



S8-232



Manual del usuario

Introducción

La controladora para servomotores S8-232 está diseñada para operar con todo tipo de servomotores estándar usados en aeromodelismo. Se provee completamente armada e incluye el cable de comunicación serie hacia la PC.

En su reducido tamaño de tan solo 50mm x 50mm puede manejar 8 servos y leer 4 entradas digitales del tipo TTL.

Es operada fácilmente desde un puerto serial RS-232 mediante un reducido set de comandos que permiten realizar todas las operaciones soportadas por la misma.

Sus ocho salidas independientes de 8-bits para el control de servos permiten 255 posiciones para cada servo. Permite activar o desactivar los servos (posición 0). Es posible configurar ocho posiciones de inicio independientes para cada uno de los servos. Incluye software de control con código fuente gratis y libre para su modificación y adaptación a su propio proyecto, el mismo puede ser descargado desde nuestro sitio web.

Principios básicos de las señales de control de los servos

Cada servo es controlado mediante un tren de pulsos a través del cable de señal (normalmente blanco o amarillo), que es el pin 'S' de los conectores del servo (CN2). La tasa de pulsos puede variar, aunque el valor medio es de 50 Hz, es decir, un pulso cada 20 miliseg.

La posición de los servos se controla a través de la amplitud de los pulsos. La amplitud de pulsos oscila entre 480 microseg. y 2520 microseg., cuyo centro está en 1500 microseg.

Cuando el servo es posicionado en el valor 1, el ancho de pulso que se envía al mismo es de 480 microseg., cuando se lo lleva al otro extremo, es decir, se lo posiciona en el valor 255, el ancho de pulso enviado pasa a ser de 2520 microseg. La posición central (valor 127) enviará un pulso de 1500 microseg.

Por cada punto que se aumente o disminuya, la posición el ancho del pulso aumentará o disminuirá en 8 microseg.

Los servos pueden ser desactivados independientemente colocando el mismo en la posición 0. En esta posición el tren de pulsos hacia dicho servo es detenido, aunque la alimentación continúa para aplicarla a los componentes electrónicos internos del servo. La ausencia de tren de pulsos libera el motor del servo y este no intentará mantener ninguna posición, esto significa que el servo podrá ser movido libremente.

Para mas información sobre la teoría de funcionamiento de los servos sugerimos descargar de nuestro sitio web el informe sobre servomotores:

