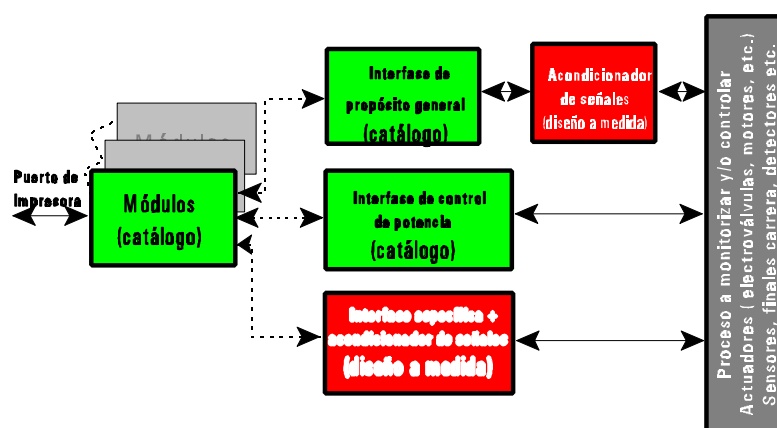


Sistema de Adquisición de Datos y Control desde PC



Estructura del sistema de adquisición de datos y control

Manual de Usuario

En la elaboración de éste manual se ha tratado de conjugar tres principios básicos : claridad, simplicidad y fiabilidad. Sin embargo, siendo conscientes de la dificultad para conseguir estos objetivos, rogamos al usuario nos envíe cuantas sugerencias crea interesantes para mejorarlo y corregirlo a "lipsoft@amsystem.es" o al Dpto. Técnico de LipSoft Electronics en Avda. Ejercito Español 22, 23005 Jaén. Tno:953-548264

Dpto. Técnico *LipSoft Electronics*. Abril 1998

Índice

I.1	Introducción	1
I.2	Conexión de los módulos	1
I.3	Precauciones importantes	2
I.4	Características de los módulos	3
I.4.1	<i>LipSoft</i> -ADO	3
I.4.2	<i>LipSoft</i> -ADT	5
I.4.3	<i>LipSoft</i> -AD8	6
I.4.4	<i>LipSoft</i> -AD12	7
I.4.5	<i>LipSoft</i> -IOT	9
I.4.6	<i>LipSoft</i> -DIG	10
I.4.7	<i>LipSoft</i> -AD2	11
I.5	Interfases de propósito general (IPG)	12
I.5.1	Introducción	12
I.5.2	Características comunes	12
I.5.3	Interfase <i>LipSoft</i> -ADO	13
I.5.4	Interfase <i>LipSoft</i> -IOT	13
I.5.5	Interfase <i>LipSoft</i> -ADT	14
I.5.6	Interfase <i>LipSoft</i> -AD8	15
I.5.7	Interfase <i>LipSoft</i> -AD12	15
I.6	Interfases de control de POTENCIA (ICP)	17
I.6.2	Interfase 8 Entradas Optoaisladas	17
I.6.3	Interfase 8 Salidas a Relé	18
I.6.4	Interfase 8 Salidas Opto - Transistor	19
I.6.5	Interfase 8 Salidas Opto - Triac	19
II.1	Introducción	21
II.2	Controlador de dispositivos LIPSOFT.SYS	21
II.2.1	Introducción	21
II.2.2	Instalación	21
II.2.3	Acceso al controlador	22
II.2.4	Apertura y Cierre del controlador	22
II.2.6	Programación del controlador	23
II.2.7	Adquisición de datos analógicos de modo automático	25
II.2.8	Control de los contadores	26
II.2.9	Gestión de la interrupción hardware (IRQ)	26
II.2.10	Acceso al controlador desde ensamblador	26
II.3	Librerías de enlace para DOS	27
II.3.1	Introducción	27
II.3.2	Declaración de las funciones contenidas en las librerías	27
II.3.3	Funciones disponibles	27
III.3.4	Funciones disponibles para lenguaje ensamblador	33
III.1	Introducción	36
III.2	Controles de usuario (para Visual Basic 3.0)	36
III.2.1	Control de Usuario AD12B* AD12R	36
III.2.2	Control de Usuario CONT	39
III.2.3	Control de Usuario DIG	41
III.2.4	Control de Usuario AD8B	43
III.3	Librerías de enlace dinámico (DLL) (Windows 3.x)	47
III.3.1	Librerías AD12B.DLL* o AD12R.DLL	48
III.3.2	Librería AD8B.DLL	48
III.3.3	Librería DIG.DLL	50
III.4	Librería de enlace dinámico DLL de 32 bits para Windows 95 y NT	53
III.4.1	Normas para utilizar las funciones	53

Manual de Usuario

Capítulo I

Características Técnicas

I.1 Introducción

LipSoft es un nuevo sistema de adquisición de datos para ordenadores personales PC compatibles. La comunicación entre el sistema y el ordenador se realiza a través del puerto de impresora (Centronics) que incorporan estos equipos. El sistema no interfiere en el proceso normal de trabajo con la impresora pudiendo ser utilizada normalmente (exceptuando el tiempo en que se está procediendo a la captación o envío de señales).

Existen diversos módulos ADT, IOT, ADQ, DIG, AD8 y AD12 que pueden ser combinados para formar cadenas complejas de adquisición y control. Estos pueden conectarse bien en serie (uno tras otro) o en paralelo (por medio de un expansor de puerto paralelo), de cualquier modo se ha de respetar el direccionamiento ya que cada tipo permite un máximo de direcciones. Para cada módulo se dispone de una tarjeta de interfase que permiten la conexión de las señales del exterior con la electrónica de los módulos a través de regletas con tornillos. Estas interfases incluyen regulador de tensión, visualización del estado de las líneas digitales mediante LEDs (diodos emisores de luz) y soporte para insertar divisores o filtros en las entradas analógicas.

La comunicación entre el ordenador y los módulos se realiza mediante rutinas escritas en lenguaje ensamblador que se encargan del protocolo de comunicación. El usuario no precisa ningún conocimiento del proceso interno de estas funciones, y solo ha de respetar las directrices para la realización de llamadas a éstas. Las funciones invocadas devolverán la información solicitada o realizarán la operación precisa.

Si programa bajo el sistema operativo DOS dispone de dos alternativas para manejar los módulos desde su aplicación:

- Mediante un controlador de dispositivos instalable (LIPSOFT.SYS). Capítulo 2 de este manual.
- Enlazando su programa con las funciones en las librerías que se suministran. Capítulo 2.

Si programa bajo Windows ver 3.x, el manejo de los módulos puede realizarse de dos modos:

- Mediante llamadas a funciones en librerías de enlace dinámico (DLL). Capítulo 3 de este manual.
- Si su lenguaje de programación es Visual Basic o Visual C dispone además de la opción a, de un conjunto de controles de usuario (custom control) que le facilitarán enormemente el desarrollo de su aplicación. Capítulo 3.

En el disco de utilidades que acompaña al producto se encuentran un conjunto de ejemplos para los distintos lenguajes y entornos de programación.

I.2 Conexión de los módulos

Los pasos que debe seguir para utilizar los módulos han de ser los siguientes:

- Con el ordenador apagado interconecte el/los módulos e interfases y en su extremo la impresora si lo desea.
- En el caso de utilizar las tarjetas de interfase o si su diseño suministra alimentación externa a los módulos, ésta debe de estar desconectada (es aconsejable).
- Encienda el ordenador. Espere hasta que aparezca el prompt del S.O.
- Alimente las interfases.
- Encienda la impresora.

Al terminar de utilizar los módulos es conveniente que, antes de apagar el ordenador, desconecte la alimentación que les puede llegar a éstos a través de las interfases (o de su sistema).

Los módulos pueden conectarse en **serie** uno tras otro, en **paralelo** o **mezclados**. Para la conexión en paralelo, figura 1, se precisa de un conector apropiado que multiplique el número de salidas del puerto. Este conector puede adquirirlo en el mismo proveedor de los módulos o bien construirlo por usted mismo. En este caso solo precisa de cable plano de 25 hilos y varios conectores hembra Sub-D 25 para cable plano. Para realizar la conexión serie, figura 2, basta conectar uno tras otro los módulos y el primero de la cadena al puerto de impresora del ordenador.

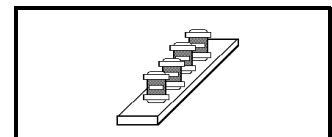


Figura. 1. Conexión de los módulos **LipSoft** en paralelo

Para que el ordenador pueda identificar el módulo con el que desea establecer comunicación es preciso que éste se

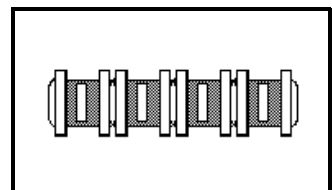


Figura. 2. Conexión de los módulos **LipSoft** en serie

encuentre bien direccionado. La programación de la dirección se realiza situando las patillas de direcciones, del conector de salida, a un nivel lógico bajo o alto (en las interfases para los módulos se realiza mediante un interruptor DIP o Jumper).

La siguiente tabla resume el consumo de direcciones que realizan cada tipo de módulo así como el máximo número de direcciones disponibles para cada puerto paralelo.

Módulo <i>LipSoft</i>	ADQ	ADT	AD8	DIG	IOT	AD12	Máx. Direc
A/D 8bits	1	1	2	-	-	-	8
A/D 12bits	-	-	-	-	-	1	3
I/O digital	1	1	-	2	1	-	8
Contador	-	1	-	-	2	-	2

Un ejemplo de conexión para obtener 20 canales A/D de 8 bits, 5 Canales D/A 8 bits, 64 líneas I/O digitales, 8 canales A/D 12 bits, 1 Canal D/A de 12 Bits y 1 Contadores de 6 décadas cada uno podría ser:

Módulos montados en un mismo puerto:

$$1 \text{ ADQ} + 2 \text{ AD8} + 3 \text{ DIG} + 1 \text{ IOT} + 1 \text{ AD12}$$

Consumo de direcciones

$$\begin{aligned} \text{A/D 8bits} & (1_{\text{ADQ}} * 1) + (2_{\text{AD8}} * 2) & = 5 \\ \text{I/O Digital} & (1_{\text{ADQ}} * 1) + (3_{\text{DIG}} * 2) + (1_{\text{IOT}} * 1) & = 8 \\ \text{Contadores} & (1_{\text{IOT}} * 1) & = 1 \\ \text{A/D 12bits} & (1_{\text{AD12}} * 1) & = 1 \end{aligned}$$

Las mismas especificaciones podrían obtenerse montando:

$$5 \text{ ADQ} + 1 \text{ IOT} + 1 \text{ DIG} + 1 \text{ AD12}$$

Consumo de direcciones

$$\begin{aligned} \text{A/D 8bits} & (5_{\text{ADQ}} * 1) & = 5 \\ \text{I/O Digital} & (5_{\text{ADQ}} * 1) + (1_{\text{IOT}} * 1) + (1_{\text{DIG}} * 2) & = 8 \\ \text{Contadores} & (1_{\text{IOT}} * 1) & = 1 \\ \text{A/D 12bits} & (1_{\text{AD12}} * 1) & = 1 \end{aligned}$$

Tenga en cuenta que existen muchas combinaciones para conseguir las mismas especificaciones. **Solamente ha de respetar que el consumo de direcciones de cada ítem sea menor o igual al máximo permitido.**

Si tiene alguna duda contacte con su proveedor que le indicará la combinación más apropiada.

1.3 Precauciones importantes

Si conecta una **impresora** a la cadena de módulos *LipSoft* es preciso que esta se encuentre en "ON LINE", es decir encendida y con papel, para que los módulos funcionen correctamente.

Si **trabaja** bajo **Windows** y utiliza el módulo *LipSoft*-AD12 es preciso que seleccione en el PANEL CONTROL / IMPRESORAS con el Botón Conectar..., la opciones de conexión "LPT1.DOS" o "LPT2.DOS" para que Windows controle la impresora tal y como lo hace bajo DOS.

Aunque los módulos han sido ampliamente probados en un gran número de configuraciones Ordenador + Placa paralelo + Impresora + Software, son tantas las posibles configuraciones que **NO se puede garantizar** que su configuración específica **no cause problemas**. En tal caso, la solución más rápida es que conecte la impresora o los módulos a otro puerto paralelo, o bien introduzca un conmutador para activar la impresora o los módulos independientemente.

Debido a su muy bajo consumo, los módulos, no precisan por lo general de alimentación exterior para su funcionamiento ya que la toman de las líneas de datos del propio puerto. Sin embargo, en algunos equipos, los niveles en estas líneas son insuficientes y es preciso disponer de alimentación auxiliar. De cualquier modo los módulos solo son capaces de entregar pequeña señal a su salida (unos pocos μA) por lo que si se trata de excitar cualquier carga será preciso suministrar exteriormente la energía.

Ningún módulo **admite** como entrada tensiones **negativas**. Un simple diodo schottky o de germanio polarizado inversamente resulta una protección adecuada figura 3. Es conveniente limitar

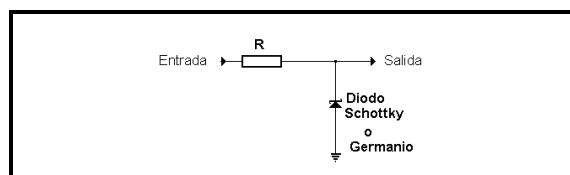


Figura. 3. Protección de las entradas contra tensiones negativas.

mediante una resistencia la posible corriente que pueda circular por este diodo.

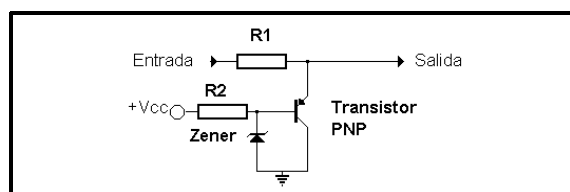


Figura. 4. Protección de las entradas analógicas contra sobretensiones.

En **ningún caso** los niveles de entrada deben **superar** los máximos **rangos** para las entradas (analógicas o digitales). Una protección simple puede realizarse mediante un diodo zener de valor adecuado. Sin embargo, este tipo de protección no resulta aconsejable en las entradas analógicas, pues introduce un error importante en la señal medida. Protecciones activas como muestra la figura 4 suele ser más apropiado. El Zener debería ser de una tensión 0'6 Voltios inferior a la que se desea proteger.

I.4 Características de los módulos

En los siguientes apartados se van a describir las características de funcionamiento, de conexión y eléctricas de los módulos *LipSoft*. Debido a que los módulos comparte ciertas características comunes, y con objeto de no duplicar éstas, se remitirá al lector al primer apartado donde se indicaron.

I.4.1 *LipSoft*-ADQ

I.4.1.1 Descripción

Este módulo permite la adquisición y control de señales tanto analógicas como digitales. Dispone de 4 Canales de entrada analógica unipolar con 8 bits de resolución. Es posible programar los canales en cuatro modos distintos. El rango de señal a su entrada va de 0 a 2.5 Voltios, esto equivale a una resolución de unos 10 mV. Contiene un canal de conversión Digital-Analógico unipolar de 8 bits de resolución. El rango de señal a su salida va de 0 a 2.5 Voltios.

El módulo incluye 8 líneas de entrada/salida cuasibidireccionales digitales que aceptan o entregan niveles TTL o CMOS a 5 V. Para utilizar una línea como entrada ha de ponerla a nivel alto (1 lógico = 5V). Al leer su estado tendremos el nivel que existe realmente.

El módulo dispone de 3 líneas para seleccionar la dirección a la que atiende.

Este módulo está indicado en aquellas aplicaciones donde se precisen controlar y registrar señales analógicas con una resolución media y que además se requieran un limitado número de líneas de control digitales (0 o 1). Ejemplos de utilización: Sistemas de control y registro de temperatura, humedad, caudal, presión etc. Equipos de medida automáticos. Generación de señales etc.

Para poder utilizar los módulos, que contienen canales A/D, con señales bipolares, deberá desplazar los niveles de tensión hasta el rango de trabajo de éstos. Un amplificador operacional de precisión, junto a una referencia estable, suele ser una solución válida.

Para obtener una óptima precisión y resolución en la medida, es conveniente, adaptar previamente las señales en las entradas analógicas mediante divisores o amplificadores de tensión, para que éstas queden lo más aproximadamente posible dentro del rango de medida.

Cada vez que se produce un cambio en el estado de las líneas digitales se pone a nivel bajo la línea de interrupción del puerto paralelo IRQ. Esta línea se encuentra conectada generalmente a la interrupción

(IRQ7 o IRQ5). Para la generación de una interrupción Hardware vía IRQ5 o IRQ7, es preciso que se produzca un **pulso** desde un nivel bajo a nivel alto. Esto puede conseguirse anteponiendo el circuito de la figura 5 a la línea I/O que deseamos produzca la interrupción. Un cambio desde nivel alto a bajo provocará la generación de un pulso bajo, cuya duración a nivel bajo vendrá determinada aproximadamente por la constante de tiempo de carga del condensador es decir $t \approx R2 * C$ segundos.

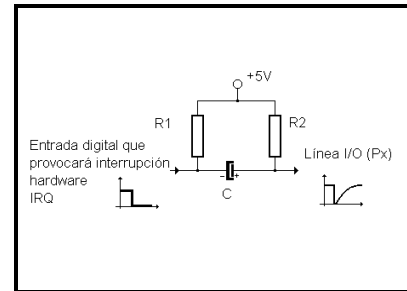


Figura. 5. Circuito para la generación de una interrupción hardware (IRQx) utilizando una línea I/O como entrada de interrupción. Activación a nivel bajo.

I.4.1.2 Asignación de patillas al conector de salida del módulo *LipSoft*-ADQ

Patilla	Nombre	Descripción
1	P3	Línea Digital bit 3
3	P2	Línea Digital bit 2
5	P1	Línea Digital bit 1
7	P0	Línea Digital bit 0
9	A1	Línea Dirección 1
11	A0	Línea Dirección 0
13	AD0	Canal Entr. Analóg. 0
15	AD1	Canal Entr. Analóg. 1
17	AD2	Canal Entr. Analóg. 2
19	AD3	Canal Entr. Analóg. 3

DA	20	19	AD3
Vcc	18	17	AD2
DATA	16	15	AD1
CLK	14	13	AD0
A 2	12	11	A0
P7	10	9	A1
P6	8	7	P0
P5	6	5	P1
P4	4	3	P2
GND	2	1	P3

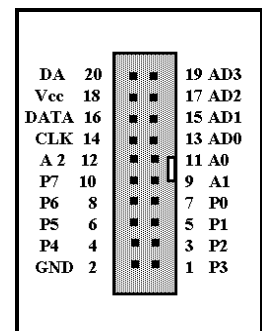


Figura. 6. Conector salida *LipSoft*-ADQ

2	GND 0V (Alim. y señal)
4	P4 Línea Digital bit 4
6	P5 Línea Digital bit 5
8	P6 Línea Digital bit 6
10	P7 Línea Digital bit 7
12	A2 Línea Dirección 2
14	CLK Sincronismo
16	DATA Datos
18	VCC Alim. exterior +5V DC
20	DA Canal Salida Analógica

1.4.1.3 Características eléctricas y de funcionamiento

Parte de las características que aquí se resumen son intrínsecas a los componentes utilizados en los módulos. Estas características son valores típicos a 25° C.

Canales conversión Analógico/Digital (AD0, AD1, AD2, AD3) y Digital/Analógico (DA)

Canales Analógico/Digital:	4
Canales Digital/Analógico:	1
Resolución:	8 Bits
Rango entrada o salida:	Unipolar de 0 a 2.5 V ⇒ (LSB=9.8 mV)
Error Offset (ADx):	30 mV
Error Linealidad:	±1 1/2 LSB
Error Ganancia:	5%
Factor rechazo modo común:	60 dB
Frecuencia Muestreo Máxima:	10 Khz ⁽¹⁾ .
Modos de trabajo canales Adx:	

Modo 0 = 4 canales simples AD0, AD1, AD2, AD3

Modo 1 = 3 Diferen. con respecto al canal 3
AD0-AD3, AD1-AD3, AD2-AD3

Modo 2 = 2 Simples y 1 Diferencial
AD0, AD1, AD2-AD3

Modo 3 = 2 Diferenciales
AD0-AD1, AD2-AD3

En modo diferencial los resultados de la conversión viene representada en complemento a 2.

Corriente Entrada (ADx):	< 100 nA
Capacidad entrada (ADx):	40 pF
Error Offset (DA):	50 mV

Notas importantes:

1) La salida del convertor D/A se encuentra protegida mediante una resistencia de 1 kΩ, y por tanto, la impedancia (resistiva) de salida será de este valor. Téngalo en cuenta si trata de excitar a sistemas con bajas impedancias de entrada. Una solución optima para evitar cualquier error debido a este fenómeno consiste en

desacoplar la fuente de la carga mediante un amplificador operacional en montaje seguidor de tensión.

2) La lectura de una entrada analógica que se encuentre al aire, es decir, sin conectar con ningún sistema que fije un nivel concreto, puede tomar valores arbitrarios y aleatorios. Se recomienda como medida de seguridad que las entradas no utilizadas se conecten a masa (0 Voltios).

Líneas Digitales entrada/salida (P0...P7) cuasidireccionales⁽²⁾

Número de Líneas I/O:	8
Entrada a nivel bajo:	-0.5 V a 1.3 V
Entrada a nivel alto:	3.5 V a 5.5 V
Máxima corriente capaz de absorber a nivel bajo (Salida=1 V):	10 .. 30 mA
Máxima corriente capaz de entregar (Salida=0 V):	30 μA
Máxima frecuencia de cambio en los estados de las líneas:	10 Khz ⁽¹⁾ .

Líneas de Direcciones (A0, A1, A2)

Estas tres líneas permiten programar las 8 direcciones posibles para los conversores A/D 8 bits e I/O Digital. Para fijar un 1 (nivel alto) es preciso poner la línea correspondiente a una tensión de 5 V mediante una resistencia cuyo valor debe estar comprendido entre 4K7 y 22K. Para fijar un 0 (nivel bajo) basta conectar la línea a 0 V=GND.

La tabla 1 muestra la dirección asignada a los subsistemas en función de la combinación de las líneas.

A2	A1	A0	Dirección
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

Tabla 1. Dirección asignada en función del ajuste de A2,A1,A0.

Líneas de control (CLK, DATA)

Estas líneas no deben ser utilizadas por el usuario (dejadas sin conectar) y sirven para futuras expansiones de los módulos.

Líneas de Alimentación (VCC y GND)

El módulo no precisa por lo general alimentación externa. En ciertos equipos donde los niveles de las señales del puerto de impresora (Centronics) son muy débiles si será preciso dotarlo de una alimentación auxiliar, si bien al ser el consumo extremadamente reducido ($< 1\text{mA}$) las especificaciones de la fuente son muy básicas ($5\text{V} \pm 5\%$, 2mA). La línea GND (0 V) se utiliza como terminal negativo de alimentación y corresponde además al terminal de menor potencial para las entradas/salidas analógicas, por tanto, deberá procurar un buen diseño en los lazos de masa para evitar errores de medida debido a diferencias de potencial en GND.

(1) Esta frecuencia corresponde a la máxima que los dispositivos internos son capaces de alcanzar, sin embargo la máxima frecuencia a la que usted podrá muestrear va a depender de múltiples factores, si bien los más importantes son: software, sistema operativo y velocidad de proceso de su ordenador.

(2) Las líneas son cuasibidireccionales ya que, para actuar como entrada, no dejan en estado de alta impedancia su excitador de salida. Como su capacidad de entregar corriente (cuando está a nivel alto) está limitada a $< 100\ \mu\text{A}$, si un dispositivo lógico fija un nivel bajo, la línea quedará a este nivel, sin problemas para éste. Por ejemplo una puerta TTL estándar, es capaz de absorber más 16 veces esa corriente sin que el nivel lógico quede fuera de los límites para esta familia. Al leer el estado de la línea el módulo leerá el valor correcto es decir un nivel bajo. En el caso de que el dispositivo fije un nivel alto no aparecerá ningún problema ya que los niveles en la línea se homogeneizaría y se leerá el valor correcto.

En resumen para que una línea (P0 .. P7) pueda actuar como entrada es preciso que se programe (mediante el software que acompaña a los módulos) a nivel alto.

gestionar de manera cómoda ciertas señales digitales imposibles de controlar con las entradas I/O simples. Ejemplo de esto podría ser el conteo de los impulsos de un codificador incremental, medida precisa de la frecuencia de un lazo V/F etc.

La alarma incluida en el contador es un registro de 6 dígitos que puede ser programado por el usuario. Cuando la totalización (el número de impulsos contados) se hace igual al contenido del registro de alarma, el nivel de la línea INT (interrupción) baja (0 V). Si esta línea se cablea, tal como muestra la figura 7, a la línea de solicitud de interrupción del puerto paralelo (IRQ7 o IRQ5) es posible controlar rápidamente este suceso. Esto permite que el software desarrollado pueda estar realizando otras tareas y solo preocuparse del contador cuando se produce el citado evento. De cualquier modo, la generación de ésta alarma puede habilitarse/deshabilitarse (vía software) en cualquier momento. Las líneas de interrupción (INT0 e INT1) de los contadores y la IRQ están disponibles en el conector de salida de modo que puedan ser cableadas según sus necesidades.

