

CAPITULO 1. OBJETIVOS

1. OBJETIVO

El objetivo de este proyecto es simular el comportamiento de un “depredador” mediante el microbot TRITTON. Para alcanzar este objetivo es necesario utilizar sensores que aporten información de la posición de la “presa” y de los obstáculos que el microbot pueda encontrar en su camino hacia ésta.

En este proyecto se han utilizado módulos digitales receptores de infrarrojos modulados a 38 KHz para la detección del intruso, mientras que para la detección de obstáculos se ha recurrido a relés mecánicos (también llamados bumpers) y a sensores analógicos de infrarrojos. Las razones por las que se han elegido estos sensores se comentarán en el Capítulo 5.

CAPITULO 2. INTRODUCCION

2. INTRODUCCIÓN

Los microbots son pequeños robots móviles, dotados de cierta inteligencia, que deben adaptarse al entorno para llevar a cabo ciertas acciones. Estos han experimentado un gran auge en los últimos años debido a las prestaciones que proporcionan a un bajo coste. Prueba de este auge son las diferentes competiciones que se llevan a cabo durante todo el año, como Sfera o Alcabot, que reúnen normalmente gran número de aficionados.

Estos microbots son una buena herramienta de aprendizaje y formación de ingenieros, por lo que muchas universidades españolas, como la UAM o la Universidad de Alcalá de Henares, están desarrollando proyectos basados en microbots.

2.1.- Desarrollo de proyectos con microbots

A continuación, y para mostrar las diferentes posibilidades de los microbots, se realiza una simple descripción de varios proyectos basados en estos pequeños robots móviles.

Microbot paletizador. Se encarga de rastrear una zona determinada en busca de palés con tres tipos distintos de productos: petit-suisses, natillas y arroz con leche. El microbot es capaz de distinguir qué tipo de palé levanta, gracias a una grúa, y lo lleva a una zona de descarga donde lo dejará en el lugar adecuado, dependiendo del tipo de producto que sea. En cuanto al hardware, son dos placas CT293+ conectadas a la CT6811. El robot consta de ocho sensores de infrarrojos, tres bumpers y tres motores.

Robot autónomo RN3. Este microbot posee un brazo robótico móvil. Está construido tomando como base un kit de Lynxmotion, controlado por microcontrolador Basic Stamp IIsx de Parallax, y tiene capacidad de movimiento proporcionada por dos orugas de goma comandadas por sendos servomotores HiTec HS300, modificados. El brazo lleva adosado en la parte superior de su pinza manipuladora un par de sensores de infrarrojo, para detección de obstáculos, proximidad y medición de luz ambiente. También posee un sistema de videocámara CCD, montada sobre el brazo, que envía imágenes a la tarjeta capturadora del PC por medio de un transmisor de TV UHF que emite imagen y sonido en una frecuencia de 224 MHz. Además está equipado con un sensor volumétrico de ultrasonidos para captar movimiento, y dos encoders ópticos en las orugas para determinar su giro. La alimentación del conjunto se hace mediante dos bloques de baterías de 6 voltios y 700 mA/h, que proporcionan 12 voltios a los circuitos de control y sensores, y una batería de 7,2 voltios y 1600 mA/hora, que se encarga de alimentar los servomotores.

Estos son solo dos ejemplos de lo que se puede conseguir con un microbot. Por ahora la salida comercial más importante para los microbots es la venta de éstos a aficionados a la robótica y para competiciones, aunque con vistas al futuro se abre un abanico de posibilidades para la integración de los microbots en el ámbito industrial y comercial. Por ejemplo, se podría modificar el robot autónomo RN3 para localizar y reparar averías en recintos o tuberías que no son accesibles o son peligrosos.

2.2.- Desarrollo y contenido del proyecto

El proyecto empieza con el estudio del microbot **TRITTON**: tarjeta controladora CT6811, tarjeta CT293+ y tarjeta de expansión de memoria CT256. Se estudiarán los distintos componentes de las tarjetas prestando especial atención al microcontrolador MC68HC11A1, por ser el que rige el funcionamiento del microbot.

Una vez realizado el estudio del microbot y sus componentes principales, y puesto que buena parte de la programación del microbot se realiza en lenguaje C, se comentará el funcionamiento y características del compilador de C usado (IMAGECRAFT 0.47V). Este compilador, que corre bajo entorno MS-DOS, genera programas para la RAM interna y EEPROM del 68HC11, pero con un ligero cambio en los archivos se puede adaptar para que los programas se ejecuten en memoria RAM externa.

Tras la descripción de todos los elementos necesarios para el desarrollo de este proyecto, se comenzará los dos capítulos centrales, en el que se describirá tanto los elementos hardware como la programación necesaria para alcanzar el objetivo de este proyecto.

A continuación se realiza una breve descripción del contenido de los capítulos y anexos que suceden a éste:

Capítulo 3. Se expone la estructura general de un microbot, y de forma específica, la del TRITTON, que es el utilizado en el desarrollo de este proyecto. Se hará una descripción general de las tarjetas CT6811, CT293+ y CT256.

Capítulo 4. Descripción del microcontrolador MC68HC11 y del compilador Imagecraft 0.47V, usado para compilar los programas en C.

Capítulo 5. Capítulo central del proyecto en el que se presentarán los elementos hardware necesarios para construir los circuitos de detección de la presa y de los obstáculos.

Capítulo 6. En este capítulo se hará una descripción detallada del comportamiento del depredador al ejecutar sobre éste los diferentes programas. El código de estos programas se encuentra en los anexos.

Capítulo 7. Se describe el plan de pruebas desarrollado sobre el microbot depredador.

Capítulo 8. Este capítulo se centra en los problemas surgidos y las soluciones adoptadas para resolver esos problemas.

Capítulo 9. En este capítulo se hace una reflexión sobre el desarrollo de este proyecto.

Capítulo 10. Aquí se hace referencia a la bibliografía, sitios web y documentación en general, de donde se ha obtenido la información necesaria para la realización de este proyecto.

En el apartado de anexos se incluyen:

ANEXO A. En este anexo están incluidas las hojas de características de los componentes usados en el circuito emisor de infrarrojos que debe llevar la presa para ser detectada.

ANEXO B. Incluye las hojas de características del módulo receptor de infrarrojos IS1U60, usado para la detección de la presa

ANEXO C. Este anexo contiene las hojas de características de los diodos emisores de infrarrojos (SFH483), los fotodiodos PIN receptores de infrarrojos (SFH203FA) y el amplificador operacional LM324. Estos componentes forman parte de los circuitos detectores de obstáculos.

ANEXO D. Programa en ensamblador, que constituye la aplicación básica que ejecuta el microbot depredador, y que irá almacenado en memoria EEPROM.

ANEXO E. Este anexo incluye la programación en lenguaje C del microbot depredador. Los programas en C se almacenarán en la memoria RAM externa.

ANEXO F. Es un apéndice donde se ha incluido el diseño y simulación de un detector analógico modulado.

CAPITULO 3. MICROBOTS

3. MICROBOTS

En una primera parte de este capítulo, y con el fin de que el lector tenga una idea más clara de lo que es y cuáles son las partes que lo forman, se estudia la estructura básica que posee todo microbot.

La segunda parte del capítulo se centra en el estudio de la estructura del microbot utilizado en el desarrollo de este proyecto, el TRITTON.

3.1.- Estructura general de un microbot

Es casi imposible, cuando se estudia de la estructura de un robot o microbot, no hacer referencia a la analogía existente entre éstos y el hombre, ya que cuando se construyen y programan los robots se hace pensando cómo llevaría a cabo esas tareas el propio hombre. Por lo tanto, siguiendo esa analogía se pueden diferenciar cuatro partes fundamentales en la estructura de un microbot:

Microcontrolador. El cerebro del microbot es el microcontrolador o microprocesador. Este rige el comportamiento del microbot, controlando todas las acciones que realiza.

Sensores. Los sensores son el medio por el que los microbots obtienen información del entorno. Para un microbot los sensores son lo mismo que para una persona los sentidos, es decir el robot ve, oye, etc., mediante los sensores. Por poner un ejemplo, los microbots usan relés mecánicos para “tocar” los objetos que le rodean, y si le quitan esos relés mecánicos es como si una persona perdiera el sentido del tacto.

Motores. Los motores son los que posibilitan el movimiento del microbot, por tanto, según la analogía entre robots y hombres, los motores serían las extremidades.

Esqueleto. Es la estructura que soporta la tarjeta del microcontrolador, los sensores y, en general, todos los demás elementos que forman el microbot. Suele estar formado por piezas Lego, pero se puede usar cualquier cosa, todo depende de la imaginación de quién lo construye.

Además de las cuatro partes mencionadas, suelen existir elementos para controlar la potencia de los motores, circuitos para adaptar la señal de los sensores, etc., pero estas cuatro son las partes fundamentales que todo microbot posee.

3.2.- Microbot TRITTON

TRITTON es el microbot sobre el que se ha desarrollado este proyecto. Lleva como sistema de control la tarjeta CT6811 con un microcontrolador 68HC11A1 y la CT293+ para controlar dos servomotores modelo Futaba 3003, trucados para conseguir giros de 360 grados. También incorpora la tarjeta de ampliación de memoria CT256 con

32K de RAM, ya que la memoria que incorpora el 68HC11A1 está limitada a 256 bytes de RAM y 512 bytes de EEPROM.

La estructura del TRITTON es de Lego e incorpora sensores de infrarrojos de corto alcance CNY70 que le permiten por ejemplo, seguir una línea negra.

Como se puede ver en la figura 3.1 el TRITTON posee todas las partes comentadas en el apartado anterior. Tiene un microcontrolador (el 68HC11A1 de Motorola), tiene unos sensores (en este caso los CNY70), tiene unos motores para que el microbot pueda moverse (Futaba 3003) y tiene un esqueleto formado por piezas Lego.

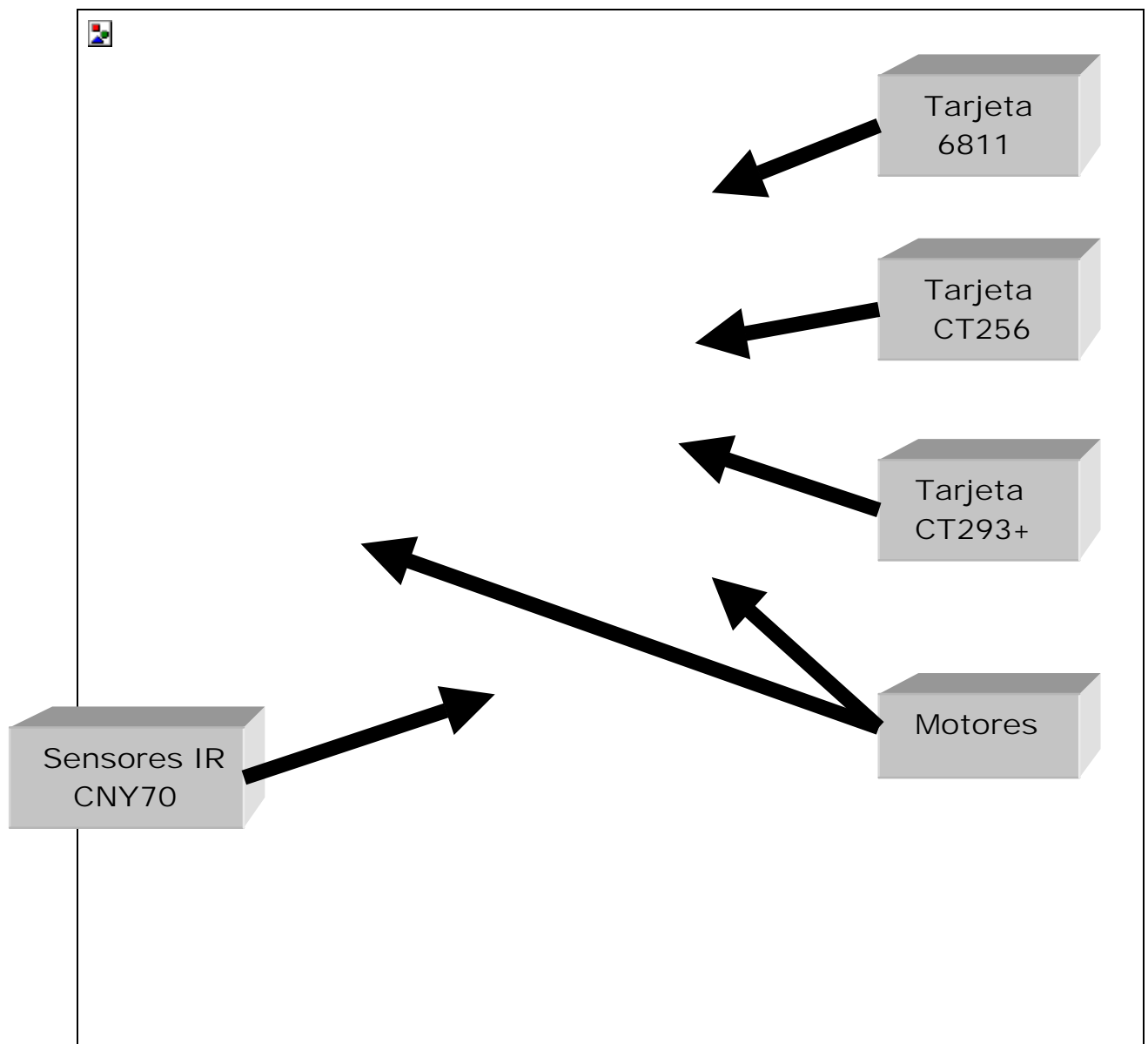


FIGURA 3.1. Microbot TRITTON

Una vez vista la estructura del TRITTON, para poder hacer un uso eficiente de los recursos que ofrece, es necesario realizar una descripción más detallada de las tarjetas que forman la inteligencia de este microbot. Por lo tanto el resto del capítulo está dedicado al estudio de las tarjetas CT6811, CT293+ y CT256.

3.2.1 Tarjeta de desarrollo CT6811

La CT6811 es una tarjeta para el desarrollo de aplicaciones hardware y software basadas en el microcontrolador 68HC11. Se trata de una tarjeta muy versátil, que se puede emplear como:

- Tarjeta entrenadora: Conectándose a ordenadores PC a través del puerto serie. Es posible enviar programas desde el PC a la CT6811 para que los ejecute. De esta manera la tarjeta es muy útil para la depuración de programas en fase de pruebas.
- Tarjeta autónoma de control: La CT6811 funciona en modo autónomo, es decir, sin conectarse al PC. Cada vez que se encienda la tarjeta, se ejecutará el programa previamente grabado en la memoria EEPROM del 68HC11. Conectando periféricos a través de los puertos de expansión, se consiguen desarrollar sistemas inteligentes de control: control de las luces de una casa, manejo de microbots, alarmas...
- Periférico inteligente del PC: También es posible utilizar la tarjeta para comunicar el PC con el mundo exterior. La CT6811 actuaría como un periférico 'inteligente' conectado al PC a través del puerto serie. Este periférico puede recibir las órdenes del PC y actuar en consecuencia.

3.2.1.1.- Características de la CT6811

- **Microcontrolador 68HC11:** Por ello la CT6811 incorpora todas las características de este micro:
 - 512 Bytes de EEPROM en los modelos A1 y A8. En el caso específico de este proyecto el modelo utilizado es el 68HC11A1.
 - 256 Bytes de memoria RAM interna.
 - Temporizador de 16 bits.
 - 3 capturadores de entrada.
 - 5 comparadores con salida hardware.
 - Un acumulador de pulsos.
 - Comunicaciones serie asíncronas.

- Comunicaciones serie síncronas.
- 8 canales de conversión analógico digital.
- Interrupción en tiempo real.
- 4 puertos de E/S.
- **Bus expansión:** Bus de expansión dividido en 6 puertos de 10 bits cada uno. Desde estos puertos se tiene acceso a todas las patas del micro.
- **Switches de configuración:** 2 switches de configuración del modo de arranque de la placa (bootstrap, single chip, expanded, special test) y 2 switches disponibles para aplicaciones de usuario.
- **Led de pruebas** conectado al bit 6 del puerto A. Este led se puede conectar y desconectar mediante un jumper.
- **Pulsador de reset/pruebas IRQ:** Pulsador que inicializa la tarjeta. Este pulsador se puede utilizar también, cambiando un jumper, como un pulsador de propósito general conectado a la entrada de interrupciones IRQ.
- **Niveles VRH y VRL** configurables a VCC y masa respectivamente mediante dos jumpers.
- **Modos de funcionamiento** entrenador/autónomo configurables mediante un jumper.
- **Alimentación** a través de clemas o conector tipo jack.
- **Conexión directa al PC** por medio de un cable tipo teléfono.
- **Reset software y hardware** de la placa.

3.2.1.2.- Modos de funcionamiento de la CT6811

La tarjeta CT6811 se ha diseñado para trabajar en dos modos diferentes: modo autónomo y modo entrenador. Estos modos se configuran colocando o eliminando un jumper (JP5).

Cuando la CT6811 está funcionando en modo entrenador, ésta se conecta a un PC a través del puerto serie. Los programas se escriben y compilan en el PC, y mediante un programa de comunicaciones serie se envían a la entrenadora para ser ejecutados. El modo entrenador también se emplea para realizar programas de autosest de la tarjeta y presentar los resultados en el monitor del PC.

En modo de funcionamiento autónomo la CT6811 al ser inicializada (pulsar botón de reset o alimentar la tarjeta) comienza a ejecutar el programa que se encuentra en

memoria EEPROM interna del micro. Este programa ha sido grabado previamente en la EEPROM en modo entrenador utilizando un software especial.

La CT6811 ha sido diseñada específicamente para funcionar en modo autónomo, reduciendo las dimensiones al máximo para poder realizar microbots pequeños.

3.2.1.3.- Diagrama de bloques de la tarjeta CT6811

Como ya se ha comentado antes, el núcleo central de la tarjeta CT6811 está constituido por el microcontrolador 68HC11A1 de Motorola. Los demás bloques surgen como extensión del 68HC11. La tarjeta se puede dividir en ocho bloques: el corazón de la placa (68HC11), el circuito de reloj, el circuito de inicialización o reset, el circuito de pruebas, el circuito de comunicaciones con el PC, la configuración del sistema, los puertos de expansión y la alimentación del sistema completo. A continuación se explica cada bloque:

- **Microcontrolador 68HC11A1:** Se trata de un microcontrolador de 8 bits. Además de una CPU que le permite ejecutar instrucciones y realizar operaciones aritmético lógicas, dispone en el propio chip de memorias ROM, RAM y EEPROM, así como una serie de periféricos integrados que hacen de este micro una herramienta muy potente.
- **Circuito de reloj:** Este circuito permite que el micro obtenga la señal de reloj necesaria para funcionar. Está construido con un cristal de cuarzo de 8 MHz, que es máximo que permite el 68HC11. Además con este cristal, el micro es capaz de comunicarse con el PC a velocidades de 1200, 2400 ó 9600 baudios.
- **Circuito de reset:** El circuito de reset permite la correcta inicialización del 68HC11. Se ha realizado de tal forma que es posible realizar un reset por hardware, apretando un pulsador, o bien realizar un reset por software desde el PC, cuando la CT6811 esté funcionando en modo entrenador.
- **Configuración:** La parte de configuración de la CT6811 está constituida por seis jumpers y cuatro switches. Dos de estos switches permiten configurar el 68HC11 para trabajar en cualquiera de los cuatro modos para los que está diseñado: bootstrap, single chip, expanded y special. Los otros dos están a disposición del usuario a través de los puertos de expansión para configurar tarjetas periféricas conectadas a la CT6811. Los seis jumpers permiten configurar la CT6811 para realizar diferentes funciones.
- **Circuito de pruebas:** La CT6811 dispone de un led conectado al bit 6 del puerto A del 68HC11. Este led, que se puede conectar o desconectar a través de un jumper, es muy útil para la realización de software de pruebas. El pulsador de la tarjeta se puede configurar mediante un jumper para actuar bien como pulsador de reset o bien como un pulsador normal conectado a la entrada de interrupción IRQ del 68HC11.
- **Comunicaciones con el PC:** A través del circuito de comunicaciones es posible comunicarse con el PC para intercambiar información. Las comunicaciones se realizan a través del puerto serie del PC, mediante la norma RS-232. La velocidad

máxima compatible con el PC es de 9600 baudios, aunque el 68HC11 es capaz de transmitir a mucha más velocidad.

- **Puertos de expansión:** La CT6811 dispone de seis puertos de expansión donde es posible conectar los diferentes periféricos. Todas las señales del 68HC11, salvo la correspondiente al reloj (EXTAL y XTAL), son accesibles desde los puertos de expansión. Cinco de los puertos coinciden con los cinco puertos del 68HC11: A, B, C y E. El sexto puerto está destinado a llevar las señales de control del 68HC11.
- **Alimentación:** La tensión de alimentación de la tarjeta CT6811 oscila entre 4.5 Y 5,5 voltios. La tarjeta dispone de un conector hembra tipo jack cilíndrico para la conexión de un transformador, y dos bornas ajustables mediante tornillos para la conexión de cables de alimentación, provenientes por ejemplo de pilas o baterías.

3.2.1.4.- Configuración de la tarjeta

La tarjeta CT6811 dispone de switches y jumpers para que el usuario la configure según sus necesidades. Existen cuatro switches y ocho jumpers, cuya posible configuración es:

- **Configuración de los switches.** De los cuatro switches, dos se utilizan para configurar los modos de funcionamiento del 68HC11 y los otros dos quedan disponibles para las aplicaciones del usuario.

El 68HC11 puede funcionar en cuatro modos diferentes. Utilizando los mismos nombres que figuran en el manual del 68HC11, los modos son: single chip, bootstrap, expanded y special test. El modo bootstrap es el más utilizado. Este modo se utiliza para trabajar con la CT6811 como tarjeta entrenadora y también como tarjeta autónoma. Sólo en aplicaciones más complejas se utilizan los otros modos.

Para configurar los modos del micro, sólo se utilizan los switches 1 y 2. Estos switches actúan directamente sobre los pines MODA y MODB del 68HC11, poniéndolos a VCC o a GND. En la figura 3.2 se encuentra resumida la información sobre la configuración de los modos.





SWITCHES 1 Y 2	MODA	MODB	MODO	DESCRIPCION
	0	0	<i>Bootstrap</i>	<i>No acceso memoria externa. Modo especial. Ejecución programa BOOTSTRAP.</i>
	0	1	<i>Single chip</i>	<i>No acceso memoria externa. Vectores de interrupción en ROM.</i>
	1	0	<i>Special test</i>	<i>Acceso a memoria externa. Modo especial. Vector interrupción en ROM.</i>
	1	1	<i>Expanded</i>	<i>Acceso a memoria externa. Vectores de interrupción en ROM ó RAM.</i>

FIGURA 3.2. Configuración de los modos mediante los switches de la tarjeta CT6811

- **Configuración de los jumpers.** Existen 8 jumpers que nos permiten configurar aspectos del 68HC11 y de la CT6811.
- **Jumpers JP1 y JP2: Configuración de los niveles VRH y VRL del micro.** Estos jumpers se utilizan para fijar los niveles de tensión de las entradas VRH y VRL del 68HC11. El jumper JP1 actúa sobre VRH. Si está colocado, la pata VRH está cortocircuitada con VCC. Si no se coloca, esta tensión se puede fijar desde el exterior, a través del puerto E. El jumper JP2 actúa sobre VRL. Si está colocado, el pin VRL se cortocircuita con GND. En caso contrario queda accesible desde el exterior a través del puerto E. Estos dos jumpers sólo se utilizan cuando se trabaja con el conversor A/D del 68HC11. Los jumpers normalmente estarán colocados. Sólo para aplicaciones muy específicas habrá que quitarlos y acceder a VRL y VRH desde el puerto E.
- **Jumper JP3: Conexión/desconexión del LED de pruebas.** El diodo LED está conectado al bit 6 del puerto A del 68HC11 cuando el jumper JP3 está colocado. Si se quita, el LED quedará desconectado y el bit 6 del puerto A podrá ser utilizado para otros propósitos que no sean encender el led.
- **Jumper JP4: Configuración del pulsador: RESET/IRQ.** El jumper JP4 se trata de un jumper triple, que puede estar colocado en la zona de la izquierda, cortocircuitando los dos pines situados más a la izquierda, o puede estar colocado en la zona de la derecha, cortocircuitando los dos pines situados más a la derecha. Cuando el jumper está situado en la zona de la derecha (posición marcada como RST en la tarjeta), el pulsador actúa sobre la entrada de reset del 68HC11. Por tanto, en esta posición, cada vez que se apriete el pulsador, **se realizará un reset hardware de la CT6811.** Cuando el jumper está situado en la posición de la izquierda (marcada como IRQ), el pulsador se conecta a la entrada de interrupción no enmascarable IRQ del 68HC11. De esta forma, podemos utilizar el pulsador en nuestros programas para realizar pruebas. Si el jumper está en esta posición, sólo se puede realizar un reset de la CT6811 a través del PC, utilizando el reset software.
- **Jumper JP5: Configuración modo de funcionamiento de la CT6811: Entrenador/autónomo.** Este es uno de los jumpers más importantes. Nos permite decidir el funcionamiento de la CT6811: como una tarjeta autónoma o como una tarjeta entrenadora conectada al PC. Cuando el jumper está quitado, la tarjeta está configurada en modo entrenador. Todo el software habrá que cargarlo desde el PC. Sin embargo, cuando este jumper está puesto, al pulsar reset, la CT6811 comenzará a ejecutar el programa que tiene grabado en la EEPROM. No necesita del PC para cargar ningún software.
- **Jumper JP6: Conexión/desconexión del LVI.** Este jumper conecta o desconecta el dispositivo MC34064 a la entrada de reset. Cuando está conectado, el dispositivo MC34064 previene que se pueda borrar el contenido de la memoria EEPROM del 68HC11. Cuando se detectan niveles de tensión por

debajo de los niveles normales de funcionamiento del 68HC11, el MC34064 realizará un reset del 68HC11. Esto ocurre al encender y apagar la CT6811. Sin embargo, puede ocurrir que mientras la CT6811 esté funcionando, aparezca un pico de caída de tensión y el micro sufra un reset. Por ello se da la opción de desconectar el MC34064 quitando el jumper JP6.

- **Jumper JP7: Conexión/Desconexión del Reset Software.** Ya se ha comentado la posibilidad de poder realizar un reset software (desde el PC) de la CT6811. Esta señal de reset se canaliza por el DTR del puerto serie, esto provoca que algunos programas de comunicaciones no puedan comunicarse con la CT6811. Por eso se ha añadido este jumper que permite separar el DTR de la señal de reset, de forma que se pierde el reset software pero se gana el poder utilizar otros programas de comunicaciones. Si el jumper está colocado abajo (on) estará activado el reset software, si está colocado arriba (off) estará desactivado el reset software.
- **Jumper JP8: Liberación de la XIRQ.** La tarjeta CT6811 se puede utilizar con otros modelos de la familia del microcontrolador 68HC11A1. Uno de ellos es el modelo 68HC11E9, que dispone de 12Kbytes de EPROM. Para poder programar esta memoria no volátil es necesario introducir 12v por la entrada XIRQ. Para poder introducir esta tensión sin dañar los circuitos de la tarjeta CT6811 es necesario quitar el jumper JP8. Esto sólo se hará cuando se vaya a programar la EPROM, una vez finalizada la programación se liberará la XIRQ y se volverá a poner el jumper.

En la figura 3.3 se ha resumido toda la información sobre la configuración de los jumpers.

JUMPER	FUNCIÓN	CONECTADO	QUITADO
JP1	Actuar sobre VRH	VRH = VCC	VRH accesible desde el bus de expansión
JP2	Actuar sobre VRL	VRL = GND	VRL accesible desde el bus de expansión
JP3	Actuar sobre el LED	LED conectado a PA6	LED desconectado PA6 liberado
JP4	Seleccionar usos del pulsador: IRQ/RESET	El jumper en la izquierda activa la IRQ	El jumper en la derecha activa el pulsador de reset
JP5	Seleccionar modo entrenador o autónomo	Modo autónomo	Modo entrenador.
JP6	Actuar sobre el LVI	LVI conectado	LVI desconectado
JP7	Seleccionar reset software ON / OFF	El jumper abajo activa el reset software	El jumper arriba desactiva el reset software
JP8	Actuar sobre XIRQ	Modo normal	Sólo en caso de programar la EPROM del 68HC11E9

FIGURA 3.3. Configuración de los jumpers de la CT6811.

3.2.1.5.- Puertos de expansión.

La tarjeta CT6811 dispone de 6 puertos de expansión. Cinco de estos puertos coinciden con los cinco puertos del 68HC11, y se han denominado igual: puerto A, puerto B, puerto C, puerto D y puerto E. El sexto puerto contiene otras señales de interés del 68HC11.

Estos puertos de expansión son los que van a permitir la comunicación de las demás tarjetas, sensores y periféricos con el microcontrolador. En la figura 3.4 se presentan las señales asociadas a cada puerto. Todos los puertos están constituidos por dos filas paralelas de 5 pines. Los puertos se han diseñado para conectar cables planos de tipo bus de 10 bits.

A continuación se realiza una breve descripción de estos puertos.

Puerto A: Este puerto está constituido por los 8 bits del puerto A del 68HC11 más un pin de VCC y otro pin de GND ambos para poder llevar alimentación a los periféricos de una forma cómoda.

Puerto B: Igual que el puerto A pero con el puerto B del 68HC11. El puerto B del 68HC11 sirve también como parte alta del bus de direcciones en caso de ampliación del micro con memoria externa. Cuando funciona en modo expandido, el bit PB0 se corresponde con A8, PB1 con A9, ..., PB7 con A15.

Puerto C: A este puerto de expansión se lleva el puerto C del 68HC11. Este puerto funciona como un puerto normal de E/S cuando no hay conectada memoria externa. Cuando se conecta memoria externa, este puerto lleva el byte bajo del bus de direcciones multiplexado con el bus de datos. El bit PC0 se corresponde con AD0, PC1 con AD1, ..., PC7 con AD7.

Puerto D: Este puerto está constituido por el puerto D del 68HC11 junto con las señales TXPC, RSTPC, RXPC y GND. El puerto D del 68HC11 dispone de 6 bits, que están compartidos con el SCI y el SPI del 68HC11: PD0/RX, PD1/TX, PD2/MISO, PD3/MOSI, PD4/SCK, PD5/SS. Las señales TXPC y RXPC se utilizan para conectar a la CT6811 otros dispositivos que utilicen la norma de comunicaciones RS-232. Se denominan TXPC y RXPC porque están cortocircuitados con las señales TX y RX provenientes del PC. Si por ejemplo se quiere conectar una calculadora HP a la CT6811, habrá que hacerlo a través de las señales TXPC y RXPC. Por TXPC se transmite la información hacia la CT6811; por RXPC se recibe la información de la CT6811.

Puerto E: Constituido por el puerto E del 68HC11. Este puerto en el 68HC11 tiene una doble función: puerto de entrada de 8 bits ó 8 canales de conversión A/D. Las señales VRL y VRH se utilizan para introducir las señales de referencia alta y baja del conversor. Si los jumpers JP1 y JP2 están colocados, se obtiene la alimentación: VRH=VCC, VRL=GND. Si no están colocados estos jumpers, es posible introducir las tensiones de referencia necesarias para llevar a cabo una aplicación que requiera de unos niveles de tensión distintos a VCC y GND.

Control: Por este puerto se han ‘recopilado’ una serie de señales del 68HC11 para el control. Por los pines SW3 y SW4 se sacan las señales correspondientes al estado de los switches de usuario 3 y 4.

En el caso específico de este proyecto, y como se verá durante el desarrollo del capítulo 5, se han utilizado todos los puertos de expansión: Puertos B y C para conectar la memoria RAM externa (tarjeta CT256), puerto D para la conexión de los sensores que informan de la posición de la presa y el puerto E para recoger la información de los sensores utilizados para detección de obstáculos.

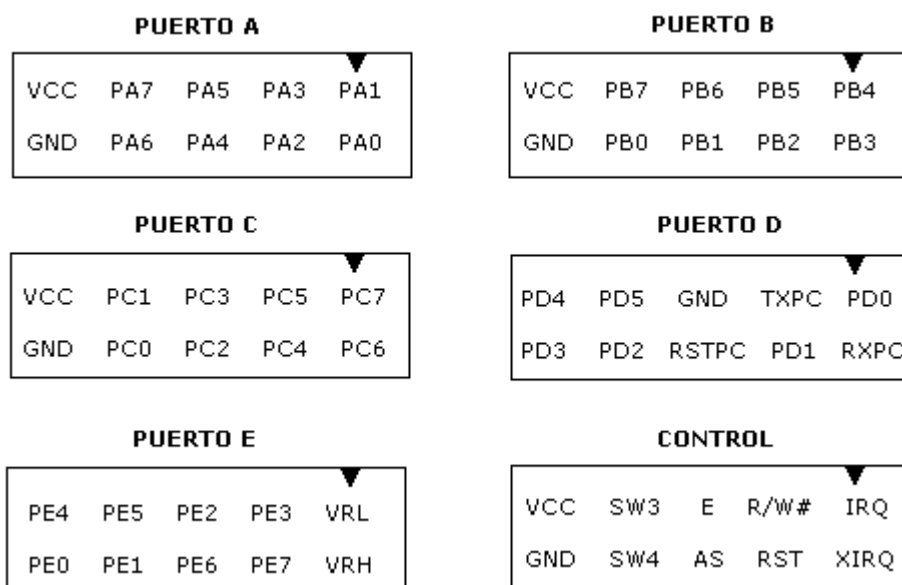


FIGURA 3.4. Señales de los puertos de expansión.

3.2.1.6.- Alimentación

La CT6811 tiene una tensión nominal de alimentación de 5 voltios, aunque se puede alimentar sin ningún problema entre 4.5 y 5.5 voltios. Dispone de dos conectores diferentes de alimentación para alimentar la placa bien por medio de un transformador entre 4.5 y 5.5 voltios conectado a la red o bien directamente a través de pilas o baterías.

La alimentación a través del transformador se emplea habitualmente cuando se está utilizando la CT6811 como una entrenadora. En este caso, puesto que la entrenadora tiene que estar junto al PC, lo mejor es alimentar mediante el transformador. Sin embargo, cuando está funcionando en modo autónomo, es más cómodo utilizar una batería independiente.

La alimentación de la CT6811 se puede hacer mediante dos bornas, que mediante dos tornillos situados en la parte superior, fijan los cables de alimentación o mediante un conector tipo jack. Los terminales de VCC de las bornas y el conector jack se encuentran cortocircuitados, lo mismo que los terminales de GND. Por ello, nunca se

debe alimentar la CT6811 mediante un transformador y baterías a la vez. O se utiliza el transformador, o se utilizan las bornas.

En lo referente a este proyecto, la CT6811 se ha alimentado con baterías cuando el microbot actúa en modo autónomo (programa en ensamblador grabado en la EEPROM del 68HC11) y con una fuente de alimentación cuando se conecta la CT256. Debido a que la tarjeta CT256 debe ir alimentada a una tensión entre 5.5 y 6 voltios, y puesto que no es recomendable alimentar la memoria RAM externa de manera independiente, la tarjeta CT6811 y la CT256 van alimentadas con la misma fuente a una tensión de aproximadamente 5.6 voltios. Todo esto se comentará con mayor detenimiento en el capítulo 5.

3.2.7.- Entorno de desarrollo

En este apartado se va a realizar una breve descripción de los programas usados para la transferencia de ficheros, comunicación entre el PC y CT6811 y la grabación de programas en EEPROM.

As11. Es el programa compilador usado para compilar programas en ensamblador.

Downmcu. Es el programa que permite la transferencia de ficheros desde el PC al microbot una vez compilado el fichero. Es necesario especificar el puerto serie de PC al que está conectado el microbot (com1 o com2).

Ctdialog. Permite ‘dialogar’ con el 68HC11, permitiendo acceder a todos los recursos del 68HC11 desde el PC: memoria RAM, registros de control, memoria EEPROM, etc.

Eeprom. No es un programa en si, sino que es un comando que se ejecuta una vez establecida la conexión mediante el ctdialog para grabar el programa en memoria EEPROM. Hay que tener en cuenta que hay que cambiar la dirección de inicio del programa de la \$0000 a la \$B600, que es la dirección donde comienza la memoria EEPROM.

MCBOOT. Es un programa de comunicaciones para comunicarse con la CT6811. Además de funcionar como terminal de comunicaciones también permite cargar programas en la RAM interna del 68HC11. Utilizando programas de comunicaciones serie en el 68HC11 se va a poder utilizar la pantalla del PC para representar datos y el teclado para enviar datos al 68HC11.

Para obtener información más detallada sobre las características de la CT6811 así como el desarrollo de programas en esta tarjeta se puede acudir al **Manual de usuario de la CT6811**.

3.2.2.- Tarjeta CT293+

Esta tarjeta proporciona al sistema la posibilidad de controlar motores y sensores. Está diseñada para adaptarse a la CT6811 y poder controlarla sin ninguna

variación tanto en *Bootstrap* como en *Single Chip*. La tarjeta también puede controlarse desde otro sistema por ejemplo el puerto paralelo de un PC.

La CT293+ proporciona el movimiento a los sistemas de control y la capacidad para analizar el entorno. Esta tarjeta y la CT6811 forman la plataforma básica del microbot.

En la figura 3.5 se representa el diagrama de bloques hardware de un microbot, la tarjeta CT6811 es el sistema microcontrolador usado para la programación y la tarjeta CT293+ es la interfaz entre la tarjeta anterior y los recursos externos (motores, sensores, entradas digitales y entradas analógicas).

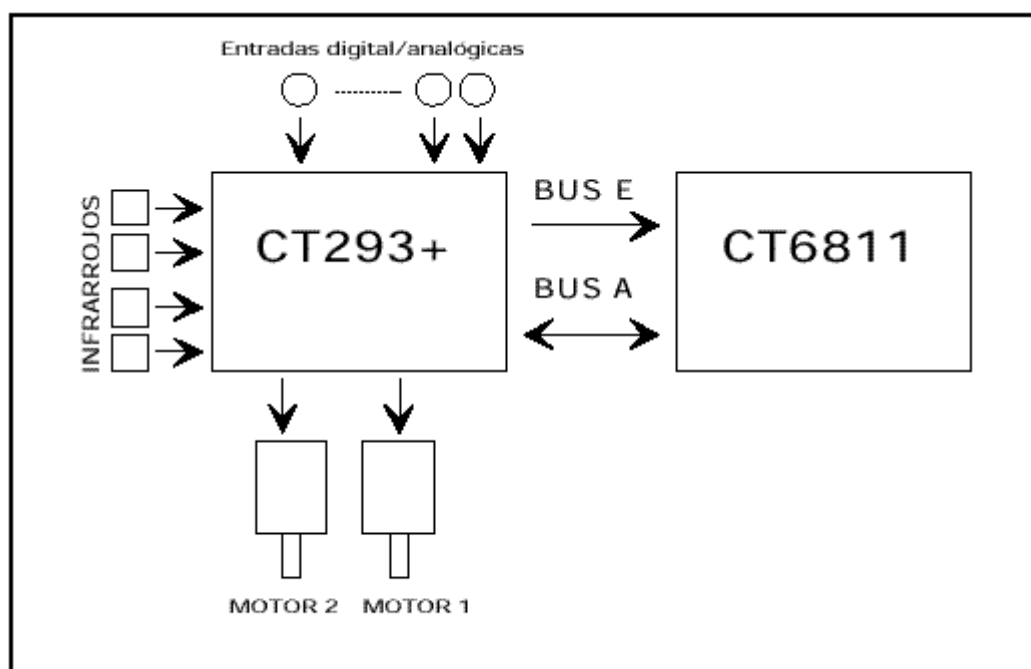


FIGURA 3.5. Diagrama de bloques del hardware de un microbot

Las características de la tarjeta CT293+ se pueden resumir en:

1. Posibilidad de control de dos motores de continua o uno paso_paso.
2. Capacidad para leer cuatro sensores de infrarrojos, pudiendo ser estos optoacopladores.
3. Disponibilidad de 8 entradas digitales de propósito general con la posibilidad de usarlas como entradas analógicas.
4. Conexión de los elementos externos con clemas.
5. Alimentación de los motores externa o interna.

A continuación se realiza una breve descripción de la CT293+ prestando especial atención al PUERTO E, puesto que por este puerto se van conectar los sensores utilizados para detección de obstáculos.

3.2.2.1.- Descripción de los elementos de la CT293+.

Básicamente la CT293+ tiene dos bloques independientes. El BLOQUE A se encarga de los motores y de los sensores de infrarrojos, mientras que el BLOQUE E controla las entradas digitales/analógicas. Hay un bus de control para cada bloque, llamados PUERTO A y PUERTO E. El bus A (Puerto A) es el que maneja el bloque A, es decir motores e infrarrojos, el bus E (Puerto E) maneja el bloque E, que tiene las entradas analógico/digitales. En la figura 3.6 se puede ver los elementos asociados a cada bit del PUERTO A y E.

El bloque A esta formado por el chip L293B1 que gestiona la potencia de los motores y el chip 40106 que tiene dos funciones. La primera consiste en realizar una pequeña lógica para facilitar el uso de los motores, la segunda en realizar una conversión de niveles en la lectura de los infrarrojos. A este bloque también pertenecen los arrays de resistencias de 220 ohmios y 47 Kohmios que se usan para polarizar los infrarrojos.

El bloque E esta formado por un array de ocho resistencias de valor 4K7. Sirve de pull-up para las entradas digitales.

PUERTO A (BUS A)

PA0: estado sensor 1
 PA1: estado sensor 2
 PA2: estado sensor 4
 PA3: motor 1 ON/OFF
 PA4: motor 2 ON/OFF
 PA5: sentido de giro Motor 1
 PA6: sentido de giro Motor 2
 PA7: estado sensor 3
 GND: Masa de la CT293+
 VCC: Alimentación TTL(5v)

PUERTO E (BUS E)

PE0: entrada digital/analógica 1
 PE1: entrada digital/analógica 2
 PE2: entrada digital/analógica 3
 PE3: entrada digital/analógica 4
 PE4: entrada digital/analógica 5
 PE5: entrada digital/analógica 6
 PE6: entrada digital/analógica 7
 PE7: entrada digital/analógica 8
 VRL: nivel bajo del conversor.
 VRH: nivel alto del conversor.

FIGURA 3.6. Bits de los puertos A y E.

3.2.2.2.- Recursos de la CT293+ fundamentales en este proyecto

La CT293+ puede incorporar bastantes tipos de sensores, también puede manejar motores paso a paso y motores de continua. En este apartado se describen los recursos de la tarjeta CT293+ utilizados en el desarrollo del presente proyecto. Con motivo de que sirva de orientación al lector, se enumeran los recursos y una breve descripción de su uso:

- Dos motores de continua Futaba 3003, con fuente de alimentación independiente (Jumper JP1 de la CT293+ quitado).

- Un sensor de infrarrojos CNY70 conectado a PA2 que se ha utilizado como encoder.
- Entradas digitales PE0-PE3 para los sensores de detección de obstáculos.
- Entradas digitales PE4-PE5 para los switches que seleccionan la opción del menú en el programa en C que se ejecuta sobre el microbot en este proyecto.

Conexión de los motores de continua a la CT293+

La tarjeta esta preparada para poder controlar dos motores de continua simultáneamente. Se realiza por medio del circuito integrado L293B que contiene dos circuitos internos con el ‘puente en H’ necesario para controlar motores. Es un integrado que se adapta perfectamente ofreciendo una etapa de control y de potencia en un mínimo espacio.

La conexión de los motores a la tarjeta se realiza a través de las clemas doubles llamadas ‘motor_1’ y ‘motor 2’.

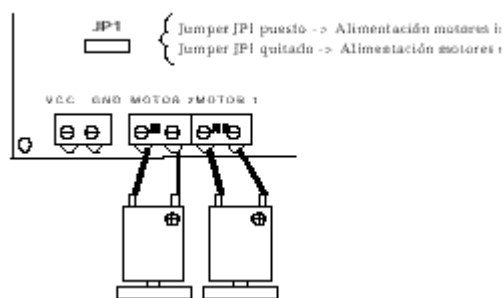


FIGURA 3.7. Esquema de conexión de los motores.

El jumper JP1 es el encargado de seleccionar entre la alimentación interna o externa de los motores. Cuando se conecta el jumper la tensión TTL (+5v) se conecta con la entrada de alimentación de los motores. La ventaja es que con una sola fuente de tensión se puede dar energía a todo el sistema. El inconveniente esta en el ruido que los motores introducen en el sistema.

Cuando el jumper JP1 de la CT293+ esta quitado significa que se ha seleccionado la alimentación de motores externa. En este caso para que los motores se muevan se tiene que introducir la alimentación por la clema situada junto a las clemas de los motores. (Ver figura 3.7.). La desventaja es la necesidad de utilizar dos fuentes de alimentación, mientras que la ventaja es el poder poner más tensión a los motores.

Conexión de los sensores de infrarrojos CNY70

Los infrarrojos CNY70 son de corta distancia. Son capaces de distinguir el negro de otro color mientras la distancia del infrarrojo a la superficie no supere unos pocos centímetros. En concreto esta tarjeta tiene capacidad para controlar directamente cuatro

a través de los conectores de tres pines situados en la parte trasera de la placa (ver figura 3.8).

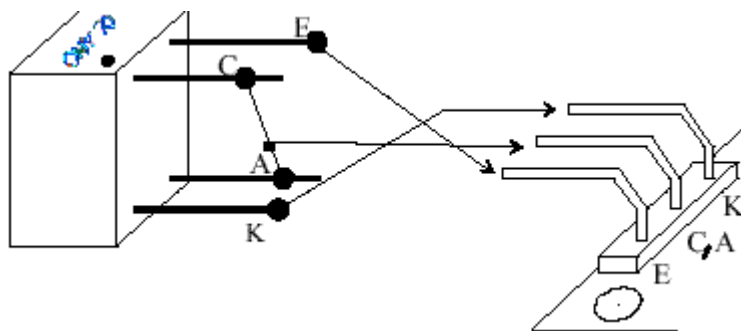


FIGURA 3.8. Esquema de montaje del sensor CNY70.

Conexión de la entradas digitales

Hay un total de 10 entradas pero sólo 8 son entradas, las otras dos se usan para establecer los niveles de referencia VRH y VRL. En modo digital estos niveles serán VRL=0 y VRH=5. Si se trabaja con la CT6811 y se tienen los jumpers JP1 y JP2 puestos los niveles se sitúan automáticamente en VRL=GND='0' y VRH=VCC='5'.

En la figura 3.9. se presenta el circuito de polarización desarrollado en la tarjeta CT293+. Según el circuito de polarización la lectura que se obtiene cuando no se conecta nada en las entradas, estando los niveles de referencia a VCC y GND, es prácticamente VRH, es decir VCC (nivel alto). Este comportamiento se debe a la resistencia de pull-up de 4.7k que se puede ver en la figura 3.9.

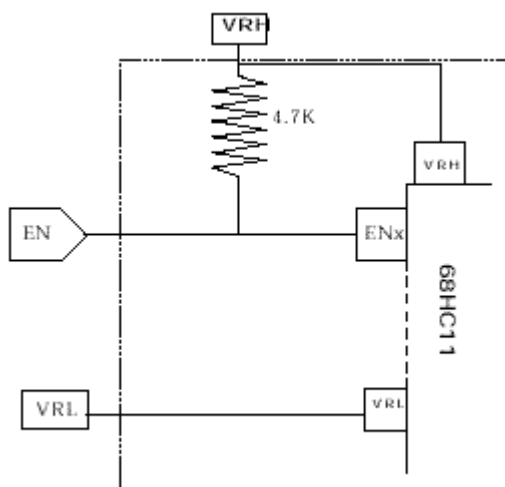


FIGURA 3.9. Circuito de polarización interno de la CT293+

Los elementos de la CT293+ descritos anteriormente son los que hay que conocer para la realización del presente proyecto. No obstante, si se quiere saber más sobre esta tarjeta y su programación se puede acudir al **Manual de la tarjeta CT293+**.

3.2.3.- Tarjeta CT256

La tarjeta CT256 es una ampliación de memoria diseñada para conectarse a la CT6811, con las siguientes características:

- Posibilidad de cubrir el espacio completo de direccionamiento del 68HC11, en total 64Kb.
- Espacio de direcciones dividido en dos bancos:
 - Banco 0: Situado en el espacio de direcciones 8000-FFFF (es el que viene por defecto).
 - Banco 1: Situado en el espacio de direcciones 0000-7FFF.
- Cada banco puede funcionar como RAM o como EEPROM, dependiendo del estado de los jumpers JP2 y JP3. Por defecto ambos bancos están configurados como RAM.
- Conexión eléctrica a la CT6811 mediante tres cables planos tipo bus.
- La alimentación se toma directamente de los cables tipo bus, es decir, que la alimentación es la misma que la de la CT6811. El conector de power J5 no se suele utilizar.
- La alimentación del conjunto CT6811-CT256 debe estar comprendida entre 5.5 y 6 voltios. Sin embargo esto depende del chip de memoria que se tenga conectado.

3.2.3.1.- Elementos de la CT256

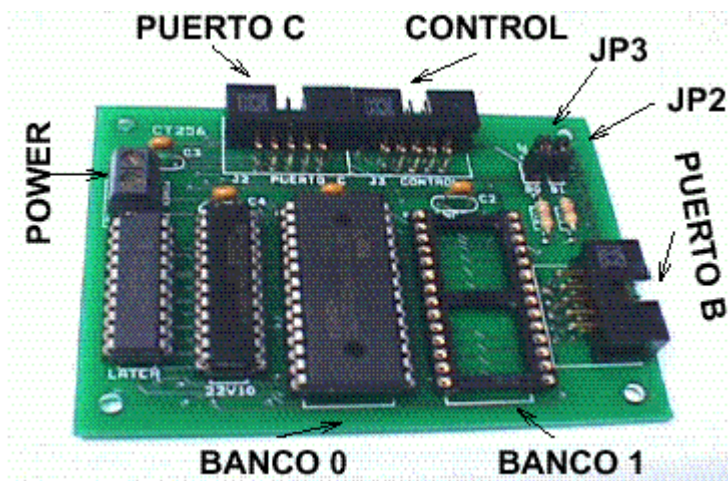


FIGURA 3.10. Elementos de la tarjeta CT256.

En la figura 3.10 se puede observar una tarjeta CT256 con sus principales elementos. A continuación se realiza una breve descripción de estos elementos:

Puerto C. Se conecta al Puerto C de la CT6811. Lleva el bus de datos y parte del bus de direcciones.

Control. Se conecta al Puerto de control de la CT6811. Lleva las señales de control.

Puerto B. Se conecta al Puerto B de la CT6811. Lleva una parte del bus de direcciones.

