

1.- INTRODUCCIÓN:

Todas las posibilidades que ofrecen la mayoría de las pantallas LCD Gráficas que se utilizan van controladas por el Chip T6963C de “**TOSHIBA**” que fue diseñado con el objetivo de controlar pantallas de cristal líquido de pequeño, mediano y gran tamaño. Es este chip el que recibe e interpreta todos los comandos que se le envían a la pantalla, ya sea desde un PC, un microcontrolador PIC o desde cualquier otro controlador que deseemos utilizar para manejar la misma. Por lo tanto, conocer el funcionamiento de las pantallas de la empresa Wintek (base de mi proyecto controlador de pantallas gráficas), nos permitirá conocer a su vez el funcionamiento básico del resto de pantallas gráficas que existen en el mercado y utilizan el mismo chip como base de control de sus opciones.

1.1 FUNCIONES BÁSICAS:

- ✓ El Chip, puede ejercer de interfaz entre la pantalla y cualquier microcontrolador (MPU) de 8 bits.
- ✓ También permite la comunicación entre la MPU y la memoria RAM que poseen las pantallas (VRAM).
- ✓ Se encarga de generar las señales de tiempo y datos necesarias para el correcto funcionamiento del resto de circuitos integrados que contribuyen al manejo de la pantalla de cristal líquido.
- ✓ Además, tiene una memoria ROM que le permite generar 128 caracteres diferentes (CG-ROM) y posee capacidad para controlar hasta 64 Kbytes de memoria RAM externa (VRAM). Esta última puede estar dedicada a textos, gráficos e incluso a generar caracteres adicionales a parte de los 128 que genera internamente el Chip.
- ✓ Por último, cabe destacar que este Chip puede soportar una amplia gama de formatos de pantallas y tiene la capacidad de combinar textos y gráficos dentro del LCD.

2.- ESPECIFICACIONES ELÉCTRICAS DEL T6963C:

2.1 RANGOS MÁXIMOS ABSOLUTOS:

Se definen a continuación, en la tabla que se adjunta bajo estas líneas, los máximos valores y rangos de las características y elementos más importantes para la alimentación, uso, etc. del chip T6963C, y por tanto de la pantalla gráfica asociada a él:

Elemento	Símbolo	Valor	Unidades
Voltaje de alimentación	V_{DD}	-0.3 a +7.0	V
Voltaje de entrada	V_{in}	-0.3 a $V_{DD}+0.3$	V
Temperatura de operación	Top	-10 a +70	°C
Temperatura de almacenamiento	Tstg	-55 a +125	°C

(Válido sólo para el chip T6963C)

2.2 CARACTERÍSTICAS ELÉCTRICAS:

A continuación se muestran en una tabla las condiciones eléctricas más interesantes del chip para su posterior implementación con una pantalla gráfica, pantalla LCD, etc:

Elemento	Símbolo	Condición	Mínimo	Máximo	Unidades
Voltaje de operación	V_{DD}	----	4.5	5.5	V
Voltaje de entrada	V_{IH} / V_{IL}	----	$V_{DD}-2$	V_{DD}	V
Voltaje de salida	V_{OH} / V_{OL}	----	$V_{DD}-0.3$	V_{DD}	V
Impedancia de salida	R_{OH} / R_{OL}	$V_{out}=V_{DD}-0.5$	----	400	Ω
Consumo de corriente	I_{DD}	$V_{DD}=5.0V$ $f_{osc}=3.0MHz$	----	6	MA
Oscilación interna	fosc	----	0.4	5.5	MHz
Frecuencia del reloj	fcp	----	----	2750	KHz
Escalón del reloj	trcp,tfcP	----	----	30	Ns

3.- INTERFAZ DE CONEXIONES DE PINES:

3.1 FUNCIONES DE LOS PINES DEL INTERFAZ CPU:

En este apartado se tratan las funciones de las distintas patillas del chip para su comunicación con la CPU. Para ello se muestra en las líneas que siguen una tabla con la nomenclatura de los pines así como con la función correspondiente que desempeñan. Además se añade la información de ver si ese pin es de entrada o de salida (o de ambos tipos alternativamente):

Nombre	Número Pin/Pata PIC	Entrada (I) / Salida (O)	Función del Pin
FGND	1	----	Tierra Virtual
GND	2	----	Tierra (0V)
V _{DD}	3	----	Fuente (+5V)
V _{EE}	4	I/O	Voltaje de operación pantalla
WR	5 / A0	I	Escritura en el T6963C desde la CPU
RD	6 / A1	I	Lectura del T6963C desde la CPU
CE	7 / A2	I	Chip activado
C/D	8 / A3	I	Selector de registros: "1"=Comandos "2"=Datos
NC	9	----	No conexión
RST	10 / A4	I/O	Reset
DB0	11 / B0	I/O	Bus de datos bidireccional (línea 0)
DB1	12 / B1	I/O	Bus de datos bidireccional (línea 1)
DB2	13 / B2	I/O	Bus de datos bidireccional (línea 2)
DB3	14 / B3	I/O	Bus de datos bidireccional (línea 3)
DB4	15 / B4	I/O	Bus de datos bidireccional (línea 4)
DB5	16 / B5	I/O	Bus de datos bidireccional (línea 5)
DB6	17 / B6	I/O	Bus de datos bidireccional (línea 6)
DB7	18 / B7	I/O	Bus de datos bidireccional (línea 7)
MD2	19	I	Selector de modo: "H"=40 columnas "H"=32 columnas
RV	20	I/O	H: reverse L: no reverse

También muestro la **asignación de pines físicos** en la **Figura 3.1**.

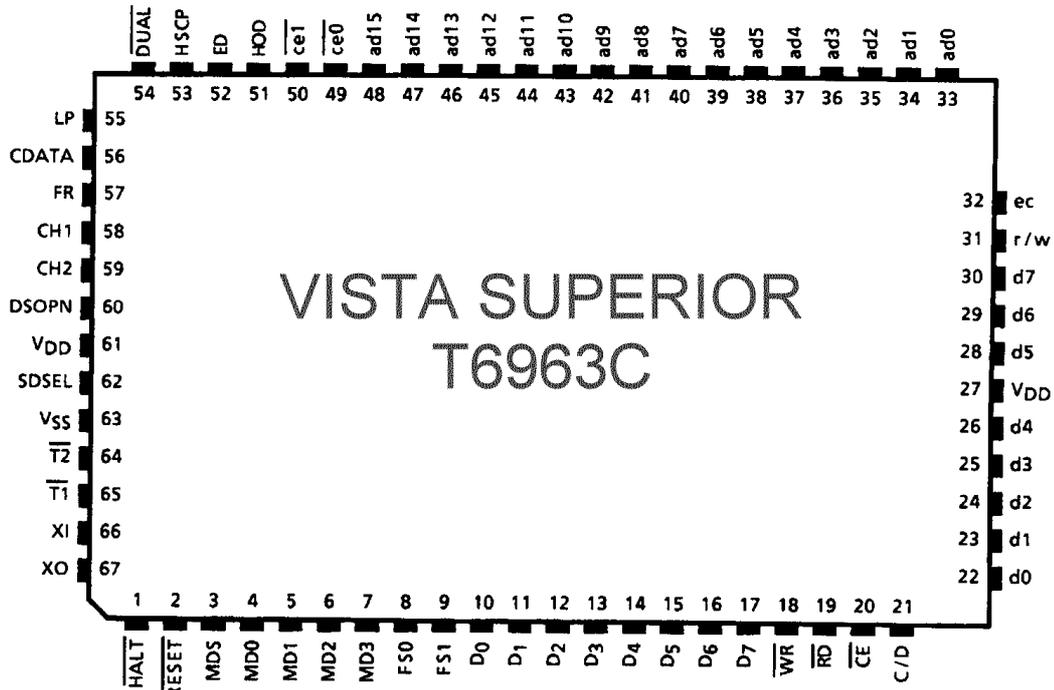


Figura 3.1

3.2 FUNCIONES DE LOS PINES DEL INTERFAZ CON EL LCD:

En este caso, en cambio, se definen los pines encargados con la comunicación (o interfaz) del chip con la propia pantalla (propia de cada sistema o proyecto). Para ello, como en el caso anterior, se da el nombre del pin correspondiente, así como la función que desempeña y si es entrada o salida (o bien ambas alternativamente):

