



KTS 系列产品功能手册

----Modbus RTU Client 调试手册

适用产品 – KTS/KAC/KCC 全系列网关

Ver.2020.3

目录

| | |
|-------------------------------------|----|
| 1、功能概述..... | 1 |
| 2、MODBUS RTU 简介..... | 1 |
| 3、MODBUS RTU CLIENT 功能模组..... | 1 |
| 3.1、主机网关基本设置..... | 1 |
| 3.2、MODBUS RTU CLIENT 基本设置..... | 2 |
| 3.3、MODBUS RTU CLIENT 集成关系对象参数..... | 3 |
| 3.4、MODBUS RTU CLIENT 集成关系实例..... | 4 |
| 4、MODBUS 调试工具使用说明..... | 8 |
| 4.1、USB-485 串口设备连接说明..... | 8 |
| 4.2、CAS MODBUS SCANNER 配置说明..... | 9 |
| 4.3、CAS MODBUS SCANNER 测试说明..... | 10 |
| 5、常见问题 FAQ..... | 12 |

1、功能概述

正爵KTS/KAC/KCC系列主机网关包含Modbus RTU/TCP Client模组和Modbus RTU/TCP Slave模组，可用于不同的控制系统与Modbus主站/从站之间的数据通讯，例如通过Modbus RTU/TCP Client模组，使用KNX系统与空调系统进行集成，通过Modbus RTU/TCP Slave模组，将KNX系统作为子系统，接入到BA控制系统。

本手册详细介绍了“Modbus RTU Client模组”的使用方法 & 项目实际应用案例分析。

2、Modbus RTU 简介

Modbus 是全球第一个真正用于工业现场的总线协议，主要包括 Modbus RTU（基于 RS485）和 Modbus TCP（基于 TCP/IP）。

Modbus RTU 采用主从工作方式，允许一台主站设备和多台从站设备通讯，从站的地址范围为 0~247，其中 0 为广播地址，实际地址范围为 1~247。状态反馈采用周期轮询读的方式，因此当节点数或地址数较多时会影响状态反馈的时间。在使用我公司主机网关产品进行系统集成时，为了不影响通讯速度，建议从站设备之间采用“手拉手”的方式进行系统布线，通讯线缆采用 RVSP2*1.0，从站设备总数不要超过 20 个。

Modbus RTU 通讯格式如下：

| 设备地址 | 功能码 | 数据 | CRC校验 | |
|------|-----|---------|-------|-----|
| | | | 低位 | 高位 |
| 1字节 | 1字节 | N x 1字节 | 1字节 | 1字节 |

3、Modbus RTU Client 功能模组

Modbus RTU Client (K1011) 功能模组采用自定义的编辑方式，可将组地址与 Modbus 协议的地址和功能码进行任意对应，提高了数据对应之间的灵活性。针对 Modbus 协议地址的连续性特点，既可对从站的单地址进行读写，也可以同时对多个地址进行读写。

Modbus RTU Client 模组支持以下标准 Modbus 功能码：

| 功能码 | 功能 |
|------|---------------|
| 0x01 | 读取线圈状态(0/1) |
| 0x02 | 读取离散输入状态(0/1) |
| 0x03 | 读取保持寄存器 |
| 0x04 | 读取输入寄存器 |
| 0x05 | 写入单个线圈(0/1) |
| 0x06 | 写入单个保持寄存器 |
| 0x10 | 写入多个保持寄存器 |

其中 0x03、0x04 和 0x06 为经常使用的功能。用户调试时，请按照如下步骤进行分析调试：

- 1、获取系统/设备的 Modbus 控制协议；
- 2、根据协议，使用 Modbus 调试工具(如 CAS Modbus Scanner)进行测试；
- 3、测试成功后，使用 Modbus RTU Client 模组进行编辑。

3.1、主机网关基本设置

3.1.1、本产品为网页编程方式，请使用非 IE 内核的浏览器进行功能配置，如 Firefox、Chrome 等；

3.1.2、网页 IP 地址：192.168.1.232，用户名：admin，密码：123；

3.1.3、登陆网关配置页面后，在“基本设置”->“串口设置”中，选择对应的串口，根据设备协议说明书修改通讯参数，之后点击“提交”按钮并重启以完成修改串口的操作。

3.1.4、串口设置包含以下参数：

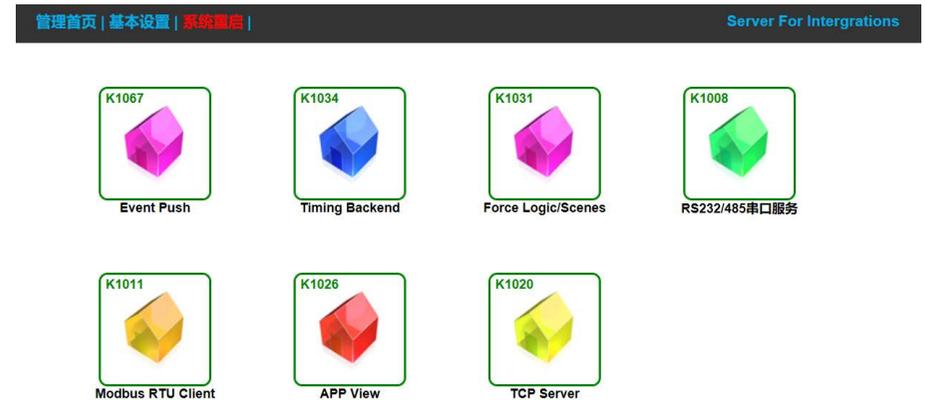
- 波特率：常见为 2400、4800、9600、1920、38400、115200 等
- 数据位：常见为 8
- 停止位：常见为 1 或者 2
- 校验位：O (奇校验)、E (偶校验), N (无校验)、S (0) 和 M (1)
- 功能分配：需设置为 Modbus RTU Client
- time(ms)：数据通讯间隔时间，默认为 100ms，可根据需要进行修改

