



Z4000A 系列产品 快速入门手册

安徽世龙电子技术有限公司

V3.1

文档使用说明

本文档适用于 Z4000A 系列智能人机界面产品的快速入门，详细产品资料请参考：Z4000 用户使用手册。

阅读本文档前，您需要具备：

- Windows 系统电脑（WinXP、Win7、Win8、Win10）
- Z4000 配套上位机软件 PZ4000Builder
- Z4000 系列开发包
- Z4000A 系列显控模组
- 8P 通信电缆线
- USB 线

如有缺失，请联系世龙电子。

本文档工程制作部分所用素材均来自 Z4000 系列开发包，按照步骤可制作出和开发包中的 Z4000A 快速入门演示工程一致的工程，演示工程适用于 Z4000A 系列产品。

在产品使用过程中，如有任何疑问，欢迎联系我们。

网址：www.worldlong.com

电话：0551-65329091 65329092

邮箱：wl_support@126.com

目录

1 使用步骤	3
1.1 软件安装	3
1.2 认识软件	3
1.3 工程制作	3
1.3.1 寄存器分配	4
1.3.2 新建工程	4
1.3.3 新建画面	5
1.3.4 添加控件	6
1.3.5 工程设置	10
1.3.6 编译	11
1.3.7 寄存器配置	11
1.4 工程下载	12
1.4.1 联机通信	12
1.4.2 开始下载	13
1.5 联机调试	14
1.6 编写程序	16
1.7 实际测试	16
2 演示工程讲解	18
2.1 静态控件和生成 BMP 图片	18
2.2 动画演示	19
2.3 时间显示	21
2.4 弹出窗口	23
2.5 数值输入与显示	27
2.6 ASCII 文本输入与显示	30
2.7 动态文本输入与显示	33
2.8 表格	36
2.9 波形	38
2.10 二维码	41
2.11 画面切换	43
3 辅助功能	45
3.1 用户字库	45
3.2 图片替换	47
3.3 画面复制	48

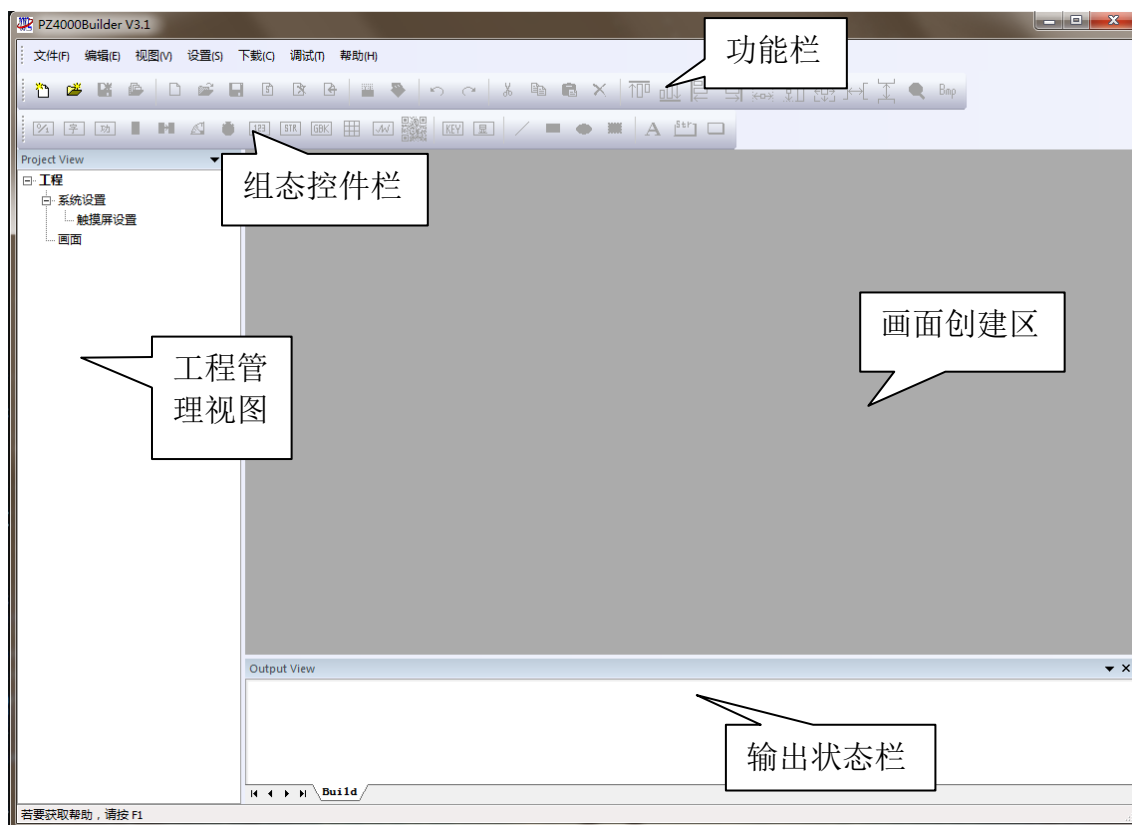
1 使用步骤

按照步骤进行，即可完成一个 Z4000 工程，实现简单功能。

1.1 软件安装

- ✧ 在 Z4000 系列开发包—开发软件—注册包内选取适合电脑系统的注册包
- ✧ 执行注册包中“使用方法.txt”的安装步骤
- ✧ 双击 PZ4000 Builder V3.1.exe，打开软件

1.2 认识软件



1.3 工程制作

Z4000 的工程制作分为寄存器分配、新建工程、新建画面、添加控件、工程设置、编译和寄存器配置七个步骤。

在开发包中附有图库，用户在工程制作的初期可以先使用图库中的图片，等工程调试成功后，再请美工进行界面的美化工作。

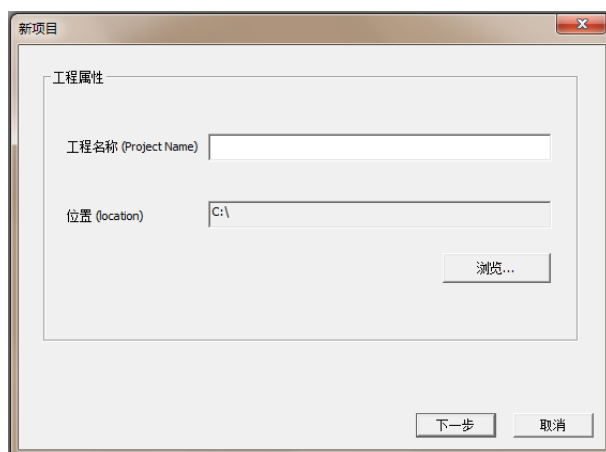
1.3.1 寄存器分配

Z4000 是基于寄存器工作的，随着寄存器值的不同，Z4000 可以变化显示内容。Z4000A 系列具有 8KB 的变量寄存器和 24KB 的字符串寄存器（共有 384 个字符串，每个字符串占 64 个字节）。

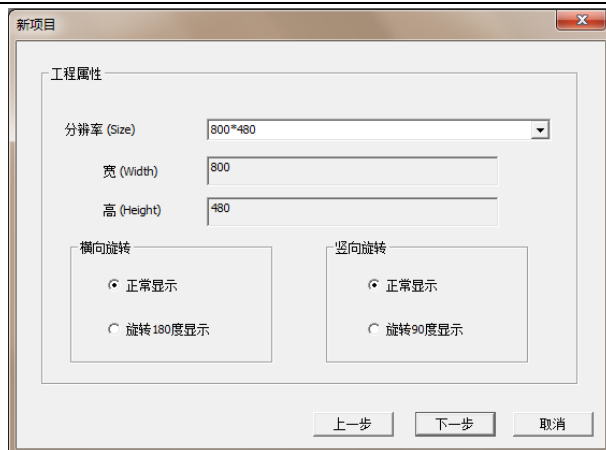
首先，用户需要进行寄存器分配工作，即根据使用需要，整理、分配变量和字符串空间，注意区分寄存器类型。建议用户使用表格进行记录和整理，方便后期的查看、修改。变量寄存器的分配可以参考开发包中的 [Z4000A 快速入门演示工程之变量寄存器分配表.xls](#)。初次使用可以先跳过这一步。

1.3.2 新建工程

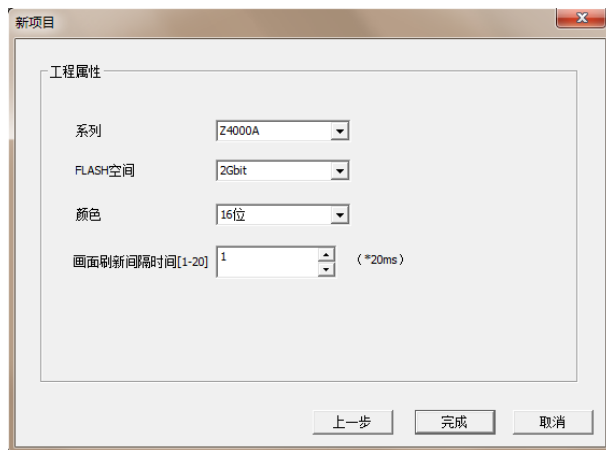
✧ 单击“新建项目”按钮，打开窗口如下：



- ✧ 输入工程名称
- ✧ 选择工程所在位置
- ✧ 单击“下一步”



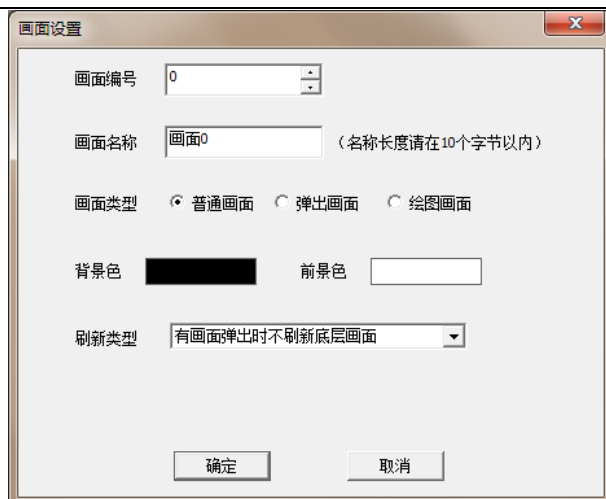
- ✧ 根据 Z4000 显控模组的型号在下拉菜单中选择分辨率
- ✧ 单击“下一步”



- ✧ 根据 Z4000 显控模组的型号在菜单中选择系列、FLASH 空间、颜色
- ✧ 设定画面刷新闻隔时间，单位为 20ms。建议先设为 1，最后根据实际情况再进行修改
- ✧ 单击“完成”

1.3.3 新建画面

- ✧ 单击“新建画面”按钮，打开窗口如下：



✧ 设定画面编号

✧ 输入画面名称

✧ 选择画面类型：普通画面、弹出画面、绘图画面

普通画面是 Z4000 常规显示的画面；弹出画面是固定在 1 层上弹出显示的画面；绘图画面只用于添加静态控件并生成.bmp 图片，绘图画面不会在 Z4000 中显示。

✧ 选择背景色，即当前画面的背景颜色

✧ 选择前景色，即在当前画面绘制静态图形时默认的图形颜色

✧ 选择刷新类型：当画面类型为普通画面时，此选项可选。当有画面从当前画面弹出时，若不需刷新该普通画面的控件时，选择“有画面弹出时不刷新底层画面”；否则，选择“有画面弹出时刷新底层画面”。

✧ 单击“确定”

以演示工程中的第一个画面为例，画面名称设为“控件示例”，其余保留默认值不变。

1.3.4 添加控件

单击组态控件栏中的控件图标，此时在画面上鼠标变为十字光标，在画面的恰当位置处单击或者进行拖动，然后在弹出的对话框中进行控件设置。最后单击“确定”即可完成控件的添加。双击控件可以再次对控件进行设置。

以演示工程中第一个画面为例，添加控件。

✧ 静态图片控件

✧ 单击组态控件栏中的“静态图片”，在画面恰当位置单击，弹出静态图片设置对话框



- ✧ 输入左上角 X、Y 坐标均为 0
- ✧ 单击“打开图片”，选择开发包中演示工程文件夹—Z4000 快速入门演示工程文件夹—图片文件夹（以下简称“图片文件夹”）中的控件示例.jpg
- ✧ 单击“确定”
- ✧ 位按钮控件
 - ✧ 单击组态控件栏中的“位按钮”，在画面恰当位置单击，弹出位按钮设置对话框
 - ✧ 在“一般”选项卡中，输入左上角 X 坐标为 156、Y 坐标为 132，将功能选为“交替”，将串口设为“返回串口”



- ✧ 在“状态”选项卡中，单击“批量添加图片”，选择图片文件夹中的按钮 0.bmp和按钮 1.bmp两张图片



- ✧ 其余保留默认值不变，单击“确定”
- ✧ 数值控件
 - ✧ 单击组态控件栏中的“数值”按钮，在画面恰当位置单击，弹出数值设置对话框
 - ✧ 在“显示”选项卡中，输入左上角 X 坐标为 452、Y 坐标为 296，设定总位数为 1



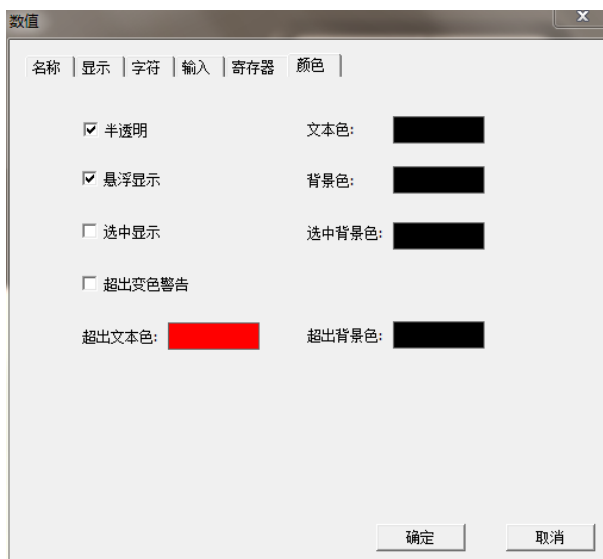
- ✧ 在“字符”选项卡中下拉设定字体高为 32



✧ 在“寄存器”选项卡中选择寄存器类型为“位寄存器”



