Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza

Universidad
Zaragoza

Escudo registrador de datos analógicos S-LOGGER para Arduino.

Memoria

AUTOR: CARLOS LACASA AGUAS

DIRECTORES: DAVID ASIAÍN ANSORENA

DR. ÓSCAR LUCÍA GIL

CONVOCATORIA: MARZO 2013

TITULACIÓN: INGENIERÍA TÉCNICA INDUSTRIAL

ESPECIALIDAD INGENIERÍA ELECTRÓNICA

AGRADECIMIENTOS.

Quiero expresar mi gratitud a todas las personas que han hecho posible la realización de este proyecto.

En primer lugar, a mis padres, mi hermana y a mi tía Begoña, por la educación que me han ofrecido y por su confianza y apoyo incondicional.

A mis directores de proyecto, David Asiaín, que me ha ayudado y siempre ha estado disponible para cualquiera de las dudas que he tenido, y al Dr. Óscar Lucía, por su interés, ayuda y consejos. Gracias a los dos por vuestra ayuda.

Finalmente, me gustaría agradecer a Alba, por su cariño y ayuda, y a mis amigos Alejandro, Lucía, Pinós, Pilar y Ros, por estar ahí durante todos estos años.

Muchas gracias a todos.

ÍNDICE.

RESUMEN.

El objetivo del presente proyecto es el diseño, montaje y puesta a punto de un prototipo de una placa de adquisición de datos compatible con las tarjetas de Arduino.

Esta placa de adquisición de datos dispone de un reloj en tiempo real, un conversor analógico/ digital de 18 bits y con una frecuencia de muestreo de hasta 120 muestras por segundo y un módulo con una tarjeta micro SD capaz de guardar los valores tomados por el conversor.

En este proyecto se han elaborado programas para la adquisición de datos de forma autónoma con la capacidad de exportar estos datos a una tabla y realizar gráficas. A su vez, es compatible con programas de adquisición de datos como LabView. Este instrumento destaca por tener la capacidad de realizar las mismas funciones que otras placas de adquisición de datos pero a un coste menor.

1. Introducción.

1.1 Historia de Arduino.

La plataforma Arduino fue creada en el año 2005 por un equipo de desarrollo compuesto por Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, David Mellis y Nicholas Zambetti, y como describe su sitio web [6] es:

"una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar. Se creó para artistas, diseñadores, aficionados y cualquiera interesado en crear entornos u objetos interactivos. "

Su principal motivación fue la de diseñar una herramienta docente moderna y potente y que su precio fuera accesible para los estudiantes: una placa de hardware costaban 70 euros.

Esta plataforma fue creada desde un principio como un concepto *Open Source* debido a que era un proyecto integrado en la universidad de Ivrea, la cual estaba en proceso de clausura. De este modo, llegado el momento de que la universidad cerrase sus puertas, la propiedad intelectual de Arduino seria libre y podría seguir desarrollándose.

Maximo Banzi y otros de sus compañeros en la universidad desarrollaron un lenguaje de programación orientado a objetos sencillo llamado *Processing.* El *Processing* rápidamente gano popularidad ya que permitía a programadores sin experiencia crear visualizaciones complejas. La idea de Banzi era crear una herramienta similar que tuviera la capacidad de programar un microcontrolador en vez de gráficos en una pantalla.

Fue entonces cuando el estudiante Hernando Barragán desarrollo el prototipo de una plataforma de programación llamada *Wiring.* La plataforma de *Wiring* desarrolla sus programas en lenguaje C y C++, pero el usuario final únicamente necesita configurar unas pocas funciones y de una forma muy sencilla.

Con el lenguaje de programación definido solo quedaba diseñar el *hardware* necesario para trabajar. La imagen 1 muestra el primer prototipo, que fue creado en 2005, con un diseño muy sencillo.

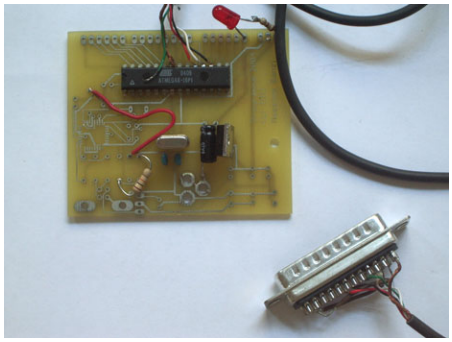


Figura 1: Primer prototipo de Arduino.

El primer prototipo fue únicamente usado por los desarrolladores para probar las compatibilidades y crear las funcionalidades que posteriormente serian aplicadas al diseño final.

Cuando este prototipo funcionaba correctamente, se pasó a la producción en serie. Para la producción en serie se tomaron varias consideraciones como: el precio debía ser menor de 30€, debían ser *Plug and Play* o lo que es lo mismo, enchufar y funcionar, y, finalmente, debían ser compatibles con todas las plataformas (Mac, Windows y Linux).

La primera producción fue únicamente de 300 placas y se distribuyeron entre los alumnos de diversas universidades como la de Ivrea o el K3 de Malmo.

Esta primera versión de Arduino tenía una comunicación mediante RS-232 y montaba un microntrolador ATmega8 de 8 bits. Posteriormente debido a la aceptación de este diseño, se desarrolló una nueva versión con comunicación USB y se mejoró su microcontrolador a un ATmega168. Estas dos placas se muestran en la figura 2.

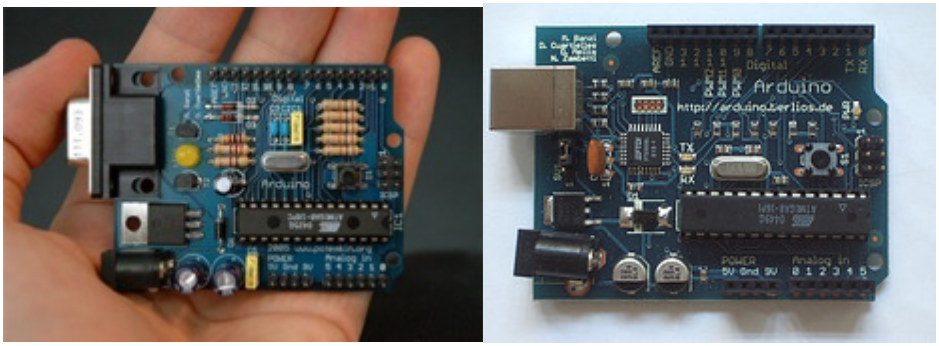


Figura 2: Arduino Serial (Rs-232) y Arduino USB.

Actualmente existe una variedad de placas de Arduino. Por ejemplo la Arduino UNO Rev3 usada en este proyecto monta un microcontrolador ATmega328 y existen otras mucho más potentes como la Mega con un ATmega2560.

Las placas de Arduino tienen la posibilidad de conectar placas de periféricos (llamadas *Shields* o escudos) que amplían una nueva función, o más, y que pueden ser controlados por la placa principal.

Existen infinidad de escudos como pueden ser: Ethernet, como el de la figura 3, Xbee, que permite conectar inalámbricamente varios Arduino, o GSM entre otros.

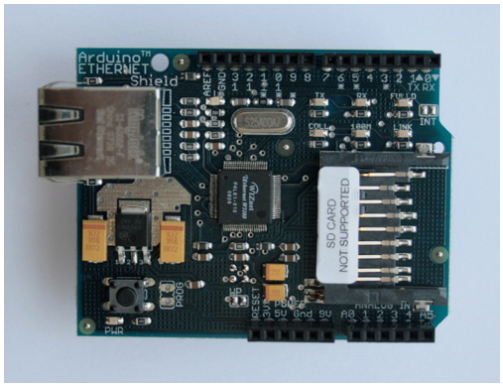


Figura 3: Escudo Ethernet.

Las posibilidades de Arduino son infinitas, se puede crear desde un programa básico como puede ser el parpadeo de un LED incorporado en la propia placa hasta el manejo de una impresora 3-D pasando por la comunicación de diversas estaciones con Xbee; el único límite lo pone el conocimiento del usuario.

1.2 Motivación.

El proyecto nace de la necesidad de implementar un hardware adecuado en la Escuela Universitaria Politécnica de La Almunia (EUPLA) para la asignatura de Instrumentación.

El hardware más usado en esta asignatura es una tarjeta de adquisición de datos, como por ejemplo la de *National instruments*, cuyo valor oscila entre los 150 y 200 euros.

Se pretende la posibilidad de diseñar un *hardware* propio utilizando las funcionalidades ya existentes de Arduino e incluyendo las que se consideran adecuadas en el escudo diseñado, con un coste contenido.

El motivo de utilizar Arduino es su bajo coste, su versatilidad y su facilidad para la comprensión y creación de nuevos proyectos por parte de los estudiantes. Por último las tarjetas Arduino son dispositivos programables, por lo que su funcionalidad aumenta al poder realizar pruebas con código, ya sea en el formato de Arduino o en lenguaje C mediante un trazador de Atmel.

1.3 Objetivo.

Se pretende diseñar y construir una placa de adquisición de datos utilizando como base la tarjeta UNO de Arduino, para la utilización en la asignatura de instrumentación en la universidad de la Almunia.

La placa de adquisición de datos debe tener características similares a otras como la de *National Instruments* también utilizada en la universidad de Zaragoza, pero con un precio menor.

Debido a que esta placa será utilizada por estudiantes, deberá incluir protecciones en las entradas para evitar que por una mala manipulación pueda estropearse el *Hardware* tanto la placa de Arduino, la *S-LOGGER* o el ordenador.

A continuación se detallan las principales características de la placa *S-LOGGER*:

- Reloj en tiempo real (RTC) autónomo.
- Modulo con tarjeta SD para almacenar los datos tomados por el conversor.
- Conversor AD con una resolución de 18 bits y 3,75 muestras por segundo
- Sistema de protección y acomodación configurable mediante componentes a la entrada de cada uno de los canales del conversor AD
- Comunicación I2C y SPI entre la tarjeta Arduino UNO y la placa *S-LOGGER.*
- Implementación de código para trabajar con *LabView*

Por último, se requiere un diseño asegurando la compatibilidad electromagnética (EMC) para que la placa pueda funcionar correctamente en los laboratorios sin ser afectada por las perturbaciones electromagnéticas que pudiese haber en los mismos.

1.4 Alcance.

Para lograr el objetivo de este proyecto se van a enumerar las distintas funciones que se deben realizar.

- Se hará un estudio del hardware de Arduino
- Se buscaran los componentes electrónicos compatibles que cumplan nuestras necesidades.
- Se diseñará y calculará el circuito electrónico que realice las funciones del objeto de este proyecto.
- Se realizarán los planos en Altium Designer.
- Se fabricará y montara el diseño de la PCB.
- Se realizará la programación del escudo mediante el software de Arduino.
- Se realizarán pruebas de tomas de datos usando el software de Arduino y Labview.

A continuación, mediante el diagrama de Gantt de la figura 4, se especifican las actividades y su distribución temporal.

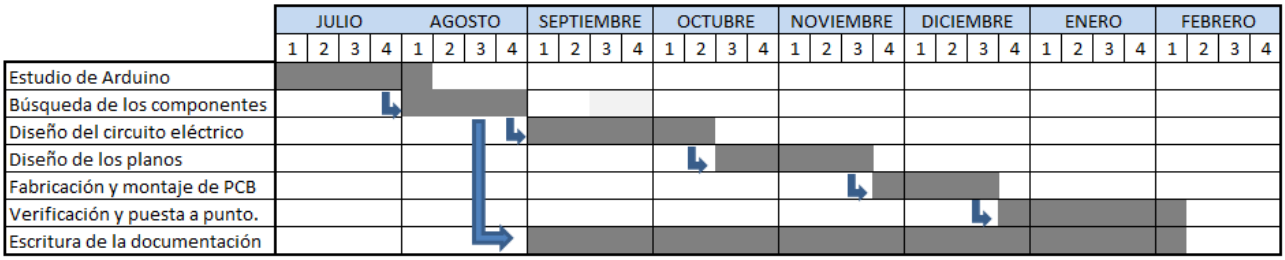


Figura 4: Diagrama de Gantt.

2. Diseño del periférico.

El objetivo de este proyecto es el diseño de una placa de adquisición de datos capaz de almacenar las muestras tomadas y tener la capacidad de integración con *LabView.* Para este objetivo se ha estudiado la placa de adquisición de datos mostrada en la figura 5, la USB-6008 de *National Instruments* [10], con la intención de igualar sus características.

Figura 5 Tarjeta USB-6008.

A continuación se muestra una tabla con las principales características de la tarjeta USB-6008.

	USB-6008
Entradas Analógicas	8 simples/4 Diferenciales
Resolución de entrada	12 Bits
Tasa de muestreo	10 KS/s
Salidas Analógicas	2
Resolución de salida	12 Bits
E/S Digitales	12
Contador de 32 Bits	1

El principal uso de la tarjeta USB-6008 es la utilización como Hardware en el programa *Labview* también de *National*.

Como se puede observar en el diagrama de bloques de la figura 6, la tarjeta se conecta al ordenador mediante USB y permite leer en el programa los valores de los sensores conectados a la tarjeta y actuar en dispositivos como servomotores, *leds* o zumbadores.

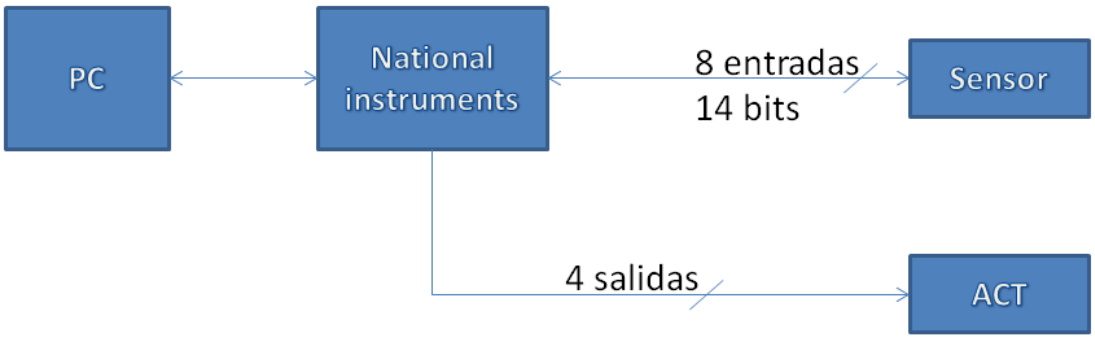


Figura 6: Diagrama de bloques tarjeta USB-6008.

2.1 Introducción.

Una vez estudiada la tarjeta de adquisición de datos USB-6008, se ha procedido al diseño de una expansión o escudo compatible con la tarjeta UNO de Arduino.

Un escudo, que proviene de la palabra inglesa *Shield*, es una placa que se superpone encima de la tarjeta de Arduino y se conecta mediante los pines laterales. En la figura 7 tenemos un ejemplo de una placa de Arduino con dos escudos conectados.

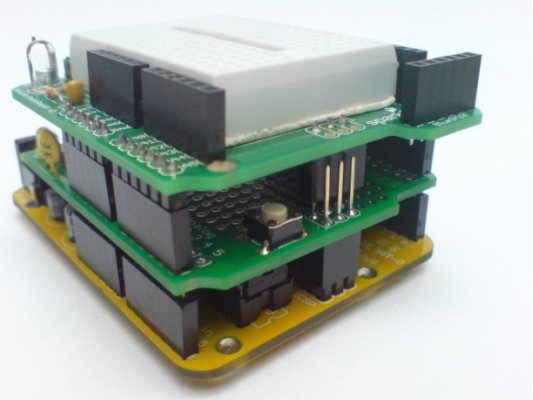


Figura 7: Placa Arduino UNO con dos escudos conectados.

El nombre con el que se va a denominar este escudo es *S-LOGGER*.

El escudo S-LOGGER tiene la función de sustituir a la placa de adquisición de datos de *National Instruments* USB-6008, por lo que debe tener unas características lo más similar posible pero con la posibilidad de aumentar alguna de sus funcionalidades.

Estas mejoras de sus funcionalidades incluyen principalmente: la mayor resolución del conversor A/D, disponibilidad de un soporte de almacenamiento físico y la posibilidad de tomar datos sin la necesidad de un ordenador.

Esta expansión dispone de un conversor de 18 bits y una capacidad de lectura de 3.75 muestras por segundo, esta velocidad de muestreo es muy baja, por lo que el tipo de sensores que se utilicen deberán ser lentos: Sensores de temperatura, galgas extensiometricas. Pero a su vez este conversor ofrece una gran resolución y un bajo ruido en las muestras tomadas.

La tarjeta de memoria añade capacidad de almacenamiento, ya que el microcontrolador de Arduino solo dispone de 30 Kb de memoria *Flash* interna disponible, y es muy difícil trabajar con ella.

Por último el reloj el tiempo real permite que el escudo pueda trabajar sin estar conectado a un ordenador y siempre teniendo la referencia horaria correcta.

Por otro lado, Arduino no es compatible directamente con LabView, pero mediante unas librerías especificas instaladas en el ordenador, y un código cargado en el microcontrolador, Arduino puede utilizar todas las funcionalidades de LabView como si fuera una tarjeta de *National Instruments.*

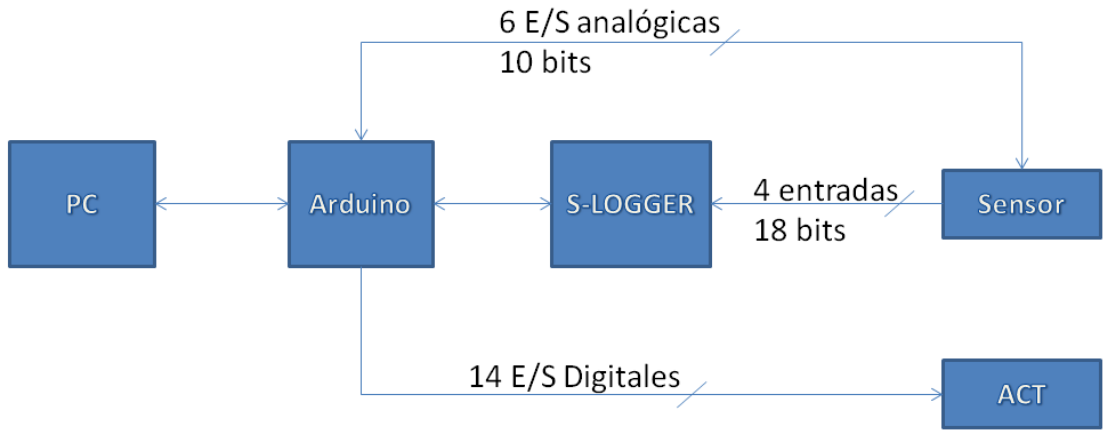


Figura 8: Diagrama de bloques Arduino.

Como se observa en la figura 8, Arduino incluye 5 entradas /salidas analógicas y 14 digitales, pero mediante el escudo *S-LOGGER* se amplía la resolución del conversor analógico/digital y se incluyen funciones como la tarjeta SD para recoger los datos de los sensores y un RTC para el funcionamiento autónomo del escudo.

2.2 Diagrama de bloques del periférico propuesto.

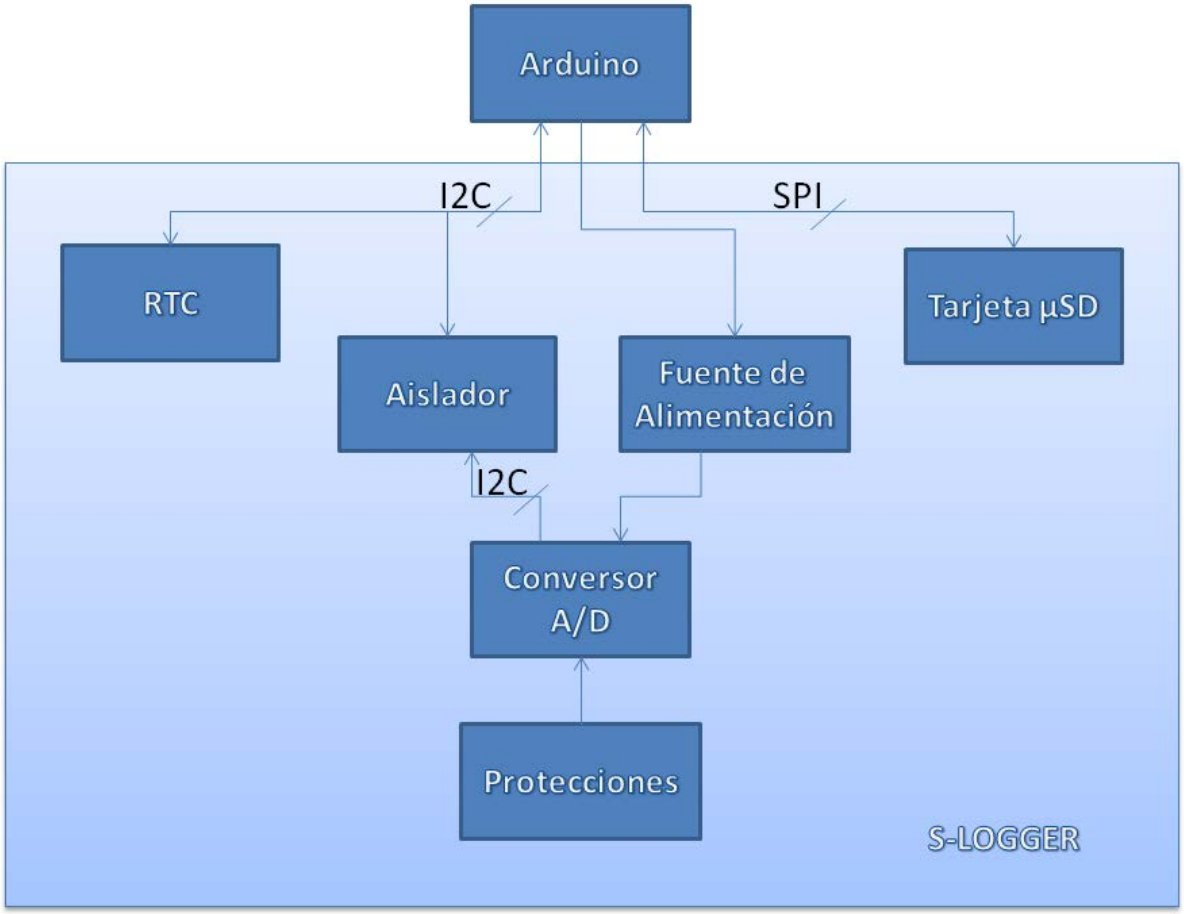


Figura 9: Diagrama de bloques.

En el diagrama de bloques de la figura 9 se pueden ver los distintos bloques de nuestro escudo y sus comunicaciones con la tarjeta de Arduino.

Como se detallara en los siguientes capítulos el *S-LOGGER* se compone, principalmente, de 4 bloques:

El RTC se trata de un reloj en tiempo real, que mediante una batería externa entrega la hora como si de un reloj digital se tratase.

La tarjeta SD permite almacenar gran cantidad de datos, ya que las placas de Arduino solo disponen de la memoria *Flash* de su micro controlador, y extraerla para su posterior lectura en un ordenador.

Para realizar un aislamiento en el escudo se precisa de dos componentes que se agrupan en el bloque de aislamiento y referencia. Mediante un aislador, que realiza un puente de comunicación, y un convertidor DC/DC, que genera una nueva referencia, es posible conectar los componentes del último bloque.

El cuarto bloque engloba la principal función de este escudo, se compone de un conversor programable de entre 12 y 18 bits, con una frecuencia de muestreo de 3.75 a 240 muestras por segundo. Este conversor no tiene la capacidad de trabajar con sensores rápidos, como acelerómetros, pero es ideal para sensores de temperatura.

Estos bloques utilizan dos tipos diferentes de comunicación para enlazarse con la placa de Arduino:

El bloque de la tarjeta micro SD se comunica mediante el protocolo SPI. El protocolo SPI es un estándar de comunicación usado para la transferencia de información entre circuitos integrados. Se trata de un bus serie de datos para la transferencia síncrona y bidireccional de información. Para la comunicación SPI son necesarias cuatro líneas: SCLK, MOSI, MISO y CS.

El reloj en tiempo real (RTC) y el conversor A/D, con un puente en la comunicación realizado por el aislante, utilizan el protocolo I2C. Este protocolo usa 2 líneas para transmitir la información: una para los datos y en la otra la señal de reloj. También en necesaria una tercera línea, pero esta es sólo la referencia (masa).

Cada uno de los dispositivos conectados al bus I2C tiene una dirección única para cada uno. En este caso la dirección del reloj será la 0x68 y la del conversor A/D la 0x6A.

