

# 4G 网关 XY\_3422 配置工具使用方法

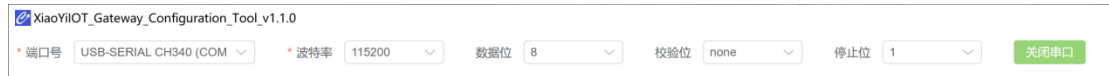
## 一、使用须知

XY\_3422 型号的网关包含 3 个 RS485 串口, 3 个串口的默认波特率均为 115200, 无校验。此外还可以通过 USB 端口进行配置。

## 二、配置步骤

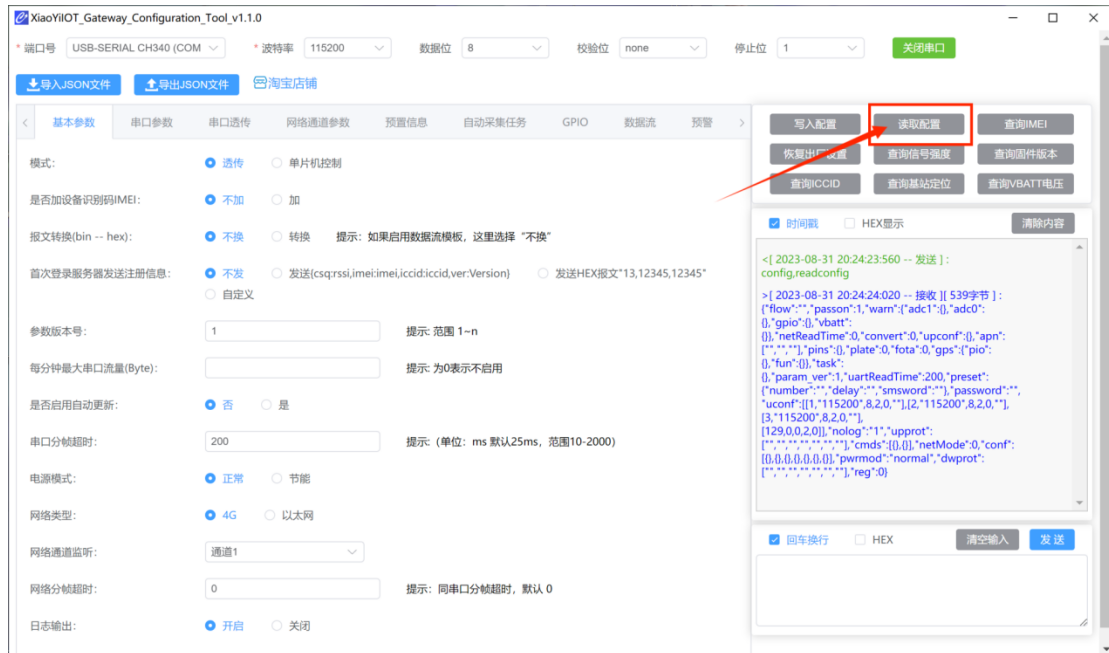
### 1、第一步

打开串口。



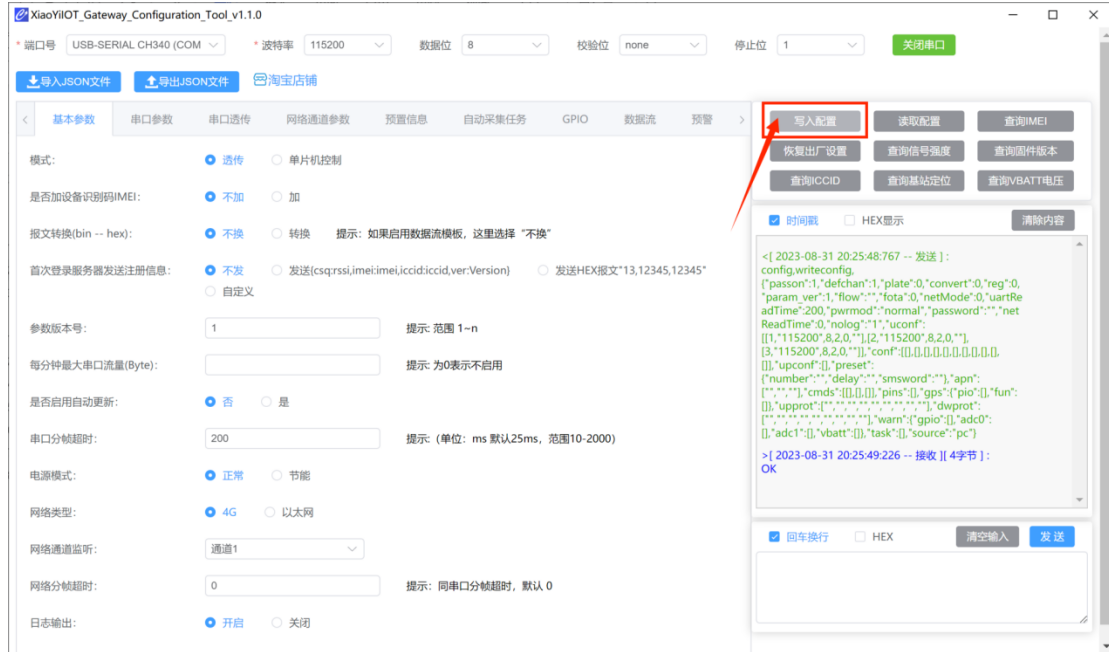
### 2、第二步

点击读取配置按钮, 读取配置。配置解析正常后, 即可进行自定义配置。



### 3、第三步

在配置完成之后点击“写入配置”，将配置写入网关中，网关返回“OK”，随后自动重启，使得配置生效。



**注意：**上面的“写入配置”按钮只能写入除 modbus、DLT645、HJ212 配置以外的配置，modbus、DLT645、HJ212 配置需要在该配置页面内单独进行读取与写入，如下。





### 三、配置参数

#### 1、基本参数

一般保持默认配置即可。如果要修改串口波特率，那么需要修改基本参数中的“串口分帧超时”。例如：用户需要将串口波特率改成 9600，那么建议将“串口分帧超时”修改为 150ms ~ 200ms。如果想让网关使用以太网而不是 4G 网络，那么只需要将“网络模式”切换为“以太网”。注意：以太网模式不支持 TCP SSL 协议，因此无法在此模式下发送 https 请求，无法连接 TCP SSL 平台与 MQTT SSL 平台等。



