



Optimización de una tarjeta de adquisición de datos



Departamento de
Electrónica y
Tecnología de los
Computadores

Pablo Barroso González

Tutores:

Miguel Ángel Carvajal Rodríguez
María Sofía Martínez García

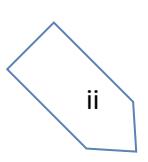
Escuela Técnica
Superior de
Ingeniería
Informática y
Telecomunicaciones

Universidad de
Granada



ugr

Universidad
de Granada





ugr

Universidad
de Granada

TRABAJO FIN DE GRADO EN EL GRADO DE INGENIERÍA DE
TECNOLOGÍAS DE LA TELECOMUNICACIÓN

OPTIMIZACIÓN DE UNA TARJETA DE ADQUISICIÓN DE DATOS

REALIZADO POR:

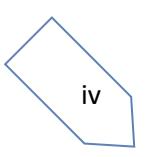
Pablo Barroso González

TUTORIZADO POR:

Miguel Ángel Carvajal Rodríguez y María Sofía Martínez García

DEPARTAMENTO:

Electrónica y Tecnología de los Computadores



iv



ugr | Universidad
de Granada

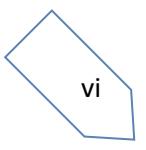
AUTORIZACIÓN DE LECTURA

D. Miguel Ángel Carvajal Rodríguez, profesor del Departamento de Electrónica y Tecnología de los Computadores de la Universidad de Granada y Dña. María Sofía Martínez García, doctorada del Departamento de Electrónica y Tecnología de los Computadores de la Universidad de Granada, como directores del Trabajo Fin de Grado de D. Pablo Barroso González, informan que: el presente trabajo, titulado "**Optimización de una tarjeta de adquisición de datos**" ha sido redactado por el mencionado alumno bajo nuestra dirección, y con esta fecha autorizamos su presentación.

Granada, a 3 de Julio de 2015

Fdo. Miguel Ángel Carvajal Rodríguez

Fdo. María Sofía Martínez García



Optimización de una tarjeta de adquisición

Pablo Barroso González

Palabras claves

Adquisición, muestreo, acondicionamiento, convertidor analógico-digital, microcontrolador, SMU, USB, hardware, firmware, software.

Resumen

En esta memoria se detalla el proceso que se ha seguido para lograr optimizar una tarjeta de adquisición de datos. En primer lugar, se comentarán las razones de me ha llevado para realizar este proyecto y los objetivos que se persiguen. Después se detallarán todas las partes del proyecto, etapas de alimentación, etapas de acondicionamiento, además de las mejoras realizadas como las SMUs incluidas, aumento de la frecuencia de muestreo, cambiar la estrategia de filtrado, etc. Para así poder explicar en profundidad la parte de diseño e implementación del hardware, el código del microcontrolador y el software de control de la tarjeta.

Por último, se expondrán una serie de medidas que se han obtenido usando la tarjeta de adquisición. Se terminaron las conclusiones obtenidas al realizar este proyecto y las líneas futuras con las que se podría continuar.

Keywords

Acquisition, sampling, conditioning, analog-to-digital converter, microcontroller, SMU, USB, hardware, firmware, software.

Abstract

The present text details the followed process to optimize a Signal acquisition card. Firstly, I will discuss the reasons for this project and the objectives I wanted to achieve. Then I will explain in detail every part of my project and its components, like, power stages or conditioning stages. In addition, I am going to explain the improvements such as increasing the sampling rate, the introduction of SMUs, filtering. After this, I will describe deeply the hardware part, the microcontroller code and the control's software of the card.

Finally, I will show and explain some measures obtained using my Signal acquisition card. I will conclude with my personal conclusion that I have gotten doing this project and a series of possible researches and improvements.

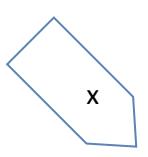
AGRADECIMIENTOS

En primer lugar dar las gracias a mis tutores Miguel Ángel Carvajal y Sofía Martínez, por su buen trato recibido y su paciencia mostrada hacia mí. No creo que haya podido tener unos tutores mejores que ellos dos, y me llevo muchas cosas aprendidas gracias a ellos.

En segundo lugar a todos mis amigos, ya sean de mi pueblo, como Rafa, Aís, Antonio, Oscar, Quiles, o a Pedro, José o Garrido, que los he conocido en la carrera. A todos los que han estado conmigo aquí en Granada. Pero también los que han estado fuera de Granada estudiando en distintas ciudades como Masegosa, Dani o Pilar. Gracias por estar a mi lado, y saber que no olvidaré los años que hemos pasado en la universidad, tanto cuando íbamos a celebrar cualquier cosa, como cuando estábamos todos en la biblioteca estudiando para los exámenes.

Por último, a toda mi familia, pero especialmente a mi padre, mi madre y mi hermano. Que me han demostrado que puedo contar con ellos para lo que sea. No solo el apoyo recibido durante la carrera, si no en mis momentos más bajos a nivel personal se han portado realmente bien.

Recordar a los que no están con nosotros, a mi abuela Lucía que siempre me preguntaba que si mi carrera me iba hacer ganar mucho dinero, pero en especial a mi abuelo Manolo, que fue el que me hizo del Betis y era con el que siempre aguantaba los palos que nos daban cuando baja a 2º división. Pero sé que estaría muy orgulloso de mí, y en la persona que me he convertido. Un beso muy fuerte a los dos.



INDICE

1. Introducción	1
1.1 Introducción	2
1.2 Motivación.....	4
1.3 Objetivos.....	5
2. Estado del arte	7
2.1 Introducción histórica.....	8
2.2 Estado del arte.....	8
2.2.1 Sistemas DAQ	9
2.2.2 Sistemas modulares.....	9
2.2.3 El mercado actual	10
3. Descripción del sistema.....	13
3.1 Alimentación del sistema	14
3.1.1 MAX743	15
3.1.2 Regulador de tensión LM7912	15
3.1.3 Regulador de tensión LM7812	16
3.1.4 Referencia de tensión LM4040.....	16
3.2 Acondicionamiento de la señal	17
3.3 Multiplexor ADG408.....	19
3.4 Microcontrolador PIC18F4553.	20
3.4.1 Convertidor analógico- digital.	20
3.4.2 Oscilador	23
3.4.3 Conexión USB	23
3.5 SMU	24
3.5.1 Generación de la señal PWM	24
3.5.2 Configuración del módulo CCP para generar señales PWM.....	25
3.5.3 Filtrado de la señal.....	27
3.5.4 Etapa DE ACONDICIONAMIENTO SEÑAL PWM	28
3.5.5 Funcionamiento del SMU	30
4. Diseño de la placa de circuito impreso	33
4.1 Introducción	34
4.2 Software de diseño Altium	34
4.2.1 Esquemático de Altium.....	35
4.2.2 PCB de Altium	38
4.3 Fabricación de la PCB	40
4.3.1 Tipos de procesos de fabricación de una PCB	40
4.3.2 Proceso de fabricación de la PCB	41
4.3.3 Resultados de la PCB terminada.....	41
5. Firmware	43
5.1 Entorno de programación	45
5.2 Firmware del PIC18F4553.....	46
5.2.1 Cabecera y preprocessamiento de datos.....	48
5.2.2 Funciones del programa	49
5.2.3 Función main	49
5.3 Programación del PIC18F4553	52
6. Aplicación de gestión	56
6.1 Software LabVIEW	57
6.2 Aplicación de gestión del PIC.....	58
6.2.1 Panel frontal	60

6.2.2 Diagrama de bloques.....	60
6.2.2.1 Comunicación con el microcontrolador	61
6.2.2.2 Generación del mensaje para el microcontrolador.....	62
6.2.2.3 Tratamiento de los datos recibidos	63
6.2.2.3 Escritura de los datos finales	64
7. Especificaciones y manual de usuario	68
7.1 Especificaciones.....	69
7.2 Manual del usuario.....	69
7.2.1 Conexión física entre la placa y las señales a medir	69
7.2.2 Configurar la aplicación de gestión	70
7.2.3 Configuración del retardo.....	71
7.2.4 Configuración de las tensiones de la SMU	72
8. Resultados obtenidos	74
8.1 Resultados obtenidos	75
8.1.1 Experiencia 1: Medir tensión en continua.....	75
8.1.2 Experiencia 2: Medir sensores de luz y temperatura	75
8.1.3 Experiencia 3: Medir señales alternas.....	77
8.1.4 Experiencia 4: Utilización del SMU	79
9. Conclusiones.....	82
9. 1 Conclusiones.....	83
9.2 Líneas futuras	83
Apéndice 1: Presupuesto	86
Apéndice 2: Diagrama de bloques de LabVIEW	90
Apéndice 3: Esquemáticos del proyecto	94
Apéndice 4: Esquema de Proteus	98
Apéndice 5: Datos de las medias	102
Bibliografía	109
Índice de acrónimos	111

INDICE DE ILUSTRACIONES

Ilustración 1.1: Componentes de un sistema generalizado de medida	3
Tabla 1.1: Diferentes tipos de transductores.....	3
Ilustración 1.2: Esquema de conversión analógico-digital.....	4
Ilustración 2.1: IBM 7700 Data Acquisition System	8
Ilustración 2.2: IBM 1800 Data Acquisition and Control System (DACS)	8
Ilustración 2.3: Partes de un sistema DAQ.....	9
Ilustración 2.4: Algunos módulos del fabricante National Instruments.	10
Ilustración 2.5: National Instruments USB-6000.....	11
Ilustración 3.1: Esquema general del dispositivo.....	14
Ilustración 3.2: Patillaje y circuito del MAX743.....	15
Ilustración 3.3: Patillaje y circuito del LM79L12.....	16
Ilustración 3.4: Patillaje y circuito del LM78L12.....	16
Ilustración 3.5: Patillaje y circuito del LM4040	17
Ilustración 3.6: Etapa de acondicionamiento para señales más grandes de cero a +5V	17
Ilustración 3.7: Simulación de la etapa de acondicionamiento	18
Ilustración 3.8: Patillaje y esquema interno del integrado ADG408	19
Ilustración 3.9: Patillaje del PIC18F4553	20
Ilustración 3.10: Función de transferencia analógico-digital	21
Ilustración 3.11: Esquema de convertidor analógico-digital.....	22
Ilustración 3.12: Esquema del circuito del oscilador.....	23
Ilustración 3.13: Esquema de la conexión USB	24
Ilustración 3.14: Señal analógica en rosa, señal resultante con la codificación PWM	24
Ilustración 3.15: Esquema interno del módulo CCP del PIC18F4553	26
Ilustración 3.16: Periodo y ciclo de trabajo de una señal PWM	27
Ilustración 3.17: Esquema de un filtro paso bajo con la configuración de Sallen Key	27
Ilustración 3.18: Respuesta en frecuencia de nuestro filtro Sallen Key.....	28
Ilustración 3.19: Etapa de acondicionamiento de la señal PWM.....	29
Ilustración 3.20: Simulación de la etapa de acondicionamiento de la señal PWM	29
Ilustración 3.21: Esquema del circuito SMU	30
Ilustración 4.1: Software utilizado para el diseño de la PCB	34
Ilustración 4.2: Etapa para el acondicionamiento de la señal.....	35
Ilustración 4.3: Etapa de potencia y regulación de la tensión	36
Ilustración 4.4: Esquema del microcontrolador y sus componentes básicos	36
Ilustración 4.5: Etapa de los SMUs.....	37
Ilustración 4.6: Conectores y multiplexor	37
Ilustración 4.7: Esquema de la capa TOP	38
Ilustración 4.8: Esquema de la capa BOTTOM	39
Ilustración 4.9: Imagen del modelo 3D de la PCB	40
Ilustración 4.10: PCB soldada	42
Ilustración 5.1: Esquema completo del programa Proteus.....	44
Ilustración 5.2: Entorno de programación CCS	45
Ilustración 5.3: Compilación de un programa en C	46
Ilustración 5.4: Diagrama de flujo del código del microcontrolador	47
Ilustración 5.5: Cabecera y preprocesado del código	48
Ilustración 5.6: Funciones del firmware.....	49

Ilustración 5.7: Primera parte de la función main.....	50
Ilustración 5.8: Código para obtener una medida	51
Ilustración 5.9: Conexiones básicas para usar el PICKIT 2	52
Ilustración 5.10: PCB con el PICKIT 2.....	53
 Ilustración 6.1: Software LabVIEW, de NI	57
Ilustración 6.2: Diagrama de flujo de la aplicación de control.....	59
Ilustración 6.3: Panel frontal de la aplicación de gestión	60
Ilustración 6.4: Ejemplo de comunicación entre PC y microcontrolador	62
Ilustración 6.5: Descripción del significado de cada bit	62
Ilustración 6.6: Diagrama de la parte encargada de generar el mensaje al microcontrolador ..	63
Ilustración 6.7: Diagrama del tratamiento de los datos recibidos	63
Ilustración 6.8: Diagrama que muestra la creación del archivo.....	64
Ilustración 6.9: Diagrama de bloques del almacenamiento de los datos	65
Ilustración 6.10: Diagrama de bloques del almacenamiento de los datos usando Excel	66
 Ilustración 7.1: Bloques conectores de la PCB con algunos cables.....	70
Ilustración 7.2: Interfaz gráfica para el usuario.....	71
 Ilustración 8.1: Gráfica de la medida del divisor de tensión	75
Ilustración 8.2: Implementación para medir luz y temperatura	76
Ilustración 8.3: Gráfica de la medida de luz y temperatura.....	76
Ilustración 8.4: Algunos datos de la gráfica de los sensores de luz y temperatura	77
Ilustración 8.5: Gráfica del resultado de medir una señal dientes de sierra	78
Ilustración 8.6: Gráfica del resultado de medir una señal triangular.....	78
Ilustración 8.7: Gráfica del resultado de medir una señal sinusoidal	79
Ilustración 8.8: Gráfica del resultado de medir la intensidad Vs la tensión de la resistencia de 321kΩ	80
 Apéndice 1.1 Presupuesto generado por RS.....	88
Apéndice 2.1 Diagrama de bloques encargado de guardar en Excel.....	91
Apéndice 2.2 Diagrama de bloques de la aplicación de gestión.....	92
Apéndice 3.1 Esquemático del proyecto parte 1	95
Apéndice 3.2 Esquemático del proyecto parte 2	96
Apéndice 4.1 Esquemático de Proteus	100

Capítulo 1

Introducción

1.1 Introducción

1.2 Motivación

1.3 Objetivos

1.1 INTRODUCCIÓN

El ser humano ha sido curioso por naturaleza, y por este deseo imperativo de conocer todo lo que nos rodea, se han podido realizar todos los avances científicos a lo largo de nuestra historia.

“Un árbol no siente curiosidad alguna por su medio ambiente, al menos en ninguna forma que podamos reconocer; por su parte, tampoco la sienten una esponja o una ostra. El viento, la lluvia y las corrientes oceánicas les llevan lo que necesitan, y toman de ellos lo que buenamente pueden. Si el azar de los acontecimientos es tal que llega hasta ellos el fuego, el veneno, los depredadores o los parásitos, mueren tan estoica y silenciosamente como vivieron.”

Sin embargo, en el esquema de la vida, algunos organismos no tardaron en desarrollar ciertos movimientos independientes. Esto significó un gran avance en el control de su medio ambiente. Con ello, un organismo móvil no tenía ya por qué esperar largo tiempo, en estólica rigidez, a que los alimentos se cruzaran en su camino, sino que podía salir a buscarlos.”

Isaac Asimov
Introducción a la ciencia, 1972

Es esa curiosidad innata la que nos hace preguntarnos sobre los fenómenos físicos que nos rodean, queriendo entender y saber más sobre ellos. Este es el origen, de que, el ser humano ha creado innumerables mecanismos, patrones, dispositivos, y así un largo etcétera, de maneras de saciar nuestra curiosidad. Actualmente, cuando uno mira a su alrededor, puede apreciar una gran cantidad instrumentos de medida por todas partes. Tal vez el más común para todos sea el reloj, pero abundan por todas partes, como termómetros por la calle, en los salpicaderos del coche, etc.

Se denomina *instrumentación* al conjunto de instrumentos o elementos que hacen posible la medida de una variable física, y por extensión, al conjunto de instrumentos de medida que permiten seguir la evolución de un proceso físico, sin importar el número de variables físicas que toman parte en este sistema.

En nuestro proyecto, cuya intención es la de captación de los parámetros básicos de las señales eléctricas, adquiriremos y almacenaremos datos de las magnitudes físicas que caracterizan a las señales eléctricas, tales como la intensidad de corriente o diferencia de potencial.

A continuación se explica, las partes básicas que consta un sistema de medida:

Optimización de una tarjeta de adquisición de datos

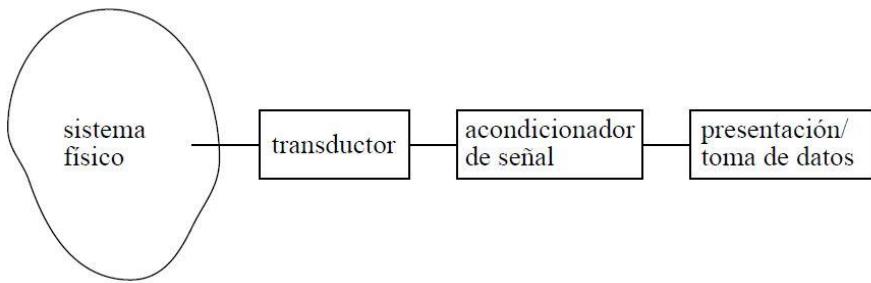


Ilustración 7.1: Componentes de un sistema generalizado de medida

Podemos definir transductor, como cualquier dispositivo que convierte una propiedad o fenómeno físico en una señal eléctrica. En nuestro caso una diferencia de potencial o una intensidad de corriente harán realizaran determinados cambios en los valores de una resistencia o de un condensador.

Transductor	Magnitud física
Fotorresistencia Célula fotovoltaica Fototransistor	Intensidad lumínica
Membranas Piezoelectrónicos Manómetros Digitales	Presión
RTD Termistor Termopar	Temperatura
Galgas extensiométricas Triaxiales	Fuerzas y deformaciones
Sensor final de carrera Sensor capacitivo Sensor inductivo	Distancia (proximidad)

Tabla 1.1: Diferentes tipos de transductores

El acondicionamiento de la señal son todos aquellos elementos del sistema de medida que ofrecen, a partir de la señal del transductor, una señal apta para ser presentada, tratada o almacenada posteriormente. Las acciones más comunes en esta etapa es la filtración, adaptación de impedancias, modulaciones, demodulaciones, etcétera. Por ejemplo, en un sistema de medida de señales cardiovasculares, al tratarse de señales muy débiles, necesitaremos amplificarlas antes de su posterior tratamiento.

En la última parte del sistema, la presentación o toma de datos, nos referimos a todas las acciones que tendremos que realizar para poder pasar de una señal eléctrica, a una serie de datos computables por una CPU. El método que se realiza es la conversión analógica-digital, que no es más que muestrear, cuantificar y codificar la señal en bits. Una vez convertidos a bits, no tendremos ningún problema en operar con ellos utilizando, por ejemplo, un procesador o un controlador. En nuestro proyecto, se utilizaran unos elementos

1. Introducción

llamados ADC (Analog – Digital Converter), elemento muy común que se encuentran en las tarjetas de adquisición de datos.

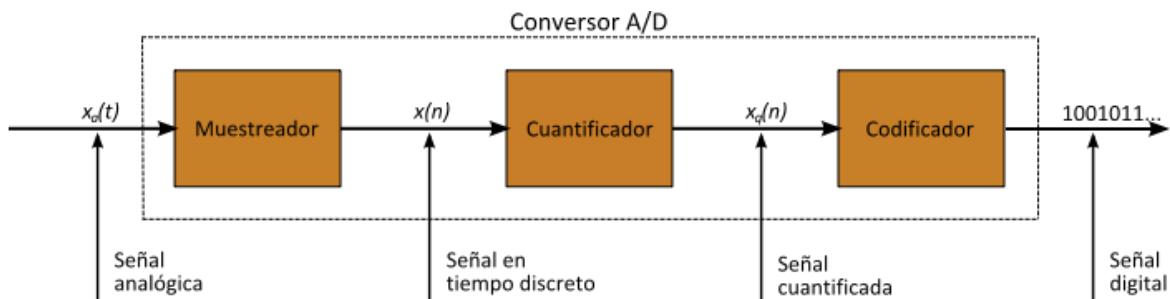


Ilustración 1.8: Esquema de conversión analógico-digital.

Para poder reconstruir la señal, o sea, el proceso inverso, es obligatorio cumplir el teorema de muestreo de Nyquist-Shannon.

“La frecuencia de muestreo debe ser mayor que dos veces el ancho de banda de la señal de entrada, para poder reconstruir la señal original de forma exacta a partir de sus muestras.”

Formulado por Harry Nyquist en 1928
Demostrado por Claude E. Shannon en 1949

1.2 MOTIVACIÓN

La elección de este proyecto la he realizado basándome en la gran variedad de aspectos y desafíos que tiene realizar este proyecto, pues se tiene que diseñar nuevos circuitos analógicos, nuevas etapas, diseñar una nueva PCB, programar nuevamente el PIC para satisfacer la nueva demanda de tareas y sistematizar un nuevo programa en el ordenador que sea capaz de gestionar toda la tarjeta.

Además, se va a continuar un proyecto de una tarjeta de un compañero que solo toma valores de tensión a una frecuencia de 3Hz. Lo que se pretende es aumentar todas estas prestaciones, aumento de la frecuencia de muestreo, introducir canales de salida para alimentar posibles circuitos e incluso poder medir intensidades de corriente, mediante unos SMU. Esto le dará un aspecto más atractivo y versátil, ya que podremos hacer curvas características I/V y monitorizar sensores con tensión y corriente.

Es una oportunidad excelente para ver y aprender, todos los pasos para el diseño y fabricación de un nuevo dispositivo.

1.3 OBJETIVOS

El objetivo de este proyecto es la optimización, en lo máximo posible, de las prestaciones que tenía ya la tarjeta en su origen. Vamos a pasar de solo medir tensiones continuas, a poder medir tensiones, ya sean continuas o alternas, además de medir intensidad de corriente. Tendremos nueve canales destinados a medir tensiones y dos canales con SMUs para medir intensidad de corriente. Además de 3 canales de salida para alimentar los posibles circuitos que se quieran medir.

Aparte de ofrecer más variedad de medidas, se optimizará el tiempo de medida entre muestra y muestra, y la reducción de cálculo y de datos que se genera para realizar las mismas o incluso más acciones que antes.

1. Introducción

Capítulo 2

Estado del arte

2.1 Introducción histórica

2.2 Estado del arte

2.2.1 Sistemas DAQ

2.2.2 Sistemas modulares

2.2.3 El mercado actual

2.1 INTRODUCCIÓN HISTÓRICA

El primer sistema de adquisición de datos, data del año 1963, IBM presentó su primer ordenador especializado en la adquisición de datos, el IBM 7700 Data Acquisition System. Era capaz de recibir datos de 32 canales simultáneamente, procesar y enviar dichos datos hasta 16 canales al mismo tiempo, canales que se conectaban con impresoras, pantallas, unidades de display, etc.

En el 1964, se introdujo el IBM 1800 Data Acquisition and Control System (DACS), y se anunció como un equipo capaz de controlar una cadena de montaje .Y así fue, se utilizó más para el control de procesos industriales como el Alto Horno Nº 5 de los talleres de la Compañía de Aceros Kawasaki [1] [2].

Se puede seguir nombrando una gran cantidad de estos sistemas y de sus innumerables prestaciones y utilizaciones, hasta que en 1974 fueron desbancados por las tarjetas de adquisición de datos Tecmar/ Scientific Solutions, las cuales permitían conectarse mediante un bus a los ordenadores Apple. Cuando en 1981, IBM salió a bolsa con sus ordenadores personales, Tecmar empezó a fabricar tarjetas de expansión para ser instaladas en éstos y poder tener la capacidad de adquirir datos. La primera tarjeta de este tipo fue la LabMaster.

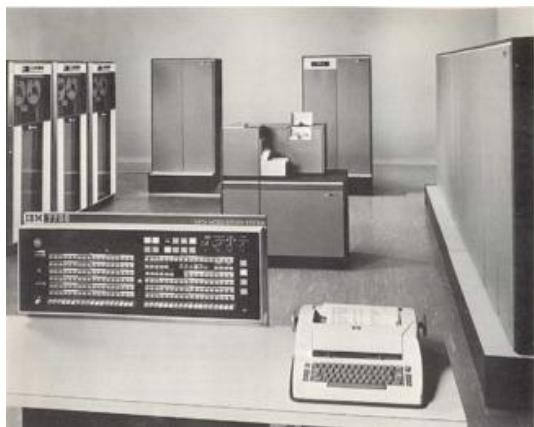


Ilustración 2.1: IBM 7700 Data Acquisition System



Ilustración 2.2: IBM 1800 Data Acquisition and Control System (DACS)

2.2 ESTADO DEL ARTE

Actualmente si uno mira el mercado, puede encontrar una abundante cantidad de sistemas de adquisición de datos, de un gran conjunto de fabricantes, tales como: National Instruments, Agilent Technologies, Tektronix, Omega, etc.

Optimización de una tarjeta de adquisición de datos

Dependiendo de la necesidad de cada usuario, se puede necesitar uno u otro sistema de adquisición de datos considerando varios factores, como por ejemplo el tipo de dato que necesitamos medir. El tipo de dato que vamos a adquirir tiene una variedad enorme, pues podemos medir desde diferencias de potencial, corrientes de intensidad, frecuencia, resistencia, etc. Y esto tan solo en el ámbito electrónico.

Para llevar a cabo estas medidas, en el entorno eléctrico, los productos o dispositivos que se encuentran se pueden clasificar en dos grandes grupos, los dispositivos DAQ (Data Acquisition) y las tarjetas modulares.

2.2.1 SISTEMAS DAQ

Un sistema DAQ, Data Acquisition, consiste en sensores, hardware de medidas DAQ y un PC con software programable. Comparados con los sistemas de medidas tradicionales, los sistemas DAQ basados en PC aprovechan la potencia del procesamiento, la productividad, la visualización y las habilidades de conectividad de los ordenadores estándares en la industria proporcionando una solución de medidas más potente, flexible y rentable [3].



Ilustración 2.3: Partes de un sistema DAQ

Existen una gran variedad de sistemas DAQ, y tal vez se pueda realizar una agrupación de ellos, dependiendo si son portátiles o de escritorio. Que sean de un tipo u otro dependerá, principalmente, de su conectividad. Un sistema portátil, normalmente, se conecta por USB, aunque también puede conectarse vía bluetooth, radio, Ethernet, etc. Sin embargo, con los sistemas de escritorio, necesitaremos habitualmente conectar una tarjeta de expansión a un ordenador de mesa, donde conectaremos los diferentes dispositivos para conectarlo y poder adquirir los datos.

2.2.2 SISTEMAS MODULARES

El origen de los sistemas modulares de adquisición de datos se basa en la utilización de módulos que ejecutan funciones específicas. Compartiendo

2. Estado del arte

recursos en común con el resto del ordenador, como puede ser la pantalla, la fuente de alimentación, algunos buses de información, etc.

Las ventajas que esto conlleva es una mayor flexibilidad al equipo, pues la tarjeta se encarga de adquirir y procesar los datos, quitando esa tarea a la CPU. La integración lo que aporta, ya que estos elementos comparten los recursos que dispone el ordenador, un tamaño más reducido, por lo que podremos usar más cantidad de módulos [4].



Ilustración 2.4: Algunos módulos del fabricante National Instruments

2.2.3 EL MERCADO ACTUAL

Actualmente, si miramos el mercado, podemos encontrar una amplia gama de todo lo anteriormente hablado. La referencia que voy a tomar para este campo es la de National Instruments, una de las marcas más importantes en este sector.

Mirando su extenso catálogo, se encuentra el dispositivo USB-6000, ellos lo definen como “Dispositivo DAQ Multifunción de Bajo Costo para Medidas Básicas de Alta Calidad” y estas son algunas de sus especificaciones más relevantes [5]:

- 8 entradas analógicas, 10 kS/s, resolución de 12 bits
- Rango máximo de voltaje de -10V a +10V
- 4 líneas de E/S digital; un contador de 32 bits
- Ligero y energizado por bus para fácil portabilidad
- Conexión fácil de los señales con conectividad de terminal de tornillo
- Conexión con el PC, mediante USB
- Alimentación por USB, para que sea más portable

Optimización de una tarjeta de adquisición de datos

- Frecuencia máxima de la fuente: 5MHz
- Precio: 95€



Ilustración 2.5: National Instruments USB-6000

El objetivo de hacer este proyecto no es crear un producto que posteriormente se vaya a lanzar al mercado, sino desarrollar un prototipo de tarjeta de adquisición de bajo coste y diseñada según las necesidades de medida. No se pretende igualar las características de las tarjetas de adquisición comerciales, pero sí llegar a un compromiso entre el coste de la misma y funcionalidad, y que pueda ser alimentada mediante un puerto USB 2.0.

2. Estado del arte

Capítulo 3

Descripción del sistema

3.1 Alimentación del sistema

3.1.1 MAX743

3.1.2 Regulador de tensión LM7912

3.1.3 Regulador de tensión LM7812

3.1.4 Referencia de tensión LM4040

3.2 Acondicionamiento de la señal

3.3 Multiplexor ADG408

3.4 Microcontrolador PIC18F4553

3.4.1 Convertidor analógico-digital

3.4.2 Oscilador

3.4.3 Conexión USB

3.5 SMU

3.5.1 Generación de la señal PWM

3.5.2 Configuración del módulo CCP para generar señales PWM

3.5.3 Filtrado de la señal

3.5.4 Etapa DC Shift

3.5.5 Funcionamiento del SMU

3. Descripción del sistema

En este apartado, vamos a tratar de explicar y dar algunos motivos de por qué hemos elegido ciertos componentes en vez de otros, la aplicación que nos va a interesar de ellos, la utilidad de estos respectos a otros, etc.