Banco 0. Memoria situada en las direcciones 8000-FFFF.

Banco 1. Memoria situada en las direcciones 0000-7FFF.

JP2. Jumper de configuración del Banco 1. Si está colocado, en el Banco 1 debe haber una memoria RAM y si está quitado una EEPROM.

JP3. Jumper de configuración del Banco 0. Si está colocado, en el Banco 0 debe haber una memoria RAM y si está quitado una EEPROM.

Power. Alimentación general. No se suele utilizar, la alimentación se introduce a la CT6811, que se lleva a la CT256 por los cables de bus y está presente en power.

3.2.3.2.- Chips de memoria

La CT256 viene por defecto con un chip de memoria RAM de 32 Kb en el Banco 0, aunque es posible situar en cualquiera de los dos bancos las siguientes familias de chips:

1. **Memoria RAM estática** (Jumper JP2 o JP3 colocados)

- Familia: 65156, 65C256, 76C256. Memorias SRAM.
- Memorias SRAM no volátil DS1230Y de DALLAS o similares.

2. **Memoria EEPROM** (Jumper JP2 o JP3 quitado)

- Familia 27C512.

Todas las memorias deben ser de 32Kb y con un tiempo de acceso menor de 200ns.

Disponer de una SRAM permite cargar programas en la RAM y que permanezcan inalterados aún sin alimentación. Estas memorias tienen la ventaja

respecto a las EEPROM de que son más rápidas y se pueden reescribir infinitas veces. Esto permite la realización de sistemas autónomos muy flexibles al cambio.

En el caso particular de este proyecto se ha utilizado la memoria RAM de 32 kb que viene con la CT256, aunque hubiera sido más cómodo utilizar una SRAM o una EEPROM puesto que no se tendría que cargar un programa en la memoria cada vez que se quisiera ejecutar o se perdiera la alimentación.

3.2.3.3.- Configuración de los switches de la CT6811

Cuando se trabaja con memoria externa, el 68HC11 debe estar en modo expandido (*expanded*), lo que obliga a que se envíen las direcciones y datos por el bus externo para acceder a la memoria. Sin embargo la posición de los switches 1 y 2 de la tarjeta CT6811 se deben configurar según se esté trabajando en modo entrenador o modo autónomo.

- **Modo entrenador.** Los switches deben estar hacia abajo (posición ON), igual que cuando se trabaja con la CT6811 en modo entrenador. La diferencia está en el software que se carga. Al emplear sólo la CT6811, sin memoria externa, se cargan programas en la RAM interna y se ejecutan. Al trabajar con la RAM externa se utilizan herramientas software que cargan en la RAM interna un programa cargador que configura mediante software el 68HC11 en modo expandido. Pero primero es necesario cargar en la RAM interna y de ahí la necesidad de tener los switches 1 y 2 en posición ON.
- **Modo autónomo.** Los switches 1 y 2 deben estar en posición OFF. Lo normal en este modo es tener memoria no volátil en los bancos de memoria, de forma que los vectores de interrupción estén modificados. Al pulsar el botón de reset de la CT6811 el microcontrolador comenzará a ejecutar código a partir de la dirección indicada por el vector de reset almacenado en las posiciones FFFE-FFFF.

Es importante decir que siempre tendrán preferencia los recursos internos del 68HC11 y por tanto, solo se intenta acceder al mapa de memoria externo si no hay nada mapeado en la RAM interna.

En este proyecto se dispone de una memoria RAM externa volátil por lo que se trabajará en modo entrenador, utilizando la memoria RAM interna para configurar el microcontrolador en modo expanded. Cómo ejecutar programas en la memoria externa se explicará en el siguiente capítulo cuando se estudie el compilador de C Imagecraft 0.47V, ya que antes de ejecutar un programa es necesario compilarlo.

CAPITULO 4. MICROCONTROLADOR MC68HC11

4. MICROCONTROLADOR MC68HC11

Como ya se ha comentado en capítulos anteriores el microcontrolador MC68HC11 es el cerebro del microbot TRITTON. Por esta razón se va a dedicar una primera parte de este capítulo a la introducción de las características principales de este microcontrolador, prestando especial atención a los recursos del 68HC11 utilizados en este proyecto. Asimismo, la parte final del capítulo se dedica a la introducción de los compiladores de ensamblador y C utilizados para compilar los programas desarrollados.

4.1.- Descripción del microcontrolador MC68HC11

El microbot usado en este proyecto tiene como cerebro un MC68HC11A1 que se diferencia de los demás microcontroladores de la familia (68HC11A8, 68HC11E2, etc.) en la memoria de la que dispone, que en el caso del 68HC11A1 es de 256 bytes de RAM y 512 bytes de EEPROM.

El 68HC11 es un microcontrolador de 8 bits fabricado con tecnología HCMOS, con una frecuencia de bus de 2 MHz y con una amplia lista de recursos internos. En concreto el modelo A1 dispone de:

- 256 bytes de memoria RAM.
- 5 puertos de 8 bits, con pines de entrada, salida y de entrada/salida.
- Conversor analógico-digital de 8 canales y 8 bits de resolución.
- Una UART para comunicaciones serie asíncronas (SCI).
- Un módulo de comunicaciones serie síncronas (SPI).
- 5 comparadores con salida hardware.
- 3 capturadores de entrada.
- Un acumulador de pulsos externos de 8 bits.
- Temporizador principal de 16 bits.
- Interrupciones en tiempo real.
- 2 entradas de interrupciones externas.
- Software en ROM para cargar un programa externo en la RAM interna (programas ctdialog, as11 y MCBOOT).

Algo ha tener en cuenta a la hora de utilizar estos recursos es que muchos de ellos no son accesibles simultáneamente. Este es el caso cuando se utiliza la memoria externa, que usa los puertos B y C para los buses de datos y direcciones. Por lo tanto hay recursos a los que no se podrán acceder.

Modos de funcionamiento del 68HC11

El 68HC11 puede funcionar en 4 modos diferentes: Single chip, expanded, bootstrap y special test.

Single chip: En este modo de funcionamiento, el mapa de memoria del 68HC11 está constituido por la memoria RAM, la memoria EEPROM, los registros de control y la memoria ROM. Este modo está pensado para funcionar cuando

existe un programa grabado en la ROM, de tal manera que al arrancar se comience a ejecutar el programa indicado por los vectores de interrupción que se encuentran en ROM.

Expanded: Además del mapa de memoria del modo single chip, es posible acceder al resto de las posiciones de memoria conectando memorias externas. El precio a pagar es que se pierden dos puertos de E/S, el puerto B y C, que se utilizarán como bus de datos y direcciones. En este modo se puede utilizar la memoria ROM interna, pero también es posible deshabilitar esta ROM y acceder a memoria externa y con ello a los vectores de interrupción que se encuentren en esa memoria externa.

Bootstrap: Este modo difiere del single chip en que los vectores de interrupción no se encuentran en la memoria ROM de 8K sino que se encuentran en otra memoria ROM, llamada ROM de arranque. Al arrancar en este modo, automáticamente comienza a ejecutarse el programa BOOTSTRAP que se encuentra en ROM.

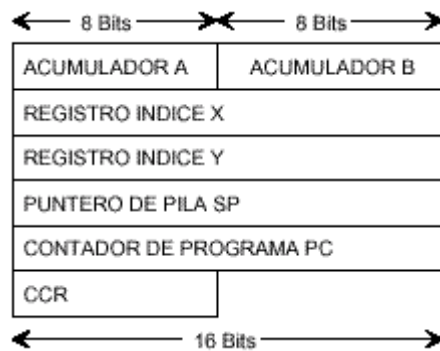
Special test: Igual que el modo Bootstrap con la salvedad de que se puede acceder a memoria externa. Este modo se utiliza para realizar pruebas de fábrica. En este modo especial se tiene acceso a determinados registros de control que en otros modos están protegidos.

Los modos de funcionamiento se pueden configurar de dos formas diferentes: configuración hardware y configuración software. La configuración hardware consiste en colocar unos determinados niveles lógicos en las patas **moda** y **modb** del 68HC11. Al realizar un reset del 68HC11 el micro arrancará en el modo especificado por las tensiones de las patas moda y modb. La configuración software del modo de funcionamiento se basa en la modificación del registro HPRIO situado en la dirección \$103C.

En este proyecto se han utilizado dos de estos modos de funcionamiento: el modo Bootstrap para la ejecución del programa en lenguaje ensamblador grabado en EEPROM (Ver figura 3.2.) y el modo Expanded para los programas cargados en memoria externa, cuya configuración se hace por software modificando el registro HPRIO.

Registros de la CPU

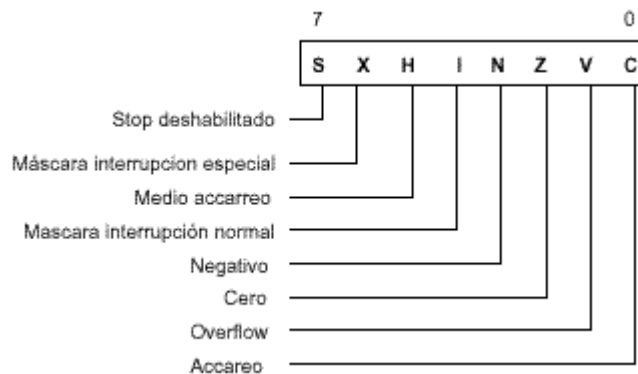
La CPU del MCU dispone de 2 registros acumuladores de 8 bits, que se unen para formar el registro D de 16 bits, siendo el acumulador A la parte alta y el acumulador B la parte baja. Además dispone de 2 registros para direccionamiento indexado X,Y ambos de 16 bits. El puntero de pila y el contador de programa son también de 16 bits, lo que permite que la longitud máxima de un programa sea de 64 Kbytes, que es el espacio máximo direccionable por el MCU. El registro CCR es el llamado registro de estado, que contiene unos bits de especial importancia que reflejan el estado de la CPU. En la figura 4.1 se puede ver los registros del 68HC11.

**FIGURA 4.1.** Registros del 68HC11.

El puntero de pila debe ser inicializado por el usuario. La pila "crece" desde direcciones altas a direcciones bajas, por lo que al introducir un elemento en la pila, SP se decrementa en 1 ó 2 bytes dependiendo del tamaño del dato metido en la pila. Al sacar un elemento de la pila, SP se incrementa.

El contador de programa PC se va incrementado según se van ejecutando las instrucciones. Por tanto, los programas se ejecutan desde direcciones bajas a altas y la pila crece de direcciones altas a bajas. Es importante dar a SP un valor "seguro" de tal manera que la pila no se solape con el código, si es que el código se encuentra en RAM.

El registro CCR es de 8 bits. Cada bit tiene una letra asignada y representa una situación diferente del estado de la CPU. Este registro se puede ver en la figura 4.2.

**FIGURA 4.2.** Registro CCR

El conocimiento de estos registros, modos de direccionamiento e instrucciones en ensamblador son esenciales para poder desarrollar y comprender programas en ensamblador como el realizado en este proyecto.

Puesto que no entra dentro de los objetivos de este proyecto ser un tutorial sobre programación en ensamblador y C, no se entrará en detalles de cómo realizar la programación. Si no se tienen conocimientos de programación en ensamblador y C se puede acudir a los manuales de programación del 68HC11, manuales de sistemas

digitales, manuales disponibles en la web. En el apartado de bibliografía se puede encontrar referencias a libros y páginas web donde se puede conseguir este tipo de información.

Interrupciones

Es necesario conocer como realiza las interrupciones el 68HC11 puesto que en la programación en ensamblador se han tomado las lecturas del **encoder** por interrupciones. Como se verá en el capítulo 5 este encoder está formado por un CNY70 que provoca una interrupción cada vez que detecta un cambio de blanco a negro o viceversa.

Las interrupciones son señales generadas interna o externamente al microcontrolador que provocan que la CPU deje de ejecutar el programa en curso y ejecute una rutina específica para atender a la interrupción. Una vez ejecutada la rutina de servicio de la interrupción, la CPU continúa con el programa que estaba ejecutando antes de producirse la interrupción.

Las direcciones de las rutinas de servicio de las interrupciones se encuentran en una tabla en memoria, denominada tabla de vectores de interrupción. Existen dos tablas de vectores de interrupción según el modo de funcionamiento del MCU. Si el modo de funcionamiento es el especial de arranque, la tabla se encuentra en memoria ROM en las direcciones \$BFD6-\$BFFE. Si funciona en modo normal o extendido, la tabla se encuentra en las direcciones \$FFC0-\$FFFE. En la tabla hay 21 vectores, y cada vector contiene una dirección de memoria que ocupa 2 bytes, por tanto las tablas de vectores ocupan $21 * 2 = 42$ bytes.

Existen 3 tipos de interrupciones: las interrupciones enmascarables, las no enmascarables y las interrupciones software.

Las interrupciones enmascarables, como su nombre indica, se pueden enmascarar, es decir, inhibir sin más que actuar sobre el bit I del CCR. ($I=0 \rightarrow$ se permiten interrupciones; $I=1 \rightarrow$ Interrupciones inhibidas). Con la instrucción CLI, el bit I del CCR se pone a cero y se permiten las interrupciones. Con SEI, I se pone a 1 y se inhiben las interrupciones.

Las interrupciones no enmascarables no se pueden inhibir. Son las interrupciones correspondientes a un fallo en el MCU, instrucción ilegal, RESET y la interrupción externa XIRQ.

Las interrupciones software son las producidas por el propio programador en unos instantes totalmente conocidos. Sólo existe una interrupción software que se produce con la instrucción SWI. (equivalente al TRAP del 68000). La CPU al leer esta instrucción actúa como si de una interrupción normal se tratase. Estas interrupciones son no enmascarables porque de lo contrario la CPU se colgaría.

Una vez que se conoce que es una interrupción y que tipos de interrupción hay, se muestra el proceso de ejecución de una interrupción: La CPU del microcontrolador está ejecutando un programa. De repente aparece una interrupción. Si las interrupciones están permitidas (Bits X e I del CCR) se introducen todos los registros en la pila, se

inhiben las interrupciones y se obtiene de la tabla de vectores de interrupción la dirección a la que tiene que bifurcar la CPU. Las interrupciones se inhiben para que no se produzcan anidamientos de interrupciones, es decir, que mientras se está atendiendo a una interrupción, por defecto no se atiende a otra. El usuario por supuesto puede activar las interrupciones en una rutina de servicio de una interrupción, con lo que provocaría un anidamiento de interrupciones (No es recomendable anidar interrupciones). Si se está ejecutando la rutina de servicio de una interrupción enmascarable y se produce una interrupción no enmascarable, como esta última tiene una prioridad mayor, la CPU procede a atenderla dejando a la anterior "congelada". Cuando termina con la no enmascarable vuelve con la enmascarable. Todas las rutinas de servicio de interrupción deben acabar con la instrucción RTI que indica a la CPU que la interrupción ha sido atendida y que se puede retornar al programa anterior: La CPU toma de la pila todos los registros que previamente había guardado y continúa ejecutando el programa que había sido interrumpido. En el microcontrolador existen muchos periféricos integrados que pueden producir interrupciones en cualquier momento. Para generar una interrupción activan un bit de un determinado registro del periférico. Este bit indica a la CPU que el periférico en cuestión ha solicitado interrupción. Cuando la CPU pasa a ejecutar la rutina de servicio del periférico, lo primero que debe hacer es desactivar este bit. Si no se hace, al retornar de la interrupción con RTI, el bit seguirá activo y la CPU lo interpretará como una nueva interrupción, con lo que se vuelve a ejecutar la rutina de servicio. Así permanecería la CPU en un bucle infinito.

Por otra parte los **vectores de interrupción** se sitúan en direcciones diferentes dependiendo del modo de funcionamiento. En los modos single chip y Expanded, los vectores de interrupción están situados a partir de la dirección FFD6. En el modo Bootstrap los vectores se encuentran en memoria ROM a partir de la dirección BFC0. Puesto que estos vectores están en ROM su valor no se puede cambiar. Estos vectores apuntan a diferentes zonas de la RAM. Es allí donde se debe realizar el salto a la subrutina de atención a la interrupción.

En el caso particular del encoder utilizado en este proyecto para contar los pasos dados por el microbot, se han permitido las interrupciones del capturador de entrada uno (IC1) que tiene asociado al sensor 4 de la CT293+ (Bit 2 del PUERTO A).

```
ACUM      EQU $00E8      /* Dirección de RAM a donde apunta el vector de
                          Interrupción. */

inic      LDAA #$7E      /* El código $7E equivale a JMP y significa
                          'salto' */
          STAA ACUM      /* Almacena JMP en dirección $00E8 */
          LDX #rut_enc   /* Almacena la dirección de rut_enc en el byte
          STX ACUM+1     /* siguiente a JMP */

ini_enc   BSET TCTL2,X $30 /* Capturador de entrada uno por flancos de
                          subida y bajada */
          BSET TMSK1,X $04 /* Se desenmascara interrupciones del
                          capturador de entrada uno (IC1) */
          CLI           /* Se permiten las interrupciones.*/
          ....
          ....

rut_enc   LDD paso_s1    /* Cuando hay una interrupción de IC1 se
          ADDD #1        /* incremeta paso_s1 */
```

```
STD paso_s1
LDAA #$04      /* Se desactiva el flag de interrupción IC1 */
STAA TFLG1,X   /* para que no se ejecute solo la interrupción
RTI            /* Retorno de interrupción (pila se vacía) */
```

FIGURA 4.3. Código ensamblador para interrupción de encoder

En la figura 4.3 se puede ver el código asociado a la interrupción que se produce cada vez que el encoder lee un paso. En este caso se ha habilitado la interrupción enmascarable producida por IC1 tanto por flanco de bajada como de subida. En el código se puede observar que la dirección de RAM a la que apunta el vector de interrupción es la \$00E8, pero al estar el programa en EEPROM se debe saltar de esta dirección a donde se encuentre la rutina de atención a la interrupción de IC1 (**rut_enc**).

Con la figura 4.3 se intenta aclarar en lo posible lo referente a las interrupciones. En este código se observa como se habilitan las interrupciones (CLI), como es necesario desactivar la interrupción una vez que se ha producido, como se devuelve el control al programa principal liberando la pila (RTI), etc.

Puertos de Entrada/Salida

Ya en el capítulo tres al describir los puertos de la tarjeta CT6811 se realizó un inciso sobre el uso de los puertos de entrada/salida en el desarrollo de este proyecto. Por si acaso no se ha leído el capítulo tres, decir que en este proyecto se han utilizado todos los puertos:

- Puerto A. Control de motores y encoder (sensor 4).
- Puertos B y C. Actúan en modo expandido. Llevan el bus de datos y direcciones al conectar la memoria externa (CT256).
- Puerto D. Por este puerto se recoge la información de los sensores de posición de la presa.
- Puerto E. Este puerto se utiliza para recoger la información de los sensores de obstáculos y de los switches que permiten cambiar de opción de recorrido en el programa en C.

En el capítulo 5 cuando se describa la programación del microbot depredador, se informará qué sensores están asociados a que bit de que puerto. Por ahora sólo se van a describir los puertos de entrada/salida del microcontrolador 68HC11.

Existen 5 puertos de 8 bits disponibles: Puerto A, B, C, D y E. Además de comportarse como puertos normales, sus pines están compartidos con alguno de los recursos internos.

Puerto A. El puerto A dispone de 3 pines de entrada, 4 pines de salida y uno configurable como entrada o salida. Se encuentra mapeado en memoria en la dirección \$1000. Los pines del puerto A están compartidos por otros recursos: comparadores,

acumulador de pulsos y capturadores. En la figura 4.4 se muestran todas las funciones que están asignadas a cada pin del puerto A.

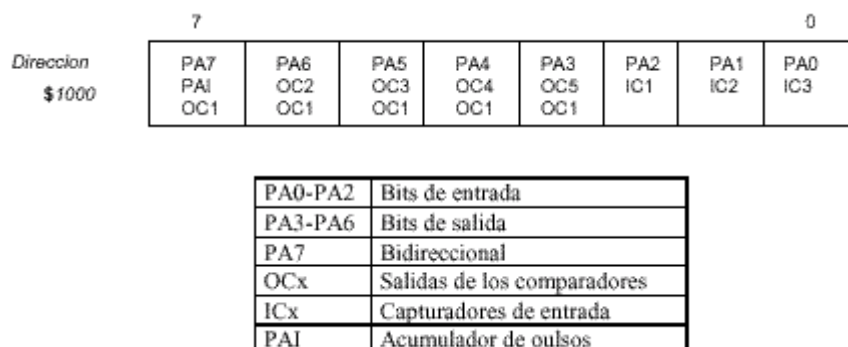


FIGURA 4.4. Recursos que utilizan cada bit del puerto A.

Por defecto los recursos internos asociados a los pines del puerto A están "desconectados". El puerto A se comporta como un puerto normal en el que si se escribe un valor en la dirección \$1000 se reflejará en los correspondientes pines de salida y si se lee un valor, se hará de los pines de entrada.

Puerto B. Sus 8 bits son de salida. En el modo no expandido del MCU se comporta como un puerto de salida. En el modo expandido se utiliza para mandar el byte alto del bus de direcciones (Ax). Su dirección es la \$1004. En la figura 4.5 se muestran los usos de los bits del puerto B.

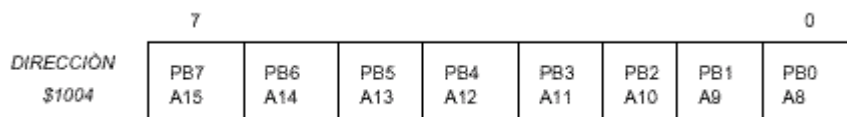


FIGURA 4.5. Usos de los bits del puerto B.

Puerto C. Es un puerto de entrada/salida. En el modo no expandido sus 8 bits pueden actuar como entradas o salidas independientes, según cómo se configuren los bits en el registro DDRC (\$1007). Un cero en un bit del registro DDRC configura el pin correspondiente para entrada. Un uno lo hace para salida. La dirección del puerto C es \$1003.

Si el microcontrolador está funcionando en el modo expandido, el puerto C (PCx) actúa como parte baja del bus de direcciones (Ax) multiplexada con el bus de datos (Dx). En la figura 4.6 se puede ver los posibles usos de los bits del este puerto.

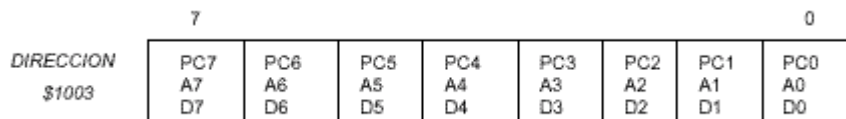


FIGURA 4.6. Uso de los bits del puerto C.

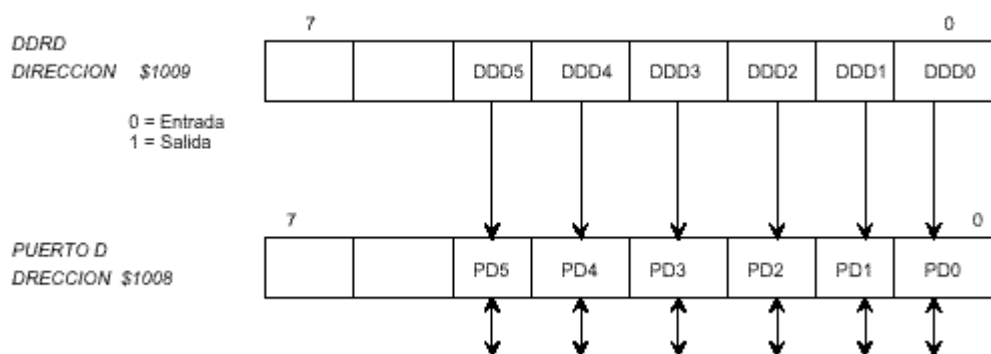
Puerto D. Igual que el puerto C, es un puerto de entrada/salida en el que se pueden configurar sus bits independientemente para entrada o salida. El registro de configuración DDRD se encuentra en la dirección \$1009. Unos se corresponden con salidas y ceros con entradas. El puerto está mapeado en la dirección \$1008. Este puerto es de sólo 6 bits. Está compartido con el SPI y el SCI.

Los bits 0 y 1 están compartidos con el puerto serie. Por ello, para poder ser utilizados es preciso que SCI esté desactivado. Los bits 2,3,4 y 5 están compartidos por el SPI. Para que funcionen es necesario asegurarse que el SPI está desactivado.

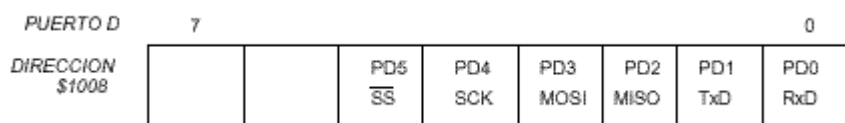
Una de las características del puerto D es que sus salidas pueden ser normales o de colector abierto. Esto se configura con el bit 5 del registro SPCR (\$1028). Si este bit está a 0, las salidas serán normales. Si está a 1, las salidas estarán en colector abierto. Por defecto están en colector abierto.

La figura 4.7 muestra: Por una parte el puerto D y su configuración como entrada o salida independiente mediante el registro DDRD. Por otra parte los usos que se le pueden dar a los bits de este puerto.

NOTA: En la tarjeta CT6811 se debe seleccionar el modo con salidas en colector abierto puesto que existen resistencias de pull-up colocadas en algunas salidas. No se deben configurar como salidas normales.



Puerto D y su configuración.



Usos de los bits del puerto D.

FIGURA 4.7. Configuración y uso del puerto D.

Puerto E. Es un puerto de 8 bits de entrada (PE_x). Está situado en la dirección \$100A. Comparte pines con canales del conversor A/D (AN_x). Para poder utilizar el puerto E como un puerto normal es preciso que el conversor A/D interno del 68HC11 esté desconectado. Por defecto este conversor está desconectado. La figura 4.8 muestra los posibles usos de los bits del puerto E.

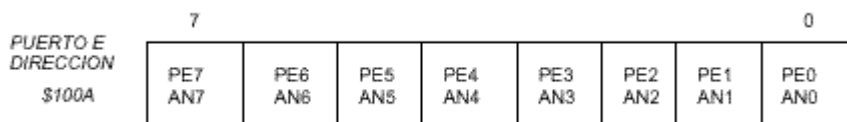


FIGURA 4.8. Usos de los bits del puerto E.

Temporizador principal

El temporizador está formado por un registro de 16 bits que se está incrementando constantemente. Arranca con el valor 0, se va incrementando con cada pulso de reloj y cuando llega a \$FFFF se produce una interrupción de overflow, pasa a valer nuevamente 0 y continúa contando.

El valor del temporizador se puede leer en cualquier momento pero no escribir. La cuenta por tanto no se para. La señal de reloj que incrementa el temporizador se obtiene dividiendo la frecuencia de la señal de reloj E por uno de estos valores: 1,4,8 ó 16.

El registro que contiene el valor del temporizador es el TCNT y se encuentra en las direcciones \$100E (Byte mayor peso) y \$100F (Byte de menor peso). Para leer su valor basta con utilizar cualquiera de los registros de 16 bits (registros X o Y).

Comparadores

Existen 5 comparadores independientes de 16 bits en el microcontrolador. El funcionamiento de cada comparador es el siguiente: Se introduce un valor en sus registros correspondientes. El temporizador principal se va incrementando. Cuando el valor del temporizador es igual al de algún comparador, se produce una interrupción (si están habilitadas) y se modifica opcionalmente el estado lógico de un pin del microcontrolador.

Puesto que los comparadores son independientes, existe una interrupción para cada uno que se enmascara mediante un bit situado en el registro de máscara de interrupciones TMSK1 (\$1022). El registro de interrupciones TFLG1 (\$1023) contiene las banderas de interrupción: los bits que se activan cuando se produce alguna de las interrupciones de los comparadores. Estos bits deben ser puestos a cero antes de terminar la rutina de servicio de interrupción.

Cuando el valor del comparador es igual al del temporizador, se produce una acción determinada en el pin sobre el que está actuando el comparador. Existen 4 acciones posibles que se seleccionan mediante 2 bits para cada comparador. Estos bits

se encuentran en el registro TCTL1 (\$1020). Las acciones posibles son: No afectar al pin correspondiente, poner el pin a '1', poner el pin a '0' y cambiar de estado el pin.

La salida hardware esta activa cuando al activarse el comparador se produce una acción sobre el pin de salida asociado. En caso contrario se dice que la salida hardware esta inhibida. En este sentido el comparador 1 es diferente al resto puesto que él puede actuar sobre los cinco pines simultáneamente. Los otros sólo actúan sobre su pin asociado.

Los 5 comparadores comparten pines con el puerto A. Si está activada la salida hardware de un comparador, automáticamente se desactiva el bit correspondiente del puerto A, y aunque se escriba en él, no tendrá efecto sobre el pin de salida. En la figura 4.4 se puede observar qué bits del puerto A están asociados a los comparadores.

En este proyecto se utilizan los comparadores conjuntamente con el temporizador para realizar retardos.

Capturadores de entrada

Los capturadores de entrada permiten detectar el instante exacto en el que se ha producido un suceso. Cada capturador está formado por un registro de 16 bits, lógica de detección de flancos y lógica de generación de interrupciones. Cada vez que por la entrada del capturador se detecta un flanco (de subida o bajada según cómo se configure) se almacena en el registro de 16 bits el valor del temporizador principal y se produce una interrupción.

El registro de máscara de interrupciones se encuentra en **TMSK1** (\$1022) y los bits 0,1 y 2 corresponden respectivamente a las interrupciones de los capturadores 3,2 y 1. En el registro de interrupciones **TFLG1** (\$1023) se encuentran los bits que se activan cuando se ha producido alguna captura en alguno de los capturadores. Los bits 0,1 y 2 corresponden a los capturadores 3,2 y 1, igual que en el registro de máscara de interrupciones. La captura se puede programar mediante 2 bits para cada capturador. Estos bits se encuentran en el registro **TCTL2** (\$1021).

La figura 4.10 muestra los registros de los capturadores. Estos registros son el de máscara, el de interrupciones y el que configura los flancos habilitados en el capturador.

TMSK1
DIRECCION \$1022

OC1I	OC2I	OC3I	OC4I	OC5I	IC1I	IC2I	IC3I
------	------	------	------	------	------	------	------

OC1I-OC5I = Permiso de interrupción de cada comparador

IC1I-IC3I = Permiso de interrupción de cada capturador

Registro de máscara de interrupciones.

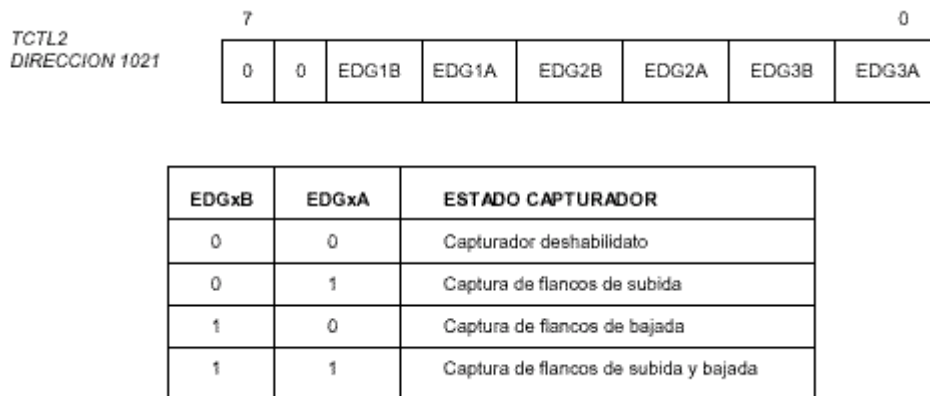
TFLG1
DIRECCION \$1023

OC1F	OC2F	OC3F	OC4F	OC5F	IC1F	IC2F	IC3F
------	------	------	------	------	------	------	------

OC1I-OC5I = Flag de interrupción de cada comparador

IC1I-IC3I = Flag de interrupción de cada capturador

Registro de interrupciones uno.

**Registro TCTL2.****FIGURA 4.10.** Registros de los capturadores.

El microbot depredador desarrollado en este proyecto utiliza un encoder conectado al capturador de entrada uno con captura por flancos de bajada y de subida.

En esta parte del capítulo se ha descrito los recursos del 68HC11 que han sido utilizados en este proyecto. Existen otros recursos que no se han comentado puesto que no son de interés en este proyecto pero que podrían interesar al lector. Por lo tanto para conocer todos los recursos de los que dispone el 68HC11 se puede acudir al Manual del **Microcontrolador MC68HC11**.

4.2.- Programación del microcontrolador MC68HC11.

Una vez que se conocen los recursos del microcontrolador usados para llevar a cabo el microbot depredador, solo queda conocer cómo ejecutar programas sobre el microbot. Pero antes de ejecutar un programa es necesario generar el código que entienda el microcontrolador a partir del que se ha escrito (lenguaje ensamblador, C, etc.).

Tras describir los compiladores utilizados, se hará breve descripción de los programas de comunicación y transferencia de ficheros entre el PC y el 68HC11.

4.2.1.- Compilador As11

Para la compilación de los programas escritos en lenguaje ensamblador se ha utilizado el **As11**. Este compilador actúa bajo entorno DOS y para compilar un programa sólo es necesario escribir: "As11 nombre_fichero.ASM", siendo nombre_fichero.ASM el programa desarrollado en lenguaje ensamblador. El uso de este compilador conlleva ciertas limitaciones, pero esto se explicará en el capítulo dedicado a los problemas surgidos durante el proyecto.

Debido a las deficiencias del As11 se recomienda el uso de otros compiladores de lenguaje ensamblador como el **RAS 0.1.2**.

4.2.2.- Compilador cruzado de C

Para la compilación de programas en C se ha utilizado el compilador cruzado de C **Imagecraft 0.47V**. El estudio de este compilador es algo más complejo que el del As11. Por ello lo que resta del capítulo 4 se dedicará a explicar las características y funcionamiento y adaptación para trabajar con la tarjeta CT6811 de este compilador de C.

4.2.2.1.- ¿Qué es un compilador cruzado de C?

Un compilador cruzado es un compilador que se ejecuta en un ordenador (Por ejemplo un PC) pero que genera código ejecutable para una CPU distinta a la del propio ordenador. El compilador cruzado de C utilizado se ejecuta en un PC pero genera código para el microcontrolador 68HC11.

4.2.2.2.- Lenguajes de alto nivel VS lenguaje ensamblador

Normalmente los microcontroladores se programan en lenguaje ensamblador porque suelen ser programas cortos y además es necesario conocer con un alto grado de detalle lo que ocupa el programa y la velocidad a la que se va a ejecutar. Al emplear un lenguaje de alto nivel se gana en comodidad, abstracción, portabilidad, legibilidad y se necesita menor tiempo de codificación, pero por el contrario se pierde un poco la noción de qué es lo que está pasando dentro del microcontrolador. Además, el volumen de código generado es mucho mayor que el que se hubiese obtenido programando el mismo programa directamente en ensamblador.

Para obtener el mejor rendimiento a los programas, lo ideal es utilizar cada lenguaje para un propósito diferente: El lenguaje ensamblador será necesario utilizarlo cuando se quiera optimizar rutinas en tiempo y espacio. Los lenguajes de alto nivel es mejor emplearlos para la descripción de algoritmos que se deban ejecutar en el micro, ya que su implementación directa en ensamblador puede resultar muy tediosa de codificar y de depurar.

Finalmente, dentro de los lenguajes de alto nivel, el lenguaje C es el más adecuado para la programación de microcontroladores puesto que es un lenguaje que permite estar muy en contacto con la máquina que hay debajo, a la vez que es un lenguaje de alto nivel.

4.2.2.3.- Características del compilador

- **Compilador exclusivo para el 68HC11:** Otros compiladores de C están diseñados para generar código para microcontroladores y microprocesadores muy diversos, lo que complica enormemente su utilización.
- **Fácil llamada a rutinas en ensamblador:** Es posible programar rutinas directamente en ensamblador y llamarlas desde los programas en C.

- **Inserción de intrucciones en ensamblador:** Es posible insertar instrucciones en ensamblador en el propio código en C, utilizando la instrucción ASM.
- **Generación de ficheros en ensamblador:** Este compilador no genera archivos ejecutables directamente, sino que crea archivos en ensamblador que luego se convierten en ejecutables. Estos archivos en ensamblador pueden ser modificados por el usuario lo que permite optimizar el código.
- **Fichero ejecutable en formato .S19:** El fichero ejecutable generado se encuentra en el formato .S19 que es el utilizado por los programas MCBOOT y DOWNMCU para enviar programas a la CT6811.
- **Fácil configuración de la situación del código, pila y datos.**
- **El lenguaje soportado es ANSI C.**

4.2.2.4.- Ficheros que componen el compilador

ICC11.EXE :Programa principal del compilador. Este programa llama a todos los demás.

ICPP.EXE: Preprocesador de C.

ICCOM11.EXE: Compilador de C propiamente dicho. Transforma archivos .C en archivos en ensamblador con extensión .S.

IASM11.EXE: Ensamblador utilizado para convertir los ficheros .S en fichero .S19.

CRT.S: Fichero de Arranque y configuración.

4.2.2.3.- Funcionamiento del compilador



FIGURA 4.11. Funcionamiento del compilador de C.

El funcionamiento del compilador es como el indicado en la figura 4.11. Primero se toma un archivo en C, se compila y se obtiene un archivo .S en ensamblador. Este archivo se vuelve a compilar utilizando el ensamblador y se crea un ejecutable con extensión .S19.

En la figura 4.12 se muestra un ejemplo de cómo se compilaría un programa llamado prueba.c. Utilizando el ejecutable ICC11 con el parámetro -S se compila el programa y se genera el fichero PRUEBA.S en ensamblador. Este programa se ensambla utilizando el ensamblador IAS11.

```
C:\6811\ICC\PRUEBA>ias11 prueba.s

C:\6811\ICC\PRUEBA>dir

Volumen en unidad C es GRUPO JJ
Número de serie de volumen es 395D-1CD5
Directorio de C:\6811\ICC\PRUEBA

.           <DIR>      04-20-97   8:22a
..          <DIR>      04-20-97   8:22a
PRUEBA     C           702 04-16-97  11:19a
PRUEBA     S19          36 04-20-97   8:23a
PRUEBA     S           59 04-20-97   8:23a
           5 archivo(s)          797 bytes
           739377152 bytes libres

C:\6811\ICC\PRUEBA>
```

FIGURA 4.12. Ejemplo de compilación del programa PRUEBA.C.

El parámetro -S se utiliza para que el compilador sólo genere un fichero con extensión .s. En la figura 4.13 se muestra otra forma de compilar. En este caso sólo es necesario llamar al programa ICC11, el cual se encarga de llamar al resto de ejecutables (preprocesador, compilador y ensamblador). Se genera directamente el archivo ejecutable PRUEBA.S19. Al utilizar esta segunda opción se incluye en el ejecutable el archivo CRT.S que es un archivo en ensamblador que contiene el arranque y configuración del sistema. Al llamar al programa ICC11 se utiliza el parámetro -o prueba para que se genere el archivo PRUEBA.S19. La primera forma de compilar es la forma más utilizada.

```
C:\6811\ICC\PRUEBA>icc11 prueba.c -o prueba

C:\6811\ICC\PRUEBA>dir

Volumen en unidad C es GRUPO JJ
Número de serie de volumen es 395D-1CD5
Directorio de C:\6811\ICC\PRUEBA

.           <DIR>      04-20-97   8:22a
..          <DIR>      04-20-97   8:22a
PRUEBA     C           702 04-16-97  11:19a
PRUEBA     S19          492 04-20-97   8:32a
PRUEBA     S           59 04-20-97   8:32a
CRT        S          1480 07-06-96   6:41a
           6 archivo(s)          2733 bytes
           735969280 bytes libres

C:\6811\ICC\PRUEBA>
```

FIGURA 4.13. Otra forma de compilar el programa PRUEBA.C.

4.2.2.5.- Interfaz entre el C y el lenguaje ensamblador.

Llamadas a funciones en ensamblador

Es posible llamar a funciones realizadas en ensamblador directamente desde los programas en C. La única condición que se debe cumplir es que las etiquetas que definen el comienzo de una función en ensamblador deben comenzar con el carácter `_`.

En la figura 4.14a se muestra un programa en C que simplemente llama a la función `ledon()` para encender un led. La función `ledon` se ha implementado en ensamblador y su código se puede ver en la figura 4.14b. Obsérvese que en el programa fuente en C no ha hecho falta especificar que la función `ledon` es externa. Al compilar el programa en C se generará un salto a la etiqueta `_ledon`. Por ello al implementar esta función en ensamblador deberemos especificar la etiqueta de entrada como `_ledon`, como se muestra en la figura.

Si se definiese una función con parámetros de entrada la cosa sería algo más complicada.

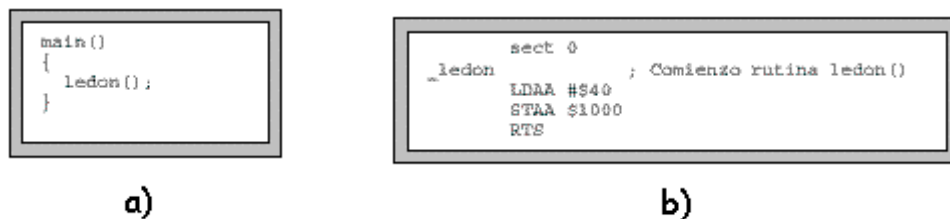


FIGURA 4.14. Programa en C e implementación de una función en ensamblador.

Inclusión de instrucciones en ensamblador en el código en C

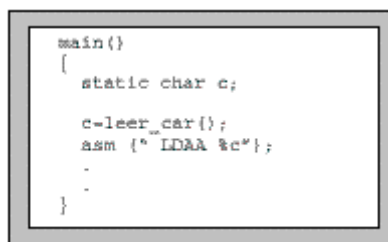
Algunas veces son necesarias ciertas instrucciones de control del 68HC11 que se encuentran en ensamblador pero que no se definen en el compilador de C como por ejemplo `RTI`, `WAIT`, `SWI`... Para poder incluir estas instrucciones en nuestro código en C el compilador dispone de la instrucción `ASM` que permite introducir una instrucción directamente en ensamblador. La sintaxis de esta instrucción es `ASM(" línea en ensamblador")`. Esta instrucción lo que hace es pasar directamente al ensamblador la cadena enviada. La cadena que se pasa como argumento a la instrucción `ASM` puede ser incorrecta. El compilador de C no la analizará. Simplemente la copiará tal cual en el programa en ensamblador. Será el ensamblador en la etapa de ensamblado el que determinará si es o no correcta.

Hay una serie de reglas importantes a la hora de utilizar la instrucción `ASM`:

1. Si se escribe una instrucción en ensamblador hay que colocar por lo menos un espacio al comienzo de la cadena, para indicar al ensamblador que es una instrucción y no una etiqueta. Ejemplos:

- **Correcto:** `asm ("RTI");`
 - **Incorrecto:** `asm ("RTT");`
2. Las etiquetas se deben escribir justo a la izquierda de la cadena, sin espacios intermedios:
- **Correcto:** `asm ("bucle LDAA #$40");`
 - **Incorrecto:** `asm (" bucle LDAA #$40");`

Es posible hacer referencia a variables en C desde la instrucción `asm`. Para ello utilizamos el carácter `%`. En la figura 4.15 se muestra un ejemplo muy simple. Queremos cargar en el acumulador A el valor de la variable `c`. La instrucción **`asm("LDAA %c")`** lo soluciona. El carácter `%` hace que se le pase al ensamblador la etiqueta asociada a la variable situada a continuación del carácter `%`, en vez del identificador de la propia variable.



```
main()
{
    static char c;

    c=leer_car();
    asm {" LDAA %c";
        -
    }
}
```

FIGURA 4.15. Ejemplo de carga de la variable `c` en el acumulador A.

4.2.2.6.- Rutinas de servicio de interrupción

A la hora de hacer rutinas de servicio de interrupción hay que tener en cuenta los tres puntos siguientes:

1. Cambio del vector de interrupción.
2. Activación de las interrupciones.
3. Codificación de la rutina de servicio de interrupción.

En la figura 4.16 se muestra el modelo a seguir para la puesta en funcionamiento de las rutinas de servicio de interrupción.

```
/* Vector de interrupción */
#define vint (*(unsigned int *)0xFFD6)

/* Rutina de servicio de interrupción */
void rsi()
{
    -
    -
    -
    asm (" RTI");
}

main()
{
    /* Cambiar vector de interrupción */
    vint = (unsigned int)rsi;

    /* Activar las interrupciones */
    asm (" CLI");
    -
    -
    -
}
```

FIGURA 4.16. Programa modelo de una rutina de servicio de interrupción.

Cambio del vector de interrupción

En el modelo de la figura 4.16 se ha modificado el vector de interrupción de la dirección 0xFFD6 para que apunte a la rutina de servicio de interrupción *rsi()*. Primero se define la situación del vector de interrupción por medio de una instrucción *define*. Lo que estamos indicando al compilador es que *vint* es una constante que representa un valor entero que debe estar situado en la dirección 0xFFD6. Para hacer que este valor apunte a la rutina que nosotros queramos simplemente se realiza la asignación *vint = (unsigned int)rsi*; que introduce la dirección de comienzo de la función *rsi()* en la dirección 0xFFD6.

Activación de las interrupciones

Una vez que se ha cambiado el vector de interrupción es necesario activar las interrupciones. No existe ninguna instrucción en el estándar ANSI C que permita hacer eso, por lo que se recurre a escribir directamente la instrucción en ensamblador mediante la instrucción *asm* (" CLI"). Si se quieren inhibir las interrupciones bastará con escribir *asm*(" SEI");

Rutina de servicio de la interrupción

Las rutinas de atención a las interrupciones deben terminar con la instrucción RTI, en lugar de la instrucción RTS que es la utilizada para retornar de una subrutina. En el estándar ANSI C no contempla ninguna palabra reservada para indicar al compilador que una función en C es una rutina de servicio de interrupción y que por tanto debe finalizar con la instrucción RTI en lugar de RTS. Por ello es necesario introducir directamente la instrucción RTI al final de la rutina de servicio de interrupción por medio de la instrucción *asm* (" RTI"). Pero esto tiene problemas. El compilador puede haber generado un código que utilice la pila. Antes de salir de la rutina

de interrupción habrá que vaciar por completo la pila o de lo contrario llegará un momento que se desbordará.

Para solucionar esto se utiliza el truco de la **subrutina de servicio de interrupción**. Para cada interrupción programada se definen dos funciones. Una función será la rutina de servicio de interrupción y la otra será la subrutina de servicio de interrupción. El nuevo vector de interrupción se modificará para que apunte a la rutina de servicio de interrupción, la cual debe acabar con la instrucción **asm ("RTI")**. En la subrutina de servicio de interrupción se introduce el código necesario para atender a la interrupción. En la figura 4.17 se muestra un esquema de funcionamiento:

```
void ssi()
/* Subrutina de servicio de interrupción */
{
    /* Código */
    ...
    ...
}

void rsi()
/* Rutina de servicio de interrupción */
{
    ssi();           /* Llamar a la subrutina de servicio de interrupción */
    asm ("RTI");     /* Retorno de interrupción */
}

main()
{
    /* Cambiar el vector de interrupción para que apunte a rsi() */
    ...
    ...
}
```

FIGURA 4.17. Esquema de funcionamiento de la rutina de interrupción.

4.2.3.- Programación de la CT6811 en modo entrenador

En este apartado se va a ver la configuración del compilador de C para ejecutar programas en la RAM externa.

4.2.3.1.- Introducción

El objetivo es configurar el compilador de C para que funcione con la RAM externa de la CT256. Lo que se quiere es crear programas en C para ser probados en la RAM externa, es decir, en el modo entrenador puesto que la RAM de la que se dispone es volátil.

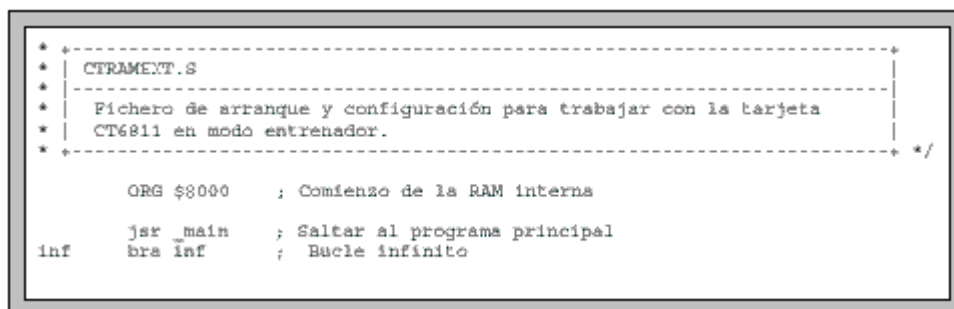
Al trabajar en modo entrenador lo que nos interesa son los 32 K de RAM externa, situados desde la dirección \$8000 hasta la \$FFFF, y los registros de control. Para poder utilizar el compilador de C tendremos que configurarlo para que se genere código para este mapa de memoria. El código debe empezar en la dirección \$8000 y no puede sobrepasar la dirección \$FFFF, o de lo contrario desbordaremos la memoria externa.

4.2.3.2.- Configuración del compilador

El fichero de arranque y configuración CTRAMEXT.S

El fichero de arranque y configuración empleado se muestra en la figura 4.18. Se ha llamado **CTRAMEXT.S** para indicar que es para generar código y datos para la RAM externa.

En este archivo se sitúa el comienzo de todos los programas a partir de la dirección \$8000 que es donde empieza la RAM externa. Este programa realiza un salto a la etiqueta **_main** que es el nombre que recibe la etiqueta de comienzo del programa principal. Si se retorna del programa principal, se entra en un bucle infinito para que el 68HC11 no ejecute instrucciones al azar.



```
*
*  CTRAMEXT.S
*
*  Fichero de arranque y configuración para trabajar con la tarjeta
*  CT6811 en modo entrenador.
*
*
      ORG $8000      ; Comienzo de la RAM interna
      jsr _main      ; Saltar al programa principal
inf   bra inf        ; Bucle infinito
```

FIGURA 4.18. Fichero de arranque y configuración para trabajar en modo entrenador.

Para que la CT6811 funcione en modo entrenador, el 68HC11 debe arrancar en el modo BOOTSTRAP, con lo que la pila ya se encuentra inicializada en la dirección \$FF y no será necesario inicializarla en el archivo de arranque y configuración.

En el fichero de arranque y configuración no se han definido secciones. Esto es debido a que el código, los datos y la pila se deben situar dentro de la misma sección. Un programa de estas características es lo que se denomina **programa unificado**.

