ROBS-1601 产品手册

广 州 欧 兹 电 子 科 技 有 限 公 司

OC-Servo Electronics Technology Co.,Ltd

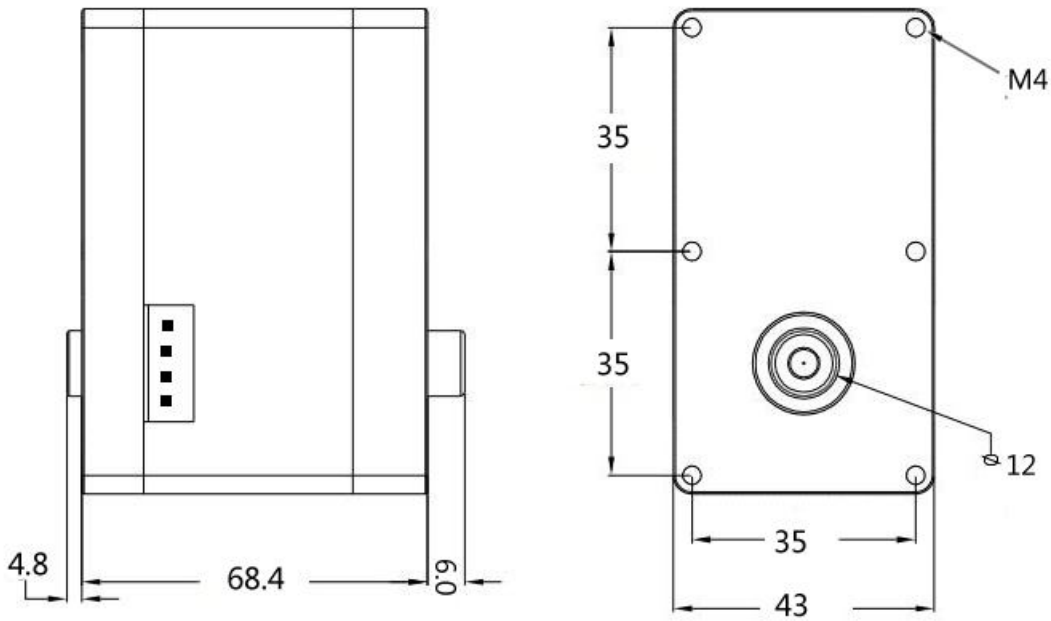
第一章 概　述

1.1　产品特性

ROBS-1601 机器人舵机是本公司研发生产的一种集电机、伺服驱动、总线式通讯接口为一体的集成伺服单元，主要用于机器人的关节、轮子驱动以及机械臂的关节驱动，也可用于其他需要精确位置控制的场合。ROBS-1601 的特点如下所示：

- 堵转扭矩：160Kg.cm （18.0V）
- 高压供电：DC 12.0V～18.0V
- 高分辨率 0.15°
- 独特的连接方式，适合多种组合拼装
- 高精度全金属齿轮组，双滚珠轴承
- 全金属外壳设计，散热效果突出
- 伺服模式下转动范围 0-360°
- 可设置为电机模式，步进模式
- 485 总线连接，抗干扰强，传输距离远
- 通讯速率：38400bps-1Mbps
- 采用 OCS 通讯协议
- 具备位置、温度、速度、电压等反馈

