一、 FR309x 屏显示方案开发介绍

1、安装 MDK5.36 以上版本,参考网上教程即可。然后打工工程。

| s | > FR3092_Dashboard > examples > | turnkey $ ightarrow$ lvgl_Dashboard $ ightarrow$ M | DK-ARM → | | |
|---|---------------------------------|--|-----------------|--------|--|
| | 名称 ^ | 修改日期 | 类型 | 大小 | |
| | 📙 Listings | 2023/11/7 11:29 | 文件夹 | | |
| | 📙 Objects | 2023/11/7 14:45 | 文件夹 | | |
| | 📙 output | 2023/11/7 11:29 | 文件夹 | | |
| | 📮 RTE | 2023/11/7 11:29 | 文件夹 | | |
| | 🃔 Project.uvguix.LH | 2023/11/7 14:45 | LH 文件 | 121 KB | |
| | Project.uvoptx | 2023/11/7 14:29 | | 149 KB | |
| | 😗 Project.uvprojx | 2023/11/7 11:40 🖊 | 礦ision5 Project | 99 KB | |
| | | | | | |
| l | | | | | |
| l | | | | | |
| | | | | | |

2、打开工程配置配置好 Device

| 👿 Options for Target 'Project' | | × |
|--|---|---|
| Options for Target 'Project' Device Target Output Listing User Software Packs Vendor: ARM Device: ARMCM33_DSP_FP Toolset: ARM Search: | C/C++ (AC6) Asm Linker Debug Vtilities | × |
| ARM Cortex M1 ARM Cortex M23 ARM Cortex M3 ARM Cortex M3 ARM Cortex M33 ARMCM33 ARMCM33 ARMCM33 ARMCM33 ARMCM33 T ARMCM34 ARMCM34 ARMCM | The Am Cortex-M33 is the most configurable of all Cortex-M processors. It is a full featured microcontrollerclass processor based on the Amv8-M mainline architecture with Am TrustZone security. DSP Instructions, Single Precision Floating Point Unit, no TrustZone | |
| OK | Cancel Defaults Help | |

3、工程目录结构介绍,包括一些外设初始化,RTOS 应用任务配置,蓝牙相关 api 接口, lvgl 相关界面字体等,编码器按键驱动,芯片外设驱动等等。



| app/lfs_port |
|---------------------------------------|
| i lfs_port.c |
| i⊇- 😂 app/lv_app |
| Iv_common_function.c |
| i∎… i batt_full_yellow.c |
| Number_HarmonyOS_bpp4_16px.c |
| I Inter_HarmonyOS_bpp4_12px.c |
| ⊕ Number_HarmonyOS_bpp4_36px.c |
| I Inter_HarmonyOS_bpp4_44px.c |
| ∎ Number_HarmonyOS_bpp4_20px.c |
| Batt_full_gren.c by alt B = = - → //+ |
| |
| □ □ □ app/lv_frame 图片/字体文件 |
| ■ fr_guimain.c |
| I fr_watch.c |
| ⊕ |
| i lv_user_sqlist.c |
| 📮 🦢 app/fr_device |
| fr_device_rtc.c |
| │ ● fr_device_button.c 外部相关设备驱动/编码器 |
| ■ ■ fr_device_pmu_io.c /按键等 |
| fr_device_encode.c |
| fr_device_vbat.c |
| |
| i fr_device_pa.c |
| 🗊 🚰 common/btdm |
| 🔁 🔁 common/dsp |
| 🔁 🦢 common/flashdb |
| · □ fal_flash_port.c |
| · fdb_app.c |



