

1 简介

近年来，智能家居有了快速而显著的发展。其目的是使我们的生活更加方便、舒适、高效和安全。智能家居是一个平台系统，可以监测和控制家里的各种设备。嵌入式系统在智能家居应用中起着重要作用，GUI（图形用户界面）也是必不可少的。LVGL（Light and Versatile Graphics Library 轻量级多功能图形库）是用于嵌入式系统的一个很好的GUI库。恩智浦已经推出并在不断改进一个名为GUI Guider的软件工具，以强力支持LVGL GUI的开发。

这篇应用笔记介绍了如何用GUI Guider中的模拟器实现一个典型的智能家居GUI演示工程。读者可以学习GUI Guider上的一些技能，用于GUI的开发，而且所实现的智能家居GUI可以为智能家居应用提供一个很好的参考和基础。

1.1 LVGL

LVGL是一个免费的开源图形库，提供了创建嵌入式GUI所需的一切，具有易于使用的图形元素、漂亮的视觉效果和低内存占用。

主要特点：

- 强大的构建模块，如按钮、图表、列表、滑块和图像
- 带有动画、抗锯齿、不透明、平滑滚动功能的高级图形
- 各种输入设备，如触摸板、鼠标、键盘和编码器等等
- UTF-8编码的多语言支持
- 支持多显示器，例如，同时连接多个TFT及单色显示器
- 完全可定制化的图形元素
- 硬件独立，可与任何微控制器或显示器一起使用
- 可扩展操作且占用很少的内存（64 kB闪存，16 kB RAM）
- 支持操作系统、外部存储器和GPU，但不是必须的
- 即使对于高级图形效果，也能进行单帧缓冲器操作
- 用C语言编写，以获得最大的兼容性（兼容C++）
- 模拟器可无需嵌入式硬件在PC上开始嵌入式GUI设计
- 可绑定MicroPython
- 用于快速GUI设计的教程、例子和主题
- MIT许可下的免费和开源

目录

1	简介	1
2	智能家居GUI演示工程介绍	2
3	智能家居GUI演示工程的实现	3
4	进行演示	9
5	修订历史	9



1.2 GUI Guider

GUI Guider是恩智浦推出的一款用户友好型图形用户界面开发工具，能够利用LVGL开源图形库快速开发高质量的显示。GUI Guider的拖放编辑器可以轻松地利用LVGL的许多功能，如小部件、动画和样式，以最少的编码或根本不需要编码来创建GUI。

只需点击一个按钮，您就可以在模拟环境中运行应用程序或将其导出到目标工程中。GUI Guider生成的代码可以很容易地添加到您的工程中，从而加速开发过程，并允许您无缝地给应用添加嵌入式用户界面。

GUI Guider可免费用于恩智浦的通用和跨界型MCU，并包含针对多个支持平台的内置项目模板。

2 智能家居GUI演示工程介绍

智能家居GUI演示工程是使用GUI Guider的模拟器开发的。因此，GUI演示是独立于MCU硬件的。然而，所生成的GUI相关代码可以直接应用到基于不同MCU芯片的嵌入式系统工程中。通过GUI Guider，模拟演示工程不仅实现了包含小部件的布局，还实现了GUI上的人机互动。在这个GUI演示工程中展示了智能家居的四个功能：温度控制、灯光控制、安全控制和音频播放器。主界面显示了这四个功能和GUI的基本布局设计，如图1所示。

