LCDGO
艾斯迪科技
http://www.lcdgo.com

三易串口屏指令介绍

版本1.0 2020-6-18

串口指令简介

通讯指令

- 通讯指令，是用户向串口屏模块发送的指令。

脚本指令

- 脚本指令，是串口屏模块内部脚本，计算数据，控制页面控件，以及发送数据给用户。

串口通讯指令介绍

功能

- 用户通过通讯指令控制串口屏模块。
- 通讯指令通过串口发送，发送方式为ASCII码方式。

举例

- 假设用户发送串口数据的函数定义为： void uart_send(char *ptr)。
- 页跳转的方式为：uart_send（“page 8\r\n”），串口数据为“page 8\r\n”。

特点

- 通过精简的方式控制串口屏模块，命令极少。
- 串口数据为 ascii码方式，更加人性化，也方便初学者上手，曲线透传数据除外。
- 串口命令以“\r\n”作为帧尾，没有数据帧头，减少客户组合数据时的操作。

回传查看所有指令 ： help

指令名	功能	参数及举例
help	回传所有可用串口指令	无参数。 串口数据举例：“help\r\n” 回传数据： help - help page - page name/page id wset - wset name value wget - wget name sset - sset name value sget - sget name event - event name val click - click name adjust - adjust factory - return to factory,only valide 100ms before power on addt - HEX format pisx - HEX format version - get information reset - reset 1

页面跳转指令 ： page

指令名	功能	参数及举例
page	页面跳转	page name/id 参数: 页面名称/页面ID 举例: 跳转到页面“main”，页面”main”的id为 0， 则命令为: “page main\r\n” 或者 “page 0\r\n”。 回传数据: 参看后面通讯指令返回值介绍 注: 只有此命令可以使用ID号来跳转。

控件属性设置指令 ： wset

指令名	功能	参数及举例

wset	设置页面以及控件的属性值	wset name value
		参数:
		页面\控件名称·属性名称 属性值。
		举例:
		“wset text1.txt 明天会更好\r\n”
		//设置当前显示页面的文本控件“text1”的“txt”为“明天会更好”
		“wset main.text1.txt “明天会更好”\r\n”
		//设置页面‘main’中文本控件“text1”的“txt”为“明天会更好”
		
		回传数据：执行结果。参看后面通讯指令返回值介绍。

说明:

页面\控件名称是带有可查找路径的名称，要设置页面的属性，则用“wset 页面名称.属性名称”，如:

```
wset main.bgColor 0xffFF0000\r\n;     //设置页面main的背景颜色为红色
wset main.num1.val 123\r\n;           //设置页面main中整数控件num1的当前值为123
```

属性值有三种类型，整数，小数，字符串，整数和小数都用ascii码表示，例如 123 12.3，发送时用“123”，“12.3”。中英文字符串，注意制作串口屏工程时选择的字符编码（工具->项目设置），和发送命令的用户编译环境选择的字符编码一致。如：VP中字符编码选择GBK，那么用户发送的字符串编码也得是GBK。

如果设置页面的属性，则必须写明页面路径。

控件属性值获取指令 ： wget

指令名	功能	参数及举例

wget	取得页面以及控件的属性值	wget name
		参数:
		页面\控件名称·属性名称。
		举例:
		“wget text1.txt\r\n”
		//读取当前页面文本控件text1的txt属性
		“wget main.text1.txt\r\n”
		//读取页面main中的文本控件text1的txt属性
		
		回传：属性值。参看后面通讯指令返回值介绍。

说明:

页面\控件名称是带有可查找路径的名称，如果要读取页面的属性，则用“wget 页面名称.属性名称”， 如:

“wget main.name\r\n” //获取main页面的页名称（读出来是字符串形类型，可赋给文本控件打印显示）

“wget page1.numf1.valf\r\n” //获取page1页面的浮点数控件numf1的浮点数值

返回的属性值有三种类型，整数，小数，字符串，整数和小数都用十六进制码表示，小端在前；中英文字符串，返回中英文的ascii码值或者区位码值，十六进制表示。注意制作串口屏工程时选择的字符编码（工具->项目设置），和发送命令的用户编译环境选择的字符编码一致。

如果获取页面的属性，则必须写明页面路径。

设置系统变量 ： sset

指令名	功能	参数及举例
sset	设置系统变量	sset name value 参数: 系统变量名称 系统变量值（全部为整数类型）。 举例: 设置系统亮度为100 。 命令: sset sys_light 100\r\n。 回传数据: 执行结果。参看后面通讯指令返回值介绍。 说明: sys_light ，是系统变量－背光亮度的名称，100是亮度级别。系统变量会在后面页面列举介绍。

获取系统变量值 ： sget

指令名	功能	参数及举例
sget	读取系统变量的值	sget name 参数： 系统变量名称 。 举例： 取得系统亮度为100 。 命令： sget sys_light\r\n 回传数据：系统变量值。参看后面通讯指令返回值介绍。 说明： sys_light ，是系统变量－ 背光亮度的名称，亮度级别 1-100。系统变量会在后面页面列举介绍。