A continuación se muestra un esquema, para facilitar la compresión de todo el dispositivo, y tener claro todas las etapas que lo componen:

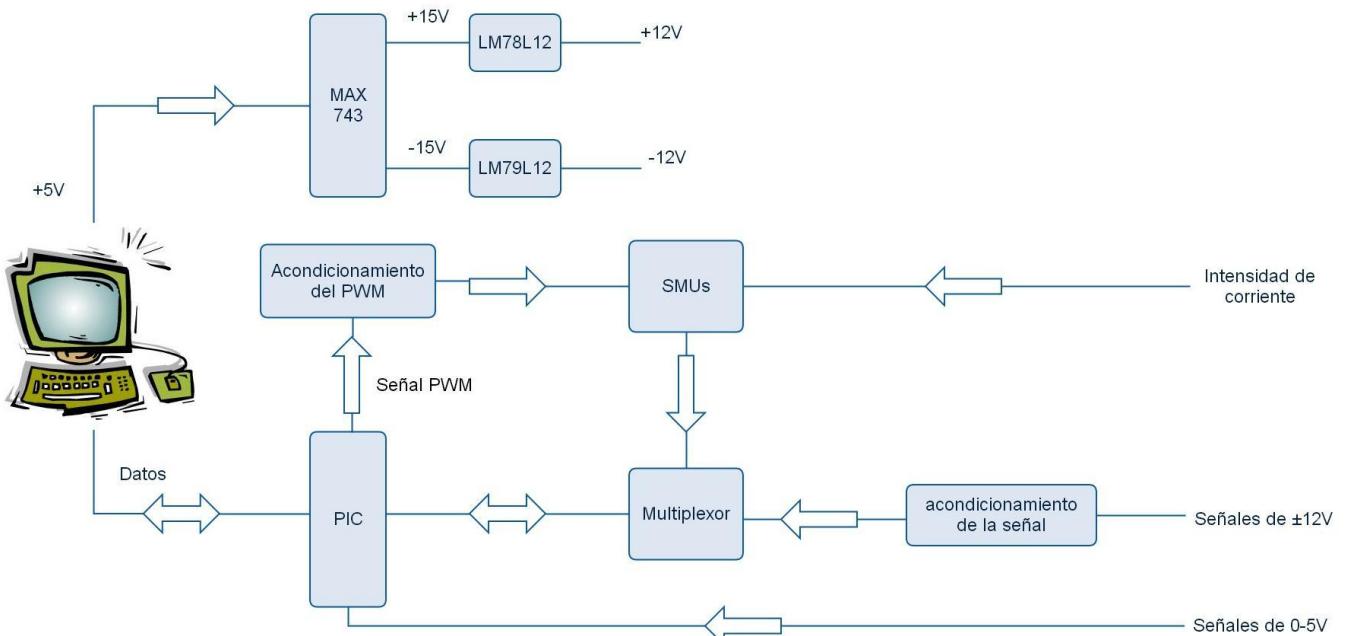


Ilustración 3.1: Esquema general del dispositivo

3.1 ALIMENTACIÓN DEL SISTEMA

Una de las características más prácticas u operativas que tiene nuestra tarjeta de adquisición de datos es, la movilidad que posee. Vamos a utilizar la propia alimentación de la conexión USB, ya que no requerimos de unas necesidades energéticas muy elevadas. Dicha conexión que se utilizará para conectar nuestra tarjeta a un ordenador, que es el encargado de analizar y guardar los datos recibidos.

Un puerto USB consta de cuatro pines, dos de ellos son para la recepción y el envío de datos, y los otros dos para la alimentación. La tensión que proporciona un USB2.0 es de 5V, con una intensidad máxima de 0.5 A, que es más que suficiente para alimentar a toda la tarjeta. [6]

En este apartado vamos a explicar los elementos que se utiliza para alimentar a toda la tarjeta:

Optimización de una tarjeta de adquisición de datos

3.1.1 MAX743

Se trata de un convertidor DC-DC, capaz de darnos a su salida una tensión de +15V y otra de -15V, desviándose, como máximo, un 4% debido a las tensiones, temperaturas, etc. Este dispositivo integrado nos va a ser realmente útil, ya que vamos a pasar de los 5V del USB a $\pm 15V$.

Además ofrece otras características que indirectamente nos pueden llegar a ser muy beneficiosas como la limitación de corriente, aporta bajo ruido al sistema, sistema de autoprotección térmico, arranque suave para cuidar los elementos de la tarjeta y un largo etcétera que hace este dispositivo una buena elección para el proyecto. [7]

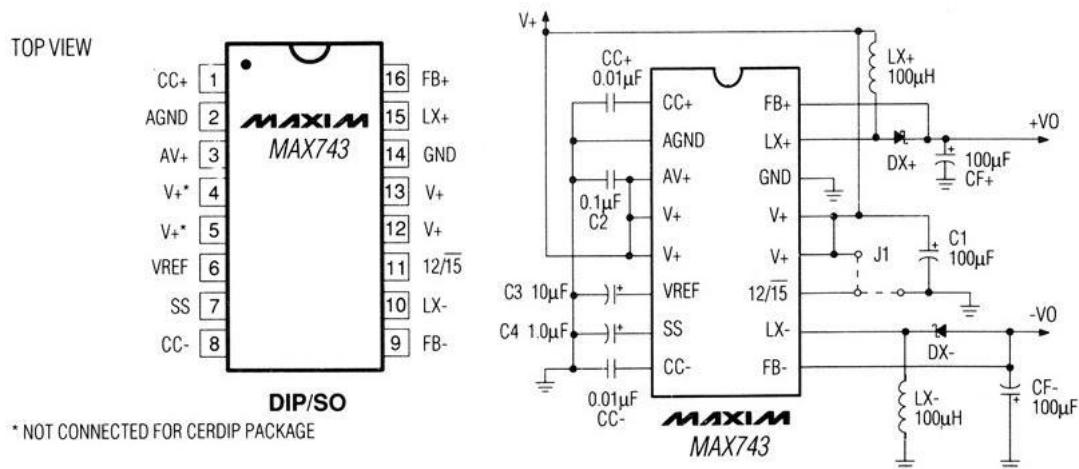


Ilustración 3.2: Patillaje y circuito del MAX743

Esta es la configuración que vamos a utilizar, conectando en V^+ los 5V del USB, tendremos los $\pm 15V$ en $+VO$ y en $-VO$, respectivamente.

3.1.2 REGULADOR DE TENSIÓN LM79L12

El anterior dispositivo integrado nos proporciona una tensión muy aproximada de $\pm 15V$, pero para la alimentar al resto de la tarjeta no nos podemos permitir tener derivas en la alimentación de todo el sistema. Por ello se renunciará a un poco de tensión a cambio de darle más estabilidad al sistema.

Para ello se utiliza el LM79L12, es un regulador de tensión, en este caso, pasaremos la tensión que salía del MAX743 de -15V a -12V. A parte de proporcionar toda la corriente que necesitamos, nos aporta: cortocircuito interno al salirnos de la zona de operación segura, también al llegar a temperaturas peligrosas, etc. [8]

A continuación, mostramos el esquema más común de este integrado y su patillaje:

3. Descripción del sistema

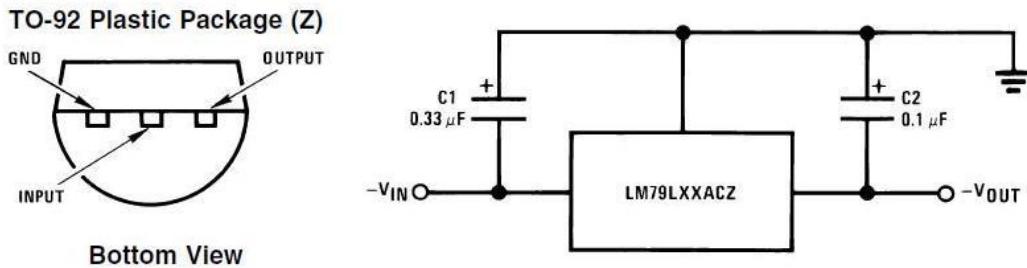


Ilustración 3.3: Patillaje y circuito del LM79L12

3.1.3 REGULADOR DE TENSIÓN LM78L12

En este apartado llevaremos exactamente la misma operación, pero en vez de pasar de una tensión de -15V a -12V, pasaremos de una tensión de +15V a una de +12V. Tiene las mismas características que en el dispositivo integrado anterior, ya que provienen de la misma casa, *National Semiconductor*. [9]

A continuación, mostramos el esquema más común de este integrado y su patillaje:

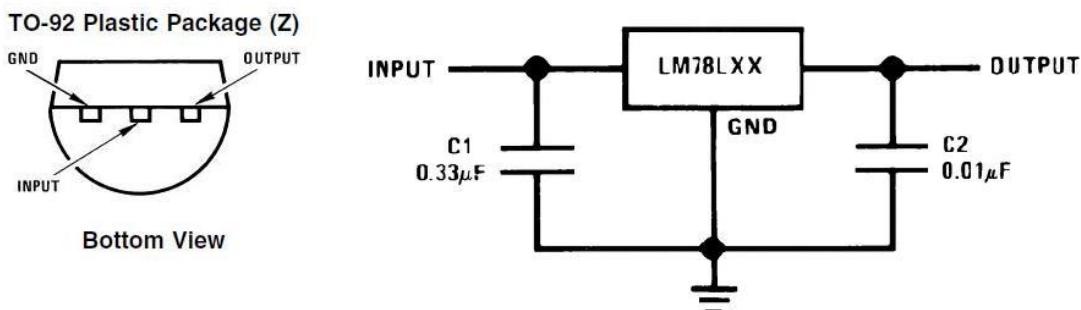


Ilustración 3.4: Patillaje y circuito del LM78L12

3.14 REFERENCIA DE TENSIÓN LM4040

Una referencia de tensión, nos garantiza una tensión en continua muy estable tanto a corto como a largo plazo. El dispositivo integrado LM4040 nos proporcionará esto: además de tener la ventaja de no tener que añadir ninguna capacidad exterior para que funcione correctamente, no influye en su estabilidad, por lo que nos ahorra introducir nuevos elementos al sistema. Otra de sus cualidades es su encapsulado, ya que en este caso usaremos el encapsulado TO-92, muy práctico en situaciones donde el espacio es vital.

Optimización de una tarjeta de adquisición de datos

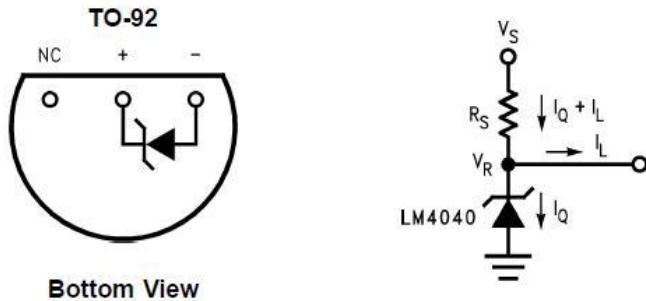


Ilustración 3.5: Patillaje y circuito del LM4040

Para saber que R_S tenemos que poner, usaremos la siguiente ecuación:

$$R_S = \frac{V_S - V_R}{I_L + I_Q}$$

Donde V_S es la tensión que aplicamos, V_R es la caída de tensión del diodo Zener, y $I_L + I_Q$ la suma de corrientes de la carga y del LM4040. En nuestro caso, R_S será una resistencia de $10\text{k}\Omega$. [10]

3.2 ACONDICIONAMIENTO DE LA SEÑAL

Algunas señales que pretendemos medir con el microcontrolador, van a tener valores más elevados que los que el microcontrolador puede manejar. Concretamente, vamos a medir señales que oscilen entre los $\pm 12\text{V}$ y el microcontrolador solo está capacitado para manejar señales de cero a $+5\text{V}$. Por este motivo vamos a tener que adaptar, que acondicionar las señales que se estén fuera del rango del microcontrolador, este es el objetivo de esta etapa.

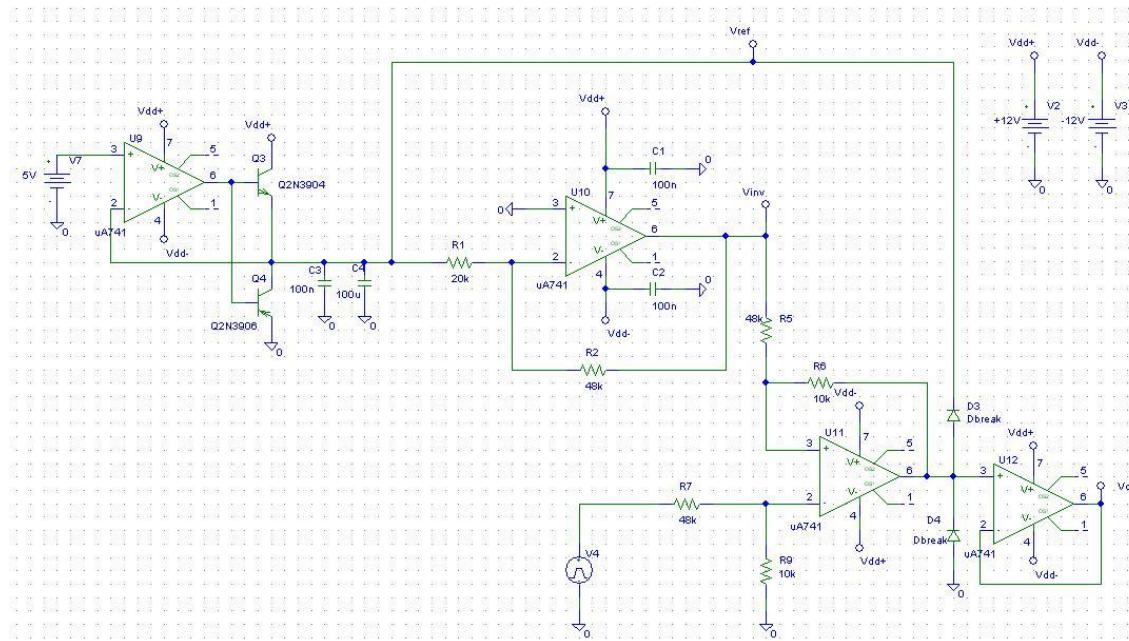


Ilustración 3.6: Etapa de acondicionamiento para señales más grandes de cero a $+5\text{V}$

3. Descripción del sistema

Partimos del integrado LM4040, con sus +5V que trasformaremos en -12V, una vez tengamos esto, pasaremos a amplificarlo y a invertirlo, teniendo un rango de cero a 24V. Con esto, lo que nos queda es reducirlo mediante resistencias para tener el rango que gestiona el microcontrolador, o sea de 0 a +5V.

El operador operacional que vamos a usar es el integrado LM358, un dispositivo integrado que cumple los requisitos eléctricos que necesitamos, tales como ancho de banda, ganancia necesaria, baja corriente de polarización, etc.

A la hora de diseñar circuitos electrónicos, es de gran utilidad contar con herramientas de simulación para saber de antemano cómo va a responder el circuito sin tener que montarlo o fabricarlo. En este caso, para realizar el diseño y la posterior simulación, se ha elegido el software PSpice 9.2, que pertenece a la compañía *Electronic Design Automation* (EDA). [11] Usaremos este programa, ya que llevamos usándolo todos los años de carrera y hemos adquirido destreza en su manejo.

Así que con la ayuda del software PSpice, tendremos la posibilidad de simular este circuito, y podemos ver su comportamiento que tiene su salida según la entrada que decidimos generar. Esto, se muestra a continuación:



Ilustración 3.7: Simulación de la etapa de acondicionamiento

Como se puede observar, el eje x representa el valor de la tensión de la entrada, es decir, de las señales que deseamos acondicionar para su posterior procesamiento en el microcontrolador, y en el eje y vemos la señal que tenemos a la salida de toda esta etapa de acondicionamiento. Podemos ver

Optimización de una tarjeta de adquisición de datos

que cuando la señal de entrada vale -12V, a la salida tenemos 0V; sin embargo, cuando tenemos una señal de +12V, tendremos la máxima tensión permitida en el microcontrolador, +5V.

3.3 MULTIPLEXOR ADG408

El integrado ADG408 basado en transistores CMOS, nos permite multiplexar 8 entradas analógicas a 1 salida analógica. Para elegir qué entrada vamos a querer en cada instante, el integrado consta de 3 patillas para decidir qué entrada comutar en cada momento, mediante líneas de dirección binarios de 3 bits.

Diseñado para proporcionar una baja disipación en potencia a una alta velocidad de conmutación, el integrado será ideal para aumentar el número de canales que nuestro microcontrolador puede llegar a medir. Como se puede deducir, el encargado de indicar a este multiplexor el canal que se desea recibir será el microcontrolador.

Además de tener un gran rango de tensión nominal, de unos 44V, y bajas resistencias, el propio fabricante nos indica que una de sus funciones para las que ha sido diseñado es para los sistemas de adquisición de datos. A continuación mostramos su patillaje y su esquema interno: [12]

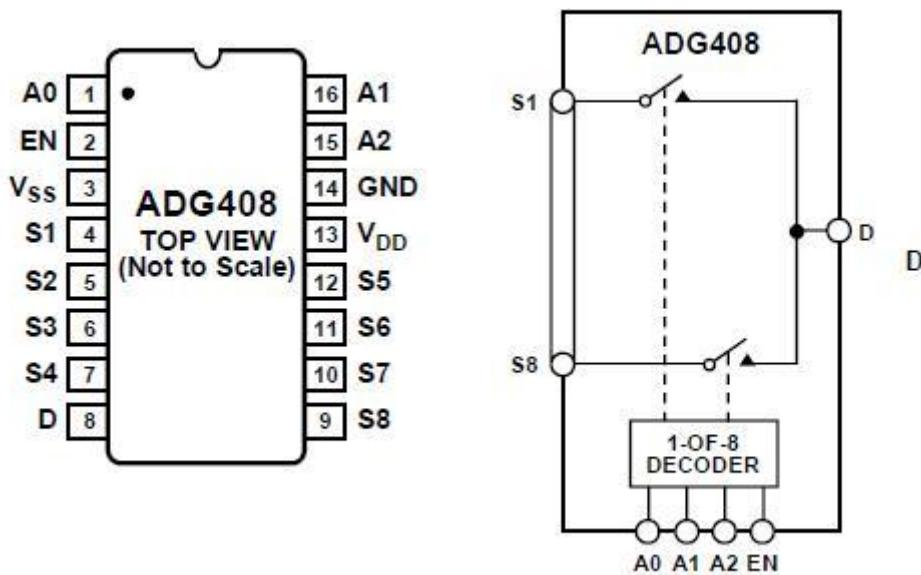


Ilustración 3.8: Patillaje y esquema interno del integrado ADG408

3.4. MICROCONTROLADOR PIC18F4553

Hemos mencionado en numerosas ocasiones la palabra microcontrolador, y no es para menos, ya que va a ser el núcleo de nuestro proyecto. Será el encargado de recibir todos los datos de tensiones e intensidades de corrientes, pasarlas del dominio analógico al dominio digital, para posteriormente ser capaces de enviarlas mediante la conexión USB al PC, para que éste se encargue de su tratamiento y almacenamiento. Además usaremos el microcontrolador para generar una señal PWM, pero esta parte se detallará más incisivamente en el siguiente apartado.

Por lo tanto, hemos buscado un microcontrolador capaz de realizar todas las tareas que nuestro proyecto exige, no nos vale cualquiera. Actualmente, en el mercado se puede encontrar una amplia variedad de microcontroladores, dependiendo del cometido que se desea llevar a cabo. Es una realidad que una empresa líder en este sector es Microchip, y justamente el microcontrolador elegido para este dispositivo, es el PIC18F4553 de Microchip.

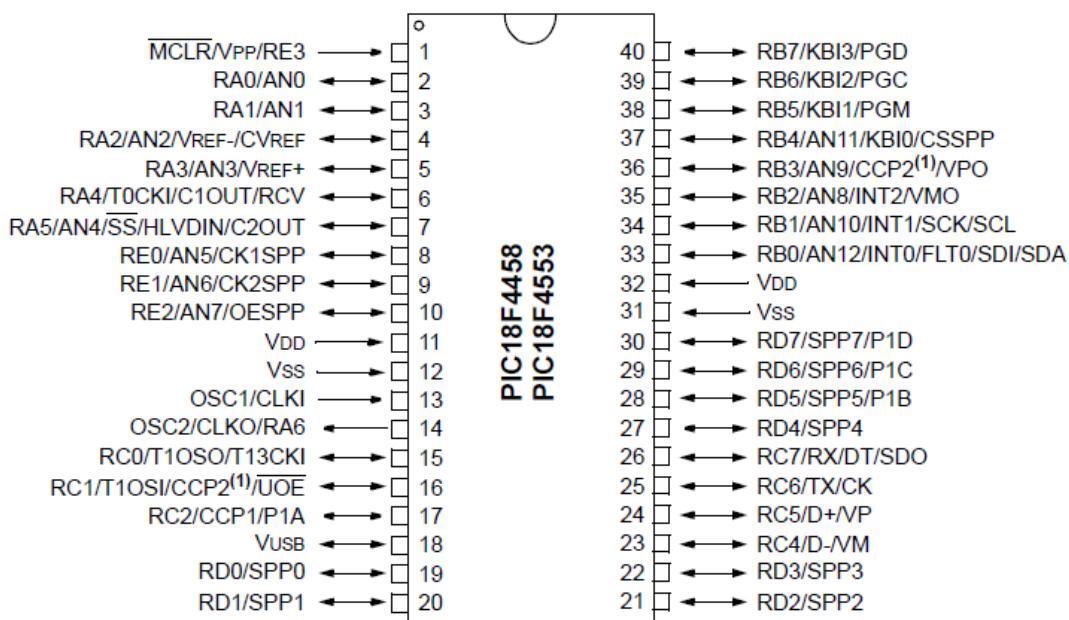


Ilustración 3.9: Patillaje del PIC18F4553

3.4.1 CONVERTIDOR ANALÓGICO-DIGITAL

El principal elemento que hace que hayamos elegido este microcontrolador es su convertidor analógico-digital, ADC (de sus siglas en inglés *analog digital converter*), de 12 bits de resolución. Para este modelo, el ADC tiene 13 entradas, que sumado a las 8 entradas del multiplexor ADG408, tenemos un número adecuado de entradas para nuestra tarjeta de adquisición. Por lo que

Optimización de una tarjeta de adquisición de datos

podemos deducir que nos será imposible tomar medidas de varias entradas simultáneamente, ya que solo disponemos de un solo convertidor analógico-digital en el microcontrolador. Esto hubiera sido una propiedad muy atractiva de la tarjeta de adquisición de datos, pero tenemos que sacrificar esta funcionalidad.

Básicamente lo que hace el ADC es convertir la variable analógica de tensión en una serie de datos digitales. Para ello, asigna un valor digital a un cierto valor analógico. Para llegar a esta conversión, lo primero que hacemos es convertir la tensión que tenemos a un determinado rango de tensión, de la siguiente manera:

$$V_{nom} = \frac{V - V_{ref(-)}}{V_{ref(+)} - V_{ref(-)}}$$

De esta modo, cuando la tensión V llegue al máximo valor, a $V_{ref(+)}$ tendremos el valor 1; en el caso contrario, obtendríamos el valor 0.

Ahora, el siguiente paso es ver en cuántos niveles de enteros podemos expresarlo. Para ello, nos fijaremos en el número de bits del convertidor, en nuestro caso 12 bits. Si queremos saber el valor exacto de estos niveles no tenemos que hacer más que $2^{12}=4096$, por lo tanto tendremos 4096 valores posibles para asignar a la señal analógica. Como sabemos de antemano que las señal de tensión analógica va a oscilar entre cero y +5V, sabemos que tenemos una resolución de $5V/4096=1.22mV$. A este valor se le suele dominar LSB, de sus siglas en inglés *least significant bit*.

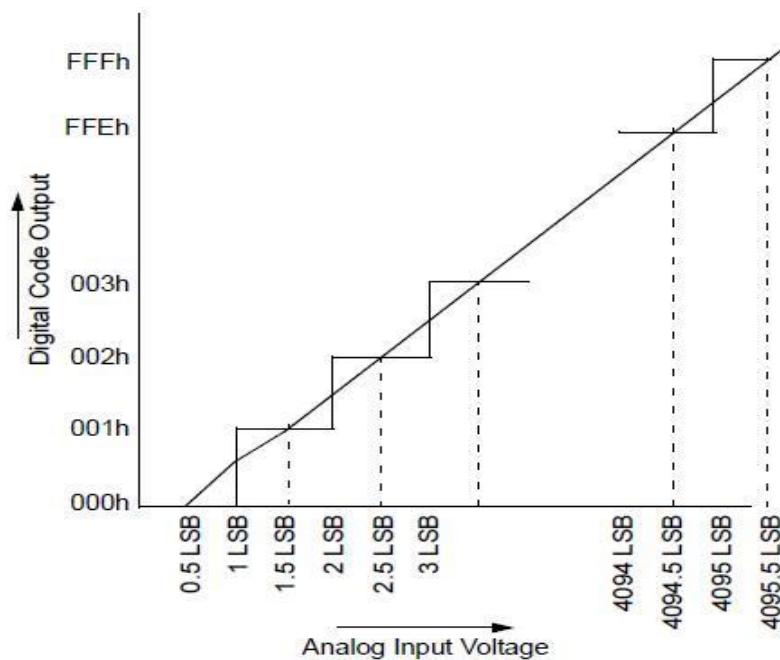


Ilustración 3.10: Función de transferencia analógico-digital

3. Descripción del sistema

El ADC nos da la libertad de que usemos las tensiones de referencia que queramos, dependiendo de nuestras necesidades, en nuestro caso usaremos una tensión de referencia positiva de +5V y una tensión de referencia negativa de 0V. La siguiente imagen muestra cómo podemos modificar la tensión, tal y como lo comentamos anteriormente:

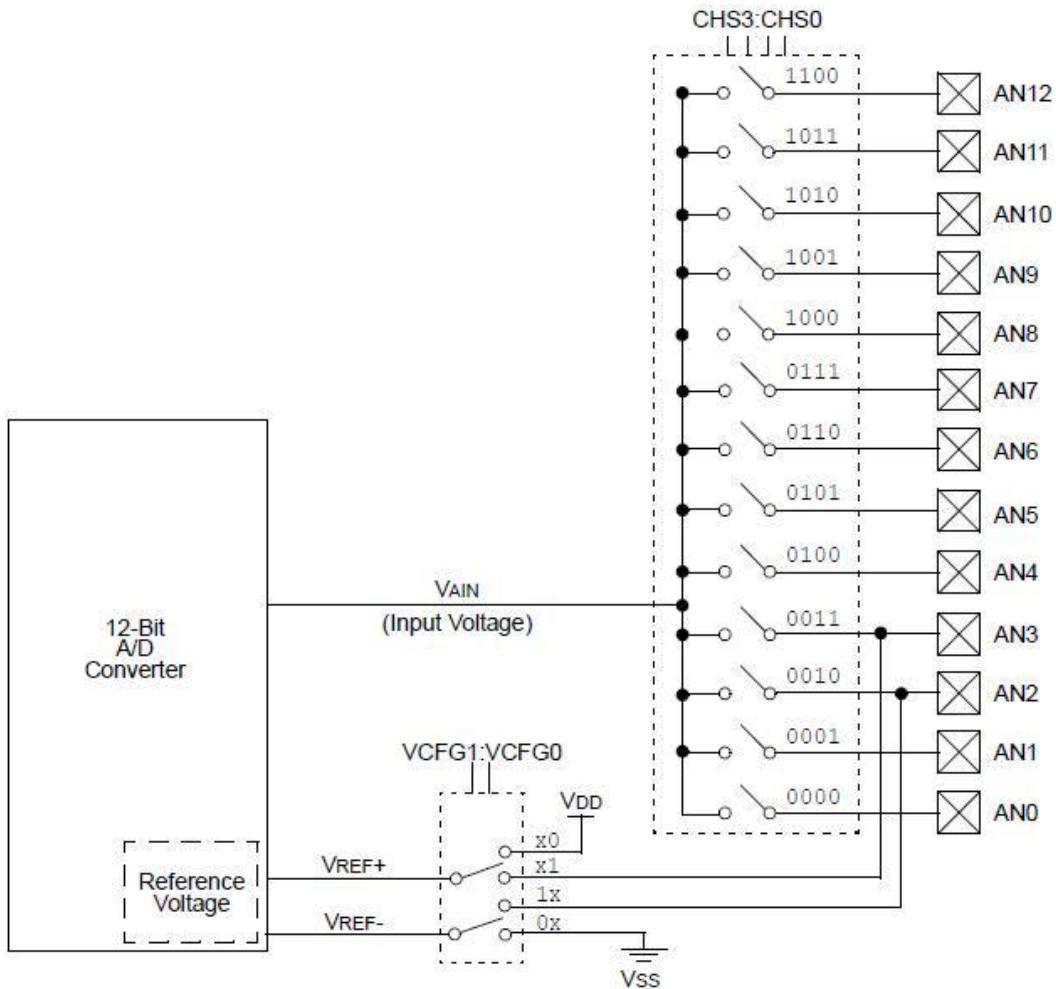


Ilustración 3.11: Esquema de convertidor analógico-digital

Por último, vamos a hablar del tiempo que necesita nuestro ADC para convertir una variable analógica en un valor digital, ya que buscamos exprimir la máxima velocidad para poder aumentar las muestras que tomamos por segundo.

En primer lugar tenemos el tiempo de adquisición, que es el tiempo que tarda el convertidor en fijar la tensión que vamos a medir. Este tiempo depende de muchos factores: de la capacidad que usemos para fijar el valor, la temperatura ambiente, etc. En general, nuestro tiempo de adquisición va a ser de $2,96\mu s$.

Optimización de una tarjeta de adquisición de datos

En segundo lugar tenemos el tiempo de conversión, que básicamente depende del número de bits que posee el convertidor. En nuestro caso este tiempo lo tomaremos como 0,8 μ s.

3.4.2 OSCILADOR

Todo microprocesador o microcontrolador requiere de un circuito que le indique a qué velocidad debe trabajar. Este circuito es conocido como un oscilador de frecuencia. Existen microcontroladores que tienen su oscilador internamente y no requieren de pequeños circuitos electrónicos externos, como en nuestro caso. Pero para este dispositivo vamos a usar un cristal de cuarzo de 4MHz.

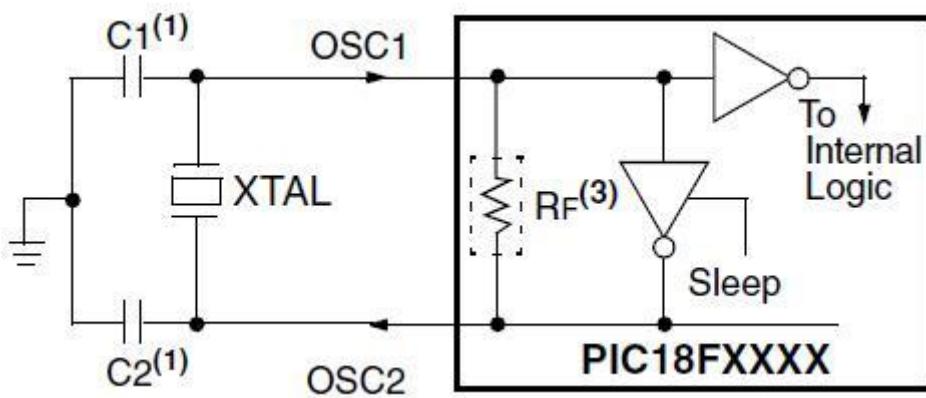


Ilustración 3.12: Esquema del circuito del oscilador

Como viene indicado por el propio fabricante, para el funcionamiento óptimo de esta parte del circuito los condensadores que añadamos tienen que tener la misma capacidad. El valor de éstos es de 33pF.

3.4.3 CONEXIÓN USB

Este microcontrolador, y en general toda su familia PIC18F455X, tiene disponible una interfaz que permite conectar el microcontrolador con cualquier host USB, lo que le da un aspecto muy interesante para la conexión con el PC, para el envío y recepción de datos.

Una curiosidad sobre esta parte, es que todo funciona con 3,3V por eso se puede ver el regulador de tensión. Pero nosotros, como hemos insistido en anteriores ocasiones, necesitaremos los +5V que nos proporciona la conexión USB para poder alimentar el resto de la tarjeta de adquisición de datos.

