Memoria descriptiva

Se presenta a continuación la descripción del desarrollo del proyecto.

Tras una descripción genérica del sistema que nos ocupa, la organización del presente proyecto, se basa en la metodología seguida para la consecución del objetivo final.

Así, se aborda en primer lugar, el estudio independiente de cada uno de los módulos del DSP TMS320F2812 que van a ser utilizados.

Se realiza una descripción teórica de los mismos, en cuanto a funcionalidad, registros característicos (destacándose aquellos bits más importantes)... A continuación, se explica la adaptación del mismo a la aplicación que se le quiere dar, indicándose someramente la configuración a cargar y, finalmente, se pasa a la fase de pruebas.

Es en este apartado en el que se explican las particularidades del diseño realizado. Este está altamente condicionado por las características eléctricas de los dispositivos a interconectar y queda patente en los planos adjuntados al final del documento.

NOTA: Todas las figuras referentes al DSP estudiado han sido extraídas del manual del mismo de Texas Instruments. De igual modo, ocurre para diagramas de tiempo y otras figuras del resto de integrados usados.

1.Introducción

El DSP TMS320F2812 pertenece a la generación de los TMS320C28x de Texas Instruments. Se trata de un DSP altamente integrable y orientado a aplicaciones de control de alta precisión. Está pensado, pues, para aplicaciones de redes ópticas, electrónica de consumo, automatización industrial y control de motores. Es muy destacable el hecho de que pueda ser programado en C/C++, favoreciendo la reducción del tiempo de desarrollo.

Sus características fundamentales son:

- Ciclo de instrucción a 150 MHz de 6,67 ns.
- SARAM (single access RAM) de palabras de 16 bit: 18K.
- 128K de memoria Flash on chip a 3,3 V.
- Boot ROM
- OTP ROM (1K x 16)
- Interfaz con memoria externa.
- Generadores de eventos : EVA, EVB. Comparadores, temporizadores de propósito general.
- Temporizador watchdog
- ADC de 12 bits
- 3 Temporizadores de CPU de 32 bits.
- SPI, SCIA, SCIB
- CAN
- McBSP
- 56 pines de entrada/salida (compartidos)
- 3 interrupciones externas.

Su arquitectura funcional es la que puede apreciarse en la siguiente figura (figura 1):

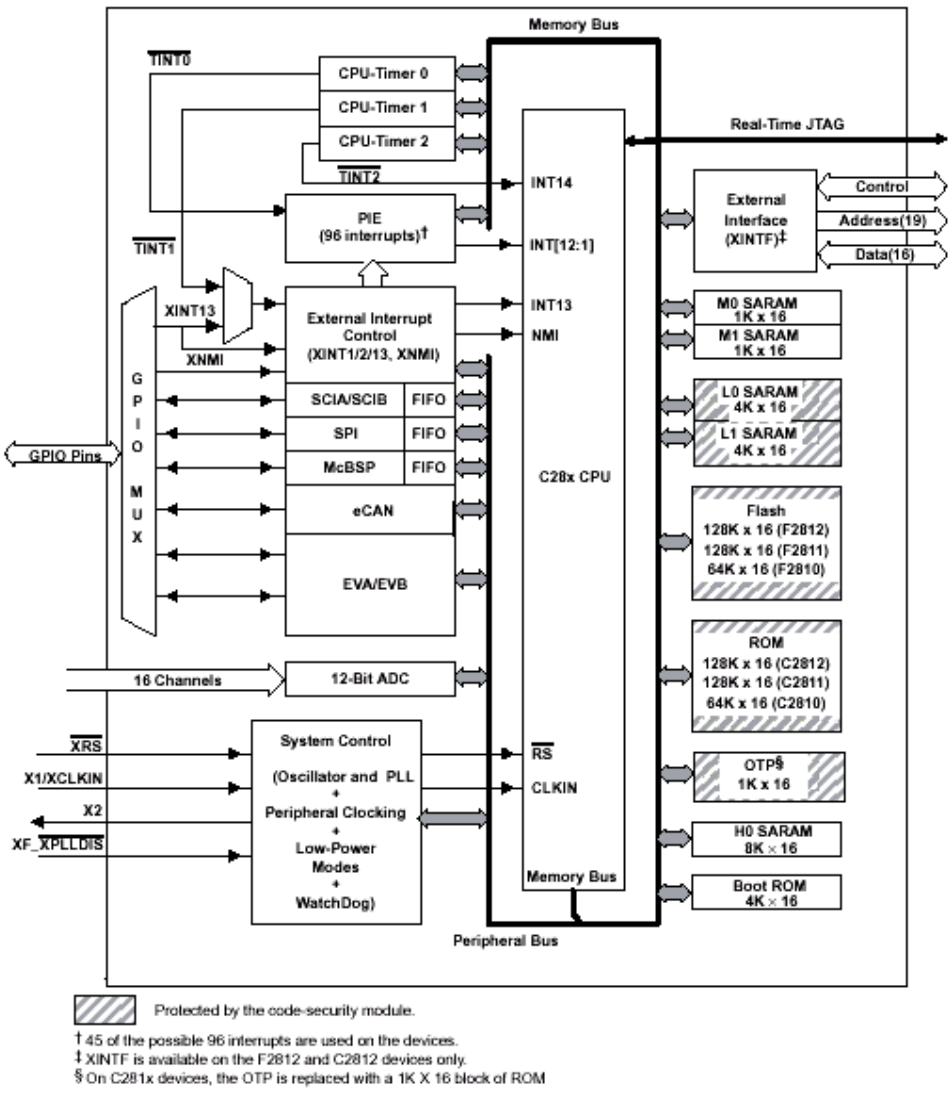
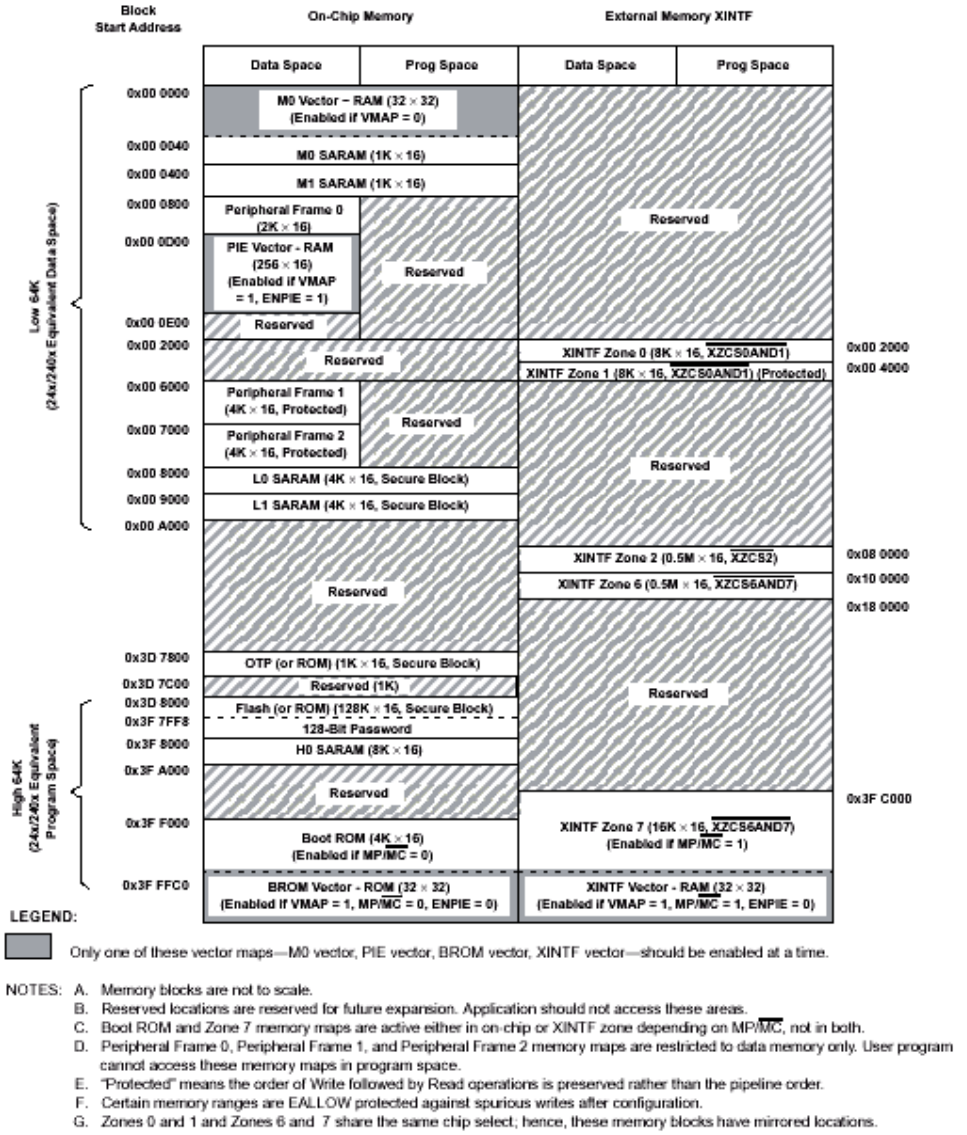


Figura 1: Diagrama de bloques funcional

Y en cuanto a su mapa de memoria:



NOTES: A. Memory blocks are not to scale.
B. Reserved locations are reserved for future expansion. Application should not access these areas.
C. Boot ROM and Zone 7 memory maps are active either in on-chip or XINTF zone depending on MP/$\overline{MC}$, not in both.
D. Peripheral Frame 0, Peripheral Frame 1, and Peripheral Frame 2 memory maps are restricted to data memory only. User program cannot access these memory maps in program space.
E. "Protected" means the order of Write followed by Read operations is preserved rather than the pipeline order.
F. Certain memory ranges are EALLOW protected against spurious writes after configuration.
G. Zones 0 and 1 and Zones 6 and 7 share the same chip select; hence, these memory blocks have mirrored locations.

Figura 2: Mapa de memoria

A continuación se presenta una descripción somera del dispositivo, tras la cual, se describirán más detalladamente, aquellos bloques del mismo que han sido utilizados en la realización del presente proyecto.

1.1. CPU de la familia C28x

La generación de DSP C28x es el miembro más novedoso de la plataforma TMS320C200. El C28x es compatible, en cuanto a código de programación se refiere, con otros dispositivos, tales como la familia C24x y se programa muy eficientemente en C/C++, lo que permite al usuario, no sólo desarrollar aplicaciones software de control en un lenguaje de alto nivel, sino

también el eficiente desarrollo de tareas matemáticas. La capacidad de 32 x 32 bit MAC del C28x y su capacidad de procesado de 64 bits, permite al C28X manejarse adecuadamente con problemas que requieren gran resolución numérica, y que, en otro caso, requerirían soluciones con un procesador de punto flotante. Debe destacarse también la velocidad de respuesta a las interrupciones; cuando una de ellas tiene lugar, se produce el salvado automático del contexto, es decir, de todos los registros críticos, lo que da lugar a un dispositivo capaz de dar servicio a muchos eventos asíncronos con latencia mínima.

1.2. Bus de Memoria

La arquitectura de los buses de memoria del C28x consta de un bus de lectura del programa, un bus de lectura de datos y un bus de escritura de datos.

El bus de lectura del programa tiene 22 líneas de direcciones y 32 líneas de datos. Los buses de lectura y escritura de datos consisten en 32 líneas de direcciones y 32 líneas de datos. La anchura del bus de datos de 32 bits permite operaciones de 32 bits en un único ciclo. Esta arquitectura de buses múltiple, llamada arquitectura Harvard, permite al C28x recoger una instrucción, leer el valor de un dato y escribir otro dato en un único ciclo.

1.3. Bus de periféricos

La estructura del bus de periféricos viene determinada por el hecho de hacer posible la migración de periféricos entre DSP´s de distintas familias. Así, el bus de periféricos multiplexa las16 líneas de direcciones y 16 o 32 líneas de datos, más sus señales de control asociadas, en un único bus.

El F2812 soporta dos versiones de este bus de periféricos: la denominada agrupación dos, que se corresponde con accesos de 16 bits; y la agrupación uno, que soporta accesos de 16 y 32 bits.

1.4. Interfaz externa

La interfaz externa del F2812 (XINTF) es una interfaz asíncrona consistente en 19 líneas de dirección, 16 líneas de datos, y 3 líneas de chip-select, las cuales son mapeadas en 5 zonas externas (zona 0, 1, 2, 6 y 7). Las zonas 0 y 1, al igual que las zonas 6 y 7, comparten una única señal de chip-

select. Cada una de las 5 zonas puede ser programada con un número de estados de espera diferente.

1.5. Flash

En cuanto a la memoria Flash de que dispone este DSP, esta se encuentra dividida en cuatro sectores de 8K x 16 y seis sectores de 16K x 16. El usuario puede programar, borrar y validar cada uno de los sectores independientemente del resto. En cualquier caso, no se puede usar un sector de la Flash o de la OTP ROM para ejecutar algoritmos que borran o programan otros sectores.

1.6. Boot ROM

El Boot ROM viene programado de fábrica con el software de bootloading. Las señales del modo boot permiten indicar al software bootloader qué modo boot usar en el arranque. El usuario puede elegir arrancar normalmente, o cargar otro software desde una conexión externa o que esté programado en la Flash interna.

1.7. Bloque de expansión del periférico de interrupciones

El bloque de expansión del periférico de interrupciones (PIE) sirve para multiplexar numerosas fuentes de interrupciones en un conjunto menor. Puede soportar hasta 96 interrupciones de periféricos, las cuales se agrupan en bloques de 8, yendo cada una de estas hacia una de las 12 líneas de interrupciones de la CPU.

Cada una de estas interrupciones es atendida por su propio vector, el cual se almacena en un bloque de RAM dedicado. En 8 ciclos de reloj, cuando se produce una interrupción, este vector es cargado y los registros típicos de la CPU son salvados. En consecuencia, la CPU responde rápidamente a los eventos de interrupción.

1.8. Oscilador, PLL y Watchdog

El F2812 puede recibir el reloj de un oscilador externo o del cristal incluido en el circuito oscilador on-chip. Existe también un PLL que soporta hasta 10 entradas para el escalado del reloj.

Consta además de un temporizador watchdog, que puede ser deshabilitado si es necesario.

1.9. Modos de bajo consumo

Los relojes de cada periférico pueden ser habilitados o deshabilitados individualmente, de forma que se reduce el consumo de energía cuando el periférico no está siendo utilizado.

Existen tres modos de bajo consumo:

- Idle: deja a la CPU en modo de bajo consumo. Los relojes de los periféricos se apagan según selección, y sólo aquellos que necesitan seguir funcionando continúan recibiendo el reloj. El procesador despierta de este estado cuando se recibe una interrupción habilitada procedente de uno de los periféricos en funcionamiento.
- Standby: apaga el reloj de la CPU y los periféricos. Una interrupción externa es la que despierta al procesador y los periféricos. La ejecución comienza en el siguiente ciclo válido después de la detección de la interrupción.
- Halt: apaga el oscilador. Deja al dispositivo en el modo de mínimo consumo. Únicamente un reset o una interrupción externa no enmascarable despiertan al dispositivo.

1.10. Agrupación de periféricos 0, 1 y 2

El F2812 separa los periféricos en 3 secciones (PF: peripheral frames):

- PF0: XINTF, PIE, Flash, Timers, CSM
- PF1: eCan
- PF2: SYS, GPIO, EV, McBSP, SCI,SPI, ADC

1.11. Entradas/salidas de propósito general

La mayoría de las señales de los periféricos están multiplexadas con señales de entrada salida de propósito general. Esto permite al usuario utilizar un pin como GPIO si la señal o función del periférico está deshabilitada.

Tras un reset, todos los pines son configurados como entradas de propósito general y es el usuario el que programa la funcionalidad de ese pin, indicando si debe estar o no asociado a otro periférico.

1.12. Temporizadores de la CPU

Los temporizadores de la CPU 0, 1 y 2 son temporizadores de 32 bits con periodos precargables y un reloj de 16 bits para el escalado. Constan de un registro contador decreciente de 32 bits que genera una interrupción cada vez que el contador llega a cero. Cuando esto ocurre, el contador es automáticamente recargado al valor del período. El contador se decrementa a la velocidad del reloj dividida por el valor determinado por el preescalado.

1.13. Periféricos para aplicaciones de control:

El F2812 tiene los dos siguientes periféricos para el control y la comunicación:

- EV: el módulo de generación de eventos incluye temporizadores de propósito general, comparadores/unidades PWM, entradas de captura (CAP) y circuitos de codificación de la cuadratura del pulso (QEP). Estos dos generadores de eventos permiten el control de dos motores de 3-fases o bien, cuatro motores de 2-fases.
- ADC: es un bloque convertidor de 12 bits con 16 canales. Contiene dos unidades de muestreo y retención para el muestreo simultáneo.

1.14. Periféricos del puerto serie

Se tienen los siguientes periféricos para comunicación serie:

- eCAN: se trata de la versión extendida del CAN (Controller Area Network)
- McBSP: multichannel buffered serial port. Se utiliza para conectar a las líneas E1/T1 codecs de calidad telefónica para aplicaciones módem o para dispositivos DAC de audio de alta calidad.
- SPI: puerto de entrada/salida síncrono a alta velocidad que permite que un flujo serie de bits, de longitud programada, sea transferido también, a velocidad programada. Normalmente, se usa para comunicaciones entre el controlador del DSP y periféricos externos u otro procesador.

- SCI: La interfaz de comunicación serie es un puerto serie asíncrono de dos cables, normalmente conocido como UART. Este puerto presenta una FIFO de 16 niveles de recepción y transmisión.

