

PROGRAMACIÓN GRÁFICA

Lenguaje Ensamblador HP Saturno

Por Marcos Navarro

PREVIEW

PARA CALCULADORAS HP49G, HP49G+ y HP50G

PROGRAMACIÓN GRÁFICA

Lenguaje Ensamblador HP Saturno

Por Marcos Navarro

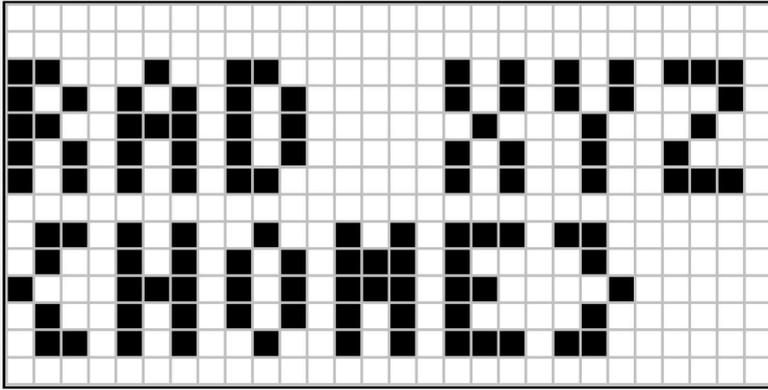
PARA CALCULADORAS HP49G, HP49G+ y HP50G

CONTENIDO

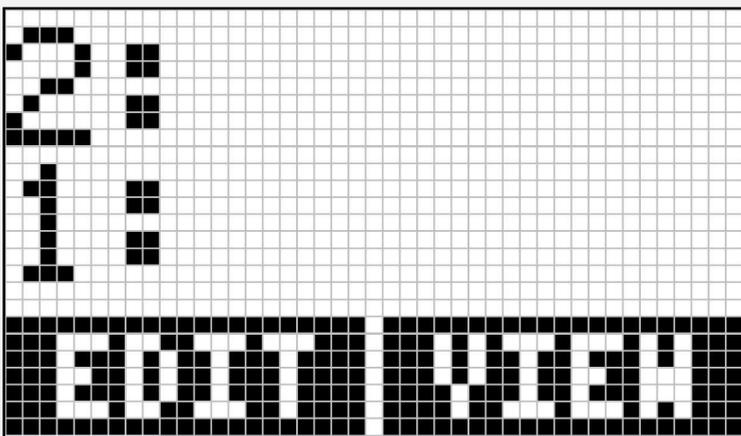
1. NOCIONES PRELIMINARES	1
1.1 El microprocesador HP Saturno	3
1.2 ¿Cuándo usar Lenguaje Ensamblador?	3
1.3 ¿Por qué estudiar este lenguaje?	3
1.4 Recursos necesarios	4
1.5 Editar Objetos Gráficos en una PC	4
1.6 Los Objetos Gráficos o Grob	5
1.7 Representación interna de un Grob	7
1.8 Grobs y bytes	13
E1 Compilando y descompilando Objetos Gráficos	19
H1 El Chip Yorke	20
2. LOS GRÁFICOS DE PANTALLA	21
2.1 Áreas de la Pantalla	23
2.2 Los Gráficos de Pantalla	23
2.3 Los Registros para gráficos del Área Principal	23
2.4 Los Registros del Área Secundaria o Área del Menú	24
2.5 Concepto de Puntero	24
2.6 Mapa de bits Principal	25
2.7 Mapa de bits Secundario	26
2.8 Direcciones Especiales	26
2.9 La subrutina D0->Row1	27
2.10 Cargando un Grob en la Pantalla	28
2.11 Leyendo datos de un grob dado en la Pila	29
E2 Usando el Depurador o Debugger para Examinar los Registros	32
Experimento 2A	32
Experimento 2B	34
Experimento 2C	37
H2 El Sistema que controla la Pantalla	38
3. REFRESCADO DE LA PANTALLA	39
3.1 El Margen Derecho de Objetos Gráficos	41
3.2 Refresco de la Pantalla (Parte I)	43
3.3 Desplazamiento de la Ventana (Parte I)	46
3.4 Refresco de la Pantalla (Parte II)	48
3.5 Desplazamiento de la Ventana (Parte II)	49
3.6 Refresco de la Pantalla (Resumen)	50
3.7 El Gráfico de la Pila y la Pantalla	52
3.8 El Grob de la Pantalla de Gráficas o PICT Grob	53
E3 Registros de Margen Derecho y el Margen Izquierdo para la Pantalla	54
Experimento 3A	56
Experimento 3B	56

