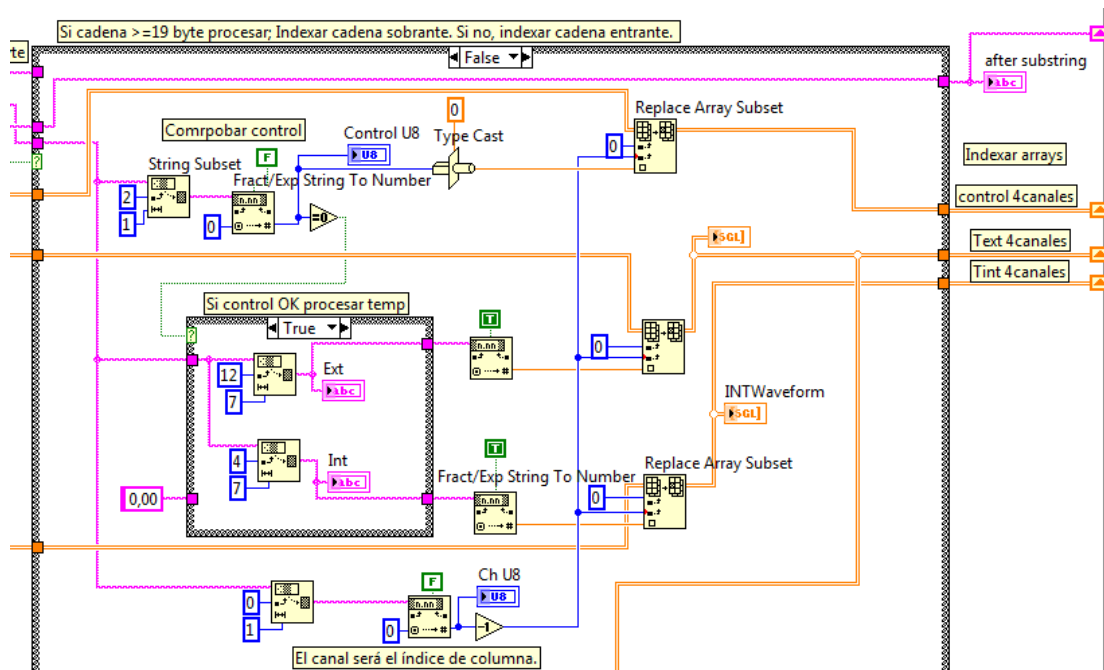


Figura 39. Procesado de la trama, código completo.

Las medidas sólo se procesarán en caso de ser válidas, se supondrá medida válida cualquier medida en la que el termopar se encuentre conectado de forma correcta, es decir, siempre que el byte de control sea cero. Como se puede apreciar en la figura 40, dicha comprobación se lleva a cabo mediante el primer byte de la trama, al tener ésta formato string (cadena de caracteres) es necesario cambiar el formato, por lo que mediante la función "Exp/Frac String To Number" se cambia a byte entero sin signo, unsigned byte o U8 para comprobar si el byte es igual a cero y, en tal caso, se procesan las temperaturas dentro de una estructura tipo case. Tras la comprobación se realiza un "Type Cast" para convertir el dato U8 en uno del tipo single, de modo que se pueda almacenar en el array.



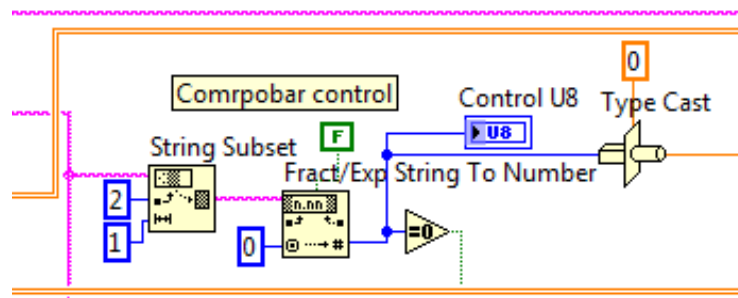


Figura 40. Procesado de la trama, comprobación del estado de la conexión. Byte de control.

En la figura 41 se puede ver el caso “True” de la estructura, aquí se procede a separar las dos medidas contenidas en la trama. Se sabe que cada temperatura tiene un tamaño de siete bytes, la temperatura interna o temperatura de la unión empieza en el cuarto byte de la trama, y la temperatura externa o la temperatura medida del termopar comienza en el byte número doce, tal como se aprecia en la figura, estos datos se pueden separar mediante la función “String Subset” de modo que a la salida de se encuentra el substring en cuestión. Estos substrings se pasan a número mediante la función “Exp/Frac String To Number” tal como se hizo con el byte de control, en este caso el bit de tipo de separador decimal aparece con valor “true” para que dicho separador sea una coma, por defecto el separador es un punto decimal. Con esto las medidas se añaden al array correspondiente. En el caso “False” es decir, si hay algún fallo en la conexión del termopar, se pasa un valor de 0,00 por defecto.

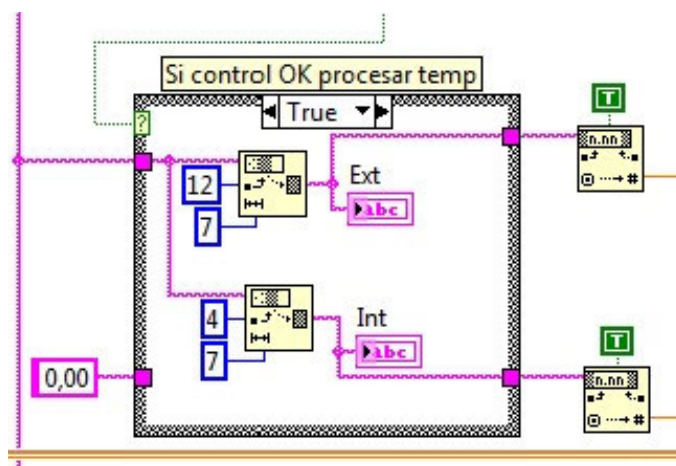


Figura 41. Procesado de la trama. Procesado de temperaturas con valor decimal.

Para identificar el canal al que corresponden las medidas almacenadas, se separa el primer byte de la cadena, que corresponde al canal medido, el substring se convierte a tipo U8 y se le resta uno para que el índice de canal se corresponda con el de la columna del array de 0 a 3, de este modo las medidas quedarán organizadas con

una columna por canal. El código que lleva a cabo este proceso se puede ver en la figura 42.

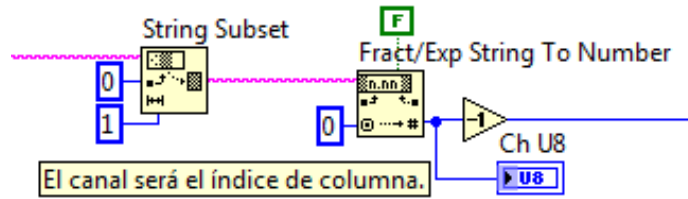


Figura 42. Procesado de trama de medida. Índice de canal medido.

Mediante la herramienta “Replace Subset Array” se añade el valor de entrada en la posición del array indicada por el índice, tomando como índice el valor obtenido del canal leído tal como se ha visto en la figura 43 ,se consigue dar el formato necesario para poder componer una tabla con el control y las medidas de cada canal. Por último, tal como se explicó en la figura 37, estos datos se añaden a un archivo compatible con Excel.

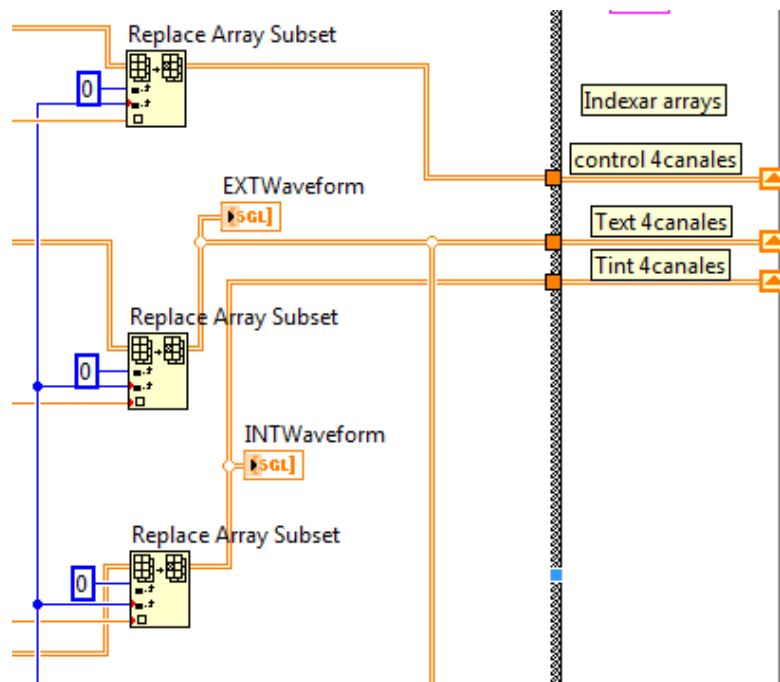


Figura 43. Procesado de trama de medidas. Guardado de datos en array.

## 9.3 Interfaz de usuario LabVIEW

En el apartado anterior se ha mostrado el código realizado mediante LabVIEW para mostrar en el PC la trama recibida por puerto serie desde la FPGA. Se han realizado dos versiones de este programa. La primera es una interfaz básica que únicamente permite la visualización de la medida y la segunda incluye opciones extra orientadas a la calibración.

### 9.3.1 Interfaz de usuario para visualización de medidas.

La interfaz básica tal como se puede observar en la figura, incluye una gráfica en la que se muestran las curvas de temperatura de termopar de los cuatro canales, representado cada uno en un color. En la parte superior se muestran los valores de temperatura en °C, tanto de la medida de termopar como de la medida de la unión fría en cada canal.

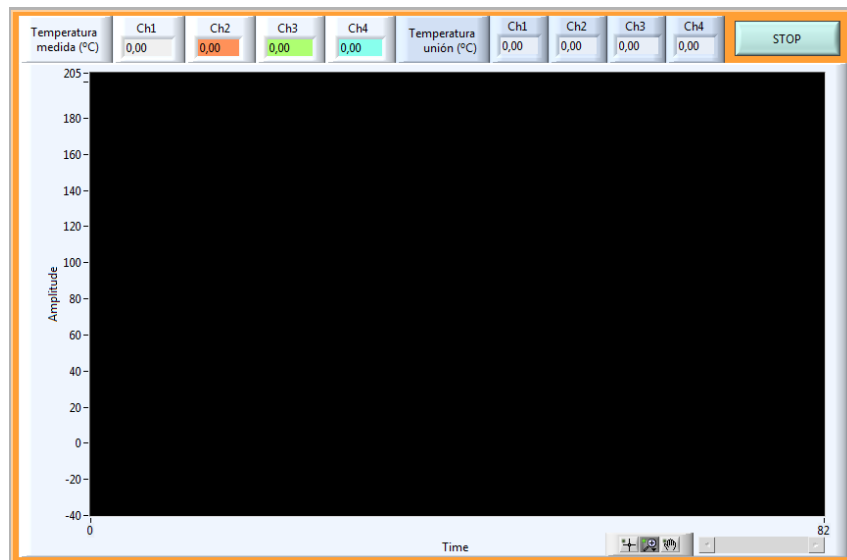


Figura 44. Interfaz gráfica de usuario en LabVIEW.

### 9.3.2 Interfaz de usuario para visualización y captura de medidas.

La interfaz de calibración está destinada al guardado de temperaturas medidas, en la figura 45 se puede ver que, al igual que la interfaz básica, incluye una gráfica en la que se muestran las curvas de temperatura de termopar de los cuatro canales. Además, dispone de opciones para configurar la rampa de medidas que se va a aplicar, mediante el tamaño de escalón y un botón de Up/Down que permite definir si la rampa es ascendente o descendente. Pulsando el botón de guardado, se capturan los valores en ese instante para guardarlos en el archivo \*.cvs. En la parte superior se muestra la trama recibida por puerto serie, para comprobar que se procesa de forma adecuada; y los valores de temperatura en °C, tanto de la medida de termopar como de la medida de la unión fría en cada canal.

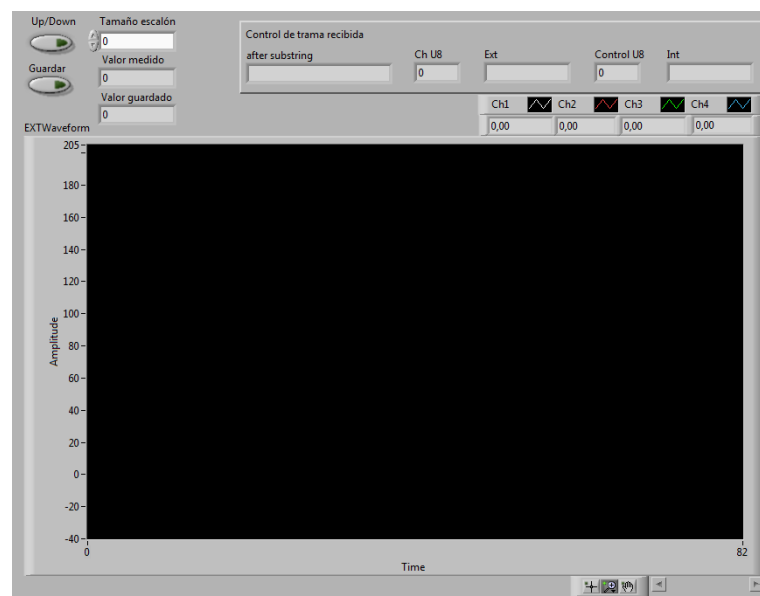


Figura 45. Interfaz gráfica de usuario en LabVIEW.