#### 2、串口配置

初始状态中所有串口都是打开的，RS485 方向控制 GPIO 默认选空，以避免 485 串口不可用。用户修改了串口波特率后，建议去“基本参数”中修改“串口分帧超时”参数，以避

免在该波特率下配置返回“JSON ERROR”的问题。



修改了波特率并成功写入配置之后，相应的，打开串口的波特率也要改为修改后的波特率。



### 3、串口透传

串口透传功能可选择最多两个主串口和一个从串口。主串口的数据会透传到从串口，从串口接收的数据会分别透传到对应的主串口。

串口透传的使用如：主串口 1 接了一个主机 1，主串口 2 接了一个主机 2，从串口 3 接了一个从机，主机 1 要读取从机采集的温度属性，主机 2 要读取从机采集的湿度属性，主机 1 不断下发读取温度属性的指令到串口 1，网关将指令透传到从串口 3，从而下发到从机，此时从机响应的数据将会透传给主串口 1，主机 2 的情况同主机 1。同时，网关本身也可作为一个主机，查询从机的数据。



### 4、网络通道参数

网络通道配置，可以让网关接入用户私有服务器或其他云。

以 MQTT 配置为例，根据提示填入参数即可。

(1) 其中“自动任务时间间隔”参数是配合“自动采集任务”使用的，如果未启用“自动采集任务”，那么这里保持默认配置即可。



基本参数 串口参数 串口透传 网络通道参数 预置信息 自动采集任务 GPIO 数据流 预警

通道1 通道2 通道3 通道4 通道5 通道6 通道7 通道8 通道9

启用  不启用

通道类型：  
 HTTP  SOCKET  MQTT  ONENET  阿里云  腾讯云  
 腾讯云(新版)  百度云(已废弃)  百度云(物联网核心套件)  华为云  天翼云

MQTT心跳间隔： (单位：秒)