Experimento 3C	56
Experimento 3D	56
Experimento 3E	57
Experimento 3F	58
H3 Conexión de la Pantalla	60
4. LOS ANUNCIADORES	63
4.1 Los Anunciadores	65
4.2 Registro de control	65
4.3 Programando los Anunciadores	66
H3 La Pantalla LCD	70
Partes de la Pantalla LCD	70
Los Polarizadores de luz	71
5. DESPLAZAMIENTO DE VENTANA	73
5.1 Desplazamiento de Ventana (Parte II)	75
5.2 Desplazamientos de Ventana (Parte III)	79
5.3 ¿Cómo funciona?	80
5.4 Cuando el Margen Izquierdo es mayor que 3	82
E5 Desplazamiento inteligente de la Ventana	85
H5 Paneles de la Pantalla	87
Panel frontal o panel de columnas	87
Panel trasero o panel de filas	87
6. LOS SPRITES	89
6.1 ¿Qué es un Sprite?	91
6.2 Modos de inserción de Sprites	91
6.3 Técnicas de recuperación de Sprites (Data Retrive)	91
6.4 Leyendo los datos desde la Pila	92
6.5 Insertando el Sprite en el código	93
6.6 Insertando un Sprite en el Stack Grob	94
6.7 Columnas de Nibbles	96
6.8 Utilizando otros modos de inserción de Sprite	99
6.9 Copiando una zona de la Pantalla	101
6.10 Métodos para copiar una zona de la Pantalla, dadas las coordenadas	105
6.11 Métodos para insertar un grob en una zona de la Pantalla	110
H6 Sello perimetral y conexión entre paneles	115
Líneas de conducción entre paneles	115
El Sello perimetral	115
7. TRABAJANDO A NIVEL DE BITS	117
7.1 Trabajando a nivel de bits	119
7.2 Operaciones lógicas con Registros	119
7.3 Desactivando pixeles específicos	119
7.4 Activando pixeles específicos	120
7.5 Desplazamiento a nivel de bits	121
7.6 Desplazamiento de un Sprite a nivel de pixeles	122
7.7 Inserción de un Sprite en la Pantalla con coordenadas x, y dadas en pixeles	130

7.8 Nuevo sistema de coordenadas	130
7.9 Codificación y decodificación de las coordenadas	132
E7 Explorando la Memoria con la Herramienta T2	137
Experimento 7A	137
Experimento 7B	138
Experimento 7C	138
8. PANTALLAS VIRTUALES	139
8.1 Pantalla Virtual	141
8.2 Insertando Sprites en una Pantalla Virtual	143
9. EL MENÚ	149
9.1 El Menú	151
9.2 Insertando un grob personalizado en el Área de Menú	151
9.3 Pantalla Virtual para el Menú	152
9.4 Insertando una etiqueta personalizada en el Área de Menú	156
E9 El Contador de Líneas	157
10. CONTRASTE DE LA PANTALLA	161
10.1 Contraste de la Pantalla	163
10.2 Utilizando el quinto bit de contraste	164
10.3 Modificando el contraste en un valor específico	165
10.4 Asignando un valor específico de contraste	166
11. GRÁFICOS EN ESCALA DE GRISES	169
11.1 Gráficos en Escala de Grises	171
11.2 ¿Cómo funcionan los Gráficos en Escala de Grises?	171
11.3 Áreas específicas de memoria para Gráficos en Escala de Grises	172
11.4 Distribución del espacio en la HP 49G	172
11.5 Distribución del espacio en la HP 50G	173
11.6 Utilizando la subrutina de Interrupción para Gráficos en Escala de Grises	175
11.7 Procedimiento General	175
ANEXO	181
EXPLORANDO EL INTERIOR DEL SISTEMA	183
I. El Primer Viaje	184
II. ¿Dónde está la RTI?	



