NXP Semiconductors
User's Guide

Document identifier: GUIGUIDERUG
Rev. 3, 10 May 2021

GUI Guider User's Guide

Contents

Chapter 1
Introduction

The GUI Guider is built on the Light and Versatile Graphics Library (LVGL) library. GUI Guider provides an IDE to design embedded graphic application UI using drag and drop widgets and helps in the editing process. The software facilitates the UI design for graphic application on embedded devices.

This document describes GUI Guider and targets embedded GUI Application developers with a basic knowledge of C on NXP MCU devices.

The major sections of this user guide are:

- Introduction — General information and feature list of GUI Guider.
- Installation — Steps to install the software and set up the environment.
- Usage — Steps to use GUI Guider and design GUI Application.
- Widget Details — Description of supported widgets and attributes.
- Event Details — Supported events and actions of each widget.
- Work with MCUXpresso — Steps to import GUI APP source in MCUXpresso project, cross-build, and deploy on device.

1.1 Supported features

- IDE
  - Supports Windows 10 and Ubuntu 20.04
  - Multi-language (English, Chinese) for IDE
  - Compatible with LVGL v7.4.0, MCUXpresso 11.3.0, and MCU SDK 2.9.x
  - Project management: create, import, edit, delete
  - What You See Is What You Get (WYSIWYG) UI design by drag & drop
  - Multi-page application design
  - Shortcut of bring forward and backward, copy, paste, delete, undo, redo
  - Code viewer for UI definition JSON file
  - Navigation bar to view the selected source file
  - LVGL C code auto-generation
  - Widget attributes group and setting
  - Screen copy function
  - GUI editor zoom in and zoom out
  - Multiple fonts support and 3rd party fonts import
  - Customizable Chinese character scope
  - Widgets alignment: left, center and right
  - PXP acceleration enable and disable
  - VGLite acceleration support
  - Supports default style and custom style
  - Integrated demo applications
  - Compatible with MCUXpresso IDE

  - Real-time log display
- Widgets
  - Supports 35 widgets
    - Button (5): button, image button, checkbox, button group, switch
    - Form (4): label, drop-down list, text area, calendar
    - Table (8): table, tab, message box, container, chart, canvas, list, window
    - Shape (10): arc, line, roller, led, spinbox, Ospinbox, gauge, line meter, color, spinner
    - Image (3): image, animation image, 3D image
    - Progress (2): bar, slider
    - Others (3): page, tile view, keyboard
    - Animation: animation image, GIF to animation, animation easing and path
    - Support event trigger and action selection, custom action code
    - Support tileview design by drag & drop operation in editor
    - Support parent/child hierarchy for container, tabview and tileview
    - Chinese display
- Target devices
  - NXP i.MX RT1050, i.MX RT1062, i.MX RT1064, i.MX RT1170 and i.MX RT595
  - NXP LPC54S018 and LPC54628
  - Device template, auto-build ,and auto-deploy for supported platforms
  - Run simulator on X86 host

Chapter 2
Installation

This section describes the prerequisities and steps to install GUI Guider.

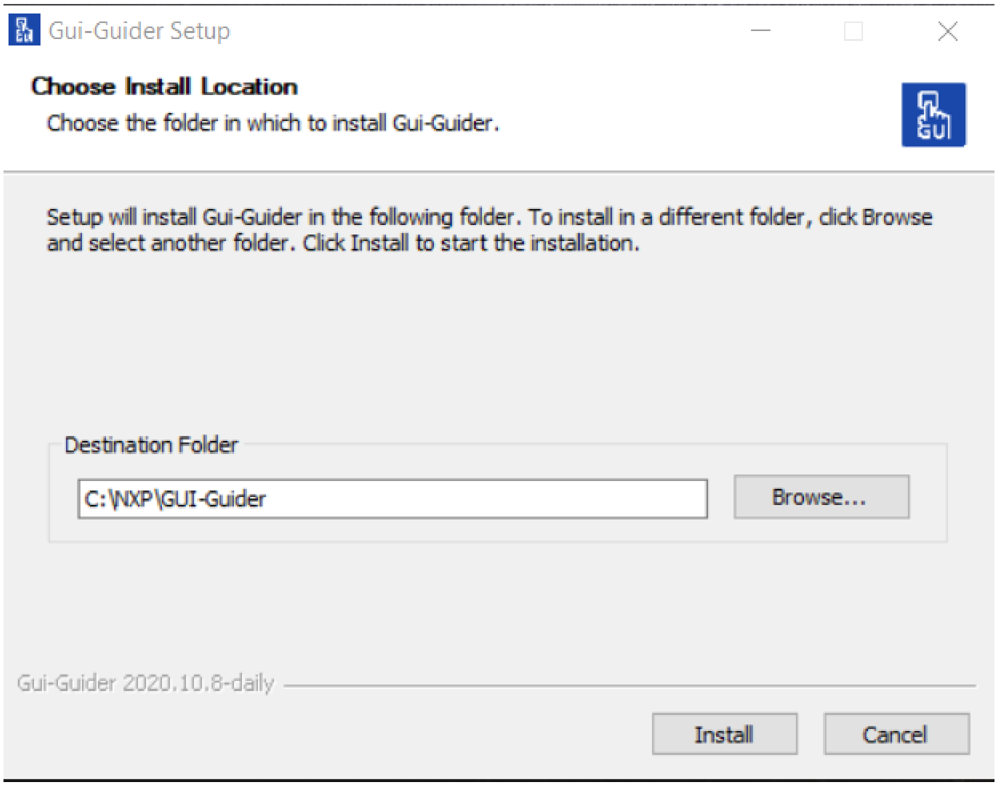
2.1 Prerequisites of Windows 10

- PC with Windows 10
- Java Runtime Environment, JRE-8 is verified

2.2 Install GUI Guider on Windows 10

To install GUI Guider on Windows, perform the following steps.

1. Download the installer from www.nxp.com/gui-guider.
2. Double-click the installer to install the software.



Gui-Guider Setup

Choose Install Location

Choose the folder in which to install Gui-Guider.

Setup will install Gui-Guider in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

Destination Folder

C:\NXP\GUI-Guider

Figure 1. Choose the install location

2.3 Set up runtime environment on Windows 10

This section describes the steps to install Java Runtime Environment and the Git command-line tool.

2.3.1 Install Java Runtime Environment

If your computer does not have JRE installed, perform the following steps.

1. Get JRE installer from the official website.

2. Double-click the installer to install JRE.
3. Add *jre path* in the system path environment variable.
    a. Open **Control Panel** and search *env*.
    b. Choose **Edit the system environment variables**.
    c. Click the **Environment Variables…** button.
    d. Under the **System Variables** section (the lower half), find the row with **Path** in the first column, and click **Edit**.
    e. The **Edit environment variable** UI appears.
    f. Click **New** and type in the java path. For example, *C:\Program Files (x86)\Common Files\Oracle\Java\javapath*.
    g. Click **OK** to save your change.

2.4 Uninstall GUI Guider on Windows 10

To uninstall GUI Guider on Windows, perform the following steps.

1. Open **Control Panel > Programs and Features**.

2. Select **Gui-Guider-<version>** and click **Uninstall**.

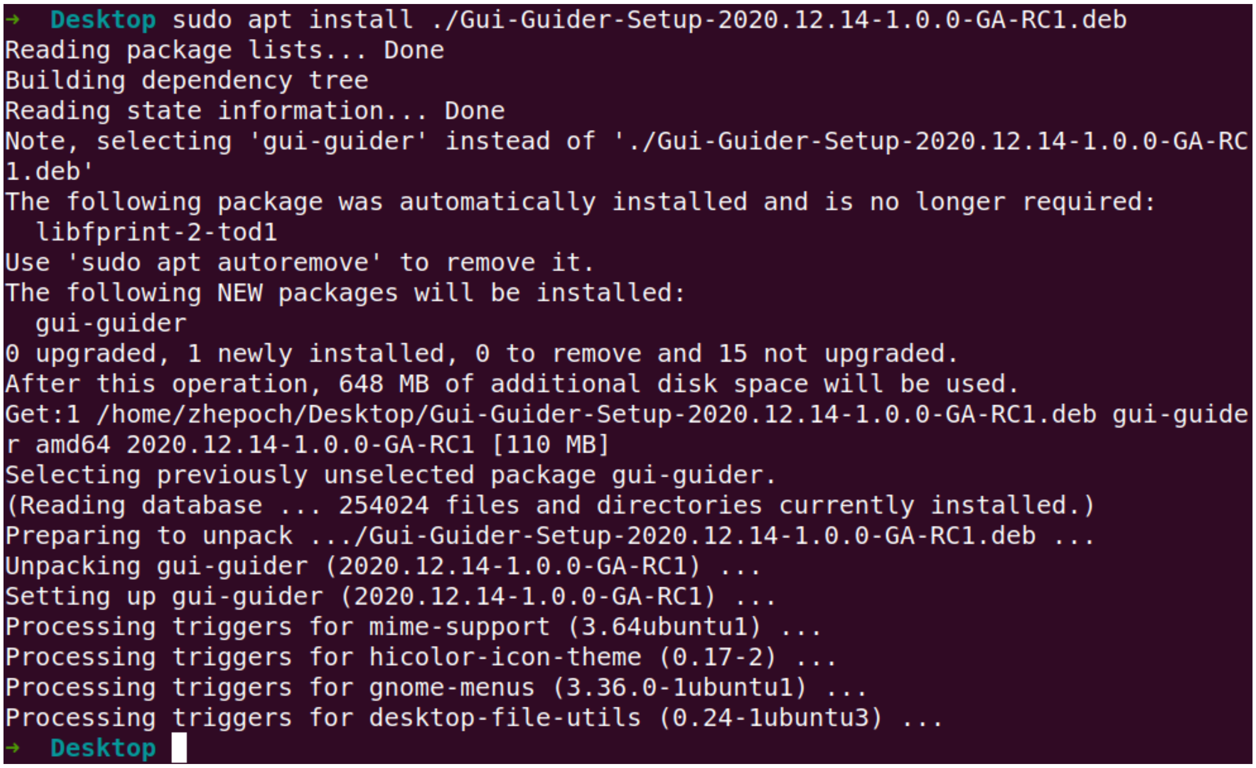
2.5 Prerequisites of Ubuntu 20.04

- PC with Ubuntu 20.04

2.6 Install GUI Guider on Ubuntu 20.04

1. Download the installer from www.nxp.com
2. Run command to install the software.

```
$ sudo apt install ./Gui-Guider-Setup-1.0.0-GA.deb
```



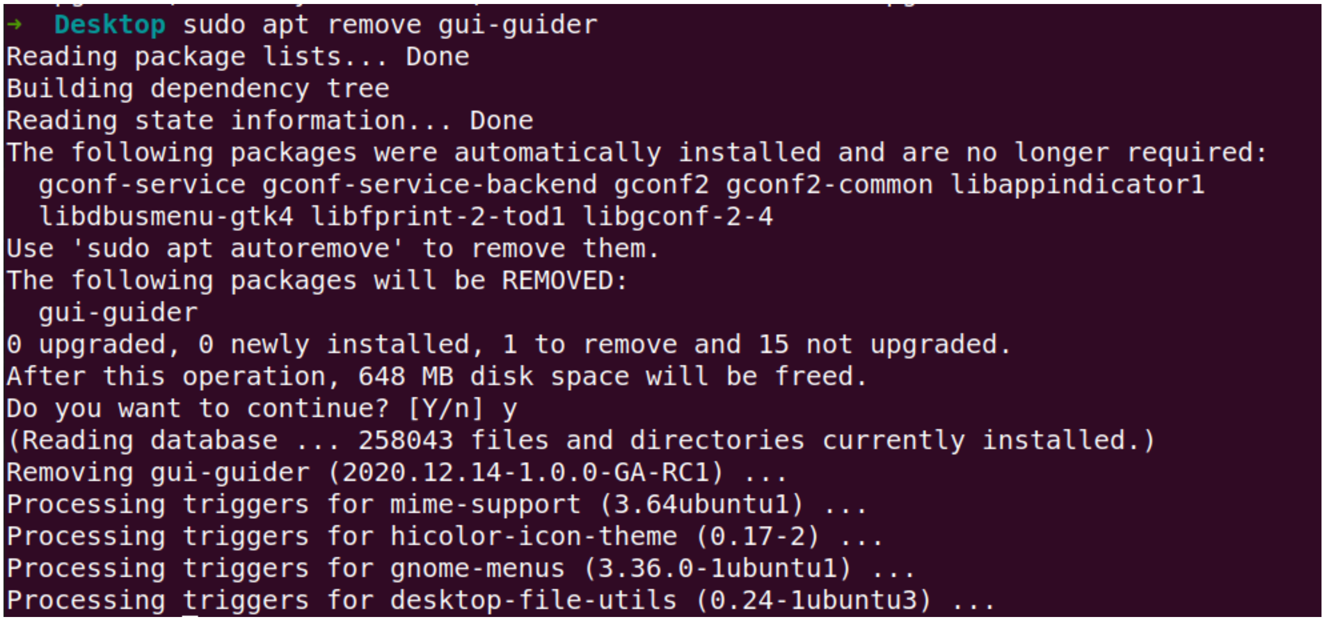
```
➜  Desktop sudo apt install ./Gui-Guider-Setup-2020.12.14-1.0.0-GA-RC1.deb
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'gui-guider' instead of './Gui-Guider-Setup-2020.12.14-1.0.0-GA-RC
1.deb'
The following package was automatically installed and is no longer required:
  libfprint-2-tod1
Use 'sudo apt autoremove' to remove it.
The following NEW packages will be installed:
  gui-guider
0 upgraded, 1 newly installed, 0 to remove and 15 not upgraded.
After this operation, 648 MB of additional disk space will be used.
Get:1 /home/zhepoch/Desktop/Gui-Guider-Setup-2020.12.14-1.0.0-GA-RC1.deb gui-guide
r amd64 2020.12.14-1.0.0-GA-RC1 [110 MB]
Selecting previously unselected package gui-guider.
(Reading database ... 254024 files and directories currently installed.)
Preparing to unpack .../Gui-Guider-Setup-2020.12.14-1.0.0-GA-RC1.deb ...
Unpacking gui-guider (2020.12.14-1.0.0-GA-RC1) ...
Setting up gui-guider (2020.12.14-1.0.0-GA-RC1) ...
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for gnome-menus (3.36.0-1ubuntu1) ...
Processing triggers for desktop-file-utils (0.24-1ubuntu3) ...
➜  Desktop
```

Figure 2. Install GUI Guider on Ubuntu

2.7 Uninstall GUI Guider on Ubuntu 20.04

Run the following command in the command line tool.

```
$ sudo apt remove gui-guider
```



```
➜  Desktop sudo apt remove gui-guider
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  gconf-service gconf-service-backend gconf2 gconf2-common libappindicator1
  libdbusmenu-gtk4 libfprint-2-tod1 libgconf-2-4
Use 'sudo apt autoremove' to remove them.
The following packages will be REMOVED:
  gui-guider
0 upgraded, 0 newly installed, 1 to remove and 15 not upgraded.
After this operation, 648 MB disk space will be freed.
Do you want to continue? [Y/n] y
(Reading database ... 258043 files and directories currently installed.)
Removing gui-guider (2020.12.14-1.0.0-GA-RC1) ...
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for gnome-menus (3.36.0-1ubuntu1) ...
Processing triggers for desktop-file-utils (0.24-1ubuntu3) ...
```

Figure 3. Uninstall GUI Guider on Ubuntu

Chapter 3
GUI Guider usage

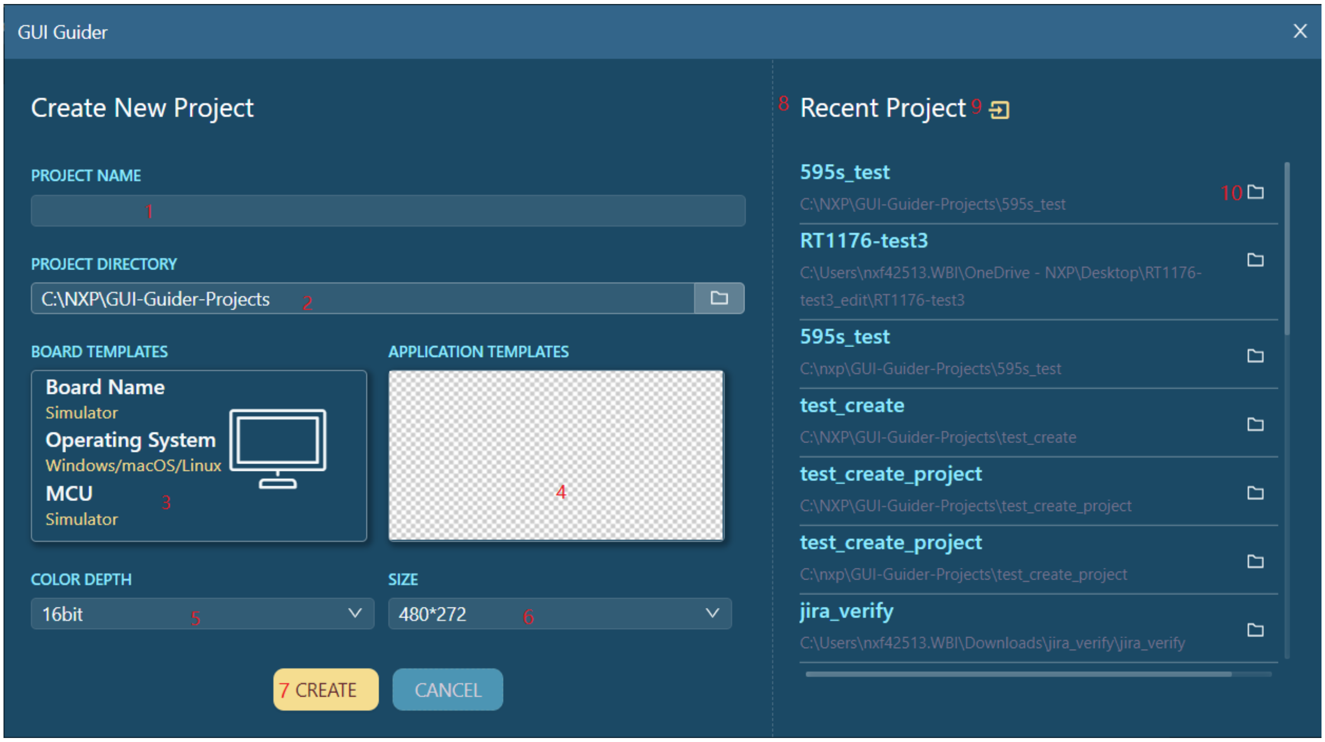
This chapter describes the GUI Guider interface elements.

3.1 Create a project

Table 1 describes the elements in the Create a new project wizard

NOTE

The value in **PROJECT NAME** and **PROJECT DIRECTORY** fields cannot incude space character.



PROJECT DIRECTORY: C:\NXP\GUI-Guider-Projects

Board Name
Simulator
Operating System
Windows/macOS/Linux
MCU
Simulator

COLOR DEPTH: 16bit

SIZE: 480*272

Recent Project

595s_test
C:\NXP\GUI-Guider-Projects\595s_test

RT1176-test3
C:\Users\nxf42513.WBI\OneDrive - NXP\Desktop\RT1176-test3_edit\RT1176-test3

595s_test
C:\nxp\GUI-Guider-Projects\595s_test

test_create
C:\NXP\GUI-Guider-Projects\test_create

test_create_project
C:\NXP\GUI-Guider-Projects\test_create_project

test_create_project
C:\nxp\GUI-Guider-Projects\test_create_project

jira_verify
C:\Users\nxf42513.WBI\Downloads\jira_verify\jira_verify

Figure 4. Create New Project wizard

Table 1. Create New Project wizard

Label	Description
1	Project name
2	Project directory
3	Board template
4	Application template (optional)
5	Color depth supported by current device

Table continues on the next page...

