

4. Manual de usuario

4.1 Introducción:

Esta es una primera versión del manual de usuario. Mediante el uso del sistema por parte de terceras personas será complementado con ejemplos y se solventarán los puntos que no queden claros. Por tanto las nuevas versiones de este manual serán publicadas en internet en la página web:

www.robozes.com

4.1.1 Descripción general

Bienvenido al sistema de desarrollo TC_FPGA+TSA_FPGA.

El sistema es una completa herramienta para el desarrollo de aplicaciones tanto software como hardware basadas en FPGAs + microcontrolador, orientadas al control de un sistema automático¹. Está formado por dos placas, una de control y otra de periféricos, un cable de programación, el software necesario para la realización de programas de aplicación tanto para la FPGA como el μ C, y la descarga y depuración de las aplicaciones. Las características principales del sistema son:

TC-FPGA (Tarjeta de Control)

Constituye el núcleo de control del sistema, estando formado por una FPGA y un μ C conectados a través de un bus de datos paralelo. Las características de estos circuitos son:

- **FPGA 10K10LC84-4 (ALTERA)**

Posee un total de 10.000 puertas lógicas, repartidas en 576 LEs², que forman 72 LABs³ y 3 EABs⁴. En total tiene 59 líneas de E/S.

- **μ C PIC16F877 (Microchip)**

Microcontrolador de 8 bits, arquitectura RISC en configuración Hardvar. 6Kx14 bits de memoria de programa, 360 bytes de memoria RAM y 256 bytes de memoria EEPROM de datos. Pila hardware de 8 niveles. En cuanto a recursos hardware posee una UART, I2C, 3 timers (1 de 16 bits y 2 de 8 bits) y 5 canales ADC.

Los recursos hardware del μ C y su memoria de programa son más extensos, sin embargo, este realiza la gestión de los programas de la FPGA así como la ayuda para la depuración de programas, a través de un programa empotrado en el mismo. Este programa nos limita la memoria real disponible para el usuario, así como su integración en el sistema elimina la posibilidad del uso de algunos recursos hardware del μ C. Por ello solo se indicarán en este manual aquellos recursos que realmente se pueden usar en este sistema.

¹ Tanto en el entorno de la domótica como robótica.

² Logic Elements, células lógicas reconfigurables.

³ Logic Array Block, formado por 8 LEs y la circuitería necesaria para su interconexión.

⁴ Embedded Array Blocks. Bloques de 256 bytes de memoria Ram

El sistema puede almacenar hasta 2 configuraciones distintas para la FPGA y 1 programa de usuario para el μ C al mismo tiempo. La descarga de programas para ambos dispositivos se realiza por medio de un puerto serie RS-232¹ y un PC. Además cuenta con una herramienta para la depuración de los programas de la FPGA, consistente en la posibilidad de observar y modificar registros de la FPGA en tiempo real a través del PC. Esta depuración también se realiza mediante el puerto serie RS-232.

TSA-FPGA (Tarjeta de Sensores y Actuadores)

Permite la interconexión de la tarjeta de control con elementos externos de forma sencilla y rápida, además cuenta con hardware adicional para el control de motores y ciertos sensores. Sus características básicas son:

- Sensor de tensión en la Batería
- 8 conectores de 3 líneas (+5V, Gnd, 1 señal de E/S)
- 8 conectores de 4 líneas (+5V, Gnd, 2 señales de E/S)
- 4 ADCs
- 4 Entradas específicas para el acoplamiento de sensores de IR con comparadores ajustables
- 1 conector para cable de bus 5x2 líneas (+5V, GND, 7 E/S)
- Control de 3 motores DC de hasta 5 A de pico y 2.26 A DC, con posibilidad de muestreo de corriente (mediante los ADCs).
- Alimentación: 6-15 V

La interconexión de ambas placas se realiza a través de conectores apilables, que permiten montarlas sin necesidad de separadores ni cables de conexión.

4.1.2 Contenido del paquete

Al desenvolver su paquete, usted debe asegurarse de que todo está intacto. Los siguientes productos deberían estar incluidos en su paquete:

- Tarjeta de control “TC_FPGA”.
- Tarjeta de aplicación “TSA_FPGA”.
- CD con el contenido:
 - Manual de usuario en formato **.pdf**.
 - Herramienta de desarrollo MPLAB
 - Manual² de usuario del MPLAB.
 - Herramienta² de desarrollo MAX+PLUS II
 - Tutorial² de la herramienta MAX+PLUS II (S.E.C.)³.
 - Tutorial de VHDL (Miguel Angel Freire)³.
 - Programa de comunicaciones para la gestión del sistema.
 - Librerías y ejemplos para diseñar programas de control para la FPGA.

¹ También existe la posibilidad de configurar la FPGA directamente a través de un JTAG, mediante un cable adicional (byteBlaster).

² En formato *.pdf

³ Disponible en web (www.sec.upm.es)

- Cable RJ11- RJ11 cruzado de 1 metro longitud.
- Adaptador DB9 hembra a RJ11 hembra.

4.1.3 Requisitos del sistema

- IBM PC Pentium 90, mayor o compatible
- Lector CD.
- Mínimo 8 MB de RAM
- Mínimo 120MB de espacio libre en disco duro.
- Ms Windows 95 o compatible.
- Puerto de comunicaciones RS232 libre.

4.1.4 Instalación del hardware

Para instalar el hardware necesario en su sistema, siga los siguientes pasos:

1. Apague su ordenador y libere uno de sus puertos de comunicaciones serie¹. Si el puerto que queda disponible es de 25 pines, es necesario el empleo de un adaptador DB25 a DB9 no incluido en el paquete.
2. Conecte el adaptador DB9/RJ11 que le ha sido suministrado en el conector disponible.

Ya está en disposición de encender su ordenador cuando desee y comenzar a trabajar.

4.1.5 Instalación del software

Su paquete viene con un completo software de utilidades para su instalación.

- Herramienta de desarrollo MPLAB
- Herramienta de desarrollo MAX+PLUS II
- Programa de comunicaciones para trabajar con el sistema.
- Librerías y ejemplos para diseñar programas de control para la FPGA.

4.1.5.1 Instalación MAX+PLUS II

La versión suministrada es gratuita proporcionada por el fabricante cuya única limitación es el número de puertas sintetizable. En el caso de este sistema no afecta de ningún modo.

Para su instalación ejecute el programa contenido en la carpeta denominada MAX+PLUS II del CD con el nombre **'student923.exe'** y siga las instrucciones.

Por otro lado necesita obtener una licencia vía WEB para obtener todas las funcionalidades del paquete, esta se realiza de forma gratuita. Debe conectarse a www.altera.com y buscar licencia para programa MAX+PLUS II 9.23 Baseline y siga las instrucciones.

¹ Solo valido COM1 o COM2

4.1.5.2 Instalación Software de comunicaciones

Este software le permitirá cargar y depurar los programas creados para su sistema. Está realizado para sistema operativo MS-DOS, por lo que deberá ejecutar el archivo comunicaciones.exe que lo instalará en el directorio de designe.

4.1.5.3 Instalación MPLAB

Ejecute el archivo “Mp5x000.EXE” contenido en el directorio MPLAB y siga instrucciones.

4.2 Tutorial

Este apartado se encuentra contenido en CD, pues se trata de un hipertexto. Consiste en varios ejemplos de programas tanto para la FPGA como el μ C, mostrando paso a paso los procesos para la creación, carga y depuración de los mismos.

4.3 Manual de conexionado de las tarjetas TC-FPGA y TSA-FPGA

Las tarjetas son el elemento físico del paquete. Su uso es tan sencillo como conectarlas, alimentarlas, colocar sensores y actuadores, y descargar un programa de aplicación.

4.3.1 Interconectando las tarjetas.

El tipo de conectores de estas placas hacen innecesario el uso de separadores, así como de cables. Para interconectarlas debe hacer coincidir los conectores *CON1* y *CON2*¹ de ambas tarjetas, que se encuentran serigrafiados en los bordes. Además puede verificar la conexión gracias a que los conectores están desplazados respecto al centro. Por ello si las placas son colocadas al contrario, ambas placas no coincidirán de forma exacta.