Nombre del Pin	Entrada (I) / Salida (O)	Función del Pin
D1	O	Datos serie línea 1
FLM (CDATA)	O	Puntero de comienzo línea
M (FR)	O	Señal de Control
CL1 (LP)	O	Reloj "cerrojo" (Latch)
CL2 (SCP)	O	Reloj para datos serie
D2	O	Datos serie línea 2
V _{DD}	----	Fuente de alimentación para circuitos lógicos
V _{SS}	----	Tierra
V _{EE}	----	Fuente de alimentación para drivers 'LC'
V _o	----	Voltaje de operación para drivers 'LC'

4.- CARACTERÍSTICAS DE TIEMPO: (TIEMPO INTERFAZ CPU \Leftrightarrow T6963C):

En este cuarto apartado del estudio del T6963C se especifican las características de tiempo de la pantalla (del chip asociado, y por tanto de la pantalla). Básicamente los tiempos que en la siguiente tabla se especifican son los relativos a la comunicación entre la CPU y el propio chip, es decir, la pantalla:

Elemento	Símbolo	Mínimo	Tipo	Máximo	Unidades
C/D tiempo de activación	t_{CDS}	100	----	----	ns
C/D tiempo de espera	t_{CDH}	10	----	----	ns
CE, RD, WR: ancho de pulso	t_{CE}, t_{RD}, t_{WR}	80	----	----	ns
Datos-tiempo de activación	t_{DS}	80	----	----	ns
Datos-tiempo de espera	t_{DH}	40	----	----	ns
Tiempo de acceso	t_{ACC}	----	----	150	ns
Salida-tiempo de espera	t_{OH}	10	----	50	ns

Asimismo, el **diagrama de tiempos** asociado a dicha comunicación es el que se muestra en la **Figura 4.1**.

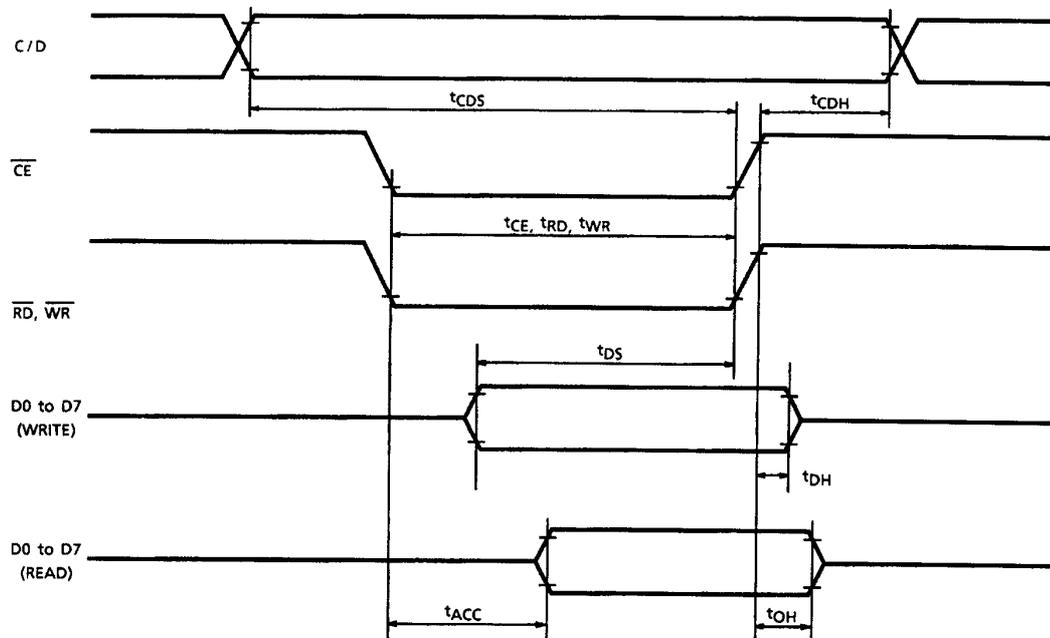


Figura 4.1

5.- ESPECIFICACIONES MECÁNICAS PROPIAS DE LA PANTALLA GRÁFICA (WINTEK):

A continuación se dan en forma de tabla las características mecánicas más importantes de la pantalla gráfica concreta que he utilizado en mis proyectos, es decir, la pantalla de Wintek WM - G2412A - 1GFW en sus dos tamaños.

ELEMENTO	VALOR ESTÁNDAR	UNIDADES
Número de segmentos	240 x 128	Pixels
Dimensión del módulo	170 x 101.2 x 14	Mm
Área efectiva del display	132(ancho)x76(alto)	Mm
Tamaño del pixel	0.47(ancho)x0.47(alto)	Mm
Tamaño pixel blanco	0.50(ancho)x0.50(alto)	Mm
Peso aproximado	250	Gr
Tipo de LCD	Oscuro (tipo positivo)	
Dirección de visualización	Orientación del reloj	
Luz de fondo	Color pálido	
Precisión	1/128	

(Nota: un LCD es de tipo negativo si el fondo es blanco por defecto. Es de tipo positivo en el caso contrario, que es precisamente el que nos ocupa)

6.- CARACTERÍSTICAS ÓPTICAS (WINTEK):

Se describen seguidamente distintos ángulos de visión posibles de la pantalla y su resolución consecuencia del ángulo de visión:

(Como se ve, ésta también es una característica propia de la pantalla y no del chip)

Elemento	Símbolo	Condición	Mínimo	Típico	Máximo	Unidades
Ángulo de visión (V)	θ	$CR \geq 2.0$	10	-	40	deg.
Ángulo de visión (H)	ϕ	$CR \geq 2.0$	-30	-	30	deg.
Ratio de contraste	C_R	-	-	5	-	-
Tiempo de respuesta	T_{ON}	-	-	200	350	mS
Tiempo de respuesta	T_{OFF}	-	-	200	350	mS

7.- DIAGRAMA DE BLOQUES:

El chip T6963C es un chip especialmente preparado para el control de pantallas LCD, sobre todo para pantallas gráficas, que a diferencia de las pantallas LCD convencionales, pueden mostrar gráficos y dibujos prediseñados. Dicho chip tiene un bus de 8 bits en paralelo para establecer la comunicación con la MPU. Además posee 128 palabras - caracteres prediseñados en memoria ROM que pueden controlarse mediante un display externo en memoria RAM de 64Kbytes.

Asimismo, los textos y diferentes gráficos que pueden aparecer en la pantalla pueden moverse con facilidad dentro de la misma. Por ello, su uso puede darse a texto, gráficos y distintas relaciones entre gráficos y textos.

