2.13. Graphic library LVGL

We are using lvgl for gui development (https://lvgl.io/). There are a lot of information, samples and demos for this library. We are using https://squareline.io/ as UI editor.

2.13.1. Single Demo

Uncompress lvgl_demo.tgz

```
mkdir lvgl_work
cd lvglwork
cp $HOME/Downloads/lvgl_demo.tgz .
tar zxvf lvgl_demo.tgz
```

This demo has the following distribution:

```
|-- littlevgl-8          lvgl library source
|-- demo_lvgl/
|   |-- fb_files         App framebuffer related source code
|   |-- liblvgl          Directory with one copy of liblvgl.a library
|   |-- main.c           App source code.
|   |-- Makefile         App Makefile
|   |-- SL_project       SquareLine Project
|   `-- ui               SqureLine generated files
```

2.13.1.1. Download toolchain

```
sudo apt-get install g++-aarch64-linux-gnu gcc-11-aarch64-linux-gnu \
gcc-11-aarch64-linux-gnu-base
```

2.13.1.2. Create lvgl library

```
cd littlevgl-8/lvgl/build
rm -rf *
cmake ../
make
When the compilation process finish you get:
[100%] Linking C static library lib/liblvgl.a
```

2.13.1.3. Compile demo application

```
Change to demo directory (lvgl_work/demo_lvgl/)
cd ../../../demo_lvgl/
make
```

2.13.1.4. Copy and run app executable to powerwatch

Copy the app executable to PowerWatch using ssh

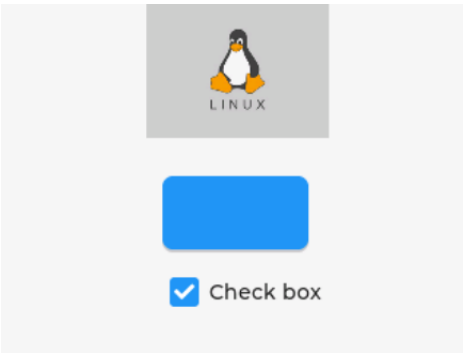

Figura 2.40: lvgl demo app

```
scp lvgl_demo root@xx.xx.xx.xx:
```

On PowerWatch console run:

```
./lvgl_demo
```

You must see the following image, you can interact with app's button and checkbox.

2.13.2. Demo internals

Demo project was created using SquareLine with the following settings
This project was stored on

```
|--lvgl_work
|   |-- demo_lvgl/
|   |   |-- SL_project        SquareLine Project
|   |   |   |-- assets        Folder to store PNG images
|   |   |   |-- autosave
|   |   |   |-- backup
|   |   |   |-- cache
|   |   |   |-- components
|   |   |   |-- single_ui_events.py
|   |   |   |-- single_ui.sll
|   |   |   |-- single_ui.spj
|   |   |   `-- Themes.slt
```

All images we want to use must be stored on asset directory. We copy two **PNG** files *Tux.png* and *Tux2.png*, both 320 x 240, 8-bit/color RGB.

2.13.2.1. Design Graphical Interface with LVGL's Widgets

Lvgl have a widget library with many common controls, we can use these widgets to create the UI:
SquareLine can generate the C code to generate the GUI, this code is stored on any folder, in our case on: **lvgl_work/demo_lvgl/ui/** (this folder is defined as part of project settings)

