

# 红外串口通信说明

## 概述

通过串口和IR模块进行通信。移植串口协议需要修改 `ir_uart_msg_deal.c` 和 `ir_uart_msg_deal.h` 文件，详细参考 [4.串口协议移植说明](#)

## 1. 串口初始化

初始化串口硬件，buf申请，目前CI110X模块这边默认设置的波特率为9600。

```
/**
 * @brief IR 串口通信初始化
 *
 * @return int -1 错误, 0 成功
 */
int ir_uart_buff_init(void);
```

## 2. 数据发送

发送串口数据，发送数据或者命令,其中的 `msg_type` 为两个字节

目前支持的命令为：

```
/**
 * @brief unsigned short
 * 如果需要发送电平数据, input 数据需要为 ushort 类型数组数据,
 * 且原始电平数据需要除以 IR_DATA_DIV_COEFFICIENT 系数, 目前为2
 */
typedef enum
{
    IR_CMD_RECEIVE_START = 0x1000, //开始接收数据
    IR_CMD_RECEIVE_STOP,           //结束接收
    IR_CMD_LEVEL_SEND,             //发送电平数据
    IR_CMD_LEVEL_MATCH,           //匹配电平数据
    IR_CMD_LEVEL_LEARN,           //学习电平数据
    IR_CMD_DEV_RECV_IR_DATA,       //设备接收到的红外数据
    IR_CMD_DEV_MATCH_OK,          //设备匹配成功, 设备返回数据, fill_data为匹配ID
    IR_CMD_DEV_MATCH_ERR,         //设备匹配失败, 设备返回匹配失败
    IR_CMD_SET_AIR_CODE_ID,        //设置空调红外码ID, ID号放到fill_data
    IR_CMD_ACK = 0xFAFA,          //ACK 回复信息
} ir_uart_cmd_t;
```

发送数据接口如下：

```

/**
 * @brief 发送消息到设备端
 *
 * @param msg_type 消息类型
 * @param input 数据包, 为NULL时, 只发对应msg_type
 * @param len 数据包长度, 为0时, 只发对应msg_type
 * @param fill_data 填充信息
 * @return int -1 错误, 返回发送数据长度
 */
int send_server_data_to_device(ir_uart_cmd_t msg_type, const unsigned char *input,
int len,unsigned int fill_data);

```

### 3. 数据接收、解码

接收串口数据, 解析出正确指令和数据包。

```

/**
 * @brief 串口数据解析
 *
 * @param char 串口数据
 * @param data_len 数据长度
 * @return int 0 解析成功, -1 解析错误
 */
int uart_data_decode(unsigned char* uart_data, int data_len);

```

### 4.串口协议移植说明

移植时需要修改 `ir_uart_msg_deal.c` 文件中部分接口。

```

/* 需要修改的接口 */
static int write_device_data_interface(unsigned char *file_data, int file_lenth);
static void *ir_malloc(unsigned int want_size);
static int ir_uart_port_hw_init(void);

```

数据解析时需要调用 `int uart_data_decode(unsigned char* uart_data, int data_len)` 解析串口接收到的数据, 用户可以根据解析出的命令自己添加处理方法, 下面为部分示例代码, 详情查看 `ir_uart_msg_deal.c` 中完整示例代码。

```
switch (msg_type)
{
    /* 接收到ACK */
    case IR_CMD_ACK:
    {
        IR_UART_DEBUG("IR_CMD_ACK");
        break;
    }

    default:
        IR_UART_DEBUG("not deal");
        break;
}
```