Table 1. Create New Project wizard (continued)

Label	Description
6	Screen resolution supported by current device
7	Button of create new project
8	History project list
9	Button of open existing project
10	Icon to open project folder in file explorer

3.2 Menu list

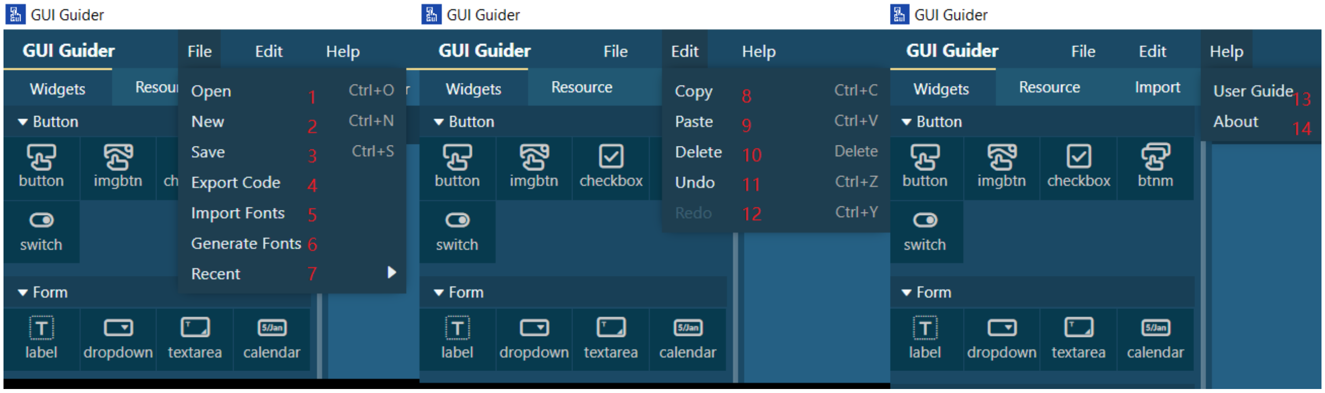


Figure 5. GUI Guider menu list

Table 2. Description of menu items

Label	Description
1	Import and open an existing project.
2	Create new project.
3	Save project.
4	Export project's generated code for the run target. Ensue that **Generate Code** is executed first.
5	Import fonts to this project.
6	Generate other C font code.
7	Recent project list.
8	Copy selected widget(s).
9	Paste widget(s).
10	Delete selected widget(s).
11	Undo action.
12	Redo action.

Table continues on the next page...

Table 2. Description of menu items (continued)

Label	Description
13	Open user guider.
14	Show version information.

3.3 Generate Fonts window

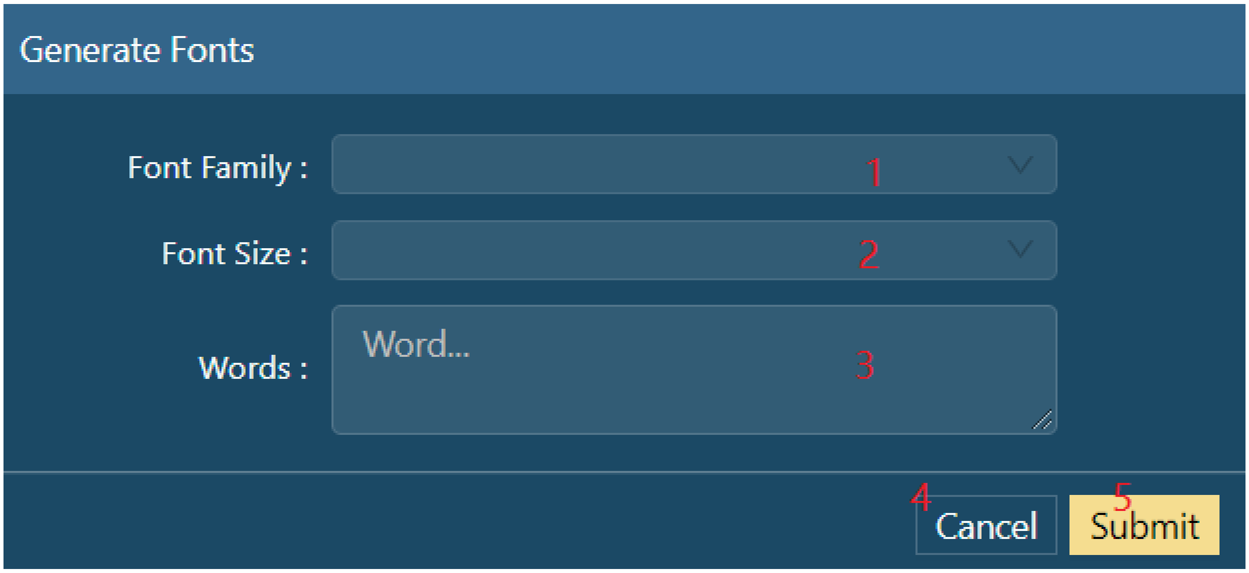


Figure 6. Generate Fonts window

Table 3. Description of Generate Fonts window options

Label	Description
1	The name of font family. NOTE Should be English.
2	Select font size.
3	Input the characters to be generated.
4	Cancel button.
5	Submit button.

3.4 Select Board Templates window

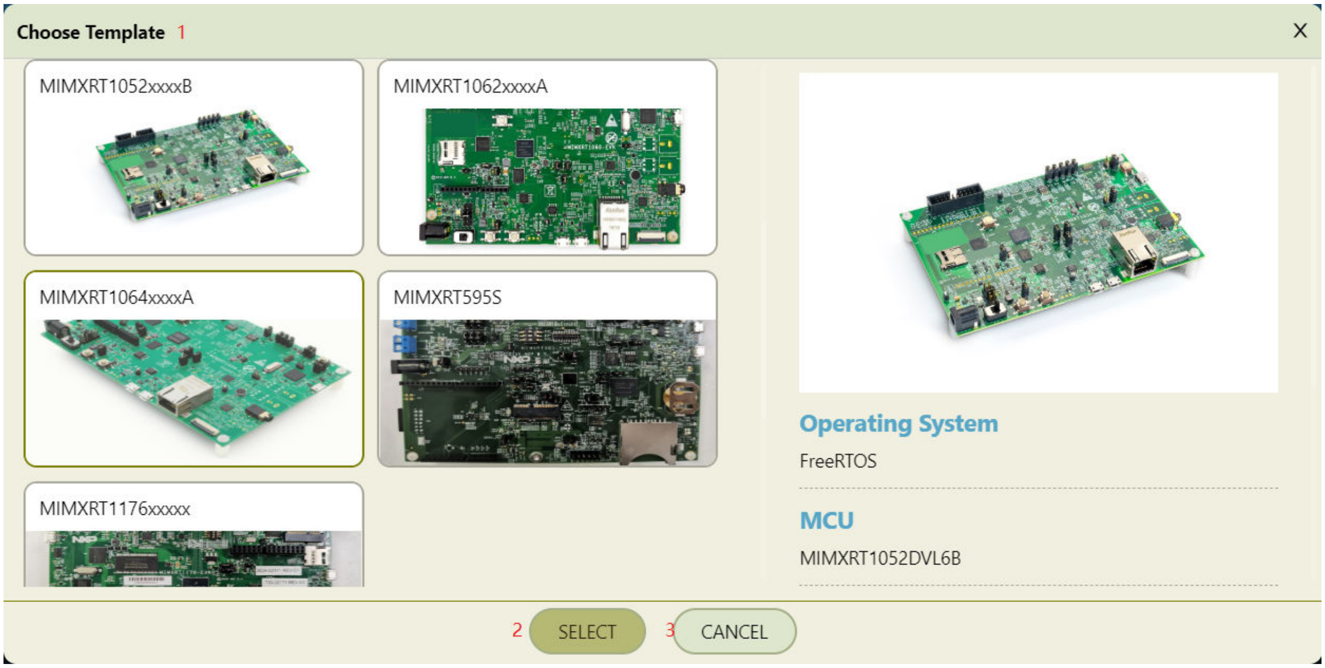


Figure 7. Select Board Templates window

Table 4. Description of Select Board Templates window

Label	Description
1	List of boards for selection.
2	Select button.
3	Cancel button.

3.5 Select Application Template window

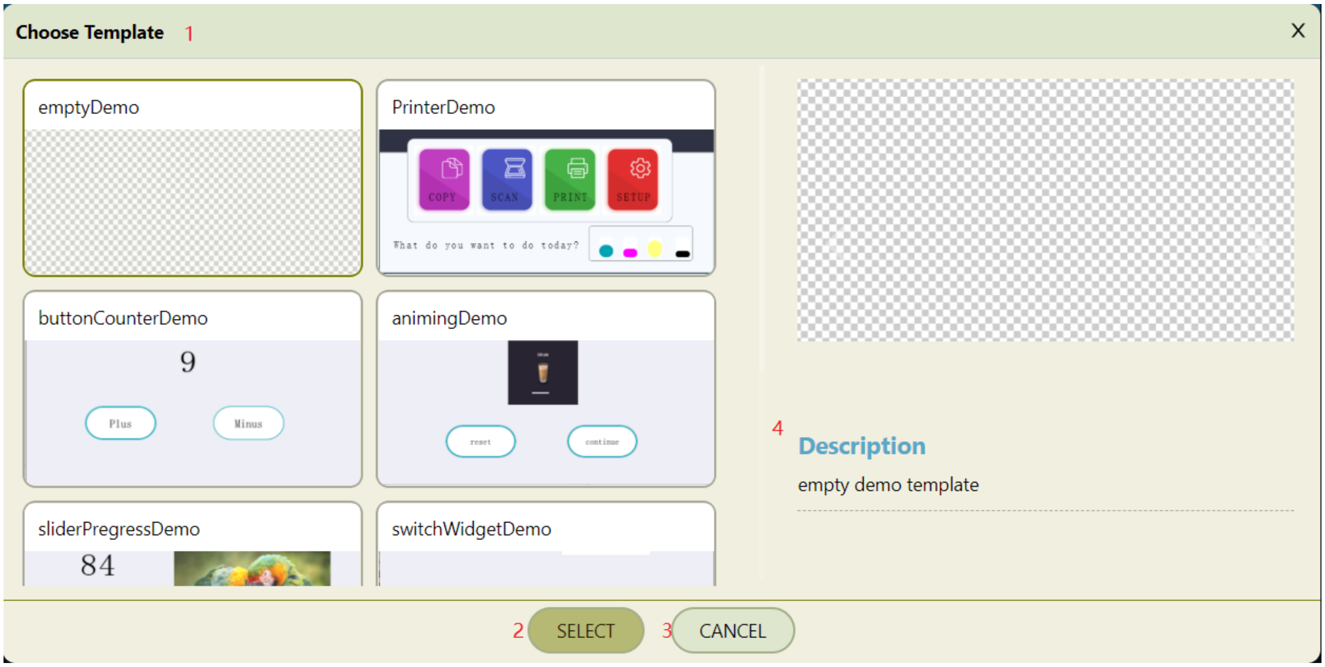


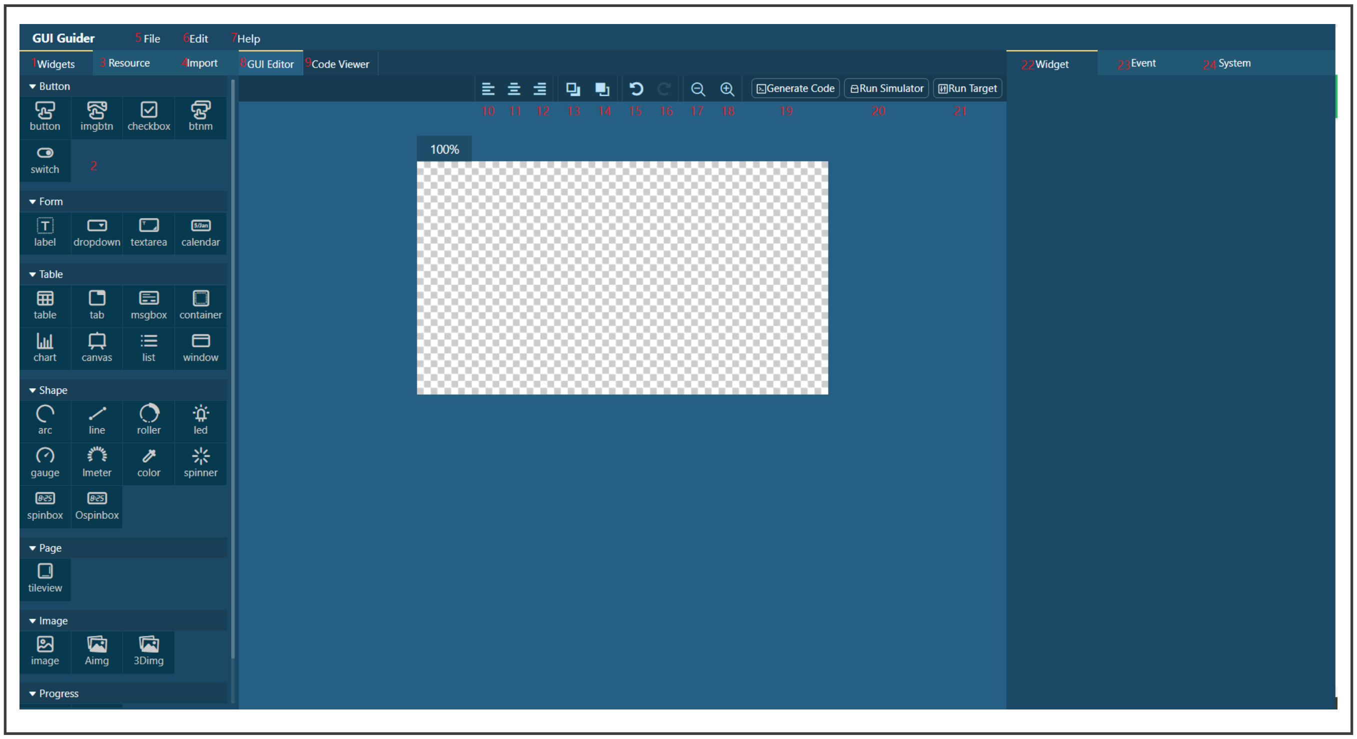
Figure 8. Select Application Template window

Table 5. Description of Select Application Template window

Label	Description
1	List of demos for selection.
2	Select button.
3	Cancel button.

3.6 IDE - Main window

Table 6 describes the elements of the Main window.



NOTE

To use the left align, center align, or right align buttons, you need to first select the widgets. There are two ways to select the widgets. Your can either click on the baseline widget and then use "Ctrl" + mouse click to individually select the other widgets to align. Or, you can click and drag over the widgets to select the widgets as a group. Next, you can click the appropriate align button and align the selected widgets.

Table 6. IDE - Main window

Label	Description
1	Widget tab. Link to widget details.
2	Supported widgets. Link to widget details.
3	Resource tab (Lists pages and widgets of current GUI APP). Link to Resource window
4	Import tab (Lists images used by current GUI APP). Link to Import window.
5	File menu: Open, New, Save, Import Fonts, Recent projects.
6	Edit menu: Copy, Paste, Delete, Undo, Redo.
7	Help menu: User Guide, About.
8	GUI editor tab.
9	GUI source code viewer tab.
10	Left align selected widgets.
11	Center align selected widgets.

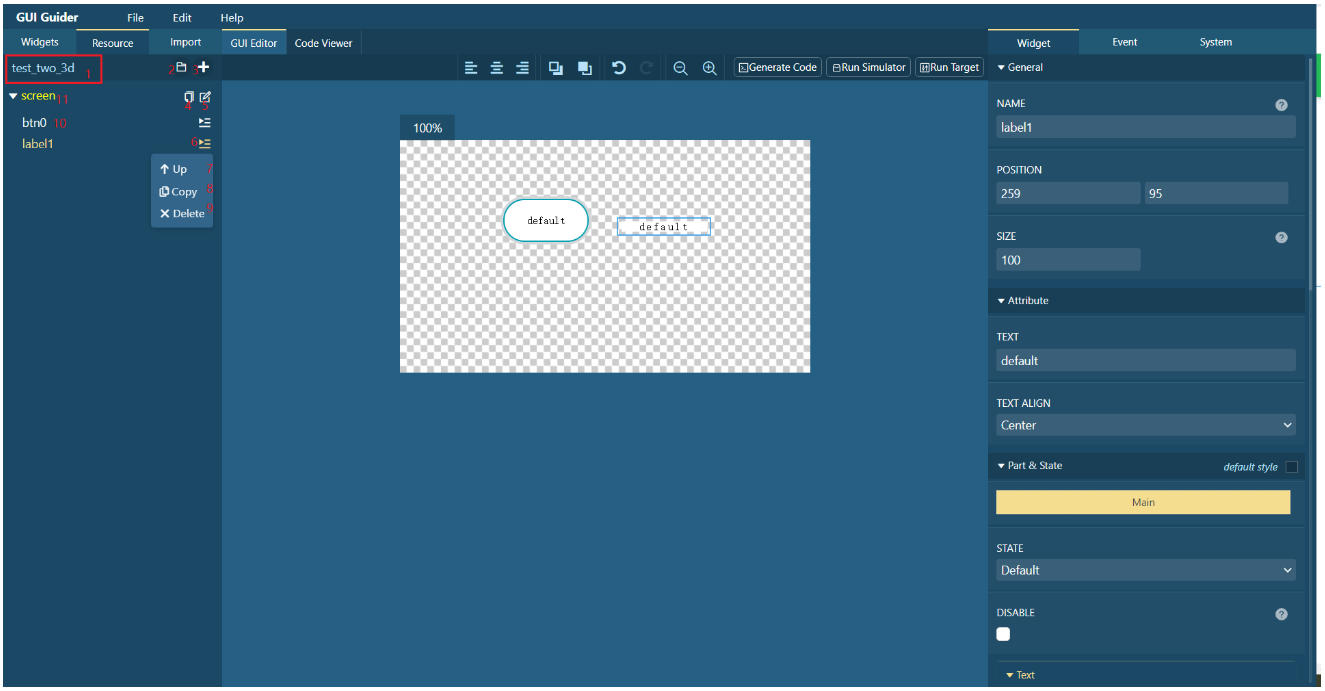
Table continues on the next page...

Table 6. IDE - Main window (continued)

Label	Description
12	Right align selected widgets.
13	Bring selected widget backward.
14	Bring selected widget forward.
15	Undo.
16	Redo.
17	Zoom out the GUI editor.
18	Zoom in the GUI editor.
19	Generate LVGL C code of GUI APP. More details
20	Run GUI APP in simulator. More Details
21	Run GUI APP on target board. More details
22	Set attributes of selected widget. More details
23	Configure the event and action of selected widget. More details
24	Configure project properties. More details

3.7 IDE - Resource window

Table 7 describes the elements in Figure 9.



NAME: label1

POSITION: 259 95

SIZE: 100

TEXT: default

TEXT ALIGN: Center

STATE: Default

DISABLE: □

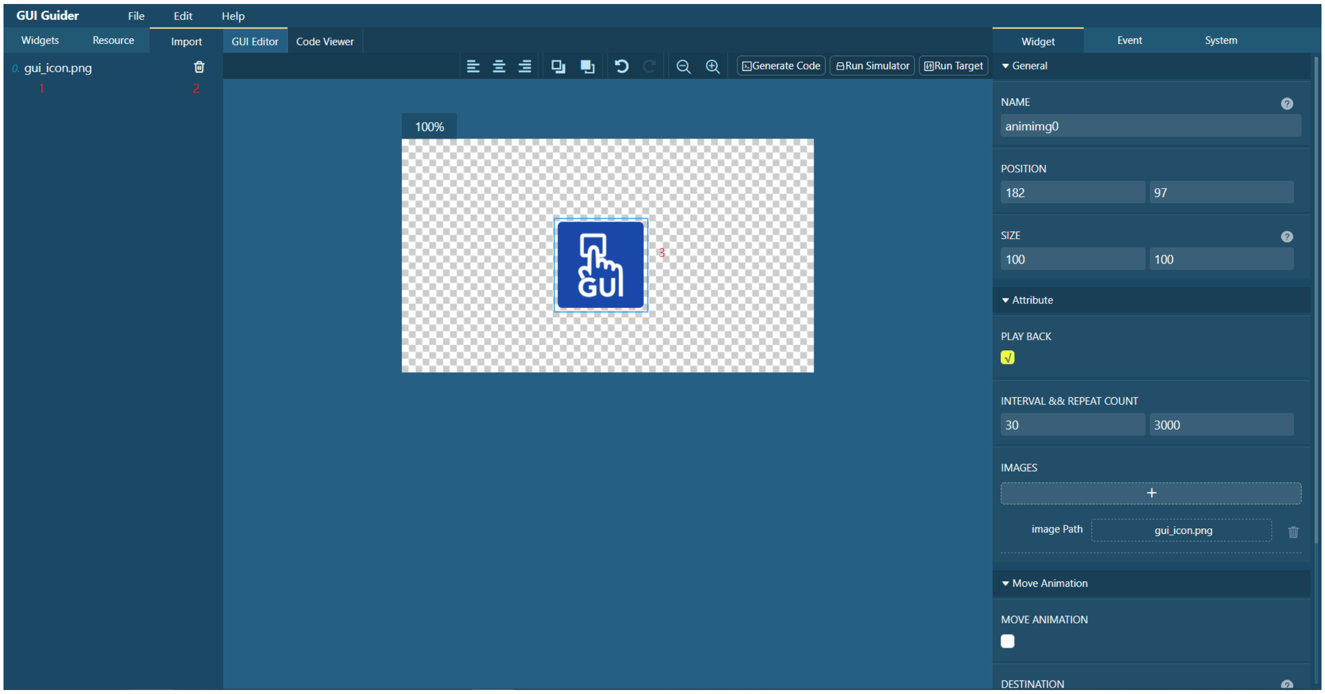
Figure 9. IDE - Resource window

Table 7. IDE - Resource window

Label	Description
1	The name of current project.
2	Open the project folder with File Explorer.
3	Add a new screen.
4	Copy the current screen.
5	Modify the screen name.
6	Open the action menu of current widget.
7	Bring forward current widget.
8	Copy selected widget.
9	Delete selected widget.
10	Widget list of current screen.
11	Screen list of current GUI application.

