

VSMD 系列协议篇

CAN2.0 协议



Vince
北京伟恩斯技术有限公司

目录

目录.....	1
1. 关于 CAN2.0 协议.....	1
1.1 概述.....	1
1.2 CAN2.0 通信帧.....	2
1.2.1 概述	2
1.2.2 标准帧定义.....	2
1.2.3 扩展帧定义.....	3
1.3 TARGET ID	3
1.4 控制字.....	3
1.5 指令/寄存器地址.....	4
1.6 数据.....	4
2. 驱动器状态.....	5
2.1 概述.....	5
2.2 当前运行速度.....	5
2.3 当前绝对位置.....	5
2.4 驱动器状态	6
2.5 驱动器固件版本信息.....	7
3. 驱动器控制指令.....	8
3.1 概述.....	8
3.2 操作指令清单.....	9
3.3 电机运行控制.....	10
3.3.1 电机运行模式.....	10
3.3.2 电机使能 (ena)	11
3.3.3 电机失能 (off)	11
3.3.4 设置当前位置值 (org)	11
3.3.5 速度模式运行 (mov)	12
3.3.6 绝对位置运行 (pos)	12
3.3.7 相对位置运行 (rmv)	13
3.3.8 停止运行 (stp)	13
3.4 传感器端口输出控制 (output)	14
3.5 归零控制.....	15
3.5.1 概述	15
3.5.2 启动归零运行 (zero start)	15
3.5.3 停止归零运行 (zero stop)	15
3.6 外部设备控制.....	16
3.6.1 概述	16
3.6.2 输出端口 1~3 (sc1~sc3) 控制 (sv)	16
3.6.3 输出端口 1~3 (sc1~sc3) 控制 (nmos)	17
3.7 振动控制 (shake)	18
3.8 保存参数至 flash (sav)	19
3.9 清除编码器错误标志 (enc_err_clr)	19
3.10 自动读取状态信息 (auto_res_sts)	19

3.11 读取状态寄存器 (read_sts_regs)	20
3.12 读取数据寄存器 (read_data_regs)	20
4. 驱动器工作参数.....	21
4.1 概述.....	21
4.2 参数说明	23
4.2.1 设备 ID (cid)	23
4.2.2 波特率 (bdr)	23
4.2.3 细分 (mcs)	24
4.2.4 运行速度 (spd)	24
4.2.5 加速度 (acc)	25
4.2.6 减速度 (dec)	25
4.2.7 加速电流 (cra)	26
4.2.8 工作电流 (crn)	26
4.2.9 保持电流 (crh)	27
4.2.10 传感器 1/传感器 2 下降沿/上升沿触发事件 (s1f/s1r/s2f/s2r)	28
4.2.11 传感器 3/传感器 4 下降沿/上升沿触发事件 (s3f/s3r/s4f/s4r)	28
4.2.12 传感器 5/传感器 6 下降沿/上升沿触发事件 (s5f/s5r/s6f/s6r)	29
4.2.13 设置传感器 1~6 的上拉/下拉模式 (pud)	29
4.2.14 传感器 1~传感器 6 的工作模式 (s1/s2/s3/s4/s5/s6)	30
4.2.15 归零模式 (zmd)	30
4.2.16 原点传感器开放电平 (osv)	30
4.2.17 原点传感器端口 (snr)	31
4.2.18 归零速度 (zsd)	31
4.2.19 归零安全位置 (zsp)	31
4.2.20 无握手启动等待时间 (dar)	32
4.2.21 外接电位器阻值 (res)	32
4.2.22 正/负极限传感器 (msr/msv/psr/psv)	33
4.2.23 上电电机使能 (pae)	34
4.2.24 上电自动归零 (zar)	34
4.2.25 编码器归零电流 (zcr)	34
4.2.26 外设控制端口 1 的输出比率 (nmos1r)	34
4.2.27 外设控制端口 2 的输出比率 (nmos2r)	35
4.2.28 外设控制端口 3 的输出比率 (nmos3r)	35
4.2.29 编码器模式 (emod)	35
4.2.30 编码器线数 (elns)	35
4.2.31 电机单圈整步数 (estp)	36
4.2.32 堵转重试次数 (erty)	36
4.2.33 编码器方向 (edir)	36
4.2.34 编码器灵敏度 (ez)	36
4.2.35 编码器错误处理方式 (ewr)	37
4.2.36 电机减速比 (eratio)	37
4.2.37 堵转检测灵敏度 (esds)	37
4.2.38 软件负限位 (sml)	38
4.2.39 软件正限位 (spl)	38
4.3 数据寄存器写入指令.....	39
4.4 参数写入示例.....	40

4.4.1 示例 1	40
4.4.2 示例 2	41
4.4.3 示例 3	42
4.4.4 示例 4	43
5. 上位机软件及协议总表文件	44
5.1 上位机软件	44
5.2 协议总表文件	44
5.2.1 概述	44
5.2.2 扩展帧标识符生成	45
5.2.3 标准帧标识符生成	45
5.2.4 进制转换工具	45
5.2.5 协议总表使用示例 1	46
5.2.6 协议总表使用示例 2	47
5.2.7 协议总表使用示例 3	48
6. 联系我们	50
7. 附录 1 - 进制转换例程	51

1. 关于 CAN2.0 协议

1.1 概述

当今工业控制系统中，CAN 总线得到日益广泛的应用，和其他形式的总线相比，CAN 总线具有以下特点：

- ◆ 传输可靠，实时性高。
- ◆ 传输距离远（最远 10km），传输速率快（最高 1MHz bps）。
- ◆ 单条总线最多可接 110 个节点，并可方便的扩充节点数。
- ◆ 可使用普通的双绞线、同轴电缆或光纤等作为通信介质。
- ◆ 总线系统结构简单、性价比高。
- ◆ 多主结构、各节点的地位平等，方便区域组网，总线利用率高。
- ◆ 实时性高，采用非破坏总线仲裁技术，对优先级高的节点可做到通信无延时。
- ◆ 会自动关闭出错的节点并切断其与总线的联系，不影响总线上其他节点的通信。
- ◆ 报文为短帧结构并有硬件 CRC 校验，受干扰概率小，数据出错率极低。
- ◆ 自动检测报文发送成功与否，可由硬件自动重发，传输可靠性高。
- ◆ 通过硬件进行报文过滤，节点设备只接收对其必要的信息以减轻处理器负担，简化软件制作。

本公司 VSMD1X6 系列的步进电机驱动器支持 CAN2.0（包括 CAN2.0A 和 CAN2.0B）通信协议，用户在使用该系列驱动器时，无需具备步进电机驱动的底层知识，只要掌握正确发送指令的方法即可方便、精确的控制步进电机的运行。

本文针对支持 CAN2.0 协议的 VSMD 电机驱动器，对电机控制、参数设置、状态获取等方法进行说明。

1.2 CAN2.0 通信帧

1.2.1 概述

CAN 总线通信协议，目前有 CAN2.0A 和 CAN2.0B 两类得到广泛应用，CAN2.0A/B 是 CAN 标准的两个部分，支持 CAN2.0B 的 CAN 芯片，同时兼容 CAN2.0A 的通信。

CAN2.0A/B 两种通信方式的区别：

- ◆ CAN2.0A 使用标准帧通信。
- ◆ CAN2.0B 使用扩展帧通信

1.2.2 标准帧定义

VSMD 驱动器通信的标准帧中定义了 11 位标识符 (SID) + 8 字节数据，标识符部分的定义如下图所示：

10	9	8	7	6	5	4	3	2	1	0
SID										
TARGET ID				C1	C0	CMD / REG ADDR				
0	0	0	0	X	X	X	X	X	X	X

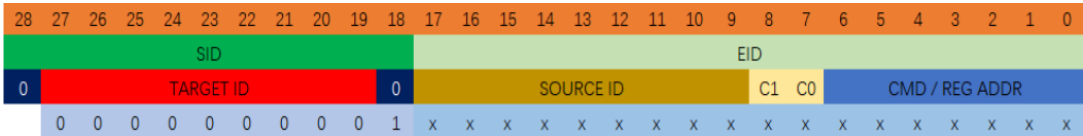
位名称	说明
TARGETID (BIT10:BIT7)	目标设备号
C1 C0 (BIT6:BIT5)	控制字
CMD / REG ADDR (BIT4:BIT0)	指令 / 寄存器地址

注意：

- ◆ 标准帧定义中使用标识符的后 5 位作为寄存器地址，因此寄存器最大寻址能力为 32。
- ◆ “TARGET ID” 设置为 “0000” 的指令为广播指令。

1.2.3 扩展帧定义

VSMD 驱动器通信的扩展帧中定义了 29 位扩展标识符 (SID) + 8 字节数据, 标识符部分的定义如下图所示:



位名称	说明
BIT28	固定 0
TARGET ID (BIT27:BIT19)	目标设备号
BIT18	非广播指令时固定 0
SOURCE ID (BIT17:BIT9)	源设备号
C1 C0 (BIT8:BIT7)	控制字
CMD / REG ADDR (BIT6:BIT0)	指令 / 寄存器地址

注意：

- ◆ 扩展帧定义使用标识符的后 7 位作为寄存器地址，因此寄存器最大寻址能力为 128。
- ◆ BIT18 置“1”的指令为广播指令。

1.3 TARGET ID

CAN 总线上可连接多个节点设备，每个节点设备都被赋予唯一的 ID 号作为 CAN 总线节点的标识符，这个 ID 号即“TARGET ID”，使用 TARGET ID 可以用于识别总线上的节点设备。

1.4 控制字

在通信帧标识符部分，通过 2bit 的控制字描述该通信帧传送数据的类型和方式。

C1	C0	说明
0	0	状态寄存器（读）：驱动器返回至上位机
0	1	数据寄存器（读）：驱动器返回至上位机
1	0	数据寄存器（写）：上位机发送至驱动器
1	1	操作指令：上位机发送至驱动器

1.5 指令/寄存器地址

在通信帧标识符中含有指令/寄存器地址（标准帧 5bit、扩展帧 7bit），其用途如下：

- ◆ 上位机向驱动器发送控制指令时设置控制指令码。指令码定义参照“3.驱动器控制指令”。
- ◆ 上位机向驱动器发送数据寄存器写入指令时，设置写入目标的数据寄存器地址。数据寄存器地址定义参照“4.驱动器工作参数”。