3. Descripción del sistema

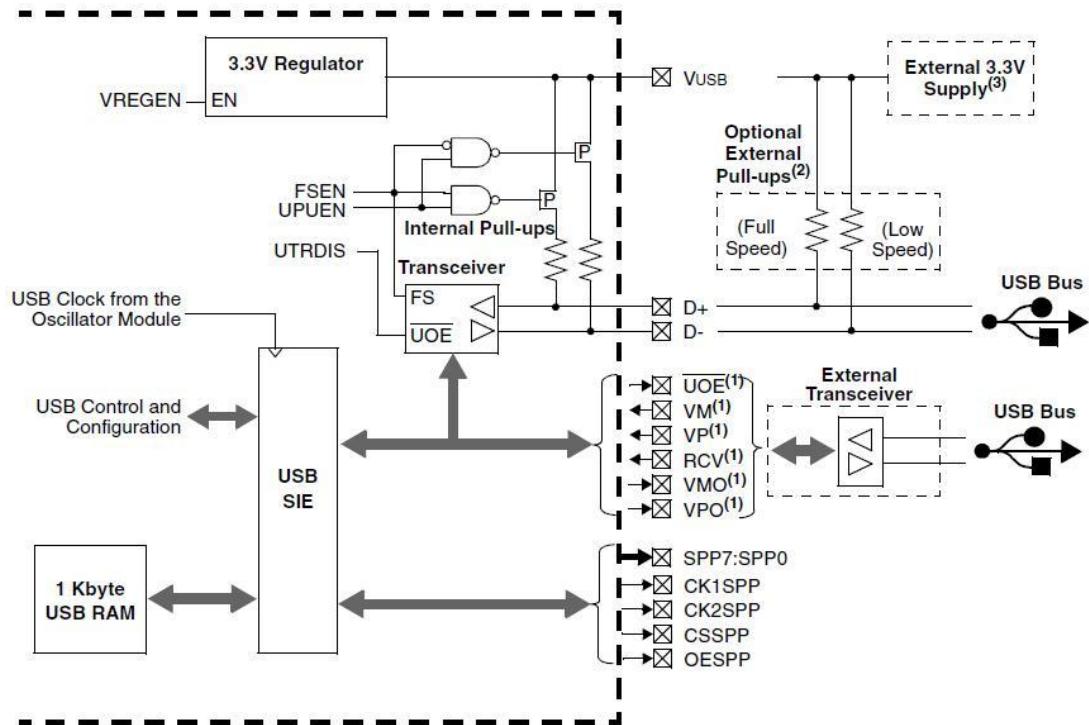


Ilustración 3.13: Esquema de la conexión USB

3.5 SMU

En este apartado lo que vamos a describir es la parte que se encarga de medir la intensidad de corriente eléctrica, mediante el uso de un SMU, Source Monitor Unit, con el cual mediremos tensiones y obtendremos corrientes. Detallaremos todas sus partes, como se muestra a continuación.

3.5.1 GENERACIÓN DE LA SEÑAL PWM

El objetivo es usar un DAC (*digital to analogue converter*), ya que queremos que el microcontrolador sea quien controle la tensión que aplicaremos en una de las partes del circuito SMU.

Para llevar a cabo esta tarea, vamos a generar una señal PWM (*Pulse Width Modulation*), en esta modulación la información viaja en la anchura del pulso, lo que es decir, en su ciclo de trabajo. A continuación mostramos un ejemplo visual, para facilitar su comprensión:

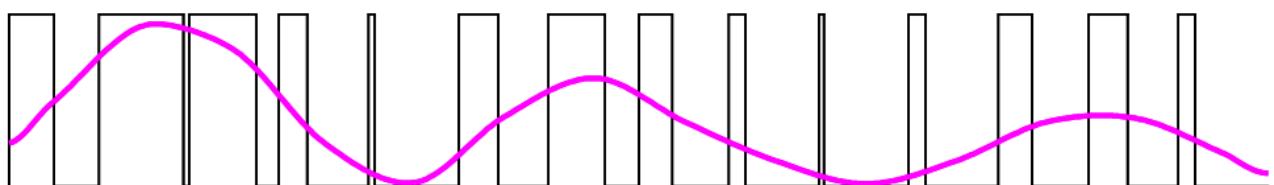


Ilustración 3.14: Señal analógica en rosa, señal resultante con la codificación PWM

Optimización de una tarjeta de adquisición de datos

Podemos apreciar que cuando el pulso es más ancho, la señal toma un valor más elevado; Por el contrario, cuando el pulso es muy estrecho, la señal tendrá los valores más bajos posibles.

Para poder generar una señal eléctrica de estas determinadas características, vamos a utilizar el módulo CCP de nuestro controlador, el PIC18F4553.

Los módulos CCP (*Capture*, *Compare*, *PWM*) de los PIC realizan por hardware las siguientes acciones:

- **Captura de impulsos:** el módulo CCP registra el instante en el que se produce un evento (un flanco o un número de flancos de pulsos en su entrada).
- **Comparación:** compara constantemente valores de dos registros. Uno de estos valores es siempre el Timer1. Este circuito también permite activar interrupciones externas.
- **PWM (Pulse Width Modulation): generación de señal modulada en anchura de impulsos.**

Como se puede deducir fácilmente, a nosotros lo que nos interesa es la generación de señales PWM, descritas anteriormente.

3.5.2 CONFIGURACIÓN DEL MÓDULO CCP PARA GENERAR SEÑALES PWM

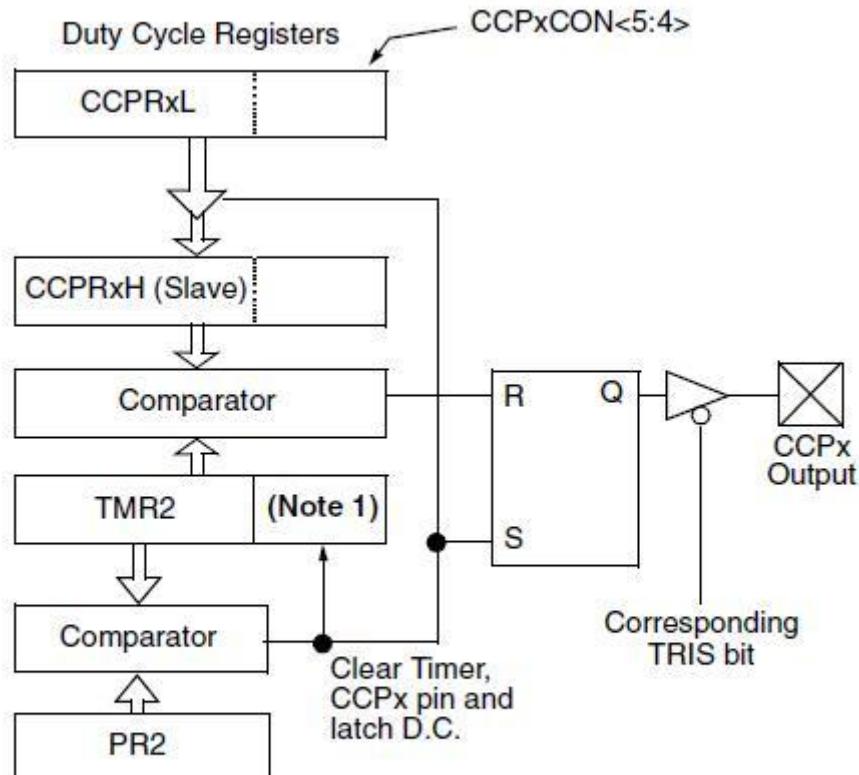
Si queremos utilizar el módulo CCP para generar señales PWM, lo primero que tenemos que hacer es configurarlo. Para esta tarea, existen unos determinados bits para indicar qué modo queremos usar, pues también existe la posibilidad de utilizar alguno de los otros existentes.

Una vez que tenemos configurado el modo PWM, lo que tenemos que calcular son los parámetros básicos que queremos para la señal PWM deseada. Para ello, en el Data Sheet del microcontrolador PIC18F4553, se nos indican unas pautas a seguir para facilitarnos este paso, como se muestra a continuación:

$$PWM\ Period = PR2 + 1 * 4 * Tosc * TMR2\ Prescale\ Value$$

A partir de esta fórmula, podemos calcular el periodo de la señal que vamos a generar, y del periodo, su frecuencia. Así que, si usamos un TMR2=1, un PR2=255 y con una frecuencia de oscilación de 4,8MHz, obtendremos una señal PWM de 4,68 KHz.

3. Descripción del sistema



Note 1: The 8-bit TMR2 value is concatenated with the 2-bit internal Q clock, or 2 bits of the prescaler, to create the 10-bit time base.

Ilustración 3.15: Esquema interno del módulo CCP del PIC18F4553

En nuestro caso, vamos a generar una señal PWM de 50kHz, para facilitarnos el filtrado, que viene en la siguiente etapa. Para tener una señal con esta determinada frecuencia, vamos a usar un TMR2=1, un PR2=23 y con una frecuencia de oscilación de 4,8MHz, con lo que tenemos una frecuencia de 50kHz.

El microcontrolador también gestionará el tamaño del ciclo de trabajo, o lo que es lo mismo, la anchura del pulso. Tal vez no parezca relevante, pero como explicaremos a continuación, la señal PWM será filtrada para obtener una señal en continua, y el valor de esta señal dependerá de la anchura del pulso. En este caso, al igual que en el ejemplo que se explicó al comienzo del apartado, a mayor anchura de pulso, mayor valor tendrá la señal en continua.

Optimización de una tarjeta de adquisición de datos

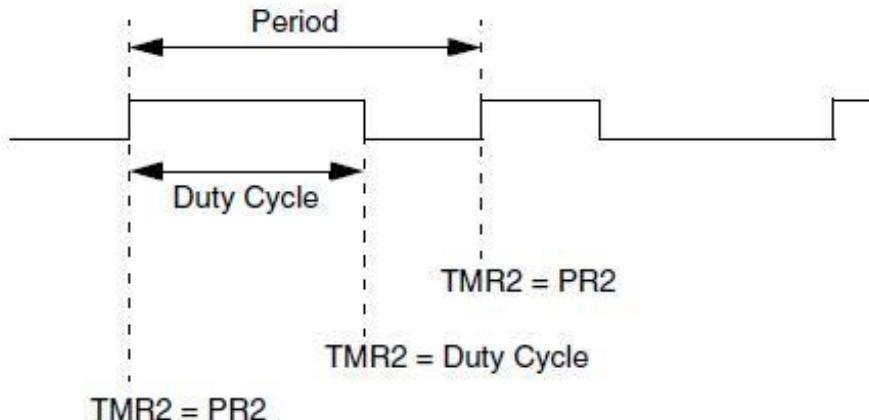


Ilustración 3.16: Período y ciclo de trabajo de una señal PWM

3.5.3 FILTRADO DE LA SEÑAL

Proseguiremos con el filtrado de la señal PWM, y para ello utilizaremos un filtro paso bajo, en este caso un filtro de Sallen Key que es un filtro activo y eficaz para su baja complejidad. Este filtro tan solo se compone de 2 resistencias, 2 condensadores y un amplificador, se podría obtener filtros de mayor orden, repitiendo su esquema en cascada, pero en nuestro caso no nos hará falta. A parte de tener una configuración muy básica, la característica por la que hemos elegido este filtro se ha basado en que no invierte la señal, característica que nos viene muy bien, ya que no lo necesitamos. Y por otro lado, la ganancia del filtro es la unidad, aspecto que no es relevante, ya que la ganancia que necesitaremos la introduciremos en las etapas posteriores.

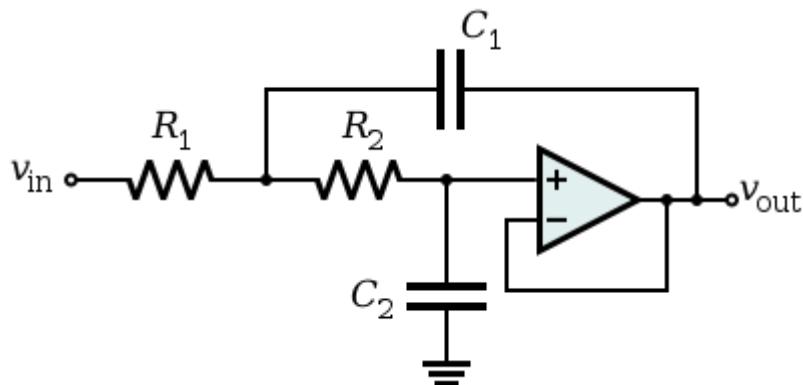


Ilustración 3.17: Esquema de un filtro paso bajo con la configuración de Sallen Key

La frecuencia de corte y el factor de calidad de este filtro son las siguientes:

$$F_c = \frac{1}{2\pi R_1 R_2 C_1 C_2}$$

3. Descripción del sistema

$$Q = \frac{2\pi}{C_2(R_1 + R_2)} \frac{R_1 R_2 C_1 C_2}{}$$

Para este filtro, hemos utilizado unas resistencias de $100\text{K}\Omega$, y para los condensadores 1 y 2, 100nF y 200nF respectivamente. Con estos valores pretendemos obtener una frecuencia de corte de $11,25$ Hz, lo cual nos es suficiente para el cometido que necesitamos que cumpla.

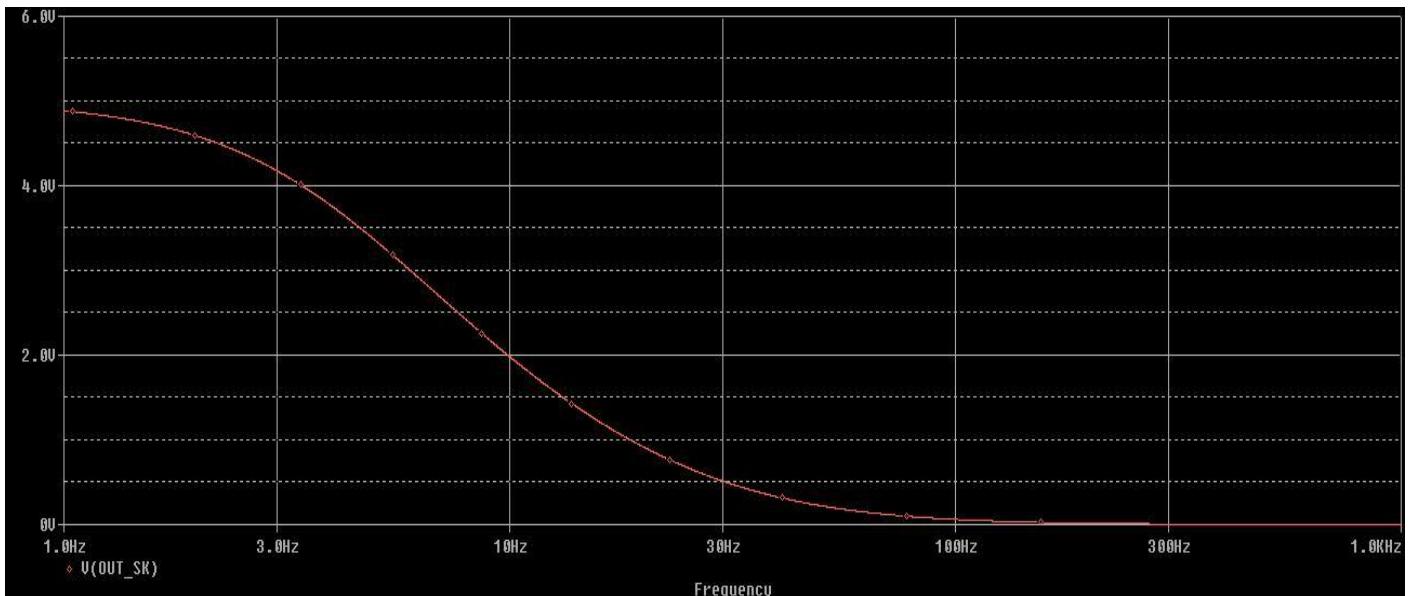


Ilustración 3.18: Respuesta en frecuencia de nuestro filtro Sallen Key

Con este filtro, lo que vamos a obtener a la salida es una ganancia, que dependiendo de la anchura del pulso que le llegue y del tamaño de su ciclo de trabajo, vamos a tener un valor en continuo u otro. La fórmula que gobierna este proceso es:

$$\text{Valor de la señal en continua} = V_{cc} * D$$

Donde V_{cc} es el valor máximo que alcanza la señal PWM, el valor del estado en alto, en nuestro caso es de 5V ; y D es el ciclo de trabajo que tiene la señal PWM, e irá de 0 a 1. El valor de la señal va a oscilar entre 0 y $+5\text{V}$.

Como se aprecia, es una relación bastante sencilla, y la cual nos proporciona una señal continua de una señal digital. Ese era nuestro objetivo, poder gestionar la tensión de una señal analógica mediante las señales que nos aporta el microcontrolador, que son de naturaleza digital.

3.5.4 ETAPA DE ACONDICIONAMIENTO SEÑAL PWM

Ahora que ya tenemos una tensión comprendida entre cero y $+5\text{V}$, el próximo objetivo es aumentar este rango, es decir vamos a pasar a tener una señal que vaya desde los -12V a los $+12\text{V}$ de una señal que solo iba de cero a $+5\text{V}$. Ese es el objetivo de la etapa DC Shift, que se muestra aquí:

Optimización de una tarjeta de adquisición de datos

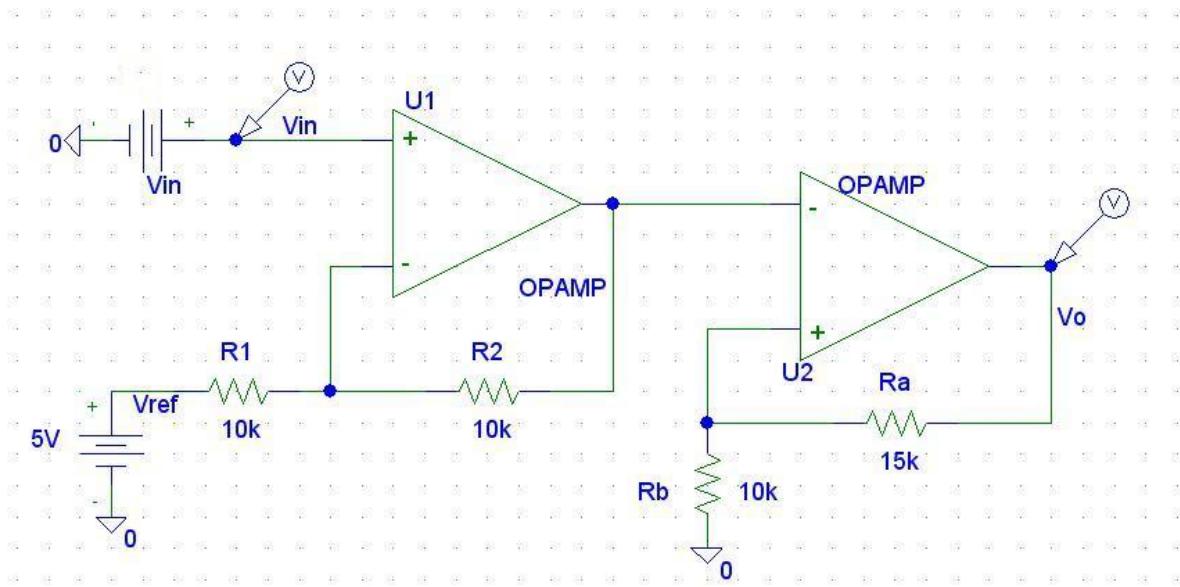


Ilustración 3.19: Etapa de acondicionamiento de la señal PWM

En el primer amplificador operacional, lo que conseguimos es desplazar la tensión de cero a +5V a de -5V a +5V. También le aplicamos una pequeña amplificación, pero no es lo suficiente. Este es el motivo de que apliquemos el segundo operador operacional, con la que alcanzaremos la ganancia necesaria.

Para tener la tensión que queremos, necesitamos 4,8 de ganancia. Este dato es el cálculo de dividir los 24 voltios que queremos tener de margen, por los 5 voltios que tenemos de margen al principio de esta etapa.

Con esta segunda etapa somos capaces de tener los deseados -12V a +12V. Podemos ver la simulación de lo anteriormente explicado justo debajo:

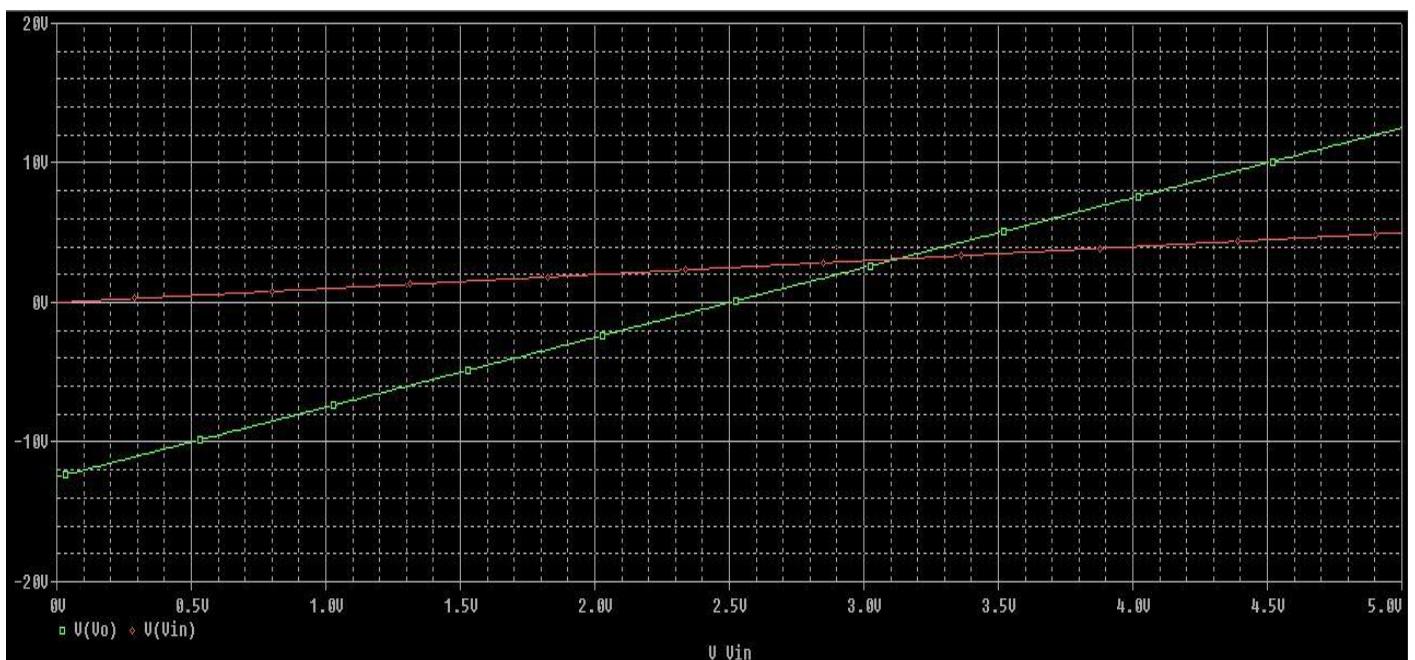


Ilustración 3.20: Simulación de la etapa de acondicionamiento de la señal PWM

3. Descripción del sistema

Como se aprecia en la imagen, la curva de color rojo representa la entrada al sistema. Esta entrada la hacemos variar entre 0V y +5V. Vemos que la otra curva de color verde, que se corresponde con la salida del sistema, cuando la entrada es de 0V la salida equivale a -12V. En el caso opuesto, cuando la entrada es de +5V, su salida adquiere el valor de +12V.

Una vez pasada esta etapa, ya tenemos una señal analógica y con un rango bastante amplio, de -12V a +12V, para poder utilizar en el SMU. Y lo mejor de todo esto es que vamos a poder controlar el valor de esta señal mediante el microcontrolador.

3.5.5 FUNCIONAMIENTO DE LA SMU

El SMU, llamado así por sus siglas en inglés Source Measure Unit, es el circuito que vamos a utilizar para poder medir la corriente. Para ello, tenemos que analizar el circuito:

$$V_o = V^- + IR_1 \rightarrow I = \frac{V_o - V^-}{R_1}$$

Sabiendo que la tensión que tenemos en la patilla no inversora, la que proviene de los apartados anteriores, es la señal PWM filtrada y acondicionada a $\pm 12V$ que tenemos en la patilla inversora.

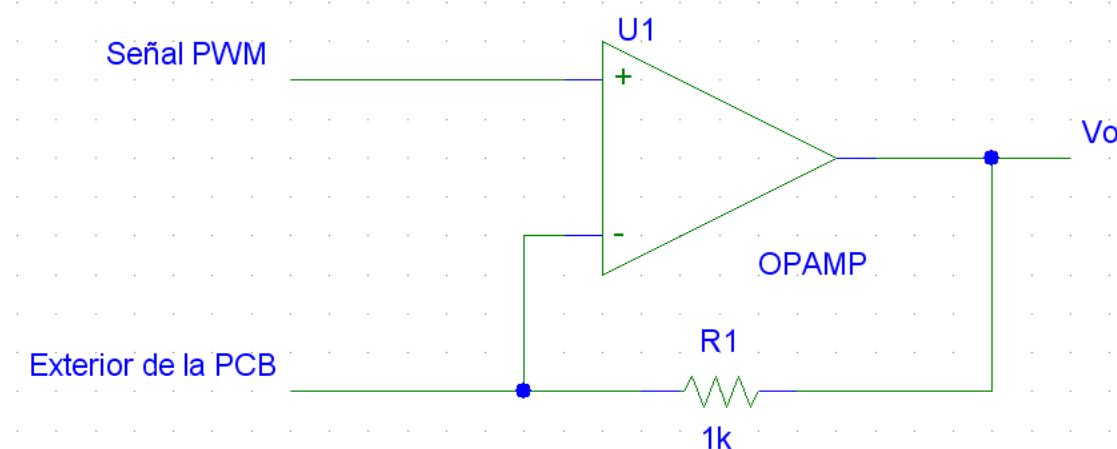


Ilustración 3.21: Esquema del circuito SMU

La salida del amplificador operacional se introduce en el multiplexor ADG408, para que podamos posteriormente acondicionarla para la entrada en el microcontrolador.

Una vez terminado de explicar todos los elementos que formaran parte de nuestro proyecto, pasaremos a la interconexión de todos ellos y a detallar como ha sido el proceso de fabricación de la PCB.

Optimización de una tarjeta de adquisición de datos

3. Descripción del sistema

Capítulo 4

Diseño de la placa de circuito impreso

4.1 Introducción

4.2 Software de diseño Altium

4.2.1 Esquemático de Altium

4.2.2 PCB de Altium

4.3 Fabricación de la PCB

4.3.1 Tipos de procesos de fabricación de una PCB

4.3.2 Proceso de fabricación de la PCB

4.3.3 Resultados de la PCB terminada

4.1 INTRODUCCIÓN

Una vez realizadas las simulaciones con PSpice de las partes anteriormente descritas, ya era posible pasar a la siguiente fase. El siguiente paso es la realización de un diseño para fabricar una PCB (de sus siglas en inglés *Printed Board Circuit*). Con esto lo que queremos obtener es un dispositivo más robusto que si, por ejemplo, utilizásemos una placa de baquelita.

Además la idea final es la de utilizar una caja metálica para poder guardar la PCB que fabricaremos, ya que nos proporcionará mayor protección para nuestro circuito, ya que es un dispositivo portable, además de un acabado mucho más elegante y funcional.

4.2 SOFTWARE DE DISEÑO ALTIUM

Altium Designer es un paquete de software de diseño electrónico de placa de circuito impreso, FPGA, software embebido, etc. Este programa, diseñado y comercializado por Altium Designer 14 (Australia), es el que se ha utilizado para realizar el diseño de la PCB de nuestro dispositivo. [14]



Ilustración 4.1: Software utilizado para el diseño de la PCB

En primer lugar, tenemos que crear un esquemático del dispositivo, ya que ésta será la base para luego proseguir con el diseño de la PCB. Este paso, aparte de ver el circuito de una manera más simplificada y esquematizada, por si tenemos que corregir algún error, nos proporcionará toda la información que vamos a necesitar cuando nos pongamos realmente a diseñar nuestra PCB, a colocar inteligentemente nuestros componentes, realizar pistas por una capa o por otra, etc.

Optimización de una tarjeta de adquisición de datos

4.2.1 ESQUEMÁTICO DE ALTIUM

Vamos a mostrar las distintas partes que tenemos en el esquemático de Altium, y la función que desempeña cada una:

- Circuito 1: Ilustra el circuito correspondiente a la referencia de tensión LM4040 que nos proporcionará un voltaje continuo de 5 voltios.
- Circuito 2: Etapa de potencia de la fuente de 5V que puede tanto proporcionar como absorber corriente.
- Circuito 3: Muestra el circuito que realiza la adaptación de niveles de las señales de $\pm 12V$. El primer amplificador nos proporciona una señal de $\pm 12V$ a partir de los 5V de la referencia de tensión. El segundo aplicador eleva la señal de entrada de $\pm 12V$ al rango positivo [0,24] V y mediante un divisor de tensión se obtiene el rango de entrada deseado de [0,5] V.
- Circuito 4: Circuito de protección contra sobretensiones.
- Circuito 5: Seguidor de tensión.

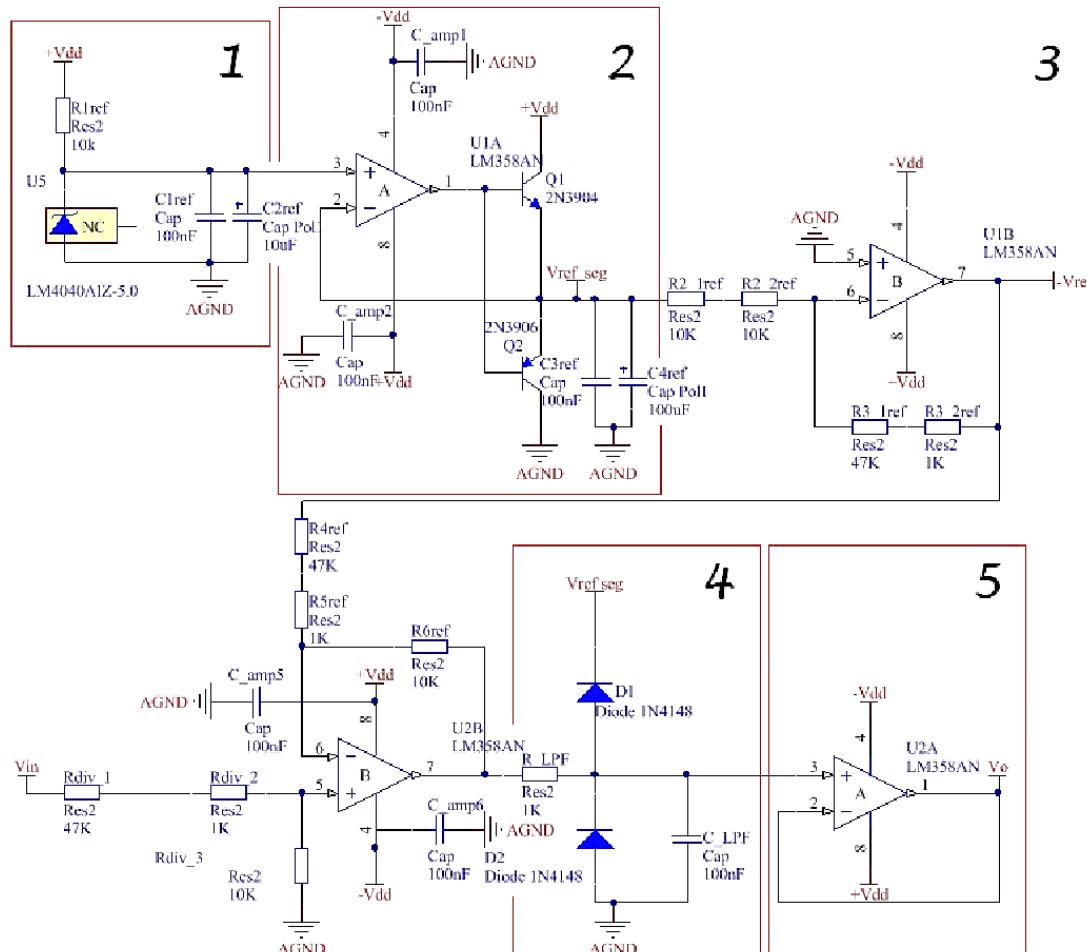


Ilustración 4.2: Etapa para el acondicionamiento de la señal

4. Diseño de la placa de circuito impreso

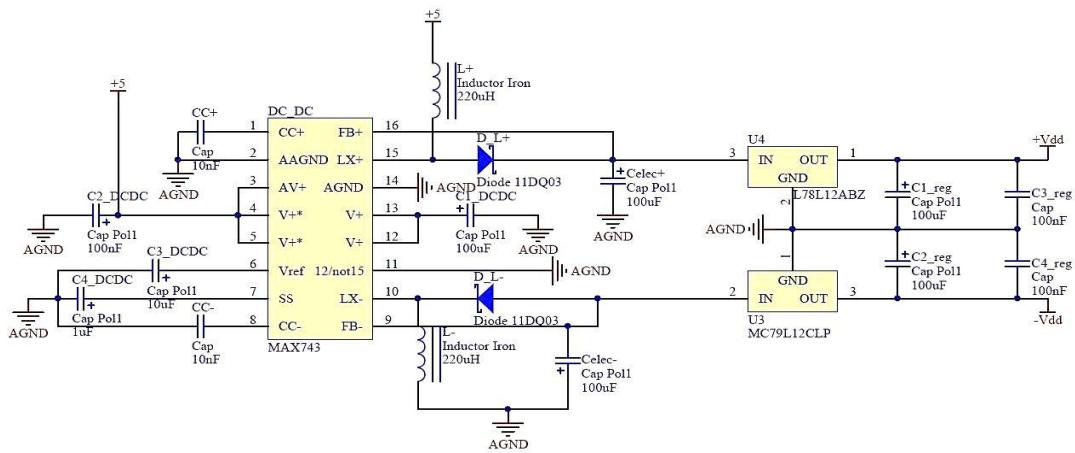


Ilustración 4.3: Etapa de potencia y regulación de la tensión

Ya se comentó en el apartado anterior la función que va a desempeñar este subcircuito. La de proporcionarnos unas tensiones de salida de $\pm 12V$ a partir de un voltaje de $+5V$.