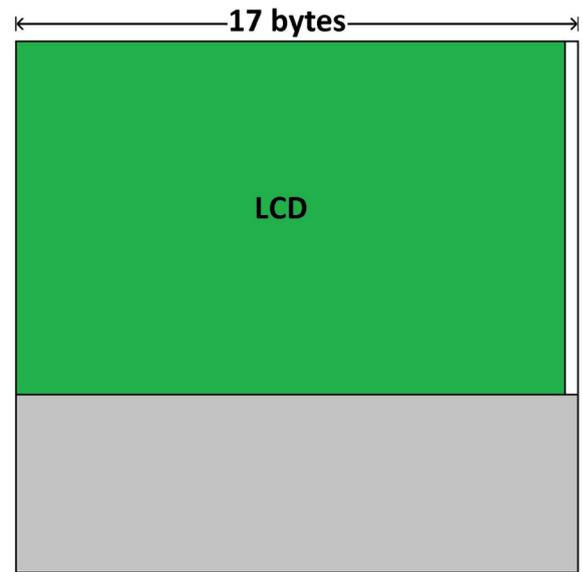
```
RAD XYZ HEX R= 'X'  
CHONES  
-----  
7:  
6:  
5:  
4:  
3:  
2:  
1:  
DOWN DOWN STACK ROL PURGE CLEAR
```



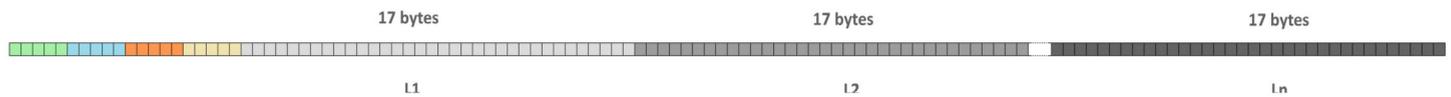
3.1 El Margen Derecho de Objetos Gráficos

Revisemos la codificación interna de un *Objeto Gráfico*. Supongamos que tenemos un *gráfico* de 131 x n píxeles.

Los primeros 20 nibbles son el *Encabezamiento* o *Header* del *Objeto Gráfico*:

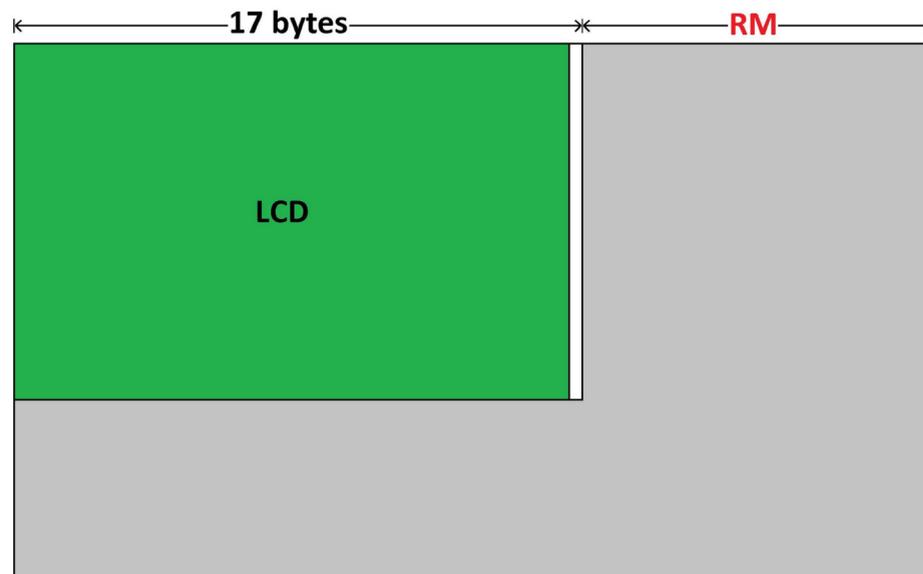


Después del *Encabezamiento* está el *Mapa de bits*, donde cada línea horizontal es codificada con 17 bytes.



Aunque 131 píxeles son menos de 17 bytes, el sistema siempre codifica el gráfico con múltiplos de 8 bits (bytes). Los bits no utilizados son completados con ceros (padding). Los bits restantes aparecen como una franja de color blanco en la imagen anterior, a la derecha de la LCD.