3.8 IDE - Import window

Table 8 describes the elements of the Figure 10.



NAME: animimg0

POSITION: 182 97

SIZE: 100 100

PLAY BACK: ☑

INTERVAL && REPEAT COUNT: 30 3000

IMAGES: +

image Path: gui_icon.png

MOVE ANIMATION: □

DESTINATION

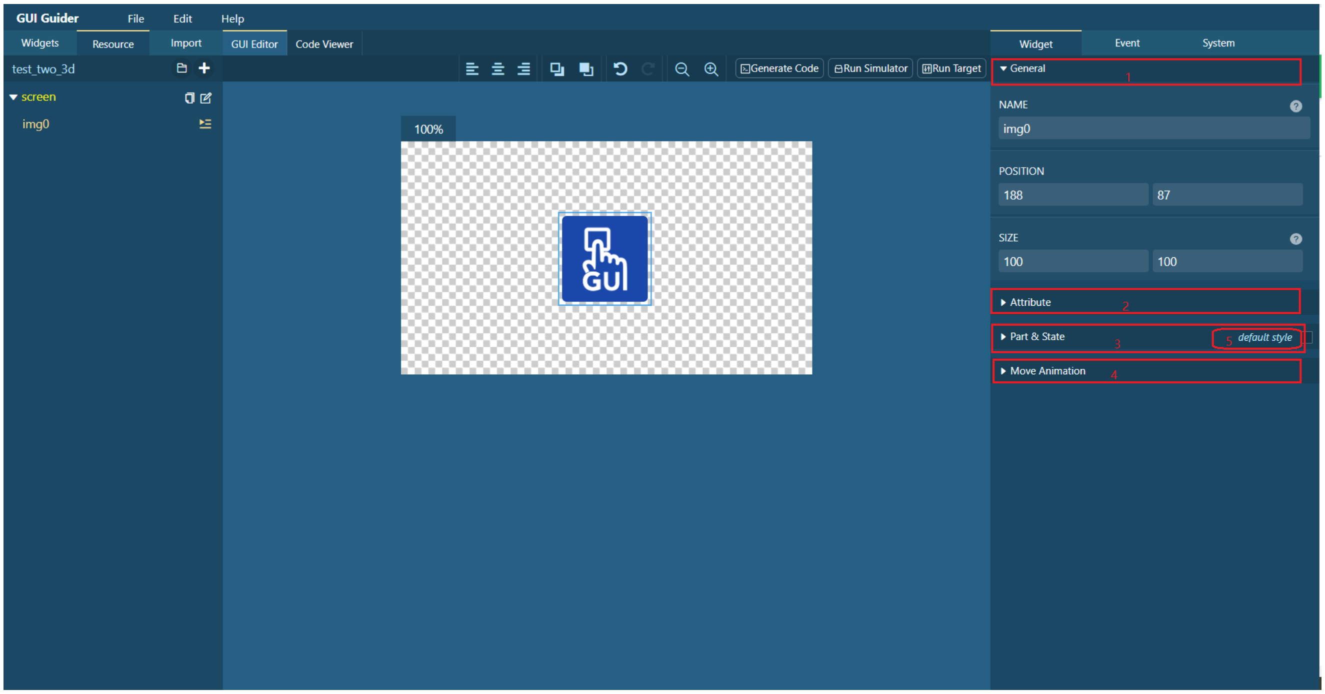
Figure 10. IDE - Import window

Table 8. IDE - Import window

Label	Description
1	Imported images of current GUI APP.
2	Delete button.
3	Drag and drop editor of GUI.

3.9 IDE - Widget Attributes window

Table 9 describes the elements of the Figure 11.



NAME
img0

POSITION
188 87

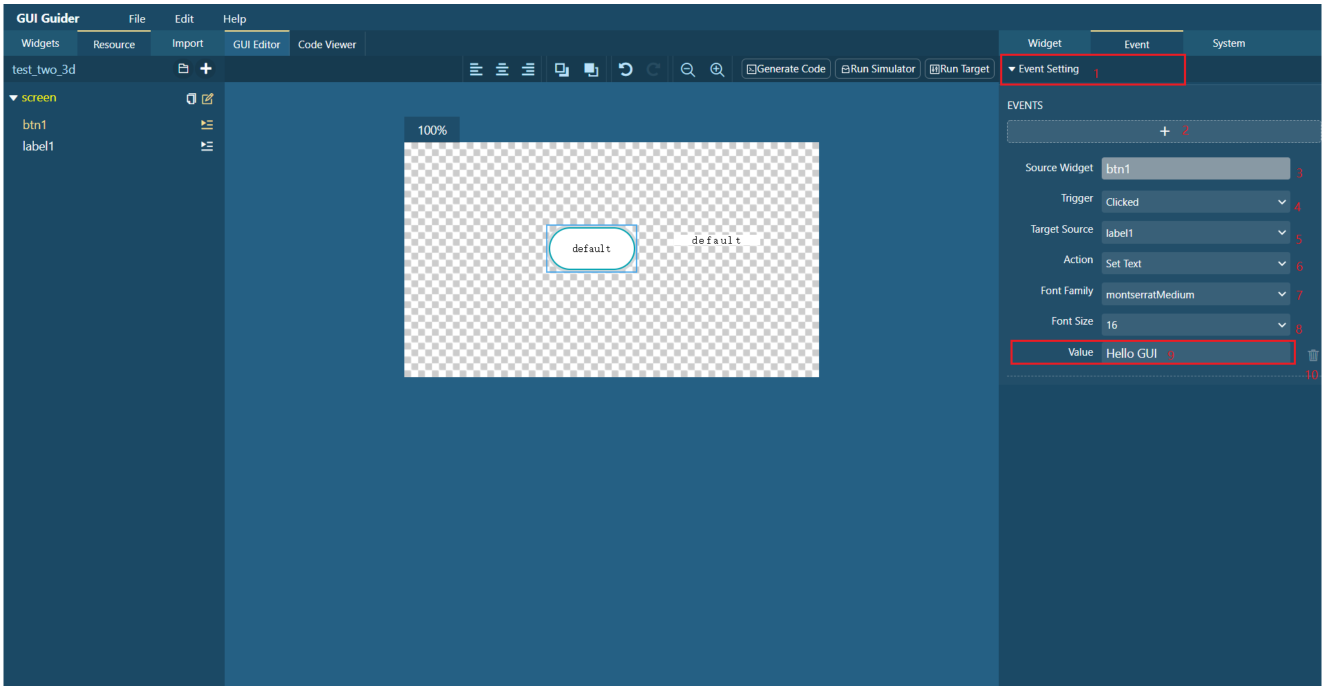
SIZE
100 100

Figure 11. IDE - Widget Attributes window

Table 9. IDE - Widget Attributes window

Label	Description
1	General attributes of the widget.
2	Widget specific attributes.
3	Style setting for the widgets.
4	Animation settings
5	Default style or custom style selection.

3.10 IDE - Widget Event window



Source Widget btn1

Trigger Clicked

Target Source label1

Action Set Text

Font Family montserratMedium

Font Size 16

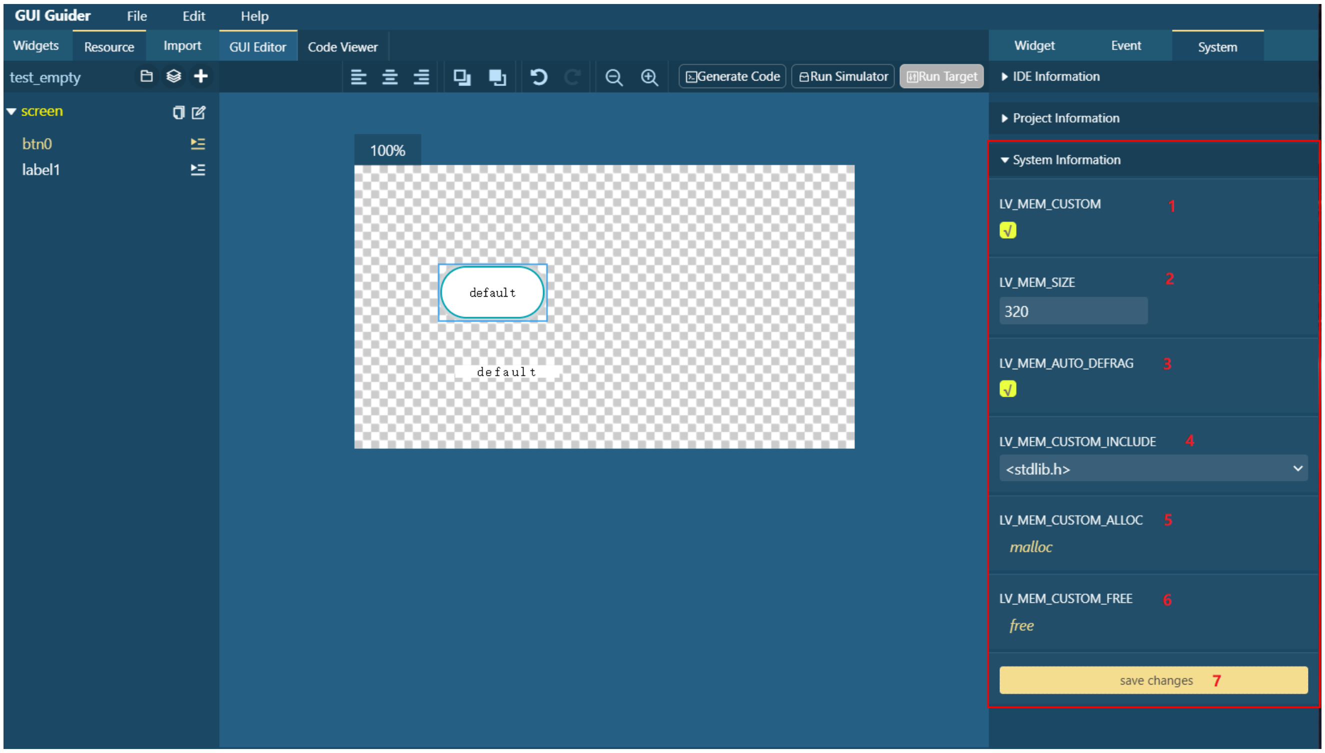
Value Hello GUI

Figure 12. IDE - Widget Event window

Table 10. Description of options in the Widget Event window

Label	Description
1	Set event of the widget.
2	Add event of this widget.
3	Source widget selection (always configured by the widget selected from the drag and drop editor).
4	Select trigger.
5	Target widget (widget to be acted upon).
6	Select action.
7	Select font family.
8	Select font size.
9	Input you want change strings.
10	Delete this event.

3.11 IDE - System settings window



LV_MEM_CUSTOM 1

☑

LV_MEM_SIZE 2

320

LV_MEM_AUTO_DEFRAG 3

☑

LV_MEM_CUSTOM_INCLUDE 4

<stdlib.h>

LV_MEM_CUSTOM_ALLOC 5

malloc

LV_MEM_CUSTOM_FREE 6

free

save changes 7

Figure 13. IDE - System settings window

Table 11. Description of System settings options

Label	Description
1	Use custom malloc/free for memory usage.
2	Set the size of memory allocated for LVGL APP usage.
3	Enable the automatically memory fragments cleanup.
4	Select the header of function for memory allocation and free.
5	Display the memory allocation function.
6	Display the memory free function.
7	Button to save the settings.

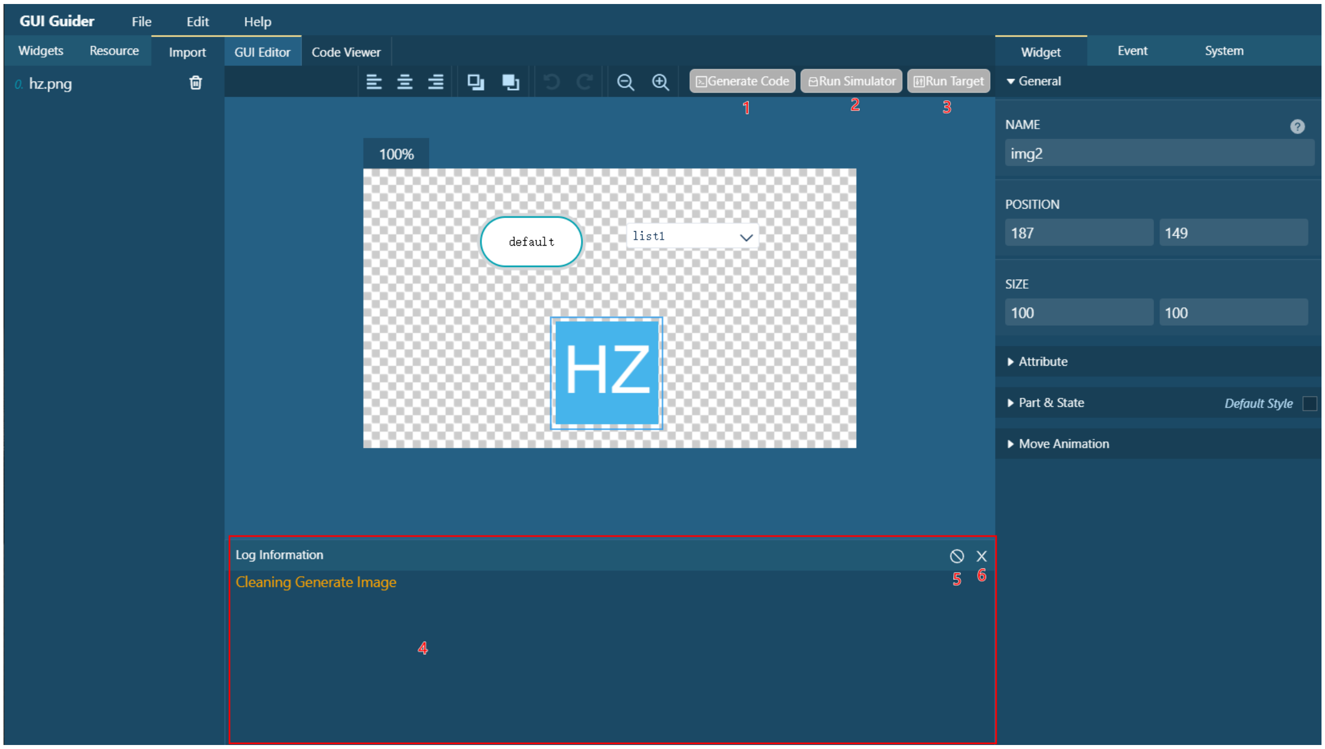
3.12 IDE - Generate LVGL C code

GUI Guider can generate the source code of the GUI APP designed by GUI Guider, including LVGL based C code, JSON file of UI definition, font source file, and image source file.

Table 12 describes the elements in Figure 14.

- **Generate Code:** Generates LVGL-based GUI source files, font files, and image files. Meanwhile, the project is saved.
- **UI source:** Includes *events_init*, *gui_guider*, and *setup_scr_**, both c files and h files. All the files are stored in *$(project dir)/generated*.
- **Font files:** Includes *lv_font_$(font name)_$(size).c* and *guider_fonts.h*. These files are stored in *$(project dir)/generated/guider_fonts*.

- **Images files:** Includes *_$(image name)_$(size).c*. All the files are stored in *$(project dir)/generated/images*.



Cleaning Generate Image

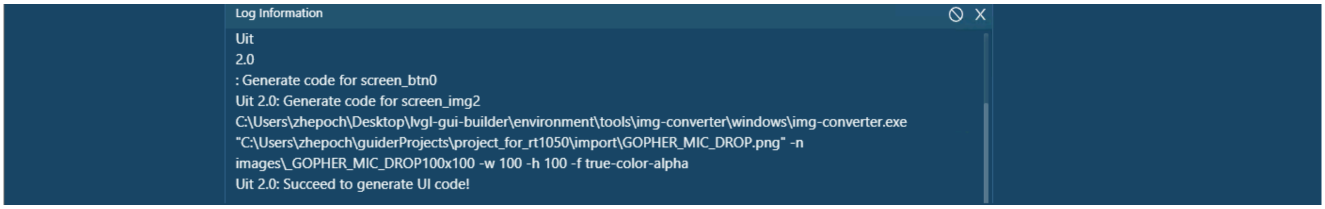
Figure 14. IDE - Generate LVGL C code

Table 12. IDE - Generate LVGL C code

Label	Description
1	Generate LVGL C code of current GUI APP.
2	Run the current GUI APP in simulator.
3	Download the current GUI APP to the target board.
4	Real-time log display.
5	Clear the logs in log window.
6	Close the log window.

NOTE

- When **Generate Code** is clicked, the window changes to modal state.
- When **Generate Code** runs, the real-time log appears in the log window.
- When the code generation completes, the full log appear in the log window.

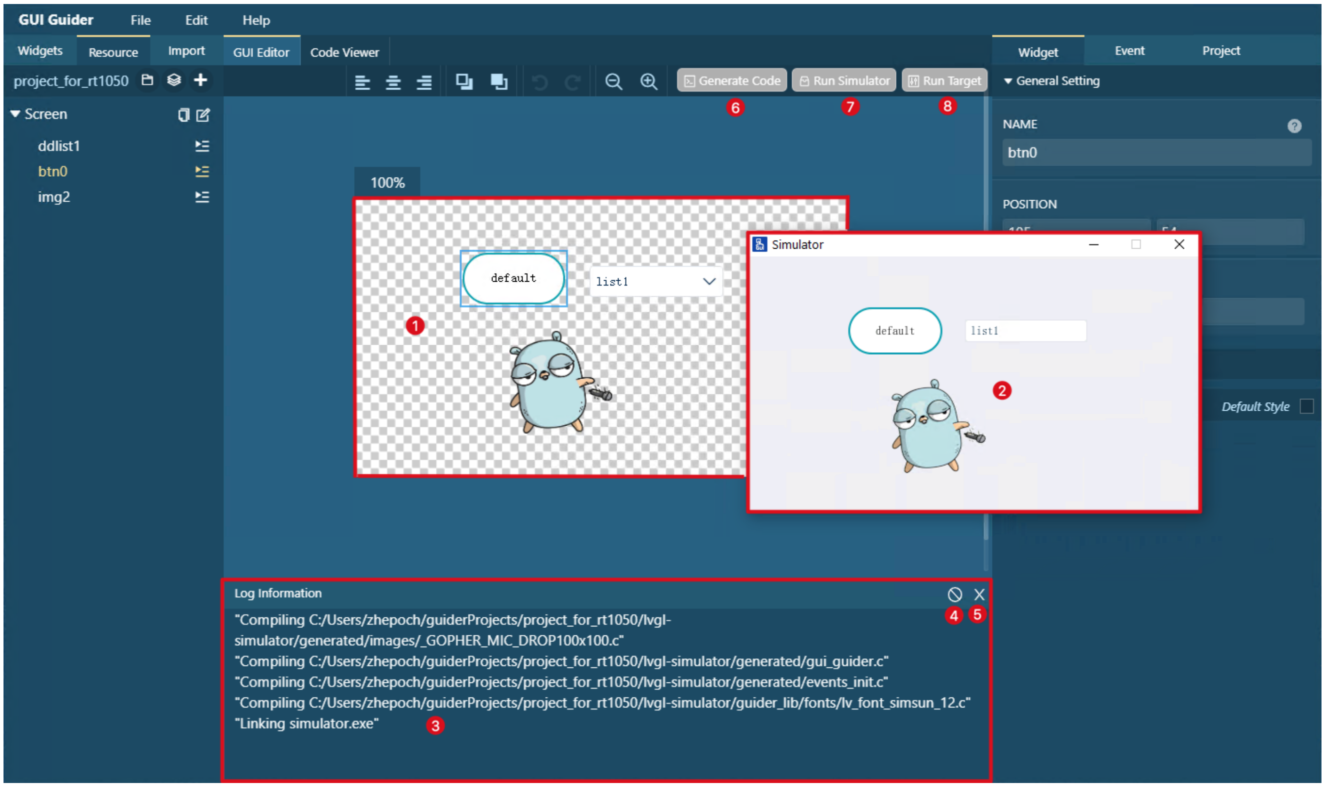


```
Uit
2.0
: Generate code for screen_btn0
Uit 2.0: Generate code for screen_img2
C:\Users\zhepoch\Desktop\lvgl-gui-builder\environment\tools\img-converter\windows\img-converter.exe
"C:\Users\zhepoch\guiderProjects\project_for_rt1050\import\GOPHER_MIC_DROP.png" -n
images\_GOPHER_MIC_DROP100x100 -w 100 -h 100 -f true-color-alpha
Uit 2.0: Succeed to generate UI code!
```

Figure 15. Log information

3.13 IDE - Run Simulator

Table 13 describes elements in the Figure 16.



```
"Compiling C:/Users/zhepoch/guiderProjects/project_for_rt1050/lvgl-
simulator/generated/images/_GOPHER_MIC_DROP100x100.c"
"Compiling C:/Users/zhepoch/guiderProjects/project_for_rt1050/lvgl-simulator/generated/gui_guider.c"
"Compiling C:/Users/zhepoch/guiderProjects/project_for_rt1050/lvgl-simulator/generated/events_init.c"
"Compiling C:/Users/zhepoch/guiderProjects/project_for_rt1050/lvgl-simulator/guider_lib/fonts/lv_font_simsun_12.c"
"Linking simulator.exe"
```

Figure 16. IDE - Run Simulator

Clicking the run simulator button (#7) generates code and launches the simulator in a separate window. Once the simulator is launched, the IDE disables the use of the **Generate Code**, **Run Simulator**, or **Run Target** buttons until the current simulator instantiation is closed. To interact with the simulation, use your PC's mouse and keyboard.