A continuación, muestro el **diagrama de bloques** asociado al citado chip. El primer diagrama es el lógico general y el segundo por su parte, es una ampliación interna del primero para mostrar más detalladamente todas las conexiones, componentes, etc. (**Figura 7.1**)

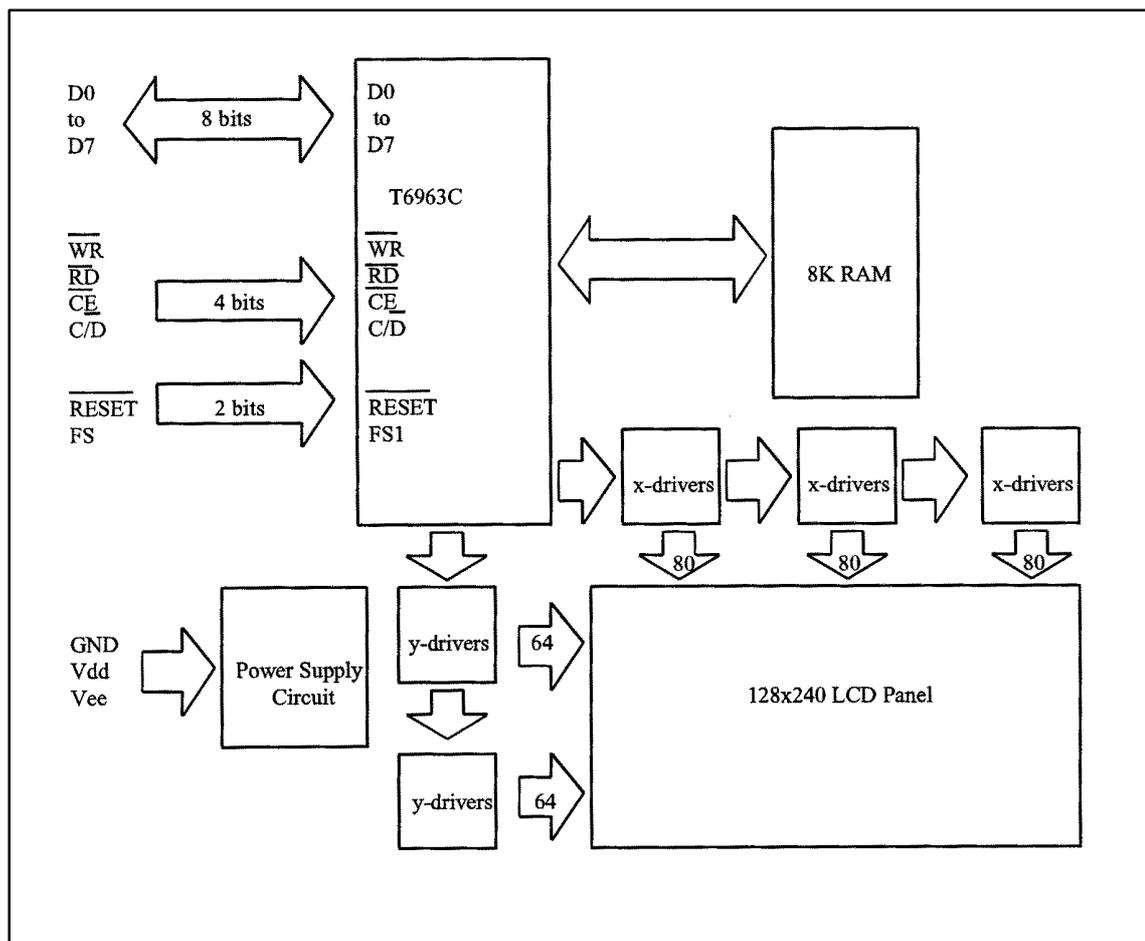


Figura 7.1

8.- DEFINICIÓN FUNCIONAL:

La definición funcional del chip puede relatarse en los puntos que a continuación se muestran:

- Después de alimentar el chip, es necesario provocar un reset. El reset se consigue mediante el oscilador de cuarzo con aproximadamente cinco flancos positivos. Cuando esto sucede, la oscilación cesa. Esto se hace para que la pantalla LCD sea protegida frente a posibles corrientes en alterna que dañarían el sistema.

- Cuando pulsamos el botón de reset el cursor que nos marca la columna o línea donde estábamos se borra.

- El registro "Status" debe ser chequeado antes de enviar datos o comandos a la pantalla, sobre todo se debe comprobar que el bit más significativo de dicho registro esté a cero. El hecho de que esto no suceda así supone una operación errónea que puede desembocar en una interrupción.

- Los registros STA0 y STA1 deben chequearse de la misma manera, teniendo en cuenta que cuando se ejecuta un comando puede suceder que se dé un error en la transmisión de los datos.

- Dicho chip sólo tiene la posibilidad de mandar un byte en cada tiempo de reloj. Es imposible que mande más en el mismo tiempo o ciclo de máquina.

- Cuando se esté utilizando el chip para enviar un comando con datos a la pantalla gráfica es importante que se mande primero el dato y a continuación que se ejecute el comando pertinente.

- El código utilizado para implementar el sistema con el chip T6963C es diferente al código ASCII convencional.

9.- EL INTERFAZ UTILIZADO CON LA MEMORIA RAM EXTERNA:

La memoria RAM externa es utilizada para almacenar datos del display (texto, gráficos y datos externos). Con el escaneo simple de datos, los datos de texto, gráficos y datos externos pueden ser libremente ubicados en el área de memoria que es de 64Kbits como máximo.

Sin embargo, con el escaneo doble de datos, los datos se parten en dos segmentos denominados LCDI y LCDII, cada una de 32Kbits como máximo y en las direcciones de 0000H a 7FFFH la primera y de 8000H a FFFFH la segunda. Los datos de texto, gráficos y datos externos (CG: generador de caracteres) pueden ser libremente ubicados en LCDI. En LCDII las mismas direcciones deben ser ubicadas igual que en LCDI, excepto la dirección "ad15". Dicha dirección ("ad15") determina la selección entre LCDI y LCDII.

Ello puede utilizar la dirección decodificada de las señales ce0 (0000 a 07FFFH), ce1 (0800 a 0FFFFH) de poco más de 4Kbits. De esta forma, ce0 y ce1 admiten decodificaciones de las direcciones situadas en los rangos de 0000 a 07FFFH y de 0800 a 0FFFFH respectivamente con una memoria aproximada de 4Kbits.

A continuación muestro en los dos gráficos que siguen, tanto el interfaz RAM con un **escaneo simple** de los datos (**Figura 9.1a**), como el interfaz RAM asociado al **escaneo doble** de los datos(**Figura 9.1b**).



Figura 9.1a

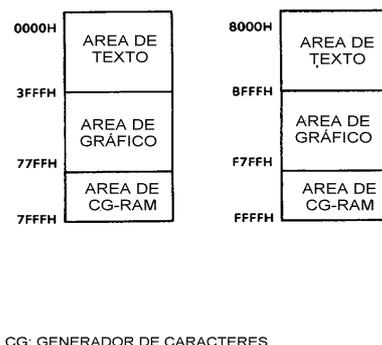


Figura 9.1b

Del mismo modo también se presentan a título informativo los **diagramas de tiempos** del modo de lectura (**Figura 9.2**) y de escritura (**Figura 9.3**) en la memoria RAM externa citada anteriormente.

