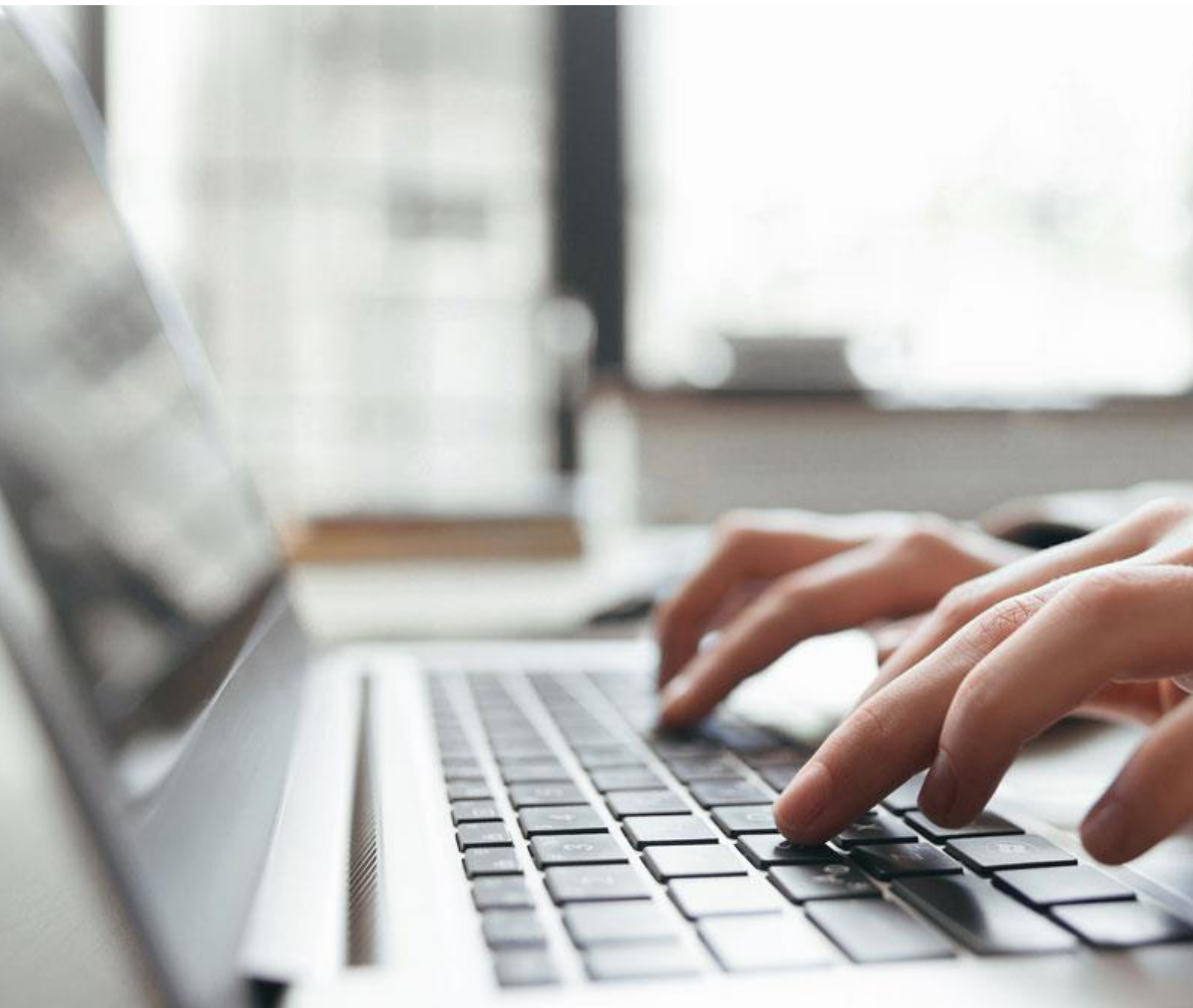




Application Note

How to start up a Yanlan controller / drive with
Modbus-rtu

Version 1.0.1

Contents

1	预期用途和受众	1
2	必要条件	1
3	配置电机	2
4	电机功能表	5
5	Modbus协议概要	6
6	调试演示	6
7	Liability	8
8	Imprint	9

