



SOC_1202_T10_120221

Rediseño de hardware SOC

Versión: 20121010

Autor: Rafel Mormeneo Melich

Inicio	10/07/2012
Final	10/10/2012
Personas	Rafel

VARIACIONES RESPECTO LA VERSIÓN ANTERIOR:

Versión anterior	Ninguna
Variación	1- Ninguna

1- Autor: Vacio. Descripción: Vacio.



1 Rediseño de hardware SOC

2	Objetivos	4
3	Plan de desarrollo de la tarea.....	4
4	Desarrollo de la tarea	4
4.1	Subtareas	4
4.2	Validación del hardware desarrollado en la tarea 4	4
4.3	Rediseño del hardware.....	4
4.3.1	Esquemáticos del diseño hardware	6
4.4	Validación del hardware rediseñado.....	10
5	Conclusiones.....	10



2 Objetivos

El objetivo de esta tarea consiste en validar el hardware diseñado en la tarea 4 (SOC_1202_T4_120221). En caso que sea necesaria se deberá rediseñar el hardware para que se adapte a las necesidades de la aplicación.

3 Plan de desarrollo de la tarea

ID	Tarea	Descripción tarea	Responsable tarea	Inicio	Final	Duración	feb 2012	mar 2012					abr 2012				may 2012				
							19/2	26/2	4/3	11/3	18/3	25/3	1/4	8/4	15/4	22/4	29/4	6/5	13/5	20/5	27/5
1	SOC_1202_T1_120221	Definir hardware cámara	Rafel	21/02/2012	02/03/2012	9d	<div></div>														
2	SOC_1202_T2_120221	Definir el microcontrolador, familiarizarse con el entorno de desarrollo, can, etc.	Fran	21/02/2012	02/03/2012	9d	<div></div>														
3	SOC_1202_T3_120221	Diseñar prototipo	Rafel	05/03/2012	30/03/2012	20d		<div></div>													
4	SOC_1202_T4_120221	Diseñar software prototipo	Fran/Ezio	05/03/2012	13/04/2012	30d		<div></div>													
5	SOC_1202_T5_120221	Pruebas de algoritmos básicos	Victor	21/02/2012	27/04/2012	49d	<div></div>														
6	SOC_1202_T6_120221	Definición protocolo de comunicación Servidor/Cliente/SOC	Ezio/Victor/Fran	16/04/2012	27/04/2012	10d								<div></div>							
7	SOC_1202_T7_120221	Implementación del protocolo de comunicación Servidor/Cliente/SOC	Ezio/Fran	30/04/2012	04/05/2012	5d											<div></div>				
8	SOC_1202_T8_120221	Cambios en TFCient derivados de la aplicación de SOC	Victor/Fran	30/04/2012	18/05/2012	15d											<div></div>				
9	SOC_1202_T9_120221	Diseño de algoritmos alternativos mejorados de visión.	Todos	30/04/2012	01/06/2012	25d											<div></div>				
10	SOC_1202_T10_120221	Rediseño de hardware SOC	Rafel/Ezio	02/04/2012	04/05/2012	25d							<div></div>								
11	SOC_1202_T11_120221	Diseño industrial SOC	Marc/Rafel/Oriol	02/04/2012	04/05/2012	25d							<div></div>								
12	SOC_1202_T12_120221	Diseño del software de visión final	Fran/Ezio/Victor	16/04/2012	18/05/2012	25d								<div></div>							
13																					

Dentro del proyecto SOC, esta tarea (Marcada en verde en el diagrama) ocupa un total de 25 días que van desde 02/04/2012 al 04/05/2012.

Debido a contratiempos en las tareas previas esta tarea se empieza el día 10/07/2012.

4 Desarrollo de la tarea

4.1 Subtareas

Para el desarrollo de esta tarea se han identificado las siguientes subtareas:

- Validación del hardware desarrollado en la tarea 4.
- Rediseño del hardware.
- Validación del hardware rediseñado.

El detalle del desarrollo de cada una de las subtareas se lleva a cabo en los siguientes apartados.

