

自定义串口协议在 SDWb 串口屏上的实现方法

——Lua 入门概述 3

SDWb 系列串口屏通过 Lua 脚本编程功能，可以实现用户自定义的串口指令协议，方便与各种类型的用户设备连接。串口屏既可以做系统主机，也可以做系统从机。

一. 串口工作模式设置 API 函数

通过下述 API 接口函数，设置串口工作模式。正确设置串口号、以及串口工作协议等。串口 0 默认为 VGUS 指令集协议，可以配置为自定义串口协议。串口 1 仅支持自定义串口协议。

```
com_set_work_mode(com_num, work_protocol, baudrate, format)
```

com_num: 串口编号。编号范围 0-1。

work_protocol: 设置工作协议。

0: VGUS 指令集协议。

1: 自定义串口协议。

2: ModbusRTU 从机。

3: ModbusASCII 从机。

4: ModbusRTU 主机。

5: ModbusASCII 主机。

baudrate: 设置串口波特率。

com_format: 设置串口格式。

二. 串口发送自定义数据指令

Lua 脚本在自定义串口协议下，串口发送自定义数据，可以通过串口发送 API 接口函数实现。

```
com_data_send (com_num , send_len, send_table)
```

com_num: 串口编号。编号范围 0-1。

send_len: 发送长度，单位字节。单次最大 1024。

send_table: 待发送数据。字节数组，从 1 开始索引。

返回值: 无。

该 API 接口函数在 VGUS 协议下也可以使用，发送自定义的串口数据。

三. 串口接收自定义数据指令

在自定义串口协议下，SDWb 串口屏为每个串口分别开辟了一个 4K 字节的接收缓冲区，所有接收到的串口数据都暂存在该缓冲区中。Lua 脚本必须及时读取接收缓冲区中的数据。



为使 Lua 脚本在自定义串口协议下读取串口接收缓冲区中的数据，SDWb 串口屏专门提供了一条串口回调函数、二条串口接收 API 接口函数。

表 1 自定义串口协议专用回调函数（仅自定义串口协议支持）

类型	说明	函数名称
串口回调函数	当串口接收缓冲区中有接收到数据时，执行该串口回调函数。	callback_uart(com_num, recv_len)

表 2 自定义串口协议专用 API 函数（仅自定义串口协议支持）

类型	说明	函数名称
串口 API 函数	获取串口缓冲区中的字节数	com_data_len(com_num)
	读取串口缓冲区中的数据	com_data_read(com_num, read_len, read_table)

callback_uart(com_num, recv_len)

com_num: 串口编号。

recv_len: 对应串口缓冲区中可以读取的字节数。**该参数是 VGUS 应用程序赋值。**

当串口接收缓冲区中有接收到的数据时，串口屏执行串口回调函数。用户在串口回调函数中编写 Lua 脚本程序，通过调用 API 接口函数 com_data_read(com_num, read_len, read_table)，读取串口缓冲区中的数据，并对读取到的数据加以解析和应用。

在调用 com_data_read(com_num, read_len, read_table) 函数读取串口缓冲区中的数据时，可以将读取长度统一默认用最大 1024，然后根据函数返回值来确认实际读取到的字节数。这样就可以不用事先去获取缓冲区中的字节数，一次性将缓冲区中的数据全部读出来（假设缓冲区中的数据不多于 1024 字节）。

当然，也可以通过串口回调函数的参数 recv_len，或者调用 API 接口函数 com_data_len(com_num)，获取到串口缓冲区中待读取数据的字节数。再调用 API 接口函数 com_data_read(com_num, read_len, read_table)，实现部分或全部读取缓冲区中的数据。

在使用自定义串口协议时，不建议在触摸回调函数中读取串口数据，因为触摸回调函数执行的不确定性，可能由于未能及时读出串口缓冲区数据造成数据丢失的问题。

参考文档：《基于 VGUS 的 Lua 脚本使用说明》

参考文档：《 Lua 入门概述 1——Lua 脚本在 SDWb 串口屏上可以实现哪些功能》

