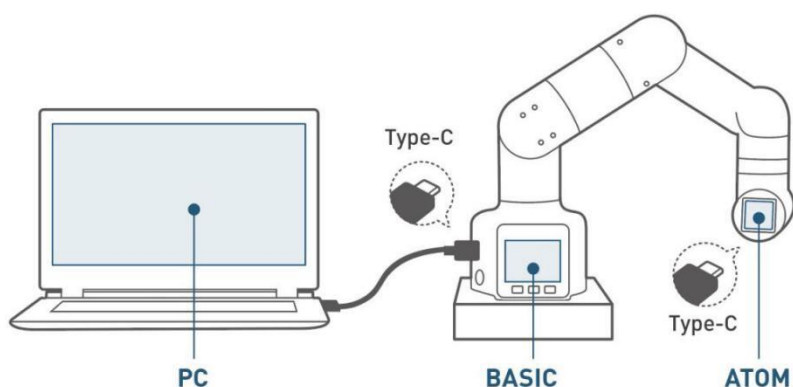


Basic开发引导手册



Arduino 创客! Maker!	Arduino IDE + M5Stack Lib 库 + MycobotBasic Lib 库	各类程序自定义 All Exapmles	atomMain
通信协议 - USB/TxRx0(G1/G3)	通信协议阅读 Read Protocol	Transponder文件	atomMain

语言：简体中文

版本号：V 2021.01.27

版权声明

未经深圳市大象机器人科技有限公司（以下简称“大象机器人”）的书面许可，任何单位和个人不得擅自摘抄、撰写、转译、复制本手册（技术文档、软件等）的任何内容，不得以任何形式（包括资料和出版物）进行传播。

除此以外，本手册提到的产品信息及其相关资源仅供参考，内容如有更新，恕不另行通知。

除本手册中有明确陈述之外，本手册中的任何内容不应解释为大象机器人对个人损失、财产损害和具体适用性等做出的任何担保或保证。

版权所有，侵权必究。

说明：本手册仅适用于中国大陆地区用户。

目录

版权声明	2
1. 关于手册	4
2. Arduino API 指令	4
2.1 系统与产品信息 System Status (开发中)	4
2.2 机器人整体运行状态 Overall Status	4
2.3 输入程序控制模式 MDI Mode and Robot Control (Manual Data Input)	5
2.4 微动控制模式 JOG Mode.....	8
2.5 运行辅助信息 Running Status and Settings	9
2.6 关节电机设置 Joint Servo Control	10
2.7 Atom 末端 IO Atom IO Control	12
2.8 坐标控制模式.....	12
2.9 运动学新增接口说明	15
3. 通信协议与数据结构	15
3.1 USB 通信设置 Communication Settings.....	16
3.2 命令帧说明及单一指令解析	16
3.3 单一指令解析	16
4.联系我们	41

1. 关于手册

本文适用于MyCobot系列机械臂。

库文件下载、例程下载、固件下载等请浏览以下链接：

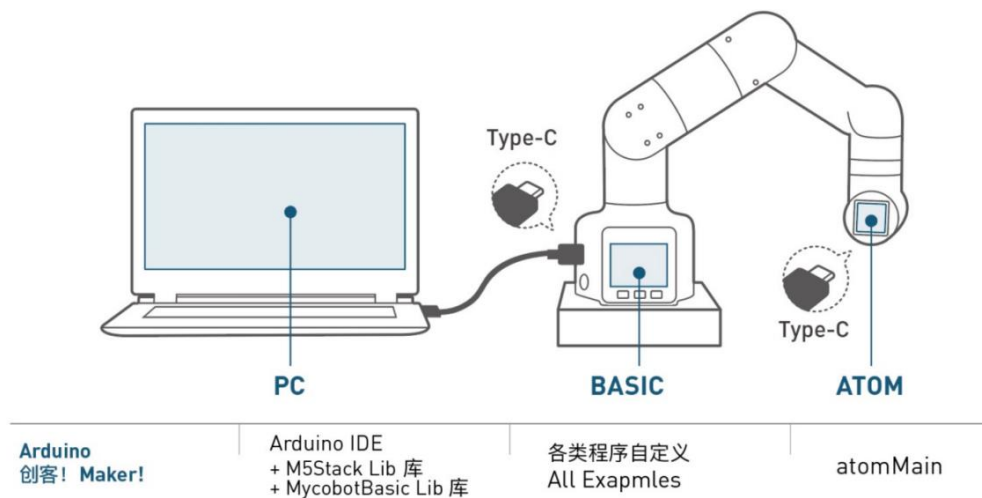
<https://github.com/elephantrobotics/myCobot>

产品细节以及周边产品请浏览以下链接：

<https://www.elephantrobotics.com/myCobot/>

2. Arduino API指令

Arduino API 需要在使用与烧录如下程序



2.1 系统与产品信息 System Status (开发中)

2.2 机器人整体运行状态 Overall Status

`powerOn();`

- 功能：atom打开通讯（默认打开）
- 返回值：无

powerOff();

- 功能：atom关闭通讯
- 返回值：无

isPoweredOn();

- 功能：atom状态查询，返回atom链接状态
- 返回值：打开TRUE、关闭FALSE

setFreeMove();

- 功能：所有关节关闭扭力输出
- 返回值：无

2.3 输入程序控制模式 MDI Mode and Robot Control (Manual Data Input)

getAngles();

- 功能：读取所有关节角度，使用时应定义一个Angles angles，来接收读取到的角度，Angles是库函数内置的变量或函数定义，可以定义一个内存为6的储存空间angles，用来储存角度变量，使用的方式与数组相同。
- 返回值：Angles 类型的数组

writeAngle(int joint, float value, int speed);

- 功能：发送单关节角度
- 参数说明：
关节序号 = joint，取值范围1-6;
指定角度值 = value，取值范围约-170°~ + 170°
指定速度 = speed，取值范围0~100;
- 返回值：无

writeAngles(Angles angles, int speed);

- 功能：关节角度同步执行，同时发送六个关节的角度给执行器Angles为库函数声明的定义类型，指定angles是容量大小为6个数据的容器，可理解为数组，赋值

时可使用for循环赋值，也可单独赋值。

- 参数说明：

Angles[0] = 具体角度，Angles[2] = 具体角度，以此类推取值范围 0 – 90（范围为定义，取值范围应和writeAngle相同）单位°

运动速度 = speed，取值范围 0~100单位 %

- 返回值：无

getCoords();

- 功能：读取当前机械臂末端的x,y,z,rx,ry,rz，使用时应定义一个Coords tempcoords，来接收读取到的角度，Coords是库函数内置的变量或函数定义，可以定义一个内存为6的储存空间tempcoords，用来储存角度变量,使用的方式与数组相同。