✧ 在“颜色”选项卡中勾选“半透明”、设置文本色为黑色

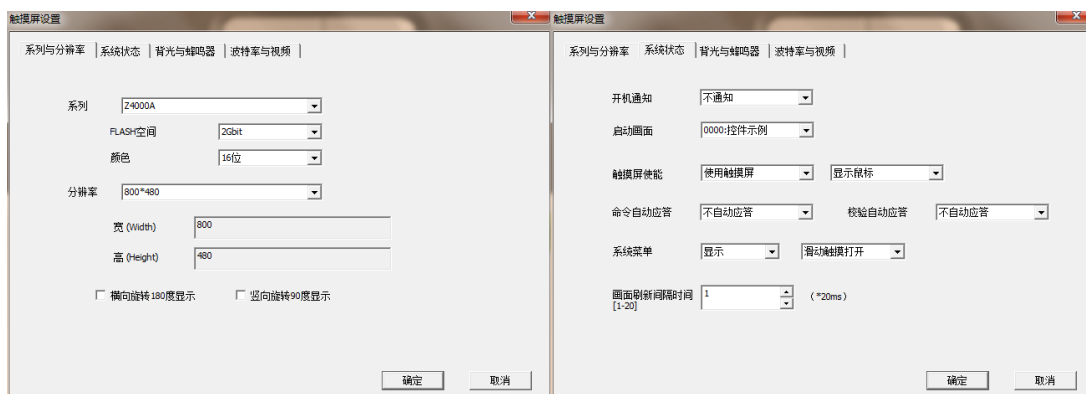


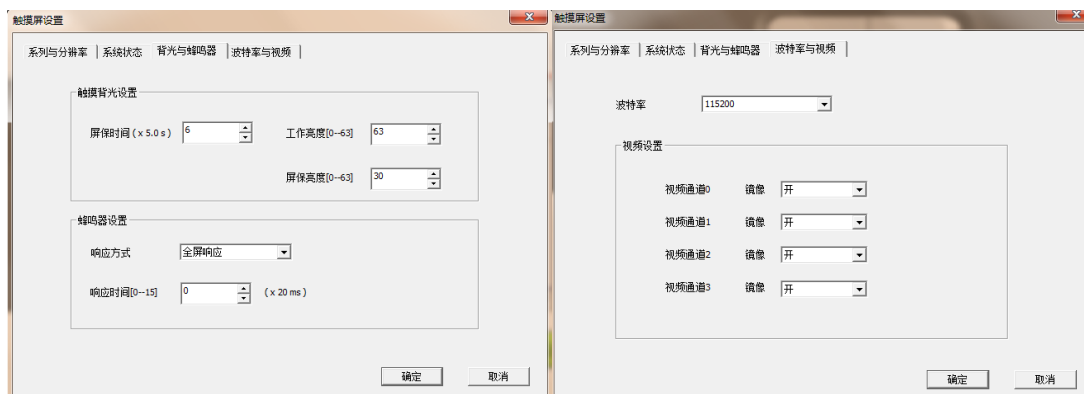
✧ 其余保留默认值不变，单击“确定”
最终画面效果如下：



1.3.5 工程设置

✧ 右击工程管理视图中的“触摸屏设置”——“设置”或是单击“设置”——“工程设置”，打开工程设置窗口

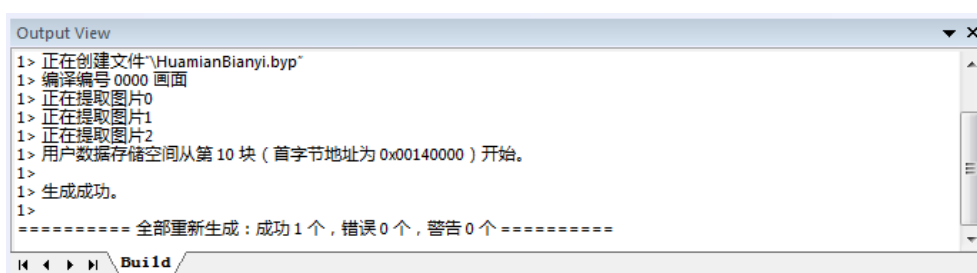




✧ 在 4 个选项卡中对工程进行设置，建议初次接触的用户保留默认值不变

1.3.6 编译

✧ 单击功能栏中的“编译”按钮 ，进行工程编译，在软件界面下端的输出框中显示编译状态

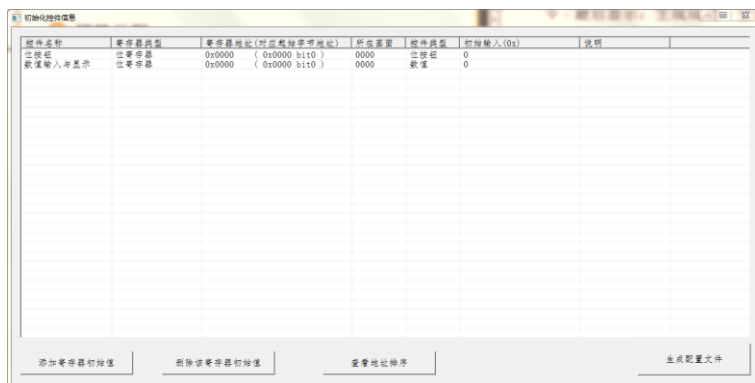


✧ 最后显示：生成成功 全部重新生成：成功 1 个，错误 0 个，表明编译成功
✧ 若出现错误，根据提示查找并修改

1.3.7 寄存器配置

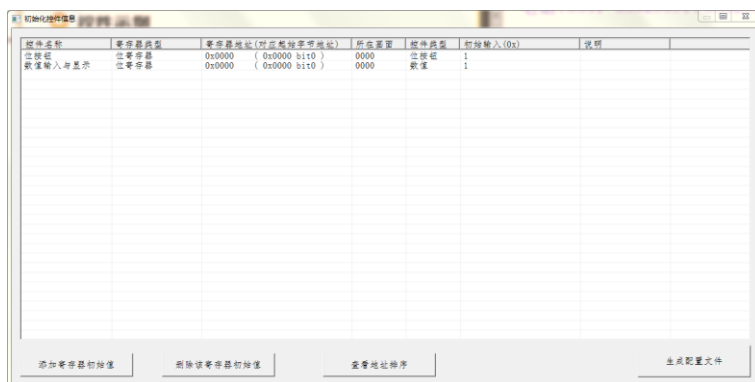
寄存器配置分为变量寄存器配置、字符串寄存器配置和波形寄存器配置，这里只介绍变量寄存器配置，字符串和波形寄存器的配置与变量寄存器配置类似。

✧ 单击“设置”——“变量寄存器配置”，弹出变量寄存器配置窗口



可以看到目前工程中有 2 个控件，分别是位按钮和数值控件，均在画面 0 中。
两个控件对应同一寄存器：地址为 0 的位寄存器

- ✧ 双击任一行，弹出初始输入窗口，默认为 0，改为 1，单击“确定”
则地址为 0 的位寄存器初始值为 1

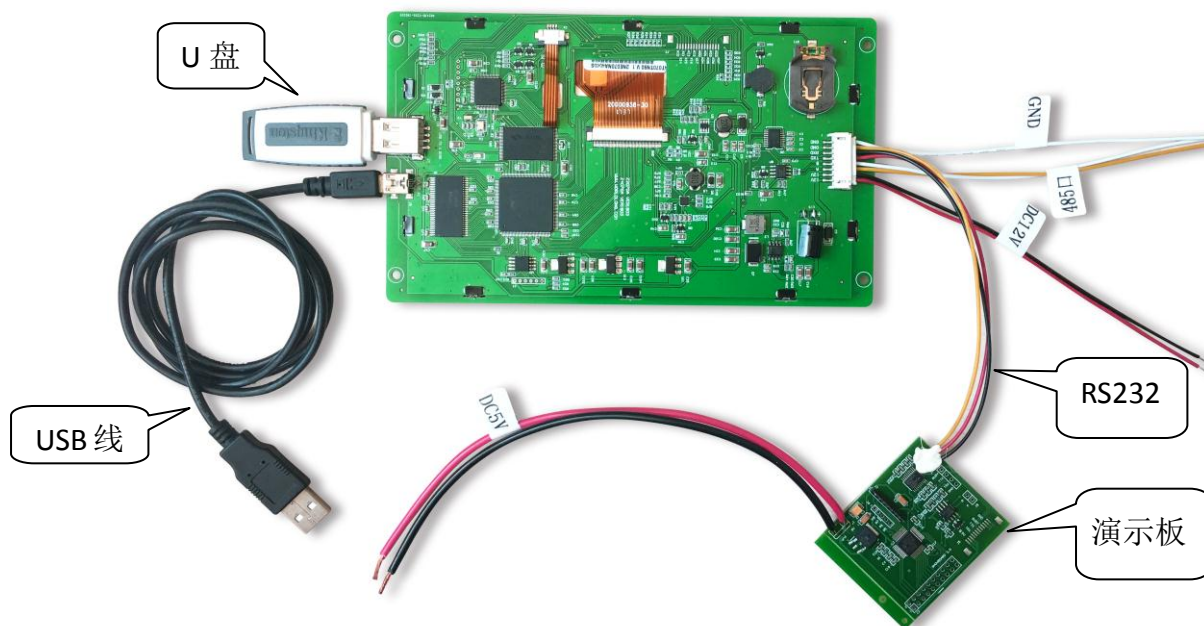


- ✧ 单击“生成配置文件”

1.4 工程下载

1.4.1 联机通信

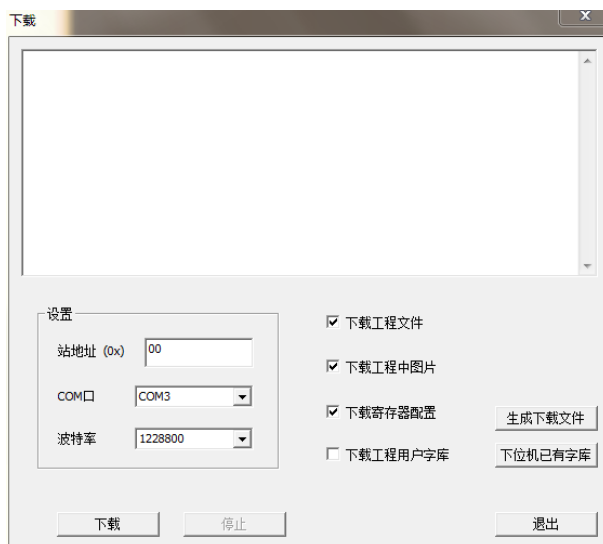
- ✧ 按图将 8P 的通信电缆线接在 Z4000 系列显控模组的用户通信接口上，8 个脚依次为：GND、GND，RXD、TXD（232 接口），B、A（485 接口），VCC、VCC。其中 VCC 接电源正极（典型值为 DC12V），GND 接电源负极，其余接口不接。



- ✧ 按图通过 USB 线将 Z4000 系列显控模组与电脑相连，U 盘暂不接。
- ✧ 供电后，查找电脑—设备管理器中，COM 口是否存在，如下图所示。若没有 COM 口，需安装开发包中的 USB 驱动（Z4000 系列开发包—开发软件—驱动）。

1.4.2 开始下载

- ✧ 单击“下载”—“工程下载”或是单击功能栏中的“下载”按钮，弹出下载窗口



- ✧ COM 口选择栏中出现和电脑设备管理器中名称一致的 COM 口
 - ✧ 保留默认值不变，单击“下载”
- 共有 4 个选项：下载工程文件、下载工程中的图片、下载寄存器配置和下载工程

用户字库，根据需要勾选需要下载的项。

✧ 下载窗口中显示下载进程，直至出现“下载完成”，表明下载结束
下载工程中，Z4000 显示“产品升级中...”

