

# Sesión de Laboratorio 7: Programación de pantalla TFT con M5Core2

---

**Tiempo estimado:** 1,5 h (1 sesión)

## 1. Trabajando con hardware real

### Parte 1: Registrar el píxel que se ha pulsado

- El programa debe comprobar si se ha tocado la pantalla o no.
- Si se ha tocado, obtener la posición X, Y del píxel que se ha tocado. Para ello será necesario utilizar las siguientes funciones:
  - `bool screen_pressed = M5.Touch.ispressed();` Devuelve `true` si se ha tocado la pantalla y `false` en caso contrario.
  - `TouchPoint_t pixel_pos = M5.Touch.getPressPoint();` devuelve la posición del píxel que se ha tocado.
- La posición del píxel tiene 2 coordenadas (X,Y), para almacenar ambas coordenadas, será necesario utilizar el tipo de variable `TouchPoint_t`. Las coordenadas se almacenan de la siguiente manera:

```
TouchPoint_t coordinate;  
int16_t coordinate.x  
int16_t coordinate.y
```

- El siguiente código de ejemplo escribe por el puerto serie los píxeles que se han tocado:

```
#include <M5Core2.h>  
  
// Variable de tipo TouchPoint_t para almacenar el píxeles tocado en  
// la pantalla  
TouchPoint_t pixel_pos;  
  
void setup()  
{  
  // Inicializar puerto serie  
  Serial.begin(115200);  
  
  M5.begin();  
  
  M5.Lcd.setTextColor(YELLOW, BLACK); // Poner el color de texto en  
  amarillo  
  M5.Lcd.setTextSize(2);              // Poner el tamaño de las letras  
  en 2  
  M5.Lcd.println("Registrar pixel");
```

```
// Rellenar el fondo de la pantalla en negro
M5.Lcd.fillScreen(BLACK);
}

void loop()
{
    M5.update();

    // Si se ha tocado la pantalla
    if (M5.Touch.ispressed())
    {
        pixel_pos = M5.Touch.getPressPoint(); // Guardo el pixel que
se ha tocado
        Serial.printf("x:%d, y:%d \r\n", pixel_pos.x, pixel_pos.y); //
Escribo por el puerto serie la posición del pixel presionado
    }
}
```

- Al probar el código anterior veréis que se repite el . Para arreglar este problema se pide modificar el código anterior para almacenar la posición del píxel que se está tocando actualmente y la del último que se tocó. En caso de ser la misma, no

## Parte 2: Hacer un script para M5Core2 que permita dibujar en la pantalla con el dedo

A partir del código de la parte 1, una vez se haya comprobado que la pantalla se ha presionado, que el píxel que se ha presionado sea distinto al presionado en el paso anterior del loop, y que se haya registrado en una variable de tipo `TouchPoint_t` se pide:

- En esa misma posición, dibujar utilizando alguna de las siguientes funciones:
  - `M5.Lcd.drawPixel(int16_t x0, int16_t y0, uint16_t color);` Rellena el píxel de la coordenada (x0, y0) en un color (p.ej., WHITE).
  - `M5.Lcd.fillCircle(int16_t x0, int16_t y0, int16_t r, uint16_t color);` Dibuja un círculo relleno con centro en el punto X0 and Y0, con radio r y un color (p.ej., WHITE).

## Parte 3: Añadir la opción de borrar el contenido de la pantalla al pulsar el botón A

Para ello será necesario utilizar las funciones `M5.Lcd.fillScreen()` o `M5.Lcd.clear()`.

## Trabajando en simulación

Para realizar el ejercicio en simulación es necesario utilizar una pantalla TFT en wokwi. [Aquí](#) hay un ejemplo que resuelve la parte 1 de la práctica en wokwi utilizando la pantalla *ILI9341* y las librerías correspondientes de dicha pantalla.

En ese ejemplo se pueden ver las similitudes de la estructura del código y las diferencias en la sintaxis por utilizar otras librerías diferentes a `M5Core2.h`.

Modificar el ejemplo para resolver las partes 2 y 3 en simulación. Para la parte 3 será necesario añadir un botón extenro.

**NOTA:** Trabajar con esta pantalla es costoso desde el punto de vista computacional, por lo que wokwi no consigue un rendimiento de tiempo real cercano al 100%.