

Modbus 驱动配置

Update 2026-1-24 增加输出强制使用 16 命令配置，增加标签只写配置。

1 概述

本系统提供的 MODBUS 驱动协议层支持 RTU 协议, 通讯层支持 TCP 网络和 RS485 总线通讯。

MODBUS 的串行和 TCP 报文依据 <http://www.modbus.org/specs.php> 中提供的协议文档：

- Modbus_Application_Protocol_V1_1b3.pdf (应用层协议定义)
- Modbus_Messaging_Implementation_Guide_V1_0b.pdf (TCP/IP 通讯层定义)
- Modbus_over_serial_line_V1_02.pdf (RS485/422/232 串行定义)

2 配置 For Windows

例子配置如下：

;modbus config.ini

;mod1cfg.ini

[modbus]

comargs = 127.0.0.1,502 ;通讯参数，自动识别

;TCP 配置 192.168.1.41,502

;格式: ip 地址,端口

;例子: 192.168.1.41,502

;-----

;RS232/485/422 配置格式:

;格式: 通讯口,波特率,数据位,校验方式,停止位

;例子: com1,9600,8,even,1

;COM 口: windows 为 com1, com2, com3, 到从 com255 之一。

;波特率: 2400, 4800, 9600, 19200, 38400,57600 之一。

;校验方式: none, odd, even, mark, space 之一。

;停止位: 1, 1.5, 2 之一

;-----

tagfile = mod1tag.csv ; 标签配置表

interval = 1000 ;[100,86400000]轮询间隔,单位毫秒,大于等于 100,小于 86400000

slavetimeout = 200 ;[100,1000],Slave 设备响应超时,单位毫秒.

;不包括通讯时间,大于等于 100,小于 1000,一般设置 200

logcompkg = no ;[yes,no] 是否输出 dbg 级别的通讯报文到日志,yes 要输出(ioserver 日志级别要配置为 dbg), no 不输出。

frmregs = 120 ; [16,120], 默认值 120, 轮询报文每次通讯最大寄存器数.01,02 命令自动 16 倍。

blockpoll = true ; [true,false], 默认值 false.

; true 表示按照块轮询寄存器(配置可以不连续, 会读取到不需要的寄存器)

; false 表示每次严格按照配置读连续的寄存器。

force16out = true ; [true,false], 默认值 false.

; true 表示强制使用 16 命令输出; 默认值 false 表示单寄存器使用 6 命令, 多寄存器使用

16 命令, fileversion 5.0.9.2 build 2026-1-24 新加

大部分参照例子配置即可。TCP 和 RS485 的 MODBUS 设备只有通讯参数不一样, 其他都一样。

通讯参数用字符串表示, 中间用西文逗号分开, TCP 只有 IP 地址和端口格式。RS485 总线

MODBUS 标准推荐的是 9600 波特率，8 位数据位，偶校验，1 位停止位。

logcompkg 一般配置为 no，只有在需要调试 MODBUS 设备通讯故障时需要开启并配置为 yes。

同时注意 ioserver 日志输出级别要配置为 dbg

3 配置 For Linux

例子配置如下：

;modbus config.ini

[modbus]

;comargs = 192.168.1.8,502 ;通讯参数，自动识别

;TCP 配置 192.168.1.41,502

;格式: ip 地址,端口

;例子: 192.168.1.41,502

;-----

comargs = ttyS0,9600,8,even,1 ;通讯参数，自动识别

;RS232/485/422 配置格式:

;格式: 通讯口,波特率,数据位,校验方式,停止位

;例子: ttyS1,9600,8,even,1

;COM 口: 为 ttyS1, ttyS2, ttyS3, 到从 ttyS1 到 ttyS32 之一。

;波特率: 2400, 4800, 9600, 19200, 38400,57600 之一。

;校验方式: none, odd, even 之一。

;停止位: 1, 2 之一

;-----

tagfile = mod_fh_tag.csv ; 标签配置表

interval = 1000 ;[100,86400000]轮询间隔,单位毫秒,大于等于 100,小于 86400000

slavetimeout = 200 ;[100,5000],Slave 设备响应超时,单位毫秒.