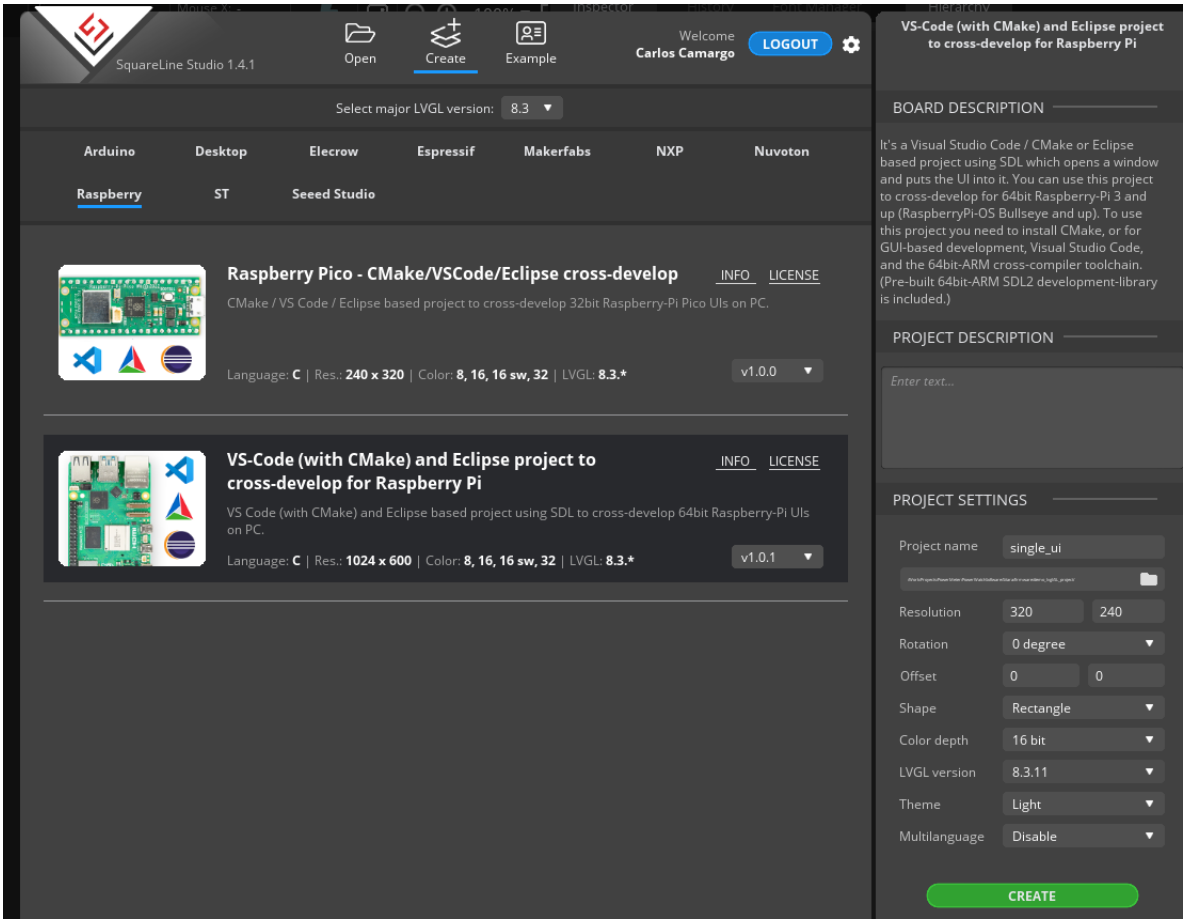

Figura 2.41: SquareLine project settings

2.13.2.2. Events

We can add events to interact with our application, lvgl offer a wide variety of such events based on touchscreen or mouse actions. For this demo, we've added events to *button* and *checkbox*.

As shown in sl_events, we can add events when the button is pressed or the check button change its state. SquareLine generate tree empty functions: *change_image unchecked* and *checked*. We must fill them with the required behavior; in this case, we change the image.

2.13.3. Addapting SL's generated code to our app

demo_lvgl folder can be used as template for development, it contains the default files for basic configuration. Figure 2.44 shows *main.c* and *Makefile* files. Variable **SRCS** contains source code to be compiled, here you can add your code; variable *VPATH* declare folders that contain source code, you can add your folder here.