- 返回值：Coords 类型下的一个数组，需要定义好Coords类型的变量

isInPosition (Coords coord,bool is_linear);

- 功能：读取当前机械臂末端的x,y,z,rx,ry,rz，测试是否到达指定点位，使用时应定义一个Coords tempcoords，来接收读取到的角度，Coords是库函数内置的变量或函数定义，可以定义一个内存为6的储存空间tempcoords，用来储存角度变量,使用的方式与数组相同。

- 返回值：Coords 类型下的一个数组，需要定义好Coords类型的变量

writeCoord(Axis axis, float value, int speed);

- 功能：发送单独坐标参数x/y/z的具体数值，末端会在单独方向上移动，

- 参数说明：

移动的路径坐标值 = value 取值范围 -300 – 300 (axis=Axis::X,aixs=Axis::Y和axis=Axis::Z为位置坐标分别为X,Y,Z，单位mm，位置坐标取值范围不统一，axis=Axis::RX, aixs=Axis::RY和axis=Axis::RZ分别为RX,RY,RZ取值范围为-180°至180°，超出取值范围会返回inverse kinematics no solution提示)

指定速度 =speed 取值范围0~100 单位 %

- 返回值：无

writeCoords(Coords coords, int speed);

- 功能：发送指定的坐标参数，参数的类型应是Coords，需要声明一个Coords类型的变量，此变量的使用方法与数组相同

- 参数说明：

coords[0] = X, coords[1] = Y, coords[2] = Z,

X,Y,Z取值范围 -300-300.00（取值范围未定义，超出范围会返回inverse kinematics no solution提示）单位mm

RX,RY,RZ取值范围 -180~180

指定速度 =speed 取值范围0~100单位 %

- 返回值：无

checkRunning();

- 功能：检查设备是否在运动
- 返回值：运动中回复TRUE，否则回复FALSE

setEncoder(int joint, int encoder);

- 功能：设定单一关节转动至指定电位值
- 参数说明：关节序号 = joint 取值范围 1-6；舵机电位值 = encoder取值范围 0-4096（该范围应与每个关节的范围正相关）
- 返回值：无

getEncoder(int joint);

- 功能：获取指定关节电位值
- 参数说明：舵机序号 = joint 取值范围 1-6
- 返回值：int 类型，参考取值范围0-4096

setEncoders(Angles angleEncoders, int speed);

- 功能：设定机械臂六个关节同步执行至指定位置
- 参数说明：需要定义一个Angles类型的变量angleEncoders,angleEncoders 的使用方法等同于数组，对数组angleEncoders赋值，取值范围0~4096（该范围

应与每个关节的范围正相关)，数组的长度范围是6

指定速度 = speed，取值范围 0~100 单位 %

- 返回值：无

2.4 微动控制模式JOG Mode

jogAngle(int joint, int direction, int speed);

- 功能：控制设备单一关节向一个方向运动
- 参数说明：

关节舵机序号 = joint 取值范围 1-6

关节运动方向 = Direction 取值范围 -1/1

指定速度 = speed，取值范围0~100 单位%

- 返回值：无

jogCoord(Axis axis, int direction, int speed);

- 功能：控制设备在笛卡尔空间中向一个方向运动
- 参数说明：

设备方向选择 = axis 取值 X,Y,Z,RX,RY,RZ

关节运动方向 = Direction 取值 -1/1

指定速度 = speed，取值范围0~100 单位 %

- 返回值：无

jogStop();

- 功能：停止已经开始的指定方向运动
- 返回值：无

pause();

- 功能：程序暂停运行
- 返回值：无

resume();

- 功能：程序继续运行
- 返回值：无

stop();

- 功能：程序停止运行
- 返回值：无

2.5 运行辅助信息 Running Status and Settings

getSpeed();

- 功能：读取设备的当前运行速度
- 返回值：int类型，数值范围0-100，单位 %

setSpeed(int percentage);

- 功能：设置设备运行速度
- 参数说明：percentage取值范围0 – 100，单位 %

getFeedOverride();

- 功能：读取FeedOverride
- 返回值：float类型的数值

sendFeedOverride(float feedOverride);

- 功能：发送FeedOverride

getAcceleration();

- 功能：读取加速度
- 返回值：float类型的数值

setAcceleration(float acceleration);

- 功能：设置加速度

- 参数说明：acceleration 取值范围0-100（范围未定义）

getJointMin(int joint);

- 功能：读取关节最小限制角度
- 参数说明：关节舵机序号 = joint，取值范围1-6
- 返回值：float类型的数值

getJointMax(int joint);

- 功能：读取关节最大限制角度
- 参数说明：关节舵机序号 = joint，取值范围1-6
- 返回值：float类型的数值

setJointMin(int joint, float angle);

- 功能：设置关节最小限制角度
- 参数说明：关节舵机序号 = joint，取值范围 1-6；关节对应角度 = angle，取值范围-180 ~ 180
- 返回值：无

setJointMax(int joint, float angle);

- 功能：设置关节最大限制角度
- 参数说明：关节舵机序号 = joint，取值范围 1-6；关节对应角度 = angle，取值范围-180 ~ 180
- 返回值：无

2.6 关节电机设置 Joint Servo Control

isServoEnabled(int joint);

- 功能：检测关节舵机是否连接正常
- 参数说明：关节舵机序号 = joint，取值范围1-6
- 返回值：正常链接返回TRUE，否则返回FALSE

isAllServoEnabled();

- 功能：检测所有关节舵机是否连接正常
- 返回值：正常链接返回TRUE，否则返回FALSE

getServoData(int joint, byte data_id);

- 功能：读取舵机指定地址的数据参数
- 参数说明：

关节舵机序号 = joint，取值范围1 - 6；

数据地址 = data_id，取值范围请参考下图1.1中地址
- 返回值：下图1.1中取值范围

地址	功能	取值范围	初始值	取值解析
20	LED 报警条件	0-254	0	对应位设置 1 为开启闪灯报警 对应位设置 0 为关闭闪灯报警
21	位置环 P 比例系数	0-254	123 关节 取值 8, 456 取值 5。	控制电机的比例系数
22	位置环 D 微分系数	0-254	123 关节 取值 20, 456 关节 取值 13.	控制电机的微分系数
23	位置环 I 积分系数	0-254	0	控制电机的积分系数
24	最小启动 力	0-1000	0	设置舵机的最小输出启动扭矩，设 1000 = 100% * 堵转扭力

setServoData(byte servo_no, byte servo_state, byte servo_data);

- 功能：设置舵机指定地址的数据参数
- 参数说明：

关节舵机序号 = servo_no，取值范围1 - 6

数据地址 = servo_state，取值范围请参考上图1.1中地址

数据内容 = servo_data，取值范围参考上图1.1取值范围
- 返回值：无

setServoCalibration(int joint);