<http://www.todorobot.com.ar/documentos/servomotor.pdf>

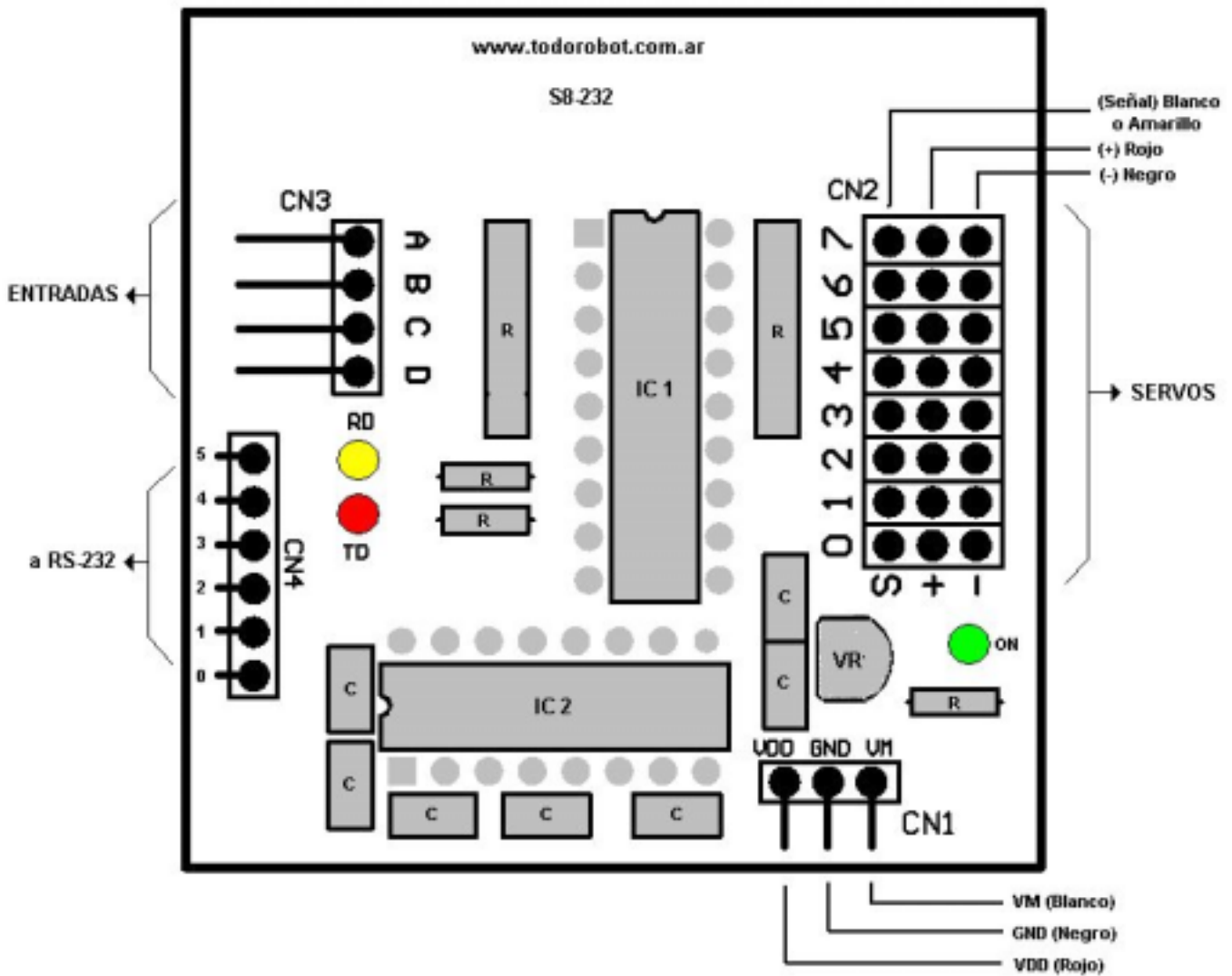
Características físicas y eléctricas

Alimentación (CN1)	
Tensión de alimentación lógica VDD	9 a 15 VDC
Tensión de alimentación Servos VM *	5 a 6 VDC
Consumo medio en reposo (sin servos)	13 mA
Comunicación RS-232 (CN4)	
Velocidad	2400 bps
Paridad	None
Palabra	8 bits
Stop Bit	1
Servos (CN2)	
Cantidad de salidas para servos	8
Desactivación independiente de servos	Si - Posición 0
Ancho mínimo de pulso de control (Posición 1)	480 microseg.
Ancho máximo de pulso de control (Posición 255)	2520 microseg.
Ancho medio de pulso de control (Posición 127)	1500 microseg.
Variación mínima del pulso de control	8 microseg.
Entradas (CN3)	
Cantidad de entradas TTL	4 - (A, B, C y D)
Niveles aceptables para entradas TTL	0v a 5v
Dimensión	
Dimensiones físicas	50mm x 50mm

* 5v a 6v es el valor estándar que los servos para aeromodelismo utilizan, sin embargo siempre consultar la hoja de datos de los mismos antes de alimentarlos.

Conectores y LED indicadores

En la siguiente figura podemos apreciar el detalle de todos los conectores y los LED indicadores de la controladora S8-232:



CN1 - Entrada de alimentación

La controladora es provista con un conector de alimentación armado siendo el cable Rojo VDD, el Negro GND y el Blanco VM.

