

文档信息

信息	内容
关键词	AN13694 , GUI Guider , LVGL , 智能家居 , LPC546xx
摘要	本应用笔记以智能家居为例, 介绍了基于 GUI Guider 的开发流程, 包括 GUI 设计和模拟、代码生成和调试、下载到评估板并在评估板上运行。

1 介绍

近年来，智能家居迅速兴起，发展势头强劲。智能家居将各种家用电器互连起来，提供照明控制、电话遥控、防盗告警和环境监测等服务。智能家居还集成了网络通信、设备自动化和信息化电器。因此，智能家居可以分配资源，在降低能耗和环保方面发挥重要作用。

具有图形界面、低功耗、高性能的嵌入式系统在智能家居领域发挥着举足轻重的作用。恩智浦提供了一款名为 GUI Guider 的完整的 GUI 解决方案，它使用 LVGL 图形库。

本应用笔记以智能家居为例，介绍了基于 GUI Guider 的开发流程，其中包括：

- GUI 设计和模拟
- 代码生成和调试
- 下载到评估板并在评估板上运行

1.1 LVGL

LVGL 是一个开源图形库，它具备创建具有易于使用的图形元素、漂亮的视觉效果和低代码量的嵌入式 GUI 所需的各种要素。

主要特点：

- 开源，在 MIT 许可下免费使用
- 采用 C 语言编写（兼容 c++）并托管在 GitHub 上
- 30 多个功能强大、完全可定制的构件（widget），如按钮、图像按钮、复选框、开关、滑块、标签、圆弧、条形图、线条、画布、图像、滚轮、滑块、仪表、表格、文本区域、动画、日历、图表、列表、菜单、消息框、选项卡视图等
- 所有分辨率的显示器，支持 GPU，支持多显示器
- 支持多种类型的输入设备：
 - 指针式输入设备，如触控板或鼠标
 - 键盘，如普通键盘或简易数字键盘
 - 带左/右转向和前推选项的编码器
 - 分配给屏幕上的特定点的外部硬件按钮
- 绘图功能：
 - 抗锯齿
 - 阴影
 - 直线、圆弧、多边形
 - 遮盖
- 文本功能：
 - 支持 UTF-8
 - 字距调整
 - 自动换行和自动文本滚动
 - 支持阿拉伯语和波斯语
 - 字体压缩
 - 亚像素渲染

- 在线和离线字体转换器
- 自定义字体引擎接口
- FreeType 集成示例
- 多语言支持
- 图像功能：
 - 各种颜色格式：RGB、ARGB、Chroma 键控、索引、仅 alpha
 - 图像实时重新着色
 - 实时缩放和旋转
 - 图像可以存储在闪存或文件中（例如，SD 卡）
 - 在线和离线图像转换器
 - 用于缓存的图像解码器接口
 - PNG 集成示例
- 样式：
 - 层叠样式（如 CSS 中的那样）
 - 重用多个构件中的样式
 - 用于简单更改的本地样式
 - 提供默认外观的主题
 - 状态更改时的转换（动画）
- 支持 MicroPython
- 丰富的演示示例和文档

如需了解详细信息，请参见 [LVGL](#)。

1.2 GUI Guider

GUI Guider 是恩智浦提供的一款用户友好的图形用户界面开发工具，可使用开源 LVGL 图形库快速开发高质量的显示。GUI Guider 配有拖放式编辑器，可轻松使用 LVGL 的许多功能。这些功能包括构件、动画和样式，只需少量编码或无需编码便可创建 GUI。

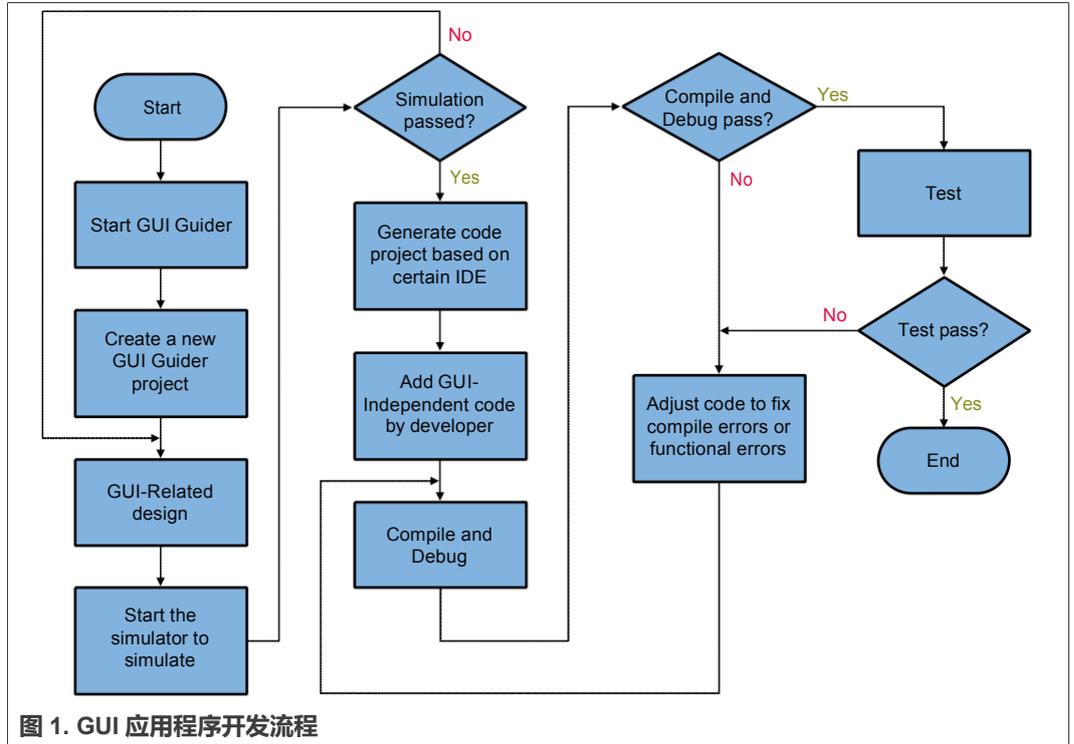
只需点击按钮，您就可以在模拟环境中运行应用程序或将其导出到目标项目中。从 GUI Guider 生成的代码可以轻松添加到 MCUXpresso IDE 或 IAR Embedded Workbench 项目中，从而加快开发过程，并允许您无缝地将嵌入式用户界面添加到应用程序中。

GUI Guider 可以免费使用恩智浦的通用和跨界 MCU，并包含内置项目模板，可用于多个所支持的平台。

如需了解详细信息，请参考 [GUI Guider](#)。

2 GUI 开发流程

总体而言，基于 GUI Guider 的 GUI 开发流程如 [图 1](#) 所示。



如图 1 所示，设计完成后，建议进行一次模拟，验证模拟结果是否符合预期。如果模拟成功且符合预期，则可以生成代码。否则，继续调整设计。

一个建议是将设计分为与 GUI 有关的部分和与 GUI 无关的部分。与 GUI 有关的部分包括各种构件，如按钮、图像、滑块、动画等，以及人机交互，如按钮单击事件。与 GUI 无关的部分是指与特定硬件系统有关的部分，如硬件定时器、板载传感器等。这样可以确保设计能够得到最大程度的验证。

3 智能家居演示介绍

本应用笔记中描述的智能家居演示是一个带 8 个显示界面的 GUI 应用。这些界面包括 5 个主界面和 3 个附属于主界面的子界面。这 8 个界面之间的关系如表 1 所示。这些界面的外观如图 2、图 3 和图 4 所示。

表 1. 主界面与子界面之间的关系

界面 ID	主界面	第 1 级子界面	第 2 级子界面
1	主界面		
2	温度		
3	灯		
4	安全	安全 PIN 输入	安全 PIN 确认

表 1. 主界面与子界面之间的关系 (续)