- 功能：校准关节舵机当前位置为角度零点，对应电位值为2048
- 参数说明：关节舵机序号 = joint，取值范围1 – 6

jointBrake(int joint);

- 功能：刹车单个电机
- 参数说明：关节舵机序号 = joint，取值范围1 – 6

setPinMode(byte pin_no, byte pin_mode);

- 功能：设置atom指定引脚的状态模式
- 参数说明：
引脚序号 = pin_no，取值范围：19、22、23、26、32、33
输出模式 = pin_mode 取值范围：0、1
- 返回值：无

2.7 Atom末端IO Atom IO Control

setLEDRGB(byte r, byte g, byte b);

- 功能：设定atom屏幕的RGB灯的颜色：
- 参数说明：
红色光对应参数值 = r，取值范围 0x00 – 0xFF；
绿色光对应参数值 = g，取值范围 0x00 – 0xFF；
蓝色光对应参数值 = b，取值范围 0x00 – 0xFF；
- 返回值：无

setGripper(int data);

- 功能：设置夹爪开合
- 参数说明：data 为0打开，为1关闭

2.8 坐标控制模式

setToolReference(Coords coords);

- 功能：设置工具坐标系
- 参数说明：
X,Y,Z取值范围 -300-300.00（取值范围未定义，超出范围会返回inverse kinematics no solution提示）单位mm
RX,RY,RZ取值范围 -180~180
- 返回值：无

setWorldReference(Coords coords);

- 功能：设置世界坐标系
- 参数说明：
X,Y,Z取值范围 -300-300.00（取值范围未定义，超出范围会返回inverse kinematics no solution提示）单位mm
RX,RY,RZ取值范围 -180~180
- 返回值：无

getToolReference();

- 功能：获取工具坐标系
- 参数说明：无
- 返回值：
X,Y,Z取值范围 -300-300.00（取值范围未定义，超出范围会返回inverse kinematics no solution提示）单位mm
RX,RY,RZ取值范围 -180~180

getWorldReference();

- 功能：获取世界坐标系
- 参数说明：无
- 返回值：
X,Y,Z取值范围 -300-300.00（取值范围未定义，超出范围会返回inverse kinematics no solution提示）单位mm

RX,RY,RZ取值范围 -180~180

setReferenceFrame(RFType rftype);

- 功能：设置法兰坐标系
- 参数说明：
RFType::BASE为将机器人基座作为基坐标，RFType::WORLD为将世界坐标系作为基坐标。
- 返回值：无

getReferenceFrame();

- 功能：获取法兰坐标系
- 参数说明：无
- 返回值：
X,Y,Z取值范围 -300-300.00（取值范围未定义，超出范围会返回inverse kinematics no solution提示）单位mm
RX,RY,RZ取值范围 -180~180

setEndType(EndType end_type)

- 功能：设置末端坐标系
- 参数说明：EndType::FLANGE为将末端设置为法兰，EndType::TOOL为将末端设置为工具末端。
- 返回值：无

getEndType();

- 功能：获取末端坐标系
- 参数说明：无
- 返回值：
X,Y,Z取值范围 -300-300.00（取值范围未定义，超出范围会返回inverse kinematics no solution提示）单位mm
RX,RY,RZ取值范围 -180~180

2.9 运动学新增接口说明

在运动学库中添加了相应的坐标变换程序，具体实现方式如下所述：

1) 改变末端坐标系

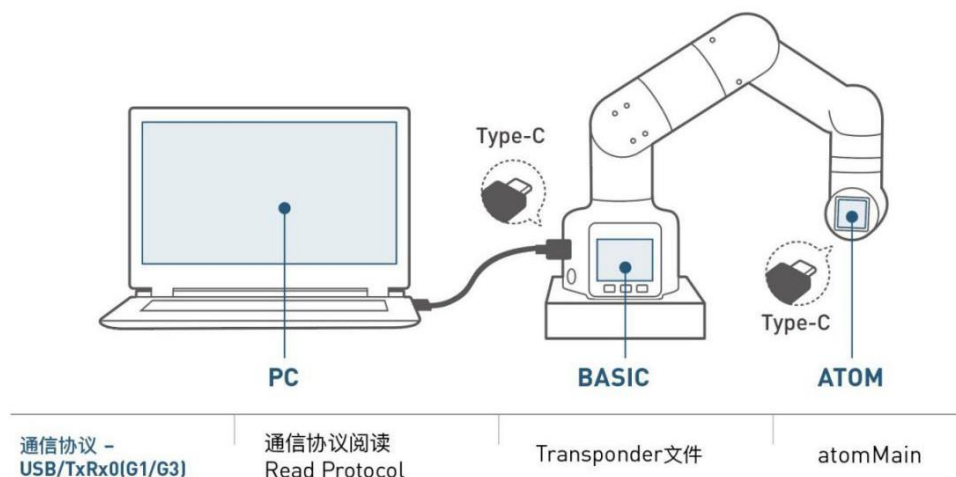
- 1、通过 `setEndType` 和 `getEndType` 函数可以设置末端坐标系，`EndType::FLANGE` 为将末端设置为法兰，`EndType::TOOL` 为将末端设置为工具末端。
- 2、通过 `setToolReference` 和 `getToolReference` 函数可以设置读取工具的坐标信息。设置是以法兰坐标系为相对坐标系，工具末端信息是相对于法兰坐标系的。
- 3、将 `EndType` 设置为 `FLANGE` 后，`GetCoords` 和 `WriteCoords` 方法均以法兰位置计算。
- 4、将 `EndType` 设置为 `TOOL` 后，`GetCoords` 和 `WriteCoords` 方法均以工具末端位置计算。

2) 改变基坐标系

- 1、通过 `setReferenceFrame` 函数可以设置基坐标系，`RType::BASE` 为将机器人基座作为基坐标，`RType::WORLD` 为将世界坐标系作为基坐标。`getReferenceFrame` 函数为读取当前基坐标系种类。
- 2、通过 `setWorldReference` 和 `getWorldReference` 函数可以设置读取基坐标系信息。设置时是以世界坐标系为相对坐标系，输入机器人的基座相对于世界坐标系的位置信息。
- 3、当基坐标系为基座时，`GetCoords` 和 `WriteCoords` 均以基座为参考坐标系。
- 4、当基坐标系为世界坐标系时，`GetCoords` 和 `WriteCoords` 方法均以世界坐标系作为参考坐标系。

