

UNIVERSIDAD AUTONOMA DE MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

MANUAL DE LA TARJETA JPS-XPC84



Versión 1.0
31/Enero/2003

Pablo A. Haya Coll (Pablo.Haya@ii.uam.es)
Juan González Gómez (juan@iearobotics.com)

Descripción del documento

Este manual se rige por la licencia *GNU GFDL 1.1 (GNU FREE DOCUMENTATION LICENSE)*, que otorga permisos para utilizar, copiar, modificar y/o redistribuir este documento. La versión más actual, junto con los ficheros fuentes, se puede bajar de:

<http://www.iearobotics.com/personal/juan/doctorado/jps-xpc84/jps-xpc84.html>

Para más información sobre los autores de este documento, visitar:

- **Pablo A. Haya:**

<http://www.ii.uam.es/~phaya/>

- **Juan González:**

<http://www.iearobotics.com/personal/juan/>

Copyright

Copyright (©) 2002. Pablo A. Haya, Juan González Gómez.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover texts and with no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Historia

He aquí los cambios más importantes sufridos por el documento. La versión indica cambios de contenido, mientras que la generación alude al grado de terminación del documento. Para saber más sobre las terminaciones, visita la ubicación en la red del documento.

ver. 1.0

31 de Ene de 2003

- Apéndices –**JGG**
- Apartados 11 y 12 (Lo que queda por hacer y FAQs) –**JGG**
- Apartado “Fabricación” –**JGG**
- Apartado “Trabajando con la JPS” –**JGG**
- Apartado de Montaje –**JGG**
- Añadidas nuevas figuras– **JGG**
- Añadido esquemático en EPS –**JGG**
- Apartado sobre la placa JPS– **JGG**

ver. 0.9.1

7 de Dic de 2002

- Documento pasado a Lyx– **JGG**
- Añadidas nuevas figuras– **JGG**
- Agradecimientos al proyecto alqua (<http://alqua.com>) de donde se ha obtenido la plantilla de este apartado, así como la versión 1.1 de la licencia GFDL en lyx.

ver. 0.9

28 de Dic de 2001

- Primera versión del documento. Realizada en Latex– **PHC**

Índice general

1. Sobre la placa JPS-XPC84	11
1.1. Autores	11
1.2. Licencia	11
1.3. Herramientas de desarrollo	11
1.4. Aplicaciones	12
1.5. Sobre este manual	12
2. Descripción	13
2.1. FPGAs Compatibles	13
2.2. FPGAs probadas	14
2.3. Características de la JPS-XPC84	14
2.4. Modos de funcionamiento	16
2.4.1. Modo entrenadora	16
2.4.2. Modo autónomo	16
3. Diagrama de bloques	19
4. Aspecto físico	21
5. Modos de programación	27
5.1. Introducción	27
5.2. Modo de configuración maestro	27
5.3. Modo de configuración esclavo	28
5.4. Puerto de control	29
5.5. Programación de la FPGA desde el PC	31
5.6. Programación de la EEPROM desde el puerto de control	32
6. Puertos	35
6.1. Puertos de expansión	35
6.2. Temporización de los diseños	36
6.3. Switches de configuración genéricos	38
6.4. Pulsador genérico	38
6.5. Led genérico	38
6.6. Distribución global	38
7. Alimentación	41

8. Montaje	43
8.1. PCB	43
8.2. Componentes	43
8.3. Disposición de los componentes	43
8.4. Pruebas	46
9. Trabajando con la JPS	47
9.1. Esquema de trabajo	47
9.2. Un ejemplo muy sencillo: un inversor	47
9.2.1. El esquema	47
9.2.2. El diseño en VHDL	49
9.2.3. El fichero restricciones: inversor.ucf	50
9.2.4. Otros ficheros: inversor.hex, inversor.mcs	50
9.3. Configuración de jumpers y switches	51
10. Fabricación	53
10.1. Introducción	53
10.2. Ficheros de fabricación	53
10.3. Fabricando la placa	54
11. Lo que queda por hacer	55
12. FAQs, preguntas más frecuentes	57
A. Numeración del Zócalo PLCC84	59
B. GNU Free Documentation License	61
B.1. Applicability and Definitions	61
B.2. Verbatim Copying	62
B.3. Copying in Quantity	62
B.4. Modifications	63
B.5. Combining Documents	64
B.6. Collections of Documents	64
B.7. Aggregation With Independent Works	64
B.8. Translation	65
B.9. Termination	65
B.10. Future Revisions of This License	65

Índice de figuras

2.1. Esquemático de la tarjeta JPS-XPC84, versión PRO.	15
2.2. Modo entrenador, usando un sistema externo distinto de un PC para cargar la FPGA	16
2.3. Modo entrenador, usando un PC	17
3.1. Diagrama de bloques de la tarjeta JPS-XPC84, versión PRO	20
4.1. Aspecto físico de la tarjeta JPS-XPC84, versión PRO.	22
4.2. Situación de los componentes de programación	22
4.3. Situación de los conectores de alimentación	23
4.4. Situación del pulsador y LED de pruebas	23
4.5. Situación del circuito de reloj	24
4.6. Situación de los puertos de conexión	24
4.7. Situación de los conmutadores genéricos	25
5.1. Conexiones entre la FPGA y la memoria serie AT17C128	28
5.2. Cronograma del proceso de lectura de una EEPROM serie AT17C128	28
5.3. Características de la EEPROM serie AT17C128	29
5.4. Señales de configuración de la FPGA en modo esclavo	30
5.5. Configuración de pines del puerto de control	30
5.6. Aspecto del cable de descarga paralelo suministrado por Xilinx	31
5.7. Foto del conector CT8	32
5.8. Componente S3, conmutadores	33
5.9. Conexión para programar la memoria EEPROM desde un sistema externo. Proporcionado por ATMEL.	34
6.1. Configuración de los pines de un puerto de expansión	36
6.2. Disposición de los pines del puerto de expansión en la tarjeta. La figura izquierda muestra el lado de los componentes (cara de arriba) y la derecha el lado de la soldadura (cara de abajo).	36
6.3. Los puertos de expansión	37
7.1. Conector hembra y jack macho para la alimentación	41
7.2. Alimentación de la tarjeta JPS mediante 4 pilas de 1.5 voltios	42
8.1. PCB, cara de arriba	44

Índice de figuras

8.2. PCB, cada de abajo	44
8.3. Disposición de componentes en la placa JPS	45
9.1. Esquema de trabajo con la JPS	48
9.2. El esquema del inversor de pruebas	49
9.3. Diseño integrado en la tarjeta JPS	49
10.1. El PCB de la tarjeta JPS, visto por la cara de arriba y la de abajo	54
A.1. Vista superior del zócalo	60
A.2. Vista inferior del zócalo	60

Índice de cuadros

2.1. Características de las FPGAs compatibles con la tarjeta JPS-XPC84	13
5.1. Modos de configuración de la tarjeta JPS-XPC84	27
5.2. Distribución de pines del puerto de control	30
5.3. Correspondencia de los pines del conector CT8 con el cable de descarga suministrado por Xilinx	31
6.1. Pines de los puertos de extensión que se pueden configurar como entradas de reloj	36
6.2. Correspondencia de los switches de configuración genéricos con pines de la FPGA	38
8.1. Listado de componentes de la placa JPS-XPC84	45

Índice de cuadros

1. Sobre la placa JPS-XPC84

1.1. Autores

Los autores de la primera versión (Feb-2002) de la placa JPS-XPC84 son:

- **Juan González Gómez.** *juan@iearobotics.com* [2]
- **Pablo A. Haya.** *Pablo.Haya@ii.uam.es* [3]
- **Sergio López-Buedo.** *Sergio.Lopez-Buedo@ii.uam.es.*

1.2. Licencia

Esta placa está pensada para la docencia, con el objetivo de que los alumnos y todos aquellos interesados puedan llevar sus diseños a la realidad, de una forma sencilla. Por ello dispone de una licencia libre, **que permite su libre copia, modificación, distribución y/o distribución de la modificaciones siempre que vayan acompañadas de esta nota así como de TODOS los siguientes documentos:**

- **Fichero de esquemáticos**, en algún formato fuente, para que otros lo puedan modificar
- **Fichero con el PCB**, también en algún formato fuente
- **Fichero de Fabricación** (Formato GERBER).

Esto garantiza que la placa siga siendo libre y que todas sus mejoras y modificaciones también lo sean.

Toda la información sobre la placa se puede obtener en:

<http://www.iearobotics.com/personal/juan/doctorado/jps-xpc84/jps-xpc84.html>

1.3. Herramientas de desarrollo

Las herramientas empleadas para su diseño han sido:

- Ordenador Portátil con S.O **Debian/GNU Linux**

1. Sobre la placa JPS-XPC84

- **Programa de diseño electrónico EAGLE**, distribuido por la empresa CADSoft [5]. Este programa es Multiplataforma y se puede descargar gratuitamente de su web. Se concede una licencia para su uso libre, totalmente funcional pero que limita el tamaño de la máxima placa que se puede hacer. Una placa como la JPS-XPC84 se puede diseñar sin problemas.
- Para la **realización de este manual** se ha empleado:
 - **Procesador de texto**: Lyx, 1.1.6.fix4 [6]
 - **Diagramas y figuras**: Xfig 3.2 [7]

1.4. Aplicaciones

La tarjeta puede ser empleada para infinidad de cosas, sin embargo resulta especialmente útil para:

- **Aprendizaje**: Es su misión fundamental. Que los alumnos o gente que está aprendiendo pueda disponer de una plataforma sencilla, libre y fácil de usar. Al ser libre y disponer de todos los esquemas, los estudiantes pueden comprender todos los detalles y construirse sus propias entrenadoras o reproducir esta misma, bien en modo prototipo o bien mandando fabricar el mismo PCB.
- **Conexion con microcontroladores**: Por tratarse de un sistema autónomo es muy útil para desarrollar periféricos para diferentes microcontroladores: 6811, 6808, PIC...
- **Robotica**: Muy útil en la construcción de robots autónomos o periféricos para ellos
- **Investigación**: Posibilidad de implementar pequeñas CPU's en la FPGA, que luego funcionarán de manera autónoma
- **Hardware abierto**: La posibilidad de que esta placa se convierta en un sistema de desarrollo libre empleado por desarrolladores de hardware libre, que diseñen cores o hardware para esta FPGA con una licencia libre. Esto hará, al igual que en el caso del software libre, que todos nos beneficiemos de las aportaciones que otras personas hacen a la comunidad.

1.5. Sobre este manual

Este documento **NO ES UN TUTORIAL SOBRE FPGA's**, ni se muestran ni enseñan las herramientas de desarrollo. Es un manual que describe en detalle la placa JPS-XP84, para que los usuarios la puedan utilizar y/o desarrollar nuevas mejoras o software.

Se supone que el usuario tiene conocimientos sobre FPGA's y que ha simulado algún diseño. Con la placa JPS-XPC84 lo podrá probar y comprobar si funciona correctamente.

No obstante, en el capítulo 9 se da un ejemplo muy sencillo y se explica a modo de referencia rápida cómo hay que configurar la placa JPS-XP84 para probar su funcionamiento.

2. Descripción

La **JPS-XPC84** es una tarjeta para el desarrollo de diseños hardware sobre dispositivos de lógica programable (**FPGA**) de las familias **4000** y **Spartan** suministradas por la empresa **Xilinx**[1]. La tarjeta aporta el **hardware mínimo para programar una FPGA, circuitería de reloj** para poder realizar **diseños síncronos**, y **puertos de expansión** para conectar la FPGA con otros sistemas.

La configuración de la FPGA se puede realizar mediante tres formas:

- **PC:** Permite cargar desde el ordenador los prototipos realizados. Para ello se puede utilizar un cable propietario suministrado por Xilinx[1].
- **Memoria EEPROM:** se pueden grabar los diseños en una EEPROM serie que se ubica en un zócalo de la tarjeta a través de la cual se programa la FPGA. Cada vez que se alimente la placa, la FPGA se cargará automáticamente desde la EEPROM.
- **Subsistema:** Un sistema externo, como puede ser un microcontrolador, puede o bien programar la EEPROM serie, o bien configurar directamente la FPGA.

