

Modbus通信协议

1. 文档修改记录

版本号	修改记录	日期
V4.14.1	-	-
V4.14.2	Add: 1. 10054: 是否有无线网卡 2. 30200: 网卡状态 3. 300202~30207: 网卡MAC地址	2022/10/19
V4.14.3	Add: 输入寄存器: 1. 30320: 左雷达 2. 30321: 右雷达 3. 30322: 感知相机 4. 30350~30365: 叉车IO输入设备 5. 30170-30179: 自定义名称	2022/10/24
V4.14.4	Add: 输入寄存器: 1. 30056 :内核类型 2. 30162-30165 :SRC/SRTOS的版本号	2022/10/26
V4.14.5	Add: 线圈寄存器: 1、叉臂上升 0021 2、叉臂下降 0022 3、模组超限复位 0023	2022/10/31
V4.14.6	Add: 线圈寄存器: 1、货物掉落监控复位按钮（实现多次掉落监控）	2022/10/31
V4.14.7	Add: 寄存器偏移地址配置	2023/09/13

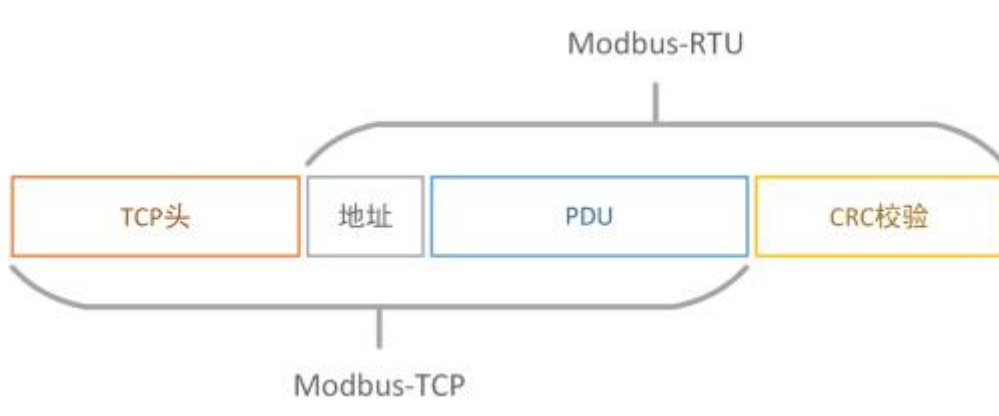
V4.14.8	Remove: 对内部使用的寄存器与调度相关, 已经移动到触摸屏v2的文档中	2024/01/24
V4.14.9	Add: [华为24年框架]需求导入的寄存器定义, 如偏载/超载传感器,网络状态,货架号,设备寿命	2024/04/01
V4.14.10	Remove: 重新定义[华为24年框架]需求导入的寄存器地址, 并清理了其它占用了EAC专用地址段的寄存器地址定义	2024/05/30
V4.14.11	Remove: 30400~之后未使用的设备状态寄存器地址 Improve: 优化所有表格表头显示; 所有未使用到的地址绿色标明增加可读性	2024/06/03
V4.14.12	定义调度模式: <ul style="list-style-type: none"> 手动模式 (原单机模式) 自动模式 (原调度模式) 维保模式 (触摸屏独占) 	2024/07/17
v4.14.13	<ol style="list-style-type: none"> 更新寄存器 30001值枚举值, 新增值0x13、0x15、0x16 更新10052 ~ 10055的描述 	2024/08/08
v4.14.14	支持下发路径, 执行路径功能	2024/08/23
v4.14.15	支持获取车辆类型, 对应寄存器30166	2024/09/25
v4.14.16	<ul style="list-style-type: none"> 关机状态 当前时区 	2025/01/02
v4.14.17	ADD: 动作133扫描到的二维码信息, 对应寄存器[30182,30189]	2025/01/03
v4.14.18	30073 - 机器人运动状态	2025/03/25
v4.14.19	30091 - 工位避障触发源	2025/07/01
v4.14.20	废弃: 30025 硬件错误码 30081 硬件错误码 1 30083 硬件错误码 2 30085 硬件错误码 3 30087 硬件错误码 4 30089 硬件错误码 5	2025/07/02
v4.14.21	Fix: 10133地址位文档书写错误	2025/07/28

注：更新记录请将删除内容也标识出来，使用 XXXX

2. 协议概述

Oasis系列目前同时支持Modbus RTU和TCP两种通信形式，用户可根据需要进行选择。在通信协议中，用户作为Client端主动连接到AGV进行查询，AGV作为Server端对查询指令进行处理和响应。

典型的Modbus协议帧如下图所示，其中Modbus-TCP协议帧由TCP头+地址+PDU组成，Modbus-RTU由地址+PDU+CRC校验组成。



Modbus的PDU（Protocol Data Unit，协议数据单元）由功能码+寄存器地址+寄存器数量（可选）+寄存器值组成，PDU在TCP和RTU两种形式上是相同的。



Modbus 为一问一答协议，发送一个请求后要等待回复才能发送第二个请求。对于 Modbus 协议格式的详细介绍，可以参考阅读 [Modbus 官网的权威资料](#)

工具与使用

使用前，请使用modbus poll调试工具，测试modbus调用是否正常，请先学习以下视频：

[modbus poll调式工具控制AGV教程.mp4](#)

以下为modbus调试工具，我们使用modbuspoll调试：

[modbus调试工具.zip](#)

以下为modbus的SDK，为python版本：

[sr-modbus-sdk-py.zip](#)

3. 连接AGV

- Modbus-RTU的物理接口为串口，如要启用Modbus-RTU功能，需要先将参数 modbus.enable_modbus_rtu 设置为True，串口设备（modbus.modbus_rtu_device）、通信波特率（modbus.modbus_rtu_baud_rate）和奇偶校验（modbus.modbus_rtu_parity）可通过参数进行配置。串口数据位固定为8位，停止位固定为1位。Modbus-RTU的Slave ID固定为17。
- Modbus-TCP的物理接口为以太网，如要启用Modbus-TCP功能，需要先将参数 modbus.enable_modbus_tcp 设置为True，其中端口号可通过参数modbus.modbus_tcp_port 进行配置，如无特殊需要，端口号不建议进行修改。

参数ID	参数key	参数名	默认值
2101	modbus.enable_modbus_tcp	是否启用 Modbus-TCP	False
2102	modbus.enable_modbus_rtu	是否启用 Modbus-RTU	False
2103	modbus.modbus_rtu_device	Modbus-RTU 通信设备名称	/dev/ttyUSB0
2104	modbus.modbus_rtu_baud_rate	Modbus-RTU 通信设备波特率	115200
2105	modbus.modbus_tcp_port	Modbus-TCP 端口号	502
2106	modbus.modbus_rtu_parity	modbus-RTU 通信的奇偶校验	NoneParity

- 其中配置项Modbus-RTU通信设备名称中配置值与物理接口对应关系如下：

RTU通信设备名称	对应物理接口
/dev/ttyUSB0	插入 VC300 USB 口的 USB 转串口模块
/dev/ttyTHS1	VC300 上的 RS232-1 接口（RS232-1 为复用端口，如果该端口已连接触摸屏设备，那么只能通过 USB 转串口模块进行 Modbus 通信，如果一定要使用该端口请关闭触摸屏和 LMNS 两项配置）

查看或设置AGV modbus寄存器设置偏移地址：

注意一定要跟plc端设置的起始偏移地址保持一致，否则会出现通信异常。

序号	名称	ID	key	当前值	默认值	取值范围	描述
1	线圈寄存器的起始地址	2110	modbus.start_bits_addr	00001	00001	[0,50000]	线圈寄存器的起始地...
2	离散输入寄存器的起始地址	2111	modbus.start_input_bits_ad...	10001	10001	[0,50000]	离散输入寄存器的起...
3	输入寄存器的起始地址	2112	modbus.start_input_registe...	30001	30001	[0,50000]	输入寄存器的起始地...
4	保持寄存器的起始地址	2113	modbus.start_registers_addr	40001	40001	[0,50000]	保持寄存器的起始地...

- 查看AGV modbus配置参数，按连接方式修改配置参数：

- 主页
- 主界面
- 地图
- 地图列表
- 任务
- 任务列表
- 任务模板
- 任务计划
- 系统
- 系统设置
- 配置参数**
- 账号管理
- 连接管理
- 系统维护
- 调试
- 硬件状态