3. 通信协议与数据结构

注意：使用通信协议直接通信，需要在basic中烧录transponder，在atom中烧录最新版的atomMain



3.1 USB通信设置Communication Settings

- 总线接口：USB Type-C连接
- 波特率：115200
- 数据位：8
- 奇偶校验：无
- 停止位：1

3.2 命令帧说明及单一指令解析

主机Basic向从机发送数据，从机接收到数据后进行解析，如包含返回值的指令，从机会在500ms内返回给主机。

类型	数据描述	数据长度	说明
命令帧	头字节0	1	帧头识别，0XFE
	头字节1	1	帧头识别，0XFE
	数据长度字节	1	不同指令对应不同长度数据
	命令字节	1	视不同命令而定
数据帧	数据	0-16	命令附带数据，视不同命令而定
结束帧	结束字节	1	停止位，0XFA

3.3 单一指令解析

1). Atom打开通讯

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X02
Data[3]	指令帧	0X10
Data[4]	结束帧	0XFA

串口发送示例：FE FE 02 10 FA

无返回值

2). Atom关闭通讯

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X02
Data[3]	指令帧	0X11
Data[4]	结束帧	0XFA

串口发送示例：FE FE 02 11 FA

无返回值

3). Atom状态查询

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X02
Data[3]	指令帧	0X12
Data[4]	结束帧	0XFA

串口发送示例：FE FE 02 12 FA

返回值

数据域	说明	数据
Data[0]	返回帧头	0XFE
Data[1]	返回帧头	0XFE
Data[2]	返回长度帧	0X02
Data[3]	返回指令帧	0X12
Data[4]	上电/断电	0X01/0X00
Data[5]	结束帧	0XFA

串口返回示例：FE FE 02 12 00 FA

4). 读取所有关节角度

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X02
Data[3]	指令帧	0X20
Data[4]	结束帧	0XFA

串口发送示例：FE FE 02 20 FA

从机返回数据结构

数据域	说明	数据
Data[0]	返回帧头	0XFE
Data[1]	返回帧头	0XFE
Data[2]	返回长度帧	0X0E
Data[3]	返回指令帧	0X20
Data[4]	1号舵机角度高位	Angle1_high

Data[5]	1号舵机角度低位	Angle1_low
Data[6]	2号舵机角度高位	Angle2_high
Data[7]	2号舵机角度低位	Angle2_low
Data[8]	3号舵机角度高位	Angle3_high
Data[9]	3号舵机角度低位	Angle3_low
Data[10]	4号舵机角度高位	Angle4_high
Data[11]	4号舵机角度低位	Angle4_low
Data[12]	5号舵机角度高位	Angle5_high
Data[13]	5号舵机角度低位	Angle5_low
Data[14]	6号舵机角度高位	Angle6_high
Data[15]	6号舵机角度低位	Angle6_low
Data[16]	结束帧	0XFA

串口返回示例：FE FE 0E 20 06 E6 EA 4E C4 81 0B BD EA C0 02 B6 FA

如何得出1号关节角度

temp = angle1_low+angle1_high*256

Angle1= (temp > 33000 ?(temp – 65536) : temp) /100

计算方式：角度值低位 + 角度高位值乘以256 先判断是否大于33000 如果大于33000就再减去65536 最后除以100 如果小于33000就直接除以100

(其余同理)

5). 发送单独角度

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X06
Data[3]	指令帧	0X21
Data[4]	舵机序号	joint_no
Data[5]	角度值高位	angle_high
Data[6]	角度值低位	angle_low
Data[7]	指定速度	sp
Data[8]	结束帧	0XFA

串口发送示例：FE FE 06 21 00 00 00 20 FA

joint_no取值范围 0~5

angle_high：数据类型byte

计算方式：角度值乘以100 先转换成int形式 再取十六进制的高字节

angle_low：数据类型byte

计算方式：角度值乘以100 先转换成int形式 再取十六进制的低字节

无返回值

6). 发送全部角度

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X0F
Data[3]	指令帧	0X22
Data[4]	1号舵机角度值高字节	Angle1_high
Data[5]	1号舵机角度值低字节	Angle1_low
Data[6]	2号舵机角度值高字节	Angle2_high
Data[7]	2号舵机角度值低字节	Angle2_low
Data[8]	3号舵机角度值高字节	Angle3_high
Data[9]	3号舵机角度值低字节	Angle3_low
Data[10]	4号舵机角度值高字节	Angle4_high
Data[11]	4号舵机角度值低字节	Angle4_low
Data[12]	5号舵机角度值高字节	Angle5_high
Data[13]	5号舵机角度值低字节	Angle5_low
Data[14]	6号舵机角度值高字节	Angle6_high
Data[15]	6号舵机角度值低字节	Angle6_low
Data[16]	指定速度	Sp
Data[17]	结束帧	0XFA

串口发送示例：FE FE 0F 22 06 E6 EA 4E C4 81 0B BD EA C0 02 B6 FA

angle1_high：数据类型byte

计算方式：1号舵机角度值乘以100 先转换成int形式 再取十六进制的高字节

angle1_low：数据类型byte

计算方式：1号舵机角度值乘以100 先转换成int形式 再取十六进制的低字节

（其余同理）

无返回值

7). 读取全部坐标参数

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X02
Data[3]	指令帧	0X23
Data[4]	结束帧	0XFA

串口发送示例：FE FE 02 23 FA

从机返回数据结构

数据域	说明	数据
Data[0]	返回帧头	0XFE
Data[1]	返回帧头	0XFE
Data[2]	返回长度帧	0X0E
Data[3]	返回指令帧	0X23
Data[4]	指定x坐标高位	x_high
Data[5]	指定x坐标低位	x_low
Data[6]	指定y坐标高位	y_high
Data[7]	指定y坐标低位	y_low
Data[8]	指定z坐标高位	z_high
Data[9]	指定z坐标低位	z_low
Data[10]	指定rx坐标高位	rx_high
Data[11]	指定rx坐标低位	rx_low
Data[12]	指定ry坐标高位	ry_high
Data[13]	指定ry坐标低位	ry_low
Data[14]	指定rz坐标高位	rz_high
Data[15]	指定rz坐标低位	rz_low
Data[16]	结束帧	0XFA

串口返回示例：FE FE 0E 23 00 00 00 00 00 00 00 00 00 00 00 00 FA

如何得出x坐标

$\text{temp} = \text{x_low} + \text{x_high} * 256$

$\text{x坐标} = (\text{temp} > 33000 ? (\text{temp} - 65536) : \text{temp}) / 10$

计算方式：x坐标值低位 + x坐标值高位乘以256 先判断是否大于33000 如果大于33000就再减去65536 最后除以100 如果小于33000就直接除以10

(y坐标z坐标同理)