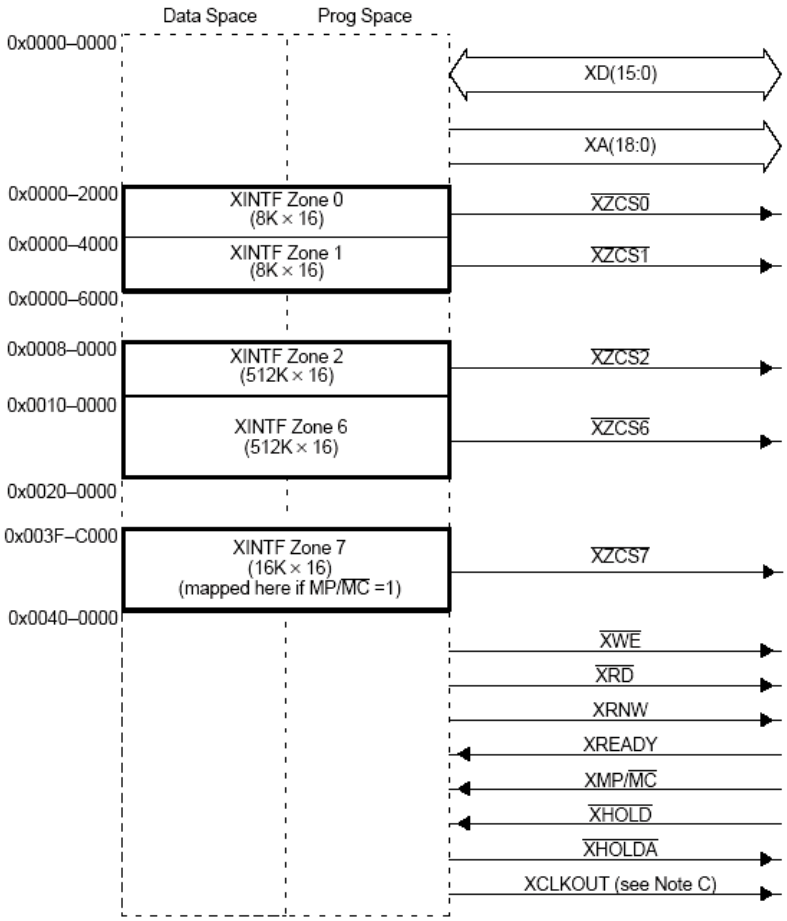
2. Descripción de bloques utilizados

2.1. Interfaz externa (XINTF)

2.1.1. Descripción funcional

La interfaz externa consiste en un bus asíncrono no multiplexado.

Esta se encuentra mapeada en cinco zonas fijas, según se ve en la figura 3.



NOTES: A. The mapping of XINTF Zone 7 is dependent on the XMP/MC device input signal and the MP/MC mode bit (bit 8 of XINTCNF2 register). Zones 0, 1, 2, and 6 are always enabled.
B. Each zone can be programmed with different wait states, setup and hold timings, and is supported by zone chip selects (XZCS0AND1, XZCS2, XZCS6AND7), which toggle when an access to a particular zone is performed. These features enable glueless connection to many external memories and peripherals.
C. XCLKOUT is also pinned out on the devices without the rest of the XINTF.

Figura 3: Diagrama de bloques de XINTF

Cada una de estas zonas del XINTF tiene su propia señal de selección de chip (CS chip-select), la cual es seleccionada cuando se hace un acceso a una zona en concreto del mapa de memoria descrito anteriormente. Para

algunos dispositivos, como el F2812, se encuentran unidas (ANDed) dos zonas de memoria con una sola señal CS.

Cada una de las cinco zonas puede ser programada con un número específico de estados de espera, señal de set-up y tiempo de mantenimiento. El número de estados de espera y la temporización del set-up y mantenimiento se puede especificar de manera independiente para accesos en lectura o escritura. Además, cada zona puede ser programada para aumentar los estados de espera usando la señal externa XREADY. Los estados de espera programables, la temporización y la señal de chip-select se habilitan de tal manera que no se solapen con otros dispositivos o memorias externas.

Configurando adecuadamente los registros XTIMINGx es posible especificar los tiempos set-up/hold y los estados de espera, de escritura y lectura, para cada una de las zonas del XINTF independientemente. Los tiempos de acceso están basados en un reloj interno llamado XTIMCLK, que puede ser configurado a la misma frecuencia que el reloj de la CPU (SYSCLKOUT) o a la mitad del mismo. Dicha frecuencia se aplica a todas las zonas del XINTF. Los ciclos de reloj del bus XINTF comienzan con el flanco de subida del XCLKOUT y todos los tiempos y eventos siguientes son generados respecto al flanco de subida del XTIMCLK.

Los valores válidos de los tiempos setup (lead) , active y hold (trail) quedan reflejados en la siguiente tabla:

	Acceso escritura	Acceso lectura
Lead	0-3 XTIMCLK ciclos	0-3 XTIMCLK ciclos
Active	1-7 XTIMCLK ciclos	1-7 XTIMCLK ciclos
Trail	0-3 XTIMCLK ciclos	0-3 XTIMCLK ciclos

Tabla 1:Valores válidos de lead/active/trail

En caso de que se use la señal externa XREADY, el valor mínimo para los accesos en lectura y escritura debe ser uno.

Para extender aún más el tiempo de acceso, cada zona XINTF posee un modo referenciado como X2TIMING que duplica los valores especificados de lead, active y trail, para los accesos en lectura y en escritura. Así, los tiempos de acceso mínimos (sin tener en cuenta XREADY), con XTIMCLK = SYSCLKOUT y X2TIMING = 0 son Lead = 0, Active = 1, Trail = 0.

Con lo cual, tenemos un solo estado de espera o lo que es equivalente una ejecución a 75 MHz si el reloj SYSCLKOUT va a 150 MHz.

Los tiempos de acceso máximos (sin tener en cuenta XREADY), con XTIMCLK =1/2 SYSCLKOUT y X2TIMING = 1 son Lead = 3, Active = 7, Trail = 3; lo que resulta en un tiempo máximo de establecimiento y mantenimiento de 12 ciclos de SYSCLKOUT, y un tiempo máximo de espera de 28 ciclos de SYSCLKOUT.

2.1.2. Configuración de los registros del XINTF

Los requisitos del sistema son los que determinan los valores que toman los parámetros configurables del XINTF. La configuración exacta a implementar viene dada por la frecuencia de operación del DSP, las características de conmutación del XINTF, y los requerimientos de temporización de los dispositivos externos con los cuales se crea la interfaz.

Dos son los relojes que se pueden usar en este módulo. La relación existente entre ambos queda patente en la siguiente figura.

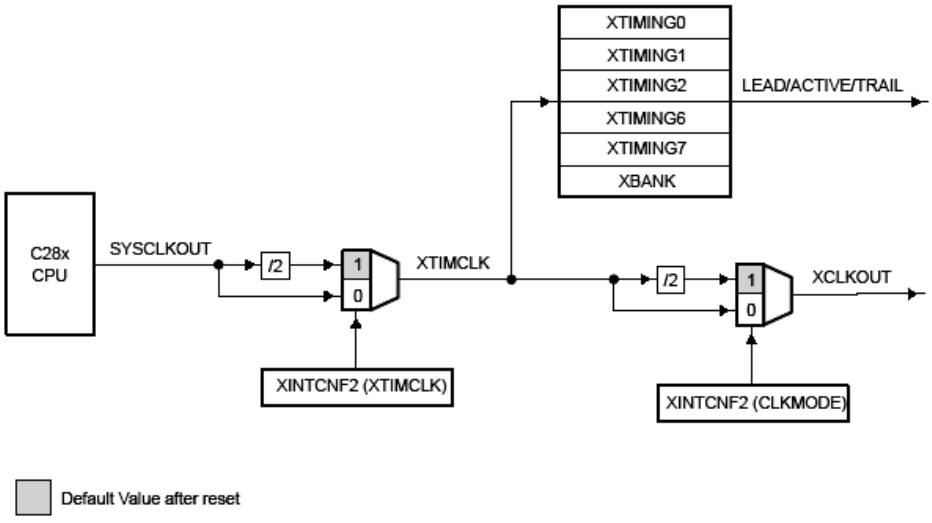


Figura 4: Relación entre XTIMCLK y SYSCLKOUT

Como se observa en la figura, todos los accesos a las distintas zonas del XINTF se basan en el reloj XTIMCLK (reloj interno del XINTF). Este puede ser

configurado como 1:1 o como 1:1/2 del SYSCLKOUT escribiendo en el bit XTIMCLK dentro del registro XINTFCNF2. Por defecto XTIMCLK es la mitad de SYSCLKOUT.

Todos los accesos al XINTF comienzan con el flanco de subida del XCLKOUT. Este reloj de salida se puede configurar a la misma frecuencia que el XTIMCLK o la mitad escribiendo en el bit CLKMDE del registro XINTFCNF2. Por defecto XCLKOUT es la mitad del XTIMCLK, es decir, un cuarto del reloj de la CPU, SYSCLKOUT.

Así pues, las señales de temporización del XINTF pueden ser sintonizadas para que coincidan con los requisitos del dispositivo externo en cuanto a tiempo de establecimiento, mantenimiento. Además de los períodos lead, active y trail, cuyo significado puede verse en los diagramas de tiempo adjuntos, existe una señal, XREADY, que permite aumentar aún más el período active del acceso correspondiente.

La señal XREADY es compartida por todas las zonas del XINTF, si bien cada una de estas puede habilitar su muestreo o ignorarla. El muestreo puede ser síncrono, si se muestrea esta señal un ciclo de XTIMCLK antes del total de ciclos lead+active especificados; o asíncrono, si es muestreada tres ciclos de XTIMCLK antes. Estas posibilidades son configuradas en los registros XTIMING, mediante los bits READYMODE y USEREADY.

Las relaciones entre los parámetros que pueden ser configurados en los registros XTIMING y la duración de los pulsos en términos de ciclos de reloj de XTIMCLK, tc(XTIM) es:

Descripción	Duración (ns)	
	X2TIMING=0	X2TIMING=1
LR Lead period, read access	XRDLEAD x tc(xtim)	(XRDLEADx2) x tc(xtim)
AR Active period, read access	(XRDACTIVE+WS+1) x tc(xtim)	(XRDACTIVEx2+WS+1) x tc(xtim)
TR Trail period, read access	(XRDACTIVE+WS+1) x tc(xtim)	(XRDACTIVEx2+WS+1) x tc(xtim)
LW Lead period, write access	XWRLEAD x tc(xtim)	(XWRLEADx2) x tc(xtim)

AW Active period, write access	(XWRACTIVE+WS+1) x tc(xtim)	(XWRACTIVEx2+WS+1) x tc(xtim)
TW Trail period, write access	XWRTRAIL x tc(xtim)	(XWRTRAILx2) x tc(xtim)

Tabla 2: Duración de pulsos en función de XTIMCLK

Para aclarar lo que se ha expuesto hasta este punto, se presentan los diagramas de tiempo de un ciclo de lectura y otro de escritura genéricos.

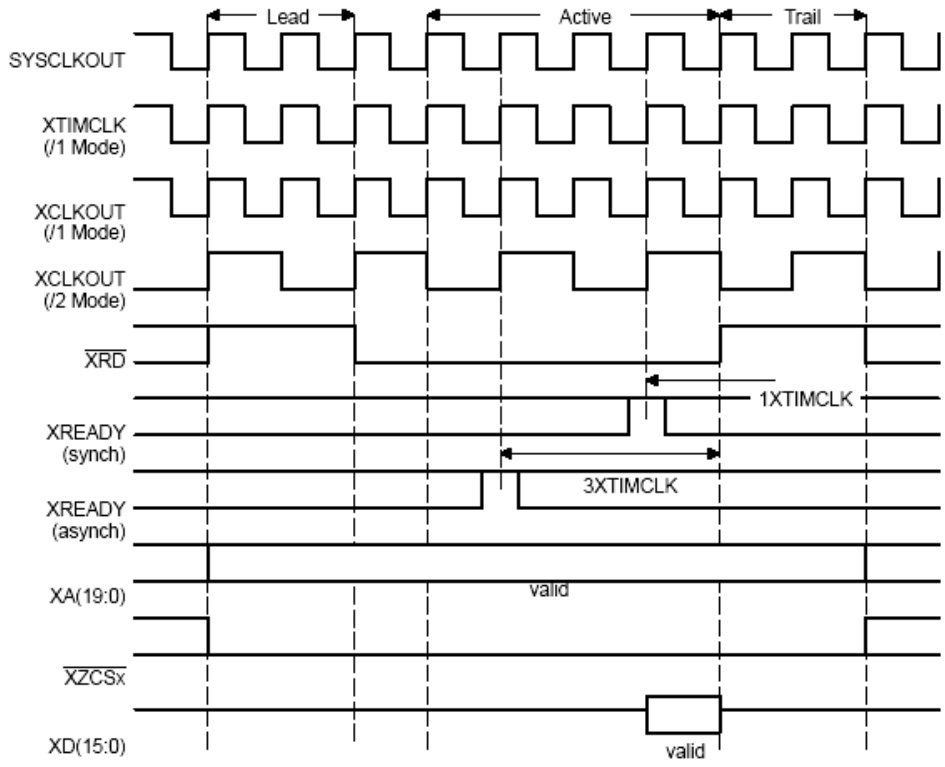


Figura 5: Ciclo de lectura (XTIMCLK=SYSCLKOUT)

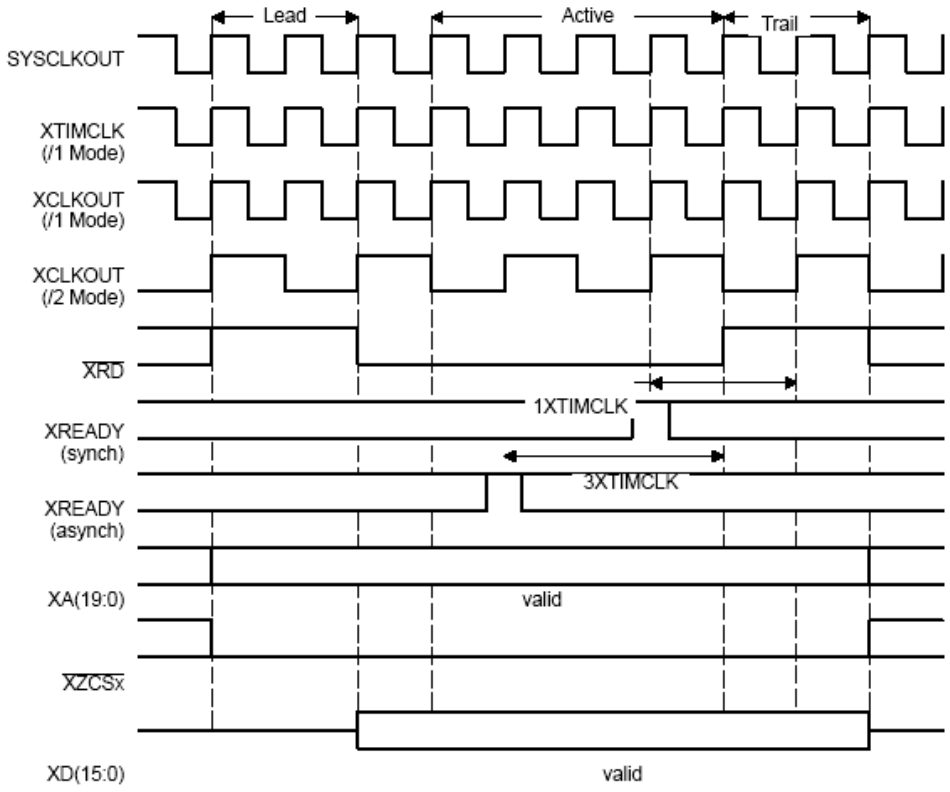


Figura 6: Ciclo de escritura genérico

La tabla 3 Muestra los registros de configuración del XINTF, cuya modificación afecta a la temporización de los accesos de este módulo.

Nombre	Tamaño (x16)	Descripción
XTIMING0	2	Temporización zona 0
XTIMING1	2	Temporización zona 1
XTIMING2	2	Temporización zona 2
XTIMING6	2	Temporización zona 6
XTIMING7	2	Temporización zona 7
XINTCNF2	2	Configuración XINTF
XBANK	1	Banco control XINTF
XREVISION	1	Revisión XINTF

Tabla 3: Registros del XINTF

2.2. Puerto serie síncrono de periféricos (SPI)

Este módulo es un puerto de entrada/salida síncrono a alta velocidad.

Normalmente se usa para comunicaciones entre el controlador del DSP y periféricos externos y otro controlador. Son aplicaciones típicas entradas/salidas externas, expansión de periféricos por medio de registros de desplazamientos, controladores de display y convertidores analógicos a digital.

Múltiples operaciones pueden ser llevadas a cabo con el SPI en modo maestro o esclavo. Este puerto tiene, además, una FIFO de recepción y transmisión para reducir el exceso de carga en la CPU.

2.2.1.Descripción somera del módulo

Entre las características del SPI destacamos:

- 4 pines externos: SPISIMO, SPISOMI, SPISTE y SPICLK.
- Dos modos de operación: maestro y esclavo.
- Velocidad en baudios: 25 posibilidades programables.
- Longitud de la palabra de datos: de 1 a 16 bits.
- 4 esquemas de temporización

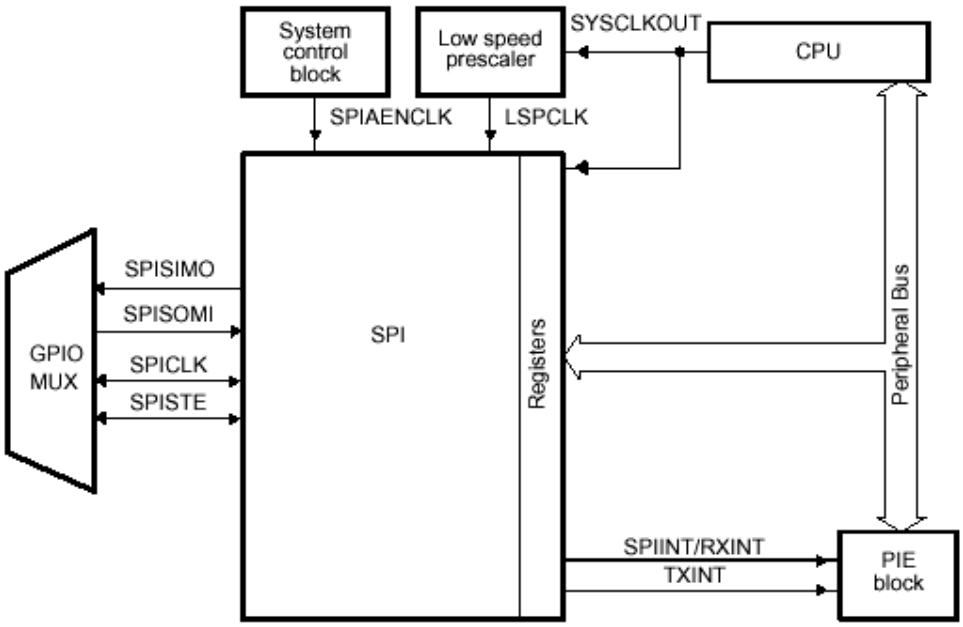


Figura 7: Interfaz del SPI con la CPU

2.2.2. Operación del SPI

Existen dos modos de funcionamiento, maestro y esclavo. La selección de uno de estos modos se lleva a cabo mediante la activación o no del bit MAESTRO/ESCLAVO en el registro SPICTL.

