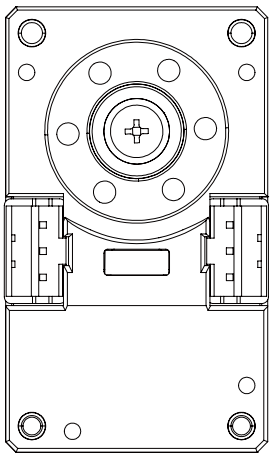
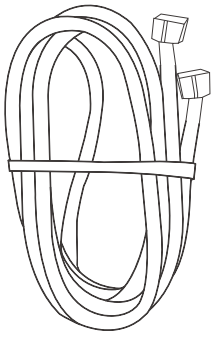
无刷数字舵机J288/S288使用手册

一、数字舵机包装清单



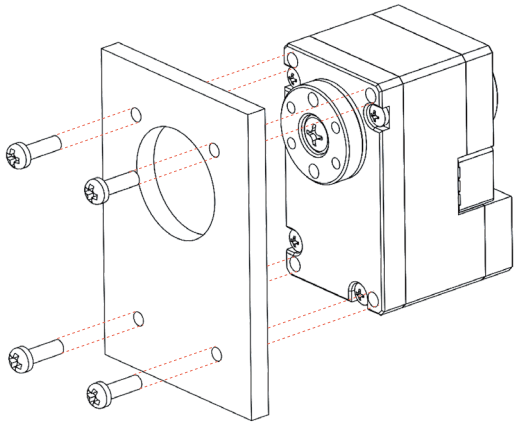
数字舵机*1



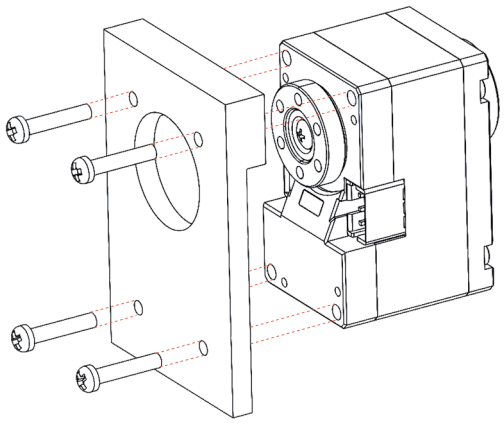