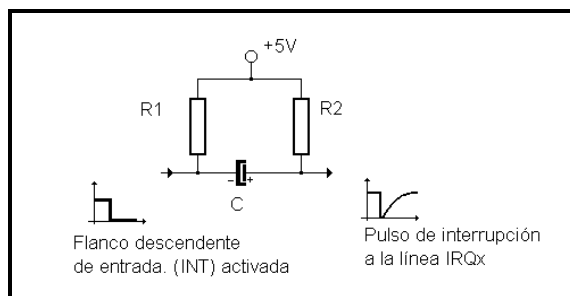


Figura. 7. Circuito para la generación de una interrupción hardware (IRQx) utilizando la salida (INTx) de interrupción de los contadores.

1.4.2 LipSoft-ADT

1.4.2.1 Descripción

Se trata de un módulo que además de las prestaciones del módulo LipSoft-ADQ incorpora un contador de 6 décadas (0..999999) con posibilidad de programación de alarmas.

El contador puede ser utilizado para múltiples tareas, las más habituales son la totalización de sucesos discretos y la generación de temporizaciones precisas. Éste acepta señales lógicas con niveles TTL o CMOS a 5V. Su frecuencia máxima de conteo es de 1 Mhz.

El funcionamiento del resto del módulo ADT es idéntico al ADQ.

El módulo dispone de 1 línea para seleccionar la dirección a la que atiende.

Las aplicaciones del módulo son muy variadas incluyendo todas las del ADQ. Además, el contador que incorpora, permite

1.4.2.2 Asignación de patillas al conector de salida del módulo LipSoft-ADT

Patilla	Nombre
Descripción	
1	P3 Línea Digital bit 3
3	P2 " " 2
5	P1 " " 1
7	P0 Línea Digital bit 0
9	IRQ Solicitud interr p. paralelo
11	A0 Línea Dirección 0
13	AD0 Canal Entr. Analógica 0
15	AD1 Canal Entr. Analógica 1
17	AD2 Canal Entr. Analógica 2
19	AD3 Canal Entr. Analógica 3
2	GND 0 V (Alimentación y señal)
4	P4 Línea Digital bit 4
6	P5 Línea Digital bit 5

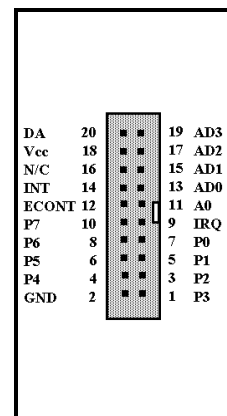


Figura. 8. Conector salida LipSoft-ADT

8	P6	Línea Digital bit 6
10	P7	Línea Digital bit 7
12	ECONT	Entrada contaje
14	INT	Interrupción
16	N/C	No conectado
18	VCC	Alimentación ext. +5VDC
20	DA	Canal Salida Analógica

1.4.2.3 Características eléctricas y de funcionamiento.

Parte de las características que aquí se resumen son intrínsecas a los componentes utilizados en los módulos. Estas características son valores típicos a 25° C.

Canales conversión Analógico/Digital (AD0, AD1, AD2, AD3) y Digital/Analógico (DA)

Ver la misma líneas en el módulo *LipSoft-ADQ*.

Líneas Digitales entrada/salida (P0...P7) cuasibidireccionales

Ver la misma líneas en el módulo *LipSoft-ADQ*.

Contador (líneas ECONT, INT)

Capacidad de contaje:	999999 impulsos
Frecuencia máxima contaje:	1 Mhz
Entrada pulsos (ECONT):	Acepta niveles lógicos TTL o CMOS 5 V
Disparo:	flanco ascendente
Salida interrupción (INT):	Tipo colector(drenador) abierto. Admite OR cableado.

Línea solicitud de interrupción del puerto impresora (IRQ)

Esta línea que corresponde a -ACK (Reconocimiento = patilla 10 conector puerto de impresora SUB-D 25) está conectada a través de la placa paralelo al controlador de interrupciones del PC. Para poder utilizarla es preciso que re programe adecuadamente el controlador y la habilite, a la vez que cree una rutina de interrupción para su atención. El software que incorporan los módulos *LipSoft* incluyen todas estas funciones de modo que la utilización de la interrupción es inmediata. Para que la activación de la alarma en el contador pueda ser atendida es preciso que conecte la línea INT (patilla 14) e IRQ (patilla 9) del conector de salida a través del circuito de la figura 7.

Línea de Dirección (A0)

Esta línea permiten programar las 2 direcciones posibles para los conversores A/D 8 bits, I/O Digital y Contador que incluye el módulo. Para fijar un 1(nivel alto) es preciso poner la línea correspondiente a una tensión de 5 V mediante una resistencia cuyo valor debe estar comprendido entre 4K7 y 22K. Para fijar un 0 (nivel bajo) basta conectar la línea a 0V=GND.

A0 = 0		A0 = 1	
Dir. del A/D 8 bits = 0		Dir. del A/D 8 bits = 1	
Dir. del I/O digital = 0		Dir. del I/O digital = 1	
Dir. del Contador = 0		Dir. del Contador = 1	

Líneas de Alimentación (VCC y GND).

Ver la misma líneas en el módulo *LipSoft-ADQ*.

1.4.3 *LipSoft-AD8*

1.4.3.1 Descripción

Este módulo permite la adquisición y control de señales analógicas. El módulo está dividido en dos grupos de 4 Canales de entrada analógica (A/D) unipolar con 8 bits de resolución y un canal de salida analógica (D/A) unipolar con idéntica resolución. Es posible programar cada grupo de canales A/D en cuatro modos distintos. El rango de señal a su entrada va de 0 a 2.5 Voltios, esto equivale a una resolución de unos 10 mV. Contiene además dos canales de conversión Digital-Analógico unipolar de 8 bits de resolución. El rango de señal a su salida va de 0 a 2.5 Voltios.

El módulo dispone de 2 líneas para seleccionar la dirección a la que atiende.

Está indicado en aquellas aplicaciones de control/ registro analógico donde el número de canales sea elevado (hasta 8). Ejemplos: monitorización multivariable, control multivariable, etc.

Para poder utilizar el módulo con señales bipolares, deberá desplazar los niveles de tensión hasta el rango de trabajo de los módulos. Un amplificador operacional de precisión junto a una referencia estable suele ser una solución válida.

Para obtener una óptima precisión y resolución en la medida es conveniente adaptar previamente las señales en las entradas analógicas mediante divisores o amplificadores de tensión para que quede lo más aproximadamente posible dentro del rango de medida de los módulos.

1.4.3.2 Asignación de patillas al conector de salida del módulo *LipSoft-AD8*

Patilla	Nombre	Descripción
---------	--------	-------------

- 1 N/C No conectada
- 3 N/C No conectada
- 5 N/C No conectada
- 7 N/C No conectada
- 9 A1 Línea Dirección 1
- 11 A2 Línea Dirección 2
- 13 AD0_0 Ent. Analo. 0 (gp 0)
- 15 AD1_0 Ent. Analo. 1 (gp 0)
- 17 AD2_0 Ent. Analo. 2 (gp 0)
- 19 AD3_0 Ent. Analo. 3 (gp 0)

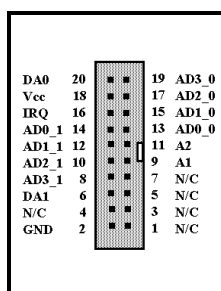


Figura. 9. Conector salida LipSoft-AD8

- 2 GND 0 V (Alimentación y señal)
- 4 N/C No conectada
- 6 DA1 Canal Salida Analo. (gp 1)
- 8 AD3_1 Canal Ent. Analo. 3 (gp 1)
- 10 AD2_1 Canal Ent. Analo. 2 (gp 1)
- 12 AD1_1 Canal Ent. Analo. 1 (gp 1)
- 14 AD0_1 Canal Ent. Analo. 0 (gp 1)
- 16 IRQ Solicitud interr. p. paralelo
- 18 VCC Aliment. exterior +5V DC
- 20 DA0 Canal Sal. Analo.(grupo 0)

0	1	2	3
1	0	4	5
1	1	6	7

Tabla 2. Asignación de direcciones para el módulo LipSoft-AD8.

Línea solicitud de interrupción del puerto impresora (IRQ)

Esta línea que corresponde a -ACK (Reconocimiento = patilla 10 conector impresora SUB-D 25) está conectada a través de la placa paralelo al controlador de interrupciones del PC. Para poder utilizarla es preciso que re programe adecuadamente el controlador y la habilite, a la vez que cree una rutina de interrupción para su atención. El software que incorporan los módulos LipSoft incluyen todas estas funciones de modo que la utilización de la interrupción es inmediata.

Líneas de Alimentación (VCC y GND)

Ver la misma líneas en el módulo LipSoft-ADQ.

I.4.3.3 Características eléctricas y de funcionamiento.

Parte de las características que aquí se resumen son intrínsecas a los componentes utilizados en los módulos. Estas características son valores típicos a 25° C.

Grupos de Canales conversión Analógico/Digital (ADO_x, AD1_x, AD2_x, AD3_x) y Digital/Analógico (DAX)

Ver la misma líneas en el módulo LipSoft-ADQ.

Líneas de Direcciones (A1, A2)

Estas dos líneas permiten programar las 8 direcciones posibles para los subsistemas de conversión A/D y D/A de 8 bits. El grupo 0 responde a direcciones pares y el grupo 1 a las impares. Para fijar un 1 (nivel alto) es preciso poner la línea correspondiente a una tensión de 5 V mediante una resistencia cuyo valor debe estar comprendido entre 4K7 y 22K. Para fijar un 0 (nivel bajo) basta conectar la línea a 0 V=GND. La tabla 2 muestra la dirección asignada a los grupos en función de la combinación de las líneas.

A2	A1	Dirección Grupo 0	Dirección Grupo 1
0	0	0	1

I.4.4 LipSoft-AD12

I.4.4.1 Descripción

Este módulo permite la adquisición y control de señales analógicas. Dispone de 8 canales de entrada analógica unipolar con 12 bits de resolución. Es posible programar los canales en modo simple o diferencial. El rango de señal a su entrada va de 0 a 4'095 Voltios, esto equivale a una resolución de 1 mV. Contiene un canal de conversión Digital-Analógico unipolar de 12 bits de resolución. El rango de señal a su salida va de 0 a 4'095 Voltios.

Incluye en su conector de salida de la línea de solicitud de interrupción (IRQ). Ésta puede ser utilizada para comandar el disparo en el proceso de muestreo .

Dispone de 2 líneas para seleccionar las 3 direcciones a las que atiende, es decir, se pueden conectar hasta 3 módulos en un mismo puerto Centronics.

Está indicado para la monitorización y control de señales analógicas con una elevada precisión. Sus 12 bits de resolución le permiten registrar y controlar de manera precisa la evolución de las magnitudes medidas. Sus aplicaciones son las mismas que las del módulo AD8 cuando se requiere una mayor precisión.

Para poder utilizar el módulo con señales bipolares, deberá desplazar los niveles de tensión hasta el rango de trabajo de éste. Un amplificador operacional de precisión junto a una referencia estable suele ser una solución válida.

Para obtener una óptima precisión y resolución en la medida es conveniente adaptar previamente las señales en las entradas analógicas mediante divisores o amplificadores de tensión para que quede lo más aproximadamente posible dentro del rango de medida de los módulos.

I.4.4.2 Asignación de patillas al conector de salida del módulo *LipSoft-AD12*

Patilla Nombre Descripción

1	AD0 Canal Ent. Analógica 0
3	AD1 Canal Ent. Analógica 1
5	AD2 Canal Ent. Analógica 2
7	AD3 Canal Ent. Analógica 3
9	AD4 Canal Ent. Analógica 4
11	AD5 Canal Ent. Analógica 5
13	AD6 Canal Ent. Analógica 6
15	AD7 Canal Ent. Analógica 7
17	A0 Selector dirección 0
19	A1 Selector dirección 1
2	GND 0V (Alimentación y señal)
4	X0 X dirección 0
6	X1 X dirección 1
8	Y0 Y dirección 0
10	Y1 Y dirección 1
12	Z0 Z dirección 0
14	Z1 Z dirección 1
16	DA Canal Salida Analógica
18	VCC Aliment. ext. +5V DC
20	IRQ Solicitud interrupción puerto paralelo

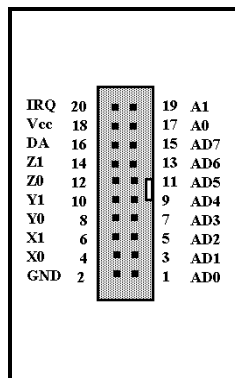


Figura. 10. Conector de salida *LipSoft-AD12*

Modos de trabajo canales ADx:

Modo 0 = 8 canales simples
AD0, AD1, AD2, AD3, AD4, AD5, AD6, AD7
Modo 1 = 4 Diferenciales
AD0-AD1, AD2-AD3, AD4-AD5, AD6-AD7

En modo diferencial el módulo realiza la conversión cargando un condensador interno con la señal en el canal + y descargando éste proporcionalmente a la señal en el canal -, después realiza la conversión de esta diferencia. Por tanto, para obtener una medida precisa es necesario que la tensión en los canales no varíe significativamente en este intervalo de tiempo.

Corriente Entrada (ADx): < 100 nA
Capacidad entrada (ADx): 20 pF
Máximo Error Offset (DA): 4 LSB

Nota importante: La lectura de una entrada analógica que se encuentre al aire, es decir, sin conectar con ningún sistema que fije un nivel concreto, puede tomar valores arbitrarios y aleatorios. Se recomienda como medida de seguridad que las entradas no utilizadas se conecten a masa (0 Voltios).

Líneas de Direcciones (A0, A1) y (X0, X1, Y0, Y1, Z0, Z1)

Estas líneas permiten programar 3 direcciones posibles de los módulos. La programación de la dirección se realiza uniendo las líneas Ax y Xx o Ax e Yx o Ax y Zx respectivamente. La tabla 3 muestra la asignación de direcciones posible.

Dirección asignada	Líneas unidas
0	A0-X0 y A1-X1
1	A0-Y0 y A1-Y1
2	A0-Z0 y A1-Z1

Tabla 3. Asignación de direcciones en el módulo *LipSoft-AD12*

Nota importante: El módulo dejará de funcionar si se utiliza cualquier otra asignación distinta a la indicada en la tabla.

I.4.4.3 Características eléctricas y de funcionamiento

Parte de las características que aquí se resumen son intrínsecas a los componentes utilizados en los módulos. Estas características son valores típicos a 25° C.

Canales conversión Analógico/Digital (AD0... AD7) y Digital/Analógico (DA)

Canales Analógico/Digital: 8
Canales Digital/Analógico: 1
Resolución: 12 Bits
Rango entrada o salida: Unipolar 0.. 4'096 V (LSB=1 mV)
Error Offset (ADx): ±3 LSB
Error Linealidad: ±1 LSB
Error Ganancia: ±1 LSB
Frecuencia Muestreo Máxima: 33 Khz⁽³⁾.

Línea solicitud de interrupción del puerto impresora (IRQ)

Esta línea que corresponde a -ACK (Reconocimiento = patilla 10 conector impresora SUB-D 25) está conectada a través de la placa paralelo al controlador de interrupciones del PC. Para poder utilizarla es preciso que re programe adecuadamente el controlador y la habilite, a la vez que cree una rutina de interrupción para su atención. El software que incorporan los módulos *LipSoft* incluyen todas estas funciones de modo que la utilización de la interrupción es inmediata.

Líneas de Alimentación (VCC y GND)

El módulo no precisan por lo general alimentación externa. En ciertos equipos donde los niveles de las señales del puerto de impresora (Centronics) son muy débiles si será preciso dotarlo de

una alimentación auxiliar, si bien al ser el consumo extremadamente reducido ($< 4\text{mA}$) las especificaciones de la fuente son muy básicas ($5\text{V} \pm 5\%$, 5mA). La línea GND (0 V) se utiliza como terminal negativo de alimentación y corresponde además al terminal de menor potencial para las entradas/salidas analógicas, por tanto, deberá procurar un buen diseño en los lazos de masa para evitar errores de medida debido a diferencias de potencial en GND.

(3) Esta frecuencia corresponde a la máxima que los dispositivos internos son capaces de alcanzar, sin embargo, la máxima frecuencia a la que usted podrá muestrear va a depender de múltiples factores. Los más importante son: Software, Sistema Operativo y Velocidad de proceso de su ordenador.

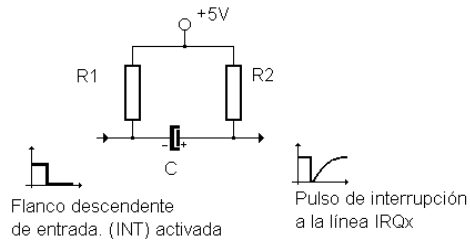


Figura. 11. Circuito para la generación de una interrupción hardware (IRQx) utilizando la salida (INTx) de interrupción de los contadores.

I.4.5 LipSoft-IOT

I.4.5.1 Descripción

Este módulo contiene 1 grupo de 8 líneas de entrada/salida digitales (I/O) y 2 contadores de 6 décadas (0..999999) con posibilidad de programar su alarma. Las líneas digitales admiten señales con niveles lógicos de las familias TTL o CMOS a 5 V.

Los contadores puede ser utilizado para múltiples tareas, las más habituales son la totalización de sucesos discretos y la generación de temporizaciones precisas. Aceptan señales lógicas con niveles TTL o CMOS a 5V. Su frecuencia máxima de conteo es de 1 Mhz.

El módulo está indicado para el control y monitorización de sistemas discretos. Con un programa adecuado sus líneas I/O pueden actuar como las líneas de cualquier autómata programable. Además, los contadores que incorpora, permite gestionar de manera cómoda ciertas señales digitales imposibles de controlar con las entradas I/O simples. Ejemplo de esto podría ser el conteo de los impulso de un codificador incremental, medida precisa de la frecuencia de un lazo V/F etc.

La alarma incluida en el contador es un registro de 6 dígitos que puede ser programado por el usuario. Cuando la totalización (el número de impulsos contados) se hace igual al contenido del registro de alarma, el nivel de la línea INT (interrupción) baja (0 V). Si esta línea se cablea, tal como muestra la figura 11, a la línea de solicitud de interrupción del puerto paralelo (IRQ7 o IRQ5) es posible controlar rápidamente este suceso. Esto permite que el software desarrollado pueda estar realizando otras tareas y solo preocuparse del contador cuando se produce el citado evento. De cualquier modo, la generación de ésta alarma puede habilitarse/deshabilitarse (vía software) en cualquier momento. Las líneas de interrupción (INT0 e INT1) de los contadores y la IRQ están disponibles en el conector de salida de modo que puedan ser cableadas según sus necesidades.

I.4.5.2 Asignación de patillas al conector de salida del módulo LipSoft-IOT

Patilla	Nombre	Descripción.
1	P3	Línea Digital bit 3
3	P2	Línea Digital bit 2
5	P1	Línea Digital bit 1
7	P0	Línea Digital bit 0
9	N/C	No conectada
11	N/C	No conectada
13	INT0	Interr. cont. dirección 0
15	INT1	Interr. cont. dirección 1
17	ECONTO	Ent. pulsos cont. direc.0
19	ECONT1	Ent. pulsos cont. direc. 1
2	GND0	V (Alimentación y señal)
4	P4	Línea Digital bit 4
6	P5	Línea Digital bit 5
8	P6	Línea Digital bit 6
10	P7	Línea Digital bit 7
12	N/C	No conectada
14	CLK	Sincronismo
16	DATA	Datos
18	VCC	Alim. ext. +5V DC
20	IRQ	Solicitud interrupción puerto paralelo

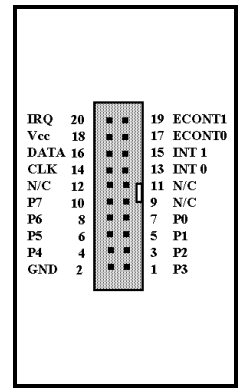


Figura. 12. Conector de salida LipSoft-IOT

I.4.5.3 Características eléctricas y de funcionamiento

Parte de las características que aquí se resumen son intrínsecas a los componentes utilizados en los módulos. Estas características son valores típicos a 25°C .

Líneas Digitales entrada/salida cuasidireccionales (P0...P7)

Ver la misma líneas en el módulo *LipSoft*-ADQ.

Contador (líneas *ECONTx*, *INTx*)

Ver la misma líneas en el módulo *LipSoft*-ADT.

Líneas de control (*CLK*, *DATA*)

Estas líneas no deben ser utilizadas por el usuario (dejadas sin conectar) y sirven para futuras expansiones de los módulos.

Direccionamiento

El módulo tiene fijadas las siguientes direcciones que **no pueden** ser alteradas.

Contador 0	Dirección 0
Contador 1	Dirección 1
I/O Digital	Dirección 7

Línea solicitud de interrupción del puerto impresora (IRQ)

Esta línea que corresponde a -ACK (Reconocimiento = patilla 10 conector impresora SUB-D 25) está conectada a través de la placa paralelo al controlador de interrupciones del PC. Para poder utilizarla es preciso que re programe adecuadamente el controlador y la habilite, a la vez que cree una rutina de interrupción para su atención. El software que incorporan los módulos *LipSoft* incluyen todas estas funciones de modo que la utilización de la interrupción es inmediata. Para que la activación de la alarma en los contadores pueda ser atendida es preciso que una la línea *INTx* (patilla 13, 15) e *IRQ* (patilla 20) del conector de salida a través de un circuito similar al de la figura 11.

Líneas de Alimentación (*VCC* y *GND*)

Ver la misma líneas en el módulo *LipSoft*-ADQ.

1.4.6 *LipSoft*-DIG

1.4.6.1 Descripción

Este módulo incorpora 2 grupos de 8 líneas de entrada/salida digital. Cada línea puede ser programada como entrada o salida. Las líneas admiten niveles lógicos TTL o CMOS a 5V.

El módulo dispone de 2 líneas para programar las 8 direcciones de los grupos I/O digitales posibles. Está indicado para aplicaciones de control que involucren solo señales digitales (todo/nada). Mediante un programa adecuado estas líneas pueden ser programadas para actuar de modo similar a como lo hace un autómata programable.

Cada vez que se produce un cambio en el estado de las líneas digitales se pone a nivel bajo la línea de interrupción del puerto paralelo *IRQ*. Esta línea se encuentra conectada generalmente a la interrupción (*IRQ7* o *IRQ5*). Ver información adicional en la descripción del módulo *LipSoft*-ADQ.

1.4.6.2 Asignación de patillas al conector de salida del módulo *LipSoft*-DIG

Patilla	Nombre	Descripción
1	P3_0	Línea Digital bit 3 grupo 0
3	P2_0	Línea Digital bit 2 grupo 0
5	P1_0	Línea Digital bit 1 grupo 0
7	P0_0	Línea Digital bit 0 grupo 0
9	A2	Línea de dirección
11	A1	Línea de dirección
13	P0_1	Línea Digital bit 0 grupo 1
15	P1_1	Línea Digital bit 1 grupo 1
17	P2_1	Línea Digital bit 2 grupo 1
19	P3_1	Línea Digital bit 3 grupo 1
2	GND	0V (Masa aliment. y señal)
4	P4_0	Línea Digital bit 4 grupo 0
6	P5_0	Línea Digital bit 5 grupo 0
8	P6_0	Línea Digital bit 6 grupo 0
10	P7_0	Línea Digital bit 7 grupo 0
12	P7_1	Línea Digital bit 7 grupo 1
14	P6_1	Línea Digital bit 6 grupo 1
16	P5_1	Línea Digital bit 5 grupo 1
18	VCC	Alimentación ext. +5V DC
20	P4_1	Línea Digital bit 4 grupo 1

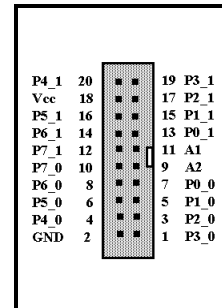


Figura. 13. Conector de salida *LipSoft*-DIG

1.4.6.3 Características eléctricas y de funcionamiento

Parte de las características que aquí se resumen son intrínsecas a los componentes utilizados en los módulos. Estas características son valores típicos a 25° C.

Líneas Digitales entrada/salida (P0_x...P7_x) cuasidireccionales

Ver la misma líneas en el módulo *LipSoft-ADQ*.

Líneas de Direcciones (A1, A2)

Estas dos líneas permiten programar 8 direcciones posibles para los grupos I/O Digitales. El grupo 0 responde a direcciones pares y el grupo 1 a las impares. Para fijar un 1 (nivel alto) es preciso poner la línea correspondiente a una tensión de 5 V mediante una resistencia cuyo valor debe estar comprendido entre 4K7 y 22K. Para fijar un 0 (nivel bajo) basta conectar la línea a 0 V=GND.

La tabla 4 muestra la dirección asignada a los grupos en función de la combinación de las líneas.

A2	A1	Dirección Grupo 0	Dirección Grupo 1
0	0	0	1
0	1	2	3
1	0	4	5
1	1	6	7

Tabla 4. Asignación de direcciones en el módulo *LipSoft-DIG*

Líneas de Alimentación (VCC y GND)

Ver la misma líneas en el módulo *LipSoft-ADQ*.