2.3 Esquema general del circuito.

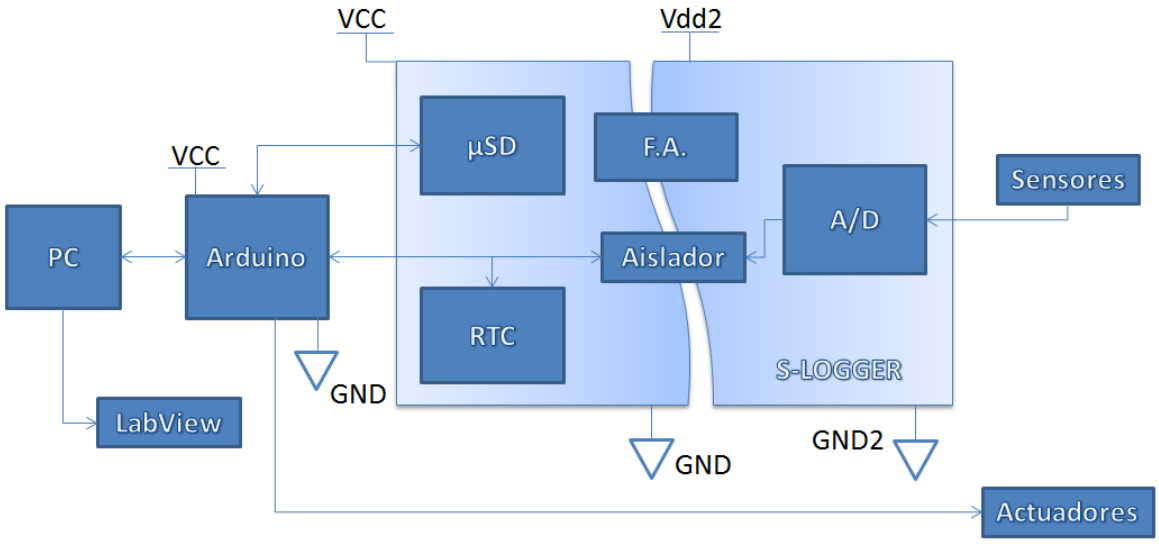


Figura 10: Esquema general de la *S-LOGGER*.

La figura 10 muestra esquema general del circuito se puede apreciar que la placa *S-LOGGER* está separada en dos conjuntos, parte digital y parte analógica.

En la parte digital se encuentran los bloques del reloj de tiempo real y el bloque del módulo de la tarjeta SD.

En la parte analógica se encuentra el bloque del conversor A/D que incluye sus protecciones en las entradas.

Para la unión de la parte digital y analógica hay un 4 bloque, denominado sistema de aislamiento y referencia, que engloba la fuente de alimentación y el aislante.

En los siguientes apartados se va a proceder a explicar detalladamente cada uno de los bloques. Para cada uno se va a describir su arquitectura y se justificara el uso de los componentes utilizados y se detallaran todos los cálculos realizados.

Está descripción será la documentación que se deberá utilizar en el caso de que se siga la línea de desarrollo de este escudo.

2.4 Bloque tarjeta SD.

2.4.1Descripción.

El escudo incorpora un bloque con una tarjeta SD, con la finalidad de guardar los datos adquiridos sin la necesidad de estar conectada la placa de Arduino a un ordenador.

La tarjeta SD necesita de cambio de los niveles de tensión para comunicarse con Arduino. Este cambio se puede realizar mediante elementos pasivos (Figura 11) [8] o mediante un componente digital (figura 12) [9].

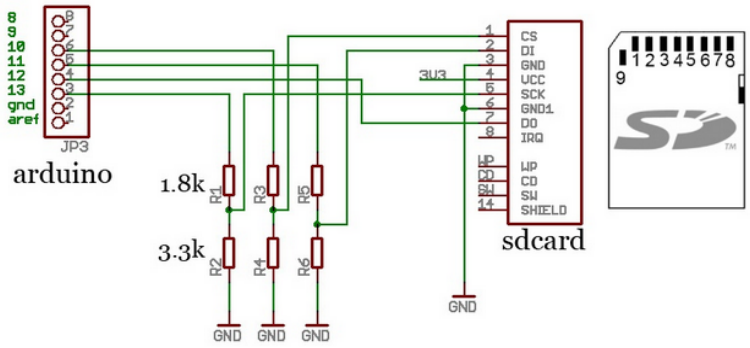


Figura 11: Esquema conexionado SD analógico.

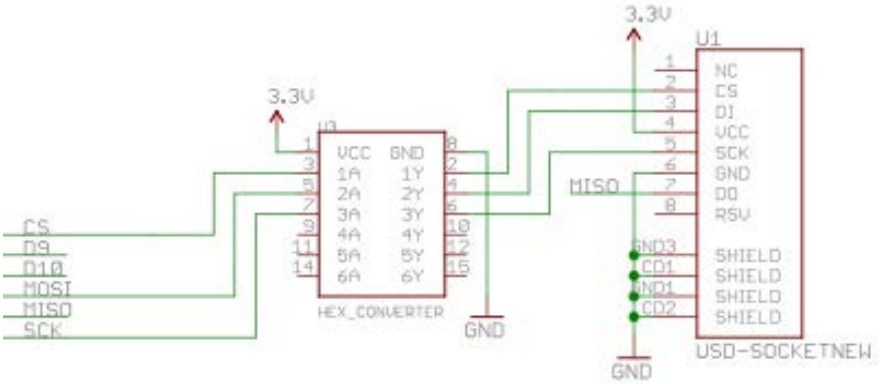


Figura 12: Esquema conexionado SD digital.

El sistema de la figura 11, al montar únicamente resistencias, tiene un coste menor, sin embargo, su principal inconveniente es que las tolerancias de las resistencias podrían llegar a producir problemas en la comunicación.

Las nuevas tarjetas, con grandes capacidades, precisan de ondas con muy poca pendiente, transiciones que se consiguen con cambiadores de nivel como el utilizado en la figura 12, un 4050.

Como se ha explicado anteriormente la comunicación de la tarjeta SD con Arduino se hace mediante SPI. Para esta comunicación son necesarias cuatro conexiones y mediante la tabla de la figura 13, proporcionada por Arduino [5], seleccionamos los correspondientes a nuestra tarjeta.

Arduino Board	MOSI	MISO	SCK	SS (slave)	SS (master)
Uno or Duemilanove	11 or ICSP-4	12 or ICSP-1	13 or ICSP-3	10	-
Mega1280 or Mega2560	51 or ICSP-4	50 or ICSP-1	52 or ICSP-3	53	-
Leonardo	ICSP-4	ICSP-1	ICSP-3	-	-
Due	ICSP-4	ICSP-1	ICSP-3	-	4, 10, 52

Figura 13: Tabla de conexiones de SPI.

Como se puede observar en la tabla, los pines utilizados son el 11, 12 y 13. Para el SS o *chip select* se puede utilizar cualquier pin, ya que se configura en el programa.

2.4.2Selección de componentes.

2.4.2.1 74HC4050.

Descripción.

El 4050, como el de figura 14, es un conversor de nivel alto a nivel bajo. Por lo que tiene la función de adaptar la tensión de Arduino (5 V) con la de trabajo de la tarjeta SD (3,3 V)

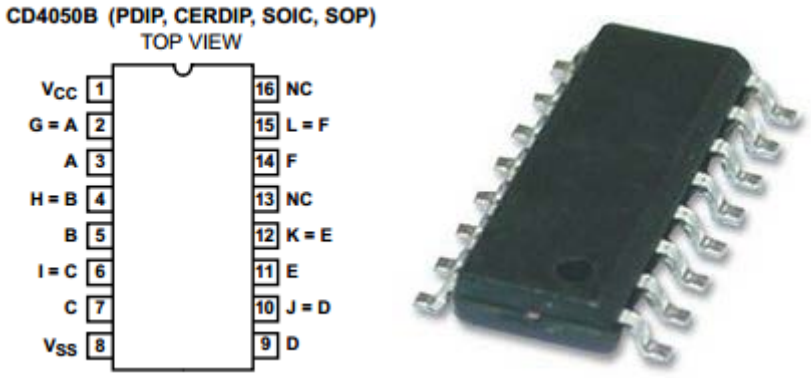


Figura 14: Patillaje y encapsulado del 74HC4050.

Este componente se compone de seis memorias no inversoras con una estructura de protección en su entrada.

El protocolo SPI solo precisa adaptación en la comunicación en tres de sus conexiones: MOSI, SCK y CS por lo que solo se usaran tres memorias del 4050.

La línea MISO se puede conectar directamente desde la tarjeta SD a Arduino.

2.4.2.2 Tarjeta micro SD.

Descripción.

En el módulo de la tarjeta de almacenamiento hay varias opciones tanto de formato como de tamaño. El formato que se va a usar es SD que tiene diferentes tamaños que son:

- SD estándar
- Mini SD
- Micro SD.

La mini SD se descartó por su escasa utilización y comercialización, la SD debido a su gran tamaño y al poco espacio disponible en el escudo.

Por lo que la opción elegida ha sido una micro SD como la que se muestra en la figura 15.

Figura 15: Patillaje y encapsulado de la tarjeta micro SD.

En el momento de seleccionar una tarjeta micro SD hay que tener en cuenta varios factores como pueden ser: Su capacidad y su capacidad de lectura/ escritura.

La capacidad elegida vendrá por el tiempo que el escudo va a estar tomando datos, pero al tratarse de datos simples la necesidad de memoria es baja.

Por otro lado tenemos la velocidad de lectura/escritura, nuestro escudo al incorporar un conversor lento, y los datos a transmitir son de un tamaño pequeño, no es necesaria una tarjeta muy rápida.

Diagrama de bloques.

A continuación se muestra, en la figura 16, el diagrama de bloques de una tarjeta SD.

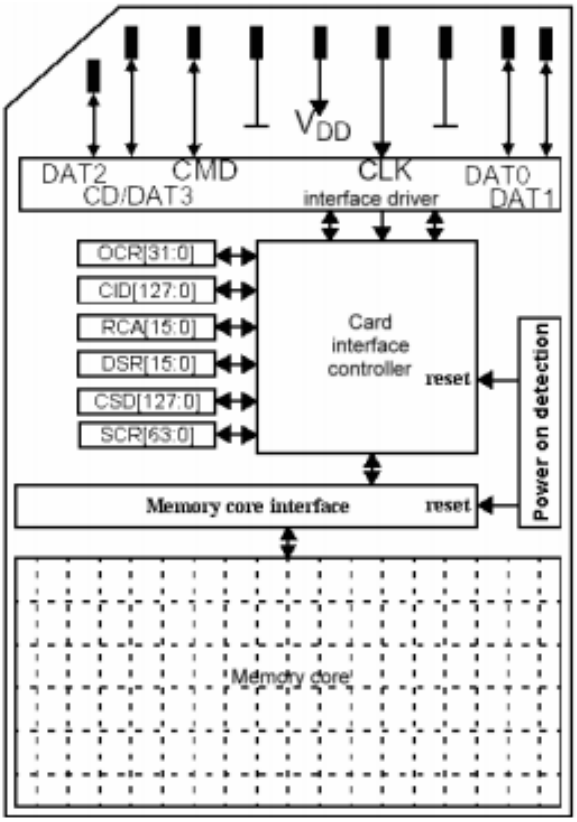


Figura 16: Diagrama de bloques de la tarjeta micro SD.

2.4.2.3 Zócalo para micro SD.

Descripción.

La tarjeta micro SD precisa de un soporte en la placa para su conexionado. El zócalo utilizado debe ser compatible con el tamaño de la tarjeta que usamos , micro SD, y debe tener unas medidas apropiadas para que no ocupe demasiado espacio en la *PCB* de la *S-LOGGER*.

Figura 17: Zócalo micro SD.

El zócalo que utilizamos, como el de la figura 17, tiene un mecanismo muy simple, únicamente hay que introducir la tarjeta por presión hasta que haga tope.

Para sacarla hay que asegurarse que la *S-LOGGER* no esté trabajando y solo hay que tirar de ella.

Este componente debido a su reducido tamaño y pequeña superficie de soldadura debe ser tratado con especial cuidado, ya que si forzáramos la conexión de la tarjeta el zócalo podría soltarse rompiendo el escudo.

2.4.2.4 Esquemático.

La figura 18 muestra el montaje utilizado para el bloque SD, con sus principales componentes.

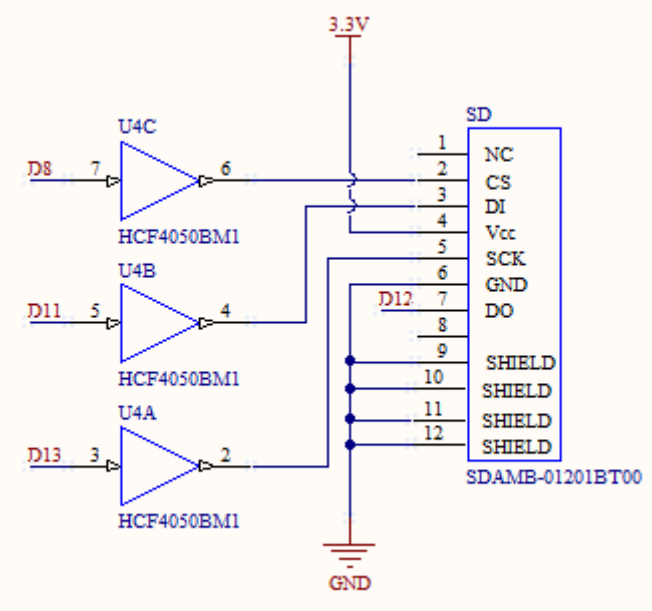


Figura 18: Esquemático micro SD.

2.5 Bloque reloj en tiempo real (RTC).

2.5.1Descripción.

La tarjeta *S-LOGGER* incluye un bloque de reloj de tiempo real o RTC, cuya finalidad es la de almacenar la información de hora y fecha reales y proporcionarla cuando el escudo este trabajando autónomamente.

Este bloque tiene la principal utilidad de permitir guardar los datos de los sensores con su correspondiente fecha y hora de cuando fueron tomados. La comunicación con Arduino será mediante I2C.

Para implementar este bloque, es necesario un integrado que realice esta función, el Ds1307, y en su hoja de características indica el montaje apropiado para su correcto funcionamiento (Figura 19).

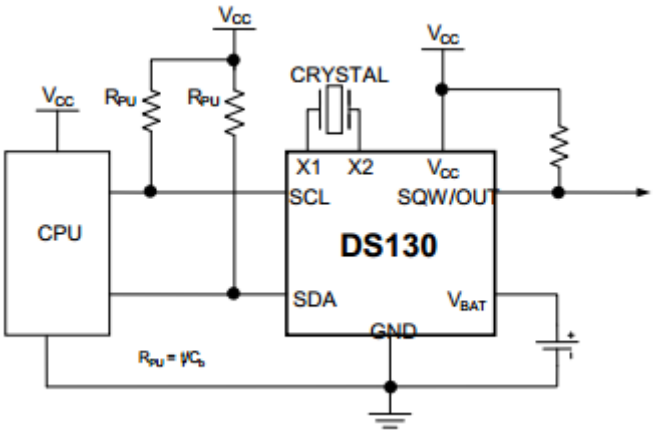


Figura 19: Esquema propuesto del DS1307.

Como se puede observar en la Figura 18, para el montaje es necesario un cristal de cuarzo y una batería, además de las resistencias de *pull-up*.

2.5.2Selección de componentes.

2.5.2.1 DS1307.

Descripción.

Para la realización de un RTC, en Arduino el componente más utilizado es el DS1307, figura 20. Este componente cumple con nuestras necesidades ya que su precio es bajo, permite la comunicación I2C, utilizada en nuestro hardware, y tener un uso tan extendido en la plataforma Arduino existe una gran cantidad de código realizado con diferentes posibilidades.

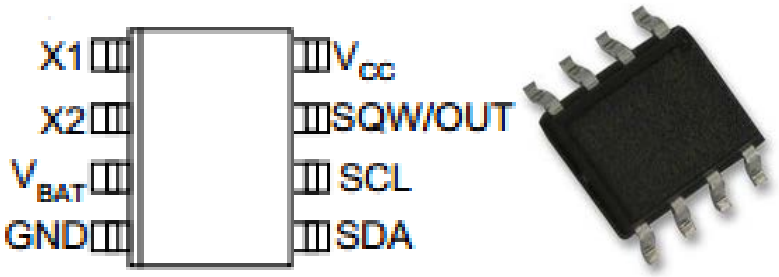


Figura 20: Patillaje y encapsulado del DS1307.

Diagrama de bloques

A continuación se muestra, en la figura 21, el diagrama de bloques del reloj en tiempo real.

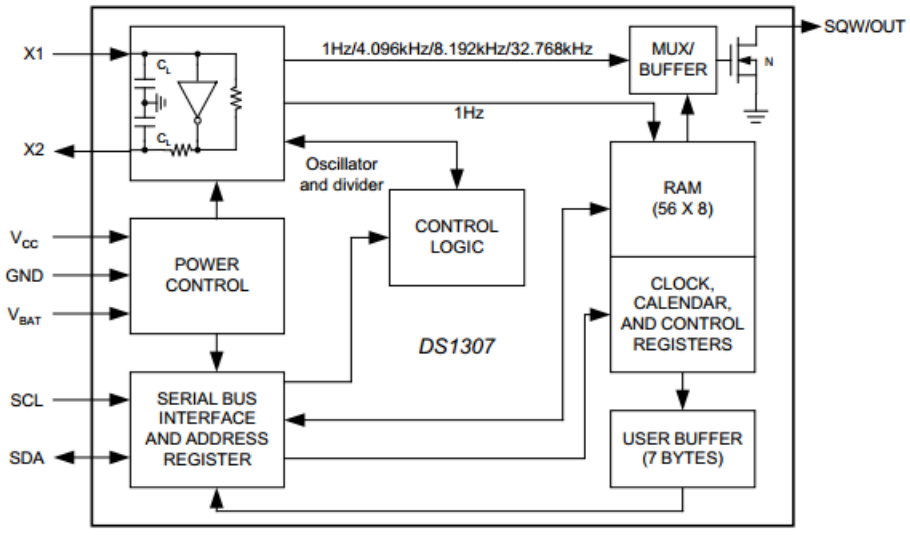


Figura 21: Diagrama de bloques del reloj en tiempo real.

2.5.2.2 Pila y porta pila.

Descripción.

El DS1307 requiere de una pila externa de 5v para que le suministre energía cuando el escudo se encuentra desconectado. Esta pila permite al componente mantener la información de la hora y actualizarla de forma correcta.

Una vez actualizada la hora en el componente, la pila no debe ser sustituida ya que la energía que necesita el DS1307 es muy baja y si se quitara se perdería la información guardada.

La pila elegida ha sido una CR2032, pila de muy bajo coste y muy usada en productos electrónicos. Esta batería de litio es capaz de alimentar al reloj DS1307, en ausencia de energía y a una temperatura media de 25ºC, durante más de 10 años. Es importante destacar que en otras condiciones de temperatura esta vida aproximada se podría reducir drásticamente.

Para la colocación de la pila en el escudo se ha utilizado un porta pilas, como el de la figura 22, compatible con las baterías CR20XX.

Figura 22: Porta pilas y batería CR2032.

2.5.2.3 Cristal de cuarzo.

Descripción.

El cristal de cuarzo utilizado para el montaje, figura 23, es uno de frecuencia 32.769kHz, indicado en las hojas de características del Ds1307 [13].

La calidad del cristal de cuarzo es muy importante ya que la exactitud de la hora viene dada por esta.

Este cristal tiene a una temperatura de 25ºC una desviación de 20 partes por millón por lo que cumple unas especificaciones mínimas de calidad

Figura 23: Cristal de cuarzo.

2.5.2.4 Esquemático.

La figura 24 muestra el montaje utilizado para reloj en tiempo real con sus principales componentes.

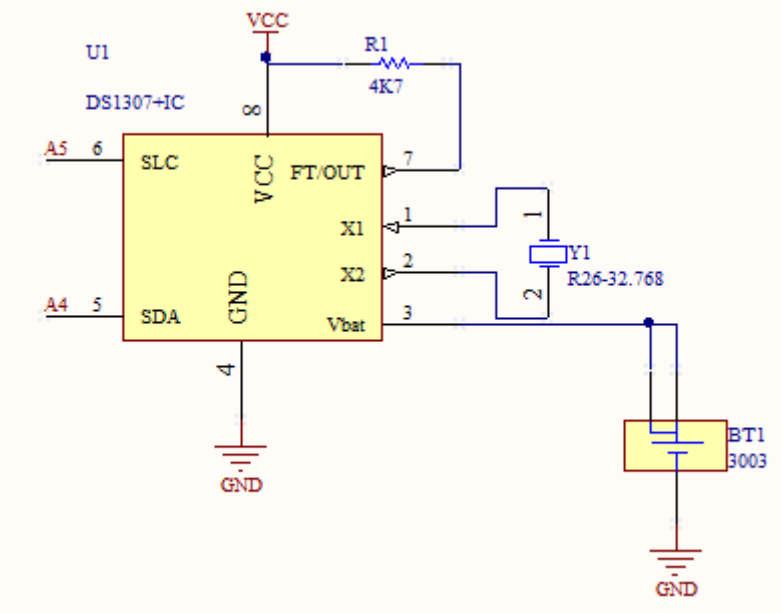


Figura 24: Esquemático reloj en tiempo real.

2.6 Bloque de Alimentación y Aislamiento.

2.6.1Descripción.

En la parte analógica, con el conversor A/D y sus protecciones, hay una separación física, o aislamiento, con la digital mediante dos componentes.

Esta separación tiene la funcionalidad de proteger el Hardware de Arduino de una mala utilización en el momento de conectar dispositivos a la entrada del conversor. En caso de conectar una tensión más alta de la que las protecciones del conversor no pudieran soportar, la parte analógica dejaría de funcionar pero sin alterar la placa principal.

Los componentes utilizados son un transformador DC/DC de 5V a 3.3V que proporciona la alimentación necesaria, y un aislador, que tiene la función de establecer un puente en la comunicación I2C, utilizada por el conversor, con la placa principal.

2.6.2Selección de componentes.

2.6.2.1 DC/DC 0503.

Descripción.

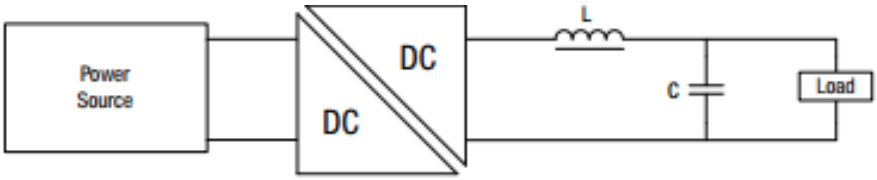
La fuente de alimentación utilizada para alimentar la parte analógica es un convertidor DC/DC de entrada 5 V y salida 3 V como el de la figura 25.

Este componente proporciona aislamiento, capaz de soportar picos de tensión de 1 KV durante un segundo, entre las dos partes creando una nueva referencia de tensión y una masa.

Pin	Function
1	-V_{IN}
3	+V_{IN}
5	NA
7	-V_{OUT}
8	+V_{OUT}
10	NA
12	NA
14	NA

Figura 25: Patillaje y encapsulado del DC/DC.

Para filtrar la tensión del convertidor DC/DC el fabricante indica, mediante la tabla de la figura 26, disponible en la hoja de características [15], los valores necesarios de una bobina en serie y un condensador que deben ser montados junto con el 0503.



	Inductor			Capacitor
	L, µH	SMD	Through Hole	C, µF
NTE0303MC	10	82103C	11R103C	4.7
NTE0305MC	47	82473C	11R103C	4.7
NTE0309MC	22	82223C	11R223C	2.2
NTE0312MC	10	82103C	11R103C	1
NTE0315MC	47	82473C	11R473C	1
NTE0503MC	10	82103C	11R103C	4.7
NTE0505MC	47	82473C	11R473C	4.7
NTE0505MEC	47	82473C	11R473C	4.7
NTE0506MC	10	82103C	11R103C	4.7
NTE0509MC	22	82223C	11R223C	2.2
NTE0512MC	47	82473C	11R473C	1
NTE0515MC	47	82473C	11R473C	1
NTE1205MC	47	82473C	11R473C	4.7
NTE1209MC	22	82223C	11R223C	2.2
NTE1212MC	47	82473C	11R473C	1
NTE1215MC	47	82473C	11R473C	1

Figura 26: Esquema de montaje y tabla de valores.

En nuestro caso el componente utilizado es el NTE0503MC por lo que es necesario incluir en el montaje una bobina de 10 µH y un condensador de 4,7 µF.

2.6.2.2 Aislador I2C ADuM1250.

Descripción.

Para la comunicación entre el conversor A/D, situado en la parte analógica, con la placa de Arduino es necesario un componente que realice un puente entre el aislamiento.

El componente utilizado es un aislador específico para I2C. Este componente dispone de dos canales bidireccionales de comunicación y permite velocidades de trasmisión de hasta 1 MHz por lo que es compatible con nuestra velocidad de comunicación, 100 KHz.

El ADuM1250 de la figura 27 [14], pese a ser un aislador magnético, presenta una gran inmunidad frente a campos magnéticos externos. Además incluye un circuito interno de protección para evitar que la conexión de un componente no alimentado al bus I2C introduzca datos inválidos.