CN2 - Conexión servos 0 a 7

En CN2 se conectan los servos respetando la polaridad indicada. Hay que tener en cuenta que algunas marcas arman sus conectores con colores y pinout diferente a lo estándar, por lo que siempre se debe revisar la hoja de datos del servo antes de conectar el mismo.

En la siguiente figura pueden verse algunos tipos de conectores utilizados por los servos.

**CN3 - Entradas TLL (A , B , C , D)**

Las entradas digitales A, B, C y D permiten la lectura de sensores y actuadores tipo fin de carro, pulsadores, etc.

Cuando estas entradas se encuentran desconectadas son leídas en nivel Alto (1 lógico). Esto permite que cualquier sensor o actuador que fuerce las mismas a GND provocará que dicha entrada sea leída en nivel Bajo (0 lógico).

CN4 - Conexión a puerto serie RS-232

CN4 es la salida de comunicación hacia el puerto serie de la PC, en el se conectará el cable de comunicación provisto con la controladora.

El pinout del cable de comunicación se detalla a continuación:

Pin en DB9 (PC)	Pin en CN4 (S8-232)
2	0
3	1
5	2

Reconociendo los LED's indicadores

LED	Descripción
VERDE	Este LED indica el encendido de la controladora. Este LED solo indica la presencia de tensión en la entrada VDD, pero no indica que la misma sea correcta así como tampoco indica la entrada de alimentación en VM
ROJO	Este LED indica que la controladora está enviando datos vía RS-232 a la PC (Transmit – TD)
AMARILLO	Este LED indica que la controladora está recibiendo datos vía RS-232 desde la PC (Receive – RD)

Comandos

Comandos soportados

Comando	Descripción
V	<i>Versión</i> - Envía a la PC la versión de Firmware actualmente cargada
RST	<i>Reset</i> - Restablece la posición de todos los servos al valor inicial cargado en memoria (ver comandos Dxy y DT)
Lx	<i>Lee Posición</i> - Lee y envía a la PC la posición actual del servo indicado por 'X'
LT	<i>Lee Todos</i> - Lee la posición actual de los 8 servos y envía el dato a la PC
Ix	<i>Entrada</i> - Envía a la PC el estado de la entrada indicada por 'X'
IT	<i>Todas las Entradas</i> - Envía a la PC el estado de las 4 entradas (A a D)
Pxy	<i>Posicionar Servo</i> - Mueve el servo indicado por 'X' a la posición indicada por 'Y'
Tyyyyyyyy	<i>Posicionar Todos</i> - Mueve todos los servos a las posiciones indicadas por 'Y'. La primer 'Y' corresponde al servo 0 y la última al servo 7
Dxy	<i>Default</i> - Graba en memoria el valor 'Y' como posición inicial para el servo 'X'
DT	<i>Default Todos</i> - Graba en memoria las posiciones de los 8 servos como posiciones iniciales de los mismos.

Estructura general de los comandos

Todos los comandos deben ser en mayúscula, no deben poseer espacios y deben terminar con un 'CR' (0x0D). Los parámetros identificados con las letras 'x' e 'y' constan de un solo byte con el valor adecuado según cada caso. Así por ejemplo si es un valor de posicionamiento el mismo puede ser 0 (0x00) a 255 (0xFF). En el caso de indicar el número de servo apuntado, el mismo puede ser 0 a 7. Si indicase una de las cuatro entradas, sería un byte correspondiente al valor ASCII que identifica cada entrada (A, B, C o D). Ver tabla ASCII al final del manual.

Todos los comandos responden con 'ACK' (0x06) + 'CR' (0x0D) si fueron exitosos. En caso de error todos los comandos responden con 'NACK' (0x15) + 'CR'. Los comandos de lectura ('V', 'Ix', 'IT', 'Lx' y 'LT') enviarán la respuesta seguidamente al ACK mas un 'CR' final.