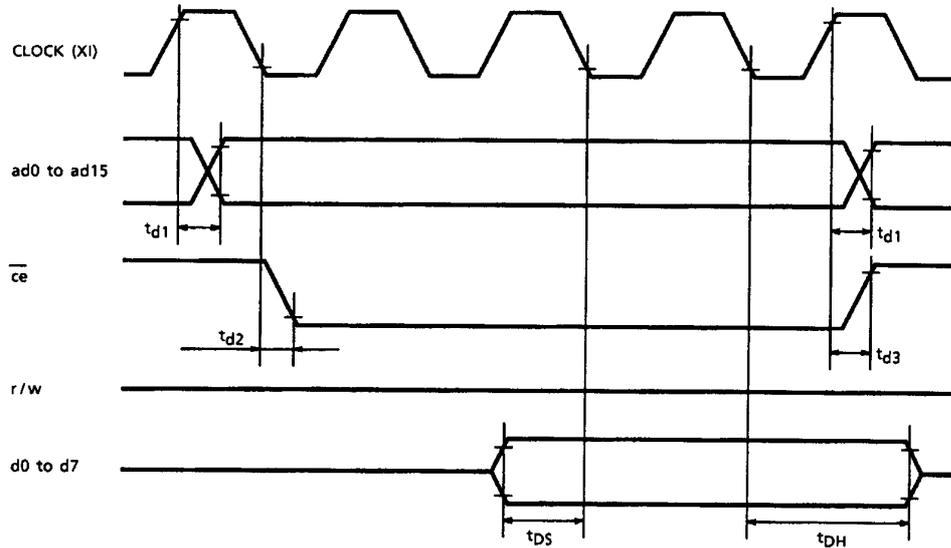


Figura 9.2

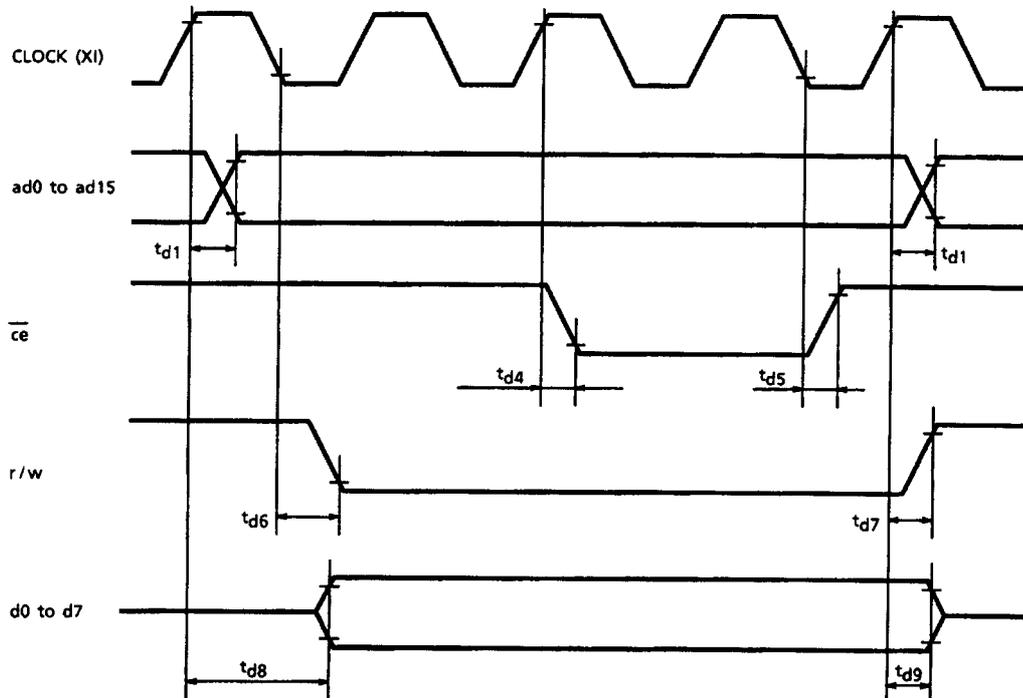


Figura 9.3

10.- COMUNICACIÓN CON LA MPU:

10.1 MODO STATUS PARA LECTURA:

El registro "Status" debe ejecutarse antes de que los datos sean leídos o escritos. Para ello, a continuación se definen los comandos de dicho registro:

```
RD   ->  L
WR   ->  H
CE   ->  L
C/D  ->  H
D0-D7 -> Palabra Status
```

El cuadro que se muestra a continuación nos indica el formato de las distintas palabras del registro Status:

MSB				LSB			
STA7	STA6	STA5	STA4	STA3	STA2	STA1	STA0
D7	D6	D5	D4	D3	D2	D1	D0

Orden desde más peso hasta menos peso

STA0	Chequea la capacidad de ejecutar el comando.	0: Desactivado 1: Activado
STA1	Chequea la capacidad de leer o escribir datos.	0: Desactivado 1: Activado
STA2	Chequea la capacidad de que el modo Auto lea datos.	0: Desactivado 1: Activado
STA3	Chequea la capacidad de que el modo Auto escriba datos.	0: Desactivado 1: Activado
STA4	No utilizado.	
STA5	Chequea la capacidad del controlador de operaciones.	0: Desactivado 1: Activado
STA6	Flag de error. Es utilizado para los comandos Screen Peek y Screen Copy.	0: No error 1: Error
STA7	Chequea la condición de parpadeo.	0: Display off 1: Display normal

Diversas funciones de "Status"

10.2 NOTAS:

A) Es necesario chequear STA0 y STA1 al mismo tiempo. Existe la posibilidad de que suceda una operación incorrecta que conlleve a una interrupción del hardware.

B) Para el modo Auto mencionado en la tabla anterior, sólo sirven las palabras STA2 y STA3 del registro Status, mientras que las palabras STA0 y STA1 no son válidas para dicho modo.

C) Cuando utilizamos el comando MSB=0, se debe ejecutar el comando Status Read. Así, la interrupción del hardware suele ocurrir durante el período del cálculo de la dirección (al final de cada línea). Si el MSB=0 se manda al T6963C durante dicho período, el T6963C entra en el modo de espera ("Wait"). Por ello, si esto sucede, tenemos la posibilidad de que el comando o el dato en cuestión no sea recibido.

Vemos en la **Figura 10.1** el **diagrama del control del modo Status**.

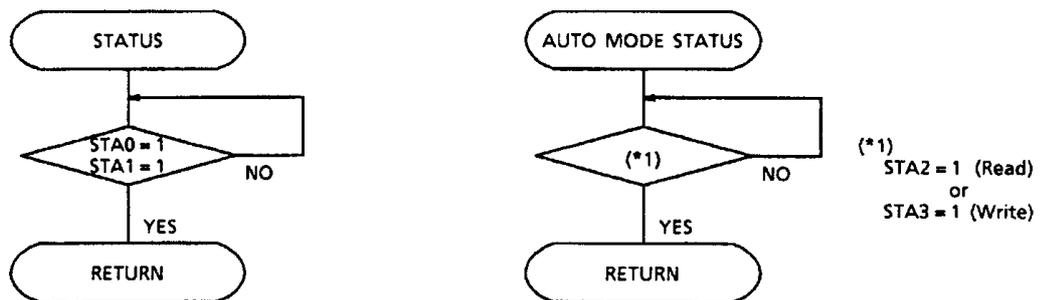


Figura 10.1

10.3 COLOCACIÓN DE LOS DATOS:

Cuando usamos el T6963C, primero se fijan los datos y posteriormente se fijan los comandos asociados a dichos datos.

