_ TELECOM ESCUELA TÉCNICA VLC SUPERIOR DE UPV INGENIEROS DE TELECOMUNICACIÓN

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingenieros de Telecomunicación
Universitat Politècnica de València

Diseño y realización de un sistema de adquisición y tratamiento de datos para la monitorización de módulos fotovoltaicas

TRABAJO FIN DE GRADO

Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación

Autor: Díaz Romero, Enrique

Tutor: Sanchis Kilders, Pablo
Ponce Alcantara, Salvador

Curso 2015-2016

Resumen

En el presente trabajo se aborda el desarrollo de un sistema de medida para caracterización de múltiples módulos solares, en escenarios reales. Se ha diseñado y fabricado un multiplexor de 8 canales para medir a 4 hilos las placas solares mediante el SourceMeter Keithley 2651A. El multiplexor se ha desarrollado como solución embebida controlable de forma remota mediante puerto serie gracias al micro controlador Atmega32U4. Este, gracias a que es compatible con el bootloader de Arduino, permite utilizar su lenguaje de programación y hacer la carga del firmware mediante el puerto USB y su IDE (Entorno de desarrollo integrado). Todo el software de control esta desarrollado mediante LabVIEW que nos permite la realización de sistemas de instrumentación automatizada e integrarlo con el control del medidor para poder realizar medidas rápidas y sincronizadas.

Palabras clave: Modulo Solar, Desarrollo Hardware, Relé, Multiplexor, Curva V-I, Atmega32u4

Resum

En el present treball s'aborda el desenvolupament d'un sistema de mesura per a caracterització de múltiples mòduls solars, en escenaris reals. S'ha dissenyat i fabricat un multiplexor de 8 canals per mesurar a 4 fils les plaques solars mitjançant el SourceMeter Keithley 2651A. El multiplexor s'ha desenvolupat com a solució embeguda controlable de forma remota mitjançant port sèrie gràcies al microcontrolador Atmega32U4. Aquest, gràcies al fet que és compatible amb el bootloader d'Arduino, permet utilitzar el seu llenguatge de programació i fer la càrrega de firmware mitjançant el port USB i el seu IDE (Entorn de desenvolupament integrat). Tot el software de control aquesta desenvolupat mitjançant LabVIEW que ens permet la realització de sistemes d'instrumentació automatitzada i integrar-se amb el control del mesurador per poder realitzar mesures ràpides i sincronitzades

Paraules clau: Modul Solar,Desenvolupament Hardware, Rele, Multiplexor, Curve V-I,Atemega32u4

Abstract

In this TFG is implemented a measurement system, which is made for characterization of multiples solar modules in real scenarios. That system was designed and manufactured as a 8-channels multiplexer to use with the Keithley 2651A SourceMeter. It can do 4 wire measurements to the solar modules. The multiplexer has been developed as embedded solution, which is controllable remotely via serial port through to the micro controller Atmega32U4. Which is compatible with the Arduino bootloader ,then can be used its programming language and the Arduino IDE (Integrated Development Environment) to load the firmware by USB. All the control software is developed using LabVIEW, that allows the realization of automated instrumentation systems and integrate it with the control meter to perform quick and synchronized measures.

Key words: Solar Module, Hardware Desing, Relay, Multiplexer, V-I Curve, Atmega32u4

Índice general

Índice de figuras

CAPÍTULO 1
Introducción

1.1 Objetivos

El objeto de este proyecto es el diseño hardware e implementación software de un sistema de multiplexado automático para la medida de curvas I-V de módulos fotovoltaicos.

Este proyecto se lleva a cabo en el Centro de Tecnología Nanofotónica (NTC) y se integra en las investigaciones que el NTC realiza para la mejora de la eficiencia de los módulos solares.

Con la realización de este proyecto, se desea adaptar y actualizar el sistema de medida I-V actual del que dispone el laboratorio de fotovoltaica del centro, y dotarlo de la capacidad de medir y caracterizar no solo las curvas I-V de una célula solar, sino de varias de estas en escenarios reales.

Todo el sistema debe ser ampliable y reconfigurable para poder adaptarlo al número de placas deseado y permitir su programación para hacer medidas a largo plazo.

Este trabajo sirve como introducción a una gestión de proyectos a largo plazo, ya que implica numerosos plazos que hay que cumplir y requisitos previos necesarios.

1.2 Motivación

La medida de curvas I-V es una de las técnicas mas utilizadas a la hora de caracterizar dispositivos fotovoltaicos, y es considerada como una fuente muy útil de información acerca del comportamiento eléctrico de los dispositivos.

En términos generales, esta técnica proporciona:

- Desde el punto de vista del diseño, el conocimiento de las curvas I-V y PV permite la implantación de aplicaciones fotovoltaicas más eficientes.
- En la industria fotovoltaica, la caracterización resulta crucial para el desarrollo de los procesos de producción.
- Como herramienta de garantía de calidad, permite la determinación de la potencia máxima en virtud de condiciones reales de funcionamiento, las cuales pueden extrapolarse a condiciones estándar de medida (STC). Por tanto, puede realizarse la comparación entre la potencia nominal instalada y la acordada entre el proveedor y el cliente, lo cual aumenta la credibilidad del sector fotovoltaico.

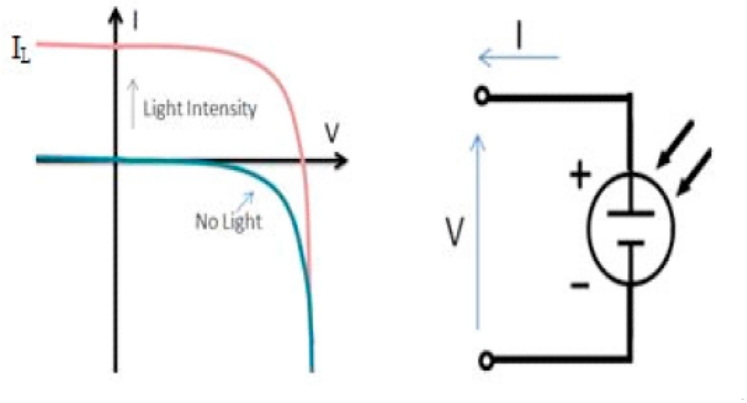


Figura 1.1: Curvas I-V de un dispositivo fotovoltaico y se esquema eléctrico equivalente

- Para los fabricantes, las sucesivas evaluaciones se convierten en puntos de referencia de la calidad del producto final fabricado. Debido a la dispersión de los parámetros eléctricos en los procesos de fabricación de módulos, uno de los problemas actuales es conocer la verdadera potencia de pico instalada en plantas fotovoltaicas.

- Como herramienta de diagnóstico y de mantenimiento, permite la detección de anomalías (sombreado, ramas desconectadas, células rotas, o diodos defectuosos), junto con la cuantificación de la degradación de la potencia real a lo largo del tiempo, comparando los resultados de medidas anteriores.

- Para la investigación y el desarrollo de equipos, es un indicador clave que ayuda a identificar las necesidades futuras, o para orientar futuras decisiones, facilitando la comparación de los sistemas que difieren en términos de diseño, tecnología o ubicación geográfica, y la validación de modelos de estimación de rendimiento durante la fase de diseño.

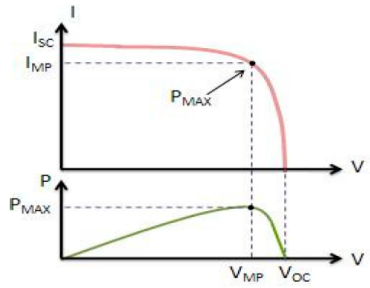
1.2.1. Curva I-V de un dispositivo fotovoltaico

Una célula fotovoltaica se puede modelar como una fuente de corriente en paralelo con un diodo. En ausencia de luz, no se genera ninguna corriente y la célula se comporta como un diodo normal. A medida que aumenta la luz incidente, la célula empieza a generar corriente como se puede apreciar en la figura 1.1.

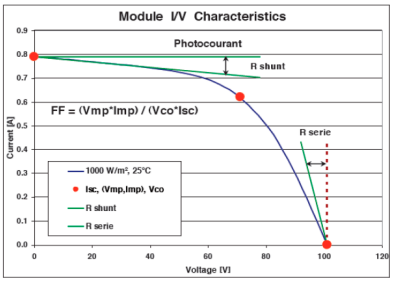
Curva I-V iluminada

Durante muchos años, la medida I-V bajo iluminación ha sido utilizada para evaluar el rendimiento eléctrico de las células fotovoltaicas y los diodos. Es un método muy sencillo y consiste básicamente en polarizar en directa la célula solar variando el voltaje a su entrada en pequeños incrementos desde cero hasta un voltaje predeterminado y registrando la corriente generada a su salida para cada paso de voltaje, obteniendo así una curva parecida a la figura 1.2b.

A continuación se detallan los parámetros más importantes de una célula solar que se pueden extraer a partir de su curva I-V iluminada.



(a) Curva de potencia de una célula fotovoltaica



(b) Curva I-V iluminada de una célula fotovoltaica

Tensión de circuito abierto

Se llama tensión de circuito abierto (VOC) a la cota superior de tensión que puede aportar una célula fotovoltaica iluminada y se da en condiciones de circuito abierto. Es la corriente máxima cuando la tensión es nula. Gráficamente, es el punto donde la curva I-V cruza el eje voltaje o de abscisas (1.2b)

$$V(I = 0) = Voc$$

Corriente de cortocircuito

Se define corriente de cortocircuito (Isc) a la cota superior de corriente que podría aportar la célula y se da cuando el voltaje es nulo. Gráficamente, es el punto donde la curva I-V cruza el eje de ordenadas o corriente (1.2b)

$$I(V = 0) = Isc$$

Punto de máxima potencia

La potencia generada por la célula fotovoltaica se calcula como el producto del voltaje y la corriente obtenidos en la medida I-V.

$$P = I * V$$

El punto de máxima potencia corresponde al par (Vmp,Imp) que hace máxima la potencia Pmax. (1.2a)

1.2.2. Multiplexión

El desarrollo de un sistema Multiplexor o Switching Board permite el aprovechamiento de recursos de alto valor como son los instrumentos de Test y medida tales como osciloscopios, multímetros, etc.

Por especificaciones del diseño y para cumplir con el objetivo de reaprovechar y adaptar los instrumentos actuales del laboratorio del NTC, se va a utilizar el equipo de medida Keithley 2651A. Se trata de un instrumento tipo SMU (Source Measurement Unit), que integra las funciones de fuente de voltaje y corriente además de un multímetro digital de alta precisión.

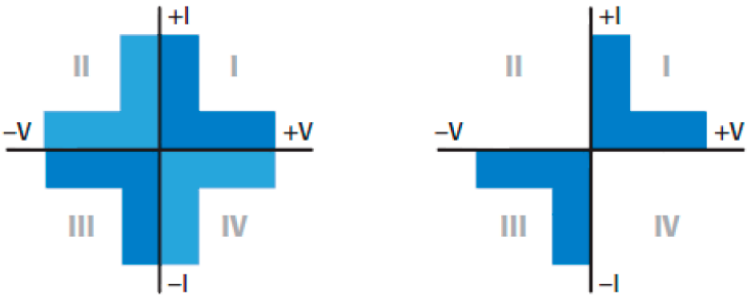


Figura 1.3: A la derecha una fuente de alimentación típica (2 cuadrantes) y a la izquierda una SMU (4 cuadrantes)

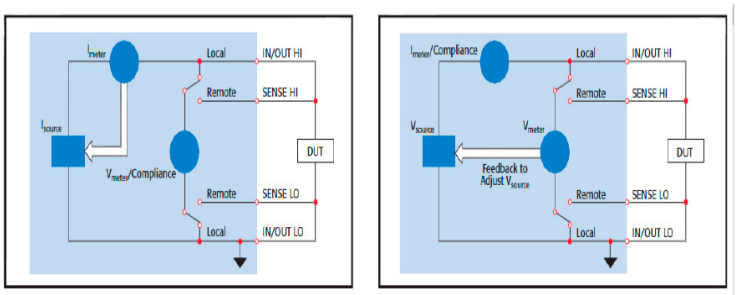


Figura 1.4: Diagrama simplificado del keithley 2651A para medida a 4 hilos, a la derecha fuente V y medida I, a la izquierda fuente I y medida V

Dentro del campo de la fotovoltaica es muy interesante poder realizar medidas de la curva V/I en condiciones reales de operación. Por lo que es necesario que el conjunto de medidas se realice en un periodo de tiempo muy corto, para poder considerar las mismas condiciones meteorológicas y, por lo tanto, poder hacer una comparativa entre todas ellas.

1.2.3. Medidas a 4 hilos

Una fuente de corriente o voltaje normal, sólo permite trabajar en los cuadrantes I y III (figura 1.3), pero este instrumento (SMU), dado que puede actuar como fuente de alimentación y carga electrónica, permite trabajar en los cuatro cuadrantes. Esto se traduce en poder suministrar voltaje o corriente y medir simultáneamente la corriente o voltaje siendo muy adecuado para la medida de curvas I-V.

Hacer una medida a 4 hilos permite reducir los errores producidos por los cables y contactos, por lo que todo lo que se encuentre dentro de los 4 hilos queda eliminado de la medida.

1.3 Estado del arte

Actualmente, como se puede observar en la tabla 1.1, contamos con múltiples soluciones comerciales capaces de multiplexar las señales a medir. Pero la corriente que son capaces de generar los módulos actuales hace inservible la mayor parte de soluciones comerciales. Gracias a realizar una solución personalizada, podemos cumplir con los requisitos que consideremos más importantes, despreciando otros que si han sido considerados en las soluciones comerciales.

KEITHLEY 7702 [1]	20 Four-Wire Measurement Channels	300V 1A
NI PXI-2527	16 Four-Wire Measurement Channels	300V 2A
Nuestra Solución	8 Four-Wire Measurement Channels	400V 16A

Tabla 1.1: Comparativa Módulos Multiplexores

(a) KEITHLEY 7702

(b) NI PXI-2527

Figura 1.5: Multiplexores Comerciales

Como ejemplo, podemos ver en la figura 1.5, que contamos con soluciones pensadas para sistemas personalizados como los que ofrece National Instruments, una de las compañías lideres en Test y Medida. Por otro lado podemos ver la versión de Kethley, que aunque se parece más a lo buscado, no cumple con los requisitos mínimos.

Por otro lado, están las soluciones de Switching que se realizan para las empresas, mediante módulos comerciales que cumplan individualmente los requisitos que se necesitan para las medidas pertinentes. Como ejemplo tenemos la solución de la figura 1.6, perteneciente a la empresa Celestica Valencia. Con esta solución, se realizan test de alta frecuencia a módulos para satélites, utilizando un sólo generador de señales y un analizador de espectros. Gracias a los sistemas de Switching se puede hacer un barrido por diferentes combinaciones de entradas y salidas en el módulo.

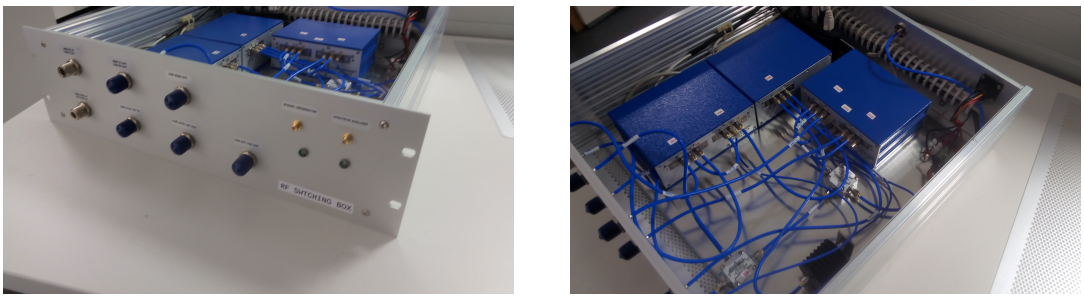


Figura 1.6: Switching Box - Celestica Valencia

CAPÍTULO 2
Metodología

2.1 Gestión del proyecto

El desarrollo del proyecto se estructuró en 4 etapas bien diferenciadas y consecutivas, que permitía controlar el estado del proyecto.

Durante la primera etapa, se concretó la mayor parte de las condiciones que tenía que cumplir el diseño y se concretó una estructura para este. A su vez, se empezaba la búsqueda de los relés que fuesen capaces de soportar los requisitos del diseño.

La segunda etapa se centro en el desarrollo de un prototipo de dos canales en breadboard, a la vez que se diseñaba el esquemático del sistema completo.

El desarrollo de la PCB se realizó en la tercera etapa, mediante la creación de una librería de footprints, el placement o colocación de componentes y, posteriormente el rutado.

El firmware se sitúa en la cuarta etapa, la preparación de este junto a todo el software necesario para su control mediante labview.

Como software de diseño se ha utilizado Altium Designer, el cual es un software de diseño para circuitos impresos, FPGA y software embebido. Entre otras muchas funciones, han sido provechosas la funciones del control de versiones, el procesado de trabajos para la gestión de la documentación, el diseño 3D del circuito y el gestor de librerías integradas. Altium esta adquiriendo fuerza en el panorama comercial frente a otros softwares de EDA como puede ser Orcad de Cadence.

2.2 Diagrama Temporal

Las cuatro etapas principales de desarrollo se han desglosado en 12 sub tareas como podemos apreciar en el diagrama de Gantt del proyecto en la figura 2.1,

Como inicio, se concretó el producto que necesitábamos y se hizo un concepto previo de este, para posteriormente concretar las especificaciones que necesitábamos cumplir. Cada vez que se cambiaba la idea del producto, se pasaba por otra fase de concepción de diseño y definición de las especificaciones. Por todo ello, son tres bloques, que con un pequeño delay, se desarrollan de forma simultánea.

Una vez que ya estaba claro el diseño en su mayoría, se pasó a buscar componentes, tanto para el producto final como para el prototipado, que se hizo simultáneamente al desarrollo de los esquemáticos.

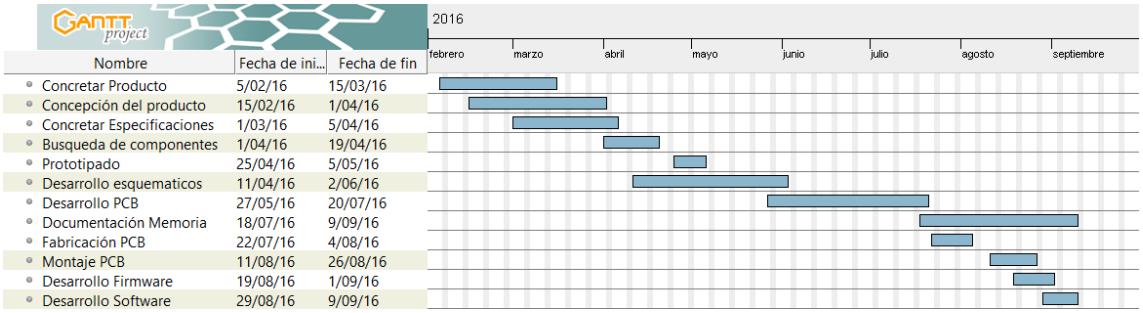
El desarrollo de la PCB, comenzó cuando se tenían unos esquemáticos finales pero que siempre se cambiaban y modificaban para adaptarlo a la PCB.

Esta memoria comenzó cuando el diseño tenía ya un resultado final y podíamos empezar a describirlo.

Como podemos ver en el apéndice K, se encargó a SAFE-PCB, una empresa con sede comercial en España, pero que se basa en la fabricación china para conseguir unos precios bajos.

Una vez llegada la placa, empezó el montaje de la parte digital de la placa y su prueba para evitar estar con todos los relés montados mientras que se verificaba la parte digital.

Cuando la parte digital estaba comprobada, se empezó a desarrollar el firmware y software necesario para su uso, mientras que se terminaba el montaje de los relés y se verificaba en su conjunto.



Nombre	Fecha de ini...	Fecha de fin
Concretar Producto	5/02/16	15/03/16
Concepción del producto	15/02/16	1/04/16
Concretar Especificaciones	1/03/16	5/04/16
Busqueda de componentes	1/04/16	19/04/16
Prototipado	25/04/16	5/05/16
Desarrollo esquematicos	11/04/16	2/06/16
Desarrollo PCB	27/05/16	20/07/16
Documentación Memoria	18/07/16	9/09/16
Fabricación PCB	22/07/16	4/08/16
Montaje PCB	11/08/16	26/08/16
Desarrollo Firmware	19/08/16	1/09/16
Desarrollo Software	29/08/16	9/09/16

Figura 2.1: Diagrama de Gantt

CAPÍTULO 3
Desarrollo Hardware

3.1 Condiciones

El sistema tenía que cumplir una serie de requisitos, para poder trabajar con los nuevos modelos de placas solares, lo cual hizo que la búsqueda de componentes fuese bastante crítica. Además había una serie de medidas de protección que debía cumplir el diseño.

Los nuevos paneles solares poseen células solares con corrientes de cortocircuito próxima a los 10A, por lo que el sistema de conmutación debería ser capaz de hacer circular corrientes superiores a la comentada. Con ello aseguramos que el multiplexor podrá utilizarse con los módulos fotovoltaicos del futuro, realizados con células solares cada vez más eficientes.

