

# LIBRERÍA TFT

## 1. Inicialización y Configuración

### a. `__init__(rst_pin=24, dc_pin=25, cs_pin=8, spi_bus=0, spi_device=0)`

#### Parámetros:

- `rst_pin`: Pin GPIO para RESET (por defecto 24)
- `dc_pin`: Pin GPIO para Data/Command (por defecto 25)
- `cs_pin`: Pin GPIO para Chip Select (por defecto 8)
- `spi_bus`: Bus SPI (por defecto 0)
- `spi_device`: Dispositivo SPI (por defecto 0)

**Utilidad:** Constructor de la clase, configura pines y SPI.

### b. `init(rotation=1)`

#### Parámetros:

- `rotation`: Rotación de pantalla (0=0°, 1=90°, 2=180°, 3=270°)

**Utilidad:** Inicializa el display con la rotación especificada.

### c. `set_rotation(rotation)`

#### Parámetros:

- `rotation`: 0=0°, 1=90°, 2=180°, 3=270°

**Utilidad:** Cambia la orientación de la pantalla.

## 2. Gestión de Colores

### a. `get_color(color_name)`

#### Parámetros:

- `color_name`: Nombre del color ('RED', 'GREEN', 'BLUE', etc.)

**Utilidad:** Devuelve el valor RGB565 del color por nombre.

#### Colores disponibles:

- 'BLACK', 'WHITE', 'RED', 'GREEN', 'BLUE'
- 'YELLOW', 'MAGENTA', 'CYAN', 'ORANGE', 'GRAY'

## 3. Funciones de Dibujo Básicas

### a. `clear(color=0x0000)`

#### Parámetros:

- `color`: Color de fondo (por defecto negro)

**Utilidad:** Limpia toda la pantalla con un color.

### b. `draw_pixel(x, y, color)`

#### Parámetros:

- `x, y`: Coordenadas del pixel
- `color`: Color del pixel

**Utilidad:** Dibuja un pixel individual.

**c. fill\_rect(x, y, width, height, color)**

**Parámetros:**

- x, y: Coordenada superior izquierda
- width, height: Ancho y alto del rectángulo
- color: Color de relleno

**Utilidad:** Dibuja un rectángulo relleno.

**d. draw\_rect(x, y, width, height, color)**

**Parámetros:**

- x, y: Coordenada superior izquierda
- width, height: Ancho y alto del rectángulo
- color: Color del borde

**Utilidad:** Dibuja un rectángulo (solo bordes).

**e. draw\_line(x0, y0, x1, y1, color)**

**Parámetros:**

- x0, y0: Coordenada inicial
- x1, y1: Coordenada final
- color: Color de la línea

**Utilidad:** Dibuja una línea entre dos puntos.

## **4. Funciones de Texto**

**a. draw\_char(x, y, char, color, bg\_color)**

**Parámetros:**

- x, y: Coordenadas del carácter
- char: Carácter a dibujar
- color: Color del texto
- bg\_color: Color de fondo

**Utilidad:** Dibuja un carácter individual (8x8 pixels).

### **b. draw\_text(x, y, text, color, bg\_color)**

#### **Parámetros:**

- x, y: Coordenadas iniciales del texto
- text: Texto a mostrar
- color: Color del texto
- bg\_color: Color de fondo

**Utilidad:** Dibuja una cadena de texto.

#### **Caracteres soportados:**

- Letras mayúsculas (A-Z)
- Letras minúsculas (a-z)
- Números (0-9)
- Símbolos (!, ?, ., ,, :, -, \_, =, +, \*, /, (, ), etc.)

## **5. Utilidades y Limpieza**

### **a. cleanup()**

**Parámetros:** Ninguno

**Utilidad:** Libera recursos GPIO y cierra SPI.

### **b. create\_tft(rst\_pin=24, dc\_pin=25, cs\_pin=8, rotation=1)**

**Parámetros:** Mismos que el constructor

**Utilidad:** Crea e inicializa el display en un solo paso.

## **6. Propiedades Útiles**

tft.WIDTH # Ancho actual de la pantalla (depende de rotación)

tft.HEIGHT # Alto actual de la pantalla (depende de rotación)

tft.colors # Diccionario de colores predefinidos

## 7. Ejemplos Prácticos

*# Uso básico*

```
from TFT_ILI9341 import create_tft  
tft = create_tft(rotation=1) # Apaisado
```

*# Limpiar pantalla*

```
tft.clear(tft.get_color('BLUE'))
```

*# Dibujar texto*

```
tft.draw_text(10, 50, "Hola Mundo", tft.get_color('WHITE'), tft.get_color('BLUE'))
```

*# Dibujar formas*

```
tft.draw_rect(50, 100, 100, 50, tft.get_color('RED'))  
tft.fill_rect(60, 110, 80, 30, tft.get_color('GREEN'))
```

*# Limpiar recursos al final*

```
tft.cleanup()
```