Veamos a continuación el **procedimiento de envío de comandos**. Para ello mostramos los diagramas correspondientes:

a) El caso de 1 dato (**Figura 10.2**)

b) El caso de 2 datos (**Figura 10.3**)

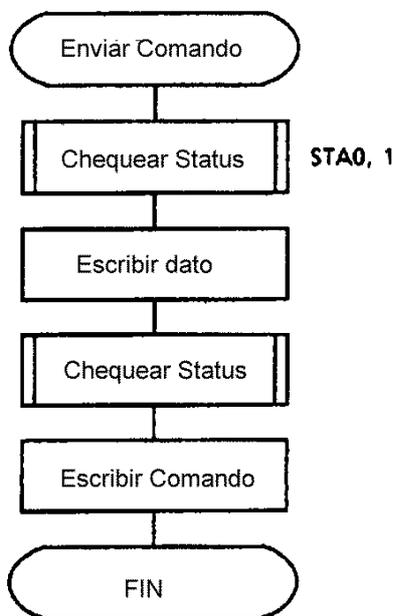


Figura 10.2

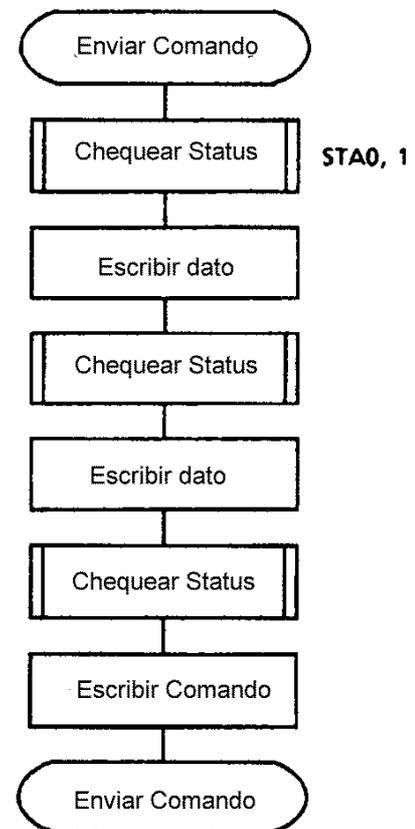


Figura 10.3

11.- MAPA DE CÓDIGOS:

Voy a representar ahora en la **Figura 11.1** el mapa de códigos de la CG-ROM que se pueden implementar.

MSB \ LSB	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
1	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
2	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
3	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
4	+	^	^	^	^	^	^	^	^	^	^	^	^	^	^	^
5	-	^	^	^	^	^	^	^	^	^	^	^	^	^	^	^
6	^	^	^	^	^	^	^	^	^	^	^	^	^	^	^	^
7	^	^	^	^	^	^	^	^	^	^	^	^	^	^	^	^

Figura 11.1

12.- ESQUEMA ELECTRÓNICO FINAL DE LA PANTALLA:

Finalmente, para terminar con el estudio del chip T6963C voy a mostrar el diagrama de bloques total de una posible pantalla gráfica (**Figura 12.1**). En él se pueden ver, entre otros, el T6963C, además de todos los demás chips que componen la pantalla.

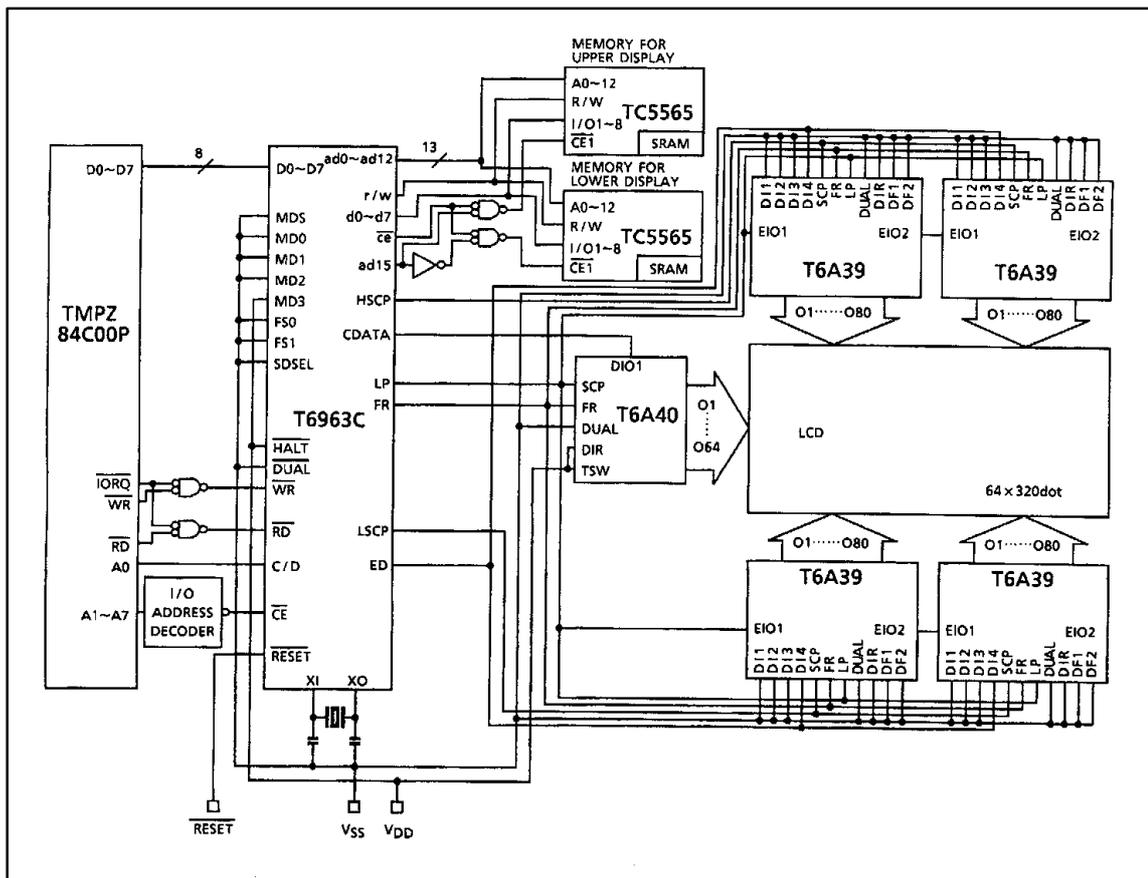


Figura 12.1

Para finalizar el estudio de la pantalla gráfica, y por tanto de su chip digamos “clave”, muestro una fotografía en la **Figura 12.2** donde aparece la parte trasera de las dos pantallas (de dos tamaños distintos) donde pueden verse físicamente todos los chips que llevan implementados, dibujados ya en el esquema de la **Figura 12.1**.