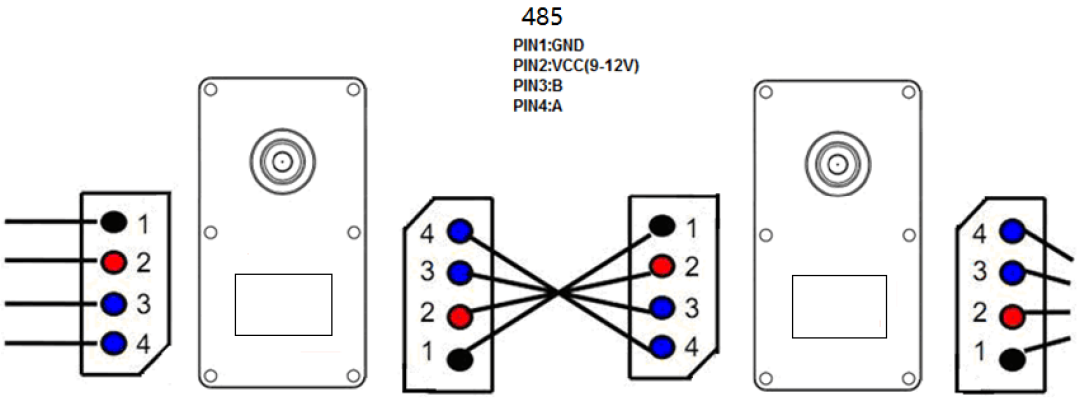
1.2 结构尺寸



1.3 电气连接

1.3.1 引脚定义

ROBS-1601 机器人舵机电气接口如下图，两组引脚定义一致的接线端子可将舵机逐个串联起来。

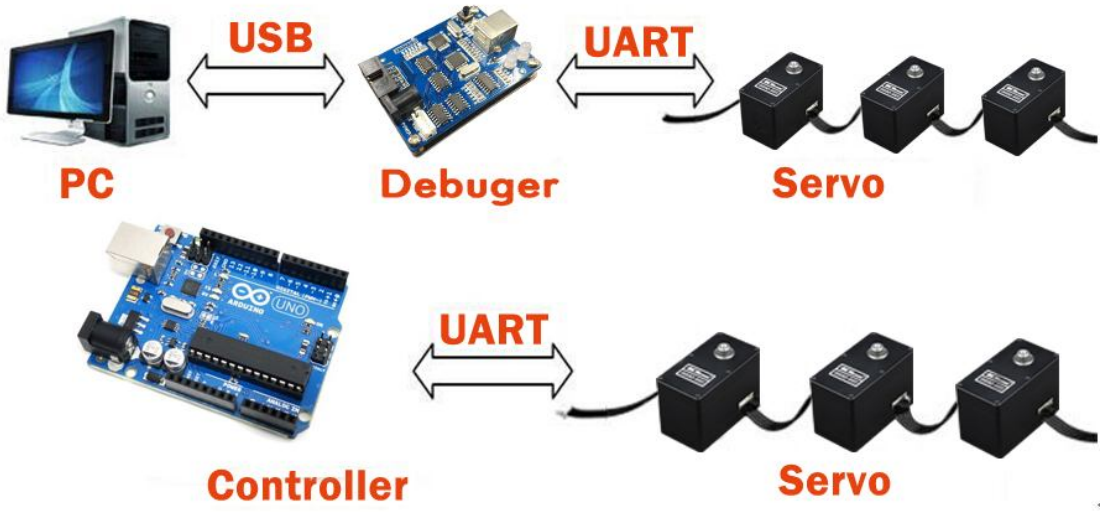


1.3.2 舵机通讯方式

ROBS-1601 采用全双工异步串行总线通讯方式，理论上 254 个机器人舵机可以通过总线组成链接，通过 UART 异步串行接口统一控制。每个舵机可以设定不同的 ID，多个舵机可以统一运动也可以单个独立控制。

ROBS-1601 的通讯指令集开放，通过异步串行接口与用户的上位机(控制器或 PC 机)通讯，您可对其进行参数设置、功能控制。通过异步串行接口发送指令，ROBS-1601 可以设置为电机控制模式或伺服模式。 在电机控制模式下，ROBS-1601 可以作为直流减速电机使用，速度可调；在伺服模式的时候，ROBS-1601 拥有 0-360° 的转动范围。在伺服模式下本产品具备精确位置控制性能，速度可调。

只要符合协议的全双工 UART 异步串行接口都可以和 ROBS-1601 进行通讯，对 ROBS-1601 进行各种控制。主要有以下几种形式：



方式 1：通过调试器控制 ROBS-1601

PC 会将调试器识别为串口设备，上位机软件通过串口发出符合协议格式的数据包，经调试器转发给 ROBS-1601。ROBS-1601 会执行数据包的指令，如果是查询指令则舵机会返回应答数据包。

您可根据本手册提供的通讯协议设计专用的 PC 端软件。

方式 2：通过 Arduino 或其他控制器控制 ROBS-1601

方式 1 可以快捷地调试本公司机器人舵机、修改舵机的各种性能与功能参数。但是，这种方式离不开 PC 机，不能搭建独立的机器人或机械臂。采用 Arduino 控制器或自行设计其他专用的控制器来控制舵机。不过您需要将 TTL 电平转换为 485 电平。

第二章 通讯协议

2.1 通信协议概要

本控制器和舵机之间采用问答方式通信，控制器发出指令包，舵机返回应答包。

一个网络中允许有多个舵机，所以每个舵机都分配有一个 ID 号。控制器发出的控制指令中包含 ID 信息，只有匹配上 ID 号的舵机才能完整接收这条指令，并返回应答信息。