# 10 . Proceso de calibración

---

## 10.1 Introducción



Figura 46. Calibrador FLUKE 725.

Para poder calibrar el sistema es necesario el uso de un calibrador especial para termopar, como el mostrado en la figura 46, mediante el que se puedan simular diferentes temperaturas a la entrada de la placa acondicionadora, de modo que se puedan comparar los valores obtenidos a la salida. Con estos datos se pueden realizar los cálculos pertinentes para calibrar el sistema mediante un ajuste de offset y ganancia. Ambos valores se aplican mediante las ecuaciones de ajuste con lo que el sistema queda calibrado y cumple el requisito de un error de medida

por debajo de 1°C en valor absoluto. La calibración de la medida se realiza por software, mediante el procesador de la FPGA, se guardan los valores de ajuste calculados y se aplican a las medidas recibidas en la FPGA por SPI.

## 10.2 Calibrador del sistema

Para probar el sistema con un rango de temperaturas amplio y controlado se emplea un calibrador específico con salida tipo termopar. El equipo empleado es el Calibrador de Temperatura Fluke725 del fabricante FLUKE. Este modelo es compatible con diez tipos de termopar, incluyendo el tipo J que concierne a este proyecto. Mediante su modo de fuente se pueden simular temperaturas con una resolución de  $0,1^{\circ}\text{C}$ . El equipo genera en su salida una tensión correspondiente a la que generaría el termopar tipo J si se le aplicase dicha temperatura. En la figura 47 se puede observar la conexión del calibrador.

Según las especificaciones del equipo, la simulación de termopar tipo J del Fluke725 tiene una exactitud de  $0,7^{\circ}\text{C}$ .

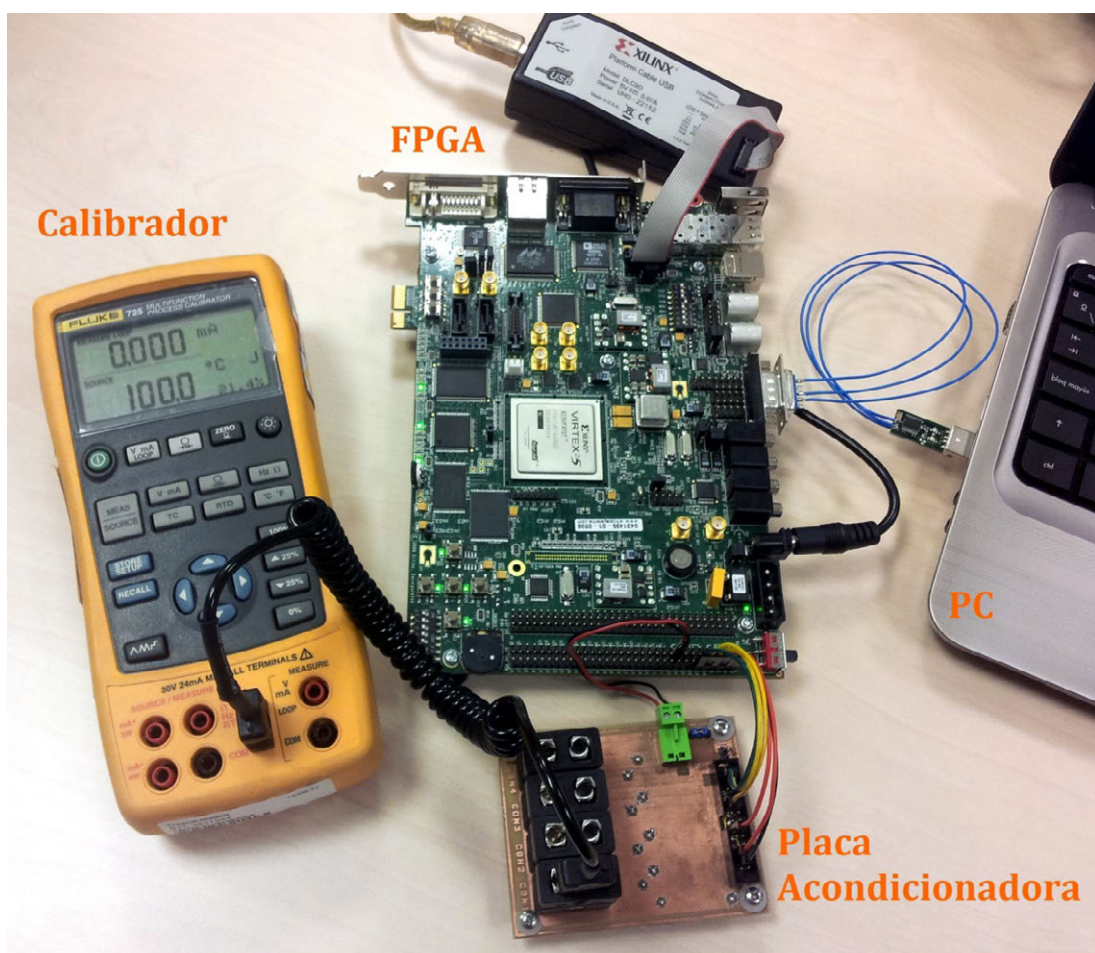


Figura 47. Sistema completo con el calibrador utilizado.

### 10.3 Cálculos de ajuste y calibración mediante Excel

Una vez capturados y guardados los datos de las medidas en LabVIEW, se abre el archivo en una hoja de cálculo de Excel. A continuación se describen los cálculos realizados en Excel para la calibración del sistema, es decir, el valor de ganancia y offset que se deberá aplicar a cada canal para ajustar las medidas. Dado que el proceso en todos los canales es el mismo, en este capítulo se presentan las medidas y cálculos realizados para el canal 1, en los anexos se incluyen los resultados obtenidos para los 4 canales.

La ecuación que se aplica para el ajuste de las medidas realizadas debe conseguir la mayor aproximación posible a los valores ideales. Tal como se presenta en la figura 48, la diferencia de pendiente entre la recta de valores ideales y la aproximación lineal de los valores medidos es la ganancia que se deberá aplicar, la diferencia de medidas entre el valor ideal cero y el real es el offset del canal.

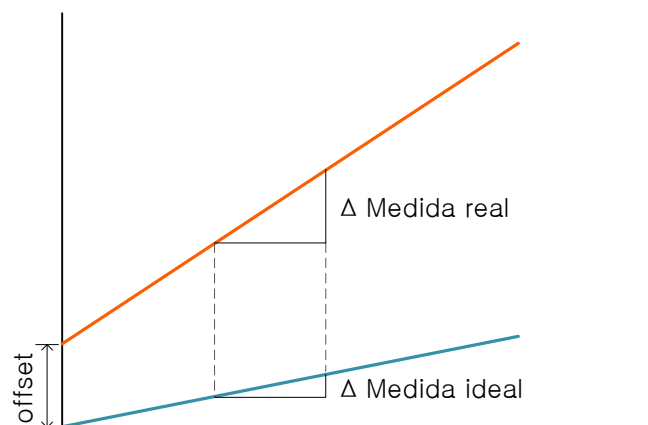


Figura 48. Representación gráfica de los errores de ganancia y offset.

En primer lugar se debe calcular el error de ganancia, este valor es la relación entre la variación de las medida ideales y las medidas reales obtenidas. Para calcular estas variaciones, se restan los valores extremos, es decir, al valor mayor se le resta el menor, tanto en los valores ideales como en los reales. El ajuste de ganancia a aplicar en la ecuación 4 se halla dividiendo ambos valores, tal como se indica en la ecuación 3

$$G = \frac{\Delta Medida ideal}{\Delta Medida real} \quad (Ec 3)$$

Ecuación 3. Ecuación de cálculo de la ganancia.

Mediante la corrección de offset se consigue el paso por cero en la gráfica de valores reales, es decir, al aplicar a la entrada una temperatura de 0°C, el resultado a la salida será también de 0°C. Para ello, tal como se indica en la ecuación 4, el valor medido al aplicar 0°C con el calibrador, multiplicado por la ganancia, es el offset del canal a calibrar. Para hacer la corrección de offset, esta medida se resta a las demás y tal como se indica en la ecuación 5.

$$\text{Offset} = \text{Medida real}|_{Ti=0^{\circ}\text{C}} * \text{Ganancia} \quad (\text{Ec. 4})$$

*Ecuación 4. Ecuación de cálculo del error de offset.*

Los valores de ganancia y offset calculados se deben sustituir en la ecuación 4 para realizar los ajustes de calibración a las medidas realizadas.

$$\text{Medida corregida} = \text{Medida real} * \text{Ganancia} - \text{Offset} \quad (\text{Ec. 5})$$

*Ecuación 5. Ecuación de corrección de ganancia y offset.*

A continuación se presentan los resultados obtenidos en los cálculos teóricos realizados en Excel, tras realizar un primer cálculo según los pasos presentados en este capítulo, se observa que el error resultante tras aplicar los ajustes queda fuera de rango. Esto se debe a que el comportamiento del termopar no es lineal, y la compensación realizada por el conversor no corrige esa no linealidad. Se ha optado por una aproximación mediante tramos lineales tal como se presenta en el segundo método de calibración.

### **Primer método de calibración**

En primer lugar se realizan los cálculos de calibración básicos mostrados anteriormente. En esta primera prueba, se observa que la desviación en los resultados de las medidas corregidas tiene una tendencia poco lineal, debido tanto a error de offset como de ganancia. En todo el rango de temperatura se aprecia como el error final, que se observa en la última columna de la tabla 12., no queda dentro del error máximo fijado en  $\pm 1^{\circ}\text{C}$ .



Ganancia = 1,075

Offset = 3,23°C

Ideal (°C)	Medida (°C)	Error (°C)	Corrección Ganancia (°C)	Corrección Offset (°C)	Error final (°C)
0,00	3,00	3,00	3,23	0,00	0,00
10,00	11,75	1,75	12,63	9,41	0,59
20,00	20,75	0,75	22,31	19,09	0,91
30,00	29,50	-0,50	31,72	28,49	1,51
40,00	38,50	-1,50	41,40	38,17	1,83
50,00	47,50	-2,50	51,08	47,85	2,15
60,00	56,75	-3,25	61,02	57,80	2,20
70,00	66,00	-4,00	70,97	67,74	2,26
80,00	75,25	-4,75	80,91	77,69	2,31
90,00	84,75	-5,25	91,13	87,90	2,10
100,00	94,00	-6,00	101,08	97,85	2,15
110,00	103,25	-6,75	111,02	107,80	2,20
120,00	112,75	-7,25	121,24	118,01	1,99
130,00	122,25	-7,75	131,45	128,23	1,77
140,00	131,75	-8,25	141,67	138,44	1,56
150,00	141,25	-8,75	151,88	148,66	1,34
160,00	150,75	-9,25	162,10	158,87	1,13
170,00	160,25	-9,75	172,31	169,09	0,91
180,00	169,75	-10,25	182,53	179,30	0,70
190,00	179,50	-10,50	193,01	189,78	0,22
200,00	189,00	-11,00	203,23	200,00	0,00

Tabla 12. Cálculos del primer método de calibración realizados con Excel.

Tras la corrección, se observa que el error de las medidas realizadas es no lineal tal como se presenta en la figura 49, dando como resultado errores de medida superiores a dos grados en el peor caso, por ello, la corrección con una sola ganancia para todo el rango de medidas no resulta para nada aplicable a este sistema.

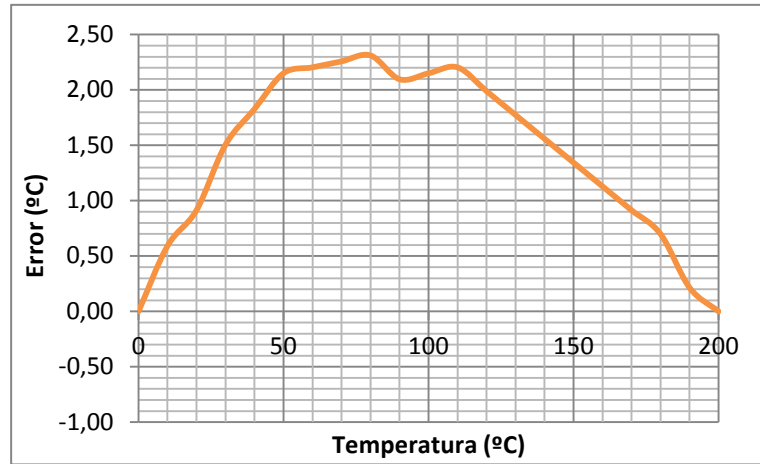


Figura 49. Error de medida tras la calibración por el primer método.

A la vista del comportamiento del error, se comprueba, como ya se había previsto, que la compensación realizada por el MAX31855 asume que los termopares tienen una sensibilidad constante y por tanto un comportamiento lineal, este hecho da como resultado los errores de medida ya presentados y una tendencia de las medidas no lineal. Debido a esto, el método de calibración realizado mediante una ganancia única no es adecuado para cumplir los requisitos del sistema.