Otra condición era la de proteger y aislar la parte de potencia de la del sistema de control, en este caso el PC que gestiona la medida y el propio microcontrolador, lo que hizo buscar soluciones de aislamiento óptico.

Durante las medidas de los módulos solares, en escenarios reales, es muy importante la monitorización de la temperatura por lo que se buscó una solución de control de esta.

La última condición era la de que fuese un sistema ampliable y modular, por lo que se pensó en un sistema capaz de crecer y comunicarse mediante un sólo interfaz de forma que pudieras poner en cascada los diferentes módulos.

3.2 Concepción y diseño

Relés

El sistema se concibió como una matriz de cuatro path simultáneos con los que conmutar entre los diferentes módulos solares, partiendo de esa premisa, se preconcibió como un array de relés 4PDT (3.1) (4 pole, double throw) para asegurar una conmutación simultanea.

Los relés 4PDT fueron descartados, debido a que para poder conseguir las corrientes que necesitábamos, o se encarecía el producto final o su factor de forma, obligaba a que el diseño fuese demasiado grande. De igual forma, se buscaron relés 2PST que permitieran reducir el circuito, pero los relés compactos no conseguían soportar las corrientes de conmutación que necesitábamos.

Figura 3.1: Relé 4PDT

Como opción final se optó por relés SPST (3.2), que mantenían un factor de forma reducido, soportaban la corriente necesaria, además de que el precio unitario compensaba sobre las demás soluciones propuestas anteriormente.

El RZ03-1A4-D005 [3] cumplía con los requisitos de corriente, además tenía un precio y tamaño que se ajustaban a la idea del proyecto. De esta forma, se estableció la opción definitiva para el diseño, quedando así este, como una matriz de bloques de 4 relés SPST, que conmutasen las 4 líneas de medida a la vez. Esto hace que el sistema de control fuera común para los 4, asegurando una conmutación simultánea.

Figura 3.2: Relé SPST

Darlington

Una vez elegido el relé, se tuvo que proporcionar la corriente necesaria para excitarlo y mantenerse estable. Por lo que se optó por un array de par Darlington capaz de alimentar dos canales de nuestro sistema multiplexor.

El ULN2803A [4] fue elegido como mejor candidato, ya que contaba con diodos de protección (3.3) contra corriente inversa, que vienen perfectos para las producidas por las bobinas de los relés.

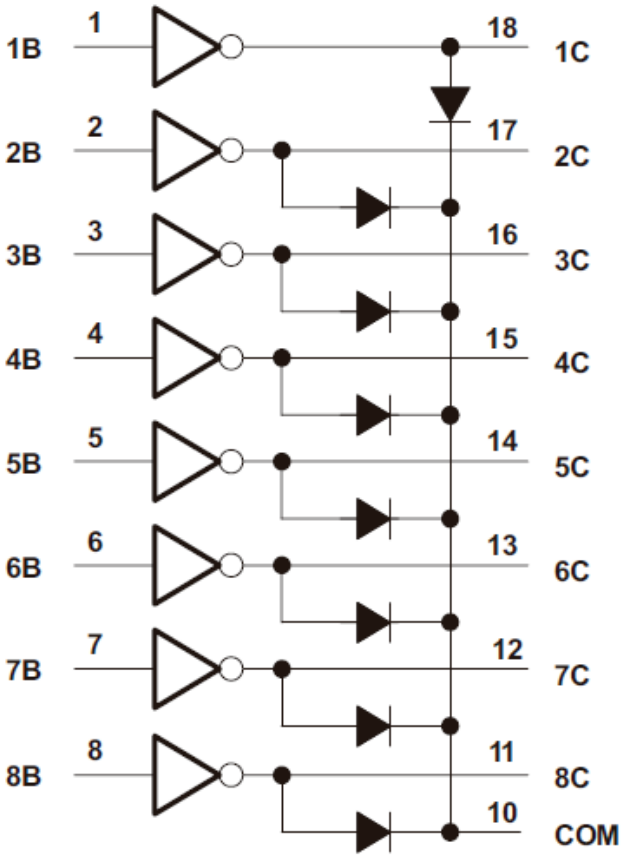


Figura 3.3: ULN2803

Al dejar de excitar las bobinas de los relés, toda la carga que han acumulado estas, vuelven al circuito. Si no hubiera algún mecanismo de protección, que fuese capaz de disipar esa corriente, esta sería disipada por otros medios que podrían estropear algunos circuitos.

Microcontrolador

La elección del controlador para el sistema fue uno de los puntos claves ya que existían diferentes opciones. En primer lugar, se barajó el uso de multiplexores con puerto serie o algún sistema hardware, pero finalmente se optó por un microcontrolador que le daba flexibilidad al sistema y posibles ampliaciones, así como poder desarrollar complejos protocolos de control con el PC.

El Atmega32U4 [5] ha sido el micro seleccionado, ya que cuenta con un bootloader USB y es utilizado en la plataforma de Hardware libre Arduino Leonardo (3.4).

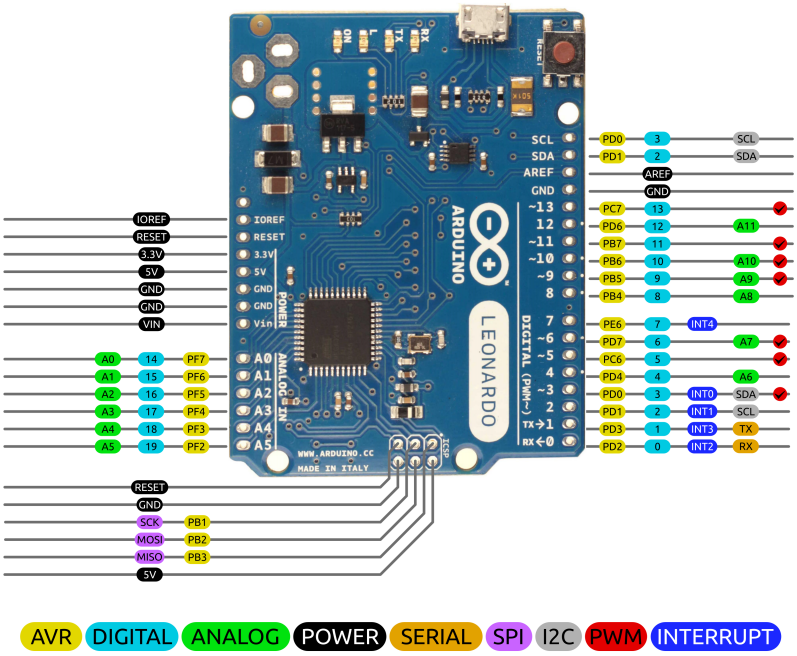


Figura 3.4: Arduino Leonardo PIN OUT

La ventaja del USB bootloader es que permite reconocer el microcontrolador directamente desde el PC. Por otro lado, el hecho de que sea compartido con el Arduino Leonardo, nos ayuda a poder utilizar este para la realización del prototipo. Y, finalmente, cargar al micro el firmware mediante el bootloader propio de Arduino, pudiendo utilizar de esta forma el IDE de este que facilita la programación de nuevas versiones y su desarrollo.

Para la carga final del bootloader de Arduino se utilizó un Arduino UNO como programador ISP mediante una serie de cables soldados al circuitos para la programación inicial (3.5).

A la hora del desarrollo del circuito para el micro nos hemos basado en el descrito por el Club de Informática, Robótica y Electrónica de Madrid [6], que cumple los requisitos mínimos para el funcionamiento correcto del microcontrolador.

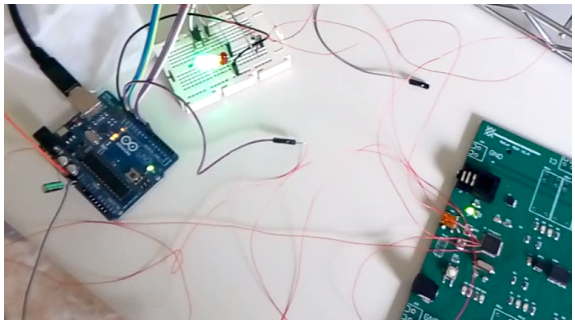


Figura 3.5: ISP

Optoacoplador

Como se ha comentado anteriormente, uno de los requisitos era proteger el sistema de control, por lo que se buscó un optoacoplador (3.6) capaz de suministrar la corriente necesaria como para excitar los Darlingtons sin necesitar de mucha corriente del sistema de control.

Se eligió el VOD207T [7], ya que alimentándolo con 10mA tiene un coeficiente de transferencia del 100 % por lo que podemos alimentar 2 de ellos por cada uno de los pines del microcontrolador, ya que este puede dar hasta 20mA.

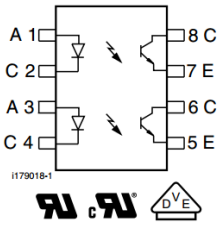


Figura 3.6: Optoacoplador

Termómetro

Para conseguir monitorizar la temperatura en el multiplexor como en tantos módulos como se deseé, se pensó en el termómetro DS18B20 [8] de la casa Maximin, que se comunica mediante el protocolo 1-Wire. Este permite comunicarse con múltiples dispositivos utilizando un solo BUS, por lo que podemos poner en cascada todos los termómetros deseados. Por otro lado, Arduino tiene implementadas las librerías necesarias para su utilización, gracias a ello, su implementación dentro del firmware se ve simplificada.

3.3 Esquemático

El desarrollo del esquemático lo podemos dividir en dos partes. Una inicial en la que se desarrolló el modelo de un canal de medida(3.7) para poder implementarlo en el prototipo y, posteriormente el diseño de la parte final del microcontrolador, que se une al diseño anterior.

El diseño final completo lo podemos ver en el Apéndice (A), donde las 4 primeras hojas son las 4 instancias de los 2 canales diseñados. A continuación contamos con la conexión a los conectores, el diagrama de bloques final y acabamos con el esquemático del microcontrolador.

3.3.1. Canal de una medida - Diseño Previo

Cada uno de los canales se diseñó con la siguiente estructura (3.7). Esta configuración destaca, por que aunque tenemos el inconveniente de que la corriente para los optoacopladores sale desde el micro controlador, aseguramos de esta forma que en caso de

no estar conectada la parte de control, ninguno de los canales estará activo, evitando así posibles cortocircuitos.

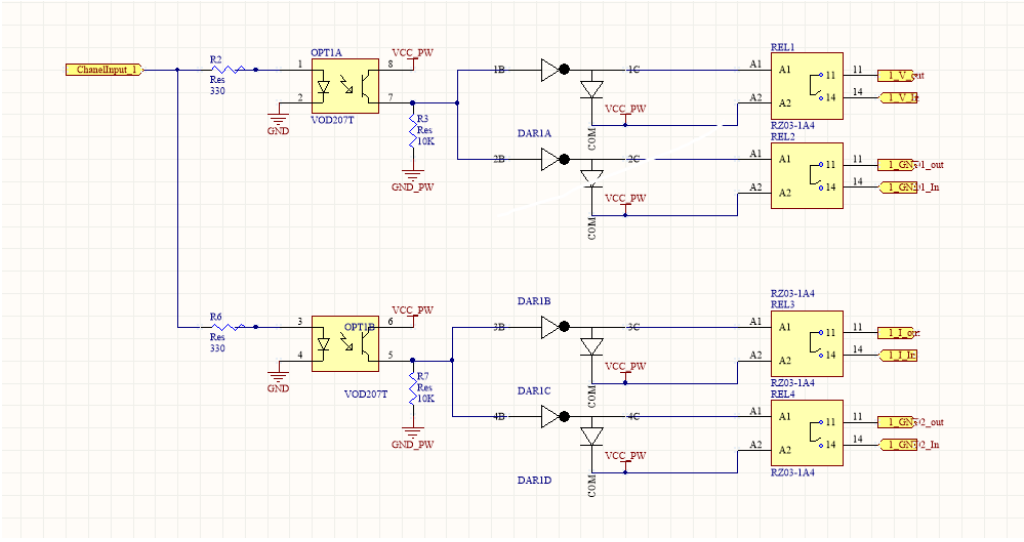


Figura 3.7: Esquemático un Canal

Cada uno de los canales del micro procesador, ataca a dos optoacopladores, que atacan a su vez a 4 pares Darlington, consiguiendo así suministrar de forma estable la corriente para cada relé.

El calculo de las resistencias de entrada, a los diodos del optoacoplador, está calculado para conseguir los 10mA con los que hay una transferencia óptica del 100 %. Gracias a esto, al otro lado tendríamos esos mismos 10mA para, con la resistencia, generar un nivel alto y así activar el Darlington.

3.3.2. Canal de una medida - Diseño Final

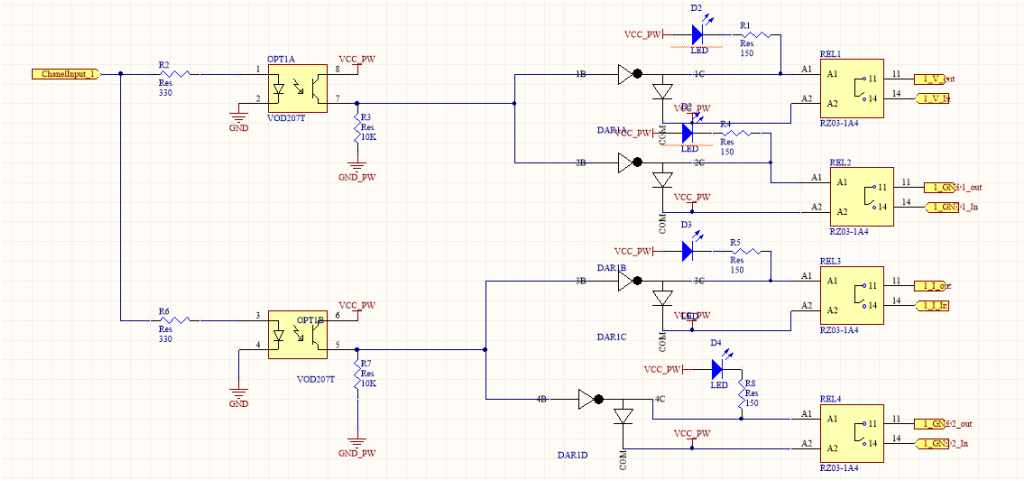


Figura 3.8: Esquemático final de un Canal

En el diseño final del canal(3.8), se ha incluido una serie de LEDs, que se encienden a la vez que conmutan los relés, lo que nos da información de depuración sobre los posibles fallos en el canal. En caso de que los LEDs estén conectados y el relé no haya conmutado, podemos descartar problemas previos, quedando como único posible problema un fallo en el relé.

3.3.3. Microcontrolador

Dentro del esquemático del microcontrolador(3.9), podemos destacar el juego de condensadores de acoplo para la alimentación y el botón de reset, que pone el micro en modo bootloader. Por otro lado tenemos los LEDs que nos sirven para depurar el sistema. Contamos con uno que nos indica el estado de la alimentación del USB y un par que nos indican la ocupación del puerto serie, tanto en transmisión como en recepción.

En la parte inferior podemos observar el cristal de 16 MHz con sus cargas paralelas de 18pF.

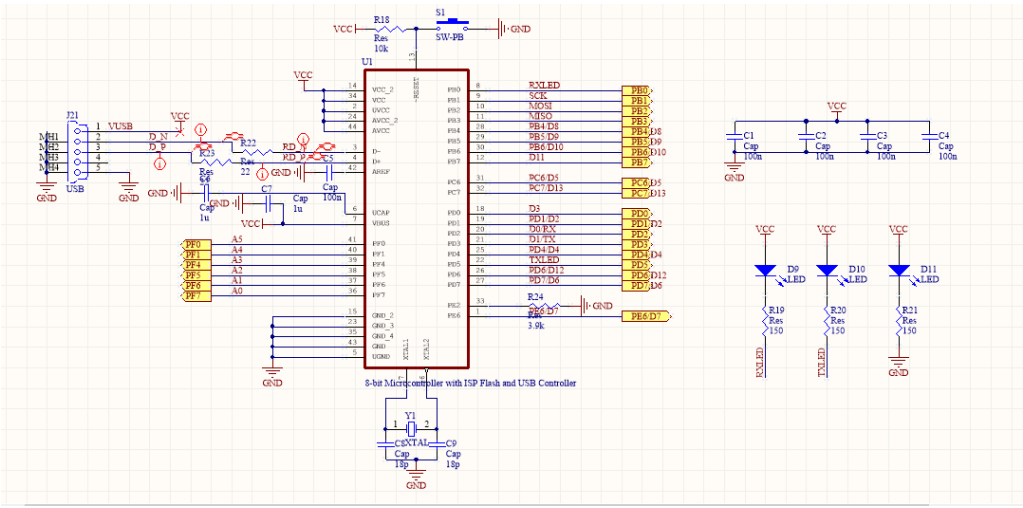


Figura 3.9: Esquemático microcontrolador

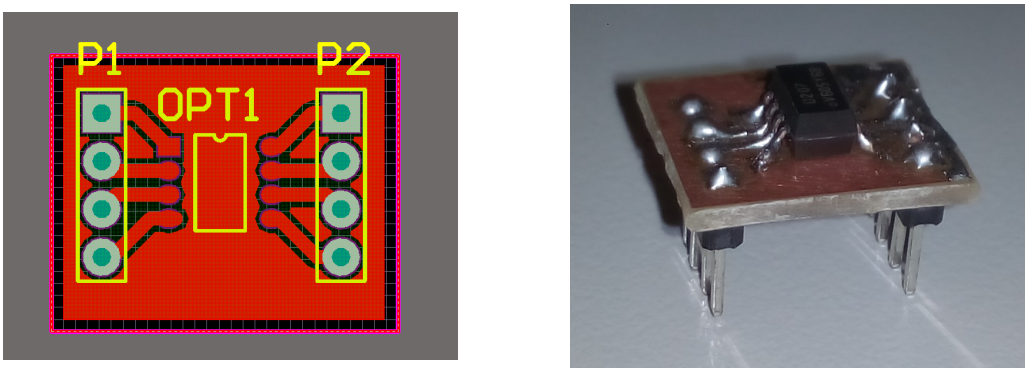
3.3.4. Comunicación entre placas

Uno de los requisitos previos era que fuese modular y ampliable, así que para cumplir con este, se ha añadido un bus I2C. Cada módulo tiene que detectar si tiene conectado el bus USB, para ponerse como esclavo y este escuche al maestro, que le indique cuando debe conmutar. Junto al bus I2c se traspasan, entre las placas, las alimentaciones, tanto del micro como la de potencia. Que la alimentación de potencia pase de placa a placa, se elije mediante un jumper, ya que puede ser mejor poner varios adaptadores de corriente si hay muchas placas en cascada.

3.4 Prototipo

Para el prototipo se montaron dos canales en una breadboard. Como controlador se utilizó una placa Arduino Leonardo, que incluía el Atmega32u4. Se pidieron samples del array de Darlington y del termómetro en package DIP. Los relés se tuvieron que conectar mediante cables de conexión ya que no se adaptaban a la breadboard, mientras que los optoacopladores no tenían versión DIP y hubo que hacerles una PCB adaptadora (3.10).

Una vez montado el prototipo completo para 2 canales (3.11), junto al termómetro, se le añadió un pulsador para poder hacer el cambio de canal de forma manual y no tener que estar mandando comandos al Arduino. En el apéndice (B) vemos el firmware que se le cargó al Arduino, el cual manda la temperatura por el puerto serie de forma periódica. Además, está alerta a las interrupciones posibles en el pin del pulsador, para así cambiar de canal.



(a) Layout adaptador DIP

(b) Resultado Final

Figura 3.10: Adaptador DIP

Uno de los primeros test que se llevaron acabo con el prototipo, fue el de verificación del tiempo de conmutación de los relés. Se consiguió que el firmware consiguiera ajustar lo máximo este tiempo a los de conmutación que indica la documentación de los relés, que es de unos 8 ms, consiguiendo conmutar de forma estable en menos de 15 ms con nuestro firmware.

Gracias al prototipo, se pudo verificar todo el circuito que hacia de interface entre el microcontrolador y los relés, comprobandose así, que el valor de las resistencias fuera correcto. Dentro de los test realizados se analizaron los gliches producidos por un posible rebote de la placa del relé a la hora de conmutar.

Una de las características principales del diseño es que, si se queda sin control, todos los relés estén desactivados. Se verificó que el circuito cumplía con este requisito y no había ninguna forma de hacer que el relé estuviera activo sin utilizar el microcontrolador. Con esta pequeña consideración se evita estropear los equipos de medida que van a haber detrás de nuestro diseño.