通信方式为串行异步方式，一帧数据分为 1 位起始位，8 位数据位和 1 位停止位，无奇偶校验位，共 10 位。

2.2 OCS 指令包

OCS 指令包格式：

字头	ID号	数据长度	指令	参数	校验和
0XFF 0XFF	ID	Length	Instruction	Parameter1 ...Parameter N	Check Sum

字头：连续收到两个 0XFF ，表示马上需要执行 OCS 指令包。

ID：每个舵机都有一个 ID 号。ID 号范围 0～253,转换为十六进制 0X00～0XFD。

广播 ID：254 为广播 ID,若指令中的 ID 号为 254(0XFE)，所有的舵机均接收指令，但都不返回应答信息。

数据长度：具体长度请参照每条指令解释。

参数：除指令外需要补充的控制信息。

校验和：校验和 Check Sum，计算方法如下：

Check Sum = ~ (ID + Length + Instruction + Parameter1 + ... Parameter N)

若括号内的计算和超出 255,则取最低的一个字节，“~”表示取反。

2.3 OCS 模式应答包

应答包是本产品对控制器部分指令的应答，应答包具有如下格式：

字头	ID	数据长度	当前状态	参数	校验和
0XFF 0XFF	ID	Length	ERROR	Parameter1 ...Parameter N	Check Sum

返回的应答包包含了舵机的当前状态 ERROR，若舵机当前工作状态不正常，会通过这个字节反映出来，每一位的代表的信息如下：

BIT	名称	详细
BIT7	0	---
BIT6	0	---
BIT5	过载	位置模式运行时输出扭矩小于负载置
BIT4	0	---
BIT3	电流错误	电流超过指定范围置
BIT2	温度错误	温度超过指定范围置
BIT1	角度错误	角度传感器出错置
BIT0	电压错误	电压超过指定范围置

若 ERROR 为 0，则舵机无报错信息。

若指令是读指令 READ DATA，则 Parameter1 ...Parameter N 是读取的信息。

2.4 OCS 指令类型

OCS 模式可用指令如下：

指令	功能	值	数据长度
PING（查询）	查询工作状态	0X01	0X02
READ（读）	查询控制表里的字符	0X02	0X04
WRITE（写）	往控制表里写入字符	0X03	N+3
REG WRITE（异步写）	类似于WRITE DATA，但是控制字符写入后并不马上动作，直到ACTION指令到达	0X04	N+3
ACTION（执行异步写）	触发REG WRITE写入的动作	0X05	0
SYNC WRITE（同步写）	用于同时控制多个舵机	0X83	(L + 1) * N + 4
BULKWRITE DATA（突发写）	用于同时控制多个舵机与不连续内存地址写入	0x09	看详细说明
RESET（复位）	把控制表复位为出厂值	0X06	0

2.4.1 查询状态指令 PING

功 能 本指令用于读取舵机的工作状态

长 度 0X02

指 令 0X01

参 数 无

示 例： 读取 1 号舵机的工作状态

指令包：0XFF 0XFF 0X01 0X02 0X01 0XFB

字头	ID	数据长度	指令	校验和
0XFF 0XFF	0X01	0X02	0X01	0XFB

应答包： 0XFF 0XFF 0X01 0X02 0X00 0XFC

字头	ID	数据长度	工作状态	校验和
0XFF 0XFF	0X01	0X02	0X00	0XFC

2.4.2 读指令 READ

功 能 本指令用于读取舵机内部的数据

长 度 0X04

指 令 0X02

参 数1 数据读出段的首地址

参 数2 读取数据的长度

示 例： 读取 1 号舵机的内部温度

通过内存控制表知道地址 0X3F（参数 1）是温度的地址，然后需要读取一个字节（0X01）。

指令包：0XFF 0XFF 0X01 0X04 0X02 0X3F 0X01 0XCB

字头	ID	数据长度	指令	参数	校验和
0XFF 0XFF	0X01	0X04	0X02	0X3F 0X01	0XB8

应答包： 0XFF 0XFF 0X01 0X03 0X00 0X20 0XDD

字头	ID	数据长度	工作状态	参数	校验和
0XFF 0XFF	0X01	0X03	0X00	0X1E	0XDD

读出的数据是 0X1E，0X1E 转换为 10 进制为 30，说明当前的温度是 30℃。

2.4.3 写指令 WRITE

功 能 本指令用于写入参数到舵机内存控制表

长 度 N+3 (N 为写入数据的个数)

