



## 自定义协议使用

深圳市欣瑞达液晶显示技术有限公司



欣瑞达液晶



400-069-8808 0755-26018666



广东省深圳市南山区高新园北松坪山路5号嘉达研发大厦A座3层



[www.xrd-lcd.com](http://www.xrd-lcd.com)



使用淘宝扫一扫  
欣瑞达官方旗舰店



使用微信扫一扫  
欣瑞达官方微信

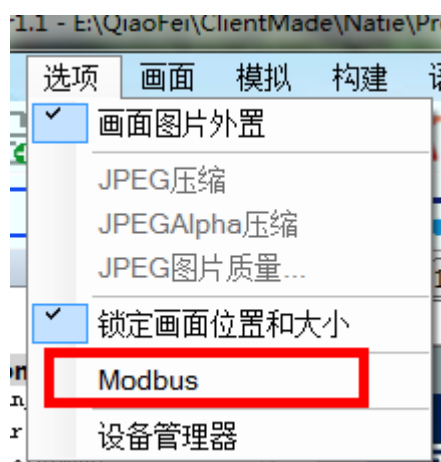
## 1. 文档概述

本文档主要讲解自定义协议的使用。

## 2. 使用

1.自定义协议是指除标准协议与 modbus 协议之外的指令格式

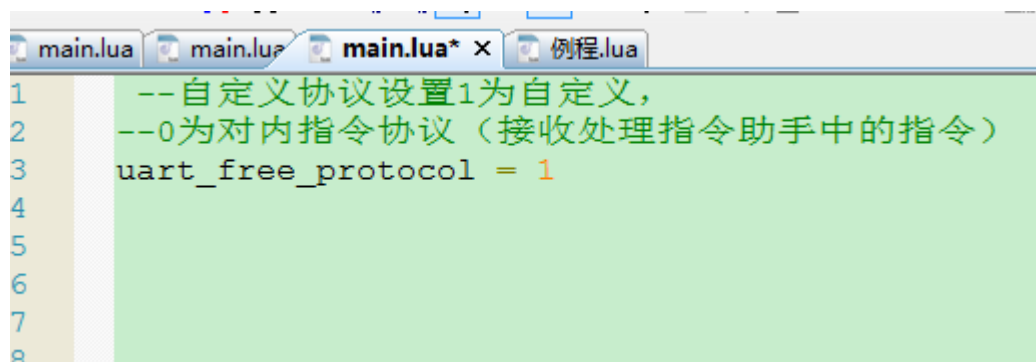
2.需关闭 modbus 指令，如下图。前面的勾去掉



3.打开 lua 脚本，所以功能由 lua 实现。找到工程文件下的 main.lua.没有则手动创建。

bk	2021/12/13 16:06	文件夹	
images	2021/12/20 18:06	文件夹	
layers	2021/12/15 10:01	文件夹	
outputs	2021/12/23 16:42	文件夹	
sounds	2021/11/16 10:18	文件夹	
videos	2021/11/16 10:18	文件夹	
config.ini	2021/12/23 16:44	配置设置	1 KB
ids.xml	2021/12/23 16:44	XML 文件	9 KB
main.lua	2021/12/23 16:44	LUA 文件	44 KB
Project.tftp	2021/12/23 16:44	TFTP 文件	28 KB
ts.conf	2021/12/23 16:44	CONF 文件	1 KB

4.设置使用自定义指令格式。将 uart\_free\_protocol 变量设置为 1

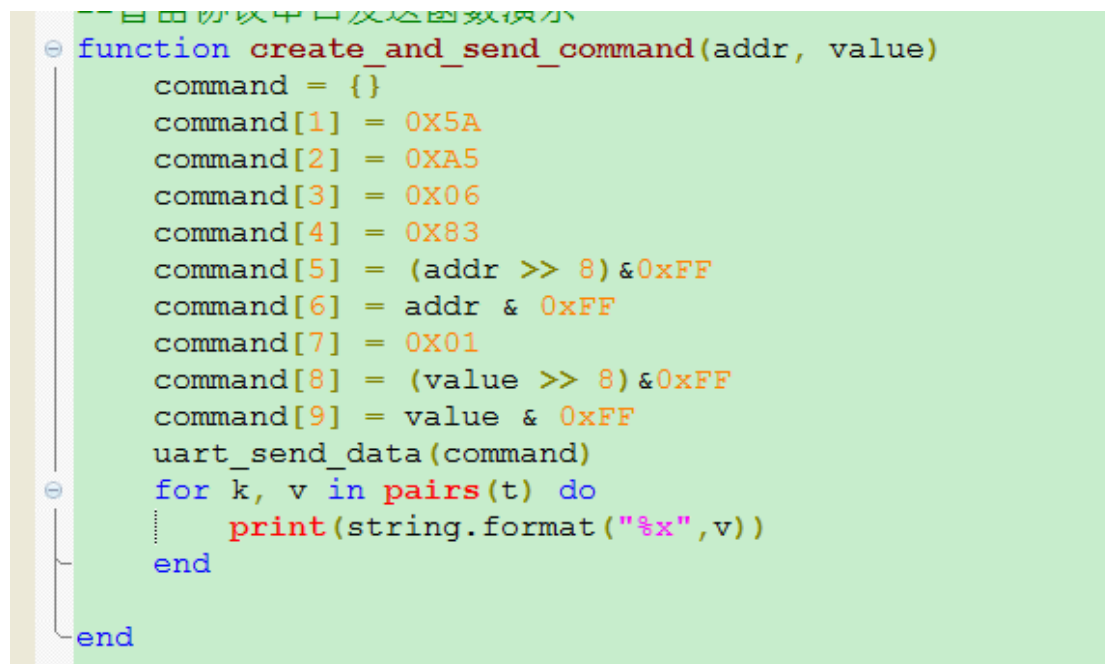


```
1  --自定义协议设置1为自定义，
2  --0为对内指令协议（接收处理指令助手中的指令）
3  uart_free_protocol = 1
4
5
6
7
8
```

