#include <lvgl.h>

#include <TFT\_eSPI.h>

TFT\_eSPI tft = TFT\_eSPI(); /\* TFT instance \*/

void my\_disp\_flush(lv\_disp\_drv\_t \*disp, const lv\_area\_t \*area, lv\_color\_t \*color\_p) {

tft.startWrite();

tft.setAddrWindow(area->x1, area->y1, area->x2 - area->x1 + 1, area->y2 - area->y1 + 1);

tft.pushColors(&color\_p->full, (area->x2 - area->x1 + 1) \* (area->y2 - area->y1 + 1), true);

tft.endWrite();

lv\_disp\_flush\_ready(disp);

}

void setup() {

tft.begin(); /\* TFT init \*/

tft.setRotation(1); /\* Landscape orientation \*/

lv\_init();

static lv\_disp\_draw\_buf\_t draw\_buf;

static lv\_color\_t buf[LV\_HOR\_RES\_MAX \* 10];

lv\_disp\_draw\_buf\_init(&draw\_buf, buf, NULL, LV\_HOR\_RES\_MAX \* 10);

static lv\_disp\_drv\_t disp\_drv;

lv\_disp\_drv\_init(&disp\_drv);

disp\_drv.hor\_res = 320;

disp\_drv.ver\_res = 480;

disp\_drv.flush\_cb = my\_disp\_flush;

disp\_drv.draw\_buf = &draw\_buf;

lv\_disp\_drv\_register(&disp\_drv);

}

void loop() {

lv\_timer\_handler(); /\* let the GUI do its work \*/

delay(5);

}