如何得出rx坐标

$\text{temp} = \text{rx_low} + \text{rx_high} * 256$

$\text{x坐标} = (\text{temp} > 33000 ? (\text{temp} - 65536) : \text{temp}) / 100$

计算方式：x坐标值低位 + x坐标值高位乘以256 先判断是否大于33000 如果大于33000就再减去65536 最后除以100 如果小于33000就直接除以100

(ry坐标rz坐标同理)

8).发送单独坐标参数

数据域	说明	数据
Data[0]	识别帧	0XFE

Data[1]	识别帧	0xFE
Data[2]	数据长度帧	0x06
Data[3]	指令帧	0x24
Data[4]	axis	x/y/z/rx/ry/rz
Data[5]	指定xyz/rxryrz参数高位	xyz/ rxryrz_high
Data[6]	指定xyz/rxryrz参数低位	xyz/rxryrz_low
Data[7]	指定速度	Sp
Data[8]	结束帧	0xFA

设定X坐标为100，目标速度20

串口发送示例：FE FE 06 24 00 00 64 20 FA

指定坐标axis：数据类型byte

取值范围：0~5

xyz_high：数据类型byte

计算方式：x/y/z坐标值乘以10 再取十六进制的高字节

xyz_low：数据类型byte

计算方式：x/y/z坐标值乘以10 再取十六进制的低字节

rxryrz_high：数据类型byte

计算方式：rx/ry/rz乘以100 再取十六进制的高字节

rxryrz_low：数据类型byte

计算方式：rx/ry/rz乘以100 再取十六进制的低字节

无返回值

9).发送全部坐标参数

数据域	说明	数据
Data[0]	识别帧	0xFE
Data[1]	识别帧	0xFE
Data[2]	数据长度帧	0x10
Data[3]	指令帧	0x25
Data[4]	指定x坐标高位	x_high
Data[5]	指定x坐标低位	x_low
Data[6]	指定y坐标高位	y_high
Data[7]	指定y坐标低位	y_low
Data[8]	指定z坐标高位	z_high
Data[9]	指定z坐标低位	z_low
Data[10]	指定rx坐标高位	rx_high
Data[11]	指定rx坐标低位	rx_low

Data[12]	指定ry坐标高位	ry_high
Data[13]	指定ry坐标低位	ry_low
Data[14]	指定rz坐标高位	rz_high
Data[15]	指定rz坐标低位	rz_low
Data[16]	指定速度	Sp
Data[17]	模式	0X01
Data[18]	结束帧	0XFA

设定机械臂末端目标点位 (-14 , -27 , 275 , -89.5 , 0.7 , -90.7) , 目标速度50

串口发送示例 : FE FE 10 25 FF 74 FE EE 0A C1 DD 05 00 48 DC 95 32 01 FA

x_high : 数据类型byte

计算方式 : x坐标乘以10 再取十六进制的高字节

x_low : 数据类型byte

计算方式 : x坐标乘以10 再取十六进制的低字节

(y轴坐标z轴坐标同理)

rx_high : 数据类型byte

计算方式 : rx坐标值乘以100 再取十六进制的高字节

rx_low : 数据类型byte

计算方式 : rx坐标值乘以100 再取十六进制的低字节

(ry轴坐标rz轴坐标同理)

无返回值

10).指定点位到达检测(开发中)

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X15
Data[3]	指令帧	0X2A
Data[4]	坐标x高位	x_high
Data[5]	坐标x低位	x_low
Data[6]	坐标y高位	y_high
Data[7]	坐标y低位	y_low
Data[8]	坐标z高位	z_high
Data[9]	坐标z低位	z_low
Data[10]	坐标rx高位	rx_high
Data[11]	坐标rx低位	rx_low
Data[12]	坐标ry高位	ry_high
Data[13]	坐标ry低位	ry_low

Data[14]	坐标rz高位	rz_high
Data[15]	坐标rz低位	rz_low
Data[16]	Is_linear	type
Data[17]	结束帧	0XFA

判断机械臂是否到达指定点位

串口发送示例：FE FE 15 2A FF 74 FE EE 0A C1 DD 05 00 48 DC 95 FA

x_high：数据类型byte

计算方式：x坐标乘以10 先转换为int类型 再取十六进制高字节

x_low：数据类型byte

计算方式：x坐标乘以10 先转换为int类型 再取十六进制低字节

（y轴坐标z轴坐标同理）

rx_high：数据类型byte

计算方式：rx坐标乘以100 先转换为int类型 再取十六进制高字节

rx_low：数据类型byte

计算方式：rx坐标乘以100 先转换为int类型 再取十六进制低字节

（ry轴坐标rz轴坐标同理）

Type：数据类型byte（暂未使用）

返回数据结构

数据域	说明	数据
Data[0]	返回识别帧	0XFE
Data[1]	返回识别帧	0XFE
Data[2]	返回数据长度帧	0X03
Data[3]	返回指令帧	0X2A
Data[4]	InPosition/noInPosition	0X01/0X00
Data[5]	结束帧	0XFA

已经到达点位；

串口返回示例：FE FE 03 2A 00 FA

11).机械臂运动检测

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X02
Data[3]	指令帧	0X2B
Data[4]	结束帧	0XFA

检查机械臂是否在运动

串口发送示例：FE FE 02 2B FA

返回数据结构

数据域	说明	数据
Data[0]	返回识别帧	0XFE
Data[1]	返回识别帧	0XFE
Data[2]	返回数据长度帧	0X02
Data[3]	返回指令帧	0X2B
Data[4]	Not running/no data - running	0X00/0X01
Data[5]	结束帧	0XFA

串口返回示例：FE FE 02 2B 00 FA

12).jog-关节方向运动

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X05
Data[3]	指令帧	0X30
Data[4]	关节舵机序号	Joint
Data[5]	关节舵机方向	di
Data[6]	指定速度	sp
Data[7]	结束帧	0XFA

设定一号舵机顺时针方向以50%速度转动

串口发送示例：FE FE 05 30 01 01 32 FA

关节序号取值范围 1~6

di：数据类型byte 取值范围 0和1

sp：数据类型byte 取值范围0-100%

无返回值

13).jog-坐标方向运动

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X05
Data[3]	指令帧	0X32
Data[4]	指定坐标	axis
Data[5]	关节舵机方向	Direction
Data[6]	指定速度	sp
Data[7]	结束帧	0XFA

设定末端向x轴逆时针方向以50%的速度运动

串口返回示例：FE FE 05 32 01 00 32 FA

Axis_number：数据类型byte（x = 0 ,y,z,rx,ry,rz）取值范围1~6

di：数据类型byte 取值范围 0~1

sp：数据类型byte 取值范围 0-100%

无返回值

14).jog-停止

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X02
Data[3]	指令帧	0X34
Data[4]	结束帧	0XFA

jog停止运动

串口返回示例：FE FE 02 34 FA

无返回值

15).发送单个舵机电位值

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X05
Data[3]	指令帧	0X3A
Data[4]	关节舵机序号	Joint
Data[5]	电位值高位	Encoder_high
Data[6]	电位值低位	Encoder_low
Data[7]	结束帧	0XFA

