AT7456 功能演示程序说明

1 概述:

1.1 程序用途:

本程序配合 AT7456 演示板使用. 读写芯片寄存器,配置字库和屏幕字符.

1.2 硬件要求:

普通 PC 机,要求有串口或 USB 串口.

1.3 系统要求:

要求 windows xp 或以上版本,安装. NetFramework2. 2 版本.

1.4 安装说明:

1) 安装. Net2. 2 框架.

微软官方下载地址:

http://www.microsoft.com/en-us/download/details.aspx?id=1639

下载对应系统的框架包,并按程序指引安装.

如果已经安装过. Net 框架 2. 2 版本,可以略过此步骤.

2) 运行 AT7456 芯片配置&字库工具. exe

1.5 数据说明:

1) 寄存器:

AT7456 芯片的寄存器,长度为 8 位. 本程序中 D7 表示最高位,D0 表示最低位.

2) 字库:

字库中包含 256(AT7456)或 512(AT7456E)个字符,每个字符由 12x18 个点阵构成,每个点有黑白灰三种颜色. 包含字库信息的文件为. mcm 文件.

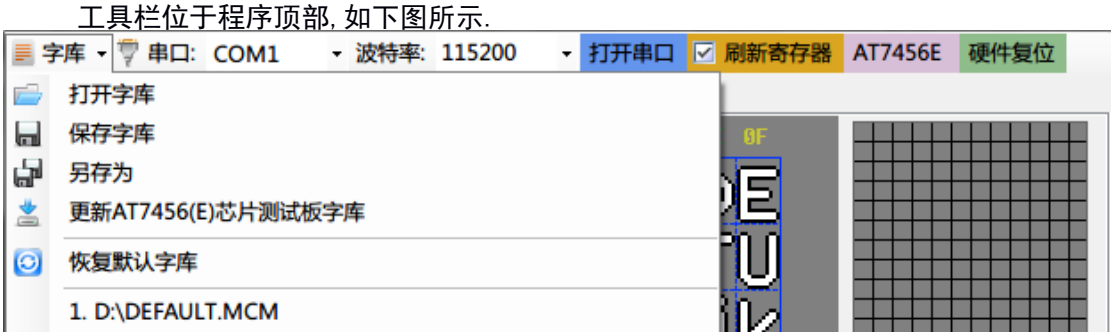
3) 屏幕:

屏幕显示 30x16 个字符,每个字符由 12x18 个点阵构成,每个点有黑白灰三种颜色. 另外字符还具有背景色,是否闪烁,是否反色的显示风格. 包含屏幕信息的文件为. mdm 文件.

2 程序界面说明:

2.1 工具栏说明:

工具栏位于程序顶部,如下图所示.



1) 打开字库: 打开含有字库信息的 mcm 文件.
2) 保存字库: 将编辑好的字库保存成 mcm 文件.
3) 另存为: 将编辑好的字库另存为选定的路径的文件.
4) 更新 AT7456(E)芯片测试板字库: 将字库烧录到生产测试板,CF1 开关拨到 ON 位置.
5) 恢复默认字库: 使用程序自带的默认字库覆盖当前界面上的字库.
6) 串口号: 选择串口号.
7) 波特率: 选择波特率.
8) 打开/关闭串口: 打开或关闭串口.
9) AT7456/AT7456(E): 切换不同芯片版本的字库集.
10) 刷新寄存器: 选中此项,将在串口打开时读所有寄存器的值.
11) 硬件复位: 对硬件进行复位.

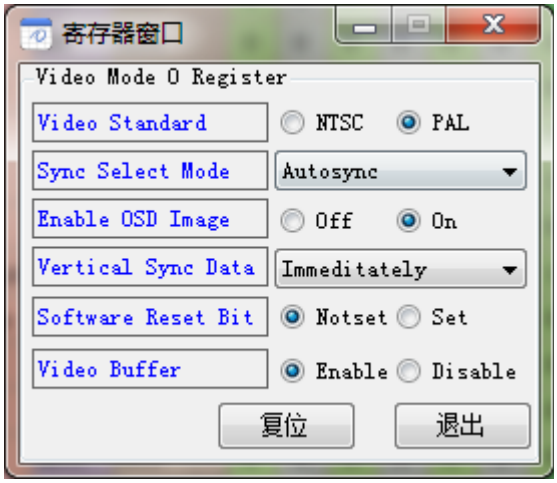
2.2 寄存器控制页:

本页列出所有寄存器,进行读写操作. 如下图所示:

寄存器名	简称	地址	D7	D6	D5	D4	D3	D2	D1	D0	寄存器值	读	写
Video Mode 0	VM0	0x00	0	0	0	0	0	0	0	0	0x00	读	写
Video Mode 1	VM1	0x01	0	1	0	0	0	1	1	1	0x47	读	写
Horizontal Offset	HOS	0x02	0	0	1	0	0	0	0	0	0x20	读	写
Vertical Offset	VOS	0x03	0	0	0	1	0	0	0	0	0x10	读	写
Display Memory Mode	DMM	0x04	0	0	0	0	0	0	0	0	0x00	读	写
Display Memory Address High	DMAH	0x05	0	0	0	0	0	0	0	0	0x00	读	写
Display Memory Address Low	DMAL	0x06	0	0	0	0	0	0	0	0	0x00	读	写
Display Memory Data In	DMDI	0x07	0	0	0	0	0	0	0	0	0x00	读	写
Character Memory Mode	CMM	0x08	0	0	0	0	0	0	0	0	0x00	读	写
Character Memory Address High	CMAH	0x09	0	0	0	0	0	0	0	0	0x00	读	写
Character Memory Address Low	CMAL	0x0A	0	0	0	0	0	0	0	0	0x00	读	写
Character Memory Data In	CMDI	0x0B	0	0	0	0	0	0	0	0	0x00	读	写
OSD Insertion Mux	OSDM	0x0C	0	0	0	1	1	0	1	1	0x1b	读	写
Row 0 Brightness	RB0	0x10	0	0	0	0	0	0	0	1	0x01	读	写
OSD Black Level	OSDBL	0x6C	0	0	0	1	0	0	0	0	0x10	读	写
Status	STAT	0xA0	0	0	0	0	0	0	0	0	0x00	读	N/A
Display Memory Data Out	DMDO	0xB0	0	0	0	0	0	0	0	0	0x00	读	N/A
Character Memory Data Out	CMDO	0xC0	0	0	0	0	0	0	0	0	0x00	读	N/A

*注：Row n Birghtness 寄存器有 16 个，根据地址下拉框的选择而切换.

1） 寄存器名：寄存器名称，双击单元格，弹出寄存器编辑窗口. 如下图所示：



*图例为 Video Mode 0 寄存器窗口，改动将即时更新到下位机.

*复位按钮：将寄存器恢复默认值.

2） 简称：寄存器缩写名.

3） 地址：寄存器地址，以十六进制显示.

4） D7-D0：寄存器的 bit 值，从高到低. 双击单元格可以在 0 和 1 之间切换.

5） 寄存器值：寄存器的 byte 值，以十六进制表示. 此单元格可编辑.

6） 读：从下位机读寄存器值.

7） 写：将寄存器值写入下位机.

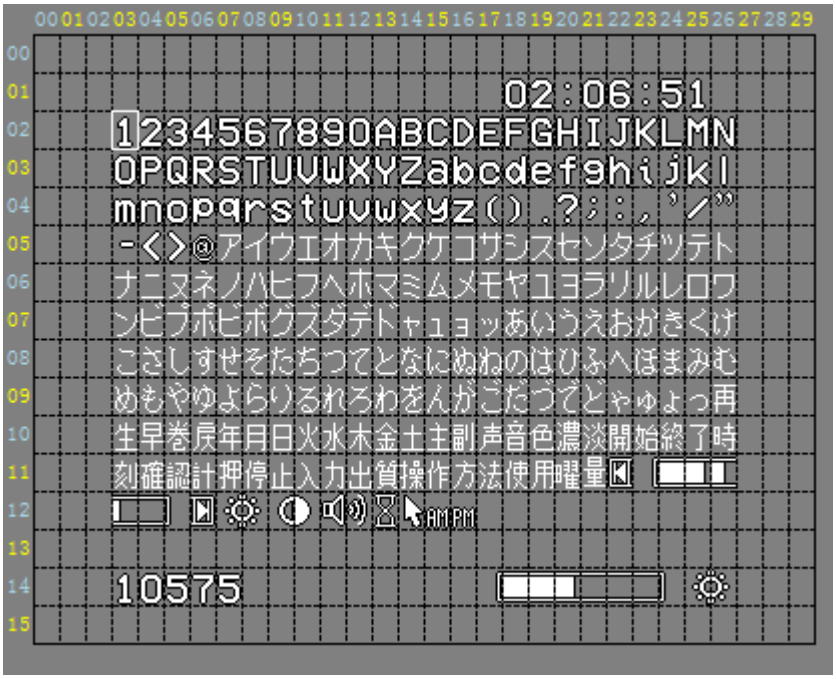
此外,页面下方提供批处理功能.