1 预期用途和受众

本应用说明向您展示了如何在使用MODBUS-Rtu协议启动研蓝控制器/驱动器。为了安全起见，请遵循有效的OEM说明书。本文件中要求的所有产品仅供经过培训的用户使用。在使用产品之前，请确保所有用户完全阅读、理解并遵循本文档中的说明。

2 必要条件

NOTICE

不兼容导致故障！这里需要的产品有各种版本。提前选择/安装/配置正确的产品。

- ▶ 满足要求。
- ▶ 只能使用兼容的设备。
- ▶ 遵循有效的OEM说明。

在使用我们的示例项目之前，您必须确保电机和驱动器的正确操作。

1 提前配置驱动器 (=YANLANMC控制器), 基于我们的调试软件，电机可以正常被控制。

2 驱动器ID和波特率匹配正确

Hardware • 客户PLC / PC

- CPBA-MR / PD...MR

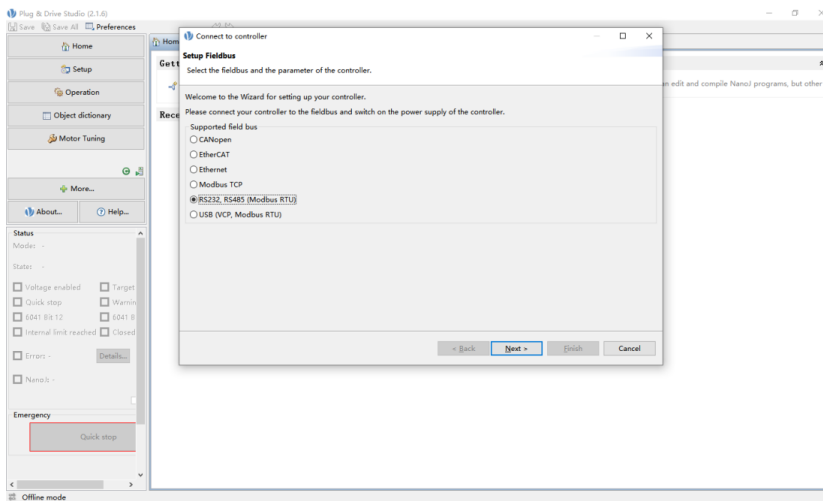
Software • Plug and Drive Studio 2.17 above / MThings_Pack-0.2.0

3 Plug and Drive Studio软件配置电机

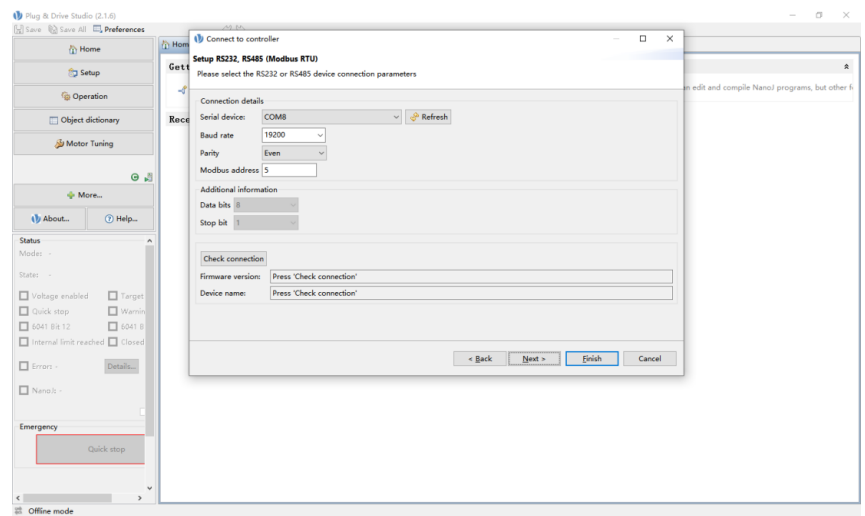
在Plug and Drive Studio软件可以自由调试和配置电机，在使用本范例项目之前，需要将电机和驱动器连接并配置好，同时配置好Modbus节点ID和波特率：

- 1 USB连接驱动器至电脑，连接软件
- 2 使用AutoSetup 自动识别电机（注意：仅限客户自有电机才需要适配），本说明略过
- 3 配置驱动器地址和波特率

