

# [writing]B1电机简要手册

## B1电机简要手册

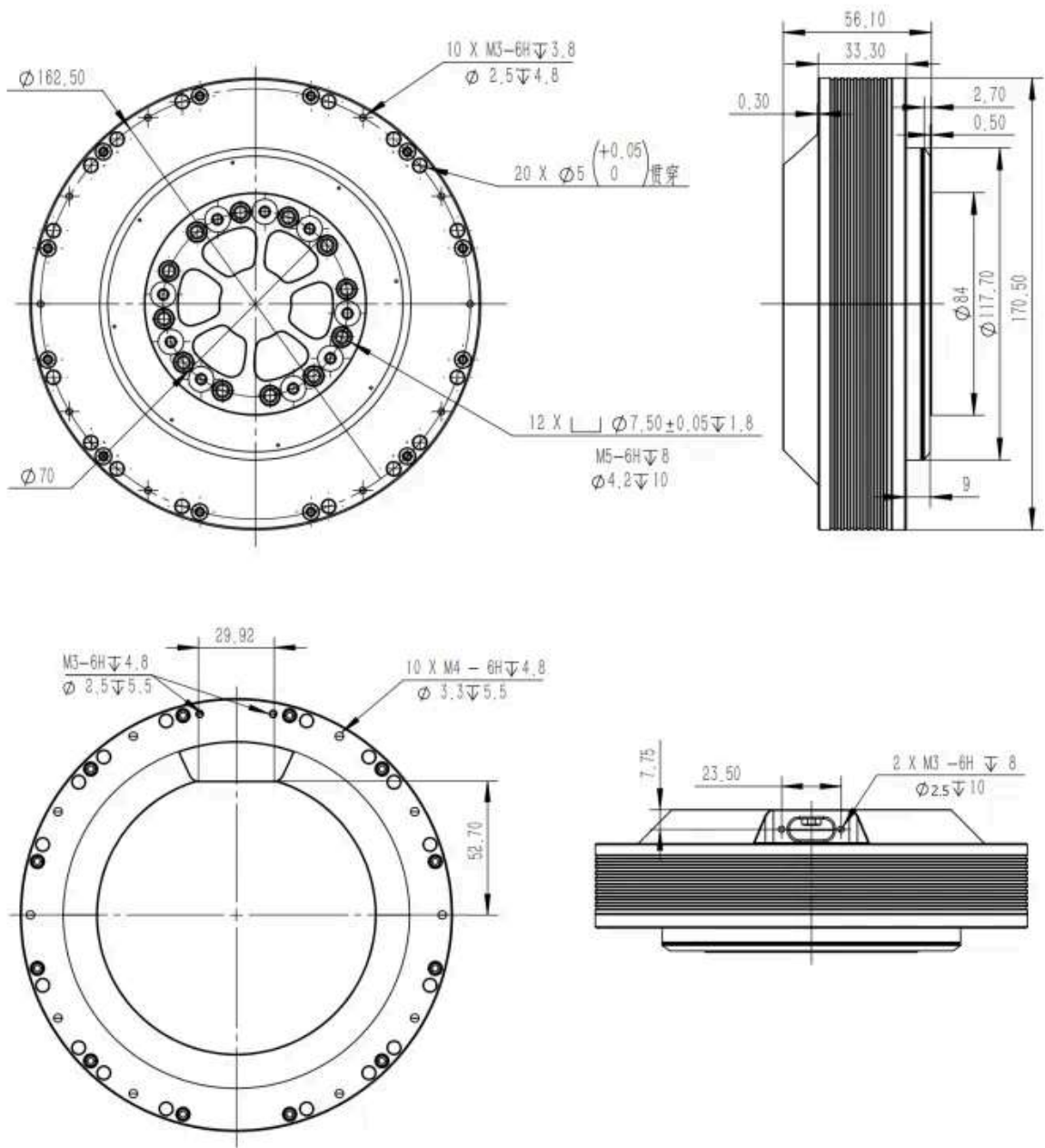
### 1. B1电机基本信息

B1电机是一款高度集成的永磁同步电机，电机由定子、转子、驱动板、减速器、编码器、轴承构成。整体外观如下图：



B1电机可以选择最高IP68的防护等级，通过对防护、专业、高效三大概念的延伸，让机器人能够更好地走进复杂环境，在更多场景中去拓展机器人的应用可能性。更灵活、更便捷、更高效的机器人，才能慢慢走进丰富多样的特定工业场景，逐渐实现例如户外巡检、农业采摘等半结构环境乃至非结构环境下的开拓，像人一样去探索这个大千世界。而防水防尘，无疑是一个至关重要的开始。