1） 刷新所有寄存器：从下位机读所有寄存器的值.

2） 批处理文件：将文件逐行发送,每行间隔20ms.用于批量发送命令.

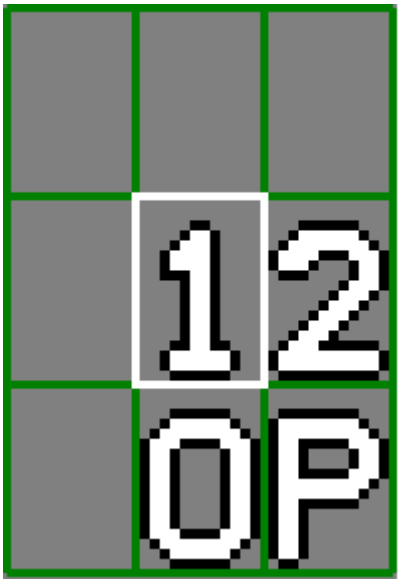
2.3 屏幕页：

本页模拟显示屏幕上的字符,并且可以对其进行修改.下图为模拟屏幕：



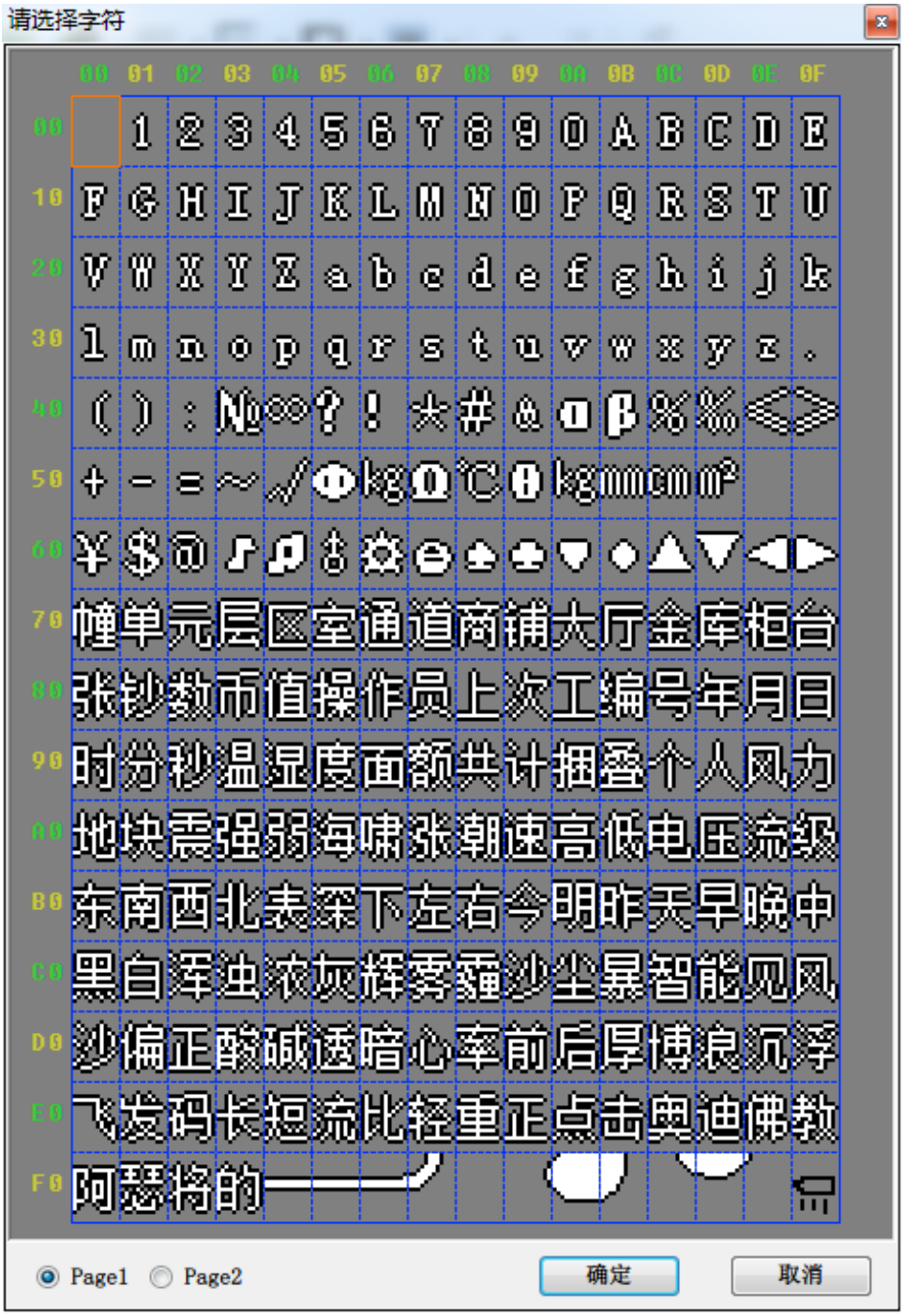
1) 显示器模拟界面为 30x16 个字符.
2) 单击屏幕中小方格：选择不同的位置(白方框表示当前位置).
3) 双击屏幕中小方格：弹出字符选择窗口. 选择需要的字符, 显示到模拟屏幕上, 并更新到下位机.