Table 13. IDE - Run Simulator

Label	Description
1	Run target button.
2	The simulator window.
3	The real-time log window.

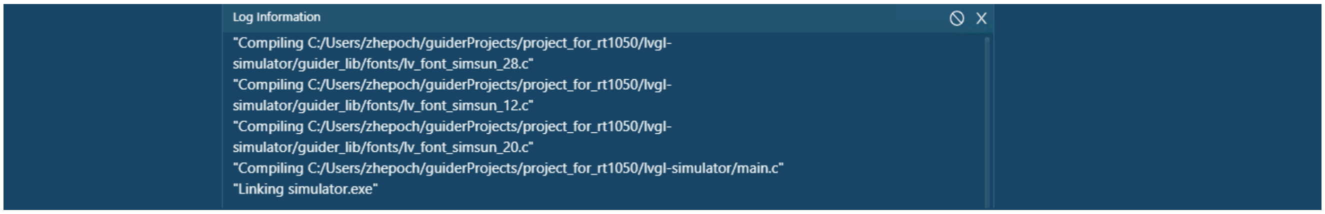
Table continues on the next page...

Table 13. IDE - Run Simulator (continued)

Label	Description
4	Clear logs in the log window.
5	Close the log window.
6	Generate LVGL C code of current GUI APP. More details
7	Run current GUI APP in simulator. More details
8	Run current GUI APP on target board. More details

NOTE

- When **Run Simulator** is clicked, the window changes to modal state.
- When **Run Simulator** runs, the real-time log appears in the log window.
- When the simulator launches, the full log appear in the log window.

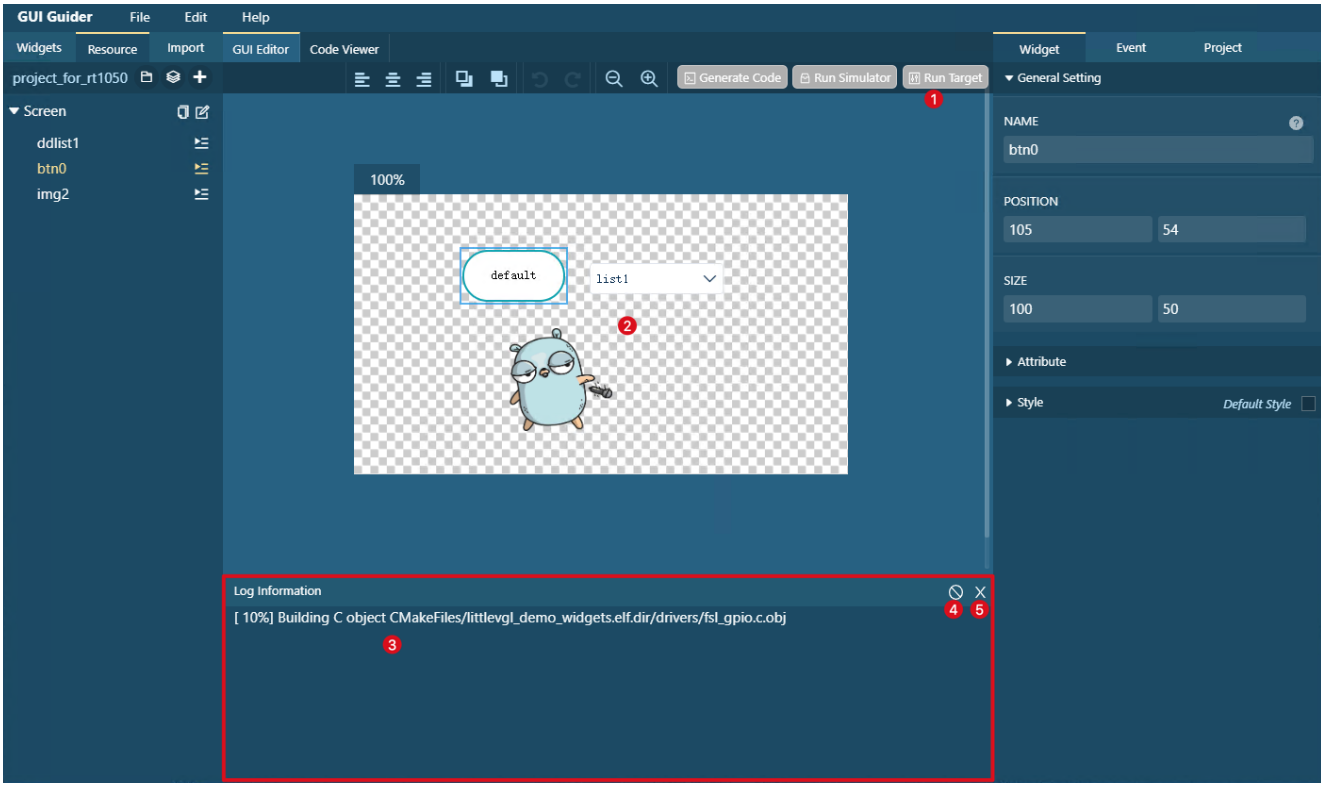


```
"Compiling C:/Users/zhepoch/guiderProjects/project_for_rt1050/lvgl-
simulator/guider_lib/fonts/lv_font_simsun_28.c"
"Compiling C:/Users/zhepoch/guiderProjects/project_for_rt1050/lvgl-
simulator/guider_lib/fonts/lv_font_simsun_12.c"
"Compiling C:/Users/zhepoch/guiderProjects/project_for_rt1050/lvgl-
simulator/guider_lib/fonts/lv_font_simsun_20.c"
"Compiling C:/Users/zhepoch/guiderProjects/project_for_rt1050/lvgl-simulator/main.c"
"Linking simulator.exe"
```

Figure 17. Log information

3.14 IDE - Run target

This GUI Guider release is based on the cross-compiler and image deploy tool from MCUXpreosso 11.3.0. Therefore, you must ensure that MCUXpresso v11.3.0 is installed on the host machine.



NAME btn0

POSITION 105 54

SIZE 100 50

```
[ 10%] Building C object CMakeFiles/littlevgl_demo_widgets.elf.dir/drivers/fsl_gpio.c.obj
```

Figure 18. IDE- Run target window

Table 14. Description of options in the Run target window

Label	Description
1	Run current GUI APP on target board. For details, see IDE - Generate LVGL C code.
2	Drag & drop editor of GUI.
3	Display the real time log.
4	Clear the logs from log window.
5	Close the log window.

NOTE

1. When **Run Target** is clicked, the window changes to modal state.
2. When **Run Target** is initiated, the real time log appears in the log window.
3. When GUI APP is running on the board, full logs appear in the log window.

3.15 IDE - Parent-child

Parent/child definition is supported for tileview, tabview, and container.

NOTE

The instructions are based on container widget, the similar steps are used for tileview and tabview.

To use the function, perform the following steps:

1. Drag the container widget to the UI editor.

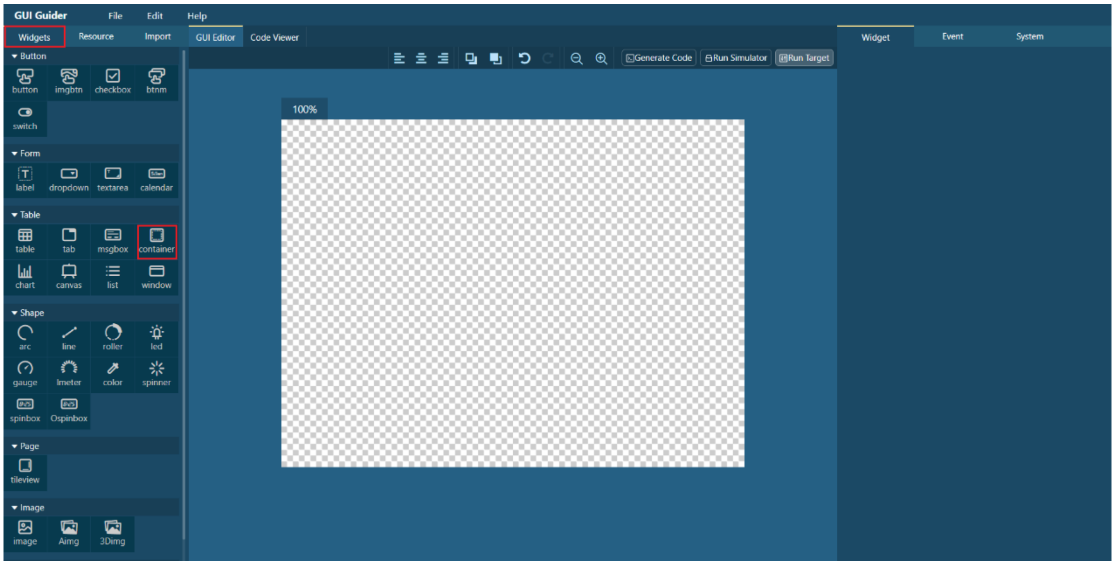
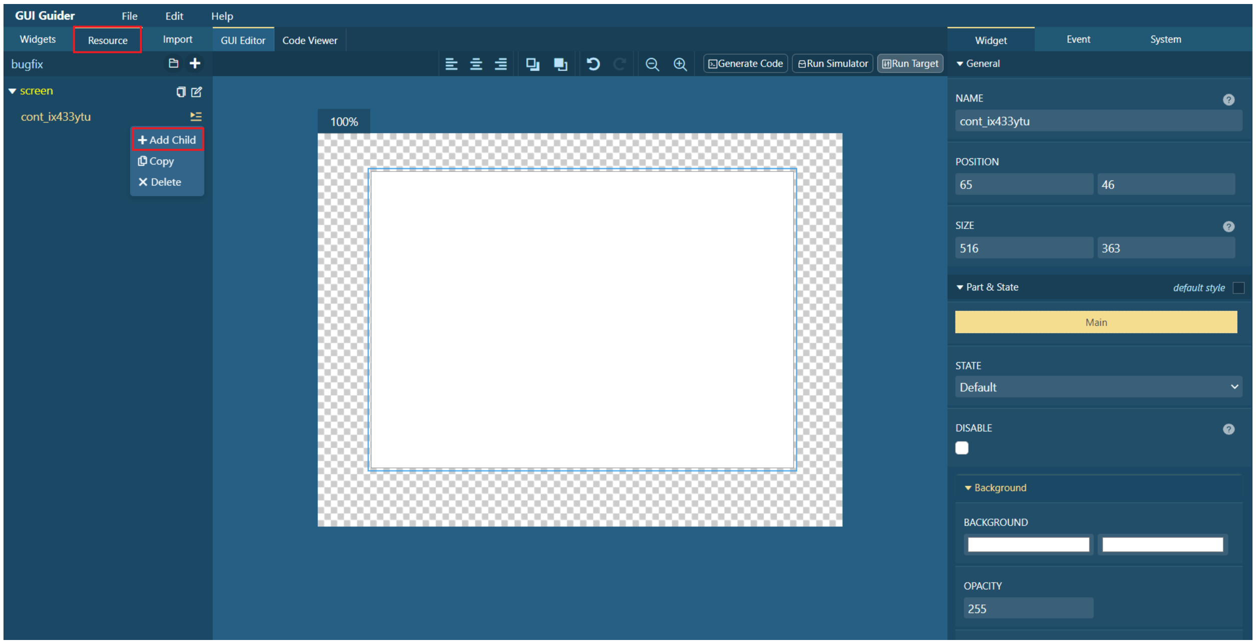


Figure 19. Drag the container widget

2. Select the container widget in **Resource** tab.

3. Click the **+Add Child** menu and add the child widget in the container.



NAME: cont_ix433ytu

POSITION: 65 46

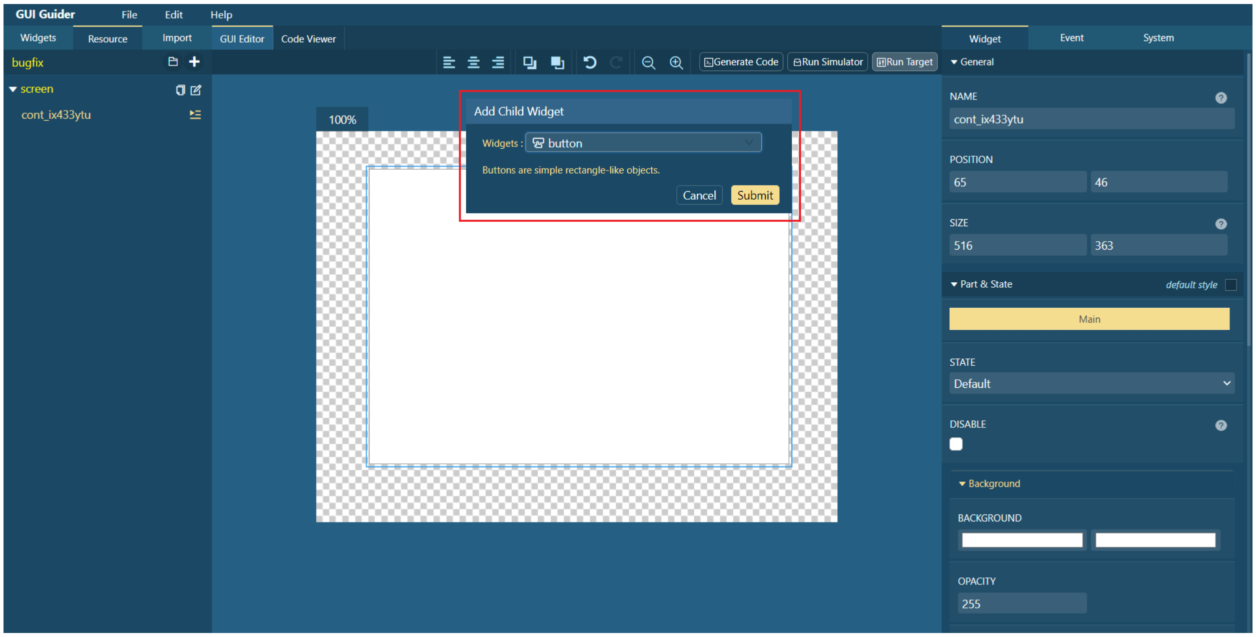
SIZE: 516 363

STATE: Default

OPACITY: 255

Figure 20. Add child

4. After adding the child widget, the parent and child widgets appear in a tree hierarchy in the **Resource** tab.



Add Child Widget

Widgets : button

Buttons are simple rectangle-like objects.

NAME: cont_ix433ytu

POSITION: 65 46

SIZE: 516 363

STATE: Default

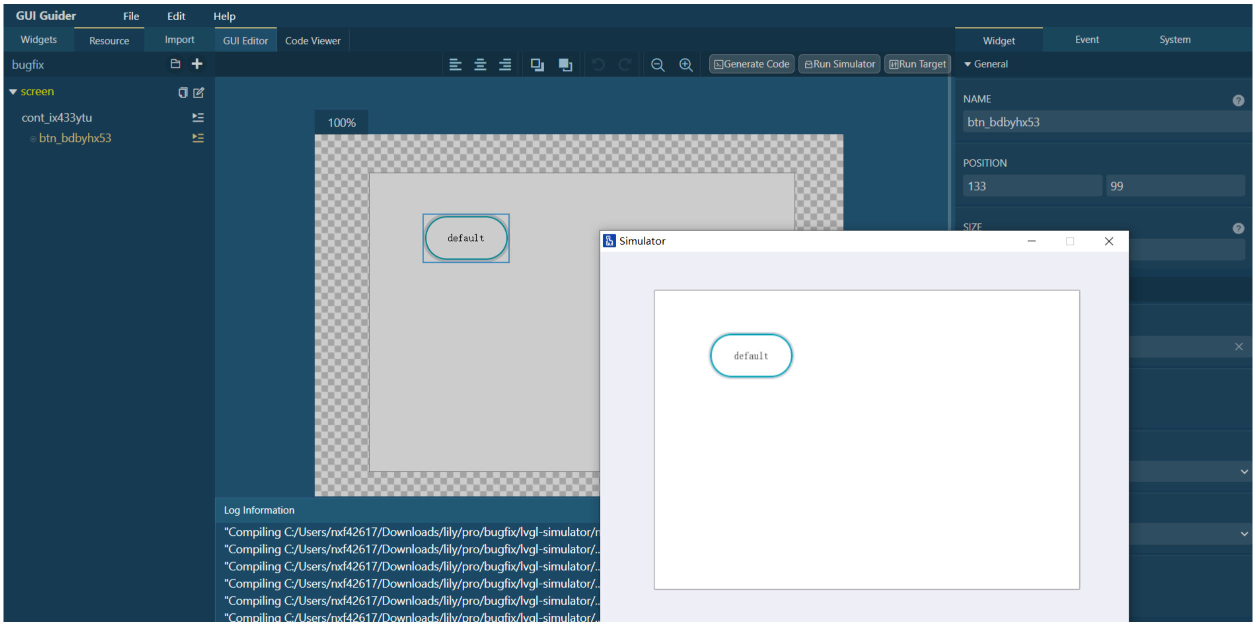
DISABLE

BACKGROUND

OPACITY: 255

Figure 21. Add child widget

5. When the design is done, click **Run Simulator** or **Run Target** to check the UI effect.



NAME: btn_bdbyhx53

POSITION: 133 99

SIZE

Log Information

```
"Compiling C:/Users/nxf42617/Downloads/lily/pro/bugfix/lvgl-simulator/r
"Compiling C:/Users/nxf42617/Downloads/lily/pro/bugfix/lvgl-simulator/..
"Compiling C:/Users/nxf42617/Downloads/lily/pro/bugfix/lvgl-simulator/..
"Compiling C:/Users/nxf42617/Downloads/lily/pro/bugfix/lvgl-simulator/..
"Compiling C:/Users/nxf42617/Downloads/lily/pro/bugfix/lvgl-simulator/..
"Compiling C:/Users/nxf42617/Downloads/lily/pro/bugfix/lvgl-simulator/..
```

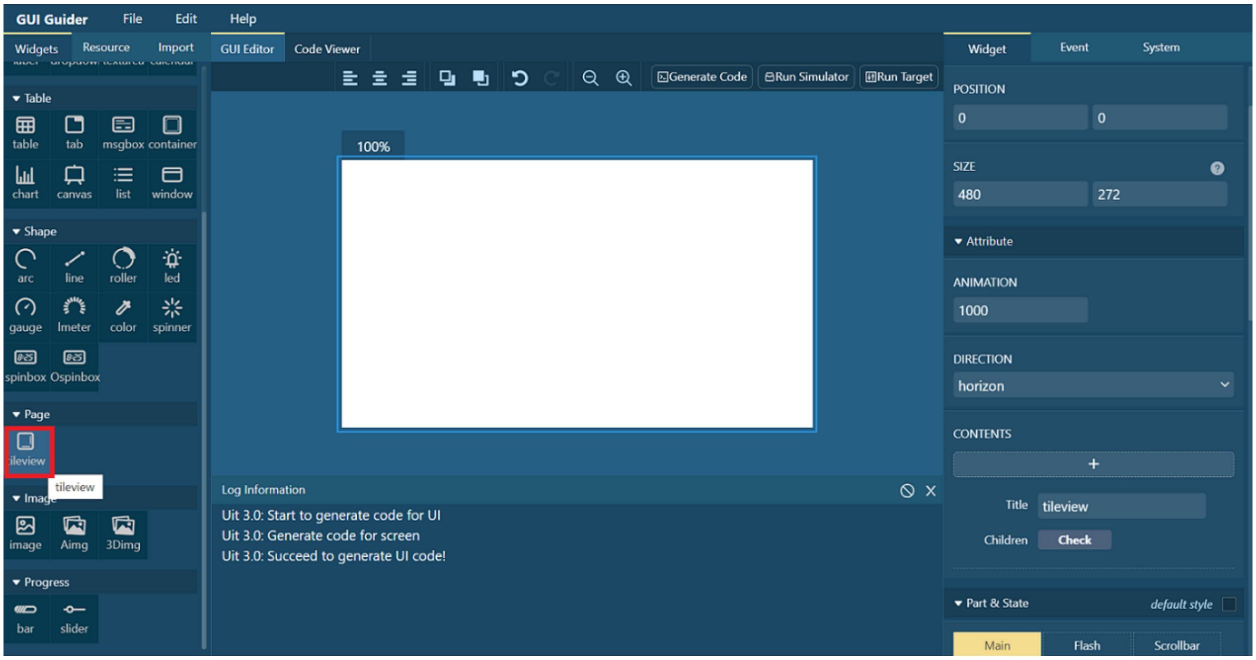
Figure 22. Check UI effect

3.16 IDE - Tileview

Tileview is implemented as a standard widget in GUI Guider and the usage is different from v1.0.0. Tileview is now used as other supported widgets.