#### 1.1 B1电机装配图



B1电机装配图

推荐使用电机一圈的过孔来安装电机。

需要注意的是，为了达到防水防尘的目的，需要用定制的XT30（2+2）防水接头来连接电机，并且保证安装好O圈、接头安装到位，最好根据孔位制作线卡卡牢。

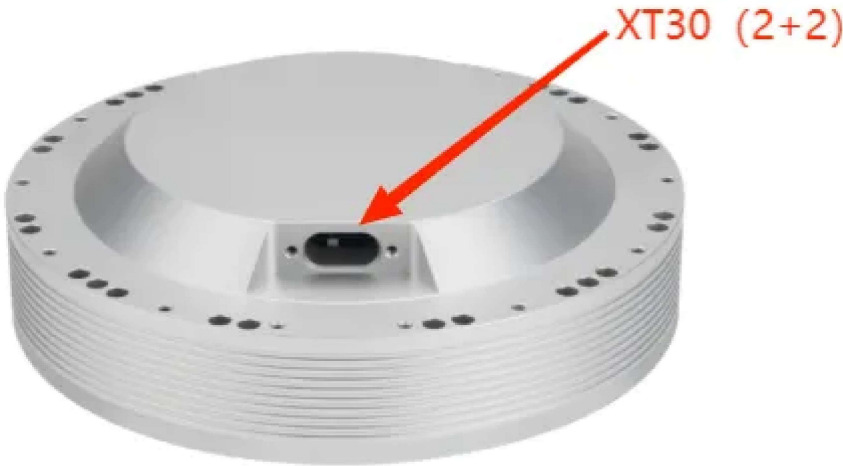


1.2 B1电机参数表

参数	数据	参数	数据
重量	1740g	工作电压	36-64VDC，额定电压58.5'
通讯方式	高速485	通讯控制频率	1KHz
最大瞬时扭矩	140N.m（瞬时） 130N.m（持续时间小于2秒）	额定扭矩	约25N.m
最大瞬时电流	13A（母线）	最大转速	190rpm（58VDC，输出端
减速比	8.66：1	扭矩常数	2.165N.m/A
温度传感器	有	编码器	15bit 转子 单圈绝对值编码
控制指令	力矩，角度，角速度，刚度，阻尼	状态反馈	力矩，角度，角速度，角加
防护等级	可选IP68		

2. B1电机硬件介绍

2.1 B1电机电气接口

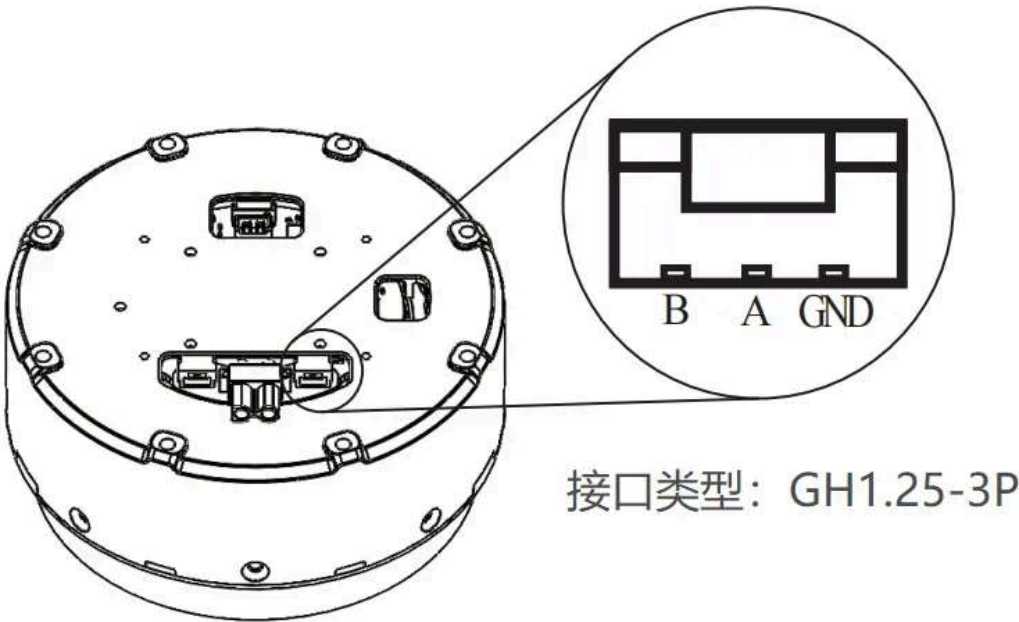


B1电机电气接口

B1电机电气接口定义如图所示，仅有一个定制的XT30（2+2）防水接头输入接口，电机工作电压是36~64VDC，推荐使用58VDC供电，接口上的两个小口就是高速485接口，单个总线回路可以串联3个电机。

2.2 A1电机 RS-485 接口类型及线序

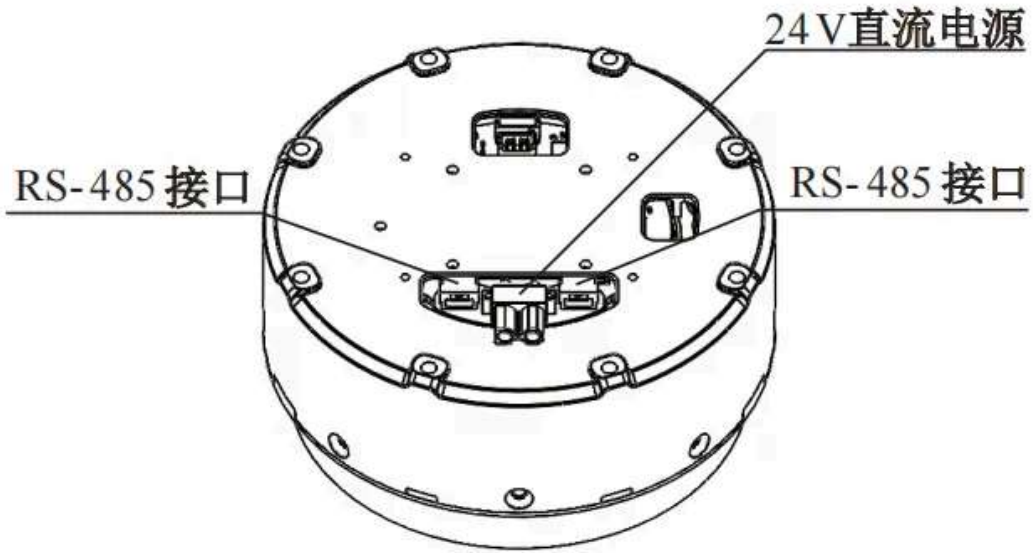
考虑到用户需要自行装配电机、链接，我们提供电机电气接口的类型和线序。



A1电机RS-485接口线序

2.3 A1电机的线路连接

A1电机采用 RS-485 接口与上位机经行通信，波特率为4.8 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电机的通信频率，我们使用了4.8 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提供的USB转 RS-485 模块。