Descripción detallada de los comandos

Comando: V

Descripción: Devuelve un String alfanumérico con la versión de firmware instalado

Parámetros: no posee

Respuesta OK: ACK + CR + Firmware (string) + CR

Respuesta Error: NACK + CR

EJEMPLO	Comando Enviado	Respuesta OK	Respuesta Error
Alfanumérico	'V' + 'CR'	'ACK' + 'CR' + 'S8v01r01'	'NACK' + 'CR'
Hexadecimal	56 0D	06 0D 53 38 76 30 31 72 30 31	15 0D

En el ejemplo se solicita la versión de firmware instalada, la cual es: *S8v01r01*

Comando: RST

Descripción: Restablece la posición de todos los servos al valor inicial cargado en memoria (ver comandos Dxy y DT)

Parámetros: no posee

Respuesta OK: ACK + CR

Respuesta Error: NACK + CR

EJEMPLO	Comando Enviado	Respuesta OK	Respuesta Error
Alfanumérico	'R' + 'S' + 'T' + 'CR'	'ACK' + 'CR'	'NACK' + 'CR'
Hexadecimal	52 53 54 0D	06 0D	15 0D

Comando: Lx

Descripción: Devuelve un byte con la posición del servo indicado por 'x'. El valor es devuelto en un byte con la posición actual del servo apuntado.

Parámetros: 'x' debe ser un byte entre 0 y 7

Respuesta OK: ACK + CR + Posición (1 byte: 0 a 255) + CR

Respuesta Error: NACK + CR

EJEMPLO	Comando Enviado	Respuesta OK	Respuesta Error
Alfanumérico	'L' + 2 + 'CR'	'ACK' + 'CR' + 123 + 'CR'	'NACK' + 'CR'
Hexadecimal	4C 02 0D	06 0D 7B 0D	15 0D

En el ejemplo se solicita la posición actual del servo 2 el cual está en la posición 123 (0x7B)

Comando: LT

Descripción: Devuelve ocho bytes con las posiciones de todos los servos. El valor es devuelto en ocho bytes consecutivos, el primer byte recibido corresponde a la posición del servo 0 y así sucesivamente hasta el servo 7.

Parámetros: no posee

Respuesta OK: ACK + CR + Posición Servo 0.....Servo 7 (8 bytes: 0 a 255) + CR

Respuesta Error: NACK + CR

EJEMPLO	Comando Enviado	Respuesta OK	Respuesta Error
Alfanumérico	'L' + 'T' + 'CR'	'ACK' + 'CR' + 22 + 10 + 51 + 44 + 8 + 12 + 110 + 203 + 'CR'	'NACK' + 'CR'
Hexadecimal	4C 54 0D	06 0D 16 0A 33 2C 08 0C 6E CB 0D	15 0D

En el ejemplo se solicita la posición actual de todos los servo, de acuerdo a la respuesta el servo 0 está en la posición 22 (0x16), el 1 en 10 (0x0A), el 2 en 51 (0x33), el 3 en 44 (0x2C), el 4 en 8 (0x08), el 5 en 12 (0x0C), el 6 en 110 (0x6E) y el 7 en 203 (0xCB)

Comando: Ix

Descripción: Devuelve un byte que refleja el estado de la entrada indicada por 'x'

Parámetros: 'x' es un byte con el caracter que representa la entrada (A, B, C o D). Ver tabla ASCII al final del manual.

Respuesta OK: ACK + CR + Estado (1 byte: 0 o 1) + CR

Respuesta Error: NACK + CR

EJEMPLO	Comando Enviado	Respuesta OK	Respuesta Error
Alfanumérico	'I' + 'B' + 'CR'	'ACK' + 'CR' + 1 + 'CR'	'NACK' + 'CR'
Hexadecimal	49 42 0D	06 0D 01 0D	15 0D

En el ejemplo se solicita el estado de la entrada digital B, la misma está en estado ALTO (1 lógico).

Comando: IT

Descripción: Devuelve 4 bytes que reflejan el estado de las cuatro entradas (A, B, C o D).

