

3IRobotix 教育机器人底盘 通讯协议

使用者需注意雷达与底盘协议不一致，且底盘协议为小端模式，雷达协议为大端模式，解析数据时需特别注意大小端问题。

底盘协议

1. 帧协议

- 帧定义（小端模式）

SSID	密码	IP	端口
rebot	12345678	192.168.5.1	5000

帧协议	包头	命令字	长度	数据	和校验
字节数	2bytes	1bytes	2bytes	不定长	2bytes
说明	0x55 0xAA	[0 ~ 255]	data.length + 2	data.length	和校验

- 帧结构体

```
typedef struct {
    uint16_t    sync = 0x55AA;
    uint8_t     cmd;
    uint16_t    length;
    uint8_t     data[0];
} MSG_t;
```

- 校验算法

整包校验，累加和校验

```
uint32_t CRC32_Calculate(uint8_t * data, uint16_t len)
{
    long packet_sum = 0;
    while ( len --)
    {
        packet_sum += *data++;
    }
    return (uint16_t)packet_sum;
}
```

2. 通讯协议

- CMD_CHASSIS(0x01)

命令字名词	命令字	类型	数据内容	说明
CMD_CHASSIS	1	uint8_t	左碰撞	1触发
	1	uint8_t	右碰撞	1触发
	1	float32_t	里程计x	单位m，机器人正前X正方向
	1	float32_t	里程计y	单位m，机器人左侧y正方向
	1	float32_t	里程计yaw角	单位rad，机器人逆时针方向
	1	float32_t	线速度	单位m/s，机器人正前X正方向
	1	float32_t	角速度yaw	单位rad/s，机器人逆时针方向

```
typedef struct {
    uint8_t bumper_left;
    uint8_t bumper_right;
    float   x;
    float   y;
    float   yaw;
    float   x_vel;
    float   yaw_vel;
} TChassisSensor;
```

- CMD_ROBOT_CONTROL_TYPE(0x02)

命令字名词	命令字	类型	数据内容	说明
CMD_ROBOT_CONTROL_TYPE	2	uint8_t	机器人控制模式	1使能，0失能

```
typedef struct {
    uint8_t mode;
} TRbotoControlType;
```

- CMD_MOTOR_CONTROL_VELOCITY(0x03)

命令字名词	命令字	类型	数据内容	说明
CMD_MOTOR_CONTROL_VELOCITY	3	float32_t	X方向速度	m/s
	3	float32_t	角速度	rad/s

```
typedef struct {
    float   x_vel;
    float   yaw_vel;
} TMotorVelocity;
```

雷达协议

1.帧协议

- 帧定义（大端模式）

SSID	密码	IP	端口
rebot	12345678	192.168.5.1	5001

帧协议	包头	总长	协议版本	类型	命令字	数据长	数据	和校验
名词	head	len	address_code	type	cmd	msg_len	data	check
字节数	1bytes	2bytes	1bytes	1bytes	1bytes	2bytes	不定长	2bytes
说明	0xaa	包头 + 数据长	0x11	预留	消息类型	数据长 + 校验长	传输内容	校验
模板帧	aa	00 09	11	61	ae	00 01	29	01 fd

- 帧结构体

```
class MSG_t
{
public:
    uint8_t      head;
    uint16_t     len;
    uint8_t      address_code;
    uint8_t      type;
    uint8_t      cmd;
    uint16_t     msg_len;
    uint8_t      data[0];
};
```

- 校验算法

整包校验，累加和校验

```
uint32_t CRC32_Calculate(uint8_t * data, uint16_t len)
{
    long packet_sum = 0;
    while ( len --)
    {
        packet_sum += *data++;
    }
    return (uint16_t)packet_sum;
}
```

2.通讯协议

• 0xAD

命令字	偏移地址	类型	数据内容	说明
0xAD	0	uint8_t	雷达速度	单位 0.2r/s
0xAD	1	uint16_t	偏移角度	0.01度
0xAD	3	uint16_t	起始点角度	0.01度
0xAD	5	uint16_t	结束点角度	0.01度
---	---	---	---	---
0xAD	3n	uint8_t	强度信息	
0xAD	3n +1	uint16_t	点距离	0.0001m

```
class MSG_0xAD_t : public MSG_t
{
    public:
    uint8_t      speed;
    uint16_t     offset_angle;
    uint16_t     start_angle;
    uint16_t     end_angle;
    struct {
        uint8_t  signal;
        uint16_t distance;
    }data[0];
};
```

• 0xAE

命令字	偏移地址	类型	数据内容	说明
0xAD	0	uint8_t	雷达速度	单位 0.05r/s

```
class MSG_0xAE_t : public MSG_t
{
    public:

    uint8_t      speed;
};
```