1.6 数据

CAN 通信帧中除标识符（SID）部分外，用户还需要对数据部分（最大 8 字节）进行设置。其用途如下：

- ◆ 上位机向驱动器设备发送控制指令时，对于带有参数的指令，在此设置参数。
- ◆ 上位机向驱动器发送数据寄存器中写入指令时，在此设置要写入的参数数据。

数据字节的设置，采取高位在前、低位在后的顺序，例如：当设置电机目标位置为 25000 时，其 16 进制数为：0x000061A8，因此数据字节依次设置为：0x00 0x00 0x61 0xA8。

此外，根据 CAN 协议的要求，数据部分的字节长度应设置在通信帧“DLC”码中。

2. 驱动器状态

2.1 概述

VSMD 驱动器在状态寄存器中保存电机当前运行速度、当前绝对位置、驱动器状态等实时信息以及驱动器使用的固件版本等信息，状态寄存器每个地址中的数据长度均为 4 字节。

VSMD 驱动器状态寄存器的地址分配及保存的数据内容如下表所示：

地址	名称	说明	数据类型
0x00	spd	电机当前运行速度	32 位浮点数
0x01	pos	电机当前绝对位置	32 位带符号整数
0x02	status	驱动器状态	32 位无符号整数
0x03~0x09	-	保留	-
0x0A~0x10	-	驱动器型号+固件版本号	字符型

上位机读取状态寄存器中数据的方法参照“3.11 读取状态寄存器（read_sts_regs）”的说明。

2.2 当前运行速度

VSMD 驱动器会将电机当前的运行速度持续更新保存在地址为 0x00 的状态寄存器中。此数据类型为 4 字节浮点数，上位机可通过读取状态寄存器中的该项数据获取电机当前运行速度的信息。

2.3 当前绝对位置

VSMD 驱动器会将电机运行中的位置信息持续更新保存在地址为 0x01 的状态寄存器中，此数据为电机当前的绝对位置，即相对于原点的位置。此数据类型为 4 字节有符号整数，上位机可通过读取状态寄存器中的该项数据获取电机当前位置的实时信息。

2.4 驱动器状态

VSMD 驱动器使用一个 32 位无符号整数保存当前驱动器及电机的工作状态信息，其中每 bit 对应一个特定的状态，VSMD 驱动器会将状态数据持续更新保存在地址为 0x02 的状态寄存器中。上位机可通过读取状态寄存器中的该项数据，对照各 bit 对应的状态含义了解设备当前的工作状态。

状态位各 bit 定义及状态值含义如下表所示：

状态位	名称	说明	值：0	值：1
0	s1	传感器 1 状态	低电平	高电平
1	s2	传感器 2 状态	低电平	高电平
2	s3	传感器 3 状态	低电平	高电平
3	s4	传感器 4 状态	低电平	高电平
4	-	保留	-	-
5	-	保留	-	-
6	flt	硬件错误	正常	错误
7	org	当前位置和原点关系	不在原点	在原点
8	stp	电机运行状态	运行中	停止中
9	cmd_wrg	指令错误标志	指令正确	指令错误
10	flash_err	存储器读写错误标志	正常	异常
11	action	离线运行状态	非离线运行	离线运行
12	hs	握手信号	无握手	有握手
13	pwr	电机使能/失能状态	电机失能	电机使能
14	zero	归零动作状态	无归零/归零中	归零结束
15	-	保留	-	-
16	s5	传感器 5 状态	低电平	高电平
17	s6	传感器 6 状态	低电平	高电平
18	-	保留	-	-
19	-	保留	-	-
20	-	保留	-	-
21	-	保留	-	-
22	-	保留	-	-
23	-	保留	-	-
24	enc_err	编码器错误标志 (※)	正常	编码器错误
25	-	保留	-	-
26	-	保留	-	-
27	act	特殊动作状态	无动作或运动中	运动完成
28	-	保留	-	-
29	-	保留	-	-
30	-	保留	-	-
31	-	保留	-	-

※ 只适用于 VSMD 闭环驱动器。

2.5 驱动器固件版本信息

驱动器的型号以及使用的固件版本号保存在地址为 0x0A~0x10 的状态驱动器中，保存的固件信息示例如下表所示：

地址	D1	D2	D3	D4
0x0A	V	S	M	D
0x0B	1	0	6	-
0x0C	0	2	5	T
0x0D	-	1	.	0
0x0E	.	0	4	0
0x0F	.	1	7	1
0x10	1	0	1	

注意：不同型号驱动器的固件版本信息存在差异，请参照具体型号驱动器的使用说明书了解。

3. 驱动器控制指令

3.1 概述

上位机可向驱动器发出操作指令以控制驱动器完成各项工作。

控制指令的通信帧中，需要进行如下设置：

目标设备 ID： 设置控制对象驱动器的设备 ID。

源设备 ID： 上位机发送指令时设置为“0”（扩展帧使用）。

控制字： C1=1、C0=1

指令码： VSMD 驱动器 CAN2.0 协议中，规定了各个指令码对应控制驱动器工作的各项指令。

指令参数： 指令分为无参数指令及有参数指令两种，使用有参数指令时，在通信帧数据部分中设置参数值。

指令的具体设置方法，可参照本节后续指令说明及“5.2 协议总表文件”中的示例。

注意：本章后续的指令说明中，均以目标设备 ID (cid) 为“1”为例，给出标准帧和扩展帧标识符 (SID) 的设置例（非广播指令）以供用户参考。

3.2 操作指令清单

VSMD 驱动器的操作指令一览表如下表所示，本章后续将对各个指令进行详细说明。

指令码	名称	说明
0x01	ena	电机使能
0x02	off	电机失能
0x03	org	设置当前位置
0x04	stp	停止运行
0x05	mov	速度模式运行
0x06	pos	绝对位置运行
0x07	sav	保存参数至 flash
0x08	output	传感器端口 1~6 输出控制
0x09	zero start	启动归零运行
0x0A	zero stop	停止归零运行
0x0B	rmv	相对位置运行
0x0C	保留	-
0x0D	sv	调节外设控制端口输出电压占空比（感型负载）
0x0E	nmos	调节外设控制端口输出电压占空比（非感型负载）
0x0F	auto_res_sts	自动返回状态
0x10	保留	-
0x11	保留	-
0x12	保留	-
0x13	保留	-
0x14	保留	-
0x15	保留	-
0x16	保留	-
0x17	保留	-
0x18	保留	-
0x19	保留	-
0x1A	保留	-
0x1B	shake	振动控制指令
0x1C	保留	-
0x1D	encoder	清除编码器错误标志
0x1E	read_sts_regs	读取状态寄存器
0x1F	read_data_regs	读取数据寄存器

注意：不同型号驱动器的指令可能存在差异，请参阅具体型号驱动器的使用说明书了解。

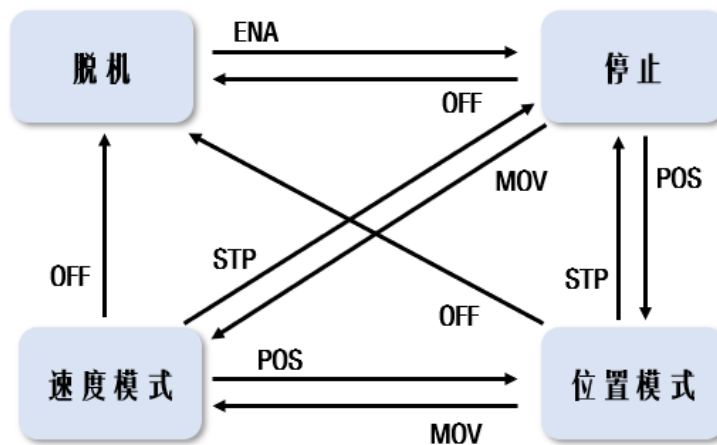
3.3 电机运行控制

3.3.1 电机运行模式

VSMD 驱动器可控制电机工作在四个运行模式中：

- ◆ 脱机：电机处于失能状态，不能运行。
- ◆ 停止：电机处于使能状态，但当前没有运行。
- ◆ 速度模式：指定电机的运行速度，控制电机运行。
- ◆ 位置模式：指定电机运行的目标位置，控制电机运行。

各模式间的状态迁移关系以及对应的控制指令如下图所示：



注意：

- 速度模式和位置模式的切换可以在得到指令后立即执行，不需要等待前一个指令执行结束。
- 同一模式内执行电机目标速度或者目标位置改变、停止的指令，以及执行速度模式与位置模式之间的切换指令时，只要是当前速度和目标速度不一致，或位置方向与速度反向时，都会自动启动加、减速过程，以避免出现电机急停或突然转向的情况，使电机平滑运转至目标速度或目标位置，在整个运动过程中，会根据当前的运行状况自动匹配相应的工作电流，以使扭矩、噪声、电机发热得到更好的控制。

3.3.2 电机使能 (ena)

指令说明：执行电机使能指令后，驱动器做如下操作：

- 状态位 pwr 置位。
- 运行模式会自动切换到停止状态。
- 电流会自动加载到保持电流 (crh) 设置的电流值。

指令码：0x01

指令参数：本指令无参数。

指令 SID：标准帧：0x00E1

扩展帧：0x00080181

3.3.3 电机失能 (off)

指令说明：执行电机失能指令后，控制器做如下操作：

- 状态位 pwr 复位。
- 运行状态会自动切换到脱机状态。
- 脱机状态电流设置无效。
- 原点初始化。
- 状态位 stp 置位。

指令码：0x02

指令参数：本指令无参数。

指令 SID：标准帧：0x00E2

扩展帧：0x00080182

3.3.4 设置当前位置值 (org)

指令说明：设置当前位置的绝对位置值。

指令码：0x03

指令参数：设置参数值及含义如下表所示：

No.	指令参数	必/选	类型	数据字节
1	当前位置的绝对位置值 单位：脉冲 设置范围：-2,147,483,647~2,147,483,648 无参数时，设置电机当前位置为原点，即当前位置为“0”。	可选	int32	D1~D4

指令 SID：标准帧：0x00E3

扩展帧：0x00080183

3.3.5 速度模式运行 (mov)

指令说明：当电机处于非脱机模式时，可在任意时间切换电机至速度模式，以指定的速度运行，速度单位为：脉冲/秒。

注 意：如当前运行速度与目标速度不一致或反向，驱动器会立即启动加减速处理，使电机平滑运转至目标速度。

指令码：0x05

指令参数：设置参数值及含义如下表所示：

No.	指令参数	必/选	类型	数据字节
1	电机运行速度 单位：脉冲/秒 设置范围：-192,000~192,000 无参数时，电机以预设的运行速度（参照“4.2.4 运行速度 (spd)”）运行。	可选	float	D1~D4