✧ 关闭下载窗口

可能出现的问题及解决办法：

1、无法打开下载窗口

解决办法：正确安装注册包

2、没有可选的 COM 口、串口连接错误、无法下载或是无法全部下载

解决办法：查看 Z4000 显控模组是否正确上电

重新插拔 USB 口

断电后再次上电

查看电脑设备管理器中 COM 是否存在

正确安装开发包中的驱动

查看通信口是否被其他软件占用

3、下载的工程与软件制作的工程不一致或是出错

解决办法：重新编译工程

下载过程中保持下载线稳定

下载工程中关闭用户串口通信

若还是无法正常使用，请联系世龙电子技术支持。

下载完成后，Z4000 显控模组会自动复位，在屏幕上即可看到之前制作的工程画面。

位按钮显示添加的按钮 1.bmp 图片，数值控件显示 1，如下左图所示。触摸位按钮后，位按钮区域显示添加的按钮 0.bmp 图片，数值控件显示 0，如下右图所示。重复触摸位按钮，则位按钮和数值控件切换显示上述两种状态。



寄存器值为 1

寄存器值为 0

由于 Z4000 显控模组采用寄存器值直接驱动的显示方式，改变寄存器的值即可实现对屏的操作。在上述操作中，通过触摸位按钮控件改变了位寄存器的值，进而改变该寄存器关联的数值控件。

1.5 联机调试

保持 Z4000 接线状态不变

- ✧ 单击“调试”——“联机调试”，弹出联机调试窗口



在 Z4000 正确接线时，可以看到 COM 口选择栏中出现和电脑设备管理器中名称一致的 COM 口，若出现错误请参照上一步工程下载中的解决办法。

- ✧ 选择“变量调试”——“读变量寄存器”

此时，下拉菜单中有 2 个选项：

0x0000 位按钮 画面 0000 位按钮 bit0

0x0000 数值输入与显示 画面 0000 数值 bit0

表明：0 号画面有位按钮和数值 2 个控件，使用同一个位寄存器，该位寄存器是字节寻址为 0 的字节寄存器的 bit0。

- ✧ 单击“打开串口”

- ✧ 单击“发送命令”

此时，在通讯记录框中出现：

发送：1B 00 04 1D 00 00 01 86 6D 表明：读取当前 0 地址的字节寄存器对应值

收到：1B 00 04 1D 00 00 01 86 6D 表明：收到当前 0 地址的字节寄存器对应值为 1

此时，左侧读到数值处显示“1”，表明读到指定的位寄存器对应值为 1。

- ✧ 选择“变量调试”——“写变量寄存器”

- ✧ 数值输入：0

- ✧ 其余保留默认值不变，单击“打开串口”

- ✧ 单击“发送命令”

此时，在通讯记录框中出现：

发送：1B 00 04 1D 00 00 01 86 6D 读取当前 0 地址的字节寄存器对应值

收到：1B 00 04 1D 00 00 01 86 6D 收到当前 0 地址的字节寄存器对应值为 1

发送：1B 00 04 1C 00 00 00 46 51 设置 0 地址的字节寄存器对应值为 0

这是因为位寄存器是字节寄存器中的一位，改变位寄存器对应值时不应该影响到其他位。而 Z4000 通信指令中最小的寄存器单元是字节寄存器，因此需要先将整个字节读出来，进行与或处理后，再将字节写入。

此时，画面 0 中的两个控件对应的位寄存器对应值为 0，Z4000 显示为如上右图“寄存器值为 0”所示状态。将写入数据框中的 0 改为 1，则将此位寄存器对应值置为 1，Z4000 显示为如上左图“寄存器值为 1”所示状态。

✧ 连续单点触摸 Z4000 中的位按钮

可以看到通讯记录中连续显示：“收到：1B 00 04 1D 00 00 01 86 6D 收到：1B 00 04 1D 00 00 00 47 AD”，表明在触摸位按钮后，Z4000 自动将位按钮对应寄存器的值发送出来，若在位按钮设置中选择“不返回串口”，则触摸后 Z4000 不会发送该命令。

✧ 选择“系统调试”——“写系统寄存器”，下拉选择“软复位”单击“发送命令”。此时，可以观察到 Z4000 进行了一次软复位。

同理，可以对 Z4000 其他寄存器进行在线调试。

注意：在工程中，控件对应寄存器有位寄存器、字节寄存器、字寄存器和双字寄存器之分，在通信指令中，一般指定的起始地址是字节地址。那么若需对某个寄存器进行操作，需先将其地址转为字节地址。

例如：

某控件对应位寄存器，地址为 0x3E，则对应字节寄存器地址为 0x07 (0x3E/8)

某控件对应字节寄存器，地址为 0x3E，则对应字节寄存器地址为 0x3E

某控件对应字寄存器，地址为 0x3E，则对应字节寄存器地址为 0x7C (0x3E*2)

某控件对应双字寄存器，地址为 0x3E，则对应字节寄存器地址为 0xF8 (0x3E*4)

1.6 编写程序

在线调试可以帮助用户熟悉操作命令、体验 Z4000 的显示效果，之后可以参考在线调试中给出的命令进行编程。

对 Z4000 的控制其实就是读写寄存器的操作，参考 Z4000 演示程序可以很容易的编写控制程序。

单击“编译”后，在工程所在的文件夹中可以自动生成两个文件“Record.h”和“Record.txt”。在文件中按照固定格式（详见文件中的说明）列出了工程中所有的变量寄存器和字符串寄存器，用户可以将文件添加到自己的程序中，方便对寄存器的操作和查看。

1.7 实际测试

✧ 在前面联机通信的基础上，去掉 USB 口的接线，将用户设备和 Z4000 通过串口相连。将用户通信接口的第 3、4 脚（RXD、TXD）或是第 5、6 脚（B、A）与用户设备的串口正确连接。

✧ 将用户设备和 Z4000 同时正确加电

✧ 观察、测试用户设备和 Z4000 是否工作正常

若出现通信失败或是出错时的解决办法如下：

- 1、查看用户设备和 Z4000 显控模组是否正确上电
- 2、查看用户设备和 Z4000 显控模组是否正确连接
- 3、查看 USB 口是否没有拔出
- 4、检查两边设定的波特率是否一致
- 5、检查用户设备发送的命令是否正确

2 演示工程讲解

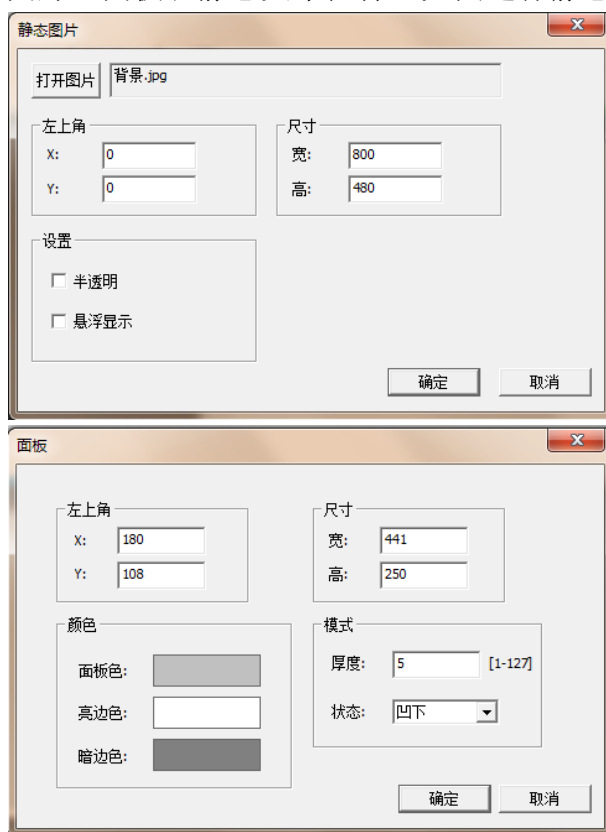
依照演示工程，逐一熟悉 Z4000 的各种控件。

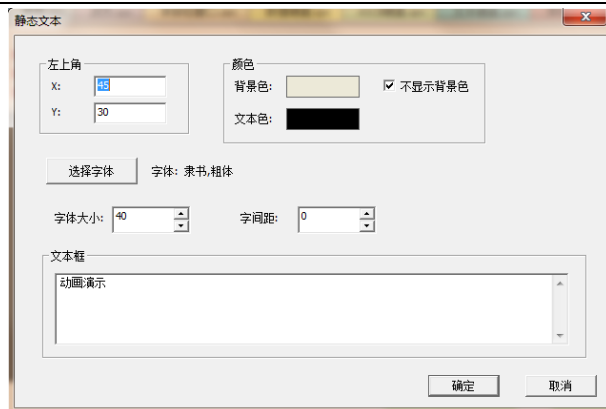
2.1 静态控件和生成 BMP 图片

实现功能：利用软件中的控件制作静态图片，方便用户作图。

参照演示工程的 BMP 生成画面。

- ✧ 新建画面，输入画面编号为 9，选择画面类型为绘图画面，画面名称改为“生成 BMP”
- ✧ 依次添加静态图片、面板和静态文本控件，如图进行静态控件的设置

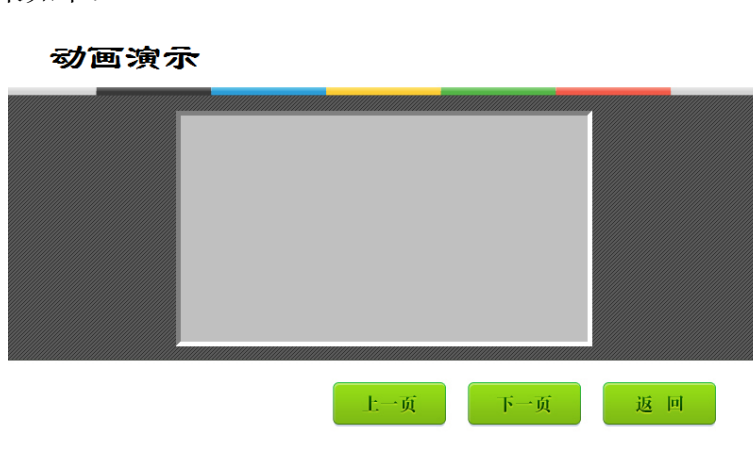




✧ 单击“生成 BMP”按钮，如图进行设置，生成 BMP 图片



最终画面效果如下：



2.2 动画演示

实现功能：图片自动切换，进行动画演示

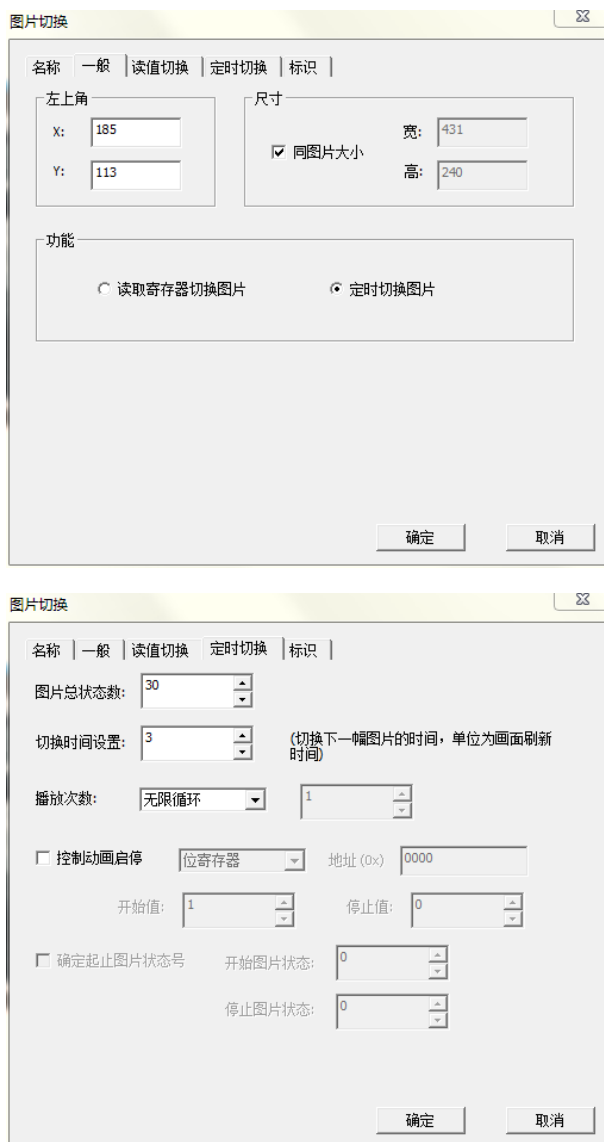
参照演示工程的动画演示画面。

✧ 新建普通画面，画面编号为 1，画面名称改为“动画演示”

✧ 添加静态图片作为画面的背景图，可选择在上一步中生成的 BMP 图片动画

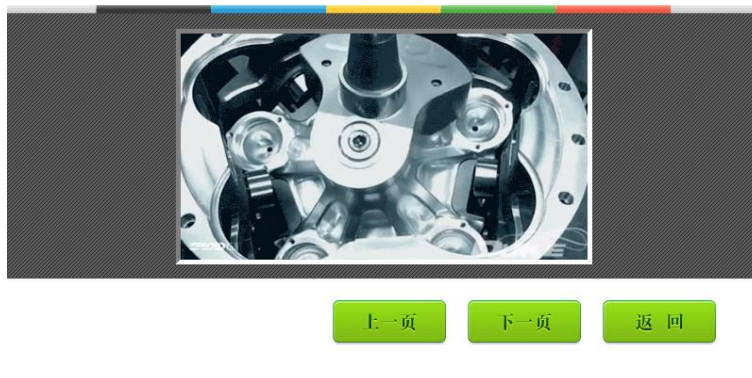
演示背景.jpg。

✧ 添加图片切换控件，如图进行图片切换控件的设置



✧ 从图片文件夹中选取图片 gif_D001.jpg~gif_D030.jpg
最终画面效果如下：

动画演示



小贴士:

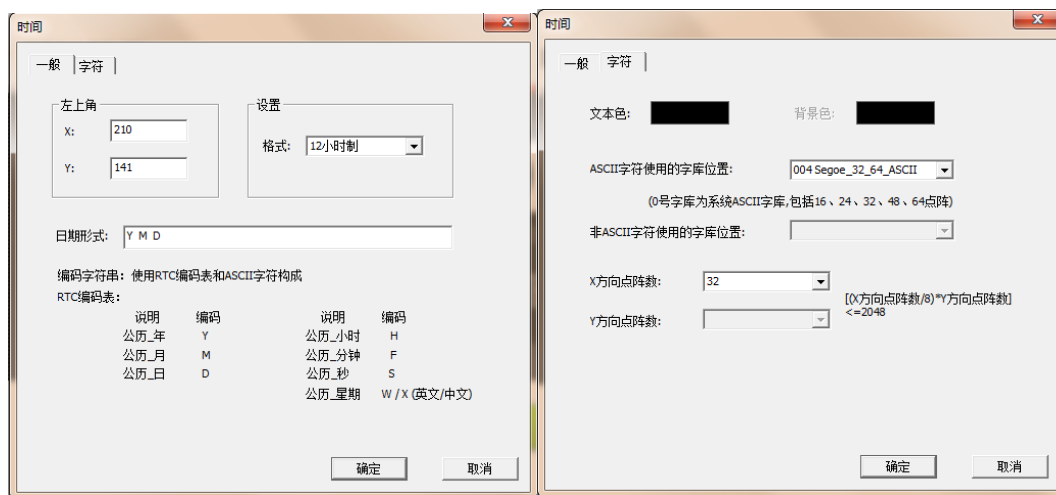
- 将一段视频或是 GIF 图片分解为多张图片，添加到图片切换控件中，可实现更好的显示效果。