下图所示为将主机网关的 COM3 端口作为 Modbus RTU Client 模组，相关参数分别是波特率 9600bps，数据位为 8，停止位为 1，校验位为 N，数据通讯间隔时间为 100ms。



3.2、Modbus RTU Client 基本设置

用户返回“管理首页”，点击“Modbus RTU Client”即可进入配置页面：



鼠标右键点击“建筑/功能”，可以“新增集成关系”，如有多个从设备，为便于区分设备和集成关系的复制，可以“添加子项/楼层”，在新建的目录下进行集成关系的设置。



3.3、Modbus RTU Client 集成关系对象参数

一条完整的集成关系如下图所示，包含以下内容

The screenshot shows the configuration window for a Modbus RTU Master/Client. It includes fields for: ① Object Name, ② KNX EIS (with a dropdown), ③ Control Group Address (0/0/0), ④ Serial Port, ⑤ Modbus Function (dropdown showing [0x01] read_coils), ⑥ Modbus Slave ID, ⑦ Modbus Start Addr., ⑧ Read Multi-Status Offset, and ⑨ Offline count detect (0). A dropdown menu for the Modbus Function is open, listing options: [0x01] read_coils, [0x02] read_discrete_inputs, [0x03] read_holding_registers, [0x04] read_input_registers, [0x05] write_single_coil, [0x06] write_single_hold_register, and [0x10] write_Multi_hold_register.

① 对象名称：集成关系的名称，可根据实现的功能进行命名；

② KNX EIS：KNX控制Modbus的地址或状态反馈给KNX的数据类型，

点击右侧下拉箭头选择：

| 请选择EIB/KNX数据类型 |
|--------------------------------------|
| READ: KNX Read |
| EIS1: EIB_switch(1bit) |
| EIS2: EIB_dimming_control(4bits) |
| EIS3: EIB_time(3Byte) |
| EIS4: EIB_date(3Byte) |
| EIS5: EIB_Float_value (2Byte) |
| EIS6: EIB_value (1Byte) |
| EIS7: EIB_drive (1bit) |
| EIS8: EIB_Priority_control(2bits) |
| EIS9: EIB_Float_Value (4Byte) |
| EIS10: EIB_Count_Value (2Byte) |
| EIS11: EIB_Count_Value (4Byte) |
| EIS13: EIB_ASCII_Character(8bits) |
| DPT5.001: Percentage(0...100%) |
| DPT6.010: Counter Pulses(-128...127) |

| 常用数据类型 | |
|--------|-----------|
| 数据类型 | 功能 |
| EIS1 | 设备开关 |
| EIS5 | 温度、亮度、浓度等 |
| EIS6 | 模式、风速、场景等 |
| EIS9 | 电量等 |
| EIS10 | 电流等 |
| EIS18 | 湿度等百分比数值 |

注：不同Modbus设备功能所对应的EIS数据类型可能有所不同，需在调试中进行测试。

③ 控制组地址：KNX控制Modbus的地址或状态反馈给KNX的组地址；

④ 串口端口号：选择在“基本设置”->“串口设置”中所设置好的端口，如已设置多个端口，请选择该系统/设备所连接的端口，避免因选择错误而造成无法控制的现象产生；

⑤ Modbus Function：Modbus功能码，点击右侧下拉箭头选择

| |
|-----------------------------------|
| [0x01] read_coils |
| [0x02] read_discrete_inputs |
| [0x03] read_holding_registers |
| [0x04] read_input_registers |
| [0x05] write_single_coil |
| [0x06] write_single_hold_register |
| [0x10] write_Multi_hold_register |

⑥ Modbus Slave ID：所需要对应的Modbus从站的地址编号，一般为1~247，设置方法及默认ID请参考该设备的说明书；

⑦ Modbus Start Addr.：所需要对应KNX组地址的Modbus线圈或寄存器的起始地址。各功能码对应的起始地址如下：

| 功能码 | 起始地址初始数值 |
|------|----------|
| 0x01 | 1 |
| 0x02 | 10001 |
| 0x03 | 40001 |
| 0x04 | 30001 |
| 0x05 | 1 |
| 0x06 | 40001 |
| 0x10 | 40001 |

对于单独控制的功能，如0x06（写入单个寄存器），起始地址即为设定地址。对于多个连续地址操作（读取状态/写入多个地址）时，起始地址为第一个地址，后续地址在“⑧偏移量窗口”定义；

⑧ 偏移量窗口：对多个线圈或者寄存器执行读取/写入命令时的偏移量窗口，当“⑤Modbus Function”选择为读取命令(0x01/02/03/04)时，显示为“Read Multi-Status Offset”，用户在此处定义除起始地址之外的后续地址的对应关系，起始地址的对应关系已在“⑦ Modbus Start Addr.”中进行定义。

当“⑤Modbus Function”选择为写入多个保持寄存器(0x10)时，显示为“Write Multi-Registers Offset”，用户在此处定义包含起始地址在内的所有地址的对应关系；

