

GUIGUIDERUG

GUI Guider User's Guide

Rev. 11 — 31 March 2023

User guide

Document information

Information	Content
Keywords	GUIGUIDERUG, IDE, GUI, MCU, LVGL, RTOS
Abstract	This document describes GUI Guider and targets embedded GUI application developers with a basic knowledge of C on NXP MCU devices.



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.
- GUI Guider 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.
- LVGL hardware acceleration — Steps to use PXP/VGLite hardware acceleration.
- Performance — Introduces the performance monitor function and tips for performance optimization for NXP MCU devices.
- Debug GUI Guider project — Description of how to debug GUI applications designed by GUI Guider in supported IDE and toolchain.
- MicroPython — Introduces how to design GUIs, generate MicroPython code, and run Python code in a simulator.
- Porting RTOS — Steps to port source code of GUIs for variant RTOS, build, and deploy binary image on NXP MCU.
- External storage — Describes how to use external storage (SD card) to store images and video in the GUI Guider project.
- External image in QSPI flash — Introduce how to store the image binary in QSPI flash and access the image data in GUI application.
- Frequently Asked Questions (FAQs) — Frequently asked questions and answers.

1.1 Supported features

- IDE
 - Supports Windows 10, MacOS 11, and Ubuntu 22.04
 - Multi-language (English, Chinese) for IDE - supports multi-instance
 - Multi-LVGL-version (v7.10.1, v8.3.2)
 - LVGL C and Micropython code auto-generation
 - RT-Thread and Zephyr RTOS (Real-Time Operating System)
 - Compatible with MCUXpresso IDE v11.7.0, MCU SDK 2.13.1, and IAR 9.32.1, Keil MDK 5.37.0
 - Project management: create, import, edit, delete
 - Wizard of creating a project
 - GUI auto-scaling (auto ratio, custom ratio)
 - Configurable display type with custom option
 - 16-bit and 24-bit color depth support
 - What You See Is What You Get (WYSIWYG) UI design by drag and drop
 - Multi-page application design
 - Shortcut of bring forward and backward, copy, paste, delete, undo, redo
 - Code viewer for UI definition JSON file, C code, and Micropython code.
 - Navigation bar to view the selected source file
 - Widget attributes group and setting, widgets group move

- Screen preview and copy function, default screen setting mouse
- GUI editor zoom in and zoom out, control by mouse scroll
- Multiple fonts support and third-party fonts import
- Customizable Chinese character scope
- Widgets alignment: horizontal (top, center, and bottom), vertical align (left, center, and right), alignment line
- PXP and VGLite acceleration enable and disable
- Support the CPU/FPS monitor
- Supports default style and custom style
- Integrated demo applications
- Information window: project info, memory info, and real-time log
- Flexible font size
- Events for list buttons
- Memory leak check
- Supports cup runtime memory monitoring
- Optimization level options: size, speed, balance - supports widget visibility
- Widgets
 - V7 supports 32 widgets
 - Button (5): button, image button, checkbox, button group, switch
 - Form (4): label, drop-down list, text area, calendar
 - Table (9): table, tab, message box, container, chart, canvas, list, window, titleview
 - Shape (7): arc, line, roller, LED, spinbox, color picker, spinner
 - Image (3): image, animation image, 3D image
 - Progress (2): bar, slider
 - Gauge (2): gauge, line meter
 - V8 supports 42 widgets
 - Button (6): button, image button, checkbox, button group, switch, radio button
 - Form (6): label, spangroup, drop-down list, text area, calendar, date text box
 - Table (10): table, tab, message box, container, chart, canvas, list, window, titleview, menu
 - Gauge (2): meter, digital clock
 - Shape (7): arc, line, roller, LED, spinbox, color picker, spinner
 - Image (3): image, animation image, 3D image
 - Progress (3): bar, slider, textprogress
 - Advanced (5): analog clock, carouse, video, lottie, QRcode
 - Shadow setting
 - Common functions
 - Animation: lottie animation, animation image, GIF to animation, animation easing, and [animation path](#)
 - Support event trigger and action selection, custom action code
 - Support tileview design by drag and drop operation in editor
 - Support parent/child hierarchy for carousel container, tabview, and tileview
 - Supports two widgets theme:
 - Default
 - Dark
 - Support four IDE themes:
 - Dark Blue
 - Light Blue
 - Light
 - Dark

- Chinese display and Chinese input
- Support fonts:
 - .simsun (can support Chinese characters)
 - .arial
 - .montserratMedium
 - .Abel_regular
 - .Acme_Regular
 - .Adventpro_regula
 - .AguafinaScript_Regular
 - .Alatsj_Regular
 - .AlexBrush_Regular
 - .AmaticSC_Regular
 - .Amiko_Regular
 - .Antonio_Regular
 - .ArchitectsDaughter
- Target devices
 - NXP i.MX RT595, i.MX RT1010, i.MX RT1015, i.MX RT1020, i.MX RT1024, i.MX RT1040, i.MX RT1050, i.MX RT1060EVKB, i.MX RT1062, i.MX RT1064, i.MX RT1160, i.MX RT1170 (portrait mode and landscape mode) and i.MX RT1170EVKB (portrait mode and landscape mode).
 - NXP LPC54628, LPC54S018, LPC54S018m, LPC55S0, LPC55S16, LPC55S28, and LPC55S69.
- NXP MCXN947BRK
 - External flash storage for LPC54628

Board	Verified display part number
LPC54S018J4M	RK043FN02H-CT, RK043FN66HS-CTG
LPC54S018	RK043FN02H-CT, RK043FN66HS-CTG
LPC54628	RK043FN02H-CT, RK043FN66HS-CTG
LPC55S69	adafruit-1947
LPC55S28	adafruit-1947
LPC55S16	adafruit-1947
LPC55S06	adafruit-1947
MCXN947BRK	Mikroe TFT Proto 5"
MIMXRT1011xxxxx	adafruit-1947
MIMXRT1015xxxxx	adafruit-1947
MIMXRT1021xxxxx	adafruit-1947
MIMXRT1024xxxxx	adafruit-1947
MIMXRT1166xxxxx	RK055AHD091, RK055MHD091
MIMXRT1176xxxxx	RK055AHD091, RK055MHD091
MIMXRT1170EVKB	RK055AHD091, RK055MHD091
MIMXRT595S	G1120B0MIPI, Mikroe TFT Proto 5", RK055AHD091, RK055MHD091
MIMXRT1062xxxxA	RK043FN02H-CT, RK043FN66HS-CTG
MIMXRT1060EVKB	RK043FN02H-CT, RK043FN66HS-CTG

Board	Verified display part number
MIMXRT1064xxxxA	RK043FN02H-CT, RK043FN66HS-CTG
MIMXRT1040xxxxB	RK043FN02H-CT, RK043FN66HS-CTG
MIMXRT1052xxxxB	RK043FN02H-CT, RK043FN66HS-CTG

- Device template, auto-build, and auto-deploy for supported platforms

Feature list	lvgl version	Toolchain	Platform
widget:lottie	v8.3.2	MCUxpresso	MIMXRT1042xxxx B, MIMXRT1052xxxx A, MIMXRT1062xxxx A, MIMXRT1064xxxx A, MIMXRT1166xxxx A, MIMXRT1176xxxxA
widget:video	v8.3.2	MCUxpresso, IAR, ARMGcc, MDK	MIMXRT1042xxxx B, MIMXRT1052xxxx A, MIMXRT1062xxxx A, MIMXRT1064xxxx A, MIMXRT1166xxxx A, MIMXRT1176xxxxA
SD card storage	v8.3.2	MCUxpresso, IAR, ARMGcc, MDK	MIMXRT1042xxxx B, MIMXRT1052xxxx A, MIMXRT1062xxxxA, MIMXRT1064xxxxA, LPC54 S018M

1.2 Hardware requirement of LVGL application

Every modern controller which is able to drive a display is suitable to run LVGL. The minimal requirements are:

- 16, 32, or 64-bit microcontroller or processor.
- 16 MHz clock speed is recommended.
- Flash/ROM: > 64 kB for the very essential components (> 180 kB is recommended).
- RAM:
 - Static RAM usage: ~2 kB depending on the used features and object types.
 - Stack: > 2 kB (> 8 kB is recommended).
 - Dynamic data (heap): > 4 kB (> 32 kB is recommended if using several objects). Set by LV_MEM_SIZE in lv_conf.h.
 - Display buffer: > "Horizontal resolution" pixels (> 10 × "Horizontal resolution" is recommended).
 - One frame buffer in the MCU or in an external display controller.
- Basic C (or C++) knowledge: pointers, structures, and callbacks.

Note: Memory usage may vary depending on architecture, compiler, and build options.

2 Installation

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

2.1 Prerequisites of Windows 10

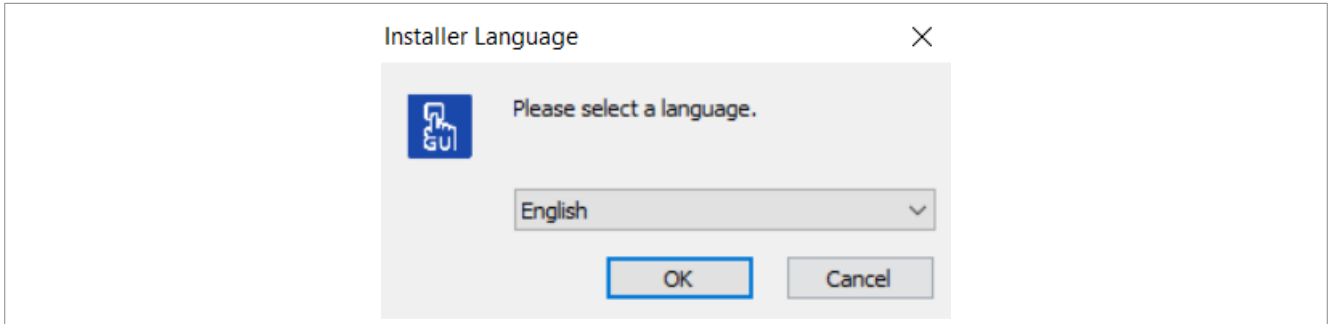
- PC with Windows 10

- Java Runtime Environment, JRE-8 is verified

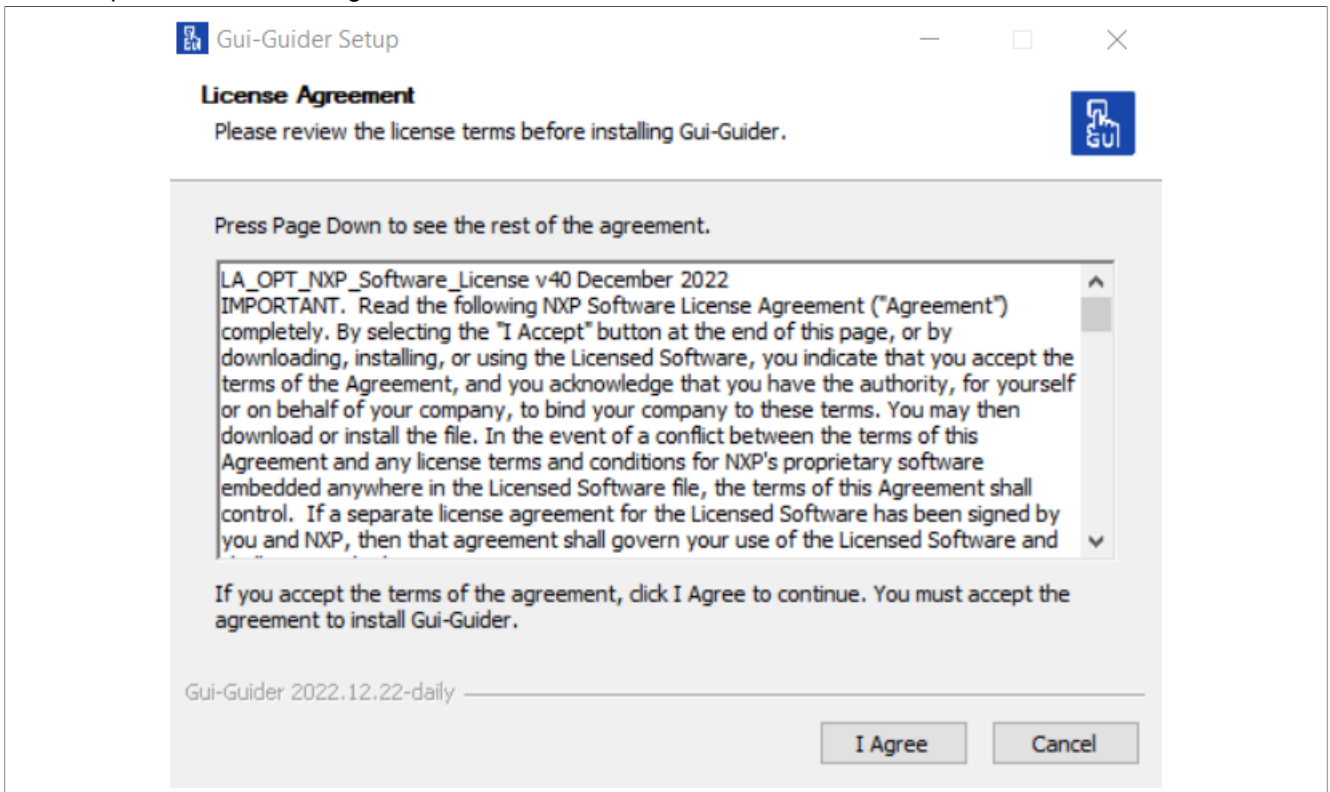
2.2 Install GUI Guider on Windows 10

To install GUI Guider on Windows 10, perform the following steps:

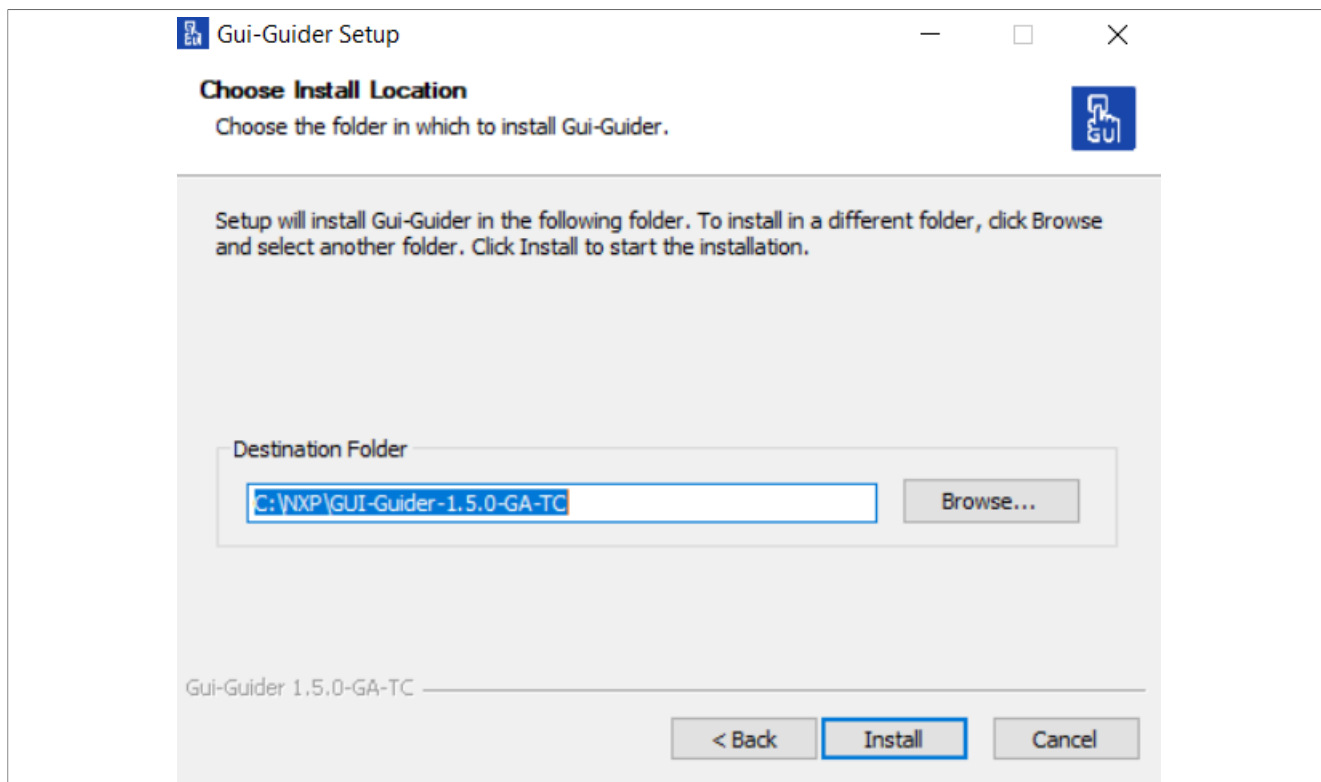
1. Download the installer from www.nxp.com/gui-guider.
2. Double-click the installer to install the software.
3. Choose the language.



4. Accept the terms of the agreement.



5. Select the location where you want to install Gui-Guider.



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 the OpenJDK source package from <http://jdk.java.net/archive/>.
2. Extract the zip file into a folder, for example, `C:\Program Files\Java\`. The extraction creates a `jdk-16` folder with a sub folder named `bin`. However, you may require administrator privileges to extract the zip file to the location.
3. Set the `PATH` environment variable.
 - a. Select **Control Panel**.
 - b. Click **System**.
 - c. Select **Advanced system settings**.
 - d. Click the **Advanced** tab.
 - e. Click the **Environment Variables** button.
The **Environment Variables** dialog box appears.
 - f. Locate the `PATH` variable.
 - g. Click the **Edit** button.
 - h. Click the **New** button.
 - i. Add the location of the `bin` folder of the JDK installation. For example:
`C:\WINDOWS\system32`
`C:\WINDOWS`

```
C:\Program Files\Java\jdk-16\bin
```

4. Set a new system variable `PATH` as `JAVA_HOME`:

- a. Under the **System variables**, click **New**.

The **New System Variable** dialog box appears.

- b. Enter the **Variable name** as `JAVA_HOME`.

- c. Enter the **Variable value** as the installation path of the JDK (without the bin subfolder). For example:

```
C:\WINDOWS\system32
```

```
C:\WINDOWS
```

```
C:\Program Files\Java\jdk-16
```

- d. To save the changes, click the **OK** button.

- e. To close the **Environment Variables** dialog box, click the **OK** button.

Your environment is set now.

5. Open the command prompt and type `java -version` and see if it prints the version of the newly installed JDK.

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 22.04

- PC with Ubuntu 22.04

2.6 Install GUI Guider on Ubuntu 22.04

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

```
$ sudo apt install ./Gui-Guider-Setup-1.5.1-GA.deb
```

2.7 Uninstall GUI Guider on Ubuntu 22.04

Run the following command in the command-line tool.

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

2.8 Prerequisites of MacOS

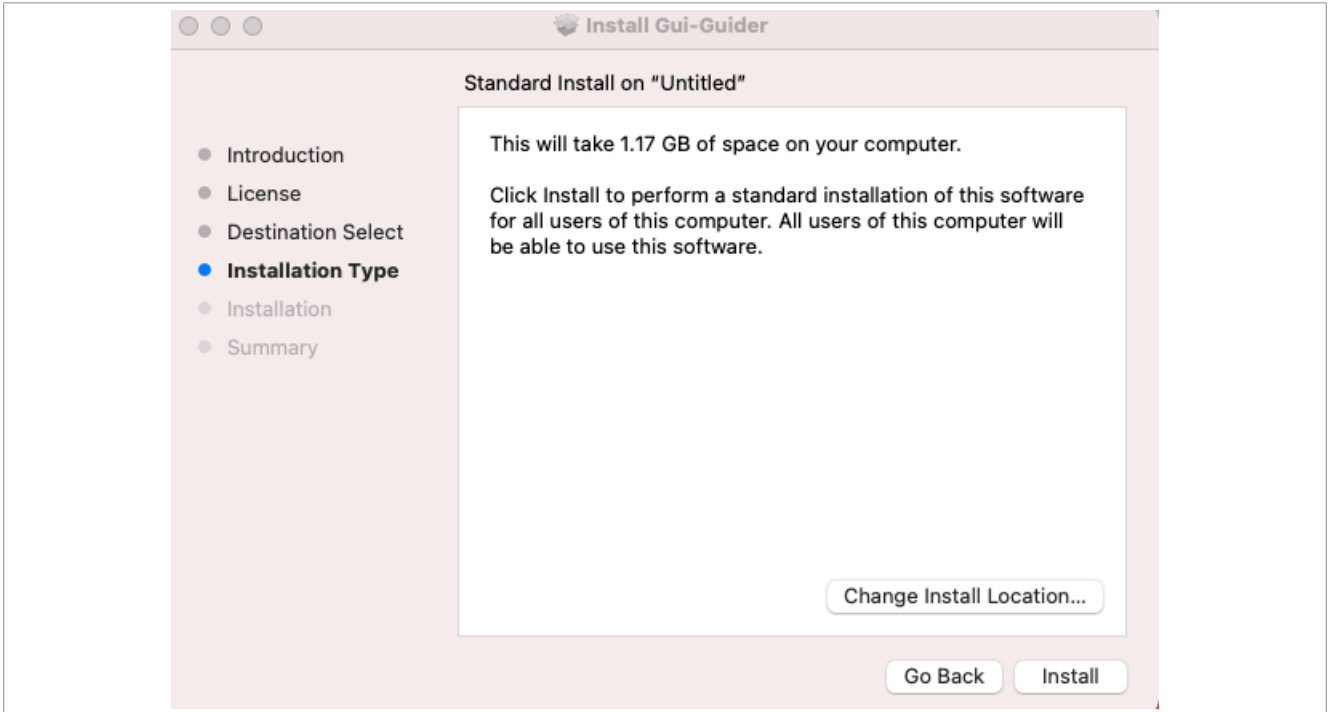
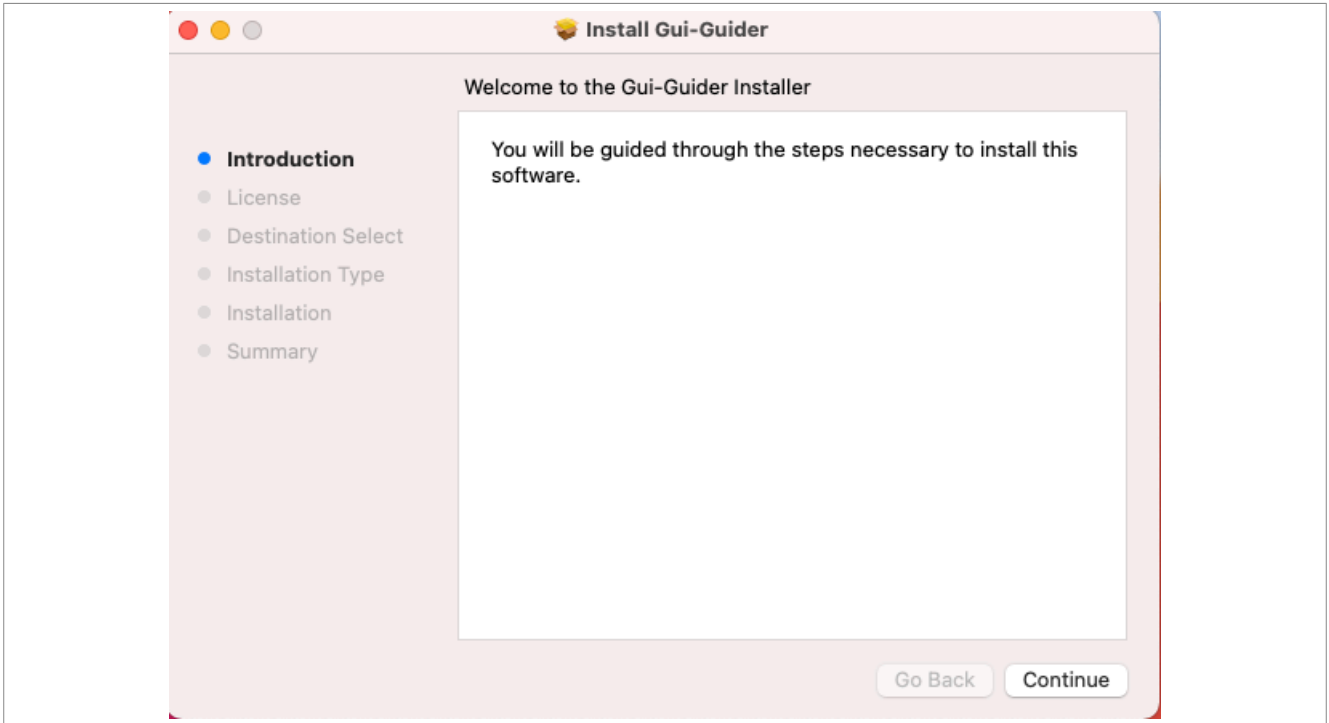
The following steps are required before installing GUI Guider on MacOS:

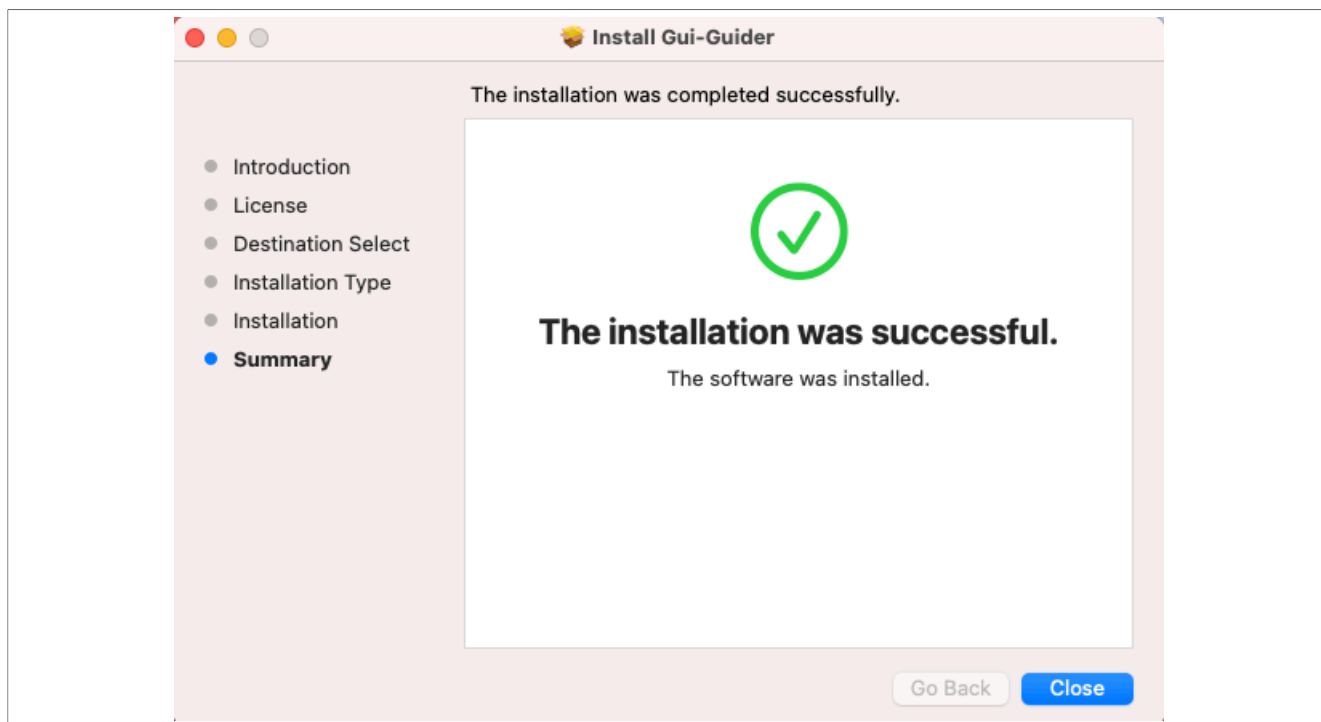
1. Install the version compatible with the OS Xcode.
2. Install the brew.
3. Use brew install JRE, SDL2, and cmake.
4. Install the MCUXpresso IDE.

2.9 Install GUI Guider on MacOS

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

1. Download the installer from www.nxp.com.
2. Click the installer package.





2.10 Uninstall GUI Guider on MacOS

To uninstall GUI Guider on MacOS, run the following command.

```
$ rm -r /Applications/Gui-Guider.app
```

3 Usage

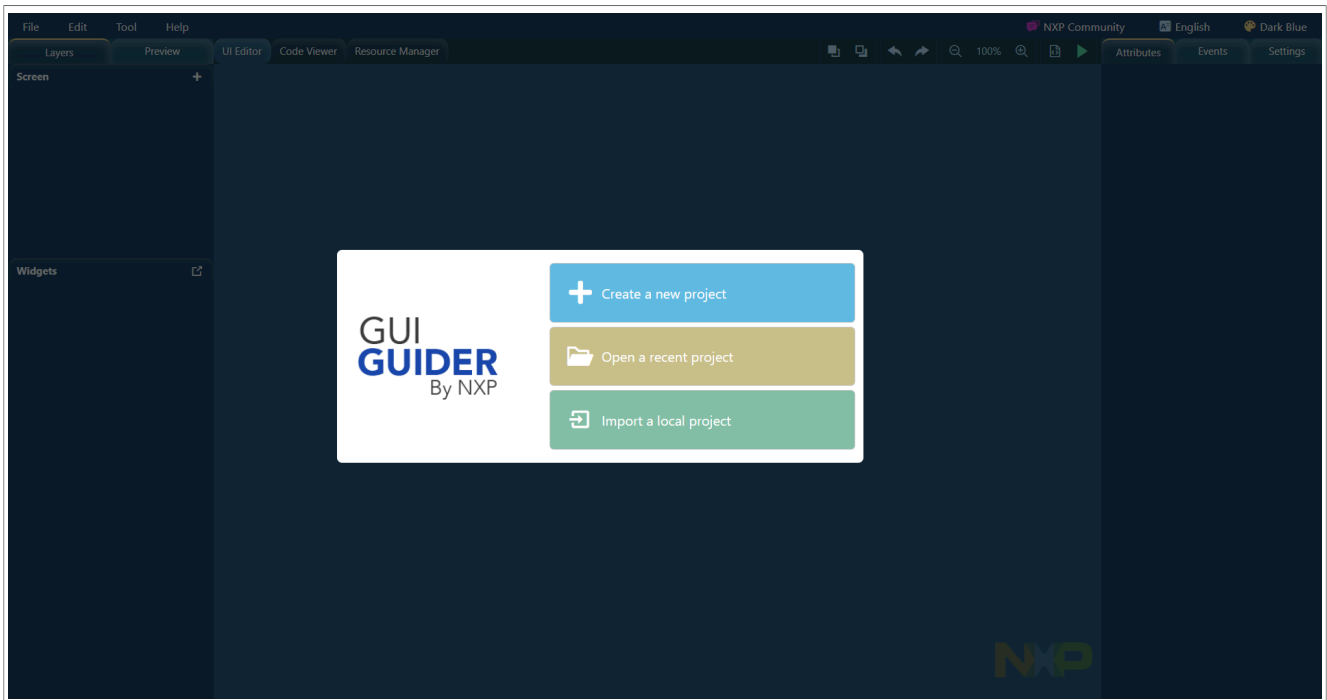
When GUI Guider is launched, the following options appear:

- Create a new project: Wizard to create a project.
- Open a recent project: Enables you to open a project from recent projects list.
- Import a local project: Enables you to import a project from the local disk.

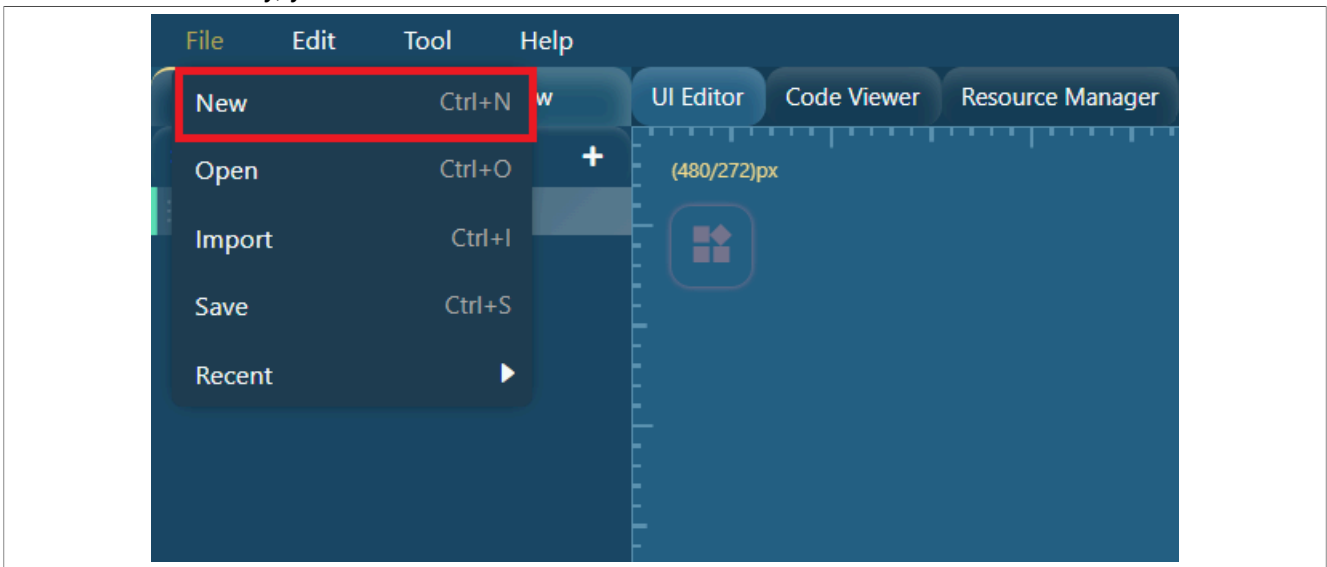
3.1 Create a project

To create a project, perform the following steps:

1. Click the **Create a new project** button.



Note: Alternatively, you can select **File > New** in the GUI editor.

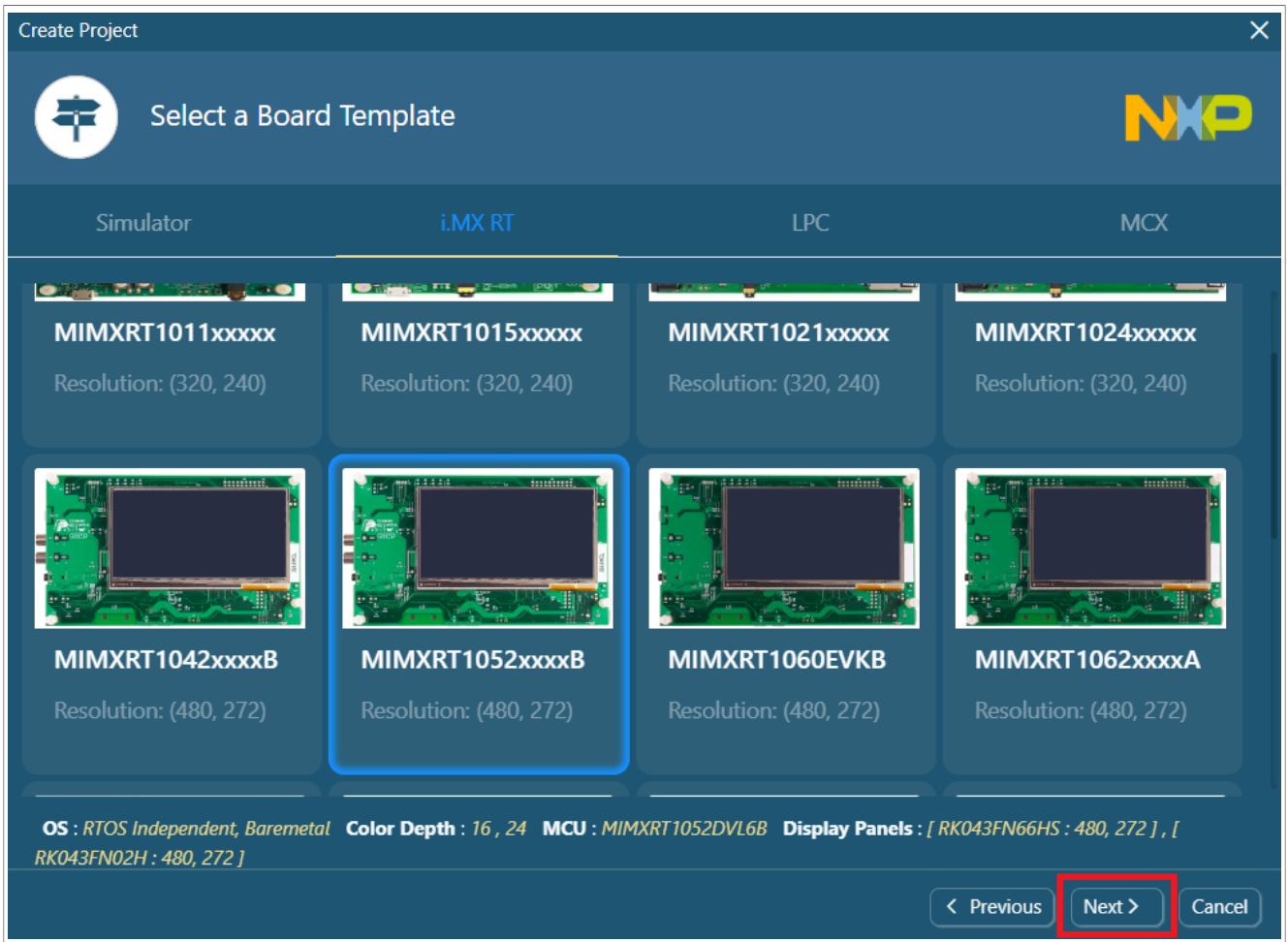


The **New Project** wizard dialog box appears.

2. Select the LVGL version that you want to use. For example, v8.

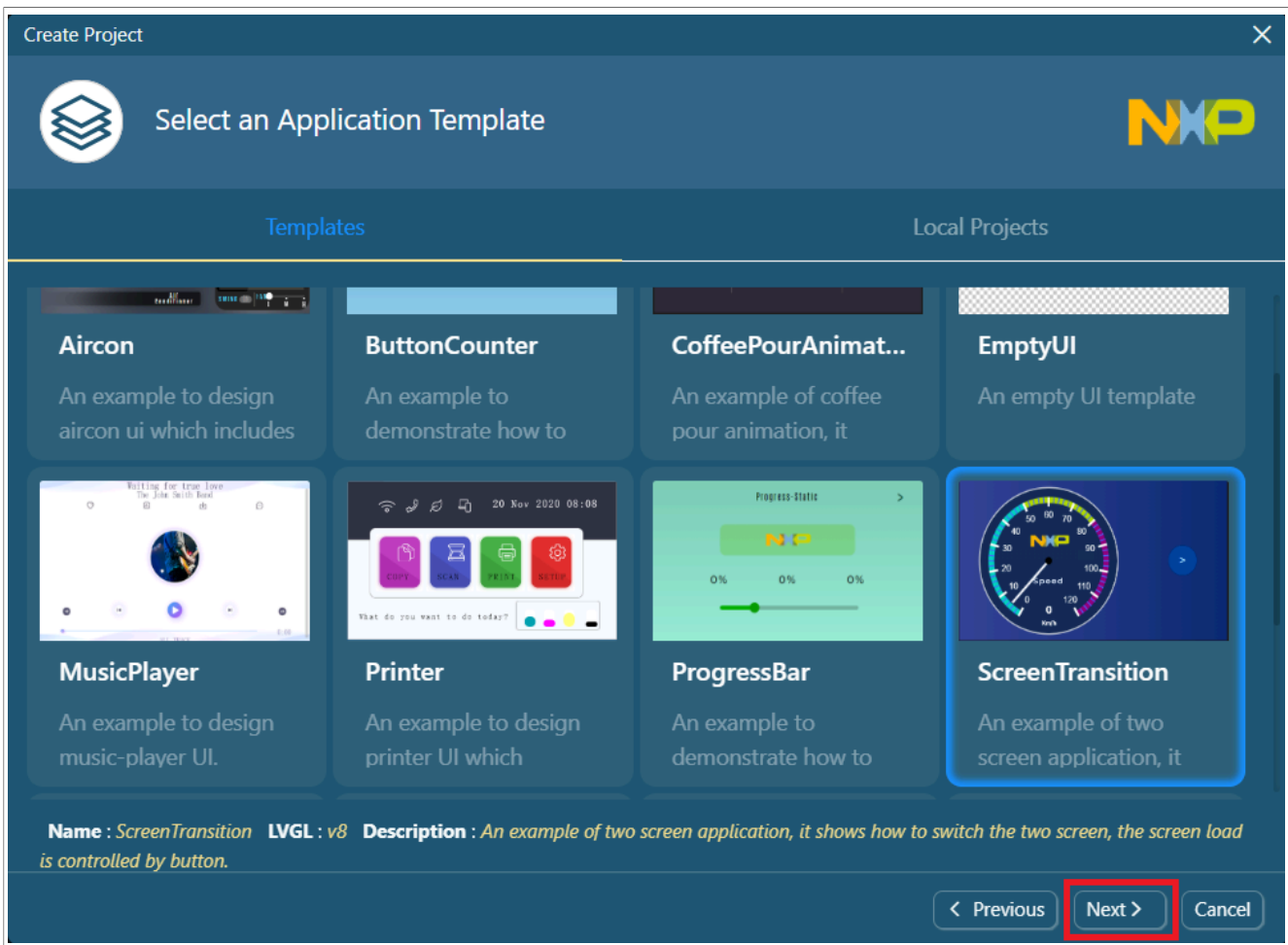


3. Click **Next** or double click.
The **Select a Board Template** page of the wizard appears.
4. Click the **Simulator**, **i.MX RT**, or **LPC** tab and select a board from the template list. For example, select MIMRT1052xxxxB.
5. Click **Next** or double click.



The **Select an Application Template** appears.

6. Select a GUI application template, For example, ScreenTransition.

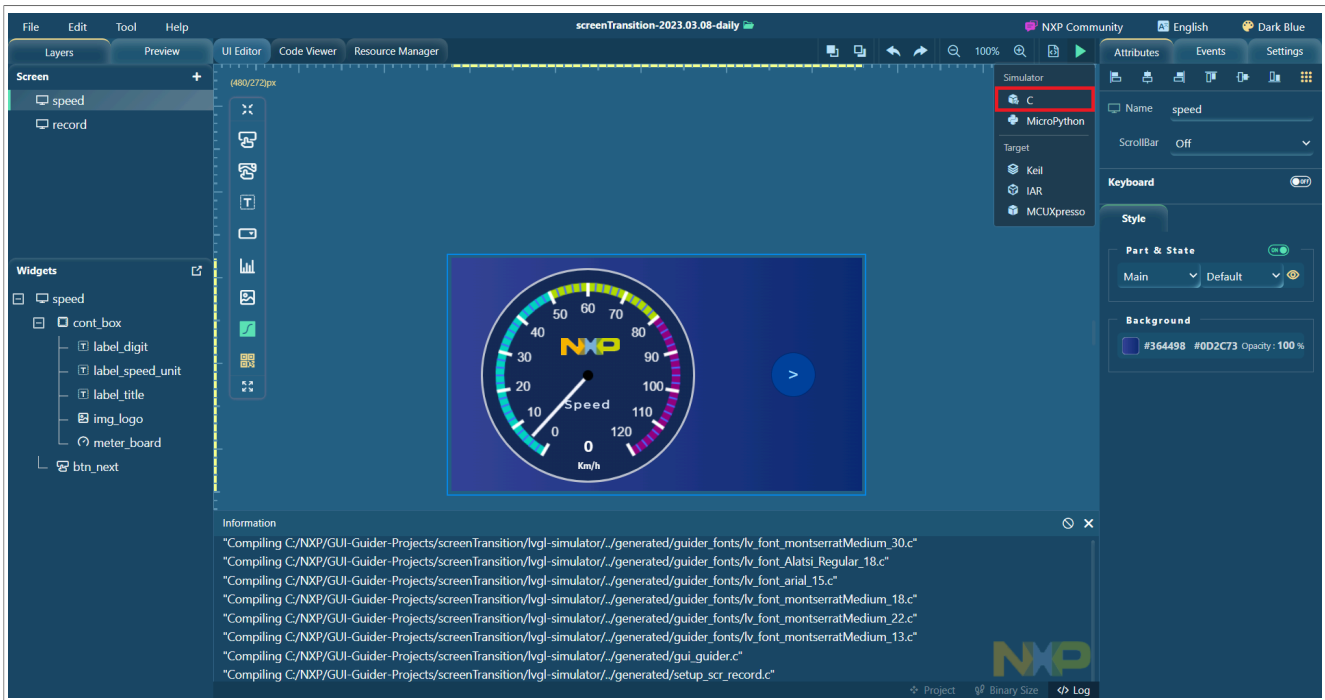


7. Click **Next** or double click.
The **Project Settings** page appears.
8. Configure the basic information of the project, including **Project Name**, **Project Directory**, **Panel Type**, **Color Depth**, and **GUI Resize**.
Note: To ensure that the GUI application is displayed normally on the board, select **Auto Ratio**. To customize the size of the application display, set the scaling ratio of width and height in custom ratio.



Note: GUI Guider supports multiple panel types for each board. The new panel is selected by default. Check the display type on your board and select the right panel type.

9. Click **Create**.
The project appears in the **Editor** tab.
10. To run the GUI application in a simulator, click  and select the simulator. For example, select C.



The project log appears in the **Information** view.

3.2 Open a project

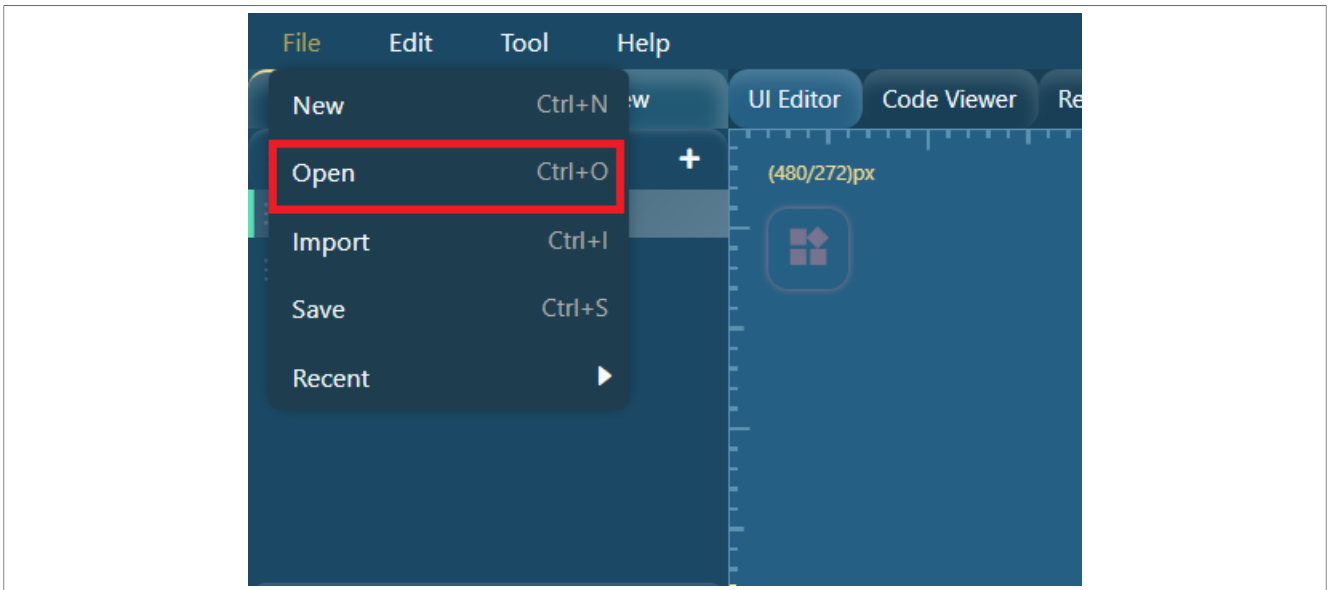
To open a recent project, perform the following steps:

1. Click the **Open a recent project** button.

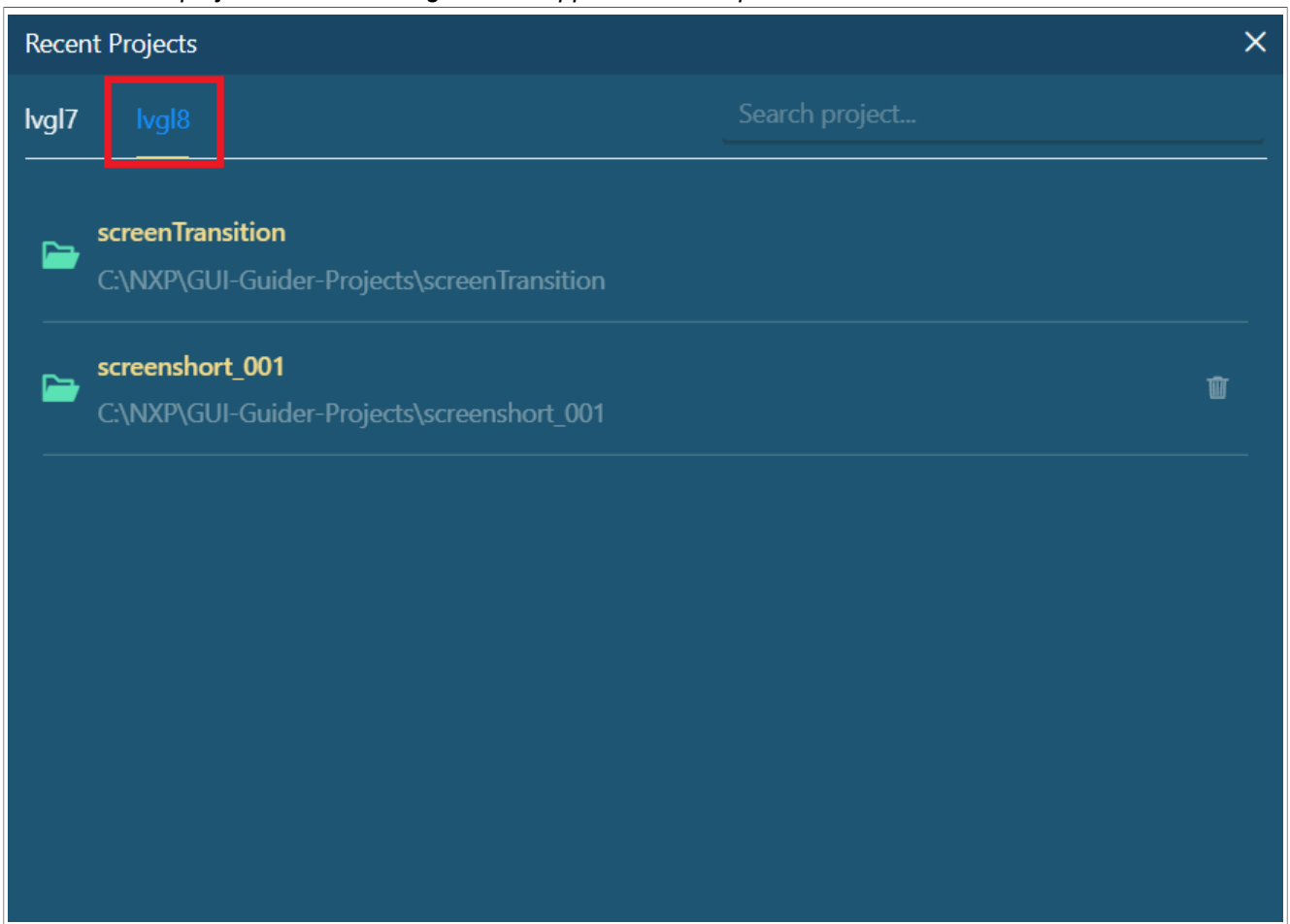


The **Manage project** dialog box appears with a list of existing projects.

Note: Alternatively, you can select **File > Open** in the GUI editor.



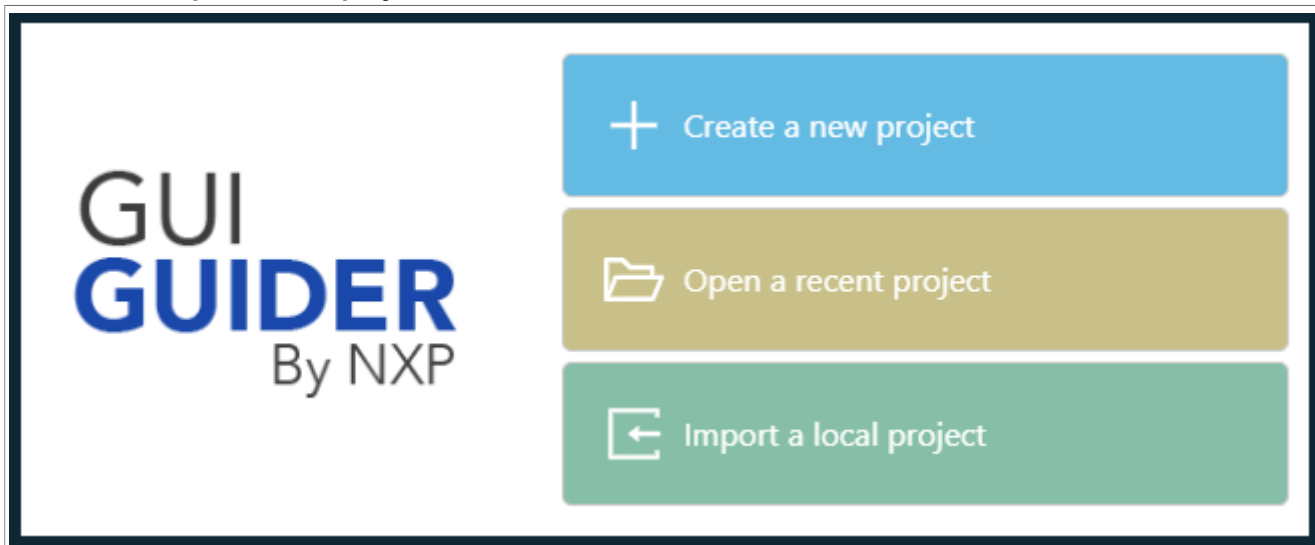
2. Select a project in the list.
The selected project opens in the GUI editor.
Note: The projects of different lvgl version appear in the respective tabs.



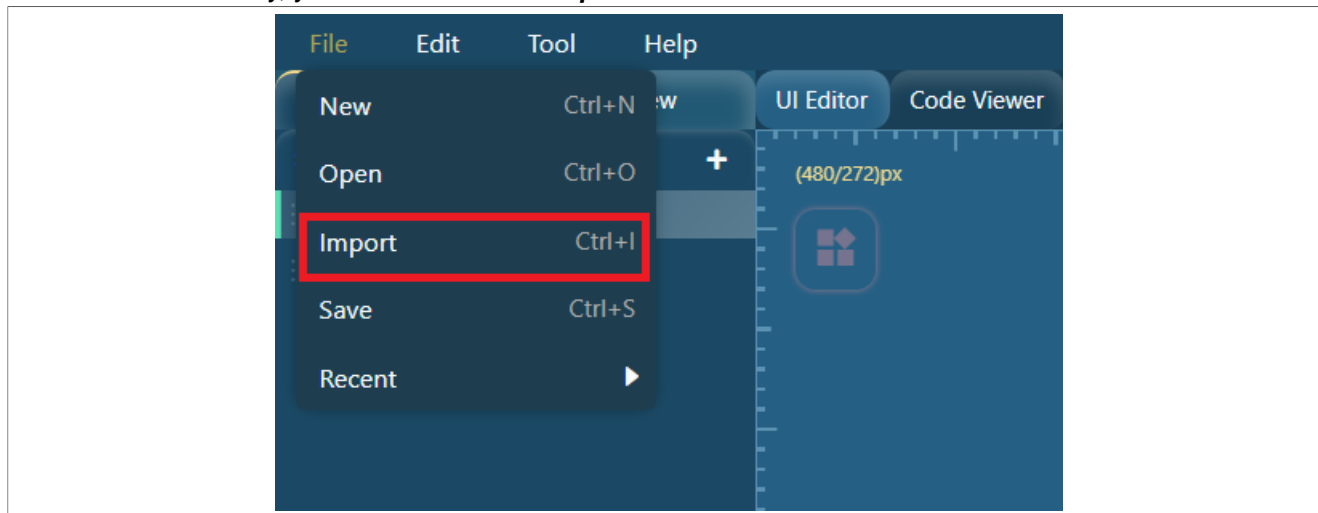
3.3 Import a local project

To import an existing project, perform the following steps.

1. Click the **Import a local project** button.

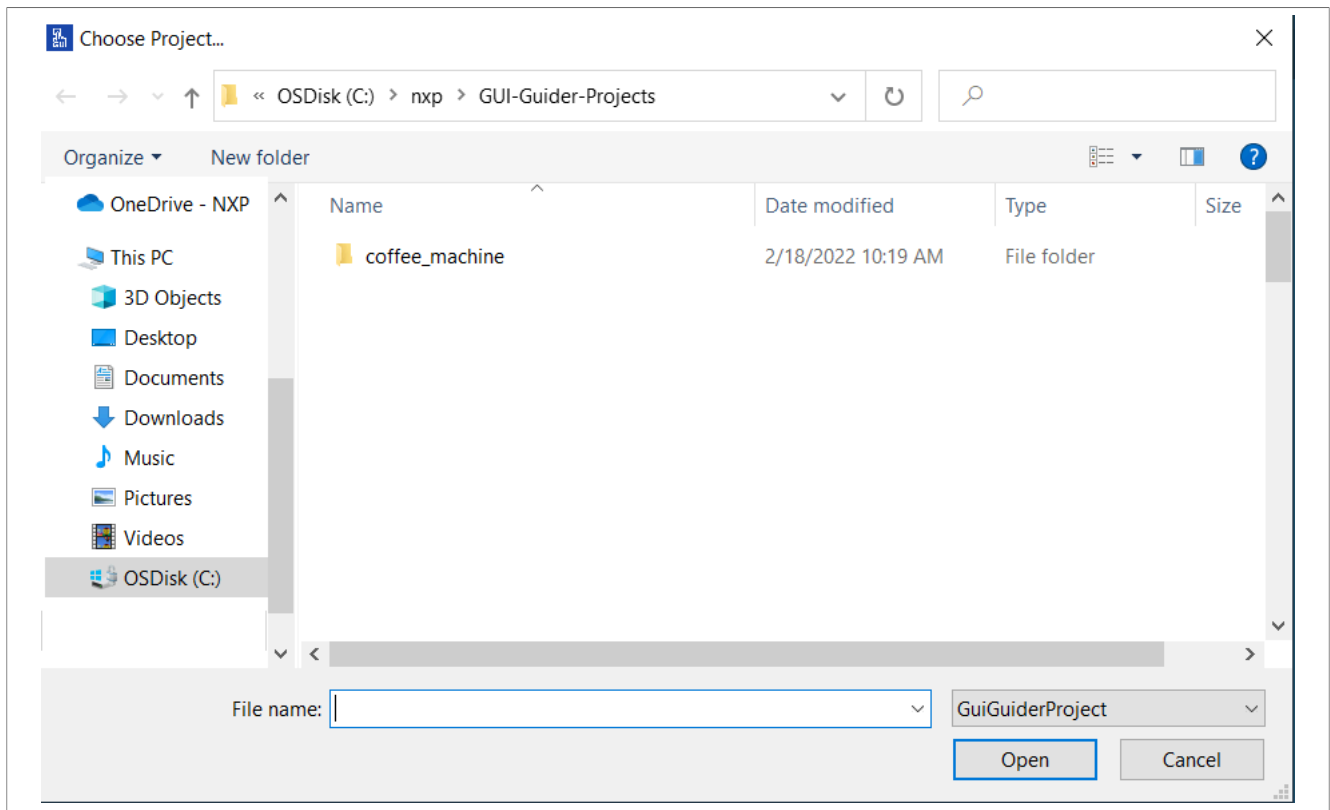


Note: Alternatively, you can select **File > Import** from the GUI editor.

