

**LT776**

**串口屏方案**

*Serial Uart TFT Panel Solution*

---

**应用手册**

**Application Note**

**V2.0**

## 版本记录

版别	发布日期	改版说明
V1.0	2021/10/12	<ul style="list-style-type: none"><li>● Preliminary version (初版)。</li></ul>
V1.1	2021/12/11	<ul style="list-style-type: none"><li>● 图片显示增加乐升专用的压缩格式, 縮減“UartTFT_Flash.bin”的档案大小。</li></ul>
V2.0	2022/1/18	<ul style="list-style-type: none"><li>● 移除第 5 章: 图文整合编译器(UartTFT_Tool.exe)</li><li>● 增加第 4.2.17 节: 实时曲线的设定</li><li>● 增加第 4.5.13 节: 键盘指令</li><li>● 增加第 4.5.14 节: 滑动手势指令</li></ul>

目 录

1. 概述.....	13
1.1 LT776 基本介绍.....	13
1.2 TFT 串口屏的软硬件架构.....	15
1.3 TFT 串口屏的演示板(Demo Kit).....	18
1.4 TFT 串口屏的升级 .....	19
2. TFT 串口屏硬件接口 .....	22
2.1 Uart 串口与 RS232/RS485 接口 .....	22
2.2 TFT 屏 RGB/LVDS/MIPI 接口 .....	23
2.3 PWM 控制背光.....	24
2.4 SPI Flash 接口.....	24
2.5 USB 接口.....	25
2.1 SDIO 接口.....	25
2.2 PWM 控制蜂鸣器/喇叭.....	26
2.3 SCI 通讯接口 .....	26
2.4 触控接口.....	27
2.5 I2S 接口 .....	27
2.6 DCMI 摄像头接口 .....	28
2.7 时钟信号与 RTC (Real Time Clock) .....	28
2.8 TFT 串口屏参考原理图 .....	29
2.9 EMC 干扰与抗干扰对策.....	31
3. 串口指令.....	34
3.1 串口屏指令集 .....	34
3.2 主控端与 TFT 串口屏协议表.....	36
3.3 RS-232(UART) 通讯协议.....	39
3.4 主控端发送指令的范例.....	42
4. 图文 UI 编辑编辑器 (UI_Editor.exe) .....	45

4.1 UI_Editor 界面的简介.....	45
4.1.1 使用 UI_Editor 的设计流程 .....	50
4.1.2 使用 UI_Editor 的注意事项 .....	52
4.1.3 屏幕初始化和背景图的设定.....	54
4.1.4 显示图的设定.....	57
4.1.5 界面的编辑与调试 .....	60
4.2 显示图片的设定 .....	65
4.2.1 设定显示多张图片 .....	65
4.2.2 设定循环显示重叠图片 .....	66
4.2.3 设定卷动出现图片 .....	67
4.2.4 设定循环卷动图片 .....	68
4.2.5 设定显示 GIF 动画图片.....	69
4.2.6 设定显示二维码图片 .....	70
4.2.7 设定显示控件功能的图片 .....	71
4.2.8 设定显示虚拟按键 .....	72
4.2.9 设定开机画面.....	73
4.2.10 设定触摸进度条图 .....	76
4.2.11 设定环形触摸进度条 .....	79
4.2.12 设定指针图.....	80
4.2.13 设定进度条图.....	83
4.2.14 设定带透明度的单张图片 .....	86
4.2.15 控件滑动及显示.....	87
4.2.16 设定显示数字键盘 .....	88
4.2.17 实时曲线的设定.....	92
4.3 显示文字的设定 .....	93
4.3.1 图像形式显示文字 .....	93
4.3.2 图片式的数字显示 .....	94
4.3.3 真彩图片式数字显示 .....	96
4.3.4 使用字库显示文字 .....	97
4.3.5 图片式的 ASCII 显示.....	98
4.4 几何绘图功能 .....	100
4.4.1 基本绘图功能.....	100
4.4.2 表格制作 .....	104
4.5 其他功能.....	105
4.5.1 全屏触摸滑动设定及显示指令 .....	105
4.5.2 背光控制指令 .....	106
4.5.3 起始控制指令 .....	107

4.5.4	图片缓冲指令 .....	107
4.5.5	声音控制指令 .....	108
4.5.6	串口屏侦测指令 .....	109
4.5.7	User 手动输入指令 .....	111
4.5.8	RTC 数字时钟指令 .....	112
4.5.9	RTC 模拟时钟指令 .....	113
4.5.10	显示星期指令 .....	115
4.5.11	寄存器操作 .....	116
4.5.12	关闭触控功能指令 .....	117
4.5.13	键盘指令 .....	118
4.5.14	滑动手势指令 .....	119
4.6	WAV 音频、字库文件的说明 .....	120
4.6.1	WAV 音频文件说明 .....	120
4.6.2	字库文件的说明 .....	120
4.6.3	真彩数字图片文件说明 .....	121
4.7	BIN 文件的生成和串口发送指令 .....	122
4.7.1	SPI Flash 的结构 .....	129
4.7.2	Userinfo.bin (128K) 更新 .....	130
4.8	UI_Editor 范例下载 .....	131
5.	制作字库与 Wav 档的 Bin 文件 .....	133
5.1	制作字库的 Bin 文件 .....	133
5.1.1	全字库制作 .....	133
5.1.2	自定义字库制作 .....	136
5.2	制作 Png ASCII 码 Bin 文件 .....	139
5.3	制作 Wav 档的 Bin 文件 .....	141
5.3.1	音频文件转 WAV .....	141
5.3.2	制作 WAV 档 Bin 文件 .....	146
5.3.3	典型音频驱动电路 .....	149
6.	版权说明 .....	150

## 表 目 录

表格 3-1: LT776 串口屏指令集 .....	34
表格 3-3: 主控端与 TFT 串口屏协议表 .....	36
表格 3-4: 串口屏接收的指令信息 .....	40
表格 3-5: 串口屏反馈的信息 .....	40
表格 4-1: 几何绘图功能 .....	100
表格 4-2: 起始控制指令 .....	109
表格 4-3: 串口屏版本信息 .....	109
表格 4-4: 指令总表 .....	126

**图目录**

图 1-1: 应用方块图..... 13

图 1-2: LT776 应用方式 - 1 ..... 14

图 1-3: LT776 应用方式 - 2 ..... 14

图 1-4: 主控端 MCU 的 Uart 与 LT776 串口屏的 Uart 连接示意图 ..... 15

图 1-5: 使用上位机软件开发的示意图..... 15

图 1-6: 主控板与 TFT 串口屏连接 ..... 16

图 1-7: 串口屏的指令协议范例一 ..... 16

图 1-8: 串口屏的指令协议范例二 ..... 17

图 1-9: 串口屏的开发演示板 (Demo Kit) ..... 18

图 1-10: SWD 烧录 ..... 19

图 1-11: SD Card 插入 SD 卡槽内..... 19

图 1-12: 以 SD Card 进行更新中 ..... 20

图 1-13: 更新的档案所在的储存目录 ..... 20

图 2-1: 主控端 MCU 透过 Uart 串口与 LT776 串口屏通讯..... 22

图 2-2: 加上 RS232 延伸通讯距离 ..... 22

图 2-3: LT776 与 TFT 屏之间采 16 位 RGB 接口示意图..... 23

图 2-4: RGB 转 LVDS 示意图 ..... 23

图 2-5: LT776 用 TFT\_PWM1 控制背光及 GPIO 控制显示 On/Off..... 24

图 2-6: LT776 与 SPI Flash 接口示意图 ..... 24

图 2-7: SPI Flash 烧录器..... 25

图 2-8: MCU 与 USB 的接口示意图..... 25

图 2-9: SDIO 接口参考原理图..... 25

图 2-10: 用 LT776 的 PWM 输出控制蜂鸣器/喇叭..... 26

图 2-11: 用 LT776 的 DAC\_OUT 输出及小功率放大器推动喇叭 ..... 26

图 2-12: LT776 与 WiFi 模块、蓝牙模块的接口示意图..... 26

图 2-13: LT776 与电容式触控 IC 接口示意图..... 27

图 2-14: I2S 连接 CODEC 参考原理图..... 27

图 2-15: DCMI 连接摄像头参考原理图..... 28

图 2-16: LT776 时钟信号结构图 ..... 28

图 2-17: LT776 TFT 串口屏开发板 (Demo Kit) ..... 29

图 2-18: LT776 TFT 串口屏原理图 ..... 30

图 2-19: 3.3V 的 DC to DC 电源输入、输出端加上 LC 电路..... 31

图 2-20: 3.3V 的 DC to DC 电源电路范例..... 31

图 2-21: TFT 屏背光的 DC to DC 电源输入、输出端加上 LC 电路..... 32

图 2-22: TFT 屏背光的 DC to DC 电源电路范例..... 32

图 2-23: TFT 屏的 FPC 做包覆处理..... 32

图 2-24: RGB 等信号输出加串接电阻及接地电容.....	33
图 2-25: 降低 EFT 干扰的参考电路.....	33
图 4-1: UI_Editor 主画面.....	45
图 4-2: UI_Editor 工具同级文件目录.....	47
图 4-3: PROJECT 文件夹下级的工程文件目录.....	48
图 4-4: UI_Editor 重装载工程文件.....	49
图 4-5: 使用 UI_Editor 的设计流程.....	50
图 4-6: 不同的工程建立不同的项目目录.....	52
图 4-7: 点击“SAVE”保存当前制作的工程.....	52
图 4-8: 点击编译“Build All”保存当前制作的工程.....	52
图 4-9: 可以自行选择删除不需要的工程文件夹.....	52
图 4-10: 图片素材不允许存在相同命名.....	53
图 4-11: 打开的 UI_Editor.....	54
图 4-12: 新建工程.....	54
图 4-13: 设置图层分辨率.....	55
图 4-14: 拖动屏幕框到任意位置.....	55
图 4-15: 取消了屏幕框选择功能.....	56
图 4-16: 增加图层.....	57
图 4-17: 新建图层与主图层.....	57
图 4-18: 新建独立的图层.....	58
图 4-19: 变更图层背景.....	58
图 4-20: 删除或移动图层.....	59
图 4-21: 删除操作方式一.....	60
图 4-22: 删除操作方式二.....	60
图 4-23: 异位置克隆操作.....	61
图 4-24: 同位置克隆操作.....	61
图 4-25: 复制功能一.....	62
图 4-26: 复制功能二.....	62
图 4-27: 复制功能三.....	63
图 4-28: 编辑状态.....	63
图 4-29: 调试状态.....	64
图 4-30: 设定显示多张图片.....	65
图 4-31: 设定循环显示重叠图片.....	66
图 4-32: 设定卷动出现图片.....	67
图 4-33: 设定循环卷动图片.....	68
图 4-34: 设定显示 GIF 动画图片.....	69
图 4-35: 设定显示二维码.....	70
图 4-36: 设定显示控件功能的图片.....	71

图 4-37: 设定显示虚拟控件 .....	72
图 4-38: 点选左上角 “mainLay” .....	73
图 4-39: 选择每一组 Command 的指令组合 .....	74
图 4-40: 开机画面所选的指令 .....	74
图 4-41: 以 9A 指令来设定执行多组指令 .....	75
图 4-42: 以 9A、01 指令来执行多组指令 .....	75
图 4-43: 设定滑动条及前景颜色与背景颜色 .....	76
图 4-44: 设定滑动条背景图片 .....	76
图 4-45: 新增滑动条的移动图标头 .....	77
图 4-46: 背景图片与滑动图标头各自对齐 .....	77
图 4-47: 设定滑动条背景图片与移动图标模式 .....	78
图 4-48: 背景图片 Command 选择窗口 .....	78
图 4-49: 滑动图标头 Command 选择窗口 .....	78
图 4-50: 选择完毕后点击确认 .....	78
图 4-51: 添加环形触控进度条底图和触控图标 .....	79
图 4-52: 设定环形进度条 .....	79
图 4-53: 设定指针对左右两侧的背景颜色 .....	80
图 4-54: 添加指针背景图片 .....	81
图 4-55: 调整指针背景图片显示位置 .....	81
图 4-56: 选择指针背景图片 Command .....	82
图 4-57: 点选指针背景图片模式第一组数据 .....	82
图 4-58: 为指针添加背景图片指令 .....	82
图 4-59: 设定进度条背景与进度条移动时的颜色 .....	83
图 4-60: 设定背景图片 .....	83
图 4-61: 调整进度条背景图片显示位置 .....	84
图 4-62: 选择进度条背景图片 Command .....	84
图 4-63: 点选进度条背景图片模式数据 .....	84
图 4-64: 选择进度图的背景图片 .....	85
图 4-65: 设定带透明度的单张图片 .....	86
图 4-66: 控件滑动及显示 .....	87
图 4-67: 添加数字键盘底图 .....	88
图 4-68: 添加数字键盘底图 .....	89
图 4-69: 数字键盘设定 .....	89
图 4-70: 数字键盘设定 .....	90
图 4-71: 数字键盘设定 .....	90
图 4-72: 数字键盘设定 .....	91
图 4-73: 曲线指令设定 .....	92
图 4-74: 曲线参数设定 .....	92

图 4-75: 图片形式显示文字 .....	93
图 4-76: 图片式的数字显示的设置窗口 .....	94
图 4-77: 调整图片式数字显示的字体框 .....	95
图 4-78: 调整真彩图片式数字显示的字体框 .....	96
图 4-79: 字库显示文字设置 .....	97
图 4-80: 图片式的 ASCII 显示设置 .....	98
图 4-81: 图片式的 ASCII 显示设置 .....	98
图 4-82: 图片式的 ASCII 显示设置 .....	99
图 4-83: 画圆设置 .....	100
图 4-84: 画四边形设置 .....	101
图 4-85: 画方柱体设置 .....	102
图 4-86: 画直线设置 .....	102
图 4-87: 画圆环设置 .....	103
图 4-88: 表格设置 .....	104
图 4-89: 触控滑动设定及显示指令 .....	105
图 4-90: Command 发送 B4 指令 .....	105
图 4-91: 背光控制指令 .....	106
图 4-92: 背光控制指令 .....	106
图 4-93: 起始控制指令 .....	107
图 4-94: 图片缓存指令 .....	107
图 4-95: UI_Editor 里的 WAV 音频文件 .....	108
图 4-96: 声音控制指令 .....	108
图 4-97: 声音控制指令 .....	108
图 4-98: User 手动输入指令 .....	111
图 4-99: 数字时钟指令 92h 设置 .....	112
图 4-100: 模拟时钟指令 93h 设置 .....	113
图 4-101: 模拟时钟指令 93h 设置 .....	113
图 4-102: 模拟时钟指令 93h 设置 .....	114
图 4-103: 模拟时钟指令 93h 设置 .....	114
图 4-104: 显示星期指令 9Dh 设置 .....	115
图 4-105: 显示星期指令 9Dh 设置 .....	115
图 4-106: CA-CF 操作举例说明 .....	116
图 4-107: 9Eh 操作举例说明 .....	117
图 4-108: 全键盘相关文件 .....	118
图 4-109: 设置 A6 指令 .....	118
图 4-110: 设置 C9 指令 .....	119
图 4-111: Wav BIN 文件 .....	120
图 4-112: FONT 文件 .....	120

图 4-113: 真彩数字图片文件.....	121
图 4-114: INFO 内配置参数.....	122
图 4-115: 工程编译.....	122
图 4-116: 工程编译成功.....	123
图 4-117: Command 发送指令给 TFT 串口屏.....	124
图 4-118: 指令表导出 PDF 格式.....	125
图 4-119: SPI Flash 的结构.....	129
图 4-120: SPI Flash 烧录器与接线方式.....	129
图 4-121: UserInfo.bin 更新按钮.....	130
图 4-122: 解压缩及打开压缩包.....	131
图 4-123: 范例目录拷贝到 UI_Editor 的 PROJECT 文件夹内.....	131
图 4-124: 打开 UI_Editor 点击 Load.....	132
图 4-125: 打开 UI_Editor 演示范例.....	132
图 5-1: 制作中文字库.....	133
图 5-2: 选择字体.....	134
图 5-3: 设置字库.....	134
图 5-4: 保存字库.....	135
图 5-5: 字库制作完成.....	135
图 5-6: 导出的字库 Bin 文件.....	136
图 5-7: 选择 txt 文档.....	136
图 5-8: 存放文字的 txt 文档.....	137
图 5-9: 保存字库.....	137
图 5-10: 字库制作完成.....	138
图 5-11: 导出的字库 Bin 文件.....	138
图 5-12: 制作 Png ASCII 码文件.....	139
图 5-13: 选择字体.....	139
图 5-14: 调整位置并生成 Bin 文件.....	140
图 5-15: 选择功能.....	141
图 5-16: wav 功能界面.....	142
图 5-17: 添加 wav 文件.....	142
图 5-18: 输出配置.....	143
图 5-19: 进入剪辑功能.....	143
图 5-20: 剪辑界面.....	144
图 5-21: 选择输出文件夹.....	144
图 5-22: 开始转换.....	145
图 5-23: 导出的 wav 文件.....	145
图 5-24: WAV Bin 文件制作界面.....	146
图 5-25: 添加 wav 文件.....	146

图 5-26: 添加成功.....	147
图 5-27: WAV Bin 文件制作界面.....	147
图 5-28: 导出 bin 文件 .....	148
图 5-29: 错误提示.....	148
图 5-30: 点击 OK 导出.....	148
图 5-31: 导出成功.....	149
图 5-32: 生成的 bin 文件 .....	149
图 5-33: DAC_OUT 输出及小功率放大器的音频驱动电路 .....	149

# 1. 概述

本应用手册主要在说明 LT776 TFT 图形加速控制器在 TFT 串口屏上的应用，及介绍串口屏开发工具 - [图文 UI 编辑器 \(UI\\_Editor.exe\)](#) 的使用方式，让 TFT 屏厂的制造业者或是系统端的客户能够依据功能需求快速地规划及完成其产品在 TFT 屏的显示应用，避免为了处理 TFT 显示画面进行冗长的程序开发。

## 1.1 LT776 基本介绍

LT776 是一款高效能 Uart TFT 串口屏控制芯片。其内部结合了 32bit Cortex M4 及 TFT 图形加速器的核心架构，主要的功能就是提供 Uart、USB 串口通讯，让主控端 MCU 透过简易的指令就能轻易的将要显示到 TFT 屏的内容传递给 TFT 驱动器 (Driver)，除了自带 32 位 MCU 之外，内部硬件还提供图形加速、PIP (Picture-in-Picture)、几何图形绘图等功能，能够提升 TFT 显示效率，及降低 MCU 处理图形显示所花费的时间，LT776 支持的 TFT 显示分辨率，可以由 320\*240 (QVGA) 到 1280\*1024 (SXGA)，显示屏则支持 16bits 的 RGB 接口。

LT776 内部的 32 位 MCU 主频可达 500MHz，含有直 256Kbytes Flash、128Kbytes SRAM，除了提供 Uart、USB 串口通讯，也提供一些模拟输入 AIN、PWM 及 INT 中断接口，这些接口也可以设置成普通 IO 接口，而为了达到多层次高分辨率的显示效果，LT776 内建 128Mb 显示内存，可以支持从每像素 1bit 的 2 灰阶到高达每像素 16bits 的 65K 颜色显示。LT776 也内建几何绘图引擎，支持画点、画线、画曲线、椭圆、三角形、矩形、圆角矩形等功能，同时内嵌的硬件图形加速引擎 (BTE) 提供了指令类型的图形操作，如显示旋转、画面镜射、画中画 (PIP/子母画面) 及图形混合透明显示等功能，若是配合 [乐升半导体](#) 的上、下位机软件更能发挥其显示效率，而不必为了 TFT 屏而去升级 MCU。LT776 强大的显示功能非常适合用在有 TFT-LCD 屏的电子产品上，或是原使用单色屏而想进行升级的产品，如各式智能家电、汽机车仪表盘、多功能事务机、工业控制、电子仪器、医疗设备、人机接口、检测设备等产品。下图为 LT776 的应用方块图：

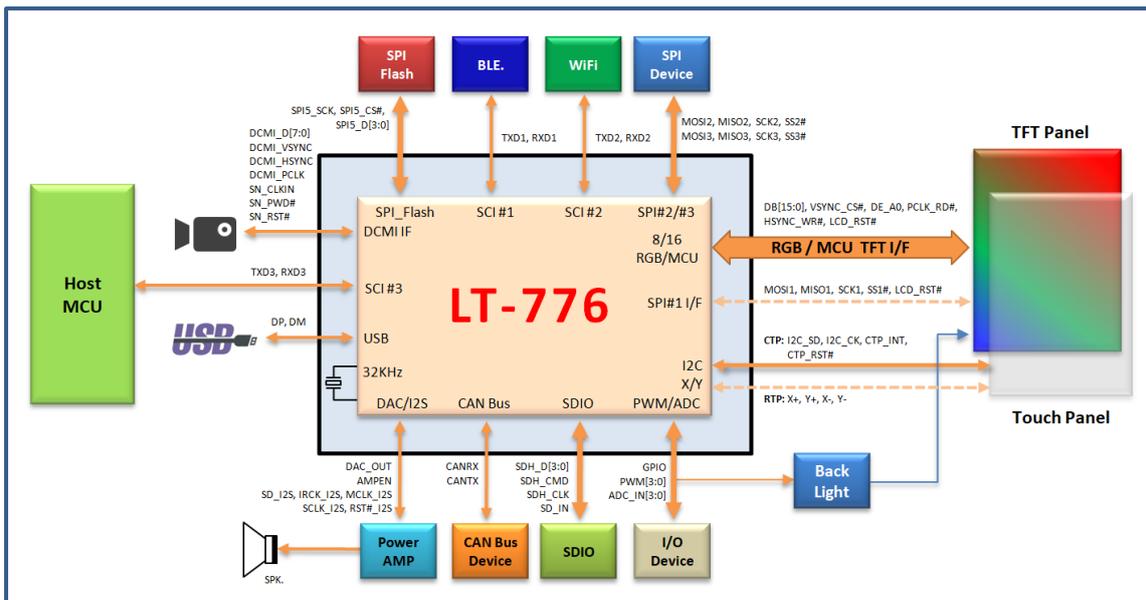


图 1-1: 应用方块图

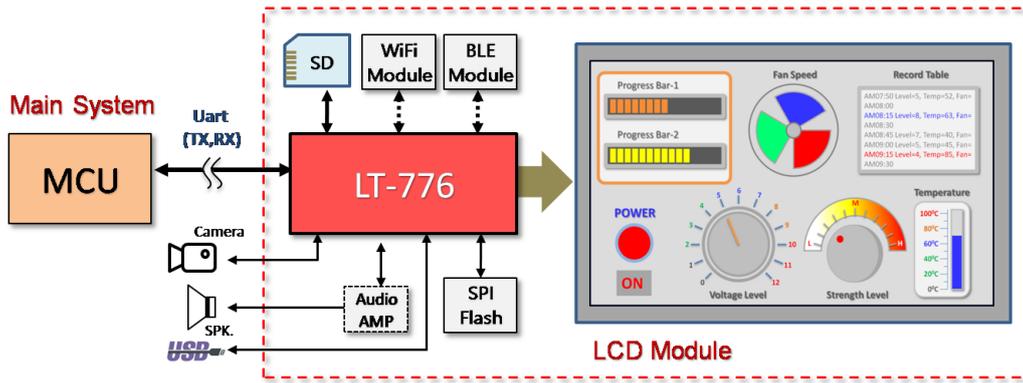


图 1-2: LT776 应用方式 - 1

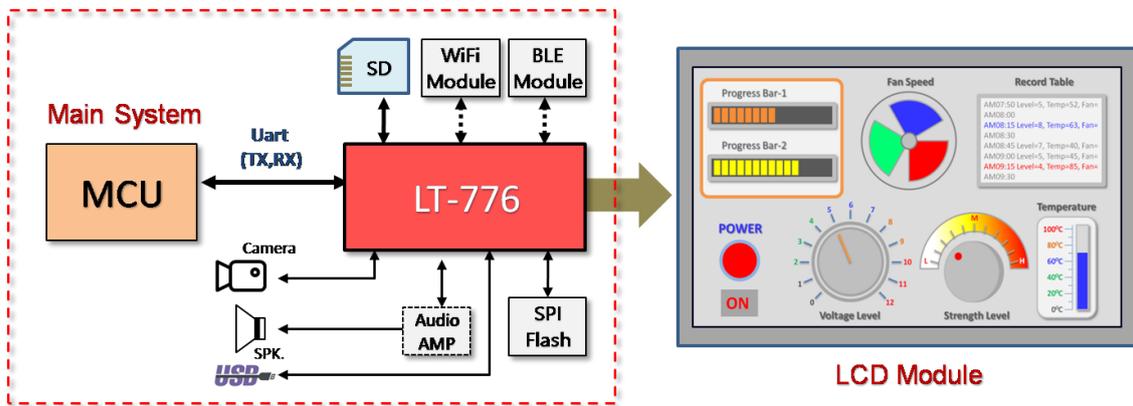


图 1-3: LT776 应用方式 - 2

LT776 的 TFT 串口屏还支持用 USB 接口更新，可以用 USB 接口对 LT776 内部核心主程序或是 SPI Flash 进行数据更新，请参考原理图及手册后面第 1.4 节说明。

## 1.2 TFT 串口屏的软硬件架构

所谓的 TFT 串口屏就是在 TFT 显示模块上加上 MCU 及 TFT 控制器, 该 MCU 负责接收主控端主板送来的串口 (Uart) 指令, 然后依据这些定义好的指令去显示出图片或是动画, 主控端主板上的 MCU 不需要为了繁琐的图片显示去编写复杂的程序, 因此 TFT 串口屏实际上就是一种指令屏的架构。

TFT 串口屏对主控端主要是透过 Uart (SCI) 接口来通讯, 如下图示意图。果要达到较远距离的通讯效果, 通常需要加上 RS232 或是 RS485 的专用驱动芯片, 可参考下一章图 2-2。

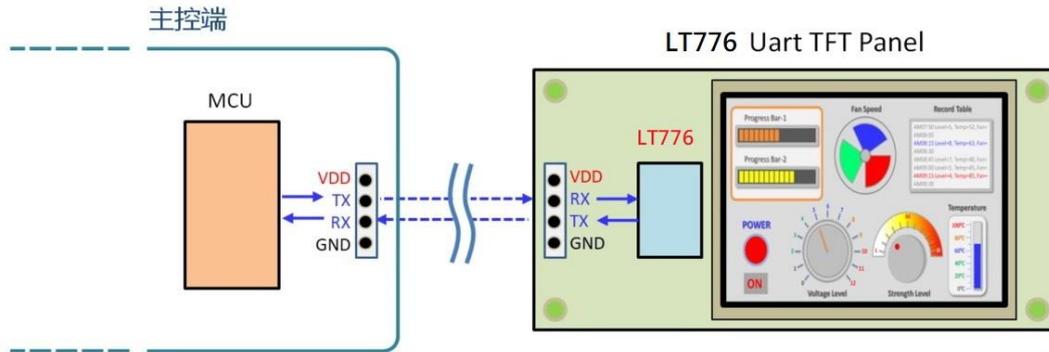


图 1-4: 主控端 MCU 的 Uart 与 LT776 串口屏的 Uart 连接示意图

在使用 LT776 串口屏之前必须要用上位机软件做开发, 乐升半导体提供了 **图文 UI 编辑器 (UI\_Editor.exe)** 上位机软件, 可以单独对 LT776 的 TFT 串口屏进行设置及显示功能的开发, 上位机软件开发时会使用到的图片、文字、动画等信息产生 Bin 档, 开发者可以透过 SD 卡、USB 碟, 或是专用的 SPI Flash 烧录器将 Bin 档烧录到 SPI Flash 内, 然后透过 USB 转 Uart (RS232) 的控制线对 TFT 串口屏进行模拟, 也就是做 TFT 屏显示画面的前期验证。



图 1-5: 使用上位机软件开发的示意图

上位机软件会依据图片出现的顺序及方式产生指令格式，而前面所说的模拟就是以电脑透过 USB 转 Uart 控制线替代主控端主控发出指令，让开发者在上位机软件上做前期验证，如果上位机软件发出的指令格式都能在 TFT 屏上显示及达到开发者所要的效果，那么主控端最终就在其 MCU 程序上植入这些指令格式，在想要显示图片时送出对应的指令。下图为主控板与 LT776 串口屏连接的示意图：

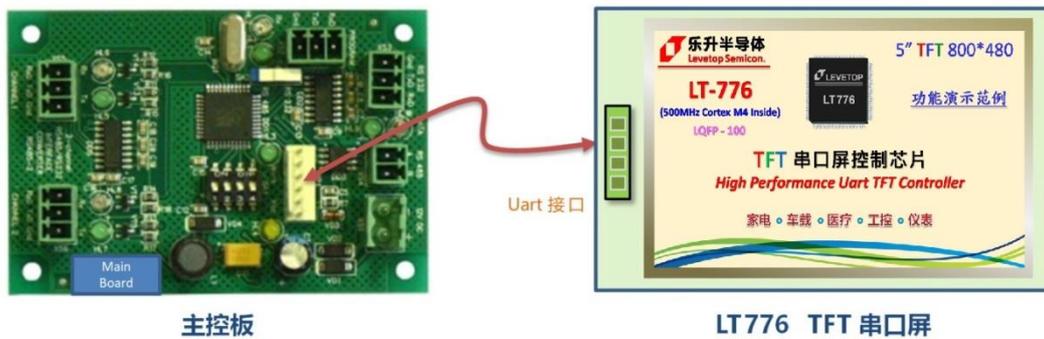


图 1-6: 主控板与 TFT 串口屏连接

每个 TFT 串口屏的显示动作都有一个固定的指令，例如 80h 就是显示图片的指令，UI\_Editor 会将使用的图片给予编号，在进行编译后会将所有图片、文字、动画等信息产生 Bin 檔，开发者将 Bin 檔烧录到 SPI Flash 内后，于验证的时候当电脑送出 80h、00h 那么 TFT 串口屏就会显示第一张图片，送出 80h、01h 就会显示第二张图片，当 UI\_Editor 发出的指令格式都能在 TFT 屏上显示及达到开发者所要的效果，就可以实地将主控端连接到 TFT 串口屏（如上图），而主控端 MCU 程序送出 0xAA(Start)、80h、00h、1Bh(CRC1)、98h(CRC2)、0xE4(End1)、0x1B(End2)、0x11(End3)、0xEE(End4) 指令后，TFT 串口屏就会显示第一张图片，同时回传信息 0xAA(Start)、80h、00h、00h、1Bh(CRC1)、98h(CRC2)、0xE4(End1)、0x1B(End2)、0x11(End3)、0xEE(End4) 给主控端，确认整个握手协议完成，如下图：

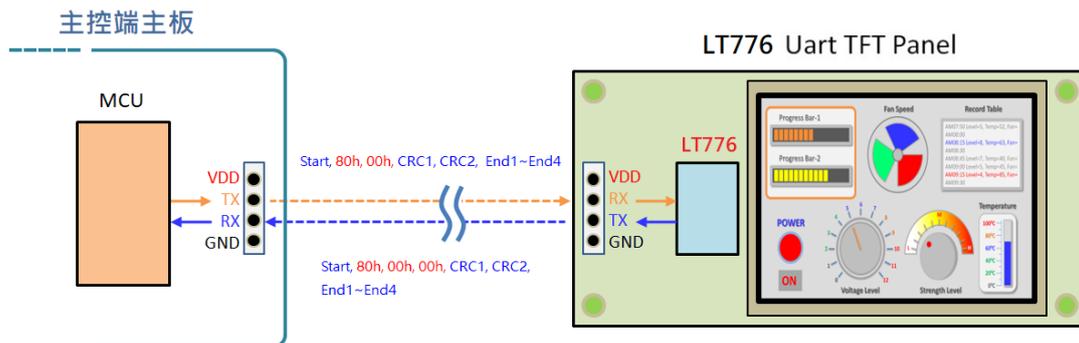


图 1-7: 串口屏的指令协议范例一

当主控端 MCU 程序送出 0xAA(Start)、80h、01h、0Bh(CRC1)、B9h(CRC2)、0xE4(End1)、0x1B(End2)、0x11(End3)、0xEE(End4) 指令后，TFT 串口屏就会显示第二张图片，同时回传信息 0xAA(Start)、80h、01h、00h、1Bh(CRC1)、98h(CRC2)、0xE4(End1)、0x1B(End2)、0x11(End3)、0xEE(End4) 给主控端，确认整个握手协议完成，如下图：

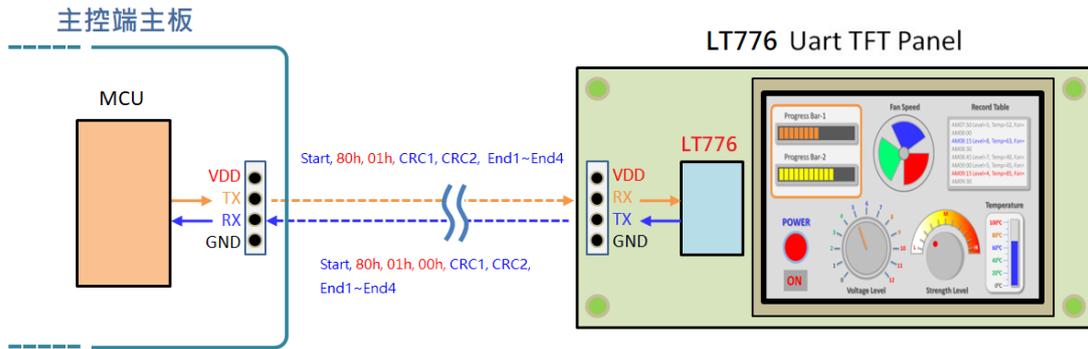


图 1-8: 串口屏的指令协议范例二

**提示:** 为了确保主控端与 TFT 串口屏之间的数据传递正确, 主控端 MCU 程序送出的指令还要加上 1 个 Byte 的 **起始码** (固定为 0xAA)、2 个 Byte 的 **CRC 码**、4 个 Byte 的 **结束码** (固定为 0xE4、0x1B、0x11、0xEE), 而 TFT 串口屏收到信息或是完成指令后会回传后信息给主控端的 MCU, 主控端与串口屏的指令握手协议表请参考手册后面第 3.2 节。有关 2 个字节 CRC 的产生方式请参考手册后面第 3.3 节。

而 LT776 的 TFT 串口屏还支持用 USB 接口更新, 可以用 USB 接口对 LT776 内部核心主程序或是 SPI Flash 进行数据更新, 请参考下一节的原理图及手册后面第 1.4 节说明。

### 1.3 TFT 串口屏的演示板(Demo Kit)

下图是 乐升半导体 的 LT776 TFT 串口屏演示板 (Demo Kit) , 可以搭配标准 40Pin 或是 50Pin TFT 屏。

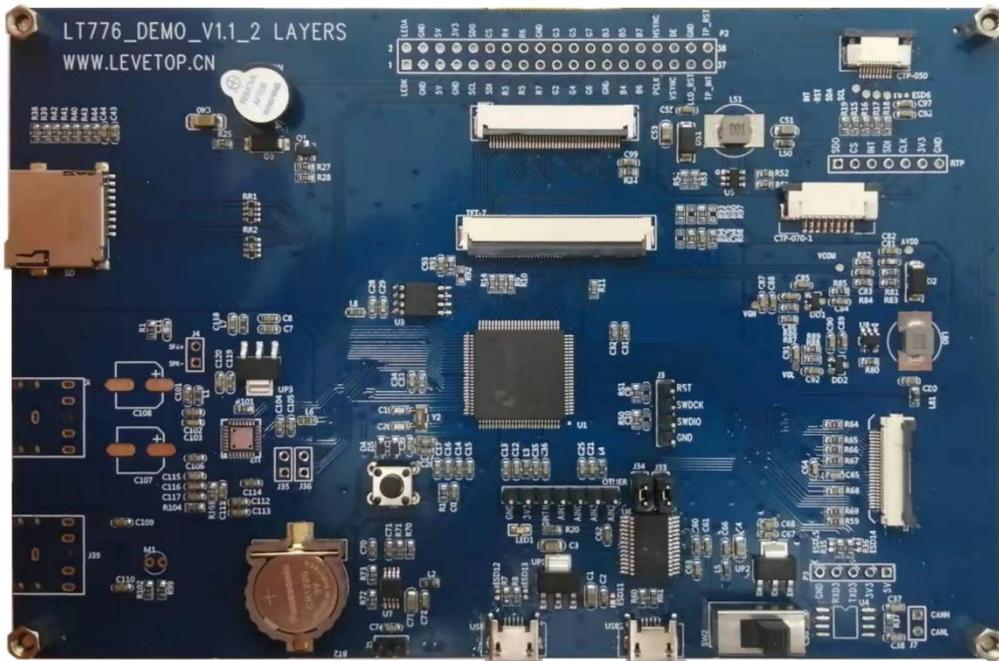


图 1-9: 串口屏的开发演示板 (Demo Kit)

LT776 内已经含有开发程序, 串口屏主板上的 SPI Flash (NOR Flash, 128Mbit) 也烧录有演示图片、GIF 动画、字库、Wav 音乐文件等所组合成的 Bin 文件, 相关 Demo 文件可以至 乐升半导体 网页 (<https://www.levetop.cn>) 的下载专区 “LT776-串口屏方案” 内下载, 或与业务人员联系。

本公司同时提供 LT776 串口屏的二次开发, 开放部分原代码, 详细模式请与业务人员联系。

### 1.4 TFT 串口屏的升级

一颗新的 LT776 IC，需要依次烧录 Bootloader、MCU\_Code、UartTFT\_Flash 等三个 bin 文件，才可正常工作，而 LT776 的 Bootloader 只能通过 SWD 软件烧录，MCU\_Code 和 UartTFT\_Flash 一般通过 SD 卡/U 盘烧录，下面会详细介绍这两种烧录方式。

首先需要通过软件 SWD 烧录 Bootloader，硬件上需要通过 ST-LINK V2 将 LT776 Demo 板与电脑相连 (SWCLK、SWDIO、GND)，然后在 SWD 上点击开始即可，如下图 1-10 所示。

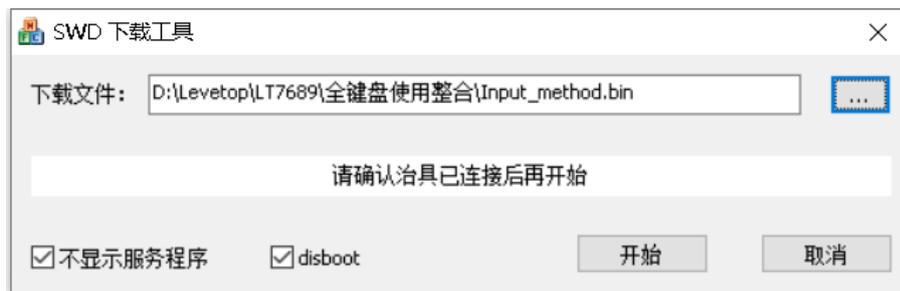


图 1-10: SWD 烧录

完成 Bootloader (SD 卡版/U 盘版) 烧录后，便可实现 SD 卡/U 盘烧录，具体的 Bootloader (SD 卡版/U 盘版) 文件可向我司工程师咨询索要。注意，烧录完成后一定要及时退出 SWD 软件，不然会影响 IC 正常工作。

MCU\_Code 和 UartTFT\_Flash 可通过 SD 卡或者 U 盘进行烧录，注意前提是已经烧录了相对应的 Bootloader (SD 卡版/U 盘版)。

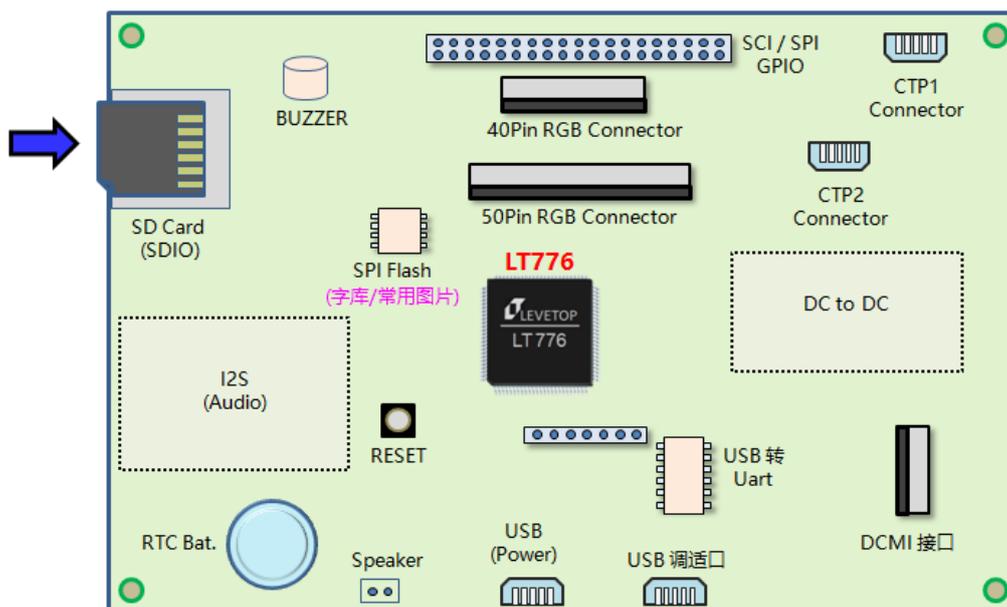


图 1-11: SD Card 插入 SD 卡槽内

LT776\_UartTFT\_AP Note\_CH / V2.0

LT776 透过 USB 盘进行更新 LT776 的 MCU 程序档案 (MCU\_Code.bin) , 或是 SPI Flash 的 Bin 檔 (UartTFT\_Flash.bin) 的速度比 SD 卡还要快, 更新方式与 SD 卡相同, 将需要更新的档案储存在 USB 盘及对应的目录内, 将 USB 盘插入 PCB 下方的 USB Disk 卡槽内后对串口屏进行开机或是复位即可自动进入更新模式。

先将 SD 卡/U 盘以 FAT32 方式格式化, 然后将更新的 MCU 程序档案 (MCU\_Code.bin) , 或是 SPI Flash 的 Bin 檔 (UartTFT\_Flash.bin, Userinfo.bin) 透过电脑写入到标准 SD 卡中固定的目录内 (参考后面图 1-13) , 当 TFT 串口屏为关机状态时再将 SD Card/U 盘插入 SD 卡槽/U 盘槽内, 注意如果是 U 盘烧录需要短接 OTG 与 GND。

SD Card 插入 SD 卡槽内后对串口屏进行开机, 此时串口屏会自动检测及 TFT 显示屏进入更新显示画面, 在更新完成后取出 SD Card, 然后重新启动 TFT 串口屏即可, 如下图所示为升级 MCU 程序码。