El maestro siempre inicia la transferencia de datos, en el momento que desee, enviando la señal SPICLK. La aplicación software es la que determina cómo el maestro se da cuenta de que el esclavo esté listo.

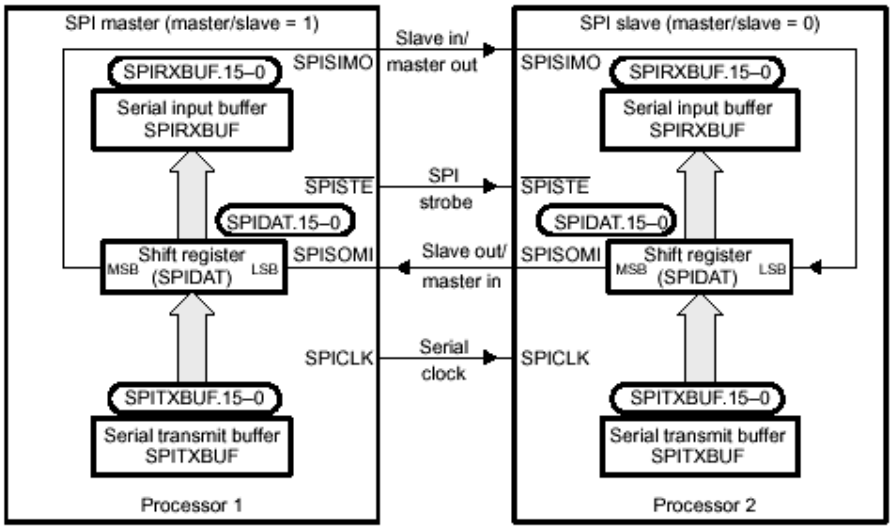


Figura 8: Conexión M/S del SPI

Se explica el modo maestro más profundamente, puesto que este será el que se utilizará en la aplicación posterior.

En el modo maestro, el SPI proporciona el reloj serie a través del pin SPICLK para todo el canal de comunicaciones. Los datos salen por el pin SPISIMO (SPI slave input/master output) y se recogen en el pin SPISOMI (SPI slave output/master input)

El valor cargado en el registro SPIBRR determina la velocidad de transferencia.

Con la escritura de datos en el SPIDAT o SPITXBUF, se inicia la transmisión. El dato debe estar justificado a la izquierda y se transmite el MSB en primer lugar. Una vez el número especificado de bits ha sido desplazado en el SPIDAT, se producen los siguientes eventos:

- el contenido del SPIDAT se vuelca en el SPIRXBUF.

- Se activa la bandera de interrupción del SPI (SPISTS.6)
- Si hay un dato válido en SPITXBUF, se transmite al SPIDAT o, si no lo hay, el SPICLK se para.
- En caso de que el bit de habilitación de interrupción esté a 1 (SPICTL.0), esta es generada.

2.2.4 Esquemas de temporización y régimen de transmisión

El SPI soporta 125 velocidades diferentes y cuatro esquemas de reloj distintos.

En el caso de que el SPI funcione en modo esclavo, a través del pin SPICLK, se recibe el reloj de una fuente externa. Si por el contrario, trabaja en modo maestro, el pin SPICLK es una salida.

El régimen en baudios, viene dado por el valor almacenado en el registro SPIBRR, y se calcula según la siguiente fórmula:

$$SpiBaudRate = \frac{LSPCLK}{SPIBRR + 1}$$

Donde LSPCLK, representa la frecuencia del reloj de baja velocidad de periféricos del dispositivo.

Los cuatro posibles esquemas del reloj son el resultado de la configuración de los bits de CLOCK POLARITY (SPICCR.6) y el CLOCK POLARITY (SPICTL.6). El primero de estos bits determina cuál es el flanco activo (de subida o bajada), mientras que el segundo selecciona un retraso de medio ciclo para el reloj. Así, las cuatro posibilidades son:

- Flanco de bajada sin retraso: el SPI transmite en el flanco de bajada del SPICLK y recibe datos en el flanco de subida del SPICLK.
- Flanco de bajada con retraso: el SPI transmite el dato medio flanco antes del flanco de bajada del SPICLK y recibe el dato en el flanco de bajada del mismo.
- Flanco de subida sin retraso: el SPI transmite el dato en el flanco de subida del SPICLK y recibe en el flanco de bajada.
- Flanco de subida con retraso: el SPI transmite el dato medio ciclo antes del flanco de subida del SPICLK y recibe en el flanco de subida.

La figura clarifica la explicación:

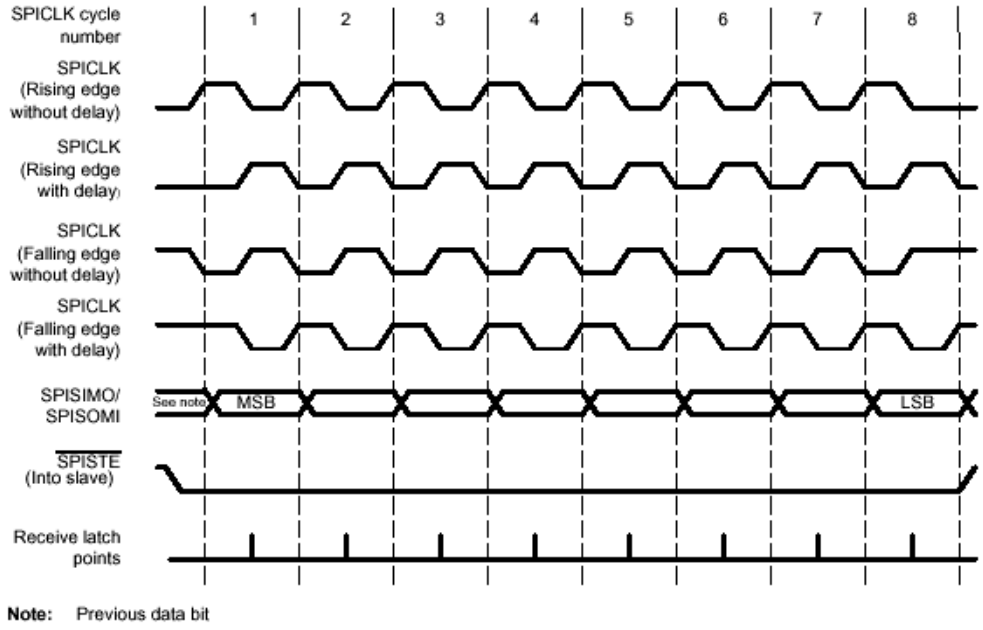


Figura 9: Opciones de la señal SPICLK

Valores de los bits para conseguir estas configuraciones:

Esquema SPICLK	CLOCK POLARITY (SPICCR.6)	CLOCK PHASE (SPICTL.3)
Flanco de subida sin retraso	0	0
Flanco de subida con retraso	0	1
Flanco de bajada sin retraso	1	0
Flanco de bajada con retraso	1	1

Tabla 4: Selección del esquema de SPICLK

2.2.5. Breve descripción de SPI FIFO

El funcionamiento en modo FIFO es habilitado al escribir un '1' en el bit SPIFFEN del registro SPIFFTX.

El esquema de generación de señales en este modo es el mostrado a continuación.

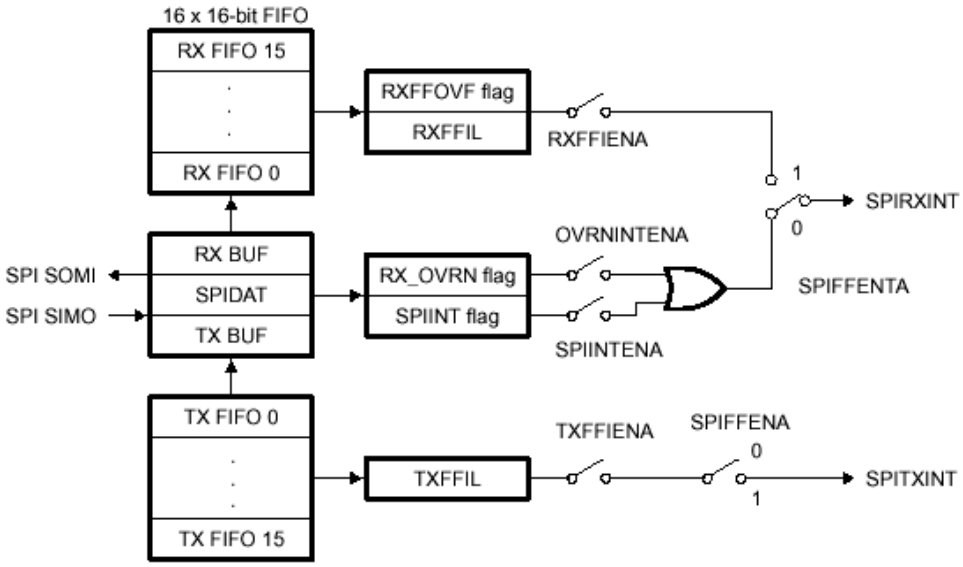


Figura 10: Banderas de interrupción de SPIFIFO

En este modo de funcionamiento existen dos interrupciones, una para la FIFO de transmisión (SPITXINT), y otra para la recepción (SPIRXINT/SPIINT). Esta última es común para la FIFO de recepción, recepción de error y la condición de desbordamiento.

Los bufferes para esta FIFO se implementan mediante dos FIFOs de 16x16.

Los bits de estado de la FIFO determinan el número de palabras disponibles en las FIFOs en un determinado momento; estos bits son TXFFST y RXFFST.

Por último, se presenta a continuación, una tabla con los registros propios de este módulo.

Nombre	Tamaño (x16)	Descripción
SPICCR	1	Registro de control de la configuración del SPI
SPICTL	1	Registro de control de operación del SPI
SPIST	1	Registro de estado del SPI
SPIBRR	1	Registro del régimen en baudios del SPI
SPIEMU	1	Registro de emulación del

		buffer del SPI
SPIRXBUF	1	Registro del buffer de entrada serie del SPI
SPITRBUF	1	Registro del buffer de salida serie del SPI
SPIDAT	1	Registro de datos serie
SPIFFTX	1	Registro de la FIFO de transmisión del SPI
SPIFFRX	1	Registro de la FIFO de recepción del SPI
SPIFFCT	1	Registro de control de la FIFO del SPI
SPIPRI	1	Registro de control de prioridad del SPI

Tabla 5: Registros del SPI

2.3. Módulo de convertidores analógico/digital (ADC)

El módulo ADC del F2812 es un convertidor analógico digital pipeline de 12 bits. La circuitería analógica, llamada núcleo o core, incluye multiplexores analógicos de entrada, circuitos de muestreo y retención, el núcleo de conversión, reguladores de tensión y otros circuitos de apoyo. En cuanto a los circuitos digitales, estos incluyen un secuenciador de conversión programable, registros de resultado, circuitos para interfaz analógica, interfaz para el bus de periféricos del dispositivo y para otros módulos.

2.3.1. Breve descripción del módulo

Las características de este módulo incluyen:

- Entrada analógica de 0 a 3 V.
- Régimen de conversión rápida: 80 ns para un CLK de ADC de 25 MHz, 12,5 MSPS.
- 16 canales de entradas multiplexadas.
- Capacidad de autosecuenciación proporcionando hasta 16 autoconversiones en una única sesión.
- El secuenciador puede funcionar como dos secuenciadotes independientes de 8 estados, o como un único secuenciador; en el primer caso, puede operar también en modo simultáneo o secuencial.
- 16 registros de resultado de la conversión.

$$ValorDigital = 4095x\frac{VoltajeEntrada - ADCLO}{3}$$

- Distintas posibilidades para disparar el inicio de conversión; vía software, a partir del EVA, o el EVB.
- Control flexible de interrupciones.
- La temporización interna del ADC viene dada por HSPCLK (high speed clock)

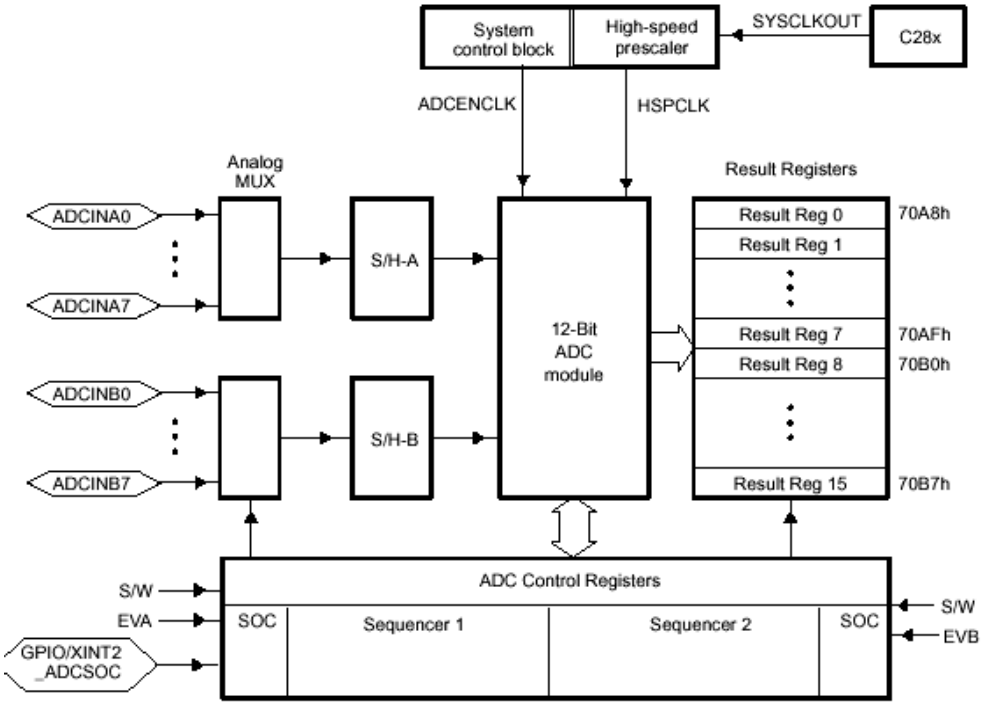


Figura 11: Diagrama de bloques funcional

2.3.2. Registros asociados al ADC

Los registros correspondientes a este módulo son los presentados a continuación.

Nombre	Descripción
ADCCTRL1	Registro de control 1
ADCCTRL2	Registro de control 2
ADCMAXCONV	Registro de máximo nº de canales a convertir
ADCCHSELSEQ1..4	Registros de control 1 a 4 para la secuenciación de selección de canal
ADCASEQSR	Registro de estado del autosecuenciador

ADCRESULT0..15	Registros del 0 al 15 del buffer de resultado de la conversión
ADCCTRL3	Registro de control 3
ADCST	Registro de estado

Tabla 6: Registros asociados al ADC

3. Conexión de un display LCD al bus de datos y direcciones

Se detalla en este apartado cómo ha sido el montaje realizado, las causas que nos han llevado a él y cuál ha sido el resultado final.

3.1. Características del display

El dispositivo externo al cual vamos a conectar el bus de datos y direcciones es un display de 2x16 de la marca Xiamen Ocular, modelo GDM 1602A, con un controlador KS0066U o equivalente.

En su datasheet, puede observarse cuáles son sus características eléctricas. Son destacables las direcciones de acceso al dispositivo.

Display position	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*
DDRAM address	*00*	*01*	*02*	---	---	---	---	---	---	---						*0FH*
DDRAM address	*40*	*41*	*42*	---	---	---	---	---	---	---						*4FH*

Figura 12: Direcciones del display accesibles

Los pines del mismo son:

PIN NO	Symbol	Function
1	VSS	GND
2	VDD	+5V
3	V0	Contrast adjustment
4	RS	H/L Register select signal
5	R/W	H/L Read/Write signal
6	E	H/L Enable signal
7	DB0	H/L Data bus line
8	DB1	H/L Data bus line
9	DB2	H/L Data bus line
10	DB3	H/L Data bus line
11	DB4	H/L Data bus line
12	DB5	H/L Data bus line
13	DB6	H/L Data bus line
14	DB7	H/L Data bus line
15	A	+4.2V for BKL
16	K	Power supply for BKL(0V)

Tabla 7: Pines de conexión

Para el propósito marcado, es necesaria la información de la temporización que requiere el dispositivo, por lo que hay que remitirse al manual del controlador.

A partir del mismo, obtenemos el diagrama de flujo de cómo debe ser la inicialización, cuyos pasos son los que se han seguido en el programa codificado.