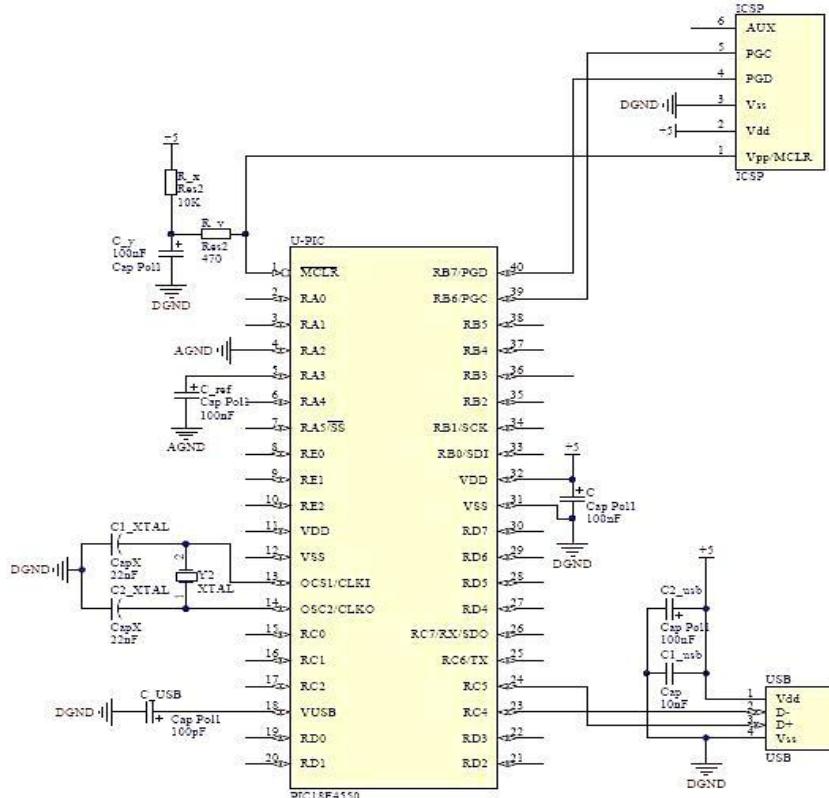


Ilustración 4.4: Esquema del microcontrolador y sus componentes básicos

En esta ilustración, podemos ver los elementos fundamentales para que el microcontrolador funcione sin ningún tipo de problema. Arriba a la derecha se aprecia el esquema necesario para que puesto ICSP funcione y seamos capaces de programar el microcontrolador. En la parte centrar a la izquierda, podemos apreciar el cristal de cuarzo con sus respectivos condensadores, para que podamos utilizar el microcontrolador a la frecuencia de trabajo deseada. Por último, en la esquina inferior derecha, encontramos la conexión del microcontrolador con el PC mediante un USB, este enlace necesita unos

Optimización de una tarjeta de adquisición de datos

condensadores, como se observa en la imagen, para su perfecto funcionamiento.

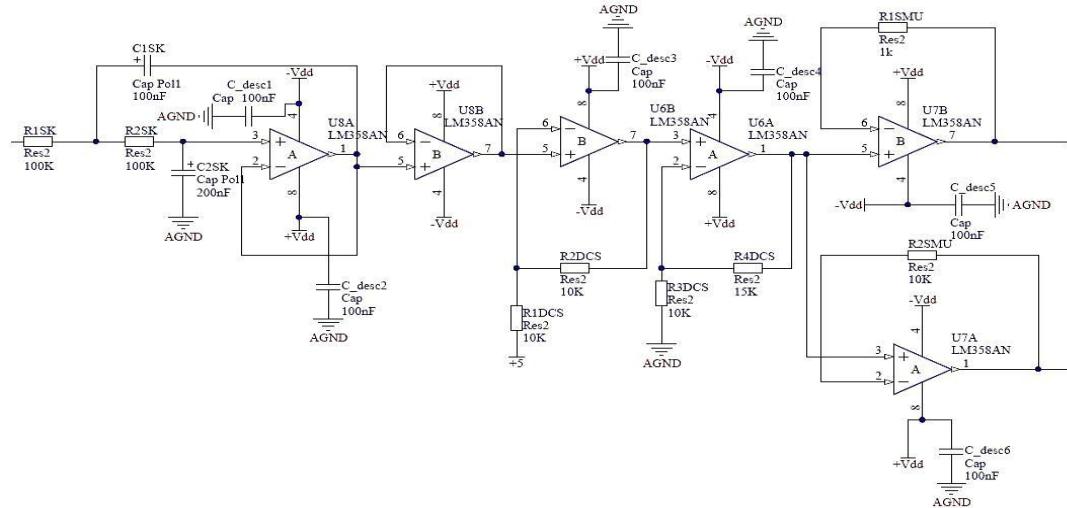


Ilustración 4.5: Etapa de los SMUs

Esta ilustración, muestra la parte referida a los SMU, los dos primeros amplificadores operacionales corresponde con el filtrado paso-baja mediante un filtro Sallen-Key. Los dos siguientes amplificadores operacionales, forman parte de la etapa de amplificación y desplazamiento de la tensión. Y por último, los amplificadores operacionales que podemos ver a la derecha de la imagen corresponden a los SMU, que se usarán para medición de la intensidad de corriente.

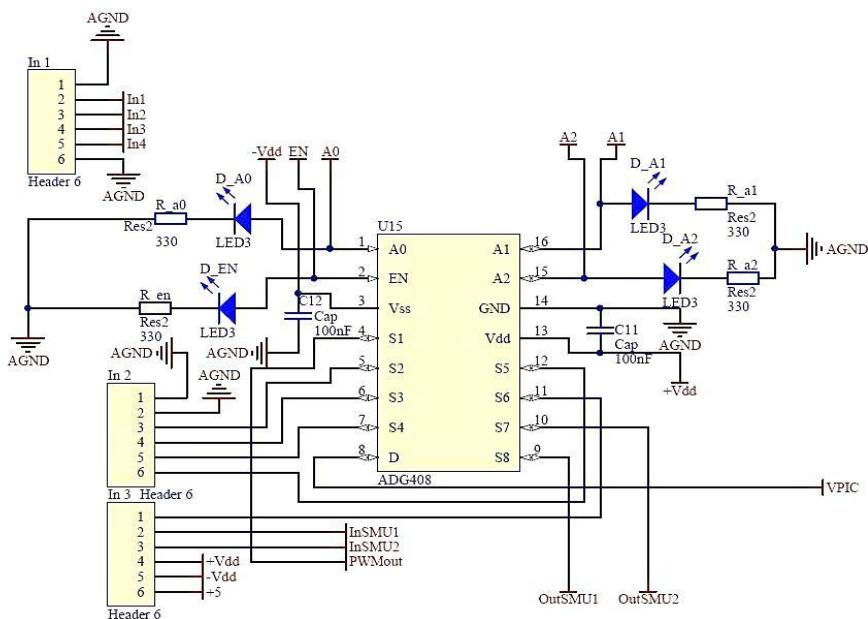


Ilustración 4.6: Conectores y multiplexor

En la imagen anterior, podemos ver un resumen de las conexiones que tendrá nuestro multiplexor y las conexiones que va a disponer nuestra tarjeta de adquisición de datos. Se observa que las señales que controlan el

4. Diseño de la placa de circuito impreso

multiplexor por parte del microcontrolador, disponen de un diodo y una resistencia para proteger al microcontrolador de posibles picos de tensiones o corrientes

Se puede ver una visión más general de la parte del esquemático del Altium en los apéndices del final.

4.2.2 PCB DE ALTIUM

En esta parte vamos a detallar y explicar la PCB, como se comentó en el apartado anterior, una vez realizado el esquemático, lo que nos queda es colocar los componentes e interconectarlos entre sí para tener el circuito que se ha planteado en la anterior parte. Puede parecer una tarea trivial, pero nada más lejos de la realidad. Aun siendo un trabajo motivador y atractivo, es una tarea complicada, que requiere de mucha concentración, y sobre todo de mucho tiempo, pues llegar a los resultados que aquí se muestran, han sido frutos de muchas horas. A continuación, se puede observar las 2 caras de la PCB:

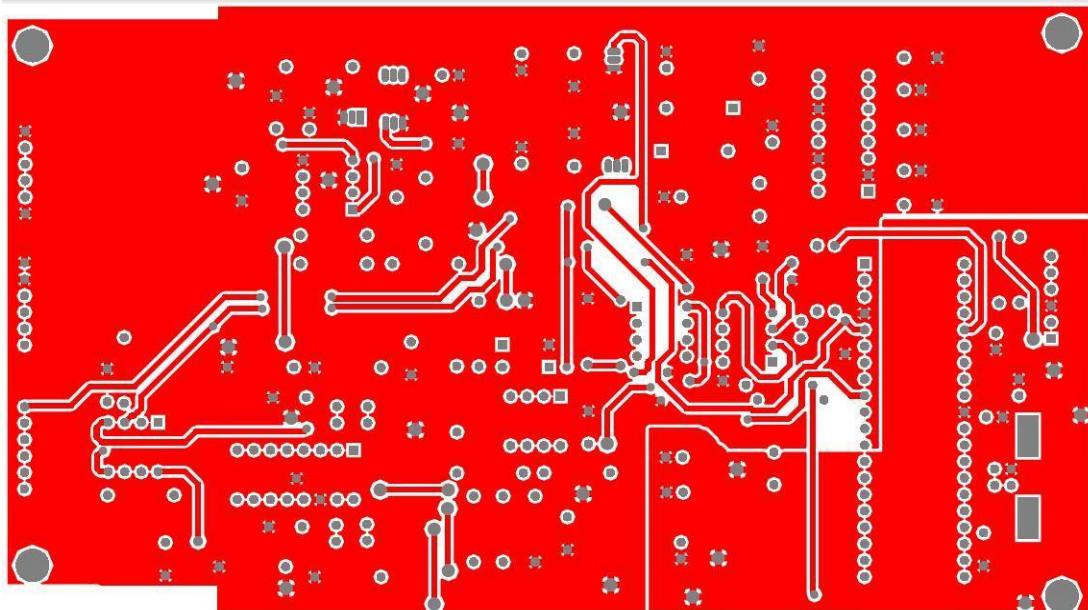


Ilustración 4.7: Esquema de la capa TOP

Optimización de una tarjeta de adquisición de datos

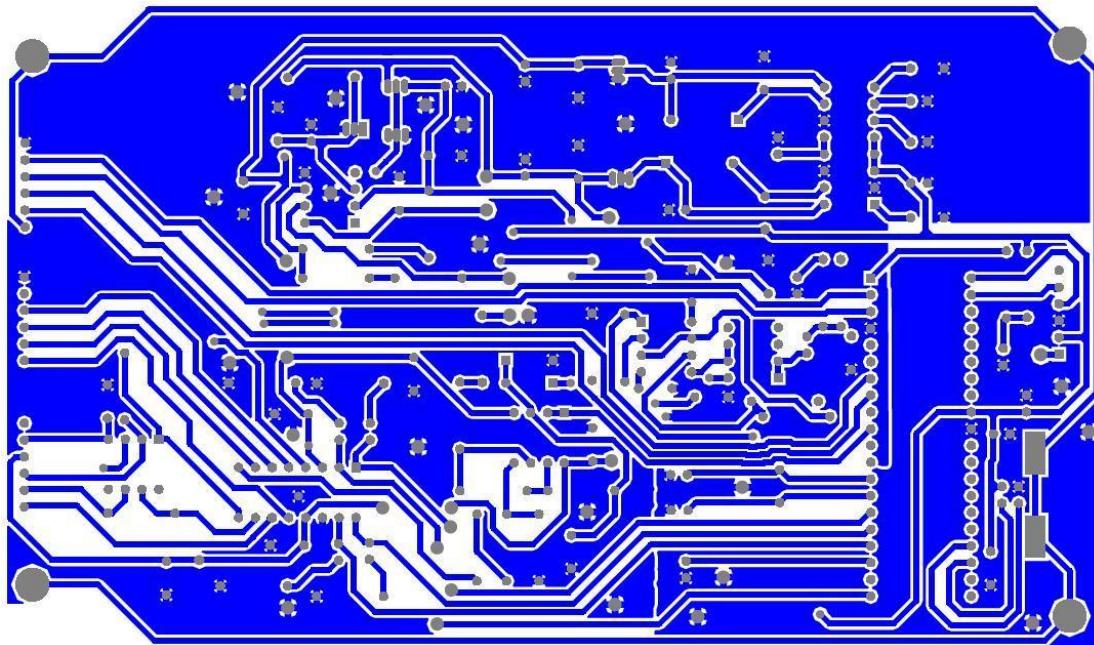


Ilustración 4.8: Esquema de la capa BOTTOM

Algunos comentarios y pautas sobre el planteamiento y el diseño de estas capas han sido:

- **Los componentes solo pueden estar en la TOP, nunca en la BOTTOM.**
- **Usamos componentes Through-Hole**, no se ha usado nada distinto.
- **Se ha intentado tirar todas las pistas posibles por la BOTTOM**, pero es casi imposible realizar todas las conexiones en ésta, por lo que se ha tenido que recurrir a **tirar pistas por la TOP puntualmente**, pero las menos posibles.
- El tamaño de la PCB viene dado por las dimensiones de la caja, como se comentó anteriormente, el objetivo es de que **la PCB repose en una caja metálica**.

El programa Altium también nos permite la posibilidad de crear un modelo 3D de toda la PCB, si se tienen los modelos 3D de los componentes, es muy fácil, y ayuda considerablemente a tener una idea más clara, sobretodo del aspecto físico de la PCB. Aquí podemos ver una imagen de lo que se estaba comentando:

4. Diseño de la placa de circuito impreso

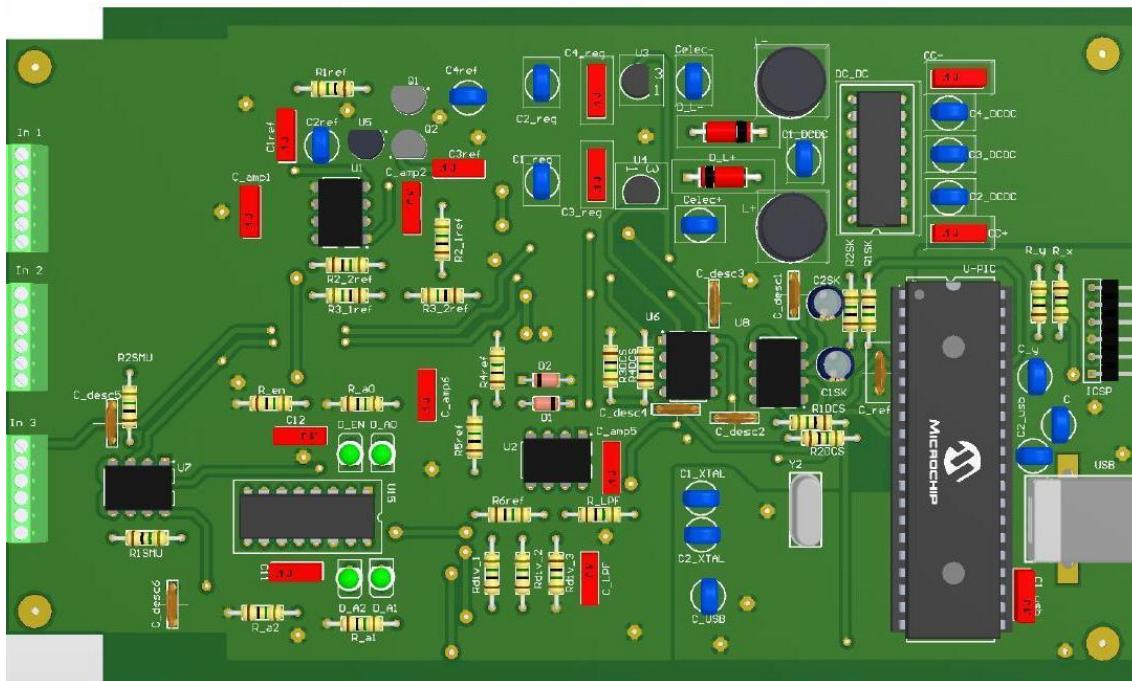


Ilustración 4.9: Imagen del modelo 3D de la PCB

4.3 FABRICACIÓN DE LA PCB

Para la fabricación de PCB, se utiliza una plancha de cobre por ambos lados. Ahora existen varios métodos para eliminar este cobre, de modo que podamos usar el cobre restante para interconectar nuestros componentes.

4.3.1 TIPOS DE PROCESOS DE FABRICACIÓN DE UNA PCB

El método químico, como su nombre indica, se usarán unos procesos químicos para “comer” el cobre no deseado, existen varias maneras de proteger las zonas de cobre donde no queremos que sea eliminado. Este no será nuestro proceso.

El proceso elegido se llevará a cabo mediante un taladro guiado por ordenador, el cual se encargara de quitar el cobre necesario para poder realizar las pistas que nosotros hemos diseñado previamente. Una ventaja de este último método respecto al químico es que usando el taladro la maquina nos hará automáticamente los agujeros en la PCB, cosa que no ocurre en el proceso químico, y tendríamos que usar un taladro manual.

Para llevar a cabo la fabricación de la PCB mediante el último método, es necesaria la creación de unos ficheros especiales, que podemos crear cómodamente en Altium. Estos ficheros son el fichero “Gerber” y el fichero “NC Drill”. El primero de éstos, contiene toda la información necesaria sobre las

Optimización de una tarjeta de adquisición de datos

pistas, las capas de la PCB, los planos de masas utilizados, etc. En cuanto al segundo, contiene toda la información sobre los agujeros que vamos a taladrar en nuestra PCB.

Con estos archivos, y utilizando otro software, somos capaces de crear todos los caminos y los distintos tipos de brocas que se van a necesitar para la fabricación de la placa. Una vez tenemos toda esta información y usando otro programa, que es realmente quien controla el taladro, nos disponemos a fabricar la placa

4.3.2 PROCESO DE FABRICACIÓN DE LA PCB

Teóricamente, si se ha realizado todo lo anterior, una vez que le damos la orden al taladro para que empiece, parece que ya solo queda esperar. Pero surgió un problema, y es que parece ser que no se ha llevado un correcto control sobre las vidas útiles de las brocas, y aunque calibrásemos las brocas antes de empezar todo el proceso, al no saber cuánta distancia ha recorrido cada broca, hubo problemas, ya que estaba realizando surcos demasiados irregulares que ponían en un serio compromiso la funcionalidad eléctrica de la tarjeta. Afortunadamente, estábamos pendientes de los surcos que estaban realizando el taladro y pudimos pararlo, y poder cambiar las brocas y volver a calibrarlas.

Una vez realizado este cambio, se pudo continuar fabricando la PCB sin ningún tipo de problemas. El tiempo de fabricación fue aproximadamente de unas 4 horas.

4.3.3 RESULTADO DE LA PCB TERMINADA

El resultado final, después de estar varias horas soldando y comprobando las soldaduras, para que no hubiera ningún cortocircuito y todo fuese igual que en los esquemáticos, es este:

4. Diseño de la placa de circuito impreso

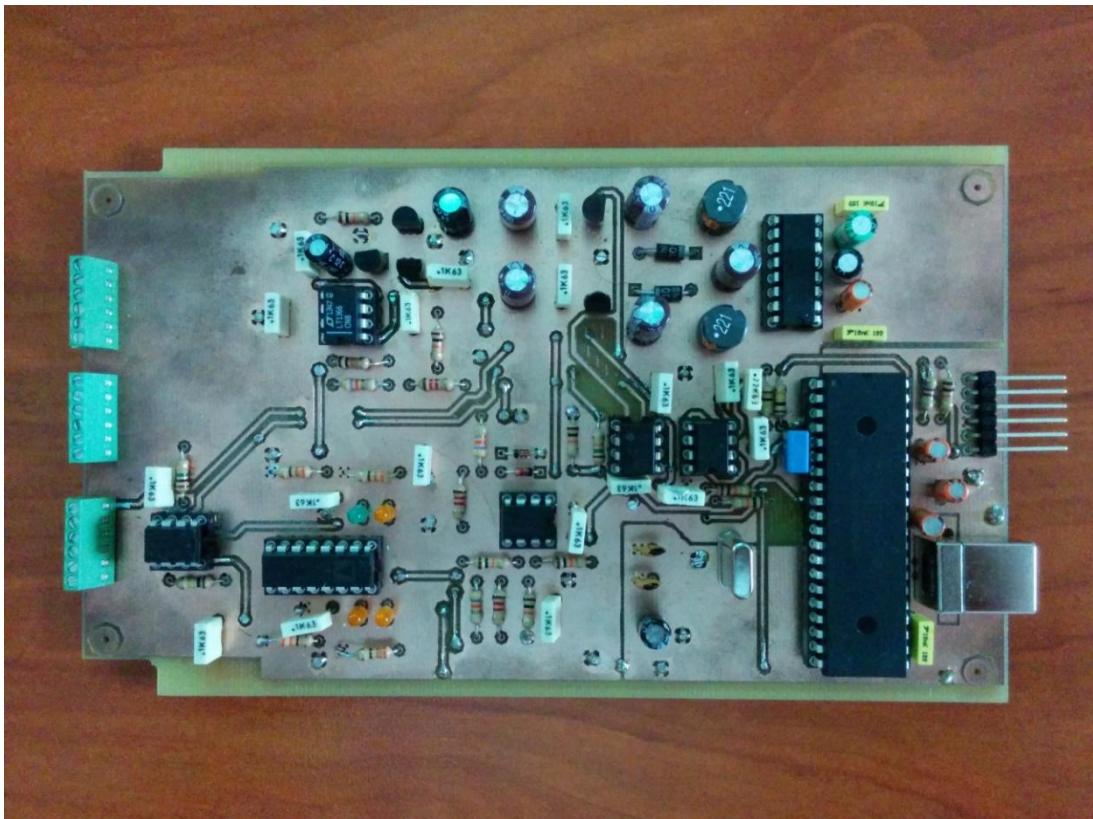


Ilustración 4.10: PCB soldada

Capítulo 5

Firmware

5.1 Entorno de programación

5.2 Firmware del pic18f4553

5.2.1 Cabecera y preprocessamiento

5.2.2 Funciones del programa

5.2.3 Función main

5.3 Programación del PIC18f4553

5. Firmware

A la hora de comenzar un proyecto de electrónica, es importante hacer todas las pruebas necesarias antes de fabricar o crear el dispositivo final. Para evitar errores que tengan difícil solución y por tanto tenga un coste económico elevado, una vez terminado el proyecto. Para intentar que no suceda esto, lo que se suele realizar es la creación de prototipos del proyecto, por ejemplo, en este caso se ha podido crear la tarjeta de adquisición de datos en *tablas protoboard* o tablas de prueba. Pero en vez de hacer eso, otra medida que podemos tomar es la de la simulación. Mediante el uso de programas, se puede llegar a simular el comportamiento de sistemas muy complicados. Esto nos hará ver los posibles fallos que pueda tener el sistema para así subsanarlos y no tener ningún tipo de complicación a la hora que tengamos el producto ya terminado.

En nuestro caso, para la simulación de la tarjeta de adquisición, se ha optado por la simulación del sistema usando el programa *Proteus*. Este programa desarrollado por Labcenter Electronics que consta de una serie de paquetes para simulación, lo hemos utilizado en nuestros estudios académicos universitarios, concretamente en la asignatura de Sistemas Electrónicos Digitales. Por lo que ya estamos más familiarizados a su interfaz gráfica, y sabemos de lo que es capaz de llegar a realizar este software.

A continuación, mostramos una imagen del esquema realizado en *Proteus* para la simulación de toda la tarjeta de adquisición:

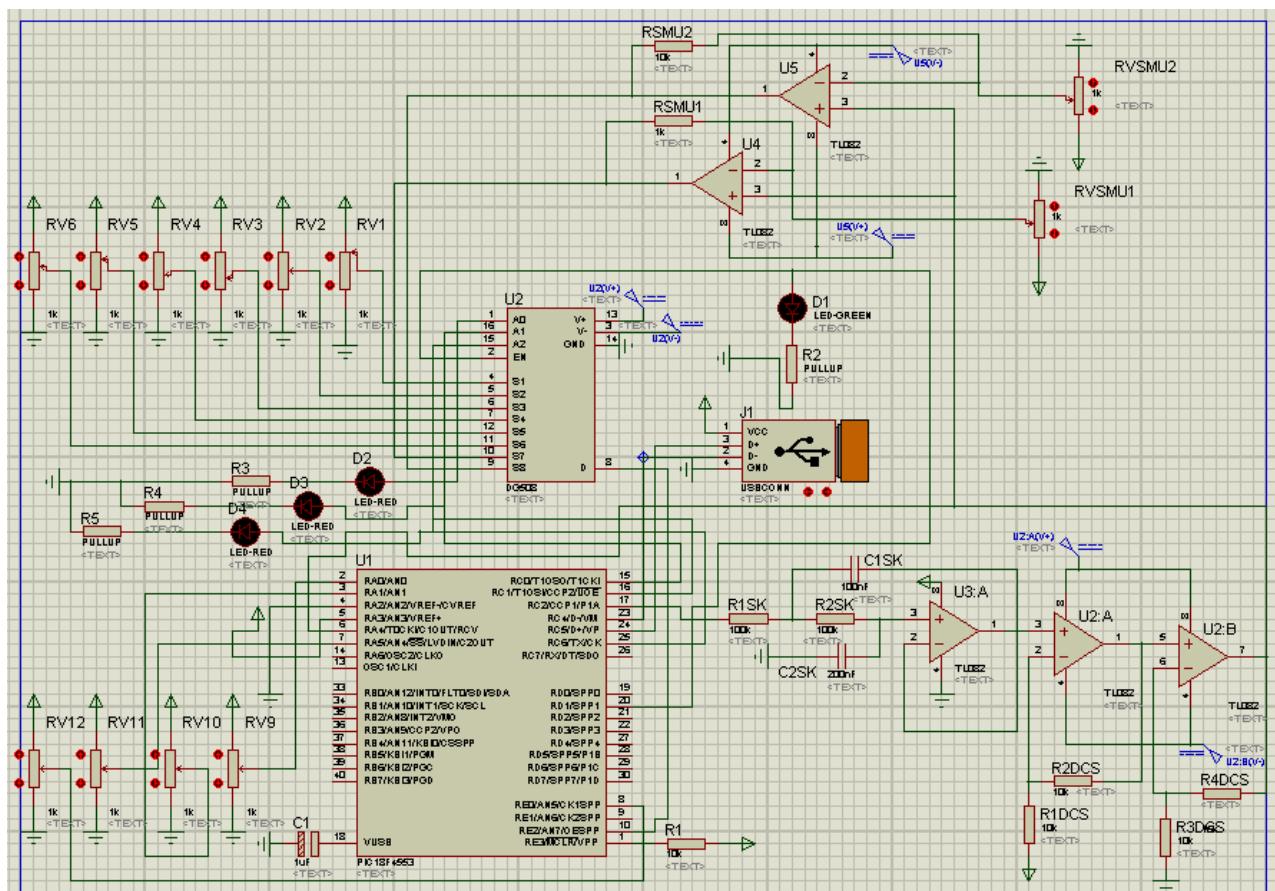


Ilustración 5.1: Esquema completo del programa Proteus

5.1 ENTORNO DE PROGRAMACIÓN

Para la programación del microcontrolador hemos utilizado el lenguaje de programación C. El entorno que hemos usado para la programación ha sido el PIC C Compiler, del fabricante CCS, *Custom Computer Services*. Este software también fue utilizado en la asignatura de Sistemas Electrónicos Digitales. Existen más entornos de programación como el que ofrece el propio fabricante de microcontroladores, Microchip. Nos referimos al MPLAB, pero por no estar familiarizados con este programa se decidió utilizar el primero de los mencionados.

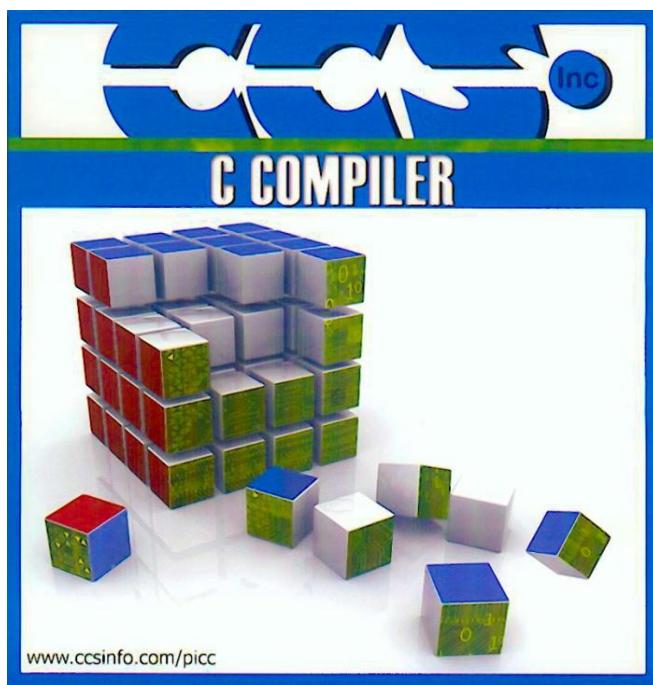


Ilustración 9: Entorno de programación CCS

La principal razón de usar este lenguaje es que es un lenguaje de alto nivel. Estos lenguajes fueron creados con el propósito de superar las desventajas del ensamblador. En lenguajes de programación de alto nivel varias instrucciones en ensamblador se sustituyen por una sentencia. El programador ya no tiene que conocer el conjunto de instrucciones o características del hardware del microcontrolador utilizado. Aunque siempre se puede insertar en el programa una secuencia escrita en ensamblador.