485连接驱动器至电脑，连接软件串口



输入准确的地址和波特率信息，并连接



3.2 通讯设置

您可以在对象2028h、202Ah和202Dh中设置通信参数。

可以执行以下设置：

配置	寄存器地址	值范围	默认设置
Slave address	2028 _h	1 to 247	5
Baud rate	202A _h	7200 to 256000	19200
Parity	202D _h	■ None: 0x00 ■ Even: 0x04 ■ Odd: 0x06	0x04 (Even)

The number of data bits is always "8" here. The number of stop bits is dependent on the parity setting:

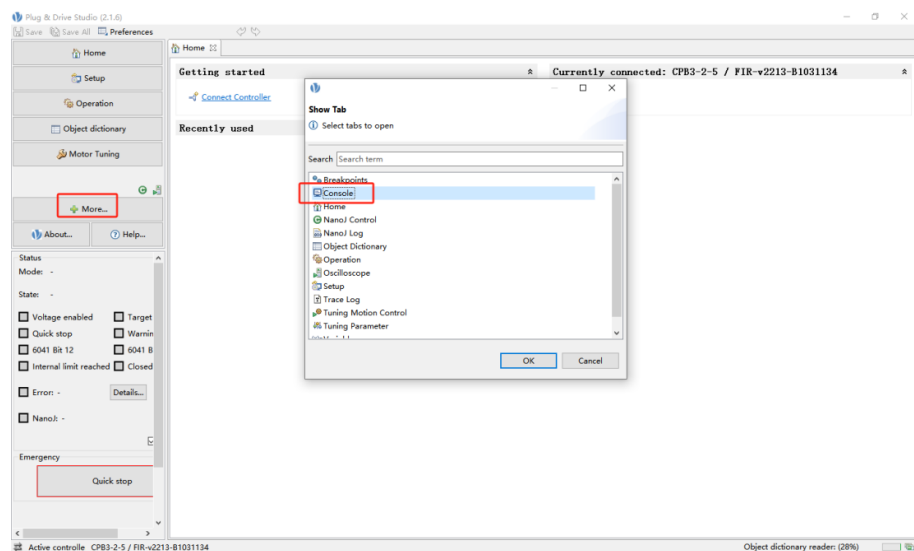
- No parity: 2 stop bits
- "Even" or "Odd" parity: 1 stop bit

The following baud rates are supported:

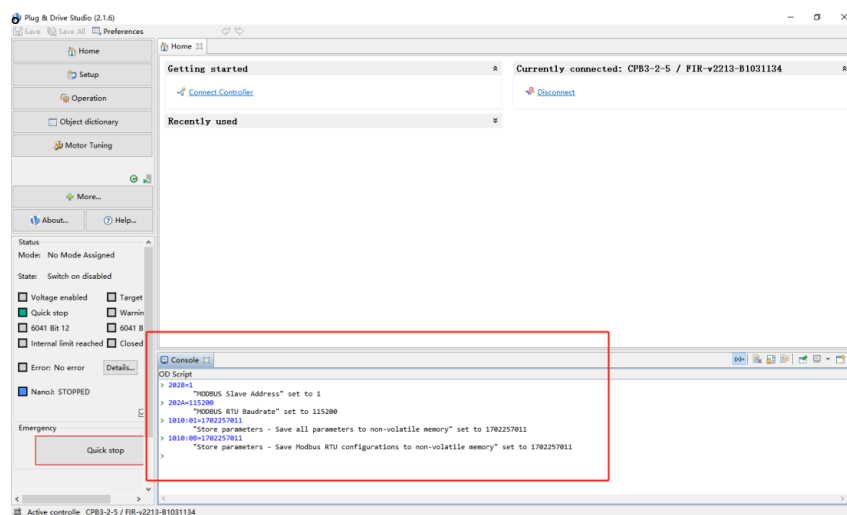
- 7200
- 9600
- 14400
- 19200
- 38400
- 56000
- 57600
- 115200
- 128000
- 256000

您必须通过在对象1010h:0Bh和：02h中写入值“65766173h”来保存更改。只有在控制器重新启动后，更改才会被接管。

连接后点击more选项，选择console窗口，点击ok



修改地址和波特率：在console窗口输入相应指令，2028设置地址，202A修改波特率，修改后输入两条保存指令（1010:01和1010:0B）



3.3 Setting the motor data

Prior to commissioning, the motor controller requires a number of values from the motor data sheet.

