

STM32GUI_使用 TouchGFX 二进制字体(Binary Font)功能实现字体动态更新

关键字: TouchGFX, GUI, Binary font, TouchGFX 二进制字体

1. 引言

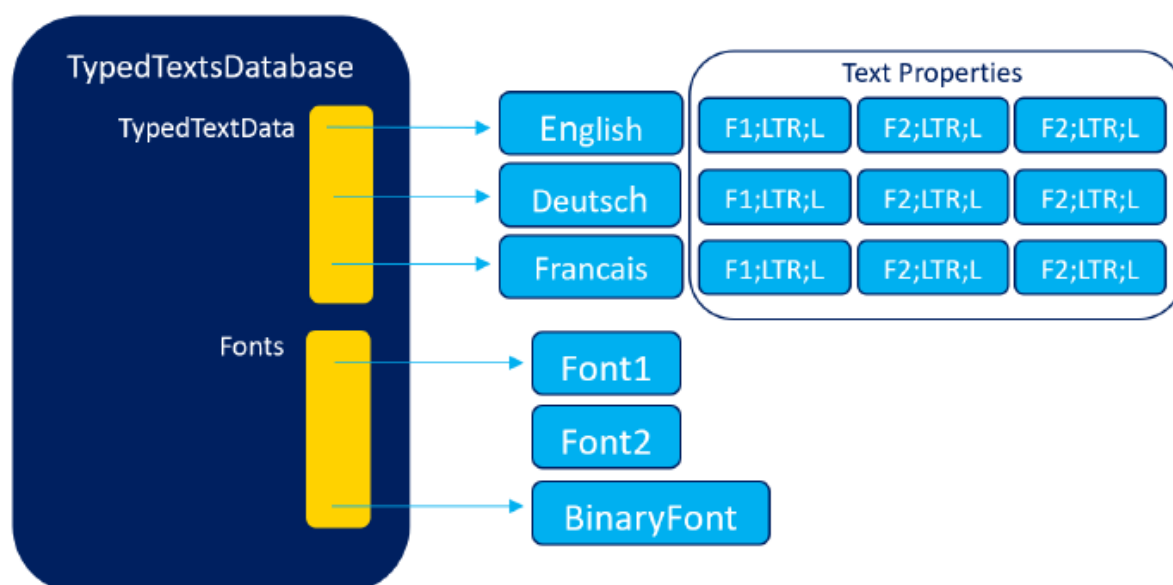
自从 2013 年第一款侧重于 GUI 应用的 STM32F429x【内嵌 Chrom-ART 图形加速和 LTDC 控制器】开始, ST 就为大家提供了 STM32MCU + X-Cube-TouchGFX 一站式 GUI 开发平台。越来越多的客户使用 STM32 + TouchGFX 开发智能手表/智能家居控制面板等嵌入式设备。

在嵌入式GUI开发的应用中, 设备厂家希望自己的设备支持用户根据自己喜好来定义所选用的字体类型。从技术上讲, 可以将字体和应用程序分离开, 这样可以在嵌入式设备有限的存储空间上根据需要更新和升级字体的类型。

2. TouchGFX Binary font 介绍

TouchGFX为客户提供了二进制字体(Binary Font)的方法, 这些字体没有链接到应用程序中, 而是作为文件与应用程序分开存储, 这些文件在运行时由应用程序加载并提供给TouchGFX。

- 例如, 应用程序可以从外部存储(如 SD 卡)加载字体, 或者从 Internet 或者手机等服务端下载字体文件。



当应用程序加载了二进制字体后, 可以在TouchGFX字体系统中安装二进制字体。这里内置的 Font2 被应用程序加载的 Binaryfont 替换。此后, TouchGFX 不再使用链接的Font2, 使用自定义的Binary font进行显示。

