



# 步进电机驱动器控制与通讯

数据手册 V3.00/2022.12.25

本手册适配产品：

低压总线型步进电机驱动器 AHD83/AHD84 系列

低压四轴步进电机驱动器 AQD84 系列

高压总线型步进电机驱动器 AHD8C 系列

低压闭环步进电机驱动器 ASD90 系列

低压伺服步进电机驱动器 ASD91/ASD92 系列



杭州时代自动化有限公司

<http://www.AgeMotion.com>

## 重要说明

---

请务必在使用前阅读 第1章【安全事项】，确保安全使用本公司产品。

请仔细阅读本手册，正确使用本公司产品。

阅读完本手册后，请妥善保管，以备随时查阅。

本手册电子版地址：

[http://www.AgeMotion.com/download/ds/CtrlAndCom\\_CHN.pdf](http://www.AgeMotion.com/download/ds/CtrlAndCom_CHN.pdf)

---

Copyright © 2001-2022 杭州时代自动化有限公司

转载或复制本手册的内容时，请保持内容的完整性

本数据手册内容如有更改，恕不另行通知，请从本公司网站获取最新版本

# 目 录

<b>1 安全事项 .....</b>	<b>1</b>	4.4.8 电流降流等待时间寄存器 CurrentLowWT.....25
<b>2 控制与通讯功能介绍 .....</b>	<b>3</b>	4.4.9 实时电流寄存器 Current.....25
2.1 USB 控制功能.....	3	4.4.10 电机实时位置寄存器 Position.....27
2.2 RS485 总线通讯功能.....	4	4.4.11 电机设定位置寄存器 PositionSet...28
2.2.1 通讯速率与通讯距离 .....	4	4.4.12 电机单齿分辨率寄存器 TResolution30
2.2.2 通讯节点 .....	4	4.4.13 脉冲步进长度寄存器 PulseLength..30
2.2.3 节点级联 .....	4	4.4.14 电机脉冲实时位置寄存器 PulsePosition .....32
2.2.4 总线终端 .....	5	4.4.15 电机脉冲设定位置寄存器 PulsePositionSet.....33
2.2.5 总线维护 .....	5	4.4.16 位差报错寄存器 PositionErrorAlarm34
2.2.6 总线电缆的制作 .....	6	4.4.17 到到位位差寄存器 PositionErrorAllowed .....35
<b>3 调试设置软件 AGEMOTION .....</b>	<b>7</b>	4.4.18 到位时间寄存器 TimeErrorAllowed36
3.1 AGEMOTION 电脑版使用说明 .....	7	4.4.19 位置误差寄存器 PositionError .....36
3.1.1 连接设置区 .....	8	4.4.20 脉冲位置误差寄存器 PulsePositionError .....37
3.1.2 参数设置区 .....	8	4.4.21 电机设定速度寄存器 VelSet .....37
3.1.3 运动控制区 .....	9	4.4.22 电机启动速度寄存器 VelStart.....38
3.1.4 传动设置区 .....	10	4.4.23 速度滤波器寄存器 VelFilter .....39
3.2.5 状态监控区 .....	10	4.4.24 电机速度系数寄存器 KV.....40
3.2.6 通讯 DLL .....	10	4.4.25 速度滤波器（通讯）寄存器 VelFilterCom .....40
<b>4 驱动器通讯协议 .....</b>	<b>11</b>	4.4.26 电机实时速度寄存器 Vel.....40
4.1 寄存器列表.....	11	4.4.27 电机零限设定速度寄存器 VelSetZero .....41
4.2 协议基本格式 .....	13	4.4.28 零限速度滤波寄存器 VelFilterZero 41
4.2.1 通讯位模式.....	14	4.4.29 总线看门狗定时寄存器 BusWDT...42
4.2.2 Big-Endian/Little-Endian.....	14	4.4.30 总线地址寄存器 BusAddress .....43
4.2.3 通讯协议帧描述.....	14	4.4.31 总线通讯速率寄存器 BusBand .....43
4.2.4 CRC 校验.....	15	4.4.32 端口寄存器 Port.....44
4.2.5 MODBUS 异常响应 .....	16	4.4.33 端口上升标志寄存器 PortHiFlag...45
4.3 寄存器的操作 .....	16	4.4.34 端口下降标志寄存器 PortLoFlag ...46
4.3.1 操作模式 .....	16	4.4.35 端口翻转标志寄存器 PortFlipFlag .47
4.3.2 操作规则 .....	16	4.4.36 端口配置寄存器 PortConfig .....48
4.4 寄存器详解及报文示例 .....	17	4.4.37 输入带宽寄存器 InputBand .....50
4.4.1 控制寄存器 Control .....	17	4.4.38 刹车电压寄存器 VoltageBreak .....50
4.4.2 故障寄存器 ErrorCode .....	20	4.4.39 电机序列号 0 寄存器 MotorSN0 ....51
4.4.3 输入类型寄存器 InputType.....	21	
4.4.4 电流最大值寄存器 CurrentMax .....	22	
4.4.5 电流最小值寄存器 CurrentMin .....	23	
4.4.6 电流设定寄存器 CurrentSet .....	23	
4.4.7 电流降流寄存器 CurrentLow .....	24	

4.4.40 电机序列号 1 寄存器 MotorSN1 ....	52
4.4.41 电机序列号 2 寄存器 MotorSN2 ....	52
4.4.42 电机序列号 3 寄存器 MotorSN3 ....	52
4.4.43 电机型号寄存器 MotorName .....	53
4.4.44 电机生产日期寄存器 MotorDate....	53
4.4.45 电机编号寄存器 MotorNum.....	54
4.4.46 伺服功能寄存器 Servo .....	54
4.4.47 驱动器 CPU 温度寄存器 CpuTemp55	
4.4.48 驱动器散热器温度寄存器 SinkTemp55	
4.4.49 驱动器型号寄存器 DriverName .....	56
4.4.50 驱动器生产日期寄存器 DriverDate	56
4.4.51 驱动器编号寄存器 DriverNum.....	56
<b>5 应用举例.....</b>	<b>57</b>
5.1 电脑、单片机通讯程序 .....	57
5.1.1 通讯测试 .....	57
5.1.2 初始化 .....	57
5.1.3 向上下限位运行.....	58
5.1.4 向指定位置运行.....	58
5.2 PLC 程序示例 .....	59
5.2.1 通讯测试 .....	59
5.2.2 运行控制 .....	60
5.2.3 从驱动器读取数据 .....	61
<b>6 售后服务.....</b>	<b>63</b>
6.1 服务条款 .....	63
6.2 公司简介 .....	63
6.3 产品列表 .....	63
6.4 联系我们 .....	65

## 1

# 安全事项

为保障使用者的人身安全，保护设备的正常使用，请务必阅读并遵守本章的安全事项。



## 危险

在操作时违反本事项所示要求，可能会导致人员重伤或者死亡。



## 警告

在操作时违反本事项所示要求，可能会引起驱动器永久损坏及附加事故。



## 危险

### 谨防触电，爆炸或其他危险

- 禁止在易爆、易燃或腐蚀性环境使用本产品；
- 禁止开启产品外壳；
- 低压驱动器带电时内部电压可能超过 **36VDC**，高压驱动器带电时内部电压可能超过 **400VDC**，驱动器和电机都必须接安全保护地线；
- 驱动器内部电压不会瞬间释放，必须先切断电源，等指示灯熄灭后才能进行插拔、接线、设置、测量、搬动等人工操作；
- 如果某接口没有明确指出允许带电插拔，禁止对该接口进行带电插拔操作；
- 驱动器故障时温度可能很高，必须先切断电源，等下降至安全温度后才能进行人工操作；
- 驱动器应用于直接涉及人身安全的设备，必须配备人身安全防范措施；
- 驱动器或设备故障时可能存在火灾隐患，必须配备消防安全防范措施。

若不遵守以上规定，可能会导致严重伤害或死亡。



## 应用与设置

- 驱动器应用必须符合性能参数的定义；
- 初次上电前应严格检查接线、参数设置；
- 初次应用应遵循先逐步提高力矩、再逐步提高速度的原则进行驱动器及电机调试；
- 禁止采用通断电源方式启动和停止电机。

## 连接

- 低压驱动器由直流电源供电，电源极性不得接反；
- 禁止螺钉端子插在驱动器上时直接接线，必须拔下螺钉端子完成接线；
- 禁止对螺钉端子的预接导线进行蘸锡处理。

## 运行

- 两次上电之间最少需要间隔 20 秒，否则可能引起驱动器损坏。

## 保养和检查

- 定期清理积灰，积灰严重时须增加清理次数；
- 驱动器每三个月须上电运行 1 小时，以保障内部电子元器件的正常。

## 拆机和修理

- 非本公司专业技术人员严禁开启产品外壳，禁止对本驱动器进行拆解或改造；
- 如需检查或维修，请退回原厂处理。

## 废弃

- 废弃驱动器时，请按工业废弃物处理，以免造成环境污染。

若不遵守以上规定，可能会引起驱动器永久损坏及附加事故。

## 2

## 控制与通讯功能介绍

本公司的驱动器产品中，支持 USB 或 RS485 总线的具体型号列表如下：

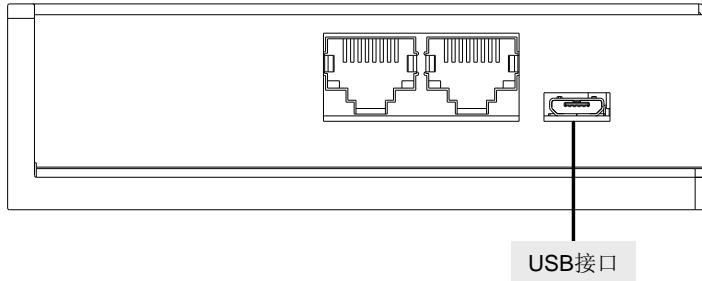
产品系列	串口	USB	RS485 总线
AHD80/AHD81/AHD82 系列	×	×	×
AHD83 系列	×	●	○
AHD84 系列	×	●	○
AQD84 系列	×	●	○
AHD86 系列	×	×	×
AHD8C 系列	×	×	●
ASD90 系列	×	●	○
ASD91/ASD92 系列	×	●	○

● 全系列支持

○ 部分型号支持

× 不支持

## 2.1 USB 控制功能



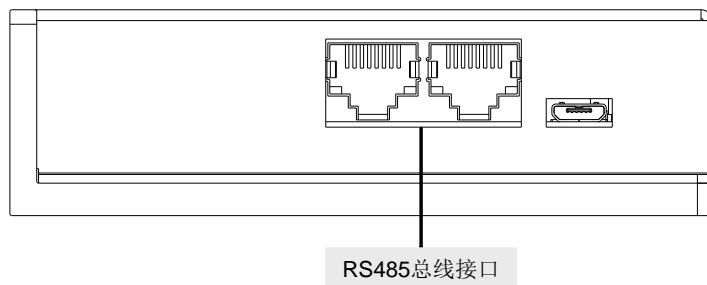
驱动器内置 USB 芯片，任何可以作为 USB 主机的设备，比如带 Windows 操作系统的电脑，可以按照 Modbus-RTU 协议，对驱动器进行参数设置，或者直接控制驱动器的运行。

本公司提供的电脑软件 AgeMotion，正是利用驱动器的 USB 控制功能，对驱动器进行参数设置和运行控制。

**提示** 驱动器端 USB 接口支持带电插拔。

关于 AgeMotion 的使用，请参考本手册第 3 章《AgeMotion 调试设置软件》。

## 2.2 RS485 总线通讯功能



驱动器内置工业级总线通讯芯片，任何带 RS485 通讯功能的工业设备，都可以按照 Modbus-RTU 协议，直接控制驱动器的运行。如果有需要，可以串联最多 247 台驱动器，在较低的成本下，实现可靠的中到大型驱动器网络的构建。

### 2.2.1 通讯速率与通讯距离

通讯速率、通讯距离是和具体应用现场相关的，速率和距离的典型关系如下：

通讯速率 (bps)	通讯距离 (m)
9600	1000
19200	1000
38400	1000
57600	800
115200	500
250000	250

提示 由于外界干扰及总线节点数量的差异，实际现场的通讯速率及通讯距离会有一定的差异。

### 2.2.2 通讯节点

本公司驱动器上的总线功能，最大支持 247 个总线节点应用（不包含总线主机），是否能支持最多的节点数量还取决于总线主机的总线驱动能力。

### 2.2.3 节点级联

驱动器的 RS485 总线接口采用 RJ45 接口，每台驱动器配置有两个 RJ45 接口，用于连接前后两级节点。这两个 RJ45 接口电路上是并联，接其中任何一个都是等效的。

提示 水晶头建议采用 TECO/AMP 的屏蔽水晶头，网线采用 AMP 的 CAT5e/24AWG 双屏蔽网线，屏蔽线不要与水晶头屏蔽壳短路。

提示 建议不要采用外部的 T 形连接拓扑。

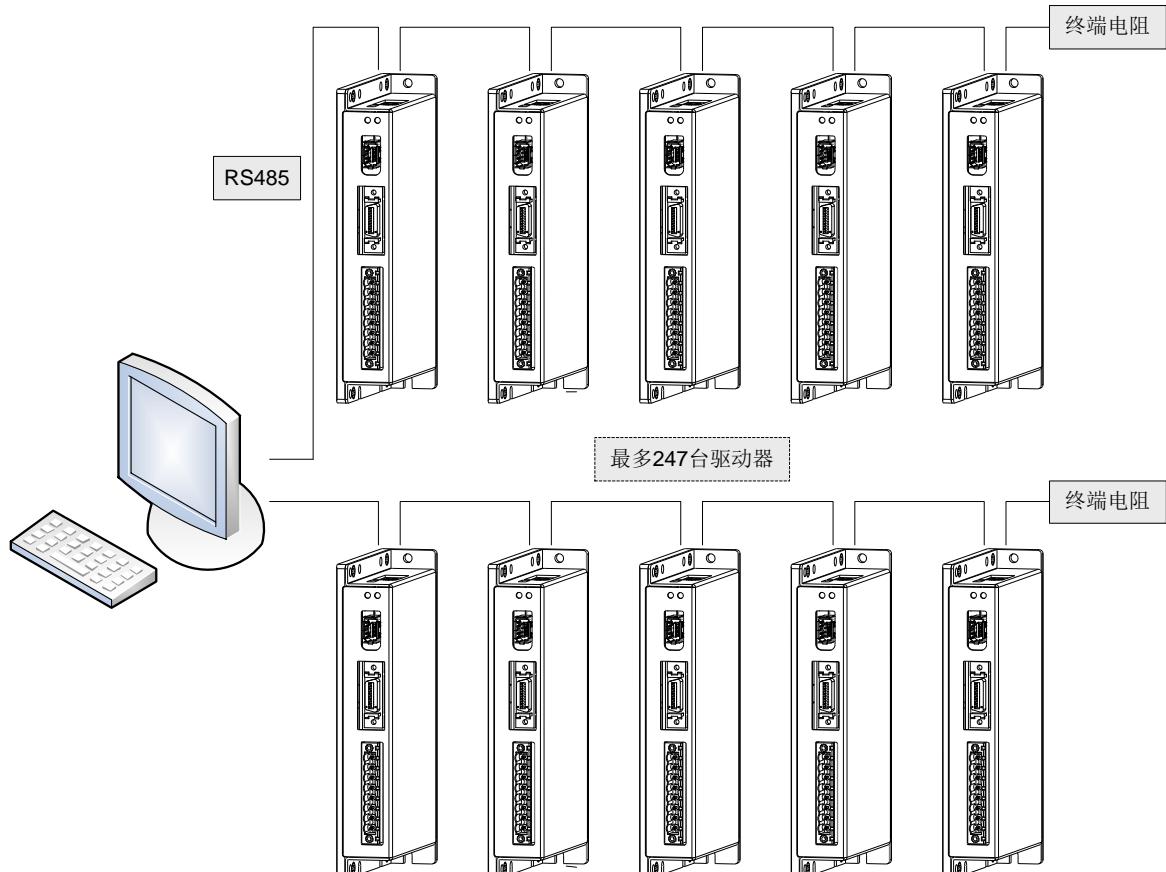
## 2.2.4 总线终端

RS485 总线链路的两个终端, 即第一个和最后一个节点的两条信号线之间, 都需要接  $120\Omega$  终端电阻, 以保证信号质量。

如果总线主机处于总线终端, 主机必须接  $120\Omega$  终端电阻。

终端数量必须是 2 个, 且只能在物理两端。接地只能有一端或没有。

**提示** 因为驱动器自身提供优化的总线终端方案, 所以建议让驱动器作为总线终端, 即总线的两个终端是驱动器, 主机处于总线中间节点的任何一个位置。



## 2.2.5 总线维护

如果有节点驱动器故障, 可拔下故障驱动器的两个 RJ45 水晶头, 再用 RJ45 双头转接器连接, 即可剔出故障节点, 待故障节点修复后再接入总线即可。

如果总线电缆故障, 用备用总线电缆替代即可。

**提示** 驱动器端 RS485 总线接口支持带电插拔。

**△危险** 带电插拔时注意操作方式, 不要接触到高压电线。

## 2.2.6 总线电缆的制作

为达到性能指标和统一接线规范，级联总线电缆采用国际标准线序 T568B（BB 标）制作。

总线电缆接口定义如下表所示：

RJ45 序号	1	2	3	4	5	6	7	8
T568B	白橙	橙	白绿	蓝	白蓝	绿	白棕	棕
信号定义	-	-	RS485+	-	-	RS485-	GND	GND

总线电缆的使用原则是整个级联电缆只能有一个接地端，因此不建议驱动器端电缆屏蔽层接地，悬空即可。主机端电缆可以考虑接地。

## 3

## 调试设置软件 AgeMotion

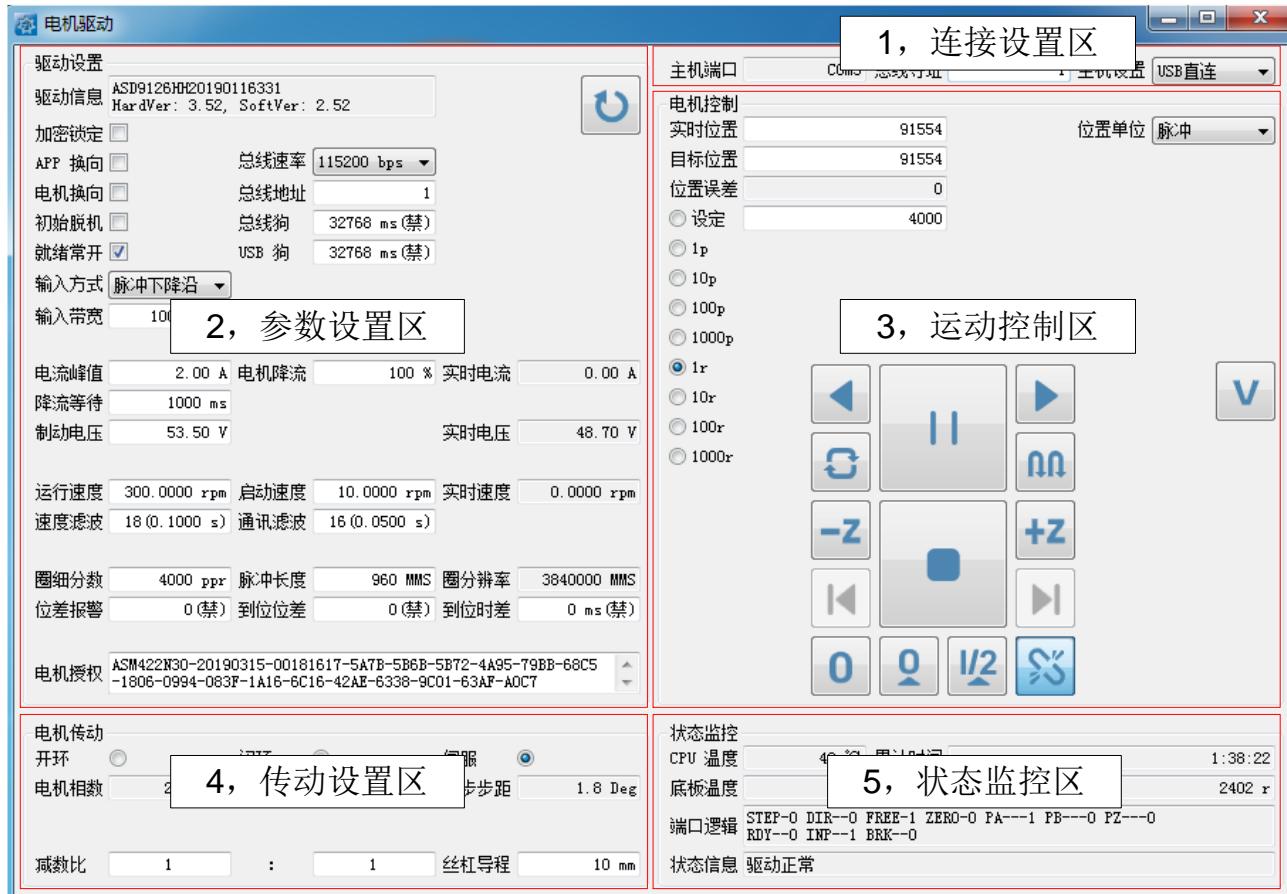
本公司免费提供驱动器调试设置软件 AgeMotion (Windows 版)。支持 USB 功能的驱动器，可通过 USB 数据线，与电脑连接。连接后，操作者可通过 AgeMotion (Windows 版)，对驱动器进行参数设定和运行测试。