El resultado de la medida calibrada respecto a la ideal se presenta en la figura 50, en naranja se puede ver la tendencia de las medidas ya corregidas, y en negro la línea de tendencia. Se observa que la ligera curvatura de las medidas respecto a la lineal es suficiente para hacer que el error de medida quede fuera del rango establecido en  $\pm 1^\circ\text{C}$

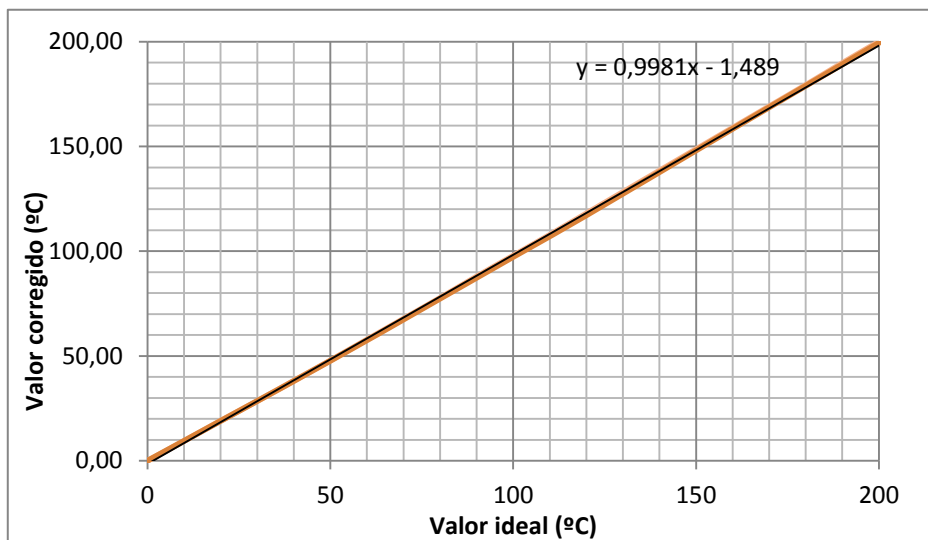


Figura 50. Comparación del valor medido respecto al ideal con el primer método de calibración.

### Segundo método de calibración

Como solución se ha optado por dividir el intervalo de temperatura en cinco regiones. De este modo, se consigue un resultado mucho más lineal, con un error de medida siempre por debajo de un grado, con lo que se cumple la especificación del diseño en lo referente al error de medida.

Los intervalos se han definido en función del valor de ganancia de cada medida, se han agrupado los valores más cercanos de modo que cada tramo sea lo más lineal posible. Tras definir los grupos, se calcula el valor medio de cada grupo, presentado en la tercera columna de la tabla 13, y se aplica a los valores medidos. De este modo, se consigue que el error del sistema quede dentro del rango especificado en  $\pm 1^\circ\text{C}$ , siendo el error más elevado el correspondiente a la medida ideal de  $200^\circ\text{C}$ .

Ideal ( $^\circ\text{C}$ )	Medida ( $^\circ\text{C}$ )	Ganancia aplicada	Corrección Ganancia ( $^\circ\text{C}$ )	Corrección Offset ( $^\circ\text{C}$ )	Error final ( $^\circ\text{C}$ )
0,00	3,00	1,13	3,38	0,00	0,00
10,00	11,75	1,13	13,24	9,86	-0,14
20,00	20,75	1,13	23,38	20,00	0,00
30,00	29,50	1,13	33,24	29,86	-0,14
40,00	38,50	1,13	43,38	40,00	0,00
50,00	47,50	1,11	52,73	49,34	-0,66
60,00	56,75	1,11	62,99	59,61	-0,39
70,00	66,00	1,11	73,26	69,88	-0,12
80,00	75,25	1,11	83,53	80,15	0,15
90,00	84,75	1,11	94,07	90,69	0,69
100,00	94,00	1,10	103,40	100,02	0,02
110,00	103,25	1,10	113,58	110,19	0,19
120,00	112,75	1,10	124,03	120,64	0,64
130,00	122,25	1,09	133,25	129,87	-0,13
140,00	131,75	1,09	143,61	140,23	0,23
150,00	141,25	1,09	153,96	150,58	0,58
160,00	150,75	1,08	162,81	159,43	-0,57
170,00	160,25	1,08	173,07	169,69	-0,31
180,00	169,75	1,08	183,33	179,95	-0,05
190,00	179,50	1,08	193,86	190,48	0,48
200,00	189,00	1,08	204,12	200,74	0,74

Tabla 13. Cálculos del segundo método de calibración realizados con Excel.

En la figura 51 se aprecia que la gráfica resultante es continua para cada tramo, al realizar la corrección de ganancia por tramos, se consigue que el error no supere el valor máximo especificado en  $\pm 1^\circ\text{C}$ .

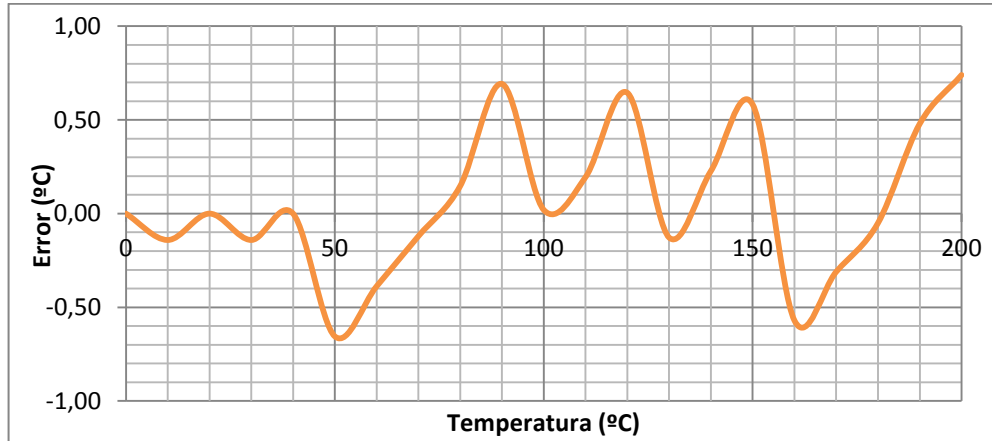


Figura 51. Error de medida tras la calibración por el segundo método.

En la figura 52 se representa el valor real ajustado frente al valor ideal de medida. Si se compara esta gráfica con la de la figura 46, se puede observar que la desviación de la curva respecto a su normal, es menor que la desviación resultante en la corrección por el primer método.

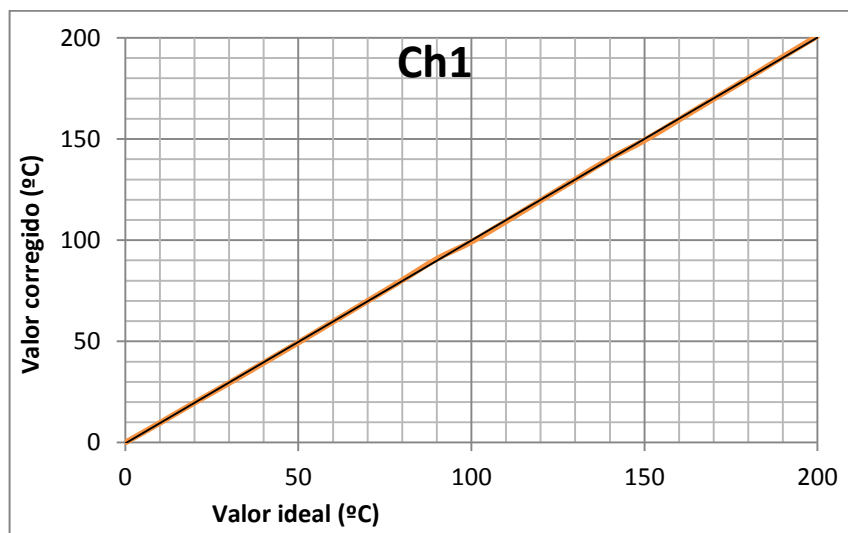


Figura 52. Comparación de la medida real frente a la ideal con el segundo método de calibración.

## 10.4 Verificación de los cálculos de calibración

Para poder hacer las pruebas del sistema completo, una vez realizados los cálculos de calibración en Excel, se añaden al código del procesador. Se monta el sistema y mediante el calibrador se aplica de nuevo la curva de temperaturas que se ha utilizado en la fase de calibración, es decir, un rango de 0 a 200°C en incrementos de 10°C. Una vez validados los resultados, verificando que el error entra en los límites especificados de  $\pm 1^\circ\text{C}$  se da el diseño por válido.

Para la visualización y adquisición de las medidas se puede utilizar la interfaz de LabVIEW, en su versión de calibración que permite exportar los resultados a Excel. Si lo que se desea es realizar una comprobación visual básica, basta con utilizar la versión de usuario.

Si se observa que el error se sale de rango, se repetirá el proceso de calibración. Una vez realizados los ajustes pertinentes, el sistema queda probado y cumple las especificaciones requeridas.



# 11 . Conclusiones y posibles mejoras

---

## 11.1 Conclusiones

Con la realización de este proyecto se han adquirido los conocimientos necesarios para la configuración de una FPGA, así como el uso de los periféricos de su tarjeta de desarrollo asociada y la programación del procesador embebido que controla dichos dispositivos. El fabricante facilita el material necesario para realizar esta labor, incluyendo los datasheet del kit de desarrollo, tutoriales y manuales asociados a la placa.

Para poder llevar a cabo la configuración de la FPGA, ha sido necesario aprender a manejar la plataforma de desarrollo de Xilinx, estudiando la documentación que el fabricante pone a disposición del desarrollador. También se ha profundizado en el lenguaje de descripción hardware VHDL para comprender la programación de los bloques que describen los drivers de la FPGA. Del mismo modo, se han estudiado las funciones en lenguaje C necesarias para el manejo de dichos drives desde el procesador.

La interfaz de usuario implementada en el PC ha requerido profundizar en el lenguaje gráfico LabVIEW de National Instruments, estudiando el mejor modo de procesar la trama recibida del procesador y automatizar la captura de medidas para los cálculos de calibración.

El diseño de la placa acondicionadora ha sido realizado en OrCAD, siguiendo los procesos que se realizan para el desarrollo de placas de mayor complicación, con un mayor número de capas y componentes. Esto ha implicado el manejo de bases de datos de componentes asociada con el Master Part List en OrCAD Capture.

## 11.2 Posibles mejoras

La realización del prototipo ha permitido estudiar el comportamiento del conversor de termopar a digital MAX-31855, y su exactitud en la conversión con termopares tipo J. Una posible mejora, o prueba adicional para una nueva versión del prototipo sería el uso de termopares tipo K, con su conversor compatible correspondiente, ya que el comportamiento de este tipo de termopar en el rango de temperaturas a medir es, teóricamente, más estable que el tipo J.

En cuanto al diseño de la placa, se debería redistribuir los componentes, de modo que el conversor quede situado en la parte inferior del conector de termopar, consiguiendo así que la medida de la temperatura de la unión fría realizada por el conversor resulte lo más exacta posible.

Los datos de calibración se han guardado en un array incluido en el código del procesador. Este proceso se podría mejorar añadiendo a la placa acondicionadora una memoria flash donde almacenar los valores de ajuste calculados. De este modo, los datos se leerían en la fase de inicialización en el código del procesador. La placa incluiría por tanto los valores de ajuste asociados a cada canal, se podría incluir también la fecha de calibración, así como una contraseña para asegurar que no se sobrescriban los datos. Al añadir una memoria flash a la placa, el código pasa a ser estándar. Disponer de un código estándar supondría mayor flexibilidad, ya que se podrían utilizar diferentes diseños de placa acondicionadora sin necesidad de modificar la configuración y programación de la FPGA.