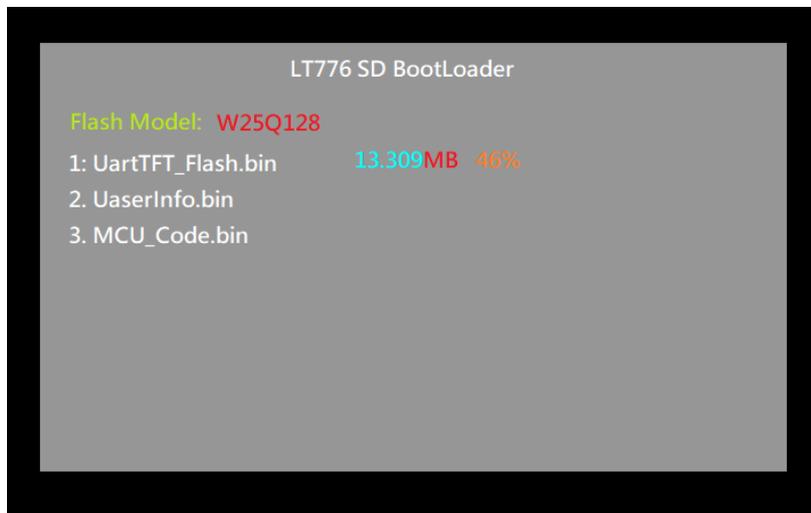


图 1-12: 以 SD Card 进行更新中

更新的 MCU 程序档案 (MCU\_Code.bin) , 或是 SPI Flash 的 Bin 檔 (UartTFT\_Flash.bin, Userinfo.bin) 必须储存在固定的目录里面, 也就是先在 SD 卡/U 盘内建立 MCU\_Code、UartTFT\_Flash、Userinfo 三个子目录, 然后将要更改的档案储存在该目录里面, 如下图所示。不想更新的 Bin 檔不要储存, 当 SD Card/U 盘插入 SD 卡槽/U 盘槽内后对串口屏进行开机, 串口屏会自动检测存在的档案然后自动进行更新。

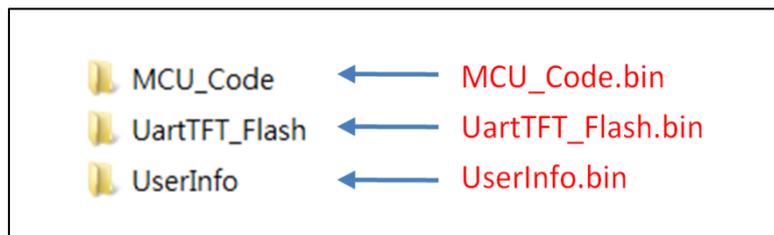


图 1-13: 更新的档案所在的储存目录

**注意 1:** 如果是使用我司的 LT776 Demo V1.3 的板子, 在进行 SD 卡升级的时候, 需要断开 TP 才能成功烧录, 如果产品上需要完整的 SD 卡功能, 要把 SPI3 复用接到 TP 和功放的信号断开, 需要修改的部分, 根据实际情况调整或向我司工程师咨询。

**注意 2:** 如果是更新 UartTFT\_Flash.bin, 那么通常 SPI Flash 会进行较长时间的擦除及写入动作, 因为这是 SPI Flash 本身的特性, 请耐心等待。而更新完后 LT776 内的 MCU 将自动启动, 但是建议使用者关机后取出 SD 卡/U 盘, 避免下次开机又进入自动更新状态

**注意 3:** 更新 Bin 档期间不得关闭电源或是做复位动作, 否则将可能造成串口屏 MCU 程序失效及无法再度更新。如果使用 SD 卡/U 盘无法更新或是不会进入图 1-12 的更新画面, 请与本公司联系。

## 2. TFT 串口屏硬件接口

本章主要详细介绍 LT776 TFT 串口屏的硬件架构，包括对外的各种接口如 RS232、SPI Flash 及 RGB 的 TFT 屏，还有其它常用组件如 USB 接口、触控功能、WiFi 模块、蓝牙模块、RTC、蜂鸣器等等，设计者可以依应用需求去考虑是否加上。

### 2.1 Uart 串口与 RS232/RS485 接口

TFT 串口屏对主控端主要是透过 Uart 串口来通讯(TTL 电平), 如果距离长可以加上 RS232 或是 RS485 驱动芯片来通讯，如下示意图。

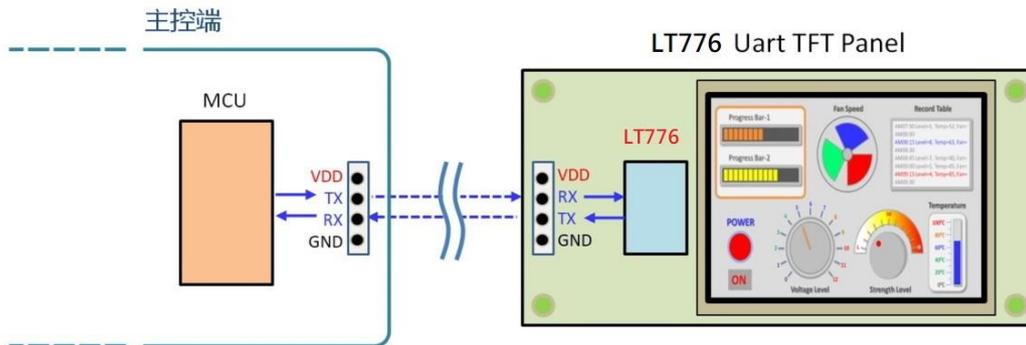


图 2-1: 主控端 MCU 透过 Uart 串口与 LT776 串口屏通讯

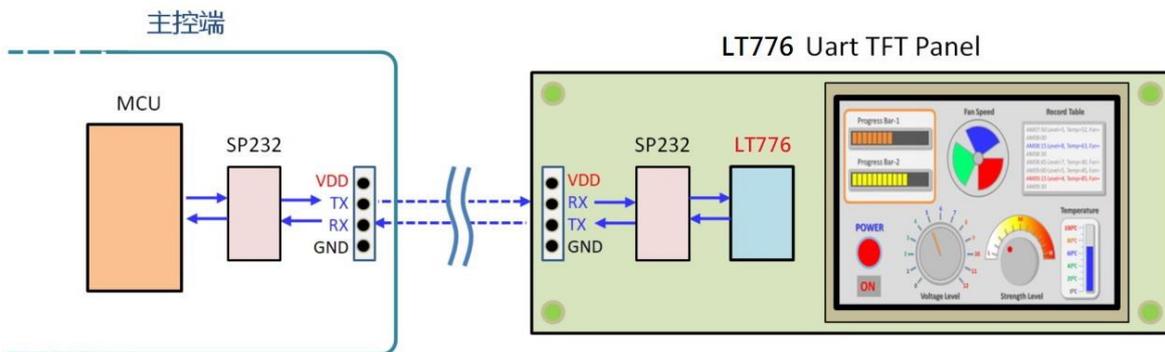


图 2-2: 加上 RS232 延伸通讯距离

## 2.2 TFT 屏 RGB/LVDS/MIPI 接口

LT776 支持 16bits 的 RGB 接口面板, 可以通过这些 RGB 接口将数据信号, 还有连同控制信号 PCLK、DE、HSYNC、VSYNC 等送到 TFT 面板上的驱动器, 如下图的接口示意图:

下图为 LT776 与 TFT 屏之间采 16 位 RGB (RGB 5:6:5) 接口示意图。

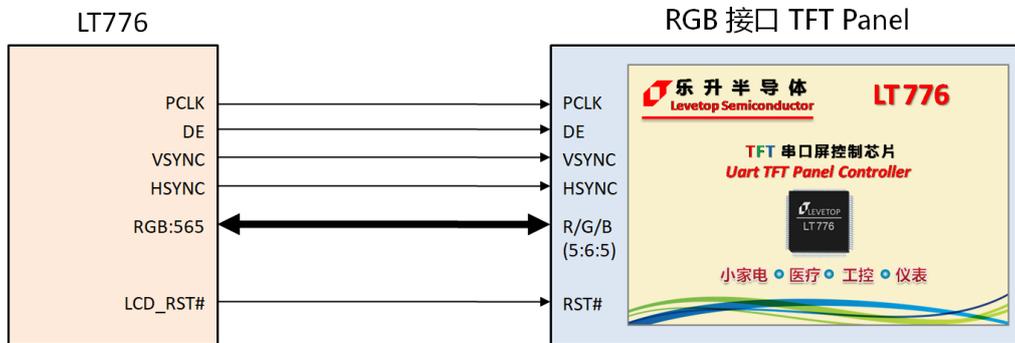


图 2-3: LT776 与 TFT 屏之间采 16 位 RGB 接口示意图

如果使用 LVDS 接口的 TFT 屏, 只要透过 RGB 转 LVDS 芯片, 就能轻易的转换, 也不需要透过任何软件修改, 目前市面上有许多价格低廉的 RGB 转 LVDS 芯片, 如 THC63LVDM83D 等, 下图为 RGB 转 LVDS 参考原理图。

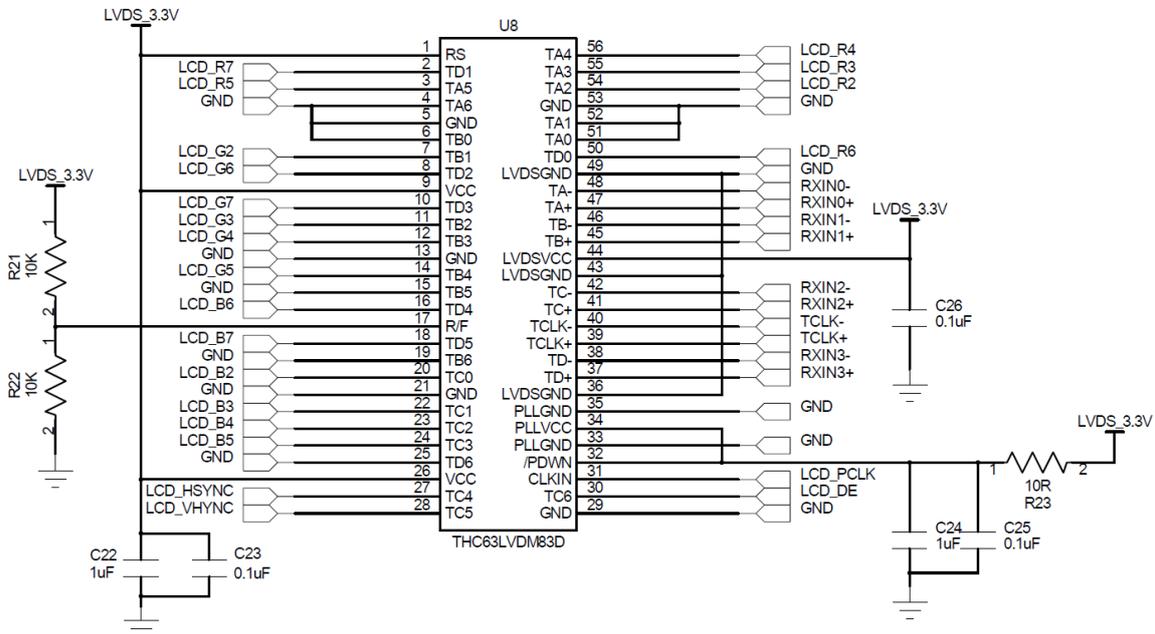


图 2-4: RGB 转 LVDS 示意图

如果使用 MIPI 接口的 TFT 屏, 则需要透过 RGB 转 MIPI 芯片, 但是必须分别对该 MIPI 转换芯片及 MIPI 的 TFT 屏做初始化的程序设定, 会比 RGB 转 LVDS 更麻烦一些, 目前市面上有许多的 RGB 转 MIPI 芯片, 如 SSD2828 等, 参考原理图可以与本公司联系或参考生产厂家数据。

### 2.3 PWM 控制背光

LT776 内建多个 PWM 输出, 其中 TFT 图形加速器提供 2 个 PWM 输出: TFT\_PWM0、TFT\_PWM1。用户可以藉由 PWM 产生的信号控制外部供给 TFT 屏的背光升压电路, 还可以透过 LT776 的一个 GPIO 口控制 TFT 显示 On/Off, 如下图的示意图:

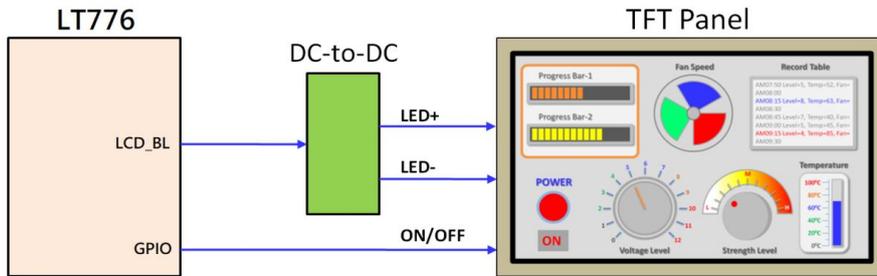


图 2-5: LT776 用 TFT\_PWM1 控制背光及 GPIO 控制显示 On/Off

### 2.4 SPI Flash 接口

SPI Flash 是用于储存编程后的指令执行信息、上/下位机设定的参数、及在上位机编程时选用到的图片、字库等, 使用者使用上位机软件或是专属编译程序进行显示功能的开发及编译完成后可以透过 SD 卡或是 SPI Flash 烧录器, 将整合的 Bin 文件烧录到 SPI Flash 内, SPI Flash 的容量需求取决于使用到的图片、字库等数据量的大小, 一般是使用 NOR Type 的 SPI Flash, 如果容量大甚至可以使用 NAND Type 的 SPI Flash。下图为 LT776 与 SPI Flash 连接的示意图:

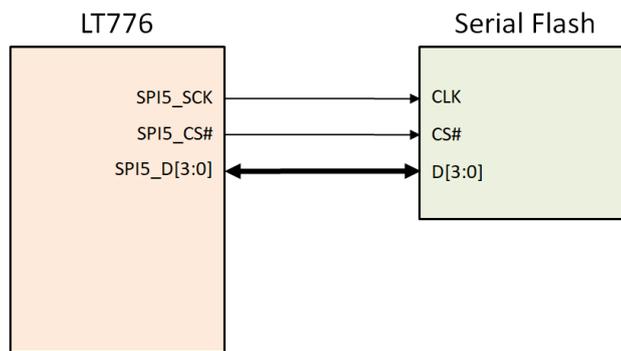


图 2-6: LT776 与 SPI Flash 接口示意图

当使用 UI\_Editor 软件开发时, 会产生供给 SPI Flash 的 2 个 Bin 档案: UartTFT\_Flash.bin 檔和 UserInfo.bin 档, 其中 UartTFT\_Flash.bin 檔实际包括了 UserInfo.bin 档案的内容, 而 UserInfo.bin 档是存放 Command 参数, 其他区域则是示图片、动画文件、文字库、声音文件等数据, 如果要烧录 UartTFT\_Flash.bin 檔, 可以使用 SD 卡或是如下专用的 SPI Flash 烧录器, 如下图, 连接方式请参考第 4.7.1 节, 而 SPI Flash 烧录器的烧录软件可以自本公司网站的“LT776-串口屏方案”下载专区下载。



图 2-7: SPI Flash 烧录器

## 2.5 USB 接口

LT776 TFT 串口屏如果需要使用 USB 功能，如下图的接口示意图。USB 接口也可以用来作为更新 LT776 MCU 程序档案 (MCU\_Code.bin)，或是 SPI Flash 的 Bin 档。

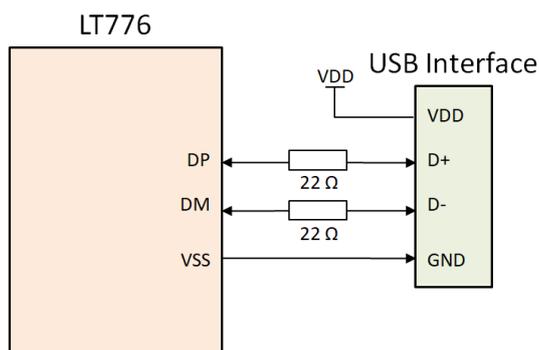


图 2-8: MCU 与 USB 的接口示意图

## 2.1 SDIO 接口

LT776 提供 SDIO 接口，支持 SD 卡的 SDIO 模式，参考原理图如下：

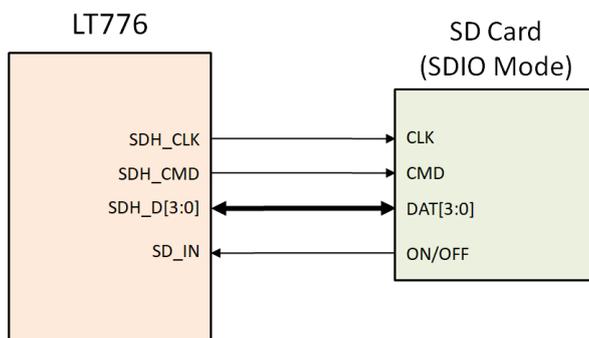


图 2-9: SDIO 接口参考原理图

## 2.2 PWM 控制蜂鸣器/喇叭

某些 TFT 串口屏在应用上需要蜂鸣器或是喇叭作为音效提示功能，设计者可以使用 PWM 输出或是 DAC\_OUT 或是来产生音频信号，再由三级管或是功率放大器推动喇叭，如下图的示意图：

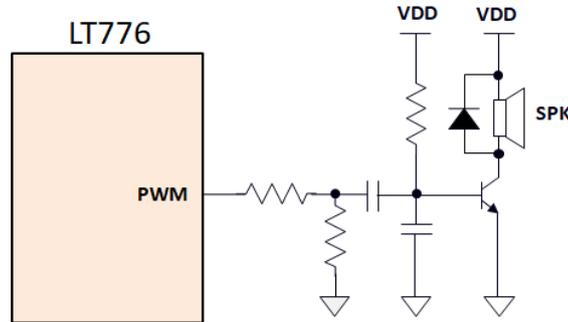


图 2-10: 用 LT776 的 PWM 输出控制蜂鸣器/喇叭

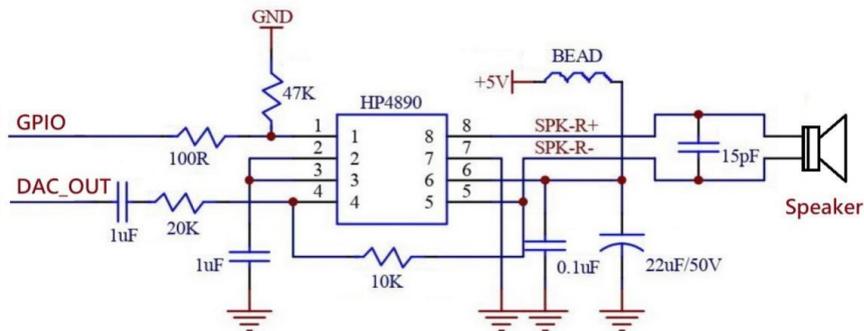


图 2-11: 用 LT776 的 DAC\_OUT 输出及小功率放大器推动喇叭

## 2.3 SCI 通讯接口

如果 TFT 串口屏需要支持 WiFi 模块、蓝牙模块，可以使用 LT776 的 SCI 信号接口，如下图的接口示意图：

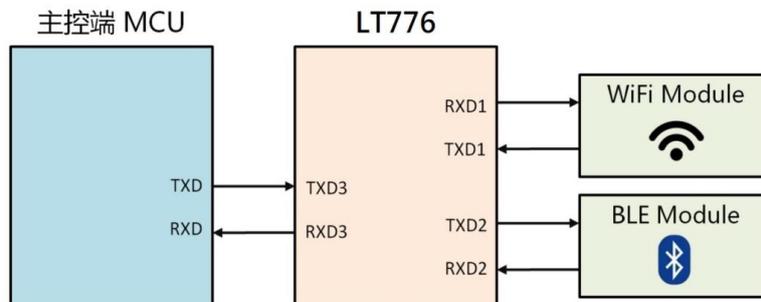


图 2-12: LT776 与 WiFi 模块、蓝牙模块的接口示意图

## 2.4 触控接口

许多 TFT 串口屏在应用上需要触控功能，电容式触控 IC 通常是使用 I2C 接口，下面是电容式触控 IC 与 LT776 TFT 串口屏的接口示意图：

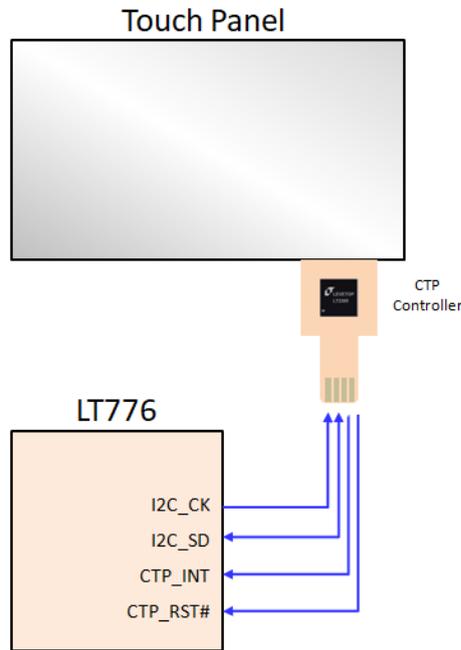


图 2-13: LT776 与电容式触控 IC 接口示意图

## 2.5 I2S 接口

LT776 除了提供一个由 DAC 产生的声音输出接口 (DAC\_OUT) 外，还提供一组 I2S 接口，可以连接到音频译码器 CODEC 作为声音拨放，其参考原理图如下：

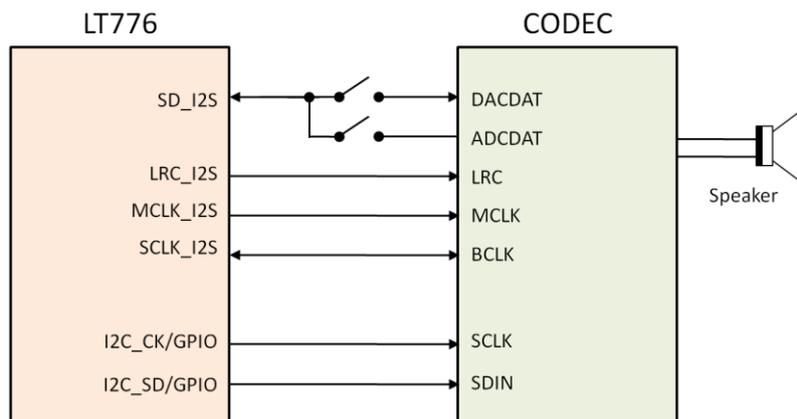


图 2-14: I2S 连接 CODEC 参考原理图

## 2.6 DCMI 摄像头接口

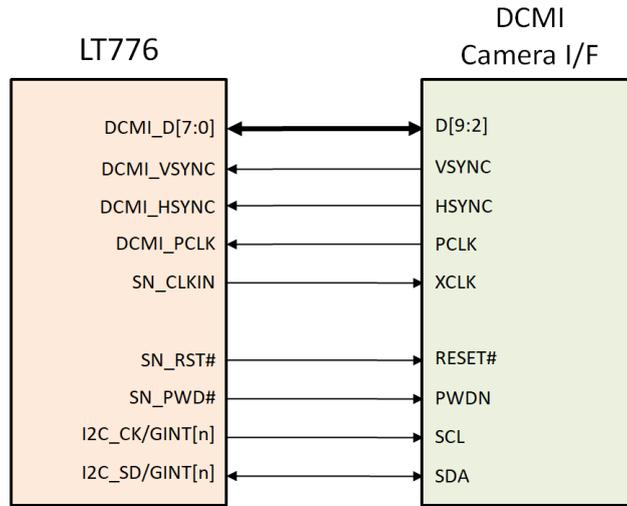


图 2-15: DCMI 连接摄像头参考原理图

## 2.7 时钟信号与 RTC (Real Time Clock)

LT776 内部有时钟源，供给内部系统时钟信号，外部不需要使用其他晶振电路，而内部最高运行时钟频率为 150MHz。如果使用内建的 RTC (Real Time Clock)，则需要提供一 32.768KHz 的晶振电路，如果要想 RTC 于断电时持续保持工作，必须在 VBAT 脚必须接上 3.3V~3.6V 电池电源。

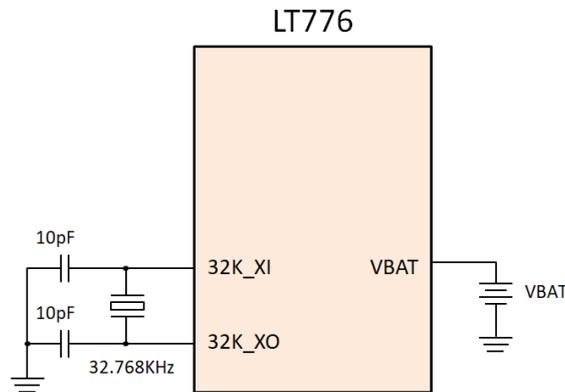


图 2-16: LT776 时钟信号结构图

## 2.8 TFT 串口屏参考原理图

乐升半导体 提供针对 LT776 的 TFT 串口屏原理图，如果需要，使用者可以自网站 <https://www.levetop.cn> 下载电子文件或是与我们业务、FAE 部门联系取得。

在本公司网站下载的 TFT 串口屏原理图，设计者可以依终端客户的应用需求去删减不需要的电路，以节省成本。下图为 LT776 的 TFT 串口屏开发板 (Demo Kit)：



图 2-17: LT776 TFT 串口屏开发板 (Demo Kit)

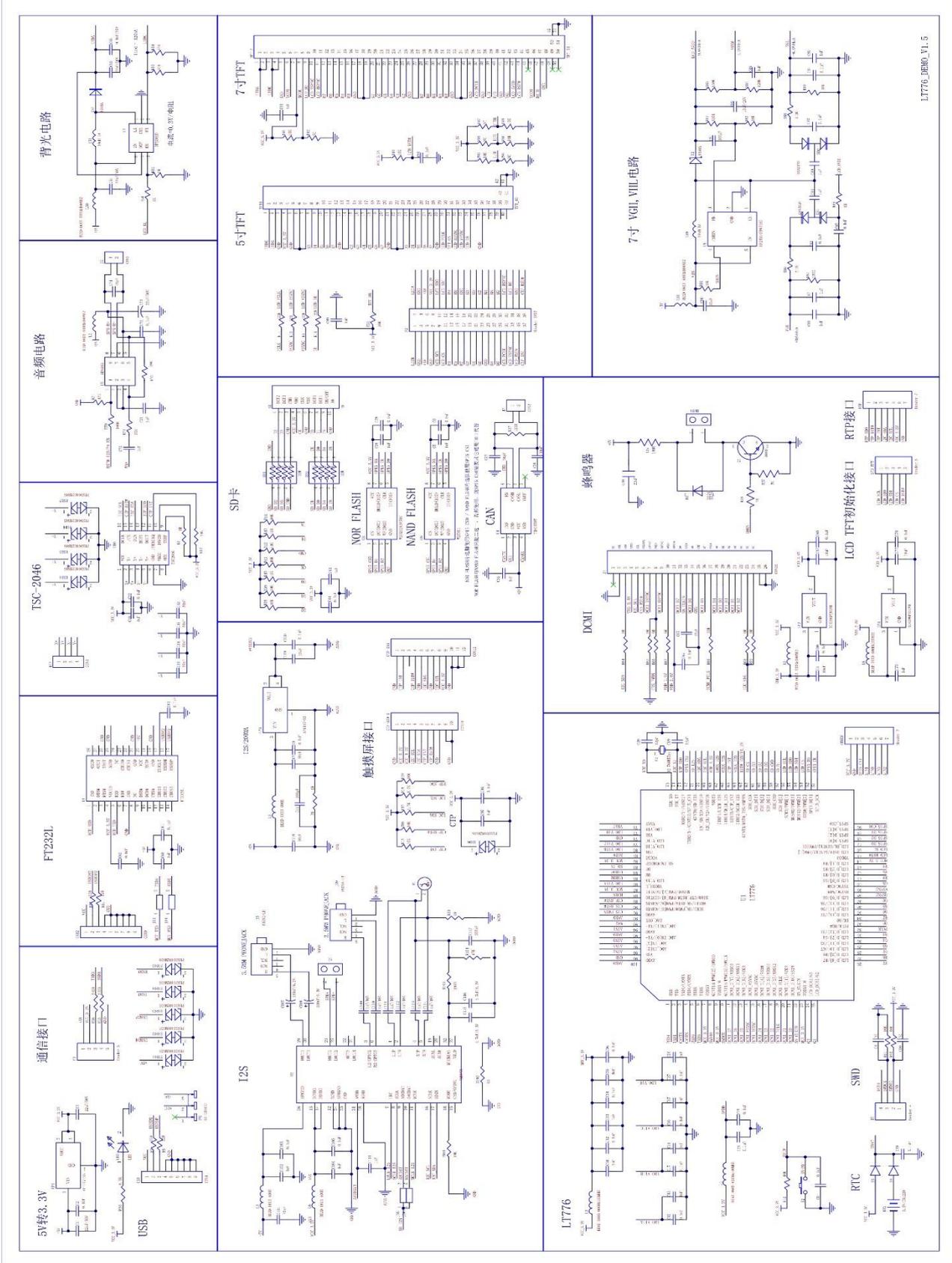


图 2-18: LT776 TFT 串口屏原理图

LT776\_UartTFT\_AP Note\_CH / V2.0

## 2.9 EMC 干扰与抗干扰对策

TFT 串口屏在许多应用上会遇到 EMC (EMI/EMS) 干扰问题或是认证问题, 可以考虑以下几种建议进行改善及补强:

- 注意电源滤波电路的布局 (PCB Layout), 电源电路上采用质量较佳的元器件 (如滤波电容)。
- 产生 3.3V 的 DC to DC 电源输入端、输出端除了原有的滤波电容外, 再加上扼流圈 (磁珠) 及 (如图 2-19、图 2-20)。
- 产生 TFT 屏背光的 DC to DC 电源输入端、输出端除了原有的滤波电容外, 再加上扼流圈 (磁珠) 及 (如图 2-21、图 2-22)。
- TFT 屏的 FPC 做包覆处理 (如图 2-23)。
- LT776 RGB 输出加串接电阻。
- PCLK、PDE、HSYNC、VHYNCR 输出加串接电阻及接地电容 (如图 2-24)。
- 增加 ESD 保护组件。
- LT776 及 MCU 加上金属罩接地。
- EFT (Electrical Fast Transient) 电快速瞬变脉冲群的电源处理: 正负两端都要用电感隔离。如图 2-25)。

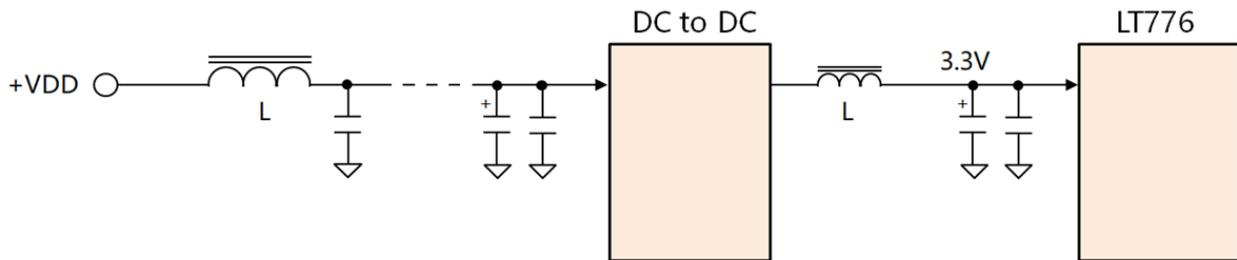


图 2-19: 3.3V 的 DC to DC 电源输入、输出端加上 LC 电路

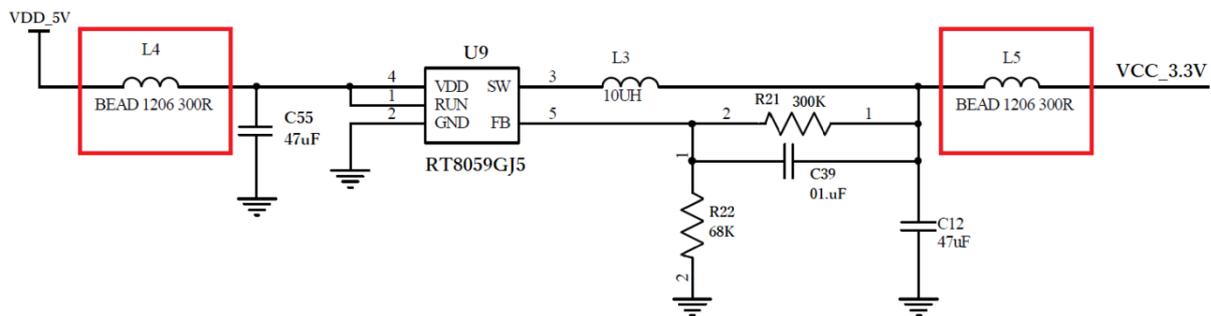


图 2-20: 3.3V 的 DC to DC 电源电路范例

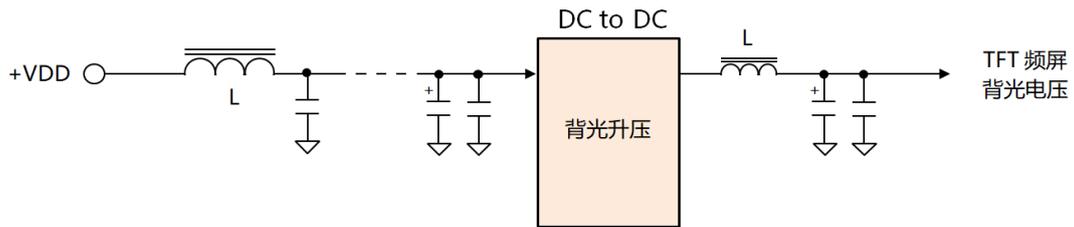


图 2-21: TFT 屏背光的 DC to DC 电源输入、输出端加上 LC 电路

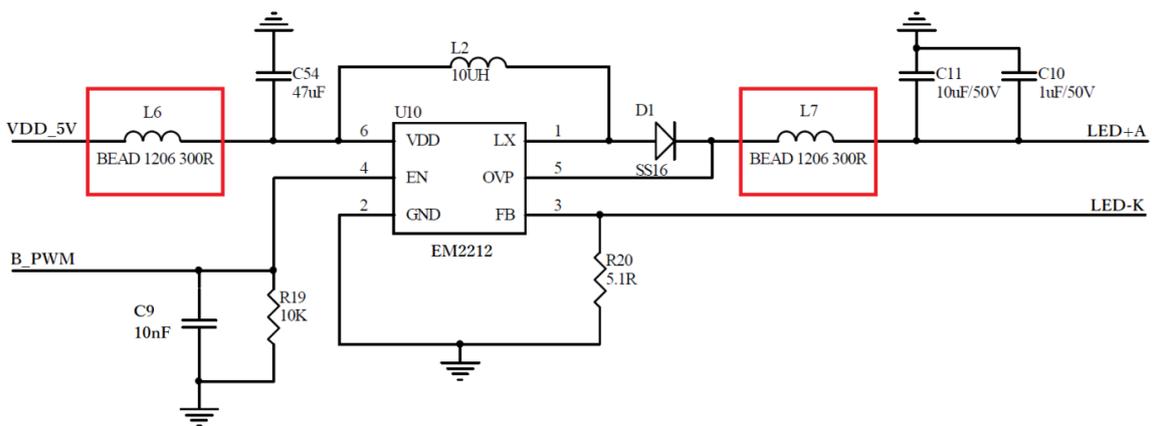


图 2-22: TFT 屏背光的 DC to DC 电源电路范例

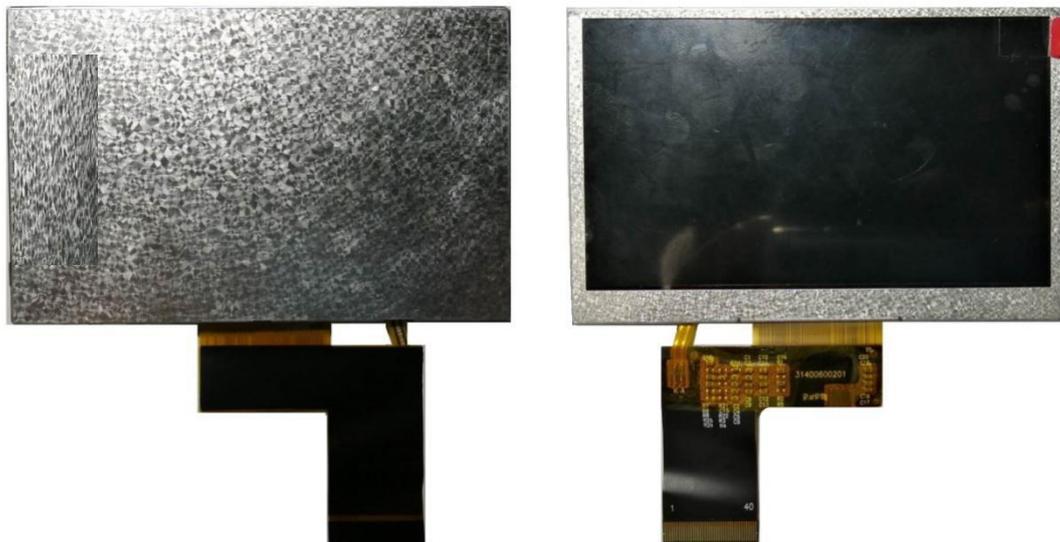


图 2-23: TFT 屏的 FPC 做包覆处理

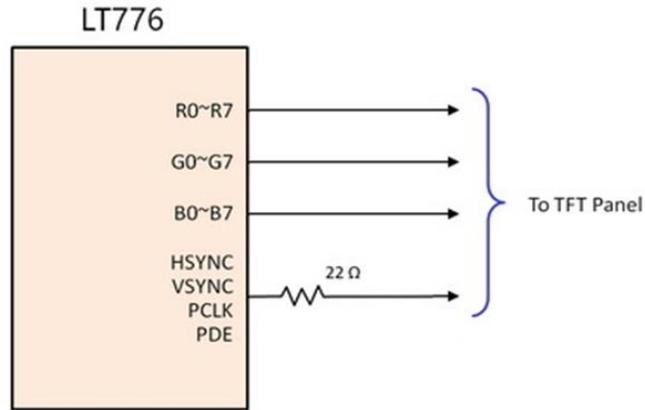


图 2-24: RGB 等信号输出加串接电阻及接地电容

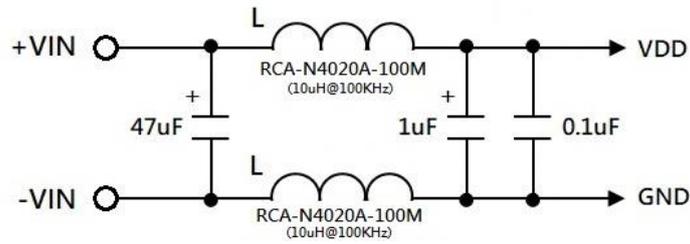


图 2-25: 降低 EFT 干扰的参考电路

### 3. 串口指令

为了让主控端的系统或是主板就能够透过 UART、SPI 等串口轻易的在 TFT 屏上显示图片或是信息，在 LT776 TFT 串口屏上规划了一个串口指令集，透过定义好的指令码配合指令参数去改变 TFT 屏上的画面，乐升半导体 提供串口屏的开发工具：[图文 UI 编辑器 \(UI\\_Editor.exe\)](#)，用户可以进行 TFT 显示屏的方案开发，同时这开发工具都可以实时模拟 TFT 显示屏的显示效果，做前期的验证。

#### 3.1 串口屏指令集

LT776 支持的 TFT 串口屏指令，包括图片静态显示、图片动态显示、文字显示、几何图形等等，如下表所示。

表格 3-1: LT776 串口屏指令集

主功能	细项功能	指令码
显示图片	单张/多张图片	80h, 85h, 8Ah, 8Fh,
	循环拨放	81h, 84h
	显示透明图片	82h
	GIF 动画	88h, 89h
	设定显示缓冲区	8Eh
	弹出图片	D8h
	循环卷动	D9h, DBh
	数字图片	90h, 91h
	显示 ASCII 码图片	9Fh
显示控件图片	控件滑条	94h, 95h
	环形触控滑条	96h, 97h
	显示单一控件图片	A0h, A1h
	虚拟控件	A2h, A3h
	全屏滑动图片	B4h
	手势滑动	C9h
	取消控件触控功能	9Eh
	显示底图及控件图片	9Bh, 9Ch
指标与造图	进度条指标图	B0h
	指针指标图	B1h
	环形指标图	DCh
	二维码生成	98h
	曲线显示	B2, B3h
显示字库	字库-1~4	C0h, C1h, C2h, C3h
	大字库-1~4	D0h, D1h, D2h, D3h

LT776\_UartTFT\_AP Note\_CH / V2.0

主功能	细项功能	指令码
背光亮度	设置亮度	BAh
	On/Off	BCh
设定时钟	设定时钟	8Ch
	读取时钟	8Dh
显示时钟/星期	显示数字时钟	92h
	显示模拟时钟	93h
	显示星期	9Dh
摄像头	显示摄像头画面	AEh, AFh
开机动作与画面	执行开机指令	9Ah/00
指令合并	执行组合指令	9Ah
Wav 音乐文件	播放 Wav 檔	B8h
	停止播放	B9h
几何图形	画点	DFh
	直线	E0h
	空心圆形	E1h
	实心圆形	E2h
	带框实心圆形	E3h
	空心椭圆	E4h
	实心椭圆形	E5h
	带框实心椭圆	E6h
	空心矩形	E7h
	实心矩形	E8h
	带框矩形	E9h
	空心圆角矩形	EAh
	实心圆角矩形	EBh
	带框圆角矩形	ECh
	空心三角形	EDh
	实心三角形	EEh
	带框三角形	EFh
	圆柱体	F4h
表格视窗	F6h	
寄存器控制	寄存器控制指令	CAh~CFh
电阻屏校验	电阻屏校验指令	8Bh
TFT 屏复位	对 LT776 复位	BDh
串口屏侦测	联机检查	BEh
	版本侦测	BFh

