技术笔记

Technical Note

教程 2 - 串口屏文本控件应用 v1.0

工程技术笔记

类别	内容
关键词	文本显示和输入、艺术字实现、文本增量
摘要	文本控件的使用方法

广州大彩光电科技有限公司

修订历史

版本	日期	原因	编制	审查
V1.0	2019/07/17	创建文档	陈鹏	刘启鑫

销售与服务

广州大彩光电科技有限公司

电话：020-82186683

传真：020-82187676

Email：hmi@gz-dc.com（公共服务）

网站：www.gz-dc.com

地址：广州高新技术产业开发区玉树工业园富康西街 8 号 C 栋 303 房

官网零售淘宝店：www.gz-dc.taobao.com

目录

1. 适合范围.....1
2. 开发环境版本.....2
3. 参考文档.....3
4. 概述.....4
5. 文本控件属性.....5
5.1 属性介绍.....5
6. 教程实现.....13
6.1 准备UI素材.....13
6.2 开发工具.....13
6.3 软件版本.....14
6.4 功能实现.....14
6.4.1 建立工程.....14
6.4.2 文本控件更新.....15
6.4.3 文本控件输入.....19
6.4.4 艺术字显示.....21
6.4.5 文本增量调节.....24
6.5 下载工程.....29
6.6 联机调试.....30
6.6.1 硬件连线.....30
6.6.2 联机设备.....30
6.6.3 指令控制.....30
6.6.4 串口屏效果.....31
7. 部分程序代码.....32
8. 免责声明.....34

1. 适合范围

文档适合全系列所有大彩协议串口屏产品。

2. 开发环境版本

1. VisualTFT 软件版本：V3.0.0.1037 及以上的版本，版本查看方式：
   (1) 打开 VisualTFT 软件启动页面如图 2-1 所示，右上角显示的软件版本号；



图 2-1 软件版本

   (2) 打开 VisualTFT，在软件右下角可以查看软件版本图 2-2 所示，最新版本可登录 http://www.gz-dc.com 下载。

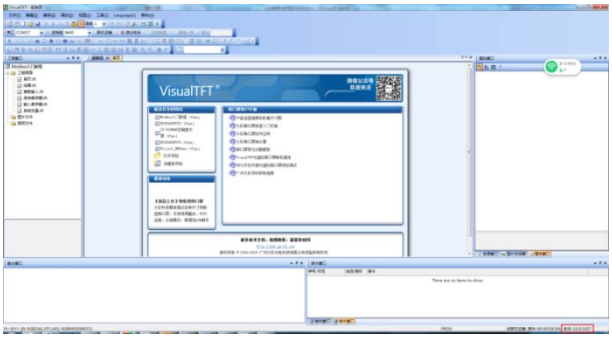


图 2-2 软件版本

2. 硬件固件版本查看方式：
   (1) 查看屏幕背面版本号贴纸，如图 2-3 所示。



图 2-3 背面贴纸

   (2) 画面中创建一个文本控件，属性【输入方式】选择“系统变量”，【系统变量】选择“固件版本号”，编译工程下载到屏中查看。

3. 参考文档

本例程中涉及所需的文档如下：

1. 《大彩串口屏指令集 V5.0 .pdf》 ：用于查看指令解析；

4. 概述

文本数据可以通过文本控件来显示。文本可以由用户主机输入，也可以通过设置弹出系统键盘或者自定义键盘，对文本进行输入编辑操作，然后显示在串口屏上。

本例程中介绍文本控件常见的应用，结合工程中的画面介绍每一个应用的配置，常见的应用如下所示。

1. 文本控件更新：实现字符串、小数、正/负数的显示。
2. 文本控件输入：点击文本框，弹出小键盘、全键盘输入。
3. 艺术字显示：使用艺术字显示字体效果。
4. 文本增量调节：通过按钮控件控制文本小数、整数增量调节。

5. 文本控件属性

5.1 属性介绍

本章节重点介绍文本控件的属性，属性窗口如图 5-1 所示。



图 5-1 文本控件属性总览

1. 文本：用户所需显示的内容。
2. 字体：选择文本显示的字体大小和样式。

串口屏支持的字库编码：ASCII、GB2312、GBK、UNICODE。

- ASCII：英文字母、数字、符号等、无汉字。
- GB2312：简体汉字、无繁体、无 ASCII 字符。
- GBK：简体汉字与繁体汉字、ASIIC、日韩文等。
- UNICODE：支持多种国际语言、如法语、俄语、波斯语等。

(1) F 型平台

该平台所生成的字库均是矢量字库，字体的实际尺寸可以任意缩放而不失真。若用户选择如图 5-2 的 VisualTFT 自带的“ASC_CHS_24”字体样式，则 ASC 表示 ASCII 字库，CHS 表示 GBK2312 字库，24 表示显示字体高度占 24 个像素。

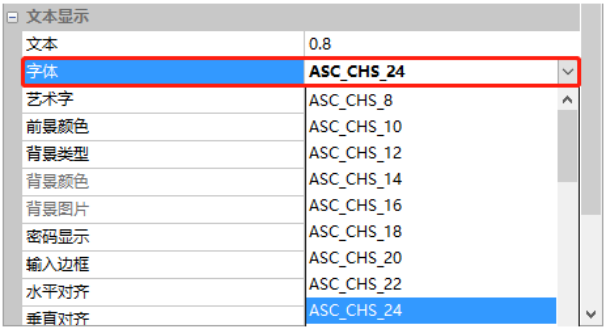


图 5-2 F 型字体样式

(2) 物联型平台

和 F 型平台一致，均是矢量字库，VisualTFT 自带的字体样式如图 5-3 所示。

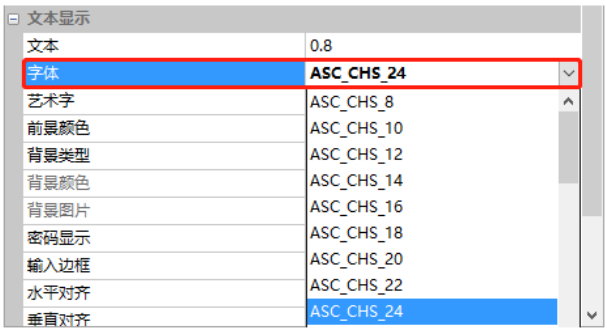


图 5-3 物联型字体样式

系统有提供常见的字体样式，用户还可以自定义字库。在 VisualTFT 软件的菜单栏，在 F 型或物联型平台上，选择【工具】→“字库配置”，如图 5-4 所示，详细参考《工程笔记-如何自定义矢量字库》文档。

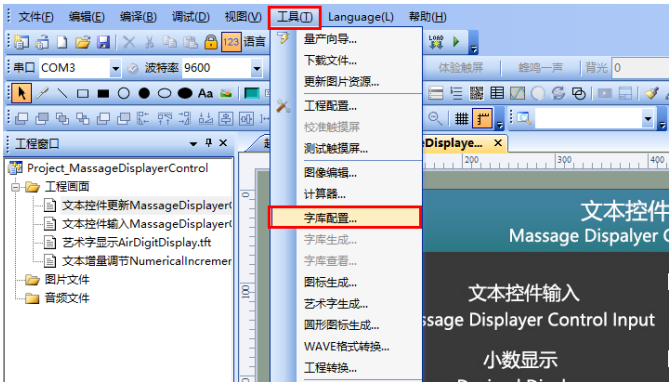


图 5-4 字库配置

(3) 基本型平台

VisualTFT 自带的字体样式，如图 5-5 所示。ASCII 6*12 表示仅显示英文字母、数字、符号，且字体大小为 6*12 个像素，ASCII 样式属于半宽格式；GBK231232*32，表示 仅显示简体汉字，且字体大小为 32*32 个像素，GBK2312 样式属于半宽格式；GBK12*12 表示显示简体汉字与繁体汉字、ASIIC、日韩文，字体大小为 12*12 个像素，GBK 属于半宽格式。