指令 SID：标准帧：0x00E5

扩展帧：0x00080185

3.3.6 绝对位置运行 (pos)

指令说明：当电机处于非脱机模式时，可在任意时间切换电机至位置模式运行至指定的绝对位置（相对原点的位置）后停止，位置单位为：脉冲。

注 意：当电机已处于位置模式并尚未运行至前一指令目标位置时，发此指令可指定新的目标位置，此时电机将从当前位置开始直接运行至新的目标位置，在此过程中自动进行电机的加减速处理以使位置移动过程更为平滑。

指令码：0x06

指令参数：本指令最多带有两个参数，参数说明见下表：

No.	指令参数	必/选	类型	数据字节
1	电机运行停止位置，使用脉冲数表示 单位：脉冲 设置范围：-2,147,483,647~2,147,483,648	必须	int32	D1~D4
2	电机运行速度 单位：脉冲/秒 设置范围：-192,000~192,000 无速度参数时，电机以预设的运行速度（参照“4.2.4 运行速度 (spd)”）运行。	可选	float	D5~D8

指令 SID：标准帧：0x00E6

扩展帧：0x00080186

3.3.7 相对位置运行 (rmv)

指令说明：当电机处于非脱机模式时，可在任意时间切换电机至位置模式运行至指定的相对位置（距离当前所处的位置）后停止，位置单位为：脉冲。

注 意：指定相对位置时，遵循以下原则：

- 参数为负数时，表示向负方向移动指定的脉冲数。
- 参数为正数时，表示向正方向移动指定的脉冲数。
- 如果相对位置值+当前位置值超出本指令参数的取值范围，指令将不会被执行。
- 在需要连续进行相对位置运行的场合，可在适当时机使用“org”指令复位当前位置，即设置电机当前位置为“0”。

指令码：0x0B

指令参数：本指令最多带有两个参数，参数说明见下表：

No.	指令参数	必/选	类型	数据字节
1	目标位置距当前位置的距离 单位：脉冲 设置范围：-2,147,483,647~2,147,483,648	必须	int32	D1~D4
2	电机运行速度 单位：脉冲/秒 设置范围：-192,000~192,000 无速度参数时，电机以预设的运行速度（参照“4.2.4 运行速度 (spd)”）运行。	可选	float	D5~D8

指令 SID：标准帧：0x00EB

扩展帧：0x0008018B

3.3.8 停止运行 (stp)

指令说明：控制电机停止运行。

指令码：0x04

指令参数：设置参数值及含义如下表所示：

No.	指令参数	必/选	类型	数据字节
1	电机停止模式 0：减速停止 1：立即停止	必须	uint32	D1

指令 SID：标准帧：0x00E4

扩展帧：0x00080184

3.4 传感器端口输出控制 (output)

指令说明： 传感器端口可作为输出使用，上位机可通过端口输出控制指令，控制与 TTL 兼容的传感器端口 1/2/3/4/5/6 输出相应的 TTL 电平信号，此方法用于驱动器向外部设备输出一个信号，但由于驱动器端口输出电流较小，一般不能用于直接控制外部设备。

注 意： 使用本指令时需注意以下几点：

- 执行本指令前，需要把对应的传感器端口 1/2/3/4/5/6 的工作模式设置为“输出”(参照“4.2.14 传感器 1~传感器 6 的工作模式 (s1/s2/s3/s4/s5/s6)”), 否则将会发生指令错误。
- 输出信号是 3.3V 的 TTL 信号，如需大驱动电流可考虑添加外部驱动电路。
- 由于 VSMD1XX_010T 系列驱动器的传感器端口和其他系列驱动器不同，所以不同的端口控制指令适用范围存在差异。

指令码： 0x08

指令参数： 本指令最多带有两个参数，参数说明见下表：

No.	指令参数	必/选	类型	数据字节
1	传感器端口号：1~6 注意： • 对于 VSMD1XX_010T 系列驱动器传感器端口 3~6 设置无效 • 对于非 VSMD1XX_010T 系列驱动器传感器端口 1~2 设置无效	必须	uint32	D1
2	输出 TTL 电平值 0：低电平 1：高电平	必须	uint32	D2

指令 SID： 标准帧：0x00E8

扩展帧：0x00080188

3.5 归零控制

3.5.1 概述

归零 (HOMING) 功能是 VSMD 驱动器极具特色的一个功能, 它可以在驱动器加电后, 或是主控机发出归零指令后控制电机自动完成一系列初始位复位功能, 这样可以在很大程度上减轻主控机的负担, 也可以帮助对归零不很熟悉的用户将归零功能快速应用到自己的系统中。关于归零动作的具体说明, 请参照本公司另行制作的说明文档, 并参照本文后续参数说明部分了解和归零动作相关的参数设置。

3.5.2 启动归零运行 (zero start)

指令说明: 控制启动归零动作。

指令码: 0x09

指令参数: 本指令无参数。

注 意: 执行本指令前, 需要对驱动器归零相关参数进行设置, 参照 “4.驱动器工作参数” 中归零相关参数的说明。

指令 SID: 标准帧: 0x00E9

扩展帧: 0x00080189

3.5.3 停止归零运行 (zero stop)

指令说明: 控制停止归零动作。

指令码: 0x0A

指令参数: 本指令无参数。

指令 SID: 标准帧: 0x00EA

扩展帧: 0x0008018A

3.6 外部设备控制

3.6.1 概述

VSMD 驱动器可通过外部设备控制端口电压输出的方式，对外接设备，例如电磁阀、刹车、RGB 灯等设备进行控制。驱动器具有 3 个外部设备连接端口，输出电压范围为 12~24V，驱动器通过调节各端口输出电压占空比的方式对外接设备进行控制。

受控外部设备可分为两类：

- ◆ 感性负载，例如：电磁阀、刹车等。
- ◆ 非感性负载，例如：RGB 灯。

用户可以根据控制对象的特点，通过参数调整输出电压占空比以取得最佳的控制效果。

注意：外部设备控制指令只适用于有外部设备控制信号输出型号的 VSMD 驱动器。

3.6.2 输出端口 1~3 (sc1~sc3) 控制 (sv)

指令说明：控制驱动器外部设备控制端口 1~3 的输出电压占空比，一般用于驱动感性负载，例如：电磁阀、刹车等。

指令码：0x0D

指令参数：本指令带有两个参数，参数说明见下表：

No.	指令参数	必/选	类型	数据字节
1	外部设备控制端口 0: sc1 1: sc2 2: sc3	必须	char	D1
2	控制信号占空比 数据范围：0x00~0xFF	必须	char	D2

指令 SID：标准帧：0x00ED

扩展帧：0x0008018D

3.6.3 输出端口 1~3 (sc1~sc3) 控制 (nmos)

指令说明：控制驱动器外部设备控制端口 1~3 的输出电压占空比，一般用于驱动非感性负载，例如：RGB 灯。

指令码：0x0E

指令参数：本指令带有两个参数，参数说明见下表：

No.	指令参数	必/选	类型	数据字节
1	外部设备控制端口 0: sc1 1: sc2 2: sc3	必须	char	D1
2	控制信号占空比 数据范围：0x00~0xFF	必须	char	D2

指令 SID：标准帧：0x00EE

扩展帧：0x0008018E

3.7 振动控制 (shake)

指令说明：振动功能是 VSMD 驱动器极具特色的一个功能，它允许用户只通过简单的设置，便可由驱动器控制电机自动以指定频率在电机指定的行程范围内进行往复运转（模拟振动过程）。振动分两种模式：

- 以次数模式启动振动：指定振动的次数，启动振动
- 以时间模式启动振动：指定振动持续的时间，启动振动

指令码：0x1B

指令参数：本指令最多带有两个参数，参数说明见下表：

No.	指令参数	必/选	类型	数据字节
1	振动控制及振动相关参数种类设置： 0x00：停止振动 0x01：以次数模式启动振动 0x02：以时间模式启动振动 0x10：振动行程开始的绝对位置 0x11：振动行程结束的绝对位置 0x12：振动频率	必须	char	D1
2	根据指定的第一个参数，设置相关参数值	可选	uint16	D2~D5

第二参数的设置内容及数据类型如下表所示：

数据设置种类	D1	D2	D3	D4	D5
停止振动运行	0x00	无	无	无	无
启动次数模式振动	0x01	振动次数，类型：int32 次数设置为“0”时振动无次数限制			
启动时间模式振动	0x02	振动时间，单位：秒，类型：int32 时间设置为“0”时振动无时间限制			
设置振动开始位置	0x10	振动开始位置，单位：脉冲，类型：int32			
设置振动结束位置	0x11	振动结束位置，单位：脉冲，类型：int32			
设置振动频率	0x12	振动频率，单位：Hz，类型 float 参数范围：0.1 ~ 20.0			

注 意：1. 控制振动时，首先需要分别发指令设置振动的起始位置、结束位置以及振动频率，之后发指令启动次数模式振动或时间模式振动。

2. 实际应用时应考虑实现的合理性设置适当的参数值。

指令 SID：标准帧：0x00FB

扩展帧：0x0008019B

3.8 保存参数至 flash (sav)

指令说明：将驱动器当前设置的参数保存至驱动器的 flash 中。当发生 flash 保存失败时，状态位“flash_err”会被置位。

指令码：0x07

指令参数：本指令无参数。

注意：由于执行本指令过程中会涉及 flash 写入，所以不适合在上位机程序中执行。建议使用电脑端的本公司上位机软件进行调试，配置好合适的参数后再使用本指令将参数保存至 flash 中，不建议频繁执行本指令。

指令 SID：标准帧：0x00E7

扩展帧：0x00080187

3.9 清除编码器错误标志 (enc_err_clr)

指令说明：当堵转解除或者编码器故障解除后，发送本指令清除编码器错误标志位。

注意：本指令只适用于使用编码器的闭环型号驱动器。

指令码：0x1D

指令参数：设置参数值及含义如下表所示：

No.	指令参数	必/选	类型	数据字节
1	0：清除编码器错误标记	必须	char	D1

指令 SID：标准帧：0x00FD

扩展帧：0x0008019D

3.10 自动读取状态信息 (auto_res_sts)

