

Setting up environment

Setting up the environment for SquareLine Studio

- Installing SquareLine Studio
- Adding supported libraries

Initiating a Project with SquareLine Studio

- Create a project
- Pre-requisites.
- Compile and Upload.



Setting up environment

For detailed instructions on,

Installing the Arduino IDE and adding the board, please refer to the [Getting Started with NORVI ESP32 HMI](#) document. Getting Started with NORVI ESP32 HMI provides step-by-step guidance on the setup process and Arduino Graphics Library Examples.

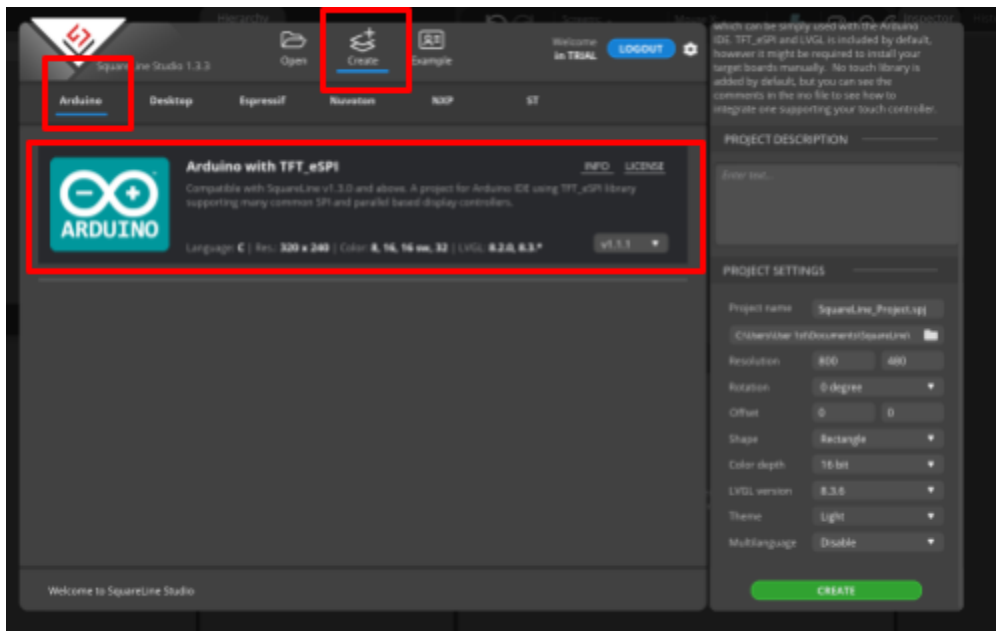
Setting up the environment for Arduino LVGL Examples, please refer to the [LVGL Example Guide for NORVI ESP32 HMI document](#). LVGL Example Guide for NORVI ESP32 HMI provides step-by-step guidance on the setup process of LVGL Library examples.

Setting up the environment for SquareLine Studio

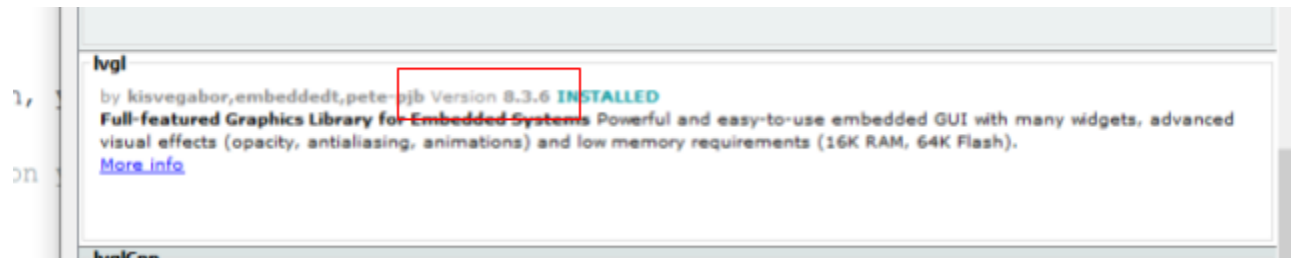
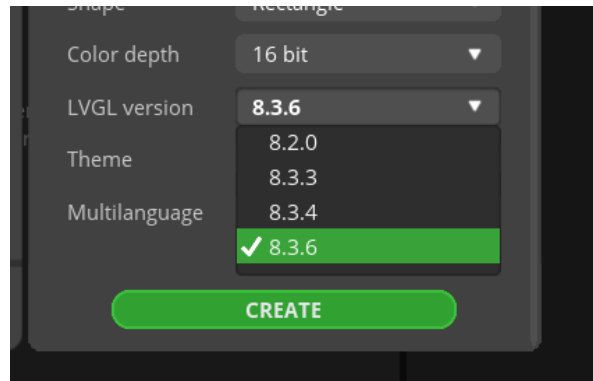
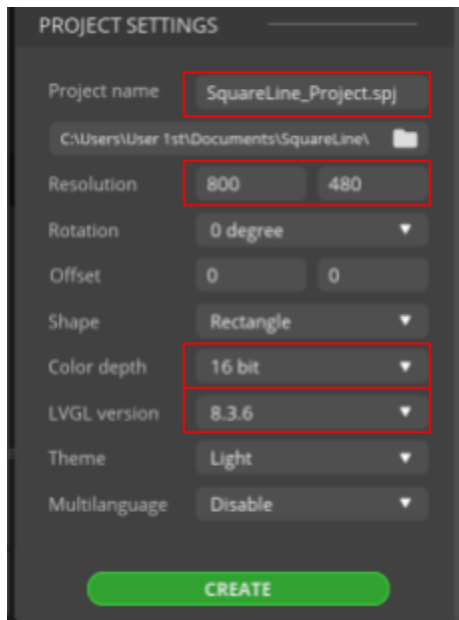
1. Installing SquareLine Studio

Start by downloading the software from the official SquareLine Studio website, and selecting the version compatible with the operating system. Once the download is complete, proceed with the installation process by executing the downloaded file.