Unificación de programas en ensamblador

El código en ensamblador generado por el compilador de C no se encuentra unificado. Es decir, que las variables se sitúan en la sección 1 y el código en la sección 0. Para solucionar esto se utiliza el programa **UNIFICA.EXE**. Este programa simplemente toma un fichero en ensamblador y genera otro fichero igual que el original pero en el que se han eliminado todas las secciones.

Será con los programas unificados con los que se trabajará cuando se quiera utilizar la CT6811 en modo entrenador.

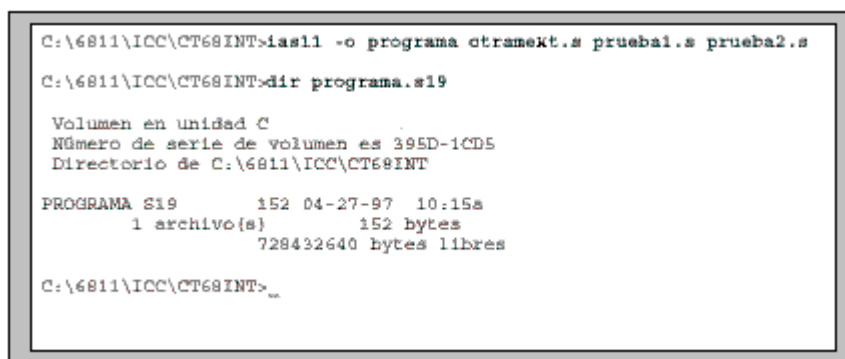
Situación de las variables al comienzo del fichero en ensamblador

No se pueden ensamblar los archivos unificados junto con el archivo de arranque y configuración para generar un ejecutable con extensión .S19 que se pueda cargar en la RAM externa. Es necesario realizar una transformación más al archivo en ensamblador. El compilador sitúa todas las variables estáticas al final del fichero en ensamblador.

El programa **GETVARS.EXE** sirve para ‘extraer’ todas las variables de un fichero en ensamblador.

Generación de archivos ejecutables .S19

Una vez que se tienen todos los ficheros .S y se quiere obtener el fichero ejecutable con extensión .S19, se utilizará el ensamblador **IAS11.EXE** según se muestra en la figura 4.19. En ese ejemplo se genera el fichero PROGRAMA.s19 a partir de los ficheros en ensamblador *prueba1.s*, *prueba2.s* y el fichero de arranque y configuración **CTRAMEXT.S**. El parámetro *-o programa* indica al ensamblador que se genera un archivo ejecutable con nombre *programa.s19*. Si no se especifica este parámetro el ensamblador generará el fichero ejecutable con el nombre del primer archivo .S que se haya pasado como argumento.



```
C:\6811\ICC\CT68INT>ias11 -o programa ctramext.s prueba1.s prueba2.s
C:\6811\ICC\CT68INT>dir programa.s19

Volumen en unidad C:
Número de serie de volumen es 395D-1CD5
Directorio de C:\6811\ICC\CT68INT

PROGRAMA S19      152 04-27-97  10:15a
    1 archivo(s)    152 bytes
                728432640 bytes libres

C:\6811\ICC\CT68INT>
```

FIGURA 4.19. Generación del fichero ejecutable programa .s19.

En la figura 4.20 se muestra un esquema con todos los pasos que hay que realizar aplicados al programa *ledp.c* para generar el fichero ejecutable *ledp.s19*.

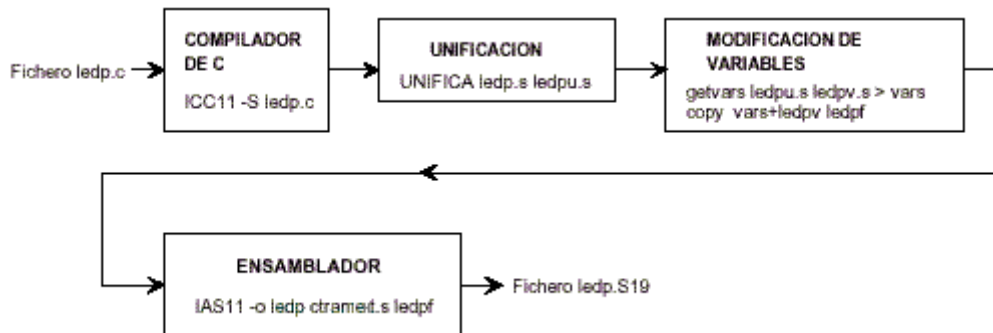


FIGURA 4.20. Pasos a realizar para crear el fichero ledp.s19 a partir del ledp.c.

El archivo CINTE.BAT

Este archivo se utiliza para compilar programas en C para la RAM externa de forma automática.

```

@REM
@REM ----- Compilar de .c a .s -----
@REM
@icc11 -S %1.c
@REM
@REM ----- Generar fichero en ensamblador unificado -----
@REM
@unifica %1.a temp.s
@REM
@REM ----- Eliminación de las variables RMB -----
@REM
@getvars temp.s temp2.s > %1.var
@REM
@REM ----- Generación del fichero .s final -----
@REM
@copy %1.var+temp2.s %1.s > NUL
@REM
@REM ----- Borrar ficheros temporales -----
@del temp.s > NUL
@del temp2.s > NUL
@REM
@REM ----- Generar fichero .S19 -----
@REM
@ias11 -o %1.ctrame.s %1.s
  
```

FIGURA 4.21. Fichero CINTE.BAT para la compilación automática de programas .C

Con este archivo es muy fácil generar archivos ejecutables .S19 directamente ya que solo ejecutando la línea de comandos *> cinte programa* se compila el programa en C que será almacenado en RAM externa posteriormente.

4.2.4.- Programas de comunicación y transferencia de ficheros

Una vez que se obtiene a través del compilador el fichero ejecutable con extensión .s19, solo resta cargar ese programa en memoria RAM o EEPROM y que el microcontrolador lo ejecute.

Existen varios programas de comunicación y transferencia de ficheros entre PC y 68HC11: pbug, ctdialog, etc., pero en este apartado solo se va a introducir los programas DOWNMCU, CTDIALOG y MCBOOT2.

4.2.4.1.- DOWNMCU

Este programa permite transferir ficheros del PC a la memoria RAM interna del 68HC11. Para realizar la transferencia se debe ejecutar la siguiente linea:

➤ *downmcu programa -com1*

Programa es el fichero ya compilado (programa.s19) y *-com1* indica el puerto serie (si la comunicación se realiza a través del puerto serie com2 habrá que cambiar *-com1* por *-com2*).

4.2.4.2.- CTDIALOG

Este programa permite ‘dialogar’ con el 68HC11, con la que se puede acceder a todos los recursos del microcontrolador desde el PC, como la memoria EEPROM, los registros, etc.

La ejecución de este programa se divide en dos partes:

- 1- Ejecutar el programa servidor CTSERVER en el 68HC11. Para transferir este programa se ha utilizado el DOWNMCU.
- 2- Ejecutar el CTDIALOG sobre el PC.

Una vez realizados estos dos pasos, existe una comunicación entre el PC y el 68HC11, y se pueden ver y modificar todos los recursos del microcontrolador mediante una serie de comandos. Algunos de esos comandos son:

CKC. Comprueba que la conexión con el programa CTSERVER es correcta.

MS. Permite almacenar un valor en una posición de memoria.

EEPROM. Permite grabar un programa en la memoria EEPROM del 68HC11 (**Nota:** Para la correcta grabación debe ponerse como dirección de inicio del programa la \$B600, que es la posición de memoria donde comienza la EEPROM).

G. Realiza un salto a la dirección especificada. Por ejemplo, si en el CTDIALOG se escribe *g b600* se ejecuta lo que haya en la EEPROM.

DASM. Desensambla el bloque de memoria a partir de la dirección especificada.

En la figura 4.22 se puede ver la ejecución del programa CTDIALOG.

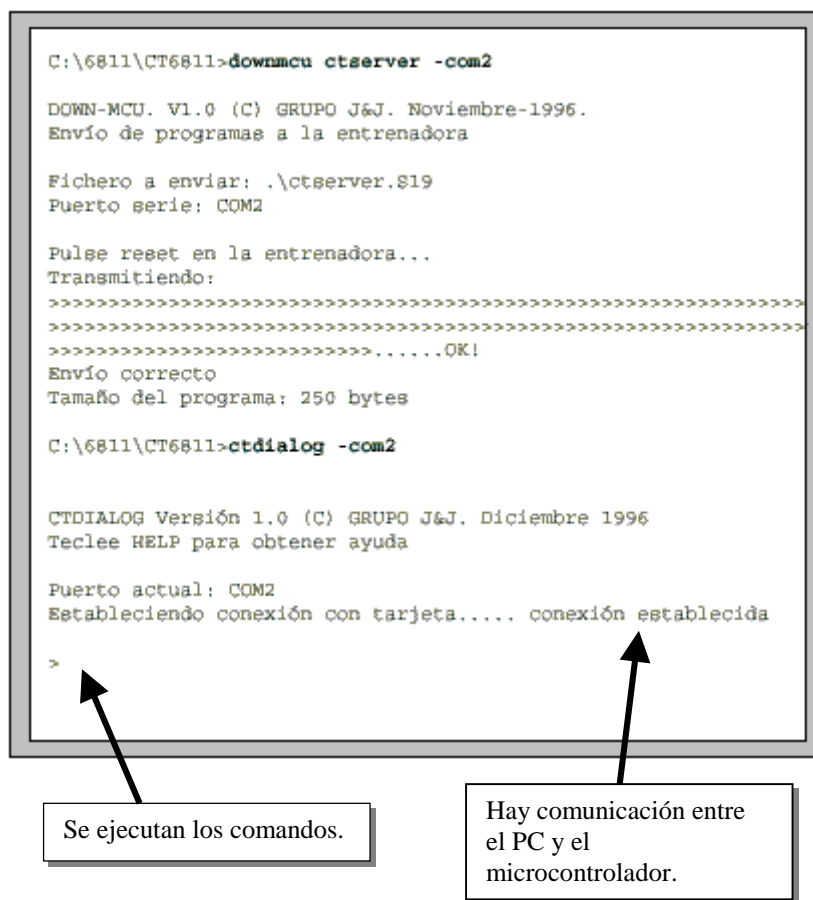


FIGURA 4.22. Ejecución del programa CTDIALOG.

4.2.4.3. MCBOOT2

El MCBOOT2 es un programa de comunicaciones para comunicarse con la tarjeta CT6811. Además de funcionar como terminal de comunicaciones también permite cargar programas, utilizar la pantalla del PC para representar datos y el teclado del PC para enviar datos al 68HC11.

Es un programa bastante sencillo de utilizar proporcionando un interfaz bastante intuitivo. Para ejecutar un programa solo hay que seleccionar la configuración (DTR OFF, modo expanded, etc.).

Su principal inconveniente es que no ejecuta correctamente programas que hagan uso de interrupciones.

4.2.4.4. Carga y ejecución de programas en modo entrenador.

En este proyecto se han desarrollado dos tipos de programas: uno en lenguaje ensamblador grabado en memoria EEPROM y el otro en lenguaje C que se carga en la RAM externa. Por lo tanto ahora se van a describir los pasos seguidos para cargar y ejecutar estos programas a través del CTDIALOG.

Carga y ejecución del programa en EEPROM

Una vez establecida la comunicación entre el PC y el 68HC11, se graba el programa en EEPROM mediante el comando *eeprom* y se ejecuta con el comando *g* (Ver figura 4.23).

Una vez grabado en programa en EEPROM también se puede ejecutar cambiando el JP5 de la CT6811 a modo autónomo. Mientras se esté en modo autónomo se ejecuta el programa grabado en EEPROM cada vez que se pulse RESET y no se puede establecer comunicación entre PC y microcontrolador hasta que se vuelva a modo entrenador.

Carga y ejecución del programa en RAM externa

Una vez establecida la comunicación entre PC y 68HC11, se configura el micro en modo expanded por software, cargando un \$F5 en el registro HPRIO. Después se carga el fichero mediante el comando *load* y, por último, se ejecuta mediante un salto a la dirección de comienzo de la RAM externa (Ver figura 4.24).

Al ser esta RAM volátil, en el momento en el que se pierda la alimentación o se termine el programa, habrá que volver a realizar la comunicación con el CTDIALOG y repetir los pasos anteriormente mencionados.

```
CYDIALOG Versión 1.0 (C) GRUPO J&J, Diciembre 1996  
Teclas WELD para obtener ayuda
```

Puerto actual: COM2
Estableciendo conexión con tarjeta..... conexión establecida

> eeprom guardia

Fichero a grabar en memoria EEPROM: GUARDIA.S19
Transmitiendo:->>>>>>>>>>>>>>>>
Grabación terminada
Número bytes grabados: 491

> g b600

Conexion perdida

* quit

Programa terminado

FIGURA 4.23. Grabación y ejecución de **guardia.asm** en EEPROM.

```
CDTIALQG Versión 1.0 (C) GRUPO J&J, Diciembre 1996  
Teclas WHELP para obtener ayuda  
  
Puerto actual: COM2  
Estableciendo conexión con tarjeta..... conexión establecida  
  
> ms 103c f5  
  
> load menu  
Transmitiendo .....>>>>>>>>>>>>>>>>>>  
  
Envío correcto  
Tamaño del programa: 4910 bytes  
> g 8000  
Conexión perdida  
  
*, quit  
Programa terminado
```

FIGURA 4.24. Carga y ejecución de **menu.c** en la RAM externa.

Guardia.asm y menu.c son los programas de ensamblador y C desarrollados en este proyecto para simular el comportamiento de un depredador mediante el microbot TRITTON.

CAPITULO 5. MICROBOT DEPREDADOR

5. MICROBOT DEPRADOR

5.1.- Introducción

El objetivo de este proyecto es simular el comportamiento de un depredador. Para lograr cumplir este objetivo es necesario que el microbot recoja información del entorno con el objeto de averiguar la posición de la presa y de los posibles obstáculos que pueda encontrarse en su camino hacia ésta. La figura 5.1 muestra el esquema de bloques del microbot depredador.

En una primera parte de este capítulo se analizan los distintos tipos de sensores que se podrían utilizar para la detección de la presa. Asimismo, se explican las razones por las que se han elegido los sensores de infrarrojos. Tras esto, se hará una descripción de los sensores de infrarrojos utilizados en este proyecto y un análisis de los circuitos implementados para la correcta detección de la presa.

En la segunda parte del capítulo se realizará una descripción de los sensores y circuitos utilizados para la detección de los obstáculos, seguida del estudio de la disposición y conexión de los sensores al microbot.

Para terminar con este capítulo se describirá la alimentación del microbot depredador.

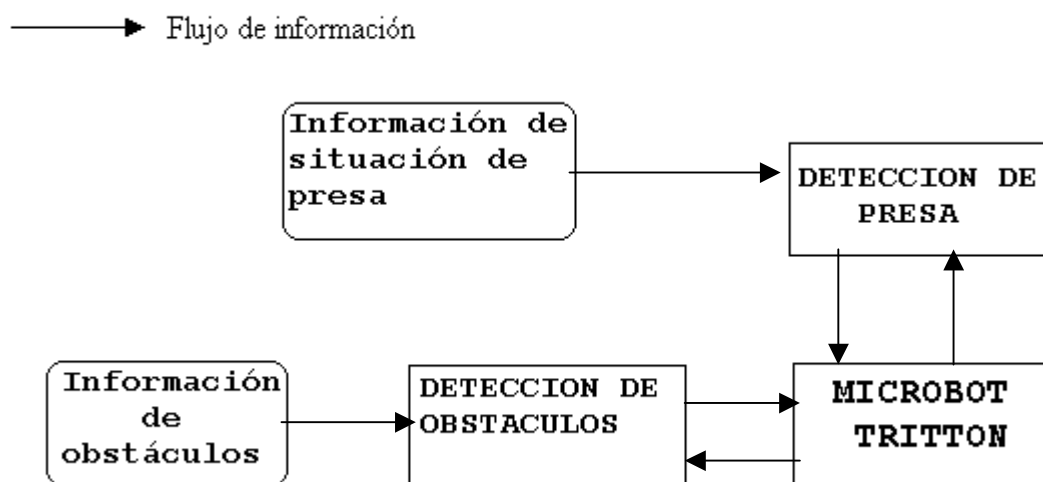


FIGURA 5.1. Esquema general del microbot depredador.

5.2.- Detección de presa

Para poder localizar a la presa, el microbot debe incorporar una serie de sensores que le permitan llevar a cabo dicha localización. Dependiendo del tipo de sensor, la localización de la presa se hará de una u otra forma y, tanto el hardware adicional, para adaptar el sensor al microbot, como la programación, tendrán una mayor o menor complejidad.

El estudio de los tipos de sensores que existen para realizar la detección de la presa es misión del siguiente apartado, en el que también se explican las razones por las que se ha optado por los sensores de infrarrojos.

5.2.1.- Elección de sensores

La elección de los sensores a utilizar para conocer la posición de la presa se ha realizado atendiendo a la condición: “ **El menor precio, siempre que cumpla con los objetivos planteados**”.

Los sensores ideales para la detección de la presa deberían cumplir los siguientes requisitos:

- Bajo coste.
- Pequeño tamaño y peso para poder incorporarlo al microbot.
- Poder atravesar obstáculos.
- Presa sin sensor que emita una señal al microbot depredador.

Teniendo en cuenta la condición anterior, se han estudiado varios tipos de sensores entre los que se han destacado los siguientes: Radiofrecuencia, Ultrasonidos e Infrarrojos.

5.2.1.1.- Sensores de Radiofrecuencia (RF)

Características generales

Una de las opciones planteadas para la detección de la presa es utilizar ondas de radio.

- Las ondas de radio son emitidas en todas direcciones y atraviesan los posibles obstáculos que haya entre emisor y receptor.
- Existe un amplio rango de frecuencias para los sensores de radiofrecuencia (RF) aunque son necesarias licencias, puesto que podrían existir interferencias debido a las emisiones de otros emisores de radio.

Sensores de RF para microbot depredador

Existen multitud de sensores de RF que se podrían utilizar para realizar este proyecto, pero puesto que la distancia a la que se puede detectar la presa no tiene que ser mucha (de 3 a 8 metros aproximadamente), basta con usar sensores de RF para control remoto, los cuales son de elevada frecuencia y sus precios son relativamente bajos.

Un ejemplo de los sensores de RF que se podrían utilizar para localizar a la presa son: el transmisor **TWS-434** y el receptor **RWS-434**. Su precio es bajo, costando cada módulo unas 2.000 ptas.(12 euros).

Estos serían los sensores ideales, si no fuera por:

1. El transmisor debe ir incorporado a la presa para que el microbot pueda detectarla.
2. Como se ha dicho antes, los emisores y receptores de RF emiten y reciben en todas direcciones. Por lo tanto, el microbot no sería capaz de averiguar de qué dirección viene la emisión y no podría localizar a la presa.

La única forma que se ha encontrado de conocer la dirección en la que se encuentra la presa es mediante antenas directivas. Las antenas directivas son utilizadas por las televisiones por satélite y son directivas, recibiendo señal sólo desde la dirección hacia la que está orientada. El problema es que no se ha encontrado una antena de las dimensiones adecuadas para el microbot y, si se hubiera encontrado, habría encarecido el conjunto considerablemente, puesto que a la antena directiva habría que sumarle un motor que le permitiera girar y actuar a modo de radar.

5.2.1.2.- Sensores de Ultrasonido

Características generales

La siguiente opción considerada es utilizar sensores de ultrasonidos. Estos emiten pulsos a frecuencias mayores que las audibles (20 KHz). Existen dos tipos de sensores de ultrasonidos:

- a) Sensores que envían una señal de sonido de baja frecuencia, esperan un tiempo a que la señal regrese, y cuando la detectan, informan al microcontrolador.
- b) En estos sensores es el microcontrolador el encargado de decidir cuándo se envía la señal ultrasónica. El microcontrolador también es el encargado de contar el tiempo hasta que recibe la señal reflejada del objeto.

Entre las características de los sensores de ultrasonidos cabe destacar:

- Los ultrasonidos no atraviesan objetos, quedando limitada su acción a una habitación.

- No es inmune a interferencias. Cualquier ruido a las frecuencias a las que trabaja el sensor puede producir interferencias (ruido de monedas, llaves, etc.).
- Son directivos.
- Dependiendo de la rugosidad del objeto se recibe un eco distinto. Aunque puede no recibirse eco o recibirse de la dirección errónea.

Sensores de Ultrasonido para microbot depredador

Los sensores de ultrasonido serían una buena opción puesto que, aunque no atraviesan obstáculos, no es necesario que la presa incorpore un sensor que mande información de su posición al microbot.

Un ejemplo de estos sensores de ultrasonidos serían los conocidos por todos **Polaroid Sonar**. Estos sensores los fabrica la empresa POLAROID para sus cámaras, pero se pueden utilizar también para aplicaciones con robots. Estos sensores se adquieren a un precio que ronda las 10.000 ptas.(60 euros), que como se verá más adelante, es un precio bastante superior al de los sensores infrarrojos modulados.

5.2.1.3.- Sensores de Infrarrojos (IR)

Características generales

Las principales características de estos sensores son:

- Al igual que los ultrasonidos no atraviesan objetos, por lo que no puede existir obstáculos entre emisor y receptor.
- Existen varias fuentes de interferencias, como luz solar, lámparas interiores, emisiones electromagnéticas, etc.
- Son directivos, dependiendo del ángulo de radiación y de ángulo de sensibilidad.

Sensores de infrarrojos para microbot depredador

Según lo visto en las características de los sensores de infrarrojos (IR), estos serían la peor opción de las tres propuestas, ya que la luz infrarroja no atraviesa objetos, y además la presa debería incorporar un emisor de infrarrojos para que el microbot pudiera localizarla. Entonces, ¿por qué utilizar estos sensores para la detección de la presa? La respuesta a esa pregunta se debe buscar en la condición que se impuso al inicio de esta sección “ El menor precio, siempre que cumpla con los objetivos planteados”.

Sensores elegidos: Los sensores de IR han sido los elegidos para implementar la búsqueda de la presa debido a su reducido precio en comparación con los demás sensores. Por otro lado, el objetivo principal de este proyecto es la localización de la presa y los sensores de IR cumplen con este objetivo, puesto que permiten saber la dirección en la que se encuentra la presa.

5.2.2.- Estudio de la luz infrarroja

La luz infrarroja es una forma de radiación cuyo espectro se sitúa entre los 800 nanómetros (nm) y 1 milímetro (mm). Aunque el espectro de la luz IR es muy amplio, los sensores IR se sitúan en la zona del espectro más cercana a la luz visible, es decir, entre los 800nm y los 1100nm, aproximadamente.

Ya que los sensores de IR son los elegidos para la detección de la presa, es necesario realizar un estudio de las características de la luz infrarroja que interesa conocer, para saber interpretar las hojas de características de los sensores de IR y hacer la mejor elección para este proyecto. Asimismo, también se considera necesario conocer las posibles fuentes de interferencias que podrían afectar a los sensores.

5.2.2.1.- Características de la luz IR

Hay que distinguir entre las características asociadas a los emisores de IR y las que se asocian con los detectores de IR. Entre las características de la luz IR cabe destacar:

- **Potencia radiante(Φ_e).** Su unidad es el vatio. Se define como la potencia total dada en forma de radiación.
- **Angulo sólido.** Su unidad es el esteradian (sr) y se utiliza para medida de ángulos en tres dimensiones.

Emisor

- **Intensidad radiante(I_e).** Su unidad es el vatio/sr. Se define como la potencia radiante por ángulo sólido. Es uno de los factores que determinan la máxima distancia posible entre emisor y detector de IR.
- **Radiancia(L_e).** Su unidad es el vatio/sr m^2 . Se define como la potencia radiante incidente por ángulo sólido.
- **Angulo de radiación.** Muestra el lóbulo de radiación IR emitida por el sensor.

Detector

- **Irradiancia(E_e).** Se mide en vatios/ m^2 y es la radiación incidente por área de superficie. Este es el factor que determina la sensibilidad del detector.
- **Espectro de sensibilidad.** Muestra la distancia de recepción relativa respecto al ángulo de desplazamiento en el que se encuentre el emisor.

La figura 5.2 muestra una tabla con algunos de los parámetros mencionados pero aplicados en términos fotométricos en lugar de radiométricos. Se ha considerado necesario insertar esta tabla, puesto que las hojas de características de los sensores de IR suelen incorporar referencias a los parámetros en términos fotométricos.


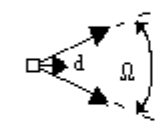
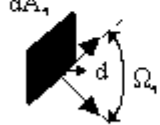

Términos fotométricos			
	Término	Símbolo	Unidad
	Flujo Luminoso	ϕ_v	lm Lumen
	Intensidad Luminosa	I_v	$\frac{\text{lm}}{\text{sr}}$ Candela
	Luminancia	L_v	$\frac{\text{cd}}{\text{cm}^2}$ Stilb
	Iluminancia	E_v	$\frac{\text{lm}}{\text{m}^2}$ Lux

FIGURA 5.2. Tabla de parámetros de la luz IR.

De los parámetros relacionados con la luz IR tienen especial importancia la **Irradiancia** (también llamada Iluminancia) y la **Intensidad radiante** (también llamada Intensidad luminosa), ya que son los parámetros que determinan la distancia de transmisión de una señal infrarroja, de tal forma que la distancia de transmisión queda definida como sigue:

$$\text{Irradiancia Incidente (mW / cm}^2\text{)} = \frac{\text{Intensidad Irradiada (mW / sr)}}{\text{Distancia de transmisión al cuadrado (cm}^2\text{)}}$$

Según esta fórmula, si la sensibilidad mínima del detector es de $4 \mu\text{W/cm}^2$ y la potencia de emisión del emisor es de 40 mW/sr , la distancia de transmisión podría llegar a 1 metro.

Otros dos parámetros fundamentales para la adecuada elección de los sensores es el espectro de sensibilidad y el ángulo de radiación. El primero determina el ángulo desde el cual el receptor es capaz de detectar las emisiones del emisor de IR. El segundo de estos dos parámetros determina el haz de radiación del emisor.

En la figura 5.3 se muestra la variación de la intensidad luminosa (I_v) cuando el emisor de IR tiene un ángulo de radiación (α_2) u otro. Como se puede apreciar en esta

figura, la intensidad luminosa que recibe el detector de IR depende de si el emisor está más o menos enfocado sobre el eje óptico ($\alpha_1 = 0^\circ$).

Por otra parte, la figura 5.4 representa el espectro de sensibilidad del fotodiodo detector de IR SFH203FA. Se puede apreciar como a medida que nos alejamos del eje 0° , la sensibilidad del detector disminuye.

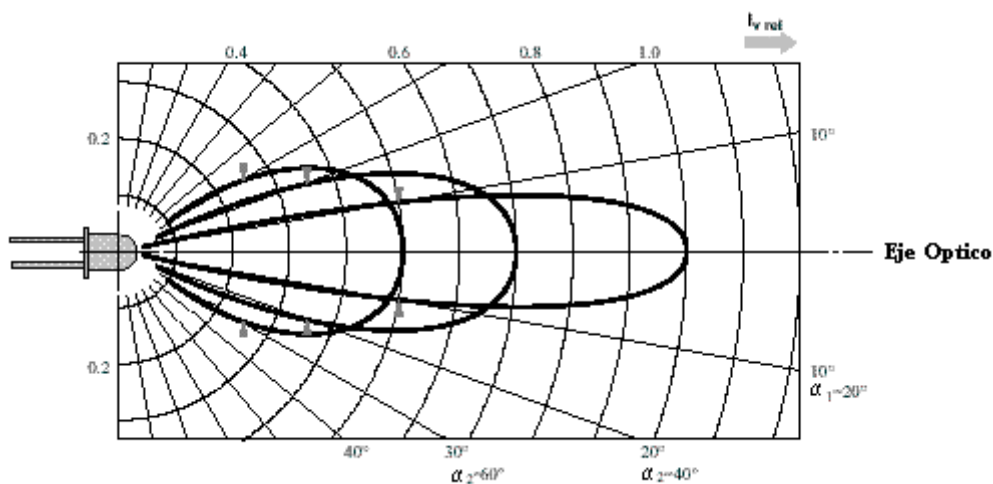


FIGURA 5.3. Angulos de radiación & Intensidad luminosa.

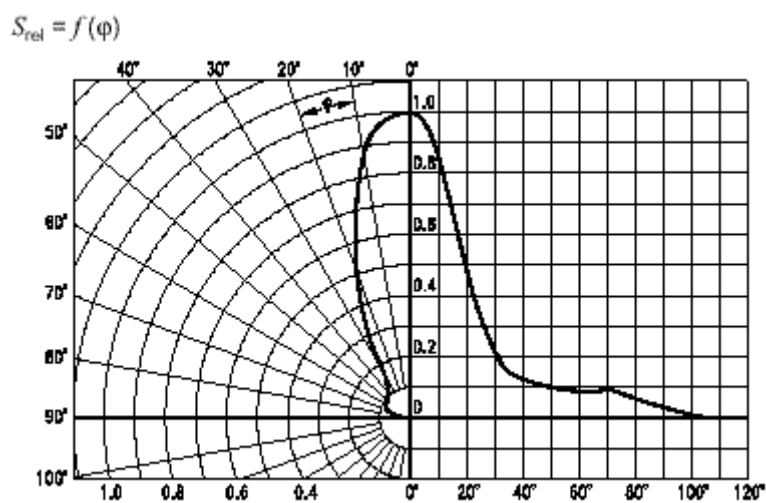


FIGURA 5.4. Espectro de sensibilidad del fotodiodo detector de IR SFH203FA.

5.2.2.2.-Fuentes de interferencias

Existen varias posibles fuentes de interferencias que afectan a los detectores de infrarrojos, pero entre todas ellas destacan dos: luz solar y lámparas fluorescentes.

Luz solar

El sol emite todo tipo de radiación electromagnética incluyendo infrarrojos. De hecho, es la fuente más intensa de luz infrarroja que existe.

La luz solar es una fuente de interferencia DC que puede forzar una corriente continua en el detector de IR, pudiendo llegar a saturar a este receptor. La influencia de la luz solar se puede limitar usando un filtro óptico, que elimina todas las emisiones que se encuentre fuera de una determinada longitud de onda. En la figura 5.5 se puede observar que la luz solar emite en las longitudes de onda del detector de IR con filtro óptico (830nm – 1100nm), pero estas emisiones son menores que las que se producen en otras longitudes de onda como la que abarca la luz visible (400nm – 700nm).

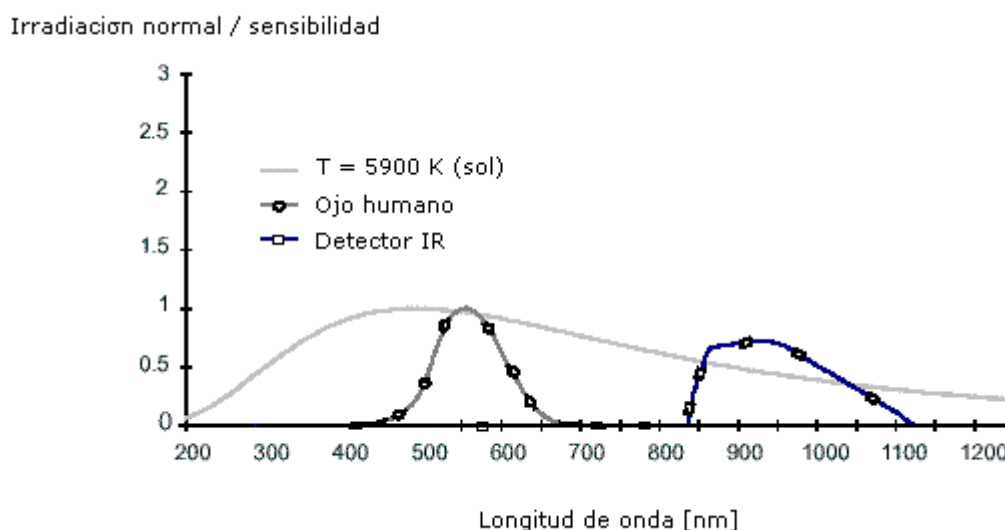


FIGURA 5.5. Emisión espectral y Sensibilidad espectral.

Lámparas fluorescentes

Estas lámparas sólo emiten una pequeña radiación en el espectro infrarrojo. Esta radiación es debida a las componentes a alta frecuencia de la lámpara.

La figura 5.6 muestra el espectro de emisión de luz infrarroja de las lámparas fluorescentes. La perturbación que afectaría en mayor medida a los detectores de IR es la que se sitúa en la longitud de onda de 1014 nm, ya que estos detectores suelen tener un espectro entre los 800nm y los 1100nm.

Dentro del espectro de radiación infrarroja de estas lámparas hay que distinguir entre los distintos tipos de modulación que usan. Así se pueden encontrar varios tipos de lámparas fluorescentes con distinta modulación. Esta modulación oscila entre los 50 Hz de las lámparas que funcionan a frecuencia de la línea de red eléctrica y la frecuencia oscilante de las señales que emiten algunas de estas lámparas, y que pueden estar moduladas entre 50 KHz y 100 KHz.

De todas formas, la mayor parte de la radiación IR que emiten estas lámparas suele ser de baja frecuencia.

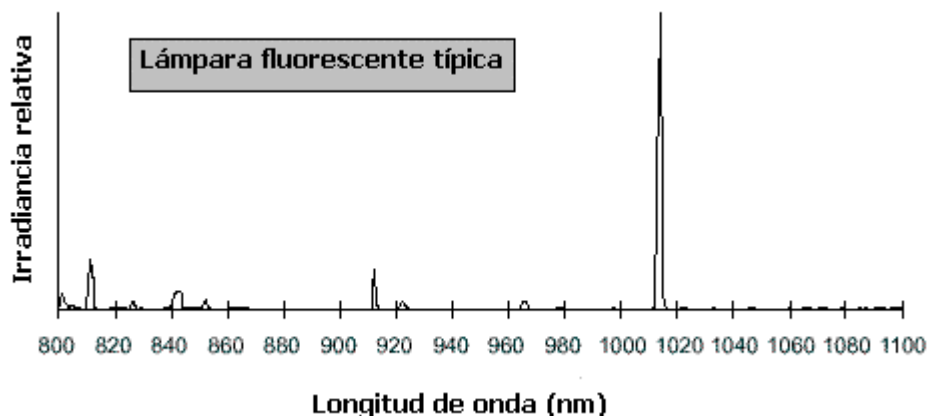


FIGURA 5.6. Emisión espectral de lámparas fluorescentes.

5.2.3.- Sensores utilizados para detección de presa

La elección de los sensores de infrarrojos para detectar a la presa se ha realizado teniendo en cuenta que:

- Los sensores deben ser de pequeño tamaño y peso para poder incorporarlos al microbot.
- El microbot depredador debería poder detectar a la presa a una cierta distancia (dos o tres metros).
- La interferencia producida en el entorno del microbot debe ser minimizada.
- Para tener una distancia de transmisión grande es necesario un diodo emisor de infrarrojos con alta intensidad radiante y un detector con baja irradiancia.

Los sensores de IR utilizados para la detección de la presa han sido: Módulo receptor de infrarrojos **IS1U60**, fotodiodo detector de infrarrojos **SFH203FA** y diodo emisor de infrarrojos **SFH485**.

5.2.3.1.- Módulo detector de infrarrojos IS1U60

El módulo receptor de infrarrojos IS1U60 modulado a 38 KHz de SHARP es un módulo utilizado habitualmente para control remoto de aparatos electrónicos. Este módulo presenta una serie de características que han sido determinantes para su elección como sensor de detección de presa. Estas características y su influencia sobre el comportamiento del microbot depredador se muestran en la Tabla 5.1.

Aunque posee una serie de características que le hacen ser un detector adecuado para conocer la dirección en la que se encuentra la presa, tiene un gran inconveniente y es que la salida que proporciona es digital y activa a nivel bajo ('0' = detectado, '1' =

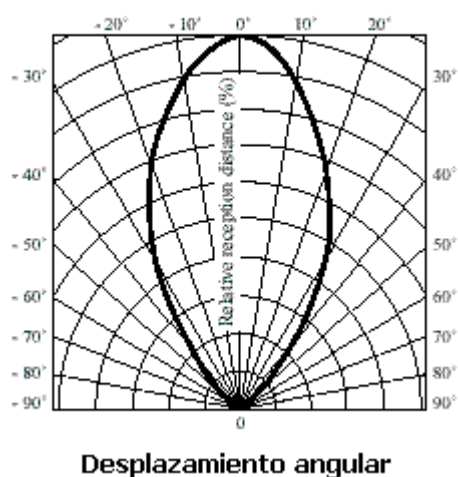
NO detectado). Al ser la salida digital no es posible saber a qué distancia se encuentra la presa, o si lo que se recibe es un ‘rebote’ (Al detector IS1U60 le llega una señal reflejada en un objeto, no la señal directa emitida por la presa).

La solución a este problema sería utilizar detectores analógicos pero no se han encontrado detectores analógicos modulados, por lo que se ha diseñado uno que no se ha llegado a implementar debido a la indisponibilidad de un fotodiodo receptor de infrarrojos adecuado. El estudio de este detector analógico modulado se puede ver en el ANEXO F.

CARACTERISTICAS DEL IS1U60	INFLUENCIA SOBRE MICROBOT
Posee un filtro paso banda centrado a una frecuencia de 38 KHz.	Al tener un filtro paso banda, se elimina la mayor parte de interferencia infrarroja que pudiera afectar al detector.
Baja irradiancia ($E_e < 10 \text{ lx}$).	A menor irradiancia mayor es la distancia que puede existir entre emisor y receptor de IR. La irradiancia del IS1U60 permite que la distancia entre el depredador y la presa sea de unos 3 metros, si no se usa señal moduladora.
Espectro de sensibilidad. $\theta = 60^\circ$ en dirección horizontal (Fig. 5.7a). $\theta = 30^\circ$ en dirección vertical (Fig. 5.7b). $\theta \equiv$ Angulo de sensibilidad	Determina el área que cubre un solo detector IS1U60. El espectro de sensibilidad se tendrá en cuenta cuando se sitúen los detectores IS1U60 sobre el microbot.
Alimentación entre 0 y 6 voltios.	Que la alimentación del detector pueda ser de hasta 6 voltios posibilita que junto con los sensores IS1U60 se alimenten otros componentes como el microcontrolador y la memoria RAM externa.

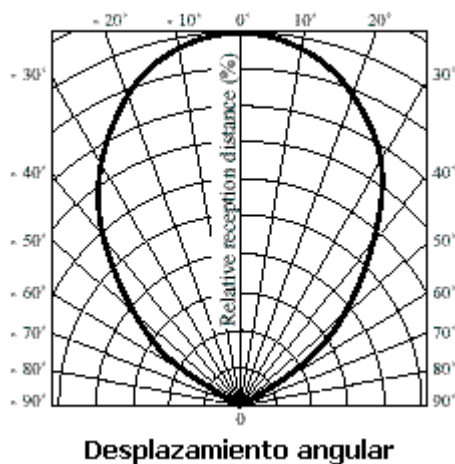
TABLA 5.1. Características del IS1U60 y su relación con el microbot depredador.

Angulo de sensibilidad (Dirección Y)



a)

Angulo de sensibilidad (Dirección X)



b)

FIGURA 5.7. Espectros de sensibilidad del IS1U60

Para obtener más información del IS1U60 consúltase el ANEXO B.

Aunque todas las características comentadas en la tabla anterior se han tenido en cuenta a la hora de elegir el IS1U60, la principal razón para su elección como detector de la presa ha sido su reducido precio en comparación con otros. Cada módulo detector de infrarrojos IS1U60 cuesta **550 ptas** (3'3 euros).

5.2.3.2.- Fotodiodo detector de infrarrojos SFH203FA

Este es un fotodiodo PIN (p-intrinsic-n) de silicio, de alta potencia y con un filtro óptico para luz visible (ver figura 5.5). Entre sus características destacan:

- Angulo de recepción de 40°.
- Sensibilidad espectral a una longitud de onda de 850 nm es de 0.59 A/W.
- Máxima sensibilidad a una longitud de onda de 900 nm.
- Consumo de potencia de 100 mW.

Este fotodiodo se utiliza para indicar al microbot depredador si la presa se encuentra a una distancia de unos 6 cm. o menos. Es el que indica que se ha capturado a la presa.

El filtro óptico evita que las perturbaciones que se produzcan fuera de la banda de 800 nm a 1100 nm afecten al fotodiodo. En este proyecto se ha situado el microbot depredador en un entorno con baja interferencia de infrarrojos, por lo que el SFH203FA no se ve afectado. En el caso de que se viera afectado se podría utilizar el detector analógico modulado o las cargas activas que se describe en algunos de los siguientes apartados.

5.2.3.3.- Diodo emisor de infrarrojos SFH 485

Este LED emisor de infrarrojos tiene que ir sobre la presa para que pueda ser detectada. Se ha escogido este emisor de infrarrojos por tres razones:

1. Tiene una intensidad radiante de 80 mW/sr ante una corriente directa de 100 mA.
2. Tiene un ángulo de radiación de 40° (Ver figura 5.8). No existen muchos diodos emisores de infrarrojos con un ángulo de radiación mayor que 40° que además, tengan una intensidad radiante alta.
3. Su precio es de 70 ptas. (0.42 euros) cada unidad.

Para obtener información más detallada del SFH203FA o SFH485, consúltase el ANEXO C.

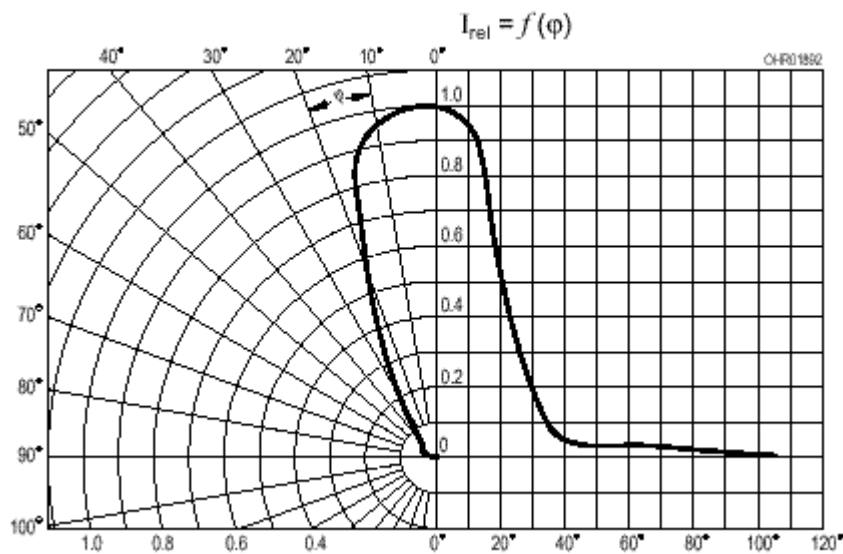


FIGURA 5.8. Espectro de radiación del SFH485.

5.2.4.- Circuitos para detección de presa

En este apartado se describen los circuitos de polarización y adaptación de la señal necesarios para el correcto funcionamiento de los sensores de detección de la presa.

Circuito para módulo detector de infrarrojos IS1U60

Este módulo recibe una señal de luz infrarroja modulada a 38 KHz y proporciona una salida digital, por tanto no necesita de un circuito externo para funcionar. Sin embargo, si la fuente de alimentación de este módulo produce interferencias se puede utilizar el circuito de la figura 5.9 para eliminar el ruido procedente de la fuente.

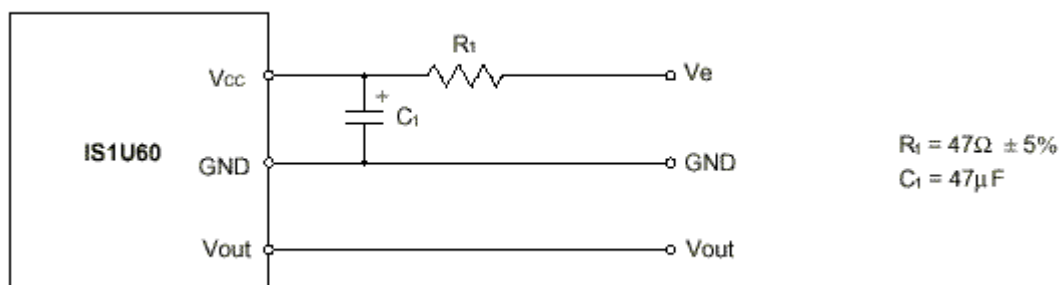


FIGURA 5.9. Circuito externo a IS1U60.

Circuito para fotodiodo detector de presa SFH203FA

El circuito asociado al SFH203FA para detectar a la presa es el mismo que el utilizado para detectar los obstáculos. Este circuito se estudiará en el apartado dedicado a la detección de los obstáculos puesto que, en un principio, el circuito sólo se utilizaba para detectar obstáculos. Fue al probar cómo recibía el circuito detector de infrarrojos las emisiones realizadas a frecuencia alta, cuando se vio que se podía usar el mismo circuito. Esto se debe a que este fotodiodo no responde bien ante la señal infrarroja, con una frecuencia portadora de 38 KHz, que emite el SFH485. El que no responda bien a esta frecuencia significa que, cuando recibe la señal de 38 KHz, el detector SFH203FA trata esta señal como una señal casi continua que baja su nivel conforme se acerca más al emisor.

Este comportamiento del detector es idóneo para nuestra aplicación, ya que permite detectar a la presa cuando está cerca, sin necesidad de utilizar más componentes (demodulador, detector de continua, etapas amplificadoras, etc.).

Circuito para emisor de infrarrojos de presa SFH485

El SFH485 debe transmitir una señal modulada a 38 KHz. Esta señal va a ser un '1' fijo ya que el IS1U60 no va a ser utilizado para recibir códigos de información, sino sólo para ver si recibe o no la señal emitida por el SFH485.

El circuito para la emisión de la señal modulada a 38 KHz se va a dividir en dos partes:

1. Circuito generador de reloj (Figura 5.10).
2. Circuito de polarización del emisor de infrarrojos (Figura 5.10).

1. Circuito generador de reloj

Es el encargado de proporcionar al SFH485 la señal portadora de 38 KHz. Para obtener esta señal se ha recurrido a un **Inversor Schmitt-Trigger** funcionando como reloj (Ver figura 5.10). Este inversor está disponible en cualquier tienda de electrónica en un encapsulado llamado **SN74LS14**, en el cual hay seis de estos inversores.

Se ha elegido este circuito porque: resulta bastante barato, se puede variar la frecuencia del reloj de manera sencilla (variando la resistencia) y en un mismo encapsulado hay seis inversores (se podrían utilizar varios circuitos de reloj si fuera necesario).

2. Circuito de polarización del emisor de infrarrojos

Consta de un transistor **2N2222A**, una resistencia, un condensador y el SFH485. A la base del transistor va conectada la salida del inversor schmitt-trigger, al emisor va conectada la resistencia y al colector va conectado el SFH485. El condensador se utiliza para las posibles interferencias de la fuente.

El transistor conmutará entre directa y corte según la señal de reloj de 38 KHz. Cuando esté en directa la corriente que pasa por la resistencia del emisor es similar a la que aparece en colector. De esta forma se tiene control sobre la corriente que se le inyecta al emisor de infrarrojos.

NOTA: La tensión de base nunca debe de ser superior a la de colector, es decir, el transistor no debe entrar en saturación.

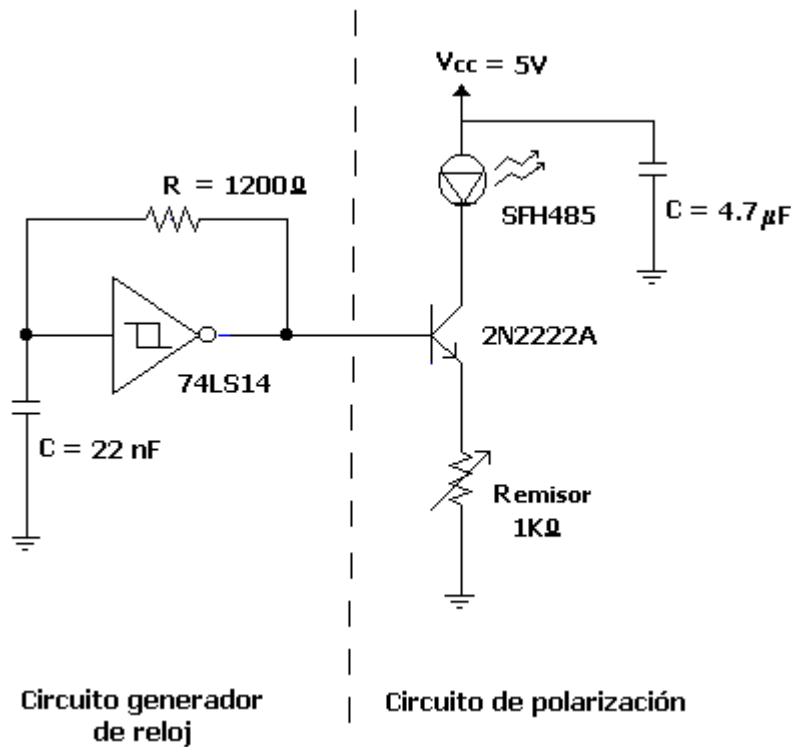


FIGURA 5.10. Circuito emisor de infrarrojos a 38 KHz.

Las hojas de características del 74LS14 y el transistor 2N2222A se encuentran incluidos en el ANEXO A.

5.3.- Detección de obstáculos

La detección de obstáculos permite al microbot depredador localizar y evitar los posibles obstáculos que se encuentre en su camino. Para lograr este fin, el microbot debe incorporar unos sensores que le suministren información sobre los obstáculos que haya en su entorno. Esta sección se va a centrar en el estudio de estos sensores y los circuitos asociados para su correcto funcionamiento.

5.3.1.- Elección de sensores

Puesto que la función de este microbot es la de depredador, sus movimientos se centrarán en ir hacia delante, girar a la derecha y girar a la izquierda. En raras ocasiones

tendrá que retroceder. Por lo tanto los sensores que se van a incorporar son: sensores para la detección frontal del obstáculo y sensores para la detección lateral del obstáculo.

Para la detección frontal se podrían haber utilizado sensores de ultrasonido o infrarrojos, pero supone menor gasto utilizar relés mecánicos (**bumpers**), ya que éstos son bastante más baratos y su gasto de energía es nulo.

Para la detección lateral no es recomendable utilizar bumpers puesto que, necesitan estar en contacto con el obstáculo para detectarlo. Tras los bumpers, la mejor elección son los sensores de infrarrojos, los cuales son mucho más baratos y de menor tamaño que los de ultrasonido.

5.3.2.- Sensores para detección frontal del obstáculo

Los bumpers son relés mecánicos a modo de pulsador que cuando entran en contacto con un objeto cierran el circuito y cortocircuitan la masa con la salida (nivel bajo '0'). La figura 5.11a muestra un bumper activado y otro desactivado.