### 3.1 AgeMotion 电脑版使用说明

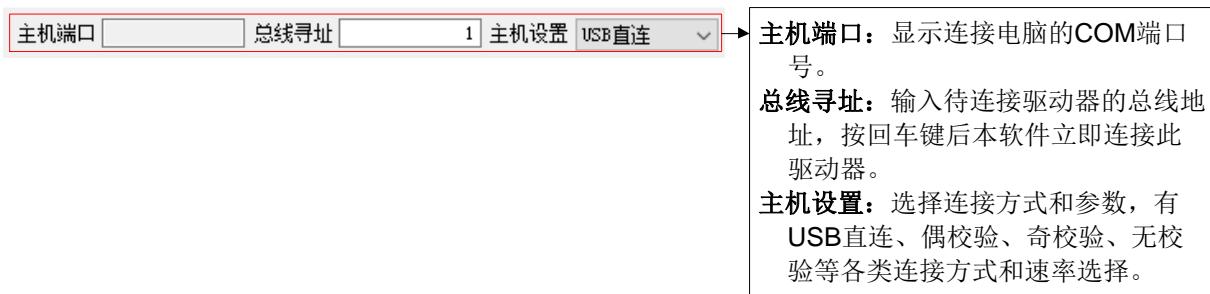
AgeMotion 电脑版支持 Windows XP、Windows7、Windows10 的 32 位、64 位版本。首先请在官网下载并安装最新版本，安装过程中软件会自行安装驱动程序，请不要退出，完成整个安装过程。

打开软件，界面如下图，分为五大区域：

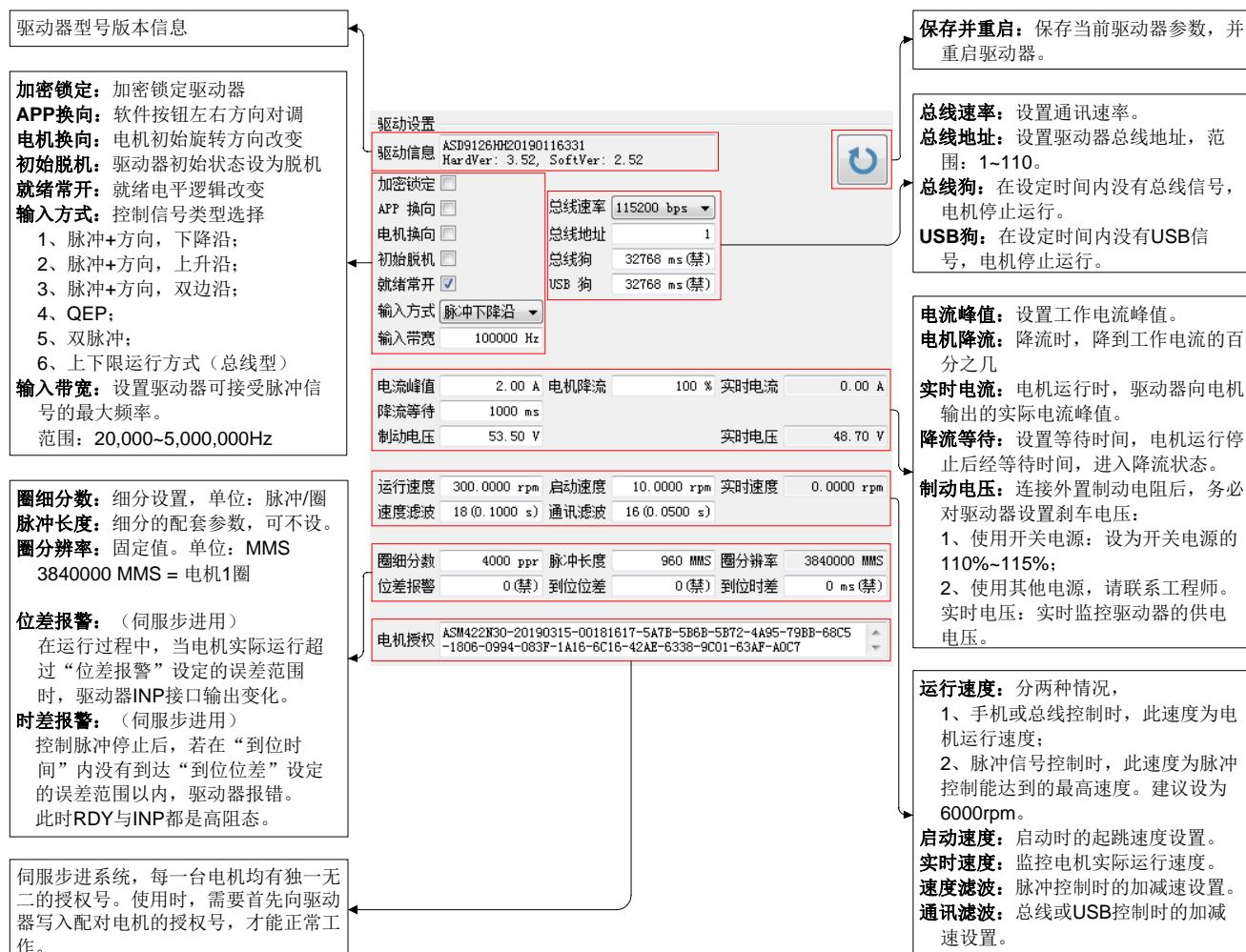
- 1, 连接设置区；
- 2, 参数设置区；
- 3, 运动控制区；
- 4, 传动设置区；
- 5, 状态监控区。



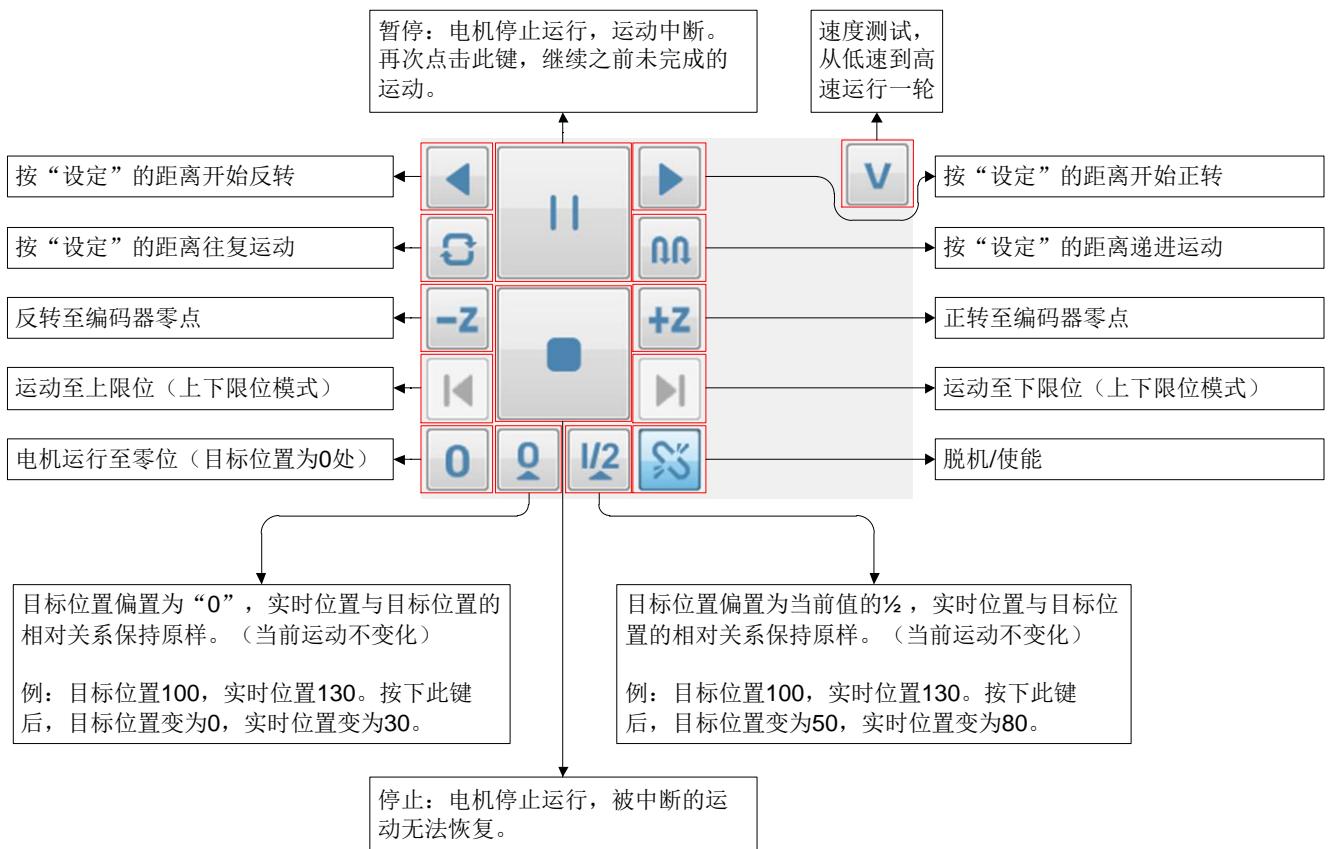
### 3.1.1 连接设置区



### 3.1.2 参数设置区



### 3.1.3 运动控制区



### 3.1.4 传动设置区



注：传动参数设置仅为表面参数，方便软件界面上进行距离换算时使用，与驱动器内部参数无关。

### 3.2.5 状态监控区

状态监控		
CPU 温度	42 °C	累计时间
底板温度		1:38:22
端口逻辑	STEP--0 DIR--0 FREE--1 ZERO--0 PA---1 PB---0 PZ---0 RDY--0 INF--1 BRK--0	2402 r
状态信息 驱动正常		

- CPU温度:** 显示驱动器芯片温度。  
**底板温度:** 显示驱动器散热底板温度（部分有此显示）。  
**累积时间:** 显示驱动器累积运行时间。  
**累计行程:** 显示驱动器累积运行距离。  
**端口逻辑:** 显示驱动器各IO引脚状态。  
**状态信息:** 显示驱动器状态，如有报错，会在此栏显示。

### 3.2.6 通讯 DLL

电脑版 AgeMotion 附带底层通讯 DLL，供用户调用，支持 C++ 与 C# 软件。

DLL 位置：安装目录/AgeMotion6/Develop；

Develop 文件夹内有两个/x64 与/x86 两个文件夹，/x64 内的 DLL 支持 64 位 Windows 系统，/x86 内的 DLL 支持 32 位 Windows 系统。

文件夹内有两个文件，分别是 AgeCOM.dll 与 AgeCOM.h，其中 AgeCOM.h 内包含了 DLL 的使用说明和相关例程，供用户参考。

# 4

# 驱动器通讯协议

支持通讯功能的步进电机驱动器，可利用 Modbus-RTU 协议访问驱动器的寄存器，进行读写操作，从而完成对驱动器的参数设置或者运行控制。

## 4.1 寄存器列表

地址	PLC	寄存器	类型	描述	读/写	记忆	授权	
0x0000	40001	Control	WORD	控制寄存器	W/W			
0x0002	40003	ErrorCode	WORD	故障寄存器	W/无			
0x0008	40009	InputType	WORD	输入类型寄存器	W/W	✓		
0x0010	40017	CurrentMax	UINT16	电流最大值寄存器	W/无			
0x0011	40018	CurrentMin	UINT16	电流最小值寄存器	W/无			
0x0012	40019	CurrentSet	UINT16	电流设定寄存器	W/W	✓		
0x0013	40020	CurrentLow	UINT16	电流降流寄存器	W/W	✓		
0x0014	40021	CurrentLowWT	UINT16	电流降流等待时间寄存器	W/W	✓		
0x0015	40022	Current	UINT16	实时相电流峰值	W/无			
0x001E	40031	VoltageBreak	UINT16	刹车电压寄存器	W/W	✓		
0x0020	40033	Position	INT64	电机实时位置寄存器	低 DW/低 DW QW/QW	✓	自由	
0x0021	40034							
0x0022	40035							
0x0023	40036							
0x0024	40037	PositionSet	INT64	电机设定位置寄存器	低 DW/低 DW QW/QW	✓		
0x0025	40038							
0x0026	40039							
0x0027	40040							
0x0028	40041	TResolution	UINT32	电机单齿分辨率寄存器	DW/无			
0x0029	40042							
0x002A	40043	PulseLength	UINT32	脉冲步进长度寄存器	DW/DW	✓		
0x002B	40044							
0x002C	40045	PulsePosition	INT32	电机脉冲实时位置寄存器	低 W/低 W DW/DW	✓		
0x002D	40046							
0x002E	40047	PulsePositionSet	INT32	电机脉冲设定位置寄存器	低 W/低 W DW/DW			
0x002F	40048							
0x0030	40049	PositionErrorAlarm	UINT32	位差报错寄存器	DW/DW	✓		
0x0031	40050							
0x0032	40051	PositionErrorAllowed	UINT32	到位位差寄存器	DW/DW	✓		
0x0033	40052							
0x0034	40053	TimeErrorAllowed	UINT16	到位时间寄存器	W/W	✓		

地址	PLC	寄存器	类型	描述	读/写	记忆	授权
0x0038	40057	PositionError	INT64	位置误差寄存器	QW/无		
0x0039	40058						
0x003A	40059						
0x003B	40060						
0x003C	40061	PulsePositionError	INT32	脉冲位置误差寄存器	DW/无		
0x003D	40062						
0x0040	40065	VelSet	UINT16	电机设定速度寄存器	W/W	✓	
0x0041	40066	VelStart	UINT16	电机启动速度寄存器	W/W	✓	
0x0042	40067	VelFilter	UINT16	速度滤波器寄存器	W/W	✓	
0x0043	40068	KV	UINT16	电机速度系数寄存器	W/无		
0x0044	40069	VelFilterCom	UINT16	速度滤波器(通讯)寄存器	W/W	✓	
0x0045	40070	Vel	SHORT	实时速度寄存器	S/无		
0x0046	40071	VelSetZero	UINT16	零限设定速度寄存器	W/W	✓	
0x0047	40072	VelFilterZero	UINT16	零限速度滤波寄存器	W/W	✓	
0x0060	40097	BusWDT	UINT16	总线看门狗定时寄存器	W/W	✓	
0x0061	40098	BusAddress	UINT16	总线地址寄存器	W/W	✓	
0x0062	40099	BusBand	UINT32	总线通讯速率寄存器	DW/DW	✓	
0x0063	40100						
0x0080	40129	Port	UINT16	端口寄存器	W/无		
0x0081	40130	PortHiFlag	UINT16	端口上升标记寄存器	W/W		
0x0082	40131	PortLoFlag	UINT16	端口下降标记寄存器	W/W		
0x0083	40132	PortFlipFlag	UINT16	端口翻转标记寄存器	W/W		
0x0084	40133	PortConfig	UINT64	端口配置寄存器	QW/QW	✓	
0x0085	40134						
0x0086	40135						
0x0087	40136						
0x0090	40145	InputBand	UINT32	输入带宽寄存器	DW/DW	✓	
0x0091	40146						
0x2000	48193	MotorSN0	UINT64	电机序列号 0 寄存器	QW/QW	✓	
0x2001	48194						
0x2002	48195						
0x2003	48196						
0x2004	48197	MotorSN1	UINT64	电机序列号 1 寄存器	QW/QW	✓	
0x2005	48198						
0x2006	48199						
0x2007	48200						
0x2008	48201	MotorSN2	UINT64	电机序列号 2 寄存器	QW/QW	✓	
0x2009	48202						
0x200A	48203						
0x200B	48204						
0x200C	48205	MotorSN3	UINT64	电机序列号 3 寄存器	QW/QW	✓	
0x200D	48206						

地址	PLC	寄存器	类型	描述	读/写	记忆	授权
0x200E	48207						
0x200F	48208						
0x2100	48449	MotorName	String	电机型号寄存器	QW QW/QW	√	
0x2101	48450						
0x2102	48451						
0x2103	48452						
0x2104	48453	MotorDate	UINT32	电机生产日期寄存器	DW/DW	√	
0x2105	48454						
0x2106	48455	MotorNum	UINT32	电机编号寄存器	DW/DW	√	
0x2107	48456						
0x2108	48457	Servo	UINT16	伺服寄存器	W/W	√	
0x0300	40769	CpuTemp	INT16	驱动器 CPU 温度寄存器	W/无		
0x0301	40770	SinkTemp	INT16	驱动器散热器温度寄存器	W/无		
0x8000		DriverName	String	驱动器型号寄存器	QW/无		
0x8001							
0x8002							
0x8003							
0x8008		DriverDate	String	驱动器生产日期寄存器	QW/无		
0x8009							
0x800A							
0x800B							
0x800C		DriverNum	String	驱动器编号寄存器	QW/无		
0x800D							
0x800E							
0x800F							

注：

- 1、内存单元为 WORD 类型，即 2 Bytes；
- 2、String 类型为标准 ASCII 编码，长度为 8 Bytes；
- 3、寄存器在内存中采用 Little-Endian，即低字节在低地址，高字节在高地址；  
寄存器通过 Modbus-RTU 传输时采用 Big-Endian，即高字节在低地址，低字节在高地址；
- 4、有记忆功能的寄存器再次上电后会自动恢复掉电前的记忆值；
- 5、如需保存最后的操作值，需进行“保存并重启”操作（不是通断电）；
- 6、建议上电后总线控制器对驱动器所有可写寄存器进行初始化；
- 7、电机运转中停电的实时位置记忆值会因为机械惯性原因有位置记忆偏差；
- 8、只能访问上述列表中的内存地址，寄存器的保留位请勿使用，必须为 0；
- 9、以上寄存器用户都可以自由使用，无需授权。