- Number of pole pairs: Object `2030h:00h` (pole pair count) The number of motor pole pairs is to be entered here. With a stepper motor, the number of pole pairs is calculated using the step angle, e.g., $1.8^\circ = 50$ pole pairs, $0.9^\circ = 100$ pole pairs (see step angle in motor data sheet). With BLDC motors, the number of pole pairs is specified directly in the motor data sheet.
- Object `2031h:00h`: maximum permissible motor current (motor protection) in mA (see motor data sheet)
- Object `6075h:00h`: rated current of the motor in mA (see motor data sheet), limited by `2031h`
- Object `6073h:00h`: maximum current (for a stepper motor, generally corresponds to the rated current, bipolar) in tenths of a percent of the set rated current (see motor data sheet). Factory settings: "1000", which corresponds to 100% of the value in `6075h`. Is limited by `2031h`.
- Object `203Bh:02h` Maximum duration of the maximum current (`6073h`) in ms (for initial commissioning, Nanotec recommends a value of 100 ms; this value is to be adapted later to the specific application).
- Setting the motor type:
 - Stepper motor:
 - Object `3202h:00h` (Motor Drive Submode Select): Defines motor type stepper motor, activates current reduction on motor standstill: 0000008h. See also chapter *Commissioning open-loop*.
 - BLDC motor:
 - Object `3202h:00h` (Motor Drive Submode Select): Defines motor type BLDC: 00000040h
- Motor with encoder without index: You must set the encoder parameters after the *Auto setup*, see chapter *Configuring the sensors*.
- □ Stepper motor, brake control (and current reduction) activated: 0000000Ch
- BLDC motor, brake control activated: 00000044h

NOTICE



Due to the sine commutation and the sinusoidal current flow, the current of a motor winding can achieve an alternating current value that is briefly greater (by max. $\sqrt{2}$ times) than the set current.

At especially slow speeds or while at a standstill with full load, one of the windings can therefore be supplied with overcurrent for a longer period of time. Take this into account when dimensioning the motor and select a motor with larger torque reserve if necessary if required by the application.

5.1.2 USB 更新配置

如果控制器通过USB电缆连接到PC，则控制器的行为类似于可移动存储设备。不需要更多的驱动程序。

显示三个文件：配置文件（`cfg.txt`）、配置程序（`vmmcode.usr`）和信息文件（`info.txt`），其中可以找到产品的序列号和固件版本。从而可以将配置文件或NanoJ程序存储在控制器上。在USB操作期间，还必须连接控制器的电源电压。

NOTICE

- 只能使用标准的USB电缆。切勿使用手机制造商在其产品中附带的USB电缆。这些USB电缆可能具有不同的插头形状或引脚分配。
- 不要在控制器上保存以下文件以外的任何文件：



1. `cfg.txt`
2. `vmmcode.usr`
3. `info.bin`
4. `reset.txt`

当控制器的电压电源打开时，任何其他文件都会被删除！

4 电机功能表

通过Modbus地址可以控制和读取我们的电机，我们预设的功能表如下：

控制指令表（PLC-电机）

寄存器地址	数据类型	功能
6000	UNSIGNED16	0x01：绝对定位模式 0x02：相对定位模式 0x04：JOG点动模式 0x08：归零模式 0x10：力矩模式 0x20：急停/停止 0x41：相对定位 0x80：复位错误 0x100：轴释放 0x200：轴使能
6001+6002	INTEGER32	JOG转速设定，单位转/分
6003+6004	INTEGER32	定位转速设定，单位转/分
6005+6006	INTEGER32	目标位置，默认3600/圈
6007+6008	UNSIGNED32	加速度设定，单位增加转速/秒
6009+6010	UNSIGNED32	减速度设定，单位减少转速/秒
6011+6012	UNSIGNED32	Begin Acceleration Jerk
6013+6014	UNSIGNED32	Begin Deceleration Jerk
6015+6016	UNSIGNED32	End Acceleration Jerk
6017+6018	UNSIGNED32	End Deceleration Jerk
6018+6019	UNSIGNED32	归零方式 19 或 21：外部传感器零点 -17 或 -18：堵转回零
6020+6021	UNSIGNED32	归零寻找感应器的速度
6022+6023	UNSIGNED32	调零速度
6024+6025	INTEGER32	堵转归零电流
6026+6027	INTEGER32	力矩模式，目标力矩
6028+6029	UNSIGNED32	力矩模式，力矩斜率

