# Integrating LVGL with the ILI9341 Driver over SPI

## 1. Overview of Integration Approach

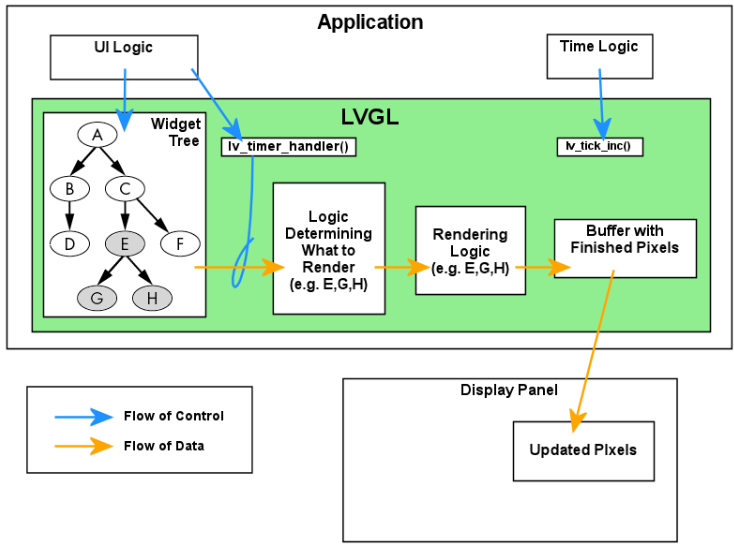
The graphical user interface of our system is rendered using LVGL v9.3, a powerful and lightweight embedded GUI library. To display LVGL's graphical output on our 240x320 TFT screen, we integrated the ILI9341 display driver over the SPI interface on the Raspberry Pi 3 using bare-metal C, without an operating system or abstraction layer.  
  
The integration leverages the driver wrapper provided in the official LVGL source for the ILI9341 controller. In our implementation, we manually configure SPI communication routines in C, then bind them to LVGL’s display driver through two critical functions:  
  
- my\_lcd\_send\_cmd(): Sends commands to the ILI9341 controller  
- my\_lcd\_send\_color(): Sends pixel color data from the LVGL buffer to the display  
  
These functions are passed to the lv\_ili9341\_create() function, which registers them with LVGL's internal display driver abstraction.  


Figure ‑: Overview of LVGL's Data Flow

## 2. Display Initialization

Before LVGL can render frames, the ILI9341 must be initialized. While the low-level initialization sequence is handled internally by lv\_ili9341\_create(), it requires that SPI communication functions are properly implemented and provided by the user.  
  
Example: SPI Command Transmission

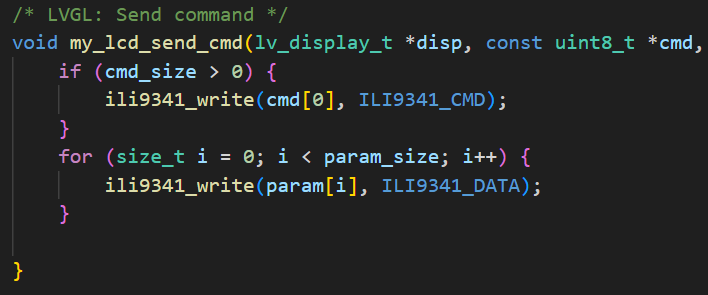


Figure ‑: Command sending logic

## 3. Transferring Pixel Data to ILI9341

LVGL periodically invokes a flush\_cb() callback to send a portion of its display buffer to the screen. The ILI9341 expects pixel data in RGB565 format, which is already compatible with LVGL’s default color depth.

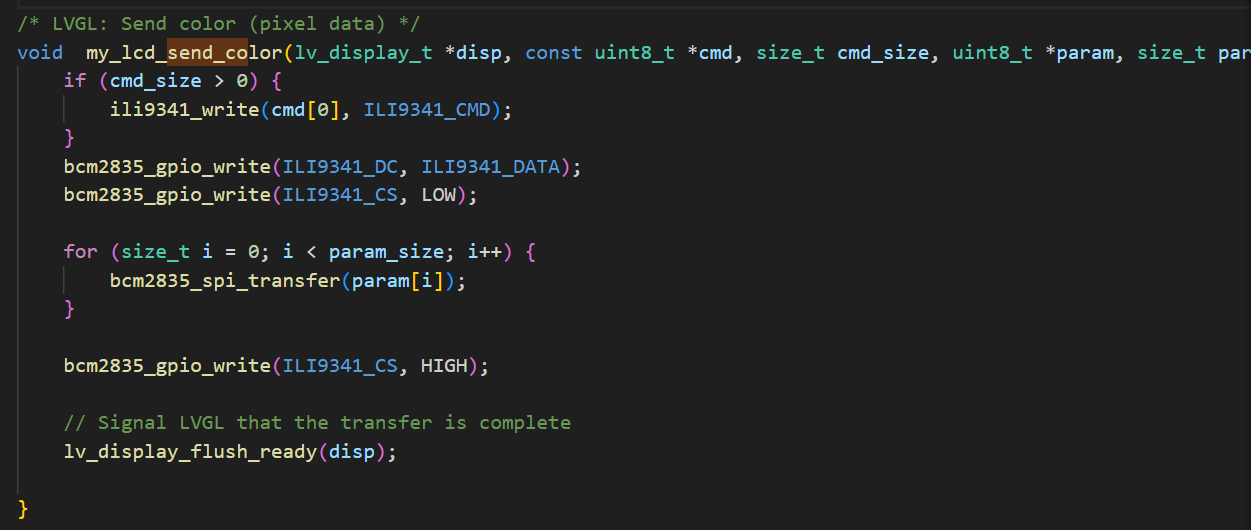


Figure ‑: Pixel data transfer

## 4. Registering the Display Driver

After defining the communication routines, we register them using the helper provided by LVGL:

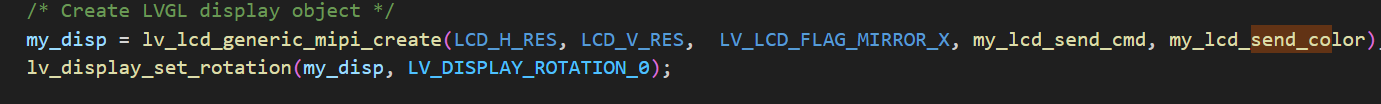


Figure ‑: Display registration using

## 5. Configuration in lv\_conf.h

The following parameters were customized in lv\_conf.h to match our hardware:

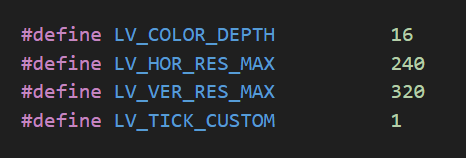


Figure ‑: Customized LVGL configuration

- LV\_COLOR\_DEPTH is set to 16 to use RGB565.  
- LV\_HOR\_RES\_MAX and LV\_VER\_RES\_MAX are configured per screen resolution.  
- LV\_TICK\_CUSTOM is enabled to allow hardware timer tick integration.

## References

[1] LVGL Official Documentation – Display Porting Guide  
https://docs.lvgl.io/latest/porting/display.html  
[2] LVGL ILI9341 Display Driver (source and usage):  
https://github.com/lvgl/lvgl