

SDWb_Lua 动态修改变量文本颜色案例说明

VGUS 串口屏上变量的显示格式都是固定的,是由 VGUS 上位机软件设置好并下载到 VGUS 串口屏中的。在串口屏显示过程中,如果用户需要临时修改变量显示格式,则可以通过描述指针实现。本案例介绍了如何通过 Lua 脚本操作描述指针动态修改变量的文本颜色。整个过程无需单片机参与、无需串口指令控制。

一. 案例功能介绍

1.1 界面功能

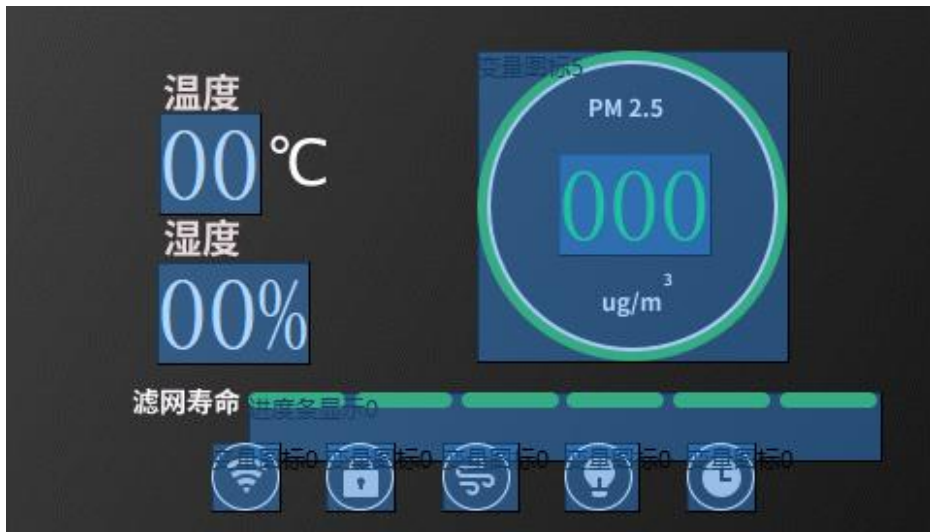


图 1 空气净化器主界面

如图 1 所示,本案例的界面工程是应用于空气净化器上的。主界面右上方的圆形背景框内有一个 3 位整数,用于显示当前 PM2.5 浓度。为了更直观的展现当前 PM2.5 浓度,图 1 中的圆形背景框的颜色、PM2.5 浓度数值显示颜色会根据当前显示的 PM2.5 浓度数值改变而改变,如表 1 所示。

表 1 PM2.5 浓度对与空气质量等级对应关系

PM2.5 浓度	空气质量等级	背景框颜色、PM2.5 数值颜色
$PM2.5 \leq 35$	优	绿色
$35 < PM2.5 \leq 75$	良	蓝色
$75 < PM2.5 \leq 115$	轻度污染	黄色
$115 < PM2.5 \leq 150$	中度污染	橙色
$150 < PM2.5$	重度污染	红色

下文将详细介绍如何通过 Lua 脚本操作描述指针、动态修改文本显示颜色,

实现上述 PM2.5 浓度的显示效果。

1.2 控件设计

如图 1 所示，本案例用到了三个数据变量控件分别显示温度、湿度、PM2.5 的数值。PM2.5 数值对应的的数据变量控件下方有一个变量图标控件，用来显示一个透明的图标，图标内容为一个圆形，用作 PM2.5 数值显示的背景框。



图 2 PM2.5 数据变量控件属性



图 3 背景框变量图标控件属性

PM2.5 对应的数据变量控件属性如图 2 所示，其变量地址设置为 0x0001，变量类型为 unsigned int 型，3 位整数，采用无效位补零的显示模式。背景框变量图标控件属性如图 3 所示，变量地址设置位 0x000A，变量下限上限范围位 0-4，分别对应 13-17 号图标，其图标样式如图 4 所示。