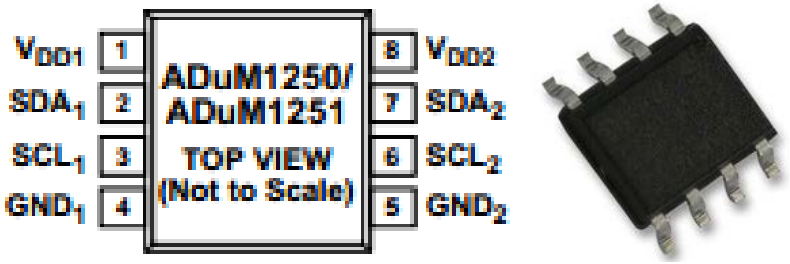


Figura 27: Patillaje y encapsulado del Aislante ADuM1250.

Diagrama de bloques.

A continuación, en la figura 28, se muestra el diagrama de bloques del aislador.

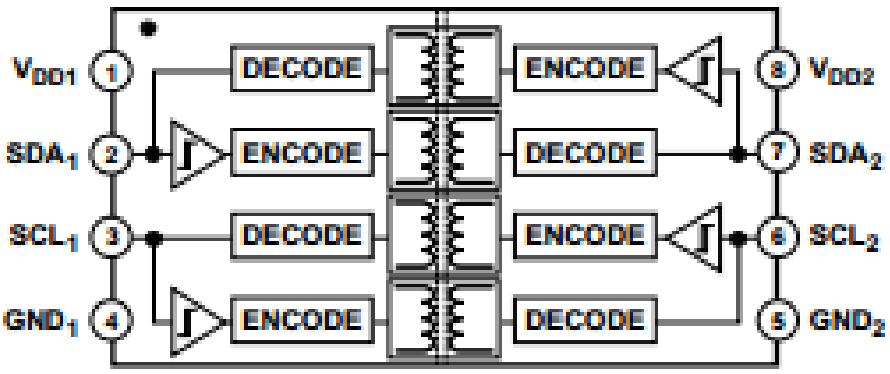


Figura 28: Diagrama de bloques de la tarjeta micro SD.

2.6.2.3 Esquemático.

DC/DC.

La figura 29 muestra el montaje utilizado para el convertidor DC/DC, con sus principales componentes.

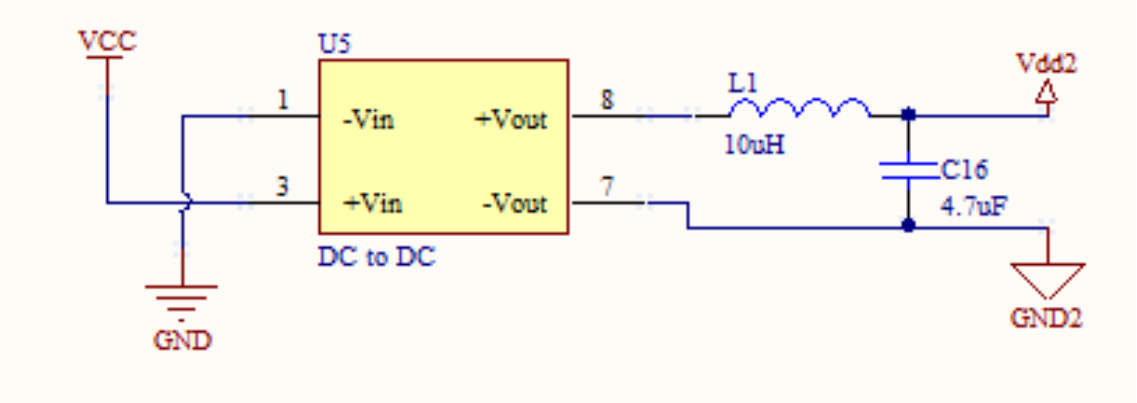


Figura 29: Esquemático del DC/DC 0503.

Aislamiento.

La figura 30 muestra el montaje utilizado para el aislante, con sus principales componentes.

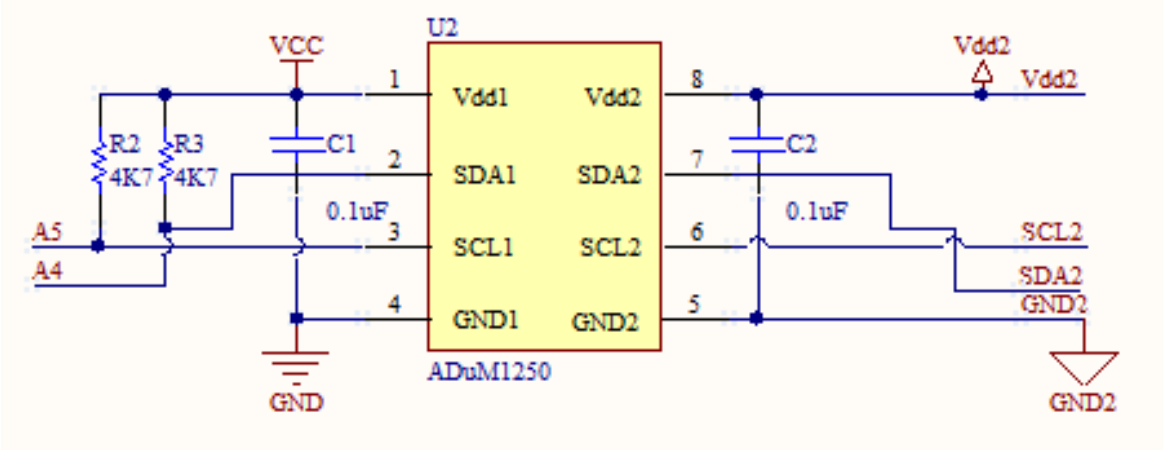


Figura 30: Esquemático del aislante ADuM1250.

2.7 Bloque Conversor A/D.

2.7.1Descripción.

El bloque del conversor A/D se puede dividir en dos partes:

En primer lugar tenemos el conversor, un MCP3424 de 18 bits que mejora el conversor de la placa Arduino, 10 bits.

En segundo lugar, para proteger las entradas del conversor de tensiones elevadas y malos usos, se han incluido una serie de protecciones pasivas compuestas por resistencias, condensadores y diodos.

Estas configuraciones son modificables en base a las necesidades de uso del usuario. Y las posibles conexiones serian:

- Entrada Unipolar.
- Puente de wheatstone.
- Sonda de termopar.

2.7.2Selección de componentes.

2.7.2.1 MCP3424.

Descripción.

El MCP3424 de la figura 31, [12] es un conversor *ΔΣ* multicanal de 18 bits con un bajo ruido y una gran precisión.

La tensión de referencia de 2,048V, incluida dentro del componente, permite un rango de entrada de +-2,048V en una entrada diferencial.

Este conversor dispone de 4 salidas diferenciales con ganancia ajustable a 1, 2, 4 u 8, resolución programable entre 18, 16, 14 o 12 bits y puede realizar conversiones simples o continuas. Durante estas conversiones, el conversor compensa la desviación y los errores de ganancia automáticamente.

Este componente es adecuado para labores de adquisición de datos de forma autónoma ya que el modo de conversión simple o "un disparo" dispone de un consumo reducido entre las conversiones.

Para la comunicación este componente utiliza el protocolo I2c, y mediante la escritura de las direcciones indicadas en su hoja de características se pueden configurar todas las funcionalidades del conversor.

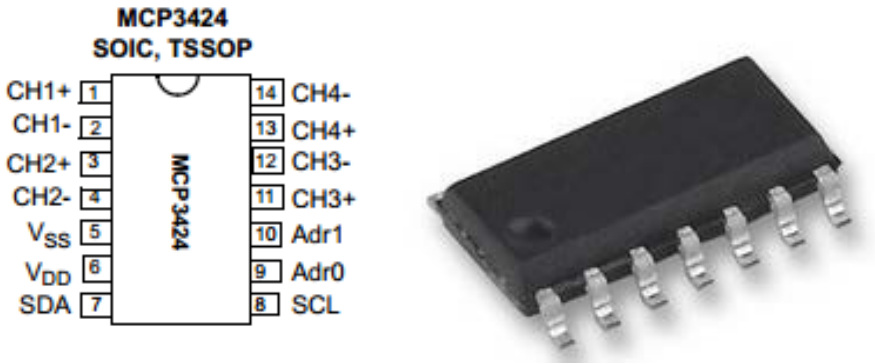


Figura 31: Patillaje y encapsulado del Conversor MCP3424.

En la tabla de la figura 32 se muestra las muestras por segundo que el conversor es capaz de tomar en función de su resolución.

18 Bits	3,75 SPS
16 Bits	15 SPS
14 Bits	60 SPS
12 Bits	240 SPS

Figura 32: Tabla de relación entre resolución y muestras.

Diagrama de bloques.

En el diagrama de bloques de la figura 33 observamos los bloques internos del conversor.

En él, se pueden apreciar dos bloques muy importantes:

Dispone de una tensión de referencia interna de 2.048 V, de esta forma no es necesario montar, de forma externa, ningún tipo de referencia.

Por otro lado, tiene un oscilador por lo tanto tampoco en necesario incluir un cristal de cuarzo, como si ocurría en el reloj en tiempo real.

Para la selección de canal incorpora un multiplexor interno y un amplificador de ganancia programable a la entrada del conversor.

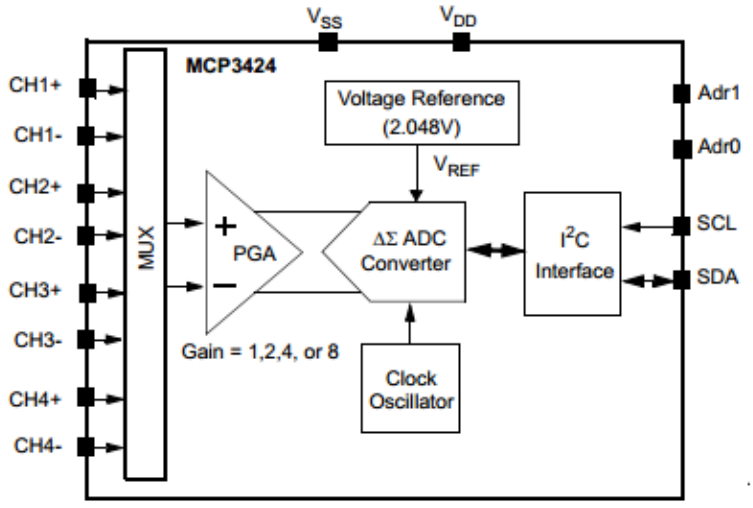


Figura 33: Diagrama de bloques del conversor MCP3424.

Configuración del montaje.

Este componente tiene la posibilidad mediante los pines Adr0 y Adr1 de seleccionar la dirección deseada. Para realizar este direccionamiento estos dos pines se tienen que referenciar a masa o Vcc según la tabla de la figura 34.

I^2C Device Address Bits			Logic Status of Address Selection Pins	
A2	A1	A0	Adr0 Pin	Adr1 Pin
0	0	0	0 (Addr_Low)	0 (Addr_Low)
0	0	1	0 (Addr_Low)	Float
0	1	0	0 (Addr_Low)	1 (Addr_High)
1	0	0	1 (Addr_High)	0 (Addr_Low)
1	0	1	1 (Addr_High)	Float
1	1	0	1 (Addr_High)	1 (Addr_High)
0	1	1	Float	0 (Addr_Low)
1	1	1	Float	1 (Addr_High)
0	0	0	Float	Float

Figura 34: Configuración de direcciones.

En este diseño el pin Adr0 se ha conectado a masa y Adr1 a Vcc. Con esta configuración la dirección de I2C seria según la tabla de la figura 34 la 010. Sustituyendo los valores de A2, A1 y A0 en la tabla de la figura 35 se obtiene la dirección final del conversor, 1101010 o lo que es lo mismo, 6A en hexadecimal.

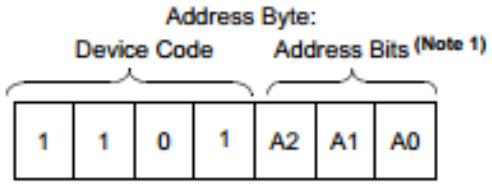


Figura 35: Dirección de memoria.

Esta dirección es necesaria configurarla en el código del conversor para su correcto funcionamiento. Si esta dirección no estuviera configurada correctamente interferiría con la comunicación del reloj en tiempo real.

Conectando la sonda del osciloscopio se puede comprobar que el direccionamiento es correcto observando las ondas de las líneas del I2C, SCL y SDA. Las ondas correctas deben ser como las de la figura 36.

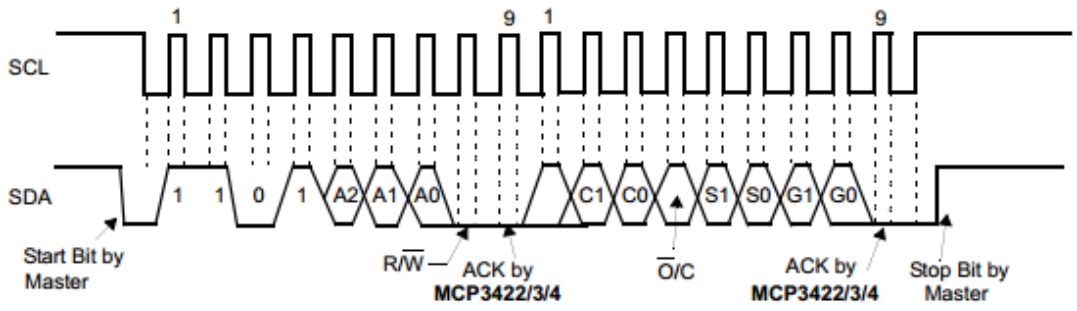


Figura 36: Ondas teóricas del conversor MCP3424.

2.7.3Sistema de Filtro y Protección.

2.7.3.1 Descripción.

La placa *S-LOGGER,* al igual que la mayoría de placas de adquisición de datos, incluye una serie de protecciones con la finalidad de evitar la destrucción por una mala manipulación de las entradas del conversor.

Estas protecciones han sido diseñadas para incorporar las acomodaciones necesarias de los sensores directamente en la *S-LOGGER,* de esta forma la conexión de los sensores es más sencilla y rápida. Para este estudio se han utilizado [2] y [3].

Para realizar las diferentes conexiones se ha diseñado un circuito único, como el de la figura 37, en el que mediante la soldadura de los componentes adecuados tendremos la acomodación correcta.

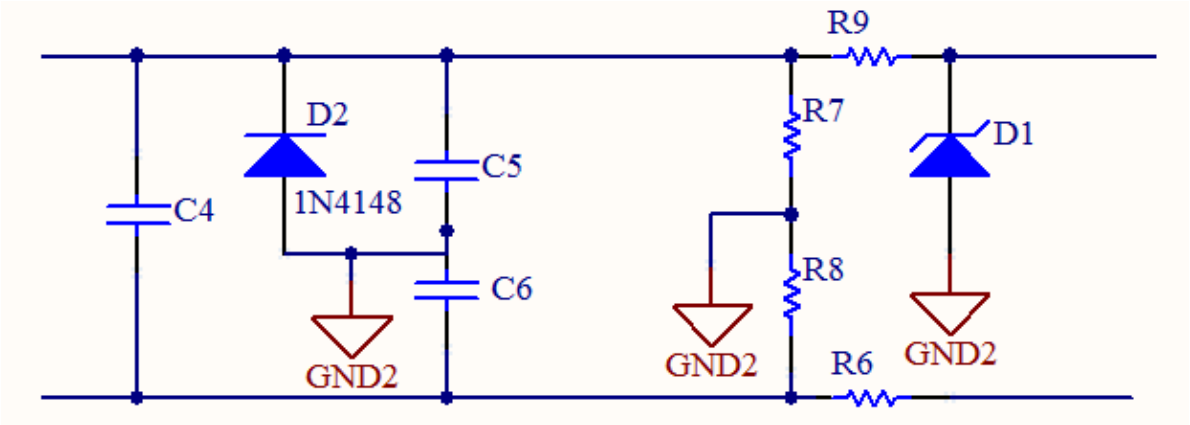


Figura 37: Circuito general de protecciones.

2.7.3.2 Configuraciones.

Entrada Unipolar.

Esta entrada está diseñada para la conexión de sensores referenciados a masa.

La tensión máxima soportada en la entradas de conversor es de Vcc, en nuestro caso 3 V, por lo que mediante el uso de un divisor de tensión de relación 1/10 esta tensión se eleva a 30 V.

Para el filtrado de la señal de entrada se incluye un filtro paso bajo y para las tensiones inversas y transitorios de voltaje se han usado diodos de señal 4148 y diodos TVS respectivamente.

En la figura 38 se muestra el esquema de protecciones para la entrada unipolar y en la figura 39 los componentes necesarios para su montaje

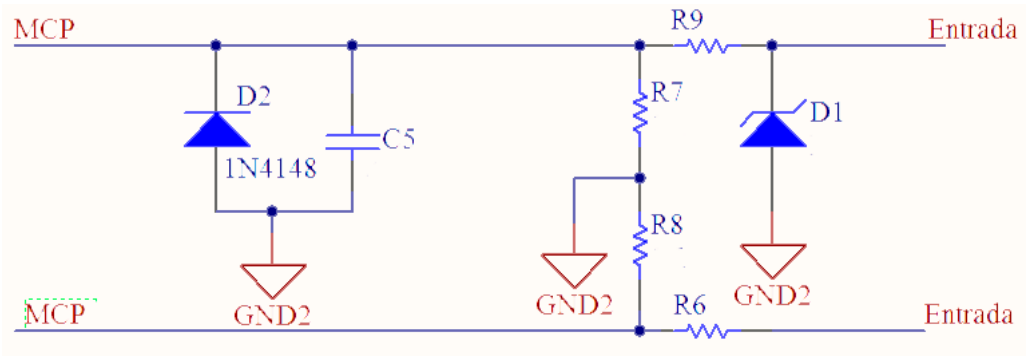


Figura 38: Circuito de protección de entrada unipolar.

Componentes	Valor
C4,C7,C10,C13	-
D2,D3,D5,D7	1N4148
C5,C8,C11,C14	100 nF
C6,C9,C12,C15	-
R6,R10,R14,R18	0 Ω
R7,R12,R16,R20	1K5 Ω
R8,R13,R17,R21	0 Ω
R9,R11,R15,R19	12 KΩ
D1,D4,D6,D8	TVS 40 V

Figura 39: Relación de componentes del circuito de entrada unipolar.

A continuación se van a detallar todos los cálculos necesarios para la selección de componentes del divisor de tensión y para el filtro de paso bajo. Para estos cálculos se ha utilizado bibliografía [5], así como diversos programas para el cálculo de filtros y divisores como *Electronic TB.*

El primer paso en los cálculos fue la selección del filtro de paso bajo. Este filtro, como el de la figura 40, tiene la función de permitir el paso de las frecuencias más bajas y atenuar las frecuencias más altas.

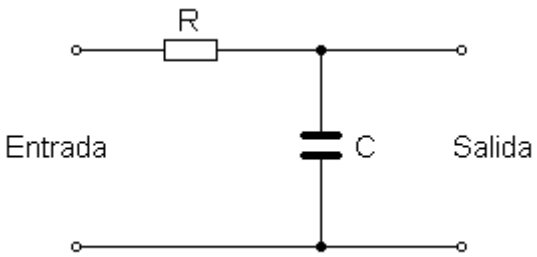


Figura 40: Filtro paso bajo.

El cálculo de un filtro paso bajo se realiza mediante la siguiente formula.

$$R = \frac{1}{2 * \pi * f_g * C}$$

Siendo f_g la frecuencia de corte la cual tiene un valor de 120 Hz.

El condensador se ha fijado a 100 nF, por lo que sustituyendo en la formula el valor de R es de 13,26 KΩ. Como es un diseño real y el valor de esta resistencia no es normalizado, se ha seleccionado el valor más cercano disponible 12 KΩ.

El valor máximo de tensión de entrada del conversor es de 2,048V y el divisor deberá proteger de tensiones aproximadamente 10 veces mayores. Por lo que mediante esta condición se ha procedido al cálculo del divisor de tensión, figura 41.

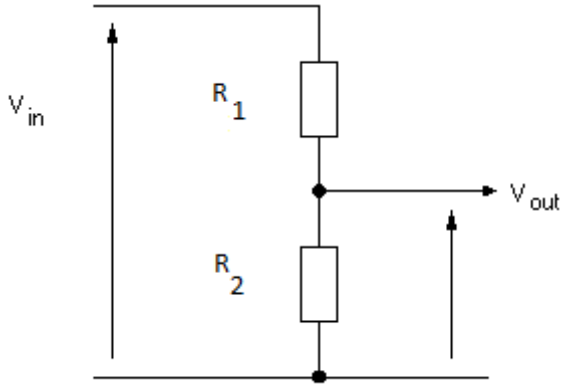


Figura 41: Divisor de tensión.

En este circuito la única incógnita que tenemos es R2, R1 es la misma que la del filtro paso bajo, 12 KΩ, la tensión de salida es la máxima del conversor 2,048 V y la de entrada 10 veces la tensión de entrada 20,48 V.

Para el cálculo de esta resistencia se usa la siguiente formula.

$$R_2 = V_{out} * \left(\frac{R_1}{V_{in} - V_{out}}\right)$$

Sustituyendo los valores R2 da un valor de 1,33 KΩ, al igual que en la selección de la resistencia del filtro, este valor no es normalizado por lo que se selecciona 1K5 Ω dando lugar a un filtro final de 1/11.

Para comprobar si el filtro tiene un buen acoplamiento con el conversor hay que comprobar que la resistencia del conversor es, como mínimo, 10 veces menor que la resistencia de entrada del conversor.

Según su hoja de características [12], la resistencia de entrada es de 25 MΩ en modo común y 2,25 MΩ en modo diferencial, por lo que cumple la condición.

Puente de Weatstone.

La entrada para el puente de Weatstone o diferencial tiene como protección un filtro de paso bajo para mejorar la lectura de los datos tomados.

El puente de resistencias será montado en un circuito anexo a nuestro escudo, conectando únicamente la salida del puente.

En la figura 42 se muestra el esquema de protecciones del puente de Weatstone y en la figura 43 los componentes necesarios para su montaje

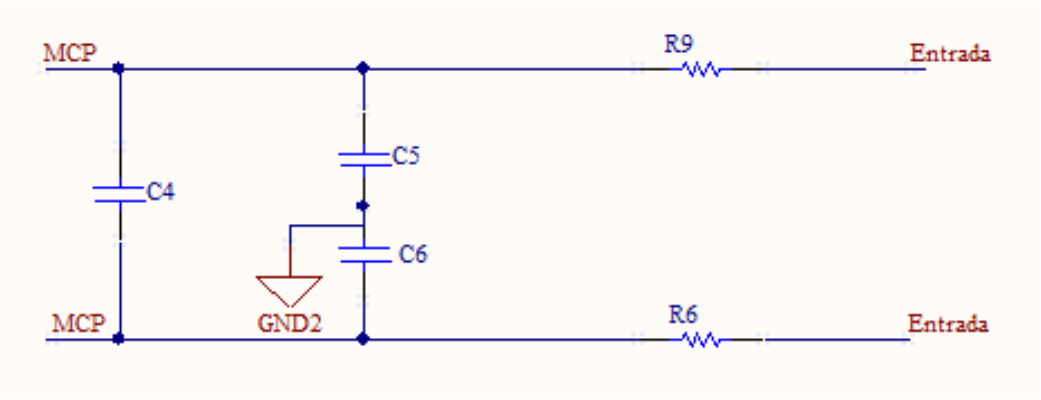


Figura 42 Circuito de protección de entrada de puente de Weatstone.

Componentes	Valor
C4,C7,C10,C13	10 nF
D2,D3,D5,D7	-
C5,C8,C11,C14	100 nF
C6,C9,C12,C15	100 nF
R6,R10,R14,R18	12 KΩ
R7,R12,R16,R20	-
R8,R13,R17,R21	-
R9,R11,R15,R19	12 KΩ
D1,D4,D6,D8	-

Figura 43 Relación de componentes del circuito de puente de Weatstone.

Para la selección del filtro de paso bajo se han utilizado los mismos cálculos del apartado de entrada unipolar.

Para el montaje final de este circuito es necesario el puente de Weatstone.

La configuración del puente vendrá dada por las necesidades del usuario final, así como de los diferentes sensores que desee utilizar: Galgas extensiométricas, NTCs, etc.

Para estabilizar las muestras tomadas se monta un condensador (C4, C7, C10, C13) en paralelo con las 2 entradas del conversor.

Sonda de termopar.

Esta entrada está diseñada para la conexión de sondas de termopar.

Esta entrada únicamente tiene dos resistencias de 10 KΩ conectadas a tierra para adaptar la entrada. Y un condensador en paralelo con las entradas para estabilizar las muestras tomadas.

Para la calibración de temperatura de la sonda de termopar será necesario un termómetro externo al escudo.

A continuación, en la figura 44, se muestra el esquema final de montaje y en la figura 45 los componentes necesarios para el montaje de sonda de termopar.