下面例程展示了如何使用BinaryFont进行字体动态更新。

3. 演示例程实现过程

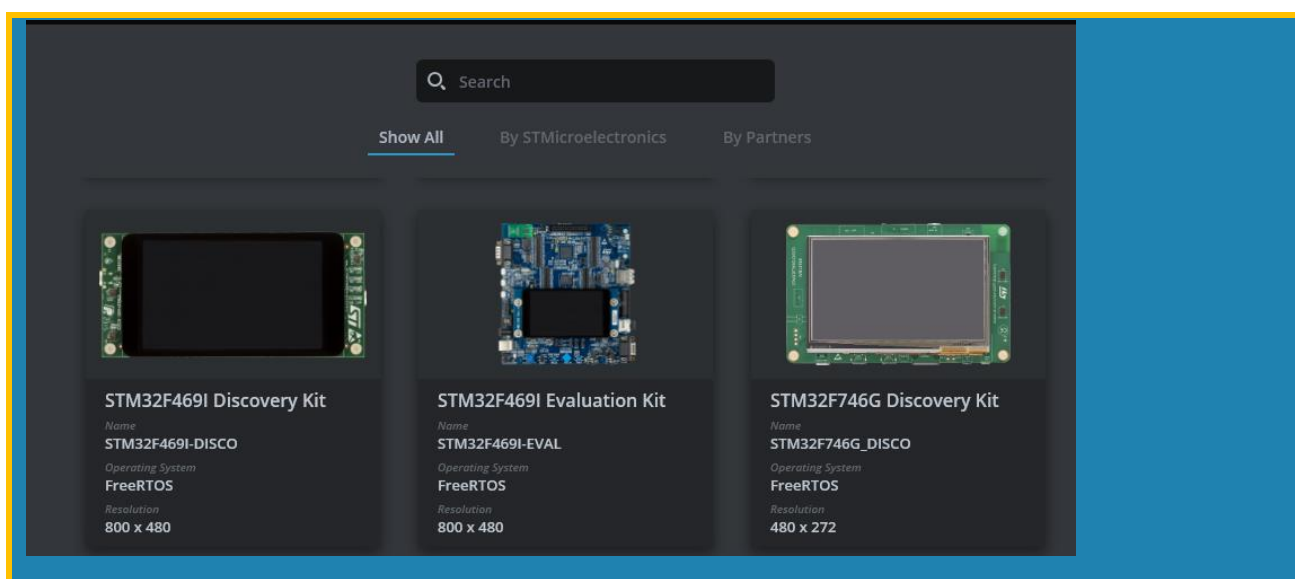
3.1.打开 TouchGFX Designer 4.18.0

- TouchGFX4.18.0 (本文使用 4.18.0 举例, 其他版本操作过程相同)
 - 环境安装请参考网址: <https://support.touchgfx.com/docs/introduction/installation>
- VSCode