2.1. FPGAs Compatibles

La tarjeta ha sido diseñada para soportar las siguientes **FPGAs**, siempre y cuando el **encapsulado** sea de tipo **PLCC84**:

- **Familia 4000:** XC4003E, XC4005E, XC4005L, XC4006E, XC4010E, XC4010L
- **Familia Spartan:** XCS05, XCS10

En tabla 2.1 se muestran las características de estas FPGAs

Device	Logic cells	CLB Matrix	Total CLBs	Total FF
XC4003E	238	10x10	100	360
XC4005E	466	14x14	196	616
XC4010E	950	20x20	400	1120
XCS05	238	10x10	100	360
XCS10	466	14x14	196	616

Cuadro 2.1.: Características de las FPGAs compatibles con la tarjeta JPS-XPC84

2. Descripción

2.2. FPGAs probadas

Las FPGAs probadas, que se puede garantizar que funcionan correctamente, son las siguientes:

- 27/12/2001: **XCS10**
- 08/03/2002: **X4010**

Si el lector ha probado cualquiera de las FPGA's descritas en el apartado 2.1, agradeceríamos que se pusiese en contacto con los autores para incluirla en esta lista.

2.3. Características de la JPS-XPC84

En la figura 2.1 se muestra el esquemático de la tarjeta. A lo largo de la documentación se van a realizar sucesivas referencias a los distintos componentes empleando la nomenclatura definida en él.

A continuación se describen las **características** más importantes **de la tarjeta**.

- **FPGA PLCC84**: la tarjeta permite utilizar las FPGAs de la familia Spartan y 4000 definidas en la tabla 2.1 que tenga el **encapsulado de tipo PLCC84**.
- **6 Puertos de expansión**: que dan acceso a las patas genéricas de entrada/salida de la FPGA.
- **Reloj**: circuitería de reloj integrada en la placa para poder realizar **diseños síncronos**.
- **Circuitería de pruebas**: se proporcionan un **led** y un **pulsador** conectados a pines fijos de la FPGA que permiten comprobar el correcto funcionamiento de la tarjeta.
- **Led de programación**: que permite comprobar el estado de la FPGA. Cuando se encuentra encendido indica que la FPGA está desprogramada. Por el contrario, si se encuentra apagado indica que la FPGA se ha programado correctamente.
- **Led de alimentación**: se mantiene encendido siempre que la tarjeta se encuentre alimentada.
- **Switches de configuración genéricos**: tres switches que se encuentran conectados a pines fijos de la FPGA y que se pueden utilizar como entradas de configuración genéricas.
- **Alimentación**: a través de clema y de jack.
- **Modos de configuración maestro/esclavo**, seleccionables desde un jumper. La configuración maestro permite programar la FPGA desde una EEPROM serie de 128 Kbyte. En el caso de esclavo, la carga se realiza mediante un sistema externo a la tarjeta, un PC, un microcontrolador, otra FPGA ...

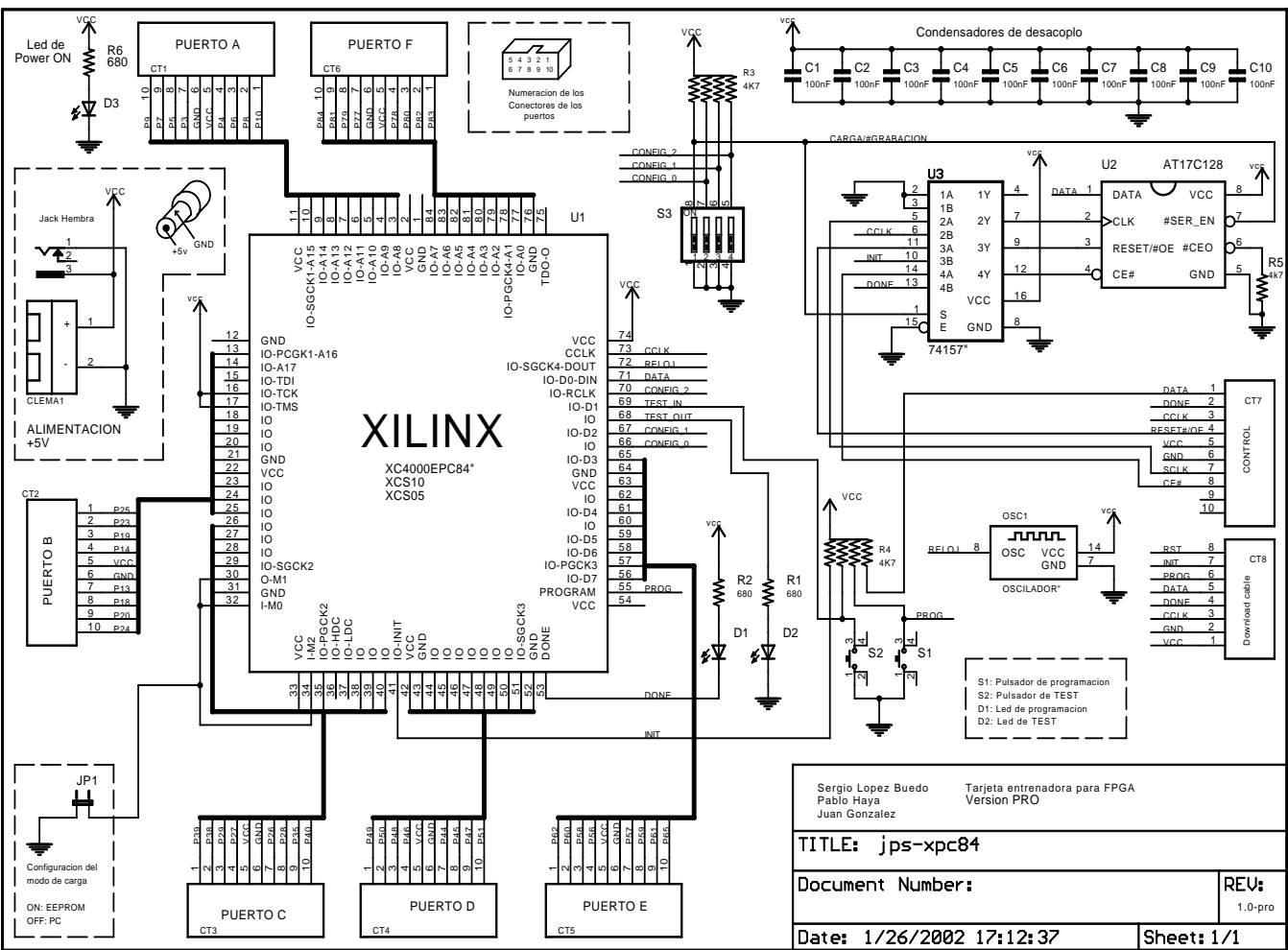


Figura 2.1.: Esquemático de la tarjeta JPS-XPC84, versión PRO.

2. Descripción

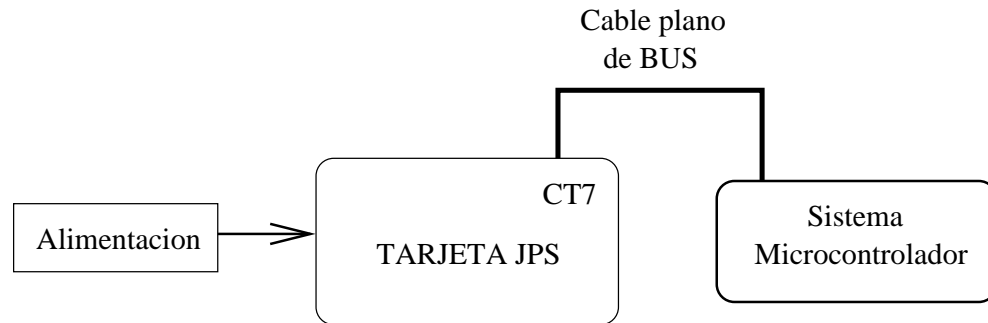


Figura 2.2.: Modo entrenador, usando un sistema externo distinto de un PC para cargar la FPGA

- **Programación *in circuit* de la EEPROM.** Se dispone de un puerto adicional que permite programar la EEPROM serie sin extraerla del zócalo.
- **Reset de la FPGA** mediante un pulsador.

2.4. Modos de funcionamiento

La tarjeta permite **dos modos de funcionamiento** claramente diferenciados que permiten al diseñador mayor flexibilidad.

2.4.1. Modo entrenadora

En este modo **la FPGA se configura utilizando un sistema externo**. Para ello se han habilitado dos conectores: **CT7** (puerto de control, ver la Figura 5.5) y **CT8** (ver la Figura 5.7).

El primer conector se denomina **puerto de control**. Permite conectar mediante un cable plano otro sistema para poder programar la FPGA (ver 5.4 y figura 2.2). Mediante este puerto también se puede programar la **EEPROM serie** (ver 5.6).

Por otro lado el **conector CT8** está pensado para poder emplear un PC (ver 5.6). Para conectar el PC con la tarjeta se puede utilizar un cable suministrado por la empresa Xilinx (Figura 5.6) o uno de fabricación propia. Este último modo permite un ciclo de diseño bastante flexible ya que permite cargar fácilmente los prototipos, de forma que las modificaciones realizadas se pueden probar rápidamente (Figura 2.3).

2.4.2. Modo autónomo

En este modo la tarjeta solamente requiere de una alimentación externa para funcionar ya que la FPGA se configura utilizando una **EEPROM serie**. El diseñador graba el *bitstream* de su aplicación en una EEPROM que se coloca en el zócalo U2 (ver Figura 2.1). La FPGA se configura automáticamente siempre que se conecta la alimentación, o se aprieta el pulsador de programación (S1).

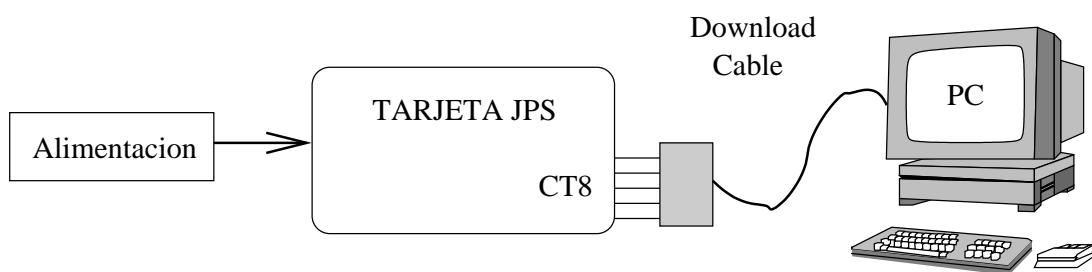


Figura 2.3.: Modo entrenador, usando un PC

2. Descripción

3. Diagrama de bloques

El componente central de la tarjeta JPS-XPC84 es la FPGA utilizada. El resto de los módulos interactúan con la FPGA aportando distintas funcionalidades.

- **Circuito de reloj:** La FPGA se puede configurar tanto con diseños combinacionales como secuenciales. Para estos últimos diseños la placa provee de un **circuito de reloj externo**. Además en cada uno de los puertos de expansión existe un pin que se puede utilizar como entrada de reloj, permitiendo tener más de un circuito de reloj. En el apartado 6.2 se detallan todas las posibilidades de temporización.
- **Circuito de prueba:** El circuito de prueba se compone de un **led** y un **pulsador** conectados cada uno a un pin fijo de la FPGA. Estos dos componentes permiten probar fácil y rápidamente el correcto funcionamiento de la placa. Los componentes del circuito de prueba se describen en el apartado 6.4 y 6.5.
- **Conmutadores genéricos:** El diseñador dispone de tres conmutadores genéricos de forma que sus prototipos pueden tener hasta ocho modos de configuración distintos. Para más información ver el apartado 6.3.
- **Puertos de expansión:** La placa cuenta con seis puertos de expansión genéricos. Cada uno tiene **ocho pines de datos** que se pueden configurar tanto de entrada como de salida. Además se proveen de **dos pines más de alimentación**, de forma que los sistemas externos pueden obtener la alimentación a través del puerto al que esté conectado. Los puertos de expansión se detallan en el apartado 6.1.
- **Alimentación:** Para la alimentación de la tarjeta se disponen de dos tipos de conectores distintos: **jack** y **clema**. Es importante reseñar que NO se pueden utilizar los dos conectores al mismo tiempo. **La alimentación de la placa es de 5 V**. Para más detalles consultar el apartado 7.
- **Circuito de programación interna:** Se utiliza para programar la FPGA desde una **EEPROM serie integrada** en la placa. La FPGA actúa en modo maestro y toma el *bitstream* de la memoria. Este modo de configuración se describe en el apartado 5.2.
- **Circuito de programación externa:** En este caso la FPGA se utiliza en modo esclavo. Un sistema externo, habitualmente un PC, se encarga de suministrar el *bitstream* mediante un conector y un cable propietario. Los detalles de este modo de configuración se describen en el apartado 5.3.