指 令 0X03

参 数 1 数据写入段的首地址

参 数 2 写入的第一个数据

参 数 3 第二个数据

参 数 N+1 第 N 个数据

示　例：　把一个任意编号的舵机 ID 设置为 1。

通过查询后文的内存控制表我们知道，保存舵机 ID 号的地址为 0X05，所以在地址 0X05 写入 1 即可。我们用广播 ID254(0XFE)发送指令即可。

指令包：　0XFF 0XFF 0XFE 0X04 0X03 0X05 0X01 0XF4

字头	ID	数据长度	指令	参数	校验和
0XFF　0XFF	0XFE	0X04	0X03	0X05　0X01	0XF4

2.4.4　异步写指令 REG WRITE

REG WRITE 指令跟 WRITE 指令有些相似，只是执行的时间不同。当收到 REG WRITE 指令包时，舵机会把收到的数据储存在缓冲区备用，并把地址 0X40 设置为 1。当收到 ACTION 指令后，储存的指令最终被执行。

功 能　　本指令用于异步写参数到舵机内存控制表

长 度　　N+3 (N 为要写入数据的个数)

指 令　　0X04

参 数 1　　数据要写入的首地址

参 数 2　　要写入的第一个数据

参 数 3　　要写入的第二个数据

参 数 N+1　　要写入的第 N 个数据

2.4.5 执行异步写指令 ACTION

功 能　　用于执行 REG WRITE 写入的全部指令

长 度　　0X02

指 令　　0X05

参 数　　无

ACTION 指令在同时控制多个舵机的时候非常有用。

在控制多个不同 ID 的舵机时，使用 ACTION 指令可以使第一个和最后一个舵机同时执行各自的动作，中间无延时。

对总线上的多个舵机发送 ACTION 指令时，要用到广播 ID254（0XFE）,因此，发送此指令不会有数据帧返回。

示　例：让 0 号舵机运动至 0°位置,1 号舵机运行至 360°位置，两个舵机需要同时动作

分　析：因为需要两个动作同时动作，所以我们可以运用上面 2.4.4 的异步写 REG_WRITE 指令以及 ACTION 指令实现它们同时动作，所以按以下步骤分别写入如，最后用 ACTION 执行 REG WRITE 写入的所有指令即可。因为本舵机 0-360°对应数值 0-4095，所以 0°位置为 0X0000,360°位置为 0X0FFF。

ID=0；指令 = REG_WRITE；地址 = 0X2A；数据 = 0X00, 0X00

ID=1；指令 = REG_WRITE；地址 = 0X2A；数据 = 0XFF, 0X0F

ID=0XFE；指令 = ACTION

0 号舵机指令包：0XFF 0XFF 0X00 0X05 0X04 0X2A 0X00 0X00 0XCC

字头	ID	数据长度	指令	参数	校验和
0XFF 0XFF	0X00	0X05	0X04	0X2A 0X00 0X00	0XCC

0号舵机应答包：0XFF 0XFF 0X00 0X02 0X01 0XFC

字头	ID	数据长度	工作状态	参数	校验和
0XFF 0XFF	0X00	0X02	0X00		0XFD

1 号舵机指令包：0XFF 0XFF 0X01 0X05 0X04 0X2A 0XFF 0X0F 0XBD

字头	ID	数据长度	指令	参数	校验和
0XFF 0XFF	0X01	0X05	0X04	0X2A 0XFF 0X0F	0XBD

1号舵机应答包：0XFF 0XFF 0X01 0X02 0X00 0XFC

字头	ID	数据长度	工作状态	参数	校验和
0XFF 0XFF	0X01	0X02	0X00		0XFC

ACTION 指令包：0XFF 0XFF 0XFE 0X02 0X05 0XFA

字头	ID	数据长度	指令	参数	校验和
0XFF 0XFF	0XFE	0X02	0X05		0XFA

注：ACTION 指令包通过 ID254 广播指令发送，所以没有应答包数据返回。

2.4.6 同步写指令 SYNC WRITE

不同于 REG WRITE+ACTION 指令的是：SYNC WRITE 实时性比它更高。一条 SYNC WRITE 指令可一次修改多个舵机内存控制表的内容，而 REG WRITE+ACTION 指令是分步做到的。使用 SYNC WRITE 指令时，写入的数据长度和保存数据的地址必须相同，即必须执行相同的动作。简单的说就是不能同时控制一个舵机运动，另外一个舵机查询温度。只能是同时控制几个舵机运动，或者同时查询几个舵机的温度，以此类推。

