```
/*This code demonstrates the Touch function
by printing "Pressed!"  */

#include <lvgl.h>
#include <TFT_eSPI.h>
#include <XPT2046_Touchscreen.h>

// --- SETTINGS & BUFFER ---
static const uint16_t screenWidth  = 320;
static const uint16_t screenHeight = 240;
static lv_disp_draw_buf_t draw_buf;
static lv_color_t buf[screenWidth * 10];

// --- CYD PINOUT CONFIGURATION ---
#define XPT2046_IRQ  36
#define XPT2046_MOSI 32
#define XPT2046_MISO 39
#define XPT2046_CLK  25
#define XPT2046_CS   33
#define TFT_CS       15

// Isolated HSPI channel for Touch to prevent TFT high-speed bus corruption
SPIClass touchSPI(HSPI);
XPT2046_Touchscreen ts(XPT2046_CS, XPT2046_IRQ);
TFT_eSPI tft = TFT_eSPI();

// Display flushing helper for LVGL
void my_disp_flush(lv_disp_drv_t *disp, const lv_area_t *area, lv_color_t *color_p) {
```

```
    uint32_t w = (area->x2 - area->x1 + 1);
    uint32_t h = (area->y2 - area->y1 + 1);

    tft.startWrite();
    tft.setAddrWindow(area->x1, area->y1, w, h);
    tft.pushColors((uint16_t *)&color_p->full, w * h, true);
    tft.endWrite();

    lv_disp_flush_ready(disp);
}

// Safe Touch read driver for LVGL
void my_touchpad_read(lv_indev_drv_t * indev_driver, lv_indev_data_t * data) {
    digitalWrite(TFT_CS, HIGH);
    digitalWrite(XPT2046_CS, LOW);

    if (ts.touched()) {
        TS_Point p = ts.getPoint();

        // Ensure data registers true finger pressure
        if (p.x > 100 && p.y > 100) {
            data->state = LV_INDEV_STATE_PRESSED;

            // --- PERFECT HARDWARE CALIBRATION MAP ---
            // Uses your exact measured raw values: X (235 to 3650), Y (370 to 3785)
            data->point.x = map(p.x, 235, 3650, 0, screenWidth);
            data->point.y = map(p.y, 370, 3785, 0, screenHeight);
```

```
            // Constrain coordinates within screen pixels boundary
            data->point.x = constrain(data->point.x, 0, screenWidth - 1);
            data->point.y = constrain(data->point.y, 0, screenHeight - 1);

            Serial.printf("CALIBRATED -> Pixel X:%d Pixel Y:%d (Raw X:%d Y:%d)\n", data->point.x,
data->point.y, p.x, p.y);
        } else {
            data->state = LV_INDEV_STATE_RELEASED;
        }
    } else {
        data->state = LV_INDEV_STATE_RELEASED;
    }

    digitalWrite(XPT2046_CS, HIGH);
    digitalWrite(TFT_CS, LOW);
}

// --- BUTTON EVENT CALLBACK ---
static void btn_event_cb(lv_event_t * e) {
    lv_event_code_t code = lv_event_get_code(e);
    lv_obj_t * btn = lv_event_get_target(e);
    lv_obj_t * label = lv_obj_get_child(btn, 0);

    if(code == LV_EVENT_PRESSED) {
        lv_label_set_text(label, "Pressed!");
        Serial.println("--> LVGL Button Event: Pressed");
    }
    if(code == LV_EVENT_RELEASED) {
```

```
        lv_label_set_text(label, "Touch Test");
        Serial.println("--> LVGL Button Event: Released");
    }
}

void setup() {
    Serial.begin(115200);

    // 1. Hardware Pin Setup
    pinMode(21, OUTPUT);
    digitalWrite(21, HIGH); // Backlight ON

    pinMode(XPT2046_CS, OUTPUT);
    digitalWrite(XPT2046_CS, HIGH);
    pinMode(TFT_CS, OUTPUT);
    digitalWrite(TFT_CS, HIGH);

    // 2. Initialize Display
    tft.begin();
    tft.setRotation(1); // Landscape

    // 3. Launch Isolated Touch SPI Bus Channel
    touchSPI.begin(XPT2046_CLK, XPT2046_MISO, XPT2046_MOSI, XPT2046_CS);
    ts.begin(touchSPI);
    ts.setRotation(1);

    // 4. LVGL Engine Setup
    lv_init();
```

```
    lv_disp_draw_buf_init(&draw_buf, buf, NULL, screenWidth * 10);

    // 5. Display Driver Registration
    static lv_disp_drv_t disp_drv;
    lv_disp_drv_init(&disp_drv);
    disp_drv.hor_res = screenWidth;
    disp_drv.ver_res = screenHeight;
    disp_drv.flush_cb = my_disp_flush;
    disp_drv.draw_buf = &draw_buf;
    lv_disp_drv_register(&disp_drv);

    // 6. Input Touch Driver Registration
    static lv_indev_drv_t indev_drv;
    lv_indev_drv_init(&indev_drv);
    indev_drv.type = LV_INDEV_TYPE_POINTER;
    indev_drv.read_cb = my_touchpad_read;
    lv_indev_drv_register(&indev_drv);

    // 7. UI Creation
    lv_obj_t * btn = lv_btn_create(lv_scr_act());
    lv_obj_set_size(btn, 150, 50);
    lv_obj_center(btn);
    lv_obj_add_event_cb(btn, btn_event_cb, LV_EVENT_ALL, NULL);

    lv_obj_t * label = lv_label_create(btn);
    lv_label_set_text(label, "Touch Test");
    lv_obj_center(label);
}
```

```
void loop() {
    lv_timer_handler();
    delay(5);
}
```