Parámetros: no posee

Respuesta OK: ACK + CR + Estado Input A Input D (4 bytes: 0 o 1) + CR

Respuesta Error: NACK + CR

EJEMPLO	Comando Enviado	Respuesta OK	Respuesta Error
Alfanumérico	'I' + 'T' + 'CR'	'ACK' + 'CR' + 1 + 0 + 0 + 1 + 'CR'	'NACK' + 'CR'
Hexadecimal	49 54 0D	06 0D 01 00 00 01 0D	15 0D

En el ejemplo se solicita el estado de las cuatro entradas digitales, según la respuesta la entrada A está en estado ALTO (1 lógico), la entrada B está en estado BAJO (0 lógico), la entrada C está en estado BAJO (0 lógico) y la entrada A está en estado ALTO (1 lógico).

Comando: Pxy**Descripción:** Posiciona el servo 'x' en la posición deseada 'y'**Parámetros:** 'x' debe ser un byte entre 0 y 7, 'y' debe ser un byte entre 0 y 255**Respuesta OK:** ACK + CR**Respuesta Error:** NACK + CR

EJEMPLO	Comando Enviado	Respuesta OK	Respuesta Error
Alfanumérico	'P' + 6 + 143 + 'CR'	'ACK' + 'CR'	'NACK' + 'CR'
Hexadecimal	50 06 8F 0D	06 0D	15 0D

En el ejemplo se moverá el servo 6 a la posición 143 (0x 8F).

Comando: Tyyyyyyyy**Descripción:** Posiciona los 8 servos de una sola vez, cada "y" es la posición de uno de los servos. La primer 'y' representa la posición del servo 0, siendo la última la posición para el servo 7.**Parámetros:** Cada una de las 'y' debe ser un byte entre 0 y 255**Respuesta OK:** ACK + CR**Respuesta Error:** NACK + CR

EJEMPLO	Comando Enviado	Respuesta OK	Respuesta Error
Alfanumérico	'T' + 10 + 14 + 9 + 110 + 18 + 25 + 220 + 102 + 'CR'	'ACK' + 'CR'	'NACK' + 'CR'
Hexadecimal	54 0A 0E 09 6E 12 19 DC 66 0D	06 0D	15 0D

En el ejemplo se moverá el servo 0 a la posición 10 (0x 0A), el servo 1 a la posición 14 (0x 0E), el servo 2 a la posición 9 (0x 09), el servo 3 a la posición 110 (0x 6E), el servo 4 a la posición 18 (0x 12), el servo 5 a la posición 25 (0x 19), el servo 6 a la posición 220 (0x DC) y el servo 7 a la posición 102 (0x 66).

Comando: Dxy**Descripción:** Graba en memoria la posición inicial indicada por 'y' del servo indicado por 'x'.**Parámetros:** 'x' debe ser un byte entre 0 y 7, 'y' debe ser un byte entre 0 y 255**Respuesta OK:** ACK + CR**Respuesta Error:** NACK + CR

EJEMPLO	Comando Enviado	Respuesta OK	Respuesta Error
Alfanumérico	'D' + 1 + 79 + 'CR'	'ACK' + 'CR'	'NACK' + 'CR'
Hexadecimal	44 01 4F 0D	06 0D	15 0D

En el ejemplo se establece el valor 79 (0x4F) como posición inicial para el servo 1.

Comando: DT

Descripción: Graba en memoria la posición actual de todos los servos estableciendo la misma como posición inicial de cada uno de los mismos. Cada vez que la controladora sea encendida o bien reseteada, los 8 servos se posicionarán en los valores grabados en memoria. Estos valores iniciales permanecen aún cuando la controladora pierda su alimentación, debido a que son grabados en memoria no volátil.

Parámetros: no posee

Respuesta OK: ACK + CR

Respuesta Error: NACK + CR

EJEMPLO	Comando Enviado	Respuesta OK	Respuesta Error
Alfanumérico	'D' + 'T' + 'CR'	'ACK' + 'CR'	'NACK' + 'CR'
Hexadecimal	44 54 0D	06 0D	15 0D

Programa de Control

Desde nuestro sitio web puede descargar el programa de control de la controladora junto con los archivos de código fuente elaborados en VisualBasic 6.0 para poder utilizar el mismo como base de su propio proyecto. El mismo posee todas las rutinas necesarias para el envío de los comandos hacia la controladora.