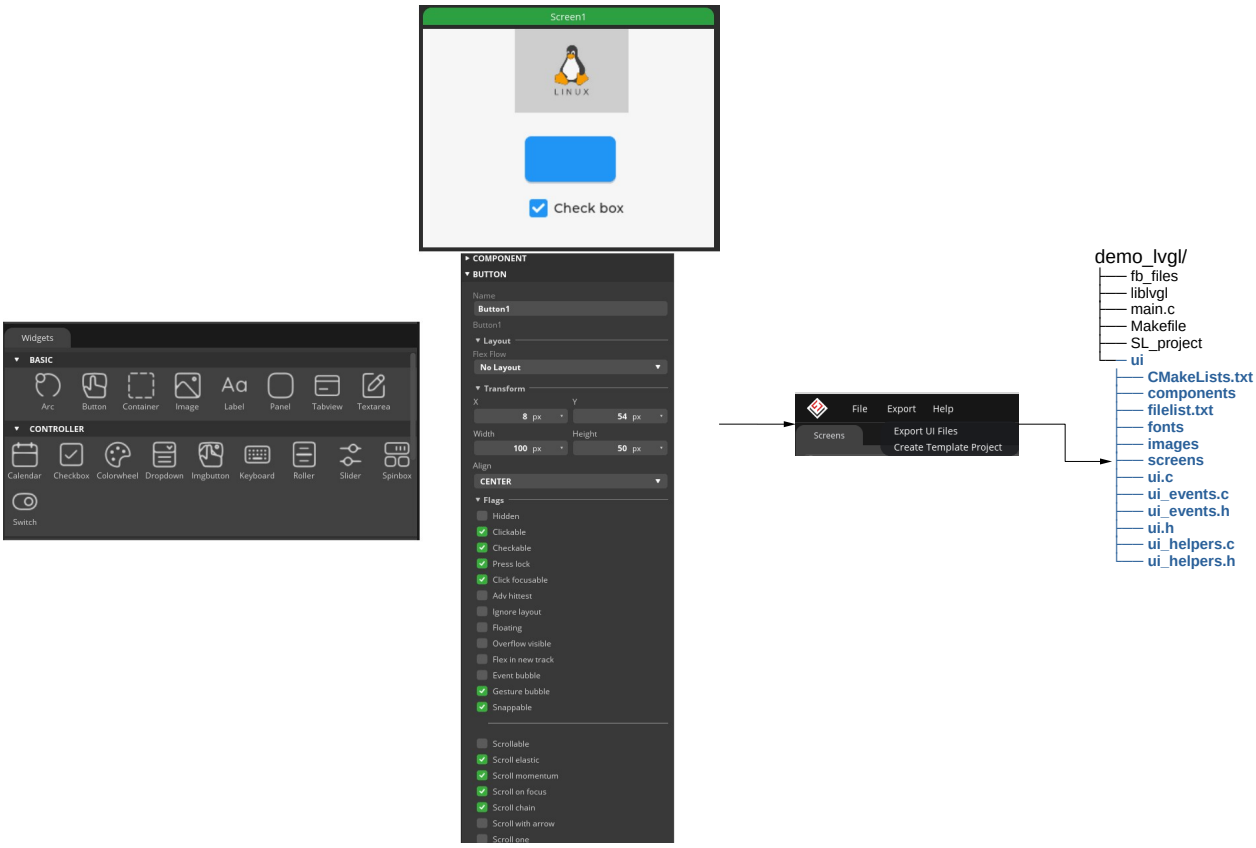

Figura 2.42: SquareLine project widgets

Figura 2.43: SquareLine Events

```
TARGET   = lvgl_demo
CROSS    = aarch64-linux-gnu
CC       = $(CROSS)-gcc-11
AS       = $(CROSS)-as-11
LD       = $(CROSS)-ld-11
LIB_DIR  = -Llib\vgl
LIBS     = -llvgl -lm
LVGL_DIR = ../littlevgl-8
CFLAGS   = -I$(LVGL_DIR)/lvgl -I$(LVGL_DIR)/lv_drivers -I$(LVGL_DIR)/ -Iui/ -I.
SRCS     :=  evdev.c  ui.c  ui_helpers.c ui_Screen1.c mouse_cursor_icon.c
             ui_img_tux_png.c ui_img_tux2_png.c ui_events.c  fbdev.c  main.c
OBJECTS := $(SRCS:.c=.o)
VPATH = ./ui:./ui/screens/:./fb_files/:./ui/images
all: $(TARGET)
lvgl_demo: $(OBJECTS)
	$(CC) $(LDFLAGS) $(OBJECTS) -o $@  ${LIB_DIR}  ${LIBS}
%.o: %.c
	$(CC) $(CFLAGS) -c $<
%.o: %.S
	$(AS) $(AFLAGS) -c $< -o $@
clean:
	rm -rf  $(OBJECTS) *.elf *.bin .*~ *~ *.map *.d *.hex *.lst $(TARGET)
.PHONY: clean
```

Makefile

```
int main(int argc, char **argv)
{
  (void)argc; /*Unused*/
  (void)argv; /*Unused*/

  /*Initialize LVGL*/
  lv_init();
  /*Initialize the HAL (display, input devices, tick) for LVGL*/
  hal_init();
  //lv_example_checkbox_1();
  ui_init();
  while(1) {
      lv_timer_handler();
      usleep(5000);

  }
  return 0;
}
```

main.c

```
void ui_init(void) {
    lv_disp_t * dispp = lv_disp_get_default();
    lv_theme_t * theme = lv_theme_default_init(dispp, lv_palette_main(LV_PALETTE_BLUE), lv_palette_main(LV_PALETTE_RED),
                                               false, LV_FONT_DEFAULT);
    lv_disp_set_theme(dispp, theme);
    ui_Screen1_screen_init();
    ui____initial_actions0 = lv_obj_create(NULL);
    lv_disp_load_scr(ui_Screen1);
}
```

ui.c

Figura 2.44: SquareLine Events

Referencias