事件触发 ： event

指令名	功能	参数及举例
event	事件触发	event name val 参数: name ： 控件名称，支持 按钮，触摸热区，双态按钮，定时器。 val：触发类型值：1 按下事件，2 弹起事件。 举例: event button0 1\r\n //触发按钮button0的按下事件。

说明:
此指令可以触发按钮控件，触摸热区控件，双态按钮控件的按下、弹起的处理脚本。
event timer 1\r\n
//触发定时器脚本，会立即执行定时器脚本，定时器的interval设置（触发事件的间隔时间）将无效。
wset timer.en 1\r\n
//打开定时器，定时器里面的脚本会在interval的设置间隔时间之后执行。

模拟点击 ： click

指令名	功能	参数及举例
click	模拟点击事件	click name 参数: name ： 控件名称，支持 按钮，触摸热区，双态按钮。 举例: click button0\r\n //模拟按钮button0被按下。 发送此指令按下按钮后，就等于模拟做了按下弹起动作，按钮会执行按下和弹起里面的脚本事件。

说明:
此指令可以模拟按下按钮控件，触摸热区控件，双态按钮控件。
方便没有触摸屏时，用户使用物理按键操作串口屏模组。

触摸校准&恢复出厂 ： adjust & factory

指令名	功能	参数及举例
adjust	触发触屏校准功能	adjust 参数：无 举例：adjust\r\n 注意：进入触屏校准页面后，需手动触屏校准完才退出。 此指令对VP编辑软件无效。
factory	恢复出厂	factory 参数：无 举例：factory\r\n 注意：此命令会擦除用户工程。对固件无影响。 如果需要重新下载串口屏工程，请重新上电下载。 此指令对VP编辑软件无效。

曲线数据透传 ： addt

指令名	功能	参数及举例
addt	曲线控件数据透传指令 该指令参数使用16进制数据，不需要结束符，回车换行\r\n(0x0D 0x0A)	命令格式： [0~3] 指令名addt，即 0x61 0x64 0x64 0x74 [4] 页ID [5] 控件ID [6] 通道号0~2，一个曲线控件最多3个通道 [7~8] 数据个数，小端模式，例如100个数据，十六进制为0x0064，命令中为64 00 [9~N] 数据。可以分为4个类型：int（4字节） short(2字节） char（1字节） float（4字节），在串口屏工程中曲线控件属性中设置，组织传输数据时，要和属性设置一致。

举例：

设置0通道61 64 64 74 00 04 00 0A 00 00 02 04 06 08 0A 0C 0E 10 12

|指令名addt | 1| 2| 3|数据个数|数据 | 1->页id 2->控件id 3->通道号

设置1通道61 64 64 74 00 04 01 0A 00 00 04 08 0C 10 14 18 1C 20 24

设置2通道61 64 64 74 00 04 02 0A 00 00 01 02 03 04 05 06 07 08 09

显示点阵 ： pisx

指令名	功能	参数及举例
pisx	显示点阵数据	参数介绍： [0~3]指令名pisx，即 0x70 0x69 0x78 0x73 [4-5] 起始横坐标 [6-7] 起始竖坐标 [8]单个字模宽度 [9]单个字模高度 [10-12]颜色，小端模式，如红色0xFF0000，[11]00，[12]00，[13]FF [13-14]字模数据字节数 [15-N]字模数据，1bit对应一个像素点 取模说明：逐行式，从第一行开始向右每取8个点作为一个字节，如果最后不足8个点就补满8位。取模顺序是从高到低，即第一个点作为最高位

说明：

此功能实现了客户想显示一些字库中没有特殊字符或者单色图片，显示时需要用户实时发送，例如 笑脸等用户自定义有个性的图标。

举例：

70 69 78 73 \\命令码 | 00 00 \\起始横坐标 | 0A 00 \\起始竖坐标 | 10 \\字模宽度16 | 10\\字模高度16
00 00 FF \\颜色0xFF0000，红色 |20 00 \\字模字节数32
08 80 08 80 08 80 11 FE 11 02 32 04 34 20 50 20 91 28 11 24 12 24 12 22 14 22 10 20 10 A0 10 40
\\字模数据”你”，16*16，宋体，不加粗，不斜体

查询版本信息 ： version

指令名	功能	参数及举例
version	查询版本信息	version 参数：无。 举例： version\r\n　查询串口屏硬件相关信息。

说明：
此指令可查询串口屏的型号，flash大小，固件的时间版本。
在VP编辑软件或其它串口助手上发送，会返回上述信息。
此指令需用串口连接串口屏后查询才有返回信息。