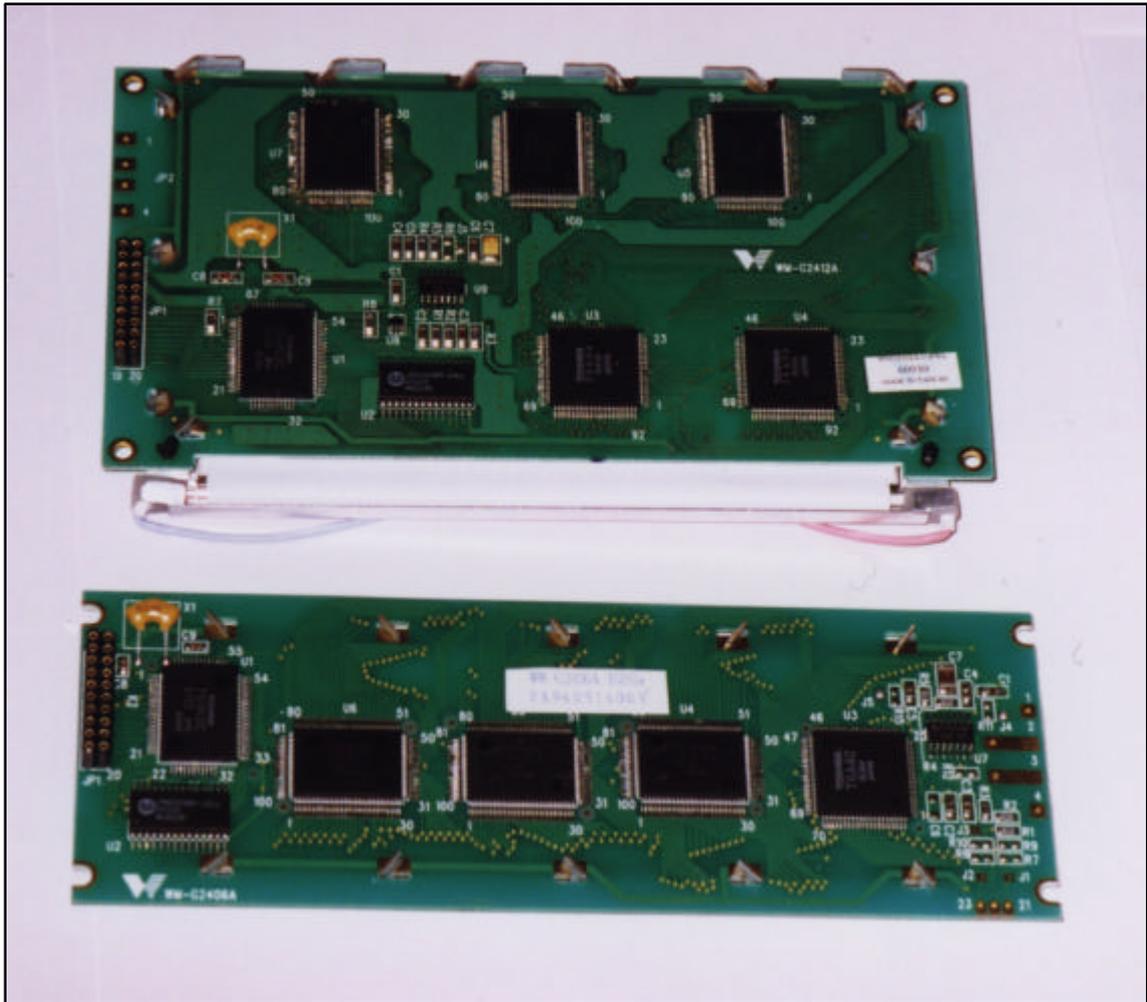


Figura 12.2

13.- COMANDOS:

Para el correcto funcionamiento de las pantallas que contienen este Chip y el buen aprovechamiento de todas sus posibilidades, es necesario conocer todas sus funciones y los comandos que debemos enviar desde el controlador para habilitar cada una de ellas y sacar el mayor jugo posible de estos displays. Por esta razón, mostro a continuación una explicación de todos los comandos que se le pueden enviar.

En primer lugar, se ofrece una primera idea de la capacidad de las pantallas que son manejadas por el T6963C de "TOSHIBA" observando la tabla que aparece en la página siguiente (**Tabla 1**), donde se incluyen todos los comandos y una pequeña descripción de su función.

Si queremos conocer con más detalle el modo de empleo y el funcionamiento de alguno de los comandos en concreto, en las hojas sucesivas ofrezco una explicación más exhaustiva de cada uno de ellos que espero facilitará su comprensión a cualquier usuario que desee utilizar las pantallas.

COMANDO	CÓDIGO	DATO1	DATO2	FUNCIÓN
POINTER SET	00100001	Dirección en X	Dirección en Y	Sitúa el cursor
	00100010	Dato	00H	Registro de Offset
	00100100	Dirección baja	Dirección alta	Puntero de dirección
CONTROL WORD SET	01000000	Dirección baja	Dirección alta	Dirección del texto
	01000001	Columnas	00H	Área del texto
	01000010	Dirección baja	Dirección alta	Dirección del gráfico
	01000011	Columnas	00H	Área del gráfico
MODE SET	1000X000	-	-	Modo "OR"
	1000X001	-	-	Modo "EXOR"
	1000X011	-	-	Modo "AND"
	1000X100	-	-	"Atributo de Texto"
	10000XXX	-	-	Modo CG-ROM interna
	10001XXX	-	-	Modo RAM externa
DISPLAY MODE	10010000	-	-	Apagar pantalla
	1001XX10	-	-	Cursor sin parpadeo
	1001XX11	-	-	Cursor con parpadeo
	100101XX	-	-	Texto habilitado
	100110XX	-	-	Gráficos habilitados
	100111XX	-	-	Gráficos + Texto
CURSOR PATTERN SELECT	10100000	-	-	1 Línea
	10100001	-	-	2 Líneas
	10100010	-	-	3 Líneas
	10100011	-	-	4 Líneas
	10100100	-	-	5 Líneas
	10100101	-	-	6 Líneas
	10100110	-	-	7 Líneas
	10100111	-	-	8 Líneas
DATA AUTO READ/ WRITE	10110000	-	-	Auto Escritura
	10110001	-	-	Auto Lectura
	10110010	-	-	Auto Reset
DATA READ WRITE	11000000	Dato	-	Escribir e incrementar
	11000001	-	-	Leer e incrementar
	11000010	Dato	-	Escribir y decrementar
	11000011	-	-	Leer y decrementar
	11000100	Dato	-	Escribir y no variar
	11000101	-	-	Leer y no variar

SCREEN	11100000	-	-	Screen Peek
PEEK/COPY	11101000	-	-	Screen Copy
BIT	11110XXX	-	-	Bit a cero
SET/RESET	11111XXX	-	-	Bit a uno
	1111X000	-	-	Bit0 (LSB)
	1111X001	-	-	Bit2
	1111X010	-	-	Bit3
	1111X011	-	-	Bit4
	1111X100	-	-	Bit5
	1111X101	-	-	Bit6
	1111X110	-	-	Bit7
	1111X111	-	-	Bit8 (MSB)

Tabla 1

**NOTA: Los bits representados por una X, se detallan en la explicación concreta de cada comando.*

13.1. POINTER SET:

13.1.1 Cursor pointer set:

Descripción	HEX	D7	D6	D5	D4	D3	D2	D1	D0
Dato 1	0-7F	*	Posición vertical del cursor						
Dato 2	0-1F	*	*	*	Posición horizontal del cursor				
Comando	21	0	0	1	0	0	0	0	1

El comando “Cursor Pointer Set” sirve para situar el puntero de caracteres en un determinado lugar de la pantalla, de tal manera que a la hora de escribir alguna letra, aparezca en esa posición y en las sucesivas. Para ello, es necesario enviar a la pantalla dos datos acompañados del comando que aparece en la tabla.

El primero de los datos enviados, indica la columna en la que queremos que se sitúe el cursor mientras que el segundo se refiere a la fila. Según sea el tamaño de la pantalla, el número de filas y de columnas será mayor o menor, así que, en la columna de la tabla que contiene los datos en formato hexadecimal (HEX), aparecen valores para una de las pantallas he manejado.

13.1.2 Offset register set:

Descripción	HEX	D7	D6	D5	D4	D3	D2	D1	D0
Dato 1	0-1F	*	*	*	Dirección de memoria CG-RAM				
Dato 2	00	0	0	0	0	0	0	0	0
Comando	22	0	0	1	0	0	0	1	0

El comando “Offset register set” sirve para indicar a la pantalla la dirección donde debe comenzar el área de la RAM generadora de caracteres. De esta manera, cuando queremos escribir una letra, primero se guarda en esa dirección de la memoria RAM que incorpora la pantalla y después se visualiza en la propia pantalla.

Los cinco bits de menos peso del primer dato, coinciden con los cinco bits de mayor peso de la dirección inicial para la CG-RAM. El segundo dato, es un cero. Para más detalles, consultar el apartado que trata sobre la utilización de la memoria RAM de la pantalla.