3. Diagrama de bloques

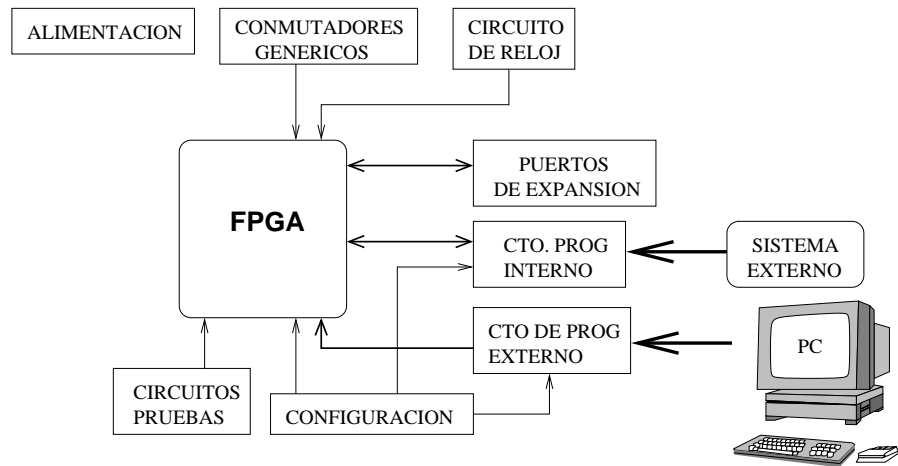


Figura 3.1.: Diagrama de bloques de la tarjeta JPS-XPC84, versión PRO

- **Configuración:** Las posibilidades de configuración de la placa versan sobre las distintas formas que se disponen de programar la FPGA. Mediante un *jumper* se elige que la programación sea desde un sistema externo o mediante los circuitos internos. Mediante un conmutador se permite que la EEPROM serie se pueda programar desde un sistema externo sin necesidad de extraerla del zócalo.

4. Aspecto físico

En esta sección se describen gráficamente la distribución en la placa de los componentes más importantes de la misma. En la figura 4.1 se muestra la serigrafía de placa, en la que se detalla la situación y referencia de todos los componentes.

- Componentes relacionados con la programación de la FPGA (figura 4.2):
 - El **puerto de control (CT7)**
 - **Switch 1**, para seleccionar si la memoria eeprom cargará la FPGA o se programará desde el puerto de control.
 - **Multiplexor** de 2 canales de 2 bits, para conectar la eeprom bien a la FPGA o bien al puerto de control
 - **Diodo D1**, se enciende mientras se está programando y se apaga una vez terminado.
 - **Pulsador S1**, cada vez que se aprieta la FPGA se reinicia y se vuelve a cargar
 - **Jumper JP1**: Especifica el modo de funcionamiento: Si está puesto se carga desde la EEPROM, si está quitado desde el sistema externo (como el PC).
 - **Memoria EEPROM serie.**
- Conectores de alimentación (Figura 4.3)
 - **Jack hembra** para circuito impreso, para la conexión de un conector tipo jack macho de 5mm
 - **Clema doble** para introducir la alimentación a través de dos cables
 - **Led D3**: Se enciende si está alimentado
- Led y pulsador de pruebas (Figura 4.4)
- Reloj (Figura 4.5)
- Puertos de conexión (Figura 4.6)
- Conmutadores genéricos (Figura 4.7)

4. Aspecto físico

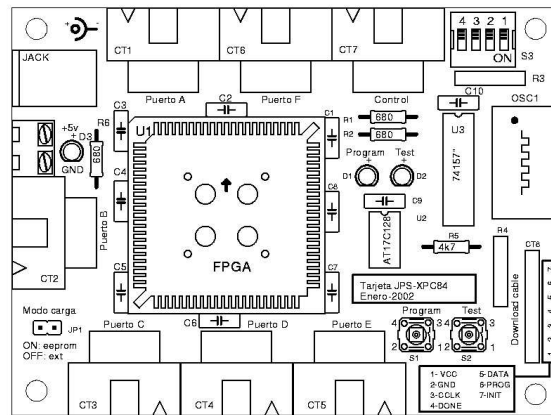


Figura 4.1.: Aspecto físico de la tarjeta JPS-XPC84, versión PRO.

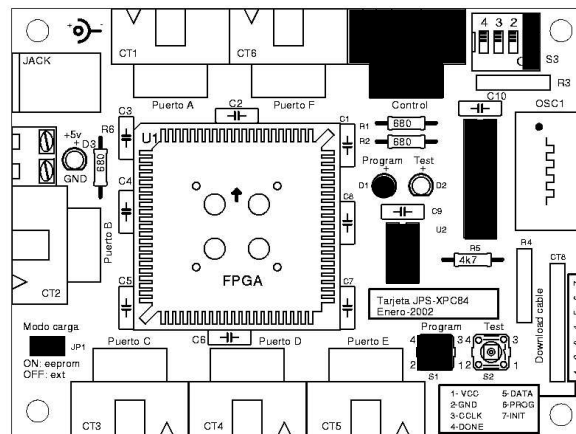


Figura 4.2.: Situación de los componentes de programación

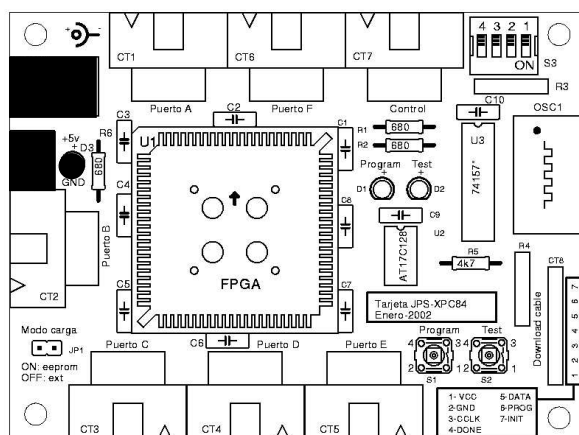


Figura 4.3.: Situación de los conectores de alimentación

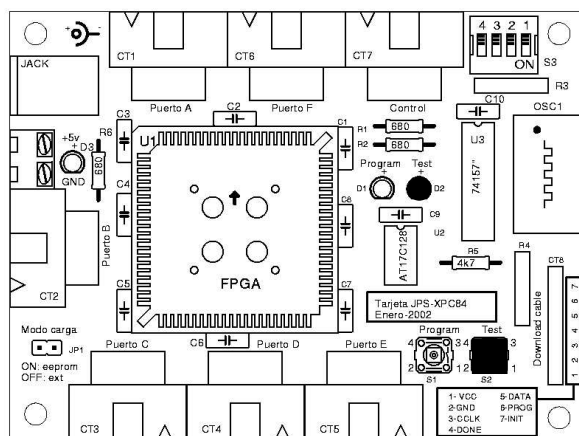


Figura 4.4.: Situación del pulsador y LED de pruebas

4. Aspecto físico

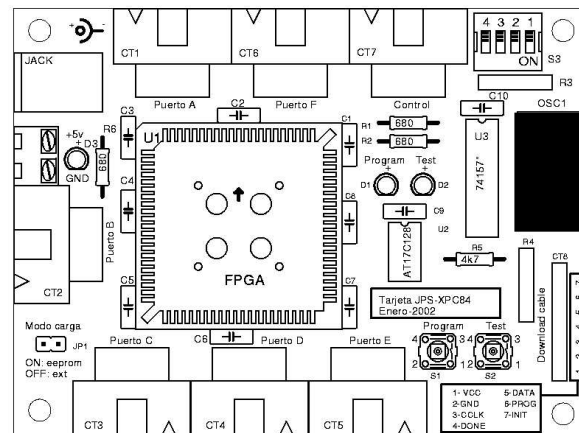


Figura 4.5.: Situación del circuito de reloj

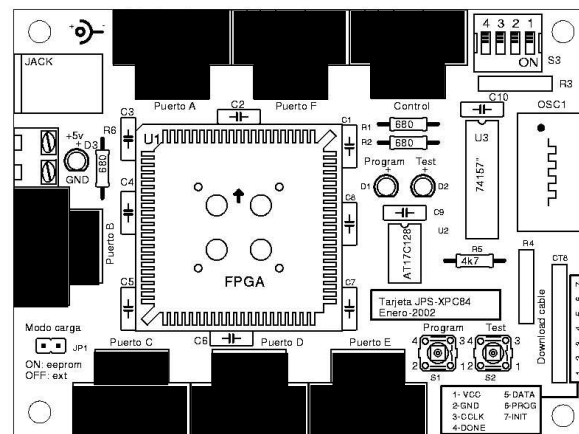


Figura 4.6.: Situación de los puertos de conexión

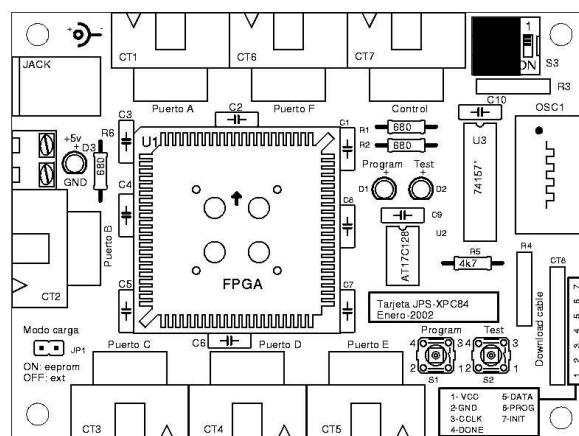


Figura 4.7.: Situación de los conmutadores genéricos

4. Aspecto físico

5. Modos de programación

5.1. Introducción

La tarjeta JPS-XPC84 permite **dos modos de configuración** de una FPGA, ya sea de la familia Spartan o de la 4000. Estos son el **modo maestro** (*master*) y el **modo esclavo serie** (*slave*). En el primero, la FPGA es la que lleva la iniciativa de la configuración mientras que en el segundo no. Ambos modos se basan en que los datos se transmiten en serie hacia la FPGA y ésta los captura en los flancos de una señal de reloj denominada **CCLK**. En el **modo maestro** la FPGA se encarga de proporcionar el reloj, mientras que el **modo esclavo** este reloj es externo.

El modo en el que se encuentra operando la FPGA se selecciona mediante el *jumper JP1*. Si JP1 está puesto, el modo es maestro, y si se encuentra quitado el modo es esclavo. Los pines de la FPGA que indican el modo son P30(M1), P32(M1), P34(M2).

La **familia 4000** tiene siete modos de configuración, incluyendo el modo esclavo y maestro, y utilizan los tres pines para seleccionarlos. En cambio la **familia Spartan** sólo tiene dos modos, por lo que únicamente utiliza el pin M0. En la tarjeta los pines se encuentran cortocircuitados de forma que en un momento dado (dependiendo del *jumper JP1*) sólo se pueden dar una de las siguientes configuraciones: 000 ó 111. Tanto en la familia 4000 como la Spartan la primera configuración indica modo maestro y la segunda modo esclavo.

La Tabla 5.1 se resumen los dos modos permitidos, la configuración de los bits M0, M1 y M2, si la señal de reloj la genera la FPGA (salida) o externamente(entrada) y cómo se suministran los datos, si en serie o en paralelo.