The following steps describe the workflow of the Tileview widget.

1. Drag **Tileview** from the **Widgets** window to the UI editor.



POSITION: 0, 0

SIZE: 480, 272

ANIMATION: 1000

DIRECTION: horizon

CONTENTS: +

Title: tileview

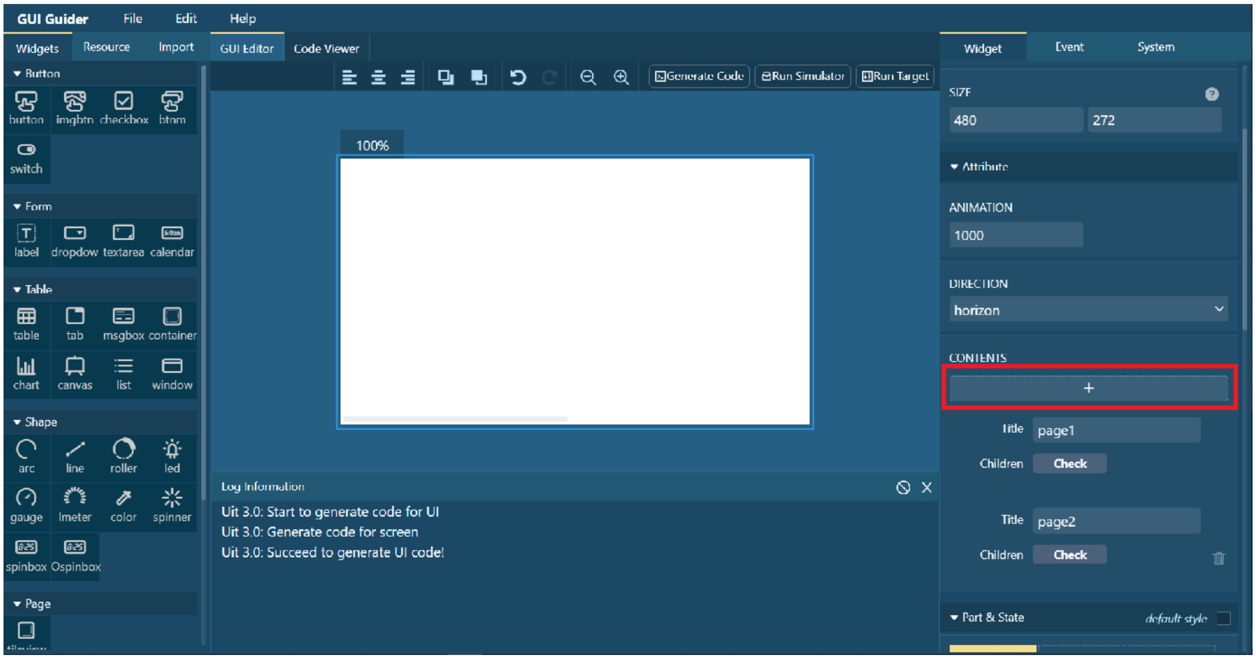
Children: Check

Log Information

```
Uit 3.0: Start to generate code for UI
Uit 3.0: Generate code for screen
Uit 3.0: Succeed to generate UI code!
```

Figure 23. Drag Tileview

2. To create a page for Tileview, click the **"+"** button in attribute window.



SIZE: 480, 272

ANIMATION: 1000

DIRECTION: horizon

CONTENTS: +

Title: page1

Children: Check

Title: page2

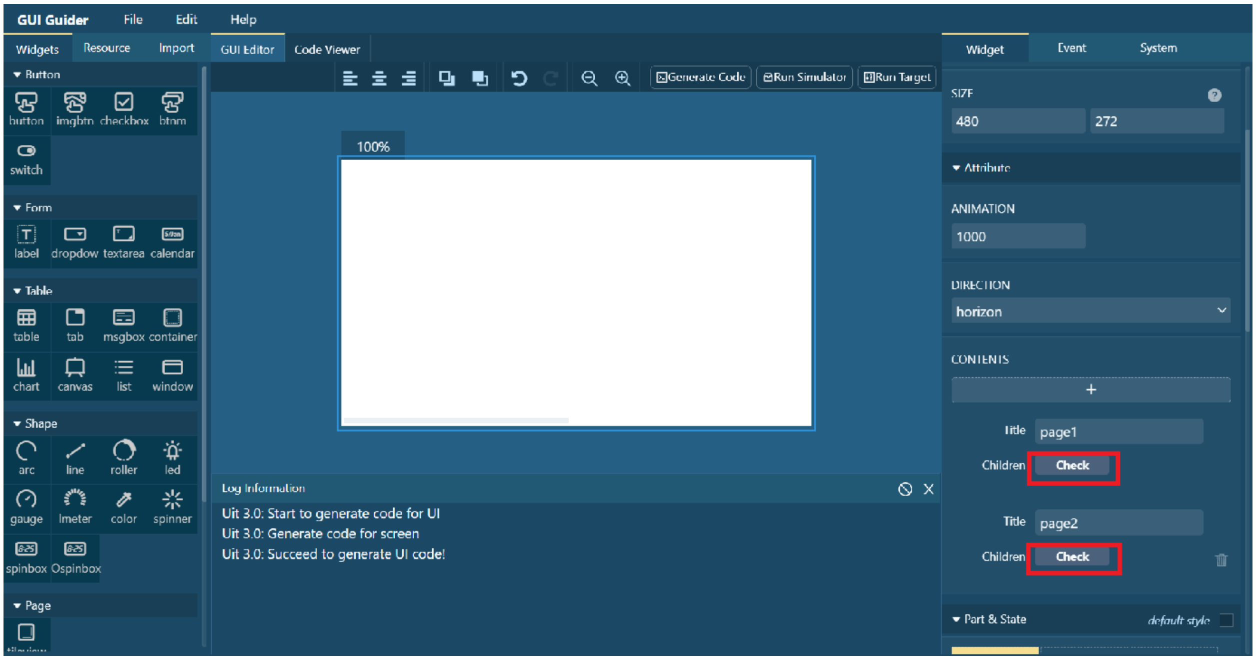
Children: Check

Log Information

```
Uit 3.0: Start to generate code for UI
Uit 3.0: Generate code for screen
Uit 3.0: Succeed to generate UI code!
```

Figure 24. Create page

3. To switch to a page, click the **Check** button of corresponding page.



SIZE 480 272

ANIMATION 1000

DIRECTION horizon

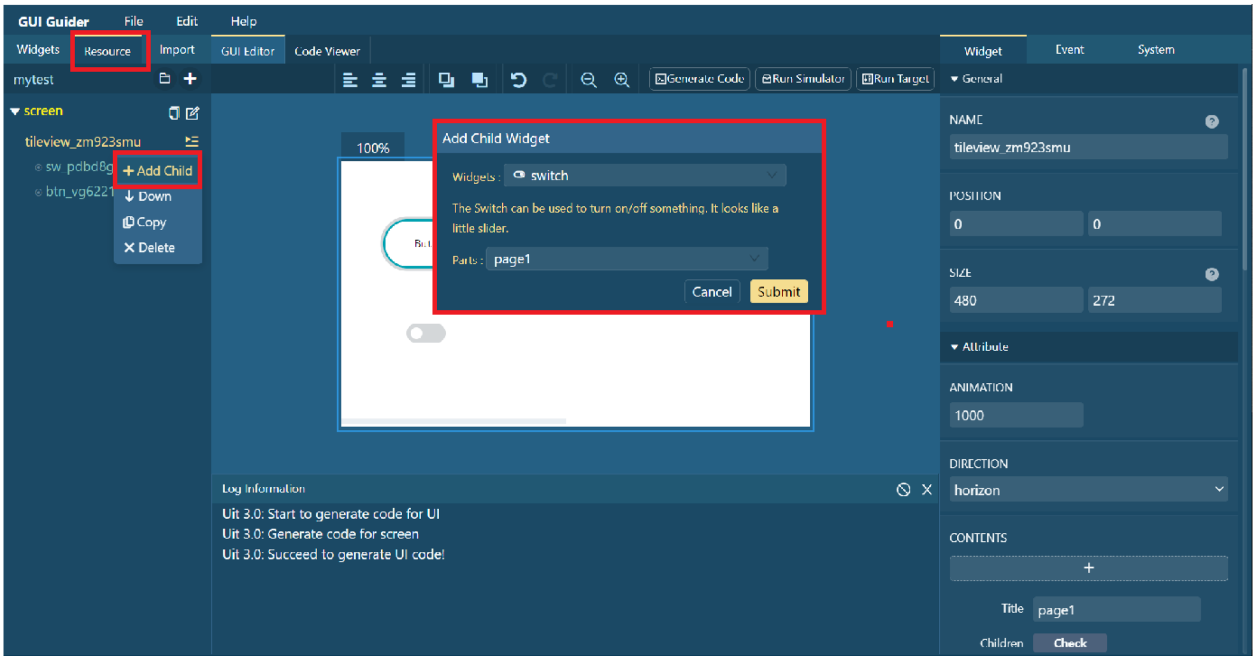
Title page1

Title page2

Uit 3.0: Start to generate code for UI
Uit 3.0: Generate code for screen
Uit 3.0: Succeed to generate UI code!

Figure 25. Switch page

4. To add child widgets in a Tileview page, select the page in **Resource window**. Click **Add Child** in the menu of this page and select the widget you want to add.



Widgets : switch

The Switch can be used to turn on/off something. It looks like a little slider.

Parts : page1

NAME tileview_zm923smu

POSITION 0 0

SIZE 480 272

ANIMATION 1000

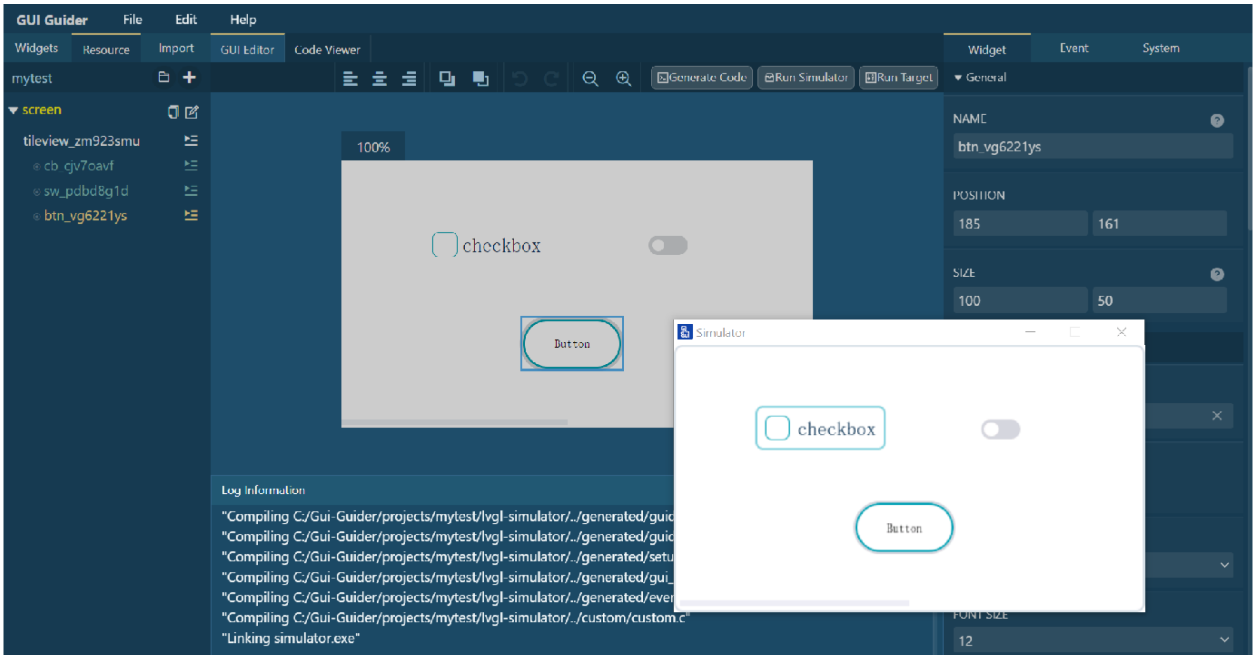
DIRECTION horizon

Title page1

Uit 3.0: Start to generate code for UI
Uit 3.0: Generate code for screen
Uit 3.0: Succeed to generate UI code!

Figure 26. Add child widget

5. When the change is done, click **Run Simulator** or **Run Target** to check the UI design.



NAME: btn_vg6221ys

POSITION: 185 161

SIZE: 100 50

FONT SIZE: 12

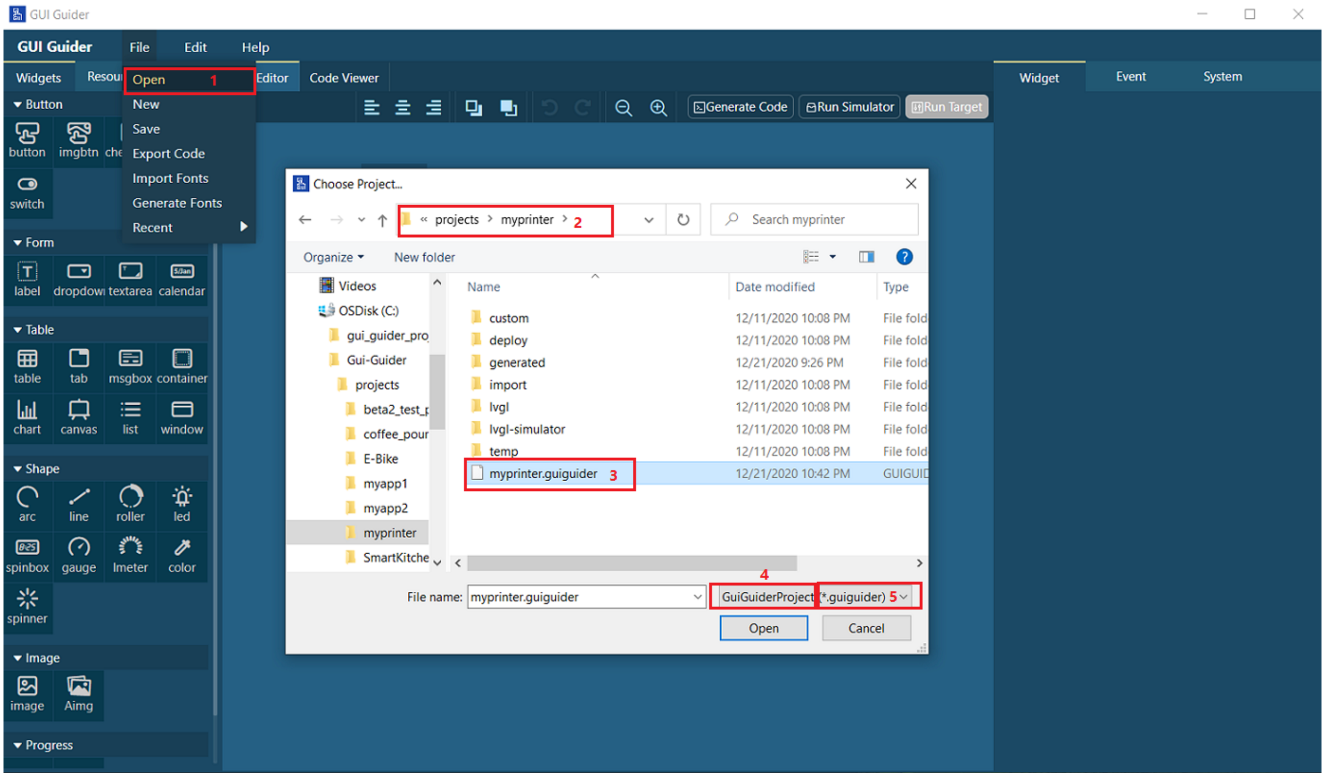
Log Information

```
"Compiling C:/Gui-Guider/projects/mytest/lvgl-simulator/../generated/guic
"Compiling C:/Gui-Guider/projects/mytest/lvgl-simulator/../generated/guic
"Compiling C:/Gui-Guider/projects/mytest/lvgl-simulator/../generated/setu
"Compiling C:/Gui-Guider/projects/mytest/lvgl-simulator/../generated/gui_
"Compiling C:/Gui-Guider/projects/mytest/lvgl-simulator/../generated/ever
"Compiling C:/Gui-Guider/projects/mytest/lvgl-simulator/../custom/custom.c"
"Linking simulator.exe"
```

Figure 27.

3.17 Import project and run simulator

To import the demo project into GUI Guider:



Name	Date modified	Type
custom	12/11/2020 10:08 PM	File fold
deploy	12/11/2020 10:08 PM	File fold
generated	12/21/2020 9:26 PM	File fold
import	12/11/2020 10:08 PM	File fold
lvgl	12/11/2020 10:08 PM	File fold
lvgl-simulator	12/11/2020 10:08 PM	File fold
temp	12/11/2020 10:08 PM	File fold
myprinter.guiguider 3	12/21/2020 10:42 PM	GUIGUID

File name: myprinter.guiguider

GuiGuiderProject (*.guiguider) 5

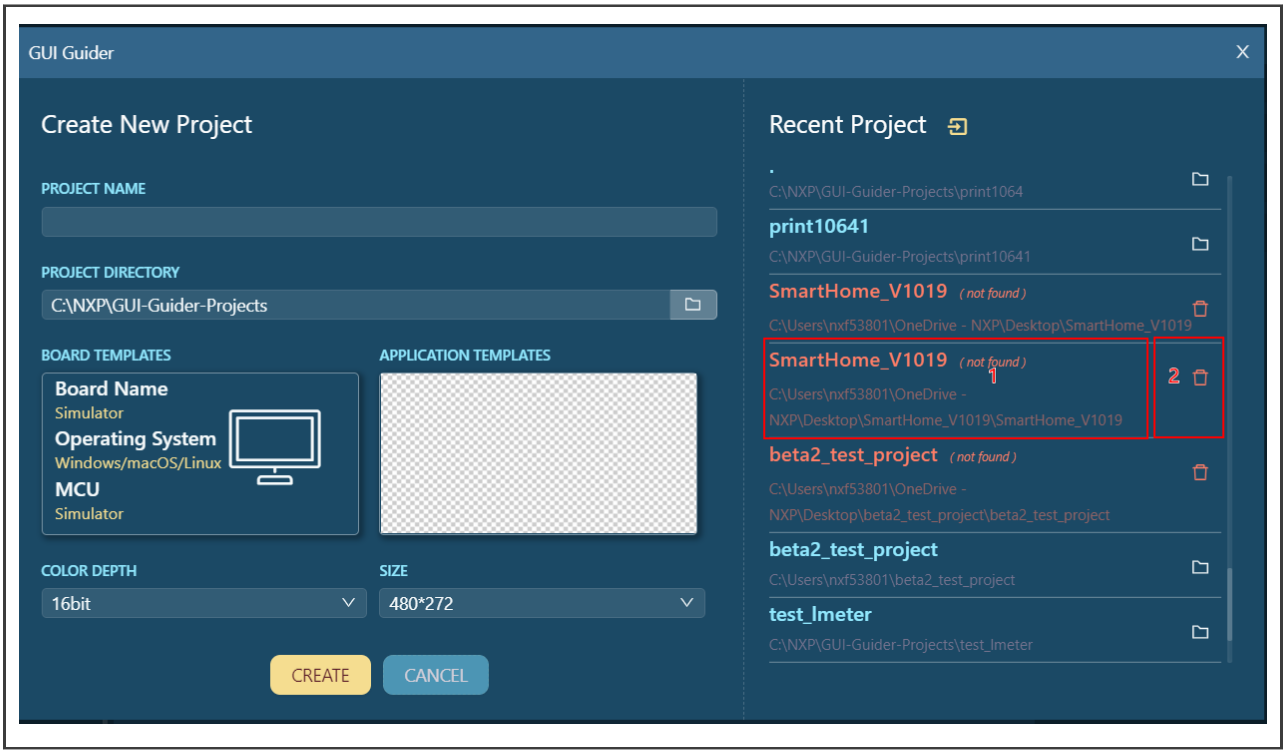
Figure 28. Import project and run simulator

Click **Run Simulator** on GUI Guider and run current GUI APP on simulator.