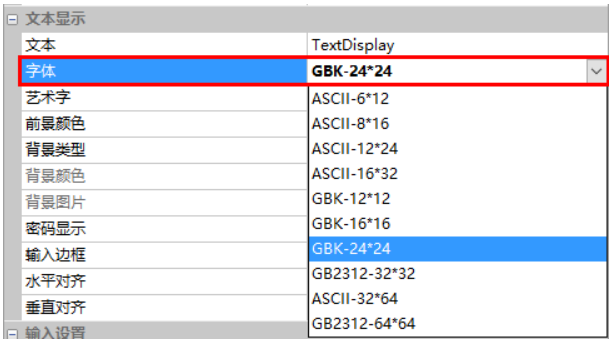


图 5-5 基本型字体样式

在基本型平台中，除了系统提供的字体样式，用户还可以自定义字库。在 VisualTFT 软件的菜单栏，选择【工具】→“字库生成”，根据需求配置参数即可，如图 5-6 所示。

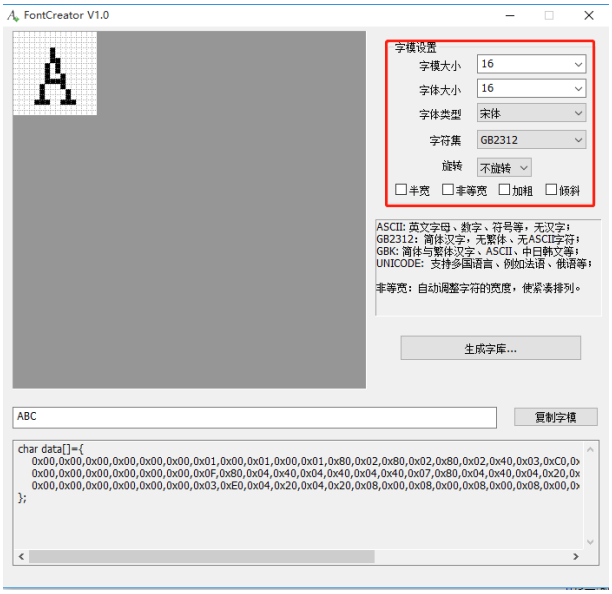


图 5-6　基本型字库生成

3. 艺术字：用于实现自定义或特殊字体的显示。
   (1) 图标文件：插入艺术字的 icon 文件。
   (2) 小数点宽度：小数点宽度比例。如图 5-7 所示。



图 5-7　艺术字

4. 前景颜色：设置字体的颜色，如图 5-8 所示。

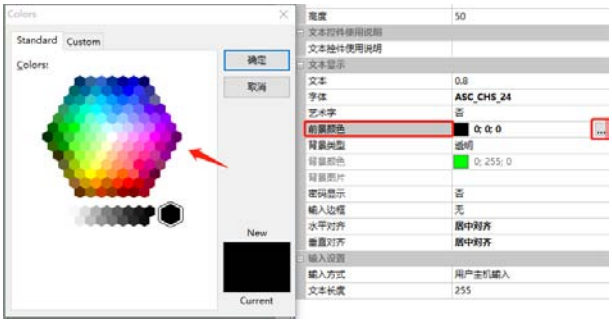


图 5-8　前景颜色

串口屏运行时，还可以通过指令修改，详细指令可查阅指令助手或《大彩串口屏指令集 V5.0 .pdf》文档。

5. 背景类型：文本控件背景类型可以选择透明、单色、图片、单色（选中时显示）。如图 5-9 所示。

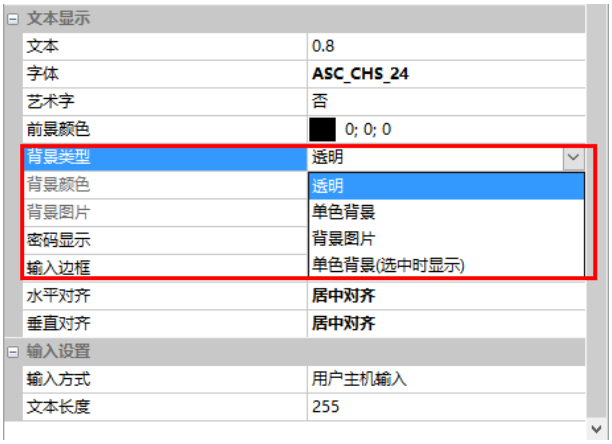


图 5-9 背景类型

(1) 透明：将文本控件背景透明化。

(2) 单色：只有一种颜色构成的背景，也称为“纯色背景”。

(3) 图片：给文本控件的背景插入图片，如图 5-10 所示。

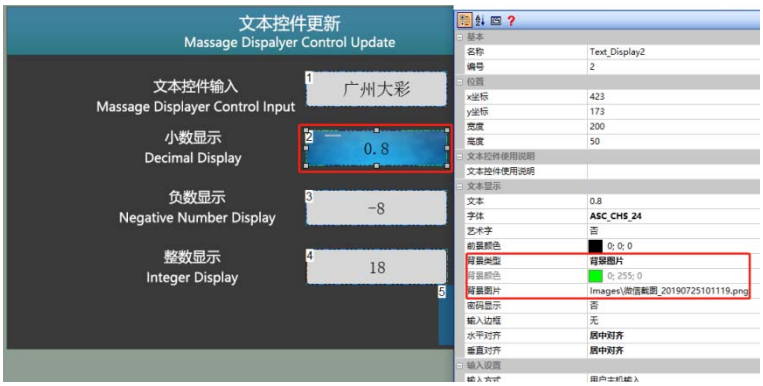


图 5-10 文本控件背景图片

(4) 单色（选中时显示）

点击文控件，弹出键盘，该控件背景色如透明显示为单色背景。该功能实现，必须设置【输入方式】为“弹出系统键盘输入（用户主机可输入）”。如设置【背景颜色】为绿色，如图 5-11 所示。