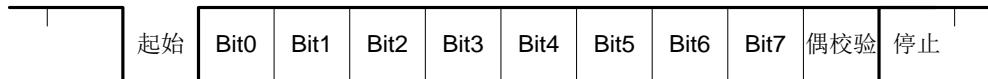
## 4.2 协议基本格式

Modbus-RTU/RS485 协议及标准请参见相关文档，本手册不做详细阐述。本手册仅对驱动器使用相关的协议及标准内容进行简单介绍。

## 4.2.1 通讯位模式

驱动器通讯位模式：

1个起始位，8个数据位（低位在前，高位在后），1个偶校验位，1个停止位。



## 4.2.2 Big-Endian/Little-Endian

**Big-Endian:** 多字节数据高位字节在内存低地址，低位字节在内存高地址。

**Little-Endian:** 多字节数据高位字节在内存高地址，低位字节在内存低地址。

## 4.2.3 通讯协议帧描述

ADU 即应用数据单元，PDU 即协议数据单元。

ADU			
地址域	功能码	数据	CRC
BYTE	BYTE	N*WORD	WORD
PDU			

传输时 PDU 寄存器地址及寄存器数据都是 Big-Endian 格式。

### 地址域：

0 广播地址，所有子节点可以识别广播地址，但不发送返回报文。

1~247 子节点 RTU 地址（RTU 即远程终端单元，这里指驱动器）。

### 功能码：

驱动器支持的功能码如下：

功能码	功能定义	数据长度
0x04	读单个寄存器	WORD
0x03	读多个寄存器	WORD/DWORD/QWORD
0x06	写单个寄存器	WORD
0x10	写多个寄存器	WORD/DWORD/QWORD

### 数据：

包含需操作的寄存器地址和操作数据。

### CRC 校验码：

进行 CRC 校验。

数据帧汇总	
操作	数据帧

读单个寄存器 0x04	请求报文	地址域	功能码	寄存器地址	寄存器数量	CRC
		1字节	0x04	2字节	2字节	2字节
	响应报文	地址域	功能码	数据字节数	返回数据	CRC
		1字节	0x04	1字节	2字节	2字节
读多个寄存器 0x03	请求报文	地址域	功能码	寄存器地址	寄存器数量	CRC
		1字节	0x03	2字节	2字节	2字节
	响应报文	地址域	功能码	数据字节数	返回数据	CRC
		1字节	0x03	1字节	2n字节	2字节
写单个寄存器 0x06	请求报文	地址域	功能码	寄存器地址	写入数据	CRC
		1字节	0x06	2字节	2字节	2字节
	响应报文	地址域	功能码	寄存器地址	写入数据	CRC
		1字节	0x06	2字节	2字节	2字节
写多个寄存器 0x10	请求报文	地址域	功能码	寄存器地址	寄存器数量	数据字节数
		1字节	0x10	2字节	2字节	1字节
	响应报文	地址域	功能码	寄存器地址	寄存器数量	CRC
		1字节	0x10	2字节	2字节	2字节

注：

- 1、n 表示数据长度，驱动器内存数据单元为 WORD 类型，即 2 字节，因此在多字节的读写操作中，字节数一定为 2 的倍数。
- 2、CRC 校验为 Little-Endian 格式，其他均为 Big-Endian 格式。

#### 4.2.4 CRC 校验

本公司产品通讯采用 CRC 循环冗余校验。

CRC 循环冗余校验域为两个字节，由发送设备计算后附加在报文最后，接收设备接收报文后重新计算 CRC 并与实际接收到的 CRC 比较。如果两个值不相等，则为错误。

CRC 计算时首先赋值 0xFFFF，然后将报文中的字节对其进行后续计算。只有字符中的 8 个数据位参与的运算，起始位、停止位和校验位不参与运算。

CRC 与每个字节异或，然后结果向最低有效位 LSB 方向右移 1 位，最高有效位 MSB 填 0，然后提取并检查移出的 LSB：如果 LSB 为 1，则 CRC 与 0xA001 异或；如果 LSB 为 0，则不进行异或操作。这个过程将重复直到执行完 8 次移位。完成最后 8 次移位后进行 CRC 与下一字节异或，然后又重复上面描述的 8 次移位。当所有报文中子节都运算之后得到的 CRC 最终值。

CRC 流程：

- 1、CRC赋值0xFFFF；
  - 2、报文第一个字节与CRC低字节异或并赋值CRC；
  - 3、CRC寄存器向LSB方向右移1位，MSB填零，检测移出的LSB：  
0：重复步骤3；  
1：CRC异或多项式值0xA001（1010 0000 0000 0001B）并赋值CRC；
  - 4、重复步骤3直到完成8次移位；
  - 5、对报文中的下一字节重复步骤2、3、4，继续此操作直至所有报文被处理完毕；
  - 6、得到最终CRC；
  - 7、当放置CRC 值于报文时，低字节在前，高字节在后；
- CRC流程的报文字节不包含CRC本身；

CPU 运算能力较弱时可采用查表法，此处不再讨论，请参考相关文档。

#### 4.2.5 MODBUS 异常响应

当 RTU 收到不正确的指令时，会返回异常相应报文。

异常响应报文格式：

地址域	差错码	异常码	CRC
BYTE	BYTE	BYTE	WORD

地址域：请求 ADU 中的地址；

差错码：MODBUS 功能码 + 0x80，即 0x84,0x83,0x46 或者 0x90；

异常码：指令异常相应的原因。驱动器支持的异常码如下：

异常码	说明
0x01	功能码错
0x02	寄存器地址错
0x03	寄存器数量错

### 4.3 寄存器的操作

#### 4.3.1 操作模式

寄存器只支持 6 种操作模式：

- 1、ReadWORD：读取 1 个 WORD；
- 2、ReadDWORD：读取 2 个 WORD；
- 3、ReadQWORD：读取 4 个 WORD；
- 4、WriteWORD：写入 1 个 WORD；
- 5、WriteDWORD：写入 2 个 WORD；
- 6、WriteQWORD：写入 4 个 WORD。

#### 4.3.2 操作规则

- 1、每次只能操作 1 个寄存器；
- 2、寄存器只能进行支持的操作；
- 3、未授权操作按错误指令响应或不响应。

## 4.4 寄存器详解及报文示例

### 4.4.1 控制寄存器 Control

地址: 0x0000      PLC 对应地址: 40001

操作: ReadWORD/WriteWORD, 无记忆

Bit	名称	类型	值域	缺省	描述
0	保存并重启 Reset	Bit	0~1	0	读 : 恒 0 写 : 0 无作用 1 保存参数并重启驱动器, 完成后自动清零 修改设置后, 如想保存, 必须进行此操作。
1	驱动器数据恢复出厂值 ResetValue	Bit	0~1	0	读 : 恒 0 写 : 0 无作用 1 无条件恢复出厂值, 完成后自动重启
2	脱机/使能 Free	Bit	0~1	0	读写: 0 驱动器使能 1 驱动器脱机, 处于低功耗模式, 输出电流为 0
3	暂停/继续 Pause	Bit	0~1	0	读写: 0 继续之前状态或等待新指令 1 无条件暂停 (保留之前的目标位置)
4	运动至下限传感器 MovePSL	Bit	0~1	0	读写: 0 静止时无作用; 如正向下限运动中, 运动停止 1 运动至下限传感器, 完成后自动清零
5	运动至上限传感器 MovePSH	Bit	0~1	0	读写: 0 静止时无作用; 如正向上限运动中, 运动停止 1 运动至上限传感器, 完成后自动清零
说明: MovePSL、MovePSH 使用时 InputType.PulseType 必须为 8。 上下限传感器分别接驱动器控制端口的步进、方向接口。 当驱动运动至下限传感器时往下限方向运动被禁止, 往上限方向运动正常; 当驱动运动至上限传感器时往上限方向运动被禁止, 往下限方向运动正常。 MovePSL、MovePSH 指令完成时该位自动清 0 , PositionSet = Position。					
6	驱动器数据标志位 DataLost	Bit	0~1	0	读 : 0 记忆数据正常 1 记忆数据丢失 写 : 0 无作用 1 强制记忆数据正常
说明: 如果驱动器保存的参数丢失, 该标志位置 1, 驱动器进入故障状态。 在清除该错误标志位前, 必须用总线控制器对驱动器各参数进行重新设置, 否则, 运动方向、速度、细分等参数的错误设置有可能引发安全事故。					
7	重启标志位 RestartFlag	Bit	0~1	0	读写: 0/1 无作用, 重启后为缺省值
说明: 驱动器启动后, 该位缺省值为 0, 用户可以置 1, 使用过程中若发现该位为 0, 则表示驱动器已经重新启动过。					
8	设定位置偏移至 0 OffsetToZero	Bit	0~1	0	读 : 恒 0 写 : 0 无作用 1 PositionSet 偏移至 0, 完成后自动清零

	<p>说明：整个过程相当于以 PositionSet 当前值处作为原点，重新建立坐标系。</p> <p>例 1：电机运动结束，驱动器处于等待状态时，Position=PositionSet。此时寄存器置 1，PositionSet, Position 同时变为 0。（PlusePositionSet 与 PlusePosition 也作相应变化）</p> <p>例 2：电机运动进行中时，如果 PositionSet、Position 分别为 150、90，执行后，PositionSet, Position 分别为 0、-60，电机运动不受此影响。（PlusePositionSet 与 PlusePosition 也作相应变化）</p>				
9	设定位置的一半处偏移至 0 OffsetToZeroInHalf	Bit	0~1	0	<p>读：恒 0 写：0 无作用 1 PositionSet 的 1/2 处偏移至 0，完成后自动清零</p>
	<p>说明：整个过程相当于 PositionSet/2 值处作为原点，重新建立坐标系。</p> <p>例如 PositionSet、Position 分别为 150、90，执行后，新的零点为执行前 PositionSet/2 处的位置，PositionSet、Position 分别变为 75、15，电机运动不受此影响。（PlusePositionSet 与 PlusePosition 也作相应变化）</p>				
10	向低位运动至编码器零点 MoveLZero	Bit	0~1	0	<p>读：恒 0 写：0 无作用 1 向低位运动至编码器零点，完成后自动清零</p>
11	向高位运动至编码器零点 MoveHZero	Bit	0~1	0	<p>读：恒 0 写：0 无作用 1 向高位运动至编码器零点，完成后自动清零</p>
12	停止 Stop	Bit	0~1	0	<p>读：恒 0 写：0 无作用 1 电机运动立刻停止（以通讯滤波设定的加减速）</p>
13	停止在当前位置 StopCurrent	Bit	0~1	0	<p>读：恒 0 写：0 无作用 1 电机运动立刻慢下来，并换向，返回到接收指令 瞬间的位置停下来（以通讯滤波设定的加减速）</p>
	<p>Stop 指令与 Pause 指令的区别：Stop 指令会修改目标位置，之前输入的目标位置被覆盖。Pause 指令只是暂停，恢复后仍会朝着之前输入的目标位置继续前进。</p>				
14~ 15	保留				

优先级：Bit0 最高，Bit15 最低。

**提示** 请慎用 **Reset**、**ResetValue** 功能。**Reset**、**ResetValue** 使用间隔最小 **20S**，作用如同上下电，每天动作次数不能大于 **10** 次。

**提示** 电机处于静止待命状态时才能进行 **Free** 或 **Enable** 操作。

RTU 即远程终端单元，这里指驱动器总线地址。所有报文示例中设 RTU 地址为 0x01，电机为 50 齿。

#### 报文示例——写入——保存并重启 Reset

	RTU 地址	功能码	寄存器地址	数据	CRC
发送报文	0x01	0x06	0x00 0x00	0x00 0x01	0x48 0x0A
说明	将寄存器 0x0000 的第 0 位置 1，保存参数并重启驱动器。				

**提示** **CRC** 校验码，低字节在前，高字节在后；其他均为高字节在前，低字节在后。

#### 报文示例——写入——复位数据 ResetValue

	RTU 地址	功能码	寄存器地址	数据	CRC
发送报文	0x01	0x06	0x00 0x00	0x00 0x02	0x08 0x0B
说明	将寄存器 0x0000 的第 1 位置 1，所有参数恢复出厂值。				

报文示例——写入——脱机 Free					
	RTU 地址	功能码	寄存器地址	数据	CRC
发送报文	0x01	0x06	0x00 0x00	0x00 0x04	0x88 0x09
响应报文	0x01	0x06	0x00 0x00	0x00 0x04	0x88 0x09
说明	将寄存器 0x0000 的第 2 位置 1，将驱动器设为脱机状态。				
报文示例——写入——使能 Enable					
发送报文	0x01	0x06	0x00 0x00	0x00 0x00	0x89 0xCA
响应报文	0x01	0x06	0x00 0x00	0x00 0x00	0x89 0xCA
说明	将寄存器 0x0000 的置 0，使能驱动器。				
报文示例——写入——暂停 Pause					
发送报文	0x01	0x06	0x00 0x00	0x00 0x08	0x88 0x0C
响应报文	0x01	0x06	0x00 0x00	0x00 0x08	0x88 0x0C
说明	将寄存器 0x0000 的第 3 位置 1，所有当前运行暂停。				
报文示例——写入——运动至下限位置 MovePSL					
发送报文	0x01	0x06	0x00 0x00	0x00 0x10	0x88 0x06
响应报文	0x01	0x06	0x00 0x00	0x00 0x10	0x88 0x06
说明	将寄存器 0x0000 的第 4 位置 1，电机向下限位开关运行。				
报文示例——写入——运动至上限位置 MovePSH					
发送报文	0x01	0x06	0x00 0x00	0x00 0x20	0x88 0x12
响应报文	0x01	0x06	0x00 0x00	0x00 0x20	0x88 0x12
说明	将寄存器 0x0000 的第 5 位置 1，电机向上限位开关运行。				
报文示例——写入——数据丢失位清零 DataLost					
发送报文	0x01	0x06	0x00 0x00	0x00 0x40	0x88 0x3A
响应报文	0x01	0x06	0x00 0x00	0x00 0x40	0x88 0x3A
说明	将寄存器 0x0000 的第 6 位置 1，数据丢失位强制清零				
报文示例——写入——重启标志位置 1 RestartFlag					
发送报文	0x01	0x06	0x00 0x00	0x00 0x80	0x88 0x6A
响应报文	0x01	0x06	0x00 0x00	0x00 0x80	0x88 0x6A
说明	将寄存器 0x0000 的第 7 位置 1，重启标志位置 1。				
报文示例——写入——设定位偏移至 0 OffsetToZero					
发送报文	0x01	0x06	0x00 0x00	0x01 0x00	0x88 0x5A
响应报文	0x01	0x06	0x00 0x00	0x01 0x00	0x88 0x5A
说明	将寄存器 0x0000 的第 8 位置 1，PositionSet 偏移至 0。				

## 报文示例——写入——设定位置一半处偏移至 0 OffsetToZeroInHalf

	RTU 地址	功能码	寄存器地址	数据	CRC
发送报文	0x01	0x06	0x00 0x00	0x02 0x00	0x88 0xAA
响应报文	0x01	0x06	0x00 0x00	0x02 0x00	0x88 0xAA
说明	将寄存器 0x0000 的第 9 位置 1, PositionSet 的 1/2 处偏移至 0。				

## 报文示例——向低位运动至编码器零点 MoveLZero

	RTU 地址	功能码	寄存器地址	数据	CRC
发送报文	0x01	0x06	0x00 0x00	0x04 0x00	0x8B 0x0A
响应报文	0x01	0x06	0x00 0x00	0x04 0x00	0x8B 0x0A
说明	将寄存器 0x0000 的第 10 位置 1, 控制电机向低位运动至编码器零点。				

## 报文示例——写入——向高位运动至编码器零点 MoveHZero

	RTU 地址	功能码	寄存器地址	数据	CRC
发送报文	0x01	0x06	0x00 0x00	0x08 0x00	0x8E 0x0A
响应报文	0x01	0x06	0x00 0x00	0x08 0x00	0x8E 0x0A
说明	将寄存器 0x0000 的第 11 位置 1, 控制电机向高位运动至编码器零点。				

## 报文示例——读取——控制寄存器 Control

	RTU 地址	功能码	寄存器地址	数据	CRC
发送报文	0x01	0x04	0x00 0x00	0x00 0x01	0x31 0xCA
响应报文	RTU 地址		功能码	数据字节数	数据内容
	0x01	0x04	0x02	0x00 0x08	0xB8 0xF6
说明	读取控制寄存器 0x0000 内的值。				

## 4.4.2 故障寄存器 ErrorCode

地址: 0x0002 PLC 对应地址: 40003

说明: 当驱动器报警或报错时, 可读取此寄存器, 获取故障原因; 无报警或无报错时, 返回值 0

操作: ReadWORD, 无记忆

故障码	含义	报错	报警
0x0101	驱动器内部固件程序紊乱	✓	
0x0102	驱动器内部固件看门狗溢出	✓	
0x0103	驱动器内部校准错误	✓	
0x0104	驱动器内部驱动电压欠压	✓	
0x0105	驱动器内部驱动电压过压	✓	
0x0106	驱动器内部 CPU 自检错	✓	
0x0107	驱动器内部主板自检错	✓	
0x0108	驱动器内部型号自检错	✓	
0x0109	驱动器内部硬件自检错	✓	
0x010A	驱动器内部固件自检错	✓	
0x010B	驱动器内部执行超时	✓	
0x010C	驱动器内部运算超时	✓	

0x0116	驱动器内部记忆数据丢失	✓	
0x0201	供电电压欠压	✓	
0x0202	供电电压过压	✓	
0x0203	驱动器低温报警		✓
0x0204	驱动器高温报错	✓	
0x0209	电机错相	✓	
0x020F	电机开路	✓	
0x0211	电机 A 相开路	✓	
0x0212	电机 B 相开路	✓	
0x0216	机机制动超时	✓	
0x0217	位差超限	✓	
0x0218	位差超时	✓	
0x0301	电机短路	✓	
0x0302	电机启动时短路	✓	
0x0303	电机检查错相时短路	✓	
0x030F	电机过流	✓	
0x0410	电机授权号错	✓	
0x0411	电机编码器错	✓	

**提示** 报警可以继续使用，报错必须重启。

报文示例——读取——控制寄存器 Control					
发送报文	RTU 地址	功能码	寄存器地址	数据	CRC
	0x01	0x04	0x00 0x02	0x00 0x01	0x90 0xA
响应报文	RTU 地址	功能码	数据字节数	数据内容	CRC
	0x01	0x04	0x02	0x04 0x11	0x7B 0xFC
说明	读取故障寄存器 0x0002，将返回数据（例中为 0x0411）对照故障表，确定故障内容。				

#### 4.4.3 输入类型寄存器 InputType

地址: 0x0008

PLC 对应地址: 40009

说明: 设置驱动器脉冲端口和方向端口的输入信号类型

操作: ReadWORD/WriteWORD, 有记忆

Bit	名称	类型	值域	缺省	描述
0~3	PulseType	UINT4	0~16	记忆值	读写: 0 方向+脉冲下降沿 1 方向+脉冲上升沿 2 方向+脉冲双边沿 3 QEP 8 下限+上限 其他设置为保留值请勿用
4~12	保留				
13	PadDir	Bit	0~1	记忆值	读写: 0/1 设定通讯模式下的电机初始正反转方向
14	QEPDir	Bit	0~1	记忆值	读写: 0/1 设定脉冲模式下的电机初始正反转方向