Verifique el correcto posicionado de las placas antes de alimentarlas².

4.3.2 Alimentación.

Para alimentar las tarjetas precisa una batería o un alimentador cuya salida sea entre 11 y 15 voltios. La tarjeta TC-FPGA tiene un consumo aproximado de 30mA con alimentación entre 11 y 15V. Mientras que el consumo de la tarjeta TSA-FPGA varía según los sensores y actuadores conectados.

La conexión de alimentación se realiza mediante el conector nombrado batería³ en cualquiera de las tarjetas, respetando la polaridad serigrafiada en la superficie de la misma. Hay que tener en cuenta al interconectar las tarjetas que solamente

¹ Ver fig. 4.3 y 4.4

² De todos modos el diseño está implementado para que no se produzcan cortocircuitos aun cuando se conecten de forma errónea.

³ Véase fig 4.3 y fig 4.4, con la referencia Alimentación.

se conectará a la batería o fuente de alimentación una de ellas, pues *CON1* y *CON2* distribuyen la alimentación de una placa a otra.

4.3.3 Cable de comunicaciones.

Permite la conexión del sistema a un PC, gracias a esta conexión podremos realizar la descarga de aplicaciones, depuración de programas y cualquier uso alternativo que se desee.

El cable es de tipo telefónico, con conectores RJ-11 en los extremos, y debe estar montado cruzado, como puede verse en la Fig. 4.1.



Fig 4.1: Montaje del cable RJ-11

Ya que el conector del PC no es un RJ-11, si no que un DB9, se usa un adaptador RJ-11/DB9 que debe montarse como se indica en la figura 4.2.

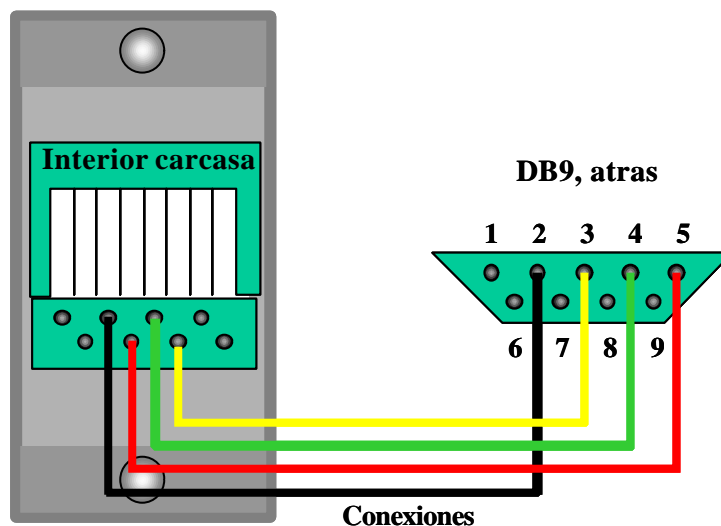


Fig 4.2: Montaje del adaptador RJ11/DB9

Los colores de los cables de la figura 4.2 son orientativos, ya que algunos modelos tienen colores distintos. Asegúrese de que las conexiones son correctas e introduzca los conectores hasta el fondo empujándolos con un destornillador u objeto punzante, y después cierre la carcasa. Si se interconecta el cable RJ11/RJ11 con el adaptador RJ11/DB9, se obtiene el cable listo para ser utilizado.