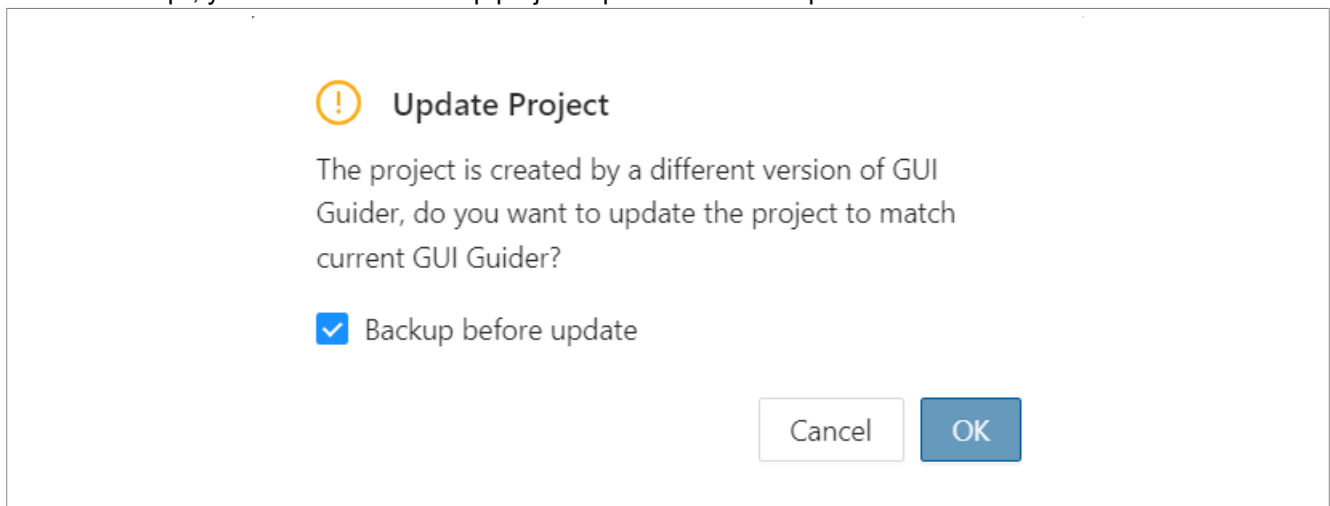


The **Choose Project** dialog box appears.

2. Navigate to the project that you want to import from your local directory.
3. Click **Open**.



The project is imported in the editor. However, if you try to import an older version of the project, a message prompts whether you want to update the project to match the current GUI Guider. Click **OK** to proceed. If you select "Backup", you can find the backup project zip file in the workspace.

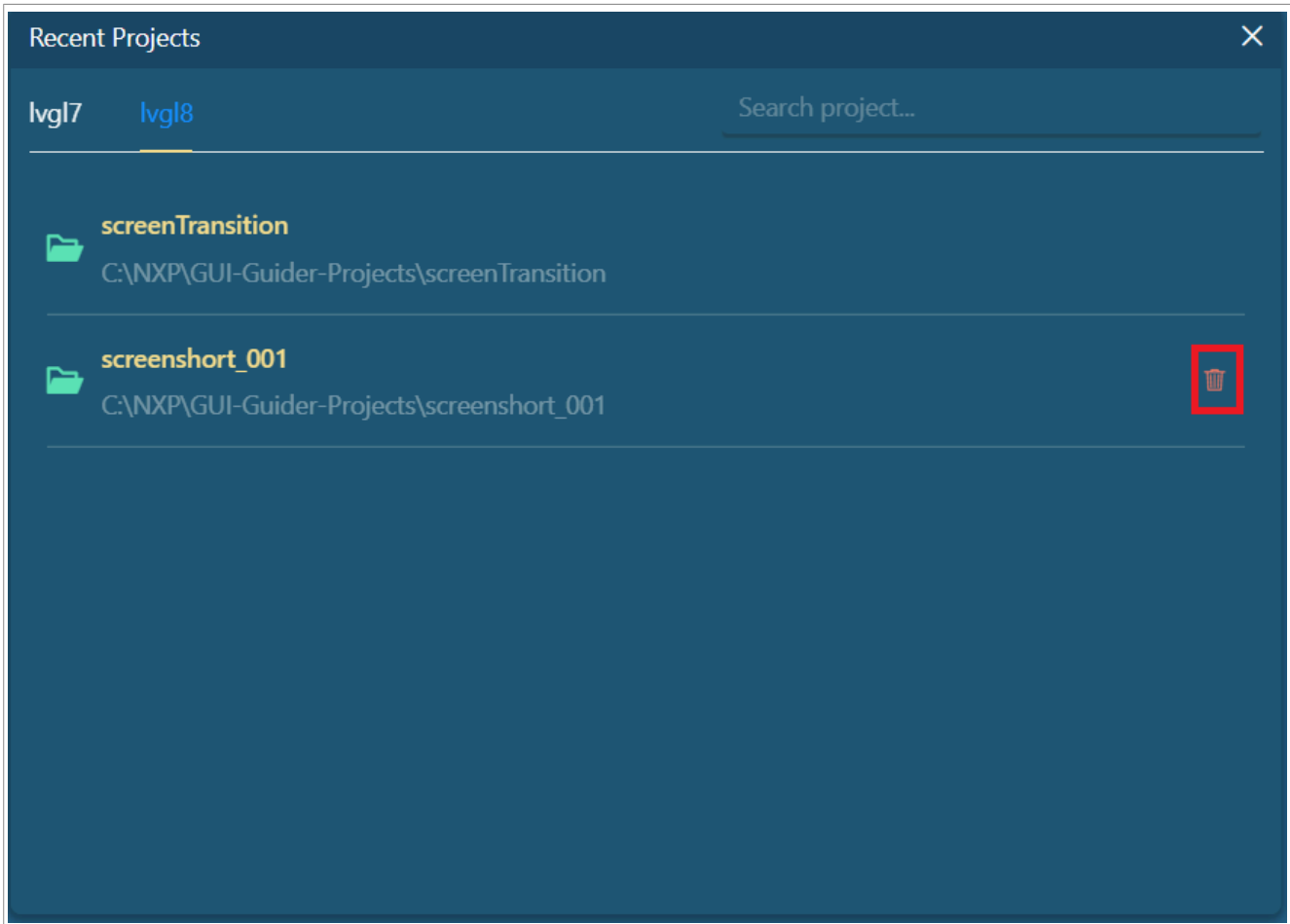


3.4 Delete a project


To delete a project, perform the following steps:

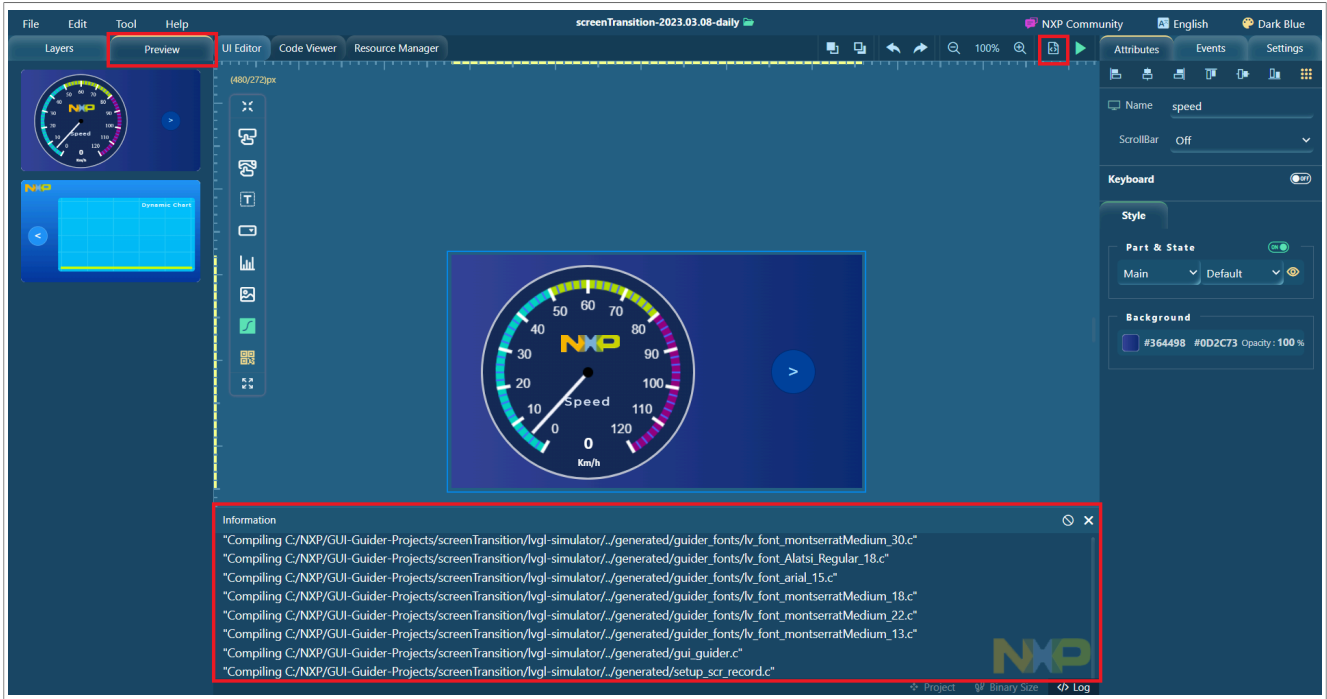
1. Exit GUI Guider IDE.
2. Delete the project folder from local file system if the project is not needed.
3. Open the GUI Guider IDE.
4. Select the **Open a recent project** button.

5. Click the delete icon corresponding to the project you want to delete.



3.5 Generated code

To generate source code of GUI project, click the  icon in the right-upper of edit window. It is possible to generate both the C and Python code automatically.

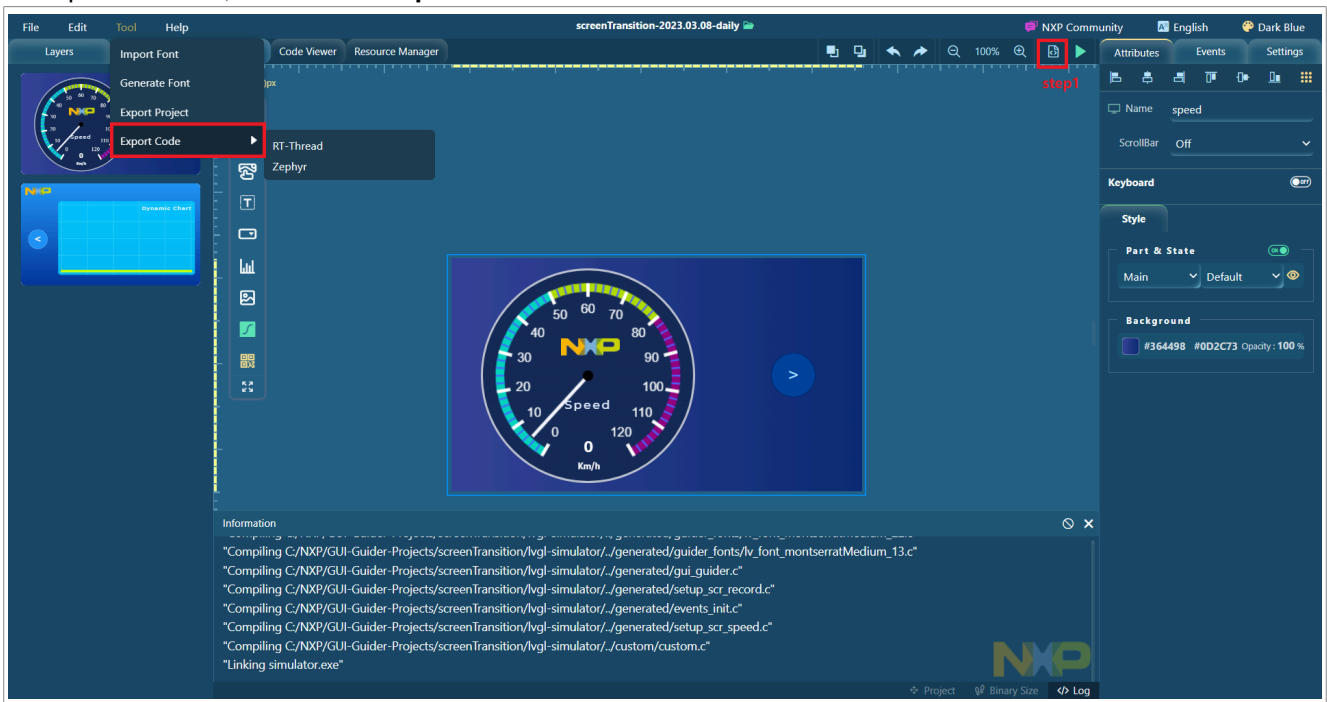


3.6 Export Code

To export the source code to a destination folder, the destination folder should be specified for the first time. The IDE remembers the path for future export.

Note: Before exporting the code, first generate it.

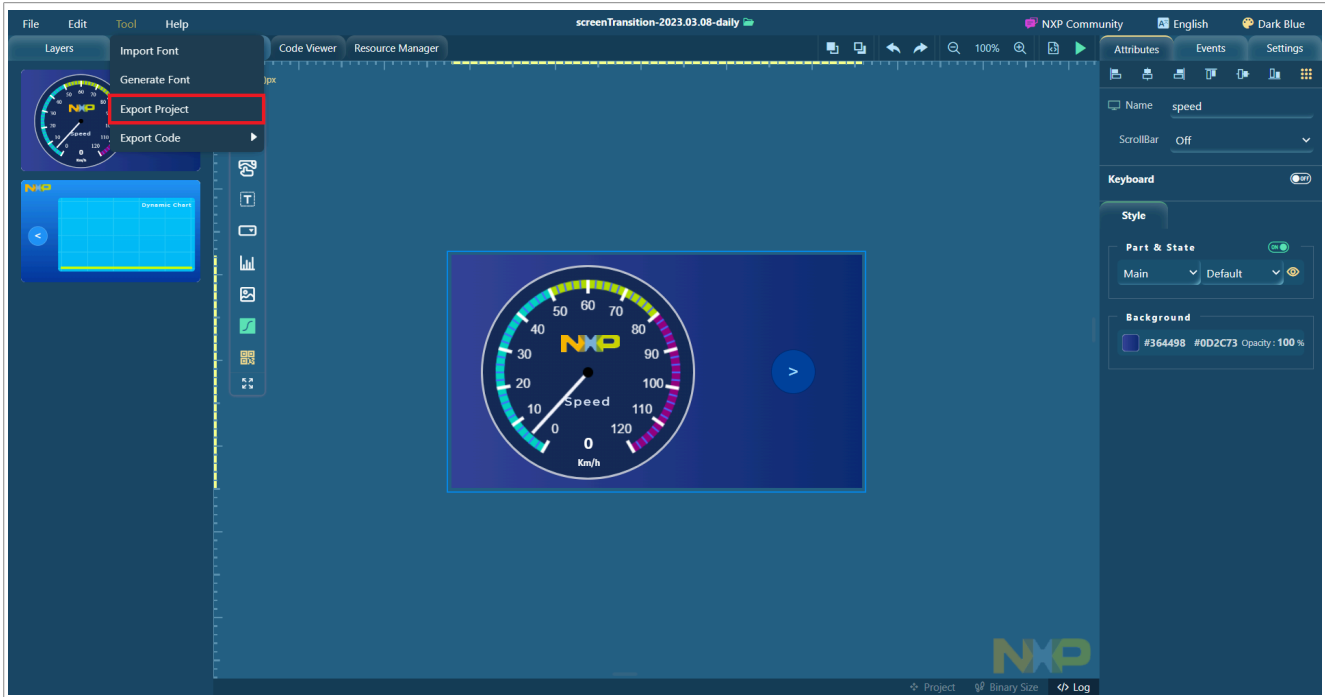
To export the code, click **Tool > Export Code**.



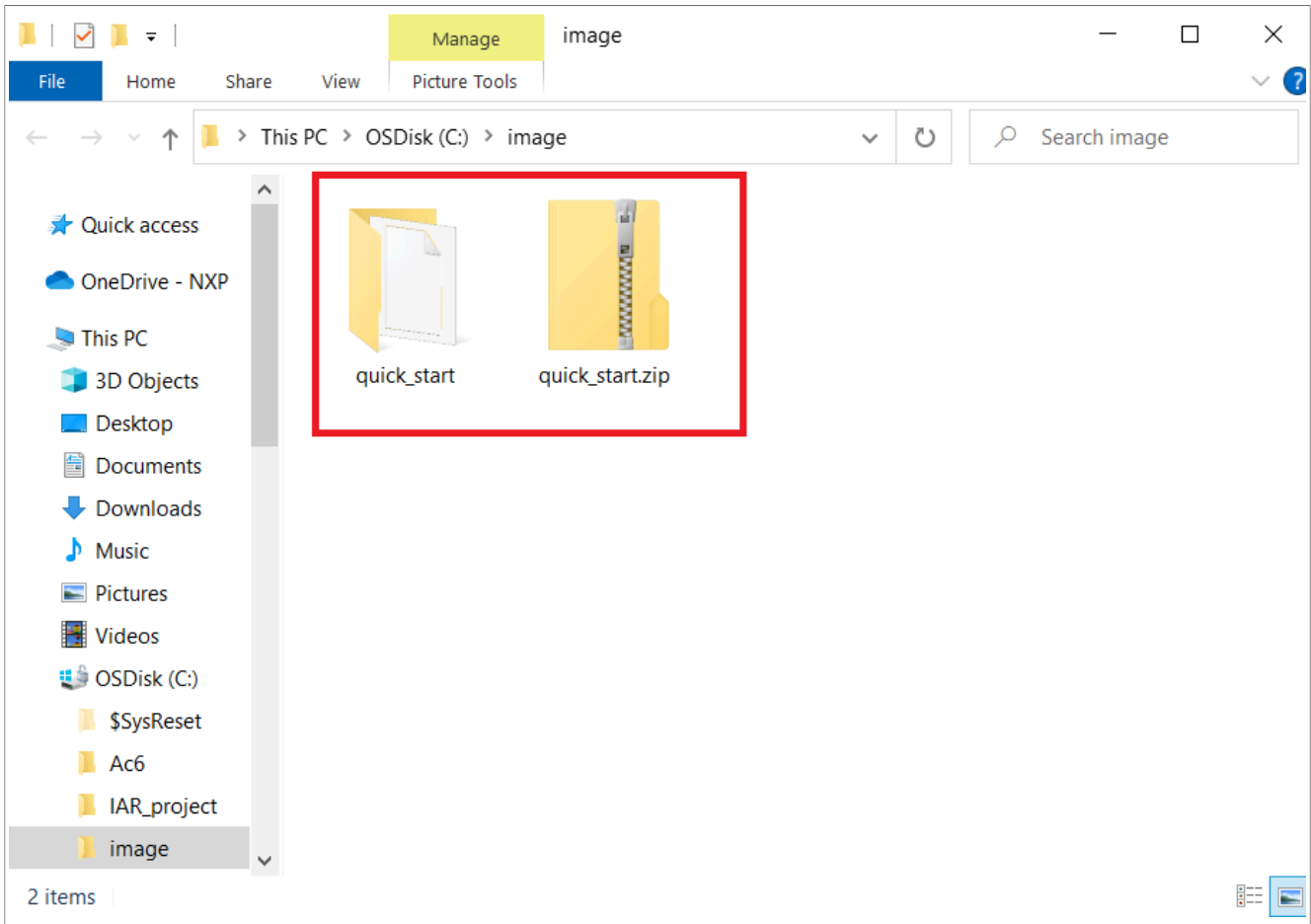
3.7 Export project

To share the GUI Guider project more conveniently, we have added the export project function. The IDE remembers the export path which is a common path in the export code function.

To export the project, click **Tool > Export project**.



The output of the export is the condensed project directory and a compressed file named <projectname>.zip. This file contains custom code, project resources, and UI configure files.



3.8 Import fonts

To import fonts, perform the following steps:

Note: GUI Guider supports TTF and WOFF font type.

1. Select **File > Import Fonts**.
The **Import Font File** dialog box appears.
2. Navigate to the folder where the font that you want to import is located.
3. Select the font.
4. Click **Open**.

Note: The extended fonts are provided at the following location:

- Windows: `${GG install path}\environment\extended_fonts`
- Linux: `/opt/Gui-Guider/environment/extended_fonts`
- Darwin: `/Applications/Gui-Guider.app/Contents/environment/extended_fonts`

Note: The font imported for the current project is available for all widgets of the project.

3.9 Generate fonts

The generated font file is stored in the `<project_name>\generated\guider_customer_fonts` folder. The purpose is to add new characters which otherwise are supported by the selected font type and size. The function is used for non-English languages. For example, Chinese.

To generate fonts, perform the following steps:

1. Select **File > Generate Fonts**.

The **Generate Fonts** dialog box appears.

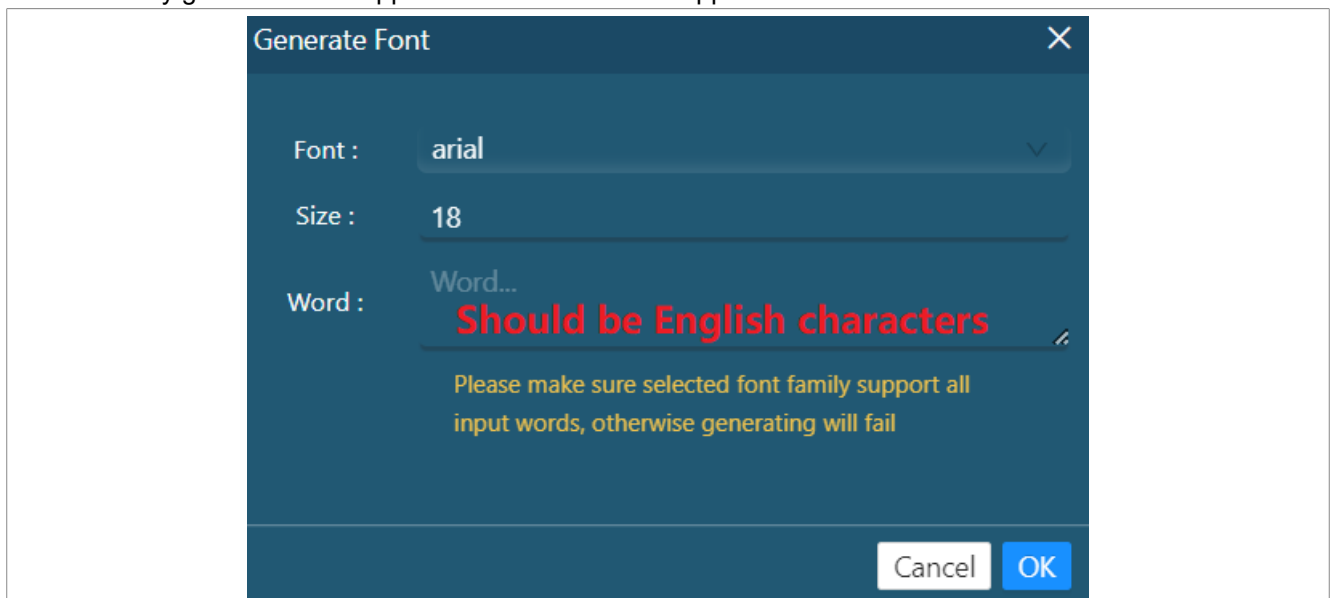
2. Select the font family and size. Ensure that the font family is for the English language.

The following image is an example of fonts generated for Chinese.



3. Click **Submit**.

The newly generated font appears normal in the GUI application.



The function provides an API to convert fonts to a C array. The C array file is generated in the generated \guider_customer_fonts folder.

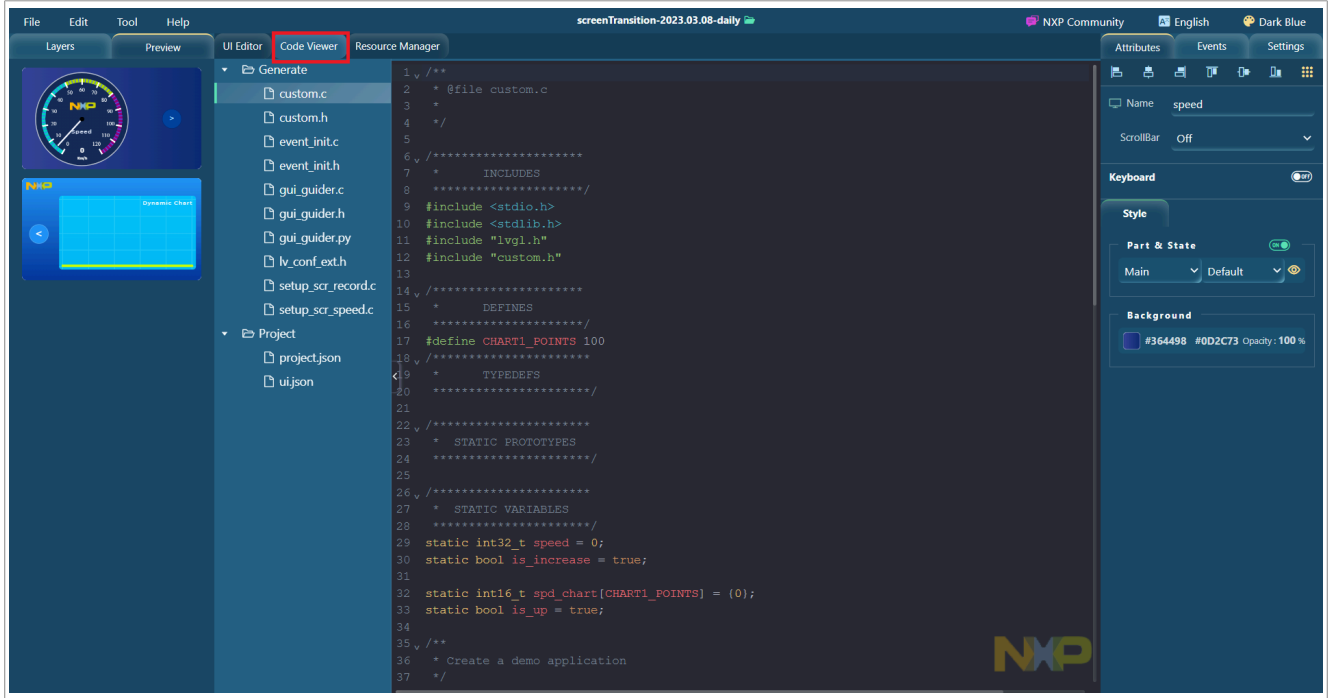
The following is the example code of using the generated font.

```
#include "lv_font.h"
LV_FONT_DECLARE(lv_font_simsun_12)
lv_style_set_text_font(&style_screen_ddlist1_selected,
LV_STATE_DEFAULT, &lv_font_simsun_12);
```



3.10 Code view

The source code generated by GUI Guider appears in the **Code Viewer** tab. The navigator is on the left side of the code viewer and switches to the source file that you want to view.

Note: *Ensure to generate the code before using the code viewer.*



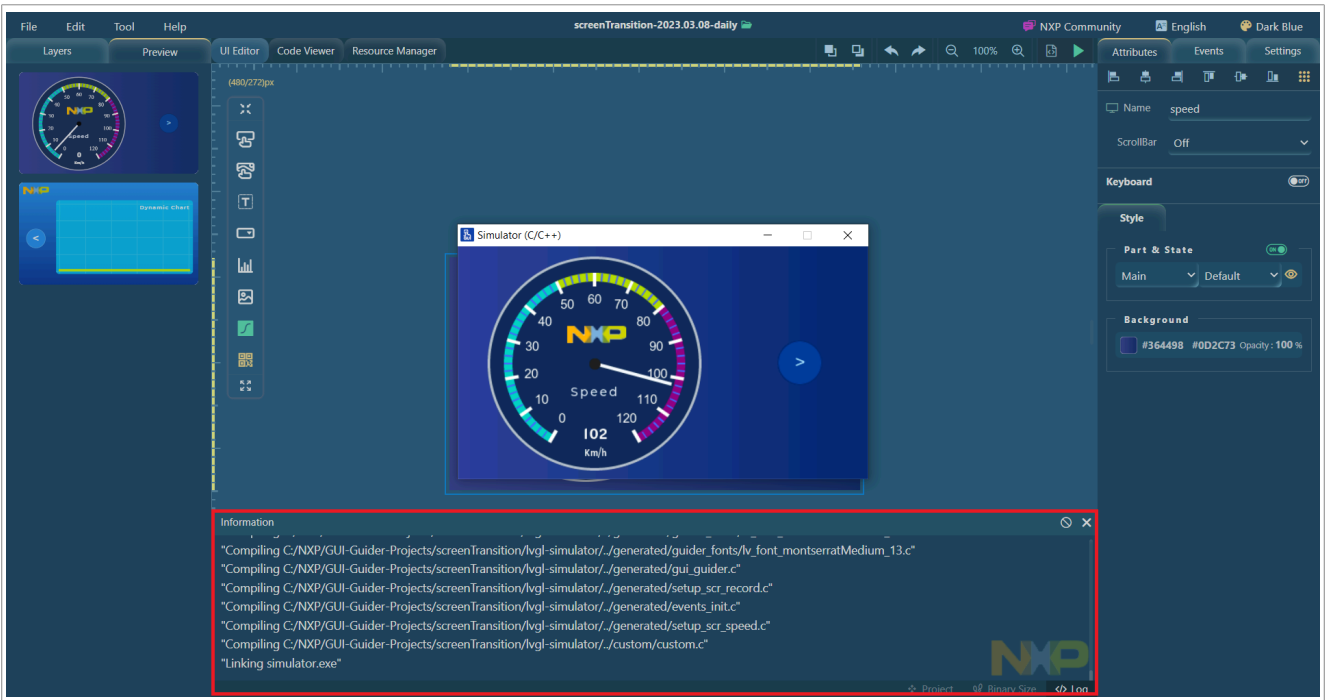
3.11 Run simulator

Both the C simulator and the MicroPython simulator are supported. To select a simulator and run it in the GUI application, click the  icon.



The simulator opens in a separate window.

Note: When the simulator is launched, the **Generate Code**, **Run Simulator**, and **Run Target** options are disabled until the simulator window is closed. You can use the mouse or the keyboard to interact with the GUI elements in the simulator.



Note: The GUI Guider main window changes to modal state when **Run simulator** is clicked. MicroPython is not supported for lvgl v7.

3.12 Run target

GUI Guider supports one-key build and deploy image on target board. GUI Guider also supports three toolchains: MCUXpresso, IAR, and Keil. Ensure that the corresponding IDE is installed on your host machine. [Table 1](#) provides information on the supported toolchain.

Table 1. Supported toolchain

Toolchain	Version	Support OS	Connector
IAR	9.30.1	Win10	USB
MCUXpresso IDE	11.7.0	Win10, OSX11, and Ubuntu 22.04	USB
Keil MDK	5.37	Win10	USB

The following prerequisites must be met to run the target successfully:

- Boards with CMSIS-DAP/mbed/DAPLink interface.
- For LPCXpresso boards, install the DFU jumper for the debug probe.
- Connect the development platform to your PC via USB cable.

[Figure 1](#) shows the window of log, project information, and memory monitor.

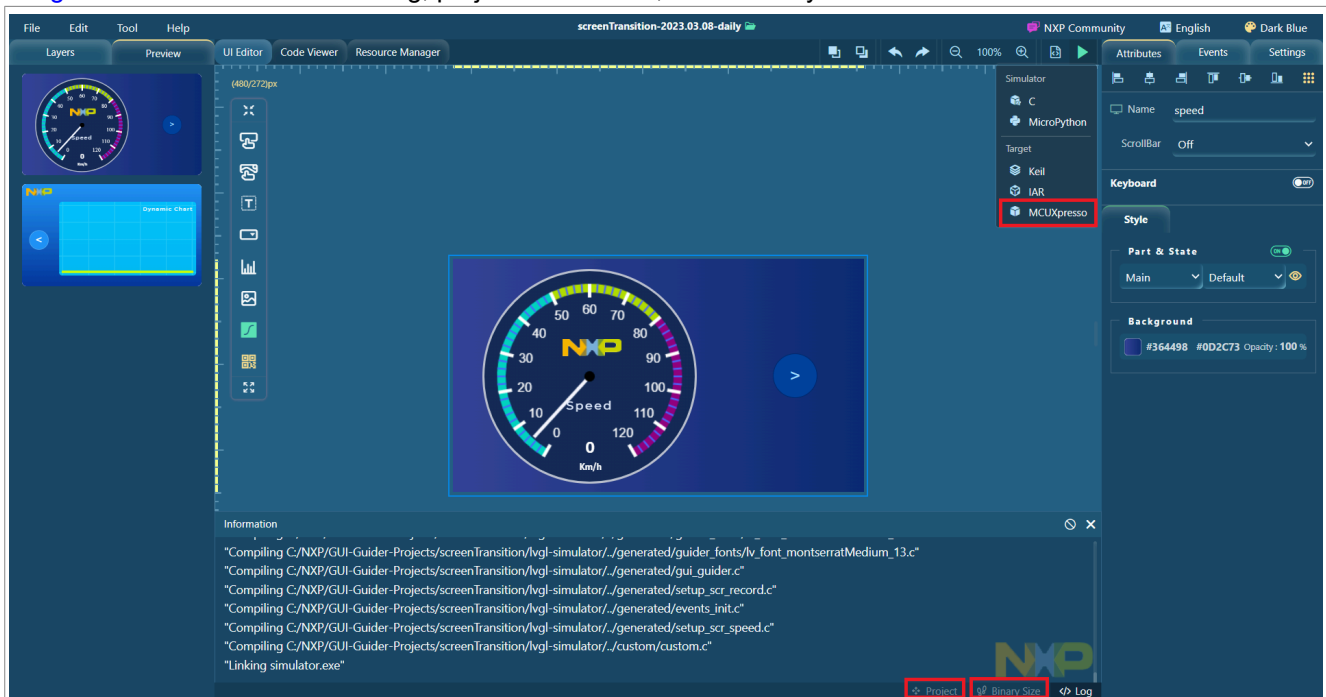




Figure 1. Project log, information, and memory monitor

Typically:

- The flash consumed by the GUI application is text + data.
- The RAM consumed by the application is data + bss.

Note:

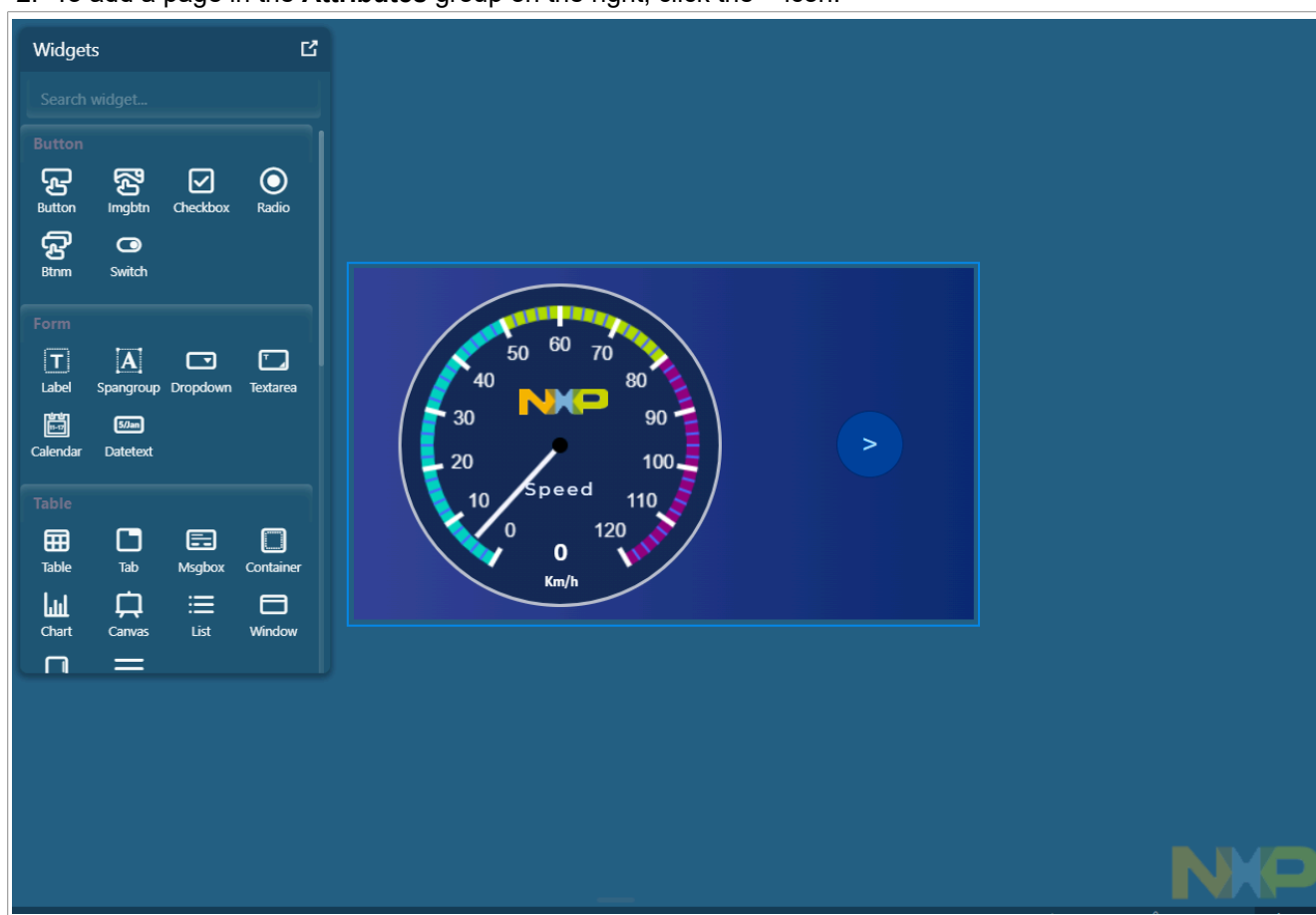
- Only MCUXpresso IDE supports memory display.
- The project does not support "Run Target" when simulator is selected as board template.

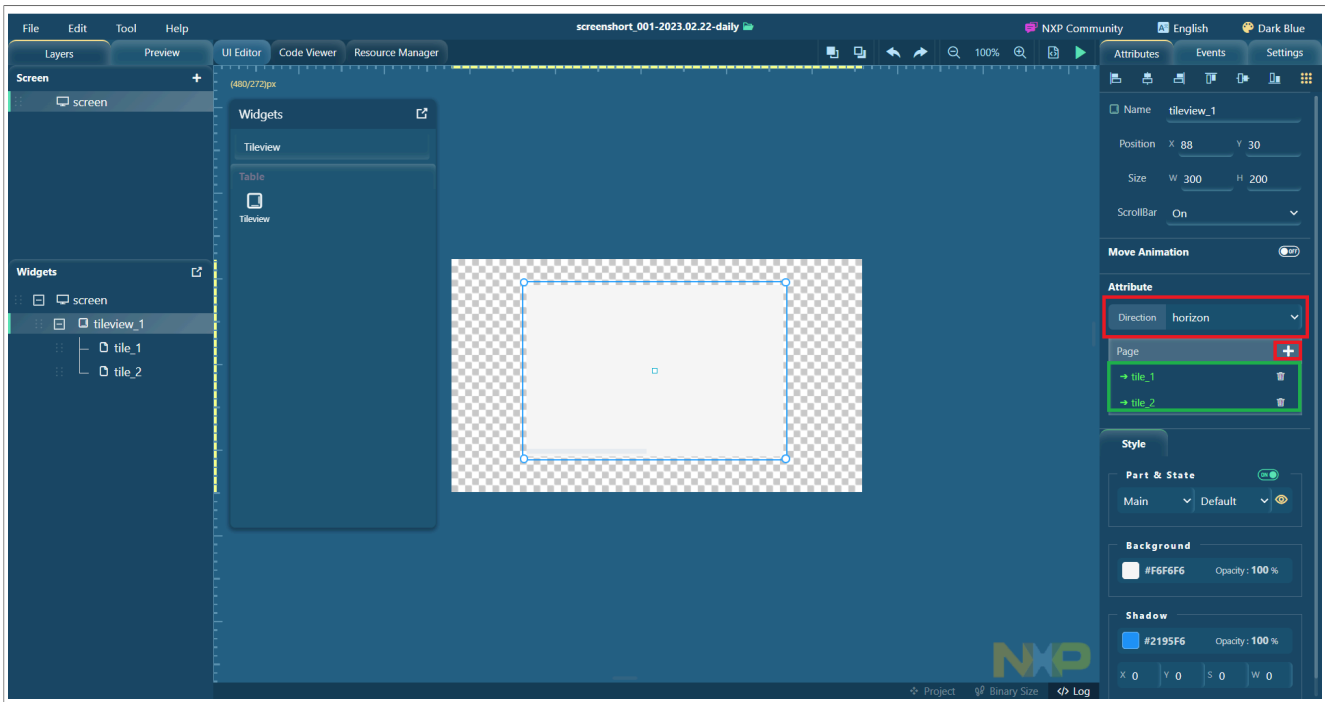
3.13 Tileview usage

Tileview is implemented as a standard widget in GUI Guider. You can design the GUI in tileview by drag and drop operation.

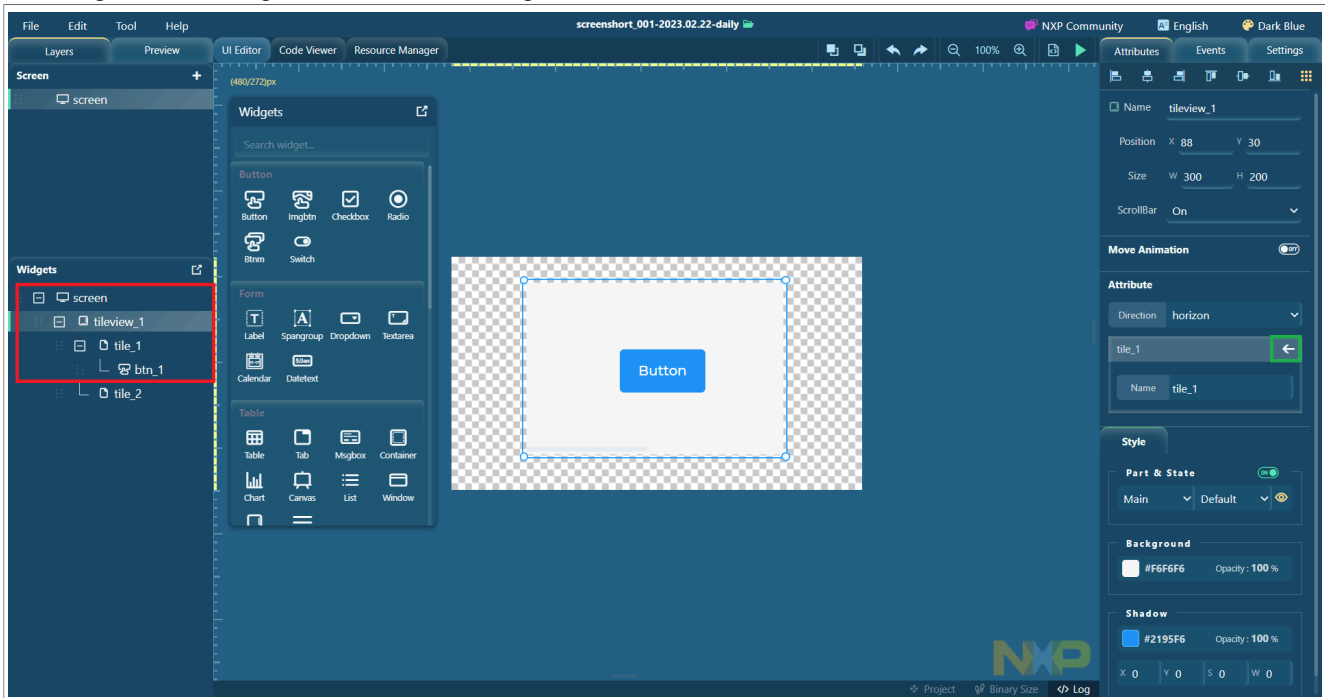
To use the **tileview** widget, perform the following steps.

1. Drag the tileview widget to the editor.
Note: If you are unable to find the widget, type the name of the widget in the search field and press Enter. The widget name appears in the search results.
2. To add a page in the **Attributes** group on the right, click the + icon.

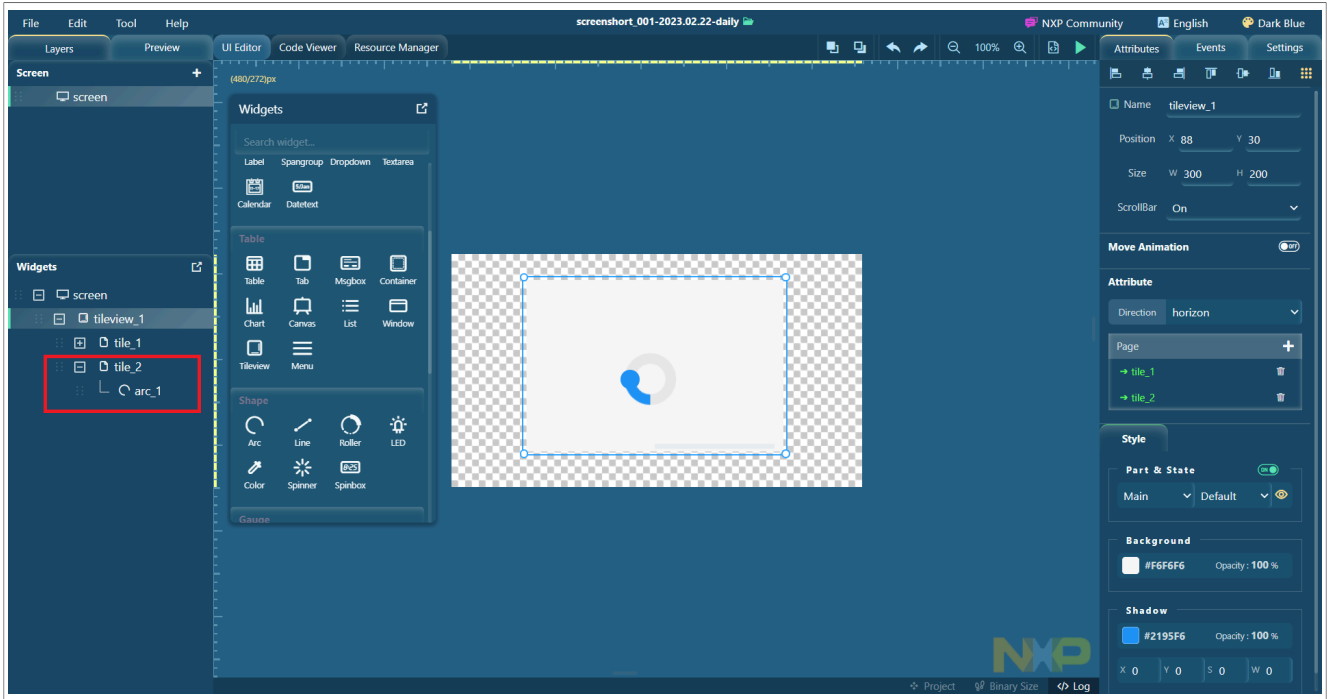




3. Select the **tile_1** tab.
4. Drag a button widget to the **tileview** widget.



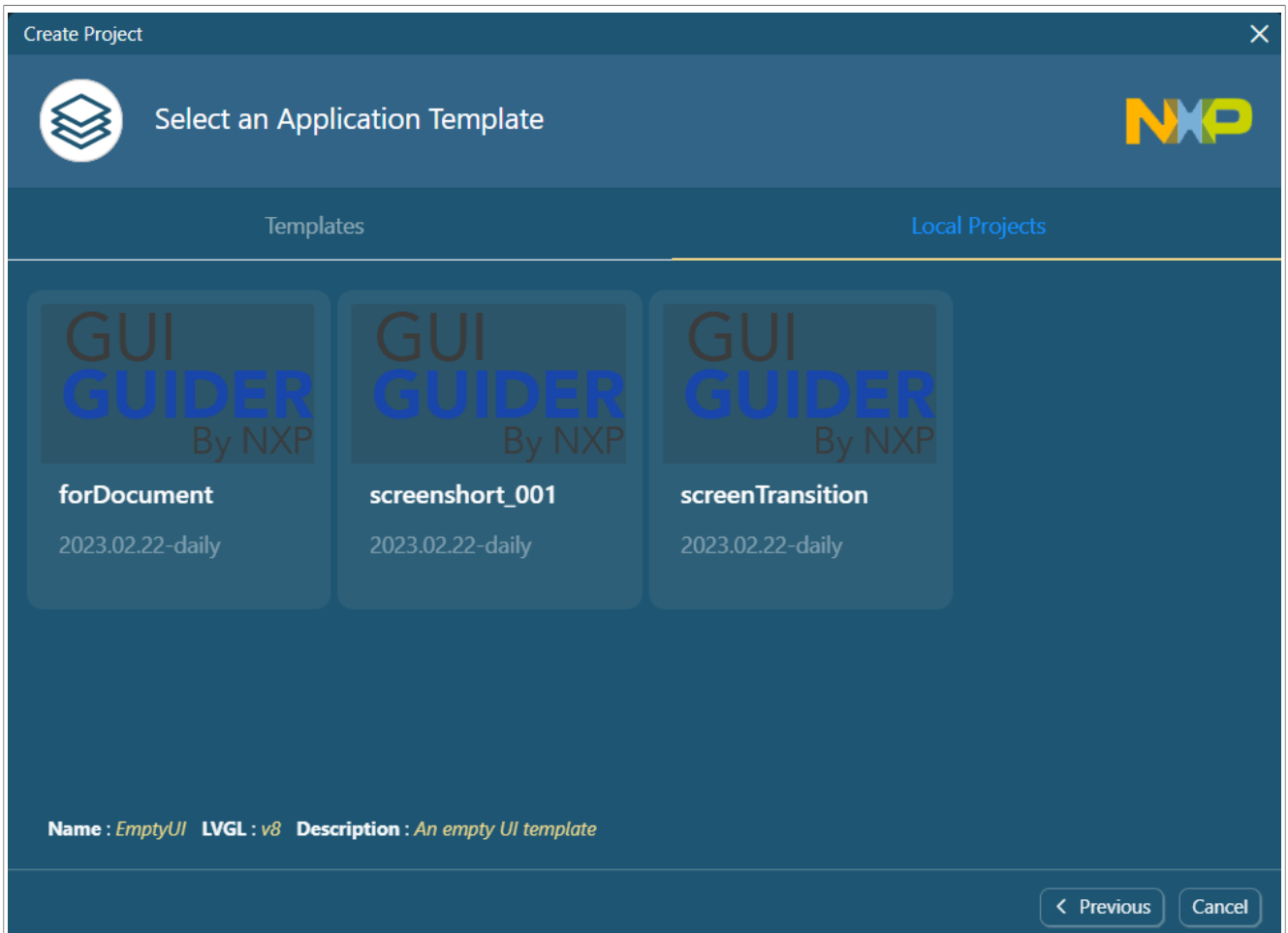
5. Select the **tile_2** tab.
6. Drag an **arc** widget to the tileview widget.

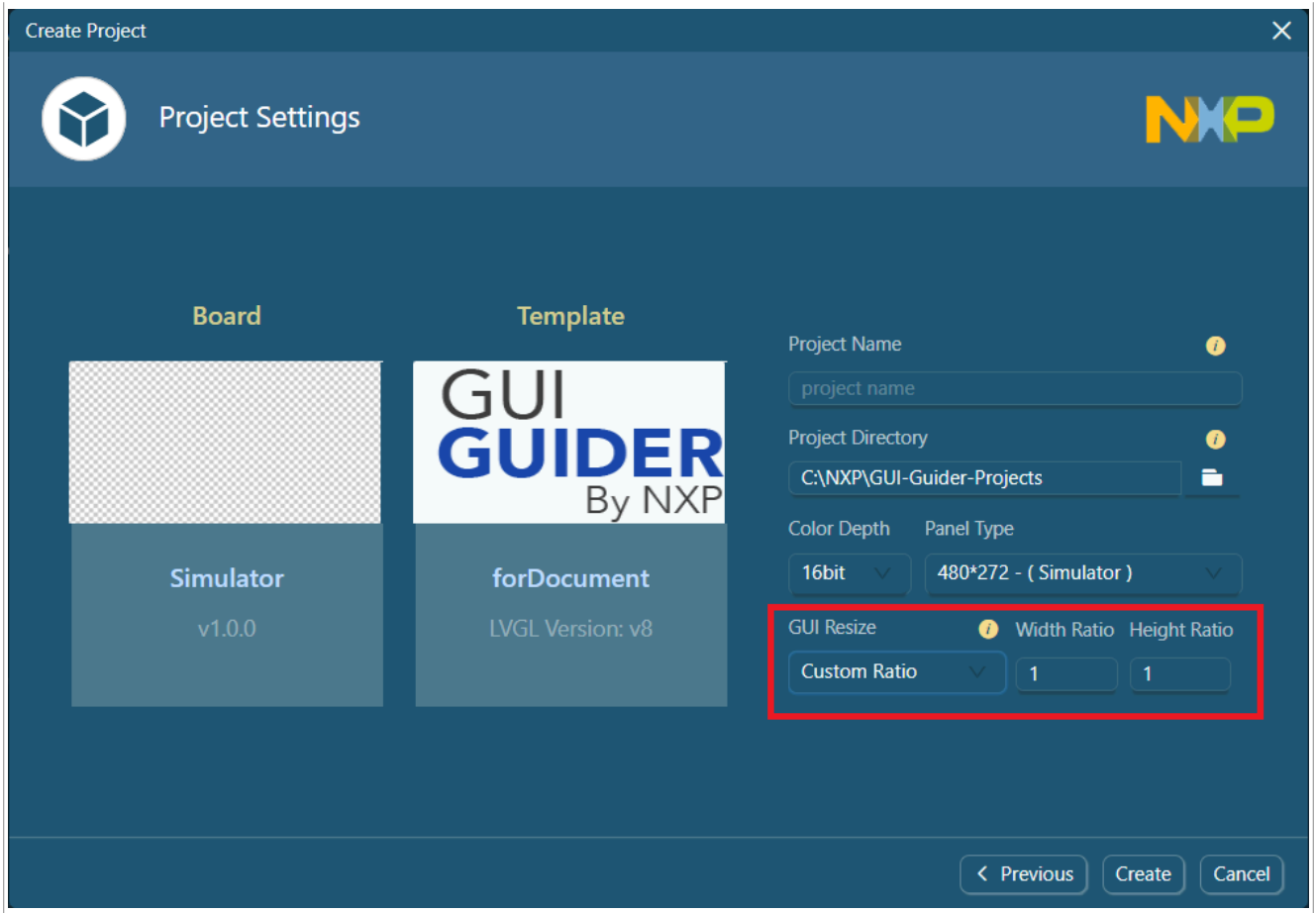


3.14 GUI auto-scaling

The auto-scaling function in the GUI Guider application, resizes the GUI to adapt the display of a different size. The auto-scaling function is useful when you want to reuse an application designed based on a particular display size. The function can support new project based on application template and local project.

Note: If there is a hard-coded size in the custom code, the position and the size-related code must be adjusted manually in the custom code.





3.15 Event window

[Table 2](#) lists the options in the **Events** window.

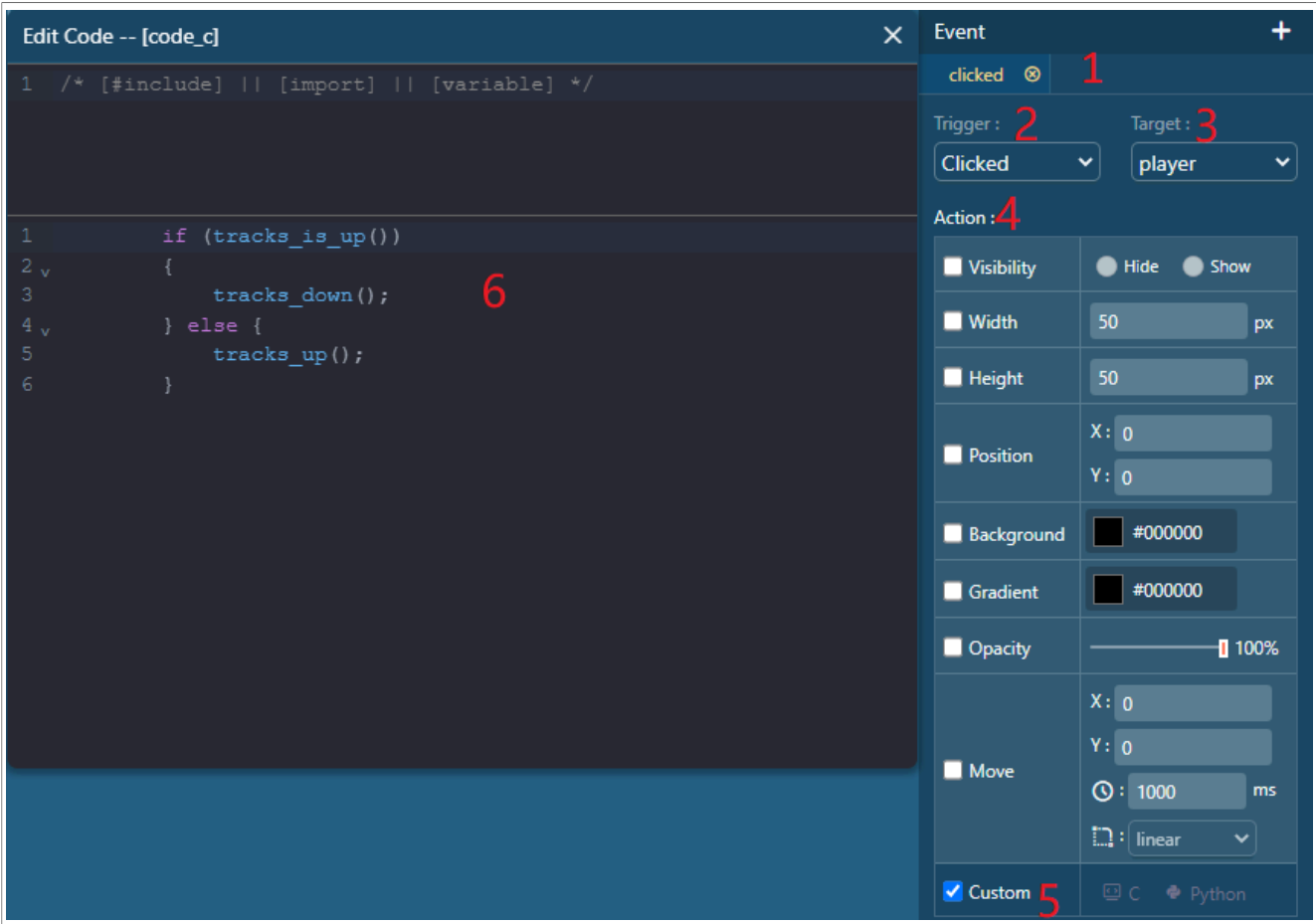


Table 2. Events window

Label	Description
1	Add event of this widget
2	Select trigger
3	Target widget (widget to be acted upon)
4	Select action (support choose multiple)
5	Select custom code function
6	Edit your custom code

3.16 Resource window

[Table 3](#) lists the options in the **Resource** window.