自动任务时间间隔： (单位：秒) 提示：默认1800 范围：60~3600

MQTT的地址或域名：

MQTT服务器的端口号： 提示：端口号范围：1~65536

MQTT的登录账号：

MQTT的登录密码：

MQTT的会话保持位： 持久会话  离线自动销毁

(2) 配置订阅或发布主题参数时，如果只订阅一个主题，那么直接填入主题即可。如果订阅了多主题，那么格式为：主题 1;QOS;主题 2;QOS;主题 3;QOS。中间用英文分号隔开，结尾无分号。发布主题同理。



MQTT的登录密码：

MQTT的会话保持位： 持久会话  离线自动销毁

订阅消息主题：

发布消息主题：

MQTT的QOS级别： 0  1  2

MQTT的publish参数retain： 0  1

(3) 建议通道 1 绑定串口 1，通道 2 绑定串口 2。

MQTT通道绑定的串口ID： 1  2  3

(4) 在“主题添加 imei”参数中，如果选“是”，那么会自动在订阅和发布的主题后面加 imei。例如：假设设备 imei 为“876543211234567”，用户填入的订阅的主题为“/msg”，那么最终设备订阅的主题为：“/msg/876543211234567”。如果选择“否”，设备订阅或发布的主题即为用户输入的主题。

MQTT的QOS级别:  0  1  2

MQTT的publish参数retain:  0  1

MQTT通道绑定的串口ID:  1  2  3

客户端ID:  提示: 不填系统用IMEI做客户端ID

主题添加IMEI:  是  否 提示: 默认添加 (/自定义主题/IMEI)

transport:  tcp  tcp\_ssl

(5) 在“MQTT 遗嘱”参数中，如果有需要，可填入遗嘱的主题，设备在离线后，该主题收到的消息为设备的 imei。

MQTT通道绑定的串口ID:  1  2  3

客户端ID:  提示: 不填系统用IMEI做客户端ID

主题添加IMEI:  是  否 提示: 默认添加 (/自定义主题/IMEI)

transport:  tcp  tcp\_ssl

MQTT的遗嘱:  提示: 可不填

## 5、自动采集任务

在自采集任务中，“采集等待”指的是两条指令之间的时间间隔，例如设置了采集等待时间为 1000ms，那么在下发了 cmd1 之后，等待 1000ms 再下发第二条指令 cmd2。在下发完一轮指令后，第二轮指令下发的间隔时间由“网络通道参数”配置中的“自动任务间隔时间”决定。下发的指令为 16 进制字符串。

基本参数 串口参数 网络通道参数 预置信息 **自动采集任务** GPIO 数据流 任务

串口1 串口2 串口3

启用  不启用

采集等待:  (单位: ms) 提示: 1~10000

cmd1:  ⊖

cmd2:  ⊖

+ 添加执行指令

## 6、GPIO

目前网关的灯的状态为：

- (1) 通电后，电源灯（绿灯）长亮。
- (2) 网络灯（红灯）：快速闪烁（亮 100ms，灭 100ms），表示正在注册 GMS。  
较慢闪烁（亮 500ms，灭 500ms），表示正在附着 GPRS。  
慢闪（亮 100ms，灭 1900ms），表示成功连接服务器。
- (3) mode 灯（黄灯）：连接服务器后长亮。

为保持上述状态，请将本配置置为“不启用”。如若遇到 led 灯状态异常，可检查是否开启了本配置。



## 7、数据流

可以在这里对上行和下行数据进行处理。数据流模板绑定的通道与网络通道配置的通道一一对应。数据流模板函数需使用 lua 语言编写。数据流模板函数的格式为：

```
function
```

```
--用 str 变量接收上行或下行数据，三个点是固定写法，表示上行或下行数据。
```

```
local str = ...
```

```
--这里对 str 进行处理，最后 return 处理后的数据
```

```
return str
```

```
end
```