### 3.2 主控端与 TFT 串口屏协议表

表格 3-2: 主控端与 TFT 串口屏协议表

主功能	细项功能	主控端发送 (TFT 串口屏接收)						主控端接收 (TFT 串口屏发送)					
		起始码 (1Bytes)	指令码 (1Byte)	序号 (1Byte)	指令参数	CRC 码 (2Bytes)	结束码 (4Bytes)	起始码 (1Bytes)	指令码 (1Byte)	序号 (1Byte)	信息码/ 反馈码 (1Bytes)	CRC 码 (2Bytes)	结束码 (4Bytes)
显示图片	显示单张/ 多张图片	Start	80h	nn		CRC	End	Start	80h	nn	信息码	CRC	End
	取消显示单 张/多张图片	Start	85h	nn		CRC	End	Start	84h	nn	信息码	CRC	End
	显示单张/ 多张图片	Start	8Ah	nn		CRC	End	Start	8Ah	nn	信息码	CRC	End
	显示单张 图片	Start	8Fh	nn	X, Y, PNG, Pnn	CRC	End	Start	8Fh	nn	信息码	CRC	End
	循环播放	Start	81h	nn		CRC	End	Start	81h	nn	信息码	CRC	End
	取消循环 播放	Start	84h	nn		CRC	End	Start	84h	nn	信息码	CRC	End
	透明图片	Start	82h	nn		CRC	End	Start	82h	nn	信息码	CRC	End
	显示 GIF 动画	Start	88h	nn		CRC	End	Start	88h	nn	信息码	CRC	End
	取消 GIF 动画	Start	89h	nn		CRC	End	Start	89h	nn	信息码	CRC	End
	设定缓冲区	Start	8Eh		0, 1	CRC	End	Start	8Eh	00	信息码	CRC	End
	弹出图片	Start	D8h	nn		CRC	End	Start	D8h	nn	信息码	CRC	End
	循环卷动	Start	D9h	nn		CRC	End	Start	D9h	nn	信息码	CRC	End
	取消循环卷 动	Start	DBh	nn		CRC	End	Start	DBh	nn	信息码	CRC	End
	数字图片	Start	90h	nn	ddd.d	CRC	End	Start	90h	nn	信息码	CRC	End
	真彩数字图 片	Start	91h	nn	ddd.d	CRC	End	Start	91h	nn	信息码	CRC	End
	显示 ASCII 码 图片	Start	9Fh	nn	ddd.d	CRC	End	Start	9Fh	nn	信息码	CRC	End
显示控件图片	全屏滑动 图片	Start	B4h	nn		CRC	End	Start	B4h	nn	信息码	CRC	End
	显示单一控 件图片	Start	A0h	nn		CRC	End	Start	A0h	nn	信息码	CRC	End
		按下控件图片时						Start	A0h	nn	31h	CRC	End
	放开控件图片时						Start	A0h	nn	30h	CRC	End	
	取消单一控 件图片	Start	A1h	nn		CRC	End	Start	A1h	nn	信息码	CRC	End
	虚拟控件	Start	A2h	nn		CRC	End	Start	A2h	nn	信息码	CRC	End
		按下控件区域时						Start	A2h	nn	31h	CRC	End
		放开控件区域时						Start	A2h	nn	30h	CRC	End
	取消虚拟 控件	Start	A3h	nn		CRC	End	Start	A3h	nn	信息码	CRC	End
	手势滑动	Start	C9h	nn		CRC	End	Start	C9h	nn	Dir+信息 码	CRC	End
取消控件触 控功能	Start	9Eh		Code (1Byte)	CRC	End	Start	9Eh	Code (1Byte)	信息码	CRC	End	

主功能	细项功能	主控端发送 (TFT 串口屏接收)						主控端接收 (TFT 串口屏发送)					
		起始码 (1Bytes)	指令码 (1Byte)	序号 (1Byte)	指令参数	CRC 码 (2Bytes)	结束码 (4Bytes)	起始码 (1Bytes)	指令码 (1Byte)	序号 (1Byte)	信息码/ 反馈码 (1Bytes)	CRC 码 (2Bytes)	结束码 (4Bytes)
显示底图 及所有控 件图片	Start	9Ch	00		CRC	End	Start	9Ch	00	信息码	CRC	End	
	屏幕滑动后						Start	9Ch	页号	信息码	CRC	Start	
	按下控件图片时						Start	98h	图标 ID 号	31h	CRC	End	
	放开控件图片时						Start	98h	图标 ID 号	30h	CRC	End	
指标与 造图	进度条 指标图	Start	B0h	nn	Value (2 Bytes)	CRC	End	Start	B0h	nn	信息码	CRC	End
	指针指标图	Start	B1h	nn	Angle (2 Bytes)	CRC	End	Start	B1h	nn	信息码	CRC	End
	环形指标图	Start	DCh	nn	S_Angle, A_Angle	CRC	End	Start	DCh	nn	信息码	CRC	End
	二维码生成	Start	98h	nn	字符串	CRC	End	Start	98h	nn	信息码	CRC	End
	曲线缓冲	Start	B2h	Chanel (0~3)	Data (n*2bytes)	CRC	End	Start	B2h	Chanel (0~3)	信息码	CRC	End
	显示曲线	Start	B3h	nn		CRC	End	Start	B3h	nn	信息码	CRC	End
触控滑 条控制	设置触控 滑条	Start	94h	nn		CRC	End	Start	94h	nn	信息码	CRC	End
	触控滑条被按下时						Start	94h	nn	Value (1 Byte)	CRC	End	
	移除触控 滑条	Start	95h	nn		CRC	End	Start	95h	nn	信息码	CRC	End
	设置环形 触控滑条	Start	96h	nn		CRC	End	Start	96h	nn	信息码	CRC	End
	环形触控滑条被按下时						Start	96h	nn	Value (1 Byte)	CRC	End	
显示字 串	字库-1	Start	C0h	nn	字符串	CRC	End	Start	C0h	nn	信息码	CRC	End
	字库-2	Start	C1h	nn	字符串	CRC	End	Start	C1h	nn	信息码	CRC	End
	字库-3	Start	C2h	nn	字符串	CRC	End	Start	C2h	nn	信息码	CRC	End
	字库-4	Start	C3h	nn	字符串	CRC	End	Start	C3h	nn	信息码	CRC	End
	大字库-1	Start	D0h	nn	字符串	CRC	End	Start	D0h	nn	信息码	CRC	End
	大字库-2	Start	D1h	nn	字符串	CRC	End	Start	D1h	nn	信息码	CRC	End
	大字库-3	Start	D2h	nn	字符串	CRC	End	Start	D2h	nn	信息码	CRC	End
	大字库-4	Start	D3h	nn	字符串	CRC	End	Start	D3h	nn	信息码	CRC	End
背光 亮度	设置亮度	Start	BAh		BL (00~0Fh)	CRC	End	Start	BAh	BL (00~0Fh)	信息码	CRC	End
	On/Off	Start	BCh		00,01,02	CRC	End	Start	BCh	00,01,02	信息码	CRC	End
Wav 檔	播放	Start	B8h		REP(Bit7) + WAV 编 号	CRC	End	Start	B8h	REP(Bit7) + WAV 编 号	信息码	CRC	End
	停止	Start	B9h			CRC	End	Start	B9h	00	信息码	CRC	End
开机 指令	开机指令	Start	9Ah	00		CRC	End	Start	9Ah	00	信息码	CRC	End

主功能	细项功能	主控端发送 (TFT 串口屏接收)						主控端接收 (TFT 串口屏发送)					
		起始码 (1Bytes)	指令码 (1Byte)	序号 (1Byte)	指令参数	CRC 码 (2Bytes)	结束码 (4Bytes)	起始码 (1Bytes)	指令码 (1Byte)	序号 (1Byte)	信息码/ 反馈码 (1Bytes)	CRC 码 (2Bytes)	结束码 (4Bytes)
合并指令	执行合并指令	Start	9Ah	nn		CRC	End	Start	9Ah	nn	信息码	CRC	End
阻屏校验	电阻屏校验	Start	8Bh			CRC	End	Start	8Bh	00	信息码	CRC	End
TFT 屏复位	对 LT776 进行复位	Start	BDh			CRC	End	Start	BDh	00	信息码	CRC	End
几何图形	画点	Start	DFh	nn	X,Y	CRC	End	Start	DFh	nn	信息码	CRC	End
	直线	Start	E0h	nn		CRC	End	Start	E0h	nn	信息码	CRC	End
	空心圆形	Start	E1h	nn		CRC	End	Start	E1h	nn	信息码	CRC	End
	实心圆形	Start	E2h	nn		CRC	End	Start	E2h	nn	信息码	CRC	End
	带框实心圆形	Start	E3h	nn		CRC	End	Start	E3h	nn	信息码	CRC	End
	空心椭圆	Start	E4h	nn		CRC	End	Start	E4h	nn	信息码	CRC	End
	实心椭圆	Start	E5h	nn		CRC	End	Start	E5h	nn	信息码	CRC	End
	带框实心椭圆	Start	E6h	nn		CRC	End	Start	E6h	nn	信息码	CRC	End
	空心矩形	Start	E7h	nn		CRC	End	Start	E7h	nn	信息码	CRC	End
	实心矩形	Start	E8h	nn		CRC	End	Start	E8h	nn	信息码	CRC	End
	带框矩形	Start	E9h	nn		CRC	End	Start	E9h	nn	信息码	CRC	End
	空心圆角矩形	Start	EAh	nn		CRC	End	Start	EAh	nn	信息码	CRC	End
	实心圆角矩形	Start	EBh	nn		CRC	End	Start	EBh	nn	信息码	CRC	End
	带框圆角矩形	Start	ECh	nn		CRC	End	Start	ECh	nn	信息码	CRC	End
	空心三角形	Start	EDh	nn		CRC	End	Start	EDh	nn	信息码	CRC	End
	实心三角形	Start	EEh	nn		CRC	End	Start	EEh	nn	信息码	CRC	End
	带框三角形	Start	EFh	nn		CRC	End	Start	EFh	nn	信息码	CRC	End
	空心四边形	Start	F0h	nn		CRC	End	Start	F0h	nn	信息码	CRC	End
	实心四边形	Start	F1h	nn		CRC	End	Start	F1h	nn	信息码	CRC	End
	空心五边形	Start	F2h	nn		CRC	End	Start	F2h	nn	信息码	CRC	End
	实心五边形	Start	F3h	nn		CRC	End	Start	F3h	nn	信息码	CRC	End
	圆柱体	Start	F4h	nn		CRC	End	Start	F4h	nn	信息码	CRC	End
	方柱体	Start	F5h	nn		CRC	End	Start	F5h	nn	信息码	CRC	End
表格视窗	Start	F6h	nn		CRC	End	Start	F6h	nn	信息码	CRC	End	
数字键	数字键盘输入	Start	A4h	00		CRC	End	Start	A4h	nn	信息码	CRC	End
	按下数字键后						Start	A4h	nn	ASCII + 信息码	CRC	End	

主功能	细项功能	主控端发送 (TFT 串口屏接收)					主控端接收 (TFT 串口屏发送)						
		起始码 (1Bytes)	指令码 (1Byte)	序号 (1Byte)	指令参数	CRC 码 (2Bytes)	结束码 (4Bytes)	起始码 (1Bytes)	指令码 (1Byte)	序号 (1Byte)	信息码/ 反馈码 (1Bytes)	CRC 码 (2Bytes)	结束码 (4Bytes)
盘		按下 CR 键后					Start	A4h	nn	ASCII + 信息码+ 内容	CRC	End	
	取消 数字键盘	Start	A5h	00		CRC	End	Start	A5h	nn	信息码	CRC	End
全功能 键盘	全键盘 输入	Start	A6h	00		CRC	End	Start	A6h	nn	信息码	CRC	End
		按下 CR 键后					Start	A6h	nn	信息码+ 内容 (ASCII+ GB2312)	CRC	End	
	取消 全键盘	Start	A7h	00		CRC	End	Start	A7h	nn	信息码	CRC	End
摄像头	显示摄像头 画面	Start	A Eh	00		CRC	End	Start	A Eh	nn	信息码	CRC	End
	取消 摄像头画面	Start	A Fh	00		CRC	End	Start	A Fh	nn	信息码	CRC	End
串口屏 侦测	联机检查	Start	B Eh			CRC	End	Start	B Eh	00	5Ah, or 55h	CRC	End
	版本检查	Start	B Fh			CRC	End	Start	B Fh	MCU Code(5) + Module Info. (42)	信息码	CRC	End
设定 时钟	设定时钟	Start	8 Ch		Y, M, D, H, M, S, W (7 Bytes)	CRC	End	Start	8 Ch	00	信息码	CRC	End
	读取时钟	Start	8 Dh			CRC	End	Start	8 Dh	Y, M, D, H, M, S, W (8)	信息码	CRC	End
显示 时钟/ 星期	显示数字 时间、日期	Start	9 2h	nn		CRC	End	Start	9 2h	nn	信息码	CRC	End
	显示模拟 时间、日期	Start	9 3h	nn		CRC	End	Start	9 3h	nn	信息码	CRC	End
	显示星期	Start	9 Dh	nn		CRC	End	Start	9 Dh	nn	信息码	CRC	End
寄存 器控 制	执行 9A 指令	Start	C Ah	Reg		CRC	End	Start	C Ah	nn	信息码	CRC	End
	设定寄存器	Start	C Bh	Reg		CRC	End	Start	C Bh	nn	信息码	CRC	End
	写入数据	Start	C Ch	Data		CRC	End	Start	C Ch	nn	信息码	CRC	End
	读取数据	Start	C Dh	00		CRC	End	Start	C Dh	Data	信息码	CRC	End
	寄存器数据 加 1	Start	C Eh	Reg		CRC	End	Start	C Eh	nn	信息码	CRC	End
	寄存器数据 减 1	Start	C Fh	Reg		CRC	End	Start	C Fh	nn	信息码	CRC	End

### 3.3 RS-232(UART) 通讯协议

主控端的系统或是主板透过 UART 串口传递显示指令给 LT776 TFT 串口屏时，除了 **指令码**、**序号**、**指令参数** 外还要加上 1 个 Byte 的 **起始码**（固定为 0xAA）、2 个 Byte 的 **CRC 码**、4 个 Byte 的 **结束码**（默认为 0xE4、0x1B、0x11、0xEE），用户可自定义编辑结束码，后面结束以默认结束为例，指令信息如下表：

表格 3-3: 串口屏接收的指令信息

起始码	指令码	序号	指令参数	CRC 码	结束码
0xAA (1 Byte)	1 Byte	1 Byte	n Bytes	2 Bytes	0xE4、0x1B、0x11、0xEE (4 Bytes)

CRC 码协议如下：

```
chkSum = Rx_CRC_CCITT(txBuf,txLen);
txBuf[txLen++] = (chkSum>>8)&0xFF;
txBuf[txLen++] = chkSum&0xFF;
```

```
unsigned int Rx_CRC_CCITT(unsigned char *puchMsg, unsigned int usDataLen)
{
    unsigned char i = 0;
    unsigned short wCRcIn = 0x0000;
    unsigned short wCPoly = 0x1021;
    unsigned char wChar = 0;

    while (usDataLen--)
    {
        wChar = *(puchMsg++);
        wCRcIn ^= (wChar << 8);
        for(i = 0; i < 8; i++)
        {
            if (wCRcIn & 0x8000)
                wCRcIn = (wCRcIn << 1) ^ wCPoly;
            else
                wCRcIn = wCRcIn << 1;
        }
    }
    return (wCRcIn);
}
```

串口屏在收到主控端的系统或是主板指令后会通常会响应 10 个 Byte 信息，包括 **起始码**、**指令码**、**序号**、**信息码**、**CRC 码**、**结束码**，第一个 Byte 是**起始码**，然后是传回所收到的指令，第三个是**序号**，第四个传回串口屏执行结果的**信息码**，第五、六个是 **CRC 码**，最后是 4 个 Bytes 的 **结束码**：

表格 3-4: 串口屏返馈的信息

起始码	指令码	序号	信息码	CRC 码	结束码
-----	-----	----	-----	-------	-----

<p>0xAA (1 Byte)</p>	<p>1 Byte</p>	<p>一般指令 (1 Byte)</p> <p>8Dh 指令 (8 Bytes)</p> <p>BFh 指令 (47 Bytes)</p>	<p>1Byte</p> <p>0x00: 执行完该指令 0x01: 串口指令参数错误 0x02: 不存在该指令 0x03: 指令 Flash 配置溢出 0x04: CRC 码校正错误 0x05: Flash 数据异常</p> <p>BEh 指令: 0x5A: Ready 0x55: Not Ready</p> <p>94h 触控滑条指令: 代表进度条指标的百分比位置</p> <p>A0h, 9Ch/9Bh 控件指令: 0x31: 按下控件 0x30: 放开控件</p>	<p>2 Bytes</p>	<p>0xE4、0x1B、 0x11、0xEE (4 Bytes)</p>
--------------------------	---------------	---	---	----------------	---

在串口屏反馈的信息结构中，序号在某些指令也代表不同的意思，如控件滑动的 9Ch 指令其序号代表页号、9Bh 指令其序号代表图标 ID 号、设置亮度 BAh 指令其序号代表背光亮度、Wav 播放 B8h 指令其序号代表 WAV 编号、读取时钟 8Dh 指令其序号有 8 个 Bytes 代表时钟信息、版本检查 BFh 指令其序号有 47 个 Bytes 代表串口屏信息。

### 3.4 主控端发送指令的范例

以下是主控端的 MCU 透过 UART 串口传递显示指令给 LT776 TFT 串口屏的程序范例, 本程序以传送显示第一张图片 (80h、00h) 为范例, 程序内会自动加入 0xAA 起始码、2 个 Byte 的 CRC 码及 4 个 Byte 的结束码:

```
int main (void)
{
    char c[] = "80 00";          //发送 80h 指令, 和 00 序号
    Send(c);

    while(1);
}

unsigned short Rx_CRC_CCITT(unsigned char *puchMsg, unsigned int usDataLen) // 生成 CRC
{
    unsigned char i = 0;
    unsigned short wCRCin = 0x0000;
    unsigned short wCPoly = 0x1021;
    unsigned char wChar = 0;

    while (usDataLen--)
    {
        wChar = *(puchMsg++);
        wCRCin ^= (wChar << 8);
        for(i = 0; i < 8; i++)
        {
            if (wCRCin & 0x8000)
                wCRCin = (wCRCin << 1) ^ wCPoly;
            else
                wCRCin = wCRCin << 1;
        }
    }
    return (wCRCin);
}

void Send(char *c)              //发送指令函数
{
    unsigned char Sendbuff[100]={0};
    unsigned short Send_CRC = 0;
    unsigned char C_flag = 0;    //判断是否在" "中

    int i = 0, j = 0;

    if(((c[0]>=0x30 && c[0]<=0x39) || (c[0]>=0x41 && c[0]<=0x5A)) || ((c[1]>=0x30 &&
c[1]<=0x39) || (c[1]>=0x41 && c[1]<=0x5A))) //只有第一个和第二个字符是有效的才是指令
    {
        while(c[i] != '\0')
        {
            if(c[i] != ' ')          //排除空格, 非空格可以进入
            {
                if(c[i] == '"')      //当" "中的字符 以 ASCII 输出, 不需要输出" "号,
            }
        }
    }
}
```

```

{
    C_flag++;
    i++;
}
if(C_flag == 1)
{
    if(c[i] != '"')
    {
        Sendbuff[j] = c[i];          //ASCII 直接输出
        i++;
        j++;
    }

}
else if(C_flag == 2)                //第二次遇到"
{
    C_flag = 0;
    i++;
}
if(C_flag == 0)
{
    if(c[i] == '/') break;

    if(c[i]>=0x30 && c[i]<=0x39) //0~9
    {
        Sendbuff[j] = ((c[i] - 0x30) << 4);
        i++;

        if(c[i]>=0x30 && c[i]<=0x39)
        {
            Sendbuff[j] += (c[i] - 0x30);
            i++;
            j++;
        }
        else if(c[i]>=0x41 && c[i]<=0x5A)
        {
            Sendbuff[j] += (c[i] - 0x37);
            i++;
            j++;
        }
    }

}

else if(c[i]>=0x41 && c[i]<=0x5A) //A~Z
{
    Sendbuff[j] = ((c[i] - 0x37) << 4);
    i++;

    if(c[i]>=0x30 && c[i]<=0x39)
    {
        Sendbuff[j] += (c[i] - 0x30);
        i++;
        j++;
    }
    else if(c[i]>=0x41 && c[i]<=0x5A)
    {
        Sendbuff[j] += (c[i] - 0x37);
        i++;
    }
}

```

```

        j++;
    }

}

}
}
else i++;
}
Sendbuff[j] = '\0';
// printf("%s\r\n",Sendbuff);

/*****CRC 和针头针尾*****/

Send_CRC = Rx_CRC_CCITT(Sendbuff,j);

Sendbuff[j] = Send_CRC>>8 & 0xff;
Sendbuff[j+1] = Send_CRC & 0xff;

for(i = 0;i<j+2;i++)
    Sendbuff[j+2-i]=Sendbuff[j+1-i];           //挪位

Sendbuff[0] = 0xAA;
Sendbuff[j+3] = 0xE4;
Sendbuff[j+4] = 0x1B;
Sendbuff[j+5] = 0x11;
Sendbuff[j+6] = 0xEE;

j+=7;

for(i = 0; i < j; i++)
{
    USART_DATA(USART0) = (uint8_t) Sendbuff[i];
    while(USART_FLAG_GET(USART0, USART_FLAG_TBE)==0){}; //循环发送,直到发送完毕
}
}
}
}

```

## 4. 图文 UI 编辑编辑器 (UI\_Editor.exe)

### 4.1 UI\_Editor 界面的简介

UI\_Editor.exe 是 乐升半导体 提供的一款以串口屏为对象的 **图文 UI 编译器**。它的功能是根据客户的需求, 将串口屏要用到的图片、文字、配置数据等信息打包生成 BIN 档。客户可以使用 UI\_Editor 简单、快捷的制作 UI 界面, 之后将生成的 BIN 文档烧录到 SPI Flash 中。

**提示:** UI\_Editor 是在 Microsoft .NET Framework 4.6.2 的环境中编写出来的, 所以电脑系统必须安装 Microsoft .NET Framework 4.6.2 才能正常使用。

UI\_Editor 的界面由各种按钮和屏幕框组成如下图所示:



图 4-1: UI\_Editor 主画面

所有的 UI 设计都在屏幕框内完成，用户根据需求选用不同的功能实现设计。其中各种功能键的详细功能如下：

1.  在 UI\_Editor 上以图片形式显示文字
2.  添加图片按钮
3.  添加控件按钮
4.  添加 GIF 图按钮
5.  添加数字时钟按钮
6.  添加星期按钮
7.  添加数字按钮（图片）
8.  添加真彩图片按钮（图片）
9.  添加文字按钮（字库）
10.  添加 ASCII 按钮
11.  添加二维码按钮
12.  添加表格
13.  添加画圆环（任意角度）
14.  添加进度条按钮
15.  添加触控滑动条按钮
16.  分别是画矩形、画圆角矩形、画圆、画椭圆
17.  分别是画线、画三角形、画四边形、画五角形、画圆柱体、画长方体
18.  添加指针按钮
19.  添加虚拟按键按钮
20.  分别是撤回操作和恢复操作按钮(当无任何操作时初始画面是 )

与 UI\_Editor 工具同级的有几个文件夹，它们的作用如下图所示。

- ★ ColorNumber 文件夹用来存放真彩数字图片文件
- ★ FONT 文件夹用来存放需要使用的字库
- ★ PICFILE 文件夹可用来先存放需要使用到的图片文件
- ★ PROJECT 文件夹备份着每次 Save 和 Build 的工程文件
- ★ SOURCE 文件夹用来存放音频 BIN 文件

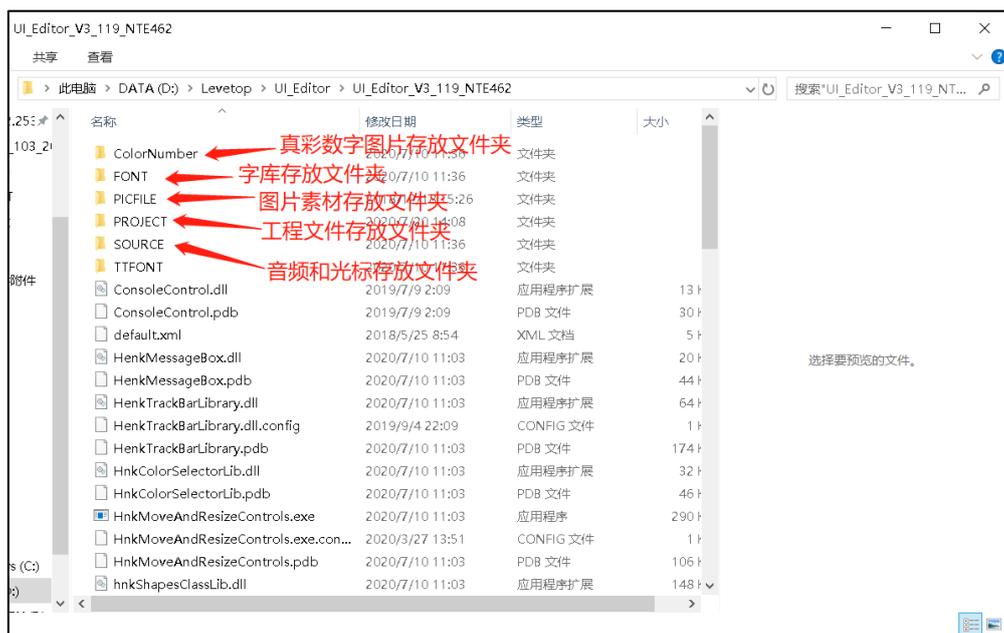


图 4-2: UI\_Editor 工具同级文件目录

PROJECT 文件夹下级的工程文件里有几个文件夹，它们的作用如下图所示。

- ★ BINFILE 文件夹存放着编译好的 BIN 文件，需要烧录的 UserInfo 和 UartTFT\_Flash 就存放在此处。
- ★ COMMANDFILE 文件夹存放着工程储存文件
- ★ PICFILE 文件夹存放着编译后的图片文件
- ★ SRCPIC 文件夹存放着编译前的图片



图 4-3: PROJECT 文件夹下级的工程文件目录

在菜单按钮里，有 New Project、load、save 三个按钮。分别用来创建新工程、装载工程文件、保存当前工程。按 save 按钮会把工程以 mainControlFiles.xml 文件保存在 PROJECT 下级中以时间命名的 COMMANDFILE 文件夹里。使用 Load 功能在 PROJECT 下级找到对应时间的文件夹里 COMMANDFILE 文件夹的 mainControlFiles.xml 文件，就可以重新加载工程。



图 4-4: UI\_Editor 重装载工程文件

4.1.1 使用 UI\_Editor 的设计流程

下图为用图文 UI 编译器 (UI\_Editor.exe) 开发的详细流程图, 用户也可以至 [乐升半导体](#) 网站下载 LT776 的 UI\_Editor 范例 (LT776\_UI\_Editor\_Demo\_800x480.rar) 来操作, 将更快速的了解开发模式。同时建议用户先依据所需功能及 TFT 屏幕大小准备好素材, 因为这些显示图片、动画文件、文字库、声音文件等是存放在 SPI Flash 内, 资料量都不小, 而 SPI Flash 的烧录所需时间较长, 因此尽量避免开发中反复对 SPI Flash 进行 UartTFT\_Flash.bin 档的烧写, 以免延误开发效率。[乐升半导体](#) 的 LT776 TFT 串口屏开发演示板套件包括了 SPI Flash 烧录器, 而烧录器的烧录软件、UI\_Editor 图文 UI 编译器及 UI\_Emulator 串口屏仿真器软件都可以自 [乐升半导体](#) 官网的“串口屏上位机软件”下载专区下载。

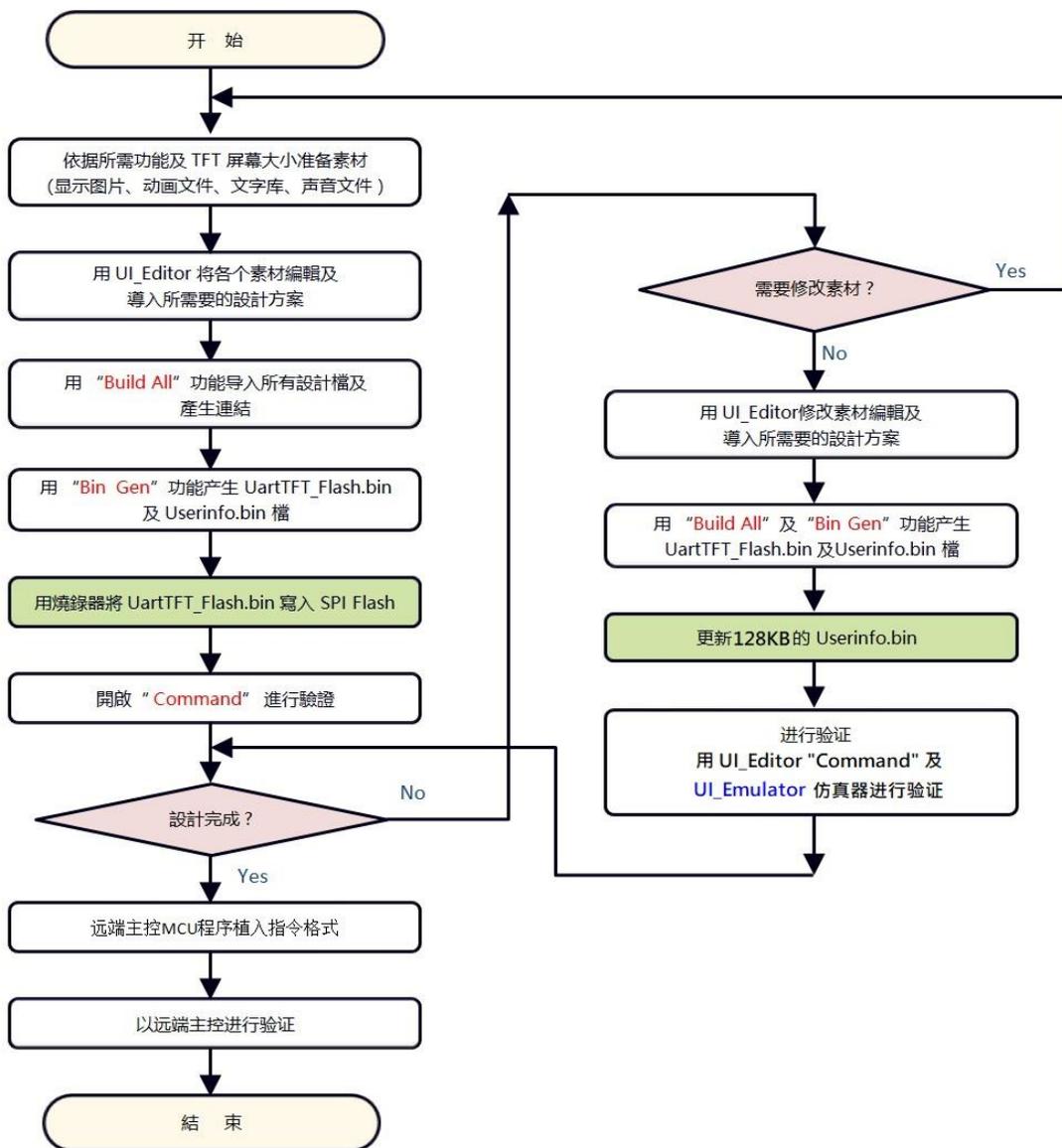


图 4-5: 使用 UI\_Editor 的设计流程

在使用 UI\_Editor 工具之前用户需依据产品的需求进行显示规划，这些规划与设计与 UI\_Editor 无关，使用者可以用绘图软件去设计图片或是动画 (Gif 檔)，顯示的流程可以用 Word、PowerPoint 或是自己熟悉的工具去纪录，规划及准备的方向大致如下所示：

1. 确认显示屏的分辨率，例如 480\*272 或是 800\*480 等等。
2. 以 1:1 方式设计产品中会需要显示的底图，例如显示屏 480\*272，那么底图就以 480\*272 分辨率来设计。虽然 UI\_Editor 可以改变导入图片的大小，但是不同的算法会导致图片的局部失真，因此强烈建议 UI 工程师以 1:1 方式设计图片。
3. 设计产品中需要显示的各个大小的图片、或是动画 (Gif 檔)。
4. 设计产品中需要显示的图片或是动画的显示坐标。
5. 设计显示的流程，例如主控发不同指令后显示屏会更动哪那些画面。这些显示流程跟用户的产品有绝对的关联，必须先思考清楚，而不是在使用 UI\_Editor 时一边思考一边导入。
6. 确认是否使用触控屏功能，如果有就要设计触控控件的图片及显示坐标位置。
7. 如果需要音乐或是中文字库，也要准备音源及字库檔利用工具转成 Bin 檔。
8. 在检视整个显示流程无误及准备好所以素材图片之后，就可以开启 UI\_Editor 进行显示图片及流程的导入。

以上流程的目的是强调用户在操作 UI\_Editor 工具之前的准备工作，这样可以大大的缩短实际设计时间，及减少非常多的编译次数，提升导入串口屏显示的开发效率。

### 4.1.2 使用 UI\_Editor 的注意事项

在使用 UI\_Editor 的制作项目时，建议使用者对不同的项目使用不同的 UI\_Editor 目录，可以通过复制 UI\_Editor 的文件夹即可。UI\_Editor 文件夹可重命名为项目名称，便于区分不同项目保存或编译后产生的工程，如下图：



图 4-6：不同的工程建立不同的项目目录

在制作工程时，可通过点击工具下方“SAVE”保存当前制作的工程，或通过编译“Build All”来保存，如下图：



图 4-7：点击“SAVE”保存当前制作的工程



图 4-8：点击编译“Build All”保存当前制作的工程

保存后会自动生成一个以时间命名的工程文件夹，该工程文件夹保持 UI\_Editor 路径下的 PROJECT 文件夹中，用户可以自行选择删除不需要的工程文件夹，如下图。若只需保存最新的工程，可在调试完成后删除旧的工程文件夹，只保留最新的即可。

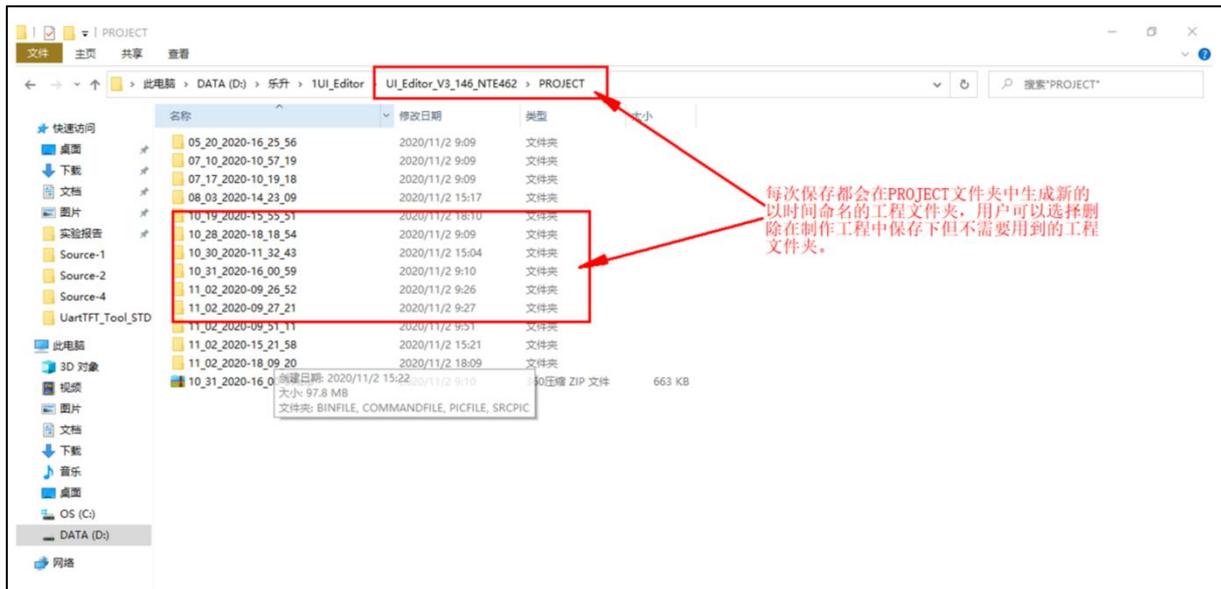


图 4-9：可以自行选择删除不需要的工程文件夹

注意 1: UI\_Editor 文件夹名称要根据原来自动产生的为主, 不可更改, 若是客户已经改了, 会 Load project 失败, 此时可将文件夹名称再改为原来的年月日时分秒名称即可; 若是忘了名称, 例如: 可打开 Projct-->COMMANDFILE-->mainControlFiles.xml, 找到 <Text>PROJECT\08\_26\_2020-09\_30\_40\.... 则 “08\_26\_2020-09\_30\_40” 即为原本的 Project 名称。

注意 2: 若是客户需要以特定文件夹名称保存 Proejct, 建议客户可用自订文件夹名称内再放入 Proejct, 再压缩备份。

注意 3: 用户在使用 UI\_Editor 制作工程时, 添加的图片素材不允许存在相同命名, 即使两个不同路径的素材, 也不能以相同名称命名, 如下图:

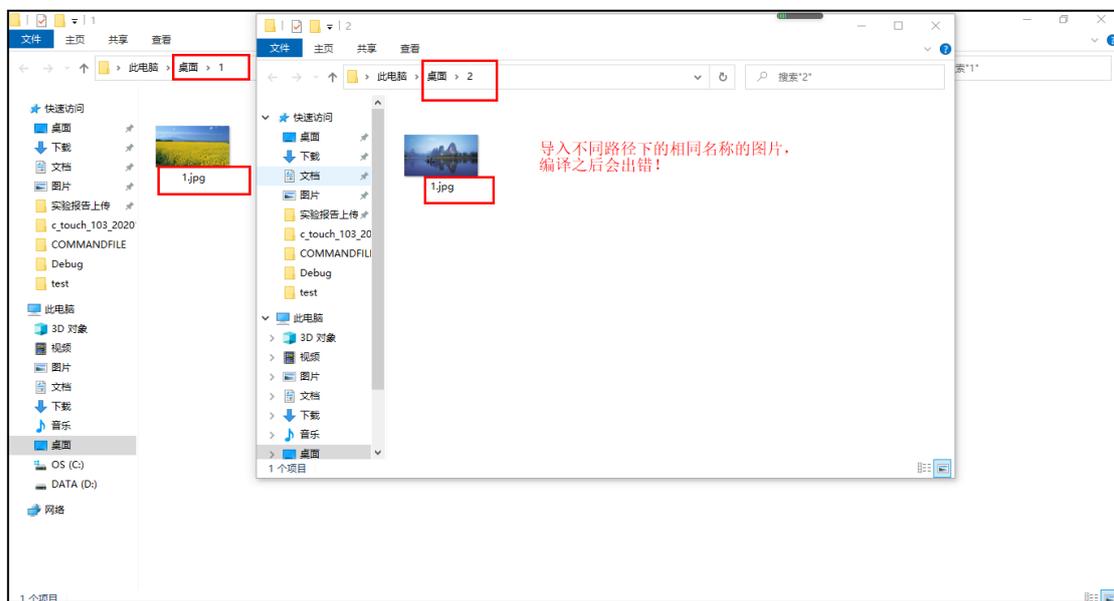


图 4-10: 图片素材不允许存在相同命名

### 4.1.3 屏幕初始化和背景图的设定

先打开 UI\_Editor。

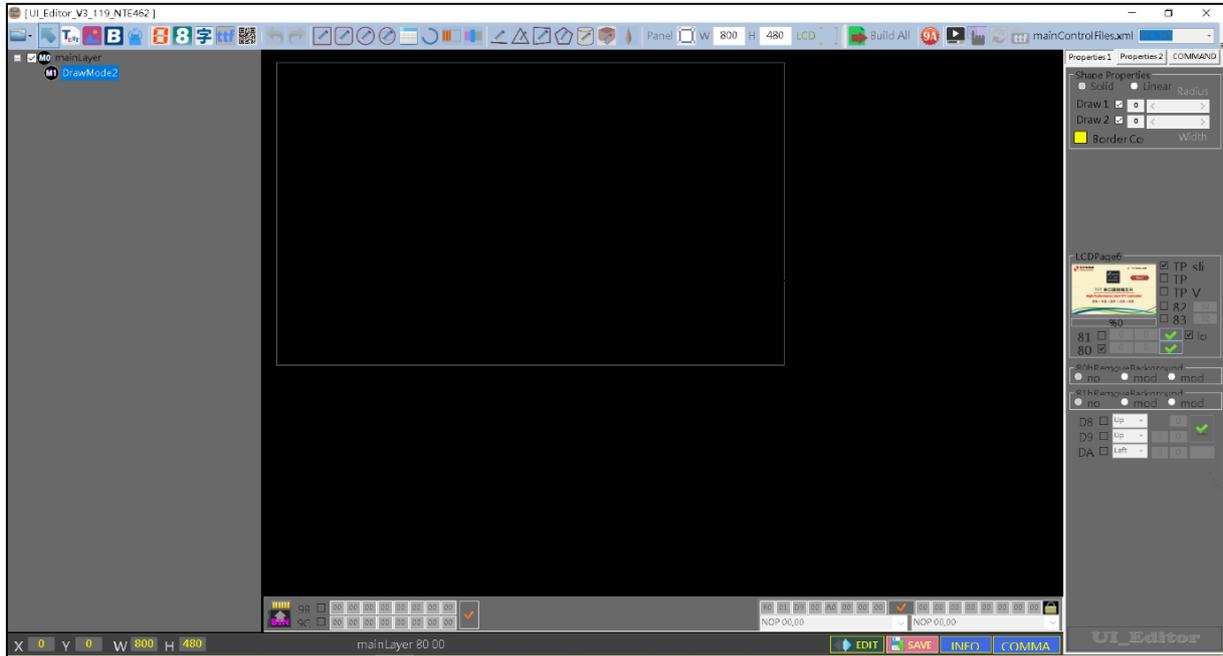


图 4-11: 打开的 UI\_Editor

先在菜单中选择“New Project”，再点击“屏幕框选择按钮”，最后再双击框内任意位置选择打开要在该图层的图片，如下图所示：



图 4-12: 新建工程

通过修改右边的屏幕大小内的数字可以改变屏幕的长、宽。要注意的是每个框中的数字修改了之后要按回车来确认，否则无法修改。例如要设定 5 寸 800X480 分辨率的屏幕。则将 W 修改为 800 按下回车键确认，再将 H 修改为 480 按下回车键确认。屏幕大小的设定可以在任何时候进行。在进行 UI 设计时一定要确认设定的屏幕大小没有错误。