3.18 Delete project from GUI Guider

To delete the project from GUI Guider, perform the following steps.

1. Close the **GUI Guider IDE** window.
2. Delete the project folder from host file system if not needed, or backup the project to another location for future usage.
3. Open GUI Guider IDE and click the delete icon to delete the corresponding project.



PROJECT NAME

PROJECT DIRECTORY: C:\NXP\GUI-Guider-Projects

BOARD TEMPLATES: Board Name Simulator; Operating System Windows/macOS/Linux; MCU Simulator

APPLICATION TEMPLATES

COLOR DEPTH: 16bit

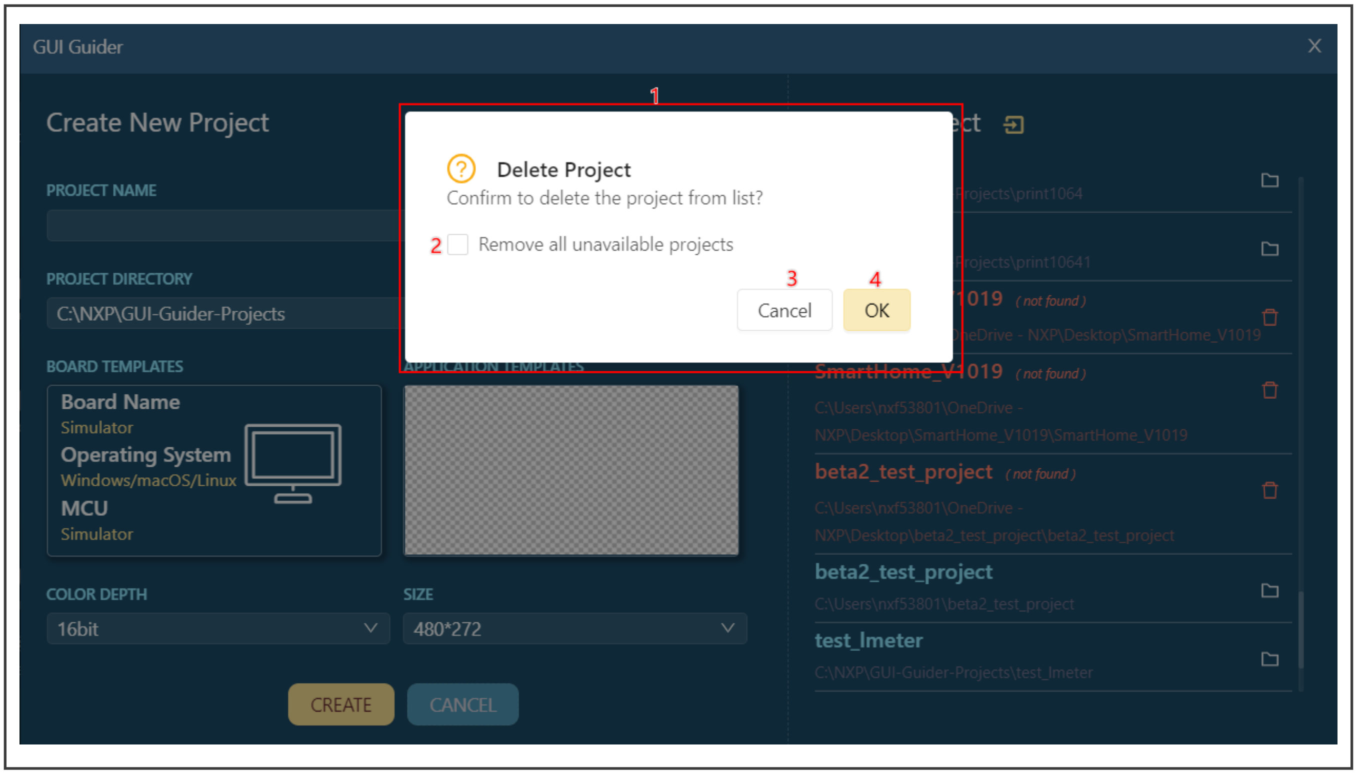
SIZE: 480*272

Recent Project:

- . — C:\NXP\GUI-Guider-Projects\print1064
- print10641 — C:\NXP\GUI-Guider-Projects\print10641
- SmartHome_V1019 (not found) — C:\Users\nxf53801\OneDrive - NXP\Desktop\SmartHome_V1019
- SmartHome_V1019 (not found) — C:\Users\nxf53801\OneDrive - NXP\Desktop\SmartHome_V1019\SmartHome_V1019
- beta2_test_project (not found) — C:\Users\nxf53801\OneDrive - NXP\Desktop\beta2_test_project\beta2_test_project
- beta2_test_project — C:\Users\nxf53801\beta2_test_project
- test_lmeter — C:\NXP\GUI-Guider-Projects\test_lmeter

Table 15. Delete a project

Label.	Description
1	GUI Guider prompts that the project does not exist. NOTE Projects that cannot be found in the file system are highlighted by red text.
2	Click the trash bin icon to delete the project from project list.



Delete Project

Confirm to delete the project from list?

2 □ Remove all unavailable projects

GUI Guider
1
3
4
Cancel
OK
Create New Project
PROJECT NAME
PROJECT DIRECTORY
C:\NXP\GUI-Guider-Projects
BOARD TEMPLATES
Board Name
Simulator
Operating System
Windows/macOS/Linux
MCU
Simulator
APPLICATION TEMPLATES
COLOR DEPTH
16bit
SIZE
480*272
CREATE
CANCEL
ct
rojects\print1064
rojects\print10641
1019 (not found)
neDrive - NXP\Desktop\SmartHome_V1019
SmartHome_V1019 (not found)
C:\Users\nxf53801\OneDrive - NXP\Desktop\SmartHome_V1019\SmartHome_V1019
beta2_test_project (not found)
C:\Users\nxf53801\OneDrive - NXP\Desktop\beta2_test_project\beta2_test_project
beta2_test_project
C:\Users\nxf53801\beta2_test_project
test_lmeter
C:\NXP\GUI-Guider-Projects\test_lmeter

Table 16. Confirm deletion

Label	Description
1	Dialog box to confirm the deletion of the project.
2	Remove all unavailable projects.
3	Cancel the project deletion.
4	Confirm to delete projects.

Chapter 4
Widget details

This section describes the GUI Guider widgets.

4.1 3Dimg

3D animation widget can rotate a given image along with X-axis, Y-axis, Z-axis or combined.

Parameter

- **General Setting**
  - name
  - position: x y
  - size: height width
- **Attribute**
  - Image path
  - interval && repeat count
  - Axisx
  - Axisy
  - Axisz
  - Frame_number

4.2 Animation Image

The animation image widget supports to produce animation based on images series or GIF image.

Parameter

- **General Setting**
  - name
  - position: x y
  - size: height width
- **Attribute**
  - play back
  - interval && repeat count
  - enable start callback function
  - enable ready callback function
  - images

NOTE

The callback function name must begin with a letter or an underscore, must be at least three characters long，can only include letter, number, and underscore.

NOTE

Limitation: The animation image does not support lpc54628 and lpc54s018.

4.3 Arc

The arc consists of a background and a foreground arc. Both can have start and end angles and thickness.

Parameter

- **General Setting**
  - name
  - position: x y
  - size: height width
- **Attribute**
  - Angle
  - Back angle
  - Line cap
  - Padding left & padding right (max:20)
  - Padding top & padding bottom (max:20)
- **Part & State**
  - **Background Style**
    - state

      1. Default 2. Focused 3. Disabled
    - disable
    - background
    - opacity (max:255)
    - background gradient direction
    - Line color
  - **Active Style**
    - state

      1. Default 2. Focused 3. Disabled
    - disable
    - Line color
    - Line width

4.4 Bar

The bar object has a background and an indicator on it. The width of the indicator is set according to the current value of the bar.

Vertical bars can be created if the width of the object is smaller than its height.

Not only end, but the start value of the bar can be set which changes the start position of the indicator.

Parameter

- **General Setting**
  - name
  - position: x y
  - size: height width

- Attribute
    - animtime
    - bar value (max:100)
    - animation
- Part & State
    - Background Style
        - state
            1. Default
        - disable
        - background
        - opacity
        - background gradient direction
        - radius
        - Padding top & bottom (max:20)
        - Padding left & right (max:20)
    - Active Style
        - state
            1. Default
        - disable
        - background
        - opacity (max:255)
        - background gradient direction
        - radius

4.5 Button

The buttons are simple rectangle-like objects. They are derived from Containers so layout and fit are also available. They can be enabled to automatically transition to checked state on click.

Parameter

- Attribute
    - font family
    - font size
    - text (max-length:32)
    - text color
    - toggle
- General Setting
    - name
    - position: x y
    - size: height width
- Part & State

- **Main Style**
  - background
  - background gradient direction
  - border color
  - border opacity (max:255)
  - border width (max:5)
  - disable
  - opacity (max:255)
  - outline color
  - outline opacity
  - radius (max:200)
  - state

    1. Default

    2. Focused

    3. Pressed

    4.Checked

    5.Disabled

4.6 Button matrix

The button matrix objects can display multiple buttons in rows and columns. The main reasons for wanting to use a button matrix instead of a container and individual button objects are:

- The button matrix is simpler to use for grid-based button layouts.
- The button matrix consumes a lot less memory per button.

Parameters

- **General Setting**
  - name
  - position: x y
  - size: height width
- **Attribute**
  - line item
- **Part & State**
  - **Main Style**
    - state

      1. Default 2. Focused 3. Disabled
    - disable
    - background
    - opacity (max:255)
    - background gradient direction
    - border color

        - ◦ border width (max:1)
        - ◦ radius (max:8)
        - ◦ Padding top & bottom(max:20)
        - ◦ Padding left & right (max:20)
        - ◦ Padding inner
    - — **Button Style**
        - ◦ state

          1. Default 2.Pressed
        - ◦ disable
        - ◦ text color
        - ◦ font family
        - ◦ font size
        - ◦ background
        - ◦ opacity (max:255)
        - ◦ background gradient direction
        - ◦ border color
        - ◦ border width (max:1)
        - ◦ radius (max:8)

4.7 Calendar

The calendar object is a classic calendar which can:

- highlight the current day
- highlight any user-defined dates
- display the name of the days
- go the next/previous month by button click
- highlight the clicked day

Parameter

- **General Setting**
    - — name
    - — position: x y
    - — size: height width
- **Part & State**
    - — **Background Style**
        - ◦ state

          1. Default 2. Focused 3. Disabled
        - ◦ disable
        - ◦ background color
        - ◦ opacity (max:255)

    - background gradient direction
    - border color
    - border width (max:4)
  - **Head Style**
    - state

      1. Default
    - disable
    - text color
    - font family
    - font size
    - letter spacing (max:5)
  - **Day Style**
    - state

      1. Default
    - disable
    - text color
    - font family
    - font size
  - **Date Style**
    - state

      1. Default 2. Focused 3. Checked 4. Pressed
    - disable
    - text color
    - font family
    - font size
    - Background color
    - opacity (max:255)
    - Background gradient direction (horizontal, vertical)

4.8 Canvas

A canvas inherits from Image where the user can draw anything. Rectangles, texts, images, lines arcs can be drawn here using lvgl's drawing engine. Besides some "effects" can be applied as well like rotation, zoom and blur.

Parameter

- **General Setting**
  - name
  - position: x y
  - size: height width
- **Part & State**
  - **Main Style**

    - state

      1. Default

    - disable
    - background
    - opacity (max:255)

4.9 Chart

The charts are a basic object to visualize data points. They support *Line* charts (connect points with lines and/or draw points on them) and *Column* charts.

Charts also support division lines, 2 y axis, axis ticks, and texts on ticks. Currently the number of chart points must be 10.

Parameter

- **General Setting**
  - name
  - position: x y
  - size: height width
- **Attribute**
  - line numbers row & col
  - Y-range (min:10, max:300)
  - chart type
  - chart data
    - chart color
    - chart points
- **Part & State**
  - **Main Style**
    - state

      1. Default 2. Focused

    - disable
    - background
    - opacity (max:255)
    - background gradient direction
    - Padding top & bottom (max:20)
    - Padding left & right (max:20)
  - **Line Style**
    - state

      1. Default

    - disable
    - line color
    - line opacity (max:255)
    - line width (max:4)

4.10 Checkbox

The checkbox objects are built from a Button background which contains an also Button *bullet* and a Label to realize a classical checkbox.

Parameter

- General Setting
  - name
  - position: x y
- Attribute
  - text
- Part & State
  - Main Style
    - state

      1. Default 2. Focused 3. Disabled
    - disable
    - text color
    - font family
    - font size
    - letter spacing
    - background
    - opacity (max:255)
    - background gradient direction
    - radius
  - Bullet Style
    - state

      1. Default 2. Focused 3. Pressed 4.Checked 5.Disabled 6.Checked and Pressed
    - disable
    - background
    - opacity (max:255)
    - background gradient direction
    - border color
    - border width (max:4)
    - radius

4.11 Color

As its name implies, *Color picker* allows to select color. The Hue, Saturation and Value of the color can be selected after each other. The widget has two forms: circle (disc) and rectangle. In both forms, be long pressing the object, the color picker will change to the next parameter of the color (hue, saturation or value). Besides, double click will reset the current parameter.

Parameter

- General Setting

- — name
- — position: x y
- — size: height width

- **Attribute**
  - — color picker type

    1.Circle 2. Rectangle
- **Part & State**
  - — **Main Style**
    - ◦ state

      1. Default
    - ◦ disable
    - ◦ scale width (max:50)
    - ◦ padding

4.12 Container

The containers are essentially a basic object with layout and automatic sizing features features.

Parameter

- **General Setting**
  - — name
  - — position: x y
  - — size: height width
- **Part & State**
  - — **Main Style**
    - ◦ state

      1. Default
    - ◦ disable
    - ◦ background
    - ◦ opacity (max:255)
    - ◦ background gradient direction
    - ◦ border color
    - ◦ border opacity (max:255)
    - ◦ border width (max:10)
    - ◦ radius (max:200)

4.13 Drop-down list

The drop-down list allows the user to select one value from a list.

The drop-down list is closed by default and displays a single value or a predefined text. When activated (by click on the drop-down list), a list is created from which the user may select one option. When the user selects a new value, the list is deleted.

Parameter

- General Setting
  - name
  - position: x y
  - size: width
- Attribute
  - list height
  - draw arrow
  - list
- Part & State
  - Main Style
    - state

      1. Default 2. Focused 3. Disabled
    - disable
    - text color
    - font family
    - font size
    - background
    - opacity (max:255)
    - background gradient direction
    - border color
    - border width (max:4)
    - radius
  - Selected Style
    - state

      1. Default
    - disable
    - text color
    - font family
    - font size
    - background
    - opacity (max:255)
    - background gradient direction
    - border color
    - border width (max:4)
    - radius
  - List Style
    - state

      1. Default

        - disable
        - text color
        - font family
        - font size
        - background
        - opacity (max:255)
        - background gradient direction
        - border color
        - border width (max:4)
        - radius

4.14 Gauge

The gauge is a meter with scale labels and one or more needles.

Parameter

- **General Setting**
    - name
    - position: x y
    - size: height width
- **Attribute**
    - major (min:1)
    - minor (min:1)
    - angle
    - critical
    - min
    - max
    - image type
    - needles (min:1)

      1. needle_color 2. needle_value
- **Part & State**
    - **Main Style**
        - state

          1. Default 2. Focused
        - disable
        - color
        - font family
        - font size
        - letter space
        - background

    - opacity (max:255)
    - background gradient direction
    - padding
    - line color
    - line opacity (max:255)
    - line width (max:20)
    - scale width
    - scale grad color
    - scale end color
    - scale border width (max:30)
    - scale end border width (max:30)
    - scale end line width (max:6)
  - Major Style
    - state

      1. Default 2. Focused
    - disable
    - line color
    - line opacity
    - line width (max:20)
    - scale width
    - scale grad color
    - scale end color
    - scale end line width
  - Needle Style
    - state

      1. Default 2. Focused
    - disable
    - background
    - opacity (max:255)
    - background gradient direction
    - padding
    - line opacity
    - line width
    - size

4.15 Image

The images are the basic object to display from the flash (as arrays) or externally as files. Images can display symbols (LV_SYMBOL_...) too.

Using the Image decoder interface custom image formats can be supported as well.

Parameter

- **General Setting**
  - name
  - position: x y
  - size: height width
- **Attribute**
  - image path
  - color format
  - rotate center (max:400)
  - rotate angle
- **Part & State**
  - **Main Style**
    - state
      1. Default
    - disable
    - opacity (max:255)
    - filter color
    - filter opacity (max:255)

4.16 Image Button

The image button is very similar to the simple 'Button' object. The only difference is that it displays user-defined images in each state instead of drawing a rectangle. See the Button section for details, before reading this section,

Parameter

- **General Setting**
  - name
  - position: x y
  - size: height width
- **Attribute**
  - text
  - toggle
  - color format
  - released picture
  - pressed picture
  - checked released picture
  - checked pressed picture
- **Part & State**
  - **Main Style**
    - state
      1. Default 2. Pressed 3.Checked

- disable
- text color
- font family
- font size
- opacity (max:255)
- filter color (image recolor)
- filter opacity (image recolor opacity, max:255)

4.17 Label

A label is the basic object type that is used to display text.

Parameter

- **General Setting**
  - name
  - position: x y
  - size: width
- **Attribute**
  - text
  - text align
- **Part & State**
  - **Main Style**
    - state

      1. Default 2. Disabled
    - disable
    - text color
    - font family
    - font size
    - letter spacing (max:10)
    - background
    - opacity (max:255)
    - background gradient direction
    - padding top & bottom
    - padding right & left (max:10)
    - radius (max:100)

4.18 LED

The LEDs are rectangle-like (or circle) object. It's brightness can be adjusted. With lower brightness the the colors of the LED become darker.

Parameter

- **General Setting**

- name
- position: x y
- size: height width

- **Attribute**
  - bright (max:255)
- **Part & Style**
  - Main Style
    - state

      1. Default 2. Focused 3. Disabled
    - disable
    - background
    - opacity
    - direction

      1. Normal 2. Vertical 3. horiontal
    - border color
    - border opacity
    - border width (max:3)
    - radius
    - shadow color
    - shadow width (max:30)

4.19 Line

The line object is capable of drawing straight lines between a set of points.

Parameter

- **General Setting**
  - name
  - position: x y
  - size: height width
- **Attribute**
  - line start
  - line end
- **Part & State**
  - **Main Style**
    - state

      1. Default
    - disable
    - line color
    - line width (max:50, min:1)

4.20 Line Meter

The line meter object consists of some radial lines which draw a scale. Setting a value for the Line meter will change the color of the scale lines proportionally.

Parameter

- **General Setting**
  - — name
  - — position: x y
  - — size: height width
- **Attribute**
  - — min
  - — max
  - — critical
  - — line
  - — angle
  - — offset
- **Part & State**
  - — **Main Style**
    - ◦ state

      1. Default 2. Focused 3. Disabled
    - ◦ disable
    - ◦ background
    - ◦ opacity (max:255)
    - ◦ background gradient direction
    - ◦ radius
    - ◦ Padding top & bottom (max:20)
    - ◦ Padding left & right (max:20)
    - ◦ line color
    - ◦ line opacity (max:255)
    - ◦ line width
    - ◦ scale color
    - ◦ scale width

4.21 List

The lists are built from a background Page and Buttons on it. The Buttons contain an optional icon-like Image (which can be a symbol too) and a Label. When the list becomes long enough it can be scrolled.

Parameter

- **General Setting**
  - — name
  - — position: x y

  - size: height width
- Attribute
  - animation
  - items
    - symbol
    - image path
    - image size (e.g:'20,20')
    - text
- Part & State
  - Main Style
    - state

      1. Default 2. Focused 3. Disabled
    - disable
    - border color
    - border width (max:4)
    - radius (max:10)
    - Padding top (max:20)
    - Padding left & right (max:20)
  - Scroll Style
    - state
    - disable
    - background
    - opacity (max:255)
    - background gradient direction
    - radius (max:10)
  - List Style
    - state
    - disable
    - text color
    - font family
    - font size
    - background
    - opacity (max:255)
    - background gradient direction
    - radius (max:10)

4.22 MessageBox

The message boxes act as pop-ups. They are built from a background Container, a Label and a Button matrix for buttons.