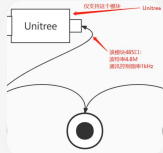


A1电机提供了两个 RS-485 接口，且这两个 RS-485 接口完全等价，因此可以使用 RS-485 线将电机串联（最多串联 3 个）同时控制。

插拔 RS-485 接口时，需要注意卡扣，不要硬拉硬扯，以免损坏接口。

在通过自己的电脑控制关节电机时，为了将指令从上位机发送到关节电机，需要将 RS-485 接口通过 USB 转 RS-485 转接口连接到上位机。在接通 24V 直流电源后，电机绿色指示灯开始闪烁，说明电机已开机。

## 2.4 宇树电机的通信控制频率

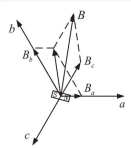


### 电机的通讯控制频率

电机的通讯控制频率一般来说，大家理解的通讯控制频率，是指一条485链路上的通讯控制频率...  
宇树科技开发知识库

## 3. A1电机软件介绍

### 3.1 电机控制原理



### 电机及其控制的基础概念

电机及其控制的基础概念四足机器人上安装有12个关节电机，这些关节电机是机器人实现复杂运...  
宇树科技开发知识库

### 3.2 A1电机SDK包

由于所有的电机底层控制算法都已经整合在电机内部，因此上位机（通常为用户的计算机）只需要完成上层控制和 RS-485 串口的数据收发。为了方便用户对关节电机的操作，宇树科技提供了 USB 转 RS-485 模块及支持该模块 RS-485 串口收发的软件开发工具包（Software Development Kit，以下简称 SDK）。

SDK包支持以下平台和系统：支持Ubuntu 18.04、Ubuntu 20.04，仅提供C++版本示例。

我们提供了测试程序，用户仿照示例就能完成对电机的控制。

SDK包：[UnitreeMotorSDK\\_A1\\_Sup221201.zip \(51 KB\)](#)

SDK包中，电机控制和状态接收相关的命令已经封装好，可以查看/include/unitreeMotor/unitreeMotor.h头文件来了解。其中主要是电机控制结构体 MOTOR\_send和电机状态接收结构体MOTOR\_recv。

MOTOR\_send

▼ Plain Text 复制代码

```
1 struct MOTOR_send{
2     // 定义 发送格式化数据
3     MasterCmdDataV3 motor_send_data; //电机控制数据结构体，详见motor_msg.h
4     MotorType motorType = MotorType::A1;
5     int hex_len = 34; //发送的16进制命令数组长度，34
6     // long long send_time; //发送该命令的时间，微秒(us)
7     // 待发送的各项数据
8     unsigned short id; //电机ID，0,1,2
9     unsigned short mode; //0:空闲，5:开环转动，10:闭环FOC控制
10    //实际给FOC的指令力矩为：
11    //K_P*delta_Pos + K_W*delta_W + T
12    float T; //期望关节的输出力矩（电机本身的力矩）（Nm）
13    float W; //期望关节速度（电机本身的速度）（rad/s）
14    float Pos; //期望关节位置（rad）
15    float K_P; //关节刚度系数
16    float K_W; //关节速度系数
17    COMData32 Res; // 通讯 保留字节 用于实现别的一些通讯内容
18 };
```

id: 电机ID，可以为0, 1, 2;

mode: 电机运行模式，0:空闲, 5:开环转动, 10:闭环FOC控制

T、W、Pos、K\_P、K\_W: 对应 mode = 10 时，电机的控制参数，原理详见3.1。

MOTOR\_recv

```
1 struct MOTOR_recv{
2     // 定义 接收数据
3     ServoCmdDataV3 motor_recv_data;    //电机接收数据结构体，详见motor_msg.h
4     MotorType motorType = MotorType::A1;
5     int hex_len = 78;                  //接收的16进制命令数组长度，78
6     // long long resv_time;            //接收该命令的时间，微秒(us)
7     bool correct = false;              //接收数据是否完整（true完整，false不完整）
8     //解读得出的电机数据
9     unsigned char motor_id;            //电机ID
10    unsigned char mode;                 //0:空闲，5:开环转动，10:闭环FOC控制
11    int Temp;                           //温度
12    unsigned char MError;               //错误码
13
14    float T;                            // 当前实际电机输出力矩
15    float W;                            // 当前实际电机速度（高速）
16    float LW;                           // 当前实际电机速度（低速）
17    int Acc;                            // 电机转子加速度
18    float Pos;                           // 当前电机位置（主控0点修正，电机关节还是以绝对位置为准）
19
20    float gyro[3];                      // 电机驱动板6轴传感器数据
21    float acc[3];
22 };
```