界面 ID	主界面	第 1 级子界面	第 2 级子界面
5	音频播放器	音频列表	

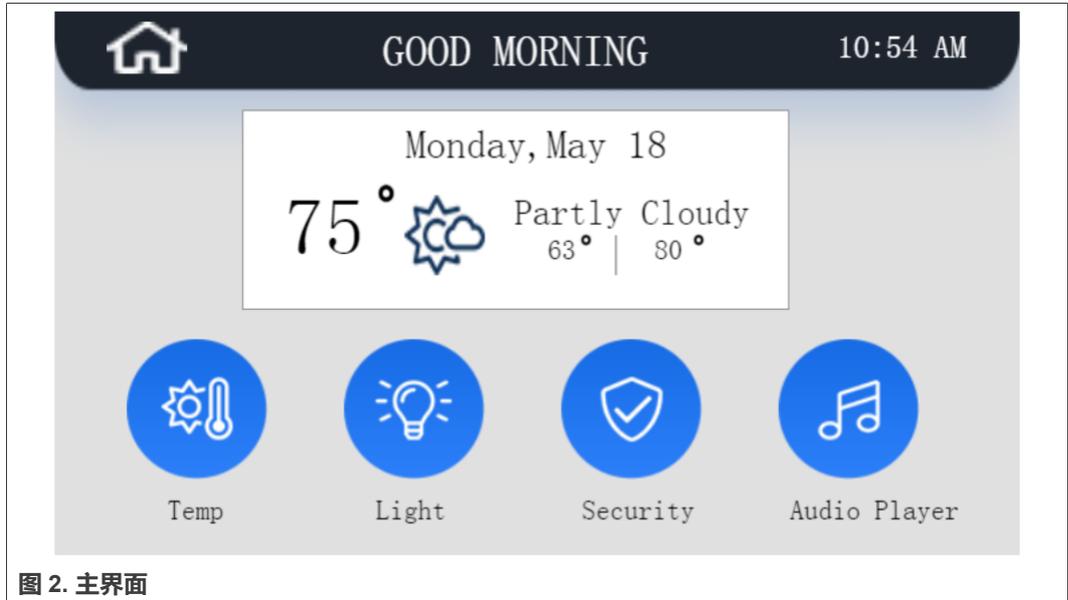


图 2. 主界面

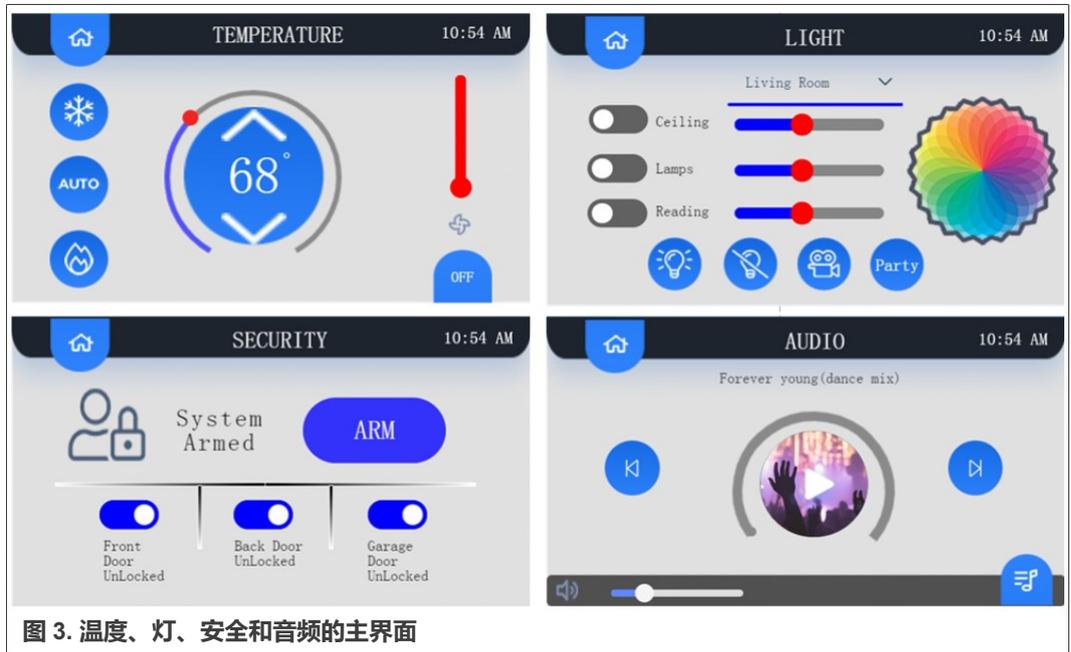


图 3. 温度、灯、安全和音频的主界面



注：此处显示的日期、时间和天气情况是无效的，因为它们依赖于设备的硬件。在嵌入式系统中，日期和时间可以通过 RTC 来控制。此 GUI 演示中的其他显示界面也是如此。

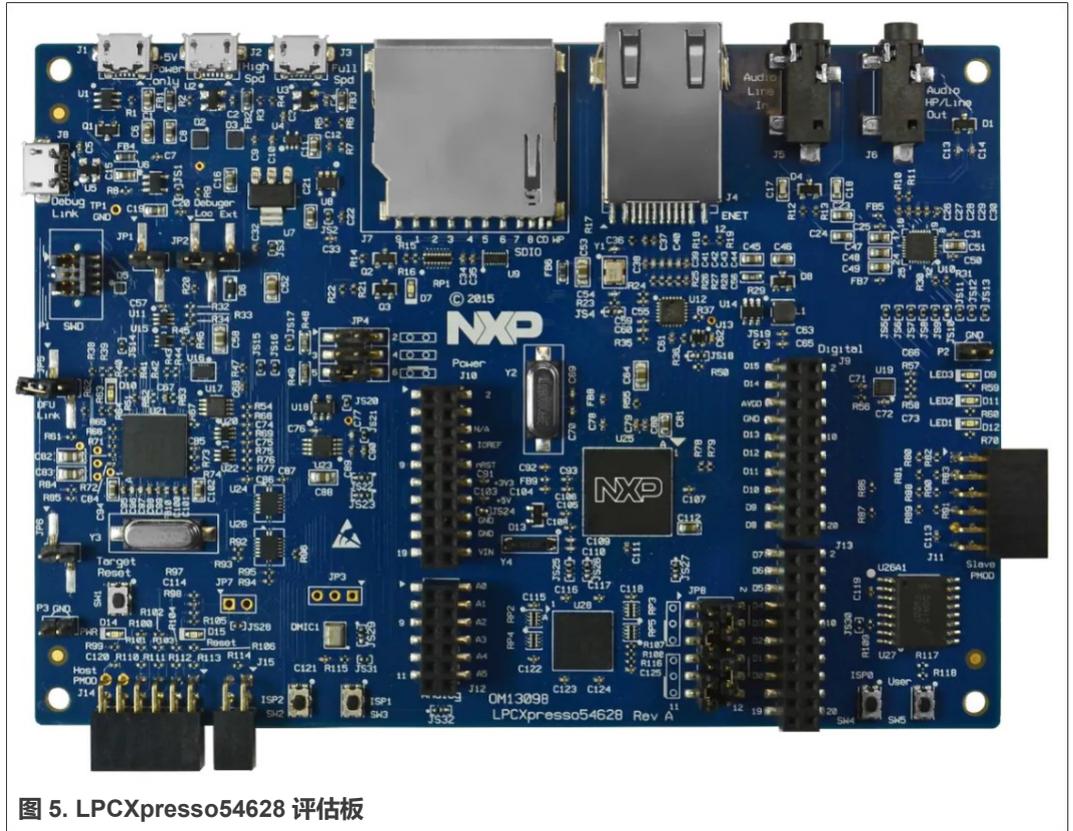
4 开发环境

本章介绍此智能家居演示的软件和硬件开发环境。

4.1 硬件环境

智能家居演示的硬件开发环境由两部分组成：安装了 GUI Guider 工具的 PC 机和配备了 LPC54628 的评估板-LPCXpresso54628 (OM13098)。

LPCXpresso54628 是与 LPCXpresso54608 和 LPCXpresso54618 二进制向后兼容的电路板。因此，这个演示可轻松移植到基于 LPC54608 或 LPC54618 的硬件平台，只需做很少的修改。



4.2 软件环境

智能家居演示需要 1.3.1 版的 GUI Guider 来搭建软件环境。1.3.1 版 GUI Guider 支持 7.10.1 和 8.0.2 版的 LVGL。本应用笔记中介绍的演示基于 8.0.2 版 LVGL。