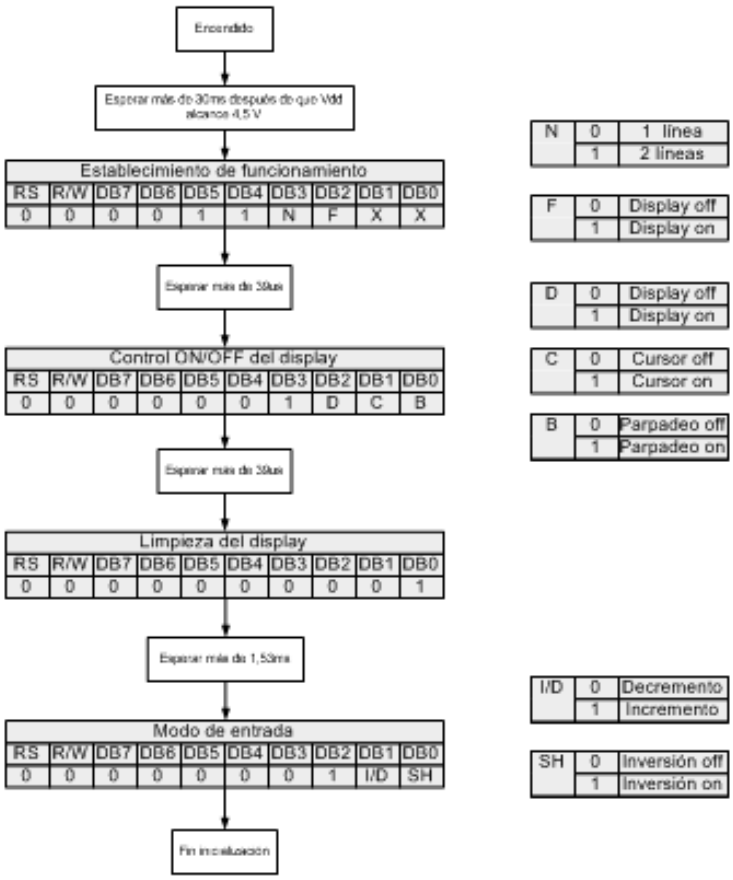


Figura 13: Diagrama de inicialización del LCD

Resultan de especial relevancia los tiempos de ejecución de las instrucciones del display, los cuales deben respetarse siempre. Estos, junto con los diagramas de tiempo, son los que han determinado el resultado final.

Instruction	Instruction Code										Description	Execution time (fosc= 270 kHz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Write "20H" to DDRAM and set DDRAM address to "00H" from AC	1.53 ms
Return Home	0	0	0	0	0	0	0	0	1	-	Set DDRAM address to "00H" from AC and return cursor to its original position if shifted. The contents of DDRAM are not changed.	1.53 ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	SH	Assign cursor moving direction and enable the shift of entire display.	39 µs
Display ON/ OFF Control	0	0	0	0	0	0	1	D	C	B	Set display(D), cursor(C), and blinking of cursor(B) on/off control bit.	39 µs
Cursor or Display Shift	0	0	0	0	0	1	S/C	R/L	-	-	Set cursor moving and display shift control bit, and the direction, without changing of DDRAM data.	39 µs
Function Set	0	0	0	0	1	DL	N	F	-	-	Set interface data length (DL: 8-bit/4-bit), numbers of display line (N: 2-line/1-line) and, display font type (F:5×11dots/5×8 dots)	39 µs
Set CGRAM Address	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	Set CGRAM address in address counter.	39 µs
Set DDRAM Address	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Set DDRAM address in address counter.	39 µs
Read Busy Flag and Address	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Whether during internal operation or not can be known by reading BF. The contents of address counter can also be read.	0 µs
Write Data to RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Write data into internal RAM (DDRAM/CGRAM).	43 µs
Read Data from RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	Read data from internal RAM (DDRAM/CGRAM).	43 µs

* "-": dont care

Tabla 8: Tabla de instrucciones

De la observación de estos tiempos, se deduce que el display es bastante lento, por lo que, inicialmente, se decidió configurar el XINTF con los tiempos de accesos más lentos posibles, es decir, sus valores por defecto.

3.2. Configuración del DSP

La zona del XINTF elegida para conectar el dispositivo fue la zona 0, que abarca desde la dirección 0x00 2000 hasta la 0x00 4000. Sólo se necesitan

definir dos direcciones, una para instrucciones de control hacia el LCD (se usará un puntero a la dirección 0x2000), y otra para los datos (puntero a la dirección 0x2001).

En principio, llegados a este punto, bastaría con realizar la correcta conexión hardware entre el display y la placa de desarrollo del DSP. Se asignarían las líneas del bus de datos a los datos del display; XA0 (línea más baja del bus de direcciones) al RS, pudiendo distinguir de esta forma cuándo se están mandando datos y cuándo instrucciones; la señal de Write enable se pondría a GND, para que la escritura esté siempre habilitada; y, por último, para el enable del display se haría la función NOR de XZCS0AND1n y XWEn.

Figura 14: Obtención de la señal de selección del LCD

No obstante, en estas circunstancias no era posible el funcionamiento del LCD, como más tarde se comprobó con la realización de medidas experimentales en el laboratorio.

Al remitirse a la tabla de características AC del dispositivo, se observa la necesidad de un pulso enable de duración 230 ns, valor en el que, a pesar de la configuración al máximo de todos los períodos de los tiempos de acceso del XINTF, nos quedábamos al límite, lo que suponía que, el display podía recibir ciertos datos correctos y otros no.

Mode	Characteristic	Symbol	Min.	Typ.	Max.	Unit
Write Mode (Refer to Fig-6)	E Cycle Time	tc	500	-	-	ns
	E Rise / Fall Time	t_R,t_F	-	-	20	
	E Pulse Width (High, Low)	tw	230	-	-	
	R/W and RS Setup Time	tsu1	40	-	-	
	R/W and RS Hold Time	t_{H1}	10	-	-	
	Data Setup Time	tsu2	80	-	-	
	Data Hold Time	t_{H2}	10	-	-	
Read Mode (Refer to Fig-7)	E Cycle Time	tc	500	-	-	ns
	E Rise / Fall Time	t_R,t_F	-	-	20	
	E Pulse Width (High, Low)	tw	230	-	-	
	R/W and RS Setup Time	tsu	40	-	-	
	R/W and RS Hold Time	t_H	10	-	-	
	Data Output Delay Time	t_D	-	-	120	
	Data Hold Time	t_{DH}	5	-	-	

Tabla 9: Características AC para Vdd =4.5 V ~ 5.5 V

Ante este problema, la solución planteada consistió en usar estados de espera adicionales mediante la señal XREADY.

La señal XREADY permite indicar que el periférico está listo para completar el acceso cuando se asegura un '1' válido. Para cada zona del XINTF, esta señal puede ser configurada para ser ignorada o para ser muestreada síncrona o asíncronamente.

La configuración de los bits y registros correspondientes puede observarse en el inicio del código del programa.

```
//Configuración del XCLKOUT
XintfRegs.XINTCNF2.bit.XTIMCLK=1;
XintfRegs.XINTCNF2.bit.CLKMODE=0;
XintfRegs.XTIMING0.all=0xFFFFFFFF;
```

Son fundamentales en esta tarea los registros XTIMING0 y XINTCNF2.

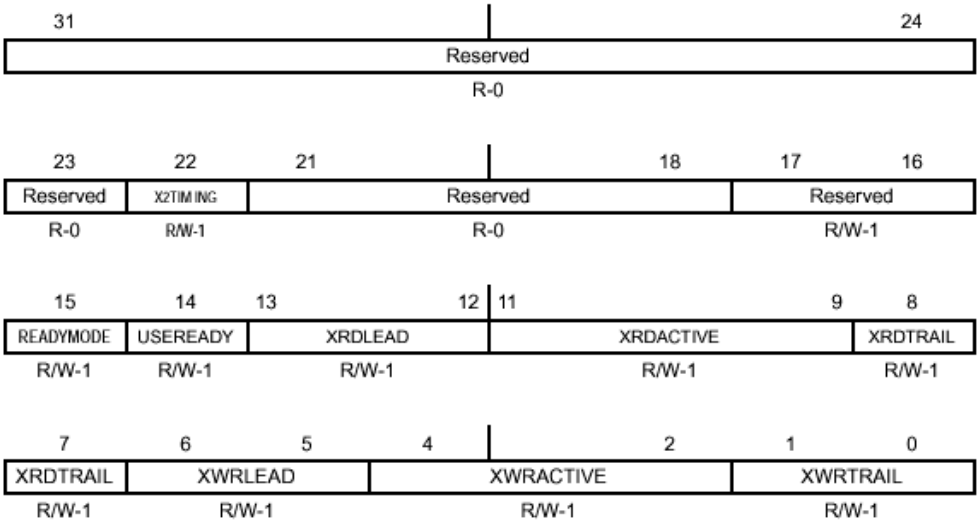


Figura 15: Registro XTIMING0

- Bit 22: Indica el factor de escala de los períodos lead, active y trail para accesos en lectura y escritura (‘1’, multiplica x2).
- Bit 15: Configura el muestreo de la señal XREADY como síncrono (‘0’) o asíncrono (‘1’)
- Bit 14: Determina si los accesos a la zona muestrean la señal XREADY (‘1’) o no (‘0’).
- Bits 13..12: Duración del período lead para lectura, en ciclos de XTIMCLK (‘00’, no válido; ‘01’, 1 ciclo; ‘10’, 2 ciclos; ‘11’, 3 ciclos).
- Bits 11..9: Duración del período active para lectura en ciclos de XTIMCLK (‘000’, 0 ciclos; ‘001’, 1 ciclo; ‘010’, 2 ciclos…)
- Bits 8..7: Duración del periodo trail para lectura en ciclos de XTIMCLK (‘00’, no válido; ‘01’, 1 ciclo; ‘10’, 2 ciclos; ‘11’, 3 ciclos).
- Bits 6..5: Duración del período lead para escritura, en ciclos de XTIMCLK (‘00’, no válido; ‘01’, 1 ciclo; ‘10’, 2 ciclos; ‘11’, 3 ciclos).
- Bits 4..2: Duración del período active para escritura en ciclos de XTIMCLK (‘000’, 0 ciclos; ‘001’, 1 ciclo; ‘010’, 2 ciclos…)
- Bits 1..0: Duración del periodo trail para escritura en ciclos de XTIMCLK (‘00’, no válido; ‘01’, 1 ciclo; ‘10’, 2 ciclos; ‘11’, 3 ciclos).

En cuanto al registro XINTCNF2, de este sólo vamos a forzar el bit CLKMODE a ‘0’, de forma que, cómo ya se vio en la figura 4, la frecuencia del reloj XCLKOUT coincida con la de XTIMCLK.

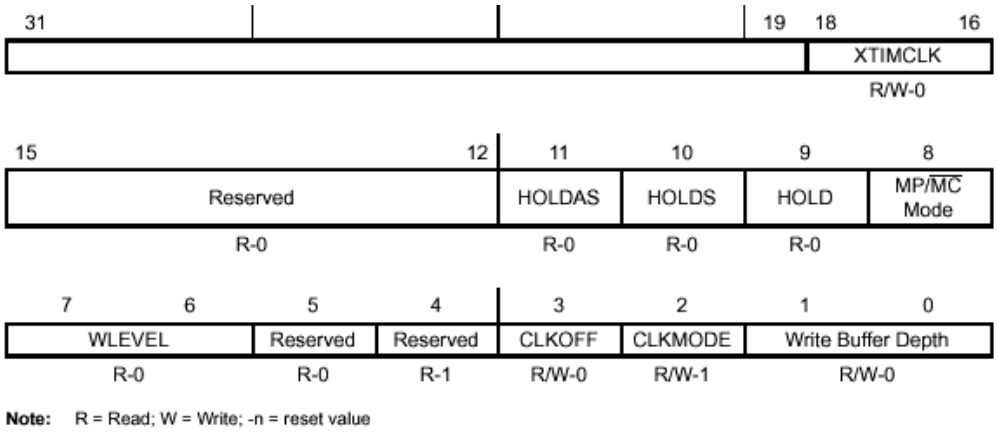


Figura 16: Registro XINTFCNF2

- Bits 18..16: Selección de la frecuencia de conmutación para la temporización en las conmutaciones lead, active y trail (‘000’, XTIMCLK=SYSCLKOUT; ‘001’, XTIMCLK= 1/2 SYSCLKOUT)
- Bit 11: Refleja el estado de la señal de salida !XHOLDA, indicando si la interfaz externa concede acceso al dispositivo externo.
- Bit 10: Refleja el estado de la señal de entrada !XHOLD, indicando si el dispositivo externo pide acceso al bus externo.
- Bit 9: Concede, según los valores de las señales XHOLDA y XHOLD, una petición a los dispositivos externos.
- Bit 8: En el reset, este bit refleja el estado de la señal XMP/!MC.
- Bits 7..6: número actual de escrituras en el buffer.
- Bit 3: Permite apagar el reloj XCLOUT.
- Bit 2: Permite configurar XCLOUT a la misma frecuencia que XTIMCLK o la mitad.
- Bits 1..0: Indica la profundidad del buffer de escritura.

Una vez realizada la configuración conveniente, lo primero que hay que determinar es la cantidad de estados de espera adicionales que son

necesarios, para ello, hay que remitirse a la hoja de características AC del controlador del LCD (Tabla 9), en ella se observa la anchura del pulso de enable necesario para el caso de escritura:

esta toma un valor de 450ns, en caso de que la alimentación se encuentre en el rango de (2,7 a 4,5) V; y 230ns, si la alimentación está entre (4,5 y 5,5) V.

Si se realizan las operaciones pertinentes, teniendo en cuenta que la frecuencia de reloj del DSP es 150 MHz y que su configuración hace que el XTIMCLK (reloj para al acceso a las distintas zonas del XINTF) sea de 75 MHz, se necesitan, en el peor de los casos 19 estados de espera. Para este caso pues, se realiza el diseño, para conseguir así una mayor generalidad, si bien, puesto que la alimentación va a 5V, los realmente necesarios son 8.

$$450x10^{-9} = \alpha \frac{1}{75x10^{6}}$$

$$\alpha = 33,75\ ;\ 34 \rightarrow 34 = XRDACTIVEx2 + WS + 1$$

$$WS = 19$$

Para conseguir estos estados de espera, se utiliza un contador, de tal forma que, hasta que este no alcance un valor dado como entrada, no se activará la señal XREADY.

3.3. Programación de la PAL22V10

La realización de los pasos descritos en la sección anterior se lleva a cabo mediante un programa implementado en Wincupl que se carga en una PAL22V10 de la marca Lattice, gracias al programa Pg4uw.exe.

Si se revisa el código del mismo, las primeras ecuaciones que aparecen son las correspondientes a un contador de 5 bits implementado mediante biestables tipo D. El esquema del mismo es:

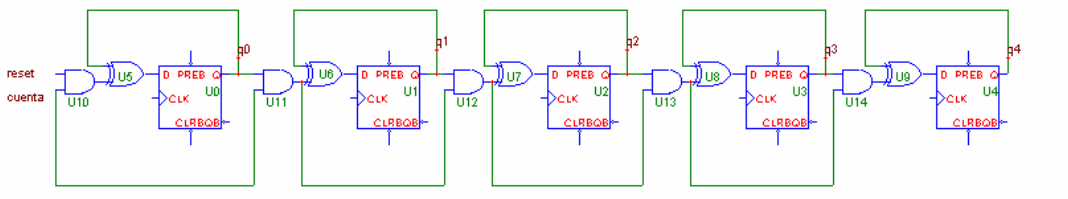


Figura 17: Contador 5 bits

La parte final del código implementa la máquina de estado que da lugar a la señal XREADY perseguida. El diagrama de estados de la misma es el que se muestra a continuación:

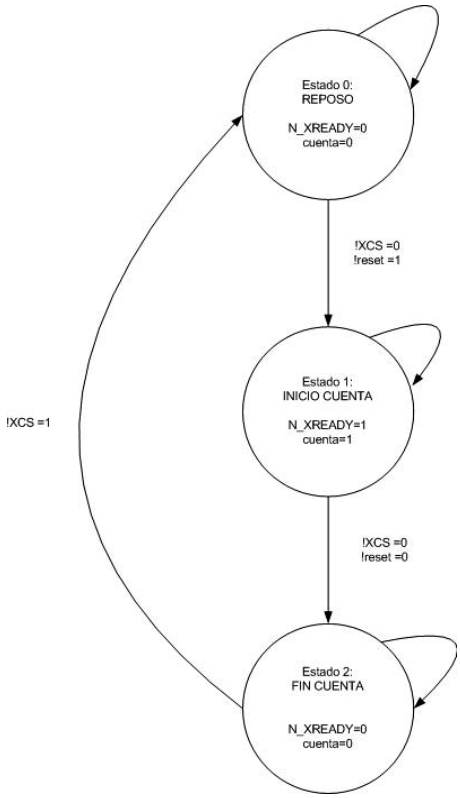


Figura 18: Diagrama de estados

La configuración final de los distintos pines de la PAL22V10 resultó:

```
                 _____________
                |  mipalpin2  |
          CLK x---|1          24|---x Vdd
           m0 x---|2          23|---x q4
           m1 x---|3          22|---x qout
           m2 x---|4          21|---x qsec
           m3 x---|5          20|---x XCS
           m4 x---|6          19|---x reset
  !RES_ASINC x---|7          18|---x q0
        N_XWE x---|8          17|---x q1
       N_XOCS x---|9          16|---x q2
              x---|10         15|---x q3
              x---|11         14|---x XREADY
          Vss x---|12         13|---x
                |_____________|
```

Figura 19: Diagrama del chip

Nótese que la asignación de señales a los distintos pines se realizó a fin de facilitar, en la máxima medida posible, el rutado de la placa realizada para la comprobación del funcionamiento de esta aplicación.

Para finalizar esta parte, se adjunta el código y la simulación resultante del mismo. Para la correcta interpretación de la simulación debe notarse que, las líneas en verde, corresponden a entradas al dispositivo; así, las señales m0…m4 son los valores de los bits correspondientes al número de estados de espera, N_X0CS representa la señal de selección de la zona 0 del XINTF (la realmente llamada XZCS0AND1n) y, N_XWE, la señal de selección de escritura (XWEn).

Por otro lado, q0…q4, son los bits del contador, desde el menos significativo, al más significativo; XCS es el enable del display y XREADY, representa la señal de idéntico nombre.