4.2 Validación del hardware desarrollado en la tarea 4

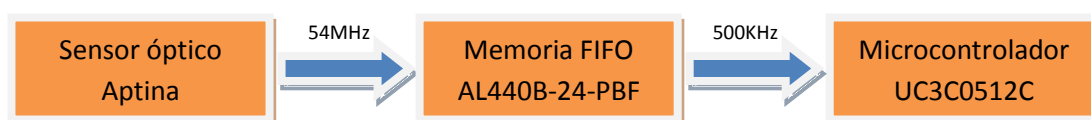
En la tarea 4 habíamos decidido que las imágenes se tomarían mediante un sensor CMOS Aptina. Después de realizar varias pruebas se observa que el diseño inicial no es robusto. Para poder sincronizar el sensor con el microcontrolador se realizó un circuito divisor del clock. Esto es debido a que el sensor como mínimo puede trabajar a 2Mhz. El microcontrolador opera a 60Mhz pero hemos observado que la máxima velocidad para capturar señales es de 700KHz.

4.3 Rediseño del hardware



Aunque el sistema diseñado funciona, la robustez del mismo no es adecuada para un prototipo y menos aún para un producto final. Después de investigar sobre el tema hemos descubierto que una posible solución consiste en introducir una memoria tipo FIFO entre el sensor óptico y el microcontrolador de forma que el sensor envía la imagen que se graba directamente a la memoria y a la velocidad requerida por el sensor y en segundo lugar el microcontrolador puede leer la memoria con una relajación de los requerimientos temporales.

En el siguiente diagrama se muestra la solución planteada al sistema:



Esta solución sería la óptima en cuanto a velocidad y a robustez del sistema. El sensor óptico y el microcontrolador dan las señales necesarias para que la memoria FIFO empiece a grabar los datos de la imagen a la velocidad del sensor (54MHz). El sensor nos da la señal de final de imagen y en ese momento el microcontrolador debe empezar a leer la memoria externa para cargar la imagen en su memoria RAM interna. La ventaja de este sistema es que la memoria separa el bus de datos en dos velocidades, sensor y microcontrolador funcionan a distintas velocidades.

Esta solución requiere más investigación para realizar la conexión entre la memoria externa y el microcontrolador. Para el primer prototipo se optará por otra solución mediante la utilización de un módulo de cámara que se comercializa a través de 123electronics. Este módulo es el C429 y se trata de un sensor óptico Aptina con un procesador de imagen que comprime las imágenes con el códec JPEG. Las imágenes se pueden leer a través de un puerto serie RS-232. El siguiente diagrama muestra la arquitectura de dicha solución:



Además, en este prototipo se ha optado por no incluir un display como se había mencionado en los puntos anteriores. Se ha valorado la integración de distintos displays como: NHD-2.4-240320SF-CTXI#-1 de New Heaven Display, o T-55149GD030J-MLW-AJN de Optrex. Para este segundo nos visitó un comercial de Venco electrónica. Se ha decidido no añadir el display al prototipo ya que el driver de estos displays no es el mismo que el del EDT024006 del que disponemos la librería de código. El display no es un elemento imprescindible en el sistema.

El display lo queríamos incluir para poder instalar correctamente el SOC en los motores. En su lugar se ha optado por otra solución consistente en enviar las imágenes por puerto serie a un PC. De esta forma, las tareas de instalación se realizarán con la ayuda de un portátil. Para este fin se ha abierto una nueva tarea, la SOC_1202_T10.1_120925. En esta nueva tarea se diseñará una aplicación de Windows Forms para la ayuda a la instalación del sensor.

Para poder tratar las imágenes deben, en primer lugar, ser descomprimidas. Para este propósito, Atmel cuenta con una librería JPEG. De todas formas, para descomprimir las imágenes necesitaremos una memoria SDRAM externa. Aprovecharemos el diseño de la placa de desarrollo EK-UC3C para este propósito.