The text will be broken into multiple lines automatically (has LV_LABEL_LONG_MODE_BREAK) and the height will be set automatically to involve the text and the buttons (LV_FIT_TIGHT fit vertically).

Parameter

- **General Setting**
  - name
  - position: x y
  - size: height width
- **Attribute**
  - button text
  - mbox text
- **Part & State**
  - **Main Style**
    - state
      1. Default
    - text color
    - font family
    - font size
    - background
    - opacity (max:255)
    - background gradient direction
    - border color
    - border width (max:4)
    - radius (max:20)
  - **Button Bar Style**
    - state
      1. Default
    - background
    - opacity (max:255)
    - background gradient direction
    - border color
    - border width (max:4)
    - radius (max:20)
  - **Button Style**
    - state
      1. Default
    - text color
    - font family
    - font size

        - background
        - opacity (max:255)
        - background gradient direction
        - border color
        - border width (max:4)
        - radius (max:20)

4.23 Ospinbox

The Ospinbox contains a number as text which can be increased or decreased by API functions. Under the hood the Ospinbox is a modified Text area.

Parameter

- General Setting
    - name
    - position: x y
    - size: width (If there are no options, it means this widget cannot set size)
- Attribute
    - integer & decimal
- Part & State
    - Background Style
        - state

          1. Default
        - disable
        - color
        - font size
        - Font family
        - background
        - opacity (max:255)
        - background gradient direction
        - border color
        - border width (max:5)
        - padding (max:20, min:5)
        - radius (max:200)
    - Cursor Style
        - state

          1. Default
        - disable
        - color
        - font size
        - Font family

    - background
    - opacity (max:255)
    - background gradient direction

4.24 Roller

The roller allows you to simply select one option from more with scrolling.

Parameter

- **General Setting**
  - name
  - position: x y
  - size: (If there are no options, it means this widget cannot set size)
- **Attribute**
  - row text (e.g:'1,2,3,4')
  - row count (min:1)
  - mode

    1. Inifiite 2.normal
  - font family
  - font size
- **Part & State**
  - **Background Style**
    - state

      1. Default 2. Focused 3. Disabled
    - disable
    - color
    - background
    - opacity (opacity:255)
    - background gradient direction
    - border color
    - border width (opacity:2)
    - radius
  - **Active Style**
    - state

      1. Default 2. Focused 3. Disabled
    - disable
    - color
    - background
    - opacity (opacity:255)
    - background gradient direction

4.25 Slider

Overview

The slider object looks like a Bar supplemented with a knob. The knob can be dragged to set a value. The Slider also can be vertical or horizontal.

Parameter

- General Setting
  - name
  - position: x y
  - size: height width
- Attribute
  - animtime
  - bar value (max:100)
  - animation
- Part & State
  - Background Style
    - state

      1. Default 2. Focused 3. Disabled
    - disable
    - background
    - Background opacity
    - background gradient direction
    - Border color
    - Border opacity
    - Border radius
    - Padding top & bottom (max:10)
    - Padding left & right (max:10)
  - Active Style
    - state

      1. Default 2. Focused 3. Disabled
    - disable
    - background
    - opacity (max:255)
    - background gradient direction
    - radius
  - Button Style
    - state

      1. Default 2. Focused 3. Disabled
    - disable

- background
- opacity (max:255)
- background gradient direction
- radius

4.26 Spin Box

The spinbox contains a number as text which can be increased or decreased by *Keys* or API functions. Under the hood the spinbox is a modified Text area.

Parameter

- **General Setting**
  - — name
  - — position: x y
  - — size: width (If there are no options, it means this widget cannot set size)
- **Attribute**
  - — integer & decimal
- **Part & State**
  - — **Background Style**
    - state
      1. Default
    - disable
    - color
    - font size
    - background
    - opacity (max:255)
    - background gradient direction
    - border color
    - border width (max:5)
    - padding (max:20, min:5)
    - radius (max:200)
  - — **Cursor Style**
    - state
      1. Default
    - disable
    - font size
    - background
    - opacity (max:255)
    - background gradient direction
  - — **Button Style**
    - state

1.Defaule

- disable
- background
- opacity (max:255)
- background gradient direction
- border color
- border width (max:5)
- radius (max:200)

4.27 Spinner

The spinner object is a spinning arc over a border.

Parameter

- **General Setting**
  - name
  - position: x y
  - size: height width
- **Attribute**
  - angle (max:359)
  - time
  - type

    1. Spinning 2. Fillspin 3. Constant
  - spin direction

    1. Forward 2. Backword
- **Part & State**
  - **Background Style**
    - state

      1. Default
    - disable
    - background
    - opacity
    - background gradient direction
    - Padding top & bottom (max:20)
    - Padding left & right (max:20)
    - line color
    - line width (max:20)
  - **Active Style**
    - state

      1. Default

        - disable
        - line color
        - line width (max:20)

4.28 Switch

The switch can be used to turn on/off something. It looks like a little slider.

Parameter

- **General Setting**
    - name
    - position: x y
    - size: height width
- **Attribute**
    - animtime
- **Part & State**
    - **Background Style**
        - state

          1. Default 2. Focused 3. Disabled
        - disable
        - background
        - opacity (max:255)
        - background gradient direction
        - radius (max:200)
    - **Indic Style**
        - state

          1. Default 2. Focused 3. Disabled
        - disable
        - background
        - opacity (max:255)
        - background gradient direction
        - radius (max:200)
    - **Knob Style**
        - state

          1. Default 2. Focused 3. Disabled
        - disable
        - background
        - opacity (max:255)
        - background gradient direction
        - radius (max:200)

4.29 Table

The tables, as usual, are built from rows, columns, and cells containing texts.

The Table object is very light weighted because only the texts are stored. No real objects are created for cells but they are just drawn on the fly.

Parameter

- General Setting
  - name
  - position: x y
  - size: height width
- Attribute
  - col&row (min:1, max:20)
  - col&row padding
  - col content
  - row content
- Part & State
  - Background Style
    - state

      1. Default 2. Focused 3. Disabled
    - disable
    - text color
    - background
    - opacity (max:255)
    - background gradient direction
    - border color
    - border width (max:2)
    - Padding top & bottom (max:20)
    - Padding left & right (max:20)
  - Cell Style
    - state

      1. Default 2. Focused 3. Disabled
    - disable
    - border color
    - border width (max:2)

4.30 TabView

The tab view object is used to organize content in tabs.

Parameter

- General Setting

- name
- position: x y
- size: height width

- **Attribute**
  - animation (max:3000)
  - contents
- **Part & State**
  - **Main Style**
    - state
      1. Default
    - disable
    - text color
    - font family
    - font size
    - letter spacing (max:6)
    - background
    - opacity (max:255)
    - background gradient direction
    - border color
    - border width (max:5)
  - **Indic Style**
    - state
      1. Default
    - disable
    - background
    - opacity (max:255)
    - background gradient direction
  - **Tab Style**
    - state
      1. Default
    - disable
    - padding (max:20)
    - background
    - opacity (max:255)
    - background gradient direction
    - border color
    - border width
  - **Button Style**

- ◦ state

  1. Default
- ◦ disable
- ◦ text color
- ◦ letter spacing (max:6)

4.31 Text Area

The text area is a Page with a Label and a cursor on it. Texts or characters can be added to it. Long lines are wrapped and when the text becomes long enough the Text area can be scrolled.

Parameter

- General Setting
  - name
  - position: x y
  - size: height width
- Attribute
  - text
  - text align
- Part & State
  - Background Style
    - state

      1. Default 2. Focused
    - disable
    - text color
    - font family
    - font size
    - letter spacing
    - border color
    - border width (max:5)
    - radius (max:200)
  - Scrollable style
    - state

      1. Default 2. Focused
    - disable
    - background
    - opacity (max:255)
    - background gradient direction
    - radius (max:200)

4.32 Tileview

The tileview a container object where its elements (called *tiles*) can be arranged in a grid form. By swiping the user can navigate between the tiles.

Parameter

- General Setting
  - name
  - position: x y
  - size: height width
- Attribute
  - animation (max:3000)
  - direction
  - Contents (min:1)
    - title
    - children
- Part & State
  - Main Style
    - state
      1. Default
    - disable
    - background
    - background gradient direction
    - opacity (max:255)
    - radius
  - Flash Style
    - state
      1. Default
    - disable
    - background
    - background gradient direction
    - opacity (max:255)
    - radius
  - Scrollbar Style
    - state
      1. Default
    - disable
    - background
    - background gradient direction
    - opacity (max:255)

        - radius

4.33 Window

The window is container-like objects built from a header with title and button and a content area.

Parameter

- **General Setting**
    - name
    - position: x y
    - size: height width
- **Attribute**
    - title
    - text
    - BTNS
        - image path
        - image size (e.g:'20,20')
        - symbol
- **Part & State**
    - **Background Style**
        - state
            1. Default
        - disable
        - background
        - gradient
        - opacity (max:255)
        - background gradient direction
        - border color
        - border width (max:12)
    - **Content Style**
        - state
            1. Default
        - disable
        - text color
        - font family
        - font size
        - background
        - gradient
        - opacity (max:255)
        - background gradient direction

- Header Style
    - state

      1. Default
    - disable
    - text color
    - font family
    - font size
    - background
    - gradient
    - opacity (max:255)
    - background gradient direction
- Symbol Style
    - state

      1. Default 2.Pressed
    - disable
    - background
    - opacity (max:255)
    - radius

Chapter 5
Event details

The event details include, event options and support widgets.

- **event options**
  - — Source Widget
  - — Trigger
  - — Target Source
  - — Action
  - — ... (according to the above action)
- **support widgets**
  1. trigger options
     a. Clicked
     b. Pressed
     c. Released
     d. Pressing
     e. Long Pressed
     f. Long pressed Repeat
     g. Short Click
     h. Value changed
  2. action options
     - — Background Color
     - — Background Gradient Color
     - — Background Opacity
     - — Width
     - — Height
     - — Position
     - — Hide
     - — C code
     - — Move Animation
     - — Set Text
     - — Load Screen

Table 17. Widgets: Available triggers and actions

Widget Name	Available Trigger	Available Action
button	a,b,c,d,e,f,g	all
imgbtn (image button)	b, c	all

Table continues on the next page...

Table 17. Widgets: Available triggers and actions (continued)

Widget Name	Available Trigger	Available Action
checkbox	h	all
btnm (button group)	a,b,c,d,e,f,g	all
switch	h	all
dropdown	h	all
calendar	b,c,h	all
msgbox (message box)	h	all
roller	h	all
spinbox	b,h	all
spinner	a,d	all
image	a,b,c,d,e,f,g	all
slider	h	all

Chapter 6
LVGL hardware acceleration

LVGL is a software library that fully implements and customizes a Graphical User Interface (drawing, partial screen refresh, input events, and animations). LVGL has software pixel-based draw engine. Several drawing features in LVGL are performed by hardware (HW) accelerators instead of CPU.

To use the CPU time while HW accelerator is running, an RTOS is required to block the LVGL drawing thread and switch to another task, or idle task, where CPU is suspended to save power. The HW accelerators process pixels faster than CPU resulting in a higher frame rendering rate.

GUI Guider can enable and disable the PXP or VGLite accelerator for the devices that support these features.

NOTE

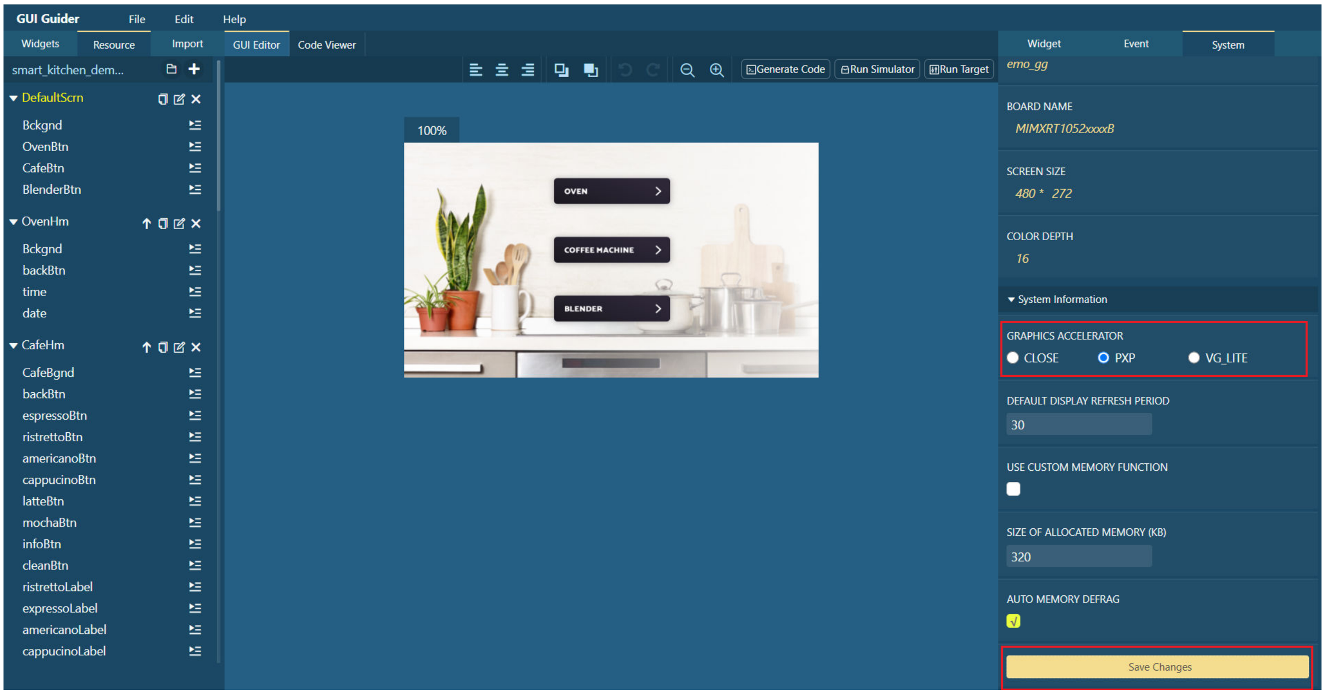
It is possible to manually enable or disable the HW accelerator.

Table 18. LVGL hardware acceleration

Accelerator	i.MX RT1050	i.MX RT1062	i.MX RT1064	i.MX RT1170	i.MX RT595
PXP	√	√	√	√	X
VGLite	X	X	X	√	√

6.1 PXP enablement

Enable the PXP accelerator in GUI Guider.



BOARD NAME
MIMXRT1052xxxxB

SCREEN SIZE
480 * 272

COLOR DEPTH
16

GRAPHICS ACCELERATOR
○ CLOSE ● PXP ○ VG_LITE

DEFAULT DISPLAY REFRESH PERIOD
30

USE CUSTOM MEMORY FUNCTION
□

SIZE OF ALLOCATED MEMORY (KB)
320

AUTO MEMORY DEFRAG
☑

Save Changes

Figure 29. Enable PXP

NOTE

Click the **Save Changes** button after selecting *PXP* as the graphics accelerator.

To enable the PXP accelerator on NXP devices, set the below flag in lv_conf.h. This is required as currently only the color format RGB565 (16 bits) is accelerated on NXP devices.

```
#define LV_COLOR_DEPTH 16
```

PXP is a pixel processing HW engine. To check whether PXP is available on your NXP device, see the Reference Manual document or the board configuration.

To enable PXP in LVGL, set the below flags to 1 in lv_conf.h.

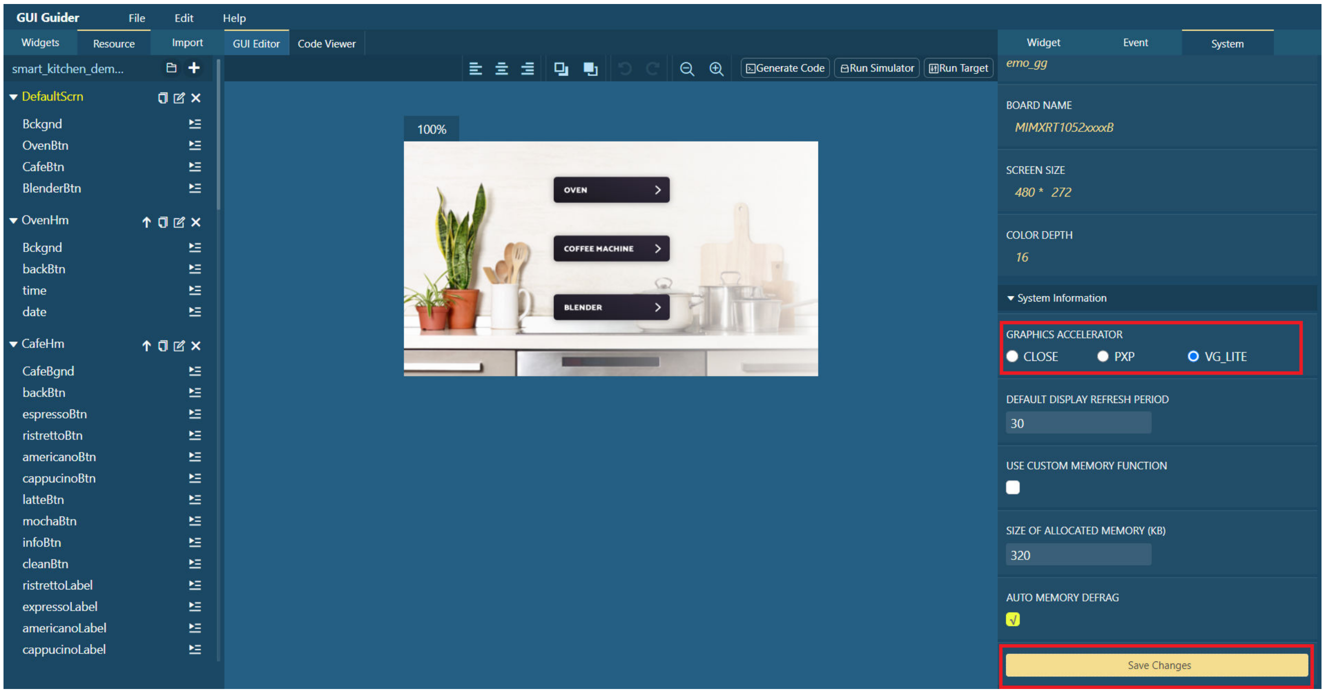
```
#define LV_USE_GPU 1
#define LV_USE_GPU_NXP_PXP 1
#define LV_USE_GPU_NXP_PXP_AUTO_INIT 1
```

In LVGL, PXP is used to accelerate:

- Area fill + optional transparency
- BLIT (BLock Image Transfer) + optional transparency
- Color keying + optional transparency
- Recoloring (color tint) + optional transparency

6.2 VGLite enablement

Enable the VGLite accelerator in GUI Guider.



BOARD NAME
MIMXRT1052xxxxB

SCREEN SIZE
480 * 272

COLOR DEPTH
16

GRAPHICS ACCELERATOR
○ CLOSE ○ PXP ● VG_LITE

DEFAULT DISPLAY REFRESH PERIOD
30

USE CUSTOM MEMORY FUNCTION
□

SIZE OF ALLOCATED MEMORY (KB)
320

AUTO MEMORY DEFRAG
☑

Save Changes

Figure 30. VGLite enablement

NOTE

Click the **Save Changes** button after selecting VG_LITE as the graphics accelerator.

To enable the VGLite accelerator on NXP devices, set the below flag in lv_conf.h. This is required as currently only the color format RGB565 (16 bits) is accelerated on NXP devices.

```
#define LV_COLOR_DEPTH 16
```

VGLite is an API that uses the Vector/Raster 2D GPU. To check whether 2D GPU is available on your NXP device, see the Reference Manual document or the board configuration.

To enable VGLite in LVGL, set the below flags to 1 in lv_conf.h.

```
#define LV_USE_GPU 1
#define LV_USE_GPU_NXP_VG_LITE 1
```

In LVGL, VGLite is used to accelerate:

- Area fill + optional transparency
- BLIT (BLock Image Transfer) + optional transparency

6.3 Recommendations to improve acceleration

This section lists general and VGLite recommendations to improve acceleration.

6.3.1 General recommendations

As a rule when a hardware accelerator processes many pixels in a single batch, it provides better performance than processing small number of pixels multiple times.

The reasons are:

1. **Caches**: Pixels previously processed by CPU are loaded in cache, and must be cleaned and invalidated. The operation takes a few cycles.
2. **Setup time**: Each time HW is used to process pixels, the associated driver configures HW registers, This operation also takes a few cycles.

Thus, NXP has defined a threshold for the minimum number of pixels necessary to trig HW acceleration. These thresholds are defined as preprocessor variables.

For PXP, default values are defined in lv_gpu/lv_gpu_nxp_pxp.h.