Estos bumpers suelen llevar una resistencia de pull-up para que el sistema que lo utilice lea nivel alto cuando el bumper esté desactivado. Al conectar el bumper al microbot no ha sido necesario incluir esta resistencia de pull-up, ya que el puerto E de la tarjeta CT293+ incorpora una resistencia de pull-up de 4'7 K Ω por cada entrada de este puerto (Ver figura 5.11b).

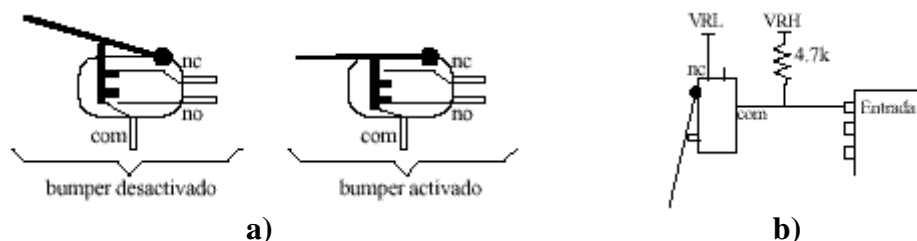


FIGURA 5.11. Estructura y conexión de un bumper.

5.3.3.- Sensores para detección lateral del obstáculo

En el apartado de detección de la presa se han comentado los sensores de infrarrojos SFH203FA y SFH485. Aprovechando que se disponían de varios de estos sensores, se decidió utilizarlos para la detección de los obstáculos, aunque se podrían haber utilizado otros, como el sensor de infrarrojos GPD02.

Si para la detección de la presa es necesario que el emisor de infrarrojos envíe una señal modulada a 38 KHz, para detectar los posibles obstáculos no se envía señal modulada. Las razones por las que no se utiliza modulación para detectar los obstáculos son:

- La modulación se utiliza para eliminar las posibles perturbaciones que se puedan producir, pero el uso de señales moduladas supone un mayor coste en

materiales y en consumo de potencia (son necesarios elementos activos como amplificadores, circuitos integrados, etc.).

- El filtro que incorpora el SFH203FA permite que las luces fluorescentes que existen en el entorno del microbot no produzcan interferencias. Este filtro no protege al detector SFH203FA de las radiaciones infrarrojas emitidas por el sol.
- La luz solar no incide directamente sobre el detector. De todas formas, se han cubierto los sensores con una carcasa para limitar la radiación infrarroja proveniente de la luz solar que el detector pueda recibir.

Para la detección del obstáculo se va a situar el emisor SFH485 junto al detector SFH203FA, de forma que el detector reciba las emisiones infrarrojas del emisor después de que hayan rebotado en el obstáculo (Ver Figura 5.12).

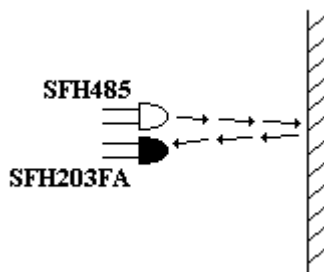


FIGURA 5.12. Situación de los sensores IR para detección de obstáculos.

Las características de estos dos sensores ya se han comentado en el apartado de detección de la presa. Sin embargo, hay que hacer referencia a los circuitos asociados a estos sensores, pues el emisor de infrarrojos para detección de obstáculos no envía señal modulada. Además, el circuito del fotodiodo SFH203FA que incorpora, tanto para detección de la presa como de los obstáculos, no se ha descrito en el apartado dedicado a la detección de la presa, por lo que se describe en éste.

Circuito para emisor de infrarrojos de obstáculos

Puesto que las emisiones del SFH485 no van moduladas, el circuito del emisor de IR se simplifica, siendo sólo necesaria una resistencia con la que se determina la corriente de polarización del emisor de infrarrojos. Para la detección de obstáculos, que realiza el microbot depredador, se ha utilizado una resistencia de $47\ \Omega$, lo que da una corriente de polarización de 75 mA aprox. (Ver figura 5.13).

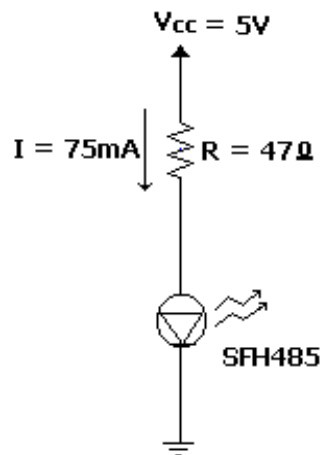


FIGURA 5.13. Circuito emisor de IR para detección de obstáculos.

Circuito para fotodiodo detector de obstáculos SFH203FA

Cuando los fotodiodos detectores de IR reciben una emisión de luz infrarroja generan una corriente inversa que será mayor, cuanto mayor sea la intensidad luminosa recibida. Los fotodiodos detectores de infrarrojos necesitan una carga sobre la que recaiga esta corriente inversa. Dependiendo de si la carga es activa o pasiva se tendrán unas características de impedancia.

Carga resistiva. Es la más simple de las cargas pasivas. Cualquier corriente inversa producida por el fotodiodo dará lugar a un voltaje a través de la resistencia, el cual puede ser amplificado para dar la salida requerida. El principal inconveniente de esta carga es que cualquier fuente luminosa, que esté en la banda de los 800 a los 1100 nm, afectará al detector, pudiendo llegar a saturarlo. La figura 5.14a representa el circuito receptor de IR utilizado por el microbot depredador para detectar obstáculos. Este circuito está diseñado para que su salida sea menor, cuanto mayor sea la radiación infrarroja recibida.

Carga activa. Esta carga permite tener alta impedancia a altas frecuencias y baja impedancia a bajas frecuencias, con lo que se consigue reducir la ganancia de las posibles interferencias a bajas frecuencias y DC. En la figura 5.14b se muestra un ejemplo de una carga activa.

Al ser bajas las interferencias infrarrojas que existen en el entorno del microbot, se ha utilizado una simple carga resistiva. Si el nivel de las interferencias producidas en el espectro infrarrojo fuera alto se podría hacer uso de una carga activa, siempre y cuando las emisiones de infrarrojos, que debe detectar el fotodiodo, se realicen a alta frecuencia.

El valor de la resistencia de carga (R_L) sobre la que se hace recaer la corriente inversa generada en el SFH203FA se obtiene de la siguiente forma:

1. Sabiendo la intensidad radiante (I_e) que irradia el emisor de IR (en este caso el SFH485) y la distancia (d) que se quiere tener entre emisor y receptor, se obtiene la irradiancia (E_e) mediante la ecuación $E_e = I_e/d^2$.
2. Después se obtiene la corriente inversa (I_p , también llamada fotocorriente) asociada a esa irradiancia, mirando la tabla que las relaciona (Ver hojas de características del SFH203FA en el ANEXO C).
3. Por último, sabiendo la alimentación (V_{cc}) utilizada es muy sencillo averiguar el valor de la resistencia de carga, aplicando la ecuación resultante del estudio de la malla ($V_{cc} - I_p \cdot R_L$).

En este caso el detector se utiliza para detección de obstáculos, por lo que hay que tener en cuenta el rebote en el obstáculo de la radiación infrarroja emitida por el SFH485. Puesto que se quiere detectar el obstáculo que haya a unos 7 cm del microbot, la distancia entre emisor y receptor deberá ser de unos 25 cm, ya que hay que tener en cuenta que los obstáculos no reflejan toda la radiación que les llega.

Si $d = 25\text{cm}$ y $V_{cc} = 5\text{V}$, la resistencia de carga R_L se puede aproximar a un valor de $680\text{ K}\Omega$.

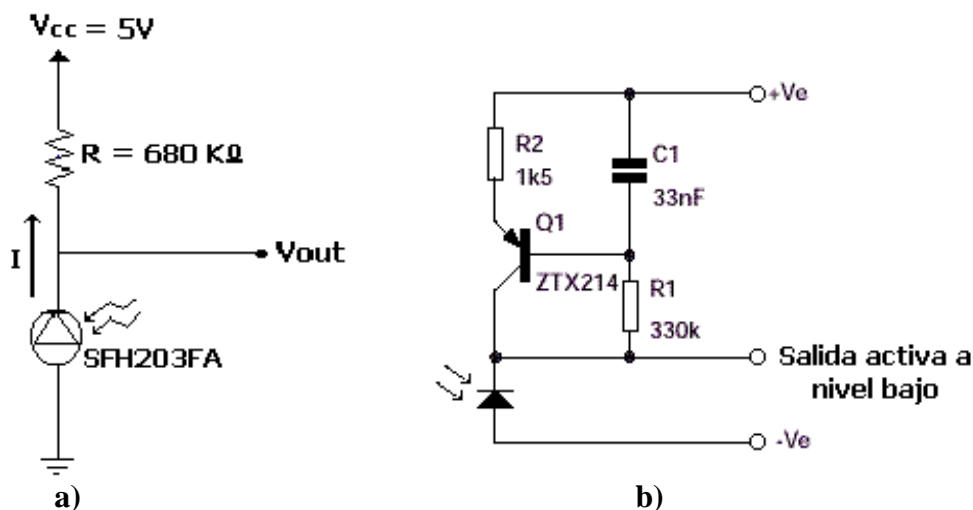


FIGURA 5.14. Carga pasiva y carga activa.

En teoría se podría realizar la conexión de la salida del detector de obstáculos (V_{out} de la figura 5.14a) al microbot a través del puerto E de la CT293+, pero existe un **problema**: “Al disponer el puerto E de resistencias de pull-up de $4'7\text{ K}\Omega$, la mayoría de la corriente inversa generada por el SFH203FA pasa por esta resistencia y no por la de $680\text{ K}\Omega$. Al ser la corriente inversa del orden de los microamperios, apenas se nota la caída de tensión producida por la recepción de infrarrojos”.

Solución: Para solucionar este problema se ha recurrido a un amplificador LM324 actuando como buffer de tensión (Ver figura 5.15). De esta forma se consigue que la resistencia de pull-up no afecte a la salida del circuito detector. Además, se puede

cambiar este buffer por una etapa amplificadora, si se considera necesario cambiar la ganancia del circuito detector.

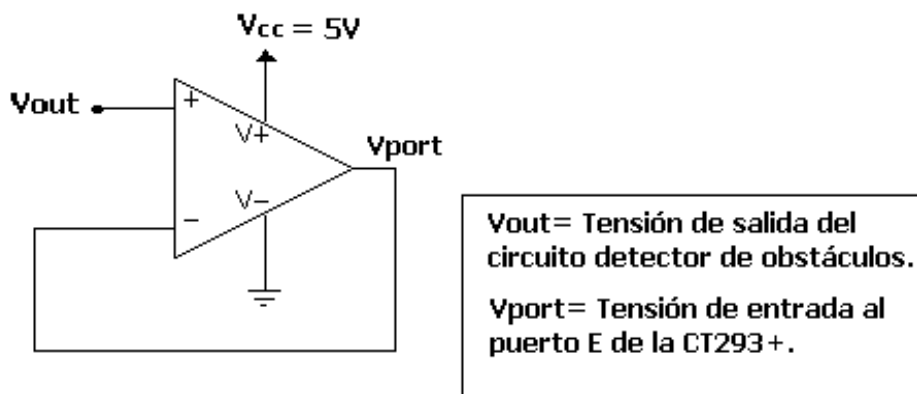


FIGURA 5.15. Buffer de tensión para adaptar salida de detector a puerto E.

5.4.- Encoder

Para tener una mayor precisión en la medida de los giros que hace, y de la distancia que recorre el microbot, se ha incorporado un sensor de infrarrojos de corta distancia CNY70, que actúa a modo de encoder.

Para que el CNY70 actúe a modo de encoder, una de las ruedas del microbot debe llevar adosado un círculo de papel con franjas blancas y negras, como el que se muestra en la figura 5.16. De esta forma, cuando la rueda está girando, el encoder detecta los cambios de blanco a negro y viceversa.

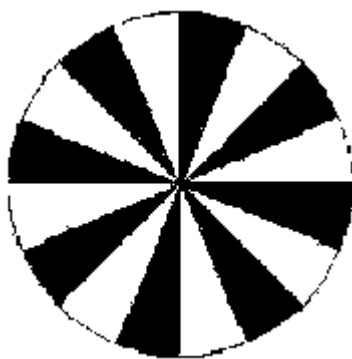


FIGURA 5.16. Círculo para que CNY70 pueda funcionar como encoder.

5.5.- Disposición y conexión de los sensores

Una vez conocidos los circuitos de detección de la presa y de obstáculos, sólo queda comentar la disposición de los sensores y la conexión de los circuitos al microbot

depredador. Pero antes hay que saber cuántos sensores se van a utilizar para la detección de presa y obstáculos:

- Para detección de la presa se han utilizado 3 módulos digitales IS1U60 y 1 fotodiodo detector de infrarrojos SFH203FA.
- Para la detección de obstáculos se han utilizado 2 bumpers y 2 parejas, formadas cada una por 1 fotodiodo detector SFH203FA y un LED emisor SFH485.

5.4.1.- Disposición de los sensores

La disposición de los sensores será distinta, dependiendo de si son los sensores de detección de la presa o son lo de detección de obstáculos.

- **Sensores de detección de presa.** Los sensores para la detección de la presa van situados sobre el microbot, encima de las demás tarjetas (Ver figura 5.17). Los IS1U60 se sitúan de tal forma que sus espectros de sensibilidad estén acoplados, como se indica en la figura 5.16. De esta forma no se pierde a la presa y el IS1U60 frontal es más directivo. El SFH203FA se sitúa delante del IS1U60 frontal para detectar si la presa está cerca.
- **Sensores de detección de obstáculos.** Los bumpers van situados en la parte delantera del microbot y cada pareja formada por un SFH203FA y un SFH485, se dispone a un lado del microbot.
- **Encoder.** El encoder va situado sobre uno de los motores, enfrenteado a la rueda que lleva adosado el círculo con franjas blancas y negras.

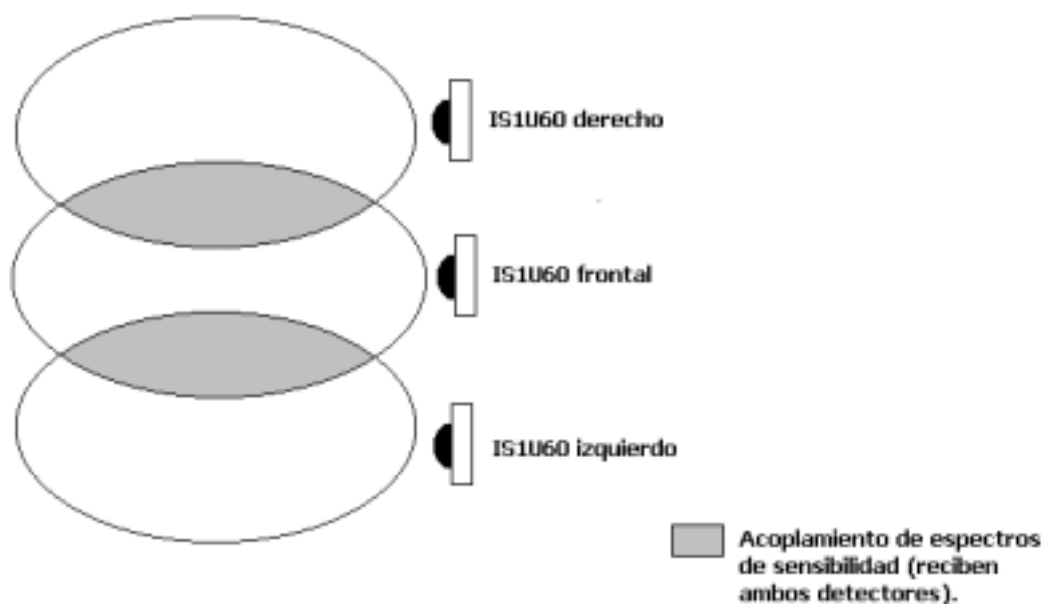


FIGURA 5.16. Acoplamiento de los espectros de sensibilidad de los IS1U60

5.4.2.- Conexión de los circuitos de detección

La conexión de los circuitos para detección de la presa se ha realizado a través del puerto D de la CT6811, mientras que los circuitos para detección de obstáculos se han conectado a través del puerto E.

Por otra parte, el encoder, al ser un CNY70, se ha conectado a la entrada del Sensor 4 de la tarjeta CT293+.

La tabla 5.2 muestra la conexión de estos sensores.

	Sensor	Bit del puerto
Sensores de detección de presa	IS1U60 Frontal	Bit 2 del puerto D
	IS1U60 Derecho	Bit 3 del puerto D
	IS1U60 Izquierdo	Bit 4 del puerto D
	SFH203FA para detectar a la Presa	Bit 5 del puerto D
Sensores de detección de obstáculos	Bumper Derecho	Bit 0 del puerto E
	Bumper Izquierdo	Bit 1 del puerto E
	SFH203FA para detectar obstáculo derecho	Bit 2 del puerto E
	SFH203FA para detectar obstáculo izquierdo	Bit 3 del puerto E
Encoder	Sensor de infrarrojos de corta distancia CNY70	Sensor 4 de la CT293+

TABLA 5.2. Conexión de sensores al microbot.

5.5.- Alimentación del microbot depredador

En el microbot depredador existen muchos elementos que consumen potencia, por lo que van a ser necesarias varias fuentes de alimentación. En este caso se necesitarán tres fuentes de alimentación, debido a que:

- Los motores tienen picos de corriente muy altos e introducen ruido en la fuente, por lo que no se pueden alimentar con los sensores ni con la memoria externa.

- Por norma, los circuitos digitales no se alimentan con la misma fuente que los analógicos. Entonces, los sensores digitales (IS1U60) y los analógicos (SFH485 y SFH203FA) no deben utilizar la misma fuente.

Teniendo en cuenta lo anterior se necesita: una fuente de alimentación independiente para los motores, otra para alimentar los sensores analógicos y otra más para alimentar los sensores digitales y el microcontrolador. En el caso de utilizar la memoria externa, al ser recomendable que se alimente ésta con el microcontrolador, y para no hacer uso de una cuarta fuente, se utilizará una única fuente que alimente el conjunto formado por microcontrolador, memoria RAM externa y sensores digitales.

El principal problema que surge de utilizar una única fuente de alimentación para microcontrolador, RAM externa y sensores digitales, es el elevado consumo de potencia del conjunto. Por esta razón cuando se utilice la memoria externa se recurrirá a un transformador.

CAPITULO 6. PROGRAMACION DEL MICROBOT DEPREDADOR

6. PROGRAMACION DEL MICROBOT DEPRADOR

Este capítulo está dedicado al estudio de la programación realizada en el microbot deprador. Se puede diferenciar entre la programación de la aplicación básica en lenguaje ensamblador y la programación de una aplicación más compleja en lenguaje C.

El programa desarrollado en lenguaje ensamblador se ha grabado en la memoria EEPROM del 68HC11, lo que permite que el microbot tenga una mayor autonomía, puesto que no depende de un ordenador que le transfiera el fichero, cada vez que quiera ejecutarse el programa.

Por otra parte, el programa desarrollado en C se carga en memoria RAM externa, la cual, al disponer de 32 K de memoria, permite programar aplicaciones más complejas.

Aunque la programación en lenguaje C es más fácil y menos tediosa que la programación en ensamblador, el lenguaje ensamblador genera un código más eficiente. Por esta razón se utiliza el lenguaje ensamblador para programar la EEPROM.

6.1.- Programación en lenguaje ensamblador

Esta sección está dedicada a la aplicación que se ha programado en lenguaje ensamblador. Primero se va a describir la actuación del microbot ante la ejecución del programa en ensamblador. Luego se incluye el flujograma del programa para que sirva de orientación al lector a la hora de ver el código asociado a este programa. La leyenda de los elementos utilizados en el flujograma se puede ver en la última hoja del presente capítulo.

El código asociado a este programa se encuentra en el ANEXO D, bajo el nombre de **VIGILA.ASM**.

6.1.1- Descripción de la aplicación básica

La aplicación programada en lenguaje ensamblador consiste en:

- El microbot realiza un recorrido en el cual, gira sobre sí mismo 375° aproximadamente. Después avanza unos 40 cm y se queda parado durante 2 segundos.
- Si durante ese recorrido se detecta a la presa (mediante los IS1U60), el microbot gira hacia un lado u otro, dependiendo del módulo detector que esté recibiendo, hasta que se sitúa de frente a la presa (solo recibe en IS1U60 frontal). Si pierde a la presa comienza otra vez con el recorrido.

- Si cuando está realizando el recorrido, o está siguiendo a la presa, detecta un obstáculo frontal, el microbot lo esquivo utilizando los bumpers y los sensores de infrarrojos laterales.
- Si, por el contrario, se detecta un obstáculo con los sensores de infrarrojos laterales, mientras el microbot está girando, éste se aleja del obstáculo y comienza de nuevo el recorrido.

6.1.2.- Diagrama de Flujo de la aplicación básica

A continuación se muestra el diagrama de flujo asociado a la aplicación básica programada en lenguaje ensamblador.

Para una mejor comprensión del diagrama de flujo, se explican algunas de las expresiones utilizadas en este diagrama de flujo:

Inicio. Indica el comienzo del programa.

Fin. Indica el final del programa.

Pasos. Es una variable que lleva la cuenta de los pasos del encoder. Hay que tener en cuenta que la cuenta de los pasos del encoder se hace por interrupciones.

Tiempo. Es la variable que lleva la cuenta del tiempo. Se cuenta en segundos.

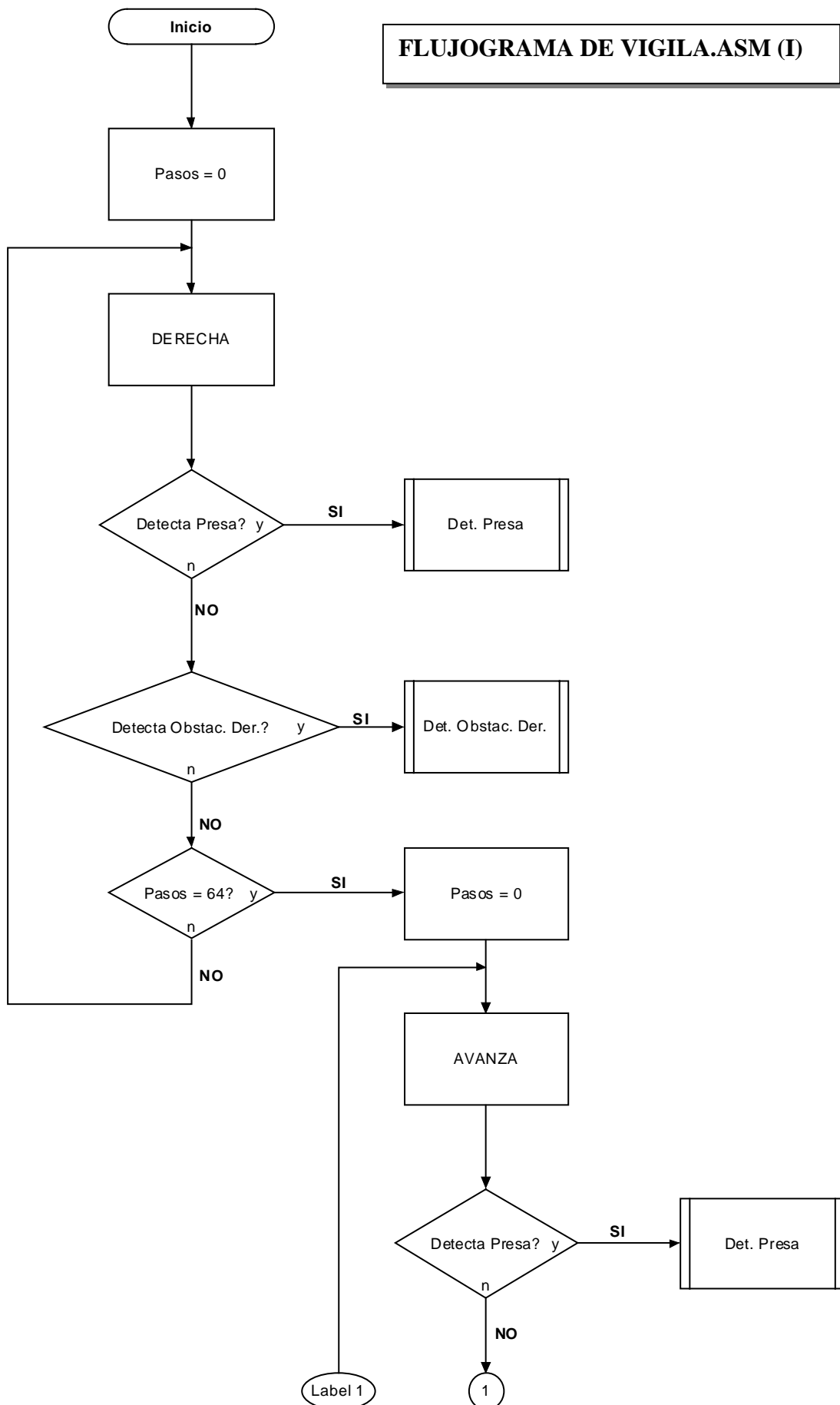
Det. Presa. Es la rutina que se asocia a la detección de la presa por parte de los módulos digitales IS1U60 y el sensor de captura SFH203FA.

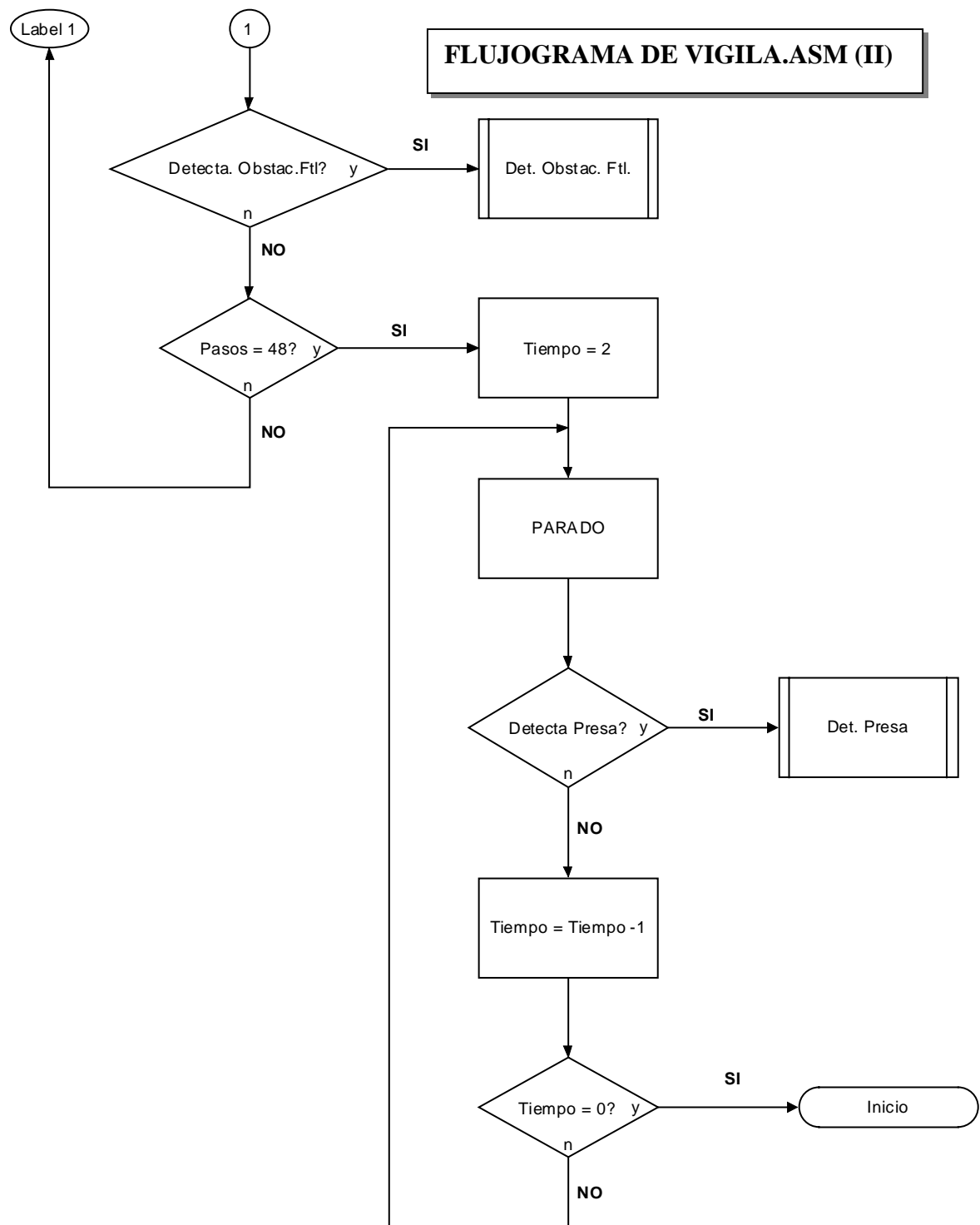
Det. Obstac. Der. Es la rutina que se asocia a la detección producida en el detector infrarrojo de obstáculos lateral derecho. Esta rutina se ejecuta cuando el detector lateral derecho de obstáculos se activa.

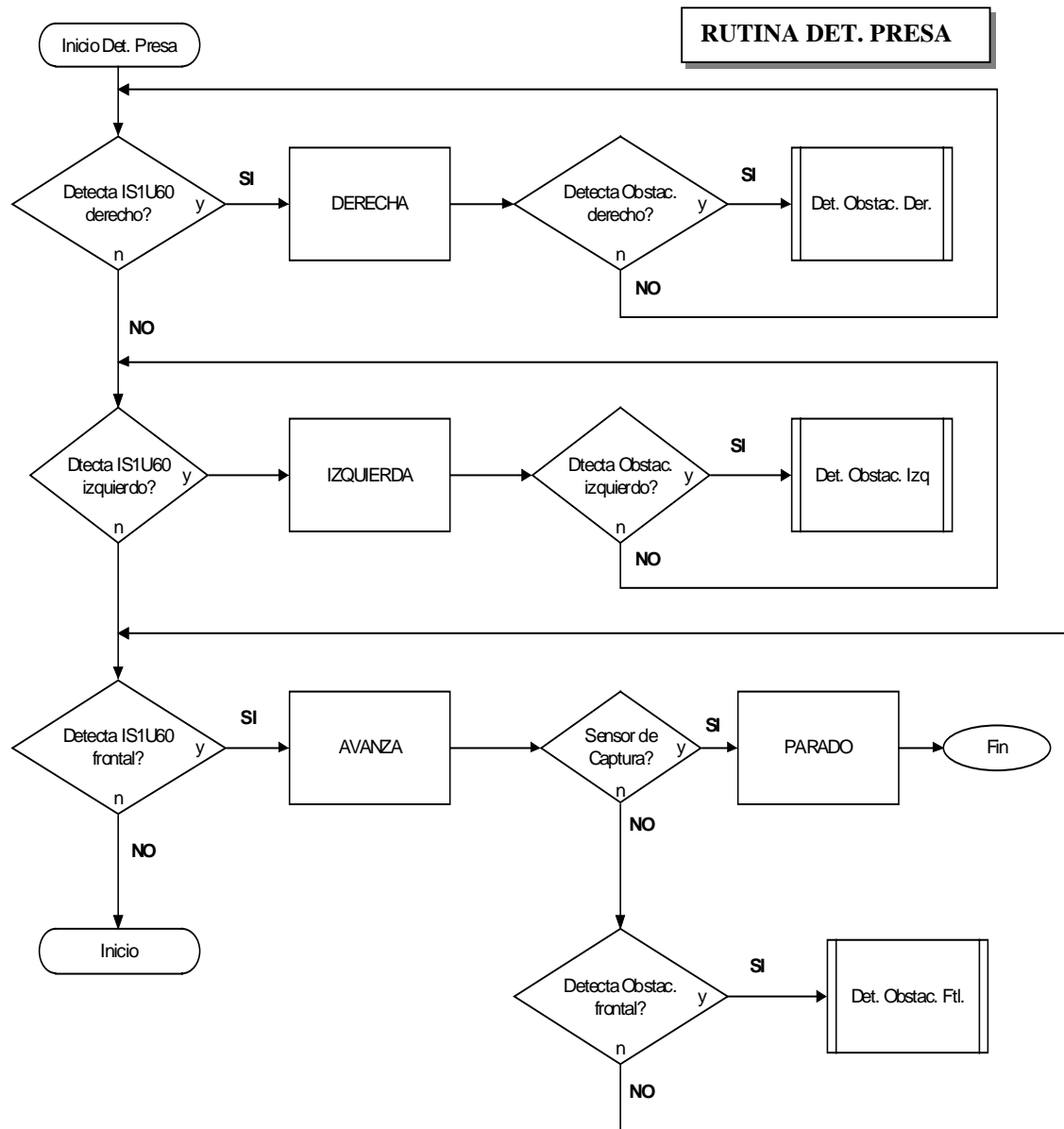
Det. Obstac. Izq. Es la rutina que se asocia a la detección producida en el detector infrarrojo de obstáculos lateral izquierdo. Esta rutina se ejecuta cuando el detector lateral izquierdo de obstáculos se activa.

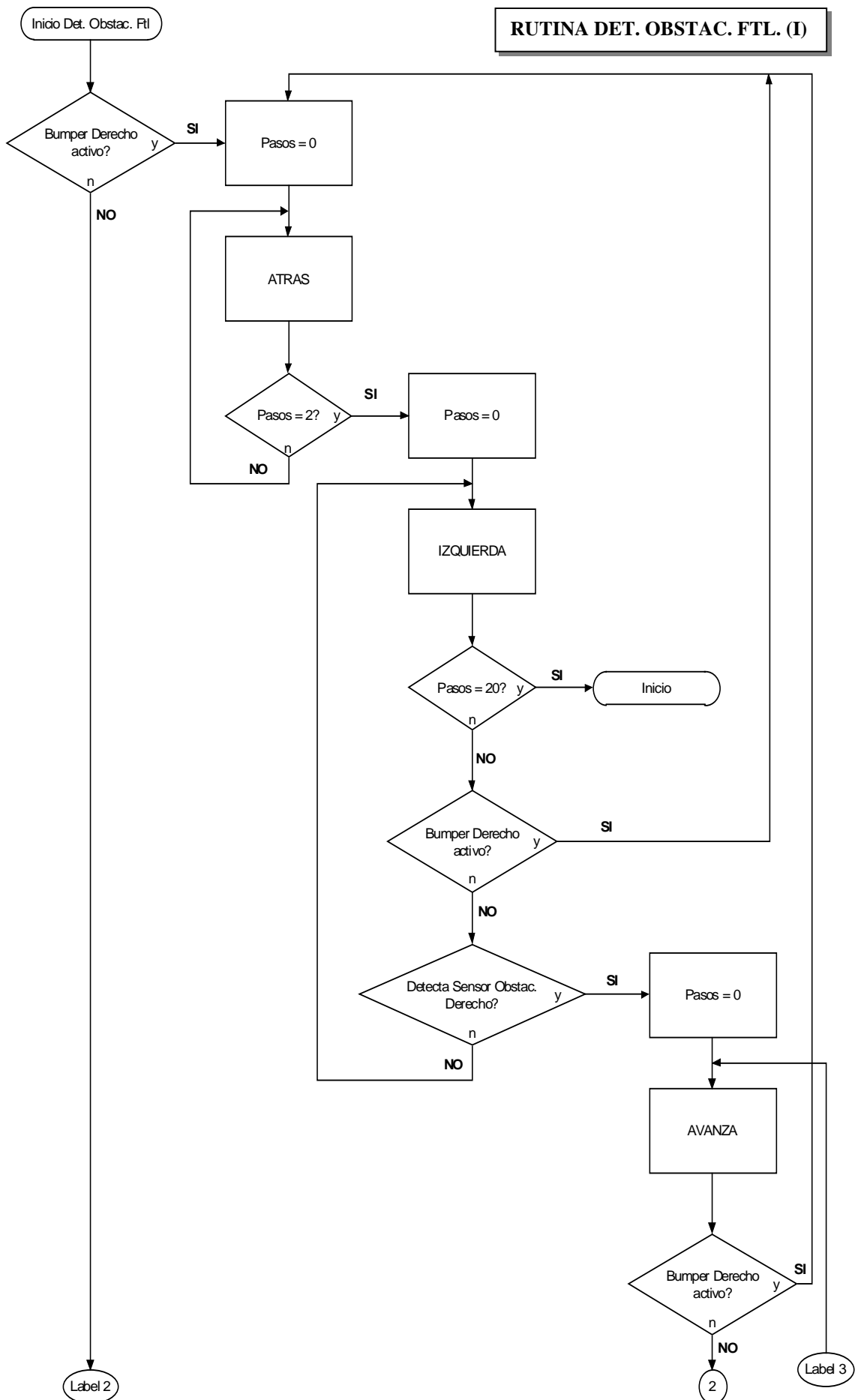
Det. Obstac. Ftl. Es la rutina que se ejecuta cuando se produce la detección de alguno de los bumpers frontales.

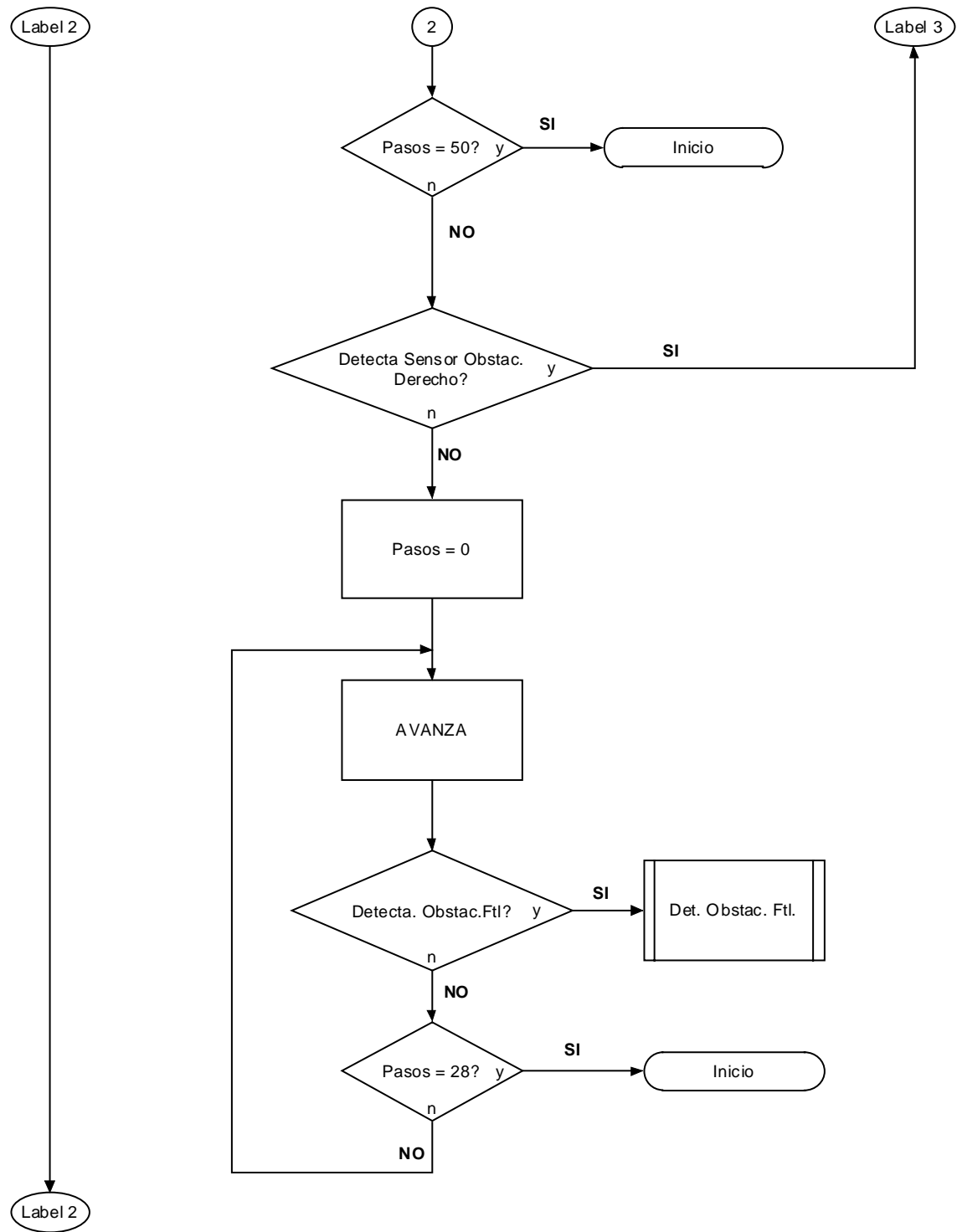
Sensor Captura. Es el SFH203FA que se utiliza para indicar la captura de la presa.

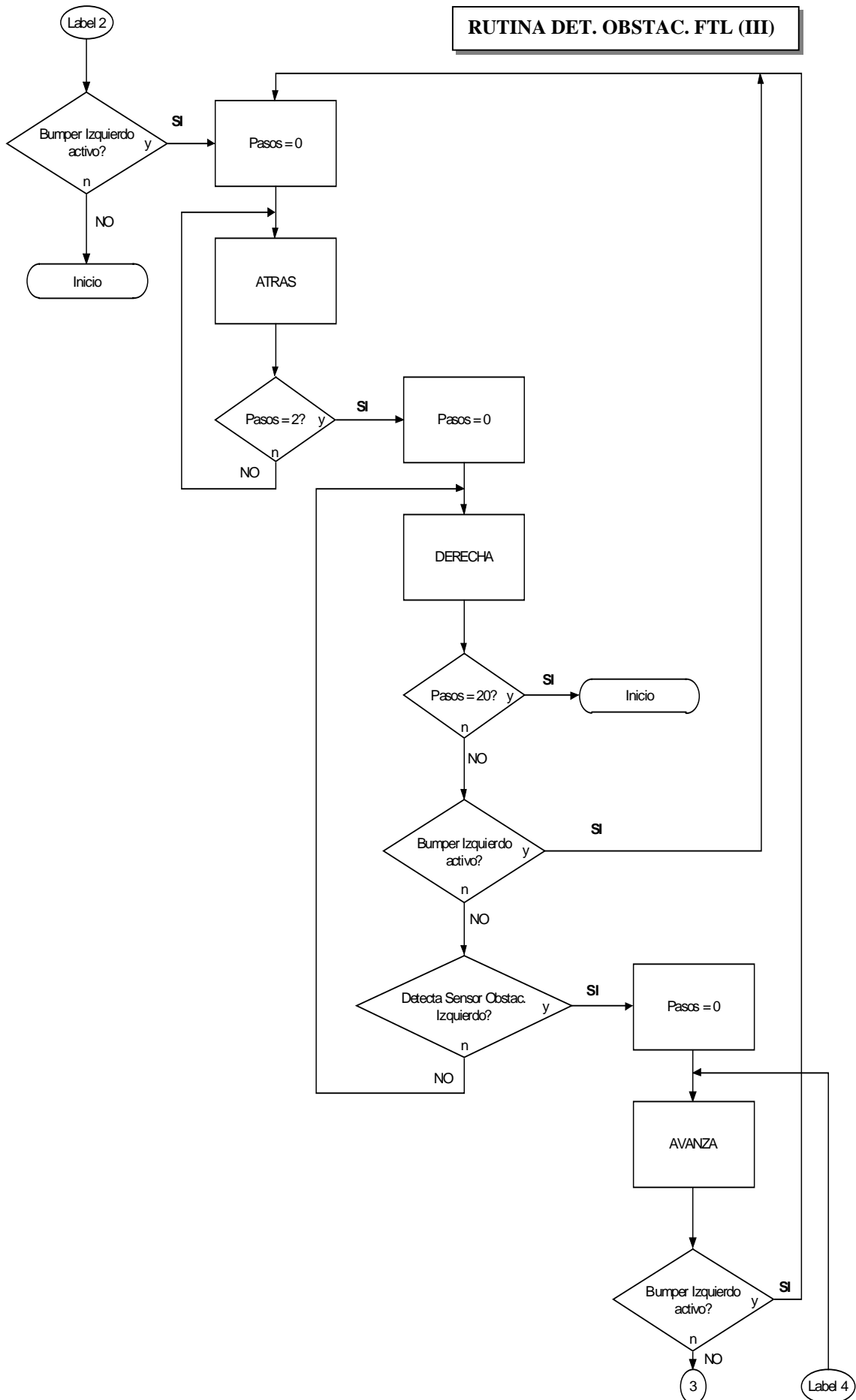
FLUJOGRAMA DE VIGILA.ASM (I)

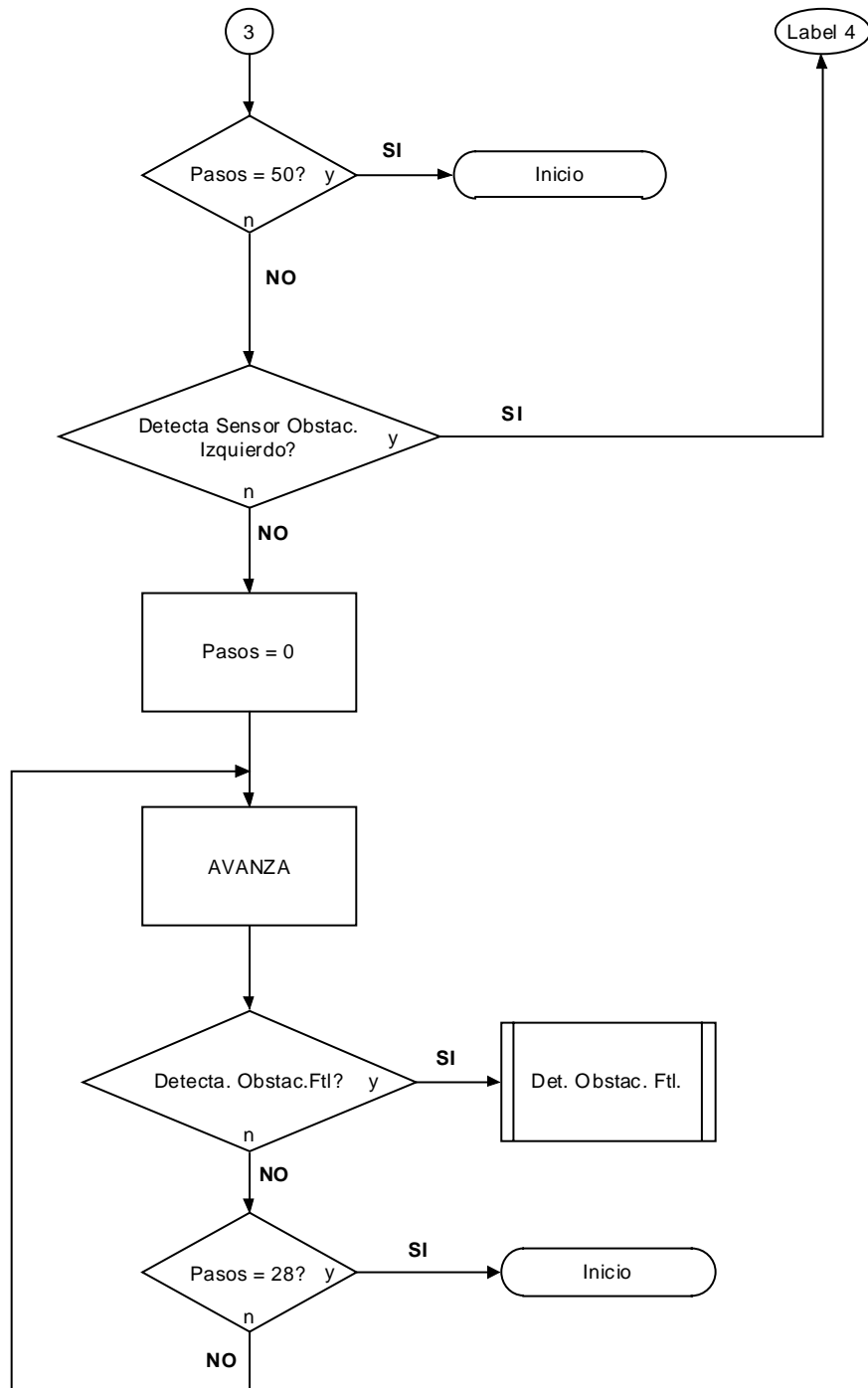


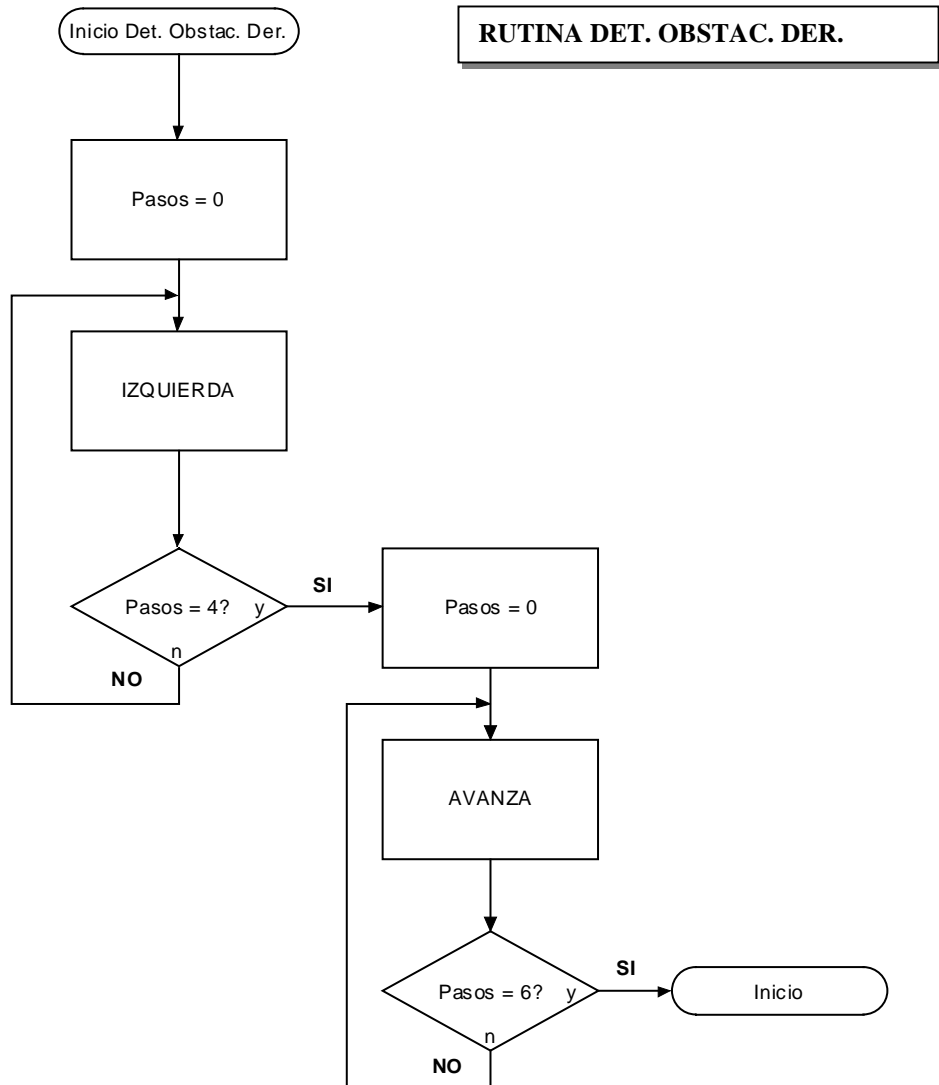


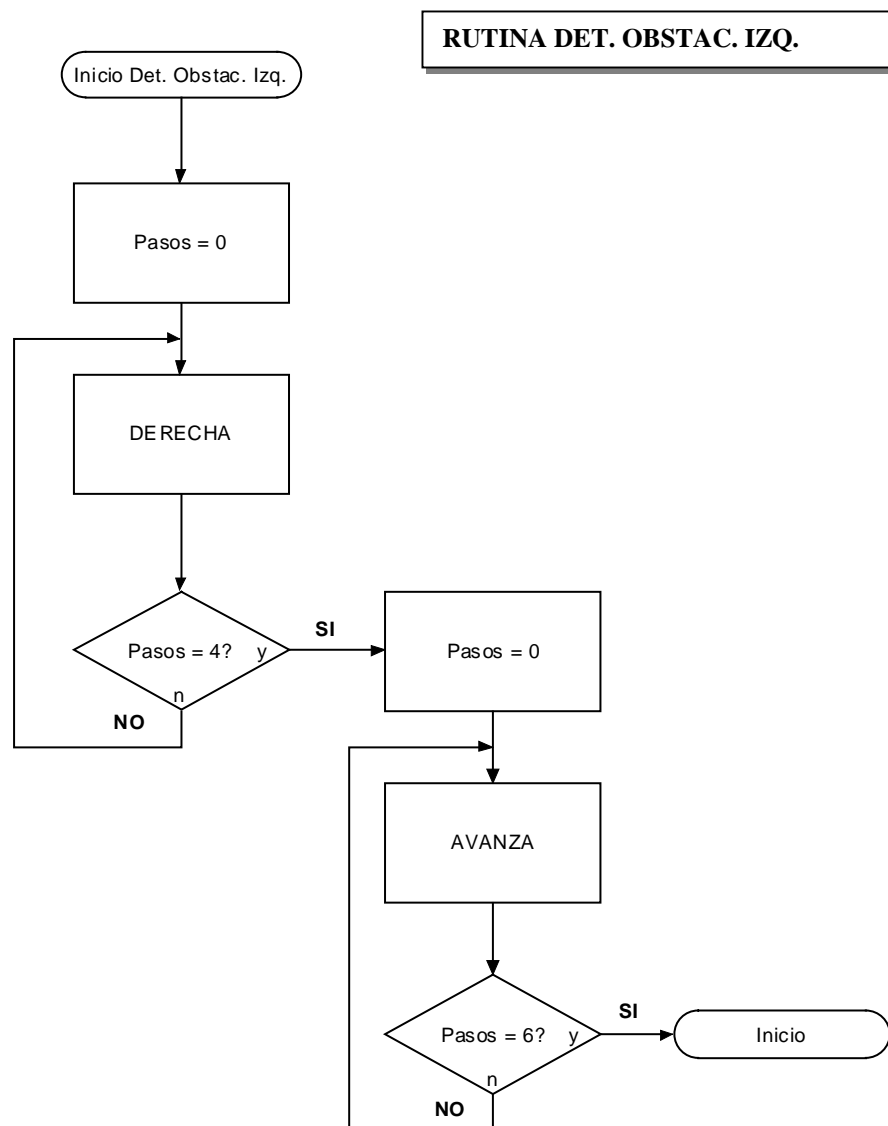


RUTINA DET. OBSTAC. FTL. (II)



RUTINA DET. OBSTAC. FTL. (IV)





6.2.- Programación en lenguaje C

Al igual que en la anterior sección, en ésta se va a comentar la aplicación programada en lenguaje C, para luego insertar el flujograma que sirva de referencia al lector.