13.1.3 Addres Pointer Set:

Descripción	HEX	D7	D6	D5	D4	D3	D2	D1	D0
Dato 1	0-FF	Dirección del puntero (8 bits de menos peso)							
Dato 2	0-FF	Dirección del puntero (8 bits de más peso)							
Comando	24	0	0	1	0	0	1	0	0

El comando "Addres Pointer Set", se usa para especificar la dirección inicial en la que se escribe o se lee de la memoria VRAM que incluye la pantalla. Debe coincidir con un lugar en el área de RAM especificado por el mapa de memoria de una pantalla determinada, por lo que los dos datos que acompañan el comando, dependerán del tipo de pantalla que tengamos entre manos. Para más detalles, hay que estudiar las especificaciones individuales de cada módulo LCD.

13.2. CONTROL WORD SET :

13.2.1. Text Home Address Set:

Descripción	HEX	D7	D6	D5	D4	D3	D2	D1	D0
Dato 1	0-FF	Dirección del texto (8 bits de menos peso)							
Dato 2	0-FF	Dirección del texto (8 bits de más peso)							
Comando	40	0	1	0	0	0	0	0	0

El comando "Text Home Address Set" indica la dirección inicial de la VRAM para datos de texto. Estos datos guardados en Text Home (TH), se visualizarán en la parte superior izquierda de la zona de caracteres (the home position).

13.2.2. Text Area Set:

Descripción	HEX	D7	D6	D5	D4	D3	D2	D1	D0
Dato 1	0-FF	Número de columnas de caracteres							
Dato 2	00	0	0	0	0	0	0	0	0
Comando	41	0	1	0	0	0	0	0	1

El comando "Text Area Set" define el número de columnas de texto que queremos que haya en la VRAM. Se puede seleccionar independientemente del número de caracteres por línea fijados por hardware en el chip controlador T6963C. A pesar de esta posibilidad, se suele fijar el valor del primer dato para que coincida con el número de caracteres por línea que permite la pantalla que manejamos.

Por ejemplo, en una pantalla de 64x240 si el tamaño de letra es 6x8, debemos fijar TA=28H; si el tamaño de letra es 8x8 fijaremos TA=1EH, ya que cabrán menos caracteres en la misma línea en el segundo caso.

13.2.3. Graphic Home Address Set:

Descripción	HEX	D7	D6	D5	D4	D3	D2	D1	D0
Dato 1	0-FF	Dirección de gráficos (8 bits de menos peso)							
Dato 2	0-FF	Dirección de gráficos (8 bits de más peso)							
Comando	42	0	1	0	0	0	0	1	0

El comando “Graphic Home Address Set”, define la dirección inicial para mostrar gráficos a través de la VRAM. Los datos que se guardan en esta zona de memoria (Graphic Home: GH), aparecerán en la parte superior izquierda de nuestra pantalla, dependiendo del tamaño de fuente que esté seleccionado.

Si la función de Atributo está habilitada, este comando indica la dirección inicial para la zona de Atributo RAM.

13.2.4. Graphic Area Set:

Descripción	HEX	D7	D6	D5	D4	D3	D2	D1	D0
Dato 1	0-FF	Número de columnas							
Dato 2	00	0	0	0	0	0	0	0	0
Comando	42	0	1	0	0	0	0	1	0

El comando “Graphic Area Set”, indica el número de columnas de datos gráficos para el área de gráficos de la VRAM. Al igual que en el caso de texto, este valor se puede fijar independientemente del número de caracteres por línea que puede mostrar la pantalla. Sin embargo, se suele elegir el número de columnas para que coincida con ese valor.

13.3. MODE SET :

Descripción	D7	D6	D5	D4	D3	D2	D1	D0
CG-ROM	1	0	0	0	0	Se detallan en la siguiente tabla.		
CG-RAM	1	0	0	0	1			

El comando "Mode Set" indica el modo de funcionamiento de la pantalla. Podemos escoger entre usar la memoria CG-ROM interna o la CG-RAM externa. También se puede utilizar ambas al mismo tiempo. Para ello se utiliza el bit D3 del comando que se le envía a la pantalla.

Además de eso, se puede escoger la lógica de funcionamiento de la pantalla de entre las principales posibilidades manejando los bits D2-D0 del mismo comando. Se observa esa posibilidad en la siguiente tabla.

Descripción	D7	D6	D5	D4	D3	D2	D1	D0
"OR"	1	0	0	0	*	0	0	0
"EXOR"	1	0	0	0	*	0	0	1
"AND"	1	0	0	0	*	0	1	1
Sólo Texto	1	0	0	0	*	1	0	0

13.4. DISPLAY MODE SET:

Descripción	D7	D6	D5	D4	D3	D2	D1	D0
Apagar	1	0	0	1	0	0	0	0
Cursor sin parpadeo	1	0	0	1	*	*	1	0
Cursor con parpadeo	1	0	0	1	*	*	1	1
Sólo Texto	1	0	0	1	0	1	*	*
Sólo Gráficos	1	0	0	1	1	0	*	*
Texto y Gráficos	1	0	0	1	1	1	*	*

El comando "Display Mode Set" se encarga de seleccionar el modo de funcionamiento de la pantalla de entre todas las posibilidades. En la tabla de arriba se observan todas las posibilidades.

Para la selección de cada una de ellas hay que manejar los cuatro últimos bits del comando de la siguiente forma:

- ✓ Cuando los cuatro bits de menor peso (D3-D0) están a cero, la pantalla está desactivada.
- ✓ D3 y D2 se usan para indicar si se quiere utilizar la pantalla en modo Gráfico, en modo de Texto o en combinación de ambos a la vez.
- ✓ D1 y D0 sirven para seleccionar las características que se desea que tenga el cursor de la pantalla, ya sea de texto o de gráficos.
- ✓ Con un solo comando, se envían todos estos datos a la pantalla que sólo necesita interpretarlos.

*NOTA: Después de un Reset, los cuatro bits de menor peso están a cero por defecto.

13.5. CURSOR PATTERN SELECT:

Descripción	D7	D6	D5	D4	D3	D2	D1	D0
1 Línea	1	0	1	0	0	0	0	0
2 Líneas	1	0	1	0	0	0	0	1
3 Líneas	1	0	1	0	0	0	1	0
4 Líneas	1	0	1	0	0	0	1	1
5 Líneas	1	0	1	0	0	1	0	0
6 Líneas	1	0	1	0	0	1	0	1
7 Líneas	1	0	1	0	0	1	1	0
8 Líneas	1	0	1	0	0	1	1	1

El comando "Cursor Pattern Select" sirve para elegir el número de líneas de grosor que se desea que tenga el cursor de la pantalla cuando está habilitado (explicado en el apartado anterior). Para ello, se manipulan los tres bits de menor peso de tal manera que el cero en binario corresponde a un cursor de una línea de grosor y el siete a un grosor de ocho. Los valores intermedios son los lógicos que se muestran en la tabla adjunta.

13.6. DATA AUTO READ/WRITE:

Descripción	D7	D6	D5	D4	D3	D2	D1	D0
Data Auto Write	1	0	1	1	0	0	0	0
Data Auto Read	1	0	1	1	0	0	0	1
Auto Mode Reset	1	0	1	1	0	0	1	*

Los comandos “Data Auto Read/Write” se utilizan cuando se transfieren bloques de información hacia o desde la VRAM. El uso de estos comandos facilita la escritura porque evita que tengamos que incrementar la dirección donde escribimos o leemos modificando el comando “Address Pointer Set” (Sección 13.1.3) cada vez que tenemos un nuevo dato. Al ejecutar los dos primeros comandos, la dirección de memoria se autoincrementa cada vez que leemos o escribimos de nuevo.