### 3.3 移植到其他平台

关于移植到其他平台，只要通过控制平台可以输出 4.8M 的波特率信号就是可以控制的，客户可以根据通信协议自行开发，宇树不提供支持。



#### 在其他上位机平台控制宇树关节电机

在其他上位机平台控制宇树关节电机考虑到部分用户会使用特殊的上位机平台来控制电机，在此...  
宇树科技开发知识库

## 4. 手把手教你使用A1电机SDK

### 4.1 查看串口名

将 USB 转 RS-485 模块连接在上位机上时，上位机会为这个串口分配一个串口名。在 Linux 系统中，这个串口名一般是以 “ttyUSB” 开头（在 Windows 系统中，串口名往往以 “COM” 开头）。

在 Linux 系统中，一切外接设备都是以文件形式存在的。USB 转 RS-485 模块也可以被视为 /dev 文件夹下的一个 “文件”。打开任意一个终端窗口（在 Linux 下快捷键为 Ctrl+Alt+t 组合键），

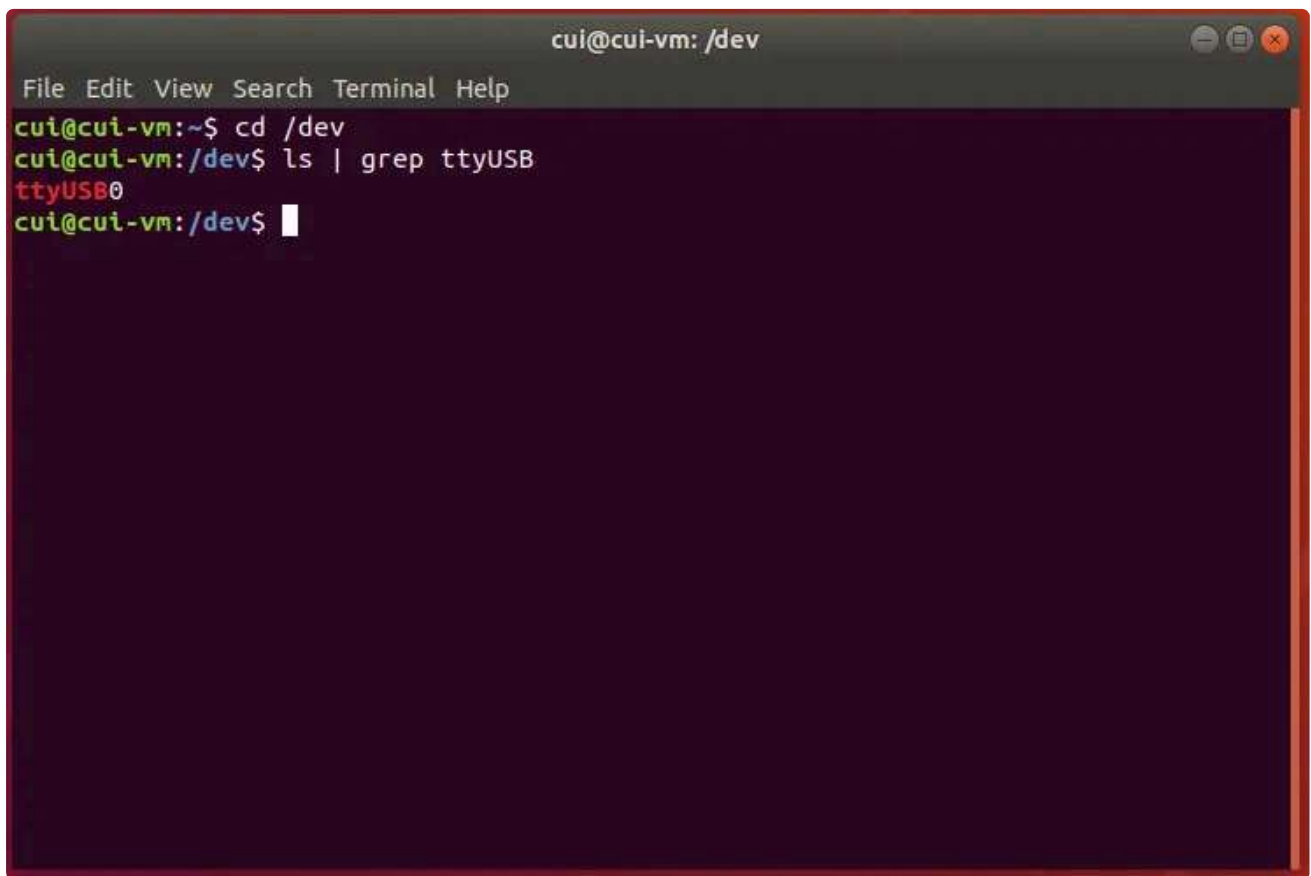


运行如下命令：

▼ Plain Text 复制代码

```
1 cd /dev
2 ls | grep ttyUSB
```

其中 `cd /dev` 命令将当前文件夹切换为 `/dev`，`ls | grep ttyUSB` 命令显示当前文件夹下所有文件名包含 `ttyUSB` 的文件。运行如上命令后，即可得到上位机当前连接的串口名。例如下图中所示，当前上位机连接的串口名为 `ttyUSB0`。考虑到串口所在的文件夹路径，其完整的串口名为 `/dev/ttyUSB0`。



## 4.2 修改电机ID

每一个电机都需要分配一个 ID，同时上位机发送的每一条控制命令也包含一个 ID，电机只会执行 ID 与自己一致的控制命令。因此，当多个关节电机串联在同一条 RS-485 线路中时，为了分别控制其中的每一个电机，必须给每一个电机分配一个唯一的 ID。

