

myCobot 通讯协议与 API

1. 上层 API 指令解析

1) Set Servo Encoder (byte servo_no, int servo_encoder, int servo_sp);

控制指定关节，以指定速度转动至指定电位值：

关节序号 = servo_no, 取值范围 1-6;

指定电位值 = servo_encoder, 取值范围 0-4096;

指定速度 = servo_sp, 取值范围 0-4000,;

2) Set Angle (byte servo_no, int angle, int sp);

控制指定关节，以指定速度旋转到对应角度：

关节序号 = servo_no, 取值范围 1-6;

指定角度 = angle, 取值范围 -170° — $+170^{\circ}$;

指定速度 = sp, 取值范围 0-4000;

3) Set Servos Encoder (int servo_encoder_1, int servo_encoder_2, int servo_encoder_3, int servo_encoder_4, int servo_encoder_5, int servo_encoder_6, int servo_sp);

指定六个关节舵机以指定速度转动至指定电位值：

1号舵机电位值 = servo_encoder_1, 取值范围 0 – 4096;

2号舵机电位值 = servo_encoder_2, 取值范围 0 – 4096;

3号舵机电位值 = servo_encoder_3, 取值范围 0 – 4096;

4号舵机电位值 = servo_encoder_4, 取值范围 0 – 4096;

5号舵机电位值 = servo_encoder_5, 取值范围 0 – 4096;

6号舵机电位值 = servo_encoder_6, 取值范围 0 – 4096;

指定速度 = servo_sp, 取值范围 0 - 4000;

4) Calibrate Servo (byte servo_no);

设定当前电位值为舵机角度零点：

关节舵机编号 = servo_no;

5) Set Servo Data (byte servo_no, byte servo_state, byte servo_data);

设定关节舵机的相关参数，可修改参数包括：PID、舵机指示灯、最小启动力：

指定关节舵机序号 = servo_no, 取值范围 1 – 6;

可修改参数地址 = servo_state, 如表 1-1;

可修改参数值 = servo_data, 如表 1-1;

表 1-1

| 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 |

6) Release All Servos();

控制舵机扭力输出为 0, 使机械臂处于卸力状态;

7) Get Angle Encoder (byte joint_no);

读取关节舵机当前的电位值:

关节舵机序号 = joint_no;

8) Get Angle (byte joint_no);

获取当前舵机的角度值:

关节舵机序号 = joint_no;

9) Set RGB (byte R, byte G, byte B);

设定 atom 屏幕的 RGB 灯的颜色:

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;

10) Set Color(byte color);

设定 atom 屏幕的 RGB 灯的颜色:

状态值取值 = color, 取值范围如下表:

| 取值 | 0 | 1 | 2 | 3 | 4 |
|----|----|----|----|----|----|
| 颜色 | 红色 | 绿色 | 蓝色 | 黄色 | 彩色 |

11) Set Atom Pin Mode (byte pin, byte mode);

设置 atom 指定引脚的输入输出状态:

指定引脚序号 = pin;

指定状态值 = mode, 取值 1 位输出、0 为输入;

12) Set Atom Digital Write (byte pin, byte data);

设置 atmo 指定引脚输出的状态值:

指定引脚序号 = pin;

指定状态值 = data;

13) Check Header();

检查接收到的数据头文件是否为指定数据, 是则处理数据, 否则继续读取;

14) rFlush Serial();

读取当前串口接收区缓存;

2. 底层通讯数据结构

1) 串口总线说明

总线接口: Serial2

波特率: 9600, 19200, 57600, 115200, 1000000(默认)

数据位: 8

奇偶校验: 无

停止位: 1

2) 命令帧说明及单一指令解析

主机 Basic 向从机发送数据, 从基接收到数据后进行解析, 如包含返回值的指令, 从机会在 0.5s 内返回给主机。

| 类型 | 数据描述 | 数据长度 | 说明 |
|-----|--------|------|-----------------|
| 帧命令 | 头字节 0 | 1 | 帧头识别, 0XFA |
| | 头字节 1 | 1 | 帧头识别, 0XFA |
| | 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 |
| 帧数据 | 数据 | 0-16 | 命令附带数据, 视不同命令而定 |
| | 结束字节 | 1 | 停止位, 0XFE |