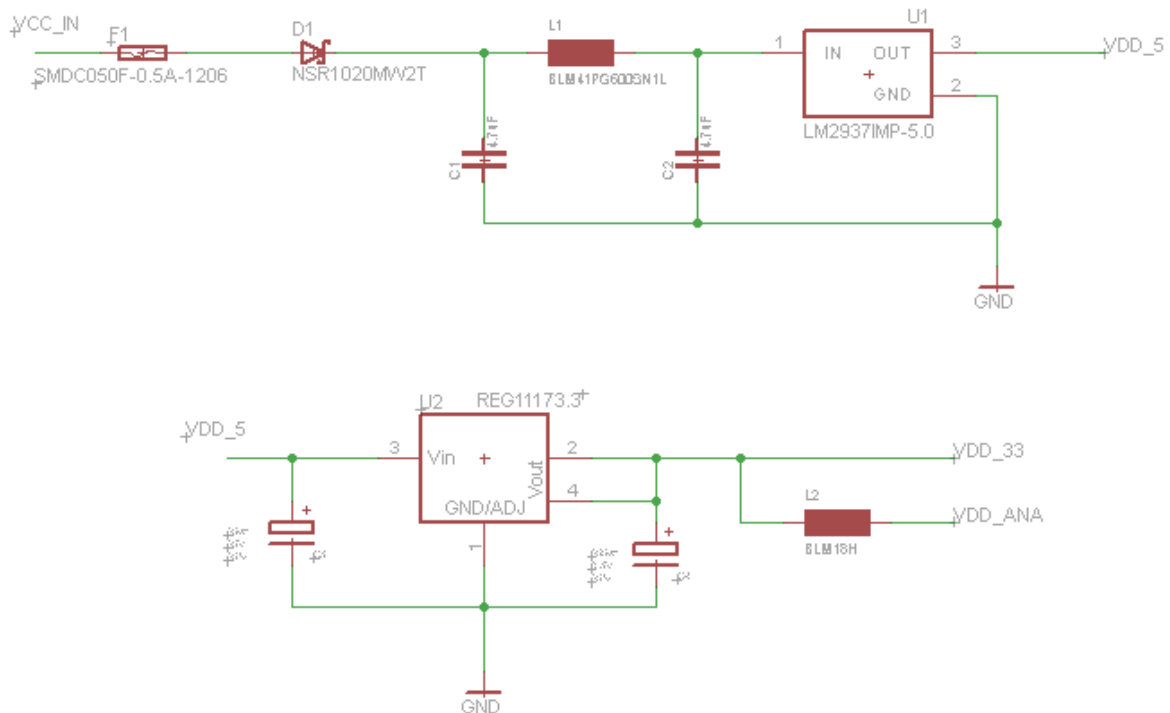
4.3.1 Esquemáticos del diseño hardware

A continuación se detallan los esquemáticos del diseño electrónico con los módulos antes descritos. En el directorio de Diseño se encuentran los archivos Eagle correspondientes a dicho diseño.

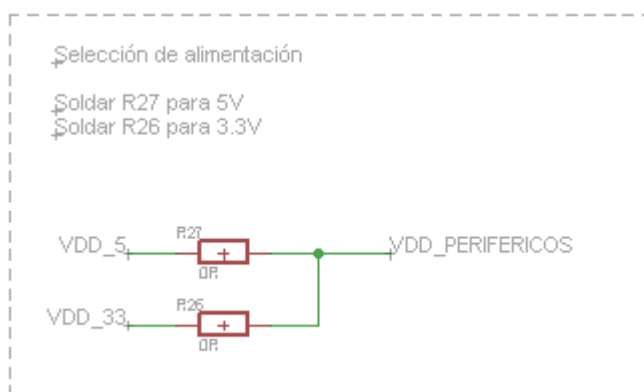
El diseño electrónico contiene 4 páginas. La alimentación, el microprocesador, los periféricos y los conectores.

Alimentación

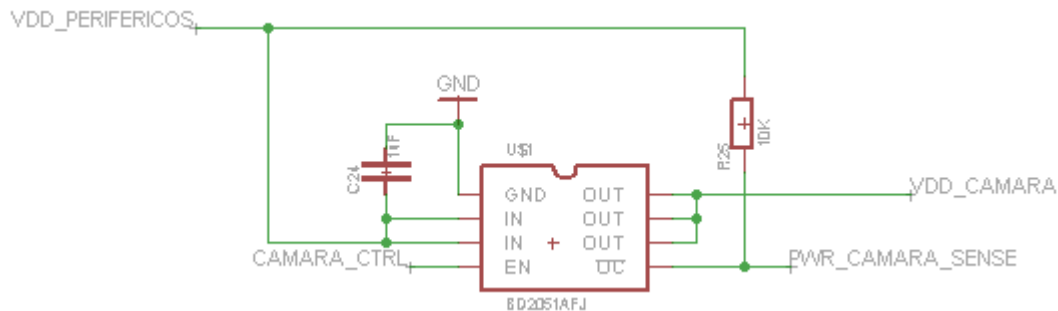
Debido a que ya hemos realizado pruebas con el módulo C429 alimentado a 5V, en este prototipo tendremos dos alimentaciones distintas. Los 5V para alimentar la cámara, el CAN y el RS232. Los 3.3V alimentarán una memoria SDRAM y el microcontrolador.