Matrix > 配置参数

参数列表

device

1944 eac通信类型 modbus_rtu

device.eac_communication_type

1945 扩展动作控制器的modbus通信地址 1

device.eac_slave_id

modbus

2101 是否启用modbus-TCP True

modbus.enable_modbus_tcp

2102 是否启用modbus-RTU True

modbus.enable_modbus_rtu

2103 modbus-RTU通信设备名称 /dev/ttyTHS1

modbus.modbus_rtu_device

2104 modbus-RTU通信设备波特率 115200

modbus.modbus_rtu_baud_rate

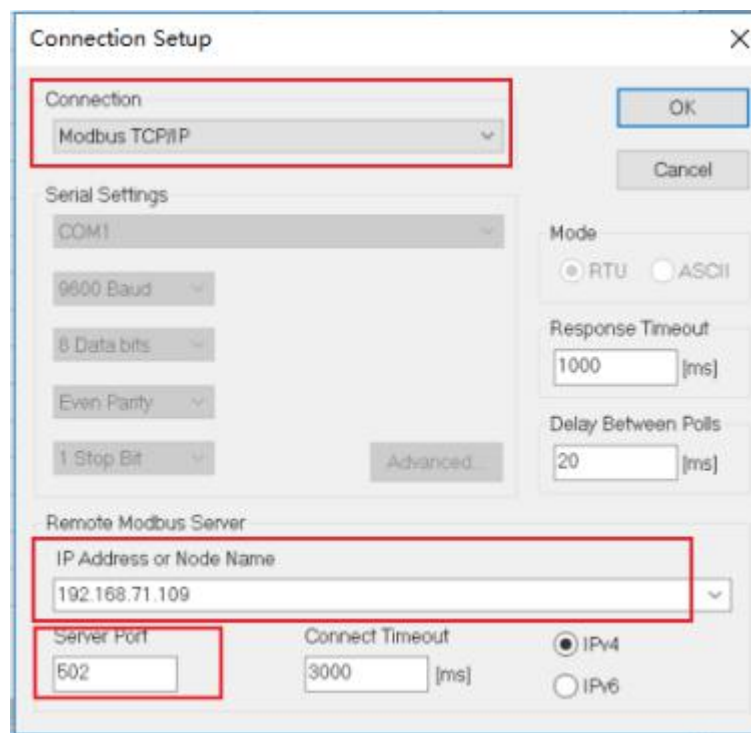
2105 modbus-TCP端口号 502

modbus.modbus_tcp_port

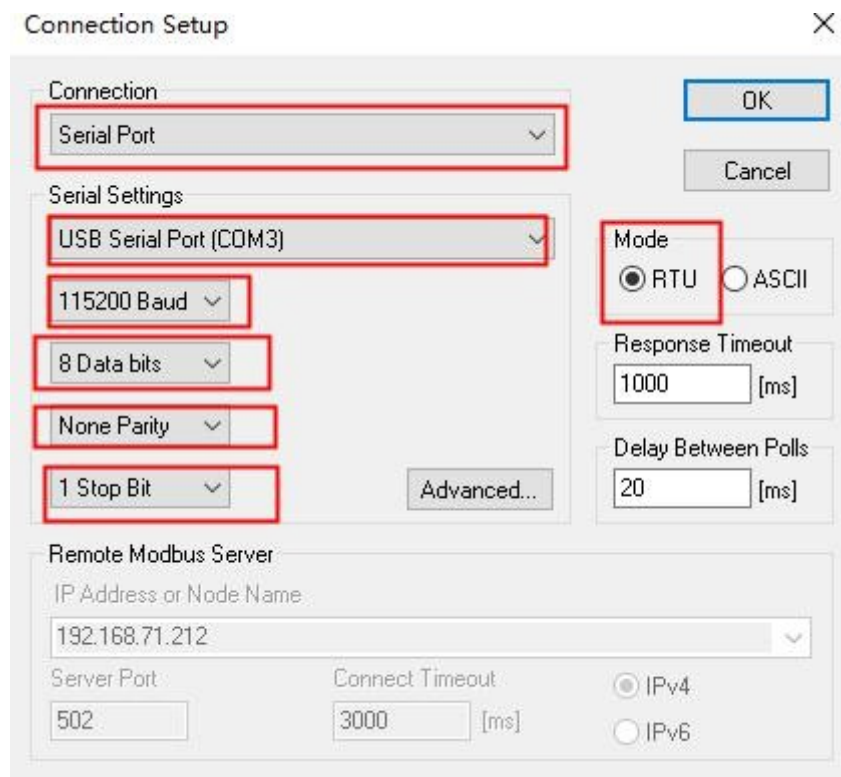
2106 modbus-RTU通信的奇偶校验 NoneParity

modbus.modbus_rtu_parity

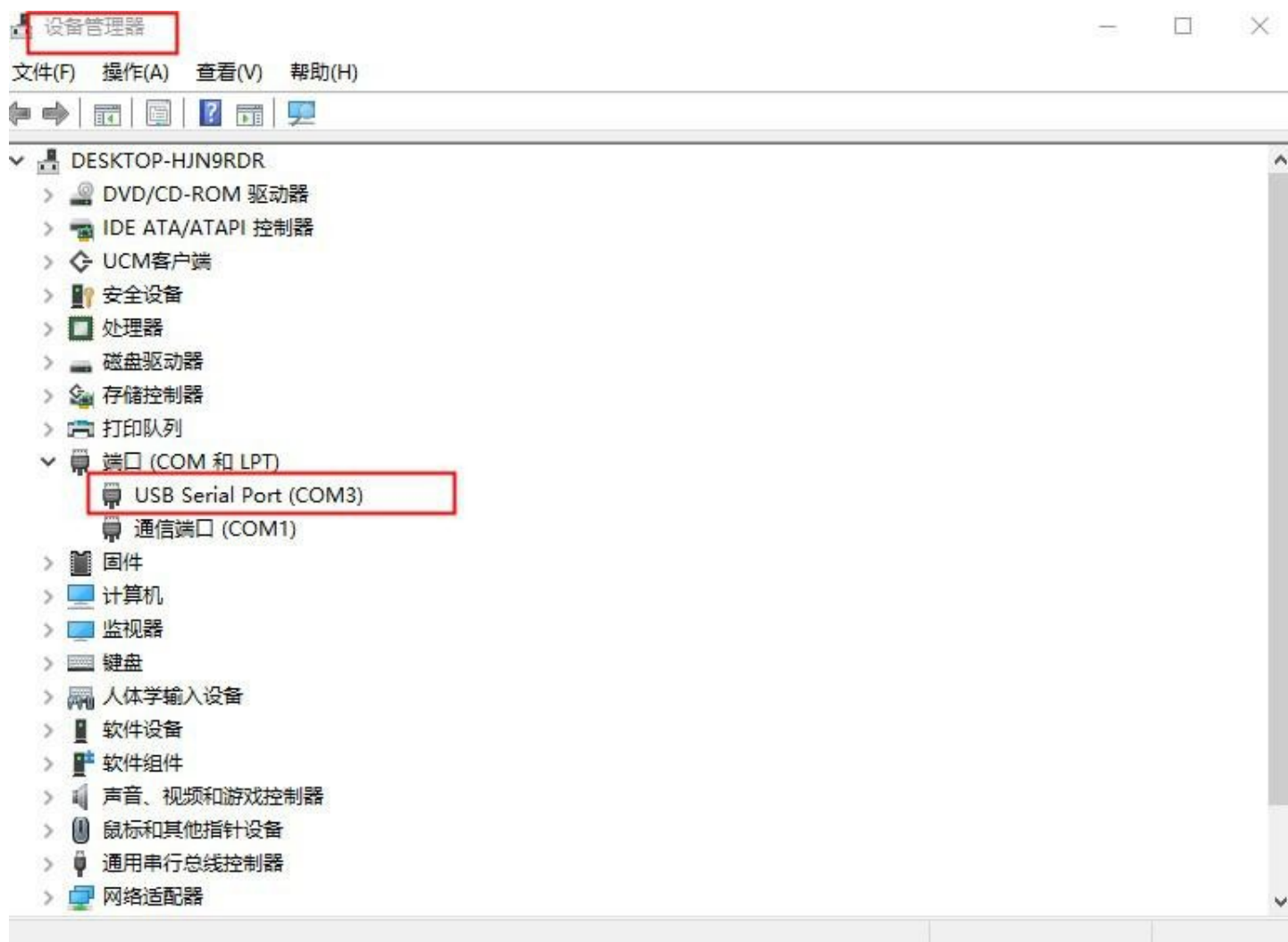
- modbus-tcp连接



- modbus-rtu连接



- 使用modbus-rtu连接，例如插入USB查看串口。

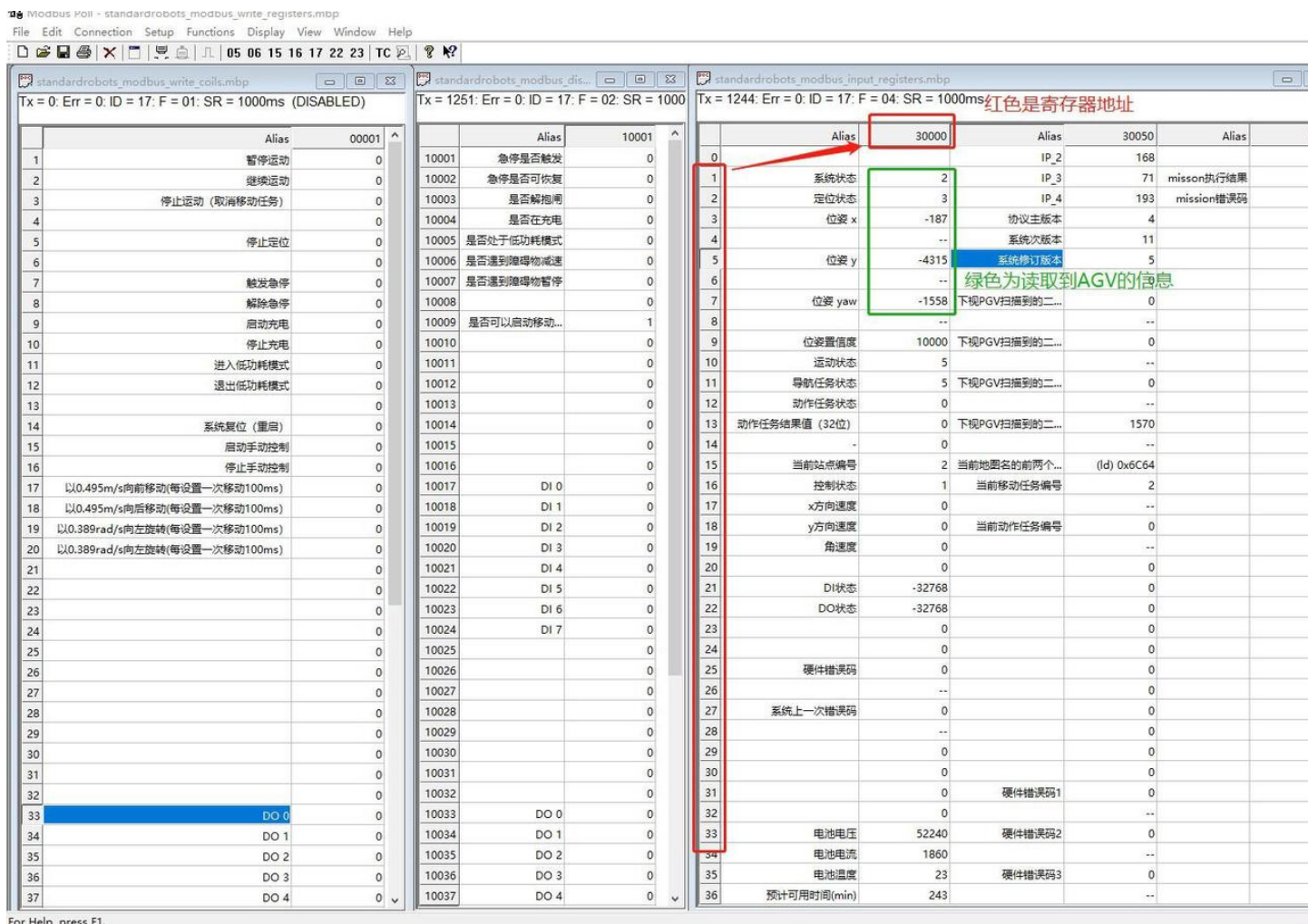


- 使用modbus打开附件中的四个文件

modbus调试工具 > 附件				
名称	修改日期	类型	大小	
standardrobots_modbus_discrete_in...	2019/6/10 12:08	Modbus Poll File	4 KB	
standardrobots_modbus_input_regist...	2019/6/13 1:09	Modbus Poll File	5 KB	
standardrobots_modbus_write_coils....	2019/6/10 11:10	Modbus Poll File	4 KB	
standardrobots_modbus_write_regist...	2019/6/10 12:08	Modbus Poll File	3 KB	

使用modbus poll打开附件中的这四个文件，可以看到modbus poll读取到AGV的信息

- 在打开的文件中可以看到AGV当前系统信息



4. 协议内容

根据Modbus协议，数据以寄存器的形式进行传送。AGV通信协议根据传送的数据类型将其分为4类，对应4种不同的寄存器，如下表所示：

寄存器类型	地址范围	读写属性	数据类型
离散输入寄存器	10001~10100	只读	简单的开关量状态，如是否处于急停
输入寄存器	30001~30120	只读	数值类型的状态，如系统状态、电量
线圈寄存器	00001~00100	可读可写	简单的开关量控制，如暂停运动
保持寄存器	40001~40100	可读可写	数值类型的控制指令，如移动到站点 / 位姿

注意：

- 只有本协议中列出来的寄存器才允许客户使用，那些未列出来的寄存器可能是预留的，也可能有其他的作用只是未对外开放，强制写入可能会带来一系列不可控的行为。

- 对于数值型寄存器，Modbus默认单个寄存器大小为2字节（即16 bit），由于部分数值的类型为32位整形数字，无法保存在单个Modbus寄存器中。在AGV通信协议中，将32位整形数字以大端序保存在连续的两个寄存器中。
- 例如对于整型数字0x11223344，其在Modbus寄存器中保存的格式如下表所示。

寄存器地址k		寄存器地址k+1	
高字节	低字节	高字节	低字节
0x11	0x22	0x33	0x44

- 协议中涉及到的数值类型、占用寄存器个数及说明如下所示。

类型	占用寄存器个数	说明
uint16	1	无符号 16 位整型数值
int16	1	有符号 16 位整型数值
enum/uint16	1	无符号 16 位整型枚举值
uint32	2	无符号 32 位整型数值，大端序（Big Endian）
int32	2	有符号 32 位整型数值，大端序（Big Endian）

- 寄存器的偏移地址是可以改变的，配置如下：

参数ID	参数key	参数名	默认值
2110	modbus.start_bits_addr	线圈寄存器的起始地址	00001
2111	modbus.start_input_bits_addr	离散输入寄存器的起始地址	10001
2112	modbus.start_input_registers_addr	输入寄存器的起始地址	30001
2113	modbus.start_registers_addr	保持寄存器的起始地址	40001

4.1 线圈状态（可读可写）

下列寄存器使用功能码 05 （ Write Single Coil ） / 15 （ Write Multiple Coils ） 进行写入，功能码 15 只支持一次写一个线圈，否则会报错。可以读取线圈状态，但读取的值为上次写入的值，并无实际意义。