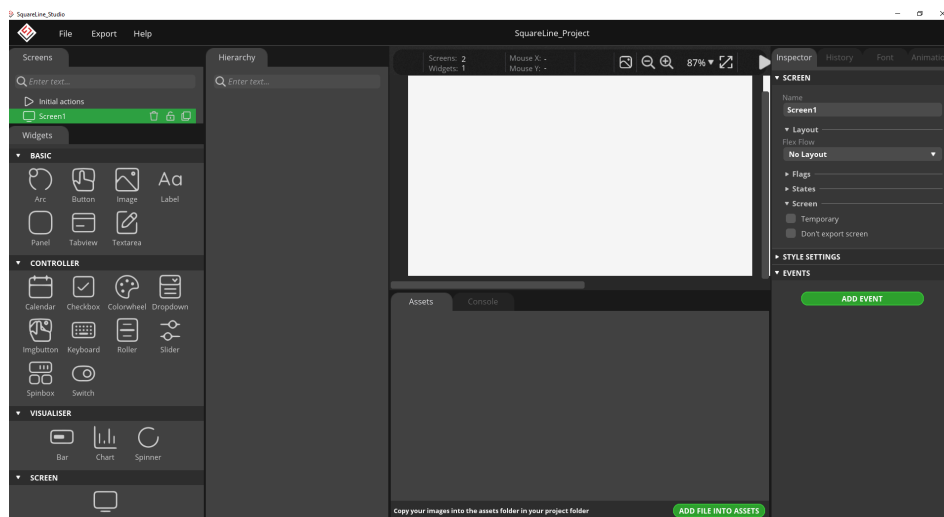
After a successful installation, open the SquareLine Studio software. A pop-up window will appear, where to select "Create/Arduino/Arduino with TFT_eSPI" to initiate the Arduino project.



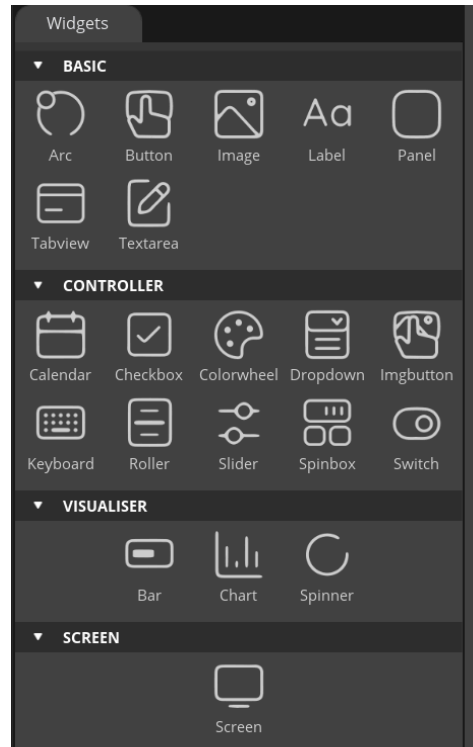
To customize settings, navigate to the options provided below. When choosing the LVGL version, please refer to the downloaded library within Arduino.



After confirming the LVGL version, select " Create " in the lower right corner, which will unveil the following interface.



There is a series of widgets organized in the left-hand widget column. These selections include basic widgets (ARC, buttons, image, label...), control widgets (calendar, keyboard, slider...), and visualization widgets (bars, charts, and spinner). Additionally, a category dedicated to screens can be found.



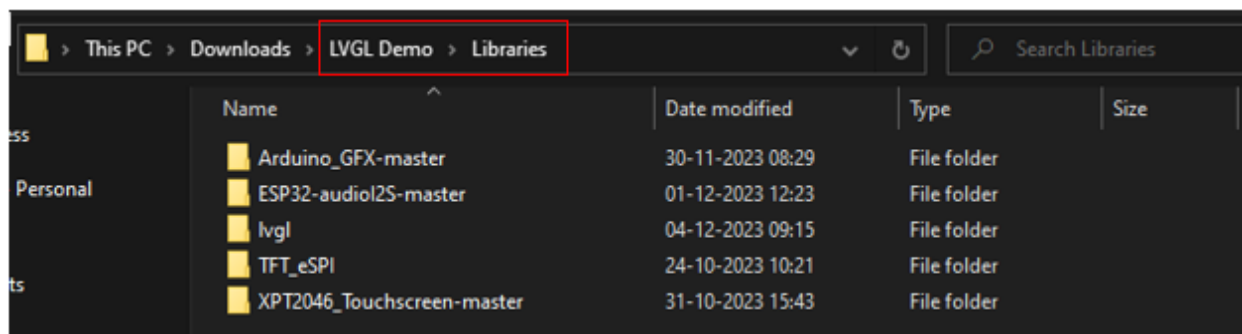
2. Adding supported libraries

Download the *LVGL Demo* folder for the essential libraries, and files from the link below.

<https://norvi.lk/wp-content/uploads/2023/12/LVGL-Demo.zip>

Copy the libraries from the downloaded *LVGL Demo* folder into the *Arduino/library* folder.

If there is already a downloaded library file, please replace it with this library file.

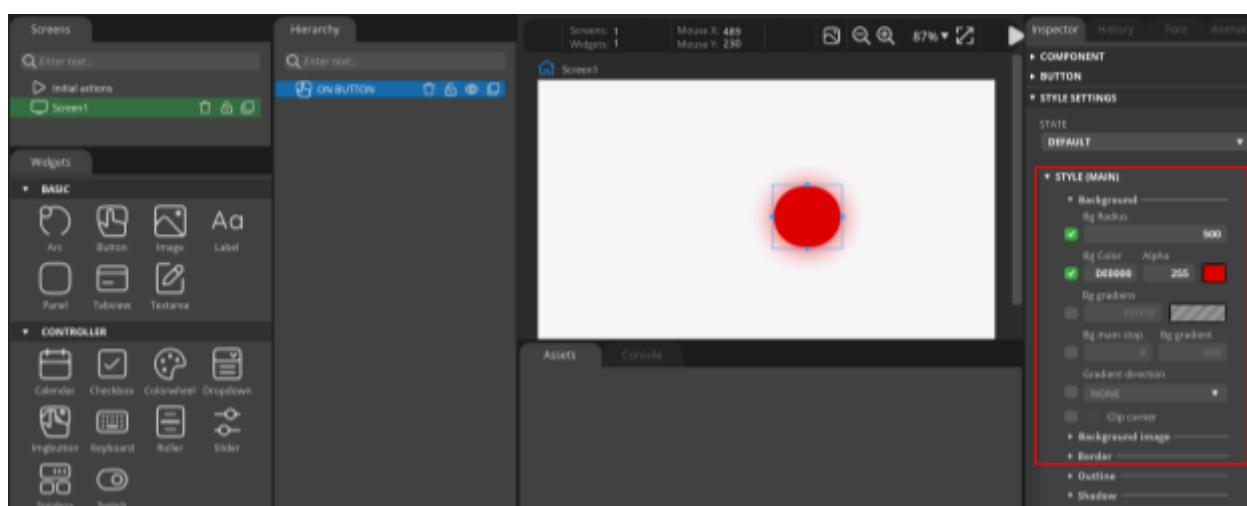
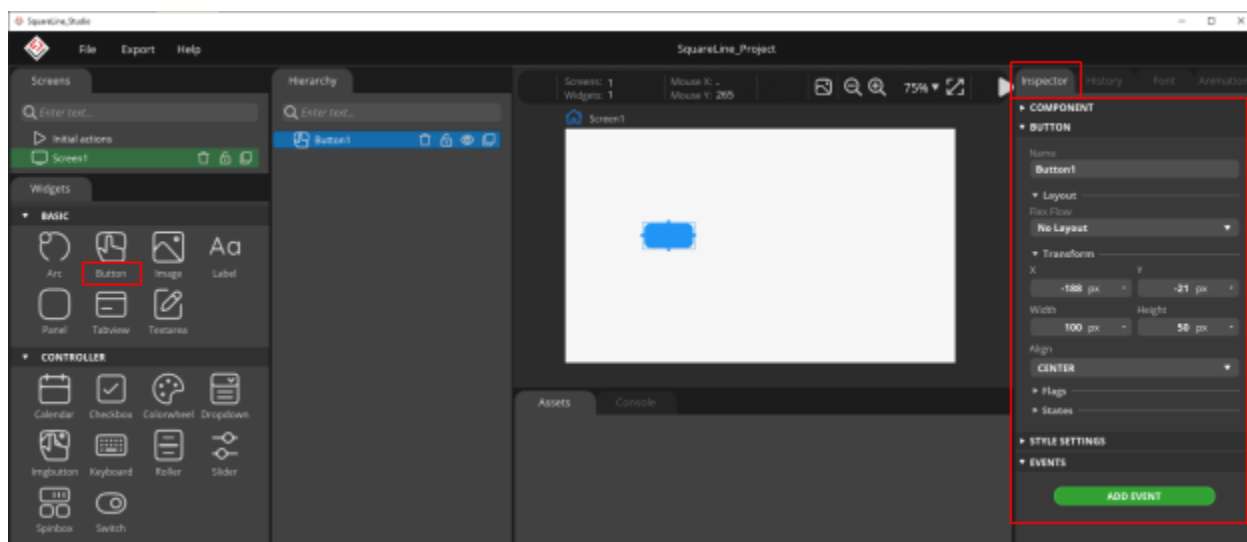