15	Free/Enable	Bit	0~1	记忆值	读写: 0 驱动器初始使能 1 驱动器初始脱机
----	-------------	-----	-----	-----	----------------------------

PulseType: 采用步进脉冲信号输入时, 建议 PLC 或控制器端采用 QEP 方式 (需控制器支持)。

报文示例——写入——输入类型寄存器 InputType					
	RTU 地址	功能码	寄存器地址	数据	CRC
发送报文	0x01	0x06	0x00 0x08	0x00 0x08	0x09 0xCE
响应报文	0x01	0x06	0x00 0x08	0x00 0x08	0x09 0xCE
说明	向寄存器 0x0008 写入 0x0008, 切换到上下限运行方式 (通讯控制时)。				
报文示例——读取——输入类型寄存器 InputType					
	RTU 地址	功能码	寄存器地址	数据	CRC
发送报文	0x01	0x04	0x00 0x08	0x00 0x01	0xB0 0x08
	0x01	0x04	0x02	0x00 0x08	0xB8 0xF6
说明	读取控制寄存器 0x0008, 返回数据 0x0008。				

#### 4.4.4 电流最大值寄存器 CurrentMax

地址: 0x0010                   PLC 对应地址: 40017

说明: 驱动器输出相电流最大值 (峰值), 该寄存器出厂时固化

操作: ReadWORD/WriteWORD, 无记忆

Bit	名称	类型	值域	缺省	描述
0~15	CurrentMax	UINT16	100I <sub>PMax</sub>	100I <sub>PMax</sub>	读 : 驱动器相电流最大峰值 写 : 非法

I<sub>PMax</sub> 是驱动器允许的相电流最大峰值, 驱动器出厂时 I<sub>PMax</sub>、CurrentMax 已固化, 用户不能更改。

驱动器相电流最大峰值

I<sub>PMax</sub> = 0.01 × CurrentMax , 单位: 安培 (A)。

即

$$\text{CurrentMax} = 100 \times I_{\text{PMax}}$$

例:

AHD8323XX 相电流最大峰值 I<sub>PMax</sub> = 3.25A, 对应的 CurrentMax = 325。

AHD8326XX 相电流最大峰值 I<sub>PMax</sub> = 6.50A, 对应的 CurrentMax = 650。

报文示例——读取——电流最大值 CurrentMax					
	RTU 地址	功能码	寄存器地址	数据	CRC
发送报文	0x01	0x04	0x00 0x10	0x00 0x01	0x30 0x0F
	0x01	0x04	0x02	0x02 0x8A	0x39 0xF7
说明	读取电流最大值寄存器 0x0010, 将返回数据 (例中为 0x028A) 换算得电流值。				

#### 4.4.5 电流最小值寄存器 CurrentMin

地址: 0x0011

PLC 对应地址: 40018

说明: 驱动器输出相电流最小值(峰值), 该寄存器出厂时固化

操作: ReadWORD/WriteWORD, 无记忆

Bit	名称	类型	值域	缺省	描述
0~15	CurrentMin	UINT16	100I <sub>PMin</sub>	100I <sub>PMin</sub>	读 : 驱动器相电流最小峰值 写 : 非法

I<sub>PMin</sub> 是驱动器允许的相电流最小峰值, 驱动器出厂时 I<sub>PMin</sub>、CurrentMin 已固化, 用户不能更改。

驱动器相电流最大峰值

I<sub>PMax</sub> = 0.01 × CurrentMin , 单位: 安培 (A)。

即

$$\text{CurrentMin} = 100 \times I_{PMax}$$

例:

AHD8323XX 相电流最小峰值 I<sub>PMax</sub> = 1.0A, 对应的 CurrentMin = 100。

AHD8326XX 相电流最小峰值 I<sub>PMax</sub> = 2.0A, 对应的 CurrentMin = 200。

报文示例——读取——电流最小值 CurrentMin					
发送报文	RTU 地址	功能码	寄存器地址	数据	CRC
	0x01	0x04	0x00 0x11	0x00 0x01	0x61 0xCF
响应报文	RTU 地址	功能码	数据字节数	数据内容	CRC
	0x01	0x04	0x02	0x00 0xFA	0x39 0x73
说明	读取电流最小值寄存器 0x0011, 将返回数据(例中为 0x00FA)换算得电流值。				

#### 4.4.6 电流设定寄存器 CurrentSet

地址: 0x0012

PLC 对应地址: 40019

说明: 设置驱动器的输出相电流(峰值)

操作: ReadWORD/WriteWORD, 有记忆

Bit	名称	类型	值域	缺省	描述
0~15	CurrentSet	UINT16	100I <sub>PMin</sub> ~100I <sub>PMax</sub>	记忆值	读写: 驱动器输出相电流峰值设定

驱动器相电流设定峰值 I<sub>PSet</sub> 由用户确定, 不能超过相电流最大峰值 I<sub>PMax</sub>。注意驱动器相电流峰值 I<sub>P</sub> 和电机相电流有效值 I<sub>RMS</sub> 的对应关系。

驱动器相电流设定峰值

I<sub>PSet</sub> = 0.01 × CurrentSet , 单位: 安培 (A)。

即

$$\text{CurrentSet} = 100 \times I_{PSet}$$

例:

AHD8326XX 相电流峰值范围 2.00~6.50A。

当用户需设定驱动器相电流峰值 I<sub>PSet</sub>= 5.50A, 可取 CurrentSet = 550。

当用户需设定驱动器相电流峰值 I<sub>PSet</sub>= 3.00A, 可取 CurrentSet = 300。

报文示例——写入——电流设定寄存器 CurrentSet					
	RTU 地址	功能码	寄存器地址	数据	CRC
发送报文	0x01	0x06	0x00 0x12	0x01 0xF4	0x29 0xD8
响应报文	0x01	0x06	0x00 0x12	0x01 0xF4	0x29 0xD8
说明	向寄存器 0x0012 写入 0x01F4，设定驱动器输出电流峰值为 5.0A。				

报文示例——读取——电流设定寄存器 CurrentSet					
	RTU 地址	功能码	寄存器地址	数据	CRC
发送报文	0x01	0x04	0x00 0x12	0x00 0x01	0x91 0xCF
响应报文	0x01	0x04	0x02	0x01 0x5E	0x39 0x58
说明	读取电流设定寄存器 0x0012，将返回数据（例中为 0x015E）换算得电流值。				

#### 4.4.7 电流降流寄存器 CurrentLow

地址: 0x0013                   PLC 对应地址: 40020

说明: 设置驱动器降流状态下的降流百分比

操作: ReadWORD/WriteWORD, 有记忆

Bit	名称	类型	值域	缺省	描述
0~15	CurrentLow	UINT16	30~100	50	读写: 驱动器相电流降流百分比

驱动器输出相电流降流值

$$I_{PLow} = I_{PSet} \times \text{CurrentLow \%}, \text{ 单位: 安培 (A)}.$$

例: CurrentLow 设为 30, 降流电流值即为正常工作电流的 30%。

降流电流小, 电机锁定力矩小, 启动瞬间力矩小, 电机停止状态下温升小。

降流电流大, 电机锁定力矩大, 启动瞬间力矩大, 电机停止状态下温升大。

报文示例——写入——电流降流寄存器 CurrentLow					
	RTU 地址	功能码	寄存器地址	数据	CRC
发送报文	0x01	0x06	0x00 0x13	0x00 0x1E	0xF8 0x07
响应报文	0x01	0x06	0x00 0x13	0x00 0x1E	0xF8 0x07
说明	向寄存器 0x0013 写入 0x001E，设定闲时驱动器输出电流降流为 30%。				

报文示例——读取——电流降流寄存器 CurrentLow					
	RTU 地址	功能码	寄存器地址	数据	CRC
发送报文	0x01	0x04	0x00 0x13	0x00 0x01	0xC0 0x0F
响应报文	0x01	0x04	0x02	0x00 0x1E	0x39 0x38
说明	读取电流降流寄存器 0x0013，将返回数据（例中为 0x001E）换算得驱动器输出电流降流百分比。				

#### 4.4.8 电流降流等待时间寄存器 CurrentLowWT

地址: 0x0014                  PLC 对应地址: 40021

说明: 设置驱动器从静止到进入降流状态的时间

操作: ReadWORD/WriteWORD, 有记忆

Bit	名称	类型	值域	缺省	描述
0~15	CurrentLowWT	UINT16	50~65535	记忆值	读写: 驱动器相电流降流等待时间 (ms)

驱动器处于等待状态时, 等待时间超过降流等待时间即进入降流状态, 相电流峰值自动调整为  $I_{PLow}$ 。

驱动器相电流降流等待时间

$$T_{CurrentLow} = 0.001 \times CurrentLow\ WT, \text{ 单位: 秒 (S)}.$$

相电流降流等待时间范围: 0.05~65.535 秒。

例: CurrentLowWT 设为 100, 降流等待时间为 0.1 秒。

CurrentLowWT 设为 2000, 降流等待时间为 2 秒。

##### 报文示例——写入——电流降流等待时间寄存器 CurrentLowWT

	RTU 地址	功能码	寄存器地址	数据	CRC
发送报文	0x01	0x06	0x00 0x14	0x01 0xC2	0x49 0xCF
响应报文	0x01	0x06	0x00 0x14	0x01 0xC2	0x49 0xCF
说明	向寄存器 0x0014 写入 0x01C2, 设定驱动器停止 0.45 秒后进入降流状态。				

##### 报文示例——读取——电流降流等待时间寄存器 CurrentLowWT

	RTU 地址	功能码	寄存器地址	数据	CRC
发送报文	0x01	0x04	0x00 0x14	0x00 0x01	0x71 0xCE
响应报文	RTU 地址	功能码	数据字节数	数据内容	CRC
说明	读取电流降流等待时间寄存器 0x0014, 将返回数据 (例中为 0x02BC) 换算得驱动器停止 0.7 秒后进入降流状态。				

#### 4.4.9 实时电流寄存器 Current

地址: 0x0015                  PLC 对应地址: 40022

说明: 当前的驱动器输出相电流值 (峰值)

操作: ReadWORD

Bit	名称	类型	值域	缺省	描述
0~15	Current	UINT16	0~65535	实时值	读: 驱动器输出相电流峰值

读取该寄存器, 即可获取驱动器当前输出相电流值 (峰值)。

当前相电流峰值=Current×100 (A)

例:

当读取 Current=150, 则实时电流 =1.50A。

当读取 Current=556, 则实时电流 =5.56A。

报文示例——读取——电流设定寄存器 CurrentSet					
发送报文	RTU 地址	功能码	寄存器地址	数据	CRC
	0x01	0x04	0x00 0x15	0x00 0x01	0x20 0x0E
响应报文	RTU 地址	功能码	数据字节数	数据内容	CRC
	0x01	0x04	0x02	0x01 0x5E	0x39 0x58
说明	读取实时电流寄存器 0x0015，将返回数据（例中为 0x015E）换算得电流值。				

#### 4.4.10 电机实时位置寄存器 Position

地址: 0x0020~0x0023

PLC 对应地址: 40033~40036

说明: 电机的当前位置 (开环系统为模拟值, 闭环系统为精确值)

操作: 低 DWORD 支持 ReadDWORD/WriteDWORD, ReadQWORD/WriteQWORD, 有记忆

Bit	名称	类型	值域	缺省	描述
0~63	Position	INT64	-0x20000000 00000000 ~ +0x1FFFFFFF FFFFFFFF	记忆值	读 : 电机实时位置 写 : 电机实时位置重新赋值 单位: 最小细分步 (MMS)

\*当主机不支持 INT64 时, 用 ReadDWORD/WriteDWORD 访问低 DWORD, 高 DWORD 不能单独访问。

提示

最小细分步 MMS 对应的电机转数:

$$\text{Revolution}_{\text{MMS}} = \frac{1}{\text{TResolution} \times T}, \text{ 单位: 转 (r)}.$$

TResolution: 电机单齿分辨率;

T: 电机齿数。

通常 TResolution = 76800, T=50, 那么

$$\text{MMS} = \frac{1}{3840000} \text{ 转}$$

即 1 转 = 3840000MMS

##### 报文示例——写入——电机实时位置寄存器 Position

发送报文	RTU 地址	功能码	寄存器地址	寄存器字数	寄存器字节数	数据	CRC
	0x01	0x10	0x00 0x20	0x00 0x04	0x08	0x00 0x00 0x00 0x00 0x02 0x49 0xF0 0x00	0xA3 0xAB
响应报文	RTU 地址	功能码	寄存器地址	寄存器字数			CRC
	0x01	0x10	0x00 0x20	0x00 0x04			0xC0 0x00

说明 向寄存器 0x0020 写入新的电机实时位置。

##### 报文示例——读取——电机实时位置寄存器 Position

发送报文	RTU 地址	功能码	寄存器地址	寄存器字数	CRC	
	0x01	0x03	0x00 0x20	0x00 0x04	0x45 0xC3	
响应报文	RTU 地址	功能码	寄存器字节数	数据		CRC
	0x01	0x03	0x08	0x00 0x00 0x00 0x00 0x02 0x49 0xF0 0x00		0x01 0xB9

说明 读取电机实时位置寄存器。

报文示例——写入低 DWORD——电机实时位置寄存器 Position										
发送报文	RTU 地址	功能码	寄存器地址	寄存器字数	寄存器字节数	数据	CRC			
	0x01	0x10	0x00 0x20	0x00 0x02	0x04	0x02 0x49 0xF0 0x00	0x65 0xD9			
响应报文	RTU 地址	功能码	寄存器地址	寄存器字数			CRC			
	0x01	0x10	0x00 0x20	0x00 0x02			0x40 0x02			
说明	向寄存器 0x0020 写入新的电机实时位置（只对低 DWORD 操作）。									
报文示例——读取低 DWORD——电机实时位置寄存器 Position										
发送报文	RTU 地址	功能码	寄存器地址	寄存器字数			CRC			
	0x01	0x03	0x00 0x20	0x00 0x02			0xC5 0xC1			
响应报文	RTU 地址	功能码	寄存器字节数	数据			CRC			
	0x01	0x03	0x04	0x02 0x49 0xF0 0x00			0x6E 0x5D			
说明	读取电机实时位置寄存器（只对低 DWORD 操作）。									

#### 4.4.11 电机设定位置寄存器 PositionSet

地址: 0x0024~0x0027

PLC 对应地址: 40037~40040

说明: 电机的目标位置 (开环系统为模拟值, 闭环系统为精确值)

操作: 低 DWORD 支持 ReadDWORD/WriteDWORD, 支持 ReadQWORD/WriteQWORD

Bit	名称	类型	值域	缺省	描述
0~63	PositionSet	INT64	-0x20000000 00000000 ~ +0x1FFFFFFF FFFFFFF	记忆值	读写: 电机设定位置 每次成功写操作将激活电机定位运动 单位: 最小细分步 (MMS)

\*当主机不支持 INT64 时, 用 ReadDWORD/WriteDWORD 访问低 DWORD, 高 DWORD 不能单独访问。

电机运动方向:

电机实时位置 Position → 电机设定位置 PositionSet, 运动到达时 Position = PositionSet。

可先设置 Control.Pause, 再设置 Position、PositionSet, 最后清除 Control.Pause, 这样电机在设置位置过程中不会出现异动。

例:

Position = 10000, PositionSet = 12000, 电机将正转 2000MMS;

Position = 10000, PositionSet = 9000, 电机将反转 1000MMS。

提示

怎样让电机朝某方向恒定运行？

假设电机转速 12000rpm，

设置 Position = -0x2000 0000 0000 0000，

设置 PositionSet = 0x1FFF FFFF FFFF FFFF，

则运行时间

$$T_{Run} = \frac{PositionSet - Position}{TResolution \times T \times RSpeed} = \frac{0x4000000000000000}{76800 \times 50 \times 12000} \approx 100079992\text{分钟}$$

，约 190 年。因此设置足够的位置差即能让电机恒定运行。

#### 报文示例——写入——电机设定位置寄存器 PositionSet

发送 报文	RTU 地址	功能码	寄存器地址	寄存器 字数	寄存器 字节数	数据	CRC
	0x01	0x10	0x00 0x24	0x00 0x04	0x08	0x00 0x00 0x00 0x00 0x02 0x49 0xF0 0x00	0x52 0x64
响应 报文	RTU 地址	功能码	寄存器地址	寄存器字数			
	0x01	0x10	0x00 0x24	0x00 0x04			
说明	向寄存器 0x0024 写入新的电机设定位置。						

#### 报文示例——读取——电机设定位置寄存器 PositionSet

发送 报文	RTU 地址	功能码	寄存器地址	寄存器字数	CRC
	0x01	0x03	0x00 0x24	0x00 0x04	0x04 0x02
响应 报文	RTU 地址	功能码	寄存器 字节数	数据	
	0x01	0x03	0x08	0x00 0x00 0x00 0x00 0x02 0x49 0xF0 0x00	
说明	读取电机设定位置寄存器。				

#### 报文示例——写入低 DWORD——电机设定位置寄存器 PositionSet

发送 报文	RTU 地址	功能码	寄存器地址	寄存器 字数	寄存器 字节数	数据	CRC
	0x01	0x10	0x00 0x24	0x00 0x02	0x04	0x02 0x49 0xF0 0x00	0x64 0x2A
响应 报文	RTU 地址	功能码	寄存器地址	寄存器字数			
	0x01	0x10	0x00 0x24	0x00 0x02			
说明	向寄存器 0x0024 写入新的电机设定位置（只对低 DWORD 操作）。						

#### 报文示例——读取低 DWORD——电机设定位置寄存器 PositionSet

发送 报文	RTU 地址	功能码	寄存器地址	寄存器字数	CRC
	0x01	0x03	0x00 0x24	0x00 0x02	0x84 0x00
响应 报文	RTU 地址	功能码	寄存器 字节数	数据	
	0x01	0x03	0x04	0x02 0x49 0xF0 0x00	
说明	读取电机设定位置寄存器（只对低 DWORD 操作）。				

#### 4.4.12 电机单齿分辨率寄存器 TResolution

地址: 0x0028~0x0029      PLC 对应地址: 40041~40042

说明: 电机单齿分辨率, 该寄存器出厂时固化

操作: ReadDWORD, 无记忆

Bit	名称	类型	值域	缺省	描述
0~31	TResolution	UINT32	1~0xFFFFFFFF	76800	读 : 步进电机单齿分辨率 写 : 非法 单位: 最小细分步 (MMS)

电机单齿分辨率即电机转过 1 个机械齿所对应的最小细分步数。

电机转数

$$\text{Revolution} = \frac{\Delta\text{Position}}{\text{TResolution} \times T}, \text{ 单位: 转 (r)}.$$

即

$$\Delta\text{Position} = \text{Revolution} \times \text{TResolution} \times T$$

$\Delta\text{Position}$ : 以最小细分步表示的运动长度;

$T$ : 电机齿数, 步进电机齿数一般为 50, 也有 100、200 或其他特殊齿数。

例:

$T = 50$ ,  $\text{Position} = 10000000$ ,  $\text{PositionSet} = 11920000$ ,

则  $\Delta\text{Position} = \text{PositionSet} - \text{Position} = 1920000$ ,  $\text{Revolution} = 0.5r$ , 即电机运动半转。

报文示例——读取——电机单齿分辨率寄存器 TResolution					
发送报文	RTU 地址	功能码	寄存器地址	寄存器字数	CRC
	0x01	0x03	0x00 0x28	0x00 0x02	0x44 0x03
响应报文	RTU 地址	功能码	寄存器字节数	数据	CRC
	0x01	0x03	0x04	0x00 0x01 0x2C 0x00	0xB7 0x33
说明	读取电机单齿分辨率寄存器。				

#### 4.4.13 脉冲步进长度寄存器 PulseLength

地址: 0x002A~0x002B      PLC 对应地址: 40043~40044

说明: 单个脉冲的电机运转距离 (即细分设定, 单位: MMS)