Para poder realizar pruebas se realiza un circuito para poder seleccionar la alimentación de los periféricos (CAN, RS232 y cámara) entre +5V y +3.3V. **Hay que tener en cuenta que el C429, el CAN y el RS232 son distintos para +5V o para +3.3V.** Véase el apartado de periféricos.

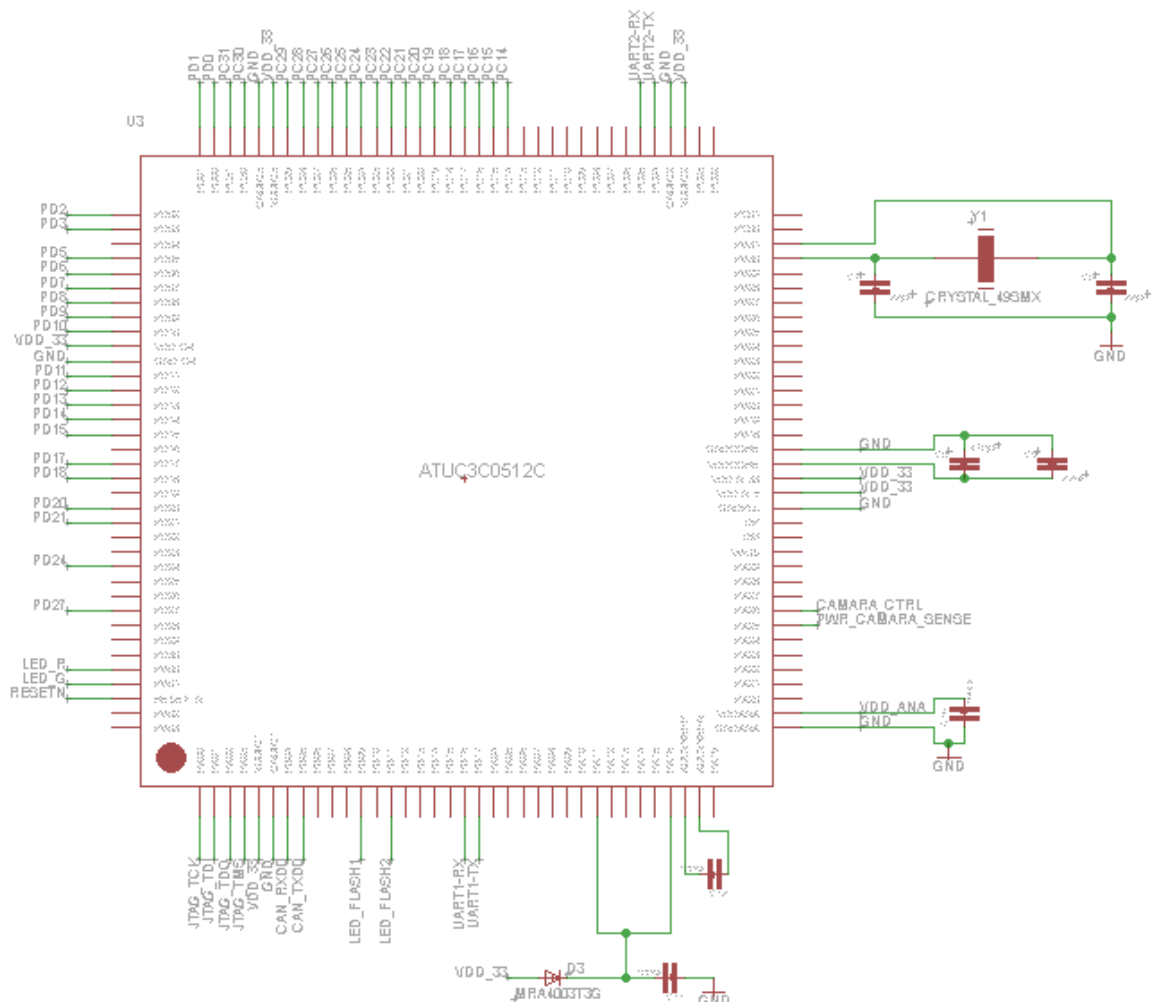


Además, se incluye el IC BD2051. Se trata de un driver para poder apagar la alimentación de la cámara electrónicamente mediante el microcontrolador.