Initiating a Project with SquareLine Studio

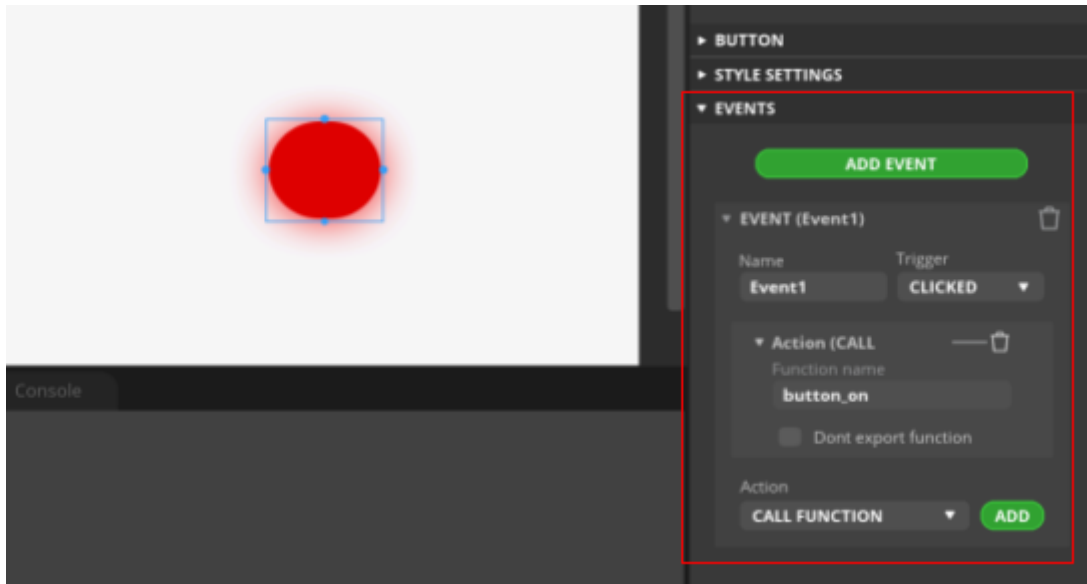
1. Create a project

Create a project with SquareLine Studio to control an LED light seamlessly. The project has to be configured to activate the LED when the "ON" button is pressed and deactivate it when the "OFF" button is selected.

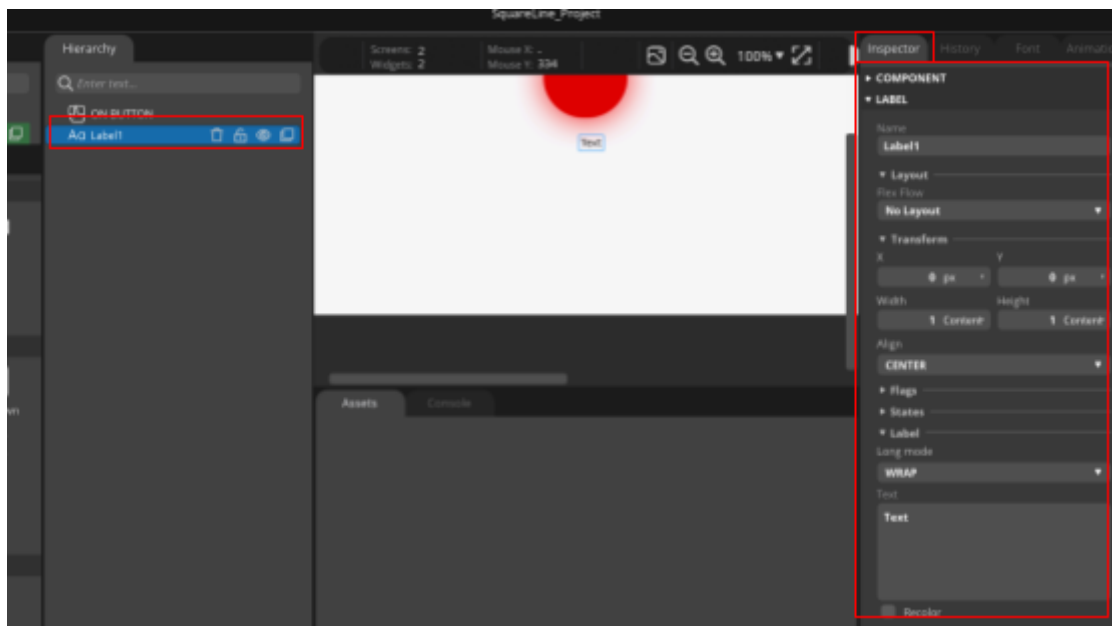
- Incorporate the buttons from the widgets bar. Utilize the Inspector Bar to configure the button's style settings, tailoring its appearance to suit your design preferences.