复位屏幕 ： reset

指令名	功能	参数及举例
reset	复位屏幕	reset 1 参数：1 举例： reset 1\r\n

说明：
此指令可以模拟屏幕断电重启。
通过串口发送指令复位，无需断电也可重启复位串口屏。
此指令对VP编辑软件无效。

系统变量介绍

编号	变量名	说明	取值范围
1	sys_pid	当前页面ID（只读）	0~255
2	sys_baud	当前波特率（只读）	设备支持的波特率：4800，9600，14400，19200，38400，56000，57600，115200，256000，460800，921600
3	sys_light	亮度（可读写）	0~100
4	sys_sleep	是否休眠（可读写）	0不休眠，1休眠
5	sys_slp_cls	休眠等级（可读写）	0为普通休眠（熄灭背光），1为深度休眠
6	sys_slp_nu	无串口数据自动睡眠时间（可读写）	单位:秒 0~255 0表示不睡眠
7	sys_slp_nt	无触摸操作自动睡眠时间（可读写）	单位:秒 0~255 0表示不睡眠
8	sys_wup_bt	睡眠模式下触摸唤醒开关（可读写）	0睡眠后触摸不会唤醒 1睡眠后触摸唤醒
9	sys_wup_bu	睡眠模式下串口数据唤醒开关（可读写）	0睡眠后串口不会唤醒 1睡眠后串口唤醒
10	sys_bkcmd	设置指令执行结果的返回（只读）	0x00不返回结果 0x01只返回成功的结果 0x02只返回失败的结果 0x03成功或者失败都返回结果

编号	变量名	说明	取值范围
11	rtc_year	使用RTC时获取时间年（可读写）	起始年：2021
12	rtc_month	使用RTC时获取时间月（可读写）	1-12
13	rtc_day	使用RTC时获取时间日（可读写）	1-28/29/30/31
14	rtc_hour	使用RTC时获取时间时（可读写）	00:00-24:00
15	rtc_minute	使用RTC时获取时间分（可读写）	00-59
16	rtc_second	使用RTC时获取时间秒（可读写）	00-59
17	rtc_week	使用RTC时获取时间周（只读）	周一到周六为：1-6，周日：0
说明：以上RTC操作在硬件支持的屏幕上有效。RTC系统变量以外的变量，本次上电事件内修改有效，掉电后改为串口屏工程内设定值。			

通讯指令返回值介绍 1

条件

- 只有当系统变量bkcmd为非0的时候才会返回指令执行成功或者失败数据。

格式

- 返回指令格式为 命令码(1字节)+参数(长度随命令码而不同)+结束符(2字节，\r\n)

数据

- 所有指令名以及参数全部16进制数据，非ASCII数据，便于软件解析。

通讯指令返回值介绍 2

命令码	含义	格式	参数说明
0x00	指令执行结果	执行结果代码 +返回数据 +结束符（\r\n）	执行结果代码：1字节 0x00　指令执行成功 0x01　命令码无效（指令第一个字符串无效） 0x02　指令参数1无效（指令第二个字符串无效） 0x03　指令参数2无效（指令第三个字符串无效） 0x04　指令参数3无效（指令第四个字符串无效） 0x05　指令参数4无效（指令第五个字符串无效） 0x06　指令参数5无效（指令第六个字符串无效） 0x07　指令参数个数错误 0x08　串口接收超时
	说明： 命令执行结果返回值，主要用于客户取得串口屏控件的属性值，响应客户发送的命令，如果客户发送的指令有错，例如命令码拼写错误，属性名称拼写错误，属性名称拼写错误等，造成串口屏不能解析，将返回不同的错误码，方便客户查错。 只有当系统变量**bkcmd**为非0的时候才会返回指令执行成功或者失败数据。 这个命令提供了一种客户的调试命令的方式，大部分情况下，在调试完毕后，都会把系统变量**bkcmd置0**，不会收到此命令。 举例：如果客户发送指令的命令码拼写错误，返回：“00010d0a”。解释：从左到右，00 表示返回的命令码，01表示命令码无效，0d表示 \r，0a表示 \n。		

通讯指令返回值介绍 3

命令码	含义	格式	参数说明
0x01	整数返回	0x01+4字节整数+结束符	4字节整数：4字节小端模式存储，低字节在前
0x02	小数返回	0x02+4字节小数+结束符	4字节小数：4字节小端模式存储，低字节在前
0x03	字符串返回	0x03+字符串+结束符	字符串：GBK编码
	返回结果包括客户发起的命令执行情况和需要返回的数据。执行情况即执行结果，需要返回的数据主要是取串口屏控件属性的值。值的类型为整数，小数，字符串。 举例：获取按钮1的x坐标。发送指令“wget button1 x\r\n”，返回为：“01880000000d0a” 解释：01 表示返回值为正数；88000000表示位button1的 x坐标，低字节在前的格式；0D0A是命令结束符。 如果系统变量bkcmd没有设置为0，这个指令会接收到两个返回指令，“00000d0a”，和“01880000000d0a”，第一个指令表示命令执行成功，第二个指令是用户需要的结果。如果系统变量bkcmd设置为0，则只返回第二个指令。		