Una vez terminada la transmisión o recepción de todos los datos es necesario ejecutar el comando “Auto Mode Reset” para desactivar esta utilidad y volver al estado normal de funcionamiento de la pantalla.

13.7. DATA READ/WRITE:

Descripción	D7	D6	D5	D4	D3	D2	D1	D0
Data Write (Incremento)	1	1	0	0	0	0	0	0
Data Read (Incremento)	1	1	0	0	0	0	0	1
Data Write (Decremento)	1	1	0	0	0	0	1	0
Data Read (Decremento)	1	1	0	0	0	0	1	1
Data Write (Igual)	1	1	0	0	0	1	*	0
Data Read (Igual)	1	1	0	0	0	1	*	1

Estos comandos se utilizan para escribir datos en la memoria VRAM o para leer información de la misma. Se envían después de indicar la dirección de memoria que se va a utilizar mediante el comando "Address Pointer Set" (Sección 13.1.3) para que el controlador sepa el lugar donde escribir o de donde leer.

Además, la dirección de memoria se puede incrementar y automáticamente con cada lectura/escritura o se puede dejar como está para cambiarla después manualmente.

- ✓ El bit de menor peso D0 sirve para indicar si se quiere escribir o leer de la VRAM.

$D0 = 0 \Rightarrow \textit{Escritura}$ $D0 = 1 \Rightarrow \textit{Lectura}$
--

- ✓ El bit D1 se usa para indicar si la dirección tiene que incrementarse o decrementarse cada vez que escribimos/leemos.

$D1 = 0 \Rightarrow \textit{Incremento}$ $D1 = 1 \Rightarrow \textit{Decremento}$
--

- ✓ El bit D2 indica si queremos que la dirección de memoria no se modifique con cada dato nuevo. Es importante destacar que cuando D2 está a uno, independientemente del valor de D1, la dirección no se incrementa ni decrementa. Podemos decir que D2 está por encima de D1 en jerarquía de prioridades.

$D2 = 0 \Rightarrow \textit{Según D1}$ $D2 = 1 \Rightarrow \textit{Sin Cambios}$

13.8. SCREEN PEEKING:

Descripción	D7	D6	D5	D4	D3	D2	D1	D0
Screen Peeking (E0H)	1	1	1	0	0	0	0	0

El comando "Screen Peeking" se utiliza para transferir 1 byte de datos visualizado por la pantalla a una zona de memoria que será un almacén temporal de datos de donde se podrá leer posteriormente utilizando el comando "Data Read" (Sección 13.7). Es muy útil para leer la combinación lógica de texto y gráficos escogida en la pantalla.

Es importante destacar que hay que consultar el bit STA6 del registro de estado cada vez que queremos ejecutar el comando "Screen Peeking" porque si este flag está a uno, el comando será ignorado por el T6963C que controla la pantalla.

13.9. SCREEN COPY:

Descripción	D7	D6	D5	D4	D3	D2	D1	D0
Screen Copy (E8H)	1	1	1	0	1	0	0	0

El comando "Screen Copy" se utiliza para copiar una fila de datos visualizados por pantalla y guardarlos en el área de RAM reservada a gráficos que habrá sido especificada mediante el comando "Address Pointer Set" (Sección 13.1.3).

Sin embargo, no se puede utilizar este comando si la fila que se quiere copiar posee algún dato con atributo de texto.

Al igual que en el caso anterior, el bit STA6 del registro de estado debe ser comprobado antes de utilizar el comando porque si está a uno, el controlador ignorará la señal.

13.10. BIT SET/RESET:

Descripción	D7	D6	D5	D4	D3	D2	D1	D0
Bit Reset(F0H – F7H)	1	1	1	1	0			
Bit Set (F8H - FFH)	1	1	1	1	1			

Estos comandos se usan para poner a uno o a cero un bit cualquiera de la RAM. Modifica uno de los bits (el que se le indica) que contiene el byte señalado por el "Addres Pointer Set" (Sección 13.1.3). No podemos cambiar más de un bit en el mismo byte de forma simultánea sino que hay que hacerlo uno por uno.

- ✓ Con el bit D3 del comando se indica si se quiere poner a uno o a cero el bit que modificamos.

$$\begin{array}{l} D3 = 0 \Rightarrow \text{Poner a cero} \\ D3 = 1 \Rightarrow \text{Poner a uno} \end{array}$$

- ✓ Con los 3 bits de menor peso se selecciona la posición del bit que se quiere poner a uno o a cero. De esta manera, si se ponen los tres a cero, se manipula el bit de menor peso (LSB) del byte indicado, mientras que si se ponen los tres a uno, se manipula el bit de mayor peso (MSB) de dicho byte.

14.- GENERADOR DE CARACTERES:

14.1 CG-ROM (Interno):

Las pantallas controladas por el chip T6963C ofrecen la posibilidad de generar caracteres para ser visualizados por el usuario. Una de las posibilidades es escoger uno de los 128 caracteres diferentes que podemos generar internamente con la propia pantalla. La otra opción es la de crear nuestros propios caracteres y guardarlos en una CG-RAM.

Para ampliar la información sobre el generador de caracteres interno del propio chip, consultar el apartado 11 de este manual.

14.2 CG-RAM (Externo):

Como se ha indicado anteriormente, existe la posibilidad de definir una serie de caracteres personalizados usando unos códigos apropiados.

Para usar la CG-RAM externa es necesario ejecutar el comando "Offset Register Set" (Sección 13.1.2) para especificar la localización que deseamos dentro de la RAM. Después, el usuario debe programar la CG-RAM con los datos necesarios para generar el carácter deseado. Una vez que se ha programado esa zona de memoria, se puede comenzar a utilizar el carácter generado de forma individual o conjuntamente con el generador interno de caracteres. Para escoger la utilización de una o de otra se ejecuta el comando "Mode Set" (Sección 13.3).

Por poner un ejemplo que se ha comprobado, se va a generar un carácter que sea una almohadilla '#' para poder visualizarlo por pantalla. La secuencia de comandos y datos que hay que enviar es la siguiente:

- ✓ Colocamos el comando "Addres Pointer Set" con el valor hexadecimal 1C78H para que comencemos a escribir en la dirección 1C00H de la memoria RAM.
- ✓ A continuación se envían ocho bits de datos consecutivos que representan el carácter utilizando el comando Data Write ocho veces consecutivas.
- ✓ Los ocho bits necesarios para generar la almohadilla son los siguientes:
[06,06,1F,06,06,1F,06,00].

14.3 FUNCIÓN ATRIBUTO:

Esta función que ofrecen las pantallas se utiliza para activar la imagen negativa (fondo negro) y/o el parpadeo de la pantalla.

Para poder usar esta opción, es necesario primero, reasignar el comando "Graphic Home Address Set" a la primera dirección de la zona de atributos de la memoria RAM usando el comando que permite escribir en la memoria RAM interna. Cabe destacar que los datos gráficos no pueden ser visualizados mientras se esté utilizando la función de atributo. La siguiente tabla define las funciones del comando Attribute Data.

Descripción	D7	D6	D5	D4	D3	D2	D1	D0
Normal	*	*	*	*	0	0	0	0
Inverso	*	*	*	*	0	1	0	1
Inhibir pantalla	*	*	*	*	0	0	1	1
Parpadeo normal	*	*	*	*	1	0	0	0
Parpadeo inverso	*	*	*	*	1	1	0	1
Inhibir pantalla.	*	*	*	*	1	0	1	1

El atributo para el primer carácter guardado en la zona de texto de la memoria RAM debe escribirse en el primer byte de la zona de la RAM habilitada para atributos y así sucesivamente.