En la siguiente figura se puede observar la pantalla correspondiente al programa de control:



Utilizando el programa de control:

Al cargar el programa el primer paso es seleccionar el puerto a utilizar (COM 1, COM 2, etc). Al seleccionar el puerto el programa intentará comunicarse con la controladora para verificar la comunicación y a su vez recuperar el estado actual de la misma para reflejarlo en la pantalla. Debido a esto, antes de seleccionar el puerto se debe tener la controladora encendida y conectada.

Si la comunicación no pudo ser establecida el programa dará error y permanecerá desactivado hasta que se intente nuevamente el proceso anterior.

Existen tres modalidades de uso. *Asistido*, *Comando Inmediato* y *Archivo de Comandos*.

Modo Asistido:

El modo asistido se utiliza mediante los botones y las barras de desplazamiento de cada servo. Así por ejemplo al desplazar una de las barras y liberar la misma, el servo correspondiente se ubicará en la posición seleccionada. Es decir, al desplazar y soltar una de las barras de desplazamiento el programa envía hacia la controladora el comando 'Pxy' con los valores correspondientes según el servo (x) y la posición (y).

El botón *ACTUALIZAR* envía el comando 'IT' que indica a la controladora que debe leer y enviar a la PC el estado de sus cuatro entradas (A, B, C y D).

El botón *RESET* envía el comando 'RST' el cual provoca que todos los servos se posicionen en el valor inicial grabado en la memoria de la controladora.

El botón *VERSION* enviará el comando 'V' el cual solicita a la controladora que envíe a la PC la versión del firmware instalado en la misma.

Por último el botón *DEFAULT* enviará el comando 'DT'. Este comando provocará que los valores actuales de cada servo sean almacenados como valores de arranque de los mismos. Así cada vez que la controladora sea encendida o reseteada todos los servos serán posicionados a estos valores.

Modo Comando Inmediato:

En esta modalidad el comando es ingresado directamente y al presionar el botón "Ejecutar" el mismo es enviado a la controladora. El resultado y/o respuesta de la controladora es visualizado en pantalla.

Los comandos se escriben separando cada parámetro mediante un espacio. Así por ejemplo el comando Pxy suponiendo que queremos posicionar el servo 2 en la posición 132 deberíamos escribirlo: P 2 132

Si bien la controladora solo acepta caracteres en mayúscula, el programa se asegura de convertir todo a mayúscula antes de enviarlo a la controladora, por lo tanto el comando anterior también puede ser escrito como: p 2 132

Los comandos sin parámetros (RST, IT, DT, etc) no deben poseer espacios, es decir, el comando Reset se escribirá RST.

Modo Archivo de Comandos:

Esta modalidad permite la ejecución automática de un archivo conteniendo una secuencia de comandos. Estos comandos deben ser previamente escritos en un archivo de texto plano y debe contener un comando por línea. La estructura de cada comando debe respetar las mismas reglas expuestas en Comando Inmediato, es decir, todos los parámetros deben ir separados por un espacio. Presionar el botón **Loop** en lugar del botón **Ejecutar** si se desea una ejecución continua del archivo, en este caso al terminar la última instrucción comenzará inmediatamente por la primera instrucción. Es recomendable en este caso que la última instrucción se una pausa (W) para que la última instrucción sea correctamente ejecutada.

A continuación se puede ver un ejemplo de un archivo de comando:

```
P 2 155
W 500
P 2 74
I A
T 44 200 132 100 54 68 96 251
W 1000
RST
W 500
```

En el ejemplo anterior podemos ver que primero se posiciona el servo 2 en la posición 155, luego se realiza una pausa de 500 milisegundos y se reposiciona el servo 2 en la posición 74. Seguidamente se solicita que informe el estado de la entrada 'A' para luego posicionar todos los servos al mismo tiempo con el comando 'Tyyyyyyy'. Finalmente se realiza una pausa de 1 segundo y se resetea la controladora.