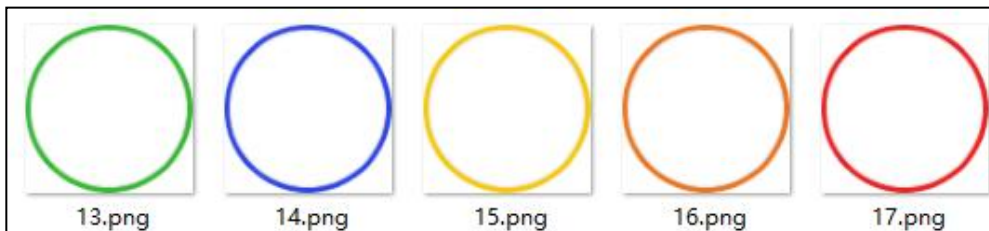


图 4 背景框图标

动态修改数据变量文本颜色需要通过描述指针实现，所以 PM2.5 数值对应的数据变量显示控件需要设置描述指针，描述指针的使用方法可以参考《VGUS

串口屏用户开发指南》。由图 2 可知，该控件的描述指针设置为 0x4000，如图 5 所示，通过查询《VGUS 串口屏用户开发指南》中的“表 6-3-8”可知数据变量文本颜色对应的描述指针偏移量为 0x03，所以存放数据变量文本颜色属性的变量地址为 0x4000 加上偏移量 0x03，也就是 0x4003。往 0x4003 里写入 16 位颜色数据就可以修改数据变量的文本颜色。同时修改背景框颜色需要往地址 0x000A 里写入上下限范围内的变量值，来显示不同背景图标从而达到修改背景框的颜色的目的。

0x03	Color	2	显示颜色
------	-------	---	------

图 5 数据变量文本颜色偏移量

1.3 流程设计

本案例的流程图如图 6 所示，串口屏上电后在 Lua 脚本初始化回调函数下开启定时器。在定时器超时回调函数中周期性读取 PM2.5 对应地址的数值，将该数值赋值给定义的变量 var1，然后通过 if 语句判断 var1 的数值大小，确定当前的空气质量等级。当前空气质量等级用变量 level1 表示，level1 数值的范围为 1-5，分别对应优、良、轻度污染、中度污染、中度污染五个空气质量等级。接下来判断当前空气质量等级与上一次读取的空气质量等级是否相同，这里定义一个变量 temp1 表示上一次读取到的空气质量等级，当 level1 不等于 temp1 时，则根据空气质量等级来改变数据变量的文本颜色，以及背景框的颜色，并将当前 level1 的值赋值给 temp1。level1 等于 temp1 时不执行改变数据变量文本颜色和背景框颜色的操作，等待下一个周期再次读取 PM2.5 数值。