PH 2.0线缆*1

二、数字舵机安装说明

固定板需用户自行设计准备，固定螺钉规格为M2。

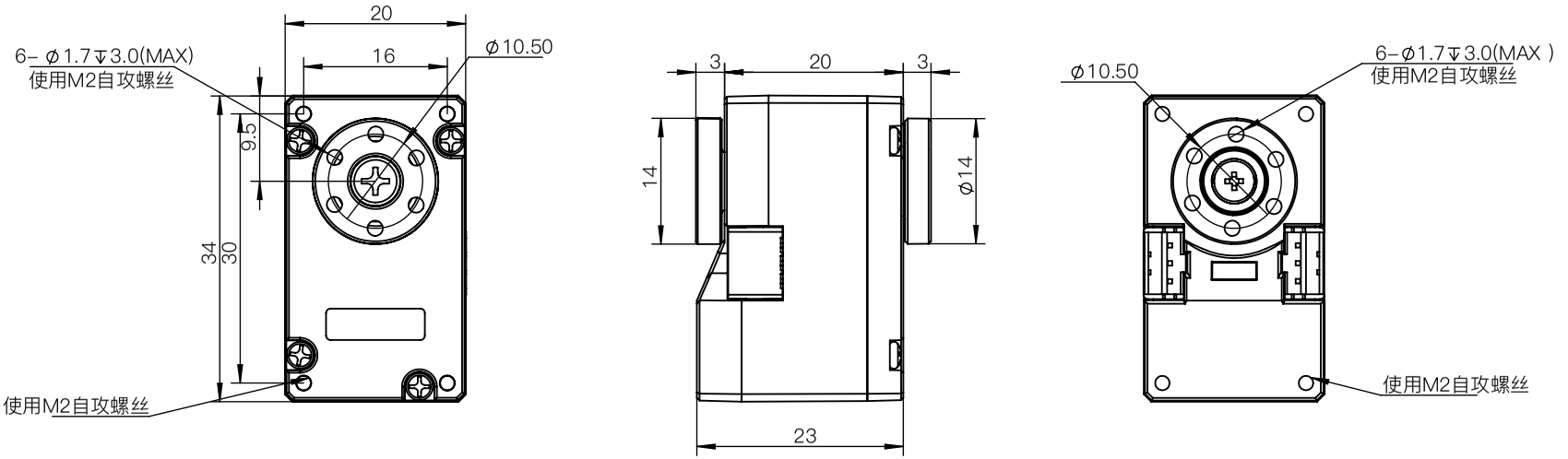


正面固定示意图

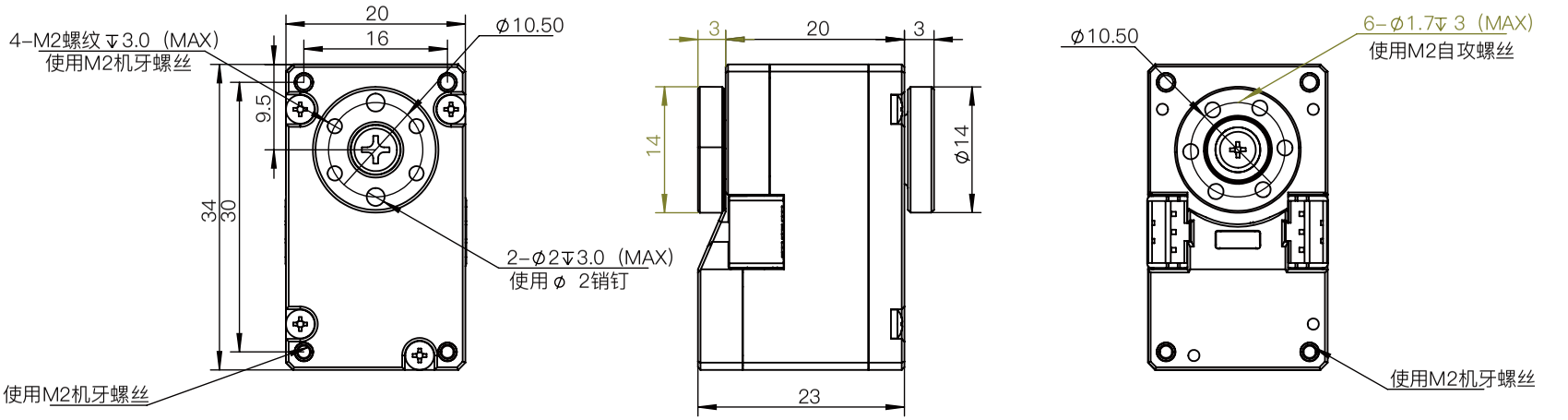


背面固定示意图

S288:

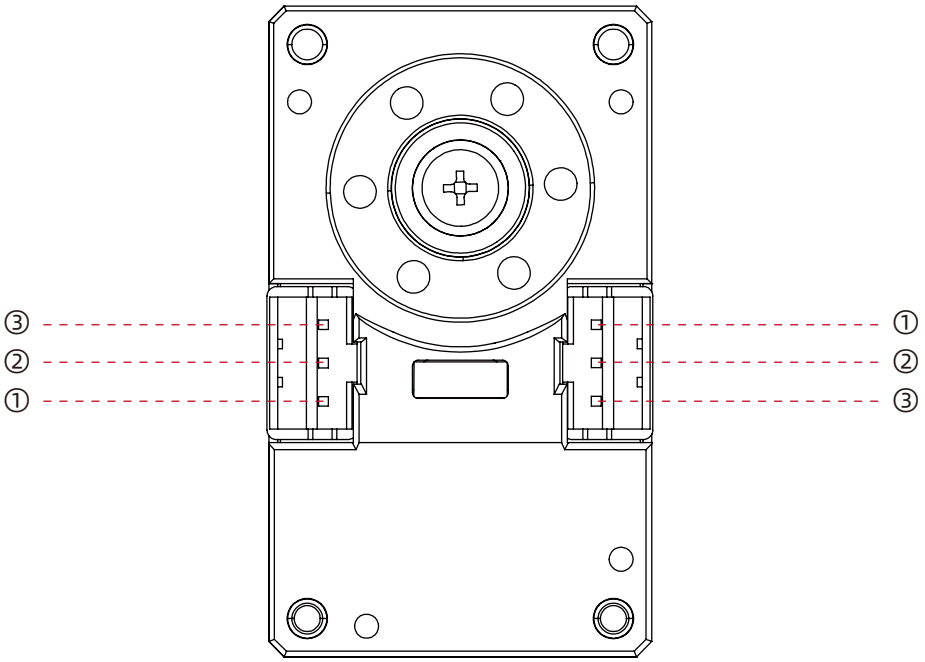


J288:



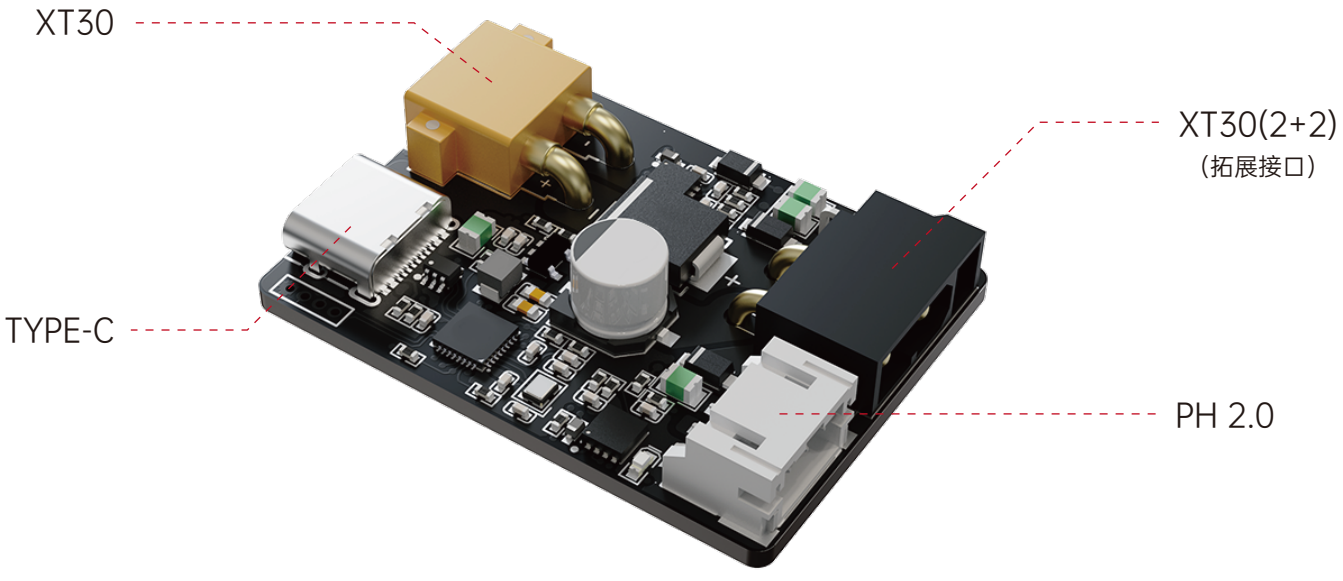
数字舵机尺寸图

三、数字舵机的接口说明



管脚序号	信号	描述
①	SIGNAL	信号通讯
②	VCC	12V /25.2V供电
③	GND	地线

四、单总线串口转USB板接口说明



五、数字舵机编码器概述

编码器 (encoder) 是用来测量旋转角度的传感器。编码器分为增量编码器、多圈绝对位置编码器与单圈绝对位置编码器等多种。我们在此只详细讨论数字舵机实际应用的单圈绝对位置编码器。

数字舵机的单圈绝对位置编码器安装在舵机转子上。对于单圈绝对位置编码器（以下简称编码器），可以将它当做一个“时钟表盘”。我们每次看表的时候，都可以读到当前的日期和时间，例如 4 月 1 日 23 点。如果时间经过了 2 个小时，那么时钟转过了 4 月 1 日 24 点，日期会增加 1 天变成 4 月 2 日，时间重新从 0 点开始计时，变成了 4 月 2 日 1 点。为了方便计算经过的时间，我们也可以说现在是 4 月 1 日 25 点。看上去 25 点超过了一天 24 个小时的范围，实际上是因为日期增加了 1 天。

单圈绝对位置编码器也是同样的道理。每次开机上电后，我们的转子可能处于任意位置，编码器会告诉我们转子所处的角度位置（0 至 2π 之间的某个值）。如果转子旋转转过了 2π 这个角度位置，编码器也能够记录转过的圈数增加了 1 ，从而输出一个超出 0 至 2π 范围的角度位置。看上去编码器也能够输出超过一圈的角度位置，那么为什么叫它“单圈”绝对位置编码器呢？原因在于这种编码器在断电之后不能够储存之前旋转的圈数，我们以下面这个例子来说明。