El lenguaje C dispone de todas las ventajas de un lenguaje de programación de alto nivel y le permite realizar algunas operaciones tanto sobre los bytes como sobre los bits (operaciones lógicas, desplazamiento etc.). Las características de C pueden ser muy útiles al programar los microcontroladores. C es un lenguaje compilado, lo que significa que los archivos fuentes que contienen el código C se traducen a lenguaje máquina por el compilador. Todas

estas características hicieron al C uno de los lenguajes de programación más populares.

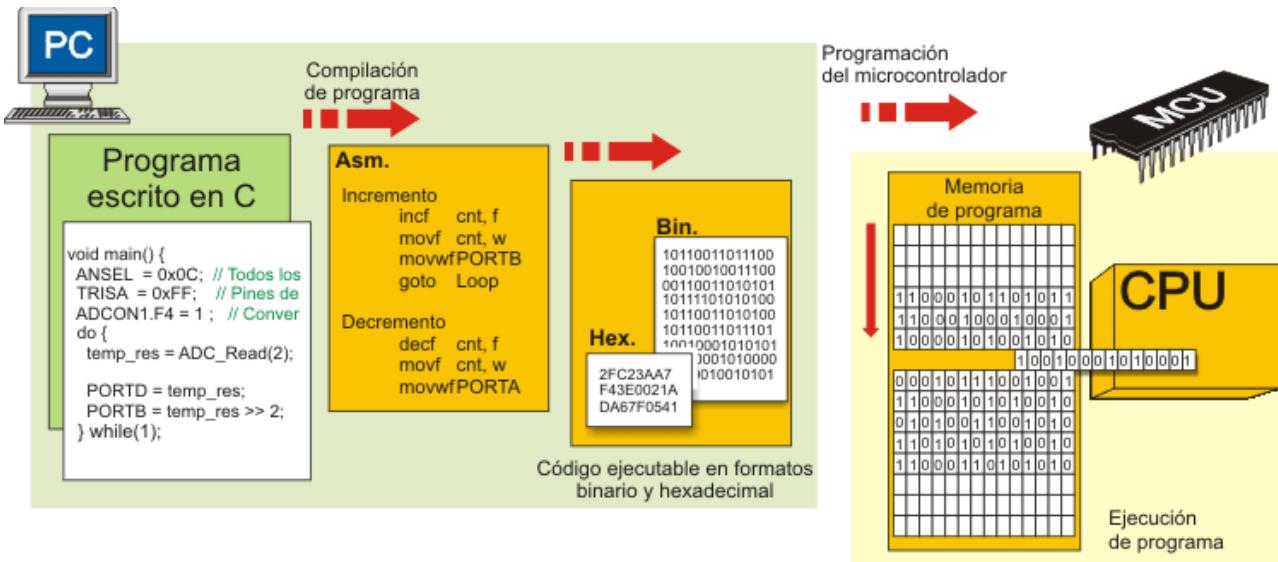


Ilustración 5.3: Compilación de un programa en C

5.2 FIRMWARE DEL PIC18F4553

En esta sección vamos a explicar todo lo referente al código usado para que el microcontrolador que hemos elegido, el PIC18F4553, realice todas las funciones que queremos efectuar.

Nuestro código se basa principalmente en una estructura if-else, la cual controlaremos mediante los bits que envíemos desde la aplicación de la tarjeta.

El siguiente diagrama de flujo muestra cómo va a operar el microcontrolador. En primer lugar vamos a configurar el microcontrolador de manera que se ajuste a nuestras necesidades, nos referimos a los *fuses* y a la configuración básica del convertidor analógico-digital. Se verifica que la conexión con el ordenador es correcta. Una vez está establecida la conexión, vamos a recibir y a leer los datos que nos envía la aplicación de control, estos datos son los canales que el usuario quiere que midamos. Proseguiremos seleccionando en el microcontrolador los canales que se van a querer medir y empezaremos a medirlos. Cada vez que mido un canal, se mandara dicho valor junto con el canal que acabamos de medir. Repetiremos este proceso para todos los canales seleccionados y para todas las iteraciones que el usuario nos haya indicado.

Optimización de una tarjeta de adquisición de datos

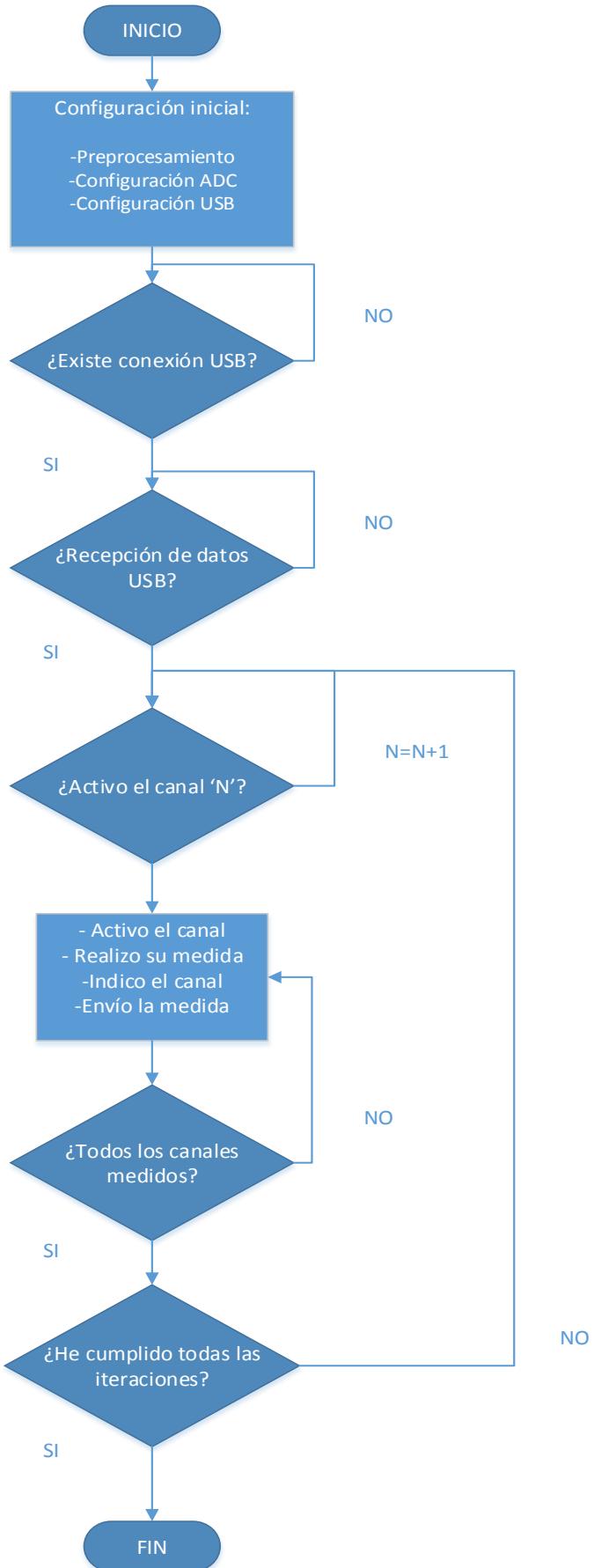


Ilustración 5.4: Diagrama de flujo del código del microcontrolador

5.2.1 CABECERA Y PREPROCESAMIENTO DE DATOS

Vamos a comenzar explicando la primera parte de cualquier código, nos referimos a la cabecera y las directivas del preprocesador.

Aquí podemos ver las directivas básicas que tiene cualquier código en este ámbito. Las sentencias que empiezan por `#include` son para incluir otros archivos para el correcto funcionamiento del proyecto. La directiva FUSES establece los distintos bits de configuración del microcontrolador. Cada microcontrolador tiene los suyos, por lo que hay que mirar en cada caso cual está disponible y cual no. En este caso indicamos el tipo de oscilador que vamos a usar, desactivamos el perro guardián, no protegemos el código, seleccionamos la salida del CCP2 por el pin B3, etc. Por último, ajustamos la frecuencia a la que vamos a trabajar, configurando su señal de reloj.

```
/*
 * TRABAJO FIN DE GRADO
 *
 * INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN
 *
 * Autor: Pablo Barroso González
 *
 * Curso: 2014/2015
 */

#include <18F4553.h>
#define ADC=12

#FUSES HSPLL,NOMCLR,NOWDT,NOPROTECT,NOLVP,USBDIV,PLL1,CPUDIV1,VREGEN,CCP2B3

#use delay(clock=48000000) // Frecuencia máxima de trabajo.

#include <string.h>
#include <stdlib.h>
#include <usb_cdc.h> // Descripción de funciones del USB.
#include <usb_desc_cdc.h> // Descriptores del dispositivo USB.
```

Ilustración 5.5: Cabecera y preprocesado del código

Cada microcontrolador tiene los suyos, por lo que hay que mirar en cada caso cual está disponible y cual no. En este caso indicamos el tipo de oscilador que vamos a usar, desactivamos el perro guardián, no protegemos el código, seleccionamos la salida del CCP2 por el pin B3, etc.

Seguiremos configurando más aspectos del microcontrolador, como la configuración de sus pines de entrada y salida, la definición de las variables globales que se van a utilizar, etc.

Optimización de una tarjeta de adquisición de datos

5.2.2 FUNCIONES DEL PROGRAMA

La siguiente parte del código, son las funciones que vamos a utilizar cuando ejecutemos el programa. Para esta aplicación se han creado 2 funciones:

```
void ordenarVector(int16 lista[], int TAM){  
    int16 temp;  
    for (i=1; i<TAM; i++){  
        temp = lista[i];  
        j = i - 1;  
        while ((lista[j] > temp) && (j >= 0)){  
            lista[j+1] = lista[j];  
            j--;  
        }  
        lista[j+1] = temp;  
    }  
  
    void PWM_init(){  
        output_low(PIN_C2); // Set CCP2 output low  
  
        setup_ccp2(CCP_PWM); // Configure CCP2 as a PWM  
  
        setup_timer_2(T2_DIV_BY_1, 18, 4);  
        setup_ccp2(CCP_PWM);  
    }  
}
```

Ilustración 5.6: Funciones del firmware

- OrdenarVector: esta función necesita que le pasemos el vector a ordenar y su tamaño. Se trata del método de inserción para ordenar los elementos de un vector, no es el más rápido que existe por ejemplo, el método *QuickSort* es más rápido pero no es viable en nuestro caso, ya que es una función recursiva y la recursividad puede desbordar la pila en ciertos microcontroladores. Requiere $O^*(n^2)$ operaciones para ordenar una lista de n elementos.
- PWM_init: esta función configura el módulo CCP2 del microcontrolador para la generación de la señal PWM. En esta función no nos vamos a preocupar todavía de la anchura del pulso, eso se configura más tarde, aquí pondremos todos los valores necesarios como la configuración del timer2, por ejemplo.

5.2.3 FUNCIÓN MAIN

La siguiente parte del código es la ejecución de la función main. A grandes rasgos, la función main podemos diferenciar 2 partes: la primera será la encargada de asegurarse que la conexión con la aplicación de la tarjeta mediante el USB funciona correctamente y obtener la información necesaria

5. Firmware

para que la tarjeta sepa que debe medir; y la segunda parte será la encargada de realizar las mediciones que nos indique la aplicación de la tarjeta.

```
while(!usb_cdc_connected()){} // espera a detectar una transmisión del PC

while(true){

    set_adc_channel(7); //Selección del canal 2 (pin RA2)
    delay_us(10);
    usb_task();

    if(usb_enumerated()){ //retorna verdadero si el dispositivo esta enumerado

        if(usb_cdc_kbhit()){ // en espera de un nuevo carácter en el buffer
            //de recepción

            can1=can2=can3=can4=can5=can6=can7=can8=can9=can10=can11=can12=can13=
            can14=can15=can16=can17=can18=can19=can20=can21=can22=can23=can24=
            can25=0;

            canal1 = usb_cdc_getc();
            canal2 = usb_cdc_getc();
            canal3 = usb_cdc_getc();
            canal4 = usb_cdc_getc();
            canal = make32(canal1,canal2,canal3,canal4);

            if(bit_test(canal,1) ==1){can1 = 1;}
            if(bit_test(canal,2) ==1){can2 = 1;}
            if(bit_test(canal,3) ==1){can3 = 1;}
```

Ilustración 5.7: Primera parte de la función main

Como se aprecia en la ilustración, hasta que no detectamos una trasmisión con la aplicación, no avanzaremos. Luego se comprueba que el dispositivo esta enumerado y nos disponemos a la espera de la recepción de los datos desde la aplicación de la tarjeta. La aplicación nos envía 4 byte de información, cada bit significa el canal que quiere que midamos, por lo que cogemos esos bits y empezamos a comprobar uno a uno, cuales son los canales que se quieren medir.

Una vez conocido los canales que vamos a necesitar medir, pasaremos a una estructura de if-else para elegir que canales se miden.

Aquí vamos a explicar cómo realiza una medida de un canal que mide tensión. Lo primero es seleccionar el canal que se va a medir, en este caso como es un canal del multiplexor ADG 408 también tenemos que indicar que canal de ese multiplexor queremos hacer la medición. En el caso que fuera un canal ligado a un SMU, se seleccionaría el ciclo de trabajo de la señal PWM. Una vez configurado correctamente el canal, realizaremos distintas mediciones y las guardaremos en un vector. Proseguiremos ordenando el vector que acabamos de escribir, utilizando el método anteriormente descrito. Después nos quedaremos con la posición del vector que esta justamente a la mitad, o lo que

Optimización de una tarjeta de adquisición de datos

es lo mismo, nos quedamos con la mediana. Por último, vemos a procesar el dato de una manera muy concreta, esto es necesario y se expondrá en el apartado posterior que nos explicará el procesado de los datos recibidos desde el microcontrolador.

```
    } else if(can2 == rdy2) {  
  
        output_low(A0);  
        output_high(A1);  
        output_low(A2);  
        output_high(EN);  
        delay_ms(100);  
  
        suma=0;  
  
        for( m=0; m < numMuestras; m++) {  
  
            muestras[m] = read_adc();  
  
        }  
  
        ordenarVector(muestras, numMuestras);  
  
        suma=muestras[7];  
  
        suma = suma + (2*33554432);  
  
        sum = suma;  
  
        printf(usb_cdc_putc, "%Lu", sum);  
  
        rdy2=0;  
  
    } else if(can3 == rdy3) {
```

Ilustración 5.8: Código para obtener una medida

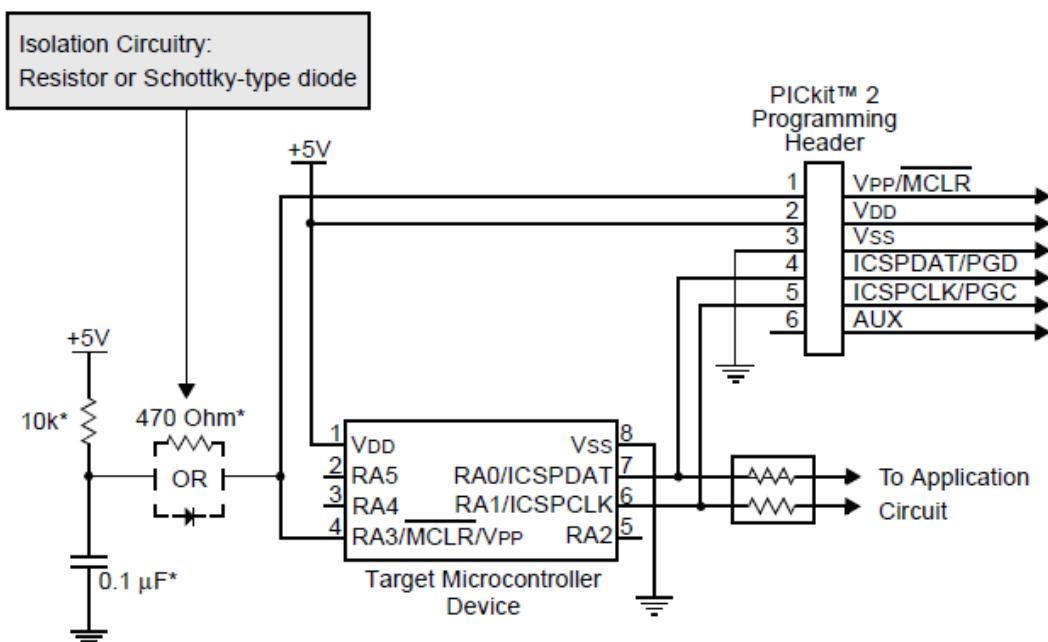
Para medir la intensidad de corriente mediante los SMU, se utilizarán estos mismos canales, lo único que tendremos que medir en dos puntos para poder sacar la diferencia de potencial, ya que la resistencia que tenemos ya la conocemos.

La mejora que se ha llevado a cabo ha sido el cambio a la hora de obtener la medida final. Anteriormente se tomaban 64 muestras, las cuales se iban sumando y al acabar se hacia la media. Se realizaba esto, para estar protegidos por el ruido que se le podía asociar a la medida. Esto llevaba mucho tiempo y no era la manera más eficiente de hacerlo. En cambio, ahora hemos pasado de medir 64 veces el canal a tan solo 16. No tenemos que estar sumando y luego dividiendo, ese tiempo lo empleamos en ordenar un vector de 16 posiciones con el método que se definió para ello. De este modo obtenemos los mismos resultados pero en menor tiempo, lo cual nos va a permitir aumentar la velocidad del proceso y poder tomar más muestras por segundo.

5.3 PROGRAMACIÓN DEL PIC18F4553

Una vez que tenemos el código necesario para el correcto funcionamiento de la tarjeta de adquisición, llega la hora de programar el microcontrolador. Existen varios métodos para la programación, en nuestro caso se va a usar la ayuda de un elemento del propio fabricante del microcontrolador, Microchip, llamado PICKit 2. [15]

El programador PICKit2, es una herramienta de programación para desarrollo de bajo costo. Es capaz de programar la mayoría de los microcontroladores y memorias seriales EEPROM de Microchip. Para usar el PICKit 2 tendremos que conectarlo a unos determinados pines del microcontrolador, como viene marcado en la siguiente imagen:



* Typical Values

Ilustración 5.9: Conexiones básicas para usar el PICKit 2

Pero aparte de su capacidad para configurar los microcontroladores, tiene otra capacidad que nos ha sido muy útil a lo largo de la realización del proyecto. El PICKit2 tiene la opción de depurar el código. Con el software que incluye, el MPLAB IDE, somos capaces de realizar modificaciones del código, programarlo y depurarlo. Sobre todo la habilidad de depurarlo lo hace especialmente atractivo, ya que nos ha servido de mucha ayuda en los momentos que no se sabía porque el código no funcionaba, o la tarjeta no llevaba a cabo las acciones que se le ordenaba.

Optimización de una tarjeta de adquisición de datos

Finalizada la parte que explica el funcionamiento del microcontrolador, viendo como ha sido programado, detallando como realiza las medidas y como las envía al ordenador. Proseguiremos en el siguiente capítulo explicando la aplicación de gestión. Será la encargada de gestionar toda la PCB, dándoleles ordenes al microcontrolador.

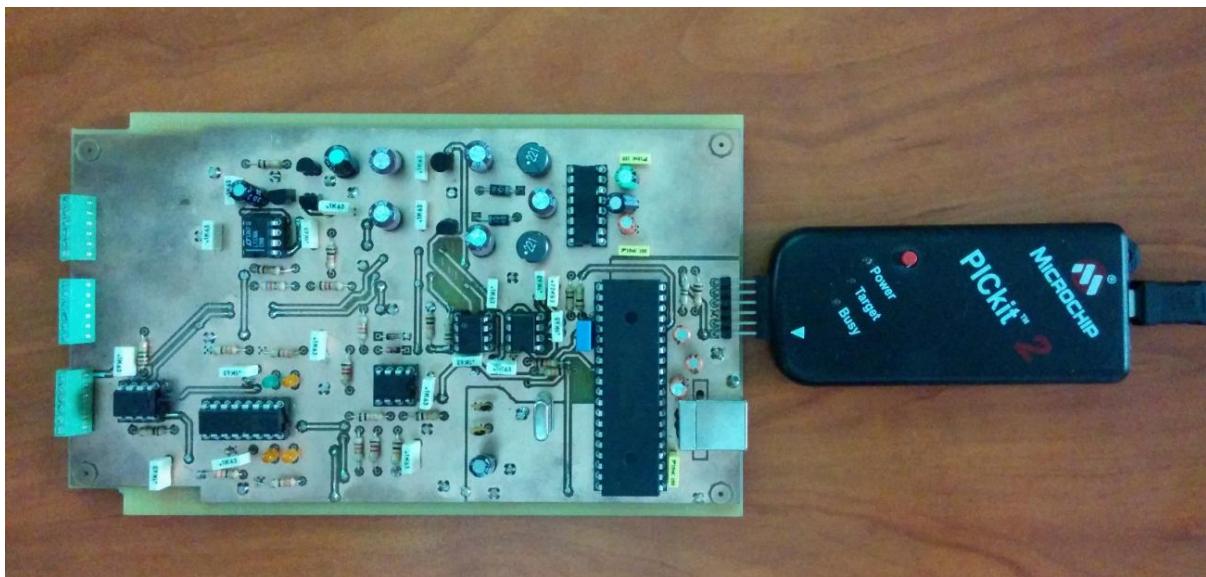


Ilustración 5.10: PCB con el PICKIT 2

5. Firmware

Capítulo 6

Aplicación de gestión

6.1 Software LabVIEW

6.2 Aplicación de gestión del PIC

6.2.1 Panel frontal

6.2.2 Diagrama de bloques

6.2.2.1 Comunicación con el microcontrolador

6.2.2.2 Generación del mensaje para el microcontrolador

6.2.2.3 Tratamiento de los datos recibidos

6.2.2.3 Escritura de los datos finales

En este capítulo vamos a explicar la aplicación que gestiona la tarjeta de adquisición de datos. Esta aplicación será la encargada de indicarle al microcontrolador que canales queremos medir, el número de veces que queremos que se midan y hasta será capaz de indicar un cierto retardo entre las muestras, ya que en algunas ocasiones no es necesario tomar muestras a su máxima capacidad, sino que tal vez nos interesa ver como evoluciona una medida a lo largo de un gran espacio de tiempo.

6.1 SOFTWARE LABVIEW

LabVIEW es una plataforma de programación gráfica que ayuda a ingenieros a escalar desde el diseño hasta pruebas y desde sistemas pequeños hasta grandes sistemas. Ofrece integración sin precedentes con software legado existente y hardware al aprovechar las últimas tecnologías de cómputo.

El software LabVIEW es ideal para cualquier sistema de medidas y control y el corazón de la plataforma de diseño de National Instruments. Al integrar todas las herramientas que los ingenieros y científicos necesitan para construir una amplia variedad de aplicaciones en mucho menos tiempo, NI LabVIEW es un entorno de desarrollo para resolver problemas, productividad acelerada y constante innovación. [16]



Ilustración 6.1: Software LabVIEW, de NI

Su principal característica es la facilidad de uso. Válido para programadores profesionales como para que personas con pocos conocimientos en programación pueden hacer programas relativamente complejos, imposibles para ellos de hacer con lenguajes tradicionales. También es muy rápido hacer programas con LabVIEW y cualquier programador, por experimentado que sea, puede beneficiarse de él.

Como se ha dicho es una herramienta gráfica de programación, esto significa que los programas no se escriben, sino que se dibujan, facilitando su comprensión. Al tener ya pre-diseñados una gran cantidad de bloques, se le facilita al usuario la creación del proyecto, con lo cual en vez de estar una gran cantidad de tiempo en programar un dispositivo/bloque, se le permite invertir

Optimización de una tarjeta de adquisición de datos

mucho menos tiempo y dedicarse un poco más en la interfaz gráfica y la interacción con el usuario final. Cada programa consta de 2 partes:

- **Panel Frontal:** El panel frontal es la interfaz con el usuario, la utilizamos para interactuar con el usuario cuando el programa se está ejecutando. Los usuarios podrán observar los datos del programa actualizados en tiempo real.
- **Diagrama de Bloques:** es el programa propiamente dicho, donde se define su funcionalidad. Aquí se colocan íconos que realizan una determinada función y se interconectan.

6.2 APLICACIÓN DE GESTIÓN DEL PIC

La aplicación que vamos a explicar es el cerebro de todo el proyecto, ya que es la encargada de gestionar toda la tarjeta de adquisición y de darles las órdenes que el usuario elija para realizar las mediciones que el usuario crea oportunas. En la ilustración 6.2 se puede observar el diagrama de flujo de la aplicación.

Comenzaremos viendo el panel frontal, así que lo primero que tiene que hacer el usuario es configurar dicho panel. Seleccionar los canales que se quieran medir, luego indicar el puerto COM donde se encuentre conectada la tarjeta de adquisición, después indicar el número de veces que se desea medir los canales previamente seleccionados, el retardo que se quiere tener entre las medidas y por último indicar el archivo donde queremos que se guarden los datos obtenidos al terminar el proceso de medición. Aquí acabaría la parte que tiene que configurar el usuario, ahora es el momento de que la tarjeta y la aplicación empiecen a trabajar.

Lo primero que se hace es comprobar que la conexión con el microcontrolador es totalmente correcta. Una vez realizado esta comprobación, la aplicación generará una cadena de bits, dependiendo de la posición y de si encontramos un 1 o un 0 significará que canales tenemos que medir y cuáles no. Y ahora la aplicación espera a que el microcontrolador le vaya pasando datos. Una vez que los recibe, tiene que procesarlo ya que a priori no sabemos de qué canal estamos recibiendo la medida (este proceso se detallará más adelante). Despues de realizar las operaciones necesarias, y teniendo el dato y el canal al que pertenece se guardará en el archivo que previamente el usuario a indicado en el panel frontal.

6. Aplicación de gestión

Existe la posibilidad de parar en cualquier momento todo el proceso, pulsando el botón STOP, situado en el panel frontal, por si no queremos esperar a que acabe todas las medidas para pararlo.

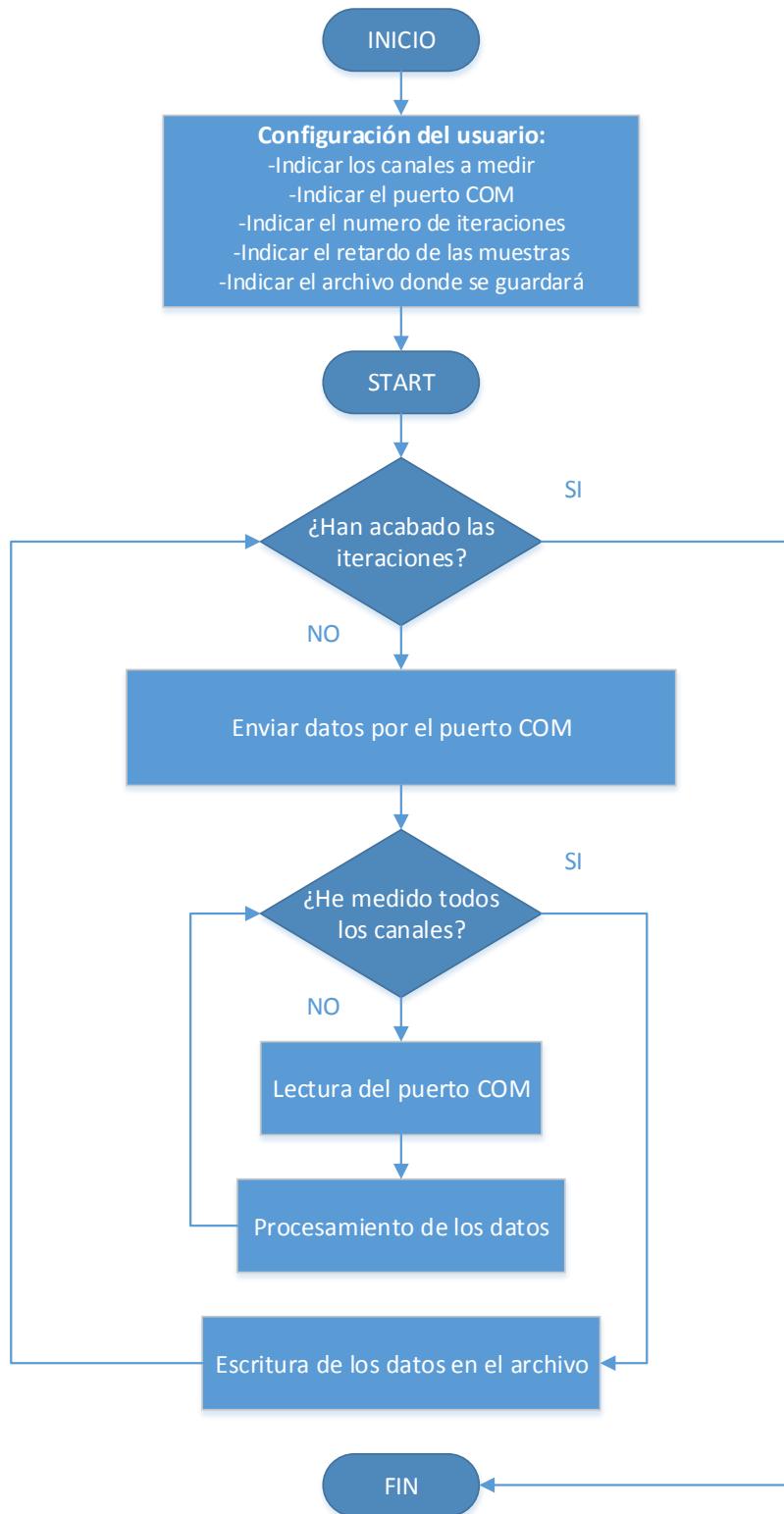


Ilustración 6.2: Diagrama de flujo de la aplicación de control

Optimización de una tarjeta de adquisición de datos

6.2.1 PANEL FRONTAL

Ya hemos comentado anteriormente lo que es el panel frontal dentro de un programa generado por LabVIEW y la función que tiene. En este apartado nos vamos a centrar en el panel creado para esta aplicación.



Ilustración 6.3: Panel frontal de la aplicación de gestión

Como se puede apreciar, el panel frontal está diseñado de tal manera de que a la hora de configurarlo siga un flujo descendente para que sea mucho más fácil para el usuario no equivocarse ni olvidarse de nada. En la primera parte, podemos observar una serie de botones, los cuales indicarán al microcontrolador que canales desea el usuario medir. Despues seguimos con una serie de opciones que tenemos que llenar, tales como el puerto COM, las repeticiones de las medidas, etc. Por ultimo, nos encontramos dos botones, el botón de la derecha, el botón START, dará la orden para comenzar con las mediciones. En cambio, el otro botón, el botón STOP, parará de tomar medidas en cualquier momento que el usuario requiera su detención.

6.2.2 DIAGRAMA DE BLOQUES

En este apartado se van a explicar las distintas bloques o subsecciones que conforman la aplicación final, vamos a diferenciar 4 partes: la

6. Aplicación de gestión

comunicación con el microcontrolador, la generación del mensaje para el microcontrolador, el procesado de los datos recibidos y la escritura de los datos finales.