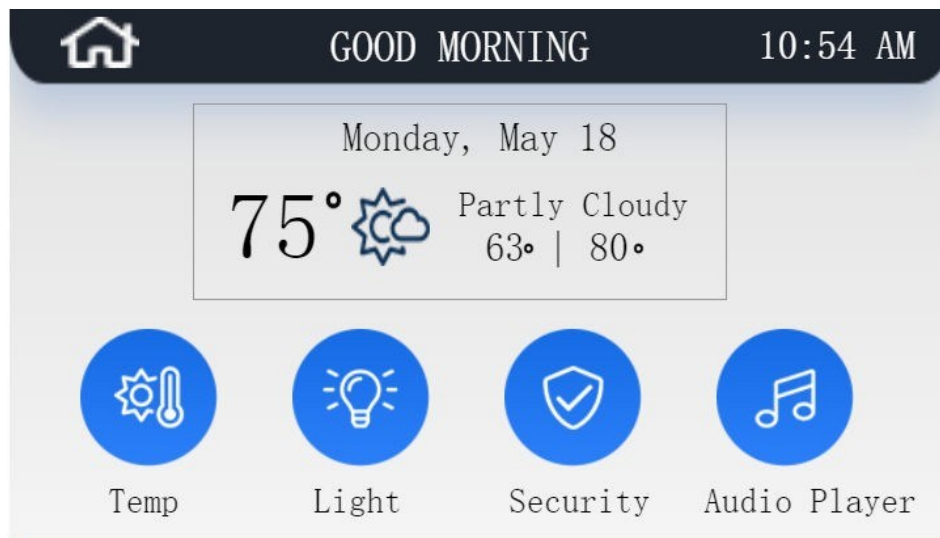


图1. 基于GUI Guider的智能家居GUI演示工程的主屏幕

注意

这里展示的时间、日期和天气是不具备功能的，因为它们取决于设备的硬件，例如日期和时间可以由嵌入式系统的RTC控制。这个GUI演示工程中的其他屏幕也是如此。

关于显示这四个功能的其他主屏幕，请参见图2。



图2. 温度、灯光、安全和音频播放器的主屏幕

3 智能家居GUI演示工程的实现

在GUI Guider上实现GUI布局并不是那么困难。本应用笔记介绍了GUI开发的基本过程和GUI Guider上的几个高级动画功能的实现。在这之前，应该先设置好开发环境。

3.1 开发环境

因为在GUI Guider上使用了模拟器，所以开发环境很简单。

3.1.1 硬件环境

只需要一台PC来设置硬件环境。

3.1.2 软件环境

只需要GUI Guider V1.2.0来设置软件环境。

备注：

- 关于GUI Guider的资源，包括概述、下载、文档、培训等，都可以在恩智浦官方网站的[GUI Guider](#)中看到。
- 阅读GUI Guider用户手册，以便安装GUI Guider并了解其工具。安装后，也可以在这个路径下看到用户手册：GUI Guider\DOCUMENTATION\。
- 使用GUI Guider很容易，本应用笔记就不介绍其使用方法了。

注意

用户可以在[GUI Guider](#)上阅读用户手册或培训材料。

3.2 基于GUI Guider的基本开发过程

开发的基本过程是创建一个GUI工程，开发和运行一个模拟器（包括构建、运行和生成代码）。

创建GUI工程：

1. 定义工程名称和目录。
2. 使用默认设置创建工程。
 - a. 选择一个应用模板为空的模拟器。
 - b. 对于LCD面板。
 - i. 选择颜色深度为16位。
 - ii. 选择分辨率为480*272。

详情请见[图3](#)。



生成GUI代码：

1. 在GUI Guider上开发GUI。
2. 运行模拟器来构建代码并进行测试。
3. 一旦模拟器运行成功，就会生成经过验证的代码。

注意

点击“生成代码”图标，在未能成功构建的情况下，也可以生成代码，但可能会有错误。

GUI Guider工程包中的“custom”和“generated”两个子文件夹都须由开发者添加到他们的应用工程中。开发者创建的代码存储在名为custom的文件夹中，那里有初始的模板文件custom.c、custom.h，而由GUI Guider生成的代码则存储在名为“generated”的文件夹中，那里存储着图像和字体资源。

注意

不要编辑“generated”文件夹中的文件，因为任何手动编辑的内容在执行代码生成时都会被GUI Guider自动改写。用户代码应该添加到“custom”文件夹中。

3.3 高级动画功能的实现

我们知道，大量具有不同风格的小部件对于在GUI中产生漂亮的视觉效果是必要的。动画是获得更好视觉效果的一个非常重要和有帮助的方法。

GUI Guider可以通过对小部件的设置中的“MOVE ANIMATION”项进行简单配置来制作一个小部件的动画。有时，它需要编码来实现一些高级动画功能。GUI Guider的一个功能是，它可以通过编码功能支持小部件的事件设置，因此它允许开发者在GUI Guider上处理事件。这给开发者带来了很大的便利和机会，使他们能够在GUI Guider上实现和观察更多的视觉效果。