注意: 32位数据，低位在前，高位在后，如6028（低16位）+6029（高16位）

4 电机功能表（接续）

通过Modbus地址可以读取我们的电机，我们预设的功能表如下：

状态指令表（电机—PLC）

寄存器地址	数据类型	功能
5000+5001	INTEGER32	实时速度 单位 转/分
5002+5003	INTEGER32	实时位置 默认 3600脉冲/转
5004	UNSIGNED16	当前错误信息
5005	UNSIGNED16	历史错误信息
5006+5007	UNSIGNED32	历史错误信息（2个字节）
5008	UNSIGNED16	系统状态字
5009	UNSIGNED32	配置日期
5010+5011	UNSIGNED32	自定义状态

注意: 32位数据，低位在前，高位在后，如5010（低16位）+5011（高16位）

5.1 FC 16 (10_h) Write Multiple Registers

With this function code, one 16-bit value or multiple 16-bit values can be written. The function can be applied to NanoJ objects (see [Process data objects \(PDO\)](#)) or process data objects (see NanoJ objects).

Request ("N" is the number of registers to be written)		
Name	Length	Value
Slave address	1 byte	

Request ("N" is the number of registers to be written)		
Name	Length	Value
Function code	1 byte	10 _h
Start address	2 bytes	0000 _h to FFFF _h
Number of registers	2 bytes	0001 _h to 007B _h
Number of bytes	1 byte	2 * N
Register value	N * 2 bytes	
CRC	2 bytes	

Response		
Name	Length	Value
Slave address	1 byte	
Function code	1 byte	10 _h
Start address	2 bytes	0000 _h to FFFF _h
Number of registers	2 bytes	0001 _h to 007B _h
CRC	2 bytes	

Error		
Name	Length	Value
Slave address	1 byte	
Error code	1 byte	90 _h
Exception code (see Exception codes)	1 byte	01, 02, 03 or 04
CRC	2 bytes	

Example

Below is an example for writing values "0102_h" and "0304_h" starting with register address 6000 (1770_h), number of registers is 2, length of the data is 4:

Request

SA	FC	Data	CRC
05	10	17 70 00 02 04 01 02 03 04	AB 44

Response

SA	FC	Data	CRC
05	10	17 70 00 02	44 23

5.2 FC 3 (03_h) Read Input Registers / FC 4 (04_h) Read Holding Registers

With this function code, one 16-bit value or multiple 16-bit values can be read. This function can be applied to NanoJ objects (see [NanoJ objects](#)) or process data objects (min. 4-byte alignment, see [Process data](#) objects (PDO)).

Request		
Name	Length	Value
Slave address	1 byte	

Request		
Name	Length	Value
Function code	1 byte	03 _h / 04 _h
Start address	2 bytes	0000 _h to FFFF _h
Number of registers	2 bytes	1 to (7D _h)
CRC	2 bytes	

Response ("M" corresponds to the number of registers to be read)		
Name	Length	Value
Slave address	1 byte	
Function code	1 byte	03 _h / 04 _h
Number of bytes	1 byte	2 * M
Register value	2 bytes	
CRC	2 bytes	

Error		
Name	Length	Value
Slave address	1 byte	
Error code	1 byte	83 _h / 84 _h
Exception code (see Exception codes)	1 byte	01, 02, 03 or 04
CRC	2 bytes	

Example

Below is an example of a read request and response of register 5000 (1388_h) and of the following register (2 registers):