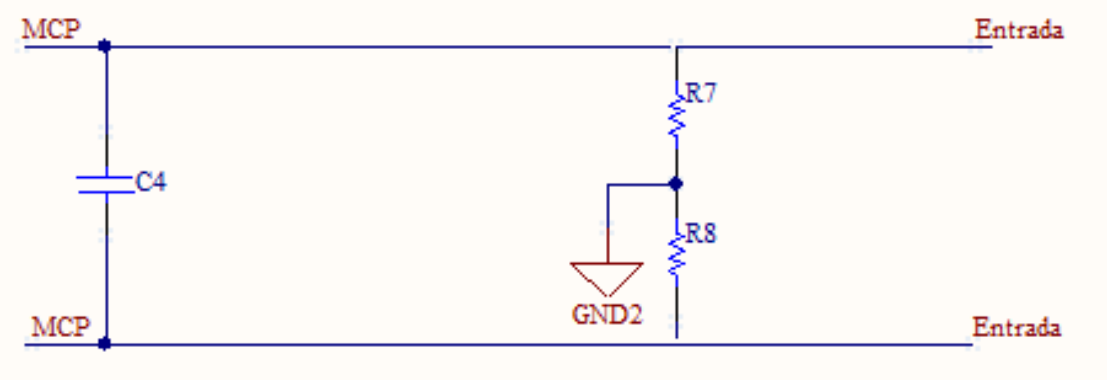


Figura 44: Circuito de protección de entrada de sonda termopar.

Componentes	Valor
C4,C7,C10,C13	10 nF
D2,D3,D5,D7	-
C5,C8,C11,C14	-
C6,C9,C12,C15	-
R6,R10,R14,R18	0 Ω
R7,R12,R16,R20	10 KΩ
R8,R13,R17,R21	10 KΩ
R9,R11,R15,R19	0 Ω
D1,D4,D6,D8	-

Figura 45 Relación de componentes del circuito de entrada de sonda termopar.

Esquemático.

La figura 46 muestra el montaje utilizado para el conversor, con sus principales componentes.

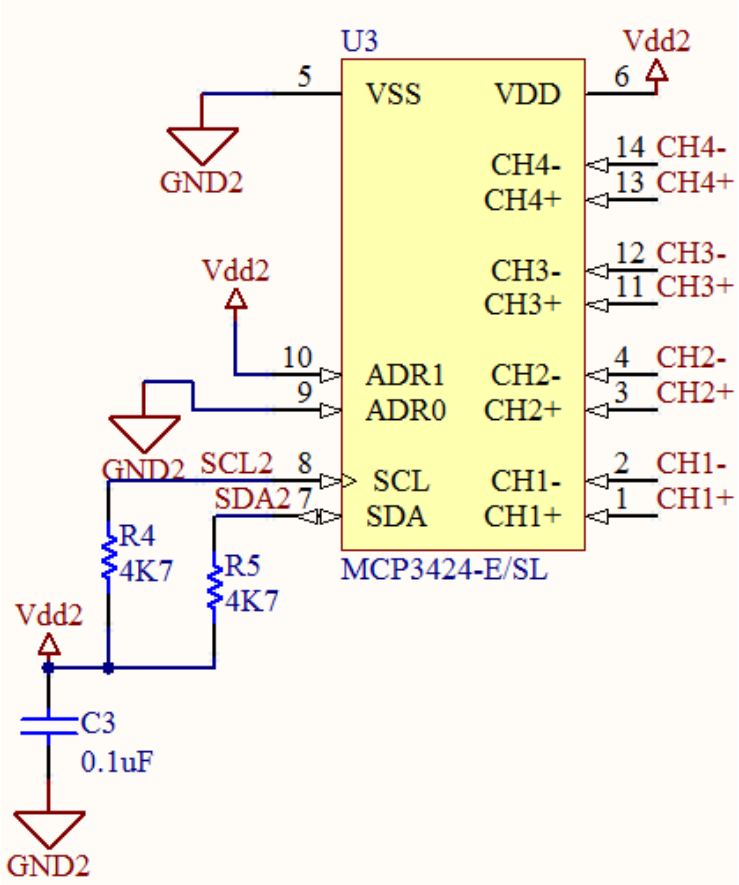


Figura 46: Esquemático integrado MCP3424

La figura 47 muestra el montaje utilizado para las protecciones, con sus principales componentes. Este circuito es el de una de las protecciones, pero esta replicado para cada uno de los canales.

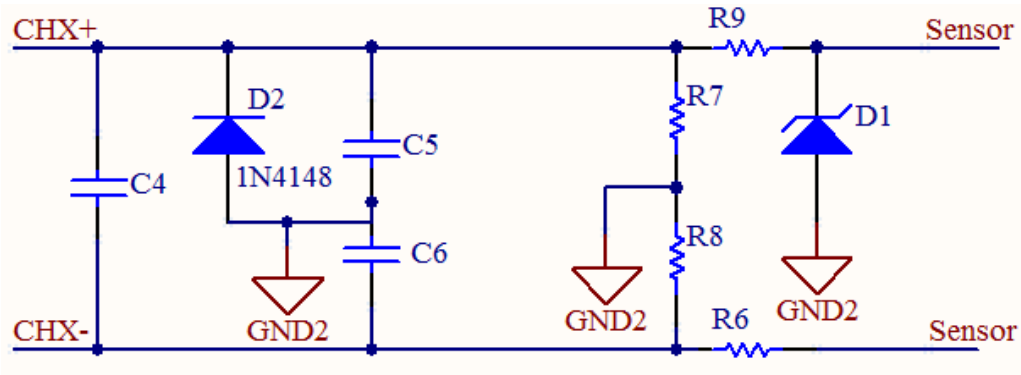


Figura 47: Circuito de protecciones.

3. Diseño de la PCB.

3.1 Introducción.

El programa de utilizado para el diseño de la PCB es el *Altium Designer 10*. La figura 48 muestra el interfaz de la pantalla donde se diseña el esquemático.

La plataforma de Arduino aconseja usar software libre, como *EAGLE,* para de esta forma favorecer la difusión de los documentos.

El motivo de uso de este software es la potencia y facilidad a la hora de realizar el proyecto, y su compatibilidad de los archivos a la hora de fabricar la *PCB* en empresas especializadas.

Para el estudio del manejo de este software se utilizó la bibliografía [1].

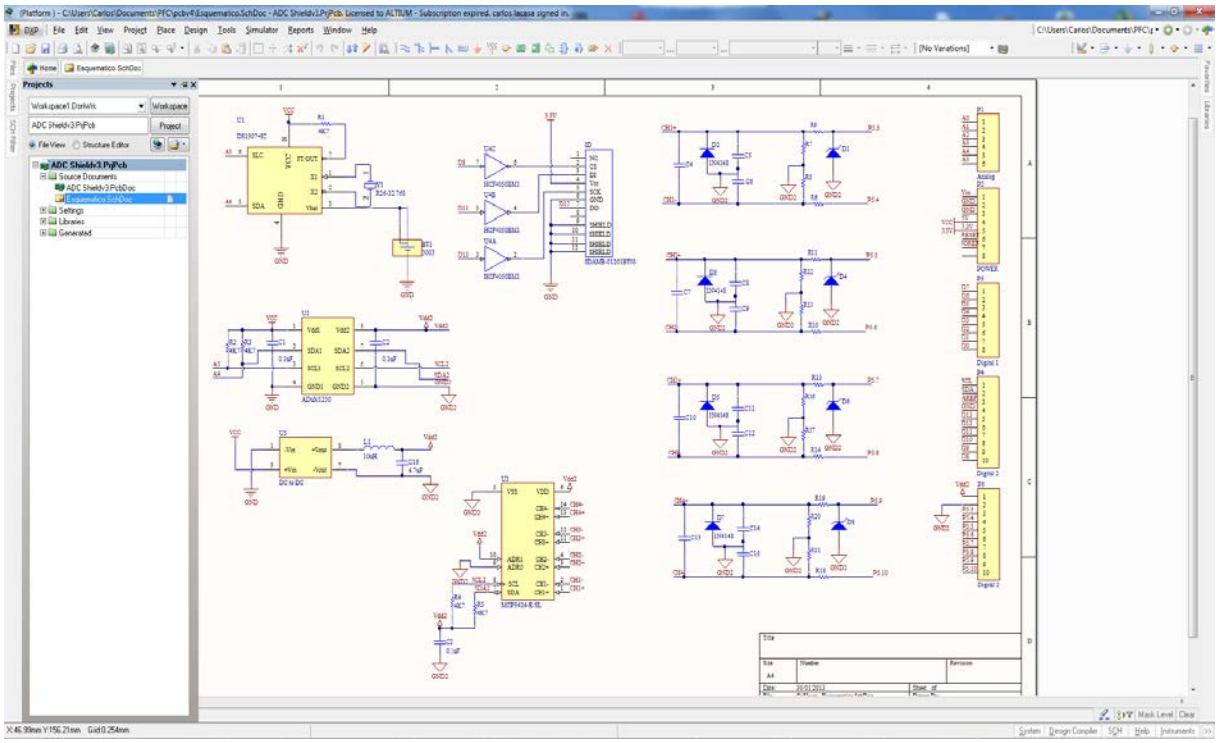


Figura 48: Interfaz *Altium Designer.*

A continuación se muestra el esquemático del escudo *S-LOGGER,* figura 49.

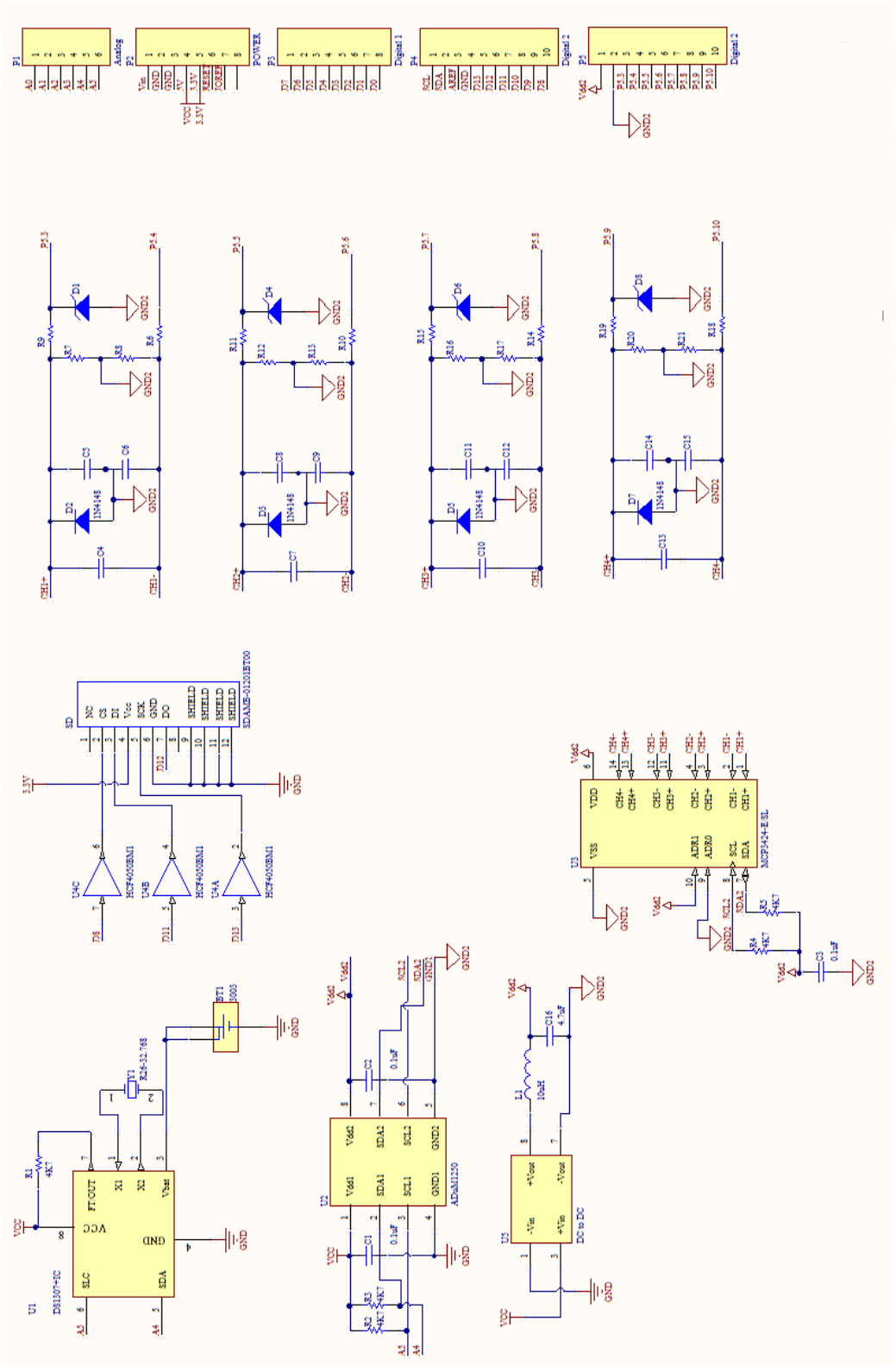


Figura 49: Esquemático *S-LOGGER*.

3.2 Implementación de la S-LOGGER.

Para el diseño de la PCB el primer requisito era el tamaño. La placa *S-LOGGER* debe ser compatible con la tarjeta UNO de Arduino, figura 50, y sus accesorios, por lo que su tamaño debe ser similar y los pines de conexión coincidir.

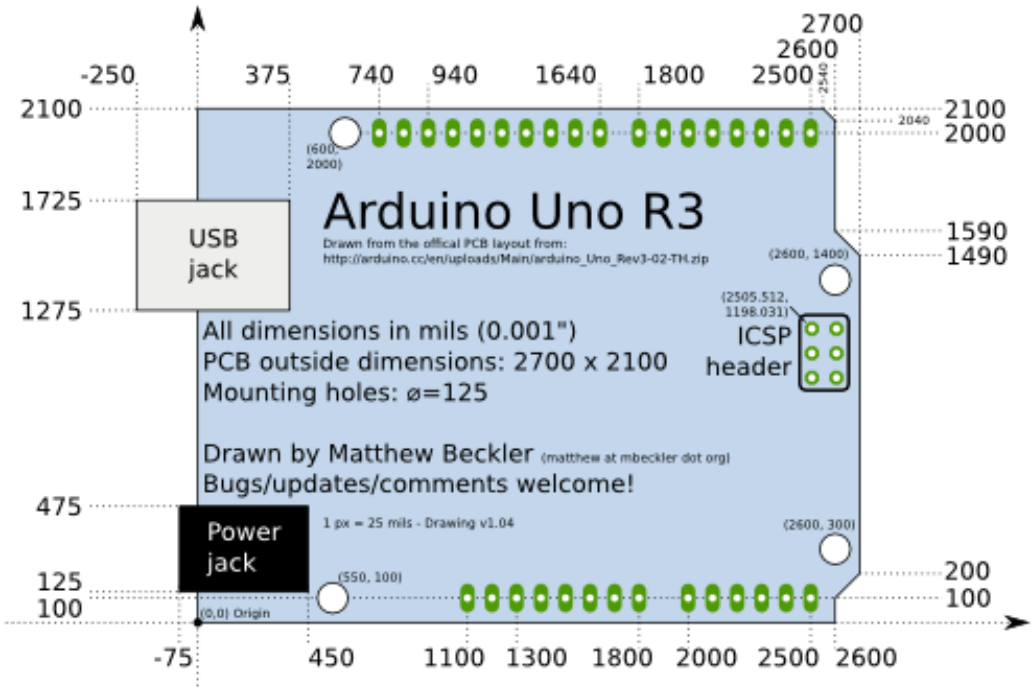


Figura 50: Placa Arduino UNO con medidas.

La tarjeta Arduino uno mide 68,58mm x 53,34mm

La versión R3 dispone de 4 pines adicionales respecto a la versión R1, en el diseño final se han incluido estos 4 pines adicionales pero no se usan.

Por lo tanto el primer paso del diseño de la tarjeta fue definir las medidas y fijar los pines de conexión en el lugar correcto. De forma adicional, se han incluido 4 orificios para hacer el escudo *S-LOGGER* compatible con las cajas de Arduino, como la de la figura 51.

Figura 51: Placa Arduino UNO y caja compatible.

3.2.1Colocación de los componentes.

Con el esquemático terminado y los encapsulados de los componentes seleccionados se procedió a la colocación de los componentes en la PCB.

La estrategia utilizada fue la de separar la PCB en 3 zonas, correspondientes a cada uno de los bloques funcionales.

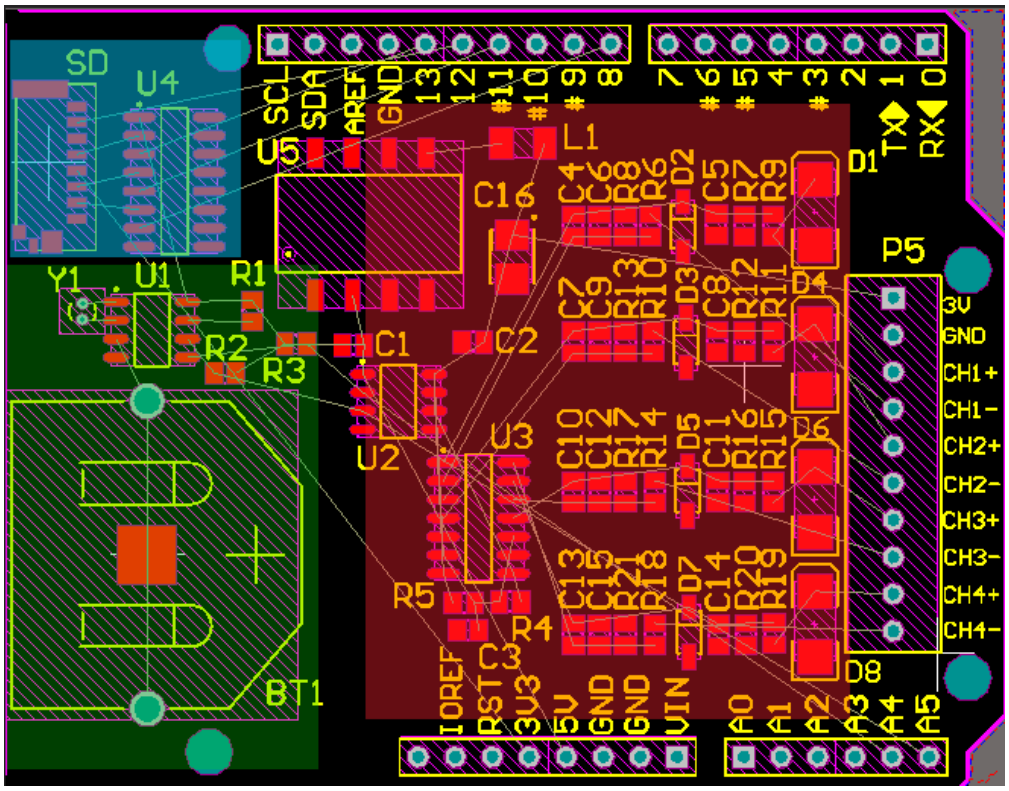


Figura 52: *PCB* con las 3 zonas diferenciadas.

Las 3 zonas que se ven en la figura 52 son:

- Azul: Tarjeta SD
- Verde: Reloj en tiempo real
- Rojo: Conversor A/D con sus protecciones.

En cada uno de los bloques se ha colocado primero el componente de mayor tamaño y después las resistencias y condensadores.

En las protecciones del converso A/D debido a la gran densidad de componentes que presentan y que son componentes configurables, se ha buscado la sencillez en el diseño utilizando una estrategia simétrica, que se muestra en la figura 53, en todas ellas.

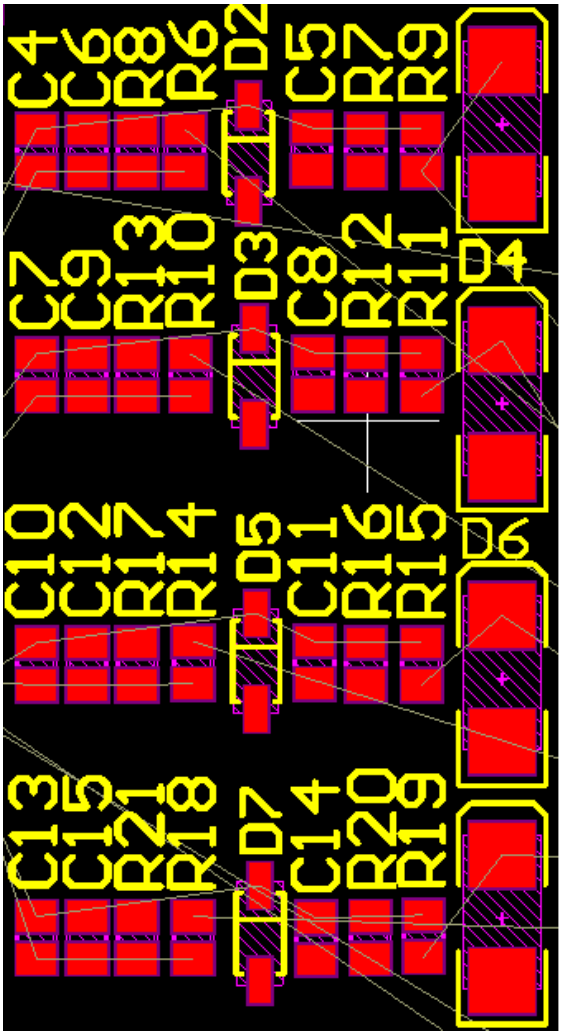


Figura 53: Estrategia de colocación de las protecciones del conversor.

La anchura mínima de las pistas de circuito impreso ha sido calculada utilizando las gráficas que nos proporciona la norma UNE 20-621-84/3, incluidas en la bibliografía [4]. Analizando la placa de Arduino se observa que sus salidas de 5 V y 3,3 V proporcionan una corriente de máxima de 40 mA y 50 mA y la fuente de alimentación del DC/DC proporciona a su salida un máximo de 300 mA.

Para el cálculo de las anchuras de pistas se facilita a continuación una gráfica en la que se especifica el ancho de pista recomendado para una corriente determinada.

Antes de proceder a analizar el ancho de pista, se debe tener en cuenta que esta PCB tendrá un espesor de pista de 35 µm.

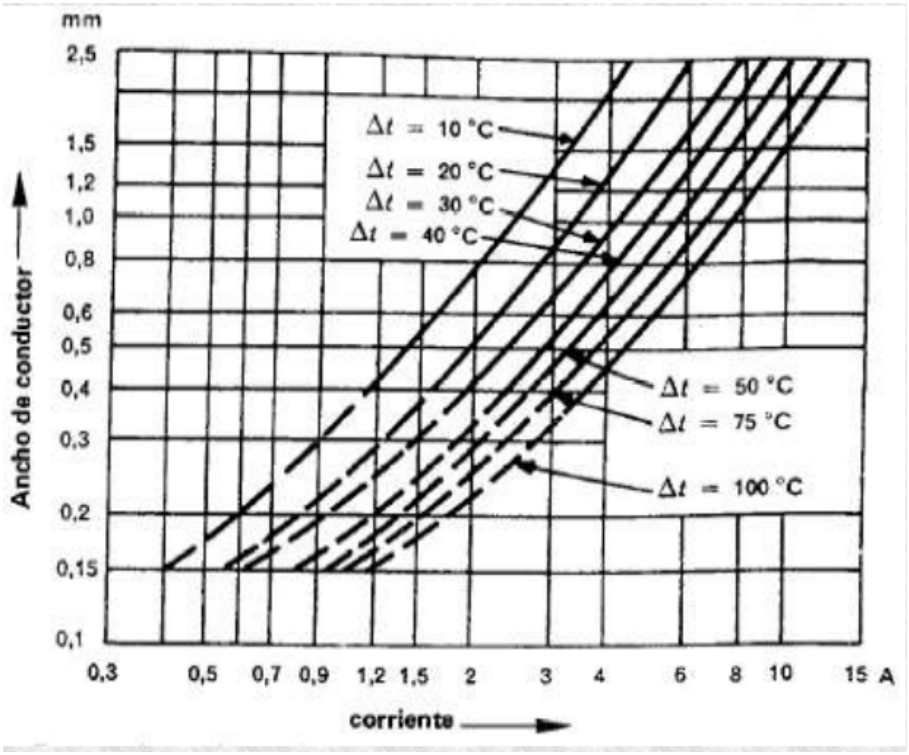


Figura 54: Gráfica para el cálculo de ancho de pistas en *PCBs.*

Por lo tanto atendiendo a la gráfica de la figura 54, y tomando como valor de corriente 300 mA, se puede calcular la anchura mínima recomendada.

Al ser una corriente tan pequeña no hay un ancho de pista mínimo, por lo que el ancho de pista utilizado ha sido de 0.254mm, medida máxima para poder realizar pistas entre los *pads* de los circuitos impresos.

En cuanto a la estrategia de ruteo se toma la decisión de usar doble capa, ya que con la densidad de componentes y conexiones es necesario realizar pistas por la cara *top y bottom* y unirlas mediante vías.