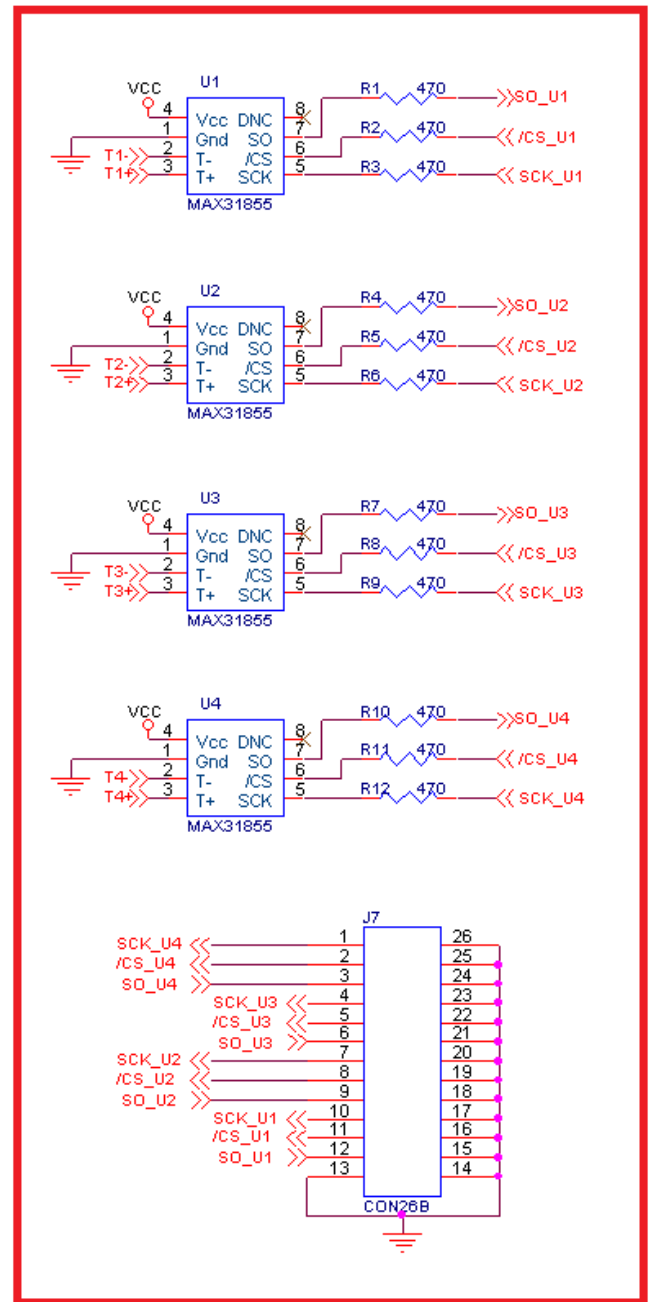
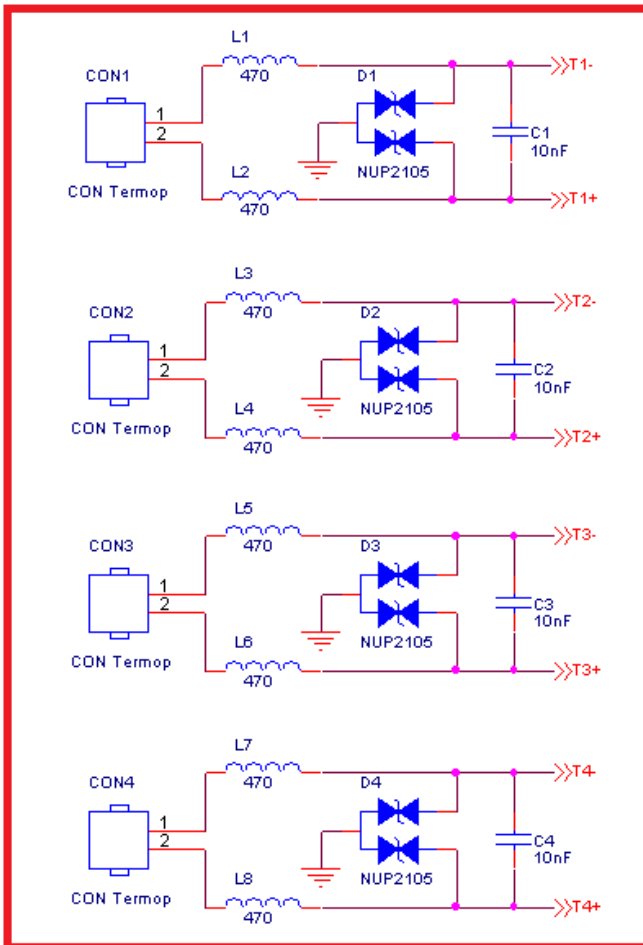
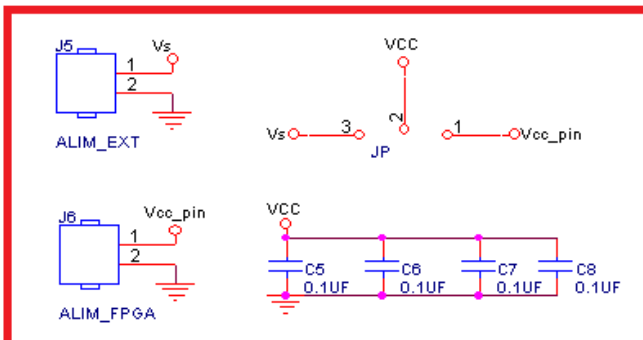
## **12 . Apéndices**

---



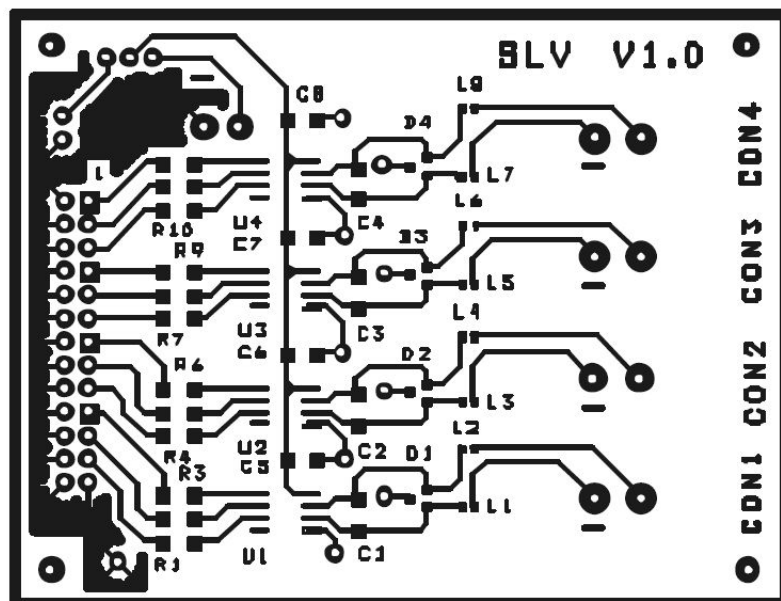
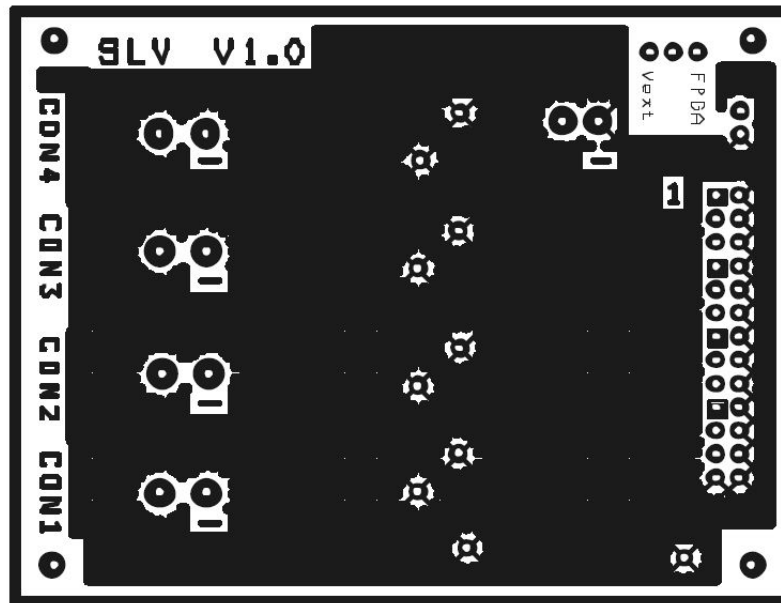
## 12.1 Esquemático completo del Bloque I

A continuación se muestra el esquemático de la placa acondicionadora de señal para los cuatro canales de medida, que compone el Bloque I del proyecto. La descripción de este bloque se encuentra en el apartado 5 de esta memoria.



## 12.2 Fitolitos para la fabricación de la placa acondicionadora del Bloque I

Bajo estas líneas se muestran los fitolitos del diseño del PCB necesarios para el insulado en el proceso manual de fabricación de la placa. En la parte superior se encuentra el fitolito correspondiente a la capa Top, y en la parte inferior de la página se puede ver el correspondiente a la capa Bottom.



### **12.3 Lista de componentes necesarios. MPL (Master Part List)**

Todos los componentes utilizados para la fabricación de la placa acondicionadora se presentan en el archivo MPL (Master Part List) correspondiente al esquemático del apartado anterior.

La tabla completa que se presenta en la siguiente página, incluye los componentes utilizados en el diseño del prototipo así como referencias de fabricante, precio unitario de los componentes y coste total.

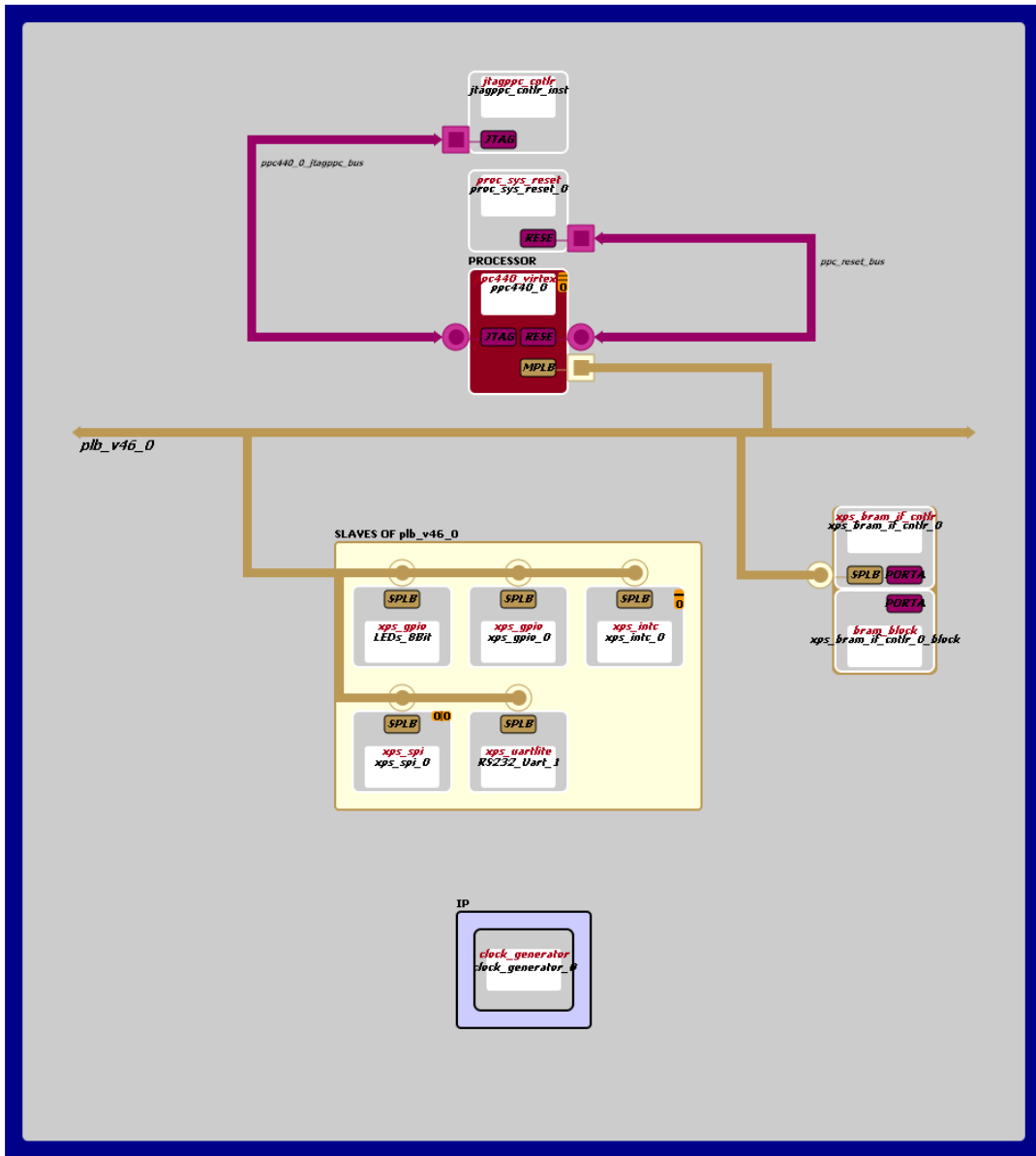
12. Apéndices

Q	Ref.	Valor	Descripción	Footprint	Fabricante	Ref fabricante	Ref distribuidor	Precio	Total
8	CON_32		Conector 32 pin	CON_32				3,00 €	3,00 €
1	J5	ALIM_EXT	Conector alimentación 2 pines extraíble	CONVCC_2				1,00 €	1,00 €
1	J6	ALIM_FPGA	Jumper 2 pin	JUMPER2				0,30 €	0,30 €
1	JP		Jumper 3 pin	JUMPER3				0,30 €	0,30 €
4	CON2, CON3,	CON Termop	Conector termopar tipo J	CON_TERMOP	LABFACILITY	IM-J-LCF	8598274	3,13 €	12,52 €
8	L1, L2, L3, L4, L5, L6, L7, L8	470,00 €	Ferrite bead SMD 0603 470	0603	WURTH ELEKTRONIK	742863147	1748576	0,30 €	2,40 €
12	R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11, R12	470,00 €	Resistencia SMD 0805 470R 1/4W	0805	VISHAY DRALORIC	CRCW0805470RFKEA	1469932	0,02 €	0,26 €
4	U1, U2, U3, U4	MAX 31855	Convertor de termopar a digital MAX31855JASA+	0805	MAXIM	MAX31855 JASA+	MAX31855 JASA+	5,68 €	22,72 €
4	C2, C3, C4, C5	0,1u	Condensador SMD 0805 0,1uF	0805	KEMET	C0805C104J5 RACTU14146 63	CRCW0805470 RFKEA	0,05 €	0,19 €
1	C1	10n	Condensador SMD 0805 10nF	0805	KEMET	C0805C103J3 GACTU	1457736	0,16 €	0,16 €
4	D1, D2, D3, D4	NUP2105	Diodo NUP2105 SOT-23	SOT-23	ON Semiconductor	NUP2105LT1 G	2101833	0,43 €	1,72 €
									44,58 €