4.3.4 Las conexiones.

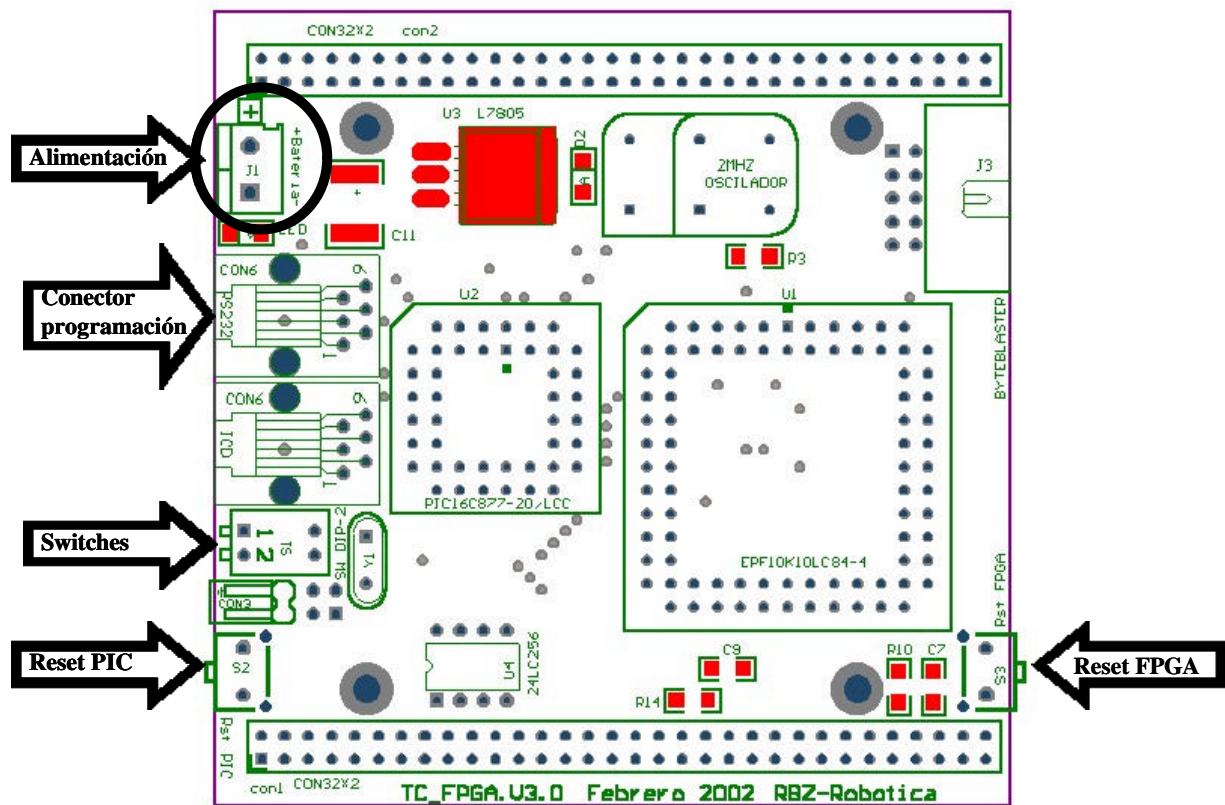


Fig 4.3: Serigrafiado de componentes de la placa TC-FPGA

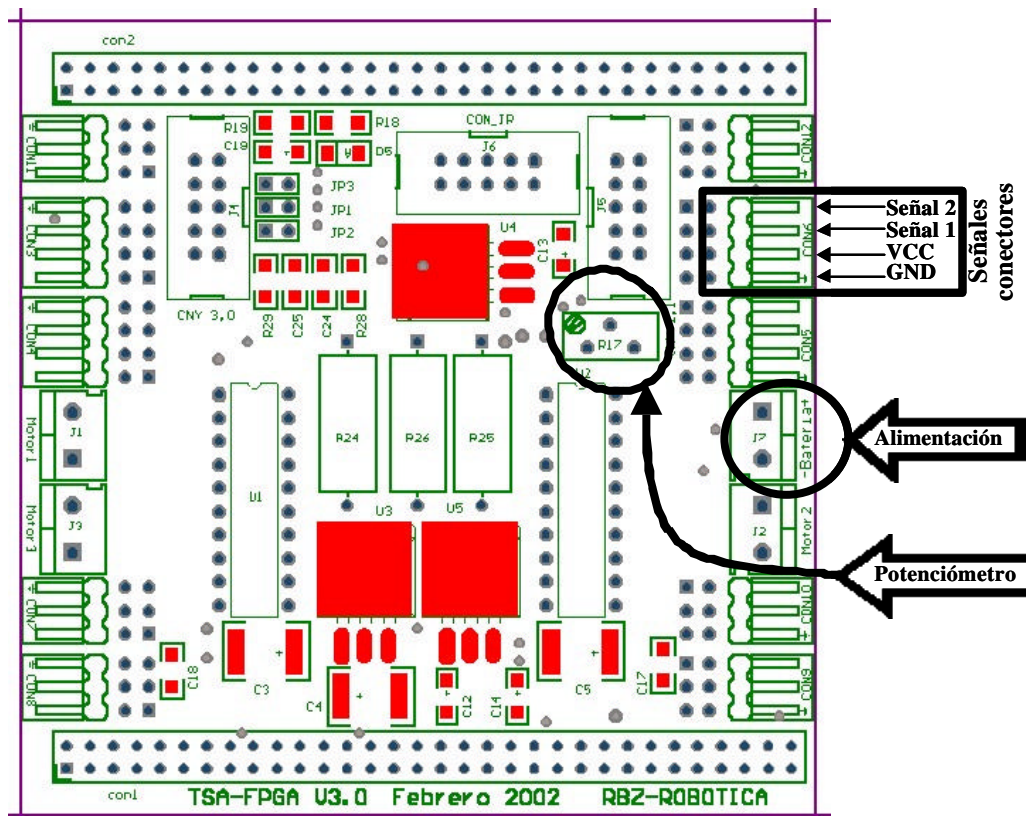


Fig 4.4: Serigrafiado de componentes de la placa TSA_FPGA

Conectores de 2 señales

Están compuestos de 2 señales de Entrada / Salida, alimentación y masa, las líneas se encuentran dispuestas en horizontal. Tal y como muestra la figura 4.5 los conectores se encuentran apilados 2 en 2, de modo que un conector está en la parte superior, al que se nombrará A, y otro en la parte inferior, al que se nombrará B.

En la serigrafía de la placa se puede ver el número correspondiente de conector, así como la referencia de colocación del pin de masa pudiendo ver un ejemplo de su pinout en la figura.4.5.

La alimentación será de 5V, referencia de masa y 2 señales conectadas a la FPGA. Este tipo de conector se utilizará principalmente para conectar sensores inteligentes en los cuales el trasiego de información se realiza en los 2 sentidos, sin embargo su uso puede ser distinto.

Los conectores de este tipo son los siguientes:
CON_3,
CON_4,
CON_5 y
CON_6.

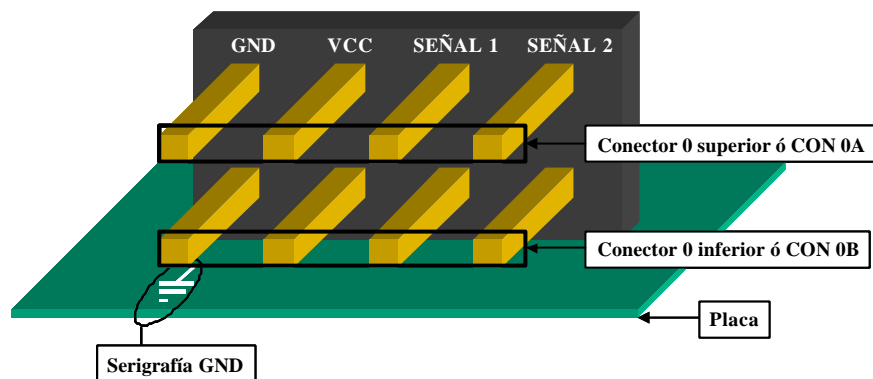


Fig 4.5: Detalle de conectores de 2 señales.

Conectores de 1 señal

La disposición de estos conectores es similar a la de los anteriores pero con una sola línea de E/S. Se puede ver un ejemplo de su pinout en la figura.4.6.

El pin correspondiente a la señal tiene incorporada una resistencia de pull-up de 39K conectada a 5V, con el fin de facilitar la conexión a interruptores y pulsadores.

La alimentación de estos conectores puede ser de 5V o 6V, dependiendo del chip soldado como U5, siendo 5V si es un 7805 y 6V si es un 7806. En caso que U5 fuese un 7806, se debe tener precaución al conectar elementos externos para que la señal de E/S no reciba 6V, pues se podría dañar la FPGA.

El regulador 7806 es utilizado para conectar servomecanismos de radiocontrol, a fin de usar la potencia máxima de estos.

Los conectores de este tipo son los siguientes:
CON_7,
CON_8,
CON_9 y
CON_10.

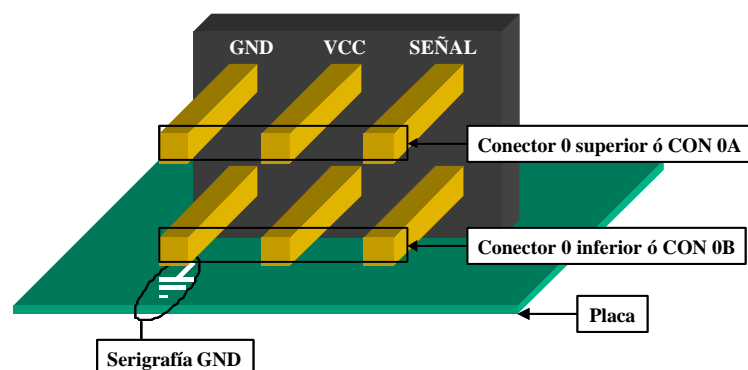


Fig 4.6: Detalle de conectores de 1 señal.

Conectores ADC

Tienen la misma forma física que los conectores de 1 señal. Están ideados para conectar sensores en los cuales la información es analógica, con tensiones entre 0 y 5V. Estos se encuentran conectados a canales ADC correspondientes al μC . La alimentación de estos conectores es de 5V.

A la hora de utilizar estos conectores hay que tener en cuenta que los ADCs se comparten como recurso con las resistencias de sensado¹ de corriente en los motores, de forma que si está conectado el JUMPER correspondiente², no debe usar el conector correspondiente para su correcto funcionamiento.

Los conectores de este tipo son los siguientes:
CON_11 y CON_12.

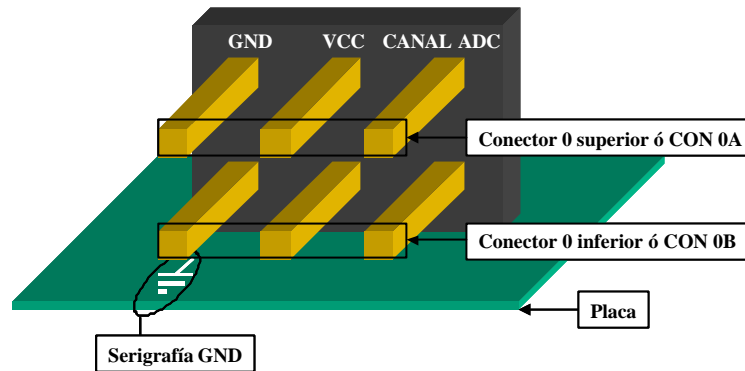


Fig 4.7: Detalle de conectores ADCs.

Conectores para CNY_70

Los conectores CNY 3,0 y CNY 2,1 utiliza un comparador ajustable mediante el potenciómetro³ de la placa para modificar el umbral del 0 y el 1 lógico que le envía a una línea de entrada en la FPGA.

El conector utilizado es de tipo bus de 10 hilos. En la siguiente figura se puede ver el patillaje del conector. Para más información sobre la conexión correspondiente a cada pin, ver tabla de descripción de conectores.

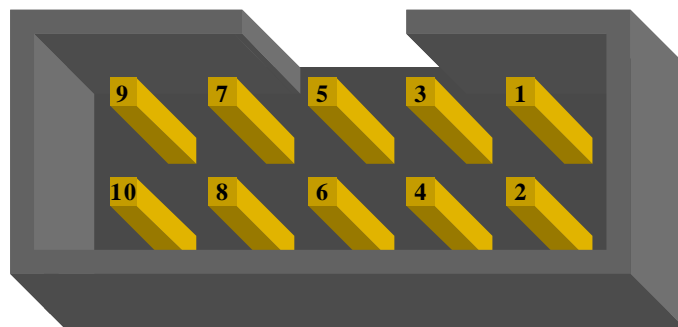


Fig 4.8: Conector para cable de bus.