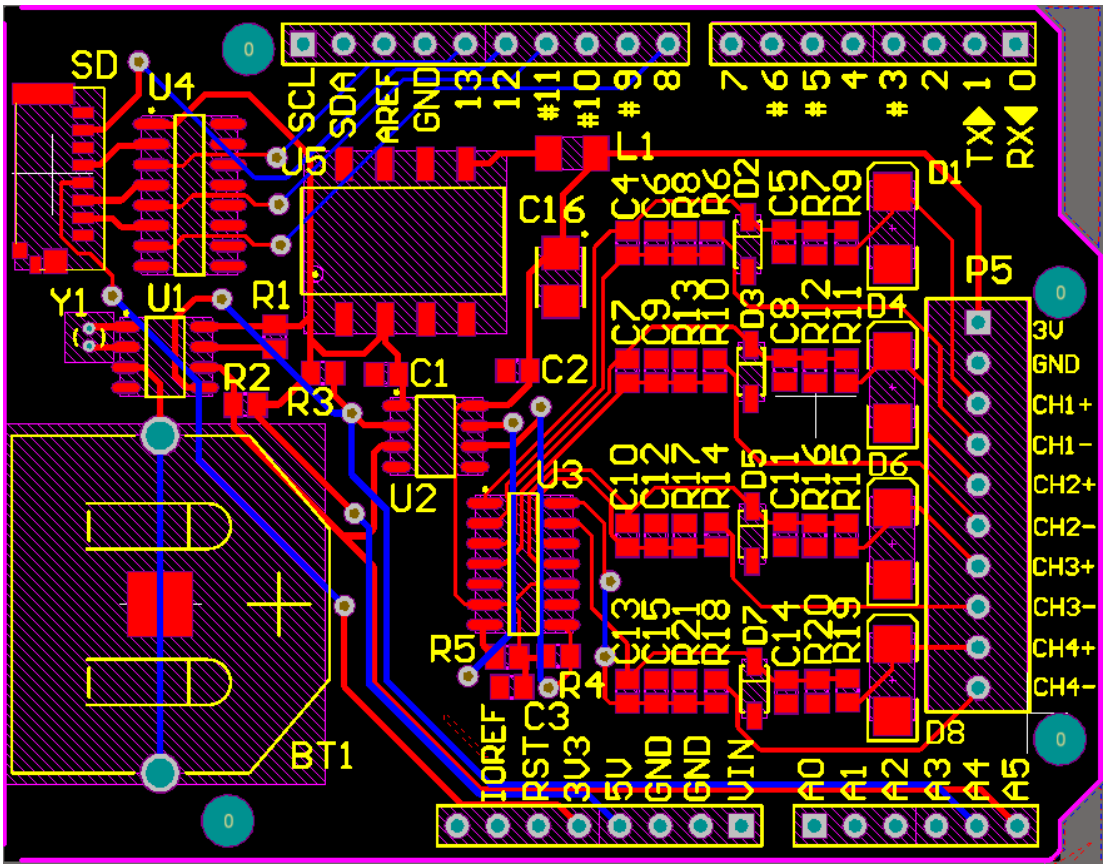


Figura 55: *PCB* con las pistas ruteadas.

En la figura 55 se muestran las diferentes pistas utilizadas, las pistas en rojo son en la cara *top,* y las azules en la *bottom*.

El último paso del diseño de la *PCB* es la formación de los planos de masa.

Los planos de masa constituyen un área de cobre gigante que hace de tierra. Esto provoca una gran cantidad de conexiones de *GND* sobre la *PCB*, lo que a su vez actúa de refuerza el blindaje de la misma frente a interferencias externas; igualmente se minimiza el ruido y el acoplamiento entre pistas.

En los planos de masa hay que evitar la formación de islas de cobre sin conectar a masa, ya que al no tener conexión podrían convertirse en una antena y producir interferencias en el funcionamiento del diseño.

En la *S-LOGGER* se pueden apreciar dos planos de masa en cada una de las caras, visibles en la figura 56, uno corresponde al plano de masa de la parte digital *GND* y el otro pertenece a la parte analógica *GND2.*

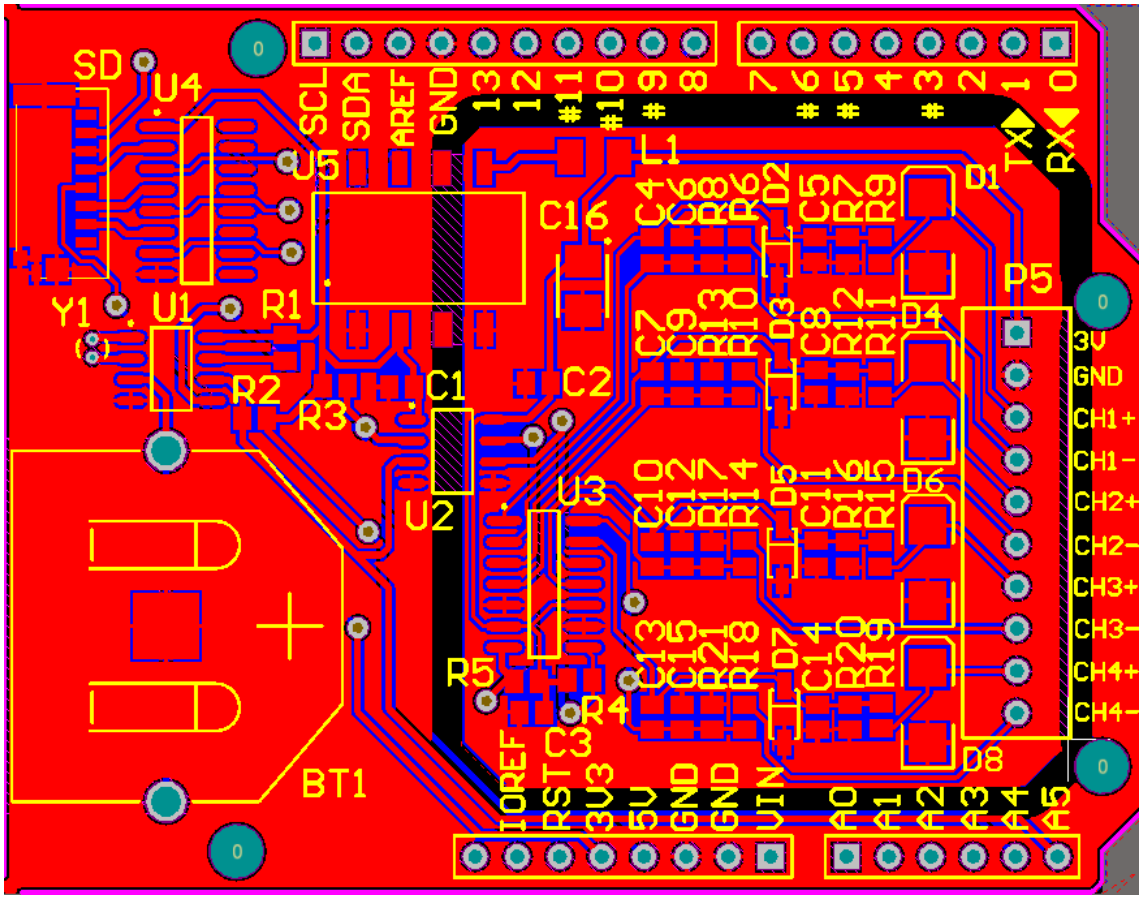


Figura 56: *PCB* con los planos de masa definidos.

3.2.2Diseño final en 3D.

La herramienta *Altium Designer* permite realizar un diseño en tres dimensiones de la *PCB* como el de la figura 57. Esta herramienta es muy útil para observar la colocación de los componentes y evitar problemas en la posterior fabricación.

La mayoría de los cuerpos en 3D están diseñados en las propias librerías del programa, pero algunos componentes específicos como el zócalo para la SD o el porta pilas hay que crearlos o descargarlos de webs específicas de diseño en 3D.

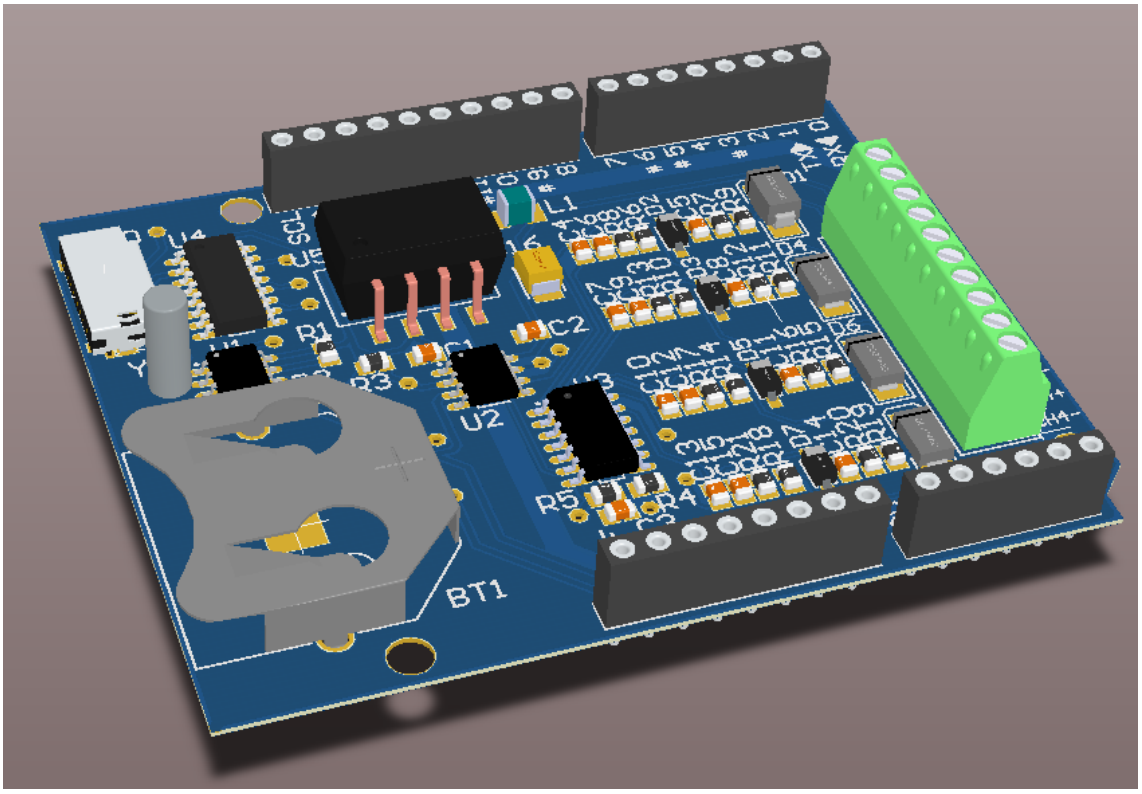


Figura 57: Diseño final en 3D.

3.3 Presupuesto.

Elemento	Proveedor	Referencia	Precio	Cantidad	Precio final
DC/DC 0503	Rs	689-5463	8,92 €	1	8,92 €
Aislante ADUM1251	Rs	427-165	6,46 €	1	6,46 €
MCP3424 18BIT	Rs	669-6086	4,34 €	1	4,34 €
RTC DS1307	Rs	761-9759	3,03 €	1	3,03 €
Cristal 32,768Khz	Rs	672-7593	0,17 €	1	0,17 €
Retenedor pila CR2032	Rs	219-7960	1,23 €	1	1,23 €
Bloque terminal 5 vías	Rs	756-0994	2,12 €	2	4,24 €
74HC4050D	Rs	483-6544	0,32 €	1	0,32 €
Conector Micro SD	Rs	702-5485	0,69 €	1	0,69 €
Bobina 10µH 320mA	Rs	191-0037	1,52 €	1	1,52 €
Diodo TVS 40v	Rs	710-3487	0,24 €	4	0,96 €
Diodo 1N4148W	Rs	451-2727	0,04 €	4	0,16 €
Condensador tántalo 4,7µF	Rs	464-7782	0,39 €	1	0,39 €
Resistencia SMD 0805 4K7Ω	Rs	721-6993	0,11 €	5	0,57 €
Resistencia SMD 0805 0Ω	Rs	721-6814	0,11 €	8	0,88 €
Resistencia SMD 0805 12KΩ	Rs	721-7028	0,11 €	4	0,44 €
Resistencia SMD 0805 1K5Ω	Rs	721-6959	0,11 €	4	0,44 €
Condensador SMD 0805 0,1µF	Rs	648-0979	0,02 €	7	0,11 €
Arduino header kit	Rs	86218-A	1,30 €	1	1,30 €
PRECIO TOTAL					36,17 €

El cálculo de coste de la *PCB* se ha realizado en una empresa especializada en el diseño y fabricación [11].

Para realizar un presupuesto real del precio de una PCB se ha simulado el pedido de 50 placas.

El precio final por placa era de 3,363€.

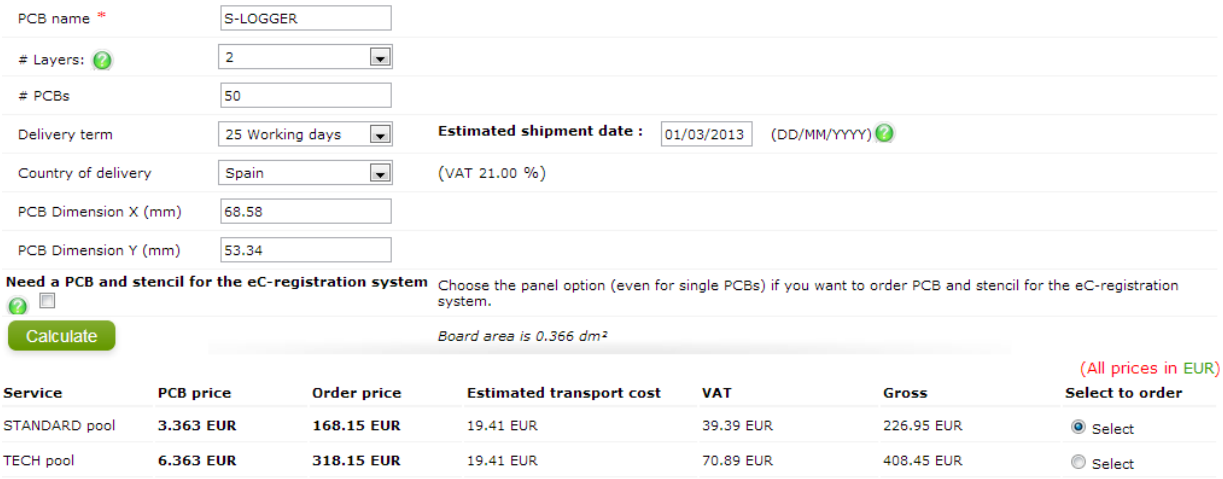


Figura 58: Presupuesto PCB en Eurocircuits.

Por lo tanto contabilizando el gasto de material, 36,17€, y el gasto de una placa, 3,363€. **El gasto total de fabricación de una tarjeta *S-LOGGER* es de 39,53€.**

Hay que tener en cuenta que el gasto de material está calculado para un pedido simple de componentes. Para montar una cantidad grandes de placas el pedido de componentes sería mucho mayor y el precio por componente aplicado sería menor al aplicarse precios al por mayor.

3.4 Impacto medioambiental.

Todos los componentes utilizados en la construcción de este prototipo, cumplen con la directica de la Unión Europea, 2002/95/CE de Restricción de ciertas Sustancias Peligrosas en aparatos eléctricos y electrónicos, (RoHS del inglés *Restriction of Hazardous Substances*).

De acuerdo con el RD 1 / 2008 del 11 de enero y su modificación en la Ley 6 / 2010 de 24 de marzo, este proyecto no conlleva ningún riesgo para la salud de la población al no producir ningún tipo de emisiones. Por este motivo no está sometido a la declaración de impacto medioambiental.

No obstante, se ha realizado una evaluación del efecto medioambiental que provoca este dispositivo. Esta evaluación se realiza desde tres puntos de vista: su construcción, la explotación durante su vida útil y su desmantelamiento al final de su vida útil.

4. Montaje y Verificación.

4.1 Montaje.

En este proyecto se han realizado 2 prototipos. El primero fue en una placa de cobre de prototipos y las pistas se realizaron con una fresadora.

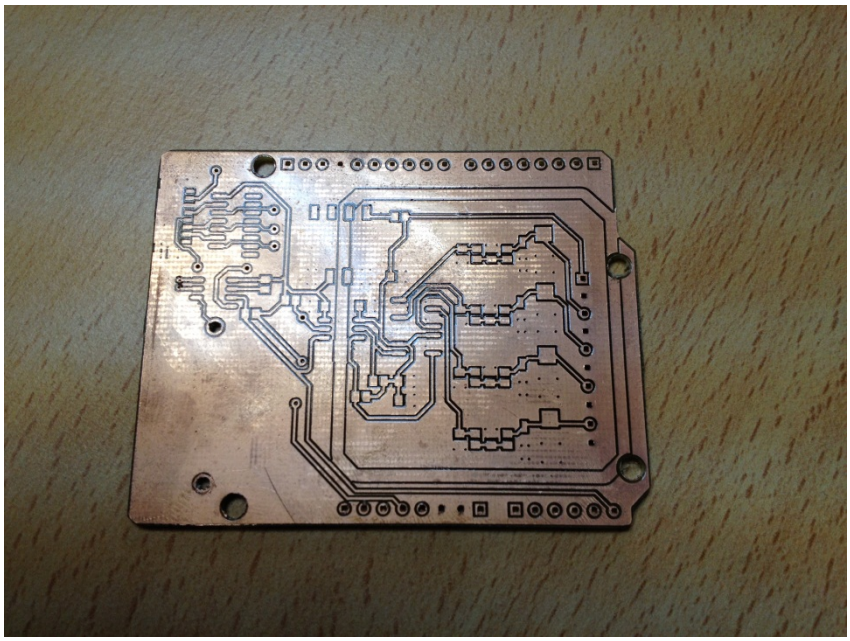


Figura 59: Prototipo en placa de cobre.

Este primer prototipo de la figura 59 tenía la función de comprobar que el montaje de los diferentes bloques diseñados funcionaba correctamente.

En este tipo de prototipos hay que seguir una serie de pasos para soldar correctamente los componentes.

El primer paso es soldar las vías, al ser un prototipo no profesional las caras *Top y Bottom* no tienen conexión entre ellas. Para ellos hay que introducir un pequeño cable y soldarlo a la placa por ambas caras.

El segundo paso es ir soldando los mínimos componentes necesarios para que funcione un bloque. Esto se realiza así para que en caso de que no funcione algo los componentes a desoldar sean los menos posibles.

Para comprobar las conexiones antes de conectar la placa, se debe medir la continuidad de las pistas con el polímetro, de esta forma se evitan los posibles cortocircuitos.

Cuando un bloque está funcionando correctamente se procede a soldar el siguiente, así hasta que todo el escudo este funcionado totalmente.

En este prototipo, como podemos observar en la figura 60, solo se ha realizado el montaje de una de las protecciones del conversor. De esta forma se comprueba que el conversor funciona y se puede montar en el diseño final.

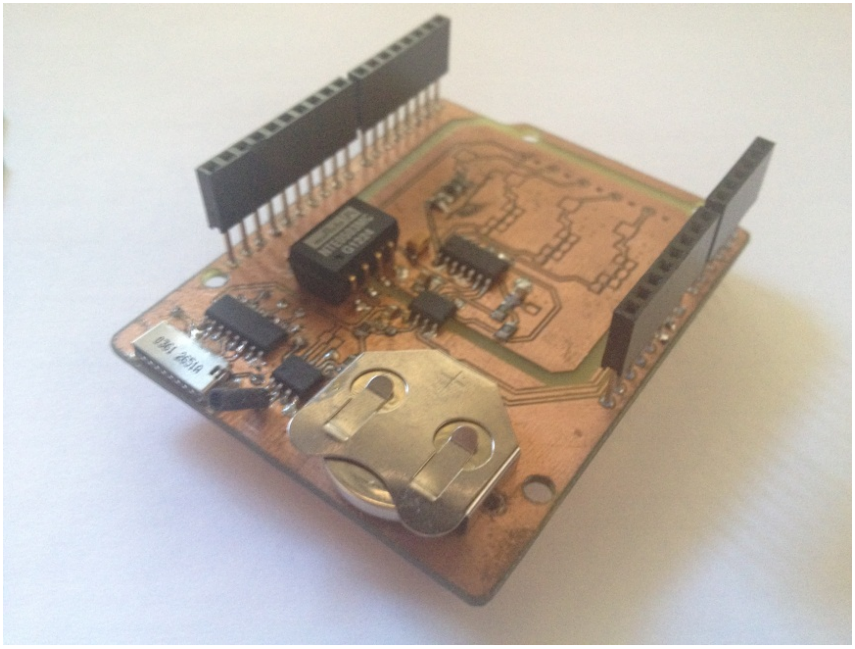


Figura 60: Prototipo montado.

Una vez comprobado que el primer prototipo funcionaba de manera correcta, se enviaron los planos a una empresa especializada para la fabricación de la *PCB* en placa verde, figura 61.

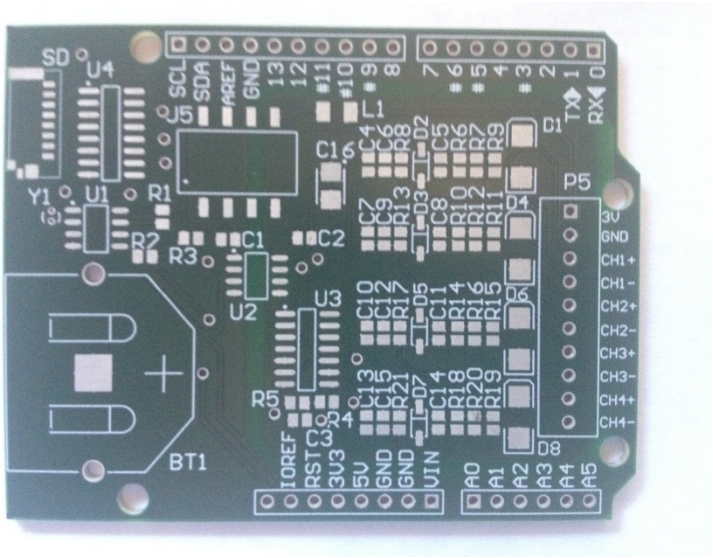


Figura 61: Prototipo en placa verde.

El proceso de montaje de la placa verde es el mismo que el del prototipo. Las únicas diferencias están en que las vías no hacen falta soldarlas, y los *pads* no necesitan de *flux* ya que tienen un acabado profesional que facilita la soldadura. La imagen 62 corresponde al prototipo final montado.

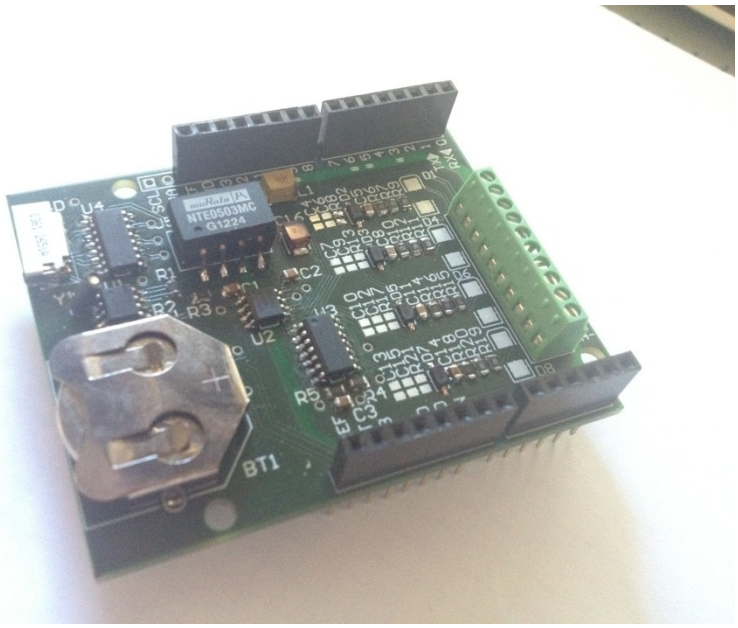


Figura 62: Prototipo en placa verde montada.

4.2 Programación.

4.2.1Funcionalidad básica.

Para comprobar que los componentes funcionan se usan programas ejemplo ya realizados.

Para usar estos programas es necesario el programa de Arduino y conectar el escudo *S-LOGGER* y la placa Arduino UNO al ordenador mediante *USB* como se ve en la figura 63.

Figura 63: Placa Arduino y escudo *S-LOGGER* conectados al ordenador.

Para ser utilizada la placa de Arduino en el ordenador es preciso descargarse de la página web de Arduino los drivers correspondientes a la placa UNO.

El programa necesario para escribir el código y cargarlo en la placa también se debe descargar de la página de Arduino.

Una vez conectada la placa de Arduino junto al escudo *S-LOGGER* y abierto el programa, figura 64, hay que seleccionar el puerto de comunicación *COM* al que están conectados.