1.4.7 LipSoft-AD2

1.4.7.1 Descripción

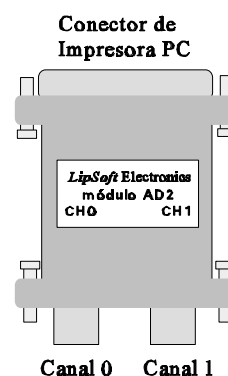
LipSoft AD2 permite adquirir señales analógicas a través de sus dos canales de entrada con una resolución de 8 bits. El módulo se conecta al puerto de impresora de ordenadores personales PC compatibles.

El módulo contiene en su interior toda la electrónica necesaria

para realizar el proceso de conversión analógica a digital y transmisión de los valores convertidos hacia el ordenador. La conexión de las señales a medir se realiza a través de dos conectores BNC situados en la parte posterior.

1.4.7.2 Características eléctricas y de funcionamiento

LipSoft AD2 dispone de dos canales de conversión analógica-digital multiplexados con una resolución de 8 bits. La referencia del convertidor Analógico a Digital interno se ha fijado en 1'25 V y por tanto cada bit equivale a $1250\text{mV}/255 = 4'9\text{mV}$. El rango de entrada es unipolar (solo tensiones positivas con respecto a masa).



Las características eléctricas del circuito que incorporan los módulos son las siguientes a 25°C:

Canales Analógico/Digital: 2	
Resolución:	8 Bits
Rango entrada o salida:	Unipolar de 0 a 1'25 V ⇒ (LSB=4.9 mV)
Error Offset (Adx):	9 mV
Error Linealidad:	±1 1/2 LSB
Error Ganancia:	5%
Factor rechazo modo común:	60 dB
Frecuencia Muestreo Máxima:	10 KHz ⁽¹⁾ .
Modos de trabajo canales Adx:	
Modo 0 = 2 canales simples	AD0, AD1
Modo 3 = 1 canal diferencial	AD0-AD1
<small>En modo diferencial los resultados de la conversión viene representada en complemento a 2.</small>	
Corriente Entrada (Adx):	< 100 nA
Capacidad entrada (Adx):	40 pF
Impedancia de entrada (Adx) :	> 100Kohm
Máxima sobretención admisible:	=10V

1.4.7.3 Manejo del módulo

Para el manejo de *LipSoft AD2* se dispone del conjunto de funciones en las librerías comunes con el resto de los módulos (ADT, AD12, AD8 etc..). De cualquier forma es preciso indicar las siguientes características específicas del módulo *LipSoft AD2*.

- 1º. El parámetro *Direccion* en la funciones de lectura de los canales A/D es fijo y siempre ha de valer 1.
- 2º. Como solo dispone de 2 canales A/D solo son significativos los valores leídos en éstos. Así, si se trata de leer el canal 3 o 4 las funciones devolverán un valor pero que en ningún caso es significativo.

- 3º. A diferencia de los demás módulos con subsistemas de 8 bits (ADT, ADQ, AD8), el módulo AD2 tiene una resolución de 4'9 miliVoltios=LSB.
- 4º. Solo son válidos los valores 0 (simple) y 3 (diferencial) en la programación del parámetro *Modo*.
- 5º. AD2 responde ante la función de búsqueda *Busca* como tipo 1.

Para el manejo desde aplicaciones Windows utilice la librería AD8B.DLL con las restricciones ya indicadas. La aplicación REGISTRO.EXE soporta las características específicas de este módulo.

1.5 Interfases de propósito general (IPG)

En los siguientes apartados se van a describir las características de funcionamiento, conexión y eléctricas de las interfases de propósito general (IPG) para los módulos *LipSoft*.

1.5.1 Introducción

Las IPG son placas de circuito impreso que contiene un conjunto de circuitos, conectores y visualizadores para facilitar la interconexión de los módulos *LipSoft* con el sistema externo a controlar/monitorizar. Todas las líneas I/O o control del módulo *LipSoft* se encuentran disponibles en cada interfase específica. En su proyecto definitivo puede prescindir de estas interfases y desarrollar el sistema de acondicionamiento y tratamiento de las señales hasta los módulos *LipSoft*. De cualquier modo, las IPG pueden servir como etapa previa para el acondicionamiento de las señales.

1.5.2 Características comunes

Aunque cada IPG está indicada para un módulo *LipSoft* distinto, todas comparten una serie de características comunes. Estas son:
 *Regletas de conexión con tornillo para las señales que manejan (analógicas, digitales contadores, interrupciones, alimentación auxiliar etc.).

*Diodos Leds de 3mm para mostrar el estado de las líneas I/O digitales figura 14.

*Diodo Led indicador de alimentación.

*Soporte para realizar divisores de tensión o filtros, figura 16, en las entradas Analógicas/Digitales y de Contadores. Es posible pinchar hasta 4 componentes en cada uno de los zócalos de las entradas.

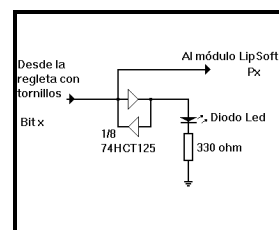


Figura. 14. Esquema de la excitación de los leds de las líneas digitales I/O.

*Protección activa, figura 15, por medio de transistores de las entradas analógicas y de contadores.

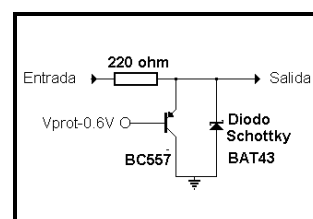


Figura. 15. Protección activa mediante transistor y diodo que incorporan las interfases.

*Regulador de tensión (7805) y filtro incorporado para alimentar la interfase y los módulos *LipSoft* si fuese preciso. La

alimentación de las interfases se procura por medio de un alimentador universal (bajo coste) que al menos suministre (7 V DC y 300 mA). La conexión del alimentador se realiza por medio de un conector tipo pila 9V. Las interfases pueden ser alimentadas por medio de pilas o baterías a 9 Volt.

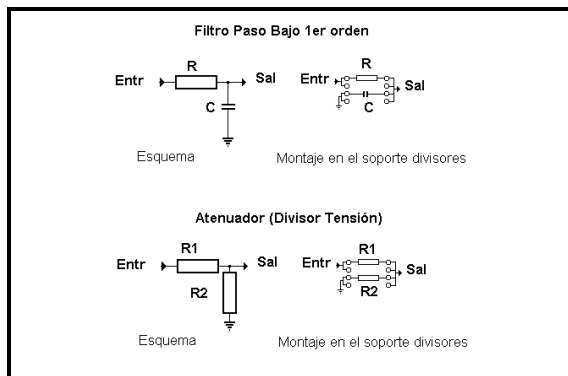


Figura. 16. Ejemplo de realización de filtro y atenuador utilizando el soporte para divisores.

*Disponen de un conector auxiliar para compartir la alimentación. En el caso de que se conecten múltiples interfases solo es preciso un alimentador (de potencia suficiente) conectado a una de las interfases. Un cable y conector apropiado (incluido) repartirá la alimentación al resto de las interfases.

*Selección de la dirección de los módulos por medio de

interruptores DIP.

*Formato 77x122 mm con 4 agujeros para sujeción con tornillos de M3.

1.5.3 Interfase *LipSoft-ADQ*

Esta IPG esta indicada para la conexión de módulos *LipSoft-ADQ*.

Dispone de regletas para la entrada/salida de las señales analógicas y digitales. Los 4 canales analógicos están protegidos por un circuito como muestra la figura 15.

La indicación del estado de las entradas/salidas digitales se realiza mediante 8 Leds excitados con un CI 74HCT245.

El microinterruptor DIP de 3 circuitos permite seleccionar las 8 direcciones posibles (ver tabla 1). Si se encuentra en la posición ON (cerrado) la línea correspondiente (A0, A1, A2) se encuentran unidas a 0V (nivel bajo). En caso contrario (OFF) se encuentran a nivel alto (+5V). Las direcciones correspondientes en función del estado de los interruptores viene reflejado en la tabla siguiente:

A2	A1	A0	Dirección
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

Dirección asignada en función del ajuste de A2, A1, A0.

Como muestra la figura 17, dispone de 4 grupos de soporte (Soporte divisores) para poder insertar elementos de filtraje o adaptación de señal para cada canal Analógico de entrada. Los dos terminales superiores a la izquierda están conectados a terminal con tornillo del canal correspondiente, los otros dos a GND (0V).

Para alimentar esta placa es preciso disponer de un alimentador de CC con conector tipo pila 9V y que al menos suministre 100 mA a 9V. La tensión máxima admisible es de 18V DC y mínima de 7V DC.

Notas importantes:

1) La salida del convertor D/A del módulo *LipSoft-ADQ* se encuentra protegida mediante una resistencia de 1 kΩ, y por tanto, la impedancia (resistiva) de salida será de este valor. Téngalo en cuenta si trata de excitar a sistemas con bajas

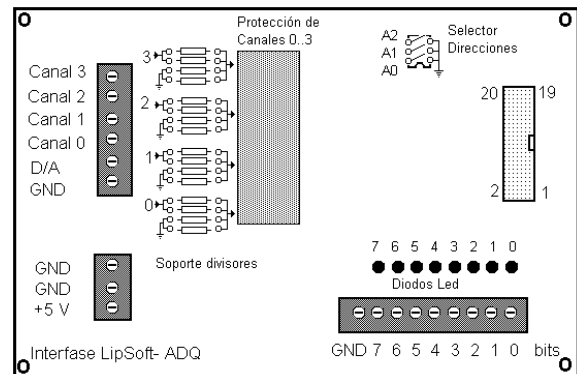


Figura. 17. Aspecto de la tarjeta de Interfase *LipSoft-ADQ*

impedancias de entrada. Una solución optima para evitar cualquier error debido a este fenómeno consiste en desacoplar la fuente de la carga mediante un amplificador operacional en montaje seguidor de tensión.

2) La lectura de una entrada analógica que se encuentre al aire, es decir, sin conectar con ningún sistema que fije un nivel concreto, puede tomar valores arbitrarios y aleatorios. Se recomienda como medida de seguridad que las entradas no utilizadas se conecten a masa (0 Voltios).

1.5.4 Interfase *LipSoft-IOT*

Esta tarjeta esta indicada para la conexión de módulos *LipSoft-IOT*.

Dispone de regletas para la entrada/salida de las señales de entrada de los contadores, interrupciones y digitales. Las 2 entradas de contadores están protegidas por un circuito como muestra la figura 15. La indicación del estado de las entradas/salidas digitales se realiza mediante 8 Led's excitados mediante un 74HCT245.

La figura 18 muestra la disposición de los conectores y elementos más importantes.

Como el módulo *LipSoft-IOT* tiene un direccionamiento fijo, no se dispone de ningún elemento para poder seleccionar las direcciones.

Para alimentar esta placa es preciso disponer de un alimentador de CC con conector tipo pila 9V y que al menos suministre 100 mA a 9V. La tensión máxima admisible es de 18V DC y mínima de 7V DC.

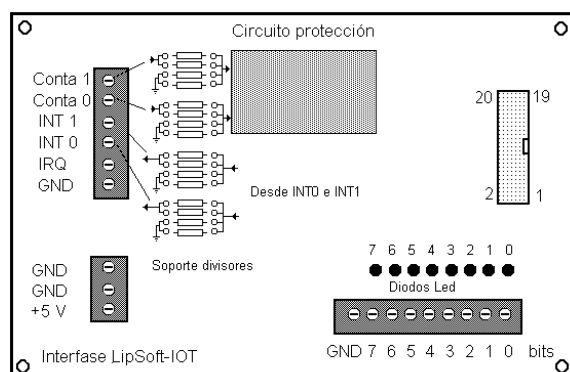


Figura. 18. Aspecto de la tarjeta de Interfase *LipSoft-IOT*

I.5.5 Interfase *LipSoft-ADT*

Esta tarjeta esta indicada para la conexión de los módulos *LipSoft-ADT*.

Dispone de regletas para la entrada/salida de las señales analógicas, digitales y contador. Los 4 canales analógicos y el canal de entrada del contador están protegidos por un circuito como muestra la figura 15.

La indicación del estado de las entradas/salidas digitales se realiza mediante 8 Leds excitados con un CI 74HCT245.

Mediante el puente JP10 situado junto al conector principal se pueden fijar las direcciones a que atiende el módulo *LipSoft-ADT*. Si se encuentra en la posición ON (cerrado) la línea correspondiente (A0) se encuentran unidas a 0V (nivel bajo) y las dirección = 0. En caso contrario (OFF) se encuentran a nivel alto (+5V) y la dirección de los elementos del módulo *LipSoft* será la 1.

A0 = 0	A0 = 1
Dir del A/D 8 bits = 0	Dir del A/D 8 bits = 1
Dir del I/O digital = 0	Dir del I/O digital = 1
Dir del Contador = 0	Dir del Contador = 1

Como se aprecia en la figura 19, dispone de 4 grupos de soporte (Soporte divisores) para poder insertar elementos de filtraje o adaptación de señal para cada canal Analógico de entrada. Además hay un soporte adicional para adaptar la señal a la entrada del contador.

Para alimentar esta placa es preciso disponer de un alimentador de CC con conector tipo pila 9V y que al menos suministre 100 mA a 9V. La tensión máxima admisible es de 18V DC y mínima de 7V DC.

Notas importantes:

1) La salida del convertor D/A del módulo *LipSoft-ADT* se encuentra protegida mediante una resistencia de 1 k Ω , y por tanto, la impedancia (resistiva) de salida será de este valor. Téngalo en cuenta si trata de excitar a sistemas con bajas

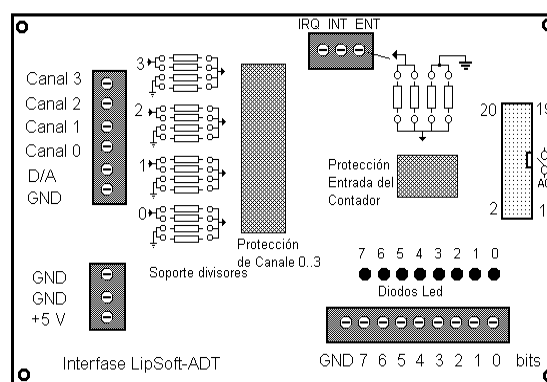


Figura. 19. Aspecto de la tarjeta de Interfase *LipSoft-ADT*

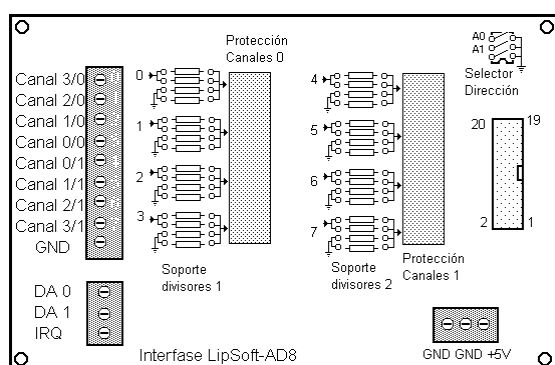
impedancias de entrada. Una solución óptima para evitar cualquier error debido a este fenómeno consiste en desacoplar la fuente de la carga mediante un amplificador operacional en montaje seguidor de tensión.

2) La lectura de una entrada analógica que se encuentre al aire, es decir, sin conectar con ningún sistema que fije un nivel concreto, puede tomar valores arbitrarios y aleatorios. Se recomienda como medida de seguridad que las entradas no utilizadas se conecten a masa (0 Voltios).

I.5.6 Interfase *LipSoft-AD8*

Esta tarjeta esta indicada para la conexión de los módulos *LipSoft-AD8*.

Cuenta con regletas con tornillo para la conexión de los 8 canales analógicos de entrada. Estos canales se encuentran protegidos por un circuito como muestra la figura 15. Además contiene 8 soportes para poder insertar elementos para el filtraje o adaptación de señal (soporte de divisores) ver figura 20.



Mediante el microinterruptor DIP (selector de direcciones) se pueden fijar 4 posibles estados. Si un circuito se encuentra en la posición ON (cerrado) la línea correspondiente (A0, A1) se encuentran unidas a 0V (nivel bajo). En caso contrario (OFF) se encuentran a nivel alto (+5V). La asignación de direcciones viene reflejada en la tabla siguiente.

A2	A1	Dirección Grupo 0	Dirección Grupo 1
0	0	0	1
0	1	2	3
1	0	4	5
1	1	6	7

Asignación de direcciones para el módulo *LipSoft-AD8*.

Para alimentar esta placa es preciso disponer de un alimentador de CC con conector tipo pila 9V y que al menos suministre 100 mA a 9V. La tensión máxima admisible es de 18V DC y mínima de 7V DC.

Notas importantes:

1) La salida de los convertidores D/A del módulo *LipSoft-AD8* se

encuentra protegida mediante una resistencia de 1 k Ω , y por tanto, la impedancia (resistiva) de salida será de este valor. Téngalo en cuenta si trata de excitar a sistemas con bajas impedancias de entrada. Una solución óptima para evitar cualquier error debido a este fenómeno consiste en desacoplar la fuente de la carga mediante un amplificador operacional en montaje seguidor de tensión.

2) La lectura de una entrada analógica que se encuentre al aire, es decir, sin conectar con ningún sistema que fije un nivel concreto, puede tomar valores arbitrarios y aleatorios. Se recomienda como medida de seguridad que las entradas no utilizadas se conecten a masa (0 Voltios).

I.5.7 Interfase *LipSoft-AD12*

Esta tarjeta esta indicada para la conexión de los módulos *LipSoft-AD12*.

Cuenta con regletas con tornillo para la conexión de los 8 canales analógicos de entrada. Estos canales se encuentran protegidos por un circuito como muestra la figura 15. Además contiene 8 soportes para poder insertar elementos para el filtraje o adaptación de señal (soporte de divisores) figura 21.

Mediante el microinterruptor DIP de 6 circuitos (selector de direcciones) se pueden fijar las 3 posibles direcciones. Tal y como se aprecia en la figura 21, para fijar cualquier dirección siempre se deben cerrar (ON) un par de circuitos (consecutivos). La asignación de direcciones en el módulo *LipSoft-AD12* correspondiente viene reflejada en la tabla siguiente.

Dirección asignada	Líneas unidas
0	A0-X0 y A1-X1
1	A0-Y0 y A1-Y1
2	A0-Z0 y A1-Z1

Asignación de direcciones en el módulo *LipSoft-AD12*

Para alimentar esta placa es preciso disponer de un alimentador de CC con conector tipo pila 9V y que al menos suministre 100 mA a 9V. La tensión máxima admisible es de 18V DC y mínima de 7V DC.

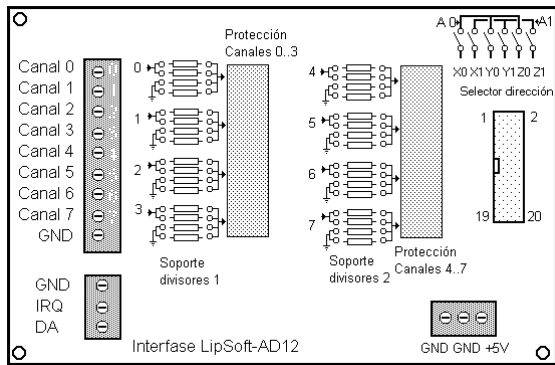


Figura. 21. Aspecto de la tarjeta de Interfase **LipSoft-AD12**

Nota importante:

2) La lectura de una entrada analógica que se encuentre al aire, es decir, sin conectar con ningún sistema que fije un nivel concreto, puede tomar valores arbitrarios y aleatorios. Se recomienda como medida de seguridad que las entradas no utilizadas se conecten a masa (0 Voltios).

1.5.8 Interfase **LipSoft-DIG**

Esta tarjeta está indicada para la conexión de módulos **LipSoft-DIG**.

Dispone de regletas para las 16 entrada/salida digitales divididas en dos grupos de 8. La indicación del estado de las entradas/salidas digitales se realiza mediante Leds excitados con un CI 74HCT245.

Mediante el microinterruptor DIP (selector de direcciones figura 22) se pueden fijar 4 posibles estados. Si un circuito se encuentra en la posición ON (cerrado) la línea correspondiente (A1, A2) se encuentran unidas a 0V (nivel bajo). En caso contrario (OFF) se encuentran a nivel alto (+5V). La asignación de direcciones viene reflejada en la tabla siguiente:

A2	A1	Dirección Grupo 0	Dirección Grupo 1
0	0	0	1
0	1	2	3
1	0	4	5
1	1	6	7

Asignación de direcciones en el módulo **LipSoft-DIG**

Para alimentar esta placa es preciso disponer de un alimentador

de CC con conector tipo pila 9V y que al menos suministre 100 mA a 9V. La tensión máxima admisible es de 18V DC y mínima de 7V DC

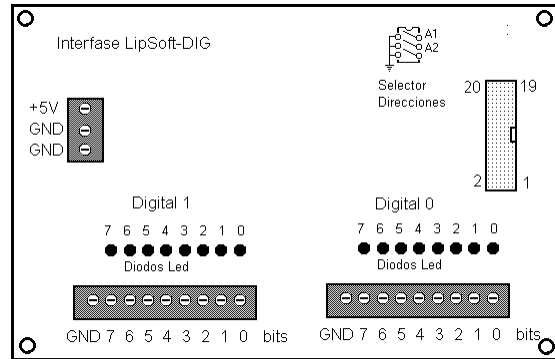


Figura. 22. Aspecto de la tarjeta de Interfase **LipSoft-DIG**.

1.6 Interfases de control de POTENCIA (ICP)

En los siguientes apartados se describen las características de funcionamiento, conexión y eléctricas de las interfases digitales para control de potencia mediante módulos *LipSoft* ADQ, ADT, IOT o DIG.

1.6.1 Características comunes

Existen 4 tipos de interfases específicamente diseñadas para su conexión con cualquiera de los módulos *LipSoft* que contienen subsistemas I/O digitales (ADQ, ADT, IOT o DIG). Las características que comparten son:

- Conexión de las entradas/salidas mediante terminales con tornillo.
- Aislamiento (optico/magnético) entre las entradas/salidas y las líneas de control.
- Regulador de tensión (7805) y filtro para la alimentación.
- Indicador de alimentación ON mediante LED verde.
- Monitorización del estado de las entradas/salidas mediante LEDs rojos.
- Conector de alimentación para acoplar en cascada la alimentación de múltiples tarjetas.
- Clemas con tornillos para alimentación auxiliar de 5 V que puede ser utilizada para alimentar elementos auxiliares ($I_{max} < 200$ mA).
- Selector de direcciones (pares impares mediante selección de zócalo) y dirección base mediante microinterruptor DIP.
- Formato común de 118 x 130 mm con 4 taladros para montaje y sujeción mediante tornillos M3.
- Alimentación: 7.. 12 V D, Intensidad > 300 mA (1A para tarjeta salida Relés)

Direccionamiento de los módulos LipSoft

Dependiendo del módulo *LipSoft* que controlará la interfase, se ha de ajustar la posición de los interruptores A0, A1, A2 del selector de direcciones de las tarjetas según las siguientes tablas:

Módulos LipSoft-DIG

El interruptor A2 ha de estar OFF para los módulos *LipSoft*-DIG

DIG	A1	A0	
0	ON	ON	(ON cerrado= hacia arriba)
2	ON	OFF	
4	OFF	ON	
6	OFF	OFF	

Situando la red de resistencias en el zócalo correspondiente (PARES o IMPARES) se podrá seleccionar la dirección final (0, 1, 2, 3, 4 .. 7)

Módulos LipSoft-ADQ

ADQ	A2	A1	A0
0	ON	ON	ON
1	ON	ON	OFF
2	ON	OFF	ON
3	ON	OFF	OFF
4	OFF	ON	ON
5	OFF	ON	OFF
6	OFF	OFF	ON
7	OFF	OFF	OFF

La red de resistencias se ha de colocar sobre el zócalo de DIRECC. PARES.

Módulos LipSoft-ADT

ADT	A2	A1	A0
0	OFF	OFF	ON
1	OFF	OFF	OFF

Cualquier otra combinación es inválida

Módulos LipSoft-IOT

En estos módulos la dirección de I/O viene fijada a 7, y es independiente de la selección de los interruptores A0, A1, A2. Es preciso que la red de resistencias se sitúe sobre el zócalo de DIRECC. PARES.

1.6.2 Interfase 8 Entradas Optoaisladas

Esta interfase junto con un módulo *LipSoft*-DIG, IOT, ADT, o ADQ le permitirá la detección del estado digital (CERRADO / ABIERTO, ENCENDIDO / APAGADO) de hasta 8 líneas de control.

Mediante una selección adecuada de las resistencias limitadoras es posible la detección de tensiones desde 2 Voltios hasta 380 V AC/DC. La tarjeta viene preparada para trabajar con tensiones entre 24 y 48 V AC/DC.

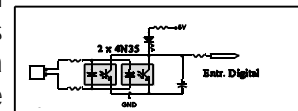


Figura. 23 Entrada AC/DC. La tarjeta viene optoaislada

Como la transmisión de la señal entre las entradas y las líneas digitales se realiza mediante un acoplamiento óptico (4N32) se

consigue un nivel de aislamiento muy elevado (unos 3000 V) lo que garantiza una alta seguridad en su funcionamiento.

El estado de las entradas se monitoriza mediante unos diodos led de color rojo.