5 智能家居演示的实现

本章介绍智能家居演示实现过程中的关键点。

5.1 新建一个 GUI Guider 项目

要创建 GUI Guider 项目，请执行以下步骤：

1. 双击 **GUI Guider<version>** 图标，启动 GUI Guider。



图 6. 启动 GUI Guider

2. 单击 **Create a new project** (“创建新项目”) 按钮，启动项目创建过程。

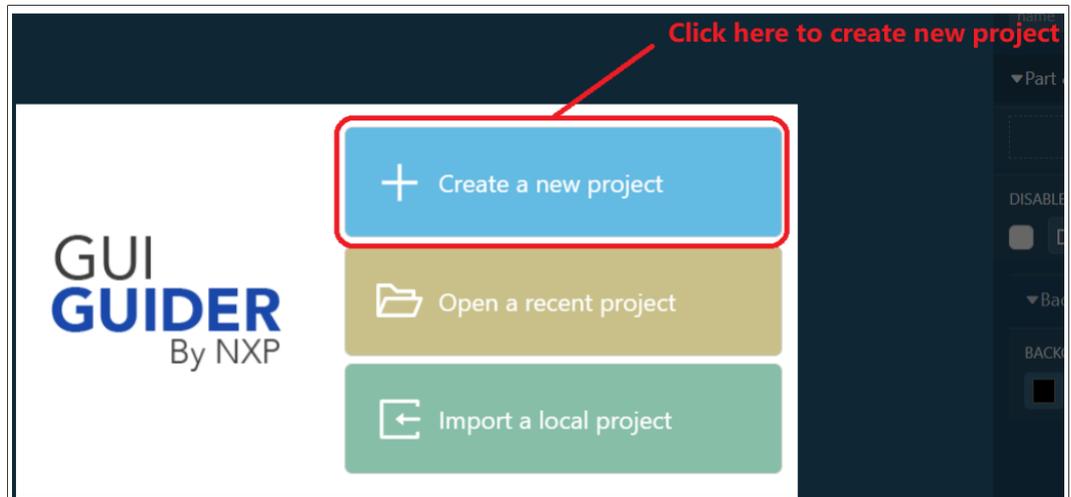


图 7. 创建项目

3. 选择 LVGL 版本为 v8，然后单击 **Next** (下一步) 按钮。



4. 选择 **LPC54628** 作为目标板模板，然后单击 **Next (下一步)** 按钮。



5. 选择空应用模板 **EmptyUI**。

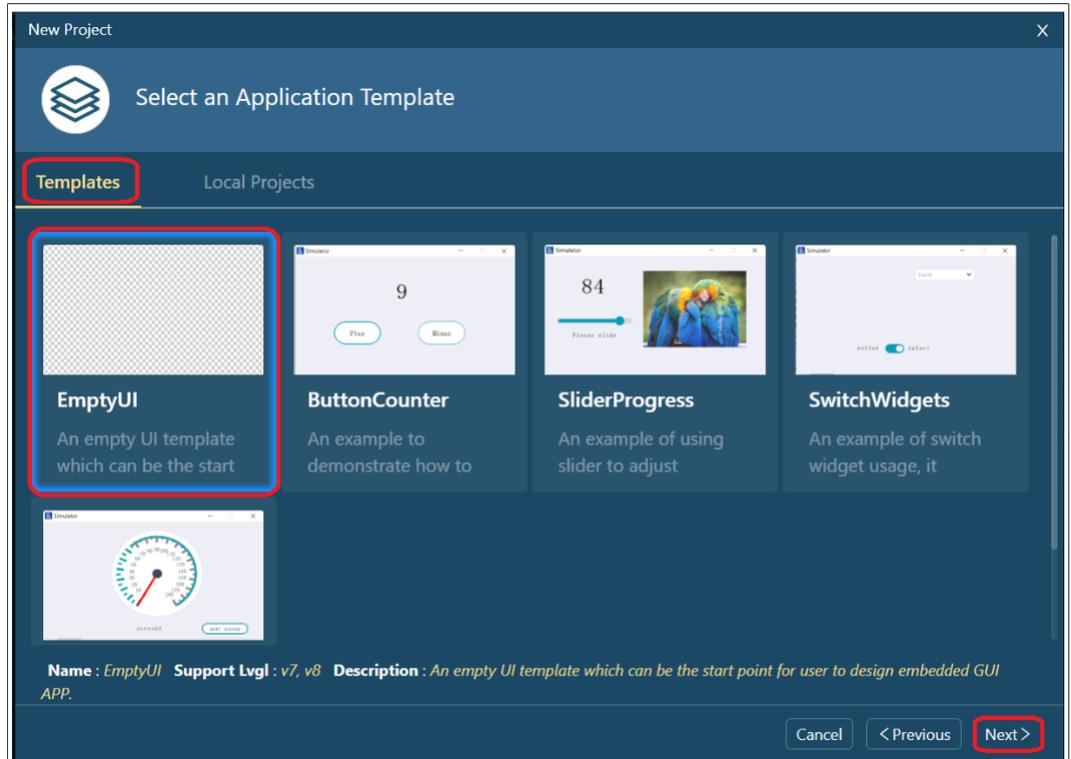


图 10. 选择应用模板

6. 执行项目设置，例如项目名称、项目位置、面板类型和颜色深度。

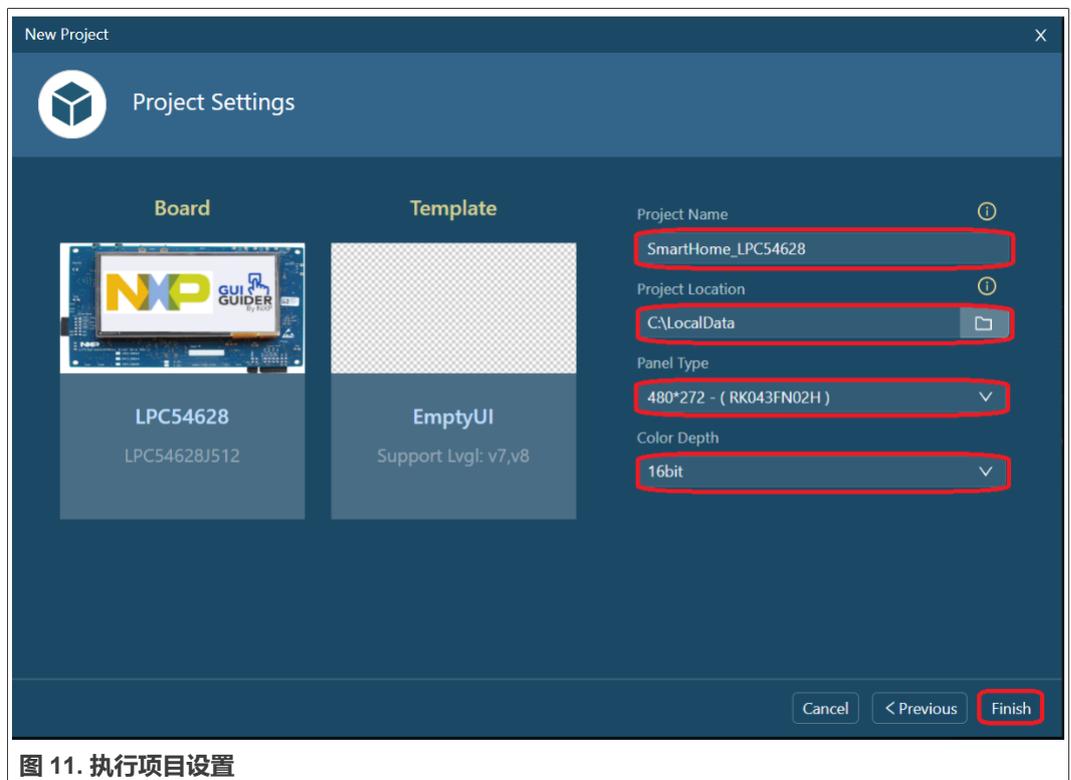


图 11. 执行项目设置

7. 创建的项目如图 12 所示。

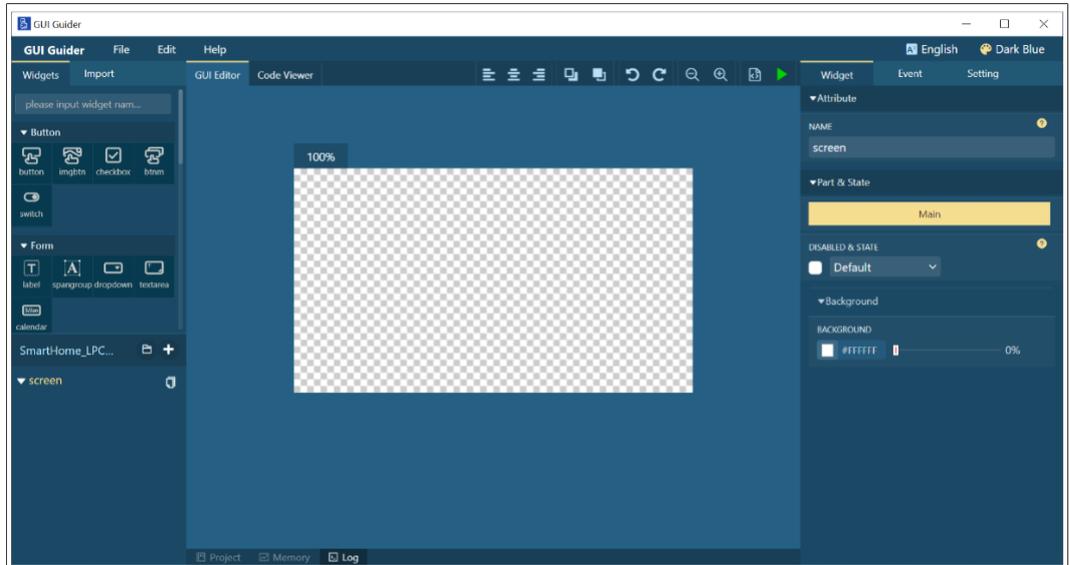


图 12. 已创建项目

5.2 添加 GUI 构件并设置构件属性

GUI Guider 支持通过拖放操作将按钮、图像、图像按钮等构件添加到工作区。为了说明如何向工作区添加构件，可以考虑以添加图像按钮为例。

要将构件添加到工作区，按以下步骤操作：

1. 在 **Widgets** (构件) 选项卡的搜索框中输入 **imgbtn**，然后将该图像按钮拖到工作区，如图 13 所示。

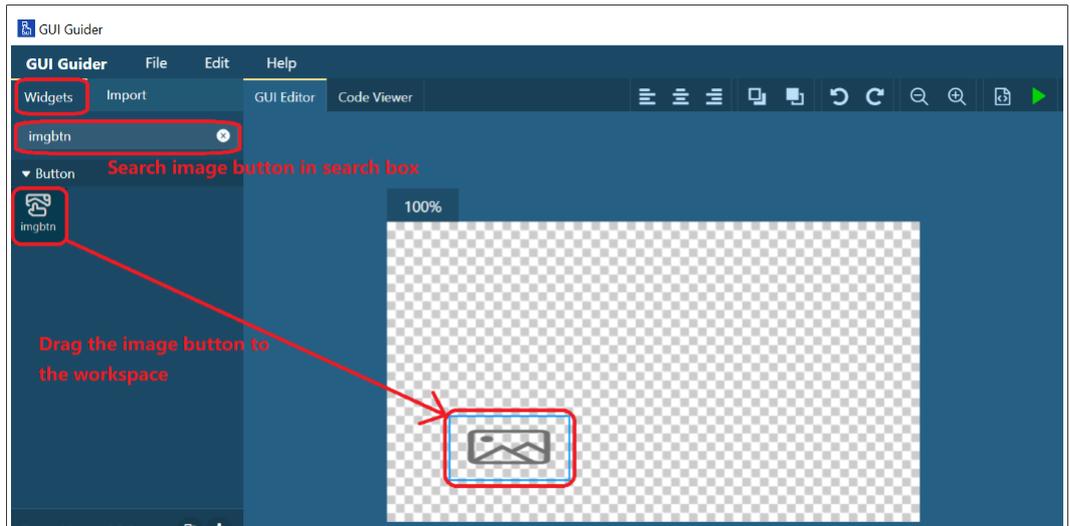


图 13. 将图像按钮添加到工作区

2. 将图像按钮放置在工作区后，设置图像按钮的属性，如图 14 所示。

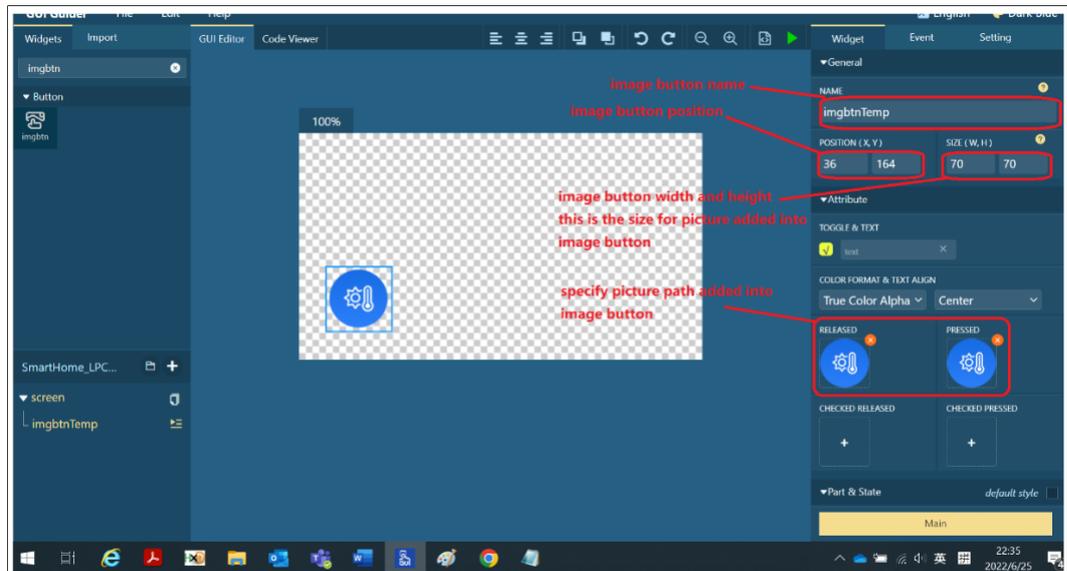


图 14. 设置图像按钮属性

注：对于需要映像路径的构件，在为构件指定映像之前，将所需的图像文件添加到 import（导入）文件夹中，如图 15 和图 16 所示。

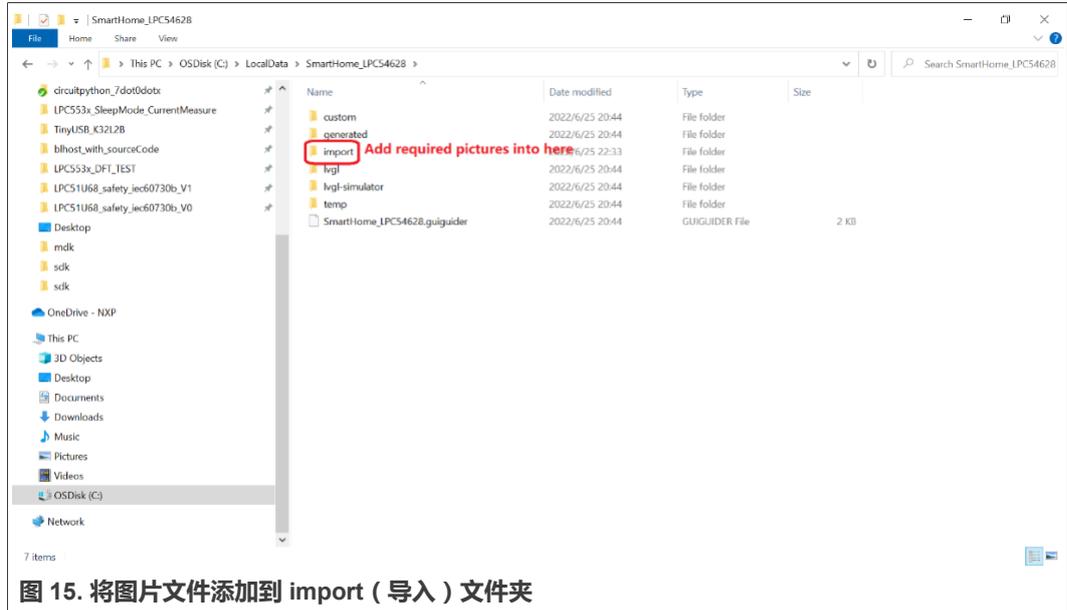


图 15. 将图片文件添加到 import（导入）文件夹

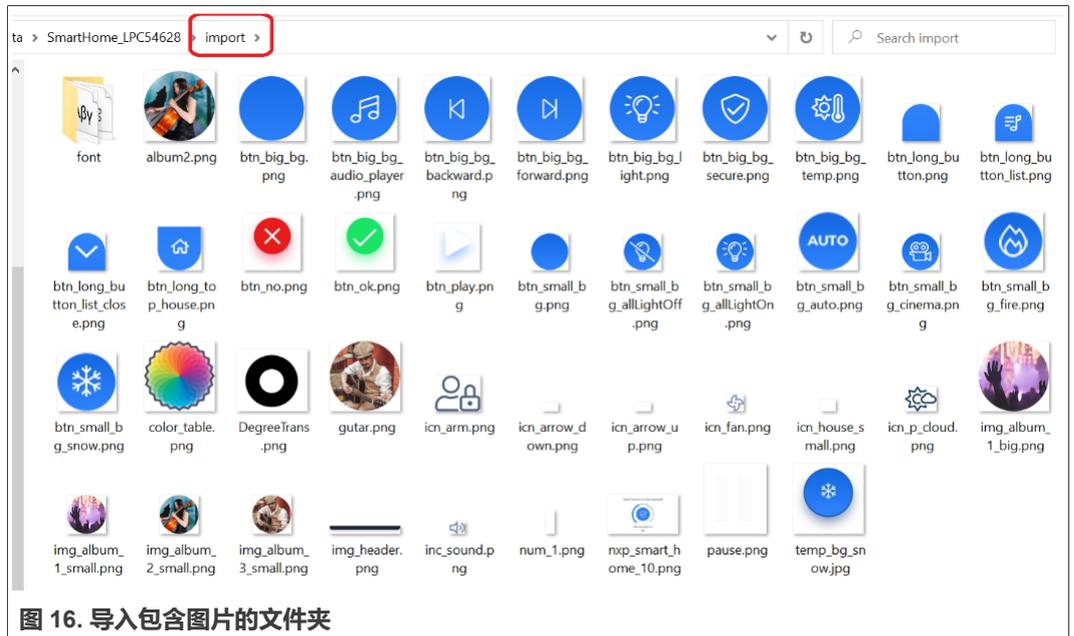


图 16. 导入包含图片的文件夹

5.3 人机交互

在多显示界面的应用中，不同的界面展示不同的功能。因此，需要提供人机交互来执行界面切换或其他功能。可以通过向 GUI 的构件添加事件来实现人机交互。

以单击按钮从一个界面切换到另一个界面为例，介绍如何向构件添加事件。

此智能家居演示的主界面上有一个名为“Temp”的按钮，单击屏幕上的“Temp”按钮，便可切换到“TEMPERATURE”界面，如图 17 所示。

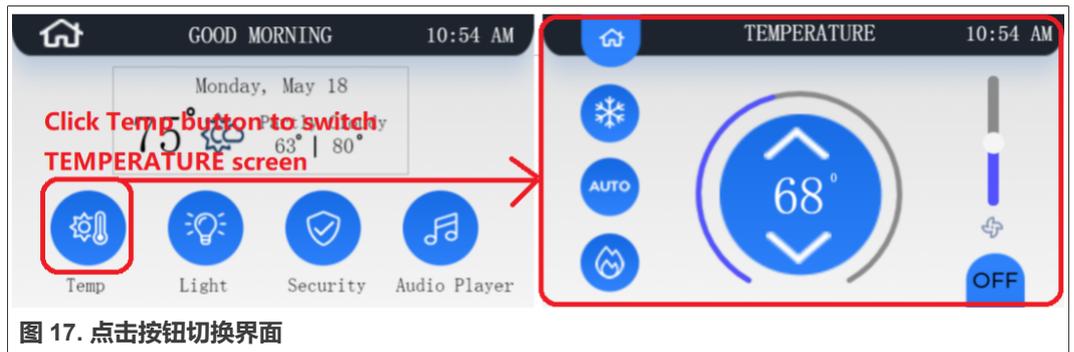


图 17. 点击按钮切换界面

要实现此功能，需要向 Temp 按钮添加事件，如图 18 所示。

