Driver para TFT1.8SP.

Generado por Doxygen 1.8.11

1. Documentación de archivos

1.1. Referencia del Archivo TftDriver.h

Módulo encargado de gestionar el TFT1.8 del shield TFT1.8SP de elecfreaks (ref. EF02005). El shield está basado en un display con el controlador ST7735S.

```
#include <xc.h>
#include <stdint.h>
```

'defines'

- #define **LANDSCAPE** 1
- #define **PORTRAIT** 0
- #define **LEFT** 0
- #define **RIGHT** 9999
- #define **CENTER** 9998
- #define **DISP_Y_SIZE** 159
- #define **DISP_X_SIZE** 127
- #define **VGA_BLACK** 0x0000
- #define **VGA_WHITE** 0xFFFF
- #define **VGA_RED** 0xF800
- #define **VGA_GREEN** 0x0400
- #define **VGA_BLUE** 0x001F
- #define **VGA_SILVER** 0xC618
- #define **VGA_GRAY** 0x8410
- #define **VGA_MAROON** 0x8000
- #define **VGA_YELLOW** 0xFFE0
- #define **VGA_OLIVE** 0x8400
- #define **VGA_LIME** 0x07E0
- #define **VGA_AQUA** 0x07FF
- #define **VGA_TEAL** 0x0410
- #define **VGA_NAVY** 0x0010
- #define **VGA_FUCHSIA** 0xF81F
- #define **VGA_PURPLE** 0x8010
- #define **VGA_TRANSPARENT** 0xFFFFFFFF

Funciones

- void inicializarTFT (int orientacion)
- void clrScr (void)
- void setColorRGB (uint8_t r, uint8_t g, uint8_t b)
- void setColor (uint16_t color)
- void setBackColorRGB (uint8_t r, uint8_t g, uint8_t b)
- void setBackColor (uint32_t color)
- void drawRect (int x1, int y1, int x2, int y2)
- void drawRoundRect (int x1, int y1, int x2, int y2)
- void drawLine (int x1, int y1, int x2, int y2)
- void drawPixel (int x, int y)
- void setFont (uint8_t *font)
- uint8_t * getFont (void)

- uint8_t getFontXsize (void)
- uint8_t getFontYsize (void)
- void print (char *st, int x, int y, int deg)
- void fillRect (int x1, int y1, int x2, int y2)
- void fillRoundRect (int x1, int y1, int x2, int y2)
- void drawCircle (int x, int y, int radius)
- void fillCircle (int x, int y, int radius)
- void fillScrRGB (uint8_t r, uint8_t g, uint8_t b)
- void fillScr (uint16_t color)
- void drawBitmap (int x, int y, int sx, int sy, uint16_t data[ ], int scale)

1.1.1. Descripción detallada

Módulo encargado de gestionar el TFT1.8 del shield TFT1.8SP de elecfreaks (ref. EF02005). El shield está basado en un display con el controlador ST7735S.

Autor

José Daniel Muñoz Frías (daniel)

Versión

1.0.0. Módulo original

Fecha

17/11/2016

```
Este driver está basado en el UTFT driver, aunque se ha portado a C
y se han eliminado el soporte para el resto de TFTs que incluía el
driver original.
```

1.1.2. Documentación de las funciones

1.1.2.1. void clrScr (void)

Borra la pantalla poniendola toda a color negro.

1.1.2.2. void drawBitmap (int *x*, int *y*, int *sx*, int *sy*, uint16_t *data[ ]*, int *scale*)

Dibuja una imagen (bitmap) en la pantalla. El bitmap ha de estar definido como un vector con los colores de todos los píxeles en formato RGB565. La forma más fácil de generarlo es mediante una aplicación web disponible en la página del autor del driver original: http://www.rinkydinkelectronics.com/t_↩imageconverter565.php

Parámetros

x	Coordenada X de la esquina superior derecha del bitmap.
y	Coordenada Y de la esquina superior derecha del bitmap.
sx	Tamaño horizontal del bitmap en pixels.
sy	Tamaño vertical del bitmap en pixels.
data	Dirección del vector que contiene el bitmap.
scale	Factor de escala para dibujar el bitmap.

1.1.2.3. void drawCircle (int *x*, int *y*, int *radius*)

Dibuja un círculo definido por su centro y su radio.

Parámetros

x	Coordenada X del centro.
y	Coordenada Y del dentro.
radius	Radio del círculo.

1.1.2.4. void drawLine (int *x1*, int *y1*, int *x2*, int *y2*)

Dibuja una línea entre los puntos (x1,y1) y (x2,y2)

Parámetros

x1	Coordenada X del punto 1.
y1	Coordenada Y del punto 1.
x2	Coordenada X del punto 2.
y2	Coordenada Y del punto 2.

1.1.2.5. void drawPixel (int *x*, int *y*)

Dibuja un pixel en las coordenadas (x,y).

Parámetros

x	Coordenada X del pixel.
y	Coordenada Y del pixel.

1.1.2.6. void drawRect (int *x1,* int *y1,* int *x2,* int *y2*)

Dibuja un rectangulo definido por sus dos esquinas.

Parámetros

x1	Coordenada X de la primera esquina.
y1	Coordenada Y de la primera esquina.
x2	Coordenada X de la segunda esquina.
y2	Coordenada Y de la segunda esquina.

1.1.2.7. void drawRoundRect (int *x1,* int *y1,* int *x2,* int *y2*)

Dibuja un rectangulo con los bordes redondeados definido por sus dos esquinas.