2.3 时间显示

实现功能: 按照自定义的形式显示日期和时间

参照演示工程的时间显示画面。

- ✧ 新建普通画面，画面编号为 2，画面名称改为“时间显示”
- ✧ 添加静态图片 时间显示.jpg 作为画面的背景图
- ✧ 添加 3 个时间控件，如图进行时间控件的设置，可自定义日期形式





最终画面效果如下：



小贴士：

- 时间悬浮在画面上显示，不显示背景色。
- 为了配合背景图的显示效果，这里添加了 3 个时间控件，每个控件显示不同的时间内容。
- 其中 2 个时间控件使用了字库位置为 4 的用户字库，需先配置对应用户字库，

安徽世龙电子技术有限公司

www.worldlong.com

电话:0551-65329091、65329092、65329093 传真: 0551-65359928

否则无法使用该字库。用户字库的配置和使用方法详见[用户字库](#)。

2.4 弹出窗口

实现功能： 点击字按钮弹出控制窗口，点击“52”按键按钮数值变为 52，点击“43”按键按钮数值变为 43，点击“增加”字按钮数值增大，点击“减少”字按钮数值减小，点击“关闭窗口”功能按钮窗口关闭显示原画面。

参照演示工程的字按钮画面和字按钮窗口画面。

- ✧ 新建普通画面，画面编号为 3，画面名称改为“字按钮”
- ✧ 添加静态图片字按钮.jpg 作为画面的背景图
- ✧ 添加字按钮控件，如图进行字按钮控件的设置，其中弹出画面的编号设为 1001




- ✧ 新建弹出画面，画面编号为 1001，画面名称改为“字按钮窗口”

- ✧ 添加静态图片按钮窗口.jpg，输入左上角 X 坐标为 269、Y 坐标为 91，作为弹出窗口的背景图
- ✧ 添加数值控件，如图进行数值控件的设置

名称 | 显示 | 字符 | 输入 | 寄存器 | 颜色

左上角

X: 478
Y: 131

尺寸

☐ 自行设置
宽: 120
高: 48

显示设置

总位数: 5 小数位数: 0
显示数据类别: 十进制无符号 ☐ 前置零
控件对齐: 垂直居中水平居中
数值对齐: 右对齐

确定 取消

名称 | 显示 | 字符 | 输入 | 寄存器 | 颜色

字库位置: 000 系统ASCII
(0号字库为系统ASCII字库,包括16、24、32、48、64点阵)

编码: ASCII
字体高: 48
字体宽: 24

确定 取消

名称 | 显示 | 字符 | 输入 | 寄存器 | 颜色

☐ 具有录入功能 ☐ 数据返回串口
显示方式: 直接显示
弹出键盘画面 编号[0-1023]: 0
☐ 录入直接显示

确定 取消

名称 | 显示 | 字符 | 输入 | 寄存器 | 颜色

寄存器类型: 字符寄存器 地址: (0x) 0001
☒ 有超出限制 ☐ 超出显示实际值
变量下限: 0 变量上限: 65535

确定 取消

- ✧ 添加 2 个按键按钮,如图进行按键按钮控件的设置,键值分别设为 0x34、0x2B (即 52 和 43),分别使用图片按键 52.jpg、按键 52 按下.jpg和按键 43.jpg、按键 43 按下.jpg

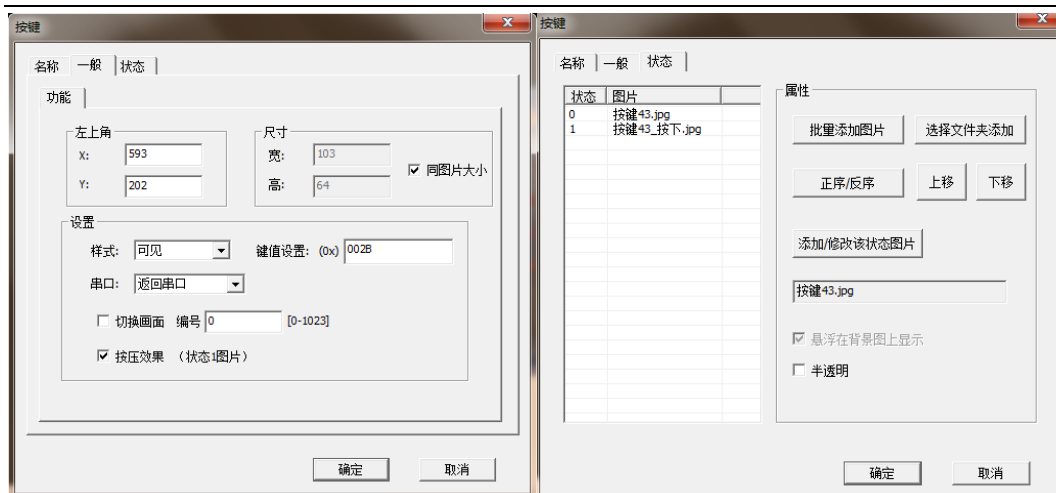
The figure consists of two side-by-side screenshots of a software configuration window titled '按键' (Button).

Left Screenshot (General Tab):

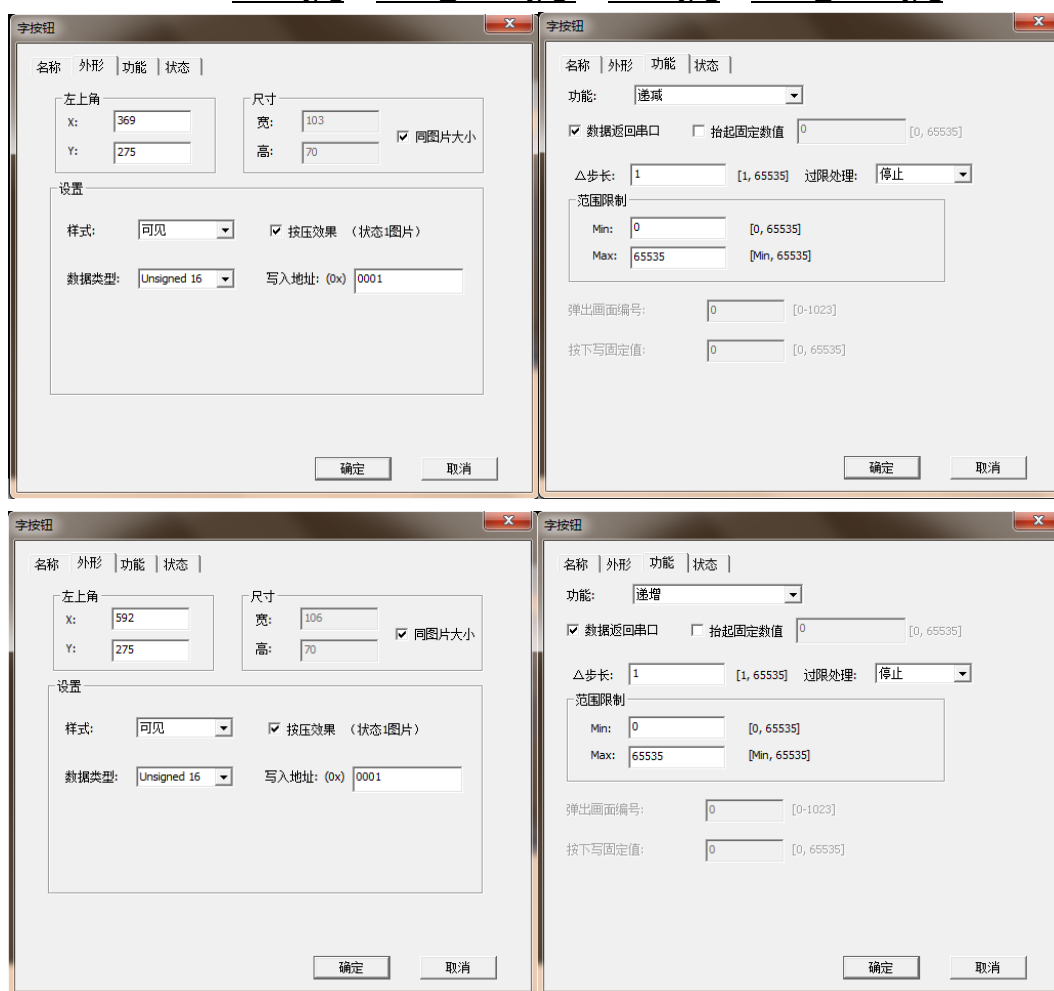
- 名称 (Name):** 一般 (General)
- 功能 (Function):** 无 (None)
- 左上角 (Top Left):** X: 370, Y: 202
- 尺寸 (Size):** 宽 (Width): 102, 高 (Height): 64
- ☒ 同图片大小 (Same as image size)
- 设置 (Settings):**
 - 样式 (Style): 可见 (Visible)
 - 键值设置: (0x) 0034
 - 串口 (Serial Port): 返回串口 (Return serial port)
 - ☐ 切换画面 (Switch screen) 编号 (Number): 0 [0-1023]
 - ☒ 按压效果 (Press effect) (状态: 图片) (State: Image)

Right Screenshot (Attributes Tab):

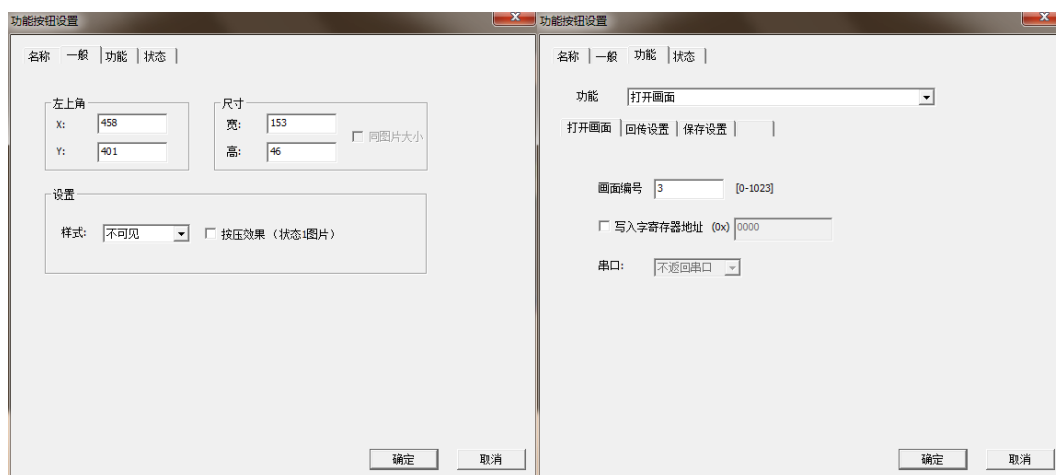
- 名称 (Name):** 一般 (General)
- 属性 (Attributes):**
 - 批量添加图片 (Batch add image)
 - 选择文件夹添加 (Select folder to add)
 - 正序/反序 (Normal/Reverse)
 - 上移 (Move up)
 - 下移 (Move down)
 - 添加/修改该状态图片 (Add/Modify this state image)
 - 按键_52.jpg
 - ☒ 悬浮在背景图上显示 (Display floating on background image)
 - ☐ 半透明 (Semi-transparent)



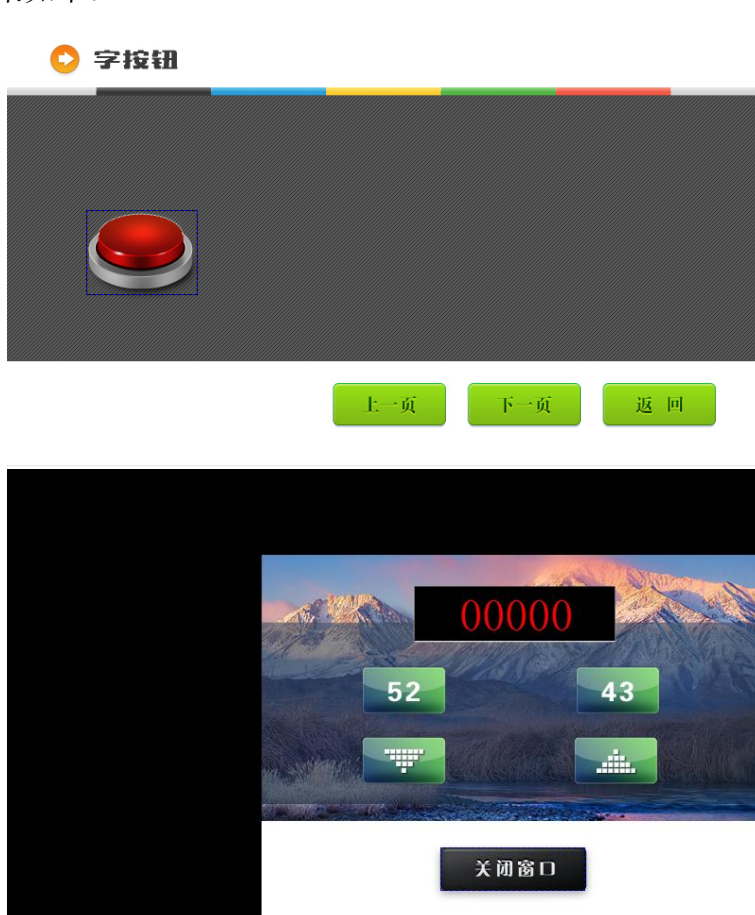
✧ 添加 2 个字按钮，如图进行字按钮控件的设置，功能分别设为递减和递增，分别使用图片递减.jpg、递减_按下.jpg和递增.jpg、递增_按下.jpg