模拟显示器右侧为当前选中位置的放大图, 如下所示.



*当前选中位置的放大图. 放大比例为 5 倍.

下图为字符选择窗口：

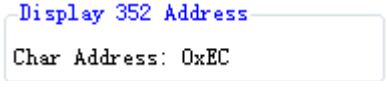


*选择字符窗口，双击需要的字符即可.

*Page1, Page2, 切换字库页面

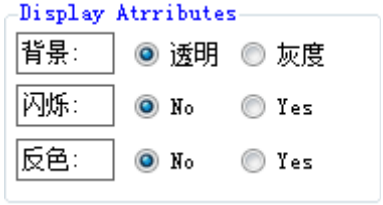
屏幕信息与编辑：

1） 屏幕选中位置和字符地址：此信息由下图所示界面显示



*字符地址为字符在字库中的地址

2） 屏幕显示属性：设置背景色，闪烁和反色．如下图：



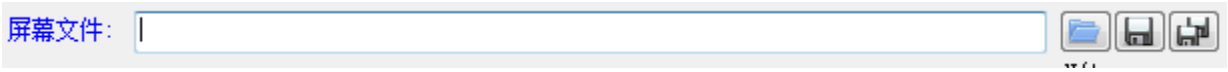
3） 屏幕交互：读取或设置下位机的屏幕字符．如下图：

*读取：获取指定行列的字符．
*获取整屏：从下位机获取整个屏幕的字符．
*发送整屏：发送整个屏幕的字符到下位机．

4） 编辑屏幕字符：编辑虚拟屏幕上的字符

*选择：图形界面选择字符，字符地址将显示在文本框中．
*更新间隔：全屏更新时，两次屏幕字符更新的间隔．
*全屏更新：以字库地址对应的字符，对全屏幕进行更新．
*清屏：将屏幕上的所有字符设置为字库中第一个字符．