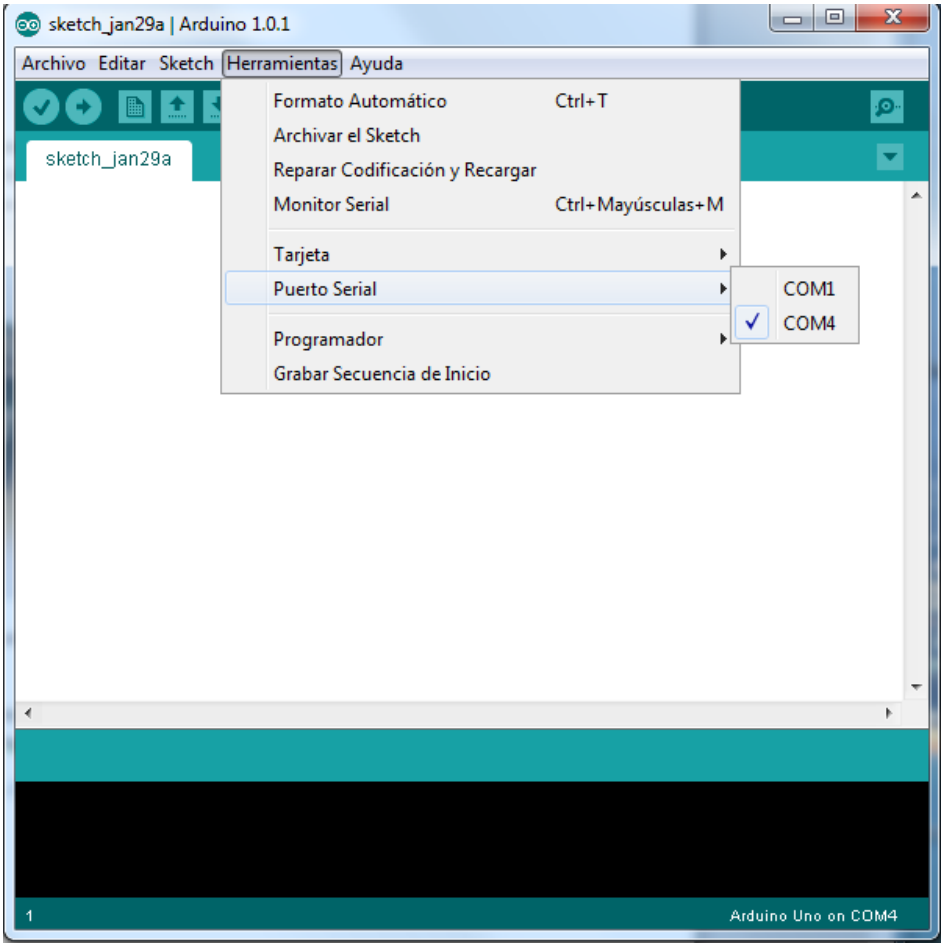


Figura 64: Selección de puerto de comunicación (*COM4*).

4.2.1.1 DS1307.

El programa básico del reloj en tiempo real Ds1307 cosiste en sincronizar la hora del componente con la del ordenador y observar cómo se actualiza correctamente.

Para comprobar que el reloj se actualiza hay que abrir el puerto serie y en caso de que funcione bien aparecerá la hora actualizándose en base al retardo aplicado.

En caso de que no funcione bien, en el monitor serie, aparecerá el texto "*RTC is not running".*

Si el montaje es correcto, en el puerto serie aparecerá la hora actual y se actualizara cada 3 segundos. Como ejemplo tenemos la imagen 65.

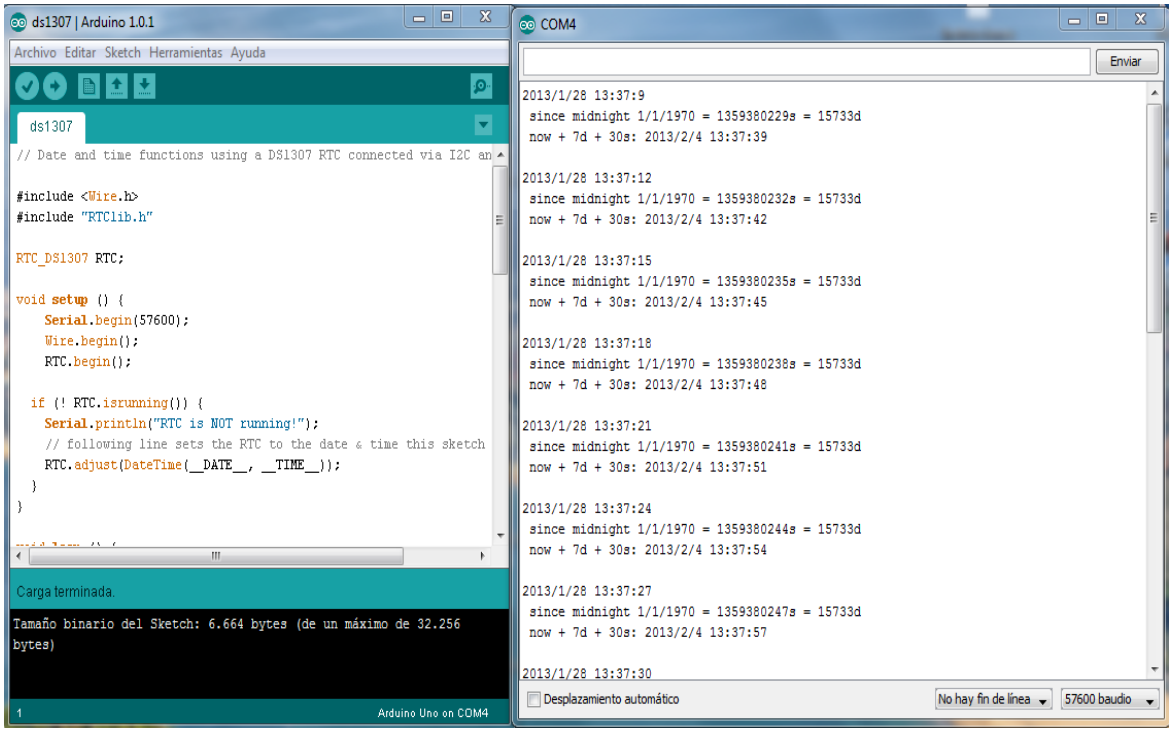


Figura 65: Código del RTC y monitor serie.

Este programa está ya totalmente configurado, no hay que realizar ningún tipo de variación en el código y en el caso de que no funcionara el problema estaría en el *hardware*.

4.2.1.2 Tarjeta SD.

El código básico de la tarjeta SD toma los valores de los canales analógicos de la tarjeta Arduino UNO, no del escudo *S-LOGGER,* y los graba en la tarjeta micro SD.

Este programa vale para diferentes escudos de Arduino con tarjeta SD y cada uno de ellos tiene una configuración diferente, por lo que lo primero que hay que hacer es configurar los pines del programa para que correspondan con nuestro escudo.

Si esta configuración no fuera la correcta o la tarjeta no estuviera conectada, el puerto serial daría este error: "*Card failed, or not present*"

Si el conexionado es correcto y la tarjeta está presente, el programa comenzara a guardar los valores de las tres entradas analógicas en un documento de texto. Insertando la tarjeta en el ordenador y abriendo el archivo se pueden ver los datos guardados de igual forma que en la figura 66.

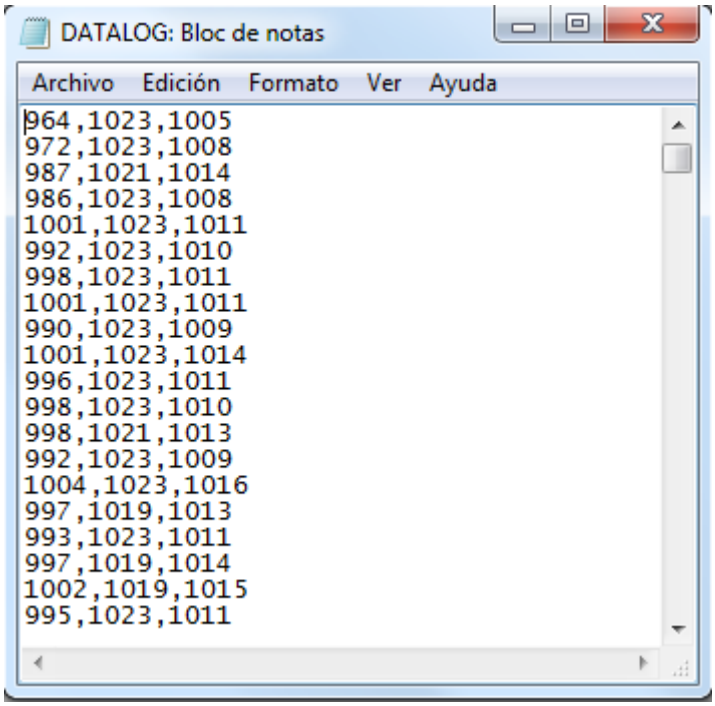


Figura 66: Datos almacenados en la tarjeta SD.

4.2.1.3 Conversor A/D MCP3424.

El programa del conversor A/D configura y recoge los valores de los sensores conectados en sus entradas. Para su correcto funcionamiento hay que realizar las configuraciones correspondientes.

Como se ha especificado en el apartado de diseño, el MCP tiene dos pines que depende de su conexión a masa y Vcc fijan la dirección la cual es 0x64.

Las otras configuraciones posibles del conversor son:

-Ganancia con 4 configuraciones posibles 0-3 que corresponden a ganancia x1, x2, x4 y x8.

-Resolución con 4 configuraciones posibles 0-3 que corresponden a 12bits, 14 bits, 16bits y 18 bits.

Este código, como se puede ver en la figura 67, lee y muestra en el puerto serie los valores de los 2 primeros canales. La tensión obtenida tiene en cuenta el divisor de tensión montado en su entrada.

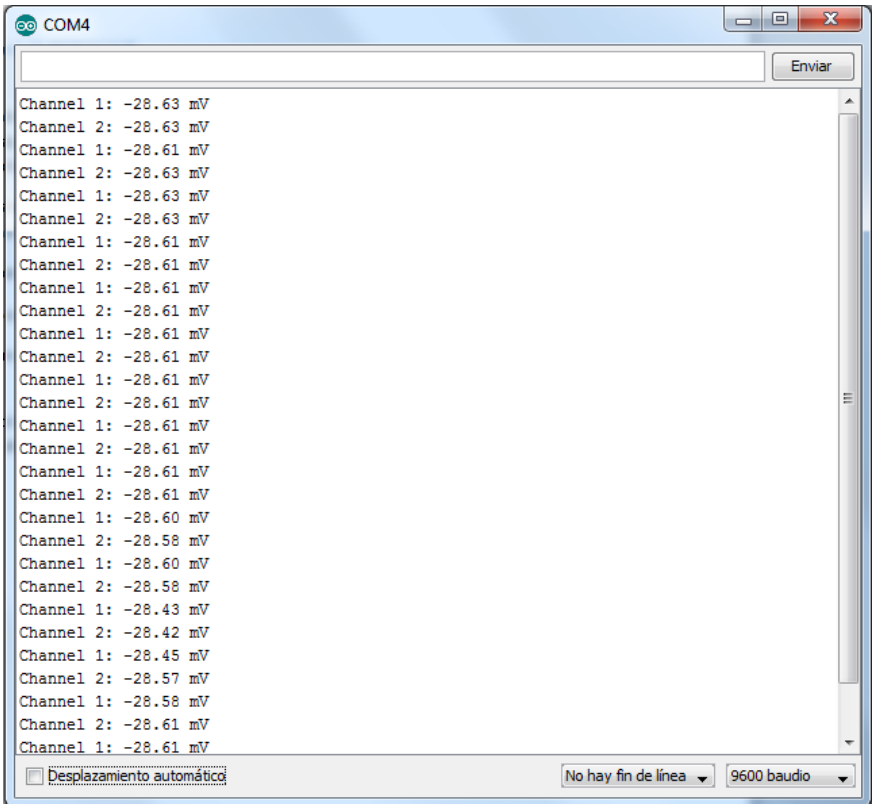


Figura 67: Monitor serie con las muestras del conversor.

4.2.2Ejemplo de *Software.*

El ejemplo de *Software propuesto consiste en agrupar los tres programas anteriores en uno único. De este modo el escudo recogerá los valores de dos sensores los guardará en la SD junto con la información de la hora a la que se han cogido los datos.*

Para guardas los datos se ha utilizado el formato .cvs para que este archivo pueda ser leído por programas de ofimática como Excel en los cuales se pueden procesar estos datos y hacer gráficas.

El flujo grama de la figura 68 muestra la rutina del programa.

En primer lugar se inician todas las variables: El *chip select* de la tarjeta SD, la dirección, ganancia y resolución del conversor AD y el RTC.

A continuación el programa comprueba que el RTC y la tarjeta SD funcionan correctamente. En caso negativo el puerto serie *muestra "Card failed, or not present"* en el caso de la tarjeta SD o *"RTC is NOT running"* en el caso del reloj.

Si alguno de estos dos módulos no funcionase correctamente el programa continuaría funcionando pero sin almacenar los valores en la tarjeta SD, únicamente mostraría los datos tomados por el conversor.

Después de la configuración de los dispositivos y de la comprobación el programa entra en el bucle infinito de toma de datos.

En este bucle primero imprime por el puerto serie la hora, a continuación los valores de los sensores y finalmente guarda todos estos datos en la tarjeta SD.

El último paso del programa es la introducción de un retardo hasta la siguiente lectura, de esta forma se establece el tiempo de muestreo depende de las necesidades.

Este bucle se repetirá mientras la placa de Arduino tenga alimentación, no siendo necesario el ordenador (En el caso de funcionar autónomamente no se visualizaran los datos impresos en el monitor serie).

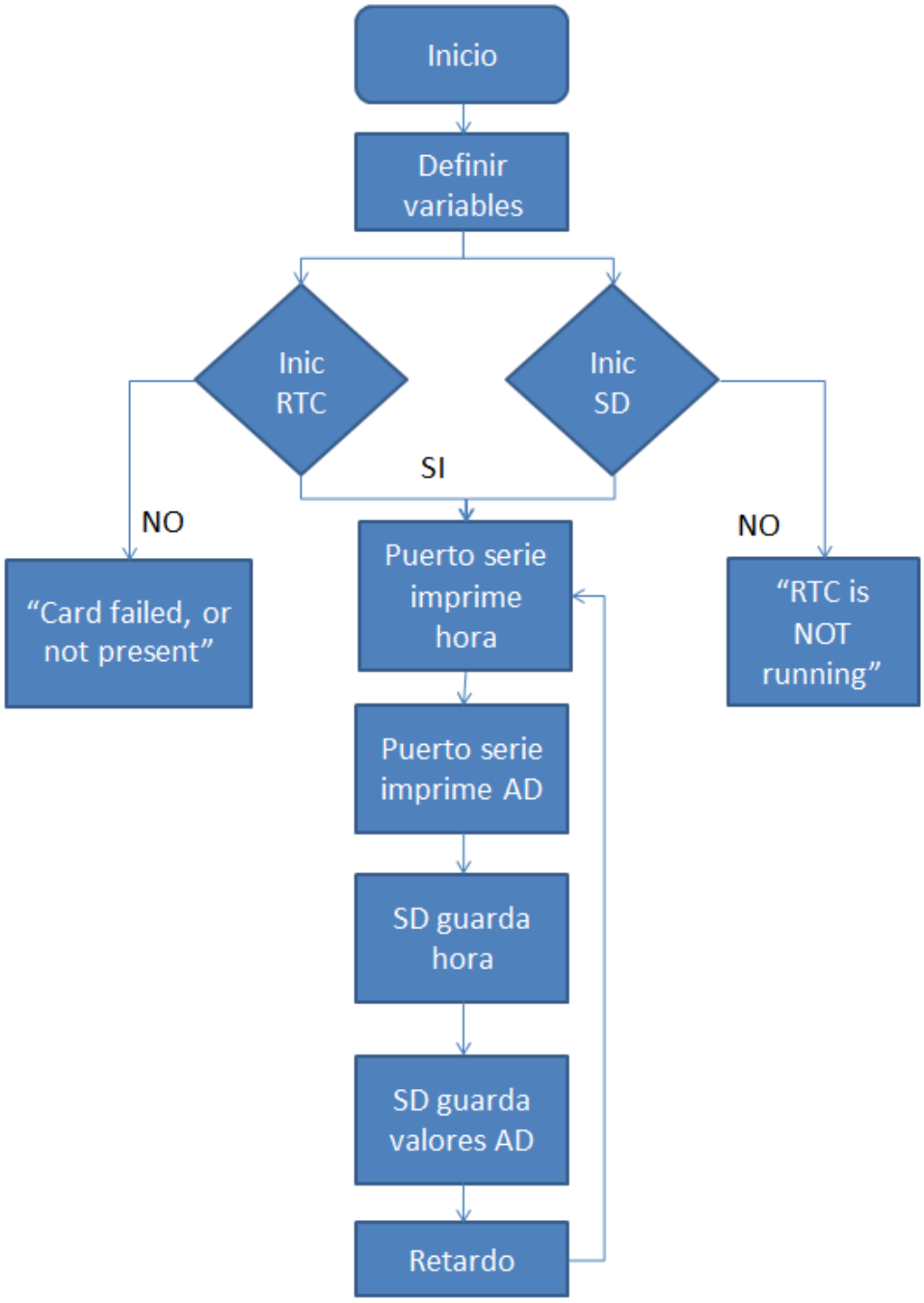


Figura 68: Flujograma del programa propuesto.

En el ejemplo propuesto se ha tomado la temperatura interior y exterior durante varias horas mediante 2 LM35.

El LM35 es un sensor de temperatura en grados centígrados. La medida de tensión que ofrece, en mili Voltios, es directamente la temperatura medida en centígrados.

El tiempo entre muestra y muestra ha sido de 1 minuto y en la figura 69 podemos observar algunas de las medidas en el monitor serie.

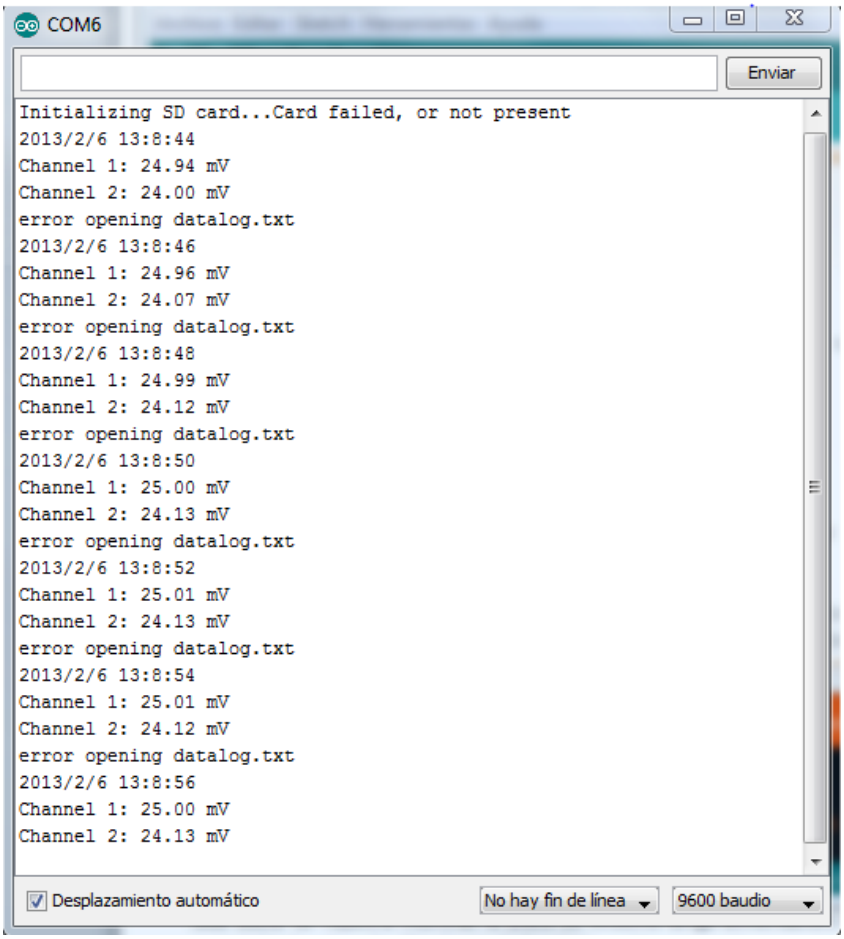


Figura 69: Medidas en el monitor Serie.

Una vez tomados los datos necesarios se extrae la tarjeta micro SD del escudo y se insertar en el ordenador mediante un adaptador.

El archivo guardado es un .DOC, pero al abrirlo con un programa de Excel automáticamente reconoce el formato .CVS y muestra cada dato en su correspondiente celda.

06/02/2013	23:27:59	4,66	19,48
06/02/2013	23:29:00	4,37	19,31
06/02/2013	23:30:01	4,37	19,15
06/02/2013	23:31:02	4,28	19,09
06/02/2013	23:32:03	4,24	19,04
06/02/2013	23:33:04	4,22	19,1
06/02/2013	23:34:05	4,24	19,01
06/02/2013	23:35:06	4,24	19,03
06/02/2013	23:36:07	4,36	18,96
06/02/2013	23:37:08	4,46	18,9
06/02/2013	23:38:09	4,28	18,93
06/02/2013	23:39:10	4,33	18,97
06/02/2013	23:40:12	4,43	19,01
06/02/2013	23:41:13	4,36	18,99
06/02/2013	23:42:14	4,31	19,04
06/02/2013	23:43:15	4,33	18,97
06/02/2013	23:44:16	4,27	19
06/02/2013	23:45:17	4,21	18,99
06/02/2013	23:46:18	4,12	19,01
06/02/2013	23:47:19	4,1	18,88
06/02/2013	23:48:20	3,96	18,97
06/02/2013	23:49:21	4	18,88
06/02/2013	23:50:22	3,99	18,94
06/02/2013	23:51:23	4,09	18,88
06/02/2013	23:52:24	4,06	18,96
06/02/2013	23:53:25	4	18,96
06/02/2013	23:54:26	3,88	18,97
06/02/2013	23:55:27	3,91	19
06/02/2013	23:56:28	3,93	19,04
06/02/2013	23:57:29	3,96	18,99
06/02/2013	23:58:30	3,99	19
06/02/2013	23:59:31	3,93	18,99
07/02/2013	0:00:32	4,07	19,07
07/02/2013	0:01:33	4,21	19,01
07/02/2013	0:02:34	4,27	19,03
07/02/2013	0:03:35	4,21	18,97

Una vez los datos están correctamente organizados en la tabla de Excel ya se pueden realizar todas las funciones deseadas como pueden ser gráficas, medias de datos etc.

A continuación se muestra una gráfica en la figura 70 con los valores de la temperatura interior y exterior.

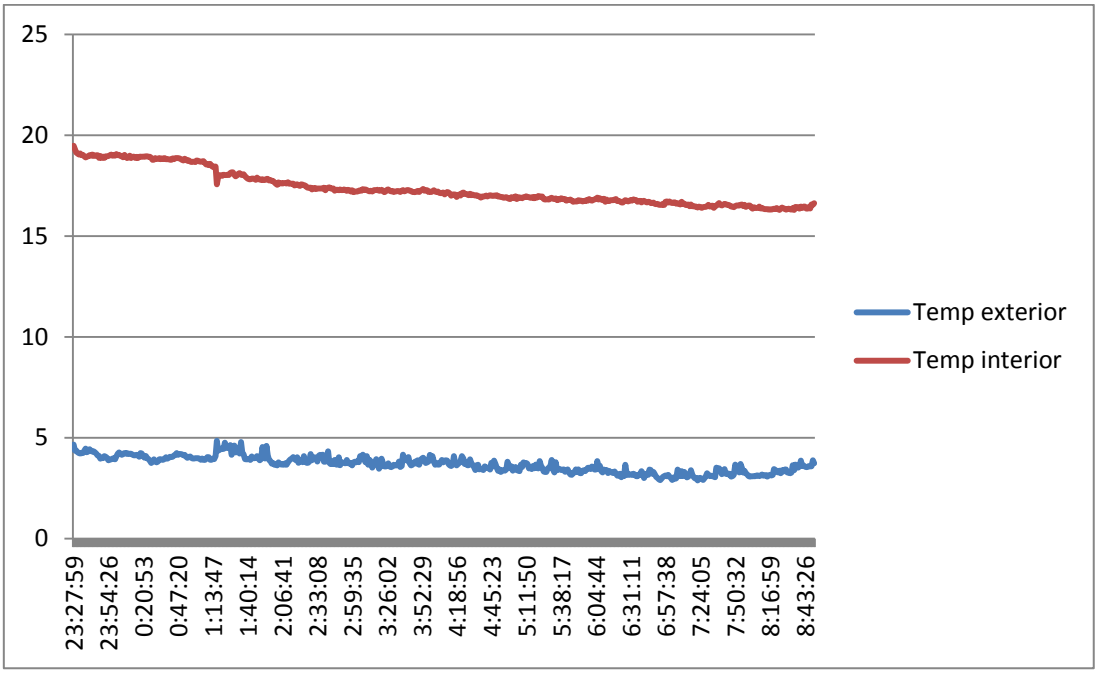


Figura 70: Grafica de temperatura.