图 5-11　效果显示

6. 密码显示：开启后输入文本将以“*”显示，例如文本为“0.8”，将显示“***”，如图 5-12 所示。

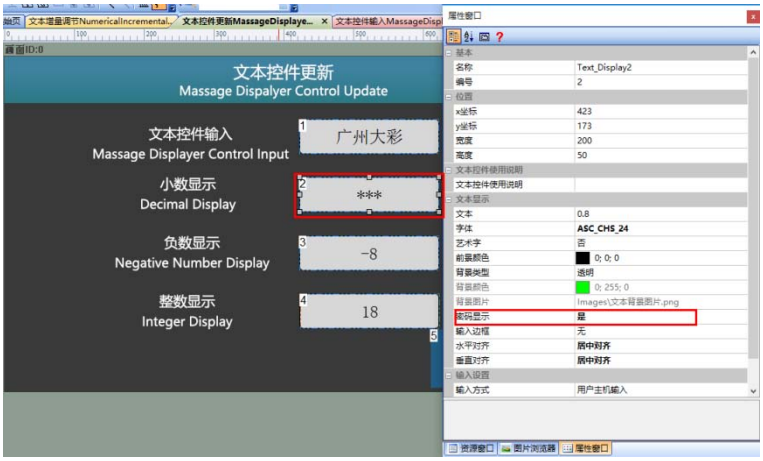


图 5-12　效果显示

7. 输入边框：开启后，文本控件输入时，显示边框。
8. 水平对齐/垂直对齐：设置文字的对齐方式；如图 5-13 所示。

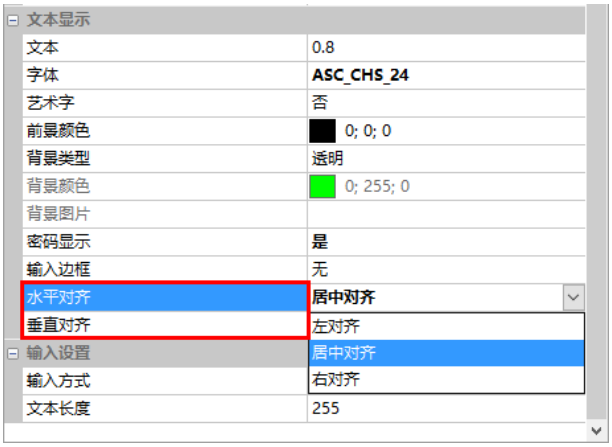


图 5-13　文本对齐

9. 输入方式：分为“用户主机输入”、“弹出系统键盘输入”、“自定义按键输入”、“系

统变量”。如图 5-14 所示。

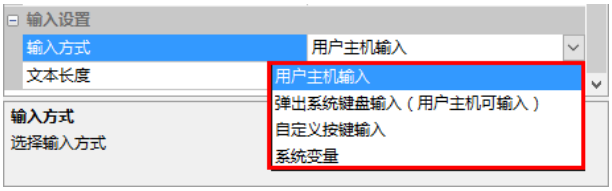


图 5-14　输入方式

(1) 用户主机输入：文本由用户的主机输入，通过串口把指令传到串口屏，把数据显示到屏幕。如图 5-15 所示。



图 5-15　用户主机输入

(2) 弹出系统键盘输入（用户主机可输入）：弹出系统键盘进行文本输入，键盘类型可选择小键盘和全键盘。如图 5-16 所示。



图 5-16　弹出系统键盘输入

a) 键盘类型：分为小键盘和全键盘。

- 小键盘可以输入数字，如图 5-17 所示。



图 5-17　小键盘

- 全键盘可以输入数字、英文大/小写的字母、中文，如图 5-18 所示。



图 5-18　全键盘

b) 提示信息：当用户点击文本框，弹出键盘的输入框里的提示信息，如填入：请输入密码，如图 5-20 所示。



图 5-19 提示信息

如需要实现键盘中的英文提示信息，请参考《【扩展功能】串口屏_文本限值提示如何修改为英文_V1.0.pdf》文档

c) 键盘位置：触摸文本框弹出，键盘的位置，分为自动和指定，指定模式下如图 5-20 所示。



图 5-20 指定位置

d) 初始值：弹出键盘时，在键盘的输入显示栏是否显示当前文本值。

e) 数值限定：弹出键盘时，限制键盘输入的阈值，如图 5-21 所示。



图 5-21 数值限定

(3) 自定义按键输入：文本由用户结合按钮控件的自定义输入，如图 5-22 所示。可实现自定义键盘的功能，可参考《【串口屏】如何自定义键盘 V_1.0》文档。



图 5-22 自定义按键输入

(4) 系统变量：可以用于显示串口屏的固件版本号、通信波特率等等，如图 5-23 所示。

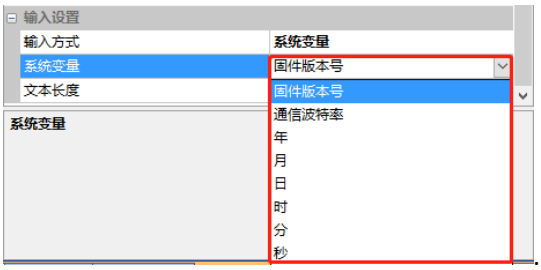


图 5-23　系统输入

10. 文本长度：文本输入的长度，最大为 255 个字节。如图 5-24 所示。



图 5-24　文本长度

6. 教程实现

该例程使用大彩 F 型 7 寸电阻串口屏 DC80480F070 为验证开发平台。如图 6-1 所示。

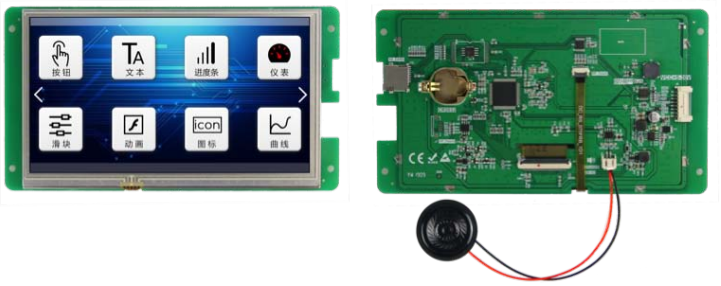


图 6-1　F 型 7 寸电阻屏

6.1　准备 UI 素材

准备好相应的美工 UI。如图 6-2 所示。

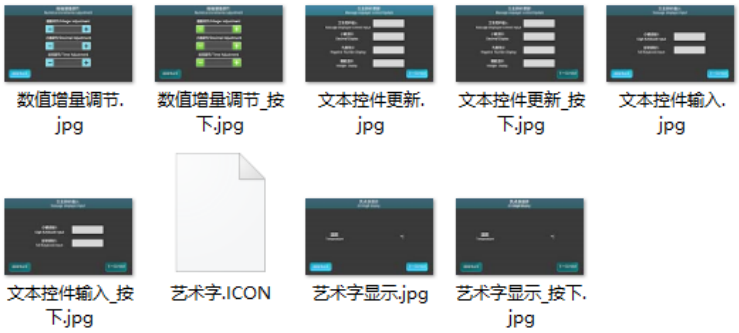


图 6-2　UI 准备

6.2　开发工具

准备以下开发工具，硬件开发工具如表 6-1 所示，软件开发工具如表 6-2 所示。

表 6-1　硬件开发工具

硬件开发工具	功能
电源适配器 5V/9V/12V　1A	用于给串口屏供电
调试转接板	用于与 PC 调试（仅开发时用到）
8P-PH2.0 线缆	用于串口屏和用户主机连接
SD 卡	用于脱机量产下载工程
RS-232 串口线	用于 PC 连接串口屏调试

表 6-2　软件开发工具

软件开发工具	功能
Configure Virtual Serial Port Driver	用于创建虚拟一对串口
VisualTFT	用于工程配置

调试连接示意图，如图 6-3 所示。



图 6-3　调试连接图

6.3　软件版本

本例程所实现的功能 VisualTFT 版本和固件版本如下表 6-3 所示。

表 6-3　版本清单

版本	版本号
VisualTFT 版本	V3.0.0.1037
基本型（经济型、商业型）固件	V2.22.912.259 及以上的版本支持
F 型固件	V4.1.66.0 及以上的版本支持
物联型固件	V3.0.301.0 及以上的版本支持

6.4　功能实现

6.4.1 建立工程

根据串口屏的实际型号，在 VisualTFT 软件菜单栏选择【文件】→“新建工程”，选择对应的系列和型号，如图 6-4 所示。

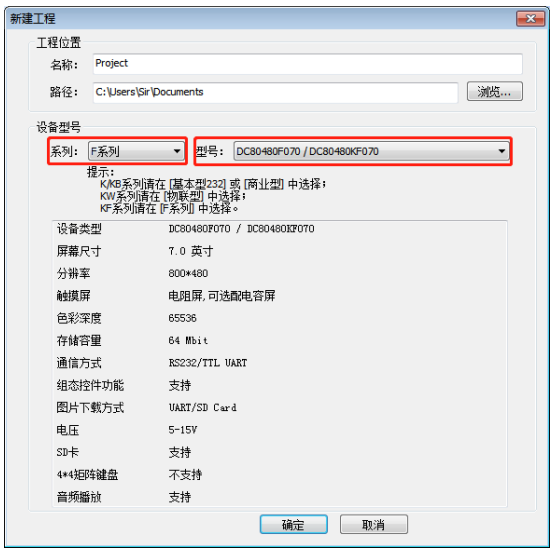


图 6-4　新建工程

6.4.2 文本控件更新

【文本控件更新】画面，介绍用户设备和串口屏通讯时，用户设备向串口屏发送指令，更新文本控件的内容。

1. 画面配置

在【文本控件更新】画面的“背景图片”导入相应的美工图片。在该画面中添加文本控件，如图 6-5 所示。

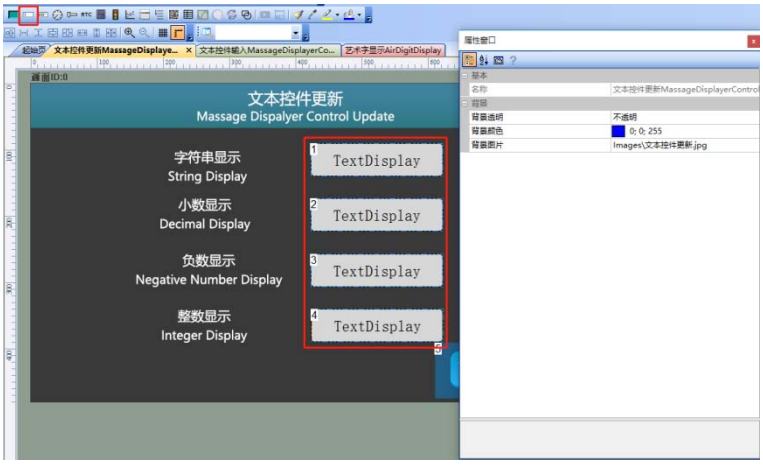


图 6-5 画面配置

2. 属性配置

以字符串显示栏为例，设置【文本】为“广州大彩”、【字体】为“ASC_CHS_24”、【垂直对齐】为“垂直居中”、【输入方式】“用户主机输入”，配置如图 6-6 所示。