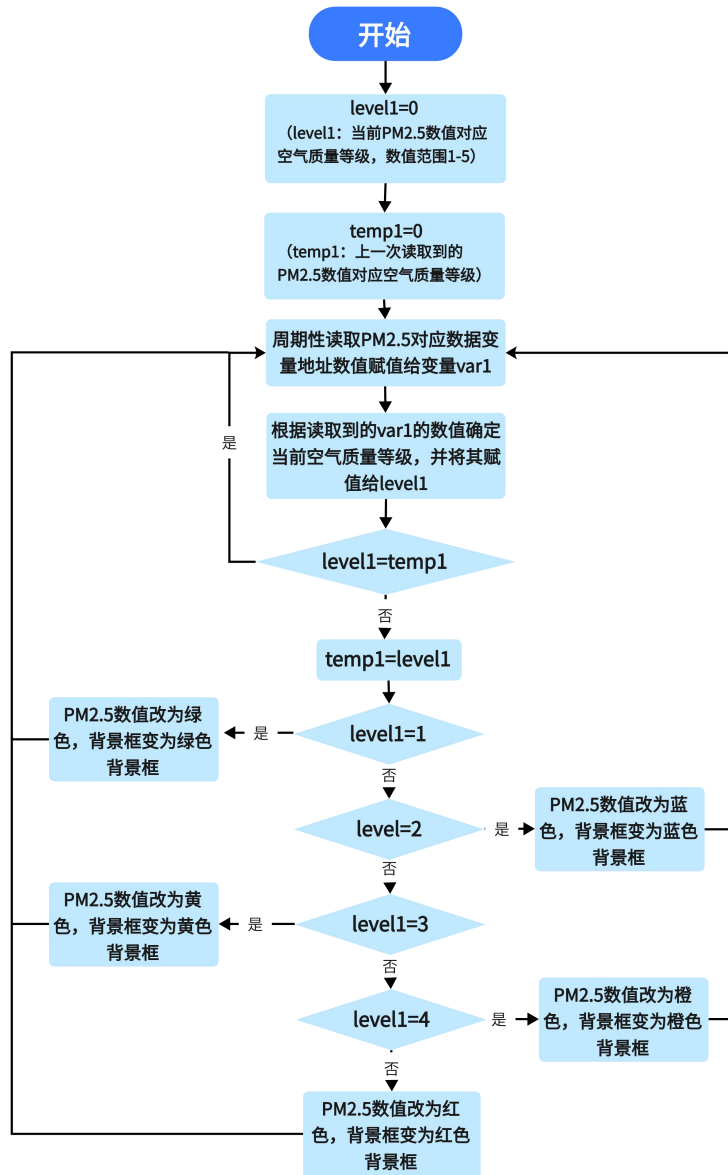


图 6 动态修改变量文本颜色流程图

二. 初始化回调函数

本案例初始化回调函数主要用于设置串口参数以及开启定时器。

```

-- 初始化回调函数
function callback_init()
    -- 设置串口屏模式为VGUS 协议, 波特率 115200, 串口格式 8N1
    com_set_work_mode(0,0,115200,4)
    -- 开启定时器 0
    vgus_timer_start(0,1,0,120)
end
    
```

本案例也可以采用默认串口设置, 不设置串口参数。因为上电后串口屏需要周期性读取 0x0001 地址的 PM2.5 数值, 并根据该数值来判断是否改变 PM2.5

数值颜色、背景框颜色。所以这里开启定时器 0，定时模式为周期模式，计数方式为向上计数，定时器超时时间设置为 120ms，开启定时器后串口屏将以 120ms 的周期不断的去读取 PM2.5 数值，为了尽可能让 PM2.5 数值与文本颜色、背景框颜色同步刷新，屏参配置中的运行周期与定时器超时时间保持一致，如图 7 所示，运行周期也设置为 120ms。运行周期就是变量的刷新周期，运行周期越小变量刷新越快，处理变量能力越弱。运行周期越大，变量刷新越慢，处理变量能力越强。用户可以自行调整，设置适合自己工程的运行周期。

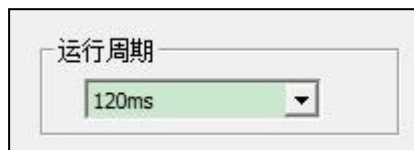


图 7 运行周期设置

三. 定时器回调函数

当开启定时器后，串口屏通过 Lua 脚本调用定时器回调函数 `callback_timer(timer_id)`，在回调函数下使用 `if` 语句来判断开启的是哪一个定时器，满足条件后开始执行嵌套在 `if` 语句里的代码。

```
-- 打开定时器 0 执行以下代码
if timer_id==0 then
```

3.1 读取 PM2.5 当前显示数值

打开定时器后首先读取当前 PM2.5 对应的数据变量地址的数值

```
-- 读取 PM2.5 对应地址数值
Var1=vgus_vp_var_read(0x0001,5)
```

通过以数值形式读取变量存储器函数 `vgus_vp_var_read(vp_addr, var_type)` 读取 0x0001 地址数值，变量类型为无符号整型，并将该数值赋值给定义的一个变量 `var1`。

3.2 确定 PM2.5 当前数值对应的空气质量等级

对变量 `var1` 数值大小进行判断，确定当前的空气质量等级。

```
-判断当前空气质量等级：空气质量等级为优
if var1<=35 then
    level1=1
-- 空气质量等级为良
elseif var1<=75 then
    level1=2
-- 空气质量为轻度污染
elseif var1<=115 then
```

```

    level1=3
    -- 空气质量等级为中度污染
elseif var1<=150 then
    level1=4
    -- 空气质量等级为重度污染
else
    level1=5
end