示例，设定2号关节到2249电位。

串口发送示例：FE FE 05 3A 01 08 C9 FA

关节序号取值范围 0~5

Joint：数据类型byte

Encoder_high：数据类型byte

计算方式：取电位值（十六进制）的高位

Encoder_low：数据类型byte

计算方式：取电位值（十六进制）的低位

无返回值

16).获取单个舵机电位值

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X03
Data[3]	指令帧	0X3B
Data[4]	关节序号	joint
Data[5]	结束帧	0XFA

获取1号舵机电位值

串口发送示例：FE FE 03 3B 00 FA

关节序号取值范围 1~6

返回数据结构

数据域	说明	数据
Data[0]	返回识别帧	0XFE
Data[1]	返回识别帧	0XFE
Data[2]	返回数据长度帧	0X04
Data[3]	返回指令帧	0X3B
Data[4]	舵机电位值高位	Encoder_high
Data[5]	舵机电位值低位	Encoders_low
Data[6]	结束帧	0XFA

串口返回示例：FE FE 04 3B 08 C9 FA

电位值 = 2249

如何计算电位值

电位值 = 电位值低位 + 电位值高位 * 256

17).发送六个舵机的电位值

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X15
Data[3]	指令帧	0X3C
Data[4]	1号舵机电位值高字节	encoder_1_high
Data[5]	1号舵机电位值低字节	encoder_1_low
Data[6]	2号舵机电位值高字节	encoder_2_high
Data[7]	2号舵机电位值低字节	encoder_2_low
Data[8]	3号舵机电位值高字节	encoder_3_high

Data[9]	3号舵机电位值低字节	encoder_3_low
Data[10]	4号舵机电位值高字节	encoder_4_high
Data[11]	4号舵机电位值低字节	encoder_4_low
Data[12]	5号舵机电位值高字节	encoder_5_high
Data[13]	5号舵机电位值低字节	encoder_5_low
Data[14]	6号舵机电位值高字节	encoder_6_high
Data[15]	6号舵机电位值低字节	encoder_6_low
Data[16]	指定速度	Sp
Data[17]	结束帧	0XFA

发送所有电机的电位值

串口发送示例：FE FE 15 3C 00 00 00 00 00 00 00 00 00 00 00 20 FA

（参考上方发送单独电位值）

encoder_1_high：数据类型byte

计算方式：1号舵机电位值先转换为int类型 再取十六进制高字节

encoder_1_low：数据类型byte

计算方式：1号舵机电位值先转换为int类型 再取十六进制低字节

（其余同理）

Sp：数据类型byte 取值范围0~100%

无返回值

18).读取设定速度

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X02
Data[3]	指令帧	0X40
Data[4]	结束帧	0XFA

串口发送示例：FE FE 02 40 FA

返回数据结构

数据域	说明	数据
Data[0]	返回识别帧	0XFE
Data[1]	返回识别帧	0XFE
Data[2]	返回数据长度帧	0X03
Data[3]	返回指令帧	0X40
Data[4]	指定速度	Sp
数据域	结束帧	0XFA

串口返回示例：FE FE 03 40 32 FA

19).设置运行速度

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X02
Data[3]	指令帧	0X41
Data[4]	指定速度	sp
Data[5]	结束帧	0XFA

串口发送示例：FE FE 02 41 32 FA

无返回值

20).读取FeedOverride (暂未开放)

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X02
Data[3]	指令帧	0X42
Data[4]	结束帧	0XFA

无返回值

21).读取加速度 (暂未开放)

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X02
Data[3]	指令帧	0X44
Data[4]	结束帧	0XFA

22).读取关节最小角度

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X03
Data[3]	指令帧	0X4A
Data[4]	关节舵机序号	Joint_number
Data[5]	结束帧	0XFA

读取

串口发送示例：FE FE 03 4A 00 FA

joint_no取值范围 0~5

返回数据结构

数据域	说明	数据
Data[0]	返回识别帧	0XFE
Data[1]	返回识别帧	0XFE
Data[2]	返回数据长度帧	0X04
Data[3]	返回指令帧	0X4A
Data[4]	舵机角度值高位	Angle_high
Data[5]	舵机角度值低位	Angle_low
Data[6]	结束帧	0XFA

串口返回示例：FE FE 04 4A 01 44 FA

角度 = 90

如何得出关节最小角度

$temp = angle1_low + angle1_high * 256$

$Angle1 = (temp > 33000 ? (temp - 65536) : temp) / 10$

计算方式：角度值低位 + 角度高位值乘以256 先判断是否大于33000 如果大于33000就再减去65536 最后除以10 如果小于33000就直接除以10

23).读取关节最大角度

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X03
Data[3]	指令帧	0X4B
Data[4]	关节舵机序号	joint_number
Data[5]	结束帧	0XFA

串口发送示例：FE FE 03 4B 01 FA

返回数据结构

数据域	说明	数据
Data[0]	返回识别帧	0XFE
Data[1]	返回识别帧	0XFE
Data[2]	返回数据长度帧	0X04
Data[3]	返回指令帧	0X4B
Data[4]	舵机角度值高位	Angle_high
Data[5]	舵机角度值低位	Angle_low
Data[6]	结束帧	0XFA

串口返回示例：FE FE 04 4B 01 44 FA

joint_no取值范围 0~5

如何得出关节最大角度

$temp = angle1_low + angle1_high * 256$

$Angle1 = (temp > 33000 ? (temp - 65536) : temp) / 10$

计算方式：角度值低位 + 角度高位值乘以256 先判断是否大于33000 如果大于33000就再减去65536 最后除以10 如果小于33000就直接除以10

24).查看连接

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X03
Data[3]	指令帧	0X50
Data[4]	结束帧	0XFA

串口发送示例：FE FE 03 50 FA

返回数据结构

数据域	说明	数据
Data[0]	返回识别帧	0XFE
Data[1]	返回识别帧	0XFE
Data[2]	返回数据长度帧	0X03
Data[3]	指令帧	0X50
Data[4]	连接/未连接	0X01/0X00
Data[5]	结束帧	0XFA

串口返回示例：FE FE 03 50 00 FA

25).检测舵机是否全部上电

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X02
Data[3]	指令帧	0X51
Data[4]	结束帧	0XFA