## 12.4 IP Cores de XPS

Name	Bus Name	IP Type	IP Version	IP Classification
ppc440_0	plb_v46_0	ppc440_virtex5	1.01.a	Processor
MPLB	No Connection			
SPLB0	No Connection			
SPLB1	No Connection			
PPC440MC	ppc440_0_ppc440mc			
MDCR	No Connection			
SDCR	No Connection			
MFCB	No Connection			
MFCM	ppc440_0_mfcm			
JTAGPPC	ppc440_0_jtagppc_bus			
RESETPPC	ppc_reset_bus			
plb_v46_0		plb_v46	1.04.a	PLBV46 Bus
xps_bram_if_cntrl_0		xps_bram_if_cntrl	1.00.b	Memory Controller
SPLB	plb_v46_0			
PORTA	xps_bram_if_cntrl_0_PORTA			
xps_bram_if_cntrl_0_block		bram_block	1.00.a	Memory
PORTA	xps_bram_if_cntrl_0_port			
PORTB	No Connection			
jtagppc_cntrl_inst		jtagppc_cntrl	2.01.c	Peripheral
JTAGPPCO	jtagppc_cntrl_inst_JTAGPPCO			
proc_sys_reset_0		proc_sys_reset	2.00.a	Peripheral
RESETPPCO	proc_sys_reset_0_RESETPPCO			
LEDx_8bit		xps_gpio	2.00.a	Peripheral
SPLB	plb_v46_0			
xps_gpio_0		xps_gpio	2.00.a	Peripheral
SPLB	plb_v46_0			
xps_spi_0		xps_spi	2.01.b	Peripheral
SPLB	plb_v46_0			
RS232_Uart_1		xps_uartlite	1.01.a	Peripheral
SPLB	plb_v46_0			
clock_generator_0		clock_generator	3.02.a	IP

## 12.5 Diagrama de bloques de XPS



SPECS	
EDK VERSION	11.5
ARCH	virtex5
PART	xc5vfx701ff1136-1
GENERATED	Tue Apr 23 20:28:20 2013

KEY			
SYMBOLS			
	<b>Bus connections</b>	<b>External Ports</b>	<b>Interrupts</b>
	master or initiator	input	Interrupt Controller
	slave or target	output	Interrupt Target
	master slave	input	Interrupt Source
	monitor		X = Controller ID Y = Interrupt Priority
COLORS			
DCR	FSL	OPB	SDCM
FCB	LMB	PLB	USER P2P
		Xilinx P2P	



## 12.6 Restricciones del usuario UCF (User Constraints File)

```
# Virtex-5 ML507 Evaluation Platform
Net fpga_0_RS232_Uart_1_RX_pin LOC = AG15 | IOSTANDARD=LVCMOS33;
Net fpga_0_RS232_Uart_1_TX_pin LOC = AG20 | IOSTANDARD=LVCMOS33;

Net fpga_0_clk_1_sys_clk_pin TNM_NET = sys_clk_pin;
TIMESPEC TS_sys_clk_pin = PERIOD sys_clk_pin 100000 kHz;
Net fpga_0_clk_1_sys_clk_pin LOC = AH15 | IOSTANDARD=LVCMOS33;

Net fpga_0_rst_1_sys_rst_pin TIG;
Net fpga_0_rst_1_sys_rst_pin LOC = E9 | IOSTANDARD=LVCMOS33 | PULLUP;

Net xps_gpio_0_GPIO_IO_pin<0> LOC=AN33 | IOSTANDARD=LVCMOS33;#pinJ6 64

Net xps_spi_0_MISO LOC = H33 | IOSTANDARD=LVCMOS33; #pinJ6 2 S0
Net xps_spi_0_SCK LOC = H32 | IOSTANDARD=LVCMOS33; #pinJ6 12 CLK

Net xps_spi_0_SS_0_pin<0> LOC = F34 | IOSTANDARD=LVCMOS33; #pinJ6 4 /CS1
Net xps_spi_0_SS_0_pin<1> LOC = H34 | IOSTANDARD=LVCMOS33; #pinJ6 6
/CS2
Net xps_spi_0_SS_0_pin<2> LOC = G33 | IOSTANDARD=LVCMOS33; #pinJ6 8
/CS3
Net xps_spi_0_SS_0_pin<3> LOC = G32 | IOSTANDARD=LVCMOS33; #pinJ6 10 /CS4

NET "clk_100_0000MHzPLL0_ADJUST" TNM_NET = "splb_clk";
TIMESPEC "TS_splb_clk" = PERIOD "splb_clk" 200 ns HIGH 50%; #reloj MAX31855
max 5MHz
```

## 12.7 Código de programa en C del PowerPC440

```

/***** Include Files *****/
#include "xparameters.h" /* XPAR parameters */
#include "xspi.h" /* SPI device driver */
#include "xspi_1.h"
#include <stdio.h>
#include <string.h>
#include "xuartlite.h"
#include "xuartlite_i.h"
/***** Constant Definitions *****/

/*
 * The following constants map to the XPAR parameters created in the
 * xparameters.h file. They are defined here such that a user can easily
 * change all the needed parameters in one place.
 */
#define SPI_DEVICE_ID XPAR_SPI_0_DEVICE_ID

// Constants for transmission and data storage
#define BUFFER_SIZE 4 // Bytes to be transmitted/received.
#define NUM_CHANNELS 4 // Number of channels.

// Masks to check the readed data
#define CONNECTION_MASK 0x00000007 // Bit 16 is 1 if connection failure.
#define NEG_TH_MASK 0x80000000 // Sign bit for thermocouple
temperature.
#define NEG_JC_MASK 0x00008000 // Sign bit for Cold-Junction
temperature.

/***** Type Definitions *****/

/*
 * The following data type is used to send and receive data on the SPI
 * interface.
 */
typedef u8 DataBuffer[BUFFER_SIZE];

typedef struct{
    char sign;
    float ftemp;
    u16 temp;
    u16 Temp_integer;
    u16 Temp_decimal;
}Temp;

typedef struct {
    Temp ext; /* External temperature. */
    Temp junct; /* Junction temperature. */
}Temps;

Temps temps [NUM_CHANNELS]; /* Array for temperatures. */
typedef struct{
    u16 Gain;
    u16 Offset;
}CalibParam;

```

```

CalibParam CalibParams [NUM_CHANNELS]; /* Flash simulation. */

/***** Function Prototypes *****/

Void DownloadCalibrationParameters( CalibParam CalibParams[], int i );
void ReadStorage( u8 Buffer[] , int i);
void FillTempStruct ( Temps temps[], int i, char signE, u16 ext, char signJ,
u16 junct );
int CheckConnection ( Temps temps[], int i );
void ProcessTemperatures( Temps temps[], int i );
void Calibrate ( Temps temps[], int i );
void Send ( Temps temps[], int i );

int SpiRead(XSpi *SpiInstancePtr, u16 SpiDeviceId, int ch);

/***** Variable Definitions *****/

/*
 * The instances to support the device drivers are global such that the
 * are initialized to zero each time the program runs.
 */
static XSpi SpiInstance; /* The instance of the SPI device. */
static XUartLite UartInstance; /* The instance of the UART Lite device. */

/*
 * The following variables are used to read the Spi device,
 * global to avoid having large buffers on the stack.
 */
u8 ReadBuffer[BUFFER_SIZE];

u8 values[1800][BUFFER_SIZE];
u8 control; /* Connection check info. */
u8 j=0; /* Measure index. */
u32 data = 0; /* Data read RAW. */

//SIMULACION FLASH
u16 Gains [NUM_CHANNELS];
u16 Offsets [NUM_CHANNELS];

/*****
 * Main function to call the Spi Reading, process the data and send it
 * via RS-232.
 *
 * @param None
 * @return XST_SUCCESS if successful, otherwise XST_FAILURE.
 * @note None
 *****/
int main(void)
{
    int Status;
    int ch_i = 0; /* Array index. */
    int chx = 0x00; /* Channel select. */
    char cadena [100];

```

```
    /* Uart Lite initialization. */
    Status = XUartLite_Initialize( &UartInstance,
XPAR_UARTLITE_0_DEVICE_ID );
    if ( Status != XST_SUCCESS )
    {
        return XST_FAILURE;
    }

    //FLASH Simulation
    Gains[0][0] = 0x00000131;
    Gains[0][1] = 0x00000129;
    Gains[0][2] = 0x00000101;
    Gains[0][3] = 0x00000124;
    Gains[0][4] = 0x00000121;

    Gains[1][0] = 0x00000131;
    Gains[1][1] = 0x00000129;
    Gains[1][2] = 0x00000126;
    Gains[1][3] = 0x00000124;
    Gains[1][4] = 0x00000121;

    Gains[2][0] = 0x00000131;
    Gains[2][1] = 0x00000129;
    Gains[2][2] = 0x00000126;
    Gains[2][3] = 0x00000124;
    Gains[2][4] = 0x00000121;

    Gains[3][0] = 0x00000132;
    Gains[3][1] = 0x00000130;
    Gains[3][2] = 0x00000126;
    Gains[3][3] = 0x00000124;
    Gains[3][4] = 0x00000121;

    Offsets[0] = 0x0000033D;
    Offsets[1] = 0x0000033D;
    Offsets[2] = 0x000003A0;
    Offsets[3] = 0x000003C0;

    for (ch_i=0; ch_i < NUM_CHANNELS; ch_i++)
    {
        DownloadCalibrationParameters( CalibParams, ch_i );
    }

    while (1)
    {
        j++;
        /* Start with channel #1 */
        chx=0x01;
        /* Get data from all channels. */
        for (ch_i=0; ch_i < NUM_CHANNELS; ch_i++)
        {
            Status = SpiRead(&SpiInstance, SPI_DEVICE_ID, chx);
            if (Status != XST_SUCCESS)
            {
                return XST_FAILURE;
            }
        }
        XSpi_Stop(&SpiInstance);
    }
}
```

```

        chx = chx << 1; /* Next channel. */

        ReadStorage( ReadBuffer, ch_i );
        control = CheckConnection (temps, ch_i);
        if (!control)
        {
            ProcessTemperatures( temps, ch_i);
            Calibrate ( temps, ch_i);
        }
        Send ( temps, ch_i);
        usleep(500000); /* Delay. */
    }
}
return XST_SUCCESS;
}

void DownloadCalibrationParameters( CalibParam CalibParams[], int i)
{
    int k;
    for (j=0; j<5;j++)
    {
        CalibParams[i].Gain[k] = Gains[i][k];
    }
    CalibParams[i].Offset = Offsets[i];
}

void ReadStorage( u8 Buffer[], int i)
{
    for ( i=0; i<BUFFER_SIZE;i++ )
    {
        data = data << 8;
        data = data | Buffer[i];
    }
}

void FillTempStruct (Temps temps[], int i, char signE, u16 ext, char signJ,
u16 junct)
{
    temps[i].ext.sign = signE;
    temps[i].ext.temp = ext; //need an extra digit to use Gain and Offset
with 3 decimal digits.
    temps[i].junct.sign = signJ;
    temps[i].junct.temp = junct;
}

int CheckConnection ( Temps temps[], int i )
{
    int ctrl = 0;

    ctrl = data & CONNECTION_MASK;
    //Ctrl 0:OK, 1:Not conected, 2:short to GND, 4:short to Vcc
    if (ctrl !=0)
    {
        if (ctrl > 3)
        {
            ctrl = 3;
        }
        FillTempStruct ( temps, i, 'F', ctrl<<4, 'F', ctrl<<4);
    }
}

```

```
    }
    return (ctrl);
}

void ProcessTemperatures( Temps temps[], int i )
{
    u32 tempE; //need extra byte to operate with gain.
    u32 tempJ;
    char signE;
    char signJ;

    u32 gain;
    u32 offset;

    offset = 0x2C0;//Offsets[0];

    /* Processing Cold-Junction temperature. */
    if (!(data & NEG_JC_MASK))
    {
        /* Check for positive temperature. */
        tempJ = ((data & 0x0000FFFF) >> 4);
        signJ = '+';
    }
    else
    {
        /* Temperature is negative. */
        tempJ = ((data & 0x0000FFFF) >> 4);
        tempJ *= -1;
        signJ = '-';
    }

    /* Processing Thermocouple temperature. */
    tempE = (data >> 16);

    if (!(tempE >> 4 & 0x00008000))
    {
        signE = '+';
    }
    else
    {
        /* Temperature is negative. */
        tempE = (tempE * -1) & 0x000FFFFF;
        signE = '-';
    }

    if (tempE <= 0x2800) //40°C
    {
        gain = CalibParams[i].Gains[0];
    }
    else if (tempE <= 0x5A00) //90°C
    {
        gain = CalibParams[i].Gains[1];
    }
    else if (tempE <= 0x7800) //120°C
    {
        gain = CalibParams[i].Gains[2];
    }
    else if (tempE <= 0x9600) //150°C
```

```

    {
        gain = CalibParams[i].Gains[3];
    }
    else
    {
        gain = CalibParams[i].Gains[4];
    }
    tempE *= gain;
    tempE = tempE<<4;
    tempE = ((tempE - CalibParams[i].Offset));
    tempE=tempE>>12;

    FillTempStruct ( temps, i, signE, tempE, signJ, tempJ);
}

void Send ( Temps temps[], int i )
{
    temps[i].junct.Temp_integer = temps[i].junct.temp>>4;
    temps[i].junct.Temp_decimal = (temps[i].junct.temp&0x0F)*10000/1600;
    temps[i].ext.Temp_integer = temps[i].ext.temp>>4;
    temps[i].ext.Temp_decimal = (temps[i].ext.temp&0x0F)*10000/1600;

    xil_printf ( "%01d\t%01d\t%c%03d,%02d\t%c%03d,%02dEOL\r\n",
        i+1, control,
        temps[i].junct.sign, temps[i].junct.Temp_integer,
        temps[i].junct.Temp_decimal,
        temps[i].ext.sign, temps[i].ext.Temp_integer,
        temps[i].ext.Temp_decimal);
}

/*****
* This function receive data from MISO pin.
* @param SpiInstancePtr is a pointer to the instance of Spi component.
* @param SpiDeviceId is the Device ID of the Spi Device and is the
* XPAR_<SPI_instance>_DEVICE_ID value from xparameters.h.
* @return XST_SUCCESS if successful, otherwise XST_FAILURE.
*****/
int SpiRead(XSpi *SpiInstancePtr, u16 SpiDeviceId, int ch )
{
    int Status;
    u32 Count;
    u8 Test;

    /*
    * Initialize the SPI driver so that it's ready to use,
    * specify the device ID that is generated in xparameters.h
    */

    Status = XSpi_Initialize(SpiInstancePtr, SpiDeviceId);
    if (Status != XST_SUCCESS)
    {
        return XST_FAILURE;
    }

    /*