⑨ Offline count detect: Modbus从设备掉线后重连次数设定，默认为0，代表不限制重连次数；

如果将“⑤ Modbus Function”选择为“0x06”，集成关系如下：

⑩ “Start bit (Min=0)”表示定义写入bit位的起始位，默认为0；“End bit (Max=15)”表示定义写入bit位的最终位，默认为15，即表示对该保持寄存器所有数据位进行写入。注：修改起始位和最终位，会影响通讯速率，如无特殊情况，建议保持默认。以下为采用标准Modbus协议的设备地址表：

| 功能码 | 实际地址 | 数据 | 功能 |
|-----------|------|------|----------|
| 0x03/0x06 | 2 | 0 | 关机 |
| | | 1 | 开机 |
| | 3 | 0 | 制冷 |
| | | 1 | 制热 |
| | | 2 | 除湿 |
| | 4 | 3 | 通风 |
| | | 5~35 | 设定温度, °C |
| | 6 | 0 | 自动风速 |
| | | 1 | 低风速 |
| | | 2 | 中风速 |
| 3 | 高风速 | | |
| 0x04 | 1 | 0~99 | 室内温度, °C |

注：所有温度返回值均放大10倍

3.4、Modbus RTU Client 集成关系实例

3.4.1、分析协议：以上一章节所提的设备协议为例，用户对其进行分析后，按照以下步骤进行配置：

- 通过两芯通讯线将设备与主机网关RS485接口连接，注意区分A和B
- 设备地址(Slave ID): 此实例中温控器地址为1
- 控制功能码: 此协议中所用到的功能码有0x03、0x04和0x06
- 控制命令对应的实际地址与数值，一般实际地址为十六进制，需转换为十进制后进行集成关系的对应
- 协议补充说明或注意事项: 此实例需注意温度数值反馈的说明
- 集成关系建立好后，需重启主机网关后，测试相应的控制功能

3.4.2、开启/关闭：首先将此温控器的地址填入“Modbus Slave ID”空格内，其次根据协议表得知开关功能码为0x06，命令数值为0和1，所以在“Modbus Function”处选择“0x06”，在“KNX EIS”处选择1bit的数据类型，并且建立14/0/1作为控制温控器开启/关闭的组地址，之后将协议中的实际地址数值加40001，得出保持寄存器地址为40003，填入“Modbus Slave ID”所对应的空格内，并且保持“Start bit”和“End bit”默认数值，最后点击“更新”按钮完成“温控器开启/关闭”集成关系的设置。

3.4.3、模式：模式选择采用EIS6的数据类型，集成关系如下：

更新 删除 复制 移动到 8 上移 下移

对象名称 模式

KNX EIS EIS6: EIB_value (1Byte)

控制组地址 14/0/2 Select

串口号 COM3: RS 485

Modbus Function [0x06] write_single_hold_regist

Modbus Slave ID 1

Modbus Start Addr. 40004

Start bit (Min=0) 0

End bit (Max=15) 15

3.4.4、风速：风速设定与模式选择类似，集成关系如下：

更新 删除 复制 移动到 8 上移 下移

对象名称 风速

KNX EIS EIS6: EIB_value (1Byte)

控制组地址 14/0/4 Select

串口号 COM3: RS 485

Modbus Function [0x06] write_single_hold_regist

Modbus Slave ID 1

Modbus Start Addr. 40007

Start bit (Min=0) 0

End bit (Max=15) 15

3.4.4、温度设定：对于设定温度，采用 EIS5 的数据类型，协议中说明温度返回值需放大 10 倍，所以将 KNX 发送 Modbus 的数值乘以 10 为 Modbus 接收数值，集成关系如下：

更新 删除 复制 移动到 8 上移 下移

对象名称 温度设定

KNX EIS EIS5: EIB_Float_value (2Byte)

控制组地址 14/0/3 Select

串口号 COM3: RS 485

Modbus Function [0x06] write_single_hold_regist

Modbus Slave ID 1

Modbus Start Addr. 40005*10

Start bit (Min=0) 0

End bit (Max=15) 15

3.4.5、状态反馈：Modbus 的状态反馈采用轮询机制，为了避免影响通讯速度，所以在 Modbus RTU Client 模组中，相同功能码且临近的反馈地址需写在同一条集成关系中。在实例的协议中，开关状态、模式状态、温度设定状态和风速状态的功能码均为 0x03，集成关系如下：

更新 删除 复制 移动到 8 上移 下移

对象名称 状态反馈

KNX EIS EIS1: EIB_switch(1bit)

控制组地址 15/0/1 Select

串口号 COM3: RS 485

Modbus Function [0x03] read_holding_registers

Modbus Slave ID 1

Modbus Start Addr. 40003

Read Multi-Status Offset: [Max. 31 items.]

40004: EIS6=15/0/2=MBV
40005: EIS5=15/0/3=MBV/10
40006: /
40007: EIS6=15/0/4=MBV
40008:
40009:
40010:
40011:
40012:
40013:
40014:

Offline count detect 0 (0=disable)

其中 KNX EIS、控制组地址和 Modbus Start Addr.均对应起始状态地址，区域的状态地址在偏移量窗口中填写，表格内每一行代表一个连续的地址，中间跳过的寄存器地址以“/”代替，格式为“数据类型=组地址=MBV”，其中“数据类型”和“组地址”根据控制对象由用户自行定义，“MBV”为固定字符，协议中说明温度返回值需放大 10 倍，对设定温度的状态反馈而言，需要除以 10 才是真实数值。

3.4.6、室内温度：根据协议内容，室内温度功能码为 0x04，与其它状态反馈功能码不同，集成关系如下：

The screenshot shows the configuration for '室内温度' (Indoor Temperature) in the Modbus RTU Master/Client software. The configuration fields are as follows:

- 对象名称: 室内温度
- KNX EIS: EIS5: EIB_Float_value (2Byte)
- 控制组地址: 15/0/5
- 串口端口号: COM3: RS 485
- Modbus Function: [0x03] read_holding_registers
- Modbus Slave ID: 1
- Modbus Start Addr.: 30002
- Read Multi-Status Offset: [Max. 31 items.]
- Offline count detect: 0 (0=disable)

The 'Read Multi-Status Offset' window is empty, indicating no offsets are currently defined for this object.