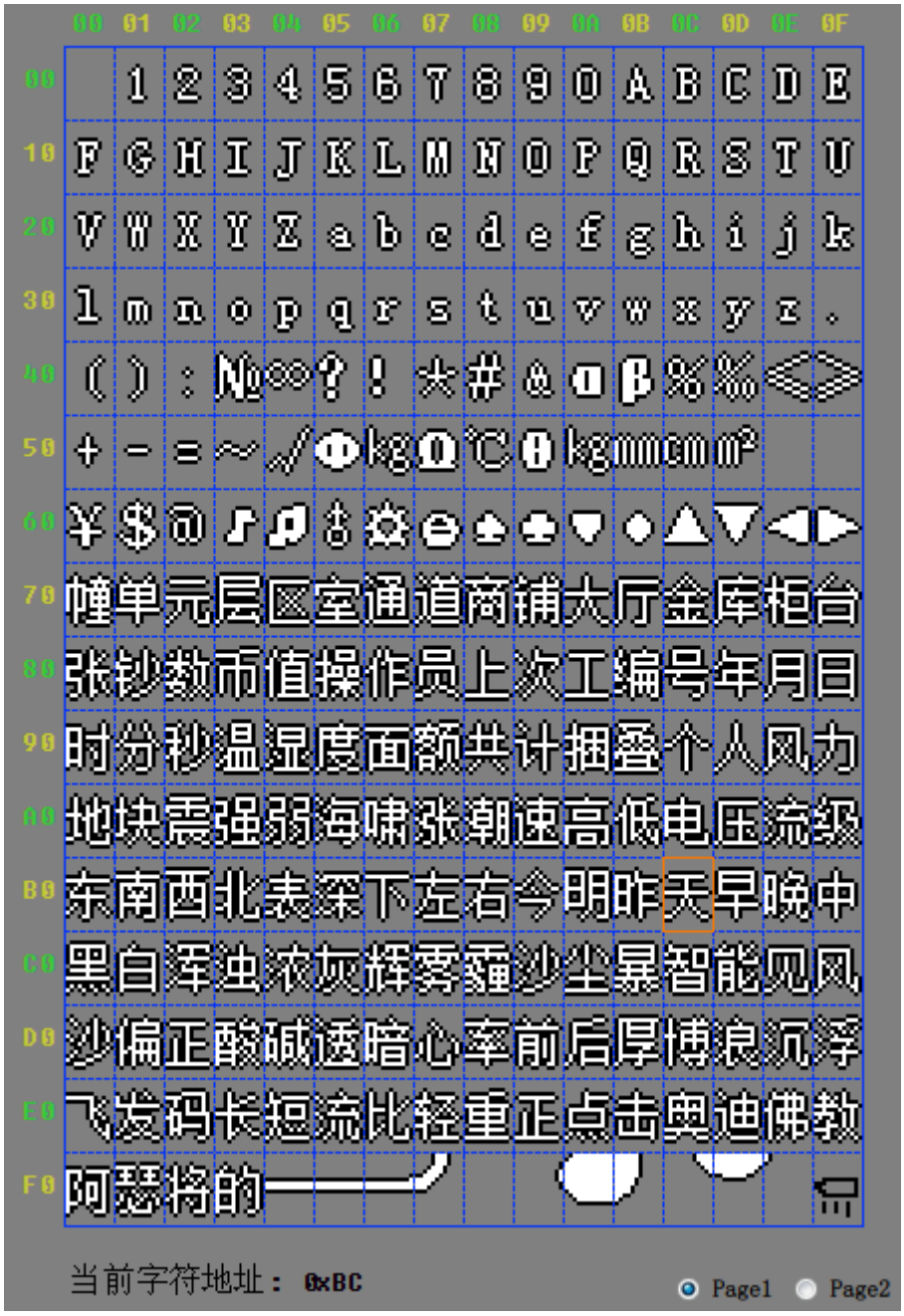
5） 屏幕文件：打开和保存屏幕信息到文件．如下图：



*屏幕文件：填写记录屏幕信息的 mdm 文件路径．
*保存：将当前屏幕保存到文本框指定的 mdm 文件．
*另存为：将当前屏幕另存为其他路径的 mdm 文件．

2.4　字符集页：

用于显示和编辑字符的页面. 下图为字库：



*字库由 16x16 个字符组成, 每个字符为 12x18 的点阵.

*组成字符的点, 有黑白灰三种颜色.

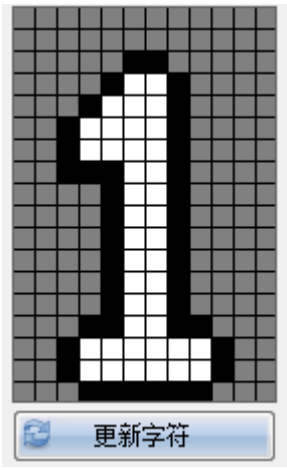
*单击字库, 可以选择相应位置的字符, 白色方框表示当前选择.