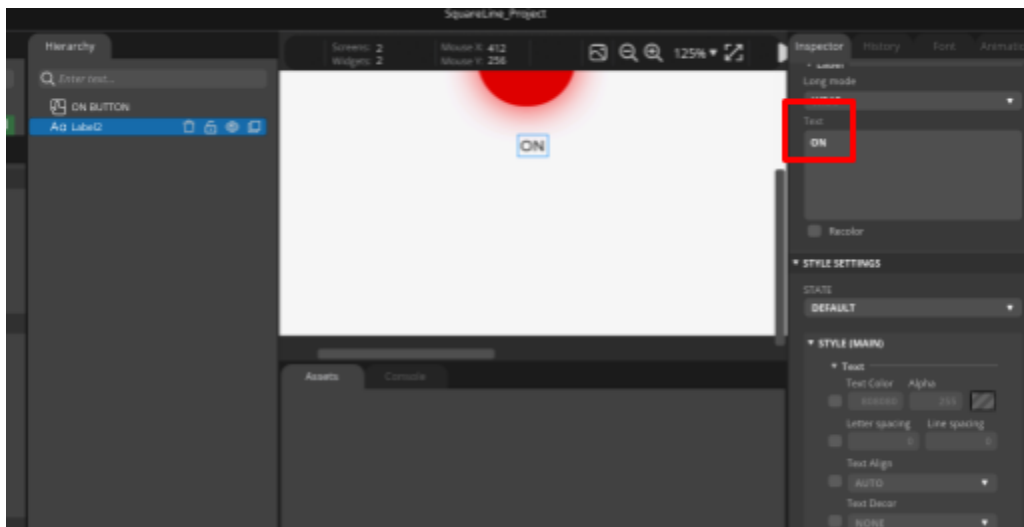


- Furthermore, leverage the Inspector Bar to add events to the button. When defining actions for events in SquareLine Studio, it is important to choose actions that align with the purpose of the event. In the context of LED on and off, "call function" must be selected.

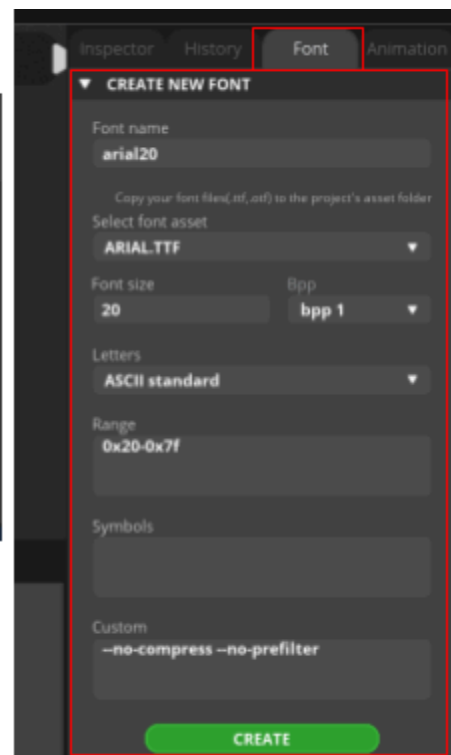
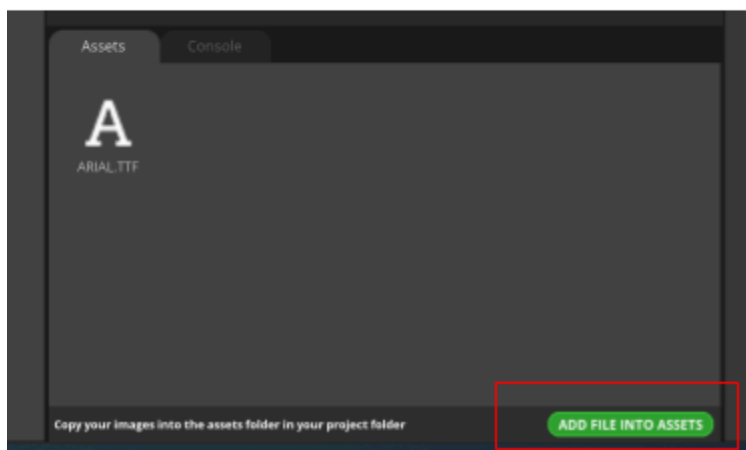


- Incorporate the labels from the widgets bar. Utilize the Inspector Bar to configure the text that should be included in the label and to configure the label's style settings, tailoring its appearance to suit your design preferences.