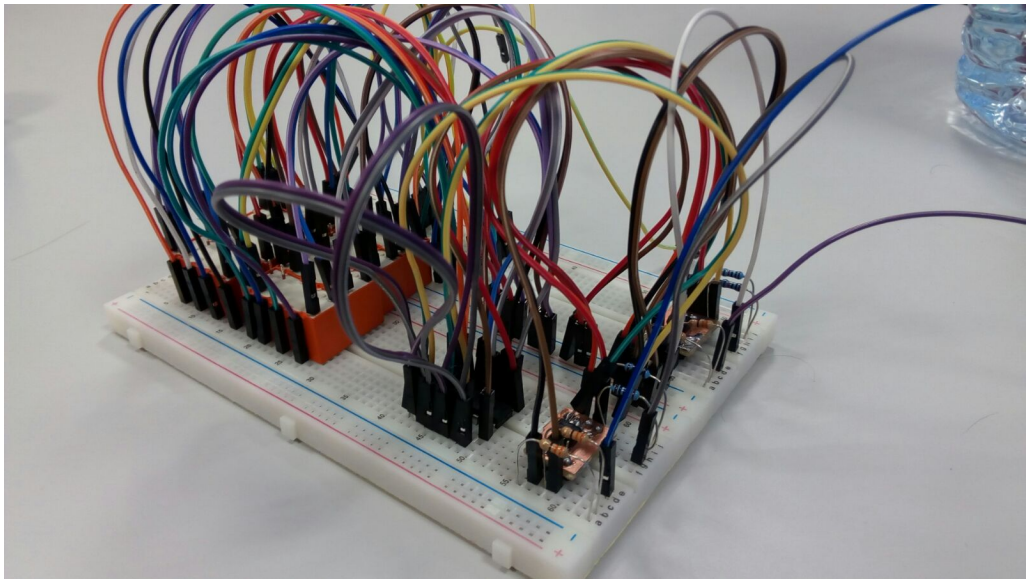


Figura 3.11: Prototipo

Entre todos los test realizados , se llevó a cabo uno de larga duración para verificar que en ningún momento conmutaba cuando no tocaba. Mediante un simple ensayo con un osciloscopio, en el cual se esperaba capturar un descenso de nivel en la linea de medida.

Producido este por una conmutación indeseada, la cual sólo tuvo lugar cuando se forzó mediante el botón incorporado en el prototipo.

3.5 PCB

La primera premisa que se tuvo en cuenta a la hora de hacer el diseño de la PCB, fue el usar solamente dos capas, ya que al principio del proyecto, no se sabía si se iba a poder realizar en una empresa externa o la placa, la tendríamos que mecanizar en alguno de los laboratorios de la Universidad.

El utilizar componentes SMD, corresponde con la idea de poder reducir al máximo el tamaño de la placa, ya que era mucho el espacio necesario para todos los relés. Sin embargo, se eligió un encapsulado grande para poder soldarse a mano sin muchos problemas. En caso de haberlo desarrollado para producción en serie y mediante una Pick and Place, se podría haber elegido un package más pequeño en los pasivos y junto a aumentar el número de capas podríamos haber conseguido un producto de reducidas dimensiones.

3.5.1. Librería

El primer paso al desarrollo de la PCB, fue la creación de una librería adaptada al proyecto, porque aunque Altium cuenta con la mayor parte de pasivos, no tiene todos los encapsulados disponibles.

Se incluyeron en la librería los siguientes componentes personalizados:

- Atmega32u4
- Condensador 0805 y 1206
- Header 2X2 y 4x2
- LED 1206
- Conector Jack
- Conector de Power (3.13)
- Resistencia 1206
- RZ03-1A4-D005
- Terminal Block (3.14)
- ULN2803
- Conector USB (3.12)
- VOD207T
- Cristal

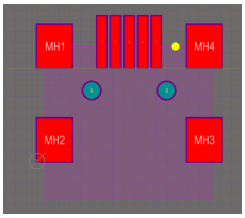


Figura 3.12: USB Footprint

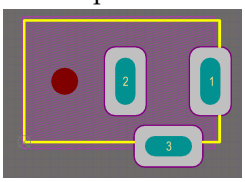


Figura 3.13: POWER JACK Footprint

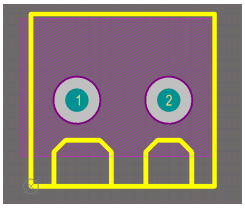


Figura 3.14: Terminal Block Footprint

El hecho de editar casi todos los componentes necesarios, se debió a que al no saber si habría que mecanizar toda la placa a mano, hubo que adaptar todos los Footprint para ese caso, que era el más desfavorable.

3.5.2. Placement - Colocación de componentes

El placement o colocación de los componentes en la placa, lo podemos observar en el apéndice (C). Apreciamos como se han mantenido todos los conectores, para los módulos solares, en los laterales, junto a todos los relés en fila, mientras que se ha dejado en el centro toda la parte de control para separarla de la parte de medida.

Por otro lado, y bajo la premisa de que fuera ampliable, se han añadido puertos de medida y de control al inicio y final de la placa para poder poner en cascada varias de estas. Además de estos puertos, se ha añadido un conector de Jack estéreo para poder conectar termómetros DS18B20 mediante un cable adaptador.

3.5.3. Rutado

En el apéndice (D), podemos apreciar el rutado de la placa. Todos los paths de medida se sitúan en la parte exterior con un ancho suficiente como para soportar las corrientes de medidas y aislados del resto del circuito.

El puerto USB fue una de las cosas en las que se puso especial atención, ya que para conseguir una máxima transferencia de velocidad necesitábamos tener la máxima adaptación de impedancia en el puerto; este tenia que estar a 90Ω diferenciales.

Se puede apreciar, como se ha intentado mantener las líneas de control en sentido horizontal, mientras que las líneas de potencia, van en sentido vertical.

En el rutado del bottom, se puede apreciar como se comunica una placa con la siguiente mediante una serie de líneas que transfieren tanto la alimentación del USB, el bus I2C con el que se interconectarán las placas y la alimentación de potencia, bajo la elección del usuario mediante un jumper.

3.5.4. 3D

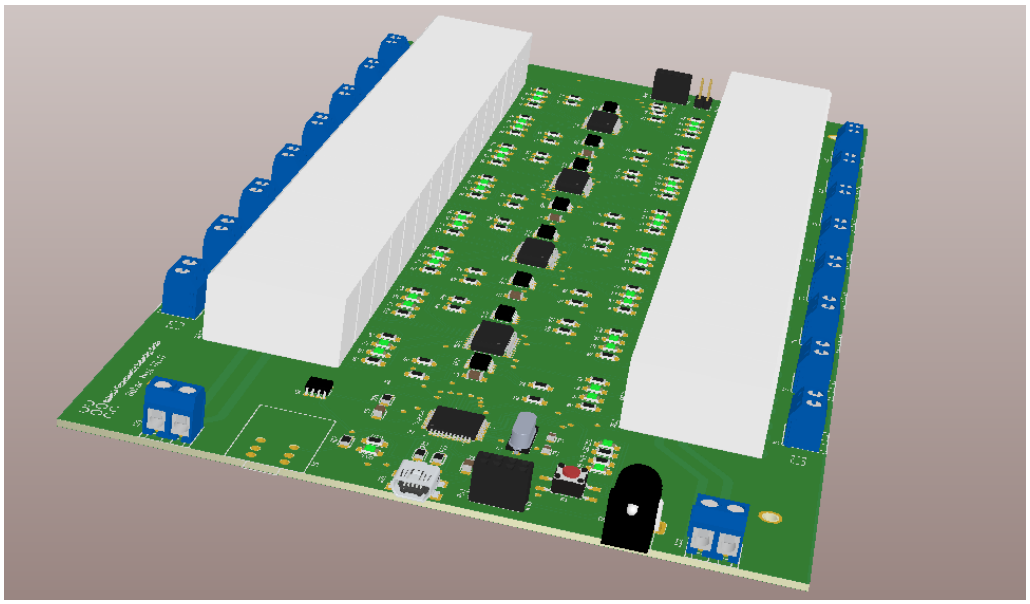


Figura 3.15: Captura 3D

Aprovechando las características que ofrece Altium, se complemento la librería del proyecto con los cuerpos en 3D de los componentes utilizados, dándonos una previsualización de la placa y su resultado final bastante fiable(3.15). Por otro lado en el apéndice E, podemos ver la salida 3D que genera Altium en su autodocumentación.

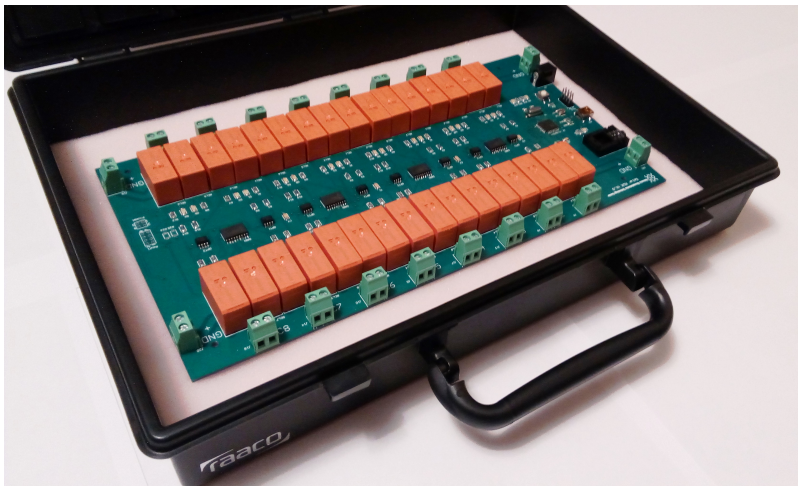


Figura 3.16: Protección Antiestática

3.6 Presupuesto

El presupuesto (3.1) lo podemos dividir en tres secciones: una primera que sería el precio de los componentes que se montan cada una de las placas, otra el precio de dicha placa en concreto y, por último el precio de los accesorios necesarios para el uso de esta, lo que en un producto final también incluiría la caja.

Se tiene como presupuesto en componentes (3.2) 132.92 €por placa, a lo se le añade el precio de la PCB (K), que en este caso fue de 95.19 €. A todo esto se le añade el coste del alimentador de los relés valorado en 14 €.

Componentes	132.92 €
PCB	95.19 €
Acesorios	14 €
Total	242.11 €

Tabla 3.1: Presupuesto

3.7 Montaje

A la hora del montaje se ha tenido precaución con las descargas electrostáticas, ya que es uno de los fenómenos más problemático en la industria de la fabricación de placas de circuito impreso. La descarga electrostática o ESD es un fenómeno que hace que circule una corriente eléctrica repentina y momentánea entre dos objetos de distinto potencial eléctrico.

Las descargas electrostáticas son un serio peligro para la electrónica de estado sólido, ya que pueden inutilizar dispositivos electrónicos. La prevención de ESD se realiza mediante un área de protección electrostática (EPA). El motivo principal de un EPA es

h

Comment	Description	Footprint	LibRef	Num	Value	Precio/Unidad	Total	RS part number
Cap	Capacitor	12 06 - C	Cap	5	100n	0.069	0.345	698-3670
Cap	Capacitor	12 06 - C	Cap	2	1u	0.034	0.068	653-0557
Cap	Capacitor	08 05 - C	Cap	2	18p	0.3059	0.6119	698-4046
LED	Typical BLUE SiC LED	12 06 - LED	LED	35		0.107	3.745	466-3908
ULN2803		SOP18	ULN2803	4		0.94	3.76	646-6311
TB		MKDSN	Terminal Block	20		0.98	19.61	804-5095
PWR2.5	Power Supply Connector	KLD-0202	PWR2.5	1		2.68	2.68	487-836
Phonejack	Jack Socket, 1/4"	ACJS-NH35	Phonejack	1		0.76	0.76	813-8875
USB	USB On-The-Go	USB	USB	1		1.49	1.49	748-0885
VOD207T		SOIC8	VOD207T	8		0.521	4.168	699-8461
Header 4X2	Header, 4-Pin, Dual row	HDR2X4	Header 4X2	2		0	0	No Montado
Header 2X2	Header, 2-Pin, Dual row	HDR2X2	Header 2X2	1		0	0	No Montado
Res	Resistor	12 06 - R	Res	35	150	0.063	2.205	223-2142
Res	Resistor	12 06 - R	Res	16	330	0.012	0.192	679-2049
Res	Resistor	12 06 - R	Res	17	10K	0.02	0.34	740-9110
Res	Resistor	12 06 - R	Res	2	3.9k	0.008	0.012	223-2344
Res	Resistor	12 06 - R	Res	2	22	0.008	0.016	223-2041
Res	Resistor	12 06 - R	Res	2	47k	0	0	No Montado
RZ03-1A4		1 form A (NO)	RZ03-1A4	32		1.87	59.84	791-7362
SW-PB	Switch	SPST-2	SW-PB	1		0.104	0.104	718-2474
8-bit Microcontroller		QFP80-44N	ATMEGA32U4-AU	1		5.28	5.28	718-2474
DS1820	1-Wire Digital Therm	SOIC8	DS1820	1		3.95	3.95	540-2811
XTAL	Crystal Oscillator	XTAL	XTAL	1		0.68	0.68	693-8885
Total							109.85	
Con IVA							132.92	

Tabla 3.2: Presupuesto de la BOM

ése: no estar cargando altamente el material en los alrededores de la electrónica sensible a ESD, poner a tierra todos los materiales conductores y poner a tierra a los trabajadores. Así la acumulación de la carga en electrónica sensible de ESD se evita. A la hora de planificar y diseñar un EPA es esencial la utilización de materiales conductores.

La primera y más esencial unidad de protección EPA para proteger el material electrónico expuesto a sufrir daños por descargas electrostáticas es todo aquel elemento que esté en contacto directo con el material a proteger. Por ello, la manipulación, transporte y almacenamiento debe realizarse utilizando productos fabricados con materiales conductores.

Para este proyecto se preparó un pequeño EPA para el montaje de la placa, contando con un conjunto de muñequera y alfombrilla de toma a tierra; además se eligió una estación de soldadura que cumpliera con la protección ESD pertinente.

Equipamiento EPA:

- Alfombra de seguridad ESD
- Conector macho de enlace a tierra
- Tornillo prisionero de cable de puesta a tierra
- Juego de cables y muñequeras de puesta a tierra ESD
- AOYUE Int 968A+

Para proteger el montaje en el transporte se dispone de un maletín antiestático 3.16 con cobertura de espuma. De esta forma está protegido tanto físicamente como eléctricamente.

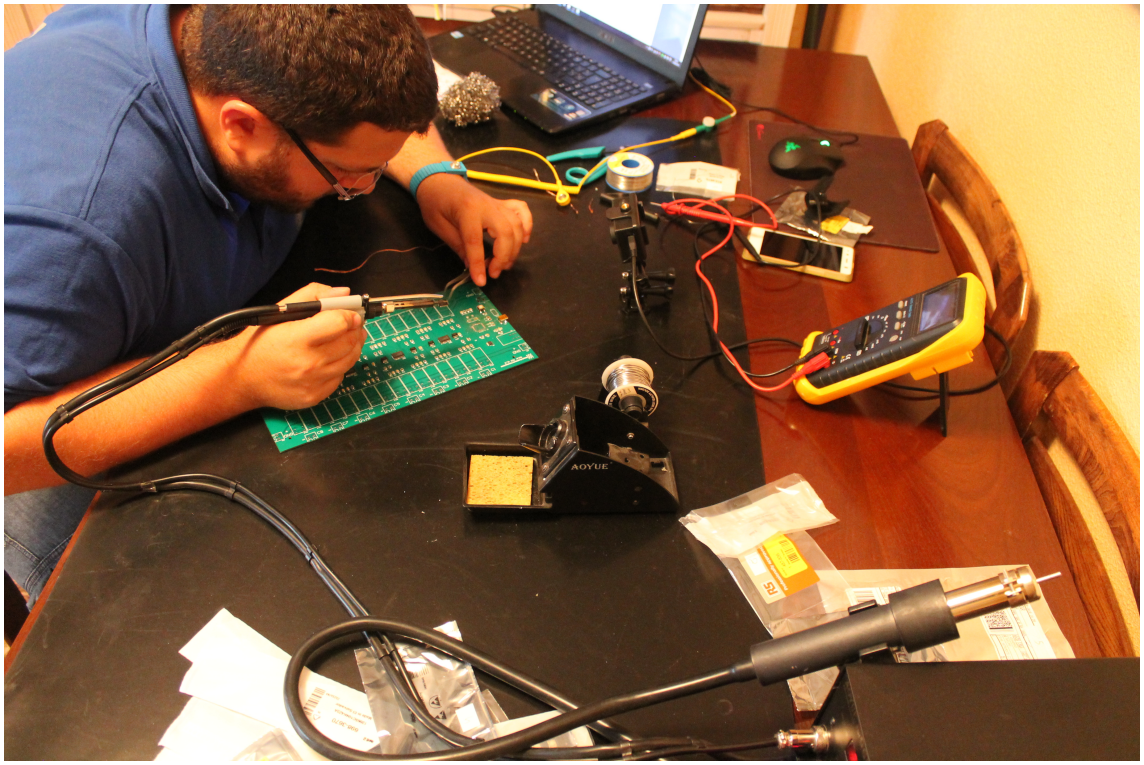


Figura 3.17: Montaje de la PCB

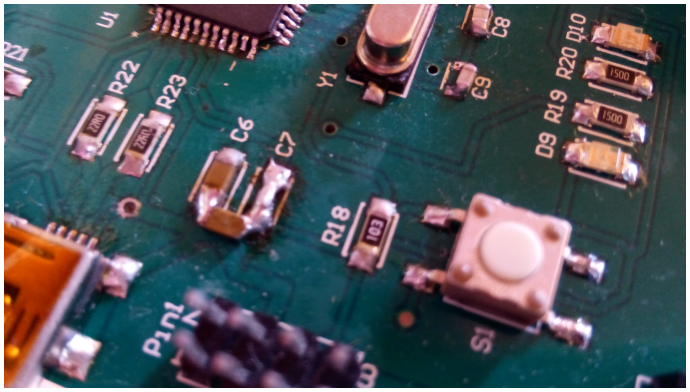


Figura 3.18: Corrección Condensador

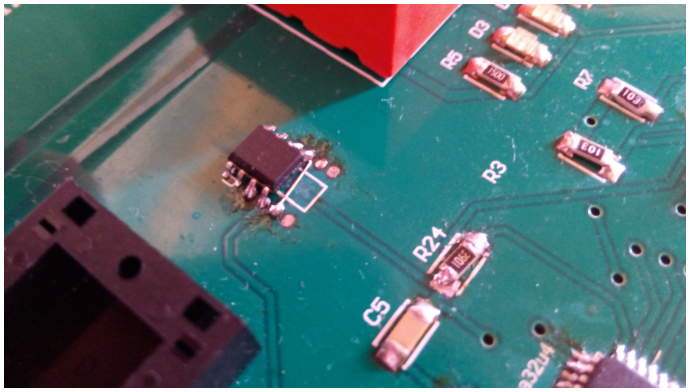


Figura 3.19: Solución termómetro

Problemas en el montaje

Una vez montado el proyecto y durante la verificación de la parte digital, se apreciaron 2 fallos dentro del diseño de la PCB. Uno primero por el que la señal de alimentación del USB no entraba al micro directamente, sino que atravesaba un condensador; y un segundo fallo en el footprint del termómetro.

El fallo en la parte del circuito del microcontrolador se produjo por un despiste, ya que en el esquemático de referencia, el condensador está bien colocado. Mientras que por otro lado el problema con el footprint del termómetro, se debió a utilizar la librería del fabricante la cual tenia asignado el pinout de package DIP y no fue revisado.

Por suerte ambos problemas pudieron solucionarse sin necesidad de modificar la placa. Se hizo un bypass en el condensador y se unió a un pad de tierra que teníamos justo al lado, figura 3.18. El termómetro, gracias a que tenía en uso pocos pines, pudo ser desplazado y recableado en la placa, figura 3.19.

CAPÍTULO 4
Desarrollo Software

El software necesario para este proyecto está divido en dos bloques: el primero es el firmware necesario para la utilización del modulo y, el segundo la librería necesaria para su uso en LabVIEW.

4.1 Condiciones y concepción

Firmware

El firmware tiene que ser capaz de una comunicación estable y que no se bloquee a la espera de comandos. A su vez tiene que poder realizar la mayor parte de las funciones deseables, dejando que el software que lo controle, no tenga que hacer ninguna gestión sobre el estado del sistema, reduciendo así todo el procesado de alto nivel para convertirlo a funciones de bajo nivel.

El firmware en su versión inicial debe ser capaz de devolver la temperatura actual del termómetro de la placa, así como el estado de los relés. Por otro lado, debe actuar seleccionando un canal en concreto, desconectando todos los canales y pasando entre uno y otro hacía arriba y hacia abajo.

LabVIEW

La librería y el software de test debe implementar todos los comandos admisibles por el Firmware y poder controlar este desde una interfaz gráfica, para poder posteriormente utilizarlo en software mas complejo, que controle otros sistemas como el medidor.

4.2 Diseño

4.2.1. Protocolo

El protocolo de comunicación serie esta basado en un sistema de comandos con confirmación de la operación, implementado mediante la clase importada SerialReceiver.

Un comando esta formado por la siguiente estructura.

[CMD;PARM1;PARM2]

Este protocolo está pensado para devolver diferentes tipos de error, si no se cumple alguna de las condiciones declaradas para el mensaje, del tipo: caracteres ilegales, demasiados parámetros o comando demasiado largo.