指令说明：设置定时自动读取状态寄存器的处理方式。

指令码：0x0F

指令参数：设置参数值及含义如下表所示：

No.	指令参数	必/选	类型	数据字节
1	读取状态寄存器方式 0：不自动读取 1：自动读取速度+位置+状态位信息 2：自动读取位置+状态位信息 3：自动读取位置信息	必须	char	D1
2	自动读取状态信息的时间间隔，单位毫秒。 当不设置本参数时，以系统当前设置的时间间隔读取。 (系统默认设置为 50 毫秒)	可选	uint16	D2~D3

指令 SID：标准帧：0x00EF

扩展帧：0x0008018F

3.11 读取状态寄存器 (read_sts_regs)

指令说明：读取驱动器状态寄存器中的内容。

指令码：0x1E

指令参数：设置参数值及含义如下表所示：

No.	指令参数	必/选	类型	数据字节
1	读取数据在状态寄存器中的起始地址	必须	uint8	D1
2	读取状态寄存器的数量（对应状态寄存器的地址数量）	必须	uint8	D2

本指令可要求一次返回地址连续的多个状态寄存器中的数据。例如：读取状态寄存器中当前速度（0x00）、当前位置（0x01）、状态位（0x02）时，可设置指令数据为：

D1=0x00（首地址），D2=0x03（读取寄存器数量）

指令 SID：标准帧：0x00FE

扩展帧：0x0008019E

3.12 读取数据寄存器 (read_data_regs)

指令说明：读取驱动器数据寄存器中设置的工作参数值。

指令码：0x1F

指令参数：设置参数值及含义如下表所示：

No.	指令参数	必/选	类型	数据字节
1	读取数据在数据寄存器中的起始地址	必须	uint8	D1
2	读取数据寄存器的数量（对应数据寄存器的地址数量）	必须	uint8	D2

本指令可要求一次返回地址连续的多个数据寄存器中的数据。

例如：读取数据寄存器中电机运行速度（0x03）、加速度（0x04）、减速度（0x05）、加速电流（0x06）这4个地址连续的参数时，可设置指令数据为：

D1=0x03（首地址），D2=0x04（读取寄存器数量）

指令 SID：标准帧：0x00FF

扩展帧：0x0008019F

4. 驱动器工作参数

4.1 概述

VSMD 驱动器在数据寄存器中保存驱动器工作所需的各种参数，上位机通过对数据寄存器寻址进行参数的读取/写入，数据寄存器中每个地址中保存的数据长度均为 4 字节。

数据寄存器的地址分配及寄存器保存的数据如下表所示，本节后续将对各参数进行详细说明。

地址	名称	说明	类型
0x00	cid	设备 ID 号	uint32
0x01	bdr	波特率	uint32
0x02	mcs	细分	uint32
0x03	spd	运行速度	float
0x04	acc	加速度	float
0x05	dec	减速度	float
0x06	cra	加速电流	float
0x07	crn	工作电流	float
0x08	crh	保持电流	float
0x09	s1f/s1r/s2f/s2r	传感器 1/传感器 2 下降沿/上升沿触发事件	uint32
0x0A	s3f/s3r/s4f/s4r	传感器 3/传感器 4 下降沿/上升沿触发事件	uint32
0x0B	s5f/s5r/s6f/s6r	传感器 5/传感器 6 下降沿/上升沿触发事件	uint32
0x0C	pud	传感器 1~传感器 6 的上拉/下拉模式	uint32
0x0D	s1/s2/s3/s4/s5/s6	传感器 1~传感器 6 的工作模式	uint32
0x0E	zmd	归零模式	uint32
0x0F	osv	原点传感器开放电平	uint32
0x10	snr	原点传感器端口	uint32
0x11	zsd	归零速度	float
0x12	zsp	归零安全位置	int32
0x13	-	保留	-
0x14	dar	无握手启动等待时间	uint32
0x15	res	外接电位器阻值	uint32
0x16	-	保留	-
0x17	msr/msv/psr/psv	msr (负极限传感器端口) msv (负极限传感器触发电平) psr (正极限传感器端口) psv (正极限传感器触发电平)	uint32
0x18	pae	上电电机使能	uint32
0x19	-	保留	-
0x1A	zar	上电自动归零	uint32
0x1B	-	保留	-
0x1C	zcr	编码器归零电流	float
0x1D	nmos1r	cs1 输出比率	float
0x1E	nmos2r	cs2 输出比率	float
0x1F	nmos3r	cs3 输出比率	float
0x20	emod	编码器模式	uint32
0x21	elns	编码器线数	uint32

地址	名称	说明	类型
0x22	estp	电机单圈整步数	uint32
0x23	erty	堵转重试次数	uint32
0x24	edir	编码器方向	uint32
0x25	ez	编码器灵敏度	uint32
0x26	ewr	编码器错误处理方式	uint32
0x27	eratio	电机减速比	float
0x28	esds	堵转检测灵敏度	float
0x29	-	保留	-
0x2A	sml	软件负限位	int32
0x2B	spl	软件正限位	int32

注意：

1. 不同型号驱动器的数据寄存器定义可能存在差异，请参照具体型号驱动器的使用说明书了解。
2. 由于 CAN 标准帧的寄存器最大寻址能力为 32，所以使用标准帧无法访问数据寄存器中地址大于 0x20 的数据。

4.2 参数说明

4.2.1 设备 ID (cid)

数据说明： 设置驱动器的设备 ID。设备 ID 号对应通信帧结构里面的 Target ID 字段，用于区分总线上不同的子设备。

注意： 修改设备 ID 后需要执行保存 (sav) 指令，并在驱动器重新上电后生效。

寄存器地址： 0x00

数据类型： uint32 (32 位无符号整数)

参数设置： 1~110

4.2.2 波特率 (bdr)

数据说明： 设置总线通信波特率，单位：bps，波特率可设置范围为 20K~1M，总线上所有设备均需设置相同的通信波特率。

注意： 修改波特率后需要执行保存 (sav) 指令，并在驱动器重新上电后生效。

寄存器地址： 0x01

参数类型： uint32 (32 位无符号整数)

参数设置： 下表显示几种常用的波特率及对应的设置参数值

设置参数值	对应通信波特率
0x1C2132	20K
0x168132	25K
0x12C132	30K
0xE1132	40K
0xB4132	50K
0x5A163	60K
0x78132	75K
0x4B153	80K
0x4B143	90K
0x5A132	100K
0x48132	125K
0x3C132	150K
0x2D132	200K
0x24132	250K
0x1E132	300K
0xF153	400K
0x12132	500K
0x9163	600K
0x8153	750K
0x6163	900K
0x9132	1000K

4.2.3 细分 (mcs)

数据说明： 电机驱动器可支持的细分种类和具体型号相关，需根据使用对象驱动器设置合适的参数值。

寄存器地址： 0x02

参数类型： uint32 (32 位无符号整数)

参数设置： 本参数可设值及其对应的细分见下表：

参数值	细分
0	整步
1	2 细分
2	4 细分
3	8 细分
4	16 细分
5	32 细分
6	64 细分
7	128 细分
8	256 细分

4.2.4 运行速度 (spd)

数据说明： 设置电机运行的目标速度，单位为：脉冲/秒，速度的正负代表运行方向。

注意：

1. 电机正在运行时更改速度参数，如当前运行速度与目标速度不一致或反向，则会立刻启动加减速以平滑运转至目标速度。
2. 位置模式运行时，驱动器会根据目标位置自行决定速度方向，并使用运行速度的绝对值运行。

寄存器地址： 0x03

数据类型： float (32 位浮点数)

参数设置： 参数设置范围：-192,000~192,000（正负表示电机运行方向），需将 10 进制数据转换成 16 进制后设置。转换示例程序如下：

```
bool motor_cfg_spd(int tid, float spd)
{
    uint8_t data[4];
    uint32_t val = *((uint32_t*)(&spd));

    data[0] = (val >> 24) & 0xff;
    data[1] = (val >> 16) & 0xff;
    data[2] = (val >> 8) & 0xff;
    data[3] = (val >> 0) & 0xff;

    return can_write_dat_regs(tid, CAN_DATA_SPD, data, 4);
}
```

4.2.5 加速度 (acc)

数据说明：设置电机加速运行过程中的加速度值，单位为：脉冲/秒²。

- 注意：
1. 在负载比较大、运行速度较高时，应适当减小加速度值，以平衡加速过程中的惯性力，否则容易出现堵转。
 2. 当设置加速度为“0”时，电机无加速过程，直接以目标速度运行。

寄存器地址：0x04

数据类型：float (32 位浮点数)

参数设置：参数设置范围：0~192,000,000，需将 10 进制数据转换成 16 进制后设置。转换示例程序如下：

```
bool motor_cfg_acc(int tid, float acc)
{
    uint8_t data[4];
    uint32_t val = *((uint32_t*)&acc);

    data[0] = (val >> 24) & 0xff;
    data[1] = (val >> 16) & 0xff;
    data[2] = (val >> 8) & 0xff;
    data[3] = (val >> 0) & 0xff;

    return can_write_dat_regs(tid, CAN_DATA_ACC, data, 4);
}
```

4.2.6 减速度 (dec)

数据说明：设置电机减速运行过程中的减速度值，单位为：脉冲/秒²。

- 注意：
- 在负载较大且速度较高时，应该适当减小减速度值，以平衡减速过程中的惯性力，使减速过程更加平滑，避免出现旋转惯量大时过冲的情况。

寄存器地址：0x05

数据类型：float (32 位浮点数)

参数设置：参数设置范围：0~192,000,000，需将 10 进制数据转换成 16 进制后设置。转换示例程序如下：

```
bool motor_cfg_dec(int tid, float dec)
{
    uint8_t data[4];
    uint32_t val = *((uint32_t*)&dec);

    data[0] = (val >> 24) & 0xff;
    data[1] = (val >> 16) & 0xff;
    data[2] = (val >> 8) & 0xff;
    data[3] = (val >> 0) & 0xff;

    return can_write_dat_regs(tid, CAN_DATA_DEC, data, 4);
}
```

4.2.7 加速电流 (cra)

数据说明：设置电机加速过程中的电流值，单位为：A。

注意：电机在加速过程中，电流会自动加载到设置的加速电流值，以维持稳定的加速运行过程，此参数一般需要根据实际负载情况进行设置。

寄存器地址：0x06

数据类型：float (32 位浮点数)