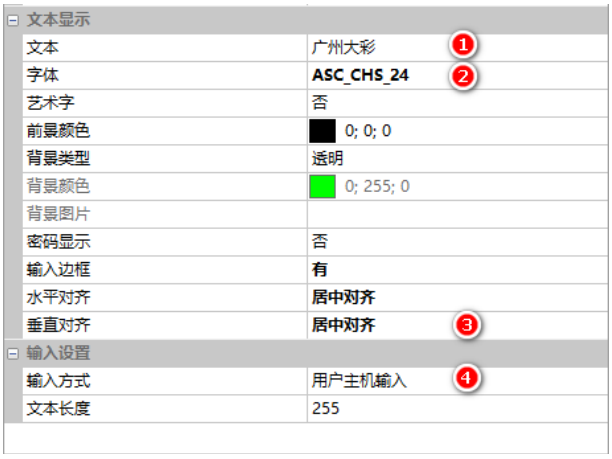


图 6-6 文本控件属性

同理，对小数显示栏、负数显示栏、整数显示栏，设置【文本】初始值依次为“0.8”、“-8”、“18”。如图 6-7 所示。



图 6-7　文本值初始化

3. 运行预览

运行虚拟屏，VisualTFT 软件和虚拟屏用“虚拟串口”建立联机。打开【指令助手】，选择左侧导航栏【文本控件 1】，设 置“更新文本控件”的指令参数。

(1) 整数显示

对画面（画面 ID：0）中的文本控件（控件 ID：1）填写【文本】为“运行预览”，点击“发送”。则在虚拟屏中，该文本控件从默认的“广州大彩”显示为“运行预览”，运行效果如图 6-8 所示。

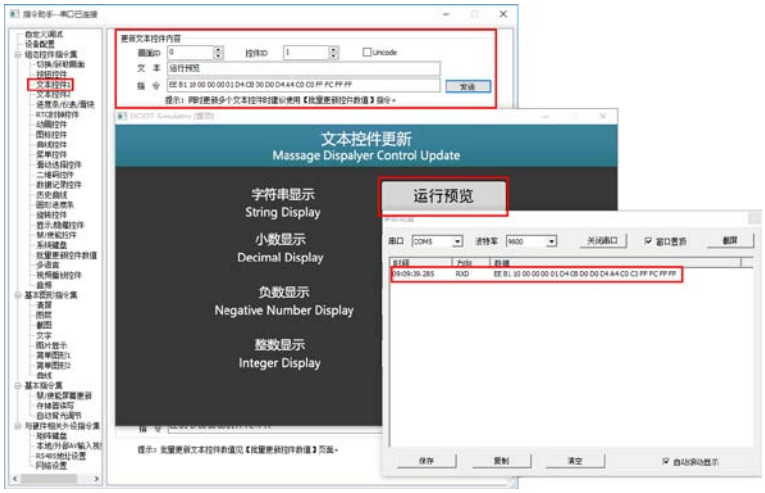


图 6-8　字符串显示

(2) 小数显示

对画面（画面 ID：0）中的文本控件（控件 ID：2）填写【文本】为“0.9”，点击“发送”，运行效果如图 6-9 所示。

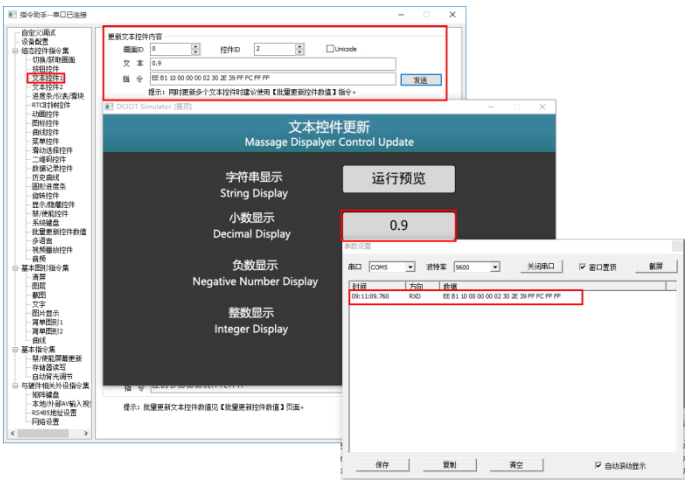


图 6-9　小数显示

(3)　负数显示

对画面（画面 ID：0）中的文本控件（控件 ID：3）填写【文本】为“-9”，点击“发送”，运行效果如图 6-10 所示。



图 6-10　负数显示

(4)　整数显示

对画面（画面 ID：0）中的文本控件（控件 ID：4）填写文本【文本】为“19”，点击“发送”，运行效果如图 6-11 所示。

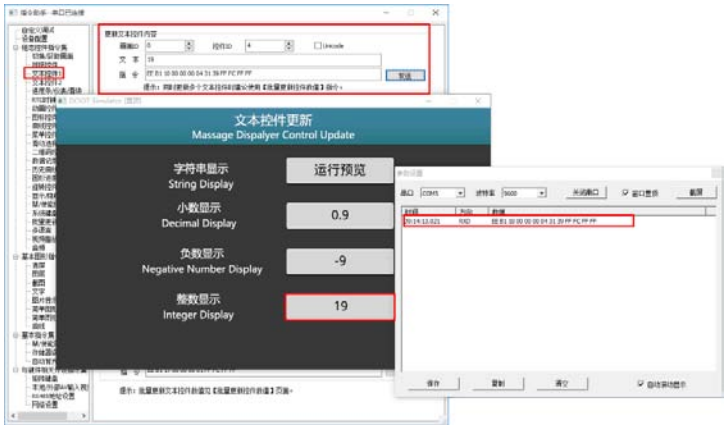


图 6-11　整数显示

4. 外部 MCU 控制

参考开发包 keil 程序中的 him.dever.h 文件函数声明以及 him.dever.c 文件中的定义。

(1) MUC 控制文本字符串显示

例如用户需实现“字符串栏显示栏”显示“运行预览”，则直接调用文本字库串显示函数 SetTextValue( )即可，代码如程序清单 1 所示。

程序清单 1 字符串显示

```
/*********************************************************************
** Function name:      void SetTextValue(uint16 screen_id,
                                         uint16 control_id,
                                         uchar *str);
** Descriptions:       设置文本字符串显示
** input parameters:    screen_id:     画面 ID
**                      control_id:    控件 ID
**                      str:           显示的字符
** output parameters:    无
** Returned value:       无
*********************************************************************/
{
   ……
   SetTextValue(0, 1, “运行预览”) //画面 ID0，控件 ID1,显示字符串“运行预览”
   ……
}
```

(2) MUC 控制显示正整数、负整数

例如用户需实现“负数显示栏”显示“-9”，“正数显示栏”显示“19”，则直接调用文本字库串显示函数 SetTextInt32( )即可，代码如程序清单 2 所示。