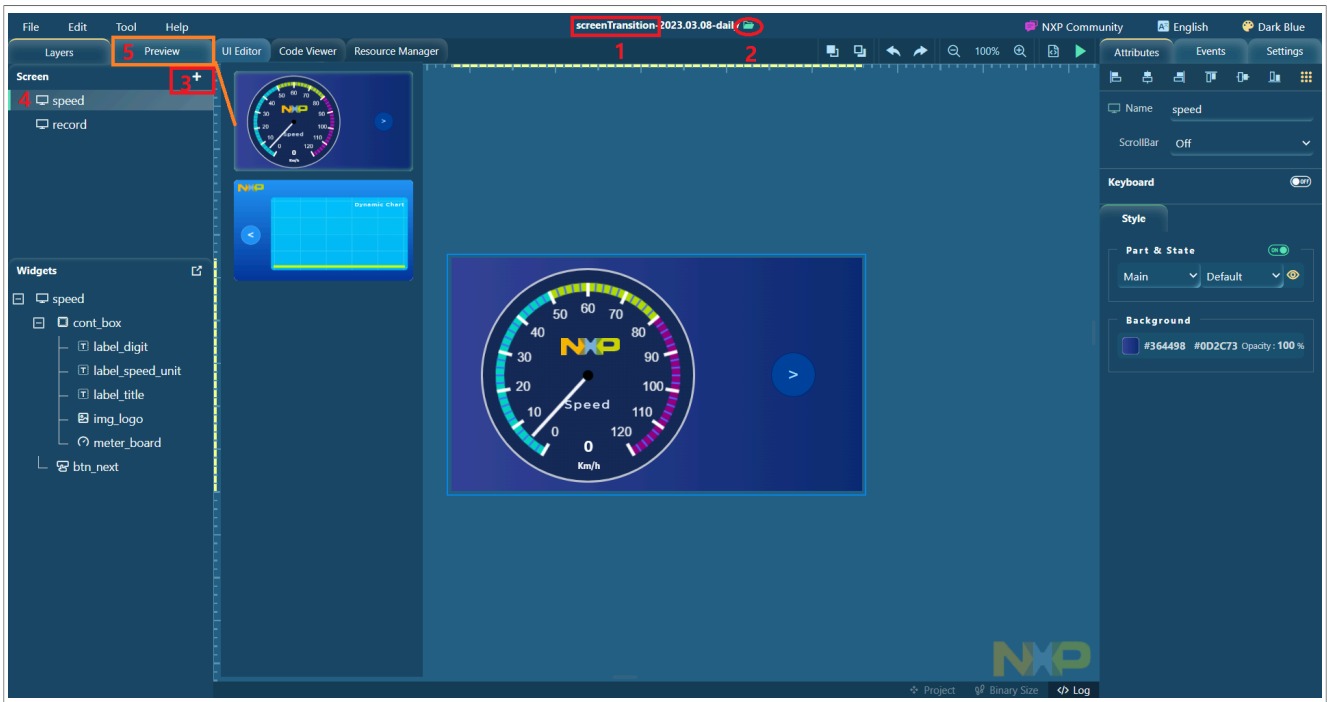


Table 3. Resource window

Label	Description
1	The name of current project
2	Open the project folder with File Explorer
3	Add a new screen
4	The name of current screen
5	Go to preview mode

3.17 Import resource

[Table 4](#) lists the options in the **Import** resource window.

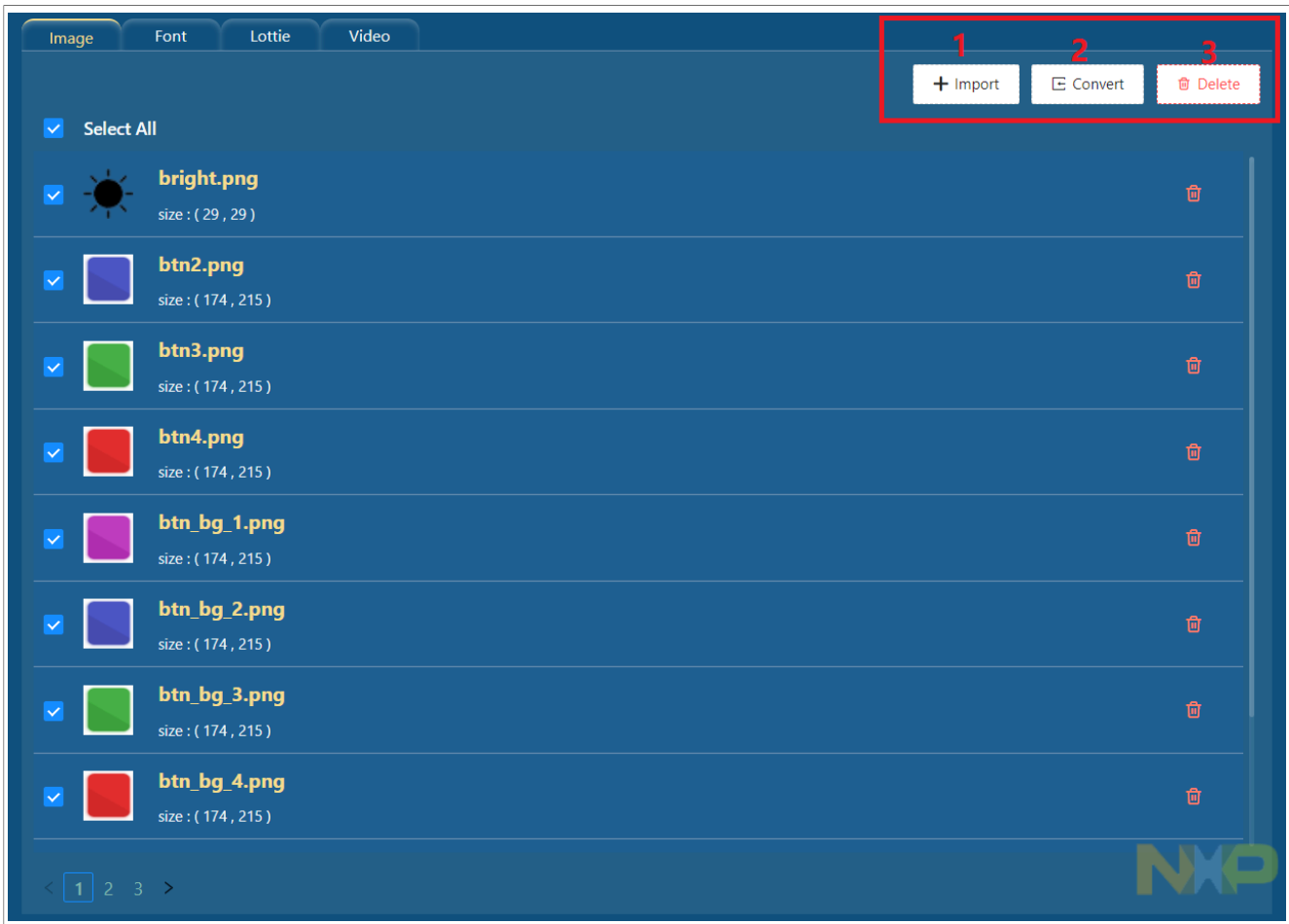


Table 4. Import resource

Label	Description
1	Import image button
2	Convert images to bin file
3	Delete button

3.18 Shortcut function

[Table 5](#) lists the shortcut functions.

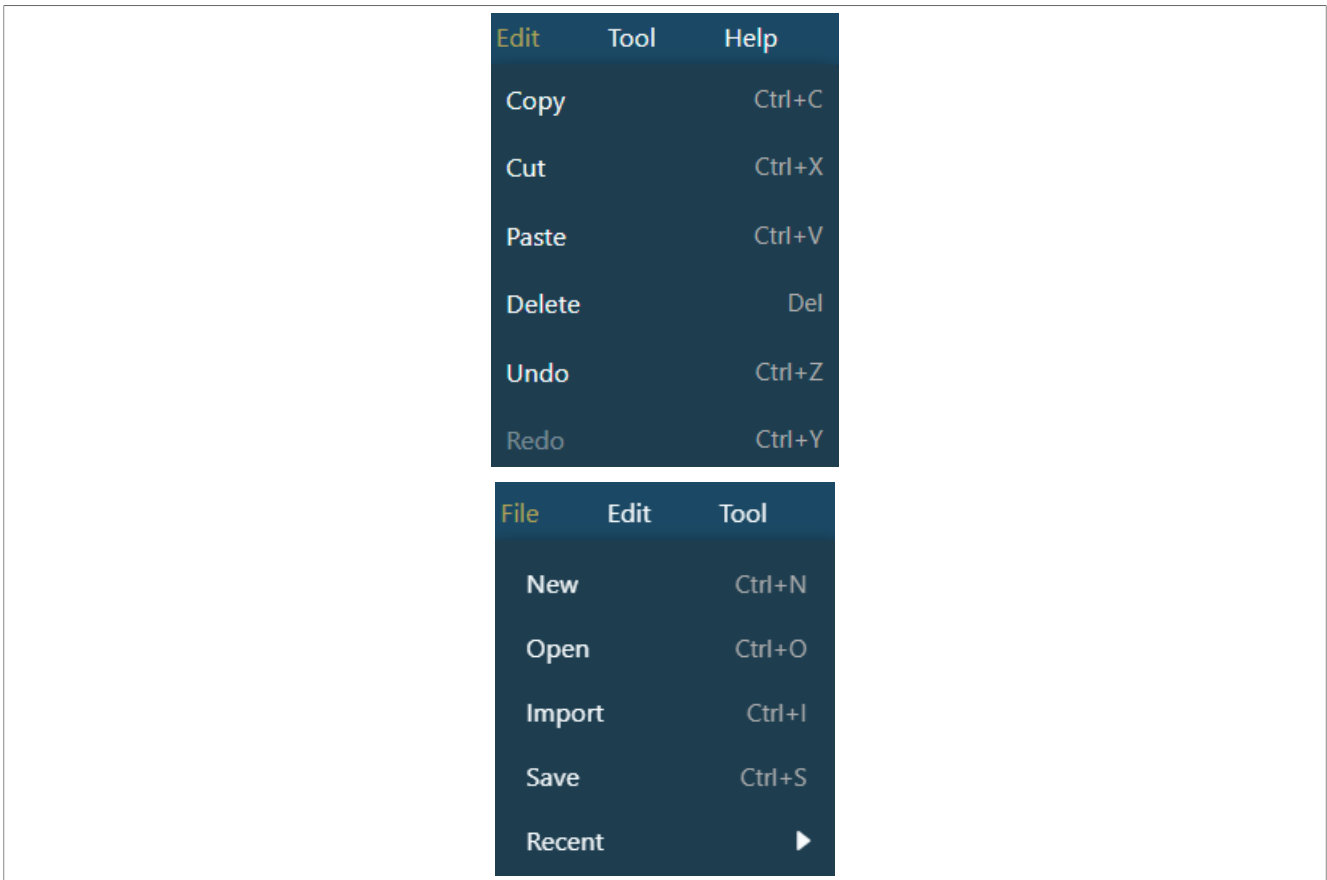
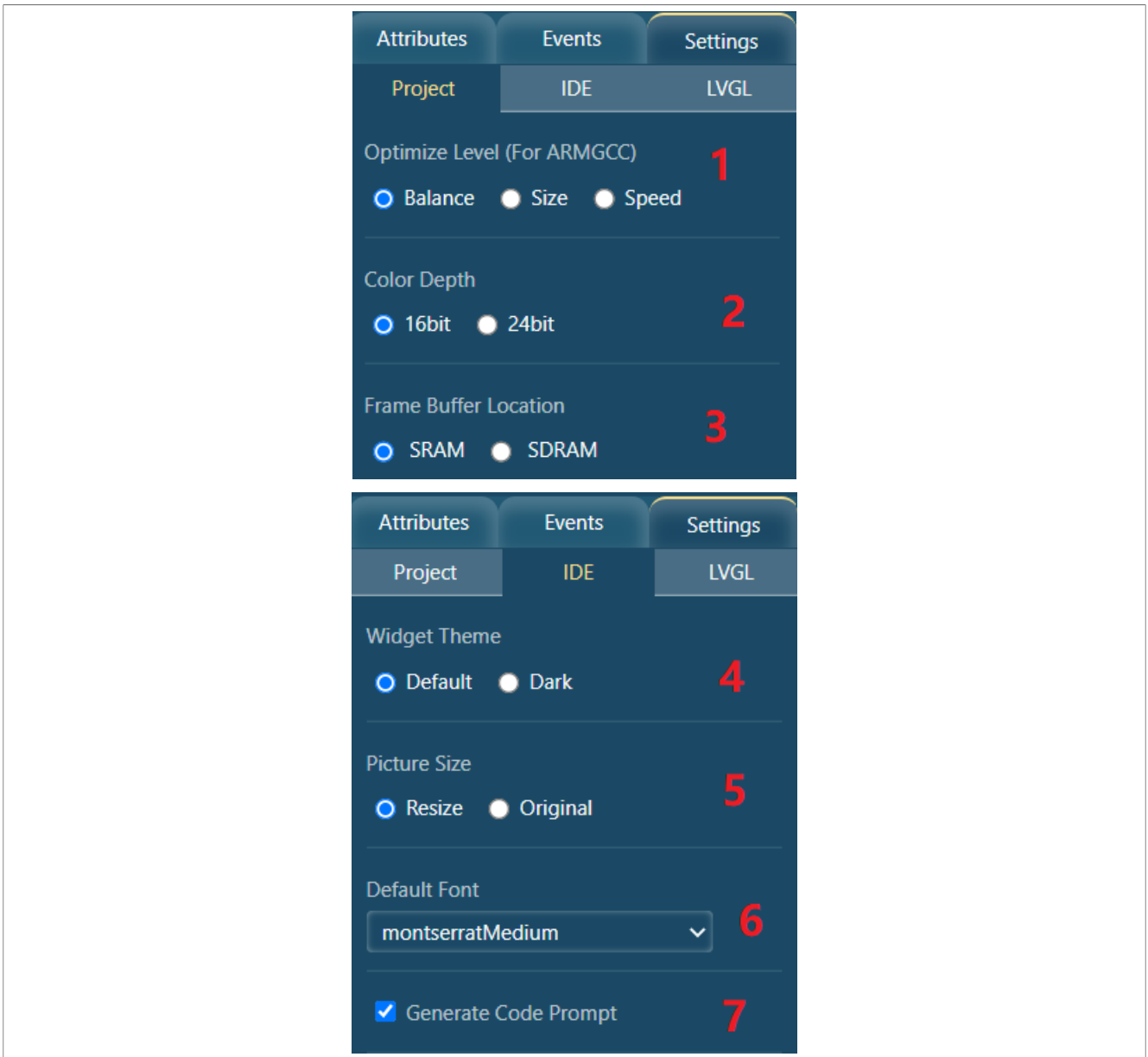


Table 5. Shortcut functions

Function Name	Shortcut
New project	Ctrl + N
Open project	Ctrl + O
Import project	Ctrl + I
Save	Ctrl + S
Copy	Ctrl + C
Paste	Ctrl + V
Delete	Del
Undo	Ctrl + Z
Redo	Ctrl + Y

3.19 Project setting

[Table 6](#) lists the options in the **Project setting** window.



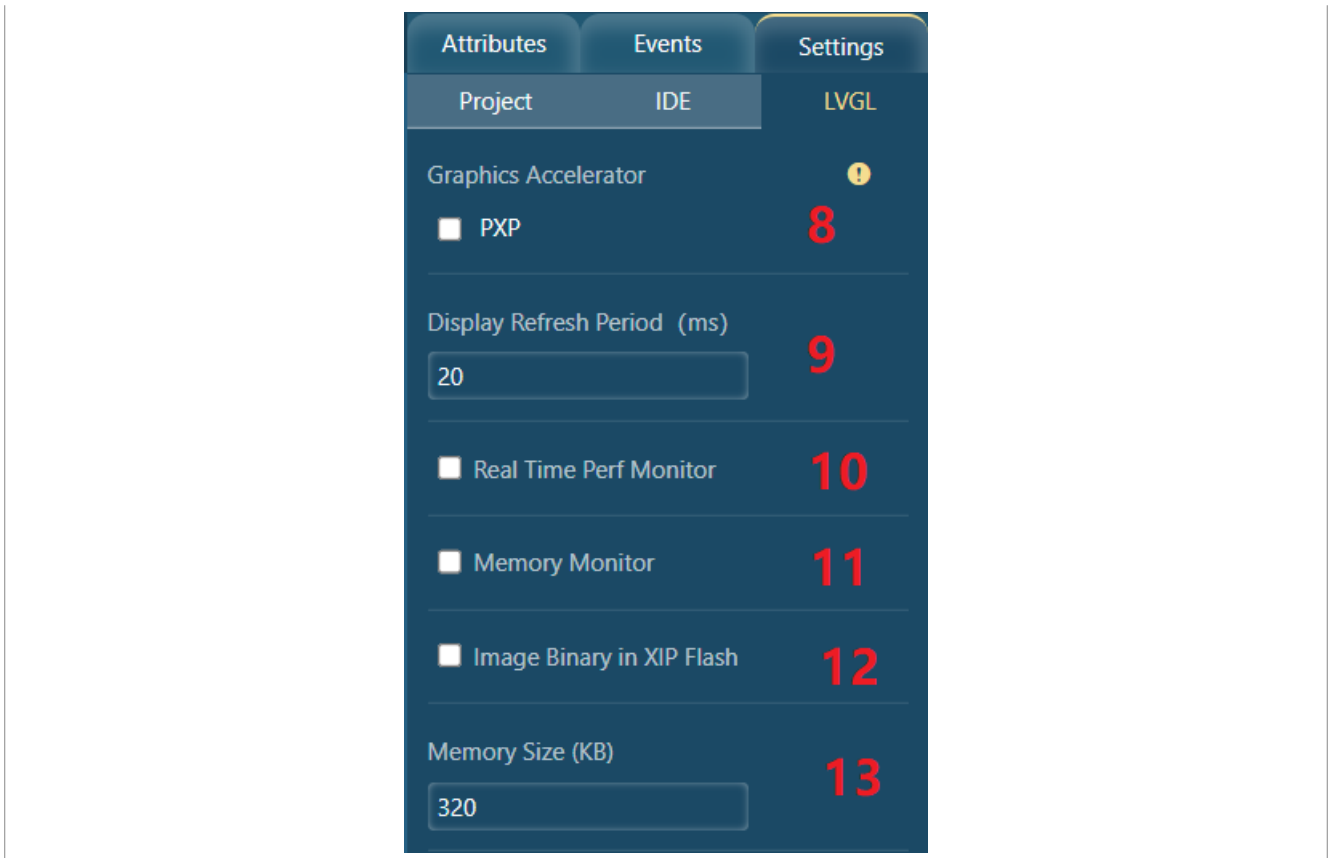


Table 6. Project setting window

Label	Description
1	Select optimization level (Only support NXP target)
2	Select color depth for display
3	Select frame buffer region (Only MIMXRT1062xxxxA support)
4	Select widget theme
5	Set the image widget default size
6	Set the project default font
7	Enable the generate code prompt
8	Enable PXP or VGLITE
9	Set the default display refresh period
10	Enable the real-time performance monitoring
11	Enable the real-time memory monitor
12	Set the image binary download base address
13	Set the size of memory allocated for LVGL application usage

3.20 IDE setting

[Table 7](#) lists the options in the **IDE setting** window.

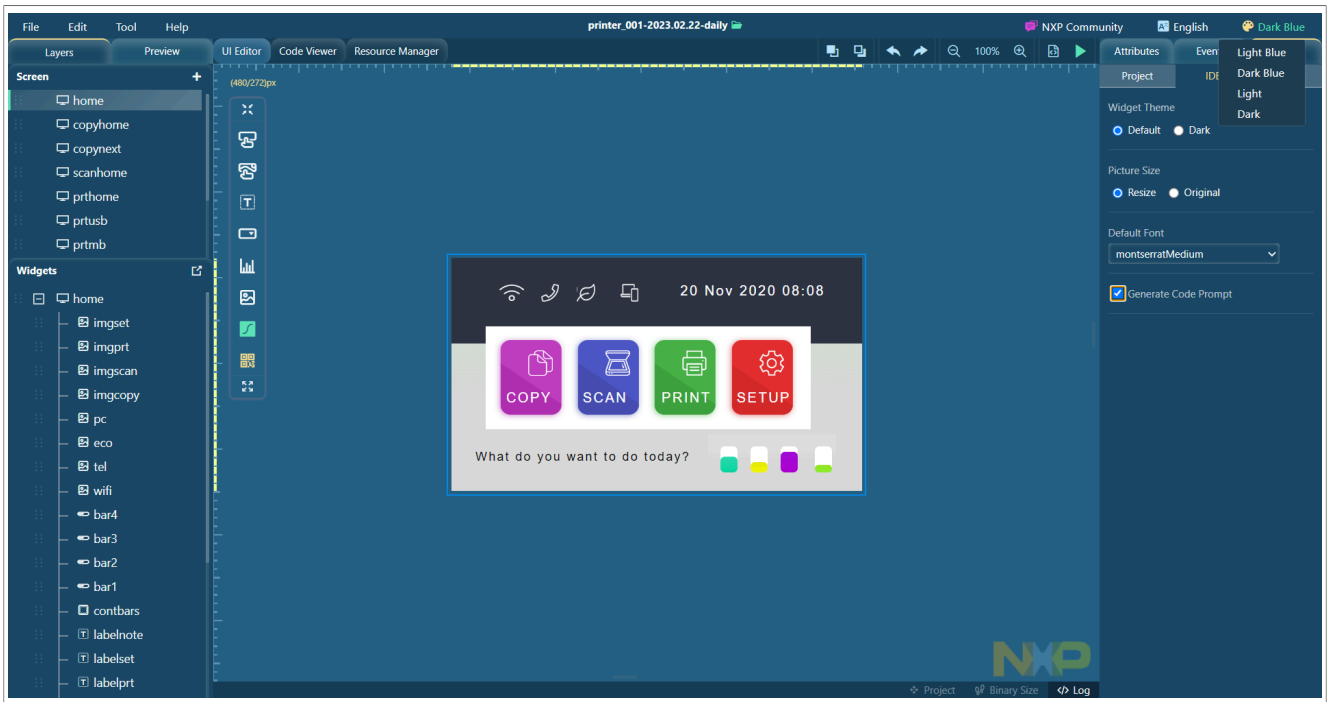


Table 7. IDE setting window

Item Name	Option
Language	English and Chinese
Theme	Light Blue, Dark Blue, Light, and Dark

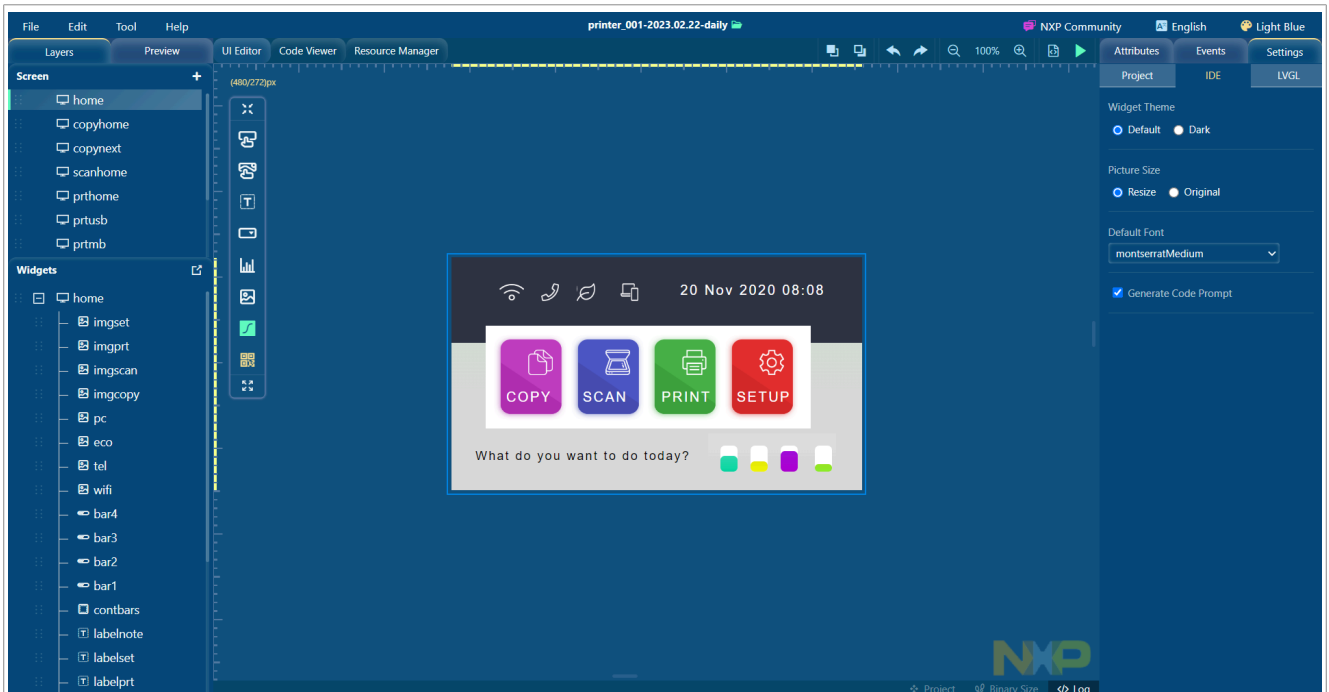


Figure 2. Light blue theme

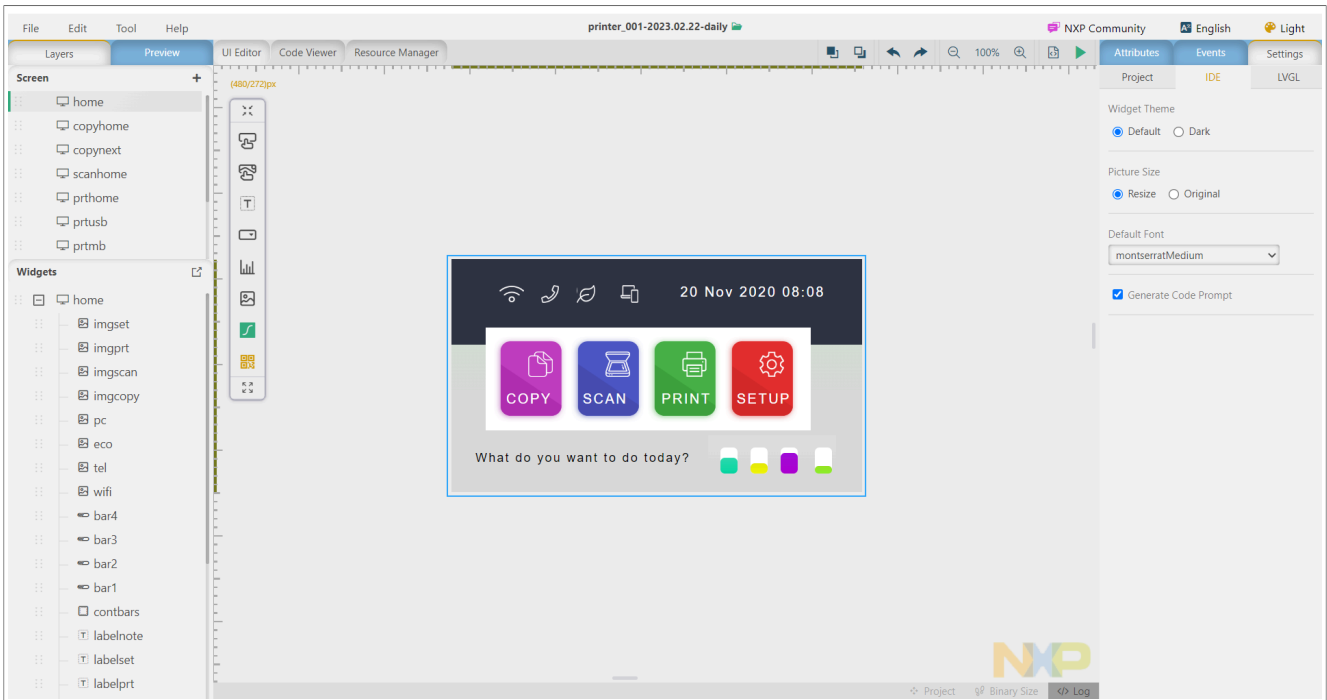


Figure 3. Light theme

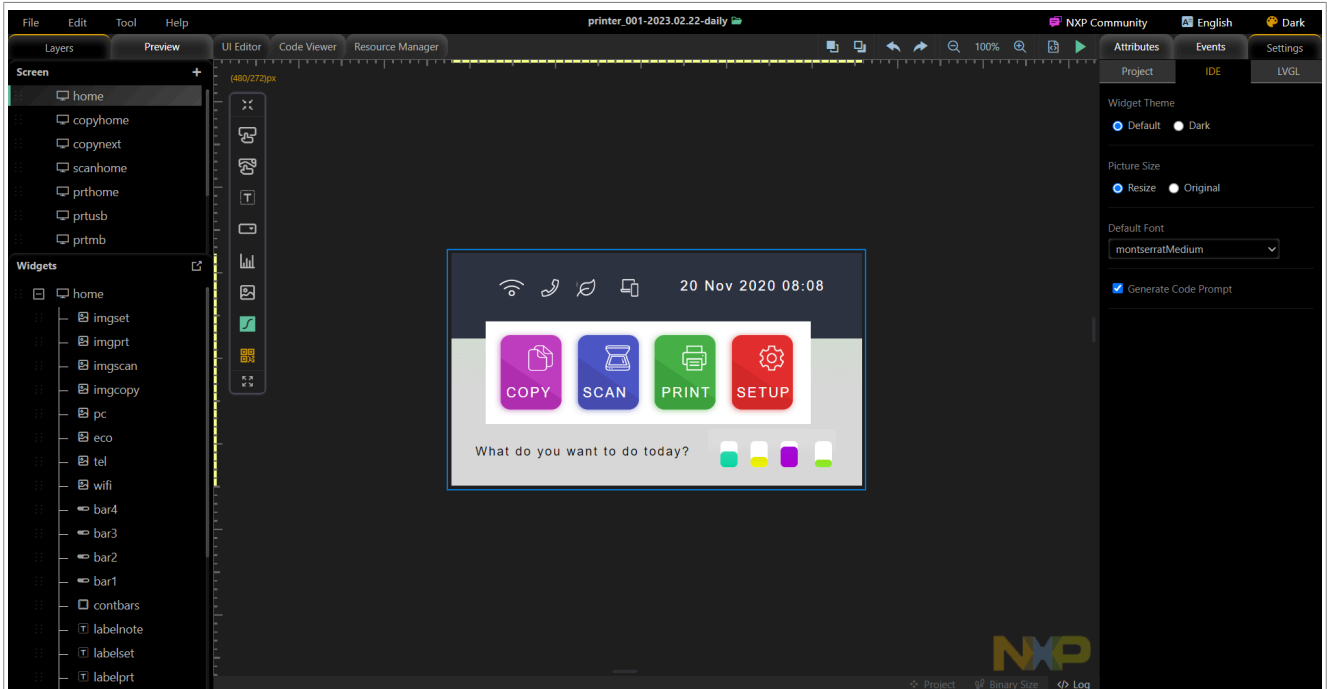


Figure 4. Dark theme

3.21 Screen setting

[Table 8](#) lists the options in the screen setting window.

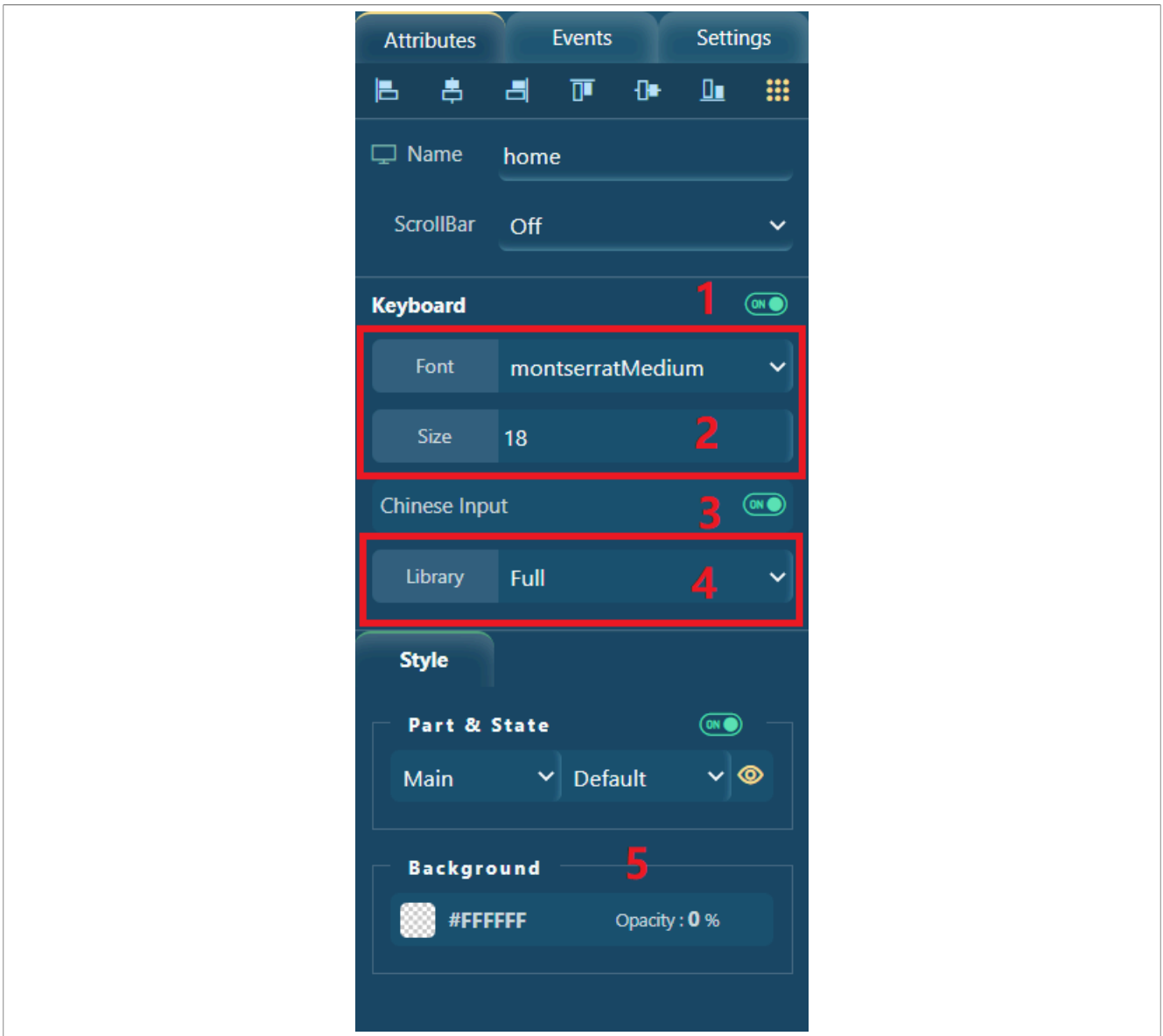


Table 8. Screen setting

Label	Description
1	Enable the keyboard for all textareas in the current screen
2	Set the keyboard font style
3	Set the Chinese input mode
4	Set the Chinese character set type
5	Set the screen background style

Note: The screen can also set events. Make sure that the Font family supports Chinese characters when using Chinese input.

4 Widget details

This section describes the GUI Guider widgets. The following are the common properties in all the widgets.

Table 9. Common properties

Property	v7	v8	Description
Name	General Setting > Name	General Setting > Name	Default name: Generated by the widget type and id. Must begin with a letter. Must be at least three characters long. Can only include letter, number, and underscore.
Position (x,y)	General Setting > Position	General Setting > Position	Position of widget, including x and y coordinate. x should be a value between 0 and width of screen. y should be a value between 0 and height of screen.
Size (W,H)	General Setting > Size	General Setting > Size	Size of widget. Set the width and height of the widget.
Scrollbar		General Setting > Scrollbar	The scrollbar option for widgets: OFF, ON, ACTIVE, AUTO
Shadow color		General Setting > Shadow color	Set the shadow color
Shadow opacity		General Setting > Shadow opacity	Set the shadow opacity
Shadow spread		General Setting > Shadow spread	Make the shadow calculation to use a larger or smaller rectangle as base
Shadow width		General Setting > Shadow width	Set the width of the shadow in pixels
Shadow position (x, y)		General Setting > Shadow position	Set an offset on the shadow in pixels

4.1 3Dimg

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



Figure 5. 3D animation widget in v7



Figure 6. 3D animation widget in v8

Table 10. 3Dimg properties

Property	v7	v8	Description
Image Path	Attribute > Image Path	Attribute > Image Path	Add the image files
Interval	Attribute > Interval	Attribute > Interval	Time between two frames
Repeat Count	Attribute > Repeat Count	Attribute > Repeat Count	Animation repeat time
Axis x, y, z	Attribute > Axis	Attribute > Axis	Rotation matrix over X,Y,Z Axis is between 0 and 360. X is the animation image horizontal rotation degree, Y is the animation image Vertical rotation degree, Z is the animation image Lateral rotation degree.
Frame number	Attribute > Frame number	Attribute > Frame number	The number of pictures converted and the value must be between 0 and 1000

4.2 Analog clock

The following are the details of the analog clock widget.

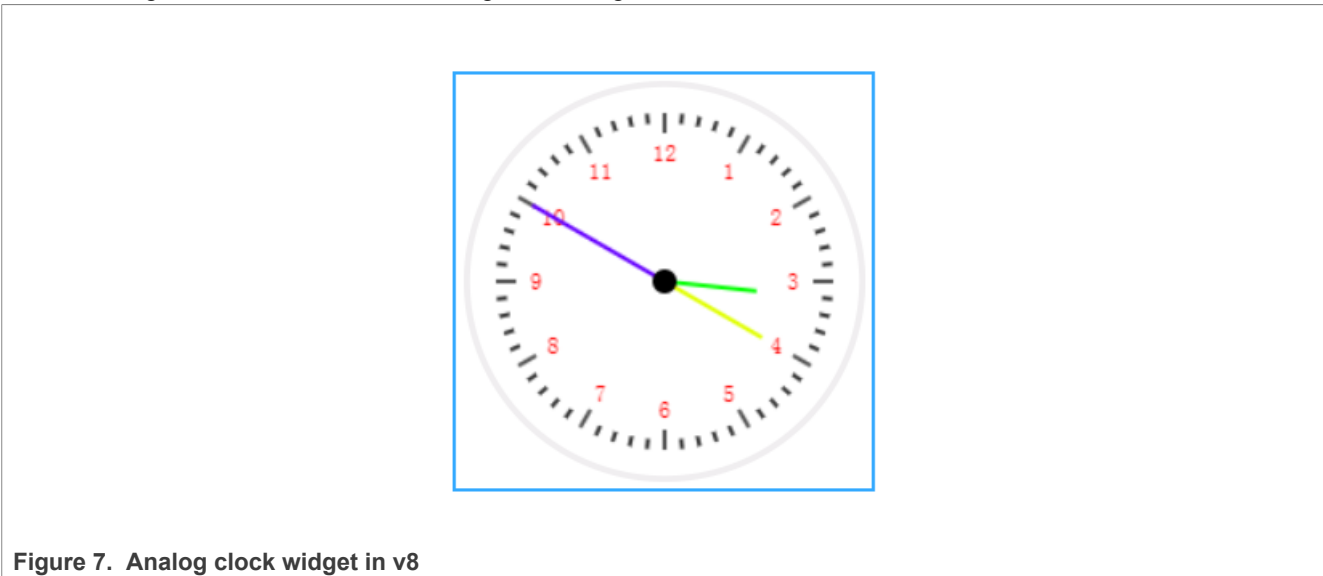


Figure 7. Analog clock widget in v8

Table 11. Analog clock properties

Property	v7	v8	Description
gap		Attribute > gap	

Table 11. Analog clock properties...continued

Property	v7	v8	Description
Hour Tick width & Length, color		Attribute > Hour Tick width & Length, color	Tick width is less than 10 and Tick length is less than 20
Minutes Tick width & Length, color		Attribute > Minutes Tick width & Length, color	Tick width is less than 10 and Tick length is less than 20
Hour, Minute, Second		Attribute > Hour, Minute, Second	
NeedleType & Value		Attribute > NeedleType & Value	
Needle width & length & Color		Attribute > Needle width & length & Color	Needle width is less than 10 and Needle length is less than 20
Image path		Attribute > Image Path	
X & Y, W & H		Attribute > X & Y, W & H	Needle width is less than 10 and Needle length is less than 20
State		Part and State > Main > State	
Disable		Part and State > Main > Disable	
Background color		Part and State > Main > Background color	
Opacity		Part and State > Main > Opacity	max: 255
Background gradient direction		Part and State > Main > Background gradient direction	
State		Part and State > Digit > State	
Disable		Part and State > Digit > Disable	
Font color		Part and State > Digit > Font color	
Font size, font family		Part and State > Digit > Font size, font family	

4.3 Animation image

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



Figure 8. Animation image in v7



Figure 9. Animation image in v8

Table 12. Animation image properties

Property	v7	v8	Description
Interval	Attribute > Interval	Attribute > Interval	Time between two frames
Repeat Count	Attribute > Repeat Count	Attribute > Repeat Count	Animation repeat time
Start callback function	Attribute > Start callback function	Attribute > Start callback function	Set the playback-customized start callback function
Ready callback function	Attribute > Ready callback function	Attribute > Ready callback function	Set the playback-customized ready callback function
Image Path	Attribute > Image Path	Attribute > Image Path	Add the image files

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.

4.4 Arc

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

Figure 10. Arc in v7



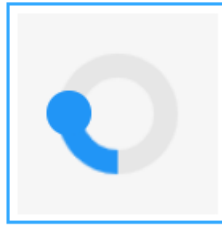


Figure 11. Arc in v8

Table 13. Arc properties

Property	v7	v8	Description
Model	Attribute > Model	Attribute > Model	Normal, symmetrical, reverse
Value	Attribute > Value	Attribute > Value	Start value and end value
Angle	Attribute > Angle	Attribute > Angle	Set the arc indicator start and end angle
Background angle	Attribute > Background angle	Attribute > Background angle	Set the arc Indicator background start and end angle
rotate	Attribute > rotate	Attribute > rotate	
Line cap	Attribute > Line cap	Attribute > Line cap	Set the style of Line cap, butt, or round
state	Part and State > Main > state	Part and State > Main > state	Style of main part: it can be defined by one state or more states
Disable	Part and State > Main > Disable	Part and State > Main > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Background color	Part and State > Main > Background color	Part and State > Main > Background color	Background color of main part: it can be set by color picker or input RGB value
Opacity	Part and State > Main > Opacity	Part and State > Main > Opacity	Background opacity of main part: the value should be between 0 and 255
Background gradient direction	Part and State > Main > Background gradient direction	Part and State > Main > Background gradient direction	Gradient direction of background color in main part, vertical, or horizontal
Line color	Part and State > Main > Line color	Part and State > Main > Line color	Line color: it can be set by color picker or input RGB value
Padding left, right, top bottom	Part and State > Main > Padding left, right, top bottom	Part and State > Main > Padding left, right, top bottom	Padding for the width: the value should be between 1 and 200
Line width	Part and State > Main > Line width	Part and State > Main > Line width	Line width: the values should be between 0 and 20
state	Part and State > Indicator > state	Part and State > Indicator > state	Style of Indicator part: it can be defined by one state or more states
Disable	Part and State > Indicator > Disable	Part and State > Indicator > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Line color	Part and State > Indicator > Line color	Part and State > Indicator > Line color	Indicator Line color: it can be set by color picker or input RGB value

Table 13. Arc properties...continued

Property	v7	v8	Description
Line width	Part and State > Indicator > Line width	Part and State > Indicator > line width	Indicator Line width: the values should be between 0 and 20
state	Part and State > Knob > state	Part and State > Knob > state	Style of knob part: it can be defined by one state or more states
Disable	Part and State > Knob > Disable	Part and State > Knob > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Knob padding	Part and State > Knob > Line color	Part and State > Knob > Line color	Knob Line color: it can be set by color picker or input RGB value
background color	Part and State > Knob > size	Part and State > Knob > size	Knob size: the values must be between 0 and 20
Background gradient direction	Part and State > Knob > Background gradient direction	Part and State > Knob > Background gradient direction	Gradient direction of background color in knob part, vertical, or horizontal.
Opacity	Part and State > Knob > opacity	Part and State > Knob > opacity	Background opacity of Knob part: the value should be between 0 and 255

4.5 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.



Figure 12. Bar in v7



Figure 13. Bar in v8

Property	v7	v8	Description
Animtime	Attribute > Animtime	Attribute > Animation time	The animation time of Bar value
Animation mode	N/A	Attribute > Animation mode	The animation mode, normal, and symmetrical
Bar value	Attribute > Bar value	Attribute > Bar value	Can set value of the Bar widget
state	Part and State > Main > state	Part and State > Main > state	Style of main part: defined by one state or more states
Disable	Part and State > Main > Disable	Part and State > Main > Disable	Enable or disable a state. The custom settings are

Property	v7	v8	Description
			invalidated when a state is disabled.
Background color	Part and State > Main > Background color	Part and State > Main > Background color	Background color of main part: it can be set by color picker or input RGB value
Opacity	Part and State > Main > Opacity	Part and State > Main > Opacity	Background opacity of main part: the value should be between 0 and 255
Background gradient direction	Part and State > Main > Background gradient direction	Part and State > Main > Background gradient direction	Gradient direction of background color in main part, vertical, or horizontal
Radius	Part and State > Main > Radius	Part and State > Main > Radius	Radius of Border: 0-200
state	Part and State > Indicator > state	Part and State > Indicator > state	Style of Active part: it can be defined by one state or more states
Disable	Part and State > Indicator > Disable	Part and State > Indicator > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Background color	Part and State > Indicator > Background color	Part and State > Indicator > Background color	Background color of Active part: it can be set by color picker or input RGB value
Opacity	Part and State > Indicator > Opacity	Part and State > Indicator > Opacity	Background opacity of Active part: the value should be between 0 and 255
Background gradient direction	Part and State > Indicator > Background gradient direction	Part and State > Indicator > Background gradient direction	Gradient direction of background color in main part, vertical, or horizontal

4.6 Button

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

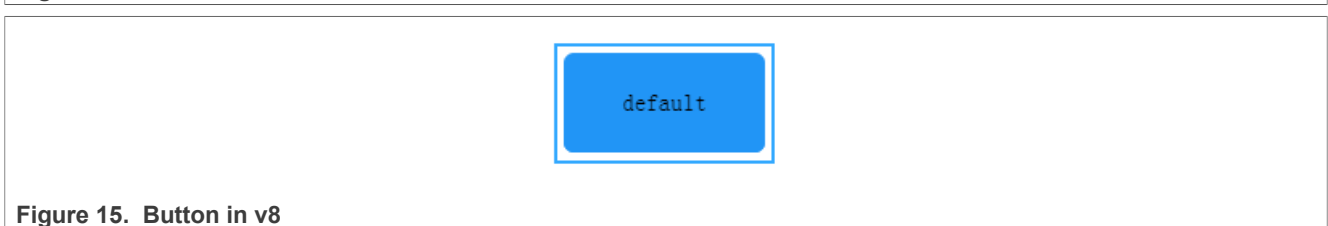
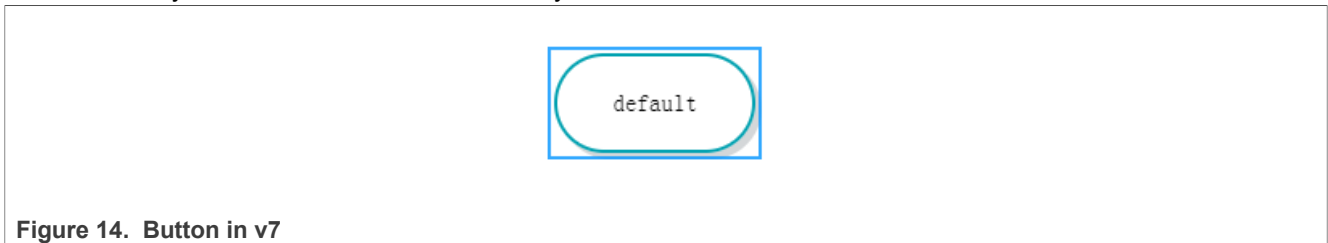


Table 14. Button properties

Property	v7	v8	Description
Text, text color	Attribute > Text, text color	Attribute > Text, text color	Text color: it can be set by color picker or input RGB value
Toggle	Attribute > Toggle	Attribute > Toggle	Enable or disable the toggle button
state	Part and State > Main > state	Part and State > Main > state	Style of main part: it can be defined by one state or more states
Disable	Part and State > Main > Disable	Part and State > Main > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Background color	Part and State > Main > Background color	Part and State > Main > Background color	Background color of widget: it can be set by color picker or input RGB value
Opacity	Part and State > Main > Opacity	Part and State > Main > Opacity	Background opacity: the value should be between 0 and 255
Background gradient direction	Part and State > Main > Background gradient direction	Part and State > Main > Background gradient direction	Gradient direction of background color: vertical or horizontal
Font color	Part and State > Main > font color	Part and State > Main > font color	
Font size	Part and State > Main > font size	Part and State > Main > font size	
align	Part and State > Main > align	Part and State > Main > align	Left, center, right
Border width	Part and State > Main > Border width	Part and State > Main > Border width	Width of border line: the value can be 1 to 5
Border color	Part and State > Main > Border color	Part and State > Main > Border color	Color of border: it can be set by color picker or input RGB value
Border opacity	Part and State > Main > Border opacity	Part and State > Main > Border opacity	Border opacity: the value should be between 0 and 255
Border radius	Part and State > Main > Border radius	Part and State > Main > Border radius	Radius of Border: 0-200

4.7 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 less memory per button.

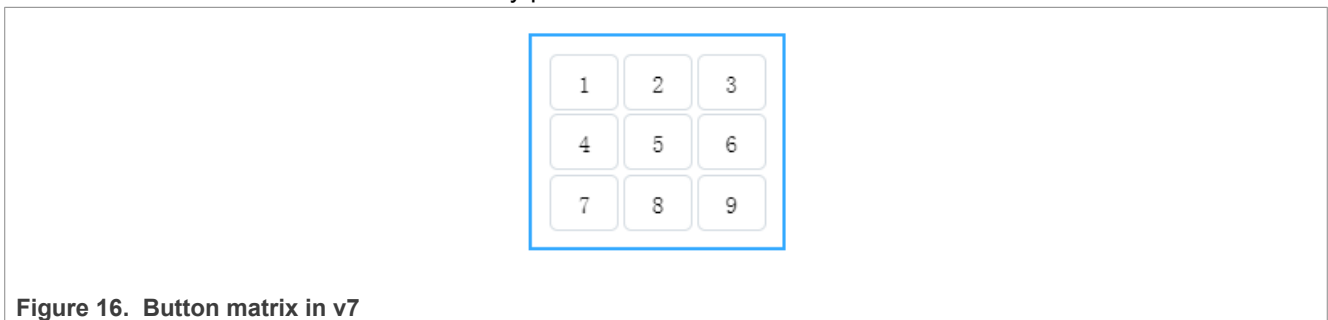


Figure 16. Button matrix in v7

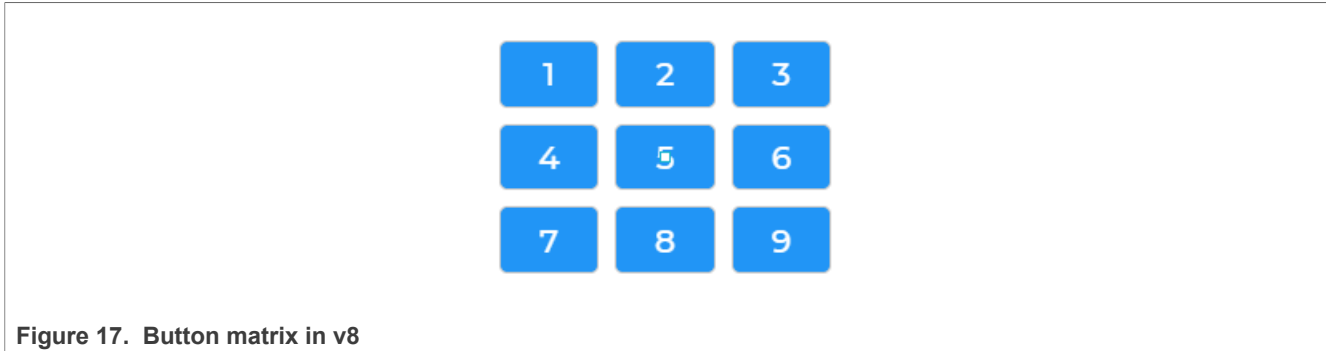


Figure 17. Button matrix in v8

Table 15. Button matrix properties

Property	v7	v8	Description
button	Attribute > button item	Attribute > button	
state	Part and State > Main > state	Part and State > Main > state	Style of main part: it can be defined by one state or more states
Disable	Part and State > Main > Disable	Part and State > Main > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Background color	Part and State > Main > Background color	Part and State > Main > Background color	Background color of widget: it can be set by color picker or input RGB value
Opacity	Part and State > Main > Opacity	Part and State > Main > Opacity	Background opacity: the value should be between 0 and 255
Background gradient direction	Part and State > Main > Background gradient direction	Part and State > Main > Background gradient direction	Gradient direction of background color: vertical or horizontal
Radius	Part and State > Main > Radius	Part and State > Main > Border Radius	Radius of Border: 0-200
Border color	Part and State > Main > Border color	Part and State > Main > Border color	Color of border: it can be set by color picker or input RGB value
Border width	Part and State > Main > Border width	Part and State > Main > Border width	Width of border line: the value should be between 1 and 5
Padding top, bottom, left, right	Part and State > Main > Padding top, bottom, left, right	Part and State > Main > Padding top, bottom, left, right	Padding for the width: the value should be between 1 and 200
Padding inner	Part and State > Main > Padding inner	N/A	Padding for the Button space
Padding row & column	N/A	Part and State > Main > Padding row & column	Row and column for Items group
state	Part and State > Items > state	Part and State > Items > state	Style of main part: it can be defined by one state or more states
Disable	Part and State > Items > Disable	Part and State > Items > Disable	Enable or disable a state: the custom settings are invalidated when a state is disabled
Text color	Part and State > Items > Text color	Part and State > Items > Font color	Color for the text
Font family, font size	Part and State > Items > Font family, font size	Part and State > Items > Font family, font size	Font family and size settings: the size can be a value between 0 and 100

Table 15. Button matrix properties...continued

Property	v7	v8	Description
Background color	Part and State > Items > Background color	Part and State > Items > Background color	Background color of widget: it can be set by color picker or input RGB value
Background gradient direction	Part and State > Items > Background gradient direction	Part and State > Items > Background gradient direction	Gradient direction of background color: vertical or horizontal
Opacity	Part and State > Items > Opacity	Part and State > Items > Opacity	Background opacity: the value should be between 0 and 255
Border color	Part and State > Items > Border color	Part and State > Items > Border color	Color of border: it can be set by color picker or input RGB value
Border width	Part and State > Items > Border width	Part and State > Items > Border width	
Radius	Part and State > Items > Radius	Part and State > Items > Border Radius	Radius of Border: 0-8

4.8 Calendar

The calendar object is a classic calendar which can:

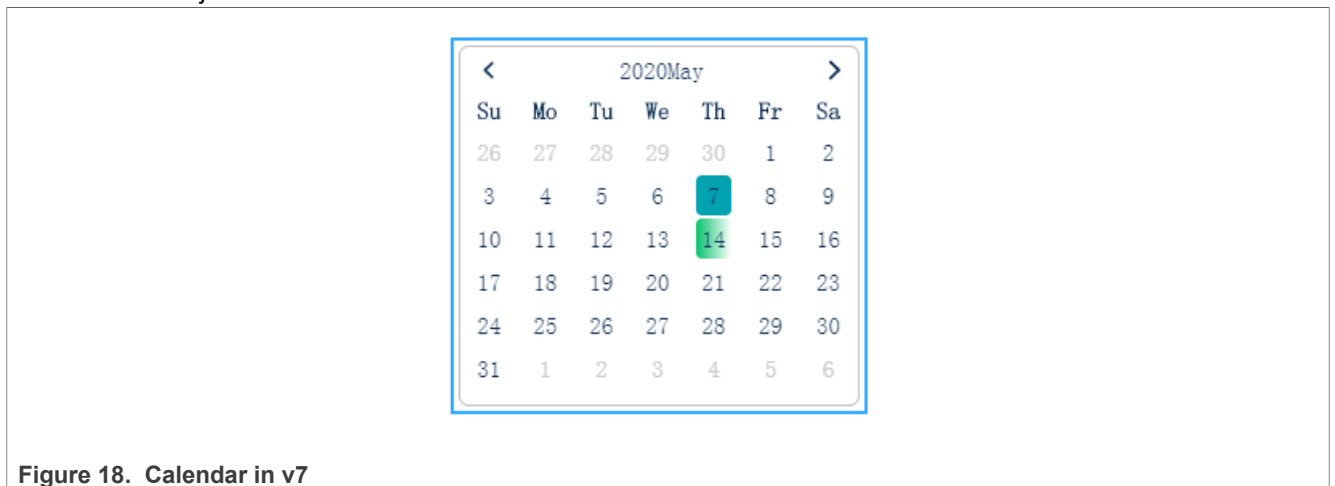


Figure 18. Calendar in v7

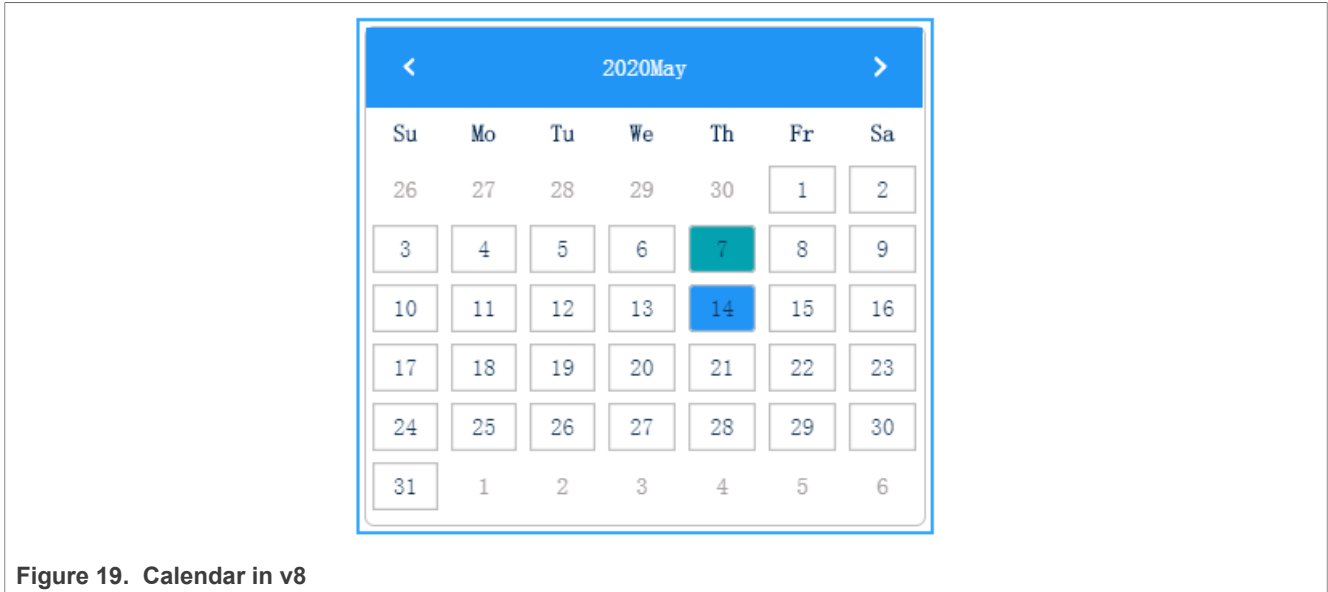


Figure 19. Calendar in v8

Table 16. Calendar properties

Property	v7	v8	Description
state	Part and State > Main > state	Part and State > Main > state	Style of main part: it can be defined by one state or more states
Disable	Part and State > Main > Disable	Part and State > Main > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Background color	Part and State > Main > Background color	Part and State > Main > Background color	Background color of widget: it can be set by color picker or input RGB value
Opacity	Part and State > Main > Opacity	Part and State > Main > Opacity	Background opacity: the value should be between 0 and 255
Background gradient direction	Part and State > Main > Background gradient direction	Part and State > Main > Background gradient direction	Gradient direction of background color: vertical or horizontal
Border color	Part and State > Main > Border color	Part and State > Main > Border color	Color of border: it can be set by color picker or input RGB value
Border width	Part and State > Main > Border width	Part and State > Main > Border width	Width of border line: value can be 1 or 2
state	Part and State > Header > state	Part and State > header > state	Style of header part: it can be defined by one state or more states
Disable	Part and State > Header > Disable	Part and State > header > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.