Parámetros

x1	Coordenada X de la primera esquina.
y1	Coordenada Y de la primera esquina.
x2	Coordenada X de la segunda esquina.
y2	Coordenada Y de la segunda esquina.

1.1.2.8. void fillCircle (int *x,* int *y,* int *radius*)

Dibuja un círculo relleno definido por su centro y su radio.

Parámetros

x	Coordenada X del centro.
y	Coordenada Y del dentro.
radius	Radio del círculo.

1.1.2.9. void fillRect (int *x1,* int *y1,* int *x2,* int *y2*)

Dibuja un rectangulo relleno definido por sus dos esquinas.

Parámetros

x1	Coordenada X de la primera esquina.
y1	Coordenada Y de la primera esquina.
x2	Coordenada X de la segunda esquina.
y2	Coordenada Y de la segunda esquina.

1.1.2.10. void fillRoundRect (int *x1,* int *y1,* int *x2,* int *y2*)

Dibuja un rectangulo relleno con los bordes redondeados definido por sus dos esquinas.

Parámetros

x1	Coordenada X de la primera esquina.
y1	Coordenada Y de la primera esquina.
x2	Coordenada X de la segunda esquina.
y2	Coordenada Y de la segunda esquina.

1.1.2.11. void fillScr (uint16_t *color*)

Rellena la pantalla con un color definido mediante un valor de 16 bits con el formato RGB565

Parámetros

color	Color en formato RGB565 (5 bits rojo, 6 bits verde, 5 bits azul).

1.1.2.12. void fillScrRGB (uint8_t *r,* uint8_t *g,* uint8_t *b*)

Rellena la pantalla con un color definido mediante sus componentes RGB.

Parámetros

r	Porcentaje de color rojo (0-255).
g	Porcentaje de color verde (0-255).
b	Porcentaje de color azul (0-255).

1.1.2.13. uint8_t* getFont (void)

Obtiene la dirección del vector de la fuente usada.

Devuelve

Dirección de la fuente usada.

1.1.2.14. uint8_t getFontXsize (void)

Retorna el ancho (en pixels) de la fuente en uso.

Devuelve

Ancho en pixels de la fuente en uso.

1.1.2.15. uint8_t getFontYsize (void)

Retorna el alto (en pixels) de la fuente en uso.

Devuelve

Alto en pixels de la fuente en uso.

1.1.2.16. void inicializarTFT (int *orientacion*)

Inicializa el display TFT. Seleccciona negro para el fondo y blanco para escribir.

Parámetros

orientacion	Orientación del display: LANDSCAPE o PORTRAIT

1.1.2.17. void print (char * *st*, int *x*, int *y*, int *deg*)

Imprime una cadena de caracteres a partir de las coordenadas (x,y) con un ángulo determinado. Las coordenadas definen la esquina superior izquierda del primer carácter de la cadena.

Parámetros

st	Cadena de caracteres.
x	Coordenada X de la esquena superior izquierda del primer carácter.
y	Coordenada Y de la esquena superior izquierda del primer carácter.
deg	Ángulo con el que se imprime la cadena (0 horizontal, 90 vertical, etc.)

1.1.2.18. void setBackColor (uint32_t *color*)

Selecciona el color de fondo usado para imprmir los caracteres. El argumento es el color en el formato RGB565. Se pueden usar constantes predefinidas con los colores usados por las tarjetas VGA: VGA_WHITE, VGA_RED, etc. (ver TftDriver.h). Además se puede seleccionar el color VGA_TRANSPARENT para que el fondo de los caracteres sea transparente.

Parámetros

color	Color en formato RGB565 (5 bits rojo, 6 bits verde, 5 bits azul). Usar 0xFFFFFFFF (VGA_TRANSPARENT) para que el fondo de los caracteres sea transparente.

1.1.2.19. void setBackColorRGB (uint8_t *r*, uint8_t *g*, uint8_t *b*)

Selecciona el color de fondo del display.

Parámetros

r	Porcentaje de color rojo (0-255). Se usan los 5 bits más significativos.
g	Porcentaje de color verde (0-255). Se usan los 6 bits más significativos.
b	Porcentaje de color azul (0-255). Se usan los 5 bits más significativos.

1.1.2.20. void setColor (uint16_t *color*)

Selecciona el color para escribir. Todas las funciones que dibujan en el display (draw, fill y print) usan este color. El color se define mediante un uint16_t con el formato RGB565. Se pueden usar constantes predefinidas con los colores usados por las tarjetas VGA: VGA_WHITE, VGA_RED, etc. (ver TftDriver.h).

Parámetros

color	Color en formato RGB565 (5 bits rojo, 6 bits verde, 5 bits azul).

1.1.2.21. void setColorRGB (uint8_t *r,* uint8_t *g,* uint8_t *b*)

Selecciona el color para escribir. Todas las funciones que dibujan en el display (drawXX, fillXX y print) usan este color.

Parámetros

r	Porcentaje de color rojo (0-255). Se usan los 5 bits más significativos.
g	Porcentaje de color verde (0-255). Se usan los 6 bits más significativos.
b	Porcentaje de color azul (0-255). Se usan los 5 bits más significativos.

1.1.2.22. void setFont (uint8_t * *font*)

Selecciona la fuente a usar por las funciones de texto. Por ejemplo, para seleccionar la fuente SmallFont definida en DefaultFonts.c basta con hacer:

```
1 extern uint8_t SmallFont[];
2 ...
3 SetFont(SmallFont[];
```

Parámetros

font	Nombre del vector que contiene la definición de la fuente.