3.4.7、写入多个保持寄存器 (0x10)：开启命令功能码为 0x10，寄存器地址为 40001，需要在偏移量窗口填写该地址的对应关系，格式为“=EIBV”，同时也支持数值运算，如=EIBV*10 等，

集成关系如下：

The screenshot shows the configuration for '开关控制' (Switch Control) in the Modbus RTU Master/Client software. The configuration fields are as follows:

- 对象名称: 开关控制
- KNX EIS: EIS1: EIB_switch(1bit)
- 控制组地址: 14/0/1
- 串口端口号: COM3: RS 485
- Modbus Function: [0x10] write_Multi_hold_registe
- Modbus Slave ID: 1
- Modbus Start Addr.: 40001
- Write Multi-Registers Offset:

The 'Write Multi-Registers Offset' window contains the following entries:

```

40001: =EIBV
40002:
40003:
40004:
40005:
40006:
40007:
40008:
40009:
40010:
40011:

```

如要同时对以 40001 为起始位的 4 个保持寄存器发送数值，需要在偏移量窗口填写包含起始地址在内所有的对应关系，集成关系如下：

The screenshot shows the configuration for '空调内机' (Air Conditioning Indoor Unit) in the Modbus RTU Master/Client software. The configuration fields are as follows:

- 对象名称: 空调内机
- KNX EIS: EIS1: EIB_switch(1bit)
- 控制组地址: 14/0/1
- 串口端口号: COM3: RS 485
- Modbus Function: [0x10] write_Multi_hold_registe
- Modbus Slave ID: 1
- Modbus Start Addr.: 40001
- Write Multi-Registers Offset:

The 'Write Multi-Registers Offset' window contains the following entries:

```

40001: =EIBV
40002: =EIBV-1
40003: =EIBV*23
40004: =EIBV*2
40005:
40006:
40007:
40008:
40009:
40010:
40011:

```

图中所示为对组地址 14/0/1 发送 1 时，空调内机运行模式为：开启（数值为 1），模式为制冷（数值为 0），温度为 23℃，风速为中速(数值为 2)，偏移量窗口内的每个寄存器所需要的数值根据接收到的 40001 的数值为基础进行运算。

3.4.8、设定风速：某空调室内的风速控制寄存器地址为 40011，且占据该寄存器 bit5 和 bit4 的位置，集成关系如下：

更新 删除 复制 移动到 0 上移 下移

对象名称: 风速

KNX EIS: EIS6: EIB_value (1Byte)

控制组地址: 14/0/4 Select

串口端口号: COM3: RS 485

Modbus Function: [0x06] write_single_hold_regist

Modbus Slave ID: 1

Modbus Start Addr.: 40011

Start bit (Min=0): 4

End bit (Max=15): 5

图中所示为对组地址 14/0/4 发送 1byte 数值时，只对该寄存器的 bit5 和 bit4 进行触发，进行风速切换。

3.4.9、数值读取高位：室内温度的功能码为 0x03，寄存器地址为 40005，数据的高八位表示整数，低八位表示小数，读取数值高八位和低八位的集成关系如下：

| | |
|--|--|
| 更新 删除 复制 移动到 0 | 更新 删除 复制 移动到 0 |
| 对象名称: 读取高八位 | 对象名称: 读取低八位 |
| KNX EIS: EIS5: EIB_Float_value (2Byte) | KNX EIS: EIS5: EIB_Float_value (2Byte) |
| 控制组地址: 1/1/1 Select | 控制组地址: 1/1/2 Select |
| 串口端口号: COM3: RS 485 | 串口端口号: COM3: RS 485 |
| Modbus Function: [0x03] read_holding_registers | Modbus Function: [0x03] read_holding_registers |
| Modbus Slave ID: 1 | Modbus Slave ID: 1 |
| Modbus Start Addr.: 40005/256 | Modbus Start Addr.: 40005>>0&255 |

3.4.10、某空调室内机开机和模式点表如下，将所需功能对应的数值按位依次填入，所得的二进制数转换为十进制数值，即为控制的数值。

例如开机制冷：0000 0000 1000 0100，十进制为 132，开机自动：0000 0000 1001 0000，十进制为 144，对 40002 发送数值 132 和 144 分别为开机制冷模式和开机自动模式。

以此类推，即可得出相应功能的命令数值。

| 功能码 | 寄存器地址 | 数据位 | 数值 | 功能 |
|------|-------|---------|----|-----|
| 0x10 | 40002 | bit15~8 | 0 | 预留 |
| | | bit7 | 1 | 开机 |
| | | | 0 | 关机 |
| | | bit6~5 | 0 | 预留 |
| | | bit4 | 1 | 自动开 |
| | | | 0 | 自动关 |
| | | bit3 | 1 | 制冷开 |
| | | | 0 | 制冷关 |
| | | bit2 | 1 | 制热开 |
| | | | 0 | 制热关 |
| | | bit1 | 1 | 除湿开 |
| | | | 0 | 除湿关 |
| bit0 | 1 | 送风开 | | |
| | 0 | 送风关 | | |

注: bit6~bit0各位互斥

集成关系如下：

更新 删除 复制 移动到 0 上移 下移

对象名称: 开机&模式

KNX EIS: EIS6: EIB_value (1Byte)

控制组地址: 14/0/1 Select

串口端口号: COM3: RS 485

Modbus Function: [0x10] write_Multi_hold_registe

Modbus Slave ID: 1

Modbus Start Addr.: 40001

Write Multi-Registers Offset:

```

40001: =EIBV
40002:
40003:
40004:
40005:
40006:
40007:
40008:
40009:
40010:
40011:
    
```