¹ Vease apartado 4.3.6.1

² JP1 → ADC de CON12_A, JP2 → ADC de CON12_B, JP3 → ADC de CON11_A

³ Vease fig 4.4, referencia potenciómetro.

Conector de mas señales

Denominado CON_IR, es de tipo bus de 10 líneas, de las cuales 7 son señales de entrada / salida de la FPGA, 1 corresponde a la batería y los 2 restantes son masa.

Conectores de motores

La tarjeta puede controlar un total de 3 motores de corriente continua. La electrónica que lleva incorporada la tarjeta se controla por medio de 3 pines de la FPGA, siendo IN1_M? y IN2_M? para la selección de dirección (adelante, atrás, cortocircuito) y EN_M? el pin de habilitación que se usará para modificar la velocidad por medio de PWM. El voltaje de alimentación de los motores corresponde con la batería. Además, la electrónica permite medir mediante los ADCs del micro, la corriente que consume en un momento determinado cada motor utilizando lo que se ha denominado “resistencias de sensado”¹.

Cada uno de los motores que se puede controlar, se conecta mediante clemas, tal y como está indicado en la serigrafía de la tarjeta.

Tabla descripción conexiones TSA_FPGA

Conector	Pin	Líneas	Pines-FPGA	Tipo
CON3_A	Señal 1	FPGA_I/O_39	83	Entrada / Salida
	Señal 2	FPGA_I/O_2	7	Entrada / Salida
CON3_B	Señal 1	FPGA_I/O_0	5	Entrada / Salida
	Señal 2	FPGA_I/O_5	10	Entrada / Salida
CON4_A	Señal 1	FPGA_I/O_6	11	Entrada / Salida
	Señal 2	FPGA_I/O_4	9	Entrada / Salida
CON4_B	Señal 1	FPGA_I/O_3	8	Entrada / Salida
	Señal 2	FPGA_I/O_1	6	Entrada / Salida
CON5_A	Señal 1	FPGA_I/O_20	54	Entrada / Salida
	Señal 2	FPGA_I/O_21	58	Entrada / Salida
CON5_B	Señal 1	FPGA_I/O_22	59	Entrada / Salida
	Señal 2	FPGA_I/O_27	65	Entrada / Salida
CON6_A	Señal 1	FPGA_I/O_29	67	Entrada / Salida
	Señal 2	FPGA_I/O_31	70	Entrada / Salida
CON6_B	Señal 1	FPGA_I/O_28	66	Entrada / Salida
	Señal 2	FPGA_I/O_30	69	Entrada / Salida
CON7_A	Señal	FPGA_I/O_10	37	Entrada / Salida
CON7_B	Señal	FPGA_I/O_9	36	Entrada / Salida
CON8_A	Señal	FPGA_I/O_8	35	Entrada / Salida
CON8_B	Señal	FPGA_I/O_7	30	Entrada / Salida
CON9_A	Señal	FPGA_I/O_23	60	Entrada / Salida
CON9_B	Señal	FPGA_I/O_24	61	Entrada / Salida
CON10_A	Señal	FPGA_I/O_25	62	Entrada / Salida
CON10_B	Señal	FPGA_I/O_26	64	Entrada / Salida
CON11_A	Canal ADC	AN3	ADC 3 del PIC	Entrada analógica
CON11_B	Canal ADC	AN4	ADC 4 del PIC	Entrada analógica
CON12_A	Canal ADC	AN0	ADC 0 del PIC	Entrada analógica
CON12_B	Canal ADC	AN1	ADC 1 del PIC	Entrada analógica
CNY 2,1	1,4,7,10	GND	-	Entrada por medio de comparador
	5,6	N.C.	-	
	3,9	R-Pull up 220	-	
	2	FPGA_IN_2	44	
	8	FPGA_IN_1	42	
CNY 3,0	1,4,7,10	GND	-	Entrada por medio de comparador
	5,6	N.C.	-	

¹ Vease apartado 4.3.6.1

	3,9	R-Pull up 220	-	
	2	FPGA_IN_3	84	
	8	FPGA_IN_0	2	
CON_IR	1,10	GND	-	GND
	2	FPGA_I/O_32	71	Entrada / Salida
	3	FPGA_I/O_34	73	Entrada / Salida
	4	FPGA_I/O_33	72	Entrada / Salida
	5	FPGA_I/O_35	78	Entrada / Salida
	6	FPGA_I/O_36	79	Entrada / Salida
	7	FPGA_I/O_37	80	Entrada / Salida
	8	FPGA_I/O_38	81	Entrada / Salida
	9	BATERIA	-	
-		AN2	ADC 2 del PIC	Entrada analógica
Motor 1		EN_M1	39	Salida
		IN1_M1	38	Salida
		IN2_M1	47	Salida
Motor 2		EN_M2	51	Salida
		IN1_M2	49	Salida
		IN2_M2	50	Salida
Motor 3		EN_M3	53	Salida
		IN1_M3	48	Salida
		IN2_M3	52	Salida
JP1		AN0	ADC 0 del PIC	Conecta resistencia sensado motor 3
JP2		AN1	ADC 1 del PIC	Conecta resistencia sensado motor 2
JP4		AN3	ADC 3 del PIC	Conecta resistencia sensado motor 1

Los conectores: Parte B es abajo, y parte A arriba.

Bus de comunicaciones FPGA-PIC

Líneas	Pines-FPGA	Tipo
RD	28	Entrada
WR	27	Entrada
DIR/DATO	21	Entrada
CS	29	Entrada
AD0	25	Entrada / salida
AD1	24	Entrada / salida
AD2	23	Entrada / salida
AD3	22	Entrada / salida
AD4	19	Entrada / salida
AD5	18	Entrada / salida
AD6	17	Entrada / salida
AD7	16	Entrada / salida

Otras líneas importantes.

Líneas	Pines-FPGA	Tipo
CLK_0	1	Reloj
Reset_FPGA	3	Reset

4.3.5 Ejemplos de conexionado a TSA_FPGA

Sensores tipo fotorelectores (CNY70)

Se conectan como muestra las Fig. 4.9 y 4.10, utilizando los conectores denominados CNY2,1 y CNY3,0.

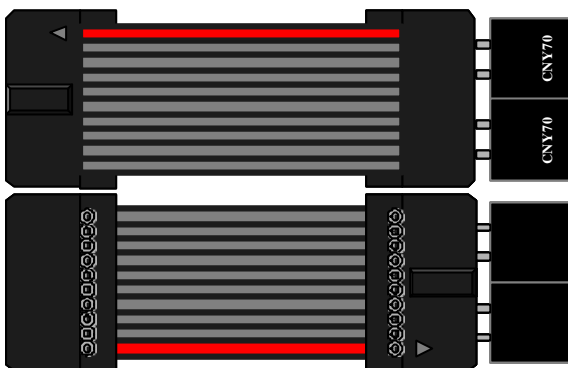


Fig 4.9 Conexión de sensores CNY_70

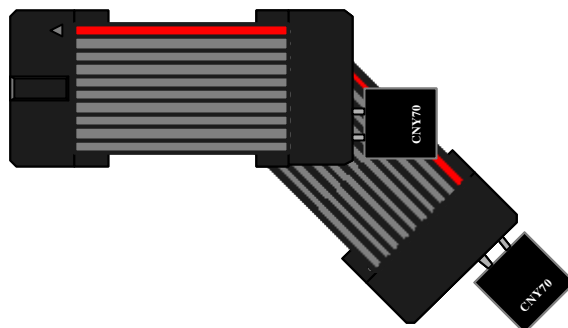


Fig 4.10 Otra forma de conectar sensores CNY_70

LED

Puede verse ejemplo de modo de conectar en la Fig. 5.11, para conectores con 1 línea de señal. Se encenderá con “1” lógico.



Fig 4.11 Conexión de un L.E.D.

BUMPER

Se utilizan los conectores de 1 señal, y aprovechando R de pull-up interna, se leerá 0 cuando pulsado y 1 en caso contrario.