4、编译工程,编译成功后会在工程的 output 目录下生成 Project_mp_burn.bin 文件,这个 文件就是我们要烧录到芯片的应用程序。注意:这里编译后后会运行一个 python 脚本打包, 所以需要电脑上提前安装了 python 环境才行。



| FR3092_Dashboard > examples > turn | key > lvgl_Dashboard > | MDK-ARM > output | |
|------------------------------------|------------------------|------------------|-----------|
| 名称 | 修改日期 | 类型 | 大小 |
| 👌 app_post_process.py | 2023/10/13 10:12 | Python File | 2 KB |
| 📄 boot_load_burn.bin | 2023/10/18 15:30 | FTE Binary Expor | 194 KB |
| fr_keil_after_build.bat | 2023/10/11 14:26 | Windows 批处理 | 1 KB |
| Project.bin | 2023/11/7 14:45 | FTE Binary Expor | 524 KB |
| Project.hex | 2023/11/7 14:45 | HEX 文件 | 1,310 KB |
| Project.txt | 2023/11/7 14:45 | 文本文档 | 11,148 KB |
| Project_burn.bin | 2023/11/7 14:45 | FTE Binary Expor | 532 KB |
| Project_mp_burn.bin | 2023/11/7 14:45 | FTE Binary Expor | 524 KB |

5、烧录程序,打开 FreqChip_Download 软件,从工程目录下找到 boot_load_burn.bin, Project_mp_burn.bin 文件添加进来,指定文件烧录在选项里面设置开启与关闭,然后第一次 烧录时需要烧录 ui 资源文件,在 Resource 目录下找到 img.hex,这里指定的地址应用程序 Project_mp_burn 的固定是 0x72000, img.hex 根据内部 flash 剩余情况而定的这里需要注意 一下。这里使用的是 0x13a000,表示给固件预留了 800KB 使用,后面用来存放 ui 文件。可 以根据自己固件占用情况调整。如果 ui 文件太大内部 flash 不足以存放,需要烧录到外部 flash 可以通过 FR5080_DSP_MCU.exe 软件直接烧录到外部 flash。

| FREQCHIP调试工具V1.3.7.8 | | | × |
|--|---|------------------------|---------|
| 文件 选项 串口设置 帮助 | | | |
| 烧汞调试 粒据惊听 | | | |
| O FR801XH O FR801XT 512K O FR800X O FR508X Image: Free of the fre | ^ | | |
| 加载读录文件 | | | |
| ✓ 目动规定 ✓ 目动重月 ✓ 整片描述 打开境录 操 燥 录 | | | |
| - 与人文(中)成末 Flash地址 0x72000 洗経文性 已导入 | | | |
| 259年又FT Project_mp_6um_bin成功 -写入文件烧录1 F1 ash地址 0x13e000 | | | |
| 选择文件 已导入img hex成功 | | | |
| 与人文件规模A2 Flasb地址 0x50000 选择文件 | | | |
| WAC地址配置 | | | |
| 起始地址: C0:12:12:12:12:12 Flash地址: 0x60000 | | 总次数: 22 当前次数: 2 | 2 |
| 已续录地址: - SIB编号配置 | | □ 十六进制 □ 加时间間 | 刊 戳 |
| 起始SIM编号: 123456789ABCDEFO Flash地址: 0x61000 | | 保存 | e X |
| 已烧录编号: 数据发送 | | | |
| 串口配置 串口端号: COM3 → 波特率: 921600 → 送闭串口 | ~ | 计数 □ 十六进制 □ 加回车排 | り 執行 |
| | ~ | 友 送 清空发送 | |