Funcionamiento

Las señales de tensión o corriente presentes en cada una de las entradas de la tarjeta (Jpx), hace que circule una corriente por los diodos emisores de los circuitos integrados optoaisladores tipo 4N35, que es función del valor de la resistencia limitadora R, figura superior. La luz emitida por éstos diodos hace que circule una corriente proporcional por el colector del transistor detector. El condensador C integra la señal en el colector de modo que transforma la señal rectificada(en caso de entrada AC) en una señal continua. Además permite eliminar posibles rebotes (falsos contactos) en las entradas.

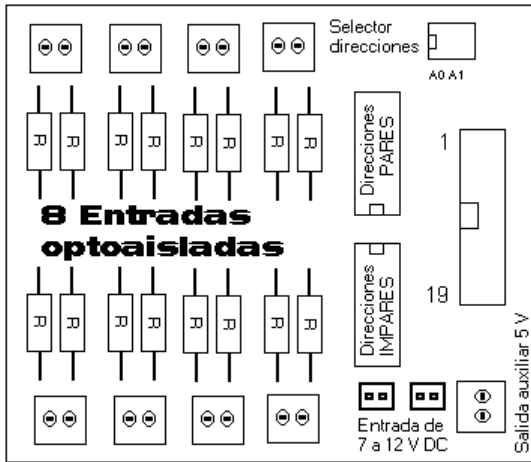


Figura. 24 Aspecto de la tarjeta 8 Entradas

Para poder detectar el estado (ON/OFF) de las líneas es preciso que programe el subsistema I/O del módulo *LipSoft*-DIG, ADT, o ADQ como entradas, es decir a nivel alto (escdig 255), de este modo si el Bit x, figura superior, está a nivel bajo (entrada activada) al realizar una lectura (leedig) leeremos el valor correcto.

Mediante la selección adecuada del valor de R (resistencia limitadora) se podrán seleccionar un valor distinto para la entrada. Los valores de R se calculan según la expresión:

$$R \text{ (Kohm)} = (\text{Ventrada máxima} / 24)$$

La potencia que disipará la resistencia vale :

$$P(\text{Wattios}) = (\text{Ventrada})^2 / R(\text{ohm})$$

Para seleccionar la dirección de atención del módulo *LipSoft* ver

tablas en el apartado I.6.1.

Características Eléctricas

Número de Entradas: 8

Rango Entrada: Sobre pedido (Defecto 24..48 V AC/DC).

I.6.3 Interfase 8 Salidas a Relé

Esta tarjeta junto con un módulo *LipSoft*-DIG, IOT, ADT, o ADQ le permitirá realizar el control (todo/nada) de 8 dispositivos. La tarjeta contiene 8 relés

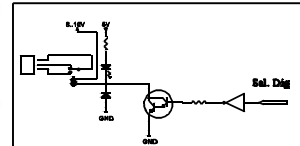


Figura. 25 Salida relé

electromecánicos, con un circuito conmutado, que puede controlar cargas de hasta 250 V y 6 A. A su vez se consigue un aislamiento galvánico entre los circuitos controlados y los de control de unos 2000 Voltios. El estado de activación de los relés se monitoriza mediante unos diodos led de color rojo.

Funcionamiento

La señal proveniente de las líneas I/O de los módulos *LipSoft*-DIG, ADT, o ADQ, es amplificada e invertida por medio de un buffer tipo 74LS240. La salida de este buffer ataca la base de un transistor de potencia tipo TIP 110. Al circular corriente por su base hace que circule por su colector lo que activa la bobina del relé correspondiente a la vez que el diodo led. Un diodo tipo 1N4004, protege al transistor contra las sobretensiones inversa producidas al apagar la corriente de la bobina del relé.

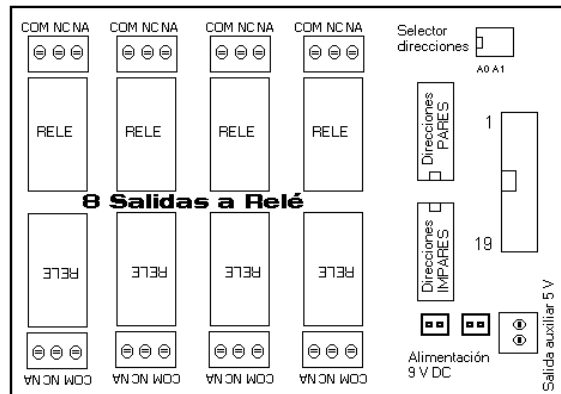


Figura. 26 Aspecto de la tarjeta de Relés

Para **activar** una salida es preciso **escribir un 0** en el bit correspondiente del byte digital escrito en el módulo *LipSoft* mediante la función (escdig). Esta inversión en la lógica de activación (0=Activado) protege durante el encendido de los módulos *LipSoft* que ponen todas sus líneas a 1 y por tanto activarían todos los relés. Para direccionar la tarjeta ver apartado

I.6.1.

Características Eléctricas

Número de Salidas: 8 (1 circuito conmutado. COM = Común, NC= Normalmente Cerrado, NA= Normalmente Abierto)

Potencia controlable: < 1500 VA (250V 6A)

I.6.4 Interfase 8 Salidas Opto - Transistor

Esta tarjeta junto con un módulo *LipSoft*-DIG, ADT, IOT o ADO le permitirá controlar la activación/desactivación de un transistor de potencia PNP. Este transistor puede ser conectado para controlar una carga en modo todo/nada.

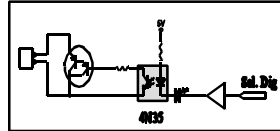


Figura. 27 Salida opto-transistor

La transmisión de la señal entre las líneas digitales y los transistores se realiza mediante un acoplamiento óptico (4N35) para conseguir un nivel de aislamiento galvánico muy elevado (unos 3000 V) lo que garantiza una alta seguridad en su funcionamiento.

El estado de las salidas se monitoriza mediante unos diodos led de color rojo.

Funcionamiento

Las señales I/O de salida del módulo *LipSoft* atacan las entradas del excitador 74LS244. La salida hacen circular corriente por el diodo led del optoacoplador que a su vez excita la base del optotransistor acoplado ópticamente con él. La conducción de este transistor hace que circule (si se encuentra polarizado) corriente por la base del transistor de potencia PNP lo que produce la saturación de éste. Cuando una línea se activa (salida digital=0) el diodo led correspondiente se enciende. En las bornas de salida el terminal "C" = Colector del transistor, y el "E" corresponde al emisor del transistor.

Si bien la placa admite otro tipo de transistores Darlington PNP en cápsula TO220 (bajo pedido), por defecto se montan los TIP126.

Para seleccionar la dirección de atención del módulo *LipSoft* ver tablas en el apartado I.6.1.

Características eléctricas de la tarjeta con Transistor TIP126

Condiciones a 25 °C capsula.

Corriente continua colector: 5A

Corriente de pico máxima = 8A

Tensión de ruptura Colector-Emisor=80V

Tensión de saturación Colector-Emisor=2V (Ic=3A)

I.6.5 Interfase 8 Salidas Opto - Triac

Esta tarjeta junto con un módulo *LipSoft*-DIG, ADT, IOT o ADO le permitirá controlar la activación/desactivación de un Triac de potencia. Mediante este triac podrá controlar todo/nada cargas que funciones con corriente alterna (motores, resistencias etc).

La transmisión de la señal entre las líneas digitales y el triac de potencia se realiza mediante un

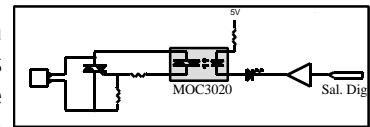


Figura. 28 Salida Opto-Triac

acoplamiento ópto-triac (tipo MOC3020) para conseguir un nivel de aislamiento galvánico muy elevado (> 3000 V) lo que garantiza una alta seguridad en su funcionamiento.

El estado de las salidas se monitoriza mediante unos diodos led de color rojo.

Funcionamiento

Las señales I/O de salida del módulo *LipSoft* atacan las entradas del driver 74LS244, que hace circular corriente por el diodo led monitor y por diodo del optotriac MOC3020, esto hace que el triac entre en conducción y a su vez active la puerta del triac de potencia (por defecto TIC216). Cuando la conducción se ha iniciado en el triac solo es posible extinguirla reduciendo la corriente por debajo del valor de mantenimiento, algo que sucede automáticamente cada semiciclo con alimentación (AC), pero no con alimentación continua (DC).

Una línea está **activada** cuando se **escribe un 0** en el bit correspondiente de la salida digital (escdig). La placa admite diversos tipos de Triac en cápsula TO220 (bajo pedido), por defecto se instalan los tipos TIC216.

Para seleccionar la dirección de atención del módulo *LipSoft* ver tablas en el apartado I.6.1.

Características Eléctricas de la tarjeta con Triac TIP216

Condiciones a 25 °C capsula.

Máxima tensión repetitiva en estado apagado (capacidad de bloqueo)=600V

Corriente continua máxima =6A

Corriente de pico no repetitivo senoidal de duración 20ms= 60A

Nota: El terminal MT1 se encuentra unido a la cápsula en este triac.

Manual de Usuario

Capítulo II

Soporte Software bajo MSDOS

II.1 Introducción

Para conseguir la comunicación entre su programa de aplicación (bajo DOS) y los módulos *LipSoft* existen dos alternativas:

a) Utilizar el controlador de dispositivos instalable LIPSOFT.SYS. Este controlador permite el acceso a las funciones de los módulos a través de un tratamiento similar a cualquier otro dispositivo de Entrada/Salida (consola, pantalla, fichero etc.). Su lenguaje de programación puede manejar los módulos, a través de LIPSOFT.SYS, como cualquier archivo. Las funciones básicas serán : abrir, leer, escribir y cerrar (cada una toma distintas formas en función del lenguaje de programación). Establecido el método de comunicación solo se precisa que envíe los comandos a los que atienden los módulos. Estos comandos serán los que determinarán la operación que han de ejecutar el módulo (convertir una señal analógica en digital, poner a un determinado nivel lógico las líneas I/O, activar el conteo de los contadores etc.).

b) Enlazar su aplicación con la librerías (ficheros .OBJ) que contienen todas las funciones necesarias para manejar los módulos. En este caso se disponen de tres librerías adaptadas a los tres lenguajes de programación mas extendidos BASIC, PASCAL y C.

La decisión de utilizar una u otra técnica queda a su elección, sin embargo, es preciso tener en cuenta los siguientes puntos:

1. Aunque la utilización del controlador de dispositivos resulta más simple, se corre el riesgo de la compatibilidad con cierto entornos operativos (controladores de memoria, redes locales etc.). Además es preciso que se modifique el fichero de configuración CONFIG.SYS para que se cargue durante el proceso de arranque.

2. La velocidad de acceso a los módulos vía controlador es algo más reducida que a través de llamadas directas. Esto es debido a que el sistema operativo ha de realizar el traspaso de parámetros por medio de buffers intermedios de comunicación y el controlador ha de interpretar cada comando para realizar la operación adecuada.

3. Una de las ventajas de utilizar el controlador es que cualquier lenguaje de programación podrá manejar los módulos, ya que todos los lenguajes disponen de funciones de E/S (Clipper, Cobol etc.).

II.2 Controlador de dispositivos LIPSOFT.SYS

II.2.1 Introducción

Para la gestión de los módulos LipSoft desde DOS dispone de un controlador de dispositivo llamado LIPSOFT.SYS que permite la lectura de las entradas del sistema, la escritura en las salidas y el control de los contadores sin preocuparse del funcionamiento a nivel hardware de los mismos.

Mediante el controlador es posible el muestreo y generación de señales a velocidad constante gracias al control del tiempo que realiza, muestreando hasta 256 datos de un canal de entrada a una frecuencia máxima de 1 KHz y/o generando señales periódicas con hasta 256 niveles diferentes a la misma frecuencia.

También es posible atender interrupciones generadas externamente gracias a las entradas de interrupción disponibles.

En los siguientes apartados se indica como manejar los módulos a través del controlador LIPSOFT.SYS, mostrándose un ejemplo de las declaraciones a realizar si programa en BASIC, PASCAL o C.

II.2.2 Instalación

Como todos los controladores de dispositivo, LIPSOFT.SYS debe ser cargado en memoria al arrancar el ordenador. Esta función la realiza el sistema operativo si se incluye en el fichero *CONFIG.SYS* la siguiente línea:

```
DEVICE = [path]LIPSOFT.SYS
```

Donde path indica la unidad de disco y directorio donde se encuentra el fichero LIPSOFT.SYS y se puede omitir si este se encuentra en el directorio raíz de la unidad por defecto (habitualmente C:).

Al arrancar el ordenador, el controlador se cargará en memoria, apareciendo en pantalla el mensaje correspondientes donde se indicará la versión del mismo (vX.X).

A partir de este momento todas las funciones del controlador estarán disponibles; debiendo referirnos al él como si de un archivo o fichero de disco se tratara pero con el nombre `_LIP`.

II.2.3 Acceso al controlador

El acceso al controlador desde cualquier programa es sencillo ya que todos los lenguajes de alto nivel poseen funciones para la gestión de ficheros de distintos tipos (ficheros secuenciales, aleatorios, binarios, etc.) En nuestro caso, el modo de acceso al controlador se realizará como si de un fichero binario se tratara.

Al controlador se puede acceder bien para leer o escribir datos desde o hacia el sistema de adquisición, (es decir, leer o escribir a las diferentes entradas y salidas de los módulos) o bien para enviar información útil al propio controlador (comandos de programación de las diferentes funciones).

El tipo de información que se envía lo detecta el controlador gracias al uso de separadores que se añaden al final de cada comando o cada dato. El símbolo separador es la barra `|` (código ASCII 124). Insertando un separador entre dos comandos, entre dos datos o entre un comando y un dato, el controlador entenderá cada información adecuadamente. Veamos un ejemplo:

Supongamos que enviamos al controlador la cadena siguiente:

```
INICIALIZA | PUERTO 0 | TIPO 0 | DIRECCION 0 | 255 | 0 |
```

Aunque no se conozca el significado de los comandos, podemos ver que esta cadena la componen 4 comandos (INICIALIZA, PUERTO 0, TIPO 0 y DIRECCION 0), y dos datos (255 y 0) que serán recibidos secuencialmente por el controlador y ejecutados cada vez que se encuentre el separador (`|`).

Si la cadena que se envía al controlador es de longitud superior a 32 caracteres, se producirá un error de dispositivo.

La información que se lee del controlador es siempre del mismo tipo: una cadena de 6 caracteres justificada con espacios a la izquierda. Cada vez que se acceda a leer información del controlador deberá solicitarse, por tanto, un dato de este tipo, es decir, 6 bytes de información. La información devuelta por el controlador será la leída desde el sistema de adquisición, es decir, el valor de las entradas del sistema (analógicas o digitales), o el de los contadores, según la selección en ese momento.

II.2.4 Apertura y Cierre del controlador

Antes de acceder al controlador es necesario **abrir** el canal de comunicación adecuadamente. Para abrir el controlador, es decir, iniciar la comunicación, se procederá como si de un fichero se tratara, incluyendo en el programa la instrucción correspondiente de apertura de fichero en el modo de acceso binario. A continuación se muestra un ejemplo en los lenguajes C y BASIC:

C	<code>int controlador; controlador=open("_LIP",O_RDWR);</code>
Pascal	<code>VAR controlador:FILE; BEGIN ASSIGN (controlador,'_LIP'); RESET (controlador,1); ... END.</code>
BASIC	<code>DIM controlador AS INTEGER controlador = 1 OPEN "_LIP" FOR BINARY AS #controlador</code>

Tabla 1. Declaraciones para abrir el controlador

Lógicamente, al finalizar el acceso al controlador, será necesario **cerrar** el canal de comunicación usando las instrucciones:

C	<code>close(controlador);</code>
Pascal	<code>CLOSE (controlador);</code>
BASIC	<code>CLOSE #controlador</code>

Tabla 2. Declaraciones para cerrar el controlador

II.2.5 Escritura y Lectura del controlador

La escritura de información en el controlador tiene por objeto programar las distintas funciones que éste puede realizar. Por ejemplo: indicarle que canal analógico se desea muestrear, que valor se quiere programar en un contador etc.. De cualquier modo la escritura sobre el controlador ha de ajustarse a un formato bien definido:

Cadena_Comando_o_Dato + "`|`"

Es decir, cualquier comando o dato se le debe de pasar como una cadena de caracteres ASCII en mayúsculas siendo el último carácter el código 124= `|`. Esto es necesario para que el controlador reconozca el final de un comando y pueda procesarlo adecuadamente. Cuando deba enviar un valor numérico como dato usted deberá realizar la conversión conveniente a cadena de

caracteres y concatenar ésta con el resto del comando.

La lectura del controlador nos devolverá (en función de la operación previamente solicitada) un valor numérico (en forma de cadena de 6 caracteres justificados a la izquierda) correspondiente a la información requerida, por ejemplo la cuenta actual de un contador, el estado de las entradas digitales etc. El controlador siempre devuelve 6 bytes que deberán convertirse a un valor numérico mediante instrucciones específicas de su lenguaje. Dependiendo de la operación solicitada deberá interpretar adecuadamente su significado.

Las instrucciones para escribir un dato (en el ejemplo un dato y un comando) al sistema de adquisición es la siguiente:

C	<pre>char cadena[32],sdato[10]; long int dato; strcpy(cadena, "COMANDO "); ltoa(dato,sdato,10); strcat(sdato," "); strcat(cadena,sdato); write(controlador,cadena,strlen(cadena));</pre>
Pascal	<pre>VAR cadena:STRING; dato:LONGINT; ... cadena = "COMANDO " + STR(dato) + ' '; BLOCKWRITE (controlador, cadena, LENGTH(cadena)+1); ...</pre>
BASIC	<pre>DIM cadena AS STRING DIM dato AS LONG cadena = "COMANDO " + STR\$(dato) + " " PUT #controlador, 1, cadena</pre>

Tabla 3. Escritura de un comando y un dato COMANDO|DATO|

La instrucción para la lectura desde el controlador:

C	<pre>char Cadena[]="*****"; long int dato; read(controlador, Cadena, 6); dato=atol(Cadena);</pre>
---	---

Pascal	<pre>Cadena:array [1..6] of CHAR; dato:LONGINT i : INTEGER; BLOCKREAD (controlador, Cadena,6); val(Cadena, dato,i);</pre>
BASIC	<pre>Dim Cadena as String *6 Dim dato as long GET #controlador, 1, Cadena dato=VAL (Cadena) ...</pre>

Tabla 4. Lectura del controlador en la variable entera larga dato

II.2.6 Programación del controlador

Como ya se indicó, desde el controlador se pueden manejar todos los módulos LipSoft. Si bien las características de los módulos son distintas, éstas pueden agruparse en 5 bloques funcionales: A/D 8bits, A/D 12bits, E/S digital, Contadores, IRQ

Módulo LipSoft	ADQ	ADT	AD8	DIG	IOT	AD12
A/D 8bits	1	1	2	-	-	-
A/D 12bits	-	-	-	-	-	1
E/S digital	1	1	-	2	1	-
IRQ	no	si	si	no	si	si
Contador	-	1	-	-	2	-

En la tabla superior se resume el número de elementos de cada tipo que contiene cada módulo LipSoft.

Donde:

A/D 8bits: corresponde a 1 grupo de 4 canales de conversión A/D y 1 canal de conversión D/A de 8 bits

A/D 12bits: corresponde a 1 grupo de 8 canales de conversión A/D y 1 canal de conversión D/A de 12 bits

E/S digital: grupo de 8 líneas entrada/salida digital cuasibidireccionales.

IRQ: indica si dispone o no en su conector de salida de línea para la solicitud de IRQ 7 o IRQ 5

Contador: contador de 6 décadas programable.

Para informar al controlador el elemento al que se trata de acceder es preciso enviarle un identificador (dato) con el TIPO. La asignación de tipos es la que muestra la tabla 5. Además, como ya se indicó en el capítulo 1, en un mismo puerto pueden conectarse varios módulos de un mismo tipo. Para indicarle al elemento concreto a acceder es preciso direccionarlos (ver documentación técnica de los módulos) e indicar la dirección asignada al controlador cuando se desee acceder a él.

ELEMENTO	TIP O	DIREC.	CANALES Entrada	MODOS
I/O digital	0	0..7	--	--
A/D 8 bits	1	0..7	0..4	0..3*
A/D12 bits	2** , 4	0..2	0..7	0..1*
Contador	3	0..1	--	--

Tabla 5. Asignación posible de tipo, direcciones, canales y modos

* Aplicable solo al funcionamiento de las entradas A/D.

** Para la versión hardware antigua del módulo LipSoft AD12 adquiridas antes del 1996.

Los canales de entrada de conversión analógica a digital, admiten diferentes modos de funcionamiento. La indicación al controlador sobre en que modo se desea que funcione se realiza mediante el dato MODO. Los modos posibles se muestran en la tabla siguiente:

TIPO	MODO 0	MODO 1	MODO 2	MODO 3
1 A/D 8bits	4 canales simples	3 canales diferenc.	2 simples y 1 diferenc.	2 canales diferenc.
2 A/D12bits **	8 canales simples	4 canales diferenciales	--	--
4 A/D 12bits	8 canales simples	4 canales diferenciales	--	--

Además de los parámetros indicados, será preciso informar al controlador la dirección física donde se encuentra el puerto paralelo (Centronics) donde se encuentran los módulos conectados. Esto se realiza enviando el comando PUERTO.

NOTA: Todos los comandos que se muestran a continuación deben ser introducidos en mayúsculas, respetando las separaciones. Esto es así debido a que el controlador no realiza comprobaciones previas, salvo en algunos casos críticos, para

aumentar la velocidad de ejecución. Si los valores pasados en los comandos exceden los rangos permitidos, según el tipo de comando se generará un error de dispositivo o bien se ajustarán a un valor permitido, pero en este caso, el valor final puede ser desconocido al usuario por lo que se recomienda no introducir valores no permitidos.

La siguiente tabla resume el conjunto de comando para el acceso a los módulos LipSoft.

COMANDO	DESCRIPCIÓN
INICIALIZA	Este comando restaura las variables internas del controlador. Se debe usar siempre tras la apertura del mismo y antes de su cierre así como si se observará una pérdida del control (generación de algún mensaje de error). Este comando no modifica los parámetros de puerto, canal, modo, etc. sólo ciertas variables de control internas no accesibles por el usuario. Este comando también para la generación y muestreo automáticos y restablece la hora del sistema que en muestreos muy rápidos puede alterarse.
PUERTO n	Especifica el puerto paralelo donde se haya conectado el sistema de adquisición. El valor de n puede ser 0 (dirección 378h), 1 (dirección 278h) o 2 (dirección 3BCh).
TIPO n	Especifica el tipo de elemento seleccionado. Su valor puede ser de 0 a 4. Por defecto es 0.
DIRECCION n	Especifica la dirección asignada al elemento que se desea acceder. Por defecto es 0.
CANAL n	Especifica la entrada analógica a la que se hará referencia. Por defecto es 0.
MODO n	Especifica el modo de funcionamiento de los conversores A/D.

Las instrucciones para mandar un comando al controlador son las siguientes:

C	<pre>char cadena[32]; strcpy(cadena, "COMANDO "); write(controlador, cadena, strlen(cadena));</pre>
---	---

Pascal	VAR cadena:STRING; ... cadena:= 'COMANDO '; BLOCKWRITE (controlador, cadena, LENGH(cadena)+1); ...
BASIC	DIM cadena AS STRING cadena = "COMANDO " PUT #controlador, 1, cadena

II.2.7 Adquisición de datos analógicos de modo automático

Para automatizar el proceso de adquisición de datos, es decir, que el sistema adquiera un número de muestras de un determinado canal con una cadencia definida, el controlador dispone de un conjunto de comandos que permiten, por una parte, el muestreo a velocidad constante de una entrada analógica y, por otra, la generación de una forma de onda periódica por una salida analógica (ambos, muestreo y generación puede estar activos simultáneamente). La entrada y salida a la que se hace referencia es la determinada según los comandos de selección (pero siempre deben pertenecer a un mismo dispositivo, es decir, a un módulo de e/s analógica del mismo tipo, 8 o 12 bits, y dirección).

Para la generación de una señal periódica deberemos almacenar los distintos niveles que conforman la onda (hasta 256) en una matriz o array de números enteros de la longitud adecuada e indicar al controlador la posición en memoria (dada por un **segmento** y un **desplazamiento**) de esa matriz así como el número de elementos que la componen; posteriormente se establece el período de generación, es decir, el tiempo entre la salida de dos valores consecutivos y por último se activa y desactiva la generación en los momentos adecuados.

Para el muestreo se realiza un proceso similar: se crea la matriz de enteros donde se almacenarán las muestras, se le indica su posición de memoria y el número de muestras al controlador; se indica el período de muestreo (este debe coincidir con el de generación en el caso de que estén activos ambos a la vez) y por último se activa el muestreo. Al finalizar el muestreo el controlador puede llamar a una rutina del usuario cuya dirección debe ser indicada al controlador. También es posible desactivar el muestreo en cualquier momento.

Los comandos que permiten todas estas acciones son los siguientes:

COMANDO	DESCRIPCIÓN
PERIODO n	Especifica el período de muestreo/generación. n es el tiempo expresado en milisegundos (de 1 a 65536). Su valor por defecto es 1 ms.
DIR_MUEST s d	Especifica la dirección de la matriz de muestreo. s indica el segmento y d el desplazamiento, ambos en decimal.
ACT_MUEST n	Activa el muestreo. n indica el número de muestras (de 1 a 256).
DES_MUEST	Desactiva el muestreo.
DIR_RUT_M s d	Establece la dirección de la rutina de final de muestreo. s indica el segmento y d el desplazamiento, ambos en decimal. Si su valor es 0, no habrá llamada al final del muestreo.
DIR_GEN s d	Especifica la dirección de la matriz de generación. s indica el segmento y d el desplazamiento, ambos en decimal.
ACT_GEN n	Activa la generación. n indica el número de datos de cada período de la señal a generar (de 1 a 256).
DES_GEN	Desactiva la generación.