*最下方显示当前选择的字库地址.

*Page1, Page2：切换字库页面.

字库辅助功能：

1） 字符编辑：显示当前选中字符，并可以对其进行编辑。 如下图：

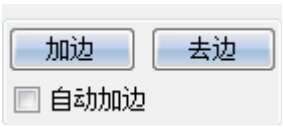


*单击小方格切换黑白灰三种颜色。
*更新字符：将当前字符发送给下位机。

2） 字符生成：根据用户的输入和指定的字体，生成字符。

*选择字体：选择字体和字号。
*提交：将文本框中的字符串，按选择好的字体，逐字写入字库。
*从文件：将文件中的字符串，按选择好的字体，逐字写入字库（忽略换行符）。

3） 字符黑边：字符白色区域周围的黑边。

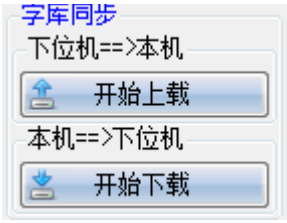


*加边：为当前字符加上黑边。
*去边：将当前字符的黑边去除。
*自动加边：选中此项后，使用字符输入将自动加黑边。

4） 清空字库和清除字符：
清空字库将整个字库清成全黑，全白或全灰，每按一次，切换一种颜色。
清除字符将当前选中字符清成全黑，全白或全灰，每按一次，切换一种颜色。

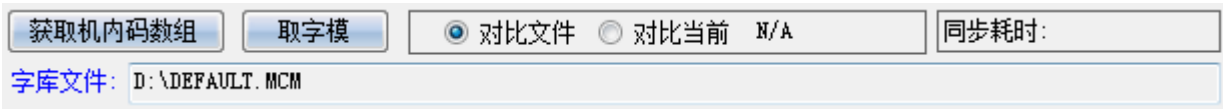
5） 字符移动：上下左右移动字符。

6） 字库的同步：



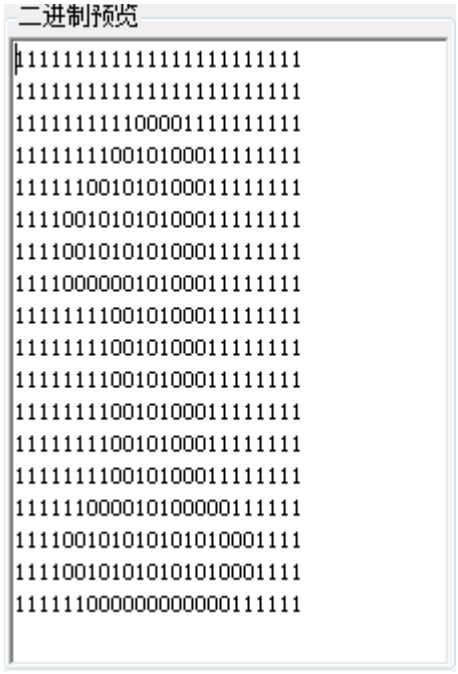
*开始上载：将下位机的字库读取到本程序，并在界面上显示.
*开始下载：将程序界面显示的字库更新到下位机.