El código asociado al programa en C se encuentra en el ANEXO E, bajo el nombre de **MENU.C**.

6.2.1.- Descripción de la aplicación en C.

La aplicación programada en lenguaje C consta de un menú de cuatro opciones. Para no tener que estar conectado al PC, mientras se está ejecutando o cargando el programa en la RAM externa, se han utilizado dos switches que permiten cambiar de opción de manera sencilla. Así, una vez que se ha cargado el programa en la RAM externa, se puede desconectar el microbot del PC.

Los switches, que a partir de ahora se nombrarán como SW1 y SW2, van conectados a los bits 4 y 5 del puerto E.

Las opciones del menú son:

1. Modo Test.
2. Búsqueda de Presa.
3. Recorrido en Espiral.
4. Recorrido en Cruz.

1. Modo Test (SW1 = 0; SW2 = 0)

Esta opción permite al usuario comprobar si los sensores funcionan correctamente. De forma que:

- Si se activa alguno de los bumpers, el microbot retrocede.
- Si se activa alguno de los detectores infrarrojos laterales, el microbot gira hacia el lado contrario de la detección.
- Si se activan los sensores digitales de detección de la presa (IS1U60), el microbot gira o avanza, dependiendo del sensor que esté activo (izquierdo, derecho o frontal).
- Si se activa el detector de captura de la presa (SFH203FA), el microbot se para y se sale de la opción modo test.

2. Búsqueda de Presa (SW1 = 1; SW2 = 0)

La principal misión de esta opción es demostrar cómo el microbot depredador es capaz de encontrar a la presa, incluso si ésta se esconde tras un obstáculo.

El comportamiento del microbot, al elegir esta opción en el menú, es el siguiente:

- El microbot realiza un recorrido muy sencillo, que consiste dar una vuelta sobre sí mismo y avanzar unos 20 cm.
- Si se activan los sensores digitales de detección de la presa (IS1U60), el microbot gira o avanza, dependiendo del sensor que esté activo (izquierdo, derecho o frontal). Si se pierde la señal de la presa, el microbot, dependiendo de cuál haya sido el último detector digital activo, realiza una de las siguientes acciones:
 - Si el último detector activo ha sido el IS1U60 frontal, quiere decir que la presa se ha escondido tras un obstáculo, por lo que el microbot avanzará hasta encontrar el obstáculo. Al encontrar el obstáculo, lo irá bordeando hasta que detecte otra vez a la presa. Si en un tiempo dado no se encuentra el obstáculo o la presa, el microbot continúa con su recorrido normal.
 - Si, por el contrario, el último detector activo ha sido el IS1U60 izquierdo o el IS1U60 derecho, la presa puede haberse escondido tras un obstáculo, o puede haberse ido por el lado del último detector activo. Por lo tanto, el microbot gira hacia el lado del último detector activo (IS1U60 derecho o IS1U60 izquierdo), buscando a la presa. Si no la encuentra, vuelve a la posición aproximada donde la perdió y realiza la misma rutina que si es el IS1U60 frontal el último en detectar a la presa.
- Si cuando está realizando el recorrido, o está siguiendo a la presa, detecta un obstáculo frontal, el microbot lo esquivo utilizando los bumpers y los sensores de infrarrojos laterales.
- Si, por el contrario, mientras el microbot está girando, se detecta un obstáculo con los sensores de infrarrojos laterales, éste se aleja del obstáculo.

La búsqueda de la presa solo se ha introducido en esta opción para que se vea de manera más clara como lleva a cabo esta búsqueda. Sin embargo, esta rutina se podría introducir en cualquiera de las opciones de recorrido en espiral y recorrido en cruz.

3. Recorrido en espiral

Tanto esta opción como la de recorrido en cruz, se han utilizado para mostrar dos de los posibles recorridos a modo de “vigilancia”, que puede realizar el microbot depredador.

Al elegir esta opción, el microbot actúa de la siguiente forma:

- El microbot comienza realizando un recorrido en espiral, como el que se muestra en la figura 6.1.
- Si se activan los sensores digitales de detección de la presa (IS1U60), el microbot: gira a la derecha, si está activo el detector IS1U60 derecho; gira a la izquierda, si está activo el detector IS1U60 izquierdo; avanza, si está activo sólo el IS1U60 frontal.

- Si cuando está realizando el recorrido, o está siguiendo a la presa, detecta un obstáculo frontal, el microbot lo esquiva utilizando los bumpers y los sensores de infrarrojos laterales.
- Si, por el contrario, mientras el microbot está girando, se detecta un obstáculo con los sensores de infrarrojos laterales, éste se aleja del obstáculo.

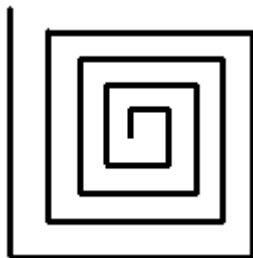


FIGURA 6.1. Recorrido en espiral.

4. Recorrido en Cruz (SW1 = '1'; SW2 = '1')

En esta opción el microbot ejecuta un recorrido en forma de cruz (Ver figura 6.2), cada vez más grande. En este recorrido, cuando el microbot llega a los bordes de la cruz, gira a izquierda y derecha, para luego volver al centro y pararse durante 2 segundos.

La forma de realizar el recorrido es la única diferencia que existe entre la opción de recorrido en espiral y ésta, ya que la forma de actuar al detectar a la presa o los obstáculos es la misma.

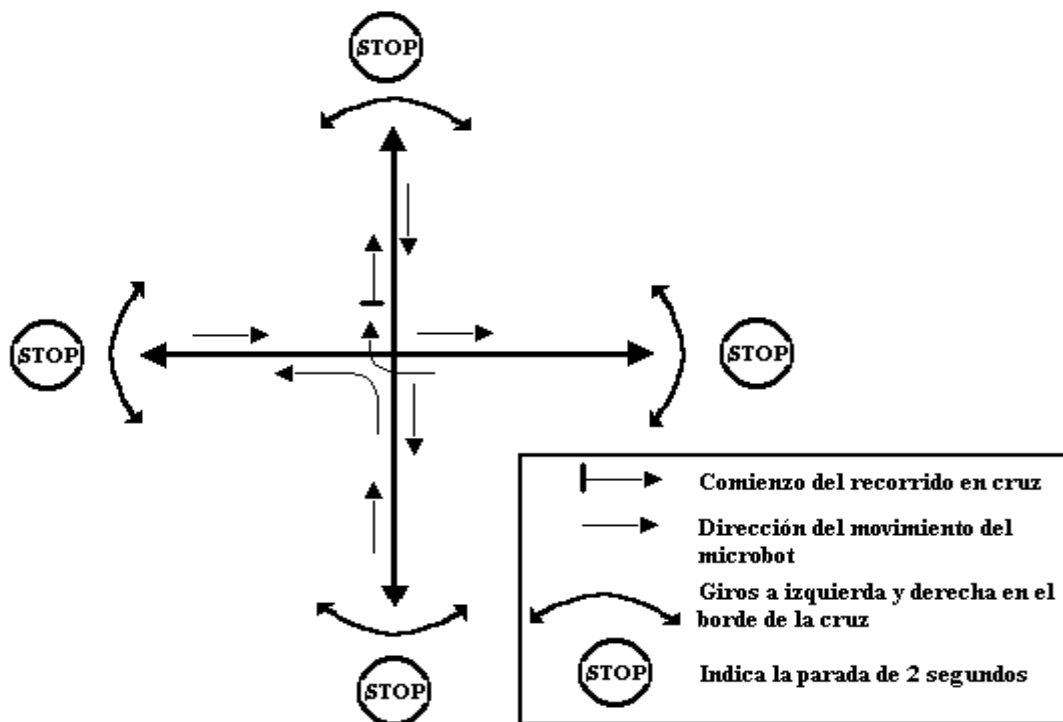


FIGURA 6.2. Recorrido en forma de cruz.

6.2.2.- Diagrama de Flujo de la aplicación en lenguaje C

En esta sección se va a introducir el diagrama de flujo de la aplicación programada en C.

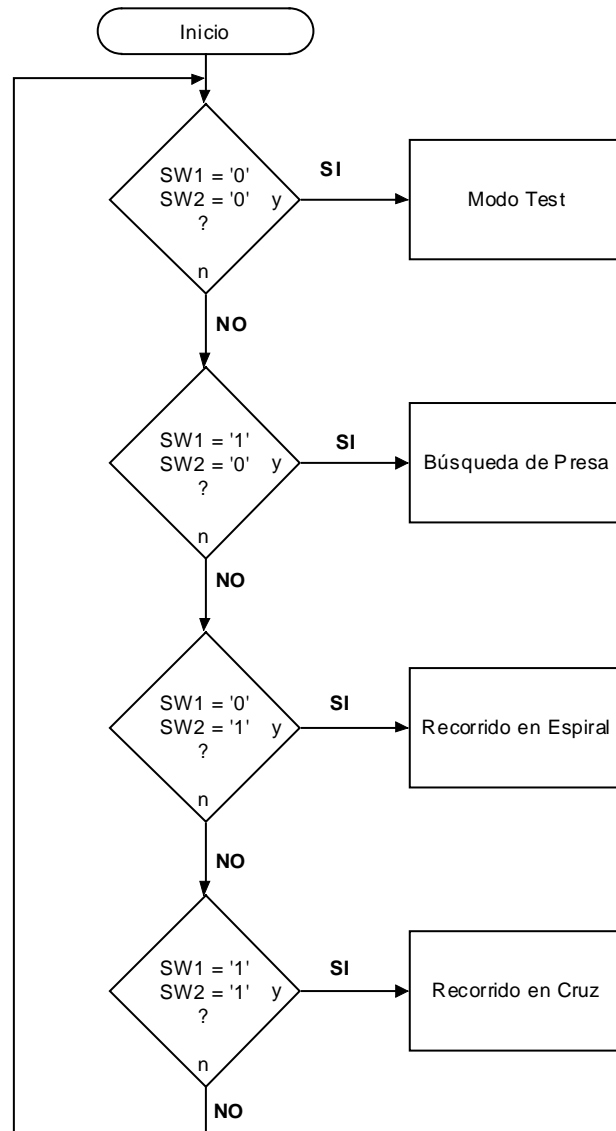
Algunas de las rutinas que se realizan en este programa son iguales a las descritas en el programa en ensamblador VIGILA.ASM. Por lo tanto, los flujogramas son idénticos y sería absurdo volver a exponer el mismo flujograma. Estas rutinas son:

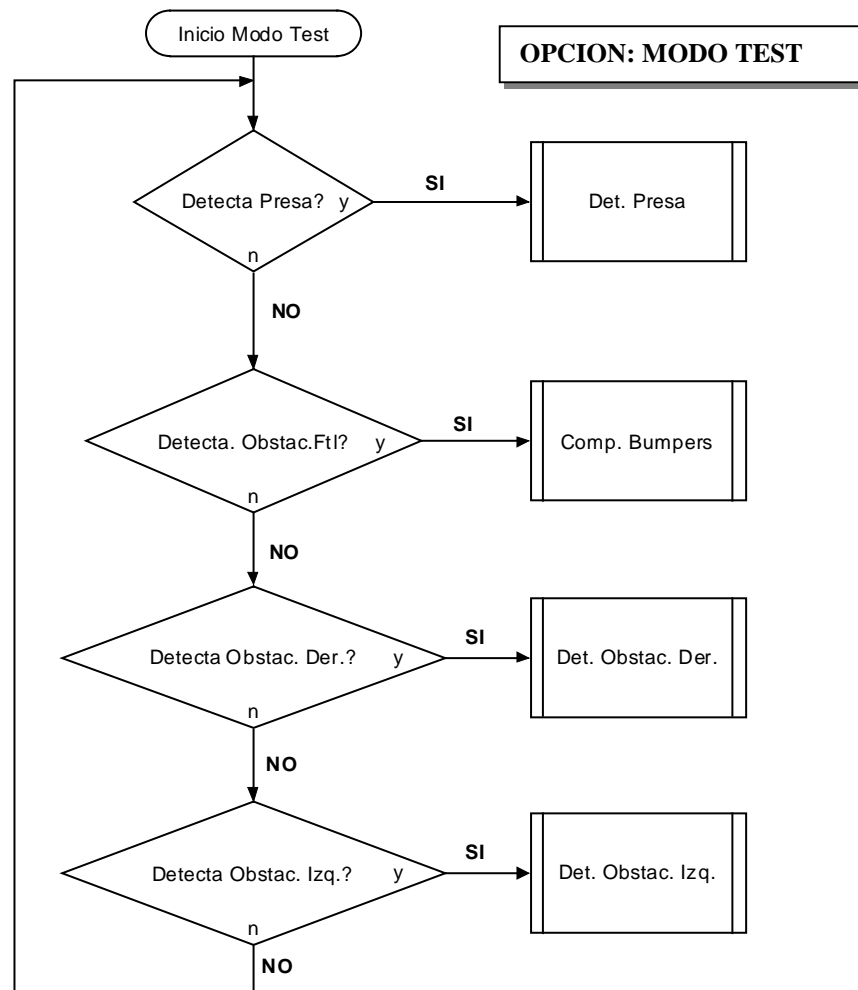
- **Det. Obstac. Ftl.**
- **Det. Obstac. Der.**
- **Det. Obstac. Izq.**

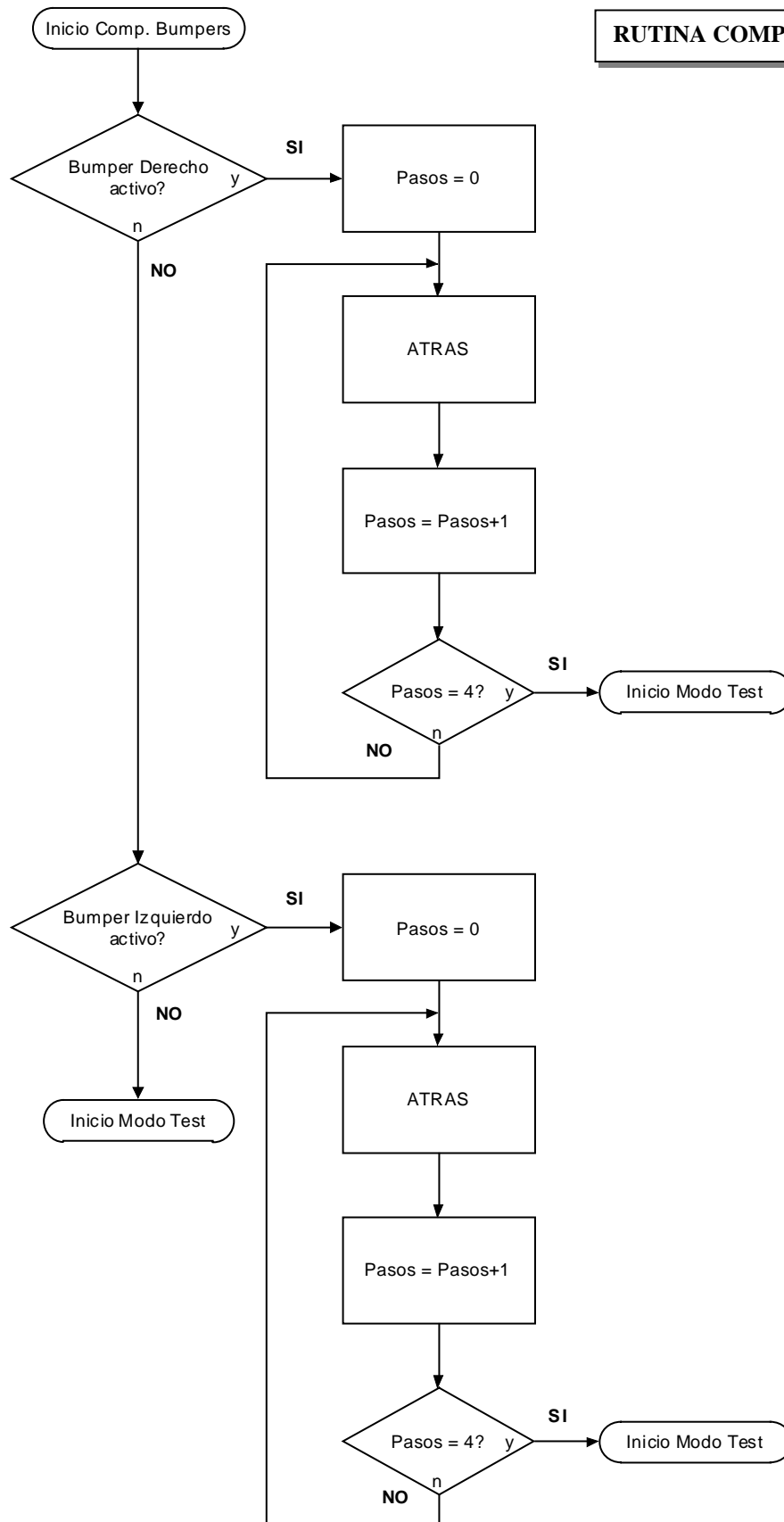
Al igual que se ha hecho al explicar el flujograma de la aplicación básica, para una mejor comprensión del diagrama de flujo, se explican algunas de las expresiones utilizadas en este diagrama de flujo:

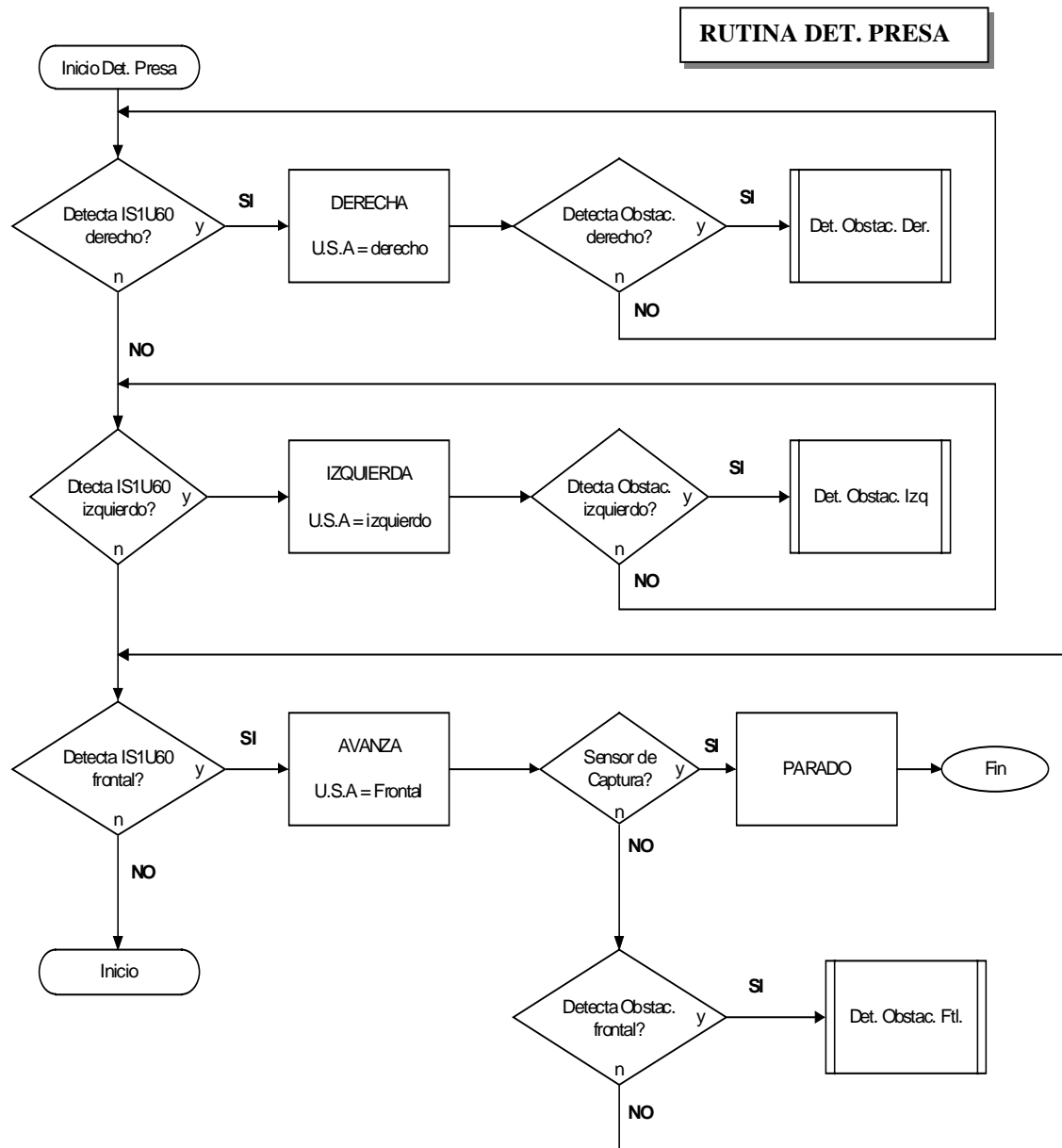
- **Inicio.** Indica el comienzo del programa.
- **Fin.** Indica el final del programa.
- **Pasos.** Es una variable que lleva la cuenta de los pasos del encoder.
- **Tiempo.** Es la variable que lleva la cuenta del tiempo. Se cuenta en segundos.
- **Det. Presa.** Es la misma rutina que la del programa en ensamblador, salvo que se carga una variable, que lleva la información de cuál ha sido el último sensor IS1U60 activo. La variable se denomina **U.S.A.**
- **Busq. Presa.** Esta rutina es la que permite que el microbot busque a la presa, si la pierde. Esto se consigue mirando la variable U.S.A.
- **Det. Obstac. Der.** Es la rutina que se asocia a la detección producida en el detector infrarrojo de obstáculos lateral derecho. Esta rutina se ejecuta cuando el detector lateral derecho de obstáculos se activa.
- **Det. Obstac. Izq.** Es la rutina que se asocia a la detección producida en el detector infrarrojo de obstáculos lateral izquierdo. Esta rutina se ejecuta cuando el detector lateral izquierdo de obstáculos se activa.
- **Det. Obstac. Ftl.** Es la rutina que se ejecuta cuando se produce la detección de alguno de los bumpers frontales.
- **Sensor Captura.** Es el SFH203FA que se utiliza para indicar la captura de la presa.
- **SW1, SW2.** Indica la elección de opción en el menú.

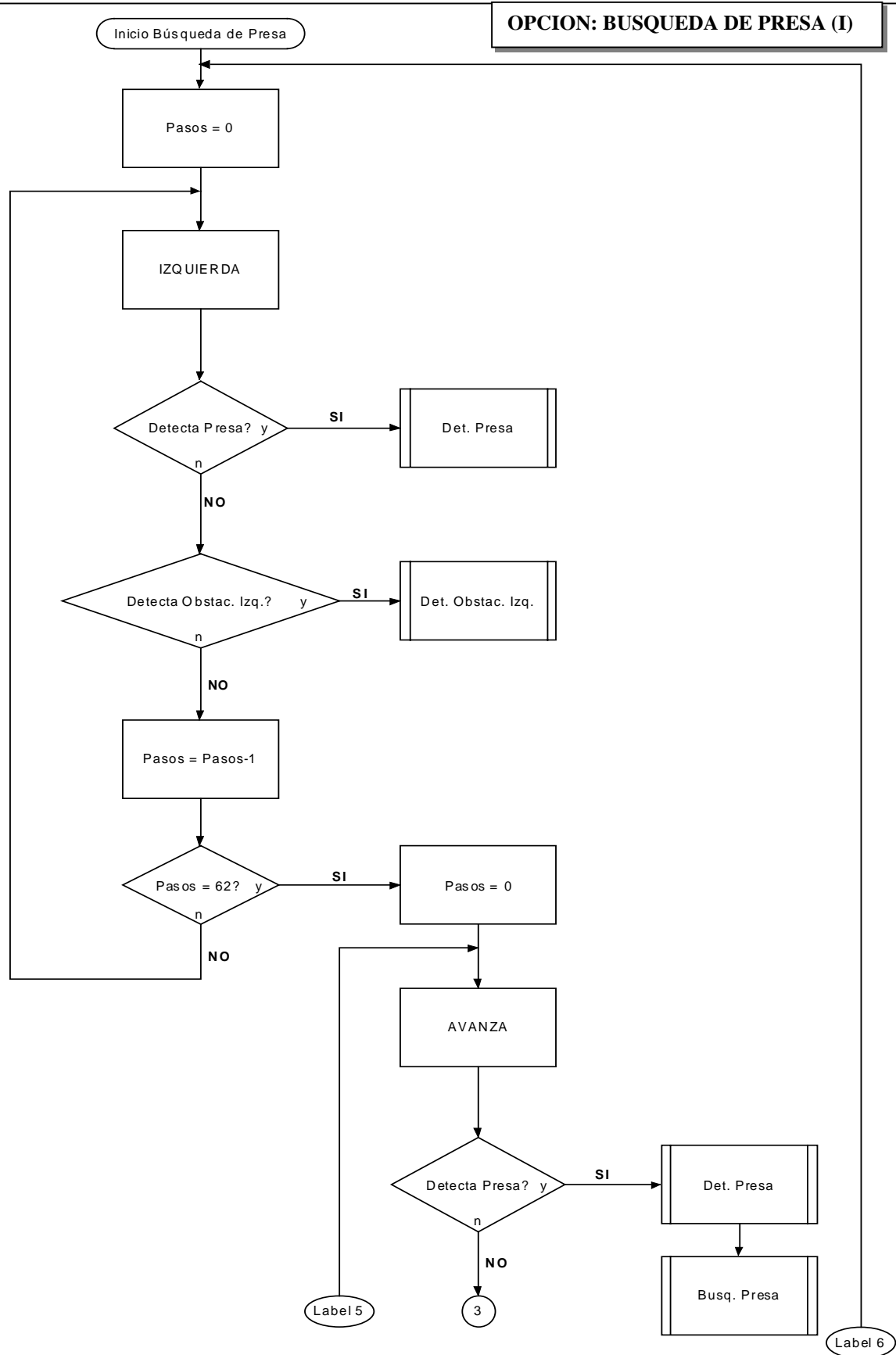
- **Comp. Bumpers.** Es la rutina que se realiza en el modo test para comprobar los bumpers, y que solo hace que el microbot retroceda un poco.
- **Bordear Obstac.** Esta rutina permite a robot bordear el obstáculo tras el que se esconde la presa.
- **Npasos.** Variable necesaria para guardar los pasos que se debe realizar en los recorrido en espiral y en cruz.
- **i, j.** Variables que se usan para no tener que repetir código en la opción de recorrido en cruz.

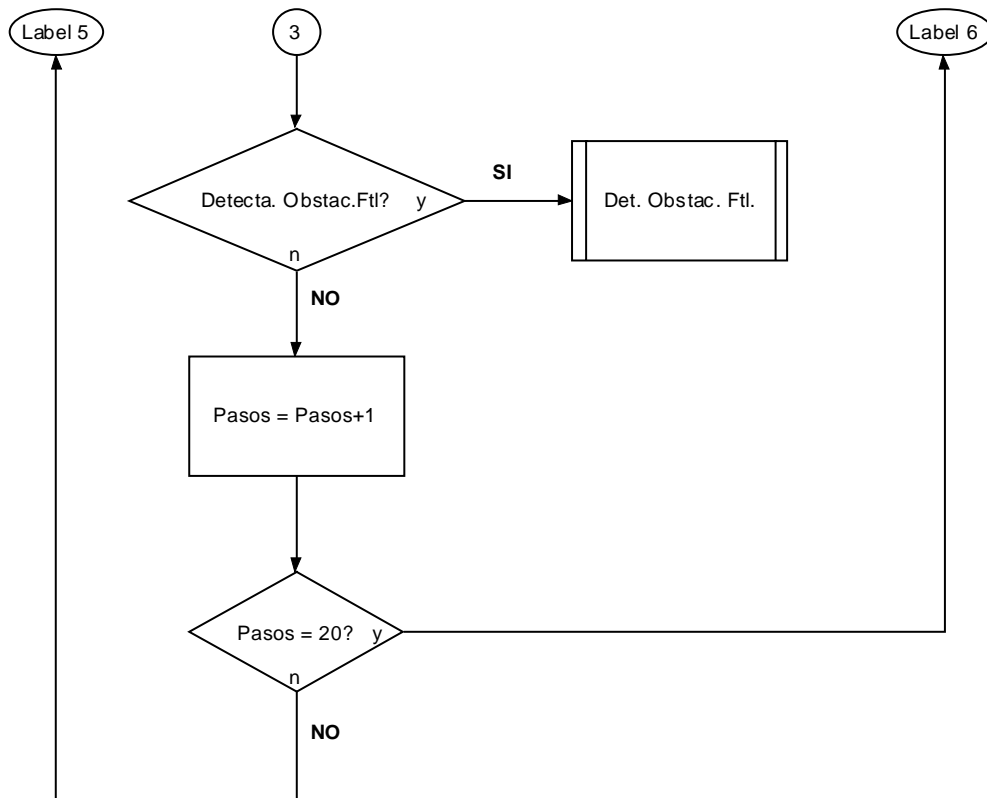
FLUJOGRAMA DE MENU.C

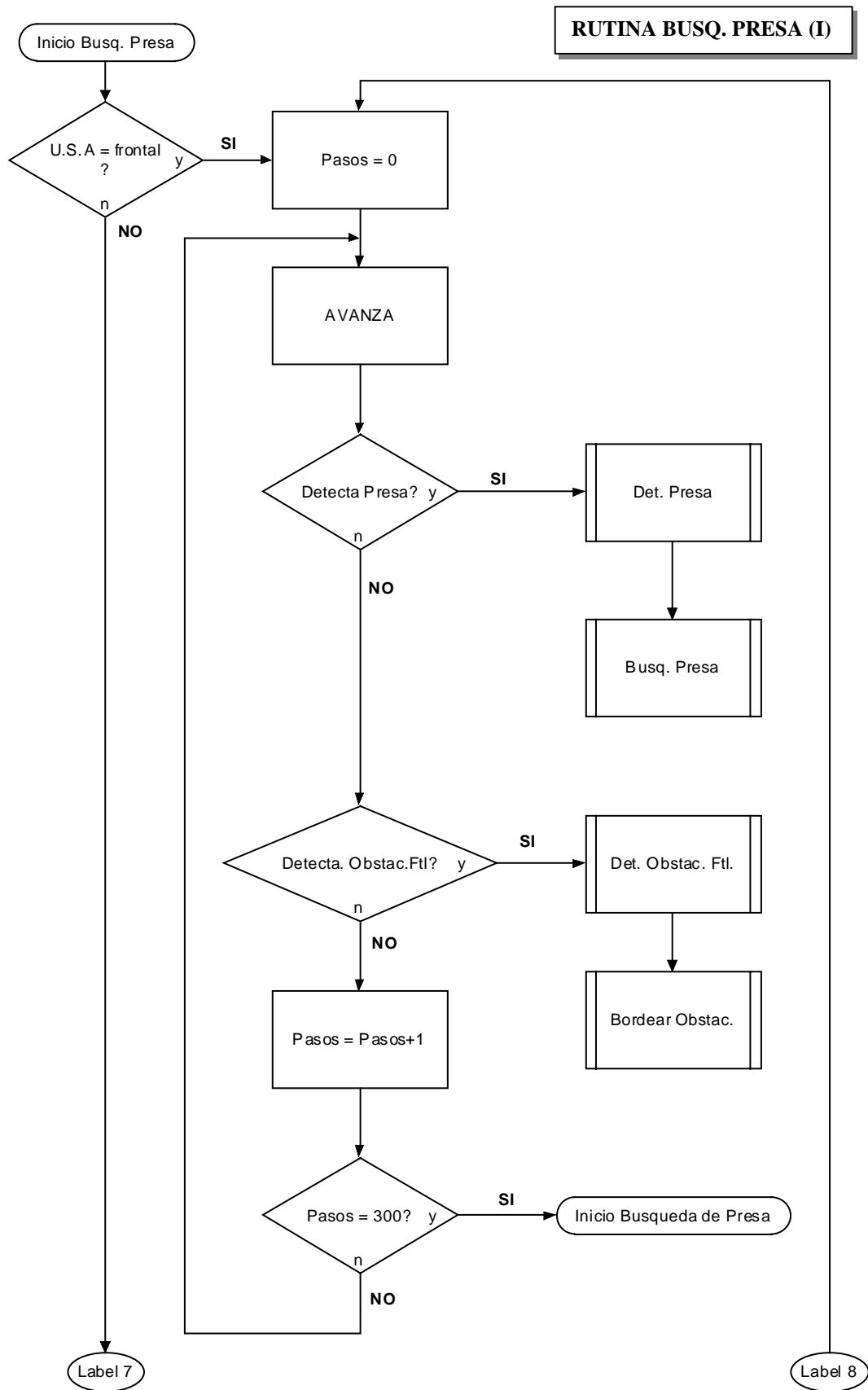


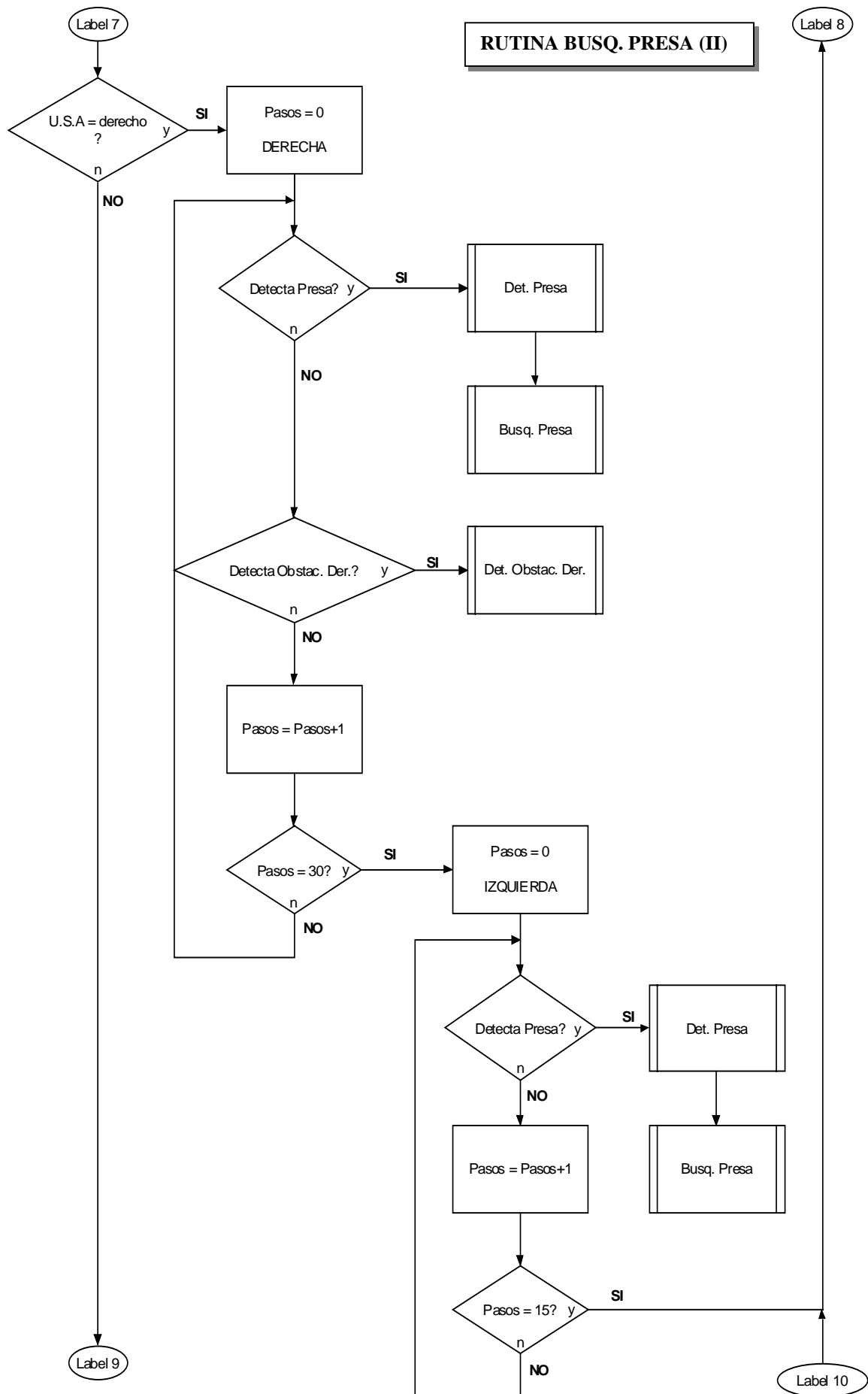


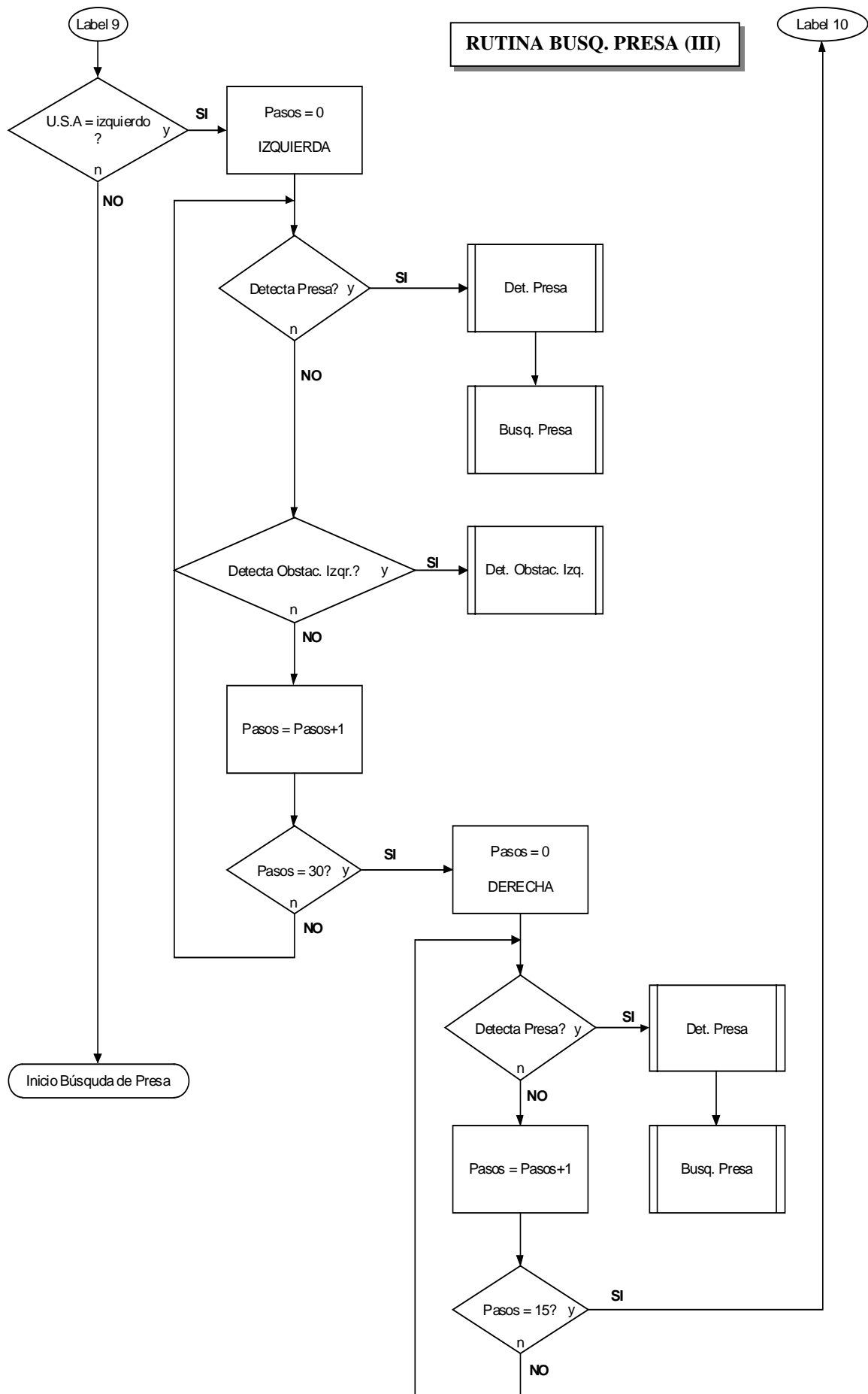


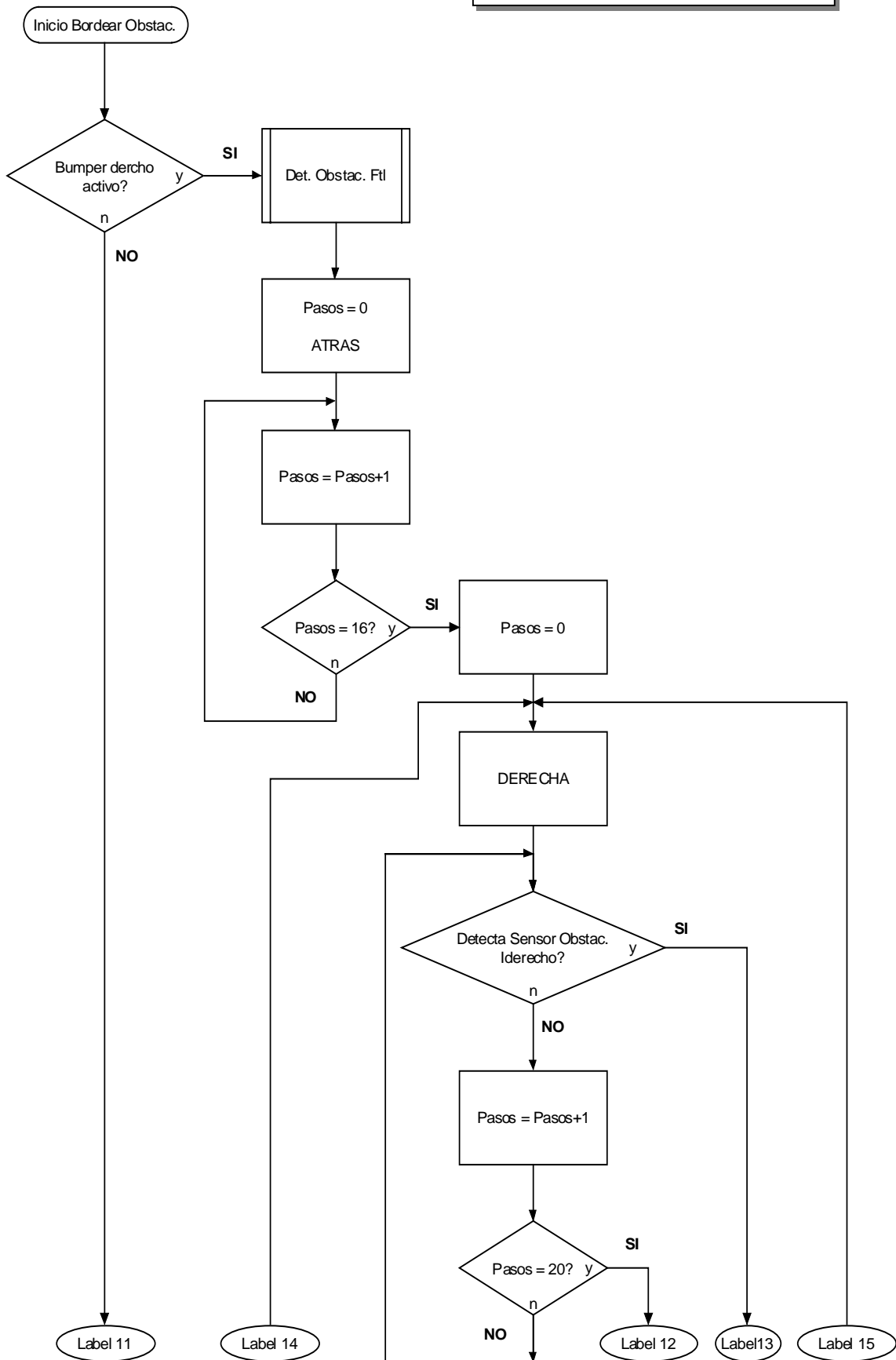


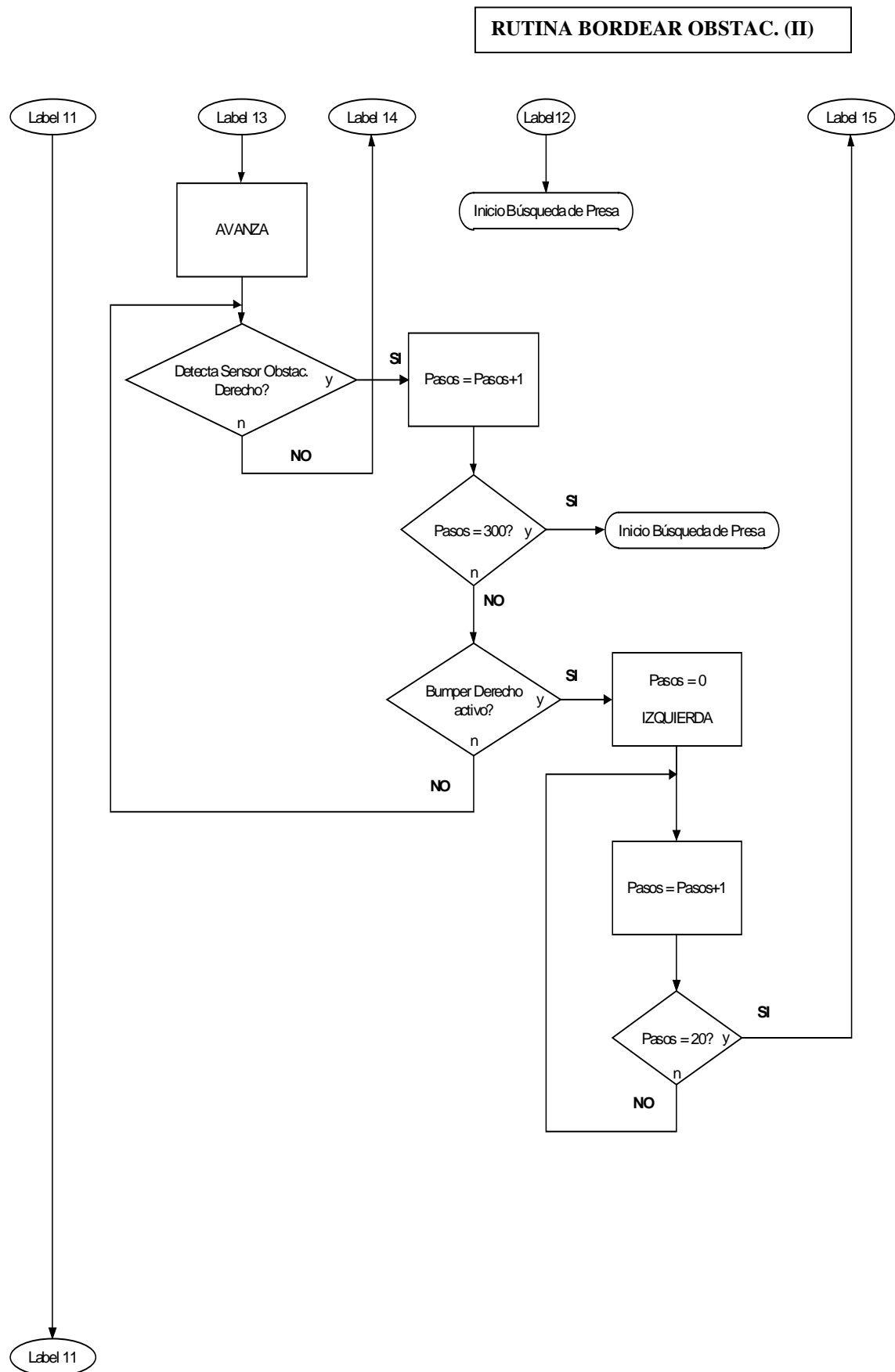
OPCION: BUSQUEDA DE PRESA (II)