En el apéndice F podemos ver la cabecera de la clase, donde se aprecian todas las características del protocolo.

4.2.2. Control de la placa

Para controlar el hardware de la placa se han desarrollado las clases DS18B20 (G) y la clase SolarMUX (H), las cuales controlan el funcionamiento del termómetro y de todos los relés.

La clase DS18B20 (G) está diseñada para contestar al comando de la medida de temperatura, realizándose toda la configuración cuando se instancia un objeto de la clase DS18B20.

La clase SolarMUX (H) incluye la definición de los pines para los relés y una serie de funciones para el control básico. Disponemos de un conjunto de funciones privadas que se encargan de la gestión y del uso de las variables de control mientras que, además tenemos una serie de funciones públicas de la clase que nos permiten interactuar con el multiplexor desde fuera.

4.2.3. Sistema de eventos

La parte clave de todo el firmware, es el gestor de eventos, que se implementa en la clase SerialHandler (I), que se encarga de estar a la espera de que haya datos en el puerto serie, para entonces comunicarse con la clase del protocolo, para ver si hay un mensaje disponible para procesarlo.

Una vez que processInput() obtiene verificación por parte del protocolo de que todo esta correcto, llama a _switchYard() que se encarga de conmutar entre los diferentes comandos programados en funciones privadas. Esta clase tiene instanciados los objetos SolarMUX y DS18B20 para poder operar sobre ellos con los comandos.

4.2.4. Comandos

En el apéndice (I) podemos ver que los comandos implementados por el protocolo son los siguientes:

- [0] Imprime una ayuda con el protocolo y los comandos aceptados.
- [1] Devuelve la versión del firmware grabado.
- [2, *ch*] Activa el canal seleccionado en ch
- [3] Devuelve el canal activo actualmente
- [4] Pasa al siguiente canal
- [5] Pasa al canal anterior
- [6] Desactiva todos los canales
- [7] Devuelve la temperatura en la placa

Como se puede ver en los comandos, estos están pensados para que cualquier aplicación pueda usar la placa, sin necesitar estar pendiente del estado de los relés. Toda la gestión se realiza dentro del SerialHandler.

4.2.5. LabVIEW

Para mantener la idea de biblioteca se ha hecho un proyecto de LabVIEW, que contiene todas las dependencias necesarias para su uso.

Al igual que en el firmware, lo primero que se preparó fue un instrumento virtual capaz de mandar cualquier comando y recoger la respuesta del módulo, para posteriormente procesarla(4.2).

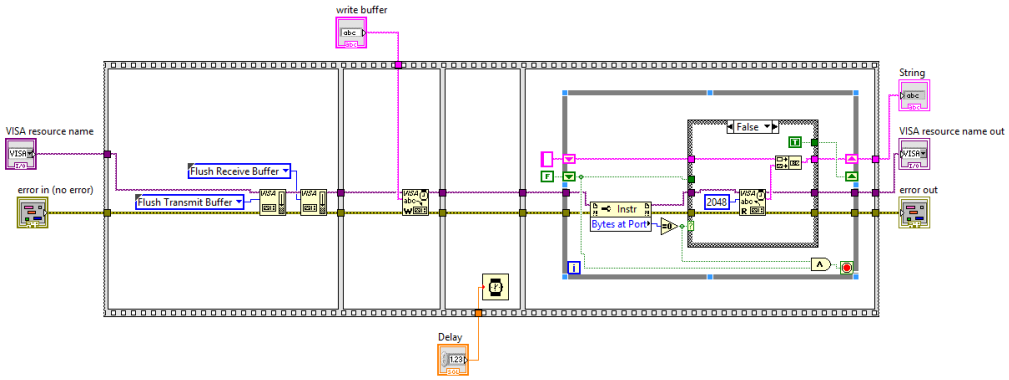


Figura 4.1: .vi para enviar un comando

En la figura 4.1 podemos ver, que para ser compatibles con el protocolo, antes de nada, se lleva a cabo una limpieza de los buffers de salida y entrada, para posteriormente, escribir en el puerto el comando deseado.

Debido a que algunos comandos, como el de la temperatura, necesitan un tiempo de procesado se ha añadido el paso de un delay para poder esperar a este tipo de medidas.

Para la realización de la lectura, se tiene un bucle con dos condiciones de parada: una que no queden Bytes en el buffer y la otra haber leído algo, ya que el protocolo siempre tiene mensaje de respuesta ante cualquier comando. La opción de parada por que no queden Bytes en el buffer no se activa hasta que no hemos leído algo, haciendo que el programa pare ahí hasta que recibe la respuesta.

Una vez que teníamos la función básica para emitir comandos y recibir sus respuestas, se ha desarrollado una serie de .vi que, utilizando el anterior como un bloque, realizan todos los comandos necesarios para poder controlar el módulo desde LabVIEW.

- clcCH.vi
- getCH.vi
- getTEMP.vi
- nextCH.vi
- prevCH.vi
- setCH.vi

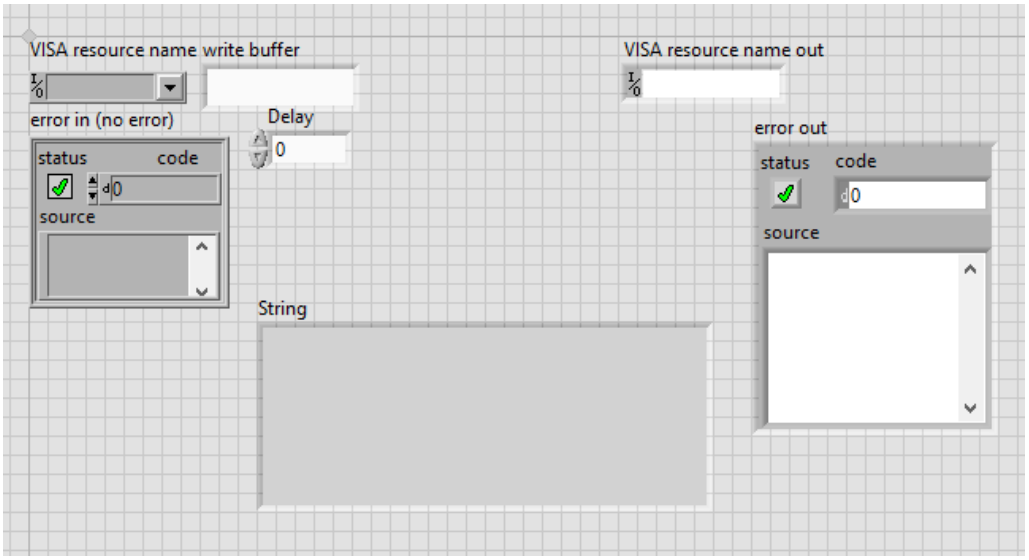


Figura 4.2: Panel frontal para enviar un comando

4.2.6. Test

Finalmente para la verificación del proyecto completo y su utilización, se ha desarrollado una interface general (4.3).

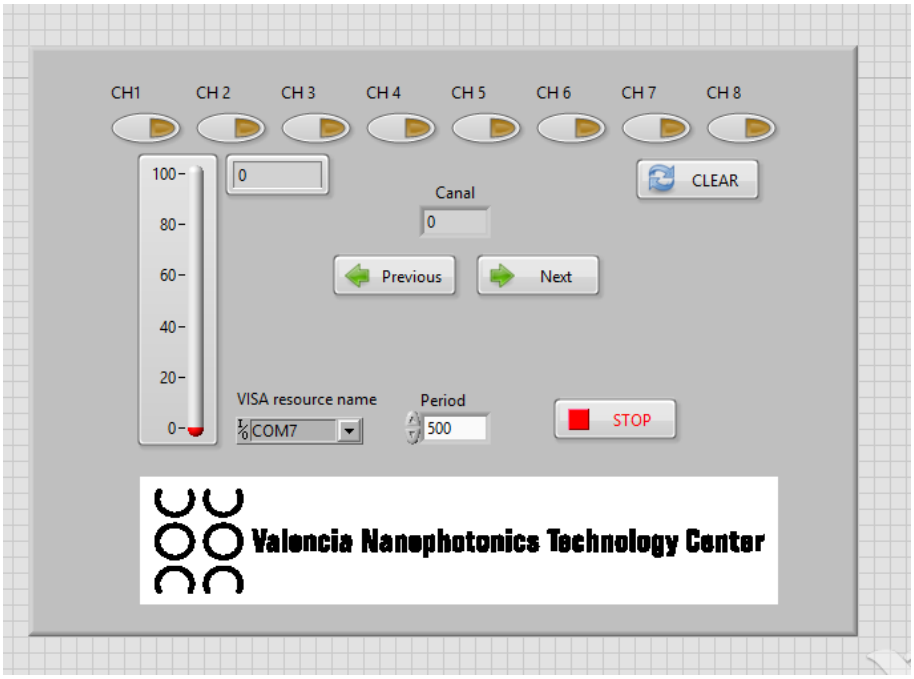


Figura 4.3: Interfaz de Test

Este interface permite el control total de la placa. Podemos desde ir cambiando de relé en un sentido u otro, desconectar todos o elegir manualmente el que queremos conmutar. Además tiene el display de un termómetro que monitoriza la temperatura de la placa con un periodo configurable.

En la figura 4.4 podemos apreciar los 2 hilos paralelos que contiene la aplicación de test, uno encargado de actualizar el termómetro y otro orientado a eventos cada vez que pulsamos alguno de los botones del interface.

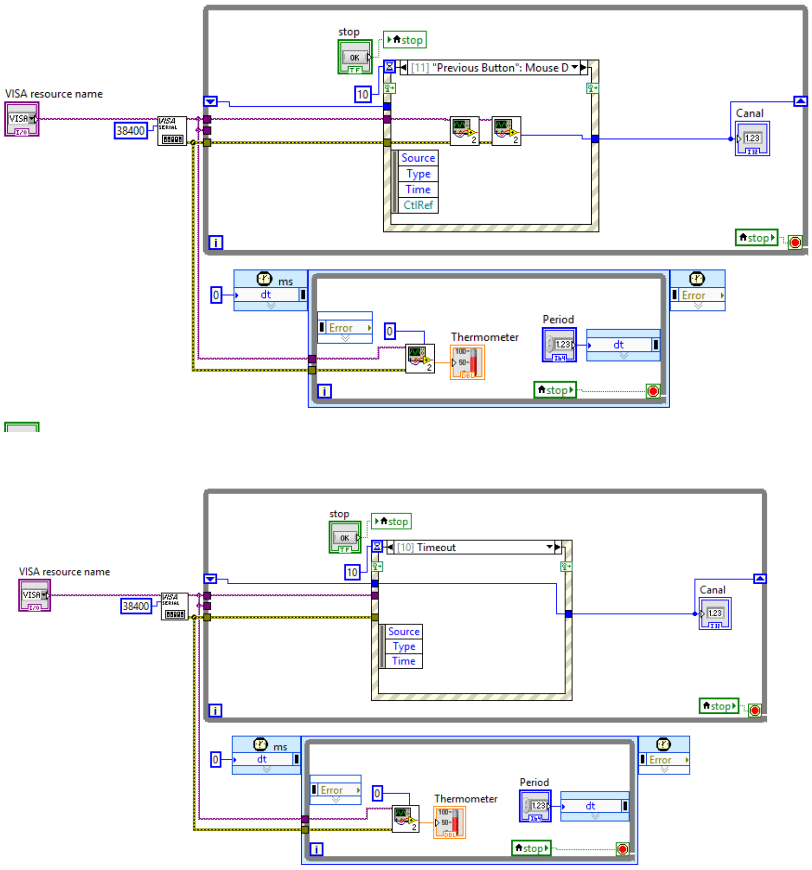


Figura 4.4: .vi de Test

El sistema de eventos tiene un caso para cada uno de los botones del interface, y otro para el timeout para que no se bloquee la ejecución del sistema, permitiendo actualizar el termómetro.

CAPÍTULO 5
Conclusiones y líneas futuras

Conclusiones

Mediante este trabajo se han podido desarrollar las competencias en gestión de proyectos de gran escala; así como su desarrollo a largo plazo, con tiempos que cumplir. El desarrollo del proyecto junto al análisis de circuitos y componentes es una buena forma de culminar los estudios y enlazarlos de esta forma con un trabajo más profesional.

Con esta primera versión de un posible producto comercial, se ha conseguido realizar una tarjeta de adquisición de señales, que permite medir hasta 8 módulos fotovoltaicos con un único multímetro. Teniendo en cuenta la baja resistencia serie de un módulo fotovoltaico actual, se ha optado por realizar medidas con la técnica de cuatro hilos con el fin de anular el efecto resistivo de los cables de medida. Consiguiendo así, un sistema final con el que poder medir corrientes de hasta 16 A

Por otro lado, se tienen unos tiempos de conmutación lo suficientemente cortos como para poder hacer todas las medidas necesarias sin tener que considerar estas como distintas, ya que las condiciones de test son las mismas para cada uno de los módulos solares con el objetivo de evitar que el paso de una nube pueda afectar a la medida. Cuanto más rápido se tomen las medidas, más probabilidades de que el nivel de irradiancia sea el mismo en todos los módulos analizados. Teniendo en cuenta que el equipo que realiza las curvas IV, requiere de sólo 330 ms para realizarla, considerando 130 puntos. De tal manera que los tiempos de conmutación de 10ms colocan el sistema dentro de las características de los sistemas comerciales.

Asimismo se puede considerar un producto modular y ampliable, ya que cuenta con un puerto para I2C y alimentaciones con el que comunicarse con todos los posibles módulos que cuelguen de el. Cabe decir que aunque actualmente contamos con sólo un termómetro montado, podemos conectar tantos como sea necesario gracias al conector Jack.

Líneas futuras

Este proyecto abre las puertas a una inmensa posibilidad de mejoras para poder llegar a hacerlo incluso un producto comercial competente.

Estas líneas futuras se pueden dividir en 3:

Hardware

Aunque ya tiene medidas reducidas, el sistema podría conseguir unas aún menores. Además se necesitarían sistemas de protección, tales como diodos TVS (Transient-voltage-suppression diode) encargados de absorber una descarga transitoria, como podría ser una descarga ESD, impidiendo que llegue a los integrados delicados desde los puertos exteriores.

También se puede rediseñar el hardware para tener un feedback del estado de los relés.

Firmware

El firmware está en una versión inicial con una funcionalidad básica. Entre otras cosas, uno de los primeros pasos para su desarrollo, debería ser el interface con los termómetros exteriores, así como con otras placas.

Software

La primera propuesta sería la creación de una DLL (Dinamic Link Library), que permite la ejecución de los comandos a bajo nivel, dejando a los programas su uso mediante una simple llamada a la librería.

El último paso del Software sería la integración con el programa de medida, para no sólo realizar medidas únicas de varios módulos, sino que también poder programar medidas periódicas.

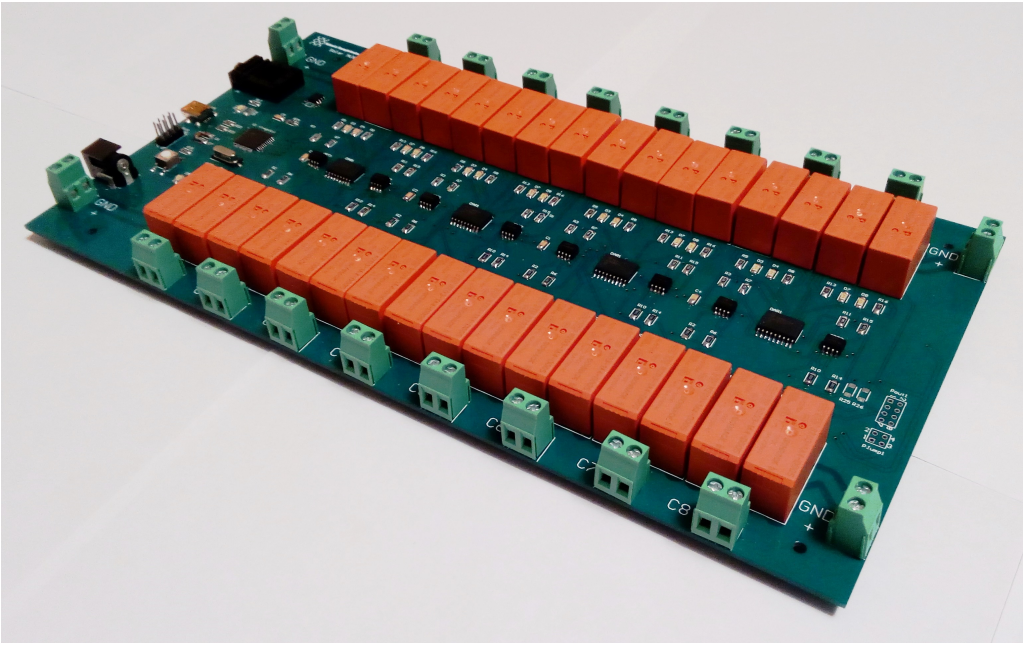


Figura 5.1: Solar MUX

Bibliografía

[1] Datasheet KEITHLEY 7702

40-channel Differential Multiplexer Module with Screw Terminals

http://www.tek.com/sites/tek.com/files/media/media/resources/7702.pdf

[2] Datasheet NI PXI-2527

32-Channel (2-Wire), 300 V Multiplexer Switches

http://www.ni.com/pdf/products/us/cat_pxie2527.pdf

[3] Datasheet RZ03-1A4-D005

Relé sin enclavamiento, SPST-NA, Montaje Pasante, 16 A, 5V dc

http://es.rs-online.com/web/p/products/7917362/

[4] Datasheet ULN2803ADW

Array de transistor Darlington, ULN2803ADW, NPN 0,5 A, 50 V, Octal, , SOIC, 18 pines Octal

http://www.ti.com/lit/ds/symlink/uln2803a.pdf

[5] Datasheet Atmega32u4

8-bit Microcontroller with 16/32K bytes of ISP Flash and USB Controller http://www.atmel.com/Images/Atmel-7766-8-bit-AVR-ATmega16U4-32U4_Datasheet.pdf

[6] Circuito Atmega

Probando el ATmega(32/16)u4 http://webdelcire.com/wordpress/archives/269

[7] Datasheet VOD207T

Optocoupler, Phototransistor Output, Dual Channel, SOIC-8 Package http://www.vishay.com/docs/81956/vod205t.pdf

[8] Datasheet DS18B20

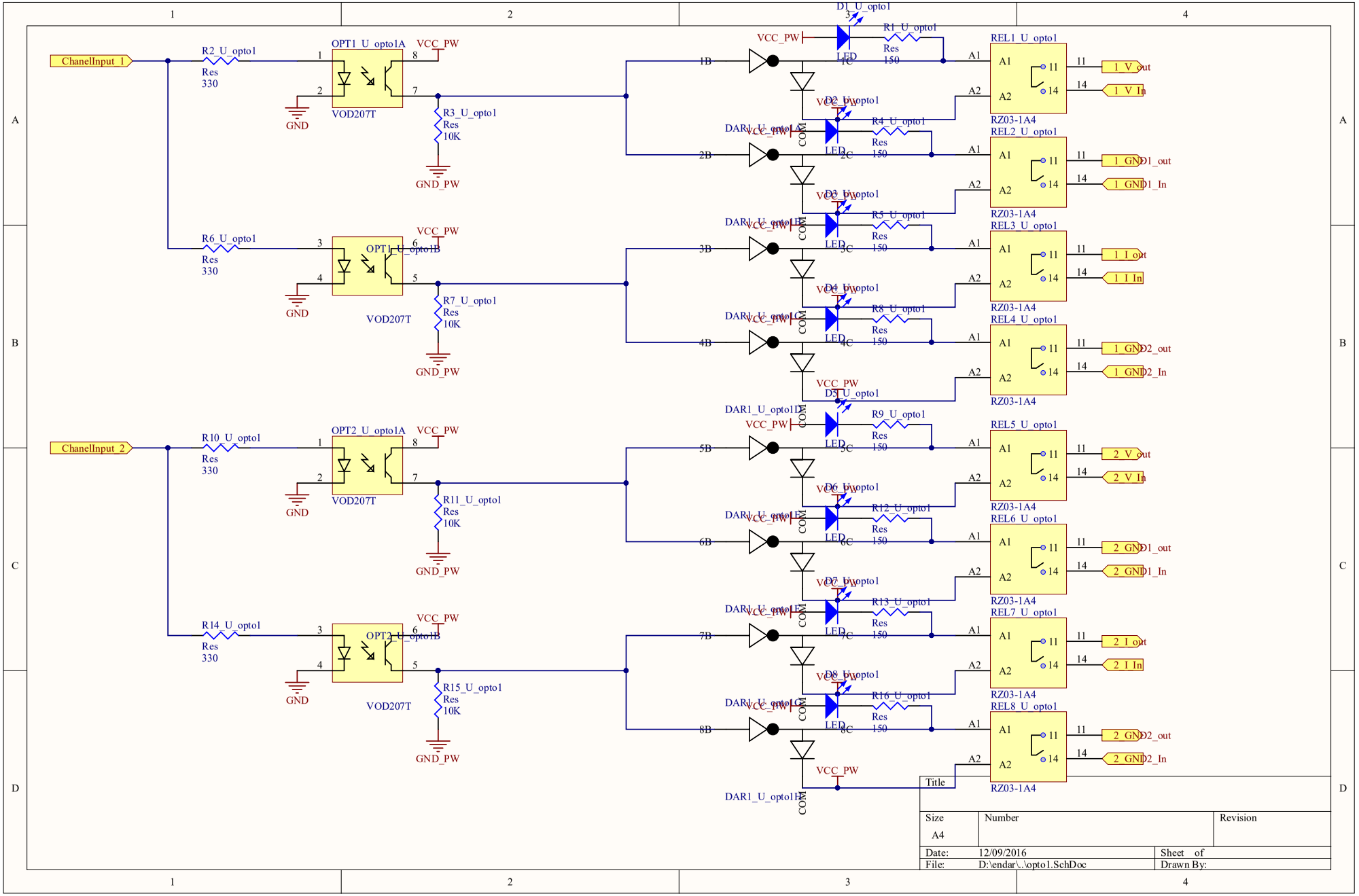
Programmable Resolution 1-Wire Digital Thermometer https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf

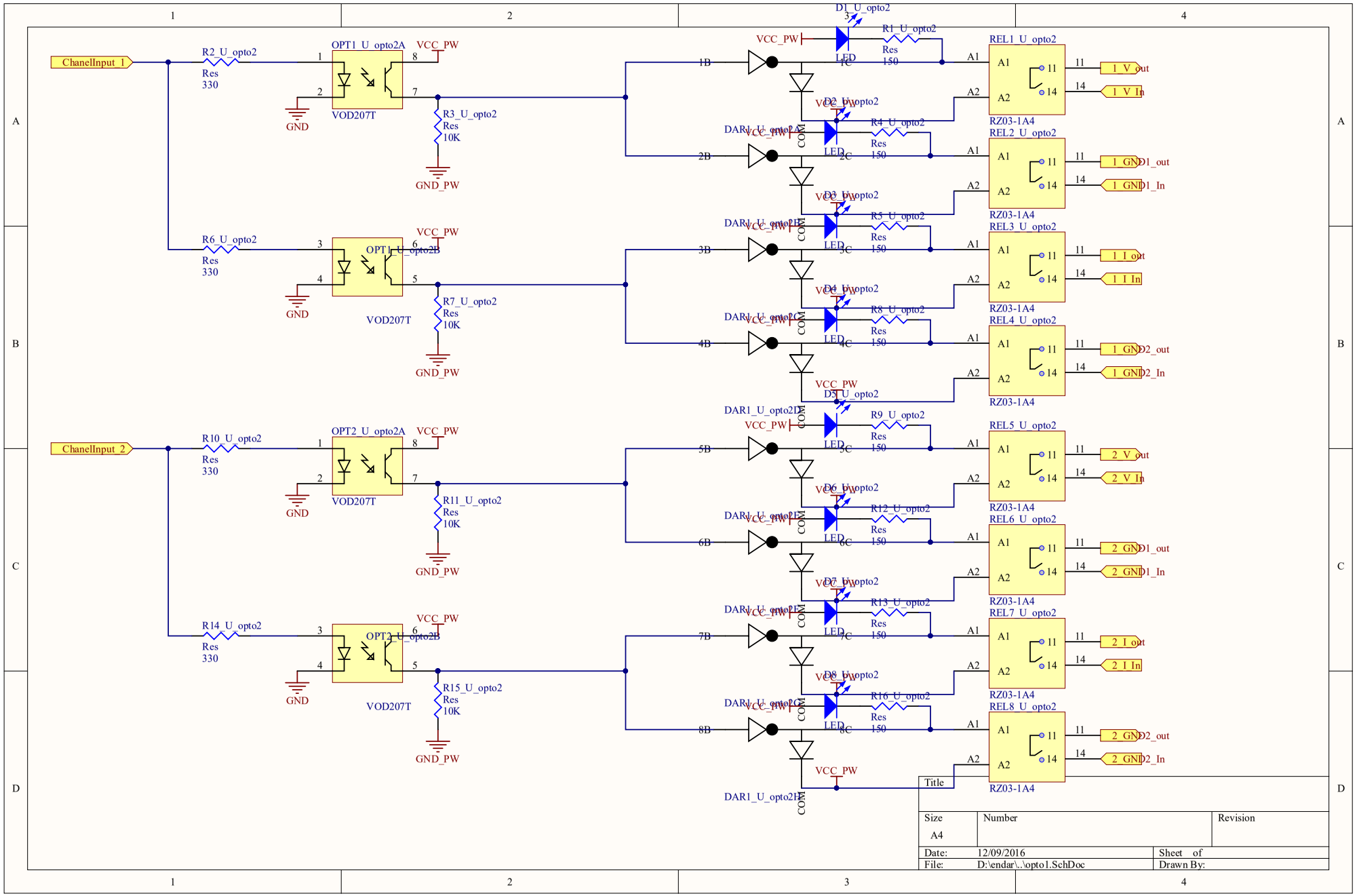
[9] Soldador AOYUE 968A+

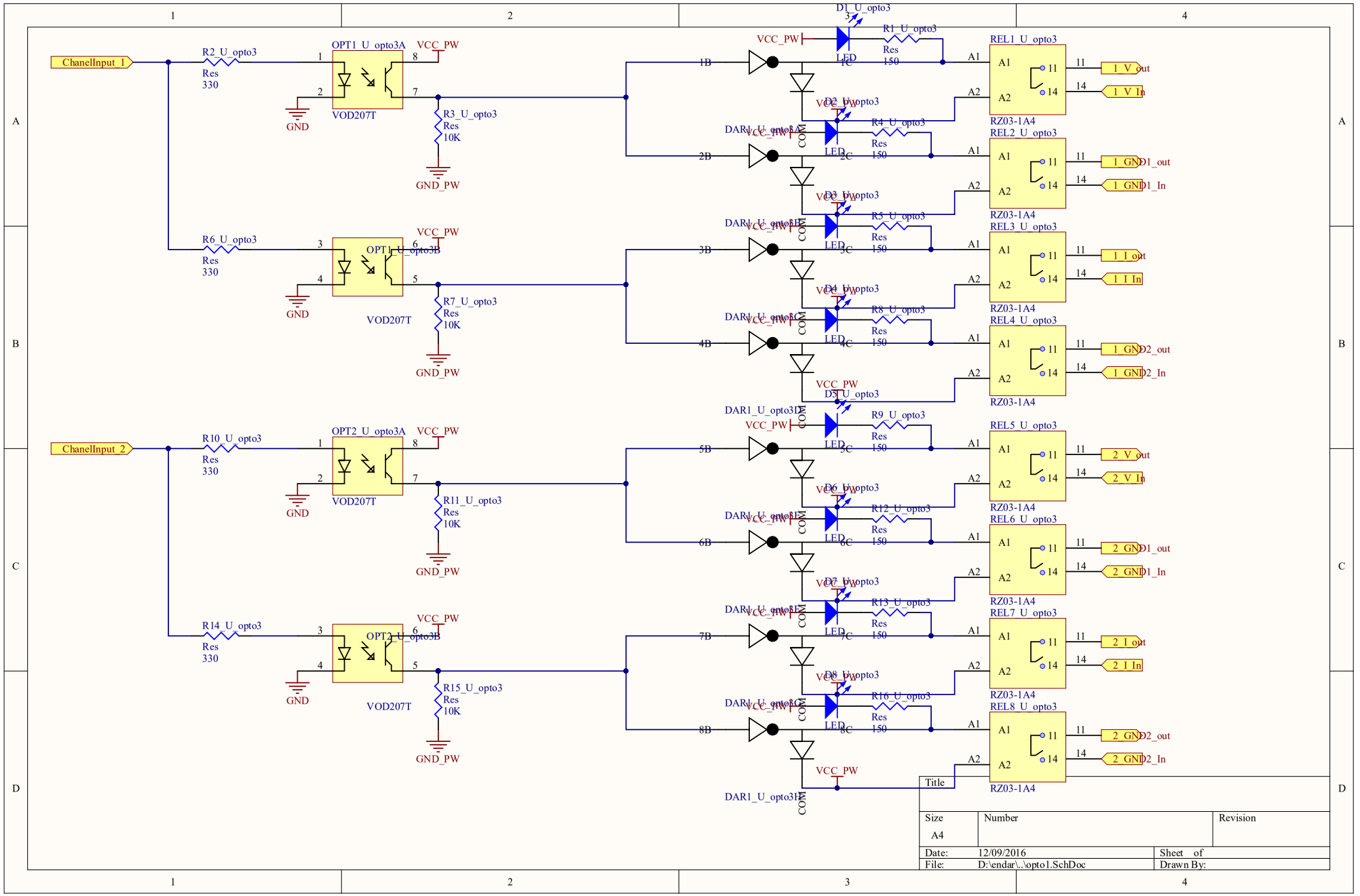
http://www.aoyue.com/es/pro/?id=16

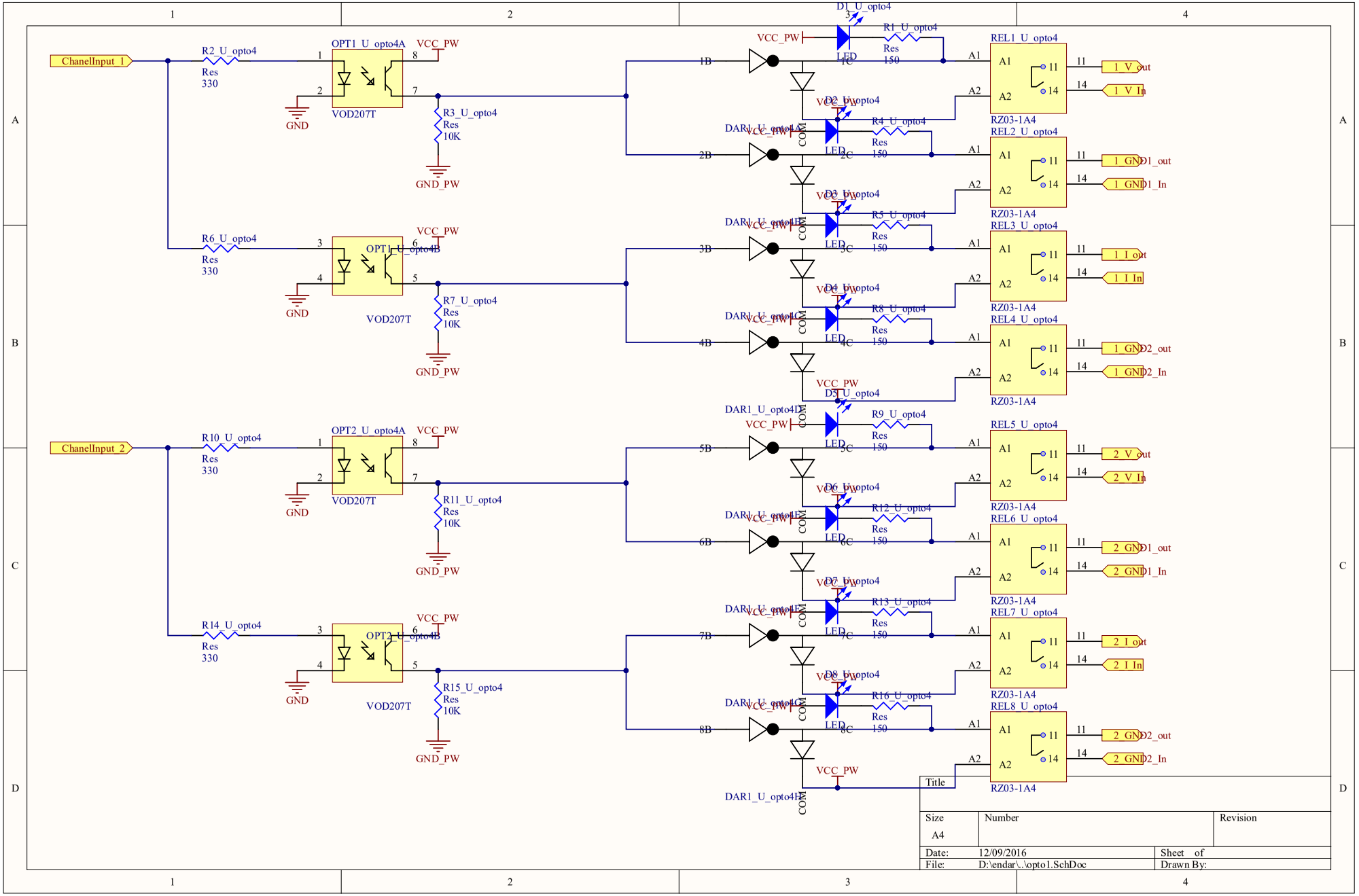
APÉNDICE A

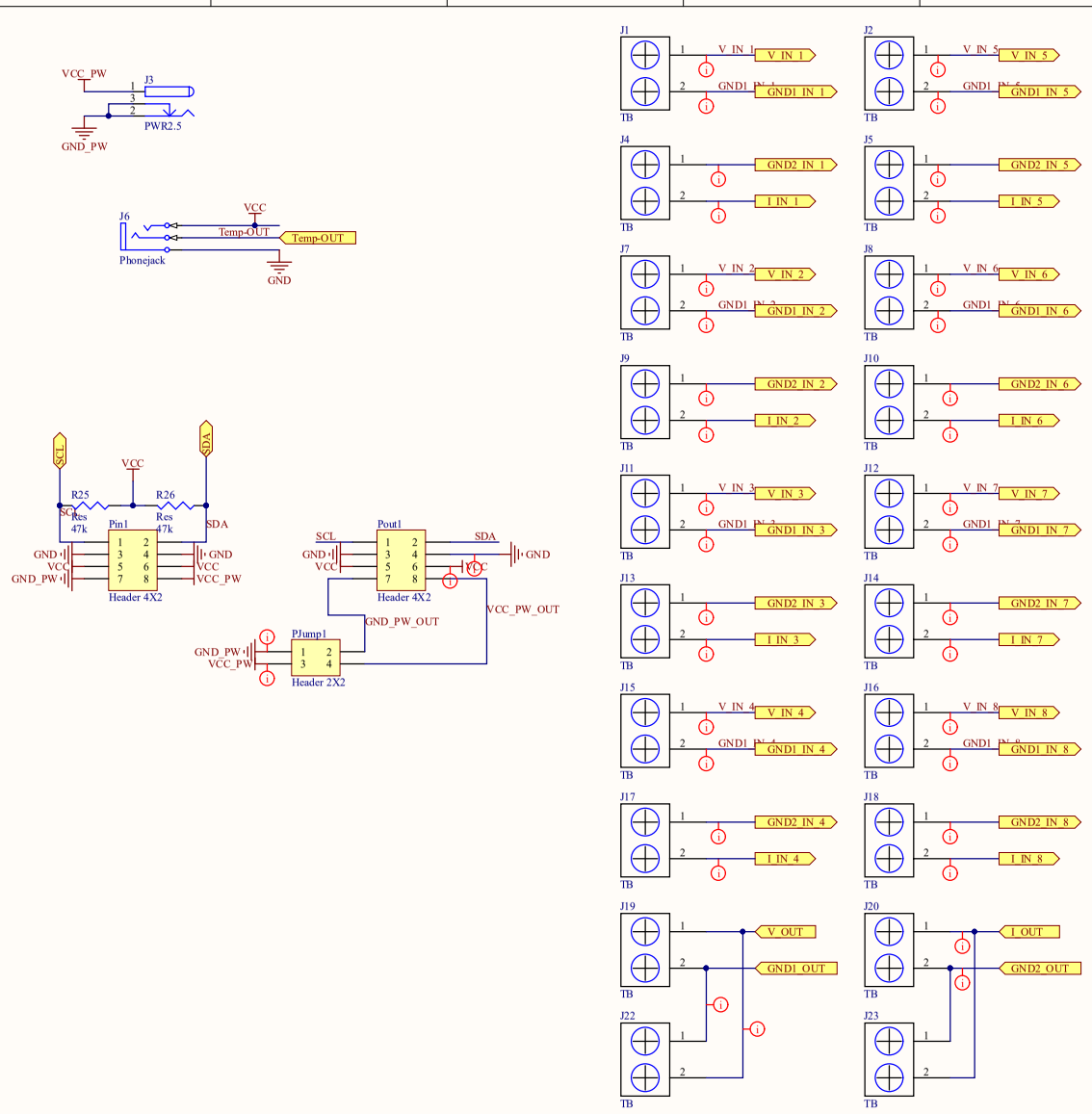
Esquemáticos Altium


ChanelInput_1
R2_U_opto1
Res
330
OPT1_U_opto1A
VOD207T
VCC_PW
GND
R3_U_opto1
Res
10K
GND_PW
R6_U_opto1
OPT1_U_opto1B
D1_U_opto1
R1_U_opto1
LED
REL1_U_opto1
RZ03-1A4
1_V_out
1_V_In
D2_U_opto1
R4_U_opto1
REL2_U_opto1
1_GND1_out
1_GND1_In
D3_U_opto1
R5_U_opto1
REL3_U_opto1
1_I_out
1_I_In
D4_U_opto1
R8_U_opto1
REL4_U_opto1
1_GND2_out
1_GND2_In
D5_U_opto1
R9_U_opto1
REL5_U_opto1
2_V_out
2_V_In
D6_U_opto1
R12_U_opto1
REL6_U_opto1
2_GND1_out
2_GND1_In
D7_U_opto1
R13_U_opto1
REL7_U_opto1
2_I_out
2_I_In
D8_U_opto1
R16_U_opto1
REL8_U_opto1
2_GND2_out
2_GND2_In
ChanelInput_2
R10_U_opto1
OPT2_U_opto1A
R11_U_opto1
R14_U_opto1
OPT2_U_opto1B
R15_U_opto1
DAR1_U_opto1A
DAR1_U_opto1B
DAR1_U_opto1C
DAR1_U_opto1D
DAR1_U_opto1E
DAR1_U_opto1F
DAR1_U_opto1G
DAR1_U_opto1H
COM
Title
Size
A4
Number
Revision
Date:
12/09/2016
Sheet of
File:
D:\endar\..\opto1.SchDoc
Drawn By:

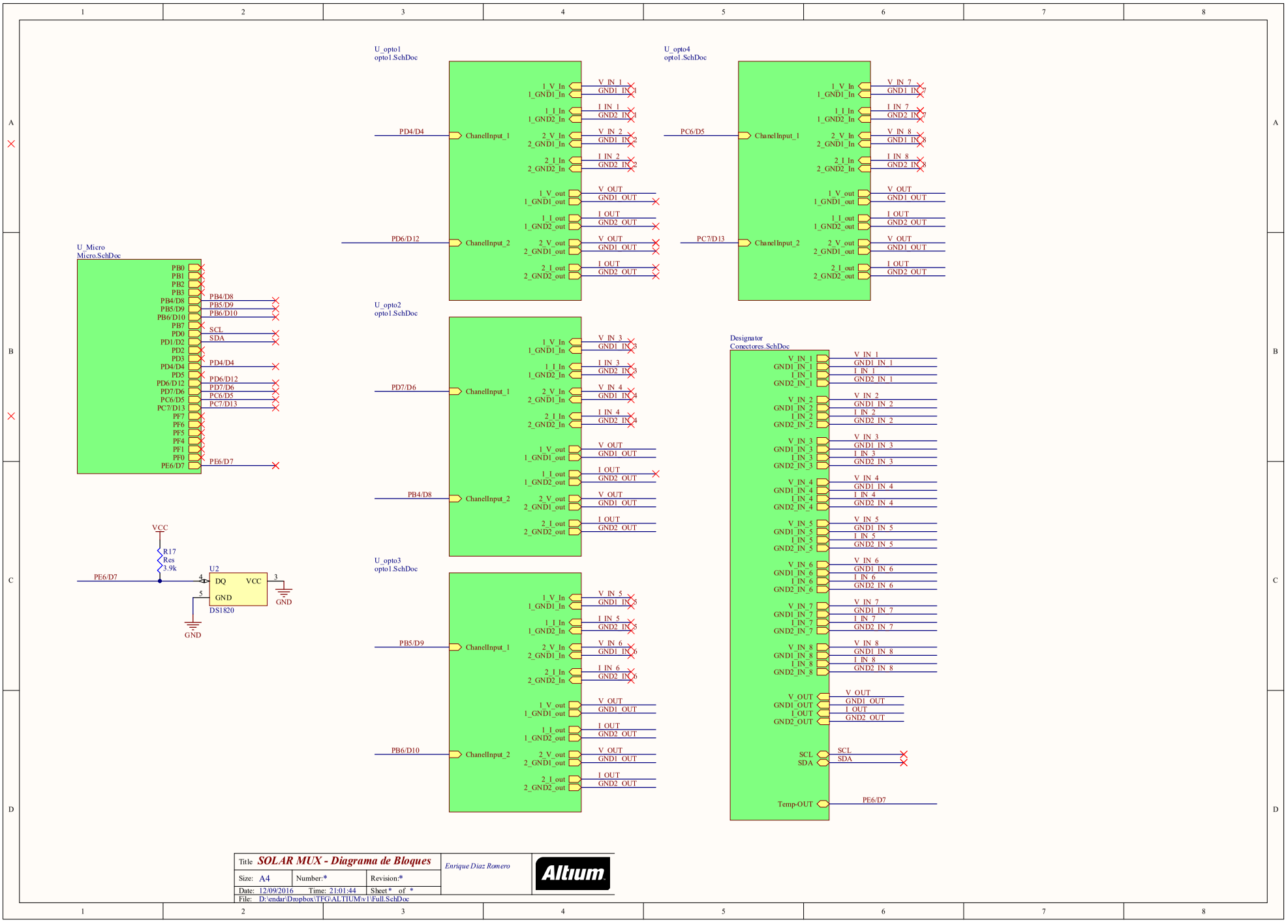

ChanelInput_1
R2_U_opto2
Res
330
OPT1_U_opto2A
VCC_PW
VOD207T
GND
R3_U_opto2
Res
10K
GND_PW
R6_U_opto2
OPT1_U_opto2B
R7_U_opto2
ChanelInput_2
R10_U_opto2
OPT2_U_opto2A
R11_U_opto2
R14_U_opto2
OPT2_U_opto2B
R15_U_opto2
D1_U_opto2
R1_U_opto2
LED
DAR1_U_opto2A
COM
D2_U_opto2
R4_U_opto2
DAR1_U_opto2B
D3_U_opto2
R5_U_opto2
DAR1_U_opto2C
D4_U_opto2
R8_U_opto2
DAR1_U_opto2D
D5_U_opto2
R9_U_opto2
DAR1_U_opto2E
D6_U_opto2
R12_U_opto2
DAR1_U_opto2F
D7_U_opto2
R13_U_opto2
DAR1_U_opto2G
D8_U_opto2
R16_U_opto2
DAR1_U_opto2H
REL1_U_opto2
REL2_U_opto2
REL3_U_opto2
REL4_U_opto2
REL5_U_opto2
REL6_U_opto2
REL7_U_opto2
REL8_U_opto2
RZ03-1A4
1_V_out
1_V_In
1_GND1_out
1_GND1_In
1_I_out
1_I_In
1_GND2_out
1_GND2_In
2_V_out
2_V_In
2_GND1_out
2_GND1_In
2_I_out
2_I_In
2_GND2_out
2_GND2_In
Title
Size
A4
Number
Revision
Date:
12/09/2016
Sheet of
File:
D:\endar\..\opto1.SchDoc
Drawn By:


ChanelInput_1
R2_U_opto3
Res
330
OPT1_U_opto3A
VOD207T
VCC_PW
GND
R3_U_opto3
Res
10K
GND_PW
R6_U_opto3
OPT1_U_opto3B
D1_U_opto3
R1_U_opto3
LED
REL1_U_opto3
RZ03-1A4
1_V_out
1_V_In
D2_U_opto3
R4_U_opto3
REL2_U_opto3
1_GND1_out
1_GND1_In
D3_U_opto3
R5_U_opto3
REL3_U_opto3
1_I_out
1_I_In
D4_U_opto3
R8_U_opto3
REL4_U_opto3
1_GND2_out
1_GND2_In
D5_U_opto3
R9_U_opto3
ChanelInput_2
R10_U_opto3
OPT2_U_opto3A
R11_U_opto3
R12_U_opto3
REL5_U_opto3
2_V_out
2_V_In
REL6_U_opto3
2_GND1_out
2_GND1_In
D7_U_opto3
R13_U_opto3
REL7_U_opto3
2_I_out
2_I_In
R14_U_opto3
OPT2_U_opto3B
R15_U_opto3
D8_U_opto3
R16_U_opto3
REL8_U_opto3
2_GND2_out
2_GND2_In
DAR1_U_opto3A
DAR1_U_opto3B
DAR1_U_opto3C
DAR1_U_opto3D
DAR1_U_opto3E
DAR1_U_opto3F
DAR1_U_opto3G
DAR1_U_opto3H
COM
Title
Size
A4
Number
Revision
Date:
12/09/2016
Sheet of
File:
D:\endar\..\opto1.SchDoc
Drawn By:

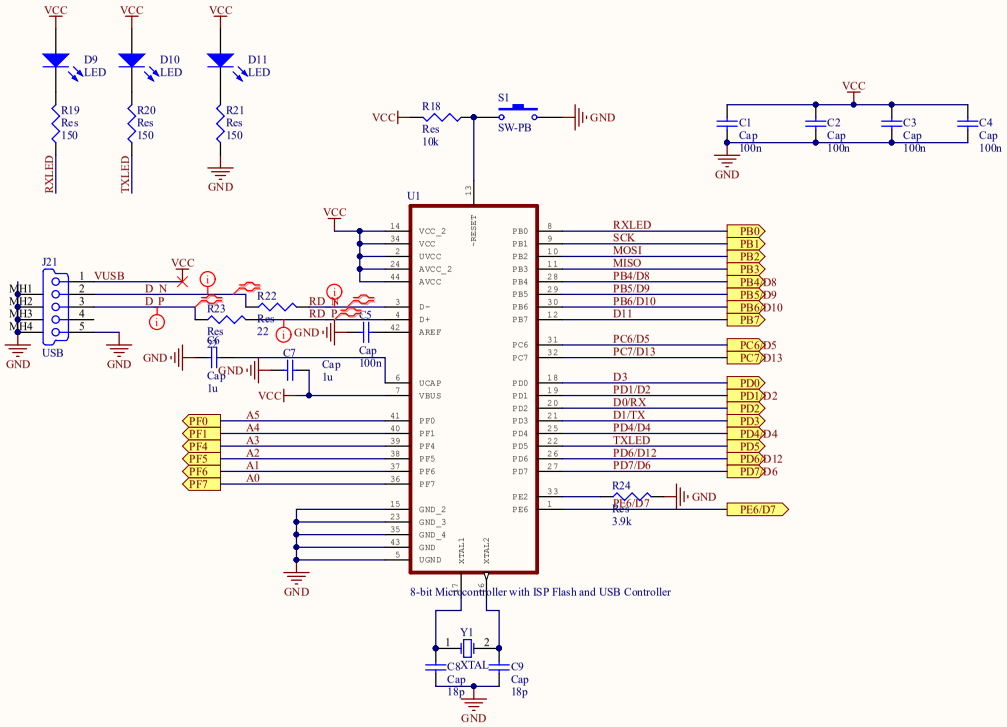

ChanelInput_1
R2_U_opto4
Res
330
OPT1_U_opto4A
VCC_PW
VOD207T
GND
R3_U_opto4
Res
10K
GND_PW
R6_U_opto4
OPT1_U_opto4B
R7_U_opto4
ChanelInput_2
R10_U_opto4
OPT2_U_opto4A
R11_U_opto4
R14_U_opto4
OPT2_U_opto4B
R15_U_opto4
D1_U_opto4
R1_U_opto4
LED
150
REL1_U_opto4
RZ03-1A4
1_V_out
1_V_In
D2_U_opto4
R4_U_opto4
DAR1_U_opto4A
COM
REL2_U_opto4
1_GND1_out
1_GND1_In
D3_U_opto4
R5_U_opto4
DAR1_U_opto4B
REL3_U_opto4
1_I_out
1_I_In
D4_U_opto4
R8_U_opto4
DAR1_U_opto4C
REL4_U_opto4
1_GND2_out
1_GND2_In
D5_U_opto4
DAR1_U_opto4D
R9_U_opto4
REL5_U_opto4
2_V_out
2_V_In
D6_U_opto4
R12_U_opto4
DAR1_U_opto4E
REL6_U_opto4
2_GND1_out
2_GND1_In
D7_U_opto4
R13_U_opto4
DAR1_U_opto4F
REL7_U_opto4
2_I_out
2_I_In
D8_U_opto4
R16_U_opto4
DAR1_U_opto4G
REL8_U_opto4
2_GND2_out
2_GND2_In
DAR1_U_opto4H
Title
Size
A4
Number
Revision
Date:
12/09/2016
Sheet of
File:
D:\endar\..\opto1.SchDoc
Drawn By:



Title	**Conectores**		*Enrique Diaz Romero*
Size: A4	Number:*	Revision:*	
Date: 12/09/2016	Time: 21:01:44	Sheet * of *	
File: D:\endar\Dropbox\TFG\ALTIUM\v1\Conectores.SchDoc			



Title	**SOLAR MUX - Diagrama de Bloques**		*Enrique Diaz Romero*
Size: A4	Number:*	Revision:*	
Date: 12/09/2016	Time: 21:01:44	Sheet * of *	
File: D:\endar\Dropbox\TFG\ALTIUM\v1\Full.SchDoc			



Title *Atmega32U4*			*Enrique Diaz Romero*
Size: A4	Number:*	Revision:*	
Date: 12/09/2016	Time: 21:01:45	Sheet * of *	
File: D:\endar\Dropbox\TFG\ALTIUM\v1\Micro.SchDoc			

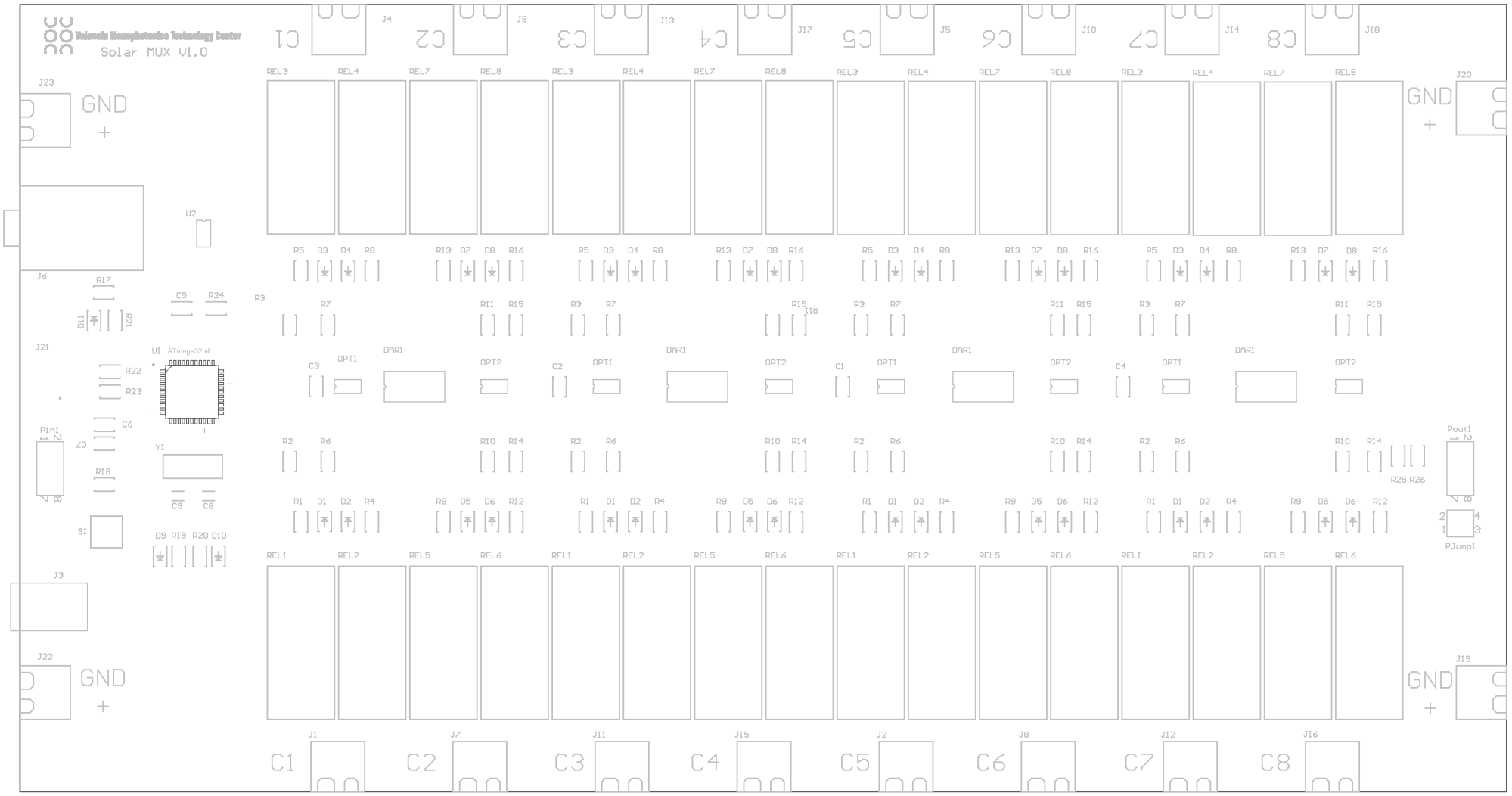
APÉNDICE B
Firmware Prototipo

```
#include <DallasTemperature.h>
#include <OneWire.h>
#include <FlexiTimer2.h>



//IO PIN
#define TEMP_PIN 3
// Entradas Reles
#define PUSH_BUTTON 2
// Salidas Reles
#define C1 7
#define C2 8
#define CONTROL_LED 13




//Configuración del termometro
OneWire oneWire(TEMP_PIN);
DallasTemperature sensor(&oneWire);
//Dirección del sensor
DeviceAddress DIR1 = {0x28, 0x5E, 0x3B, 0x07, 0x08, 0x00, 0x00, 0x74};  //
    Sensor Temperatura

// Variable control del estado del rele
int lastState=0;

void setup() {

   //Puerto serie a 57600
  Serial.begin(57600);

  //Configuración de los pines

  pinMode(PUSH_BUTTON, INPUT);
  pinMode(C1, OUTPUT);
  pinMode(C2, OUTPUT);
  pinMode(CONTROL_LED, OUTPUT);

    sensor.begin();  //Iniciamos los sensores
    //Establecemos la resolucion para cada sensor, PRECISION es a 9 bits
  sensor.setResolution(DIR1, 12);  //Resolucion a 9 bits 0.50 ºC

  FlexiTimer2::set(1000, int_temp);
  FlexiTimer2::start();

  attachInterrupt(digitalPinToInterrupt(PUSH_BUTTON), sw, CHANGE);
```

```
48
49 }
50
51 void loop() {
52
53 }
54
55 void sw(){
56     // read the input pin:
57   int buttonState = digitalRead(PUSH_BUTTON);
58   // print out the state of the button:
59   if(lastState!=buttonState){
60   if (buttonState == 1) {
61     digitalWrite(C2, LOW);
62     digitalWrite(C1, HIGH);
63     digitalWrite(CONTROL_LED,HIGH);
64
65   } else {
66     digitalWrite(C1, LOW);
67     digitalWrite(C2, HIGH);
68       digitalWrite(CONTROL_LED,LOW);
69   }
70   lastState=buttonState;
71
72   }
73
74 }
75
76 void int_temp()
77 {
78   //Pedimos Temperaturas
79   sensor.requestTemperatures();
80   Mostrar_Temperatura(DIR1);
81   Serial.println();
82 }
83
84
85
86 void Mostrar_Direccion(DeviceAddress direccion)
87 {
88   for (uint8_t i = 0; i < 8; i++)
89   {
90     if (direccion[i] < 16)
91       Serial.print("0");
92     Serial.print(direccion[i], HEX);
93   }
94 }
95
96 //Funcion que muestra la temperatura en grados centigrados del sensor
97 void Mostrar_Temperatura(DeviceAddress direccion)
98 {
99   float tempC = sensor.getTempC(direccion);
100 //  Serial.print("Temperatura: ");
101   Serial.print(tempC);
102   Serial.println();
103 }
104
105 void Mostrar_Resolucion(DeviceAddress direccion)
106 {
107   Serial.print("Resolucion: ");
108   Serial.print(sensor.getResolution(direccion));
109   Serial.println();
110 }
```