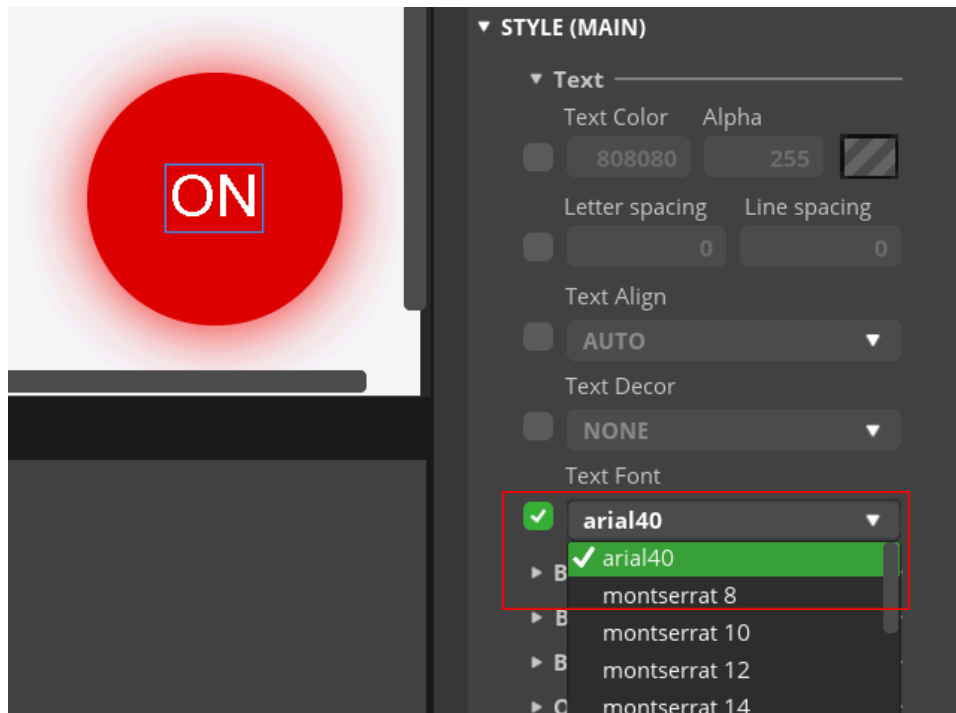




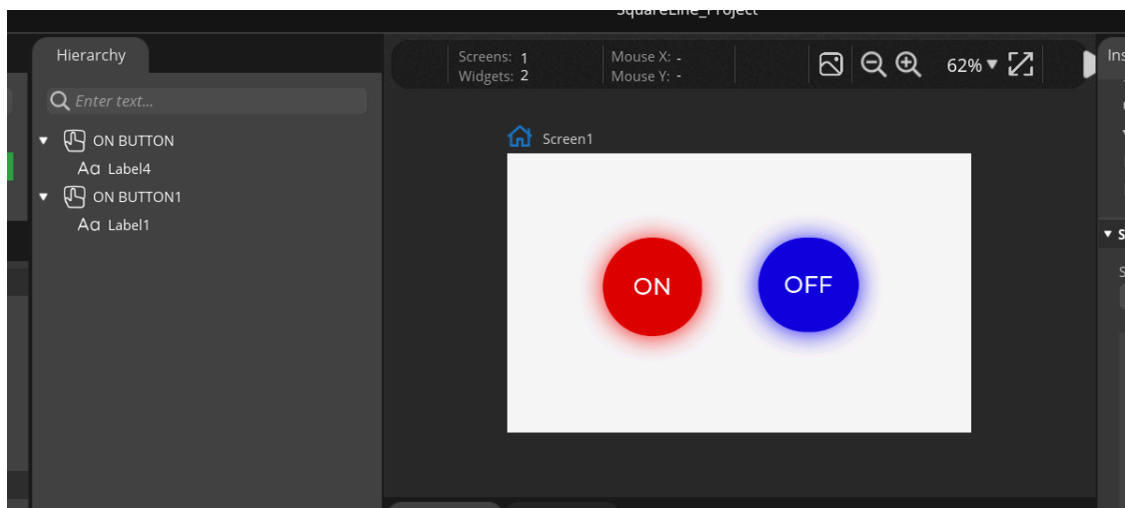
- It should incorporate custom fonts when adding labels within SquareLine Studio.
- Download a preferred font and seamlessly integrate it into the project by utilizing the "ADD FILE INTO ASSETS" option located at the bottom of the software interface.
- Craft a font to your project's needs by navigating to the 'font menu' located in the top right corner of SquareLine Studio.
- Select the desired size, allowing for meticulous customization of the font to suit the project's requirements.



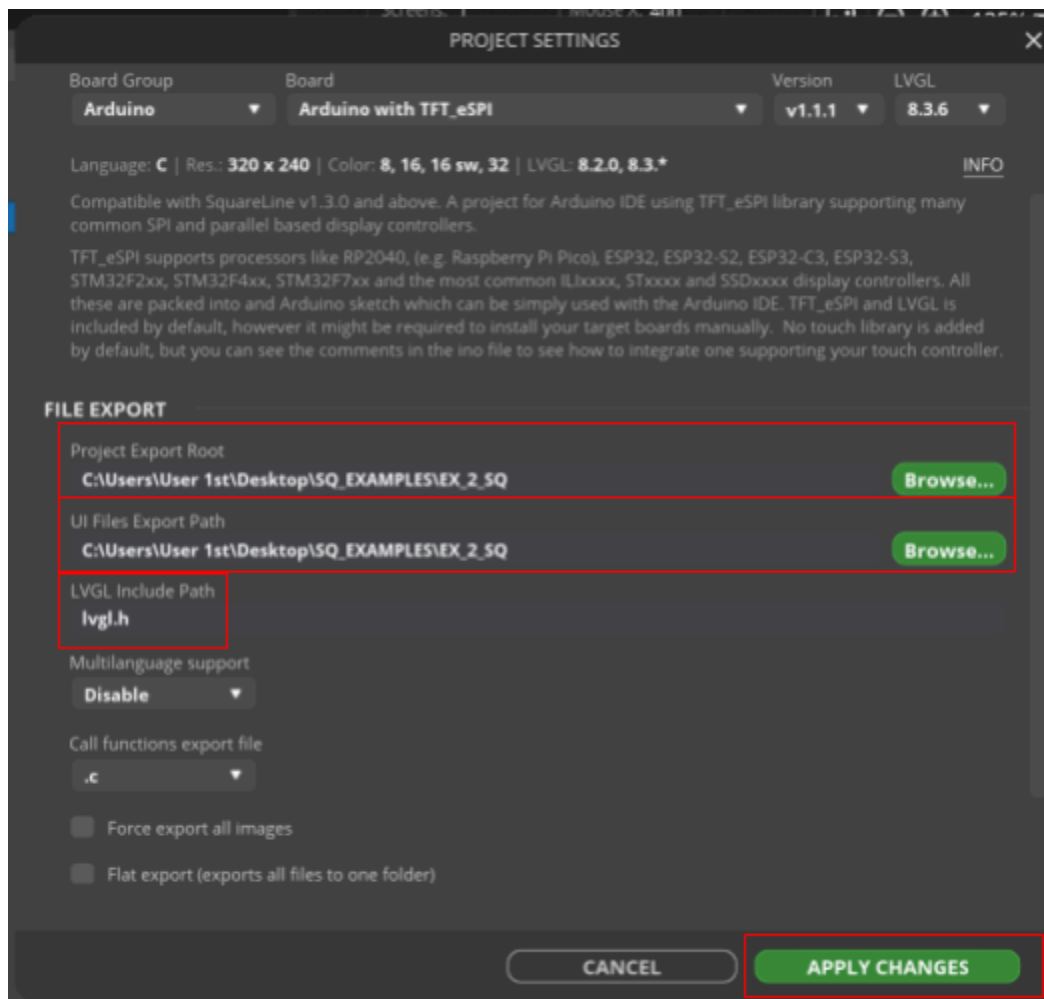
- Then the font can be selected when changing the label settings.



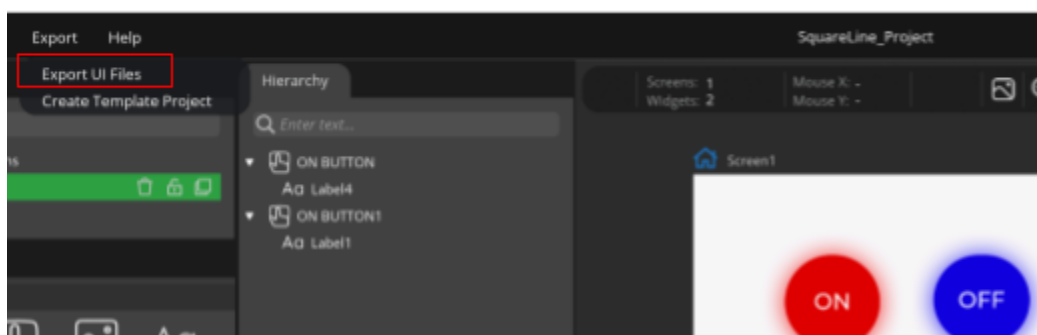
- Make a button for the "off" state, complementing the previously configured "on" state. Then the entire project comes together seamlessly.