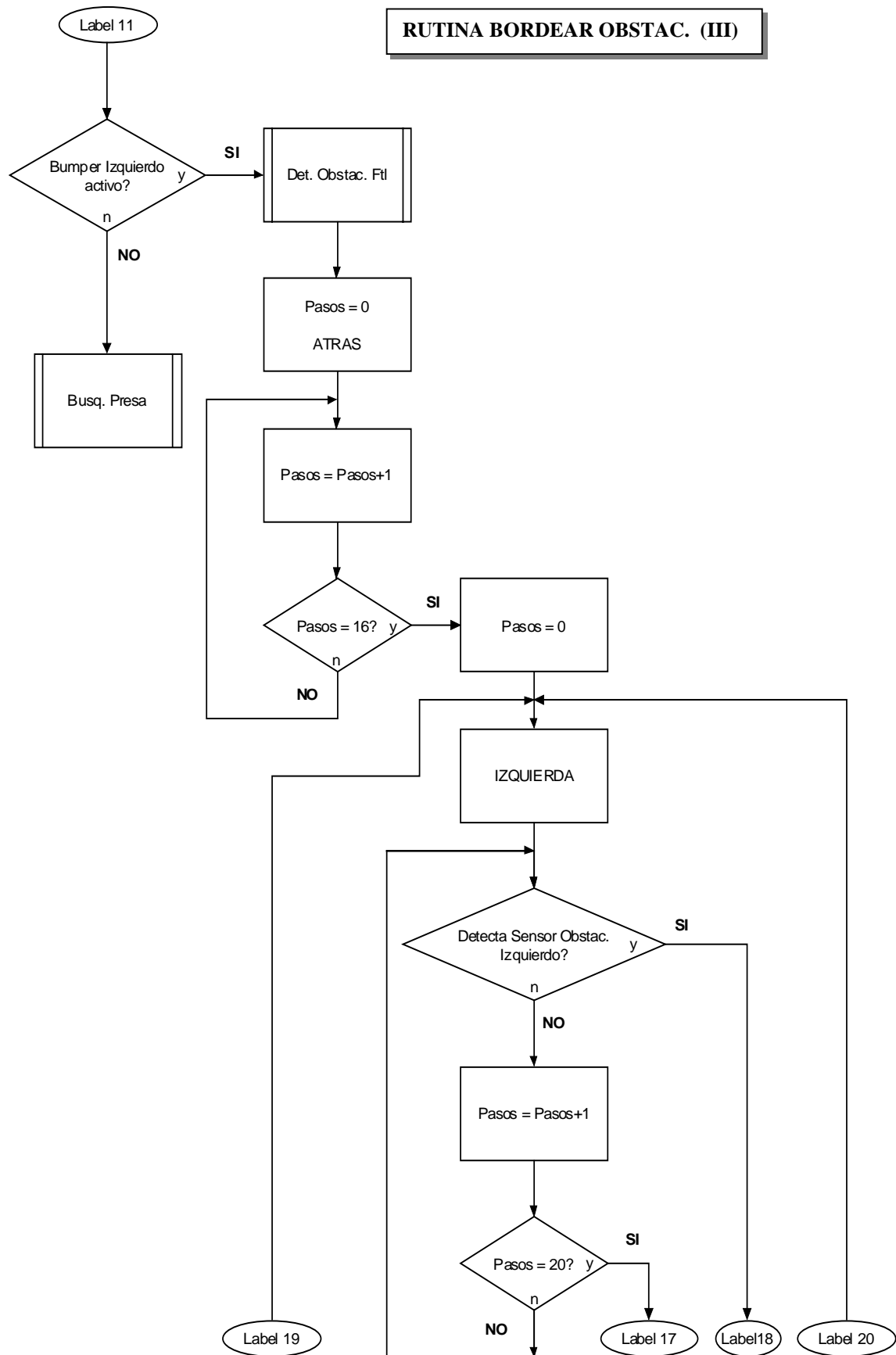


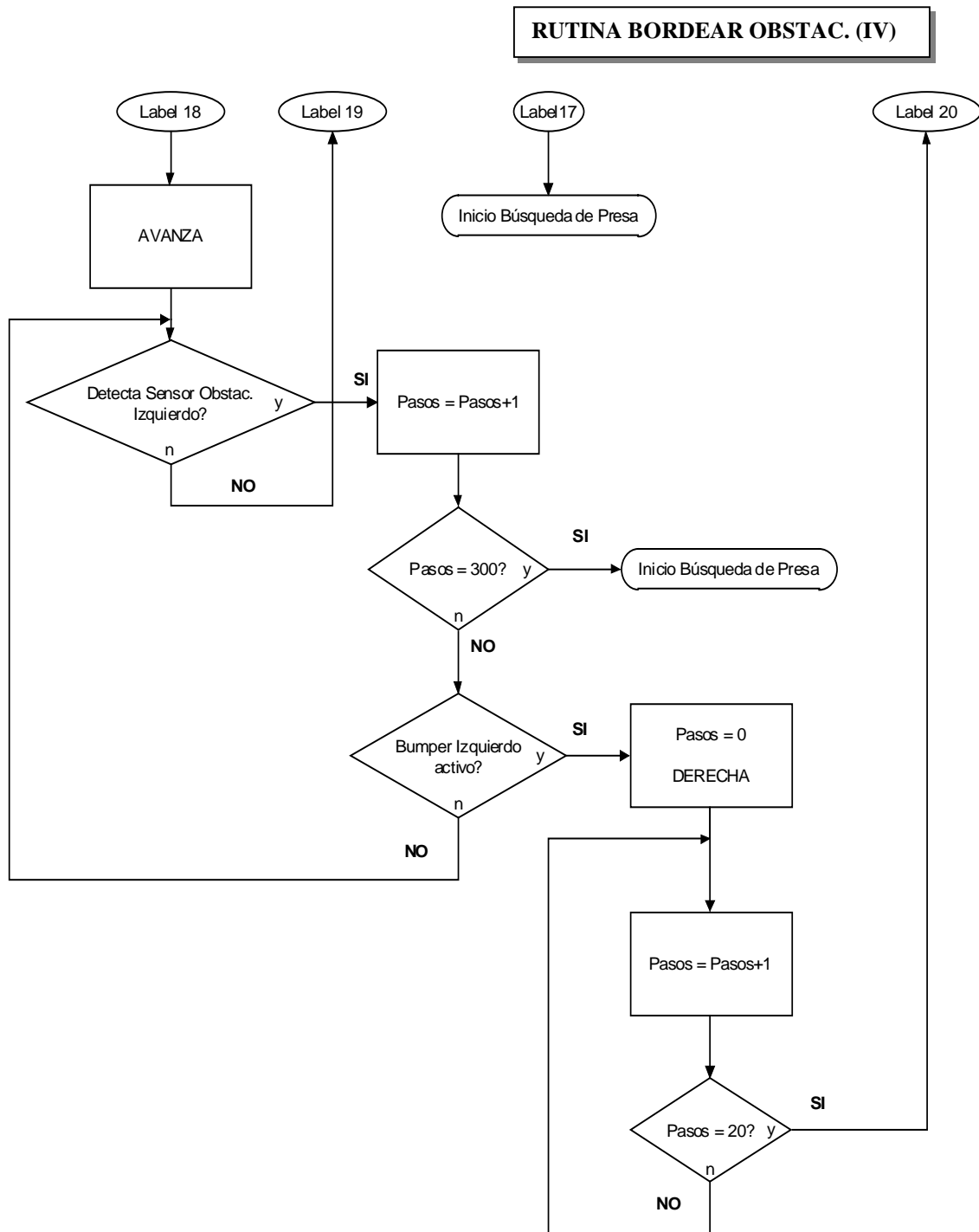


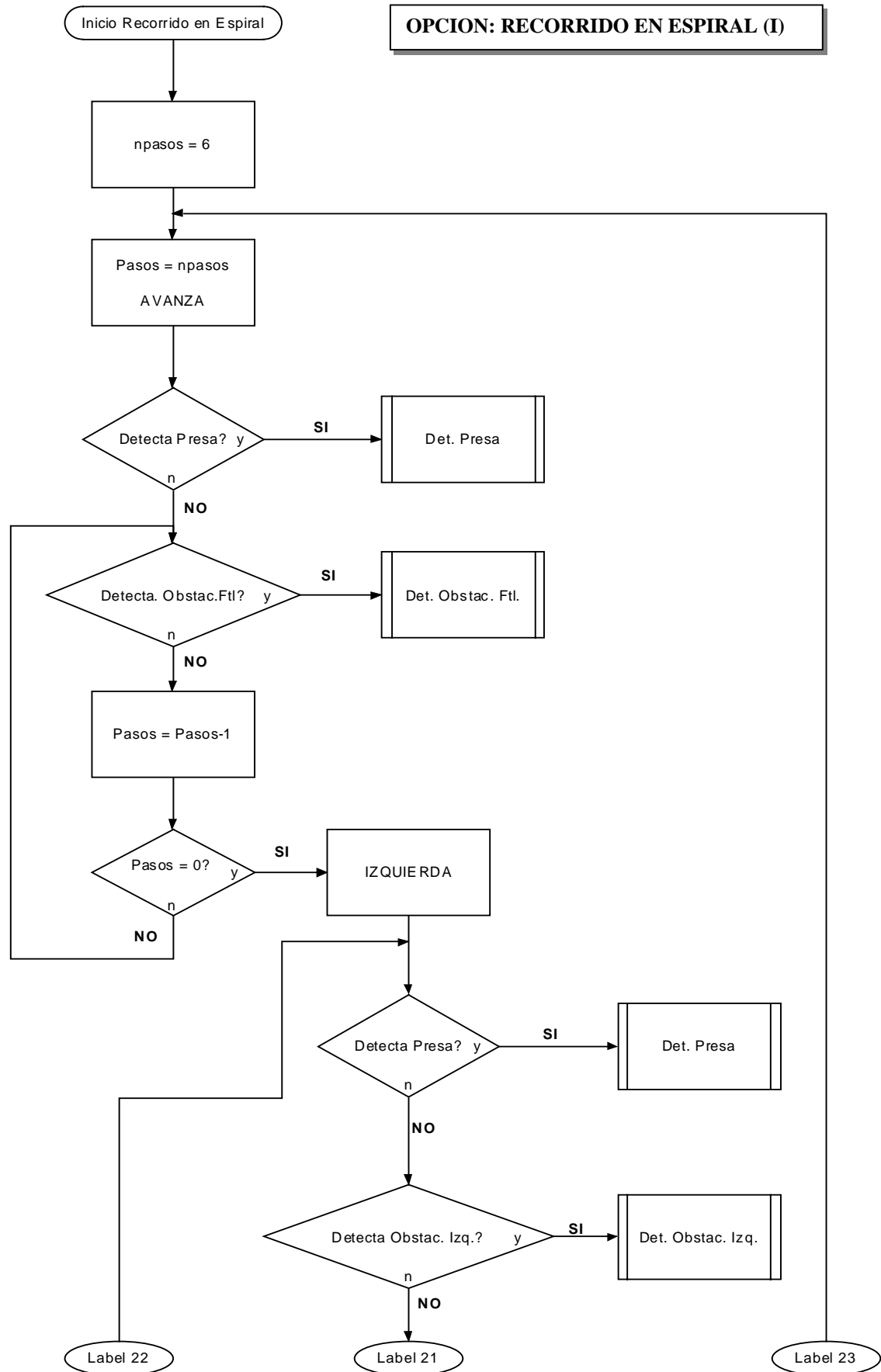


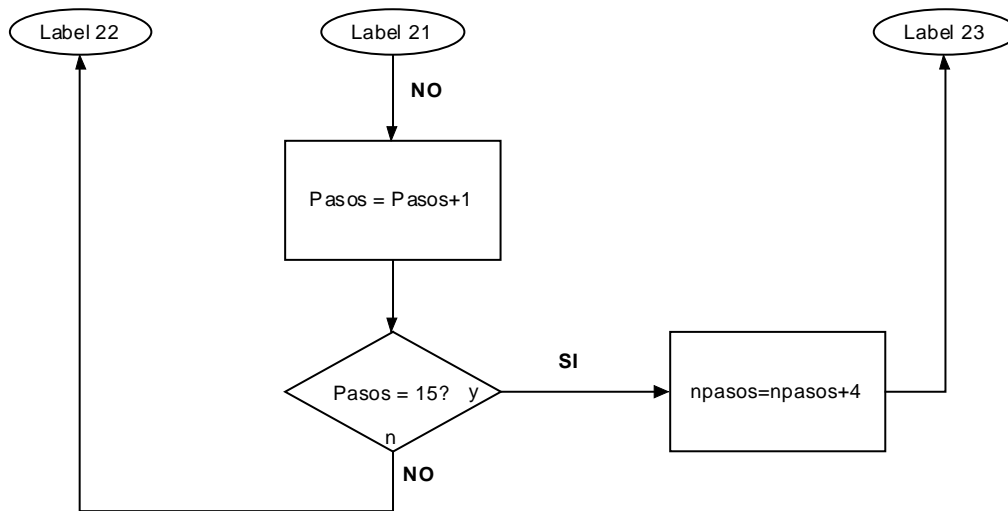
RUTINA BORDEAR OBSTAC. (I)

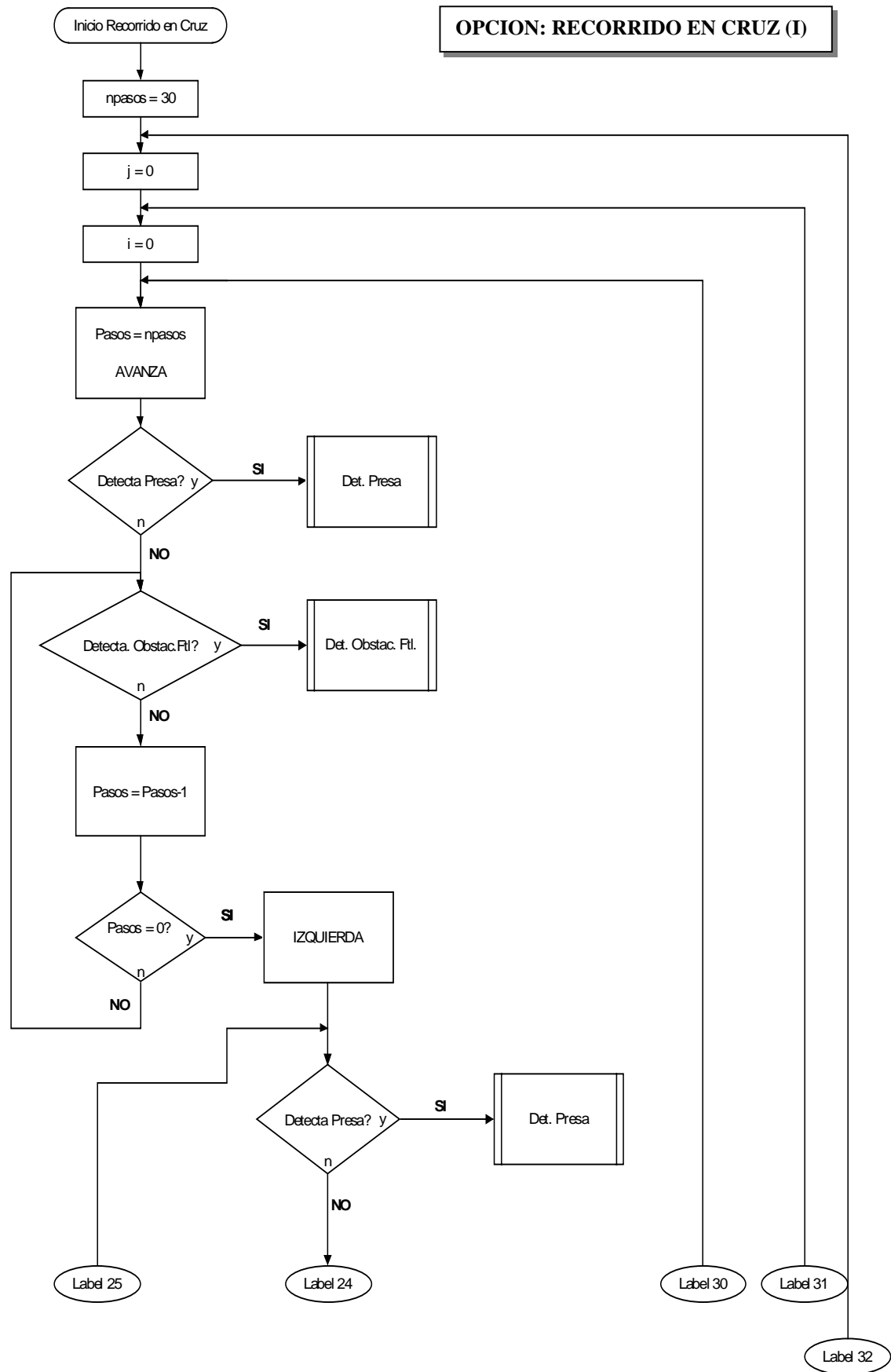




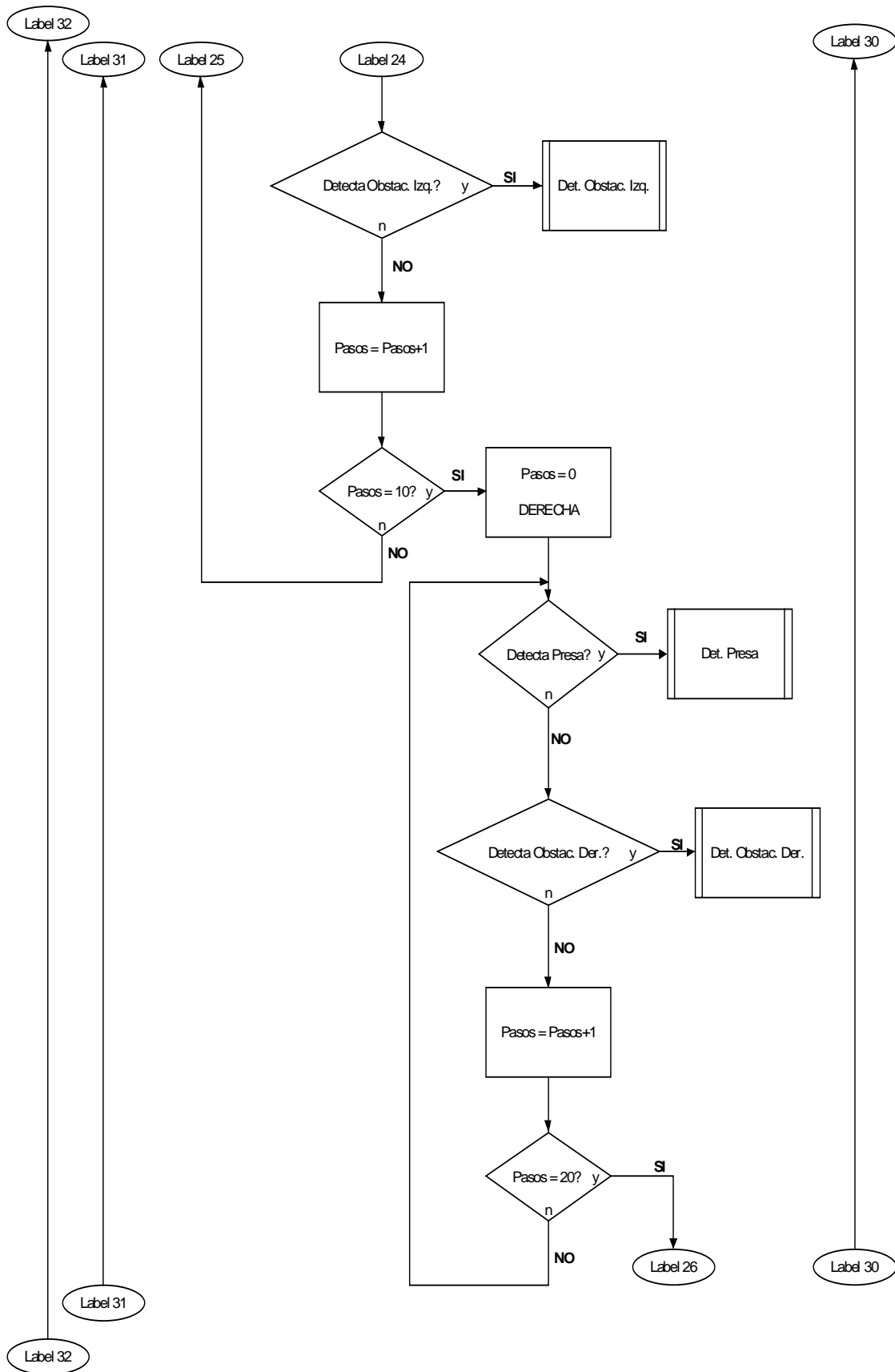


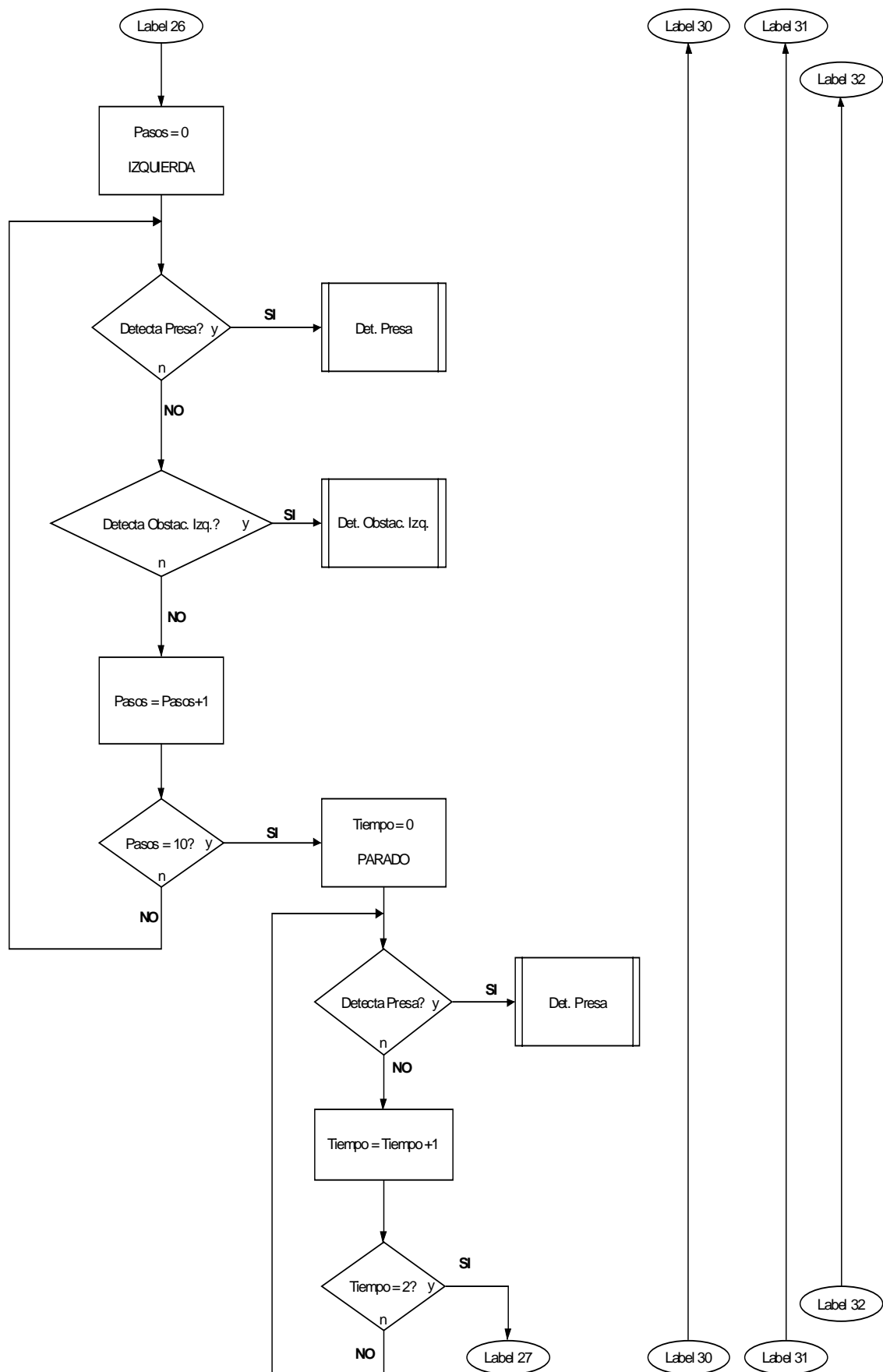


OPCION: RECORRIDO EN ESPIRAL (II)

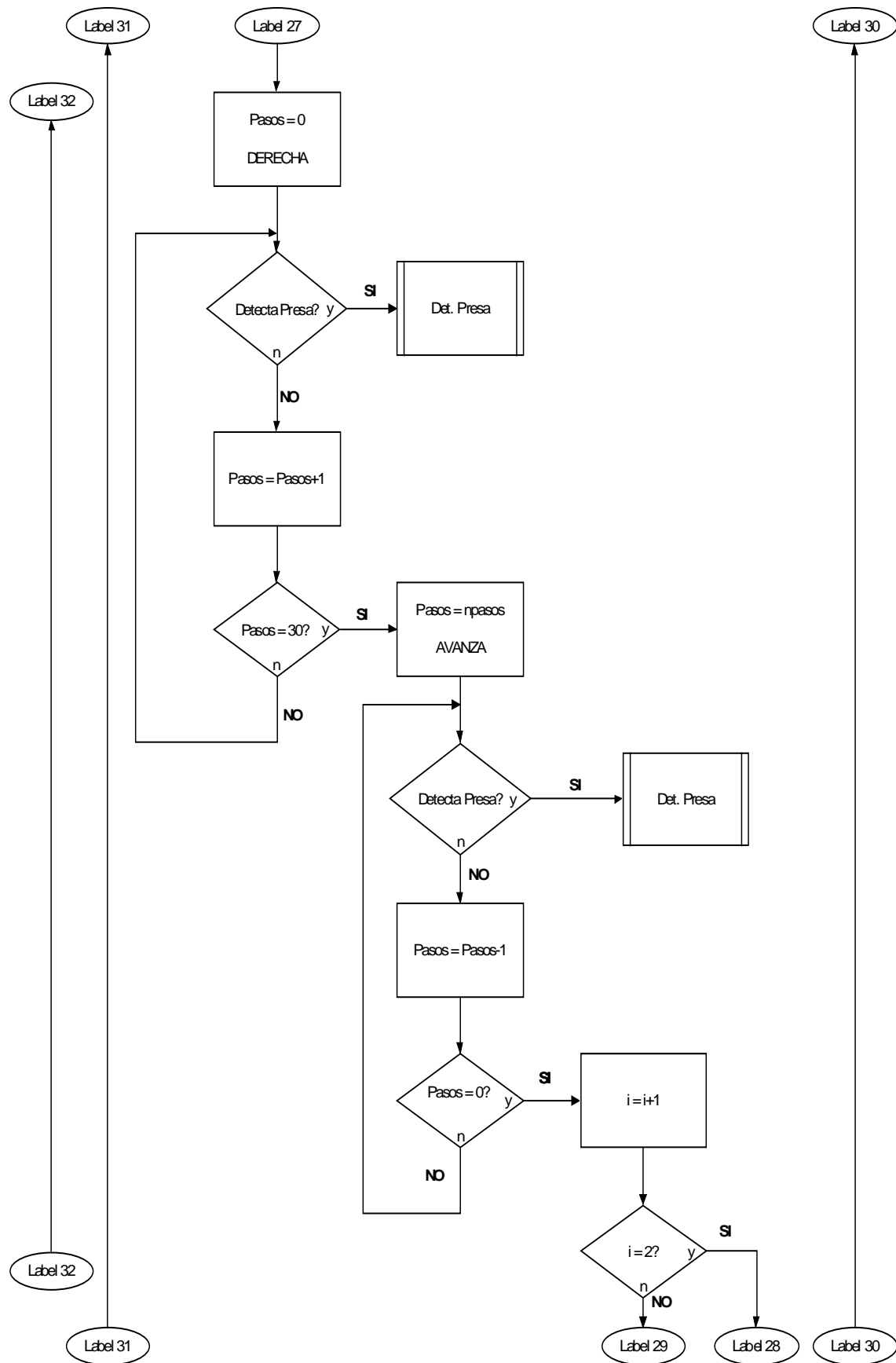


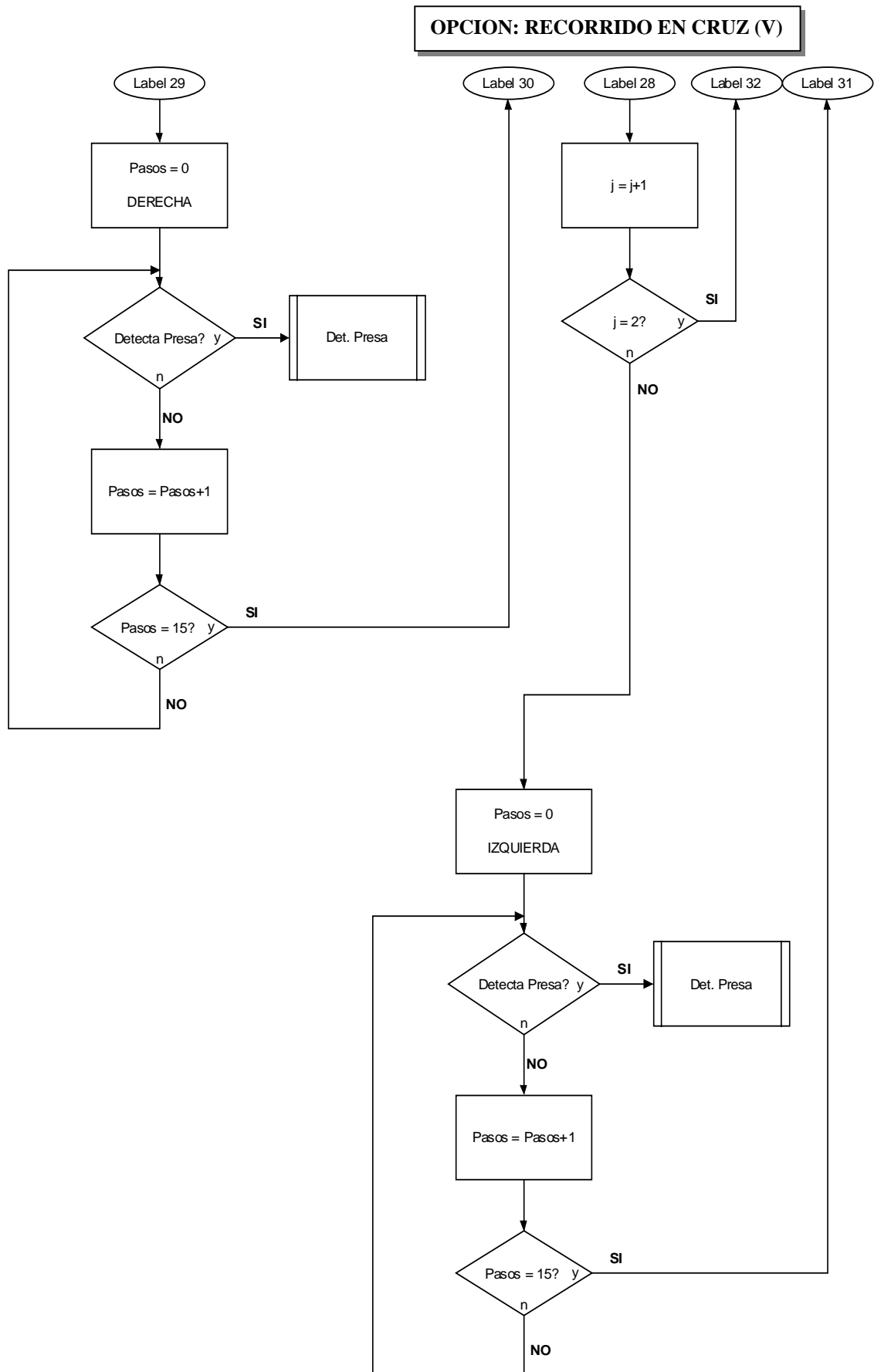
OPCION: RECORRIDO EN CRUZ (II)

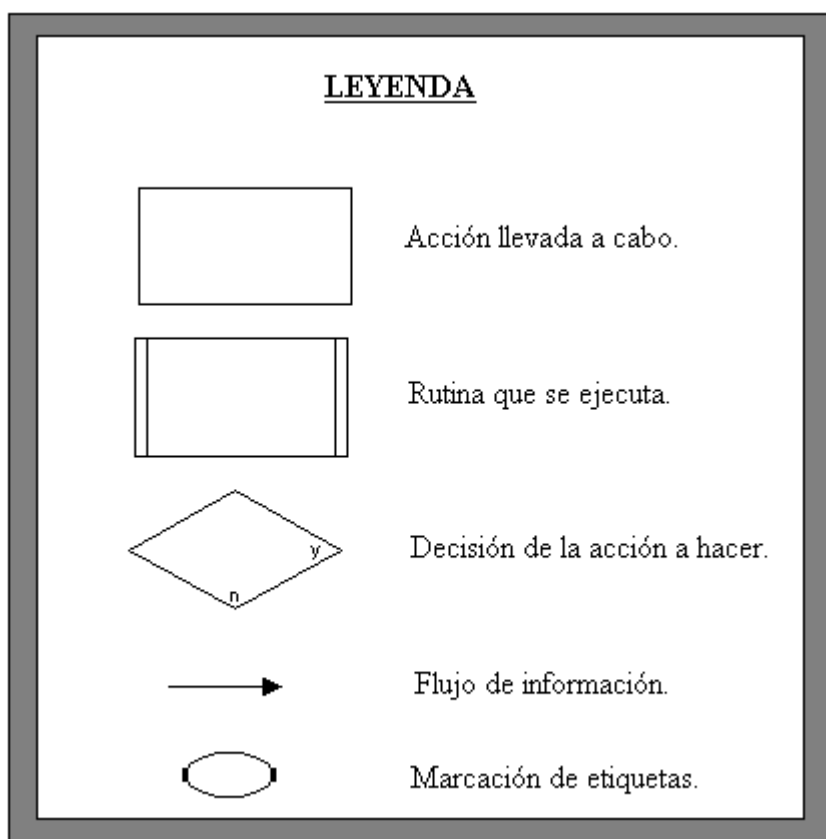


OPCION: RECORRIDO EN CRUZ (III)

OPCION: RECORRIDO EN CRUZ (IV)







CAPITULO 7. PLAN DE PRUEBAS

7. PLAN DE PRUEBAS

Durante el desarrollo del proyecto, y para encontrar la mejor elección del número de sensores a utilizar para la detección de la presa y de los obstáculos, se realizaron una serie de pruebas.

Se va a distinguir entre las pruebas con sensores para la detección de la presa y las pruebas con sensores para la detección de obstáculos.

7.1- Pruebas con los sensores de detección de la presa

***1ª Prueba.** Un único módulo digital IS1U60 situado en la parte delantera del microbot (El ángulo cubierto por el detector es de 80°).*

Al realizar las pruebas con un único IS1U60, se consiguieron buenos resultados en la detección de la presa pero, si en algún momento se perdía a la presa, no se sabía en qué dirección se había movido.

***2ª Prueba.** Tres módulos digitales IS1U60 con los espectros de sensibilidad acoplados (El ángulo cubierto es de 180° aprox.).*

Con tres IS1U60 se consigue detectar a la presa, incluso si ésta está en movimiento. De esta forma, una vez que el microbot detecta a la presa, puede seguirla y atraparla, siempre y cuando la presa no sea más rápida que el microbot depredador.

***3ª Prueba.** Seis módulos digitales IS1U60 cubriendo todas las direcciones (El ángulo cubierto es de prácticamente 360°).*

Con seis IS1U60 se consigue tener detectada a la presa en todo momento, a no ser que haya un obstáculo entre el emisor y el detector de infrarrojos.

Conclusión. En este proyecto se ha decidido usar tres detectores IS1U60. Un solo detector no es una buena opción si la presa está en movimiento. Por otra parte, usar seis detectores permite una localización continuada de la presa, pero supone un gasto innecesario si se tiene en cuenta que con tres detectores es complicado perder a la presa, una vez localizada.

7.2.- Pruebsa con los sensores de detección de obstáculos

***1ª Prueba.** Dos bumpers situados en la parte delantera del microbot.*

El utilizar solo los bumpers para detectar obstáculos tiene dos grandes inconvenientes:

- Se tarda más en esquivar un obstáculo que utilizando los detectores de IR laterales junto a los bumpers.

- No es posible detectar los obstáculos que pueda encontrarse el microbot, al girar.

2ª Prueba. *Dos bumpers situados en la parte delantera y un detector de IR situado a cada lado del microbot.*

Esta configuración permite sortear los obstáculos con mayor rapidez que usando solo los bumpers. Además, permite detectar los obstáculos laterales.

3ª Prueba. *Dos bumpers situados en la parte delantera y dos detectores de IR a cada lado del microbot.*

Como es lógico, a mayor número de sensores, más información del entorno tiene el microbot. Con esta configuración de sensores para la detección de los obstáculos se consigue, aún más rapidez sorteando el obstáculo, que con un detector de IR a cada lado. Además, se tiene una mejor referencia de la forma del obstáculo.

Conclusión. En un principio, se iban a utilizar dos bumpers delanteros y dos detectores de IR laterales a cada lado del microbot. La única razón por la que no se ha utilizado esta opción es por el gasto de potencia que conlleva.

4ª Prueba. *Prueba en esquinas.*

Se han realizado pruebas del comportamiento del microbot depredador ante esquinas que forman ángulos de 90° , más de 90° y menos de 90° . Los resultados de estas pruebas han sido: Al microbot no le cuesta ningún trabajo salir de una esquina de 90° o mayor de 90° , sin embargo, tiene algunos problemas para salir de las esquinas de menos de 90° , incluso, en alguna ocasión, puede quedarse atascado durante un tiempo intentando salir de la esquina.

CAPITULO 8. PROBLEMAS SURGIDOS EN EL PROYECTO

8. PROBLEMAS SURGIDOS EN EL PROYECTO

Aunque han surgido varios problemas durante el desarrollo del proyecto, destacan dos: Alimentación y Compiladores.

8.1.- Problemas con la alimentación

En el microbot depredador existen muchos elementos que consumen potencia, tales como el microcontrolador, los sensores digitales IS1U60, los sensores analógicos (SFH485, SFH 203FA), los motores y la memoria RAM externa (tarjeta CT256).

Como ya se ha comentado en el capítulo 5, son necesarias tres fuentes de alimentación. Además, la memoria RAM externa consume mucho y debe ir alimentada entre 5'5 y 6 voltios. Estas dos circunstancias hacen que no se puedan utilizar baterías normales (En este proyecto se han usado 4 pilas AA para cada una de las fuentes de alimentación) para alimentar a la memoria, por lo que, lo único que se puede hacer es utilizar un transformador o una fuente de alimentación, conectada a la red eléctrica. Esto supone tener que llevar la alimentación hasta el microbot a través de cables, lo que le resta movilidad y puede ocasionar que la memoria externa pierda la alimentación por un momento, debido a un movimiento del cable. Al perder la alimentación, se pierde el programa cargado en la RAM externa, lo que significa que hay que cargar el programa de nuevo en la memoria.

8.2.- Problema con los compiladores

Los compiladores utilizados en este proyecto han sido el AS11 y el ICC11. Estos son GPL, es decir, son herramientas libres, sin embargo estos compiladores están muy limitados.

El AS11 no permite realizar saltos condicionales a etiquetas mayores de 256 bytes. Asimismo, tampoco permite los saltos hacia delante con la instrucción JSR

Por su parte, el compilador de Imagecraft no está preparado para realizar programas de extensión mayor que una dada, dando errores en la compilación, tales como “ *Assertion Failure at line...*”.

Se tiene conocimiento de otros compiladores que funcionan mejor. Estos compiladores son: el RAS 1.0.2, para compilar programas en ensamblador, y el GNU, que es un compilador de C que funciona en entorno UNIX.

CAPITULO 9. CONCLUSIONES

9. CONCLUSIONES

En este apartado se pretende dar una orientación de cuál ha sido la evolución del proyecto desde el planteamiento inicial hasta llegar al microbot depredador, resultado final del proyecto.

En un principio se pretendía que el microbot depredador interactuara con otro microbot, el cual haría las funciones de presa. Para conseguirlo se pensó en detectores de infrarrojos modulados, los cuales suponen un menor coste que utilizar otro tipo de sensores, como radiofrecuencia o ultrasonidos. Estos detectores de infrarrojos disponían de modulación a distinta frecuencia con lo que, sobre un microbot se situaban los detectores modulados a una frecuencia, y sobre el otro se situaban los detectores a otra frecuencia distinta. De esta forma solo se detectarían las emisiones realizadas por el otro robot. El problema fue que los detectores a frecuencia distinta de 38 KHz son casi imposibles de conseguir en pequeñas cantidades, por lo que se tuvo que desechar la opción de utilizar detectores a distinta frecuencia.

Una vez que se consiguieron los detectores modulados a 38 KHz (módulos digitales de infrarrojos IS1U60), se vio que el microbot podía tropezar con obstáculos mientras seguía a la presa, por lo que eran necesarios sensores que informaran al microbot de la posición de los obstáculos. Para la detección de obstáculos, primero se incorporaron dos bumpers situados en la parte delantera del microbot. Viendo que el microbot tardaba mucho en sortear el obstáculo y no podía evitar chocar con obstáculos cuando giraba, se consideró oportuno incluir unos sensores de infrarrojos (diodo emisor de infrarrojos SFH485 y fotodiodo detector de infrarrojos SFH203FA), situados a los laterales del microbot, que permiten sortear los obstáculos con mayor rapidez y detectar la presencia de obstáculos laterales.

Con los módulos digitales de infrarrojos IS1U60 y los sensores para la detección de obstáculos, se podía seguir a la presa y evitar los obstáculos que pudieran existir entre el microbot depredador y la presa. Sin embargo, ¿Cómo sabía el microbot si había capturado a la presa? Con los módulos digitales no era posible, ya que la salida que proporcionan es digital ('0', detectado; '1', no detectado). Fue necesario utilizar un sensor analógico que informara de la captura de la presa, y ya que se disponía de fotodiodos detectores de infrarrojos SFH203FA, se consideró oportuno utilizarlo para este fin.

En el planteamiento inicial de este proyecto se propuso utilizar el microbot TRITT para simular el comportamiento de un depredador. Este microbot se diferencia del TRITTON en que no incorpora la tarjeta CT256 de expansión de memoria. Para aprovechar al máximo la escasa memoria de la que se disponía (256 bytes de RAM y 512 bytes de EEPROM), se pensó que lo mejor era programar en lenguaje ensamblador, de forma que el código fuera lo más eficiente posible. Algún tiempo después se pudo disponer de la tarjeta CT256, lo que permitió desarrollar programas en lenguaje C de mayor complejidad.

La conclusión a la que se ha llegado tras todo el desarrollo del proyecto es que, aunque se ha conseguido cumplir el principal objetivo de este proyecto con un coste bastante bajo (unas 3.500 ptas.), es posible realizar multitud de mejoras al microbot depredador. Algunas de estas posibles mejoras son: utilizar radiofrecuencia para la detección de la presa, utilizar sensores de infrarrojos analógicos modulados para la detección de la presa y los obstáculos o programar algoritmos de intercepción de la presa, para que la captura se llevara a cabo lo antes posible. Las mejoras que se pueden realizar sólo están limitadas por la imaginación y los recursos de que se disponga.

ANEXO A

HOJAS DE CARACTERÍSTICAS DE INVERSOR SCHMITT-TRIGGER SN74LS14 Y TRANSISTOR 2N2222A

SN5414, SN54LS14, SN7414, SN74LS14 HEX SCHMITT-TRIGGER INVERTERS

DECEMBER 1983—REVISED MARCH 1988

- Operation from Very Slow Edges
- Improved Line-Receiving Characteristics
- High Noise Immunity

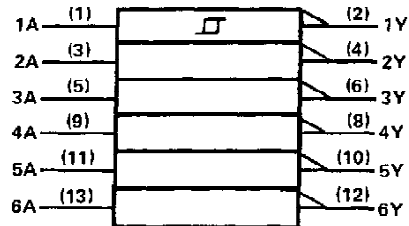
description

Each circuit functions as an inverter, but because of the Schmitt action, it has different input threshold levels for positive (V_{T+}) and for negative going (V_{T-}) signals.

These circuits are temperature-compensated and can be triggered from the slowest of input ramps and still give clean, jitter-free output signals.

The SN5414 and SN54LS14 are characterized for operation over the full military temperature range of -55°C to 125°C . The SN7414 and the SN74LS14 are characterized for operation from 0°C to 70°C .

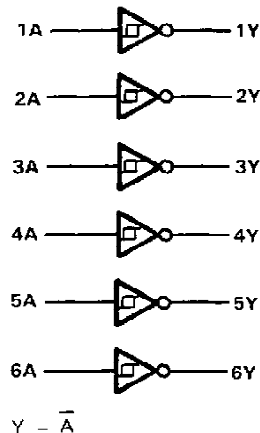
logic symbol†



† This symbol is in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12.

Pin numbers shown are for D, J, N, and W packages.

logic diagram (positive logic)

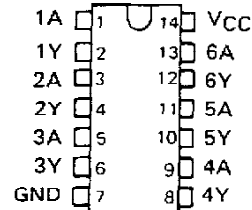


SN5414, SN54LS14 . . . J OR W PACKAGE

SN7414 . . . N PACKAGE

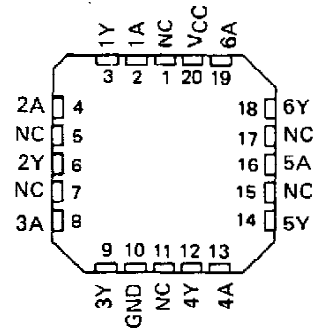
SN74LS14 . . . D OR N PACKAGE

(TOP VIEW)



SN54LS14 . . . FK PACKAGE

(TOP VIEW)



NC—No internal connection

PRODUCTION DATA documents contain information current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

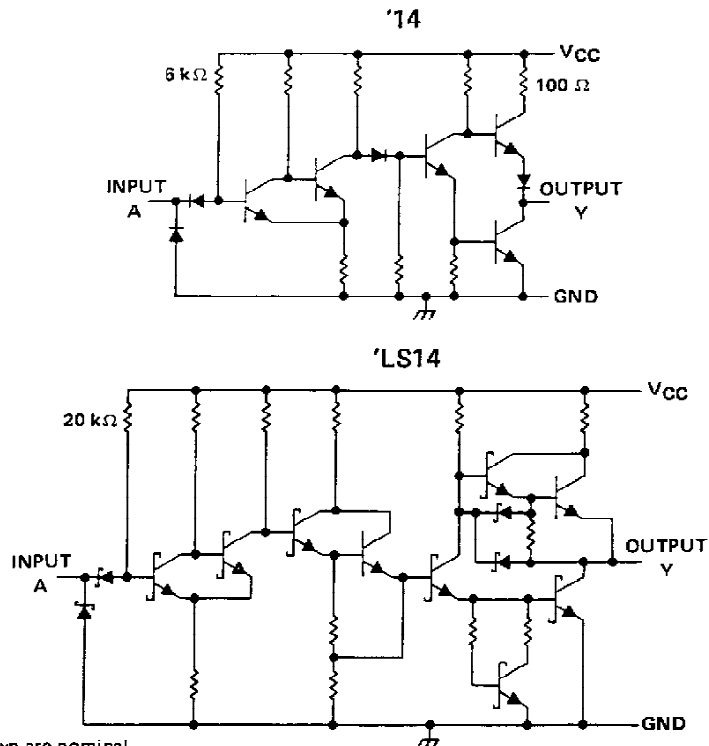
TEXAS
INSTRUMENTS

POST OFFICE BOX 655012 • DALLAS, TEXAS 75265

SN5414, SN54LS14, SN7414, SN74LS14

HEX SCHMITT-TRIGGER INVERTERS

schematics



Resistor values shown are nominal.

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V _{CC} (see Note 1)	7 V
Input voltage: '14	5.5 V
'LS14	7 V
Operating free-air temperature: SN54'	−55°C to 125°C
SN74'	0°C to 70°C
Storage temperature range	−65°C to 150°C

NOTE 1: Voltage values are with respect to network ground terminal.