参数设置：参数设置范围下限为“0”，上限由驱动器型号决定，例如：VSMD136_025T 型，峰值电流为 2.5A，则参数值上限为“2.5”，需将 10 进制数据转换成 16 进制后设置。转换示例程序如下：

```
bool motor_cfg_cra(int tid, float cra)
{
    uint8_t data[4];
    uint32_t val = *((uint32_t*)&cra);

    data[0] = (val >> 24) & 0xff;
    data[1] = (val >> 16) & 0xff;
    data[2] = (val >> 8) & 0xff;
    data[3] = (val >> 0) & 0xff;

    return can_write_dat_regs(tid, CAN_DATA_CRA, data, 4);
}
```

4.2.8 工作电流 (crn)

数据说明：设置电机以目标速度匀速运行时的电流值，单位为：A。

注意：当电机匀速运行时，电流会自动加载到工作电流值 (crn)，此时在保持稳定运行的同时，产生的噪声也较小。电机连续运行时大部分时间均工作在匀速状态，此时不同于加速过程，不需要较大的扭矩，因此可根据实际负载情况，设置适当的工作电流，即可减小噪声，也可减小电机的发热现象。

寄存器地址：0x07

数据类型：float (32 位浮点数)

参数设置：参数设置范围下限为“0”，上限由驱动器型号决定，例如：VSMD136_025T 型，峰值电流为 2.5A，则参数值上限为“2.5”，需将 10 进制数据转换成 16 进制后设置。转换示例程序如下：

```
bool motor_cfg_crn(int tid, float crn)
{
    uint8_t data[4];
    uint32_t val = *((uint32_t*)&crn);

    data[0] = (val >> 24) & 0xff;
    data[1] = (val >> 16) & 0xff;
    data[2] = (val >> 8) & 0xff;
    data[3] = (val >> 0) & 0xff;

    return can_write_dat_regs(tid, CAN_DATA_CRN, data, 4);
}
```

4.2.9 保持电流 (crh)

数 据 说 明: 设置电机停止运行时的电流值, 单位为: A。

注 意: 电机在停止状态时, 有时需要施加一定的力量以保持电机静止不随着外力转动, 例如, 垂直滑动机构在静止时需抵消重力的影响。在电机处于停止状态时, 会加载保持电流值 (crh) 的电流, 以保持电机静止稳定。保持电流一般需要根据实际负载情况设置, 设置适当的保持电流值, 即可减小噪声, 也可减小电机的发热现象。在某些工控或仪器领域的设备, 电机大部分时间处于静止状态, 只有在需要运行时才会启动。

寄存器地址: 0x08

数 据 类 型: float (32 位浮点数)

参 数 设 置: 参数设置范围下限为 “0”, 上限由驱动器型号决定, 例如: VSMD136_025T 型, 峰值电流为 2.5A, 则参数值上限为 “2.5”, 需将 10 进制数据转换成 16 进制后设置。转换示例程序如下:

```
bool motor_cfg_crh(int tid, float crh)
{
    uint8_t data[4];
    uint32_t val = *((uint32_t*)&crh);

    data[0] = (val >> 24) & 0xff;
    data[1] = (val >> 16) & 0xff;
    data[2] = (val >> 8) & 0xff;
    data[3] = (val >> 0) & 0xff;

    return can_write_dat_regs(tid, CAN_DATA_CRH, data, 4);
}
```

4.2.10 传感器 1/传感器 2 下降沿/上升沿触发事件 (s1f/s1r/s2f/s2r)

数据说明: 设置当传感器 1/传感器 2 端口电平出现下降沿/上升沿时驱动器的响应动作。

注意: 使用传感器 1/传感器 2 下降沿、上升沿触发时, 需将对应传感器的工作模式设置为“0”(输入模式)。

寄存器地址: 0x09

数据类型: uint32 (32 位无符号整数)

参数设置: 4 字节数据中各字节对应含义如下表所示:

byte4	byte3	byte2	byte1
s1f (下降沿)	s1r (上升沿)	s2f (下降沿)	s2r (上升沿)

设置参数值及其对应的响应动作如下表所示:

参数设置值	动作说明
0	无动作 (只有状态位变化通知)
1	重新设置原点位置
2	减速停止
3	减速停止, 并在停止后重新设置原点位置
4	立刻停止
5	立刻停止, 并在停止后重新设置原点位置
6	正向连续运转 (正速度)
7	反向连续运转 (负速度)
8	启动离线功能
9	停止离线功能

4.2.11 传感器 3/传感器 4 下降沿/上升沿触发事件 (s3f/s3r/s4f/s4r)

数据说明: 设置当传感器 3/传感器 4 端口电平出现下降沿/上升沿时驱动器的响应动作。

注意: 1. 使用传感器 3/传感器 4 下降沿、上升沿触发时, 需将对应传感器的工作模式设置为“0”(输入模式)。

2. 本参数不适用于 VSMD1XX_010T 系列的驱动器。

寄存器地址: 0x0A

数据类型: uint32 (32 位无符号整数)

参数设置: 4 字节数据中各字节对应含义如下表所示:

byte4	byte3	byte2	byte1
s3f (下降沿)	s3r (上升沿)	s4f (下降沿)	s4r (上升沿)

设置参数值及其对应的响应动作说明参照“4.2.10 传感器 1/传感器 2 下降沿/上升沿触发事件 (s1f/s1r/s2f/s2r)”的参数设置说明。

4.2.12 传感器 5/传感器 6 下降沿/上升沿触发事件 (s5f/s5r/s6f/s6r)

数据说明： 设置当传感器 5/传感器 6 端口电平出现下降沿/上升沿时驱动器的响应动作。

注意： 1. 使用传感器 5/传感器 6 下降沿、上升沿触发时，需将对应传感器的工作模式设置为“0”（输入模式）。

2. 本参数不适用于 VSMD1XX_010T 系列的驱动器。

寄存器地址： 0x0B

数据类型： uint32（32 位无符号整数）

参数设置： 4 字节数据中各字节对应含义如下表所示：

byte4	byte3	byte2	byte1
s5f（下降沿）	s5r（上升沿）	s6f（下降沿）	s6r（上升沿）

设置参数值及其对应的响应动作说明参照“4.2.10 传感器 1/传感器 2 下降沿/上升沿触发事件 (s1f/s1r/s2f/s2r)”的参数设置说明。

4.2.13 设置传感器 1~6 的上拉/下拉模式 (pud)

数据说明： 设置传感器 1~传感器 6 的内部上拉/下拉模式，即传感器工作模式为输入时，驱动器内部设置该端口电平是上拉还是下拉（默认设置是上拉）。端口连接 NPN 型光电开关或微动开关时，需要设置为上拉；端口连接 PNP 型光电开关时需要设置为下拉。

注意： 1. 对 VSMD1XX_010T 系列驱动器，只可设置传感器 1~传感器 2 的输入模式。
2. 对非 VSMD1XX_010T 系列驱动器，只可设置传感器 3~传感器 6 的输入模式，传感器 1~传感器 2 的输入模式固定为上拉。

寄存器地址： 0x0C

数据类型： uint32（32 位无符号整数）

参数设置： 使用 4 字节数据的最低位字节中的 bit0~bit5，分别对应传感器 1~传感器 6 的输入模式（数据中其他非使用 bit 置“0”）。各 bit 设置为“0”时，表示该传感器端口为“内部下拉”；设置为“1”时，表示该传感器端口为“内部上拉”。

byte4	byte3	byte2	byte1
00000000	00000000	00000000	0 0 S6 S5 S4 S3 S2 S1

4.2.14 传感器 1~传感器 6 的工作模式 (s1/s2/s3/s4/s5/s6)

数据说明: 设置传感器 1/2/3/4/5/6 端口的工作模式是输入或是输出 (默认设置为输入)。

注意: 1. 对于 VSMD1XX_010T 系列驱动器, 只可设置传感器 1~传感器 2 的工作模式。
2. 对非 VSMD1XX_010T 系列驱动器, 只可设置传感器 3~传感器 6 的工作模式。

寄存器地址: 0x0D

数据类型: uint32 (32 位无符号整数)

参数设置: 使用 4 字节数据的最低位字节中的 bit0~bit5, 分别对应传感器 1~传感器 6 的工作模式 (数据中其他非使用 bit 置 "0")。各 bit 设置为 "0" 时, 表示该传感器端口为 "输入"; 设置为 "1" 时, 表示该传感器端口为 "输出"。

byte4	byte3	byte2	byte1
00000000	00000000	00000000	0 0 S6 S5 S4 S3 S2 S1

4.2.15 归零模式 (zmd)

数据说明: 设置归零的动作模式。关于归零的详细说明, 请参阅本公司另行制作的文档。

寄存器地址: 0x0E

数据类型: uint32 (32 位无符号整数)

参数设置: 设置参数值及含义如下表所示:

参数值	说明
0	归零功能关闭
1	一次归零
2	一次归零+安全位置
3	二次归零
4	二次归零+安全位置
6	编码器归零
7	编码器 Z 归零

4.2.16 原点传感器开放电平 (osv)

数据说明: 设置原点传感器在开放状态 (未触发状态) 时的电平。

注意: 传感器一般有开放状态和触发状态两个状态, 设置前务必确认归零用原点传感器状态, 否则归零动作不能正确进行。

寄存器地址: 0x0F

数据类型: uint32 (32 位无符号整数)

参数设置: 设置参数值及含义如下表所示:

参数值	说明
0	低电平
1	高电平

4.2.17 原点传感器端口 (snr)

数据说明: 设置原点传感器使用的传感器端口。

寄存器地址: 0x10

数据类型: uint32 (32 位无符号整数)

注意: 原点传感器使用端口的其他相关参数应作如下设置:

- 如传感器端口 1~6 中的某个传感器被指定为归零用原点传感器使用时, 则该传感器的工作模式需设置为“0” (输入模式)。
- 当第 n 号传感器被指定为归零传感器使用时, 如无确实需要, 推荐将该传感器的上升沿触发参数 (s[n]r) 及下降沿触发参数 (s[n]r) 需设置为“0” (无动作)。

参数设置: 设置参数值及含义如下表所示:

参数值	说明
0	设置传感器 1 为原点传感器
1	设置传感器 2 为原点传感器
2	设置传感器 3 为原点传感器
3	设置传感器 4 为原点传感器
4	设置传感器 5 为原点传感器
5	设置传感器 6 为原点传感器