6.2.2.1 Comunicación con el microcontrolador

La comunicación serie se inicia con el bloque de configuración de los parámetros de la conexión. Estos parámetros configuran la tasa de transmisión (en Baudios), número de bits del dato, paridad, control de flujo, bit de parada, entre otros. Estas opciones de configuración pueden ser visibles o no para el usuario. En este proyecto los parámetros no son visibles al usuario ya que los valores establecidos son óptimos y mostrarlos al usuario sólo crearía confusión y posibles errores.

Una vez iniciada y configurada la comunicación serie, se procede a enviar datos mediante el bloque de escritura y a recibirlos mediante el bloque de lectura. En la ilustración 6.3 se puede observar una estructura típica de una comunicación serie. El bloque de escritura se puede identificar por la letra "W"(write), mientras que el de lectura se identifica por la letra "R"(read). Entre estos dos bloques se introduce un retardo para dar tiempo a los datos a llegar antes de iniciar el proceso de lectura. Por último se cierra la comunicación una vez se ha analizado el proceso de transmisión mediante el bloque identificado con la letra "C"(close).

Entre todos estos bloques existen dos líneas de conexión, una que identifica el puerto de comunicaciones utilizado y que conecta los bloques entre sí y una línea de comprobación de errores. En caso de producirse cualquier error a lo largo de la comunicación se mostrará un mensaje con el motivo del problema.

Optimización de una tarjeta de adquisición de datos

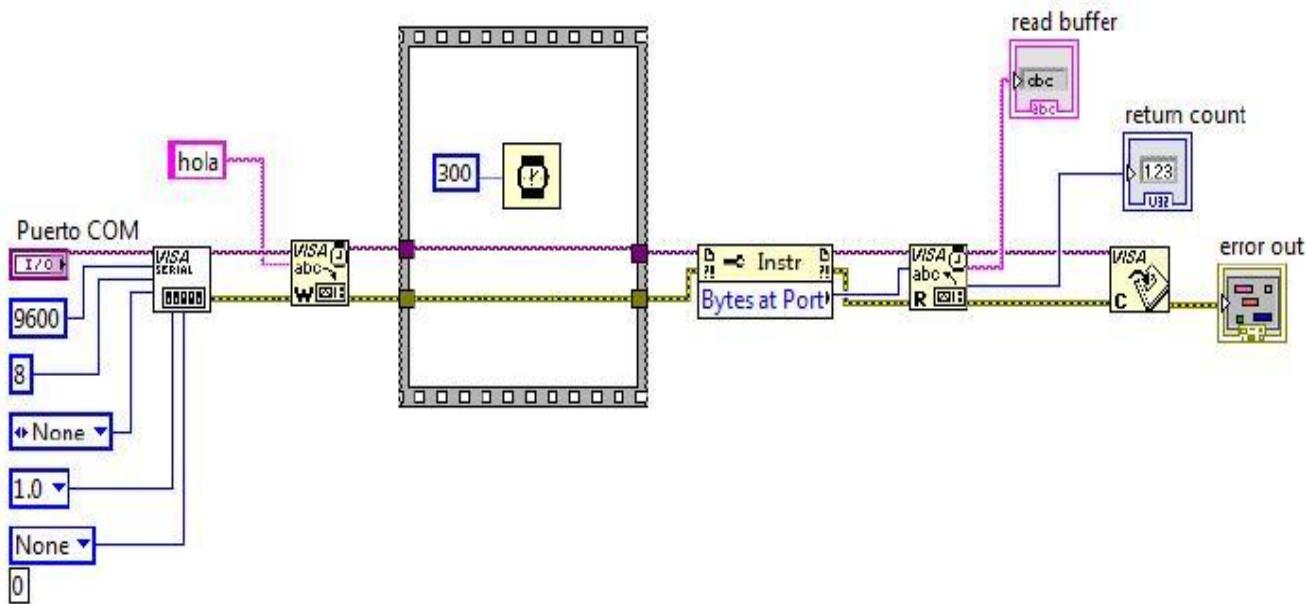


Ilustración 6.4: Ejemplo de comunicación entre PC y microcontrolador

6.2.2.2 Generación del mensaje para el microcontrolador

En esta apartado se explicara como es el proceso de transmitir las órdenes del usuario, mediante la configuración del panel frontal, al microcontrolador que es el encargado de realizar las medidas.

Como se vió en el panel frontal, tenemos una serie de botones para elegir el canal que deseamos medir. Estos botones son los encargados de crear una señal booleana, es decir 1 si está presionado y 0 si no lo está. Una vez tenemos todas estas señales, las juntaremos y crearemos una cadena de bits, concretamente, de 32 bits. Cada bit tiene su significado, el cual explicamos a continuación:

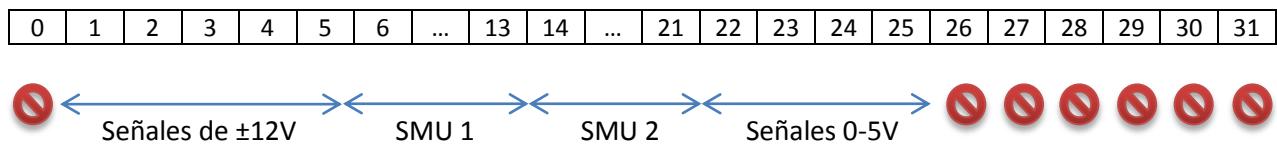


Ilustración 6.5: Descripción del significado de cada bit

Podemos observar cómo dejamos un bit sin ninguna función, a partir del segundo todos son significativos. Los cinco siguientes sirven para indicar que canal, de los que pueden medir $\pm 12V$, queremos medir. Los siguientes 18 bits sirven para indicar que queremos medir, los ocho primeros para el SMU 1 y los ocho siguientes para el SMU2. Le siguen cuatro bits para indicar que canales, de los que miden de 0-5V, queremos seleccionar. Por último dejaremos los seis bits restantes sin uso.

6. Aplicación de gestión

Una vez tenemos la cadena de bits, se la enviaremos al microcontrolador en 4 bytes. Para ello tenemos que descomponer la cadena previamente.

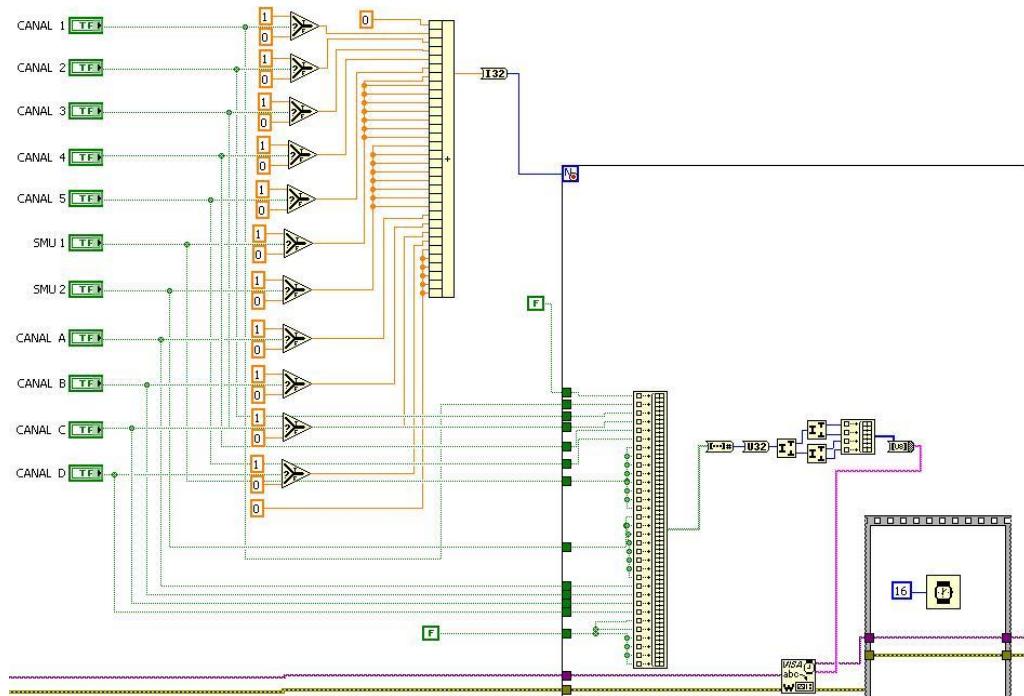


Ilustración 6.6: Diagrama de la parte encargada de generar el mensaje al microcontrolador

6.2.2.3 Tratamiento de los datos recibidos

Los datos recibidos del microcontrolador deben ser procesados, ya que si recordamos, el microcontrolador envía 12 bits (resolución del ADC) con un valor perteneciente al intervalo [0,4095]. Aparte de estos 12 bits, se envían más para saber la medición de donde proviene. A continuación vamos a explicar cómo se procesan los datos:

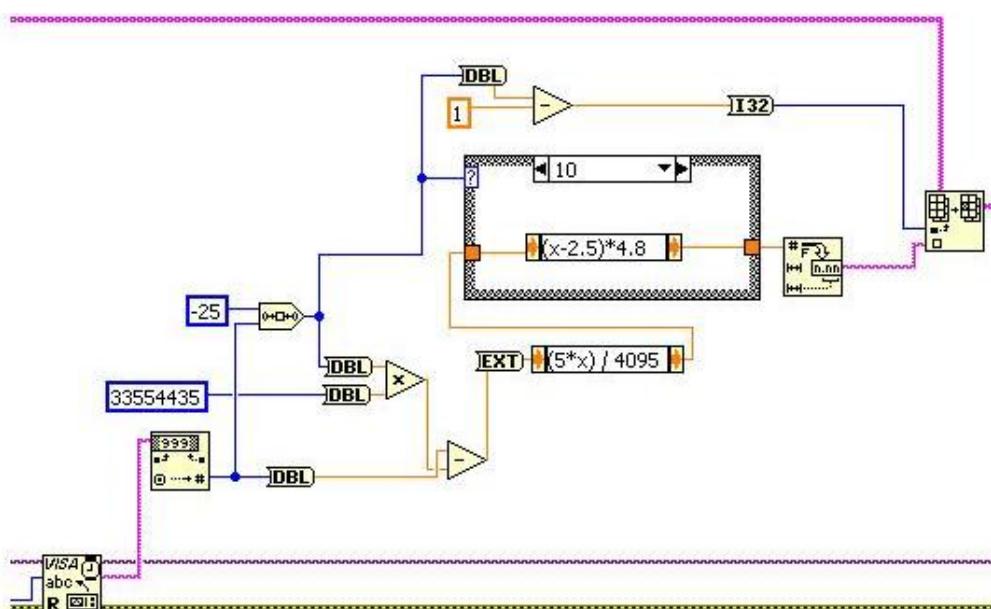


Ilustración 6.7: Diagrama del tratamiento de los datos recibidos

Optimización de una tarjeta de adquisición de datos

Lo primero que hacemos es convertir el valor recibido a un número, ya que se envía en una cadena de caracteres. Después desplazamos 25 bits para saber el canal del que procede la medición, ya que esa información se encuentra aquí y se resta el número del canal multiplicado por 33554435 para tener el valor original que proporcionó el convertidor analógico-digital del microcontrolador. Con este valor, pasamos a obtener el valor original de 0-5V y para acabar pasamos por una estructura case, que dependiendo de si el canal procedía de unos de los canales de $\pm 12V$ o de un SMU, tendremos que convertir el valor de 0-5V a $\pm 12V$. En el caso de los canales de 0-5V, la estructura case no aplicaría ninguna fórmula matemática.

Lo único que nos queda es pasar el valor de la medida a una cadena de caracteres que, junto a su número de canal pasarán a la siguiente fase, la de guardar los datos en un archivo.

6.2.2.3 Escritura de los datos finales

La última parte que queda es la explicación de cómo guarda los datos en un archivo. Para esta parte se han creado dos ejecutables, Tarjeta_De_Adquisicion_txt.exe y Tarjeta_De_Adquisicion_excel.exe. Se han creado estos ejecutables para que el usuario no tenga la necesidad de instalar el LabVIEW para poder usar la aplicación. La diferencia entre ellos es el formato final en el que se van a presentar los datos, si los queremos en un archivo de texto o en una hoja de cálculo.

En las dos versiones actuaremos de la misma manera, lo único que cambia es el final, a la hora de guardar y presentar los datos. Comenzaremos creando un archivo donde se van a guardar los datos como muestra la siguiente imagen:

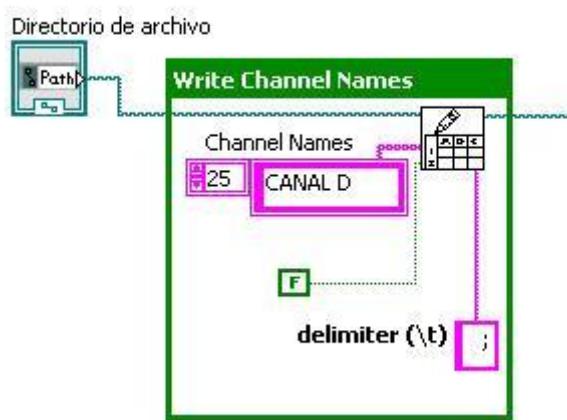


Ilustración 6.8: Diagrama que muestra la creación del archivo

Este archivo será el utilizado para almacenar todos los datos que posteriormente necesitaremos guardar.

6. Aplicación de gestión

Y es en el apartado que hemos explicado anteriormente, donde se explica el tratamiento de los datos, donde vamos a usar este archivo. Añadiremos un valor para guardarlos en triples, guardaremos tres valores. Ya se mencionó que se guardaría la medida junto a su canal, pero a esto debemos sumarle la hora exacta a la que se realiza la medida para completar la información.

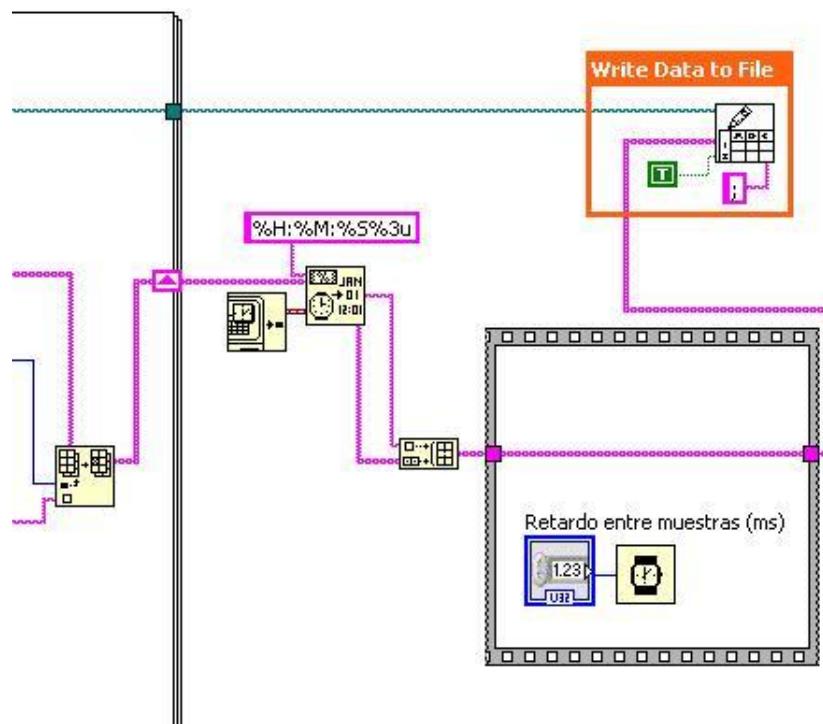


Ilustración 6.9: Diagrama de bloques del almacenamiento de los datos

Después de tener todos los datos que vamos a almacenar, nos dispondremos a guardar los datos en el archivo que previamente creamos en la primera parte, no sin antes aplicar el retardo de tiempo que indicamos en la parte del panel frontal. Este retardo nos permite espaciar las medidas a lo largo del tiempo, sin tener que medir a la máxima velocidad que nos ofrece la tarjeta de adquisición. Para finalizar, mostramos cómo sería la parte final si quisiéramos guardar los datos en una hoja de cálculo, tal y como muestra la figura de la página siguiente.

Como se puede apreciar, es exactamente igual que antes, pero se le añade un subcircuito para poder guardar los datos en una hoja de cálculo.

Una vez terminado este capítulo, y sabiendo cómo funciona la aplicación de gestión, ya sabemos cómo se toman las medidas y como se procesan en el ordenador. Los siguientes capítulos serán para comentar los resultados obtenidos y hacer un balance total del proyecto.

Optimización de una tarjeta de adquisición de datos

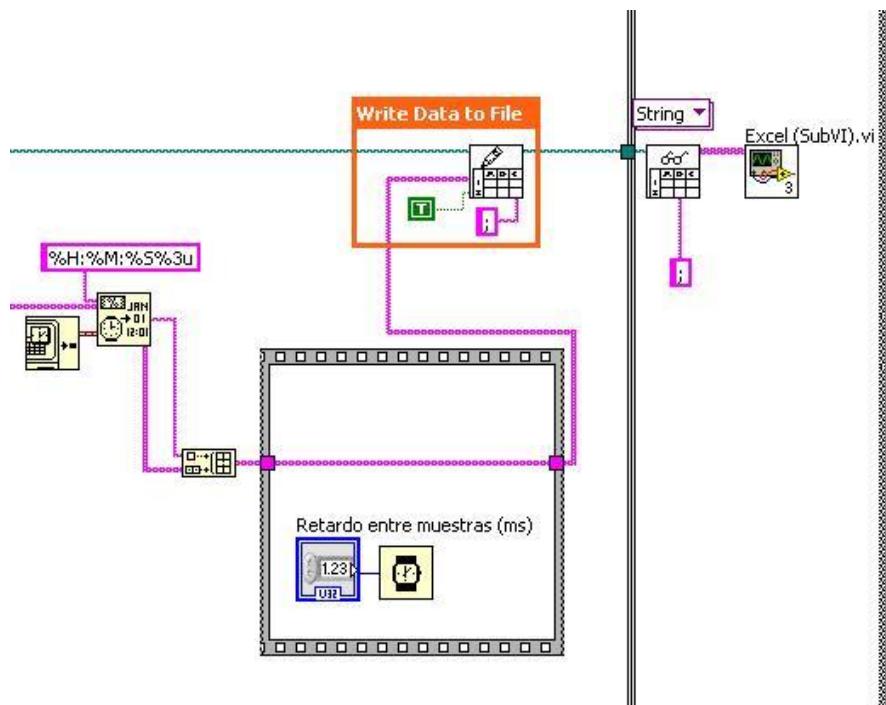


Ilustración 6.10: Diagrama de bloques del almacenamiento de los datos usando Excel

6. Aplicación de gestión

Capítulo 7

Especificaciones y manual de usuario

7.1 Especificaciones

7.2 Manual del usuario

7.2.1 Conexión física entre la placa y las señales a medir

7.2.2 Configurar la aplicación de gestión

7.2.3 Configuración del retardo

7.1 ESPECIFICACIONES DEL PROYECTO

En este apartado, vamos a resumir las características más relevantes de la tarjeta de adquisición de datos que hemos construido:

- 4 canales para medir tensiones entre 0 y +5V
- 5 canales para medir tensiones entre ±12V
- 2 canales para medir intensidad de corriente mediante los SMU
- Salidas de tensión de alimentación de +5V, +12V y -12V
- Resolución de 12 bits para todos los canales
- Dispositivo portable
- Conexión mediante PC por USB
- Plug-and-play
- No necesita alimentación externa, solo estar conectado al PC
- No realiza mediciones simultáneas
- Frecuencia máxima de muestreo de 62,5Hz (midiendo un único canal)
- Posibilidad de cambiar su firmware con la ayuda de PICKit 2
- Almacenamiento de datos en fichero de texto
- Interfaz de usuario fácil de usar

7.2 MANUAL DE USUARIO

Siempre que adquirimos cualquier producto, viene con unas instrucciones de uso o un manual para aprender a usarlo adecuadamente. Aquí también vamos a dar unas pautas al usuario como debería usar el dispositivo de una manera correcta y segura.

7.2.1 CONEXIÓN FÍSICA ENTRE LA PLACA Y LAS SEÑALES A MEDIR

Para conectar la tarjeta de adquisición con las tensiones o corrientes de intensidad, se han colocado una serie de bloques terminales PCB, el cual tienen un tornillo en la parte superior para poder fijar el cable y asegurar que se produzca el contacto. En estos bloques terminales tendremos tanto los canales de entrada como los de salida. Lo único que vamos a necesitar es un destornillador de punta plana de como máximo 1,6 mm, ya que es el tamaño del diámetro del tornillo. Se recomienda tener la tarjeta de adquisición no conectada cuando se realicen los cambios en los diferentes terminales.

Optimización de una tarjeta de adquisición de datos

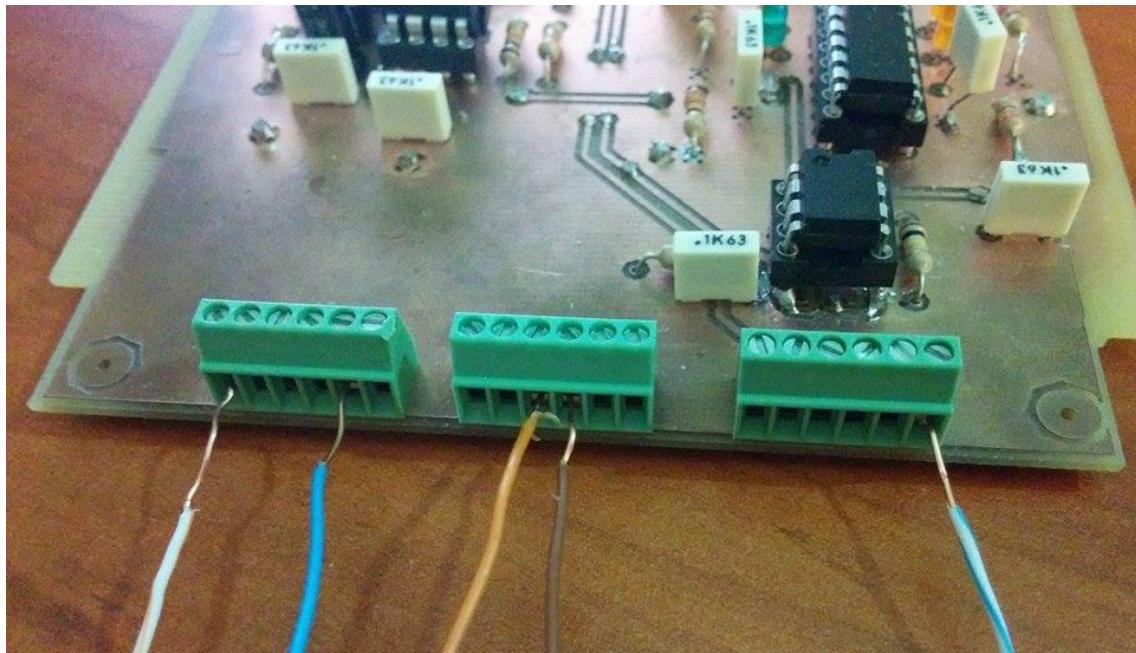


Ilustración 7.1: Bloques conectores de la PCB con algunos cables

7.2.2 CONFIGURAR LA APLICACIÓN DE GESTIÓN

Como se explicó en anteriores apartados, para que funcione el dispositivo se tiene que indicar mediante la aplicación de gestión de la tarjeta una serie de elementos para que todo funcione correctamente. A continuación mostramos una lista con los parámetros que el usuario tiene que configurar:

- Seleccionar los canales que desea medir
- Seleccionar el puerto COM en donde está conectada la tarjeta
- Seleccionar el número de muestras que se desean medir
- Seleccionar el retardo que se quiere tener entre las muestras
- Seleccionar el archivo donde se desea guardar todas las muestras

Una vez elegido todo lo anteriormente citado, estamos listos para comenzar a medir. Para ello, solo tenemos que asegurarnos que presionamos el botón START, botón notablemente más grande que cualquier otro en el panel del usuario. Asegurarse de que la flecha blanca en la parte superior izquierda de la pantalla está presionada, si la flecha está en blanco significa que no se está ejecutado y está a la espera de arrancar, si esta flecha por lo contrario esta negra y rasgada, significa que el programa está ejecutándose y si se desea parar dicha ejecución tenemos el botón STOP, situado al lado del botón START para parar la ejecución en cualquier momento.

7. Especificaciones y manual de usuario



Ilustración 7.2: Interfaz gráfica para el usuario

7.2.3 CONFIGURACIÓN DEL RETARDO

Como se dijo en las especificaciones, la tarjeta de adquisición puede llegar a una frecuencia de muestreo de 66,6Hz. Debido a que el microcontrolador solo tiene un convertidor analógico-digital no es posible realizar varias medidas simultáneamente. Y debido a este problema hay que tener en cuenta la siguiente fórmula matemática, para los casos en que la frecuencia de muestreo sea un problema a tener en cuenta:

$$\text{Retardo mínimo} = 0,016 * N_{\text{Canales}}$$

Donde N_{Canales} representa el número de canales que se quieren medir. Esto quiere decir que si vamos a medir 3 canales, tenemos que tener en cuenta que el tiempo que va a trascurrir para medir el mismo canal no será de 62,5Hz, si no que será de tres veces mayor. Por lo que si queremos es medir una señal con un poco de frecuencia, lo recomendable es medir solo ese canal para no aumentar el retardo a la hora de realizar las mediciones.

Cabe destacar que cuando seleccionamos un SMU, se van a realizar 8 medidas. Por lo que recomendamos que si la frecuencia de muestreo es vital para la realización de las medidas, no utilicen los SMU, ya que bajará considerablemente la frecuencia de muestreo.

7.2.4 CONFIGURACIÓN DE LAS TENSIONES DE LA SMU

Cuando seleccionamos medir por un canal donde se haya conectado una SMU, el microcontrolador tiene programado 4 valores de tensión con los que tiene que trabajar para tomar las medidas. Estas medidas estan fijadas y para el usuario, desde el panel frontal, es imposible que las pueda cambiar.

La única manera de cambiarlas sería reprogramando el microcontrolador y cambiando las tensiones que están fijadas por otras. Para ello el usuario necesitaría usar el PICKit2 que se detalló en los apartados anteriores.

Se recomienda que no se cambie esto a no ser que sea de vital importancia tener mayor precisión para realizar una curva característica I/V con puntos en una determinada zona.

7. Especificaciones y manual de usuario

Capítulo 8

Resultados obtenidos

8.1 Resultados obtenidos

8.1.1 Experiencia 1: Medir tensión en continua

8.1.2 Experiencia 2: Medir sensores de luz y temperatura

8.1.3 Experiencia 3: Medir señales alternas

8.1.4 Experiencia 4: Utilización del SMU

8.1 RESULTADOS OBTENIDOS

En este apartado vamos a mostrar algunas medidas que hemos realizado para la comprobación del dispositivo. Vamos a realizar 3 pruebas distintas: ver que puede medir varios canales, medir señales alternas y el funcionamiento de las SMUs.

8.1.1 EXPERIENCIA 1: MEDIR TENSIONES CONTINUAS

En este circuito lo que se está midiendo son las tensiones en continua de un divisor de tensión. Se ha medido tanto la tensión de alimentación que es de unos 5V aproximadamente, y luego otros dos canales miden el punto donde se conectan las dos resistencias. Este es el resultado:

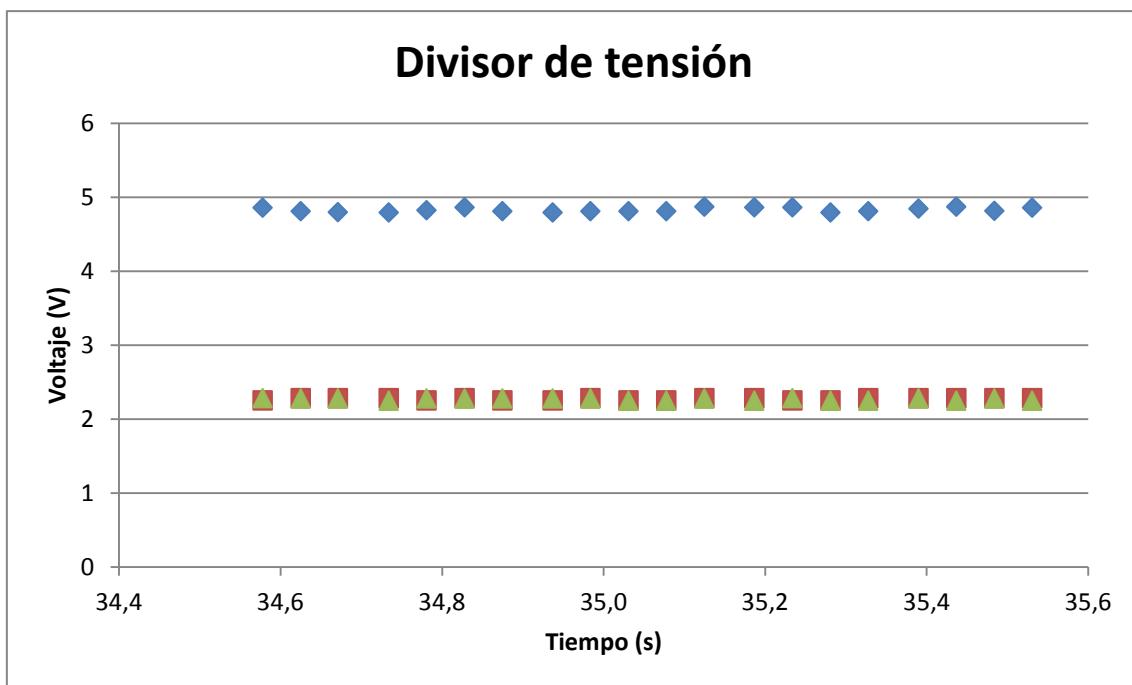


Ilustración 8.1: Gráfica de la medida del divisor de tensión

Donde podemos observar que se ha medido a lo largo de un segundo y hemos obtenido 19 muestras. En este caso en particular, no nos vamos a preocupar del tiempo de muestreo, ya que son tensiones continuas. Se ve claramente como las tensiones son prácticamente constantes, no son exactas porque en este caso no se usó el canal de la tarjeta, si no que se usó el cargador de un móvil para realizar estas mediciones.