图 4-13：设置图层分辨率

点击“锁定/移动屏幕框”按钮后，整个屏幕框可以被拖动到任意位置，可放在视野适合的位置。再次点击该按钮后，即取消了屏幕框选择功能，此时不能修改背景图和拖动屏幕框。



图 4-14：拖动屏幕框到任意位置



图 4-15: 取消了屏幕框选择功能

#### 4.1.4 显示图的设定

初始化了屏幕框之后，默认是屏幕框显示的是一个主图层。我们可以通过在项目栏中主图层选项中右键，选择 Add New Page 来增加图层。



图 4-16: 增加图层

如果所建图层只需与主图层共享一张图片，这时在此界面点击“取消”即可。

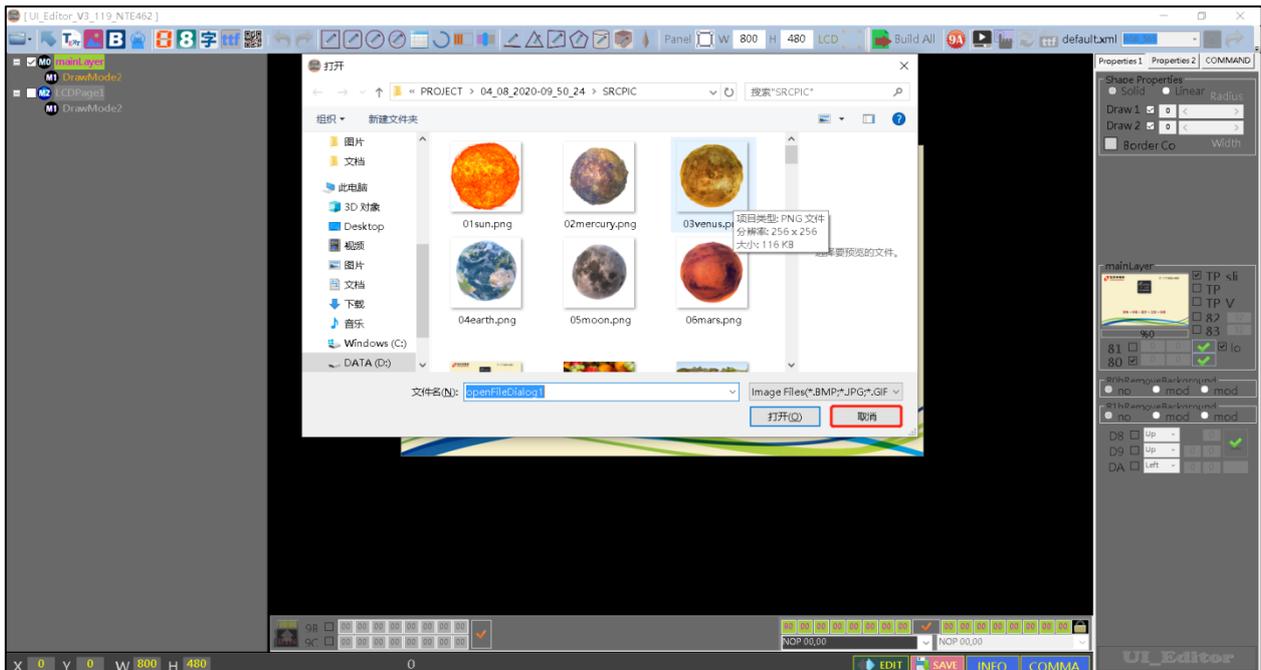


图 4-17: 新建图层与主图层

如果所建图层需要新建一张独立的图片，点然后击“打开”。

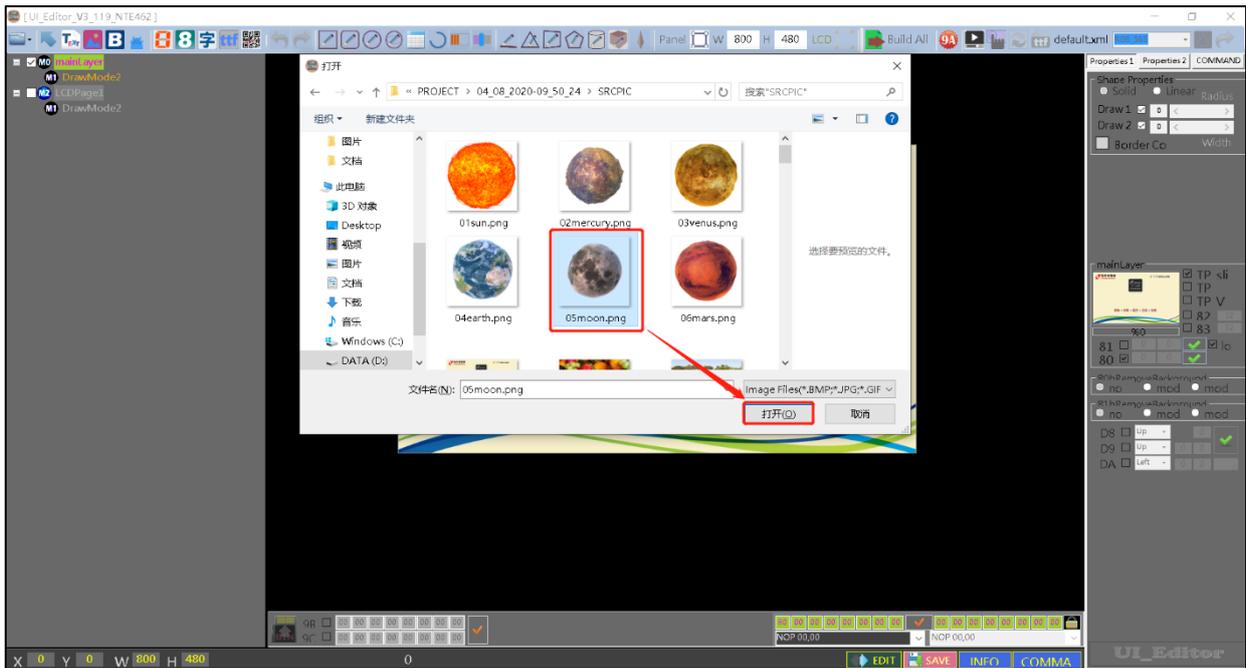


图 4-18: 新建独立的图层

如果需要变更图层的背景图，先勾选需要变更的图层，点击该图层的“DrawMode”，然后点击“LCDPage”，接着双击缩略的显示框，即可根据需求来选择图层的背景。

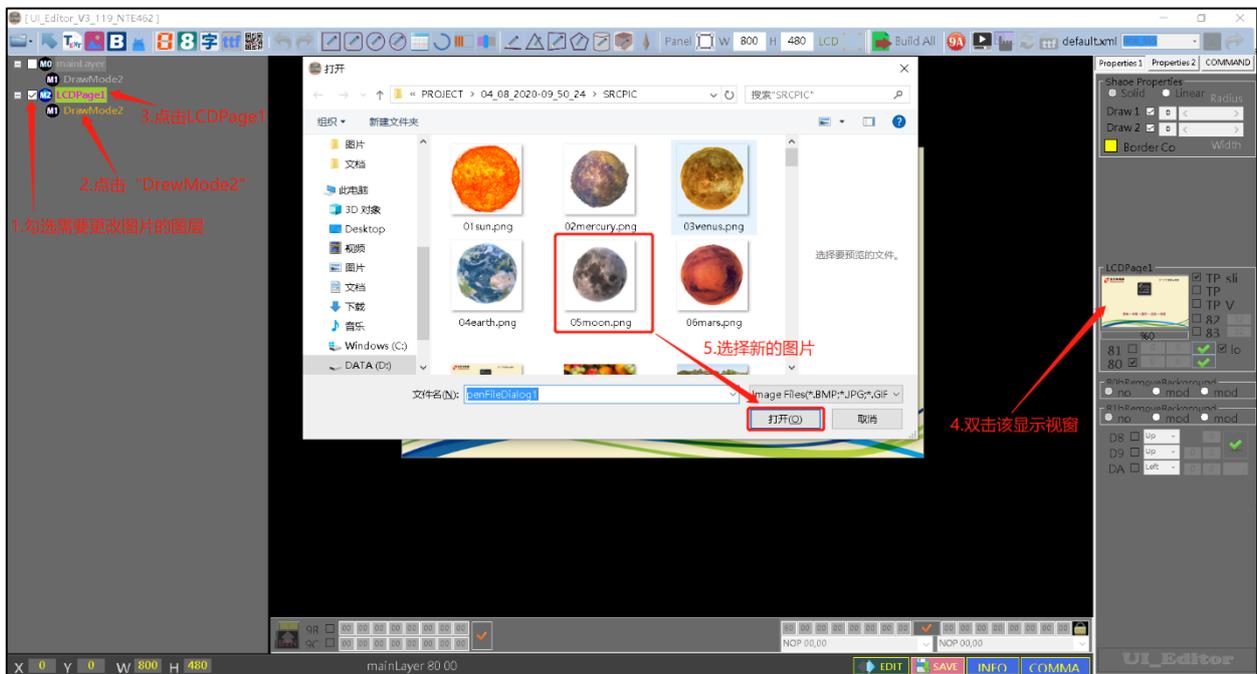


图 4-19: 变更图层背景

当存在多个图层的时候，可以通过勾选图层前面的小方格来显示和隐藏当前图层。在当前的图层，右击该图层可对当图层进行删除或者移动。



图 4-20：删除或移动图层

### 4.1.5 界面的编辑与调试

删除操作有两种以下方式

方式一：先选中要删除的目标，点击右键后选择“Remove”删除该控件，如下图所示：

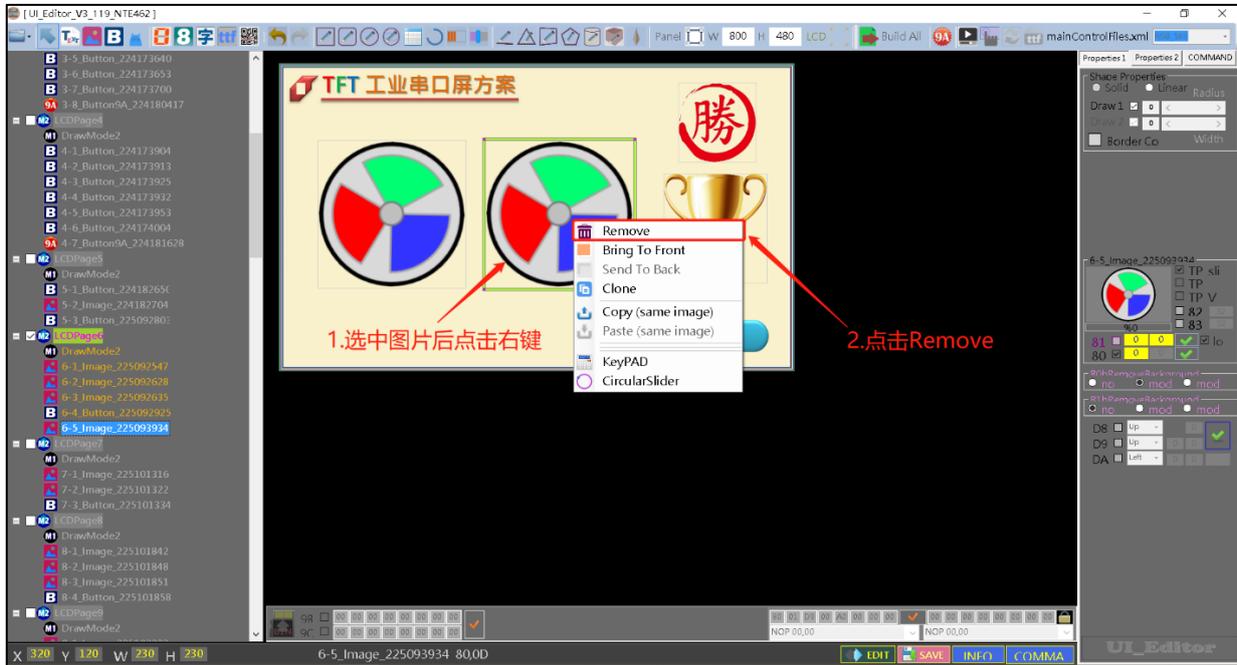


图 4-21：删除操作方式一

方式二：先在项目栏选中要删除的目标后点击右键，然后点击“Remove”即可删除该目标，如下图所示：



图 4-22：删除操作方式二

如下图所示，默认的状态下克隆的功能是异位置克隆，先选中要克隆的控件，点击右键后选择“Clone”就会产生一个相同但异位置的控件。



图 4-23: 异位置克隆操作

如下图所示，克隆前先双击右下方的灰色框，此时状态会更改为 false，选中要克隆的控件，点击右键后选择“Clone”就会产生一个相同且同位置的控件。



图 4-24: 同位置克隆操作

如下图所示，选中的控件右键后可出现控件的菜单，选择 copy 复制该控件。



图 4-25：复制功能一

如下图所示，选择 LCDpage 后，右键点击 paste 生成一个与复制来源相同的控件，可选择同位置复制或异位置复制。

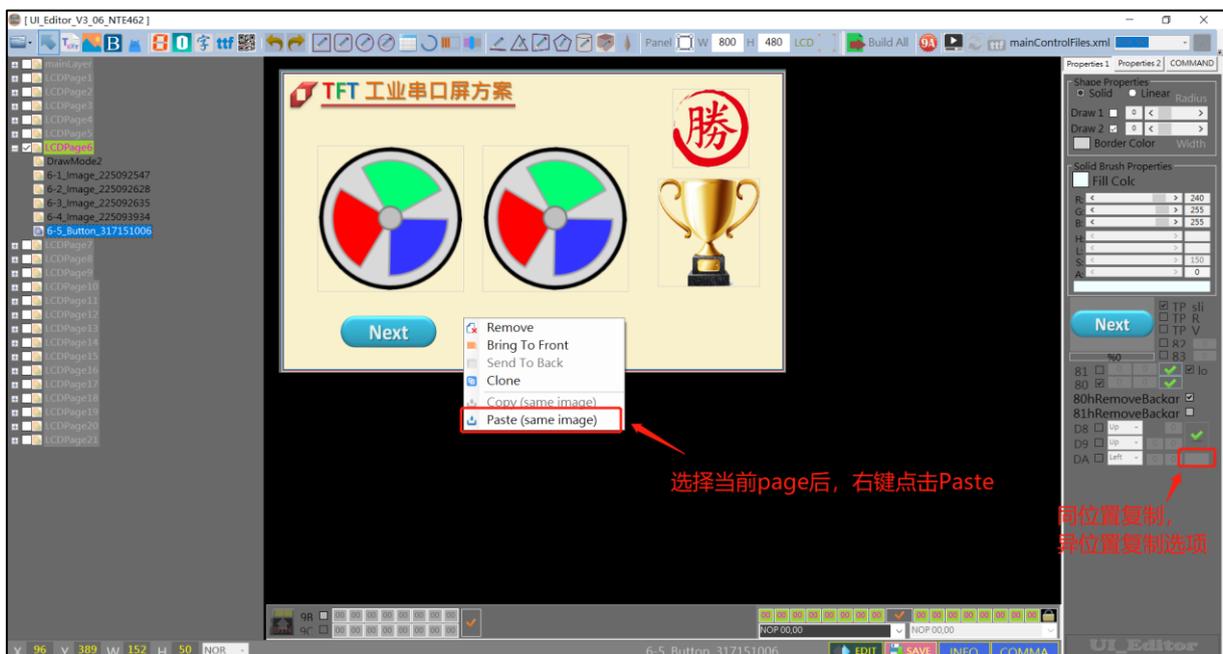


图 4-26：复制功能二

如下图所示，复制后会生成一个相同的控件，该控件与来源控件相同。



图 4-27: 复制功能三

如下图所示，表示当前处在正常编辑的状态下，所有的添加、删除和编辑功能都能正常执行。



图 4-28: 编辑状态

如下图所示,表示当前处在调试的状态下,此时不能对界面进行编辑,选中对应的控件点击右键,此时会弹出相关的指令,点击相应的指令即为发送该串口指令,要想重新对界面进行编辑,再次点击下方红色框的这个图标即可。



图 4-29: 调试状态

## 4.2 显示图片的设置

### 4.2.1 设定显示多张图片

先点击图片添加按钮，在屏幕框中点击拖动，出现一个四角带点的白线框。这个白线框就是图片的大小。松开手后，在弹出的资源管理器中选中一张图片。图片的大小可以通过拉动白线框的大小来改变。如果想更换图片可以在图片被选中的前提下，双击图片来更改。如果图片不能被选中（没有出现四角白线框），可以单击项目栏中的项目来选中对应的图片对象。成功添加图片之后，可以看到右边操作栏中有 80h 指令的功能框。

**显示单张图片：** 如果想在发送指令后单独显示一张图片，则可以不用设置组号。

**显示多张图片：** 要是想要用一个指令显示多张图片，就设定组号，相同组号的图片会一起显示。在设定了组号之后，务必按确认按键完成设定。

其中如果图片是 PNG 的话可以勾选去掉 PNG 背景框 (80h RemoveBackground) 来去掉 PNG 图片的背景。



图 4-30：设定显示多张图片

### 4.2.2 设定循环显示重叠图片

在添加了图片之后，可以设定循环显示图片的组号和显示之间的时间间隔。其中时间间隔是以 10ms 为单位。例如图片 A 和图片 B 要以 200ms 时间间隔循环显示。则应该在图片 A 的 81h 指令操作框处设定 1 和 20，按下确定键。在图片 B 的 81h 指令操作框处设定 1 和 0，按下确定键。图片 A 和图片 B 就会形成一个组合 1，在接收到对应的指令后，循环显示。其中如果图片是 PNG 的话可以勾选去掉 PNG 背景框 81h RemoveBackground (mode1:软件 PNG, mode2 硬件 PNG) 来去掉 PNG 图片的背景。其中每生成一个 81h 指令 (循环显示重叠图片)，同时也会生成一个 84h 指令 (取消循环显示重叠图片)。



图 4-31：设定循环显示重叠图片

### 4.2.3 设定卷动出现图片

卷动图片效果是从某一个地方按时间间隔逐渐显示一张图片。在添加图片之后，勾选 D8h 操作栏中的小方格，对图片开启卷动功能。之后可以选择图片卷动的方向。有四个选项，分别是：向上卷动、向下卷动、向左卷动和向右卷动。选择一个自己需要的方向后，填写卷动的时间间隔，最后按确认按钮。



图 4-32：设定卷动出现图片

### 4.2.4 设定循环卷动图片

循环卷动图片是在卷动图片的基础上，可以实现一张图片循环卷动出现，或者多张图片持续的顺序卷动出现。具体的操作方法和上面的设定卷动出现图片的基本类似。其中假如要实现一张图片循环卷动，则只在那张图片的 D9h 指令操作栏中设定就可以了。假如要实现多张图片的循环卷动，则在对应图片的 D9h 指令操作栏中同时设定，时间间隔可以不一样。其中每生成一个 D9h 指令（显示循环卷动图片），同时也会生成一个 DBh 指令（取消显示循环卷动图片）。



图 4-33：设定循环卷动图片

### 4.2.5 设定显示 GIF 动画图片

按 GIF 添加按钮,从资源管理器中选一张 GIF 图片。在右边的操作栏中可以找到 GIF 的操作栏。在小演示窗口的右下角的数据框是 GIF 图播放间隔的设定框 (以 100ms 为单位), LOOP 为是否循环播放 GIF 图。此时 81h 指令的确认设定按钮变为了 GIF 的确认设定按钮。先设定了播放间隔和是否循环播放 GIF 图后再按确认按钮,即可完成设定。其中每生成一个 88h 指令 (显示 GIF 动画图片), 同时也会生成一个 89h 指令 (取消显示 GIF 动画图片)。



图 4-34: 设定显示 GIF 动画图片

### 4.2.6 设定显示二维码图片

点击二维码添加按钮，在屏幕框中点击拖动，形成白线框。松开，就会弹出二维码设定窗口。在二维码的信息栏中输入要显示的信息。二维码支持显示图标功能。点击了选用图标按钮后，点击图标选择按钮。在弹出的资源管理其中选一个图标。通过图标大小调节条来改变图标的大小。可以点击 Preview 按钮预览二维码显示效果。(注：98h 指令需要的内存为 27k，默认为关闭 98h 指令，若要开启 98h 指令，则必须关闭其它指令使空闲内存达到 27k 以上。)



图 4-35：设定显示二维码

### 4.2.7 设定显示控件功能的图片

可以通过添加按钮控件来实现按键功能，即在屏幕上显示一个按钮图片，触摸到这个图片的时候，执行一系列动作。如显示图片、显示 GIF、播放音频等。

先点击按键添加按钮，在屏幕框中点击、拖动生成白线框。在资源管理器中选一张图片作为按钮图片。之后可以在屏幕框的下边找到按钮的操作框。只有选中了按钮图片，它们才会可用。点击第一组指令输入处（两格组成一个指令，最多可执行 8 个指令，若需要执行 8 个以上的指令可使用 9A 进行嵌套），在下面的指令可选列表中选择要执行的指令。设置好后点击确认设置按钮即可。指令可选列表是由当前工程中其他已生成动作指令组成的。例如在设定按钮功能前，已经设定了显示一张图片和一张 GIF 图的指令。则指令列表中会有这两个指令。特别说明，两个按钮操作框使用方法一样。如果控件图片是 32 位 PNG 格式，希望去掉背景，则可以勾选 80h RemoveBackground 的去背景框 (mode1:软件 PNG, mode2 硬件 PNG)。其中每生成一个 A0h 指令（显示控件），同时也会生成一个 A1h 指令（取消控件）。



图 4-36：设定显示控件功能的图片

#### 4.2.8 设定显示虚拟按键

可以通过添加虚拟按钮控件来实现按键功能，即在屏幕上设定一个区域，但是 user 看不到这个区域的变化（即虚拟按钮为完全透明），当触摸到这个区域的时候，执行一系列动作。如显示图片、显示 GIF、播放音频等。

先点击虚拟按钮添加按钮，在屏幕框中点击、拖动生成白线框，此框即为虚拟按钮的有效范围。选中虚拟按钮之后可以在屏幕框的下边找到该虚拟按钮的操作框。点击第一组指令输入处（两格组成一个指令，最多可执行 8 个指令），在下面的指令可选列表中选择要执行的指令。设置好后点击确认设置按钮即可。指令可选列表是由当前工程中其他已生成动作指令组成的。例如在设定虚拟按钮功能前，已经设定了显示一张图片和一张 GIF 图的指令。则指令列表中会有这两个指令。特别说明，两个按钮操作框使用方法一样。其中每生成一个 A2h 指令（显示虚拟控件），同时也会生成一个 A3h 指令（取消虚拟控件）。

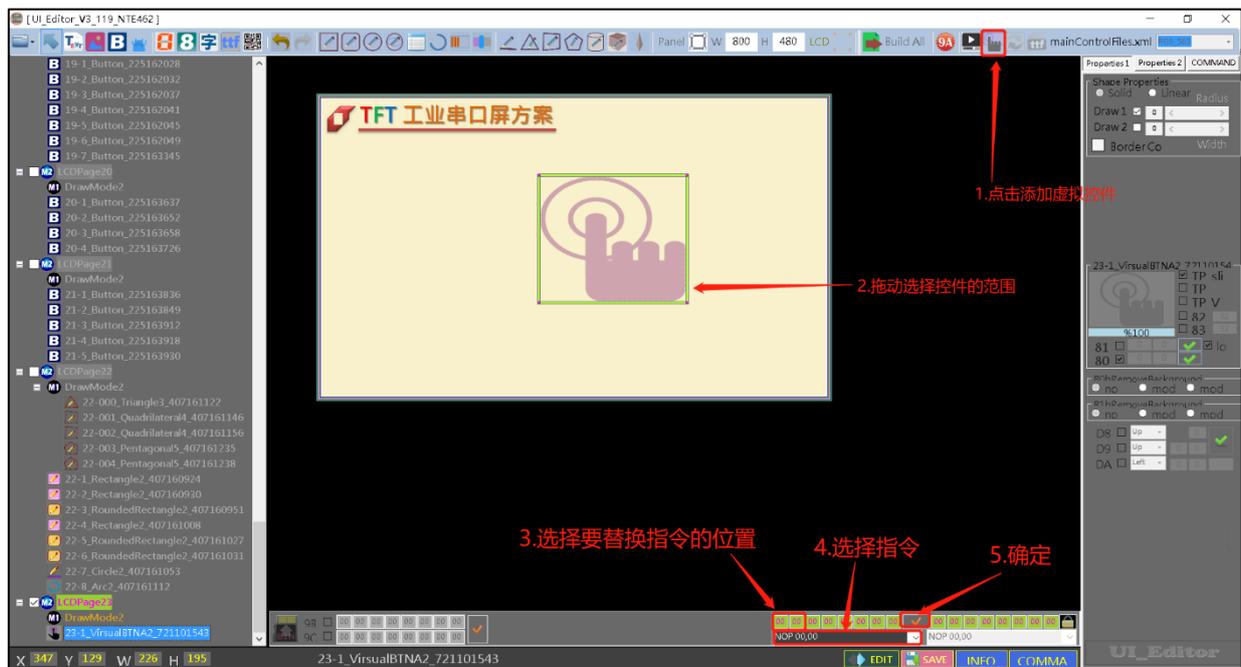


图 4-37：设定显示虚拟控件

### 4.2.9 设定开机画面

开机画面设定是用来选择那些控件在开机时被预先执行，而不需要主控端先发送指令，以下示例假设构建的控件已经都存在：

1. 先点选左上角 “mainLayer”，此时会出现 8 组在开机时可预先选择要发送的指令。



图 4-38：点选左上角 “mainLay”

只要点选任一组数字，即会显示黄色框背景与红色字体，再根据如下图所示位置点选并自动上拉展开 Command 群组，此时即可选择每一组 Command 的指令组合。



图 4-39：选择每一组 Command 的指令组合

2. 开机画面设定如下如所选的指令 (User Command)



图 4-40：开机画面所选的指令

9A 指令中除了 9A, 00 为开机指令外, 也可以用来设定执行多组的指令, 例如新增加 9A 01、9A 02 等等之后的使用者指令, 这些指令码主要是让使用者可以直接从串口输出 9A 01、9A 02 等等指令, 控制 UI\_Editor 编辑时预先设定要呼叫的其他指令, 范例如下图所示 (实际于 LCD 并不会显示 9A 图标)。



图 4-41: 以 9A 指令来设定执行多组指令

49	9A 01	6-7_Button9A_827101400	SEND	AA	2019/8/27 17:14:06
50	80 12	LCDPage7	SEND		
51	B4 07	PageSlide7	SEND		
80001B98E900AA2AEC0055DFE30045E1E600BA1480001B98F600B967A0001D7E9A00F720					

图 4-42: 以 9A、01 指令来执行多组指令

9A 指令可以用来合并多个指令, 非常好用, 建议使用者多加利用, 除了简化许多图片显示的设计外, 也可以简化主控端的设计。

### 4.2.10 设定触摸进度条图

此功能是结合触控屏的滑动，来改变 TFT 屏上所显示的滑动条长短，同时反馈信息给主控端。

1. 请点 [94h] 滑动条图示，并可开始在屏幕范围内做拖拉移动的动作，此时即可产生滑动条。

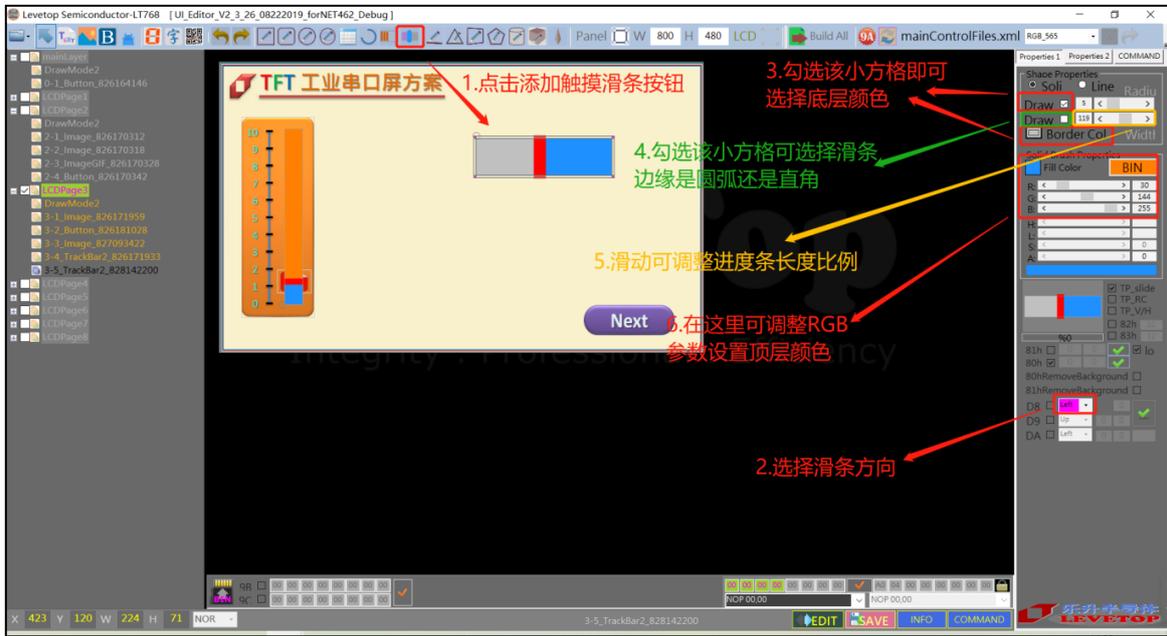


图 4-43：设定滑动条及前景颜色与背景颜色

2. 设定滑动条移动时前景颜色与背景颜色
3. 务必要设定滑动条背景图片如下图所示，请拖拉移动一张图片当作滑动条背景，并且要将图片设定为去除边框透明功能。

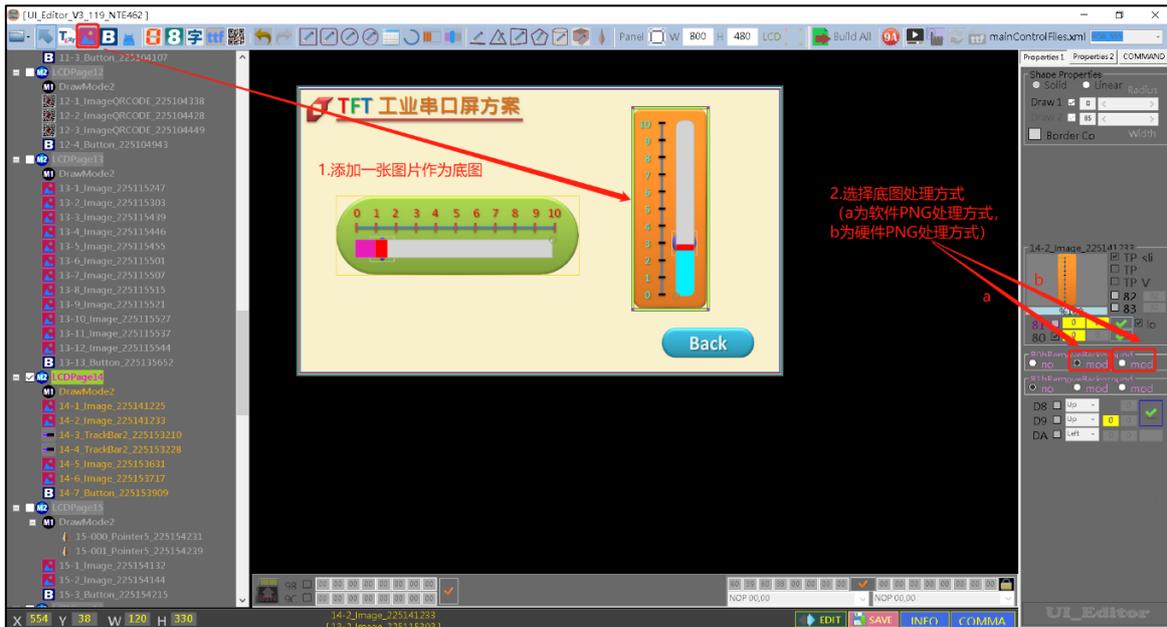


图 4-44：设定滑动条背景图片

- 再新增一张图片当作滑动条的移动图标头，如下图所示，并且也要将图片设定为去除边框透明功能。

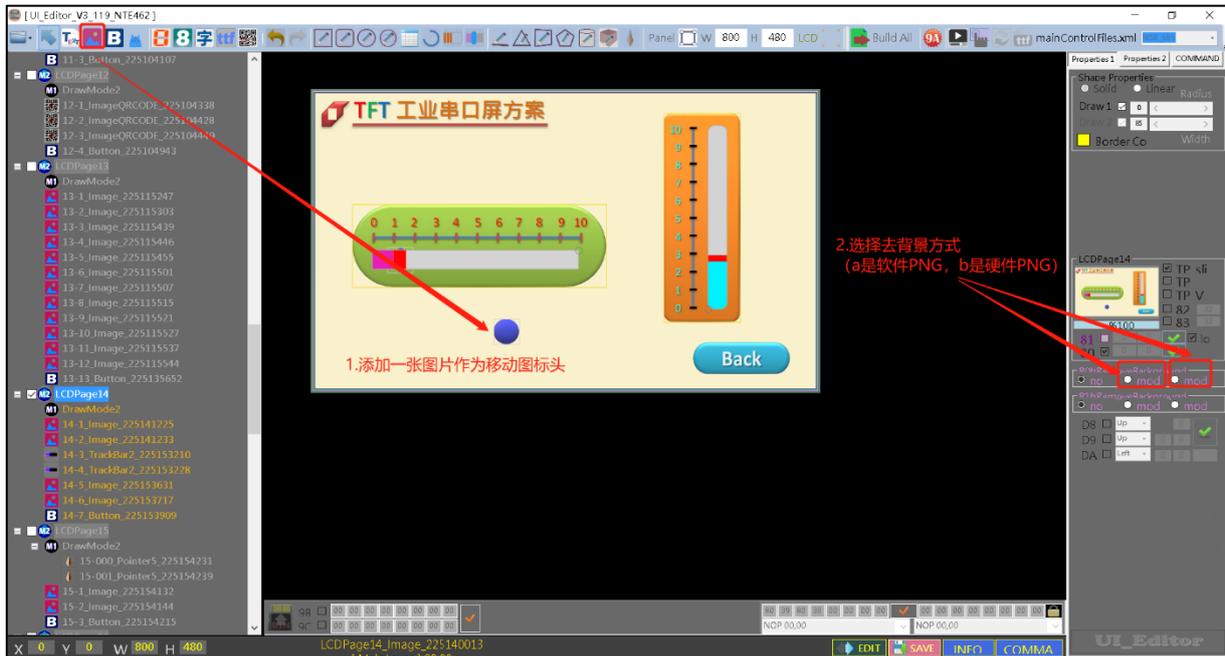


图 4-45：新增滑动条的移动图标头

根据用户需求将滑动条与背景图片与滑动用图标头各自对齐，如下图参考。

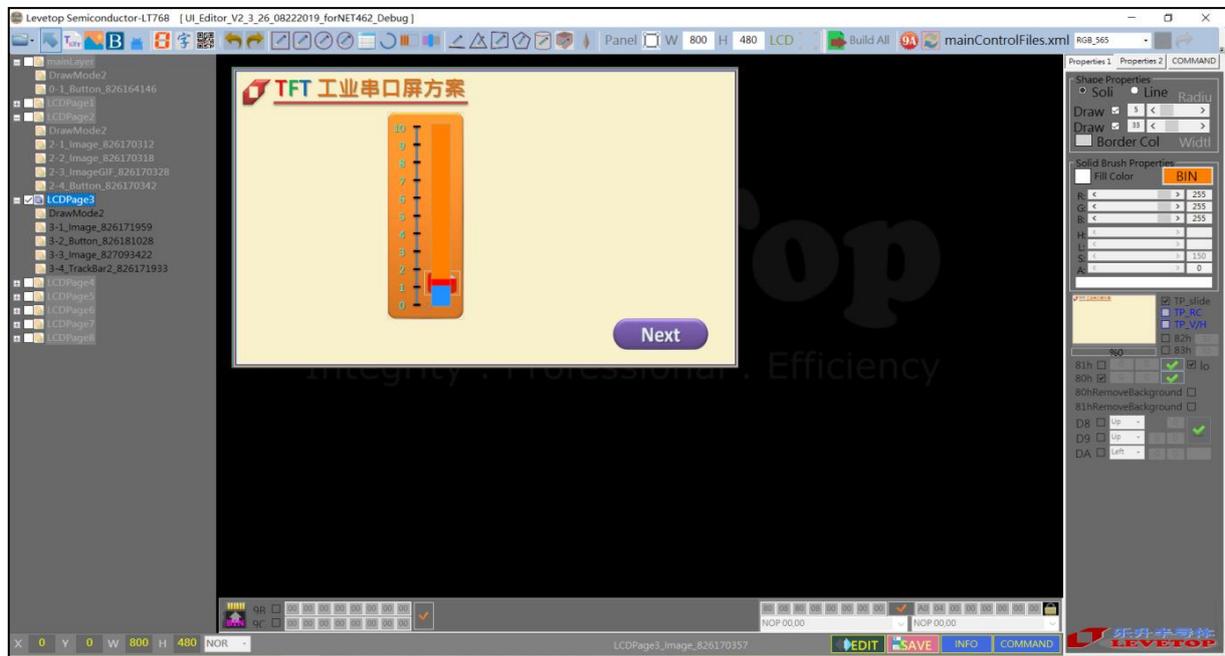


图 4-46：背景图片与滑动图标头各自对齐

5. 选择滑动条背景图片 Command 与滑动图标 Command, 请先点选滑动条以确保是要设定滑动条背景图片与移动图标模式, 接着点选屏幕下方 Command 选择窗口, 如下图所示,



图 4-47: 设定滑动条背景图片与移动图标模式

Command 选择窗口, 共分两组, 第一组是背景图片 Command, 第二组是滑动图标头 Command。如下图所示, 只要分别点选再展开 Command 列表, 再选择需要的显示图片编号即可。

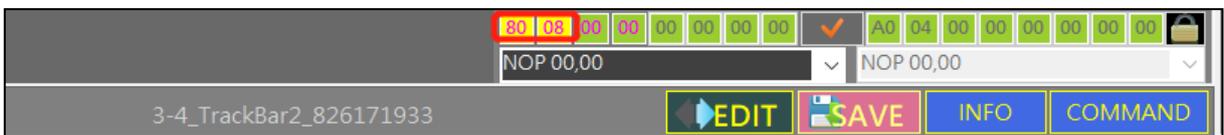


图 4-48: 背景图片 Command 选择窗口



图 4-49: 滑动图标头 Command 选择窗口

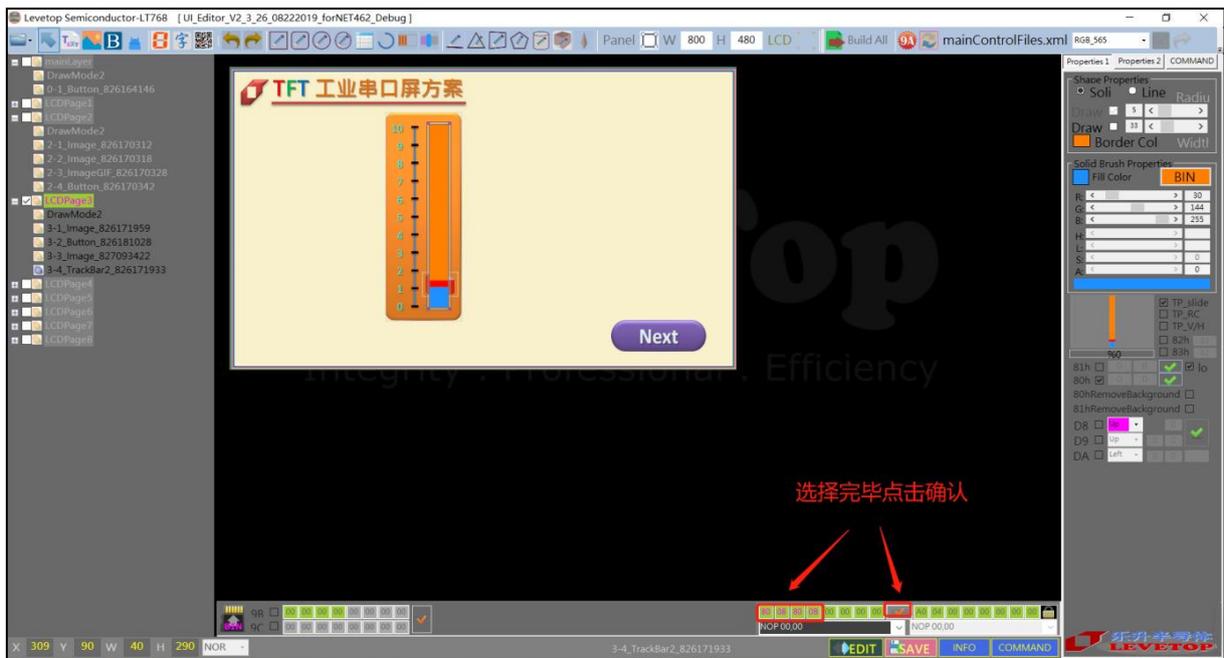


图 4-50: 选择完毕后点击确认

### 4.2.11 设定环形触摸进度条

此功能是结合触控屏的滑动，来改变 TFT 屏上所显示的环形触摸进度的大小，同时反馈信息给主控端。触控进度条底图和触控图标可设置 80 Removebackground，触控图标必须设置在环形触控进度条底图的中心正上分处。