在此说明一下电机修改 ID 的流程。首先通过上位机的广播模式，将 RS-485 串口下的所有关节电机切换到修改 ID 模式。此时所有电机的输出轴都变成了电子棘轮，所谓电子棘轮，即在旋转输出轴时，能感受到类似棘轮的明显的顿挫感。这时通过旋转输出轴，就能够将电机的 ID 设置为对应的值。最后再由上位机给电机发送保存 ID 的指令，即可完成电机 ID 的修改。

为了在上位机发送上述指令，SDK 提供了电机 ID 修改程序 `changeID.cpp`。该程序在 SDK 中 `src` 目录下，需要先编译再使用。在 SDK 目录下打开终端，执行如下命令编译：

3790 字



▼ Plain Text 复制代码

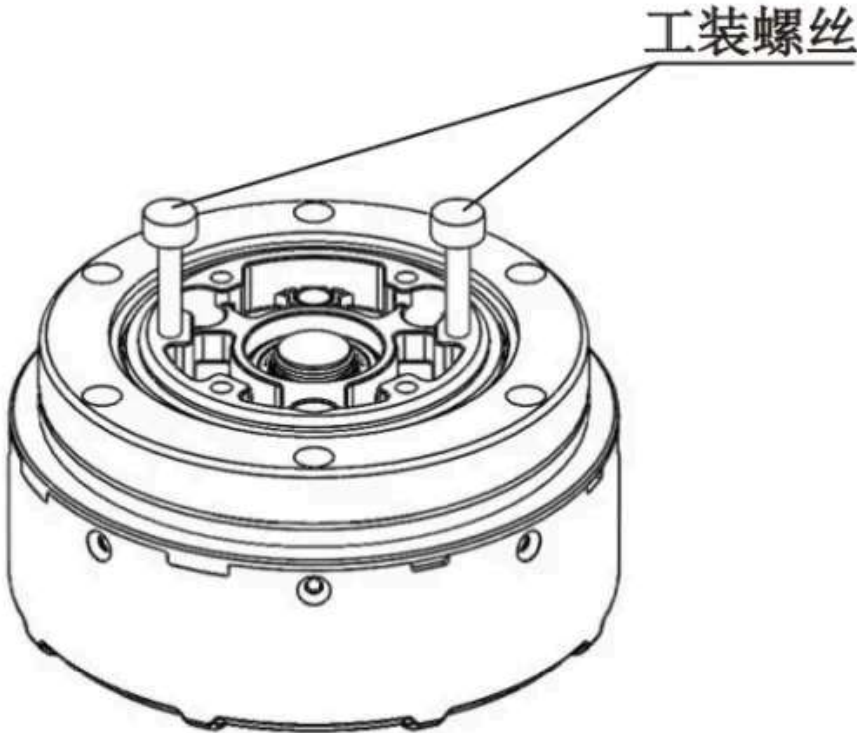
```
1  mkdir build
2  cd build
3  cmake ..
4  make
```

编译完之后，会在build目录下看到可执行文件changeID，下面我们来演示如何修改电机的 ID 。  
首先在管理员权限下运行build目录下的可执行文件changeID：