假设当前编码器输出的角度值为 2.3π ，这意味着编码器在开机之后经过了 2π ，旋转圈数从 0 变为 1，同时编码器当前位于 0.3π 这个位置。此刻我们将编码器关机，不做任何旋转，再将编码器开机，那么此时编码器输出的角度值就会变为 0.3π 。因为关机之后旋转圈数重置为 0，所以编码器只会输出当前位置 0.3π 。

六、数字舵机的混合控制

无刷数字舵机 J288/S288 作为一个高度集成的动力单元，其内部已经封装了舵机底层的控制算法。与其他舵机的控制方式不同，本舵机使用的控制方式为机器人领域的混合控制。

作为用户，只需要给数字舵机发送相关的命令，舵机就能完成从接收命令到关节力矩输出的全部工作。

对于舵机的底层控制算法，唯一需要的控制目标就是输出力矩。可是对于机器人，我们通常需要给关节设定位置、速度和力矩。这时就需要对数字舵机进行混合控制。

宇树科技的数字舵机包含如下 5 个控制指令：

1. 前馈力矩：τ_(ff)；

2. 期望角度位置：p_(des)；

3. 期望角速度：ω_(des)；

4. 位置刚度：k_p；

5. 阻尼系数：k_d；

在数字舵机的混合控制中，使用 PD 控制器将舵机在输出位置的偏差反馈到力矩输出上：

$$\tau=\tau\_ff+k\_p\times(p\_des-p)+k\_d\times(\omega\_des-\omega)$$

式中，τ 为数字舵机的舵机转子输出力矩，p 为舵机转子的当前角度位置，为 ω 舵机转子的角速度。在实际使用数字舵机时，需要注意将舵机输出端的控制目标量与发送的舵机转子的指令进行换算。

七、数字舵机的线路连接

我们使用单总线 (1-Wire) 作为物理层。物理层是指使用何种物理现象来表达和传输信息。实际上，可以将单总线视为一种基于单线通信的串行接口。

单总线仅利用一根数据线（以及公共地线）进行双向数据传输。该总线采用特殊的通信时序，在同一根线上实现设备数据传输。其单线连接方式极大简化了线缆连接，提高了硬件连接的可靠性，并降低了布线成本。

单总线信号采用和 TTL 相同的串口时序表示逻辑电平，由于单总线仅使用一根数据线进行双向通信，因此同一时刻总线上只能进行一个方向的通信，即半双工通信。总线通常由一个主机（上位机）控制通信时序。主机发送的指令中包含目标设备地址信息，总线上所有设备均会接收该指令，但只有地址匹配的设备会响应并回复数据，从而完成一轮通信。

J288/S288 舵机最大支持一条总线上连接 15 个舵机（地址 0~14，15 为广播地址）。总线上不得有地址相同的舵机，否则整个总线通信将出现异常。

为了将运动控制指令发送给舵机，我们需要通过串口将指令下发。舵机通过单总线接口与上位机通信，通信速率（波特率）固定为：6Mbps（8N1）。

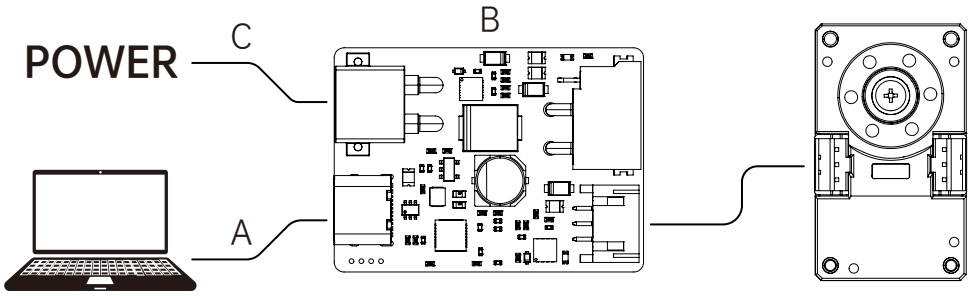


图 2 数字舵机线路连接

如图 2 所示，接口采用 PH 2.0 线缆连接至 B（单总线串口转 USB 模块），再通过 A（USB–TypeC 线缆）连接到电脑，通过 C（XT30 电源线）进行供电。数字舵机包装内仅包含 PH 2.0 线缆，其他需另行购买。