```

```
    * Perform a self-test to ensure that the hardware was built
correctly.
    */

    Status = XSpi_SelfTest(SpiInstancePtr);
    if (Status != XST_SUCCESS)
    {
        return XST_FAILURE;
        print ("XSpi_SelfTest FAIL\n\r");
    }

    /*
    * Set the Spi device as a master and With manual slave selection.
    */

    Status = XSpi_SetOptions(SpiInstancePtr, XSP_MASTER_OPTION |

        XSP_MANUAL_SSELECT_OPTION);
    if (Status != XST_SUCCESS)
    {
        print ("XSpi_SetOptions FAIL\n\r");
        return XST_FAILURE;
    }

    /*
    * Slave Select.
    */

    Status = XSpi_SetSlaveSelect(SpiInstancePtr, ch);
    if (Status != XST_SUCCESS)
    {
        print ("XSpi_SetSlaveSelect FAIL\n\r");
        return XST_FAILURE;
    }

    /*
    * Start the SPI driver so that the device is enabled.
    */

    XSpi_Start(SpiInstancePtr);

    /*
    * Disable Global interrupt to use polled mode operation
    */

    XSpi_mIntrGlobalDisable(SpiInstancePtr);

    /*
    * Initialize the read buffer to zero so it can be verified
    * after the read.
    */
    for (Count = 0; Count < BUFFER_SIZE; Count++)
    {
        ReadBuffer[Count] = 0;
    }
}
```



```
    /*  
    * Transmit data.  
    */  
    XSpi_Transfer(SpiInstancePtr, ReadBuffer, ReadBuffer, BUFFER_SIZE);  
    return XST_SUCCESS;  
}
```

## 12.8 Código del archivo MHS

```

#####
# Created by Base System Builder Wizard for Xilinx EDK 11.5 Build EDK_LS5.70
# Fri Jul 06 11:45:03 2012
# Target Board: Xilinx Virtex-5 ML507 Evaluation Platform Rev A
# Family:      Virtex-5
# Device:      xc5vfx70t
# Package:     ff1136
# Speed Grade: -1
# Processor number: 1
# Processor 1: ppc440_0
# Processor clock frequency: 400.0
# Bus clock frequency: 100.0
# Debug Interface: FPGA JTAG
#####

PARAMETER VERSION = 2.1.0

PORT fpga_0_RS232_Uart_1_RX_pin = fpga_0_RS232_Uart_1_RX_pin, DIR = I
PORT fpga_0_RS232_Uart_1_TX_pin = fpga_0_RS232_Uart_1_TX_pin, DIR = O
PORT fpga_0_LEDs_8Bit_GPIO_IO_pin = fpga_0_LEDs_8Bit_GPIO_IO_pin, DIR = IO,
VEC = [0:7]
PORT fpga_0_clk_1_sys_clk_pin = dcm_clk_s, DIR = I, SIGIS = CLK, CLK_FREQ =
100000000
PORT fpga_0_rst_1_sys_rst_pin = sys_rst_s, DIR = I, SIGIS = RST, RST_POLARITY
= 0
PORT clock_generator_0_LOCKED_pin = Dcm_all_locked, DIR = O
PORT xps_gpio_0_GPIO_IO_pin = xps_gpio_0_GPIO_IO_pin, DIR = IO, VEC = [0:6]
PORT xps_spi_0_SCK = xps_spi_0_SCK, DIR = IO
PORT xps_spi_0_MISO = xps_spi_0_MISO, DIR = IO
PORT xps_spi_0_IP2INTC_Irpt_pin = xps_spi_0_IP2INTC_Irpt, DIR = O
PORT xps_spi_0_SS_0_pin = xps_spi_0_SS_0, DIR = O, VEC = [0:3]
PORT xps_spi_0_SS = xps_spi_0_SS, DIR = IO, VEC = [0:3]

BEGIN ppc440_Virtex-5
PARAMETER INSTANCE = ppc440_0
PARAMETER C_IDCR_BASEADDR = 0b0000000000
PARAMETER C_IDCR_HIGHADDR = 0b0011111111
PARAMETER C_SPLB0_NUM_MPLB_ADDR_RNG = 0
PARAMETER C_SPLB1_NUM_MPLB_ADDR_RNG = 0
PARAMETER HW_VER = 1.01.a
BUS_INTERFACE MPLB = plb_v46_0
BUS_INTERFACE JTAGPPC = ppc440_0_jtagppc_bus
BUS_INTERFACE RESETPPC = ppc_reset_bus
PORT EICC440EXTIRQ = EICC440EXTIRQ
PORT CPMC440CLK = clk_400_000MHzPLL0
PORT CPMINTERCONNECTCLK = clk_200_000MHzPLL0
PORT CPMINTERCONNECTCLKNT01 = net_vcc
PORT CPMCCCLK = clk_100_000MHzPLL0_ADJUST
END

```

```
BEGIN plb_v46
  PARAMETER INSTANCE = plb_v46_0
  PARAMETER C_DCR_INTFCE = 0
  PARAMETER HW_VER = 1.04.a
  PORT PLB_Clk = clk_100_0000MHzPLL0_ADJUST
  PORT SYS_Rst = sys_bus_reset
END

BEGIN xps_bram_if_cntlr
  PARAMETER INSTANCE = xps_bram_if_cntlr_0
  PARAMETER C_SPLB_NATIVE_DWIDTH = 64
  PARAMETER HW_VER = 1.00.b
  PARAMETER C_BASEADDR = 0xfffe0000
  PARAMETER C_HIGHADDR = 0xffffffff
  BUS_INTERFACE SPLB = plb_v46_0
  BUS_INTERFACE PORTA = xps_bram_if_cntlr_0_port
END

BEGIN xps_uartlite
  PARAMETER INSTANCE = RS232_Uart_1
  PARAMETER C_BAUDRATE = 9600
  PARAMETER C_DATA_BITS = 8
  PARAMETER C_USE_PARITY = 0
  PARAMETER C_ODD_PARITY = 0
  PARAMETER HW_VER = 1.01.a
  PARAMETER C_BASEADDR = 0x84000000
  PARAMETER C_HIGHADDR = 0x8400ffff
  BUS_INTERFACE SPLB = plb_v46_0
  PORT RX = fpga_0_RS232_Uart_1_RX_pin
  PORT TX = fpga_0_RS232_Uart_1_TX_pin
END

BEGIN xps_gpio
  PARAMETER INSTANCE = LEDs_8Bit
  PARAMETER C_ALL_INPUTS = 0
  PARAMETER C_GPIO_WIDTH = 8
  PARAMETER C_INTERRUPT_PRESENT = 0
  PARAMETER C_IS_DUAL = 0
  PARAMETER HW_VER = 2.00.a
  PARAMETER C_BASEADDR = 0x08140000
  PARAMETER C_HIGHADDR = 0x0814ffff
  BUS_INTERFACE SPLB = plb_v46_0
  PORT GPIO_IO = fpga_0_LEDs_8Bit_GPIO_IO_pin
END

BEGIN bram_block
  PARAMETER INSTANCE = xps_bram_if_cntlr_0_block
  PARAMETER HW_VER = 1.00.a
  BUS_INTERFACE PORTA = xps_bram_if_cntlr_0_port
END

BEGIN clock_generator
  PARAMETER INSTANCE = clock_generator_0
  PARAMETER C_EXT_RESET_HIGH = 0
  PARAMETER C_CLKIN_FREQ = 100000000
  PARAMETER C_CLKOUT0_FREQ = 100000000
  PARAMETER C_CLKOUT0_PHASE = 0
  PARAMETER C_CLKOUT0_GROUP = PLL0_ADJUST
  PARAMETER C_CLKOUT0_BUF = TRUE
```

```
PARAMETER C_CLKOUT1_FREQ = 200000000
PARAMETER C_CLKOUT1_PHASE = 0
PARAMETER C_CLKOUT1_GROUP = PLL0
PARAMETER C_CLKOUT1_BUF = TRUE
PARAMETER C_CLKOUT2_FREQ = 400000000
PARAMETER C_CLKOUT2_PHASE = 0
PARAMETER C_CLKOUT2_GROUP = PLL0
PARAMETER C_CLKOUT2_BUF = TRUE
PARAMETER HW_VER = 3.02.a
PORT CLKIN = dcm_clk_s
PORT CLKOUT0 = clk_100_0000MHzPLL0_ADJUST
PORT CLKOUT1 = clk_200_0000MHzPLL0
PORT CLKOUT2 = clk_400_0000MHzPLL0
PORT RST = sys_rst_s
PORT LOCKED = Dcm_all_locked
END

BEGIN jtagppc_cntlr
PARAMETER INSTANCE = jtagppc_cntlr_inst
PARAMETER HW_VER = 2.01.c
BUS_INTERFACE JTAGPPC0 = ppc440_0_jtagppc_bus
END

BEGIN proc_sys_reset
PARAMETER INSTANCE = proc_sys_reset_0
PARAMETER C_EXT_RESET_HIGH = 0
PARAMETER HW_VER = 2.00.a
BUS_INTERFACE RESETPPC0 = ppc_reset_bus
PORT Slowest_sync_clk = clk_100_0000MHzPLL0_ADJUST
PORT Ext_Reset_In = sys_rst_s
PORT Dcm_locked = Dcm_all_locked
PORT Bus_Struct_Reset = sys_bus_reset
END

BEGIN xps_gpio
PARAMETER INSTANCE = xps_gpio_0
PARAMETER HW_VER = 2.00.a
PARAMETER C_GPIO_WIDTH = 7
PARAMETER C_BASEADDR = 0x81440000
PARAMETER C_HIGHADDR = 0x8144ffff
BUS_INTERFACE SPLB = plb_v46_0
PORT GPIO_IO = xps_gpio_0_GPIO_IO_pin
END

BEGIN xps_spi
PARAMETER INSTANCE = xps_spi_0
PARAMETER HW_VER = 2.01.b
PARAMETER C_FIFO_EXIST = 0
PARAMETER C_NUM_TRANSFER_BITS = 8
PARAMETER C_NUM_SS_BITS = 4
PARAMETER C_BASEADDR = 0x81480000
PARAMETER C_HIGHADDR = 0x81487fff
BUS_INTERFACE SPLB = plb_v46_0
PORT IP2INTC_Irpt = xps_spi_0_IP2INTC_Irpt
PORT SCK = xps_spi_0_SCK
PORT MISO = xps_spi_0_MISO
PORT SS_0 = xps_spi_0_SS_0
PORT SS = xps_spi_0_SS
END
```