功 能	用于同时控制多个舵机执行相同功能。(本指令数据顺序为 H-L)
ID	0XFE
长 度	(L + 1) * N + 4 (L: 发给每个舵机的数据长度, N: 舵机的个数)
指 令	0X83
参 数 1	写入数据的首地址
参 数 2	写入的数据的长度(L)
参 数 3	第一个舵机的 ID 号
参 数 4	写入第一个舵机的第一个数据
参 数 5	写入第一个舵机的第二个数据
...	
参 数 L+3	写入第一个舵机的第 L 个数据
参 数 L+4	第二个舵机的 ID 号
参 数 L+5	写入第二个舵机的第一个数据
参 数 L+6	写入第二个舵机的第二个数据
...	
参 数 2L+4	写入第二个舵机的第 L 个数据

示　例:用 SYNC WRITE 指令同时控制 0 号舵机 2000 毫秒运行到 180° 的位置,1 号舵机 3000 毫秒运行到 180° 的位置，4 号舵机以 4000 毫秒的时间运行到 0° 的位置。

分　析：控制几个舵机，所以我们采用广播 ID254（0XFE）。数据长度为 (L + 1) * N + 4，本例中写入的数据长度为 4，舵机个数是 3 个，所以指令数据长度为（4+1）*3+4=19，转换为 16 进制的话就是 0X13。舵机位置首地址为 0X2A,写入的数据长度为 0X04。所以形成以下内容(高位字节在前，低位字节在后)：

首地址、写入数据长度：0X2A 0X04

ID0：目标位置：0X07FF；运行时间：0X07D0

ID1：目标位置：0X07FF；运行时间：0X0BB8

ID4：目标位置：0X0000；运行时间：0X0FA0

依据以上分析，我们能得到以下指令包内容：

指令包 ：0XFF 0XFF 0XFE 0X13 0X83 0X2A 0X04 0X00 0XFF 0X07 0XD0 0X07 0X01 0XFF 0X07 0XB8 0X0B 0X04 0X00 0X00 0XA0 0X0F 0XE3

字头	ID	数据长度	指令	参数	校验和
0XFF 0XFF	0XFE	0X13	0X83	0X2A 0X04 0X00 0X07 0XFF 0X07 0XD0 0X01 0X07 0XFF 0X0B 0XB8 0X04 0X00 0X00 0X0F 0XA0	0XE3

2.4.7 突发写指令 BULK WRITE

功 能　　用于同时控制多个舵机与不连续内存地址。

长 度　　L1 + L2 + L3+ … + 2

指 令　　0X09

参数 1-1　　第一个舵机的 ID 号
参数 1-2　　第一个舵机写入第一组数据的首地址
参数 1-3　　第一个舵机写入的第一组数据的长度 L1
参数 1-4　　写入第一个舵机第一组数据的第一个数据
参数 1-5　　写入第一个舵机第一组数据的第二个数据
……
参数 2-1　　第一个舵机的 ID 号
参数 2-2　　第一个舵机写入第二组数据的首地址
参数 2-3　　第一个舵机写入的第二组数据的长度 L2
参数 2-4　　写入第一个舵机第二组数据的第一个数据
参数 2-5　　写入第一个舵机第二组数据的第二个数据
...
参数（1-L）　　第 L 个舵机的 ID 号
参数（1-L）+1　　第 L 个舵机写入第一组数据的首地址
参数（1-L）+2　　第 L 个舵机写入的第一组数据的长度 L3
参数（1-L）+3　　写入第 L 个舵机第一组的第一个数据
参数（1-L）+4　　写入第 L 个舵机第一组的第二个数据
...

参数（2-L）　　第 L 个舵机的 ID 号
参数（2-L）+1　　第 L 个舵机写入第二组数据的首地址

参数（2-L）+2　　第 L 个舵机写入的第二组数据的长度 L4
参数（2-L）+3　　写入第 L 个舵机第二组的第一个数据
参数（2-L）+4　　写入第 L 个舵机第二组的第二个数据
...