Estando activos el muestreo y/o la generación automáticos, no es posible acceder a ningún dispositivo ni en lectura ni en escritura, generándose un error de ocupado en el caso de intentarlo. Tampoco es posible modificar los parámetros PUERTO, TIPO, DIRECCION, CANAL y MODO.

Durante el muestreo y generación, es posible que el ordenador se ralentice. Además, el período real de muestreo puede variar ligeramente del especificado según la velocidad del ordenador. También es posible que se atrase el reloj de tiempo real del sistema pudiendo restablecerse la hora real mediante el comando INIC del controlador o al cerrar el mismo.

En el disco de utilidades se incluyen programas de ejemplo en C, BASIC Y PASCAL de los distintos comandos y modos de funcionamiento donde se puede ver la forma de obtener las direcciones necesarias para el muestreo, generación y activación de las interrupciones en cada lenguaje.

II.2.8. Control de los contadores

Como se indica en el capítulo I, los contadores que incorporan los módulos ADT, e IOT pueden ser programados con una cuenta inicial y con un valor determinado en su registro de alarma, de modo que cuando el número de impulsos contados sea igual al valor de la alarma se produzca un cambio (nivel bajo) en la línea de interrupción. Este cambio, junto con el circuito que se muestra en las características técnicas del módulo, pueden producir una interrupción hardware (IRQ).

Para el manejo de los contadores se dispone de un conjunto de comandos que permiten un control sencillo de todas sus posibilidades. La tabla siguiente refleja el conjunto de comandos disponibles.

COMANDO	DESCRIPCIÓN
ACT_CONT	Activa el contador permitiendo la cuenta de impulsos.
DES_CONT	Desactiva el contador inhibiendo la cuenta de impulsos.
ACT_ALARMA n	Activa la alarma del contador. n es un valor de 0 a 999999 indicando el valor de la cuenta en el que será activada la salida del contador y pedida la interrupción.
DES_ALARMA	Desactiva la alarma del contador.

Nota: La bajada en la línea de interrupción (INT) solo se produce cuando la igualdad de la cuenta y alarma se consigue a través de la llegada de un impulso. La escritura en el registro de cuenta de un valor igual al de alarma no produce interrupción.

II.2.9 Gestión de la interrupción hardware (IRQ)

En la documentación técnica de los módulos se indicó que algunos de ellos disponen en su conector de salida de la línea de reconocimiento (-ACK) del puerto Centronics. Esta línea esta cableada hacia el controlador de interrupciones del ordenador, asignándole la IRQ7 o IRQ5 (dependiendo el tipo de placa paralelo). Si se habilita la interrupción es posible conseguir un control muy eficiente del trabajo de los módulos + ordenador. Por ejemplo, si programamos en la rutina de atención de esta interrupción el comienzo de un proceso de adquisición de datos, esta línea actuará como un control de disparo. Su programa podrá estar realizando cualquier otra misión y automáticamente entrará en funcionamiento el proceso de adquisición sin que tenga que estar constantemente comprobando el estado de

determinada línea.

El controlador dispone de comandos para habilitar/deshabilitar la interrupción. Es preciso que le indique donde se encuentra la rutina que ha de ejecutarse cuando se reciba la solicitud de interrupción. La determinación de la localización de esta rutina de atención depende del lenguaje de programación utilizado, revise los ejemplos en el disco de utilidades para ver como hacerlo.

COMANDO	DESCRIPCIÓN
ACT_INT n	Activa la interrupción hardware. n puede ser 5 o 7 según el puerto utilizado, pero normalmente es la 7.
DES_INT	Desactiva la interrupción hardware.
DIR_RUT_ l s d	Establece la dirección de la rutina de interrupción hardware. s indica el segmento y d el desplazamiento, ambos en decimal. Si su valor es 0, no habrá llamada al producirse una interrupción.

II.2.10 Acceso al controlador desde ensamblador

Para poder acceder al controlador desde lenguaje ensamblador podemos utilizar las funciones del DOS que se describen a continuación, todas ellas a través de la interrupción 21H y con el código de función adecuado en el registro AH:

Acción	Función del DOS
ABRIR el controlador	3D h
CERRAR el controlador	3E h
LEER del controlador	3F h
ESCRIBIR en el controlador	40 h

Para el adecuado manejo de estas funciones será necesario recurrir a manuales avanzados de DOS, escapando de este manual la descripción exhaustiva de las mismas.

II.3 Librerías de enlace para DOS

II.3.1 Introducción

Una técnica alternativa para que su programa pueda establecer comunicación con los módulos LipSoft, consiste en realizar llamadas a las funciones incluidas en librerías al efecto. Estas librerías, que se suministran junto con los módulos, le permitirá conseguir unas prestaciones de los módulos similares a si utilizase el controlador de dispositivos.

Una ventaja adicional de utilizar ésta técnica es la mejora en la velocidad de comunicación con los módulos, ya que se evita el paso intermedio de gestión DOS propia del controlador.

Esta técnica sin embargo, presenta ciertos inconvenientes. Uno de ellos radica en que para cada lenguaje de programación se precisa una librería distinta pues el método utilizado para transferir parámetros difiere sustancialmente entre lenguajes. Por otra parte se ralentiza el proceso de diseño pues el compilador debe linkar sus módulos con los incluidos en la librería.

Actualmente se dispone de librerías para los lenguajes de programación bajo DOS más extendidos BASIC (Qbasic y Basic Compiler de Microsoft), PASCAL (Turbo Pascal y Borland Pascal de Borland) y C (Turbo C, Borland C y MicroSoft C). Si su lenguaje de programación no está incluido entre éstos debe consultar con su proveedor para estudiar la creación de nuevas librerías.

II.3.2 Declaración de las funciones contenidas en las librerías

Todas las funciones necesarias para el control de los módulos LipSoft se encuentran en cada uno de los ficheros siguientes:

C	LIPC.OBJ
Pascal	LIPPAS.OBJ
BASIC	LIPBAS.LIB

El proceso para utilizar las funciones es siempre el mismo.

1º Declare un molde de la función a utilizar. Esto le permite al compilador verificar que las llamadas que se realicen está ajustadas a los requerimientos de la función. Los moldes de las funciones se encuentran incluidos en los ficheros:

C	LIPC.H
Pascal	LIPPAS.INC
BASIC	LIPBAS.BI

Para que queden definidas es suficiente que inserte en su programa la cláusula de inclusión de ficheros que toma la siguiente forma:

C	#include "LIPC.H"
Pascal	{ \$I LIPPAS.INC }
BASIC	'\$INCLUDE:'LIPBAS.BI'

2º Cree su programa con las llamadas a las funciones que precise y compílelo.

3º Enlace (linke) su programa junto con el módulo .OBJ o .LIB que contiene las funciones. Dependiendo de su lenguaje y del entorno de programación dispondrá de varias alternativas. Revise el manual de programación de su lenguaje para ver como se realiza el enlace con módulos externos.

4º Pruebe exhaustivamente su aplicación para detectar cualquier error.

II.3.3 Funciones disponibles

A continuación se describirá una a una todas las funciones que contienen las librerías para DOS, así como, el molde a utilizar y el significado de los parámetros. En la lista inferior se resumen todas las funciones disponibles con indicación del subsistema al que se aplican y una breve descripción sobre la misión que realizan.

Función	Subsistema	Descripción
inicializa	Todos	Establece el puerto de trabajo
restaura	Todos	Restaura el puerto
busca	Todos	Busca un subsistema determinado
escdig	E/S digitales	Escribe un valor en las líneas
leedig	E/S digitales	Lee el estado de las líneas

escda8	A/D 8 bits	Escribe un valor en el D/A
lead8	A/D 8 bits	Lee un canal A/D
esccont	Contador	Escribe un valor inicial cuenta
leecont	Contador	Lee la cuenta actual
actcont	Contador	Habilita el contaje de pulsos
descont	Contador	Deshabilita el contaje
actalarm	Contador	Habilita la alarma
desalarm	Contador	Deshabilita la alarma
leealarm	Contador	Lee la alarma actual
leeestcont	Contador	Lee el estado de trabajo actual
escda12	A/D12	Escribe un valor en el D/A*
lead12	A/D12	Lee un canal A/D*
escda12r	A/D12	Escribe un valor en el D/A
lead12r	A/D12	Lee un canal A/D

* Funciones para la versión hardware antigua del módulo AD12 adquiridas antes de 1996.

Es preciso indicar que todas las funciones pasan los parámetros por **referencia** (dirección del parámetro) y el orden en que ponen éstos en la pila es descendente (tipo Pascal). Las funciones no devuelven directamente ningún valor (como lo procedimientos de Pascal o SUB de Basic). Los resultados se transfieren a través de los parámetros que se le envían (modifican el valor de ciertos parámetros enviados). Generalmente cuando se produce un error en la comunicación con el/los módulos LipSoft se devuelve en el parámetro *dirección* (común a casi todas las funciones) un código de error (-1).

Función: **inicializa** (aplicable a todos los módulos)

Declaración:

C	extern void inicializa(int *puerto);
Pascal	PROCEDURE inicializa (VAR puerto:INTEGER):EXTERNAL
BASIC	DECLARE SUB inicializa (puerto%)

Descripción:

Esta rutina inicializa todo el bloque de rutinas, estableciendo el puerto Centronics en el que se encuentra el sistema LIPSOFT.

El valor de *puerto* puede ser 0, 1 o 2 para los puertos (en hexadecimal) 378, 278 y 3BC respectivamente. Si el puerto existe devolverá la dirección del mismo en la variable *puerto*, si no existe o el valor enviado es incorrecto devolverá -1.

Esta rutina deberá ser llamada en primer lugar, antes de usar

cualquier otra, y siempre después de **restaura**.

Función: **restaura** (aplicable a todos los subsistemas)

Declaración:

C	extern void restaura(void);
Pascal	PROCEDURE restaura ():EXTERNAL
BASIC	DECLARE SUB restaura ()

Descripción:

Esta rutina restaura las condiciones iniciales del puerto Centronics que se está utilizando. Debe usarse antes de acabar un programa y antes de utilizar la impresora.

Función: **busca** (aplicable a todos los subsistemas)

Declaración:

C	extern void busca(int *tipo, int *direccion);
Pascal	PROCEDURE busca (VAR tipo:INTEGER; VAR direccion:INTEGER):EXTERNAL
BASIC	DECLARE SUB busca (tipo% ,direccion%)

Descripción:

Esta rutina permite detectar la presencia de un determinado subsistema en el/los módulos LipSoft instalados.

La variable *tipo* identifica al subsistema. Puede tomar los valores: 0 (e/s digital), 1 (e/s analógica de 8 bits), 2 (e/s analógica de 12 bits) y 3 (contador).

La variable *direccion* debe tomar un valor adecuado al *tipo*. (Ver características técnicas de los módulos).

La rutina devuelve en la variable *direccion* un valor de los siguientes según se encuentre o no el subsistema:

- 0: dirección excesiva para el tipo indicado.
- 32: no existente el puerto Centronics seleccionado.
- 64: no es posible comprobar el subsistema especificado (tipo 2).
- 96: no se encuentra el módulo especificado.
- 128: módulo encontrado.

Función: escdig (Aplicable a los subsistemas de Entrada/Salida digital que contienen los módulos LipSoft-ADQ, ADT, DIG, IOT)

Declaración:

C	extern void escdig(int *dato, int *direccion);
Pascal	PROCEDURE escdig (VAR dato:INTEGER; VAR direccion:INTEGER):EXTERNAL
BASIC	DECLARE SUB escdig (dato% ,direccion%)

Descripción:

Esta rutina escribe un valor (entero entre 0 y 255) *dato* a un puerto de e/s digital.

Si *dato* toma un valor fuera del rango(0..255) los resultados son imprevisibles.

La variable *direccion* debe tomar un valor de 0 a 7.

La rutina devuelve en la variable *direccion* un valor -1 en el caso de que ocurra cualquier error (dirección inexistente, error de escritura, etc.)

Función: leedig (Aplicable a los subsistemas de Entrada/Salida digital que contienen los módulos LipSoft-ADQ, ADT, DIG, IOT)

Declaración:

C	extern void leedig(int *dato, int *direccion);
Pascal	PROCEDURE leedig (VAR dato:INTEGER; VAR direccion:INTEGER):EXTERNAL
BASIC	DECLARE SUB leedig (dato% ,direccion%)

Descripción:

Esta rutina lee un valor (*dato*) de un puerto de e/s digital. La variable *direccion* debe tomar un valor de 0 a 7.

La rutina devuelve en la variable *direccion* un valor -1 en el caso de que ocurra cualquier error (dirección inexistente, error de

lectura, etc.)

En la variable *dato* se devuelve valor leído en el rango (0..255)

Función: escda8 (Aplicable a los subsistemas de adquisición A/D y D/A analógicos de 8 bits de resolución que contienen los módulos LipSoft- ADQ, ADT, AD8)

Declaración:

C	extern void escda8(int *dato, int *direccion);
Pascal	PROCEDURE escda8 (VAR dato:INTEGER; VAR direccion:INTEGER):EXTERNAL
BASIC	DECLARE SUB escda8 (dato% ,direccion%)

Descripción:

Esta rutina escribe un valor (*dato*) a una salida analógica de 8 bits. Éste puede tomar cualquier valor entre 0 y 255; si excede este margen será ajustado de forma indeterminada.

La variable *direccion* debe tomar un valor de 0 a 7.

La rutina devuelve en la variable *direccion* un valor -1 en el caso de que ocurra cualquier error (dirección inexistente, error de escritura, etc.)

Función: leead8 (Aplicable a los subsistemas de adquisición A/D y D/A analógicos de 8 bits de resolución que contienen los módulos LipSoft- ADQ, ADT, AD8)

Declaración:

C	extern void leead8 (int *modo, int *canal, int *dato, int *direccion);
---	--

Pascal	PROCEDURE leead8 (VAR modo:integer; VAR canal:INTEGER; VAR dato:INTEGER; VAR direccion:INTEGER):EXTERNAL
BASIC	DECLARE SUB leead8 (modo%, canal%, dato% ,direccion%)

Descripción:

Esta rutina lee un valor (*dato*) de una entrada analógica de 8 bits. La variable *modo* puede tomar un valor de 0 a 3, indicando:

- 0: 4 canales independientes.
- 1: 3 canales diferenciales con común.
- 2: 2 canales independientes y uno diferencial.
- 3: 2 canales diferenciales.

si excede de este margen será ajustado de forma indeterminada. Ver características técnicas de los módulos que contienen subsistemas de conversión A/D de 8 bits (ADQ, ADT o AD8).

La variable *canal* indica el canal a leer. Puede tomar un valor de 0 a 3 y si excede de este margen será ajustado de forma indeterminada.

La variable *direccion* debe tomar un valor de 0 a 7.

La rutina devuelve en la variable *direccion* un valor -1 en el caso de que ocurra cualquier error (dirección inexistente, error de lectura, etc.)

En la variable *dato* se devuelve valor leído. Este podrá tomar cualquier valor entre 0 y 255.

Función: *escont* (Aplicable a los subsistemas Contadores que contienen los módulos LipSoft- ADT, IOT)

Declaración:

C	extern void escont(long *dato!, int *direccion);
Pascal	PROCEDURE escont (VAR dato!:LONGINT; VAR direccion:INTEGER):EXTERNAL
BASIC	DECLARE SUB escont (dato!& ,direccion%)

Descripción:

Esta rutina escribe un valor (*dato!*) en el registro de contaje del contador direccionado mediante la variable *direccion*.

La variable *dato!* es el dato a escribir. Este puede tomar cualquier

valor entre 0 y 999999; si excede de este margen será ajustado de forma indeterminada.

La variable *direccion* debe tomar un valor de 0 a 1.

La rutina devuelve en la variable *direccion* un valor -1 en el caso de que ocurra cualquier error (dirección inexistente, error de escritura, etc.)

Función: *leecont* (Aplicable a los subsistemas Contadores que contienen los módulos LipSoft- ADT, IOT)

Declaración:

C	extern void leecont(long *dato!, int *direccion);
Pascal	PROCEDURE leecont (VAR dato!:LONGINT; VAR direccion:INTEGER):EXTERNAL
BASIC	DECLARE SUB leecont (dato!& ,direccion%)

Descripción:

Esta rutina lee la cuenta actual de un contador.

La variable *direccion* debe tomar un valor de 0 a 1.

La rutina devuelve en la variable *direccion* un valor -1 en el caso de que ocurra cualquier error (dirección inexistente, error de lectura, etc.)

En la variable *dato!* se devuelve valor leído. Este puede tomar cualquier valor entre 0 y 999999.

Función: *actcont* (Aplicable a los subsistemas Contadores que contienen los módulos LipSoft- ADT, IOT)

Declaración:

C	extern void actcont(int *direccion);
Pascal	PROCEDURE actcont (VAR direccion:INTEGER):EXTERNAL
BASIC	DECLARE SUB actcont (direccion%)

Descripción:

Esta rutina activa (permite el conteo de pulsos) un contador.

La variable *direccion* debe tomar un valor de 0 a 1.

La rutina devuelve en la variable *direccion* un valor -1 en el caso de que ocurra cualquier error (dirección inexistente, error de escritura, etc.)

Función: descuent (Aplicable a los subsistemas Contadores que contienen los módulos LipSoft- ADT, IOT)

Declaración:

C	extern void descuent(int *direccion);
Pascal	PROCEDURE descuent (VAR direccion:INTEGER):EXTERNAL
BASIC	DECLARE SUB descuent (direccion%)

Descripción:

Esta rutina desactiva (impide el conteo de pulsos) un contador.

La variable *direccion* debe tomar un valor de 0 a 1

La rutina devuelve en la variable *direccion* un valor -1 en el caso de que ocurra cualquier error (dirección inexistente, error de escritura, etc.)

Función: actalarm (Aplicable a los subsistemas Contadores que contienen los módulos LipSoft- ADT, IOT)

Declaración:

C	extern void actalarm(long *datol, int *direccion);
Pascal	PROCEDURE actalarm (VAR datol:LONGINT; VAR direccion:INTEGER):EXTERNAL
BASIC	DECLARE SUB actalarm (datol&, direccion%)

Descripción:

Esta rutina activa la alarma de un contador.

La variable *datol* indica el valor de alarma a programar. Ésta puede tomar cualquier valor entre 0 y 999999; si excede de este margen será ajustado de forma indeterminada.

La variable *direccion* debe tomar un valor de 0 a 1.

La rutina devuelve en la variable *direccion* un valor -1 en el caso de que ocurra cualquier error (dirección inexistente, error de escritura, etc.)

Función: desalarm (Aplicable a los subsistemas Contadores que contienen los módulos LipSoft- ADT, IOT)

Declaración:

C	extern void desalarm(int *direccion);
Pascal	PROCEDURE desalarm (VAR direccion:INTEGER):EXTERNAL
BASIC	DECLARE SUB desalarm (direccion%)

Descripción:

Esta rutina desactiva la alarma de un contador. Ver características técnicas de los módulos ADT o IOT para obtener más información sobre el significado de la alarma.

La variable *direccion* debe tomar un valor de 0 a 1.

La rutina devuelve en la variable *direccion* un valor -1 en el caso de que ocurra cualquier error (dirección inexistente, error de escritura, etc.)

Función: leealarm (Aplicable a los subsistemas Contadores que contienen los módulos LipSoft- ADT, IOT)

Declaración:

C	extern void leealarm (long *datol, int *direccion);
Pascal	PROCEDURE leealarm (VAR datol:LONGINT; VAR direccion:INTEGER):EXTERNAL
BASIC	DECLARE SUB leealarm (datol&, direccion%)

Descripción:

Esta rutina lee el valor programado en la alarma de un contador. Ver características técnicas de los módulos ADT o IOT para obtener más información sobre el significado de la alarma.

La variable *direccion* debe tomar un valor de 0 a 1.

La rutina devuelve en la variable *direccion* un valor -1 en el caso de que ocurra cualquier error (dirección inexistente, error de lectura, etc.)

En la variable *dato* se devuelve valor leído. Éste estará dentro del rango 0..999999.

Función: leeestcont (Aplicable a los subsistemas Contadores que contienen los módulos LipSoft- ADT, IOT)

Declaración:

C	extern void leeestcont(int *dato, int *direccion);
Pascal	PROCEDURE leeestcont (VAR dato:INTEGER; VAR direccion:INTEGER):EXTERNAL
BASIC	DECLARE SUB leeestcont (dato% ,direccion%)

Descripción:

Esta rutina lee el estado de un contador.

La variable *direccion* debe tomar un valor de 0 a 1.

La rutina devuelve en la variable *direccion* un valor -1 en el caso de que ocurra cualquier error (dirección inexistente, error de lectura, etc.)

En la variable *dato* se devuelve la palabra de estado leída. Este puede tomar los valores:

- 0: contaje y alarma desactivados.
- 4: contaje desactivado, alarma activada.
- 128: contaje desactivado, alarma desactivada.
- 132: contaje y alarma activados

Función: escda12 (Aplicable a los subsistemas de adquisición A/D y D/A analógicos de 12 bits de resolución que contienen el módulo LipSoft- AD12) (Versión hardware antigua)

escda12r

Versión para los módulos adquiridos a partir del 20 Enero 1996.

Declaración:

C	extern void escda12 (int *dato, int *direccion); extern void escda12r (int *dato, int *direccion);
Pascal	PROCEDURE escda12 (VAR dato:INTEGER; VAR direccion:INTEGER):EXTERNAL PROCEDURE escda12r (VAR dato:INTEGER; VAR direccion:INTEGER):EXTERNAL
BASIC	DECLARE SUB escda12 (dato% ,direccion%) DECLARE SUB escda12r (dato% ,direccion%)

Descripción:

Esta rutina escribe un *dato* a una salida analógica de 12 bits.

La variable *dato* es el valor a escribir. Éste debe de estar dentro del rango 0 y 4095; si excede de este margen será ajustado de forma indeterminada.

La variable *direccion* debe tomar un valor de 0 a 2.

La rutina devuelve en la variable *direccion* un valor -1 en el caso de que ocurra cualquier error (dirección inexistente, error de escritura, etc.)

Función: leead12

(Aplicable a los subsistemas de adquisición A/D y D/A analógicos de 12 bits de resolución que contienen el módulo LipSoft- AD12) (Versión hardware antigua)

leead12r

Versión para los módulos adquiridos a partir del 20 Enero 1996.

Declaración:

C	extern void leead12 (int *modo, int *canal, int *dato, int *direccion); extern void leead12r (int *modo, int *canal, int *dato, int *direccion);
---	---

Pascal	PROCEDURE leead12 (VAR modo:integer; VAR canal:INTEGER; VAR dato:INTEGER; VAR direccion:INTEGER):EXTERNAL; PROCEDURE leead12 (VAR modo:integer; VAR canal:INTEGER; VAR dato:INTEGER; VAR direccion:INTEGER):EXTERNAL;
BASIC	DECLARE SUB leead12 (modo%, canal%, dato% ,direccion%) DECLARE SUB leead12 (modo%, canal%, dato% ,direccion%)

Descripción:

Esta rutina lee un dato de una entrada analógica de 12 bits.

La variable *modo* puede tomar un valor de 0 o 1 , indicando:

- 0: 8 canales independientes.
- 1: 4 canales diferenciales.

si excede de este margen será ajustado de forma indeterminada.

La variable *canal* indica el canal a leer. Puede tomar un valor de 0 a 7 y si excede de este margen será ajustado de forma indeterminada.

La variable *direccion* debe tomar un valor de 0 a 2.

La rutina devuelve en la variable *direccion* un valor -1 en el caso de que ocurra cualquier error (direccion inexistente, error de lectura, etc.)

En la variable *dato* se devuelve valor leído. Este puede tomar cualquier valor entre 0 y 4095.

Nota importante: Para los versión hardware antigua de los módulos AD12, en modo diferencial la lectura obtenida de cada canal corresponde a la diferencia entre las señales $ch0=ch0-ch1$ $ch1=ch1-ch0$ y así respectivamente. Si el resultado de esta operación fuese negativo la lectura sería 0 en el canal correspondiente y el valor correcto en el siguiente. Por ejemplo si $ch1=300$ y $ch0=400$ obtendríamos al leer los canales 0 y 1 , $ch0=100$ y $ch1=0$.

Función: dirrut**Declaración:**

BASIC	DECLARE SUB dirrut (sgm%, dsp%)
-------	---------------------------------

Descripción:

Esta rutina sólo está disponible para BASIC. Su función es devolver los valores del segmento (en la variable *sgm*) y desplazamiento (en la variable *dsp*) de la dirección **SetUevent** necesarios para usar las posibilidades de interrupciones del controlador LIPSOFT.SYS con la instrucción ON UEVENT GOSUB del BASIC.

III.3.4 Funciones disponibles para lenguaje ensamblador

Todas las funciones descritas en el apartado anterior están disponibles, con el mismo nombre, en la librería LIPASM.LIB para poder ser invocadas desde lenguaje ensamblador. En este caso, los parámetros necesarios se envían y reciben en registros.