寄存器地址	描述	取值
00001	暂停运动	写入 0xFF00 使能
00002	继续运动	写入 0xFF00 使能
00003	停止运动	写入 0xFF00 使能
00004	可用	
00005	停止定位	写入 0xFF00 使能
00006	屏蔽避障区域	0xFF00： 开启屏蔽蔽障模式 0x0000： 退出屏蔽蔽障模式
00007	触发急停	写入 0xFF00 使能
00008	解除急停	写入 0xFF00 使能
00009	启动充电	写入 0xFF00 使能
00010	停止充电	写入 0xFF00 使能
00011	进入低功耗模式	写入 0xFF00 使能
00012	退出低功耗模式	写入 0xFF00 使能
00013	可用	
00014	系统复位（重启）	写入 0xFF00 使能
00015	启动手动控制	写入 0xFF00 使能
00016	停止手动控制	写入 0xFF00 使能
00017	以 0.495m/s 向前移动（每设置一次移动 10 0ms，此处是提供给开关专用，不支持速度配置，推荐使用 40022-40024 寄存器进行手动控制）	写入 0xFF00 使能
00018	以 0.495m/s 向后移动（每设置一次移动 10 0ms）	写入 0xFF00 使能

00019	以 0.389rad/s 向左旋转 (每设置一次移动 1 00ms)	写入 0xFF00 使能
00020	以 0.389rad/s 向右旋转 (每设置一次移动 1 00ms)	写入 0xFF00 使能
00021	以 0.1m/s 上升叉臂 (每设置一次动作 100ms)	写入 0xFF00 使能 (待开发)
00022	以 0.1m/s 下降叉臂 (每设置一次动作 100ms)	写入 0xFF00 使能 (待开发)
00023	模组超限复位	写入 0xFF00 使能 (sros5.51.2及以上支持) 如：模组电机回零
00024	载重传感器数据清零	
[00025, 00032]	可用	
[00033, 00040]	DO 0~7	写入 0xFF00 使能
[00041, 00048]	可用	
00049	放行，即：发送此信号动作 (131,0,0) 会结束	写入 0xFF00 使能
00051	调度模式	0xFF00：进入调度模式 0x0000：退出调度模式
[00052, 00096]	可用	
00097	暂停 mission 任务	写入 0xFF00 使能
00098	继续 mission 任务	写入 0xFF00 使能

00099	取消 mission 任务	写入 0xFF00 使能
[00100, 00123]	VSC300_DO1 0 - VSC300_DO5 1	写入 0xFF00 使能
[00124, 10000)	可用	

4.2 离散量输入状态（只读）

下列寄存器使用功能码 02 （ Read Discrete Inputs ） 进行读取。推荐一次性全部读取，因Modbus一次最多只支持读取125个寄存器，超过125个寄存器，需要分批读取。

寄存器地址	描述	取值
10001	急停是否触发	1 表示有效， 0 表示无效
10002	急停是否可恢复	1 表示有效， 0 表示无效
10003	是否解抱闸	1 表示有效， 0 表示无效
10004	是否正在充电	1 表示有效， 0 表示无效
10005	是否处于低功耗模式	1 表示有效， 0 表示无效
10006	是否遇到障碍物减速	1 表示有效， 0 表示无效
10007	是否遇到障碍物暂停	1 表示有效， 0 表示无效
10008	可用	
10009	当前是否可以运行移动任务，同时满足一下条件时有效。系统空闲、没有急停、没有解抱闸、不是低电量模式、定位成功、非手动控制模式、没有移动任务或是移动任务已经结束、雷达、VSC、MOTOR1、MOTOR2、SRC都正常	1 表示有效， 0 表示无效
[10010, 10016]	可用	
[10017, 10024]	DI 0~7	1 表示有效， 0 表示无效

[10025, 10032]	可用	
[10033, 10040]	DO 0~7	1 表示有效，0 表示无效
[10041, 10048]	可用	
10049	放行信号，即：动作（131,0,0）会等待此信号	1 表示有效，0 表示无效
10050	可用	
10051	是否处于自动模式	1 表示有效，0 表示无效
10052	上层调度系统是否掉线【废弃】	true是掉线，false是在线或者没有开启调度模式
10053	可用	
10054	车辆载荷（载货）状态	true是有载荷，false是空载
10055	是否有无线网卡	1表示有效，0表示无效
[10056, 10059]	可用	
[10060, 10123]	VSC300_DI1 0 - VSC300_DO5 1	1 表示有效，0 表示无效
[10124, 10132]	1353 9个IO	1 表示有效，0 表示无效
[10133, 10135]	可用	
10136	工位避障信号，matrix绘制类型为365的工位避障区，agv和障碍物都在该区域内触发避障信号	1表示有障碍，0表示没有
[10137, 30000)	可用	

4.3 输入寄存器（只读）

下列寄存器使用功能码 04 （ Read Input Register ） 进行读取。推荐一次性全部读取。

寄存器地址	描述	类型	取值	单位
30001	系统状态	enum/ int16	0x01: 系统正在初始化 0x02: 系统空闲 0x03: 系统出错 0x04: 正在启动定位 0x05: 导航正在初始化 0x06: 导航正在寻路 0x07: 正在等待到达目标位置 0x08: 检测到障碍 , 减速 0x09: 导航正在重新寻路 0x0A: 遇到障碍暂停运动 0x0B: 无法抵达目标位置 0x0E: 正在初始化执行固定路径 0x0F: 正在等待固定路径执行结束 0x10: 执行固定路径检测到障碍 , 减速前进 0x11: 执行固定路径遇到障碍暂停运动 0x12: 无法检测到目标站点 0x13: 执行固定路径时用户暂停了 0x15: 导航过程中出错, 如: 偏离路径 0x16: 硬件错误	
30002	定位状态	enum/ int16	0x01: 定位未启动 0x02: 定位正在初始化 0x03: 定位成功 0x04: 正在重定位 0x05: 定位错误, 需要重新启动定位	
30003	位姿 x	int32		mm
30005	位姿 y	int32		
30007	位姿 yaw	int32		rad*1 000

30009	位姿置信度	uint16		0.01 %
30010	移动任务状态【废弃】	enum/uint16	0x00: 无效状态 0x01: 未运动 0x02: 正在正常运动中 0x03: 手动暂停 0x04: 遇到障碍物减速运行 0x05: 遇到障碍物暂停运行	
30011	导航任务状态【废弃】 请用寄存器 30113 替代	enum/uint16	0x00: 无效状态 0x02: 等待开始执行 0x03: 正在执行 0x04: 暂停执行 0x05: 执行结束 0x06: 正在取消	
30012	动作任务状态【废弃】 请用寄存器 30129 替代	enum/uint16	0x00: 无效状态 0x02: 等待开始执行 0x03: 正在执行 0x04: 暂停执行 0x05: 执行结束 0x06: 正在取消	
30013	动作任务结果值【废弃】 请用寄存器30139替代	uint32		
30015	当前站点编号	uint16		
30016	操作状态	enum/uint16	0x00 状态不可用 0x01 自动控制模式 0x02 手动控制模式	
30017	x 方向线速度	int16		mm/s
30018	y 方向线速度	int16		
30019	角速度	int16		(1/1000)ra

				d/s
30020	调度模式	enum/uint16	0x01 手动模式 0x02 自动模式 0x03 维保模式	
30021	DI 状态	uint16		
30022	DO 状态	uint16		
[30023, 30024]	可用			
30025	硬件错误码 【废弃】	uint32		
30027	系统上一次错误	uint32		
[30028, 30032]	可用			
30033	电池电压	uint16		mV
30034	电池电流	int16	负数为充电电流	mA
30035	电池温度	int16		°C
30036	电池预计使用时间	uint16		min
30037	当前电量百分比	uint16	[1, 100]	
30038	当前电池的状态	enum/uint16	0x00: 无效状态 0x02: 正在充电 0x03: 未充电	
30039	电池循环次数	uint16	电池充放电循环使用了的次数	
30040	电池标称容量	uint16		mAh
30041	运动总里程	uint32		m
30043	开机总时间	uint32		s
30045	开机总次数	uint32		次
30047	系统当前时间	uint32	Linux 时间戳	
[30049,	IP.1~4	uint16	对外通信IP地址，比如192.168.71.50	

30052]				
30053	系统主版本号	uint16		
30054	系统次版本号	uint16		
30055	系统修订版本号	uint16		
30056	内核版本	uint16	0:无 1: TK1 2: NXP	
30057	下视 PGV 扫描到的 二维码 ID	int32	需要有下视 pgv 备注：下视扫码同样显示	
30059	下视 PGV 扫描到的 二维码 x	int32		
30061	下视 PGV 扫描到的 二维码 y	int32		
30063	下视 PGV 扫描到的 二维码 yaw	int32		
30065	当前地图名的前两 个字节编码	uint16	比如当前地图名为：“1aa”，那么此寄存器的值为： 0x3161	
30066	当前移动任务编号 【废弃】	int32		
30068	当前动作任务编号 【废弃】	Int32		
30070	当前系统音量	uint16	1-100	
30071	关机状态	enum/u int16	0x00: 无效 0x01: 车辆即将关机	
30072	当前时区	int16	系统所在时区	
30073	机器人运动状态	enum/u int16	0x00: 未知状态 0x01: 静止 0x02: 前进 0x03: 后退 0x04: 左转	

			0x05: 右转 0x06: 原地左转 0x07: 原地右转 0x08: 左移 0x09: 右移	
[30074, 30080]	可用			
30081	硬件错误码 1【废弃】	uint32		
30083	硬件错误码 2【废弃】	uint32		
30085	硬件错误码 3【废弃】	uint32		
30087	硬件错误码 4【废弃】	uint32		
30089	硬件错误码 5【废弃】	uint32		
30091	工位避障触发源	uint32	按bit位分布 0x01 主雷达 0x02 辅雷达 0x04 深度相机1 0x08 深度相机2 0x10 深度相机3 0x20 深度相机4 0x40 避障雷达1 0x80 避障雷达2 0x100 避障雷达3 0x200 避障雷达4	
[30093, 30096]	可用			
30097	当前正在运行的 mission id	uint32		

30099	Mission 运行状态	enum/ int16	0x00: 无效状态 0x02: 任务在队列中，但是又还没有启动的状态 0x03: 正在执行 0x04: 暂停执行 0x05: 执行结束 0x06: 正在取消	
30100	Mission 执行结果	enum/ int16	0x00: 无效状态 0x01: 任务执行成功 0x02: 任务取消 0x03: 任务执行出错	
30101	Mission 错误码	uint32		
[30103, 30112]	可用			
30113	移动任务状态	enum/ int16	0x00: 无效状态 0x02: 等待开始执行 0x03: 正在执行 0x04: 暂停执行 0x05: 执行结束 0x06: 正在取消 0x08: 交通管制	
30114	当前移动任务 no	int32		
30116	当前移动任务目标 站点	uint16		
30117	当前路径编号，移 动任务运行过程中 有效	uint16		
30118	当前移动任务起始 站点	uint16		
[30119, 30121]	可用			
30122	移动任务结果	enum/ int16	0x00: 无效状态 0x01: 任务执行成功	

			0x02: 任务取消 0x03: 任务执行出错	
30123	移动任务结果值	uint32		
[30125, 30128]	可用			
30129	动作任务状态	enum/uint16	0x00: 无效状态 0x02: 等待开始执行 0x03: 正在执行 0x04: 暂停执行 0x05: 执行结束 0x06: 正在取消	
30130	当前动作任务 no	int32		
30132	当前动作任务 ID	int32		
30134	当前动作任务参数 0	int32		
30136	当前动作任务参数 1	int32		
30138	动作任务结果（执行失败时，请从30139寄存器读取结果）	enum/uint16	0x00: 无效状态 0x01: 任务执行成功 0x02: 任务取消 0x03: 任务执行出错	
30139	动作任务结果值	int32		
[30141, 30150]	可用			
30151	电机顶升高度	int16	[0, 200]，单位mm	
[30152, 30159]	可用			
30160	叉臂当前高度	uint16		
30161	叉车舵角	uint16		
30162	底盘控制器类型	uint16	0: 无 1: src	

2: SRTOS

30163	底盘控制器系统主版本	uint16		
30164	底盘控制器系统次版本	uint16		
30165	底盘控制器系统修订版本	uint16		
30166	车辆类型	uint16	0 : Error-type 101 : Oasis-300E 102 : Oasis-300EL 103 : Oasis-600E 104 : Oasis-600EL 105 : Oasis-1200E 106 : Oasis-SMT-E 201 : ELFIN-600E-SRD 301 : Gulf-1400-CDD 302 : Gulf-1500-CPD 303 : Gulf-3000-QDD 304 : Gulf-MSL14 305 : Gulf-MP15 306 : Gulf-MD20 307 : Gulf-KIOXIA 308 : Gulf-QZ16	
[30167, 30169]	可用			
[30170, 30181]	自定义名称-0	int16	备注：【AGV自定义名称】：最大支持20个字符，通过0-9号寄存器实现，用二进制ASCII码表示.[小端模式]	
30182	动作133扫描到的二维码ID	int32	备注：执行动作133后的二维码信息反馈 表示：下面信息均为整数，小数部分不显示	
30184	动作133扫描到的二维码x	int32		