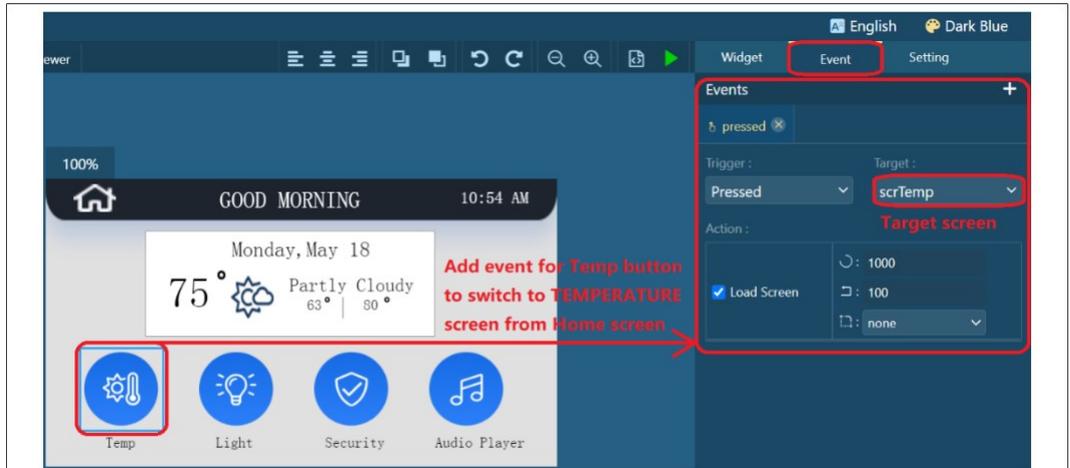


图 18. 向按钮构件添加事件

5.4 模拟

设计完成后，可以启动模拟器，检查模拟结果是否符合预期功能。模拟器的启动如图 19 所示。

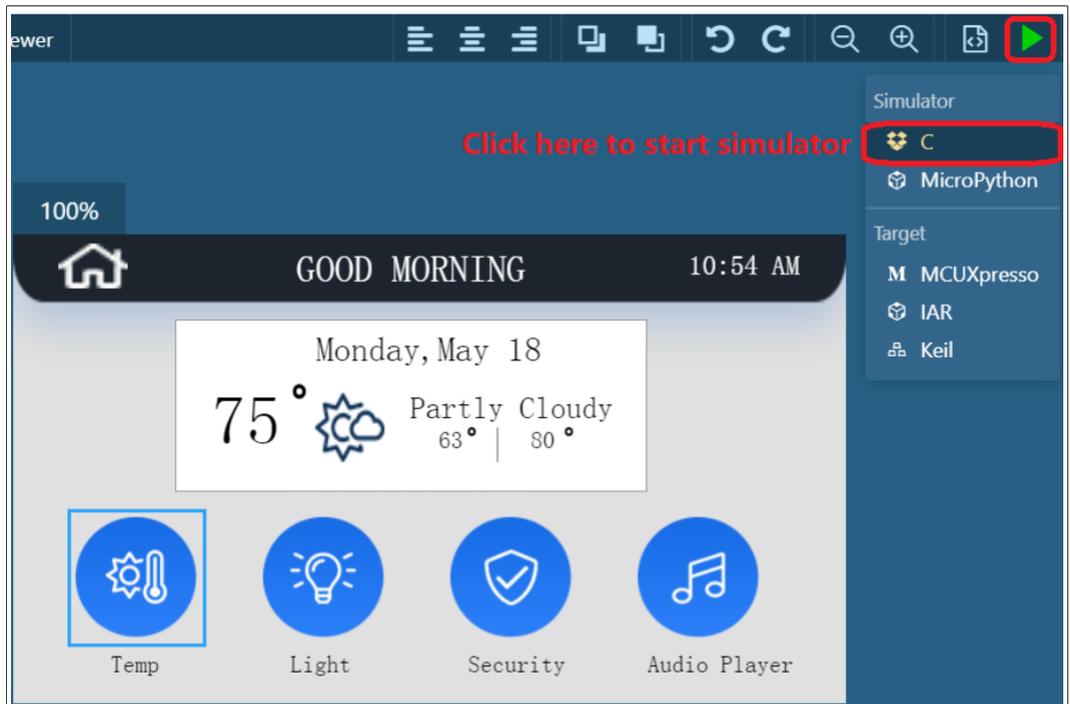


图 19. 启动模拟器

在模拟器中进行智能家居演示的运行结果如图 20 所示。

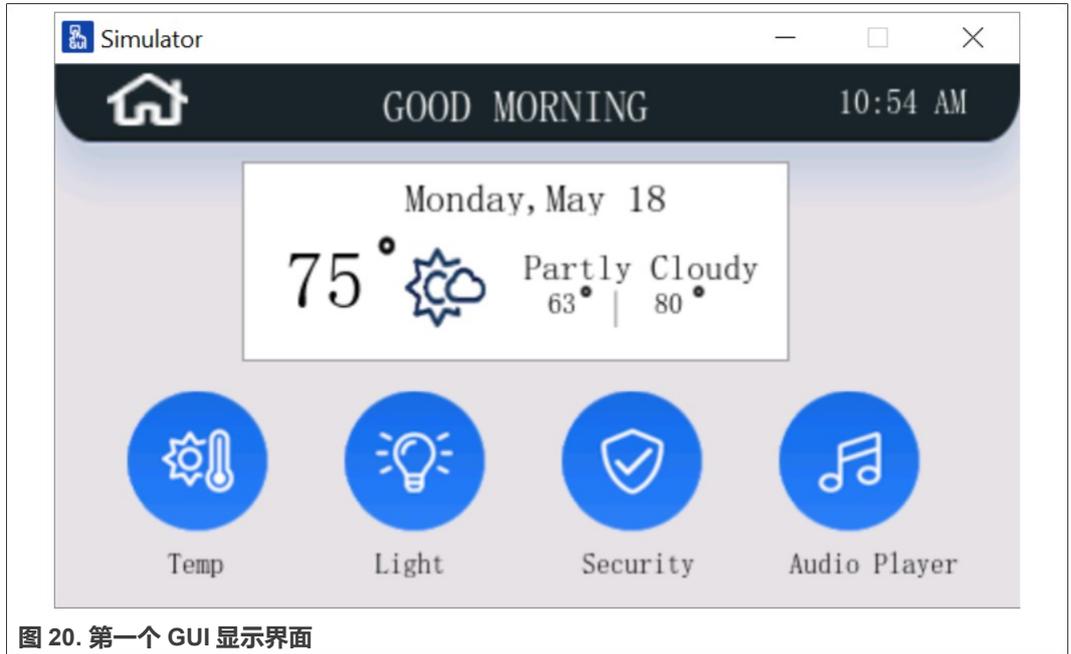


图 20. 第一个 GUI 显示界面

5.5 导出代码项目

成功执行模拟后，可使用 GUI Guider 的代码生成功能导出基于 Keil、IAR 和 MCUXpresso IDE 的智能家居演示代码。

如图 21 所示，单击 **Run > Target Keil** 导出代码。当然，也可以导出基于 IAR 或 MCUXpresso 的项目。

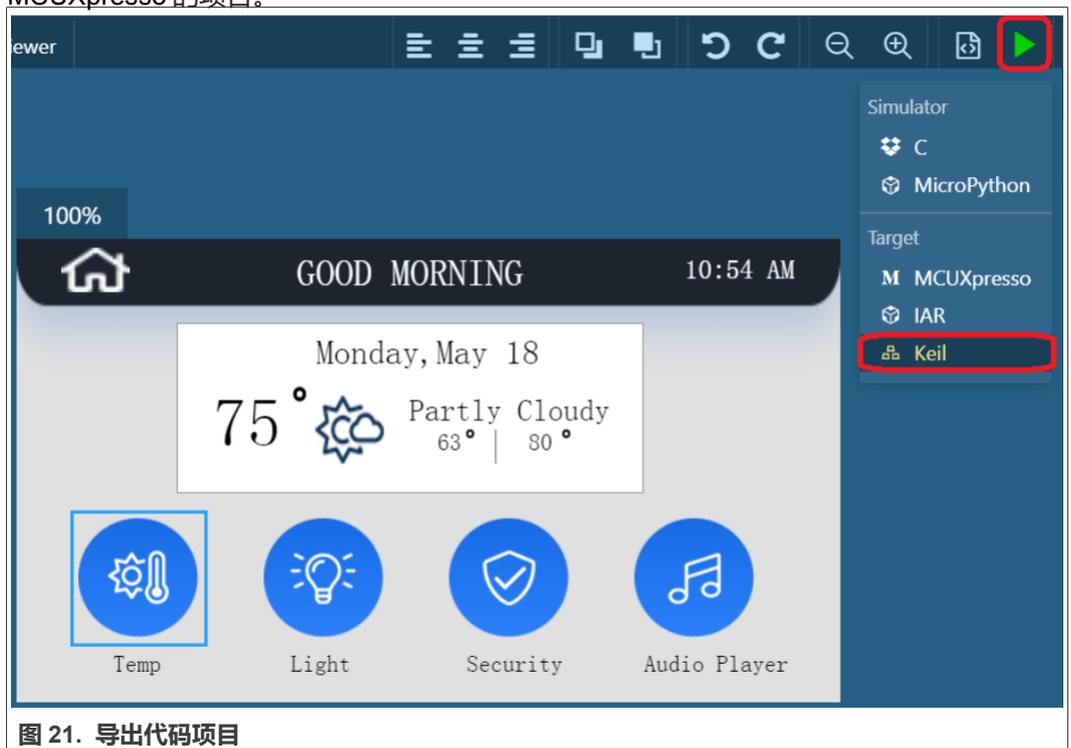


图 21. 导出代码项目

5.6 外部 SPI Flash 的支持

导出智能家居演示的代码后，需要对其进行编译。图 22 展示了编译输出窗口中的编译错误。

```
Build Output
release\lvgl_guider.out: Error: L6406E: No space in execution regions with .ANY selector matching lvgl_support.o(.text.lv_port_pre_init).
release\lvgl_guider.out: Error: L6406E: No space in execution regions with .ANY selector matching custom.o(.text.custom_init).
release\lvgl_guider.out: Error: L6406E: No space in execution regions with .ANY selector matching system_lpc54628.o(.text.SystemInitHook).
release\lvgl_guider.out: Error: L6406E: No space in execution regions with .ANY selector matching events_init.o(.text.events_init).
release\lvgl_guider.out: Error: L6406E: No space in execution regions with .ANY selector matching lv_refr.o(.text.lv_refr_init).
release\lvgl_guider.out: Error: L6406E: No space in execution regions with .ANY selector matching lv_draw_label.o(.rodata.lv_bpp opa_table).
release\lvgl_guider.out: Error: L6406E: No space in execution regions with .ANY selector matching lv_printf.o(.text.out_null).