(1) 例如，设备采集到的数据为 modbus\_rtu 协议格式的，而服务器要求上传 JSON 格式数据，就可以通过数据流模板进行处理。处理示例：

```

function
  local str = ...
  local idx, crc = pack.unpack(str:sub(-2, -1), "H")
  local tmp = str:sub(1, -3)
  if crc == crypto.crc16("MODBUS", tmp) then
    local idx, addr, fun, length = pack.unpack(tmp:sub(1, 3), ">b3")
    local data = str:sub(4, -3)
    local t = {}
    local values = {}
    t.timei = misc.getTimei()
    t.addr = addr --从机地址
    t.fun = fun --功能码
    t.byte = length --数据字节数
    if reqAddr ~= nil then
      t.reqAddr = reqAddr --起始寄存器地址
    end
    if fun == 0x03 or fun == 0x04 then
      for i = 1, length, 2 do
        local indx, val = pack.unpack(data, ">H", i)
        table.insert(values, val)
      end
      t.data = values
      return json.encode(t)
    elseif fun == 0x01 or fun == 0x02 then
      for j = 1, length, 1 do
        local indx, val = pack.unpack(data, "b", j)
        table.insert(values, val)
      end
      t.data = values
      return json.encode(t)
    else
      return str
    end
  end
  return str
end
end

```

(2) 在配置了 MQTT 多主题发布时，可以通过“发送数据流模板”将消息发布到指定主题。在数据流模板函数最后 return 数据时，在后面添加主题对应的下标。

假设发布主题为：topic1;0;topic2;0;topic3;0

现需要向 topic2 主题发布消息，可以将其视为一个 lua 数组：{ topic1, 0, topic2, 0, topic3, 0 }，那么其中每个元素的下标依次为 1,2,3,4,5,6，由此可知 topic2 的下标为 3。

那么数据流函数格式为：

```

function
  local str = ...
  return str,3
end

```

这里直接将所有数据发布到 topic2 主题了，用户可以根据自己的需要，在函数中写条件判断，以决定数据发布到对应的主题。

(3) MQTT 订阅了多个主题时，如果想知道接收的消息对应的是哪个订阅的主题，可以在“接收数据流模板”中多解析一个主题字段。如下：

```

function
  local str, topic = ...
  local jsonTable = {}
  jsonTable["msg"] = str
  jsonTable["topic"] = topic
  return json.encode(jsonTable)
end

```



end

上面的模板函数中的“topic”就是此次消息的主题，这里将消息和主题组成了一个JSON字符串返回了，用户可根据自己的需求进行其他处理。

## 8、任务

这里也用lua语言写任务函数。但是不能通过local str = ...的方式接收参数。可以在这里写一些定时任务之类的。例如定时向串口写入数据之类的。

**特别注意，如果编写的代码中有错误，那么4G-网关内的程序将无法正常运行，此时，无法通过配置修复问题，只能重刷固件才能恢复，请谨慎使用此功能。**



基本参数 串口参数 网络通道参数 预置信息 自动采集任务 GPIO 数据流 任务

启用  不启用

任务1:

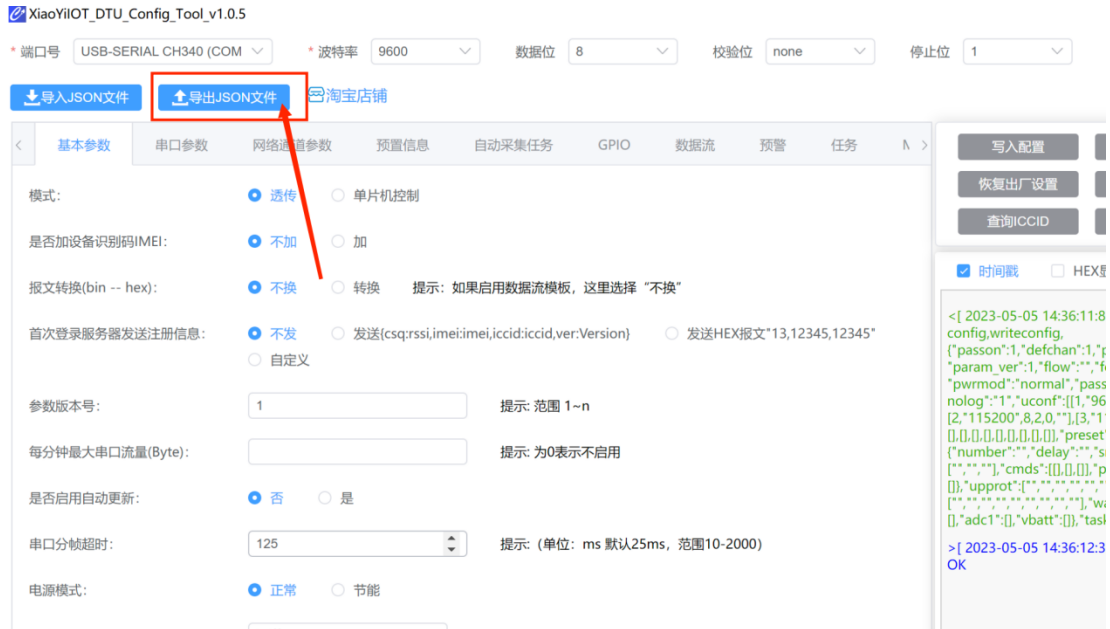
```
function
while true do
sys.wait(30000)
uart.write(1,"Hello World")
end
end
```