En cualquiera de los dos modos el led D1 (ver Figura 2.1) se mantendrá encendido durante todo el proceso de programación, y se apagará cuando la FPGA esté correctamente cargada. **Si D1 no se apaga entonces significa que la FPGA no se ha programado correctamente.**

5.2. Modo de configuración maestro

		M0	M1	M2	JP1
Modo maestro	EEPROM	0	0	0	ON

Modo	M2	M1	M0	CCLK	Datos
Maestro serie	0	0	0	salida	Bit-serie
Esclavo serie	1	1	1	entrada	Bit-serie

Cuadro 5.1.: Modos de configuración de la tarjeta JPS-XPC84

5. Modos de programación

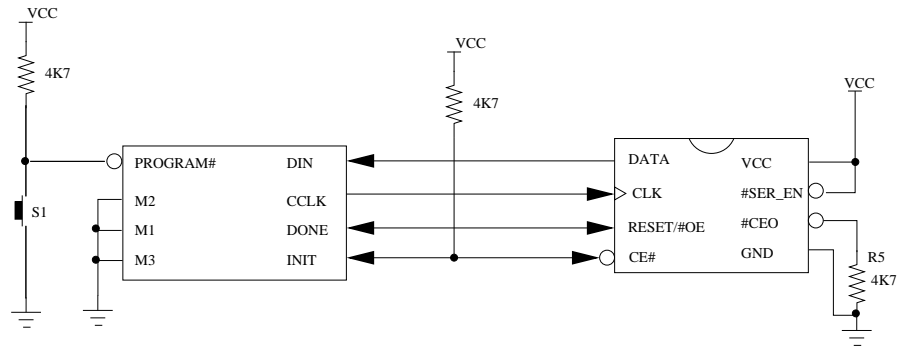


Figura 5.1.: Conexiones entre la FPGA y la memoria serie AT17C128

AC Characteristic

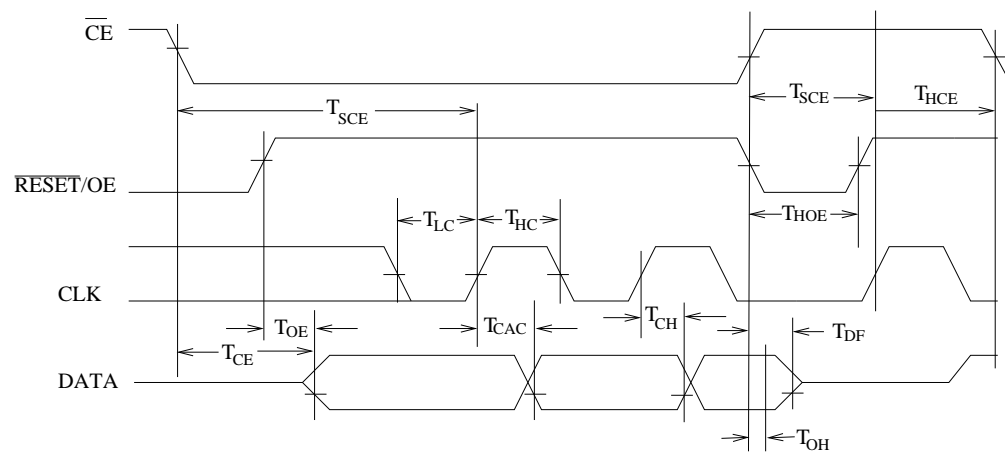


Figura 5.2.: Cronograma del proceso de lectura de una EEPROM serie AT17C128

En este modo la FPGA lee la configuración de una memoria EEPROM serie colocado en el zócalo U2 (ver Figura 2.1). La placa ha sido probada con la memoria AT17C128 de la casa ATMEL[8]. Para iniciar la programación en modo maestro se utiliza el pulsador S1 (ver Figura 2.1), de tal forma que cada vez que se pulse se cargará el diseño almacenado en la FPGA. En este modo, también se programará la FPGA cada vez que se conecte la alimentación al circuito. En la Figura 5.1 se muestra el esquema de conexión de la memoria EEPROM con la FPGA.

En la Figura 5.2 se muestra el cronograma de las señales que provee la FPGA para leer la memoria serie.

5.3. Modo de configuración esclavo

		M0	M1	M2	JP1
Modo esclavo	Sistema externo o PC	1	1	1	OFF

AC Characteristics for AT17C65/128/256

Symbol	Description	Min	Max	Units
T _{OE}	OE to Data Delay		30	ns
T _{CE}	CE to Data Delay		45	ns
T _{CAC}	CLK to Data Delay		50	ns
T _{OH}	Data Hold from CE, OE, or CLK	0		ns
T _{OF}	CE or OE to Data Float Delay		50	ns
T _{LC}	CLK Low Time	20		ns
T _{HC}	CLK High Time	20		ns
I _{SCE}	CE Setup Time to CLK (to guarantee proper counting)	35		ns
T _{HCE}	CE Hold Time from CLK (to guarantee proper counting)	0		ns
T _{HOE}	OE High Time (Guarantees counter is reset)	20		ns
F _{MAX}	Maximum Input Clock Frequency	12.5		MHZ

Figura 5.3.: Características de la EEPROM serie AT17C128

La configuración se realiza desde un sistema externo. Se han habilitado dos conectores distintos CT7 y CT8.

AVISO: La tarjeta no se configurará correctamente en este modo si la EEPROM serie se encuentra colocada en el zócalo U2, independientemente de la posición del jumper JP1

En **modo esclavo serie** la FPGA recibe los datos en serie por el pin DIN y los captura en los flancos de subida de la señal CCLK, que se proporciona externamente. En la Figura 5.4 se muestra los cronogramas de ambas señales y los tiempos mínimos que hay que respetar.

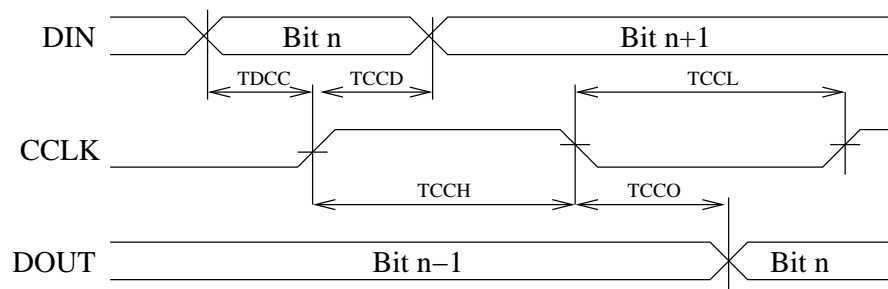
Para comenzar el proceso de configuración hay que poner a nivel bajo la señal PROGRAM (pin 55). La FPGA pondrá a nivel alto la señal INIT (pin p41). Cuando la FPGA haya sido correctamente programada la señal DONE se pondrá a nivel alto, e indicará al sistema que la esté configurando que el proceso se ha realizado correctamente.

5.4. Puerto de control

Este puerto lo constituye el **conector CT7**. Esta pensado para programar la FPGA desde otra tarjeta¹. En la Figura 5.5 se muestra el conexionado de pines de puerto de control.

¹Al día de escribir esta documentación (8-dic-2002) todavía no se probado este puerto, por lo que no se puede garantizar al 100 % su correcto funcionamiento.

5. Modos de programación



Symbol	Description	Min	Max	Units
TDCC	DIN setup	20		ns
TCCD	DIN hold	0		ns
TCCO	DIN to DOUT		30	ns
TCCH	High Time	40		ns
TCCL	Low time	40		ns
FCC	Frequency		10	MHZ

Figura 5.4.: Señales de configuración de la FPGA en modo esclavo

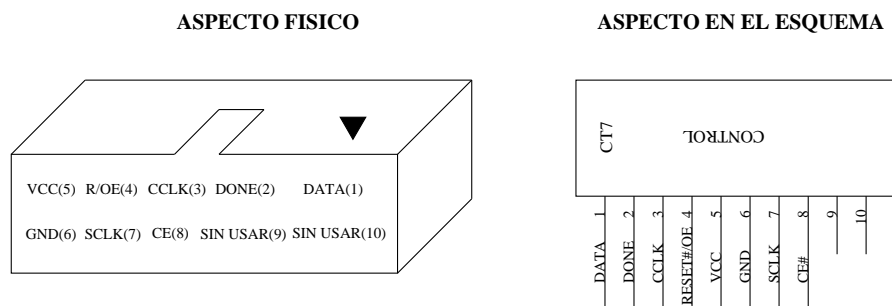


Figura 5.5.: Configuración de pines del puerto de control

Conector	1	2	3	4	5	6	7	8	9	10
P1	DATA	DONE	CCLK	RESET#/OE	VCC	GND	SCLK	CE#		

Cuadro 5.2.: Distribución de pines del puerto de control

	1	2	3	4	5	6	7	8
CT8	VCC	GND	CCLK	DONE	DATA	PROG	INIT	RST
Cable	VCC	GND	CCLK	D/P	DIN	PROG	INIT	RST

Cuadro 5.3.: Correspondencia de los pines del conector CT8 con el cable de descarga suministrado por Xilinx



Figura 5.6.: Aspecto del cable de descarga paralelo suministrado por Xilinx

El puerto de control no tiene entrada para la señal PROGRAM, lo cual obliga a que el usuario tenga que iniciar manualmente el proceso de configuración accionando el pulsador S1.

5.5. Programación de la FPGA desde el PC

El conector CT8 se ha diseñado para facilitar la programación de la FPGA desde un PC utilizando las herramientas desarrollados por la empresa Xilinx[1]. La conexión entre en el PC y la tarjeta se realiza utilizando un cable especial suministrado por Xilinx denominado cable de descarga (ver Figura 5.6). El cable se conecta por un lado al puerto serie de PC, por el otro a cada uno de los ocho pines torneados del conector CT8. A cada pin torneado se conecta una de las ocho terminaciones del cable. Cada terminación tiene un color distinto y una etiqueta identificativa. La correspondencia entre los pines del conector y de las etiquetas de las terminaciones se resume en la tabla 5.3.

5. Modos de programación



Figura 5.7.: Foto del conector CT8

En la Figura 5.7 se observa el aspecto real del conector CT8. Junto al conector se ha colocado en la serigrafía de la tarjeta un cuadro recordatorio de la configuración de los pines.

5.6. Programación de la EEPROM desde el puerto de control

En el caso que se utilice una memoria **AT17C128** se dispone de la posibilidad de programar la memoria “in circuit”. Para ello se utiliza el **puerto de control** (ver Figura 5.5), a través del cual otro sistema, como puede ser un microprocesador u otra FPGA, puede programar la memoria serie sin necesidad de tener que quitar del zócalo la memoria.

Para poder habilitar la programación de la EEPROM **el pin 1 del switch de configuración (5.8) tiene que estar en ON**. Mientras que la tarjeta se encuentre en este modo la FPGA no se puede cargar desde la memoria EEPROM. Tampoco se recomienda programar la FPGA desde el PC mientras que la tarjeta se encuentra en modo programación de EEPROM.

La tabla 5.2 muestra la distribución de pines del puerto de control. La Figura 5.9 representa el conexionado propuesto por ATMEL para realizar la programación de la EEPROM. En la placa JPS-XPC84 se ha respetado el esquema con la salvedad de que la señal SER_EN# se encuentra conectada al bit 0 del switch de configuración(S3, ver 5.8). Esta señal es la que habilita

5.6. Programación de la EEPROM desde el puerto de control

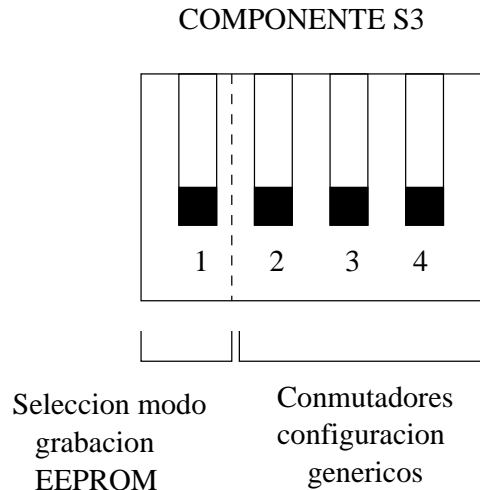


Figura 5.8.: Componente S3, conmutadores

si la EEPROM se va a programar o a leer, por lo que con este bit se selecciona el modo de funcionamiento de la EEPROM.

Se utiliza un multiplexor de dos canales de 4 bits para que las señales que lleguen a la EEPROM procedan bien de la FPGA(o del Download cable) o bien de un sistema grabador externo. Así por ejemplo, la señal de reloj CLK que llega a la EEPROM, puede venir de la señal CCLK de la FPGA o bien de la señal de reloj SCLK externa, según el estado de la entrada de selección S. Lo mismo ocurre con las señales RESET/#OE y CE#.

En la programación de la EEPROM intervienen cinco señales: SER_EN#, CE#, RESET/#OE, SCLK y DATA. La señal **SER_EN#** es la que conmuta el modo de funcionamiento de la memoria, si está a nivel alto se puede leer y si esta a nivel bajo se puede escribir. Esta señal se controla mediante el **pin 1 del switch de configuración**, así que debe estar activada antes de empezar a programar.

La señales **CE#**, **RESET/#OE**, **SCLK** y **DATA** controlan el flujo de la programación y tienen que ser suministradas por el sistema externo. Las señales CE#, RESET/#OE tienen que ser activadas durante todo el proceso de la programación. CE# se activa a nivel bajo mientras que RESET/#OE se activa a nivel alto. Los datos que se quieren introducir en la memoria se suministrarán en serie por la señal DATA, y estos serán capturados en cada flanco bajada de la señal SCLK. (Para más información consultar el la hoja de datos de la familia AT17XX de Atmel).