操作: ReadDWORD/WriteDWORD, 有记忆

Bit	名称	类型	值域	缺省	描述
0~31	PulseLength	UINT32	1~3840000	记忆值	读写: 以最小细分步表示的单脉冲步进长度

单脉冲的电机运动转数

$$\text{Revolution}_{\text{Step}} = \frac{\text{PulseLength}}{\text{TResolution} \times T}, \text{ 单位: 转 (r)}.$$

通常  $\text{TResolution}=76800$ ,  $T=50$ ,

即

$$\text{细分值} = \frac{1}{\text{RevolutionStep}} = \frac{3840000}{\text{PulseLength}} \text{ (脉冲/圈)}$$

因此电机运动 1 转对应的脉冲不一定是整数，但仍然可以正常使用而没有其它数学误差。

例：

$T = 50, \text{PulseLength} = 1536$ , 则细分 2500 脉冲/转。

$T = 100, \text{PulseLength} = 1536$ , 则细分 5000 脉冲/转。

报文示例——写入——脉冲步进长度寄存器 PulseLength										
发送报文	RTU 地址	功能码	寄存器地址	寄存器字数	寄存器字节数	数据	CRC			
	0x01	0x10	0x00 0x2A	0x00 0x02	0x04	0x00 0x00 0x06 0x00	0x72 0x68			
响应报文	RTU 地址	功能码	寄存器地址	寄存器字数			CRC			
	0x01	0x10	0x00 0x2A	0x00 0x02			0x60 0x00			
说明	向寄存器 0x002A 写入脉冲步进长度值。									
报文示例——读取——脉冲步进长度寄存器 PulseLength										
发送报文	RTU 地址	功能码	寄存器地址	寄存器字数			CRC			
	0x01	0x03	0x00 0x2A	0x00 0x02			0xE5 0xC3			
响应报文	RTU 地址	功能码	寄存器字节数	数据			CRC			
	0x01	0x03	0x04	0x00 0x00 0x06 0x00			0xF9 0x93			
说明	读取脉冲步进长度寄存器，将返回数据（例中为 0x0000 0x0600）换算得细分值。									

#### 4.4.14 电机脉冲实时位置寄存器 PulsePosition

地址: 0x002C~0x002D      PLC 对应地址: 40045~40046

说明: 电机的当前位置, 以 PulseLength 为单位 (开环系统为模拟值, 闭环系统为精确值)

操作: 低 WORD 支持 ReadWORD/WriteWORD, ReadDWORD/WriteDWORD, 无记忆

Bit	名称	类型	值域	缺省	描述
0~31	PulsePosition	INT32	-0x80000000 ~ +0x7FFFFFFF	无	读 : 电机脉冲实时位置, 由 Position 计算而来 写 : 电机脉冲实时位置重新赋值 单位: 脉冲步进长度 (PulseLength)

\*当主机不支持 INT32 时, 用 ReadWORD/WriteWORD 访问低 WORD, 高 WORD 不能单独访问。

PulsePosition 的变化量 ( $\Delta$ PulsePosition) 对应的电机转数

$$\text{Revolution}_{\text{PulsePosition}} = \frac{\Delta \text{PulsePosition} \times \text{PulseLength}}{\text{TResolution} \times \text{T}}, \text{ 单位: 转 (r).}$$

PulsePosition 对应的 Position

$$\text{Position} = \text{PulsePosition} \times \text{PulseLength}, \text{ 单位: 最小细分步 (MMS)}$$

对应 Position 的 PulsePosition

$$\text{PulsePosition} = \frac{\text{Position}}{\text{PulseLength}}, \text{ 单位: PulseLength}$$

写 PulsePosition 时 Position 的位置会相应更新, 读 PulsePosition 时则是上述公式的整数化计算结果。

Position 及 PositionSet 是 64 位整数, 对控制系统的要求较高; PulsePosition 及 PulsePositionSet 是 32 位整数, 对控制系统的要求较低, 如果把驱动器设为 1000 步/转, 程序员会更容易理解电机的实际位置。例:

Position = 19, PulseLength = 5, 读 PulsePosition = 3;

Position = 19, PulseLength = 5, 写 PulsePosition = 3, 则 Position = 15。

报文示例——写入——电机脉冲实时位置寄存器 PulsePosition										
发送报文	RTU 地址	功能码	寄存器地址	寄存器字数	寄存器字节数	数据	CRC			
	0x01	0x10	0x00 0x2C	0x00 0x02	0x04	0x00 0x00 0x27 0x10	0xEB 0xDE			
响应报文	RTU 地址	功能码	寄存器地址	寄存器字数			CRC			
	0x01	0x10	0x00 0x2C	0x00 0x02			0x80 0x01			
说明	向寄存器 0x002C 写入电机脉冲实时位置。									
报文示例——读取——电机脉冲实时位置寄存器 PulsePosition										
发送报文	RTU 地址	功能码	寄存器地址	寄存器字数			CRC			
	0x01	0x03	0x00 0x2C	0x00 0x02			0x05 0xC2			
响应报文	RTU 地址	功能码	寄存器字节数	数据			CRC			
	0x01	0x03	0x04	0x00 0x00 0x27 0x10			0xE0 0x0F			
说明	读取电机脉冲实时位置寄存器, 将返回数据 (例中为 0x0000 0x2710) 换算得脉冲实时位置值。									

报文示例——写入低 WORD——电机脉冲实时位置寄存器 PulsePosition					
	RTU 地址	功能码	寄存器地址	数据	CRC
发送报文	0x01	0x06	0x00 0x2C	0x27 0x10	0x52 0x3F
响应报文	0x01	0x06	0x00 0x2C	0x27 0x10	0x52 0x3F
说明	向寄存器 0x002C 写入电机脉冲实时位置。(只对低 WORD 操作)				
报文示例——读取低 WORD——电机脉冲实时位置寄存器 PulsePosition					
	RTU 地址	功能码	寄存器地址	数据	CRC
发送报文	0x01	0x04	0x00 0x2C	0x00 0x01	0xF0 0x03
响应报文	RTU 地址	功能码	数据字节数	数据内容	CRC
	0x01	0x04	0x02	0x27 0x10	0xA3 0x0C
说明	读取电机脉冲实时位置寄存器，将返回数据(例中为 0x2710)换算得脉冲实时位置值。(只对低 WORD 操作)				

#### 4.4.15 电机脉冲设定位置寄存器 PulsePositionSet

地址: 0x002E~0x002F

PLC 对应地址: 40047~40048

说明: 电机的目标位置, 以 PulseLength 为单位(开环系统为模拟值, 闭环系统为精确值)

操作: 低 WORD 支持 ReadWORD/WriteWORD, 支持 ReadDWORD/WriteDWORD, 无记忆

Bit	名称	类型	值域	缺省	描述
0~31	PulsePositionSet	INT32	-0x80000000 ~ +0x7FFFFFFF	无	读: 电机脉冲设定位置, 由 PositionSet 计算而来 写: 电机脉冲设定位置重置 每次成功写操作将激活电机定位运动 单位: 脉冲步进长度(PulseLength)

\*当主机不支持 INT32 时, 用 ReadWORD/WriteWORD 访问低 WORD, 高 WORD 不能单独访问。

电机运动方向: 脉冲实时位置 PulsePostion → 脉冲设定位置 PulsePositionSet, 运动到达时 Position = PositionSet, 注意运动的条件是 Position 和 PositionSet, 而不是 PulsePositon 和 PulsePositionSet。 PulsePosition 到达时 Position 可能还没到达, 而 Position 到达时 PulsePosition 一定也已到达。

例:

驱动器设置为 1000 步/转, 则

PulsePosition = 10000, PulsePositionSet = 12000, 电机将正转 2000 步, 即 2 转;

PulsePosition = 10000, PulsePositionSet = 9000, 电机将反转 1000 步, 即 1 转。

报文示例——写入——电机脉冲设定位置寄存器 PulsePositionSet							
发送报文	RTU 地址	功能码	寄存器地址	寄存器字数	寄存器字节数	数据	CRC
	0x01	0x10	0x00 0x2E	0x00 0x02	0x04	0x00 0x00 0x27 0x10	0x6A 0x07
响应报文	RTU 地址	功能码	寄存器地址	寄存器字数			CRC
	0x01	0x10	0x00 0x2E	0x00 0x02			0x21 0xC1
说明	向寄存器 0x002E 写入电机脉冲设定位置。						

报文示例——读取——电机脉冲设定位置寄存器 PulsePositionSet						
发送报文	RTU 地址	功能码	寄存器地址	寄存器字数		CRC
	0x01	0x03	0x00 0x2E	0x00 0x02		0XA4 0x02
响应报文	RTU 地址	功能码	寄存器字节数	数据		CRC
	0x01	0x03	0x04	0x00 0x00 0x27 0x10		0xE0 0x0F
说明	读取电机脉冲设定位置寄存器，将返回数据（例中为 0x0000 0x2710）换算得脉冲设定位置值。					

报文示例——写入低 WORD——电机脉冲设定位置寄存器 PulsePositionSet					
	RTU 地址	功能码	寄存器地址	数据	CRC
发送报文	0x01	0x06	0x00 0x2E	0x27 0x10	0XF3 0xFF
响应报文	0x01	0x06	0x00 0x2E	0x27 0x10	0XF3 0xFF
说明	向寄存器 0x002E 写入电机脉冲设定位置。（只对低 WORD 操作）				
报文示例——读取低 WORD——电机脉冲设定位置寄存器 PulsePositionSet					
发送报文	RTU 地址	功能码	寄存器地址	数据	CRC
	0x01	0x04	0x00 0x2E	0x00 0x01	0x51 0xC3
响应报文	RTU 地址	功能码	数据字节数	数据内容	CRC
	0x01	0x04	0x02	0x27 0x10	0xA3 0x0C
说明	读取电机脉冲设定位置寄存器，将返回数据（例中为 0x2710）换算得脉冲设定位置值。（只对低 WORD 操作）				

#### 4.4.16 位差报错寄存器 PositionErrorAlarm

地址: 0x0030~0x0031

PLC 对应地址: 40049~40050

说明: 当电机运动误差 (PositionError) 大于该寄存器的设定值时，驱动器报错

操作: ReadDWORD/WriteDWORD，有记忆

Bit	名称	类型	值域	缺省	描述
0~31	PositionErrorAlarm	UINT32	0~3840000*32	记忆值	读写: 位差报错设定值 单位: 最小细分步, MMS

提示 此寄存器为伺服功能专用。

通常情况下，当  $0 \leq \text{设定值} < 76800$  时，位差报错功能关闭。

在光栅闭环应用中，当  $0 \leq \text{设定值} < 384$  时，位差报错功能关闭。

当位差报错功能关闭时，读取此寄存器的返回值为 0。

报文示例——写入——位差报错寄存器 PositionErrorAlarm						
发送报文	RTU 地址	功能码	寄存器地址	寄存器字数	寄存器字节数	CRC
	0x01	0x10	0x00 0x30	0x00 0x02	0x04	0XF0 0xBB
响应报文	RTU 地址	功能码	寄存器地址	寄存器字数		CRC
	0x01	0x10	0x00 0x30	0x00 0x02		0X41 0xC7
说明	向寄存器 0x0030 写入位差报错寄存器。					

报文示例——读取——位差报错寄存器 PositionErrorAlarm						
发送 报文	RTU 地址	功能码	寄存器地址	寄存器字数		CRC
	0x01	0x03	0x00 0x30	0x00 0x02		0XC4 0x04
响应 报文	RTU 地址	功能码	寄存器 字节数	数据		CRC
	0x01	0x03	0x04	0x00 0x00 0x00 0x00		0xFA 0x33
说明	读取位差报错寄存器，将返回数据（例中为 0x0000 0x0000）换算得位差报错参数。					

#### 4.4.17 到位位差寄存器 PositionErrorAllowed

地址: 0x0032~0x0033      PLC 对应地址: 40051~40052

说明: 允许的最大静态误差, 如静态误差超过该寄存器的设定值时, 驱动器报错

操作: ReadDWORD/WriteDWORD, 有记忆

Bit	名称	类型	值域	缺省	描述
0~31	PositionErrorAlarm	UINT32	0 ~ 3840000*32	记忆值	读写: 到位位差设定值 单位: 最小细分步, MMS

(提示) 此寄存器为伺服功能专用。

通常情况下, 当 0≤设定值<76800 时, 到位位差功能关闭。

在光栅闭环应用中, 当 0≤设定值<384 时, 到位位差功能关闭。

当到位位差功能关闭时, 读取此寄存器的返回值为 0。

报文示例——写入——到位位差寄存器 PositionErrorAllowed							
发送 报文	RTU 地址	功能码	寄存器地址	寄存器 字数	寄存器 字节数	数据	CRC
	0x01	0x10	0x00 0x32	0x00 0x02	0x04	0x00 0x00 0x00 0x00	0X71 0x62
响应 报文	RTU 地址	功能码	寄存器地址	寄存器字数			CRC
	0x01	0x10	0x00 0x32	0x00 0x02			0XE0 0x07
说明	向寄存器 0x0032 写入到位位差参数。						

报文示例——读取——到位位差寄存器 PositionErrorAllowed							
发送 报文	RTU 地址	功能码	寄存器地址	寄存器字数			CRC
	0x01	0x03	0x00 0x32	0x00 0x02			0X65 0xC4
响应 报文	RTU 地址	功能码	寄存器 字节数	数据			CRC
	0x01	0x03	0x04	0x00 0x00 0x00 0x00			0xFA 0x33
说明	读取到位位差寄存器, 将返回数据（例中为 0x0000 0x0000）换算得到位差参数。						

#### 4.4.18 到位时间寄存器 TimeErrorAllowed

地址: 0x0034      PLC 对应地址: 40053

说明: 当电机在该寄存器的设定时间内, 未到达期望位置 (离期望位置的距离大于到位位差), 驱动器报错

操作: ReadDWORD/WriteDWORD, 有记忆

Bit	名称	类型	值域	缺省	描述
0~15	PositionErrorAlarm	UINT32	0~300	记忆值	读写: 到位时间设定值 单位: 毫秒, ms

**提示** 此寄存器为伺服功能专用。

当  $0 \leq \text{设定值} < 100$  时, 到位时间功能关闭。

当到位时间功能关闭时, 读取此寄存器的返回值为 0。

**报文示例——写入——到位时间寄存器 TimeErrorAllowed**

	RTU 地址	功能码	寄存器地址	数据	CRC
发送报文	0x01	0x06	0x00 0x34	0x00 0x00	0XC8 0x04
响应报文	0x01	0x06	0x00 0x34	0x00 0x00	0XC8 0x04
说明	向寄存器 0x0034 写入到位时间参数。				

**报文示例——读取——到位时间寄存器 TimeErrorAllowed**

	RTU 地址	功能码	寄存器地址	数据	CRC
发送报文	0x01	0x04	0x00 0x34	0x00 0x01	0x70 0x04
响应报文	RTU 地址	功能码	数据字节数	数据内容	CRC
说明	读取到位时间寄存器, 将返回数据 (例中为 0x0000) 换算得到位时间参数。				

#### 4.4.19 位置误差寄存器 PositionError

地址: 0x0038~0x003B      PLC 对应地址: 40057~40060

说明: 电机运动误差, 包括动态误差和静态误差

操作: ReadQWORD, 无记忆

Bit	名称	类型	值域	缺省	描述
0~63	PositionError	INT64	-0x200000000 00000000 ~ +0x1FFFFFFF FFFFFF	无	读 : 电机运动误差, 由 Position 计算而来 写 : 非法 单位: 最小细分步, MMS

**提示** 此寄存器为伺服功能专用。

**报文示例——读取——位置误差寄存器 PositionError**

发送报文	RTU 地址	功能码	寄存器地址	寄存器字数	CRC
	0x01	0x03	0x00 0x38	0x00 0x04	0xC5 0xC4

响应报文	RTU 地址	功能码	寄存器字节数	数据	CRC
	0x01	0x03	0x08	0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00	0x95 0xD7
说明	读取位置误差寄存器。				

#### 4.4.20 脉冲位置误差寄存器 PulsePositionError

地址: 0x003C~0x003D                   PLC 对应地址: 40061~40062

说明: 电机运动误差, 包括动态误差和静态误差, 以 PulseLength 为单位

操作: ReadDWORD, 无记忆

Bit	名称	类型	值域	缺省	描述
0~31	PulsePositionError	INT32	-0x20000000 ~ +0x1FFFFFFF	无	读 : 电机运动误差, 由 Position 计算而来 写 : 非法 单位: 脉冲步进长度, PulseLength

(提示) 此寄存器为伺服功能专用。

报文示例——读取——到位位差寄存器 PositionErrorAllowed					
发送报文	RTU 地址	功能码	寄存器地址	寄存器字数	CRC
	0x01	0x03	0x00 0x3C	0x00 0x02	0X04 0x07
响应报文	RTU 地址	功能码	寄存器字节数	数据	CRC
	0x01	0x03	0x04	0x00 0x00 0x00 0x00	0xFA 0x33
说明	读取脉冲位置误差寄存器, 将返回数据 (例中为 0x0000 0x0000) 换算得到脉冲位置误差参数。				

#### 4.4.21 电机设定速度寄存器 VelSet

地址: 0x0040                   PLC 对应地址: 40065

说明: 设置电机的运行速度

操作: ReadWORD/WriteWORD, 有记忆

Bit	名称	类型	值域	缺省	描述
0~15	VelSet	UINT16	1~38400	记忆值	读写: 电机设定速度

电机转速

$$RSpeed = \frac{VelSet \times KV \times 60000}{TResolution \times T}, \text{ 单位: 转/分 (rpm)}.$$

通常 KV=20, 齿数 T=50, TResolution=76800, 以上公式可简化为

$$VelSet = \frac{16 \times RSpeed}{5}$$

例: 如需电机转速 RSpeed 为 60rpm 时, 设 VelSet = 192,

如需电机转速 RSpeed 为 300rpm 时, 设 VelSet = 960。

VelSet 值域 1~38400, 即启动速度范围 0.3125~12000rpm。

报文示例——写入——电机设定速度寄存器 VelSet					
	RTU 地址	功能码	寄存器地址	数据	CRC
发送报文	0x01	0x06	0x00 0x40	0x00 0xC0	0X88 0x4E
响应报文	0x01	0x06	0x00 0x40	0x00 0xC0	0X88 0x4E
说明	向寄存器 0x0040 写入电机运转速度。				
报文示例——读取——电机设定速度寄存器 VelSet					
	RTU 地址	功能码	寄存器地址	数据	CRC
发送报文	0x01	0x04	0x00 0x40	0x00 0x01	0x30 0x1E
响应报文	0x01	0x04	0x02	0x00 0xC0	0XB9 0x60
说明	读取电机设定速度寄存器，将返回数据（例中为 0x00C0）换算得速度值。				

#### 4.4.22 电机启动速度寄存器 VelStart

地址: 0x0041                   PLC 对应地址: 40066

说明: 电机运行的起跳转速

操作: ReadWORD/WriteWORD, 有记忆

Bit	名称	类型	值域	缺省	描述
0~15	VelStart	UINT16	1~192	记忆值	读写: 电机启动速度

电机加速时会从静止状态直接阶跃至启动速度，而电机减速时会从启动速度直接进入静止状态。较小的启动速度能使电机的启停瞬间更平稳，但会需要更多的启停时间。

电机启动转速

$$RSpeed_{Start} = \frac{VelStart \times 1200000}{TResolution \times T}, \text{ 单位: 转/分 (rpm)}.$$