程序清单 2 整数显示

```
/*********************************************************************
** Function name:      void SetTextInt32(uint16 screen_id,
                                          uint16 control_id,
                                          uint32 value,
                                          uint8 sign,
                                          uint8 fill_zero);
** Descriptions:       设置文本为正/负整数
** input parameters:    screen_id:    画面 ID
**                     control_id:    控件 ID
**                     value:     显示的整数
**                     sign:     0-无符号，1-有符号
**                     fill_zero:     数字位数，不足时左侧补零
** output parameters:   无
** Returned value:      无
*********************************************************************/
{
```

```
    ……
    SetTextInt32 (0, 3, -9，1, 0) //设置画面 ID0 控件 ID3,显示-9
    SetTextInt32 (0, 4, 19, 0, 0) //设置画面 ID0 控件 ID4,显示 19
    ……
}
```

(3) MUC 控制显示小数

例如用户需实现“负数显示栏”显示“-9”，“正数显示栏”显示“19”，则直接调用文本字库串显示函数 SetTextFloat ( )即可，代码如程序清单 3 所示。

程序清单 3 小数显示

```
/*********************************************************************
** Function name:      void SetTextFloat(uint16 screen_id,
                                         uint16 control_id,
                                         float value,
                                         uint8 precision,
                                         uint8 show_zeros);
** Descriptions:       设置文本单精度浮点值/小数
** input parameters:    screen_id:    画面 ID
**                     control_id:    控件 ID
**                     value:    显示的浮点数
**                     precision:     小数位数
**                     show _zero:     为 1 时，显示末尾 0
** output parameters:   无
** Returned value:     无
*********************************************************************/
{
    ……
    SetTextFloat (0, 3, 0.9，1,1) //设置画面 ID0 控件 ID2,显示 0.9
    ……
}
```

6.4.3 文本控件输入

【文本控件输入】画面，介绍点击串口屏的文本控件弹出键盘，输入数据，主动下发数据给用户单片机。

1. 画面配置

在【文本控件输入】画面的“背景图片”导入相应的美工图片。在画面中添加 2 个文本控件（控件 ID：1 和 2），如 图 6-12 所示。

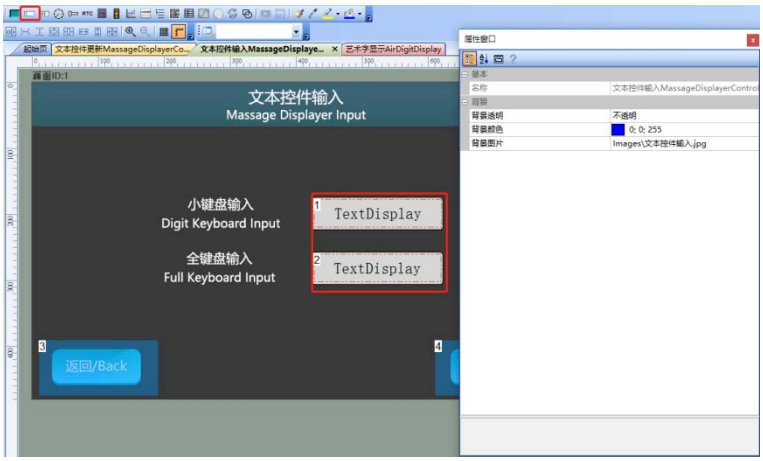


图 6-12　文本控件输入

2. 属性配置

以控件 ID 为 1 为例，设置【文本】为“88”、【字体】为“ASC_CHS_24”、【垂直对齐】为“居中对齐”、【输入方式】为“弹出系统键盘输入（用户主机可输入）”、【键盘类型】为“小键盘”、【键盘位置】为“指定”、坐标为“（300,220）”，【数值限定】为“是”，限制范围为“0-100”，置如图 6-13 所示。

① 文本	88
② 字体	ASC_CHS_24
艺术字	否
前景颜色	0; 0; 0
背景类型	透明
背景颜色	0; 255; 0
背景图片	
密码显示	否
输入边框	无
水平对齐	居中对齐
③ 垂直对齐	居中对齐
输入设置	
④ 输入方式	弹出系统键盘输入（用户主机可输入）
⑤ 键盘类型	小键盘
⑥ 提示信息	请输入
⑦ 键盘位置	指定
X坐标	300
Y坐标	220
初始值	无
文本长度	255
⑧ 数值限定	是
小数位数	0
最大值	100
最小值	0

图 6-13　键盘设置

同理，全键盘输入的配置只需将“键盘类型”改为“全键盘”即可，其他参数根据需求配置。

3. 运行预览

运行虚拟屏，点击画面（画面 ID：1）中的文本控件（控件 ID：1），弹出键盘，输入“66”，回车。则虚拟屏下发对应的指令，运行效果如图 6-14 所示。



图 6-14　运行预览

6.4.4 艺术字显示

【艺术字显示】画面，使用艺术字显示效果，首先生成对应的 ICON 文件，在文本控件的属性栏窗口，将 ICON 文件添加到“图标文件”配置选项中。

1. 画面配置

在【艺术字显示】画面的“背景图片”导入相应的美工图片。画面中添加 1 个文本控件（控件 ID：1），如 图 6-15 所示。

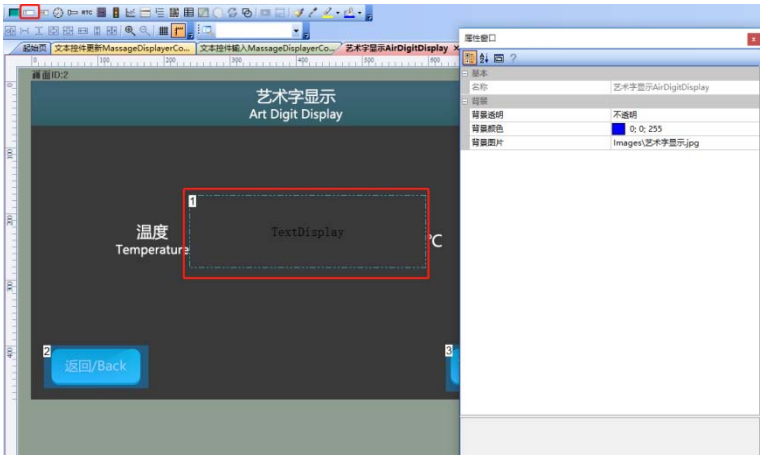


图 6-15　艺术字显示

2. 艺术字生成

图标文件制作有两种方式，分别为图标生成和艺术字生成。

(1) 图标文件生成艺术字

a) 打开图标生成界面

在 VisualTFT 软件菜单栏，选择【工具】→“图标生成”，如图 6-16 所示。

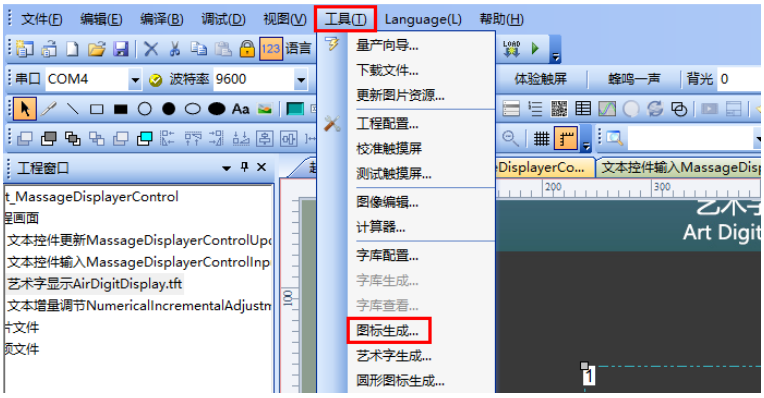


图 6-16 图标生成

b) 图标制作

点击【增加帧】,将准备好的素材严格按照“[0-9]、[: . * - + ? /]、[A-Z]和[a-z]”顺序添加，否则会导致显示错误。用户导入图片默认为原始的宽和高显示，也可以对宽度和高度的微调及任意拖动进行位置调整，最后点击“生成图标”。如图 6-17 所示。