30186	动作133扫描到的二维码y	int32		
30188	动作133扫描到的二维码yaw	int32		
[30190, 30199]	货架号, 二维码上的字符串	int16	最大支持20个字符，通过10个寄存器实现，用二进制ASCII码表示。 [小端模式]	
[30200, 30207]	无线网卡状态	int16		
[30208, 30218]	EAC	int16		
[30219, 30229]	可用		IRA_EAC_HEART_SEQ = 207; IRA_EAC_ACTION_CONTROL_CMD = 208; IRA_EAC_ACTION_ID = 209;	
30230	网络信号强度信息	int16		
30231	网络设备接入厂商	uint16_t	0-网络切换为WiFi，1-网络切换为以太网（5G、节点通、多倍通、有人、摩沙）	
[30232, 30299]	可用			
30300 ~ 30322	设备状态寄存器	int16	地址不符合约定，故废弃	
30350 ~ 30363	部分叉车传感器使用	int16	地址不符合约定待修复	
30370 ~ 30379	叉车预留输入寄存器	int16	30300-30400给EAC，故废弃	
30380 ~ 30389	叉车预留输出寄存器	int16	30300-30400给EAC，故废弃	
[30300, 30399]	EAC专用			
[30400, 30499]	安全PLC/手控盒专用			

[30501, 30600]	自定义专用，用户可任意修改			
[30599, 30639]	可用			
30640	载重传感器数据显示	int16	有正负 const uint16_t DOC_RESET_SENSOR = 23; // 载重传感器数据清零	
30641	偏载位置数据 坐标x	int16	车前+x	mm
30642	偏载位置数据 坐标y	int16	车左+y	
[30643, 30645]	称重传感器结果	int16		
[30646, 30699]	可用			
[30700, 30740]	关键元器件寿命维保状况	int16	多个设备	
[30741, 30748]	电池序列号 1			
[30749, 30756]	电池序列号 2			
[30757, 30764]	电池序列号 3			
[30765, 30772]	电池序列号 4			
30773	电池电压	uint16		mV
30774	电池电流	int16	负数为充电电流	mA
30775	电池温度	int16		°C
30776	电池预计使用时间	uint16		min
30777	当前电量百分比	uint16	[1, 100]	
30778	当前电池的状态	enum/ uint16	0x00: 无效状态 0x02: 正在充电	

			0x03: 未充电	
30779	电池循环次数	uint16	电池充放电循环使用了的次数	
30780	电池标称容量	uint16		mAh
30781	电池剩余容量（规划中）	int16	预留位	
30782	电池学习容量（规划中）	int16	预留位	
30783	电池保护状态（规划中）	int16	预留位	
[30784, 40000)	可用			

4.4 保持寄存器（可读可写）

下列寄存器使用功能码 06（Write Single Register）、16（Write Multiple Registers）进行写入，读取的值为上次写入的值，并无实际意义。

由于所需要输入的信息字节数较多，部分功能需要同时写入多个寄存器才可生效，例如通过位姿定位，需要同时设置 40001、40002、40003、40005、40006 这 6 个寄存器才可触发通过位姿定位功能。

1	* ## 地址段分配	
2	* .	
3	* —— [0, 999]/	# SROS专用地址
4	* —— [300, 400]/	# 扩展动作控制器专用
5	* —— [400, 500]/	# 安全PLC/手控盒专用
6	* —— [500, 600]/	# 自定义专用，用户可任意修改
7	* —— [5000, 5999] /	# src 寄存器
8	* —— [5500, 5599]/	# src 动作状态寄存器 src v1
9	* —— [5600, 5799]/	# src 动作状态寄存器 src v2(SRTOS)
10	* —— [9000, 9999] /	# 调度

寄存器地址	描述	类型	取值	单位	备注
40001	通过位姿定位 x			mm	必须一次性设置

		int32			<div>1. 若当前环境特征与地图特征匹配度过低将会定位失败</div> <div>2. 若是切换地图场景，特征点不是很明显的环境请使用“通过位姿强制定位”</div>
40003	通过位姿定位 y	int32			
40005	通过位姿定位 yaw	int32		(1/1000)rad	
40007	通过站点定位	uint16			若当前环境特征与地图特征匹配度过低将会定位失败
40009	自主导航移动到位姿 x 【废弃】	int32		mm	<div>必须一次性设置</div> <div>此接口没有设置任务编号所以默认ID为0</div>
40011	自主导航移动到位姿 y 【废弃】	int32			
40013	自主导航移动到位姿 yaw 【废弃】	int32		(1/1000)rad	
40015	自主导航移动到站点	uint16			此接口没有设置任务编号所以默认ID为0
40016	设置避障策略	enum/uint16	0x01: 暂停运动直至障碍消失 0x02: 重新规划路径绕过障碍 0x10: 不避障（可能导致事故，谨慎使用，自由导航有效）		设置避障策略只是暂时保存在modbus模块中，所有的移动任务都用这个避障策略。同理在Matrix上显示的避障策略也只是保存在Matrix上的，所以统一时刻两处的臂章策略可能不一样。
40017	执行动作任务ID【废弃】	uint16			<div>必须一次性设置</div> <div>（此接口没有设置任务编号所以默认ID为0）</div>
40018	执行动作任务参数0【废弃】	uint16			
40019	执行动作任务参数1【废弃】	uint16			

	十				
[40020, 40021]	可用				
40022	手动控制车辆以 Vx 线速度运行	int16	前进为正 后退为负	mm/s	必须一次性设置（40022-40024，或 40022-40025） 1. 需要在手动模式才生效； 2. 每次只执行100ms若要连续运行需要不间断发送 3. 若是无舵轮的车，可以只一次性设置前面三个寄存器
40023	手动控制车辆以 Vy 线速度运行（保留）	int16			
40024	手动控制车辆以 w 角速度运行	int16	向左为正 向右为负	(1/1000)rad	
40025	手动控制叉车舵轮转到指定角度	int16	向左为正 向右为负	0.01度	
40026	设置速度级别	uint16	[1, 100]		
40027	设置当前站点	uint16			
40028	设置扬声器音量	uint16	[1, 100]		
40029	设置当前地图	uint16			1. 取需要设置地图名字字符串的前两个字节 2. 若存在多张地图字符串的前两个字节相同，则按照字符排序，取第一张地图。 3. 建议用此功能时将地图名都命名为纯 ASCII 字符形式 4. 举例如下： 5. 需要将当前地图设置为：“12楼地图”，此处设置的值为：0x3132. 6. 若存在两张地图名分别为：“55a” 和 “55 aa”，当设置 0x3535 时，设置的当前地图为：“55a”。
40030	设置 GPIO output 的值	uint16			必须一次性设置 现在硬件只支持低8位；
40031					

	设置 GPIO output 的掩码	uint 16	硬件暂时不 支持掩码， 所以掩码被 强制设置为 0xFFFF		
40032	设置调度模式	enu m/ui nt16	0x01：手动 模式 0x02：自动 模式 0x03：维保 模式		
[40033, 40040]	mission中的通用 寄存器[0,7]	uint 16			
[40041, 40048]	可用				
40049	通过位姿强制定 位 x	int3 2		mm	必须一次性设置 强制定位不会匹配周围环境特征，直接定 位到目标位置，定位必定成功。 常用的使用场景为切换地图时强制定位， 比如：楼层切换的电梯口前，此处的特征 点不一定能匹配上，所以需要强制定位。 请区别于 “通过位姿定位” 和 “通过站点 定位”
40051	通过位姿强制定 位 y	int3 2			
40053	通过位姿强制定 位 yaw	int3 2		(1/1000)rad	
[40055, 40056]	可用				
40057	自主导航移动到 位姿任务编 号	int3 2			必须一次性设置 和40009的效果是一样的，但是这个接口 可以设置任务编号
40059	自主导航移动到 位姿 x	int3 2		mm	
40061					

	自主导航移动到 位姿 y	int3 2			
40063	自主导航移动到 位姿 yaw	int3 2		(1/1000)rad	
40065	可用				
40066	自主导航移动到 站点任务编 号	int3 2			必须一次性设置 和40015的效果是一样的，但是这个接口 可以设置任务编号
40068	自主导航移动到 站点	uint 16			
40069	可用				
40070	执行动作任务编 号	int3 2			必须一次性设置 和40017的效果是一样的，但是这个接口 可以设置任务编号
40072	执行动作任务 ID	int3 2			
40074	执行动作任务参 数 0	int3 2			
40076	执行动作任务参 数 1	int3 2			
[40078, 40096]	可用				
40097	执行 mission 任 务	uint 32	Matrix 上能 看到 mission 对应 的 ID		写一次此寄存器，将会将一个任务加入 mission 队列，推荐 modbus 用户 mission 队列不要超过 1，若超过 1 个 时， modbus 拿不到 mission 的结果，因 为上一个结果会被覆盖掉。
40099	可用				
[40100, 40105]	1353的6个模拟 量	uint 16	一个模拟量 占16bit		
40106	叉臂手动升降控 制	int1 6	上升为正 下降为负	mm/s	
40107	叉臂手动横移控 制	int1 6	向左为正	mm/s	

			向右为负		
40108	叉臂手动前移控制	int16	前移为正 后移为负	mm/s	
40111	导航路径编号	int32	1-127	id	必须一次性设置 如果想发送多条路径，需使用该寄存器累积id下发；id最小值为1，最大127,超过127条路径无法执行。角度单位为千分之一弧度，如，90°为1570； 发送完成后，需要发送路径确认(40151-40153)完成路径下发
40113	导航路径类型	int32	1(直线);2(贝塞尔);4(原地旋转)	枚举	
40115	导航路径方向	int32	1(前进);2(后退)	枚举	
40117	路径起始位置x	int32		mm	
40119	路径起始位置y	int32		mm	
40121	路径终点位置x	int32		mm	
40123	路径终点位置y	int32		mm	
40125	旋转路径朝向	int32		0.001rad	
40127	贝塞尔控制点1x	int32		mm	
40129	贝塞尔控制点1y	int32		mm	
40131	贝塞尔控制点2x	int32		mm	
40133	贝塞尔控制点2y	int32		mm	
[40135, 40149]	保留				
40151	导航任务编号	int32			必须一次性设置 路径编号与其他导航任务功能相同，需要累加；
40153	导航路径条数				

		int3 2			下发总路径条数=导航路径条数=执行路径条数； 三者都为0时，清空当前路径。
40155	需要执行路径条数	int3 2			
[40157, 40299]	可用				
[40300, 40399]	扩展动作控制器专用				
[40400, 40499]	安全PLC/手控盒专用				
[40500, 40600]	自定义专用，用户可任意修改				1、必须一次性设置，连续100寄存器。不能高频访问，建议访问周期100ms及以上； 2、使用功能示例：如riot与plc数据透传调试
[40601, 40999)	可用				

5. 功能示例

modbus在机器人上的调试日志可通过matrix看到：

- 系统设置
- 配置参数
- 账号管理
- 连接管理
- 系统维护
- 调试
- 硬件状态
- 性能统计
- 系统日志**
- 告警管理
- 调试信息
- 帮助
- 关于我们

