

---

## Instrucciones para incluir la librería TFT\_Ophyra en el proyecto.

### 1. En el proyecto:

Incluir en la carpeta "Inc" los archivos:

`TFT_Ophyra.h ; ST7735_Driver.h ; Intesc_Std_Types.h`

Incluir en la carpeta "Src" los archivos:

`TFT_Ophyra.c ; libST7735_Driver.a`

**Nota:** Dentro de las carpetas *Inc* y *Src* que se proporcionan junto con este documento, se encuentran los archivos antes mencionados.

### 2. Dentro de "las propiedades (*properties*)" del proyecto, agregar lo siguiente:

C/C++Build -> Settings -> (pestaña)Tool Settings -> Libraries  
-> (ventana) Libraries ->

`:libST7735_Driver.a`

C/C++Build -> Settings -> (pestaña)Tool Settings -> Libraries  
-> (ventana) Library search path ->

`"${workspace_loc}/${ProjName}/Src"`

### 3. Copiar y pegar las líneas que a continuación se muestran, dentro de la función principal "main.c" del proyecto.

```
#include "Intesc_Std_Types.h"
```

```
#include "ST7735_Driver.h"
```

```
#include "TFT_Ophyra.h"
```

**Nota:** Estas líneas y las instrucciones anteriores, también se encuentran repetidas dentro el archivo `TFT_Ophyra.h`

### 4. Si se desplegarán imágenes, es necesario activar y configurar el puerto SD en el *STM32CubeMX*, para leer la imagen desde una memoria. La librería puede leer imágenes BMP de 8 bits y 24bits. Después de activar la SD, en el archivo "TFT\_Ophyra.h" descomentar la línea 8:

```
// #define __TFT_BMP
```

**Nota:** Si no se utiliza la memoria mantener comentada esta línea para evitar errores de compilación.

---

5. A continuación se explica las funciones disponibles de la librería.

- **Función:** `TFT_Init(Arg);`  
**Descripción:** Inicializa la pantalla TFT. Ejecute primero esta instrucción antes de usar la pantalla.  
**Argumento:**  
Si `Arg=0` la pantalla inicia en horizontal.  
Si `Arg=1` la pantalla inicia en vertical.
- **Función:** `TFT_BackLight(Arg);`  
**Descripción:** Enciende/Apaga la iluminación de la pantalla TFT.  
**Argumento:**  
Si `Arg=0` la pantalla se apaga.  
Si `Arg=1` la pantalla se enciende.
- **Función:** `TFT_ClearDisplay(Arg);`  
**Descripción:** Limpia la pantalla TFT al color que se le indique.  
**Argumento:**  
`Arg` debe ser un número de 32 bits en formato RGB.  
Ejemplo: `0xFFFFFFFF` (color blanco); `0x000000` (color negro); `0xFF0000` (color rojo); `0x00FF00` (color verde); etc.
- **Función:** `TFT_PrintStr(X, Y, Arg, Color1, Color2);`  
**Descripción:** Escribe una cadena de caracteres en la pantalla. La cadena debe de estar dentro de comillas, ejemplo: "HOLA MUNDO". También se pueden desplegar cadenas contenidas dentro de arreglos de caracteres.  
**Argumentos:**  
`X, Y` indican la coordenadas donde se comenzarán a escribir los caracteres.  
`Arg` es la cadena de caracteres a escribir en la pantalla.  
`Color1` es el color de los caracteres, en formato numérico de 32bits RGB.  
`Color2` es el color de fondo de cada carácter, en formato numérico de 32bits RGB. Si en lugar del número se escribe la palabra *Transparent*, el fondo de las letras será transparente.

**Nota:** Los caracteres tienen un tamaño de 5x8 pixeles.

- **Función:** `TFT_PrintLine(X1, Y1, X2, Y2, Color);`  
**Descripción:** Dibuja una línea recta en la pantalla.  
**Argumentos:**  
    *X1, Y1* indican las coordenadas de inicio de la línea.  
    *X2, Y2* indican las coordenadas finales de la línea.  
    *Color* es el color de la línea, en formato numérico de 32bits RGB.  
**Nota:** *X1* y *X2* deben ser valores numéricos enteros entre 0 y 127. *Y1* e *Y2* deben ser valores entre 0 y 159. Exceder estos rangos puede provocar un comportamiento inestable de la pantalla.
- **Función:** `TFT_PrintImgBMP(X1, Y1, Arg);`  
**Descripción:** Despliega una imagen en formato BMP de 8 o 24 bits. La imagen es leída desde la memoria uSD.  
**Argumentos:**  
    *X1, Y1* indican las coordenadas de inicio de la imagen.  
    *Arg* es el nombre de la imagen a leer entre comillas, ejemplo: *"Logo.bmp"*.  
**Nota:** Se debe especificar la dirección exacta de la imagen dentro de la memoria uSD. Preferentemente las imágenes deberían estar en la raíz de la memoria. La imagen no debe exceder las dimensiones de 128x160 píxeles. Esta función despliega errores en la parte superior de la pantalla dependiendo del caso.  
**Errores:**  
    *Error: only 8 or 24 bits is supported*  
    *Error: Image too large*  
    *Error: Image not is bmp*  
    *Error: Image no found*  
    *Error: No SD mount*