三易串口屏脚本指令简介

定义

- 编辑脚本是用户制作串口屏功能的重要工具，脚本的编辑方式完全兼容C语言语法，脚本的编译器是真正的C语言编译器。

功能

- 脚本指令，是一段执行代码，C语言格式，可以通过按钮，定时器，触摸热区，滑块，用户串口命令等来触发。

脚本指令基本语法概要介绍

项目	功能	介绍
分隔符	语句结束符，“；”	每条语句以此为分隔符，同C语言。
标识符	标识变量名称	一个标识符以字母 A-Z 或 a-z 或下划线 _ 开始，后跟零个或多个字母、下划线和数字（0-9）。标识符区分大小写。同C语言。
关键字	语言组成的关键字	目前支持：if,else ,for, while,do while, break ,continue,int , float
数据类型	变量的数据类型定义	目前支持：int float 4字节。
运算符	变量常量之间的运算	目前支持：算术运算符，关系运算符，逻辑运算符
函数	带参数的功能处理集合	支持带变量的函数处理，方式同C语言。
字符串	字符串常量	目前支持：字符串相加，用“+”运算符执行
变量	可变数值的代表标识符	目前支持：脚本定义类型支持 int float byte，作用区域只在当前事件编辑区里，而变量控件可作用于全局。

脚本指令详细介绍：变量与数据类型

介绍

- 变量分为两种变量
- 控件变量，带有属性，属性值可以作为变量直接使用。
- 脚本变量，同C语言变量。

声明

- 变量的数据类型支持：整型 int，浮点型 float，字符串 string，字节 byte。
- 声明一个整型变量： int a； 声明一个浮点型变量：float b；定义一个byte数组：byte a[3];
- 字符串型，在变量控件的变量类型属性中选择。

赋值

- 整型变量： int a=100;
- 浮点型变量：float b=0.01;
- byte数组： byte a[3]={0x01,0x02,0x03};
- 其他各种操作符同C语言。