通常 KV=20, 齿数 T=50, TResolution=76800, 以上公式可简化为

$$Vel_{Start} = \frac{16 \times RSpeed_{Start}}{5}$$

例: 如需电机启动速度 RSpeed<sub>Start</sub> 为 30rpm 时, 设 VelStart = 96。

如需电机启动速度 RSpeed<sub>Start</sub> 为 10rpm 时, 设 VelStart = 32。

VelStart 值域 1~192, 即启动速度范围 0.3125~60rpm。

报文示例——写入——电机启动速度寄存器 VelStart					
	RTU 地址	功能码	寄存器地址	数据	CRC
发送报文	0x01	0x06	0x00 0x41	0x00 0x60	0XD9 0xF6
响应报文	0x01	0x06	0x00 0x41	0x00 0x60	0XD9 0xF6
说明	向寄存器 0x0041 写入电机启动速度。				
报文示例——读取——电机启动速度寄存器 VelStart					
	RTU 地址	功能码	寄存器地址	数据	CRC
发送报文	0x01	0x04	0x00 0x41	0x00 0x01	0x61 0xDE
响应报文	0x01	0x04	0x02	0x00 0x60	0XB9 0x18
说明	读取电机启动速度寄存器，将返回数据（例中为 0x0060）换算得启动速度值。				

#### 4.4.23 速度滤波器寄存器 VelFilter

地址: 0x0042      PLC 对应地址: 40067

说明: 速度滤波等级设定, 即脉冲控制时的加减速曲线选择

操作: ReadWORD/WriteWORD, 有记忆

Bit	名称	类型	值域	缺省	描述
0~15	VelFilter	UINT16	0~31	记忆值	读写: 电机速度滤波器

引入滤波器使电机的启动及运行更平稳, 这种处理能比外部运动控制脉冲更平稳。如果合理地设置滤波器, 即使单脉冲步进长度很大 (例: 1 个脉冲 1 转) 或直接要求运动很多转, 电机也能很平稳地运行。

滤波器会引起动态定位误差, 即电机运行中实时位置和目标位置有一定误差。

滤波器不会引起静态定位误差, 即电机延后  $T_{Delay}$  到达目标位置, 静止后定位误差为 0。

滤波器会产生电机运动延时, 近似公式

$$T_{Delay} \approx 0.1 \times 2^{\frac{VelFilter-18}{2}}, \text{ 单位: 秒 (S).}$$

例:

$T = 50$ ,  $RSpeed_{Start} = 30\text{rpm}$ , 缺省延时为 0.2S。

在此条件下最大动态定位误差

$$Position_{Err} \approx 2^{\frac{VelFilter-18}{2}} \times \frac{RSpeed}{2400}, \text{ 单位: 转 (r).}$$

$RSpeed = 600\text{rpm}$  时最大动态误差为 0.5 转。

$RSpeed_{Start}$  增大,  $Position_{Err}$  随之减小;  $RSpeed_{Start}$  减小,  $Position_{Err}$  随之增大。

例:

$VelFilter = 0$ , 则  $T_{Delay} = 0\text{S}$ 。

$VelFilter = 16$ , 则  $T_{Delay} \approx 0.05\text{S}$ 。

$VelFilter = 18$ , 则  $T_{Delay} \approx 0.1\text{S}$ 。

$VelFilter = 20$ , 则  $T_{Delay} \approx 0.2\text{S}$ 。

$VelFilter = 22$ , 则  $T_{Delay} = 0.4\text{S}$ 。

报文示例——写入——速度滤波器寄存器 VelFilter					
	RTU 地址	功能码	寄存器地址	数据	CRC
发送报文	0x01	0x06	0x00 0x42	0x00 0x14	0X29 0xD1
响应报文	0x01	0x06	0x00 0x42	0x00 0x14	0X29 0xD1
说明	向寄存器 0x0042 写入脉冲控制时的速度滤波等级。				
报文示例——读取——速度滤波器寄存器 VelFilter					
发送报文	RTU 地址	功能码	寄存器地址	数据	CRC
	0x01	0x04	0x00 0x42	0x00 0x01	0x91 0xDE
响应报文	RTU 地址	功能码	数据字节数	数据内容	CRC
	0x01	0x04	0x02	0x00 0x14	0XB9 0x3F
说明	读取速度滤波器寄存器, 将返回数据 (例中为 0x0014) 换算得速度滤波等级。				

#### 4.4.24 电机速度系数寄存器 KV

地址: 0x0043      PLC 对应地址: 40068

操作: ReadWORD, 无记忆, 该寄存器出厂时固化

Bit	名称	类型	值域	缺省	描述
0~15	KV	UINT16	1~65535	20	读 : 电机速度系数, 用于计算速度 写 : 非法

#### 4.4.25 速度滤波器 (通讯) 寄存器 VelFilterCom

地址: 0x0044      PLC 对应地址: 40069

说明: 该速度滤波器对 总线/USB 的运动控制起作用。参数意义参考 VelFilter 寄存器

操作: ReadWORD/WriteWORD, 有记忆

Bit	名称	类型	值域	缺省	描述
0~15	VelFilterCom	UINT16	0~31	记忆值	读写: 电机速度滤波器

报文示例——写入——速度滤波器 (通讯) 寄存器 VelFilterCom					
	RTU 地址	功能码	寄存器地址	数据	CRC
发送报文	0x01	0x06	0x00 0x44	0x00 0x15	0X08 0x10
响应报文	0x01	0x06	0x00 0x44	0x00 0x15	0X08 0x10
说明	向寄存器 0x0044 写入通讯控制时的速度滤波等级。				
报文示例——读取——速度滤波器 (通讯) 寄存器 VelFilterCom					
发送报文	RTU 地址	功能码	寄存器地址	数据	CRC
	0x01	0x04	0x00 0x44	0x00 0x01	0x71 0xDF
响应报文	RTU 地址	功能码	数据字节数	数据内容	CRC
	0x01	0x04	0x02	0x00 0x15	0X78 0xFF
说明	读取速度滤波器寄存器, 将返回数据 (例中为 0x0015) 换算得速度滤波等级。				

#### 4.4.26 电机实时速度寄存器 Vel

地址: 0x0045      PLC 对应地址: 40070

说明: 电机的实时运行速度

操作: ReadWORD/WriteWORD, 无记忆

Bit	名称	类型	值域	缺省	描述
0~15	Vel	SHORT	-32768~32767	无	读: 电机的实时运行速度

$$RSpeed = \frac{5 \times Vel}{16}, RSpeed \text{ 为电机的实时转速, 单位 rpm.}$$

例: 如读取 Vel=192, 则电机实时转速 RSpeed 为 60rpm,

如读取 Vel=960, 则电机实时转速 RSpeed 为 300rpm.

负数表示反向。

报文示例——读取——电机设定速度寄存器 Vel					
发送报文	RTU 地址	功能码	寄存器地址	数据	CRC
	0x01	0x04	0x00 0x45	0x00 0x01	0x20 0x1F
响应报文	RTU 地址	功能码	数据字节数	数据内容	CRC
	0x01	0x04	0x02	0x00 0xC0	0xB9 0x60
说明	读取电机实时速度寄存器，将返回数据（例中为 0x00C0）换算得速度值。				

#### 4.4.27 电机零限设定速度寄存器 VelSetZero

地址: 0x0046 PLC 对应地址: 40071

说明: 设置电机的回零位、回限位的运行速度

操作: ReadWORD/WriteWORD, 有记忆

Bit	名称	类型	值域	缺省	描述
0~15	VelSetZero	UINT16	1~38400	记忆值	读写: 电机回零位、回限位的设定速度

$$\text{VelSetZero} = \frac{16 \times \text{RSpeed}}{5}, \text{ RSpeed 为期望的真实速度, 单位 rpm}$$

例: 如需回零限的电机转速 RSpeed 为 60rpm 时, 设 VelSetZero = 192,

如需回零限的电机转速 RSpeed 为 100rpm 时, 设 VelSetZero = 320。

VelSetZero 值域 1~38400。

报文示例——写入——电机零限设定速度寄存器 VelSetZero					
	RTU 地址	功能码	寄存器地址	数据	CRC
发送报文	0x01	0x06	0x00 0x46	0x00 0xC0	0x68 0x4F
响应报文	0x01	0x06	0x00 0x46	0x00 0xC0	0x68 0x4F
说明	向寄存器 0x0046 写入电机回零限开关的运转速度。				
报文示例——读取——电机零限设定速度寄存器 VelSetZero					
发送报文	RTU 地址	功能码	寄存器地址	数据	CRC
	0x01	0x04	0x00 0x46	0x00 0x01	0xD0 0x1F
响应报文	RTU 地址	功能码	数据字节数	数据内容	CRC
	0x01	0x04	0x02	0x00 0xC0	0XB9 0x60
说明	读取电机回零限开关的设定速度寄存器, 将返回数据（例中为 0x00C0）换算得速度值。				

#### 4.4.28 零限速度滤波寄存器 VelFilterZero

地址: 0x0047 PLC 对应地址: 40072

说明: 该速度滤波器对 回零位/上下限位 的运动控制起作用。参数意义参考 VelFilter 寄存器

操作: ReadWORD/WriteWORD, 有记忆

Bit	名称	类型	值域	缺省	描述
0~15	VelFilterZero	UINT16	0~31	记忆值	读写: 电机回零限速度滤波器

报文示例——写入——速度滤波器（通讯）寄存器 VelFilterZero					
	RTU 地址	功能码	寄存器地址	数据	CRC
发送报文	0x01	0x06	0x00 0x47	0x00 0x15	0xF8 0x10
响应报文	0x01	0x06	0x00 0x47	0x00 0x15	0xF8 0x10
说明	向寄存器 0x0047 写入回零限开关时的速度滤波等级。				
报文示例——读取——速度滤波器（通讯）寄存器 VelFilterZero					
发送报文	RTU 地址	功能码	寄存器地址	数据	CRC
	0x01	0x04	0x00 0x47	0x00 0x01	0x81 0xDF
响应报文	RTU 地址	功能码	数据字节数	数据内容	CRC
	0x01	0x04	0x02	0x00 0x15	0x78 0xFF
说明	读取速度滤波器寄存器，将返回数据（例中为 0x0015）换算得速度滤波等级。				

#### 4.4.29 总线看门狗定时寄存器 BusWDT

地址: 0x0060                   PLC 对应地址: 40097

说明: 总线看门狗定时器

操作: ReadWORD/WriteWORD, 有记忆

Bit	名称	类型	值域	缺省	描述
0~15	BusWDT	UINT16	2~65535	记忆值	读写: 总线看门狗定时 单位: 毫秒 ms

若 BusWDT  $\geq 0x8000$ , 即看门狗定时大于等于 32.768 秒时, 总线看门狗关闭。

若 BusWDT < 0x8000, 则驱动器在 BusWDT 设定的时间内没有收到总线指令会自动暂停, 收到总线指令后自动取消因总线引起的自动暂停。

报文示例——写入——总线看门狗定时寄存器 BusWDT					
	RTU 地址	功能码	寄存器地址	数据	CRC
发送报文	0x01	0x06	0x00 0x60	0x03 0xE8	0X89 0x6A
响应报文	0x01	0x06	0x00 0x60	0x03 0xE8	0X89 0x6A
说明	向寄存器 0x0060 写入总线看门狗定时器参数。				
报文示例——读取——总线看门狗定时寄存器 BusWDT					
发送报文	RTU 地址	功能码	寄存器地址	数据	CRC
	0x01	0x04	0x00 0x60	0x00 0x01	0x31 0xD4
响应报文	RTU 地址	功能码	数据字节数	数据内容	CRC
	0x01	0x04	0x02	0x03 0xE8	0XB9 0x8E
说明	读取总线看门狗定时寄存器, 将返回数据（例中为 0x03E8）换算得总线看门狗启动时间。				

#### 4.4.30 总线地址寄存器 BusAddress

地址: 0x0061      PLC 对应地址: 40098

说明: 设定当前驱动器的总线地址 (RTU 地址), 用于给驱动器编号

操作: ReadWORD/WriteWORD, 有记忆

Bit	名称	类型	值域	缺省	描述
0~15	BusAddress	UINT16	1~247	记忆值	读写: 总线地址设定值

报文示例——写入——总线地址寄存器 BusAddress					
	RTU 地址	功能码	寄存器地址	数据	CRC
发送报文	0x01	0x06	0x00 0x61	0x00 0x01	0X19 0xD4
响应报文	0x01	0x06	0x00 0x61	0x00 0x01	0X19 0xD4
说明	向寄存器 0x0061 写入总线地址。				
报文示例——读取——总线地址寄存器 BusAddress					
发送报文	RTU 地址	功能码	寄存器地址	数据	CRC
	0x01	0x04	0x00 0x61	0x00 0x01	0x60 0x14
响应报文	RTU 地址	功能码	数据字节数	数据内容	CRC
	0x01	0x04	0x02	0x00 0x01	0x78 0xF0
说明	读取总线地址寄存器, 将返回数据 (例中为 0x0001) 换算得该驱动器总线地址。				

#### 4.4.31 总线通讯速率寄存器 BusBand

地址: 0x0062~0x0063      PLC 对应地址: 40099~40100

说明: 设置 RS485 总线的通讯带宽

操作: ReadDWORD/WriteDWORD, 有记忆

Bit	名称	类型	值域	缺省	描述
0~31	BusBand	UINT32	9600~250000	记忆值	读写: 总线带宽 (比特率) 设定值 单位: 比特/秒, bps

常用总线带宽: 9600bps, 19200bps, 38400bps, 57600bps, 115200bps, 250000bps。

报文示例——写入——总线通讯速率寄存器 BusBand										
发送 报文	RTU 地址	功能码	寄存器地址	寄存器 字数	寄存器 字节数	数据	CRC			
	0x01	0x10	0x00 0x62	0x00 0x02	0x04	0x00 0x00 0x25 0x80	0X6F 0x6E			
响应 报文	RTU 地址	功能码	寄存器地址	寄存器字数			CRC			
	0x01	0x10	0x00 0x62	0x00 0x02			0XE0 0x16			
说明	向寄存器 0x0062 写入总线通讯速率参数。									
报文示例——读取——总线通讯速率寄存器 BusBand										
发送 报文	RTU 地址	功能码	寄存器地址	寄存器字数			CRC			
	0x01	0x03	0x00 0x62	0x00 0x02			0X65 0xD5			

响应报文	RTU地址	功能码	寄存器字节数	数据	CRC
	0x01	0x03	0x04	0x00 0x00 0x25 0x80	0xE1 0x03
说明	读取总线通讯速率寄存器，将返回数据（例中为 0x0000 0x2580）换算得当前总线通讯速率。				

#### 4.4.32 端口寄存器 Port

地址: 0x0080 PLC 对应地址: 40129

说明: Port 端口状态寄存器

操作: ReadWORD, 无记忆

Bit	名称	类型	值域	缺省	描述
0	Step	Bit	0~1	0	读 : 0 STEP/QA 光耦 LED 导通 1 STEP/QA 光耦 LED 截止 写 : 非法
1	Dir	Bit	0~1	0	读 : 0 DIR/QB 光耦 LED 导通 1 DIR/QB 光耦 LED 截止 写 : 非法
2	Free	Bit	0~1	0	读 : 0 FR/EN 光耦 LED 导通 1 FR/EN 光耦 LED 截止 写 : 非法
3	Zero	Bit	0~1	0	读 : 0 Zero 光耦 LED 导通 1 Zero 光耦 LED 截止 写 : 非法
4~7	保留				
8	Ready	Bit	0~1	0	读 : 0 RDY 截止 1 RDY 导通 写 : 非法
9	INP	Bit	0~1	0	读 : 0 INP 截止 1 INP 导通 写 : 非法
10	BRK	Bit	0~1	0	读 : 0 BRK 截止 1 BRK 导通 写 : 非法
11	保留				
12	PSL	Bit	0~1	0	读 : 0 PSL 光耦 LED 导通 1 PSL 光耦 LED 截止 写 : 非法
13	PSH	Bit	0~1	0	读 : 0 PSH 光耦 LED 导通 1 PSH 光耦 LED 截止 写 : 非法
14~15	保留				

报文示例——读取——端口寄存器 Port					
发送报文	RTU 地址	功能码	寄存器地址	数据	CRC
	0x01	0x04	0x00 0x80	0x00 0x01	0x30 0x22
响应报文	RTU 地址	功能码	数据字节数	数据内容	CRC
	0x01	0x04	0x02	0x00 0x01	0x78 0xF0
说明	读取端口寄存器，根据返回数据（例中为 0x0001）可得输入端口状态。				

#### 4.4.33 端口上升标志寄存器 PortHiFlag

地址: 0x0081                   PLC 对应地址: 40130

说明: Port 端口上升沿标记

操作: ReadWORD/ WriteWORD, 无记忆

Bit	名称	类型	值域	缺省	描述
0	Step	Bit	0~1	0	读 : 0 Port. Step 无上升沿 1 Port. Step 有上升沿 写 : 0 无作用 1 清除
1	Dir	Bit	0~1	0	读 : 0 Port. Dir 无上升沿 1 Port. Dir 有上升沿 写 : 0 无作用 1 清除
2	Free	Bit	0~1	0	读 : 0 Port. Free 无上升沿 1 Port. Free 有上升沿 写 : 0 无作用 1 清除
3	Zero	Bit	0~1	0	读 : 0 Port. Zero 无上升沿 1 Port. Zero 有上升沿 写 : 0 无作用 1 清除
4~7	保留				
8	Ready	Bit	0~1	0	读 : 0 Port. Ready 无上升沿 1 Port. Ready 有上升沿 写 : 0 无作用 1 清除
9	INP	Bit	0~1	0	读 : 0 Port. INP 无上升沿 1 Port. INP 有上升沿 写 : 0 无作用 1 清除
10	BRK	Bit	0~1	0	读 : 0 Port. BRK 无上升沿 1 Port. BRK 有上升沿 写 : 0 无作用 1 清除
11	保留				
12	PSL	Bit	0~1	0	读 : 0 Port. PSL 无上升沿 1 Port. PSL 有上升沿 写 : 0 无作用 1 清除
13	PSH	Bit	0~1	0	读 : 0 Port. PSH 无上升沿 1 Port. PSH 有上升沿 写 : 0 无作用 1 清除
14~15	保留				

#### 4.4.34 端口下降标志寄存器 PortLoFlag

地址: 0x0082      PLC 对应地址: 40131

说明: Port 端口下降沿标记

操作: ReadWORD/ WriteWORD, 无记忆

Bit	名称	类型	值域	缺省	描述
0	Step	Bit	0~1	0	读 : 0 Port. Step 无下降沿 1 Port. Step 有下降沿 写 : 0 无作用 1 清除
1	Dir	Bit	0~1	0	读 : 0 Port. Dir 无下降沿 1 Port. Dir 有下降沿 写 : 0 无作用 1 清除
2	Free	Bit	0~1	0	读 : 0 Port. Free 无下降沿 1 Port. Free 有下降沿 写 : 0 无作用 1 清除
3	Zero	Bit	0~1	0	读 : 0 Port. Zero 无下降沿 1 Port. Zero 有下降沿 写 : 0 无作用 1 清除
4~7	保留				
8	Ready	Bit	0~1	0	读 : 0 Port. Ready 无下降沿 1 Port. Ready 有下降沿 写 : 0 无作用 1 清除
9	INP	Bit	0~1	0	读 : 0 Port. INP 无下降沿 1 Port. INP 有下降沿 写 : 0 无作用 1 清除
10	BRK	Bit	0~1	0	读 : 0 Port. BRK 无下降沿 1 Port. BRK 有下降沿 写 : 0 无作用 1 清除
11	保留				
12	PSL	Bit	0~1	0	读 : 0 Port. PSL 无下降沿 1 Port. PSL 有下降沿 写 : 0 无作用 1 清除
13	PSH	Bit	0~1	0	读 : 0 Port. PSH 无下降沿 1 Port. PSH 有下降沿 写 : 0 无作用 1 清除
14~15	保留				