- **LV_GPU_NXP_PXP_BLIT_SIZE_LIMIT**: Size threshold for image BLIT, BLIT with color keying, and BLIT with recolor (OPA > LV_OPA_MAX).
- **LV_GPU_NXP_PXP_BLIT_OPA_SIZE_LIMIT**: Size threshold for image BLIT and BLIT with color keying with transparency (OPA < LV_OPA_MAX).
- **LV_GPU_NXP_PXP_FILL_SIZE_LIMIT**: Size threshold for fill operation (OPA > LV_OPA_MAX).
- **LV_GPU_NXP_PXP_FILL_OPA_SIZE_LIMIT**: Size threshold for fill operation with transparency (OPA < LV_OPA_MAX).

For VGLite, default values are defined lv_gpu/lv_gpu_nxp_vglite.h.

- **LV_GPU_NXP_VG_LITE_BLIT_SIZE_LIMIT**: Size threshold for image BLIT (OPA > LV_OPA_MAX).
- **LV_GPU_NXP_VG_LITE_BLIT_OPA_SIZE_LIMIT**: Size threshold for image BLIT with transparency (OPA < LV_OPA_MAX).
- **LV_GPU_NXP_VG_LITE_FILL_SIZE_LIMIT**: Size threshold for fill operation (OPA > LV_OPA_MAX).
- **LV_GPU_NXP_VG_LITE_FILL_OPA_SIZE_LIMIT**: Size threshold for fill operation with transparency (OPA < LV_OPA_MAX).

6.3.2 VGLite recommendations

The 2D GPU behind VGLite has some constraints on the processed buffers:

1. **Address alignment**: Always ensure that the FrameBuffer and pixel buffers are aligned to LV_ATTRIBUTE_MEM_ALIGN_SIZE. You should use the macro LV_ATTRIBUTE_MEM_ALIGN as attribute for statically allocated pixel buffers.
2. **Stride**: Stride is the byte offset between 2 lines of pixels. 2D GPU requires a stride multiple of 16 pixels.

In LVGL: stride = width, so use assets and widgets with a width multiple of 16 pixels.

On platforms like i.MXRT1170 which has both PXP and 2D GPU, prefer 2D GPU as it draws faster than PXP. However, if the GUI contains many pre-rendered semi-tranparent images, PXP may be better.

On platforms with only 2D GPU acceleration (VGLite), try to draw widgets rather than using pre-rendered images as widget, as semi-tranparent image blitting is not yet accelerated.

6.4 Port LVGL code with accelerator to MCUXpresso

After enabling the PXP or VGLite graphics accelerator in GUI Guider, the generated LVGL code is ported to the MCUXpresso example projects.

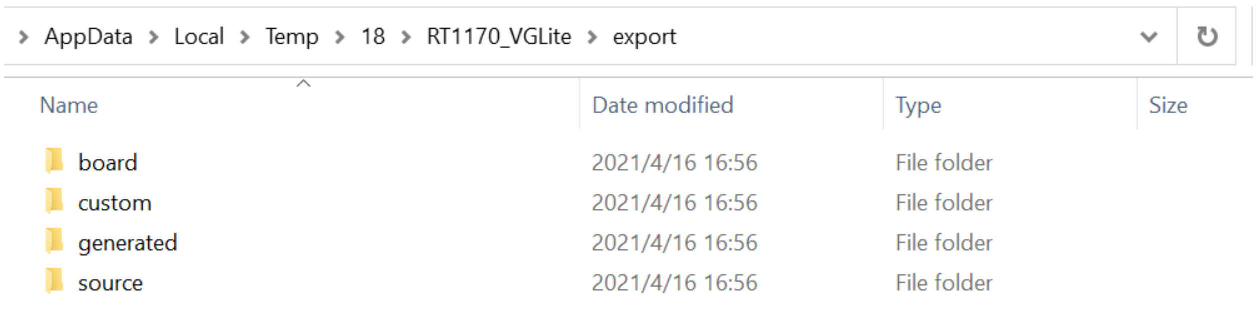
For details on PXP, see Work with MCUXpresso.

For details on VGLite, see Enable VGLite in MCUXpresso projects.

6.4.1 Enable VGLite in MCUXpresso projects

To enable VGLite in MCUXpresso projects, perform the following steps.

1. Design the GUI App using the GUI Guider and export the application source code.



› AppData › Local › Temp › 18 › RT1170_VGLite › export

Name	Date modified	Type	Size
board	2021/4/16 16:56	File folder	
custom	2021/4/16 16:56	File folder	
generated	2021/4/16 16:56	File folder	
source	2021/4/16 16:56	File folder	

Figure 31. Example VGLite export

2. Import the *'littlevgl_guider'* MCUXpresso project. For details, see Work with MCUXpresso.
3. Replace all the folders, listed in Figure 31 with the version generated by GUI Guider.

Chapter 7
Work with MCUXpresso

7.1 Prerequisites

- The C source files generated by GUI Guider.
- MCUXpresso V11.3.0 to compile, download or debug the project.
- SDK of the platform to support related configuration for compile; can be installed via MCUXpresso.

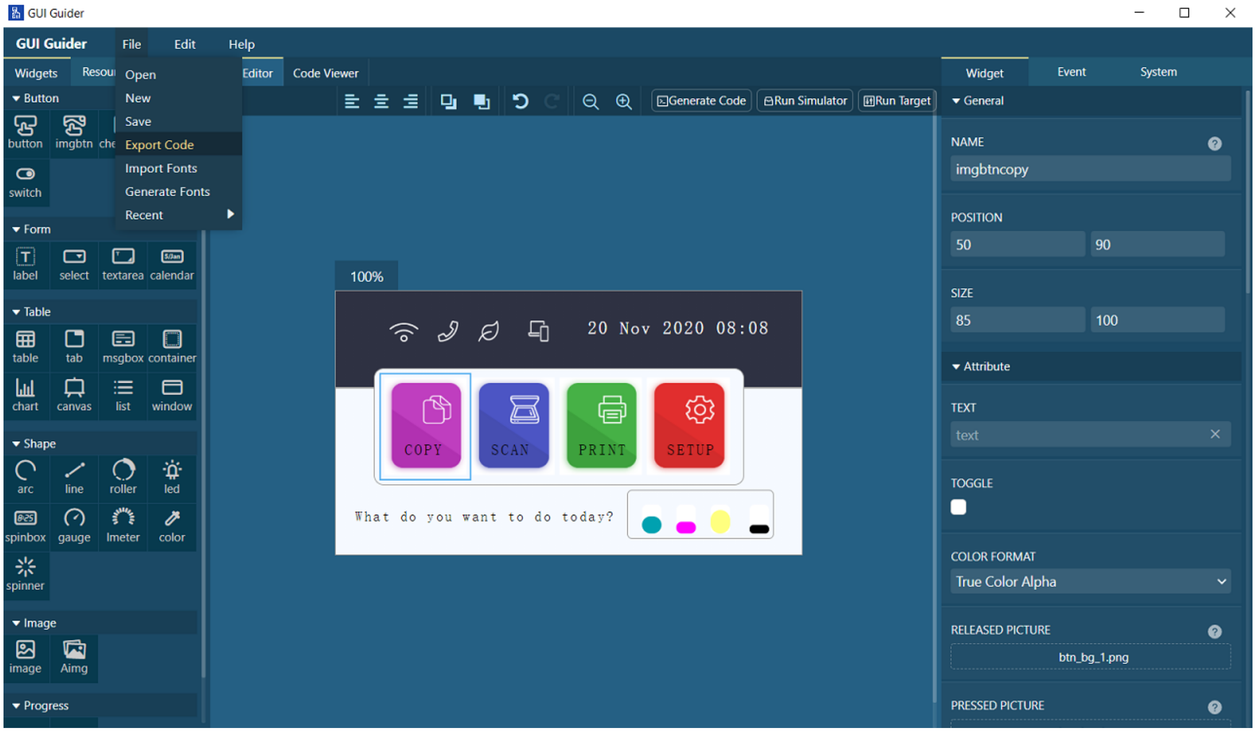
NOTE

For more information on how to download and install the SDK for a device, see the Getting Started page of the particular development board or the development board supporting the device.

7.2 Port LVGL C source file generated by GUI Guider to the Template project

To port LVGL C source file generated by GUI Guider to the Template project, perform the following steps.

1. Design the GUI APP by GUI Guider.
2. Click **Generate Code** to generate source code.
3. Click **File > Export Code** to export the GUI APP source code.



NAME: imgbtncopy

POSITION: 50 90

SIZE: 85 100

TEXT: text

TOGGLE: □

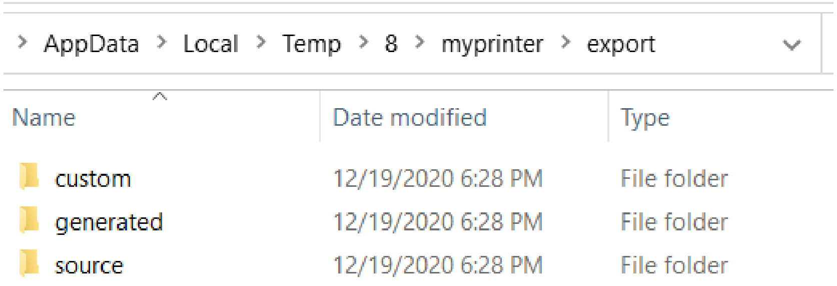
COLOR FORMAT: True Color Alpha

RELEASED PICTURE: btn_bg_1.png

PRESSED PICTURE

Figure 32. Export code

4. The source folder with source files generated by GUI Guider opens.



> AppData > Local > Temp > 8 > myprinter > export

Name	Date modified	Type
custom	12/19/2020 6:28 PM	File folder
generated	12/19/2020 6:28 PM	File folder
source	12/19/2020 6:28 PM	File folder

Figure 33. Generated files

5. Download and install SDK for the board if not already done.

   a. In the **Quickstart Panel**, click **Import SDK example(s)…**.

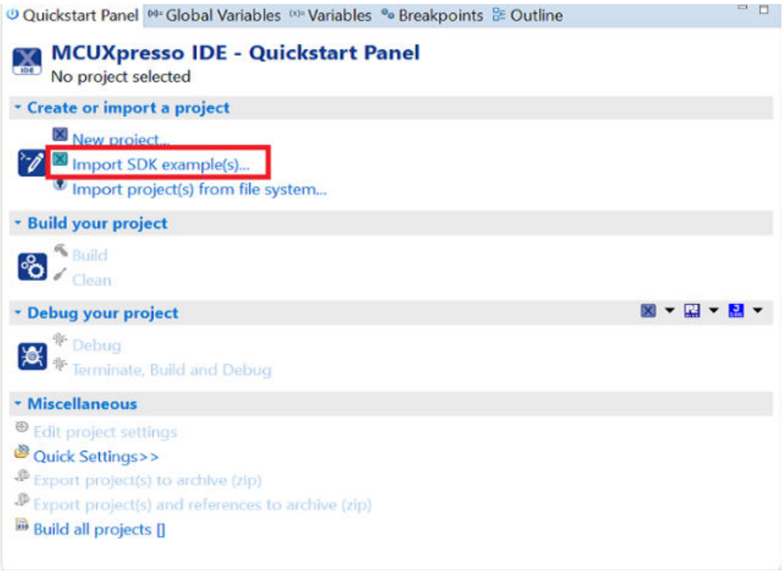
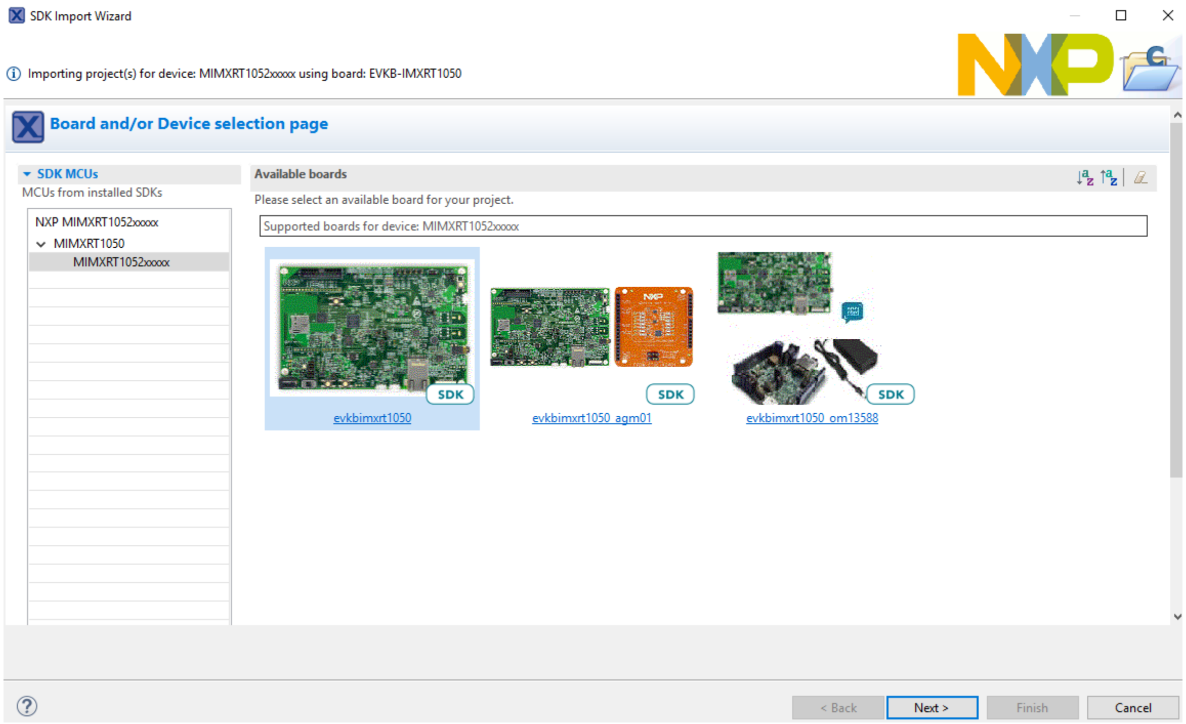


Figure 34. Import SDK example(s)

   b. In the **SDK Import Wizard**, expand the platform folder, select board, and select the specific board from the available boards in the right pane. For example, select *MIMXRT1050 > MIMXRT1052xxxxx > evkbimxrt1050*.



SDK Import Wizard

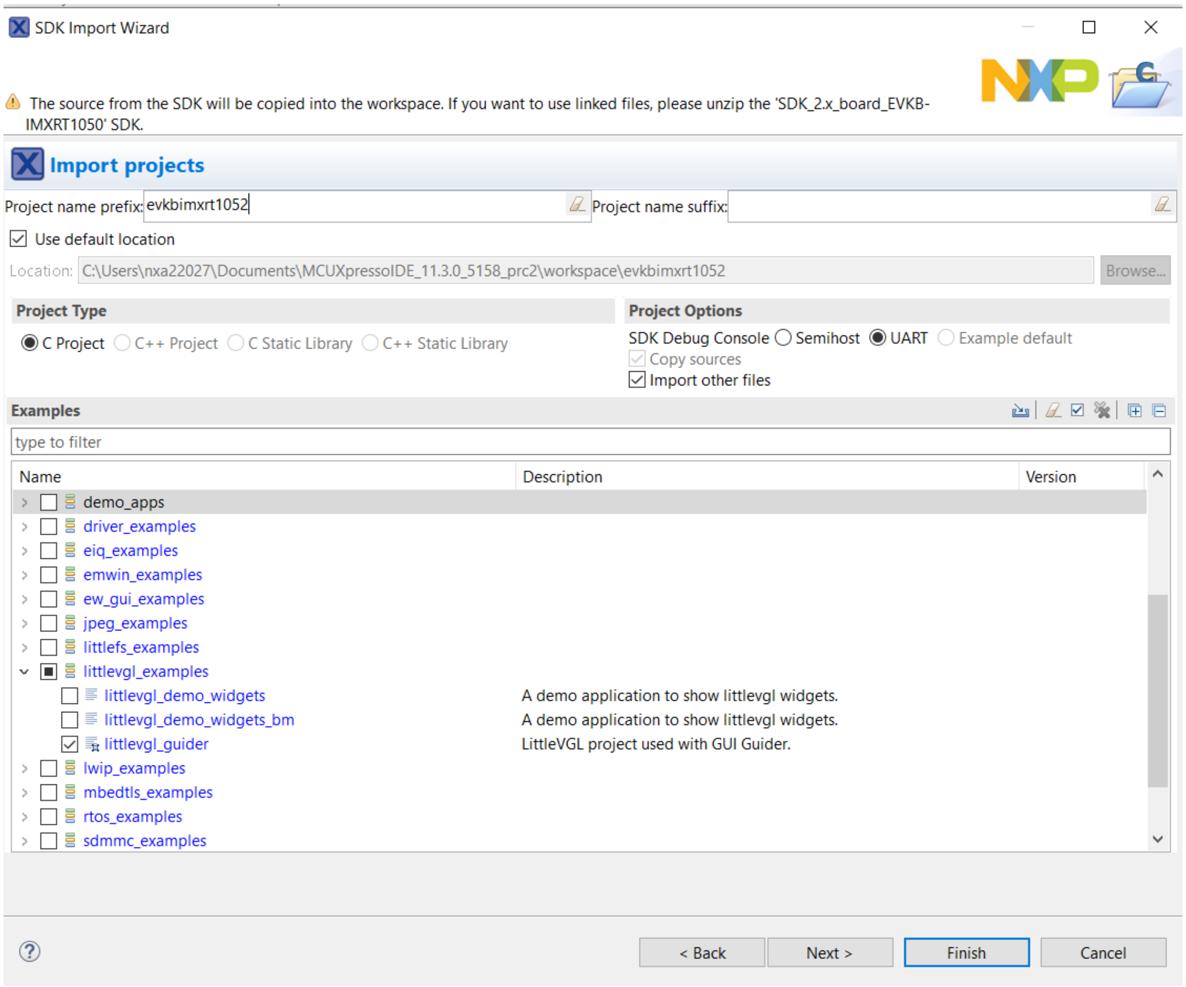
Importing project(s) for device: MIMXRT1052xxxxx using board: EVKB-IMXRT1050

Board and/or Device selection page

Figure 35. Select an available baord for your poject

c. Click the **Next** button.

d. Expand the *littlevgl_examples* folder and select *littlevgl_guider*.



SDK Import Wizard

⚠ The source from the SDK will be copied into the workspace. If you want to use linked files, please unzip the 'SDK_2.x_board_EVKB-IMXRT1050' SDK.

Import projects

Project name prefix: evkbimxrt1052 Project name suffix:

☑ Use default location

Location: C:\Users\nxa22027\Documents\MCUXpressoIDE_11.3.0_5158_prc2\workspace\evkbimxrt1052 Browse...

Project Type

● C Project ○ C++ Project ○ C Static Library ○ C++ Static Library

Project Options

SDK Debug Console ○ Semihost ● UART ○ Example default

☑ Copy sources

☑ Import other files

Examples

type to filter

Name	Description	Version
› □ demo_apps		
› □ driver_examples		
› □ eiq_examples		
› □ emwin_examples		
› □ ew_gui_examples		
› □ jpeg_examples		
› □ littlefs_examples		
⌄ ■ littlevgl_examples		
□ littlevgl_demo_widgets	A demo application to show littlevgl widgets.	
□ littlevgl_demo_widgets_bm	A demo application to show littlevgl widgets.	
☑ littlevgl_guider	LittleVGL project used with GUI Guider.	
› □ lwip_examples		
› □ mbedtls_examples		
› □ rtos_examples		
› □ sdmmc_examples		

Figure 36. Select littlevgl_guider

e. Click the **Finish** button to create the MCUXpresso project.

6. Replace the "custom", "generated", and "source/lv_conf.h" folders with the version generated by GUI Guider.
7. Build the project in MCUXpresso IDE.
8. Deploy image to the board with MCUXpresso.

NOTE

For details on downloading projects to your development board, see the development board's Getting Started information.

Chapter 8
Frequently Asked Questions (FAQs)

This chapter lists the Frequently Asked Questions (FAQs) about GUI Guider.

Question: How to avoid simulator running the MCU-specific code?

Answer: GUI Guider provides a predefined macro LV_USE_GUIDER_SIMULATOR in lv_conf.h. Do the following changes in your source files:

```
#if !LV_USE_GUIDER_SIMULATOR  // or LV_USE_GUIDER_SIMULATOR == 0
...(MCU specific Code)
#endif
```

Chapter 9
Revision history

This table summarizes revisions to this document.

Table 19. Revision history

Revision number	Date	Substantive changes
0	29 October 2020	Initial release.
1	17 November 2020	Updated Work with MCUXpresso.
2	11 January 2021	Added and updated multiple sections.
3	10 May 2021	Added and updated multiple sections for v1.1.

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

Right to make changes - NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Security — Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer’s applications and products. Customer’s responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer’s applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately. Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 10 May 2021
Document identifier: GUIGUIDERUG