Matrix > 系统日志

时间	模块	等级	日志
2024-08-23 17:27:13	modbus	info	add path12
2024-08-23 17:28:36	modbus	info	write multi register, start_addr:40161 quantity:6, map
2024-08-23 18:30:44	modbus	info	Connection closed on socket 44
2024-08-23 19:08:31	modbus	info	ModbusModule: TCP start listen connection!
2024-08-23 19:08:31	modbus	info	ModbusModule module start succeed!
2024-08-23 19:08:31	schedule	info	ScheduleModule: run
2024-08-23 19:08:40	modbus	info	New connection from 127.0.0.1:32653 on socket 50
2024-08-23 19:09:17	modbus	info	New connection from 10.10.68.99:45344 on socket 46
2024-08-23 19:09:41	modbus	info	write multi register, start_addr:40151 quantity:6, map
2024-08-23 19:09:41	modbus	info	will clear all path!
2024-08-23 19:10:35	modbus	info	write multi register, start_addr:40111 quantity:24, map
2024-08-23 19:10:35	modbus	info	get path from query: PATH_ROTATE angle:3140
2024-08-23 19:10:35	modbus	error	exception id is 3
2024-08-23 19:11:40	modbus	info	write multi register, start_addr:40111 quantity:24, map
2024-08-23 19:11:40	modbus	info	get path from query: PATH_ROTATE angle:3140
2024-08-23 19:11:40	modbus	info	add path11
2024-08-23 19:13:00	modbus	info	write multi register, start_addr:40111 quantity:24, mapping_address:150
2024-08-23 19:13:00	modbus	info	get path from query: PATH_LINE s(252,-332) e(-252,-332) FORWARD s_f:3.14159 e_f:3.14159
2024-08-23 19:13:00	modbus	info	add path12
2024-08-23 19:13:43	modbus	info	write multi register, start_addr:40151 quantity:6, mapping_address:150
2024-08-23 19:15:36	modbus	info	Connection closed on socket 46

筛选 清空

- ☐ 全选
- ☐ command-handler
- ☐ protobuf
- ☒ modbus
- ☐ movement-task
- ☐ action-task
- ☐ exec-error
- ☐ sros
- ☐ task
- ☐ device

确认

注意：后续所有例子为 TCP 协议和 RTU 协议公共部分数据（即 PDU）。

如果是 TCP 协议，在 Modbus PDU 数据前需加上 7byte 数据头，格式如下：

【01 02 00 00 00 06 01】 06 9C 47 00 05

其中 【】 中的数据为 TCP 协议 7 byte 头数据

TCP 头	01 02	用户自定义，2byte
	00 00	固定字节，用户不可修改， 2byte
	00 06	有效数据长度，该字段之后的所有数据 字节数， 2byte
地址	01	固定，用户不可修改， 1byte
PDU	06 9C 47 00 05	Protocol Data Unit ，协议数据单元

如果是 RTU 协议，在 Modbus PDU 数据前需加上 1byte 服务端地址数据， Modbus PDU 数据后需加上 2 byte CRC 校验数据

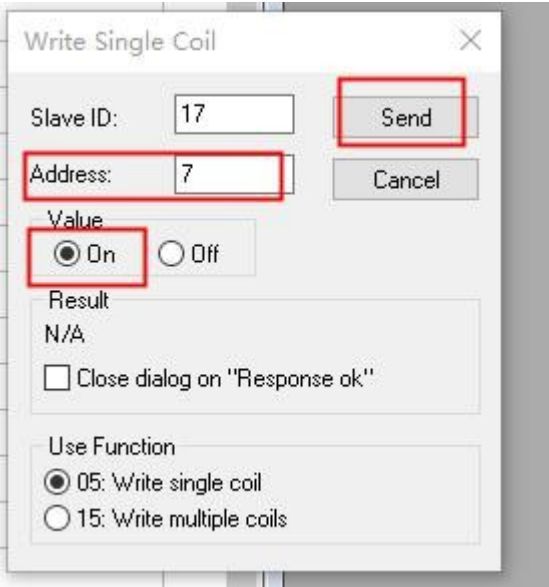
【0A】 06 9C 47 00 05 【F7 D6】

地址	11	服务端地址，固定为 0x11 ， 1byte
PDU	06 9C 47 00 05	Protocol Data Unit ， 协议数据单元
CRC 校验	F7 D6	CRC16 校验值， 2byte

5.1 写线圈状态

功能码： 0x05

PDU	功能码	05	写线圈
	寄存器地址	00 07	寄存器地址 00007
	数值	FF 00	ON: 0xFF00 OFF:0x0000



5.1.1 触发急停

- 1. Tx: 05 00 07 FF 00
- 2. Rx: 05 00 07 FF 00

5.1.2 解除急停

- 1. Tx: 05 00 08 FF 00
- 2. Rx: 05 00 08 FF 00

5.1.3 暂停运动

- 1. Tx: 05 00 01 FF 00
- 2. Rx: 05 00 01 FF 00

5.1.4 继续运动

- 1. Tx: 05 00 02 FF 00
- 2. Rx: 05 00 02 FF 00

5.1.5 停止运动

- 1. Tx: 05 00 03 FF 00
- 2. Rx: 05 00 03 FF 00

5.2 查询输入寄存器

功能码： 0x04

5.2.1 查询指定输入寄存器

以30001为起始地址，查询3个寄存器指令码：

- 1. Tx: 04 75 31 00 03
- 2. Rx: 04 06 00 02 00 03 00 00

Tx 发送请求：

PDU	功能码	04	读输入寄存器
	寄存器地址	75 31	寄存器起始地址 30001
	数值	00 03	读取 3 个输入寄存器，即 30001 ~30003

Rx 应答数据：

PDU	功能码	04	功能码
	字节数	06	读数据字节数
	数值	00 02 00 03 00 00	读数据

Read/Write Definition

Slave ID: 17

OK

Function: 04 Read Input Registers (3x)

Cancel

Address: 30001

Protocol address E.g. 30011 -> 10

Quantity: 3

Scan Rate: 1000 [ms]

Apply

Disable

☐ Read/Write Disabled

☐ Disable on error

Read/Write Once

View

Rows

☐ 10

☐ 20

☒ 50

☐ 100

☐ Fit to Quantity

☐ Hide Alias Columns

☐ PLC Addresses (Base 1)

☐ Address in Cell

☐ Enron/Daniel Mode

5.2.2 查询全部输入寄存器

以30001为起始地址，查询55个输入寄存器：

Tx: 04 75 31 00 37

PDU	功能码	04	读输入寄存器
	寄存器地址	75 31	寄存器起始地址 30001
	数值	00 37	读取 55 个输入寄存器， 即 3000 1~30055

Read/Write Definition

Slave ID: 17

Function: 04 Read Input Registers (3x)

Address: 30001

Quantity: 55

Scan Rate: 1000 [ms]

Disable

☐ Read/Write Disabled

☐ Disable on error

View

Rows

☐ 10☐ 20☒ 50☐ 100☐ Fit to Quantity

☐ Hide Alias Columns

☐ PLC Addresses (Base 1)

☐ Address in Cell

☐ Enron/Daniel Mode

OK

Cancel

Apply

Read/Write Once

Protocol address E.g. 30011 -> 10

5.3 查询离散输入状态

功能码 : 0x02

5.3.1 查询所有离散量输入状态

以10001为起始地址，查询所有离散量（50个）输入状态：

1. Tx: 02 27 11 00 32
2. Rx: 02 07 23 00 00 00 00 00

Tx 发送请求：

PDU	功能码	02	读输入寄存器
	寄存器地址	27 11	离散输入寄存器起始地址， 10001
	寄存器数量	00 32	读取 50 个寄存器，即 10001~1 0051

Rx 应答数据：

PDU	功能码	02	功能码
	字节数	07	读数据字节数

	读数据	23 00 00 00 00 00 00	读取到的数据
--	-----	----------------------	--------

Read/Write Definition

Slave ID: 17

Function: 02 Read Discrete Inputs (1x)

Address: 10001

Quantity: 50

Scan Rate: 1000 [ms]

Disable

☐ Read/Write Disabled

☐ Disable on error

View

Rows

☐ 10☐ 20☒ 50☐ 100☐ Fit to Quantity

☐ Hide Alias Columns

☐ PLC Addresses (Base 1)

☐ Address in Cell

☐ Enron/Daniel Mode

OK

Cancel

Apply

Read/Write Once

Protocol address. E.g. 10011 -> 10

5.4 写保持寄存器

功能码：写单个寄存器（0x06），写多个寄存器（0x10）

5.4.1 根据站点定位

以站点5的坐标作为初始位姿进行定位。

Tx: 06 9C 47 00 05

Rx: 06 9C 47 00 05

PDU	功能码	06	写单个寄存器
	寄存器地址	9C 47	地址为 40007
	数据	00 05	以站点 5 作为起始站点进行定位

Write Single Register

Slave ID:

17

Send

Address:

40007

Cancel

Value:

5

Result

N/A

☐ Close dialog on "Response ok"

Use Function

☒ 06: Write single register

☐ 16: Write multiple registers

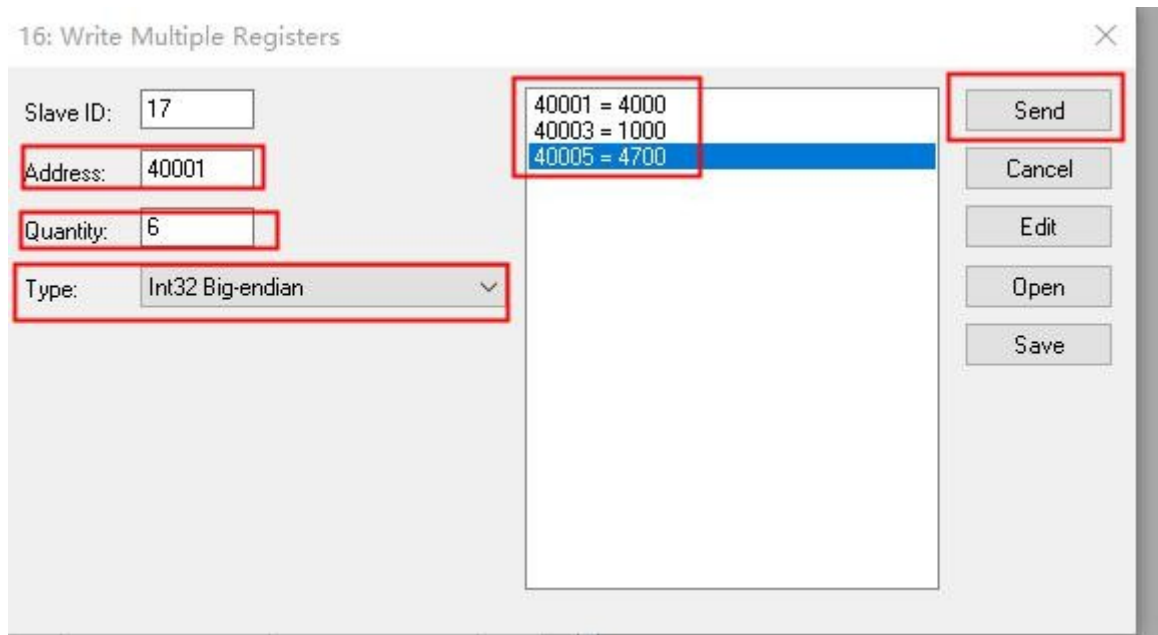
5.4.2 根据坐标定位

以坐标(x = 4000, y = 1000, yaw = 4700)作为初始位姿进行定位。

Tx: 10 9C 41 00 06 0C 00 00 0F A0 00 00 03 E8 00 00 12 5C

Rx: 10 9C 41 00 06 0C 00 00 0F A0 00 00 03 E8 00 00 12 5C

PDU	功能码	10	功能码，一次写多个寄存器
	寄存器起始地址	9C 41	地址 40001
	寄存器个数	00 06	寄存器个数为 6
	写数据字节数	0C	写数据字节数为 12
	数据	00 00 0F A0	位姿 x 坐标， 4000mm
		00 00 03 E8	位姿 y 坐标， 1000mm
		00 00 12 5C	位姿角度， 4700(1/1000 rad)



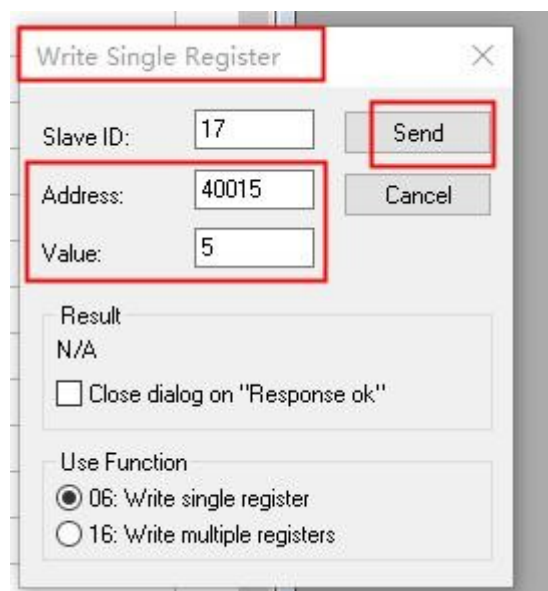
5.4.3 导航到站点

导航到站点5.