5. Modos de programación

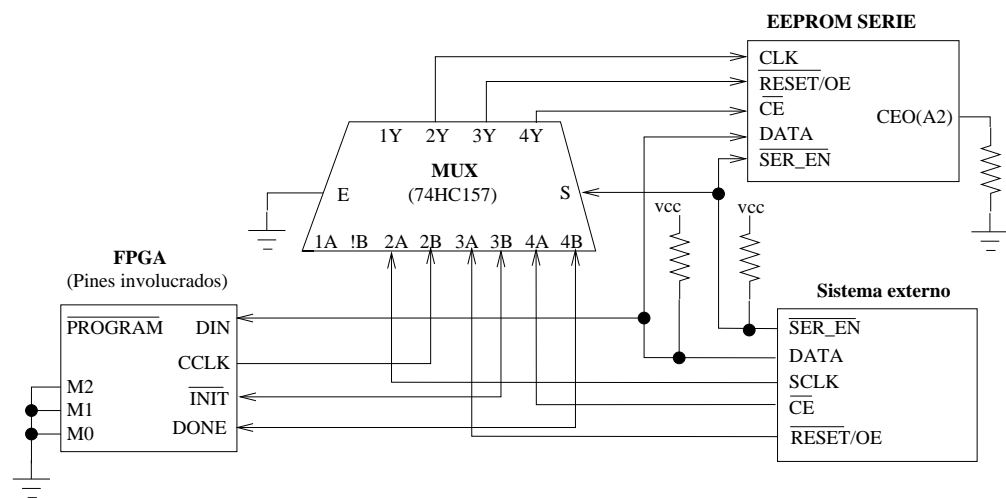


Figura 5.9.: Conexión para programar la memoria EEPROM desde un sistema externo. Proporcionado por ATMEL.

6. Puertos

6.1. Puertos de expansión

La tarjeta JPS-XPC84 consta de siete puertos de diez pines cada uno que permiten acceder a los patas de entrada/salida de la FPGA. Cada puerto presenta un pin de alimentación, un pin de masa y ocho pines de entrada/salida genéricos. En la siguientes tablas queda expuesta la correlación entre los pines de cada conector y el pin correspondiente de la FPGA.

■ **Puerto A (CT1):**

Conector	1	2	3	4	5	6	7	8	9	10
FPGA	P10	P8	P6	P4	VCC	GND	P3	P5	P7	P9

■ **Puerto B (CT2):**

Conector	1	2	3	4	5	6	7	8	9	10
FPGA	P25	P23	P19	P14	VCC	GND	P13	P18	P20	P24

■ **Puerto C (CT3):**

Conector	1	2	3	4	5	6	7	8	9	10
FPGA	P39	P38	P29	P27	VCC	GND	P26	P28	P35	P40

■ **Puerto D (CT4):**

Conector	1	2	3	4	5	6	7	8	9	10
FPGA	P49	P50	P48	P46	VCC	GND	P44	P45	P47	P51

■ **Puerto E (CT5):**

Conector	1	2	3	4	5	6	7	8	9	10
FPGA	P62	P60	P58	P56	VCC	GND	P57	P59	P61	P65

■ **Puerto F (CT1):**

Conector	1	2	3	4	5	6	7	8	9	10
FPGA	P83	P82	P80	P78	VCC	GND	P77	P79	P81	P84

En la figura 6.1 se muestra como quedan gráficamente los pines asociados a cada puerto.

6. Puertos

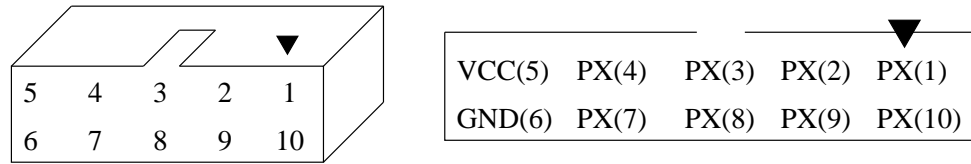


Figura 6.1.: Configuración de los pines de un puerto de expansión

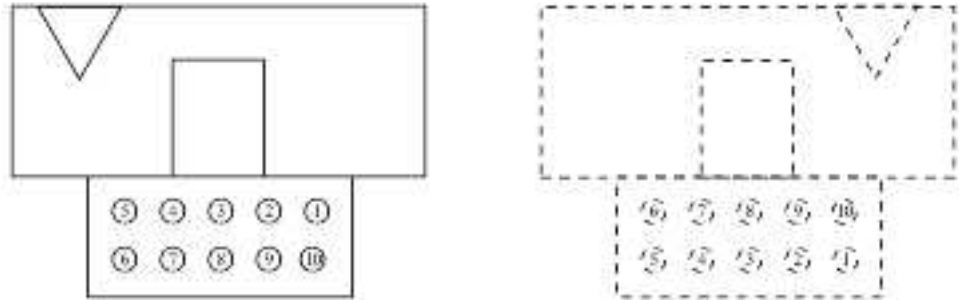


Figura 6.2.: Disposición de los pines del puerto de expansión en la tarjeta. La figura izquierda muestra el lado de los componentes (cara de arriba) y la derecha el lado de la soldadura (cara de abajo).

6.2. Temporización de los diseños

La tarjeta JPS-XPC84 permite utilizar un reloj integrado en placa. Para ello se ha reservado el pin 72 de la FPGA. El diseñador cuando utilice el reloj de la tarjeta tendrá que indicar explícitamente en el diseño que el **pin 72 funciona como un pin de reloj secundario**.

El reloj se conecta al componente OSC1. También se pueden conectar relojes externos a través de los puertos de expansión. Dentro de cada puerto existen algunos pines que aparte de utilizarlos como pines de entrada/salida genéricos se pueden configurar como entrada de reloj, ya sea primario o secundario. La tabla 6.1 resume qué pines se pueden utilizar como entrada de reloj, a qué puerto pertenecen y si son relojes primarios o secundarios.

PIN	Puerto	Tipo de reloj
P13	Puerto B	Primario (PCGK)
P35	Puerto C	Primario (PCGK)
P57	Puerto E	Primario (PCGK)
P78	Puerto F	Primario (PCGK)
P10	Puerto A	Secundario (SCGK)
P29	Puerto C	Secundario (SCGK)
P51	Puerto D	Secundario (SCGK)

Cuadro 6.1.: Pines de los puertos de extensión que se pueden configurar como entradas de reloj

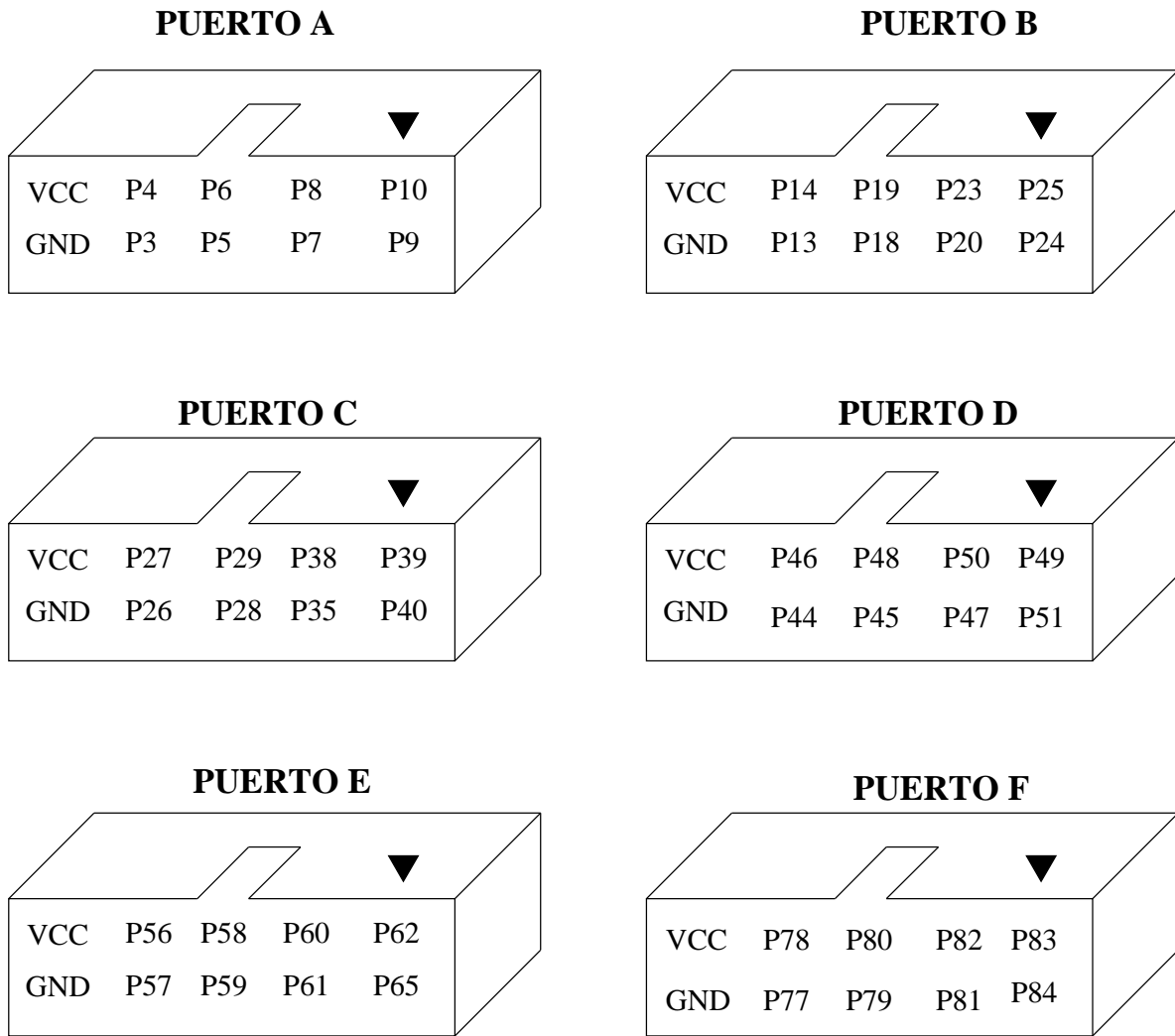


Figura 6.3.: Los puertos de expansión

6. Puertos

Switch-PIN	FPGA-PIN	Nombre
2	P66	CONFIG_0
3	P67	CONFIG_1
4	P70	CONFIG_2

Cuadro 6.2.: Correspondencia de los switches de configuración genéricos con pines de la FPGA

6.3. Switches de configuración genéricos

La tarjeta JPS-XPC84 incluye tres switches que se conecta a pines de la FPGA. Estos switches se utilizan como entradas genéricas de configuración. Cada switch está conectado a VCC mediante una resistencia de pull-up y cuando se encuentra activado se conecta directamente a masa. Los diseñadores disponen de hasta 8 posibles combinaciones de entrada. Los switches genéricos se corresponden con los pines del 2 al 4 del componente S3. (ver 5.8). En la tabla 6.1 se muestra gráficamente los distintos aspectos del componente S3 y qué resultados da.

6.4. Pulsador genérico

La tarjeta JPS-XPC84 dispone de un **pulsador genérico conectado al pin P69**. Se utiliza para comprobar que la tarjeta configura correctamente una FPGA, pero también puede ser utilizado como una entrada por los diseñadores. Cada vez que se pulsa el **pin 69** se conecta a masa, mientras que en posición de reposo se encuentra conectado a VCC.

6.5. Led genérico

A parte del pulsador los diseños de test utilizan un **led prueba que se conecta como al pin P68**. Los diseñadores también pueden utilizarlo como indicador luminoso en sus diseños. Para ello hay que configurar el **pin 68** como salida. El led se enciende siempre que se ponga un uno lógico en dicho pin.

6.6. Distribución global

A continuación se muestran todos los pines de la FPGA y su correspondiente nombre en la placa JPS-XPC84.