下一节将介绍智能家居GUI演示中的两个高级动画。

3.3.1 动画按钮

智能家居GUI演示中的所有按钮都被设计成向下移动一小段距离，然后在它们被释放时上升到原来的位置，从而显示出动画效果。这是由事件设置和添加代码来完成的。关于事件按钮的典型设置，见图4。

1. 在Source Widget中选择当前的按钮。
2. 从触发器列表中选择已释放（Released）。
3. 从目标小部件列表中选择空值（Null）。
4. 从动作列表中选择C代码，因为按钮的动画动作是自定义的，且适用于所有按钮。
5. 在Include field中添加包含“C”代码的文件。
6. 在Code field中输入用户的代码。

作为一个典型的例子，图4显示了主屏幕中标记为温度（Temp）按钮的事件设置，见图1。其功能是：按钮沿Y坐标向下移动10个像素点，当点击的按钮被释放时，再向上移动到原来的位置，然后通过调用函数loadTempScreen()加载温度控制屏幕。

注意

建议添加的代码尽可能简短，例如，使用像btnAnimCustom()这样的API调用，见图4。

这些代码会在运行模拟器时，在generated文件夹中的events_init.c文件中自动生成。根据图4的设置生成的代码见图5。

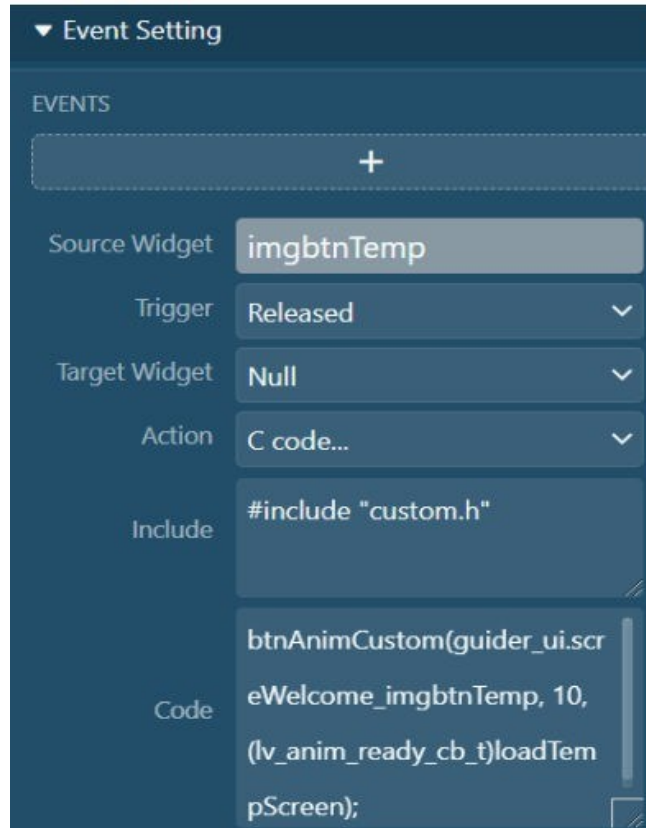


图4. “动画”按钮的事件设置

```

static void screWelcome_imgbtnTempevent_handler(lv_obj_t * obj, lv_event_t event)
{
    switch (event)
    {
        case LV_EVENT_RELEASED:
        {
            btnAnimCustom(guider_ui.screWelcome_imgbtnTemp, 10, (lv_anim_ready_cb_t)loadTempScreen);
        }
        break;
        default:
        break;
    }
}

```

图5. 为“事件”按钮生成的代码

函数btnAnimCustom()在文件custom.c中实现，该函数描述如下：

原型：void btnAnimCustom(lv_obj_t *obj, int delta, lv_anim_ready_cb_t animCb)