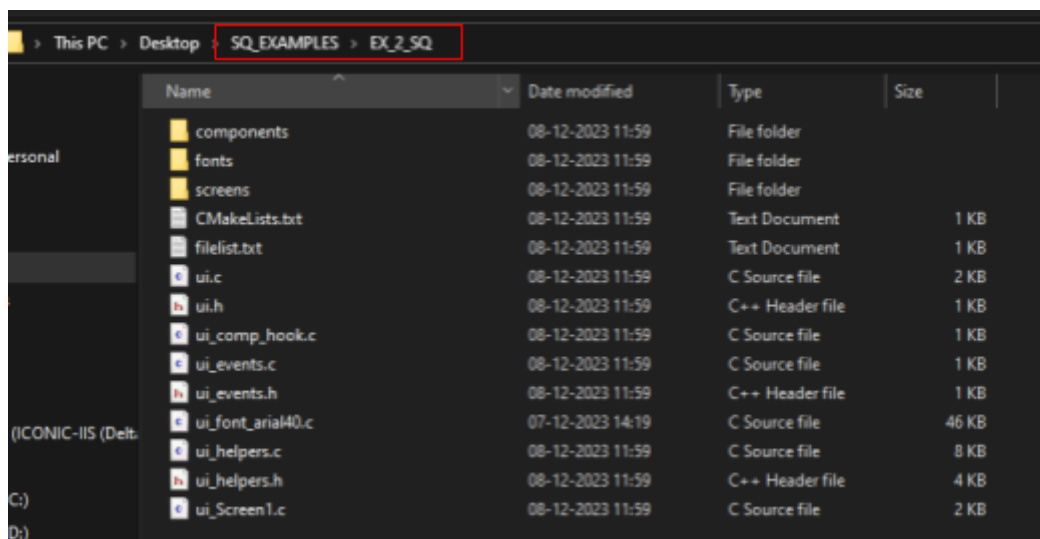


- Select the *File/Project Settings* and make the changes in the pop-up window. Browse, select the *file export* path, and *call functions export file* type. After making the changes, proceed by selecting "APPLY CHANGES" located in the lower right corner.



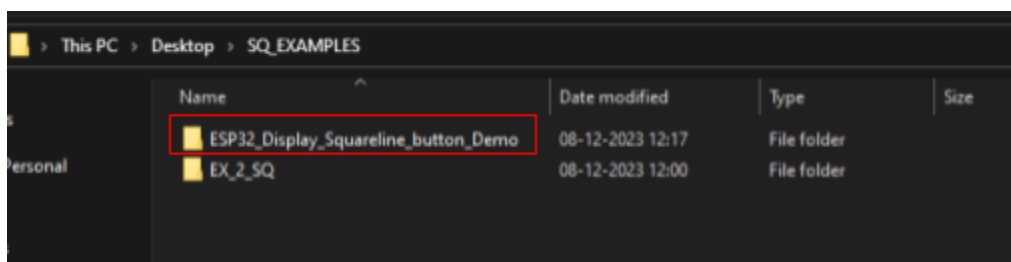
Then select *File/Save* to save the project file and *Export/Export UI Files* to export the project.



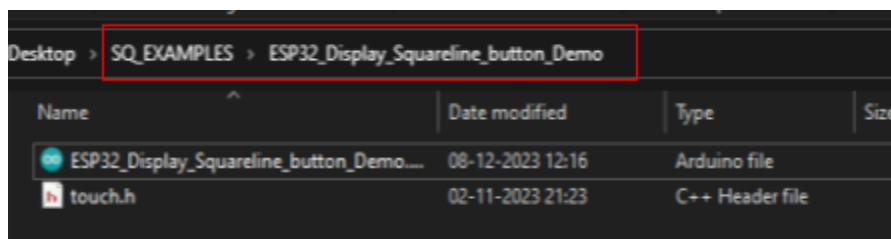


2. Pre-requisites.

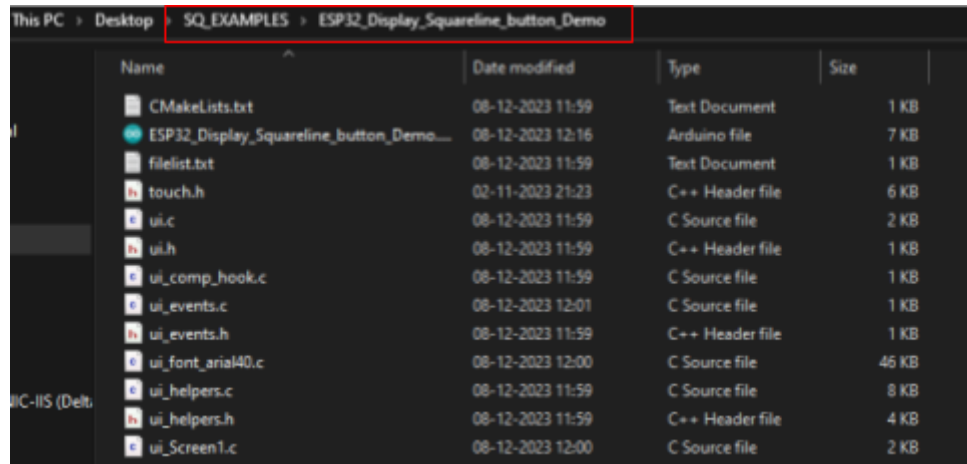
Copy the *ESP32_Display_Squareline_button_Demo* folder from the downloaded *LVGL Demo/NORVI Squareline demo* folder and paste it to the location of the created SQ project folder.



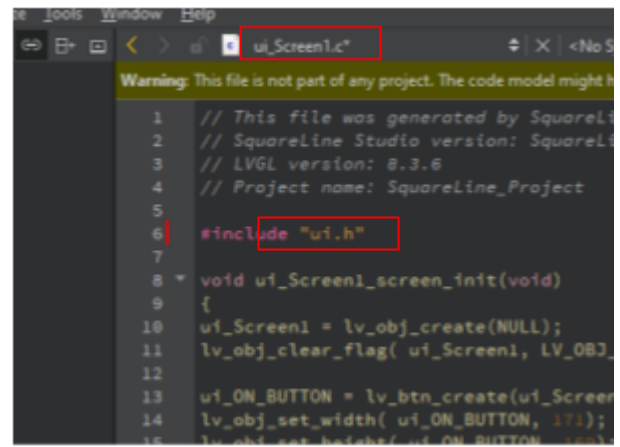
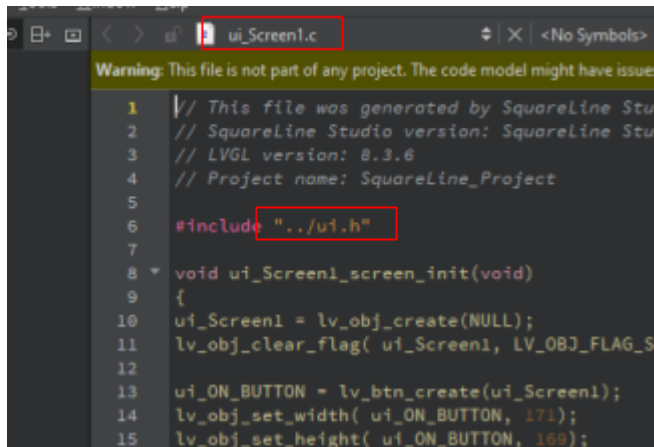
Remove all existing files from the *ESP32_Display_Squareline_button_Demo* folder in the location of the created SQ project folder, without the *ESP32_Display_Squareline_button_Demo.ino* file and the *touch.h* file.