8.1.2 EXPERIENCIA 2: MEDIR SENSORES DE LUZ Y TEMPERATURA

Si hemos medido antes unas tensiones continuas en un breve espacio de tiempo, lo que se va a hacer ahora es medir en un gran espacio de tiempo

Optimización de una tarjeta de adquisición de datos

como van a variar dos sensores: uno de luz y otro de temperatura. Para ello, hemos utilizado una fotorresistencia y el integrado lm 35, de National Semiconductor. [17]

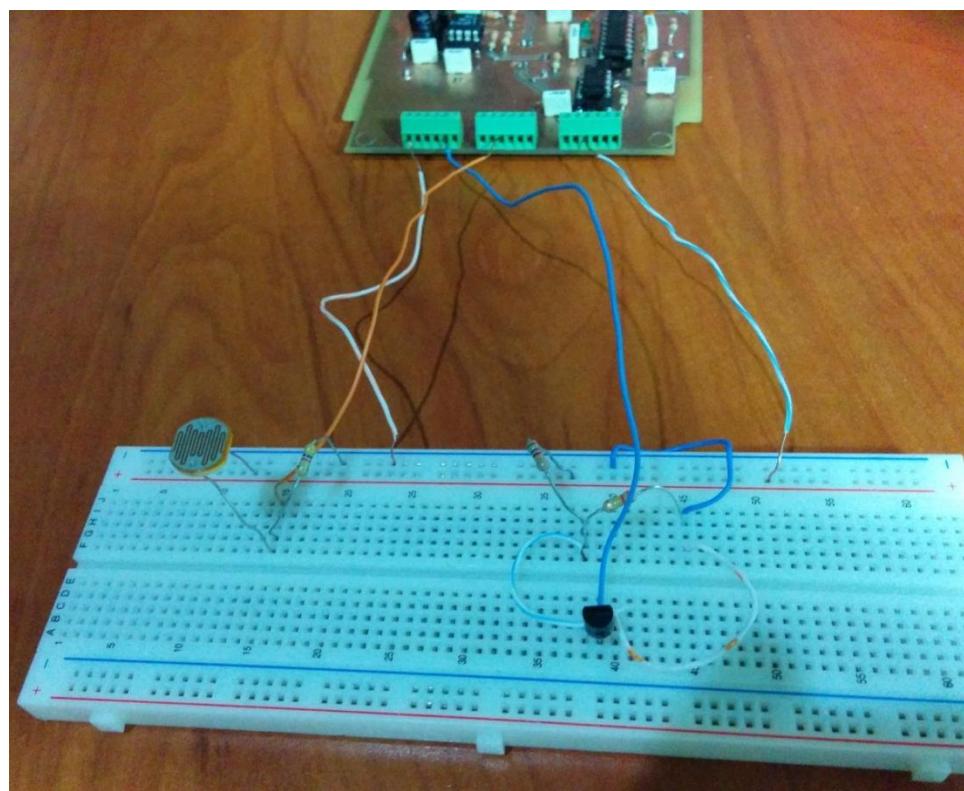


Ilustración 10 Implementación para medir luz y temperatura

Intensidad lumínica y temepratura al atardecer en Granada

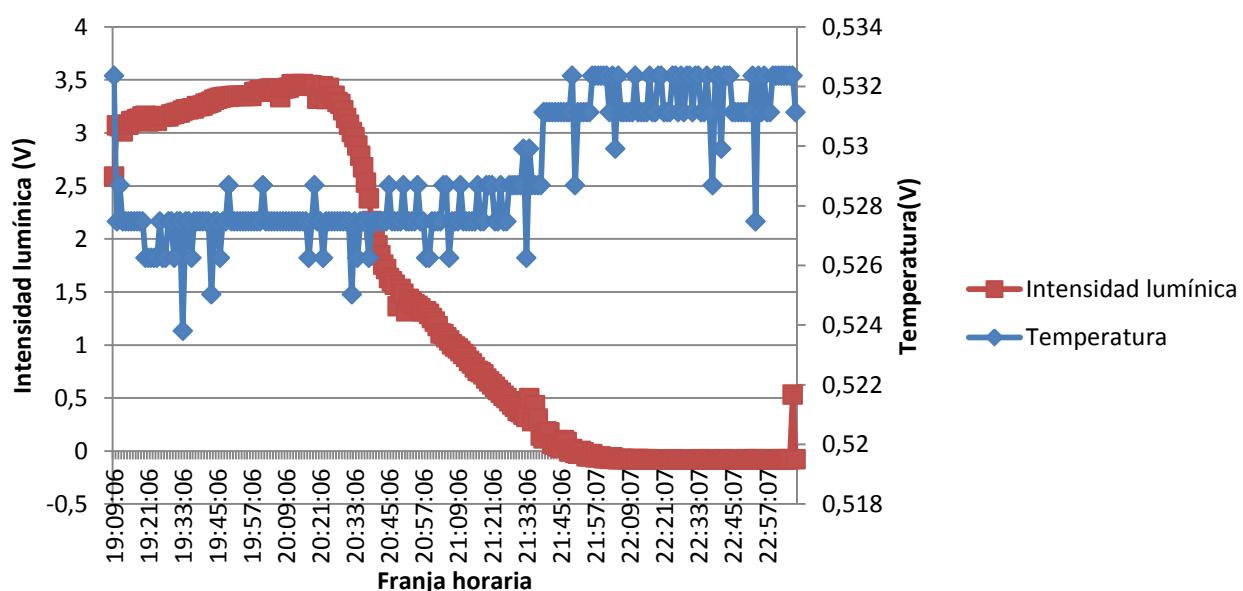


Ilustración 8.3: Gráfica de la medida de luz y temperatura

8. Resultados obtenidos

Como era natural, conforme avanza la tarde, va decreciendo la intensidad lumínica. Lo que parece que no es lógico, es el aumento de la temperatura en horas donde ya no hay sol.

Aquí se muestra un fragmento de los datos que se obtuvieron para realizar la gráfica anterior:

TIEMPO	CANAL 5	CANAL A	CANAL B	CANAL C	CANAL D
19:09:06	0,000000	0,000000	0,000000	2,587302	0,532357
19:10:06	0,000000	0,000000	0,000000	3,069597	0,527473
19:11:06	0,000000	0,000000	0,000000	3,062271	0,528694
19:12:06	0,000000	0,000000	0,000000	3,01221	0,527473
19:13:06	0,000000	0,000000	0,000000	3,075702	0,527473
19:14:06	0,000000	0,000000	0,000000	3,075702	0,527473
19:15:06	0,000000	0,000000	0,000000	3,10989	0,527473
19:16:06	0,000000	0,000000	0,000000	3,114774	0,527473
19:17:06	0,000000	0,000000	0,000000	3,126984	0,527473
19:18:06	0,000000	0,000000	0,000000	3,135531	0,527473
19:19:06	0,000000	0,000000	0,000000	3,155067	0,527473
19:20:06	0,000000	0,000000	0,000000	3,162393	0,526252
19:21:06	0,000000	0,000000	0,000000	3,108669	0,526252
19:22:06	0,000000	0,000000	0,000000	3,123321	0,526252

Ilustración 8.4: Algunos datos de la gráfica de los sensores de luz y temperatura

8.1.3 EXPERIENCIA 3: MEDIR SEÑALES ALTERNAS

Pasaremos ahora a realizar medidas de señales en alterna para poder demostrar que la tarjeta de adquisición puede medir señales de una cierta frecuencia. Las siguientes señales que se van a ver son señales de 2Hz. Las mediciones se han realizado mediante un canal de $\pm 12V$.

Optimización de una tarjeta de adquisición de datos

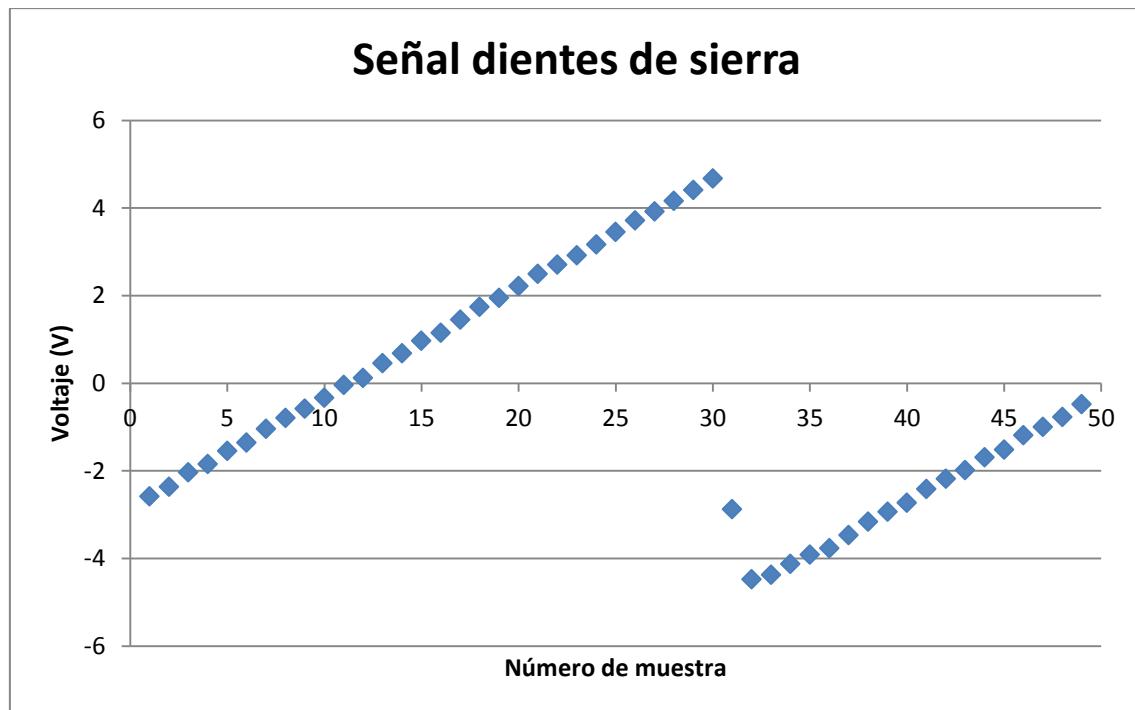


Ilustración 8.5: Gráfica del resultado de medir una señal dientes de sierra

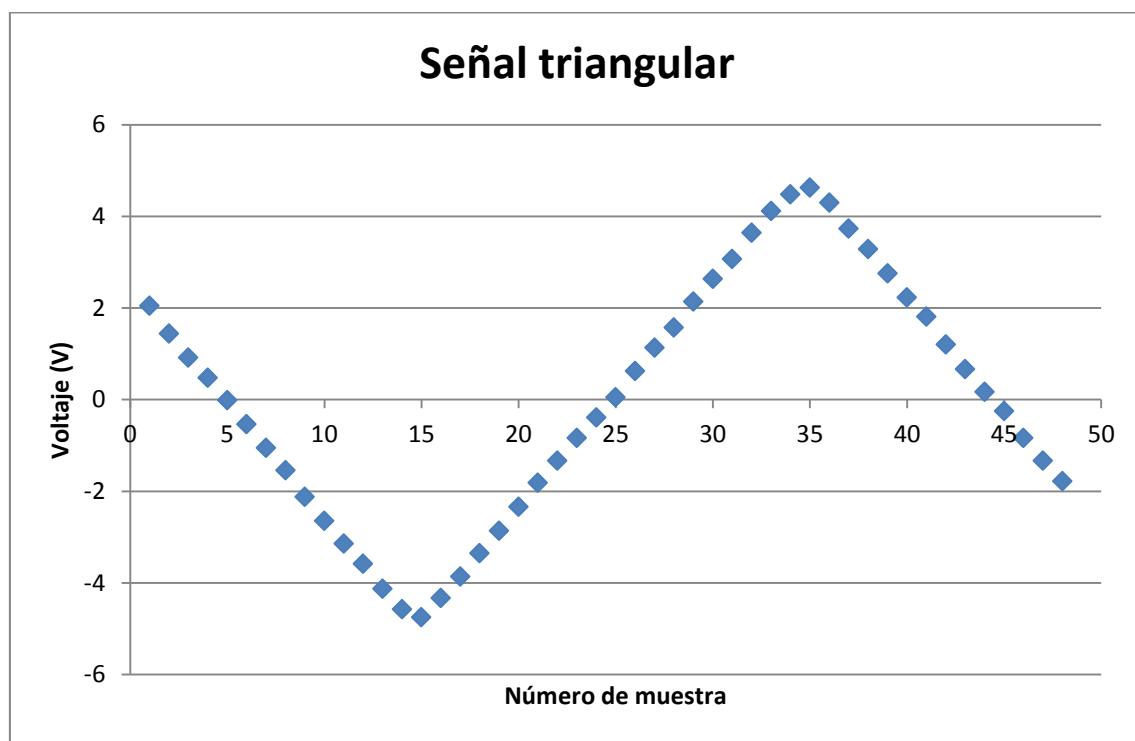


Ilustración 8.6: Gráfica del resultado de medir una señal triangular

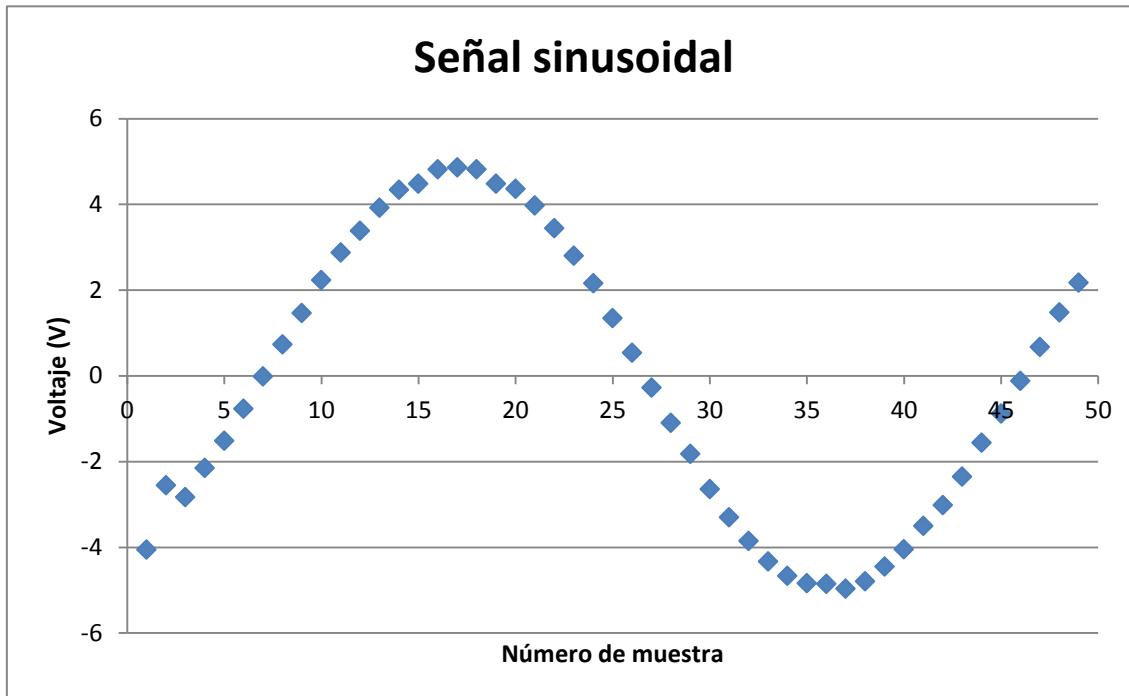


Ilustración 8.7: Gráfica del resultado de medir una señal sinusoidal

Como se ven en las tres gráficas, podemos comprobar que se ha realizado un muestreo adecuado de todas ellas. Sabemos que podemos muestrear señales con una frecuencia mayor a 2 Hz, pero se ha elegido esta frecuencia para que se vea claramente la forma de cada una de las señales elegidas.

8.1.4 EXPERIMENTO 4: UTILIZACIÓN DEL SMU

Por último, vamos a mostrar los datos obtenidos usando los SMU. Con la ayuda del microcontrolador iremos sacando una señal PWM, con un determinado duty, dependiendo del duty a la hora de ser filtrada esta señal vamos a tener una mayor o menor tensión. Después pasará por la etapa de acondicionamiento, que será la encargada de pasar la señal de 0 a 5 V a ± 12 V. El circuito para este experimento es muy simple, tan solo vamos a conectar la salida de +12V de la tarjeta a una de las patillas de la resistencia, y en la otra conectaremos directamente con el canal SMU1 del dispositivo. En primer lugar vamos a ver la curva I-V que obtenemos al conectar una resistencia de $321\text{k}\Omega$ al SMU1, que tiene una resistencia de $100\text{k}\Omega$. Estos son los resultados obtenidos:

Optimización de una tarjeta de adquisición de datos

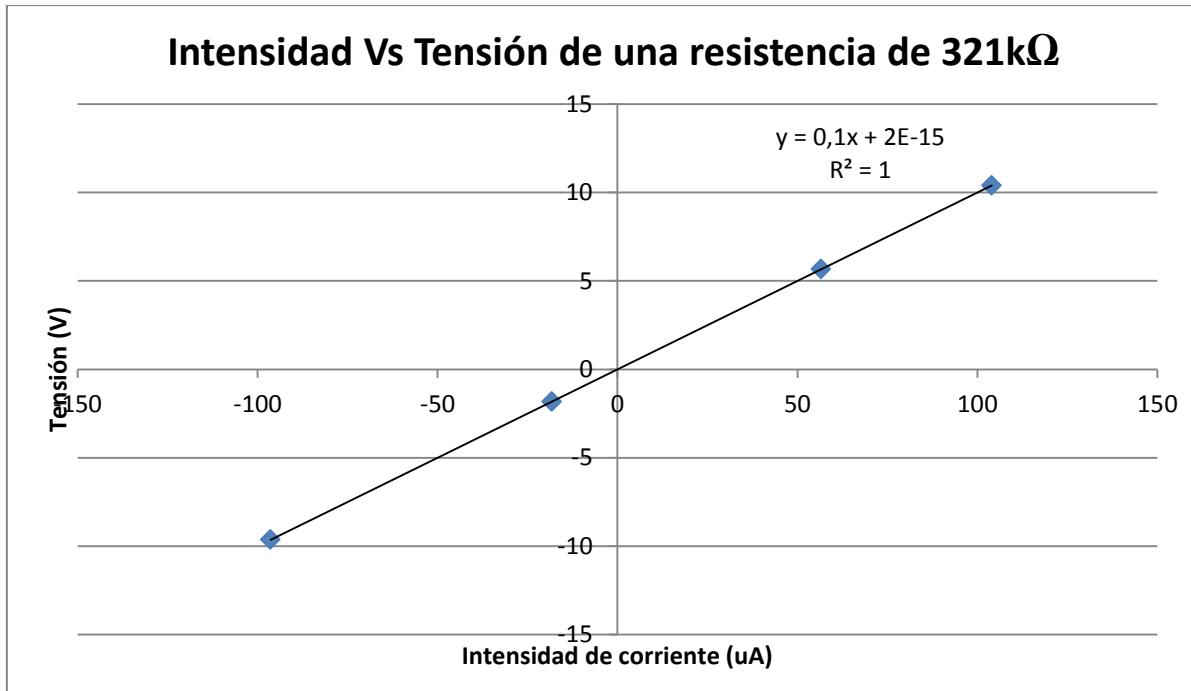


Ilustración 8.8: Gráfica del resultado de medir la intensidad Vs la tensión de la resistencia de 321kΩ

Como era de esperar, el comportamiento que tiene intensidad de corriente en función de la tensión es totalmente lineal.

Podemos concluir con todas estas mediciones que todos los canales funcionan perfectamente, ya sea los canales de 0 a +5V, ±12V o los SMU, como si son en un espacio breve de tiempo o muy prolongado, imponiendo un cierto retardo cuando sea necesario o no. Por lo que queda demostrado que la tarjeta de adquisición de datos ha sido mejorada y está lista para funcionar.

Si se requiere más información sobre los datos y valores obtenidos, se pueden revisar todas las medidas en el apéndice 5, al final de esta memoria.

8. Resultados obtenidos

Capítulo 9

Conclusiones

9. Conclusiones

9.2 Líneas futuras

9.1 CONCLUSIONES

Como se mencionó al principio de esta memoria, para realizar este gran proyecto, se ha tenido que diseñar circuitos electrónicos, diseño hardware, y una programación software capaz de controlar lo que se ha creado.

Cabe destacar que los objetivos del proyecto se han cumplido, en mayor o menor medida, como se muestra a continuación:

- Incorporación de canales para medir intensidad de corriente.
- Incorporación de salidas para alimentar circuitos externos.
- Pasar de medir señales en continua a señales alternas debido al aumento de la frecuencia de muestreo de 20 veces mayor.
- Optimización a la hora de realizar las medidas pasando de medir 64 veces a solo 16, gracias a la sustitución del promediado por un filtrado de mediana.
- Reducir el tiempo de cálculo para procesar los datos.

A parte, como se explica en el capítulo anterior, todos los canales están funcionando y listos para realizar mediciones.

Queremos resaltar que el tiempo de muestreo que alcanza la tarjeta de adquisición de datos no es lo suficientemente alto como pensábamos que íbamos a alcanzar al principio, pero estamos muy limitados por la aplicación que gestiona toda la tarjeta, el LabVIEW.

La realización de este trabajo no ha sido una tarea nada fácil, al contrario, ha sido una tarea ardua, trabajosa y larga en el tiempo, pero deja una sensación muy gratificante poder mirar atrás y ver como se han ido solucionando los innumerables fallos, contratiempos o problemas que plantea realizar un trabajo de esta magnitud.

9.2 LÍNEAS FUTURAS

Realmente la tarjeta de adquisición está bastante completa, ya mide diferencia de tensión e intensidad de corriente, hasta tiene sus salidas para alimentar los circuitos que se quieren medir. Como se comentó en el apartado anterior, la frecuencia de muestreo puede ser mucho más alta. Cuando uno lee las características del convertidor analógico-digital que está integrado en el

Optimización de una tarjeta de adquisición de datos

microcontrolador puede comprobar que el tiempo que tarda en tomar una medida es sobre unos $3\mu\text{s}$. Traducido a frecuencia es 333,33KHz. Entonces no nos es suficiente quedarnos con la frecuencia de muestreo que nos proporciona la aplicación de LabVIEW.

El siguiente paso lógico sería realizar una aplicación, en cualquier otro entorno, que no perjudique tanto en el aspecto de la frecuencia de muestreo. Se ha pensado hacerlo en Java, por su facilidad de código, sus prestaciones y su portabilidad.

Otra mejora será la de que el usuario pueda elegir los valores de tensión salida de la SMU. Ahora el usuario solo elige que SMU va a querer medir, pero lo ideal sería que también pudiese elegir que tensiones quiere para realizar la medida. Con esta mejora se podrán medir características I-V de dispositivos de dos terminales con una única SMU, o bien de tres terminales, usando ambas SMUs.

Apéndice 1

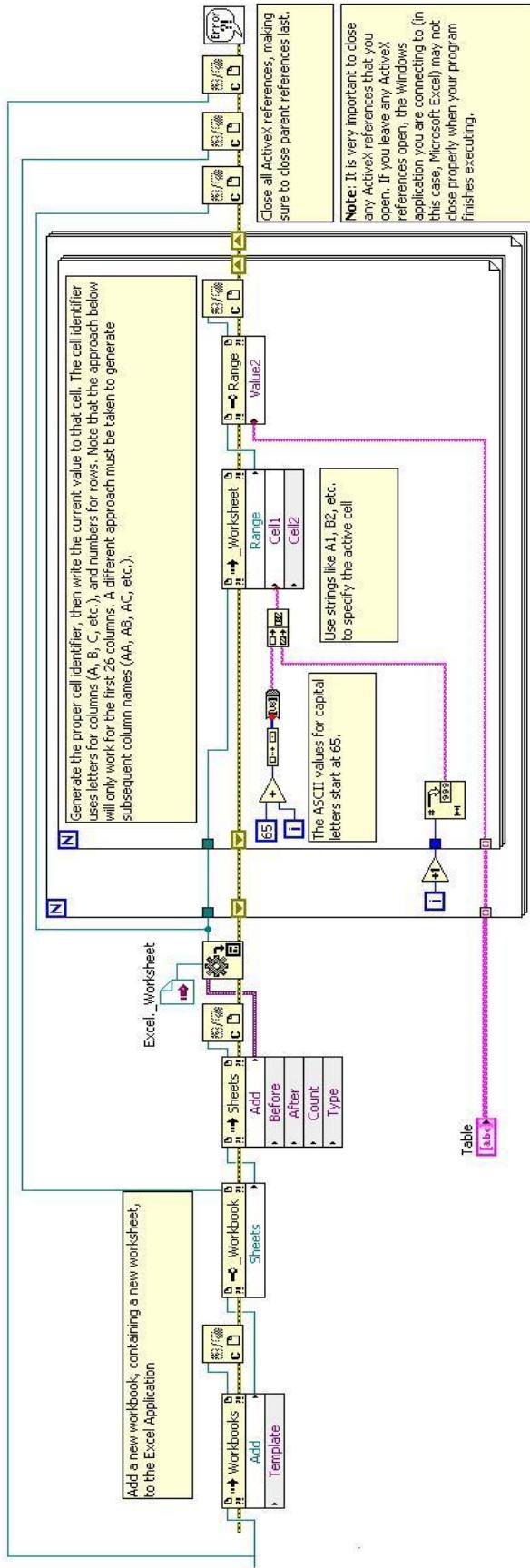
Presupuesto

Componente	Código RS	Cantidad	Precio unitario	Precio total
ADG408	522-9925	1	5,23 €	5,23 €
BJT NPN	813-6700	1	0,17 €	0,17 €
BJT PNP	739-0363P	1	0,18 €	0,18 €
Bloques terminales PCB	220-4305	3	2,65 €	7,95 €
Caja metálica	741-7506	1	16,08 €	16,08 €
Condensador 22pF	538-1461	2	0,20 €	0,40 €
Condensador 10nF	312-1431	3	0,09 €	0,28 €
Condensador 100nF	699-5027	11	0,12 €	1,32 €
Condensador 100nF	711-1396	10	0,03 €	0,30 €
Condensador 200nF	807-6979	1	0,95 €	0,95 €
Condensador 1µF	365-4795	1	0,20 €	0,20 €
Condensador 10µF	228-6717	2	0,10 €	0,20 €
Condensador 100µF	228-6767	11	0,13 €	1,43 €
Conektor USB Tipo B	800-6860	1	1,24 €	1,24 €
Cristal de cuarzo 4MHz	672-0173	1	0,43 €	0,43 €
Diodo 1N4148	652-7343	2	0,02 €	0,04 €
Diodo 1N5817	652-7321	2	0,10 €	0,20 €
Inductor 220µH	233-5415	2	1,99 €	3,98 €
LED	262-2955	4	0,28 €	1,12 €
LM19L12	536-0010	1	0,68 €	0,68 €
LM4040	761-5905	1	0,53 €	0,53 €
LM78L12	535-9981	1	0,57 €	0,57 €
LT1366	545-6509	5	8,27 €	41,35 €
MAX743	332-714	1	14,92 €	14,92 €
PIC18F4553	400-318	1	6,57 €	6,57 €
Placa fotosensible	397-0104	1	8,29 €	8,29 €
Resistencia 330Ω	740-0823	4	0,03 €	0,12 €
Resistencia 470Ω	739-8723	1	0,01 €	0,01 €
Resistencia 1KΩ	739-8758	4	0,01 €	0,04 €
Resistencia 10KΩ	131-895	11	0,09 €	0,99 €
Resistencia 15KΩ	755-0959	1	0,34 €	0,34 €
Resistencia 47KΩ	131-902	3	0,07 €	0,21 €
Resistencia 100KΩ	755-0701	2	0,25 €	0,50 €
Tira de pines	701-9777	1	0,26 €	0,26 €
Zócalo 8 pines	813-115	5	0,55 €	2,75 €
Zócalo 16 pines	745-5426	1	0,50 €	0,50 €
Zócalo 40pines	782-9990	1	1,66 €	1,66 €
Subtotal				121,99 €
Gastos de envío				0 €
IVA 21%				25,62 €
TOTAL				147,61 €

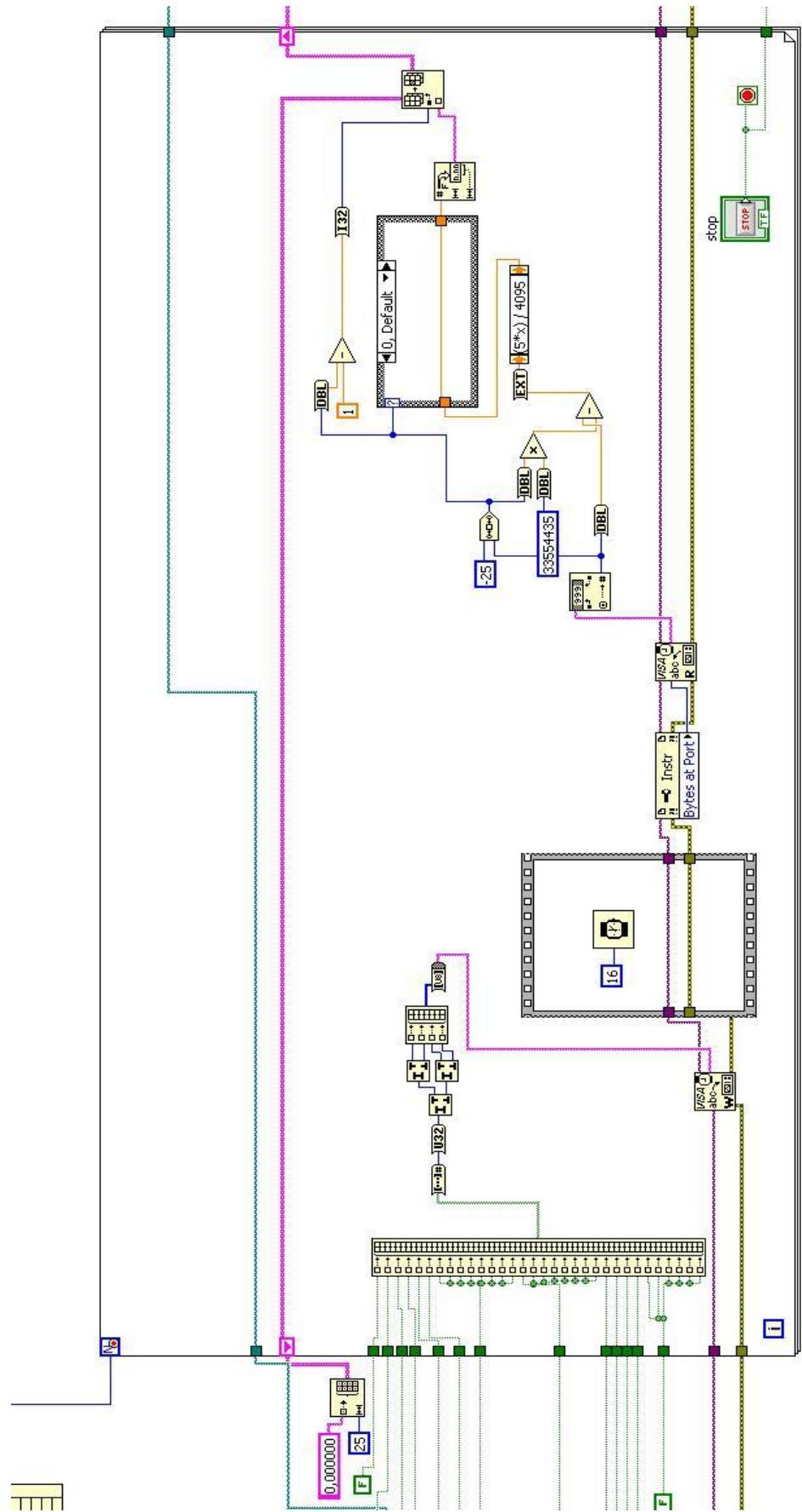
Apéndice 1.1: Presupuesto generado por RS [18]