选择好文件后打开烧录,按下板子复位按钮/或重新上电,提示已经连接自动开始烧录,成功提示如下信息。

| FREQCHIP调试工具V1.3.7.8 | | _ | | × |
|--|--|--------|--|----------|
| 文件 选项 串口设置 帮助 | | | | |
| 烧录调试 | | | | |
| | 数据接收 | _ | | |
| ○ FR801XH ○ FR801XT 512K √ ○ FR800X ○ FR506X ④ FR509X | *****已连援****** | ^ | | |
| O FR201X O FR101X | ******自动开始烧录****** | | | |
| 加载烧录文件 こ与入 boot_load_burn.bin成功 | *****开始烧录指定文件***** | | | |
| | ******烧录指定文件成功***** | | | |
| ☑ 自动烧录 ☑ 自动重启 ☑ 整片擦除 | *****开始烧录指定文件***** | | | |
| | *********施录指定文件1成功****** | | | |
| Flash地址 0x72000 | ******烧录指定文件2失败***** | | | |
| 送择文件 已与入 Project_mp_bum.bin成功 | ******CRC 校验成功****** | | | |
| 与人又件皖东1 Flash地址 0x13a000 | ******烧录成功***** | | | |
| 选择文件 已导入img.hex成功 | □@鸴O?€€€€€€€€(/EasyLogger V2.2.0 is initialize success. A/Hello ELOG LVL ASSERT! | | | |
| 写入文件烧录2 Flash地址 0x50000 | E/Hello ELOG_LVL_ERROR! | | | |
| 选择文件 | I/Hello ELOG_LVL_WARN | | | |
| | D/Hello ELOG_LVL_DEBUG! V/Hello ELOG_LVL_VERBOSE! | | | |
| MAC地址配置 | □[32;22m[l/FAL] ==================================== | | 345 L-111 - | |
| 起始地址: C0:12:12:12:12:12 | =[0m=[32;22m[I/FAL] | | 忠大数: 4 当前次称· | 24 |
| Flash通道: 0x60000 已烧录地址: | [I/FAL] FlashEnv flashdb_onchip 0x001a9000 0x00003000 □[0m□[32;22m[I/FAL] ==================================== | | | |
| SIN编号配置 | [0m:[32:22m[I/FAL] Flash Abstraction Layer (V0.5.99) initialize success.::[0mflashdb get random seed :4 | | □ 十六进 □ 加时间 | :制]戳 |
| 起始SIN编号: 123456789ABCDEFO | flash db get rand num: c6abbc92 | | 保 | 存 |
| Flash地址: 0x61000 | app_task DSP is released. | v | 清空接 | 枚 |
| 出版來编写: | 数据发送 | _ | | |
| | | ^ | 计数 | |
| 串山端号: ○0mm1 → 渋特室: 921600 → 关闭串ロ | | | □ 十六进 | 制 |
| ax13 | | | □ 加回车 | .换行 |
| | | | 发 j | Ě |
| | | \vee | 清空发 | Ě |

6、UI 转换&打包工具使用说明

先打开 lvgl_image_convert 软件将需要转换的图片文件选中,然后配置好颜色格式,输出格式,一般应用使用 NONE-TrueColor, bin_565 就可以了,如果涉及到图层叠加需要加透明的就选择 Alpha byte-TrueColor。转换成功后会在目录下生成对应图片的 bin 文件。这里转换工具也可以用 lvgl 官方在线工具转换:

Online image converter - BMP, JPG or PNG to C array or binary | LVGL

| 쮆 | 🦉 lvgl图片转换工具 (By | 夏雨夜寐) 1.0.3.0 | | - 🗆 × |
|--------------------|---|-------------------------------------|---|-------------------------------------|
| _ [| C:\Users\LH\Desktop\i | mg_Dashboard_ui | 图片路径 | |
| e_ m fc d | -batt_full -batt_full -oenter_ar -oiroul ar_ -instrumer -instrumer -instrumer -instrumer -instrumer -instrumer -instrumer -instrumer | () Line | 宽度: 23 高: 15, 长: 23 像素格式: Form 图片格式: PNG | 高度: 15 |
| | instrumer instrumer instrumer instrumer instrumer instrumer instrumer instrumer instrumer | | Transparency= lv_img_dso_t= 输出文件格式: | NONE-TrueColor ~ Bin_565 ~ 转换 |
| | < > C:/Users/LH/Desktop/i | instrument_1005 mg_Dashboard_ui/ | 保存路径 | Email联系我 QQ联系我 |