3.4.11、某电表的“有功电能总和”占用 40001 和 40002 两个保持寄存器地址，字长均为 2 字节，如需读取有功电能总和，集成关系如下：

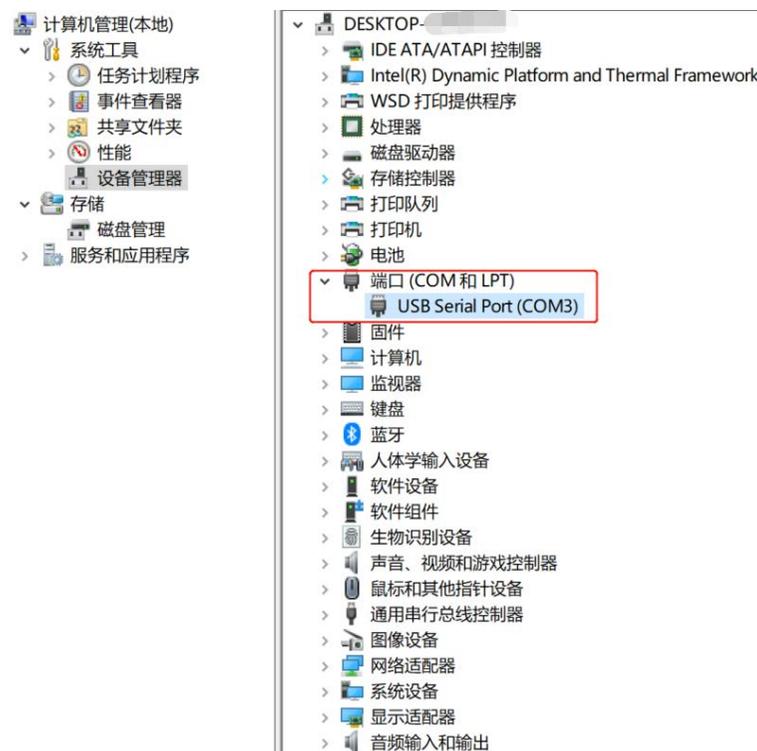
| | | | | |
|--------------------|-------------------------------|--------|-----|---|
| 更新 | 删除 | 复制 | 移动到 | 0 |
| 对象名称 | 有功电能总和 | | | |
| KNX EIS | EIS11: EIB_Count_Value (4Byl) | | | |
| 控制组地址 | 0/0/0 | Select | | |
| 串口端口号 | COM3: RS 485 | | | |
| Modbus Function | [0x03] read_holding_registers | | | |
| Modbus Slave ID | 1 | | | |
| Modbus Start Addr. | 40001~40002 | | | |

4、Modbus 调试工具使用说明

当进行系统集成对接时，可先使用 USB-485 串口设备并配合 Modbus 调试工具对设备进行测试。

4.1、USB-485 串口设备连接说明

用户需将 USB-485 串口设备的 USB 端接入 PC，查询到所使用的的 COM 端口号：右键“计算机”->“管理”->“系统工具”->“设备管理器”->“端口”，如串口设备正常，即可发现该串口并显示端口号。如无法找到该硬件设备，请正确安装产品驱动或者更换串口设备。

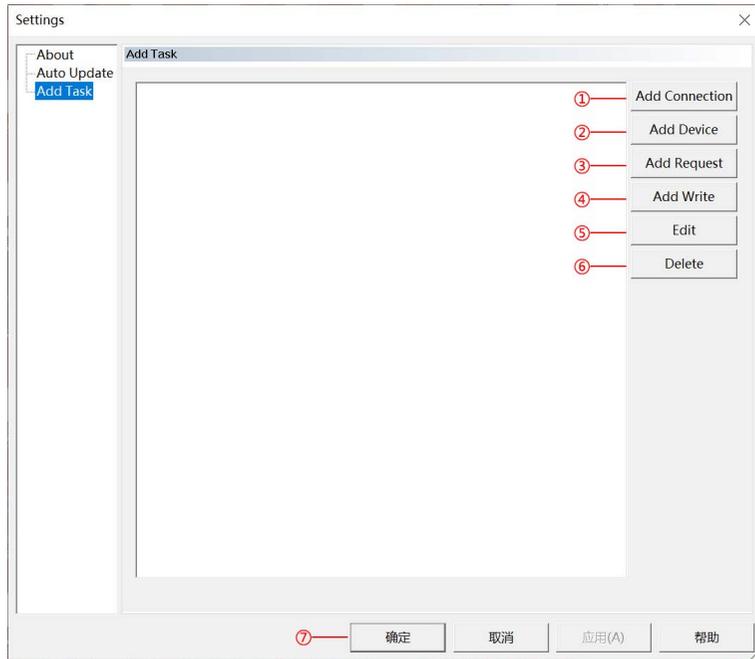


4.2、CAS Modbus Scanner 配置说明

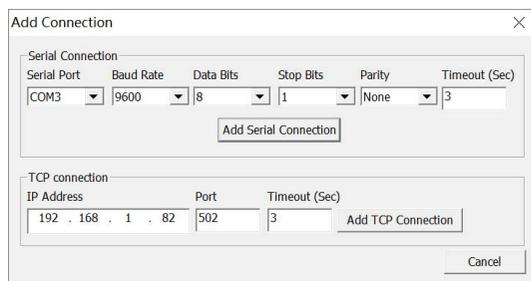
CAS Modbus Scanner 是一款免费测试工具，可以通过 PC 对 Modbus 从设备进行命令控制与状态读取。软件可以在我公司网站上下载：