TEXAS
INSTRUMENTS

POST OFFICE BOX 655012 • DALLAS, TEXAS 75265

SN5414, SN7414

HEX SCHMITT-TRIGGER INVERTERS

recommended operating conditions

	SN5414			SN7414			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
V_{CC} Supply voltage	4.5	5	5.5	4.75	5	5.25	V
I_{OH} High-level output current			-0.8			-0.8	mA
I_{OL} Low-level output current			16			16	mA
T_A Operating free-air temperature	-55		125	0		70	°C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS†	MIN	TYP‡	MAX	UNIT
V_{T+}	$V_{CC} = 5\text{ V}$	1.5	1.7	2	V
V_{T-}	$V_{CC} = 5\text{ V}$	0.6	0.9	1.1	V
Hysteresis ($V_{T+} - V_{T-}$)	$V_{CC} = 5\text{ V}$	0.4	0.8		V
V_{IK}	$V_{CC} = \text{MIN.}$, $I_I = -12\text{ mA}$			-1.5	V
V_{OH}	$V_{CC} = \text{MIN.}$, $V_I = 0.6\text{ V}$, $I_{OH} = -0.8\text{ mA}$	2.4	3.4		V
V_{OL}	$V_{CC} = \text{MIN.}$, $V_I = 2\text{ V}$, $I_{OL} = 16\text{ mA}$		0.2	0.4	V
I_{T+}	$V_{CC} = 5\text{ V}$, $V_I = V_{T+}$	-0.43			mA
I_{T-}	$V_{CC} = 5\text{ V}$, $V_I = V_{T-}$	-0.56			mA
I_I	$V_{CC} = \text{MAX.}$, $V_I = 5.5\text{ V}$			1	mA
I_{IH}	$V_{CC} = \text{MAX.}$, $V_{IH} = 2.4\text{ V}$			40	µA
I_{IL}	$V_{CC} = \text{MAX.}$, $V_{IL} = 0.4\text{ V}$		-0.8	-1.2	mA
$I_{OS}\$$	$V_{CC} = \text{MAX.}$	-18		-55	mA
I_{CCH}	$V_{CC} = \text{MAX.}$		22	36	mA
I_{CCL}	$V_{CC} = \text{MAX.}$		39	60	mA

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

‡ All typical values are at $V_{CC} = 5\text{ V}$, $T_A = 25^\circ\text{C}$.

§ Not more than one output should be shorted at a time.

switching characteristics, $V_{CC} = 5\text{ V}$, $T_A = 25^\circ\text{C}$

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	A	Y	$R_L = 400\ \Omega$, $C_L = 15\text{ pF}$		15	22	ns
t_{PHL}					15	22	ns

TEXAS
INSTRUMENTS

POST OFFICE BOX 655012 • DALLAS, TEXAS 75265

SN54LS14, SN74LS14 **HEX SCHMITT-TRIGGER INVERTERS**

recommended operating conditions

	SN54LS14			SN74LS14			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
V_{CC} Supply voltage	4.5	5	5.5	4.75	5	5.25	V
I_{OH} High-level output current			-0.4			-0.4	mA
I_{OL} Low-level output current			4			8	mA
T_A Operating free-air temperature	-55		125	0		70	°C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS†	SN54LS14			SN74LS14			UNIT	
		MIN	TYP‡	MAX	MIN	TYP‡	MAX		
V_{T+}	$V_{CC} = 5\text{ V}$	1.4	1.6	1.9	1.4	1.6	1.9	V	
V_{T-}	$V_{CC} = 5\text{ V}$	0.5	0.8	1	0.5	0.8	1	V	
Hysteresis ($V_{T+} - V_{T-}$)	$V_{CC} = 5\text{ V}$	0.4	0.8		0.4	0.8		V	
V_{IK}	$V_{CC} = \text{MIN}$, $I_I = -18\text{ mA}$			-1.5			-1.5	V	
V_{OH}	$V_{CC} = \text{MIN}$, $V_I = 0.5\text{ V}$, $I_{OH} = -0.4\text{ mA}$	2.5	3.4		2.7	3.4		V	
V_{OL}	$V_{CC} = \text{MIN}$, $V_I = 1.9\text{ V}$	$I_{OL} = 4\text{ mA}$		0.25	0.4	0.25		0.4	V
		$I_{OL} = 8\text{ mA}$				0.35		0.5	
I_{T+}	$V_{CC} = 5\text{ V}$, $V_I = V_{T+}$	-0.14			-0.14			mA	
I_{T-}	$V_{CC} = 5\text{ V}$, $V_I = V_{T-}$	-0.18			-0.18			mA	
I_I	$V_{CC} = \text{MAX}$, $V_I = 7\text{ V}$	0.1			0.1			mA	
I_{IH}	$V_{CC} = \text{MAX}$, $V_{IH} = 2.7\text{ V}$	20			20			µA	
I_{IL}	$V_{CC} = \text{MAX}$, $V_{IL} = 0.4\text{ V}$	-0.4			-0.4			mA	
$I_{OS}§$	$V_{CC} = \text{MAX}$	-20		-100	-20		-100	mA	
I_{CCH}	$V_{CC} = \text{MAX}$	8.6		16	8.6		16	mA	
I_{CCL}	$V_{CC} = \text{MAX}$	12		21	12		21	mA	

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

‡ All typical values are at $V_{CC} = 5\text{ V}$, $T_A = 25^\circ\text{C}$.

§ Not more than one output should be shorted at a time, and duration of the short-circuit should not exceed one second.

switching characteristics, $V_{CC} = 5\text{ V}$, $T_A = 25^\circ\text{C}$

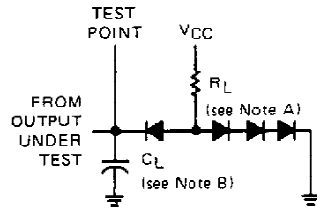
PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	A	Y	$R_L = 2\text{ k}\Omega$, $C_L = 15\text{ pF}$		15	22	ns
t_{PHL}					15	22	ns

TEXAS
INSTRUMENTS

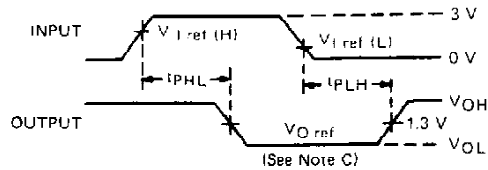
POST OFFICE BOX 655012 • DALLAS, TEXAS 75265

SN5414, SN54LS14, SN7414, SN74LS14 HEX SCHMITT-TRIGGER INVERTERS

PARAMETER MEASUREMENT INFORMATION



LOAD CIRCUIT



VOLTAGE WAVEFORMS

- NOTES: A. All diodes are 1N3064 or equivalent.
B. C_L includes probe and jig capacitance.
C. Generator characteristics and reference voltage are:

	Generator Characteristics				Reference Voltages		
	Z_{out}	PRR	t_r	t_f	$V_{I ref(H)}$	$V_{I ref(L)}$	$V_{O ref}$
SN54'/SN74'	50 Ω	1 MHz	10 ns	10 ns	1.7 V	0.9 V	1.5 V
SN54LS'/SN74LS'	50 Ω	1 MHz	15 ns	6 ns	1.6 V	0.8 V	1.3 V

TYPICAL CHARACTERISTICS OF '14 CIRCUITS

POSITIVE-GOING THRESHOLD VOLTAGE

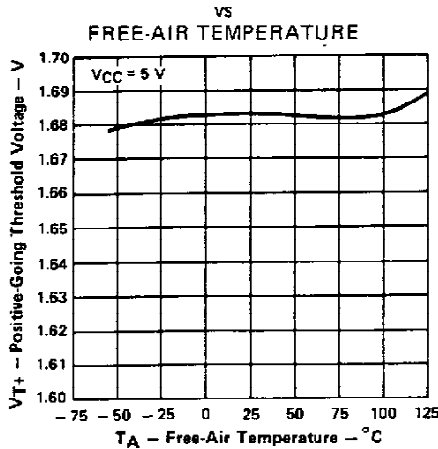


FIGURE 1

NEGATIVE-GOING THRESHOLD VOLTAGE

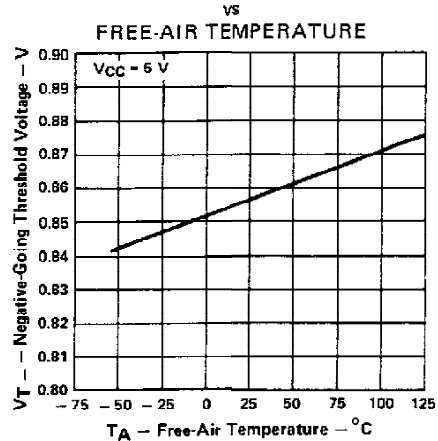


FIGURE 2

HYSTERESIS

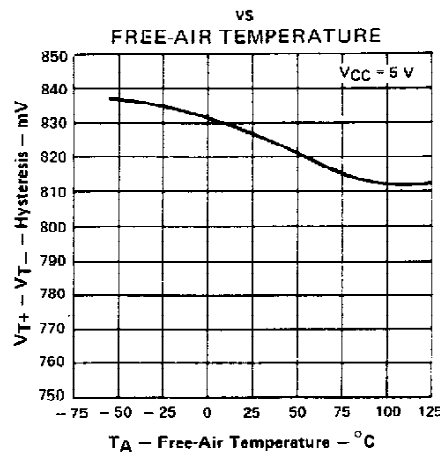


FIGURE 3

Data for temperatures below 0°C and 70°C and supply voltages below 4.75V and above 5.25 V are applicable for SN5414 only.

TEXAS
INSTRUMENTS

POST OFFICE BOX 655012 • DALLAS, TEXAS 75265

SN5414, SN7414

HEX SCHMITT-TRIGGER INVERTERS

TYPICAL CHARACTERISTICS OF '14 CIRCUITS

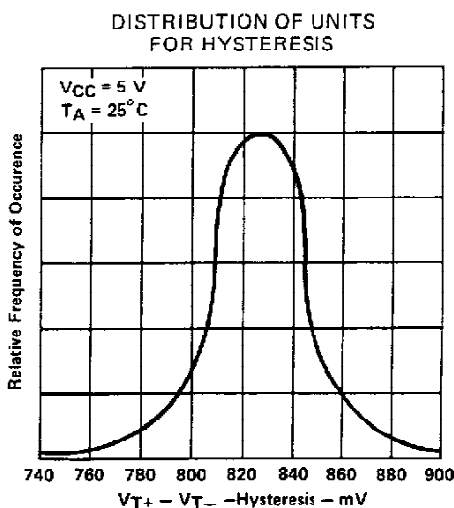


FIGURE 4

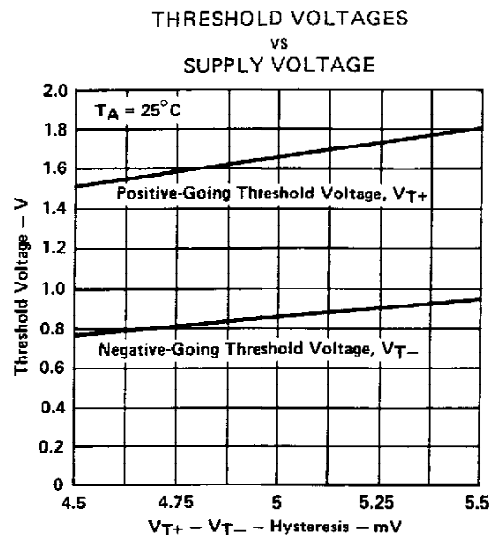


FIGURE 5

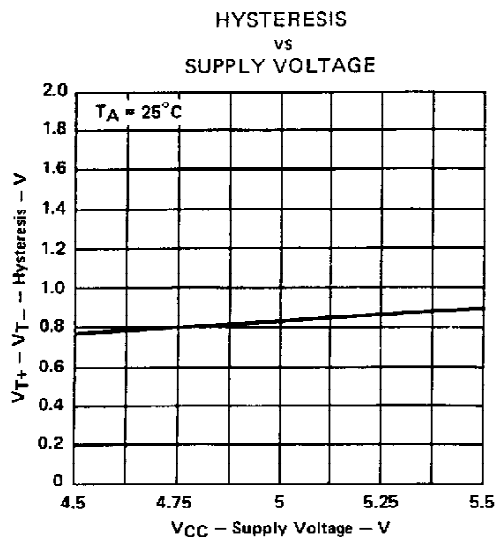


FIGURE 6

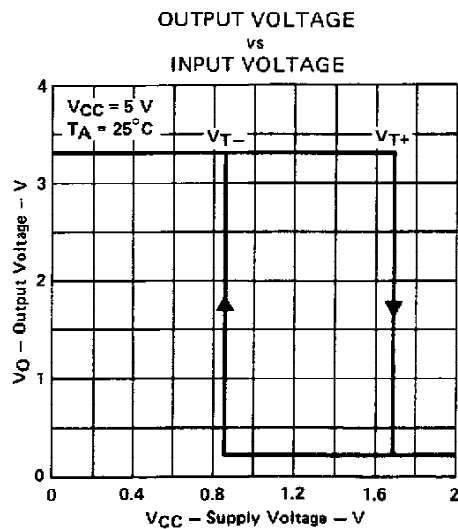


FIGURE 7

Data for temperatures below 0°C and 70°C and supply voltages below 4.75 V and above 5.25 V are applicable for SN5414 only.

TEXAS
INSTRUMENTS

POST OFFICE BOX 655012 • DALLAS, TEXAS 75265

SN54LS14, SN74LS14 HEX SCHMITT-TRIGGER INVERTERS

TYPICAL CHARACTERISTICS OF 'LS14 CIRCUITS

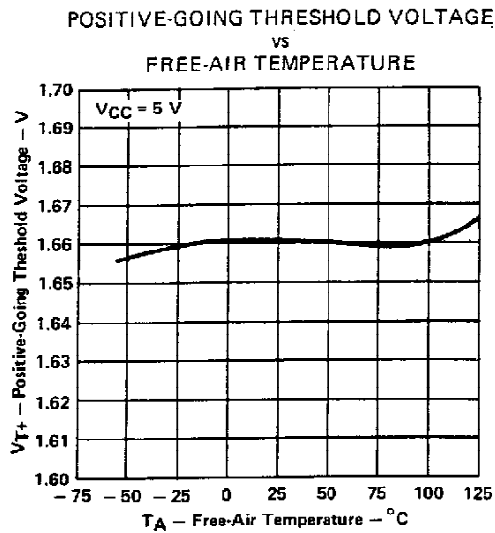


FIGURE 8

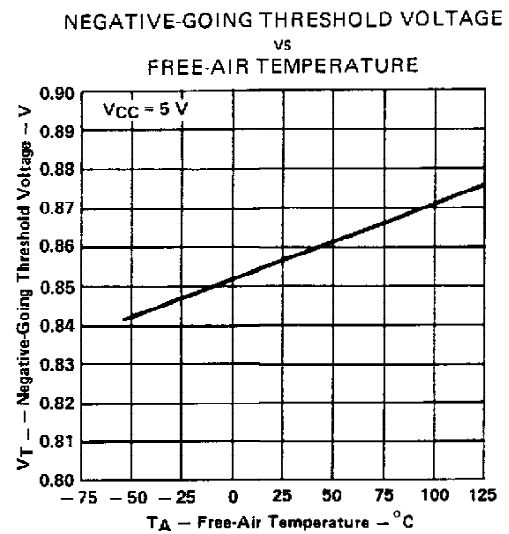


FIGURE 9

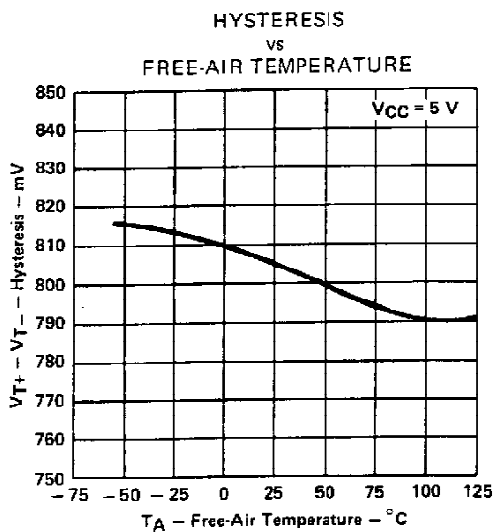


FIGURE 10

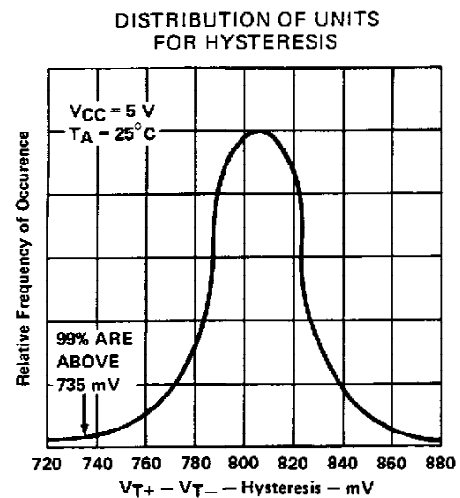


FIGURE 11

Data for temperatures below 0°C and above 70°C and supply voltages below 4.75 V and above 5.25 V are applicable for SN54LS14 only.

SN54LS14, SN74LS14 **HEX SCHMITT-TRIGGER INVERTERS**

TYPICAL CHARACTERISTICS OF 'LS14 CIRCUITS

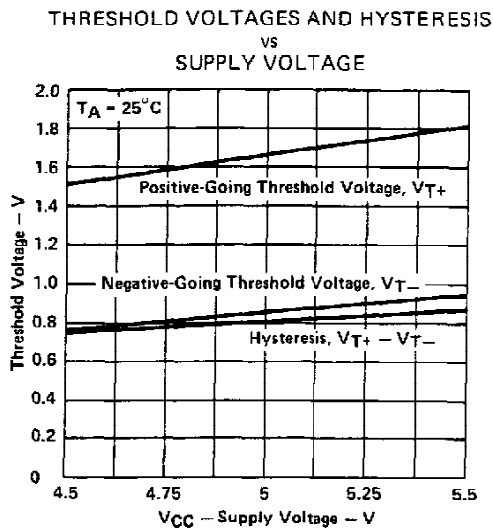


FIGURE 12

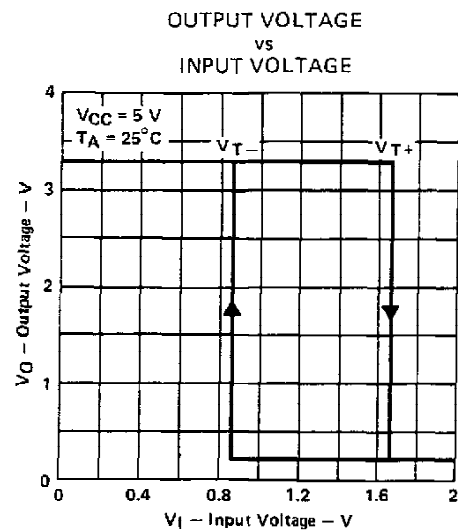
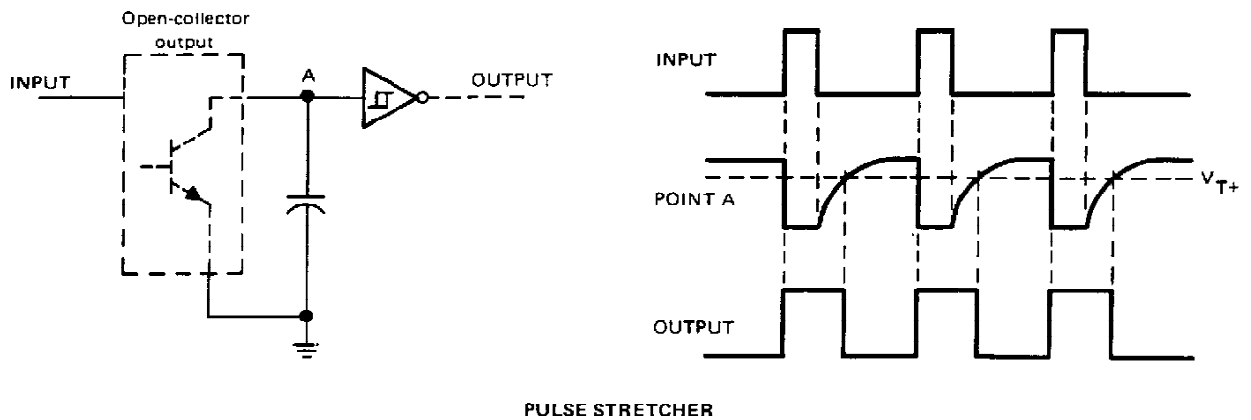
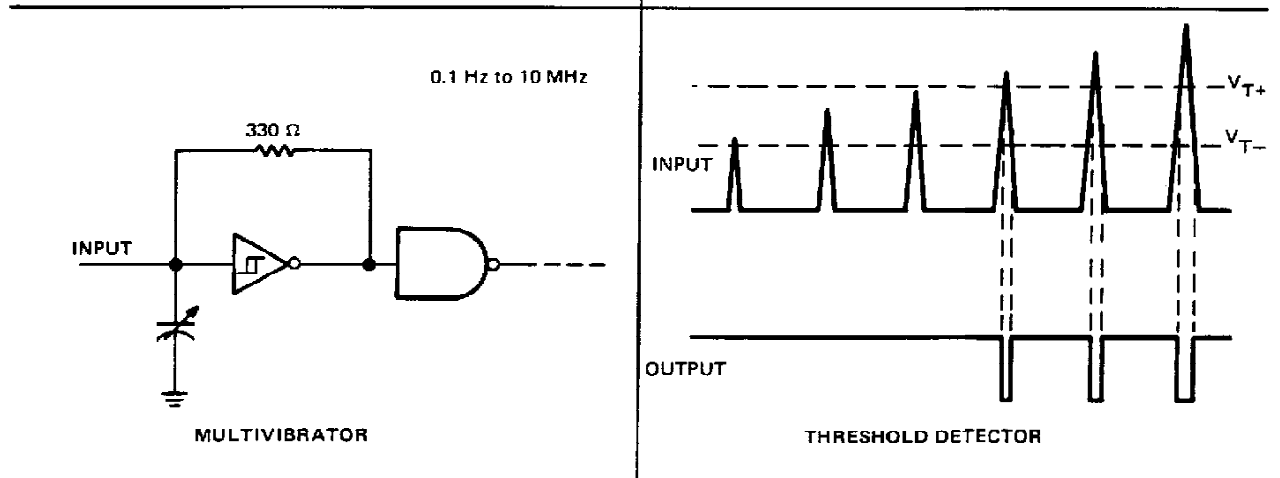
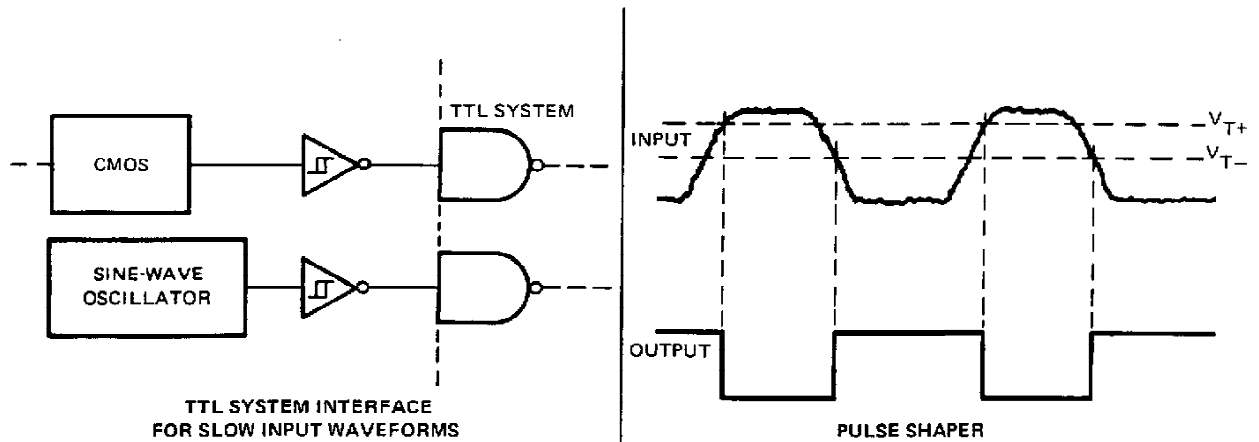


FIGURE 13

Data for temperatures below 0°C and above 70°C and supply voltages below 4.75 V and above 5.25 V are applicable for SN54LS14 only.

SN5414, SN54LS14,
SN7414, SN74LS14
HEX SCHMITT-TRIGGER INVERTERS

TYPICAL APPLICATION DATA



IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

IMPORTANT NOTICE

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgement, including those pertaining to warranty, patent infringement, and limitation of liability.

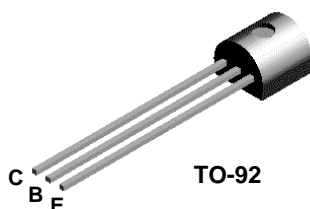
TI warrants performance of its semiconductor products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

CERTAIN APPLICATIONS USING SEMICONDUCTOR PRODUCTS MAY INVOLVE POTENTIAL RISKS OF DEATH, PERSONAL INJURY, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE ("CRITICAL APPLICATIONS"). TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS. INCLUSION OF TI PRODUCTS IN SUCH APPLICATIONS IS UNDERSTOOD TO BE FULLY AT THE CUSTOMER'S RISK.

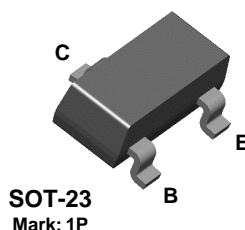
In order to minimize risks associated with the customer's applications, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance or customer product design. TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, warranty or endorsement thereof.

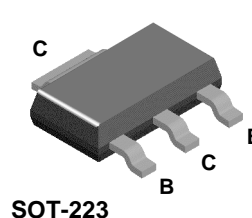
PN2222A



MMBT2222A



PZT2222A



NPN General Purpose Amplifier

This device is for use as a medium power amplifier and switch requiring collector currents up to 500 mA. Sourced from Process 19.

Absolute Maximum Ratings*

$T_A = 25^\circ\text{C}$ unless otherwise noted

Symbol	Parameter	Value	Units
V_{CEO}	Collector-Emitter Voltage	40	V
V_{CBO}	Collector-Base Voltage	75	V
V_{EBO}	Emitter-Base Voltage	6.0	V
I_C	Collector Current - Continuous	1.0	A
T_J, T_{stg}	Operating and Storage Junction Temperature Range	-55 to +150	$^\circ\text{C}$

*These ratings are limiting values above which the serviceability of any semiconductor device may be impaired.

NOTES:

- 1) These ratings are based on a maximum junction temperature of 150 degrees C.
- 2) These are steady state limits. The factory should be consulted on applications involving pulsed or low duty cycle operations.

Thermal Characteristics

$T_A = 25^\circ\text{C}$ unless otherwise noted

Symbol	Characteristic	Max			Units
		PN2222A	*MMBT2222A	**PZT2222A	
P_D	Total Device Dissipation	625	350	1,000	mW
	Derate above 25°C	5.0	2.8	8.0	mW/ $^\circ\text{C}$
$R_{\theta JC}$	Thermal Resistance, Junction to Case	83.3			$^\circ\text{C/W}$
$R_{\theta JA}$	Thermal Resistance, Junction to Ambient	200	357	125	$^\circ\text{C/W}$

* Device mounted on FR-4 PCB 1.6" X 1.6" X 0.06."

** Device mounted on FR-4 PCB 36 mm X 18 mm X 1.5 mm; mounting pad for the collector lead min. 6 cm².

NPN General Purpose Amplifier

(continued)

Electrical Characteristics

T_A = 25°C unless otherwise noted

Symbol	Parameter	Test Conditions	Min	Max	Units
OFF CHARACTERISTICS					
V _{(BR)CEO}	Collector-Emitter Breakdown Voltage*	I _C = 10 mA, I _B = 0	40		V
V _{(BR)CBO}	Collector-Base Breakdown Voltage	I _C = 10 μA, I _E = 0	75		V
V _{(BR)EBO}	Emitter-Base Breakdown Voltage	I _E = 10 μA, I _C = 0	6.0		V
I _{CEX}	Collector Cutoff Current	V _{CE} = 60 V, V _{EB(OFF)} = 3.0 V		10	nA
I _{CBO}	Collector Cutoff Current	V _{CB} = 60 V, I _E = 0 V _{CB} = 60 V, I _E = 0, T _A = 150°C		0.01 10	μA μA
I _{EBO}	Emitter Cutoff Current	V _{EB} = 3.0 V, I _C = 0		10	nA
I _{BL}	Base Cutoff Current	V _{CE} = 60 V, V _{EB(OFF)} = 3.0 V		20	nA

ON CHARACTERISTICS

h _{FE}	DC Current Gain	I _C = 0.1 mA, V _{CE} = 10 V I _C = 1.0 mA, V _{CE} = 10 V I _C = 10 mA, V _{CE} = 10 V I _C = 10 mA, V _{CE} = 10 V, T _A = -55°C I _C = 150 mA, V _{CE} = 10 V* I _C = 150 mA, V _{CE} = 1.0 V* I _C = 500 mA, V _{CE} = 10 V*	35 50 75 35 100 50 40	300	
V _{CE(sat)}	Collector-Emitter Saturation Voltage*	I _C = 150 mA, I _B = 15 mA I _C = 500 mA, I _B = 50 mA		0.3 1.0	V V
V _{BE(sat)}	Base-Emitter Saturation Voltage*	I _C = 150 mA, I _B = 15 mA I _C = 500 mA, I _B = 50 mA	0.6	1.2 2.0	V V

SMALL SIGNAL CHARACTERISTICS

f _T	Current Gain - Bandwidth Product	I _C = 20 mA, V _{CE} = 20 V, f = 100 MHz	300		MHz
C _{obo}	Output Capacitance	V _{CB} = 10 V, I _E = 0, f = 100 kHz		8.0	pF
C _{ibo}	Input Capacitance	V _{EB} = 0.5 V, I _C = 0, f = 100 kHz		25	pF
rb'C _C	Collector Base Time Constant	I _C = 20 mA, V _{CB} = 20 V, f = 31.8 MHz		150	pS
NF	Noise Figure	I _C = 100 μA, V _{CE} = 10 V, R _S = 1.0 kΩ, f = 1.0 kHz		4.0	dB
Re(h _{ie})	Real Part of Common-Emitter High Frequency Input Impedance	I _C = 20 mA, V _{CE} = 20 V, f = 300 MHz		60	Ω

SWITCHING CHARACTERISTICS

t _d	Delay Time	V _{CC} = 30 V, V _{BE(OFF)} = 0.5 V,		10	ns
t _r	Rise Time	I _C = 150 mA, I _{B1} = 15 mA		25	ns
t _s	Storage Time	V _{CC} = 30 V, I _C = 150 mA,		225	ns
t _f	Fall Time	I _{B1} = I _{B2} = 15 mA		60	ns

*Pulse Test: Pulse Width ≤ 300 μs, Duty Cycle ≤ 2.0%

Spice Model

NPN (Is=14.34f Xti=3 Eg=1.11 Vaf=74.03 Bf=255.9 Ne=1.307 Ise=14.34f Ikf=.2847 Xtb=1.5 Br=6.092 Nc=2 Isc=0 lkr=0 Rc=1 Cjc=7.306p Mjc=.3416 Vjc=.75 Fc=.5 Cje=22.01p Mje=.377 Vje=.75 Tr=46.91n Tf=411.1p Itf=.6 Vtf=1.7 Xtf=3 Rb=10)

PN2222A / MMBT2222A / PZT2222A

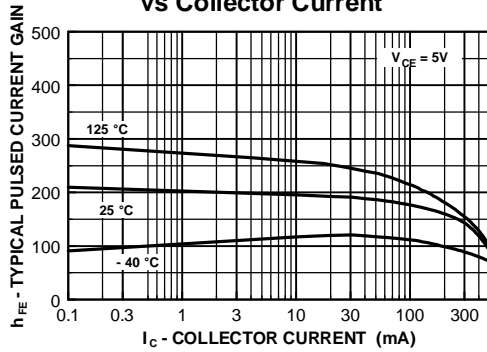
NPN General Purpose Amplifier

(continued)

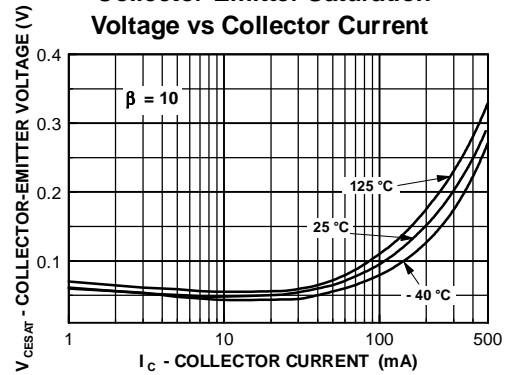
PN2222A / MMBT2222A / PZT2222A

Typical Characteristics

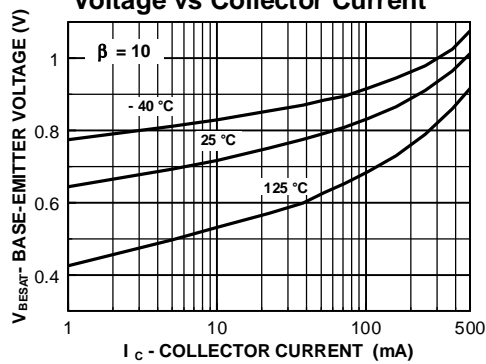
Typical Pulsed Current Gain
vs Collector Current



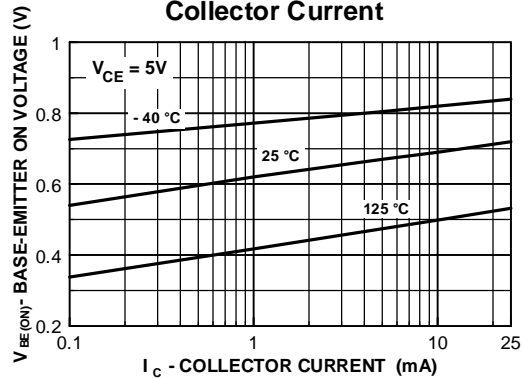
Collector-Emitter Saturation
Voltage vs Collector Current



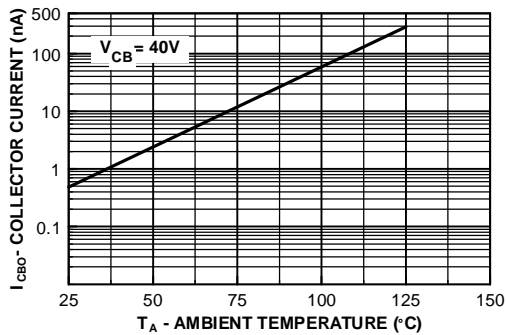
Base-Emitter Saturation
Voltage vs Collector Current



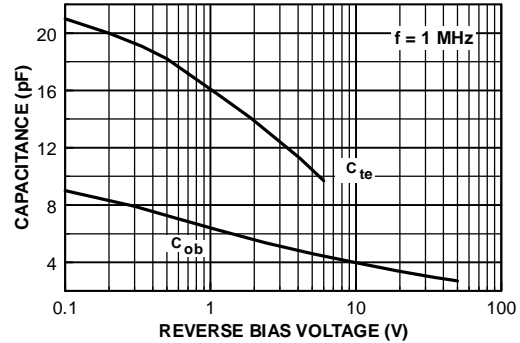
Base-Emitter ON Voltage vs
Collector Current



Collector-Cutoff Current
vs Ambient Temperature

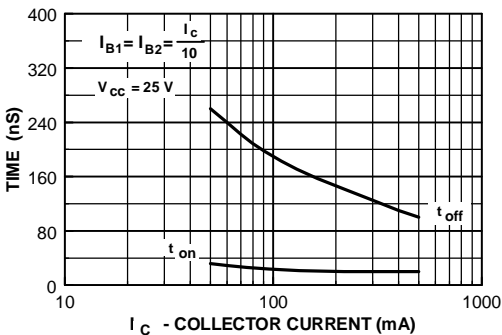


Emitter Transition and Output
Capacitance vs Reverse Bias Voltage

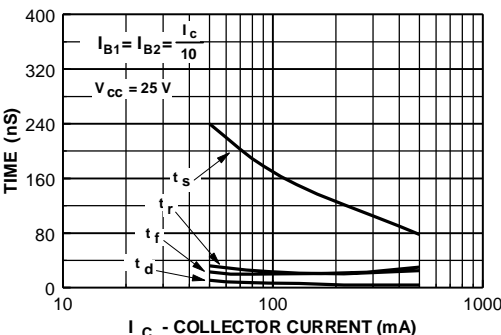


Typical Characteristics (continued)

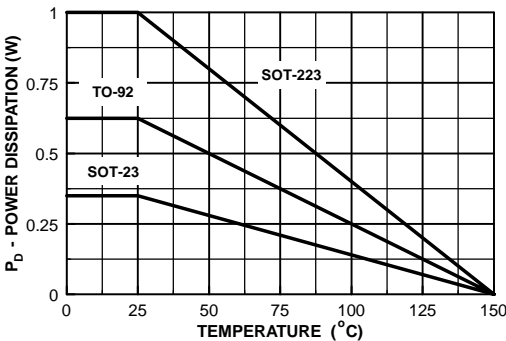
Turn On and Turn Off Times
vs Collector Current



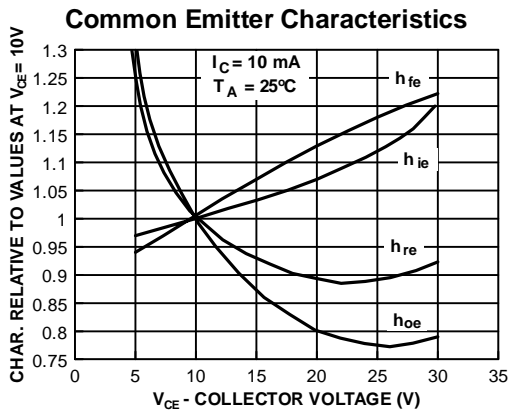
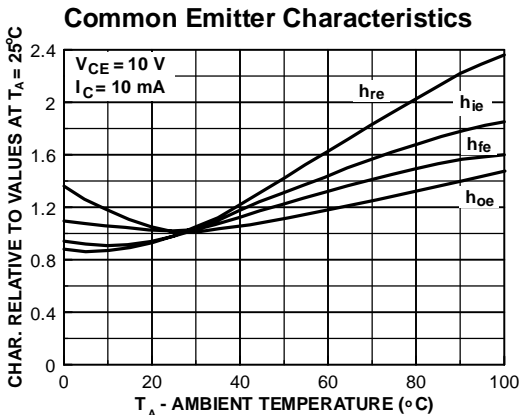
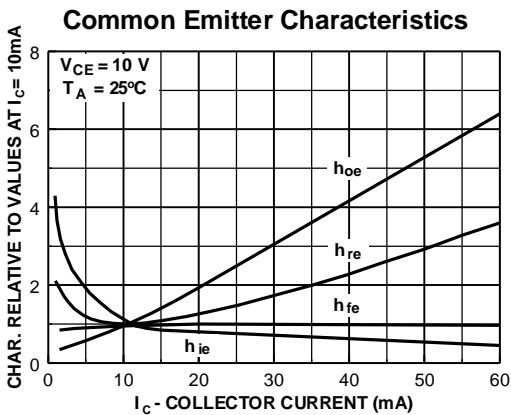
Switching Times
vs Collector Current



Power Dissipation vs
Ambient Temperature



Typical Common Emitter Characteristics (f = 1.0kHz)



Test Circuits

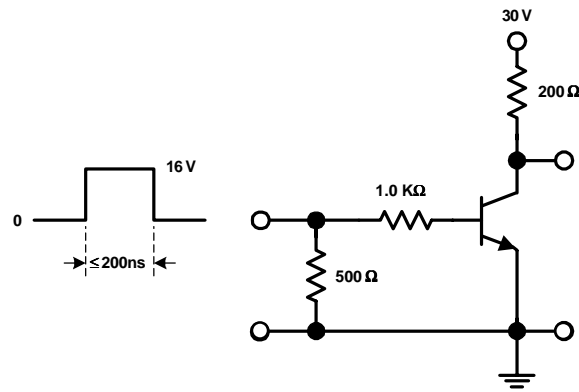


FIGURE 1: Saturated Turn-On Switching Time

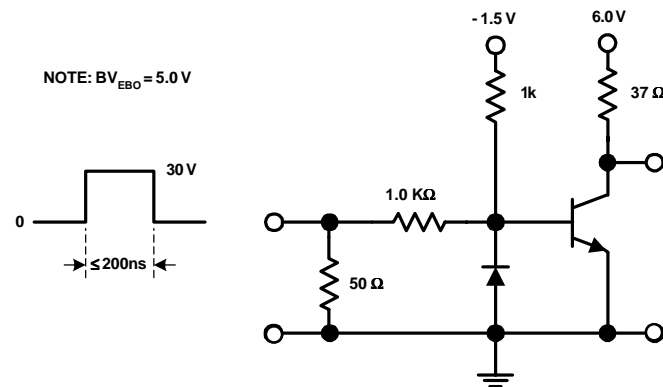


FIGURE 2: Saturated Turn-Off Switching Time

TO-92 Tape and Reel Data



TO-92 Packaging Configuration: Figure 1.0

FSCINT Label sample



F63TNR Label sample



TAPE and REEL OPTION See Fig 2.0 for various Reeling Styles



TO-92 TNR/AMMO PACKING INFORMATION

Packing	Style	Quantity	EOL code
Reel	A	2,000	D26Z
	E	2,000	D27Z
Ammo	M	2,000	D74Z
	P	2,000	D75Z

Unit weight = 0.22 gm
 Reel weight with components = 1.04 kg
 Ammo weight with components = 1.02 kg
 Max quantity per intermediate box = 10,000 units

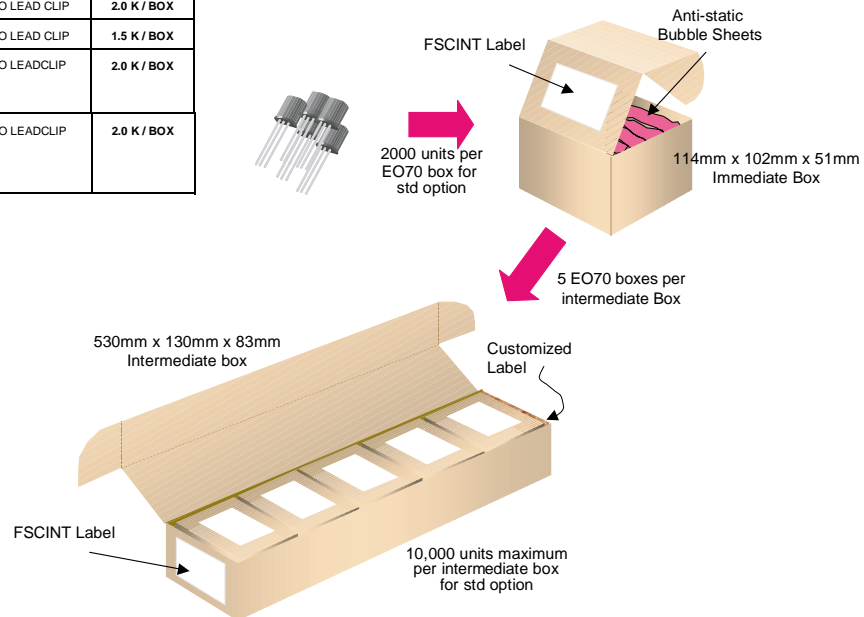
AMMO PACK OPTION See Fig 3.0 for 2 Ammo Pack Options



(TO-92) BULK PACKING INFORMATION

EOL CODE	DESCRIPTION	LEADCLIP DIMENSION	QUANTITY
J18Z	TO-18 OPTION STD	NO LEAD CLIP	2.0 K / BOX
J05Z	TO-5 OPTION STD	NO LEAD CLIP	1.5 K / BOX
NO EOL CODE	TO-92 STANDARD STRAIGHT FOR: PKG 92, 94 (NON PROELECTRON SERIES), 96	NO LEADCLIP	2.0 K / BOX
L34Z	TO-92 STANDARD STRAIGHT FOR: PKG 94 (PROELECTRON SERIES BCXXX, BFXXX, BSRXXX), 97, 98	NO LEADCLIP	2.0 K / BOX

BULK OPTION See Bulk Packing Information table

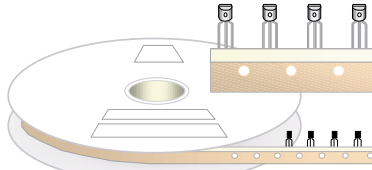


TO-92 Tape and Reel Data, continued

TO-92 Reeling Style

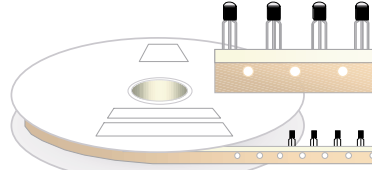
Configuration: Figure 2.0

Machine Option "A" (H)



Style "A", D26Z, D70Z (s/h)

Machine Option "E" (J)

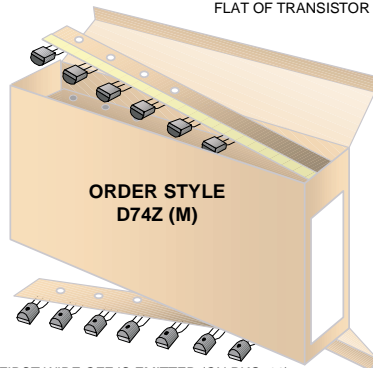


Style "E", D27Z, D71Z (s/h)

TO-92 Radial Ammo Packaging

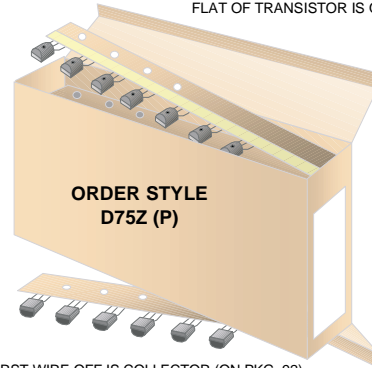
Configuration: Figure 3.0

FIRST WIRE OFF IS COLLECTOR
ADHESIVE TAPE IS ON THE TOP SIDE
FLAT OF TRANSISTOR IS ON TOP



FIRST WIRE OFF IS EMITTER (ON PKG. 92)
ADHESIVE TAPE IS ON BOTTOM SIDE
FLAT OF TRANSISTOR IS ON BOTTOM

FIRST WIRE OFF IS EMITTER
ADHESIVE TAPE IS ON THE TOP SIDE
FLAT OF TRANSISTOR IS ON BOTTOM



FIRST WIRE OFF IS COLLECTOR (ON PKG. 92)
ADHESIVE TAPE IS ON BOTTOM SIDE
FLAT OF TRANSISTOR IS ON TOP

TO-92 Tape and Reel Data, continued

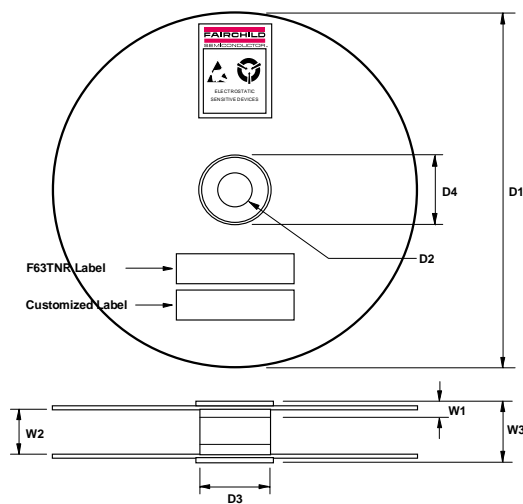
TO-92 Tape and Reel Taping Dimension Configuration: Figure 4.0



ITEM DESCRIPTION	SYMBOL	DIMENSION
Base of Package to Lead Bend	b	0.098 (max)
Component Height	Ha	0.928 (+/- 0.025)
Lead Clinch Height	HO	0.630 (+/- 0.020)
Component Base Height	H1	0.748 (+/- 0.020)
Component Alignment (side/side)	Pd	0.040 (max)
Component Alignment (front/back)	Hd	0.031 (max)
Component Pitch	P	0.500 (+/- 0.020)
Feed Hole Pitch	PO	0.500 (+/- 0.008)
Hole Center to First Lead	P1	0.150 (+0.009, -0.010)
Hole Center to Component Center	P2	0.247 (+/- 0.007)
Lead Spread	F1/F2	0.104 (+/- 0.010)
Lead Thickness	d	0.018 (+0.002, -0.003)
Cut Lead Length	L	0.429 (max)
Taped Lead Length	L1	0.209 (+0.051, -0.052)
Taped Lead Thickness	t	0.032 (+/- 0.006)
Carrier Tape Thickness	t1	0.021 (+/- 0.006)
Carrier Tape Width	W	0.708 (+0.020, -0.019)
Hold - down Tape Width	WO	0.236 (+/- 0.012)
Hold - down Tape position	W1	0.035 (max)
Feed Hole Position	W2	0.360 (+/- 0.025)
Sprocket Hole Diameter	DO	0.157 (+0.008, -0.007)
Lead Spring Out	S	0.004 (max)

Note : All dimensions are in inches.