Request

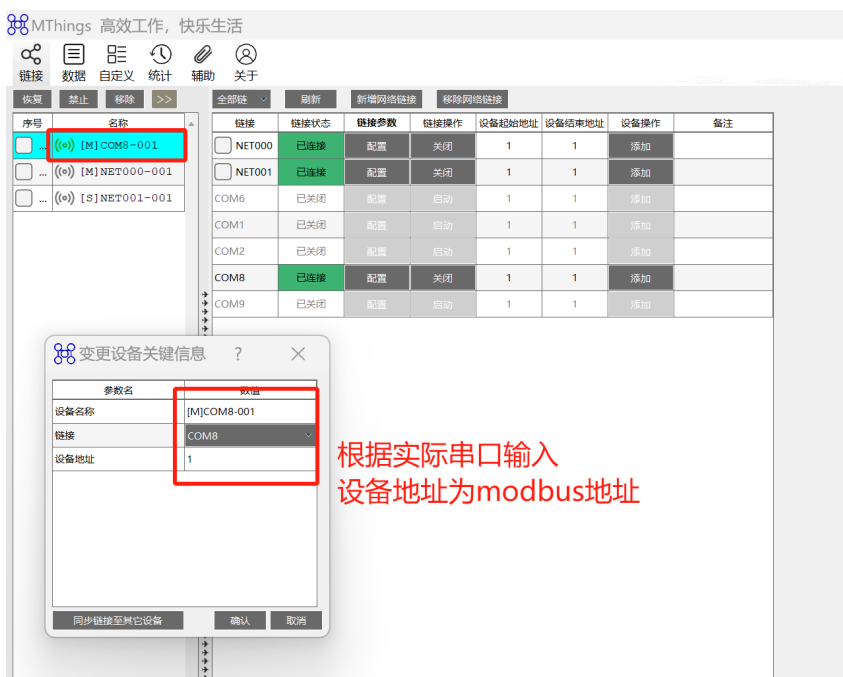
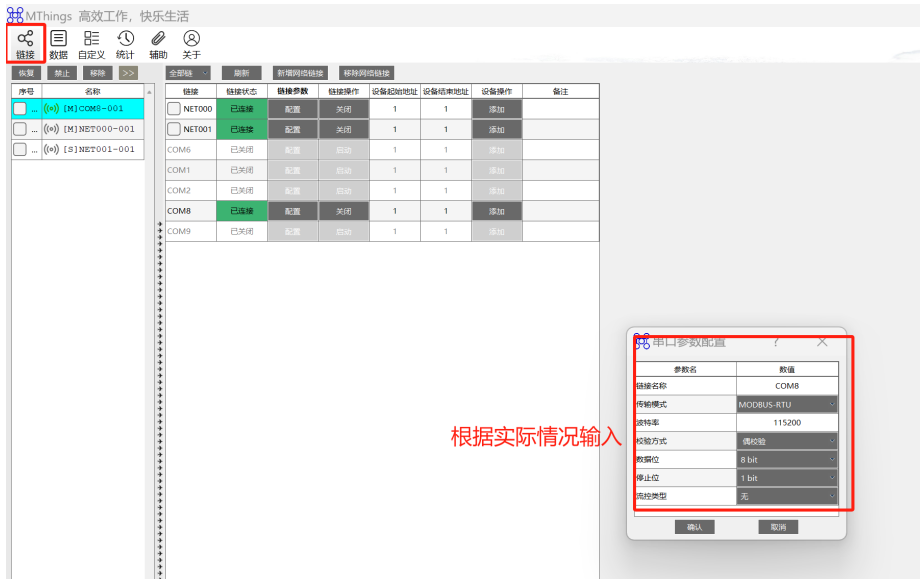
SA	FC	Data	CRC
05	03	13 88 00 02	41 21