Al llamar a cualquiera de las rutinas, en el registro **DX** debe especificarse la dirección del *puerto* Centronics a utilizar más 2, es decir, si se usa el puerto 0378H, DX valdrá 037Ah (0378h + 2 = 037Ah); en el registro **BL** debe indicarse la *dirección* del módulo a leer o escribir. Al finalizar cada rutina, el **flag de acarreo** quedará desactivado salvo que ocurra algún error de lectura o escritura. Aparte de estos datos comunes, cada rutina debe ser llamada con los siguientes registros:

BUSCA	AL: tipo. Devuelve código en BL.
ESCDIG	AL: valor a escribir.
LEEDIG	Devuelve valor leído en AL.
ESCDA8	AL: valor a escribir.
LEEAD8	AL: modo, AH: canal. Devuelve valor leído en AL.
ESCCONT	CX:AX: valor a escribir (CX la parte más significativa).
LEECONT	Devuelve valor leído en BX:AX. (BX la parte más significativa).
ACTCONT	--
DESCONT	--
ACTALARM	CX:AX: valor a programar. (CX la parte más significativa).
DESALARM	--
LEEALARM	Devuelve valor programado en BX:AX. (BX la parte más significativa).
LEEESTCONT	Devuelve estado en AL.

ESCDA12	AX: valor a escribir.*
LEEAD12	AL: modo, AH: canal. Devuelve valor leído en AX. *
ESCDA12R	AX: valor a escribir.
LEEAD12R	AL: modo, AH: canal. Devuelve valor leído en AX.

* Para la versión hardware antigua del módulo AD12 (adquiridos antes del 1996)

Manual de Usuario

Capítulo III

Soporte Software bajo Windows

III.1 Introducción

La comunicación entre el ordenador y los módulos *LipSoft* se realiza mediante un protocolo de comunicación específico y patentado a través de las líneas de control y estado del puerto Centronics. Cuando se trata de realizar una función, por ejemplo leer el valor convertido en un canal A/D, una rutina de bajo nivel genera una secuencia en las líneas del puerto de modo que entran en comunicación el ordenador y el módulo. La información que comparten es adaptada y transmitida desde/ hacia su lenguaje de alto nivel.

El programador no necesita conocer nada a cerca de este protocolo, ya que su relación con los módulos se realiza a través de funciones de alto nivel que internamente realizarán todo el trabajo para establecer esta comunicación, y devolverán un conjunto de datos correspondiente a la información solicitada.

Para el manejo de los módulos LipSoft desde aplicaciones Windows se dispone de dos alternativas:

- a) Realizar llamadas a las funciones contenidas en librerías de enlace dinámico (DLL). Esta técnica puede ser empleada bajo cualquier lenguaje de programación para el entorno Windows, ya que se encuentra estandarizado el modo de traspaso de parámetros.
- b) Si programa bajo Visual Basic ver < 4.0 o Visual C, puede utilizar los controles de usuario que se incluyen. Los archivos de éstos controles son los mismos de las librerías DLL, si bien solo Visual Basic y Visual C, pueden sacar partido de su utilización como Controles de Usuario. De cualquier modo podrá seguir utilizando las llamadas del apartado a.

Nota: El archivo LIPSOFT.HLP contiene la ayuda disponible para el manejo de las funciones y controles bajo Windows.

Nota importante: Si utiliza el control de usuario AD12B.DLL o las funciones de esta librería, deberá seleccionar en el panel de control de Windows /IMPRESORAS Conectar..., el puerto de impresora tipo LPTx.DOS para poder imprimir sin problemas.

III.2 Controles de usuario (para Visual Basic 3.0)

En los siguientes apartados se describirán los diferentes controles de usuario (Custom Control) disponibles para su utilización desde el lenguaje Visual Basic.

Estos controles han sido desarrollados para conseguir una gran simplicidad en el manejo de los módulos LipSoft. Disponen de un amplio conjunto de propiedades y eventos que le permitirá conseguir un elevado rendimiento con un mínimo esfuerzo de programación.

Los ficheros correspondientes a los Controles pueden funcionar como controles de usuario o como librerías de enlace dinámico (DLL). Por tanto, podrá utilizar simultáneamente las funciones descritas en los apartados III.3 y siguientes a la vez que las propias del control.

Para utilizar los controles desde Visual Basic ver < 4.0, basta que añada en su proyecto el fichero "AD12B.DLL" y/o "AD12R.DLL" y/o "AD8B.DLL" y/o "DIG.DLL" y/o "CONT.DLL" y/o "IRQ.VBX". Instantáneamente aparecerá en la caja de herramientas (ToolBox) el icono correspondiente del control. Los controles no son visibles en tiempo de ejecución, al igual que el control estándar Timer, por tanto, no se ha de preocupar donde lo sitúa en su Form. Si intenta redimensionarlo éste tomará siempre el tamaño del icono.

Para obtener ayuda en tiempo de diseño, basta pulsar F1 cuando tenga seleccionado algún control.

En el disco de utilidades se incluye el código fuente de la aplicación CC_DEMO para Visual Basic 3.0 que maneja todos los controles de usuario disponibles para los módulos LipSoft.

Nota: No olvide incluir los ficheros correspondientes (*.DLL) o (IRQ.VBX) en su producto final y hacer que se instalen en el lugar preciso (generalmente en el subdirectorio WINDOWS\SYSTEM) para que hacer accesibles las funciones.

III.2.1 Control de Usuario AD12B* AD12R



III.2.1.1 Descripción

Estos controles permiten gestionar completamente el módulo de adquisición de datos analógico/digital de 12 bits LipSoft-AD12 (* Versión hardware antigua anterior 1996).

Los controles disponen de un conjunto de propiedades que permiten programar el proceso de adquisición de datos

analógicos de cualquiera de los 8 canales con que cuenta el módulo LipSoft-AD12. La funcionalidad de los controles es idéntica.

Como en todo proceso de adquisición es preciso definir varios parámetros:

a) Canal o grupo de canales desde donde se capturarán los datos. Los valores digitales adquiridos se almacenarán automáticamente en cuatro arrays denominados Buffer_ADx (que corresponden a propiedades no visibles). Estos arrays mantienen secuencialmente el valor convertido, por ejemplo Ad12b1.Buffer_AD2[13] o Ad12r1.Buffer_AD2[13] contiene la muestra 14ª del canal analógico 2.

b) Tiempo entre cada muestra, o lo que es igual Período. Este parámetro indica el lapso de tiempo que transcurrirá entre cada muestra adquirida. En su funcionamiento el control crea un temporizador propio (Timer) con intervalo=Período. Cada vez que ha transcurrido el tiempo fijado en intervalo entra en funcionamiento la rutina encargada de leer la información del grupo de canales correspondientes y traspasarla a los Buffers.

c) Muestras. Mediante este parámetro se indica el número total de muestras que se han de tomar, espaciadas en el tiempo un intervalo=Período.

d) Método de disparo, o inicio de la adquisición. En esta propiedad se fija la condición para el comienzo de la adquisición. Existen dos posibilidades, por Software y por Nivel. El disparo por Software hace que se inicie el proceso de adquisición en el instante en que se da la orden por programa, haciendo la propiedad Acción=1. Si se elige la opción de disparo por nivel el proceso real de adquisición no comienza hasta que no se ha alcanzado un determinado valor en uno de los canales analógicos (el canal programado como canal de disparo).

El módulo LipSoft-AD12 dispone también de un canal de conversión Digital/Analógica. Para conseguir la salida de una señal por este canal es preciso cargar el conjunto de valores a convertir en el array Buffer_DA[]. Este array debe contener en cada uno de sus elementos el valor instantáneo que se desea para la señal. En el momento en que se inicie el proceso (haciendo la propiedad Acción=1) comenzarán a aparecer en el canal analógico una señal eléctrica proporcional al valor digital almacenado en el array correspondiente a esa muestra. Por ejemplo si se fija un Período=1000 ms(1 segundo), Muestras=20, y en la propiedad Buffer_DA[10]=3075, transcurridos 10 períodos=10 segundos desde que se active la propiedad Acción=1 aparecerá en el canal analógico un valor de 3.075 Voltios que se mantendrá hasta la siguiente muestra. Cuando se haya generado todas las muestras programadas y si

el proceso de adquisición no ha terminado (esto solo es posible en el caso de disparo por nivel) se reiniciará la cuenta desde la primera muestra.

Para indicar al control que se inicie/(finalice prematuramente) el proceso de adquisición se dispone de la propiedad *Acción*. Si se pone a 1 se arranca el proceso, si se hace 0 se termina inmediatamente.

El control dispone además de dos propiedades para direccionar el módulo, estas son *Puerto* y *Dirección*. Mediante la propiedad *Puerto* se indica al control la dirección física del puerto paralelo del ordenador, estas pueden ser en hexadecimal (&H278, &H378 y &H3BC). Mediante la propiedad *Dirección* se indica qué módulo de los 3 posibles es al que se quiere acceder. Tenga en cuenta que es posible mantener varios módulos conectados al mismo puerto, siempre y cuando se hayan programado (mediante puentes) las direcciones de forma correcta (ver documentación técnica).

Nombre del Fichero: AD12B.DLL o AD12R.DLL

Tipo de Objeto: Ad12b o Ad12r

III.2.1.2 Propiedades

El acceso en tiempo de ejecución a las propiedades desde Visual Basic se realiza siempre del mismo modo:

```
[Form.]Ad12?.Propiedad = valor
valor = [Form.]Ad12?.Propiedad
```

Como todas las propiedades son del tipo numérico el acceso a éstas desde Visual C es siempre idéntico, solo cambia el nombre de la propiedad a la que se accede y el valor. Ejemplo:

```
pAd12?- > GetNumProperty("Propiedad_a_Leer")
```

```
pAd12?- > SetNumProperty("Propiedad_a_Escribir",
valor)
```

Donde **Propiedad** identifica el nombre de la propiedad a leer o escribir y valor es un tipo compatible (entero, entero largo etc..)

Donde ?=b o r según la versión hardware del módulo que se quiera controlar (b antigua: anterior 1996, r moderna)

VISIBLES (Las que aparecen en la ventana "Properties").

Canal_0 Canal_7 (Entero 0.. ± 4095)

Estas 7 propiedades mantienen constantemente el valor actual en los canales correspondientes. Son propiedades de solo lectura. El refresco (actualización de su valor) en tiempo de diseño se realiza cada vez que se activa la propiedad *Test*. Cada vez que se lee su valor en tiempo de ejecución se produce un proceso de adquisición de datos y se actualiza su valor. En procesos de adquisición automática (haciendo *Acción*=1 y fijando el *Período* y *Muestras*) NO es conveniente acceder al valor instantáneo del *Canal_x* leyendo estas propiedades ya que lanzaríamos procesos intermedios de adquisición que ralentizarían excesivamente el proceso. La adquisición utilizando estas propiedades tiene en cuenta el *Modo* actual. En modo diferencial los valores pueden ser negativos (si el valor en el 2º canal es superior al del primero).

Canal_Disparo (0=1º, 1=2º, 2=3º, 3=4º)

Identifica el canal que servirá para realizar el disparo por nivel en el caso de que se haya seleccionado esta opción en la propiedad *Disparo*. Puede tomar los valores 0,1,2,3 correspondientes al 1º,2º,3º o 4º canal del *Grupo_Canales* seleccionados.

Grupo_Canales (0=Inferior, 1=Superior)

Identifica el grupo de canales del que se muestrearán. Puede tomar los valores 0 para el grupo de canales Inferior o 1 para el grupo Superior. Los valores captados se van almacenando en consecutivamente cada muestra en las propiedades no visibles *Buffer_ADx()* donde x (0,1,2,3).

Por ejemplo: Tras finalizar un proceso de muestreo para acceder al valor muestreado en el instante 16 del canal 2 (puede ser el canal 2 o el canal 6 según el grupo de canales) pondremos a%=Ad12?1.Buffer_AD2(16).

Dirección (0,1,2)

Permite seleccionar las tres direcciones posibles 0,1,2 de LipSoft AD12.

Disparo (0=Software, 1=Nivel)

Se selecciona el tipo de disparo que iniciará el proceso de muestreo puede tomar los valores:

0. Software. Cada vez que se activa la propiedad no visible *Acción*=1 se producirá el evento *SOFT*. En el cuerpo de este evento puede realizarse todas las acciones que se consideren oportunas.

1. Nivel. Hace que comience la adquisición cuando se haya alcanzado en el canal seleccionado *Canal_Disparo* el valor ajustado en la propiedad *Nivel_Disparo*. Cuando se ha

alcanzado el nivel programado se producirá el evento *NIVEL*.

Finalizado el proceso de adquisición siempre se producirá el evento *FIN*.

Muestras (Entero 1..255)

Permite seleccionar el número de muestras totales a adquirir. Debe ser > 0 y < 256.

Modo (0 = Simple, 1= Diferencial)

Indica el modo en que funcionarán los canales 0=Simple, 1=Diferencial. En modo diferencial ch0=ch0-ch1, ch1=ch2-ch3, ch2=ch4-ch5, ch3=ch6-ch7. En modo diferencial los valores pueden ser positivos o negativos según el signo de la diferencia entre canales. Además, en este modo, los valores en las propiedades *Canal_4*, *Canal_5*, *Canal_6* y *Canal_7* no tienen sentido.

Nivel_Disparo (Entero 0.. 4095)

Indica el valor en el canal seleccionado como canal de disparo a partir del cual comenzará el proceso de adquisición. Solo tiene efecto si se ha seleccionado el tipo de disparo "1. Nivel".

Período (Entero > 5 < 32000)

Identifica el lapsus de tiempo que transcurrirá entre la adquisición de cada muestra en milisegundos. (Atención la resolución es de 55 ms. por tanto un intento de muestrear con una cadencia menor no tendrá efecto).

Puerto (0,1,2)

Permite seleccionar el puerto de impresora donde se encuentra conectado LipSoft AD12. Los valores pueden ser 0-&H378, 1-&H278, 2-&H3BC que generalmente corresponden a LPT1, LPT2, LPT3.

Salida_DA (Entero 0..4095)

Esta propiedad, de solo escritura, hace que aparezca inmediatamente en el canal Digital/Analógico una tensión proporcional (1 unidad = 10mV) al valor asignado.

Test (0=OFF, 1=ON)


Al pasar de OFF a ON se produce un refresco de los valores en las propiedades *Canal_X*. Se debe utilizar solo en tiempo de diseño, ya que en tiempo de ejecución cualquier acceso a los canales actualiza su valor.

Tag e Index

Son propiedades estándar de Visual Basic.

Top y Left

Son propiedades estándar que solo tienen efecto en tiempo

	Nº Error	Mensaje y Explicación
de diseño ya que el control es no visible en tiempo de ejecución.		
NO VISIBLES		
Acción (0,1) Esta propiedad permite disparar o detener el proceso de adquisición. Puede tomar los valores: 0. Termina el proceso de adquisición en curso. 1. Inicia el proceso de adquisición.	380	Invalid property value Se produce al tratar de asignar a una propiedad un valor fuera del rango o tipo.
Acción (0,1) Esta propiedad permite disparar o detener el proceso de adquisición. Puede tomar los valores: 0. Termina el proceso de adquisición en curso. 1. Inicia el proceso de adquisición.	381	Invalid property array index Se genera cuando se comete un error en el índice durante el acceso a los Buffer_ADx() o Buffer_DA()
Buffer_DA(j%) (Array 0..255 Enteros 0..4095) En esta matriz se almacenan los valores que se desean aparezcan en el canal de salida Digital /Analógico. A partir del momento en que se dispare la adquisición mediante la propiedad <i>Acción=1</i> comenzarán a aparecer sincronizadamente con el período de muestreo uno a uno los valores almacenados en esta propiedad. Se admiten solo valores enteros entre 0..4.095 -> 0 a 4'095 Voltios.	32105	No hay TIMERS disponibles Este error se produce cuando se solicita a Windows la creación de un nuevo temporizador para la gestión de un proceso de muestreo.
Buffer_ADx(j%) (Array 0..255 Enteros 0..4095) Estas matrices almacenan los valores captados en los canales x=0,1,2,3 cada período de muestreo. Los valores almacenados serán enteros (0..4095).		
III.2.1.3 Eventos	III.2.2 Control de Usuario CONT	
FIN Este evento se produce cada vez que se han adquirido todas las muestras indicada en la propiedad <i>Muestras</i> .	III.2.2.1 Descripción	
NIVEL Se lanza este evento cuando se ha alcanzado en el canal seleccionado como canal de disparo el nivel establecido en la propiedad <i>Nivel_Disparo</i> y siempre y cuando se haya seleccionado como técnica de disparo "Nivel" en la propiedad <i>Disparo</i> .	Este control permite gestionar los módulos LipSoft que contienen contadores, estos son : LipSoft-ADT y LipSoft-IOT. El primero contiene un solo contador mientras que el segundo contiene dos contadores.	
SOFT Se produce este evento cada vez que se inicia un proceso de conversión haciendo <i>Acción=1</i> y se haya seleccionado como método de disparo "Disparo=0 Software".	Los contadores que incluyen estos módulos permiten totalizar hasta 999999 eventos (cambios en nivel lógico "0" a "1"). El valor almacenado en los contadores puede ser reescrito o leído en cualquier momento mediante la propiedad <i>Cuenta</i> . Además se dispone de la posibilidad de programación y activación de los registros de alarma que contienen los contadores mediante la propiedad <i>Alarma</i> y <i>Habilitación_Alarma</i> . Mediante las propiedades <i>Puerto</i> y <i>Dirección</i> es posible seleccionar el puerto de impresora donde se encuentra conectado el/los módulos y la dirección asignada (mediante puentes) del contador.	
III.2.1.4 Mensajes de Error	Nombre del Fichero: CONT.DLL	
La lista siguiente indica el número de error y su significado devueltos por el control AD12B.	Tipo de Objeto: Cont	
	III.2.2.2 Propiedades	
	El acceso en tiempo de ejecución a las propiedades desde Visual Basic se realiza siempre del mismo modo:	

[Form.]Cont.**Propiedad**= valor

valor =[Form.]Cont.**Propiedad**

Como todas las propiedades son del tipo numérico el acceso a éstas desde Visual C es siempre idéntico, solo cambia el nombre de la propiedad a la que se accede y el valor. Ejemplo:

```
pCont-> GetNumProperty("Propiedad_a_Leer")
```

```
pCont-> SetNumProperty("Propiedad_a_Escribir",
valor)
```

Donde **Propiedad** identifica el nombre de la propiedad a leer o escribir y valor es un tipo compatible (entero, entero largo etc..)

VISIBLES (Las que aparecen en la ventana "Properties").

Alarma (TRUE,FALSE)

Si TRUE, activa la generación de un nivel bajo en la línea "INT" cuando se reciba un impulso que hace que el valor totalizado en contador sea igual al contenido en su registro de alarma.

La "INT" no se activa por el simple hecho de que $Valor_Cuenta=Valor_Alarma$. Es necesario que esta **igualdad se produzca como consecuencia de un incremento** (recepción de un impulso en la entrada del contador). Al desactivar la alarma, $Alarma=FALSE$, la línea "INT" se desactiva (queda abierta). El hacer nuevamente $Alarma=TRUE$ no activa automáticamente la línea "INT" ya que es preciso que se reciba un impulso de contaje para la activación y en tal caso $Valor_Cuenta > Valor_Alarma$. La línea "INT" puede cablearse de modo que produzca una activación de la interrupción Hardware IRQ7 o IRQ5 (ver documentación técnica).

Contar (TRUE,FALSE)

Esta propiedad permite iniciar o parar el proceso de contaje. Puede tomar los valores:

0= False. Detiene el contaje.

1=True. Inicia el contaje.

Dirección (0,1)

Permite seleccionar una de las dos (0,1) direcciones posibles para los contadores.

Puerto (0,1,2,3)

Permite seleccionar uno de los tres puertos de impresora donde pueden estar conectados los módulos que contienen contadores (LipSoft-IOT o LipSoft-ADT). Los valores pueden ser:

0. Autodetección. Hace que al disparar la propiedad *Test* se busque en todos los posibles puertos de impresora del ordenador hasta encontrar una un módulo que contenga contadores.

1.&H378 Corresponde generalmente a LPT1:

2.&H278 Corresponde normalmente a LPT2:

3.&H3BC

Test (0=OFF, 1=ON)

Al dispararlo (pasar de OFF=0 a ON=1) se produce un proceso de autodetección de módulos LipSoft que contengan contadores en el puerto paralelo indicado en la propiedad *Puerto*. La búsqueda es ascendente a partir del valor actual en la propiedad *Dirección*. Si se encuentra algún módulo se actualizará automáticamente los valores en *Dirección* y *Puerto*. Si no se encuentra ningún módulo LipSoft con contadores se producirá un error.

Con la propiedad desactivada ($Test=0$ OFF) se evita que se produzca mensajes de error al intentar acceder a propiedades que por no estar conectado ningún módulo LipSoft provocan un error por falta de comunicación (Ejemplo al tratar de escribir *Cuenta* en el contador y no existir ninguno conectado)

Valor_Alarma (Long 0..999999)

Permite seleccionar el valor de contaje para el cual se activará la alarma (Salida INT a nivel bajo) si la propiedad *Alarma* esta activada " $=TRUE$ ".

Valor_Cuenta (Long 0..999999)

Contiene el valor actual de contaje. Si se escribe se guarda esta cuenta en los contadores. En lectura siempre lee el estado actual de la cuenta.

Tag e Index

Son propiedades estándar de Visual Basic.

Top y Left

Son propiedades estándar que solo tienen efecto en tiempo de diseño ya que el control es no visible en tiempo de ejecución.

NO VISIBLES

(Ninguna)

III.2.2.3 Eventos

(Ninguno)

III.2.2.4 Mensajes de error

La lista siguiente indica el número de error y su significado devueltos por el control CONT.

Nº Error	Mensaje y Explicación
380	Invalid property value Se produce al tratar de asignar a una propiedad un valor fuera del rango o tipo.
32200	Autodetección Se genera este error cuando se trata de buscar algún módulo LipSoft en el puerto especificado y no se encuentra. Falla la comunicación.
32201	Escribir Cuenta Se produce al intentar escribir la cuenta en el contador direccionado.
32202	Leer Cuenta Se produce al intentar leer la cuenta actual del contador direccionado.
32203	Escribir Alarma Se produce al intentar escribir un valor para la Alarma en el contador direccionado.
32204	Leer Alarma Se produce al intentar leer la Alarma actualmente programada.
32205	Activar Contaje Se produce al intentar habilitar el contaje en el contador direccionado.
32206	Parar Contaje Se produce al intentar deshabilitar el contaje en el contador direccionado.
32207	Activar Alarma Se produce al intentar habilitar la Alarma en el contador direccionado.

32208

Parar Alarma

Se produce al intentar deshabilitar la Alarma en el contador direccionado.

Nota: Los errores del 32200 en adelante se produce cuando no se consigue establecer comunicación con el módulo LipSoft correspondiente (direccionado). Esto puede deberse a tres causas fundamentales:

- No se ha direccionado convenientemente (Seleccione bien las propiedades Puerto y Dirección).
- Si tiene conectada en el extremo final de la cadena de módulos LipSoft una impresora, ésta debe de estar encendida y con papel.
- Fallo eléctrico en el módulo o el puerto Centronics. Pruebe con otro puerto o con otro módulo.

III.2.3 Control de Usuario DIG



III.2.3.1 Descripción

Este control permite gestionar las líneas Entrada/Salida digital que contienen los módulos LipSoft-ADQ, LipSoft-IOT y LipSoft-DIG.

La propiedad *Valor* permite controlar el estado eléctrico de las líneas digitales. Al escribir un byte en esta propiedad aparecerá su equivalente binario 0 (nivel bajo) 1 (nivel alto) en las líneas digitales del módulo LipSoft direccionado. Al leer la propiedad *Valor* se obtendrá el estado actual de las líneas. Tenga en cuenta que se trata de líneas Cuasi-Bidireccionales es decir para que una línea actúe como entrada es preciso que se programe a un nivel lógico 1, así si el sistema exterior fuerza un 1 o un 0 lógico en esa línea se leerá el valor correcto (ver características técnicas de los módulos).

Mediante las propiedades *Puerto* y *Dirección* es posible seleccionar el puerto de impresora donde se encuentra conectado el/los módulos y la dirección asignada (ver características técnicas de los módulos).

Nombre del Archivo: DIG.DLL

Tipo de Objeto: Dig

III.2.3.2 Propiedades

El acceso en tiempo de ejecución a las propiedades desde Visual Basic se realiza siempre del mismo modo:

[Form.]Dig.**Propiedad**= valor

valor =[Form.]Dig.**Propiedad**

Como todas las propiedades son del tipo numérico el acceso a éstas desde Visual C es siempre idéntico, solo cambia el nombre de la propiedad a la que se accede y el valor. Ejemplo:

pDig-> **GetNumProperty**("Propiedad_a_Leer")

pDig-> **SetNumProperty**("Propiedad_a_Escribir",
valor)

Donde **Propiedad** identifica el nombre de la propiedad a leer o escribir y valor es un tipo compatible (entero, entero largo etc..)

VISIBLES (Las que aparecen en la ventana "Properties").

Dirección (Entero 0..7)

Permite seleccionar una de las 8 (0..7) direcciones posibles para los canales I/O digitales.