一条 BULK WRITE 指令可一次修改多个舵机不连续的控制表内容。

注意：本指令数据顺序为 L-H，先写低字节，再写高字节，切记！！

2.4.8 复位指令 RESET

功 能　　把内存控制表里的数据复位为出厂值默认值

长 度　　0X02

指 令　　0X06

参 数　　无

示　例：　让 1 号舵机恢复出厂设置

指令包：0XFF 0XFF 0X01 0X02 0X06 0XF6`

字头	ID	数据长度	指令	参数	校验和
0XFF 0XFF	0X01	0X02	0X06		0XF6

应答包：0XFF 0XFF 0X01 0X02 0X00 0XFC

字头	ID	数据长度	工作状态	参数	校验和
0XFF 0XFF	0X01	0X02	0X00		0XFC

2.5 OCS 模式内存控制表

机器人舵机本身的信息和控制参数形成了一张表，保存在其控制芯片的 RAM 和 EEPROM 区域。通过实时修改表里的内容，可以达到实时控制舵机的目的。这张表称之为内存控制表。

2.5.1 关于内存控制表的几点说明

2.5.1.1 关于 EEPROM 和 RAM

保存在 EEPROM 中的数据不会由于断电而改变，而保存在 RAM 中的数据在每次重新上电以后都会被重置，将不保存。

2.5.1.2 关于高低字节 L 和 H

当一个数据需要用到 16 位的时候就会产生高低字节，例如:我们的舵机能够进行 360°的伺服控制，通过前面的示例我们知道，本舵机 0-360°对应的 AD 值为 0-4095。

H L

4095 转化为十六进制 0X0FFF，也就是 0000 1111 1111 1111，其中红色字体部分为高字节 H，蓝色字体部分为低字节 L。从之前的示例以及内存表可以知道，我们在往内存表写参数时有时候先写低字节 L，有时候先写高字节 H。所以一定要注意 2.4.6 和 2.4.7 中写入顺序是不一样的。

2.5.2 OCS 模式内存控制表如下：

地址	命令项	读/写	初始值	存储区
0(0X00)	--	--	--	EEPROM
1(0X01)	--	--	--	
2(0X02)	--	--	--	
3(0X03)	固件版本(L)	读	--	
4(0X04)	固件版本(H)	读	--	
5(0X05)	舵机ID	读/写	1(0X01)	
6(0X06)	波特率	读/写	0(0X00)	
7(0X07)	应答延迟	读/写	0(0X00)	
8(0X08)	应答级别	读/写	1(0X02)	
9(0X09)	最小角度限制（L）	读/写	0(0X00)	
10(0X0A)	最小角度限制（H）	读/写	0(0X00)	

11(0X0B)	最大角度限制（L）	读/写	255(0XFF)	
12(0X0C)	最大角度限制（H）	读/写	15(0X0F)	
13(0X0D)	最高温度限制	读/写	80(0X50)	
14(0X0E)	最高输入电压	读/写	130(0X82)	
15(0X0F)	最低输入电压	读/写	70(0X46)	
16(0X10)	最大扭矩（L）	读/写	255(0XFF)	
17(0X11)	最大扭矩（H）	读/写	3(0X03)	
18(0X12)	马达驱动PWM相位	读/写	0(0X00)	
19(0X13)	卸载条件	读/写	37(0X25)	
20(0X14)	LED报警条件	读/写	37(0X25)	
21(0X15)	PID控制 P参数	读/写	15(0X0F)	
22(0X16)	PID控制 D参数	读/写	00(0X00)	
23(0X17)	PID控制 I参数	读/写	00(0X00)	
24(0X18)	舵机马达启动力(L)	读/写	0(0X00)	
25(0X19)	舵机马达启动力(H)	读/写	0(0X00)	
26(0X1A)	顺时针死区宽度	读/写	1(0X02)	
27(0X1B)	逆时针死区宽度	读/写	1(0X02)	
28(0X1C)	PID控制中积分限制（L）	读/写	0(0X00)	
29(0X1D)	PID控制中积分限制（H）	读/写	0(0X00)	
30(0X1E)	--	--	--	
31(0X1F)	--	--	--	
32(0X20)	--	--	--	
33(0X21)	输出轴中立点校正（L）	读/写	0(0X00)	
34(0X22)	输出轴中立点校正（H）	读/写	0(0X00)	
35(0X23)	运行模式	读/写	0(0X00)	
36(0X24)	保护电流	读/写	7(0X07)	
37-39	--	--	--	
40(0X28)	扭矩开关	读/写	0(0X00)	
41(0X29)	--	--	--	
42(0X2A)	目标位置（L）	读/写	--	
43(0X2B)	目标位置（H）	读/写	--	
44(0X2C)	运行时间（L）	读/写	0(0X00)	
45(0X2D)	运行时间（H）	读/写	0(0X00)	
46(0X2E)	运行速度（L）	读/写	0(0X00)	RAM
47(0X2F)	运行速度（H）	读/写	0(0X00)	
48(0X30)	锁定标志	读/写	1(0X01)	
49(0X31)	--	--	--	
50(0X32)	--	--	--	
51(0X33)	相对移动标志	读/写	0(0X00)	
52-55	--	--	--	