PIN FPGA	Nombre	Descripción
P1	GND	
P2	VCC	
P3	PA7	Puerto A
P4	PA4	Puerto A
P5	PA8	Puerto A
P6	PA3	Puerto A

6.6. Distribución global

P7	PA9	Puerto A
P8	PA2	Puerto A
P9	PA10	Puerto A
P10	PA1	Puerto A
P11	VCC	
P12	GND	
P13	PB7	Puerto B
P14	PB4	Puerto B
P15	TDI	No conectado. No se utiliza
P16	TCK	Conectado a VCC
P17	TMS	Conectado a VCC
P18	PB8	Puerto B
P19	PB3	Puerto B
P20	PB9	Puerto B
P21	GND	
P22	VCC	
P23	PB2	Puerto B
P24	PB10	Puerto B
P25	PB1	Puerto B
P26	PC7	Puerto C
P27	PC4	Puerto C
P28	PC8	Puerto C
P29	PC3	Puerto C
P30	M1	Bit 1 Configuración maestro/esclavo. Conectdo al jumper JP1
P31	GND	
P32	M0	Bit 0 configuración maestro/esclavo. Conectado al jumper JP1
P33	VCC	
P34	M2	Bit 2 configuración maestro/esclavo. Conectado al jumper JP1
P35	PC9	
P36	HDC	No conectado
P37	LDC	No conectado
P38	PC2	Puerto C
P39	PC1	Puerto C
P40	PC10	Puerto C
P41	INIT	
P42	VCC	
P43	GND	
P44	PD7	Puerto D
P45	PD8	Puerto D
P46	PD4	Puerto D
P47	PD9	Puerto D

6. Puertos

P48	PD3	Puerto D
P49	PD1	Puerto D
P50	PD2	Puerto D
P51	PD10	Puerto D
P52	GND	
P53	DONE	
P54	VCC	
P55	PROG	Pin Program de la FPGA
P56	PE4	Puerto E
P57	PE7	Puerto E
P58	PE3	Puerto E
P59	PE8	Puerto E
P60	PE2	Puerto E
P61	PE9	Puerto E
P62	PE1	Puerto E
P63	VCC	
P64	GND	
P65	PE10	Puerto E
P66	CONFIG_0	
P67	CONFIG_1	
P68	TEST_OUT	Led de pruebas
P69	TEST_IN	Pulsador de pruebas
P70	CONFIG_2	
P71	DATA	
P72	RELOJ	
P73	CCLK	
P74	VCC	
P75	TDO	No conectada
P76	GND	
P77	PF7	Puerto F
P78	PF4	Puerto F
P79	PF8	Puerto F
P80	PF3	Puerto F
P81	PF9	Puerto F
P82	PF2	Puerto F
P83	PF1	Puerto F
P84	PF10	Puerto F

7. Alimentación

La JPS-XPC84 tiene una tensión nominal de alimentación de 5V, aunque se puede alimentar entre el rango de 4.75V a 5.25V. Los modelos de FPGAs, tanto de la familia 4000 como Spartan, que se alimenten con 3.3V no podrán ser utilizadas en esta placa.

Dispone de dos conectores distintos de alimentación, uno de tipo clema (CLEMA1) y otro de tipo jack (JACK1). Estos dos conectores facilitan distintas formas de alimentación, como por ejemplo mediante una fuente de alimentación, pilas/baterías, o un transformador conectada a la red.

En la Figura 7.1 se ha representado el conector hembra de alimentación y el jack macho cilíndrico. Es importante destacar que **la parte exterior del jack macho deber ser GND**, mientras que **la parte interior debe ser VCC**. En la mayoría de los transformadores comerciales, la polaridad se puede invertir cambiando un switch. **Antes de conectar un transformador hay que asegurarse de que la polaridad es la correcta: masa en la parte exterior(-) y alimentación(+) en la parte interior.**

El conector de tipo clema se compone de dos bornas, cada una tiene un tornillo que permite fijar los cables de alimentación que se introduzcan. La polaridad se encuentra impresa en la tarjeta.

Los terminales de VCC de la clema y del jack se encuentran cortocircuitados, lo mismo que los terminales GND. Por esto, **nunca se debe alimentar la tarjeta utilizando a la vez los dos conectores.**

En la figura 7.2 se muestra la tarjeta JPS alimentada a través de 4 pilas de 1.5voltios. Este tipo de alimentación es muy cómoda para realizar sistemas autónomos.

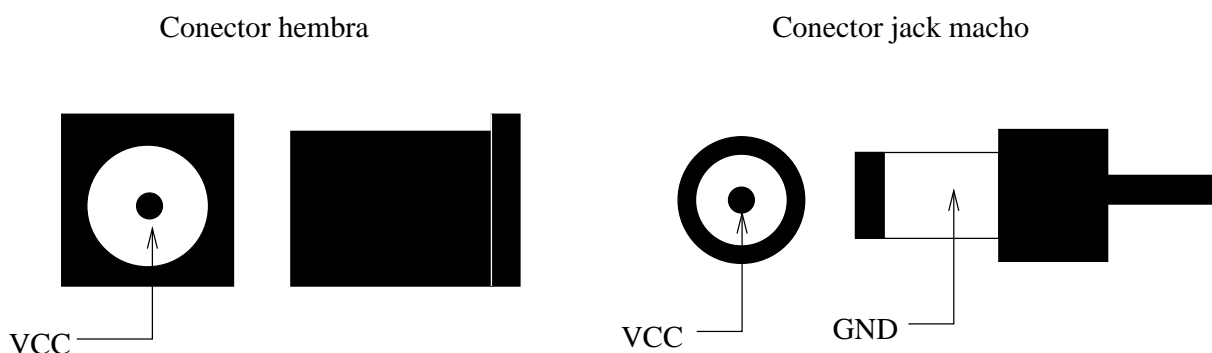


Figura 7.1.: Conector hembra y jack macho para la alimentación

7. Alimentación

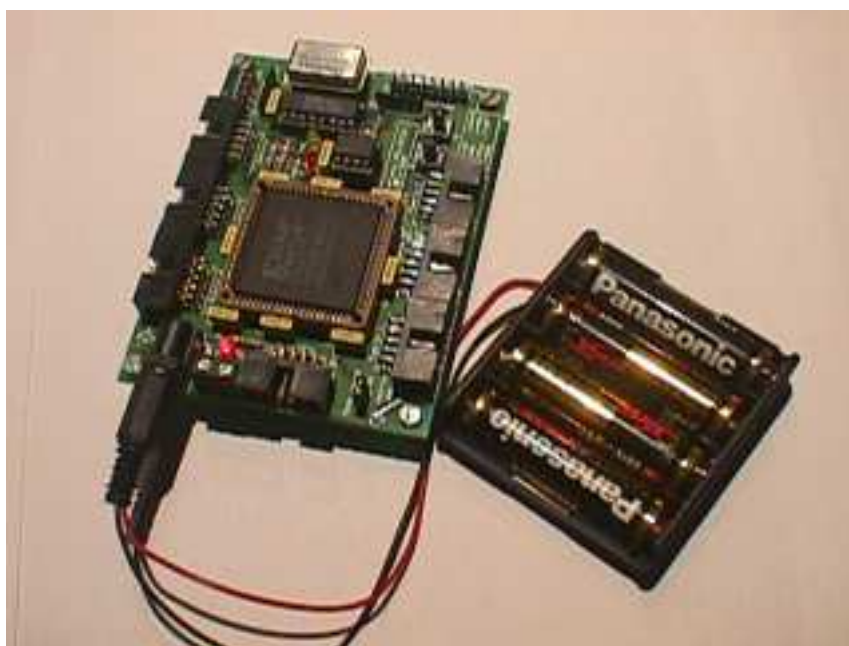


Figura 7.2.: Alimentación de la tarjeta JPS mediante 4 pilas de 1.5 voltios

8. Montaje

8.1. PCB

El PCB es a doble cara. En la figura 8.1 se puede ver la cara de arriba y en la figura 8.2 la cara de abajo.

8.2. Componentes

A continuación se detallan los componentes necesarios para construir la placa JPS-XPC84. En la tabla 8.1 se describe el componente y su correspondencia dentro del esquema de la Figura 2.1.

8.3. Disposición de los componentes

En la figura 8.3 se muestra una foto de la tarjeta JPS en la que se indican algunos de los componentes, para que sirvan de referencia a la hora de montar la placa a partir del PCB.

A la hora de realizar el montaje hay que tener en cuenta lo siguiente:

- Cuidado al colocar los arrays de resistencias R3 y R4. Pata común, marcada con un punto en los arrays, debe situarse sobre el pad cuadrado. Según tenemos la placa como se indica en la figura 8.3, la pata común del array R3 debe estar a la derecha y la del array R4 arriba.
- La placa está pensada para usarse switches acodados, de manera que si se sitúa otra placa encima, se tenga acceso a ellos sin problemas. Sin embargo, es posible situar switches no acodados, quizás porque el proveedor no tenga otros. En ese caso puede surgir la duda de cómo colocarlos. Se puede colocar en cualquiera de sus posiciones, sin embargo, para que estén acordes a esta documentación, el microswitch que tenga el número 1 deberá estar en el lado derecho, según se mira la placa como se indica en la figura 8.3 (aunque el número quede al revés).
- ¡¡Cuidado con la polaridad de los led!!! Los leds son el único medio para saber rápidamente si la placa está funcionando o no. Prestar atención a la polaridad a la hora de conectarlos.

8. Montaje

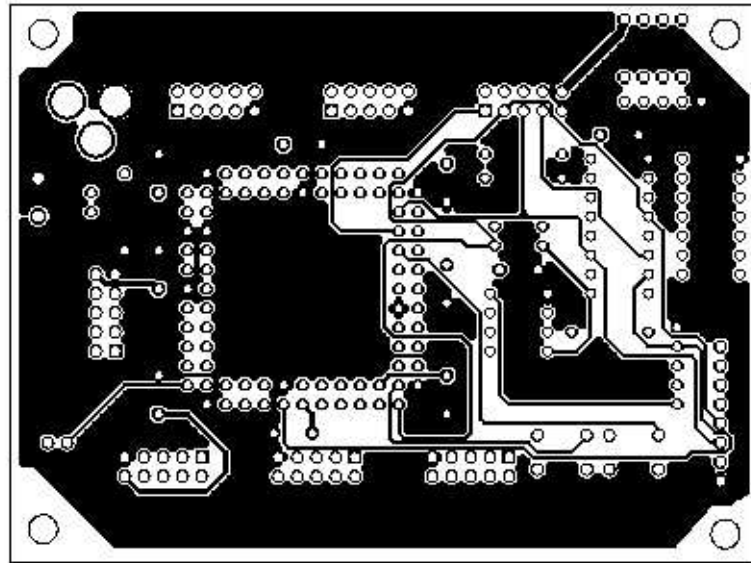


Figura 8.1.: PCB, cara de arriba

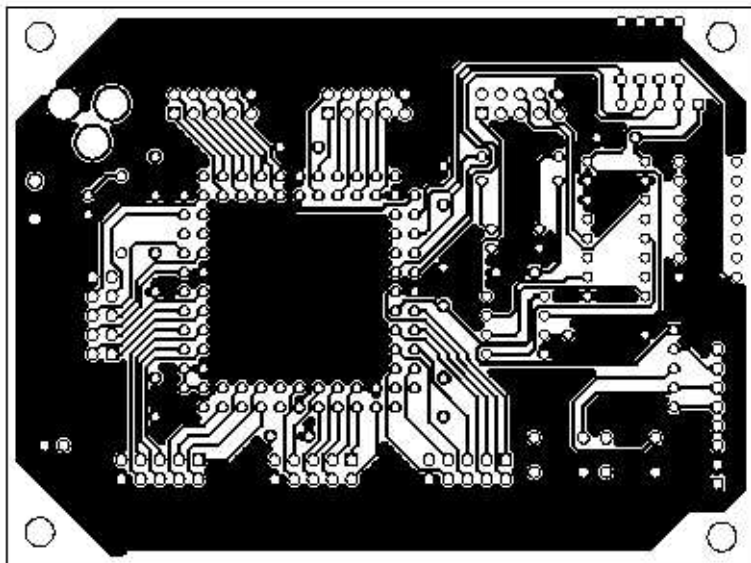


Figura 8.2.: PCB, cara de abajo

8.3. Disposición de los componentes

Cantidad	Descripción	Ref.	Notas
1	Zócalo PLCC84	U1	Para ubicar la FPGA
1	Zócalo 2x4 (8-lead LAP)	U2	Para ubicar memoria AT17C128
1	Zócalo 2x8	U3	Para ubicar multiplexor 74HC157
1	Zócalo 2x7	OSC1	Para ubicar un reloj
10	Condensadores de desacoplo	C1-C10	Condensadores de desacoplo
1	Led rojo 3mm	D1	Led de programación
1	Led rojo 3mm	D3	Led de alimentación
1	Led verde (ó amarillo) 3mm	D2	Led de pruebas
2	Pulsadores	S1-S2	Programación y pruebas
3	Resistencias 1/4w 680Ω	R1,R2,R6	Polarización leds
2	Array resistencias 4+1, 4K7	R3-R4	Resistencias de pull-up
1	Resistencia de 4K7	R5	Resistencia de pull-down
7	Conector acodado macho, 10 vías	CT1-CT7	Puertos de expansión y control
1	Tira de pines machos, no torneados	CT8	Conector cable JTAG
1	Jumper	JP1	Jumper modo programación
1	Clema para circuito impreso, 2 vías	CLEMA1	Alimentación
1	Jack hembra para circuito impreso	JACK1	Alimentación
1	Array de 4 switches	S3	Switch de configuración
4	Separadores hembra-hembra + tornillo		Soportes para la placa