3) 单一指令解析

A. 控制指定关节，以指定速度转动至指定电位值

| 数据域 | 说明 | 数据 |
|---------|---------|------------|
| Data[0] | 识别帧 | 0XFA |
| Data[1] | 识别帧 | 0XFA |
| Data[2] | 数据长度帧 | 0X07 |
| Data[3] | 指令帧 | 0X20 |
| Data[4] | 关节舵机 ID | Servo_no |
| Data[5] | 位置低字节 | Angle_low |
| Data[6] | 位置高字节 | Angle_high |
| Data[7] | 速度低字节 | Speed_low |
| Data[8] | 速度高字节 | Speed_high |
| Data[9] | 结束帧 | 0XFE |

B. 设定指定关节舵机位置电位值为 2048，角度为 0°

| 数据域 | 说明 | 数据 |
|---------|---------|----------|
| Data[0] | 识别帧 | 0XFA |
| Data[1] | 识别帧 | 0XFA |
| Data[2] | 数据长度帧 | 0X03 |
| Data[3] | 指令帧 | 0X25 |
| Data[4] | 关节舵机 ID | Servo_no |
| Data[5] | 结束帧 | 0XFE |

C. 设定制定关节舵机的参数

| 数据域 | 说明 | 数据 |
|---------|---------|-------------|
| Data[0] | 识别帧 | 0XFA |
| Data[1] | 识别帧 | 0XFA |
| Data[2] | 数据长度帧 | 0X05 |
| Data[3] | 指令帧 | 0X24 |
| Data[4] | 关节舵机 ID | Servo_no |
| Data[5] | 位置低字节 | Servo_state |
| Data[6] | 位置高字节 | Servo_data |
| Data[7] | 结束帧 | 0XFE |

D. 读取制定关节舵机的电位值

| 数据域 | 说明 | 数据 |
|---------|---------|----------|
| Data[0] | 识别帧 | 0XFA |
| Data[1] | 识别帧 | 0XFA |
| Data[2] | 数据长度帧 | 0X07 |
| Data[3] | 指令帧 | 0X20 |
| Data[4] | 关节舵机 ID | Servo_no |
| Data[5] | 结束帧 | 0XFE |

从机返回数据结构

| 数据域 | 说明 | 数据 |
|---------|---------|------------|
| Data[0] | 识别帧 | 0XFA |
| Data[1] | 识别帧 | 0XFA |
| Data[2] | 关节舵机 ID | Servo_no |
| Data[3] | 位置低字节 | Angle_low |
| Data[4] | 位置高字节 | Angle_high |
| Data[5] | 结束帧 | 0XFE |

E. 设定 atom 的显示屏 RGB 灯颜色

| 数据域 | 说明 | 数据 |
|---------|-------|------|
| Data[0] | 识别帧 | 0XFA |
| Data[1] | 识别帧 | 0XFA |
| Data[2] | 数据长度帧 | 0X05 |
| Data[3] | 指令帧 | 0X33 |
| Data[4] | R | R |
| Data[5] | G | G |
| Data[6] | B | B |
| Data[7] | 结束帧 | 0XFE |

F. 设定 Atom 的引脚输入输出状态

| 数据域 | 说明 | 数据 |
|---------|--------|------|
| Data[0] | 识别帧 | 0XFA |
| Data[1] | 识别帧 | 0XFA |
| Data[2] | 数据长度帧 | 0X04 |
| Data[3] | 指令帧 | 0X33 |
| Data[4] | 引脚序号 | Pin |
| Data[5] | 输入输出状态 | Mode |
| Data[6] | 结束帧 | 0XFE |

G. 设定 Atom 的引脚输出状态值

| 数据域 | 说明 | 数据 |
|---------|-------|------|
| Data[0] | 识别帧 | 0XFA |
| Data[1] | 识别帧 | 0XFA |
| Data[2] | 数据长度帧 | 0X04 |
| Data[3] | 指令帧 | 0X33 |
| Data[4] | 引脚序号 | Pin |
| Data[5] | 输出状态 | Data |
| Data[6] | 结束帧 | 0XFE |