56(0X38)	当前位置（L）	读	?	
57(0X39)	当前位置（H）	读	?	
58(0X3A)	当前速度（L）	读	?	
59(0X3B)	当前速度（H）	读	?	
60(0X3C)	当前负载（L）	读	?	
61(0X3D)	当前负载（H）	读	?	
62(0X3E)	当前电压	读	?	
63(0X3F)	当前温度	读	?	
64(0X40)	REG WRITE标志	读	0(0X00)	
65(0X41)	--	--	--	
66(0X42)	舵机运转标志	读	?	
67(0X43)	当前目标位置（L）	读	?	
68(0X44)	当前目标位置（H）	读	?	
69(0X45)	当前电流（L）	读	?	
70(0X46)	当前电流（H）	读	?	

2.5.2 内存表详细描述如下：

地址：0X05

用于保存舵机 ID 号的地址，可以读写，默认值为 1(0X01)

地址：0X06

用于保存舵机波特率参数的地址，可以读写，默认值为 0（0X00），波特率为 1M。

Data	实际	目标	误差
0（0X00）	1M	1M	0.0%
1（0X01）	500000	500000	0.0%
2（0X02）	250000	250000	0.0%
3（0X03）	128000	128000	0.0%
4（0X04）	115107.9	115200	0.079%
5（0X05）	76923	76800	-0.16%
6（0X06）	57553.9	57600	0.008%
7（0X07）	38461.5	38400	-0.16%

地址：0X07

用于保存舵机应答延迟的地址，可以读写，默认值为 0(0X00)

当舵机收到一条需要应答的指令后，延迟多长时间应答可由您设置。时间范围：参数（0~255）*2US，若参数 100，即 200us 后应答。默认参数为 0，表示以最短的时间应答，由于舵机需要约 8us 的最小反应时间，所以实际最小应答时间为 8us。

地址：0X08

用于设定舵机应答级别的地址，可以读写，默认为参数 2， 舵机会对对所有指令返回。

地址0X10的参数	对应应答级别
0	只应答读指令和Ping指令
1	所有指令都返回应答包（广播除外）

地址：0X09～0X0C

设置舵机可运行的角度范围，可以读写。

最小角度限制和最大角度限制对目标位置有效。最小角度限制必须小于最大角度限制。

地址：0X0D

用于设定舵机最高工作温度的地址，可以读写，最高工作温度设定为 80℃。

地址：0X0E～0X0F

用于限定舵机输入电压上限和下限的地址，可以读写。

地址：0X10～0X11

用于设定舵机最大输出扭矩的地址，可以读写，1000 对应最大输出。

地址：0X13-0X14

用于设定舵机卸载条件的地址，可以读写

BIT	功能
BIT7	--
BIT6	--
BIT5	如果设置为1，则发生指超载时扭力输出降低--LED报警
BIT4	--
BIT3	如果设置为1，则发生过流时卸载扭力--LED报警
BIT2	如果设置为1，则发生过热时卸载扭力--LED报警
BIT1	如果设置为1，则角度传感器出错时卸载扭力--LED报警
BIT0	如果设置为1，则发生超过电压范围时卸载扭力--LED报警