release\lvgl_guider.out: Error: L6406E: No space in execution regions with .ANY selector matching lv_tlwf.o(.text.default_walker).
release\lvgl_guider.out: Error: L6406E: No space in execution regions with .ANY selector matching lv_textarea.o(.text.pwd_char_hider_anim).
release\lvgl_guider.out: Error: L6406E: No space in execution regions with .ANY selector matching lvgl_guider.o(.text.vApplicationMallocFailedHook).
release\lvgl_guider.out: Error: L6406E: No space in execution regions with .ANY selector matching lvgl_guider.o(.text.vApplicationStackOverflowHook).
release\lvgl_guider.out: Error: L6406E: No space in execution regions with .ANY selector matching setup_scr_scr_securitypin.o(.rodata.str1.1).
release\lvgl_guider.out: Error: L6406E: No space in execution regions with .ANY selector matching lv_tabView.o(.rodata.str1.1).
release\lvgl_guider.out: Error: L6406E: No space in execution regions with .ANY selector matching lv_fa.o(.rodata.str1.1).
release\lvgl_guider.out: Error: L6406E: No space in execution regions with .ANY selector matching lv_textarea.o(.rodata.str1.1).
release\lvgl_guider.out: Error: L6407E: Sections of aggregate size 0x6b530 bytes could not fit into .ANY selector(s).
Not enough information to list image symbols.
Not enough information to list load addresses in the image map.
Finished: 2 information, 1 warning and 1902 error messages.
"release\lvgl_guider.out" - 1902 Error(s), 4 Warning(s).