5.定义指令的发送

1. uart\_send\_data ( ) 为发送函数，需要传入一个 table,按指定格式放入数据。

演示的帧格式为 帧头+长度+功能+地址+数据长度+值



```
--自定义协议串口屏发送函数演示
function create_and_send_command(addr, value)
    command = {}
    command[1] = 0X5A
    command[2] = 0XA5
    command[3] = 0X06
    command[4] = 0X83
    command[5] = (addr >> 8) & 0xFF
    command[6] = addr & 0xFF
    command[7] = 0X01
    command[8] = (value >> 8) & 0xFF
    command[9] = value & 0xFF
    uart_send_data(command)
    for k, v in pairs(t) do
        print(string.format("%x", v))
    end
end
```

6. on\_uart\_recv\_data ( ) 为自定义协议接收回调函数，格式解析这里面实现。

注意 packet 数组的第一个数据下标是 0。

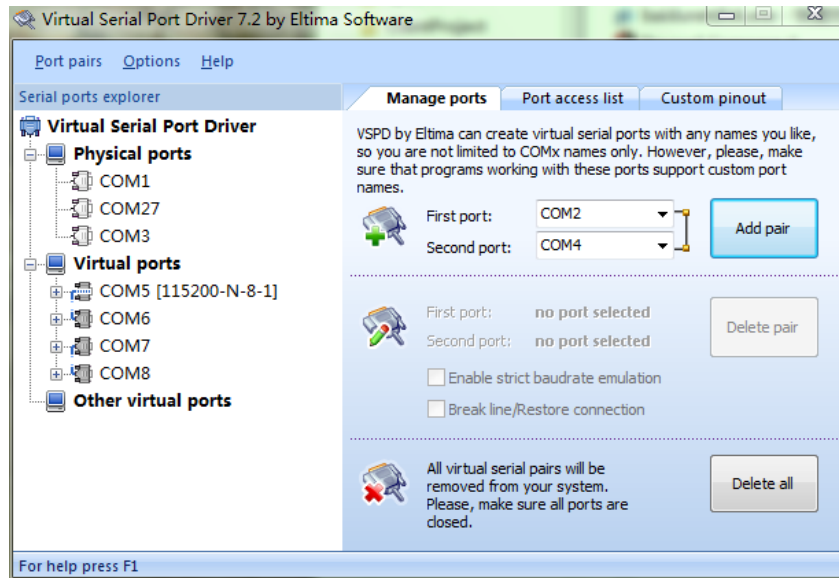
```
function on_uart_recv_data(packet)
    table.insert(packet,0,0)
    offset = 0
    leng = #packet
    while(1) do
        packet2 = { }
        --print(packet[offset],offset,packet[offset + 1])
        if packet[offset] == 0x5A and packet[offset + 1] == 0xA5 then
            --定义头
            -- print("hellp1")
            if leng >= (packet[2 + offset] + 2) then
                repeat_data()

                for i=1,packet[2 + offset]+1,1 do
                    packet2[i] = packet[2 + offset +i-1]
                end

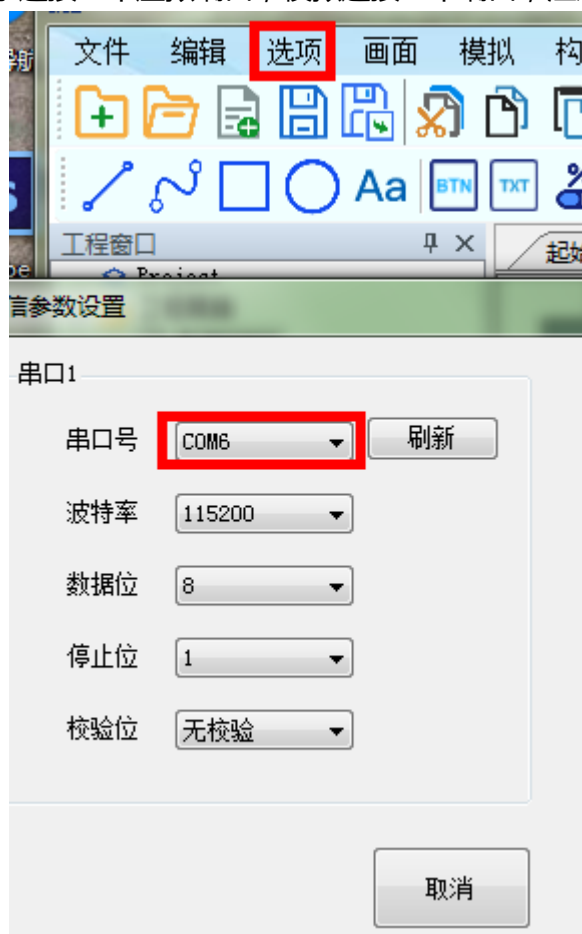
                process_recv_data(packet2)
                --print("hellp2")
                offset = offset + packet[2 + offset] + 3
            else
                --print("error_length")
                return
            end
        else
            offset = offset +1
            if packet[offset] == nil then
                return
            end
        end
    end
end
end
```

## 7. 电脑模拟调试

### 1. 先创建虚拟串口



2.再用串口助手连接一个虚拟端口，模拟连接一个端口，注意要连接成对的串



口。