4.2.18 归零速度 (zsd)

数据说明: 设置电机在执行归零动作时的速度, 单位为: 脉冲/秒。归零速度是归零过程中, 电机运转逼近原点传感器时所使用的速度, 设置的归零速度越低归零精度越高, 但是归零动作持续的时间越长, 因此需要根据实际情况设置合适的归零速度。

寄存器地址: 0x11

数据类型: float (32 位浮点数)

参数设置: -192,000~192,000 (正负表示电机运行方向)

注意: 参数设置时需转换成 16 进制, 示例程序参照“4.2.4 运行速度 (spd)”。

4.2.19 归零安全位置 (zsp)

数据说明: 设置归零动作结束后电机停止的位置, 此位置为距离原点的绝对位置, 单位为: 脉冲。归零结束后可以控制电机停止在离开原点一定距离的安全位置, 此位置的设置值一般和归零速度值 (zsd) 的方向相反。

寄存器地址: 0x12

数据类型: int32 (32 位有符号整数)

参数设置: -2,147,483,647~2,147,483,648

4.2.20 无握手启动等待时间 (dar)

数据说明: 上电后自动启动离线模式运行前的等待时间, 单位为: 秒。

寄存器地址: 0x14

数据类型: uint32 (32 位无符号整数)

参数设置: 设置参数值及含义如下表所示:

参数值	说明
0	上电后不自动执行离线运行模式
1 ~ 60	上电后无握手信号情况下, 自动启动离线运行模式的等待秒数

4.2.21 外接电位器阻值 (res)

数据说明: 驱动器可通过传感器接口连接滑动电位器, 并通过滑动电位器改变电阻值的方式控制电机运行速度, 本参数用于设置连接的滑动电位器最大阻值, 单位为: 欧姆。

注意: 本参数只适用于 VSMD1XX_080T 系列驱动器。

寄存器地址: 0x15

数据类型: uint32 (32 位无符号整数)

参数设置: 设置范围: 1,000~100,000

参数为 “0” 时表示关闭此功能。

4.2.22 正/负极限传感器 (msr/msv/psr/psv)

数据说明： 设置负极限传感器 (msr) 使用的传感器端口，以及负极限传感器在触发状态时的电平 (msv)；设置正极限传感器 (psr) 使用的传感器端口，以及正极限传感器在触发状态时的电平 (psv)。当负极限传感器触发后，电机不可以继续向负方向（速度为负）运行；当正极限传感器触发后，电机不可以继续向正方向（速度为正）运行。

注意： 正/负极限传感器使用端口的其他相关参数应作如下设置：

- 如 s3~s6 中的某个传感器被指定为正/负极限传感器使用时，则该传感器工作模式需设置为“0-输入模式”。
- 当第 n 号传感器被指定为正/负极限传感器使用时，如无确实需要，推荐将该传感器的上升沿触发参数 (s[n]r) 及下降沿触发参数 (s[n]r) 需设置为“0：无动作”。

寄存器地址： 0x17

数据类型： uint32 (32 位无符号整数)

参数设置： 4 字节数据中各字节对应含义如下表所示：

byte4	byte3	byte2	byte1
负极限传感器使用端口	负极限传感器的触发电平	正极限传感器使用端口	正极限传感器的触发电平

1. 传感器使用端口设置（正、负极限传感器设置方法相同）：

参数值	说明
0	相关极限传感器无设置
1	设置传感器 1 为极限传感器
2	设置传感器 2 为极限传感器
3	设置传感器 3 为极限传感器
4	设置传感器 4 为极限传感器
5	设置传感器 5 为极限传感器
6	设置传感器 6 为极限传感器

2. 传感器触发电平值的设置（正、负极限传感器设置方法相同）：

参数值	说明
0	低电平触发（平时为高电平）
1	高电平触发（平时为低电平）

4.2.23 上电电机使能 (pae)

数 据 说 明: 设置驱动器上电后是否自动设置电机使能。

寄存器地址: 0x18

数 据 类 型: uint32 (32 位无符号整数)

参 数 设 置: 设置参数值及含义如下表所示:

参数值	说明
0	上电不自动使能
1	上电自动使能

4.2.24 上电自动归零 (zar)

数 据 说 明: 设置驱动器上电后是否自动执行归零动作。

寄存器地址: 0x1A

数 据 类 型: uint32 (32 位无符号整数)

参 数 设 置: 设置参数值及含义如下表所示:

参数值	说明
0	上电后不执行归零
1	上电后自动执行归零

4.2.25 编码器归零电流 (zcr)

数 据 说 明: 当归零模式设置为“6: 编码器归零”时, 使用本参数设置使用编码器进行归零动作时的工作电流值, 单位为: A。

寄存器地址: 0x1C

数 据 类 型: float (32 位浮点数)

参 数 设 置: 参数设置范围下限为“0”, 上限由驱动器型号决定, 例如: VSMD136_025T 型, 峰值电流为 2.5A, 则参数值上限为“2.5”, 需将 10 进制数据转换成 16 进制后设置。

注 意: 本参数只适用于使用编码器的闭环型号驱动器。

4.2.26 外设控制端口 1 的输出比率 (nmos1r)

数 据 说 明: 当外部设备为 RGB 灯时, 设置外部设备控制端口 1 (cs1) 的输出比率用于调节白平衡, 配合输出端口 1 (cs1) 控制指令 nmos 使用。

寄存器地址: 0x1D

数 据 类 型: float (32 位浮点数)

参 数 设 置: 0 ~ 1.0

注 意: 本指令只适用于支持外部设备控制的 VSMD1XXE_XXXT 型号的驱动器。

4.2.27 外设控制端口 2 的输出比率 (nmos2r)

数据说明: 当外部设备为 RGB 灯时, 设置外部设备控制端口 2 (cs2) 的输出比率用于调节白平衡, 配合输出端口 2 (cs2) 控制指令 nmos 使用。

寄存器地址: 0x1E

数据类型: float (32 位浮点数)

参数设置: 0 ~ 1.0

注意: 本指令只适用于支持外部设备控制的 VSDM1XXE_XXXT 型号的驱动器。

4.2.28 外设控制端口 3 的输出比率 (nmos3r)

数据说明: 当外部设备为 RGB 灯时, 设置外部设备控制端口 3 (cs3) 的输出比率用于调节白平衡, 配合输出端口 3 (cs3) 控制指令 nmos 使用。

寄存器地址: 0x1F

数据类型: float (32 位浮点数)

参数设置: 0 ~ 1.0

注意: 本指令只适用于支持外部设备控制的 VSDM1XXE_XXXT 型号的驱动器。

4.2.29 编码器模式 (emod)

数据说明: 设置编码器的工作模式。编码器设置有效时, 传感器 3 端口以及传感器 4 端口作为正交编码器 A-、B-信号输入使用。

注意: 1. 本参数只适用于使用编码器的闭环型号驱动器。
2. 修改编码器模式后需要执行保存 (sav) 指令, 并在驱动器重新上电后生效。

寄存器地址: 0x20

数据类型: uint32 (32 位无符号整数)

参数设置: 设置参数值及含义如下表所示:

参数值	说明
0	编码器无效
1	编码器 A、B 信号有效, Z 信号无效
2	编码器 A、B、Z 信号都有效

4.2.30 编码器线数 (elns)

数据说明: 设置编码器线数。

注意: 本参数只适用于使用编码器的闭环型号驱动器。

寄存器地址: 0x21

数据类型: uint32 (32 位无符号整数)

参数设置: 设置参数范围: 10~10,000, 实际应用中线数上限值参照编码器厂商给出的参数。

4.2.31 电机单圈整步数 (estp)

数据说明: 设置电机单圈运行的整步数, 即步进电机在整步时转动一圈所需要的脉冲数, 例如对于步距角为 1.8° 的电机, 单圈所需的脉冲数为: $360 \div 1.8 = 200$ 。

注意: 本参数只适用于使用编码器的闭环型号驱动器。

寄存器地址: 0x22

数据类型: uint32 (32 位无符号整数)

参数设置: 根据步进电机步距角指标计算设置。

4.2.32 堵转重试次数 (erty)

数据说明: 步进电机发生堵转时进行补偿的次数, 当重试次数达到参数 ERTY 指定的次数后, 如果补偿不成功, 编码器错误标志 (enc_err) 置位。

注意: 本参数只适用于使用编码器的闭环型号驱动器。

寄存器地址: 0x23

数据类型: uint32 (32 位无符号整数)

参数设置: 设置参数值及含义如下表所示:

参数值	说明
0	忽略编码器错误, 无限次数补偿
1~100	补偿次数

4.2.33 编码器方向 (edir)

数据说明: 设置编码器的计数方向与电机速度方向的关系。

注意: 本参数只适用于使用编码器的闭环型号驱动器。

寄存器地址: 0x24

数据类型: uint32 (32 位无符号整数)

参数设置: 设置参数值及含义如下表所示:

参数值	说明
0	负方向 (电机速度方向为正时, 编码器方向为负)
1	正方向 (电机速度方向为正时, 编码器方向也为正)

4.2.34 编码器灵敏度 (ez)

数据说明: 编码器灵敏度, 此参数设置值越小, 表示编码器灵敏度越高。

注意: 本参数只适用于使用编码器的闭环型号驱动器。

寄存器地址: 0x25

数据类型: uint32 (32 位无符号整数)

参数设置: 0~100

4.2.35 编码器错误处理方式 (ewr)

数 据 说 明: 设置编码器出现错误时(编码器错误标志置位)自动执行的后续处理方式。

注 意: 本参数只适用于使用编码器的闭环型号驱动器。

寄存器地址: 0x26

数 据 类 型: uint32 (32 位无符号整数)

参 数 设 置: 设置参数值及含义如下表所示:

参数值	说明
0	无动作
1	电机立即停止
2	电机立即停止, 同时置电机失能

4.2.36 电机减速比 (eratio)

数 据 说 明: 设置编码器齿轮箱速比。

注 意: 本指令只适用于使用编码器的闭环型号驱动器, 且编码器连接减速机输出轴时使用。

寄存器地址: 0x27

数 据 类 型: float (32 位浮点数)

参 数 设 置: 参数设置范围: >0

4.2.37 堵转检测灵敏度 (esds)

数 据 说 明: 编码器对电机堵转检测的灵敏度, 此参数值越大表示检测灵敏度越高。

注 意: 本参数只适用于使用编码器的闭环型号驱动器。

寄存器地址: 0x28

数 据 类 型: float (32 位浮点数)