TO-92 Reel Configuration: Figure 5.0



ITEM DESCRIPTION	SYMBOL	MINIMUM	MAXIMUM
Reel Diameter	D1	13.975	14.025
Arbor Hole Diameter (Standard)	D2	1.160	1.200
Arbor Hole Diameter (Small Hole)	D2	0.650	0.700
Core Diameter	D3	3.100	3.300
Hub Recess Inner Diameter	D4	2.700	3.100
Hub Recess Depth	W1	0.370	0.570
Flange to Flange Inner Width	W2	1.630	1.690
Hub to Hub Center Width	W3		2.090

Note: All dimensions are inches

TO-92 Package Dimensions



TO-92 (FS PKG Code 92, 94, 96)



Scale 1:1 on letter size paper

Dimensions shown below are in:
inches [millimeters]

Part Weight per unit (gram): 0.1977

TO-92 (92,94,96)

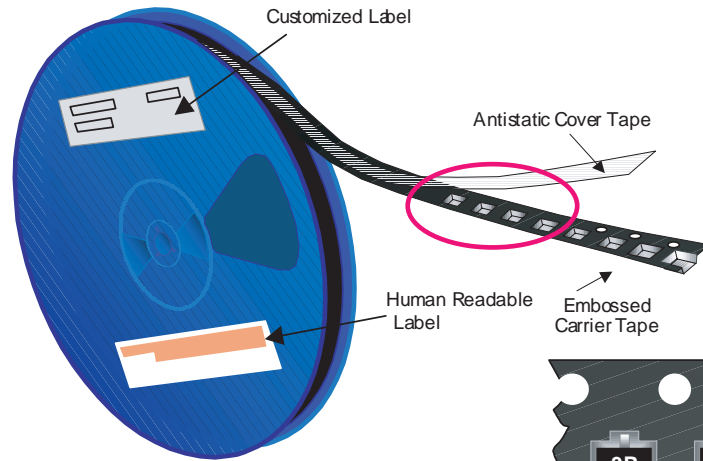
PIN	92		94		96	
	B	F	B	F	B	F
1	E	D	E	D	B	S
2	B	S	C	G	E	D
3	C	G	B	S	C	G



SOT-23 Tape and Reel Data



SOT-23 Packaging Configuration: Figure 10

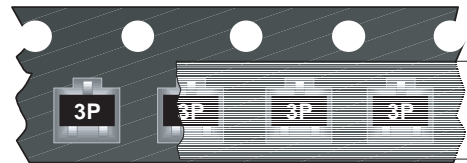


Packaging Description:

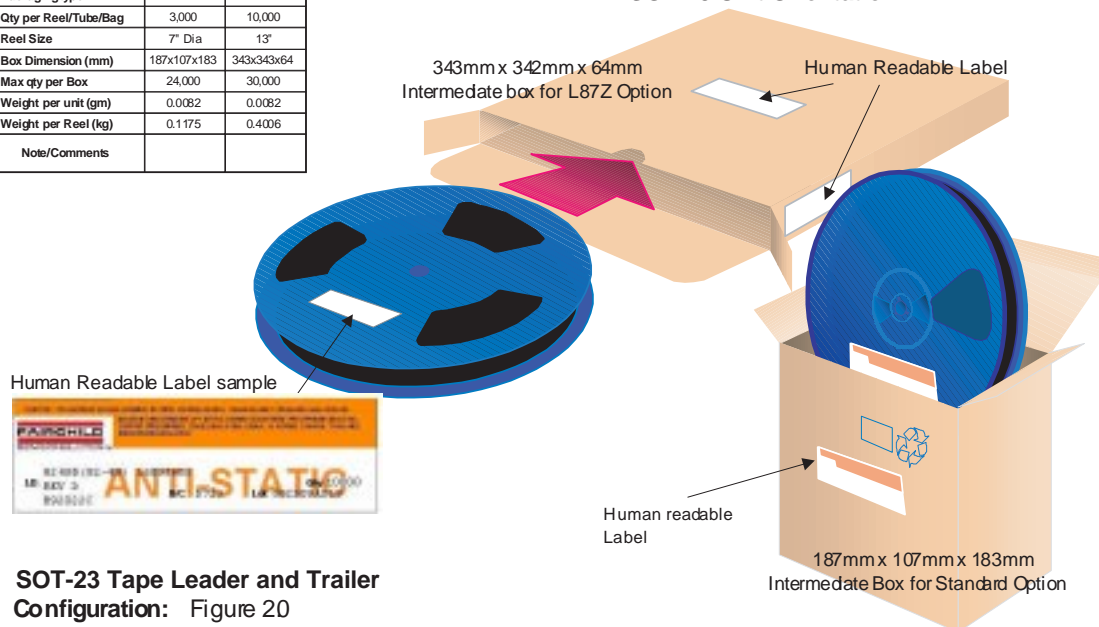
SOT-23 parts are shipped in tape. The carrier tape is made from a dissipative (carbon filled) polycarbonate resin. The cover tape is a multilayer film (Heat Activated Adhesive in nature) primarily composed of polyester film, adhesive layer, sealant, and anti-static sprayed agent. These reeled parts in standard option are shipped with 3,000 units per 7" or 177cm diameter reel. The reels are dark blue in color and is made of polystyrene plastic (anti-static coated). Other option comes in 10,000 units per 13" or 330cm diameter reel. This and some other options are described in the Packaging Information table.

These full reels are individually labeled and placed inside a standard intermediate made of recyclable corrugated brown paper with a Fairchild logo printing. One pizza box contains eight reels maximum. And these intermediate boxes are placed inside a labeled shipping box which comes in different sizes depending on the number of parts shipped.

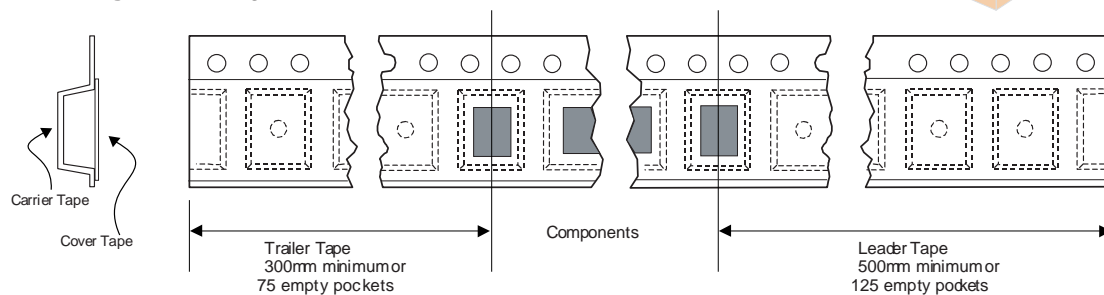
SOT-23 Packaging Information		
Packaging Option	Standard (no flow code)	D87Z
Packaging type	TNR	TNR
Qty per Reel/Tube/Bag	3,000	10,000
Reel Size	7" Dia	13"
Box Dimension (mm)	187x107x183	343x343x64
Max qty per Box	24,000	30,000
Weight per unit (gm)	0.0082	0.0082
Weight per Reel (kg)	0.1175	0.4006
Note/Comments		



SOT-23 Unit Orientation



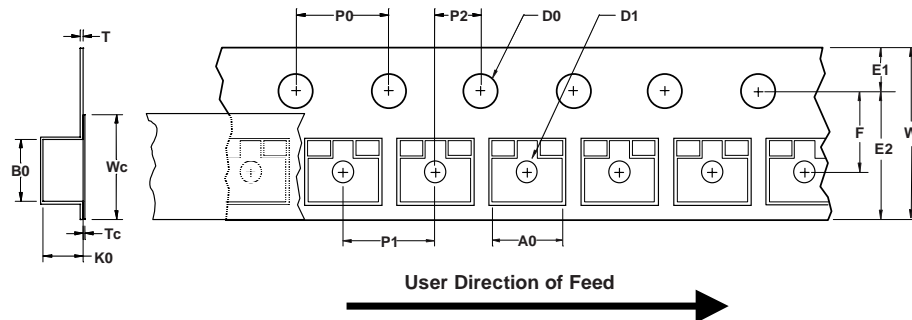
SOT-23 Tape Leader and Trailer Configuration: Figure 20



SOT-23 Tape and Reel Data, continued

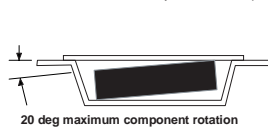
SOT-23 Embossed Carrier Tape

Configuration: Figure 3.0

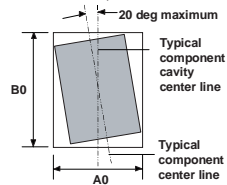


Dimensions are in millimeter														
Pkg type	A0	B0	W	D0	D1	E1	E2	F	P1	P0	K0	T	Wc	Tc
SOT-23 (8mm)	3.15 +/-0.10	2.77 +/-0.10	8.0 +/-0.3	1.55 +/-0.05	1.125 +/-0.125	1.75 +/-0.10	6.25 min	3.50 +/-0.05	4.0 +/-0.1	4.0 +/-0.1	1.30 +/-0.10	0.228 +/-0.013	5.2 +/-0.3	0.06 +/-0.02

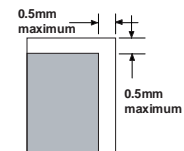
Notes: A0, B0, and K0 dimensions are determined with respect to the EIA/Jedec RS-481 rotational and lateral movement requirements (see sketches A, B, and C).



Sketch A (Side or Front Sectional View)
Component Rotation

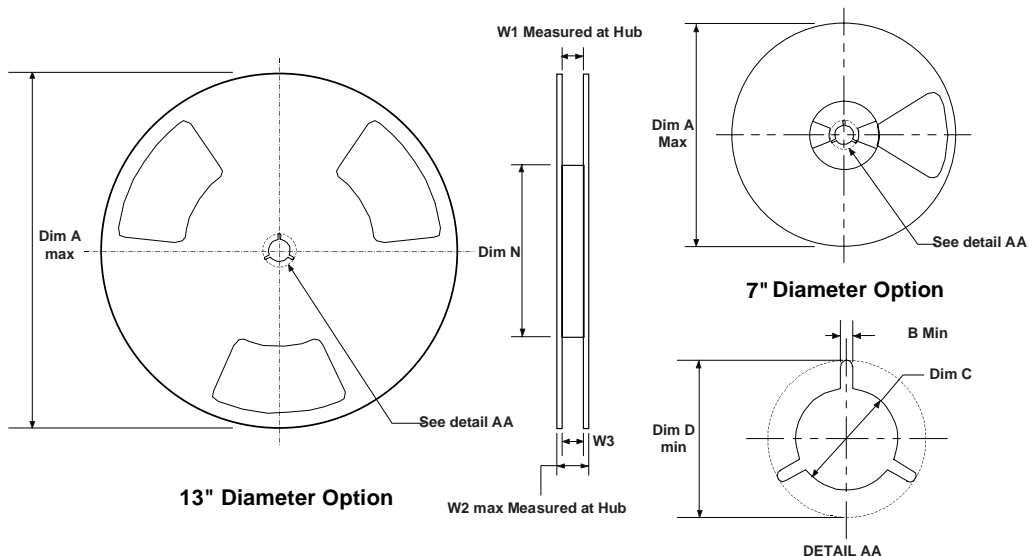


Sketch B (Top View)
Component Rotation



Sketch C (Top View)
Component lateral movement

SOT-23 Reel Configuration: Figure 4.0

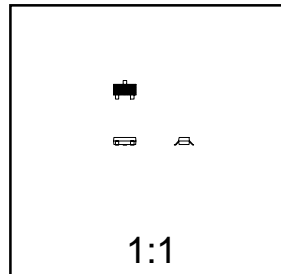
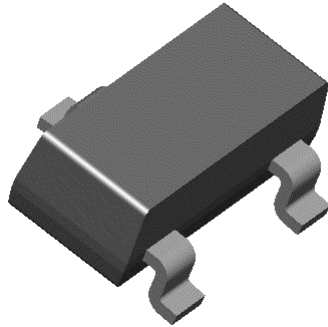


Dimensions are in inches and millimeters									
Tape Size	Reel Option	Dim A	Dim B	Dim C	Dim D	Dim N	Dim W1	Dim W2	Dim W3 (LSL-USL)
8mm	7" Dia	7.00 177.8	0.059 1.5	512 +0.020/-0.008 13 +0.5/-0.2	0.795 20.2	2.165 55	0.331 +0.059/-0.000 8.4 +1.5/0	0.567 14.4	0.311 - 0.429 7.9 - 10.9
8mm	13" Dia	13.00 330	0.059 1.5	512 +0.020/-0.008 13 +0.5/-0.2	0.795 20.2	4.00 100	0.331 +0.059/-0.000 8.4 +1.5/0	0.567 14.4	0.311 - 0.429 7.9 - 10.9

SOT-23 Package Dimensions



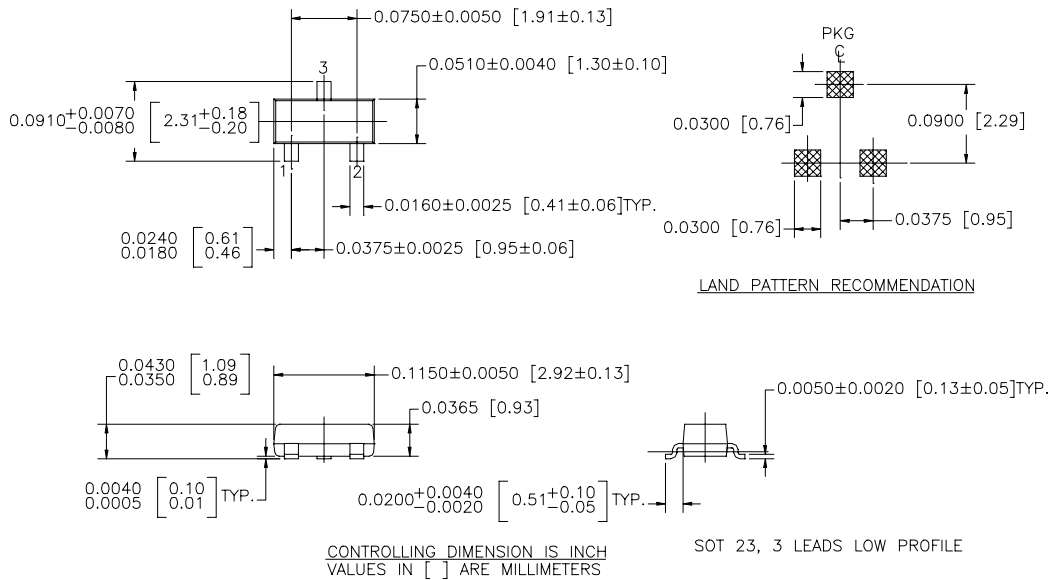
SOT-23 (FS PKG Code 49)



Scale 1:1 on letter size paper

Dimensions shown below are in:
inches [millimeters]

Part Weight per unit (gram): 0.0082



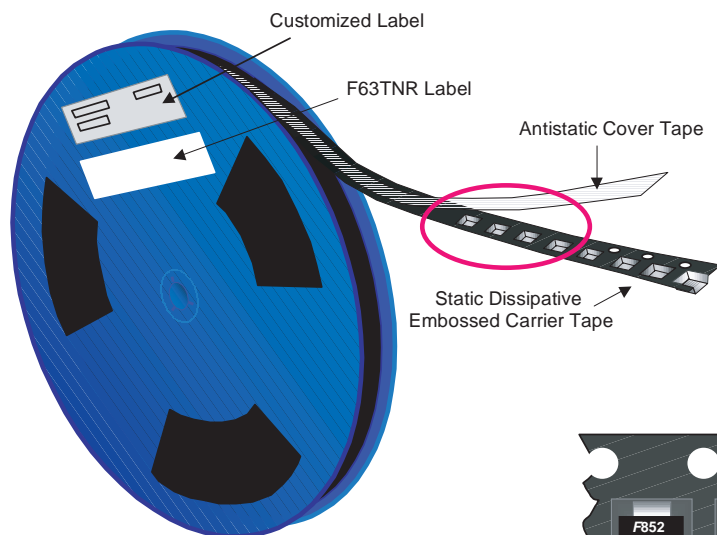
NOTE : UNLESS OTHERWISE SPECIFIED

1. STANDARD LEAD FINISH 150 MICROINCHES / 3.81 MICROMETERS
MINIMUM TIN / LEAD (SOLDER) ON ALLOY 42
2. REFERENCE JEDEC REGISTRATION TO-236, VARIATION AB, ISSUE G, DATED JUL 1993

SOT-223 Tape and Reel Data



SOT-223 Packaging Configuration: Figure 1.0

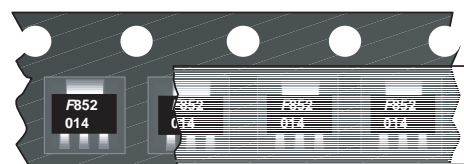


Packaging Description:

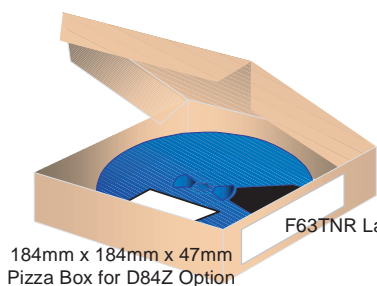
SOT-223 parts are shipped in tape. The carrier tape is made from a dissipative (carbon filled) polycarbonate resin. The cover tape is a multilayer film (Heat Activated Adhesive in nature) primarily composed of polyester film, adhesive layer, sealant, and anti-static sprayed agent. These reeled parts in standard option are shipped with 2,500 units per 13" or 330cm diameter reel. The reels are dark blue in color and is made of polystyrene plastic (anti-static coated). Other option comes in 500 units per 7" or 177cm diameter reel. This and some other options are further described in the Packaging Information table.

These full reels are individually barcode labeled and placed inside a standard intermediate box (illustrated in figure 1.0) made of recyclable corrugated brown paper. One box contains two reels maximum. And these boxes are placed inside a barcode labeled shipping box which comes in different sizes depending on the number of parts shipped.

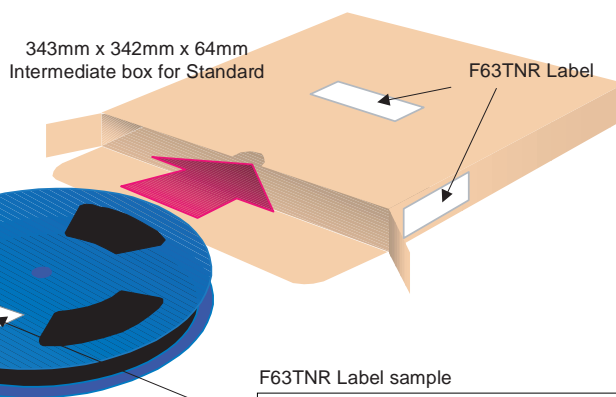
SOT-223 Packaging Information		
Packaging Option	Standard (no flow code)	D84Z
Packaging type	TNR	TNR
Qty per Reel/Tube/Bag	2,500	500
Reel Size	13" Dia	7" Dia
Box Dimension (mm)	343x64x343	184x187x47
Max qty per Box	5,000	1,000
Weight per unit (gm)	0.1246	0.1246
Weight per Reel (kg)	0.7250	0.1532
Note/Comments		



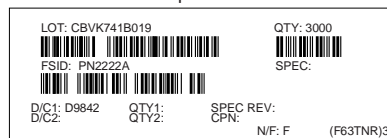
SOT-223 Unit Orientation



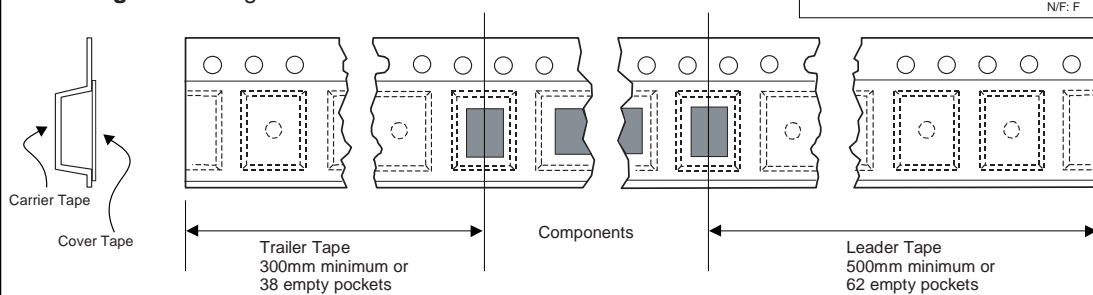
343mm x 342mm x 64mm
Intermediate box for Standard



F63TNR Label sample



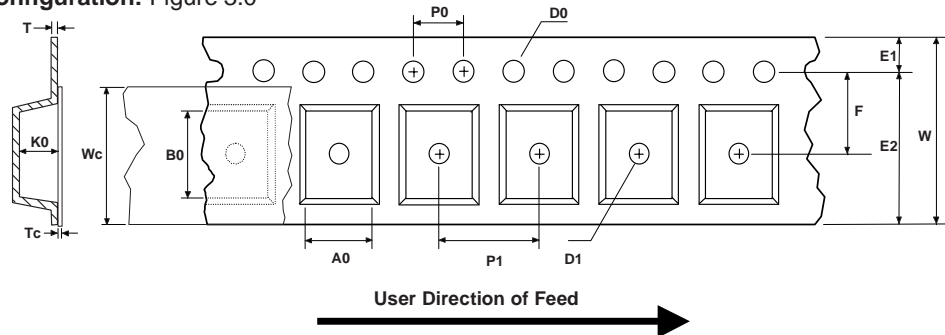
SOT-223 Tape Leader and Trailer Configuration: Figure 2.0



SOT-223 Tape and Reel Data, continued

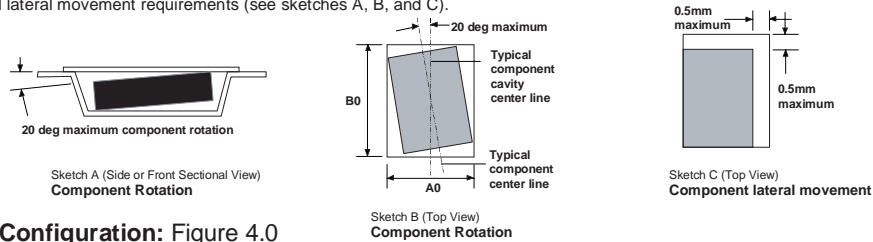
SOT-223 Embossed Carrier Tape

Configuration: Figure 3.0

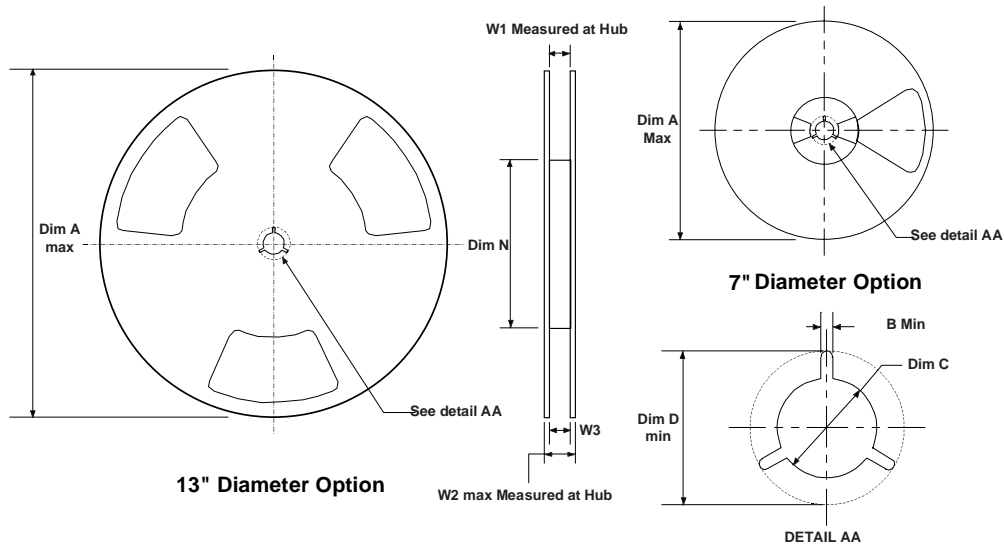


Dimensions are in millimeter														
Pkg type	A0	B0	W	D0	D1	E1	E2	F	P1	P0	K0	T	Wc	Tc
SOT-223 (12mm)	6.83 +/-0.10	7.42 +/-0.10	12.0 +/-0.3	1.55 +/-0.05	1.50 +/-0.10	1.75 +/-0.10	10.25 min	5.50 +/-0.05	8.0 +/-0.1	4.0 +/-0.1	1.88 +/-0.10	0.292 +/- 0.0130	9.5 +/-0.025	0.06 +/-0.02

Notes: A0, B0, and K0 dimensions are determined with respect to the EIA/Jedec RS-481 rotational and lateral movement requirements (see sketches A, B, and C).



SOT-223 Reel Configuration: Figure 4.0

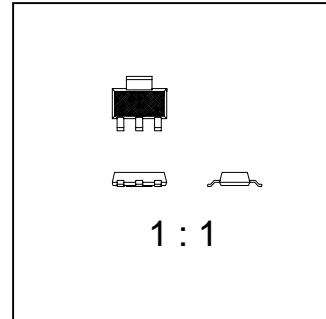
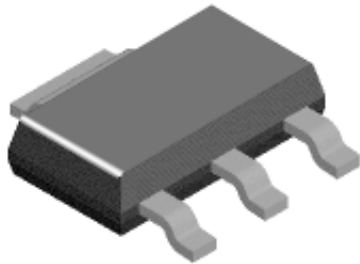


Dimensions are in inches and millimeters									
Tape Size	Reel Option	Dim A	Dim B	Dim C	Dim D	Dim N	Dim W1	Dim W2	Dim W3 (LSL-USL)
12mm	7" Dia	7.00 177.8	0.059 1.5	512 +0.020/-0.008 13 +0.5/-0.2	0.795 20.2	5.906 150	0.488 +0.078/-0.000 12.4 +2/0	0.724 18.4	0.469 - 0.606 11.9 - 15.4
12mm	13" Dia	13.00 330	0.059 1.5	512 +0.020/-0.008 13 +0.5/-0.2	0.795 20.2	7.00 178	0.488 +0.078/-0.000 12.4 +2/0	0.724 18.4	0.469 - 0.606 11.9 - 15.4

SOT-223 Package Dimensions

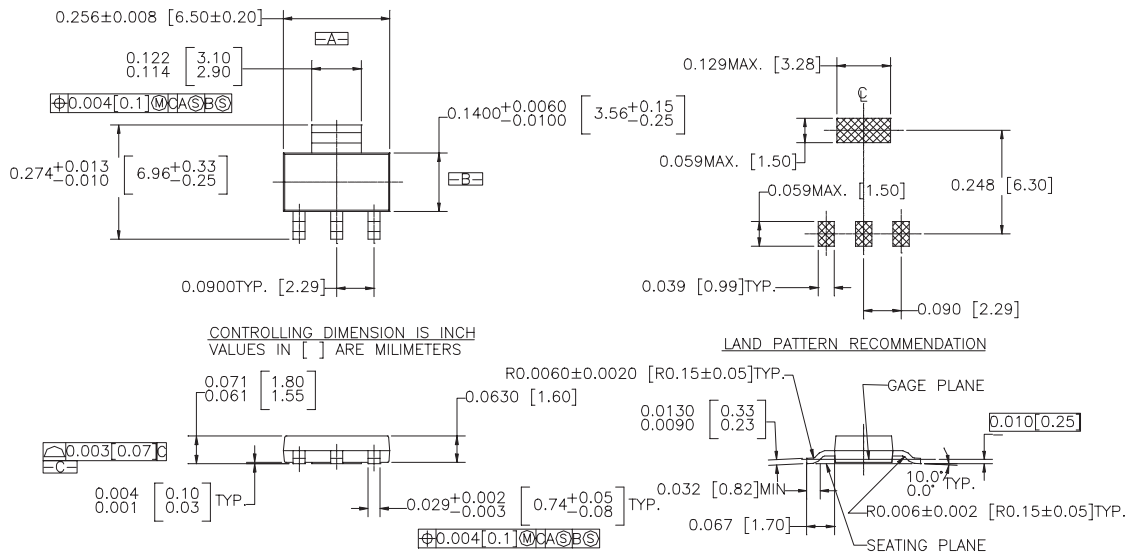


SOT-223 (FS PKG Code 47)



Scale 1:1 on letter size paper

Part Weight per unit (gram): 0.1246



NOTES : UNLESS OTHERWISE SPECIFIED

1. STANDARD LEAD FINISH TO BE 150 MICRONS/ 3.81 MICROMETERS

MINIMUM TIN/LEAD (SOLDER) ON COPPER.

2. REFERENCE JEDEC REGISTRATION TO-261, VARIATION AA, ISSUE A, DATED JAN 1990

TRADEMARKS

The following are registered and unregistered trademarks Fairchild Semiconductor owns or is authorized to use and is not intended to be an exhaustive list of all such trademarks.

ACEx™	FASTr™	PowerTrench®	SyncFET™
Bottomless™	GlobalOptoisolator™	QFET™	TinyLogic™
CoolFET™	GTO™	QS™	UHC™
CROSSVOLT™	HiSeC™	QT Optoelectronics™	VCX™
DOME™	ISOPLANAR™	Quiet Series™	
E ² CMOS™	MICROWIRE™	SILENT SWITCHER®	
EnSigna™	OPTOLOGIC™	SMART START™	
FACT™	OPTOPLANAR™	SuperSOT™-3	
FACT Quiet Series™	PACMAN™	SuperSOT™-6	
FAST®	POP™	SuperSOT™-8	

DISCLAIMER

FAIRCHILD SEMICONDUCTOR RESERVES THE RIGHT TO MAKE CHANGES WITHOUT FURTHER NOTICE TO ANY PRODUCTS HEREIN TO IMPROVE RELIABILITY, FUNCTION OR DESIGN. FAIRCHILD DOES NOT ASSUME ANY LIABILITY ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT OR CIRCUIT DESCRIBED HEREIN; NEITHER DOES IT CONVEY ANY LICENSE UNDER ITS PATENT RIGHTS, NOR THE RIGHTS OF OTHERS.

LIFE SUPPORT POLICY

FAIRCHILD'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF FAIRCHILD SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, or (c) whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

PRODUCT STATUS DEFINITIONS

Definition of Terms

Datasheet Identification	Product Status	Definition
Advance Information	Formative or In Design	This datasheet contains the design specifications for product development. Specifications may change in any manner without notice.
Preliminary	First Production	This datasheet contains preliminary data, and supplementary data will be published at a later date. Fairchild Semiconductor reserves the right to make changes at any time without notice in order to improve design.
No Identification Needed	Full Production	This datasheet contains final specifications. Fairchild Semiconductor reserves the right to make changes at any time without notice in order to improve design.
Obsolete	Not In Production	This datasheet contains specifications on a product that has been discontinued by Fairchild semiconductor. The datasheet is printed for reference information only.

ANEXO B

HOJAS DE CARACTERÍSTICAS DE MODULOS DIGITALES IS1U60

ANEXO C

HOJAS DE CARACTERÍSTICAS DE SFH203FA, SFH485 Y LM324

ANEXO D

PROGRAMA EN ENSAMBLADOR VIGILA.ASM

martes, 05 de febrero de 2002

```
;
;
; PROGRAMA PARA QUE ROBOT DETECTE LA PRESENCIA DE LA PRESA.
;
; Sensores: - Un solo detector digital para detección de la presa.
;           - Cuatro sensores analógicos para detección de posibles
;           obstaculos formados por emisores() y receptores().
;           - Dos bumpers frontales.
;
; Descripción: El robot busca a la presa girando sobre si mismo hacia
; la derecha hasta completar una vuelta y si no lo detecta se para. La
programación se ha
; hecho a base de temporización por lo que si el robot gira en busca de la
; presa y detecta un obstaculo gira 0.5 seg. a la izquierda y otros 0.5 seg.
; hacia delante.
;
PORTA EQU $00      ; Puerto desde el que activan los motores
PORTD EQU $08
PORTE EQU $0A
DDRD EQU $09
TMSK1 EQU $22
TCTL2 EQU $21
TFLG1 EQU $23      ; Flags de los temporizadores
TCNT EQU $0E       ; Lleva la cuenta actual del temporizador
TOC1 EQU $16       ; Registro que almacena el tiempo final
ACUM EQU $00E8

ORG $0000

paso_s1 RMB 2
paso_s2 RMB 2

ORG $B600

inic LDAA #$7E
     STAA ACUM
     LDX #rut_enc
     STX ACUM+1
     BSR borr_s1
     LDX #$1000
     BCLR DDRD,X $00

inict LDAB #$80      ; Inicializa el flag de interrupción de OC1
     STAB TFLG1,X

ini_enc BSET TCTL2,X $30
        BSET TMSK1,X $04
        CLI

guard BRCLR PORTD,X $08 go_der
      BRCLR PORTD,X $10 go_izq
      BRCLR PORTD,X $04 go_rec
      BSR borr_s1
      BSR gira_de
guard1 BRCLR PORTE,X $04 go_detod
      BRCLR PORTD,X $08 go_der
      BRCLR PORTD,X $10 go_izq
      BRCLR PORTD,X $04 go_rec
      LDD paso_s1
      CPD #64
      BNE guard1
      BSR borr_s1
      BSR sig_rec
guard2 BRCLR PORTE,X $01 testbd
      BRCLR PORTE,X $02 testbi
      BRCLR PORTD,X $08 go_der
      BRCLR PORTD,X $10 go_izq
      BRCLR PORTD,X $04 go_rec
```

martes, 05 de febrero de 2002

```
LDD paso_s1
CPD #48
BNE guard2
CLRA
STAA PORTA,X
LDY #200
guard3 BRCLR PORTD,X $08 go_der
BRCLR PORTD,X $10 go_izq
BRCLR PORTD,X $04 go_rec
BSR temp
DEY
BNE guard3
BRA guard

borr_s1 CLRA
CLRB
STD paso_s1
RTS

borr_s2 CLRA
CLRB
STD paso_s2
RTS

go_izq JMP izquier
go_der JMP derecha
go_rec JMP avanza
go_detod JMP detobsd
go_detoj JMP detobsi

gira_de LDAA #$38
STAA PORTA,X
RTS

sig_rec LDAA #$18
STAA PORTA,X
RTS

gira_iz LDAA #$58
STAA PORTA,X
RTS

testbd JMP bumpd
testbi JMP bumpi

temp LDD TCNT,X ; Temporizaci3n de 10 ms.
ADDD #20000
STD TOC1,X ; Carga TOC1 con cuenta final (10 ms)
BRCLR TFLG1,X $80 loop ; Esperamos a que pasen 10 ms
BCLR TFLG1,X $7F ; Desactivamos OC1F para siguiente cuenta
RTS

esp_det LDY #20
wait BSR temp
DEY
BNE wait
RTS

rut_enc LDD paso_s1
ADDD #1
STD paso_s1
LDD paso_s2
ADDD #1
```

martes, 05 de febrero de 2002

```
STD paso_s2
LDAA #$04
STAA TFLG1,X
RTI

izquier JSR gira_iz
BRCLR PORTE,X $08 detobsi
JSR esp_det
JMP guard

derecha JSR gira_de
BRCLR PORTE,X $04 detobsd
JSR esp_det
JMP guard

avanza JSR sig_rec
BRCLR PORTD,X $20 go_fin
BRCLR PORTE,X $01 testbd
BRCLR PORTE,X $02 testbi
JSR esp_det
JMP guard

detobsd JSR gira_iz
BRA rut_gi

detobsi JSR gira_de
rut_gi JSR borr_s1
giro LDD paso_s1
CPD #4
BNE giro
recto JSR sig_rec
JSR borr_s1
recto1 BRCLR PORTE,X $01 testbd
BRCLR PORTE,X $02 testbi
LDD paso_s1
CPD #6
BNE recto1
JMP guard

go_fin CLRA
STAA PORTA,X
LDY #500
esp_fin JSR temp
DEY
BNE esp_fin
JMP fin

dere JSR gira_de
de1 JSR borr_s1
LDD paso_s1
CPD #20
BEQ fin_fi
BRCLR PORTE,X $02 bumpi
BRSET PORTE,X $08 de1
BRA fin_de

izqui JSR gira_iz
JSR borr_s1
iz1 LDD paso_s1
CPD #20
BEQ fin_fd
BRCLR PORTE,X $01 bumpd
BRSET PORTE,X $04 iz1
BRA fin_iz

frented JSR borr_s2
fd1 JSR sig_rec
LDD paso_s2
```

martes, 05 de febrero de 2002

```
CPD #50
BEQ fin_fd
BRCLR PORTE,X $01 bumpd
BRCLR PORTE,X $04 fd1
JMP fin_fd

frente1 JSR borr_s2
        JSR sig_rec
fi1      LDD paso_s2
        CPD #50
        BEQ fin_fi
        BRCLR PORTE,X $02 bumpi
        BRCLR PORTE,X $08 fi1
        JMP fin_fi

bumpd    BSR atras
        BRA izqui
fin_iz   LDY #30
espera1  JSR temp
        DEY
        BNE espera1
        BRA frented
fin_fd   BRA pasa_obst

bumpi    BSR atras                                ; Si no se activan los dos el objeto
        STLO a_izq.
        BRA dere
fin_de   LDY #30
espera   JSR temp
        DEY
        BNE espera
        BRA frente1
fin_fi   BRA pasa_obst

atras    LDAA #$78                                ; Atr s durante un tiempo fijo
        STAA PORTA,X
        JSR borr_s1
a1       LDD paso_s1
        CPD #2
        BNE a1
        RTS

pasa_obst JSR sig_rec
        JSR borr_s1
sal_obst BRCLR PORTE,X $01 bumpd
        BRCLR PORTE,X $02 bumpi
        LDD paso_s1
        CPD #28
        BNE sal_obst
        JMP guard

fin      END
```

ANEXO E

PROGRAMA EN LENGUAJE C MENU.C

```
#include "REGS6811.H"
#include "OPTION.H"

char sw1, sw2;

main()
{
    while(1)
    {
        captura=1;
        retardo(200);
        while (captura!=0)
        {
            captura=PORTD & 0x20;
        }
        retardo(200);
        sw1=PORTE & 0x10;
        sw2=PORTE & 0x20;
        if (sw1!=0 && sw2!=0)
        {
            patrulla_cruz();
        }
        if (sw1!=0 && sw2==0)
        {
            esconder();
        }
        if (sw1==0 && sw2!=0)
        {
            patrulla_espiral();
        }
        if (sw1==0 && sw2==0)
        {
            modo_test();
        }
    }
}
```

ANEXO F

DISEÑO Y SIMULACIÓN DE DETECTOR ANALÓGICO MODULADO

DETECTOR ANALOGICO MODULADO

En este anexo están incluidos el diseño y simulación de un detector analógico modulado. Este detector, al ser analógico, no presenta las limitaciones que supone utilizar módulos detectores de infrarrojos digitales para la detección de la presa. De esta forma se conseguiría tres mejoras en el microbot depredador:

- Se podría saber la distancia aproximada que le separa de la presa, ya que la distancia es proporcional al voltaje de salida del detector.
- Se eliminaría el problema de los rebotes, siempre y cuando uno de los detectores analógicos recibiera la señal reflejada y otro, la señal directa de la presa.
- En este detector se podría variar su frecuencia de modulación, lo que implica una mayor inmunidad a las interferencias.

Este detector también tiene algunos inconvenientes, como su mayor ocupación de área, ya que está compuesto por componentes discretos.

Diseño del detector analógico modulado

Este detector analógico tiene un diseño muy básico. Consta de: un fotodiodo detector de infrarrojos, una etapa de eliminación de continua, un filtro paso banda y un detector de continua. El esquema de este detector se muestra en la figura AnexoF.1.

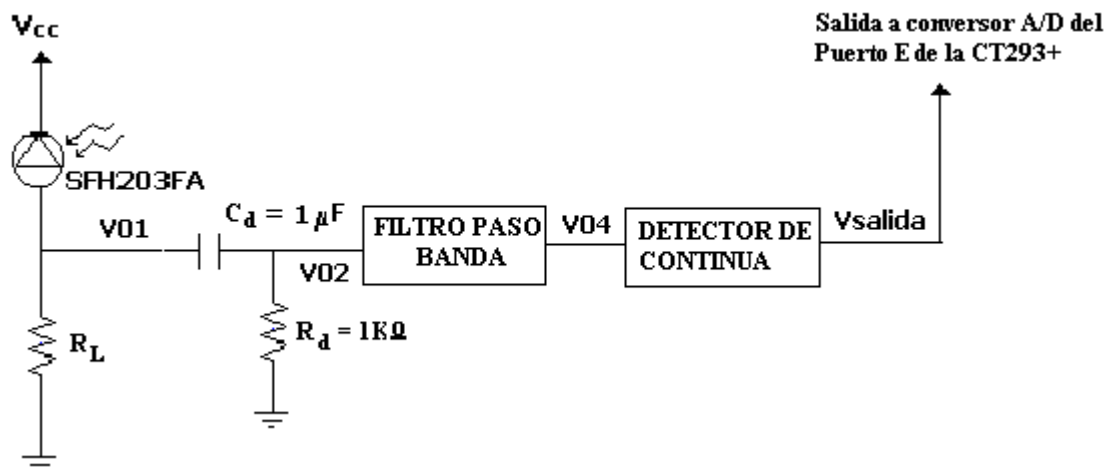


FIGURA AnexoF.1.- Esquema de detector analógico modulado.

Ahora se van a describir, de modo muy general, las distintas etapas de este detector.

Fotodiodo detector de infrarrojos. El fotodiodo detector es fundamental puesto es el elemento que debe detectar la señal emitida por un diodo emisor de infrarrojos. Este fotodiodo interesa que posea baja irradiancia, ángulo de sensibilidad estrecho y una respuesta en frecuencia adecuada. Si la respuesta en frecuencia del

fotodiodo no permite alcanzar la frecuencia a la que se ha centrado el filtro paso banda, el detector analógico modulado no funcionará de forma correcta.

Etapas de eliminación de continua. Consta de un simple condensador (C_d) y una resistencia (R_d). Esta etapa elimina la continua y las componentes a baja frecuencia. Esta etapa se utiliza puesto que las emisiones en el espectro infrarrojo que emite la luz solar son en continua y a baja frecuencia. Con esta etapa se reduce estas interferencias.

Filtro paso banda. El filtro diseñado para este detector es un filtro de cuarto orden, centrado a una frecuencia de 1 KHz y con un ancho de banda de 100 Hz, aunque se puede variar la frecuencia escalando los condensadores del filtro (NOTA: Hay que tener en cuenta que al escalar los condensadores el ancho de banda del filtro se ve afectado). La ganancia de este filtro es de aproximadamente la unidad, aunque se puede variar con el divisor de tensión que hay al principio del filtro. En la figura AnexoF.2 se puede observar que este filtro está formado por dos etapas.

Detector de continua. Puesto que necesitamos un nivel de continua para llevar al conversor analógico-digital, se puede usar un rectificador con condensador. Con este simple detector de continua se consigue una señal casi continua con un valor algo menor que la señal que llega al fotodiodo detector. Esta señal continua, conseguida a partir de la señal filtrada, es la que se llevaría a los conversores analógico-digitales de la tarjeta CT293+. En la figura AnexoF.3 se muestra este detector de continua, donde **V04** es la salida del filtro paso banda y **Vsalida** es la señal que se lleva a los conversores.

Aunque se podría haber introducido un demodulador, no se ha considerado necesario, ya que este detector de continua es suficiente para lograr tener un nivel de continua que pueda interpretar el conversor.

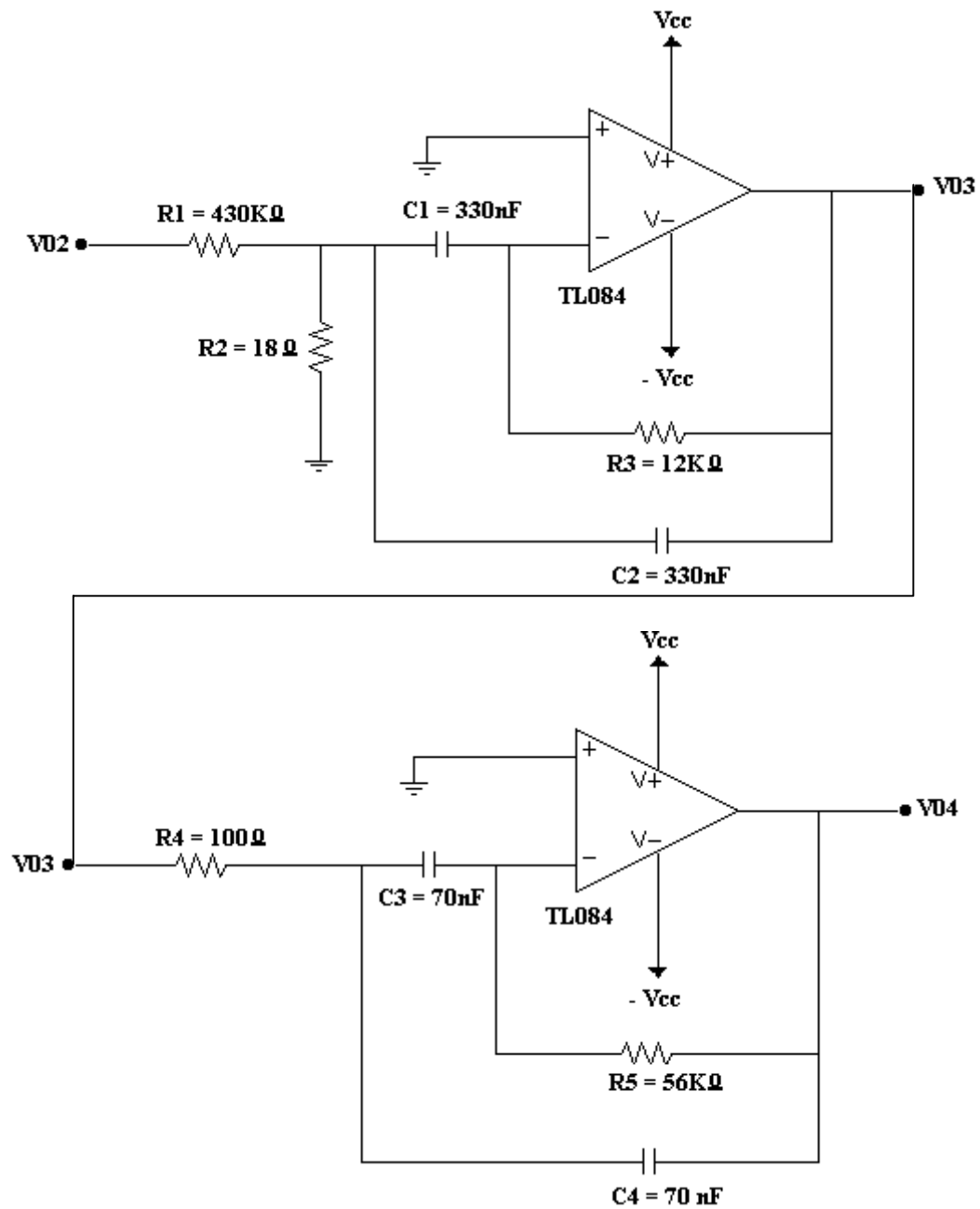


FIGURA AnexoF.2. Filtro Paso Banda de 4° Orden.

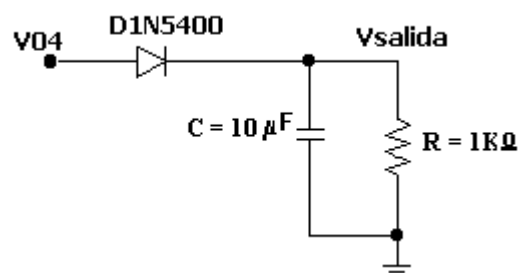
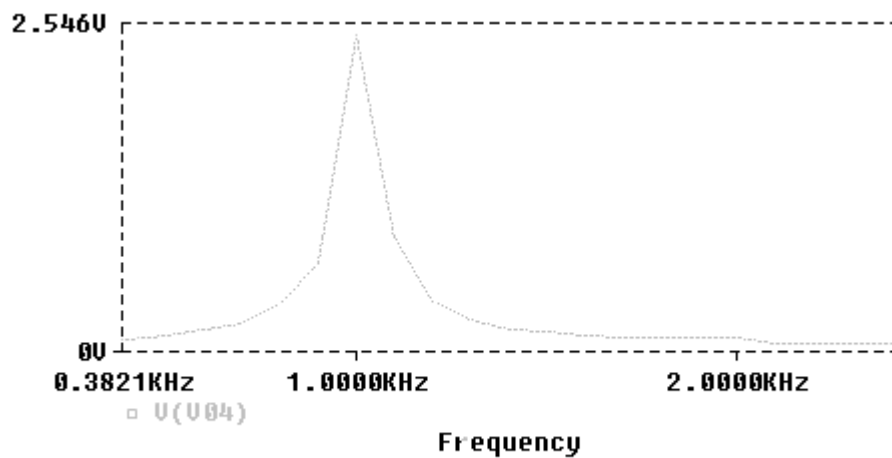


FIGURA AnexoF.3. Detector de continua.

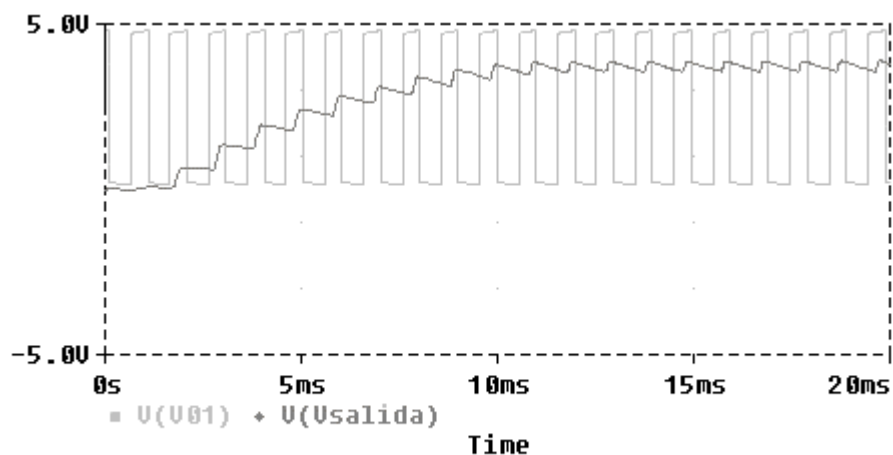
Simulación en PSPICE del detector analógico modulado

Para comprobar el correcto funcionamiento del detector se han llevado a cabo tres simulaciones. La primera (**Gráfica 1**) muestra la respuesta en frecuencia del filtro, y como se puede observar éste está centrado a una frecuencia de 1 KHz. En la segunda (**Gráfica 2**) se puede ver la salida del detector cuando a éste le llega una emisión modulada a 1 KHz. Por último, la tercera simulación (**Gráfica 3**) muestra la salida del detector si recibe una emisión modulada a 5 KHz.

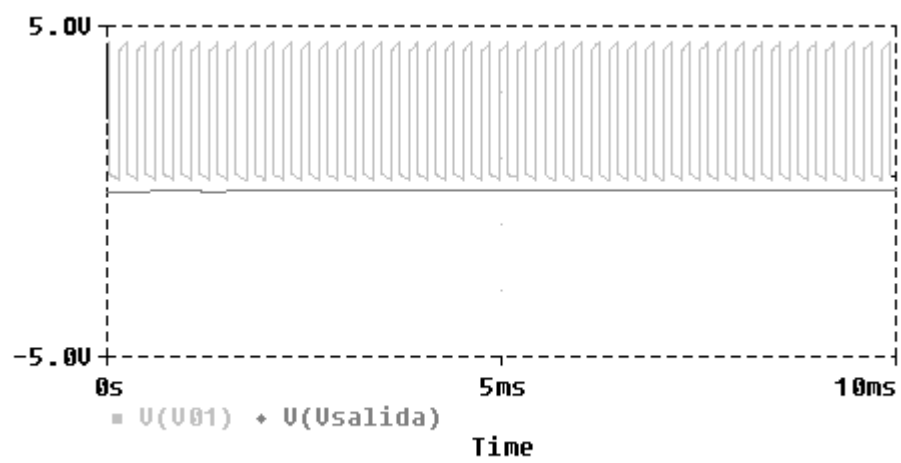
Como se puede ver en las gráficas, el filtro está centrado a 1 KHz, y las señales detectadas a otra frecuencia, fuera de la banda del filtro, son eliminadas.



GRAFICA 1. RESPUESTA EN FRECUENCIA DEL FILTRO PASO BANDA



GRAFICA 2. DETECCION DE UNA SEÑAL MODULADA A 1 KHz



GRAFICA 3. DETECCION DE UNA SEÑAL MODULADA A 5 KHz