图 6-17 制作过程

(2) 系统自动生成艺术字

a) 打开艺术字生成界面

在 VisualTFT 软件菜单栏，选择【工具】→“图标生成”，如图 6-18 所示。

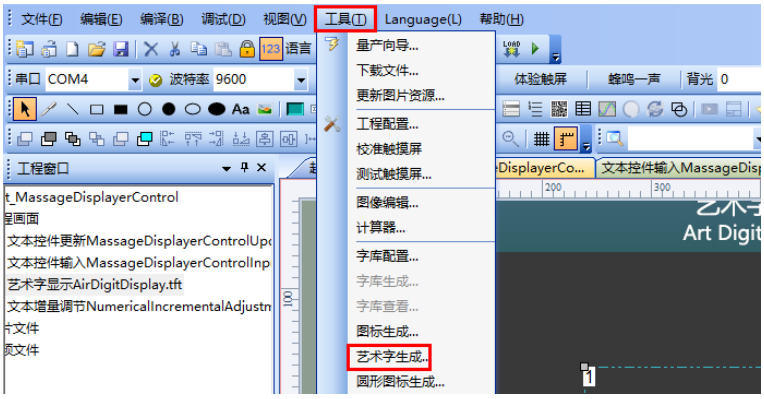


图 6-18 艺术字生成

b) 图标制作

- 设置图标的大小：高度和宽度为 128*64。
- 字体名称：根据显示效果，选择相应 TFT 字库文件，可网上下载获得。
- 字体大小：默认显示的字体大小。
- 字体颜色：设置字体显示的颜色。
- 字符集合：共 3 中，如下所示。

  数字：数字 0~9。

  数字和符号：数字 0~9、字符[:.*-+?/]。

  数字/符号/字母：数字 0~9、字符 [:.*-+?/]、大写 A~Z，小写 a~z。
- 渐变处理：禁止渐变、填充背景色、ALPH 通道。
- 背景颜色：显示字体默认的背景色。

设置如图 6-19 所示。点击【生成图标】完成图标生成。

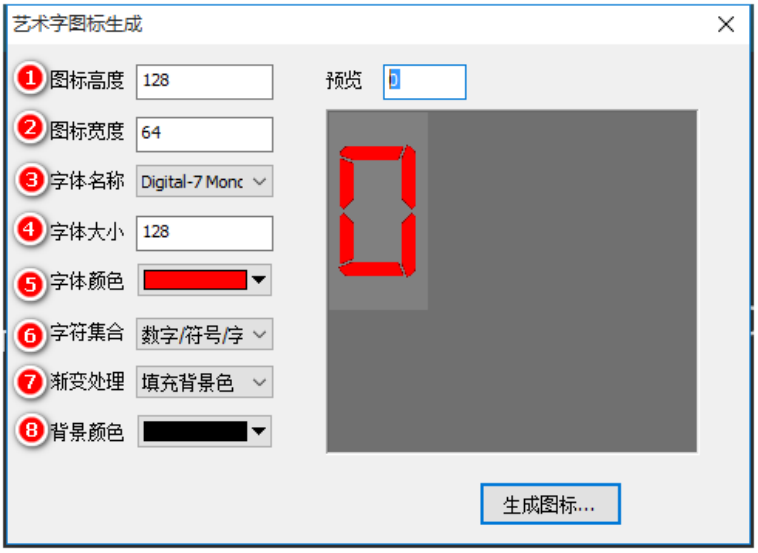


图 6-19 制作过程

3. 属性配置

设置【艺术字】为“是”、【图标文件】插入“艺术字.icon 文件”、【小数点宽度】“为 25%”、【垂直对齐】为“居中对齐”、【输入方式】为“用户主机输入”，配置如图 6-20 所示。



图 6-20 艺术字配置

4. 运行预览

运行虚拟屏，VisualTFT 软件和虚拟屏用“虚拟串口”建立联机。打开【指令助手】，选择左侧导航栏【文本控 1】，设置“更新文本控件”指令参数。如例程中，控制画面（画面 ID：2）中的文本控件（控件 ID：1），填写【文本】为“123”，点击“发送”。则虚拟屏文本控件 1 显示从默认的“18.8”显示为“123”，运行效果如图 6-21 所示。

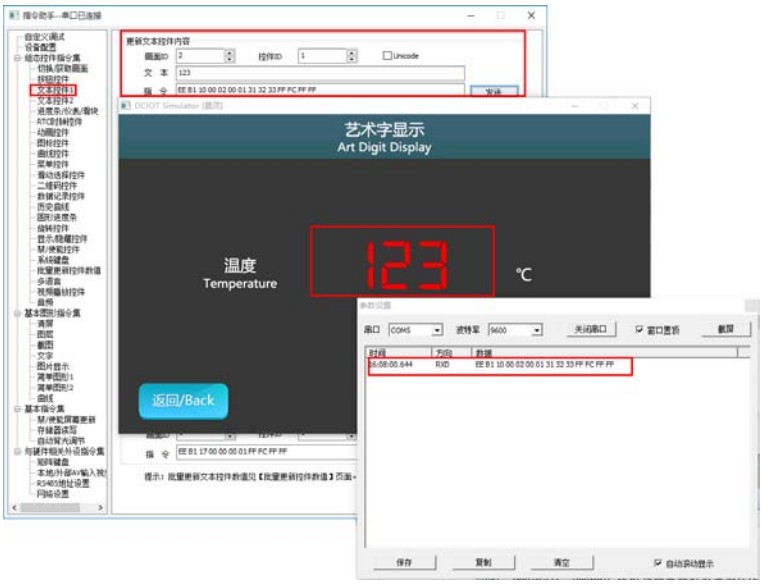


图 6-21 运行预览

6.4.5 文本增量调节

通过按钮控件控制文本控件显示值“自增”或“自减”操作。若调节的步长为整数，则增量调节的操作在屏内即可实现；若调节的步长为小数，则需要 MCU 数据交互实现。

1. 整数调节

(1) 画面配置

在【文本增量调节】画面的“背景图片”导入相应的美工图片。画面中添加 1 个文本控件（控件 ID：3），2 个按钮控件（控件 ID：1 和 2），如图 6-22 所示。



图 6-22 整数增量调节

(2) 属性配置

a) 文本属性

控件 ID 为 3 的文本控件，设置文本为“0”、字体为“ASC_CHS_24”、垂

直对齐为“垂直居中”、输入方式“用户主机输入”，配置如图 6-23 所示。

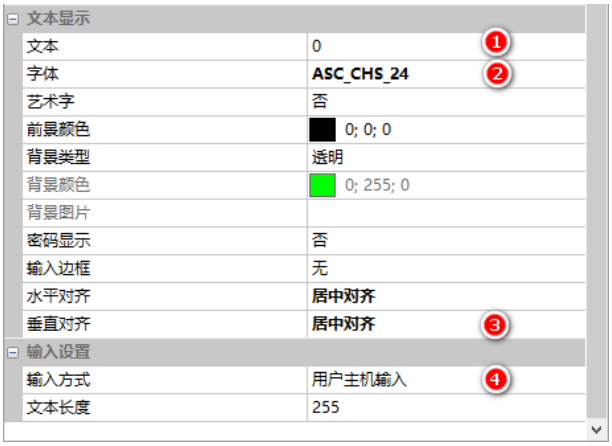


图 6-23　文本控件属性

b)　按钮属性

选择按钮控件（控件 ID：1）属性栏中的【对内指令】→“按钮按下”，弹出指令填写框，选择“指令助手”，如图 6-24 所示。

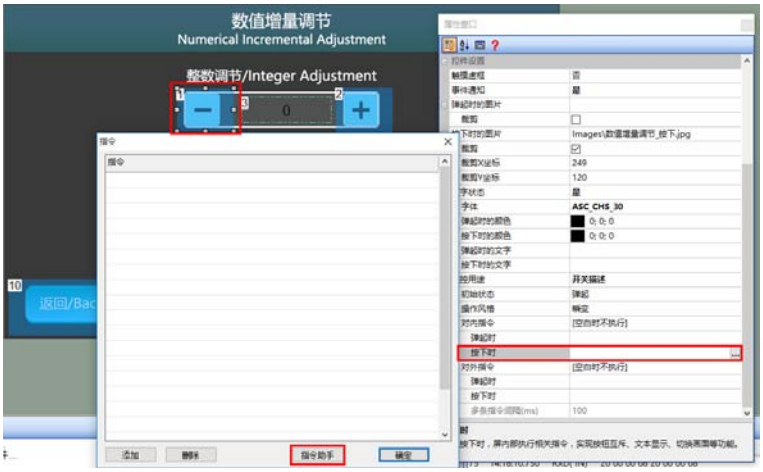


图 6-24　对内指令

在指令助手界面，左侧导航栏选择【文本控 2】，设置“增量调节”区域的指令参数。如例程中，控制画面（画面 ID：3）中的文本控件（控件 ID：3）在 0-100 范围内以步长为 1 的单位递减，则填写参数如图 6-25 所示。