## 四、配置保存

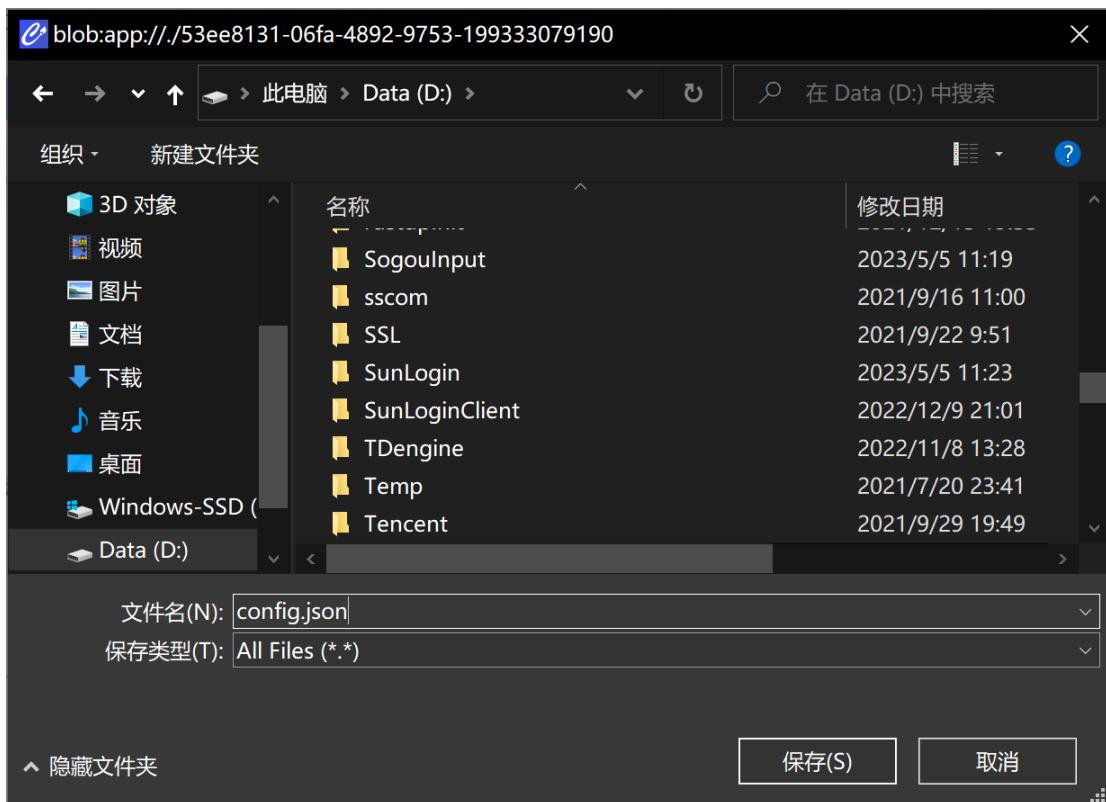
用户进行了自定义配置之后，可以将配置结果保存到文件，以便后续进行配置的时候不用将配置参数重新手动输入一遍。

操作步骤：

(1) 用户在页面中配置完成之后，点击“导出JSON按钮”。（导出的JSON文件中不包含modbus配置）



(2) 保存生成的 JSON 文件。



(3) 需要使用此配置的时候, 点击“导入 JSON 文件”。将前面保存的 JSON 文件导入进来, 可将文件中的配置参数解析到页面中。



注意：“导入JSON文件”功能只是将JSON文件中的配置参数解析到页面中，并非直接写入到网关中，因此还需要执行“写入配置”才会将配置写入到设备中去。此外，modbus配置需要在该配置页面中单独进行导入导出操作，如下。



感谢阅读!