Target not created.
Build Time Elapsed: 00:00:29
```

图 22. 无空间编译错误

当 1.3.1 版本的 GUI Guider 导出代码时，应用程序的 **Code** 段和 **RO** 段（即图片数据）默认存储在片上闪存中。由于 LPC54628 的片上闪存只有 512 kB，而所有图片数据有 800 kB 左右，不能将所有图像数据存储在片上闪存中。因此，在编译代码时会产生一些无空间编译错误。

LPCXpresso54628 评估板具有板载 SPI Flash，因此我们可以将图片数据映射到 SPI Flash。

要支持 SPI Flash，请执行以下步骤：

1. 为连接到 SPI Flash 的引脚设置引脚复用功能。相应的代码位于 `pin_mux.c` 中。
2. 将包含 `fsl_spifi.c` 和 `fsl_spifi.h` 的 SPIFI 驱动程序添加到代码项目中。

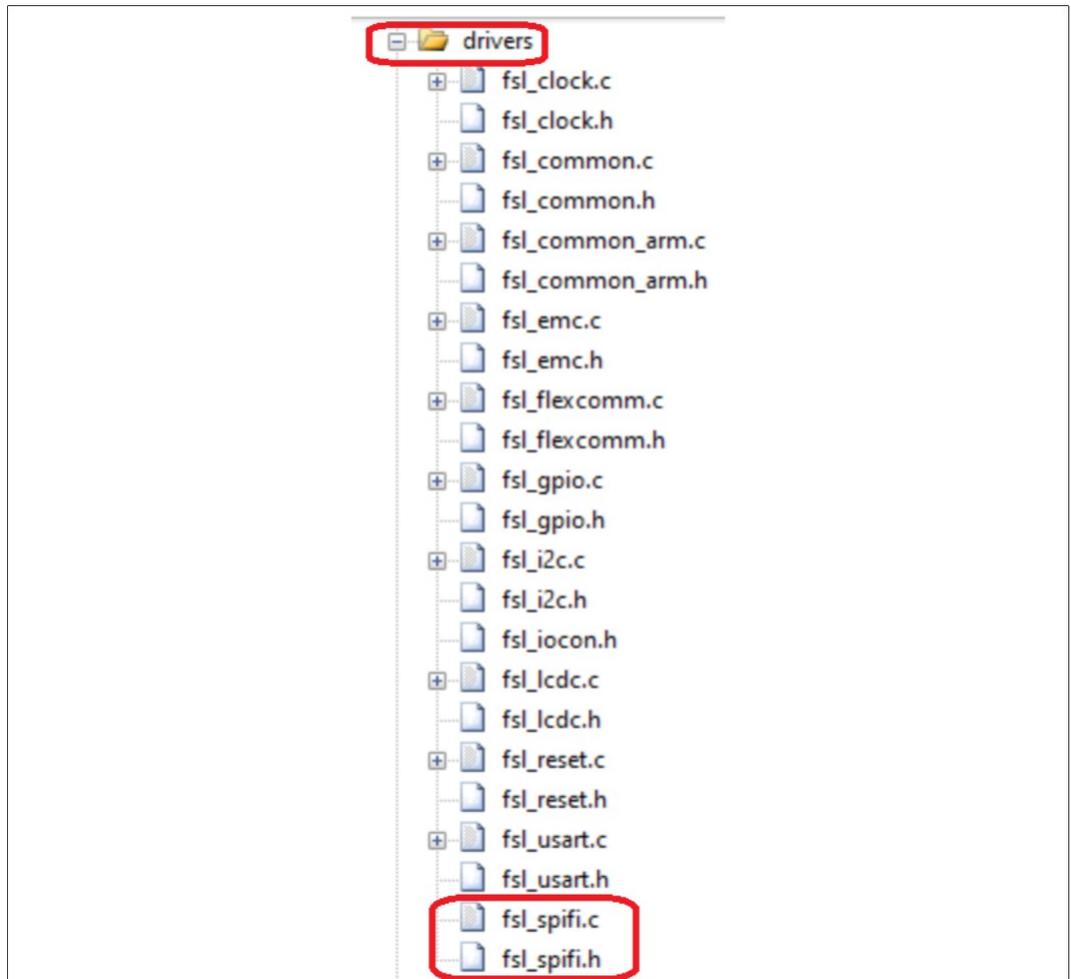


图 23. 添加 SPIFI 驱动程序文件

3. 将 SPI Flash 驱动程序添加到代码中。

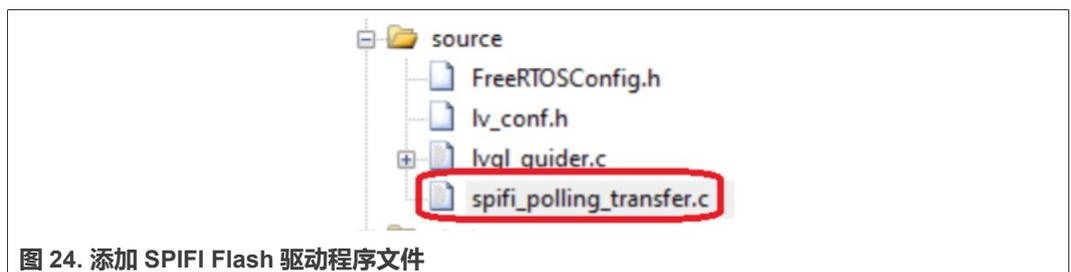


图 24. 添加 SPIFI Flash 驱动程序文件

4. lvgl_guider.c 中的主函数调用 SPI Flash 初始化函数，该函数名为 spifi_flash_init，如图 25 所示。

```
int main(void)
{
    BaseType_t stat;

    /* Init board hardware. */
    /* attach 12 MHz clock to FLEXCOMMO (debug console) */
    CLOCK_AttachClk(BOARD_DEBUG_UART_CLK_ATTACH);
    /* Enable the asynchronous bridge */
    SYSCON->ASYNCAPCTRL = 1;
    /* Use 12 MHz clock for some of the Ctimers */
    CLOCK_AttachClk(kFRO12M_to_ASYNC_APB);

    BOARD_InitPins();
    BOARD_BootClockPLL220M();
    BOARD_InitDebugConsole();
    BOARD_InitSDRAM();
    spifi_flash_init();

    PRINTF("This is Smart home demo for LPC54608!\r\n");
    stat = xTaskCreate(AppTask, "lvgl", configMINIMAL_STACK_SIZE, NULL, 1, &stat);

    if (pdPASS != stat)
    {
        PRINTF("Failed to create lvgl task");
        while (1)
        {
            ;
        }
    }
}
```