| Euro. | | | |
|--------------------------|------------------|------------------|--------|
| instrument_batt_ico5.bin | 2023/10/23 14:16 | FTE Binary Expor | 5 KB |
| instrument_batt_ico5.png | 2023/10/23 13:01 | PNG 文件 | 2 KB |
| 📄 instrument_bg1.bin | 2023/10/23 10:26 | FTE Binary Expor | 256 KB |
| 🖻 instrument_bg1.png | 2023/10/17 10:35 | PNG 文件 | 90 KB |
| instrument_bg2.bin | 2023/10/24 19:28 | FTE Binary Expor | 256 KB |
| 🖻 instrument_bg2.png | 2023/10/23 13:05 | PNG 文件 | 52 KB |
| instrument_ico1.bin | 2023/10/23 10:23 | FTE Binary Expor | 1 KB |
| instrument_ico1.png | 2023/10/17 12:09 | PNG 文件 | 1 KB |
| instrument_ico2.bin | 2023/10/23 10:24 | FTE Binary Expor | 1 KB |
| instrument_ico2.png | 2023/10/17 12:09 | PNG 文件 | 1 KB |
| instrument_ico3.bin | 2023/10/23 10:24 | FTE Binary Expor | 2 KB |
| instrument_ico3.png | 2023/10/17 12:09 | PNG 文件 | 2 KB |
| instrument_ico4.bin | 2023/10/23 10:24 | FTE Binary Expor | 1 KB |
| 🖻 instrument_ico4.png | 2023/10/17 12:09 | PNG 文件 | 1 KB |
| instrument_ico5.bin | 2023/10/23 10:24 | FTE Binary Expor | 1 KB |
| instrument ico5.png | 2023/10/17 12:09 | PNG 文件 | 1 KB |

转换好 bin 文件后,将打包工具 freqchip_tool_img_cut_rle_encode_v09.exe,config.txt 两个文件复制到 bin 文件当前的目录下,通过 config.txt 进行配置一些文件打包的选项,

img_head_address=0x813A000 这个表示图片存放到 flash 的起始地址,存在内部 flash 的话 要加上 0x8000000 偏移量,外部 flash 和 pc 模拟器上使用时设置为 0 即可。

img_cut_w, img_cut_h 这两个参数是根据图片大小要进行切图的行数, 这里内部缓冲区大小 是 55936Byte, 图片大小超出这个范围的就要进行切图, 根据屏幕宽度大小进行调整,

例如 368x448 的屏幕, img_cut_w=屏幕宽度大小 368, h=img_cut_w/(368*2)=76, img_cut_h 就设置为 76。

例如 480x272 的屏幕, img_cut_w =480,h 向下取整设置为 56。

rle_enable=1,设置是否压缩使能,模拟器上跑不要加压缩设置为 0, 真机上设置为 1。图片 会进行 rle 压缩。

upper_enable=1 设置图片名称宏字符大小写,设置为1后名称宏全部转换成大写格式,设置为0不改变名称。根据文件名定义图片的宏定义。

cut_exclude_char=clock_hand 默认使用 clock_hand 不能更改。

align_left_space=80 设置图片名称宏字符串左边对齐字符宽度,一般不用改。

配置好后双击 freqchip_tool_img_cut_rle_encode_v09.exe 脚本,就会将所有 bin 文件打包到 一起输出一个 img.hex,这个文件就是烧录 flash 中的文件, img_def.h 文件为每张定义存到 flash 中地址信息,要添加到工程里面去,程序里面调用显示图片设置时就调用对应的宏就 可以了。