✧ 添加功能按钮，如图进行功能按钮控件的设置，按钮不可见，按下后回到字按钮画面



最终画面效果如下：



小贴士：

- 弹出窗口的背景图片可以设为半透明，使弹出窗口的效果更佳丰富、逼真。
- 在弹出窗口中的控件不允许再次使用弹出窗口功能，即不允许二次弹出功能。
- 这里，点击 2 个按键按钮时，将按键的键值写入普通画面中字按钮对应的寄存器中。

- 这里，所有控件都对应同一寄存器地址，组合完成对寄存器值的显示与控制。

2.5 数值输入与显示

实现功能：数值的键盘输入与显示，同时结合进度条调节和指针显示。

参照演示工程的数值画面和数值键盘画面。

- ✧ 新建普通画面，画面编号为 4，画面名称改为“数值”
- ✧ 添加静态图片数值.jpg作为画面的背景图
- ✧ 添加数值控件，如图进行数值控件的设置，其中弹出键盘画面的编号设为 1002

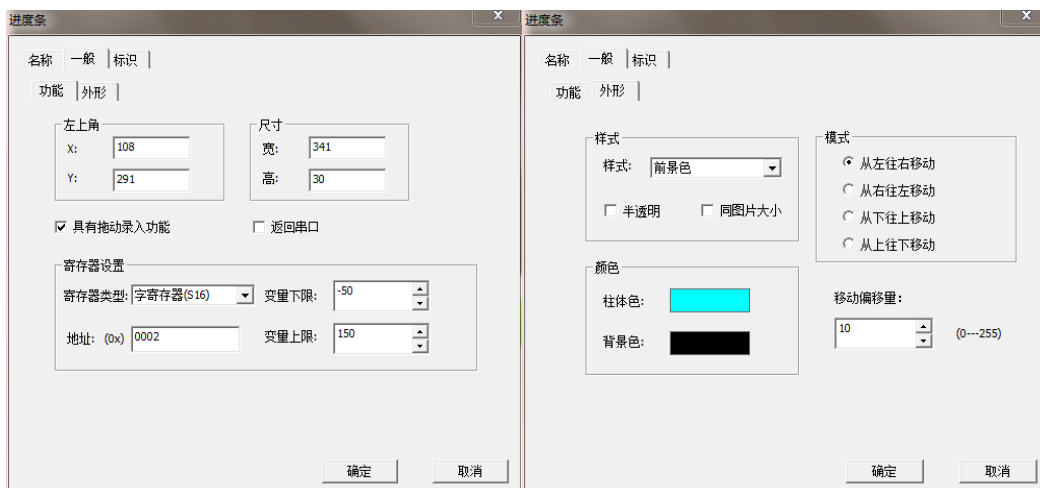


The figure shows four screenshots of the '数值' (Value) control configuration dialog box, arranged in a 2x2 grid. Each dialog has tabs for '名称' (Name), '显示' (Display), '字符' (Character), '输入' (Input), '寄存器' (Register), and '颜色' (Color).

- Top-left screenshot:** The '显示' (Display) tab is active. It shows '左上角' (Top-left) coordinates (X: 225, Y: 172) and '尺寸' (Size) (宽: 128, 高: 64). Under '显示设置' (Display settings), '总位数' (Total digits) is 3, '小数位数' (Decimal digits) is 0, '显示数据类型' (Display data type) is '十进制有符号(显示正)' (Signed decimal (display positive)), '前置零' (Leading zero) is checked, '控件对齐' (Control alignment) is '垂直居中水平居中' (Centered vertically and horizontally), and '数值对齐' (Value alignment) is '左对齐' (Left-aligned).
- Top-right screenshot:** The '字符' (Character) tab is active. It shows '字库位置' (Font location) as '000 系统ASCII' (000 System ASCII), '编码' (Encoding) as 'ASCII', '字体高' (Font height) as 64, and '字体宽' (Font width) as 32.
- Bottom-left screenshot:** The '输入' (Input) tab is active. It shows '具有录入功能' (Has input function) checked, '数据返回串口' (Data return serial port) unchecked, '显示方式' (Display mode) as '直接显示' (Direct display), '录入模式' (Input mode) as '重新输入' (Re-enter), '弹出键盘画面 编号[0-1023]:' (Pop up keyboard screen number [0-1023]:) as 1002, and '录入直接显示' (Input direct display) unchecked.
- Bottom-right screenshot:** The '寄存器' (Register) tab is active. It shows '寄存器类型' (Register type) as '字寄存器' (Word register), '地址: (0x)' (Address: (0x)) as 0002, '有超出限制' (Exceeds limit) checked, '超出显示实际值' (Exceeds actual value display) unchecked, '变量下限' (Variable lower limit) as -50, and '变量上限' (Variable upper limit) as 150.

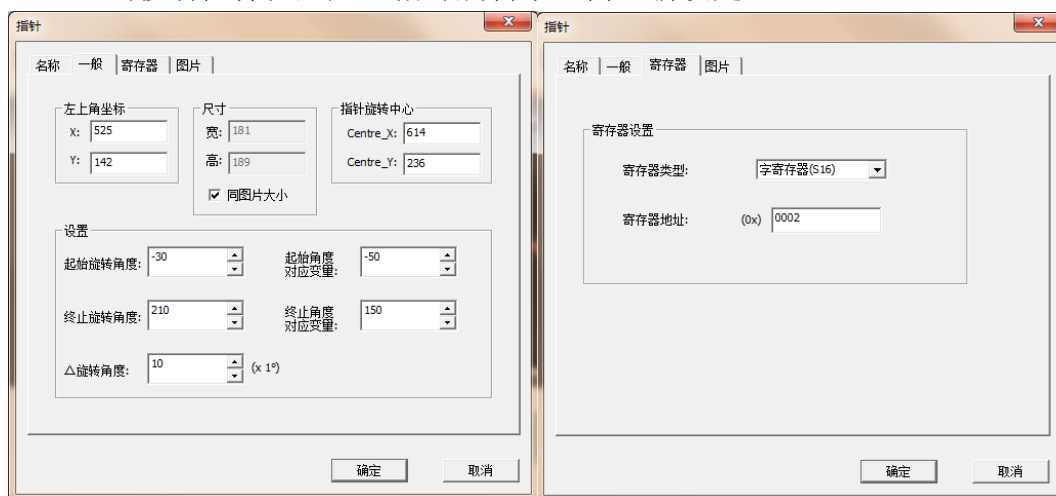


✧ 添加进度条控件，如图进行进度条控件的设置



✧ 添加指针控件，如图进行指针控件的设置，使用图片 指针.png

注意：指针旋转中心以当前指针图片的实际旋转为准，即需将鼠标放在指针图片的旋转中心上，此时软件左下角显示的坐标值为该指针控件的旋转中心。当移动控件位置后，指针旋转中心需重新设定。



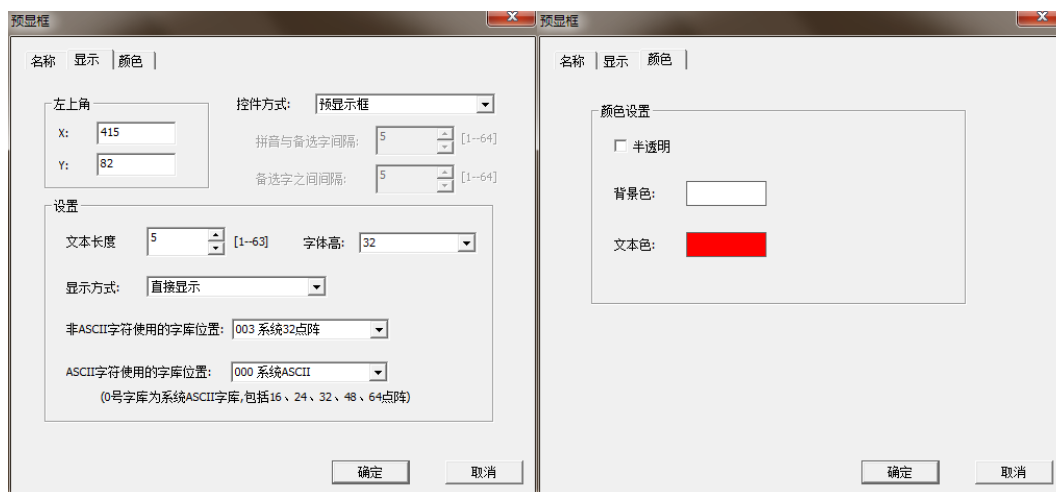
- ✧ 新建弹出画面，画面编号为 1002，画面名称改为“数值键盘”
- ✧ 添加静态图片数值键盘.jpg，输入左上角 X 坐标为 389、Y 坐标为 50，作为数值键盘的背景图
- ✧ 添加按钮控件，对数值键盘，有效键值是：

定义	键值
0~9	0x0030~0x0039
A~F（用于十六进制）	0x0041~0x0046
加号 +	0x002B
减号 -	0x002D
小数点 .	0x002E
清空	0x00F8
删除	0x00FD
确定	0x00FE
取消	0x00FF

这里共 17 个按钮，以按钮“1”和“Enter”为例，如图进行按钮控件的设置



- ✧ 添加预显框，如图进行预显框控件的设置



最终画面效果如下：



小贴士：

- 预显框设置中的文本长度只是方便制作工程时做参考，实际 Z4000 运行时，预显框的文本长度与原控件（这里指数值画面中的数值控件）的文本长度一致。
- 在使用 Z4000 快速入门演示工程配套的演示程序（“Z4000 快速入门演示程序”）时，为了突显演示效果，将此画面中的进度条控件对应的寄存器地址改为 0x0A 并勾选“返回串口”，其余不变。则连接演示板后，在拖动进度条时，演示程序将接收到的进度条数据发送到数值和指针对应的寄存器中，进而实现由进度条控制数值和指针控件的目的。

2.6 ASCII 文本输入与显示

实现功能：ASCII 文本的键盘输入与显示

参照演示工程的 ASCII 文本画面和 ASCII 键盘画面。

- ✧ 新建普通画面，画面编号为 5，画面名称改为“ASCII 文本”
- ✧ 添加静态图片 ASCII 文本.jpg 作为画面的背景图
- ✧ 添加 ASCII 文本控件，如图进行 ASCII 文本控件的设置，其中弹出键盘画面的

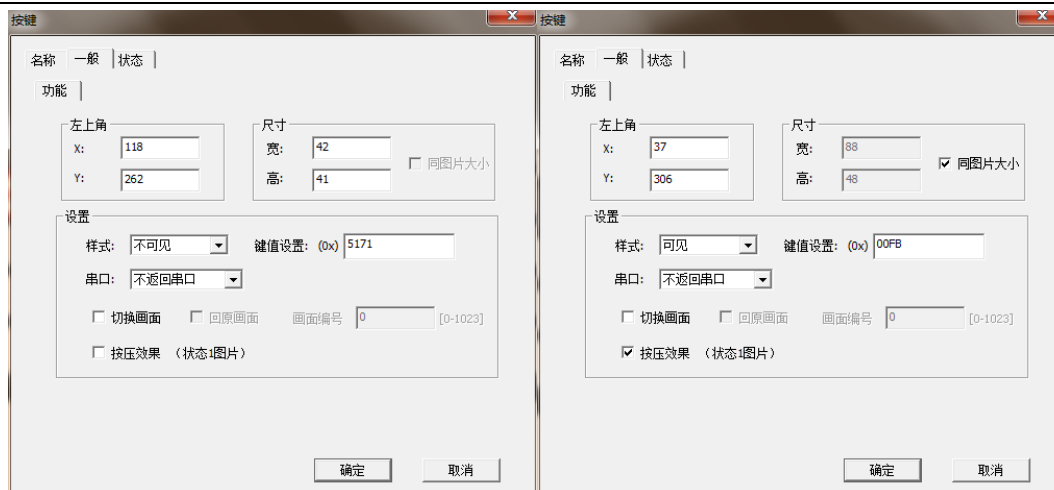
编号设为 1003，勾选“录入直接显示”



- ✧ 新建弹出画面，画面编号为 1003，画面名称改为“ASCII 键盘”
- ✧ 添加静态图片 ASCII 键盘.jpg，输入左上角 X 坐标为 26、Y 坐标为 205，作为 ASCII 键盘的背景图
- ✧ 添加按键控件，对 ASCII 文本键盘，有效键值是：

定义	键值
字符	0x0001~0x7F7F 当大写锁定时读取高字节，否则读取低字节。
清空	0x00F8
光标左移	0x00F9
光标右移	0x00FA
大写锁定	0x00FB
删除	0x00FD
确定	0x00FE
取消	0x00FF

以按键“Q”和“caps lock”为例，如图进行按键控件的设置



最终画面效果如下：



小贴士：

- 勾选“录入直接显示”后，在键盘输入过程中，输入的文本直接在原控件上显示；若不勾选“录入直接显示”，则需添加预显框控件，输入过程中的文本在预显框中显示。
- 在添加按键控件时，可灵活使用上位机软件中的对齐、等间距、等大小、全选、多选功能进行控件位置和大小调整。

- 典型 ASCII 文本键盘的键值定义如下表所示：

大写	非大写	键值	大写	非大写	键值
~	`	0x7E60	}	]	0x7D5D
!	1	0x2131		\	0x7C5C
@	2	0x4032	A	a	0x4161
#	3	0x2333	S	s	0x5373
\$	4	0x2434	D	d	0x4464
%	5	0x2535	F	f	0x4666
^	6	0x5E36	G	g	0x4767
&	7	0x2637	H	h	0x4868
*	8	0x2A38	J	j	0x4A6A
(	9	0x2839	K	k	0x4B6B
)	0	0x2930	L	l	0x4C6C
_	-	0x5F2D	:	;	0x3A3B
+	=	0x2B3D	"	'	0x2227
Q	q	0x5171	Z	z	0x5A7A
W	w	0x5777	X	x	0x5878
E	e	0x4565	C	c	0x4363
R	r	0x5272	V	v	0x5676
T	t	0x5474	B	b	0x4262
Y	y	0x5979	N	n	0x4E6E
U	u	0x5575	M	m	0x4D6D
I	i	0x4969	<	,	0x3C2C
O	o	0x4F6F	>	.	0x3E2E
P	p	0x5070	?	/	0x3F2F
{	[	0x7B5B	空格	空格	0x2020