Cuadro 8.1.: Listado de componentes de la placa JPS-XPC84

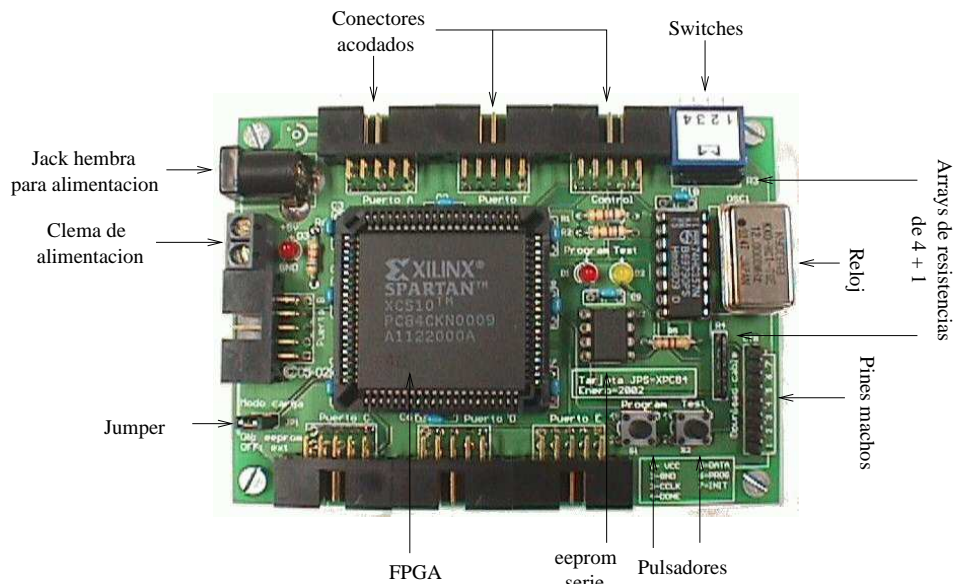


Figura 8.3.: Disposición de componentes en la placa JPS

8.4. Pruebas

Una vez montada la placa, se pueden hacer las siguientes pruebas:

1. Alimentar la placa a 5 voltios y comprobar si se enciende el led de power-on. Si no es así, puede ser que esté colocado al revés, que no hayamos soldado correctamente los conectores de alimentación o que no estemos alimentando correctamente.
2. Con la FPGA colocada (y la eeprom no) el led indicado como **program** debe estar **encendido**. Esto indica que la FPGA NO está cargada. Si no es así conviene revisar si lo hemos colocado con la polaridad adecuada.
3. Conseguir una eeprom grabada con algún programa para nuestra FPGA (si no tenemos pasar al siguiente punto). Colocar el jumper Jp1, y poner el switch 1 en posición ON¹. En esta situación, y si el programa grabado en la eeprom es correcto, al pulsar el botón de Program se encenderá el led de program durante medio segundo aproximadamente. Una vez que está apagado, la FPGA estará cargada correctamente.
4. Conectar la placa JPS a través del “download” cable y comprobar si con el software de Xilinx se pueden descargar los diseños. (Puede ocurrir que no se cargue correctamente aunque la placa esté bien montada. A lo mejor es que no está el software bien configurado)

¹El que haya que colocarlo a ON o a OFF depende del switch que hayamos comprado y de cómo lo hayamos colocado. Lo que hay que garantizar es que NO haya continuidad entre las patas del switch 1. También se puede medir que haya 5v por una de las dos patas, con la placa alimentada.

9. Trabajando con la JPS

9.1. Esquema de trabajo

La forma de trabajo es la habitual con las FPGA's. El software con el que se ha probado la placa y con el que se ha hecho el ejemplo del inversor es el **Xilinx Foundation Series F3.1i**, para Windows.

El proceso de trabajo se resume en la figura 9.1. El diseño se puede realizar directamente dibujando el esquema o bien describiéndolo en VHDL. Esta segunda opción es la recomendada puesto que así nuestros diseños serán más portables y se podrán sintetizar con otro programa diferentes al Xilinx Foundation.

El primer paso es la **simulación**, para comprobar si el circuito funciona correctamente y en caso contrario poder detectar dónde están los problemas.

Lo siguiente es **sintetizar el diseño**, que consiste en obtener el fichero **bitstream** necesario para que la FPGA se reconfigure. En este proceso tenemos que utilizar opcionalmente el fichero de restricciones, que tiene extensión .ucf, en el que indicamos por qué pin de la FPGA queremos que entre o salga cada uno de las señales de nuestro diseño. Para la creación de este fichero necesitamos conocer cuáles son los pines que están disponibles para ser usados con la tarjeta JPS. Ver apartado 6.6. Por ejemplo el Pin P68 de la FPGA está conectado en la tarjeta JPS al led de pruebas (test).

Una vez generado el bitstream, lo podemos **descargar** directamente a la JPS utilizando el *download cable* de Xilinx (Ver apartado 5.5). Para ello es necesario quitar el jumper JP1 y sacar la eeprom de su zócalo.

Para grabar el bitstream en la memoria eeprom obtener un diseño autónomo del PC es necesario primero generar un fichero en formato HEX de intel, que tiene extensión .MCS. Este fichero lo podemos leer con un grabador que soporte las memorias ATMEL AT17C128 y grabar la eeprom. Finalmente la introducimos en la placa JPS, ponemos el jumper JP1 y colocamos correctamente el switch 1.

9.2. Un ejemplo muy sencillo: un inversor

9.2.1. El esquema

Vamos a realizar el diseño más sencillo posible, un inversor (puerta NOT) que se coloque entre los pines **P69** y **P68** de la FPGA, que se corresponden con el **pulsador** y **led de pruebas** de la tarjeta JPS. Cuando lo tengamos funcionando, cada vez que se apriete el pulsador se encenderá el led de test.

9. Trabajando con la JPS

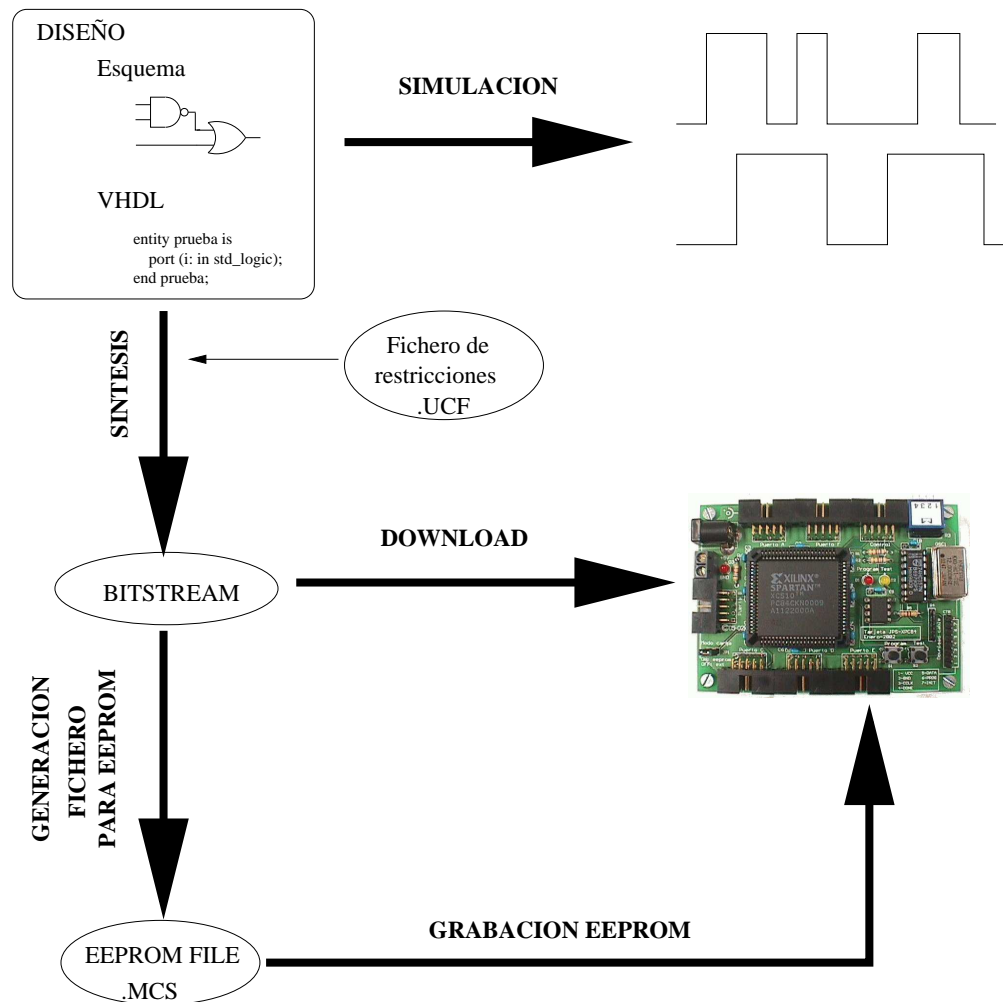


Figura 9.1.: Esquema de trabajo con la JPS

9.2. Un ejemplo muy sencillo: un inversor

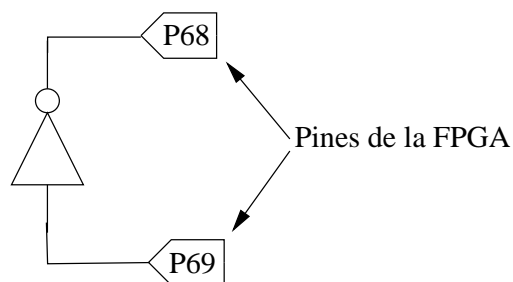


Figura 9.2.: El esquema del inversor de pruebas

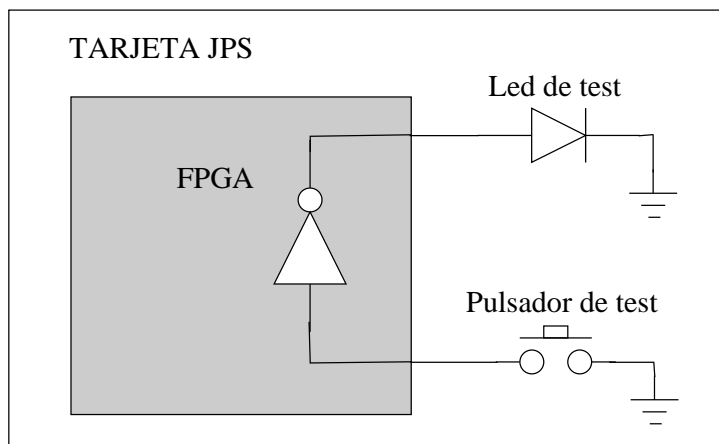


Figura 9.3.: Diseño integrado en la tarjeta JPS

El diseño se muestra en la figura 9.2. Cuando lo tengamos dentro de la FPGA, se comportará como el diseño mostrado en la figura 9.3.

9.2.2. El diseño en VHDL

El diseño en vhdl se encuentra en el fichero **inversor.vhd**, que se muestra a continuación:

```
-----  
-- inversor.vhdl  (c) Juan Gonzalez. Dic   2002  --  
-----  
-- Ejemplos para aprender VHDL                  --  
--                                                --  
-- Creacion de una entidad que es un inversor    --  
-- Licencia GPL                                  --  
-----  
  
library ieee;  
use ieee.std_logic_1164.all;
```

9. Trabajando con la JPS

```
-----  
-- Se define una entidad que es un inversor  
-- i: Entrada  
-- o: Salida  
-----  
entity inversor is  
    port (i : in std_logic; o: out std_logic);  
end inversor;  
  
architecture ttl of inversor is  
begin  
    o <= not i;  
end ttl;
```

La entrada del inversor se ha llamado *i* y la salida *o*. Esto será necesario conocerlo para definir el fichero restricciones.

9.2.3. El fichero restricciones: inversor.ucf

Este fichero contiene la información que relaciona las señales de mi diseño con los pines de la FPGA. Para el diseño del inversor tiene el siguiente aspecto:

```
NET "i" LOC="P69";  
NET "o" LOC="P68";
```

La señal “i”, que es la entrada de nuestro inversor se asocia al pin 69 de la FPGA, que en la tarjeta JPS está conectado al pulsador de test. La señal “o” se asocia al pin 68, que se corresponde con el led de pruebas.

Sólo con cambiar este fichero y volver a sintetizar conseguiremos tener un inversor entre dos pines cualquiera de la FPGA.

9.2.4. Otros ficheros: inversor.hex, inversor.mcs

El bitstream generado al sintetizar es el **inversor.hex**, que contiene la información en una única línea, codificada mediante caracteres en ascii en hexadecimal.

Para grabar la eeprom generamos el fichero **inversor.mcs**, en formato HEX de intel. Las primeras líneas del fichero se muestran a continuación:

```
:02000000200000FC  
:10000000FF04E88CF9EA7F7FFDF5D75F7FFDF5B748  
:10001000BFFEFAEEEFBEFEFAFB93FFEFBFFFFEFB63
```

:10002000EFBFFFFEF7DF7F7FFDB7DF7FFFFEAEFF58

.....