注意:

img_cut_w, img_cut_h 修改后程序 lv_conf.h 里面也要修改。



生成 img.hex 后不能随意去修改 img_def.h 文件里面的地址信息烧录到其他地址。只能通过 调整 config.txt 文件去修改然后重新生成文件。单独在程序里面去调整 img_head_address 这 个地址不会生效,程序运行会出问题的。

| | 1 | 1 | - | | | |
|---|------------------------|------------|--------|------------------------|---------|--------|
| Ĵ | lv_conf.h | img_def.h | 🗋 f | r_lv_dashbo | ard.c | main.c |
| | 1 ⊟ <mark>#ifnd</mark> | efIMG_D | DEF_H | | | |
| | 2 #defi | neIMG_D | DEF_H | | | |
| | 3 | | | | | |
| | 4 | | | | | |
| | 5 #defi | ne img_hea | ad_ado | lress <mark>0</mark> x | 813a000 | |
| | 6 | | | | | |
| | 7 | | | | | |
| | 8 | | | | | |

7、字体转换工具介绍

如果字体存放到内部 flash 的话直接使用 lvgl 在线转换工具将转换好的 c 文件添加到工程使用即可。

Online font converter - TTF or WOFF fonts to C array | LVGL

| Name | test_font_36px | |
|---|---|--|
| Size | 36 | |
| Врр | 4 bit-per-pixel | ~ |
| Enable Font of Horizontal su Try to use gly Since gray tones | compression (reduces size l bpixel rendering (may imp ph color info from font to c are emulated via transparency, re | but results in slower rendering) rove font quality but results in larger fonts) reate grayscale icons. sult will be good on contrast background only. |
| TTF/WOFF file | 选择文件 arialuni.ttf | |
| Range | 0x20-0x7f | |
| Symbols | 好的你好在线转换测试 | |
| Include anot You can use both "A Convert | her font | of them |

如果有很多种字体文件或者文件较大比如全字库的需要放到外部 flash 中存放,按以下步骤进行操作。

先将所有需要转换的字体文件转换成.c 文件, 然后放到一个文件夹里面, 再将 freqchip_tool_string_c_to_bin_V02.exe 和 config.txt 文件复制进来, 如下图。

| 名称 | 修改日期 | 类型 | 大小 |
|---------------------------------------|------------------|----------------------|-----------|
| 🥌 arialuni_bbp1_32px.c | 2023/9/18 13:51 | sourceinsight.c_file | 26,692 KB |
| 📄 config.txt | 2023/11/6 17:21 | 文本文档 | 1 KB |
| freqchip_tool_string_c_to_bin_V02.exe | 2023/11/6 17:17 | 应用程序 | 6,004 KB |
| Wumber_HarmonyOS_bpp4_16px.c | 2023/10/23 11:18 | sourceinsight.c_file | 46 KB |
| Wumber_HarmonyOS_bpp4_20px.c | 2023/9/18 13:51 | sourceinsight.c_file | 61 KB |
| Wumber_HarmonyOS_bpp4_36px.c | 2023/9/18 13:51 | sourceinsight.c_file | 123 KB |
| Wumber_HarmonyOS_bpp4_44px.c | 2023/9/18 13:51 | sourceinsight.c_file | 33 KB |
| Wumber_HarmonyOS_bpp4_46px.c | 2023/9/18 13:51 | sourceinsight.c_file | 36 KB |
| Wumber_HarmonyOS_bpp4_50px.c | 2023/9/18 13:51 | sourceinsight.c_file | 41 KB |
| Wumber_HarmonyOS_bpp4_62px.c | 2023/9/18 13:51 | sourceinsight.c_file | 59 KB |
| Wumber_HarmonyOS_bpp4_80px.c | 2023/9/18 13:51 | sourceinsight.c_file | 94 KB |
| Wumber_HarmonyOS_bpp4_92px.c | 2023/9/18 13:51 | sourceinsight.c_file | 121 KB |
| Wumber_HarmonyOS_bpp4_128px.c | 2023/9/18 13:51 | sourceinsight.c_file | 275 KB |
| Wumber_HarmonyOS_bpp4_180px.c | 2023/9/18 13:51 | sourceinsight.c_file | 386 KB |
| OSD_arialuni_bbp4_32px.c | 2023/9/21 17:33 | sourceinsight.c_file | 1,598 KB |