功能：将小部件（widget）对象向下移动一段距离，再沿着“Y”坐标向上移动到原始位置，然后调用回调函数。这用于处理动画按钮和事件按钮。

参数：

- obj – 小部件（widget）对象
- delta – 距离
- animcb – 当动画准备好时调用的回调函数，它用于动画之后的事件处理

该功能的实现，见图6。

```
void btnAnimCustom(lv_obj_t *obj, int delta, lv_anim_ready_cb_t animCb)
{
    lv_anim_t a;

    //Initialize an animation variable
    lv_anim_init(&a);
    //Set the function "lv_obj_set_y()" to set the y coordinate in animation
    lv_anim_set_exec_cb(&a, (lv_anim_exec_xcb_t)lv_obj_set_y);
    //Set the object for animation
    lv_anim_set_var(&a, obj);
    //Set the duration of animation to 100ms
    lv_anim_set_time(&a, 100);
    //Get the current y coordinate
    int y = lv_obj_get_y(obj);
    //Set the start and end value of the animation on y coordinate
    lv_anim_set_values(&a, y, y+delta);
    //Set the duration of animation to play back to 100ms
    lv_anim_set_playback_time(&a, 100);
    lv_anim_set_delay(&a, 0);
    if(animCb)
    { //Set the function call when the animation is ready
        lv_anim_set_ready_cb(&a, animCb);
    }
    //Create the animation
    lv_anim_start(&a);
}
```