Apéndice 2

Diagramas de bloques del LabVIEW



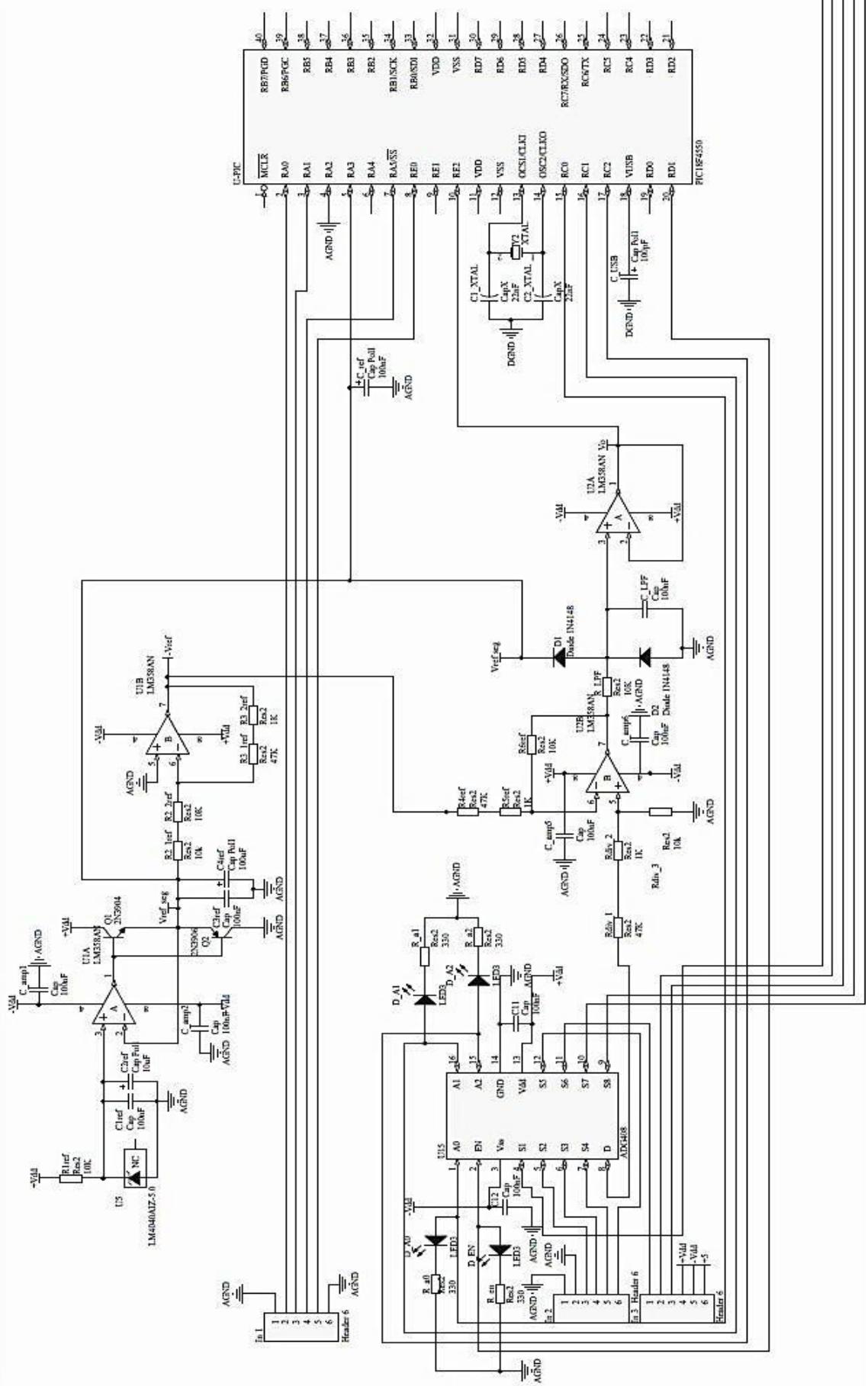
Apéndice 2.1 Diagrama de bloques encargado de guardar en Excel



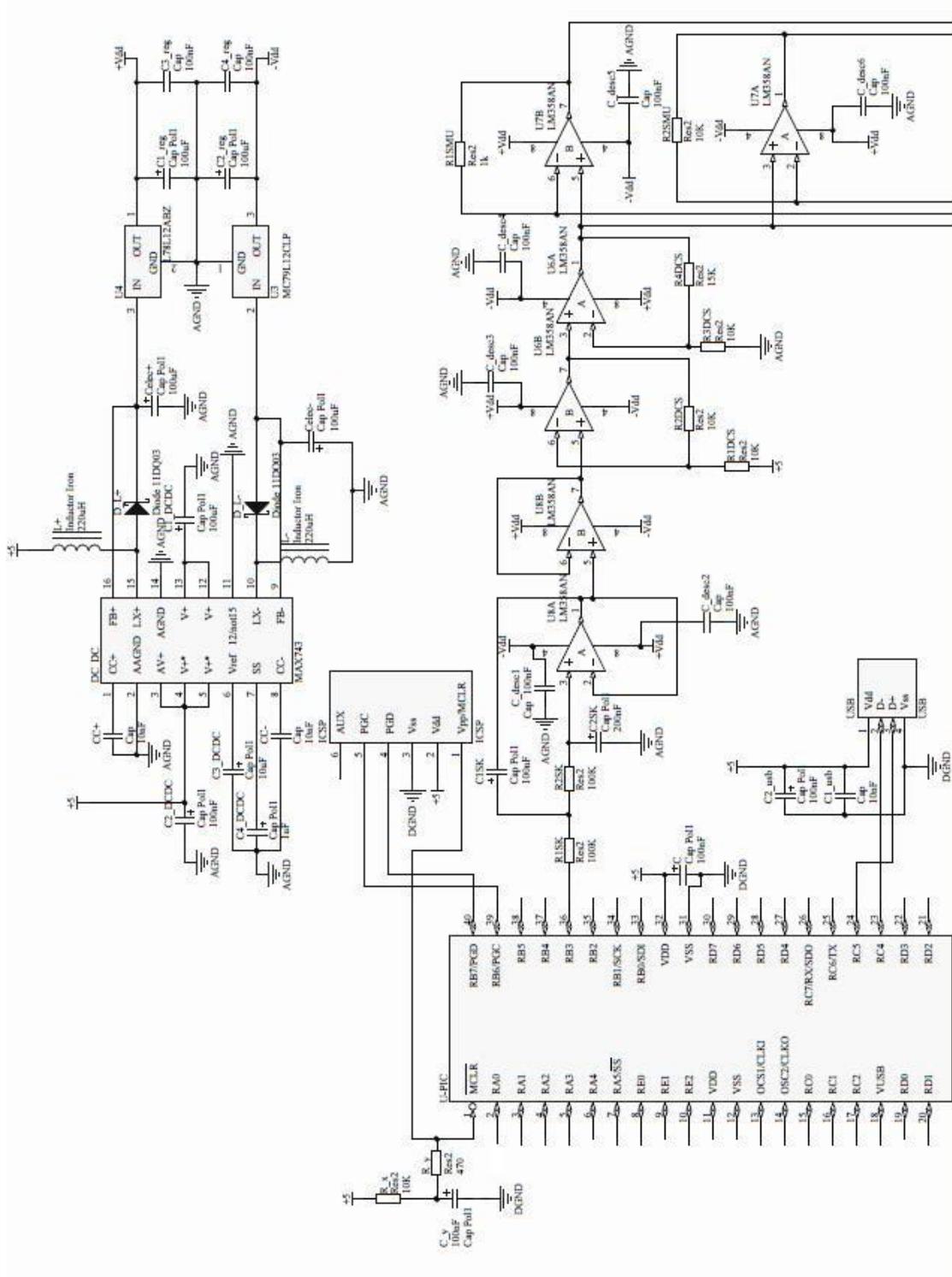
Apéndice 2.2 Diagrama de bloques de la aplicación de gestión

Apéndice 3

Esquemáticos del proyecto



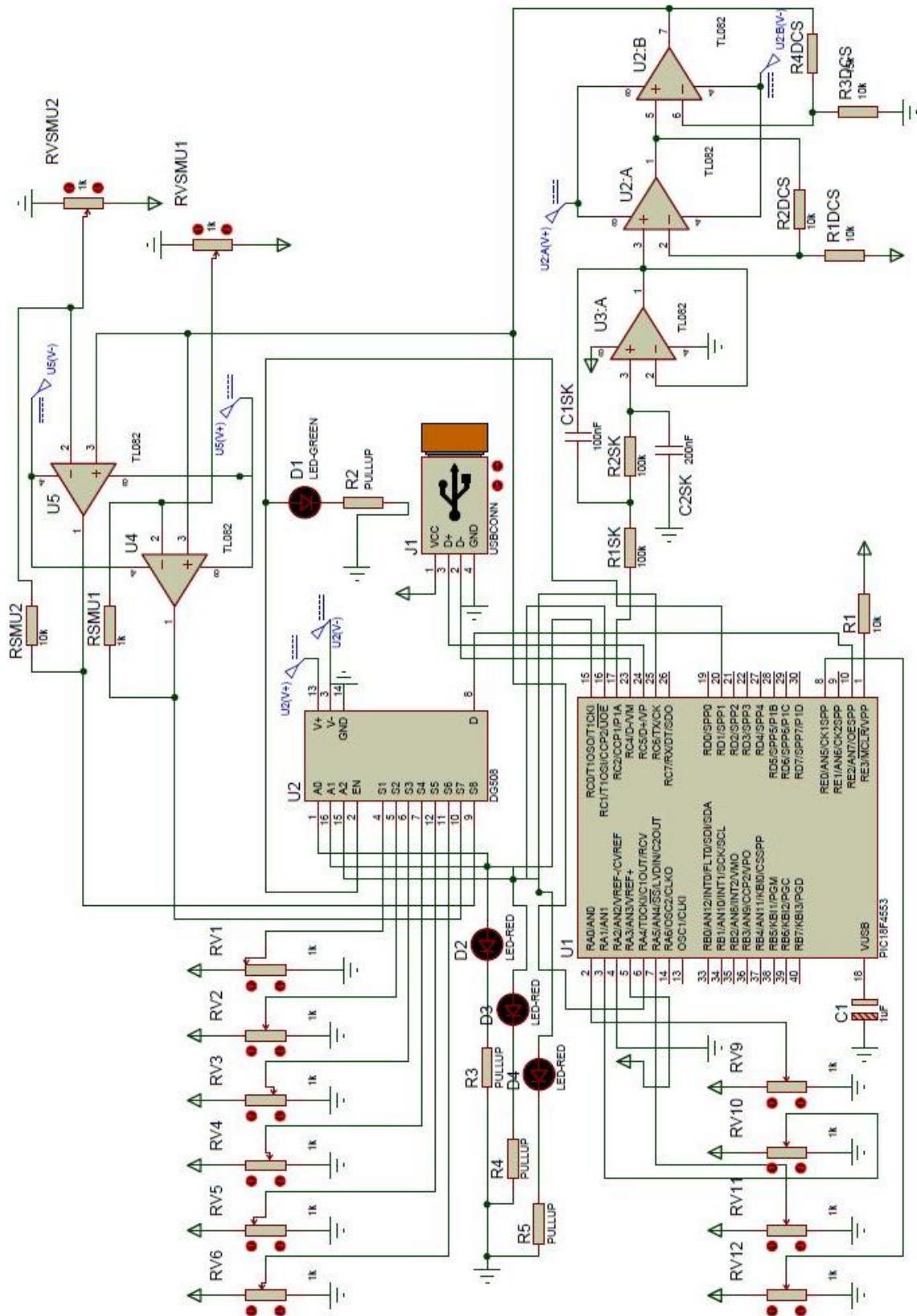
Apéndice 3.1 Esquemático del proyecto parte 1



Apéndice 3.2 Esquemático del proyecto parte 2

Apéndice 4

Esquema de Proteus



Apéndice 4.1 Esquemático de Proteus

Apéndice 5

Datos de las medidas

Datos de las medidas de luz (canal C)
y temperatura (canal D)

TIEMPO	CANAL C	CANAL D
19:09:06	2,587302	0,532357
19:10:06	3,069597	0,527473
19:11:06	3,062271	0,528694
19:12:06	3,01221	0,527473
19:13:06	3,075702	0,527473
19:14:06	3,075702	0,527473
19:15:06	3,10989	0,527473
19:16:06	3,114774	0,527473
19:17:06	3,126984	0,527473
19:18:06	3,135531	0,527473
19:19:06	3,155067	0,527473
19:20:06	3,162393	0,526252
19:21:06	3,108669	0,526252
19:22:06	3,123321	0,526252
19:23:06	3,142857	0,526252
19:24:06	3,113553	0,526252
19:25:06	3,155067	0,527473
19:26:06	3,15873	0,526252
19:27:06	3,155067	0,526252
19:28:06	3,156288	0,527473
19:29:06	3,174603	0,527473
19:30:06	3,179487	0,526252
19:31:06	3,188034	0,527473
19:32:06	3,192918	0,527473
19:33:06	3,211233	0,52381
19:34:06	3,221001	0,527473
19:35:06	3,223443	0,527473
19:36:06	3,23199	0,526252
19:37:06	3,23199	0,527473
19:38:06	3,251526	0,527473
19:39:06	3,251526	0,527473
19:40:06	3,257631	0,527473
19:41:06	3,277167	0,527473
19:42:06	3,282051	0,527473
19:43:06	3,295482	0,525031
19:44:06	3,311355	0,527473
19:45:06	3,321123	0,527473
19:46:06	3,32967	0,526252
19:47:06	3,335775	0,527473
19:48:06	3,339438	0,527473
19:49:06	3,339438	0,528694
19:50:06	3,345543	0,527473
19:51:06	3,349206	0,527473

Datos de las medidas de tensión en continua con el divisor de tensión

TIEMPO	CANAL 3	CANAL C	CANAL D
34,578	4,858242	2,255189	2,285714
34,625	4,811355	2,285714	2,280830
34,671	4,799634	2,285714	2,280830
34,734	4,793773	2,285714	2,251526
34,781	4,823077	2,255189	2,280830
34,828	4,864103	2,285714	2,280830
34,875	4,811355	2,255189	2,280830
34,937	4,793773	2,255189	2,280830
34,984	4,811355	2,283272	2,279609
35,031	4,811355	2,255189	2,251526
35,078	4,811355	2,255189	2,251526
35,125	4,869963	2,283272	2,279609
35,187	4,864103	2,285714	2,251526
35,234	4,864103	2,255189	2,282051
35,281	4,793773	2,255189	2,251526
35,328	4,811355	2,284493	2,251526
35,39	4,84652	2,284493	2,279609
35,437	4,869963	2,284493	2,251526
35,484	4,817216	2,284493	2,280830
35,531	4,858242	2,284493	2,251526
Datos de la medida de la señal alterna con forma de dientes de sierra			
MUESTRA	CANAL 1		
1	0,000000		
1	-2,587546		
2	-2,364835		
3	-2,03663		
4	-1,849084		
5	-1,550183		
6	-1,356777		
7	-1,040293		
8	-0,794139		
9	-0,58315		
10	-0,331136		
11	-0,043956		
12	0,120147		
13	0,454212		
14	0,682784		
15	0,969963		
16	1,151648		
17	1,450549		
18	1,74359		
19	1,948718		

19:52:06	3,349206	0,527473	20	2,212454
19:53:06	3,349206	0,527473	21	2,493773
19:54:06	3,349206	0,527473	22	2,704762
19:55:06	3,349206	0,527473	23	2,915751
19:56:06	3,349206	0,527473	24	3,167766
19:57:06	3,349206	0,527473	25	3,449084
19:58:06	3,384615	0,527473	26	3,712821
19:59:06	3,387057	0,527473	27	3,917949
20:00:06	3,401709	0,527473	28	4,158242
20:01:06	3,388278	0,528694	29	4,410256
20:02:06	3,399267	0,527473	30	4,668132
20:03:06	3,416361	0,527473	31	-2,880586
20:04:06	3,418803	0,527473	32	-4,474725
20:05:06	3,416361	0,527473	33	-4,375092
20:06:06	3,416361	0,527473	34	-4,123077
20:07:06	3,338217	0,527473	35	-3,912088
20:08:06	3,418803	0,527473	36	-3,771429
20:09:06	3,422466	0,527473	37	-3,466667
20:10:06	3,42735	0,527473	38	-3,161905
20:11:06	3,457875	0,527473	39	-2,933333
20:12:06	3,461538	0,527473	40	-2,734066
20:13:06	3,45177	0,527473	41	-2,417582
20:14:06	3,446886	0,527473	42	-2,18315
20:15:06	3,460317	0,527473	43	-1,983883
20:16:06	3,445665	0,527473	44	-1,690842
20:17:06	3,45177	0,526252	45	-1,520879
20:18:06	3,449328	0,527473	46	-1,186813
20:19:06	3,443223	0,528694	47	-0,999267
20:20:06	3,321123	0,527473	48	-0,770696
20:21:06	3,349206	0,527473	49	-0,483516
20:22:06	3,43956	0,526252	Datos de las medidas de la señal alterna con forma de sinusoidal	
20:23:06	3,319902	0,527473	MUESTRA	CANAL 2
20:24:06	3,423687	0,527473	1	0,000000
20:25:06	3,349206	0,527473	1	-4,058608
20:26:06	3,349206	0,527473	2	-2,558242
20:27:06	3,290598	0,527473	3	-2,8337
20:28:06	3,272283	0,527473	4	-2,153846
20:29:06	3,212454	0,527473	5	-1,520879
20:30:06	3,139194	0,527473	6	-0,770696
20:31:06	3,075702	0,527473	7	-0,014652
20:32:06	3,008547	0,525031	8	0,72967
20:33:06	2,959707	0,527473	9	1,462271
20:34:06	2,880342	0,526252	10	2,230037
20:35:06	2,783883	0,527473	11	2,874725
20:36:06	2,675214	0,527473	12	3,378755

20:38:06	2,383394	0,526252	13	3,917949
20:39:06	2,167277	0,527473	14	4,334066
20:40:06	2,081807	0,527473	15	4,480586
20:41:06	1,93895	0,527473	16	4,814652
20:42:06	1,846154	0,527473	17	4,855678
20:43:06	1,758242	0,527473	18	4,814652
20:44:06	1,714286	0,527473	19	4,480586
20:45:06	1,630037	0,528694	20	4,357509
20:46:06	1,601954	0,527473	21	3,970696
20:47:06	1,566545	0,527473	22	3,443223
20:48:06	1,365079	0,527473	23	2,798535
20:49:06	1,527473	0,527473	24	2,153846
20:50:06	1,483516	0,528694	25	1,339194
20:51:06	1,31746	0,527473	26	0,536264
20:52:06	1,435897	0,527473	27	-0,278388
20:53:06	1,411477	0,527473	28	-1,104762
20:54:06	1,383394	0,527473	29	-1,825641
20:55:06	1,367521	0,528694	30	-2,646154
20:56:06	1,346764	0,527473	31	-3,302564
20:57:06	1,31746	0,527473	32	-3,859341
20:58:06	1,31746	0,526252	33	-4,334066
20:59:06	1,294261	0,526252	34	-4,668132
21:00:06	1,255189	0,527473	35	-4,843956
21:01:06	1,221001	0,527473	36	-4,855678
21:02:06	1,179487	0,527473	37	-4,967033
21:03:06	1,107448	0,527473	38	-4,79707
21:04:06	1,102564	0,528694	39	-4,457143
21:05:06	1,072039	0,528694	40	-4,052747
21:06:06	1,043956	0,526252	41	-3,507692
21:07:06	1,004884	0,527473	42	-3,021245
21:08:06	0,986569	0,527473	43	-2,358974
21:09:06	0,967033	0,527473	44	-1,567766
21:10:06	0,941392	0,528694	45	-0,887912
21:11:06	0,916972	0,527473	46	-0,120147
21:12:06	0,887668	0,527473	47	0,665201
21:13:06	0,848596	0,527473	48	1,473993
21:14:06	0,825397	0,527473	49	2,171429
21:15:06	0,789988	0,527473	Datos de las medidas de la señal alterna con forma triangular	
21:16:06	0,757021	0,528694	MUESTRA	CANAL 3
21:17:06	0,744811	0,527473	1	0,000000
21:18:06	0,721612	0,527473	2	2,586745
21:19:06	0,683761	0,528694	1	2,048352
21:20:06	0,656899	0,528694	2	1,438828
21:21:06	0,628816	0,528694	3	0,917216
21:22:06	0,605617	0,527473	4	0,471795

21:24:06	0,550672	0,528694	5	-0,014652
21:25:06	0,520147	0,527473	6	-0,536264
21:26:06	0,498168	0,527473	7	-1,057875
21:27:06	0,467643	0,528694	8	-1,544322
21:28:06	0,438339	0,528694	9	-2,124542
21:29:06	0,417582	0,528694	10	-2,646154
21:30:06	0,379731	0,528694	11	-3,144322
21:31:06	0,3663	0,528694	12	-3,583883
21:32:06	0,34188	0,529915	13	-4,128938
21:33:06	0,321123	0,526252	14	-4,574359
21:34:06	0,498168	0,529915	15	-4,750183
21:35:06	0,282051	0,528694	16	-4,334066
21:36:06	0,433455	0,528694	17	-3,865201
21:37:06	0,306471	0,528694	18	-3,355311
21:38:06	0,14652	0,528694	19	-2,863004
21:39:06	0,119658	0,531136	20	-2,341392
21:40:06	0,185592	0,531136	21	-1,81978
21:41:06	0,177045	0,531136	22	-1,333333
21:42:06	0,067155	0,531136	23	-0,841026
21:43:06	0,057387	0,531136	24	-0,389744
21:44:06	0,042735	0,531136	25	0,049817
21:45:06	0,029304	0,531136	26	0,618315
21:46:06	0,102564	0,531136	27	1,134066
21:47:06	0,084249	0,531136	28	1,573626
21:48:06	-0,004884	0,531136	29	2,136264
21:49:06	0,019536	0,532357	30	2,634432
21:50:06	-0,015873	0,528694	31	3,068132
21:51:06	-0,025641	0,531136	32	3,63663
21:52:07	0	0,531136	33	4,111355
21:53:07	-0,010989	0,531136	34	4,480586
21:54:07	-0,050061	0,531136	35	4,627106
21:55:07	-0,050061	0,531136	36	4,298901
21:56:07	-0,029304	0,532357	37	3,730403
21:57:07	-0,058608	0,532357	38	3,284982
21:58:07	-0,059829	0,532357	39	2,751648
21:59:07	-0,050061	0,532357	40	2,230037
22:00:07	-0,069597	0,532357	41	1,808059
22:01:07	-0,069597	0,532357	42	1,198535
22:02:07	-0,074481	0,531136	43	0,665201
22:03:07	-0,059829	0,532357	44	0,167033
22:04:07	-0,074481	0,529915	45	-0,249084
22:05:07	-0,076923	0,532357	46	-0,835165
22:06:07	-0,075702	0,531136	47	-1,333333
22:07:07	-0,074481	0,531136	48	-1,778755

22:08:07	-0,078144	0,531136
22:09:07	-0,078144	0,531136
22:10:07	-0,079365	0,531136
22:11:07	-0,078144	0,532357
22:12:07	-0,075702	0,531136
22:13:07	-0,079365	0,531136
22:14:07	-0,078144	0,531136
22:15:07	-0,078144	0,531136
22:16:07	-0,076923	0,532357
22:17:07	-0,079365	0,531136
22:18:07	-0,080586	0,531136
22:19:07	-0,080586	0,532357
22:20:07	-0,080586	0,532357
22:21:07	-0,080586	0,531136
22:22:07	-0,081807	0,531136
22:23:07	-0,079365	0,531136
22:24:07	-0,078144	0,532357
22:25:07	-0,079365	0,532357
22:26:07	-0,080586	0,531136
22:27:07	-0,079365	0,532357
22:28:07	-0,079365	0,531136
22:29:07	-0,081807	0,532357
22:30:07	-0,080586	0,532357
22:31:07	-0,081807	0,531136
22:32:07	-0,079365	0,532357
22:33:07	-0,080586	0,532357
22:34:07	-0,078144	0,531136
22:35:07	-0,079365	0,531136
22:36:07	-0,080586	0,532357
22:37:07	-0,076923	0,531136
22:38:07	-0,080586	0,528694
22:39:07	-0,079365	0,532357
22:40:07	-0,079365	0,532357
22:41:07	-0,080586	0,529915
22:42:07	-0,078144	0,532357
22:43:07	-0,080586	0,532357
22:44:07	-0,079365	0,532357
22:45:07	-0,078144	0,531136
22:46:07	-0,080586	0,531136
22:47:07	-0,081807	0,531136
22:48:07	-0,079365	0,531136
22:49:07	-0,079365	0,531136
22:50:07	-0,078144	0,531136
22:51:07	-0,079365	0,531136
22:52:07	-0,078144	0,532357
22:53:07	-0,078144	0,527473

Datos obtenidos de las medidas del SMU con la resistencia, sabiendo que la R_{SMU} es de $100k\Omega$

Vo	v+	diferencia	i(μA)
-1,139927	-11,54285	10,40293	104,0293
-0,501099	-6,162637	5,661538	56,61538
0,161172	1,983883	-1,822711	-18,22711
0,917216	10,558242	-9,641026	-96,41026

22:54:07	-0,078144	0,532357
22:55:07	-0,080586	0,531136
22:56:07	-0,080586	0,532357
22:57:07	-0,080586	0,531136
22:58:07	-0,080586	0,531136
22:59:07	-0,080586	0,532357
23:00:07	-0,078144	0,532357
23:01:07	-0,078144	0,532357
23:02:07	-0,078144	0,532357
23:03:07	-0,079365	0,532357
23:04:07	-0,078144	0,532357
23:05:07	-0,078144	0,532357
23:06:07	0,532357	0,532357
23:07:07	-0,078144	0,531136
23:08:07	2,412698	0,532357

Bibliografía

[1] Historia sobre la vida y evolución de IBM.

http://www-03.ibm.com/ibm/history/exhibits/dpd50/dpd50_chronology2.html

[2] Folleto informativo de la época sobre el IBM 1800 Data Acquisition and Control System.

<http://ed-thelen.org/comp-hist/IBM-ProdAnn/1800.pdf>

[3] Sistemas de adquisición de datos, definiciones básicas, National Instruments.

<http://www.ni.com/data-acquisition/what-is/esa/>

[4] Información sobre sistemas modulares de adquisición de datos, National Instruments.

<http://www.ni.com/modularinstruments/whatis/esa/>

[5] Información general y específica del dispositivo de National Instruments USB-6000.

<http://sine.ni.com/nips/cds/view/p/lang/es/nid/211872>

[6] Características específica sobre las conexiones USB.

http://www.usb.org/developers/docs/usb20_docs/

[7] Datasheet integrado MAX743, del fabricante MAXIM

<http://www.maximintegrated.com/en/products/power/switching-regulators/MAX743.html>

[8] Datasheet integrado LM79L12, del fabricante National Semiconductor (TI)

<http://www.ti.com/product/lm79l12?keyMatch=lm79l12&tisearch=Search-EN-Everything>

[9] Datasheet integrado LM78L12, del fabricante National Semiconductor (TI)

<http://www.ti.com/product/lm78l12?keyMatch=lm78l12&tisearch=Search-EN-Everything>

[10] Datasheet I integrado LM4040, del fabricante National Semiconductor (TI)

<http://www.ti.com/product/lm4040-n?keyMatch=lm4040&tisearch=Search-EN-Everything>

[11] Página web oficial del software PSpice

<http://www.orcad.com/products/orcad-capture/overview>

[12] Datasheet integrado ADG408, del fabricante Analog Devices

<http://www.analog.com/en/products/switches-multiplexers/analog-switches-multiplexers/adg408.html#product-overview>

[13] Datasheet microcontrolador PIC18F4553, del fabricante Microchip

<http://www.microchip.com/wwwproducts/Devices.aspx?product=PIC18F4553>

[14] Software de diseño de esquemáticos y PCB, Altium Designer

<http://www.altium.com/>

[15] Página oficial del PICKit 2, de Microchip

<http://www.microchip.com/Developmenttools/ProductDetails.aspx?PartNO=DV164121>

[16] Página oficial del software LabVIEW, de National Instruments

<http://www.ni.com/labview/esa/>

[17] Datasheet integrado LM 35, del fabricante National Semiconductor (TI)

<http://www.ti.com/lit/ds/symlink/lm35.pdf>

[18] Distribuidor RS

<http://es.rs-online.com/web/>

Índice de acrónimos

- **ADC:** Analogic Digital Converter, consiste en la transcripción de señales analógicas en señales digitales, con el propósito de facilitar su procesamiento (codificación, compresión, etcétera) y hacer la señal resultante (digital) más inmune al ruido y otras interferencias.
- **CCP:** Capture, Compare, PWM, son módulos que permiten junto a otros registros, temporizar de otras maneras y formas posibles. Cada módulo puede operar en cualquiera de las tres modalidades disponibles.
- **CMOS:** Complementary metal-oxide-semiconductor, es una de las familias lógicas empleadas en la fabricación de circuitos integrados. Su principal característica consiste en la utilización conjunta de transistores de tipo pMOS y tipo nMOS.
- **CPU:** Central Processing Unit, es el hardware dentro de una computadora u otros dispositivos programables, que interpreta las instrucciones de un programa informático mediante la realización de las operaciones básicas aritméticas, lógicas y de entrada/salida del sistema.
- **DAC:** Digital to Analogue Converter, es un dispositivo para convertir señales digitales con datos binarios en señales de corriente o de tensión analógica.
- **DAQ:** Data Acquisition, consiste en la toma de muestras del mundo real (sistema analógico) para generar datos que puedan ser manipulados por un ordenador.
- **DACS:** Data Acquisition and Control System, uno de los primeros DAQ de la historia.
- **DC-DC:** dispositivo que transforma corriente continua de una tensión a otra, ya sea aumentándola o disminuyéndola.
- **E/S:** en inglés Input/Output (I/O), es un dispositivo que permite la comunicación entre un sistema de procesamiento de información.
- **EEPROM:** son las siglas de Electrically Erasable Programmable Read-Only Memory, es un tipo de memoria ROM que puede ser programada, borrada y reprogramada eléctricamente.
- **FPGA:** Field Programmable Gate Array, es un dispositivo semiconductor que contiene bloques de lógica cuya interconexión y funcionalidad puede ser configurada 'in situ' mediante un lenguaje de descripción especializado.
- **I/V:** curva característica intensidad en función de la tensión.
- **LSB:** least significant bit, es la posición de bit en un número binario que tiene el menor valor (el situado más a la derecha).
- **PC:** Personal Computer, tipo de microcomputadora diseñada en principio para ser utilizada por una sola persona a la vez.
- **PCB:** Printed Board Circuit, es la superficie constituida por caminos, pistas o buses de material conductor laminadas sobre una base no conductora.

- **PIC:** son una familia de microcontroladores tipo RISC fabricados por Microchip Technology Inc. y derivados del PIC1650, originalmente desarrollado por la división de microelectrónica de General Instrument.
- **Puerto COM:** es una interfaz de comunicaciones de datos digitales, frecuentemente utilizado por computadoras y periféricos, donde la información es transmitida bit a bit enviando un solo bit a la vez, en contraste con el puerto paralelo que envía varios bits simultáneamente.
- **Puerto USB:** Universal Serial Bus, es un bus estándar industrial que define los cables, conectores y protocolos usados en un bus para conectar, comunicar y proveer de alimentación eléctrica entre computadoras, periféricos y dispositivos electrónicos.
- **PWM:** Pulse-Width Modulation, técnica en la que se modifica el ciclo de trabajo de una señal periódica ya sea para transmitir información a través de un canal de comunicaciones o para controlar la cantidad de energía que se envía a una carga.
- **RTD:** Resistance Temperature detector, es un detector de temperatura resistivo, es decir, un sensor de temperatura basado en la variación de la resistencia de un conductor con la temperatura.
- **SMU:** Source Measure Unit, dispositivo que se usará para medir tensiones y transfórmalas en intensidades de corrientes mediante cálculos matemáticos.