Response

SA	FC	Data	CRC
05	03	04 02 40 00 00	41 21

6 使用第三方 MThings_Pack 测试电机

485连接驱动器至电脑，连接软件串口



配置和运行演示

MThings 高效工作, 快乐生活

链接 数据 自定义 统计 辅助 关于

恢复 禁止 移除 >>

输入查找名称

序号	名称	ID	名称	数值	单位	读	指令	写	区块	地址	数量	位偏移	位数	系数	范围	批读	批写	曲线	传输类型	呈现类型	字节序	字序	间隔时间(ms)	超时时间(ms)	重复次数
1	模式启动	0X0004	--	读	0x04	写	保持寄存器(RW)	6000	1	0	16	1	--	--	--	☒	☐	☐	UINT	INT(DEC)	大端	大端	50	2000	0
2	JOG速度	100	--	读	--	写	保持寄存器(RW)	6001	2	0	32	1	--	--	--	☒	☐	☐	UINT	INT(DEC)	大端	大端	50	2000	0
3	目标速度	500	--	读	--	写	保持寄存器(RW)	6003	2	0	32	1	--	--	--	☒	☐	☐	UINT	INT(DEC)	大端	大端	50	2000	0
4	目标定位	10000	--	读	--	写	保持寄存器(RW)	6005	2	0	32	1	--	--	--	☒	☐	☐	UINT	INT(DEC)	大端	大端	50	2000	0
5	加速度	5000	--	读	--	写	保持寄存器(RW)	6007	2	0	32	1	--	--	--	☒	☐	☐	UINT	INT(DEC)	大端	大端	50	2000	0
6	减速度	5000	--	读	--	写	保持寄存器(RW)	6009	2	0	32	1	--	--	--	☒	☐	☐	UINT	INT(DEC)	大端	大端	50	2000	0
7	开始加速度Begin ...	5000	--	读	--	写	保持寄存器(RW)	6011	2	0	32	1	--	--	--	☒	☐	☐	UINT	INT(DEC)	大端	大端	50	2000	0
8	开始减速度 Begin ...	5000	--	读	--	写	保持寄存器(RW)	6013	2	0	32	1	--	--	--	☒	☐	☐	UINT	INT(DEC)	大端	大端	50	2000	0
9	结束加速度End ...	5000	--	读	--	写	保持寄存器(RW)	6015	2	0	32	1	--	--	--	☒	☐	☐	UINT	INT(DEC)	大端	大端	50	2000	0
10	结束减速度End ...	5000	--	读	--	写	保持寄存器(RW)	6017	2	0	32	1	--	--	--	☒	☐	☐	UINT	INT(DEC)	大端	大端	50	2000	0
11	归零模式	35	--	读	--	写	保持寄存器(RW)	6019	2	0	32	1	--	--	--	☒	☐	☐	UINT	INT(DEC)	大端	大端	50	2000	0
12	找开速度	100	--	读	--	写	保持寄存器(RW)	6021	2	0	32	1	--	--	--	☒	☐	☐	UINT	INT(DEC)	大端	大端	50	2000	0
13	调零速度	100	--	读	--	写	保持寄存器(RW)	6023	2	0	32	1	--	--	--	☒	☐	☐	UINT	INT(DEC)	大端	大端	50	2000	0
14	堵转电流 (堵转找零模式)	500	--	读	--	写	保持寄存器(RW)	6025	2	0	32	1	--	--	--	☒	☐	☐	UINT	INT(DEC)	大端	大端	50	2000	0
15	目标力矩	500	--	读	--	写	保持寄存器(RW)	6027	2	0	32	1	--	--	--	☒	☐	☐	UINT	INT(DEC)	大端	大端	50	2000	0
16	力矩斜率	5000	--	读	--	写	保持寄存器(RW)	6029	2	0	32	1	--	--	--	☒	☐	☐	UINT	INT(DEC)	大端	大端	50	2000	0
17	实时速度	99	--	读	--	写	保持寄存器(RW)	5000	2	0	32	1	--	--	--	☒	☐	☐	UINT	INT(DEC)	大端	大端	50	2000	0
18	实时位置	160903	--	读	--	写	保持寄存器(RW)	5002	2	0	32	1	--	--	--	☒	☐	☐	UINT	INT(DEC)	大端	大端	50	2000	0
19	错误信息	0	--	读	--	写	保持寄存器(RW)	5004	1	0	16	1	--	--	--	☒	☐	☐	UINT	INT(DEC)	大端	大端	50	2000	0
20	历史错误代码	0	--	读	--	写	保持寄存器(RW)	5005	1	0	16	1	--	--	--	☒	☐	☐	UINT	INT(DEC)	大端	大端	50	2000	0
21	历史错误代码	2251760646	--	读	--	写	保持寄存器(RW)	5007	2	0	32	1	--	--	--	☒	☐	☐	UINT	INT(DEC)	大端	大端	50	2000	0
22	状态字	285278208	--	读	--	写	保持寄存器(RW)	5009	2	0	32	1	--	--	--	☒	☐	☐	UINT	INT(DEC)	大端	大端	50	2000	0