2.7 动态文本输入与显示

实现功能： 动态文本的键盘输入与显示

参照演示工程的动态文本画面和文本键盘画面。

- ✧ 新建普通画面，画面编号为 6，画面名称改为“动态文本”
- ✧ 添加静态图片 动态文本.jpg 作为画面的背景图
- ✧ 添加动态文本控件，如图进行动态文本控件的设置，其中弹出键盘画面的编号设为 1004

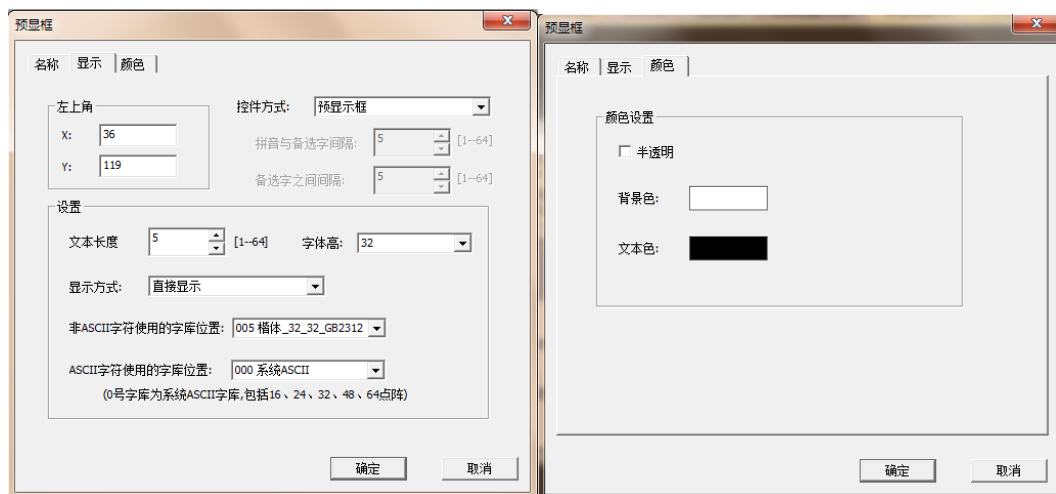


- ✧ 新建弹出画面，画面编号为 1004，画面名称改为“文本键盘”
- ✧ 添加静态图片文本键盘.jpg，输入左上角 X 坐标为 25、Y 坐标为 92，作为动态文本键盘的背景图
- ✧ 添加按钮控件，对动态文本键盘，有效键值是：

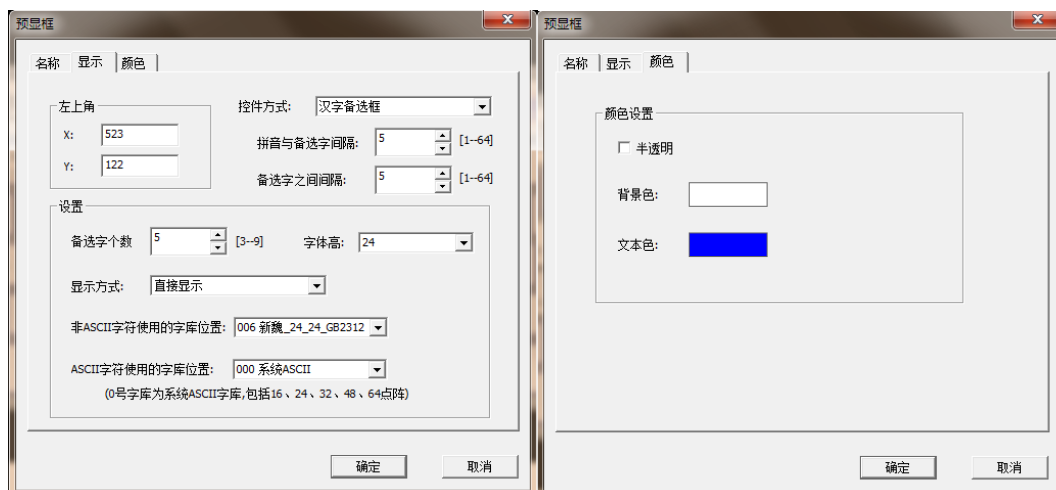
定义	键值
字符	0x0001~0x7F7F, 0xA0A0~0xF7FF 当键值在 0x0001~0x7F7F 范围内时：当大写锁定时读取高字节，否则读取低字节。 当键值在 0xA0A0~0xF7FF 范围内且在中文输入状态下时：读取 2 个字节。显示对应全角字符。
清空	0x00F8
光标左移	0x00F9
光标右移	0x00FA
大写锁定	0x00FB
中英切换	0x00FC
删除	0x00FD
确定	0x00FE
取消	0x00FF

按键控件的设置可以参考数值键盘画面和 ASCII 键盘画面中的按键设置。

✧ 添加预显框，如图进行预显框控件的设置，选择控件方式为预显示框



✧ 添加预显框，如图进行预显框控件的设置，选择控件方式为汉字备选框



最终画面效果如下：





小贴士:

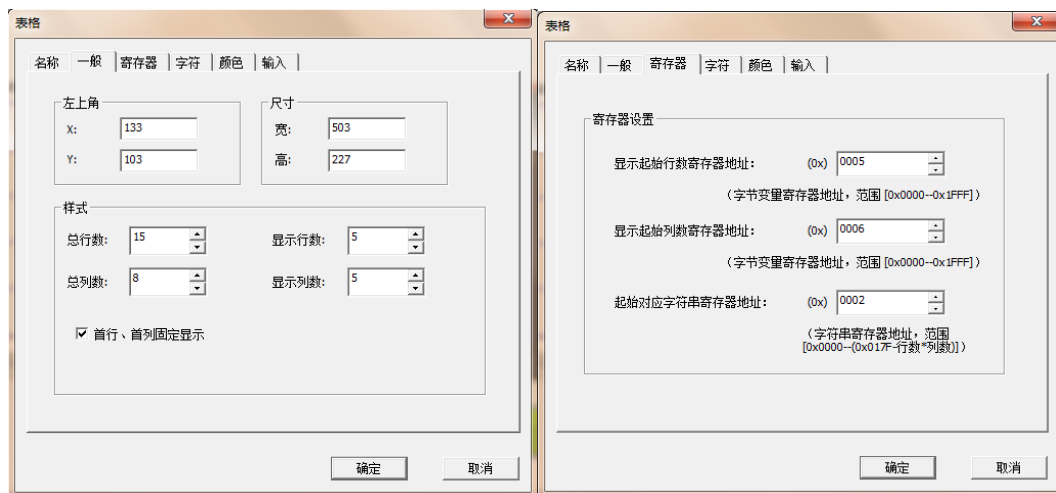
- 使用中文拼音输入法时，需要添加汉字备选框来显示拼音对应的备选汉字。
- 当键值设为 0x00FB 或 0x00FC，即大写锁定按键或中英切换按键时，建议勾选“按压效果”并添加两张图片，方便使用时查看当前输入状态。
- 编辑文本键盘画面时，可使用上位机软件中的画面复制功能，复制 ASCII 键盘画面后再进行修改，可避免重复添加按键控件。
- 2 个预显框控件使用了字库位置为 5 和 6 的用户字库，需先配置对应用户字库，否则无法使用该字库。用户字库的配置和使用方法详见[用户字库](#)。

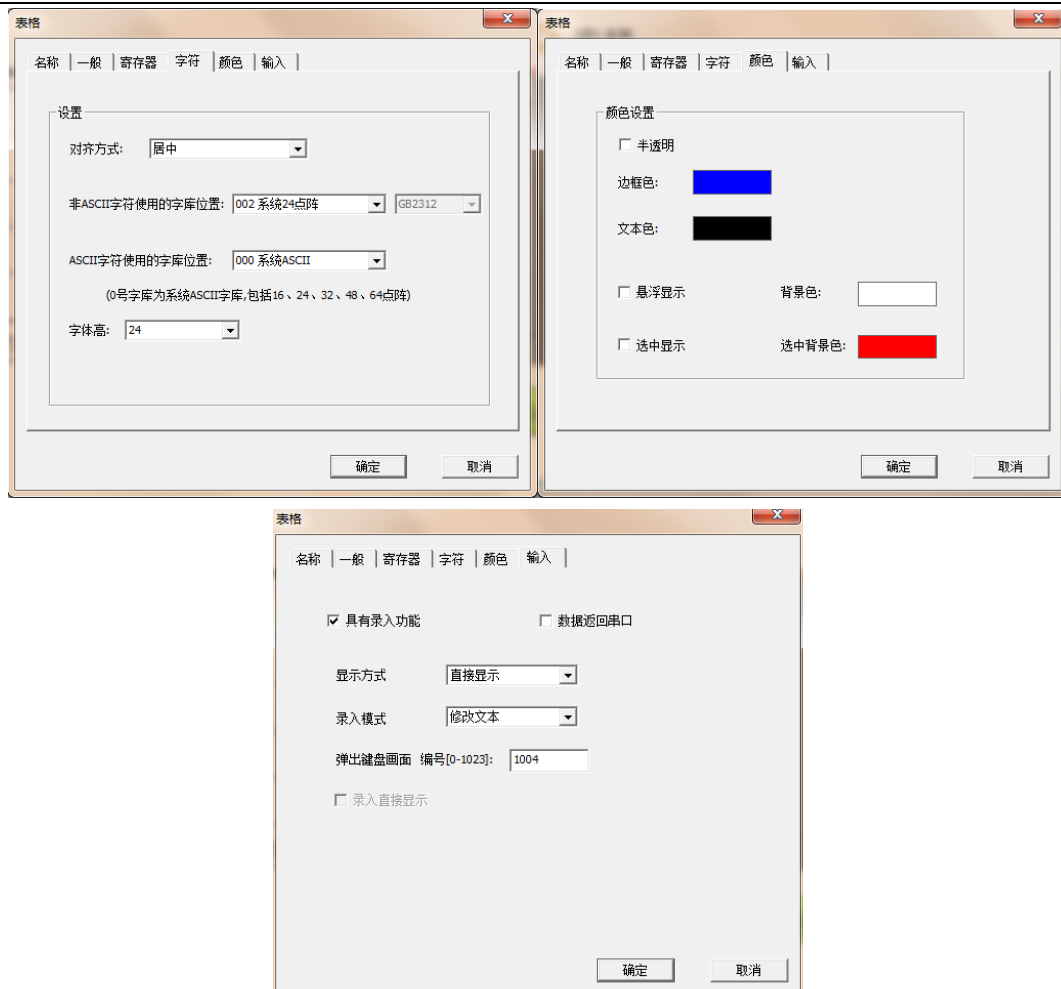
2.8 表格

实现功能: 以表格的形式组合显示字符串寄存器中的信息，结合进度条实现表格上下、左右移动显示的目的。

参照演示工程的表格画面和表格键盘画面。

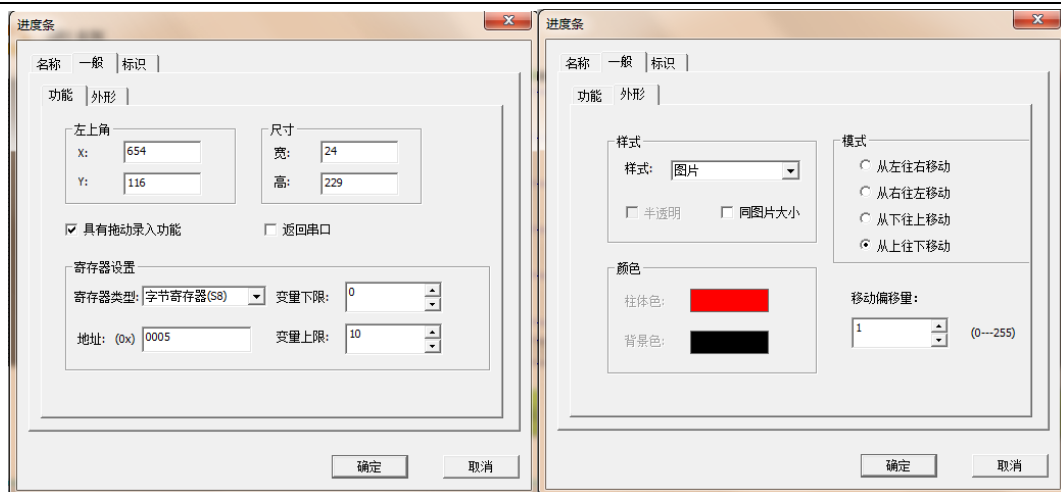
- ✧ 新建普通画面，画面编号为 7，画面名称改为“表格”
- ✧ 添加静态图片表格.jpg 作为画面的背景图
- ✧ 添加表格控件，如图进行表格控件的设置，其中弹出画面的编号设为 1004，即使用文本键盘画面





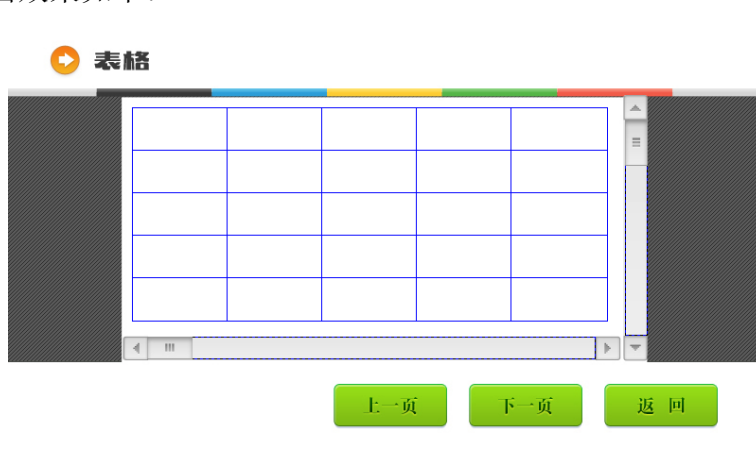
✧ 添加进度条控件，如图进行进度条控件的设置，使用图片横向进度.jpg 和竖向进度.jpg