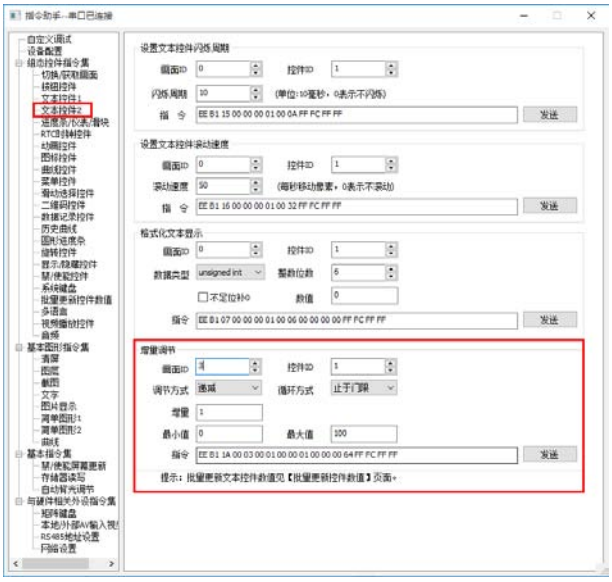


图 6-25　指令助手

复制此指令，关闭指令助手，将指令粘贴到指令框中，如图 6-26 所示。

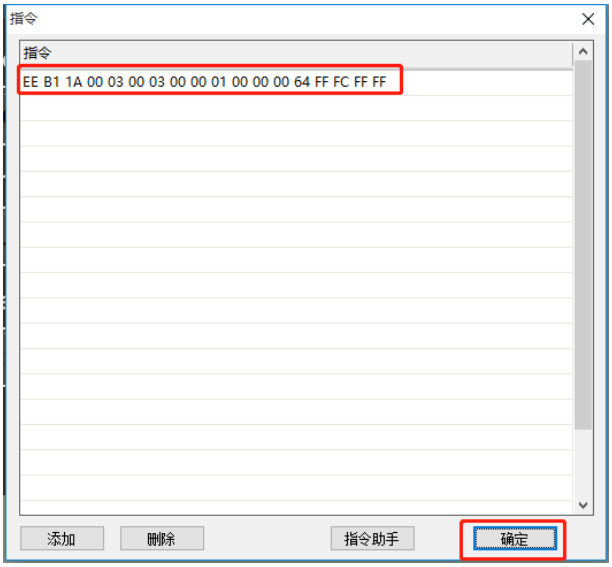


图 6-26　填写指令

同理，递增的操作修改【调节方式】为“递增”即可。注意：该对内指令增量调节的方法仅适合调节的步长为整数，不支持小数。

(3)　运行预览

运行虚拟屏，点击画面（画面 ID：3）中的按钮控件（控件 ID：1 或 2），以 1 的步长调整文本控件的数值，运行效果如图 6-27 所示。



图 6-27 运行预览

2. 小数调节

实现在 0~6 范围内，长按按钮控件以 0.1 步长增量调节数值，需要和单片机交互处理。参考开发包 keil 程序中的 him.dever.h 文件函数声明以及 him.dever.c 文件中的定义。

(1) 画面配置

在【文本增量调节】画面中画面中添加 1 个文本控件（控件 ID：6），2 个按钮控件（控件 ID：4 和 5），如图 6-28 所示。



图 6-28 小数调节

(2) 属性配置

控件 ID 为 6 的文本控件，设置文本为“0”、字体为“ASC_CHS_24”、垂直对齐为“垂直居中”、输入方式“用户主机输入”，配置如图 6-29 所示。

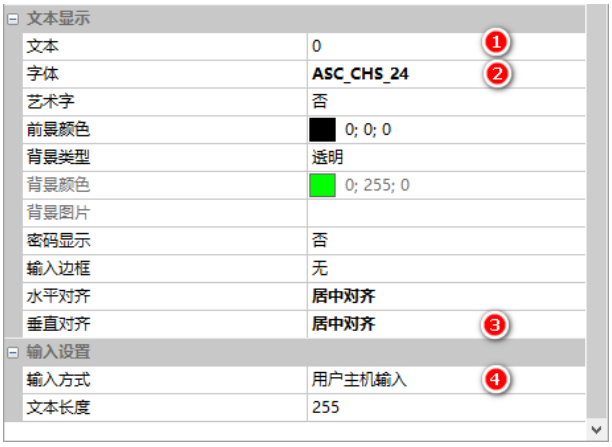


图 6-29　文本控件属性

(3)　实现方法

a)　按下递增按钮（控件 ID：5）时，屏幕下发按钮控件的指令到单片机。

b)　单片机解析按钮控件指令后，若判断是按钮控件（控件 ID：5）按下，则发送获取文本控件（控件 ID：6）的指令，获取文本控件值。相关代码程序清单 4 所示。

程序清单 4　获取文本值

```
/*************************************************************************
** Function name:      NotifyButton(uint16 screen_id,
                                    uint16 control_id,
                                    uint8  state)
** Descriptions:       当按钮状态改变(或调用 GetControlValue)时，执行此函数
** input parameters:    screen_id:    画面 ID
**                      control_id:    控件 ID
**                      state:    按钮控件的状态，0 弹起、1 按下、2 长按
** output parameters:   无
** Returned value:      无
*************************************************************************/
void NotifyButton(uint16 screen_id, uint16 control_id, uint8  state)
{
    ……
    //按下画面 ID3 控件 ID5 的按钮控件
    if(screen_id == 3 && control_id == 5 && state == 1) {
    /*****************************************************************
    ** Function name:      GetControlValue(uint16 screen_id,
                                           uint16 control_id)
    ** Descriptions:       获取控件值
    ** input parameters:    screen_id:    画面 ID
    **                      control_id:    控件 ID
    ** output parameters:   无
    ** Returned value:      无
```

```
        ****************************************************************/
            GetControlValue(3,6);    //获取画面 ID3，控件 ID6 的文本控件值
        }
        ……
    }
```

c) 获取的文本控件值默认是字符串，调用 C 库函数的 sscanf 函数，将字符串转成单精度浮点数（小数），在进行 0.1 个单位递增运算，在将文本控件值更新到串口屏显示。代码如程序清单 5 所示。

程序清单 5 显示文本值

```
void NotifyText(uint16 screen_id, uint16 control_id, uint8 *str)
{
    float TestValue = 0.0;
    ……
    if(screen_id == 3 && control_id == 6)    //画面 ID2：文本设置和显示
    {
        sscanf(str,"%%.1f",&TestValue);    //把字符串转换为浮点数
        TestValue += 0.1;    // 文本值自增
        /*****************************************
        ** Function name:      void SetTextFloat(uint16 screen_id,
                                                  uint16 control_id,
                                                  float value,
                                                  uint8 precision,
                                                  uint8 show_zeros);
        ** Descriptions:       设置文本单精度浮点值/小数
        ** input parameters:    screen_id:    画面 ID
        **                     control_id:    控件 ID
        **                     value:    显示的浮点数
        **                     precision:     小数位数
        **                     show _zero:     为 1 时，显示末尾 0
        ** output parameters:   无
        ** Returned value:     无
        ****************************************************************/
        SetTextFloat(3,6,TestValue,1,1);//设置画面 ID3、控件 ID6 每次自增的数值
    }
    ……
}
```