串口发送示例：FE FE 02 51 FA

返回数据结构

数据域	说明	数据
Data[0]	返回识别帧	0XFE
Data[1]	返回识别帧	0XFE
Data[2]	返回数据长度帧	0X03

Data[3]	指令帧	0X51
Data[4]	上电/未上电	0X01/0X00
Data[5]	结束帧	0XFA

串口返回示例：FE FE 03 51 00 FA

26).读取单个舵机配置参数

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X04
Data[3]	指令帧	0X53
Data[4]	关节舵机序号	joint_number
Data[5]	数据	data_id
Data[6]	结束帧	0XFA

读取1号舵机位置P比例参数

串口发送示例：FE FE 04 53 00 21 FA

joint_no取值范围 0~5

Data_id：数据类型byte

取值如下表

地址	功能	取值范围	初始值	取值解析
20	LED 报警条件	0-254	0	对应位设置 1 为开启闪灯报警 对应位设置 0 为关闭闪灯报警
21	位置环 P 比例系数	0-254	123 关节 取值 8, 456 取值 5。	控制电机的比例系数
22	位置环 D 微分系数	0-254	123 关节 取值 20, 456 关节 取值 13.	控制电机的微分系数
23	位置环 I 积分系数	0-254	0	控制电机的积分系数
24	最小启动 力	0-1000	0	设置舵机的最小输出启动扭矩，设 1000 = 100% * 堵转扭力

返回数据结构

数据域	说明	数据
Data[0]	返回识别帧	0XFE
Data[1]	返回识别帧	0XFE
Data[2]	返回数据长度帧	0X03
Data[3]	返回指令帧	0X53
Data[4]	返回数据	data_state
Data[5]	结束帧	0XFA

串口返回示例：FE FE 03 53 00 FA

27).设置单个关节舵机零点

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X03
Data[3]	指令帧	0X55
Data[4]	关节舵机序号	joint_number
Data[5]	结束帧	0XFA

串口发送示例：FE FE 03 55 00 FA

无返回值

28).刹车单个电机(暂未开放)

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X03
Data[3]	指令帧	0X56
Data[4]	关节舵机序号	joint_number
Data[5]	结束帧	0XFA

串口发送示例：FE FE 03 56 00 FA

无返回值

29).设置atom引脚模式

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X04
Data[3]	指令帧	0X60
Data[4]	引脚序号	pin_no
Data[5]	引脚模式	pin_mode
Data[6]	结束帧	0XFA

设置atom pin16为输出模式0

串口发送示例：FE FE 04 60 19 00 FA

Pin_no：数据类型byte

Pin_mode：数据类型byte

无返回值

30).程序暂停运行

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X02
Data[3]	指令帧	0X26
Data[4]	结束帧	0XFA

串口发送示例：FE FE 02 26 FA

无返回值

31).程序继续运行

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X02
Data[3]	指令帧	0X27
Data[6]	结束帧	0XFA

串口发送示例：FE FE 02 27 FA

无返回值

32).程序停止运行

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X02
Data[3]	指令帧	0X28
Data[4]	结束帧	0XFA

串口发送示例：FE FE 02 28 FA

无返回值

33).设置单个关节舵机参数

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X05
Data[3]	指令帧	0X52
Data[4]	关节舵机序号	servo_no

Data[5]	舵机参数地址	servo_state
Data[6]	舵机参数	servo_data
Data[7]	结束帧	0XFA

设置1号舵机位置P比例参数为1

串口发送示例：FE FE 05 52 00 21 01 FA

无返回值

34).机器人关闭扭力输出

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X02
Data[3]	指令帧	0X13
Data[4]	结束帧	0XFA

串口发送示例：FE FE 02 13 FA

无返回值

35).设定atom屏幕RGB灯的颜色

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X05
Data[3]	指令帧	0X6A
Data[4]	R	R
Data[5]	G	G
Data[6]	B	B
Data[7]	结束帧	0XFA

设置RGB为蓝色

串口发送示例：FE FE 05 6A 00 00 FF FA

无返回值

36).设置夹爪状态

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X03
Data[3]	指令帧	0X66

Data[4]	夹爪数据	Gripper_data
Data[5]	结束帧	0XFA

串口发送示例：FE FE 03 66 00 FA

Gripper_data：数据类型byte 取值范围0-1

无返回值

37).设置FeedOverride (暂未开放)

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X04
Data[3]	指令帧	0X43
Data[4]	Feed_override高位	Feed_override_high
Data[5]	Feed_override低位	Feed_override_low
Data[6]	结束帧	0XFA

38).设置加速度 (暂未开放)

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X04
Data[3]	指令帧	0X45
Data[4]	加速度值高位	acceleration_high
Data[5]	加速度值低位	acceleration_low
Data[6]	结束帧	0XFA

acceleration_high：数据类型byte

计算方式：加速度值乘以10 先转换为int格式 再取十六进制的高字节

acceleration_low：数据类型byte

计算方式：加速度值乘以10 先转换为int格式 再取十六进制的低字节

39).设置关节最小角度

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X05
Data[3]	指令帧	0X4C
Data[4]	关节舵机序号	Joint

Data[5]	角度值高位	Angle_high
Data[6]	角度值低位	Angle_low
Data[7]	结束帧	0XFA

设置最小角度为90

串口发送示例：FE FE 05 4C 00 01 44 FA

Joint 取值范围0~5

angle_high：数据类型byte

计算方式：角度值乘以10 先转换成int形式 再取十六进制的高字节

angle_low：数据类型byte

计算方式：角度值乘以10 先转换成int形式 再取十六进制的低字节

无返回值

40).设置关节最大角度

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X05
Data[3]	指令帧	0X4D
Data[4]	关节舵机序号	Joint
Data[5]	角度值高位	Angle_high
Data[6]	角度值低位	Angle_low
Data[7]	结束帧	0XFA

设置最大角度为90

串口发送示例：FE FE 05 4D 00 01 44 FA

Joint 取值范围0~5

angle_high：数据类型byte

计算方式：角度值乘以10 先转换成int形式 再取十六进制的高字节

angle_low：数据类型byte

计算方式：角度值乘以10 先转换成int形式 再取十六进制的低字节

无返回值

41).设置工具坐标系

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X14
Data[3]	指令帧	0X81

Data[4]	X坐标高字节	x_high
Data[5]	X坐标低字节	x_low
Data[6]	Y坐标高字节	y_high
Data[7]	Y坐标低字节	y_low
Data[8]	Z坐标高字节	z_high
Data[9]	Z坐标低字节	z_low
Data[10]	RX坐标高字节	rx_high
Data[11]	RX坐标低字节	rx_low
Data[12]	RY坐标高字节	ry_high
Data[13]	RY坐标低字节	ry_low
Data[14]	RZ坐标高字节	rz_high
Data[15]	RZ坐标低字节	rz_low
Data[16]	结束帧	0XFA