7） 其他辅助功能：



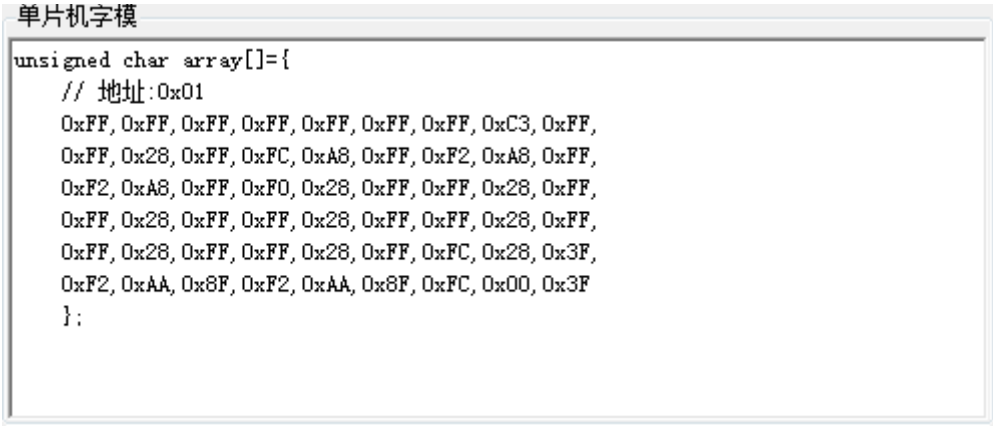
*获取机内码数组：获取字库字符的机内码，并组织成数组，用于下位机编程.
*取字模：获取字库字符的字模，并组织成代码，用于下位机编程.
*对比文件：上载字库时，下位机的字库是否和字库文件指向的文件一致.
*对比当前：上载字库时，下位机的字库是否和当前界面中的字库一致.
*同步耗时：显示同步字库所花去的时间.

8） 取字模窗口：为方便下位机开发提供的功能.
a） 二进制预览：为当前字符的二进制代码. 如下图所示：



*2 位数字表示一个像素点. 11 表示灰, 00 表示黑, 10 表示白.

b） 单片机字模：当前字符的 c 代码数组, 如下图所示：



*将二进制的字符信息转换成十六进制数, 生成字节(byte)数组.

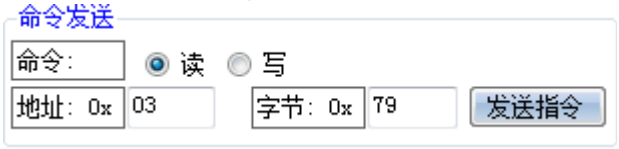
c） 字模文件的生成和查看：

*起始地址：当前字符的地址.
*长度：要生成字模的字符个数.
*不补零：不对字模做处理.
*行末补零：字符每行后补 8 个 0, 组成每行 32 位.
*字后补零：字符最后补 10 个字节的 0x00, 使每个字符为 64 字节.
*生成文件：填写字模文件路径.
*二进制文件：生成二进制格式的文件.
*字模代码：生成字模代码格式的文件.
*生成：生成字模文件.
*刷新：刷新单片机字模和二进制预览的显示.
*查看：查看当前文本框指向的文件.

2.5 串口信息页:

串口的接收,发送与记录.

1) 命令发送:寄存器读写命令的发送. 如下图:



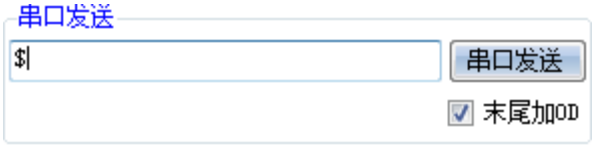
*命令:指定读写命令.

*地址:读写寄存器的地址.

*字节:写寄存器的值.

*发送指令:发送当前参数的命令.

2) 串口发送:发送自定义的字符串.



*串口发送:送文本框中的字符.

*尾加 0D:选中此项, 发送文本框字符+”\r”.

3) 串口数据记录:记录接收到和发送出的数据.

*清除显示:清除文本框显示.

*HEX:选中此项,接收文本框以十六进制显示收到的数据.

*RX:接收字节计数.

3 程序基本操作:

3.1 寄存器读写:

1) 选中寄存器 Tab 页.
2) 修改寄存器列表中的值,按下读写按钮.
3) 或双击寄存器名,在弹出的高级界面中修改.

3.2 屏幕编辑与交互:

1) 选中屏幕 Tab 页.
2) 使用获取整平从下位机获取屏幕信息.
3) 双击模拟屏幕要改写的位置,选择需要的字符.
4) 如果需要修改显示属性,则选择需要的背景色,闪烁和反色属性.
5) 使用发送整屏修改下位机显示.

3.3 字库编辑与交互:

1) 选中字库 Tab 页.
2) 打开字库文件,或使用上载字库从下位机获取字库.
3) 根据需求,对字库进行修改.
4) 使用下载字库将编辑好的字库更新到下位机.

杭州中科微电子应用系统部
2016 年 11 月 15 日