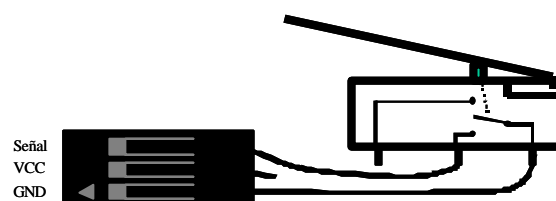


Fig 4.12 Conexión de un bumper.

Detectores de objetos por IR: GP2D02 y GP2D12

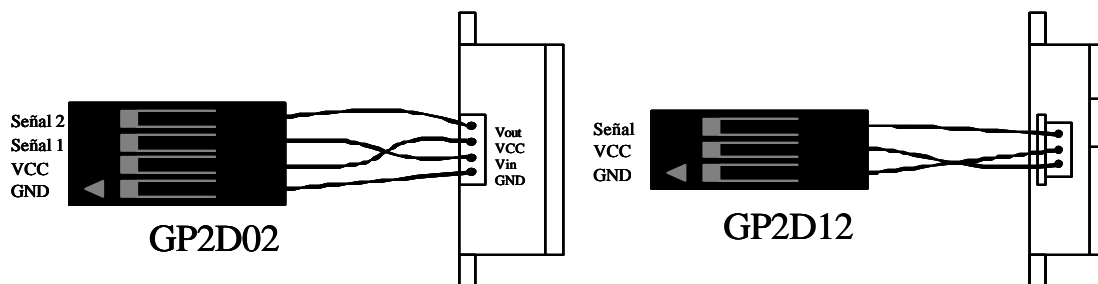


Fig 4.13. Como conectar sensores GP2D02 y GP2D12

El GP2D12 ofrece una salida con valor analógico, por tanto se conecta en los conectores de los ADCs. Para leer los valores desde la FPGA se precisa utilizar el módulo de comunicaciones con el PIC.

El GP2D02, por el número de líneas a utilizar, se conecta en los conectores de 2 líneas de señal, usando el módulo diseñado específicamente para controlarlo y realizar la lectura.

Sonar de ultrasonidos modelo SRF04

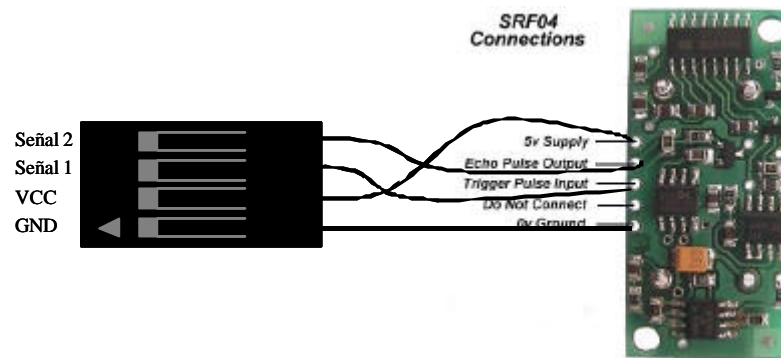


Fig 4.14. Conectando sonar de ultrasonidos SRF04.

Al igual que el GP2D02, se conecta en los conectores de 2 señales, y para leer datos se utilizará módulo diseñado específicamente para controlarlo y realizar la lectura de los datos ofrecidos.

4.3.6 Sensores incorporados en la tarjeta TSA-FPGA

4.3.6.1 Resistencias de sensado

El objetivo medir la corriente que circula por el motor correspondiente para dar posibilidad al programador de:

- Realizar un control de torque controlando que la corriente por el motor sea constante.
- Detectar bloqueo del motor por consumo mayor o que le falta carga por un consumo menor.
- Además puede usarse para controlar los motores cuyas características de consumo medio o de pico no fuese soportado por el puente en H. Si el consumo llega a un umbral, se reduce la tensión de alimentación del motor (en este caso el PWM) para que consuma menos. Se trata en definitiva un uso más suavizado de los motores.

Para realizar la medición de corriente a través del motor se ha colocado una resistencia de sensado de 0.39Ω , y la tensión en esta es tomada mediante el ADC. Sin embargo, puesto que los motores se controlan en velocidad mediante PWM, la corriente que circule por el motor tendrá la misma forma de onda que el PWM. Por este motivo se ha colocado un filtro paso bajo pasivo de primer orden, tal y como muestra la figura 4.15. En definitiva lo que se medirá es la corriente media¹ que circula por el motor. A continuación se muestra el cálculo de los elementos del filtro.

Este tiene relación con la frecuencia del PWM que se aplica a los motores, la cual se tomará de 7 KHz, ya que esta es la frecuencia de los módulos software para control de motores implementa. Puesto que se desea medir la corriente media, se precisa de un filtro paso bajo de una frecuencia de corte cercana a 0Hz, ya que de este modo se obtendría la señal continua pura. Sin embargo tampoco se debe escoger una frecuencia demasiado baja, ya que la estabilización

¹ Si lo que se quiere medir son los picos de corriente el filtro no será necesario.

de la señal sería demasiado lenta. Se ha elegido una frecuencia de corte de un orden 100 veces inferior a la frecuencia de modulación. De este modo:

$$f_c = \frac{1}{2\pi RC} \Rightarrow RC = \frac{1}{2\pi f_c} = \frac{1}{2\pi 70\text{Hz}} = 2.27 \cdot 10^{-3}$$

Si se toma un condensador de 220nF se obtiene:

$$C = 220\text{nF} = 10K$$

Con estos valores se consigue una constante de tiempo para el filtro de:

$$t = RC = 2.27\text{ms} \Rightarrow 5 \cdot t = 11.35\text{ms}$$

Lo que quiere decir que pasarán unos 11ms desde que cambia la corriente a otro valor hasta que se lee el valor verdadero. Por tanto no se puede provocar cambios en el motor a razón de 90Hz o más, pero mecánicamente el motor no tiene capacidad suficiente para soportar estos cambios, y por tanto nunca se dará este caso.

Para calcular la corriente que pasa por el motor tan solo hay que aplicar la ley de ohm al voltaje leído del ADC.

Ejemplo para voltaje de 0,7V: $I_{MOTOR} = \frac{V_{SENSE}}{R_{SENSE}} = \frac{0,7V}{0,39} = 1,79\text{A}$

El voltaje que se puede leer en V_{SENSE} varía entre 0 y 0,88V por la resistencia de sensado colocada ($0,39\Omega$) y límite de corriente físico sistema de control del motor (2,26 A). Se ha diseñado de esta forma porque si la resistencia de sensado fuese mayor, la pérdida de energía en esta sería mayor quitándosela al motor.

Por otro lado ha de tenerse en cuenta la cuantificación realizada por el ADC del PIC, es respecto a sus 5V, si el valor de entrada no superará 1V. Si se cuantifica con 8 bit, solo serán efectivos 6 bit.

Cabe destacar que el valor de la corriente que se obtendrá no será exactamente el verdadero, ya que el filtro proporciona el valor medio e introduce una pequeña atenuación (por no ser la impedancia de entrada del ADC infinita).

Por último es importante recordar que para utilizar estos sensores, se ha de conectar el **JUMPER** correspondiente¹ y no colocar nada en el conector del ADC respectivo para evitar medidas erróneas.

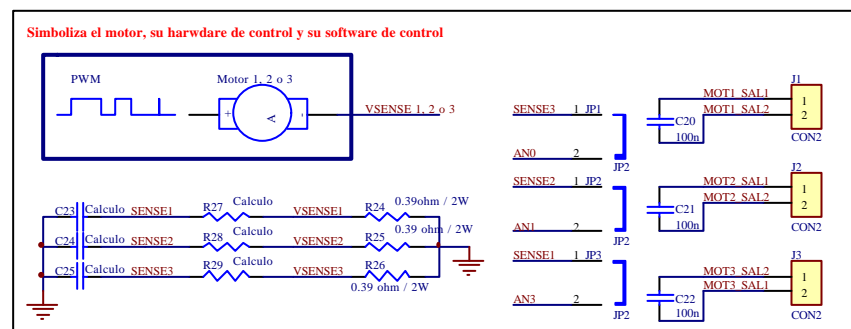


Fig 4.15: Esquema del sensado de corriente por los motores.

¹ Vease fig 4.15.