;不包括通讯时间,大于等于 100,小于 1000,一般设置 200

logcompkg = no ;[yes,no] 是否输出通讯报文到日志,yes 要输出, no 不输出。

frmregs = 120 ; [16,120], 默认值 120, 轮询报文每次通讯最大寄存器数. 01,02 命令自动 16 倍。

blockpoll = true ; [true,false], 默认值 false.

; true 表示按照块轮询寄存器(配置可以不连续, 会读取到不需要的寄存器)

; false 表示每次严格按照配置读连续的寄存器。

force16out = true ; [true,false], 默认值 false.

; true 表示强制使用 16 命令输出; 默认值 false 表示单寄存器使用 6 命令, 多寄存器使用

16 命令, fileversion 5.0.9.2 build 2026-1-24 新加

注意: 如果使用了 USB 串口转换器, 那么 COM 口可能为 ttyUSB0,ttyUSB1 等。在 Linux 中查看串口使用如下命令

dmesg | grep tty*

4MODBUS 标签表

按照 MODBUS 的规范, 数据按照寄存器组织, 有 1bit 寄存器和 16bit 的寄存器, 16bit 也叫 WORD 或中文的“字”, 大于一个 WORD 的数据采用连续的多个 WORD 寄存器存储,字节顺序约定为大头格式 (big-Endian), 即高字节在前(低地址)。

寄存器地址约定, 采用 MODBUS MASTER 协议地址, 使用 6 位 10 进制数表示地址:

寄存器地址	说明
1-65536	Coils 输入: 对应 MODBUS 命令 0x01 地址 0-0xFFFF 输出: 对应 MODBUS 命令 0x05/0x0F 地址 0-0xFFFF
100001-165536	Discrete Inputs

	输入：对应 MODBUS 命令 0x02 地址 0-0xFFFF
300001-365536	Input Registers 输入：对应 MODBUS 命令 0x04 地址 0-0xFFFF
400001-465536	Holding Registers 输入：对应 MODBUS 命令 0x03 地址 0-0xFFFF 输出：对应 MODBUS 命令 0x06/0x10 地址 0-0xFFFF

为了便于配置，本系统约定 MODBUS 的数据类型如下：配置是也可以填写括号里的类型，比如 i2 等价于 int16

MODBUS 数据类型	描述
bit	1 位，MODBUS MASTER 协议寄存器地址为 1-65536 和 100000-165536 使用。
i2(int16)	16 位带符号整数,占 MODBUS MASTER 协议寄存器地址为 300000-365536 和 400000-465536 一个地址
ui2(uint16)	16 位无符号整数,占 MODBUS MASTER 协议寄存器地址为 300000-365536 和 400000-465536 一个地址
i4(int32)	32 位带符号整数,占 MODBUS MASTER 协议寄存器地址为 300000-365536 和 400000-465536 连续两个地址
ui4(uint32)	32 位无符号整数,占 MODBUS MASTER 协议寄存器地址为 300000-365536 和 400000-465536 连续两个地址
f4(float)	32 位 IEEE754 浮点数,占 MODBUS MASTER 协议寄存器地址为 300000-365536 和 400000-465536 连续两个地址
f8(double)	64 位 IEEE754 浮点数,占 MODBUS MASTER 协议寄存器地址为 300000-365536 和 400000-465536 连续四个地址
i8(int64)	64 位带符号整数,占 MODBUS MASTER 协议寄存器地址为 300000-365536 和 400000-465536 连续四个地址

ui8(uint64)	64 位无符号整数,占 MODBUS MASTER 协议寄存器地址为 300000-365536 和 400000-465536 连续四个地址
-------------	---