图6. 为“动画”按钮定制代码

这个API函数用于此演示工程中的所有动画按钮。

3.3.2 图像旋转

在这个智能家居GUI演示工程中，有一个音频播放器的功能。在GUI窗口中，有一个功能是一旦点击播放按钮，专辑的图像就会围绕其中心旋转并播放音乐，见图7。

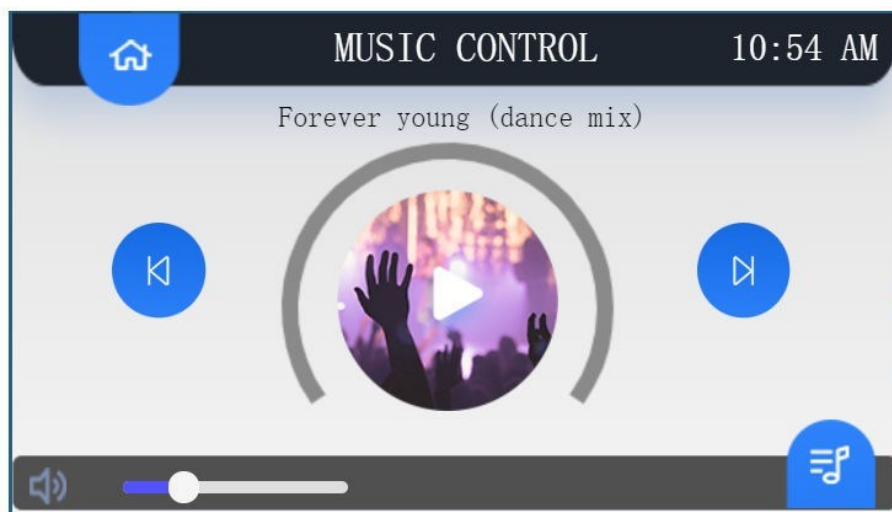


图7. GUI的屏幕截图，音频播放器

如何使图像旋转：这需要对音频播放按钮进行事件设置，因为旋转是通过事件按钮的设置进行配置的。事件设置请见图8，它与图4非常相似。

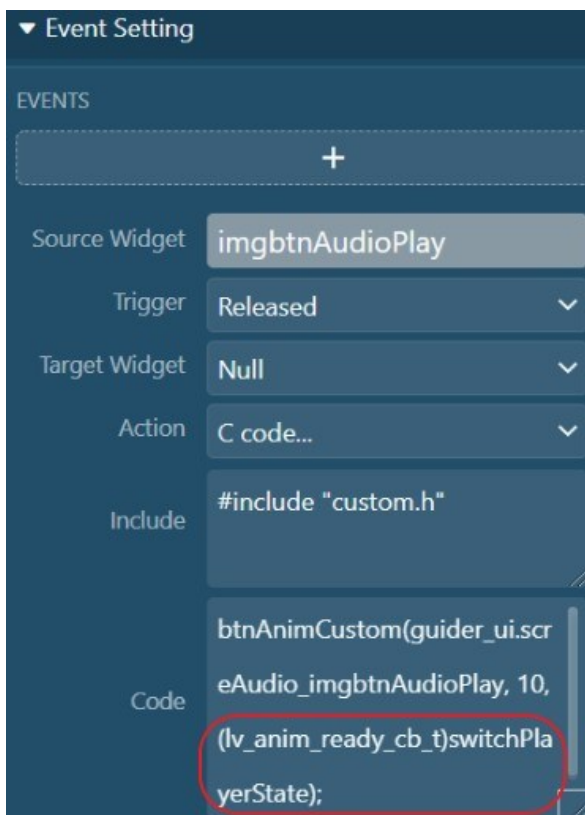


图8. 音频“播放”按钮的事件设置

Code field中的代码函数`btnAnimCustom()`被调用作音频播放按钮动画，回调代码函数`switchPlayerState()`被调用作图像旋转。回调函数是在`custom.c`中实现的自定义代码。

专辑图像的旋转本质上是一个动画，其实现与图6中的按钮动画相同。有两个主要区别：

1. 因为动画是要旋转图像的，`lv_anim_set_exec_cb()` 中为动画设置的代码函数是 `lv_img_set_angle()`。代码函数 `lv_img_set_angle()` 是为了设置图像的旋转角度。在动画中，该函数代码改变图像旋转的角度。
2. `lv_anim_set_values()` 中设置的动画开始和结束值分别为0度和3600度（精度为0.1度）的旋转。

阅读 `custom.c` 文件中的函数 `switchPlayerState()` 和 `initAudioAlbumImgAnim()`，以获得实现的详细信息。

旋转中心：旋转中心是本演示工程中专辑图像小组件的中心，如上节所述。中心的坐标是相对于图像的，可以在小组件的设置中进行配置。关于专辑的图像的相关设置，见图9。ROTATE CENTER（旋转中心）的“X”和“Y”坐标相对于图像尺寸（120×120）被设置为60和60。

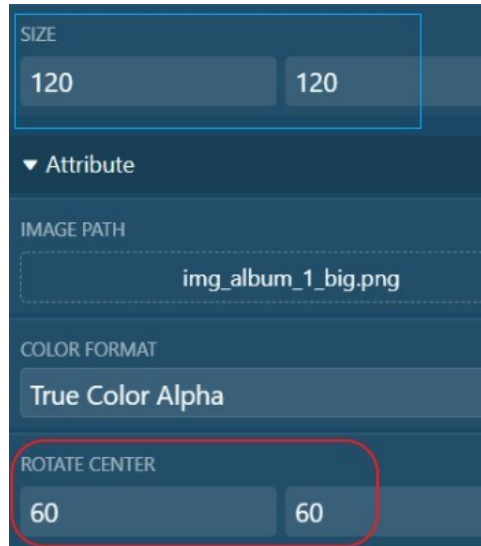


图9. 专辑图像旋转中心的设置

4 进行演示

请按照以下步骤进行演示：

1. 解压缩所附的软件包。
2. 打开GUI Guider，在软件包的根路径下导入“SmartHome_GARC1.guiguider”工程。
3. 点击“运行模拟器”（Run Simulator）来运行演示工程。

随着演示工程的运行，用户可以通过点击GUI窗口上的按钮来尝试体验GUI，操作非常简单。

注意

在 SECURITY 对话框中，输入密码“1234”。

5 修订历史

表1列出了自初次发布以来对该文件所做的修改。

表1. 修订历史

版本号	日期	实质性变更
0	2021年12月15日	初版发布

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

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

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

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

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

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, ICODE, JCOP, LIFE, VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, Altivec, CodeWarrior, ColdFire, ColdFire+, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, Tower, TurboLink, EdgeScale, EdgeLock, eIQ, and Immersive3D are trademarks of NXP B.V. All other product or service names are the property of their respective owners. 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 or registered trademarks of Arm Limited (or its subsidiaries) 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. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org. M, M Mobileye and other Mobileye trademarks or logos appearing herein are trademarks of Mobileye Vision Technologies Ltd. in the United States, the EU and/or other jurisdictions.

© NXP B.V. 2021.

All rights reserved.

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

For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 15 December 2021

Document identifier: AN13450