Table 16. Calendar properties...continued

Property	v7	v8	Description
Text color	Part and State > Header > Text color	Part and State > header > Font color	Text color in head part: it can be set by color picker or input RGB value.
Font family, font size	Part and State > Header > Font family, font size	Part and State > header > Font family, font size	Font family and size settings in the calendar header: the size can be a value between 0 and 100
Letter spacing	Part and State > Header > Letter spacing	N/A	The spacing size between letters in the header part
Background color	N/A	Part and State > header > Background color	Background color of header part: it can be set by color picker or input RGB value
Opacity	N/A	Part and State > header > Opacity	Background opacity of header part: the value should be between 0 and 255
Background gradient direction	N/A	Part and State > header > Background gradient direction	Gradient direction of background color in head part: vertical or horizontal
state	Part and State > Week > state	N/A	Style of day part: it can be defined by one state or more states
Disable	Part and State > Week > Disable	N/A	Enable or disable a state of day part. The custom settings are invalidated when a state is disabled
Text color	Part and State > Week > Text color	N/A	Text color in day part: it can be set by color picker or input RGB value
Font family, font size	Part and State > Week > Font family, font size	N/A	Font family and size settings in the calendar day part: the size can be a value between 0 and 100
state	Part and State > Date > state	N/A	Style of date part: it can be defined by one state or more states
Disable	Part and State > Date > Disable	N/A	Enable or disable a state of date part. The custom settings are invalidated when a state is disabled.
Text color	Part and State > Date > Text color	N/A	Text color in date part: it can be set by color picker or input RGB value
Font family, font size	Part and State > Date > Font family, font size	N/A	Font family and size settings in the calendar date part: the size can be a value between 0 and 100

Table 16. Calendar properties...continued

Property	v7	v8	Description
Background color	Part and State > Date > Background color	N/A	Background color of date part: it can be set by color picker or input RGB value
Opacity	Part and State > Date > Opacity	N/A	Background opacity of date part: the value should be between 0 and 255
Background gradient direction	Part and State > Date > Background gradient direction	N/A	Gradient direction of background color in date part: vertical or horizontal
state	N/A	Part and State > Disabled > state	Style of disabled part: it can be defined by one state or more states
Disable	N/A	Part and State > Days in other month > disabled	Enable or disable a state of disabled part. The custom settings are invalidated when a state is disabled.
Background color	N/A	Part and State > Days in other month > Background color	Background color of disabled part: it can be set by color picker or input RGB value
Opacity	N/A	Part and State > Days in other month > Opacity	The background opacity of the next month or last month date
Background gradient direction	N/A	Part and State > Days in other month > Background gradient direction	Gradient direction of background color in disabled part, vertical, or horizontal
Font color	N/A	Part and State > Days in other month > Font color	Text color in disabled part: it can be set by color picker or input RGB value
Font family, font size	N/A	Part and State > Days in other month > Font family, font size	Font family and size settings in the calendar disabled part: the size can be a value between 0 and 100
state	N/A	Part and State > Days in current month > state	Style of current month: it can be defined by one state or more states
Disable	N/A	Part and State > Days in current month > Disabled	Enable or disable a state of month part. The custom settings are invalidated when a state is disabled.
Background color	N/A	Part and State > Days in current month > Background color	Background color of current month date: it can be set by color picker or input RGB value
Opacity	N/A	Part and State > Days in current month > Opacity	The background opacity of current month date
Background gradient direction	N/A	Part and State > Days in current month > Background gradient direction	Gradient direction of background color in current

Table 16. Calendar properties...continued

Property	v7	v8	Description
			month date: vertical or horizontal
Border color	N/A	Part and State > Days in current month > Border color	Border color of current month date: it can be set by color picker or input RGB value
Border width	N/A	Part and State > Days in current month > Border width	Border width of current month date: the value can be 1 or 2
Font color	N/A	Part and State > Days in current month > Font color	Text color of current month date: it can be set by color picker or input RGB value
Font family, font size	N/A	Part and State > Days in current month > Font family, font size	Font family and size settings for current month date: the size can be a value between 0 and 100
state	N/A	Part and State > weekday names > state	Style of week part: it can be defined by one state or more states
Disable	N/A	Part and State > weekday names > disabled	Enable or disable a state of week part. The custom settings are invalidated when a state is disabled.
Font color	N/A	Part and State > weekday names > Font color	Text color of week part: it can be set by color picker or input RGB value
Font family, font size	N/A	Part and State > weekday names > Font family, font size	Font family and size settings for week part: the size can be a value between 0 and 100
state	N/A	Part and State > Today > state	Style of today: it can be defined by one state or more states
Disable	N/A	Part and State > Today > Disabled	Enable or disable a state of today. The custom settings are invalidated when a state is disabled.
Background color	N/A	Part and State > Today > Background color	Background color of today: it can be set by color picker or input RGB value
Opacity	N/A	Part and State > Today > Opacity	The background opacity of today
Background gradient direction	N/A	Part and State > Today > Background gradient direction	Gradient direction of background color for today: vertical or horizontal
Font color	N/A	Part and State > Today > Font color	Text color of today: it can be set by color picker or input RGB value

Table 16. Calendar properties...continued

Property	v7	v8	Description
Font family, font size	N/A	Part and State > Today > Font family, font size	Font family and size settings for today: the size can be a value between 0 and 100
state	N/A	Part and State > Highlight day > state	Style of highlight part: it can be defined by one state or more states
Disable	N/A	Part and State > Highlight day > Disabled	Enable or disable a state of highlight part. The custom settings are invalidated when a state is disabled.
Background color	N/A	Part and State > Highlight day > Background color	Background color of highlight part: it can be set by color picker or input RGB value
Opacity	N/A	Part and State > Highlight day > Opacity	The background opacity of highlight date
Background gradient direction	N/A	Part and State > Highlight day > Background gradient direction	Gradient direction of background color for highlight date: vertical or horizontal
Font color	N/A	Part and State > Highlight day > Font color	Text color of highlight date: it can be set by color picker or input RGB value
Font family, font size	N/A	Part and State > Highlight day > Font family, font size	Font family and size settings for highlight date: the size can be a value between 0 and 100

4.9 Canvas

A canvas inherits from [Image](#) where the user can draw anything. Rectangles, texts, images, and lines arcs can be drawn here using the drawing engine of LVGL. Besides some "effects" can be applied as well like rotation, zoom, and blur.

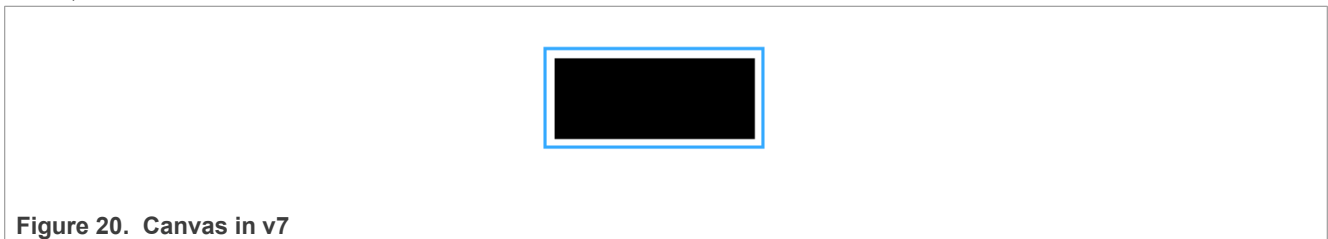


Table 17. Canvas properties

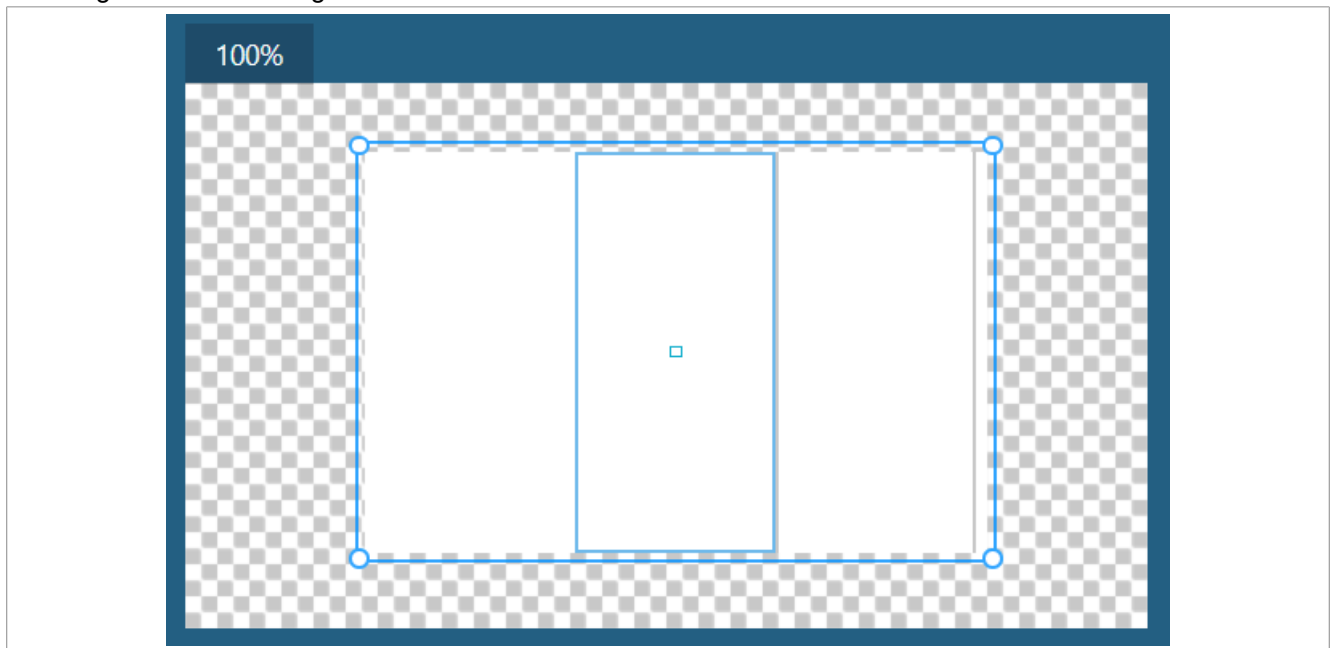
Property	v7	v8	Description
state	Part and State > Main > state	Part and State > Main > state	Style of main part: it can be defined by one state or more states
Disable	Part and State > Main > disable	Part and State > Main > disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Background color	Part and State > Main > Background color	Part and State > Main > Background color	Background color of widget: it can be set by color picker or input RGB value
Opacity	Part and State > Main > Opacity	Part and State > Main > Opacity	Background opacity: the value should be between 0 and 255

4.10 Carousel

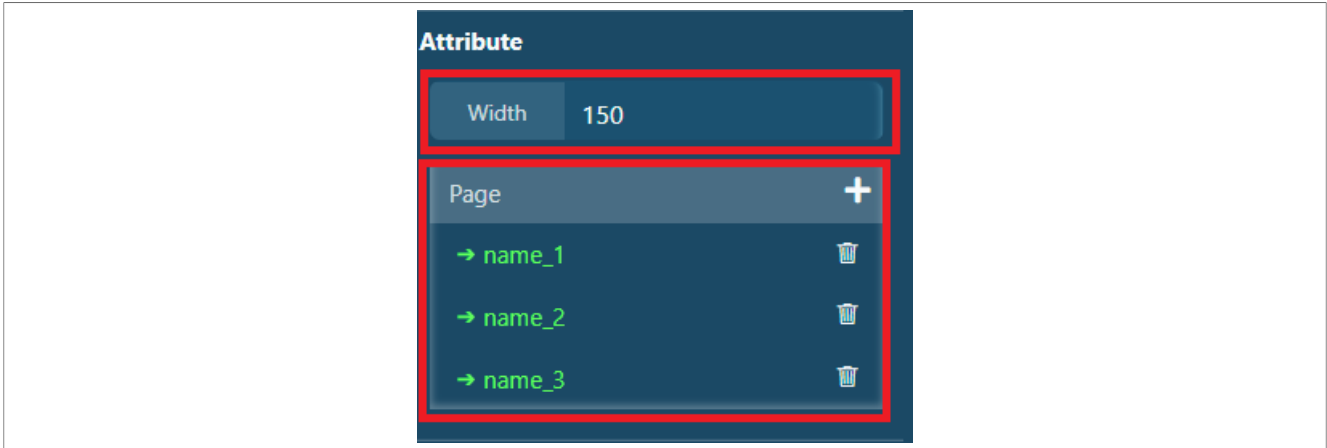
The carousel widget can display two or more pieces of content in a carousel format.

The following is the usage example of carousel widget.

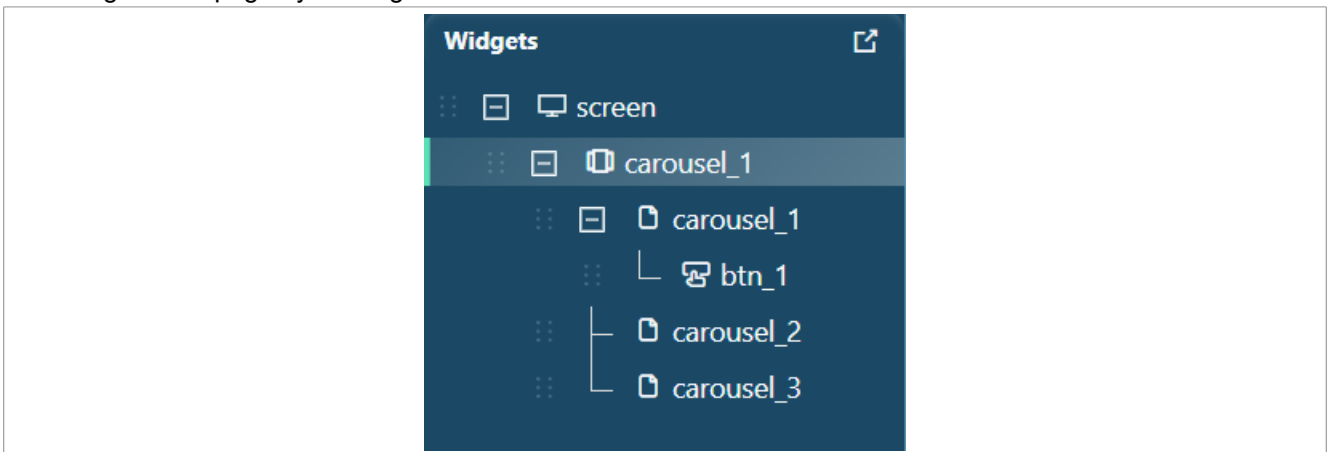
1. Drag the carousel widget into the editor.



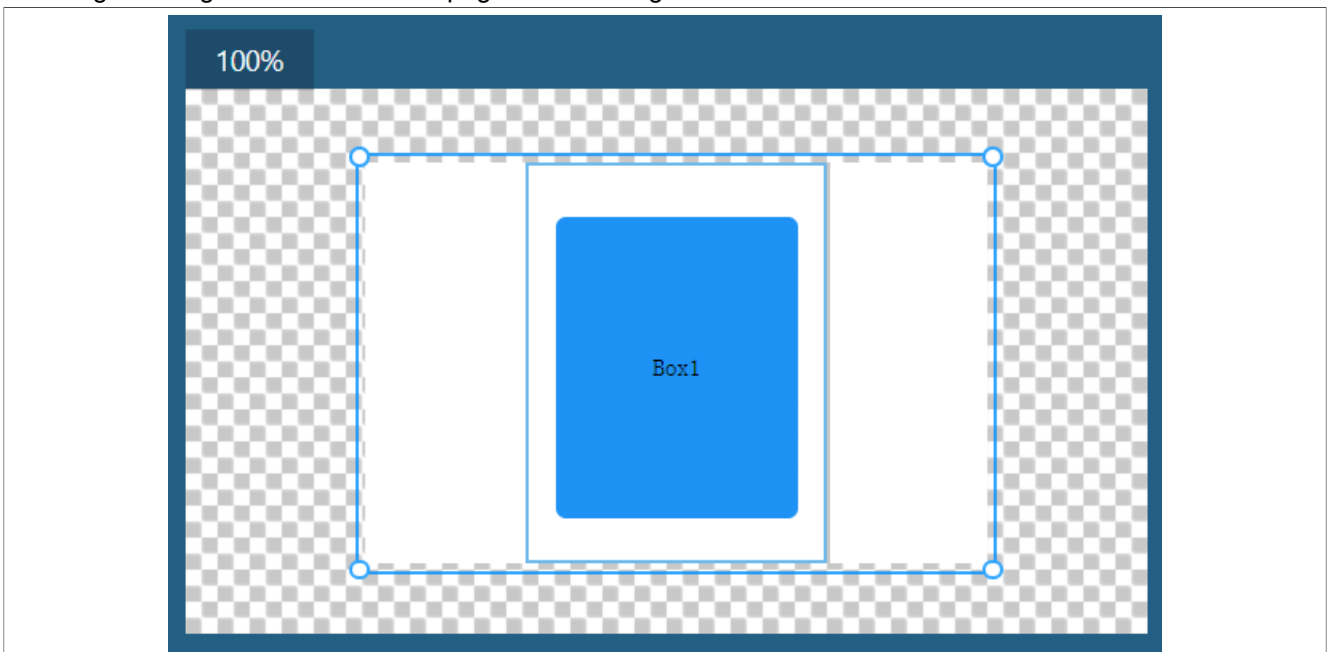
2. Set the page width and page number in the attribute setting window.



3. Navigate to a page by clicking the resource tree.



4. Drag the widget into the carousel page and set widget attributes.



5. Repeat step 4 for each page of the carousel widget.

6. Run the project. The pages are displayed in a carousel format.

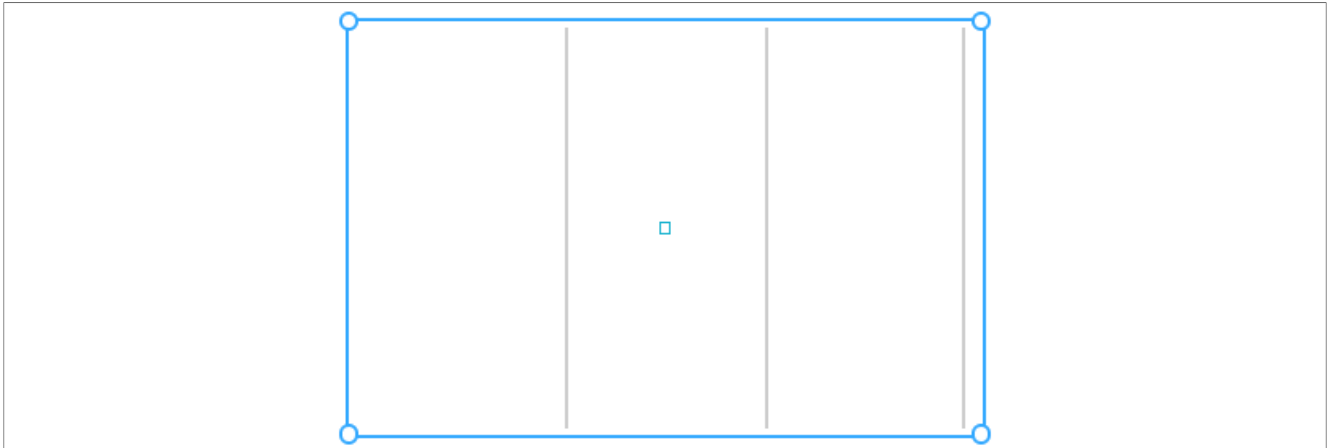


Figure 22. Carousel in v8

Table 18. Carousel properties

Property	v8	Description
width	Attribute > width	The page width
Page	Attribute > add page	Add the multiple pages
state	Part and State > Main > state	Style of main part: it can be defined by one state or more states
Disable	Part and State > Main > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
background color	Part and State > Main > background color	Background color of widget: it can be set by color picker or input RGB value
Background gradient direction	Part and State > Main > Background gradient direction	Gradient direction of background color: vertical or horizontal
Opacity	Part and State > Main > Opacity	Background opacity: the value should be between 0 and 255
state	Part and State > Scrollbar > state	Style of main part: it can be defined by one state or more states
Disable	Part and State > Scrollbar > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
background color	Part and State > Scrollbar > background color	Scrollbar color of widget: it can be set by color picker or input RGB value
Opacity	Part and State > Scrollbar > Opacity	Scrollbar opacity: the value should be between 0 and 255

4.11 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. The number of chart points must be 10.

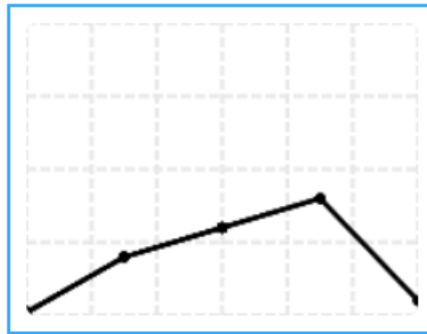


Figure 23. Chart in v7

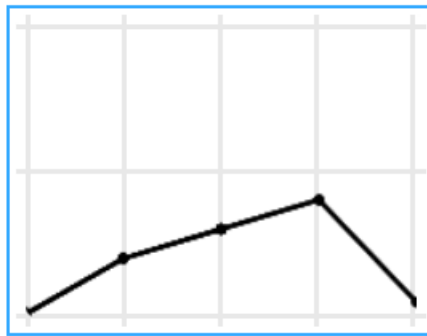


Figure 24. Chart in v8

Table 19. Chart properties

Property	v7	v8	Description
Line numbers row and column	Attribute > Line numbers row and column	Attribute > Line numbers row and column	Line of chart: row line and column line of chart
Range	Attribute > Range	Attribute > Range	Value range of the chart: the min value is 0; There is no max value limit.
Chart type	Attribute > Chart type	Attribute > Chart type	Type of chart: it includes line and bar
Chart data	Attribute > Chart data	Attribute > Chart type	Data of the chart: the length of data is point count
state	Part and State > Main > state	Part and State > Main > state	Style of highlight part: it can be defined by one state or more states
Disable	Part and State > Main > Disable	Part and State > Main > Disable	Enable or disable a state of today. The custom settings are invalidated when a state is disabled.
Background color	Part and State > Main > Background color	Part and State > Main > Background color	Background color of widget: it can be set by color picker or input RGB value

Table 19. Chart properties...continued

Property	v7	v8	Description
Opacity	Part and State > Main > Opacity	Part and State > Main > Opacity	Background opacity: the value should be between 0 and 255
Background gradient direction	Part and State > Main > Background gradient direction	Part and State > Main > Background gradient direction	Gradient direction of background color for highlight date: vertical or horizontal
state	Part and State > Lines > state	N/A	Style of highlight part: it can be defined by one state or more states
Disable	Part and State > Lines > Disable	N/A	Enable or disable a state of today. The custom settings are invalidated when a state is disabled
Line color	Part and State > Lines > Line color	Part and State > Main > Line color	Color of background line
Line opacity	Part and State > Lines > Line opacity	Part and State > Main > Line opacity	Background line opacity: the value should be between 0 and 255
Line width	Part and State > Lines > Line width	Part and State > Main > Line width	Background line width: the value should be between 0 and 4

4.12 Checkbox

The checkbox objects are built from a [Button](#) background which contains a Button *bullet* and a [Label](#) to realize a classical checkbox.

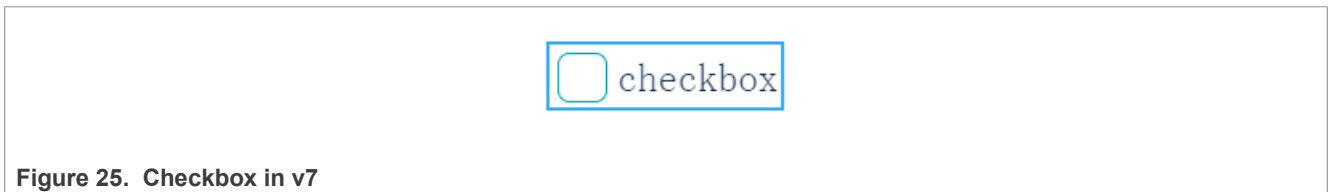


Figure 25. Checkbox in v7

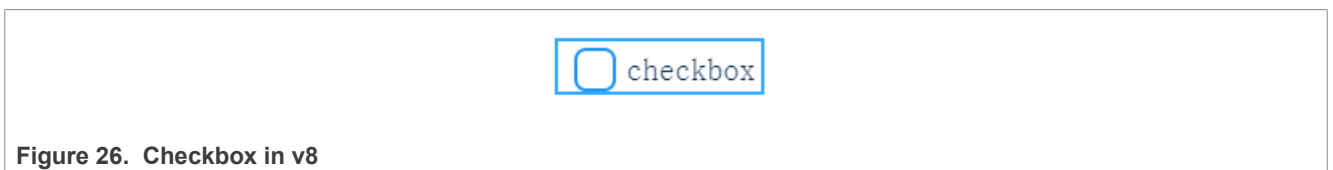


Figure 26. Checkbox in v8

Table 20. Checkbox properties

Property	v7	v8	Description
Text	Attribute > Text	Attribute > Text	Text for checkbox
state	Part and State > Main > state	Part and State > Main > state	Style of highlight part: it can be defined by one state or more states
Text color	Part and State > Main > Text color	Part and State > Main > Font color	Color of the text
Font famil, font size	Part and State > Main > Font famil, font size	Part and State > Main > Font famil, font size	Text font setting: includes font family and font size

Table 20. Checkbox properties...continued

Property	v7	v8	Description
Letter spacing	Part and State > Main > Letter spacing	Part and State > Main > Letter spacing	The spacing size between letters in the Text
Disable	Part and State > Main > Disable	Part and State > Main > Disable	Enable or disable a state of today. The custom settings are invalidated when a state is disabled.
Background color	Part and State > Main > Background color	Part and State > Main > Background color	Background color of widget: it can be set by color picker or input RGB value
Opacity	Part and State > Main > Opacity	Part and State > Main > Opacity	Background opacity: the value should be between 0 and 255
Background gradient direction	Part and State > Main > Background gradient direction	Part and State > Main > Background gradient direction	Gradient direction of background color for highlight date: vertical or horizontal
Radius	Part and State > Main > Radius	Part and State > Main > Radius	Radius of Border: 0-200
state	Part and State > indicator > state	Part and State > indicator > state	Style of highlight part: it can be defined by one state or more states
Background color	Part and State > indicator > Background color	Part and State > indicator > Background color	Background color of widget: it can be set by color picker or input RGB value
Opacity	Part and State > indicator > Opacity	Part and State > indicator > Opacity	Background opacity: the value should be between 0 and 255
Background gradient direction	Part and State > indicator > Background gradient direction	Part and State > indicator > Background gradient direction	Gradient direction of background color for highlight date: vertical or horizontal
Border color	Part and State > indicator > Border color	Part and State > indicator > Border color	Border color of widget: it can be set by color picker or input RGB value
Border width	Part and State > indicator > Border width	Part and State > indicator > Border width	Border width: The value should be between 0 and 4
Radius	Part and State > indicator > Radius	Part and State > indicator > Radius	Radius of Border: 0-200

4.13 Color picker

As its name implies, the *Color picker* helps select the 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, when long pressing the object, the color picker changes to the next parameter of the color (hue, saturation or value). The double-click resets the current parameter.



Figure 27. Color picker in v7



Figure 28. Color picker in v8

Table 21. Color picker properties

Property	v7	v8	Description
state	Part and State > Main > state	Part and State > Main > state	Size of widget: set the width and height of the widget
Disable	Part and State > Main > Disable	Part and State > Main > Disable	Style of main part: it can be defined by one state or more states
Scale width	Part and State > Main > Scale width	Part and State > Main > Scale width	Scale width of the color picker: 0-50
Type	Part and State > Main > Type	N/A	Color picker type: Disc and Rect

4.14 Container

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

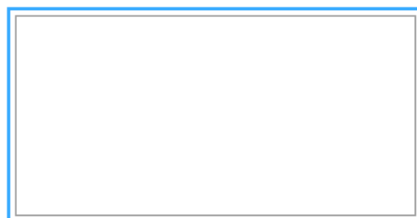


Figure 29. Container in v7

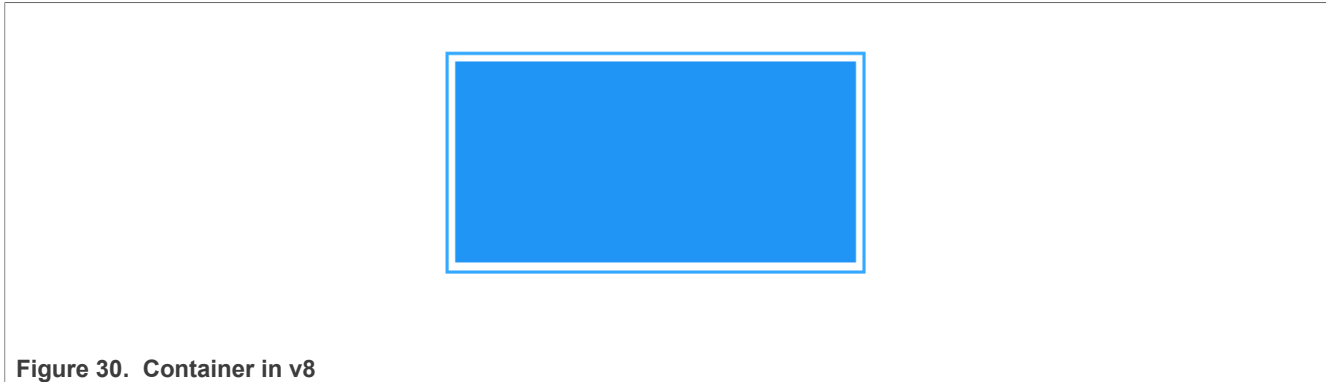


Figure 30. Container in v8

Table 22. Container properties

Property	v7	v8	Description
Remove all style	General Setting > Remove all style	General Setting > Remove all style	Enabling this item removes all custom added style
state	Part and State > Main > state	Part and State > Main > state	Style of main: it can be defined by one state or more states
Disable	Part and State > Main > Disable	Part and State > Main > Disable	Enable or disable a state of month part. The custom settings are invalidated when a state is disabled.
Background color	Part and State > Main > Background color	Part and State > Main > Background color	Background color of widget: it can be set by color picker or input RGB value
Opacity	Part and State > Main > Opacity	Part and State > Main > Opacity	Background opacity: the value should be between 0 and 255
Background gradient direction	Part and State > Main > Background gradient direction	Part and State > Main > Background gradient direction	Gradient direction of background color: vertical or horizontal
Border color	Part and State > Main > Border color	Part and State > Main > Border color	Color of border: it can be set by color picker or input RGB value
Border opacity	Part and State > Main > Border opacity	Part and State > Main > Border opacity	Opacity of border line: it can be set by the slider
Border width	Part and State > Main > Border width	Part and State > Main > Border width	Width of Border: max:10
Padding top, bottom, left, right	Part and State > Main > Padding top, bottom, left, right	Part and State > Main > Padding top, bottom, left, right	Padding for the width: the value should be between 1 and 200
Radius	Part and State > Main > Radius	Part and State > Main > Border Radius	Radius of Border: 0-200

4.15 Date text box

A date text box is a widget that allows users to select a date in the calendar and display the date in the label.

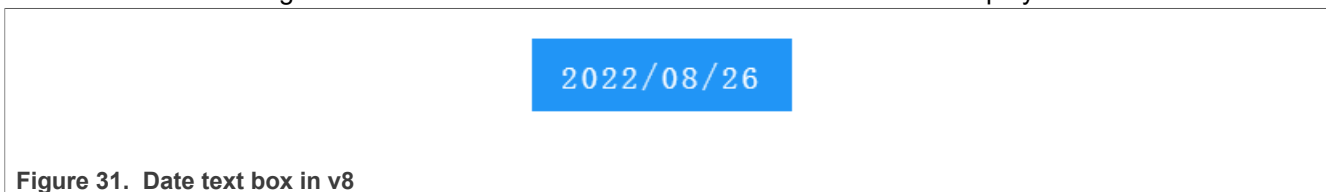


Figure 31. Date text box in v8

Table 23. Date text box properties

Property	v7	v8	Description
Datetext	N/A	Attribute > Datetext	Can select by calendar
Disable	N/A	Part and State > Main > Disable	
Background color	N/A	Part and State > Main > Background color	
Opacity	N/A	Part and State > Main > Opacity	
Background gradient direction	N/A	Part and State > Main > Background gradient direction	
Font color	N/A	Part and State > Main > Font color	
Font size, font family	N/A	Part and State > Main > Font size, font family	
Letter spacing	N/A	Part and State > Main > Letter spacing	
Padding top, left, right	N/A	Part and State > Main > Padding top, left, right	

4.16 Digital clock

The following are the details of the digital clock widget.

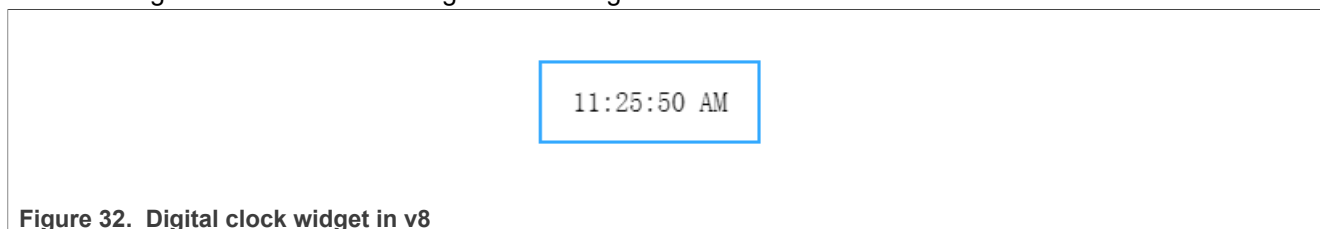


Figure 32. Digital clock widget in v8

Table 24. Digital clock properties

Property	v7	v8	Description
Initial Time		Attribute > Initial Time	Initial time format is 00:00:00
Show Second		Attribute > Second	
Use AM/PM		Attribute > AM/PM	
State		Part and State > Main > State	
Disable		Part and State > Main > Disable	
Font color		Part and State > Main > Font color	
Font family, font size		Part and State > Main > Font family, font size	
Letter spacing		Part and State > Main > Letter spacing	max:100
Background color		Part and State > Main > Background color	

Table 24. Digital clock properties...continued

Property	v7	v8	Description
Opacity		Part and State > Main > Opacity	
Background gradient direction		Part and State > Main > Background gradient direction	max: 255
Radius		Part and State > Main > Radius	max: 100

4.17 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. Click the drop-down list and select an option. When the user selects a new value, the list is deleted.

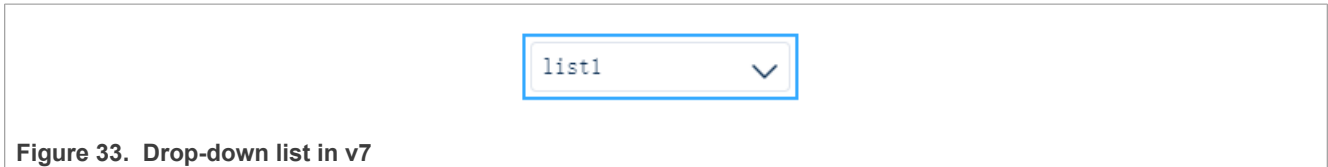


Figure 33. Drop-down list in v7

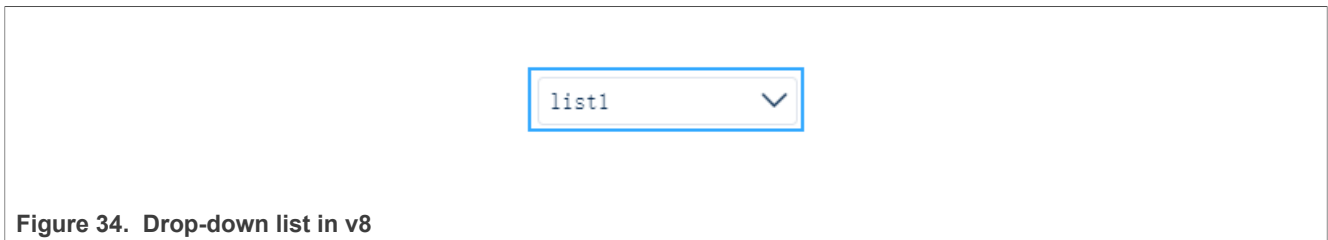


Figure 34. Drop-down list in v8

Table 25. Drop-down list properties

Property	v7	v8	Description
Draw arrow	Attribute > arrow	Attribute > arrow	Set whether to add a drop-down arrow
Text	Attribute > Text	Attribute > Text	The contents of each row in the list
Add list	Attribute > Add item	Attribute > Add item	To add a list
state	Part and State > Main > state	Part and State > Main > state	Style of main: it can be defined by one state or more states
Disable	Part and State > Main > Disable	Part and State > Main > Disable	Enable or disable a state of month part. The custom settings are invalidated when a state is disabled.
Text color	Part and State > Main > Text color	Part and State > Main > Font color	Font color in the header: it can be set by color picker or input RGB value
Font family, font size	Part and State > Main > Font family, font size	Part and State > Main > Font family, font size	Font family and size settings in the header part: the size can be a value between 0 and 100
Background color	Part and State > Main > Background color	Part and State > Main > Background color	Background color of widget: it can be set by color picker or input RGB value
Opacity	Part and State > Main > Opacity	Part and State > Main > Opacity	Background opacity: the value should be between 0 and 255
Background gradient direction	Part and State > Main > Background gradient direction	Part and State > Main > Background gradient direction	Gradient direction of background color: vertical or horizontal

Table 25. Drop-down list properties...continued

Property	v7	v8	Description
Border color	Part and State > Main > Border color	Part and State > Main > Border color	Color of border: it can be set by color picker or input RGB value
Border width	Part and State > Main > Border width	Part and State > Main > Border width	Width of the border: max:4
Radius	Part and State > Main > Radius	Part and State > Main > Radius	Radius of the border: 0-200
Padding top, left, right	Part and State > Main > Padding top, left, right	Part and State > Main > Padding top, left, right	Padding for the width: the value should be between 1 and 200
state	Part and State > Selected > state	Part and State > Selected > state	Style of selected part: it can be defined by one state or more states
Disable	Part and State > Selected > Disable	Part and State > Selected > Disable	Enable or disable a state: the custom settings are invalidated when a state is disabled
Text color	Part and State > Selected > Text color	Part and State > Selected > Font color	Font color in the selected item: it can be set by color picker or input RGB value
Font family, font size	Part and State > Selected > Font family, font size	Part and State > Selected > Font family, font size	Font family and size settings in the selected item: the size can be a value between 0 and 100
Background color	Part and State > Selected > Background color	Part and State > Selected > Background color	Background color of widget: it can be set by color picker or input RGB value
Opacity	Part and State > Selected > Opacity	Part and State > Selected > Opacity	Background opacity in the selected item: the value should be between 0 and 255
Background gradient direction	Part and State > Selected > Background gradient direction	Part and State > Selected > Background gradient direction	Gradient direction of background color: vertical or horizontal
Border color	Part and State > Selected > Border color	Part and State > Selected > Border color	Color of the selected item border: it can be set by color picker or input RGB value
Border width	Part and State > Selected > Border width	Part and State > Selected > Border width	width of the selected item Border: max:4
Radius	Part and State > Selected > Radius	Part and State > Selected > Radius	Radius of the selected item Border: 0-200
Padding top, bottom, left, right	Part and State > Selected > Padding top, bottom, left, right	Part and State > Selected > Padding top, bottom, left, right	Padding for the width: the value should be between 1 and 20
state	Part and State > List > state	Part and State > List > state	Style of listed part: it can be defined by one state or more states
Disable	Part and State > List > Disable	Part and State > List > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Text color	Part and State > List > Text color	Part and State > List > Text color	Font color in other unselected items: it can be set by color picker or input RGB value
Font family, font size	Part and State > List > Font family, font size	Part and State > List > Font family, font size	Font family and size settings in other unselected items: the size can be a value between 0 and 100
Background color	Part and State > List > Background color	Part and State > List > Background color	Background color of widget: it can be set by color picker or input RGB value

Table 25. Drop-down list properties...continued

Property	v7	v8	Description
Opacity	Part and State > List > Opacity	Part and State > List > Opacity	Background opacity in other unselected items: the value should be between 0 and 255
Background gradient direction	Part and State > List > Background gradient direction	Part and State > List > Background gradient direction	Gradient direction of background color: vertical or horizontal
Border color	Part and State > List > Border color	Part and State > List > Border color	Color of drop-down part border. it can be set by color picker or input RGB value
Border width	Part and State > List > Border width	Part and State > List > Border width	Width of drop-down part. Border: max:4
Radius	Part and State > List > Radius	Part and State > List > Radius	Radius of drop-down part. Border: 0-200
Padding top, left, right	Part and State > List > Padding top, left, right	Part and State > List > Padding top, left, right	Padding for the width: the value should be between 1 and 200
List height	x	Part and State > List > List Height	The height of the drop-down part
state	x	Part and State > scrollbar > state	Style of scroll bar: it can be defined by one state or more states
Disable	x	Part and State > scrollbar > Disable	Enable or disable a state: the custom settings are invalidated when a state is disabled
Background color	x	Part and State > scrollbar > Background color	Background color of widget: it can be set by color picker or input RGB value
Opacity	x	Part and State > scrollbar > Opacity	Background opacity in other unselected items: the value should be between 0 and 255
Background gradient direction	x	Part and State > scrollbar > Background gradient direction	Gradient direction of background color: vertical or horizontal
Radius	x	Part and State > scrollbar > Radius	Radius of scroll bar: 0-2000

4.18 Gauge

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



Figure 35. Gauge in v7

Table 26. Gauge properties

Property	v7	v8	Description
Dial Major & Dial Minor	Attribute > Major & Minor		The value of the major and minor
Min value & Max value	Attribute > Min & Max		The value of the start and stop
Image Needle	Attribute > Image Needle		To set the needle image
Angle	Attribute > Angle		To set the angle value
needles color & value	Attribute > needles color & value		The value of needles color
Critical	Attribute > Critical		The value of the critical
state	Part and State > Main > state		Style of head part: it can be defined by one state or more states
Disable	Part and State > Main > Disable		Enable or disable a state. The custom settings are invalidated when a state is disabled.
Color	Part and State > Main > Color		Font color: It can be set by color picker or input RGB value
Font family, font size	Part and State > Main > Font family, font size		Font family and size settings in the main part: the size can be a value between 0 and 100
Letter space	Part and State > Main > Letter space		To set the letter space of the value
Background color	Part and State > Main > Background color		Background color of widget: it can be set by color picker or input RGB value
Opacity	Part and State > Main > Opacity		Background opacity: the value should be between 0 and 255
Background gradient direction	Part and State > Main > Background gradient direction		Gradient direction of background color: vertical or horizontal
padding	Part and State > Main > padding		The padding of the number and line
Line color	Part and State > Main > Line color		The line color: it can be set by color picker or input RGB value

Table 26. Gauge properties...continued

Property	v7	v8	Description
Line opacity	Part and State > Main > Line opacity		The line opacity: 0-255
Line width	Part and State > Main > Line width		The line width: 0-20
Scale width	Part and State > Main > Scale width		The scale line width: 0-21
Scale grad color	Part and State > Main > Scale grad color		The scale grad color: it can be set by color picker or input RGB value
Scale end color	Part and State > Main > Scale end color		The scale end color: it can be set by color picker or input RGB value
Scale border width	Part and State > Main > Scale border width		The start scale border line width: 0-30
Scale end border width	Part and State > Main > Scale end border width		The end scale border width: 0-30
Scale end line width	Part and State > Main > Scale end line width		The end scale border line width: 0-6
state	Part and State > Major > state		Style of head part: it can be defined by one state or more states
Disable	Part and State > Major > Disable		Enable or disable a state. The custom settings are invalidated when a state is disabled.
Line color	Part and State > Major > Line color		The major line color
Line opacity	Part and State > Major > Line opacity		The major line opacity: 0-255
Line width	Part and State > Major > Line width		The major line width: 0-20
Scale width	Part and State > Major > Scale width		The major scale width: 0-30
Scale grad color	Part and State > Major > Scale grad color		The scale grad color: it can be set by color picker or input RGB value
Scale end color	Part and State > Major > Scale end color		The scale end color: it can be set by color picker or input RGB value
Scale end line width	Part and State > Major > Scale end line width		max:6
state	Part and State > Needle > state		Style of needle part: it can be defined by one state or more states
Disable	Part and State > Needle > Disable		Enable or disable a state. The custom settings are invalidated when a state is disabled.
Background color	Part and State > Needle > Background color		Background color of needle: it can be set by color picker or input RGB value
Opacity	Part and State > Needle > Opacity		Background opacity. The value should be between 0 and 255.

Table 26. Gauge properties...continued

Property	v7	v8	Description
Background gradient direction	Part and State > Needle > Background gradient direction		Gradient direction of background color: vertical or horizontal
Line opacity	Part and State > Needle > Line opacity		The opacity of the needle: 0-255
Line width	Part and State > Needle > Line width		The line width of the needle: 0-20
size	Part and State > Needle > size		The center pointer size of the needle: 0-20

4.19 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.



Figure 36. Image in v7



Figure 37. Image in v8

Table 27. Image properties

Property	v7	v8	Description
Color format	Attribute > Color format	Attribute > Color format	True Color is for RGB image and True Color Alpha is for ARGB image
Rotate center	Attribute > Rotate center	Attribute > Rotate X,Y	Rotate center of widget. The pivot point of the rotation.
Rotate angle	Attribute > Rotate angle	Attribute > Rotate angle	Rotate angle of widget. The angle to rotate the widget.
Image Path	Attribute > Image Path	Attribute > Image Path	Image path of widget. Choose an image.

Table 27. Image properties...continued

Property	v7	v8	Description
state	Part and State > Main > state	Part and State > Main > state	Style of main part: it can be defined by one state or more states
Disable	Part and State > Main > Disable	Part and State > Main > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Opacity	Part and State > Main > Opacity	Part and State > Main > Opacity	Background opacity: the value should be between 0 and 255
Filter color	Part and State > Main > Filter color	Part and State > Main > Filter color	Filter color: a color can be mixed with every pixel of image
Filter opacity	Part and State > Main > Filter opacity	Part and State > Main > Filter opacity	Filter color opacity: the value should be between 0 and 255

4.20 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.



Figure 38. Image button in v7



Figure 39. Image button in v8

Table 28. Image button properties

Property	v7	v8	Description
Text	Attribute > Text	Attribute > Text	Text of widget: text showed on the widget
Toggle	Attribute > Toggle	Attribute > Toggle	Toggle of widget: the button could be toggled or not
Released picture	Attribute > Released picture	Attribute > Released picture	Released picture of widget: the image when widget is released
Pressed picture	Attribute > Pressed picture	Attribute > Pressed picture	Pressed picture of widget: the image when widget is pressed
Checked released picture	Attribute > Checked released picture	Attribute > Checked released picture	Checked released picture of widget: the image when widget is checked and released
Checked pressed picture	Attribute > Checked pressed picture	Attribute > Checked pressed picture	Checked pressed picture of widget: the image when widget is checked and pressed
state	Part and State > Main > state	Part and State > Main > state	Style of main part: it can be defined by one state or more states

Table 28. Image button properties...continued

Property	v7	v8	Description
Disable	Part and State > Main > Disable	Part and State > Main > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Text color	Part and State > Main > Text color	Part and State > Main > Font color	Color of font: it can be set by color picker or input RGB value
Font family & font size	Part and State > Main > Font family & font size	Part and State > Main > Font family & font size	Font family and size settings for today: the size can be a value between 0 and 10
Font Align	Part and State > Main > Font Align	Part and State Main > Font Align	Left, center, right
Opacity	Part and State > Main > Opacity	Part and State > Main > Opacity	Background opacity: the value should be between 0 and 255
Filter color (image recolor)	Part and State > Main > Filter color	Part and State > Main > Filter color	Filter color: a color can be mixed with every pixel of image
Filter opacity	Part and State > Main > Filter opacity	Part and State > Main > Filter opacity	Filter color opacity: the value should be between 0 and 255

4.21 Label

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

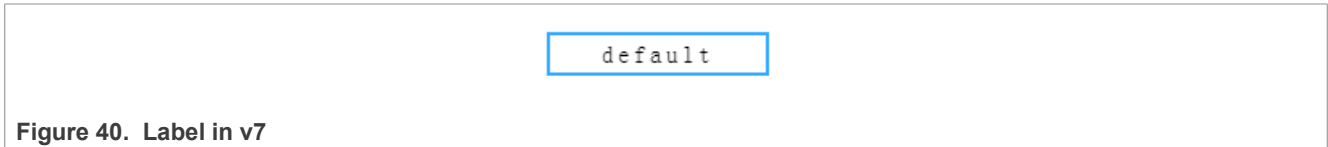


Figure 40. Label in v7

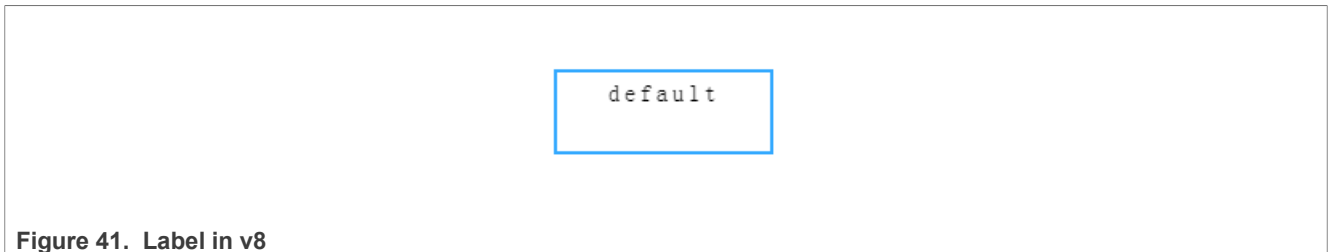


Figure 41. Label in v8

Table 29. Label properties

Property	v7	v8	Description
Text	Attribute > Text	Attribute > Text	Text of widget: text showed on widget
Long Mode	N/A	Attribute > Mode	Long mode of widget: By default, the width and height of the label is set to LV_SIZE_CONTENT. Therefore the size of the label is automatically expanded to the text size. Otherwise, if the width or height are explicitly set (using, for example, lv_obj_set_width or a layout), the lines wider than the label's width can be manipulated according to several long mode policies.
state	Part and State > Main > state	Part and State > Main > state	Style of main part: it can be defined by one state or more states
Disable	Part and State > Main > Disable	Part and State > Main > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.

Table 29. Label properties...continued

Property	v7	v8	Description
Text color	Part and State > Main > Text color	Part and State > Main > Font color	Color of font: it can be set by color picker or input RGB value
Font family, font size	Part and State > Main > Font family, font size	Part and State > Main > Font family, font size	Font family and size settings for today: the size can be a value between 0 and 100
Font Align	Part and State > Main > Font Align	Part and State Main > Font Align	Left, center, right
letter spacing	Part and State > Main > letter spacing	Part and State > Main > letter spacing	Letter space: the space between letters, a value between 0 and 100
Background color	Part and State > Main > Background color	Part and State > Main > Background color	Background color: it can be set by color picker or input RGB value
Opacity	Part and State > Main > Opacity	Part and State > Main > Opacity	The background opacity
Background gradient direction	Part and State > Main > Background gradient direction	Part and State > Main > Background gradient direction	Gradient direction of background color: vertical or horizontal
Padding top, bottom	Part and State > Main > Padding top, bottom	Part and State > Main > Padding top, bottom	Padding in top and bottom side: the value should be between 1 and 200
Padding left, right	Part and State > Main > Padding left, right	Part and State > Main > Padding left, right	Padding in left and right side: the value should be between 1 and 200
Radius	Part and State > Main > Radius	Part and State > Main > Radius	Radius of widget

4.22 LED

The LED is a rectangle-like (or circle) object. Its brightness can be adjusted. With lower brightness, the colors of the LED become darker.



Figure 42. LED in v7

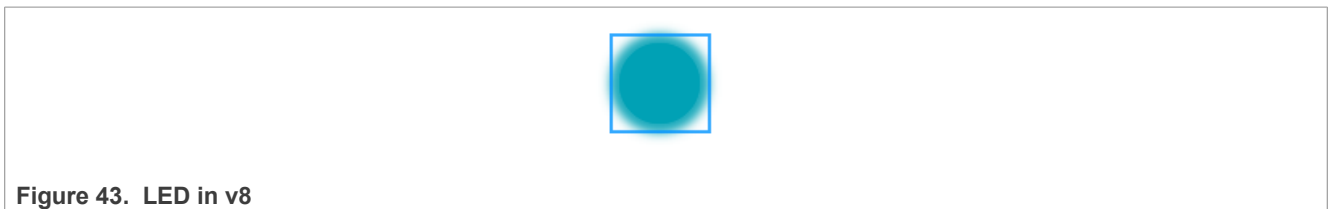


Figure 43. LED in v8

Table 30. LED properties

Property	v7	v8	Description
Bright	Attribute > Bright	N/A	Brightness of widget
Led Color	N/A	Attribute > LED Color	Color of widget: it can be set by color picker or input RGB value

4.23 Line

The line object helps draw straight lines between a set of points.

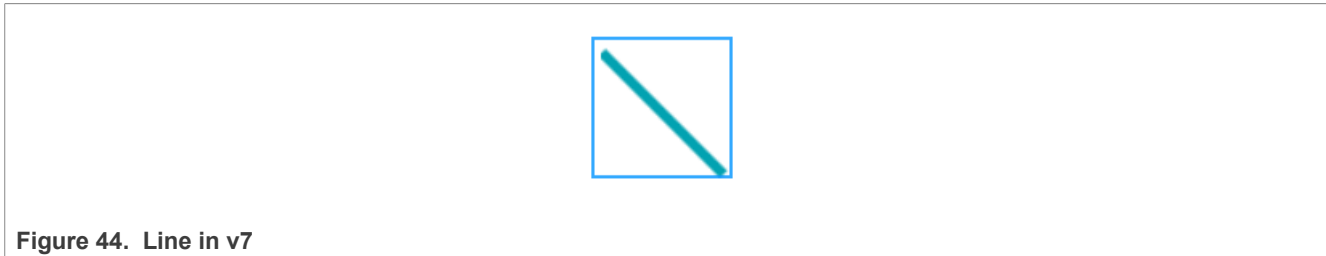


Figure 44. Line in v7

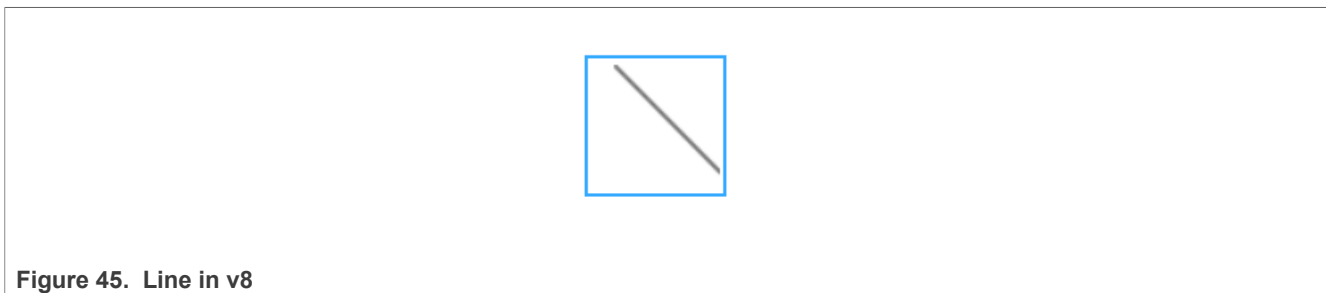


Figure 45. Line in v8

Table 31. Line properties

Property	v7	v8	Description
Line start	Attribute > Line start	N/A	Start point of the line
Line end	Attribute > Line end	N/A	End point of the line
line_point	Attribute > Line point	Attribute > Line point	Points connected in the line
Add points	Attribute > add points	Attribute > add points	Can add multiple points
state	Part and State > Main > state	Part and State > Main > state	Style of main part: it can be defined by one state or more states
Disable	Part and State > Main > Disable	Part and State > Main > Disable	Enable or disable a state: the custom settings are invalidated when a state is disabled
Line color	Part and State > Main > Line color	Part and State > Main > Line color	Line color: it can be set by color picker or input RGB value
Line width	Part and State > Main > Line width	Part and State > Main > Line width	
Line rounder	Part and State > Main > Line rounded	Part and State > Main > Line rounded	Enable line rounding function

4.24 Line meter

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



Figure 46. Line meter in v7

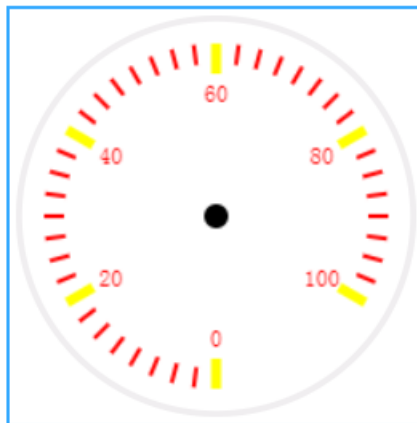


Figure 47. Line meter renamed as Meter in v8

Table 32. Line meter properties

Property	v7	Description
Min & Max	Attribute > Min & Max	
Line	Attribute > Line	
Angle	Attribute > Angle	
Offset	Attribute > Offset	
state	Part and State > Main > state	
Disable	Part and State > Main > Disable	
Background color	Part and State > Main > Background color	
Opacity	Part and State > Main > Opacity	max:255
Background gradient direction	Part and State > Main > Background gradient direction	
Radius	Part and State > Main > Radius	

Table 32. Line meter properties...continued

Property	v7	Description
padding top, bottom, left, right	Part and State > Main > padding top, bottom, left, right	max:20
Line color	Part and State > Main > Line color	
Line opacity	Part and State > Main > Line opacity	max:255
Line width	Part and State > Main > Line width	
Scale color	Part and State > Main > Scale color	
Scale width	Part and State > Main > Scale width	

4.25 List

The lists are built from a background [Page](#) and [Buttons](#) on it. The "Button" contain an optional icon-like [Image](#) (which can be a symbol too) and a [Label](#). Scrolling capability is included for lists that exceed the defined size.