脚本指令详细介绍：运算符

算术

- 算术运算符包括：+ - * / %，对应 加 减 乘 除 取余，同C语言。
- 精度：4字节的整数或者浮点数。

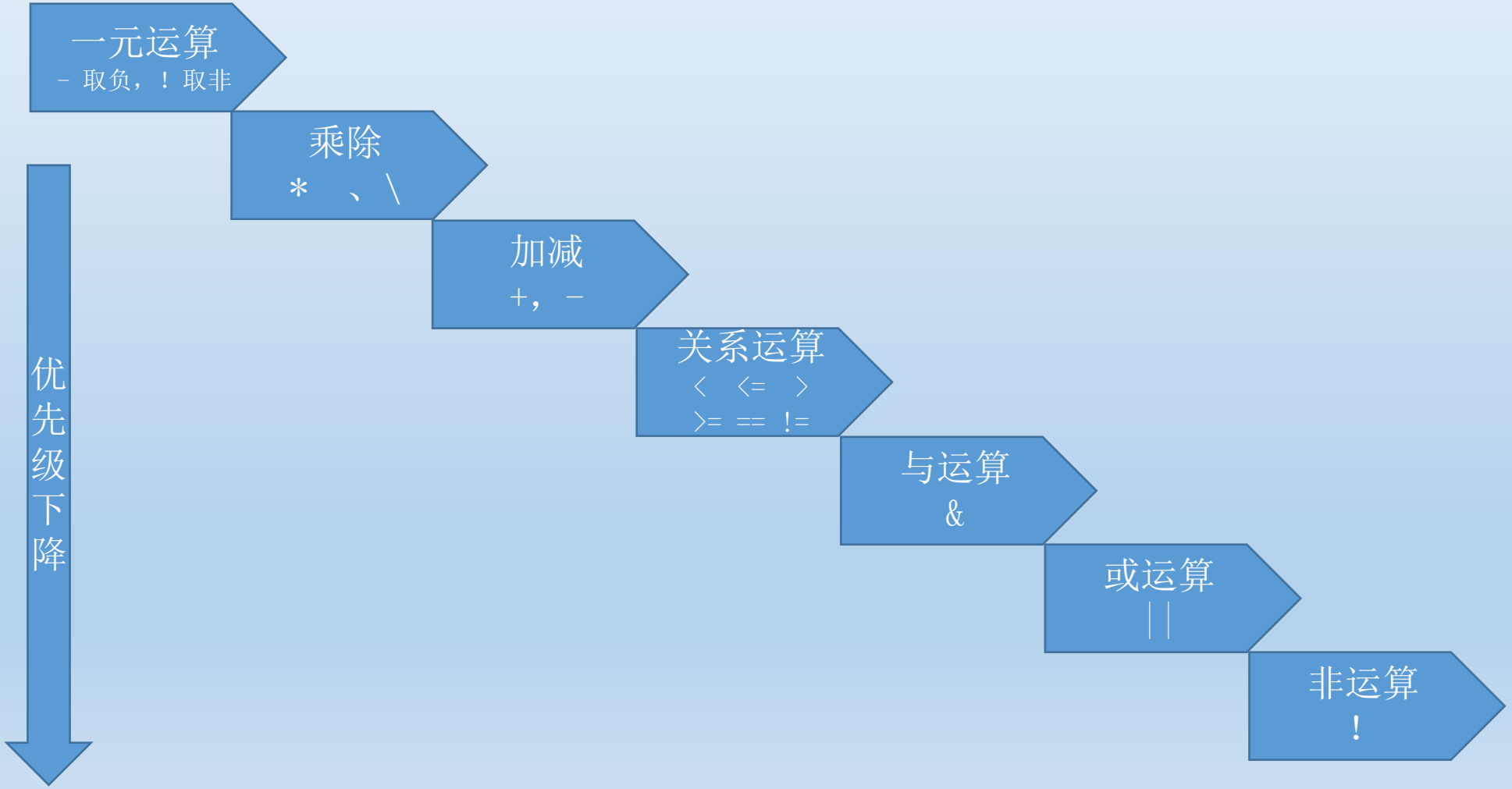
关系

- 关系运算符包括：< 小于，<= 小于等于，>大于 ，>=大于等于， ==是否相等， !=是否不等。
- 同C语言。

逻辑

- 逻辑运算符包括：&& 与运算， || 或运算，！非运算。
- 同C语言。

脚本指令详细介绍：运算符优先级



脚本指令详细介绍：条件

基本

- 目前支持条件关键字为 if … else.
- 一个 **if 语句** 由一个布尔表达式后跟一个或多个语句组成。如果是一条语句，if，else后不用大括号{}，如果为多条，请用大括号{}，同C语言。

嵌套

- 用户可以在一个 **if** 或 **else if** 语句内使用另一个 **if** 或 **else if** 语句。
- **同C语言。**

举例：

```
if(b0.txt==“确定”)
{
  b0.txt=“取消”;
}else if(b0.txt==“取消”)
{
  b0.txt=“等待”;
}else
{
  b0.txt=“确定”;
}
```

b0是按钮控件的名字，txt是按钮控件的属性。这个例子完成按钮控件上文字的变换。

脚本指令详细介绍：循环

for

- 固定多次执行一个语句序列，简化管理循环变量的代码。同C语言。

while

- 当给定条件为真时，重复语句或语句组。它会在执行循环主体之前测试条件。同C语言。

for

- 除了它是在循环主体结尾测试条件外，其他与 while 语句类似，同C语言。

break

- 终止循环语句，程序流将继续执行紧接着循环的下一条语句。同C语言。

continue

- 告诉一个循环体立刻停止本次循环迭代，重新开始下次循环迭代。同C语言。

```
while(a>10)
{
……
}

do
{
……
}while(a>10)

for (a = 10; a < 100; a = a + 1)
{
 ……
}
```

脚本指令详细介绍：函数

介绍

- 串口屏脚本中的函数为内置函数，类似系统提供的工具，完成一些脚本不能实现的功能。

使用

- 使用方式同C语言

脚本指令：翻页函数 - showPage

介绍

- 跳转到某一页。

使用

- void showPage(int a)

说明：

此函数可以实现跳转到某一页，参数为页的id号。这是一个常用和方便的函数。

举例：

跳转到主页，主页id为整数0；

showPage（0）；

脚本指令：亮度调节函数 - setBright

介绍

- 设置背光亮度，100级可调。

定义

- Void setBright（int）

举例：
此函数参数范围 0-100；

设置背光亮度为最大值
setBright（100）；

关掉背光
setBright（0）；

脚本指令：随机数函数 - getRandom

介绍

- 取得a~b之间的一个随机数。

定义

- int getRandom(int a, int b)

说明：

此函数返回整数 a到 b之间（包含a和b）的一个随机数，可以使用这个函数获得随机数，方便做一些特效。

举例：

```
Int ran;

ran = getRandom（1,9999）;
```

脚本指令：外发字符串函数 - uartSend

介绍

- 串口发送字符串数据给用户端。

定义

- void uartSend(string a)

字符串中用到的转义符定义		
字符	ASCII	含义
\n	010	换行(LF)，将当前位置移到下一行开头
\\	092	代表一个反斜线字符’’\’
\″	034	代表一个双引号字符

说明：

此函数用来发送字符串数据给用户的控制端。客户根据自己需要的格式组织字符串，调用这个函数发送给控制端。
注意：发送的字符串没有‘\0’结尾，只有字符串内容。

举例：

发送 一个文本控件的txt属性值给控制端：

uartSend（text1.txt）；
text1.txt 内容可以是中英文符号混杂。

脚本指令：前退格函数 - stringTrimStart

介绍

- 移除字符串S从第一个字符开始，指定个数字符，返回新字符串。

定义

- string stringTrimStart(string s,int count)