Puerto (0,1,2,3)

Permite seleccionar uno de los tres puertos de impresora donde pueden estar conectados los módulos que contienen canales I/O digitales (LipSoft-DIG, LipSoft-IOT o LipSoft-ADQ). Los valores pueden ser:

0. Autodetección. Hace que al disparar la propiedad *Test* se busque en todos los posibles puertos de impresora del ordenador hasta encontrar un módulo que contenga canales I/O.

1.&H378 Corresponde generalmente a LPT1:

2.&H278 Corresponde normalmente a LPT2:

3.&H3BC

Valor (Entero 0..255)

En un proceso de escritura de la propiedad se transfiere esta cantidad al registro de salida de los canales I/O, apareciendo cada 1 de la palabra binaria correspondiente como un nivel alto (5 V) y cada 0 como un nivel bajo (0 V). Al poner un "1" en el bit -> línea I/O correspondiente esta línea puede actuar como línea de entrada.

En un proceso de lectura de la propiedad se transfiere el estado eléctrico (0 o 1) de las líneas creando la palabra binaria correspondiente.

Test(0=OFF, 1=ON)

Al dispararlo (pasar de OFF=0 a ON=1) se produce un

proceso de autodetección de módulos LipSoft que contengan líneas I/O en el puerto paralelo indicado en la propiedad *Puerto*. La búsqueda es ascendente a partir del valor actual en la propiedad *Dirección*. Si se encuentra algún módulo se actualizará automáticamente los valores en *Dirección* y *Puerto*. Si no se encuentra ningún módulo LipSoft se producirá un error.

Con la propiedad desactivada (*Test*=0 OFF) se evita que se produzca mensajes de error al intentar acceder a propiedades, que por no estar conectado ningún módulo LipSoft, provocan un error por falta de comunicación.

Tag e Index

Son propiedades estándar de Visual Basic.

Top y Left

Son propiedades estándar que solo tienen efecto en tiempo de diseño ya que el control es no visible en tiempo de ejecución.

NO VISIBLES

(Ninguna)

III.2.3.3 Eventos

(Ninguno)

III.2.3.4 Mensajes de error

La lista siguiente indica el número de error y su significado devueltos por el control DIG.

Nº Error	Mensaje y Explicación
380	Invalid property value Se produce al tratar de asignar a una propiedad un valor fuera del rango o tipo.
32110	Autodetección Se genera este error cuando se trata de buscar algún módulo LipSoft en el puerto especificado y no se encuentra. Falla la comunicación.
32111	Lectura Se produce al intentar leer en el

32112

grupo de líneas I/O del módulo LipSoft direccionado.

Escritura

Se produce al intentar escribir un valor en el grupo de líneas I/O del módulo LipSoft direccionado.

Nota: Los errores del 32110 en adelante se produce cuando no se consigue establecer comunicación con el módulo LipSoft correspondiente (direccionado). Esto puede deberse a tres causas fundamentales:

- a) No se ha direccionado convenientemente (Seleccione bien las propiedades Puerto y Dirección).
- b) Si tiene conectada en el extremo final de la cadena de módulos LipSoft una impresora, ésta debe de estar encendida y con papel.
- c) Fallo eléctrico en el módulo o el puerto Centronics. Pruebe con otro puerto o con otro módulo.

control crea un temporizador propio (Timer) con intervalo=*Periodo*. Cada vez que ha transcurrido el tiempo fijado en intervalo entra en funcionamiento la rutina encargada de leer la información del grupo de canales correspondientes y traspasarla a los Buffers.

c) *Muestras*. Mediante este parámetro se indica el número total de muestras que se han de tomar, espaciadas en el tiempo un intervalo=*Periodo*.

d) Método de disparo, o inicio de la adquisición. En esta propiedad se fija la condición para el comienzo de la adquisición. Existen dos posibilidades, por Software y por Nivel. El disparo por Software hace que se inicie el proceso de adquisición en el instante en que se da la orden por programa, haciendo la propiedad *Acción*=1. Si se elige la opción de disparo por nivel el proceso real de adquisición no comienza hasta que no se ha alcanzado un determinado valor en uno de los canales analógicos (el canal programado como canal de disparo).

III.2.4 Control de Usuario AD8B



III.2.4.1 Descripción

Este control de usuario permite gestionar los módulos LipSoft que contienen canales Analógicos/Digitales de 8 Bits de resolución. Estos son: LipSoft-ADQ, LipSoft-ADT y LipSoft-AD8.

El control dispone de un conjunto de propiedades que permiten programar el proceso de adquisición de datos analógicos de cualquiera de los canales con que cuenta los módulos LipSoft-ADQ, ADT, AD8. Debe tener presente que cada instancia del control AD8B permite gestionar 4 canales A/D y un canal D/A, por tanto, si utiliza un módulo AD8, será preciso que inserte en su Form dos instancias de AD8B, asignando a cada una la dirección programada físicamente en el módulo.

Como en todo proceso de adquisición es preciso definir varios parámetros:

- a) Canal o grupo de canales desde donde se capturarán los datos. Los valores digitales adquiridos se almacenarán automáticamente en cuatro arrays denominados *Buffer_ADx* (que corresponden a propiedades no visibles). Estos arrays mantienen secuencialmente el valor convertido, por ejemplo Ad8b1.Buffer_AD2[13] contiene la muestra 14ª del canal analógico 2.
- b) Tiempo entre cada muestra, o lo que es igual *Periodo*. Este parámetro indica el lapso de tiempo que transcurrirá entre cada muestra adquirida. En su funcionamiento el

Los módulos LipSoft-ADQ, ADT, AD8 dispone también de un canal de conversión Digital/Analógica. Para conseguir la salida de una señal por este canal es preciso cargar el conjunto de valores a convertir en el array *Buffer_DA[]*. Este array debe contener en cada uno de sus elementos el valor instantáneo que se desea para la señal. En el momento en que se inicie el proceso (haciendo la propiedad *Acción*=1) comenzarán a aparecer en el canal analógico una señal eléctrica proporcional al valor digital almacenado en el array correspondiente a esa muestra. Por ejemplo si se fija un *Periodo*=1000 ms(1 segundo), *Muestras*=20, y en la propiedad *Buffer_DA[10]*=175, transcurridos 10 periodos=10 segundos desde que se active la propiedad *Acción*=1 aparecerá en el canal analógico un valor aproximado de 1.75 Voltios que se mantendrá hasta la siguiente muestra. Cuando se haya generado todas las muestras programadas y si el proceso de adquisición no ha terminado (esto solo es posible en el caso de disparo por nivel) se reiniciará la cuenta desde la primera muestra.

Para indicar al control que se inicie/(finalice prematuramente) el proceso de adquisición se dispone de la propiedad *Acción*. Si se pone a 1 se arranca el proceso, si se hace 0 se termina inmediatamente.

El control dispone además de dos propiedades para direccionar el módulo, estas son *Puerto* y *Dirección*. Mediante la propiedad *Puerto* se indica al control la dirección física del puerto paralelo del ordenador, estas pueden ser en hexadecimal (&H278, &H378 y &H3BC). Mediante la propiedad *Dirección* se indica qué módulo de los 8 posibles es al que se quiere acceder. Tenga en cuenta que es posible mantener varios módulos conectados al mismo

puerto siempre y cuando se hayan programado (mediante puentes) las direcciones de forma correcta.

Nombre del Fichero: AD8B.DLL

Tipo de Objeto: Ad8b

III.2.4.2 Propiedades

El acceso en tiempo de ejecución a las propiedades desde Visual Basic se realiza siempre del mismo modo:

```
[Form.]Ad8b.Propiedad= valor
```

```
valor =[Form.]Ad8b.Propiedad
```

Como todas las propiedades son del tipo numérico el acceso a éstas desde Visual C es siempre idéntico, solo cambia el nombre de la propiedad a la que se accede y el valor. Ejemplo:

```
pAd8b-> GetNumProperty("Propiedad_a_Leer")
```

```
pAd8b-> SetNumProperty("Propiedad_a_Escribir",
valor)
```

Donde **Propiedad** identifica el nombre de la propiedad a leer o escribir y valor es un tipo compatible (entero, entero largo etc..)

VISIBLES (Las que aparecen en la ventana "Properties").

Canal_0, Canal_1, Canal_2, Canal_3 (Entero 0..255)

Estas 4 propiedades mantienen constantemente el valor actual en los canales correspondientes. Son propiedades de solo Lectura. El refresco (actualización de su valor) en tiempo de diseño se realiza cada vez que se activa la propiedad *Test*. Siempre que se lee su valor en tiempo de ejecución se produce un proceso de adquisición de datos y se actualiza su valor. En procesos de adquisición automática (haciendo *Acción*=1 y fijando el *Período* y *Muestras*) NO es conveniente acceder al valor instantáneo del *Canal_x* leyendo estas propiedades ya que lanzaríamos procesos intermedios de adquisición que ralentizarían excesivamente el proceso. La adquisición utilizando estas propiedades tiene en cuenta el *Modo* actual.

Canal_Disparo (0,1)

Identifica el canal que servirá para realizar el disparo por nivel, en el caso de que se haya seleccionado esta opción

en la propiedad *Disparo*. Puede tomar los valores 0,1 correspondientes al 1º y 2º respectivamente.

Dirección (0..7)

Permite seleccionar una de las 8 (0..7) direcciones posible para los canales A/D.

Disparo (0,1)

Se selecciona el tipo de disparo que iniciará el proceso de muestreo. Puede tomar los valores:

0. Software. Cada vez que se activa la propiedad no visible *Acción*=1 se producirá el evento *SOFT*. En el cuerpo de este evento puede realizarse todas las acciones que se consideren oportunas.

1. Nivel. Hace que comience la adquisición cuando se haya alcanzado en el canal seleccionado *Canal_Disparo* el valor ajustado en la propiedad *Nivel_Disparo*. Cuando se ha alcanzado el nivel programado se producirá el evento *NIVEL*.

Finalizado el proceso de adquisición siempre se producirá el evento *FIN*.

Modo (0,1,2,3)

Indica el modo en que funcionarán los 4 canales A/D. Puede tomar los siguientes valores:

0. Cuatro canales simples.

1. Tres canales diferenciales con respecto al canal 3, es decir $ADO=Canal0-Canal3$, $AD1=Canal1-Canal3$, $AD2=Canal2-Canal3$.

2. Dos canales simples y uno diferencial. $ADO=Canal0$, $AD1=Canal1$, $AD2=Canal2-Canal3$.

3. Dos canales diferenciales. $ADO=Canal0-Canal1$, $AD1=Canal2-Canal3$.

*En los canales diferenciales los valores vienen representados en complemento a dos.

*ADx indica la asignación que se realizará en los Buffer's AD donde se guardan los datos muestreados.

Muestras (Entero 1..255)

Permite seleccionar el número de muestras totales a adquirir. Debe ser > 0 y < 256 .

Nivel_Disparo (Entero 0.. 255)

Indica el valor en el canal seleccionado como *Canal_Disparo* a partir del cual comenzará el proceso de

adquisición. Solo tiene efecto si se ha seleccionado el tipo de disparo "1. Nivel".

Período (Entero > 5 < 32000)

Identifica el lapso de tiempo que transcurrirá entre la adquisición de cada muestra en milisegundos. (Atención la resolución es de 55 ms. por tanto un intento de muestrear con una cadencia menor no tendrá efecto).

Puerto (0,1,2,3)

Permite seleccionar uno de los tres puertos de impresora donde pueden estar conectados los módulos que contienen canales A/D (LipSoft-ADT, LipSoft-AD8 o LipSoft-ADQ). Los valores pueden ser:

0. Autodetección. Hace que al disparar la propiedad *Test* se busque en todos los posibles puertos de impresora del ordenador hasta encontrar una un módulo que contenga canales A/D.

1.&H378 Corresponde generalmente a LPT1:

2.&H278 Corresponde normalmente a LPT2:

3.&H3BC

Salida_DA(Entero 0..255)

Esta propiedad, de solo escritura, hace que aparezca inmediatamente en el canal Digital/Analógico una tensión proporcional (1 unidad \approx 10mV) al valor asignado.

Test(0=OFF, 1=ON)

Al dispararlo (pasar de OFF=0 a ON=1) se produce un proceso de autodetección de módulos LipSoft que contengan canales A/D en el puerto paralelo indicado en la propiedad *Puerto*. La búsqueda es ascendente a partir del valor actual en la propiedad *Dirección*. Si se encuentra algún módulo se actualizará automáticamente los valores en *Dirección* y *Puerto*. Si no se encuentra ningún módulo LipSoft se producirá un error.

Con la propiedad desactivada (*Test*=0 OFF) se evita que se produzca mensajes de error al intentar acceder a propiedades que por no estar conectado ningún módulo LipSoft provocan un error por falta de comunicación, por ejemplo al tratar de acceder al valor en cada *Canal_x*.

Tag e Index

Son propiedades estándar de Visual Basic.

Top y Left

Son propiedades estándar que solo tienen efecto en tiempo de diseño ya que el control es no visible en tiempo de ejecución.

NO VISIBLES

Acción. (0,1)

Esta propiedad permite disparar o detener el proceso de adquisición. Puede tomar los valores:

0. Termina el proceso de adquisición en curso.

1. Inicia el proceso de adquisición.

Buffer_DA(j%).

En esta matriz se almacenan los valores que se desean aparezcan en el canal de salida Digital /Analógico. A partir del momento en que se dispare la adquisición mediante la propiedad *Acción*=1 comenzarán a aparecer sincronizadamente con el período de muestreo uno a uno los valores almacenados en esta propiedad. Se admiten solo valores enteros entre 0..255 -> 0 a 2,5 Voltios.

Buffer_ADx(j%).

Estas matrices almacenan los valores captados en los canales $x=0,1,2,3$ cada período de muestreo. Los valores almacenados serán enteros (0..255).

III.2.4.3 Eventos

FIN

Este evento se produce cada vez que se han adquirido todas las muestras indicada en la propiedad *Muestras*.

NIVEL

Se lanza este evento cuando se ha alcanzado en el canal seleccionado como canal de disparo el nivel establecido en la propiedad *Nivel_Disparo* y siempre y cuando se haya seleccionado como técnica de disparo "Nivel" en la propiedad *Disparo*.

SOFT

Se produce este evento cada vez que se inicia un proceso de conversión haciendo la propiedad *Acción*=1 y se haya seleccionado como método de disparo "Disparo=0 Software".

III.2.4.4 Mensajes de Error

La lista siguiente indica el número de error y su significado devueltos por el control AD8B.

Nº Error	Mensaje y Explicación
380	Invalid property value Se produce al tratar de asignar

	a una propiedad un valor fuera del rango o tipo.
381	Invalid property array index Se genera cuando se comete un error en el índice durante el acceso a los Buffer_ADx() o Buffer_DA()
32100	No hay TIMERS disponibles Este error se produce cuando se solicita a Windows la creación de un nuevo temporizador para la gestión de un proceso de muestreo.
32101	Autodetección Se genera este error cuando se trata de buscar algún módulo LipSoft en el puerto especificado y no se encuentra. Falla la comunicación.
32102	Escritura Se produce al intentar escribir un valor en el canal D/A del módulo LipSoft direccionado.
32103	Lectura Se produce al intentar leer los canales A/D del módulo LipSoft direccionado.
32104	Poner Modo Se produce al intentar ajustar un nuevo modo de trabajo para los canales A/D del módulo LipSoft direccionado.

Nota: Los errores del 32101 en adelante se produce cuando no se consigue establecer comunicación con el módulo LipSoft correspondiente (direccionado). Esto puede deberse a tres causas fundamentales:

- No se ha direccionado convenientemente (Seleccione bien las propiedades Puerto y Dirección).
- Si tiene conectada en el extremo final de la cadena de módulos LipSoft una impresora, ésta debe de estar encendida y con papel.
- Fallo eléctrico en el módulo o el puerto Centronics. Pruebe con otro puerto o con otro módulo.

III.2.5 Control de Usuario IRQ



III.2.5.1 Descripción

Este control permite gestionar el manejo de interrupciones hardware redireccionadas a través de la línea de interrupción asignada por el PC a la impresora. Para conseguir una comunicación Ordenador-Impresora más eficiente, se suele establecer un canal de interrupción. Éste suele estar asignado a las líneas IRQ7 o IRQ5 del controlador de interrupciones del PC. Mediante los módulos LipSoft se dispone de acceso directo (hardware) a esta línea de interrupción. Por ejemplo, el módulo LipSoft-ADQ lleva cableado internamente la INT de las línea I/O digitales a la interrupción. Otros módulos ofrecen al exterior esta línea para que pueda ser cableada según las necesidades (LipSoft-ADT) (ver información técnica de los módulos).

Al disponer de una única línea de interrupción por puerto de impresora (IRQ5 o IRQ7) todas las interrupciones de los módulos que se deseen manejar deben ir cableadas juntas. En cuanto se produzca un **pulso** a nivel bajo en la línea de interrupción, siempre y cuando el control esté activado, se producirá un evento *Recibida*. En el código de atención de este evento debe disponer las funciones convenientes para la atención a dicho suceso. En cualquier caso y al estar todas unidas deberá encontrar cual de los módulos LipSoft conectados ha generado la interrupción (Contadores o Líneas I/O digitales) a partir de aquí procederá como crea más conveniente.

Nombre del Fichero: IRQ.VBX

Tipo de Objeto: Irq

III.2.5.2 Propiedades

El acceso en tiempo de ejecución a las propiedades desde Visual Basic se realiza siempre del mismo modo:

[Form.]Irq.**Propiedad**= valor

valor =[Form.]Irq.**Propiedad**

Como todas las propiedades son del tipo numérico el acceso a éstas desde Visual C es siempre idéntico, solo cambia el nombre de la propiedad a la que se accede y el valor. Ejemplo:

```
plrq-> GetNumProperty("Propiedad_a_Leer")
```

```
plrq-> SetNumProperty("Propiedad_a_Escribir",  
valor)
```

Donde **Propiedad** identifica el nombre de la propiedad a leer o escribir y valor es un tipo compatible (entero, entero largo etc..)

VISIBLES (Las que aparecen en la ventana "Properties").

Habilitar (TRUE , FALSE)

Esta propiedad permite habilitar o deshabilitar la atención a la interrupción IRQ seleccionada. La atención a esta interrupción produce el evento *Recibida*.

Puerto (0,1,2)

Permite seleccionar el puerto de impresora donde existe un canal de interrupción. Los valores pueden ser 0-&H378, 1-&H278, 2-&H3BC que generalmente corresponden a LPT1, LPT2, LPT3.

Irq (0 = irq7, 1=irq5)

Esta propiedad permite seleccionar el nivel de la interrupción a atender. Puede tomar los valores:

0. IRQ7. Interrupción hardware normal para el puerto de impresora primario (LPT1)

1. IRQ5. Interrupción hardware normal para el puerto de impresora secundario (LPT2)

Para realizar una adecuada selección del canal de interrupción es conveniente que consulte el manual hardware de su equipo o de la tarjeta que contiene el puerto paralelo. Utilidades tales como PC-TOOLS o NORTON le permitirán detectar también como están asignadas estas interrupciones hardware.

Tag e Index

Son propiedades estándar de Visual Basic.

Top y Left

Son propiedades estándar que solo tienen efecto en tiempo de diseño ya que el control es no visible en tiempo de ejecución.

III.2.5.3 Eventos

Recibida

Este evento se produce cada vez que se produce un pulso a nivel bajo en la línea de interrupción, siempre y cuando se encuentren habilitadas la interrupciones.

III.2.5.4 Mensajes de Error

La lista siguiente indica el número de error y su significado devueltos por el control IRQ.

Nº Error	Mensaje y Explicación
380	Invalid property value Se produce al tratar de asignar a una propiedad un valor fuera del rango o tipo.

III.3 Librerías de enlace dinámico (DLL) (Windows 3.x)

En los siguientes apartados se van a describir el conjunto de funciones disponibles en las librerías que acompañan a los módulos *LipSoft* y que permiten la comunicación entre éstos y programas desarrollados para el entorno Windows.

Como debe ser conocido por el usuario, bajo el entorno operativo Windows existe un procedimiento estandarizado de comunicación entre lenguajes de programación. Esto se realiza utilizando llamadas a funciones incluidas en librerías de enlace dinámico (DLL). Independientemente del lenguaje de programación (C, Pascal, Basic etc..) que utilice para desarrollar su aplicación Windows, todos soportan llamadas a funciones y procedimientos en DLL's. Este tipo de librerías tienen una estructura propia, a la vez que emplean una técnica para el traspaso de parámetros. Concretamente las funciones son del tipo FAR PASCAL (modelo de memoria largo y paso de parámetros tipo PASCAL). Debe revisar los manuales de programación de su lenguaje para obtener una información más detallada de como realizar llamadas a funciones en DLL's.

En cada función descrita se indicarán los parámetros transferidos y su significado así como el proceso realizado por la función. Si se invoca a una función con los parámetros incorrectos es posible que Windows emita un error crítico. En tal caso cierre la aplicación vuelva al DOS y vuelva a entrar en Windows.

Se muestran las declaraciones para los lenguajes de programación para Windows más extendidos (Visual Basic, Borland Pascal, Borland C, MicroSoft C).

NOTA:

Es preciso que el fichero correspondiente (*.DLL) se incluya en su producto para que usuario final disponga de acceso a las funciones.

En el caso de programación en "C" deberá indicar en el fichero de definición de módulos (*.DEF) las funciones utilizadas.

Si programa bajo Visual Basic y utiliza llamadas a las funciones de la librería, junto con asignaciones de propiedades del Custom Control, tenga en cuenta que éstas llamadas pueden alterar, sin que Visual Basic lo detecte, el estado de funcionamiento de los módulos y producirse un mal funcionamiento. Por ejemplo si mediante el CC AD8B ajusta la propiedad Modo=0 y por medio de una llamada a la función Poner_Modo_Ad8b cambia a otro, el módulo quedará funcionando con ese nuevo Modo hasta que Visual Basic reasigne esta propiedad.

III.3.1 Librerías AD12B.DLL* o AD12R.DLL

Esta librería contiene el conjunto de funciones que permite el acceso desde su programa al módulo de adquisición Analógico/Digital LipSoft-AD12 (* para la versión hardware antigua del módulo). Este sistema de adquisición dispone de 8 canales A/D y un canal D/A ambos de 12 bits de resolución.

III.3.1.1 Funciones

Declaración:

Vis. Basic *Declare Function Escribir_Ad12? Lib "AD12?.DLL" (ByVal Dato%, ByVal Puerto%, ByVal Dirección%) as integer*

C *int FAR PASCAL Escribir_Ad12?(int Dato, int Puerto, int Dirección);*

Pascal *function Escribir_Ad12?(Dato:Integer, Puerto:Integer, Dirección:Integer) Integer;*

donde ?=b o r según la librería (b para la hardware antigua, r para la versión actual)

Descripción:

Esta función se utiliza para realizar la conversión de una palabra digital *Dato*, en una tensión analógica de salida a través del canal D/A (pin 16 del conector de LipSoft-AD12). Cada unidad equivale a 1 milivoltio. Por ejemplo Dato=2375 equivale a 2'375 Voltios =2375 mV. El rango de valores admisibles va de 0 a 4095.

Puerto identifica la dirección física del puerto de datos, siendo lo normal &H378, &H278 o &H3BC (que generalmente corresponden a LPT1:, LPT2:)

Dirección indica la dirección asignada al módulo LipSoft-AD12. Estas direcciones se fijan puenteando los pines 17 y 19 con los correspondientes 4 a 14 del conector de salida (ver documentación técnica del módulo). Los valores posibles son 0, 1 o 2.

Declaración:

Vis. Basic *Declare Function Leer_Ad12? Lib "AD12?.DLL" (ch0%, ch1%, ch2%, ch3%, ByVal Puerto%, ByVal Dirección%, ByVal Modo%, ByVal Conjunto%) as integer*

C *int FAR PASCAL Leer_Ad12? (int *ch0, int *ch1, int *ch2, int *ch3, int Puerto, int Dirección, int Modo, int Conjunto);*

PASCAL *function Leer_Ad12? (var ch0, ch1, ch2, ch3:integer, Puerto, Dirección, Modo, Conjunto : integer)integer;*

donde ?=b o r según la librería (b para la hardware antigua, r para la versión actual)

Descripción:

Esta función se emplea para leer el valor digitalizado actual en 4 de los 8 canales de conversión analógicos/digitales que contiene el módulo. Se le pasan por referencia las variables ch0, ch1, ch2 y ch3 que tras la llamada contendrán los valores convertidos de los correspondientes canales. Dependiendo del valor del parámetro *Conjunto* estos corresponderán a:

Conjunto=0 : ch0=Canal 0, ch1=Canal 1, ch2=Canal 2, ch3=Canal 3

Conjunto=1 : ch0=Canal 4, ch1=Canal 5, ch2=Canal 6, ch3=Canal 7

Modo hace que el valor correspondiente leído en los canales sea Simple (*Modo=0*) o Diferencial (*Modo=1*). En el modo Simple los valores leídos corresponde a las tensiones que existen entre el canal y masa (0 Voltios). En modo Diferencial el valor digitalizado corresponde a la diferencia entre los canales ch0-ch4, ch1-ch5, ch2-ch6 y ch3-ch7 y por tanto podrá tomar valores negativos. Además, en este modo, el parámetro *Conjunto* no tiene significado.