在使用个人电脑控制舵机时，需将单总线接口通过 USB 转单总线适配器连接至上位机。接通 12 V 直流电源后，舵机绿色指示灯开始闪烁，表示舵机已正常开机。

八、舵机配置及快速使用

通过舵机调试助手，您可以对舵机执行以下配置操作：

- 查询及修改 ID
- 开启自动引导功能
- 更新固件
- 查询版本号
- 恢复舵机模式
- 临时关闭引导

通过舵机调试助手，您可以快速使用舵机，实现以下功能

- 舵机运动控制
- 数据曲线绘制

具体操作方法详见《电机调试助手使用指南》。

九、舵机协议

考虑到部分用户可能会使用特殊平台来控制舵机，在此我们将讲解如何编写自定义的舵机控制程序。根据本节介绍的方法，读者可以在任何满足硬件要求的平台上向舵机发送控制命令并接收舵机状态反馈。协议中的所有参数均为转子端数据，减速比为 288.35，若需将转子数据换算为输出端数据，则需乘以或除以该减速比。

J288/S288 舵机采用串口通信，通信标准为单总线（1–Wire），波特率为 6.0 Mbps。串口的数据位为 8 bit，无奇偶校验位，停止位为 1 bit 。需要注意的是为了提高舵机的通信频率，我们使用了 6.0 Mbps 这一很高的波特率，用户需要检查自己的硬件是否支持这么高的波特率。如果您的硬件无法支持该波特率，可以使用 Unitree 的 USB 转单总线（1–Wire）模块。

在控制舵机时，我们会通过串口给舵机发送一个长度为 20 字节的命令，之后舵机会返回一个长度为 26 字节的信息 。如果不给舵机发送命令，那么舵机也不会返回状态。舵机发送命令格式与返回格式见下面结构体，结构体详细说明了各个字节的含义，下面我们解释其中部分细节。

首先，在发送给舵机的命令中，第 5 和第 6 字节表示舵机前馈力矩τ_ff，其中 256000 对应 1 牛・米（NM）。这相当于我们对前馈力矩值乘以了 256000，然后将其赋值给一个 2 字节的带符号短整型（signed short int）变量。在此赋值过程中会进行强制取整，从而使我们能够仅用两个字节来传输前馈力矩τ_ff 数据。当舵机接收到此数据后，只需将其除以 256000，即可还原前馈力矩τ_ff 的实际数值。这种方法虽然会损失一定精度，但对于实际应用场景而言完全满足需求。

此外，需要注意的是，对于一个 2 字节（16 位）的变量，其中 1 位用作符号位表示正负，因此只有 15 位可用于表示数值大小。这意味着命令中τ_ff 的数值不能 2^15 超过特定上限。考虑到我们曾对原始数据τ_ff 乘以了 256000，因此其绝对值存在相应的上限值：

$$|\tau\_ff| < 2^{15}/256000 = 0.128$$

同时需要注意的是，由于赋值过程中存在强制取整操作，τ_ff 因此数值越大，保存的小数精度就越低。结构体变量注释中提到的 ”256000 表示 1” 实际上是指上文所述的乘以 256000 的操作，其他变量中类似的 ”xx 表示 1” 描述也遵循相同的原理。

并且，对于 2 个字节的 τ 变量来说，它的低位在前，即第 5 字节，高位在后，即第 6 字节。在发送命令和接收状态的末尾，我们可以看到一个 4 字节的 CRC32 校验。在命令发送之前，我们会计算这些命令字节的发送前 CRC 校验值，并且和命令一起发送给舵机。当舵机收到命令之后，还会根据收到的命令计算发送后 CRC 校验值。