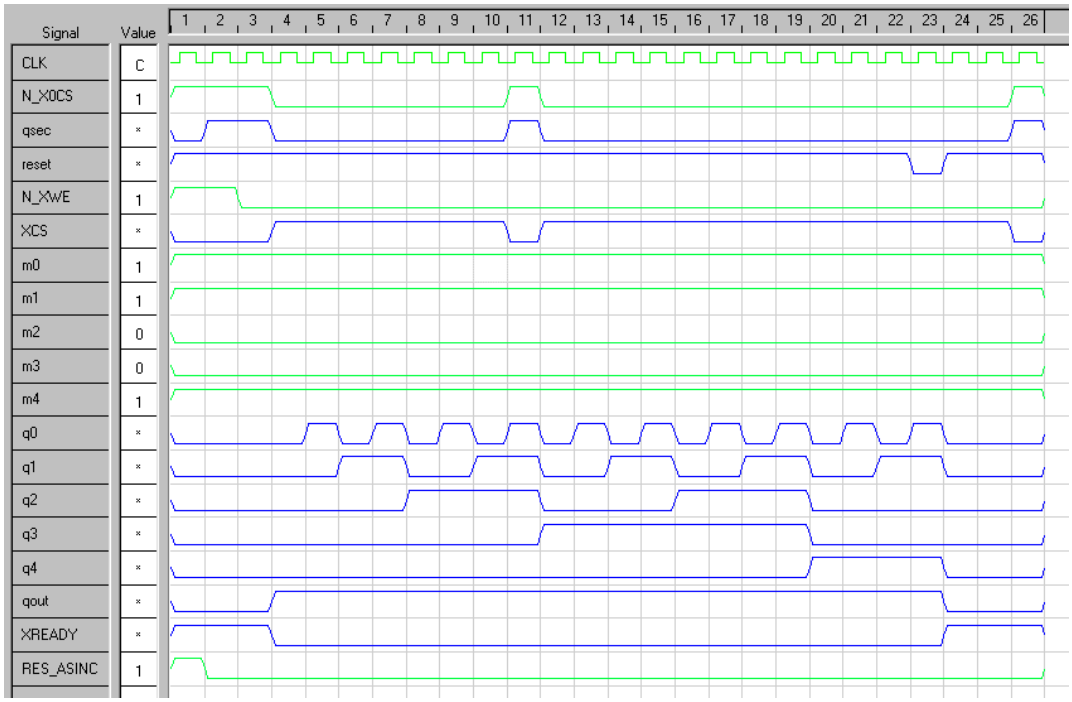


Figura 20: Resultados de simulación

3.3.1. Núcleo del código en Wincupl

```
/* variables intermedias del contador*/
reset   = ( q0 $ m0 )# (q1 $ m1 )#( q2 $ m2 )#( q3 $ m3 )#( q4 $ m4 );
```

```
/* Ecuaciones intermedias de la maquina de estado*/

XCS = !(N_XWE # N_X0CS);
XREADY = !qout;

/* ecuaciones logicas del contador*/

q0.ar = RES_ASINC;
q1.ar = RES_ASINC;
q2.ar = RES_ASINC;
q3.ar = RES_ASINC;
q4.ar = RES_ASINC;
qout.ar = RES_ASINC;
qsec.ar = RES_ASINC;

q0.d = reset & (q0 $ qout);
q1.d = reset & (q1 $ (q0&qout));
q2.d = reset & (q2 $ (q1&q0&qout));
q3.d = reset & (q3 $ (q2&q1&q0&qout));
q4.d = reset & (q4 $ (q3&q2&q1&q0&qout));


/*Ecuaciones que describen el funcionamiento de la maquina de estado*/

qsec.d = !XCS # (qsec & !reset);
qout.d = (qout & reset) # (qsec & XCS & reset);
```

3.4. Diseño de la placa de pruebas

Para la conexión de los distintos dispositivos ya estudiados y configurados, otro problema es el que debe ser resuelto en este momento: la adaptación de los niveles de señal.

La PAL22V10 utilizada se alimenta a una tensión de 5V y sus salidas son CMOS compatibles TTL, en consecuencia, no soportables por las entradas del DSP. Este es el problema al que nos enfrentamos con la señal XREADY.

La solución la encontramos en una página sobre la aplicación de este DSP en el control de motores digitales (véase Bibliografía). Según esta información, la forma adecuada de conectar una salida CMOS a 5 V a una entrada CMOS del DSP a 3,3 V, es mediante un divisor de resistencias de valores 18 KΩ y 22 KΩ. Con estos valores se consigue, que una salida a 5,25 V de un componente off-chip, se vea a la entrada del DSP como 2,9 V.

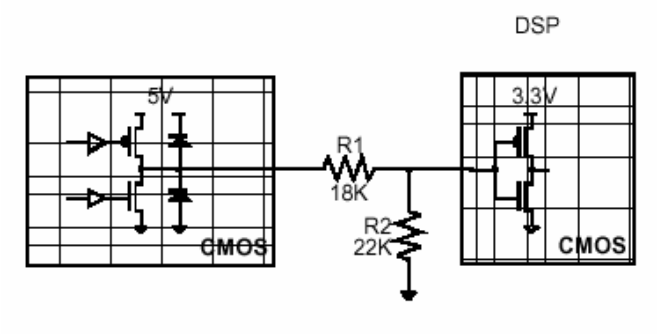


Figura 21: Esquema de conexión para la señal XREADY

Realmente, si se observa la placa construida, estos valores de resistencias no son los montados, ya que tras un estudio de la placa con ayuda de un osciloscopio digital se comprobó la conveniencia de disminuirlos en un orden de magnitud; de forma que, para la adaptación de la señal XREADY a la entrada del DSP, se usaron resistencias de 1'8KΩ y 2'2KΩ en idénticas posiciones a las de la figura.

Por último, y como finalización de esta parte, se realizó la correspondiente placa con el programa P-CAD, cuyo esquemático y huella pueden consultarse en el apartado de planos.

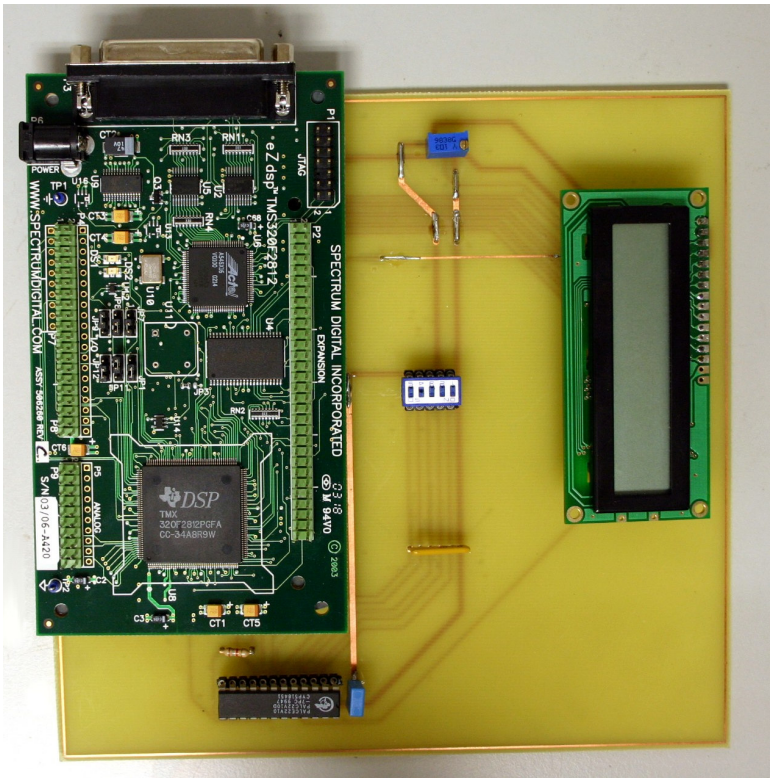


Figura 22: Placa de conexión del LCD al bus

4. Conexión de un display al DSP por medio del SPI

La operación que se describe en este apartado es muy semejante a la anterior, de hecho, el dispositivo display LCD que se emplea, es el mismo; no obstante, existen claras diferencias en cuanto a hardware y software se refiere.

Ya se ha descrito con anterioridad las peculiaridades del periférico SPI del DSP, quizá sea mejor, antes de abordar su configuración, comentar la conexión hardware, pues así puede entenderse mejor el sentido de las señales utilizadas.

4.1. Conexión hardware de los dispositivos

En este caso, el display no puede conectarse directamente a las señales del periférico SPI, puesto que, como se sabe, la salida del SPI es un flujo de datos en serie que, antes de ir hacia el display, deben ser pasados a paralelo. Para conseguir esto, se utiliza un nuevo dispositivo, consistente en un registro de desplazamiento de 8 bits, el 74HC595.

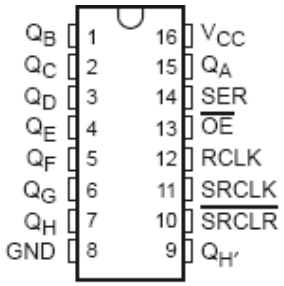


Figura 23: Vista del dispositivo 74HC595

Lo primero a destacar es la señal SER, entrada de datos serie al dispositivo que, posteriormente saldrán divididos en bits por los pines Q_A a Q_H.

Este registro de desplazamiento está formado por dos columnas de registros. En la primera de ellas, se desplaza la información (primera operación) y, a continuación, una vez desplazada, se almacena en la segunda columna del registro. Estas dos acciones, desplazar y guardar información, son gobernadas por las dos señales de reloj que le entran al dispositivo, SRCLK y RCLK, las cuales están desfasadas medio ciclo. Esto permite que con el flanco de subida de cada una de ellas se produzcan las operaciones descritas. En el

flanco de subida de la señal SRCLK se desplaza el dato al siguiente bit dentro del mismo registro y con el flanco de subida de la señal RCLK se produce el almacenamiento de la información en un nuevo registro.

Esta es la acción descrita en la hoja del fabricante y la que se puede observar en el diagrama de tiempo, no obstante, no será este, exactamente, el modo en que se usará este dispositivo.

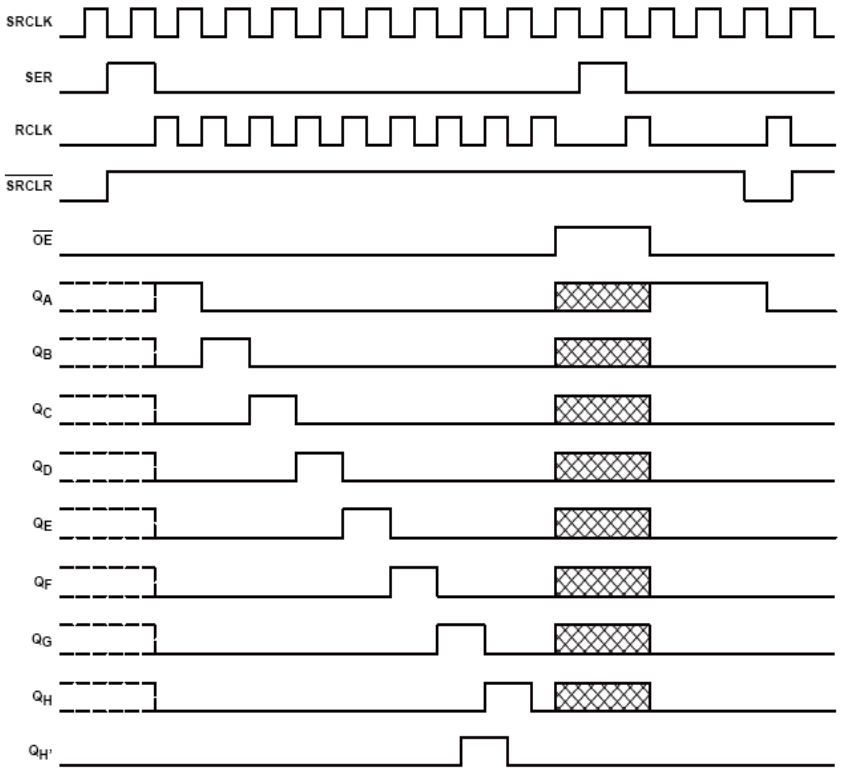


Figura 24: Diagrama de tiempo del 74HC595

En el esquemático del conexionado que se ha realizado (ver figura 25), se observa cómo la entrada SER está conectada con la salida SIMO del DSP, el reloj SRCLK, se corresponde con el SPICLK y, la señal RCLK se conecta a una salida de propósito general del DSP, concretamente a la salida GPIOF5; de modo que, cuando esta salida se active, tendrá lugar el volcado de los bits desplazados en el siguiente registro. Puesto que el bit de OE es activo a nivel bajo, al estar conectado a GND, la salida estará siempre habilitada, pero únicamente tendremos un valor en ella cuando GPIOF5 esté a '1'.

La conexión de los bits de datos de salida del 74HC595 es la lógicamente esperada, cableados a los bits de datos del LCD en el que se pretende escribir.

Sólo queda por resaltar, la conexión del pin SRCLR a Vcc, con lo que su acción queda anulada, ya que es activo a nivel bajo y no es conveniente que entradas digitales queden al aire.

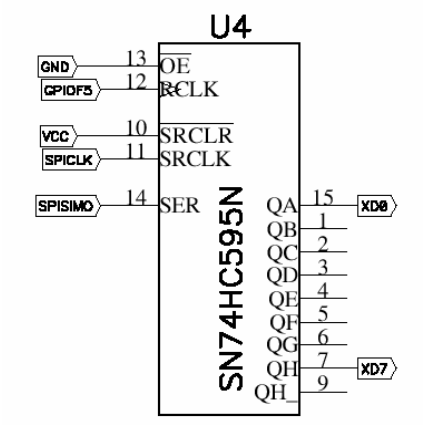


Figura 25: Conexión del integrado 75HC595

En cuanto al display se refiere, la conexión de sus señales se completa al indicar que, su señal de Enable coincide con la señal de SRCLK del 74HC595, es decir, la señal de propósito general GPIOF5, que será habilitada adecuadamente según programación del DSP. La señal de RS, también proviene de una salida de propósito general del DSP, GPIOF4, convenientemente programada; el resto de señales de llegada, son las mismas que en el apartado 3 anterior.

4.2. Configuración del DSP

Puesto que en este apartado vamos a utilizar el periférico SPI, se comienza el estudio de la configuración del DSP por este módulo.

4.2.1. Configuración del SPI

Para conseguir que el SPI se comporte como se desea, debemos cargar los valores adecuados en los registros SPICCR, SPICTL, SPIBRR y SPIPRI.

El registro SPICCR se compone de los siguientes bits.

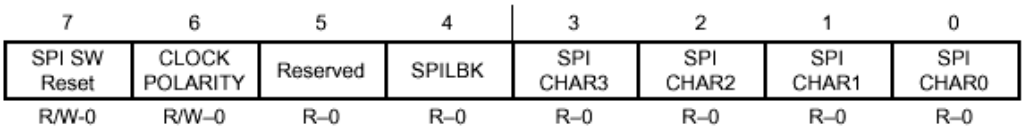


Figura 26: Registro SPICCR

El valor que se carga en este registro es 0x07, de forma que:

- Bit 7: a '0', resetea mientras se cambia la configuración, al final de la inicialización se pondrá a '1'.
- Bit 6: se encuentra a '0', por lo que se fija la polaridad del reloj SPICLK, La fase del reloj en este caso es 0; de forma que los datos son salidas en el flanco de bajada del SPICLK, y los datos de entrada son recogidos en el flanco de subida del SPICLK.
- Bit 4: al estar a 0, se encuentra deshabilitada la opción de Loopback, que sí se usó en la fase de simulación, y por tanto, se activará más adelante.
- Bits 3..0: se encuentran todos a '1', indicando que la longitud de los caracteres es de 8 bits.

El registro SPICTL es configurado de la siguiente forma al cargar el valor 0x0E:

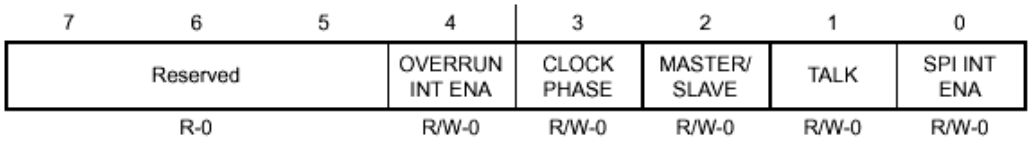


Figura 27: Registro SPICTL

- Bits 5..7: reservados.
- Bit 4: habilitación de la interrupción de sobreescritura desactivado.
- Bit 3: con el valor de este bit queda fijado el esquema de reloj a usar de entre los cuatro posibles. Al estar a '1', la señal de SPICLK es retrasada medio ciclo (véase figura 9).
- Bit 2: se elige el modo maestro de funcionamiento (a '1').
- Bit 1: habilita la transmisión al estar a '1'.
- Bit 0: al estar a '0' se deshabilita la posibilidad de generación de interrupciones del SPI por recepción o transmisión.

Antes de continuar viendo registros de configuración, es conveniente explicar por qué se elige este esquema de funcionamiento del reloj: flanco de subida con medio ciclo de retraso.

La explicación es sencilla si se observa la relación existente entre los datos que salen por la línea SPISIMO y el esquema de reloj seleccionado. Se puede observar (figura 9) cómo, el flanco activo de reloj se produce cuando el dato ya está estable, de forma que, cuando este flanco llegue a la entrada del 74HC595, el dato que va a comenzar a desplazarse, es recogido sin problema de la línea de entrada del susodicho integrado, haciéndose posible así, el correcto funcionamiento de esta parte del esquema.

El registro SPIBRR se inicializa a un valor de 0.

7	6	5	4	3	2	1	0
Reserved	SPI BIT RATE 6	SPI BIT RATE 5	SPI BIT RATE 4	SPI BIT RATE 3	SPI BIT RATE 2	SPI BIT RATE 1	SPI BIT RATE 0
R-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

Figura 28: Registro SPIBRR

En consecuencia, el régimen en baudios del SPI será ¼ LSPCLK, donde LSPCLK es el reloj resultante de dividir el SYSCLKOUT entre el valor dado por el campo de 3 bits LSPCLK, del registro LOSPCP. Siendo su funcionamiento como sigue:

000: LSPCLK = SYSCLKOUT

001: LSPCLK =1/2 SYSCLKOUT

010: LSPCLK = 1/4 SYSCLKOUT (opción por defecto)

011: LSPCLK =1/6 SYSCLKOUT

100: LSPCLK =1/8 SYSCLKOUT

101: LSPCLK = 1/10 SYSCLKOUT

110: LSPCLK = 1/12 SYSCLKOUT

111: LSPCLK = 1/14 SYSCLKOUT

En la aplicación desarrollada se hizo uso de la FIFO, por lo que, brevemente, se revisa la inicialización de esta.

En el registro SPIFFTX se carga un valor de 0xE040.