Microprocesador

Mantenemos el AT32UC3C0512C en el core del dispositivo.



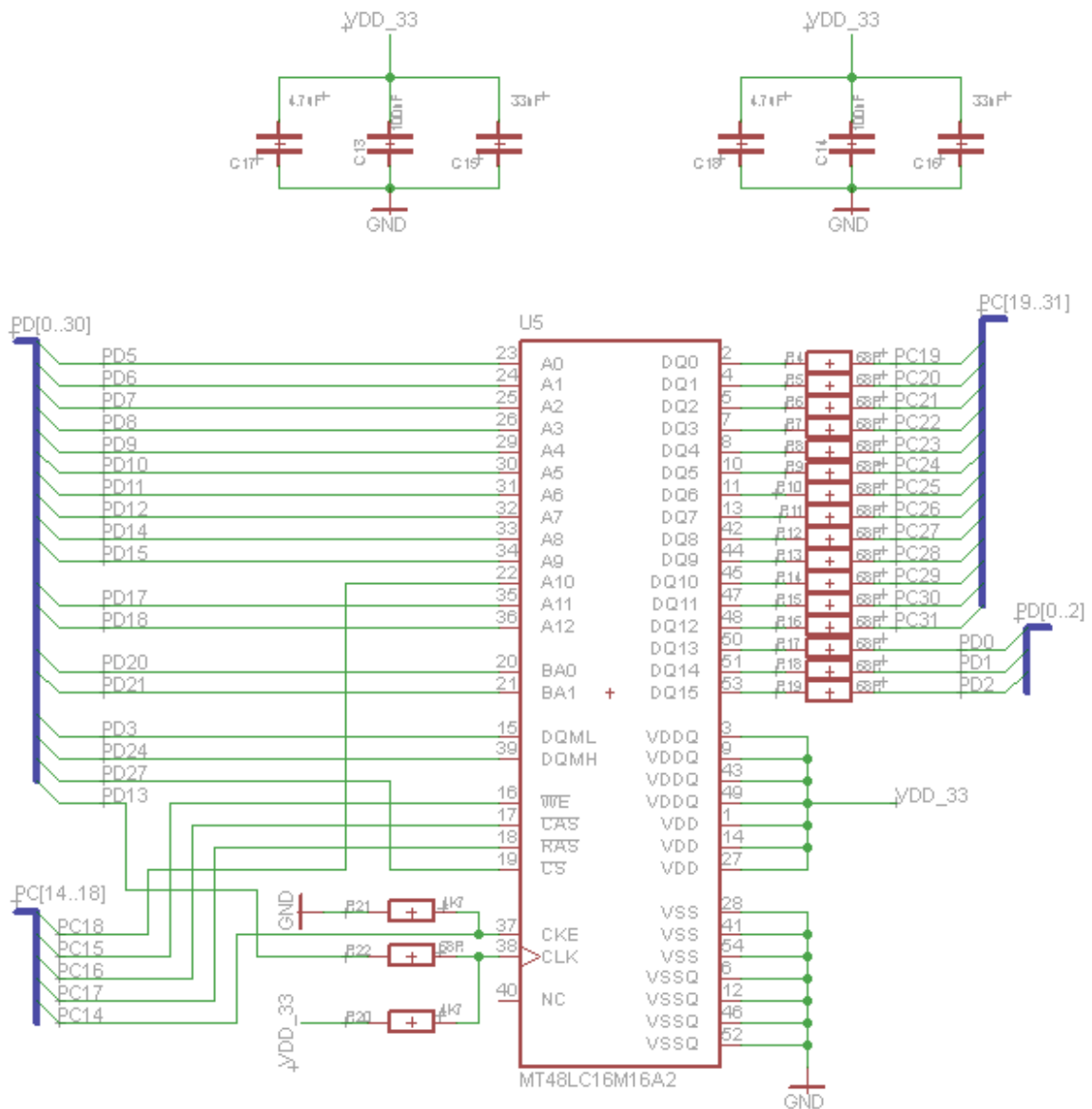
Periféricos

Los periféricos son:

- Una memoria SDRAM para la descompresión de imágenes.

