SDWb_Lua 数据库掉电保存案例说明

SDWb 串口屏提供有 64K 字节的数据库，用于掉电保存用户数据。用户通过串口指令写寄存器 0x56-0x5F 实现访问（读/写）该数据库，可以将变量存储器指定位置的数据写入保存到数据库里，也可以将数据库里面的指定位置数据读出加载到变量存储器里。详细介绍参见《VGUS 串口屏用户开发指南》3.2.12 节。

本案例给出了使用 Lua 脚本写寄存器 0x56-0x5F，实现访问（读/写）数据库的目的，将界面中设置的参数保存到数据库中，整个过程不需要用户串口指令配合。

一．案例功能介绍

1.1 界面功能

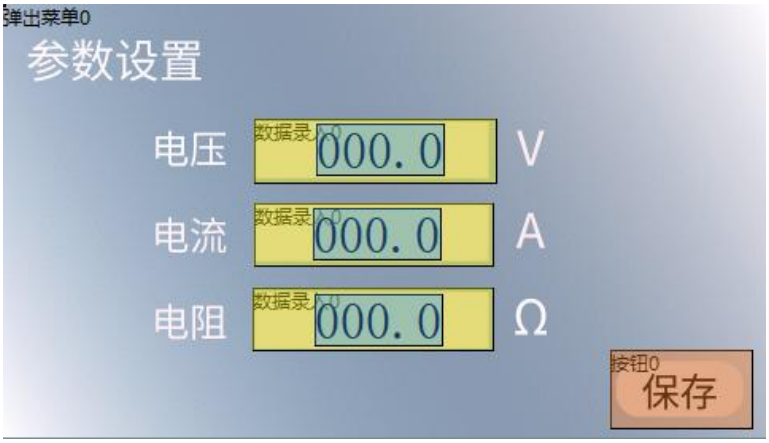

图 1 参数设置界面

参数设置界面如图 1 所示，该界面中有电压、电流、电阻三组参数需要设置，设置完成后点击“保存”按钮，将设置好的参数保存到数据库的指定位置。

点击“保存”按钮，有两种方法可以将设置好的参数保存到数据库的指定位置。一种方法是用户单片机通过串口写寄存器 0x56-0x5F。另一种方法是通过 Lua 脚本编程写寄存器 0x56-0x5F。后一种方法无需用户单片机参与、无需发送串口指令。下文将重点讨论后一种方法的实现方法。

1.2 控件设计

在图 1 中，电压、电流和电阻三组参数的录入，需要使用数据变量控件和数据录入控件组合实现。使用数据录入控件录入数据，再通过数据变量控件将录入的数据显示出来，此过程不需要 lua 脚本参与。该案例中，电压、电流和电阻三个变量的地址分别设置为 0x0001，0x0002，0x0003。

“保存”按钮用于触发保存操作，此按钮需要设置按键键码，以便在触摸事件发生时，能够执行 Lua 中的触摸回调函数，通过页面编号以及按键键码，可以在 Lua 代码中定位到该按钮控件。保存按钮的按键键码设置如图 2 所示。

弹出菜单控件用于弹出保存成功的提示框。此控件同样需要设置按键键码，但是功能与保存按钮完全不同，该按键键码用于软件方式触发弹出菜单功能。弹出菜单的按键键码设置如图 3 所示。

图 2　保存按钮键码值

图 3　弹出菜单键码值

1.3 流程设计

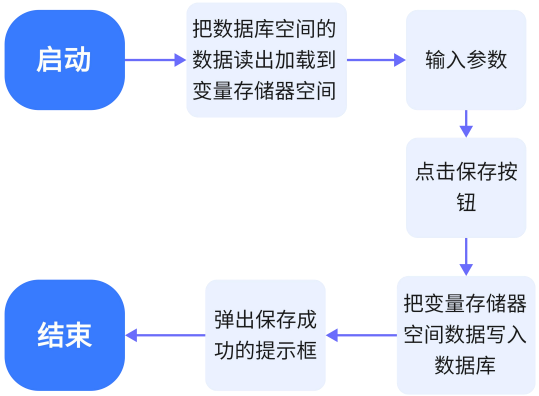

图 4　数据库掉电保存程序流程图

案例功能实现流程如图 4 所示。串口屏上电时在初始化回调函数中执行“将数据库空间数据加载到变量存储器空间”功能，目的是将当前存储在数据库中的之前设置的电压、电流、电阻分别读出并加载到变量存储器 0x0001、0x0002、0x0003 单元里。当串口屏进入图 1 所示的参数设置界面中时，就能够自动将上次