MThings 高效工作, 快乐生活

链接

数据

自定义

统计

辅助

关于

恢复

禁止

移除

>>

输入查找名称

配置

筛选列

筛选行

筛选区块

解析报文

曲线

批量读

批量写

序号	名称
☐ ...	((o) [M] COM8-001
☐ ...	((o) [M] NET000-001
☐ ...	((o) [S] NET001-001

ID	名称	数值	单位	读	指令	写	区块	地址	数量	位偏移	位数
1	模式启动	0X0004	--	读	0x04	写	保持寄存器(RW)	6000	1	0	16
2	JOG速度	100	--	读	--	写	保持寄存器(RW)	6001	2	0	32
3	目标速度	500	--	读	--	写	保持寄存器(RW)	6003	2	0	32
4	目标定位	10000	--	读	--	写	保持寄存器(RW)	6005	2	0	32
5	加速度	5000	--	读	--	写	保持寄存器(RW)	6007	2	0	32
6	减速度	5000	--	读	--	写	保持寄存器(RW)	6009	2	0	32
7	开始加速度Begin ...	5000	--	读	--	写	保持寄存器(RW)	6011	2	0	32
8	开始减速度 Begin ...	5000	--	读	--	写	保持寄存器(RW)	6013	2	0	32
9	结束加加速End ...	5000	--	读	--	写	保持寄存器(RW)	6015	2	0	32
10	结束减加速End ...	5000	--	读	--	写	保持寄存器(RW)	6017	2	0	32
11	归零模式	35	--	读	--	写	保持寄存器(RW)	6019	2	0	32
12	找开速度	100	--	读	--	写	保持寄存器(RW)	6021	2	0	32
13	调零速度	100	--	读	--	写	保持寄存器(RW)	6023	2	0	32
14	堵转电流 (堵转找零模式)	500	--	读	--	写	保持寄存器(RW)	6025	2	0	32
15	目标力矩	500	--	读	--	写	保持寄存器(RW)	6027	2	0	32
16	力矩斜率	5000	--	读	--	写	保持寄存器(RW)	6029	2	0	32
17	实时速度	99	--	读	--	写	保持寄存器(RW)	5000	2	0	32
18	实时位置	339238	--	读	--	写	保持寄存器(RW)	5002	2	0	32
19	错误信息	0	--	读	--	写	保持寄存器(RW)	5004	1	0	16
20	历史错误代码	0	--	读	--	写	保持寄存器(RW)	5005	1	0	16
21	历史错误代码	2251760646	--	读	--	写	保持寄存器(RW)	5007	2	0	32
22	状态字	285278208	--	读	--	写	保持寄存器(RW)	5009	2	0	32

速度模式

7 责任

本应用说明基于我们在广泛的工业应用中对典型用户需求的经验。本申请说明中的信息不保证其正确性和完整性，YANLANMC可随时更改，恕不另行通知。

它是一种通用指南，不应被解释为YANLANMC承诺保证其适用于所有客户应用，而无需在特定条件下进行额外测试，并在必要时由客户进行修改。

所提供的信息不会取代数据表和其他产品文档。对于最新版本-数据表和技术文档，请访问我们网站 www.yanlanmc.com 或联系我们客户经理获取。

应用说明在特定客户应用中的适用性和使用责任完全由客户授权。客户有责任评估、调查和决定申请说明是否有效并适用于各自的客户申请。

由于设备和模块处理不当而导致的缺陷不在保修范围内。在任何情况下，YANLANMC都不会对与所提供信息有关的任何直接、间接、附带或后果性损坏负责。

此外，我们的销售和交付条款和条件中关于责任的规定应适用。

8 联络

© 2023 Shanghai Yanlan Automation Technology Co., Ltd, all rights reserved. AutoShop are trademarks of Inovance. Original version.

Shanghai Yanlan Automation Technology Co., Ltd | No.1 Dongchen South Road|Shanghai
Tel. +21 61507699 | Fax +21 61507699 | info@yanlan.net | www.yanlanmc.com