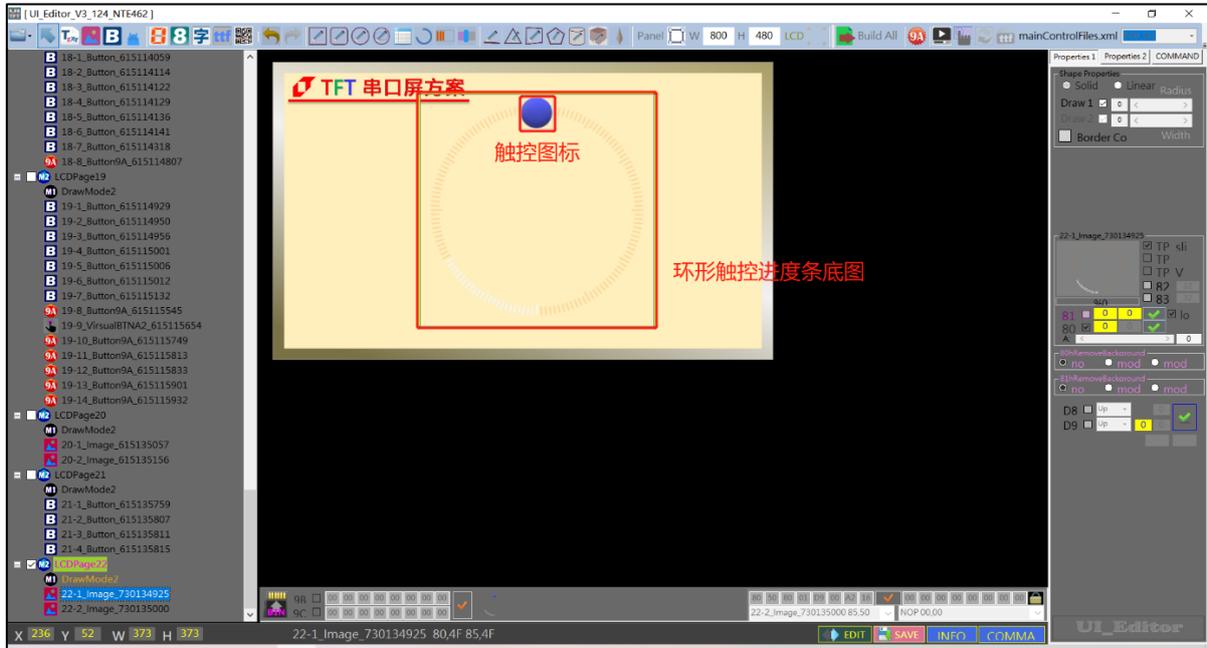


图 4-51：添加环形触控进度条底图和触控图标

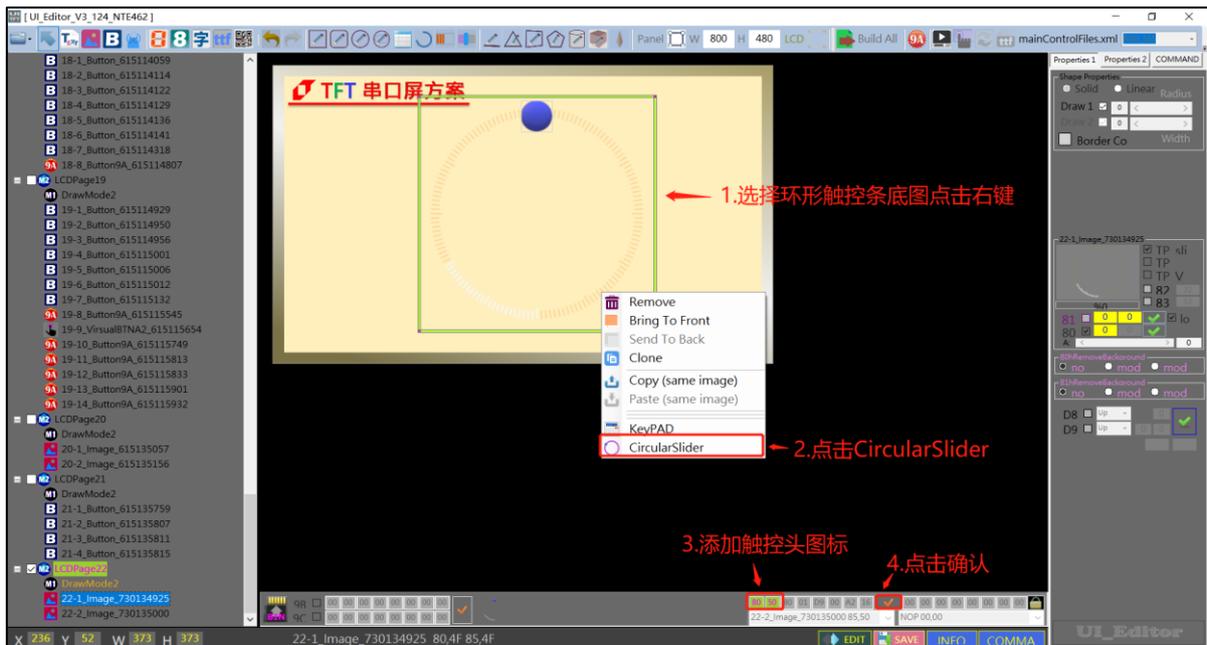


图 4-52：设定环形进度条

### 4.2.12 设定指针图

此功能是设计一个指针图，由主控端传来信息改变指针显示的方向及角度：

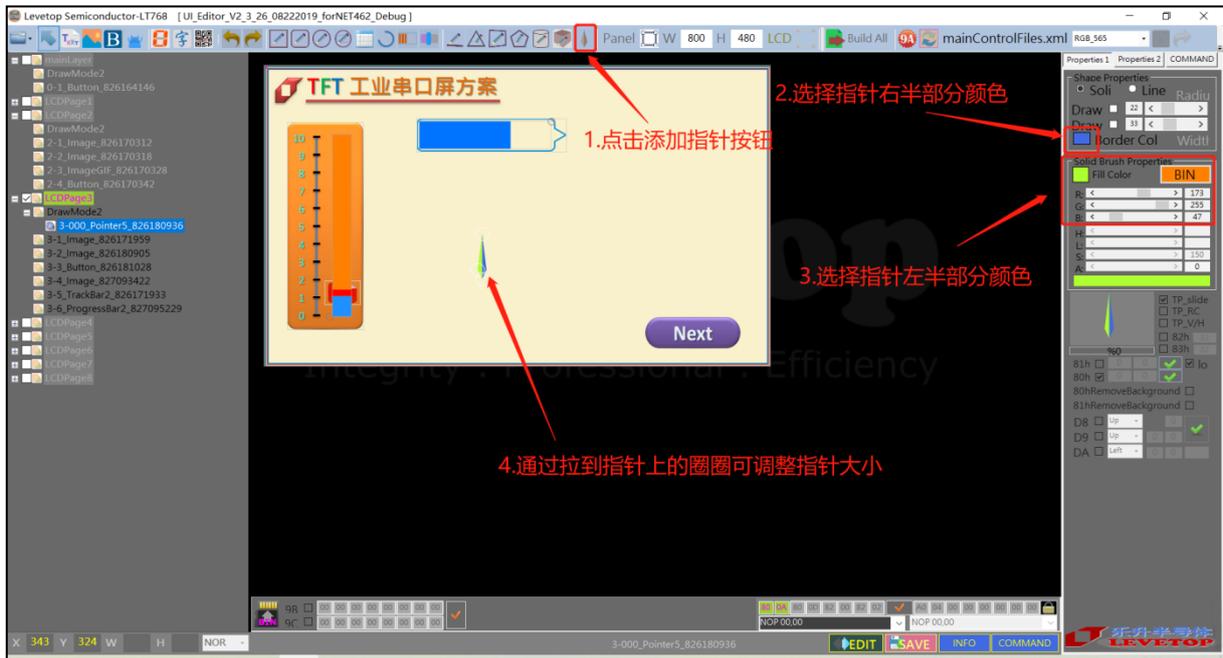


图 4-53：设定指针对左右两侧的背景颜色

1. 点选指针图标，并在屏幕任意位置拖拉移动显示出指针，此时可针对功能需求透过调色盘调整指针左右两侧的背景颜色。
2. 指针上有三个圆圈，当鼠标点选任一点圆圈时即可移动并改变指针尺寸大小。若是鼠标点选至三个圆圈点区域的中间时，鼠标图标会改变成手掌图标，此时表示指针是可以移动的。但若是点选指针都没有任何反应，请先点选左上角树状结构目录上“mainLayer”下的“DrawMode2”，再点选预计选择的指针即可。(EX :3-000\_Pointer5\_826180936)

3. 添加指针背景图，请先点选图标并在屏幕拖拉出一张图标外框，此时 UI\_Editor 会自动跳出图片选择视窗，请再点选背景图片即可。如下图所示：



图 4-54：添加指针背景图片

4. 针对功能需求可将背景图片与指针移动重迭显示出仪表的外观，指针也可适时重设大小与颜色，如下图所示：

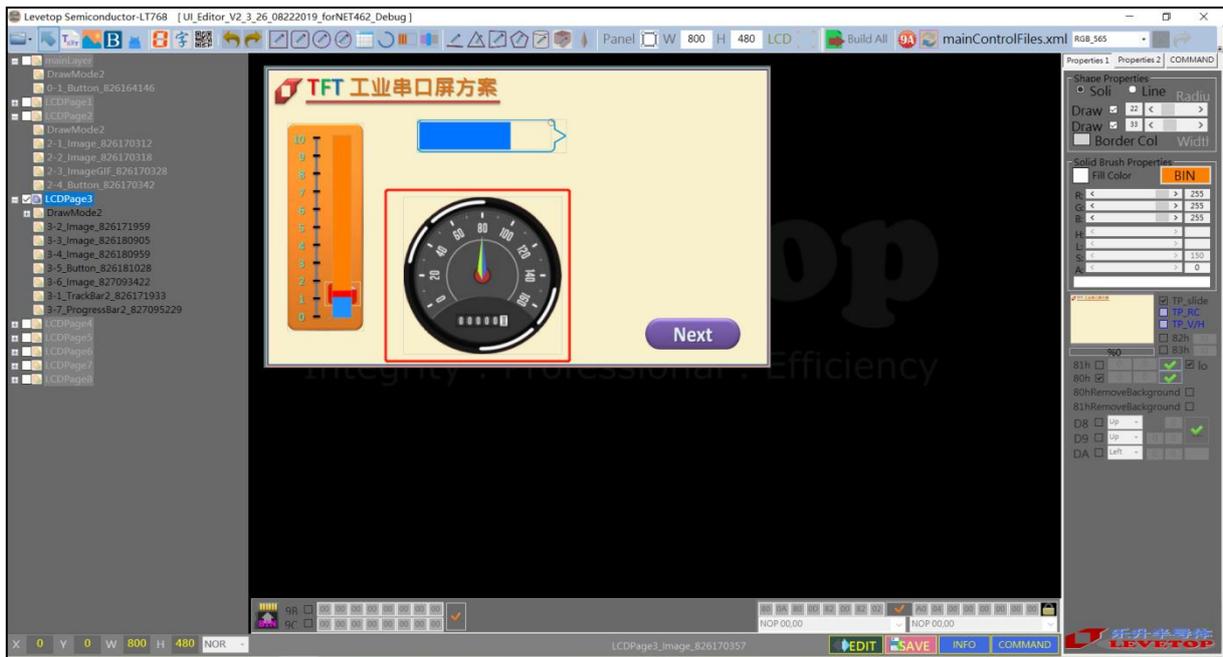


图 4-55：调整指针背景图片显示位置

5. 选择指针背景图片 Command，请先点选指针以确保是要设定指针背景图片模式：



图 4-56：选择指针背景图片 Command

点选第一组数据 (00 00 任意点选一个，数据背景变成黄色)



图 4-57：点选指针背景图片模式第一组数据

接着点选屏幕下方 Command 选择窗口列表，如下图所示，再展开 Command 列表：



图 4-58：为指针添加背景图片指令

### 4.2.13 设定进度条图

此进度条功能与指针图类似，由主控端传来信息改变进度条显示的长短：

1. 点选进度条图标，并在屏幕任意位置拖拉移动显示出进度条，此时可针对功能需求透过调色盘调整进度条背景与进度条移动时的颜色。

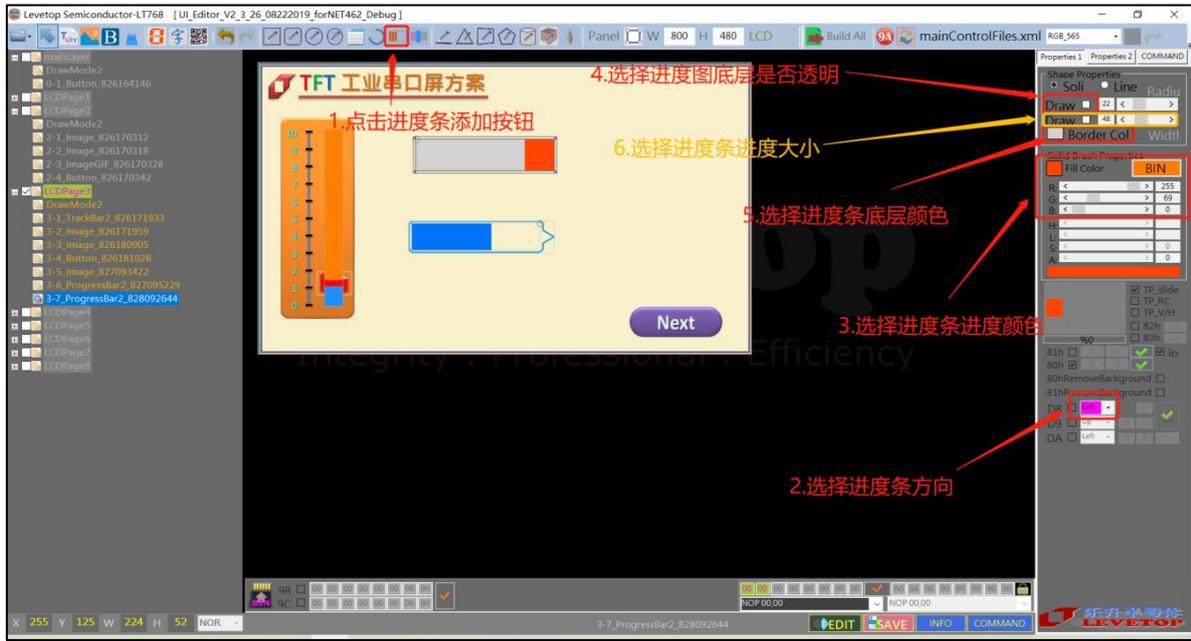


图 4-59：设定进度条背景与进度条移动时的颜色

2. 添加进度条背景图片，如下图所示：



图 4-60：设定背景图片

3. 请针对功能需求将进度条与背景图片进行对齐调整，如下图所示：

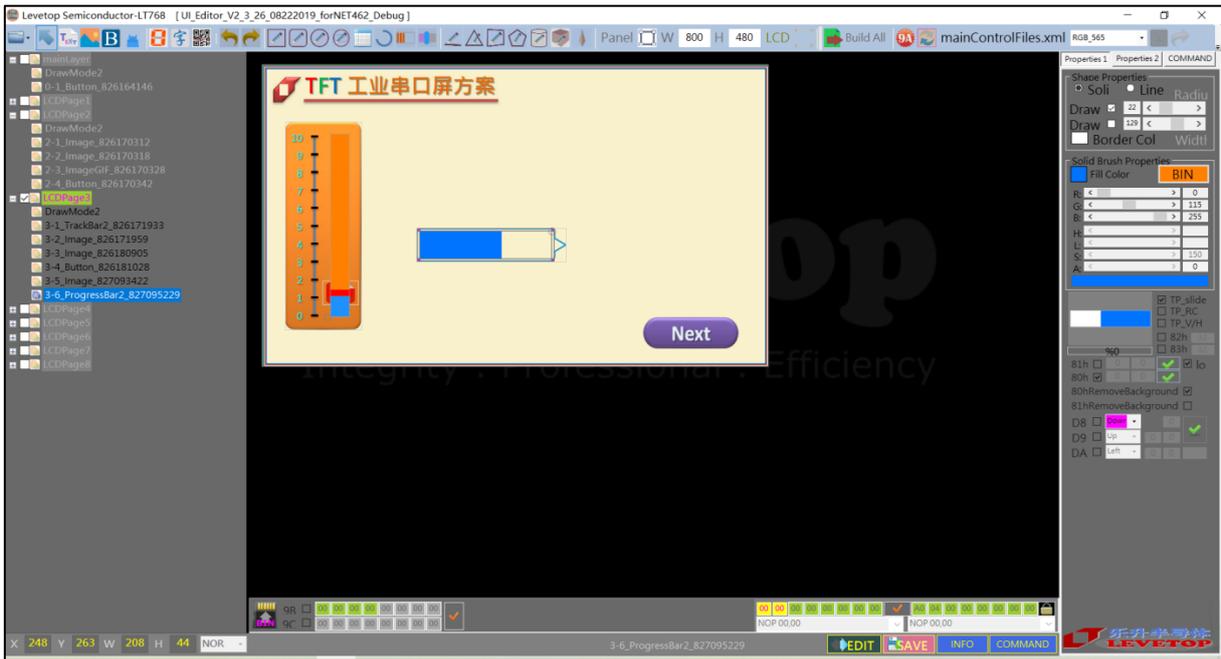


图 4-61：调整进度条背景图片显示位置

4. 选择进度条背景图片 Command，请先点选进度条以确保是要设定进度条背景图片模式，



图 4-62：选择进度条背景图片 Command

点选第一组数据 (00 00 任意点选一个，数据背景变成黄色)



图 4-63：点选进度条背景图片模式数据

接着点击屏幕下方 Command 选择窗口，如下图所示，再展开 Command 列表：

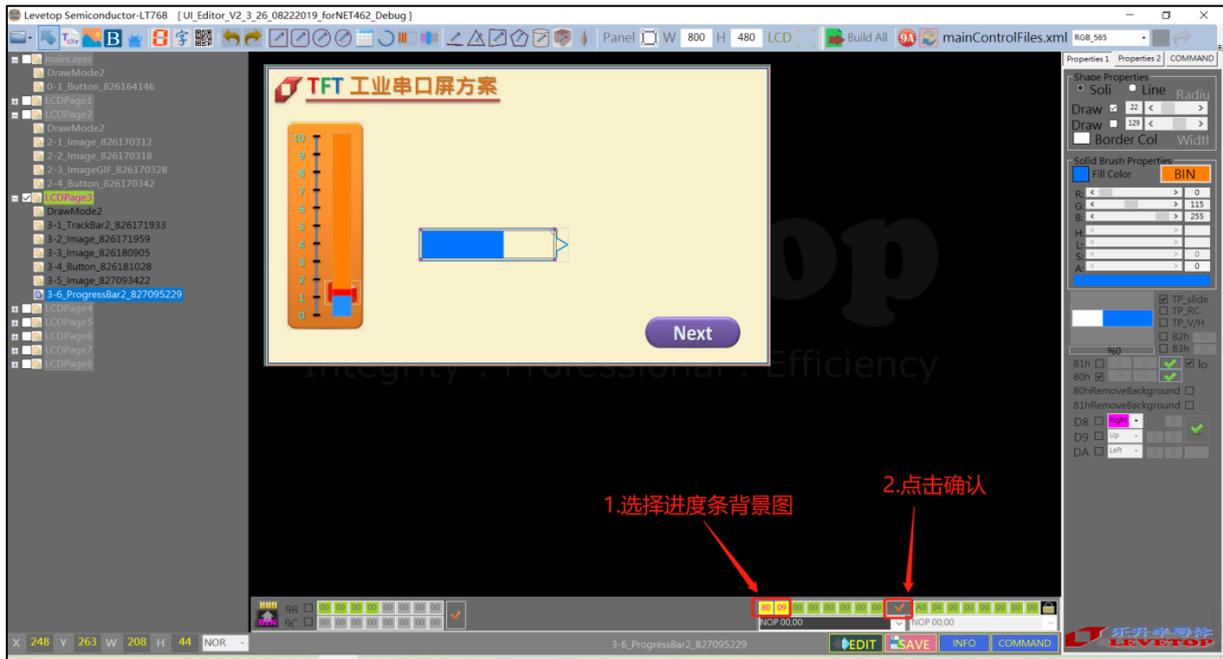


图 4-64：选择进度图的背景图片

#### 4.2.14 设定带透明度的单张图片

成功添加图片之后，可以看到右边操作栏中有 82h 指令的功能框，勾选 82h 指令后，点击右边的小框，接着就可以根据需求来调节透明度大小。



图 4-65：设定带透明度的单张图片

### 4.2.15 控件滑动及显示

成功添加控件之后，勾选左下方 9Bh 指令的小方框，双击小方框右侧的指令栏，根据需求选择相应的指令，9Bh 设定最多可执行 2 个指令。放在同一个图层下的控件则视为处于同一个页，底图只能显示主图层的图片，最后要实现控件滑动的功能，只需要发送 9Ch 的指令即可。



图 4-66: 控件滑动及显示

#### 4.2.16 设定显示数字键盘

数字键盘是由用户交互界面输入信息去反馈给上位机，上位机可以通过反馈信息去执行一些约定的内容，例如开锁等操作。该数字键盘最多可以设置 30 个键盘（包括三个功能键 0x0D: return; 0x08:BackSpace; 0x18:Cancel）每个密码键盘都必须带有三个功能键。用户需要设置最多显示的数字个数，各按键的位置与代表的键值等参数。

添加两张大小和位置一样的图片，一张用于作为数字键盘显示，一张用于数字键盘被按下时在设定的按键区域显示。



图 4-67：添加数字键盘底图



图 4-68: 添加数字键盘底图

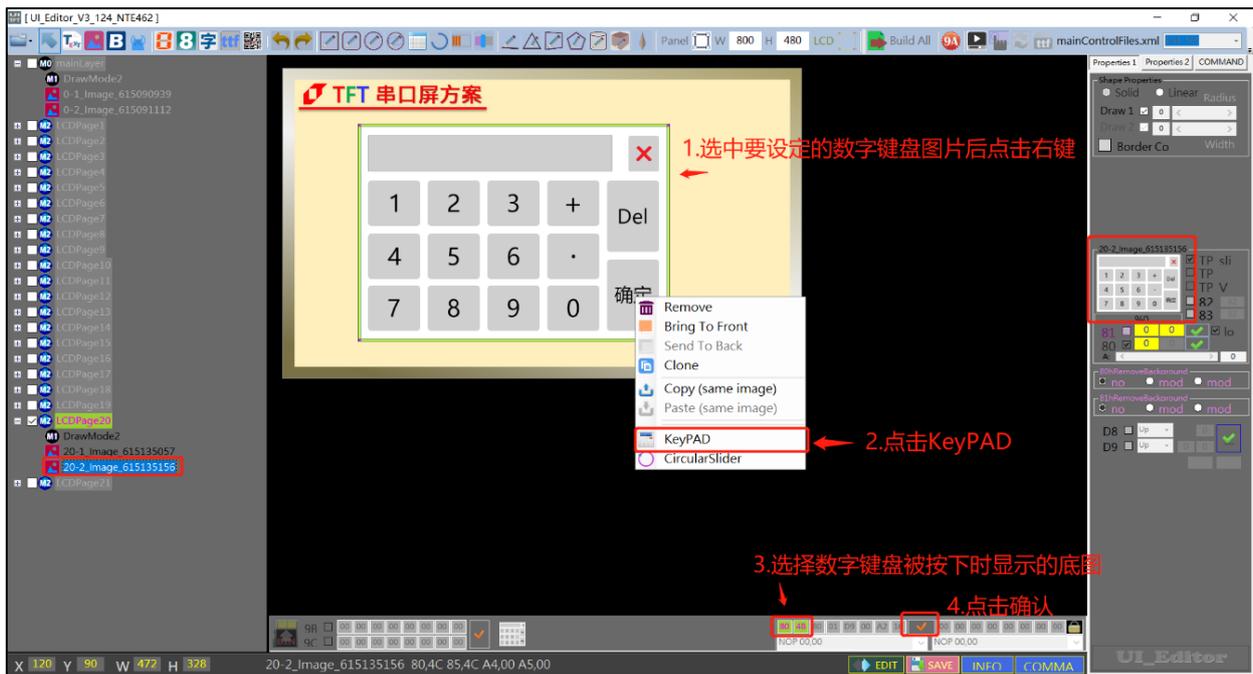


图 4-69: 数字键盘设定



图 4-70: 数字键盘设定

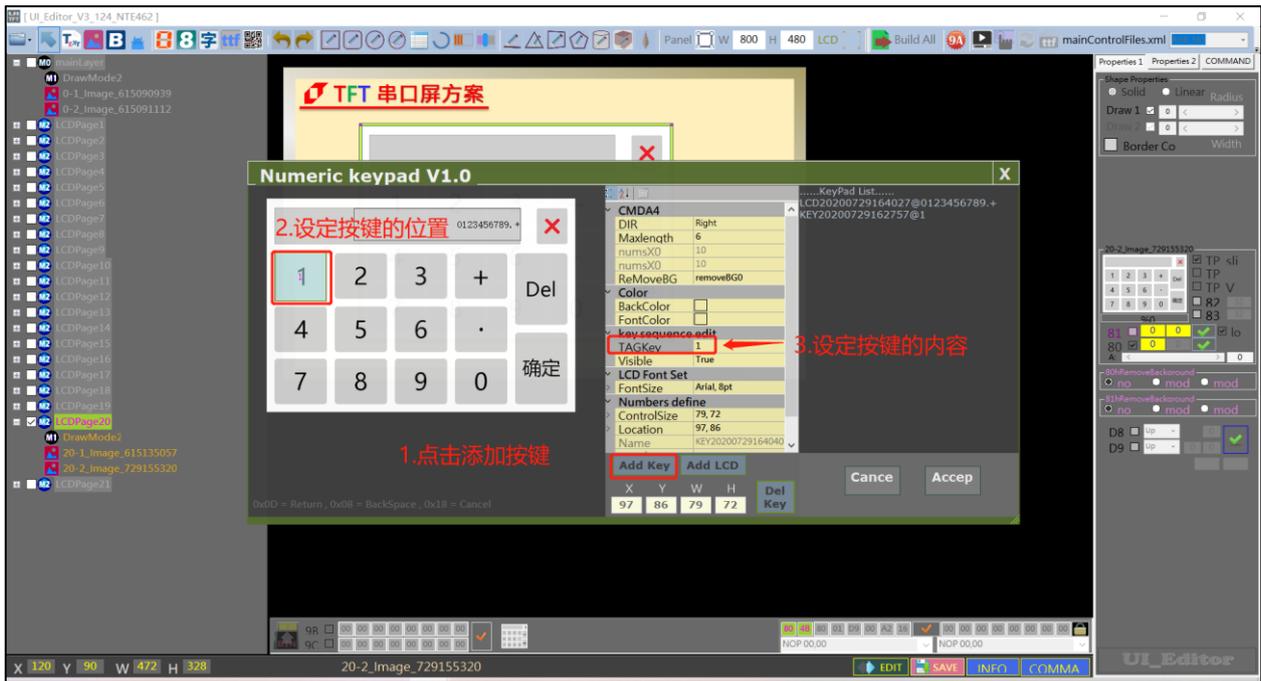


图 4-71: 数字键盘设定



图 4-72: 数字键盘设定

### 4.2.17 实时曲线的设定

B2h&&B3h 指令可实现实时曲线绘制，先由上位机设置好曲线显示的区域及相关参数生成 B3h 指令，再由下位机通过 B2h 指令发送数据进行绘线显示。B2h 发送的指令格式为 B2+Chanel(0-3)(1byte)+Data(N\*2bytes)，曲线数据会更新到对应信道的 B3h 显示。

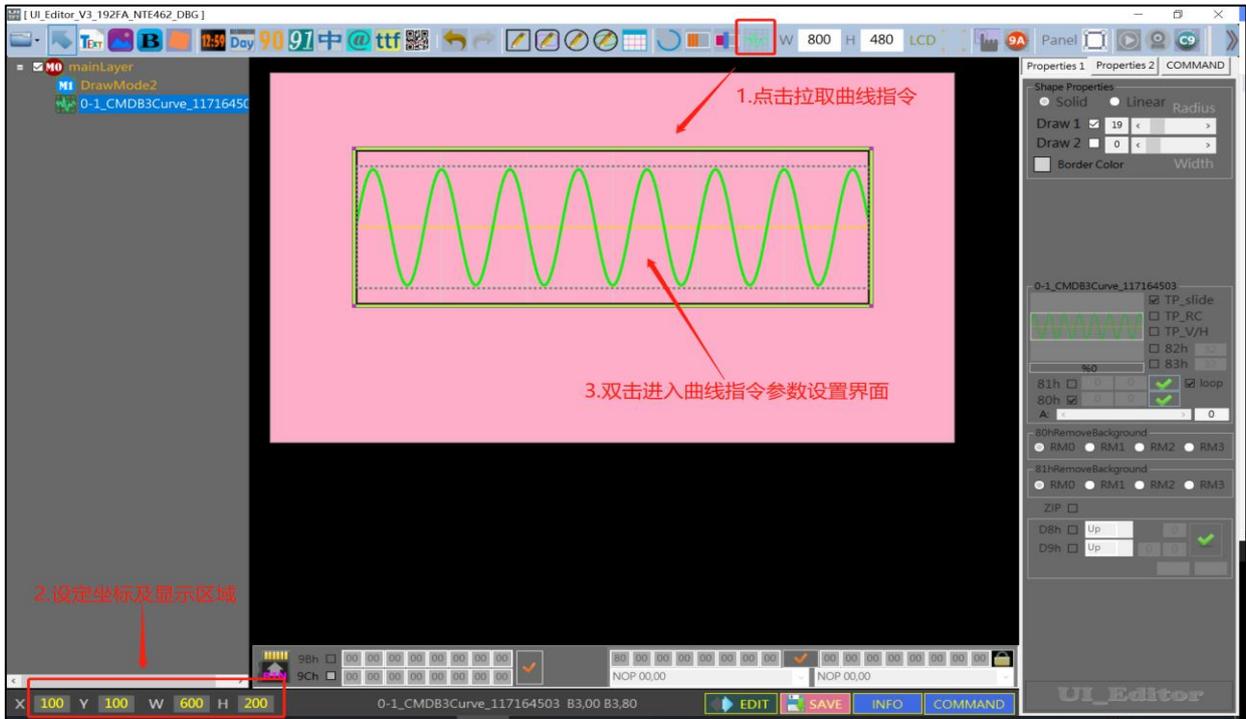


图 4-73: 曲线指令设定

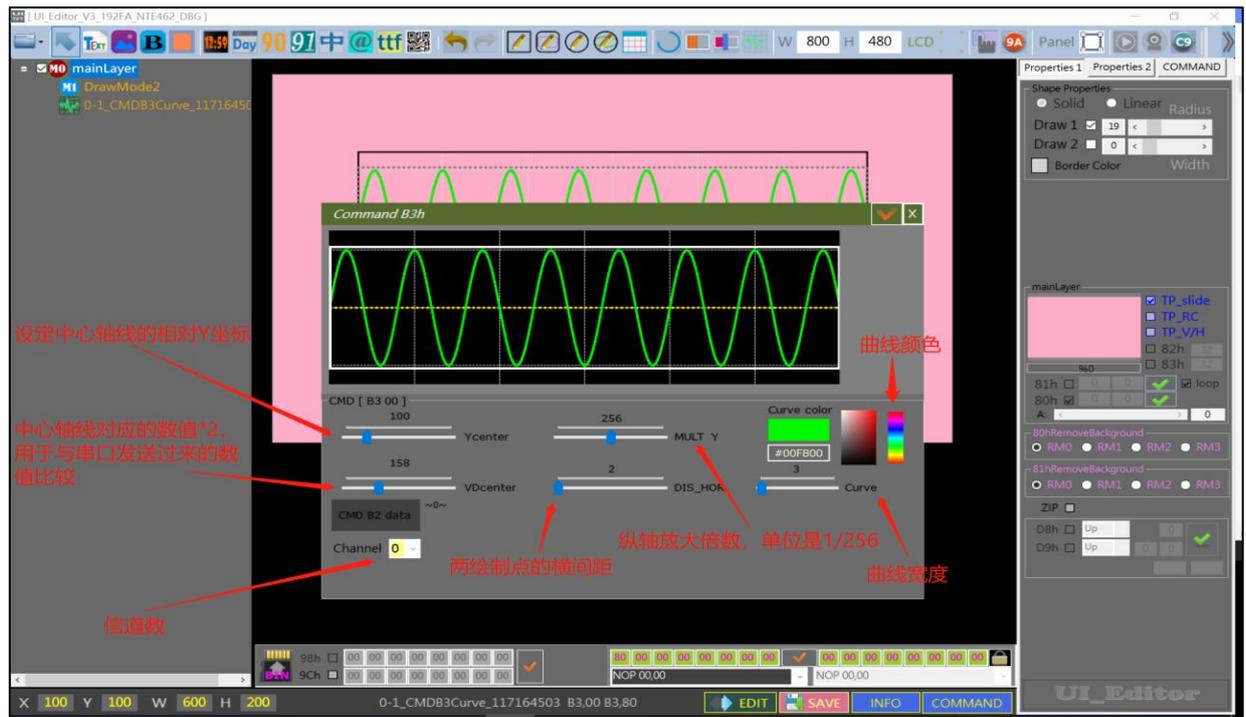


图 4-74: 曲线参数设定

### 4.3 显示文字的设置

#### 4.3.1 图像形式显示文字

图片式显示的意思是用图片来显示文字，处理的对象是图片。是在 UI\_Editor 上将文字生成为图片，在 TFT 串口屏中以图片形式显示文字。这里的文字是指电脑上可以输入的一切文字，包括中文、英文、数字和符号。这种文字显示只适用于显示固定不变的文字。

点击 UI\_Editor 中  添加文字的按钮，是在屏幕框中拉一个框，松开按键后会弹出一个设置框。在框中输入要显示的文字。可以根据需要选择心仪的字体和字号，可以改变字体和背景的颜色，还可以选择背景是否透明。注意：选择了背景透明 (BackColor 勾选代表不透明，不勾选代表透明)，则背景颜色设置失效。确认设置后，要调节显示框的大小使全部文字得以显示。

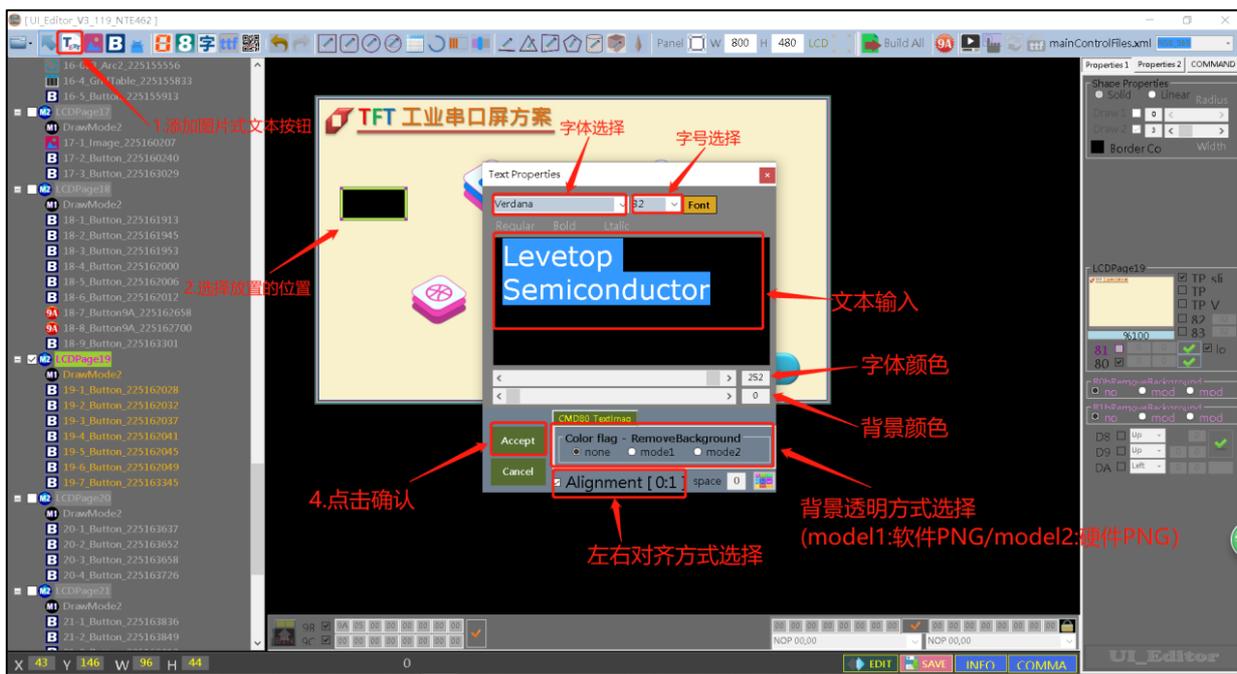


图 4-75: 图片形式显示文字

### 4.3.2 图片式的数字显示

图片式的数字显示是指利用提前制作好的数字图片，根据串口接受到的数据，分析得到应该显示的数字。所以这种数字显示具有多字体和动态显示的优点。

点击 UI\_Editor 中的  按钮。在屏幕框中拉一个框，松开按键同样会弹出一个设置框。这个设置框和上面的提到的设置框有不同。这个设置框中的文字不能改变，同时可以设置数字的对齐方式（左对齐或者右对齐）。Alignment 勾选代表左对齐，不勾选代表右对齐。

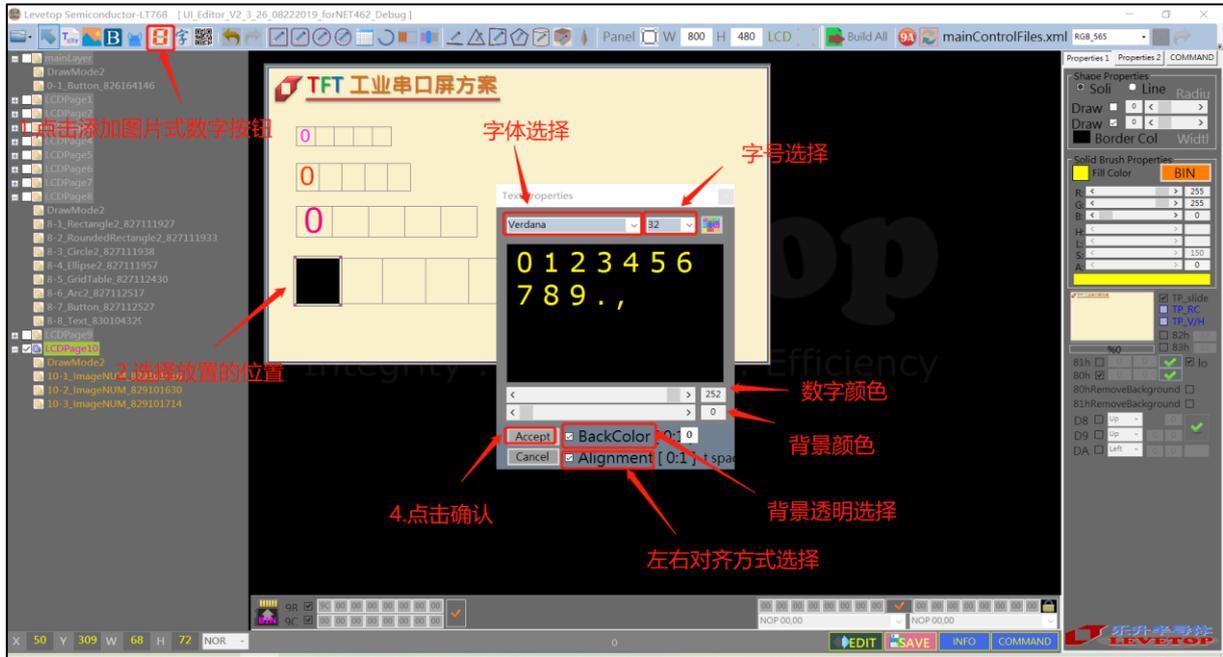


图 4-76：图片式的数字显示的设置窗口

确认设置后，要调整文字框到刚好显示一个数字。如下图所示，每一个小框代表一个数字所占用的面积大小，只需要调整第一个数字框的大小，其他数字框也会跟着改变。在生成 BIN 文件后，用串口和 TFT 串口屏通信的时发送要显示的数字就可以在 TFT 串口屏屏幕中显示出来。



图 4-77：调整图片式数字显示的字体框

### 4.3.3 真彩图片式数字显示

真彩图片式的数字显示是指利用提前制作好“0”~“9”和“.”及“,”的真彩数字图片,根据串口接收数据,分析得到应该显示的数字,所以这种数字显示具有多字体和动态显示的优点。真彩图片存放在 UI\_Editor\ColorNumber 文件内,可以由用户去替换真彩图片。

点击 UI\_Editor 中的 **0** 按钮。在屏幕框中拉一个框,该框要与真彩数字图片大小一致,松开按键同样会弹出一个设置框,该可以设置数字图片的来源、数字图片对齐方式(左对齐或者右对齐)和图片背景透明。Alignment 不勾选代表左对齐,勾选代表右对齐,BackColor 勾选表示透明背景,不勾选表示不透明。

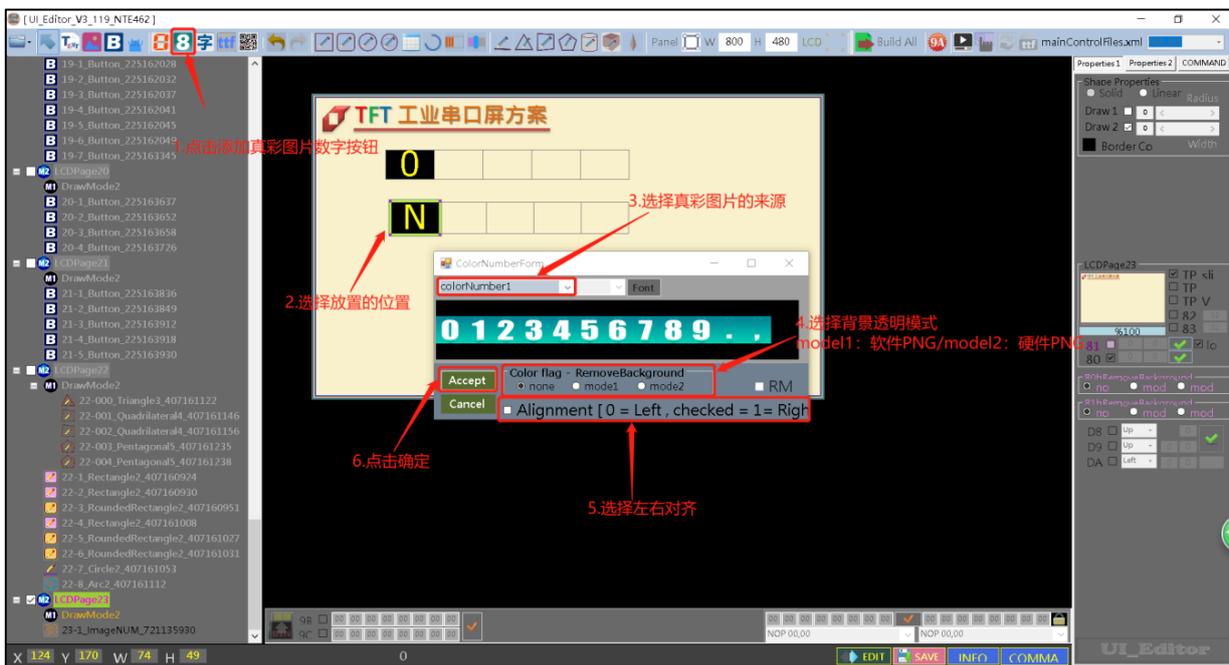


图 4-78: 调整真彩图片式数字显示的字体框

### 4.3.4 使用字库显示文字

动态的中文、英文数字显示是利用 LT776 利用内部或者 Flash 里的字库直接显示。最多可以选择 8 种字库，其中 C0-C3 字库可以显示数字、英文和中文，D0-D1 字库只能显示中文。

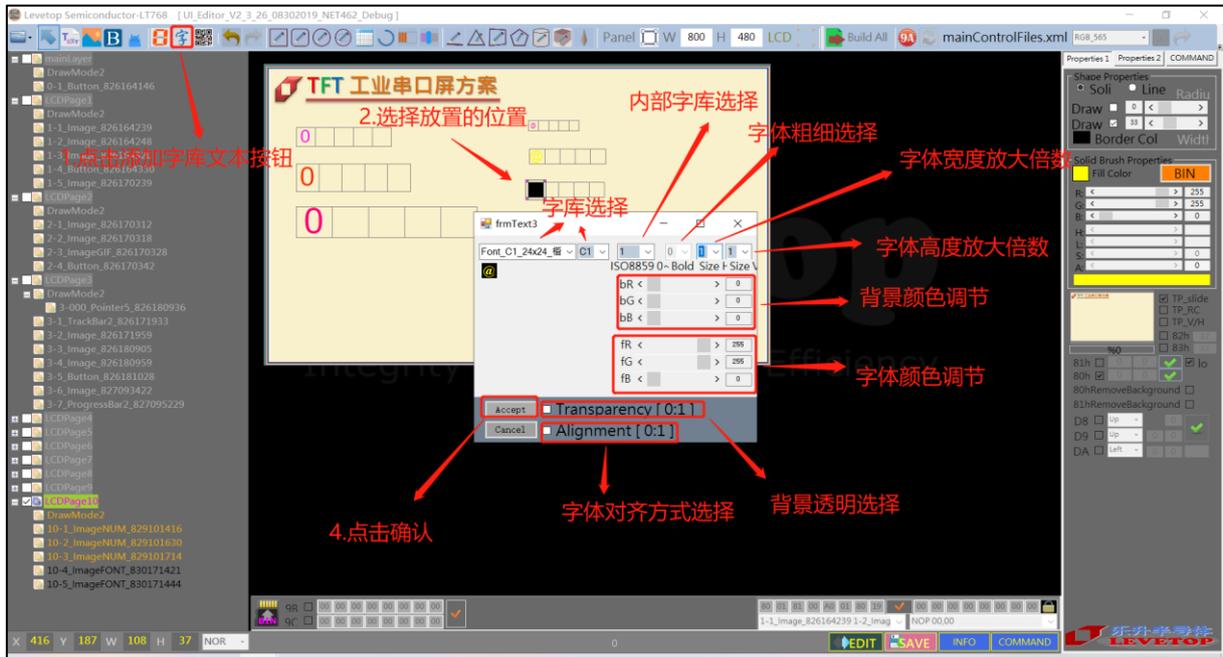


图 4-79：字库显示文字设置

点击文字(字库)添加按钮，点击一下屏幕框，会弹出设置框。框中有相关的各种设置，C0 到 C3 字库的字可以在水平方向和垂直方向分别放大 1 到 4 倍，而 D0 到 D1 的字库不可以。D0 到 D1 的字库可以加粗字体 1 到 2 倍，C0 到 C3 的字库不可以加粗字体。背景颜色、字体颜色、字体对齐等设置都和上面的设置方法一样。