5. Conclusiones.

En este proyecto, se ha diseñado e implementando un escudo para una placa de desarrollo Arduino. El objetivo principal de este escudo, que ha sido cubierto, es tener la capacidad para realizar funciones similares a otras placas de adquisición de datos, como puede ser la USB-6008 de *National Instruments*, con un coste contenido. El diseño ha sido implementado y verificado experimentalmente obteniendo unos resultados satisfactorios.

Las principales características reseñables son:

El escudo diseñado dispone de una serie de entradas y salidas en las que se pueden conectar sensores varios. El sistema de adquisición de datos de *National Instruments* dispone de 8 entradas analógicas de 12 bits, nuestro diseño solo dispone de 4, pero de mayor resolución, 18 bits.

El escudo permite la comunicación con el programa *LabView.* Mediante la creación de un parche para los drivers de Arduino en *LabView,* el escudo es totalmente compatible con todas las funciones de dicho software.

Además, el sistema de adquisición de datos de *National Instruments* cuesta alrededor de 150€, el escudo *S-LOGGER* 40€ y la placa Arduino UNO necesaria para su funcionamiento, 20€.

El escudo *S-LOGGER* además incluye más funcionalidades respecto a la USB-6008 tales como:

-Microprocesador programable, en el cual se pueden guardar y ejecutar programas.

-Posibilidad de funcionamiento autónomo gracias a su almacenamiento en una tarjeta micro SD

-En caso de rotura de un componente se puede sustituir fácilmente ya que todos los componentes se pueden comprar por separado.

A continuación se muestra una tabla con una comparativa entre las dos placas de adquisición de datos.

NOMBRE	S-LOGGER	USB-6008
PRECIO	40€ + 20€ = 60€	160 €
ENTRADAS ANALOGICAS	4 x 18 Bits	8 x 12 Bits
SALIDAS ANALOGICAS	6 X 10 Bits	2 x 12 Bits
E/S DIGITALES	14	12
COMPATIBILIDAD CON LABWIEW	SI	SI
PROGRAMABLE	SI	NO
FUNCIONAMIENTO AUTONOMO	SI	NO
GARANTIA	NO	SI
POSIBILIDAD DE SUSTITUIR COMPONENTES	SI	NO
MAXIMA FRECUENCIA DE MUESTREO	120 SPS	10 KS/s
Hardware y Sofware libre	SI	NO

Para finalizar, cabe destacar que todos los diseños y componentes utilizados han sido correctamente documentados para su uso posterior. Además, el proyecto se ha realizado siguiendo el plan de trabajo previamente establecido.

5.1 Líneas de investigaciones futuras.

Este escudo permite varias líneas de investigación futuras.

Una de las principales líneas debe ser la total implementación del conversor del escudo con LabView. Hasta el momento se ha conseguido leer la dirección de I2C del conversor y ver la variación de los valores de la línea de comunicación cuando se varía un potenciómetro.

Para una correcta lectura primero habría que configurar el conversor, ganancia y resolución, lo segundo seria seleccionar mediante el multiplexor interno el canal, o canales, que se desean leer y por ultimo mostrar el dato convertido en la pantalla del ordenador.

En el caso de funcionamiento mediante código las alternativas son muy variadas:

-Mostrar los datos de los valores convertidos en una pantalla conectada a Arduino con comunicación I2c.

-Controlar actuadores a partir de los datos de sensores o el reloj en tiempo real (temporizadores o alarmas). Todos los datos de los sensores y de las acciones realizadas por los actuadores podrían ser guardadas en la tarjeta micro SD.

6. Bibliografía.

A continuación se listan los libros, páginas web, apuntes, así como las hojas de datos de los distintos fabricantes, utilizados en la elaboración de este proyecto tanto referente a la descripción de la memoria, como a la realización del escudo *S-LOGGER*.

Libros:

- [1] Torres, Manuel, Diseño e ingeniería electrónica asistida por ordenador con PROTEL DXP.

Apuntes:

- [2] Bono, Antonio, Microprocesadores EUITIZ, curso 3º Electrónica industrial.
- [3] Asiain, David, Microprocesadores EUPLA, curso 3º Electrónica Industrial.
- [4] Torres, Miguel Ángel, Oficina Técnica, curso 3º Electrónica Industrial.
- [5] Fernández, Vicente, Electrónica analógica, curso 2º Electrónica Industrial.

Páginas WEB.

- [6] http://www.arduino.cc/es/ Consultada el: 30/9/12.
- [7] http://es.rs-online.com/web/ Consultada el: 25/1/13.
- [8]http://www.ardublog.com/lector-de-tarjetas-sd-para-arduino/ Consultada el: 13/7/12.
- [9] https://www.sparkfun.com/ Consultada el: 13/7/12.

- [10] http://spain.ni.com/ (*National Instruments*) Consultada el: 10/12/12.
- [11] http://www.eurocircuits.com/. Consultada el: 1/2/2013.

Hojas de características

- [12] MICROCHIP: http://ww1.microchip.com/downloads/en/DeviceDoc/22088c.pdf
- [13] MAXIM: http://www.farnell.com/datasheets/46359.pdf
- [14] ANALOG DEVICES: http://www.farnell.com/datasheets/1396721.pdf
- [15] MURATA POWER SOLUTIONS: http://www.murata-ps.com/data/power/ncl/kdc_nte.pdf

ANEXO 1:

Programación.

DS1307.

```
// Date and time functions using a DS1307 RTC connected via I2C and Wire
lib


#include <Wire.h>

#include "RTClib.h"


RTC_DS1307 RTC;


void setup () {
   Serial.begin(57600);
   Wire.begin();
   RTC.begin();


  if (! RTC.isrunning()) {
   Serial.println("RTC is NOT running!");
   // following line sets the RTC to the date & time this sketch was compiled
   RTC.adjust(DateTime(__DATE__, __TIME__));
  }
}


void loop () {
   DateTime now = RTC.now();
```

```
    Serial.print(now.year(), DEC);
    Serial.print('/');
    Serial.print(now.month(), DEC);
    Serial.print('/');
    Serial.print(now.day(), DEC);
    Serial.print(' ');
    Serial.print(now.hour(), DEC);
    Serial.print(':');
    Serial.print(now.minute(), DEC);
    Serial.print(':');
    Serial.print(now.second(), DEC);
    Serial.println();

    Serial.print(" since midnight 1/1/1970 = ");
    Serial.print(now.unixtime());
    Serial.print("s = ");
    Serial.print(now.unixtime() / 86400L);
    Serial.println("d");

    // calculate a date which is 7 days and 30 seconds into the future
    DateTime future (now.unixtime() + 7 * 86400L + 30);

    Serial.print(" now + 7d + 30s: ");
```

```
    Serial.print(future.year(), DEC);
    Serial.print('/');
    Serial.print(future.month(), DEC);
    Serial.print('/');
    Serial.print(future.day(), DEC);
    Serial.print(' ');
    Serial.print(future.hour(), DEC);
    Serial.print(':');
    Serial.print(future.minute(), DEC);
    Serial.print(':');
    Serial.print(future.second(), DEC);
    Serial.println();

    Serial.println();
    delay(3000);
}
```

Tarjeta SD.

```
*
 SD card datalogger

This example shows how to log data from three analog sensors
to an SD card using the SD library.

The circuit:
* analog sensors on analog ins 0, 1, and 2
* SD card attached to SPI bus as follows:
** MOSI - pin 11
** MISO - pin 12
** CLK - pin 13
** CS - pin 4

created  24 Nov 2010
modified 9 Apr 2012
by Tom Igoe

This example code is in the public domain.

*/
```

```
#include <SD.h>

// On the Ethernet Shield, CS is pin 4. Note that even if it's not
// used as the CS pin, the hardware CS pin (10 on most Arduino boards,
// 53 on the Mega) must be left as an output or the SD library
// functions will not work.
const int chipSelect = 4;

void setup()
{
 // Open serial communications and wait for port to open:
  Serial.begin(9600);
   while (!Serial) {
    ; // wait for serial port to connect. Needed for Leonardo only
  }


  Serial.print("Initializing SD card...");
  // make sure that the default chip select pin is set to
  // output, even if you don't use it:
  pinMode(10, OUTPUT);

  // see if the card is present and can be initialized:
  if (!SD.begin(chipSelect)) {
```

```
    Serial.println("Card failed, or not present");
    // don't do anything more:
    return;
  }
  Serial.println("card initialized.");
}

void loop()
{
  // make a string for assembling the data to log:
  String dataString = "";

  // read three sensors and append to the string:
  for (int analogPin = 0; analogPin < 3; analogPin++) {
    int sensor = analogRead(analogPin);
    dataString += String(sensor);
    if (analogPin < 2) {
      dataString += ",";
    }
  }

  // open the file. note that only one file can be open at a time,
  // so you have to close this one before opening another.
  File dataFile = SD.open("datalog.txt", FILE_WRITE);
```

```
  // if the file is available, write to it:
  if (dataFile) {
    dataFile.println(dataString);
    dataFile.close();
    // print to the serial port too:
    Serial.println(dataString);
  }
  // if the file isn't open, pop up an error:
  else {
    Serial.println("error opening datalog.txt");
  }
}
```

MPC3424.

```
/*
Example for using the MCP3424 libary for reading values from a certain channel from the ADC
and setting some of the variables in the configuration register

see MCP3424 datasheet for information on paramters and setting register bits

(c) Jeroen Cappaert for Nanosatisfi, August 2012

*/

#include <Wire.h>
#include <MCP3424.h>

// Configuration variables for MCP3424
#define address  0x6A // address of this MCP3424. For setting address see datasheet
byte gain = 0; // gain = x1
byte resolution = 3; // resolution = 18bits, 3SPS
```

```
MCP3424 ADC1(address, gain, resolution); // create MCP3424 instance.




//---------
// SETUP
//---------

void setup()
{
  Serial.begin(9600);
  Wire.begin();
}

//-------
// LOOP
//-------

void loop()
{

  // get channel data with function
  double CH1 = ADC1.getChannelmV(0);
```

```
  double CH2 = ADC1.getChannelmV(1);
  Serial.print("Channel 1: ");Serial.print(CH1);Serial.println(" mV");
  Serial.print("Channel 2: ");Serial.print(CH2);Serial.println(" mV");



  delay(100);
}
```

Programa propuesto.

```
#include <SD.h>

#include <Wire.h>

#include <MCP3424.h>

#include <RTClib.h>


// Configuration variables for MCP3424

#define address  0x6A // address of this MCP3424. For setting address see datasheet

byte gain = 0; // gain = x1

byte resolution = 3; // resolution = 18bits, 3SPS


MCP3424 ADC1(address, gain, resolution); // create MCP3424 instance.


const int chipSelect = 8;


RTC_DS1307 RTC;


//---------

// SETUP

//---------


void setup ()
```

```
{
   Serial.begin(9600);
   Wire.begin();
   RTC.begin();

Serial.print("Initializing SD card...");
  // make sure that the default chip select pin is set to
  // output, even if you don't use it:
  pinMode(10, OUTPUT);

  // see if the card is present and can be initialized:
  if (!SD.begin(chipSelect)) {
    Serial.println("Card failed, or not present");
    // don't do anything more:
    return;
  }
  Serial.println("card initialized.");

 if (! RTC.isrunning()) {
    Serial.println("RTC is NOT running!");
    // following line sets the RTC to the date & time this sketch was compiled
    RTC.adjust(DateTime(__DATE__, __TIME__));
  }
}
```

```
//-------
// LOOP
//-------

void loop()
{
  //print RTC

  DateTime now = RTC.now();

  Serial.print(now.year(), DEC);
  Serial.print('/');
  Serial.print(now.month(), DEC);
  Serial.print('/');
  Serial.print(now.day(), DEC);
  Serial.print(' ');
  Serial.print(now.hour(), DEC);
  Serial.print(':');
  Serial.print(now.minute(), DEC);
  Serial.print(':');
  Serial.print(now.second(), DEC);
```

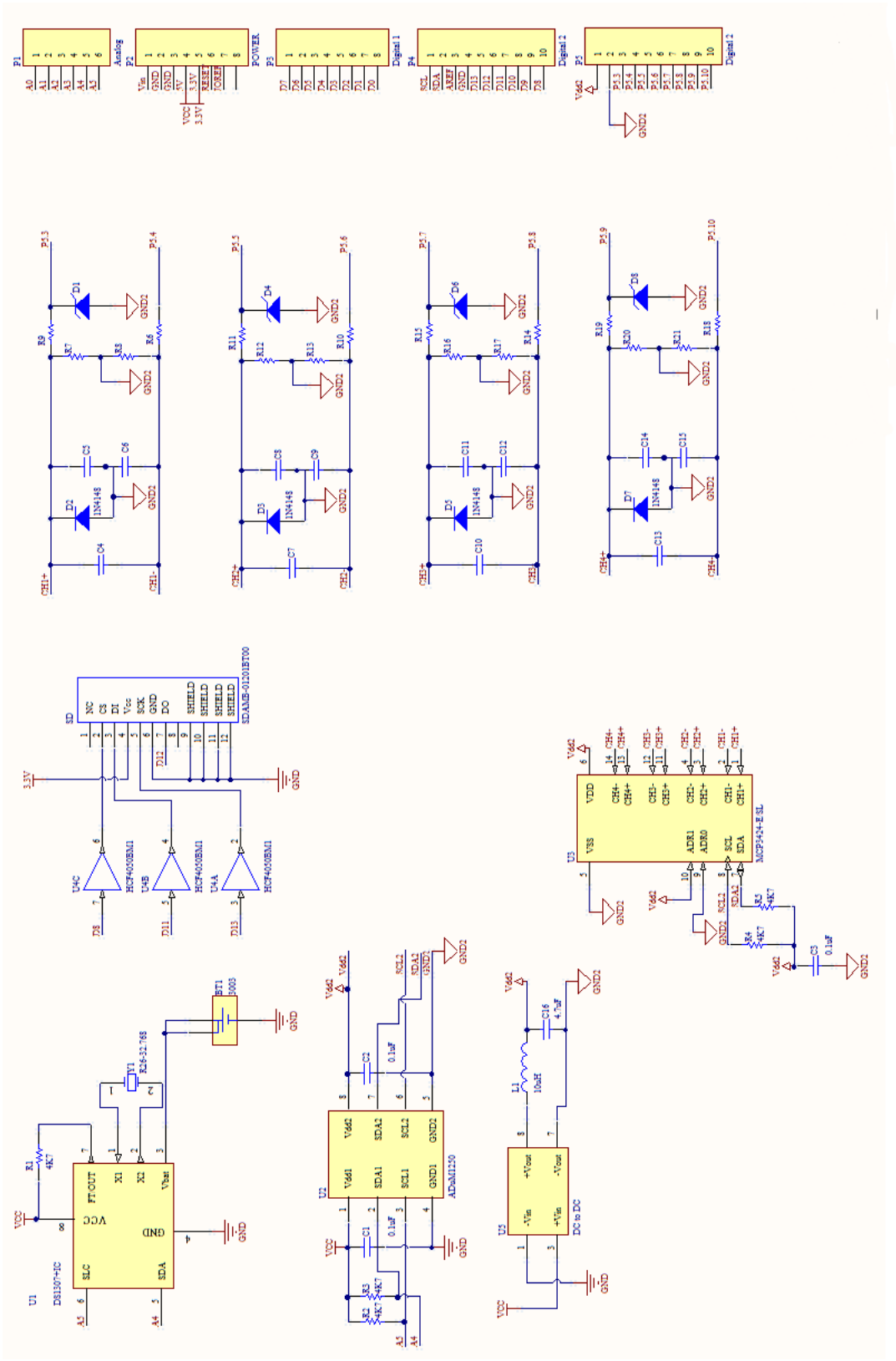
```
 Serial.println();



 // get channel data with function
 double CH1 = ADC1.getChannelmV(0)*9.009;
 double CH2 = ADC1.getChannelmV(1)*9.009;

 Serial.print("Channel 1: ");Serial.print(CH1);Serial.println(" mV");
 Serial.print("Channel 2: ");Serial.print(CH2);Serial.println(" mV");

 delay(1000);




 // open the file. note that only one file can be open at a time,
 // so you have to close this one before opening another.
 File dataFile = SD.open("datalog.txt", FILE_WRITE);

 // if the file is available, write to it:
 if (dataFile) {
     dataFile.print(now.year(), DEC),dataFile.print("/");
     dataFile.print(now.month(), DEC),dataFile.print("/");
     dataFile.print(now.day(), DEC),dataFile.print(",");
```

```
      dataFile.print(now.hour(), DEC),dataFile.print(":");
      dataFile.print(now.minute(), DEC),dataFile.print(":");
      dataFile.print(now.second(), DEC),dataFile.print(",");
      dataFile.print(CH1),dataFile.print(","),dataFile.println(CH2);
   dataFile.close();
    // print to the serial port too:
   // Serial.println(CH1);
  }
  // if the file isn't open, pop up an error:
  else {
   Serial.println("error opening datalog.txt");
  }
}
```

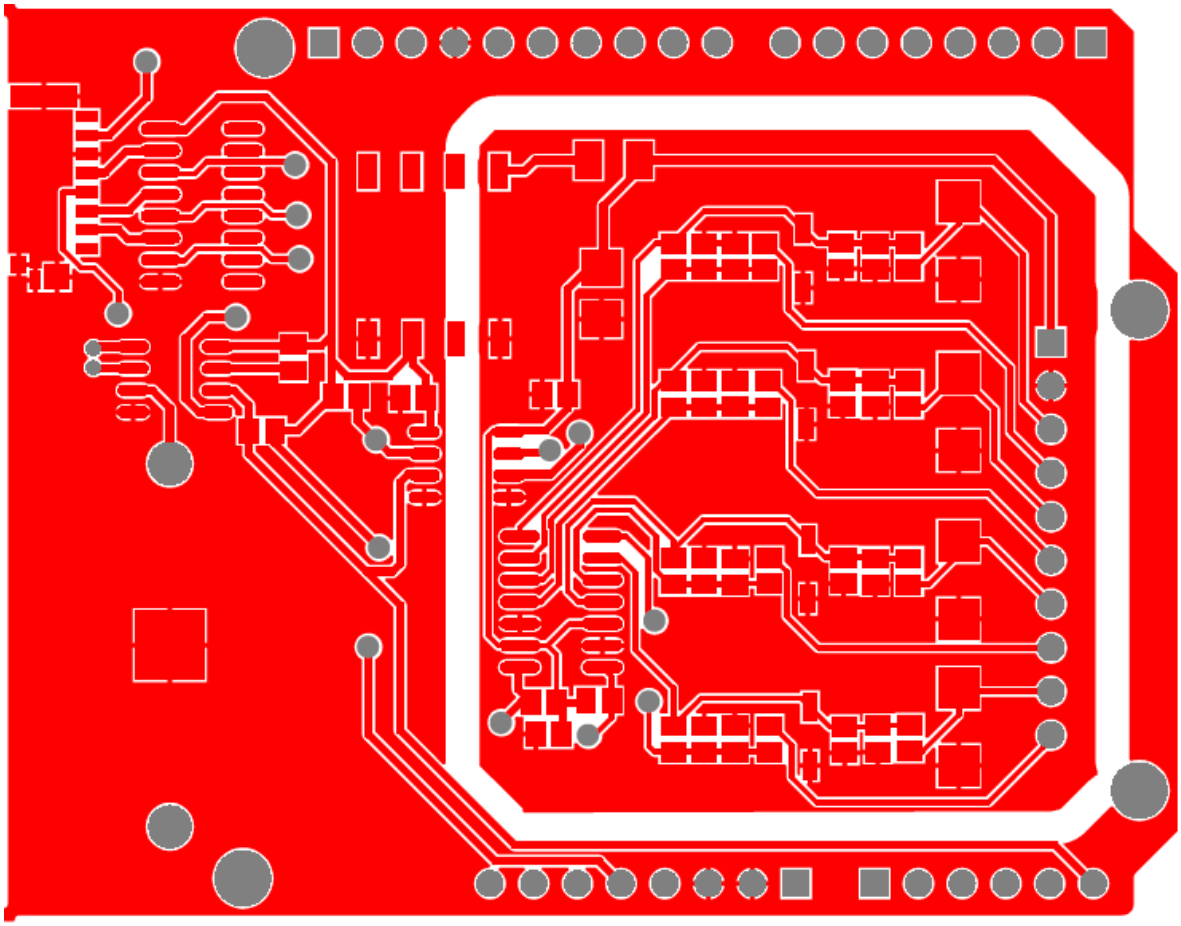
ANEXO 2:

Planos:

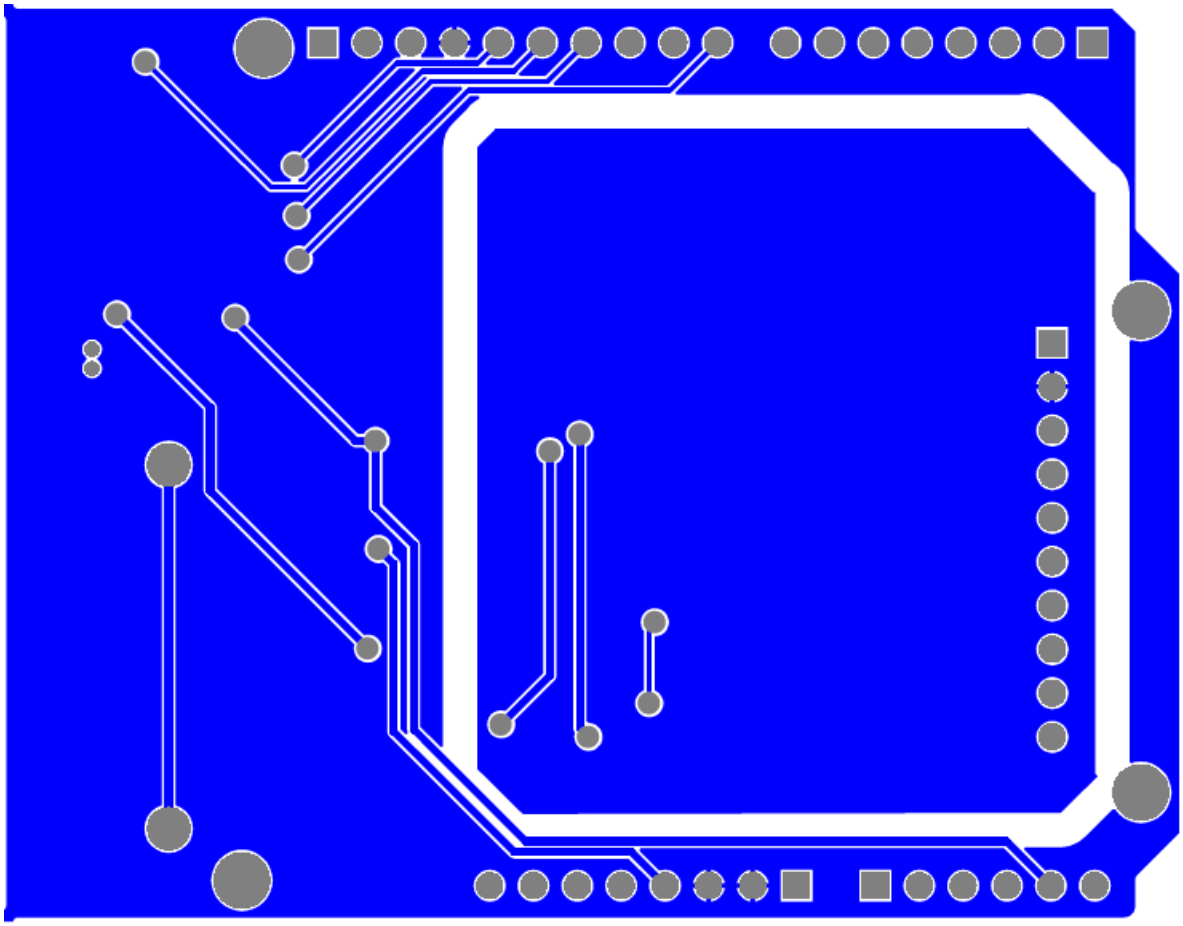
1. Esquema Eléctrico.