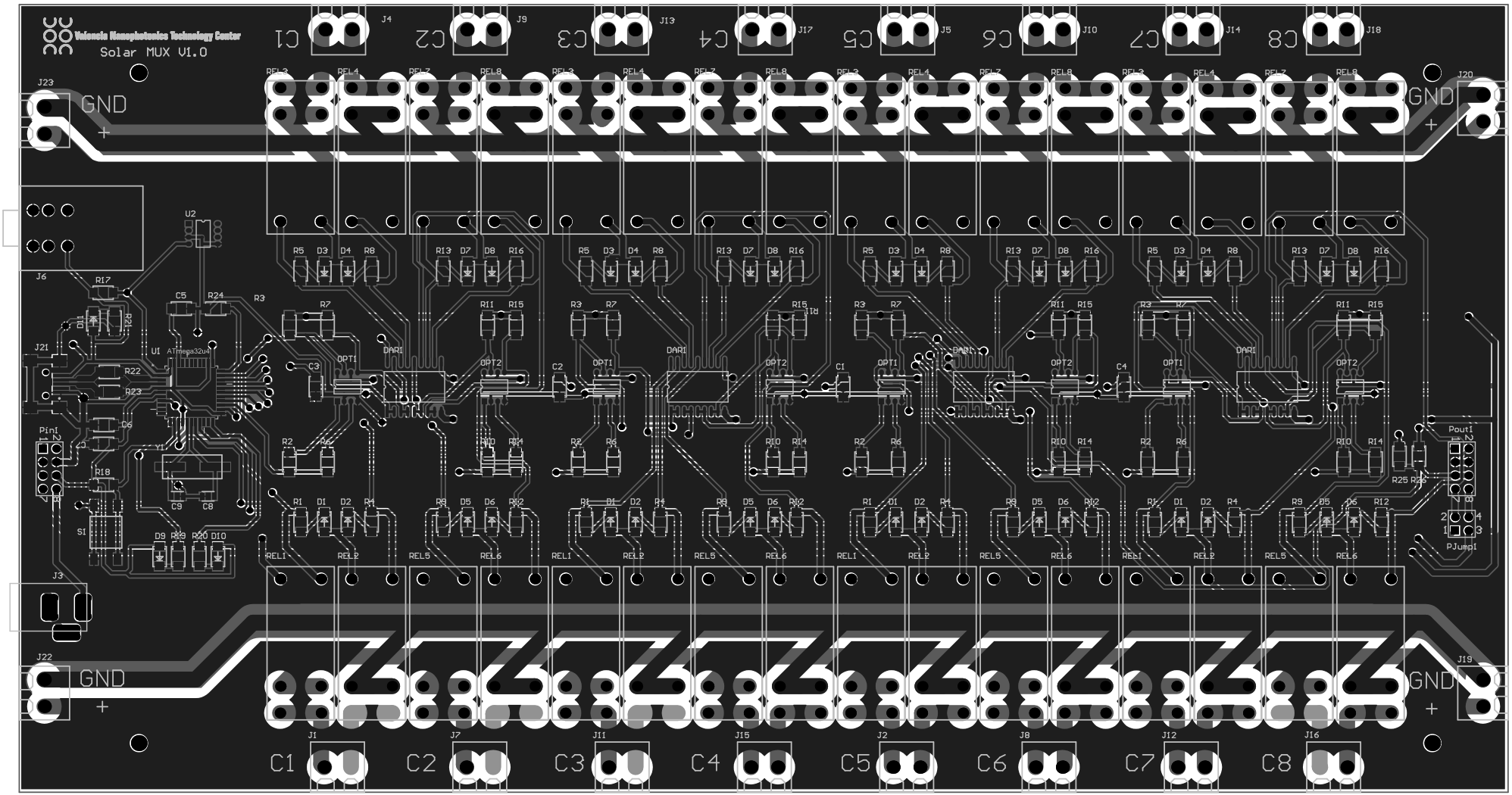
APÉNDICE C

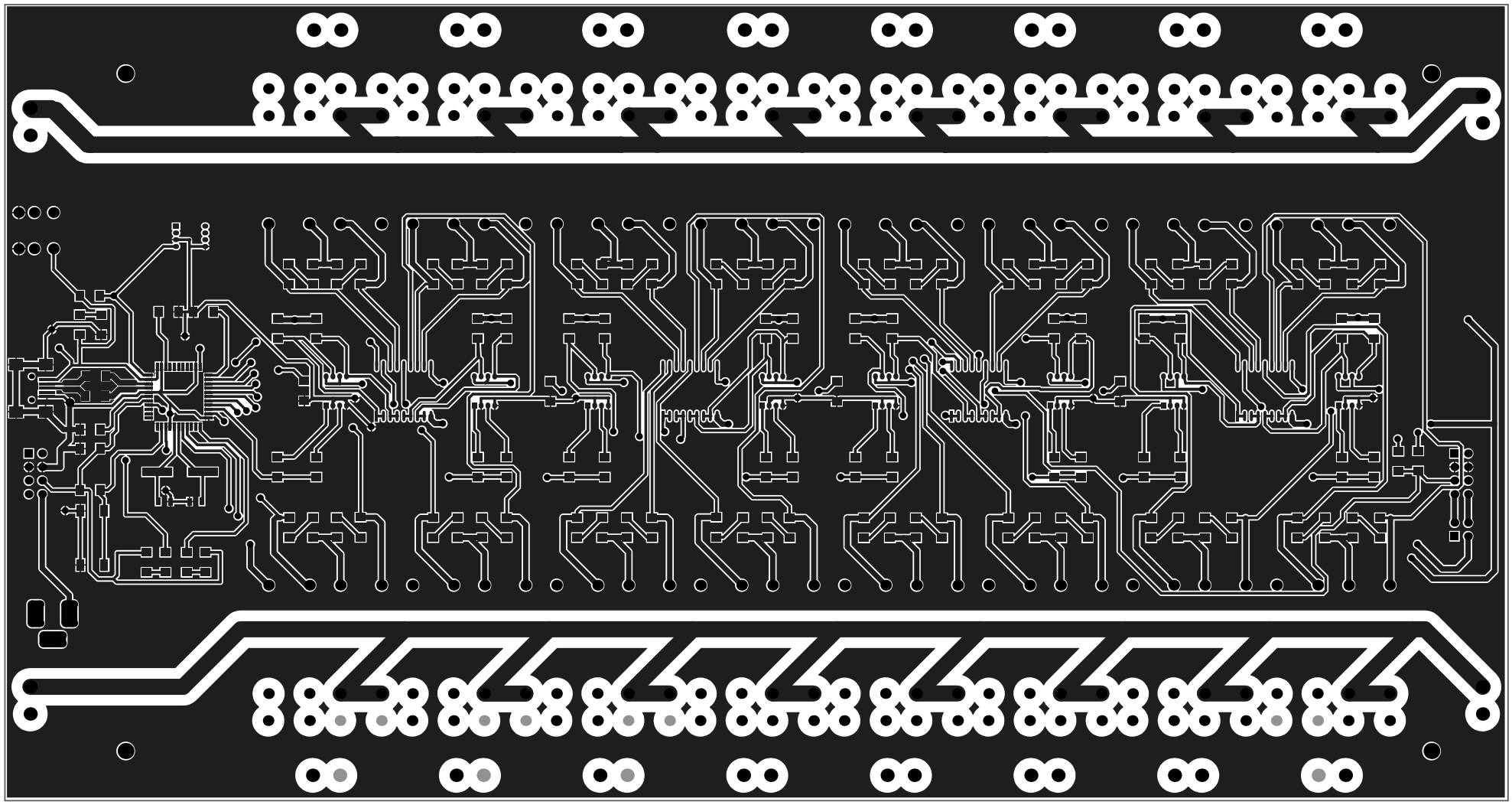
Placement Altium

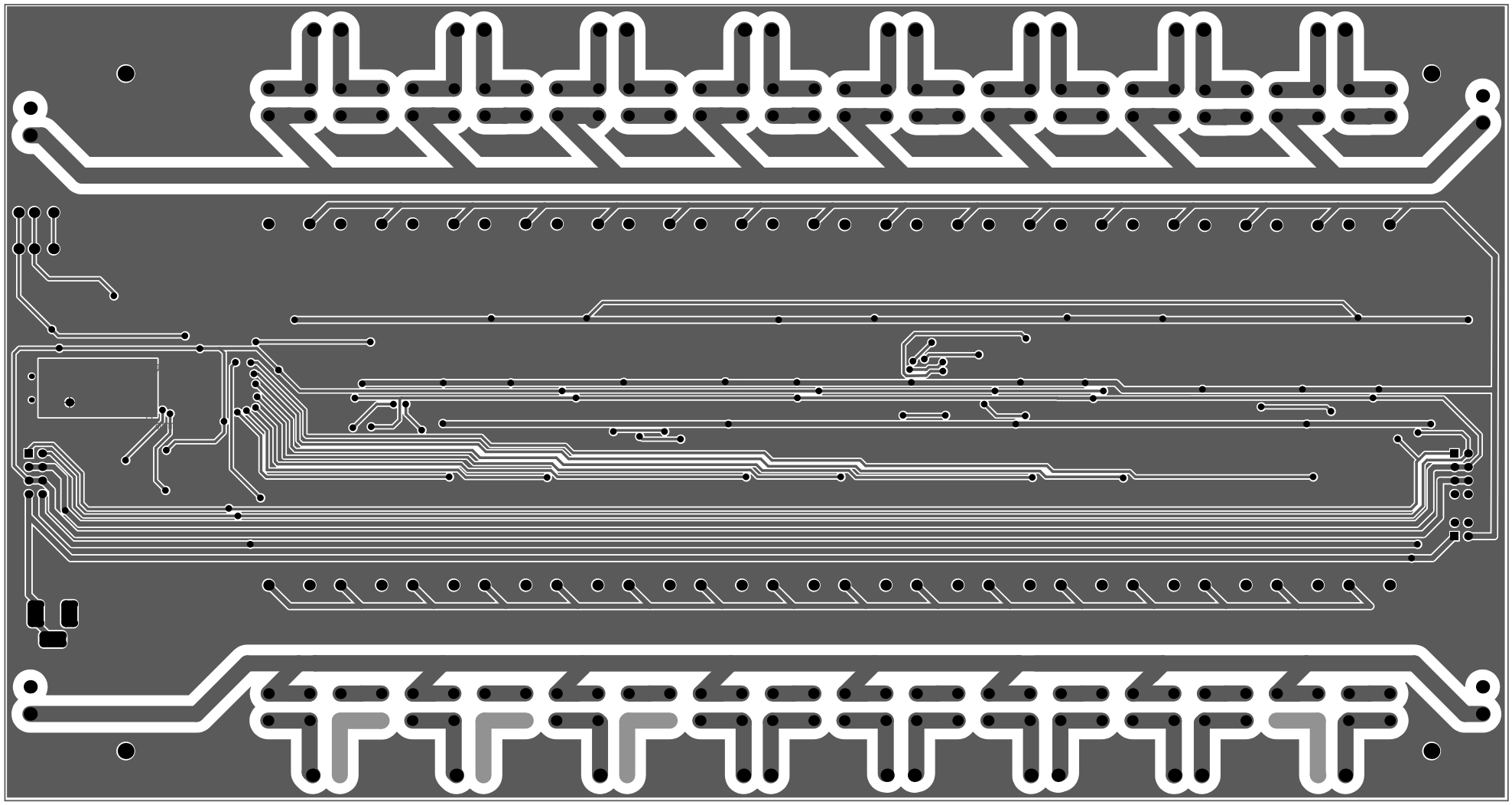

Solar MUX V1.0
GND
GND
GND
GND
U1 ATmega32u4
Pin1
Pout1
PJumpI
C1
C2
C3
C4
C5
C6
C7
C8

APÉNDICE D

Rutado Altium

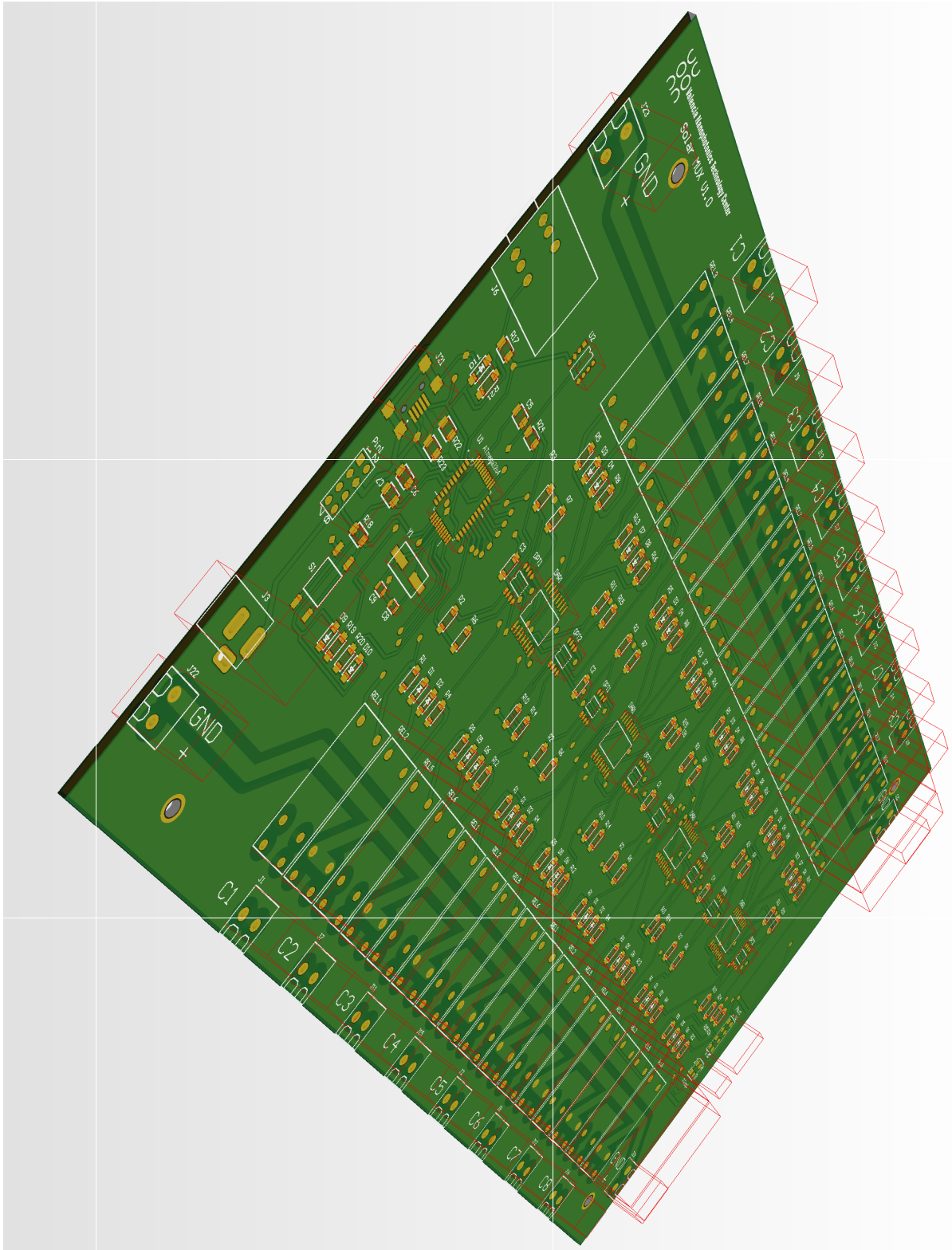

Valencia Nanophotonics Technology Center
Solar MUX V1.0
GND
GND
GND
GND
U2
ATmega32u4
C1
C2
C3
C4
C5
C6
C7
C8

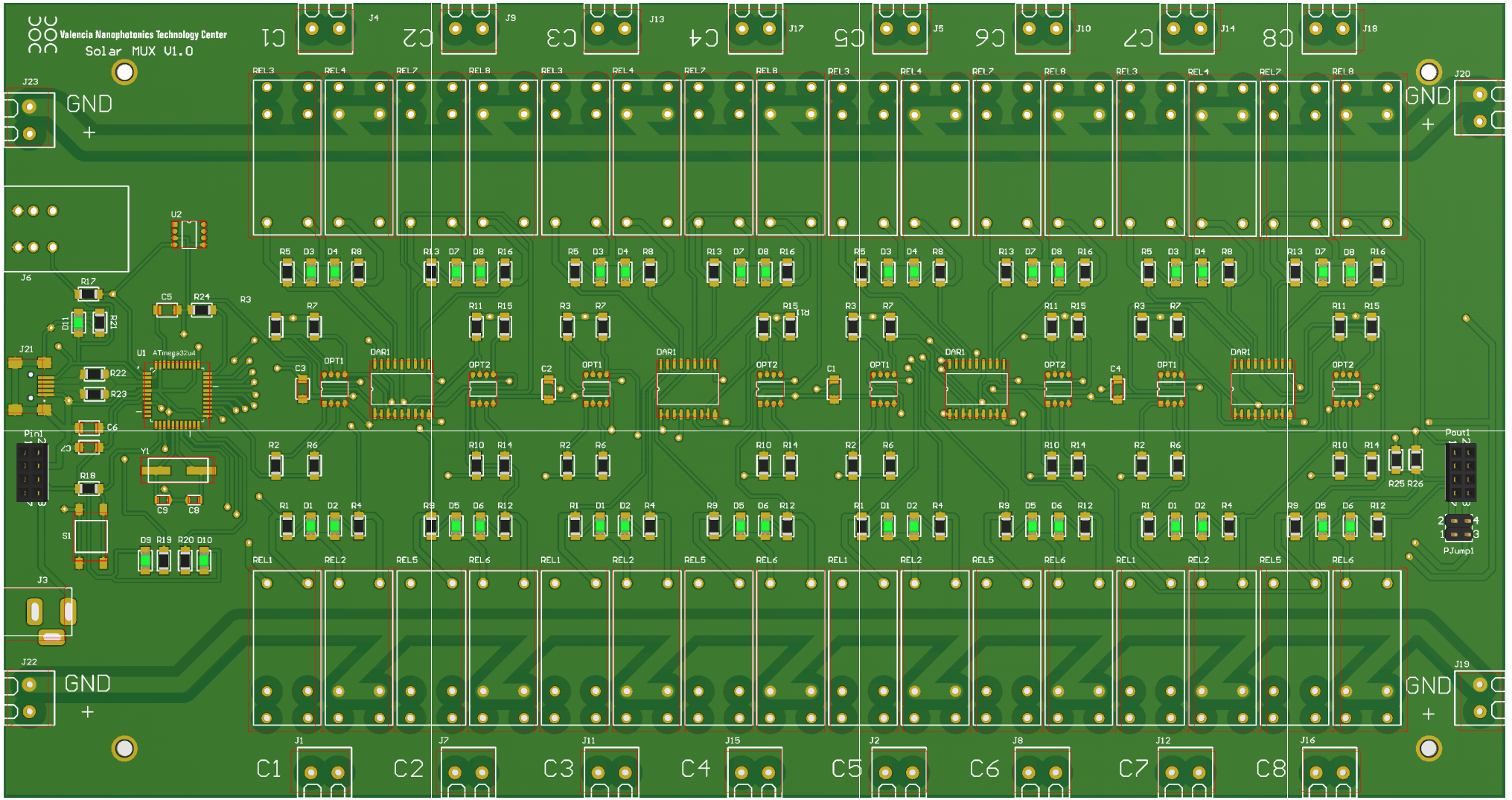


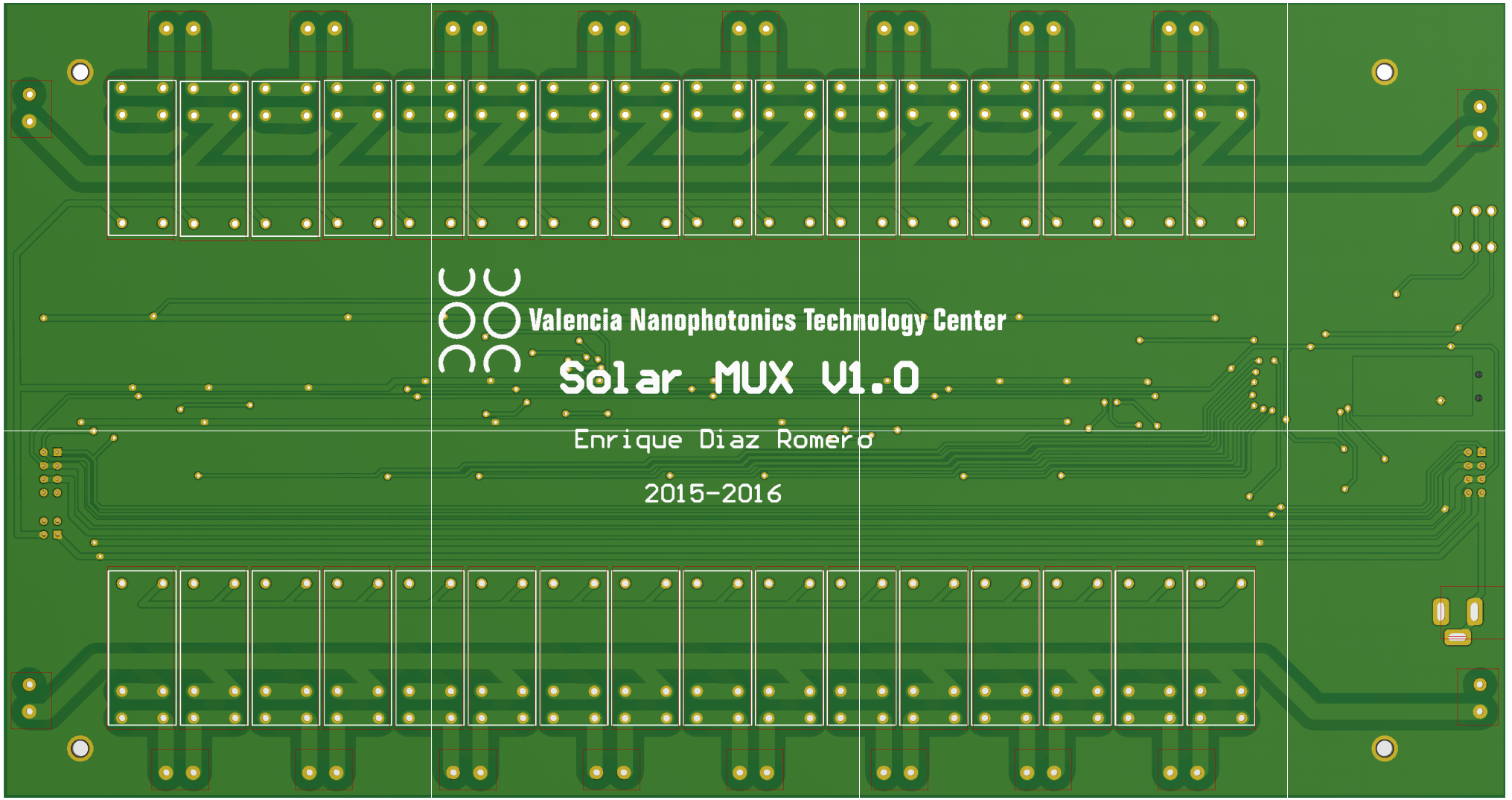


APÉNDICE E

3D Altium


GND
+
GND
+


Valencia Nanophotonics Technology Center
Solar MUX V1.0
GND
GND
GND
GND
U1 ATmega32u4
U2
Pin1
Pout1
PJump1
C1
C2
C3
C4
C5
C6
C7
C8


Valencia Nanophotonics Technology Center
Solar MUX V1.0
Enrique Diaz Romero
2015-2016

APÉNDICE F
Protocolo Serie

```
#if defined(ARDUINO) && ARDUINO >= 100
#include "Arduino.h"
#else
#include "WProgram.h"
#endif
#include "Streaming.h"
#ifndef SerialReceiver_h
#define SerialReceiver_h

enum {SR_MAX_ITEM_SZ = 25};
enum {SR_MAX_ITEMS = 4};

const char SR_DFLT_START_CHAR = '[';
const char SR_DFLT_STOP_CHAR = ']';
const char SR_DFLT_SEP_CHAR = ';';

const uint8_t SR_STATE_IDLE = 0;
const uint8_t SR_STATE_RECEIVING = 1;
const uint8_t SR_STATE_MESSAGE = 2;

const uint8_t SR_ERR_NONE = 0;
const uint8_t SR_ERR_ILLEGAL_CHAR = 1;
const uint8_t SR_ERR_ITEM_LENGTH = 2;
const uint8_t SR_ERR_MESSAGE_LENGTH = 3;

class SerialReceiver {

    public:
        SerialReceiver();
        void process(int serialByte);
        bool messageReady();
        void reset();
        uint8_t numberOfItems();
        uint8_t itemLength(uint8_t itemNum);
        char readChar(uint8_t itemNum, uint8_t ind);
        int readInt(uint8_t itemNum);
        long readLong(uint8_t itemNum);
        float readFloat(uint8_t itemNum);
        double readDouble(uint8_t itemNum);
        void copyString(uint8_t itemNum, char *string, uint8_t size);
        void printInfo();
        void printMessageInfo();
        void printMessage();

    private:
        uint8_t state;
        uint8_t error;
        char itemBuffer[SR_MAX_ITEMS][SR_MAX_ITEM_SZ+1];
```

```
49          uint8_t itemLenBuffer[SR_MAX_ITEMS];
50          uint8_t itemCnt;
51          uint8_t itemPos;
52          char startChar;
53          char stopChar;
54          char sepChar;
55          void resetItems();
56          void resetState();
57          void processNewMsg(int serialByte);
58          void processCurMsg(int serialByte);
59          void handleNewChar(int serialByte);
60          void handleSepChar(int serialByte);
61          void handleStopChar(int serialByte);
62          void handleStartChar(int serialByte);
63          bool checkItemRange(uint8_t itemNum);
64  };
65
66
67  #endif
```

APÉNDICE G
DS18B20

```
#ifndef _DS18B20_H_
#define _DS18B20_H_
#if defined(ARDUINO) && ARDUINO >= 100
#include "Arduino.h"
#else
#include "WProgram.h"
#endif

#include "io_pins.h"
#include <OneWire.h>
#include <DallasTemperature.h>



class DS18B20
{



public:
 DS18B20();
 float Measure();

};


#endif
```