注意：两个进度条控件的寄存器地址分别与表格控件显示起始行数、列数寄存器地址相同。

- ✧ 表格控件与动态文本控件使用同一键盘画面，在键盘中点击“Enter”和“Cancel”时，返回原画面。
- ✧ 最终画面效果如下：



小贴士：

- 表格控件可以将多个字符串组合显示，使用时要特别注意寄存器地址的设置。
- 在使用 Z4000 快速入门演示工程配套的演示程序（“Z4000 快速入门演示程序”）并连接演示板时，通过软键盘改变此画面表格控件中地址为 0x000B 的单元格内容后，演示程序接收到该单元格对应字符串寄存器的值，并将其中第一个字符的 ASCII 码值以递增的方式发送到后两个单元格对应的字符串寄存器中。

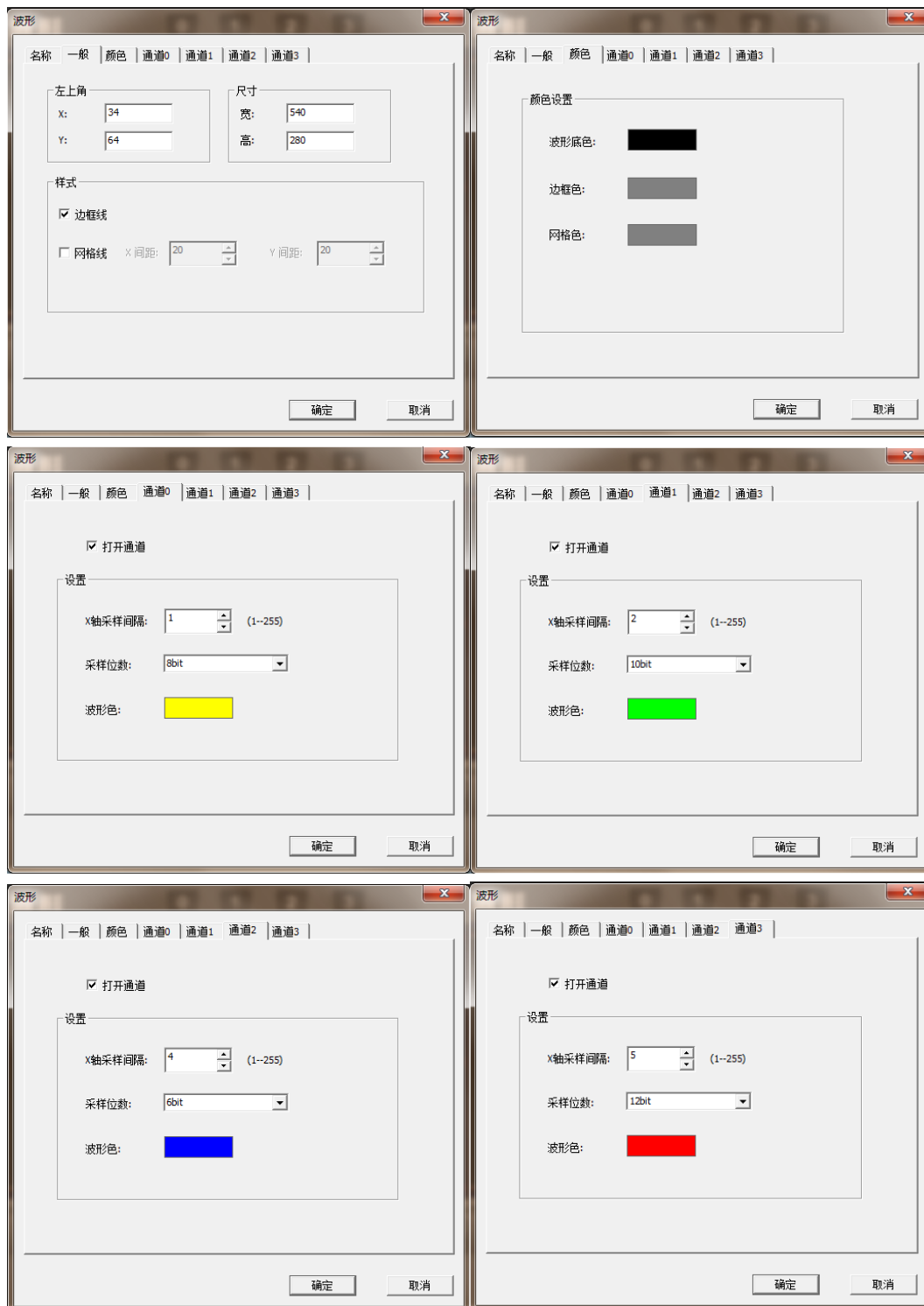
2.9 波形

实现功能：以波形的形式快速显示用户发送的实时数据，使用进度条、字按钮、

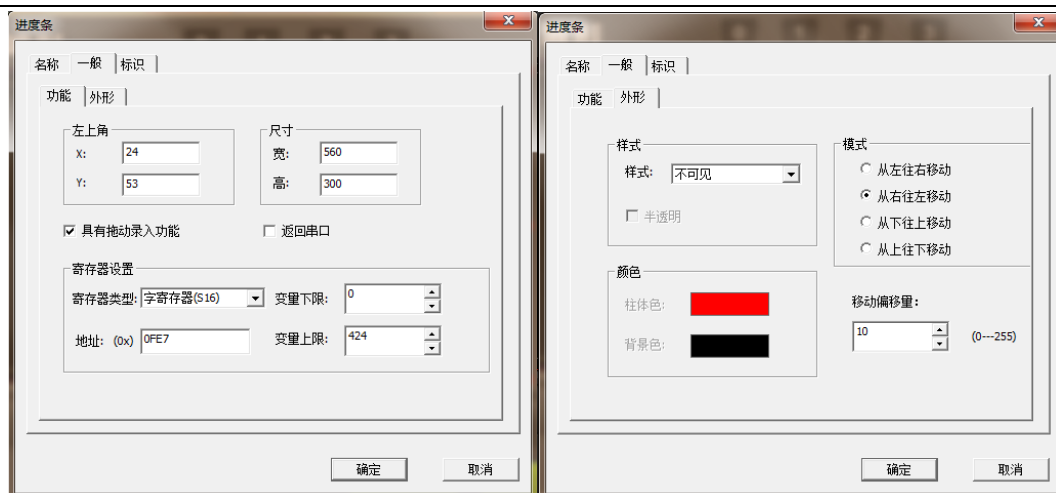
数值控件对波形进行控制。

参照演示工程的波形画面。

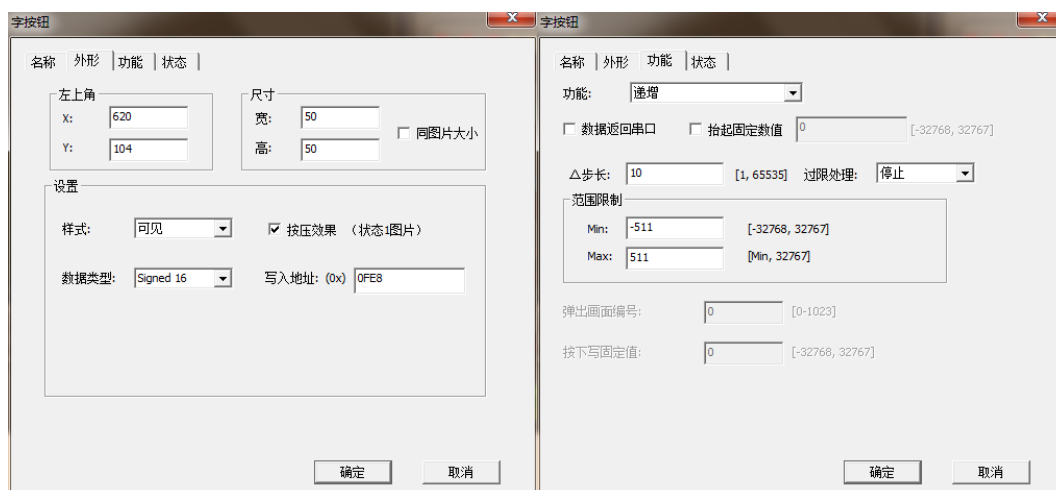
- ✧ 新建普通画面，画面编号为 8，画面名称改为“波形”
- ✧ 添加静态图片波形.jpg作为画面的背景图
- ✧ 添加波形控件，如图进行波形控件的设置