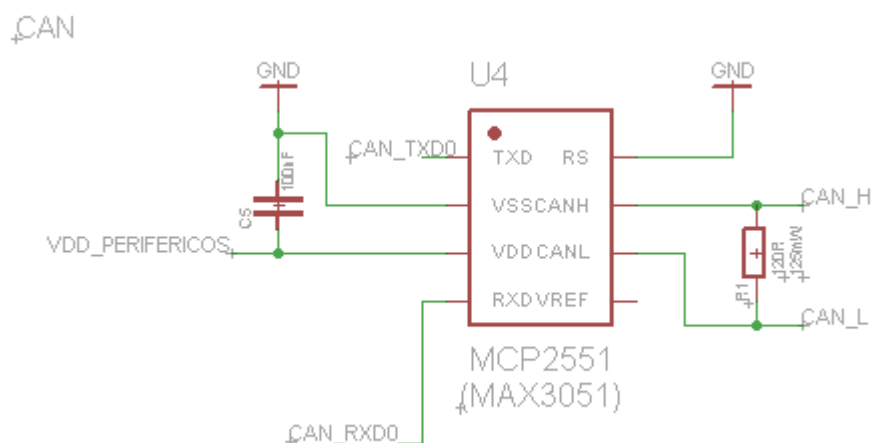


SDRAM

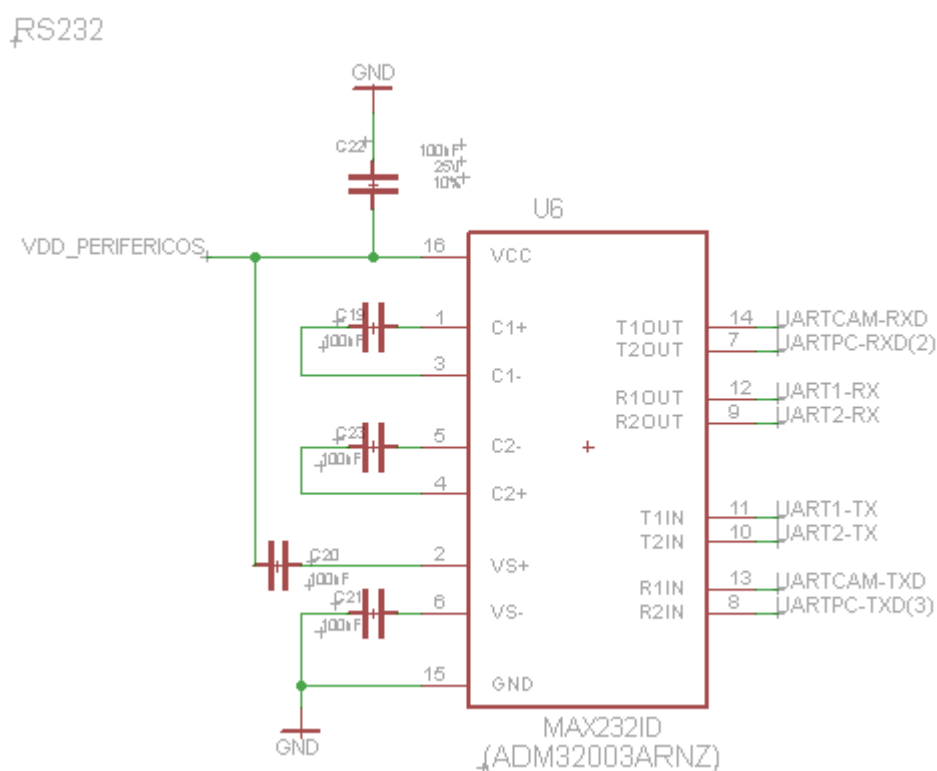


Un transceiver CAN para la comunicación con el DAEN asociado o con el ECON. En el caso que la alimentación sea +5V se instalará el MCP2551. *En el caso de la alimentación a +3.3V optaremos por el MAX3051.* Los packages y footprints de los dos componentes son idénticos.





Un driver RS-232 doble, uno para la comunicación con el módulo de la cámara C429 y otro para la comunicación con el PC. *En el caso que la alimentación sea +5V se instalará el MAX232ID. En el caso de la alimentación a +3.3V optaremos por el ADM32003ARNZ.* Los packages y footprints de los dos componentes son idénticos.