说明：

此函数方便用户做字符串裁剪。

举例：

```
String s1, s2;
s1=“123456”;
s2=stringTrimStart(s1,3);
```

s2的值变为“456”。

脚本指令：后退格函数 - stringTrimEnd

介绍

- 移除字符串S从最后一个字符开始，指定个数字符，返回新字符串。

定义

- string stringTrimEnd(string s,int count)

说明：

此函数方便用户做字符串裁剪。

举例：

```
String s1, s2;
s1=“123456”;
s2=stringTrimEnd(s1,3);
```

s2的值变为“123”。

脚本指令：整型转字符串型函数 - intToString

介绍

- 转一个整数为字符串形式。

定义

- string intToString(int i);

说明：

此函数提供客户转整型数据为字符串数据，方便显示到控件中，因为控件的显示方式都是字符串。

举例：

```
int a;
string b;
 a=2020;
b=intToString(a);
```

b的值变为“2020”。

脚本指令：浮点数型转字符串型函数 - floatToString

介绍

- 转换浮点数为字符串。

定义

- string floatToString(float f,int num);

说明：

此函数提供用户取得浮点数的字符串标识，小数部分需要指定多少位。

举例：

```
float a;
int b;
string c;

 a=2020.0101;
 b=2;
 c=floatToString (a, b);
```

c 的值变为“2020.01”。

脚本指令：浮点数型转整型函数 - floatToInt

介绍

- 返回浮点数的整数部分。

定义

- int floatToInt(float f);

说明：

此函数用来取得浮点数的整数部分。

举例：

```
float a;

 a=2020.0101;
 c=floatToString (a);
 c 的值变为“2020”。
```

脚本指令：字符串型转浮点数型函数 - stringToFloat

介绍

- 转换字符串为浮点数。

定义

- float stringToFloat(string s);

说明：

此函数用来将字符串转换成浮点数。

举例：

```
string s;

 s=2020.0101;
 float a;
 a=stringToFloat(s);
```

a 的值变为浮点数“2020.0101”。

脚本指令：字符串型转整型函数 - stringToInt

介绍

- 转换字符串为整数。

定义

- int stringToInt(string s);

说明：

此函数用来将字符串转换成整数。

举例：

```
string s;

 s=2020.0101;
 int a;
 a=stringToInt(s);
```

a 的值变为整数“2020”。

脚本指令：串口屏校准函数 - screenAdjust

介绍

- 屏幕校准函数

定义

- void screenAdjust(void);

说明：

此函数用来进行屏幕校准。

举例：

screenAdjust();
可直接进入校准，此函数无参数。

脚本指令：字节数组发送函数 - uartSendBytes

介绍

- 串口发送字节数组函数

定义

- void uartSendBytes(byte array[], int start, int len);

说明：
byte array[]：发送字节数组
int start：起始位置为start
int len：长度为 len
此函数用来通过串口发送数据。
举例：
定义一个数组array：

```
byte array[5];
array[0] = 0x0F;
array[1] = 0xF0;
array[2] = 0xFF;
array[3] = 0x33;
array[4] = 0x03;
uartSendBytes(array,0,5);
```

通过串口发送数组array的定义值,起始位置从下标0开始，长度为5。
具体用法可参考协议解析器说明文档，或者下载官网的参考工程。

脚本指令：CRC16函数 - crc16

介绍

- 计算字节数组的CRC16函数

定义

- int crc16(byte array[], int start, int len);

说明：
byte array[]：字节数组
int start：起始位置为 start
int len：长度为 len
此函数用来计算校验码。
配合上位机协议解析器控件使用。
举例：
定义一个数组array：

```
byte array[5];
array[0] = 0x0F;
array[1] = 0xF0;
array[2] = 0xFF;
array[3] = 0x33;
array[4] = 0x03;
crc16(array,0,5);
```

通过函数计算array的校验码,起始位置从下标0开始，长度为5。
具体用法可参考协议解析器说明文档，或者下载官网的参考工程。

脚本指令：提取Ascii字符串函数 - bytesToAscii

介绍

- 从字节数组中提取Ascii字符串

定义

- string bytesToAscii(byte array[], int start, int len);

说明：
byte array[]：字节数组
int start：起始位置为 start
int len：提取多少个字节
此函数用来从字节数组中提取Ascii字符串。
配合上位机协议解析器控件使用。
举例：
protocol5.rxBuf：协议解析器的接收数据缓存区
text1.txt = bytesToAscii(protocol5.rxBuf,11,20);
取出数组中字符串中的一段，赋给文本控件显示（从十一位开始，向后取二十个）
可在官方网站下载相关工程参考。

脚本指令：提取标准字符串函数 - bytesToString

介绍

- 从字节数组中提取标准字符串。2个字节表示一个字。

定义

- string bytesToString(byte array[], int start, int len, int mode);