El comando 'W' (wait) no es un comando soportado por la controladora sino que es un comando solo interpretado por el programa de control. Este comando realiza una pausa de 1 a 10.000 milisegundos (0.001 a 10 segundos) durante la ejecución del archivo de comandos. Esto es particularmente útil cuando se reposiciona un servo continuamente, así el comando W permitirá realizar una pausa que permita al servo posicionarse antes de enviarle nuevamente un comando con una nueva posición.

Tabla de Caracteres ASCII

CARACTERES NO IMPRIMIBLES

DEC	HEX	CHARACTER (CODE)	DEC	HEX	CHARACTER (CODE)
0	0	NULL	16	10	DATA LINK ESCAPE (DLE)
1	1	START OF HEADING (SOH)	17	11	DEVICE CONTROL 1 (DC1)
2	2	START OF TEXT (STX)	18	12	DEVICE CONTROL 2 (DC2)
3	3	END OF TEXT (ETX)	19	13	DEVICE CONTROL 3 (DC3)
4	4	END OF TRANSMISSION (EOT)	20	14	DEVICE CONTROL 4 (DC4)
5	5	END OF QUERY (ENQ)	21	15	NEGATIVE ACKNOWLEDGEMENT (NAK)
6	6	ACKNOWLEDGE (ACK)	22	16	SYNCHRONIZE (SYN)
7	7	BEEP (BEL)	23	17	END OF TRANSMISSION BLOCK (ETB)
8	8	BACKSPACE (BS)	24	18	CANCEL (CAN)
9	9	HORIZONTAL TAB (HT)	25	19	END OF MEDIUM (EM)
10	A	LINE FEED (LF)	26	1A	SUBSTITUTE (SUB)
11	B	VERTICAL TAB (VT)	27	1B	ESCAPE (ESC)
12	C	FF (FORM FEED)	28	1C	FILE SEPARATOR (FS) RIGHT ARROW
13	D	CR (CARRIAGE RETURN)	29	1D	GROUP SEPARATOR (GS) LEFT ARROW
14	E	SO (SHIFT OUT)	30	1E	RECORD SEPARATOR (RS) UP ARROW
15	F	SI (SHIFT IN)	31	1F	UNIT SEPARATOR (US) DOWN ARROW

CARACTERES IMPRIMIBLES

DEC	HEX	CHARACTER	DEC	HEX	CHARACTER	DEC	HEX	CHARACTER
32	0x20	<SPACE>	64	0x40	@	96	0x60	`
33	0x21	!	65	0x41	A	97	0x61	a
34	0x22	"	66	0x42	B	98	0x62	b
35	0x23	#	67	0x43	C	99	0x63	c
36	0x24	\$	68	0x44	D	100	0x64	d
37	0x25	%	69	0x45	E	101	0x65	e
38	0x26	&	70	0x46	F	102	0x66	f
39	0x27	'	71	0x47	G	103	0x67	g
40	0x28	(72	0x48	H	104	0x68	h
41	0x29)	73	0x49	I	105	0x69	i
42	0x2A	*	74	0x4A	J	106	0x6A	j
43	0x2B	+	75	0x4B	K	107	0x6B	k
44	0x2C	,	76	0x4C	L	108	0x6C	l
45	0x2D	-	77	0x4D	M	109	0x6D	m
46	0x2E	.	78	0x4E	N	110	0x6E	n
47	0x2F	/	79	0x4F	O	111	0x6F	o
48	0x30	0	80	0x50	P	112	0x70	p
49	0x31	1	81	0x51	Q	113	0x71	q
50	0x32	2	82	0x52	R	114	0x72	r
51	0x33	3	83	0x53	S	115	0x73	s
52	0x34	4	84	0x54	T	116	0x74	t
53	0x35	5	85	0x55	U	117	0x75	u
54	0x36	6	86	0x56	V	118	0x76	v
55	0x37	7	87	0x57	W	119	0x77	w
56	0x38	8	88	0x58	X	120	0x78	x
57	0x39	9	89	0x59	Y	121	0x79	y
58	0x3A	:	90	0x5A	Z	122	0x7A	z
59	0x3B	;	91	0x5B	[123	0x7B	{
60	0x3C	<	92	0x5C	\	124	0x7C	
61	0x3D	=	93	0x5D]	125	0x7D	}
62	0x3E	>	94	0x5E	^	126	0x7E	~
63	0x3F	?	95	0x5F	_	127	0x7F	