Figura 29: Registro SPIFFTX

- ♣ Bit 15: Resetea los canales y registros de configuración a '0'.
- ♣ Bit 14: Cuando está activo a '1', habilita el uso de la FIFO.
- ♣ Bit 13: La escritura de un '1' en este bit permite rehabilitar la operación de la FIFO de transmisión.
- ♣ Bits 12..8: Indican el estado de la FIFO diciendo el número de palabras almacenadas.
- ♣ Bit 7: Bandera que indica la ocurrencia de una interrupción.
- ♣ Bit 6: Cuando se escribe un '1' en él se borra la bandera TXFFINT
- ♣ Bit 5: permite habilitar las interrupciones de la FIFO de transmisión cuando está a '1'.
- ♣ Bits 0..4: La FIFO de transmisión genera una interrupción cuando se alcanza un valor menor o igual que el cargado en estos bits en el campo TXFFST.

Por otro lado, en el registro SPIFFRX, el valor de inicialización es 0x204F.

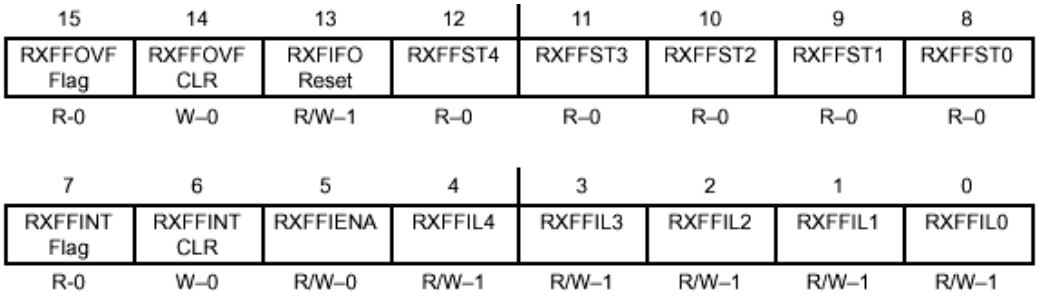


Figura 30: Registro SPIFFRX

Revisión de los bits que lo componen:

- ♣ Bit 15: bandera que se activa a '1' cuando se produce desbordamiento de la FIFO de recepción.
- ♣ Bit 14: Cuando se escribe '1' en este bit se borra la bandera RXFFOVF.
- ♣ Bit 13: La escritura de un '1' en este bit permite rehabilitar la operación de la FIFO de recepción.
- ♣ Bits 12..8: Indican el estado de la FIFO diciendo el número de palabras almacenadas.
- ♣ Bit 7: Bandera que indica la ocurrencia de una interrupción.
- ♣ Bit 6: Cuando se escribe un '1' en él se borra la bandera RXFFINT
- ♣ Bit 5 : permite habilitar las interrupciones de la FIFO de recepción cuando está a '1'.
- ♣ Bits 4..0: La FIFO de recepción genera una interrupción cuando se alcanza un valor mayor o igual que el cargado en estos bits en el campo RXFFST.

4.2.2. Configuración del GPIO

Para usar las salidas digitales de propósito general, es necesario actuar sobre determinados registros, tales como son GPxMUX y GPxDIR. Concretamente, se va a actuar sobre GPFMUX y GPFDIR.

La escritura de '1' en los bits del GPFMUX permite asociar los pines a los periféricos correspondientes, mientras que '0', hace que se puedan tratar como entradas/salidas de propósito general. La dirección de estos pines es elegida en el registro GPFDIR ('1' configura el pin como salida y '0' como entrada)

Los bits que componen estos registros son:

Bit GPFMUX	Nombre asociado al periférico (bit ='1')	Nombre del GPIO (bit = '0')
0	SPISIMO	GPIOF0
1	SPISOMI	GPIOF1
2	SPICLK	GPIOF2
3	SPISTE	GPIOF3
4	SCITXDA	GPIOF4

5	SCIRXDA	GPIOF5
6	CANTX	GPIOF6
7	CANRX	GPIOF7
8	MCLKX	GPIOF8
9	MCLKR	GPIOF9
10	MFSX	GPIOF10
11	MFSR	GPIOF11
12	MDX	GPIOF12
13	MDR	GPIOF13
14	XF	GPIOF14

Tabla 10: Significado de los bits del registro GPFMUX

Para finalizar la configuración de este módulo, se da un valor inicial a los pines seleccionados como entrada/salida de propósito general escribiendo en el registro GPFDAT.

Las líneas del código correspondiente son:

```
// Pines del GPIO asociados al SPI
   EALLOW;
   GpioMuxRegs.GPFMUX.all=0x000F;   // GPIOs  asociados a pines SPI
   GpioMuxRegs.GPFDIR.bit.GPIOF4=1;    //Configurados como salidas
   GpioMuxRegs.GPFDIR.bit.GPIOF5=1;   //  Serán  Enable  y  R_S  del  LCD
EDIS;
// Valores iniciales para las salidas de propósito general
      GpioDataRegs.GPFDAT.bit.GPIOF4=0;
      GpioDataRegs.GPFDAT.bit.GPIOF5=0;
```

4.3. Diseño de la placa de pruebas

El diseño coincide con el ya mostrado en el esquemático de la figura 25. En él puede observarse la adición de un conjunto de diodos y resistencias no contemplado en el apartado 4.1. La explicación de su inclusión proviene de la necesidad de adaptar los niveles de tensión de la salida del DSP a 3,3 V, a los niveles de tensión demandados por el 74HC595, 5 V.

Si se consulta la página ya nombrada en la sección 3.4, la recomendación realizada es la siguiente:

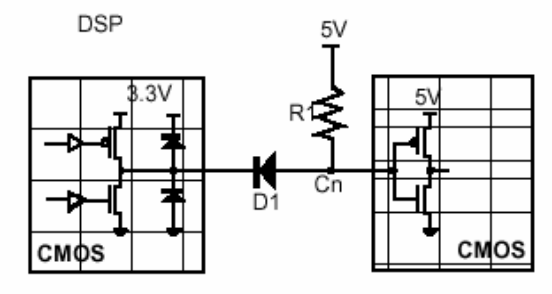


Figura 31: Conexión de una salida CMOS a 3,3V a una entrada CMOS a 5V

Al analizar someramente el comportamiento de esta conexión, se puede apreciar cómo la salida del DSP, con R1 del orden de 10 KΩ, tomará valores de 0,2 o 3,3 V, en consecuencia, en el ánodo del diodo, se tendrán tensiones de 0,8 o 3,9 V. Si se tiene en cuenta que los valores umbrales de la entrada CMOS para asumir '0' o '1' son 1 y 3,5 V, ya se ha conseguido la adaptación deseada.

Este montaje es añadido para las señales GPIOF4, GPIOF5, SPISIMO y SPICLK, que son las que se encuentran en esta situación.

Para la comprobación del funcionamiento de este módulo no fue realizada la correspondiente placa, sino que se probó en una regleta tal como muestra la figura.

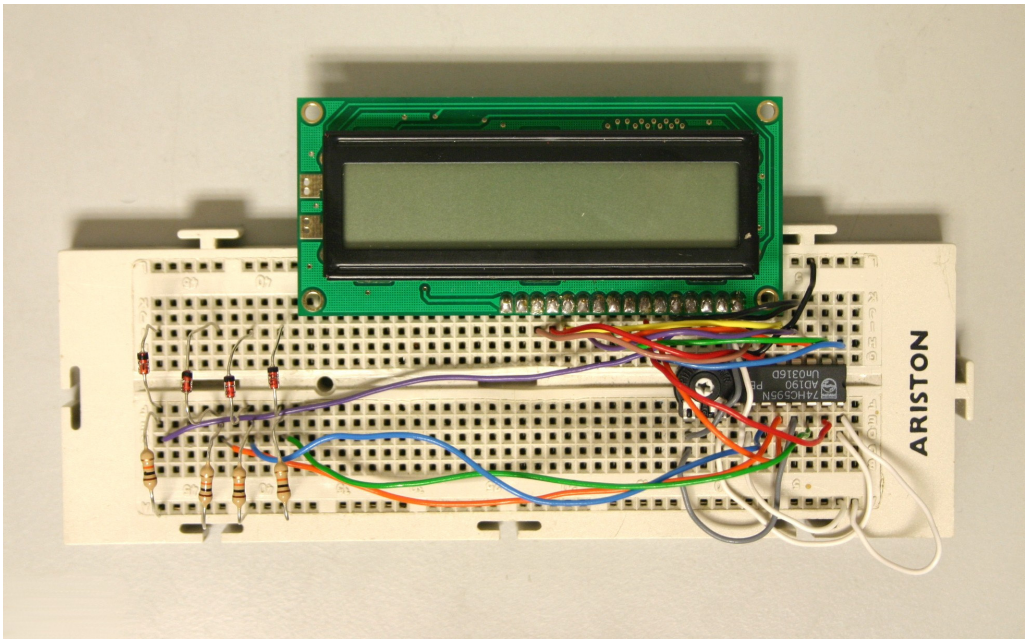


Figura 32: Montaje de conexión del LCD al SPI

5. Conexión de un teclado al DSP

Se trata en esta sección la conexión de un teclado al bus de datos del DSP, usando para ellos los integrados intermedios necesarios para la adecuada consecución del objetivo marcado.

Con este apartado, se pretende mostrar el uso del bus de datos como un bus de entrada, al que se van a volcar las teclas leídas, siendo este, el único caso en el que se le va a dar esta funcionalidad a lo largo del presente proyecto.

La consecución de este objetivo lleva también en su camino el estudio de la adaptación de señales de distintos niveles procedentes de distintos integrados al DSP.

5.1. Conexión hardware de los dispositivos

5.1.1. El teclado

Un teclado no es más que una matriz de líneas horizontales y verticales, en cuyos cruces, existen unos interruptores que se cierran en el momento en que una tecla es pulsada.

El teclado que se usa es de 4 filas por 3 columnas, tal como aparece en la figura 33.

Figura 33: Teclado

Para el correcto uso del mismo, las columnas están conectadas mediante resistencias a GND, mientras que las filas son muestreadas

periódicamente, conectándolas a través de otro conjunto de resistencias a Vcc. De este modo, se consigue que, si el dimensionado de las resistencias es el correcto, cuando coincida que una tecla sea pulsada y, al mismo tiempo, se esté muestreando la fila correspondiente a esta tecla, en la línea de su columna, en lugar de un '0', aparezca un '1'.

Para que se cumpla la condición descrita, las resistencias que van conectadas desde las líneas de las columnas a GND son un orden de magnitud superior a las conectadas entre las líneas de las filas y Vcc; concretamente, en el caso que nos ocupa, 4,7KΩ y 470 Ω.

A partir de lo expuesto hasta ahora, se deduce la necesidad de generar un reloj a la frecuencia adecuada para el muestreo de las filas, una máquina de estado que sea la que muestree las susodichas filas y, por último, una máquina de estado capaz de codificar las teclas en valores binarios que puedan ser enviados al DSP.

5.1.2. Generación del reloj

Para muestrear las teclas pulsadas, no es necesaria una frecuencia superior a 1KHz. Un reloj de esta frecuencia puede ser obtenido a partir del SYSCLKOUT, pero debido a su elevada frecuencia, 150 MHz, resulta más eficiente generar este reloj a partir de otro integrado: el NE555.

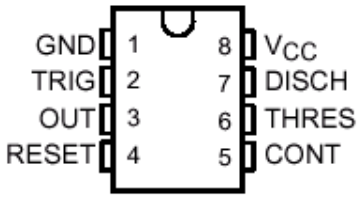


Figura 34: Vista del NE555

Este dispositivo es un circuito temporizador capaz de originar precisos retrasos de tiempo u oscilaciones. Para producir cada una de estas salidas, debe utilizarse en dos modos de funcionamiento distintos; monoestable o astable.

En el primero de los modos, el intervalo de tiempo es controlado por una resistencia externa y un condensador; mientras que, en el modo de funcionamiento inestable, la frecuencia y el "duty cycle" son controlados

mediante dos resistencias externas y un condensador. Este último modo es el que va a ser usado, para ello se determinan los valores de los elementos adecuados para la frecuencia de 1KHz deseada.

El montaje y las fórmulas de cálculo necesarias se toman de la hoja de características del dispositivo.

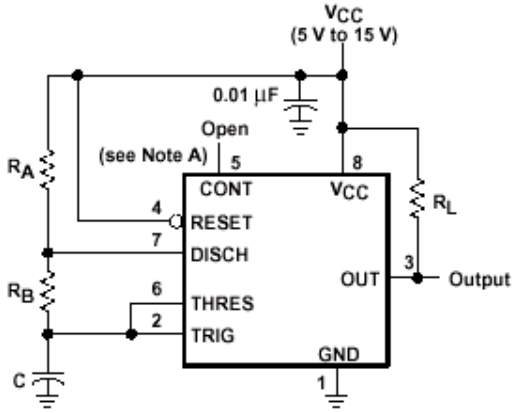


Figura 35: Esquema de conexión para funcionamiento astable

$$frequencia = \frac{1,44}{(R_A + 2R_B)C} \qquad \frac{t_L}{t_H} = \frac{R_B}{R_A + R_B}$$

La señal que se va a obtener no es exactamente la de un reloj, porque no se impone que el tiempo que esta se mantiene a nivel alto coincida con el tiempo que permanece a nivel bajo; es decir, t_H no tiene por qué ser igual a t_L. Esto no supone ningún impedimento para el correcto funcionamiento del circuito, puesto que el dispositivo que recibe y funciona según esta señal, se activa con su flanco de subida, algo que sucede periódicamente, como se espera de un CLK.

De forma que, sólo se busca conseguir una frecuencia de 1KHz.

La elección de los valores de resistencias, 4K7 y 1KΩ, así como el valor del condensador, 1 µF se basa en el hecho de que estos son valores normalizados que se ajustan a la necesidad presente.

Con ellos, la frecuencia real a la que va a funcionar el circuito es 1'384 KHz.

5.1.3. Máquinas de estado

En primer lugar debe realizarse un sistema que muestree las filas, es decir, que ponga un '1' en cada una de ellas periódicamente. La forma más sencilla de conseguir esto es mediante un registro de desplazamiento circular, precargado con "1000", de manera que el "1" va moviéndose de un bit al siguiente a cada ciclo de reloj, se consigue así una salida de "0100", "0010", "0001" y vuelta al principio.

La señal de reset síncrono, R_S, permite iniciar el funcionamiento del sistema. Teniendo en cuenta que es activa a nivel bajo, para poner a '0' un biestable en función de ella, basta con multiplicar su entrada por R_S y ya se tiene implementado el reset síncrono, mientras que para ponerlo a '1' hay que sumarle a la entrada la señal R_S negada. Por tanto el circuito resultante es como se observa en la figura.

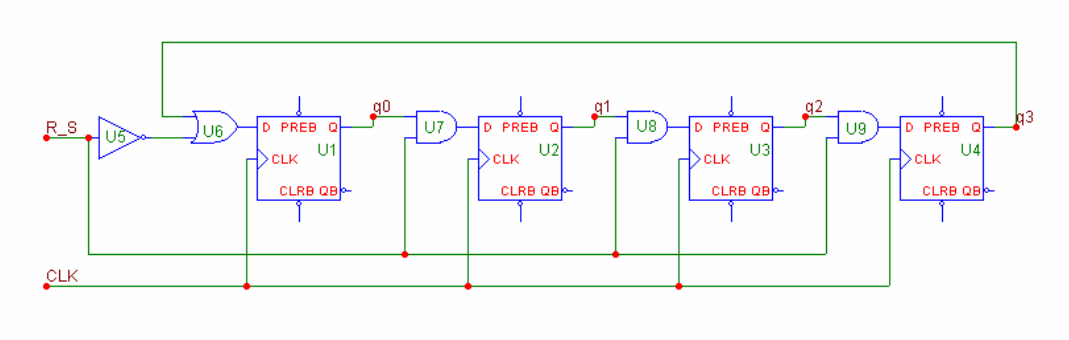


Figura 36: Desplazador circular

La siguiente tarea, una vez resuelto el muestreado de las filas, consiste en implementar un detector y codificador de las teclas pulsadas.

Puesto que el teclado consta de doce teclas, para la codificación binaria se necesitan 4 bits. La tabla muestra la codificación elegida.

	B3	B2	B1	B0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0

9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	0	1	1
ERROR (E)	1	1	1	1

Tabla 11: Codificación de teclas

Esta codificación podría resolverse mediante una máquina de estado, pero para evitar el complejo desarrollo en mapas de Karnaugh, las ecuaciones lógicas se deducen de la observación directa teniendo en cuenta en qué casos se activan cada uno de los bits.

La señal de error, E, que aparece ahí codificada, se activa, es decir, se pone a '1', cuando más de una columna está pulsada al mismo tiempo.

La detección de las teclas pulsadas, se consigue al hacer que la codificación sea válida si se activa una columna, o se mantenga el estado anterior, en caso contrario.

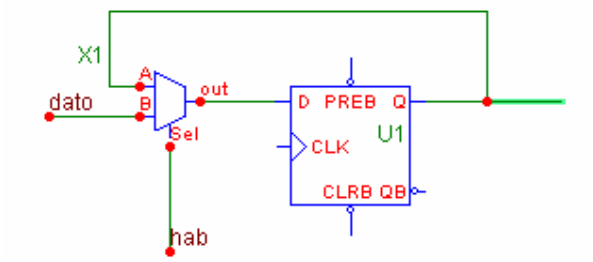


Figura 37: Celda del codificador

La implementación de estos bloques se lleva a cabo mediante la programación en Wincupl y, posterior cargado del programa con Pg4uw.exe en una PALCE22V10 de la marca Lattice, como ya se hizo para la primera aplicación desarrollada.

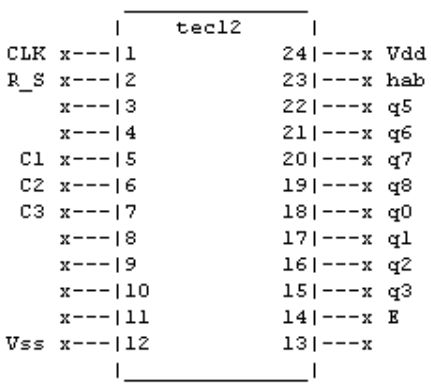


Figura 38: Diagrama del chip

Las señales de entrada son las ya explicadas, columnas activas, reset síncrono y reloj; en cuanto a las de salida, destacar la señal hab, que se activa a nivel alto en caso de que alguna de las columnas se haya activado.

El resultado de la simulación del funcionamiento de esta pal es el siguiente.