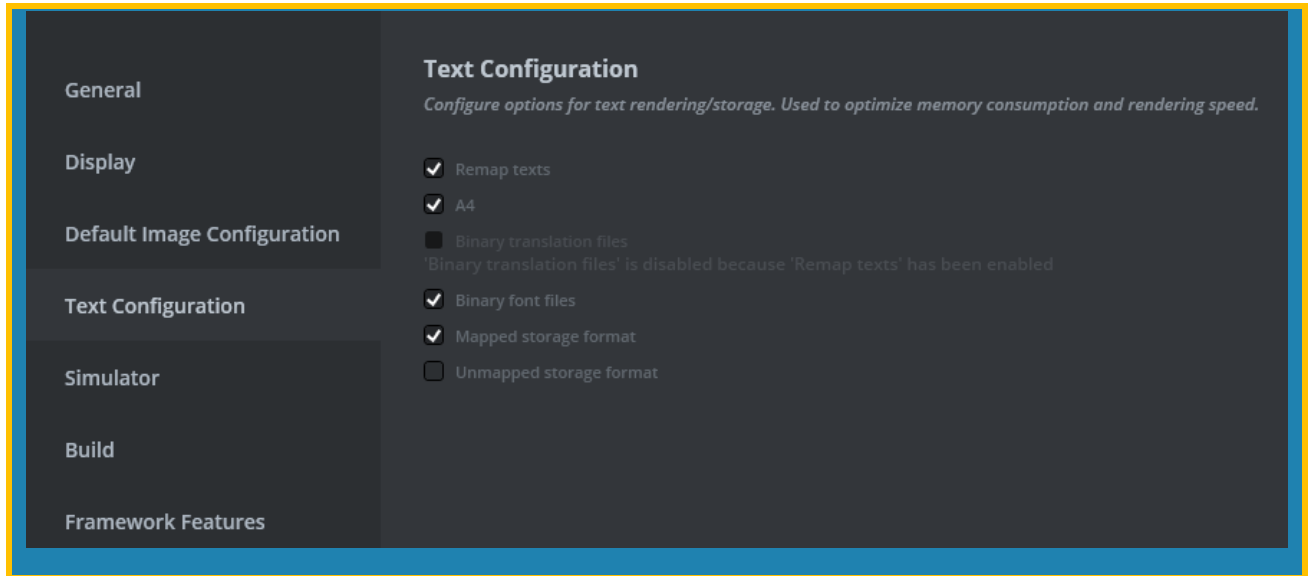
3.2.打开 TouchGFX Designer 4.18.0



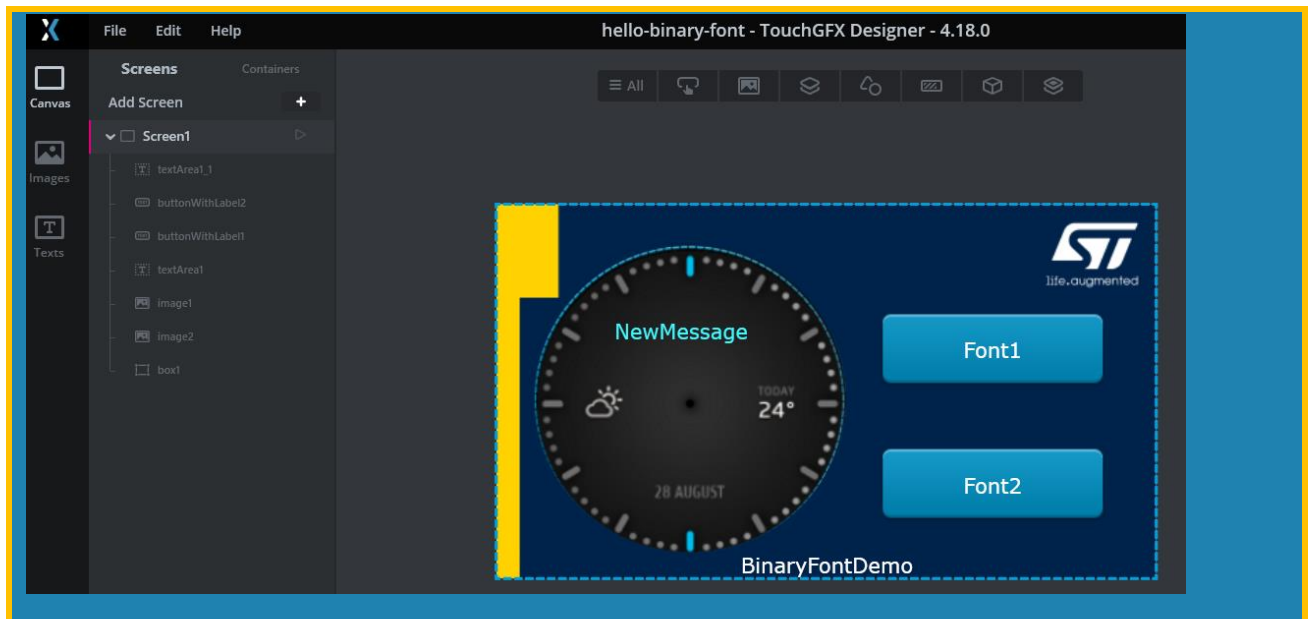
3.2.选择 STM32F746DK 探索板,生成工程 :



3.3 选择“文本配置”，选择“二进制字体文件”