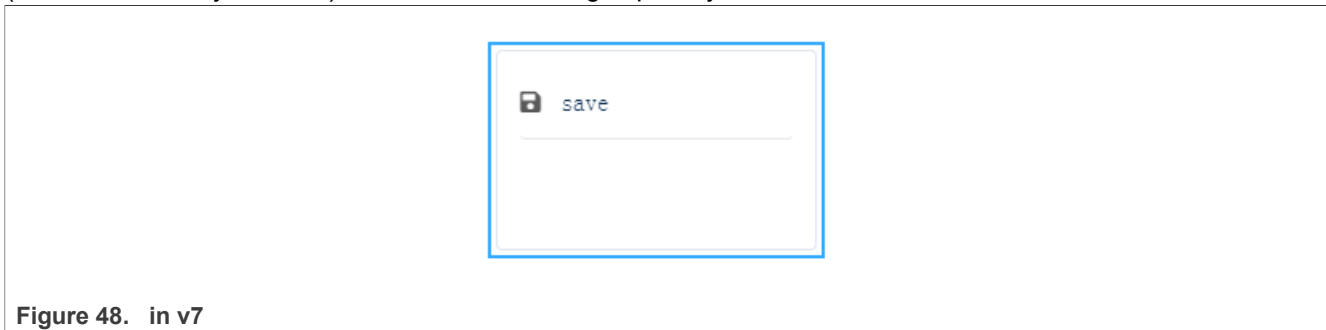


Figure 48. in v7

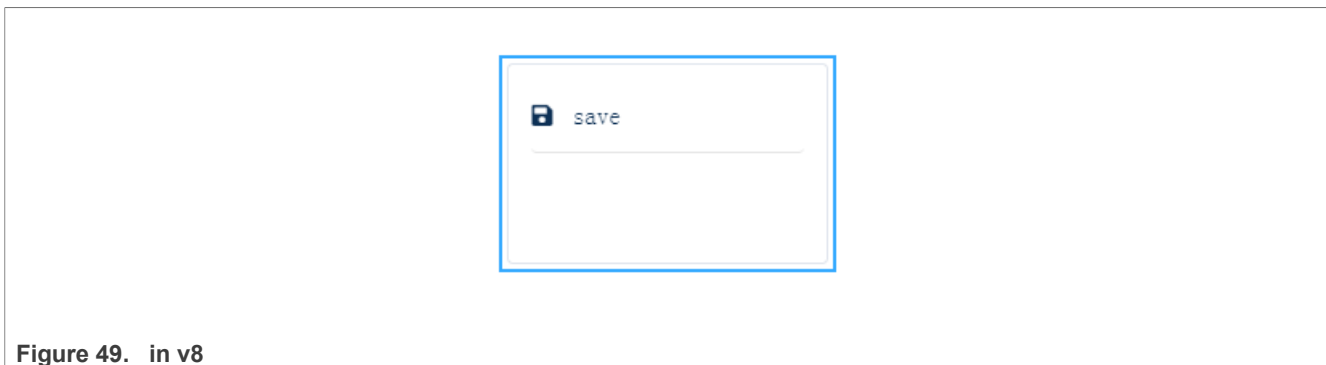


Figure 49. in v8

Table 33. List properties

Property	v7	v8	Description
Symbol	Attribute > items > Symbol	Attribute > items > Symbol	Symbol which is used as the button icon
image path	Attribute > items > Image path	Attribute > items > Image path	Image which is used as the button icon
Text	Attribute > items > Text	Attribute > items > Text	Text which is used as the button description
image size	Attribute > items > image width height	Attribute > items > image width, height	Example: '20, 20'
state	Part and State > Main > state	Part and State > Main > state	Style of main part: it can be defined by one state or more states

Table 33. List properties...continued

Property	v7	v8	Description
Disable	Part and State > Main > Disable	Part and State > Main > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Border color	Part and State > Main > Border color	Part and State > Main > Border color	Color of border: it can be set by color picker or input RGB value
Border width	Part and State > Main > Border width	Part and State > Main > Border width	max:50
Radius	Part and State > Main > Radius	Part and State > Main > Radius	max:2000
Padding top	Part and State > Main > Padding top	Part and State > Main > Padding top	max:200
Padding left, right	Part and State > Main > Padding left, right	Part and State > Main > Padding left, right	max:200
Background color	N/A	Part and State > Main > Background color	Background color of widget: it can be set by color picker or input RGB value
Opacity	N/A	Part and State > Main > Opacity	Background opacity: the value should be between 0 and 255
Background gradient direction	N/A	Part and State > Main > Background gradient direction	vertical or horizontal
state	Part and State > Scrollbar > state	Part and State > Scrollbar > state	Style of scroll part: it can be defined by one state or more states
Disable	Part and State > Scrollbar > Disable	Part and State > Scrollbar > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Background color	Part and State > Scrollbar > Background color	Part and State > Scrollbar > Background color	Background color of widget: it can be set by color picker or input RGB value
Opacity	Part and State > Scrollbar > Opacity	Part and State > Scrollbar > Opacity	max:255
Background gradient direction	Part and State > Scrollbar > Background gradient direction	Part and State > Scrollbar > Background gradient direction	Vertical or horizontal
state	Part and State > Buttons > state	Part and State > items > state	Style of list part: it can be defined by one state or more states
Disable	Part and State > Buttons > Disable	Part and State > items > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Text color	Part and State > Buttons > Text color	Part and State > items > Font color	Text color: it can be set by color picker or input RGB value
Font family, font size	Part and State > Buttons > Font family, font size	Part and State > items > Font family, font size	Font family and size settings: the size can be a value between 0 and 100
Background color	Part and State > Buttons > Background color	Part and State > items > Background color	Background color of widget: it can be set by color picker or input RGB value
Opacity	Part and State > Buttons > Opacity	Part and State > items > Opacity	max: 255

Table 33. List properties...continued

Property	v7	v8	Description
Background gradient direction	Part and State > Buttons > Background gradient direction	Part and State > items > Background gradient direction	Vertical or horizontal

4.26 Lottie

It is a light-weight implementation of animations by loading Lottie JSON files.



Figure 50. Lottie in v8

Table 34. Lottie properties

Property	v7	v8	Description
JsonFile	N/A	Attribute > json file	
Disable	N/A	Part and State > Main > Disable	
Background color	N/A	Part and State > Main > Background color	
Background gradient direction	N/A	Part and State > Main > Background gradient direction	

4.27 Menu

The menu widget can be used to easily create multi-level menus. It handles the traversal between pages automatically.

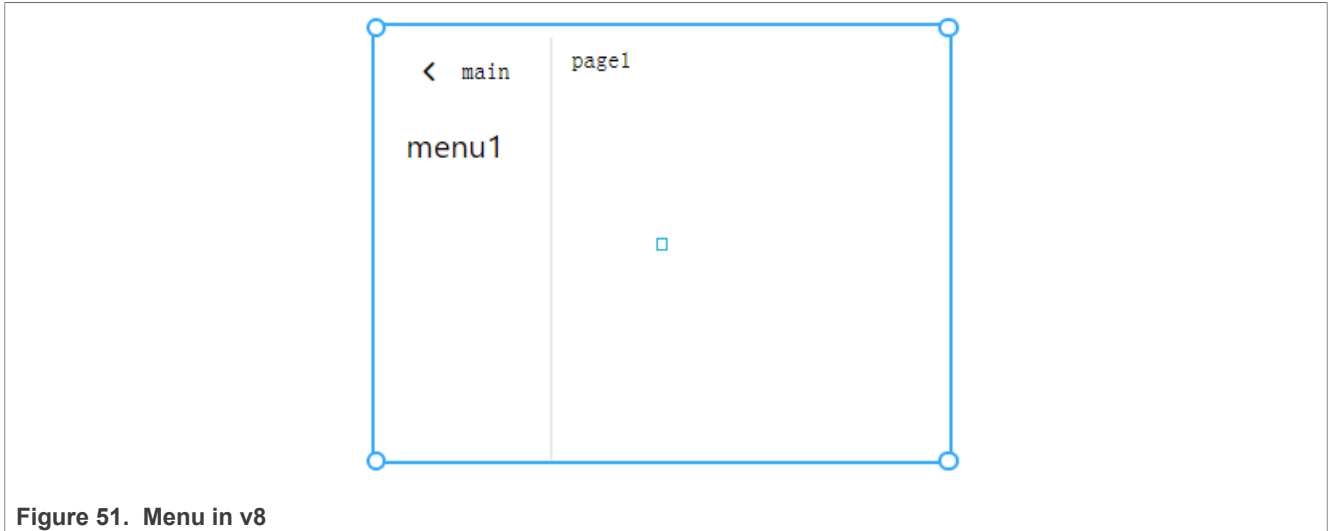


Figure 51. Menu in v8

Table 35. Menu properties

Property	v8	Description
title	Attribute > title	Add the title for widgets
mode	Attribute > mode	Top fixed, Top unfixed, Bottom fixed
Root back button	Attribute > Root back button	Enable the root back button
Enable page title	Attribute > Enable page title	Enable the page title display
Menu in side	Attribute > Menu in side	Enable the menu in side
state	Part and State > Main > state	Style of main part: it can be defined by one state or more states
Disable	Part and State > Main > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
background color	Part and State > Main > background color	Background color of widget: it can be set by color picker or input RGB value
Background gradient direction	Part and State > Main > Background gradient direction	Gradient direction of background color: vertical or horizontal
Opacity	Part and State > Main > Opacity	Background opacity: the value should be between 0 and 255
state	Part and State > Header > state	Style of main part: it can be defined by one state or more states
Disable	Part and State > Header > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
font color	Part and State > Header > font color	Set the Header font color
font family & size	Part and State > Header > font family & size	Set the Header font family and size
state	Part and State > Header back btn > state	Style of main part: it can be defined by one state or more states

Table 35. Menu properties...continued

Property	v8	Description
Disable	Part and State > Header back btn > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
font color	Part and State > Header back btn > font color	Set the Header back btn font color
font size	Part and State > Header back btn > font size	Set the Header back btn font size
state	Part and State > Sider header > state	Style of main part: it can be defined by one state or more states
Disable	Part and State > Sider header > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
font color	Part and State > Sider header > font color	Set the Sider header font color
font family & size	Part and State > Sider header > font family & size	Set the Sider header font family and size
state	Part and State > Sider back btn > state	Style of main part: it can be defined by one state or more states
Disable	Part and State > Sider back btn > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
font color	Part and State > Sider back btn > font color	Set the Sider back btn font color
font size	Part and State > Sider back btn > font size	Set the Sider back btn font size
state	Part and State > Page > state	Style of main part: it can be defined by one state or more states
Disable	Part and State > Page > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
background color	Part and State > Page > background color	Background color of widget: it can be set by color picker or input RGB value
Background gradient direction	Part and State > Page > Background gradient direction	Gradient direction of background color: vertical or horizontal
Opacity	Part and State > Page > Opacity	Background opacity: the value should be between 0 and 255
state	Part and State > Sub page > state	Style of main part: it can be defined by one state or more states
Disable	Part and State > Sub page > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
background color	Part and State > Sub page > background color	Background color of widget: it can be set by color picker or input RGB value
Background gradient direction	Part and State > Sub page > Background gradient direction	Gradient direction of background color: vertical or horizontal

Table 35. Menu properties...continued

Property	v8	Description
Opacity	Part and State > Sub page > Opacity	Background opacity: the value should be between 0 and 255
state	Part and State > Btn > state	Style of main part: it can be defined by one state or more states
Disable	Part and State > Btn > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
background color	Part and State > Btn > background color	Background color of widget: it can be set by color picker or input RGB value
Background gradient direction	Part and State > Btn > Background gradient direction	Gradient direction of background color: vertical or horizontal
Opacity	Part and State > Btn > Opacity	Background opacity: the value should be between 0 and 255
font color	Part and State > Btn > Font color	Set the Btn font color

4.28 Message box

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

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



Figure 52. Message box in v7

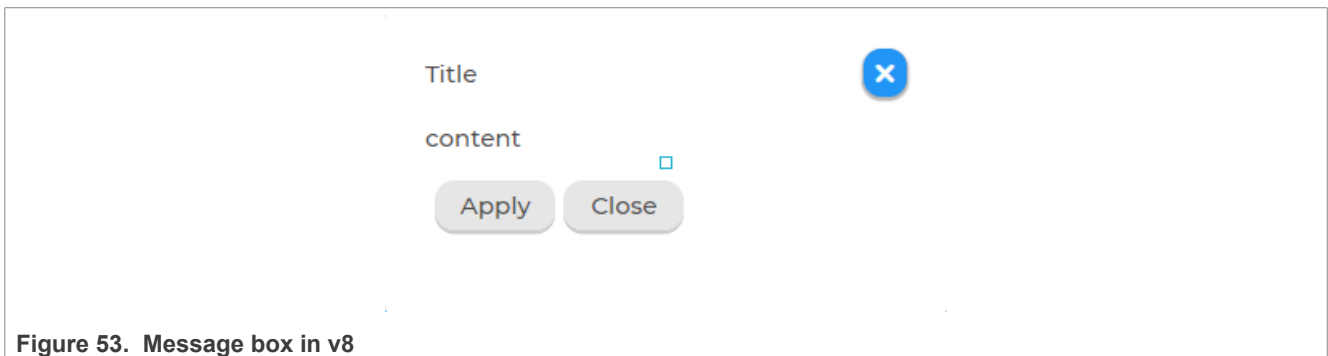


Figure 53. Message box in v8

Table 36. Message box properties

Property	v7	v8	Description
Mbx Button	Attribute > Mbx Button	x	Button label in message box
Mbox Text	Attribute > Mbox Text	Attribute > Mbox Text	Message box texts
Mbox Title	N/A	Attribute > Mbox Title	Message box title
Button width, height	N/A	Attribute > Button width, height	Button width, height
Show close btn	N/A	Attribute > Show close btn	The message box can be closed automatically after delay milliseconds with an animation
state	Part and State > Main > state	Part and State > Main > state	Style of main part: it can be defined by one state or more states
Disable	Part and State > Main > Disable	Part and State > Main > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Text color	Part and State > Main > Text color	N/A	Text color in main part: it can be set by color picker or input RGB value
Font family, font size	Part and State > Main > Font family, font size	N/A	Font family and size settings: the size can be a value between 0 and 100
Background color	Part and State > Main > Background color	Part and State > Main > Background color	Background color of main part: it can be set by color picker or input RGB value
Opacity	Part and State > Main > Opacity	Part and State > Main > Opacity	max: 255
Background gradient direction	Part and State > Main > Background gradient direction	Part and State > Main > Background gradient direction	Vertical or horizontal
Border color	Part and State > Main > Border color	Part and State > Main > Border color	Border color
Border width	Part and State > Main > Border width	Part and State > Main > Border width	v7max: 4, v8max: 1
Radius	Part and State > Main > Radius	Part and State > Main > Radius	v7max: 20, v8max: 8
state	Part and State > Button background > state	N/A	Style of button bar part: it can be defined by one state or more states
Disable	Part and State > Button background > Disable	N/A	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Background color	Part and State > Button background > Background color	N/A	Background color of button bar part: it can be set by color picker or input RGB value
Opacity	Part and State > Button Bar > Opacity	N/A	max: 255
Background gradient direction	Part and State > Button background > Background gradient direction	N/A	Vertical or horizontal
Border color	Part and State > Button background > Border color	N/A	Border color

Table 36. Message box properties...continued

Property	v7	v8	Description
Border width	Part and State > Button background > Border width	N/A	max: 4
Radius	Part and State > Button background > Radius	N/A	max: 20
state	Part and State > Button > state	Part and State > btns > state	Style of button part: it can be defined by one state or more states
Disable	Part and State > Button > Disable	Part and State > btns > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Text color	Part and State > Button > Text color	Part and State > btns > Font color	Text color in button label: it can be set by color picker or input RGB value
Font family, font size	Part and State > Button > Font family, font size	Part and State > btns > Font family, font size	Font family and size settings in the button label: the size can be a value between 0 and 100
Background color	Part and State > Button > Background color	Part and State > btns > Background color	Background color of button part: it can be set by color picker or input RGB value
Opacity	Part and State > Button > Opacity	Part and State > btns > Opacity	max: 255
Background gradient direction	Part and State > Button > Background gradient direction	Part and State > btns > Background gradient direction	Vertical or horizontal
Border color	Part and State > Button > Border color	Part and State > btns > Border color	Border color
Border width	Part and State > Button > Border width	Part and State > btns > Border width	max: 4
Radius	Part and State > Button > Radius	Part and State > btns > Radius	v7max: 20, v8max: 8
state	N/A	Part and State > Title > state	Style of title part: it can be defined by one state or more states
Disable	N/A	Part and State > Title > Disable	Enable or disable a state: The custom settings are invalidated when a state is disabled
Text color	N/A	Part and State > Title > Font color	Font color in title: It can be set by color picker or input RGB value
Font family, font size	N/A	Part and State > Title > Font family, font size	Font family and size settings in the title: the size can be a value between 0 and 100
Line space & Letter space	N/A	Part and State > Title > Line space & Letter space	Line space and letter space
state	N/A	Part and State > Content > state	Style of content part: it can be defined by one state or more states
Disable	N/A	Part and State > Content > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Text color	N/A	Part and State > Content > Font color	Font color in content: it can be set by color picker or input RGB value
Font family, font size	N/A	Part and State > Content > Font family, font size	Font family and size settings in the content: the size can be a value between 0 and 100

Table 36. Message box properties...continued

Property	v7	v8	Description
Line space & Letter space	N/A	Part and State > Content > Line space & Letter space	Line space and letter space

4.29 QR code

It generates a QR code based on the input text, typically used for storing URLs or information.



Figure 54. QR code in v8

Table 37. QR code properties

Property	v7	v8	Description
text	N/A	Attribute > text	
State	N/A	Part and State > Main	
Disable	N/A	Part and State > Main > Disable	
Background color	N/A	Part and State > Main > Background color	
Background gradient direction	N/A	Part and State > Main > Background gradient direction	

4.30 Radio button

The radio button allows user to choose only one of a predefined set of mutually exclusive options.



Figure 55. Radio button in v8

Table 38. Radio button properties

Property	v8	Description
item	Attribute > add item	Add the item for widgets
state	Part and State > Main > state	Style of main part: it can be defined by one state or more states

Table 38. Radio button properties...continued

Property	v8	Description
Disable	Part and State > Main > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
background color	Part and State > Main > background color	Background color of widget: it can be set by color picker or input RGB value
Background gradient direction	Part and State > Main > Background gradient direction	Gradient direction of background color: vertical or horizontal
Opacity	Part and State > Main > Opacity	Background opacity: the value should be between 0 and 255
border color	Part and State > Main > border color	Set the border color
border opacity	Part and State > Main > border opacity	set the opacity of the widget border
border width	Part and State > Main > border width	set the border width
border radius	Part and State > Main > border radius	set the border radius
padding	Part and State > Main > padding	top, right, bottom, left
state	Part and State > Button > state	Style of main part: it can be defined by one state or more states
Disable	Part and State > Button > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
background color	Part and State > Button > background color	Button color of widget: it can be set by color picker or input RGB value
Background gradient direction	Part and State > Button > Background gradient direction	Gradient direction of button color: vertical or horizontal
Opacity	Part and State > Button > Opacity	Button opacity: the value should be between 0 and 255
font color	Part and State > Button > font color	Set the button font color
font family & size	Part and State > Button > font family & size	Set the button font family and size
state	Part and State > indicator > state	Style of main part: it can be defined by one state or more states
Disable	Part and State > indicator > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
background color	Part and State > indicator > background color	Indicator color of widget: it can be set by color picker or input RGB value
Background gradient direction	Part and State > indicator > Background gradient direction	Gradient direction of indicator color: vertical or horizontal
Opacity	Part and State > indicator > Opacity	Indicator opacity: the value should be between 0 and 255
border color	Part and State > indicator > border color	Set the indicator border color
border opacity	Part and State > indicator > border opacity	Set the indicator border opacity

Table 38. Radio button properties...continued

Property	v8	Description
border width	Part and State > indicator > border width	Set the indicator border radius
border radius	Part and State > indicator > border radius	Set the indicator border radius

4.31 Meter

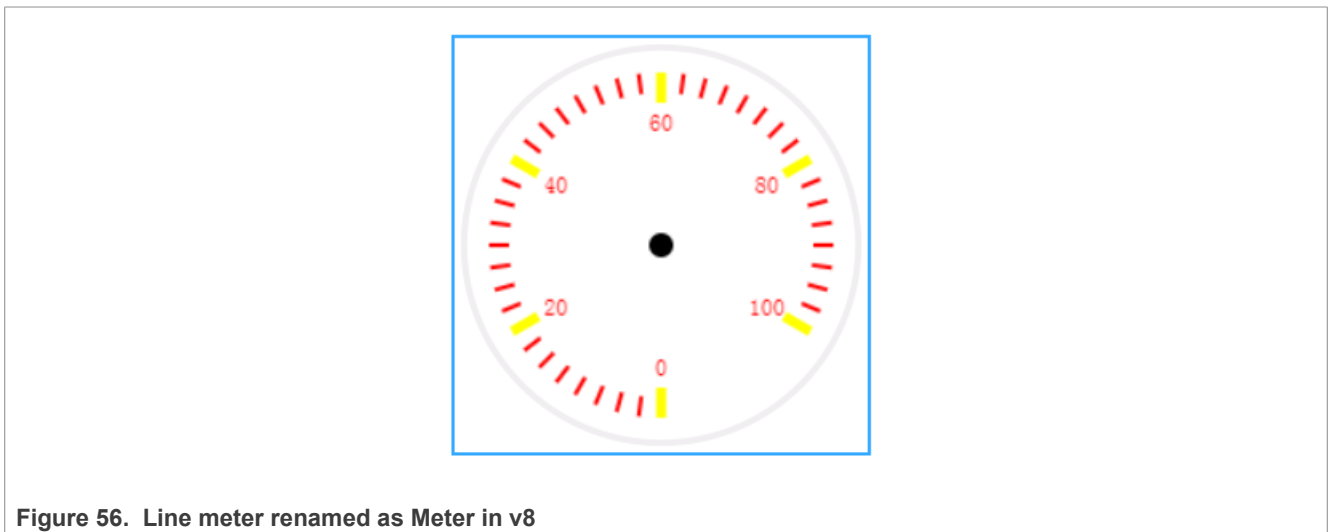


Figure 56. Line meter renamed as Meter in v8

Table 39. Meter properties

Property	v8	Description
Add Dial	Attribute > Add dial	Add multiple dials
Label gap	Attribute > Label gap	Gap between label and radial lines
Tick count & width	Attribute > Tick count & width	Tick counter and width
Tick length & Color	Attribute > Tick length & Color	Tick length and color
Major enable	Attribute > Major enable	Make some "normal" ticks major ticks
Major index & width	Attribute > Major index & width	major ticket index and width
Major length& Color	Attribute > Major length & Color	major ticket length and color
Range enable	Attribute > Range enable	Set the value and angular range of a scale
Start angle	Attribute > Start angle	Start angles
Angle range	Attribute > Angle range	Angle range
Range min & max	Attribute > Range min & max	Range min and max
Needle add	Attribute > Needle add	Add needle
Needle image add	Attribute > Needle image add	Set image as needle background
Arcs add	Attribute > Arcs add	Add arcs
Scale Lines add	Attribute > Scale Lines add	Add scale lines

Table 39. Meter properties...continued

Property	v8	Description
state	Part and State > Main > state	Style of the main part: it can be defined by one state or more states
Disable	Part and State > Main > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Background color	Part and State > Main > Background color	Background color of main part: it can be set by color picker or input RGB value
Opacity	Part and State > Main > Opacity	max: 255
Background gradient direction	Part and State > Main > Background gradient direction	Vertical or horizontal
state	Part and State > digit > state	Style of digit part: it can be defined by one state or more states
Disable	Part and State > digit > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
font color	Part and State > digit > font color	Font color in digit part: it can be set by color picker or input RGB value
font size, font family	Part and State > digit > font size, font family	Font family and size settings in the digit part: the size can be a value between 0 and 100

4.32 Roller

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

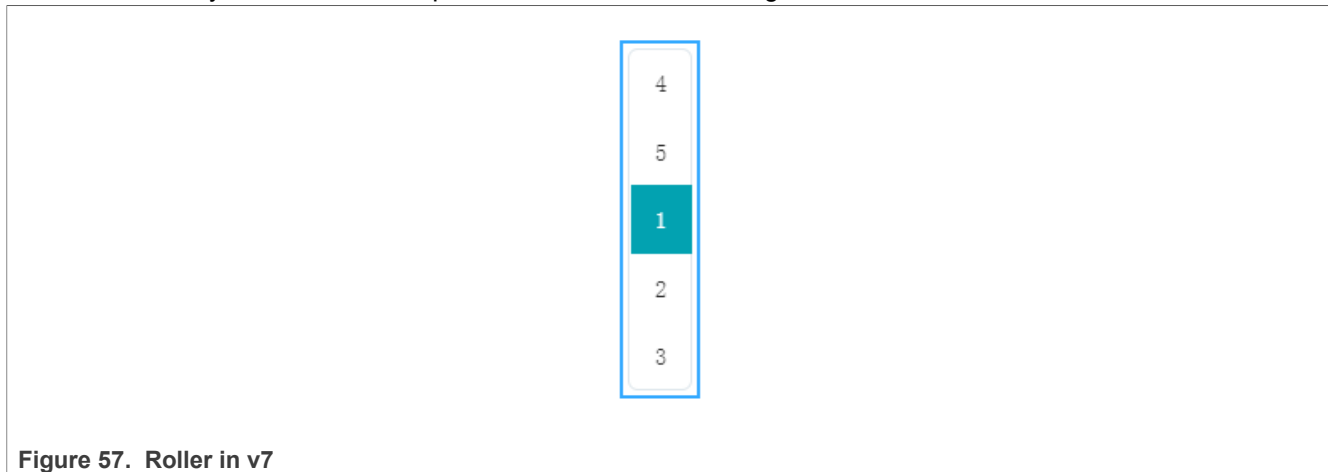


Figure 57. Roller in v7

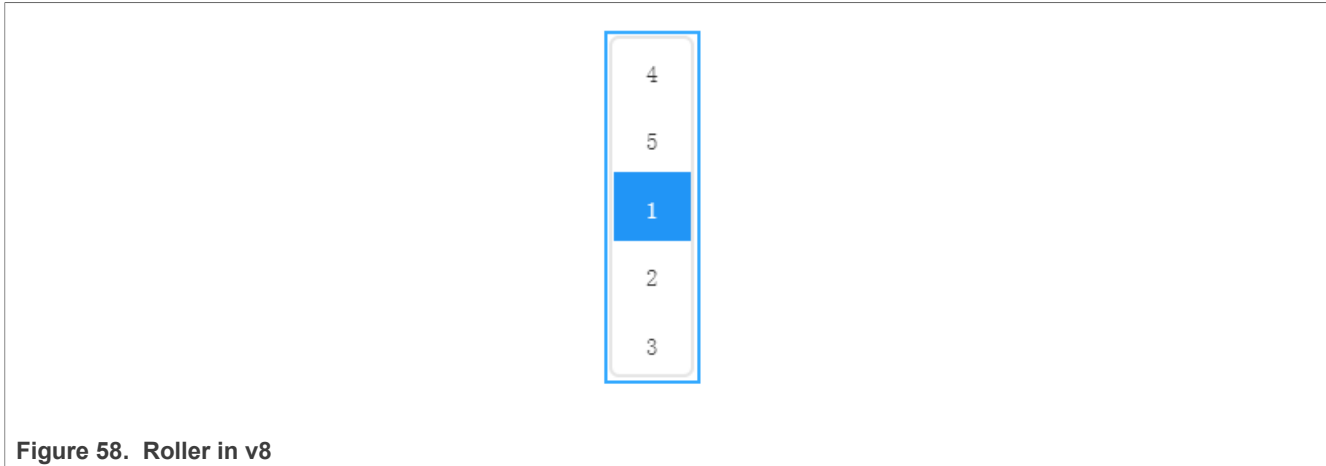


Figure 58. Roller in v8

Table 40. Roller properties

Property	v7	v8	Description
Direction	Attribute > direction	Attribute > direction	Roller mode: Normal mode (roller ends at the end of the options), Infinite mode (roller can be scrolled forever)
Row text	Attribute > Row text	Attribute > Row text	Roller option items, for example, 1,2,3,4,5
Row	Attribute > Row	Attribute > Row	Roller option count
state	Part and State > Background > state	Part and State > Main > state	Style of main part: it can be defined by one state or more states
Disable	Part and State > Background > Disable	Part and State > Main > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Font color	Part and State > Background > Font color	Part and State > Background > Font color	Font color: it can be set by color picker or input RGB value
Font size	Part and State > Background > Font size	Part and State > Background > Font size	
Font family	Part and State > Background > Font family	Part and State > Background > Font family	
Background color	Part and State > Background > Background color	Part and State > Main > Background color	Background color: it can be set by color picker or input RGB value
Opacity	Part and State > Background > Opacity	Part and State > Main > Opacity	max: 255
Background gradient direction	Part and State > Background > Background gradient direction	Part and State > Main > Background gradient direction	Vertical or horizontal
Border color	Part and State > Background > Border color	Part and State > Main > Border color	Border color
Border width	Part and State > Background > Border width	Part and State > Main > Border width	max: 2
Radius	Part and State > Background > Radius	Part and State > Main > Radius	Radius
state	Part and State > Selected > State	Part and State > Selected > state	Style of indicator part: it can be defined by one state or more states

Table 40. Roller properties...continued

Property	v7	v8	Description
Disable	Part and State > Selected > Disable	Part and State > Selected > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Background color	Part and State > Selected > Line color	Part and State > Selected > Line color	Line color
Opacity	Part and State > Selected > Opacity	Part and State > Selected > Opacity	max: 255
Background gradient direction	Part and State > Selected > Background gradient direction	Part and State > Selected > Background gradient direction	Vertical or horizontal
Font size & family	Part and State > Selected > Font size & family	Part and State > Selected > Font size & family	

4.33 Slider

The slider object looks like a [Bar](#) supplemented with a knob. The knob can be dragged to set a value. The slider can be vertical or horizontal.

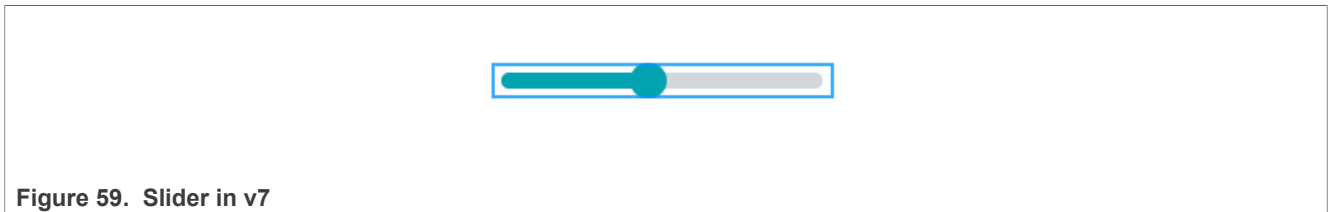


Figure 59. Slider in v7

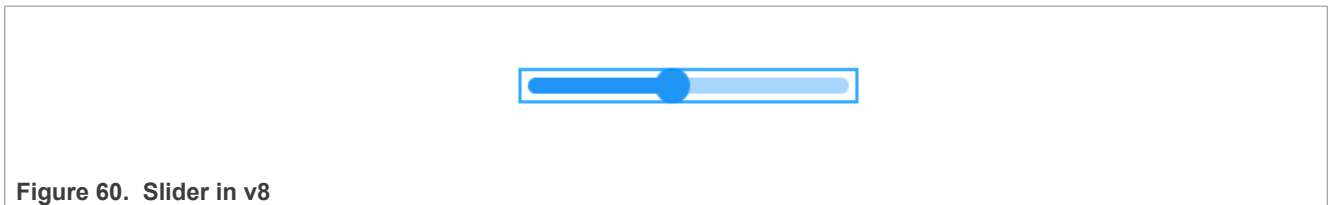


Figure 60. Slider in v8

Table 41. Slider properties

Property	v7	v8	Description
Max& Init Value	Attribute > Max& Init Value	Attribute > Max& Init Value	To set an initial value in special range value (Min: 0, Max: 100)
state	Part and State > Main > state	Part and State > Main > state	Style of highlight part: it can be defined by one state or more states
Disable	Part and State > Main > Disable	Part and State > Main > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Background color	Part and State > Main > Background color	Part and State > Main > Background color	Background color of widget: it can be set by color picker or input RGB value
Opacity	Part and State > Main > Opacity	Part and State > Main > Opacity	Background opacity: the value should be between 0 and 255
Background gradient direction	Part and State > Main > Background gradient direction	Part and State > Main > Background gradient direction	Gradient direction of background color: vertical or horizontal

Table 41. Slider properties...continued

Property	v7	v8	Description
Border color	Part and State > Main > Border color	N/A	Color of border: it can be set by color picker or input RGB value
Border opacity	Part and State > Main > Border opacity	N/A	Background opacity of head part: the value should be between 0 and 255
Border Radius	Part and State > Main > Border Radius	N/A	Radius of Border: 0-200
Outline color	N/A	Part and State > Main > Outline color	Color of Outline: it can be set by color picker or input RGB value
Outline opacity	N/A	Part and State > Main > Outline opacity	Outline opacity: the value should be between 0 and 255
Outline width	N/A	Part and State > Main > Outline width	Border width of outline: the value can be 1 or 5
Outline Radius	N/A	Part and State > Main > Outline Radius	Radius of outline: 0-200
state	Part and State > Indicator > state	Part and State > Indicator > state	Style of highlight part: it can be defined by one state or more states
Disable	Part and State > Indicator > Disable	Part and State > Indicator > Disable	Enable or disable a state: the custom settings are invalidated when a state is disabled
Background color	Part and State > Indicator > Background color	Part and State > Indicator > Background color	Background color of widget: it can be set by color picker or input RGB value
Opacity	Part and State > Indicator > Opacity	Part and State > Indicator > Opacity	Background opacity: the value should be between 0 and 255
Background gradient direction	Part and State > Indicator > Background gradient direction	Part and State > Indicator > Background gradient direction	Gradient direction of background color: vertical or horizontal
state	Part and State > Knob > state	Part and State > Knob > state	Style of highlight part: it can be defined by one state or more states
Disable	Part and State > Knob > Disable	Part and State > Knob > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Background color	Part and State > Knob > Background color	Part and State > Knob > Background color	Background color of widget: it can be set by color picker or input RGB value
Opacity	Part and State > Knob > Opacity	Part and State > Knob > Opacity	Background opacity: the value should be between 0 and 255
Background gradient direction	Part and State > Knob > Background gradient direction	Part and State > Knob > Background gradient direction	Gradient direction of background color: vertical or horizontal

4.34 Spangroup

A spangroup is the object that is used to display rich text. Different from the label object, spangroup can render text styled with different fonts, colors, and sizes into the spangroup object.

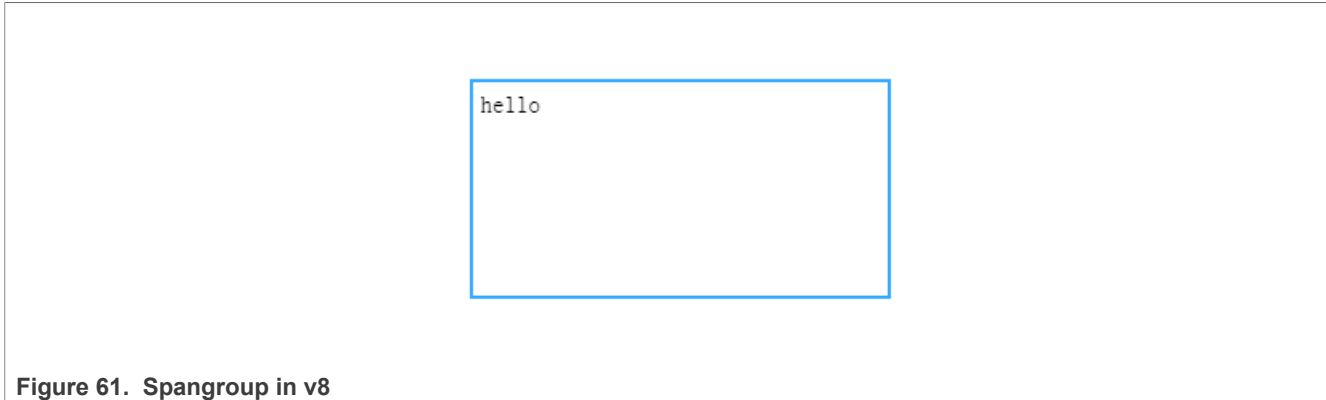


Figure 61. Spangroup in v8

Table 42. Spangroup properties

Property	v8	Description
Text	Attribute > Text	The object that is used to display rich text
Text align	Attribute > Text align	Set the span group to different align mode (left/right/center/auto)
Mode	Attribute > Mode	Set the span group to different mode (Fixed/Expand/Break)
items	Attribute > items	Add one new item of Text
font color	Attribute > font color	Text color in disabled part: it can be set by color picker or input RGB value
Font family, font size	Attribute > Font family, font size	Font family and size settings in the window content: the size can be a value between 0 and 100
state	Part and State > Main > state	State of header part: it can be defined by one state or more states
Disable	Part and State > Main > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Background color	Part and State > Main > Background color	Background color of Symbol part: it can be set by color picker or input RGB value
Opacity	Part and State > Main > Opacity	Background opacity of Symbol part: the value should be between 0 and 255
Background gradient direction	Part and State > Main > Background gradient direction	Gradient direction of background color in Symbol part: vertical or horizontal
Border color	Part and State > Main > Border color	Border color in ticks part: it can be set by color picker or input RGB value
Border width	Part and State > Main > Border width	Width of border line: value can be one value from 0 to 1
padding top, bottom, left, right	Part and State > Main > padding top, bottom, left, right	The padding value of top, bottom, left, and right; value can be one value from 0 to 200

4.35 Spinbox

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](#).

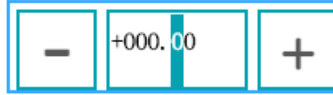


Figure 62. Spinbox in v7

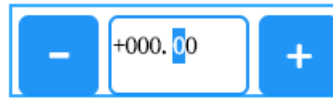


Figure 63. Spinbox in v8

Table 43. Spinbox properties

Property	v7	v8	Description
Digit	Attribute > Digit	Attribute > Digit	Sets the number format
state	Part and State > Main > state	Part and State > Main > state	Style of highlight part: it can be defined by one state or more states
Disable	Part and State > Main > Disable	Part and State > Main > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Text color	Part and State > Main > Text color	Part and State > Main > Font color	Text color in head part: it can be set by color picker or input RGB value
Font size, font family	Part and State > Main > Font size, font family	Part and State > Main > Font size, font family	Font family and size settings in the calendar header: The size can be a value between 0 and 100
Background color	Part and State > Main > Background color	Part and State > Main > Background color	Background color of widget: it can be set by color picker or input RGB value
Opacity	Part and State > Main > Opacity	Part and State > Main > Opacity	Background opacity: the value should be between 0 and 255
Background gradient direction	Part and State > Main > Background gradient direction	Part and State > Main > Background gradient direction	Gradient direction of background color: vertical or horizontal
Border color	Part and State > Main > Border color	Part and State > Main > Border color	Color of border: it can be set by color picker or input RGB value
Border width	Part and State > Main > Border width	Part and State > Main > Border width	Border width: the value can be 1 or 5
Radius	Part and State > Main > Radius	Part and State > Main > Radius	Radius of Border: 0-200
state	Part and State > Cursor > state	Part and State > Cursor > state	Style of highlight part: it can be defined by one state or more states
Disable	Part and State > Cursor > Disable	Part and State > Cursor > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.

Table 43. Spinbox properties...continued

Property	v7	v8	Description
Font size, font family	Part and State > Cursor > Font size, font family	Part and State > Cursor > Font size, font family	Font family and size settings in the calendar header: the size can be a value between 0 and 100
Background color	Part and State > Cursor > Background color	Part and State > Cursor > Background color	Background color of widget: it can be set by color picker or input RGB value
Opacity	Part and State > Cursor > Opacity	Part and State > Cursor > Opacity	Background opacity: the value should be between 0 and 255
Background gradient direction	Part and State > Cursor > Background gradient direction	Part and State > Cursor > Background gradient direction	Gradient direction of background color: vertical or horizontal
state	Part and State > Buttons > state	Part and State > Button > state	Style of highlight part: it can be defined by one state or more states
Disable	Part and State > Buttons > Disable	Part and State > Button > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Font size, font family	Part and State > Buttons > Font size, font family	Part and State > Button > Font size, font family	Font family and size settings in the calendar header: the size can be a value between 0 and 100
Background color	Part and State > Buttons > Background color	Part and State > Button > Background color	Background color of widget: it can be set by color picker or input RGB value
Opacity	Part and State > Buttons > Opacity	Part and State > Button > Opacity	Background opacity: the value should be between 0 and 255
Background gradient direction	Part and State > Buttons > Background gradient direction	Part and State > Button > Background gradient direction	Gradient direction of background color: vertical or horizontal
Border color	Part and State > Buttons > Border color	Part and State > Button > Border color	Color of border: it can be set by color picker or input RGB value
Border width	Part and State > Buttons > Border width	Part and State > Button > Border width	Border width: the value can be 1 or 5
Radius	Part and State > Buttons > Radius	Part and State > Button > Radius	Radius of Border: 0-200

4.36 Spinner

The spinner object is a spinning arc over a border.



Figure 64. Spinner in v7



Figure 65. Spinner in v8

Table 44. Spinner properties

Property	v7	v8	Description
Length	Attribute > Length	Attribute > Length	Sets the length of the spinning arc in degrees: the max: 359
Time	Attribute > Time	Attribute > Time	Sets the spin time in milliseconds
state	Part and State > Main > state	Part and State > Main > state	Style of highlight part: it can be defined by one state or more states
Disable	Part and State > Main > Disable	Part and State > Main > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Background color	Part and State > Main > Background color	Part and State > Main > Background color	Background color of widget: it can be set by color picker or input RGB value
Opacity	Part and State > Main > Opacity	Part and State > Main > Opacity	Background opacity: the value should be between 0 and 255
Background gradient direction	Part and State > Main > Background gradient direction	Part and State > Main > Background gradient direction	Gradient direction of background color: vertical or horizontal
padding top, bottom, left, right	Part and State > Main > padding top, bottom, left, right	Part and State > Main > padding top, bottom, left, right	Padding of the scale: 0-200
Line color	Part and State > Main > Line color	Part and State > Main > Arc color	Color of Line: it can be set by color picker or input RGB value
Line width	Part and State > Main > Line width	Part and State > Main > Arc width	Line width: the value can be 1 or 20
Line rounded	Part and State > Main > Line rounded	Part and State > Main > Line rounded	Enable the line rounded
state	Part and State > indicator > state	Part and State > indicator > state	Style of highlight part: it can be defined by one state or more states
Disable	Part and State > indicator > Disable	Part and State > indicator > Disable	Enable or disable a state: the custom settings are invalidated when a state is disabled
Line color	Part and State > indicator > Line color	Part and State > indicator > Arc color	Color of Line: it can be set by color picker or input RGB value
Line width	Part and State > indicator > Line width	Part and State > indicator > Arc width	Line width: the value can be 1 or 20
Line rounded	Part and State > Main > Line rounded	Part and State > Main > Line rounded	Enable the line rounded

4.37 Switch

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

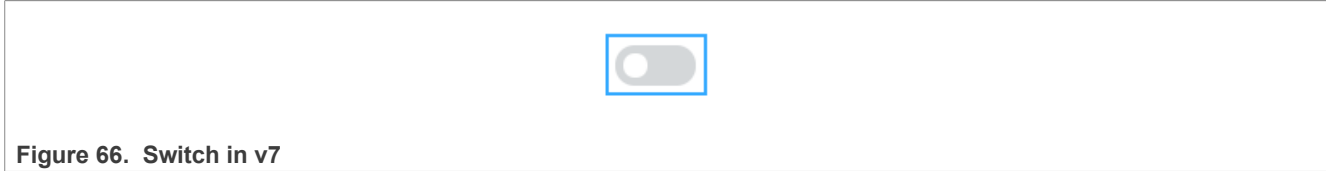


Figure 66. Switch in v7

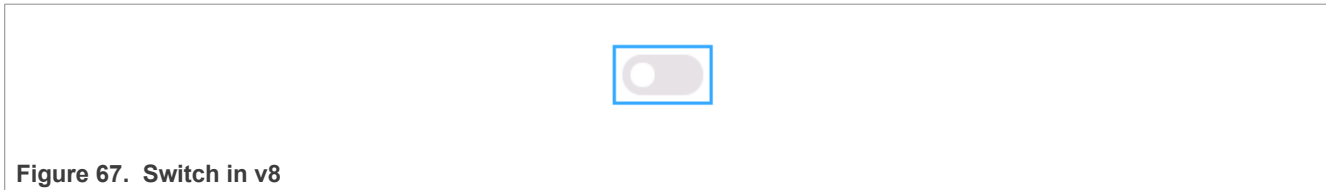


Figure 67. Switch in v8

Table 45. Switch properties

Property	v7	v8	Description
Animtime	Attribute > Animtime	N/A	Set the time of animations, when the switch changes state
state	Part and State > Main > state	Part and State > Main > state	Style of highlight part: it can be defined by one state or more states
Disable	Part and State > Main > Disable	Part and State > Main > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Background color	Part and State > Main > Background color	Part and State > Main > Background color	Background color of widget: it can be set by color picker or input RGB value
Opacity	Part and State > Main > Opacity	Part and State > Main > Opacity	Background opacity: the value should be between 0 and 255
Background gradient direction	Part and State > Main > Background gradient direction	Part and State > Main > Background gradient direction	Gradient direction of background color: vertical or horizontal
Radius	Part and State > Main > Radius	Part and State > Main > Radius	Radius of Border: 0-200
state	Part and State > Indicator > state	Part and State > Indicator > state	Style of highlight part: it can be defined by one state or more states
Disable	Part and State > Indicator > Disable	Part and State > Indicator > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Background color	Part and State > Indicator > Background color	Part and State > Indicator > Background color	Background color of widget: it can be set by color picker or input RGB value
Opacity	Part and State > Indicator > Opacity	Part and State > Indicator > Opacity	Background opacity: the value should be between 0 and 255
Background gradient direction	Part and State > Indicator > Background gradient direction	Part and State > Indicator > Background gradient direction	Gradient direction of background color: vertical or horizontal
Radius	Part and State > Indicator > Radius	Part and State > Indicator > Radius	Radius of Border: 0-200
state	Part and State > Knob > state	Part and State > Knob > state	Style of highlight part: it can be defined by one state or more states

Table 45. Switch properties...continued

Property	v7	v8	Description
Disable	Part and State > Knob > Disable	Part and State > Knob > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Background color	Part and State > Knob > Background color	Part and State > Knob > Background color	Background color of widget: it can be set by color picker or input RGB value
Opacity	Part and State > Knob > Opacity	Part and State > Knob > Opacity	Background opacity: the value should be between 0 and 255
Background gradient direction	Part and State > Knob > Background gradient direction	Part and State > Knob > Background gradient direction	Gradient direction of background color: vertical or horizontal

4.38 Table

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

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



Figure 68. Table in v7

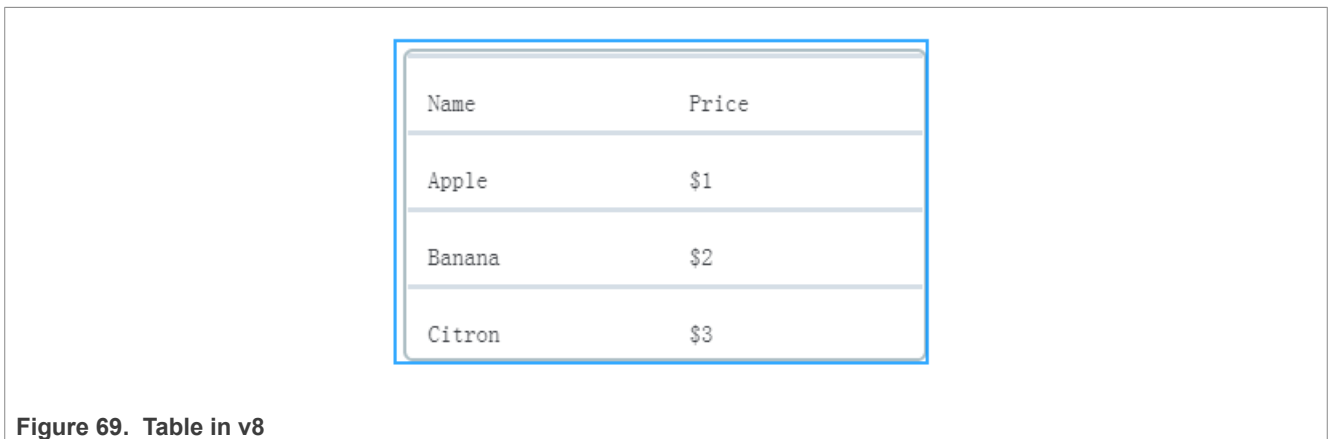


Figure 69. Table in v8

Table 46. Table properties

Property	v7	v8	Description
Column and row	Attribute > Column and row	Attribute > Column and row	Column number and row number of table: they should be one value from 1 to 300

Table 46. Table properties...continued

Property	v7	v8	Description
Column and row content	Attribute > Column and row content	Attribute > Column and row content	Content of column and row: the contents of each row and column are separated by commas
state	Part and State > main > state	Part and State > main > state	State of main part: it can be defined by one state or more states
Disable	Part and State > main > Disable	Part and State > main > Disable	Enable or disable a state, the custom settings are invalidated when a state is disabled.
Font color	Part and State > main > Font color	N/A	Text color of item: it can be set by color picker or input RGB value
Font family & size	Part and State > main > Font family & size	N/A	
Background color	Part and State > main > Background color	Part and State > main > Background color	Background color of widget: it can be set by color picker or input RGB value
Opacity	Part and State > main > Opacity	Part and State > main > Opacity	Background opacity: the value should be between 0 and 255
Background gradient direction	Part and State > main > Background gradient direction	Part and State > main > Background gradient direction	Gradient direction of background color: vertical or horizontal
border color	Part and State > main > border color	Part and State > main > border color	Color of border: it can be set by color picker or input RGB value
border width	Part and State > main > border width	Part and State > main > border width	Width of border line: value can be one value from 0 to 2
padding top, bottom, left, right	Part and State > main > padding top, bottom, left, right	Part and State > main > padding top, bottom, left, right	The padding value of top, bottom, left, and right; value can be one value from 0 to 200
state	Part and State > items > state	Part and State > items > state	State of items part: it can be defined by one state or more states
Disable	Part and State > items > Disable	Part and State > items > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
border color	Part and State > items > Border color	Part and State > items > Border color	Color of item border: it can be set by color picker or input RGB value
Border width	Part and State > items > Border width	Part and State > items > Border width	Width of item border line: value can be one value from 0 to 2
Background color	N/A	Part and State > items > Background color	Background color of items part: it can be set by color picker or input RGB value
Opacity	N/A	Part and State > items > Opacity	Background opacity: the value should be between 0 and 255
Background gradient direction	N/A	Part and State > items > Background gradient direction	Gradient direction of background color: vertical or horizontal
Font color	N/A	Part and State > items > Font color	Text color of item: it can be set by color picker or input RGB value
Font family, font size	N/A	Part and State > items > Font family, font size	Font family and size settings of item content: the size can be a value between 0 and 100

4.39 Tab view

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

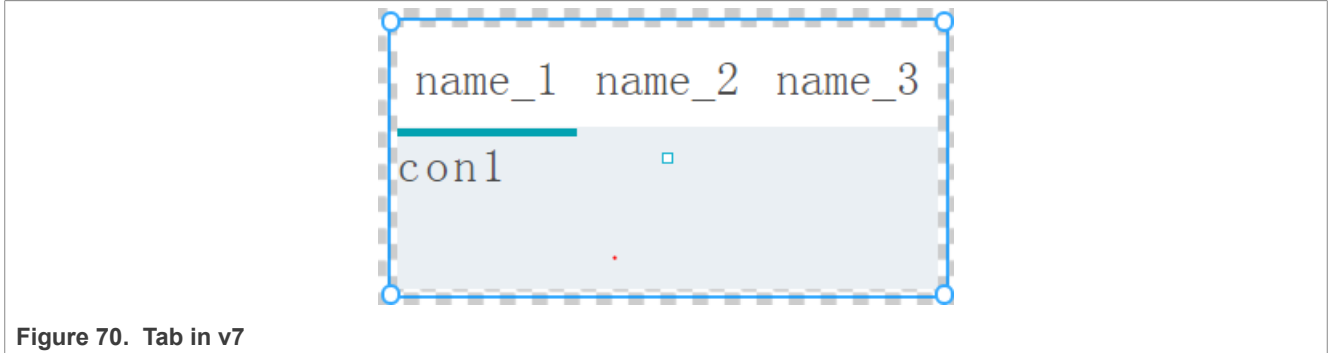


Figure 70. Tab in v7

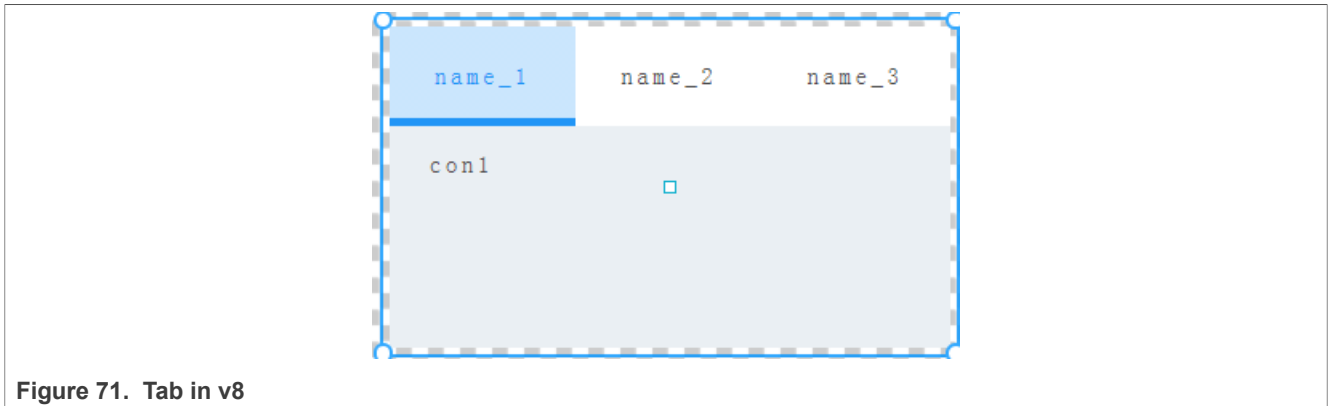


Figure 71. Tab in v8

Table 47. Tab view properties

Property	v7	v8	Description
Animation	Attribute > Animation	N/A	Animation time of tab view when a new tab is loaded: the value should be one value from 0 to 3000
Contents	Attribute > Contents	Attribute > Contents	The title and content of tab page
Tab Height	N/A	Attribute > Tab size	The height of Tab title part
Tab Position	N/A	Attribute > Tab Position	The position of tab: the value can be top, bottom, left, and right
state	Part and State > Main > state	Part and State > Main > state	State of main part: it can be defined by one state or more states
Disable	Part and State > Main > Disable	Part and State > Main > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled
Text color	Part and State > Main > Text color	Part and State > Main > Text color	Content text color in each tab page: it can be set by color picker or input RGB value
Font family, font size	Part and State > Main > Font family, font size	Part and State > Main > Font family, font size	Font family and size settings of tab content: the size can be a value between 0 and 100
Letter spacing	Part and State > Main > Letter spacing	Part and State > Main > Letter spacing	The letter space: the space can be a value between 0 and 100

Table 47. Tab view properties...continued

Property	v7	v8	Description
Background color	Part and State > Main > Background color	Part and State > Main > Background color	Background color of main part: it can be set by color picker or input RGB value
Background Opacity	Part and State > Main > Opacity	Part and State > Main > Opacity	Background opacity: the value should be between 0 and 255
Background gradient direction	Part and State > Main > Background gradient direction	Part and State > Main > Background gradient direction	Gradient direction of background color: vertical or horizontal
border color	Part and State > Main > border color	Part and State > Main > border color	Color of border: it can be set by color picker or input RGB value
border width	Part and State > Main > border width	Part and State > Main > border width	Width of border line: value can be one value from 0 to 5
border opacity	Part and State > Main > border opacity	Part and State > Main > border opacity	border opacity: the value should be between 0 and 255
state	Part and State > Indic > state	N/A	State of indic part: it can be defined by one state or more states
Disable	Part and State > Indic > Disable	N/A	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Background color	Part and State > Indic > Background color	N/A	Background color of Indic part: it can be set by color picker or input RGB value
Opacity	Part and State > Indic > Opacity	N/A	Indic opacity: the value should be between 0 and 255
Background gradient direction	Part and State > Indic > Background gradient direction	N/A	Gradient direction of Indic background color: vertical or horizontal
state	Part and State > Tab main > state	Part and State > Tab > state	State of Tab part: it can be defined by one state or more states
Disable	Part and State > Tab main > Disable	Part and State > Tab > Disable	Enable or disable a state: the custom settings are invalidated when a state is disabled
Background color	Part and State > Tab main > Background color	Part and State > Tab > Background color	Background color of Tab part: it can be set by color picker or input RGB value
Opacity	Part and State > Tab main > Opacity	Part and State > Tab > Opacity	Tab Background opacity: the value should be between 0 and 255
Background gradient direction	Part and State > Tab main > Background gradient direction	Part and State > Tab > Background gradient direction	Gradient direction of background color: vertical or horizontal
Border width	Part and State > Tab main > Border width	Part and State > Tab > Border width	Width of Tab border line: value can be one value from 0 to 5
border color	Part and State > Tab main > Border color	Part and State > Tab > Border color	Color of border: it can be set by color picker or input RGB value
Board size	Part and State > Tab main > Board size	Part and State > Tab > Board size	Board width and board radius

Table 47. Tab view properties...continued

Property	v7	v8	Description
state	Part and State > Tab > state	Part and State > items > state	State of Button part: it can be defined by one state or more states
Disable	Part and State > Tab > Disable	Part and State > items > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Text color	Part and State > Tab > Text color	N/A	Text color in Button part: it can be set by color picker or input RGB value
Letter spacing	Part and State > Tab > Letter spacing	N/A	The letter space: the space can be a value between 0 and 100
Background color	N/A	Part and State > items > Background color	Background color of Button part: it can be set by color picker or input RGB value
Background Opacity	N/A	Part and State > items > Opacity	Opacity of Button part: the value should be between 0 and 255
Background gradient direction	N/A	Part and State > items > Background gradient direction	Gradient direction of Button part background color: vertical or horizontal
border color	N/A	Part and State > items > border color	Color of button border: it can be set by color picker or input RGB value
border width	N/A	Part and State > items > border width	Width of button border line: the value can be one value from 0 to 5
border opacity	N/A	Part and State > items > border opacity	Border opacity: the value should be between 0 and 255

4.40 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.