9.3. Configuración de jumpers y switches

Para trabajar en modo autónomo (Carga desde la eeprom)

- Asegurarse que el switch 1 esté a ON
- Que el jumper JP1 esté conectado
- Pulsar el botón de programación

Para trabajar en modo entrenador (Carga desde el PC)

- Quitar la eeprom del zócalo
- Conectar el download cable
- Quitar el jumper JP1

9. *Trabajando con la JPS*

10. Fabricación

10.1. Introducción

Una de las características de la placa JPS es que es **libre** y que los **ficheros para su fabricación están disponibles**. Cualquiera, sea empresa o particular la puede fabricar, incluso la puede vender y ganar dinero con ella. La única restricción es que se tiene que entregar junto con todos sus esquemas. Y si realiza alguna modificación también tiene la obligación de publicar los nuevos esquemas e incluirse en la lista de autores.

En este apartado se describen los ficheros necesarios para la fabricación industrial y los datos del fabricante donde se hizo la primera tirada, de 30 unidades.

10.2. Ficheros de fabricación

El fichero para la fabricación es **fabricacion.zip**, disponible en la web de la tarjeta JPS[4]. Dentro de este archivo se encuentran los siguientes ficheros:

- LEEME.TXT: Indicación de los ficheros incluidos en el paquete.
- Ficheros en formato GERBER:
 - **jps-xpc84.cmp**: Cara de componentes
 - **jps-xpc84.plc**: Serigrafía
 - **jps-xpc84.sol**: Cara de las soldaduras (Abajo)
 - **jps-xpc84.stc**: Mascara de soldadura de arriba (Solder Mask)
 - **jps-xpc84.sts**: Máscara de soldadura de abajo (Solder Mask)
- Otros:
 - **jps-xpc84.drd**: Mapa de taladros
 - **jps-xpc84.drl**: Información sobre los taladros empleados

Los ficheros en formato GERBER se pueden visualizar en Linux con la aplicación **gerbv**, que también está disponible para Debian¹

¹Para instalarlo en Debian/Linux sólo hay que hacer: apt-get install gerbv

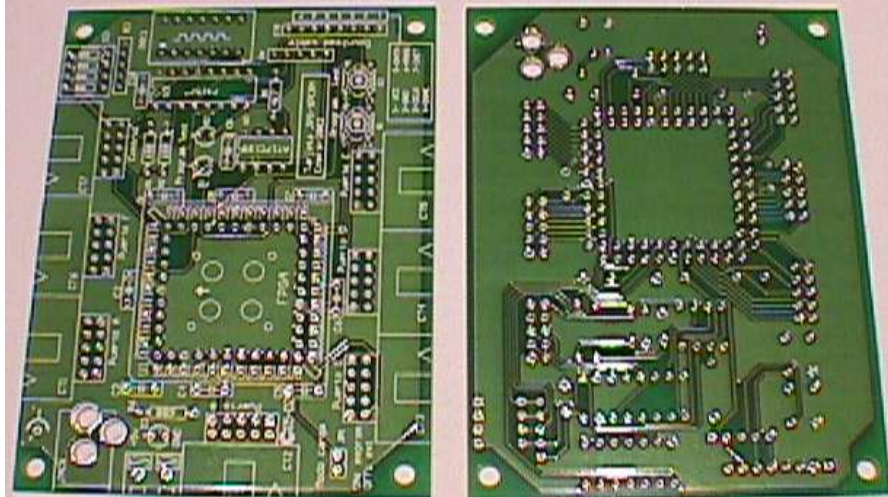


Figura 10.1.: El PCB de la tarjeta JPS, visto por la cara de arriba y la de abajo

10.3. Fabricando la placa

Los pasos seguidos para fabricar la primera tirada fue el siguiente:

1. El fabricante fue Electrocir, S.A[9], con el que ya habíamos trabajado previamente.
2. Se le envió por mail el fichero **fabricación.zip**.
3. A las 2 semanas se recibió por mensajería privada los PCBs :-)

A la hora realizar la fabricación hay que tener en cuenta que hay unos gastos iniciales de preparación que son fijos y que dependen del tipo de placa que se esté construyendo (depende de la superficie, el número de caras, etc...). Como son fijos, cuanto mayor sea la tirada que se saque, más baratas saldrán los PCB's (Suelen rondar entre 180 y 240 euros).

Luego hay un coste por cada PCB sacado.

En la primera tirada de placas JPS que se hizo, cada PCB salió por unos 18 euros (Contando todo: transporte, fabricación, IVA, etc...)

Una vez recibidos los PCB's, se compran los componentes, se sueldan y se prueba la placa. En la figura 10.1 se muestra una foto del PCB, visto por la cara de arriba y la de abajo.

11. Lo que queda por hacer

La lista de tareas pendientes son las siguientes:

- Hacer un software multiplataforma en el PC (Windows/Linux) para poder descargar los bitsream directamente en la JPS, sin usar el software de Xilinx. Esto es especialmente útil para los usuarios de Linux.
- Hacerse un cable libre, usando un microcontrolador, que permita cargar la JPS a través del puerto serie.
- Cargar la FPGA desde un microcontrolador externo, a través del puerto de control
- Hacer un circuito de grabación “in circuit” de la EEPROM, para no tener que usar un grabador externo y no tener que sacar la eeprom del zócalo.
- Biblioteca de diseños para la tarjeta JPS

Cualquier colaboración, propuesta de nuevas mejoras, etc, será bienvenida :-)

11. Lo que queda por hacer

12. FAQs, preguntas más frecuentes

- **¿Qué es la tarjeta JPS?**

Es una tarjeta entrenadora para trabajar con FPGAs. Permite cargar diseños desde el PC, utilizando el cable suministrado por Xilinx, cargarlos desde otro microcontrolador o bien cargarlos desde una eeprom interna, previamente grabada. Es especialmente útil para la docencia y para la realización de robots, ya que podemos implementar nuestro propio hardware a medida.

- **¿Qué significa que sea hardware abierto?**

Significa que están disponibles todos los esquemas, el PCB y los ficheros de fabricación, pero lo más importante, es que se cede el derecho de copia, modificación y redistribución de las modificaciones. Esta placa se puede copiar, fabricar y modificar, con la única restricción de que se tiene que mantener esta nota. Los diseños derivados deben ser también libres.

- **Quiero fabricar la placa JPS, ¿Qué hago?**

En la web están disponibles los ficheros de fabricación, en formato Gerber. En el apartado 10 puedes encontrar más información.

- **¿Puedo vender la placa JPS y ganar dinero con ella?**

Por supuesto. Por ello es una placa libre. Una empresa o un particular tiene licencia para fabricar la placa y venderla, cobrando lo que estime oportuno, sin embargo debe hacer constar que es una placa libre y la debe distribuir junto con todos los ficheros de fabricación, esquemas y PCB. El beneficio que obtiene el que lo fabrica se deriva de ofrecer el servicio de fabricación y no de cobrar derechos sobre una tecnología.

- **Estoy soldando la placa JPS, ¿Cómo conecto los arrays R3 y R4?**

Los arrays de resistencias tienen una pata común, indicada con un punto. Esta pata no se puede situar en cualquier posición. En el apartado 8.3 se dan más detalles. Según se mira la placa de manera que las serigrafías se lean correctamente, la pata común del array R3 se debe situar a la derecha y la del array R4 arriba.

- **¿Qué es un bitstream?**

Es un fichero que se obtiene como resultado de la síntesis de nuestro diseño. Contiene toda la información necesaria para que la FPGA se reconfigure. Sería el equivalente a un fichero ejecutable en el mundo del software.

12. FAQs, preguntas más frecuentes

■ Ya tengo generado un bitstream, ¿Cómo lo pruebo en la JPS?

Conectar la tarjeta JPS a través del download cable de xilinx, quitar la eeprom de su zócalo, quitar el jumper JP1 y descargarlo usando el software de Xilinx. Una de las cosas que quedan por hacer es desarrollar una tecnología libre e independiente de Xilinx para poder descargar los bitstreams en la JPS.

Otra posibilidad es generar un fichero en formato HEX de intel (.mcs) y grabarlo en la eeprom. Situar la eeprom en el zócalo, poner a on el switch 1, colocar el jumper JP1 y pulsar el botón de programación.

A. Numeración del Zócalo PLCC84

Esta numeración puede resultar útil para leer con el polímetro determinados pines de la FPGA o bien para montarnos nuestra propia placa prototipo. La numeración se muestra en las figuras A.1 y A.2.

A. Numeración del Zócalo PLCC84

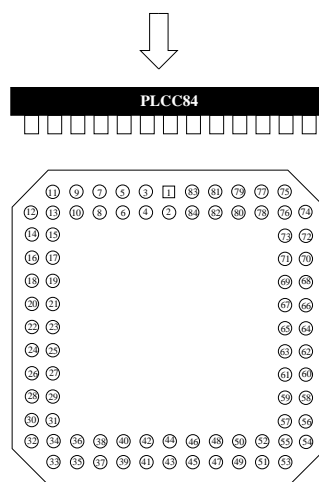


Figura A.1.: Vista superior del zócalo

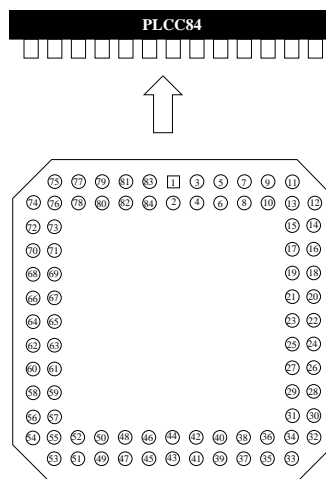


Figura A.2.: Vista inferior del zócalo

B. GNU Free Documentation License

Version 1.1, March 2000

Copyright © 2000 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The purpose of this License is to make a manual, textbook, or other written document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

B.1. Applicability and Definitions

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The ‘Document’, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as ‘you’.

A ‘Modified Version’ of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications

and/or translated into another language.

A ‘Secondary Section’ is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial,

B. GNU Free Documentation License

philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, L^AT_EX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

B.2. Verbatim Copying

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

B.3. Copying in Quantity

If you publish printed copies of the Document numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has ac-

cess to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

B.4. Modifications

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).
- State on the Title page the name of the publisher of the Modified Version, as the publisher.
- Preserve all the copyright notices of the Document.
- Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- Include an unaltered copy of this License.
- Preserve the section entitled 'History', and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled 'History' in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the 'History' section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- In any section entitled 'Acknowledgements' or 'Dedications', preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

B. GNU Free Documentation License

- Delete any section entitled “Endorsements”. Such a section may not be included in the Modified Version.
- Do not retitle any existing section as “Endorsements” or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties – for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

B.5. Combining Documents

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Inva-

riant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled “History” in the various original documents, forming one section entitled “History”; likewise combine any sections entitled “Acknowledgements”, and any sections entitled “Dedications”. You must delete all sections entitled “Endorsements.”

B.6. Collections of Documents

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

B.7. Aggregation With Independent Works

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no

compilation copyright is claimed for the compilation. Such a compilation is called an “aggregate”, and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document’s Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

B.8. Translation

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

B.9. Termination

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

B.10. Future Revisions of This License

The Free Software Foundation may publish new,

revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/> <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright © YEAR YOUR NAME.
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST. A copy of the license is included in the section entitled “GNU Free Documentation License”.

If you have no Invariant Sections, write “with no Invariant Sections” instead of saying which ones are invariant. If you have no Front-Cover Texts, write “no Front-Cover Texts” instead of “Front-Cover Texts being LIST”; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these

B. GNU Free Documentation License

examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Bibliografía

- [1] Empresa Xilinx. <http://www.xilinx.com>
- [2] Página personal de Juan González. <http://www.iearobotics.com/personal/juan/index.html>
- [3] Página personal de Pablo A. Haya. <http://www.ii.uam.es/~phaya/>
- [4] Página de la tarjeta JPS. <http://www.iearobotics.com/personal/juan/doctorado/jps-xpc84/jps-xpc84.html>
- [5] CADsoft, empresa distribuidora del programa de diseño electrónico EAGLE. <http://www.cadsoft.de>
- [6] Procesador de textos, Lyx. <http://www.lyx.org>
- [7] Programa de diseño gráfico vectorial XFIG. <http://www.xfig.org/>
- [8] ATMEL. <http://www.atmel.com>
- [9] Fabricante de Circuitos Impresos (PCB). <http://www.electrocir.arrakis.es/>