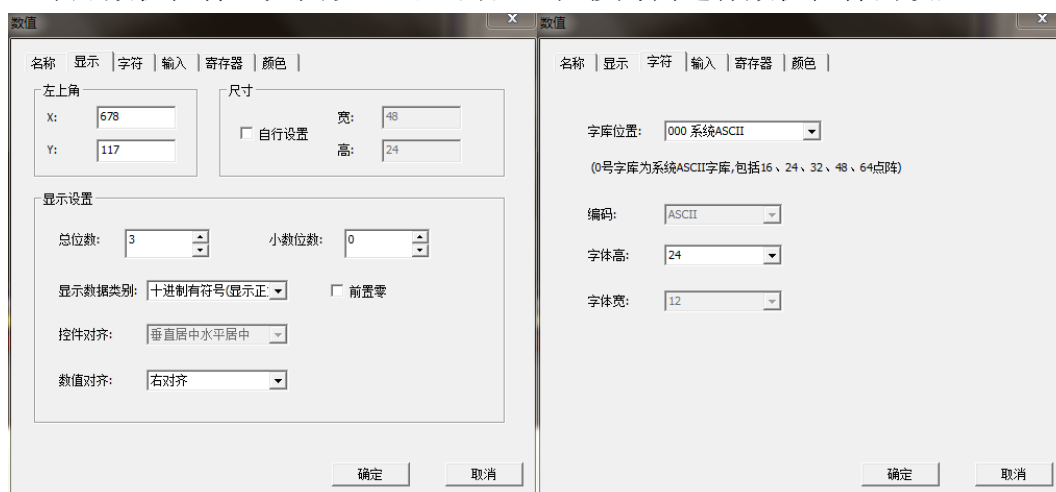
- ✧ 添加进度条控件，如图进行进度条控件的设置

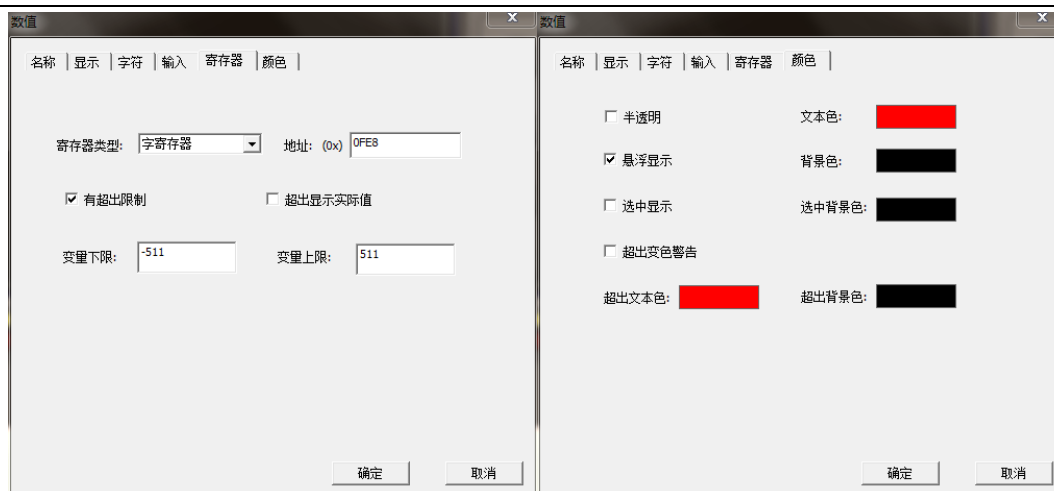


- ✧ 添加字按钮控件，用作波形水平位移、垂直位移和垂直缩放。如图以通道 0 垂直位移的左按钮为例进行字按钮控件的设置



- ✧ 添加数值控件，如图以通道 0 的垂直位移为例进行数值控件的设置





最终画面效果如下：



小贴士：

- 波形控件对应的寄存器在变量寄存器中占固定的地址，可在波形寄存器配置中修改初始值，要注意寄存器类型，区分字节地址和字地址。

2.10 二维码

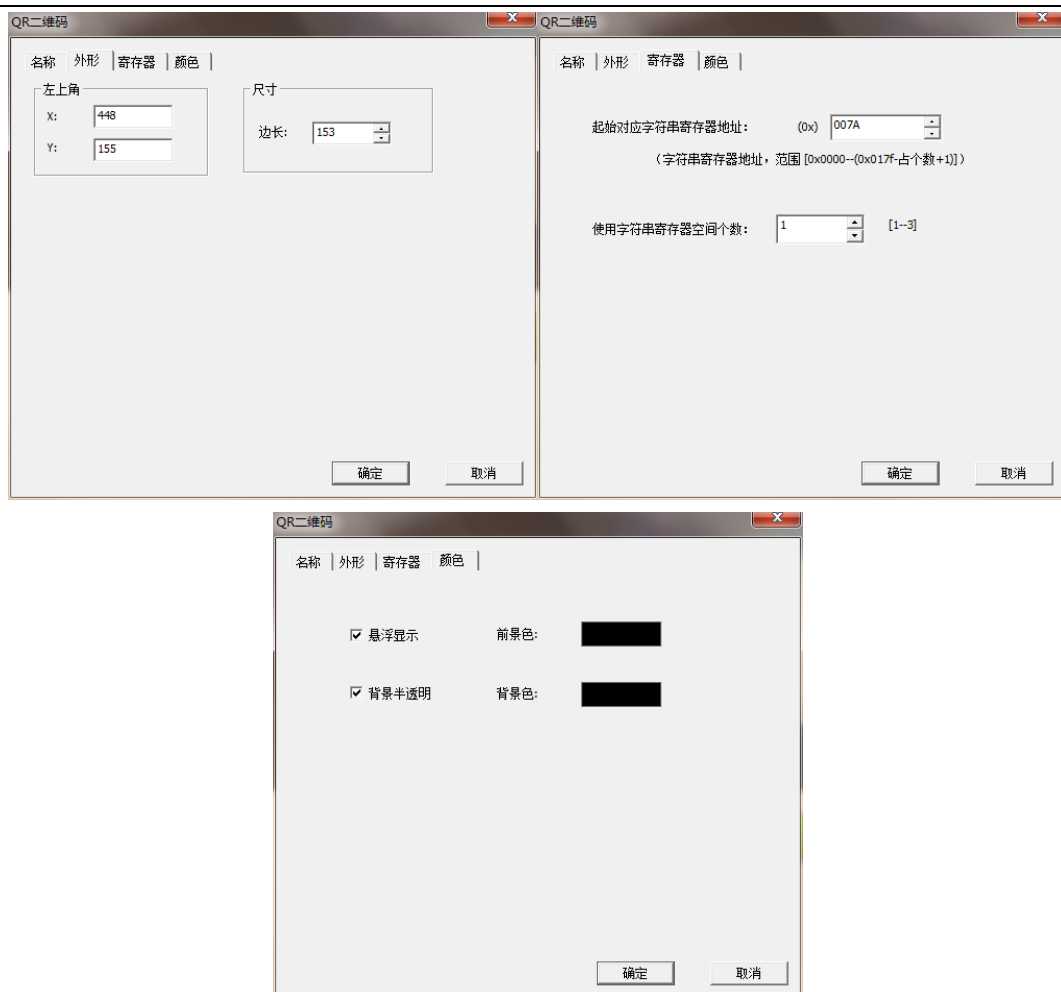
实现功能：根据字符串内容显示二维码图形

参照演示工程的二维码画面。

✧ 新建普通画面，画面编号为 10，画面名称改为“二维码”

✧ 添加静态图片 二维码.jpg 作为画面的背景图

添加二维码控件，如图进行二维码控件的设置，起始对应字符串地址为 0x7A。



最终画面效果如下:



小贴士:

- 可在字符串寄存器配置中设置二维码对应字符串的内容，也可通过发命令的方式改变二维码内容。
- 通过勾选“悬浮”和“背景半透明”，可露出背景图案，丰富二维码的显示

安徽世龙电子科技有限公司

www.worldlong.com

电话:0551-65329091、65329092、65329093 传真: 0551-65359928

效果。

2.11 画面切换

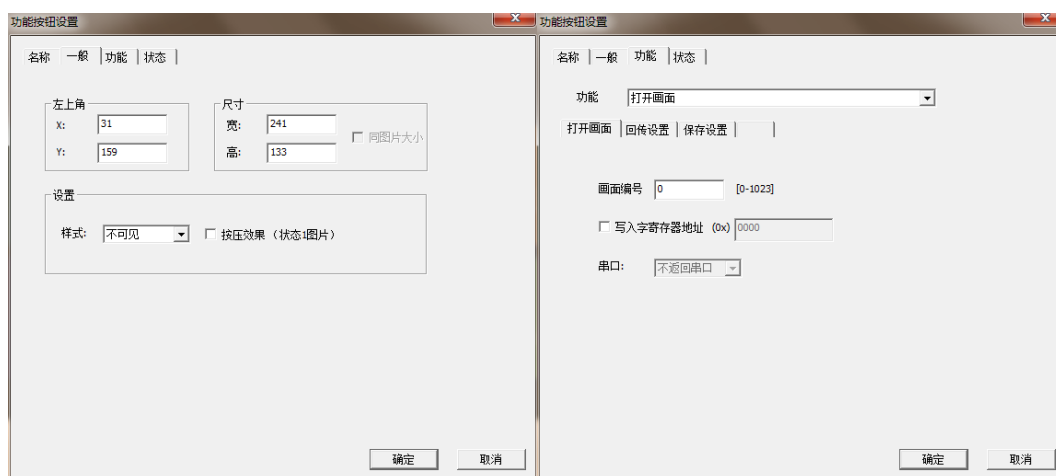
实现功能：在各个画面间进行切换。在开机画面中点击按钮切换到对应画面，在各个普通画面中可进行返回开机画面、进入上一页和下一页的画面切换。

参照演示工程的主画面和其他普通画面。

✧ 新建普通画面，画面编号为 9，画面名称改为“主画面”

✧ 添加静态图片 **开机画面.jpg** 作为画面的背景图

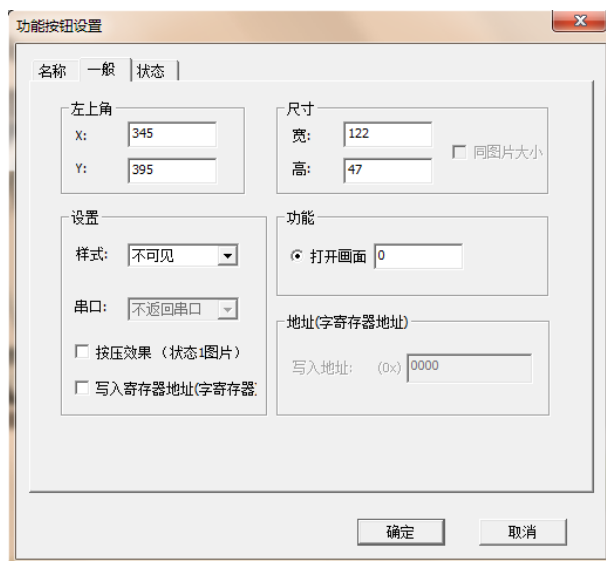
依次添加 9 个功能按钮控件，如图进行功能按钮控件的设置，其中打开画面的编号分别设为 0、1、2、3、4、5、6、7、8 和 10。



最终画面效果如下：



✧ 在其他 10 个普通画面中添加功能按钮，以动画演示画面为例，依次添加 3 个功能按钮控件，如图进行功能按钮控件的设置，其中打开画面的编号分别设为 0、2 和 9。



小贴士：

- 可在工程中设置启动画面为“0009：主画面”，则开机显示该画面。
- 在设置功能按钮时选择样式为“可见”、勾选“按压效果”并添加两张图片，则在点击按钮时，会先显示第二张图片再切换画面，即具有按压按钮的显示效果。
- 在使用 Z4000 快速入门演示工程配套的演示程序（“Z4000 快速入门演示程序”）时，为了突显演示效果，将工程中的三个功能按钮控件（分别是主画面中的“波形”、表格画面中的“下一页”和控件示例中的“上一页”）设置为“返回串口”、“写入寄存器地址”，并将写入的地址设为 0x000B，其余不变。则连接演示板后，通过点击功能按钮进入波形画面时，演示程序将发送波形数据。

3 辅助功能

为了丰富产品功能、优化显示效果、提升使用感受，Z4000 还具备以下辅助功能。

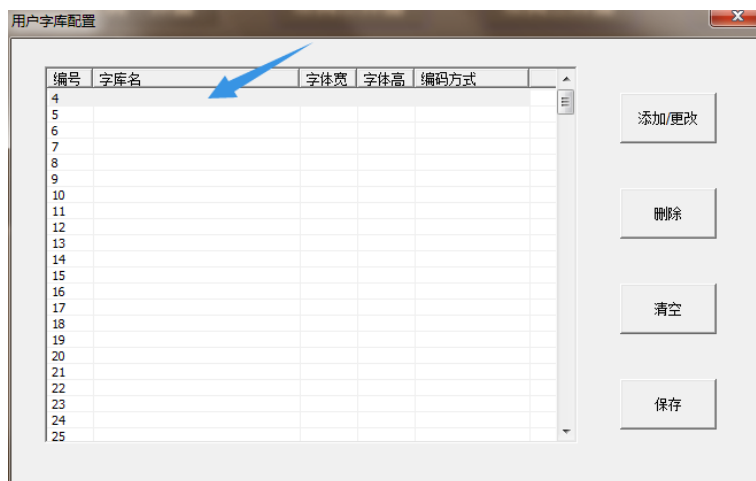
3.1 用户字库

Z4000 自带 8 种系统字库：0 号字库：包含 16、24、32、48、64 点阵的宋体 ASCII 码字库；1 号字库：16 点阵的宋体 GB2312 汉字库；2 号字库：24 点阵的宋体 GB2312 汉字库；3 号字库：32 点阵的宋体 GB2312 汉字库。用户若需使用其他字库，可以自行下载。

✧ 使用开发包中的 Z4000 点阵字库生成器 Z4000Fonts 生成需要的字库文件



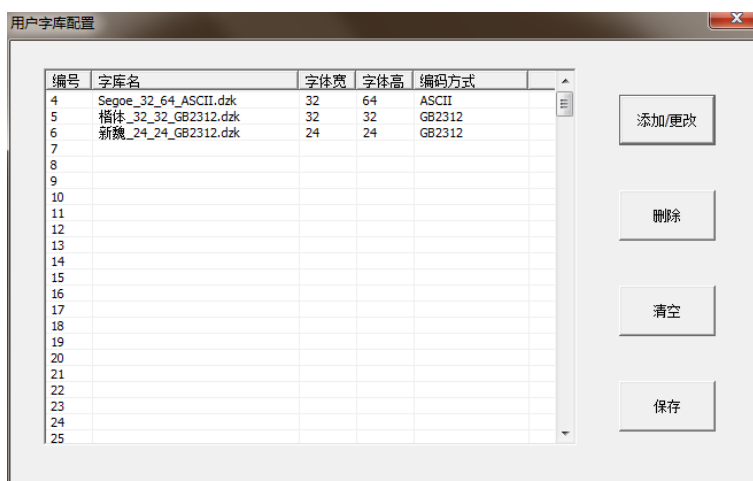
✧ 单击“设置”——“用户字库配置”，弹出配置窗口



- ✧ 在需要配置的字库编号处单击（如需要添加或修改 4 号字库，则单击第 1 行）
- ✧ 单击“添加/更改”，弹出添加用户字库窗口



- ✧ 选择要添加字库的参数，单击“确定”
- ✧ 单击“保存”则保存用户信息成功；单击“删除”或“清空”可以修改信息以快速入门演示工程为例，配置好的用户字库如下图所示：



- ✧ 单击“下载”——“工程下载”或是单击功能栏中的“下载”按钮，弹出下载窗口
- ✧ 单击“下位机已有字库”可以查看当前与电脑相连的 Z4000 中存储的用户字库的信息



✧ 勾选“下载工程用户字库”，即可将之前添加的用户字库下载到 Z4000 中

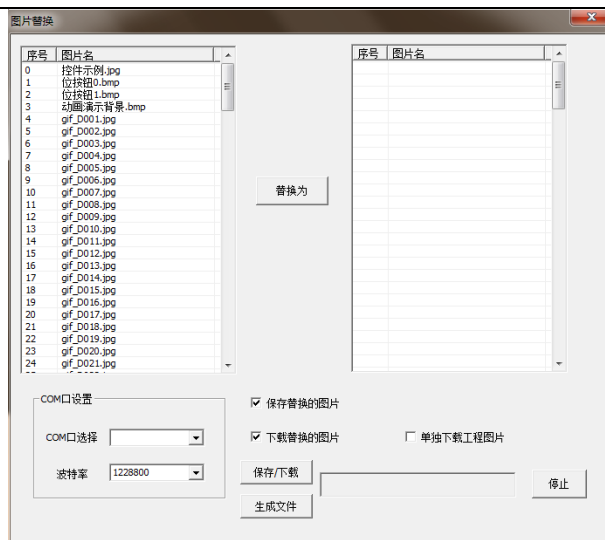
小贴士：

- 在添加和使用用户字库时，要格外注意字体宽、高、编码方式和字库号的准确性和一致性，否则可能会显示出错。
- Z4000 同时支持数值控件的艺术字显示，具体的使用方法详见 Z4000A Z4000M 用户使用手册。

3.2 图片替换

使用图片替换功能可以替换工程图片中单张或是某几张图片，节省用户下载图片的时间。当替换的图片与原图尺寸不一致时，Z4000 会将替换的图片调整到原图的大小。

✧ 单击“下载”——“图片替换”，弹出图片替换窗口



- ✧ 左侧框中为当前工程使用的图片名，单击需要替换的图片名
- ✧ 单击“替换为”，选择新图片
- ✧ COM 口的设置与在线调试、下载窗口一致
- ✧ 保存替换的图片：将当前新替换的图片保存进工程
- ✧ 下载替换的图片：将当前新替换的图片下载进屏中进行查看
- ✧ 单独下载工程图片：下载左边框中选中的图片
- ✧ 保存/下载：根据勾选情况保存或下载
- ✧ 生成文件：根据勾选情况生成.zwl 下载文件

3.3 画面复制

当两个图片相同或大致相同时，可以采用画面复制的方式节省制作时间。画面复制分为工程内复制和跨工程复制。

工程内复制：在工程管理视图中右击需要复制的画面号，选择“复制”，在弹出的新画面中设置新画面名称，单击“确定”即可生成画面。在新画面的画面设置中可以修改画面编号。

跨工程复制：将原工程所在文件夹中的画面文件（.zpc）复制到新工程所在文件夹中，在新工程中打开该画面，单击工具栏中的“添加到工程”按钮，将新画面添加到新工程中。注意：新添加的画面必须与新工程中已有的画面编号、名称均不一致，否则会出错。