## 12.9 Código del archivo MSS

```
PARAMETER VERSION = 2.2.0

BEGIN OS
  PARAMETER OS_NAME = standalone
  PARAMETER OS_VER = 2.00.a
  PARAMETER PROC_INSTANCE = ppc440_0
  PARAMETER stdin = RS232_Uart_1
  PARAMETER stdout = RS232_Uart_1
END

BEGIN PROCESSOR
  PARAMETER DRIVER_NAME = cpu_ppc440
  PARAMETER DRIVER_VER = 1.01.a
  PARAMETER HW_INSTANCE = ppc440_0
  PARAMETER COMPILER = powerpc-eabi-gcc
  PARAMETER ARCHIVER = powerpc-eabi-ar
END

BEGIN DRIVER
  PARAMETER DRIVER_NAME = bram
  PARAMETER DRIVER_VER = 1.00.a
  PARAMETER HW_INSTANCE = xps_bram_if_cntlr_0
END

BEGIN DRIVER
  PARAMETER DRIVER_NAME = uartlite
  PARAMETER DRIVER_VER = 1.14.a
  PARAMETER HW_INSTANCE = RS232_Uart_1
END

BEGIN DRIVER
  PARAMETER DRIVER_NAME = gpio
  PARAMETER DRIVER_VER = 2.13.a
  PARAMETER HW_INSTANCE = LEDs_8Bit
END

BEGIN DRIVER
  PARAMETER DRIVER_NAME = generic
  PARAMETER DRIVER_VER = 1.00.a
  PARAMETER HW_INSTANCE = xps_bram_if_cntlr_0_block
END

BEGIN DRIVER
  PARAMETER DRIVER_NAME = generic
  PARAMETER DRIVER_VER = 1.00.a
  PARAMETER HW_INSTANCE = clock_generator_0
END

BEGIN DRIVER
  PARAMETER DRIVER_NAME = generic
  PARAMETER DRIVER_VER = 1.00.a
  PARAMETER HW_INSTANCE = jtagppc_cntlr_inst
END
```

```
BEGIN DRIVER
  PARAMETER DRIVER_NAME = generic
  PARAMETER DRIVER_VER = 1.00.a
  PARAMETER HW_INSTANCE = proc_sys_reset_0
END
```

```
BEGIN DRIVER
  PARAMETER DRIVER_NAME = gpio
  PARAMETER DRIVER_VER = 2.13.a
  PARAMETER HW_INSTANCE = xps_gpio_0
END
```

```
BEGIN DRIVER
  PARAMETER DRIVER_NAME = spi
  PARAMETER DRIVER_VER = 2.01.b
  PARAMETER HW_INSTANCE = xps_spi_0
END
```

```
BEGIN DRIVER
  PARAMETER DRIVER_NAME = generic
  PARAMETER DRIVER_VER = 1.00.a
  PARAMETER HW_INSTANCE = plb_v46_0
END
```

## 12.10 Código del archivo \*.UCF

```
# Virtex-5 ML507 Evaluation Platform
Net fpga_0_RS232_Uart_1_RX_pin LOC = AG15 | IOSTANDARD=LVCMOS33;
Net fpga_0_RS232_Uart_1_TX_pin LOC = AG20 | IOSTANDARD=LVCMOS33;
Net fpga_0_LEDs_8Bit_GPIO_IO_pin<0> LOC = AE24 | IOSTANDARD=LVCMOS18 |
PULLDOWN | SLEW=SLOW | DRIVE=2;
Net fpga_0_LEDs_8Bit_GPIO_IO_pin<1> LOC = AD24 | IOSTANDARD=LVCMOS18 |
PULLDOWN | SLEW=SLOW | DRIVE=2;
Net fpga_0_LEDs_8Bit_GPIO_IO_pin<2> LOC = AD25 | IOSTANDARD=LVCMOS18 |
PULLDOWN | SLEW=SLOW | DRIVE=2;
Net fpga_0_LEDs_8Bit_GPIO_IO_pin<3> LOC = G16 | IOSTANDARD=LVCMOS25 |
PULLDOWN | SLEW=SLOW | DRIVE=2;
Net fpga_0_LEDs_8Bit_GPIO_IO_pin<4> LOC = AD26 | IOSTANDARD=LVCMOS18 |
PULLDOWN | SLEW=SLOW | DRIVE=2;
Net fpga_0_LEDs_8Bit_GPIO_IO_pin<5> LOC = G15 | IOSTANDARD=LVCMOS25 |
PULLDOWN | SLEW=SLOW | DRIVE=2;
Net fpga_0_LEDs_8Bit_GPIO_IO_pin<6> LOC = L18 | IOSTANDARD=LVCMOS25 |
PULLDOWN | SLEW=SLOW | DRIVE=2;
Net fpga_0_LEDs_8Bit_GPIO_IO_pin<7> LOC = H18 | IOSTANDARD=LVCMOS25 |
PULLDOWN | SLEW=SLOW | DRIVE=2;
Net fpga_0_clk_1_sys_clk_pin TNM_NET = sys_clk_pin;
TIMESPEC TS_sys_clk_pin = PERIOD sys_clk_pin 100000 kHz;
Net fpga_0_clk_1_sys_clk_pin LOC = AH15 | IOSTANDARD=LVCMOS33;
Net fpga_0_rst_1_sys_rst_pin TIG;
Net fpga_0_rst_1_sys_rst_pin LOC = E9 | IOSTANDARD=LVCMOS33 | PULLUP;

Net xps_gpio_0_GPIO_IO_pin<0> LOC=AN33 | IOSTANDARD=LVCMOS33;#pinJ6 64

Net xps_spi_0_MISO LOC = H33 | IOSTANDARD=LVCMOS33; #pinJ6 2 S0
Net xps_spi_0_SCK LOC = H32 | IOSTANDARD=LVCMOS33; #pinJ6 12 CLK

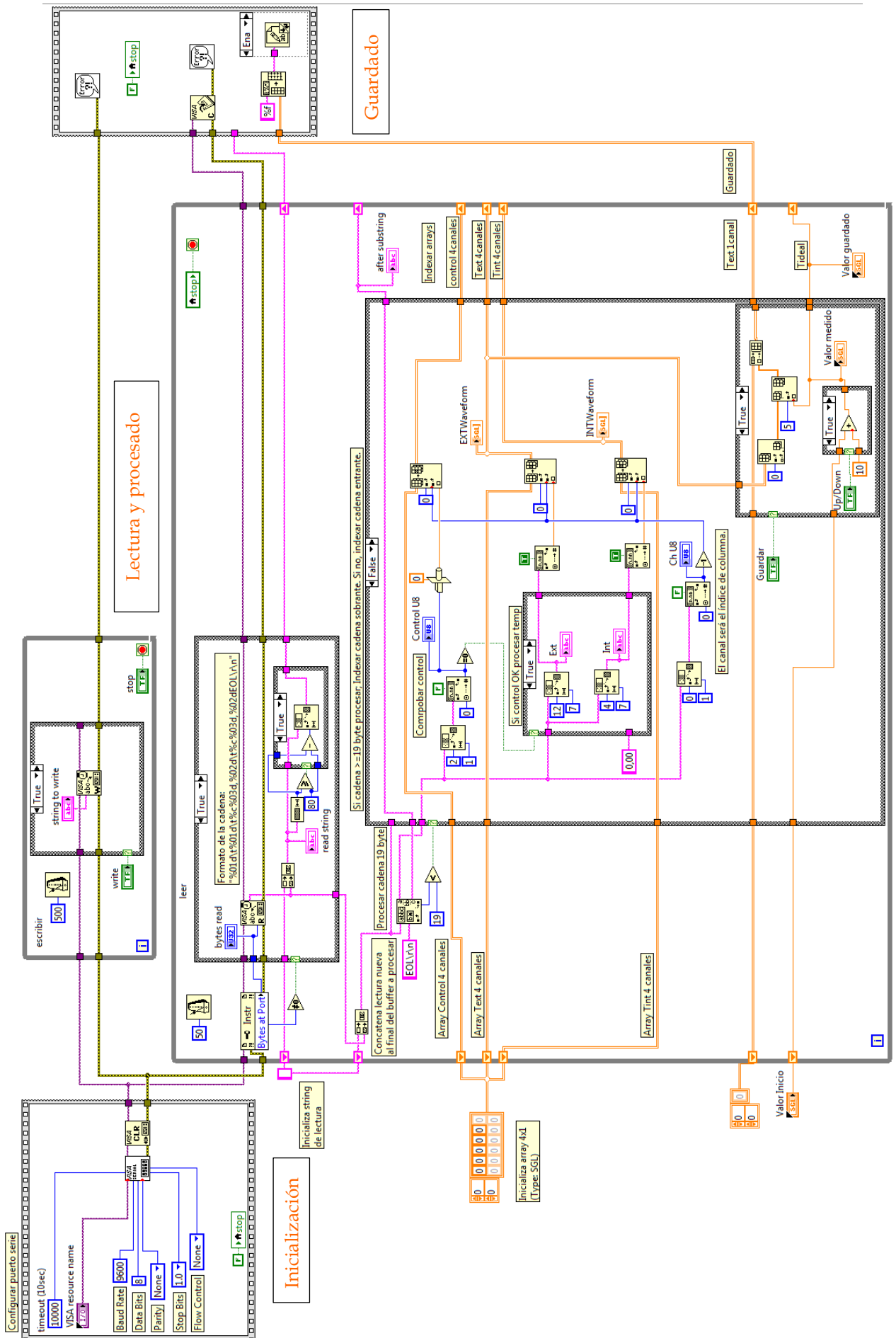
Net xps_spi_0_SS_0_pin<0> LOC = F34 | IOSTANDARD=LVCMOS33; #pinJ6 4 /CS1
Net xps_spi_0_SS_0_pin<1> LOC = H34 | IOSTANDARD=LVCMOS33; #pinJ6 6
/CS2
Net xps_spi_0_SS_0_pin<2> LOC = G33 | IOSTANDARD=LVCMOS33; #pinJ6 8
/CS3
Net xps_spi_0_SS_0_pin<3> LOC = G32 | IOSTANDARD=LVCMOS33; #pinJ6 10 /CS4

NET "clk_100_000MHzPLL0_ADJUST" TNM_NET = "splb_clk";
TIMESPEC "TS_splb_clk" = PERIOD "splb_clk" 200 ns HIGH 50%; #reloj MAX31855
max 5MHz
```

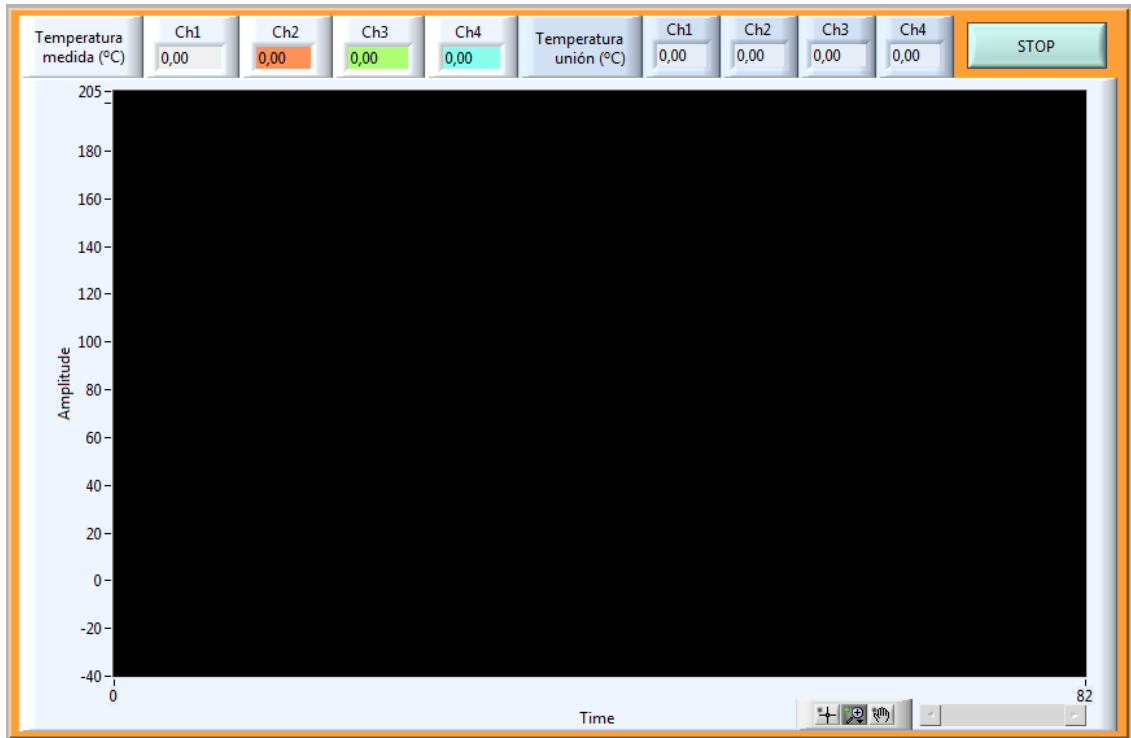
## **12.11 Código LabVIEW**

En las siguientes páginas se presenta el código en LabVIEW completo descrito en el apartado 8.2, así como los paneles frontales tanto de Interfaz de usuario básico con la visualización de las temperaturas en tiempo real, como la de usuario avanzado que incluye herramientas para realizar las medidas de calibración y ajuste del sistema.