#### 4.4.35 端口翻转标志寄存器 PortFlipFlag

地址: 0x0083

PLC 对应地址: 40132

说明: Port 端口信号翻转标记

操作: ReadWORD/ WriteWORD, 无记忆

Bit	名称	类型	值域	缺省	描述
0	Step	Bit	0~1	0	读 : 0 Port. Step 无翻转 1 Port. Step 有翻转 写 : 0 无作用 1 清除
1	Dir	Bit	0~1	0	读 : 0 Port. Dir 无翻转 1 Port. Dir 有翻转 写 : 0 无作用 1 清除
2	Free	Bit	0~1	0	读 : 0 Port. Free 无翻转 1 Port. Free 有翻转 写 : 0 无作用 1 清除
3	Zero	Bit	0~1	0	读 : 0 Port. Zero 无翻转 1 Port. Zero 有翻转 写 : 0 无作用 1 清除
4~7	保留				
8	Ready	Bit	0~1	0	读 : 0 Port. Ready 无翻转 1 Port. Ready 有翻转 写 : 0 无作用 1 清除
9	INP	Bit	0~1	0	读 : 0 Port. INP 无翻转 1 Port. INP 有翻转 写 : 0 无作用 1 清除
10	BRK	Bit	0~1	0	读 : 0 Port. BRK 无翻转 1 Port. BRK 有翻转 写 : 0 无作用 1 清除
11	保留				
12	PSL	Bit	0~1	0	读 : 0 Port. PSL 无翻转 1 Port. PSL 有翻转 写 : 0 无作用 1 清除
13	PSH	Bit	0~1	0	读 : 0 Port. PSH 无翻转 1 Port. PSH 有翻转 写 : 0 无作用 1 清除
14~15					

#### 4.4.36 端口配置寄存器 PortConfig

地址: 0x0084~0x0087

PLC 对应地址: 40133~40136

说明: Port 端口配置寄存器, 用于配置各端口的基本属性 (高有效, 低有效)

操作: ReadQWORD/WriteQWORD, 有记忆

0x0085 Bit	0x0084 Bit	名称	类型	值域	缺省	描述
0	0	Step	Bit	00~11	00	读写: 00 原始信号 10 原始信号取反 01 强制为 1 11 强制为 0
1	1	Dir	Bit	00~11	00	读写: 00 原始信号 10 原始信号取反 01 强制为 1 11 强制为 0
2	2	Free	Bit	00~11	00	读写: 00 光耦断开, 使能; 光耦导通, 脱机 10 光耦断开, 脱机; 光耦导通, 使能 01 强制使能 11 强制脱机
3	3	Zero	Bit	00~11	00	读写: 00 光耦断开, 非零点; 光耦导通, 到零点 10 光耦断开, 到零点; 光耦导通, 非零点 01 强制不在零点 11 强制在零点
4	4	PA	Bit	00~11	00	读写: 00 原始信号 10 原始信号取反 01 强制为 1 11 强制为 0
5	5	PB	Bit	00~11	00	读写: 00 原始信号 10 原始信号取反 01 强制为 1 11 强制为 0
6	6	PZ	Bit	00~11	00	读写: 00 原始信号 10 原始信号取反 01 强制为 1 11 强制为 0
7	7	保留				
8	8	ERR	Bit	00~11	00	读写: 00 正常, 光耦断开; 报警, 光耦导通 10 报警, 光耦断开; 正常, 光耦导通 01 强制光耦导通 11 强制光耦断开
9	9	INP	Bit	00~11	00	读写: 00 没到位, 光耦断开; 到位, 光耦导通 10 到位, 光耦断开; 没到位, 光耦导通 01 强制光耦导通 11 强制光耦断开

10	10	BRK	Bit	00~11	00	读写: 00 不制动, 光耦断开; 制动, 光耦导通 10 制动, 光耦断开; 不制动, 光耦导通 01 强制光耦导通 11 强制光耦断开
11	11	保留				
12	12	PSL	Bit	00~11	00	读写: 00 光耦断开, 没到位; 光耦导通, 到位 10 光耦断开, 到位; 光耦导通, 没到位 01 强制光耦导通, 恒到位 11 强制光耦断开, 恒不到位
13	13	PSH	Bit	00~11	00	读写: 00 光耦断开, 没到位; 光耦导通, 到位 10 光耦断开, 到位; 光耦导通, 没到位 01 强制光耦导通, 恒到位 11 强制光耦断开, 恒不到位
14~15	14~15	保留				

本寄存器数据格式为 **UINT64**, 读写操作时必须对整个寄存器一起操作。0x0086 与 0x0087 为预留字, 目前全填零即可。

如果想对某端口的高低有效性进行配置, 需对本寄存器中 0x0085 的第 n 位与 0x0084 的第 n 位同时赋值, 组成 00、01、10、11 四种形式之一, 各形式对各端口的含义请参考上表。

报文示例——写入——端口配置寄存器 PortConfig										
发送报文	RTU 地址	功能码	寄存器地址	寄存器字数	寄存器字节数	数据	CRC			
	0x01	0x10	0x00 0x84	0x00 0x04	0x08	0x00 0x00 0x00 0x00 0x10 0x00 0x00 0x00	0x41 0x0B			
响应报文	RTU 地址	功能码	寄存器地址	寄存器字数			CRC			
	0x01	0x10	0x00 0x84	0x00 0x04			0x81 0xE3			
说明	向寄存器 0x0084 写入新的端口配置参数, 将 PSL 端口 (Bit12) 设置为 10, 即光耦断开表示到位。									
报文示例——读取——端口配置寄存器 PortConfig										
发送报文	RTU 地址	功能码	寄存器地址	寄存器字数			CRC			
	0x01	0x03	0x00 0x84	0x00 0x04			0x04 0x20			
响应报文	RTU 地址	功能码	寄存器字节数	数据			CRC			
	0x01	0x03	0x08	0x00 0x00 0x00 0x00 0x10 0x00 0x00 0x00			0x91 0x17			
说明	读取端口配置寄存器。									

#### 4.4.37 输入带宽寄存器 InputBand

地址: 0x0090~0x0091      PLC 对应地址: 40145~40146

说明: 可接收的脉冲信号的最大频率

操作: ReadDWORD/ WriteDWORD, 有记忆

Bit	名称	类型	值域	缺省	描述
0~31	InputBand	UINT32	20k~5M	记忆值	读/写: 输入带宽 单位: 赫兹, Hz

输入带宽越高, 驱动器摄入干扰的几率也增大, 所以, 驱动器的输入带宽设为脉冲信号最大频率的 2~5 倍为宜。

报文示例——写入——输入带宽寄存器 InputBand										
发送报文	RTU 地址	功能码	寄存器地址	寄存器字数	寄存器字节数	数据	CRC			
	0x01	0x10	0x00 0x90	0x00 0x02	0x04	0x00 0x00 0x4E 0x20	0XCE 0xBB			
响应报文	RTU 地址	功能码	寄存器地址	寄存器字数			CRC			
	0x01	0x10	0x00 0x90	0x00 0x02			0X41 0xE5			
说明	向寄存器 0x0090 写入输入带宽参数。									
报文示例——读取——输入带宽寄存器 InputBand										
发送报文	RTU 地址	功能码	寄存器地址	寄存器字数			CRC			
	0x01	0x03	0x00 0x90	0x00 0x02			0XC4 0x26			
响应报文	RTU 地址	功能码	寄存器字节数	数据			CRC			
	0x01	0x03	0x04	0x00 0x00 0x4E 0x20			0xCE 0x4B			
说明	读取输入带宽寄存器, 将返回数据(例中为 0x0000 0x4E20)换算得当前输入带宽。									

#### 4.4.38 刹车电压寄存器 VoltageBreak

地址: 0x001E      PLC 对应地址: 40031

说明: 当驱动器外接刹车(制动)电阻时, 需设置此参数(伺服步进功能用)

操作: ReadDWORD/ WriteDWORD, 有记忆

Bit	名称	类型	值域	缺省	描述
0~31	VoltageBreak	UINT16	120~800	记忆值	读/写: 刹车电压 单位: 伏特, V

**提示** 此寄存器为伺服功能专用。

该寄存器值域与驱动器型号相关

建议设定为开关电源电压的 1.1 倍, 需要刹车电压配合。

报文示例——写入——刹车电压寄存器 VoltageBreak					
	RTU 地址	功能码	寄存器地址	数据	CRC
发送报文	0x01	0x06	0x00 0x1E	0x00 0x1B	0XA9 0xC7
响应报文	0x01	0x06	0x00 0x1E	0x00 0x1B	0XA9 0xC7
说明	向寄存器 0x001E 写入刹车电压参数。				
报文示例——读取——刹车电压寄存器 VoltageBreak					
	RTU 地址	功能码	寄存器地址	数据	CRC
发送报文	0x01	0x04	0x00 0x1E	0x00 0x01	0x51 0xCC
响应报文	RTU 地址	功能码	数据字节数	数据内容	CRC
	0x01	0x04	0x02	0x00 0x1B	0xF9 0x3B
说明	读取刹车电压寄存器，返回数据（例中为 0x001B）换算得刹车电压值。				

#### 4.4.39 电机序列号 0 寄存器 MotorSN0

地址: 0x2000~0x2003

PLC 对应地址: 48193~48196

说明: 伺服步进电机序列号组 0

操作: ReadQWORD/ WriteQWORD, 有记忆

Bit	名称	类型	值域	缺省	描述
0~63	MotorSN0	UINT64	0x0000 0000 0000 0000 ~ 0xFFFF FFFF FFFF FFFF	记忆值	读/写: 电机序列号 0

(提示) 此寄存器为伺服功能专用。

伺服步进电机序列号由电机序列号 0, 电机序列号 1, 电机序列号 2, 电机序列号 3 组成。

伺服步进功能需要在电机序列号无误的情况下, 才能正常工作。

报文示例——写入——电机序列号 0 寄存器 MotorSN0										
	RTU 地址	功能码	寄存器地址	寄存器字数	寄存器字节数	数据	CRC			
发送报文	0x01	0x10	0x20 0x00	0x00 0x04	0x08	0x44 0x55 0x22 0x33 0xEF 0x11 0xAB 0xCD	0x11 0xE2			
响应报文	RTU 地址	功能码	寄存器地址	寄存器字数			CRC			
	0x01	0x10	0x20 0x00	0x00 0x04			0xCA 0x0A			
说明	向寄存器 0x2000 写入电机序列号第 1 段 (此例为 0xABCD 0xEF11 0x2233 0x4455)。									
报文示例——读取——电机序列号 0 寄存器 MotorSN0										
	RTU 地址	功能码	寄存器地址	寄存器字数			CRC			
发送报文	0x01	0x03	0x20 0x00	0x00 0x04			0x4F 0xC9			
响应报文	RTU 地址	功能码	寄存器字节数	数据			CRC			
	0x01	0x03	0x08	0x44 0x55 0x22 0x33 0xEF 0x11 0xAB 0xCD			0x4D 0xE3			
说明	读取电机序列号 0 寄存器。									

#### 4.4.40 电机序列号 1 寄存器 MotorSN1

地址: 0x2004~0x2007

PLC 对应地址: 48197~48200

说明: 伺服步进电机序列号组 1

操作: ReadQWORD/ WriteQWORD, 有记忆

Bit	名称	类型	值域	缺省	描述
0~63	MotorSN1	UINT64	0x0000 0000 0000 0000 ~ 0xFFFF FFFF FFFF FFFF	记忆值	读/写: 电机序列号 1

提示 此寄存器为伺服功能专用。

#### 4.4.41 电机序列号 2 寄存器 MotorSN2

地址: 0x2008~0x200B

PLC 对应地址: 48201~48204

说明: 伺服步进电机序列号组 2

操作: ReadQWORD/ WriteQWORD, 有记忆

Bit	名称	类型	值域	缺省	描述
0~63	MotorSN2	UINT64	0x0000 0000 0000 0000 ~ 0xFFFF FFFF FFFF FFFF	记忆值	读/写: 电机序列号 2

提示 此寄存器为伺服功能专用。

#### 4.4.42 电机序列号 3 寄存器 MotorSN3

地址: 0x200C~0x200F

PLC 对应地址: 48205~48208

说明: 伺服步进电机序列号组 3

操作: ReadQWORD/ WriteQWORD, 有记忆

Bit	名称	类型	值域	缺省	描述
0~63	MotorSN3	UINT64	0x0000 0000 0000 0000 ~ 0xFFFF FFFF FFFF FFFF	记忆值	读/写: 电机序列号 3

提示 此寄存器为伺服功能专用。

#### 4.4.43 电机型号寄存器 MotorName

地址: 0x2100~0x2103      PLC 对应地址: 48449~48452

说明: 配套的伺服步进电机型号, ASCII 编码

操作: ReadQWORD/ WriteQWORD, 有记忆

Bit	名称	类型	值域	缺省	描述
0~63	MotorName	String	0x0000 0000 0000 0000 ~ 0xFFFF FFFF FFFF FFFF	记忆值	读/写: 电机型号

**提示** 此寄存器为伺服功能专用。

报文示例——写入——电机型号寄存器 MotorName										
发送报文	RTU 地址	功能码	寄存器地址	寄存器字数	寄存器字节数	数据	CRC			
	0x01	0x10	0x21 0x00	0x00 0x04	0x08	0x6E 0x6F 0x69 0x74 0x6F 0x4D 0x65 0x67	0x8D 0xB2			
响应报文	RTU 地址	功能码	寄存器地址	寄存器字数			CRC			
	0x01	0x10	0x21 0x00	0x00 0x04			0xCB 0xF6			
说明	向寄存器 0x2100 写入配对电机型号 (此例为 “geMotion”)。									
报文示例——读取——电机型号寄存器 MotorName										
发送报文	RTU 地址	功能码	寄存器地址	寄存器字数			CRC			
	0x01	0x03	0x21 0x00	0x00 0x04			0x4E 0x35			
响应报文	RTU 地址	功能码	寄存器字节数	数据			CRC			
	0x01	0x03	0x08	0x6E 0x6F 0x69 0x74 0x6F 0x4D 0x65 0x67			0xD3 0xFE			
说明	读取配对电机的型号。									

#### 4.4.44 电机生产日期寄存器 MotorDate

地址: 0x2104~0x2105      PLC 对应地址: 48453~48454

说明: 配套的伺服步进电机生产日期

操作: ReadQWORD/ WriteQWORD, 有记忆

Bit	名称	类型	值域	缺省	描述
0~31	MotorDate	UNIT32	0x0000 0000 ~ 0xFFFF FFFF	记忆值	读/写: 电机生产日期

**提示** 此寄存器为伺服功能专用。

报文示例——写入——电机生产日期寄存器 MotorDate										
发送报文	RTU地址	功能码	寄存器地址	寄存器字数	寄存器字节数	数据	CRC			
	0x01	0x10	0x21 0x04	0x00 0x02	0x04	0x01 0x31 0x54 0xDB	0X48 0xA5			
响应报文	RTU地址	功能码	寄存器地址	寄存器字数			CRC			
	0x01	0x10	0x21 0x04	0x00 0x02			0X0A 0x35			
说明	向寄存器 0x2104 写入电机生产日期（此例为 20010203）。									
报文示例——读取——电机生产日期寄存器 MotorDate										
发送报文	RTU地址	功能码	寄存器地址	寄存器字数			CRC			
	0x01	0x03	0x21 0x04	0x00 0x02			0X8F 0xF6			
响应报文	RTU地址	功能码	寄存器字节数	数据			CRC			
	0x01	0x03	0x04	0x01 0x32 0x54 0xDB			0x24 0x9B			
说明	读取电机生产日期寄存器，将返回数据（例中为 0x0132 0x54DB）换算得电机生产日期。									

#### 4.4.45 电机编号寄存器 MotorNum

地址: 0x2106~0x2107

PLC 对应地址: 48455~48456

说明: 配套的伺服步进电机编号

操作: ReadQWORD/ WriteQWORD, 有记忆

Bit	名称	类型	值域	缺省	描述
0~31	MotorNum	UNIT32	0x0000 0000 ~ 0xFFFF FFFF	记忆值	读/写: 电机编号

提示 此寄存器为伺服功能专用。

#### 4.4.46 伺服功能寄存器 Servo

地址: 0x2108

PLC 对应地址: 48457

说明: 和伺服步进功能相关的寄存器

操作: ReadWORD, 无记忆

Bit	名称	类型	值域	缺省	描述
0	SerialNumSelfcheck	Bit	0~1	0	读 : 0 电机序列号自检错误 1 电机序列号自检正常 写 : 非法
1~15	保留				

提示 此寄存器为伺服功能专用。

电机序列号内部包好序列号的检验信息，该符号位用于验证序列号是否有个别字符错误。

#### 4.4.47 驱动器 CPU 温度寄存器 CpuTemp

地址: 0x0300      PLC 对应地址: 40769

说明: 驱动器 CPU 温度

操作: ReadWORD, 无记忆

Bit	名称	类型	值域	缺省	描述
0~15	CpuTemp	INT16	-40~125	0	读 : 驱动器 CPU 温度, 单位°C 写 : 非法

报文示例——读取——驱动器 CPU 温度寄存器 CpuTemp					
发送报文	RTU 地址	功能码	寄存器地址	数据	CRC
	0x01	0x04	0x03 0x00	0x00 0x01	0x31 0x8E
响应报文	RTU 地址	功能码	数据字节数	数据内容	CRC
	0x01	0x04	0x02	0x00 0x16	0x38 0xFE
说明	读取驱动器 CPU 温度寄存器, 根据返回数据(例中为 0x0016)得 CPU 温度。				

#### 4.4.48 驱动器散热器温度寄存器 SinkTemp

地址: 0x0301      PLC 对应地址: 40770

说明: 驱动器散热器温度

操作: ReadWORD, 无记忆

Bit	名称	类型	值域	缺省	描述
0~15	SinkTemp	INT16	-40~125	0	读 : 驱动器散热器温度, 单位°C 写 : 非法

(提示) 特定型号此寄存器有效。

报文示例——读取——驱动器散热器温度寄存器 SinkTemp					
发送报文	RTU 地址	功能码	寄存器地址	数据	CRC
	0x01	0x04	0x03 0x01	0x00 0x01	0x60 0x4E
响应报文	RTU 地址	功能码	数据字节数	数据内容	CRC
	0x01	0x04	0x02	0x00 0x26	0x38 0xEA
说明	读取驱动器散热器温度寄存器, 根据返回数据(例中为 0x0026)得散热器温度。				

#### 4.4.49 驱动器型号寄存器 DriverName

地址: 0x8000~0x8003

说明: 驱动器型号, 该寄存器出厂时固化, ASCII 编码。

操作: ReadQWORD, 无记忆

Bit	名称	类型	值域	缺省	描述
0~63	DriverName	String	0x0000 0000 0000 0000 ~ 0xFFFF FFFF FFFF FFFF	出厂值	读 : 驱动器名称 写 : 非法

##### 报文示例——读取——电机型号寄存器 MotorName

发送报文	RTU 地址	功能码	寄存器地址	寄存器字数	CRC
	0x01	0x03	0x80 0x00	0x00 0x04	0x6D 0xC9
响应报文	RTU 地址	功能码	寄存器字节数	数据	CRC
	0x01	0x03	0x08	0x6E 0x6F 0x69 0x74 0x6F 0x4D 0x65 0x67	0xD3 0xFE
说明	读取驱动器型号。				