▼ Plain Text 复制代码

```
1  sudo ./changeID
```

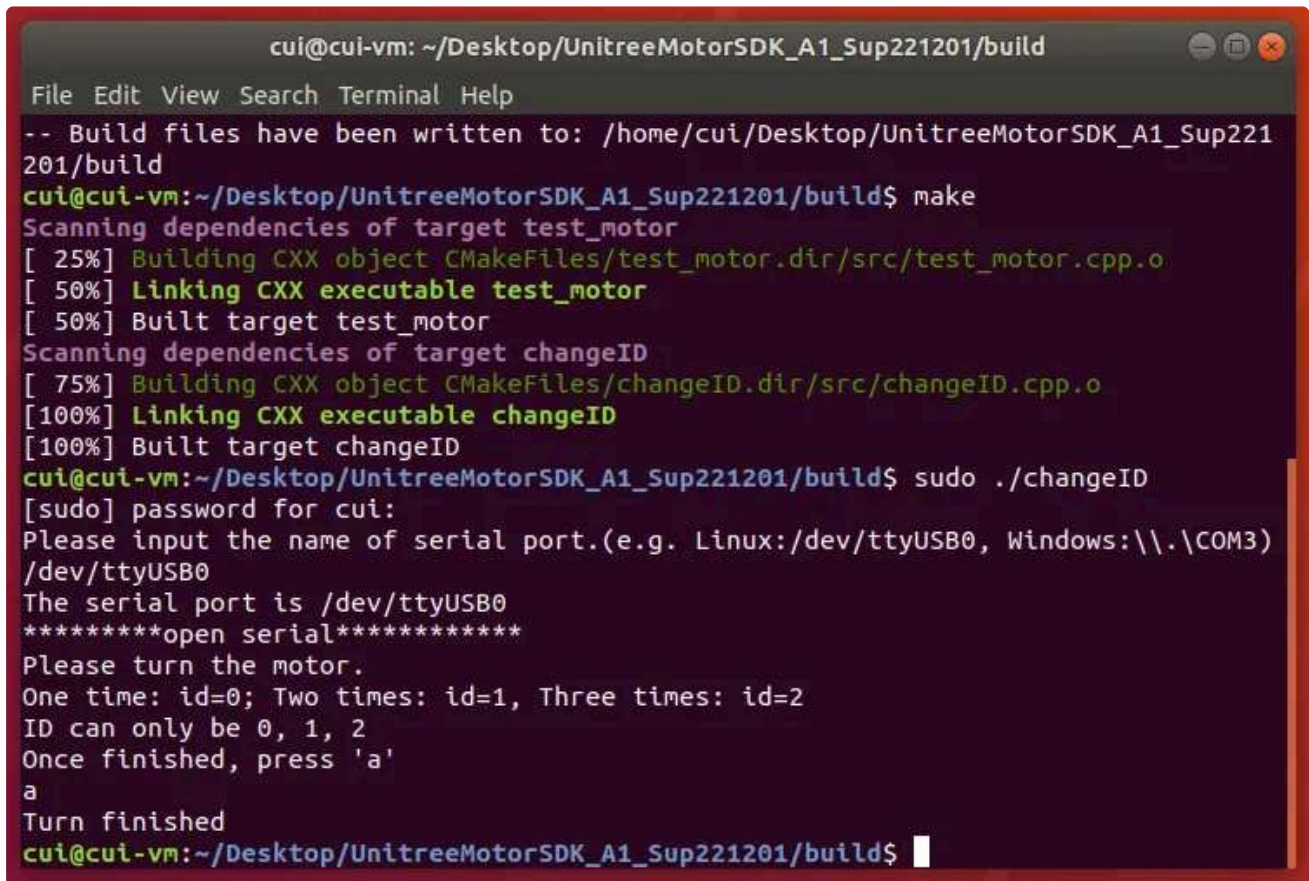
其中sudo的含义为以管理员权限运行，因此在按下回车之后需要输入管理员密码。请注意在 Linux 中，输入密码时并不会显示 “\*\*\*\*” 等标识，只需要在密码输入完毕后按下回车即可。后面的 ./changeID 表示执行当前目录下 (./) 的可执行文件 changeID。  
开始执行changeID程序后，第一步为输入当前的串口名。正如上文所述，在本实例中该串口名为 /dev/ttyUSB0，输入后按下回车，所有电机都会进入修改 ID 模式。  
由于电机输出轴的电子棘轮刚度较大，所以建议操作者按照下图所示安装工装螺丝，并使用杠杆旋转电机输出轴。



需要注意，对于A1电机，工装螺丝为 M4 螺丝，且旋入深度不可超过 5mm。

正如程序所提示，转动输出轴一次，则电机的 ID 被设置为 0，以此类推，且 ID 只能为 0，1 和 2。在转动完每个电机的输出轴后，在终端窗口输入 a 并且按下回车，即完成电机 ID 的修改。

程序执行过程参见下图。



```
cui@cui-vm: ~/Desktop/UnitreeMotorSDK_A1_Sup221201/build
File Edit View Search Terminal Help
-- Build files have been written to: /home/cui/Desktop/UnitreeMotorSDK_A1_Sup221201/build
cui@cui-vm:~/Desktop/UnitreeMotorSDK_A1_Sup221201/build$ make
Scanning dependencies of target test_motor
[ 25%] Building CXX object CMakeFiles/test_motor.dir/src/test_motor.cpp.o
[ 50%] Linking CXX executable test_motor
[ 50%] Built target test_motor
Scanning dependencies of target changeID
[ 75%] Building CXX object CMakeFiles/changeID.dir/src/changeID.cpp.o
[100%] Linking CXX executable changeID
[100%] Built target changeID
cui@cui-vm:~/Desktop/UnitreeMotorSDK_A1_Sup221201/build$ sudo ./changeID
[sudo] password for cui:
Please input the name of serial port.(e.g. Linux:/dev/ttyUSB0, Windows:\\.\\COM3)
/dev/ttyUSB0
The serial port is /dev/ttyUSB0
*****open serial*****
Please turn the motor.
One time: id=0; Two times: id=1, Three times: id=2
ID can only be 0, 1, 2
Once finished, press 'a'
a
Turn finished
cui@cui-vm:~/Desktop/UnitreeMotorSDK_A1_Sup221201/build$
```

## 4.3 测试程序

SDK中提供了电机测试程序test\_motor.cpp，用于测试电机通信、运转是否正常。该程序在SDK中src目录下，需要先按实际情况修改电机 ID 和系统串口名，然后再编译使用。

▼ test\_motor.cpp

Plain Text

📄 复制代码

```
1  #include "serialPort/SerialPort.h"
2  #include <csignal>
3
4  int main(){
5      // set the serial port name
6      SerialPort serial("/dev/ttyUSB0");
7
8      // send message struct
9      MOTOR_send motor_run, motor_stop;
10     // receive message struct
11     MOTOR_recv motor_r;
12
13     // set the id of motor
14     motor_run.id = 0;
15     // set the motor type, A1 or B1
16     motor_run.motorType = MotorType::A1;
17     motor_run.mode = 5;
18     motor_run.T = 0.0;
19     motor_run.W = 0.0;
20     motor_run.Pos = 0.0;
21     motor_run.K_P = 0.0;
22     motor_run.K_W = 0.0;
23
24     motor_stop.id = motor_run.id;
25     motor_stop.motorType = motor_run.motorType;
26     motor_stop.mode = 0;
27
28     motor_r.motorType = motor_run.motorType;
29
30     // encode data into motor commands
31     modify_data(&motor_run);
32     modify_data(&motor_stop);
33
34     // turn for 3 second
35     for(int i(0); i<3; ++i){
36         serial.sendRecv(&motor_run, &motor_r);
37         // decode data from motor states
38         extract_data(&motor_r);
39         std::cout << "Pos:    " << motor_r.Pos << std::endl;
40         std::cout << "Temp:   " << (int)motor_r.Temp << std::endl;
41         std::cout << "MError: " << (int)motor_r.MError << std::endl;
42         usleep(1000000);
43     }
44
45     // stop the motor
46     while(!serial.sendRecv(&motor_stop, &motor_r)){
47         usleep(100000);
48     }
49 }
```