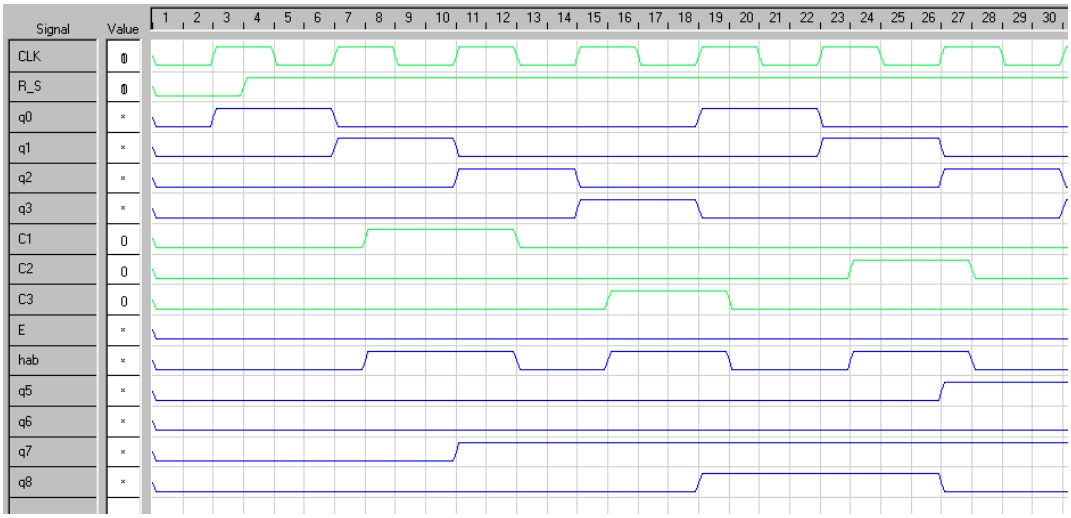


Figura 39: Primera parte de la simulación.

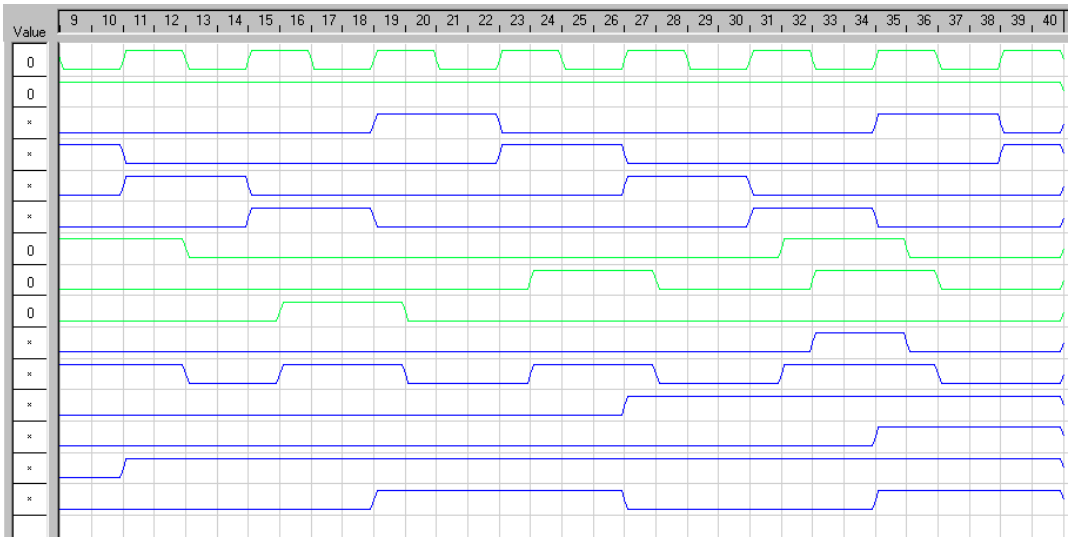


Figura 40: Segunda parte de la simulación.

Del análisis de la figura, puede verse cómo q0, q1, q2 y q3 están a nivel alto secuencialmente en el primer flanco de reloj tras el reset síncrono; la señal de hab se activa también al mismo tiempo que lo hace alguna de las columnas; y cómo q4, q5, q6 y q7 codifican adecuadamente el valor de la tecla pulsada en el primer flanco de subida del reloj tras la detección de la tecla.

5.1.4. Núcleo del código en Wincupl

```
/*Ecuaciones intermedias*/

hab = C1 # C2 # C3;
E = C1&C2 #  C1&C3 # C2&C3;

/*Ecuaciones de funcionamiento*/


q0.d = !R_S # q3;
q1.d = R_S & q0;
q2.d = R_S & q1;
q3.d = R_S & q2;


/*Identificacion de teclas */

q5.d= (q0 & C1 # q0 & C3 # q1 & C2 # q2 & C1 # q2 & C3 # q3 & C2 # E) &hab # q5 & !hab;

q6.d = (q0 & C2 # q0 & C3 # q1 & C3 # q2 & C1 # q3 & C1 # q3 & C2 # E) &hab # q6 & !hab;

q7.d = (q1 & C1 # q1 & C2 # q1 & C3 # q2 & C1 # q3 & C3 # E) &hab # q7 & !hab;

q8.d = (q2 & C2 # q2 & C3 #  q3 & C1 # q3 & C2 # q3 & C3 # E) &hab # q8 & !hab;
```

5.1.5. Integrados auxiliares

Por último, para conectar la salida de datos al bus del DSP, debe garantizarse que estos se encuentren estables, por ello, entre la PALCE22V10 y las líneas del bus de datos, se sitúa el chip SN74LVTH573 correspondiente a latches con salidas triestado.

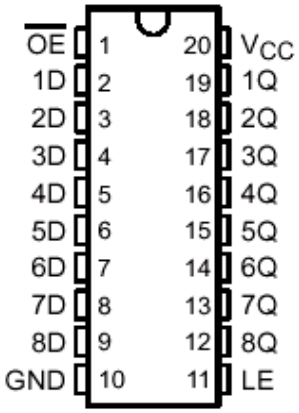


Figura 41: Vista del 74LVTH573

Para suplir la necesidad que se tiene, el latch enable (LE) de este dispositivo se mantiene siempre a nivel alto al ser conectado a Vcc, y el OE es el que permite volcar la salida al exterior según la lógica a continuación descrita.

El DSP leerá los datos procedentes de la PAL22V10, cuando se activen sus señales de lectura XRDn, la señal de selección de la zona adecuada, en este caso la zona 1 (XZCS0AND1n) y, además, los datos de la PAL22V10 estén estables. Esta última condición se garantiza cuando hab esté a '0', es decir, cuando no haya ninguna columna activa y no se esté pulsando ninguna de las teclas. Puesto que todas estas señales son activas a nivel bajo, el cumplimiento de todas se garantiza con la función OR de las mismas; esta es la línea de entrada al OE del latch 573.

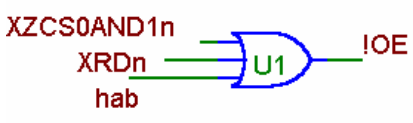


Figura 42: Obtención de !OE

Para la realización de esta función lógica se usó el integrado de puertas OR sn74ahct32n.

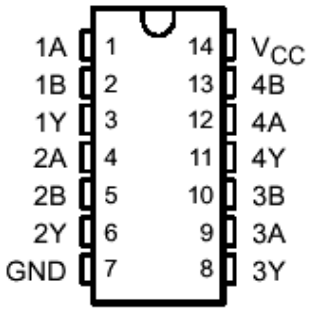


Figura 43: Vista del 74AHCT32

5.2. Configuración del DSP

En este caso, sólo cabe señalar que la zona elegida para el almacenamiento de las teclas recogidas es la zona 1 del XINTF, ubicada a partir de la dirección 0x4000.

En cuanto a la configuración de los relojes del DSP, el bit XTIMCLK se inicializa a '1', para conseguir una ejecución a 75 MHz y, en el registro XTIMING0, los bits USEREADY y READYMODE son '0' porque no son necesarios estados de espera adicionales.

5.3. Diseño de la placa de pruebas

El esquemático de la placa de pruebas es el que se muestra a continuación. En él puede observarse cómo es la interconexión entre los distintos dispositivos ya estudiados.

Lo único aún no comentado es el conjunto de resistencias existentes entre la salida del latch 573 y las líneas del bus de datos. Su funcionalidad es clara y ha sido ya explicada en apartados anteriores: permiten la adaptación de los niveles de tensión de las salidas a 5 V a los 3,3 V admisibles por las entradas del DSP. Los valores y configuración de las resistencias también coinciden con los del apartado 3.4 (Ver figura 21)

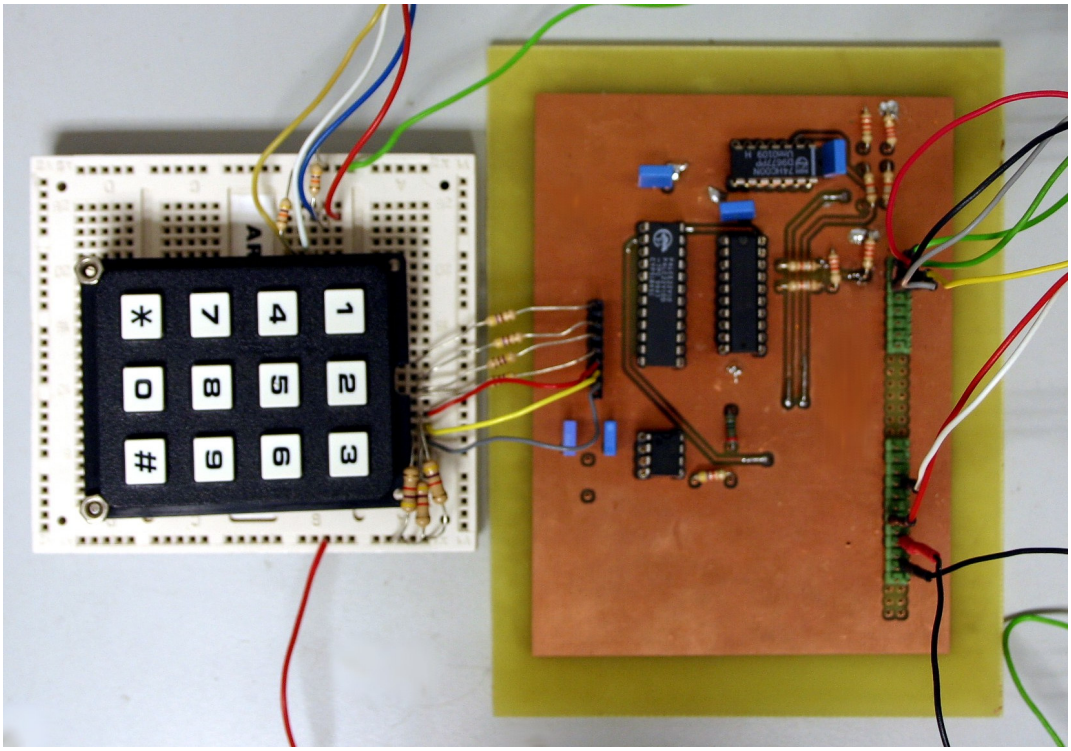


Figura 44: Placa de conexión del teclado al bus

6. Uso de los convertidores analógico/digital

El objetivo de este punto es mostrar el funcionamiento del módulo ADC del DSP, desde la configuración del mismo, pasando por la conexión hardware, hasta conseguir la salida esperada.

6.1. Conexión hardware de los dispositivos

Para la presente prueba, la conexión es bien sencilla: en este caso, sólo hay que introducir una señal analógica por el pin determinado del ADC, pero esto, debe hacerse teniendo en cuenta cuáles son las especificaciones eléctricas del DSP.

El rango de la tensión analógica de entrada tiene que ser de 0 a 3 V, pues si lo sobrepasamos, la conversión ya no será correcta. Para conseguir esto, así como funciones de filtrado de la señal y adaptación de impedancias a las entradas del ADC, se suelen usar circuitos como el de la figura 45.

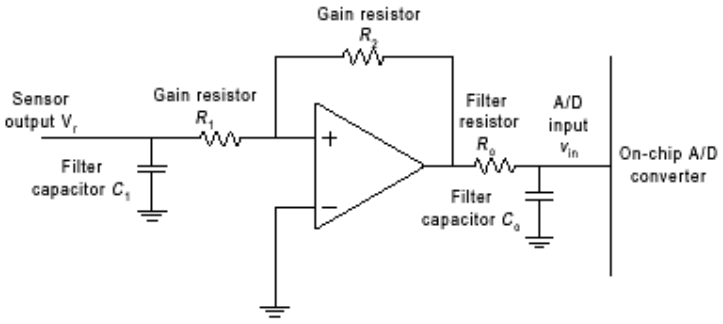


Figura 45: Interfaz para el ADC

El montaje finalmente implementado es más sencillo que el presentado, ya que, el amplificador operacional es usado como limitador, en configuración de seguidor de tensión.

El integrado elegido es el OPAMP4342 de Burr-Brown.

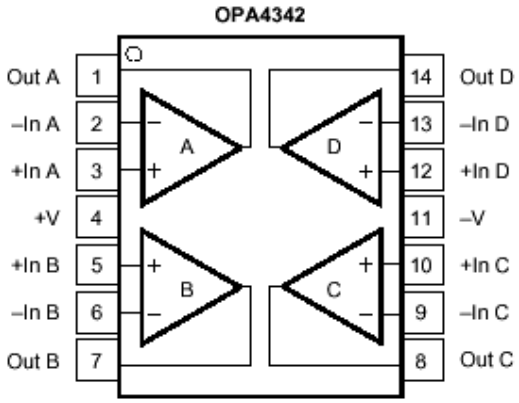


Figura 46: Pines del dispositivo

Atendiendo a la gráfica siguiente obtenida de su hoja de características, se puede observar cómo, si este dispositivo es alimentado a 3 V, esta será su tensión máxima de salida, cumpliendo entonces el propósito buscado.

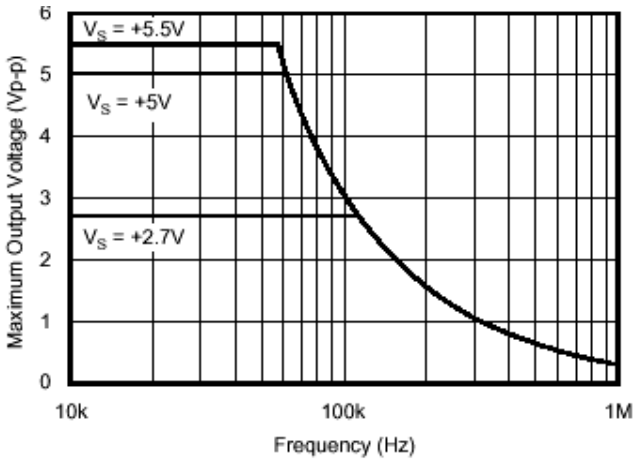


Figura 47: Tensión de salida máxima frente a frecuencia

Atendiendo también a esta gráfica, se elige la frecuencia de funcionamiento del orden de las decenas de KHz, concretamente, para las pruebas y simulaciones posteriores, 20 KHz. En consecuencia, para el diseño del filtro que aparece en la figura 45, se toma como frecuencia de corte 1 MHz.

El valor de R a elegir, viene determinado por las características eléctricas de este módulo. El sistema está alimentado a 3,3 V, pero a la entrada del pin sólo deben llegar 3 V como máximo; luego este exceso de tensión debe ser consumido en la resistencia. Para ello, es necesario conocer cuál es la

corriente circulante: según datos del fabricante, a la entrada de cada pin del ADC, existen un par de diodos conectados a las tensiones de alimentación y tierra, y la corriente continua que circula por este circuito es de 2 mA.

$$\frac{(3,3-3)V}{2x10^{-3}A} = 150\Omega \rightarrow 220\Omega$$

Se toma R =220 Ω, al ser este el valor normalizado más próximo a 150 Ω. Puesto que el polo deseado está a 1MHz, de la fórmula $\tau = RxC = 10^{-6}$ se despeja el valor de C, obteniéndose 4,5 nF, que se aproxima al valor normalizado 4,7 nF.

6.2. Configuración del DSP

6.2.1. Configuración del reloj del módulo

Para este apartado es básica la frecuencia de funcionamiento.

El reloj que gobierna el módulo de ADC del F2812 es el HSPCLK, cuya frecuencia viene dada por el valor de SYSCLKOUT dividido por el valor que se almacene en el campo de 3 bits HSPCLK del registro HISPCP.

Los valores posibles son:

000: HSPCLK = SYSCLKOUT

001: HSPCLK =1/2 SYSCLKOUT

010: HSPCLK = 1/4SYSCLKOUT

011: HSPCLK = 1/6 SYSCLKOUT

100: HSPCLK = 1/8 SYSCLKOUT

101: HSPCLK = 1/10 SYSCLKOUT

110: HSPCLK = 1/12 SYSCLKOUT

111: HSPCLK = 1/14 SYSCLKOUT

Este reloj de entrada al módulo es dividido entre el valor almacenado en el campo ADCCLKPS del registro ADCCTRL3, salvo que estos bits valgan '000'. Una división más entre 2 tiene lugar en caso de activación del bit CPS del registro ADCCTRL1.

6.2.2. Configuración del ADC

La configuración del ADC viene dada por los valores que se carguen en los registros que a continuación se detallan.

Los campos de los registros ADCCTRL1 y ADCCTRL3 no se detallan por no ser necesario en el desarrollo de la aplicación variar los valores que estos tiene por defecto tras un reset.

Registro ADCCTRL2:

15	14	13	12	11	10	9	8
EVB SOC SEQ	RST SEQ1	SOC SEQ1	Reserved	INT ENA SEQ1	INT MOD SEQ1	Reserved	EVA SOC SEQ1
R/W-0	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R-0	R/W-0
7	6	5	4	3	2	1	0
EXT SOC SEQ1	RST SEQ2	SOC SEQ2	Reserved	INT ENA SEQ2	INT MOD SEQ2	Reserved	EVB SOC SEQ2
R/W-0	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R-0	R/W-0

Figura 48: Registro ADCCTRL2

La revisión de los bits que lo componen, se fija sólo en los que hacen referencia al secuenciador 1, puesto que para el secuenciador 2, tienen significado análogo.