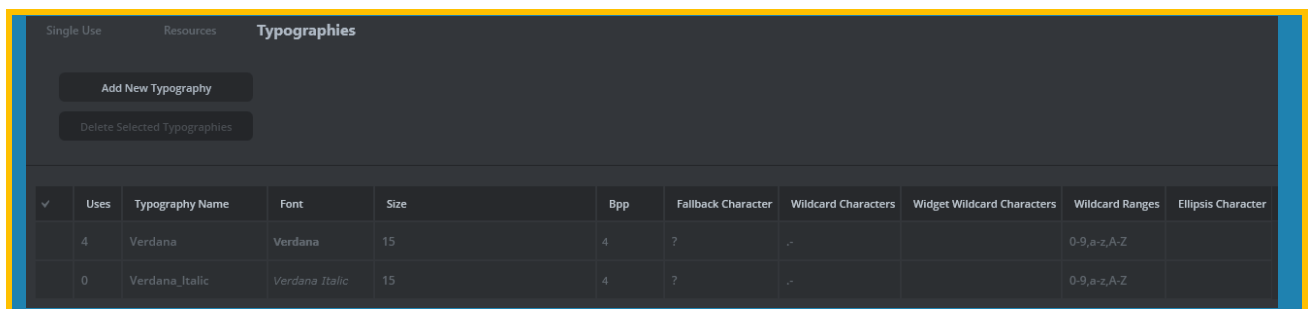


3.4 设计 Screen1 (添加 Images/TextArea/Buttons)



3.5 设置 Typographies :

- Typographies 设置 :



- Single Use 设置:

✓	Location	Widget	Typography	Alignment	Direction	GB	☰ ☲ ☱
	Screen1	textArea1	Verdana	☰ ☲ ☱	☰ ☲	BinaryFontDemo	
	Screen1	buttonWithLabel1	Verdana	☰ ☲ ☱	☰ ☲	Font1	
	Screen1	buttonWithLabel2	Verdana	☰ ☲ ☱	☰ ☲	Font2	
	Screen1	textArea1_1	Verdana	☰ ☲ ☱	☰ ☲	NewMessage	

3.6 TouchGFX Designer 生成项目

可以在以下目录下找到verdana和verdana italicic的二进制字体文件：

hello-tgfx4180-binary-font > TouchGFX > generated > fonts > bin		
Name	Date modified	Type
 Font_verdana_15_4bpp.bin	12/27/2021 12:45 ...	BIN File
 Font_verdanai_15_4bpp.bin	12/27/2021 12:45 ...	BIN File

3.7 在代码中安装二进制字体

在 TouchGFX 使用文本前，二进制字体必须从文件或其他存储中复制到可直接寻址的存储中。当应用程序将二进制字体加载到内存中时，它可以在 TouchGFX 中安装字体，之后 TouchGFX 将使用该字体而不是编译后的字体。

- 在FrontApplication.cpp安装默认的二进制字体:

```
#include <gui/common/FrontendApplication.hpp>
#include <stdio.h>
#include <stdlib.h>
#include <touchgfx/Texts.hpp>
#include <texts/TypedTextDatabase.hpp>
#include <fonts/GeneratedFont.hpp>
#include <touchgfx/Utils.hpp>

//read the file into this array in internal RAM
uint8_t fontdata[10000];

//binary font object using the data
BinaryFont bf;

FrontendApplication::FrontendApplication(Model& m, FrontendHeap& heap)
: FrontendApplicationBase(m, heap)
{
    //read the binary font from a file
#ifdef __GNUC__
    FILE* font = fopen("generated/fonts/bin/Font_verdana_15_4bpp.bin", "rb");
```

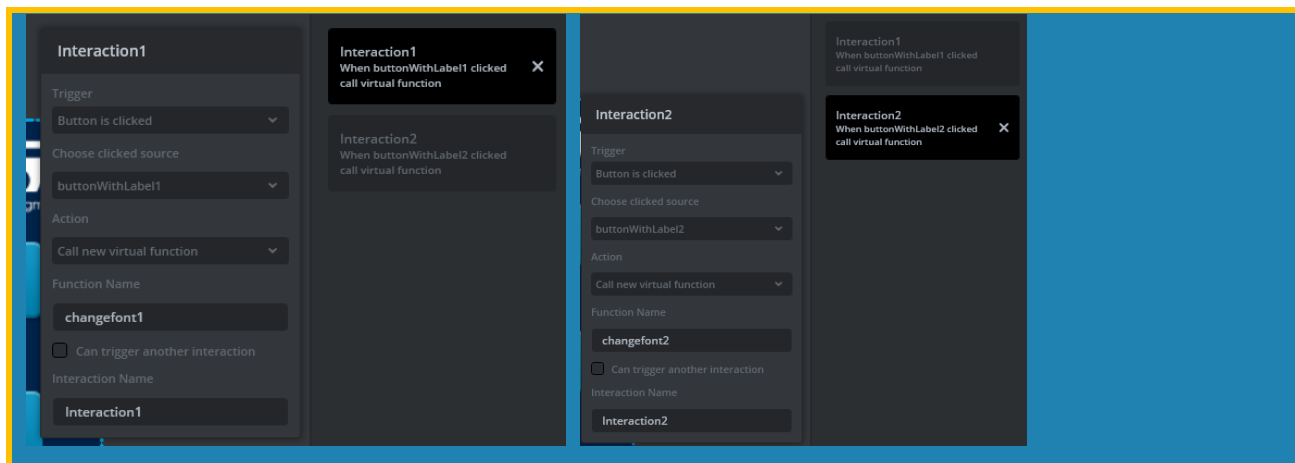
```
#else
FILE* font = 0;
fopen_s(&font, "../generated/fonts/bin/Font_verdana_15_4bpp.bin", "rb");
#endif
if (font)
{
    touchgfx_printf("open font file ok !!!\n");
    //read data from the file
    fread(fontdata, 1, 10000, font);
    fclose(font);

    //initialize BinaryFont object in bf using placement new
    new (&bf) BinaryFont((const struct touchgfx::BinaryFontData*)fontdata);

    //replace application font 'DEFAULT' with the binary font
    TypedTextDatabase::setFont(0, &bf); //verdana_20_4bpp
}
else{
    touchgfx_printf("open font file fail !!!\n");
}
}
```