说明：
byte array[]：字节数组
int start：起始位置为 start
int len：提取多少个字节，必须是2的整数倍
mode：大小端。0 为小端模式，1为大端模式
配合上位机协议解析器控件使用。
举例：
protocol5.rxBuf：协议解析器的接收数据缓存区
text1.txt = bytesToString(protocol5.rxBuf,11, 20, 1);
取出数组中字符串中的一段，赋给文本控件显示（从十一位开始，向后取二十个，大端模式）

脚本指令：提取整数函数 - bytesToInt

介绍

- 从字节数组中提取整数。

定义

- int bytesToInt(byte array[], int start, int mode);

说明：
byte array[]：字节数组
int start：起始位置为 start
mode：大小端。0 为小端模式，1为大端模式
配合上位机协议解析器控件使用。
举例：
protocol5.rxBuf：协议解析器的接收数据缓存区
num1.val = bytesToInt(protocol5.rxBuf,11，1);
取出字节字节数组中的整数，赋给整数控件显示（从十一位开始，大端模式）

脚本指令：提取浮点数函数 - bytesToFloat

介绍

- 从字节数组中提取浮点数。

定义

- float bytesToFloat(byte array[], int start, int mode);

说明：
byte array[]：字节数组
int start：起始位置为 start
mode：大小端。0 为小端模式，1为大端模式
配合上位机协议解析器控件使用。
举例：
protocol5.rxBuf：协议解析器的接收数据缓存区
numf1.val = bytesToFloat(protocol5.rxBuf,11, 1);
取出字节字节数组中的浮点数，赋给浮点数控件显示（从十一位开始，，大端模式）

脚本指令：提取短整数函数 - bytesToShort

介绍

- 从字节数组中提取短整数

定义

- int bytesToShort(byte array[], int start, int mode);

说明：
byte array[]：字节数组
int start：起始位置为 start
mode：大小端。0 为小端模式，1为大端模式
配合上位机协议解析器控件使用。
举例：
protocol5.rxBuf：协议解析器的接收数据缓存区
num1.val = bytesToShort(protocol5.rxBuf,11, 1);
取出字节字节数组中的短整数，赋给整数控件显示（从十一位开始，大端模式）

脚本指令：提取无符号短整数函数 - bytesToUshort

介绍

- 从字节数组中提取无符号短整数

定义

- int bytesToUshort(byte array[], int start, int mode);

说明：
byte array[]：字节数组
int start：起始位置为 start
mode：大小端。0 为小端模式，1为大端模式
配合上位机协议解析器控件使用。
举例：
protocol5.rxBuf：协议解析器的接收数据缓存区
num1.val = bytesToUshort(protocol5.rxBuf,11, 1);
取出字节字节数组中的无符号短整数，赋给整数控件显示（从十一位开始，大端模式）

脚本指令：用户flash写入整数函数 - fwInt

介绍

- 用户可设置使用一块自定义大小的flash空间，用作掉电保存

定义

- void fwInt(int offset, int val);

说明：
offset：flash中的偏移（可理解为空间存储数据的起始位置）
val：要写入的整数
内部变量控件在掉电后，其所存的变量数据会丢失，此函数会将数据写入用户自己设置的一块flash空间中，用作掉电保存。
使用：
VP软件左上菜单栏->工具->项目设置->最底部的用户Flash大小，最大空间是512字节，设置一个合适的大小->保存；
举例：

```
fwInt(0, 100);      //写入整数100
fSave();            //将数据保存到flash中
```

将数据写入到flash中，这两个函数必须同时使用，才能将数据写入到flash空间中并保存；
备注：如果offset小于0或大于flash的尺寸，那么不执行写入

脚本指令：用户flash读取整数函数 - frInt

介绍

- 读取flash中已存入的整数数据

定义

- void frInt(int offset);

说明：
offset：flash中的偏移（可理解为空间存储数据的起始位置）
此函数用以读取已写入保存的数据。
举例：

```
num1.val=frInt(0);
//读取flash空间中的整数并赋给整数控件num1显示（偏移数需和写入的偏移一致）
```

此函数和上一页的写入函数配合使用，即可完成数据读写的整个操作；

备注：如果offset小于0或大于flash的尺寸，那么返回0

脚本指令：用户flash写入浮点数函数 - fwFloat

介绍

- 用户可设置使用一块自定义大小的flash空间，用作掉电保存

定义

- void fwFloat(int offset, float val);

说明：
offset：flash中的偏移（可理解为空间存储数据的起始位置）
val：要写入的浮点数
内部变量控件在掉电后，其所存的变量数据会丢失，此函数会将数据写入用户自己设置的一块flash空间中，用作掉电保存。
使用：
VP软件左上菜单栏->工具->项目设置->最底部的用户Flash大小，最大空间是512字节，设置一个合适的大小->保存；
举例：

```
fwFloat(0, 10.0);    //写入浮点数10.0
fSave();             //将数据保存到flash中
```