Tx: 06 9C 4F 00 05

Rx: 06 9C 4F 00 05

PDU	功能码	06	写单个寄存器
	寄存器地址	9C 4F	寄存器起始地址为 40015
	数据	00 05	导航到站点 5



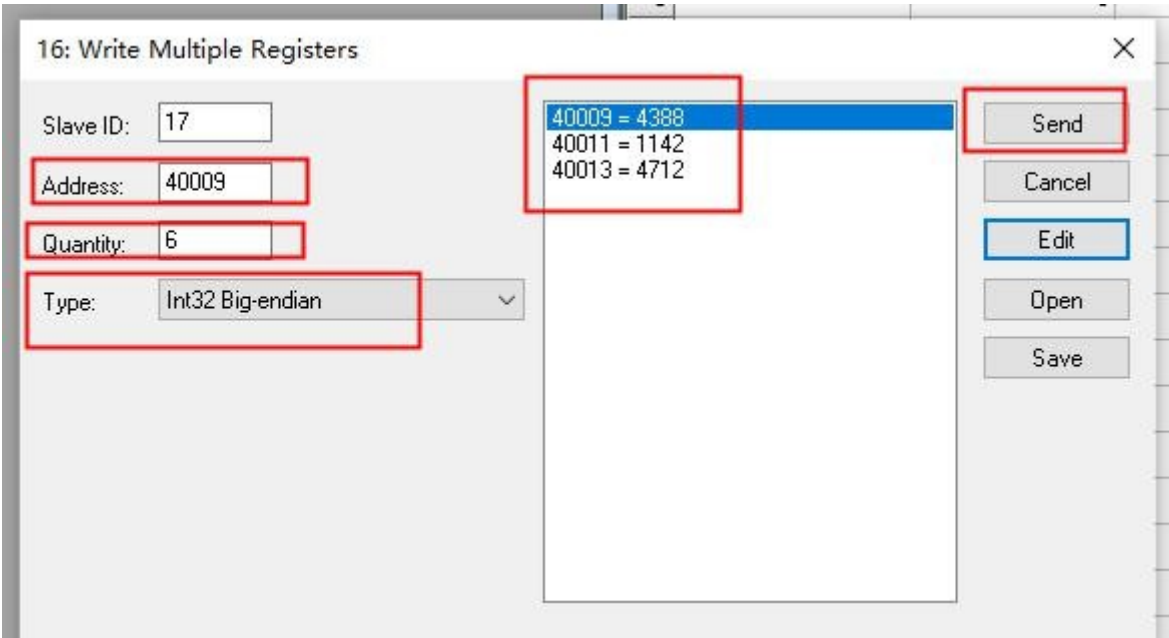
5.4.4 导航到坐标点

导航到坐标(x = 4388, y = 1142, yaw = 4712)

Tx: 10 9C 49 00 06 0C 00 00 11 24 00 00 04 76 00 00 12 68

Rx: 10 9C 49 00 06 0C 00 00 11 24 00 00 04 76 00 00 12 68

PDU	功能码	10	写多个寄存器
	寄存器地址	9C 49	寄存器起始地址为 40009
	寄存器个数	00 06	寄存器个数 6 个
	数据字节数	0C	数据字节数为 12
	数据	00 00 11 24	位姿 x 坐标，4388mm
		00 00 04 76	位姿 y 坐标，1142mm
		00 00 12 68	位姿角度，4712(1/1000 rad)



5.4.5 通过下发路径实现小车移动

我们以小车从（0m,0m,0°）位置，旋转90°，移动到（0m,4m,90°）,旋转-90°，移动到(4m,4m,0°)，在旋转到（4m,4m,180°）为例，以下为路径执行顺序：

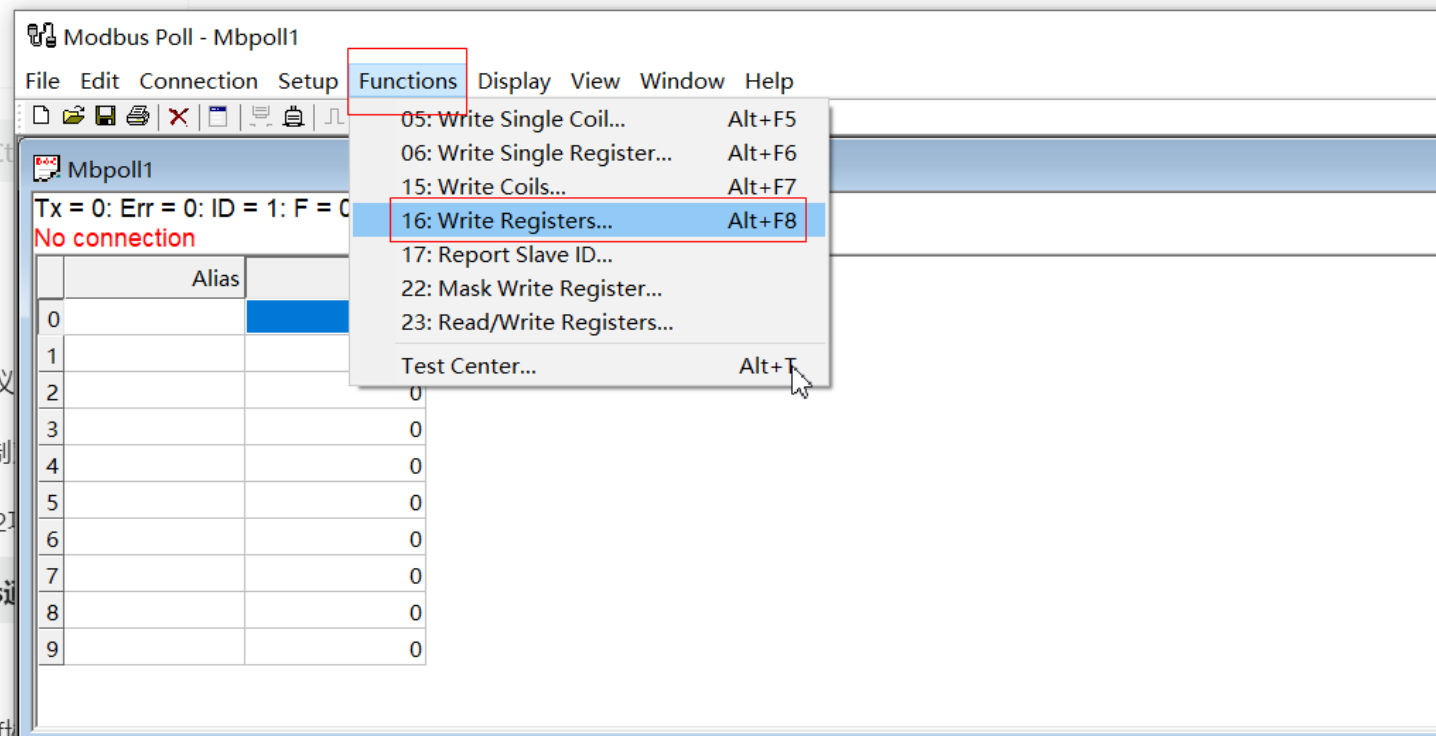
- 1. 旋转到90°
- 2. 直线运动4m
- 3. 旋转到0°
- 4. 直线运动4m
- 5. 旋转到180°

以下为路径信息：（注意表格很长）

编号	路径类型	运动方向	起始x	起始y	终点x	终点y	旋转朝向	控制点1x	控制点1y	控制点2x	控制点2y
1	4	1	0	0	0	0	1570	0	0	0	0
2	1	1	0	0	0	4000	0	0	0	0	0
3	4	1	0	4000	0	0	0	0	0	0	0
4	1	1	0	4000	4000	4000	0	0	0	0	0
5	4	1	4000	4000	0	0	3140	0	0	0	0

以下为发送步骤：

打开下发寄存器设置：



1. 清空当前缓存路径：

16: Write Multiple Registers

Slave ID: 17

Address: 40151

Quantity: 6

Type: Int32 Big-endian

40151 = 0

40153 = 0

40155 = 0

Send

Cancel

Edit

Open

Save

2. 下发路径1:

16: Write Multiple Registers

Slave ID: 17

Address: 40111

Quantity: 24

Type: Int32 Big-endian

40111 = 1

40113 = 4

40115 = 1

40117 = 0

40119 = 0

40121 = 0

40123 = 0

40125 = 1570

40127 = 0

40129 = 0

40131 = 0

40133 = 0

Send

Cancel

Edit

Open

Save

3. 下发路径2:

16: Write Multiple Registers

Slave ID: 17

Address: 40111

Quantity: 24

Type: Int32 Big-endian

40111 = 2

40113 = 1

40115 = 1

40117 = 0

40119 = 0

40121 = 0

40123 = 4000

40125 = 0

40127 = 0

40129 = 0

40131 = 0

40133 = 0

Send

Cancel

Edit

Open

Save

4. 下发路径3:

16: Write Multiple Registers

Slave ID: 17

Address: 40111

Quantity: 24

Type: Int32 Big-endian

40111 = 3

40113 = 4

40115 = 1

40117 = 0

40119 = 4000

40121 = 0

40123 = 0

40125 = 0

40127 = 0

40129 = 0

40131 = 0

40133 = 0

Send

Cancel

Edit

Open

Save

5. 下发路径4:

16: Write Multiple Registers

Slave ID: 17

Address: 40111

Quantity: 24

Type: Int32 Big-endian

40111 = 4

40113 = 1

40115 = 1

40117 = 0

40119 = 4000

40121 = 4000

40123 = 4000

40125 = 0

40127 = 0

40129 = 0

40131 = 0

40133 = 0

Send

Cancel

Edit

Open

Save

6. 下发路径5:

16: Write Multiple Registers

Slave ID: 17

Address: 40111

Quantity: 24

Type: Int32 Big-endian

40111 = 5

40113 = 4

40115 = 1

40117 = 4000

40119 = 4000

40121 = 0

40123 = 0

40125 = 3140

40127 = 0

40129 = 0

40131 = 0

40133 = 0

Send

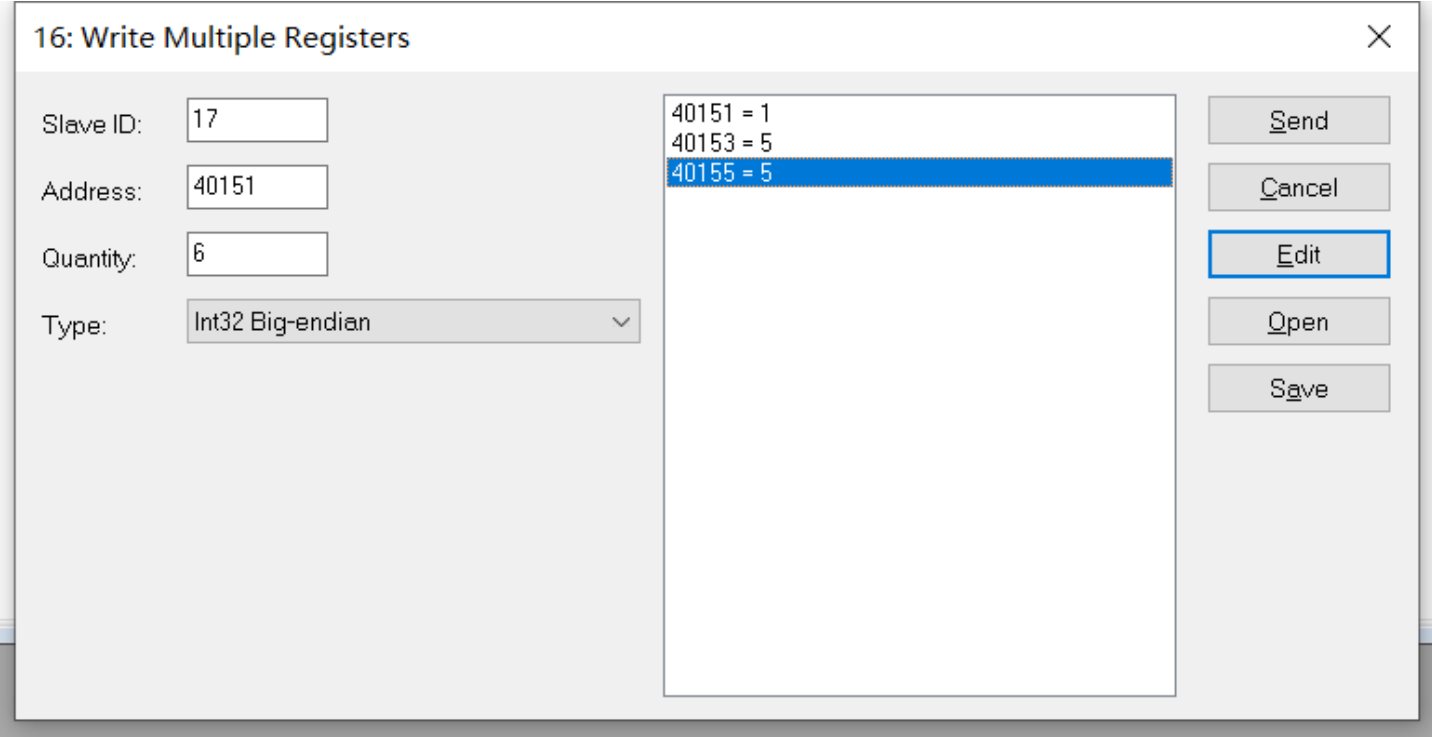
Cancel

Edit

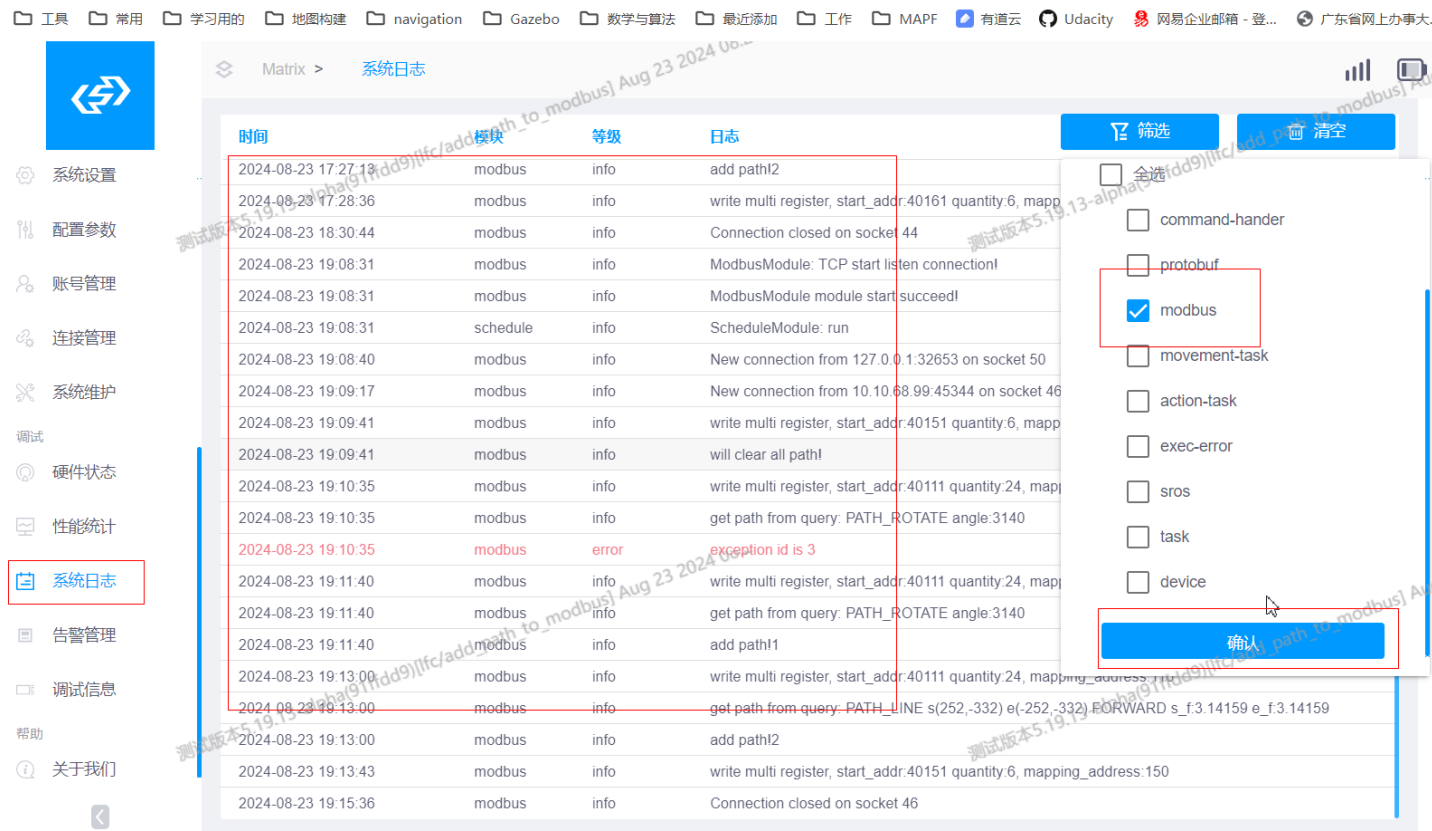
Open

Save

7. 下发路径确认指令：（路径确认指令只能发一次）



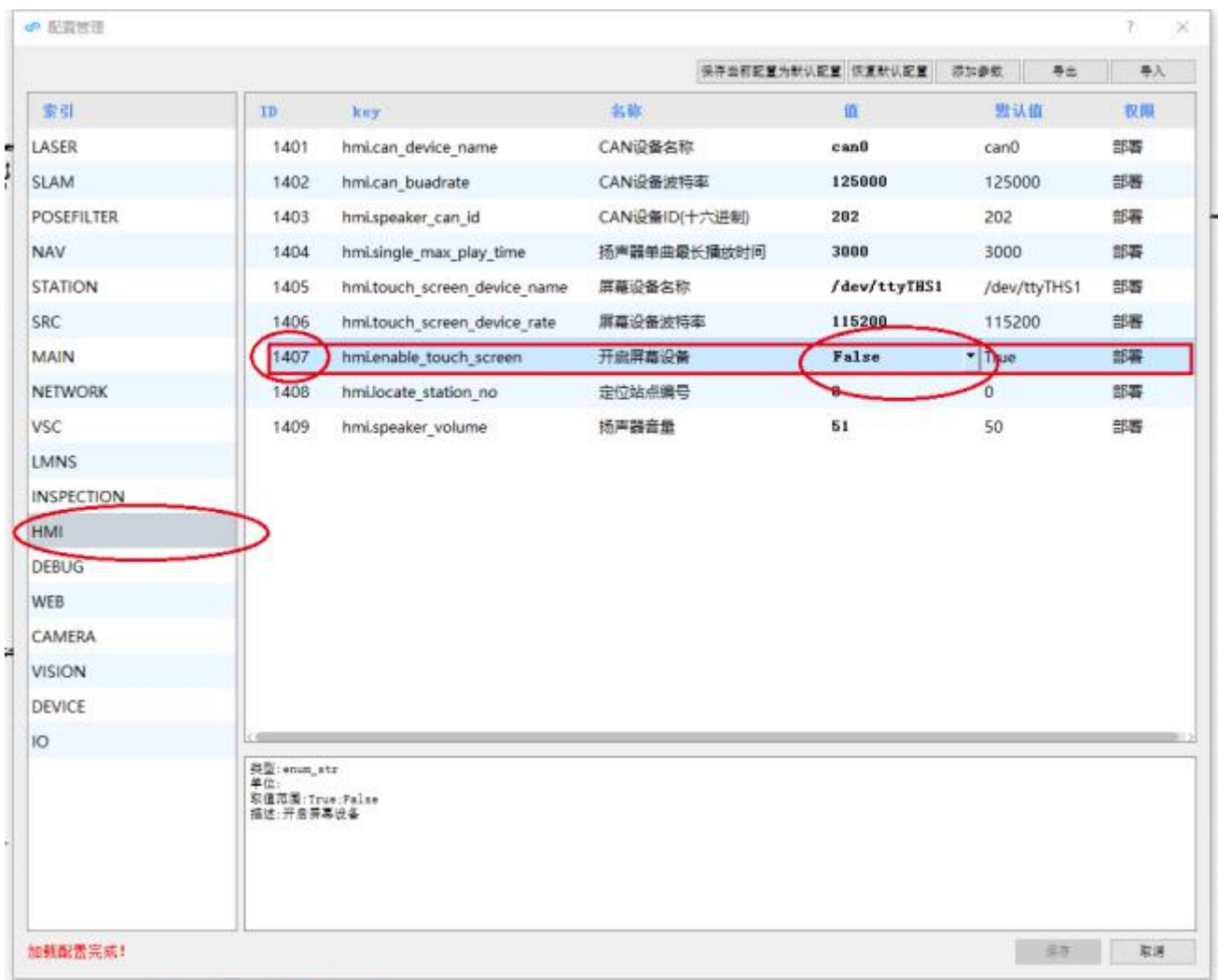
通过调试日志可查看路径下发情况：



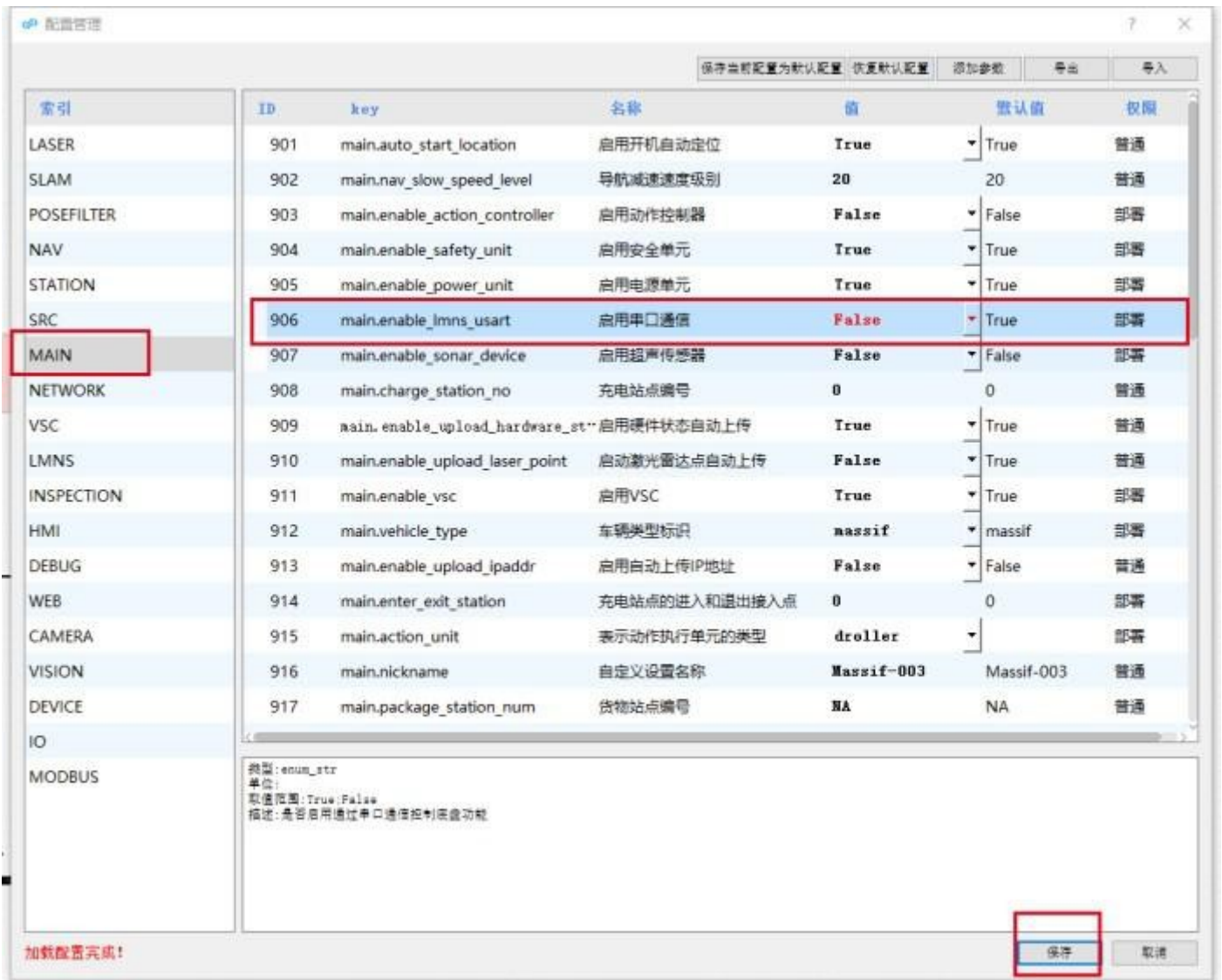
6. 常见问题

6.1 如何复用RS232-1端口

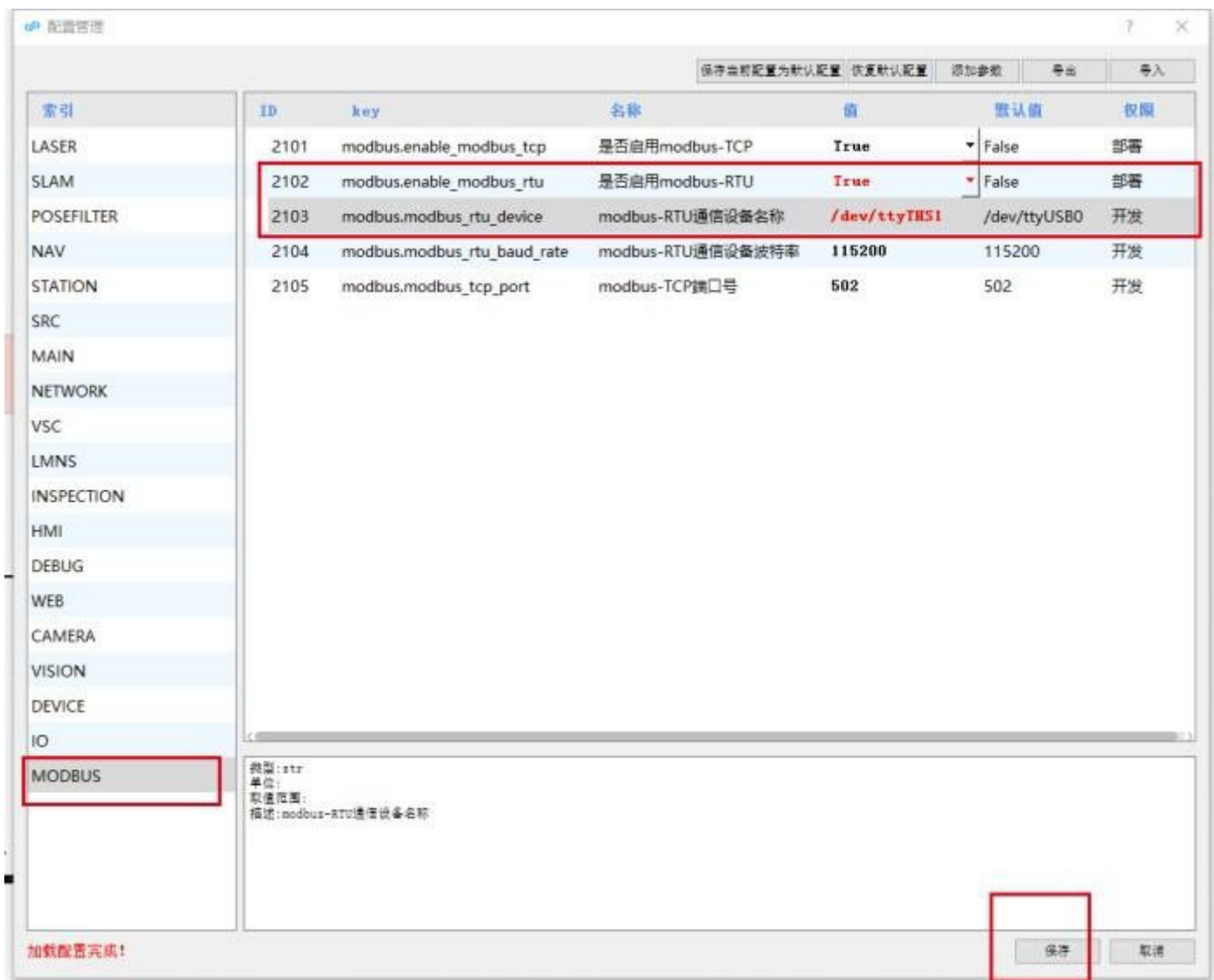
1. 关闭触摸屏功能：配置⇒管理⇒HMI⇒1407 将值设置为 False ，如下图所示：



2. 关闭串口通信: MAIN ⇒ 906 设置为 False , 如下图所示:



3. 将 MODBUS ⇒ 2102 设置为 True，将 2103 设置为 “/dev/ttyTHS1”。



4. 保存，重启 SROS：系统设置⇒ 重启系统

6.2 用modbus-RTU读取数据为啥一直timeout？，如何读取数据？

- 可能原因是没有设置 slave id ，用 modbus-RTU 时要记得将 slave id 设置为 17.

6.3 外部Modbus写多个寄存器无响应？

SROS只支持以下这样的基址和个数

寄存器地址	个数	描述	备注
40001	6	通过位姿定位	必须一次性设置
40009	6	移动到位姿	

40017	3	执行动作	
40022	3	手动控制模式下设置Vx线速度	
40030	2	设置GPIO输出值	
40049	6	通过位姿强制定位	
40057	8	HRA_MOVE_TO_POSE_WITH_NO	
40066	3	HRA_MOVE_TO_STATION_WITH_NO	
40070	8	HRA_SET_ACTION_WITH_NO	
40097	2	HRA_START_MISSION	
40330	1~10	Modbus与EAC数据透传通信	不能是其它基地址
40501	1~100	自定义功能通信，如riot与plc数据透传通信	不能是其它基地址

7. 内部使用的寄存器

与调度相关的寄存器参考这个文档：<https://standard-robots.yuque.com/group-rd/os4rbk/tesq2wsesfum4qlb#24PG>

7.1 线圈状态（可读可写）

下列寄存器使用功能码 05（Write Single Coil） / 15（Write Multiple Coils）进行写入，功能码15只支持一次写一个线圈，否则会报错。可以读取线圈状态，但读取的值为上次写入的值，并无实际意义。

寄存器地址	描述	取值
09200	载货状态复位按钮	写入 0xFF00 使能

7.2 保持寄存器（可读可写）

下列寄存器使用功能码 06（Write Single Register）、16（Write Multiple Registers）进行写入，读取的值为上次写入的值，并无实际意义。

寄存器	描述	类型	取值	备注地址
40900			0x00:无	废弃

	检测子步骤1状态	enum/uint16_t	0x01:检测中 0x02:检测成功 0x03:检测失败	
40901	检测子步骤2状态	enum/uint16_t	0x00:无 0x01:检测中 0x02:检测成功 0x03:检测失败	废弃
40902	检测子步骤3状态	enum/uint16_t	0x00:无 0x01:检测中 0x02:检测成功 0x03:检测失败	废弃
40903	检测子步骤4状态	enum/uint16_t	0x00:无 0x01:检测中 0x02:检测成功 0x03:检测失败	废弃
40904	当前设备检测状态	enum/uint16_t	0x00:无 0x01:检测中 0x02:检测成功 0x03:检测失败	废弃
40905	开始/结束设备检	enum/uint16_t	0x00:无	废弃

	测		0x01:开始 0x02:结束	