Conectores

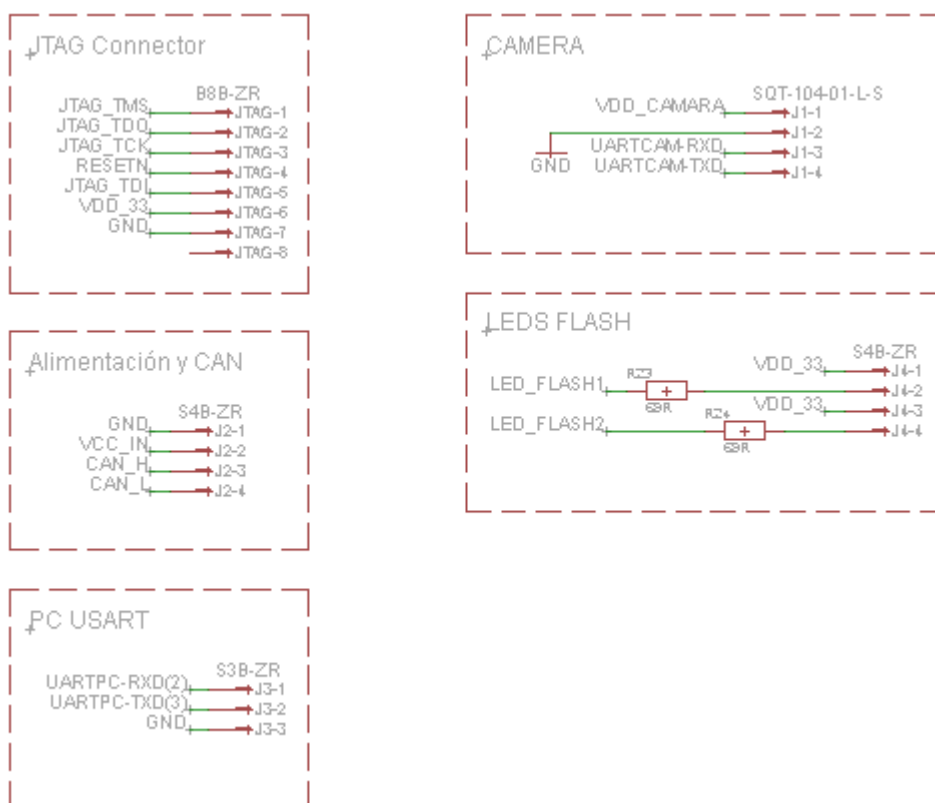
Se instalarán 5 conectores en total. 4 JST de la serie ZR y uno para la interconexión de la placa con el módulo C429. Todos los conectores son THD para mejorar la robustez del anclaje.

El conector de programación a través de JTAG requiere 7 señales. Hemos optado por el conector B8B-ZR de 8 pines ya que para el de siete el socket no estaba disponible en la web de los distribuidores.

Para la alimentación y CAN y para los LEDS se utiliza un S4B-ZR en ángulo recto.



Para la conexión serie al PC se opta por el S3B-ZR también en ángulo recto.



4.4 Validación del hardware rediseñado

Se ha comprobado con la placa de desarrollo EK-UC3C que el diseño realizado funciona correctamente. Para continuar con la validación será necesario la fabricación y montaje de las PCB.

5 Conclusiones

Se ha realizado el rediseño del hardware teniendo en cuenta que queremos realizar un prototipo funcional lo antes posible. Para ello se ha elegido un módulo con cámara y procesador que comprime las imágenes con el códec JPEG. Para tratar dichas imágenes se ha tenido que dotar al microcontrolador de una memoria SDRAM externa.

Se ha decidido no incluir el display en el dispositivo de soporte a la instalación. En lugar de ello se ha sustituido este dispositivo por un PC. Para ello se ha abierto una nueva tarea para realizar una aplicación con Windows Forms para poder instalar correctamente el SOC. La nueva tarea es la [SOC_1202_T10.1_120925](#).

En el futuro se estudiará la posibilidad de alimentar todos los componentes del dispositivo a +3.3V. Se ha hecho el diseño del hardware de forma que sea fácil realizar este cambio.

Tarea completada: 306 horas o 38 días.

