

## 附录 A EasyARM 软件的使用

为了给系统提供更友好的人机界面,我们可以通过上位机软件实现各种显示输出或操作输入, EasyARM 软件是上位机人机界面软件,通过 RS232 串口通讯完成各种功能控制。

### A.1 EasyARM 软件窗口介绍

全仿真的 DOS 字符窗口是具有 25 行 80 列的字符显示窗(显示字符的前景/背景颜色可设置),具有 8 个仿真 LED 数码管和 8 个仿真发光二极管,还有 20 个模拟按键(按键名可重新定义)。串口模式可设置,具有单独的数据发送/接收调试窗,方便地监视串口接收到的数据或调试串口。另外,具有一个万年历的界面,可以用于 LPC2000 系列微控制器的实时时钟实验。EasyARM 软件主窗口如图 A.1 所示。

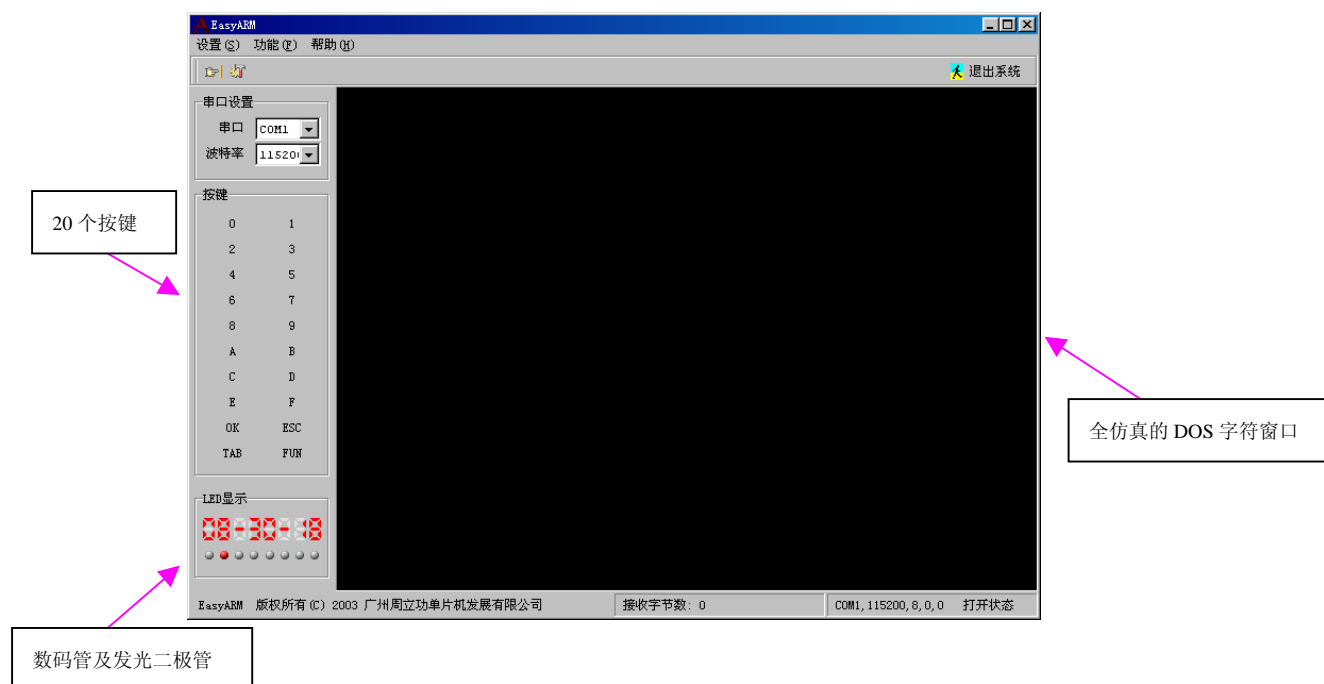


图 A.1 EasyARM 主窗口

打开菜单“功能”中的“万年历”,即弹出仿真万年历窗口,用于 RTC 实验显示等,如图 A.2 所示。



图 A.2 仿真万年历窗口

打开菜单“设置”中的“串行口设置”,即弹出串行口设置对话框,用于设置串口工作模式,如图 A.3 所示。

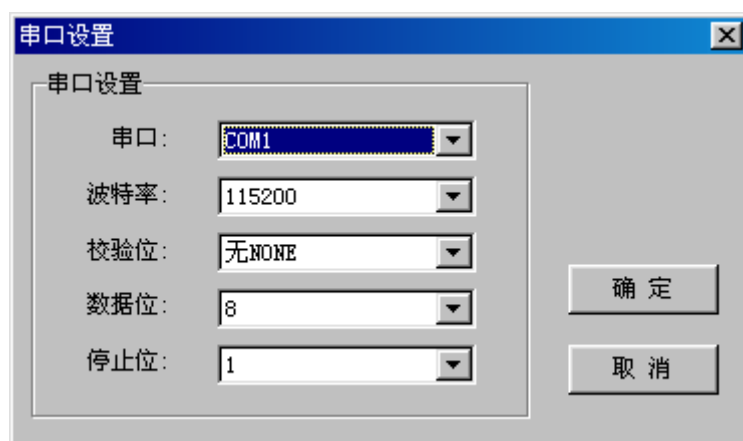


图 A.3 EasyARM 串口设置

打开菜单“设置”中的“发送数据”，即可打开串口发送/接收窗口，用于串口调试，如图 A.4 所示。



图 A.4 EasyARM 串口发送/接收窗口

## A. 2 EasyARM 软件通讯协议

### 1. 全仿真的 DOS 字符窗口显示

发送数据格式为： 0xff x y chr color （先发送 0xff，最后发送 color）

0xff: 起始字节

x: 显示位置的纵坐标，0~79

y: 显示位置的横坐标，0~24

chr: 显示的字符，不能为 0xff

color: 显示的状态包括前景色、背景色、闪耀位。它与 DOS 的字符显示状态一样。  
即 0~3 位：前景色，4~6 位：背景色，7 位：闪耀位。

color 的颜色取值参考程序清单 A.1。

## 2. 仿真的 LED 数码显示器显示

发送数据格式为: 0xff 0x80 x data 0

0xff: 起始字节

0x80: 表明在 LED 上显示

x: 显示位置 0~8, 其中 8 为 LED 灯

data: 显示的笔画, 其中 1 为点亮, 0 为熄灭。

0: 仅避免出现 0xff

## 3. 仿真的万年历显示器显示

发送数据格式为: 0xff 0x81 x data 0

0xff: 起始字节

0x81: 表明在 LED 上显示

x: 显示位置 0~14, 分别对应于年、月、日、星期、时、分、针

data: 显示的笔画, 其中 1 为点亮, 0 为熄灭。