背景透明的设置则和前面的不同，Transparency 被勾选了表示背景透明，不勾选代表背景不透明。设置完成后，文字框中小框的大小不能被改变。在生成 BIN 文件后，用串口和 TFT 串口屏通信的时发送要显示的文字就可以在 TFT 串口屏屏幕中显示出来。

### 4.3.5 图片式的 ASCII 显示

首先用我司开发的取模软件 `ASCIIBitmap_Font_Generator_Vxx.exe` 取模一组 ASCII 图片，选择字体类型，字体大小，显示方式，随后点击 BIN GEN 进行取模。取模后生成一组 `ASCFONT_xx` 的文件夹，然后把该文件夹复制到 `UI_Editor_XXX\ASCIIFONT` 目录下，按照指定格式修改文件夹命名。以上操作完成后在 `UI_Editor` 引用即可。

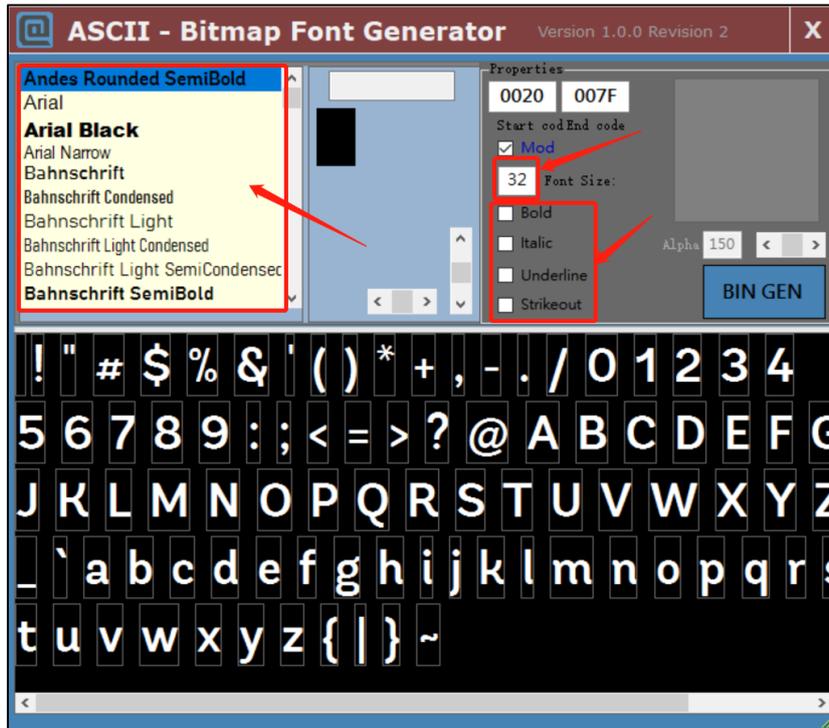


图 4-80: 图片式的 ASCII 显示设置



图 4-81: 图片式的 ASCII 显示设置

点击添加 ASCII 显示控件，在弹出的窗口选取一组 ASCII，设置对齐方式，显示颜色，最后点击 Accep 保存即可。

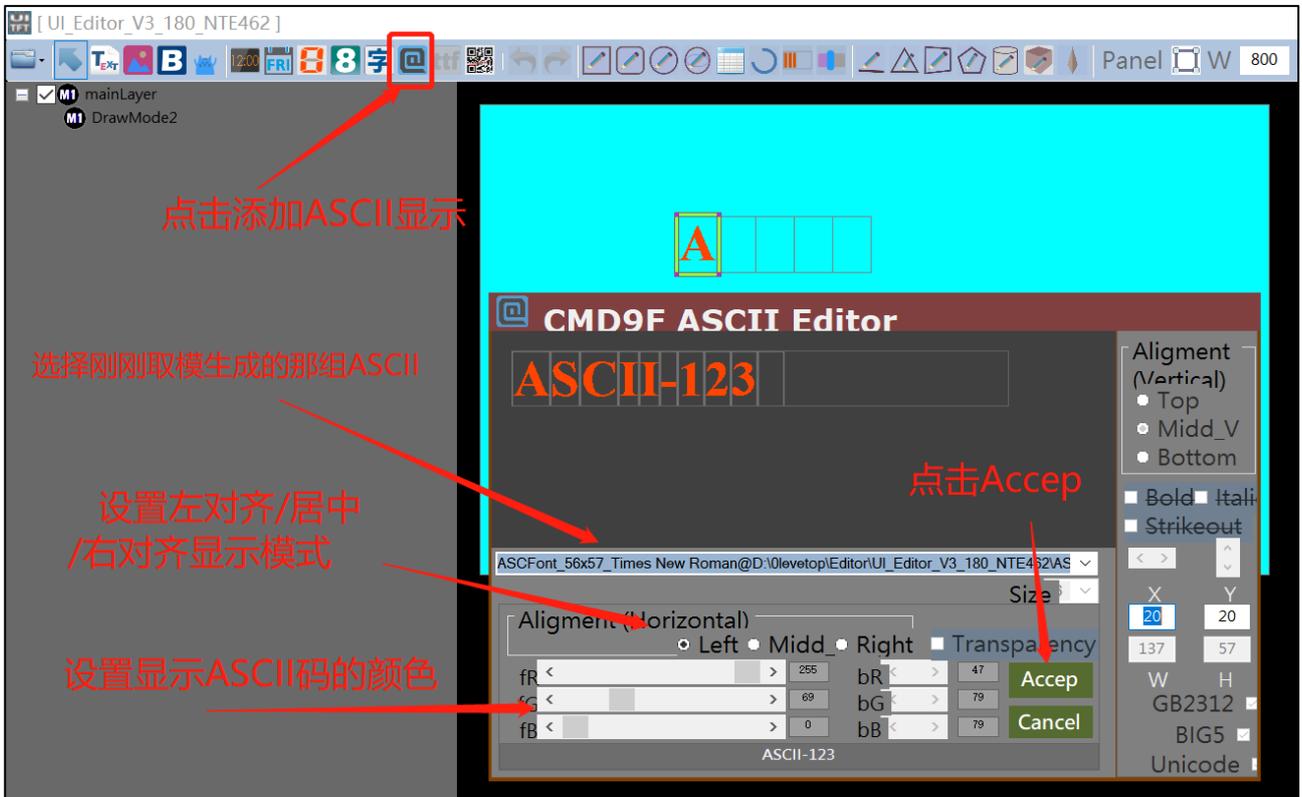


图 4-82：图片式的 ASCII 显示设置

## 4.4 几何绘图功能

### 4.4.1 基本绘图功能

UI\_Editor 一共包含 11 种几何绘图功能，这是 LT776 的 TFT 串口屏具有的硬件功能，其中 9 种类似的功能可以分为 3 类，如下表所示。

表格 4-1：几何绘图功能

	空心、实心、带框实心	空心、实心	带框实心
圆	✓		
椭圆	✓		
矩形	✓		
圆角矩形	✓		
三角	✓		
四边形		✓	
五边形		✓	
圆柱体			✓
方柱体			✓

另外两种功能是画直线和画任意角度的圆环。由于同一类的绘图功能操作基本一致，所以每一类功能只拿一个例子来举例说明。

画圆：点击画圆按钮，在屏幕框中点击并拉动操作框，即可出现圆形。调整外面的操作框可以改变圆的大小。其他的设置都是右边的 Properties1 设置栏中。



图 4-83：画圆设置

其中 Draw1 和 Draw2 的具体作用如下：

Draw1：空心、实心的选择框，Draw2 没有被勾选的前提下，勾选 Draw1 代表选择空心功能，不勾选 Draw1 代表选择实心功能。如果 Draw2 被勾选了，Draw1 将不受控。

Draw2：带框实心的选择框。勾选上则选择带框实心功能，不受 Draw1 控制。不勾选则选择空心、实心功能。只有在带框功能下才可以调节外边框的粗细。

要改变外边框的颜色，只需要点击外边框颜色选项，在其中选择一种心仪的颜色。要调整内部颜色，可以改变颜色调节框中 RGB 三色的数值，选一种喜欢的颜色，最后要按内部颜色确定按钮。画椭圆、画矩形、画圆角矩形、画三角形的操作和画圆是大体类似的，其中调整三角形的大小是通过拉动三角形的三个顶点来实现的。画四边形，先点击四边形添加按钮，在屏幕框中点击一下，就可以产生一个四边形。通过改变四边形四个角的位置可以更改四边形的形状。拖动四边形可以改变位置。假如无法选中四边形，可以在左边的任务栏中单击对应的任务选项，即可成功选中四边形。

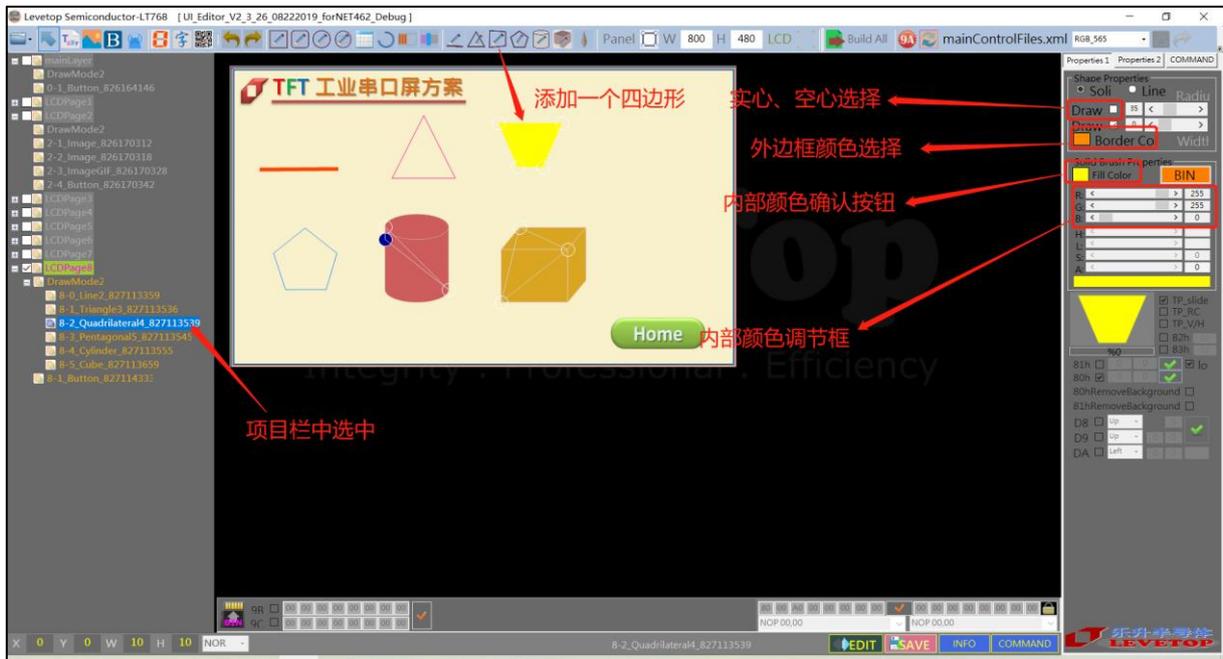


图 4-84：画四边形设置

由于 LT776 的 TFT 串口屏中四边形只有空心和实心功能，所以只有 Draw1 有效，Draw1 被勾选则选用空心功能，Draw1 没有被勾选则选用实心功能。外边框颜色和内部颜色的设置与画圆功能的一样。画五边形的操作与画四边形的操作一样。画圆柱体，点击圆柱体添加按钮，在屏幕框内点击一下，就会出现一个圆柱体，改变圆柱体上的三个点可以改变圆柱体的形状。因为 LT776 的 TFT 串口屏只支持画带外边框实心圆柱体，所以 Draw1 和 Draw2 都失效。边框颜色和内部颜色都和其他绘图功能的一样。

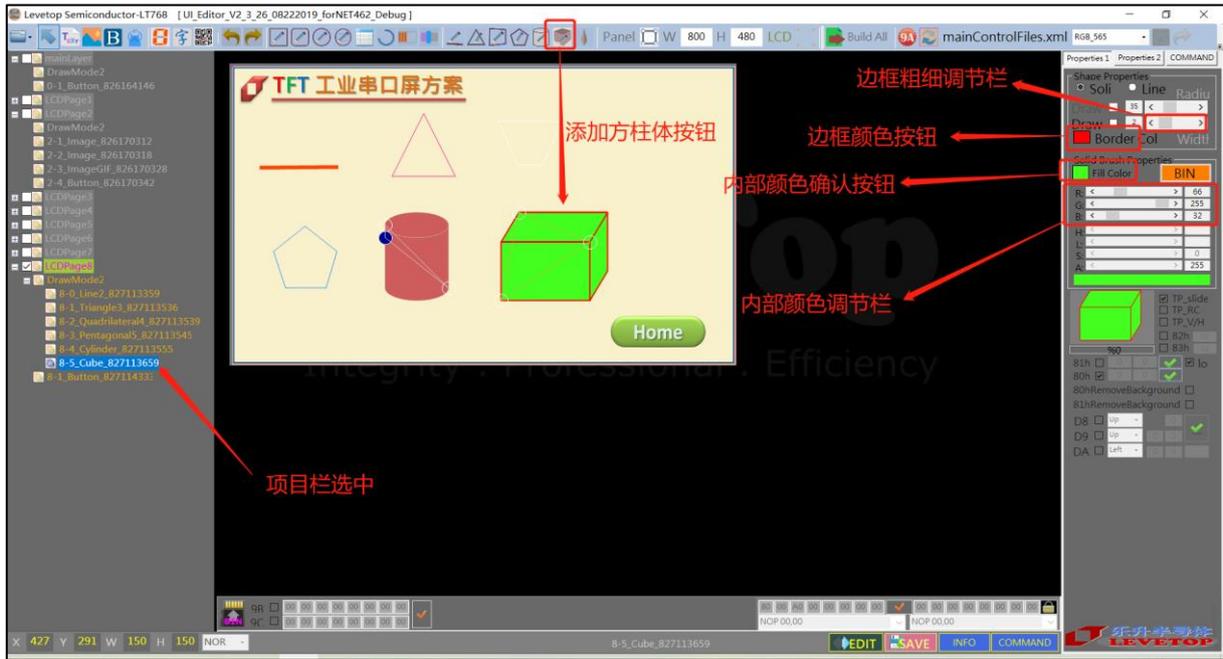


图 4-85: 画方柱体设置

画方柱体的操作和画圆柱体的操作一致。画直线，点击画直线按钮，在屏幕框中点击一下，即可产生一条直线。拖动直线上的两个端点，可以改变直线的长度和方向。改变线条粗细设置框里的数值可以改变线条粗细。线条的颜色可以通过选择颜色按钮里的颜色来改变。



图 4-86: 画直线设置

画任意角度圆环，点击添加圆环按钮，在屏幕框上点击一下，会出现带调整框的圆环。改变调整框的大小可以改变圆环的半径。圆环的粗细和颜色通过改变相应的按钮处的内容即可。在主控端通过串口与 TFT 串口屏连接后，发送指令和圆环的起始角度与转动角度就可以在 TFT 串口屏的屏幕上显示相应角度的圆环。圆环的转动方向是逆时针的，以 12 点钟方向为 0 角度点。

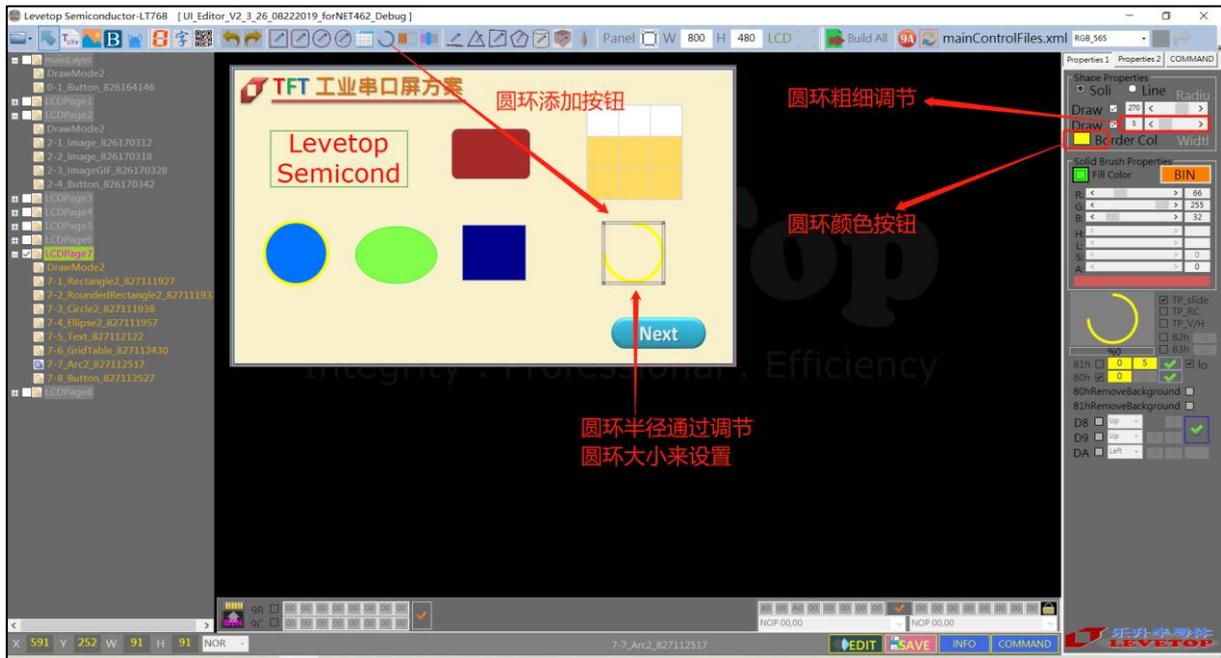


图 4-87：画圆环设置

### 4.4.2 表格制作

按表格添加按钮, 在屏幕框内点击一下就会出现一个表格。表格的大小不能通过拉动边框来调整, 必须使用右边 Properties2 栏中的 N-Col 和 N-Row 来调整。表格的每一个参数调整都可以在 Properties2 中进行设置。其中如果要更换项目栏的方向 (横向、纵向), 先选中表格改变 Properties2 中的 Mode 的勾选状态 (勾选: 横向、不勾选: 纵向), 再点一下表格就改变项目栏的方向。如果要改变项目栏和内容栏的底色, 先选中表格调整对应颜色下面的 RGB 参数, 得到自己想要的颜色后, 点击一下 Fill color 前面的颜色选项即可。



图 4-88: 表格设置

## 4.5 其他功能

### 4.5.1 全屏触摸滑动设定及显示指令

触控滑动控制指令主要是提供触控时，能进行显示画面切换，也就是利用手指在触控屏上进行左右滑动或是上下滑动，显示画面会随滑方向移动，达到平滑切换画面功能，优化显示效能。画面切换是指当前工程中所有图层的底图之间进行切换。

在选中图层，在右边操作栏中找到蓝色的 TP\_RC 和 TR\_V/H 选项。勾选 TP\_RC 代表画面循环切换，不勾选代表不循环切换。勾选 TP\_V/H 代表画面切换是在垂直方面切换，不勾选代表水平切换。



图 4-89：触控滑动设定及显示指令

点击 Command, 打开串口, 点击 B4 指令的 SEND 按钮, 再滑动屏幕, 屏幕画面就会跟着切换。

Item	CMD	Value	Name & ID	CMD	Receive_UARTdata	Parameter
0	80 00		MainLayer	SEND		
1	B4 00		PageSlide0	SEND		
2	A0 00		0-1_Button_826164146	SEND		
3	A1 00		0-1_Button_826164146	SEND		
4	80 01		LCDPage1	SEND		
5	B4 01		PageSlide1	SEND		
6	80 02		1-2_Image_826164320	SEND		

图 4-90：Command 发送 B4 指令

4.5.2 背光控制指令

User Cmd 下面默认有 LCD\_brightness1 BA.0A 指令，其中 0A 的大小可以在添加 BA 指令后修改。

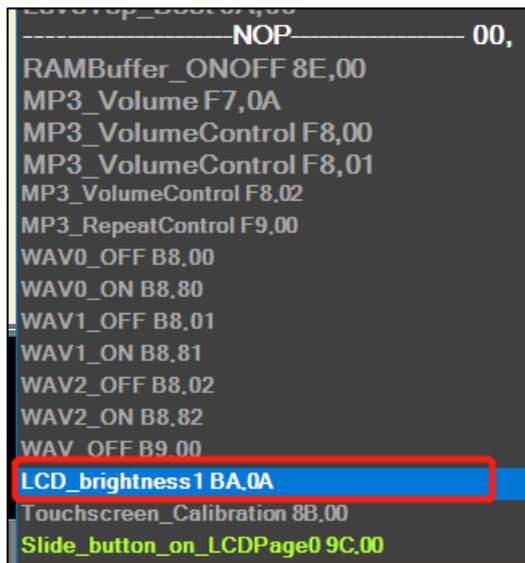


图 4-91: 背光控制指令

背景大小调整是通过选中 BA 指令后，右键弹出可变的范围去选择。

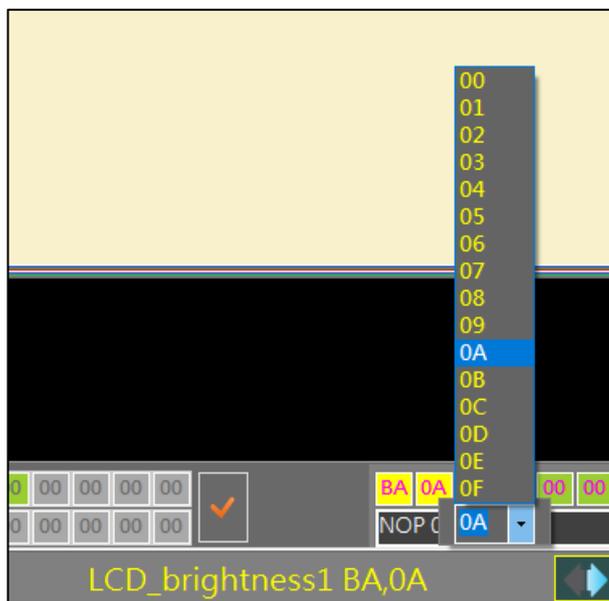


图 4-92: 背光控制指令

### 4.5.3 起始控制指令

当用 Command 发送指令 BCh 00/BCh 02 (00/02 代表 BCh 后面的数值) 给 TFT 串口屏, 那么 TFT 串口屏的显示会被关闭 (Display Off) ; 发送指令 BCh 01 给 TFT 串口屏, 那么 TFT 串口屏的显示会被开启 (Display On) , 如果为 BCh 02 关闭 TFT 串口屏显示, 那么点击触控屏也能开启 TFT 串口屏显示。**注意:** 显示屏 Off 后, TFT 串口屏将不接受任何绘图功能的指令, 直到串口屏收到显示屏 On 指令。

当用 Command 发送指令 BEh 给 TFT 串口屏, 如果串口屏初始化完成或是处于 Ready 状态, 串口屏会回应 5Ah 给 Command。如果串口屏是处于 Busy 忙状态, 串口屏会回应 55h 给 Command。若是 TFT 串口屏与 Command 的 Uart 通讯口未连接则不会响应任何信息。

BC	0	LCD ON/OFF	SEND
BE		Initial Status	SEND

图 4-93: 起始控制指令

### 4.5.4 图片缓冲指令

图片缓冲指令就是通过先在图片在别的图层内缓存完成后, 再去显示到主视窗上, 从而避免了刷图的显示工程。当收到 8E 00 时, 表示要开启缓存区。当收到 8E 01 时, 将缓冲区的内容恢复到主视窗上。选中 8E 指令可修改 8E 指令的序号。

```

-----NOP----- 00,
RAMBuffer_ONOFF 8E,00
MP3_Volume F7,0A
MP3_VolumeControl F8,00
MP3_VolumeControl F8,01
MP3_VolumeControl F8,02
MP3_RepeatControl F9,00
WAV0_OFF B8,00
WAV0_ON B8,80
WAV1_OFF B8,01
WAV1_ON B8,81
    
```

图 4-94: 图片缓存指令

### 4.5.5 声音控制指令

UI\_Editor 会将 UI\_Editor\SOURCE\WAV 路径下的三个 WAV 音频文件分别生成一条 B8h 指令，打开 Command，连接串口。点击对应 B8h 指令的 SEND 键，TFT 串口屏就会播放对应的 WAV 音频，也可以通过按键添加 B8 指令，按下按键播放音频。**注意：**①要有硬件电路支持才能播放音频。②串口屏最多支持 128 首 WAV 音频。



图 4-95: UI\_Editor 里的 WAV 音频文件

B8	00	Wav-1:ON & RP=B7	SEND
B8	01	Wav-2:ON & RP=B7	SEND
B8	02	Wav-3:ON & RP=B7	SEND
B9		Wav-1~3-OFF	SEND

图 4-96: 声音控制指令

```

WAV0_OFF B8,00
WAV0_ON B8,80
WAV1_OFF B8,01
WAV1_ON B8,81
WAV2_OFF B8,02
WAV2_ON B8,82
WAV_OFF B9,00
    
```

图 4-97: 声音控制指令

4.5.6 串口屏侦测指令

串口屏侦测指令只需要透过 UART 或是 SPI 传递到 LT776 的 TFT 串口屏就会执行，它是主控端系统或是主板的固定指令，这指令不需要在 UI\_Editor 上位机软件内设定。

表格 4-2: 起始控制指令

指令功能	指令码	序号	指令参数	指令说明
检查 TFT 串口屏	BEh	--		串口屏初始化完成或是处于 Ready 状态: 5Ah → Ready; 55h → Not Ready (忙状态);
检查 TFT 串口屏版本	BFh	--		读取 LT776 串口屏版本信息: MCU Code 版本(5Bytes) + 串口屏模块. (42Bytes)

当 UART 串口传递指令 BEh 给 TFT 串口屏，如果串口屏初始化完成或是处于 Ready 状态，串口屏会回应 5Ah 给主控端。如果串口屏是处于 Busy 忙状态，串口屏会回应 55h 给主控端。若是 TFT 串口屏与主控端的 Uart 通讯口未连接则不会响应任何信息。

串口指令 BFh 是用来读取 LT776 的 TFT 串口屏版本，包括 5 个 Byte 的 LT776 程序码版本，及 42 个 Bytes 的串口屏模块信息。当 UART 串口传递指令 BFh 给 TFT 串口屏，串口屏则依序送出这些信息给主控端 MCU。详细的版本信息如下表：

表格 4-3: 串口屏版本信息

Items	No.	定义	说明	举例	
MCU Code 版本内容	1	年		07	07E3=2019
	2			E3	
	3	月		07	7 月
	4	日		05	5 日
	5	自定义号 0~255		CC	
串口屏保存信息	6	上位机版本号	0x10 代表 V1.0, 0x11 代表 V1.1, 0x12 代表 V1.2	10	V1.0
	7	通信接口	0x00-Uart, 0x01-SPI, 0x02-IIC	00	Uart
	8	通信接口速率	高位在前, 低位在后	01	01C200=115200
	9			C2	
	10			00	
	11	PCB 版本号	0x10 代表 V1.0, 0x11 代表 V1.1, 0x12 代表 V1.2	10	V1.0
	12	MCU	0x00 代表 8 位单片机, 0x01 代表 16 位, 0x02 代表 42 位	02	32bit
	13	MCU 厂家	0x00: Levetop, 0x01: ST, 0x02 = STC	01	STM
	14	Part Number	0x00: 7680A, 0x01: 7680B, 0x02: 7681, 0x03: 7683, 0x04: 7686, 0x05: 7688, 0x06: 268A, 0x07: 268B, 0x08: 3688, 0x09: 7689, 0x0A: 8688	09	LT776
	15	MCU 与 LT776 通信接口	MCU to LTx68 I/F: 0x00 = 8080-8, 0x01 = 8080-16, 0x02 = SPI, 0x03 = I2C	02	SPI
	16	水平像数		01	01E0=480
	17			E0	
	18	垂直像数		01	0110=272

Items	No.	定义	说明	举例	
	19			10	
	20	VBPD		00	
	21		14	0014=20	
	22	VFPD		00	
	23		0C	000C=12	
	24	VSPW		00	
	25		03	0003=3	
	26	HBPD		00	
	27		8C	008C=140	
	28	HFPD		00	
	29		A0	00A0=160	
	30	HFPW		00	
	31		14	0014=20	
	32	PCLKRISING		01	1
	33	HSYNCPolarity		00	0
	34	VSYNCPolarity		00	0
	35	DEPolarity		01	1
	36	RGBSequence	000b : RGB,001b : RBG,010b : GRB,011b : GBR,100b : BRG,101b : BGR	00	RGB
	37	ColorDepth	0x00: 8bits, 0x01: 16bits, 0x02: 24bits	01	16Bits
	38	FlashType	0x00: NorFlash, 0x01: NandFlash	00	NOR Flash
	39	FlashSize	单位: 1MBytes	00	0080=128Mbyte
	40			80	
	41	UI Editor 编译工程版本	0x06: 06 (ex: 06_04_2019-09_24_53)	07	7月
	42		0x04: 04	05	5日
	43		0x14: 20	14	2019
	44		0x13: 19	13	
	45		0x09: 09	0A	时
	46		0x18: 24	11	分
	47		0x35: 53	25	秒

### 4.5.7 User 手动输入指令

双击右下角的 CMD CALL are input by the user 进行 User 手动输入指令模式, 此时只能由 User 输入 CMD, 再次 CMD CALL are input by the user 退出 User 手动输入指令模式, 此时可正常选择指令栏内的 CMD。



图 4-98: User 手动输入指令

### 4.5.8 RTC 数字时钟指令

数字时钟指令 92h，可显示时间和日期，选择一个适合的字体确定后 UI 会自动获取系统当前的信息。当同时显示时间日期的时候，可在设置界面单独设置时间的坐标，在 UI 左下角单独设置日期坐标，效果在模拟器和板子上会呈现出来。

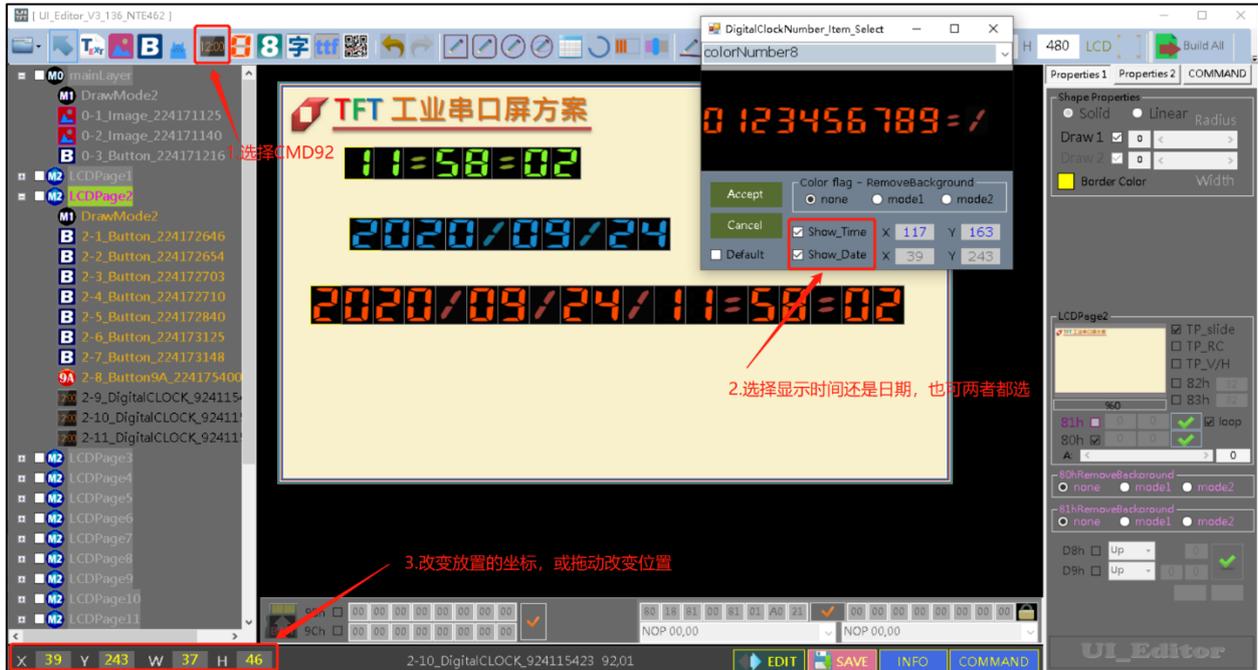


图 4-99: 数字时钟指令 92h 设置

### 4.5.9 RTC 模拟时钟指令

首先拉入两张正方形图片（一张时钟背景图一张中心点图标），然后将中心点图标置于背景图的范围内，

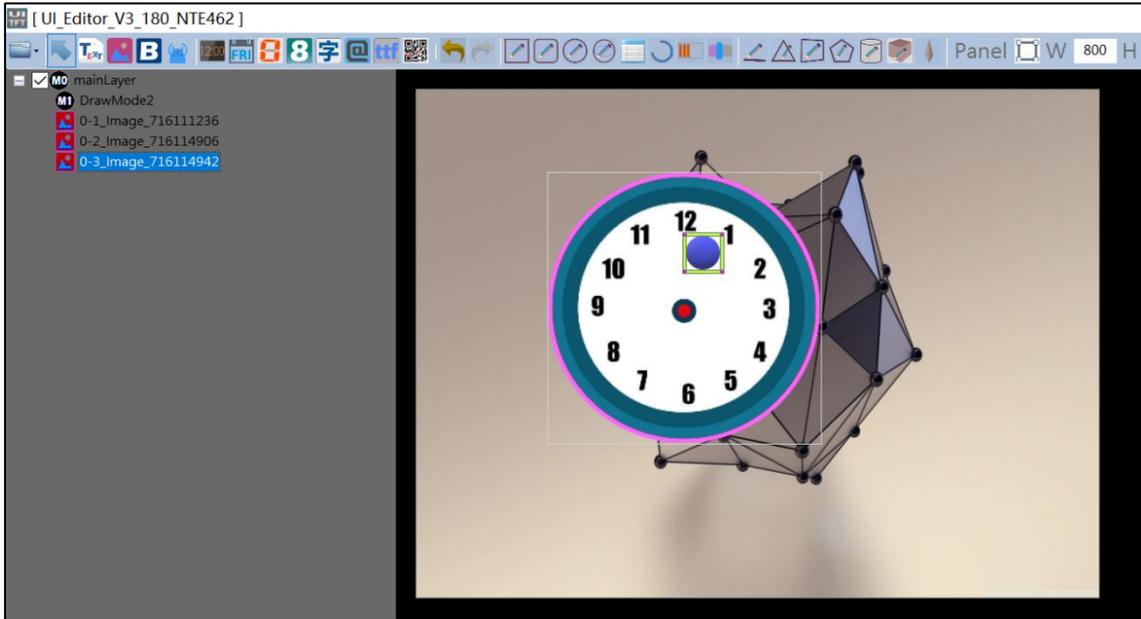


图 4-100: 模拟时钟指令 93h 设置

点选背景图，右击选择 AnalogClock，

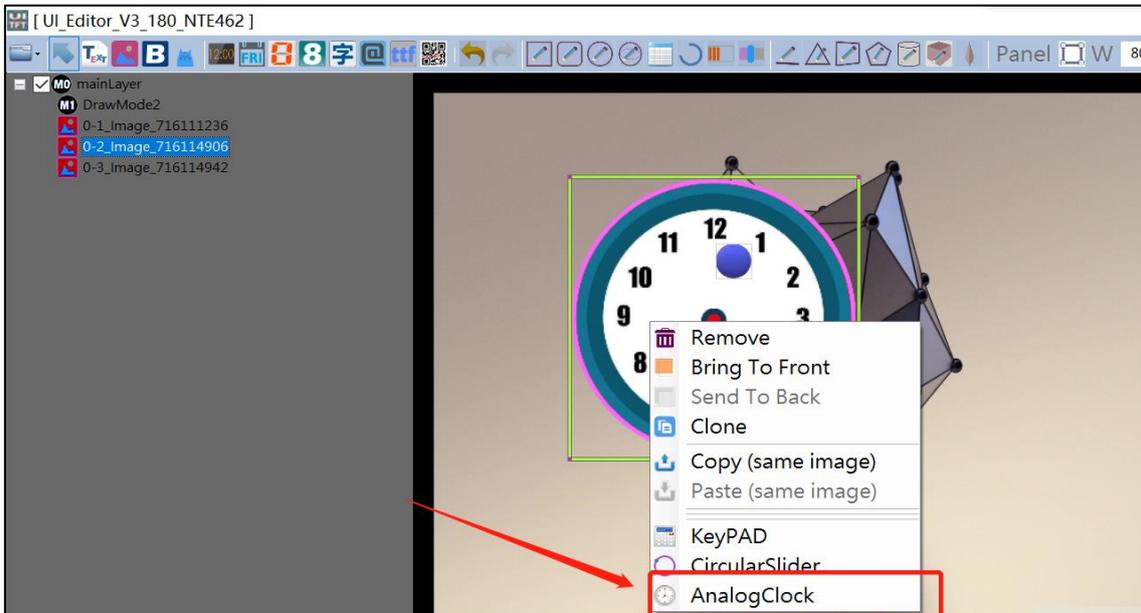


图 4-101: 模拟时钟指令 93h 设置

在下面指令框设置中心点图标的指令，接着双击下边的小时钟图标，

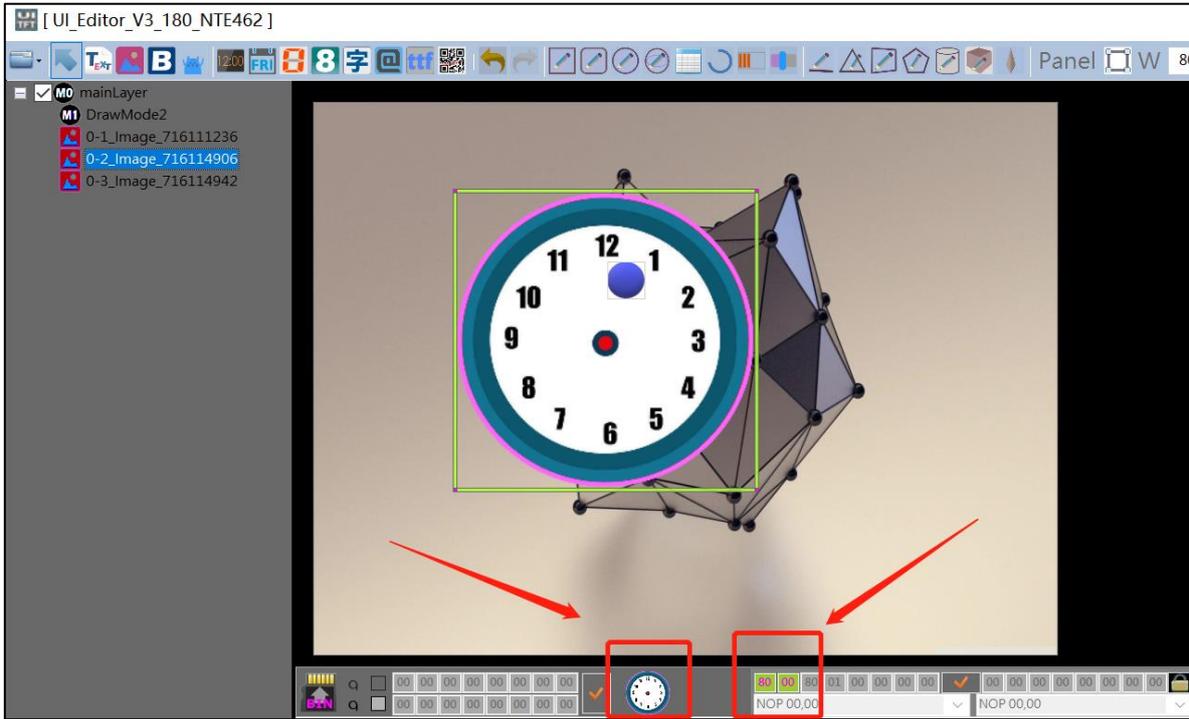


图 4-102: 模拟时钟指令 93h 设置

在跳出来的小窗口中设置时针分针秒针的参数，设置好后点勾选保存完成。

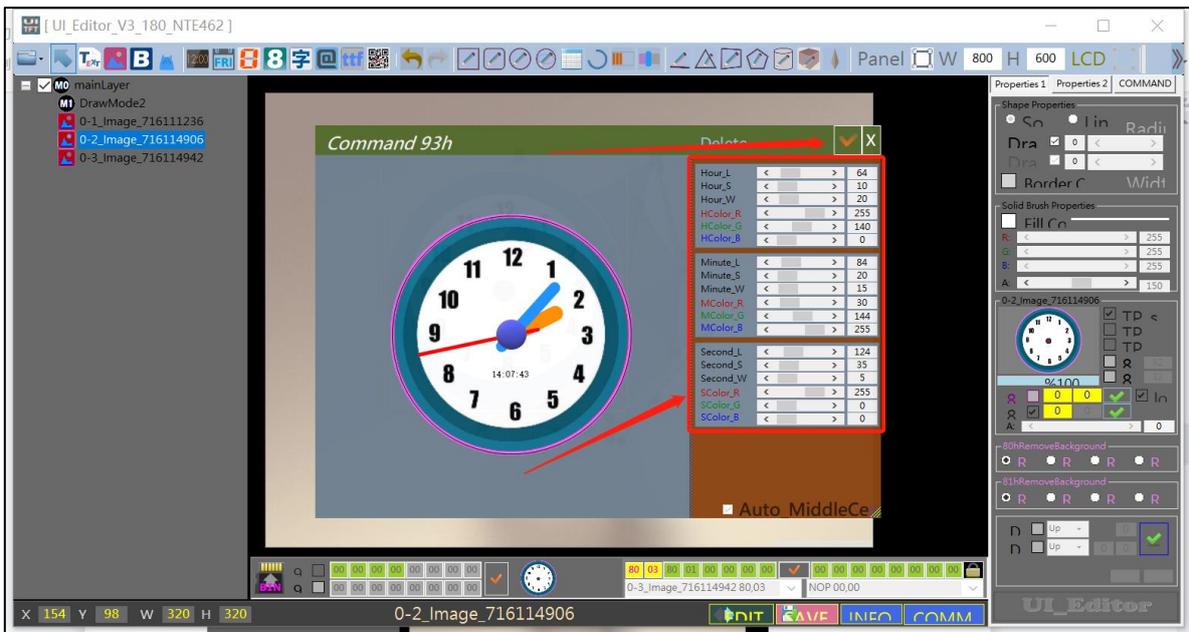


图 4-103: 模拟时钟指令 93h 设置

### 4.5.10 显示星期指令

将准备好的图片素材放至指定路径，文件夹名称及文件夹里面图片命名都要按照指定命名来，

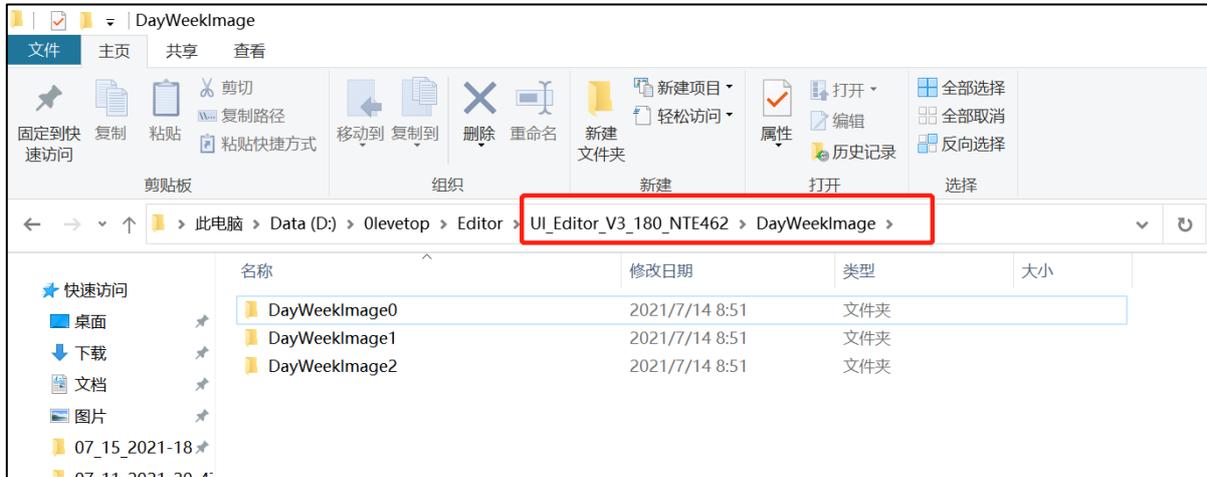


图 4-104：显示星期指令 9Dh 设置

点击添加显示星期指令，在弹出的窗口选择一组准备好的图片素材，再选择图片处理模式后点击 Accep 保存。



图 4-105：显示星期指令 9Dh 设置

### 4.5.11 寄存器操作

CAh ~ CFh 指令主要是对寄存器进行操作，适用于数字变化、亮度调节、音量调节等情况。

CAh: 根据寄存器的值来调用 9A 指令，CAh 00h 即调用指令 9Ah 【00】

CBh: 此指令为指向寄存器，CBh 00h 即指向寄存器 00;