APÉNDICE H
SolarMUX

```
#ifndef _SOLARMUX_H_
#define _SOLARMUX_H_
#if defined(ARDUINO) && ARDUINO >= 100
#include "Arduino.h"
#else
#include "WProgram.h"
#endif

#include "io_pins.h"


class SolarMUX
{
  private:
 const int port[8]={4,12,6,8,9,10,5,13};
  int activo;
  void clc();
  void set(int);
  int gtc();

public:
 SolarMUX();
void getVersion(int& major,int& minor,int& patch);
void setChannel(int ch);
int getChannel();
void clcChannel();
};



#endif
```

APÉNDICE I
SerialHandler

```
#ifndef _SERIAL_HANDLER_H_
#define _SERIAL_HANDLER_H_
#include "SerialReceiver.h"
#include "SolarMUX.h"
#include "Streaming.h"
#include "DS18B20.h"

class SerialHandler: public SerialReceiver
{
  public:
    void processInput();
  private:
    void _switchYard();
    void _getVersion();
    void _sendHelp();
    void _setChannel();
    void _getChannel();
    void _nextChannel();
    void _prevChannel();
    void _unknownCmd();
    void _clcChannel();
    void _getTemp();
};

extern SolarMUX mux;
extern DS18B20 term;

#endif
```

```
#if defined(ARDUINO) && ARDUINO >= 100
#include "Arduino.h"
#else
#include "WProgram.h"
#endif
#include "SerialHandler.h"
#include "Streaming.h"



// Constants
// ------------------------------------------------------------------
const int CMD_HELP=0;
const int CMD_GET_VERSION=1;
const int CMD_SET_CHANNEL= 2;
const int CMD_GET_CHANNEL= 3;
const int CMD_NEXT_CHANNEL=4;
const int CMD_PREV_CHANNEL=5;
```

```
19 const int CMD_CLR_CHANNEL=6;
20 const int CMD_GET_TEMP=7;
21
22
23 const int RSP_ERROR = 0;
24 const int RSP_SUCCESS = 1;
25 // Methods
26 // ------------------------------------------------------------------------
27 void SerialHandler::processInput()
28 {
29   while (Serial.available() > 0)
30   {
31     process(Serial.read());
32     if (messageReady())
33     {
34       _switchYard();
35       reset();
36     }
37   }
38 }
39
40 void SerialHandler::_switchYard()
41 {
42   switch ((uint8_t)readInt(0))
43  {
44
45     case CMD_HELP:              _sendHelp();                    break;
46     case CMD_GET_VERSION:       _getVersion();                  break;
47     case CMD_SET_CHANNEL:       _setChannel();                  break;
48     case CMD_GET_CHANNEL:       _getChannel();                  break;
49     case CMD_NEXT_CHANNEL:      _nextChannel();                 break;
50     case CMD_PREV_CHANNEL:      _prevChannel();                 break;
51     case CMD_CLR_CHANNEL:       _clcChannel();                  break;
52     case CMD_GET_TEMP:          _getTemp();                     break;
53
54     default:                    _unknownCmd();                  break;
55   }
56 }
57
58 void SerialHandler::_unknownCmd()
59 {
60     // un-recognized command. Send error message.
61     Serial << SR_DFLT_START_CHAR << RSP_ERROR;
62     Serial << SR_DFLT_SEP_CHAR << "unknown command";
63     Serial << SR_DFLT_STOP_CHAR << endl;
64 }
65
66 void SerialHandler::_sendHelp()
67 {
68     int major,minor,patch;
69   mux.getVersion(major,minor,patch);
70   Serial << endl;
71   Serial << "------------------------------------------------" << endl;
72   Serial << "------------------------------------------------" << endl;
73   Serial << " -- SolarMUX command line control v" ;
74   Serial << _DEC(major) << ".";
75   Serial << _DEC(minor) << ".";
76   Serial << _DEC(patch) << " -- ";
77   Serial << endl;
78   Serial << "------------------------------------------------" << endl;
79   Serial << "------------------------------------------------" << endl;
80   Serial << endl;
81   Serial << " Usage:" << endl;
82   Serial << "[COMMAND ; ARG1(opt)]" << endl;
```

```
83    Serial << "[ " << (uint8_t)CMD_HELP << " ] This HELP " << endl;
84    Serial << "[ " << (uint8_t)CMD_GET_VERSION << " ] Firmware Version " << endl;
85    Serial << "[ " << (uint8_t)CMD_SET_CHANNEL << " (Set MUX); 1-8 (Channel ID)]"
          << endl;
86    Serial << "[ " << (uint8_t)CMD_GET_CHANNEL << " ] Active Channel " << endl;
87    Serial << "[ " << (uint8_t)CMD_NEXT_CHANNEL << " ] Next Channel " << endl;
88    Serial << "[ " << (uint8_t)CMD_PREV_CHANNEL << " ] Previous Channel " << endl
          ;
89    Serial << "[ " << (uint8_t)CMD_CLR_CHANNEL << " ] Clear Channels " << endl;
90    Serial << "[ " << (uint8_t)CMD_GET_TEMP << " ] Get Temperature " << endl;
91
92
93
94
95  }
96
97  void SerialHandler::_getVersion()
98  {
99    int major,minor,patch;
100   mux.getVersion(major,minor,patch);
101   Serial << SR_DFLT_START_CHAR << RSP_SUCCESS;
102   Serial << SR_DFLT_SEP_CHAR << major;
103   Serial << SR_DFLT_SEP_CHAR << minor;
104   Serial << SR_DFLT_SEP_CHAR << patch;
105   Serial << SR_DFLT_STOP_CHAR << endl;
106 }
107
108 void SerialHandler::_setChannel()
109 {
110   uint16_t channel;
111   channel = readInt(1);
112    if ((channel > 0) && (channel <= 8))
113   {
114     mux.setChannel((uint16_t) channel);
115     Serial << SR_DFLT_START_CHAR << RSP_SUCCESS << SR_DFLT_STOP_CHAR << endl;
116   }
117   else
118   {
119     Serial << SR_DFLT_START_CHAR << RSP_ERROR;
120     Serial << SR_DFLT_SEP_CHAR << "out of range";
121     Serial << SR_DFLT_STOP_CHAR << endl;
122   }
123
124 }
125
126
127 void SerialHandler::_getChannel()
128 {
129     Serial << SR_DFLT_START_CHAR << RSP_SUCCESS;
130   Serial << SR_DFLT_SEP_CHAR << mux.getChannel();
131   Serial << SR_DFLT_STOP_CHAR << endl;
132
133 }
134 void SerialHandler::_nextChannel()
135 {
136   int ch=mux.getChannel();
137   if(ch==8){
138     ch=1;
139   }
140   else{
141     ch++;
142   }
143   mux.setChannel(ch);
144     Serial << SR_DFLT_START_CHAR << RSP_SUCCESS;
```

```
145   Serial << SR_DFLT_SEP_CHAR << ch;
146   Serial << SR_DFLT_STOP_CHAR << endl;
147 }
148 void SerialHandler::_prevChannel()
149 {
150   int ch=mux.getChannel();
151   if(ch==0||ch==1){
152     ch=8;
153   }
154   else {
155     ch--;
156   }
157   mux.setChannel(ch);
158     Serial << SR_DFLT_START_CHAR << RSP_SUCCESS;
159   Serial << SR_DFLT_SEP_CHAR << ch;
160   Serial << SR_DFLT_STOP_CHAR << endl;
161 }
162 void SerialHandler::_clcChannel()
163 {
164       mux.clcChannel();
165     Serial << SR_DFLT_START_CHAR << RSP_SUCCESS << SR_DFLT_STOP_CHAR << endl;
166 }
167
168 void SerialHandler::_getTemp()
169 {
170   float temp=term.Measure();
171   Serial << SR_DFLT_START_CHAR << RSP_SUCCESS;
172   Serial << SR_DFLT_SEP_CHAR << temp;
173   Serial << SR_DFLT_STOP_CHAR << endl;
174
175 }
```

APÉNDICE J
Comandos LabVIEW

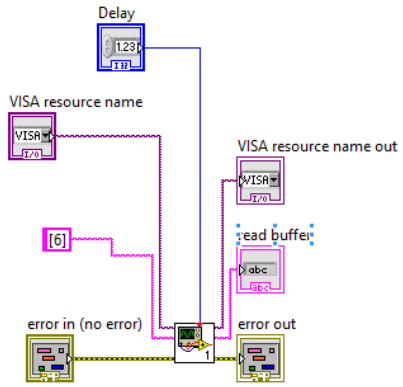


Figura J.1: Comando clcCH

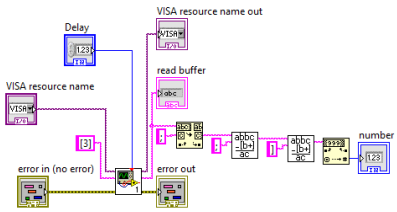


Figura J.2: Comando getCH

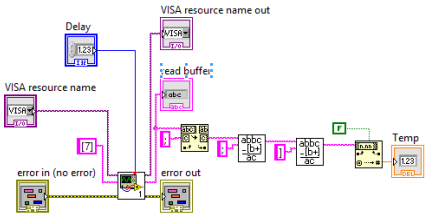


Figura J.3: Comando getTEMP

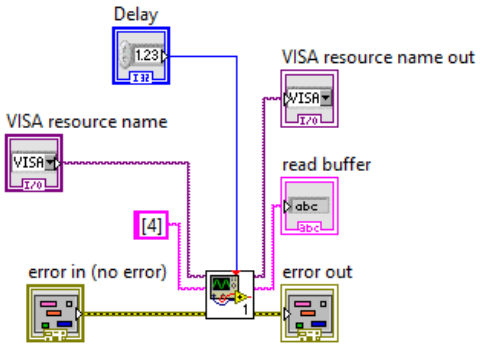


Figura J.4: Comando nextCH

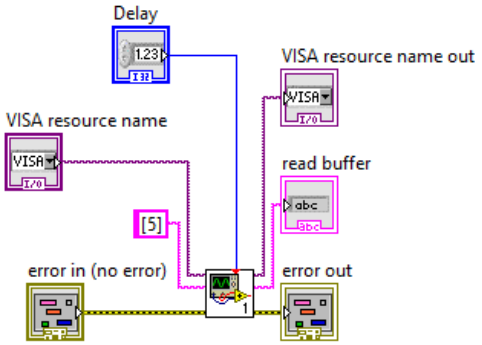


Figura J.5: Comando prevCH

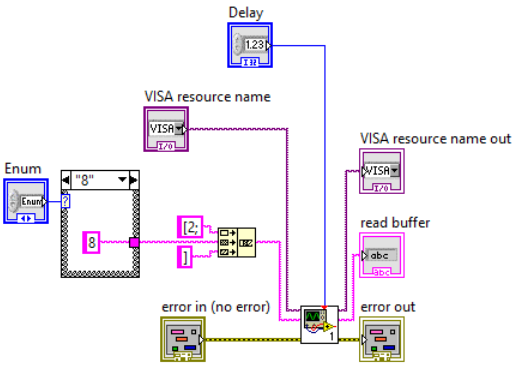


Figura J.6: Comando setCH

APÉNDICE K
Factura PCB

Empresa :
Safe-PCB Spain
Enric Granados 18
08403 - GRANOLLERS (Barcelona)
España
Tel : +34 934 619 988
Fax : +34 934 619 988
Contacto : Mr Daniel Ruiz
Mail : daniel@safe-pcb.com
N°IVA : B-66329889

Banco :
BBVA
c/ doctor Ferran, 5
08440 - CARDEDEU

Cuenta :
IBAN : ES66 0182 9796 3102 0152 9770
Swift : BBVAESMM

CENTRO DE TECNOLOGÍA NANOFOTÓN
Dr. Salvador Ponce Alcántara
Camino de Vera s/n, Edificio 8
46022 - Valencia
ESPAÑA
Salvador Ponce Alcántara - Tel : +34 963877000

N Pedido CDP39442

Acuse de recepción de pedido: circuito validado

Código de cliente: **C01937**

Tiempo de producción: **8 dias**

Dirección de entrega:
CEN DE TEC. NANOFOTÓNICA UPV
Dr. Salvador Ponce Alcántara
Camino de Vera s/n, Edificio 8
46022 - Valencia
Espana

Referencia pedido cliente: **Circuito CEN TEC NAN**

Fecha de envío: **03/08/2016**

Fecha de recepc.: **05/08/2016**

Fecha pedido: **22/07/2016**

Nº pedido SAFE: **CDP39442**

IVA: **Q4618002B**

Forma de pago: **Fact. Proforma -**

Tamaño del circuito

PCB single	☑	☐	Panel
Misma PCB	☐	☐	Diferente PCB
Diseño SAFE	☐	☐	Diseño Cust

Tamaño PCB: **X = 281.00 mm** **Y = 149.00 mm**

PCB quantity

PCB cantidad : **1**

Especific. circuito

Material	**FR4 - 1.6mm**
Capas	**2**
Acab Cu ext	**35 µm**
Acab Cu int	**sin**
Acabado	**HAL - Standard**

Gerber file

Archivo gerber
ger.zip

Pedido especial

Máscaras

Máscaras	**2 caras**
Color máscaras	**verde**
Máscara pelable	**sin**
Serigrafía	**2 caras**
Color	**blanco**
Marc. Rohs	**sin**
Marc. UL	**sin**
Marc. fecha	**sin**

Otras opciones

Prueba eléctrica	**si**	El espacio entre la vía	**> 0.20mm**
Control Impedancia	**no**	Agujeros tamaño mini	**> 0.20mm**
Ranura V-cut en Marg. Tec.	**si**	Carbono	**sin**
Agujero ciego	**No**	Via tapada	**sin**
Corte metalográfico	**no**	Suplemento	**sin**
Borde metalizado	**no**	Taladr. avellanados	**sin**
Press-fit	**no**	Taldr. avell. (qty/pcb)	**sin**
		Espesor Gold finger	**sin**
		Cant Gold finger	**sin**
		Conector Biselado	**sin**
		Adhesivo	**sin**

Stencil

Stencil	**no**
Tipo	**---**
Espesor	**---**
Reducción de PAD	**---**

Material	Marca	Referencia	características
Laminado	**VENTEC**	**VT481**	**FR4 Lamniate - TG150**
preimpregnado	**VENTEC**	**VT481**	**Prepreg - TG150**
Máscara para soldar	**KSM**	**KSM-S6188-G3**	**Green soldermask**
máscara pelable	**-**	**-**	**-**
Tinta	**KSM**	**KSM-388**	**White ink**

Nombre de PCB	*Precio unitario*	*Cantidad*		*El precio total (sin IVA)*
Circuito CEN TEC NAN	**78.667 €**	**1**		**78.67 €**

		IVA %	*IVA*			*Total a pagar*
		21 %	**16.52 €**			**95.19 €**

APÉNDICE L
BOM

Comment	Description	Designator	Footprint	LibRef	Quantity	Value
Cap	Capacitor	C1, C2, C3, C4, C5	12 06 - C	Cap	5	100n
Cap	Capacitor	C6, C7	12 06 - C	Cap	2	1u
Cap	Capacitor	C8, C9	08 05 - C	Cap	2	18p
LED	Typical BLUE SiC LED, Typical BLUE SiC LED, Typical BLUE SiC LED, Typical BLUE SiC LED, Typical BLUE SiC LED, Typical BLUE SiC LED, Led, Led, Led, Led, Led, Led, Typical BLUE SiC LED, Typical BLUE SiC LED, Typical BLUE SiC LED, Typical BLUE SiC LED, Typical BLUE SiC LED, Typical BLUE SiC LED, Typical BLUE SiC LED, Led, Led, Led, Led, Led, Typical BLUE SiC LED, Typical BLUE SiC LED, Typical BLUE SiC LED, Typical BLUE SiC LED, Typical BLUE SiC LED, Led, Typical BLUE SiC LED, Led	D1_U_opto1, D1_U_opto2, D1_U_opto3, D1_U_opto4, D2_U_opto1, D2_U_opto2, D2_U_opto3, D2_U_opto4, D3_U_opto1, D3_U_opto2, D3_U_opto3, D3_U_opto4, D4_U_opto1, D4_U_opto2, D4_U_opto3, D4_U_opto4, D5_U_opto1, D5_U_opto2, D5_U_opto3, D5_U_opto4, D6_U_opto1, D6_U_opto2, D6_U_opto3, D6_U_opto4, D7_U_opto1, D7_U_opto2, D7_U_opto3, D7_U_opto4, D8_U_opto1, D8_U_opto2, D8_U_opto3, D8_U_opto4, D9, D10, D11	12 06 - LED	LED3, LED3, LED3, LED3, LED3, LED3, LED3, LED3, LED, LED, LED, LED, LED, LED, LED3, LED3, LED3, LED3, LED3, LED3, LED3, LED, LED, LED, LED3, LED3, LED3, LED3, LED, LED3, LED3	35	
ULN2803		DAR1_U_opto1, DAR1_U_opto2, DAR1_U_opto3, DAR1_U_opto4	SOP18	ULN2803	4	
TB		J1, J2, J4, J5, J7, J8, J9, J10, J11, J12, J13, J14, J15, J16, J17, J18, J19, J20, J22, J23	MKDSN 2,5/ 2-5,08	Terminal Block	20	
PWR2.5	Low Voltage Power Supply Connector	J3	KLD-0202	PWR2.5	1	
Phonejack	Jack Socket, 1/4" (6.5mm), Thru-Hole, Vertical, 3-Conductor Open Circuit (Non-Normalling)	J6	ACJS-NH35	Phonejack	1	
USB	USB On-The-Go (OTG) Mini-B Receptacle, Right Angle, SMT, 0.80mm (.031") Pitch, Solder Tabs with Back Cover, Recessed Type	J21	USB	USB	1	
VOD207T		OPT1_U_opto1, OPT1_U_opto2, OPT1_U_opto3, OPT1_U_opto4, OPT2_U_opto1, OPT2_U_opto2, OPT2_U_opto3, OPT2_U_opto4	SOIC8	VOD207T	8	
Header 4X2	Header, 4-Pin, Dual row	Pin1, Pout1	HDR2X4	Header 4X2	2	
Header 2X2	Header, 2-Pin, Dual row	Pjump1	HDR2X2	Header 2X2	1	
Res	Resistor	R1_U_opto1, R1_U_opto2, R1_U_opto3, R1_U_opto4, R4_U_opto1, R4_U_opto2, R4_U_opto3, R4_U_opto4, R5_U_opto1, R5_U_opto2, R5_U_opto3, R5_U_opto4, R8_U_opto1, R8_U_opto2, R8_U_opto3, R8_U_opto4, R9_U_opto1, R9_U_opto2, R9_U_opto3, R9_U_opto4, R12_U_opto1, R12_U_opto2, R12_U_opto3, R12_U_opto4, R13_U_opto1, R13_U_opto2, R13_U_opto3, R13_U_opto4, R16_U_opto1, R16_U_opto2, R16_U_opto3, R16_U_opto4, R19, R20, R21	12 06 - R	Res	35	150
Res	Resistor	R2_U_opto1, R2_U_opto2, R2_U_opto3, R2_U_opto4, R6_U_opto1, R6_U_opto2, R6_U_opto3, R6_U_opto4, R10_U_opto1, R10_U_opto2, R10_U_opto3, R10_U_opto4, R14_U_opto1, R14_U_opto2, R14_U_opto3, R14_U_opto4	12 06 - R	Res	16	330
Res	Resistor	R3_U_opto1, R3_U_opto2, R3_U_opto3, R3_U_opto4, R7_U_opto1, R7_U_opto2, R7_U_opto3, R7_U_opto4, R11_U_opto1, R11_U_opto2, R11_U_opto3, R11_U_opto4, R15_U_opto1, R15_U_opto2, R15_U_opto3, R15_U_opto4, R18	12 06 - R	Res	17	10K
Res	Resistor	R17, R24	12 06 - R	Res	2	3.9k
Res	Resistor	R22, R23	12 06 - R	Res	2	22
Res	Resistor	R25, R26	12 06 - R	Res	2	47k
RZ03-1A4		REL1_U_opto1, REL1_U_opto2, REL1_U_opto3, REL1_U_opto4, REL2_U_opto1, REL2_U_opto2, REL2_U_opto3, REL2_U_opto4, REL3_U_opto1, REL3_U_opto2, REL3_U_opto3, REL3_U_opto4, REL4_U_opto1, REL4_U_opto2, REL4_U_opto3, REL4_U_opto4, REL5_U_opto1, REL5_U_opto2, REL5_U_opto3, REL5_U_opto4, REL6_U_opto1, REL6_U_opto2, REL6_U_opto3, REL6_U_opto4, REL7_U_opto1, REL7_U_opto2, REL7_U_opto3, REL7_U_opto4, REL8_U_opto1, REL8_U_opto2, REL8_U_opto3, REL8_U_opto4	1 form A (NO)	RZ03-1A4	32	
SW-PB	Switch	S1	SPST-2	SW-PB	1	
8-bit Microcontroller with ISP Flash and USB Controller		U1	QFP80P1200X1200X120-44N	ATMEGA32U4-AU	1	
DS1820	1-Wire Digital Thermometer	U2	SOIC8	DS1820	1	
XTAL	Crystal Oscillator	Y1	XTAL	XTAL	1	