CARACTERES ASCII EXTENDIDO								
DEC	HEX	CHARACTER	DEC	HEX	CHARACTER	DEC	HEX	CHARACTER
128	0x80	Ç	171	0xab	½	214	0xD6	Ö
129	0x81	Ù	172	0xAC	¼	215	0xD7	×
130	0x82	É	173	0xAD	¡	216	0xD8	Ø
131	0x83	â	174	0xAE	«	217	0xD9	Û
132	0x84	ä	175	0xAF	»	218	0xDA	Ü
133	0x85	à	176	0xB0	°	219	0xDB	Û
134	0x86	À	177	0xB1	±	220	0xDC	Û
135	0x87	‡	178	0xB2	²	221	0xDD	Ý
136	0x88	^	179	0xB3	³	222	0xDE	Þ
137	0x89	‰	180	0xB4	´	223	0xDF	ß
138	0x8A	Š	181	0xB5	µ	224	0xE0	à
139	0x8B	‹	182	0xB6	¶	225	0xE1	á
140	0x8C	Œ	183	0xB7	·	226	0xE2	â
141	0x8D		184	0xB8	¸	227	0xE3	ã
142	0x8E	Ž	185	0xB9	¹	228	0xE4	ä
143	0x8F		186	0xBA	º	229	0xE5	å
144	0x90		187	0xBB	»	230	0xE6	æ
145	0x91	‘	188	0xBC	¼	231	0xE7	ç
146	0x92	’	189	0xBD	½	232	0xE8	è
147	0x93	“	190	0xBE	¾	233	0xE9	é
148	0x94	”	191	0xBF	¿	234	0xEA	ê
149	0x95	•	192	0xC0	À	235	0xEB	ë
150	0x96	—	193	0xC1	Á	236	0xEC	ì
151	0x97	—	194	0xC2	Â	237	0xED	í
152	0x98	˜	195	0xC3	Ã	238	0xEE	î
153	0x99	™	196	0xC4	Ä	239	0xEF	ï
154	0x9A	š	197	0xC5	Å	240	0xF0	ð
155	0x9B	›	198	0xC6	Æ	241	0xF1	ñ
156	0x9C	œ	199	0xC7	Ç	242	0xF2	ò
157	0x9D		200	0xC8	È	243	0xF3	ó
158	0x9E	ž	201	0xC9	É	244	0xF4	ô
159	0x9F	ÿ	202	0xCA	Ê	245	0xF5	õ
160	0xA0		203	0xCB	Ë	246	0xF6	ö
161	0xA1	¡	204	0xCC	Ì	247	0xF7	÷
162	0xA2	¢	205	0xCD	Í	248	0xF8	ø
163	0xA3	£	206	0xCE	Î	249	0xF9	ù
164	0xA4	¤	207	0xCF	Ï	250	0xFA	ú
165	0xA5	¥	208	0xD0	Ð	251	0xFB	û
166	0xA6	¦	209	0xD1	Ñ	252	0xFC	ü
167	0xA7	§	210	0xD2	Ò	253	0xFD	ý
168	0xA8	¨	211	0xD3	Ó	254	0xFE	þ
169	0xA9	©	212	0xD4	Ô	255	0xFF	ÿ
170	0xAA	ª	213	0xD5	Õ			