- Bit 15: para el secuenciador funcionando en modo cascada, permite habilitar el inicio de conversión (SOC) a partir de una señal procedente del EVB.
- Bit 14: reset del secuenciador 1
- Bit 13: Inicio de conversión del secuenciador 1.
- Bit 11: habilita la interrupción del secuenciador 1
- Bit 10: selecciona el modo de interrupción, cuándo se activa la bandera ING FLAG SEQ 1.
- Bit 8: bit de máscara para el inicio de conversión del secuenciador 1 por EVA.
- Bit 7: permite el inicio de una autoconversión en el secuenciador 1 a partir de una señal externa.

El registro MAXCONV, permite indicar el número máximo de conversiones ejecutadas en una sesión de autoconversión, ya se esté trabajando con alguno de los secuenciadores de forma independiente, o en modo cascada. Es destacable el hecho de que, el número de conversiones realizadas siempre excede en uno al valor en este registro cargado.

Los registros ADCCHSELSEQ1,2,3 y 4 permiten indicar para cada conversión, hasta las 16 posibles en modo cascada, qué secuenciador es el que se utiliza y cuál de sus entradas.

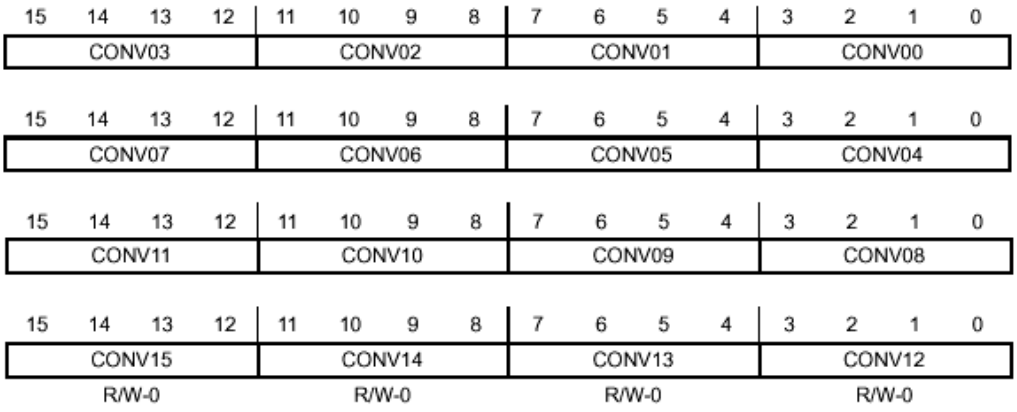


Figura 49: Registros ADCCHSELSEQ1, 2, 3 y 4

Así pues, para seleccionar el secuenciador 1, el valor cargado en CONVnn, tendrá como MSB '0', y los 3 bits restantes, determinan cuál es el canal de entrada. Si el MSB valiese '1', entonces el secuenciador elegido es el 2, y de igual modo, los 3 bits restantes codifican el canal de entrada. Por ejemplo, el valor 1010, selecciona el canal 2 del secuenciador 2.

6.2.3. Configuración del Event Manager

Como ya se ha comentado son distintas las fuentes que pueden disparar el inicio de conversión de los secuenciadores. Una de ellas, es a partir de una señal generada por los temporizadores del Event Manager.

El Event Manager es un módulo que proporciona gran cantidad de funciones y características especialmente útiles en el control de movimiento y aplicaciones de motores. Este módulo incluye temporizadores de propósito general, unidades de comparación/PWM, unidades de captura y circuitos de codificación de la cuadratura del pulso.

El Event Manager se divide en dos partes, EVA y EVB, totalmente idénticas.

Los posibles eventos que pueden ocurrir en un temporizador para dar lugar a la activación del bit SOC son desbordamiento, que el contador iguale un determinado valor de comparación o bien, que se alcance el periodo. Estas circunstancias vienen dadas por la configuración de los registros propios de este módulo.

Registro T1CON, a continuación se detallan los campos que lo componen.

15	14	13	12	11	10	9	8
Free	Soft	Reserved	TMODE1	TMODE0	TPS2	TPS1	TPS0
RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

7	6	5	4	3	2	1	0
T2SWT1/ T4SWT3†	TENABLE	TCLKS1	TCLKS0	TCLD1	TCLD0	TECMPR	SELT1PR/ SELT3PR†
RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

Figura 50: Registro T1CON

- Bits 15,14: Bits de control en simulación
- Bits 12..11: Selección del modo de cuenta; 00, parar/mantener; 01,cuenta continua arriba/abajo; 10, cuenta creciente continua; 11,cuenta direccionable arriba/abajo.
- Bits 10..8: Preescalado del reloj de entrada.
- Bit 7: Indica si se usa el TENABLE bit propio o el de otro contador
- Bit 6: Habilitador del temporizador
- Bits 5..4: Indica cuál es la fuente de reloj
- Bits 3..2: Condición de recarga del registro de comparación del temporizador.
- Bit 1: Habilitador de comparación del temporizador.

Registro GPTCONA, compuesto de los siguientes campos.

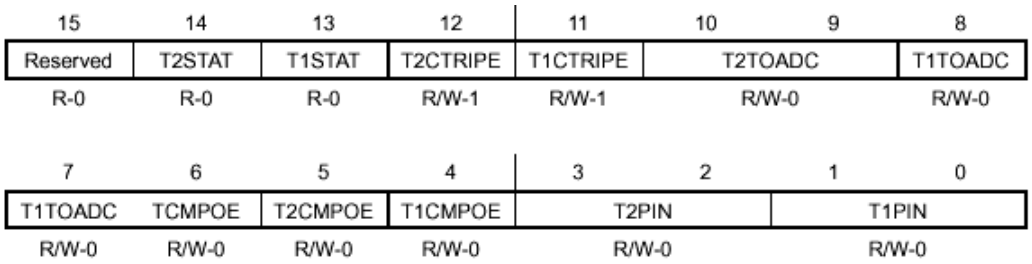


Figura 51 : Registro GPTCONA

- Bit 14: estado del contador de propósito general 2.
- Bit 13: estado del contador de propósito general 1.
- Bit 12: habilitación de T2CTRIP.
- Bit 11: habilitación de T1CTRIP.
- Bits 10..9: Indican el evento del temporizador 2 que inicia el ADC.
- Bits 8..7: Indican el evento del temporizador 1 que inicia el ADC.
- Bit 6: habilitación de la salida de comparación del temporizador.
- Bit 5: habilitación de la salida de comparación del temporizador 2.
- Bit 4: habilitación de la salida de comparación del temporizador 1.
- Bits 3..2: Indican la polaridad de la salida de comparación del temporizador de propósito general 2.
- Bits 1..0: Indican la polaridad de la salida de comparación del temporizador de propósito general 1.

6.3. Diseño de la placa de pruebas

Para poner en marcha un único convertidor, que es lo que se pretende en este módulo, basta con una placa tan sencilla como la apreciada en la figura, compuesta por un amplificador operacional, un filtro RC previamente diseñado y unas bornas para introducir la señal de entrada.

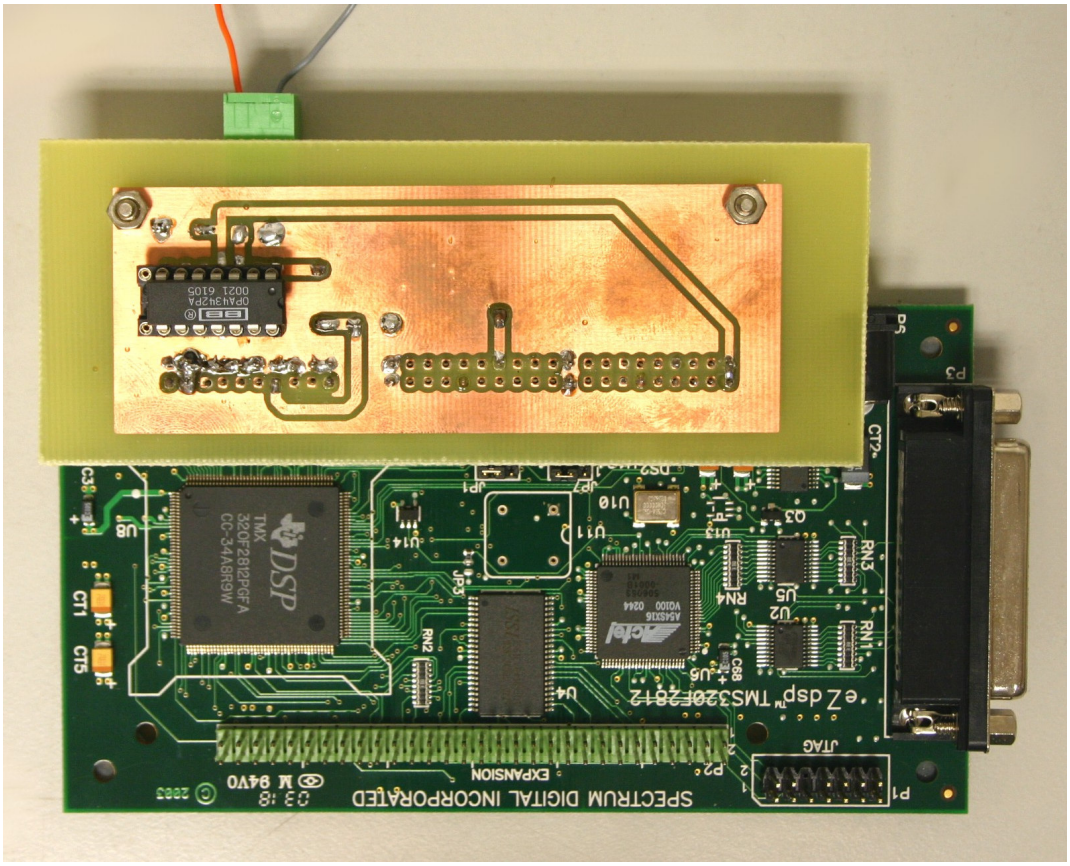


Figura 52: Placa de prueba del ADC

7. Unión de los módulos desarrollados

Como culminación del trabajo realizado, se presenta finalmente, la placa que incluye todos los pasos previos ya explicados.

7.1. Configuración del DSP

La configuración del DSP es un compendio de todas las configuraciones anteriores, teniendo en cuenta que no existe solapamiento entre ellas y que todos los módulos funcionan correctamente con independencia de la presencia del resto.

7.2. Diseño de la placa de pruebas

En cuanto a la conexión hardware de los dispositivos, se respeta lo expuesto en cada uno de los apartados previos, salvo por una pequeña variación incluida en la conexión del LCD al bus.

Previo al diseño de esta placa final, se realizaron distintas pruebas para la unión de los módulos de LCD y teclado. A partir de ellas, se observó cómo los niveles de las señales enviadas hacia el LCD, no eran los adecuados para que este fuera capaz de responder, por tanto, y tras comprobar la imposibilidad de utilizar un dispositivo desplazador de nivel de tensión, se optó por la utilización del integrado SN74LS245 para la conexión de los datos, y el conjunto diodo-resistencia (véase figura 31), ya utilizado para la adaptación de niveles de tensión en el módulo SPI, para la señal RS de entrada al LCD.

Este integrado es un transceptor de 8 bits que, alimentado a 5V, con entradas compatibles TTL, proporciona a su salida, señales de tensión más elevada que las dadas por el F2812, y que, por tanto, sí son reconocibles por el LCD.

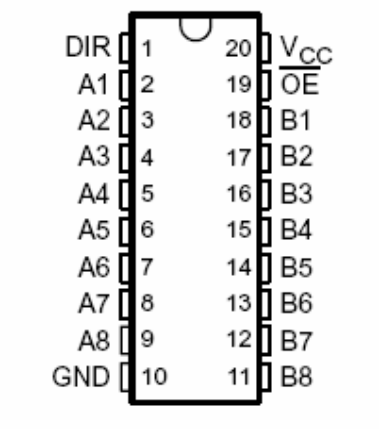


Figura 53: Vista del SN74LS245

La concreción de este argumento tiene lugar en la siguiente placa mostrada, que fue desarrollada para probar el adecuado funcionamiento del LCD y el teclado conectados al bus de datos.

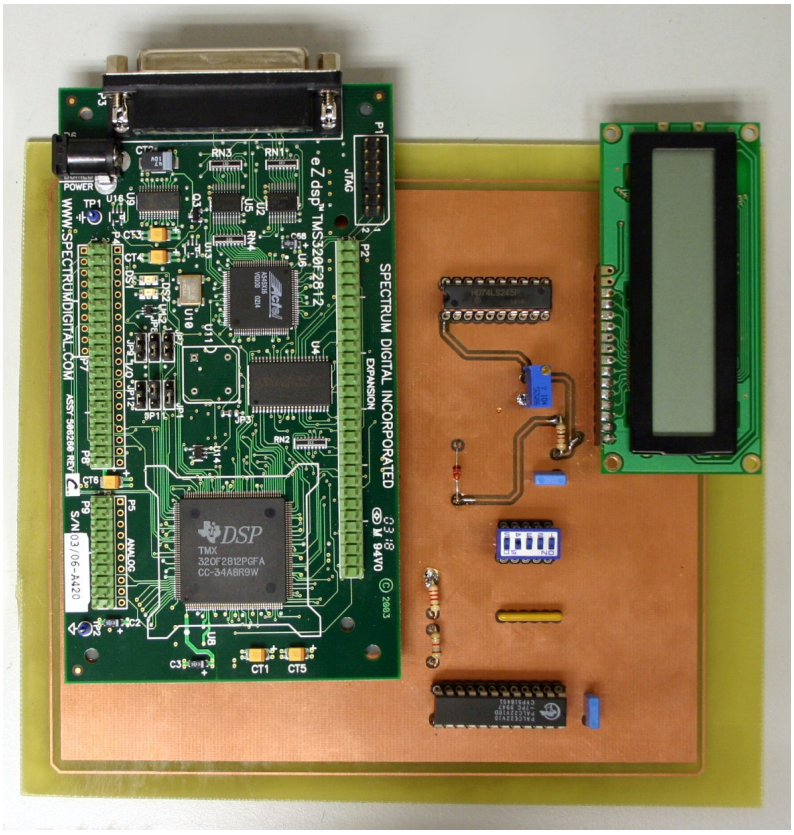


Figura 54: Placa para unión de módulos LCD y teclado

Tras la obtención de resultados satisfactorios en las pruebas realizadas con el montaje anterior, se diseñó la placa final, que puede observarse en la imagen siguiente.

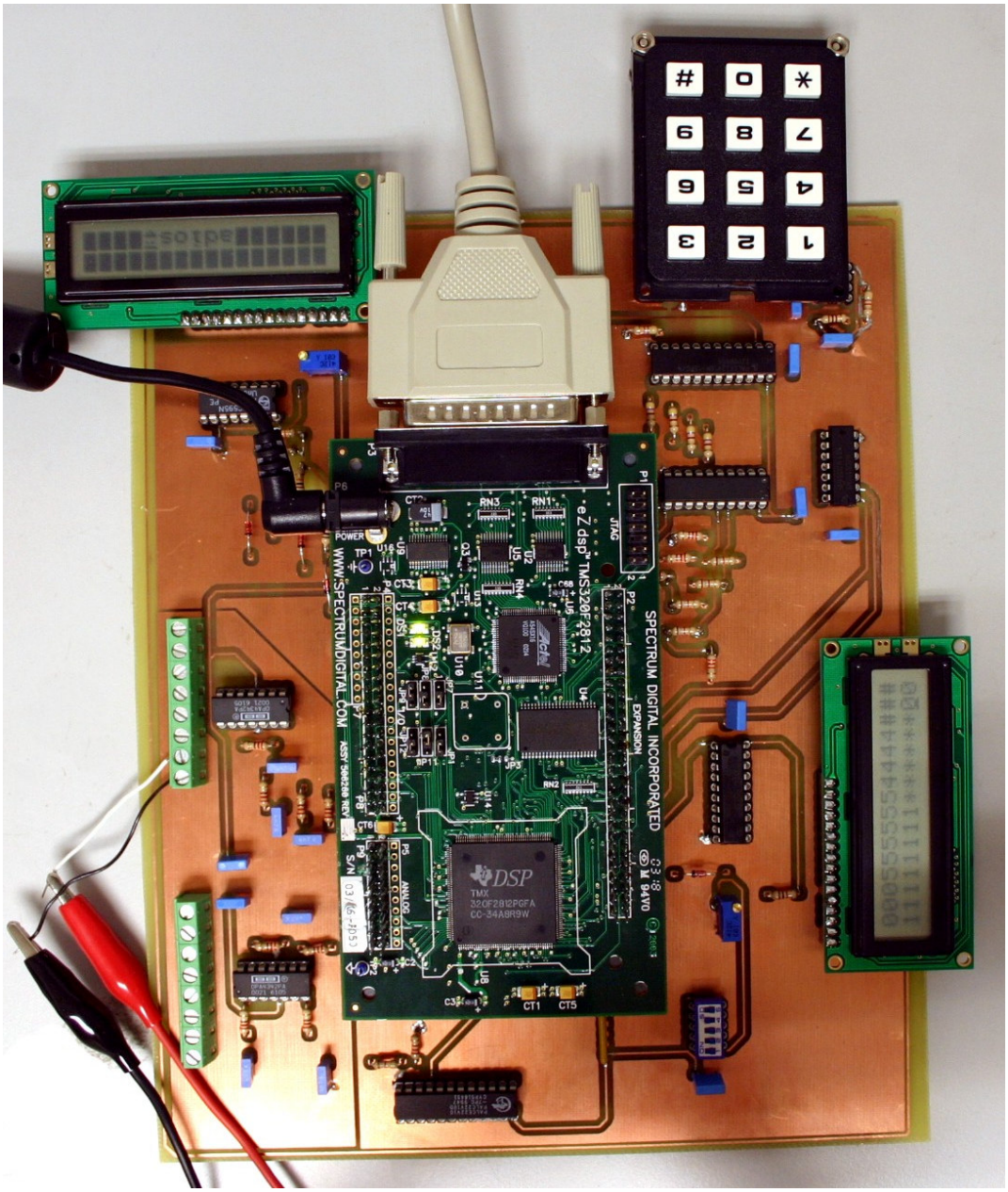


Figura 55: Placa final