参 数 设 置: 参数设置范围: 0.1~0.9

4.2.38 软件负限位 (sml)

数 据 说 明: 通过驱动器程序控制电机运行在负方向上的极限位置, 此位置为相对于原点的绝对位置, 单位为: 脉冲。

注 意: 当电机向负方向运行到达软件负限位时, 电机立即停止运行; 当电机已经处于软件负限位外的位置时, 电机无法运行。

寄存器地址: 0x2A

数 据 类 型: uint32 (32 位无符号整数)

参 数 设 置: 参数设置范围: -2,147,483,647~2,147,483,648

当参数设置为“0”时, 表示软件负限位无控制。

4.2.39 软件正限位 (spl)

数 据 说 明: 通过驱动器程序控制电机运行在正方向上的极限位置, 此位置为相对于原点的绝对位置, 单位为: 脉冲。

注 意: 当电机向正方向运行到达软件正限位时, 电机立即停止运行; 当电机已经处于软件正限位外的位置时, 电机无法运行。

寄存器地址: 0x2B

数 据 类 型: uint32 (32 位无符号整数)

参 数 设 置: 参数设置范围: -2,147,483,647~2,147,483,648

当参数设置为“0”时, 表示软件负限位无控制。

4.3 数据寄存器写入指令

上位机发出指令对驱动器数据寄存器进行写处理时，指令通信帧中需要做如下设置：

SID 设置：

目标设备 ID：设置对象的驱动器 ID。

源设备 ID：从上位机发送指令时设置为“1”。

控制字：C1=1、C0=0

寄存器地址：数据寄存器中操作对象项目的地址。

数据字节设置：

设置写入数据寄存器的参数需注意参数类型需和定义类型一致，如数据定义类型为浮点数时，需将 10 进制浮点数转换为 16 进制浮点数后设置在数据字节中。

根据 CAN 协议定义，数据部分最大可使用 8 字节，由于数据寄存器单地址可保存 4 字节数据，因此可通过一条指令同时写入两个数据至数据寄存器地址连续的两个寄存器中。

4.4 参数写入示例

4.4.1 示例 1

指令：上位机发出指令至设备号为“1”的驱动器，将细分（mcs）修改为“6：64 细分”。

■ SID 设置内容：

目标设备 ID： 1

源设备 ID： 0

控制字： C1=1、C0=0

数据寄存器地址：0x02（细分参数在数据寄存器中的地址）

由以上设置内容，得到标准帧 SID 设置值为：**0x00C2**

标准帧 SID 中具体各 bit 设置如下所示：

10	9	8	7	6	5	4	3	2	1	0
TARGET ID				C1	C0	CMD/REG ADDR				
0	0	0	1	1	0	0	0	0	1	0

由以上设置内容，得到扩展帧 SID 设置值为：**0x00080182**

扩展帧 SID 中具体各 bit 设置如下所示：

28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
0	TARGET ID										0	SOURCE ID										C1	C0	CMD/REG ADDR							
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0			

■ 数据设置：

细分（mcs）的数据类型定义为“uint32”，因此将细分参数设置为“6”时，4 字节数据分别设置为：**0x00 0x00 0x00 0x06**

4.4.2 示例 2

指令：上位机发出指令至设备 ID 为 “1” 的驱动器，将电机运行速度（spd）设置为 “192000”。

■ SID 设置内容：

目标设备 ID： 1

源设备 ID： 0

控制字： C1=1、C0=0

数据寄存器地址：0x03（电机运行速度参数在数据寄存器中的地址）

由以上设置内容，得到标准帧 SID 设置值为：**0x00C3**

标准帧 SID 中具体各 bit 设置如下所示：

10	9	8	7	6	5	4	3	2	1	0
TARGET ID				C1	C0	CMD/REG ADDR				
0	0	0	1	1	0	0	0	0	1	1

由以上设置内容，得到扩展帧 SID 设置值为：**0x00080103**

扩展帧 SID 中具体各 bit 设置如下所示：

28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	TARGET ID									0	SOURCE ID									C1	C0	CMD/REG ADDR						
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1

■ 数据设置：

运行速度（spd）的数据类型定义为 “float”，将十进制浮点数 192000 转换为 16 进制为 0x483b8000，因此 4 字节数据分别设置为：**0x48 0x3B 0x80 0x00**

4.4.3 示例 3

指令： 上位机发出指令至设备 ID 为 “1” 的驱动器，将传感器 2 的上升沿触发动作为 “5：立刻停止，并在停止后重新设置原点位置”。

■ SID 设置内容：

目标设备 ID： 1

源设备 ID： 0

控制字： C1=1、C0=0

数据寄存器地址： 0x09（传感器 1/传感器 2 下降沿/上升沿触发事件参数在数据寄存器中的地址）

由以上设置内容，得到标准帧 SID 设置值为：**0x00C9**

标准帧 SID 中具体各 bit 设置如下所示：

10	9	8	7	6	5	4	3	2	1	0
TARGET ID				C1	C0	CMD/REG ADDR				
0	0	0	1	1	0	0	1	0	0	1

由以上设置内容，得到扩展帧 SID 设置值为：**0x00080109**

扩展帧 SID 中具体各 bit 设置如下所示：

28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0	TARGET ID									0	SOURCE ID										C1	C0	CMD/REG ADDR							
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1		

■ 数据设置：

传感器 1/传感器 2 下降沿/上升沿触发事件（s1f/s1r/s2f/s2r）参数在数据寄存器中的地址为 “0x09”、数据类型定义为 “uint32”，传感器 2 的上升沿触发动作（s2r）参数在数据的第 4 字节处设置，此时先要读出数据寄存器中该参数当前的设置值，如当前数据寄存器地址 “0x09” 中的数据为：0x01020304，将传感器 2 上升沿触发动作设置为 “5：立刻停止，并在停止后重新设置原点位置” 时，4 字节数据分别设置为：**0x01 0x02 0x03 0x05**

4.4.4 示例 4

指令：上位机发出指令至设备 ID 为 “1” 的驱动器，同时设置加速度值 (acc) 为 “10000”、减速度值 (dec) 为 “20000”。

■ 分析：

从数据寄存器的地址定义中可以看到，加速度参数的地址和减速度参数的地址连续，分别是 0x04 和 0x05，由于数据寄存器单个地址的数据长度为 4 字节，而且 CAN 协议规定在一个通信帧中最多可传送 8 字节数据，因此可以使用一条写入指令同时完成加速度、减速度参数的写入。

■ SID 设置内容：

目标设备 ID: 1

源设备 ID: 0

控制字: C1=1、C0=0

数据寄存器地址: 0x04 (连续地址的首地址)

由以上设置内容，得到标准帧 SID 设置值为：**0x00C4**

标准帧 SID 中具体各 bit 设置如下所示：

10	9	8	7	6	5	4	3	2	1	0
TARGET ID				C1	C0	CMD/REG ADDR				
0	0	0	1	1	0	0	0	1	0	0

由以上设置内容，得到扩展帧 SID 设置值为：**0x00080104**

扩展帧 SID 中具体各 bit 设置如下所示：

28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	TARGET ID									0	SOURCE ID									C1	C0	CMD/REG ADDR						
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0

■ 数据设置：

加速度 (acc)、减速度 (dec) 的数据类型定义均为 “float”，将加速度值 10000 转换为 16 进制浮点数为 0x461C4000，减速度值 20000 转换为 16 进制浮点数为 0x469C4000，8 字节数据分别设置为：**0x46 0x1C 0x40 0x00 0x46 0x9C 0x40 0x00**

5. 上位机软件及协议总表文件

5.1 上位机软件

为方便客户进行 VSMD 驱动器产品的动作确认和调试，本公司同时开发了运行于 PC 端的上位机软件并随产品提供给客户。上位机软件可以协助客户完成以下工作：

- ◆ 确认驱动器、电机的工作状态。
- ◆ 设置驱动器、电机的各个工作参数。
- ◆ 发布控制电机运行的各项指令。

关于上位机软件的详细使用说明，请参照本公司另行制作的上位机软件使用说明文档。

5.2 协议总表文件

5.2.1 概述

主控设备和驱动器子设备间的通信联系，是通过收发符合 CAN2.0 协议要求的通信帧读取控制器的工作状态、向驱动器写入各项工作参数以及发送动作控制指令完成的。为帮助客户正确生成通信帧，本公司随产品同时提供了“**VSMD_CAN_BUS_V1.1.xlsm**”的 CAN2.0 协议总表文件，在其中除列出了指令码、各寄存器分配的地址、参数数据类型等以使用户速查外，还提供了指令生成工具及浮点数转换工具，用户可以使用这些工具迅速生成用于与 VSMD 驱动器通信的通信帧。

本文件中包含下面 5 个 Sheet：

- ◆ definition：适用于通信扩展帧的指令码、状态寄存器、数据寄存器定义。
- ◆ definition_std：适用于通信标准帧的指令码、状态寄存器、数据寄存器定义。
- ◆ definition_vmvd：其他用途，可忽视。
- ◆ convert：标准帧、扩展帧生成工具以及 10 进制整数、10 进制浮点数转 16 进制工具。
- ◆ baudrate：通信波特率一览表。

5.2.2 扩展帧标识符生成

用户可在“convert” sheet 下面位置上，对应填写接收方设备 ID (TARGET ID)、发送方设备 ID (SOURCE ID)、控制字 (CTRL ID)，参照“definition” sheet 内容填写地址/指令 (ADDR/CMD) 后，会在“EXT-ID FOR NORMAL”处生成普通指令的扩展帧标识符 (SID)，在“EXT-ID FOR BROADCAST”处生成广播指令的扩展帧标识符 (SID)。见下图所示：

TARGET ID	SOURCE ID	CTRL ID	ADDR/CMD	EXT-ID FOR NORMAL	EXT-ID FOR BROADCAST
1	0	2	3	0x00080103	0x00040103

目标设备ID

源设备ID

控制字

地址/指令

普通指令

广播指令

5.2.3 标准帧标识符生成

用户可在“convert” sheet 下面位置上，对应填写接收方设备 ID (TARGET ID)、控制字 (CTRL ID)，参照“definition_std” sheet 内容填写地址/指令 (ADDR/CMD) 后，会在“EXT-ID FOR NORMAL”处生成普通指令的标准帧标识符 (SID)，在“EXT-ID FOR BROADCAST”处生成广播指令的标准帧标识符 (SID)。见下图所示：