注：同理，时间的增量调节，用户可参考小数增量的方法，结合用户主机完成数据交互，用户主机只要对应调用字符串处理函数实现即可

6.5 下载工程

在我司的上层软件 Visual TFT 把工程编译成功后，可以通过以下两种方法下载到大彩串口屏设备中：

1. 串口下载；
2. SD 卡下载

注：本例程中以 F 型为例，不同平台下载方式略有不同，具体参考不同平台的下载文档。

6.6 联机调试

6.6.1 硬件连线

硬件连接接线如图 6-30 所示。



图 6-30 硬件连线

6.6.2 联机设备

将工程下载到串口屏后，在 VisualTTF 的菜单栏，点击【联机设备】，如图 6-31 所示。

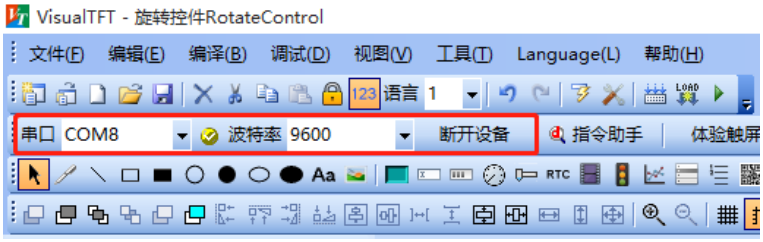


图 6-31 联机设备

6.6.3 指令控制

打开【指令助手】，选择左侧导航栏【文本控件】，设置“更新文本控件”区域的指令参数。如设置画面 0 的文本控件 3 的显示“-18.5”，如 图 6-32 所示

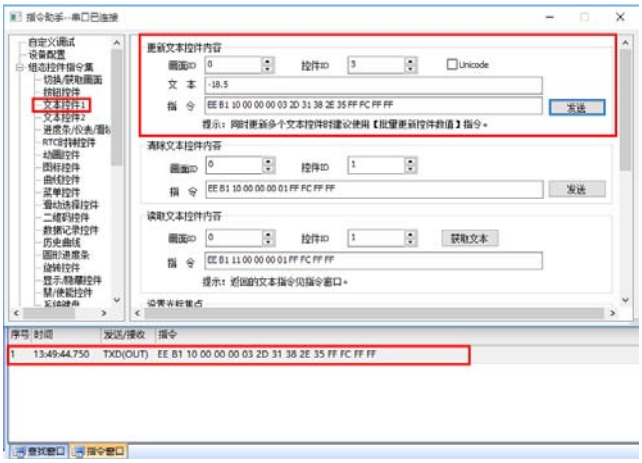


图 6-32 文本更新显示

在指令助手里，选择左侧导航栏【文本控件 2】，设置“文本滚动速度”区域和“文本闪烁周期”的指令参数。如设置画面 0 的文本控件 3 以 500ms 的周期闪烁，并以 500ms 的速度滚动，如图 6-33 所示

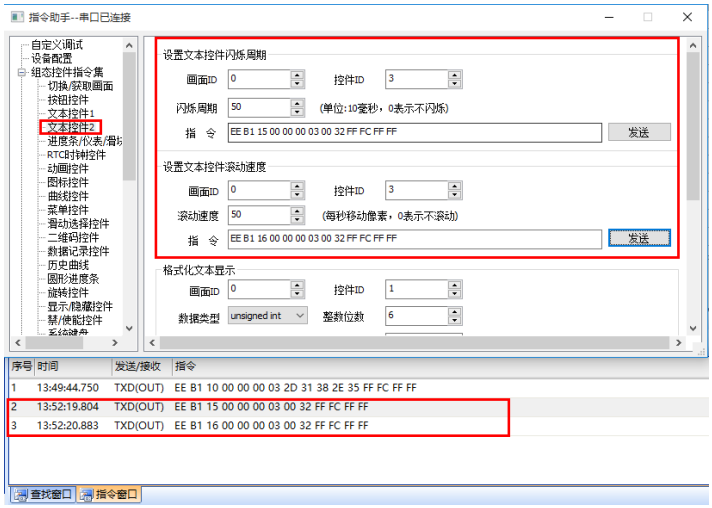


图 6-33　设置文本闪烁和滚动

6.6.4 串口屏效果

串口屏接收到指令后，文本控件 3 显示-18.8，并从右向左以 500ms 周期滚动和闪烁。

7. 部分程序代码

我司开发包里提供用户单片机控制程序，里面提供 API 函数接口，用户可移植相应的程序；本例程涉及的用户单片机控制文本控件更新的程序，参考开发包 keil 程序中的 him.dever.h 文件函数声明以及 him.dever.c 文件中的定义。

keil 程序中的函数定声明如程序清单 6 所示。

程序清单 6　文本控件 keil 程序

```
/*********************************************************************
** Function name:      void SetTextValue(uint16 screen_id,
                                         uint16 control_id,
                                         uchar *str);
** Descriptions:       设置文本字符串显示
** input parameters:    screen_id:     画面 ID
**                      control_id:    控件 ID
**                      str:           显示的字符
** output parameters:    无
** Returned value:      无
*********************************************************************/
{
    ……
    SetTextValue(0, 1, “运行预览”) //设置画面 ID0、控件 ID1 显示“运行预览” ,
    ……
}

/*********************************************************************
** Function name:      void SetTextFloat(uint16 screen_id,
                                         uint16 control_id,
                                         float value,
                                         uint8 precision,
                                         uint8 show_zeros);
** Descriptions:       设置文本单精度浮点值/小数
** input parameters:    screen_id:    画面 ID
**                     control_id:    控件 ID
**                     value:    显示的浮点数
**                     precision:     小数位数
**                     show _zero:     为 1 时，显示末尾 0
** output parameters:   无
** Returned value:      无
*********************************************************************/
{
    ……
    SetTextFloat (0, 3, 0.9，1,1) //设置画面 ID0 控件 ID2,显示 0.9
    ……
```

```
}

/*********************************************************************
** Function name:      void SetTextInt32(uint16 screen_id,
                                         uint16 control_id,
                                         uint32 value,
                                         uint8 sign,
                                         uint8 fill_zero);
** Descriptions:       设置文本为正/负整数
** input parameters:    screen_id:    画面 ID
**                      control_id:    控件 ID
**                      value:    显示的整数
**                      sign:    0-无符号，1-有符号
**                      fill_zero:    数字位数不足补零
** output parameters:   无
** Returned value:     无
*********************************************************************/
{
   ……
   SetTextInt32 (0, 3, -9, 1, 0) //设置画面 ID0 控件 ID3,显示-9
   SetTextInt32 (0, 4, 19, 0, 0) //设置画面 ID0 控件 ID4,显示 19
   ……
}
 /*********************************************************************
 ** Function name:      GetControlValue(uint16 screen_id,
                                        uint16 control_id)
 ** Descriptions:       获取控件值
 ** input parameters:   screen_id:    画面 ID
 **                      control_id:    控件 ID
 ** output parameters:   无
 ** Returned value:      无
 *********************************************************************/
 {
    ……
       GetControlValue(3,6);    //获取画面 ID3，控件 ID6 的文本控件值
    ……
 }
```

8. 免责声明

本文档提供有关广州大彩光电科技有限公司（以下简称：大彩科技）产品的信息，旨在协助客户加速产品的研发进度，在服务过程中或者其他渠道所提供的任何例程程序、技术文档、CAD 图等资料和信息都仅供参考，客户有权不使用或自行参考修改。本公司不提供任何的完整性、可靠性等保证，若在客户使用过程中因任何原因造成的特别的、偶然的或间接的损失，本公司不承担任何责任。大彩科技产品不能在用于军事、医疗、救生或维生等用途中作为唯一控制设备。

本文档并未授予任何知识产权的许可，并未以明示或暗示，或以禁止发言或其它方式授予任何知识产权许可。除大彩科技在其产品的销售条款和条件中声明的责任之外，大彩科技概不承担任何其它责任。并且，大彩科技对大彩科技产品的销售和/或使用不作任何明示或暗示的担保，包括对产品的特定用途适用性、适销性或对任何专利权、版权或其它知识产权的侵权责任等，均不作担保。大彩科技可能随时对产品规格及产品描述做出修改，恕不另行通知。