配置文件信息如下

align_left_space=50 字体名称宏定义对齐字符 一般不用设置

flash_head_address=0xA12000 存放外部 flash 的起始地址

output_c_file_dir= //设置输出.c 文件目录 可以设置为工程对应的路径,也可以不设置输出 文件自行复制到对应的地方。

配置完成后双击 freqchip_tool_string_c_to_bin_V02.exe 脚本就可以自动生成 bin_font.bin 文件和相关的.c.h 文件,通过下载工具将 bin_font.bin 烧录到外部 flash,将 c_files_output 目录下的.c 文件和 bin_font.h 文件添加到工程里面编译。

| 名称 | 修改日期 | 类型 | 大小 |
|---------------------------------------|------------------|------------------------|-----------|
| 🖡 c_files_output | 2023/11/7_18:39 | 文件夹 | |
| 🍯 arialuni_bbp1_32px.c | 2023/9/18 13:51 | sourceinsight.c_file | 26,692 KB |
| 🗋 bin_font.bin | 2023/11/7 18:39 | FTE Binary Export File | 4,557 KB |
| 🍯 bin_font.h | 2023/11/7 18:39 | H文件 | 3 KB |
| 📄 config.txt | 2023/11/6 17:21 | 文本文档 | 1 KB |
| freqchip_tool_string_c_to_bin_V02.exe | 2023/11/6 17:17 | 应用程序 | 6,004 KB |
| Mumber HarmonyOS boot 16py c | 2022/10/22 11:19 | courcoinsight s file | 16 VP |

| ui_tools > freqchip_tool_string_c_to_bin_V02 > c_files_output | | | | | |
|---|------------------------------|-----------------|----------------------|-------|--|
| | 名称 | 修改日期 | 类型 | 大小 | |
| | 🥌 arialuni_bbp1_32pxc | 2023/11/7 18:39 | sourceinsight.c_file | 64 KB | |
| | Wumber_HarmonyOS_bpp4_16pxc | 2023/11/7 18:39 | sourceinsight.c_file | 9 KB | |
| | Wumber_HarmonyOS_bpp4_20pxc | 2023/11/7 18:39 | sourceinsight.c_file | 10 KB | |
| | Wumber_HarmonyOS_bpp4_36pxc | 2023/11/7 18:39 | sourceinsight.c_file | 10 KB | |
| | Wumber_HarmonyOS_bpp4_44pxc | 2023/11/7 18:39 | sourceinsight.c_file | 4 KB | |
| | Wumber_HarmonyOS_bpp4_46pxc | 2023/11/7 18:39 | sourceinsight.c_file | 4 KB | |
| | Wumber_HarmonyOS_bpp4_50pxc | 2023/11/7 18:39 | sourceinsight.c_file | 4 KB | |
| | Wumber_HarmonyOS_bpp4_62pxc | 2023/11/7 18:39 | sourceinsight.c_file | 4 KB | |
| | Wumber_HarmonyOS_bpp4_80pxc | 2023/11/7 18:39 | sourceinsight.c_file | 4 KB | |
| | Wumber_HarmonyOS_bpp4_92pxc | 2023/11/7 18:39 | sourceinsight.c_file | 4 KB | |
| | Wumber_HarmonyOS_bpp4_128pxc | 2023/11/7 18:39 | sourceinsight.c_file | 4 KB | |
| | Wumber_HarmonyOS_bpp4_180pxc | 2023/11/7 18:39 | sourceinsight.c_file | 4 KB | |
| | OSD_arialuni_bbp4_32pxc | 2023/11/7 18:39 | sourceinsight.c_file | 15 KB | |
| | | | | | |

版本说明

| 版本号 | 日期 | 作者 | 修改内容 |
|-------|------------|----|------|
| 1.0.0 | 2023-11-07 | 刘浩 | 初版 |
| | | | |
| | | | |