图 25. 调用 SPI Flash 初始化函数

5. 修改 scatter 文件。为 SPI flash 提供地址空间，并将图片文件映射到 SPI flash 对应的地址空间。scatter 文件是 LPC54628J512_flash.scf，存储路径为 deploy\Keil\sdk\LPC54628\arm。

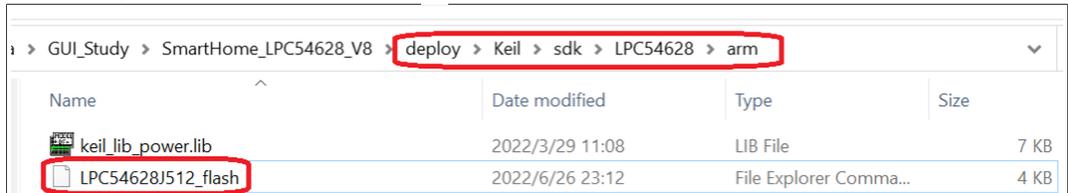


图 26. scatter 文件的位置

```
#define m_interrupts_start 0x00000000
#define m_interrupts_size 0x00000400

#define m_text_start 0x00000400
#define m_text_size 0x0007FC00

#define m_text_spifi_start 0x10000000
#define m_text_spifi_size 0x10000000

#define m_data_start 0x20000000
#define m_data_size 0x00028000

#define m_usb_sram_start 0x40100000
#define m_usb_sram_size 0x00002000
```

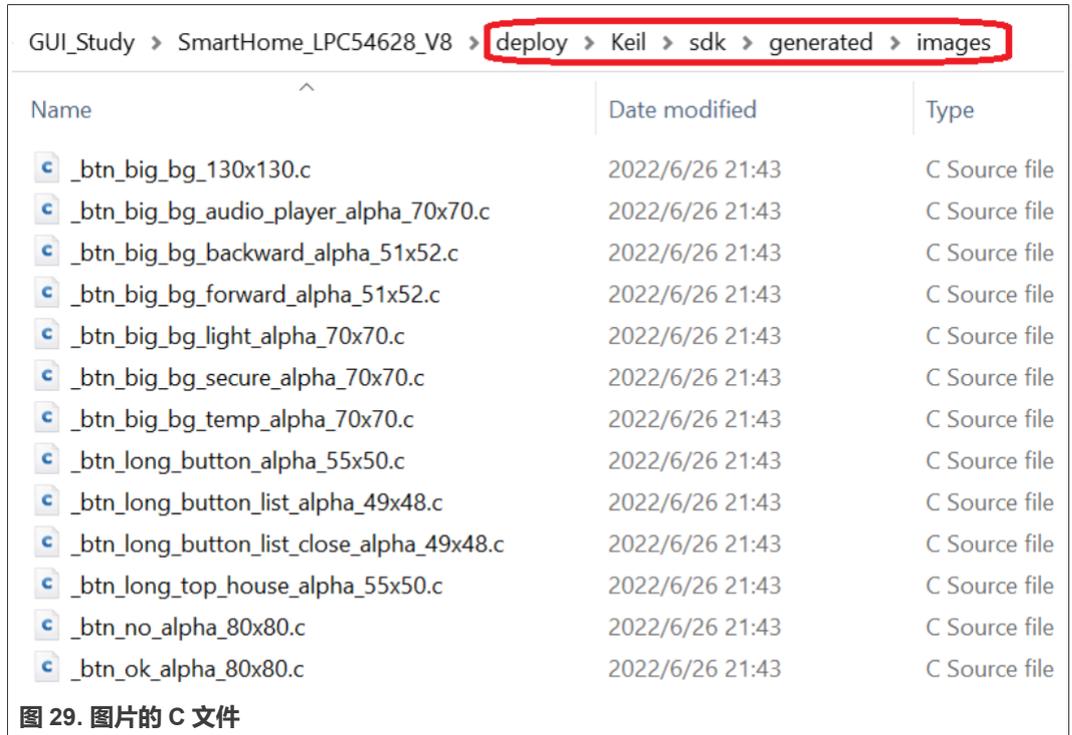
图 27. SPI Flash 地址空间

图 28 展示了映射到 SPI flash 地址空间的扩展名为.o 的文件。

```
LR_m_spifi_text m_text_spifi_start m_text_spifi_size {
ER_m_spifi_text m_text_spifi_start m_text_spifi_size {
  _btn_big_bg_130x130.o
  _btn_big_bg_audio_player_alpha_70x70.o
  _btn_big_bg_backward_alpha_51x52.o
  _btn_big_bg_forward_alpha_51x52.o
  _btn_big_bg_light_alpha_70x70.o
  _btn_big_bg_secure_alpha_70x70.o
  _btn_big_bg_temp_alpha_70x70.o
  _btn_long_button_alpha_55x50.o
  _btn_long_button_list_alpha_49x48.o
  _btn_long_button_list_close_alpha_49x48.o
  _btn_long_top_house_alpha_55x50.o
  _btn_no_alpha_80x80.o
  _btn_ok_alpha_80x80.o
  _btn_play_alpha_48x48.o
  _btn_pause_alpha_48x48.o
  _btn_small_bg_allLightOff_alpha_50x50.o
  _btn_small_bg_allLightOn_alpha_50x50.o
  _btn_small_bg_alpha_50x50.o
  _btn_small_bg_auto_alpha_55x55.o
  _btn_small_bg_cinema_alpha_50x50.o
  _btn_small_bg_fire_alpha_55x55.o
  _btn_small_bg_snow_alpha_55x55.o
  _color_table_alpha_140x140
  _DegreeTrans_6x6.o
  _DegreeTrans_10x10.o
  _icn_arm_72x68.o
  _icn_arrow_down_alpha_60x30.o
  _icn_arrow_up_alpha_60x30.o
  _icn_fan_20x20.o
  _icn_house_small_40x26.o
  _icn_p_cloud_40x40.o
  _img_album_1_big_100x100.o
  _img_album_1_small_alpha_40x40.o
  _img_album_2_big_100x100.o
  _img_album_2_small_alpha_40x40.o
  _img_album_3_big_100x100.o
  _img_album_3_small_alpha_40x40.o
  _img_header_480x80.o
  _inc_sound_20x20.o
  _nxp_smart_home_10_480x272.o
}
}
```

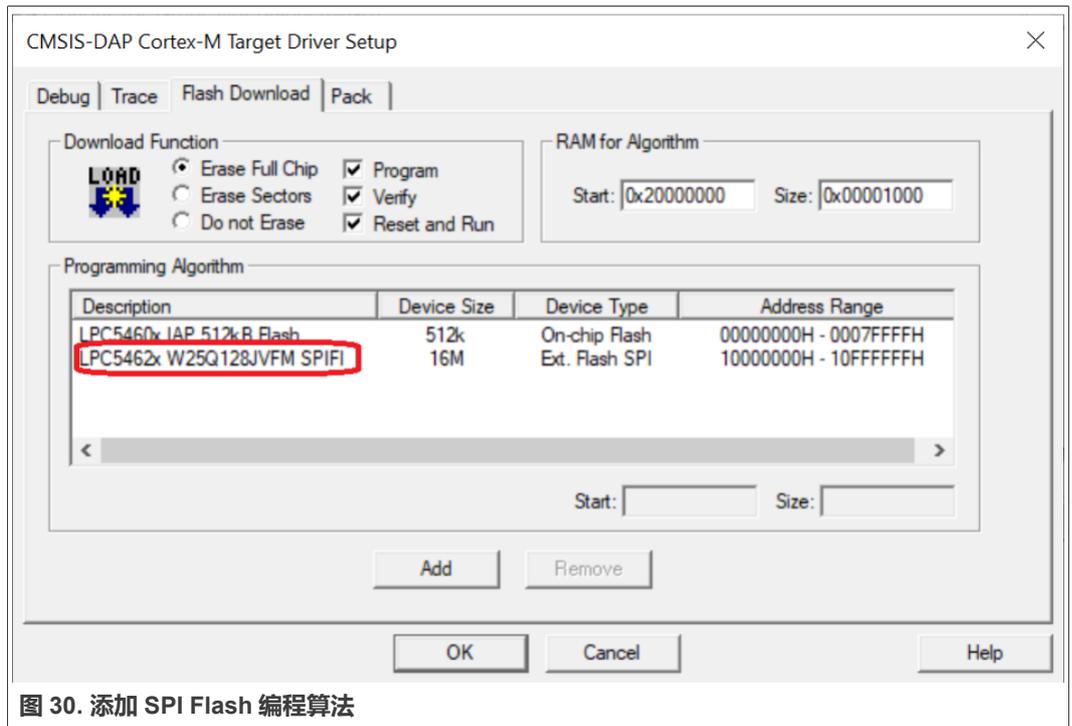
图 28. 将图片对象文件映射到 SPI Flash 的地址空间

这些文件是由位于 `deploy\Keil\SDK\generated\images` 中的图片的 C 文件编译而来的 objective 文件，如图 29 所示。



5.7 设置 Flash 编程算法

由于 SPI Flash 用于存储图片，必须将 SPI Flash 的编程算法添加到 Keil IDE 中。这样在下载过程开始后，可以擦除 SPI Flash，并且可以将图片数据下载到 SPI Flash。图 30 展示了将 Flash 编程算法添加到 Keil 的方法。