0: 仅避免出现 0xff

## 4. 模拟键盘输入

PC 直接向串口发送键盘编码 0~19 (一个字节)。

按键名可在 EasyARM.exe 文件所在目录下的 EasyARM.ini 文件定义, 即在[KeyName]项下更改对应的键名, 如 Key10=A, Key11=B 等等。

## 5. 仿真的 DOS 字符窗口字符颜色

程序清单 A.1 定义了各种颜色值的常量。

程序清单 A.1 仿真 DOS 窗字符颜色

```
/*
*****
* 文件名: COLOR.H
* 功能: DOS 窗口字符显示颜色定义。
* 说明:
*****
/* 前景颜色 */

#define DISP_FGND_BLACK          0x00
#define DISP_FGND_BLUE           0x01
#define DISP_FGND_GREEN          0x02
#define DISP_FGND_CYAN           0x03
#define DISP_FGND_RED             0x04
#define DISP_FGND_PURPLE          0x05
#define DISP_FGND_BROWN          0x06
#define DISP_FGND_LIGHT_GRAY      0x07
#define DISP_FGND_DARK_GRAY       0x08
#define DISP_FGND_LIGHT_BLUE      0x09
#define DISP_FGND_LIGHT_GREEN     0x0A
#define DISP_FGND_LIGHT_CYAN      0x0B
#define DISP_FGND_LIGHT_RED       0x0C
#define DISP_FGND_LIGHT_PURPLE    0x0D
```

```
#define DISP_FGND_YELLOW      0x0E
#define DISP_FGND_WHITE      0x0F

/* 背景颜色 */
#define DISP_BGND_BLACK      0x00
#define DISP_BGND_BLUE      0x10
#define DISP_BGND_GREEN      0x20
#define DISP_BGND_CYAN      0x30
#define DISP_BGND_RED        0x40
#define DISP_BGND_PURPLE     0x50
#define DISP_BGND_BROWN     0x60
#define DISP_BGND_LIGHT_GRAY 0x70

/* 闪烁控制 */
#define DISP_BLINK            0x80
```