CCh: 此指令为写寄存器，CCh 01h 与 CBh 00h 一起使用即往寄存器 00 写入值 01, 此时【00】变成 01;

CDh: 此指令为读寄存器，CDh 03h 就是读此时寄存器 03 的值，然后上传到主控端;

CEh: 此指令为寄存器内的值加一，CEh 02h 即寄存器 02 此时的值加一，【02】+ 1;

CFh: 此指令为寄存器内的值减一，CFh 05h 即寄存器 05 此时的值减一，【05】- 1。

下面以数字变化为例，讲解如何在 UI\_Editor 操作使用 CAh ~ CFh 指令。



图 4-106: CA-CF 操作举例说明

#### 4.5.12 关闭触控功能指令

9E 指令用于关闭当前界面下所有控件的触控功能，9E A0 表示关闭所有 A0 的触控功能，9E A2 表示关闭所有 A2 的触控功能，此功能适用于某些特定场景。

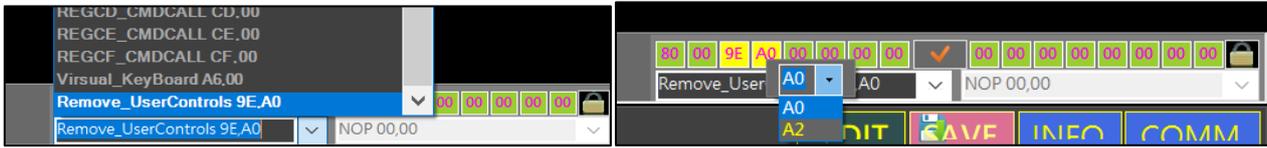


图 4-107: 9Eh 操作举例说明

### 4.5.13 键盘指令

在 UI\_Editor 的 \SOURCE\VKeyboard 目录下放入 4 个文件（默认自带，若没有可询问我司提供）后，在 UI\_Editor 直接设置 A6 指令即可。



图 4-108: 全键盘相关文件

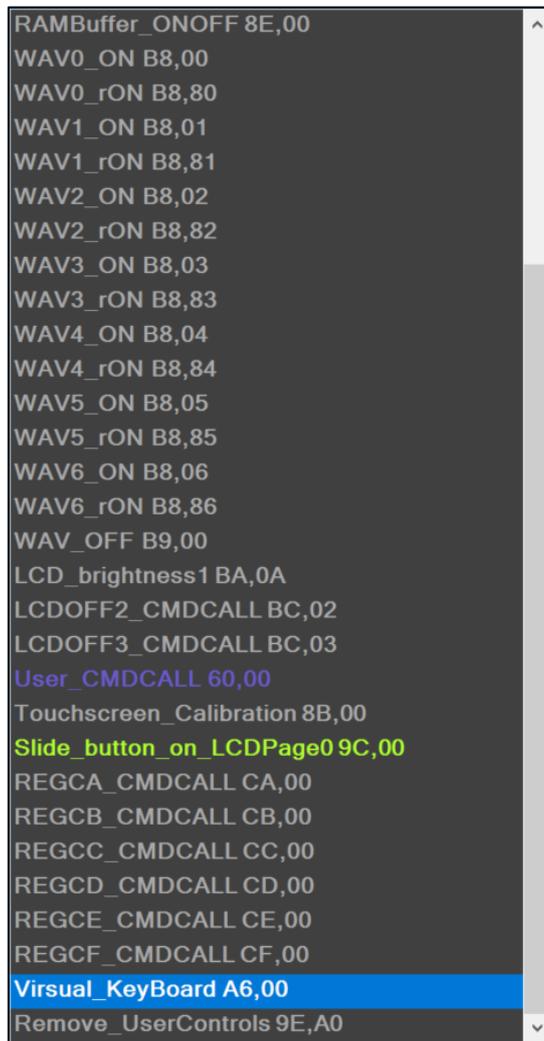


图 4-109: 设置 A6 指令

#### 4.5.14 滑动手势指令

当用户希望使用滑动屏幕去执行指令时，可通过设置手势操作 C9h 指令实现，C9h 支持上滑、下滑、左滑和右滑执行指令的操作。



图 4-110: 设置 C9 指令

## 4.6 WAV 音频、字库文件的说明

### 4.6.1 WAV 音频文件说明

WAV 音频文件存放于 UI\_Editor\SOURCE\WAV 路径下。

名称	修改日期	类型	大小
Wav-1.bin	2018/6/25 10:36	BIN 文件	44 KB
Wav-2.bin	2018/6/10 18:03	BIN 文件	33 KB
Wav-3.bin	2018/6/10 18:04	BIN 文件	657 KB

图 4-111: Wav BIN 文件

有关 WAV 音频文件的制作请参考第 5.3 节的内容。

### 4.6.2 字库文件的说明

字库文件存放于 UI\_Editor\FONT 路径下。其中 fontList.txt 文件的内容是所有字库文件的名称按排列顺序编写，如下图：

名称	修改日期	类型	大小
Font_C0_16x16_宋体.bin	2018/8/13 16:09	BIN 文件	261 KB
Font_C1_24x24_楷体.bin	2018/8/13 16:11	BIN 文件	588 KB
Font_C2_32x32_黑体.bin	2018/8/13 16:12	BIN 文件	1,044 KB
Font_C3_32x32_黑体2.bin	2018/8/13 16:12	BIN 文件	1,044 KB
Font_D0_48x48_微软雅黑.bin	2018/8/13 16:14	BIN 文件	2,349 KB
Font_D1_72x72_新宋体.bin	2018/8/13 16:16	BIN 文件	5,286 KB
fontList.txt	2019/8/30 16:52	文本文档	1 KB

图 4-112: FONT 文件

有关字库文件的制作请参考本文第 5.1 节的内容。

### 4.6.3 真彩数字图片文件说明

真彩数字图片文件存放于 UI\_Editor\ColorNumber 路径下。



名称	修改日期	类型	大小
colorNumber1	2020/3/16 16:47	文件夹	
colorNumber2	2020/3/16 16:47	文件夹	
colorNumber3	2020/3/17 14:39	文件夹	
colorNumber4	2020/3/16 16:47	文件夹	
colorNumber5	2020/3/16 16:47	文件夹	
colorNumber6	2020/3/16 16:47	文件夹	
colorNumber7	2020/3/16 16:47	文件夹	

图 4-113: 真彩数字图片文件

### 4.7 BIN 文件的生成和串口发送指令

生成 BIN 文件之前，要设置 TFT 串口屏中屏幕的参数和主控端与 TFT 串口屏通信的参数。通过点击 UI\_Editor 界面中的 INFO 按钮来打开参数设置。设置完后，点击 INI SAVE 按钮确定设置。

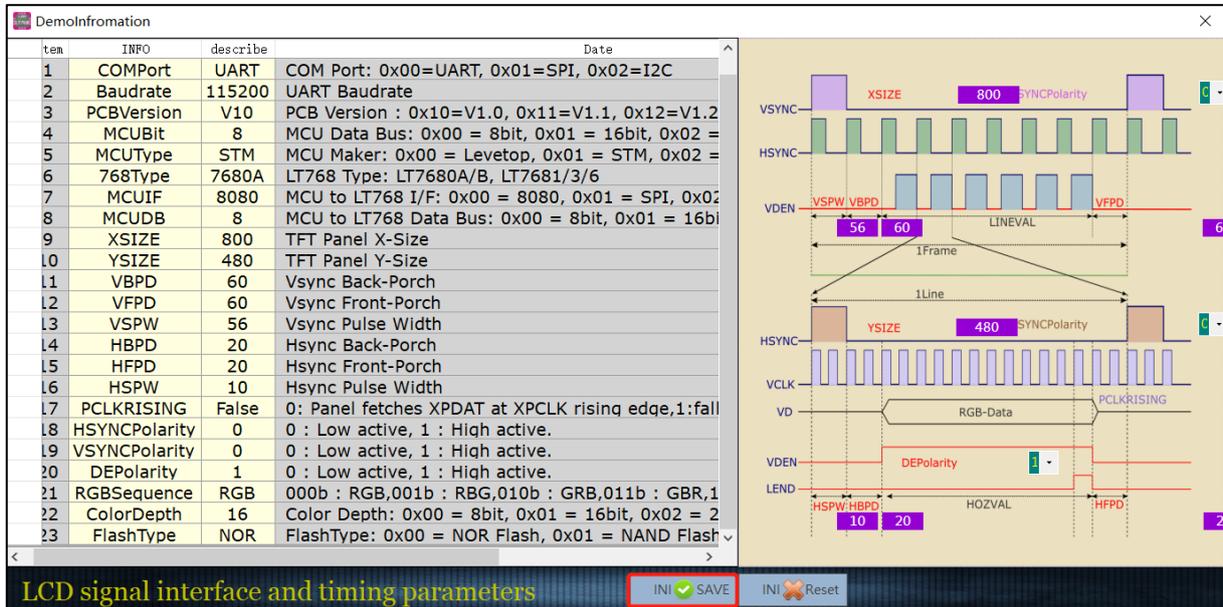


图 4-114: INFO 内配置参数

设置好 INFO 中的参数后，按工程编译按钮。



图 4-115: 工程编译



图 4-116: 工程编译成功

等待工程编译成功，之后再按 BIN GEN 按钮生成 BIN 文件到 BINFILE 下级文件夹中按时间命名的文件夹中。

通过烧录工具将生成的 BIN 文件烧录到 TFT 串口屏中,连接 TFT 串口屏和主控端。点击 Command 按钮,弹出指令发送界面。设置串口波特率,选择串口号,打开串口,就可以发送指令。如果是要带数值的指令, Value 框中会有默认数值,可以修改数值来实现其他的效果。按 SEND 按钮即可发送指令给 TFT 串口屏。

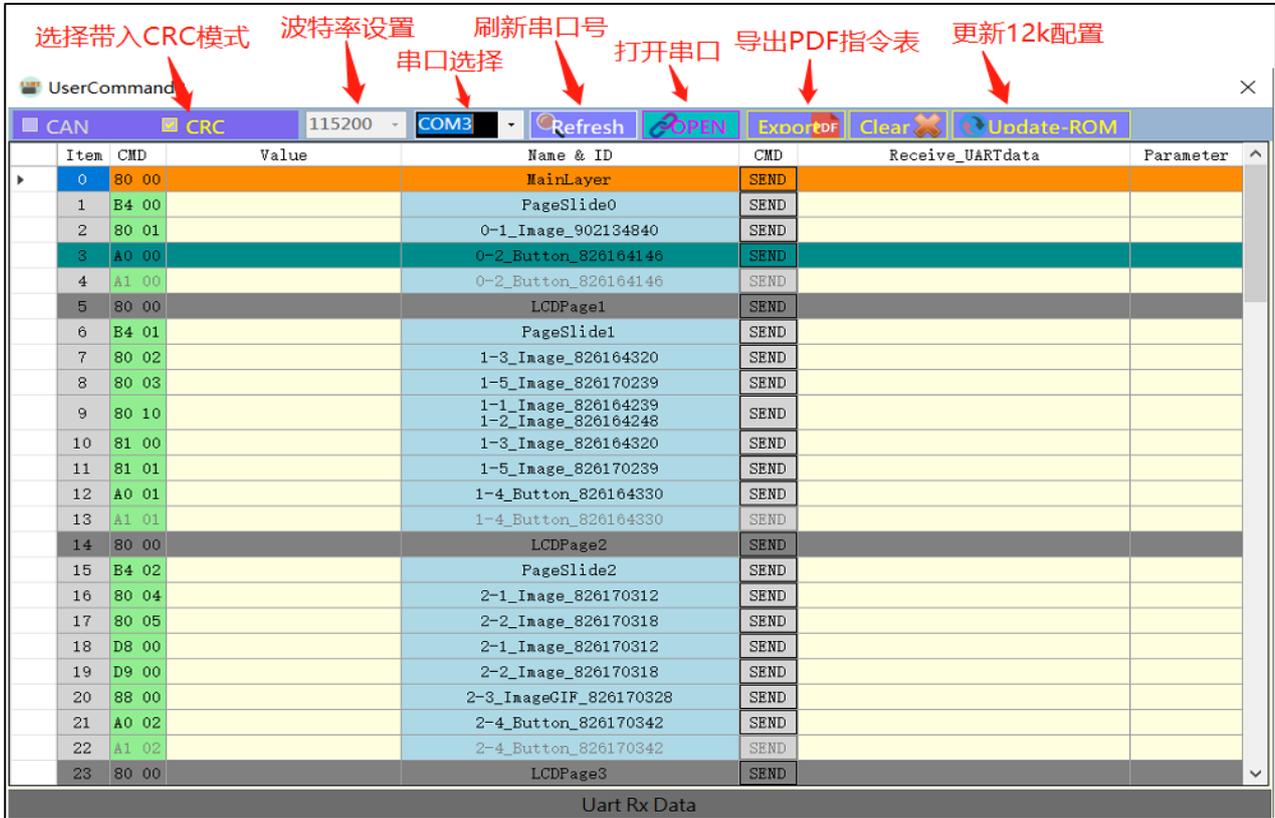


图 4-117: Command 发送指令给 TFT 串口屏

当开发者用 UI\_Editor 将产品的 UI 接口规划及验证后,可以导出整个 UI 所运用到指令表,主控端依据这些指令在其 MCU 程序上植入这些指令格式,在想要显示图片或是要求 TFT 串口屏做出动作时只要送出对映的指令及 CRC 码即可,同时主控端可以依据 TFT 串口屏的响应判断是否被正确执行。下图为设计完成后导出的指令表。

由图上举例来说,主控端 MCU 程序送出 80h、00h、1Bh(CRC1)、98h(CRC2) 指令后, TFT 串口屏就会显示第一张 MainLayer 图片;主控 MCU 程序送出 80h、01h、0Bh(CRC1)、B9h(CRC2) 指令后, TFT 串口屏就会显示第二张图片 Image\_902134840, 依此类推。有关 2 个字节 CRC 的产生方式请参考手册前面第 3.3 节。

**LT7680A : LCD Serial Communication Command Reference**

Author : LeveTop

Run Date : 2019/9/9

Item	CMD	Value	Name & ID	CMD	Receive_UARTdata	Parameter
0	80 00		MainLayer	SEND		
1	B4 00		PageSlide0	SEND		
2	80 01		0-1 Image 902134840	SEND		
3	A0 00		0-2 Button 826164146	SEND		
4	A1 00		0-2 Button 826164146	SEND		
5	80 00		LCDPage1	SEND		
6	B4 01		PageSlide1	SEND		
7	80 02		1-3 Image 826164320	SEND		
8	80 03		1-5 Image 826170239 1-2 Image 826164248	SEND		
9	81 00		1-3 Image 826164320	SEND		
10	81 01		1-5 Image 826170239	SEND		
11	A0 01		1-4 Button 826164330	SEND		
12	A1 01		1-4 Button 826164330	SEND		
13	80 00		LCDPage2	SEND		
14	B4 02		PageSlide2	SEND		
15	80 04		2-1 Image 826170312	SEND		
16	80 05		2-2 Image 826170318	SEND		
17	D8 00		2-1 Image 826170312	SEND		
18	D9 00		2-2 Image 826170318	SEND		
19	88 00		2-3 ImageGIF 826170328	SEND		
20	A0 02		2-4 Button 826170342	SEND		
21	A1 02		2-4 Button 826170342	SEND		
22	80 00		LCDPage3	SEND		

**图 4-118: 指令表导出 PDF 格式**

下表是 LT776 用 UI\_Editor 设计后所能支持的指令总表，如果 UI\_Editor 设计时没用到的功能将不会在导出的 PDF 内出现。

**表格 4-4: 指令总表**

指令码	序号	指令参数	指令功能
80h	nn		显示图片
81h	nn		循环显示重迭图片
82h	nn		显示带透明效果的单张图片
84h	nn		取消循环显示重迭图片
85h	nn		取消 80 指令的显示图片
88h	nn		显示 GIF 图片
89h	nn		取消显示 GIF 图片
8Ah	nn		显示单张或多张图片
8Bh	00		电阻屏校验控制
8Ch		Y, M, D, H, M, S, W	设定时钟 年/月/日/时/分/秒/周
90h	nn	ddd.d	显示单色的数字
91h	nn	ddd.d	设定显示真彩的数字图片
92h	nn		显示数字时钟
93h	nn		显示模拟时钟
94h	nn		显示触控滑条
95h	nn		移除触控滑条。
96h	nn		显示环形触控滑条
97h	nn		移除环形触控滑条
98h	nn	String	显示二维码
9Ah	nn		执行多组的指令,当 nn=00 时为执行开机指令
9Ch	00		执行显示控件滑动
9Dh	nn		显示星期
9Eh	nn		取消控件触控功能
9Fh	nn		显示 ASCII 码图片
A0h	nn		显示控件图片

指令码	序号	指令参数	指令功能
A1h	nn		取消显示控件图片及功能
A2h	nn		设置虚拟控件
A3h	nn		取消虚拟控件及功能
A4h	nn		设置数字键盘输入界面
A5h	nn		取消数字键盘输入界面
B0h	nn	Value(2Bytes)	显示进度条指标图
B1h	nn	Angle (2 Bytes)	显示指针指标图
B2h			更新缓冲区数据
B3h			设定曲线显示
B4h	nn		滑动多张全屏图片
B8h		REP(Bit7) + WAV	播放 Wav 檔
B9h			停止播放 Wav 檔
BAh		BL (00~0Fh)	调整背光亮度
BCh		00 or 01	显示 On/Off
BDh			串口屏复位
BEh			检查 TFT 串口屏
BFh			版本侦测
C0h	nn	String	显示字库-1 文字
C1h	nn	String	显示字库-2 文字
C2h	nn	String	显示字库-3 文字
C3h	nn	String	显示字库-4 文字
C9h	nn		滑动手势
CAh	Reg		执行 9A [Reg] 指令
CBh	Reg		设定指到的寄存器
CCh	Data		写入数据到寄存器内
CDh	00		读取寄存器的内容值
CEh	Reg		寄存器内的数据加 1
CFh	Reg		寄存器内的数据减 1
D0h	nn	String	显示大字库-1 文字
D1h	nn	String	显示大字库-2 文字

指令码	序号	指令参数	指令功能
D2h	nn	String	显示大字库-3 文字
D3h	nn	String	显示大字库-4 文字
D8h	nn		显示卷动出现图片
D9h	nn		显示循环卷动图片
DBh	nn		取消循环卷动图片
DCh	nn	S_Angle, A_Angle	显示环形指标图
DFh	nn	X, Y	显示画点
E0h	nn		显示一条直线
E1h	nn		显示一空心圆形
E2h	nn		显示一实心圆形
E3h	nn		显示一带框实心圆形
E4h	nn		显示一空心椭圆形
E5h	nn		显示一实心椭圆形
E6h	nn		显示一带框实心椭圆形
E7h	nn		显示一空心矩形
E8h	nn		显示一实心矩形
E9h	nn		显示一带框实心矩形
EAh	nn		显示一空心圆角矩形
EBh	nn		显示一实心圆角矩形
ECh	nn		显示一带框实心圆角矩形
EDh	nn		显示一空心三角形
EEh	nn		显示一实心三角形
EFh	nn		显示一带框实心三角形
F0h	nn		显示一空心四边形
F1h	nn		显示一实心四边形
F4h	nn		显示一圆柱体
F5h	nn		显示一画方柱体
F6h	nn		显示一表格视窗

### 4.7.1 SPI Flash 的结构

工程编译之后按 BIN GEN 按钮会在 Binfile 的文件夹中产生 UartTFT\_Flash.bin 档和 UserInfo.bin 档案，这是要让开发者烧录到 SPI Flash 的档案。下图为 SPI Flash 的结构，可以看出 UartTFT\_Flash.bin 档实际包括了 UserInfo.bin 档案的内容，而 UserInfo.bin 档是存放 Command 参数，其他区域则是示图片、动画文件、文字库等数据，如果只是修改 Command 参数，在 Make Bin 时虽然会同时产生 UartTFT\_Flash.bin 档及 UserInfo.bin 档案，但实际上只需要更新 UserInfo.bin 档即可，UserInfo.bin 档案的大小固定为 128Kbytes，因此直接在 UI\_Editor 环境下用 USB 连接线快速更新即可，不需要使用专用的 SPI Flash 烧录器。

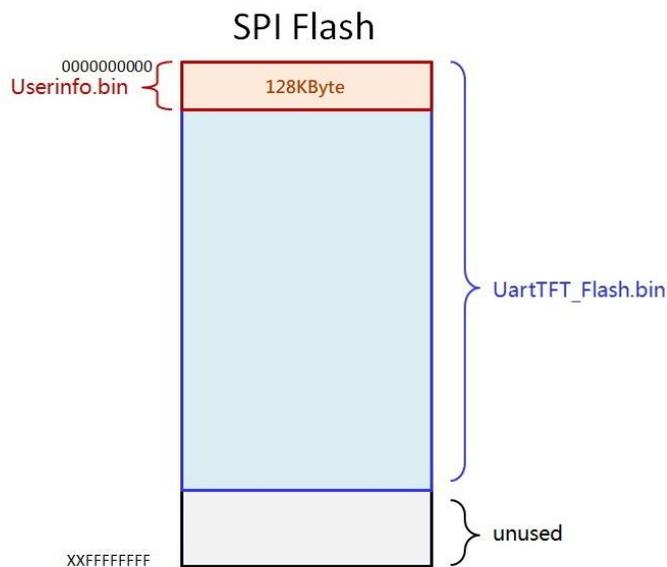


图 4-119: SPI Flash 的结构



图 4-120: SPI Flash 烧录器与接线方式

#### 4.7.2 Userinfo.bin (128K) 更新



图 4-121: UserInfo.bin 更新按钮

如果只是更新部分指令的个别参数，例如某张图片的显示位置、GIF 动画的播放速度、按键控件所执行的指令等不需要重新烧 Flash 的改变。可以在改变了这些参数后重新编译工程，产生 BIN 文件。之后打开 Command 使用 128K 更新去更改 SPI Flash 内的 UserInfo.bin，就可以更改 TFT 串口屏中的参数。

### 4.8 UI\_Editor 范例下载

关于 UI\_Editor 的使用，用户可以至 [乐升半导体](http://www.levetop.cn) 网站下载 LT776 的 UI\_Editor 范例（如 [LT776\\_UI\\_Editor\\_Demo\\_800x480.rar](#)）来操作：

1. 自本公司网站 ([www.levetop.cn](http://www.levetop.cn)) 下载 UI\_Editor 压缩文件 (UI\_Editor\_Vx\_x\_x.rar)，然后解压缩 “UI\_Editor\_Vx\_x\_x.rar”。
2. 下载 UI\_Editor 范例（如 [LT776\\_UI\\_Editor\\_Demo\\_800x480.rar](#)）解压缩及打开压缩包。

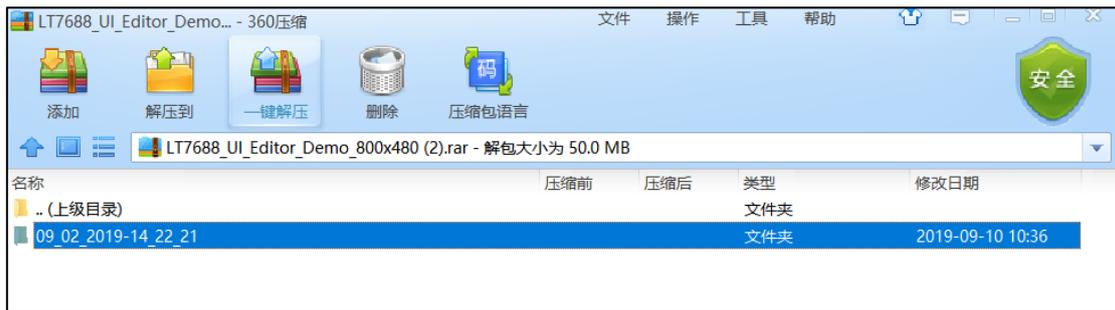


图 4-122: 解压缩及打开压缩包

3. 将演示范例目录如 “09\_02\_2019-14\_22\_21” 文件夹解压缩到 UI\_Editor 的 PROJECT 文件夹内。



图 4-123: 范例目录拷贝到 UI\_Editor 的 PROJECT 文件夹内

**注意:**

- ① 不能修改压缩包文件夹名称及内容。
- ② 压缩包内的文件夹必须解压到 UI\_Editor 的 PROJECT 文件夹内。

4. 打开 UI\_Editor, 打开菜单选项后点击 Load:



图 4-124: 打开 UI\_Editor 点击 Load

5. 打开文件夹内的 COMMANDFILE, 选择工程文件后点击打开。

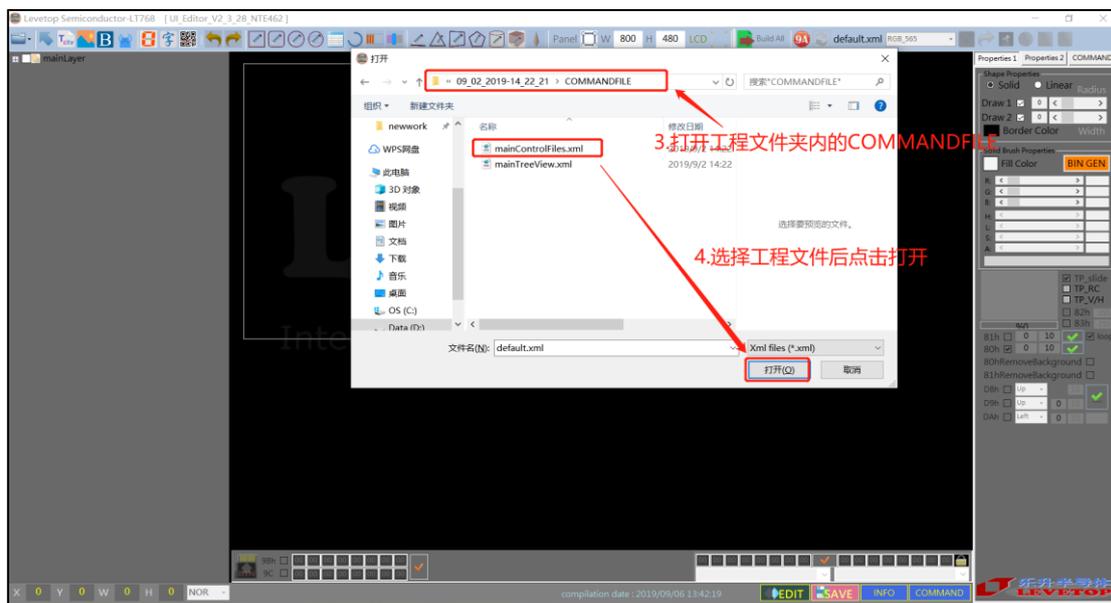


图 4-125: 打开 UI\_Editor 演示范例

## 5. 制作字库与 Wav 档的 Bin 文件

### 5.1 制作字库的 Bin 文件

#### 5.1.1 全字库制作

在应用端如果要使用到中文字库，Levetop 提供一个制作字库 Bin 文件的程序 - “BWFont.exe”，能将字库信息转成 Bin 文件，然后透过 DMA 传输方式将字库数据存到 LT7689 的内建显示内存中，之后如果要在 TFT 屏上显示中文，MCU 只须送 GB 码 (2 个 Bytes) 就可以在设定的位置上显示出中文，因此可以提升中文显示效能，还降低 MCU 处理中文显示的负担。对于字库 Bin 文件的制作，使用者可以参考以下的说明，举例产生一个 16\*16 的宋体字库 Bin 文件：

1、执行 “BWFont.exe” 软件即可打开中文字库 Bin 文件制作界面：

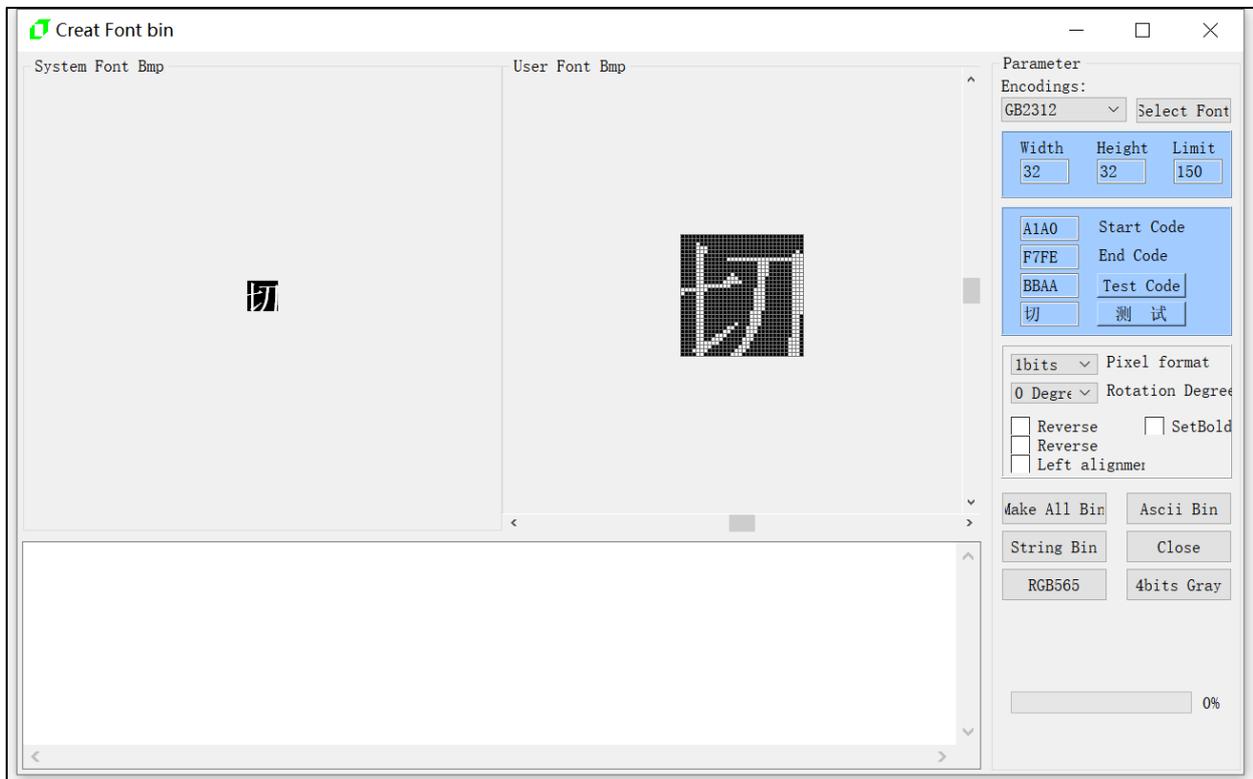


图 5-1: 制作中文字库

2、 点击【Select Font】按钮，可设置字体、字形、大小等，设置完毕后，按确定保存：

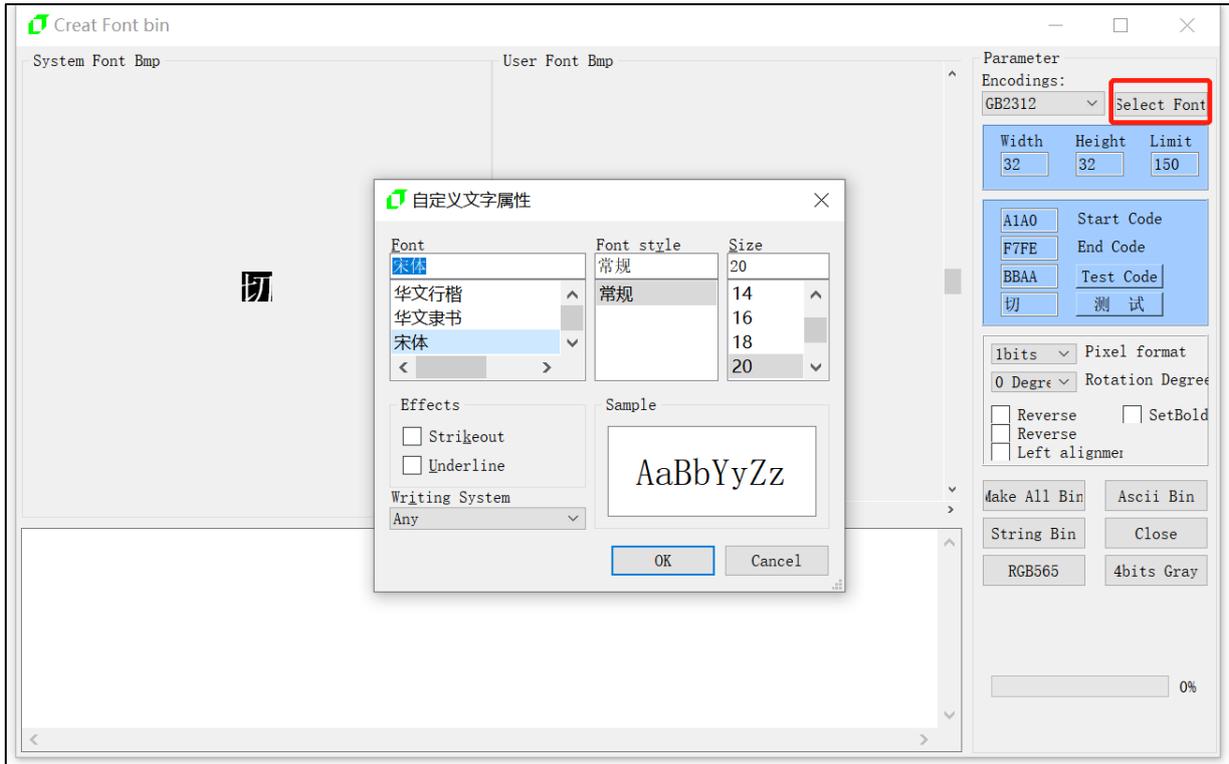


图 5-2：选择字体

3、 设置字库可设置任意字体大小，点击【Test Code】按钮即可查看该字符的数据。

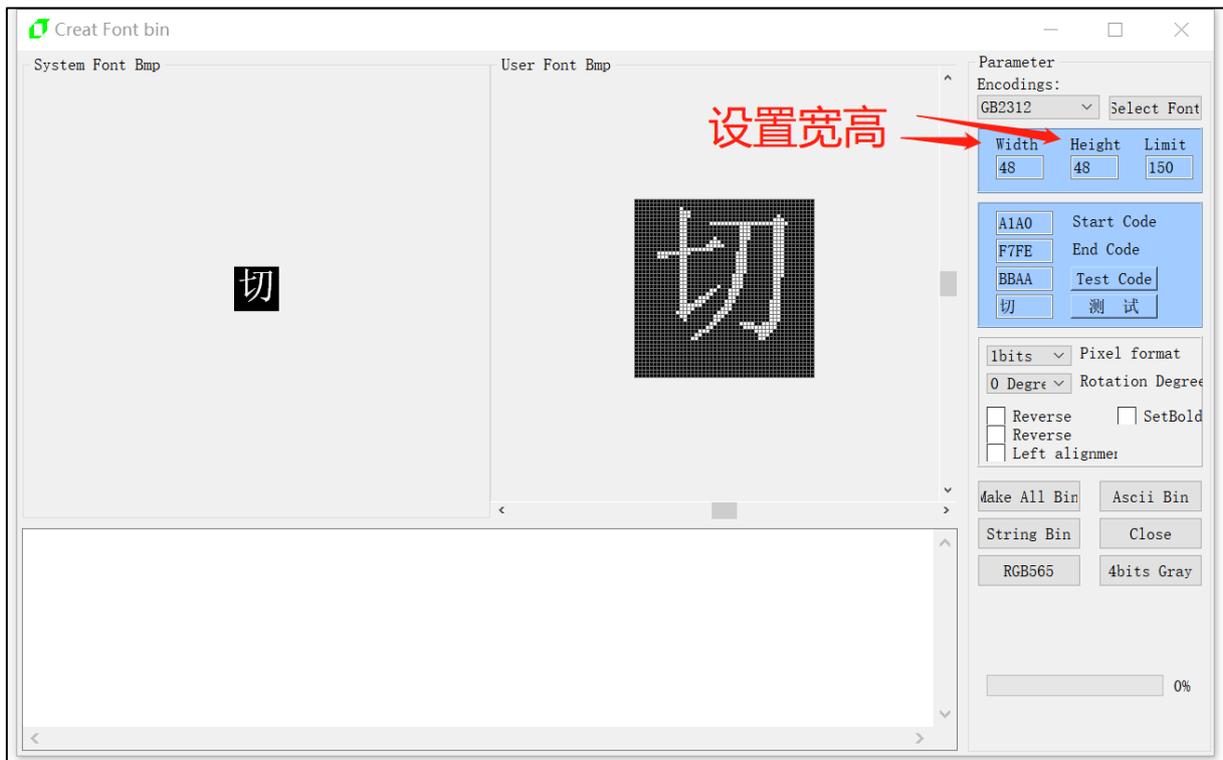


图 5-3：设置字库

4、点击【Make All Bin】即可输出字库 Bin 文件。**注意：**输入文件名时文件名中不能包含下面这些字符，如：? \* / \ < > : " |，否则无法保存。