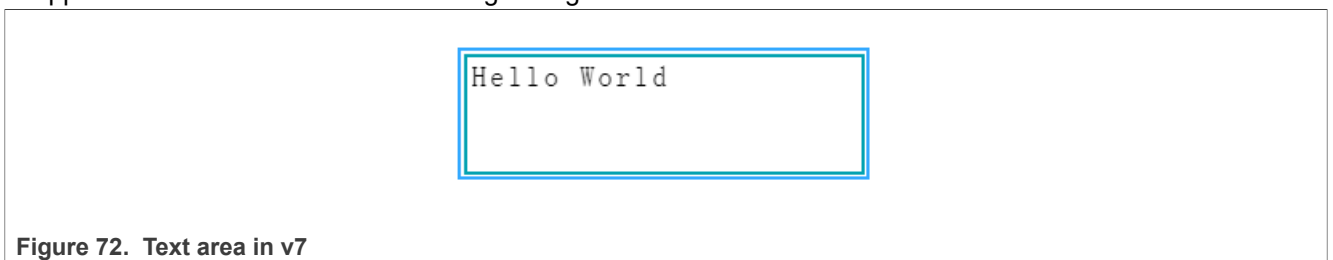


Table 48. Text area properties

Property	v7	v8	Description
Text	Attribute > Text	Attribute > Text	Widget text
Keyboard	Attribute > Keyboard	Attribute > Keyboard	Enable the input keyboard
Chinese input	N/A	Attribute > Chinese input	Enable the Chinese input
state	Part and State > Main > state	Part and State > Main > state	State of main part: it can be defined by one state or more states
Disable	Part and State > Main > Disable	Part and State > Main > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Background color	N/A	Part and State > Main > Background color	Background color of widget: it can be set by color picker or input RGB value
Opacity	N/A	Part and State > Main > Opacity	Background opacity: the value should be between 0 and 255
Background gradient direction	N/A	Part and State > Main > Background gradient direction	Gradient direction of background color: vertical or horizontal
Font	N/A	Part and State > Main > Align	Set the alignment of the text
Font color	Part and State > Main > Font color	Part and State > Main > Font color	Text color in main part: it can be set by color picker or input RGB value
Font family, font size	Part and State > Main > Font family, font size	Part and State > Main > Font family, font size	Font family and size settings in the window content: the size can be a value between 0 and 100
Letter spacing	Part and State > Main > Letter spacing	Part and State > Main > Letter spacing	The letter space: the space can be a value between 0 and 100
Border color	Part and State > Main > Border color	Part and State > Main > Border color	Color of border: it can be set by color picker or input RGB value
Border width	Part and State > Main > Border width	Part and State > Main > Border width	Width of border line: value can be one value from 0 to 5
Radius	Part and State > Main > Radius	Part and State > Main > Radius	Radius of Border: 0-200
state	Part and State > scrollbar > state	Part and State > scrollbar > state	State of Scrollable part: it can be defined by one state or more states
Disable	Part and State > scrollbar > Disable	Part and State > scrollbar > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Background color	Part and State > scrollbar > Background color	Part and State > scrollbar > Background color	Background color of Scrollable part: it can be set by color picker or input RGB value
Opacity	Part and State > scrollbar > Opacity	Part and State > scrollbar > Opacity	Scrollable opacity: the value should be between 0 and 255
Background gradient direction	Part and State > scrollbar > Background gradient direction	Part and State > scrollbar > Background gradient direction	Gradient direction of background color: vertical or horizontal

4.41 Text progress bar

The text progress bar displays the progress by a percent number which can include a custom number of decimals.

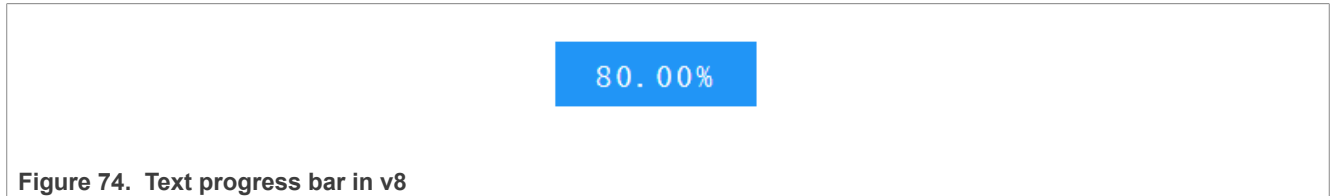


Figure 74. Text progress bar in v8

Table 49. Text progress bar properties

Property	v7	v8	Description
Range	N/A	Attribute > textprogress	
Step	N/A	Attribute > Step	
Initial value	N/A	Attribute > Initial value	
Decimals	N/A	Attribute > Decimals	
State	N/A	Part and State > Main	
Disable	N/A	Part and State > Main > Disable	
Background color	N/A	Part and State > Main > Background color	
Opacity	N/A	Part and State > Main > Opacity	
Background gradient direction	N/A	Part and State > Main > Background gradient direction	
Font color	N/A	Part and State > Main > Font color	
Font size, font family	N/A	Part and State > Main > Font size, font family	
Letter spacing	N/A	Part and State > Main > Letter spacing	
Padding top, bottom, left, right	N/A	Part and State > Main > Padding top,bottom, left, right	

4.42 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.

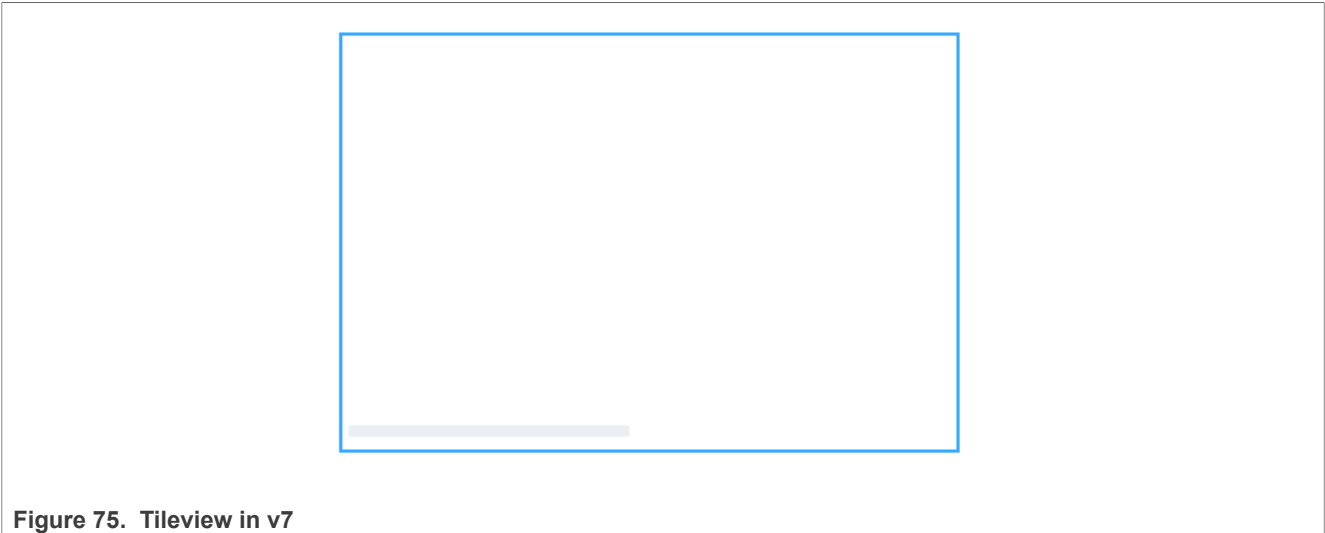


Figure 75. Tileview in v7

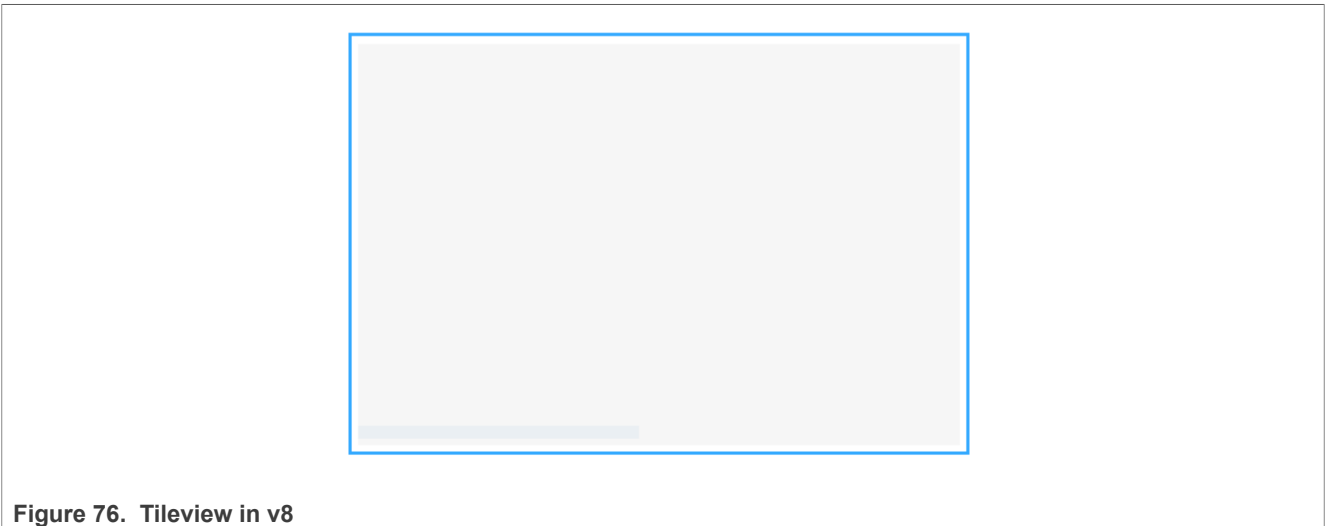


Figure 76. Tileview in v8

Table 50. Tileview properties

Property	v7	v8	Description
Animation	Attribute > Animation	N/A	Animation time of widget when a tile is loaded: the value should be one value from 0 to 3000
Direction	Attribute > Direction	Attribute > Direction	The direction to change the tile: it can be horizon or vertical
Page	Attribute > Page, Name	Attribute > Page, Name	Add new tile and set tile title
state	Part and State > Main > state	Part and State > Main > state	State of main part: it can be defined by one state or more states
Disable	Part and State > Main > Disable	Part and State > Main > Disable	Enable or disable a state: the custom settings are invalidated when a state is disabled
Background color	Part and State > Main > Background color	Part and State > Main > Background color	Background color of widget: it can be set by color picker or input RGB value
Opacity	Part and State > Main > Opacity	Part and State > Main > Opacity	Background opacity: the value should be between 0 and 255

Table 50. Tileview properties...continued

Property	v7	v8	Description
Background gradient direction	Part and State > Main > Background gradient direction	Part and State > Main > Background gradient direction	Gradient direction of background color: vertical or horizontal
Radius	Part and State > Main > Radius	Part and State > Main > Radius	Radius of Border: 0-200
state	Part and State > Flash > state	N/A	State of Flash part: it can be defined by one state or more states
Disable	Part and State > Flash > Disable	N/A	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Background color	Part and State > Flash > Background color	N/A	Background color of Flash part: it can be set by color picker or input RGB value
Opacity	Part and State > Flash > opacity	N/A	Background opacity: the value should be between 0 and 255
Background gradient direction	Part and State > Flash > Background gradient direction	N/A	Gradient direction of background color: vertical or horizontal
Radius	Part and State > Flash > Radius	N/A	Border Radius of Flash part: 0-200
state	Part and State > scrollbar > state	Part and State > scrollbar > state	State of Scrollable part: it can be defined by one state or more states
Disable	Part and State > scrollbar > Disable	Part and State > scrollbar > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Background color	Part and State > scrollbar > Background color	Part and State > scrollbar > Background color	Background color of Scrollable part: it can be set by color picker or input RGB value
Opacity	Part and State > scrollbar > Opacity	Part and State > scrollbar > Opacity	Background opacity of Scrollable part: the value should be between 0 and 255
Background gradient direction	Part and State > scrollbar > Background gradient direction	Part and State > scrollbar > Background gradient direction	Gradient direction of background color: vertical or horizontal

4.43 Video

It can play video files of *.h264 format, which can be converted from other formats by FFmpeg.

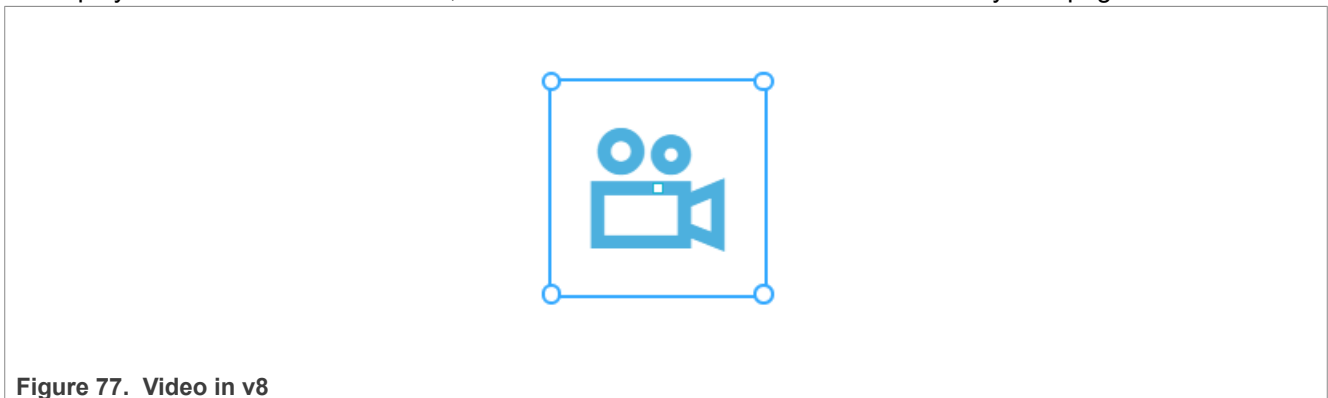


Figure 77. Video in v8

Table 51. Video properties

Property	v7	v8	Description
video file	N/A	Attribute > videoFile	Only supports h264 file

4.44 Window

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

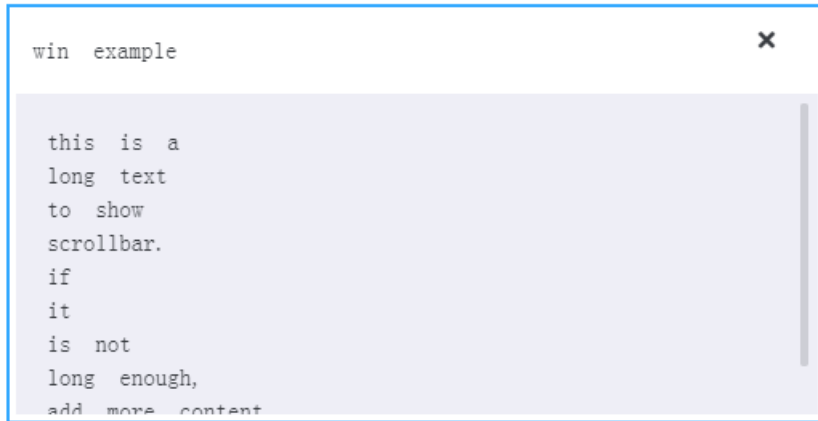


Figure 78. Window in v7

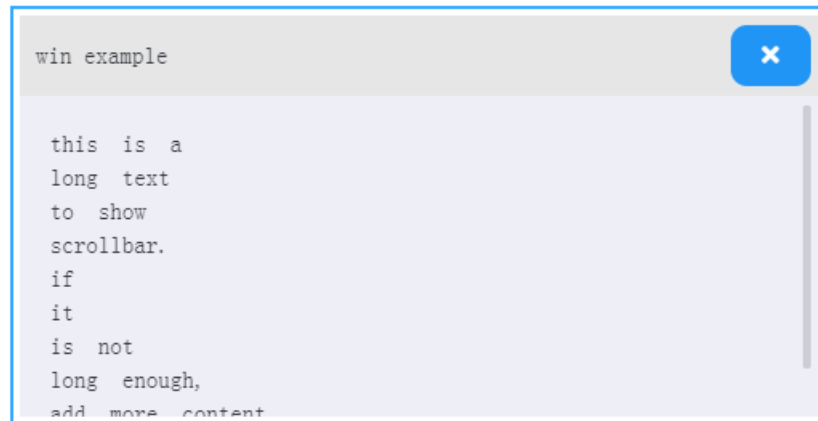


Figure 79. Window in v8

Table 52. Window properties

Property	v7	v8	Description
Head size	N/A	Attribute > Height	Header height
Title	Attribute > Title	Attribute > Title	The title of window
Text	Attribute > Text	Attribute > Text	The detailed content text of window
BTNS	Attribute > Button > image path, image size, symbol	Attribute > Button > image path, image size, button size, symbol	Button size is the button width, and image path and image size is the image setting on the button. Symbol defines the symbol on button.

Table 52. Window properties...continued

Property	v7	v8	Description
state	Part and State > Main > state	Part and State > Main > state	State of main part: it can be defined by one state or more states
Disable	Part and State > Main > Disable	Part and State > Main > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Background color	Part and State > Main > Background color	Part and State > Main > Background color	Background color of widget: it can be set by color picker or input RGB value
Opacity	Part and State > Main > Opacity	Part and State > Main > Opacity	Background opacity: the value should be between 0 and 255
Background gradient direction	Part and State > Main > Background gradient direction	Part and State > Main > Background gradient direction	Gradient direction of background color: vertical or horizontal
Border color	Part and State > Main > Border color	N/A	Color of border: it can be set by color picker or input RGB value
Border width	Part and State > Main > Border width	N/A	Width of border line: value can be one value from 0 to 12
state	Part and State > Content > state	Part and State > Content > state	State of content part: it can be defined by one state or more states
Disable	Part and State > Content > Disable	Part and State > Content > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Text color	Part and State > Content > Text color	Part and State > Content > Text color	Text color in content part: it can be set by color picker or input RGB value
Font size, font family	Part and State > Content > Font size, font family	Part and State > Content > Font size, font family	Font family and size settings in the window content: the size can be a value between 0 and 100
Background color	Part and State > Content > Background color	Part and State > Content > Background color	Background color of content part: it can be set by color picker or input RGB value
Opacity	Part and State > Content > opacity	Part and State > Content > opacity	Background opacity of content part: the value should be between 0 and 255
Background gradient direction	Part and State > Content > Background gradient direction	Part and State > Content > Background gradient direction	Gradient direction of background color in content part: vertical or horizontal
state	Part and State > Header > state	Part and State > Header > state	State of header part: it can be defined by one state or more states
Disable	Part and State > Header > Disable	Part and State > Header > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Text color	Part and State > Header > Text color	Part and State > Header > Text color	Text color in header part: it can be set by color picker or input RGB value
Font size, font family	Part and State > Header > Font size, font family	Part and State > Header > Font size, font family	Font family and size settings in the window content: the size can be a value between 0 and 100
Background color	Part and State > Header > Background color	Part and State > Header > Background color	Background color of header part: it can be set by color picker or input RGB value
Opacity	Part and State > Header > opacity	Part and State > Header > opacity	Background opacity of header part: the value should be between 0 and 255

Table 52. Window properties...continued

Property	v7	v8	Description
Background gradient direction	Part and State > Header > Background gradient direction	Part and State > Header > Background gradient direction	Gradient direction of background color in header part: vertical or horizontal
state	Part and State > Buttons > state	Part and State > Button > state	State of Symbol part: it can be defined by one state or more states
Disable	Part and State > Buttons > Disable	Part and State > Button > Disable	Enable or disable a state. The custom settings are invalidated when a state is disabled.
Opacity	Part and State > Buttons > Opacity	Part and State > Button > Opacity	Background opacity of Symbol part: the value should be between 0 and 255
Background color	Part and State > Buttons > Background color	Part and State > Button > Background color	Background color of Symbol part: it can be set by color picker or input RGB value
Background gradient direction	Part and State > Buttons > Background gradient direction	Part and State > Button > Background gradient direction	Gradient direction of background color in Symbol part: vertical or horizontal

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
 - i. Loaded (only available for screen)
 - j. Load Start (only available for screen)
 - k. Unloaded (only available for screen)
 - l. Unloaded Start (only available for screen)
 2. Action options
 - Background Color
 - Background Gradient Color
 - Background Opacity
 - Width
 - Height
 - Position

- Visibility
- C code
- Python code
- Move Animation
- Set Text
- Load Screen

Table 53. Widgets: Available triggers and actions

Widget Name	Available Trigger	Available Action
3Dimage	none	none
Animation Image	none	none
arc	none	none
btnm (button group)	a,b,c,d,e,f,g	all
button	a,b,c,d,e,f,g	all
calendar	none	none
canvas	none	none
chart	none	none
checkbox	h	all
color	none	none
container	none	none
dropdown	h	all
gauge	none	none
image	a,b,c,d,e,f,g	all
imgbtn (image button)	b, c	all
label	none	none
led	none	none
line	none	none
linemeter	none	none
list	none	none
msgbox (message box)	h	all
roller	h	all
screen	a,b,e, i, j, k, l	all
slider	h	all
spinbox	b,h	all
spinner	a,d	all
switch	h	all
tab	h	all
table	none	none
textarea	none	none
tileview	none	none

Table 53. Widgets: Available triggers and actions...continued

Widget Name	Available Trigger	Available Action
window	none	none

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 54. LVGL hardware acceleration

Accelerator	i.MX RT1050	i.MX RT1062	i.MX RT1064	i.MX RT1170	i.MX RT1160	i.MX RT595
PXP	√	√	√	√	√	X
VGLite	X	X	X	√	√	√

6.1 PXP enablement

Enable the PXP accelerator in GUI Guider.

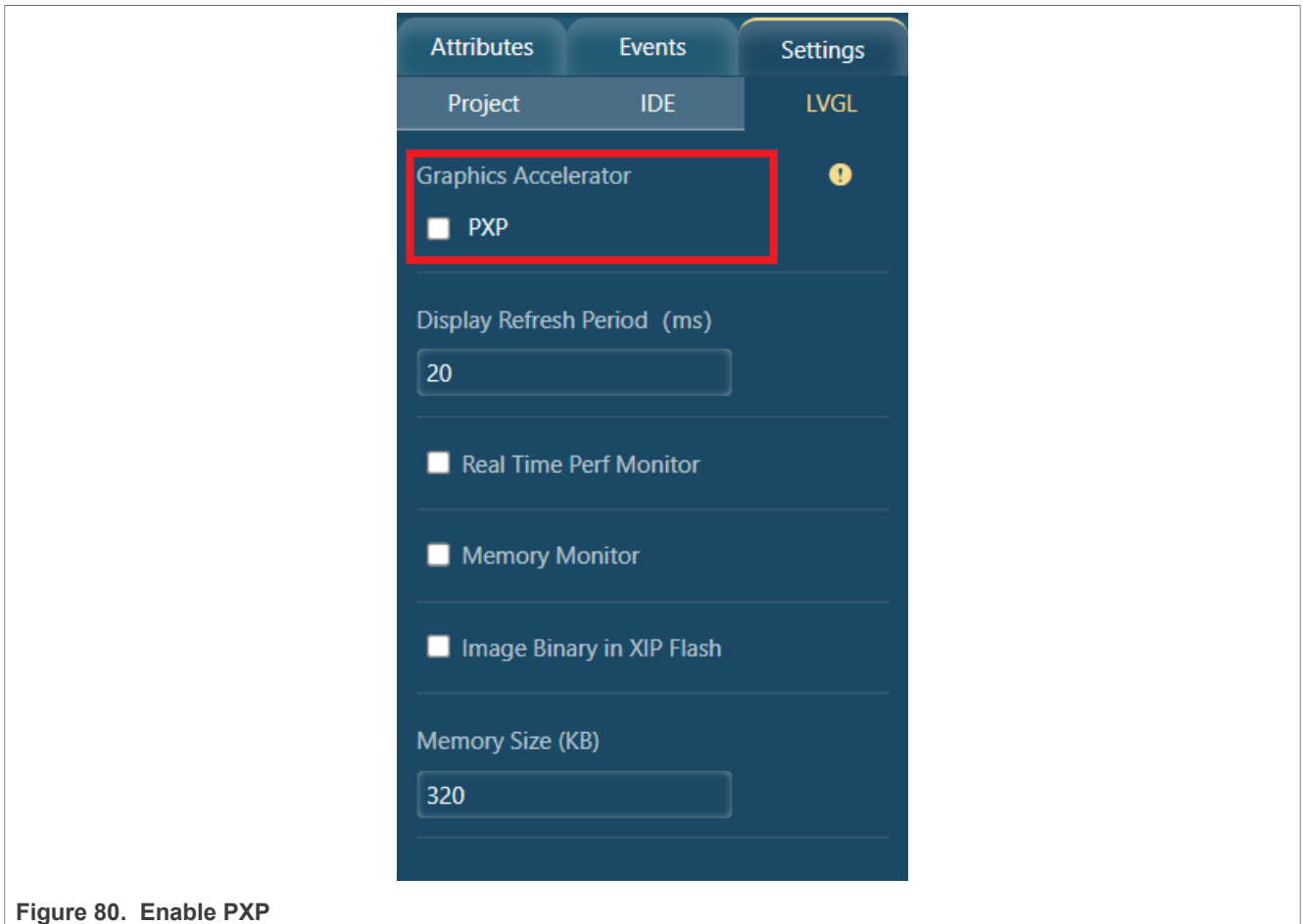


Figure 80. Enable PXP

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.

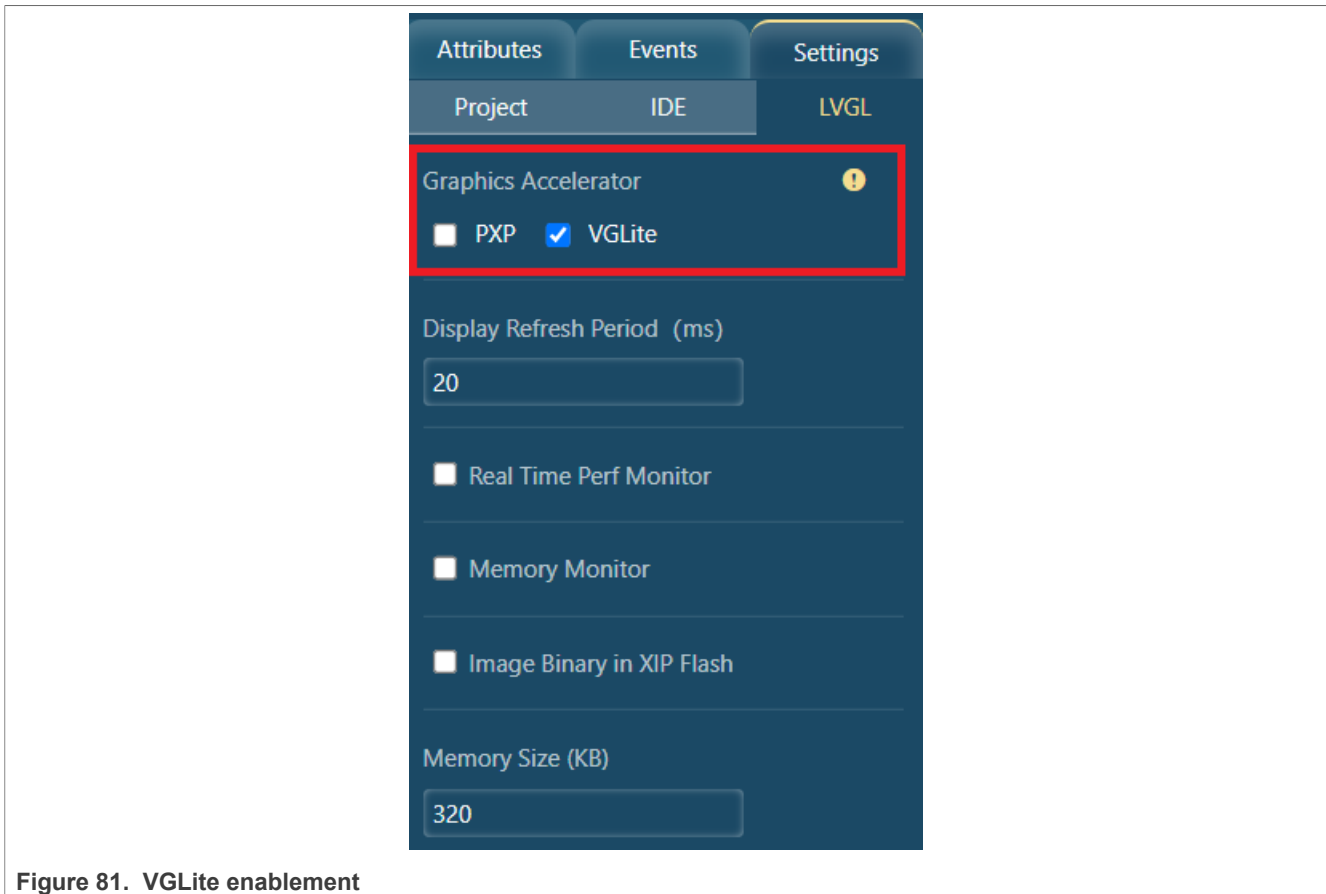


Figure 81. VGLite enablement

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 (BBlock 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 in `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`. 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-transparent images, PXP may be better.

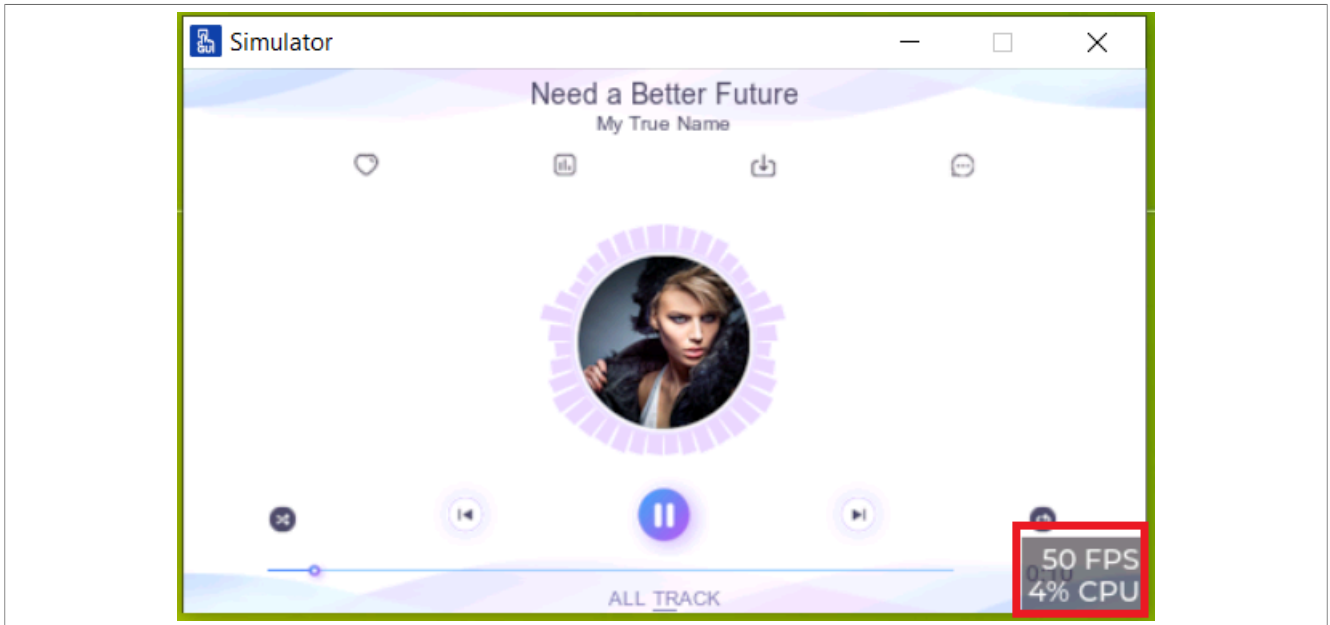
On platforms with only 2D GPU acceleration (VGLite), try to draw widgets rather than using pre-rendered images as widget, as semi-transparent image blitting is not yet accelerated.

7 Performance

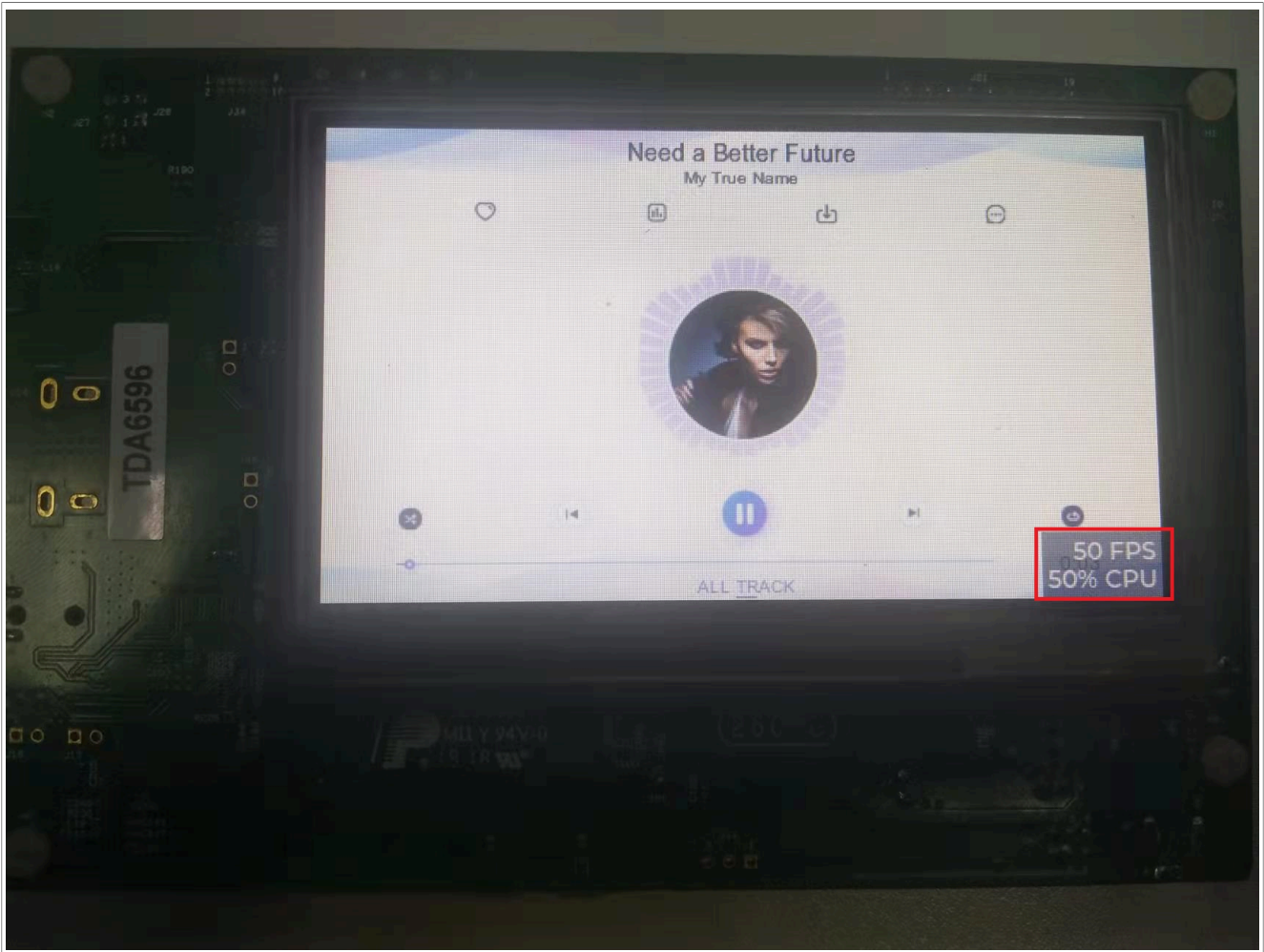
The high graphics performance means a high frame rate (FPS) with required graphical effects. This chapter provides the introduction to enable/disable FPS/CPU usage monitor and the tips on how to improve the graphics performances on NXP MCU devices. i.MX RT595 is used as an example platform for performance optimization.

7.1 Performance monitor enablement

1. Enable the performance monitor in GUI Guider.
2. Check the real-time performance results in simulator.



3. Check the real-time performance results on boards.



7.2 Tips to improve the performance

Here is a summary of tips to get a good FPS performance using lvgl:

- **Use hardware acceleration**
The capability of a board with hardware acceleration (PXP or VGLite) is often higher than a board without. Consider using a board with hardware acceleration. For details, see [Section 6](#).
- **Use Internal SRAM**
The SRAM has better performance than other RAM. If a board has enough SRAM, the SRAM is a preferred place to store the frame buffers and other important data.
- **Use suitable C library**
The Newlib library has good memcpy performance than the NewlibNano library. The Newlib library is preferred for applications with lots of data copy.
- **Use suitable compiler optimization level**
In general, the -O2 and -O3 have better performance than other optimization level. GUI Guider can update the optimization level used in the demo example project, shown in [Figure 82](#).

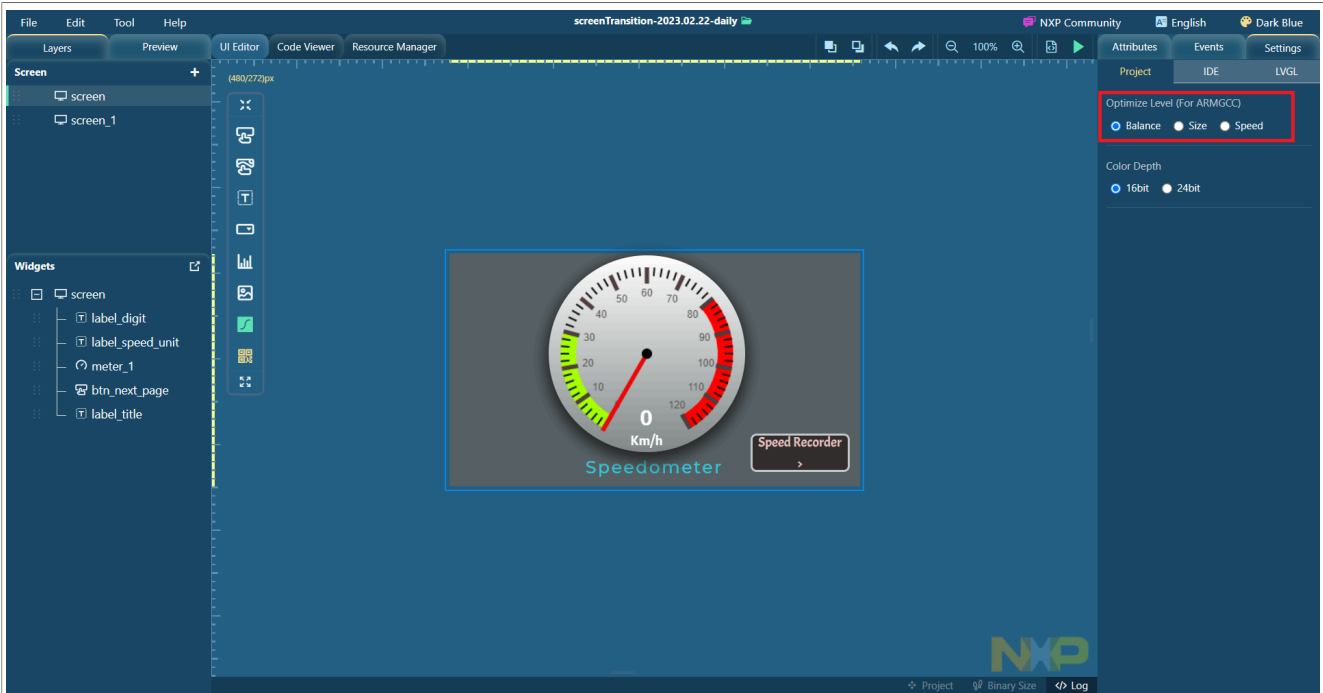


Figure 82. Optimization level

In [Figure 82](#), **Balance** means the -O2 option, **Size** means the -Os option, **Speed** means the -O3 option.

- **Only redraw the changed things**

Make sure that you only invalidate necessary parts of the display.

- **Adjust display refresh period**

The display refresh rate is a hard limit for your frame rate. In general, the frame rate is better when the display refresh period is lower. If the refresh rate of the display is 60 Hz, the refresh period is $1 \text{ s} / 60 = 0.01667 \text{ s} = 16.67 \text{ ms}$. GUI Guider supports updating the refresh period, shown in [Figure 83](#).

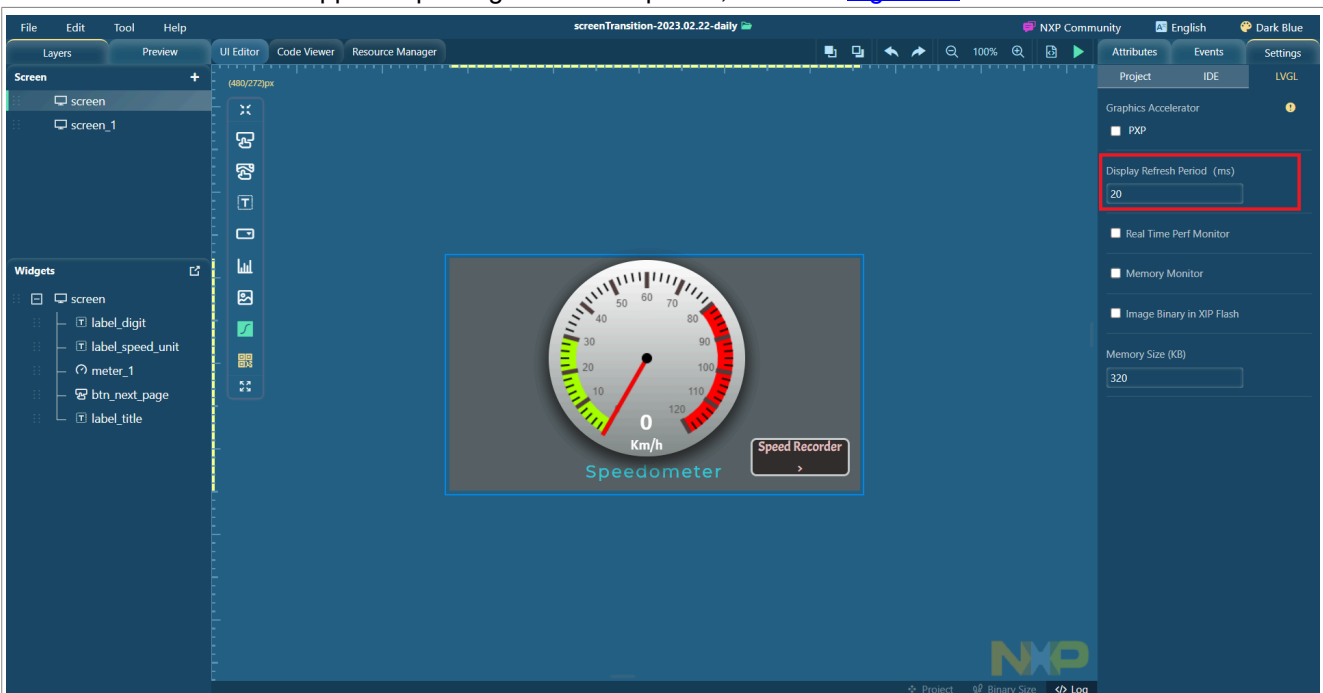


Figure 83. Display refresh rate

7.3 Improve the performance for i.MX RT boards

This section provides information on how to improve the performance on i.MX RT595 when working with MCUXpresso IDE.

7.3.1 Prerequisites

Design a GUI application using the GUI Guider and port the generated LVGL C source file to the template project imported by MCUXpresso IDE.

7.3.2 Improve the performance

1. To use the Release build configuration, -O2 optimization level, and Newlib library, update the MCUXpresso setting. For details, see the MCUXpresso IDE documentation.
2. To change the display refresh period, update the following line in source/lv_conf.h.

```
#define LV_DISP_DEF_REFR_PERIOD    30    /*[ms]*/
```

For example, if the refresh rate of the display is 60 Hz, the value can be set to 16.67.

3. Enable the hardware VGLite acceleration by changing the following line in source/lv_conf.h.

```
#define LV_USE_GPU_NXP_VG_LITE    0    // change to 1 to enable VGLite.
```

4. If the NXP "G1120B0MIPI" MIPI Circular Display is selected, the frame buffer can be placed in SRAM. You can update the following lines in board/display_support.h:

```
#define DEMO_BUFFER0_ADDR 0x28000000U // i.e. Change to 0x20000000U
#define DEMO_BUFFER1_ADDR 0x28200000U //i.e. Change to 0x20100000U
```

5. If the NXP "G1120B0MIPI" MIPI Circular Display is selected and few images are used. The image arrays can also be placed in the SRAM. To place the image array in SRAM, you can add the following macro definition in source/lv_conf.h first.

```
#define LV_ATTRIBUTE_IMG_CONST
__attribute__((section("DataQuickAccess")))
Then updated the C array definition in the Image C source files to add above
macro:
const LV_ATTRIBUTE_IMG_CONST LV_ATTRIBUTE_MEM_ALIGN
LV_ATTRIBUTE_LARGE_CONST LV_ATTRIBUTE_IMG_XXXXXXX uint8_t xxxxxxxxxxxxxx_map[]
= { ... },
LV_ATTRIBUTE_IMG_CONST const lv_img_dsc_t xxxxxxxxxxxxxxxxxxxxxxxxxx = { ... },
```

8 Debug GUI Guider project

MCUXpresso IDE, Keil MDK, and IAR are integrated in GUI Guider which makes it easy to debug GUI application on NXP MCU devices.

8.1 MCUXpresso

To debug the GUI Guider project on MCUXpresso, perform the following steps:

1. Open the link <https://mcuxpresso.nxp.com/en/select>.
2. Select the development board. For example, EVK-MIMXRT1064.
3. Click the **Build MCUXpresso SDK** button.
4. Select the two middleware LVGL and FreeRTOS from the **Build SDK for <target>** page.
5. Make sure to select the MCUXpresso (toolchain).

6. Click the **Download SDK** button.

Build SDK for EVK-MIMXRT1064
 Generate a downloadable SDK archive for use with desktop MCUXpresso Tools.
 Developer Environment Settings
 Selections here will impact files and examples projects included in the SDK and Generated Projects

Host OS: [Windows] [Mac] [Linux]

Toolchain / IDE: [ECL] [GCC] [ARM KEIL] [AT]

SDK Version: 2.11.1 (released 2022-03-24)
 SDK Tag: REL_2_11_1_MINOR_RFP

SELECT ALL | UNSELECT ALL

Name	Category	Description	Dependencies
<input type="checkbox"/> FreeMASTER	Middleware	FreeMASTER communication driver for 32bit platforms	
<input type="checkbox"/> JPEG library	Middleware	JPEG library	
<input type="checkbox"/> LittleFS	Middleware	LittleFS filesystem stack	
<input checked="" type="checkbox"/> LVGL	Middleware	LVGL Open Source Graphics Library	
<input type="checkbox"/> lwIP	Middleware	Lightweight IP open-source TCP/IP stack	
<input type="checkbox"/> Maestro Audio Framework for MCU	Middleware	Maestro Audio Framework library for MCU	
<input type="checkbox"/> mbedtls	Middleware	mbedtls SSL/TLS library	
<input type="checkbox"/> Azure RTOS		Azure RTOS	
<input checked="" type="checkbox"/> FreeRTOS		Real-time operating system for microcontrollers from Amazon	
<input type="checkbox"/> AWS IoT Core	Middleware	Amazon Web Service (AWS) IoT Core SDK	FreeRTOS, lwIP, mbedtls, NXP Wi-Fi
<input type="checkbox"/> Crank Storyboard GUI	Middleware	Crank Storyboard GUI Engine from Crank Software	FreeRTOS
<input type="checkbox"/> NXP Wi-Fi	Middleware	NXP Wi-Fi	FreeRTOS, lwIP, SDMMC Stack

DOWNLOAD SDK

7. Import the downloaded SDK into the IDE.

8. Click **File > import > General**.

workspace - evkbimxrt1050_lvgl_guider/board/lvgl_support.h - MCUXpresso IDE

File Edit Source Refactor Navigate Search Project ConfigTools Run RTOS Analysis Window Help

Project Explorer: evkbimxrt1050_lvgl_guider

Import Dialog:

Select

Create new projects from an archive file or directory.

Select an import wizard:

Type filter text

- General
 - Archive File
 - Existing Projects into Workspace**
 - File System
 - Import projects(s) from XML description
 - Preferences
 - Projects from Folder or Archive
- C/C++
- CVS

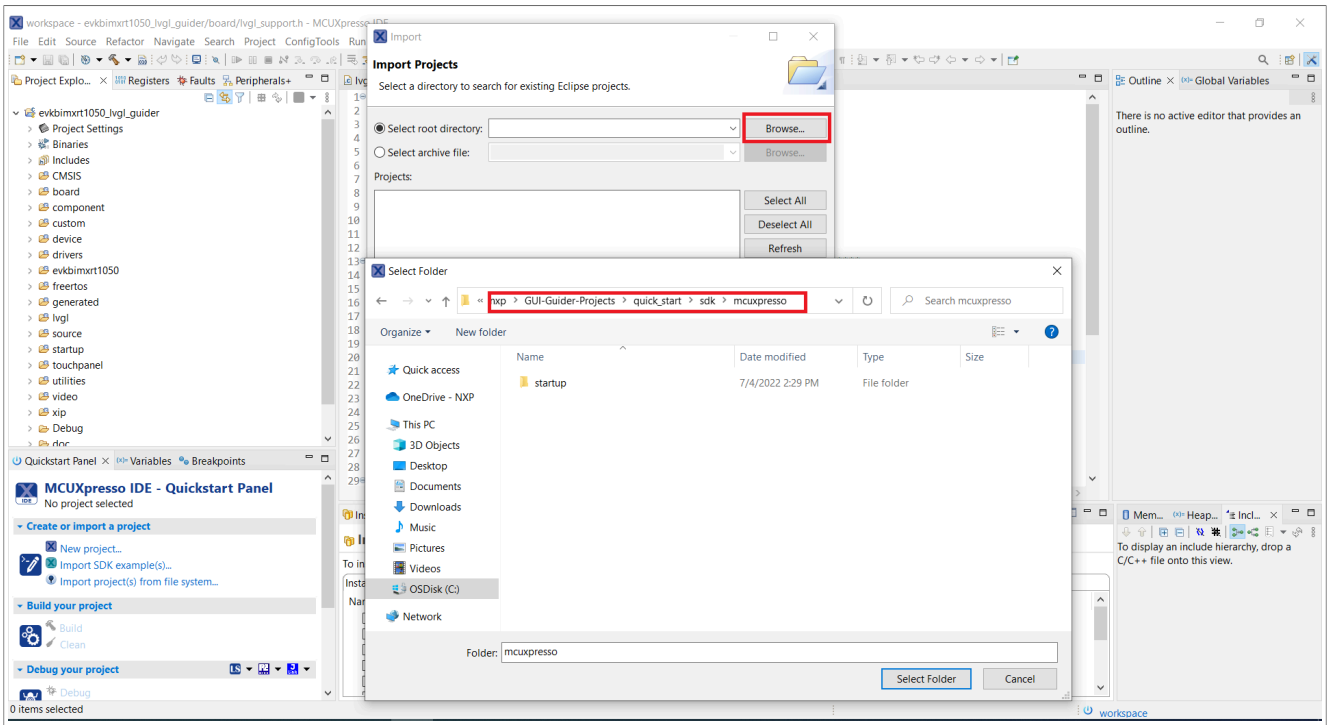
< Back | **Next >** | Finish | Cancel

Installed SDKs

To install an SDK, simply drag and drop an SDK (zip file/folder) or an SDK Git repository into the 'Installed SDKs' view. [Common 'mcuxpresso' folder]

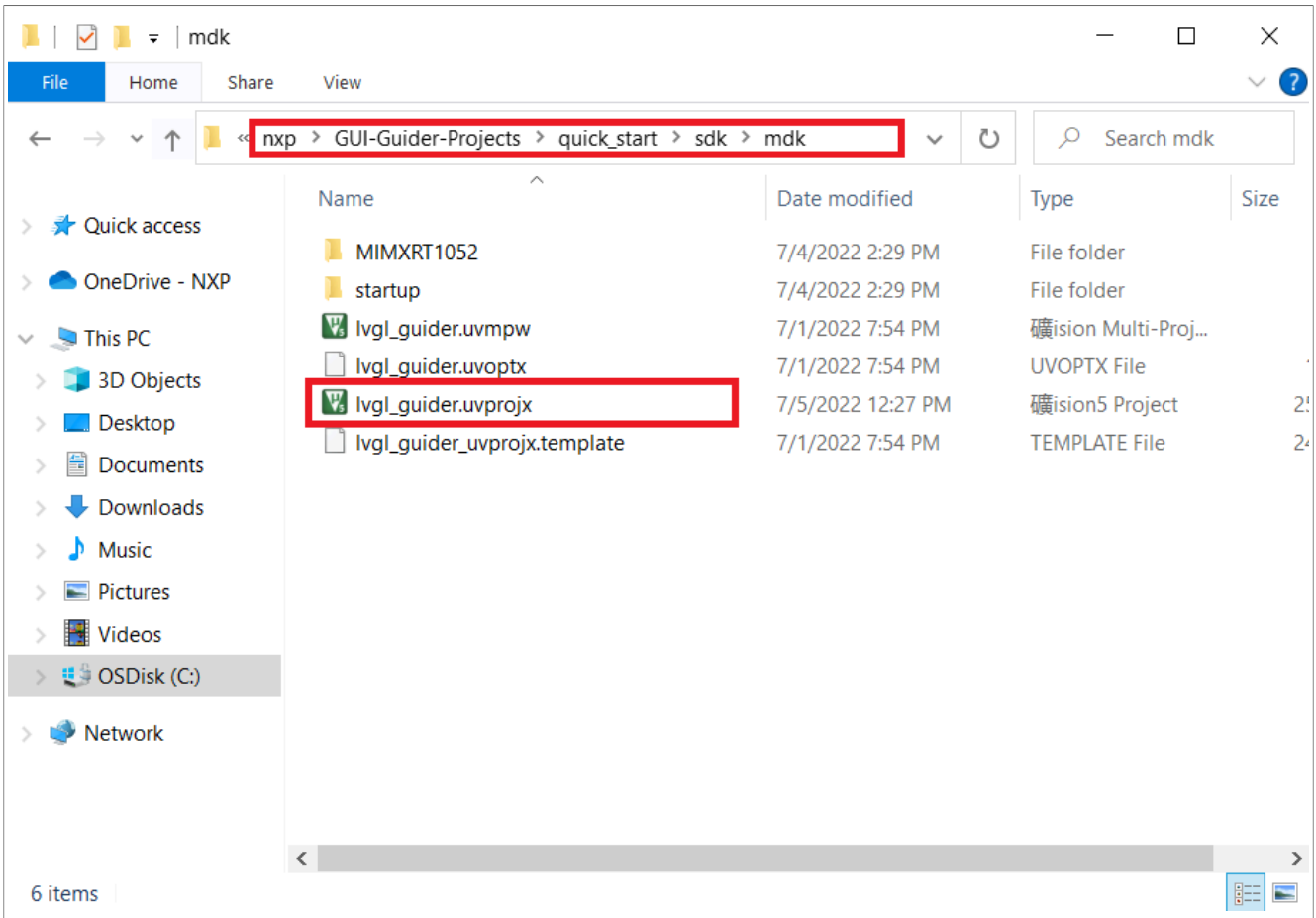
Name	SDK Version	Manifest Version	Location
<input type="checkbox"/> SDK_2_x_EVK-MIMXRT1010	2.9.1 (440 2021-01-20)	3.8.0	C:\Common\SDK_2.9.1_EVK-MIMXRT1010.zip
<input type="checkbox"/> SDK_2_x_EVK-MIMXRT1015	2.9.1 (440 2021-01-20)	3.8.0	C:\Common\SDK_2.9.1_EVK-MIMXRT1015.zip
<input type="checkbox"/> SDK_2_x_EVK-MIMXRT1020	2.9.1 (440 2021-01-20)	3.8.0	C:\Common\SDK_2.9.1_EVK-MIMXRT1020.zip
<input type="checkbox"/> SDK_2_x_EVK-MIMXRT595	2.9.0 (429 2021-01-04)	3.8.0	C:\Common\SDK_2.9.0_EVK-MIMXRT595.zip
<input type="checkbox"/> SDK_2_x_LPCXpresso54628	2.9.0 (435 2021-01-15)	3.8.0	C:\Common\SDK_2.9.0_LPCXpresso54628.zip

9. Select the GUI Guider project MCUXpresso path.



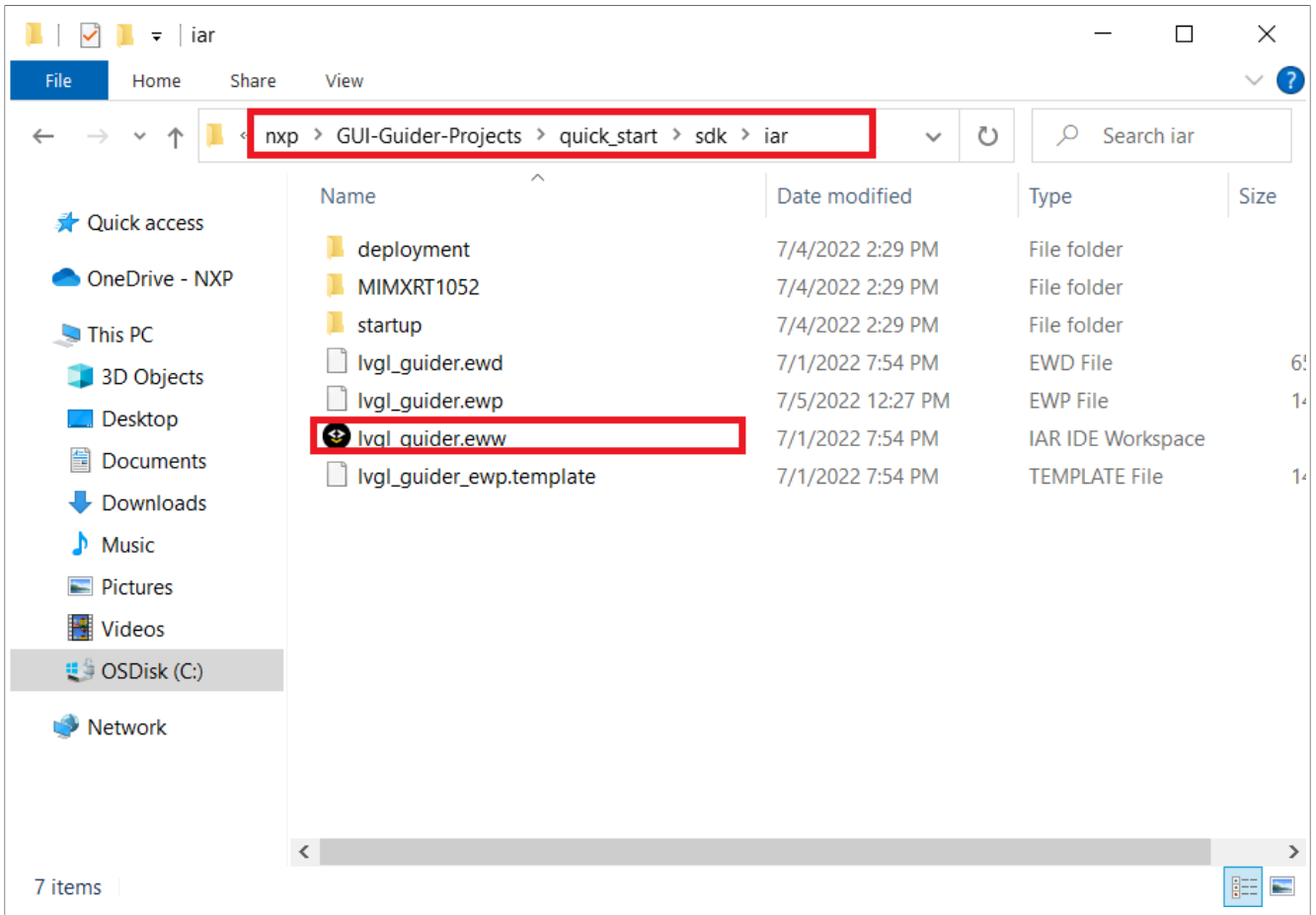
8.2 Keil MDK

Find the path named "mdk", double click **lvgl_guider.uvprojx**.