设定工具坐标系 (-14 , -27 , 275 , -89.5 , 0.7 , -90.7) ,

串口发送示例 : FE FE 14 81 FF 74 FE EE 0A C1 DD 05 00 48 DC 95 FA

无返回值

42).设置世界坐标系

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X14
Data[3]	指令帧	0X83
Data[4]	X坐标高字节	x_high
Data[5]	X坐标低字节	x_low
Data[6]	Y坐标高字节	y_high
Data[7]	Y坐标低字节	y_low
Data[8]	Z坐标高字节	z_high
Data[9]	Z坐标低字节	z_low
Data[10]	RX坐标高字节	rx_high
Data[11]	RX坐标低字节	rx_low
Data[12]	RY坐标高字节	ry_high
Data[13]	RY坐标低字节	ry_low
Data[14]	RZ坐标高字节	rz_high
Data[15]	RZ坐标低字节	rz_low
Data[16]	结束帧	0XFA

设定世界坐标系 (-14 , -27 , 275 , -89.5 , 0.7 , -90.7) ,

串口发送示例：FE FE 14 83 FF 74 FE EE 0A C1 DD 05 00 48 DC 95 FA

无返回值

43).获取工具坐标系

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X02
Data[3]	指令帧	0X82
Data[4]	结束帧	0XFA

串口发送示例：FE FE 02 82 FA

返回数据结构

数据域	说明	数据
Data[0]	返回识别帧	0XFE
Data[1]	返回识别帧	0XFE
Data[2]	返回数据长度帧	0X14
Data[3]	返回指令帧	0X82
Data[4]	X坐标高字节	x_high
Data[5]	X坐标低字节	x_low
Data[6]	Y坐标高字节	y_high
Data[7]	Y坐标低字节	y_low
Data[8]	Z坐标高字节	z_high
Data[9]	Z坐标低字节	z_low
Data[10]	RX坐标高字节	rx_high
Data[11]	RX坐标低字节	rx_low
Data[12]	RY坐标高字节	ry_high
Data[13]	RY坐标低字节	ry_low
Data[14]	RZ坐标高字节	rz_high
Data[15]	RZ坐标低字节	rz_low
Data[16]	结束帧	0XFA

串口返回示例：FE FE 14 82 FF 74 FE EE 0A C1 DD 05 00 48 DC 95 FA

x_high：数据类型byte

计算方式：x坐标乘以10 先转换为int类型 再取十六进制高字节

x_low：数据类型byte

计算方式：x坐标乘以10 先转换为int类型 再取十六进制低字节

（y轴坐标z轴坐标同理）

rx_high：数据类型byte

计算方式：rx坐标乘以100 先转换为int类型 再取十六进制高字节

rx_low：数据类型byte

计算方式：rx坐标乘以100 先转换为int类型 再取十六进制低字节

(ry轴坐标rz轴坐标同理)

44).获取世界坐标系

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X02
Data[3]	指令帧	0X84
Data[4]	结束帧	0XFA

串口发送示例：FE FE 02 84 FA

返回数据结构

数据域	说明	数据
Data[0]	返回识别帧	0XFE
Data[1]	返回识别帧	0XFE
Data[2]	返回数据长度帧	0X14
Data[3]	返回指令帧	0X84
Data[4]	X坐标高字节	x_high
Data[5]	X坐标低字节	x_low
Data[6]	Y坐标高字节	y_high
Data[7]	Y坐标低字节	y_low
Data[8]	Z坐标高字节	z_high
Data[9]	Z坐标低字节	z_low
Data[10]	RX坐标高字节	rx_high
Data[11]	RX坐标低字节	rx_low
Data[12]	RY坐标高字节	ry_high
Data[13]	RY坐标低字节	ry_low
Data[14]	RZ坐标高字节	rz_high
Data[15]	RZ坐标低字节	rz_low
Data[16]	结束帧	0XFA

串口返回示例：FE FE 14 84 FF 74 FE EE 0A C1 DD 05 00 48 DC 95 FA

45).设置法兰基坐标系

数据域	说明	数据
Data[0]	识别帧	0XFE

Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X03
Data[3]	指令帧	0X85
Data[4]	RFTYPE	0x00/0x01
Data[5]	结束帧	0XFA

串口发送示例：FE FE 03 85 00 FA

RFTYPE：数据类型byte

取值范围：0~1 BASE = 0；WORLD = 1；

RFTYPE::BASE为将机器人基座作为基坐标，RFTYPE::WORLD为将世界坐标系作为基坐标。

46).获取法兰基坐标系

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X02
Data[3]	指令帧	0X86
Data[4]	结束帧	0XFA

串口发送示例：FE FE 02 86 FA

返回数据结构

数据域	说明	数据
Data[0]	返回识别帧	0XFE
Data[1]	返回识别帧	0XFE
Data[2]	返回数据长度帧	0X03
Data[3]	返回指令帧	0X86
Data[4]	RFTYPE	0x00/0x01
Data[5]	结束帧	0XFA

串口返回示例：FE FE 03 86 00 FA

47).设置末端坐标系

数据域	说明	数据
Data[0]	返回识别帧	0XFE
Data[1]	返回识别帧	0XFE
Data[2]	返回数据长度帧	0X03
Data[3]	返回指令帧	0X89
Data[4]	EndType	0x00/0x01
Data[5]	结束帧	0XFA

串口发送示例：FE FE 03 89 00 FA

无返回值

EndType：数据类型byte

取值范围：0~1 FLANGE = 0；TOOL = 1；

EndType::FLANGE为将末端设置为法兰，EndType::TOOL为将末端设置为工具末端。

48).获取末端坐标系

数据域	说明	数据
Data[0]	识别帧	0XFE
Data[1]	识别帧	0XFE
Data[2]	数据长度帧	0X02
Data[3]	指令帧	0X8A
Data[4]	结束帧	0XFA

串口发送示例：FE FE 02 8A FA

返回数据结构

数据域	说明	数据
Data[0]	返回识别帧	0XFE
Data[1]	返回识别帧	0XFE
Data[2]	返回数据长度帧	0X03
Data[3]	返回指令帧	0X8A
Data[4]	EndType	0x00/0x01
Data[5]	结束帧	0XFA

串口返回示例：FE FE 03 8A 00 FA

4.联系我们

如有任何需求帮助，可根据以下方式联系我们。

微信号：大象机器人小管家（18123841923）



深圳市大象机器人科技有限公司

地址：深圳市南山区桃源街道留仙大道南山云谷创新产业园二期七栋二楼

邮箱：support@elephantrobotics.com

电话：+86(0755)-8696-8565（工作日9:30-18:30）

网址：www.elephantrobotics.com