4.3.6.2 Nivel de batería

El sensor tiene como objetivo comprobar el estado de carga de la batería. Con esta información, se puede programar para que en un momento determinado avise que le queda poca carga o incluso que funcione en algún modo de bajo consumo como desconectar algún sensor o mover los motores más despacio.

Se pretende medir la tensión de la batería ya que es proporcional a la carga energética pero:

- El sistema introduce ruidos en el voltaje de alimentación.
- La tensión de batería depende también del consumo debido a su resistencia interna.

El montaje decidido fue un detector de pico con un filtro para medir mediante un ADC el voltaje más aproximado al voltaje de batería sin carga.

$$f_c = \frac{1}{2\pi RC} = 7,3Hz$$

Con estos valores se consigue una constante de tiempo para el filtro de:

$$t = RC = 21,8ms \Rightarrow 5 \cdot t = 109ms$$

La forma de medir será por medio de un conversor ADC del PIC, que para ello se han colocado un divisor resistivo, pues el ADC tiene un rango de 0 a 5V, y en cualquier caso la batería será de 7 voltios o más.

Según la figura del sensor (2.4.2), la relación del voltaje a cuantificar (AN2) con la de batería será la siguiente:

$$V_{AN2} = \frac{(V_{BATERIA} - V_{D3}) \cdot R_{19}}{R_{18} + R_{19}} = 0,312 \cdot (V_{BATERIA} - 0,7)$$

Por otro lado el límite de voltaje de batería para no dañar el ADC del PIC será el siguiente:

$$V_{BATERIA MAX} = \frac{V_{AN2 MAX}}{0,312} + V_{D3} = 16,72V$$

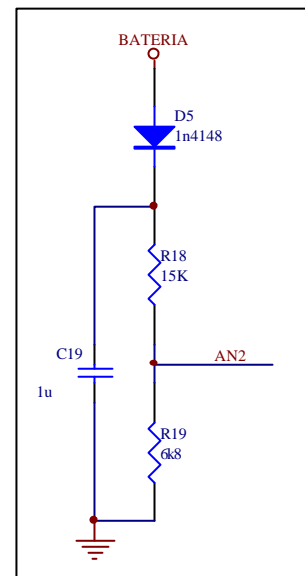


Fig 4.16 Esquema de sensor nivel de batería

Para determinar el estado de la carga de la batería, hay que tener en cuenta tanto su tensión nominal como su tecnología de fabricación.

Para las baterías elegidas en el proyecto, las cuales son de NI-MH cuya tensión nominal son 7,2V. Se considera cargada para tensión de un 15% mayor, y descargada al revés 15% menor.

- Tensión nominal batería = 7,2V → Valor cuantificado ideal =104d
- Batería cargada = 8,28V → Valor cuantificado ideal =122d
- Batería descargada = 6,12 V → Valor cuantificado ideal =087d

Una vez determinado los umbrales a partir de los cuales se tomará la decisión del estado de batería, se recomienda realizar una calibración con el sistema debido errores introducidos, por la diferencia del voltaje del diodo, impedancia del ADC finita, tolerancia de resistencias, etc.

4.4 Programación de la TC-FPGA

Se puede realizar la programación tanto de la FPGA como del μ C de la tarjeta mediante uso del puerto serie de su PC. Además se podrá realizar una depuración en tiempo real de programas de aplicación de la FPGA.

El sistema permite almacenar 2 configuraciones distintas para la FPGA y una aplicación para el μ C. El modo en que se gestionan las 2 configuraciones de la FPGA es el siguiente:

- Existen 2 bancos para almacenamiento de las configuraciones de la FPGA, denominados banco de Pruebas y banco Definitivo.
- El banco Definitivo almacena una configuración ya probada y validada
- El banco de Pruebas es donde se cargan **siempre** las configuraciones procedentes del PC.
- Cuando validamos un programa del banco de Pruebas, este es almacenado en el banco Definitivo, mientras que el programa que era definitivo pasa a la zona de pruebas, y será borrado cuando una nueva configuración sea cargada.

Por otro lado tenemos 2 modos de arranque, normal y programación:

- **Modo Normal:** Al inicializarse el sistema se carga la configuración seleccionada por el switch 2 (fig. 4.17) en la FPGA, y posteriormente ejecuta el programa de aplicación cargado en el μ C, si es que lo hay.
- **Modo Programación:** Al arrancar el sistema no configura la FPGA, sino que ofrece unos servicios a través del puerto serie, que el usuario manejará mediante el programa TC_FPGA.EXE. Estos servicios permiten:
 - Descargar una aplicación para la FPGA (se carga en el banco de pruebas).
 - Descargar una aplicación para el μ C
 - Validar el programa del banco de Pruebas, pasando este al banco Definitivo.
 - Realizar una depuración de la aplicación. Según el estado del switch 2 (fig. 4.17), depuraremos el programa del banco de Pruebas o el Definitivo.

Para elegir uno de estos modos deben seleccionarse mediante el switch 1 (Fig. 4.17).

Modos de arranque

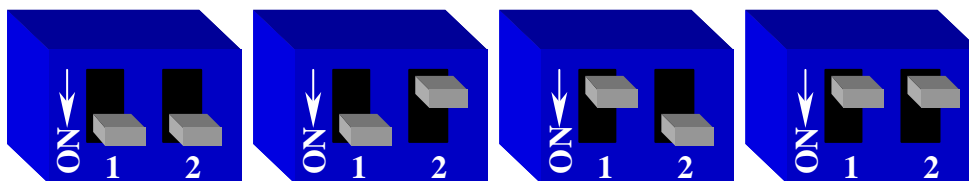


Fig 4.17.: Modos de arranque. De izquierda a derecha: 1 Modo programación y en caso de debug se carga en FPGA el programa de pruebas. 2. Modo programación y en caso de debug se carga en FPGA el programa definitivo. 3. Modo normal, configurando FPGA con programa en pruebas. 4. Modo normal, configurando FPGA con programa definitivo. definitivo.

4.4.1 Estableciendo conexión

Pasamos a explicar los pasos fundamentales para la conexión de la TC_FPGA con el PC para su programación.

1. Seleccione cualquiera de los modos de programación (1 o 2 Fig. 4.17) por medio de los switches.
2. Conecte el cable RJ-11 en el conector de la TC_FPGA con la serigrafía RS-232¹
3. Encienda el sistema.
4. Arranque el programa de comunicaciones TC_FPGA.EXE en su PC.
5. Seleccione el puerto en el haya conectado la placa.
6. Aparecerá el menú de opciones, en caso contrario asegurese de haber realizado los pasos anteriores correctamente, y que ningún programa ocupa el puerto de comunicaciones que esté usando.

4.4.2 Menú de programación

Una vez establecida la conexión aparecerá un menú que le permite llevar a cabo las distintas acciones comentadas en el punto 4.4 y que pasamos a detallar.

4.4.2.1 Cargar programa en PIC.

Mediante esta opción se pueden cargar programas de aplicación realizados para el μ C. Para ello le será requerido el nombre del programa sin su extensión, la cual debe ser .HEX² en todos los casos. Como se trata de un programa bajo MS-DOS será necesario marcar la ruta completa del archivo si este no se encuentra en el mismo directorio que el programa TC_FPGA.EXE.

Si se produce algún error durante la descarga del archivo, ó se interrumpen las comunicaciones, este no será grabado correctamente y por tanto no será ejecutado en el modo normal.

Para salir de esa opción sin descargar ningún programa pulse “Q”+ENTER.

4.4.2.2 Cargar programa en FPGA.

Permite la descarga de los archivos de configuración generados para la FPGA. Estos siempre serán cargados en el banco de Pruebas como ya se ha explicado. Le será requerido el nombre del archivo sin extensión, que será .RBF³. Este archivo no es generado automáticamente por el MAX PLUS II, por lo que debe referirse al tutorial de ejemplo del manual de usuario, donde se indica como generar este archivo.

Como en el caso anterior, debe indicar la ruta completa del archivo si este no se encuentra en el mismo directorio que el programa TC_FPGA.EXE.

Si se produce algún error durante la descarga del archivo, ó se interrumpen las comunicaciones, este no será grabado correctamente y por tanto no será ejecutado en el modo normal.