将数据写入到flash中，这两个函数必须同时使用，才能将数据写入到flash空间中并保存；
备注：如果offset小于0或大于flash的尺寸，那么不执行写入

脚本指令：用户flash读取浮点数函数 - frFloat

介绍

- 读取flash中已存入的浮点数数据

定义

- void frFloat(int offset);

说明：
offset：flash中的偏移（可理解为空间存储数据的起始位置）
此函数用以读取已写入保存的数据。
举例：

```
numf1.valf=frFloat(0);
//读取flash空间中的浮点数并赋给浮点数控件numf1显示（偏移数需和写入的偏移一致）
```

此函数和上一页的写入函数配合使用，即可完成数据读写的整个操作；

备注：如果offset小于0或大于flash的尺寸，那么返回0

脚本指令：用户flash写入字符串函数 - fwString

介绍

- 用户可设置使用一块自定义大小的flash空间，用作掉电保存

定义

- void fwString(int offset,string val);

说明：
offset：flash中的偏移（可理解为空间存储数据的起始位置）
val：要写入的字符串
内部变量控件在掉电后，其所存的变量数据会丢失，此函数会将数据写入用户自己设置的一块flash空间中，用作掉电保存。
使用：
VP软件左上菜单栏->工具->项目设置->最底部的用户Flash大小，最大空间是512字节，设置一个合适的大小->保存；
举例：
fwString(0，Good)； //写入字符串Good
fSave()； //将数据保存到flash中
将数据写入到flash中，这两个函数必须同时使用，才能将数据写入到flash空间中并保存；
备注：如果offset小于0或大于flash的尺寸，那么不执行写入

脚本指令：用户flash读取字符串函数 - frString

介绍

- 读取flash中已存入的字符串数据

定义

- void frString(int offset);

说明：
offset：flash中的偏移（可理解为空间存储数据的起始位置）
此函数用以读取已写入保存的数据。
举例：

```
text1.txt=frString(0);
//读取flash空间中的字符串并赋给文本控件text1显示（偏移数需和写入的偏移一致）
```

此函数和上一页的写入函数配合使用，即可完成数据读写的整个操作；

备注：如果offset小于0或大于flash的尺寸，那么返回0

脚本指令：用户flash写入字节数组函数 - fwBytes

介绍

- 用户可设置使用一块自定义大小的flash空间，用作掉电保存

定义

- `void fwBytes(offset, byte[] array, int start, int len);`

说明：
offset：flash中的偏移
array：写入数据的字节数组
start：需要保存的数据的起始位置
len：要保存的数据长度
使用：
VP软件左上菜单栏->工具->项目设置->最底部的用户Flash大小，最大空间是512字节，设置一个合适的大小->保存；
举例：

```
byte a[5]={0x01,0x02,0x03,0x04,0x05};
fwBytes(0,a,0,5);
//写入偏移为0，名称为a，起始位置为0，长度为5的字节数组
fSave();
//将数据保存到flash中
```

备注：如果offset小于0或大于flash的尺寸，那么不执行写入

脚本指令：用户flash读取字节数组函数 - frString

介绍

- 读取flash中已存入的字符串数据

定义

- void frBytes(offset, byte[] array, int start, int len);

说明：
offset：flash中的偏移
array：读取数据的字节数组
start：需要保存的数据的起始位置
len：要读取的数据长度
此函数用以读取已写入保存的数据。
举例：

```
byte a[1]={};
frBytes(0,a,0,1);
num2.val = a[1];
//读取flash空间中的字节数组a[1]并赋给整数控件num2显示（偏移数需和写入的偏移一致）
```

此函数和上一页的写入函数配合使用，即可完成数据读写的整个操作；

备注：如果offset小于0或大于flash的尺寸，那么返回0

脚本指令：用户flash写入保存函数 - fSave

介绍

- 保存用户写入的数据到flash中

定义

- void fSave();

说明：
此函数无参数，用来保存要写入flash空间的数据。
此函数必须写在写入函数之后；

举例：
在不同空间段存储不同数据

```
fwInt(0, 100);
fwFloat(100, 10.0);
fwString(200, Good);
fSave();
```

需要写入保存多条不同类型数据时，只需在最后写一条保存函数（考虑到flash的擦写次数有限）

脚本指令：CAN口发送字节数组函数 - canSendBytes

介绍

- CAN版本屏幕中，CAN口发送字节数组函数

定义

- void canSendBytes(int id, byte[] array, int len, int fmd);

说明：
id：目标id
array：保存发送数据的字节数组
len：发送的数据长度
fmd：帧模式

举例：

```
byte a[8]={0x65,0x66,0x67,0x68,
0x69,0x6a,0x6b,0x6c};
canSendBytes(0x12345678,a,7,2);
//发送id为0x12345678，名称为a,长度为7，帧模式为2（拓展数据帧）的字节数组
```