标签表的配置参见例子，其中 A1 单元格为文件格式标识字符串“modbustags”，不能更改，否则 dac_modrtu.dll 驱动不认。

配置列定义：

UnitID	设备地址，总线上唯一。
RegAddr	寄存器地址，6 位 10 进制表示法。
DataType	MODBUS 设备的数据类型。
RdbTagName	实时库表签名
RdbDT	实时库数据类型
EngUnits	工程单位
Description	描述
byteorder	4 字节和 8 字节数据的字节顺序(从 2020.7 版开始两字节的数据也支持位置交换)，不填写为默认模式， 填写 inverse 表示 big-endian 模式(即 modbus_slave 工具中的 float_inverse, long_inverse 等) float 为例： 默认模式: LH LL HH HL 或 modbus slave(CD AB) inverse 模式: HH HL LH LL 或 modbus slave(AB CD) 从 2021.11 版开始，增加下面 6 种模式： WORD 两种模式： 12 : LH; 小头 WORD，等于原 inverse 模式 21 : HL; 大头 WORD，原默认模式 双 WORD,4 种模式：适合 int32, float32 1234 : LL LH HL HH;

2143	: LH LL HH HL; 原默认模式, modbus slave(CD AB)
4321	: HH HL LH LL; 原 inverse 模式 modbus slave(AB CD)
3412	: HL HH LL LH;
四 WORD,4 种模式; 适合 int64, float64	
1234	: DWL(LL LH HL HH) DWH(LL LH HL HH), 即字节顺序为“12345678”;
2143	: DWL(LH LL HH HL) DWH(LH LL HH HL) , 即字节顺序为“21436587”;
原默认模式, modbus slave(GH EF CD AB)	
4321	: DWH(HH HL LH LL) DWL(HH HL LH LL) , 即字节顺序为“87654321”;
原 inverse 模式 modbus slave(AB CD EF GH)	
3412	: DWH(HL HH LL LH) DWL(HL HH LL LH) , 即字节顺序为 “78563412”;

MODBUS 的例子标签表如下图：

Linux 和 windows 版的标签表是完全一样的。

UnitID	RegAddr	Data Type	RdbTagName	RdbDT	EngUnits	Description	byteorder	access
1	300001	i2	mod1.i2toi4	int32		input		
1	300002	i2	mod1.i2tof4	float32	℃	input		
1	300003	i4	mod1.i4toi4	int32		input		
1	300005	f4	mod1.f4tof4	float32		input		
1	300007	f8	mod1.f8tof8	float64	m	input		
1	300011	i8	mod1.i8toi8	int64	kg	input		
1	400001	i2	mod1.i2tof4_rw	float32		rwtest		
1	400002	i2	mod1.i2tof4_w1	float32		onlywrite1		ow
1	400003	i2	mod1.i2tof4_rw2	float32		rwtest2		
1	400004	i2	mod1.i2tof4_w2	float32		onlywrite2		ow
1	400005	i2	mod1.i2tof4_rw3	float32		rwtest3		
1	400006	i2	mod1.i2tof4_rw4	float32		rwtest4		
1	1	1 bit	mod1.bit1_rw	digital		bit read/write		
1	100001	bit	mod1.bit1	digital		bit read only		

从 2020.7 版开始，modbus 驱动支持驱动级别的位分解。寄存器后面用小数点分开后跟 bit 位，0 表示 bit0,比如 300050.0 表示 bit0 位, 300050.3 表示 bit3 位，modbus 类型填写 bit，实时库类型填写 didital,如下

1,300050.0,bit,mod1.bit5000,digital,,300050

1,300050.1,bit,mod1.bit5001,digital,,300050

经过以上配置后，`rdbdac_ux` 会自动在 MODBUS 的数据类型和实时库数据类型之间转换。但是，很多 MODBUS 设备采用定点数据约点小数点方式表示浮点数。比如 MODBUS 使用 16 位整数（即一个字）值 `12345`，来表示小数 `123.45`，这是通过以上标签表就不能正确转换，还需要配合表达式转换表来实现，详见 `loserver` 使用说明中关于采集值和工程值的转换。

`fileversion 5.0.9.2 build 2026-1-24` 版在最后增加了 `access` 列，不配置表示可读写，配置 `ow` 表示只写(不会读取，一样可以映射到实时库使用 `rdb_writedev` 命令控制输出。