Copy all the files including the insides of the folders in the created SQ project folder and paste them to the *ESP32_Display_Squareline_button_Demo* folder in the SQ project folder.



Open *ui_Screen1.c* remove ". ./" and save. If the project contains images and fonts, do the same for those files.

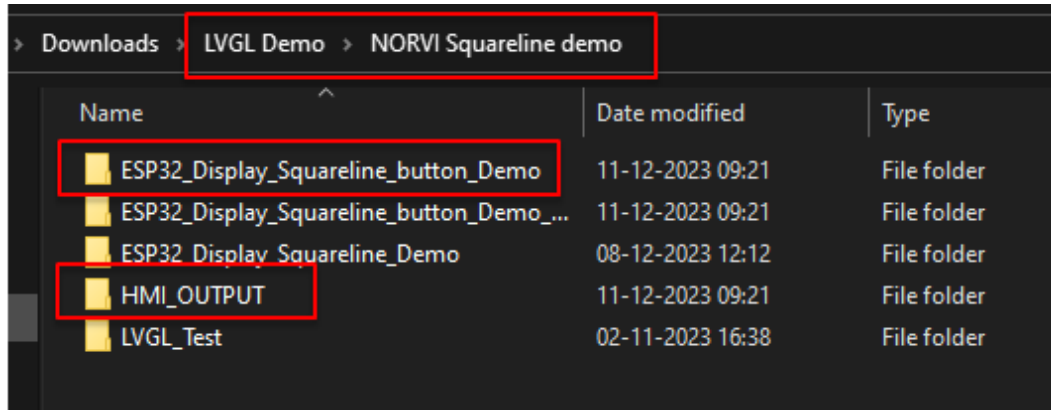


Since this project controls an LED, the code must have some modifications. Because of the function called when creating the project.

For that, open the *ESP32_Display_Squareline_button_Demo.ino* file in the SQ project folder.

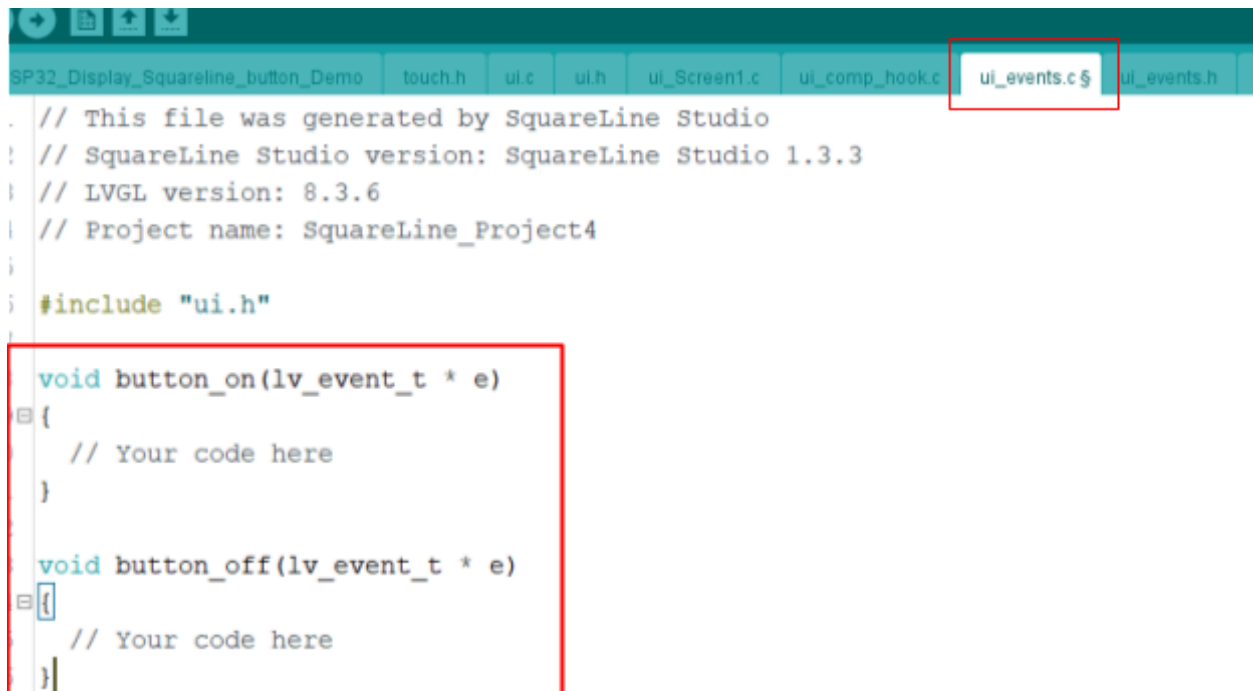
Also, open *hmi_output.ino* file in *LVGL Demo* folder.

hmi_output - Outputs control code for Norvi HMI.



Without the inside part of the void loop() add all the parts of the hmi output.ino file to the ESP32_Display_Squareline_button Demo.ino file to the correct places.

Then select the *ui_events.c* in ESP32_Display_Squareline_button_Demo.ino file and copy the highlighted part.



Paste it in the main code, (before the *void setup()* function).



```
ESP32_Display_Squareline_button_Demo$ touch.h ui.c ui.h ui_S
142     data->state = LV_INDEV_STATE_REL;
143 }
144     delay(15);
145 }
146 #endif
147
148 void button_on(lv_event_t * e)
149 {
150     // Your code here
151 }
152
153 void button_off(lv_event_t * e)
154 {
155     // Your code here
156 }
157
158 void setup()
159 {
160     Serial.begin(9600);
161     Serial.println("LVGL Widgets Demo");
```

Then add the below coding part as shown in the figure.