A veces, tenemos un *Objeto Gráfico* cuyas filas son mucho más largas que la *Pantalla LCD*:



registros de Margen Derecho y el Margen Izquierdo para la Pantalla (HP49G, HP49G+ y HP50G)

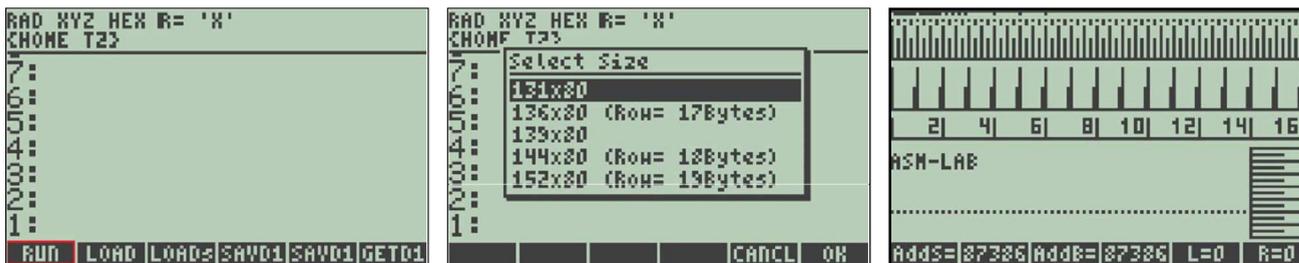
Descarga el paquete de Herramientas de Experimentación en el siguiente link:

<https://sites.google.com/view/pgsaturno50g/home>

Dentro del archivo descargado encontrarás varios programas diseñados para ayudarte a comprender mejor los temas tratados en este libro.

Para los experimentos de la presente sección, sigue los siguientes pasos:

- 1) Transfiere el archivo T2.HP a tu calculadora.
- 2) En la calculadora, ejecuta el archivo RUN, el cual está dentro del directorio T2.
- 3) Selecciona las dimensiones para el *Gráfico de Fondo*, y presiona OK.



- 4) Ahora puedes modificar los márgenes, mientras ves los efectos en la *Pantalla*. Esta *Herramienta* también puede ser utilizando para explorar la memoria de la calculadora y ver los datos representados como pixeles

TECLA	FUNCION
X	Salto de 9 filas hacia arriba
.	Salto de 9 filas hacia abajo
G	Salto de 32 filas hacia arriba
J	Salto de 32 filas hacia abajo
H	Salto de 64 filas hacia arriba
K	Salto de 64 filas hacia abajo
I	Salto de 128 filas hacia arriba
L	Salto de 128 filas hacia abajo
UP	Salto de 1 fila hacia arriba
DOWN	Salto de 1 fila hacia abajo
LEFT	Salto de 1 byte hacia arriba
RIGHT	Salto de 1 byte hacia abajo
←	Restaura el <i>Puntero de Pantalla</i>

5.1 Desplazamiento de Ventana (Parte II)

Como ya sabemos, la *Pantalla* LCD de la calculadora HP49G tiene una resolución de 131 x 64 pixeles, mientras que en la HP 50G es de 131x80 pixeles. Pero a veces el controlador de *Pantalla* está accediendo a un gráfico que es más grande que la *Pantalla* de la calculadora y, por lo tanto, no puede ser mostrado por completo en la *Pantalla*. A la parte del gráfico que podemos ver en un momento determinado lo llamaremos la *Ventana*.

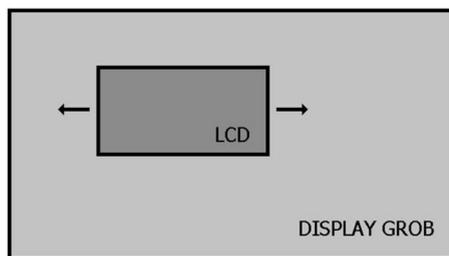
Entonces, cuando el gráfico es de mayor tamaño que la *Pantalla* de la calculadora, no podremos ver todos sus puntos al mismo tiempo, pero podemos desplazar la *Ventana* hacia las distintas zonas según necesitemos.

El desplazamiento de la *Ventana* o parte visible del gráfico puede ser de varios tipos: podemos hacer un desplazamiento horizontal o vertical, y cada uno de estos puede realizarse en dos sentidos diferentes. Así que tenemos:

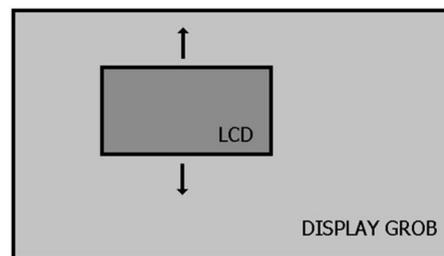
Desplazamiento de *Pantalla*:

- 1- Horizontal:
 - a- Hacia la derecha
 - b- Hacia la izquierda

- 2- Vertical
 - c- Hacia arriba
 - d- Hacia abajo



Horizontal

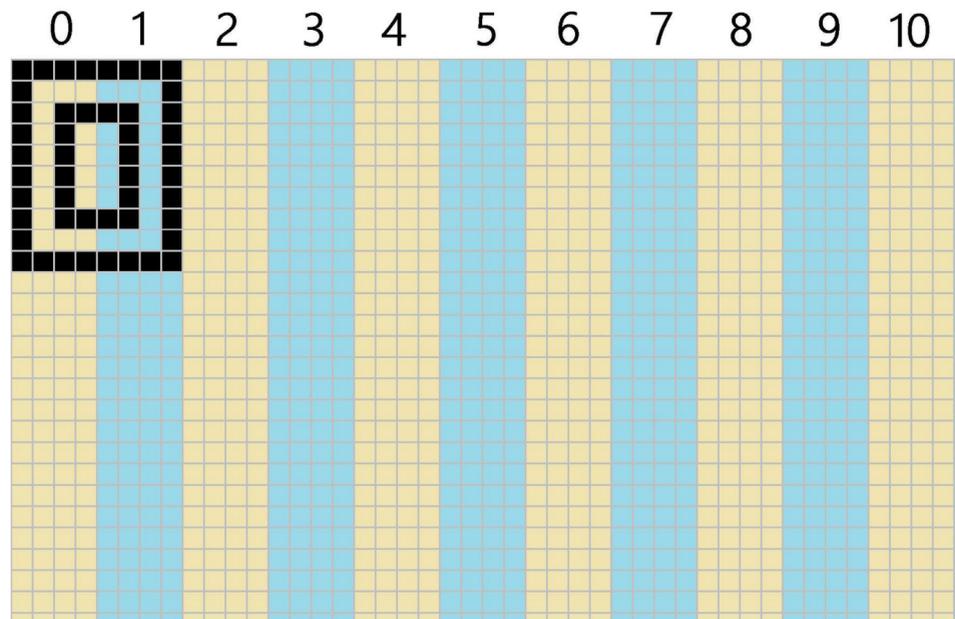


Vertical

¿Cómo sería el código para desplazar la *Ventana* verticalmente hacia abajo?

Iniciaremos analizando el caso más simple, cuando la *Ventana* se desplaza un byte (8 pixeles) hacia la derecha. Pero antes de iniciar con ese tema revisemos los siguientes:

Como vimos anteriormente, el *Ciclo de Refresco* de la *Pantalla* normalmente inicia con el *Puntero de Pantalla* apuntando hacia el primer byte del *Mapa de bits*. El controlador de la *Pantalla* leerá 17 bytes, uno a la vez, incrementando el valor del *Puntero de Pantalla* a medida que va leyendo.



Basta iniciar la carga del Sprite con D0 apuntando a otra *Columna de Nibbles* de Stack Grob para que el Sprite se muestre en esa zona. Veamos un ejemplo:

Modifiquemos el código del ejemplo anterior para que el *Sprite* se muestre en la tercera *Columna de Nibbles* del *Stack Grob*:

```

CODE
SAVE

A=DAT1 A
D1=A
D1+10
D1+10      % Ahora D1 apunta al Mapa de bits del Sprite

D0=806D5  % D0->ADISP
C=DAT0 A  % C = direccion del Stack Grob
D0=C      % D0->Stack Grob
D0+10
D0+10     % D0->Mapa de bits del Stack Grob
D0+ 2    % Incrementamos D0 para que el Sprite sea mostrado en la tercera columna

LA 09     % 10 filas menos uno, para el contador del bucle siguiente

```

```

C=0 W
A=0 W

*LoadBMap          % Leer los datos del Stack Grob, fila por fila, y escribirlos en nuevo grob.

LA FFC03           % FFC03h= 1111 1111 1100 0000 0011
C=DAT0 A
A=A&C A

C=0 A
C=DAT1 B           % B= 2 nibbles = 8 pixeles

C=C+C A
C=C+C A
C=C!A A

DAT0=C A
D0+34              % Siguiete fila del Stack Grob
D1+2               % Siguiete fila del nuevo grob.
                   % La cantidad de nibbles en una fila es siempre un entero par
B=B-1 A           % Decrementar contador
GONC LoadBMap

LOAD
D1+15
D+3 A

RPL
ENDCODE
@

```

Ejercicio: Modifica el código anterior para que inserte en la *Pantalla* un *Sprite* 8xN dado. El *Sprite* debe insertarse en el **cuarto pixel del nibble x**. Utilice FF807h= 1111 1111 1000 0000 0111

Find this book on
[AMAZON.COM](https://www.amazon.com)