<http://www.kanontec.com/uploads/soft/200225/CASModbusScanner.rar>

安装完成后，第一次打开软件时，会提示创建新的任务，点击确认后，软件界面如下：



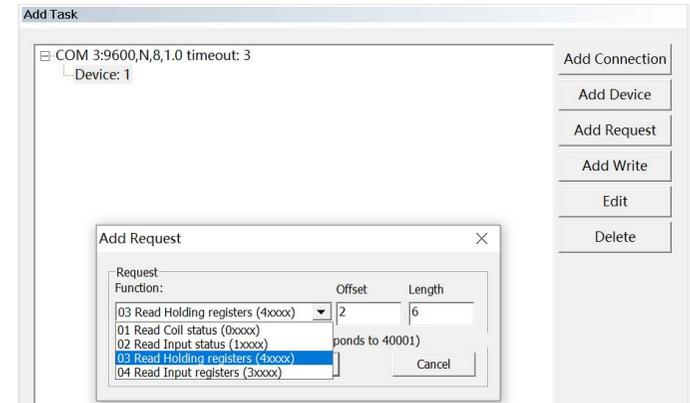
① Add Connection: 点击后按照设备通讯参数进行设置并建立串口连接



② Add Device: 选中建立好的串口之后，点击“Add Device”进行设备添加，将设备地址填入“Slave ID”栏中，并添加；



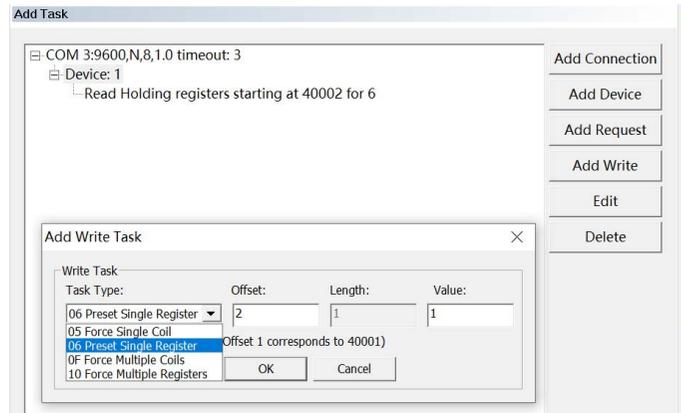
③ Add Request: 选中“Device: 1”，点击“Add Request”，建立状态读取指令，根据设备协议，选择相应的功能、地址和数量：



- Function: 功能码，包含0x01, 0x02, 0x03和0x04
- Offset: 状态读取命令的起始地址
- Length: 读取状态命令的数量

图中所示为对从40002开始的连续6位保持寄存器进行数值读取。

④ Add Write: 选中“Device: 1”, 点击“Add Write”, 建立控制指令, 根据设备协议, 选择相应的功能、地址和数值:



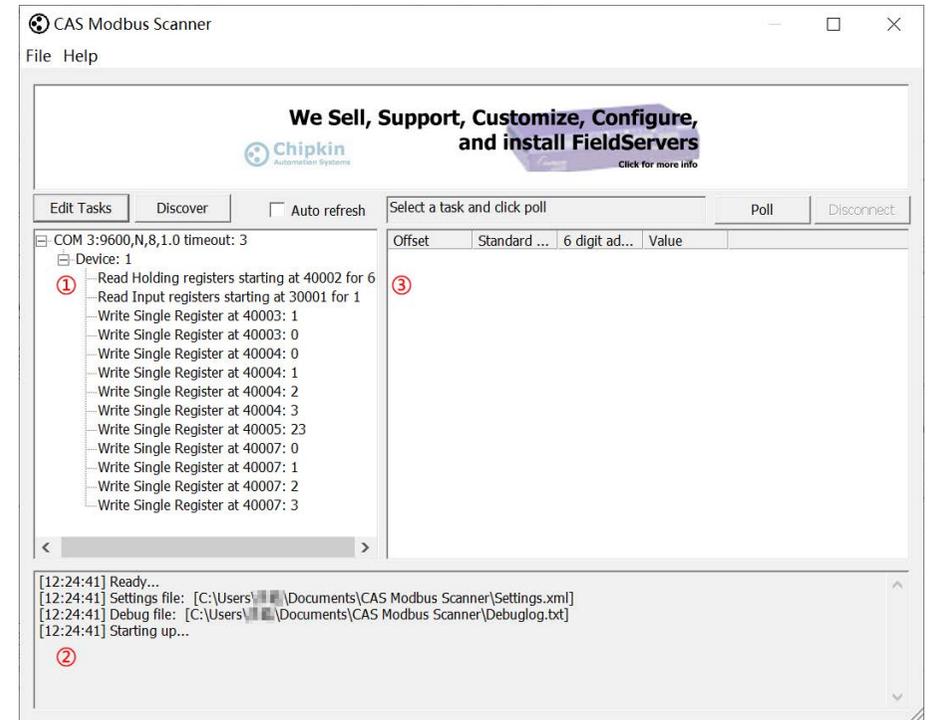
- Task Type: 功能码, 包含0x05, 0x0F和0x10
- Offset: 控制命令的地址
- Length: 控制命令的数量, 当选择0F和10时有效
- Value: 命令执行的数值

图中设置表示对地址为40002的保持寄存器发送数值1。

- ⑤ Edit: 可以对端口、设备、读取命令和控制命令进行修改
- ⑥ Delete: 可以对端口、设备、读取命令和控制命令进行删除
- ⑦ 确定: 在添加好相应的命令之后, 可以点击确定, 返回主页面。