设置值在相应录入框中显示出来。

电压、电流、电阻参数设置好后，点击保存按钮，将在 Lua 脚本中触发触摸回调函数。在该回调函数中，首先通过对寄存器 0x56-0x5F 赋值，完成“将变量存储器空间数据写入数据库”的功能；然后再通过对寄存器 0x4F 赋值，弹出提示保存成功的弹出菜单。

二．初始化回调函数

该案例中，初始化回调函数主要用于设置串口相关参数，以及将当前存储在数据库中的之前设置的电压、电流、电阻分别读出并加载到变量存储器 0x0001、0x0002、0x0003 单元里。

```
--初始化回调函数
function callback_init()
  --将串口 0 设置为 VGUS 协议，波特率 115200，串口格式 8N1
   com_set_work_mode(0,0,115200,4)
   --通过操作寄存器 0x56-0x5F，将数据库空间数据加载到变量存储器空间
   vgus_reg_write (0x56, 10, database_table)
end
```

在 Lua 脚本程序开头首先定义一个数据表，名称 database_table，并给数据表的第 1 到第 10 个元素赋值，其代码如下：

```
--定义读写用户数据库寄存器指令初始数据，0xA0：把用户数据库空间数据加载到变量存储器空间
Local database_table={0x5A,0xA0,0x00,0x00,0x00,0x01,0x00,0x01,0x00,0x03}
```

通过写寄存器函数 vgus_reg_write(reg_addr, write_len, write_table)将“database_table”的第 1 到第 10 个元素数据 database_table[1]~database_table[10]写入到寄存器 0x56-0x5F 当中。对照 0x56-0x5F 寄存器功能说明，我们可以获取到该条指令的含义：将数据库 0x00000001-0x00000003 单元里的数据读出并加载到变量存储器的 0x0001-0x0003 单元里。其中电压、电流和电阻的变量地址分别为 0x0001、0x0002、0x0003。当串口屏进入图 1 所示的参数设置界面中时，就能够自动将上次设置值在相应录入框中显示出来。

三．触摸回调函数

点击保存按钮，如图 2 所示，因为保存按钮设置了“按键键码”1，当点击保存按钮，串口屏会通过 Lua 脚本调用触摸回调函数 callback_touch(pic_id,key_c

ode,touch_state)，在回调函数下用 if 语句来判断是否触发该按键，当满足条件时开始执行嵌套在 if 语句里的代码。

```
--“保存”按钮点击并抬起时执行以下内容
if  pic_id==0 and key_code==1 and touch_state==2 then
```

3.1 把变量地址里数据加载到数据库空间

当“保存”按钮点击并抬起，将执行以下代码：

```
--“保存”按钮点击并抬起时执行以下内容
database_table[2]=0x50
--通过操作寄存器 0x56-0x5F，将数据库空间数据写入变量存储器空间
vgus_reg_write (0x56, 10, database_table)
```

这里将数据表 database_table 的第 2 个元素的数据改为 0x50，通过写寄存器函数 vgus_reg_write(reg_addr,write_len,write_table) 再将 database_table 的第 1 到第 10 个元素的数据写入到 0x56-0x5F 寄存器中。对照 0x56-0x5F 寄存器功能说明可知，该语句执行后，会将变量存储器 0x0001-0x0003 单元里的数据写入到数据库 0x00000001-0x00000003 单元里，完成将录入数据保存到数据库里的目的。

3.2 触发弹出菜单提示保存成功

如图 3 所示，弹出菜单按键键码为 2，数据保到数据库文件后，执行以下代码：

```
--弹出菜单键码值为 2
reg_table[1]=2
--通过操作 0x4F 寄存器触发弹出菜单
vgus_reg_write(0x4F,1,reg_table)
```

这里给 reg_table 第 1 个元素 reg_table[1]赋值 2，并把这个数据通过写寄存器函数 vgus_reg_write(reg_addr, write_len, write_table)写入到控制按键触发的寄存器 0x4F 中，从而触发弹出菜单，提示保存成功。

以上是实现掉电保存的基本步骤，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》 。