```

这里通过 if 语句对变量 var1 值进行判断，确定当前的空气质量等级 level1 的数值。

3.3 判断当前空气质量等级是否和上一次读取到的空气质量等级相同

获取到当前空气质量等级 level1 的值后，判断 level1 是否等于 temp1。

```

-- 判断当前空气质量是否和上一次读取时空气质量等级相同，不相同执行以下代码：
if level1~=temp1 then

```

3.3.1 当前空气质量等级和上一次读取到的空气质量等级相同

当前空气质量等级和上一次读取的空气质量等级相同则不需要执行修改数据变量文本颜色以及文本框颜色的操作，等待下一个周期读取到空气质量等级后再做比较，此时仍然显示上一次刷新的颜色，这样有利于减小 Lua 程序的任务量。

3.3.2 当前空气质量等级和上一次读取到的空气质量等级不相同

当前空气质量等级和上一次读取的空气质量等级不相同同时执行一下代码：

```

--temp1 的值修改为当前空气质量等级
temp1=level1
-- 当前空气质量等级为优，数据变量颜色显示绿色，圆形背景框修改为绿色图标
if level1==1 then
    vqus_vp_var_write(0x4003,5,0x07F0)
    vqus_vp_var_write(0x000A,5,0)
-- 当前空气质量为良，数据变量颜色显示蓝色，圆形背景框修改为蓝色图标
elseif level1==2 then
    vqus_vp_var_write(0x4003,5,0x041F)
    vqus_vp_var_write(0x000A,5,1)
-- 当前空气质量为轻度污染，数据变量颜色显示黄色，圆形背景框修改为黄色图标
elseif level1==3 then
    vqus_vp_var_write(0x4003,5,0xFFE0)
    vqus_vp_var_write(0x000A,5,2)
-- 当前空气质量等级为中度污染，数据变量颜色显示橙色，圆形背景框修改为橙色图标
elseif level1==4 then
    vqus_vp_var_write(0x4003,5,0xFC08)
    vqus_vp_var_write(0x000A,5,3)

```

```
-- 当前空气质量等级为重度污染，数据变量颜色显示红色，圆形背景框修改为红色图标  
else  
    vqus_vp_var_write(0x4003,5,0xF800)  
    vqus_vp_var_write(0x000A,5,4)  
end
```

当检测到空气质量等级发生变化后，首先将读取到的当前空气质量等级赋值给 temp1，因为下一次读取空气质量等级的时候应当与此次读取到的空气质量等级做比较。接下来根据当前空气等级来改变 PM2.5 数值颜色，以及背景框颜色，当前空气等级为优时，往 0x4003 里写入 0x07F0,此时 PM2.5 数值显示为绿色，往 0x000A 里写入变量 0 对应 13 号图标。当前空气质量等级为良时，往 0x4003 里写入 0x041F,此时 PM2.5 数值显示为蓝色，往 0x000A 里写入变量值 1 对应 14 号图标。当前空气质量等级为轻度污染时，往 0x4003 里写入 0xFFE0,此时 PM2.5 数值显示为黄色，往 0x000A 里写入变量值 2 对应显示 15 号图标。当前空气质量等级为中度污染时，往 0x4003 里写入 0xFC08,此时 PM2.5 数值显示为橙色，往 0x000A 里写入变量值 3 对应 16 号图标。当前空气质量等级为重度污染时，往 0x4003 里写入 0xF800,此时 PM2.5 数值显示为红色，往 0x000A 里写入变量 0 对应 17 号图标。

以上就是实现动态修改变量文本颜色的基本步骤，官网可以下载完整的案例工程，包括 Lua 脚本代码和界面工程。控件的使用说明、寄存器功能说明都可以参考文档《VGUS 串口屏用户开发指南》。Lua 脚本函数可以参考文档《基于 VGUS 的 Lua 脚本使用说明》。因为虚拟串口屏不支持描述指针功能，所以该例程无法用虚拟串口屏模拟。可以通过 USB 转串口工具，用电脑串口调试助手发送 0x82 指令往 0x0001 地址写入数据来验证该案例的功能。