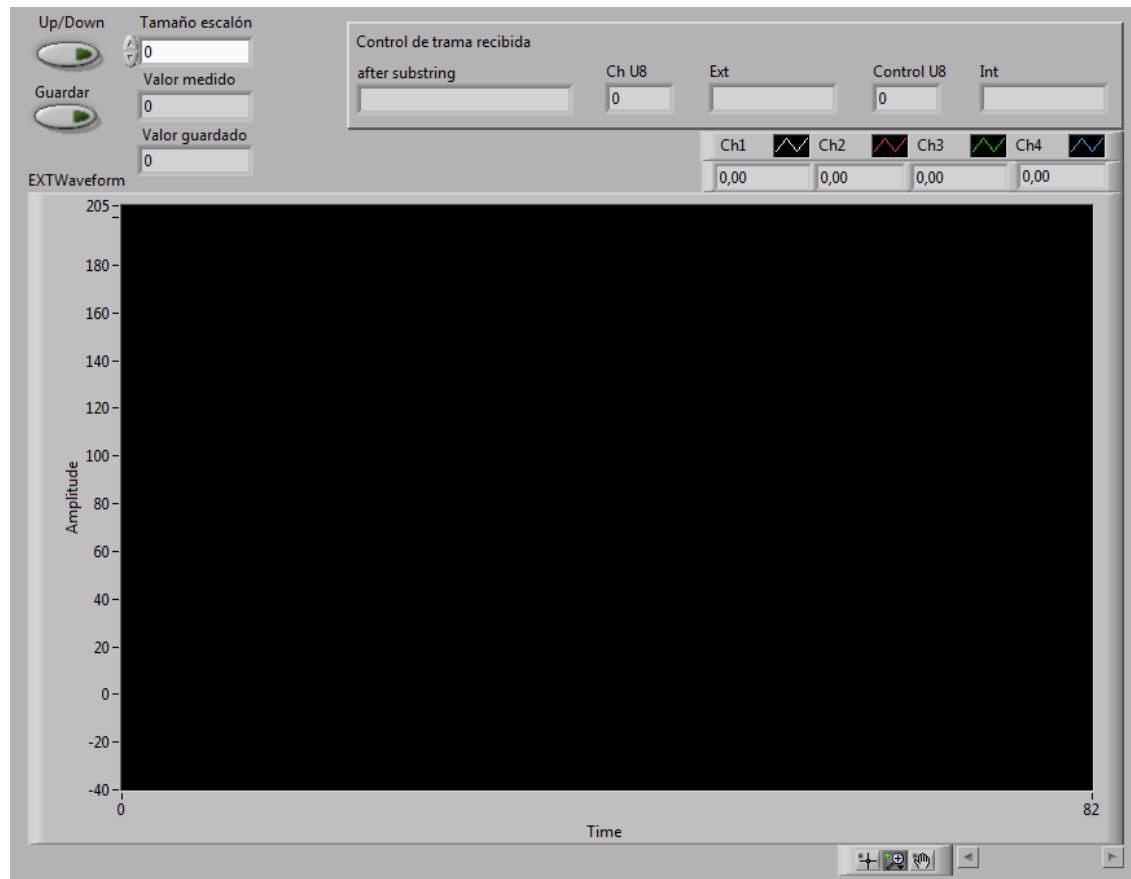




**Panel frontal básico de usuario**



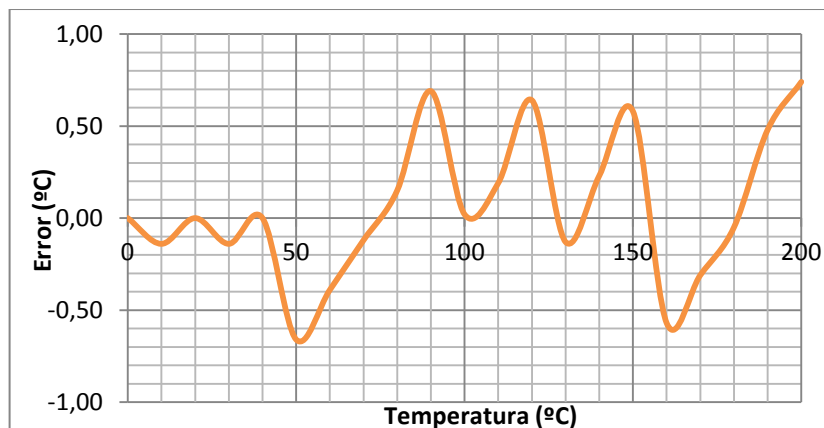
**Panel frontal de calibración**



## 12.12 Cálculos y gráficas en Excel de los cuatro canales

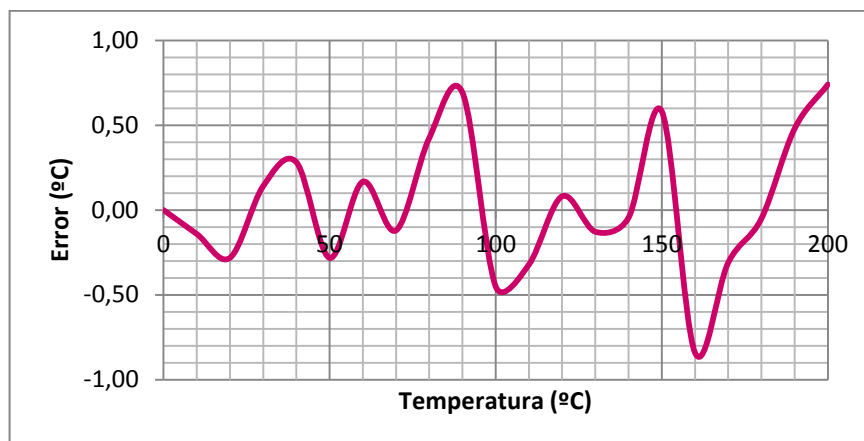
### Canal 1

Ideal	Medida	Ganancia aplicada	Corrección Ganancia	Corrección Offset	Error final
0	3,00	1,13	3,38	0,00	0,00
10	11,75	1,13	13,24	9,86	-0,14
20	20,75	1,13	23,38	20,00	0,00
30	29,50	1,13	33,24	29,86	-0,14
40	38,50	1,13	43,38	40,00	0,00
50	47,50	1,11	52,73	49,34	-0,66
60	56,75	1,11	62,99	59,61	-0,39
70	66,00	1,11	73,26	69,88	-0,12
80	75,25	1,11	83,53	80,15	0,15
90	84,75	1,11	94,07	90,69	0,69
100	94,00	1,10	103,40	100,02	0,02
110	103,25	1,10	113,58	110,19	0,19
120	112,75	1,10	124,03	120,64	0,64
130	122,25	1,09	133,25	129,87	-0,13
140	131,75	1,09	143,61	140,23	0,23
150	141,25	1,09	153,96	150,58	0,58
160	150,75	1,08	162,81	159,43	-0,57
170	160,25	1,08	173,07	169,69	-0,31
180	169,75	1,08	183,33	179,95	-0,05
190	179,50	1,08	193,86	190,48	0,48
200	189,00	1,08	204,12	200,74	0,74



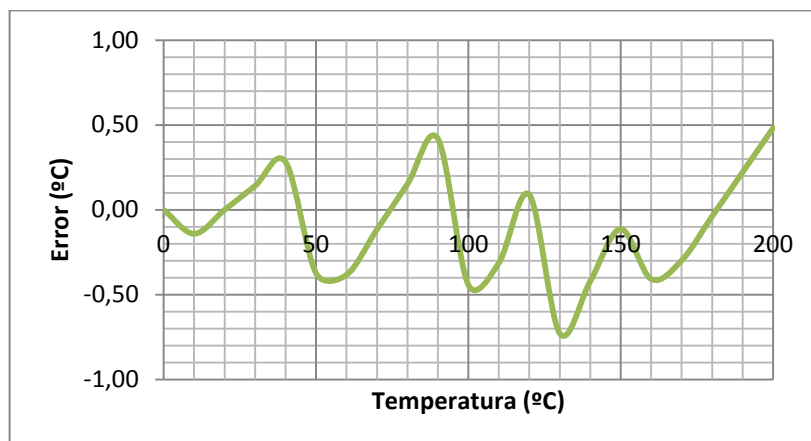
## Canal 2

Ideal	Medida	Ganancia aplicada	Corrección Ganancia	Corrección Offset	Error final
0	3,00	1,13	3,38	0,00	0,00
10	11,75	1,13	13,24	9,86	-0,14
20	20,50	1,13	23,10	19,72	-0,28
30	29,75	1,13	33,52	30,14	0,14
40	38,75	1,13	43,66	40,28	0,28
50	47,75	1,11	53,10	49,72	-0,28
60	57,25	1,11	63,55	60,17	0,17
70	66,00	1,11	73,26	69,88	-0,12
80	75,50	1,11	83,81	80,42	0,42
90	84,75	1,11	94,07	90,69	0,69
100	94,00	1,10	102,93	99,55	-0,45
110	103,25	1,10	113,06	109,68	-0,32
120	112,75	1,10	123,46	120,08	0,08
130	122,25	1,09	133,25	129,87	-0,13
140	131,50	1,09	143,34	139,95	-0,05
150	141,25	1,09	153,96	150,58	0,58
160	150,50	1,08	162,54	159,16	-0,84
170	160,25	1,08	173,07	169,69	-0,31
180	169,75	1,08	183,33	179,95	-0,05
190	179,50	1,08	193,86	190,48	0,48
200	189,00	1,08	204,12	200,74	0,74



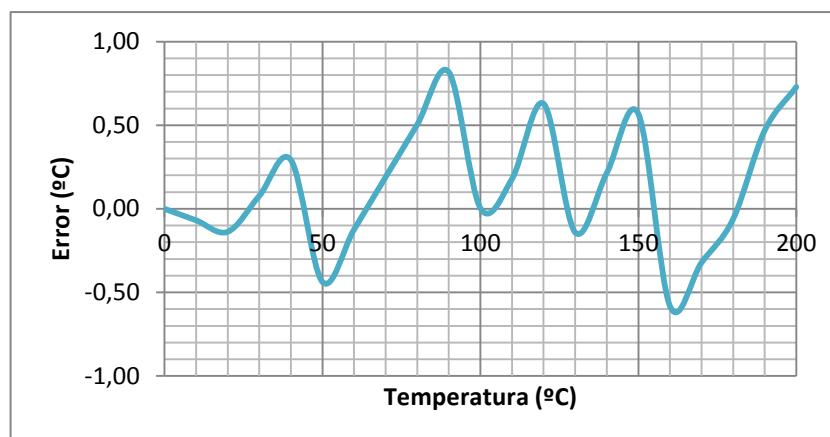
## Canal3

Ideal	Medida	Ganancia aplicada	Corrección Ganancia	Corrección Offset	Error final
0	2,75	1,13	3,10	0,00	0,00
10	11,50	1,13	12,96	9,86	-0,14
20	20,50	1,13	23,10	20,00	0,00
30	29,50	1,13	33,24	30,14	0,14
40	38,50	1,13	43,38	40,28	0,28
50	47,50	1,11	52,73	49,63	-0,37
60	56,50	1,11	62,72	59,62	-0,38
70	65,75	1,11	72,98	69,88	-0,12
80	75,00	1,11	83,25	80,15	0,15
90	84,25	1,11	93,52	90,42	0,42
100	93,75	1,10	102,66	99,56	-0,44
110	103,00	1,10	112,79	109,69	-0,31
120	112,50	1,10	123,19	120,09	0,09
130	122,00	1,09	132,37	129,27	-0,73
140	131,50	1,09	142,68	139,58	-0,42
150	141,00	1,09	152,99	149,89	-0,11
160	150,50	1,08	162,69	159,59	-0,41
170	160,00	1,08	172,80	169,70	-0,30
180	169,50	1,08	183,06	179,96	-0,04
190	179,00	1,08	193,32	190,22	0,22
200	188,50	1,08	203,58	200,48	0,48



## Canal4

Ideal	Medida	Ganancia aplicada	Corrección Ganancia	Corrección Offset	Error final
0	2,75	1,14	3,12	0,00	0,00
10	11,50	1,14	13,05	9,93	-0,07
20	20,25	1,14	22,98	19,86	-0,14
30	29,25	1,14	33,20	30,08	0,08
40	38,25	1,14	43,41	40,29	0,29
50	47,25	1,12	52,68	49,56	-0,44
60	56,50	1,12	63,00	59,88	-0,12
70	65,75	1,12	73,31	70,19	0,19
80	75,00	1,12	83,63	80,50	0,50
90	84,25	1,12	93,94	90,82	0,82
100	93,75	1,10	103,13	100,00	0,00
110	103,00	1,10	113,30	110,18	0,18
120	112,50	1,10	123,75	120,63	0,63
130	122,00	1,09	132,98	129,86	-0,14
140	131,50	1,09	143,34	140,21	0,21
150	141,00	1,09	153,69	150,57	0,57
160	150,50	1,08	162,54	159,42	-0,58
170	160,00	1,08	172,80	169,68	-0,32
180	169,50	1,08	183,06	179,94	-0,06
190	179,25	1,08	193,59	190,47	0,47
200	188,75	1,08	203,85	200,73	0,73



# Bibliografía

---

Xilinx. <http://www.xilinx.com/>

Xilinx EDK.

[http://www.xilinx.com/support/documentation/sw\\_manuals/xilinx11/edk\\_ctt.pdf](http://www.xilinx.com/support/documentation/sw_manuals/xilinx11/edk_ctt.pdf)

Xilinx ISE in-depth Tutorial.

[http://www.xilinx.com/support/documentation/sw\\_manuals/xilinx11/ise11tut.pdf](http://www.xilinx.com/support/documentation/sw_manuals/xilinx11/ise11tut.pdf)

Xilinx Drivers. [http://www.xilinx.com/ise/embedded/xilinx\\_drivers.pdf](http://www.xilinx.com/ise/embedded/xilinx_drivers.pdf)

Maxim MAX31855 Datasheet. [http://www.xilinx.com/ise/embedded/xilinx\\_drivers.pdf](http://www.xilinx.com/ise/embedded/xilinx_drivers.pdf)

National Instruments LabVIEW. <http://www.ni.com/labview/whatis/esa/>

National Instruments Termopares. <http://www.ni.com/white-paper/7108/es>

José Rafael Lajara “LabVIEW: Entorno gráfico de programación” National Instruments. 2ª Edición.

Hassan Hassan, Mohab Anis “Low-Power Design of Nanometer FPGAs: Architecture and EDA”MK

“The theory and properties of Thermocouple Elements.” American society for testing and materials.

“Manual on the Use of Thermocouples in Temperature Measurement” ASTM Committee E20 on Temperature Measurement. 4ª Edición.