如果数据传输过程中没有发生任何错误，那么发送前 CRC 校验值等于发送后 CRC 校验值。如果在数据传输过程中出现了数据错误，那么发送后 CRC 校验值的计算结果就与发送前 CRC 校验值不相等，舵机就能够知道数据发生了损坏，从而帮助我们避免错误数据。

```
// 舵机控制命令数据包 20 字节
typedef struct
{
uint8_t head[2]; // 帧头 0xFE 0xEE 参与 CRC
uint8_t id : 4; // 舵机 ID: 0–14, 15 表示向所有舵机广播数据 ( 此时无返回 )
uint8_t status : 3; // 工作模式 : 0. 锁定 1.FOC 闭环
uint8_t timeout : 1; // Master–>Motor: 0. 禁用超时保护 1. 开启（默认 1s 超时）
uint8_t res; // 保留位
int16_t tor_des; // 期望转子输出扭矩 unit: N.m 256000 表示 1NM
int16_t spd_des; // 期望转子输出速度 unit: rad/s 2.56/2π表示 1rad/s
int32_t pos_des; // 期望转子输出位置 unit: rad 32768/2π表示 1rad
int16_t k_pos; // 期望转子刚度系数 unit: N.m/rad 1280000 表示 1
int16_t k_spd; // 期望转子阻尼系数 unit: N.m/(rad/s) 128000000 表示 1
uint32_t CRC32; // CRC32
} ControlData_t;

// 舵机控制模式
typedef struct
{
uint8_t id : 4; // 舵机 ID: 0–14, 15 表示向所有舵机广播数据 ( 此时无返回 )
uint8_t status : 3; // 工作模式 : 0. 锁定 1.FOC 闭环
uint8_t timeout : 1; // Master–>Motor: 0. 禁用超时保护 1. 开启（默认 1s 超时）
// Motor–>Master: 0. 没有超时 1. 触发超时保护 ( 需要控制位发 0 清除 )
} RIS_Mode_t;
```

```
// 2- 舵机返回数据 26 字节
typedef struct
{
uint8_t NoUse[2]; // 通信中不存在(结构体 4 字节对齐), 在发送通信数据时必须去除该数据位
uint8_t head[2]; // 帧头 0xFC 0xEE 不参与 CRC
RIS_Mode_t mode; // 舵机控制模式
int8_t temp; // 壳体温度 : -128~127℃
uint8_t sensor; // 绕组温度 : 0-255℃
uint8_t vol; // 舵机端电压 0-127.5V 255 表示 127.5V
int16_t torque; // 当前转子扭矩 256000 表示 1NM
int16_t speed; // 当前转子速度 2.56/2π表示 1rad/s
int32_t pos; // 当前转子位置 32768/2π表示 1rad
uint32_t MError; // 舵机错误码 : 0x0. 正常 bit0 过流 ,...
uint16_t OutPos : 13; // 输出端传感器 2^13 表示 1 圈
uint16_t ExFlag : 3; // 警告码
uint16_t res; // 保留位
uint32_t CRC32; // CRC32
} MotorData_t;

// CRC32 校验
uint32_t crc32_core(uint32_t *ptr, uint32_t len)
{
    uint32_t xbit = 0;
    uint32_t data = 0;
    uint32_t CRC32 = 0xFFFFFFFF;
    const uint32_t dwPolynomial = 0x04c11db7;
    for (uint32_t i = 0; i < len; i++)
    {
```

```
        xbit = 0x80000000;
        data = ptr[i];
        for (uint32_t bits = 0; bits < 32; bits++)
        {
            if (CRC32 & 0x80000000)
            {
                CRC32 <<= 1;
                CRC32 ^= dwPolynomial;
            }
            else
                CRC32 <<= 1;
            if (data & xbit)
                CRC32 ^= dwPolynomial;

            xbit >>= 1;
        }
    }
    return CRC32;
}
```

十、舵机故障码表

故障码	位(bit)	说明	故障码	位(bit)	说明
0x01	0	过流	0x02	1	瞬态过压
0x04	2	持续过压	0x08	3	欠压
0x10	4	芯片过热	0x20	5	MOS 过热 / 冷
0x40	6	MOS 温度异常	0x80	7	壳体过热 / 冷
0x100	8	壳体温度异常	0x200	9	绕组过热
0x400	10	转子编码器 1 错误	0x800	11	转子编码器 2 错误
0x1000	12	输出编码器错误	0x2000	13	储存数据错误
0x4000	14	异常复位	0x8000	15	等待解锁
0x10000	16	驱动验证错误	0x20000	17	标定模式
0x40000	18	通信校验错误	0x80000	19	驱动版本过低
0x100000	20	固件型号错误	0x200000	21	编码器角度异常
0x...	...	未知错误			

表1.舵机故障码表

十一、警告码表

故障码	位(bit)	说明	故障码	位(bit)	说明
0x01	0	低压电源异常	0x...	...	未知警告

表2.舵机警告码表