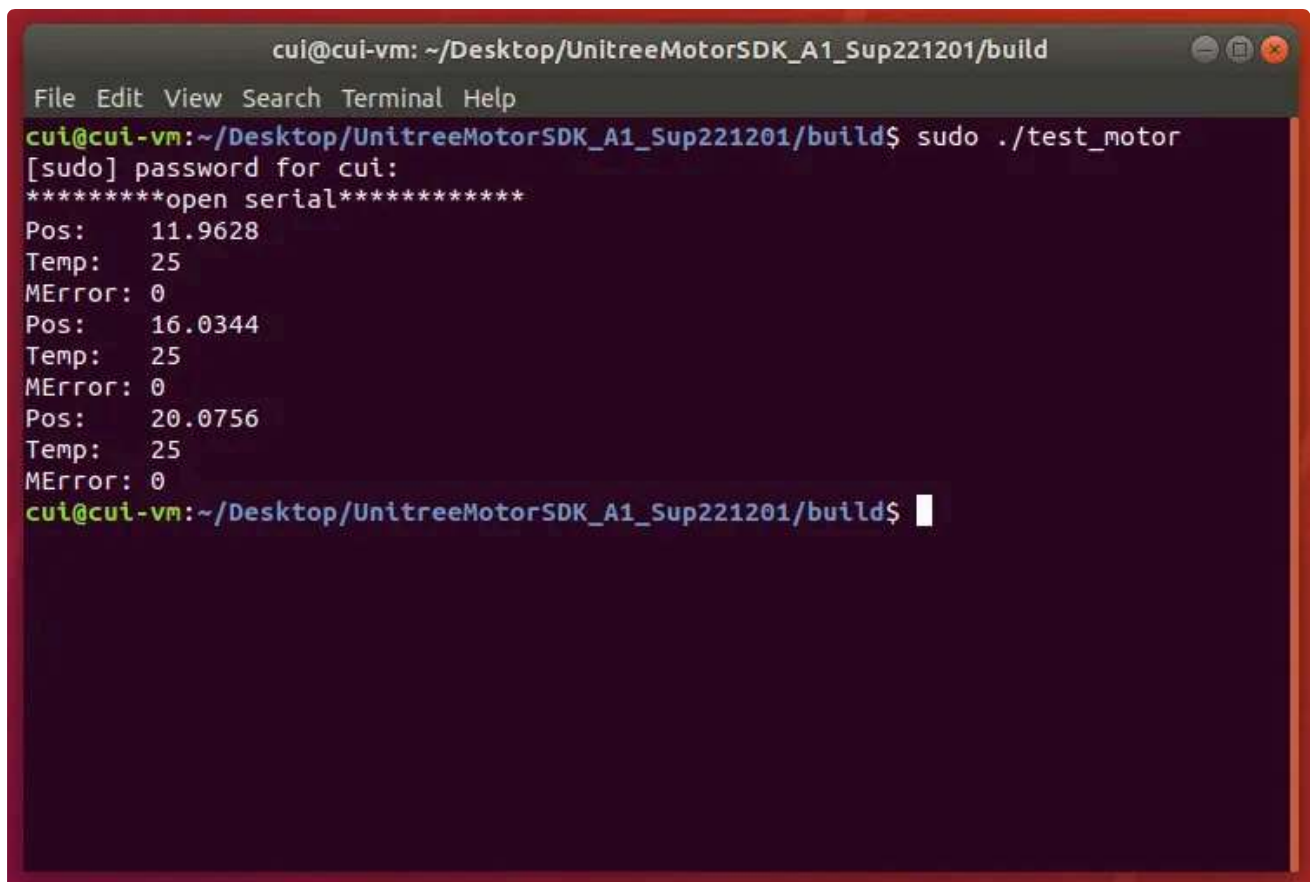
3790 字

```
50     return 0;
51 }
52
```

测试程序中默认的电机电 ID = 0，系统串口名为/dev/ttyUSB0。修改好后保存代码，编译运行命令同修改ID程序：

Plain Text | 复制代码

```
1  mkdir build
2  cd build
3  cmake ..
4  make
5  sudo ./test_motor
```



```
cui@cui-vm: ~/Desktop/UnitreeMotorSDK_A1_Sup221201/build
File Edit View Search Terminal Help
cui@cui-vm:~/Desktop/UnitreeMotorSDK_A1_Sup221201/build$ sudo ./test_motor
[sudo] password for cui:
*****open serial*****
Pos:    11.9628
Temp:   25
MError: 0
Pos:    16.0344
Temp:   25
MError: 0
Pos:    20.0756
Temp:   25
MError: 0
cui@cui-vm:~/Desktop/UnitreeMotorSDK_A1_Sup221201/build$
```