Puerto identifica la dirección física del puerto de datos, siendo lo normal &H378, &H278 o &H3BC (que generalmente corresponde a LPT1:, LPT2:)

Dirección indica la dirección asignada al módulo LipSoft-AD12. Estas direcciones se fijan puenteando los pines 17 y 19 con los correspondientes 4 a 14 del conector de salida (ver documentación técnica del módulo). Los valores posibles son 0, 1 o 2.

III.3.2 Librería AD8B.DLL

Esta librería contiene el conjunto de funciones que permite manejar los subsistemas de conversión Analógico/Digital y Digital/Analógico que contienen los módulos LipSoft-ADQ, LipSoft-ADT, LipSoft-AD8. Estos subsistemas presenta una resolución de 8 bits conteniendo los dos primeros un conversor A/D de 4 canales y un D/A y el último dos conversores A/D de 4

canales y dos D/A. Se disponen de funciones que permiten la lectura (del valor convertido en los canales) la escritura (de un valor en el canal D/A) y de funciones auxiliares para el direccionamiento y test.

III.3.2.1 Funciones

Declaración:

Vis. Basic *Declare Function Autodeteccion_Ad8b Lib "AD8B.DLL" (Puerto% , Dirección%) as integer*

C *int FAR PASCAL Autodeteccion_Ad8b (int *Puerto, int *Dirección);*

PASCAL *function Autodeteccion_Ad8b (var Puerto, Dirección: integer) integer;*

Descripción:

Esta función permite detectar la presencia de un subsistema A/D de un módulo LipSoft en un determinado puerto de impresora. Si el proceso tiene éxito devolverá en *Puerto* y *Dirección* los valores correspondientes al *Puerto* (&H378, &H278 o &H3BC) y *Dirección* (0 a 7) donde se encuentra. Si se llama a la función con un valor en *Puerto=0*, ésta buscará en todos los puertos y en direcciones ascendentes a partir de la dirección con que se llamó. Así para obligar a una búsqueda completa conviene que se llame a la función con *Puerto=0* y *Dirección=0* inicialmente. Si la llamada tiene éxito la función devuelve 0 en caso contrario (-1)*.

Declaración:

Vis Basic *Declare Function Escribir_Ad8b Lib "AD8B.DLL" (ByVal Valor%, ByVal Puerto%, ByVal Dirección%) as integer*

C *int FAR PASCAL Escribir_Ad8b (int Valor, int Puerto, int Dirección);*

PASCAL *function Escribir_Ad8b (Valor, Puerto, Dirección:integer) integer;*

Descripción:

Esta función permite escribir un valor (0..255 -> 0 a 2'5 volt) en el canal D/A direccionado mediante *Puerto* y *Dirección*. *Puerto* puede tomar los valores en hexadecimal &H378, &H278 o &H3BC y *Dirección* (0..7).

Si la llamada tiene éxito la función devuelve 0 en caso contrario (-1)*.

Declaración:

Vis Basic *Declare Function Leer_Ad8b Lib "AD8B.DLL" (ch0%, ch1%, ch2%, ch3%, ByVal Puerto%, ByVal Dirección%) as integer*

C *int FAR PASCAL Leer_Ad8b (int *ch0, int *ch1, int *ch2, int *ch3, int Puerto, int Dirección);*

PASCAL *function Leer_Ad8b (var ch0, ch1, ch2, ch3 : integer, Puerto, Dirección:integer) integer;*

Descripción:

Esta función se emplea para leer el valor digitalizado actual en los 4 canales de conversión analógicos/digitales del subsistema. Se le pasan por referencia las variables ch0, ch1, ch2 y ch3 que tras la llamada contendrán los valores convertidos de los correspondientes canales.

Según el *Modo* actual (ver función Poner_Modo_Ad8b) los valores leídos tendrán distintos significados.

En cuanto a los parámetros *Puerto* y *Dirección* tienen el mismo significado que en la función anterior.

Si la llamada tiene éxito la función devuelve 0 en caso contrario (-1)*.

3. Dos canales diferenciales. ch0=Canal0-Canal1, ch1=Canal2-Canal3.

*En los canales diferenciales los valores leídos vendrán representados en complemento a dos.

Los parámetros *Puerto* y *Dirección* tienen el mismo significado que en la funciones previas.

Si la llamada tiene éxito la función devuelve 0 en caso contrario (-1)*.

* La función puede fallar si no se establece comunicación con el módulo LipSoft debido a un mal direccionamiento (consultar documentación técnica de los módulos).

III.3.3 Librería DIG.DLL

En esta librería se incluyen las funciones necesarias para el control de las líneas Entrada/Salida (I/O) digital que contienen los módulos LipSoft-ADQ, LipSoft-ADT, LipSoft-DIG. Estos contienen 1, 1 y 2 grupos de 8 líneas I/O cuasibidireccionales respectivamente (ver documentación técnica).

III.3.3.1 Funciones

Declaración:

Vis Basic *Declare Function Poner_Modo_Ad8b Lib "AD8B.DLL" (ByVal Modo%, ByVal Puerto%, ByVal Dirección%) as integer*

C *int FAR PASCAL Poner_Modo_Ad8b (int Modo, int Puerto, int Dirección);*

PASCAL *function Poner_Modo_Ad8b (Modo, Puerto, Dirección :integer) integer;*

Descripción:

Esta función se utiliza para modificar el *Modo* actual de funcionamiento del convertidor A/D y D/A direccionado mediante *Puerto* y *Dirección*. Los valores posibles para *Modo* son:

0. Cuatro canales simples. ch0=Canal0, ch1=Canal1, ch2=Canal2, ch3=Canal3

1. Tres canales diferenciales con respecto al canal 3, es decir ch0=Canal0-Canal3, ch1=Canal1-Canal3, ch2=Canal2-Canal3.

2. Dos canales simples y uno diferencial. ch0=Canal0, ch1=Canal1, ch2=Canal2-Canal3.

Declaración:

Vis Basic *Declare Function Autodeteccion_Dig Lib "DIG.DLL" (Puerto%, Dirección%) as integer*

C *int FAR PASCAL Autodeteccion_Dig (int *Puerto, int *Dirección);*

PASCAL *function Autodeteccion_Dig (var Puerto, Dirección : integer) integer;*

Descripción:

Esta función permite detectar la presencia de un grupo I/O (conjunto de 8 líneas Entrada/Salida) de un módulo LipSoft conectado en un determinado puerto de impresora. Si el proceso tiene éxito, devolverá en *Puerto* y *Dirección* los valores correspondientes al *Puerto* (&H378, &H278 o &H3BC) y *Dirección* (0 a 7) donde se encuentra. Si se llama a la función con un valor en *Puerto*=0 ésta buscará en todos los puertos y en direcciones ascendentes a partir de la *Dirección* con que se llamó. Por tanto, para obligar a una búsqueda completa, conviene que se llame a la función con *Puerto*=0 y *Dirección*=0 inicialmente.

Si la llamada tiene éxito la función devuelve 0 en caso contrario (-1)*.

Declaración:

Vis Basic *Declare Function Escribir_Dig Lib "DIG.DLL" (ByVal Valor%, ByVal Puerto%, ByVal Dirección%) as integer*

C *int FAR PASCAL Escribir_Dig (int Valor, int Puerto, int Dirección);*

PASCAL *function Escribir_Dig (Valor, Puerto, Dirección :integer) integer;*

Descripción:

Esta función permite escribir un *Valor* (0..255) en un grupo I/O digital direccionado mediante *Puerto* y *Dirección*. Si un bit de *Valor* se pone a 1 la línea correspondiente podrá actuar como entrada, de modo que si un elemento exterior fuerza la línea a 0 (0 Voltios) el valor que se leerá en el bit correspondiente mediante la función de lectura corresponderá al valor que realmente existe en la línea.

Puerto corresponde a la dirección física del puerto de impresora (paralelo) donde se encuentra conectado el módulo LipSoft. Puede tomar los valores en hexadecimal &H378, &H278 o &H3BC.

Dirección es un entero 0..7 correspondiente a las 8 direcciones físicas de los grupos I/O (ver documentación técnica).

Si la llamada tiene éxito la función devuelve 0 en caso contrario (-1)*.

Declaración:

Vis Basic *Declare Function Leer_Dig Lib "DIG.DLL" (ByVal Puerto%, ByVal Dirección%) as integer*

C *int FAR PASCAL Leer_Dig (int Puerto, int Dirección);*

PASCAL *function Leer_Dig (Puerto, Dirección :integer) integer;*

Descripción:

Esta función permite la lectura del estado actual de un grupo I/O direccionado mediante *Puerto* y *Dirección*. Si la llamada tiene éxito la función devuelve el valor actual del estado de las líneas (un entero). Si una línea se encuentra a nivel bajo (0 Volt) el bit correspondiente será cero, en caso contrario 1. Si se presenta

algún problema en la comunicación la función devuelve -1*.

El significado de *Puerto* y *Dirección* es idéntico a la función anterior.

* La función puede fallar si no se establece comunicación con el módulo LipSoft debido a un mal direccionamiento (consultar documentación técnica de los módulos).

III.3.4 Librería CONT.DLL

Esta librería contiene las funciones precisas para el manejo de los contadores que incluyen los módulos LipSoft-ADT y LipSoft-IOT. El primero contiene 1 contador y el segundo 2 contadores respectivamente, ambos de 6 décadas (0..999999). La librería contiene funciones para la lectura y escritura de los contadores, para iniciar o parar el contaje, así como funciones para programar las alarmas.

III.3.4.1 Funciones

Declaración:

Vis Basic *Declare Function Autodeteccion_Cont Lib "CONT.DLL" (Puerto%, Dirección%) as integer*

C *int FAR PASCAL Autodeteccion_Cont (int *Puerto, int *Dirección);*

PASCAL *function Autodeteccion_Cont (var Puerto, Dirección : integer) integer;*

Descripción:

Esta función permite detectar la presencia de un subsistema contador de un módulo LipSoft conectado en un determinado puerto de impresora. Si el proceso tiene éxito devolverá en *Puerto* y *Dirección* los valores correspondientes al *Puerto* (&H378, &H278 o &H3BC) y *Dirección* (0 o 1) donde se encuentra. Si se llama a la función con un valor en *Puerto*=0 ésta buscará en todos los puertos y en direcciones ascendentes a partir de la *Dirección* con que se llamó. Por tanto, para obligar a una búsqueda completa, conviene que se llame a la función con *Puerto*=0 y *Dirección*=0 inicialmente.

Si la llamada tiene éxito la función devuelve 0 en caso contrario (-1)*.

Declaración:

Vis Basic *Declare Function Leer_Cuenta Lib "CONT.DLL" (ByVal Puerto% , ByVal Dirección%) as long*

C *long int FAR PASCAL Leer_Cuenta (int Puerto , int Dirección);*

PASCAL *function Leer_Cuenta (Puerto , Dirección:integer) longint;*

Descripción:

Esta función devuelve el estado de la cuenta actual almacenada en el contador direccionado mediante los parámetros *Puerto* y *Dirección*. Si falla la comunicación el valor devuelto será -1*.

Puerto corresponde a la dirección física del puerto de impresora (paralelo) donde se encuentre conectado el módulo LipSoft. Puede tomar los valores en hexadecimal &H378, &H278 o &H3BC.

Dirección es un entero 0 o 1 correspondiente a las 2 direcciones físicas que pueden programarse en los subsistemas contadores (ver documentación técnica).

Declaración:

Vis Basic *Declare Function Escribir_Cuenta Lib "CONT.DLL" (ByVal InicioCuenta as Long , ByVal Puerto% , ByVal Dirección%) as integer*

C *int FAR PASCAL Escribir_Cuenta (long InicioCuenta, int Puerto , int Dirección);*

PASCAL *function Escribir_Cuenta (InicioCuenta: longint, Puerto , Dirección:integer) integer;*

Descripción:

Esta función permite escribir un valor cualquiera (InicioCuenta entero largo entre 0 y 999999) en el contador direccionado mediante *Puerto* y *Dirección*. Este valor será el de partida en el proceso de contaje y se irá incrementando en una unidad cada vez que a la entrada del contador se introduzca un pulso (cambio de nivel lógico bajo a alto). Generalmente se utiliza para poner a 0 el contador.

El significado de los parámetros *Puerto* y *Dirección* son idénticos a los de la función anterior

Si la llamada tiene éxito la función devuelve 0 en caso contrario (-1)*.

Declaración:

Vis Basic *Declare Function Parar_Contaje Lib "CONT.DLL" (ByVal Puerto% , ByVal Dirección%) as integer*

C *int FAR PASCAL Parar_Contaje (int Puerto, int Dirección);*

PASCAL *function Parar_Contaje (Puerto, Dirección : integer) integer;*

Descripción:

Esta función sirve para detener el contaje de los pulsos que lleguen al contador direccionado mediante *Puerto* y *Dirección*. A partir del momento en que se invoque la función los pulsos que le lleguen al contador no incrementarán la cuenta. Para reactivar el contaje utilice la función *Activar_Contaje*.

El significado de los parámetros *Puerto* y *Dirección* son idénticos a los de la función anterior.

Si la llamada tiene éxito la función devuelve 0 en caso contrario (-1)*.

Declaración:

Vis Basic *Function Activar_Contaje Lib "CONT.DLL" (ByVal Puerto% , ByVal Dirección%) as integer*

C *int FAR PASCAL Activar_Contaje (int Puerto, int Dirección);*

PASCAL *function Activar_Contaje (Puerto, Dirección : integer) integer;*

Descripción:

Esta función sirve para habilitar el contaje de los pulsos que lleguen al contador direccionado mediante *Puerto* y *Dirección*. A partir del momento en que se invoque la función los pulsos que le lleguen al contador incrementarán la cuenta. Para detener el contaje utilice la función *Parar_Contaje*. El significado de los parámetros *Puerto* y *Dirección* son idénticos a los de la función anterior.

Si la llamada tiene éxito la función devuelve 0 en caso contrario (-1)*.

Declaración:

Vis Basic *Declare Function Parar_Alarma Lib "CONT.DLL" (ByVal Puerto%, ByVal Dirección%) as integer*

C *int FAR PASCAL Parar_Alarma (int Puerto, int Dirección);*

PASCAL *function Parar_Alarma (Puerto, Dirección : integer) integer;*

Descripción:

Esta función sirve para Deshabilitar la alarma. La alarma, activación a nivel bajo de la línea "INT" de los contadores, se produce cuando el valor almacenado en el contador (cuenta) es igual al programado en el registro de alarma. La igualdad para que la alarma se active, debe alcanzarse a través de los pulsos de contaje, no se activa ninguna alarma por el simple hecho de escribir el mismo valor en los registros de alarma y cuenta. Para habilitar la alarma utilice la función Activar_Alarma.

El significado de los parámetros *Puerto* y *Dirección* son idénticos a los de la función anterior.

Si la llamada tiene éxito la función devuelve 0 en caso contrario (-1)*.

Declaración:

Vis Basic *Declare Function Activar_Alarma Lib "CONT.DLL" (ByVal Puerto%, ByVal Dirección%) as integer*

C *int FAR PASCAL Activar_Alarma (int Puerto, int Dirección);*

PASCAL *function Activar_Alarma (Puerto, Dirección: integer) integer;*

Descripción:

Esta función sirve para Habilitar la alarma. La alarma, activación a nivel bajo de la línea "INT" de los contadores, se produce cuando el valor almacenado en el contador (cuenta) es igual al programado en el registro de alarma. La igualdad para que la alarma se active, debe alcanzarse a través de los pulsos de contaje, no se activa ninguna alarma por el simple hecho de escribir el mismo valor en los registros de alarma y cuenta. Para deshabilitar la alarma utilice la función Parar_Alarma.

El significado de los parámetros *Puerto* y *Dirección* son idénticos a los de la función anterior.

Si la llamada tiene éxito la función devuelve 0 en caso contrario (-1)*.

Declaración:

Vis Basic *Declare Function Escribir_Alarma Lib "CONT.DLL" (ByVal MiAlarma as long , ByVal Puerto%, ByVal Dirección%) as integer*

C *int FAR PASCAL Escribir_Alarma (long int MiAlarma, int Puerto, int Dirección);*

PASCAL *function Escribir_Alarma (MiAlarma: longint, Puerto, Dirección: integer) integer;*

Descripción:

Esta función permite escribir un valor (*MiAlarma* entero largo 0..999999) en el registro de alarma. Si la alarma se activa (función Activar_Alarma) se pondrá a nivel bajo (0 Volt) la línea "INT" del contador cuando el contador reciba un pulso que haga que cuenta igual a *MiAlarma*.

El significado de los parámetros *Puerto* y *Dirección* son idénticos a los de la función anterior.

Si la llamada tiene éxito la función devuelve 0 en caso contrario (-1)*.

* La función puede fallar si no se establece comunicación con el módulo LipSoft debido a un mal direccionamiento (consultar documentación técnica de los módulos).

III.4 Librería de enlace dinámico DLL de 32 bits para Windows 95 y NT

En este apartado se van a describir las funciones disponibles en la librería LIPS_32.DLL que componen el soporte para 32 bits.

Todas las funciones se han incluido dentro de un único archivo (LIPS_32.DLL) que deberá copiarse bien en el directorio \SYSTEM\ de Windows o en el directorio desde donde se ejecute su aplicación.

El modo de operación, parámetros etc. de las nuevas funciones es idéntico a la versión para 16 bits y se remitirá en su descripción a la función de 16 bits correspondiente.

Para hacer funcionar los módulos LipSoft en aplicaciones Windows NT es preciso instalar y registrar el controlador de dispositivos "LSDRIVER.SYS". Para ello, deberá copiar el archivo "LSDRIVER.SYS" al subdirectorio "\system32\drivers" de Windows NT. Además, deberá crear las entradas en el REGISTRO de Windows NT tal como se relaciona en "LSDRIVER.INI". Para facilitar este proceso se incluye el programa "INS_LIPS.EXE" mediante el cual se copiará "LSDRIVER.SYS" al sitio adecuado y se registrará el controlador.

Tras la instalación deberá reiniciar el equipo para que el controlador quede operativo.

Las funciones en la librería "LIPS_32.DLL" trabajan en NT y en Windows 95, internamente realizan una comprobación para detectar el sistema bajo el cual funcionan.

La aplicación Visual Basic 32 bits que se encuentra en el subdirectorio DEMO32 funciona en NT y en Windows 95.

III.4.1 Normas para utilizar las funciones

Dependiendo del lenguaje de programación que utilice deberá ubicar en un lugar apropiado el fichero LIPS_32.DLL (generalmente en el directorio \SYSTEM de Windows o en su directorio de ejecutables).

No olvide que deberá incluir una copia de la librería cuando distribuya su software.

No existe un procedimiento general para poder utilizar las funciones de la DLL desde su lenguaje de programación, pues

cada lenguaje suele utilizar un método propio. Como norma general tenga en cuenta que las funciones de la librería utilizan la misma convención en las llamadas que las funciones de la API de Windows. Se remite al usuario a la documentación de su lenguaje donde haga referencia a llamadas a la API de Windows.

Todas las funciones utilizan el procedimiento estándar de comunicación para Windows 32 bits que se traduce en:

*Se hace distinción entre mayúsculas y minúsculas en el nombre de las funciones.

*Por defecto los parámetros se pasan por valor.

*Se ha de utilizar el indicador `_stdcall` para que la pila se descargue al terminar la función.

Por lo general las funciones devuelven un entero de 16 bits que normalmente se define como short.

III.4.2 Funciones disponibles

III.4.2.1 Funciones para el módulo LipSoft - AD12

Declaración:

Visual Basic 4.0 32 bits

```
Public Declare Function Leer_Ad12r32 Lib  
"LIPS_32.DLL" (ch0%, ch1%, ch2%, ch3%, ByVal  
Puerto%, ByVal Direccion%, ByVal Modo%, ByVal  
Conjunto%) As Integer
```

```
Public Declare Function Escribir_Ad12r32 Lib  
"LIPS_32.DLL" (ByVal Valor%, ByVal Puerto%,  
ByVal Direccion%) As Integer
```

C (Borland)

```
short _stdcall Escribir_Ad12r32(short dato,short  
Puerto, short Dir);
```

```
short _stdcall Leer_Ad12r32(SHORT FAR  
*ch0,SHORT FAR *ch1,SHORT FAR *ch2,SHORT  
FAR *ch3,short Puerto, short Dir,short Modo,short  
Conj);
```

Estas funciones equivalen a sus homólogas de la librería AD12r.DLL

III.4.2.2 Funciones para el subsistema analógico-digital de 8 bits que contienen los módulos ADQ, ADT, AD8 y AD2.

Declaración:

Visual Basic 4.0 32 bits

```
Public Declare Function Leer_Ad8b32 Lib
"LIPS_32.DLL" (ch0%, ch1%, ch2%, ch3%, ByVal
Puerto%, ByVal Dirección%) As Integer
```

```
Public Declare Function Escribir_Ad8b32 Lib
"LIPS_32.DLL" (ByVal Valor%, ByVal Puerto%,
ByVal Dirección%) As Integer
```

```
Public Declare Function Poner_Modo_Ad8b32 Lib
"LIPS_32.DLL" (ByVal Modo%, ByVal Puerto%,
ByVal Dirección%) As Integer
```

```
Public Declare Function Autodeteccion_Ad8b32 Lib
"LIPS_32.DLL" (Puerto%, Dirección%) As Integer
```

C (Borland)

```
short_stdcall Autodeteccion_Ad8b32(SHORT FAR
*Puerto,SHORT FAR *Dir);
```

```
short_stdcall Poner_Modo_Ad8b32(short
modo,short Puerto,short Dir);
```

```
short_stdcall Leer_Ad8b32(SHORT FAR
*ch0,SHORT FAR *ch1,SHORT FAR *ch2,SHORT
FAR *ch3,short Puerto, short Dir);
```

```
short_stdcall Escribir_Ad8b32(short dato,short
Puerto, short Dir);
```

Estas funciones equivalen a sus homólogas de la librería AD8B.DLL

III.4.2.3 Funciones para el subsistema contador que contienen los módulos ADT e IOT.

Declaración:

Visual Basic 4.0 32 bits

```
Public Declare Function Autodeteccion_Cont32 Lib
"LIPS_32.DLL" (Puerto%, Dirección%) As Integer
```

```
Public Declare Function Activar_Alarma32 Lib
"LIPS_32.DLL" (ByVal Puerto%, ByVal Dirección%)
```

As Integer

```
Public Declare Function Activar_Contaje32 Lib
"LIPS_32.DLL" (ByVal Puerto%, ByVal Dirección%)
As Integer
```

```
Public Declare Function Escribir_Alarma32 Lib
"LIPS_32.DLL" (ByVal MiAlarma As Long, ByVal
Puerto%, ByVal Dirección%) As Integer
```

```
Public Declare Function Escribir_Cuenta32 Lib
"LIPS_32.DLL" (ByVal InicioCuenta As Long, ByVal
Puerto%, ByVal Dirección%) As Integer
```

```
Public Declare Function Leer_Cuenta32 Lib
"LIPS_32.DLL" (ByVal Puerto%, ByVal Dirección%)
As Long
```

```
Public Declare Function Parar_Alarma32 Lib
"LIPS_32.DLL" (ByVal Puerto%, ByVal Dirección%)
As Integer
```

```
Public Declare Function Parar_Contaje32 Lib
"LIPS_32.DLL" (ByVal Puerto%, ByVal Dirección%)
As Integer
```

C (Borland)

```
short_stdcall Autodeteccion_Cont32(SHORT FAR
*puerto,SHORT FAR *Dir);
```

```
short_stdcall Escribir_Cuenta32 (LONG
InicioCuenta,short Puerto, short Dir);
```

```
short_stdcall Parar_Contaje32(short Puerto,
short Dir);
```

```
short_stdcall Activar_Contaje32(short Puerto,
short Dir);
```

```
short_stdcall Parar_Alarma32(short Puerto, short
Dir);
```

```
short_stdcall Activar_Alarma32(short Puerto,
short Dir);
```

```
short_stdcall Escribir_Alarma32(LONG
MiAlarma,short Puerto,short Dir);
```

```
LONG_stdcall Leer_Cuenta32(short Puerto, short
Dir);
```

```
LONG_stdcall Leer_Alarma32(short Puerto, short
Dir);
```

Estas funciones equivalen a sus homólogas de la librería CONT.DLL. Entiendase como LONG un entero de 32 bits

III.4.2.4 Funciones para el subsistema entrada-salida digital que contienen los módulos ADT, ADQ, DIG e IOT.

Declaración:

Visual Basic 4.0 32 bits

```
Public Declare Function Autodeteccion_Dig32 Lib  
"LIPS_32.DLL" (Puerto%, Dirección%) As Integer
```

```
Public Declare Function Escribir_Dig32 Lib  
"LIPS_32.DLL" (ByVal Valor%, ByVal Puerto%,  
ByVal Dirección%) As Integer
```

```
Public Declare Function Leer_Dig32 Lib  
"LIPS_32.DLL" (ByVal Puerto%, ByVal Dirección%)  
As Integer
```

C (Borland)

```
short _stdcall Autodeteccion_Dig32 (SHORT FAR*  
,SHORT FAR* );
```

```
short _stdcall Escribir_Dig32(short dato,short  
Puerto,short Dir);
```

```
short _stdcall Leer_Dig32(short Puerto, short Dir);
```

Estas funciones equivalen a sus homólogas de la librería DIG.DLL

© *1998 LipSoft Electronics*

Sistema de Adquisición de Datos y Control desde PC