40906	正在检测的设备	enum/ui nt16_t	0x00:无 0x33:battery 0x34:bu100 0x35:主雷达 0x36:避障雷达 0x37:SR C 0x38:上视相机 0x39:下视相机 0x3a:避障相机 0x3b:IMU 0x3c:Light 0x3d:Speaker 0x3e:触摸屏 0x3f:行走电机 0x40:升降电机 0x41:旋转电机 0x42:前触边	废弃

			0x43:后触边 0x44:前急停 0x45:后急停 0x46:复位按钮 0x47:电源按钮 0x48:解抱闸按钮 0x49:充电口 0x4a:旋转原点开关 0x4b:升降上下限位开关 0x4c:WIFI 0x4d:cup100 0x4e:主动轮制动电阻	
40907	“继续”确认	enum/uint16_t	0x00:否 0x01:是	

7.3 输入寄存器（可读）

寄存器	描述	类型	取值	备注地址
30200	无线网卡ap状态	enum/uint32_t	0x00: 热点AP关闭 0x01: 热点AP开启	

30202	网卡 MAC.1	uint16_t	MAC地址 1	16进制
30203	网卡 MAC.2	uint16_t	MAC地址 2	16进制
30204	网卡 MAC.3	uint16_t	MAC地址 3	16进制
30205	网卡 MAC.4	uint16_t	MAC地址 4	16进制
30206	网卡 MAC.5	uint16_t	MAC地址 5	16进制
30207	网卡 MAC.6	uint16_t	MAC地址 6	16进制
30350	左叉尖光 电	enum/ui nt16_t	0x00：无 0x01：正 常 0x02：异 常	注：叉车 使用
30351	右叉尖光 电	enum/ui nt16_t	0x00：无 0x01：正 常 0x02：异 常	注：叉车 使用
30352	左叉尖触 发	enum/ui nt16_t	0x00：无 0x01：正 常 0x02：异 常	注：叉车 使用
30353	右叉尖触 发	enum/ui nt16_t	0x00：无 0x01：正 常	注：叉车 使用

			0x02: 异常	
30354	左到位检测	enum/uint16_t	0x00: 无 0x01: 正常 0x02: 异常	注: 叉车使用
30355	右到位检测	enum/uint16_t	0x00: 无 0x01: 正常 0x02: 异常	注: 叉车使用
30356	左货物下压开关检测	enum/uint16_t	0x00: 无 0x01: 正常 0x02: 异常	https://standard-robots.yuque.com/hw-ee/itiq58/cxqb2b#pEHnZ
30357	右货物下压开关检测	enum/uint16_t	0x00: 无 0x01: 正常 0x02: 异常	https://standard-robots.yuque.com/hw-ee/itiq58/cxqb2b#pEHnZ
30358	前移属具前限位	enum/uint16_t	0x00: 无 0x01: 正常 0x02: 异常	注: 叉车使用
30359	前移属具后限位	enum/uint16_t	0x00: 无 0x01: 正常 0x02: 异常	注: 叉车使用

30360	俯仰属具 上限位	enum/ui nt16_t	0x00：无 0x01：正 常 0x02：异 常	注：叉车 使用
30361	俯仰属具 下限位	enum/ui nt16_t	0x00：无 0x01：正 常 0x02：异 常	注：叉车 使用
30362	侧移属具 左限位	enum/ui nt16_t	0x00：无 0x01：正 常 0x02：异 常	注：叉车 使用
30363	侧移属具 右限位	enum/ui nt16_t	0x00：无 0x01：正 常 0x02：异 常	注：叉车 使用
30300+ [300, 400]	扩展动作 控制器专 用			
30300 + [400, 500]	安全PLC/ 手控盒专 用			
30301 + [500, 600]	自定义专 用，用户 可任意修 改			

说明

版权声明

2020 © 斯坦德机器人（深圳）有限公司

适用版本

本说明手册仅适用于 SROS_v4.12.0 及以上版本。

协议会随着软件版本升级发生变化，本说明手册所介绍内容可能会过时，如需最新版本的说明手册，请与项目经理或技术支持团队联系。

联系方式

技术支持团队

- 电子邮箱： support@standard-robots.com
- 联系电话： 400-800-7908
- 地址： 深圳市宝安区华丰国际机器人产业园2期D座1-4层