8.3 IAR

Find the path named "iar", double click **lvgl_guider.eww**.



9 MicroPython

[MicroPython](#) is a lean and efficient implementation of the [Python 3](#) programming language. MicroPython includes a small subset of the Python standard library and is optimized to run on microcontrollers and in constrained environments.

9.1 Using LVGL in MicroPython

By building LVGL as a MicroPython module, user can have a high-level GUI library for fast prototyping GUI, taking advantage of Python's language features. These features include *Inheritance*, *Closures*, *List Comprehension*, *Generators*, *Exception Handling*, *Arbitrary Precision Integers*, and others.

9.1.1 Advantages

- Develop GUI in Python, a very popular high-level language.
- Use paradigms such as Object-Oriented Programming.
- Usually, GUI development requires multiple iterations to get things right. With C, each iteration consists of **Change code > Build > Flash > Run**. In MicroPython, it is **Change code > Run**.

9.2 MicroPython in GUI Guider

GUI Guider ships prebuilt MicroPython binaries by default. For more information on how to build, see [lv_MicroPython README](#).

9.2.1 Generate code

When the **Generate code** button on GUI Guider UI is clicked, the code for both C and MicroPython is generated under the folder `<GUI-Guider-Project-name>/generated`. The MicroPython file `gui_guider.py` is available in the UI.

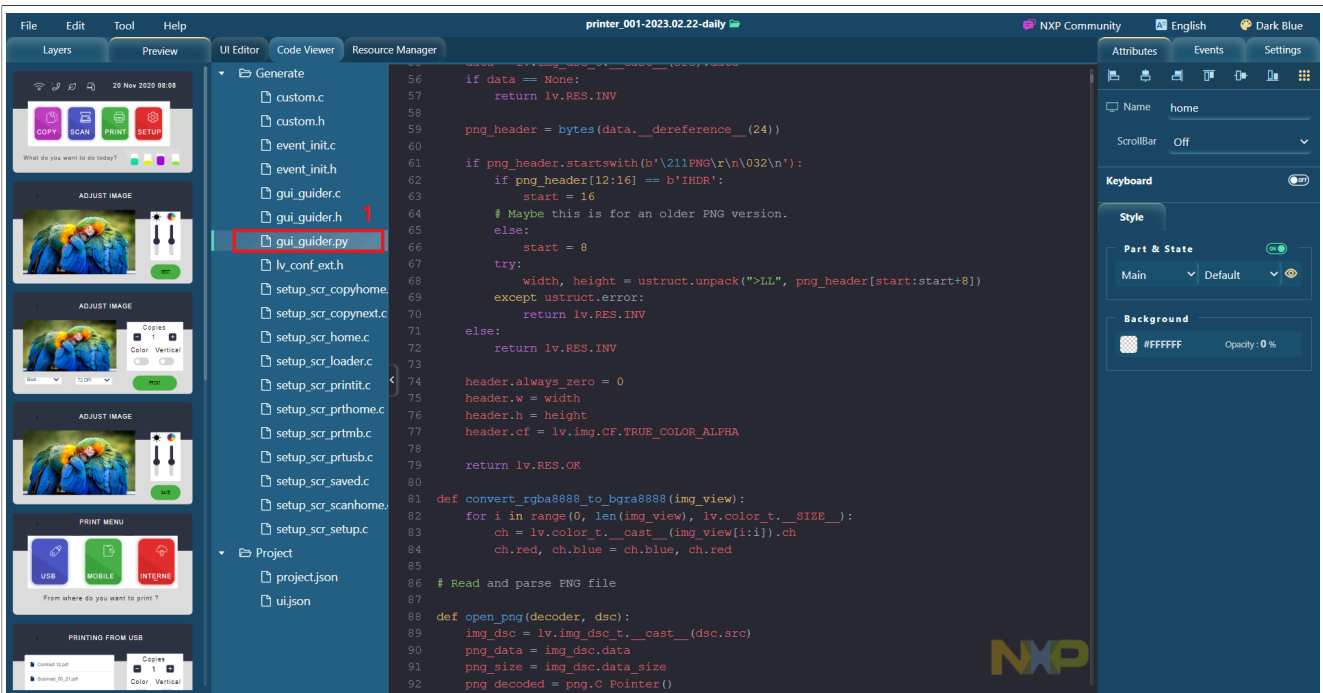


Figure 84. View MicroPython code

9.2.2 Run simulator

Click the **Run simulator > MicroPython** button. The GUI Guider generates code and launches the simulator in a separate window.

9.2.3 Add custom code

Like C, GUI Guider supports adding custom Python code, either as event action, or as independent "custom.py" file under the folder "custom".

Note: *Indentation is a very important concept of Python because without proper indenting the code, `IndentationError` appears and the code is not compiled. To avoid this, GUI Guider follows the below assumptions during the code generation:*

- Each line of a block is indented with four spaces.
- Tab is replaced with four spaces automatically.

9.2.3.1 As event action

[Table 55](#) provides a description of the custom Python code options.

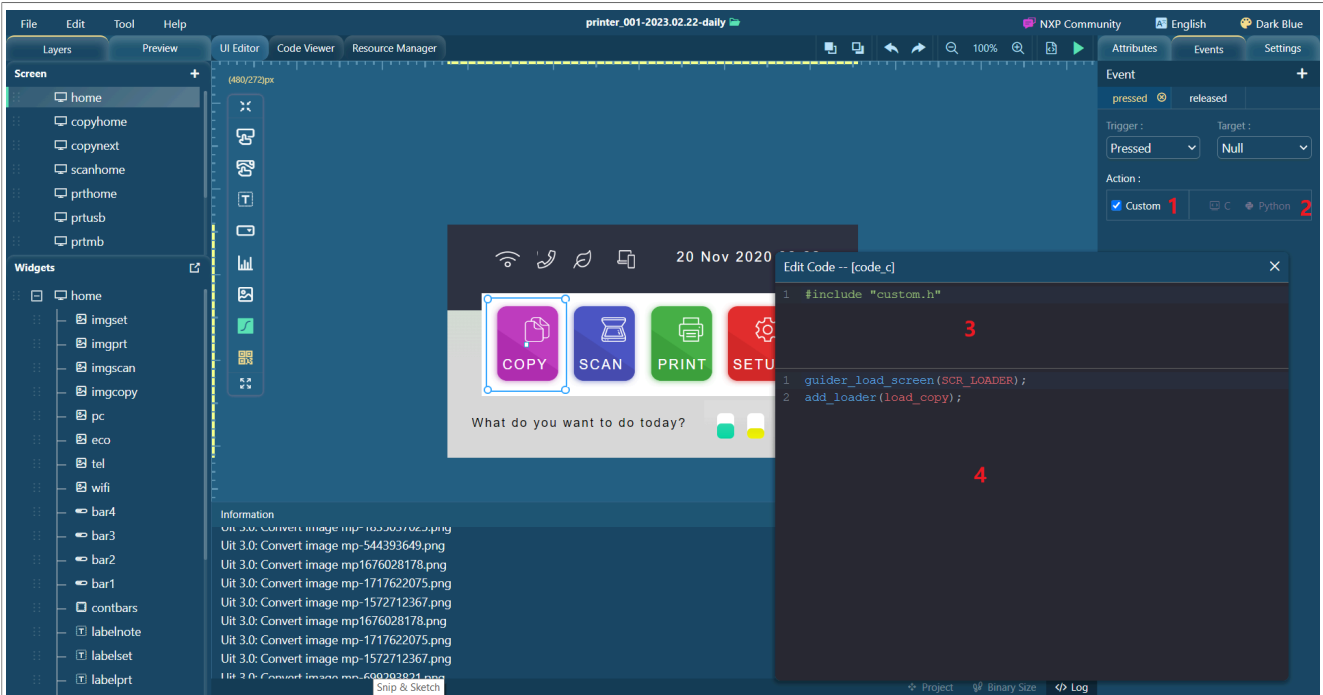


Figure 85. Python code options

Table 55. Custom Python code options

Label	Description
1	Event Type: Customer code
2	Event Code Type: Python code
3	Global variable or function
4	Codes that are wrapped in event callback

9.2.3.2 As custom.py

Put the custom.py file into the folder <GUI-Guider-Project-name>/custom/. The content appears merged into the final gui_guider.py file, replacing the tab with four spaces.

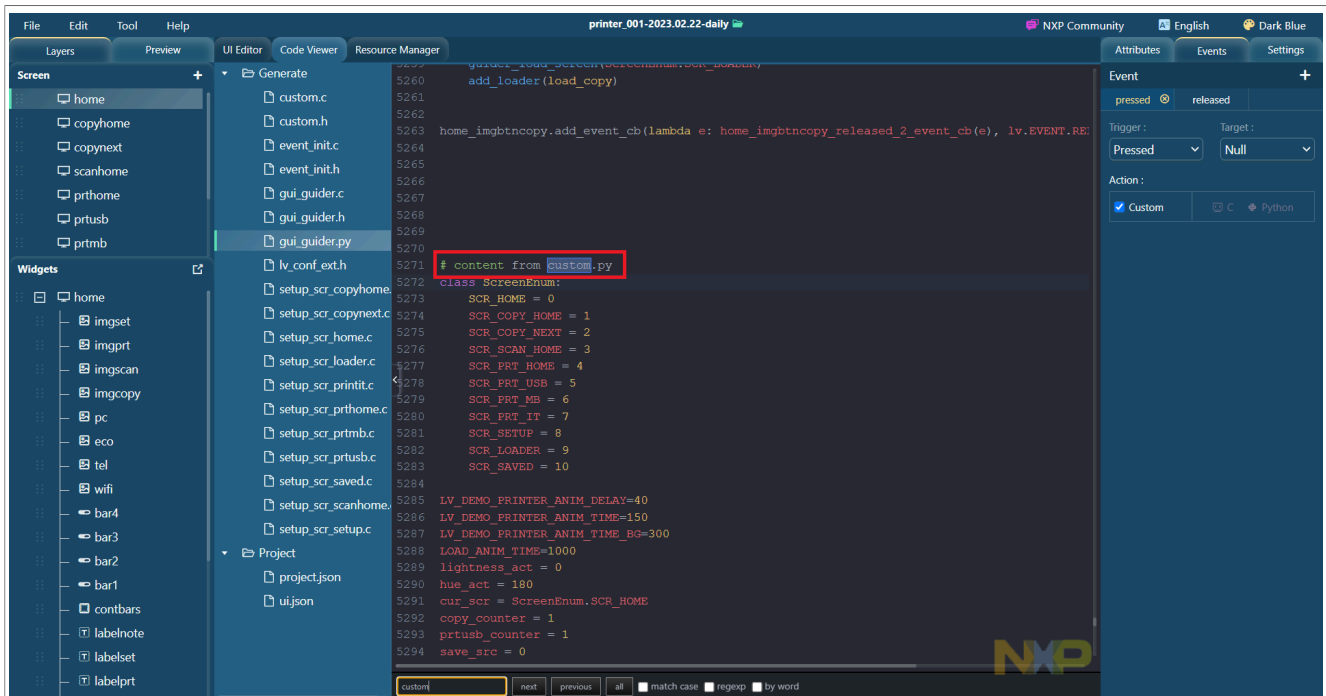


Figure 86. Content appears merged into the final gui_guider.py file

9.2.4 Limitations

- Only LVGL v8 is supported.
- Only limited fonts are built in. See [Built-in fonts](#) in LVGL.
- Compared to C, MicroPython runs slower. Due to this, some animations are not added in Music player demo.

10 Porting RTOS

This section lists the steps to port GUI APP to RTOS (Real-Time Operating System).

10.1 RT-Thread

To port the LVGL C source file generated by GUI Guider to the RT-Thread project, see the following sections.

10.1.1 Prerequisite

- Keil v5.35 or newer.
- Latest GUI Guider GA.
- Connect i.MX RT1060 to the host with a USB cable.

Note: In the working environment, all paths are not allowed to have Chinese characters or spaces.

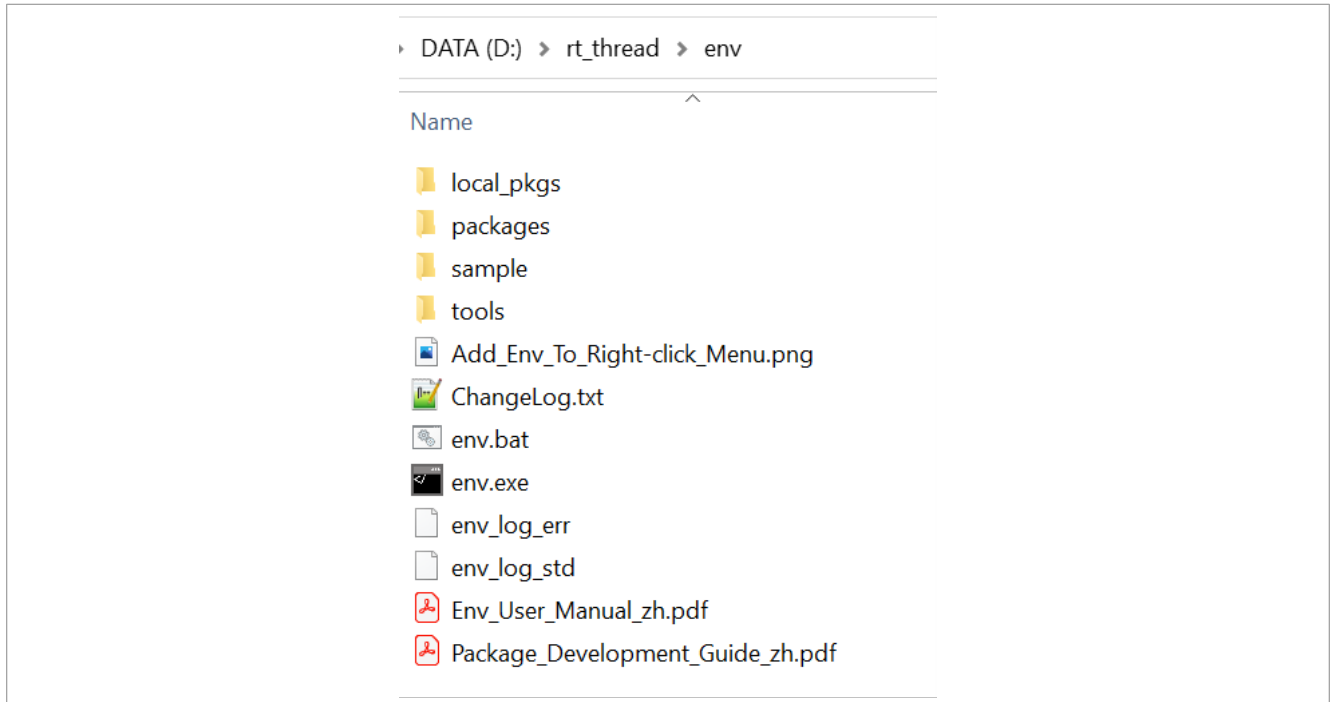
10.1.2 Install Git

Git supports the software package management. Download Git from <https://git-scm.com/downloads>. Install and add the install path into the system environment variable PATH.

10.1.3 Configure the Env tool

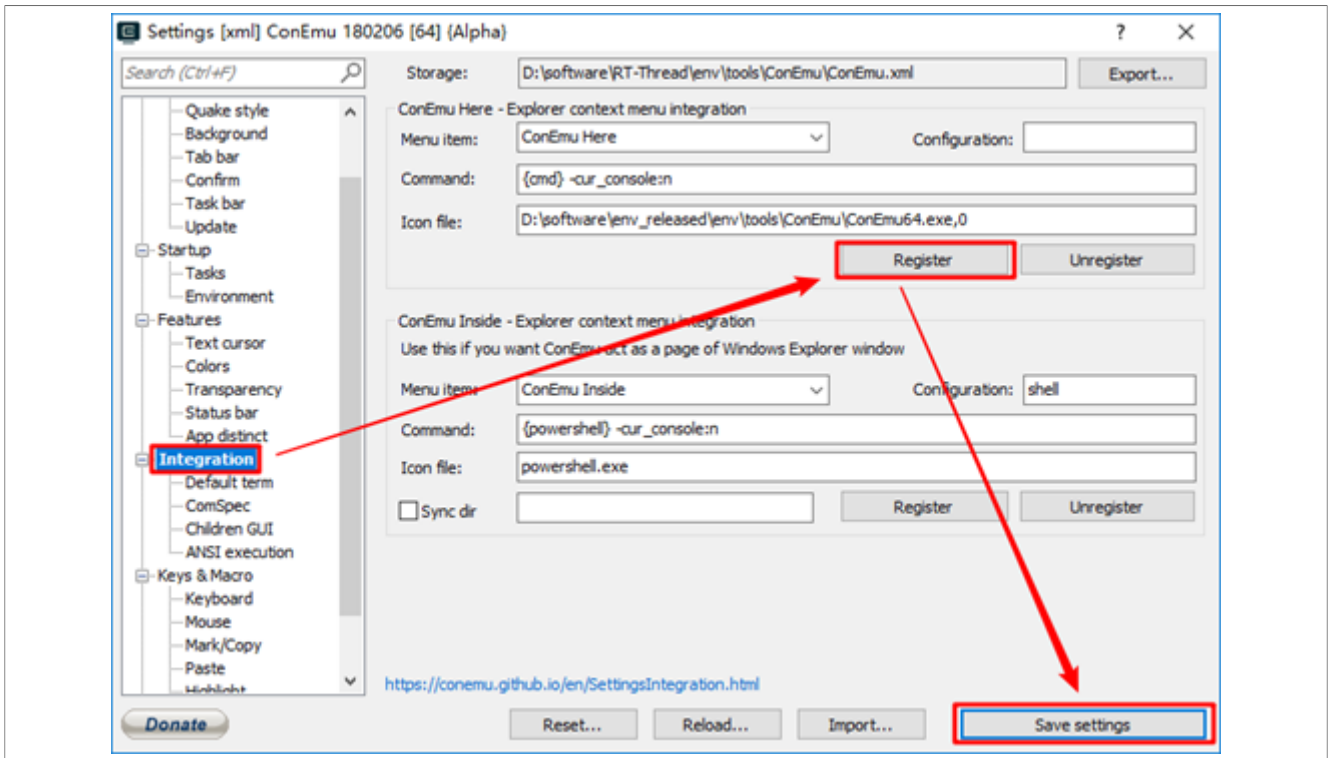
To configure the Env tool, perform the following steps:

1. Download the Env tool: [env_released_1.2.0.7z](#).
2. Extract the file env_released_1.2.0.7z to a local folder. For example, D:\rt-thread\.



3. In the env directory (D:\rt-thread\env), run env.exe. If it fails to open, you can try to use "env.bat".
4. Register env utility in the right-click menu.

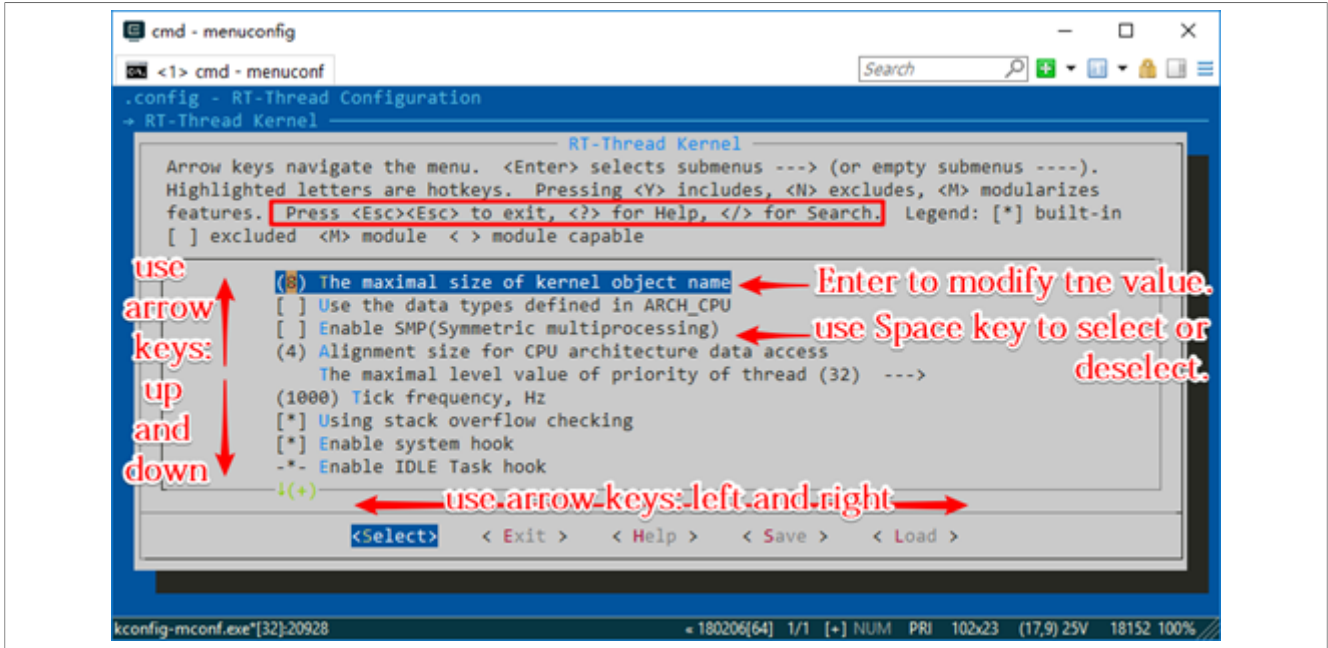




10.1.4 Download RT-Thread and apply patches

To download RT-Thread, perform the following steps:

1. Go to the root folder of RT-thread. For example, D:\rt-thread\.
2. Run Git clone <https://github.com/RT-Thread/rt-thread.git> to download RT-thread source code. Use the committed ID: "e7f64fe07fab03750faa1305b0be4a49b21278cf".
3. Go to the i.mxrt1060 bsp folder (D:\rt-thread\rt-thread\bsp\imxrt\imxrt1060-nxp-evk). Right-click the window and select **ConEmu Here** to open env console.



2. Enable LVGL GUI Guider support.
Location:
 - Hardware Drivers Config
 - Onboard Peripheral Drivers
 - Enable LVGL for LCD
 - Support NXP GUI Guider
- To download the selected packages, run `pkgs --update`.

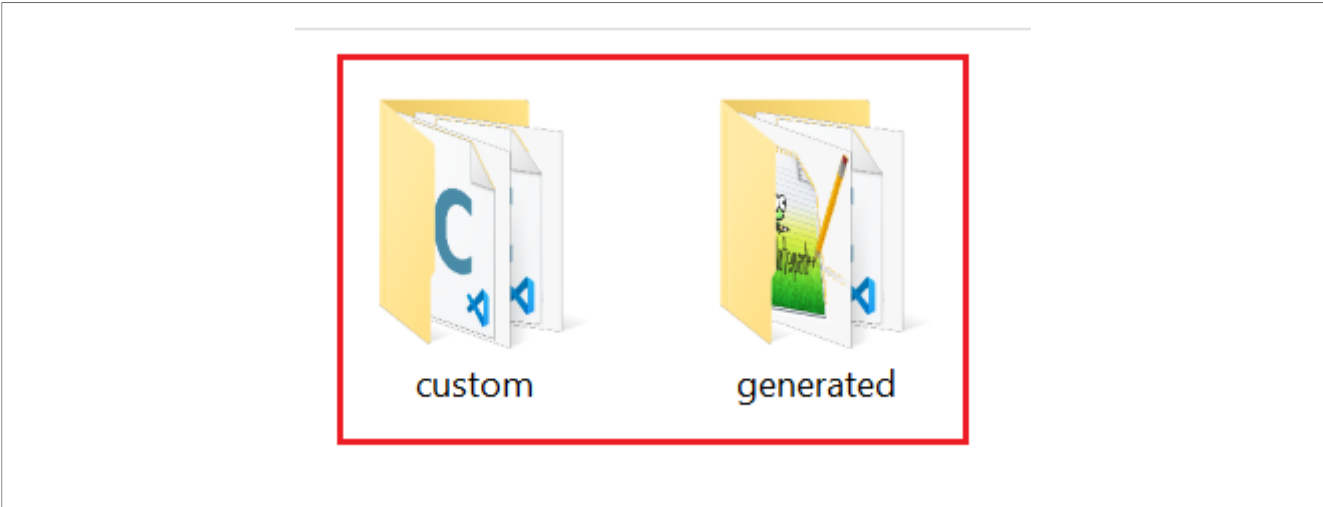
10.1.6 Export source of GUI designed by GUI Guider

To export the source, perform the following steps:

1. Use GUI Guider to design a GUI application.
2. Click **Generate Code** in the GUI Guider IDE.



3. Click **File > Export Code > RT-Thread** on menu bar to export source code of GUI designed by GUI Guider to a template project folder (`D:\rt-thread\rt-thread\bsp\imxrt\imxrt1060-nxp-evk\packages\gui_guider_demo-latest\`).



4. Run `scons --target=mdk5 -s` to generate/update Keil project file "project.uvprojx" which is located at `D:\rt-thread\rt-thread\bsp\imxrt\imxrt1060-nxp-evk`.

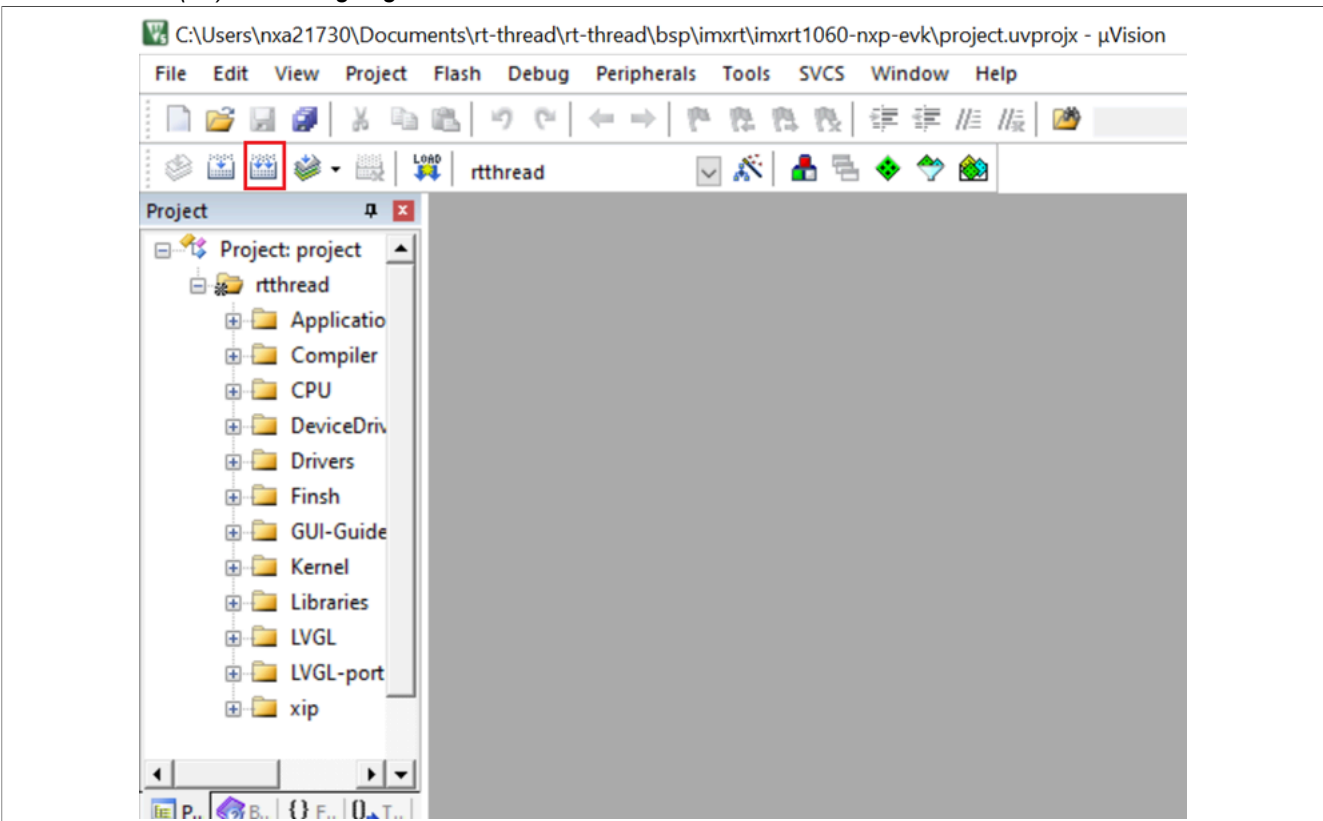
10.1.7 Build and compile

To build and compile, perform the following steps:

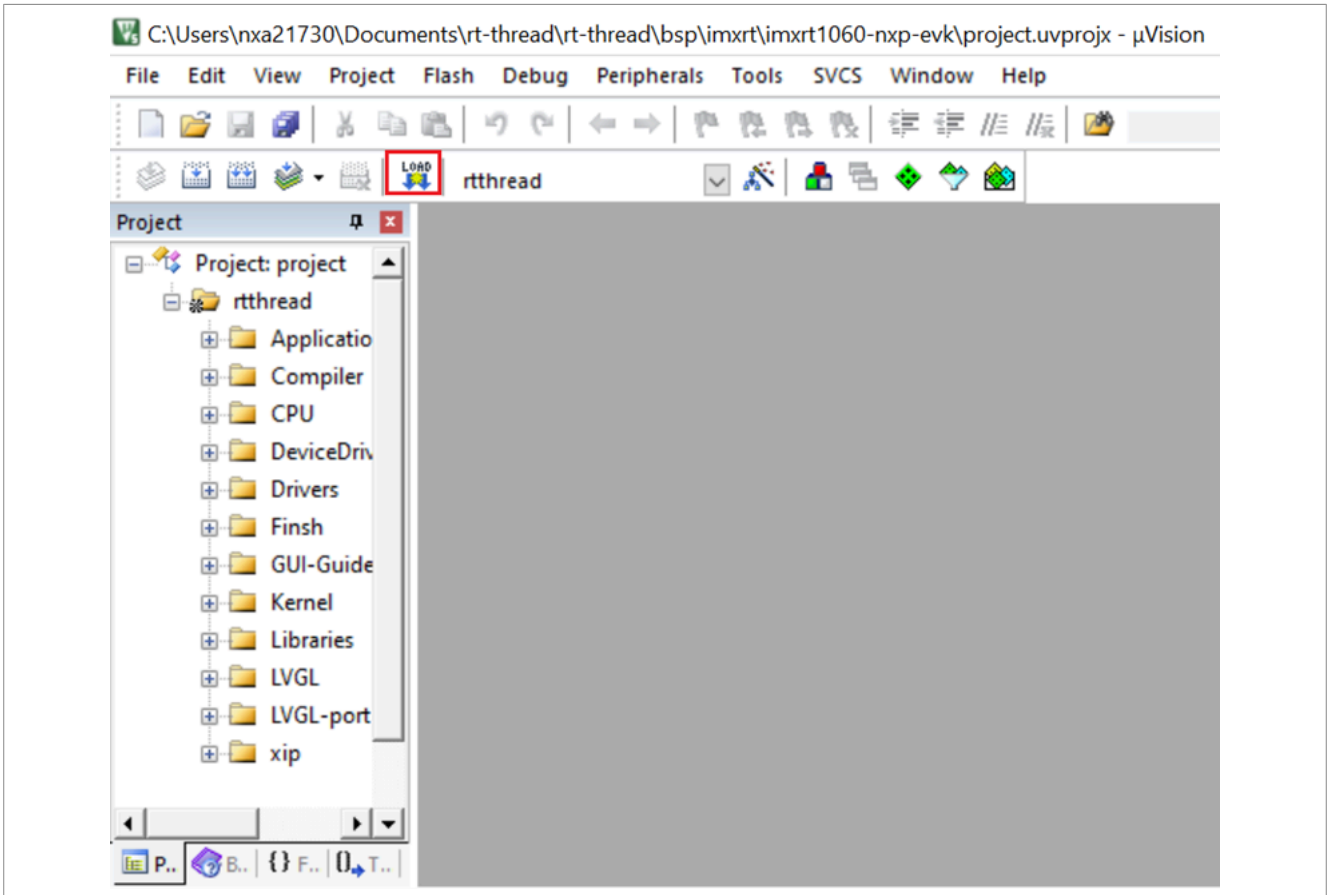
1. Double-click Keil project file "project.uvprojx" in `D:\rt-thread\rt-thread\bsp\imxrt\imxrt1060-nxp-evk` and rebuild all the files.

Note: If the following error appears, update the corresponding source file to replace "lvgl/lvgl.h" with "lvgl.h".

Error: src.c(10): error: 'lvgl/lvgl.h' file not found s



2. Connect i.MX RT1062 to PC with a USB cable. Click "Download (F8)".



After performing the above steps, the GUI application designed by GUI Guider can be compiled in RT-Thread and run on i.MX RT1060 board.

10.1.8 Known Issues

- If option **Event > load screen > Delete current screen** is enabled, then the PC hangs when switching between different screens. The workaround is to disable the **Delete current screen** when loading a new screen.

10.2 Zephyr

This section describes how to port the LVGL C source file generated by GUI Guider for Zephyr.

10.2.1 Set up Zephyr build environment

To set up Zephyr build environment on build machine, perform the steps below. Refer to [Zephyr getting started guide](#) for more details.

1. Install chocolatey (<https://chocolatey.org/install>).
2. Open the `cmd.exe` window as Administrator. To do so, press the Windows key, type "cmd.exe", right-click the result, and choose Run as Administrator.
3. To avoid having to confirm the installation of individual programs, disable global confirmation.

```
choco feature enable -n allowGlobalConfirmation
```

4. Use choco to install the required dependencies.

```
choco install cmake --installargs 'ADD_CMAKE_TO_PATH=System' choco install
ninja gperf python git dtc-msys2 wget unzip
```

5. Install pyocd.

```
python3 -m pip install -U pyocd
```

6. Close the window and open a new `cmd.exe` window as a regular user to continue.

10.2.2 Get Zephyr and install Python dependencies

To download Zephyr and install Python dependencies, perform the steps below:

1. Install west. It is recommended to install west as Administrator.

```
pip3 install -U west
```

2. Get the Zephyr source code.

```
cd %HOMEPATH%
west init zephyrproject
cd zephyrproject
west update
```

3. Export a Zephyr CMake package. This allows CMake to automatically load boilerplate code required for building Zephyr applications.

```
west zephyr-export
```

4. Zephyr's `scripts\requirements.txt` file declares additional Python dependencies. Install them with pip3.

```
pip3 install -r %HOMEPATH%\zephyrproject\zephyr\scripts\requirements.txt
```

10.2.3 Install Zephyr SDK

To install Zephyr SDK on build machine, perform the steps below:

1. Open the `cmd.exe` window.
2. Download the latest Zephyr SDK bundle.

```
cd %HOMEPATH%
wget https://github.com/zephyrproject-rtos/sdk-ng/releases/download/v0.15.2/
zephyr-sdk-0.15.2_windows-x86_64.zip
```

3. Extract the Zephyr SDK bundle.

```
unzip zephyr-sdk-0.15.2_windows-x86_64.zip
```

Note: Extract the Zephyr SDK bundle at one of the following locations.

```
%HOMEPATH%
%PROGRAMFILES%
```

The Zephyr SDK bundle contains the `zephyr-sdk-0.15.2` directory and, when extracted under `%HOMEPATH%`, the resulting installation path is `%HOMEPATH%\zephyr-sdk-0.15.2`.

4. Run the Zephyr SDK bundle setup script.

```
cd zephyr-sdk-0.15.2
setup.cmd
```

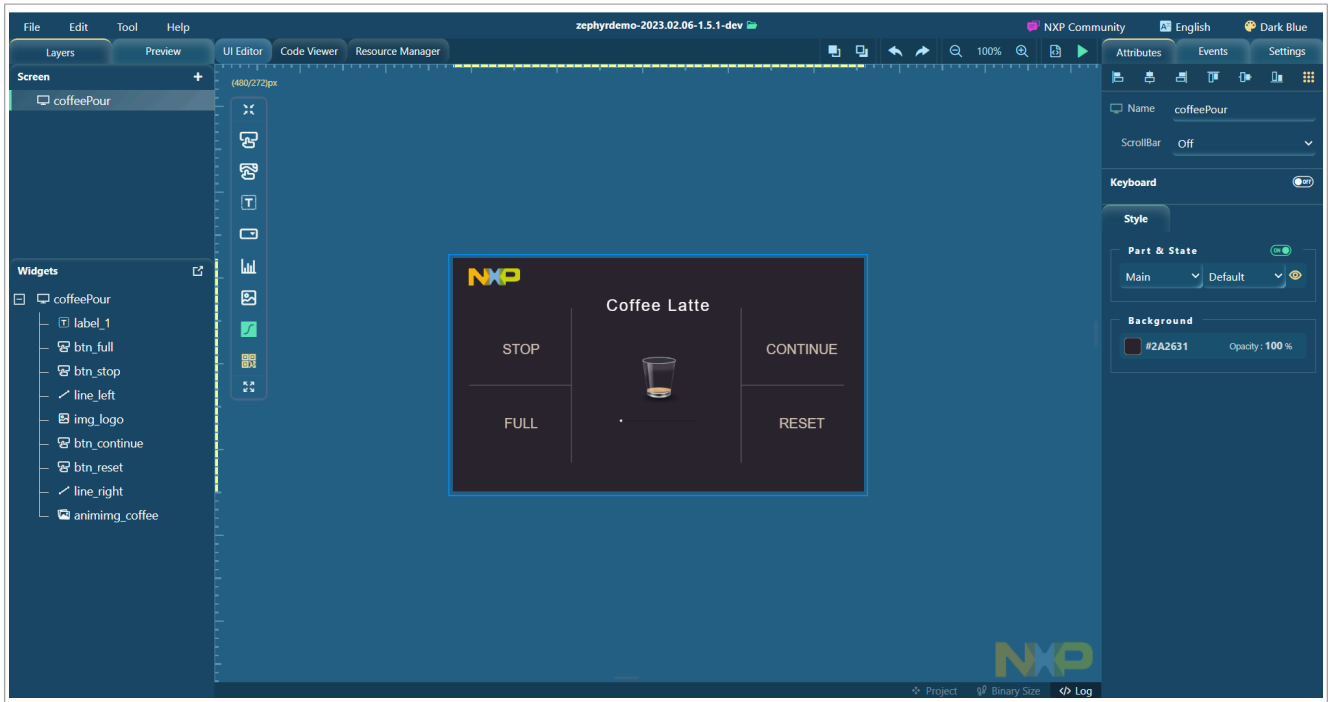
Note: You only need to run the setup script once after extracting the Zephyr SDK bundle.

Note: If you relocate the Zephyr SDK bundle directory after the initial setup, Rerun the setup script.

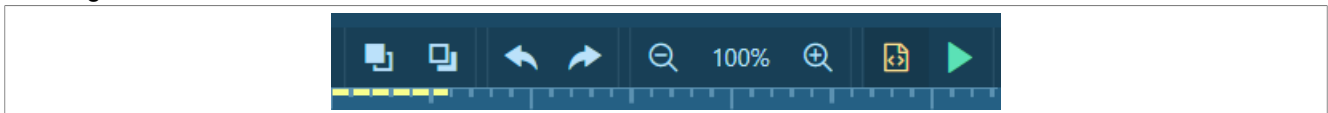
10.2.4 Design GUI and export code by GUI Guider

To export the source code of GUI application designed by GUI Guider to Zephyr, perform the steps below:

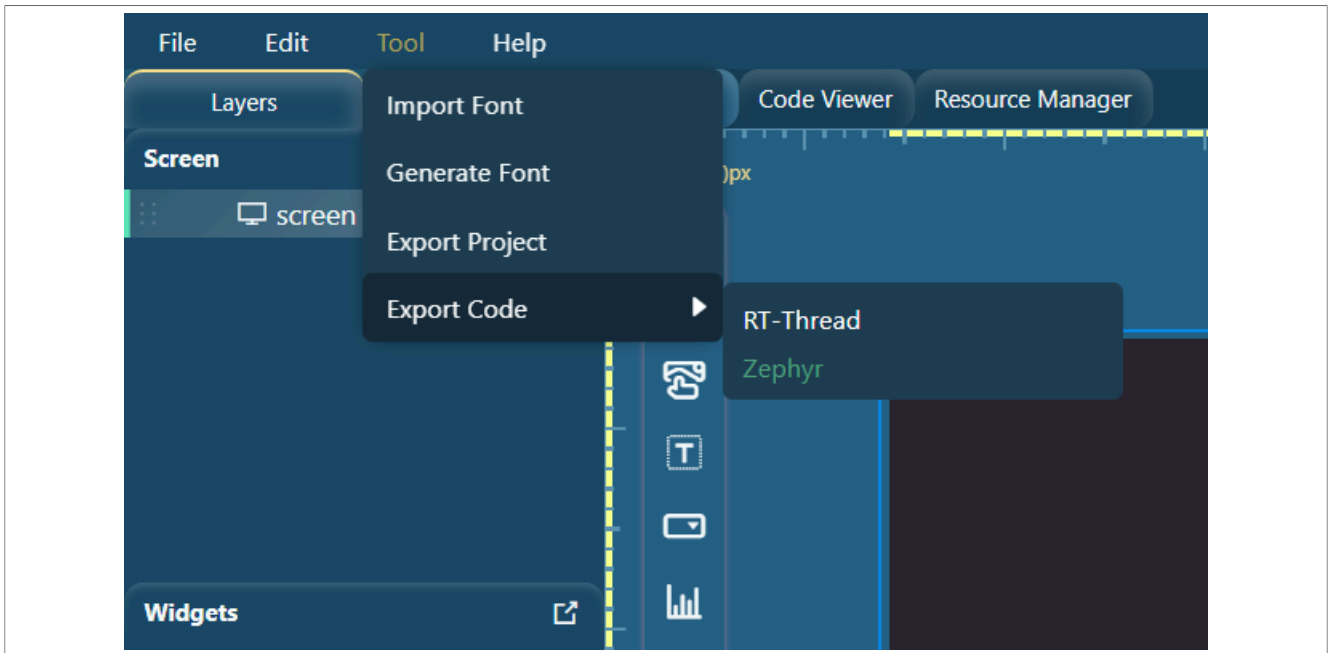
1. Design GUI application using GUI Guider.



2. To generate the GUI source code, click the "Generate Code" button.



3. Export the source code of GUI application to a local folder, for example, C:\Users\user1\zephyrproject\gui_guider_demo.



10.2.5 Build and deploy Zephyr image

Run the following commands to build and deploy Zephyr images on target board.

```
> west build -c -p always -b mimxrt1050_evk C:\Users\user1\zephyrproject
\gui_guider_demo
> west flash --runner pyocd
```

11 External storage

This section describes how to use external storage (SD card) in the GUI Guider project.

11.1 Supported devices

External storage is supported on selected devices currently.

- Video file (H264)
 - NXP i.MX RT1050
- Image file (BIN, BMP, JPG, PNG)
 - NXP i.MX RT1064, LPC54S018J4M

11.2 Video

The format of the input video should be *.h264. The demo application implements video play on NXP MCU by using the LVGL library.

11.2.1 Prerequisites

- MCUXpresso 11.7.1 / IAR 9.30.1 / Keil MDK 5.37.0.
- GUI Guider v1.5.1 GA.

- Connect the supported device to the host by a USB cable.

11.2.2 Prepare h264 file

To prepare an h264 video file, perform the following steps:

1. Install [FFmpeg](https://ffmpeg.org) (<https://ffmpeg.org>) on your host.
2. Use the following command to convert the mp4 video file to an h264 video file.

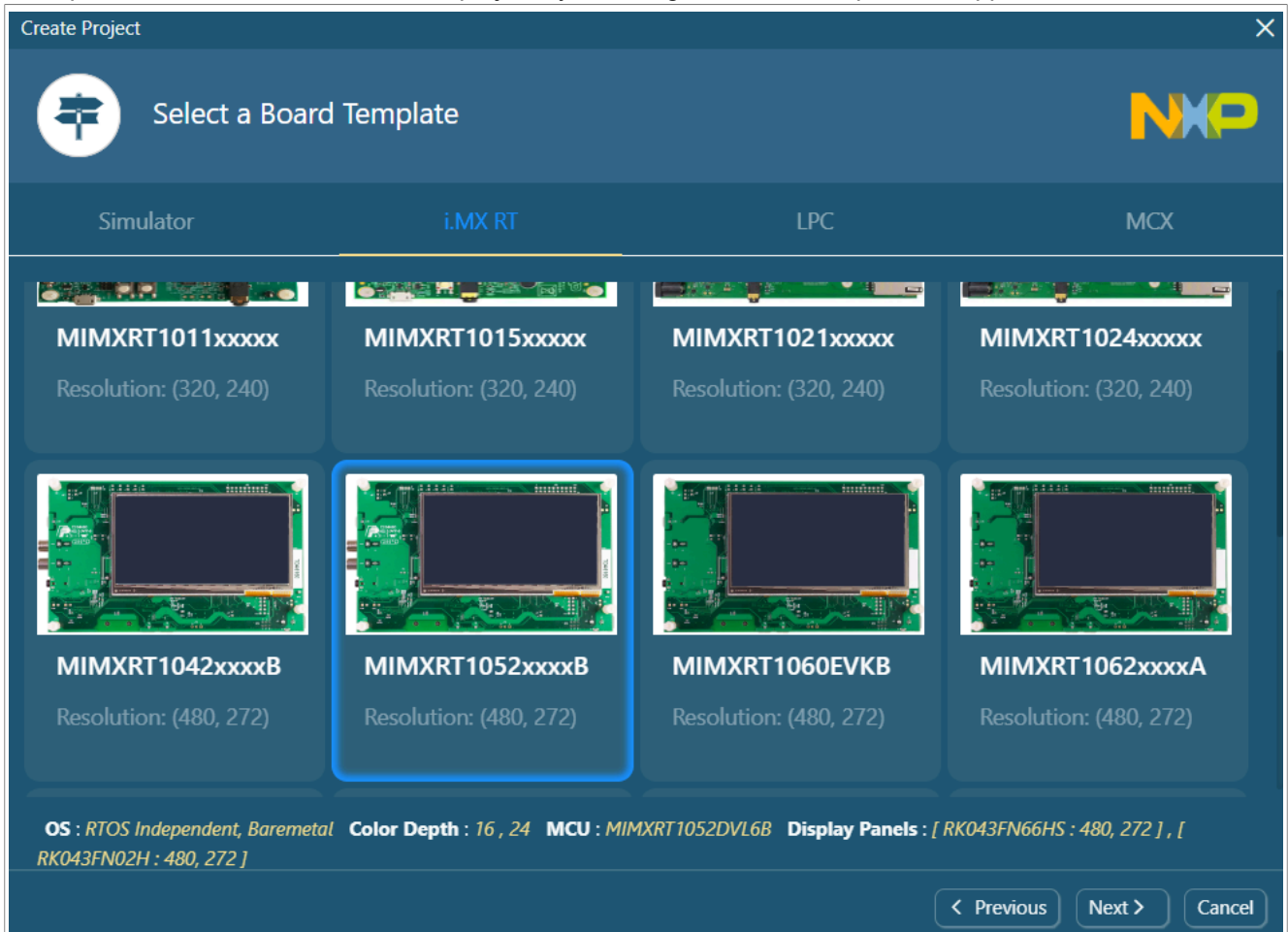
```
$ ffmpeg -i input.mp4 -vf scale=480:272 -c h264 output.h264
```

3. To get the best play effect, make the *scale* identical to the expected size in the GUI application.
4. Load the *.h264 video on the SD card of the MCU device.

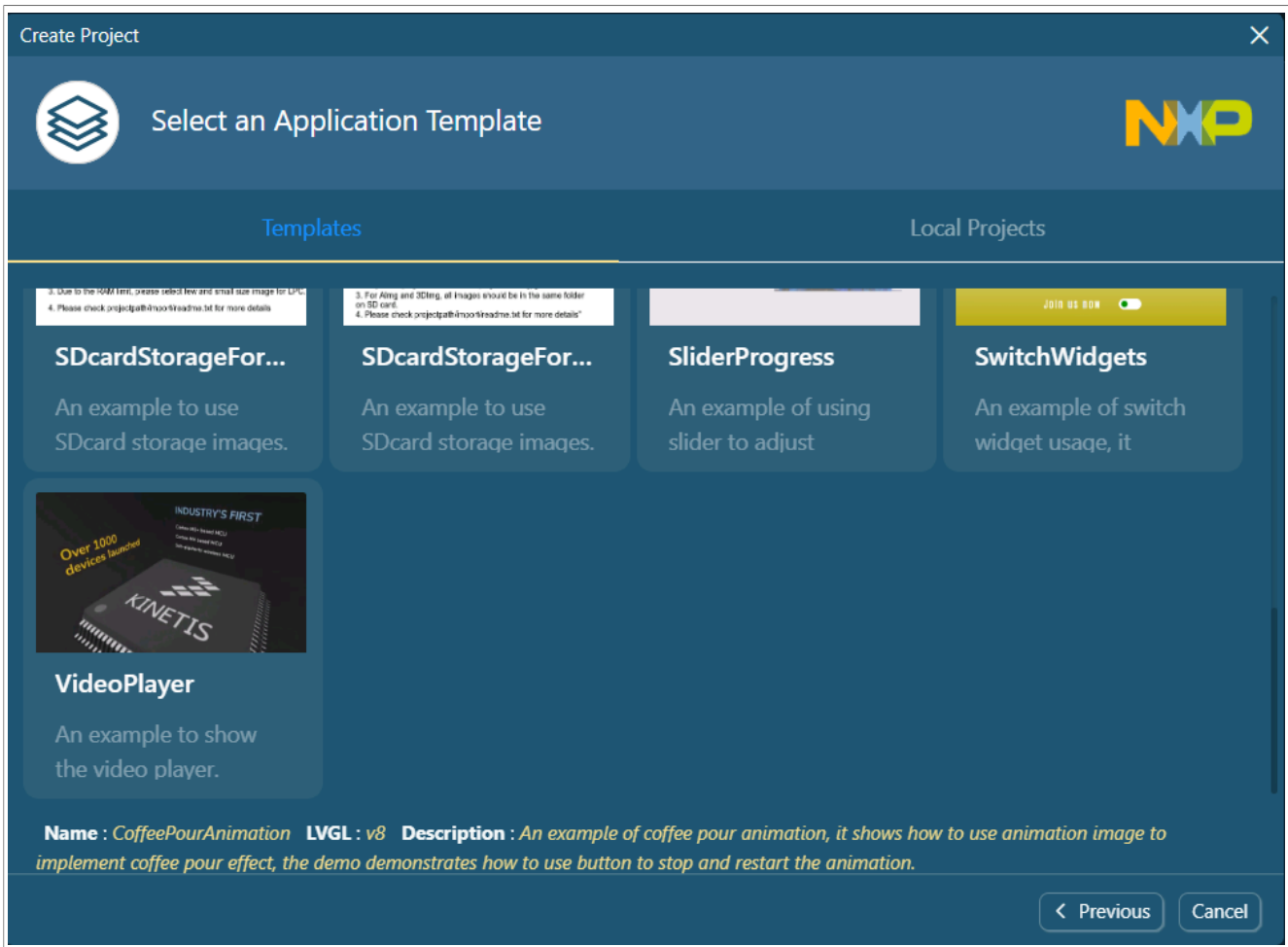
11.2.3 Create project

To create a project, perform the following steps:

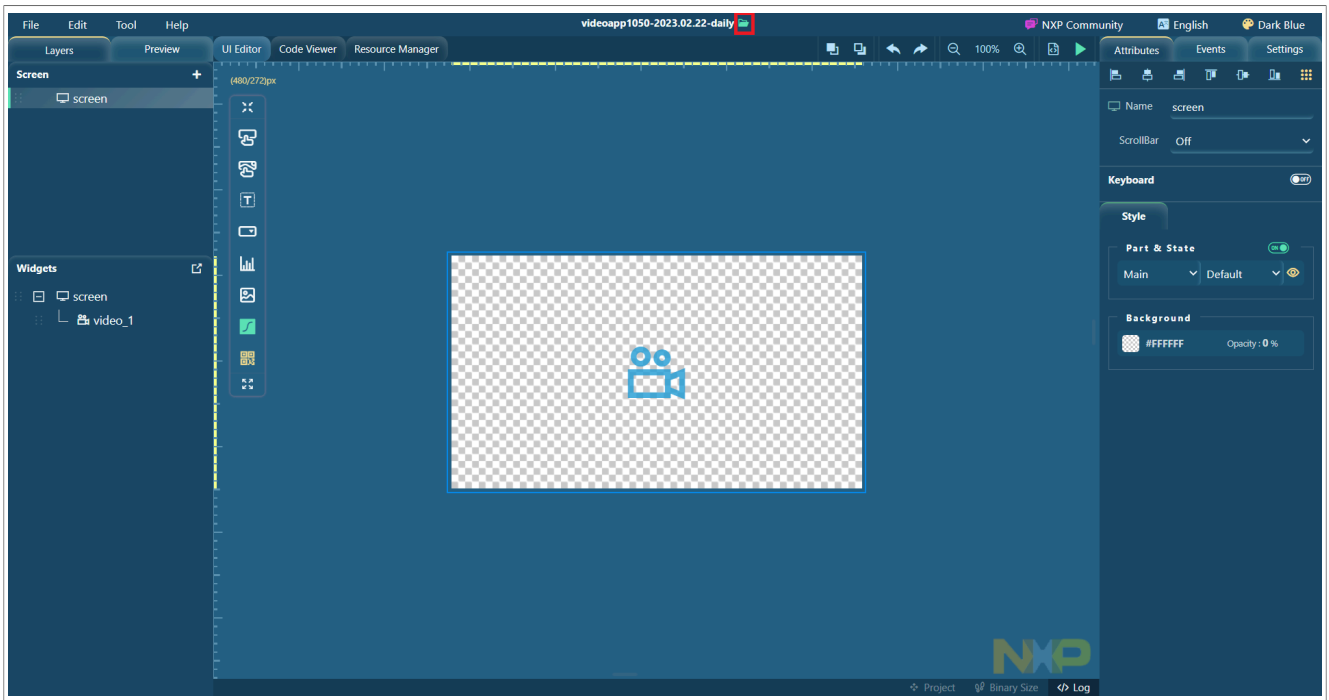
1. Open GUI Guider IDE and create a project by selecting the device template of supported boards.



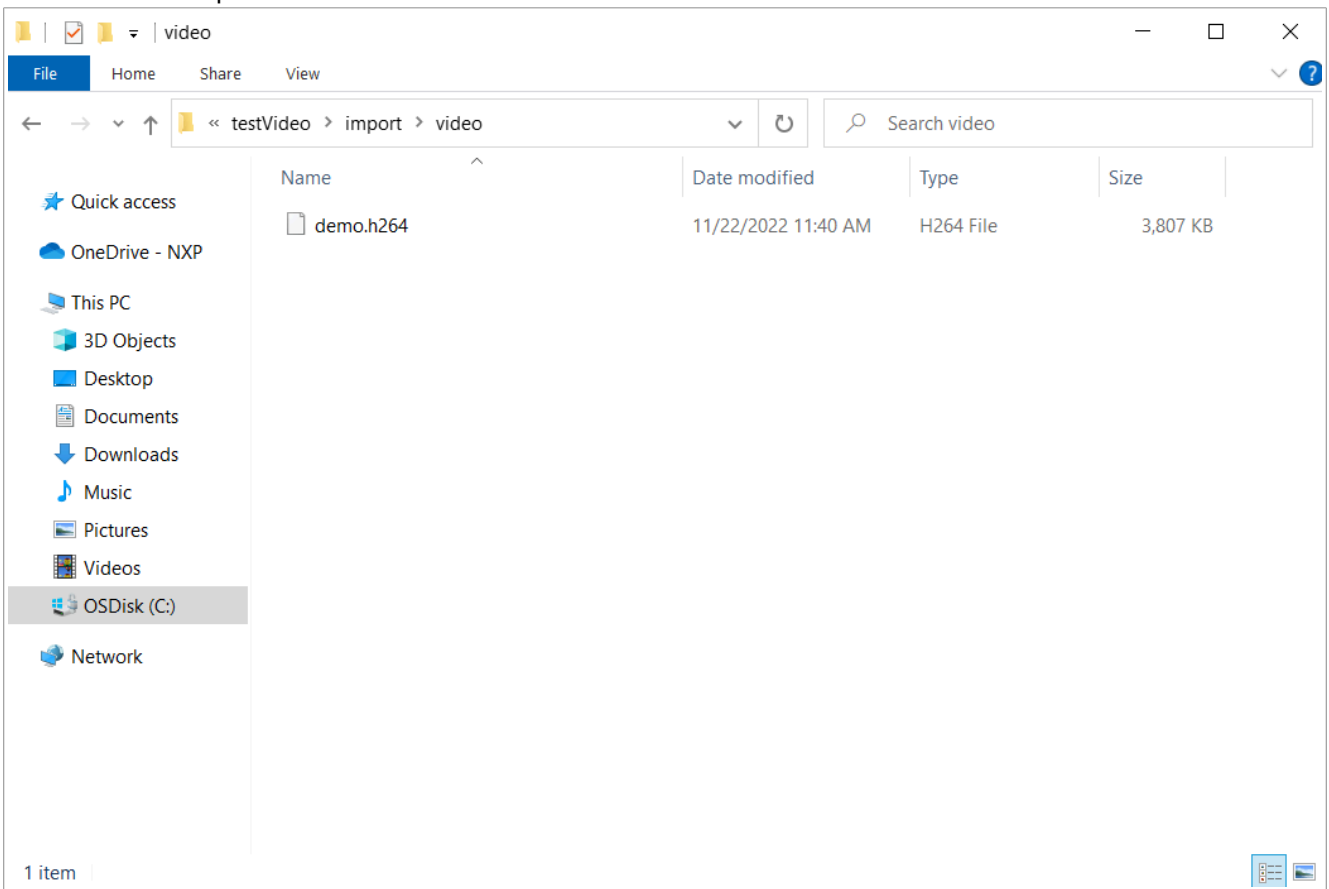
2. Select VideoPlayer application template.



3. Open the project folder by clicking the "Open project directory" icon, which is in the right-upper corner of the preview window.



4. Go into the import folder and locate the demo.h264 file.



5. Copy the demo.h264 file into SD card of the MCU device.

6. To load the code into your device, click **Run > Target > MCUXpresso**. After finishing, you can play this demo video on display.

Note: Because the conversion of YUV420 to rgb565 at the board end is through PXP, PXP line selection cannot be enabled in the video demo in the GUI Guider.

11.2.4 Customize project

The name and path of the *.h264 video file can be changed in "custom.c". Different variables are used to set the simulator and target.

11.3 Image

The images (BIN, BMP, JPG, PNG) can be stored on an SD card and used by image-related widgets (imgbtn, image, Aimg, 3Dimg). The images are decoded in runtime.