4.3、CAS Modbus Scanner 测试说明

4.3.1、基本操作: 按照本文中的设备协议, 在CAS Modbus Scanner中添加相应的命令后, 即可进行测试。双击对话框1内的任意一条命令, 控制的状态会在对话框2中显示, 如果进行状态的读取, 读取后的数值会显示在对话框3内。



4.3.2、控制命令：对地址为40002的保持寄存器发送数值1，如通讯正常并成功控制，显示如下：

```
[15:12:00] Write task has completed successfully
[15:12:00] <= Response: 01 06 00 02 00 00 28 0A
[15:11:59] => Poll: 01 06 00 02 00 00 28 0A
[15:11:53] Write task has completed successfully
[15:11:53] <= Response: 01 06 00 02 00 01 E9 CA
[15:11:52] => Poll: 01 06 00 02 00 01 E9 CA
[15:11:52] Connect to COM 3:9600,N,8,1.0 timeout: 3
[15:11:44] Ready...
```

信息具体含义如下：

- Connect to COM 3:9600,N,8,1.0 timeout: 3 表示所使用端口的通讯参数
- => Poll: 01 06 00 02 00 01 E9 CA: 表示由软件发送至设备的控制命令
- <= Response: 01 06 00 02 00 01 E9 CA: 表示设备接收控制命令的响应
- Write task has completed successfully: 表示命令发送成功

如果发送命令后，通讯信息显示正常，但是Modbus设备没有执行动作，说明通讯正常，需要查看设备协议，确认寄存器地址、命令数值是否正确。

如果发送命令后，显示“Error, Timeout - No response to poll”，则表示发送命令超时，设备没有接收到控制命令，可能是以下原因引起的：

- USB-485串口设备与Modbus设备之间的通讯线是否连接
- USB-485串口设备的串口端是否可以正常工作
- 测试软件的端口通讯参数与设备是否一致
- Modbus设备是否工作正常

如果发送命令后，显示“Disconnected”和“Could not connect to COM 3:9600,N,8,1.0 timeout: 3”，则表示软件无法与USB-485串口设备建立通讯，可能是以下原因引起的：

- Modbus测试软件的端口选择错误
- USB-485串口设备的USB端是否可以正常工作
- 查看设备管理器，确定USB-485串口设备是否正确安装

4.3.3、读取状态：对地址为30001的输入寄存器进行读取，如通讯正常，可成功读取到室内温度为24℃，显示如下：

| Offset | Standard ... | 6 digit ad... | Hex | char | uint... | int... | uint32 | int32 | float32 |
|--------|--------------|---------------|-------|------|---------|--------|---------|--------|-----------|
| 1 | 30001 | 300001 | 0x... | □ | 24 | 24 | inco... | inc... | incomp... |

```
[15:38:39] <= Response: 01 04 02 00 18 B9 3A
[15:38:38] => Poll: 01 04 00 00 00 01 31 CA
[15:38:28] Write task has completed successfully
[15:38:28] <= Response: 01 06 00 02 00 00 28 0A
[15:38:28] => Poll: 01 06 00 02 00 00 28 0A
[15:38:27] Write task has completed successfully
[15:38:27] <= Response: 01 06 00 02 00 01 E9 CA
[15:38:26] => Poll: 01 06 00 02 00 01 E9 CA
```

对地址从40002开始的连续6位保持寄存器进行数值读取，如通讯正常，也可成功读取到当前设备各个状态的数值。

5、常见问题 FAQ

(1) 使用Modbus调试工具可以控制设备，但是通过主机网关的端口无法控制Modbus设备

Step.1 确认主机网关型号，若为KTS/KAC系列主机，需将主机接入KNX系统，并使用KNX专用电源为KNX系统供电；

Step.2 在“基本设置”->“串口设置”中查看主机网关端口的通讯参数与受控Modbus设备通讯参数是否一致，并且“功能分配”是否选择为“Modbus RTU Client”，同时查看Modbus RTU Client中所添加的集成关系对象是否正确，串口端口号的选择是否正确；

Step.3 确认主机网关的COM端口的A和B连接是否与Modbus从站设备的A和B一致；

Step.4 检查是否使用主机网关的COM3端口(KTS1/2/3或KCC1/2/3)，如接在COM3端口，可将随包装盒附送的120Ω终端电阻并联在COM3端口再进行测试；如接在其它485端口无法控制，可接入COM3端口进行测试；

Step.5 若还是无法通过主机网关进行控制，请使用USB-485串口设备将PC端USB接口与主机网关端口进行连接，通过串口调试工具测试网关的端口通讯是否正常。测试方法：将该串口功能设为“自设定232/485”，在“RS232/485串口服务”模组中新增两个集成关系：

| 更新 | 删除 | 复制 | 移动到 | 0 | 更新 | 删除 | 复制 | 移动到 | 0 |
|------------------------------------|--|---|-----|---|------------------------------------|---|--|-----|---|
| 对象名称 | KNX->485 | | | | 对象名称 | 485->KNX | | | |
| 数据类型 | EIS1: EIB_switch(1bit) | | | | 数据类型 | EIS1: EIB_switch(1bit) | | | |
| 控制组地址 | 1/1/1 | Select | | | 控制组地址 | 1/2/1 | Select | | |
| 控制数值 | 1 | | | | 控制数值 | 1 | | | |
| 数据对应流向 | <input checked="" type="radio"/> KNX -> RS 232/485 | <input type="radio"/> RS 232/485 -> KNX | | | 数据对应流向 | <input type="radio"/> KNX -> RS 232/485 | <input checked="" type="radio"/> RS 232/485 -> KNX | | |
| 串口数据格式 | <input checked="" type="radio"/> Hex 十六进制 | <input type="radio"/> ASCII 字符串 | | | 串口数据格式 | <input checked="" type="radio"/> Hex 十六进制 | <input type="radio"/> ASCII 字符串 | | |
| 串口端口号 | COM3: RS 485 | | | | 串口端口号 | COM3: RS 485 | | | |
| 串口数据 (十六进制时, 空格隔开, 如'A5 01 02 5A') | 11 22 33 | | | | 串口数据 (十六进制时, 空格隔开, 如'A5 01 02 5A') | AA BB CC | | | |