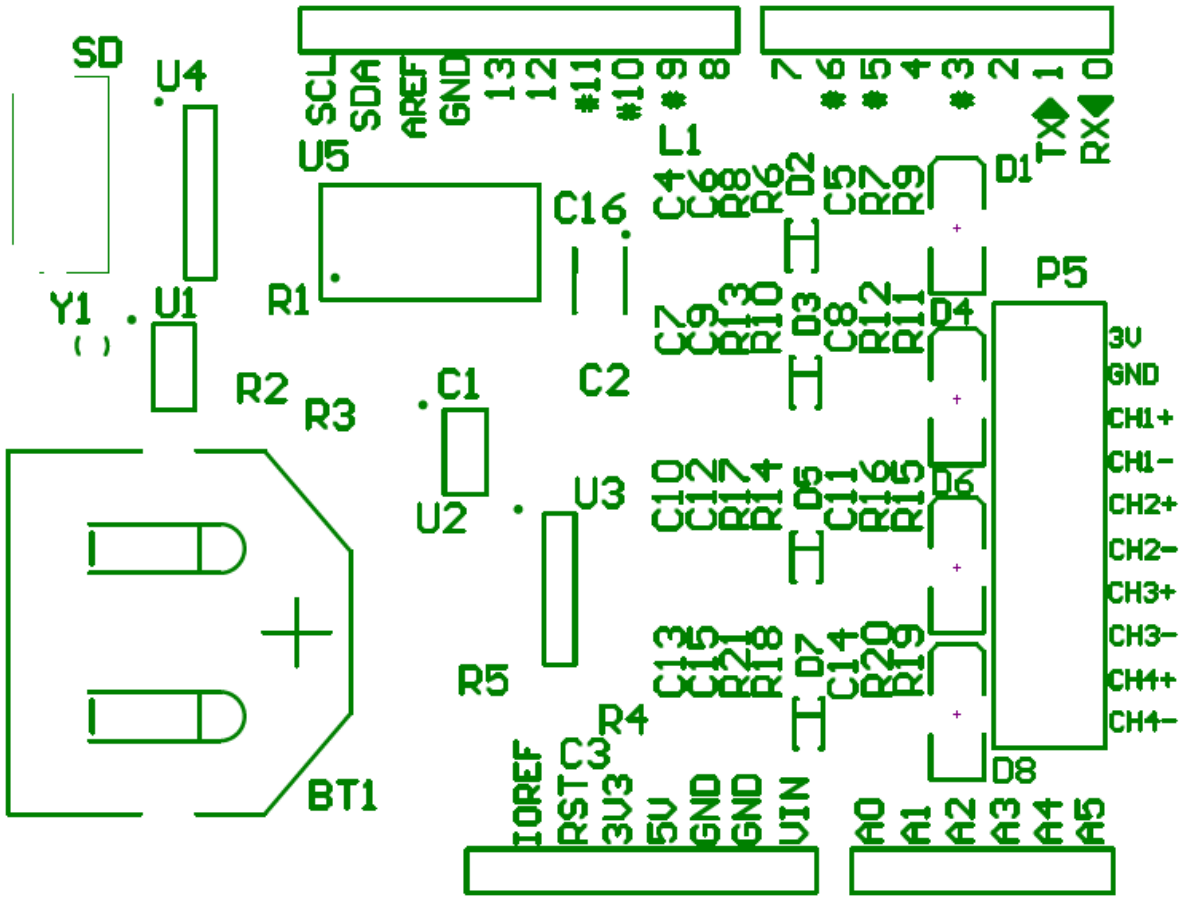
2. Placa de circuito impreso cara top.



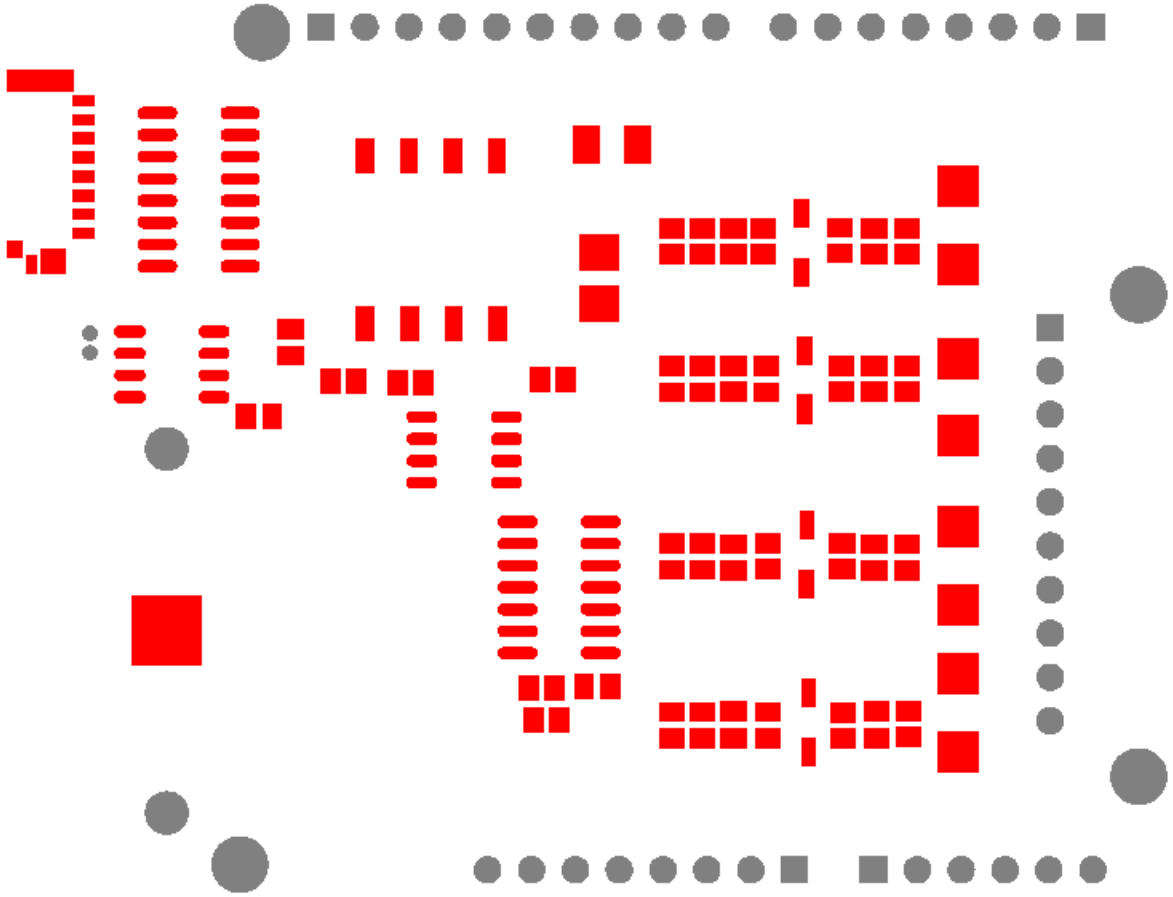
3. Placa de circuito impreso cara bottom.



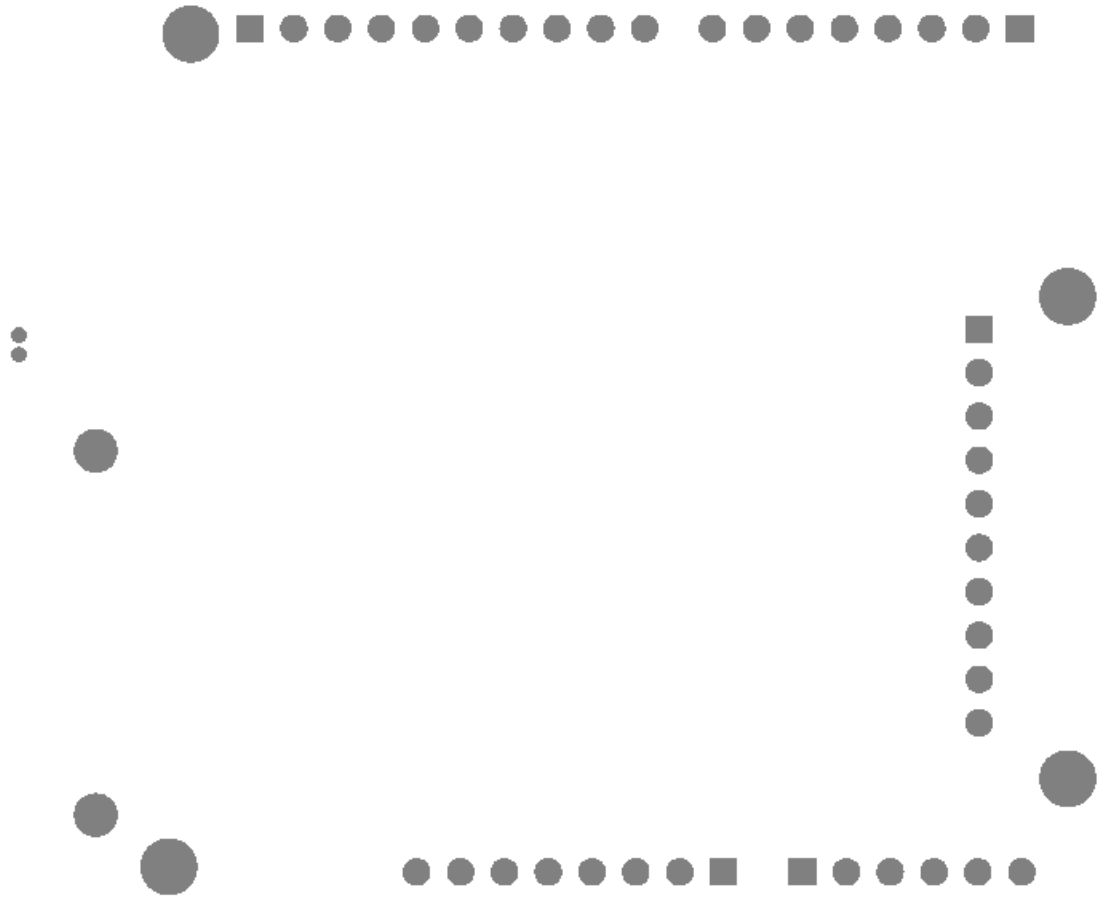
4. Plano de serigrafía.



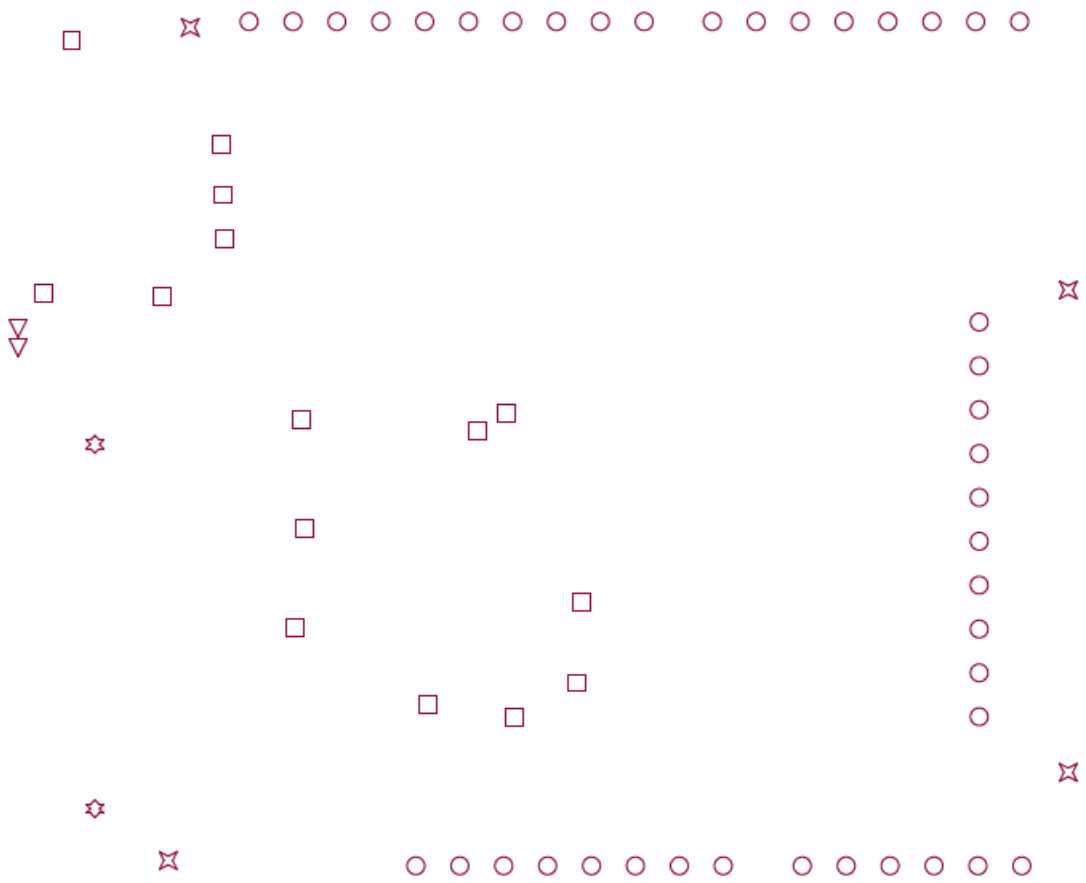
5. Plano de mascarilla cara top.



6. Plano de mascarilla cara bottom.



7. Plano de taladrado.



ANEXO 3

Hojas de características.

DALLAS SEMICONDUCTOR MAXIM

DS1307
64 x 8, Serial, I²C Real-Time Clock

GENERAL DESCRIPTION

The DS1307 serial real-time clock (RTC) is a low-power, full binary-coded decimal (BCD) clock/calendar plus 56 bytes of NV SRAM. Address and data are transferred serially through an I²C*, bidirectional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with AM/PM indicator. The DS1307 has a built-in power-sense circuit that detects power failures and automatically switches to the battery supply.

** I²C is a trademark of Philips Corp. Purchase of I²C components of Maxim Integrated Products, Inc., or one of its sublicensed Associated Companies, conveys a license under the Philips I²C Patent Rights to use these components in an I²C system, provided that the system conforms to the I²C Standard Specification as defined by Philips Corp.*

FEATURES

- Real-Time Clock (RTC) Counts Seconds, Minutes, Hours, Date of the Month, Month, Day of the week, and Year with Leap-Year Compensation Valid Up to 2100
- 56-Byte, Battery-Backed, Nonvolatile (NV) RAM for Data Storage
- I²C Serial Interface
- Programmable Square-Wave Output Signal
- Automatic Power-Fail Detect and Switch Circuitry
- Consumes Less than 500nA in Battery-Backup Mode with Oscillator Running
- Optional Industrial Temperature Range: -40°C to +85°C
- Available in 8-Pin Plastic DIP or SO
- Underwriters Laboratory (UL) Recognized

Typical Operating Circuit and Pin Configurations appear at end of data sheet.

ORDERING INFORMATION

PART	TEMP RANGE	VOLTAGE (V)	PIN-PACKAGE	TOP MARK*
DS1307	0°C to +70°C	5.0	8 PDIP (300 mils)	DS1307
DS1307+	0°C to +70°C	5.0	8 PDIP (300 mils)	DS1307
DS1307N	-40°C to +85°C	5.0	8 PDIP (300 mils)	DS1307
DS1307Z	0°C to +70°C	5.0	8 SO (150 mils)	DS1307
DS1307Z+	0°C to +70°C	5.0	8 SO (150 mils)	DS1307
DS1307ZN	-40°C to +85°C	5.0	8 SO (150 mils)	DS1307N
DS1307ZN+	-40°C to +85°C	5.0	8 SO (150 mils)	DS1307N
DS1307Z/T&R	0°C to +70°C	5.0	8 SO (150 mils) Tape and Reel	DS1307
DS1307Z+T&R	0°C to +70°C	5.0	8 SO (150 mils) Tape and Reel	DS1307
DS1307ZN/T&R	-40°C to +85°C	5.0	8 SO (150 mils) Tape and Reel	DS1307N
DS1307ZN+T&R	-40°C to +85°C	5.0	8 SO (150 mils) Tape and Reel	DS1307N

+ Denotes a lead-free/RoHS-compliant device.
** A "+" anywhere on the top mark indicates a lead-free device. An "N" on the lower left corner of the top mark indicates an industrial temperature grade device.*

Note: *Some revisions of this device may incorporate deviations from published specifications known as errata. Multiple revisions of any device may be simultaneously available through various sales channels. For information about device errata, click here: www.maxim-ic.com/errata.*

ANALOG DEVICES

Hot Swappable, Dual I²C Isolators

ADuM1250/ADuM1251

FEATURES

- Bidirectional I²C communication
- Open-drain interfaces
- Suitable for hot swap applications
- 30 mA current sink capability
- 1000 kHz operation
- 3.0 V to 5.5 V supply/logic levels
- 8-lead SOIC RoHS-compliant package
- High temperature operation: 125°C
- Qualified for automotive applications
- Safety and regulatory approvals
  - UL recognition
    - 2500 V rms for 1 minute per UL 1577
  - CSA Component Acceptance Notice #5A
  - VDE certificate of conformity
    - DIN V VDE V 0884-10 (VDE V 0884-10):2006-12
    - V_{IORM} = 560 V peak

APPLICATIONS

- Isolated I²C, SMBus, or PMBus interfaces
- Multilevel I²C interfaces
- Power supplies
- Networking
- Power-over-Ethernet
- Hybrid electric vehicle battery management

FUNCTIONAL BLOCK DIAGRAMS

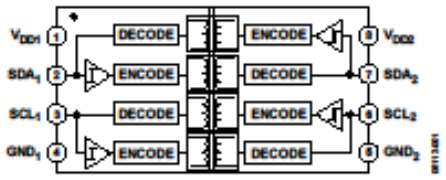


Figure 1. ADuM1250

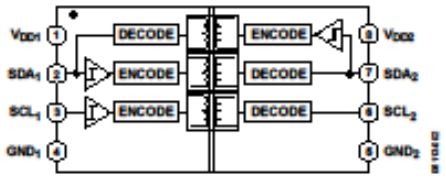


Figure 2. ADuM1251

GENERAL DESCRIPTION

The ADuM1250/ADuM1251[1] are hot swappable digital isolators with nonlatching, bidirectional communication channels compatible with I²C* interfaces. This eliminates the need for splitting I²C signals into separate transmit and receive signals for use with standalone optocouplers.

The ADuM1250 provides two bidirectional channels, supporting a complete isolated I²C interface. The ADuM1251 provides one bidirectional channel and one unidirectional channel for those applications where a bidirectional clock is not required.

Both the ADuM1250 and ADuM1251 contain hot swap circuitry to prevent glitching data when an unpowered card is inserted onto an active bus.

These isolators are based on *i*Coupler® chip scale transformer technology from Analog Devices, Inc. *i*Coupler is a magnetic isolation technology with functional, performance, size, and power consumption advantages as compared to optocouplers. With the ADuM1250/ADuM1251, *i*Coupler channels can be integrated with semiconductor circuitry, which enables a complete isolated I²C interface to be implemented in a small form factor.

[1] Protected by U.S. Patents 5,952,849; 6,873,065; and 7,075,329.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781.329.4700 www.analog.com
Fax: 781.461.3113

Murata Power Solutions

NTE Series

Isolated 1W Single Output SM DC/DC Converters

FEATURES

- RoHS compliant
- Lead frame technology
- Single isolated output
- 1kVDC Isolation
- Efficiency up to 78%
- Power density 1.8W/cm³
- Wide temperature performance at full 1 Watt load, –40°C to 85°C
- UL 94V-0 Package material
- Footprint over pins 1.37cm²
- 3.3V, 5V & 12V Input
- 3.3V, 5V, 9V, 12V & 15V output
- No heatsink required
- Internal SMD construction
- Toroidal magnetics
- MTTF up to 6.8 million hours
- Custom solutions available
- Multi-layer ceramic capacitors

PRODUCT OVERVIEW

The NTE series of miniature surface mounted DC/DC Converters employ leadframe technology and transfer moulding techniques to bring all of the benefits of IC style packaging to hybrid circuitry. The co-planarity of the pin positions is based upon IEC 191-6:1990. The devices are suitable for all applications where high volume production is envisaged.

SELECTION GUIDE

Order Code[1]	Nominal Input Voltage	Output Voltage	Output Current	Input Current at Rated Load	Efficiency	Isolation Capacitance	MTTF[2]
	V	V	mA	mA	%	pF	kHrs
NTE0303MC	3.3	3.3	303	410	73	30	5348
NTE0305MC	3.3	5	200	390	78	35	3847
NTE0309MC	3.3	9	111	400	77	31	3134
NTE0312MC	3.3	12	83	400	77	28	3473
NTE0315MC	3.3	15	66	400	77	29	2473
NTE0503MC	5	3.3	303	270	74	40	5515
NTE0505MC	5	5	200	294	68	35	6857
NTE0505MEC	5	5	200	260	77	40	3933
NTE0506MC	5	6	167	278	72	39	6677
NTE0509MC	5	9	111	267	75	43	5501
NTE0512MC	5	12	83	260	77	42	3957
NTE0515MC	5	15	66	256	78	44	2747
NTE1205MC	12	5	200	124	67	47	4683
NTE1209MC	12	9	111	114	73	77	4008
NTE1212MC	12	12	83	113	74	88	3121
NTE1215MC	12	15	66	111	75	95	2316

INPUT CHARACTERISTICS

Parameter	Conditions	Min.	Typ.	Max.	Units
Voltage range	Continuous operation, 3.3V input types	2.97	3.3	3.63	V
	Continuous operation, 5V input types	4.5	5.0	5.5	
	Continuous operation, 12V input types	10.8	12.0	13.2	
Reflected ripple current			30	47	mA p-p

ISOLATION CHARACTERISTICS

Parameter	Conditions	Min.	Typ.	Max.	Units
Isolation voltage	Flash tested for 1 second	1000			VDC
Resistance	Viso= 1000VDC	10			GΩ

GENERAL CHARACTERISTICS

Parameter	Conditions	Min.	Typ.	Max.	Units
Switching frequency	All output types		110		kHz

ABSOLUTE MAXIMUM RATINGS

Lead temperature 1.5mm from case for 10 seconds	300°C
Internal power dissipation	600mW
Input voltage Vin, NTE03 types	5.5V
Input voltage Vin, NTE05 types	7V
Input voltage Vin, NTE12 types	15V

1. If components are required in tape and reel format suffix order code with -R, e.g. NTE0505MC-R.
2. Calculated using MIL-HDBK-217 FN2 calculation model with nominal input voltage at full load.

All specifications typical at TA=25°C, nominal input voltage and rated output current unless otherwise specified.

For full details go to www.murata-ps.com/rohs

MCP3422/3/4

18-Bit, Multi-Channel ΔΣ Analog-to-Digital Converter with I²C™ Interface and On-Board Reference

Features

- 18-bit ΔΣ ADC with Differential Inputs:
  - 2 channels: MCP3422 and MCP3423
  - 4 channels: MCP3424
- Differential Input Full Scale Range: $-V_{REF}$ to $+V_{REF}$
- Self Calibration of Internal Offset and Gain per Each Conversion
- On-Board Voltage Reference (V_{REF}):
  - Accuracy: 2.048V ± 0.05%
  - Drift: 15 ppm/°C
- On-Board Programmable Gain Amplifier (PGA):
  - Gains of 1, 2, 4 or 8
- INL: 10 ppm of Full Scale Range
- Programmable Data Rate Options:
  - 3.75 SPS (18 bits)
  - 15 SPS (16 bits)
  - 60 SPS (14 bits)
  - 240 SPS (12 bits)
- One-Shot or Continuous Conversion Options
- Low Current Consumption:
  - 135 µA typical (V_{DD}= 3V, Continuous Conversion)
  - 36 µA typical (V_{DD}= 3V, One-Shot Conversion with 1 SPS)
- On-Board Oscillator
- I²C™ Interface:
  - Standard, Fast and High Speed Modes
  - User configurable two external address pins for MCP3423 and MCP3424
- Single Supply Operation: 2.7V to 5.5V
- Extended Temperature Range: -40°C to +125°C

Typical Applications

- Portable Instrumentation and Consumer Goods
- Temperature Sensing with RTD, Thermistor, and Thermocouple
- Bridge Sensing for Pressure, Strain, and Force
- Weigh Scales
- Battery Fuel Gauges
- Factory Automation Equipment

Description

The MCP3422, MCP3423 and MCP3424 devices (MCP3422/3/4) are the low noise and high accuracy 18-Bit delta-sigma analog-to-digital (ΔΣ A/D) converter family members of the MCP342X series from Microchip Technology Inc. These devices can convert analog inputs to digital codes with up to 18 bits of resolution.

The on-board 2.048V reference voltage enables an input range of ± 2.048V differentially (full scale range = 4.096V/PGA).

These devices can output analog-to-digital conversion results at rates of 3.75, 15, 60, or 240 samples per second depending on the user controllable configuration bit settings using the two-wire I²C serial interface. During each conversion, the device calibrates offset and gain errors automatically. This provides accurate conversion results from conversion to conversion over variations in temperature and power supply fluctuation.

The user can select the PGA gain of x1, x2, x4, or x8 before the analog-to-digital conversion takes place. This allows the MCP3422/3/4 devices to convert a very weak input signal with high resolution.

The MCP3422/3/4 devices have two conversion modes: (a) One-Shot Conversion mode and (b) Continuous Conversion mode. In One-Shot conversion mode, the device performs a single conversion and enters a low current standby mode automatically until it receives another conversion command. This reduces current consumption greatly during idle periods. In Continuous conversion mode, the conversion takes place continuously at the set conversion speed. The device updates its output buffer with the most recent conversion data.

The devices operate from a single 2.7V to 5.5V power supply and have a two-wire I²C compatible serial interface for a standard (100 kHz), fast (400 kHz), or high-speed (3.4 MHz) mode.

The I²C address bits for the MCP3423 and MCP3424 are selected by using two external I²C address selection pins (Adr0 and Adr1). The user can configure the device to one of eight available addresses by connecting these two address selection pins to V_{DD}, V_{SS} or float. The I²C address bits of the MCP3422 are programmed at the factory during production.