

Modbus Master 使用说明

Modbus Master是遵循Modbus标准协议开发的一个FreeIOE 通用应用，您可以在任何内置了FreeIOE的网关中使用Modbus Master和任何遵循Modbus协议（包含Modbus RTU、Modbus ASCII、Modbus TCP、标准的设备或软件通讯交互，读写这些设备或软件的数据。支持串口、TCP套接字方式。

通讯协议

Modbus协议简介

应用概述

Modbus Master应用在Modbus RTU/Modbus ASCII协议中扮演主站的角色，主动向Modbus从站（设备）发起询问指令，请求设备返回符合请求的数据。

Modbus Master应用在Modbus TCP协议中扮演TCP Client的角色，主动连接Modbus Server（设备）并发起询问指令，请求设备返回符合请求的数据。

Modbus Master应用遵循Modbus协议开发，完全兼容Modbus协议标准，使用Modbus Master应用和Modbus设备通讯时，需要清楚的知道Modbus设备的Modbus协议地址，设备变量的功能码以及寄存器协议地址。Modbus Master应用中使用的寄存器地址是**寄存器协议地址(十进制)**，这和寄存器PLC地址不太一样，下图是寄存器协议地址和寄存器PLC地址的映射关系。

寄存器PLC地址	寄存器协议地址	适用功能	寄存器种类	读写状态
00001-09999	0000H-FFFFH	01H 05H 0FH	线圈状态	可读可写
10001-19999	0000H-FFFFH	02H	离散输入状态	可读
30001-39999	0000H-FFFFH	04H	输入寄存器	可读
40001-49999	0000H-FFFFH	03H 06H 0FH	保持寄存器	可读可写

准备工作

连接物理设备

为使用Modbus Master应用和Modbus设备通讯时能更快更顺利的达到目标，我们有必要先了解及做一些准备工作。

1. 了解并确认Modbus设备具体是使用Modbus协议的哪一种子协议（Modbus RTU、Modbus ASCII、Modbus TCP）对外提供数据，一般来说，可以阅读设备提供的数据通讯类文档了解情况，还可以咨询设备的调试人员或者设备厂商的服务支持人员获得帮助。

- 2. 获取所有需要通讯的Modbus设备的物理链路信息，如是通过串口连接，需要知道Modbus设备串口的参数（波特率、数据位、停止位等信息），如是通过以太网连接，需要知道Modbus设备的IP地址及端口号。
- 3. 获取所有需要通讯的Modbus设备的Modbus设备地址信息，没有Modbus设备地址信息，无法进行后面的工作。
- 4. 获取Modbus设备对外提供变量数据的列表文件，一般来说，多少电子表格格式比较多，这种文件在自动化行业多称位设备点表，在这份点表中，我们可以看到设备的很多变量及其Modbus属性都会被罗列出来，一般会包含：名称、描述、Modbus功能码、Modbus寄存器地址、数据类型、运算系数等信息，如这份设备点表中无法获取到这些信息，那么就无法进行后面的工作。
- 5. 前面3个步骤的准备工作都完成后，还需要将Modbus设备提供点表整理为Modbus Master应用所需的点表，这个整理点表的工作通过电子表格软件（如Office EXCEL）可达到事半功倍的效果、Modbus Master应用的[参考设备模板](#)。
- 6. 确认网关和设备之间的物理连接是否正常，如是通过串口方式连接Modbus设备，检查串口接线是否正确；如是通过以太网方式连接Modbus设备，检查网线是否连接、网卡状态灯是否正常、还需要通过tcp/ip诊断工具检测是否可以连接到Modbus设备的IP地址（一般多使用ping命令）。

连接模拟软件

Modbus的仿真工具很多，使用符合标准的哪一种Modbus仿真工具都可以。可使用[Modsim32](#)（[点击下载](#)）软件来模拟Modbus设备，作为配对的测试工具、modscan32是一个Modbus协议的验证工具。 [查看Modsim32快速使用](#)

应用配置

Modbus Master 应用配置界面如下：

应用配置信息

采集间隔:

协议类型:

通讯类型:

端口:

波特率:

停止位:

数据位:

流控:

校验:

参数选项	参数描述
采集间隔	Modbus Master和Modbus设备通讯的间隔，默认1000ms
协议类型	RTU/ASCII/TCP
通讯类型	串口/以太网
(串口) 端口	COM1/COM2/... □默认值COM1
(串口) 波特率	1200/2400/4800/9600/19200/38400/57600/115200,默认值9600
(串口) 停止位	1/2，默认值1
(串口) 数据位	7/8，默认值8
(串口) 检验	None/Even/Odd□默认值None
(以太网) IP地址	Modbus设备的IP地址
(以太网) 端口	Modbus设备提供服务的TCP端口

设备模板

在平台上通过Modbus Master应用 [设备模板样例](#)克隆一份到自己账户名下后，参考设备模板样例的格式制作目标Modbus设备的设备模板并上传。模板样例的格式如下图：

COMMENT	名称	描述	型号									
META	BMS	BMS Device	XXX									
COMMENT	名称	描述	单位	读写权限	解析数据类型	数值类型	功能码	寄存器地址[从0开始]	运算系数	数据位偏移[03,04功能码有效]	写功能码[可不填]	字符长度[数据类型为string时有效]
PROP	tag1	测点1		RW	uint16	int	3	100	1			
PROP	tag2	测点2		RW	uint16	int	3	101	1			
PROP	tag3	测点3		RW	uint16	int	3	102	1			
PROP	tag4	测点4		RW	uint16	int	3	103	1			
PROP	tag5	测点5		RW	uint16	int	3	104	1			
PROP	tag6	测点6		RW	uint16	int	3	105	1			
PROP	tag7	测点7		RW	uint16	int	3	106	1			
PROP	tag8	测点8		RW	uint16	int	3	107	1			
PROP	tag9	测点9		RW	uint16	int	3	108	1			
PROP	tag10	测点10		RW	uint16	int	3	109	1			

Modbus Master应用设备模板的字段描述如下表：

META——定义设备的属性型号等

项目	描述
名称	一般用于定义设备当前作用，根据实际情况填写
描述	一般指设备大类，根据实际情况填写
型号	一般指设备型号，根据实际情况填写

PROP——设备中属性点定义，包含寄存器地址，读取方式等等

项目	描述
名称	设备中的属性/标签/变量等的名称
描述	属性等的描述
单位	属性等的单位
读写权限	RO-只读□WO-只写□RW-读写

项目	描述
解析数据类型	对Modbus返回数据进行解码时的数据类型可用: [bit,int8,uint8,int16,uint16,int32,int32_r,uint,uint32_r,float,float_r,double,double_r,string] 其中int32_r uint32_r float_r double_r表示使用内存数据是反向排序 (排序单位是两个字节), 例如: int32的值为A1B2C3D4 int32_r的值为D4C3B2A1
数值类型	设备属性点数值类型[]FreeIOE支持的类型有int, float, string三种类型
功能码	Modbus读取指令码的十进制, 支持01, 02, 03, 04。 01, 02功能码的data_type只能是bit
运算系数	将获取的modbus数据按照数据类型(data_type)进行解析后, 乘以rate作为属性数据。缺省为1
数据位偏移[从0开始]	在03, 04功能码读取寄存器时, 可以指定offset. []01, 02不支持指定offset操作), 在解析数据类型为bit的时候offset是指位偏移数, 解析数据类型为int8,uint8类型时是指字节偏移数 (0或者1)。
写功能码[可不填]	指定写操作的功能码, 默认情况下, 与功能码的对应关系如下: 01 05, 03 06
字符长度[数据类型为string时有有效]	当按照裸字符串进行读写(data_type 为 string或raw)时, 需要指定此长度。

设备添加

添加Modbus设备的界面如下图：



表格中的参数选项描述如下表：

项目	描述
Address	Modbus地址
Device Name	设备功能描述
Device Serial Number	设备序列号, 当前网关下面需要保证序列号唯一。
Template	设备模板
Use gateway sn as prefix for device sn[]	默认选中, 尽量不要去掉勾选, 除非你设备的虚拟号能保证全局唯一。

常见问题

常见问题

From:

<https://wiki.freeioe.org/> - **FreeIOE** 知识库

Permanent link:

<https://wiki.freeioe.org/apps/app00000025?rev=1576469815>

Last update: **2022/07/12 11:29**