Para salir de esa opción sin descargar ningún programa pulse “Q”+ENTER.

¹ Vease fig 5.3, Conector programación.

² Los programas deben ser en formato INTEL 8 bits, cuya extensión siempre es .HEX. En el entorno MPLAB, los ficheros compilados tienen este formato por defecto.

³ Raw Binary File.

4.4.2.3 Cambio de Banco

Esta opción le permitirá validar el programa del banco de Pruebas, pasando automáticamente al banco Definitivo. Esta opción no se llevará a cabo si no existe una configuración cargada correctamente en el banco de Pruebas.

4.4.2.4 Depurar un programa

Le permitirá observar y modificar hasta 8 registros de la FPGA en tiempo real, interactuado directamente sobre el hardware. Antes de ejecutar este modo debe tener en cuenta las siguientes restricciones:

- **Perdida del programa de aplicación del mC**

Para realizar esta depuración es necesario tener cargado un programa especial sobre la zona de programas de aplicación del μ C. Este programa será descargado automáticamente por el programa TC_FPGA.EXE, siempre y cuando no este cargado. Por tanto, si cargó un programa sobre el μ C lo perderá, siendo necesario cargarlo de nuevo al finalizar la depuración, mediante la opción del menú.

- **Inclusión del módulo de comunicaciones en el programa de la FPGA**

Para ejecutar la depuración debe incluir en el programa de la FPGA el módulo “Comunicaciones_PIC” (apartado 4.5.1.7), sobre el que se asignarán las señales que se desee observar y modificar. Para más información, referirse al tutorial del manual de usuario.

Una vez seleccionado el modo depuración, debe tener en cuenta que será depurado el programa seleccionado por el switch 2 (Fig. 4.17).

En la pantalla aparecerán dos ventanas en las que se pueden observar los registros de recepción y los de transmisión. Siendo los de recepción los que proceden de la FPGA, y los de transmisión los que modificamos en la FPGA.

Para modificar los registro utilice los cursores, ARRIBA y ABAJO para moverse entre los diferentes registros, y DERECHA e IZQUIERDA para incrementarlos o decrementarlos respectivamente. También puede introducir un valor mediante el teclado numérico. Como los registros del depurador son de 8 bits, solo se admiten valores entre 0 y 255.

Para abandonar la depuración y regresar a las opciones anteriores pulse la tecla ESCAPE.

4.5 Módulos de control para la FPGA.

A continuación se describe una serie de módulos de control que han sido creados para facilitar el manejo y control de algunos sensores y actuadores que se pueden conectar a las tarjetas TC_FPGA y TSA_FPGA, así como para la generación de programas sencillos de control.

4.5.1 Reloj_sistema

Este módulo ha sido creado para proporcionar a cada uno de los elementos del sistema de una señal de reloj de frecuencia adecuada, además de los datos del TIMER para el PWM. En principio no es necesario, pues cada módulo puede tomar directamente la señal de reloj del sistema y acondicionarla según necesidades, pero por motivos de ocupación de la FPGA resulta necesario.

A la entrada se conecta el reloj de la FPGA, correspondiente al oscilador de 2MHz, ofreciendo a la salida las siguientes señales:

- PWM [n-1..0] → Son los datos del timer para el módulo que genera el PWM¹ para el control del motor. Al insertar el componente se configurará el parámetro “n” para seleccionar el número de bits del timer.
- Reloj_sonar → Reloj para el módulo de control del sonar SRF04 de frecuencia 15625Hz
- Reloj_GP2 → Reloj para el módulo de control del GP2D02 de frecuencia 7812,5 Hz.
- Reloj_38Khz. → Periodo = 26,315µs
- Reloj_10Khz → Periodo = 100us
- Reloj_1Khz → Periodo = 1ms.
- Reloj_100Hz → Periodo = 10ms.
- Reloj_10Hz → Periodo = 100ms.

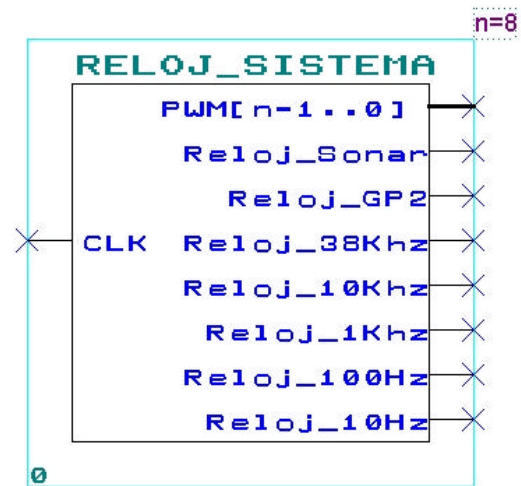


Fig 4.18: Símbolo del módulo reloj_sistema.

4.5.2 Sonar.

El siguiente módulo se utiliza para controlar el Sonar SRF04. La descripción de las señales es la siguiente:

- CLK → Su señal de reloj, que se conectará al pin “Reloj_sonar” del módulo Reloj de sistema.
- RST → Reset del módulo, a nivel bajo.
- DATOS [7..0] → Es un bus de 8 hilos que utiliza para devolver en paralelo la lectura realizada del sonar. Devolverá datos de 0 a 254,



Fig 4.19: Símbolo del módulo control del sonar.

¹ Modulación por ancho de pulso.

correspondiente a la distancia en cm que detecta el sensor. Si el dato es 255 significa que reset está activo o el sensor no funciona o no ha sido conectado.

- TRIGUER → Según se ha recomendado conectar este sensor, será la Señal1 del conector correspondiente.
- ECHO→ Según se ha recomendado conectar este sensor, será la Señal2 del conector correspondiente.

4.5.3 GP2D02

El módulo tiene las siguientes señales

- CLK → Su señal de reloj, que se conectará al pin “Reloj_GP2” del módulo Reloj de sistema.
- RST → Reset del módulo, a nivel bajo.
- DATOS [7..0] → Es un bus de 8 hilos que utiliza para devolver en paralelo la lectura realizada del sonar. Ofrece datos según gráficas de características del sensor. Si el dato es 0 significa que reset está activo o el sensor no funciona o no ha sido conectado.
- TRIG → Según se ha recomendado conectar este sensor, será la Señal1 del conector correspondiente. Es el pin denominado en el sensor como Vout.
- Data→ Según se ha recomendado conectar este sensor, será la Señal2 del conector correspondiente. Es el pin denominado en el sensor como Vin.



Fig 4.20: Símbolo del módulo control del GP2D02

4.5.4 MOTOR.

El módulo de control del motor, se encarga de generar la señal de PWM según los datos introducidos como referencia, además de activar el motor en el sentido de giro indicado.

Al introducir el símbolo, hay que configurar el parámetro “n” correspondiente al n° de bits para el PWM.

Las señales son las siguientes:

- RST→ Reset del modulo, que para el motor si está activo (nivel bajo).
- Timer[n-1..0]→ Contador para generar el PWM, que se conectará al módulo Reloj de sistema con la salida denominada PWM. El número de bits se configura mediante el parámetro “n”.
- Motor[n+1..0]→ Es un bus de “n” líneas mas 2, de las cuales las 2 de posición más alta indican la dirección del motor y el resto la referencia del PWM. Según las líneas n+1 y n, la dirección para el motor es: “00” atrás, “01” parada rápida, “10” parada rápida, “11” adelante.

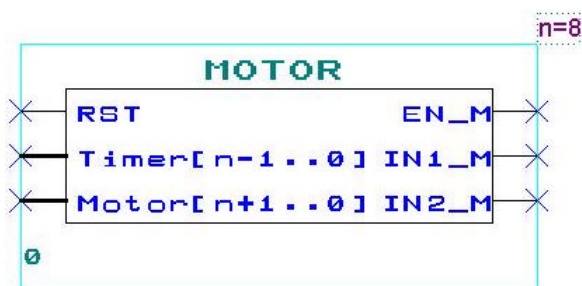


Fig .4.21: Símbolo del módulo control de motores.

- EN_M, IN1_M, IN2_M → Líneas de salida para controlar la electrónica del motor, que se conectará según el motor a controlar en los pines de la FPGA descritos en la tabla de conexiones.