如果修改ID过程中，已经建立了 build 目录，则修改好代码后，直接切换到 build 目录 make 即可。

cd build

make

sudo ./test\_motor

下面对测试程序做简要分析：



Plain Text

复制代码

```
1 // set the serial port name
2 SerialPort serial("/dev/ttyUSB0");
```

上述代码设置串口的名称，详见4.1节。



Plain Text

复制代码

```
1 // send message struct
2 MOTOR_send motor_run, motor_stop;
3 // receive message struct
4 MOTOR_recv motor_r;
5
6 // set the id of motor
7 motor_run.id = 0;
8 // set the motor type, A1 or B1
9 motor_run.motorType = MotorType::A1;
10 motor_run.mode = 5;
11 motor_run.T = 0.0;
12 motor_run.W = 0.0;
13 motor_run.Pos = 0.0;
14 motor_run.K_P = 0.0;
15 motor_run.K_W = 0.0;
16
17 motor_stop.id = motor_run.id;
18 motor_stop.motorType = motor_run.motorType;
19 motor_stop.mode = 0;
20
21 motor_r.motorType = motor_run.motorType;
```

上述代码定义了两个MOTOR\_send结构体变量motor\_run、motor\_stop和一个MOTOR\_recv结构体变量motor\_r。

motor\_run用来发送控制命令，motor\_stop用来发送停止命令，motor\_r用来接收电机状态。

依次设置motor\_run、motor\_stop的id、motorType、运行模式mode、T、W、Pos、K\_P、K\_W。

这里采用mode=5，开环转动。当 mode 的值等于 0 或 5 时，后面的5个控制参数并没有任何作用。

motor\_r仅需设置电机类型motorType。

▼ Plain Text 复制代码

```
1 // turn for 3 second
2 for(int i(0); i<3; ++i){
3     serial.sendRecv(&motor_run, &motor_r);
4     // decode data from motor states
5     extract_data(&motor_r);
6     std::cout << "Pos:    " << motor_r.Pos << std::endl;
7     std::cout << "Temp:   " << (int)motor_r.Temp << std::endl;
8     std::cout << "MError: " << (int)motor_r.MError << std::endl;
9     usleep(1000000);
10 }
```

上述代码用一个循环，每隔1秒发送一次控制命令，并输出电机状态。

实际上如果只是要让电机持续运转，并不需要持续发送同一条命令。之所以这么做，是为了持续获取电机当前的状态，如力矩、位置、速度等。因为电机只有在接收到命令后，才会返回自身的状态，所以需要持续发送命令。电机返回的状态是经过压缩编码的，所以需要通过函数 `extract_data()` 将其解码。

▼ Plain Text 复制代码

```
1 // stop the motor
2 while(!serial.sendRecv(&motor_stop, &motor_r)){
3     usleep(1000000);
4 }
```

上述代码发送停止命令，并用while循环确保成功。

## 4.4 SDK的常见错误

1. 程序串口名错误或者没有连接宇树 USB 转 RS-485 模块



```
cui@cui-vm: ~/Desktop/UnitreeMotorSDK_A1_Sup221201/build
File Edit View Search Terminal Help
cui@cui-vm:~/Desktop/UnitreeMotorSDK_A1_Sup221201/build$ sudo ./test_motor
[sudo] password for cui:
*****open serial*****
terminate called after throwing an instance of 'IOException'
what():  IO Exception (2): No such file or directory, file /home/supereasy/Des
ktop/unitree_actuator_sdk/src/serialPort/src/SerialPort.cpp, line 205.
Aborted
cui@cui-vm:~/Desktop/UnitreeMotorSDK_A1_Sup221201/build$
```

解决办法：查看4.1节内容

## 2. 程序权限不够

```
cui@cui-vm: ~/Desktop/UnitreeMotorSDK_A1_Sup221201/build
File Edit View Search Terminal Help
cui@cui-vm:~/Desktop/UnitreeMotorSDK_A1_Sup221201/build$ ./test_motor
*****open serial*****
terminate called after throwing an instance of 'IOException'
what():  IO Exception (13): Permission denied, file /home/supereasy/Desktop/un
itree_actuator_sdk/src/serialPort/src/SerialPort.cpp, line 205.
Aborted (core dumped)
cui@cui-vm:~/Desktop/UnitreeMotorSDK_A1_Sup221201/build$
```

解决办法：sudo权限运行

## 3. 电机ID错误 or 电机没有供电 or 电机通讯故障



```
cui@cui-vm: ~/Desktop/UnitreeMotorSDK_A1_Sup221201/build
File Edit View Search Terminal Help
[100%] Built target changeID
cui@cui-vm:~/Desktop/UnitreeMotorSDK_A1_Sup221201/build$ sudo ./test_motor
[sudo] password for cui:
*****open serial*****
[WARNING] SerialPort::recv, unblock version, wait time out
[WARNING] motor id=1 does not reply
[WARNING] Receive data CRC error
Pos: 0
Temp: 0
MError: 0
[WARNING] SerialPort::recv, unblock version, wait time out
[WARNING] motor id=1 does not reply
[WARNING] Receive data CRC error
Pos: 0
Temp: 0
MError: 0
[WARNING] SerialPort::recv, unblock version, wait time out
[WARNING] motor id=1 does not reply
[WARNING] Receive data CRC error
Pos: 0
Temp: 0
MError: 0
^C
cui@cui-vm:~/Desktop/UnitreeMotorSDK_A1_Sup221201/build$
```

解决办法：检查电机ID或者重新设置电机ID，检查电机供电，检查485通讯线等

f71c7d6505b5.png&title=%5Bwriting%5DB1%E7%94%B5%E6%9C%BA%E7%AE%80%E8%A6%81%