3.7 添加 Button 交互，用于选择使用不同的二进制字体

- 添加交互，当按键按下，切换成相应的二进制字体：



- 添加代码 Screen1View.hpp

```
#ifndef SCREEN1VIEW_HPP
#define SCREEN1VIEW_HPP

#include <gui_generated/screen1_screen/Screen1ViewBase.hpp>
#include <gui/screen1_screen/Screen1Presenter.hpp>
#include <stdio.h>
#include <stdlib.h>
#include <touchgfx/Texts.hpp>
```

```
#include <texts/TextKeysAndLanguages.hpp>
#include <texts/TypedTextDatabase.hpp>
#include <fonts/GeneratedFont.hpp>
#include <touchgfx/Utils.hpp>

class Screen1View : public Screen1ViewBase
{
public:
    Screen1View();
    virtual ~Screen1View() {}
    virtual void setupScreen();
    virtual void tearDownScreen();
    virtual void changefont1();
    virtual void changefont2();
    //read the file into this array in internal RAM
    uint8_t fontdata[10000];
    BinaryFont bf;
    bool flag = 0;

#ifdef __GNUC__
    FILE* font;
#else
    FILE* font = 0;
#endif
protected:
};

#endif // SCREEN1VIEW_HPP
```

- 添加代码 Screen1View.cpp

```
void Screen1View::changefont1()
{
    //read the binary font from a file
#ifdef __GNUC__
    FILE* font = fopen("generated/fonts/bin/Font_verdana_15_4bpp.bin", "rb");
#else
    FILE* font = 0;
    fopen_s(&font, "../generated/fonts/bin/Font_verdana_15_4bpp.bin", "rb");
#endif
    if (font)
    {
        //touchgfx_printf("open font file ok !!!\n");
        //read data from the file
        fread(fontdata, 1, 10000, font);
        fclose(font);

        //initialize BinaryFont object in bf using placement new
        new (&bf) BinaryFont((const struct touchgfx::BinaryFontData*)fontdata);
    }
}
```

```

        //replace application font 'DEFAULT' with the binary font
        TypedTextDatabase::setFont(0, &bf); //verdana_20_4bpp
    }
    else{
        //touchgfx_printf("open font file fail !!!\n");
    }
    invalidate();
}

void Screen1View::changeFont2()
{
    //read the binary font from a file
#ifdef __GNUC__