建立好对应关系并重启主机网关之后，打开串口调试软件，首先使用软件以十六进制发送设定好的代码“AA BB CC”，观察ETS群组监视器内是否收到组地址1/2/1=1的信号，其次通过ETS群组监视器，发送1/1/1=1，观察命令接收区是否接收到设定好的“11 22 33”的命令。



| Service | Destination | DPT | Info |
|----------|-------------|--------------|-----------|
| from bus | 1/2/1 | | \$01 On |
| to bus | 1/1/1 | 1.001 switch | \$01 On |

若为上图所示，则表示主机网关的端口通讯正常。接下来需通过触发KNX组地址执行Modbus命令，将接收区收到的串口代码与受控Modbus设备协议进行对比，综合判断问题所在。

(2) 一些数值需要显示一位或者两位小数，如温度值、电流值等，如何在Modbus RTU Client进行读取并在App view中显示？

设定温度数值的返回值放大十倍，若读取寄存器数值为265，则实际温度为26.5，需在Modbus RTU Client中选择数据类型为EIS5，并对读取的寄存器数值进行运算，集成关系如下：

| | | | | | |
|--------------------|-------------------------------|----|-------|----|--------|
| 更新 | 删除 | 复制 | 移动到 0 | 上移 | 下移 |
| 对象名称 | 温度读取一位小数 | | | | |
| KNX EIS | EIS5: EIB_Float_value (2Byte) | | | | |
| 控制组地址 | 1/1/1 | | | | Select |
| 串口端口号 | COM3: RS 485 | | | | |
| Modbus Function | [0x03] read_holding_registers | | | | |
| Modbus Slave ID | 1 | | | | |
| Modbus Start Addr. | 40001/10 | | | | |

在App View模组中，对应的显示数据类型选择为“EISx”：

| | | | | | |
|--------|---|----|-------|----|--------|
| 更新 | 删除 | 复制 | 移动到 1 | 上移 | 下移 |
| 对象名称 | 温度 <input type="checkbox"/> 隐藏标题 | | | | |
| 对象类别 | 其他 Others | | | | |
| 数据类型 | (2byte) EISx: EIB_Floatpoint_F | | | | |
| 功能选择 | <input type="radio"/> 数值控制/场景控制 <input checked="" type="radio"/> 数据显示/传感器 | | | | |
| Unit | °C | | | | |
| 状态组地址 | 1/1/1 | | | | Select |
| 位置属性 | 距上 0 px; 距左 0 px; Position | | | | |
| 开启显示 | <input type="text"/> Select | | | | |
| 关闭显示 | <input type="text"/> Select | | | | |
| 上电读标记 | <input type="checkbox"/> | | | | |
| 该对象显示于 | <input checked="" type="radio"/> ALL <input type="radio"/> PAD Only <input type="radio"/> Mobile Only | | | | |

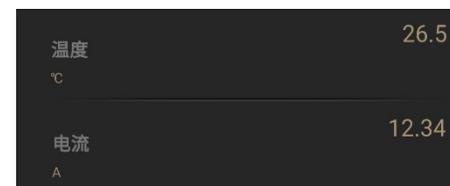
设定电流值的返回值为放大一百倍，若读取寄存器数值为1234，则实际温度为12.34，需在Modbus RTU Client中选择数据类型为EIS9，并对读取的寄存器数值进行运算，集成关系如下：

| | | | | | |
|--------------------|-------------------------------|----|-------|----|--------|
| 更新 | 删除 | 复制 | 移动到 0 | 上移 | 下移 |
| 对象名称 | 电流读取两位小数 | | | | |
| KNX EIS | EIS9: EIB_Float_Value (4Byte) | | | | |
| 控制组地址 | 1/1/2 | | | | Select |
| 串口端口号 | COM3: RS 485 | | | | |
| Modbus Function | [0x03] read_holding_registers | | | | |
| Modbus Slave ID | 1 | | | | |
| Modbus Start Addr. | 40002/100 | | | | |

在App View模组中，对应的显示数据类型选择为“EIS9”：

| | | | | | |
|--------|---|----|-------|----|--------|
| 更新 | 删除 | 复制 | 移动到 1 | 上移 | 下移 |
| 对象名称 | 电流 <input type="checkbox"/> 隐藏标题 | | | | |
| 对象类别 | 其他 Others | | | | |
| 数据类型 | (4byte) EIS9: EIB_Float_Value | | | | |
| 功能选择 | <input type="radio"/> 数值控制/场景控制 <input checked="" type="radio"/> 数据显示/传感器 | | | | |
| Unit | A | | | | |
| 状态组地址 | 1/1/2 | | | | Select |
| 位置属性 | 距上 0 px; 距左 0 px; Position | | | | |
| 开启显示 | <input type="text"/> Select | | | | |
| 关闭显示 | <input type="text"/> Select | | | | |
| 上电读标记 | <input type="checkbox"/> | | | | |
| 该对象显示于 | <input checked="" type="radio"/> ALL <input type="radio"/> PAD Only <input type="radio"/> Mobile Only | | | | |

在App View中显示效果如下：



上海正爵电子有限公司
上海市宝山区罗宁路1288弄联东U谷12B-501室
<http://www.kanontec.com>
E: support@kanontec.com
T: 4008-216-843