#### 4.4.50 驱动器生产日期寄存器 DriverDate

地址: 0x8008~0x800B

说明: 驱动器生产日期, 该寄存器出厂时固化

操作: ReadQWORD, 无记忆

Bit	名称	类型	值域	缺省	描述
0~63	DriverDate	String	0x0000 0000 0000 0000 ~ 0xFFFF FFFF FFFF FFFF	出厂值	读 : 驱动器生产日期 写 : 非法

#### 4.4.51 驱动器编号寄存器 DriverNum

地址: 0x800C~0x800F

说明: 驱动器编号, 该寄存器出厂时固化

操作: ReadQWORD, 无记忆

Bit	名称	类型	值域	缺省	描述
0~63	DriverNum	String	0x0000 0000 0000 0000 ~ 0xFFFF FFFF FFFF FFFF	出厂值	读 : 驱动器编号 写 : 非法

## 5 应用举例

### 5.1 电脑、单片机通讯程序

#### 5.1.1 通讯测试

在进行通讯编程之前，首先要确认主机的通讯设置、RS485 接线等基本条件已经正确完成，同时验证基本的通讯代码是否能到达驱动器，所以要进行通讯测试。通讯测试方法如下：

步骤	说明	发送报文（HEX）
1, 检查接线	检查主机与驱动器之间的 RS485 信号线连接。	
2, 检查设置	检查主机通讯帧格式（8 个数据位，1 个偶校验位，一个停止位），波特率，驱动器总线地址。	
3, 发送代码	发送脱机代码，驱动器红色 LED 开始闪烁，电机无保持力矩。发送使能代码，红色 LED 恢复常亮，电机有保持力矩。	脱机 01 06 00 00 00 04 88 09 使能 01 06 00 00 00 00 89 CA

脱机使能测试成功，表明主机与驱动器之间通讯已无障碍，可以进行更深入的操作。

#### 5.1.2 初始化

对驱动器的初始化设置，可以利用电脑或手机版的 AgeMotion 完成，也可以在程序的初始化段落中完成。以下设定内容可根据实际需要选择。

设定内容	说明	发送报文（HEX）
电流 CurrentSet	根据不同的电机，设定峰值电流值。同时，可根据实际运行情况，稍稍调大或调小输出电流值。	下例中将电流设为 5A: 01 06 00 12 01 F4 29 D8
电流降流 CurrentLow	根据运行情况，设定静止时的电流降流百分比。	下例中将电流降流设为 30%: 01 06 00 13 00 1E F8 07
等待时间设定 CurrentLowWT	根据运行情况，设定降流等待时间。	下例中将降流等待时间设为 0.45s: 01 06 00 14 01 C2 49 CF
输入类型选择 InputType	如需上下限位开关的运行方式，需进行此操作。	下例中将输入设为上下限位方式: 01 06 00 08 00 08 09 CE
脉冲步进长度 PulseLength (细分值设定)	如果使用脉冲位置（电机脉冲实时位置、电机脉冲设定位置）来控制电机运行，需预先设定细分值。	下例中将细分设为 2500 脉冲/圈: 01 10 00 2A 00 02 04 00 00 06 00 72 68
速度滤波（通讯） VelFilterCom	设定通讯控制电机运行时的加减速	下例中将通讯速度滤波设为 21 级: 01 06 00 44 00 15 08 10

上表仅供示例，另有许多初始化设置，可根据需求选择。

### 5.1.3 向上下限位运行

总线驱动器可以设置为向上下限位开关运动，CN2[1..2]接上限位开关，CN2[3..4]接下限位开关。

设定内容	说明	发送报文 (HEX)
输入类型寄存器 InputType	将输入类型选择为向上下限位运动	向 InputType 寄存器输入 0x08: 01 06 00 08 00 08 09 CE
设定速度 VelSet	设定电机运行速度，例：设为 60rpm	向 VelSet 寄存器输入目标速度： 01 06 00 40 00 C0 88 4E
向下限位运动	运动至下限位开关 CN2[3..4]，到位后停止	01 06 00 00 00 10 88 06
向上限位运动	运动至上限位开关 CN2[1..2]，到位后停止	01 06 00 00 00 20 88 12
运动中停止	运动中途停止，把相应寄存器位清零即可	01 06 00 00 00 00 89 CA

### 5.1.4 向指定位置运行

设定内容	说明	发送报文 (HEX)
设定位置偏移至 0 OffsetToZero	将设定位置偏移至 0，即将位置信息清零，方便计算。	01 06 00 00 01 00 88 5A
设定速度 VelSet	运行前设定电机运行速度，可随时改变。 例：设为 60rpm	向 VelSet 寄存器输入目标速度： 01 06 00 40 00 C0 88 4E
电机脉冲设定位置 PulsePositionSet	电机脉冲当前位置为 0，细分为 2500 脉冲/圈时，向 PulsePosition 写入 10000，电机运行 4 圈。 电机脉冲当前位置为-10000，细分为 2500 脉冲/圈时，向 PulsePosition 写入 10000，电机运行 8 圈。	01 10 00 2E 00 02 04 00 00 27 10 6A 07

提示

电机运动主要有两组寄存器可以使用：

一组是 Position 与 PositionSet，单位为 MMS，1 圈=3840000MMS。

另一组是 PulsePosition 与 PulsePositionSet，单位为细分后 1 脉冲所代表的距离。

电机运动方向为 PositionSet-Position 或者 PulsePositionSet-PulsePosition。所得差值为正，电机正向运转；所得差值为负，电机反向运转。

## 5.2 PLC 程序示例

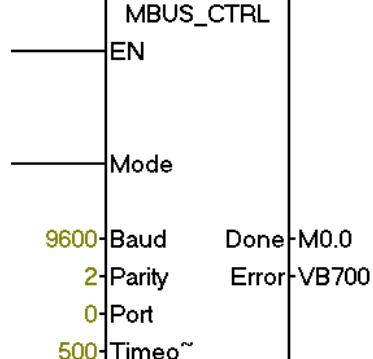
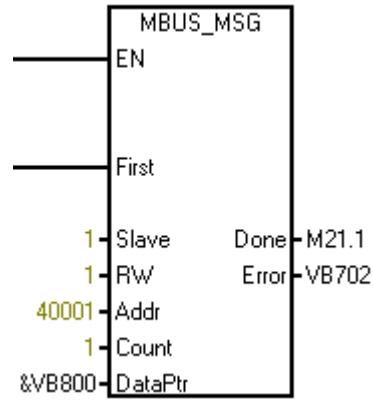
本公司支持 RS485 的驱动器产品采用标准的 Modbus-RTU 协议，可直接与西门子等各类工业场合广泛使用的 PLC 进行通讯和控制。

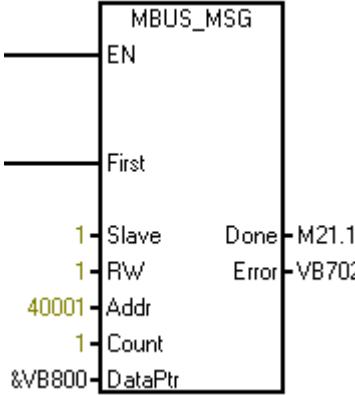
测试硬件：Simatic S7-200

测试软件：STEP 7-Micro/WIN SMART V02.01.00.01\_01.05

### 5.2.1 通讯测试

在进行通讯编程之前，首先要确认主机的通讯设置、RS485 接线等基本条件已经正确完成，同时验证基本的通讯代码是否能到达驱动器，所以要进行通讯测试。通讯测试方法如下：

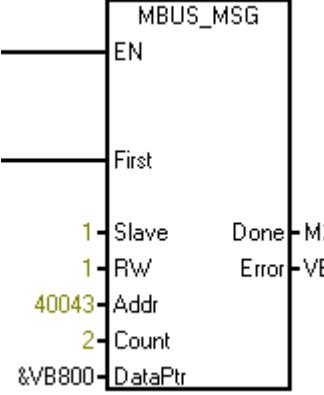
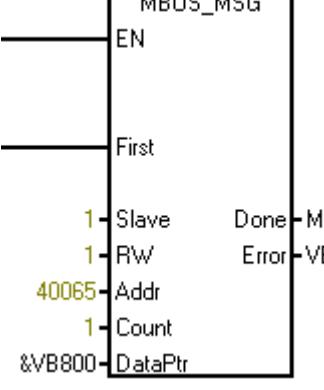
步骤	说明	梯形图
1, 检查接线	检查 PLC 与驱动器之间的 RS485 信号线连接。	
2, 检查设置	<p>检查主机通讯帧格式（8 个数据位，1 个偶校验位，一个停止位），波特率，驱动器总线地址。  <b>Baud</b> (波特率): 9600  <b>Parity</b> (奇偶校验): 2 (偶校验)  <b>Port</b> (端口号): 输入 PLC 上与驱动器连接的通讯端口号</p>	 <pre>         MBUS_CTRL         ---EN         ---Mode         9600---Baud         2---Parity         0---Port         500---Timeo~&gt;         Done---M0.0         Error---VB700     </pre>
3, 脱机指令	<p>发送脱机代码，驱动器红色 LED 开始闪烁，电机无保持力矩。</p>	 <pre>         MBUS_MSG         ---EN         ---First         1---Slave         1---RW         40001---Addr         1---Count         &amp;VB800---DataPtr         Done---M21.1         Error---VB702     </pre> <p style="text-align: center;">VB800 16#0 VB801 16#04 或者 VB800 16#0, 16#04</p>

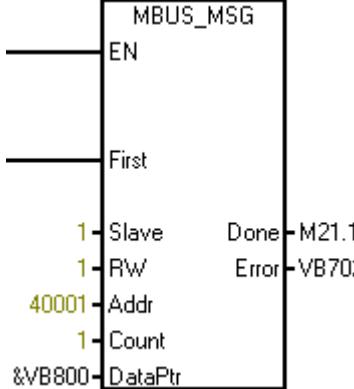
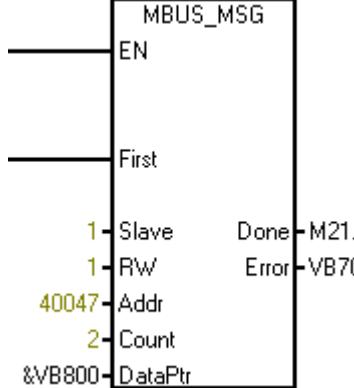
4, 使能指令	发送使能代码，红色 LED 恢复常亮，电机有保持力矩。 Slave (驱动器总线地址): 1 RW (读写): 1 (写入) Addr (访问寄存器地址): 40001 Count (写入字数): 1 DataPtr (待写入数据): VB800	 VB800 16#0, 16#0
---------	--	--

脱机使能测试成功，表明主机与驱动器之间通讯已无障碍，可以进行更进一步的操作。

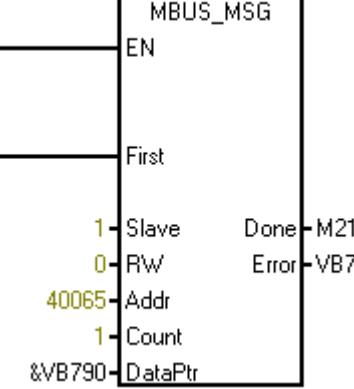
### 5.2.2 运行控制

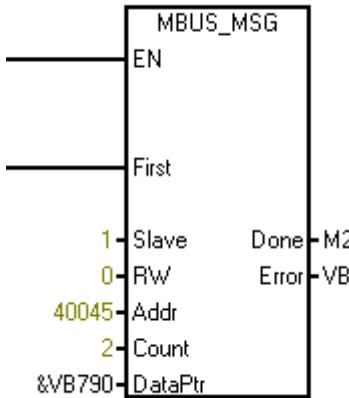
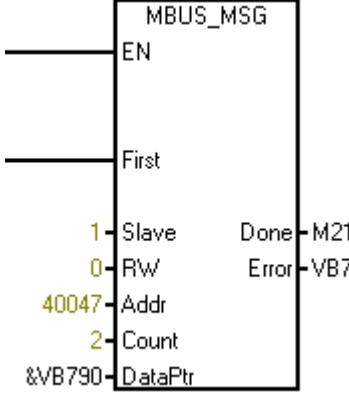
对驱动器的初始化设置，可以利用电脑或手机版的 AgeMotion 完成，也可以在程序的初始化段落中完成。以下进行简单的控制说明：

设定内容	说明	发送报文 (HEX)
设定细分	将细分设为 2500 脉冲/转。 Slave (驱动器总线地址): 1 RW (读写): 1 (写入) Addr (访问寄存器地址): 40043 Count (写入字数): 2 DataPtr (待写入数据): VB800	 VB800 16#0, 16#0, 16#06, 16#0
设定速度	设定转速为 60 转/分。 Slave (驱动器总线地址): 1 RW (读写): 1 (写入) Addr (访问寄存器地址): 40065 Count (写入字数): 1 DataPtr (待写入数据): VB800	 VB800 16#0, 16#C0

运动至下限位	<p>运动至下限位。</p> <p>Slave (驱动器总线地址): 1</p> <p>RW (读写): 1 (写入)</p> <p>Addr (访问寄存器地址): 40001</p> <p>Count (写入字数): 1</p> <p>DataPtr (待写入数据): VB800</p> <p>注: 需提前把输入类型设为上限+下限方式</p>	
脉冲设定位置 PulsePositionSet	<p>脉冲设定位置 10000 pulse, (0x2710)</p> <p>假设细分为 2500p/r:</p> <p>如果脉冲实时位置为 0,</p> <p>则运行 4 圈。</p> <p>如果脉冲实时位置为 5000 pulse,</p> <p>则运行 2 圈。</p> <p>行进距离=脉冲设定位置-脉冲实时位置</p>	

### 5.2.3 从驱动器读取数据

读取内容	说明	发送报文 (HEX)
读取 当前速度	<p>Slave (驱动器总线地址): 1</p> <p>RW (读写): 0 (读取)</p> <p>Addr (访问寄存器地址): 40065</p> <p>Count (读取字数): 1</p> <p>DataPtr (数据存入变量地址): VB790</p> <p>读取结果:</p> <p>VB790 16#0, 16#C0 0x00c0=192, 即 60 转/分</p>	

读取 脉冲实时位置	<p>Slave (驱动器总线地址): 1        RW (读写): 0 (读取)        Addr (访问寄存器地址): 40045        Count (读取字数): 2        DataPtr (数据存入变量地址): VB790</p> <p>读取结果:        VB790 16#0, 16#0, 16#27, 16#10        脉冲实时位置: 10000 pulse</p>	
读取 脉冲设定位置	<p>Slave (驱动器总线地址): 1        RW (读写): 0 (读取)        Addr (访问寄存器地址): 40047        Count (读取字数): 2        DataPtr (数据存入变量地址): VB790</p> <p>读取结果:        VB790 16#0, 16#0, 16#27, 16#10        脉冲设定位置: 10000 pulse</p>	

## 6

## 售后服务

## 6.1 服务条款

非常感谢您购买本公司的产品。本产品内部有唯一的产品序号、质保期及其它出厂信息。本产品出厂时贴有保修凭证，保修凭证上注有产品序号，注意保护保修凭证，请勿撕下、撕开、撕毁或污染。

## 售后服务条款：

## 1、本公司各产品质保期不同，具体质保期以合同为准。

质保期内的产品享受本公司提供的免费维修服务，但由以下原因引起故障或损坏的，本产品将不再享受免费维修服务并且本公司不承担任何责任：

- A) 未按数据手册正确使用；
- B) 超出数据手册参数规范；
- C) 未经允许擅自拆卸、改造或维修；
- D) 保修凭证丢失、撕开、破损、污染；
- E) 跌落、挤压、碰撞、浸水、暴晒、污染；
- F) 地震、火灾、雷击等不可抗拒力及由此引起的二次灾害。

## 2、未经允许擅自拆卸、改造或维修本公司产品，本公司一律不予维修；

## 3、质保期内由于本产品内部质量问题所造成的本产品故障或损坏、事故、设备损坏或人员伤亡，本公司将只对本产品提供免费维修或产品赔偿，产品赔偿以本产品价值为限，产品运费由本公司负担，产品保价费由用户负担；

## 4、本公司对超出质保期的产品质量、事故、设备损坏或人员伤亡不承担任何责任，并对本产品采取收费维修，运费及保价费由用户负担；

## 5、质保期内的返修产品，如经本公司检测后确认无故障的，运费及保价费由用户承担；

## 6、如有任何技术问题，请与销售商或本公司联系，本手册内容如有改动，恕不另行通知，最新版本请到本公司网站下载或与我们联系；

## 7、凡购买本产品的用户一律视为已同意本数据手册中所提及的全部条款。

## 6.2 公司简介

杭州时代自动化有限公司专业从事数控软件、数控系统、数控面板及驱动系统的开发、生产和销售，提供数控系统、步进电机及驱动、电加工高频电源等产品。

聚迅®为本公司注册商标。

## 6.3 产品列表

数控软件	<b>AgeMotion</b>
	专用数控软件，支持单轴及多轴的机床运动控制、多机种联动控制。
	<b>AgeCNet</b>
	电缆网自动检测软件，主要应用于大型电缆网及批量生产电缆束的自动检测。
数控系统	<b>AgeMTSn</b>
	专用密码授权软件，对 AgeMotion 数控产品进行多种密码授权管理。
	<b>ANC515X</b>
	针对激光雕刻切割等工艺设计的小型数控系统。
数控面板	<b>ANC525X</b>
	针对大型电缆网和电缆束批量生产的组网型电缆网自动检测仪，可实现对目标电缆网的导通、绝缘及耐压的完全检测。
	<b>ANP4147</b>
	全金属外壳手持操作器，能承受高强度生产操作，支持 40 米通讯距离。
驱动系统	<b>ANP4149</b>
	精密制造全铝手持面板，全密封防污，带点阵显示屏，支持 40 米通讯距离。
	<b>AHD80XX/AHD81XX</b>
	低压步进电机驱动器，支持动态细分调整(AHD80XX)，支持串口。
	<b>AHD82XX</b>
	低压迷你型步进电机驱动器，支持串口。
	<b>AHD83XX</b>
	增强型低压步进电机驱动器，支持 USB 和 Modbus 总线通讯控制。
	<b>AHD84XX</b>
	低压总线步进电机驱动器，支持 Modbus 总线通讯控制。
	<b>AHD86XX</b>
	高压步进电机驱动器，支持 USB 通讯控制。
线切割附件	<b>AQD84XX</b>
	低压四轴步进电机驱动器，支持 USB 和 Modbus 总线通讯控制，支持脉冲手轮接口。
	<b>ASD90XX</b>
	低压闭环步进电机驱动器，支持 USB 和 Modbus 总线通讯控制。
	<b>ASD91XX/ASD92XX</b>
线切割附件	低压伺服步进电机驱动器，支持 USB 和 Modbus 总线通讯控制。
	<b>ADP0106XX</b>
	线切割高频电源，最小脉宽 300ns。
	<b>ALD1012XX</b>
	线切割低丝损模块。
<b>ANP4107XX</b>	
	线切割高频控制面板

## 6.4 联系我们

公司名称：杭州时代自动化有限公司

公司地址：杭州市余杭区五常街道荆长路 81 号迪兴西溪双创园 B 框 902 室

邮 编：310023

电 话：0571-85022190

传 真：0571-85022590

网 址：<http://www.AgeMotion.com/>

Email : Sales@AgeMotion.com (建议使用公司具体联系人 Email)