    FILE* font = fopen("generated/fonts/bin/Font_verdanai_15_4bpp.bin", "rb");
#else
    FILE* font = 0;
    fopen_s(&font, "../generated/fonts/bin/Font_verdanai_15_4bpp.bin", "rb");
#endif
    if (font)
    {
        //touchgfx_printf("open font file ok !!!\n");
        //read data from the file
        fread(fontdata, 1, 10000, font);
        fclose(font);

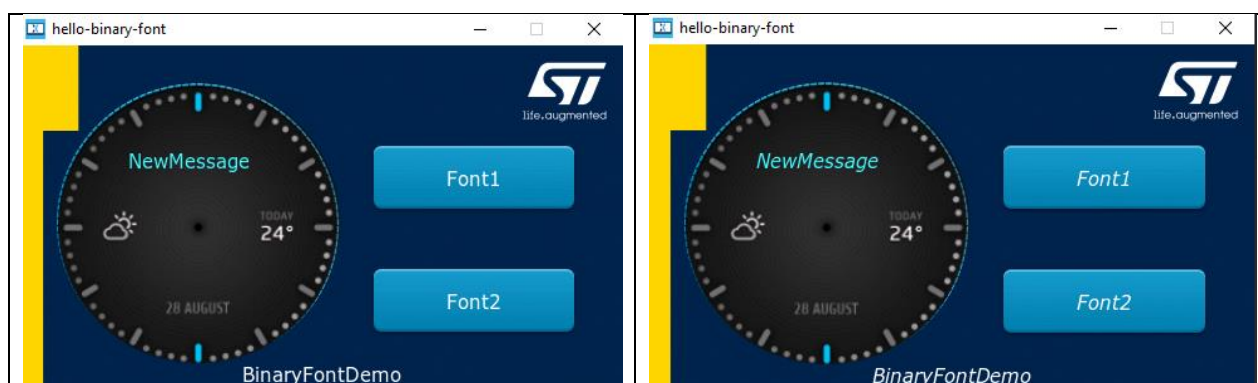
        //Initialize BinaryFont object in bf using placement new
        new (&bf) BinaryFont((const struct touchgfx::BinaryFontData*)fontdata);

        //replace application font 'DEFAULT' with the binary font
        TypedTextDatabase::setFont(0, &bf); //verdana_20_4bpp
    }
    else{
        //touchgfx_printf("open font file fail !!!\n");
    }
    invalidate();
}

```

3.8 编译运行结果如下：

按 Font1 按钮	按 Font2 按钮
------------	------------



4. 小结

从以上的演示可以看到，通过TouchGFX Designer的简单配置和调用其提供的API，就可以轻松实现字体的动态更新。

厂家和用户可以根据场景需求灵活使用本功能。比如，厂家可以通过针对的不同用户群体来提供OTA服务，让用户动态更新自己的字体类型。这样由于各二进制字体可以占用同一块内存区域，厂家的硬件成本也会得到降低。

具体示例实现过程，请参考示例代码：

《hello-tgfx4180-binary-font-v1.0-202112.7z》

参考文献

文件编号	文件标题	版本号	发布日期
Binary Fonts	https://support.touchgfx.com/4.13/docs/development/ui-development/touchgfx-engine-features/using-binary-fonts		

文档中所用到的工具及版本

TouchGFX Designer 4.18.0

Visual Code 1.63.2

LAT 中的附件

《 hello-tgfx4180-binary-font-v1.0-202112.7z 》

版本历史

日期	版本	变更
2022 年 04 月 11 日	1.0	首版发布

重要通知 - 请仔细阅读

意法半导体公司及其子公司（“ST”）保留随时对 ST 产品和 / 或本文档进行变更的权利，恕不另行通知。买方在订货之前应获取关于 ST 产品的最新信息。ST 产品的销售依照订单确认时的相关 ST 销售条款。

买方自行负责对 ST 产品的选择和使用，ST 概不承担与应用协助或买方产品设计相关的任何责任。

ST 不对任何知识产权进行任何明示或默示的授权或许可。

转售的 ST 产品如有不同于此处提供的信息的规定，将导致 ST 针对该产品授予的任何保证失效。

ST 和 ST 徽标是 ST 的商标。若需 ST 商标的更多信息，请参考 www.st.com/trademarks。所有其他产品或服务名称均为其各自所有者的财产。

本文档是 ST 中国本地团队的技术性文章，旨在交流与分享，并期望借此给予客户产品应用上足够的帮助或提醒。若文中内容存有局限或与 ST 官网资料不一致，请以实际应用验证结果和 ST 官网最新发布的内容为准。您拥有完全自主权是否采纳本文档（包括代码、电路图等信息），我们也不承担因使用或采纳本文档内容而导致的任何风险。

本文档中的信息取代本文档所有早期版本中提供的信息。