TARGET ID	CTRL ID	ADDR/CMD	STD-ID FOR NORMAL	STD-ID FOR BROADCAST
1	2	3	0x000000C3	0x00000043

目标设备ID

控制字

地址/指令

普通指令

广播指令

注意：由于标准帧中只使用 5bit 作为地址描述，因此可寻址范围小于扩展帧指令。

5.2.4 进制转换工具

在文件“convert” Sheet 中，提供了“10 进制转 16 进制”、“浮点数转 16 进制”两个工具供用户在测试时使用：

输入

DEC2HEX	FLOAT2HEX
-12800	5000.00
0xFFFFFCE00	0x459C4000

输出

10进制转16进制

浮点数转16进制

进制转换例程参照“7.附录 1 - 进制转换例程”。

5.2.5 协议总表使用示例 1

指令设置：主控向设备号为“5”的驱动器的数据寄存器中写入电机运行的目标速度“19200”。

- ① 在“definition”或“definition_std” sheet 中，查询电机速度“speed (spd)”在数据存储器中的地址为“0x03”，数据定义类型为“float”。

0x0C	pps		position		
0x0D	sv	id	dr	delay	
0x0E	nmos	id	dr		
0x0F	auto res status	mod	period		
0x10	action	code	data		
0x11	action add position	position	speed		
0x12	action add rmv	distance	speed		
0x13	reserved				
0x14	reserved				
0x15	reserved				
0x16	reserved				
0x17	reserved				
0x18	reserved				
0x19	reserved				
0x1A	reserved				
0x1B	shake	cmd	data		
0x1C	reserved				
0x1D	encoder	ext			
0x1E	read status regs	addr	size		
0x1F	read data regs	addr	size		

ADDR	DATA REGISTERS	TYPE
0x00	device id (cid)	uint32
0x01	baudrate (bdr)	uint32
0x02	micro-steps (mcs)	uint32
0x03	speed (spd)	float
0x04	acceleration (acc)	float
0x05	deceleration (dec)	float
0x06	current (accelariton) (cra)	float
0x07	current (normal) (cm)	float
0x08	current (holding) (crh)	float
0x09	s1f - s1r - s2f - s2r	uint32
0x0A	s3f - s3r - s4f - s4r	uint32
0x0B	s5f - s5r - s6f - f6r	uint32
0x0C	-	
0x0D	s1 - s6 config (0 - input / 1 - output)	uint32
0x0E	zero mode (zmd)	uint32
0x0F	sensor level in open-state (osv)	uint32
0x10	sensor for zero function (snr)	uint32
0x11	zero speed (zsd)	float
0x12	safe position (zsp)	uint32
0x13	action mode (dmd)	uint32
0x14	action automatic running delay time (dar)	uint32

- ② 在“convert 型为“ftd 中，如下图所示在“TARGET ID 王处填入“5”（指令接收驱动器 CID），“CTRL ID”处填入“2”（上位机发送至驱动器的数据寄存器写入指令，控制字 C1=1，C0=0），在“ADDR/CMD”处填入“3”（电机速度在数据寄存器中的地址为 0x03）后，在“EXT-ID FOR NORMAL”处得到扩展帧指令 SID 为“0x00280103”以及标准帧指令 SID 为“0x000002C3”。同时，在“EXT-ID FOR BOARDCASTI 以及“STD-ID FOR BOARDCASTI 处显示该指令作为广播指令使用时的 SID 设置内容。

TARGET ID	SOURCE ID	CTRL ID	ADDR/CMD	EXT-ID FOR NORMAL	EXT-ID FOR BROADCAST
5	0	2	3	0x00280103	0x00040103

TARGET ID	CTRL ID	ADDR/CMD	STD-ID FOR NORMAL	STD-ID FOR BROADCAST
5	2	3	0x000002C3	0x00000043

- ③ 由于电机速度类型定义为浮点数，使用同 sheet 中的数据类型转换工具将写入速度值“19200”转换为 16 进制浮点数为“0x46960000”，因此 4 字节数据设置为：46 96 00 00

DEC2HEX	FLOAT2HEX
10000	19200.00
0x00002710	0x46960000

如此根据工具转换的结果，帮助用户生成将电机速度的写入数据寄存器的指令。

5.2.6 协议总表使用示例 2

指令设置：主控向设备号为“10”的驱动器发出指令，以“32000”运行速度运行至绝对位置为“50000”处停止。

- ① 在“definition”或“definition_std” sheet 中，查询控制电机运行至指定位置的指令“pos”的指令码为“0x06”，同时可看到该指令可设置两个参数：运行目标位置（position、必须），运行速度（speed、可选）

CMD	DESCRIPTION	D1	D2	D3	D4	D5	D6	D7	D8
0x01	enable								
0x02	disable / off								
0x03	org								
0x04	stop	0/1							
0x05	move	speed							
0x06	pos	position					speed		
0x07	save								
0x08	output	s1-8	0/1						
0x09	zero start								
0x0A	zero stop								
0x0B	rmv	distance					speed		
0x0C	pps	position							
0x0D	sv	id	dr	delay					
0x0E	nmos	id	dr						
0x0F	auto res status	mode	period						
0x10	action	code	data						
definition		definition_std	definition_vmv	convert					

- ② 在“convert” sheet 中，如下图所示在“TARGET ID”处填入“10”（指令接收驱动器 CID），“CTRL ID”处填入“3”（上位机发送至驱动器的操作指令，控制字 C1=1，C0=1），在“ADDR/CMD”处填入“6”（“pos”指令的指令码为“0x06”）后，在“EXT-ID FOR NORMAL”处得到扩展帧指令 SID 为“0x00500186”以及标准帧指令 SID 为“0x00000566”。

TARGET ID	SOURCE ID	CTRL ID	ADDR/CMD	EXT-ID FOR NORMAL	EXT-ID FOR BROADCAST
10	0	3	6	0x00500186	0x00040186

TARGET ID	CTRL ID	ADDR/CMD	STD-ID FOR NORMAL	STD-ID FOR BROADCAST
10	3	6	0x00000566	0x00000066

- ③ 由于目标位置（position）类型定义为“int32”、速度（speed）类型定义为“float”（参照“3.3.6 绝对位置运行（pos）”），因此可使用同 sheet 中的数据类型转换工具将目标位置值“50000”转换为 16 进制数为“0x0000C350”，速度值“32000”转换为 16 进制浮点数为“0x46FA0000”，最终得到该指令的 8 字节数据为“00 00 C3 50 46 FA 00 00”

DEC2HEX	FLOAT2HEX
50000	32000.00
0x0000C350	0x46FA0000

如此根据工具转换的结果，帮助用户生成控制电机以指定速度运行至指令位置的操作指令。

5.2.7 协议总表使用示例 3

指令设置：读取设备号为“8”的驱动器连接电机的当前速度、当前位置及当前状态信息。

- ① 在“definition”或“definition_std”sheet中，查询读取驱动器状态信息的指令“read status regs”的指令码为“0x1E”，同时可看到该指令必须设置两个参数：状况寄存器地址（addr），读取寄存器个数（size）。

0x17	reserved		
0x18	reserved		
0x19	reserved		
0x1A	reserved		
0x1B	shake	cmd	data
0x1C	reserved		
0x1D	encoder	evt	
0x1E	read status regs	addr	size
0x1F	read data regs	addr	size

definition	definition_std	definition_vmvd	convert
------------	----------------	-----------------	---------

- ② 在同sheet上的状态寄存器一览处，可以看到当前速度、当前位置以及当前状态的寄存器地址是连续的，首地址为“0x00”。因此只需要发一条指令即可读取状态寄存器中这三项数据。

ADDR	STATUS REGISTERS				TYPE
0x00	actual speed				float
0x01	actual position				int32
0x02	actual status bits				uint32
0x03	reversed				
0x04	reversed				
0x05	reversed				
0x06	reversed				
0x07	reversed				
0x08	reversed				
0x09	reversed				
0x0A	V	S	M	D	char
0x0B	1	0	6	-	char
0x0C	0	2	5	T	char
0x0D	1	0	0	0	char

- ③ 在 “convert” sheet 中，如下图所示在 “TARGET ID” 处填入 “8”（指令接收驱动器 CID）， “CTRL ID” 处填入 “3”（上位机发送至驱动器的操作指令，控制字 C1=1，C0=1），在 “ADDR/CMD” 处填入 “1E”（“read status regs” 指令的指令码为 “0x1E”）后，在 “EXT-ID FOR NORMAL” 处得到扩展帧指令 SID 为 “0x0040019E” 以及标准帧指令 SID 为 “0x0000047E”。

TARGET ID	SOURCE ID	CTRL ID	ADDR/CMD	EXT-ID FOR NORMAL	EXT-ID FOR BROADCAST
8	0	3	1E	0x0040019E	0x0004019E

TARGET ID	CTRL ID	ADDR/CMD	STD-ID FOR NORMAL	STD-ID FOR BROADCAST
8	3	1E	0x0000047E	0x0000007E

- ④ 由于指令参数的寄存器地址 (addr) 类型定义为 “uint8”、读取寄存器个数 (size) 类型定义为 “uint8”（参照 “3.11 读取状态寄存器 (read_sts_regs)”），因此该指令的 2 字节数据为 “00 03”。

如此根据工具转换的结果，帮助用户生成读取电机当前速度、当前位置及当前状态的操作指令。

6. 联系我们

北京伟恩斯技术有限公司

<http://www.vincetech.com>

■ 北京公司

北京市朝阳区芍药居 101 号世奥国际中心 B 座 3011

■ 深圳公司

广东省深圳市龙华区福城街道科利邦工业园 C 栋 7 楼

7. 附录 1 – 进制转换例程

驱动器参数中的部分参数数据类型定义为浮点数，这样设置参数时经常遇到需要将十进制整数转换为 16 进制的浮点数的情况，反之，对从驱动器寄存器中读取的数据也会有进行 16 进制浮点数转十进制后使用的情况。针对这两个常用处理，这里给出两个例程供用户在编程时参考。

1. 十进制无符号整数转 16 进制浮点数：

```
float to_float(unsigned int data)
{
    float* f = *((float*)(&data));
    return *f;
}
```

2. 16 进制浮点数转十进制无符号整数：

```
unsigned int from_float(float f)
{
    unsigned int* d = *((unsigned int*)(&f));
    return *d;
}
```