4.5.5 Monoestable.

El siguiente módulo ha sido creado para realizar temporizaciones en el diseño deseado.

Al introducirlo, se configura el parámetro “**ciclos**” correspondiente al número de ciclos de reloj de durará disparado desde que la señal *DISPARO* vuelva a su estado normal. La señal *ACTIVO* se pondrá a “1” lógico cuando la señal *DISPARO* pase a “0” lógico, volviendo *ACTIVO* a “0” lógico pasados los **ciclos** de reloj configurados después de volver *DISPARO* a “1” lógico. Para una mejor comprensión, vea la fig 5.xx que muestra una simulación del módulo con un reloj de periodo 200ns y 10 **ciclos** que durará disparado.



Fig 4.22: Símbolo del módulo monoestable.

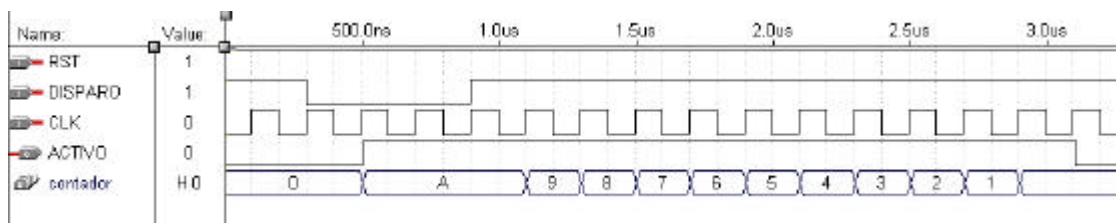


Fig 4.xx: Diagrama de tiempos del módulo monoestable.

4.5.6 Comparador

Activa la señal *Mayor*, *Menor* o *Igual* a “1” lógico en caso de producir lo correspondiente.

Como parámetros configurables, “**n**” es el ancho del bus de la entrada, y “**nivel**” es el dato fijo con que se realiza la comparación de los datos para activar la señal correspondiente de salida.

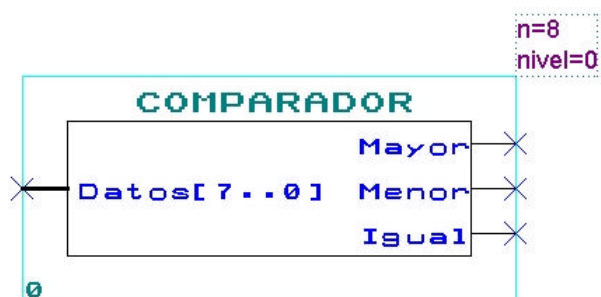


Fig 4.23: Símbolo del módulo comparador.

4.5.7 Módulo comunicaciones con PIC.

Permite comunicar la FPGA con el μ C de forma transparente al usuario. Se crean unos registros de lectura y otros de escritura a través de los cuales se intercambian información ambos dispositivos. Es decir, los registros que puede leer la FPGA son escritos por el μ C y viceversa. Además existen 5 registro a través de los cuales el μ C transfiere el valor de los conversores AD a la FPGA periódicamente, la frecuencia de muestreo de estos puede ser variada mediante un programa de aplicación sobre el μ C. Es necesario el uso de este módulo para realizar la depuración de programas, realizando la conexión de las señales que queramos visualizar y modificar.

El significado de los parámetros de configuración es:

- **adX_on:** 1 si usamos el canal X de los ADCs. 0 en caso contrario.
- **Registros_recepcion:** Define el número de registros de 8 bits, que recibirán datos desde el μ C. Pueden tomar valores de 0 a 8.
- **Registros_transmisión:** Define el número de registros de 8 bits, que transmitirán datos al μ C. Pueden tomar valores de 0 a 8.

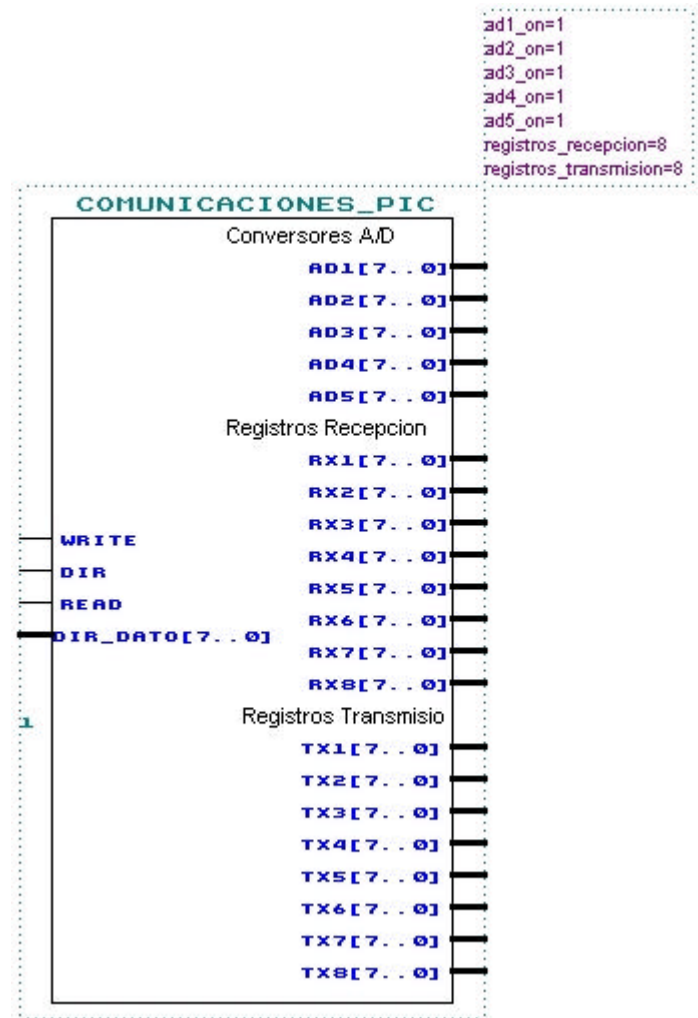


Fig 4.24: Símbolo del módulo de comunicaciones con el μ C.

Por otro lado las señales son:

- **WRITE, DIR, READ y DIR_DAT0:** Señales que forman el canal de transmisión con el μ C. Para la conexión a los pines de la FPGA referirse a la *Tabla de Conexiones*.
- **AD1 a AD5:** contienen el valor de los conversores AD 1 al 5, que son actualizados periódicamente.
- **RX1 a RX8:** Contienen los datos recibidos desde el μ C. En el caso del modo depuración, contendrán los datos que el usuario imponga desde el PC. En los programas de aplicación para el μ C podrán tener la funcionalidad que se desee.

- **TX1 a TX8:** Los datos escritos en estos registros serán enviados al μC . Durante la depuración contendrán las señales que el usuario desee visualizar en la pantalla del PC. En los programas de aplicación del μC podrán tener la funcionalidad que se desee.

Para más información sobre el funcionamiento de este módulo y de las funciones para el microcontrolador que implementan las comunicaciones, referirse al ANEXO A.

4.5.8 Nodo supresor de conductas.

Se parametriza el ancho de bus “ n ”, los bits $n-1$ a 0 proporcionan la señal de control de los actuadores. Mientras que el bit n indica si:

- 1 \rightarrow Conducta “ACTIVA”
- 0 \rightarrow Conducta “INACTIVA”
- $e1[n..0] \rightarrow$ Bus de señales procedentes de la conducta que puede ser suprimida.
- $e2[n..0] \rightarrow$ Bus de señales procedentes de la conducta supresora.
- $s[n..0] \rightarrow$ Bus de señales de la conducta más prioritaria.

Se parametriza el ancho de bus, siendo el más significativo el de Enable.

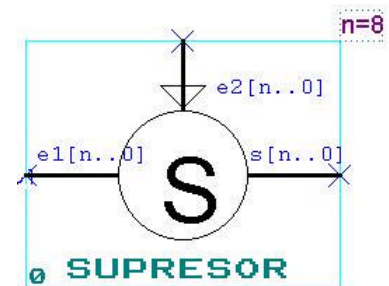


Fig 4.25: Símbolo del módulo supresor.