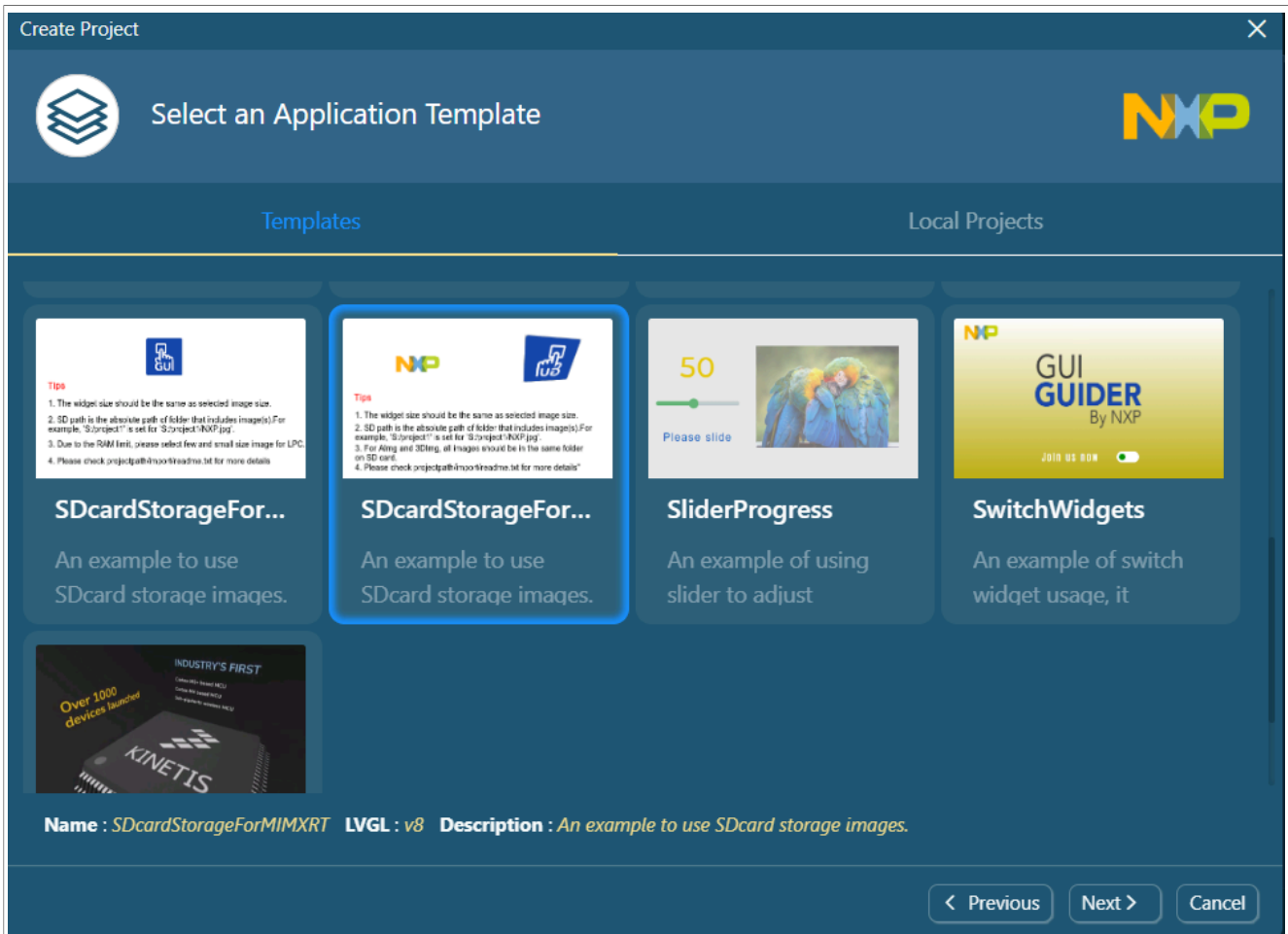
11.3.1 Prerequisites

- GUI Guider 1.5.1 GA.
- Connect the supported device to the host by a USB cable.

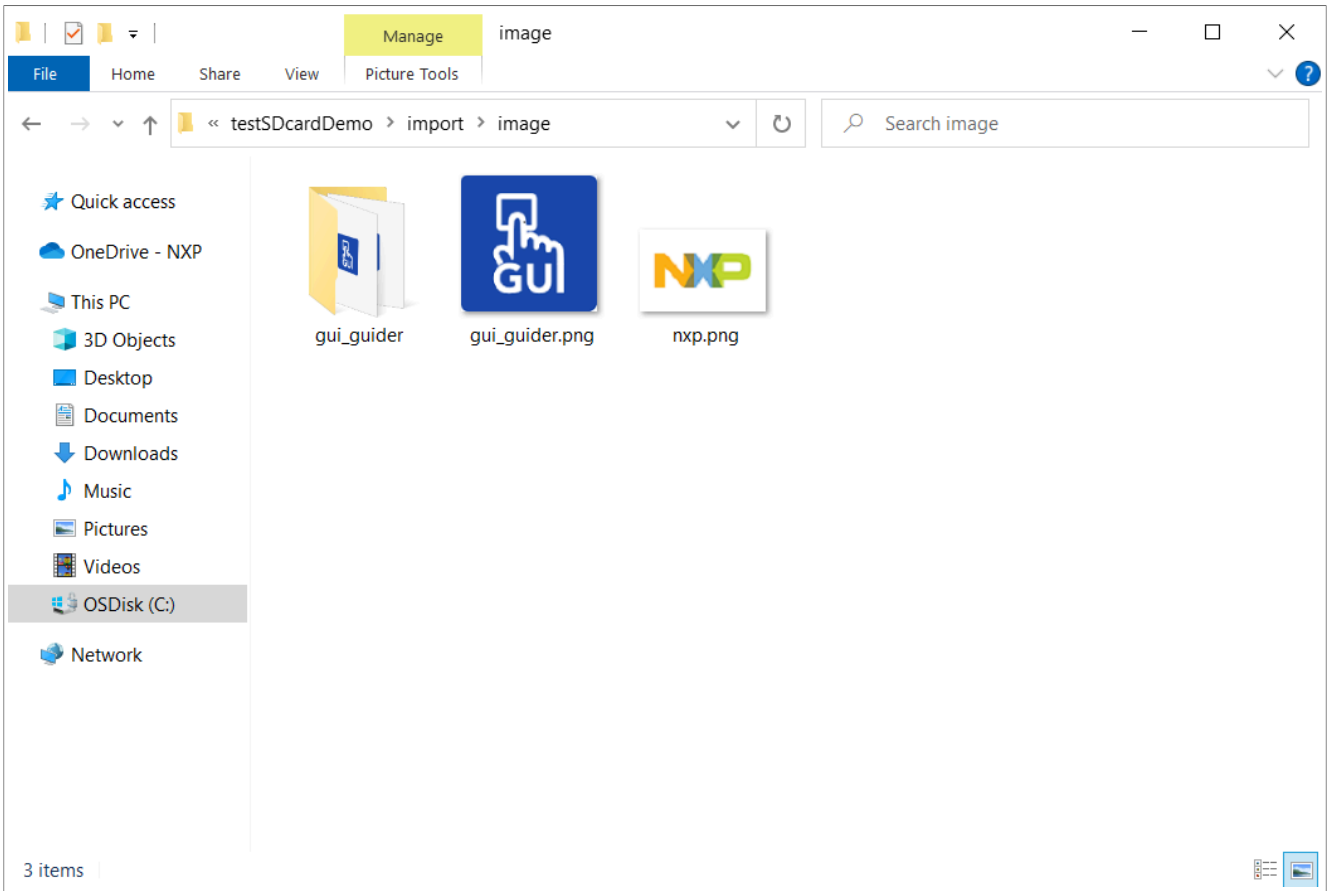
11.3.2 Project template

There is an application template for supported boards. The usage tips are displayed on the screen.

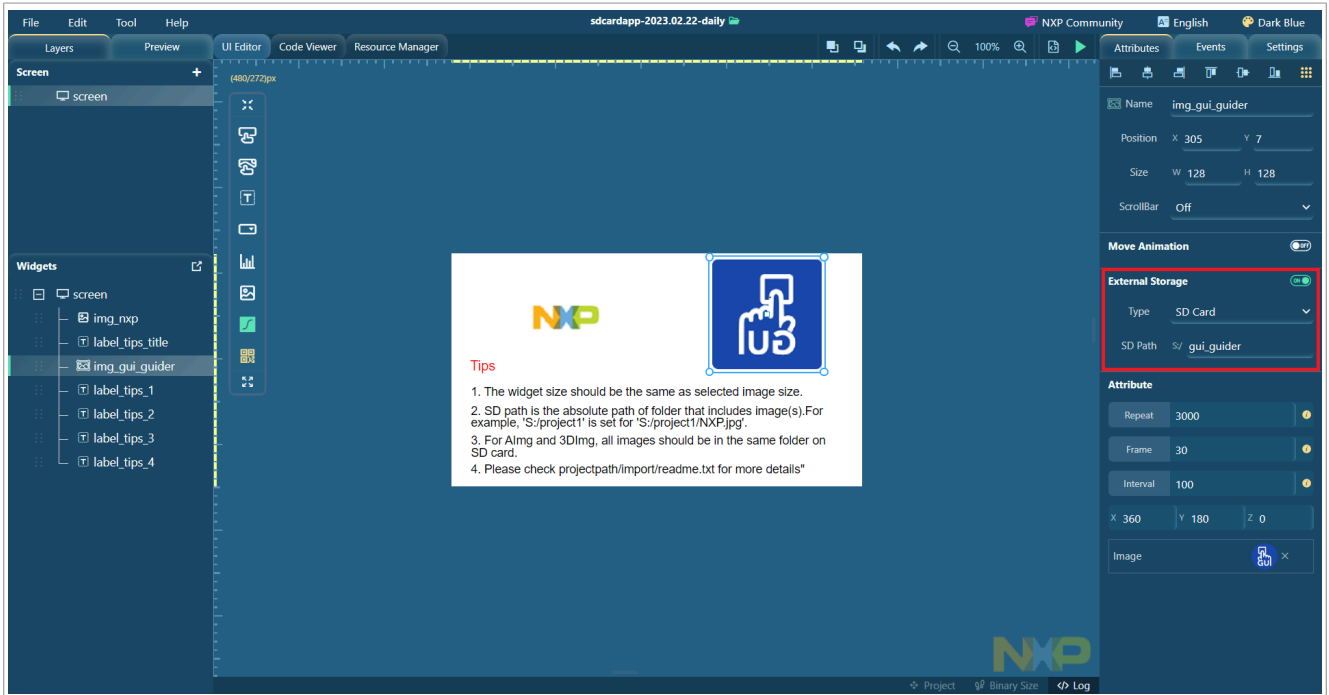
1. Create a project by selecting one supported board and the "SDcardStorage" application template.



2. Open the project folder by clicking the "Open project directory" icon, which is in the right-upper corner of the preview window.
3. The image files are located in the import folder.



4. The SD card storage can be enabled in the attribute setting window of image-related widgets.



Copy the images into the SD card of the MCU device. If SD card storage is enabled, keep the SD path the same as the actual file folder on SD.

Note: For the LVGL file system, use single / as a path separator.

11.3.3 Customized project

When using image-related widgets, follow the steps to store images on the SD card.

1. Drag and drop the widget into the editor and set the attributes.
2. Open project folder > import. Confirm one or more image files that you want to save in SD, and copy those images to the SD card of the MCU device.

Note: All image files of *Aimg* and *3Dimg* should be in the same folder on the SD card.

3. Enable "External Storage" in the attributes setting window, and input the absolute path of the folder that includes one or more images.
4. Insert SD card to MCU device. To deploy the application on the board, click **Run > Target**.

12 External image in QSPI Flash

To provide maximum flexibility, LVGL supports several image sources:

- A variable in code (a C array with the pixels)
- A file stored externally (for example, on an SD card or QSPI flash)
- A text with Symbols

An attractive GUI is reliant upon well-designed images. The more complex the GUI is, the more these assets are required. Building the images into firmware is impossible if the selected MCU does not have adequate flash storage. Therefore, storing the images externally is a way to fit this case with reduced firmware size.

To use an external image, you must use LVGL's file system module and register a driver with some functions for the basic file operation. Go to the [File system](#) to learn more. Currently, GUI Guider supports:

- Images with PNG/JPG/BMP/binary format on an SD card with FatFs file system
- Images with binary format on QSPI flash with rawfs file system

As a demo, this document mainly focuses on putting images on imxrt1060evk QSPI flash.

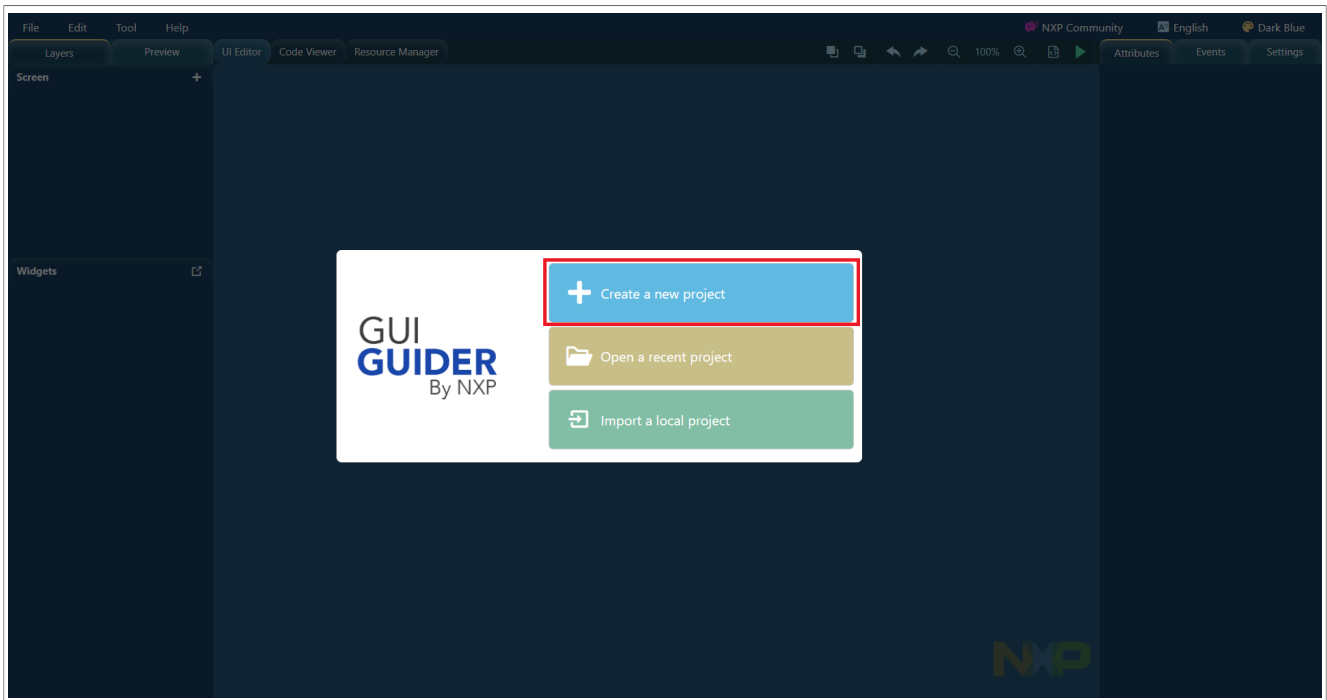
12.1 Prerequisites

- GUI Guider 1.5.1
- MCUXpressoIDE or Keil or IAR

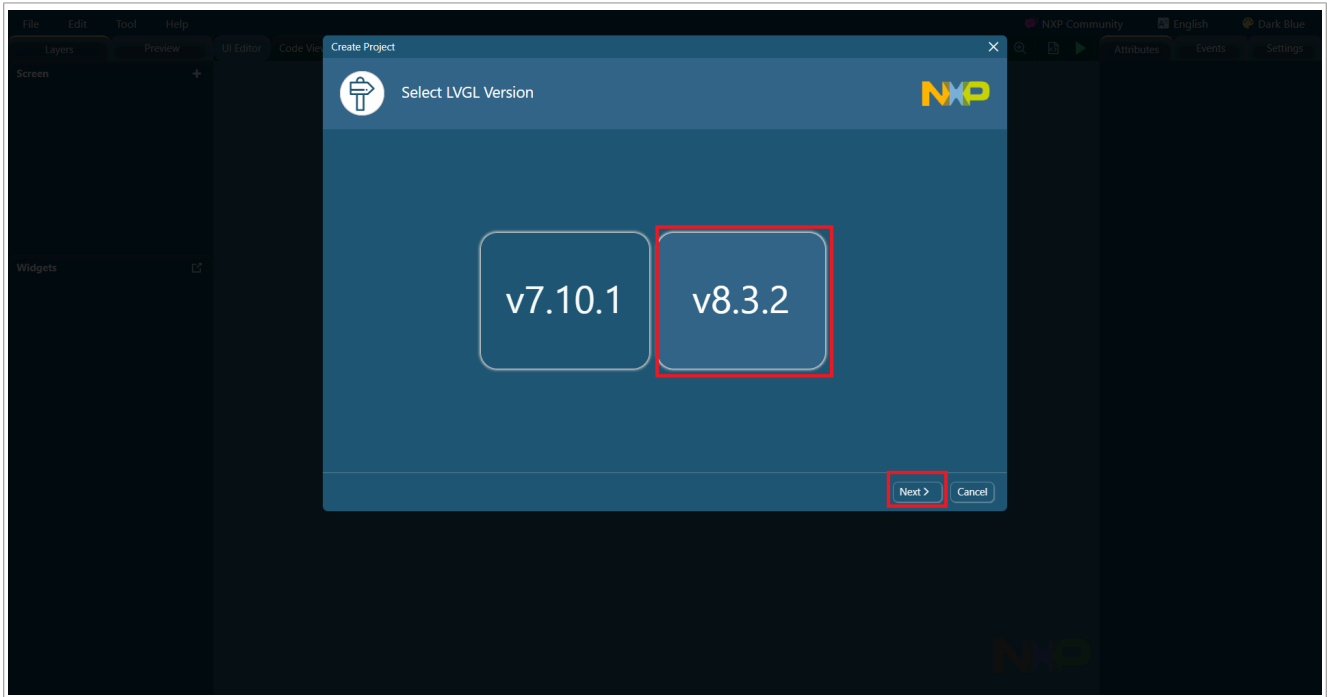
12.2 Create a new project

To create a GUI Guider project, follow the steps below:

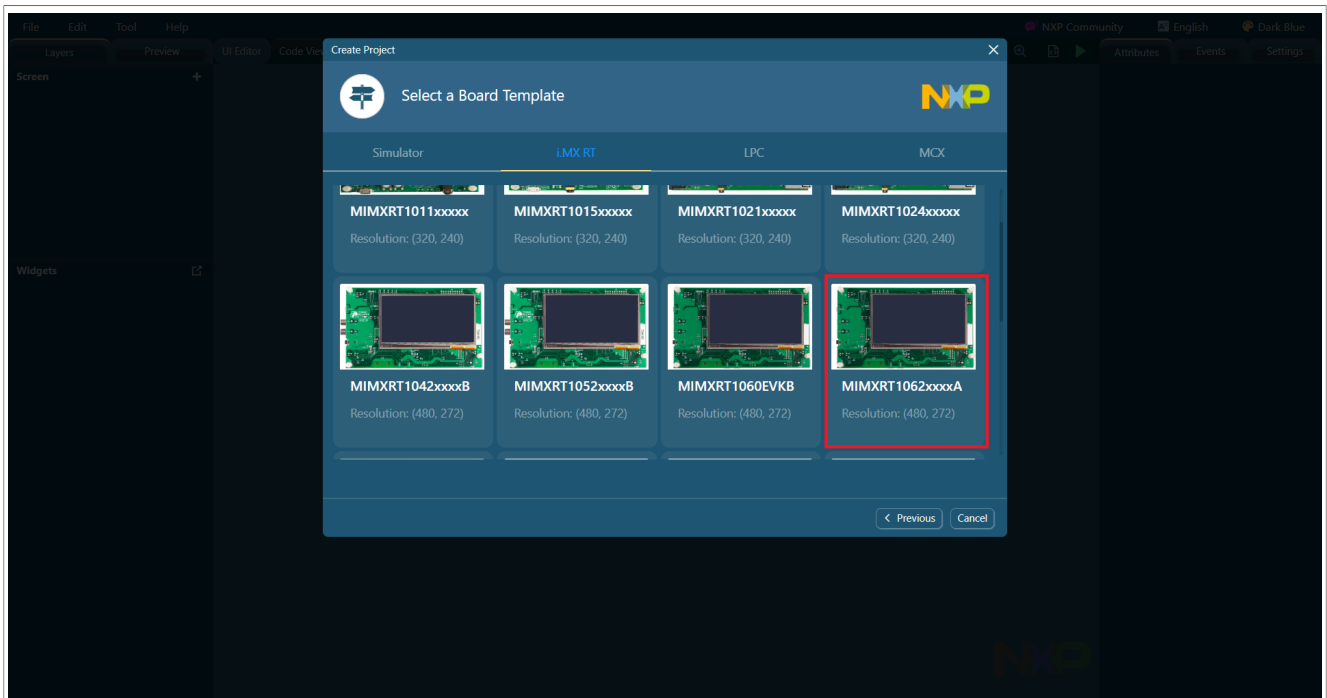
1. select the **Create a new project** button.



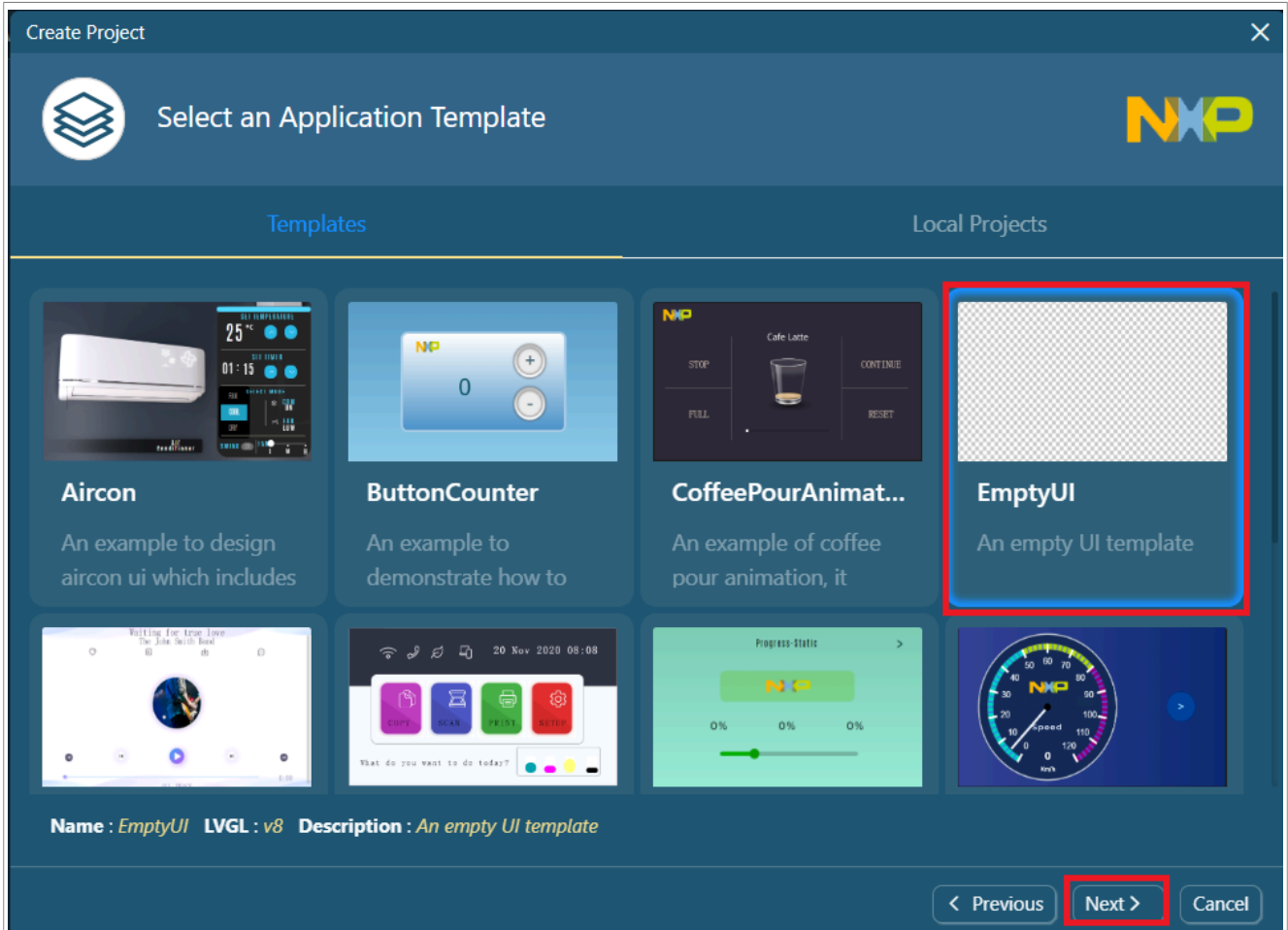
2. Select LVGL version as **v8.3.2** and click the **Next** button.



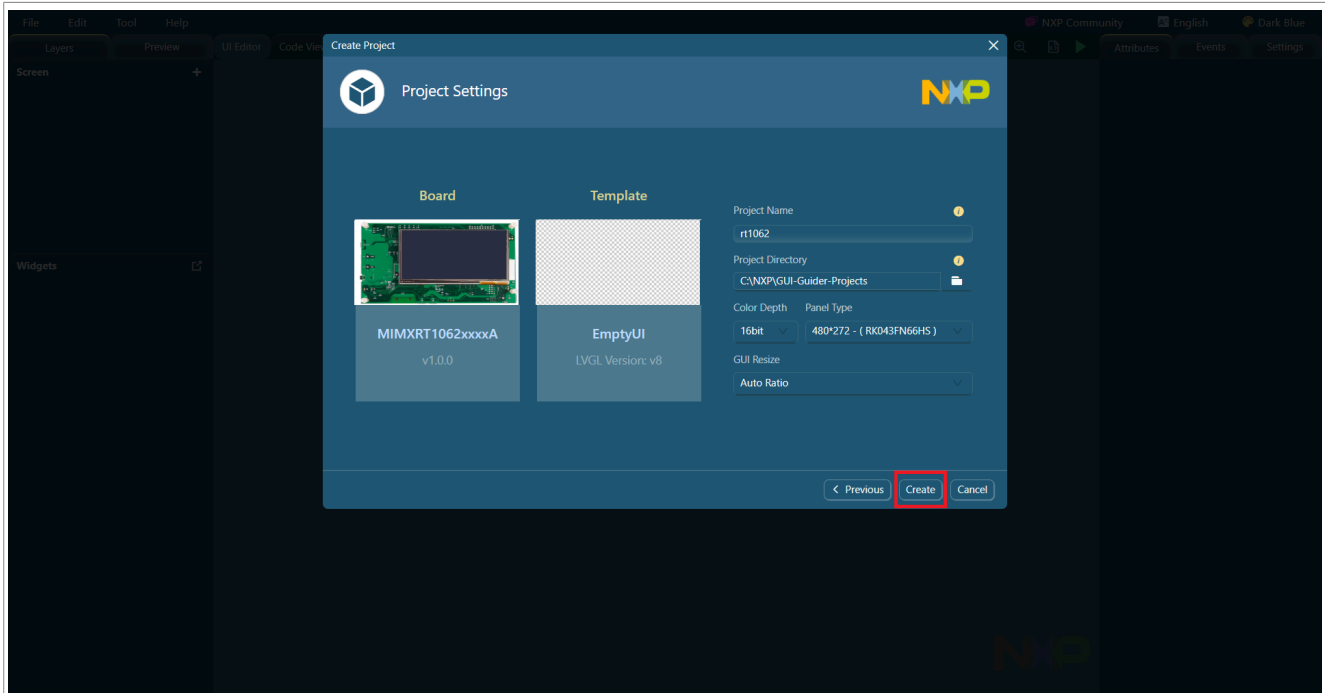
3. Select **MIMXRT1062XXXX** as target board template and then click the **Next** button.



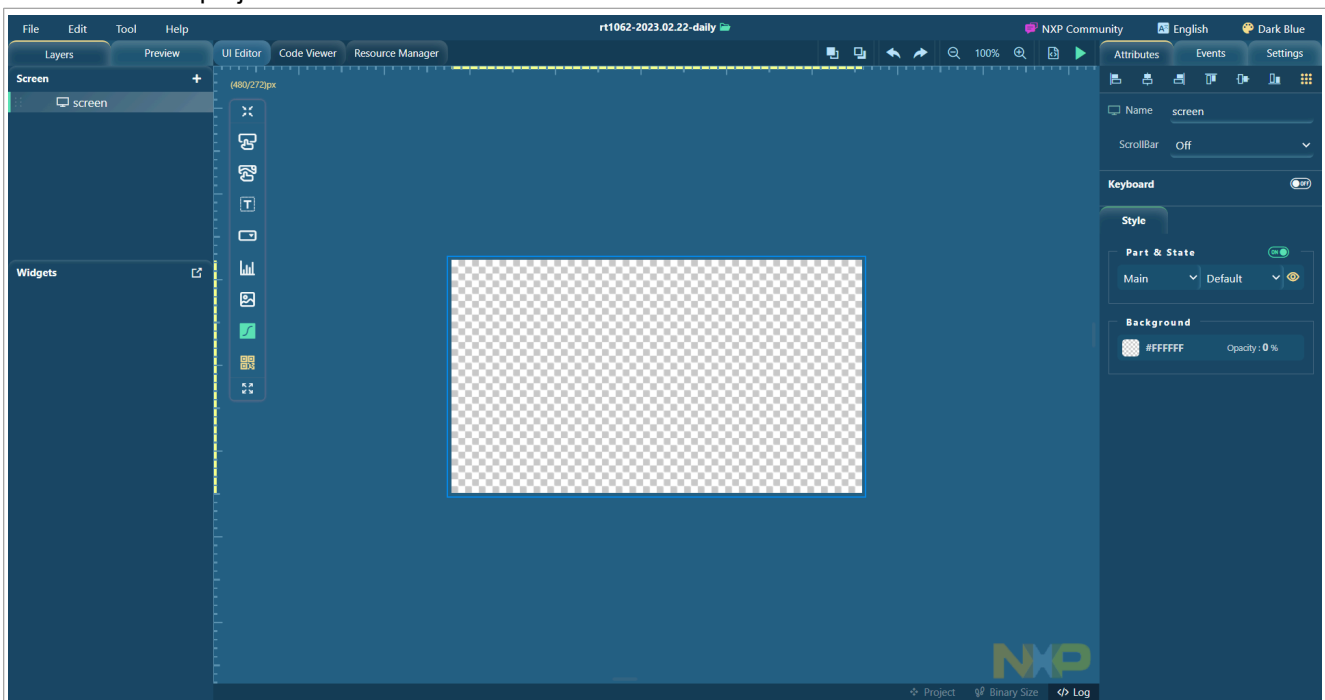
4. Select an empty application template, **EmptyUI**.



5. Perform project settings, such as project name, project location, panel type, and color depth. Click **Create**.



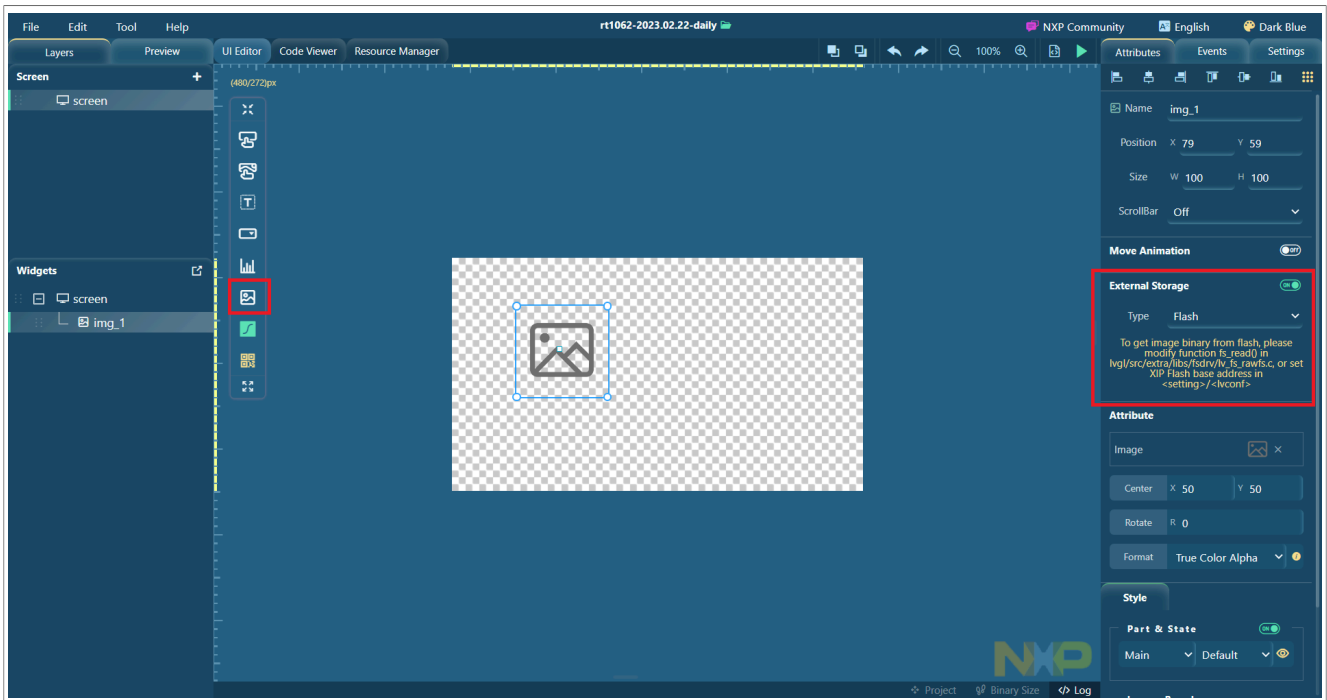
6. The created project is shown.



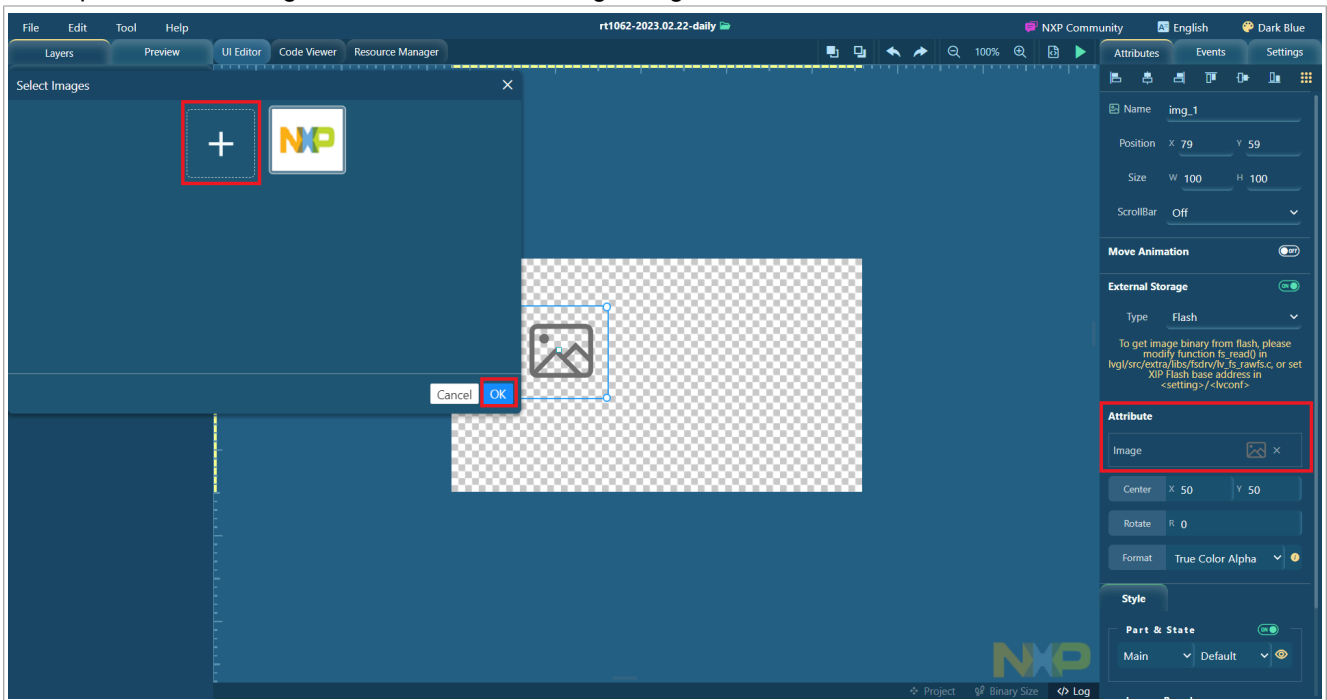
12.3 Add image widget and set property

To add an image widget and set its properties, follow the steps below:

1. Click **img** in the **Widgets** tab and a new image widget is added into the workspace. Set "External Storage" as "on" and make sure "flash" type is selected.



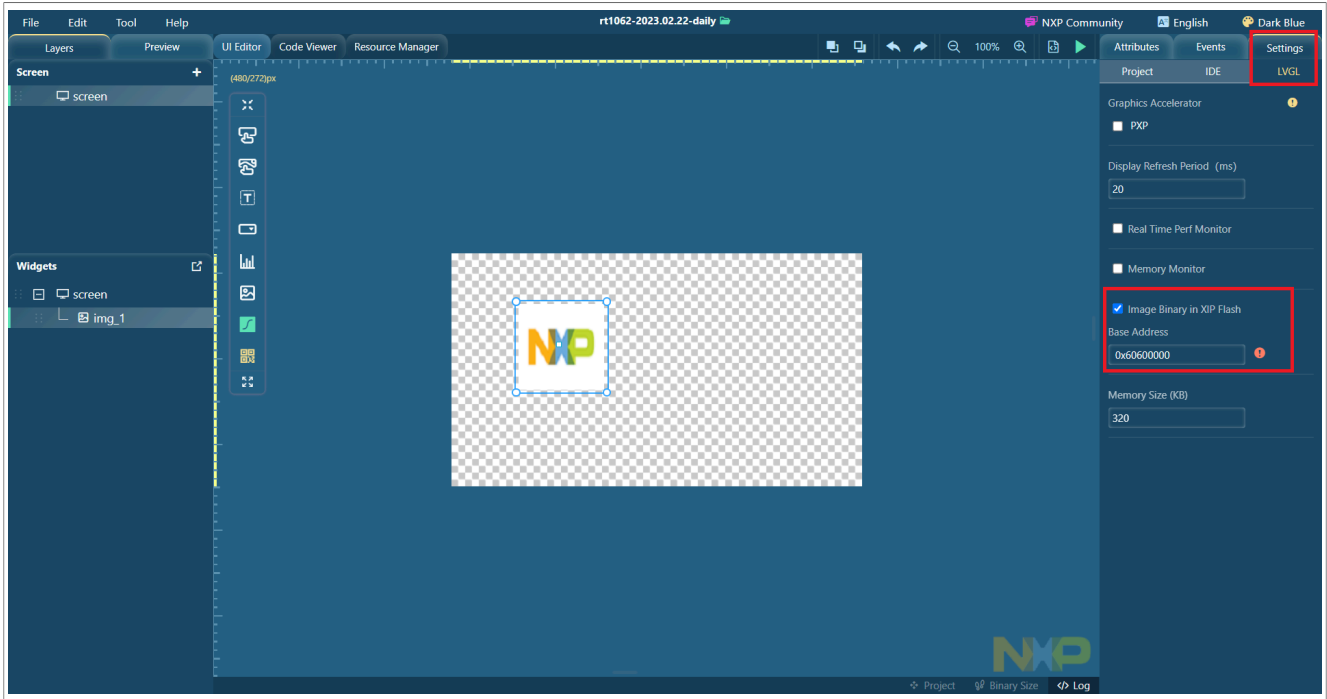
2. Import the local image file and set it in the image widget attribute.



12.4 Build and deploy

Connect the supported device to the host by a USB cable. Follow the steps below:

1. Select **Image Binary in XIP flash** in **Setting > IvConf**, and set "Base Address" as **0x60600000**.

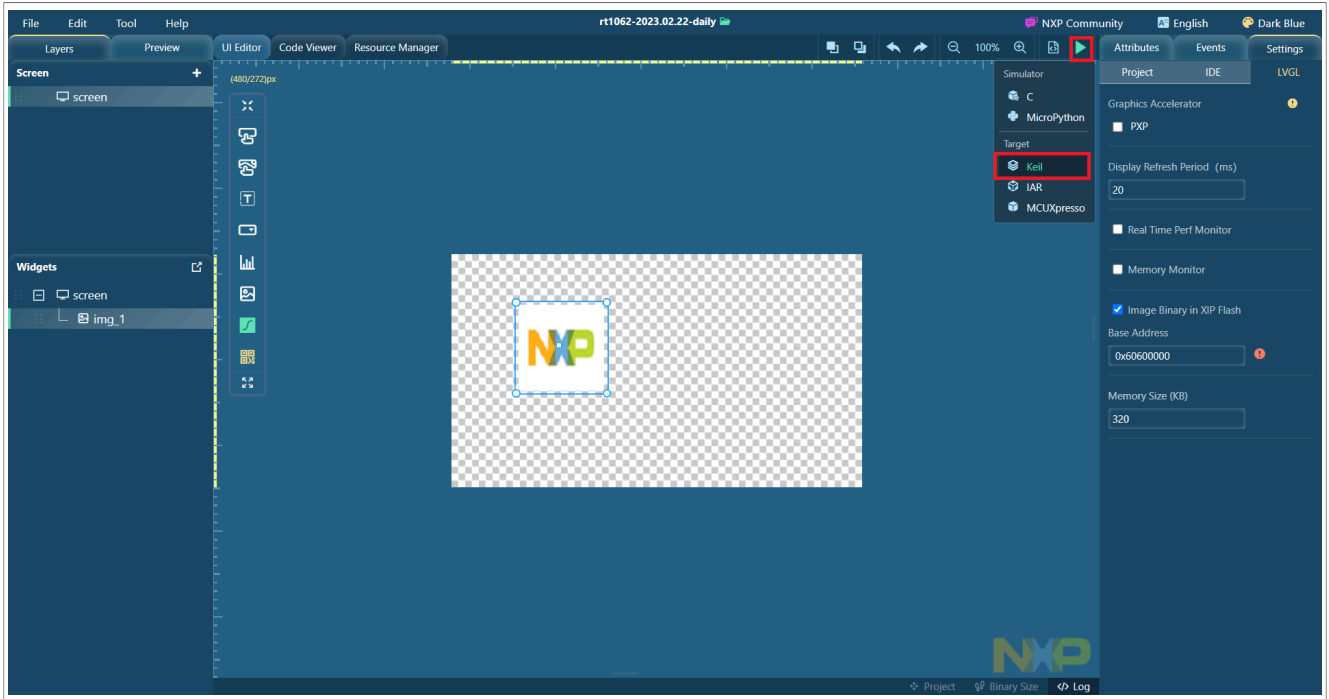


Note: In computer science, **execute in place (XIP)** is a method of executing programs directly from long-term storage. For this to work, one requirement is that the storage must provide a similar interface to the CPU as regular memory. In this demo, we put the image binary on XIP flash, which can be accessed directly via a base address. This is just for the demonstration of how external images are supported. It is more efficient and simple to link the image as a C array with the pixels in the firmware.

Note: Another real case in which the flash storage cannot be accessed directly via a base address is covered in the subsection below.

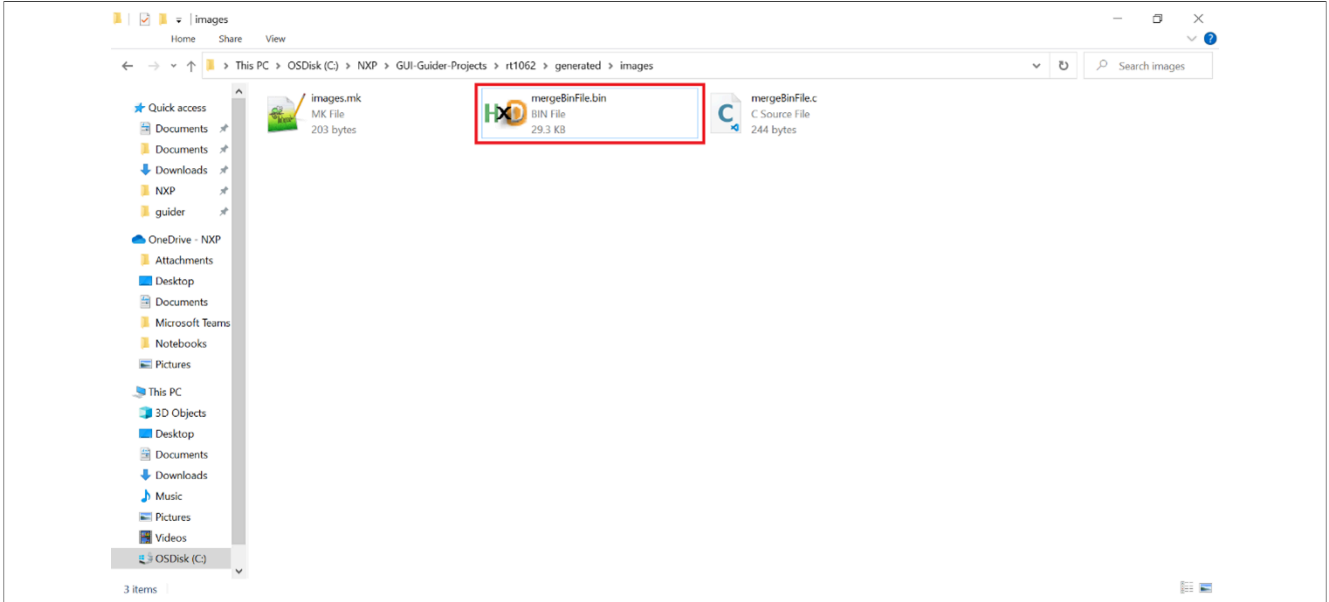
Note: imxrt1060evk ships an 8 MB Quad SPI flash, with the address from 0x60000000 to 0x607FFFFFFF. In this demo, we put the image binary to flash starting at address **0x60600000**, leaving 2 MB space for the image binary and 6 MB space for the firmware. You can adjust it based on the size of the image binary and firmware.

2. Generate source code, build, and deploy. We choose "Keil" as an example here.



Note: There is no output as the image binary is not programmed to the flash until now.

3. Program the merged image binary to flash at address **0x60600000**. The image binary can be found at `<project path>/generated/images/mergeBinFile.bin`.

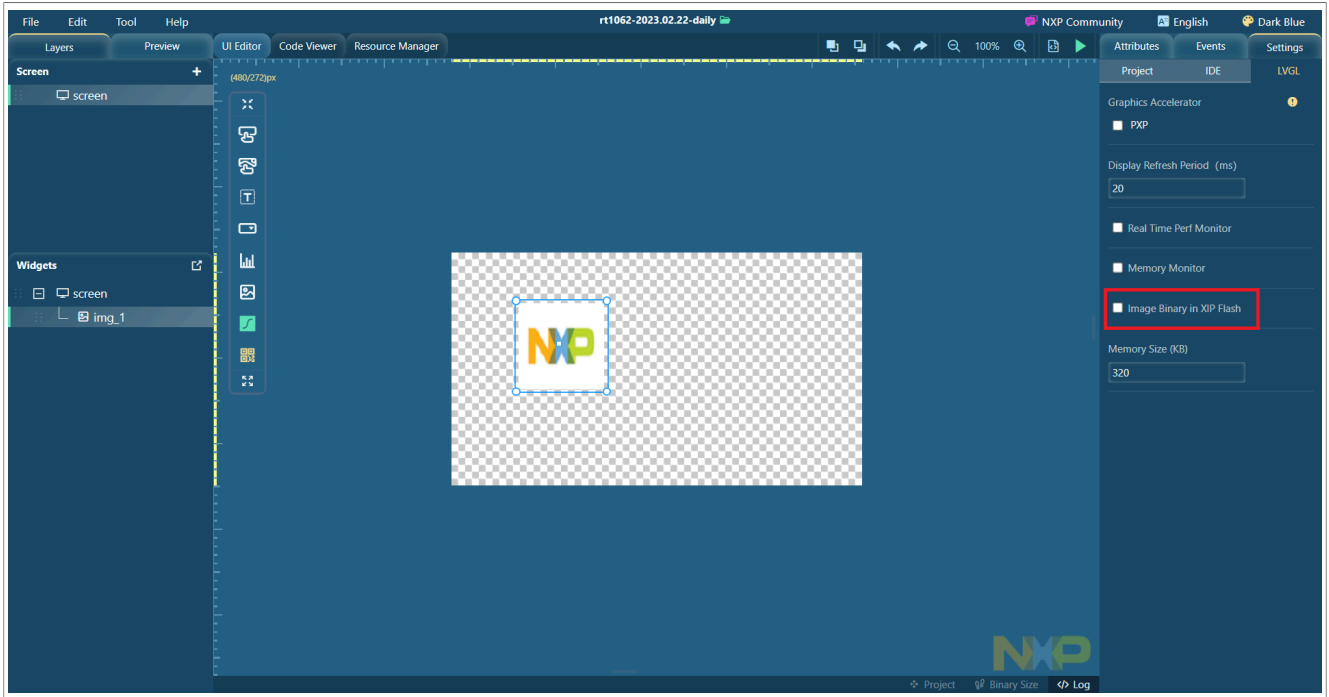


4. Reset the board. The image widget should be shown normally.

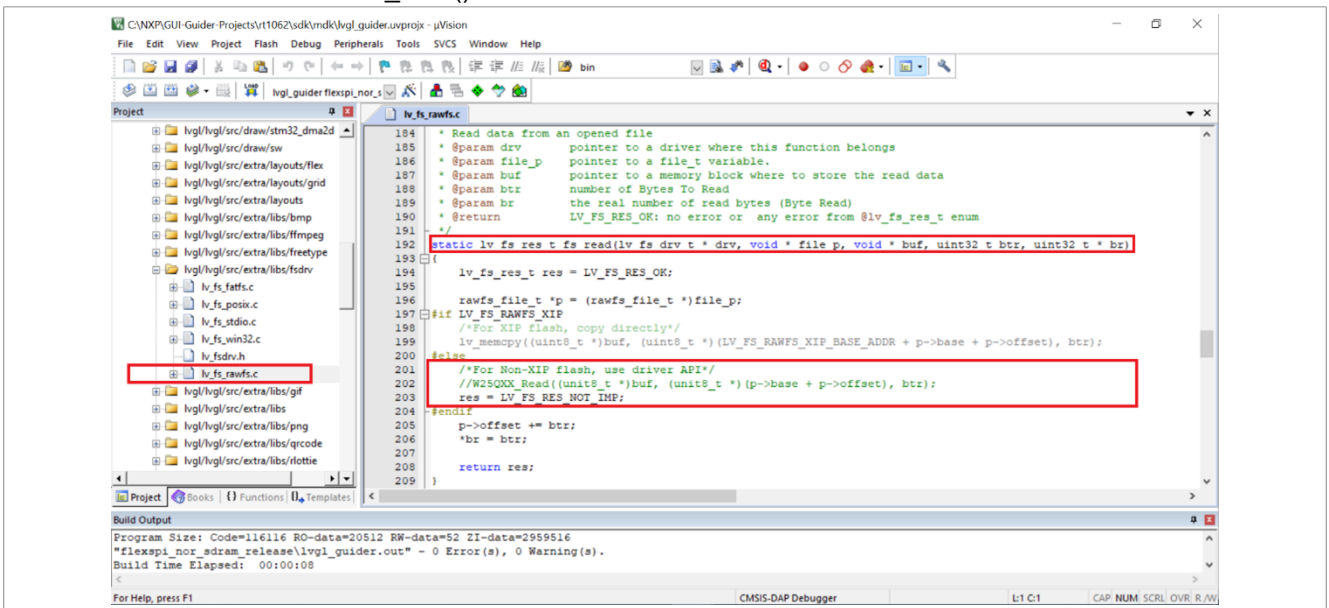
12.5 Image Binary in non-XIP flash

One real case is that the flash cannot be accessed directly via a base address. In this case, you must import the corresponding flash driver and implement the read logic in function `fs_read()` of file `lvgl/src/extra/libs/fsdrv/lv_fs_rawfs.c`. Follow the steps below:

1. Deselect the **Image in XIP Flash** checkbox.



2. Modify the file `lvgl/src/extra/libs/fsdrc/lv_fs_rawfs.c`. Add code to implement the logic to read from flash in function `fs_read()`.



3. Rebuild the firmware and deploy it to the flash.
4. Program the merged image binary to flash. The image binary can be found at `<project path>/generated/images/mergeBinFile.bin`.
5. Reset the board.

13 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
```

Question: What should I do if the project upgrade fails?

Answer: The backup project can be found in the workspace folder if you enable the backup function during upgrade. Use previous version for development and report the issue in <https://community.nxp.com/t5/GUI-Guider/bd-p/GUI-Guider>.

Question: What to do if the effect is inconsistent after upgrading the project?

Answer: It is necessary to manually adjust the style to compare with the old project, for example, font size, font position.

Question: How to reuse applications on other boards?

Answer: When creating a project, select an existing project as a template, according to the auto-size function configuration board display.

Question: How to do the Meter control realize the rounded rectangle needle?

Answer: Use the image needle.

Question: How to resolve the issue if touch or display does not work?

Answer: The issue is possibly caused if an incorrect panel type is selected. Check the panel type of your project.

Question: When lottie widget is used in the project, some black lines flicker at the bottom of the screen. What should I do to fix this issue?

Answer: Import the project using the MCUXpresso IDE. See [Section 8.1](#), then build and deploy the image using the MCUXpresso IDE.

Question: How to customize and modify lv_conf macro without being automatically overwritten by GG?

Answer: Customize lv_conf.h options using lv_conf_ext.h in custom folder.

Question: How to resolve the mingw32-make build failure.

Answer: Check the Java version on the host machine and make sure that Java version is 1.8.0 or higher.

Question: How to add custom code for different widgets in event setting window?

Answer: When the custom code is updated for a widget, close the custom code editor and click another widget. Then, open the custom code editor in event setting window and input the custom code for selected widget.

14 Revision history

[Table 56](#) summarizes the changes done to this document since the initial release.

Table 56. Revision history

Revision number	Date	Substantive changes
0	29 October 2020	Initial release
1	17 November 2020	Updated Work with MCUXpresso IDE

Table 56. Revision history...continued

Revision number	Date	Substantive changes
2	11 January 2021	Added and updated multiple sections
3	10 May 2021	Added and updated multiple sections for v1.1
4	30 July 2021	Added and updated multiple sections for v1.2
5	29 September 2021	Updated multiple sections for v1.2.1
6	07 January 2022	Updated multiple sections for v1.3.0
7	31 March 2022	Updated multiple sections for v1.3.1
8	25 July 2022	Updated multiple sections for v1.4.0
9	23 September 2022	Updated multiple sections for v1.4.1
10	10 January 2023	Added and updated multiple sections for v1.5.0
11	31 March 2023	Updated multiple sections for v1.5.1

15 Legal information

15.1 Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

15.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

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.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at <http://www.nxp.com/profile/terms>, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this data sheet expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. 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.

15.3 Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile — are trademarks and/or registered trademarks of Arm Limited (or its subsidiaries or affiliates) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved.

i.MX — is a trademark of NXP B.V.

Oracle and Java — are registered trademarks of Oracle and/or its affiliates.

Contents

1	Introduction	2	4.19	Image	72
1.1	Supported features	2	4.20	Image button	73
1.2	Hardware requirement of LVGL application	5	4.21	Label	74
2	Installation	5	4.22	LED	75
2.1	Prerequisites of Windows 10	5	4.23	Line	76
2.2	Install GUI Guider on Windows 10	6	4.24	Line meter	76
2.3	Set up runtime environment on Windows 10	7	4.25	List	78
2.3.1	Install Java Runtime Environment	7	4.26	Lottie	80
2.4	Uninstall GUI Guider on Windows 10	8	4.27	Menu	80
2.5	Prerequisites of Ubuntu 22.04	8	4.28	Message box	83
2.6	Install GUI Guider on Ubuntu 22.04	8	4.29	QR code	86
2.7	Uninstall GUI Guider on Ubuntu 22.04	8	4.30	Radio button	86
2.8	Prerequisites of MacOS	8	4.31	Meter	88
2.9	Install GUI Guider on MacOS	8	4.32	Roller	89
2.10	Uninstall GUI Guider on MacOS	10	4.33	Slider	91
3	Usage	10	4.34	Spangroup	92
3.1	Create a project	10	4.35	Spinbox	94
3.2	Open a project	16	4.36	Spinner	95
3.3	Import a local project	18	4.37	Switch	97
3.4	Delete a project	19	4.38	Table	98
3.5	Generated code	20	4.39	Tab view	100
3.6	Export Code	21	4.40	Text area	102
3.7	Export project	22	4.41	Text progress bar	104
3.8	Import fonts	23	4.42	Tileview	104
3.9	Generate fonts	23	4.43	Video	106
3.10	Code view	25	4.44	Window	107
3.11	Run simulator	25	5	Event details	109
3.12	Run target	27	6	LVGL hardware acceleration	111
3.13	Tileview usage	29	6.1	PXP enablement	111
3.14	GUI auto-scaling	31	6.2	VGLite enablement	112
3.15	Event window	33	6.3	Recommendations to improve acceleration	113
3.16	Resource window	34	6.3.1	General recommendations	114
3.17	Import resource	35	6.3.2	VGLite recommendations	114
3.18	Shortcut function	36	7	Performance	114
3.19	Project setting	37	7.1	Performance monitor enablement	115
3.20	IDE setting	39	7.2	Tips to improve the performance	116
3.21	Screen setting	41	7.3	Improve the performance for i.MX RT boards	118
4	Widget details	43	7.3.1	Prerequisites	118
4.1	3Dimg	43	7.3.2	Improve the performance	118
4.2	Analog clock	44	8	Debug GUI Guider project	118
4.3	Animation image	45	8.1	MCUXpresso	118
4.4	Arc	46	8.2	Keil MDK	120
4.5	Bar	48	8.3	IAR	121
4.6	Button	49	9	MicroPython	122
4.7	Button matrix	50	9.1	Using LVGL in MicroPython	122
4.8	Calendar	52	9.1.1	Advantages	122
4.9	Canvas	57	9.2	MicroPython in GUI Guider	123
4.10	Carousel	58	9.2.1	Generate code	123
4.11	Chart	60	9.2.2	Run simulator	123
4.12	Checkbox	62	9.2.3	Add custom code	123
4.13	Color picker	63	9.2.3.1	As event action	123
4.14	Container	64	9.2.3.2	As custom.py	124
4.15	Date text box	65	9.2.4	Limitations	125
4.16	Digital clock	66	10	Porting RTOS	125
4.17	Drop-down list	67	10.1	RT-Thread	125
4.18	Gauge	69			

10.1.1	Prerequisite	125
10.1.2	Install Git	125
10.1.3	Configure the Env tool	126
10.1.4	Download RT-Thread and apply patches	127
10.1.5	Enable GUI demo project in RT-Thread	128
10.1.6	Export source of GUI designed by GUI Guider	129
10.1.7	Build and compile	130
10.1.8	Known Issues	131
10.2	Zephyr	131
10.2.1	Set up Zephyr build environment	131
10.2.2	Get Zephyr and install Python dependencies	132
10.2.3	Install Zephyr SDK	132
10.2.4	Design GUI and export code by GUI Guider ..	133
10.2.5	Build and deploy Zephyr image	134
11	External storage	134
11.1	Supported devices	134
11.2	Video	134
11.2.1	Prerequisites	134
11.2.2	Prepare h264 file	135
11.2.3	Create project	135
11.2.4	Customize project	138
11.3	Image	138
11.3.1	Prerequisites	138
11.3.2	Project template	138
11.3.3	Customized project	141
12	External image in QSPI Flash	141
12.1	Prerequisites	141
12.2	Create a new project	141
12.3	Add image widget and set property	144
12.4	Build and deploy	145
12.5	Image Binary in non-XIP flash	147
13	Frequently Asked Questions (FAQs)	148
14	Revision history	149
15	Legal information	151

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.