以上若同时发生，遵行逻辑或的原则。LED 报警条件（0X14）设置为 0 关闭 LED，否则打开 LED

地址：0X15～0X17

用于 PID 控制系统的 P、D、I 参数的地址，可以读写。

P 参数增加可以提高舵机输出力度，但容易导致舵机过冲，或抖动，

D 参数可以抑制由于 P 参数提高造成的过冲现象

I 参数一般情况下不需要调节

地址：0X18～0X19

用于控制舵机马达驱动占空比，修改本参数可以改变马达驱动力

地址：0X1A～0X1B

顺时针和逆时针死区大小设定的地址顺时针与逆时针都设置为 1 则死区大小约为 0.087 度。

地址：0X1C～0X1D

PID 控制中积分最大值上限的设定地址。

地址：0X21～0X22

用于修正舵机的 0 点位置，0-2047 表示正方向，2048-4095 为负方向。

地址：0X23

运行模式，

模式	功能
模式 0	设置为 0，0-360° 伺服控制模式
--	--
模式 2	设置为 2，恒力矩输出模式，见 2.5.5 章节

地址：0X28

用于打开关闭舵机扭矩输出的地址。

地址：0X2A～0X2B

用于设定舵机目标位置的地址，想要舵机运行到某个位置，就想要在这两个地址写入相应的位置。0 到 4095（0XFFF）可用。

地址：0X2C～0X2D

用来设定舵机运行到目标位置时间参数的地址，可以读写。0-65535（0XFFFF）都可以使用，单位 1 毫秒。

如果它被设定成 0，这意味着舵机按照最大的速度运转。

比如，将它设置为 3000，那么时间是用 3 秒的时间到达目标位置。

地址：0X2E～0X2F:

用来设定舵机运行到目标位置速度参数的地址，可以读写。0-65535（0XFFFF）都可以使用，如果参数超过马达转速极限，则会以最快速度运行。

本参数的范围与取值根据以下运行模式而变化。

伺服模式

0-65535（0XFFFF）可以使用，单位为 1AD/毫秒

例如，将它设置为 1000，则转速为 1000AD/sec。

电机模式

0-65535（0XFFFF）可以使用，单位为 1AD/毫秒

如果取值在 0-32767 范围内，它将因设置到 0 而旋转到 CCW 方向而停止。

如果取值在 32767-65535 范围内，它将会因设置在 32768 而转向 CW 方向停止。

第十五字节变成位置字节来控制方向，如下示例：

0000 1011 1011 1000 顺时针 3000 的速度，换算成十六进制为 0X0BB8，按照 L-H 的写入顺序就是 B8 0B

1000 1011 1011 1000 逆时针 3000 的速度 换算成十六进制为 0X8BB8（0X0BB8+0X8000），按照 L-H 写入顺序就是 B8 8B

地址：0X30

锁功能地址

数值	功 能
0（0X00）	EEPROM 区域能被修改
1（0X01）	EEPROM 区域不能被修改.

注：频繁对舵机 EEPROM 进行写入操作会影响舵机寿命

地址：0X38～0X3F

当前位置，速度，温度，电压，负载等可以反馈信息的参数地址，只读。

地址：0X40

若有 REG WRITE 指令等待执行，则显示为 1，当 REG WRITE 指令执行完毕后显示为 0。

地址：**0X42**

舵机已经到达目标位置，则显示 0，如果还没到打目标位置，则显示为 1。

地址：**0X43、0X44**

当前指令发送给舵机的目标位置。

地址：**0X45、0X46**

当前舵机的工作电流。

地址：**0X47、0X48**

多圈模式下舵机已经运行圈数。

2.5.5 OCS 模式下的恒扭力模式

本舵机可以切换为恒力矩输出模式，可用于轮子，履带等需要恒扭矩输出的执行机构上。把运行模式（0X23）设置为 2，再给一个时间（0X2C～0X2D），舵机就以恒扭力输出转动起来，扭力大小和方向的控制方式，如下表所示：

BIT	10	9~0
Value	0/1	时间值

地址 0X2C～0X2D：Bit10 是方向位，0 为逆时针方向，1 为顺时针方向。Bit0~9 为扭力大小，输入范围 0~1000。

0000 0001 1111 0100 红色位表示方向，蓝色表示扭力 500，换为 16 进制为：01F4

0000 0111 1110 1000 红色位表示方向，蓝色表示扭力 1000，换为 16 进制为：07E8

示 例：让 1 号舵机以 500 的扭力逆时针转动

切换运行模式：FF FF 01 04 03 23 02 D2

运转扭力及方向：FF FF 01 05 03 2C F4 01 D5

警告：

1、本产品为高精度产品，请勿人为大力转动摆臂，以免产品内部损坏

2、本产品为大扭矩舵机，使用时务必小心谨慎，防止不慎造成人身伤害

3、切记不要在舵机工作时再向总线上增加舵机

4、本产品为类似机电类产品，所以尽量不要超负荷运转，合理运行转矩≈1/3 堵转扭矩

5、请勿超压使用，否则容易导致产品损坏