



Comentario técnico: CTC-048

Componente: **Displays LCD gráficos 320x240 con S1D13700**

Autor: Sergio R. Caprile, Senior Engineer

Revisiones	Fecha	Comentarios
0	06/06/07	

Los displays LCD gráficos de 320x240 que comercializa Cika Electrónica han utilizado por mucho tiempo el controlador S1D13305 (SED1335). Actualmente, dicho chip ha sido reemplazado por el S1D13700, el cual es levemente diferente y soporta algunas características nuevas. Entre las diferencias que afectan el funcionamiento en sistemas ya desarrollados en torno al S1D13305, tenemos las siguientes:

- El S1D13700 ejecuta cada comando a medida que recibe los parámetros
- El S1D13700 requiere de un cierto tiempo para encender el oscilador a cristal, lo cual sucede luego de recibido el comando de inicialización SYSTEM SET. La inicialización se realiza entonces enviando el comando (o el comando y el primer parámetro) y esperando un cierto tiempo antes de seguir con la inicialización.
- El reset del S1D13700 requiere de un mínimo de duración del pulso de 1ms, el fabricante del display puede especificar un tiempo mayor.

En lo que respecta a diferencias entre los displays, los PG320240xxx-xE9 están basados en un clock de 12MHz, internamente dividido por cuatro. Esto ocasiona que **el tiempo mínimo de duración de la señal de \overline{WR} deba ser de 1,7us (5T)**.

Ejemplos prácticos

Para portar una aplicación como la desarrollada en CAN-012, con un display PG320240xxx-xE9 conectado al bus, deberemos modificar la rutina de inicialización *LCD_init()* para dar tiempo a la estabilización del oscilador:

```
WrtPortI(PECR, &PECRShadow, (PECRShadow & ~0xFF));
MsDelay ( 1000 ); // espera reset del LCD

LCD_WriteStrCmd ( init_string1,2 );
MsDelay ( 100 ); // da tiempo al osc a arrancar
LCD_WriteStrCmd ( init_string1,sizeof(init_string1) );
```

Además, deberemos extender el tiempo de acceso. Debido a la velocidad del Rabbit 2000, esto requiere habilitar el divisor por ocho y ajustar la cantidad de ciclos de wait. Para el último caso, modificamos la rutina de inicialización *LCD_init()* otra vez:

```
#define LCDSTROBE      0x80
#define LCDCSRREGISTER IB7CR
#define LCDCSSHADOW   IB7CRShadow
#define LCDCSCONFIG   0x48
```

para el primer caso, deberemos reescribir las rutinas de escritura a fin de habilitar e inhabilitar el divisor en ese momento, y no demorar demasiado las demás operaciones:

```
#asm
;@sp+2= 1st param, dirección
;@sp+4= 2nd param, dato
;
LCD_Write::
    ld hl,(sp+4)          ; dato (LSB)
    ipset 3              ; no ints
    ld a, 0x05           ; /8
    ioi ld (GCSR), a    ; set
```

CTC-048, Displays LCD gráficos 320x240 con S1D13700

```
ld a,1
ld hl,(sp+2)          ; address (LSB)
ioe ld (HL),a        ; escribe
ld a,0x09             ; /1
ioi ld (GCSR),a      ; set
ipres
ret

;
LCD_ReadData::
ld hl,(sp+2)         ; dato (LSB)
ipset 3              ; no ints
ld a,0x05            ; /8
ioi ld (GCSR),a     ; set
ld a,1
ioe ld a,(0xE001)    ;
ld l,a
ld a,0x09            ; /1
ioi ld (GCSR),a     ; set
ipres
ld h,0
ret

#endasm
```

El archivo adjunto incluye el código de CAN-012 portado y corregido (necesitará el resto de los archivos de CAN-012 para compilarlo), y una nueva versión de Cika320240FRST.lib con una versión modificada de la library de Rabbit para soporte de S1D13305 y S1D13700. La selección del controlador se realiza incluyendo (o no) la macro:

```
#define S1D13700

antes de incluir la library:

#define TSCONTROLLER 2
#define S1D13700
#use "Cika320240FRST.lib"
```

Para mayor información sobre Cika320240FRST.lib consulte CAN-016 y CAN-021