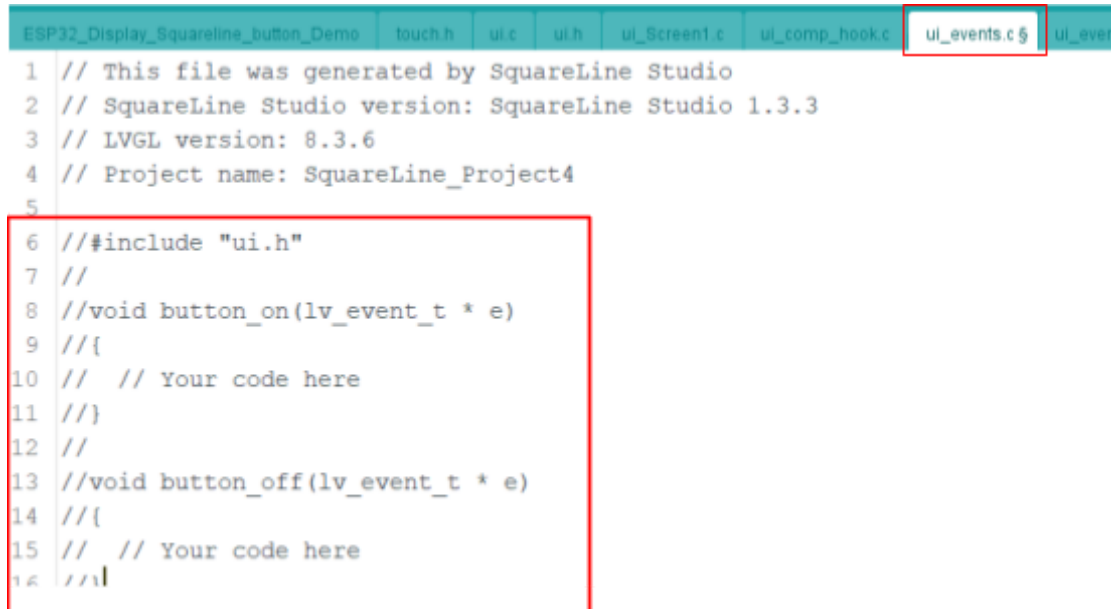
```
void button_on(lv_event_t * e);
{
    writeOutput(GPI05, 1);
    writeOutput(GPI06, 1);
    writeOutput(GPI07, 1);
    writeOutput(GPI08, 1);
}

void button_off(lv_event_t * e);
{
    writeOutput(GPI05, 0);
    writeOutput(GPI06, 0);
    writeOutput(GPI07, 0);
    writeOutput(GPI08, 0);
}
```



```
ESP32_Display_Squareline_button_Demo$ touch.h ui.c ui.h ui_Screen1.x ui_S
142     data->state = LV_INDEV_STATE_REL;
143 }
144     delay(15);
145 }
146 #endif
147
148 void button_on(lv_event_t * e)
149 {
150     writeOutput(GPI05, 1);
151     writeOutput(GPI06, 1);
152     writeOutput(GPI07, 1);
153     writeOutput(GPI08, 1);
154 }
155
156 void button_off(lv_event_t * e)
157 {
158     writeOutput(GPI05, 0);
159     writeOutput(GPI06, 0);
160     writeOutput(GPI07, 0);
161     writeOutput(GPI08, 0);
162     delay(300);
163 }
164
165 void setup()
166 {
167     Serial.begin(9600);
```

Comment the full code of the *ui_events.c*.



```
ESP32_Display_Squareline_button_Demo touch.h ui.c ui.h ui_Screen1.c ui_comp_hook.c ui_events.c § ui_events.c

1 // This file was generated by SquareLine Studio
2 // SquareLine Studio version: SquareLine Studio 1.3.3
3 // LVGL version: 8.3.6
4 // Project name: SquareLine_Project4
5
6 // #include "ui.h"
7 //
8 // void button_on(lv_event_t * e)
9 //{
10 // // Your code here
11 //}
12 //
13 // void button_off(lv_event_t * e)
14 //{
15 // // Your code here
16 //}
```

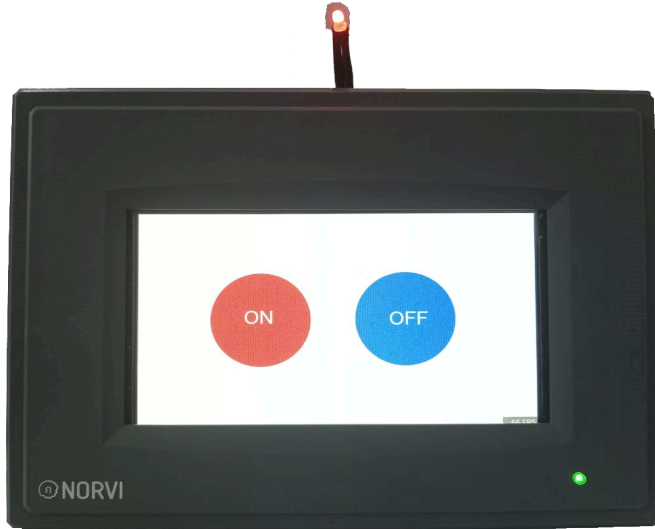
Choose the chip and settings according to the chip as shown below.



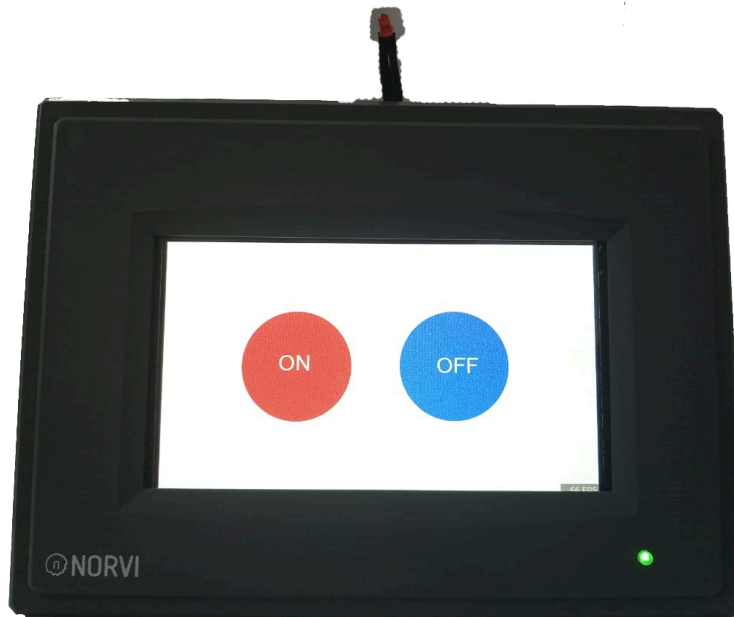
3. Compile and Upload.

Check the pin configuration of the program. After compiling and uploading the program, the User interface of the NORVI HMI should be as below.

- On state



- Off state





REACH US

Technical Support

E-mail : support@icd.lk

Sales Inquiries

E-mail bhanuka@icd.lk

Web www.norvi.lk/support

Order Online www.norvi.lk