5.8 存储优化

要提高存储器的使用效率，需要采取措施优化存储器的使用。常用的优化措施如下所示：

- 设置 Keil IDE 的 IDE 优化级别，如[图 31](#)所示。

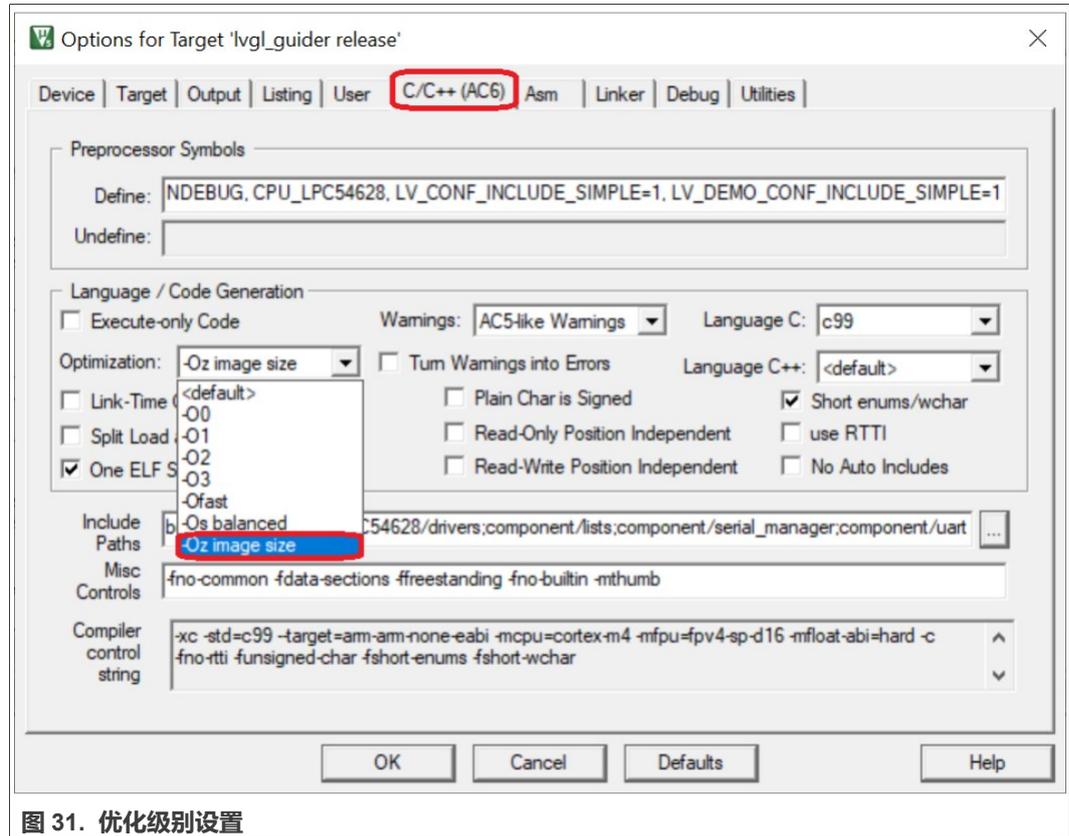


图 31. 优化级别设置

- 使用发布版本编译代码。[图 32](#)展示了 Keil IDE 中编译版本的选择。

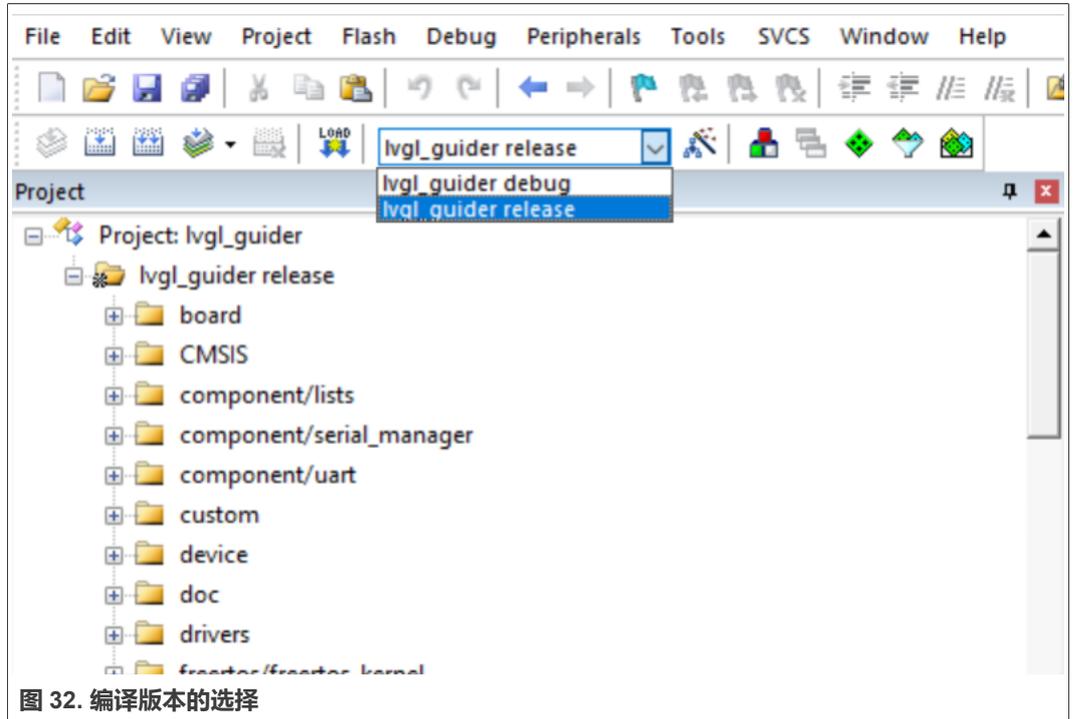


图 32. 编译版本的选择

- 裁剪名为 `lv_conf.h` 的 LVGL 配置文件，请勿编译未使用的构件的代码。例如，智能家居不使用 Spinner 构件。将宏 `LV_USE_SPINNER` 配置为 0，如图 33 所示。

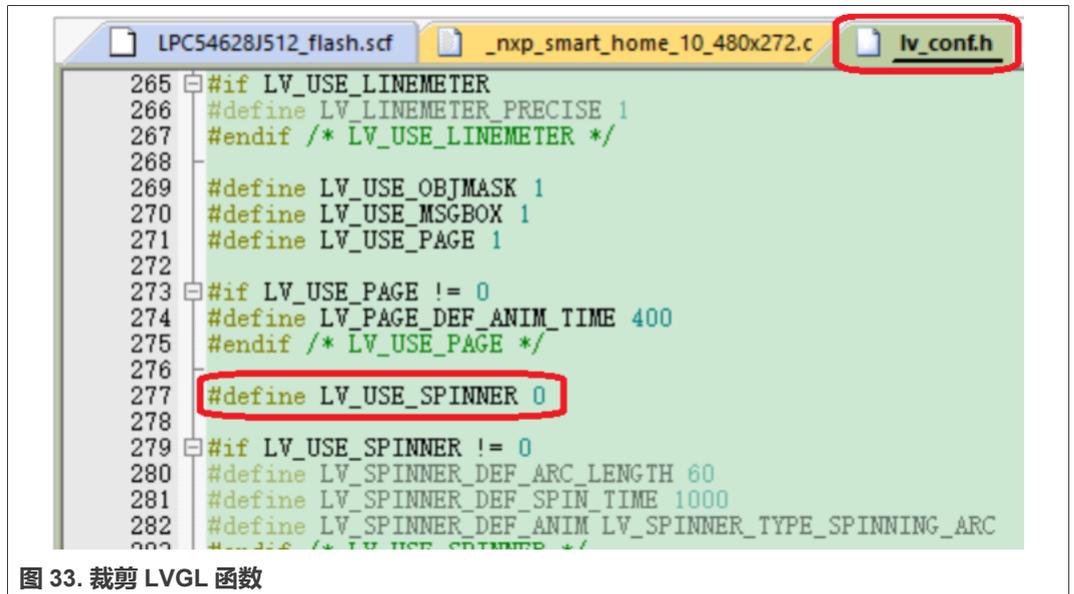


图 33. 裁剪 LVGL 函数

6 演示

本节包括以下演示步骤：

- 下载本应用笔记所附的代码项目。
- 编译代码项目，并将可执行二进制文件下载到 LPC54628 EVK 板。

- 要启动应用，请为 EVK 板重新上电或将其复位。

7 修订历史

表 2. 修订历史

版本号	日期	说明
0	2022 年 7 月 20 日	初版发布

8 Legal information

8.1 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.

8.2 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.

8.3 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.

目录

1	介绍	2
1.1	LVGL	2
1.2	GUI Guider	3
2	GUI 开发流程	3
3	智能家居演示介绍	4
4	开发环境	6
4.1	硬件环境	6
4.2	软件环境	7
5	智能家居演示的实现	7
5.1	新建 GUI Guider 项目	7
5.2	添加 GUI 构件并设置构件属性	11
5.3	人机交互	13
5.4	模拟	14
5.5	导出代码项目	15
5.6	支持外部 SPI Flash	16
5.7	设置 Flash 编程算法	20
5.8	存储优化	21
6	演示	22
7	修订历史	23
8	法律声明	24

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

© 2022 NXP B.V.

All rights reserved.

For more information, please visit: <http://www.nxp.com>

Date of release: 20 July 2022
Document identifier: AN13694