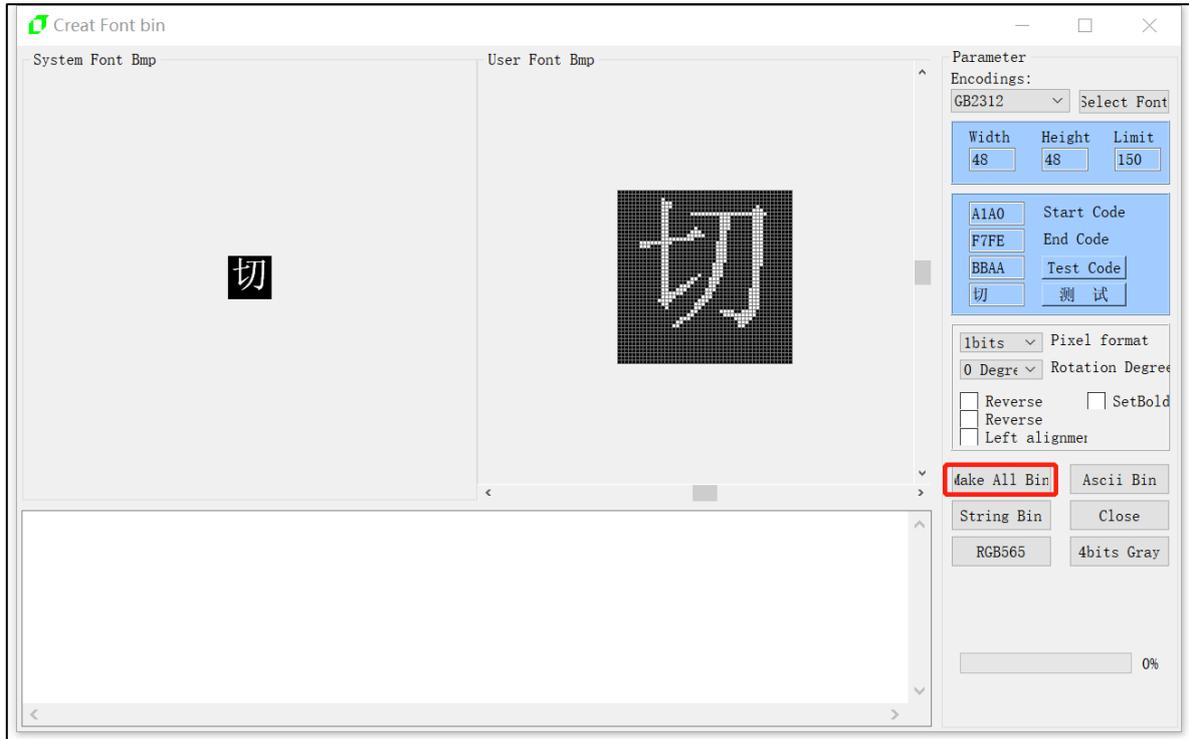


图 5-4：保存字库

当显示 Font Lib ok 时，即全字库制作成功：

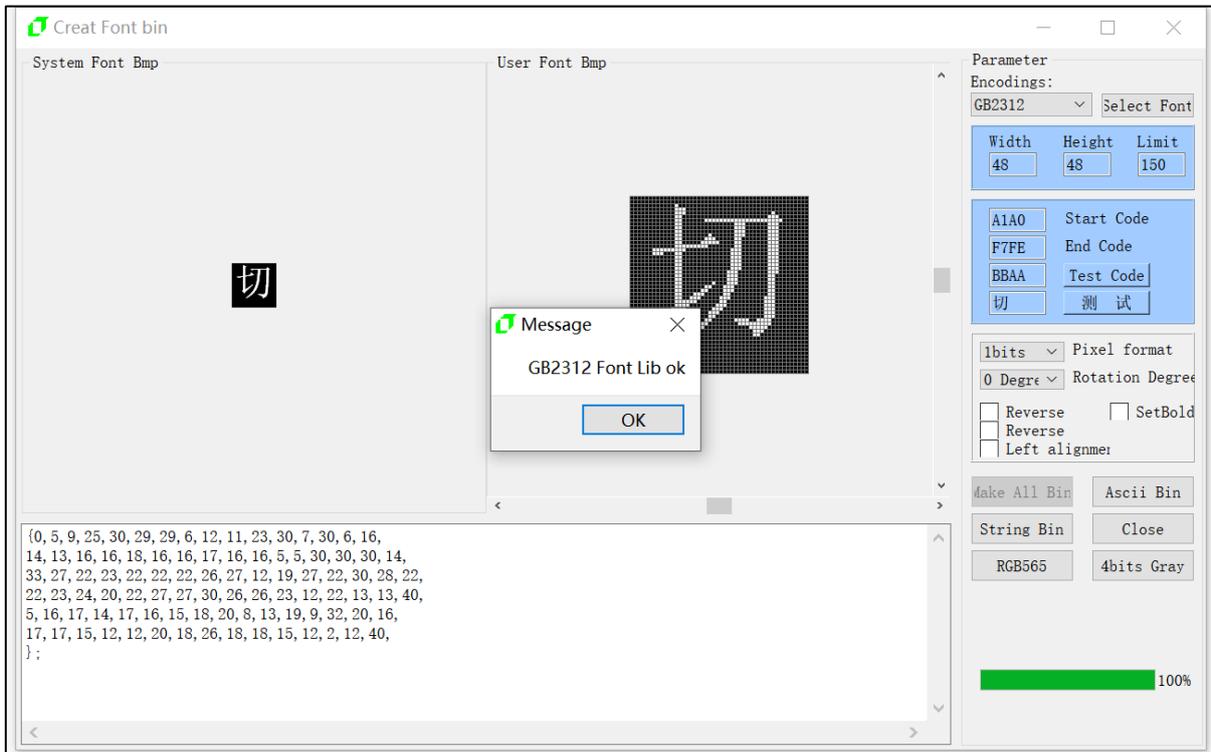


图 5-5：字库制作完成

5、制作完成后可以在目标文件夹中看到导出的 简体 16\_16 宋体.bin 文件：



图 5-6: 导出的字库 Bin 文件

### 5.1.2 自定义字库制作

由于大字库容量相对比较大，考虑到会占用 Flash 比较多的空间，因此该主控端提供了自定义字库的功能，客户可根据需要提前想好要调用的文字，从而仅仅将这部分文字打包成一个字库，进而减少 Flash 容量的浪费。对于自定义字库 Bin 文件的制作，使用者可以参考以下的说明，举例产生一个 48\*48 的宋体自定义字库 Bin 文件：

1. 前期选择字体大小和形状，步骤与 4.2.5.1 全字库制作相同，点击【String Bin】后选择一个存放要调用文字的 txt 文档：

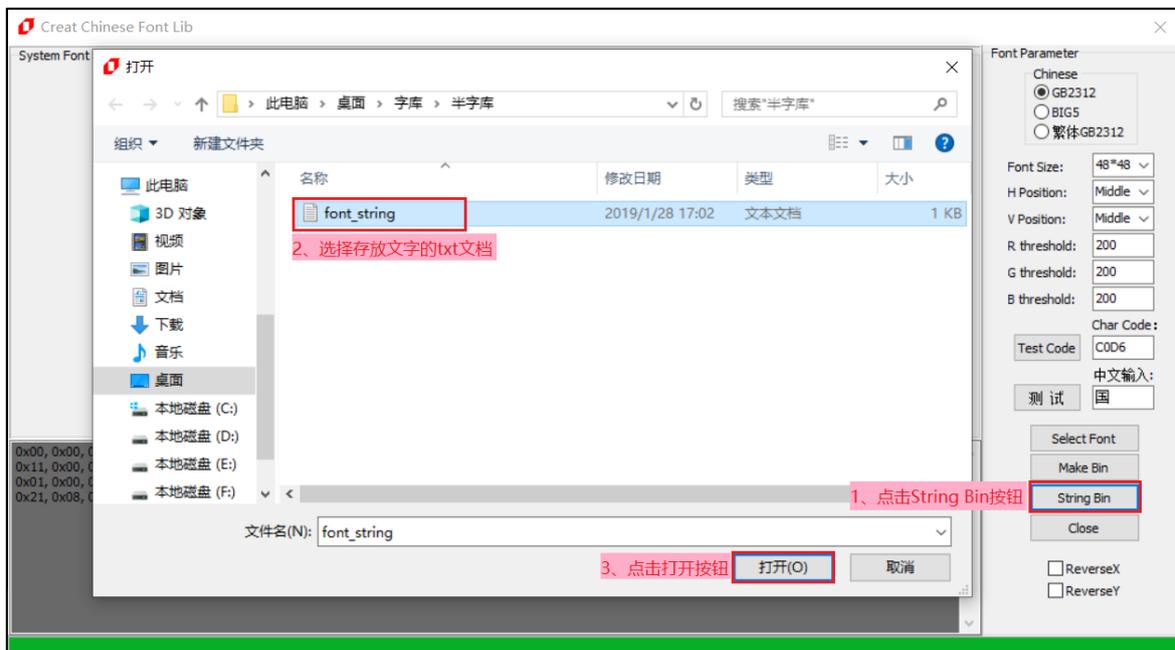


图 5-7: 选择 txt 文档

2. 提前将需要调用显示的文字存放在一个记事本下:

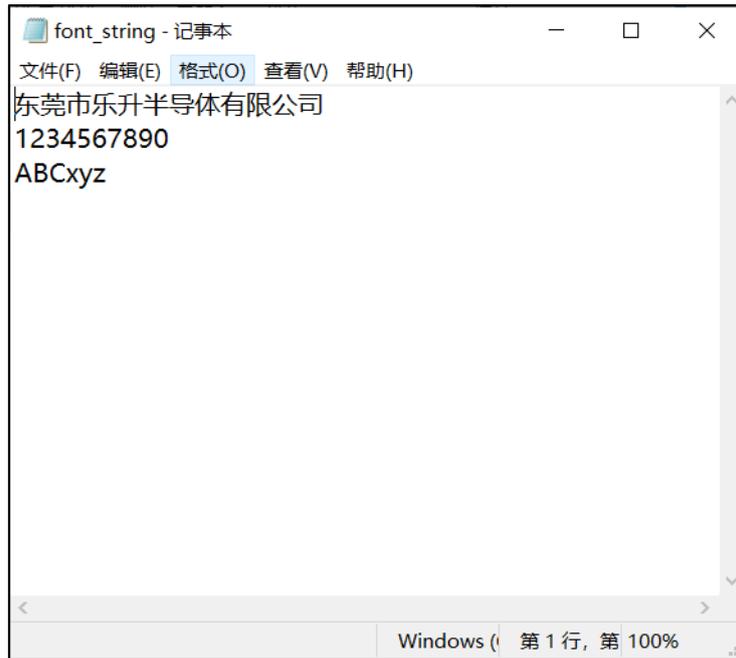


图 5-8: 存放文字的 txt 文档

3. 指定一个路径存放将要生成的自定义字库，命名好后点击保存:

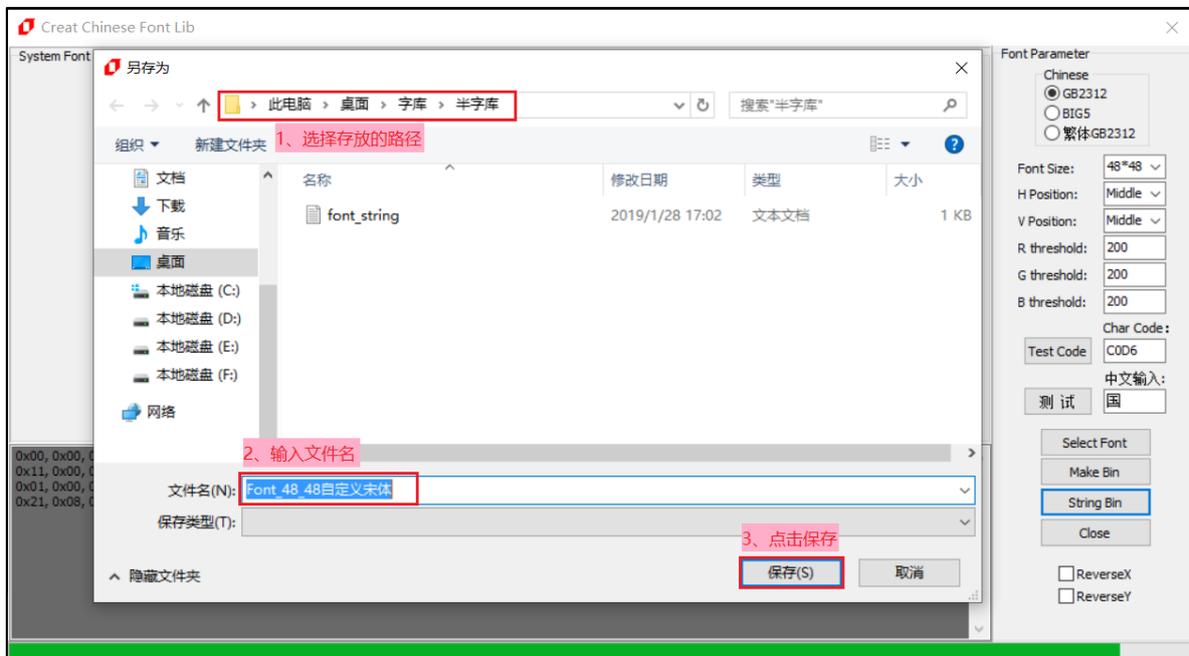


图 5-9: 保存字库

当显示 String Font Lib ok 时，即自定义字库制作成功：

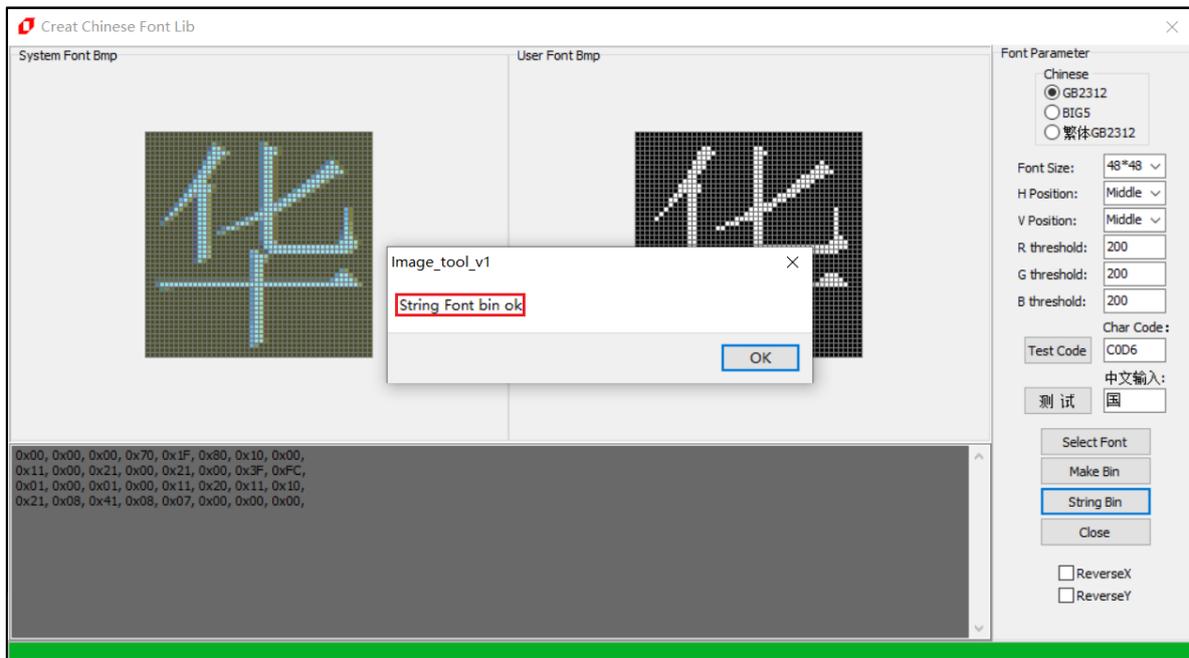


图 5-10：字库制作完成

4. 完成后可以在目标文件夹中看到导出的 `Font_48_48 自定义宋体.bin` 文件：

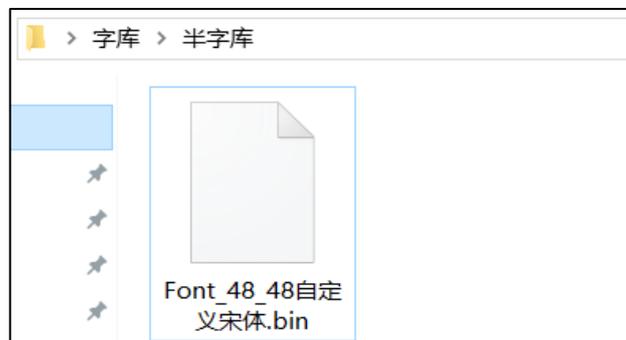


图 5-11：导出的字库 Bin 文件

## 5.2 制作 Png ASCII 码 Bin 文件

在应用端会常用到 ASCII 码显示，通过字库去显示 ASCII 效果对于部分应用可能不理想，使用这个软件可以将字库内的 ASCII 码转换为 png 图片的 bin，然后通过指令去显示出来，下面介绍 png ASCII 码文件的生成流程：

- 1、执行“Uart\_Bmp\_GB\_PNG\_Font\_Vxx.exe”软件即可打开 png ASCII 码 bin 文件制作界面：



图 5-12：制作 Png ASCII 码文件

- 2、点击【Select Font】按钮，可设置字体、字形、大小等，设置完毕后，按确定保存：

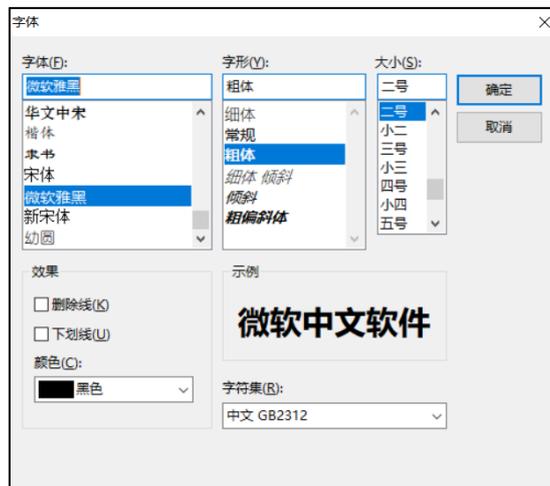


图 5-13：选择字体

3、设置字体的宽度和高度大小，微调字体的位置，调整到适合的位置，预览文字，点击【Test Char】按钮即可查看该字符的数据,观察是否合适，合适后按键【ASCII Bin】即可生成 Png ASCII bin 文件。

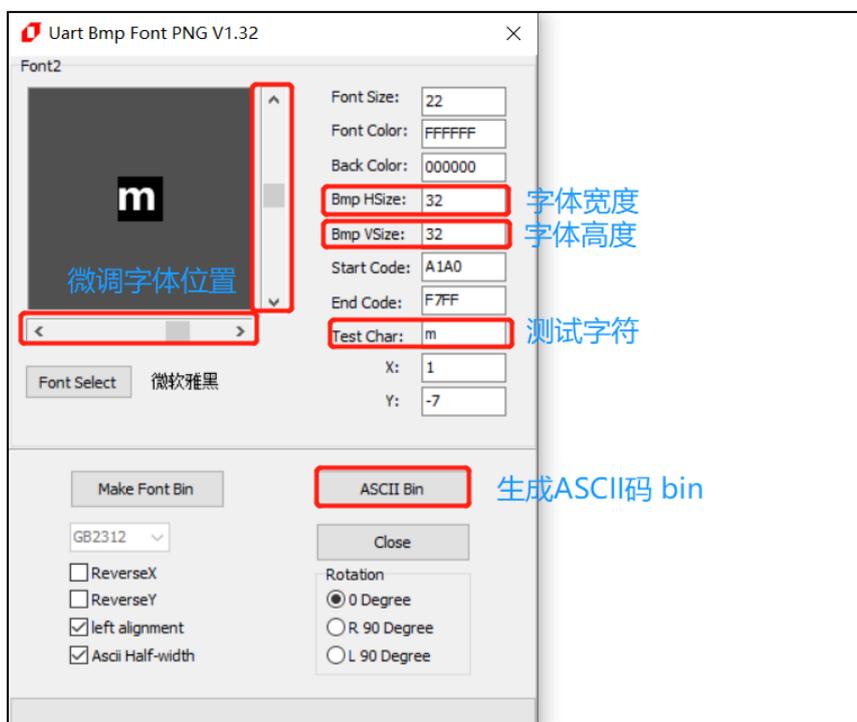


图 5-14: 调整位置并生成 Bin 文件

## 5.3 制作 Wav 檔的 Bin 文件

### 5.3.1 音频文件转 WAV

1. 若音频素材格式不是 WAV 格式，则需要通过格式转换来获取 WAV 格式文件。下面以“格式工厂”免费版为转换平台进行操作。首先打开软件，选择音频，选择“-> WAV”，进入添加文件界面。



图 5-15: 选择功能



图 5-16: wav 功能界面

2. 点击添加文件按钮，选择需要转换的音频文件。

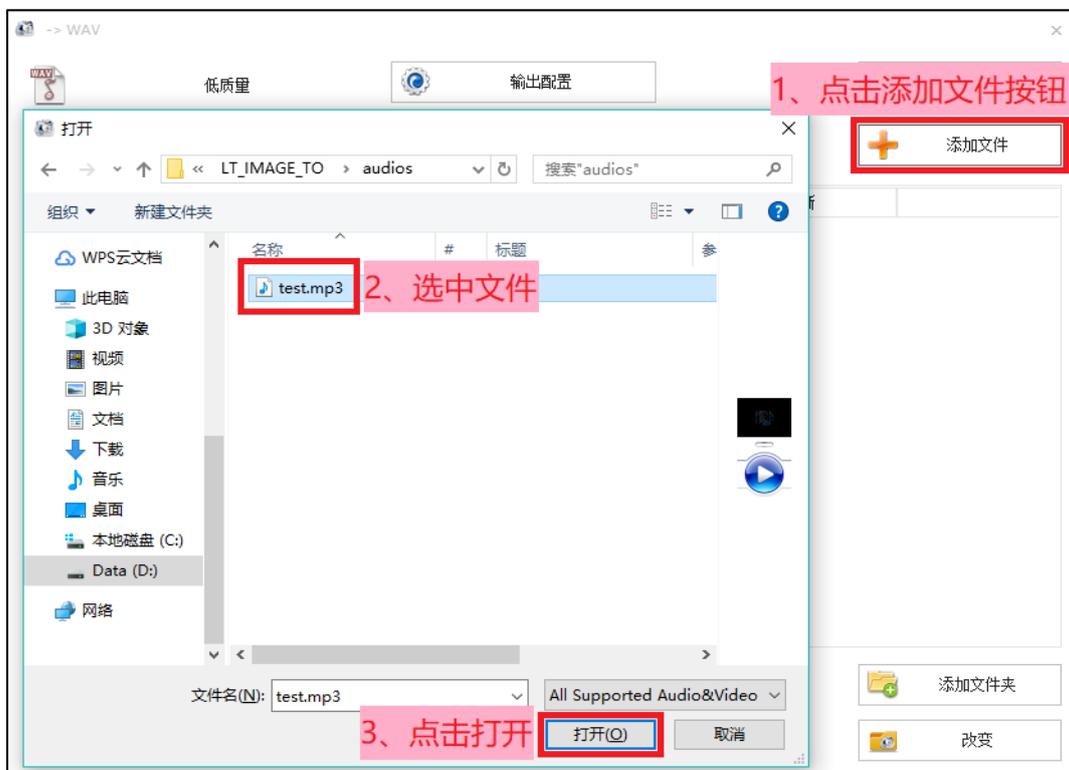


图 5-17: 添加 wav 文件

3. 点击输出配置按钮，进行音频设置，其中，采样率可选 11025 或 22050。由于采用低成本的功能方案，同时为了节省 bin 文件存放空间，推荐选择采样率为 11025。



图 5-18: 输出配置

4. 选中文件，点击剪辑按钮，进入音频剪辑界面，可选择需要的音频段。如不需要，请直接转到步骤 5。



图 5-19: 进入剪辑功能

剪辑音频，可调节音量大小，以及截取其中的片段，剪辑完成后，点击确定按钮保存：

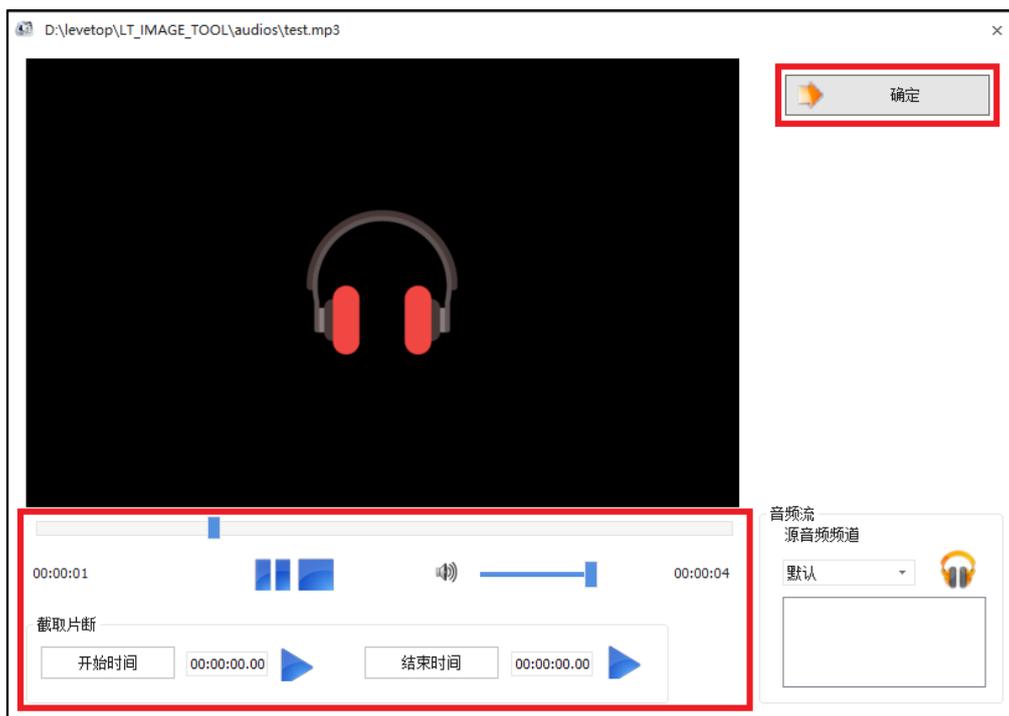


图 5-20：剪辑界面

5. 点击右下角的改变按钮，可重新选择输出目标文件夹，点击确定按钮添加任务。



图 5-21：选择输出文件夹

6. 点击开始按钮，开始转换，转换完毕后，可在目标文件夹查看导出的 WAV 文件。

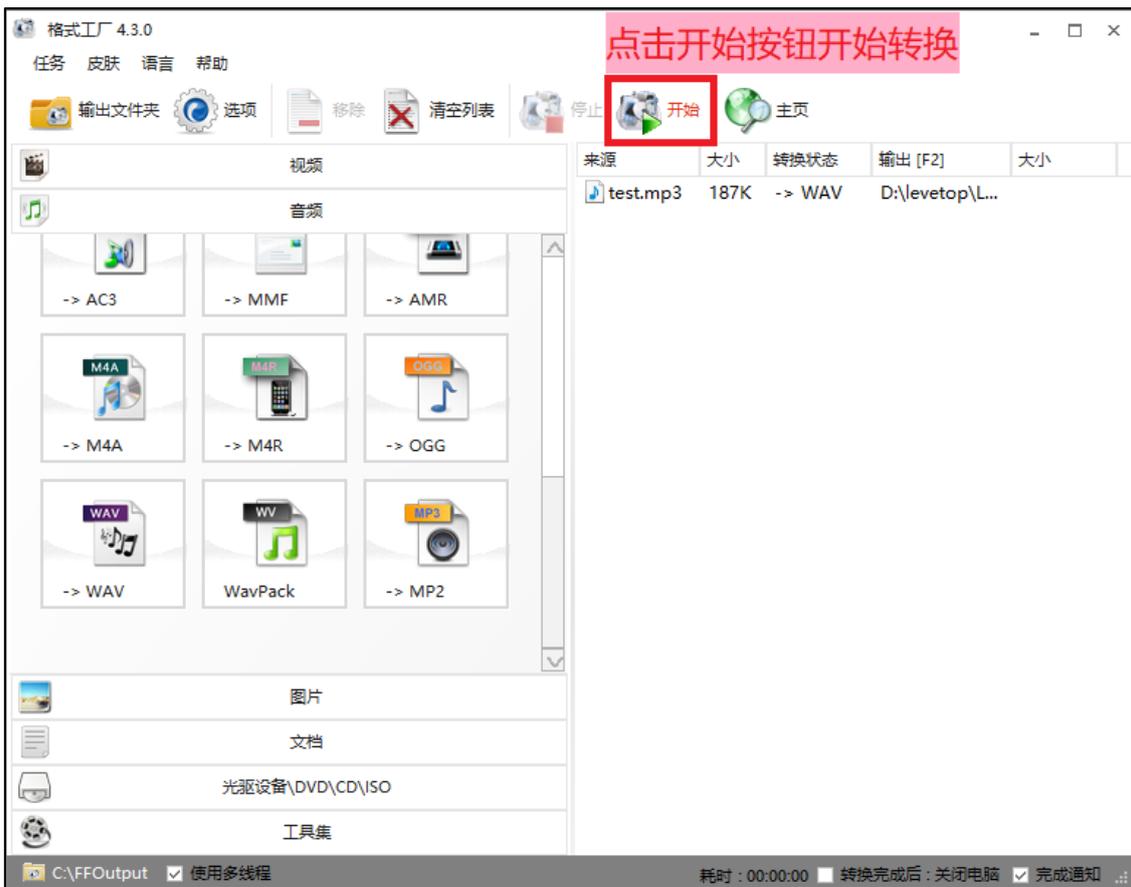


图 5-22: 开始转换



图 5-23: 导出的 wav 文件

### 5.3.2 制作 WAV 档 Bin 文件

1. 执行 “WavTool.exe” 软件即可打开 WAV Bin 文件制作界面：

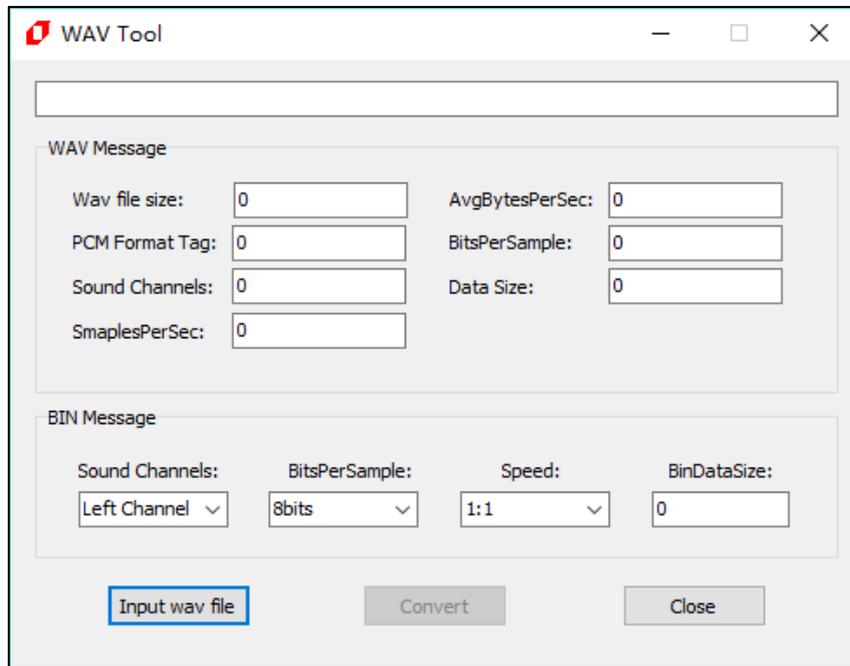


图 5-24: WAV Bin 文件制作界面

2. 导入 WAV 文件, 点击 Input wav file 按钮, 选择需要转换的 WAV 文件, 点击打开, 即可添加:

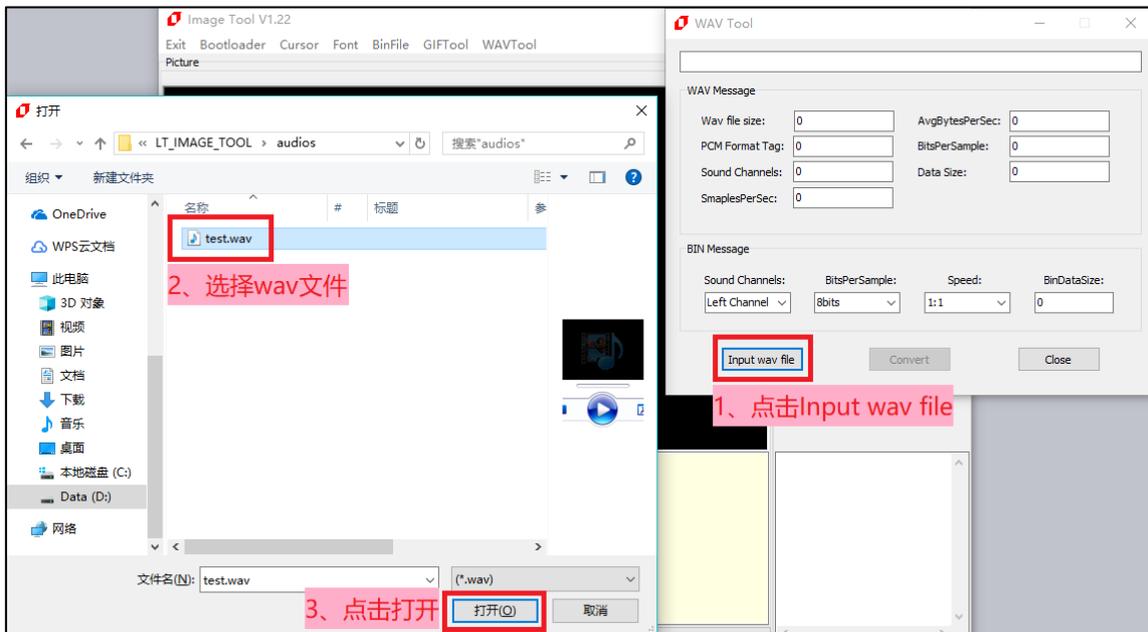


图 5-25: 添加 wav 文件

添加成功后，在界面上方，可以看到文件所在路径，并显示 wav 文件的相关数据（如下图）。如果采样率 (SamplesPerSec) 不是 11025 或 22050，建议改变 Speed 的比率，或是先用其他音频软件将采样率变成 11025 或 22050。

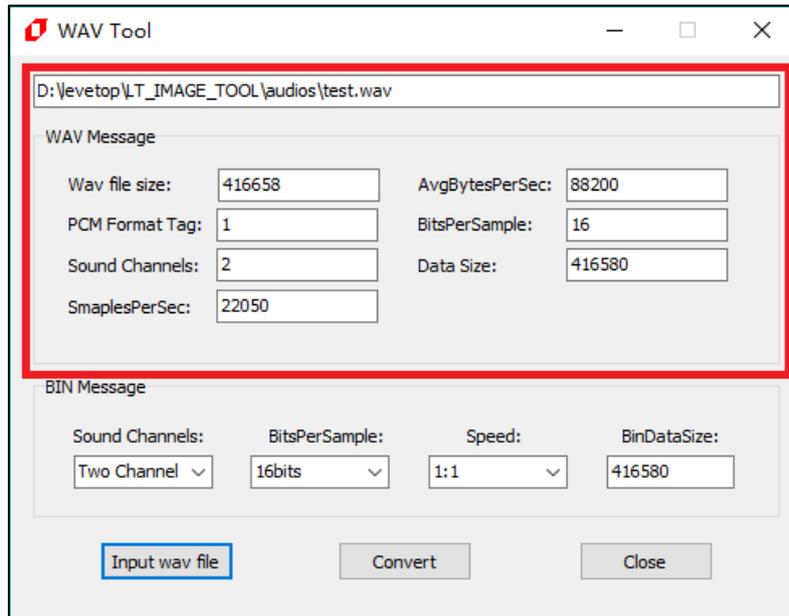


图 5-26: 添加成功

3. 设置 bin 文件的参数, Sound Channels 选项可选左声道、右声道或双立体声道, BitsPerSample 选项可选 8bits 或 16bits。一般选用单声道、8bits 就足够, bin 文件较小, 便于储存。Speed 选项选择生成 bin 文件时的采样速度, 速度越高, 音质也会相对下降, 同时需要改变程序的定时器更新数据的时间。

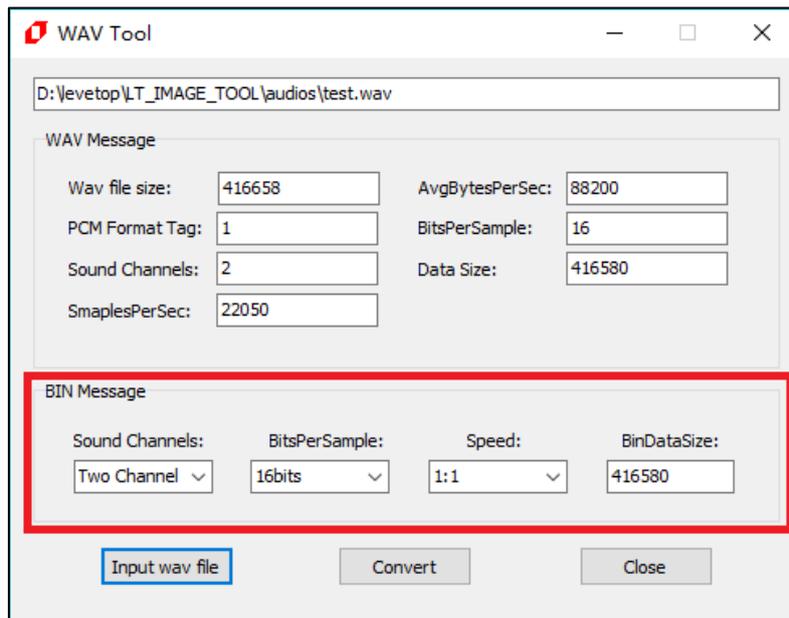


图 5-27: WAV Bin 文件制作界面

4. 导出 bin 文件，**注意**：输入文件名时文件名中不能包含下面这些字符，如：? \* / \ < > : " |，否则无法保存。

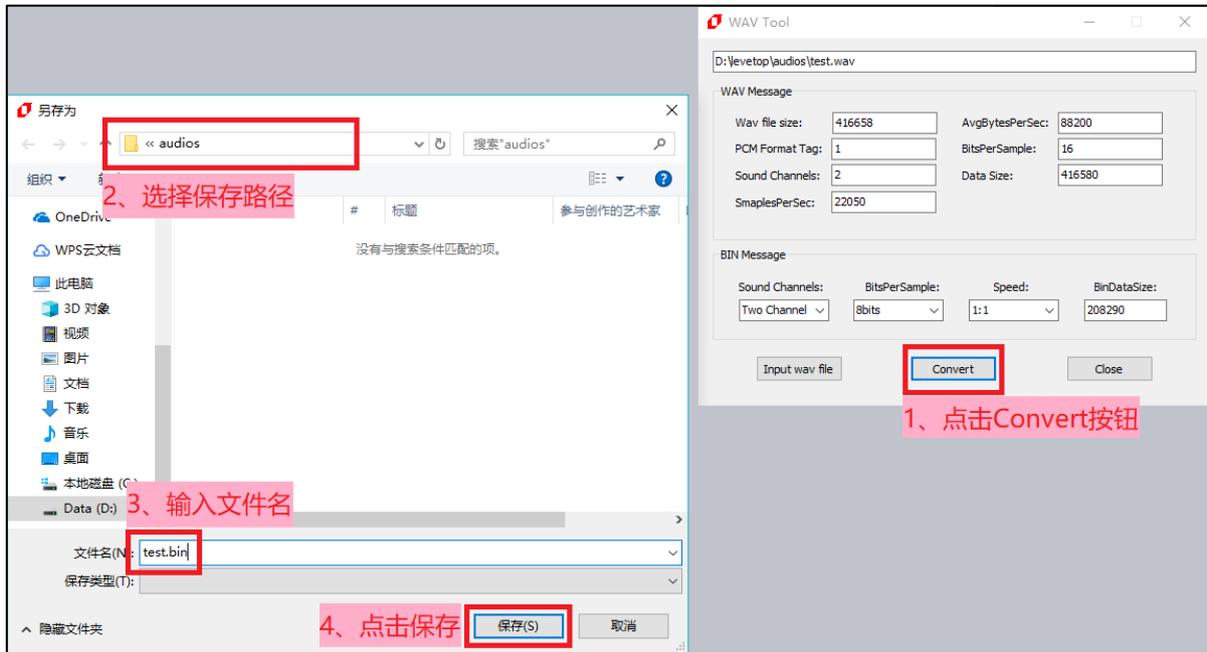


图 5-28：导出 bin 文件

若该音频文件与 bin 设置参数不符合，则会提示 Can't suport this bits，请按照步骤 3 重新设置：

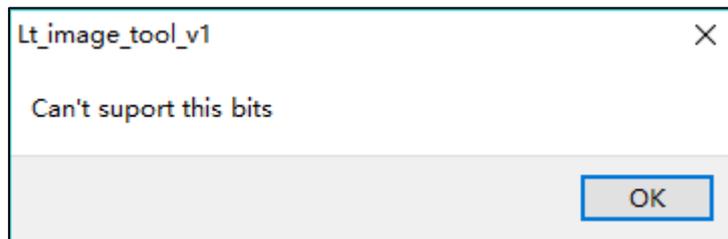


图 5-29：错误提示

若符合要求，则点击 OK 按钮开始导出：

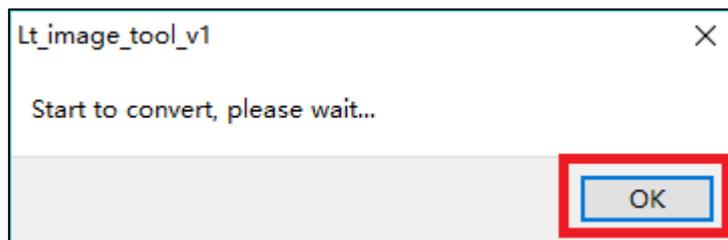


图 5-30：点击 OK 导出

5. 成功导出 WAV bin 文件，并可在目标文件夹看到导出的 test.bin 文件：

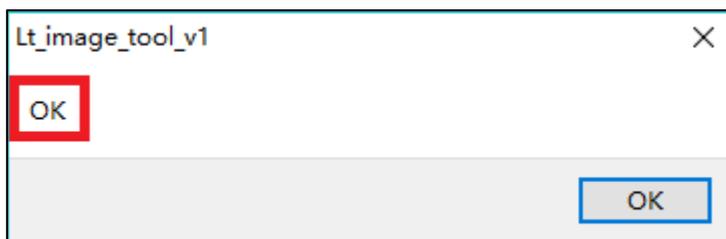


图 5-31：导出成功

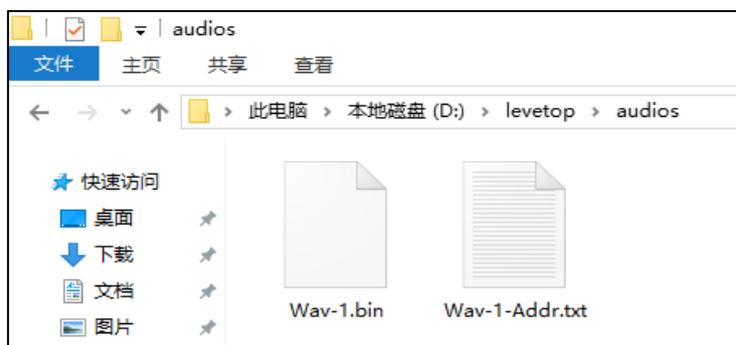


图 5-32：生成的 bin 文件

### 5.3.3 典型音频驱动电路

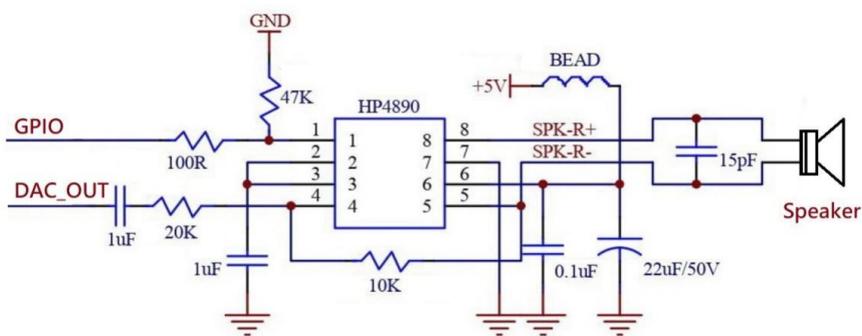


图 5-33：DAC\_OUT 输出及音频驱动电路

## 6. 版权说明

本文件之版权属于 深圳市乐升半导体 所有，若需要复制或复印请事先得到 乐升半导体 的许可。本文件记载之信息虽然都有经过校对，但是 乐升半导体 对文件使用说明的规格不承担任何责任，文件内提到的应用程序仅用于参考，乐升半导体 不保证此类应用程序不需要进一步修改。乐升半导体 保留在不事先通知的情况下更改其产品规格或文件的权利。有关最新产品信息，请访问我们的网站 [Https://www.levetop.cn](https://www.levetop.cn) 。