



苏州咖乐美电器有限公司

SIP Kalerm Electric Appliances Corporation

保密文档

Confidential Material

版本历史：

作者	版本	日期	说明
taoyangjun	V1.0	2021.05.14	K95 外接开放协议 V1.0 版本
taoyangjun	V1.1	2021.08.11	修改设备状态的协议，传输指令长度需要控制在 32 字节以下

目录

- 1 说明.....2
 - 1.1 版权声明2
 - 1.2 文档目的3
- 2 功能概述.....3
- 3 通信设置.....3
- 4 串口协议.....4
 - 4.1 协议格式4
 - 4.2 功能协议5
 - 咖啡制作5
 - 配方咖啡制作6
 - 取消制作8
 - 放杯确认8
 - 清洗冲泡器.....8
 - 清洗奶沫器.....9
 - 状态查询9
- 附件一：故障列表..... 11
- 附录二：附校验码生成代码..... 12

1 说明

1.1 版权声明

本文档属于技术保密资料，苏州工业园区咖乐美咖啡机科技有限公司对本文档拥有最终解释权，未经咖乐美咖啡机科技有限公司书面许可，严禁任何组织或个人复制传播本文档内容，违者追究责任。

1.2 文档目的

本文档描述了对外提供开放串口协议，第三方可通过文档中定义的协议规则，对接我们的 K95 机型咖啡机，实现对咖啡机的部分功能控制。

2 功能概述

1. 控制咖啡机做咖啡
2. 反馈咖啡的制作进度（开始，进度，结束状态）
3. 反馈咖啡机的状态
4. 修改咖啡配方

3 通信设置

波特率：38400

数据位：8

停止位：1

奇偶校验：无

4 串口协议

4.1 协议格式

控制报文：

帧标识	地址	控制命令	控制内容	校验位	结束符
:	01	02	XXXX	LRC	\r\n

帧标识：固定冒号“:”

地址：固定“01”

控制命令：根据不同功能，控制命令不同，详见下面功能协议

控制内容：不同内容，长度和内容不同

校验位：校验数字段为“地址+控制命令+控制内容”，[附录二：附校验码生成代码](#)

结束符：固定“\r\n”

报文内容格式为字符串

响应报文：

帧标识	地址	控制命令	响应数据	校验位	结束符
:	01	02	响应说明	LRC	\r\n

响应说明：00 代表发送成功

01 代表机器正忙

02 代表串口指令错误

4.2 功能协议

咖啡制作

控制报文：

帧标识	地址	控制命令	控制内容	校验位	结束符
:	01	01	XXXX	LRC	\r\n

响应报文：

帧标识	地址	控制命令	响应数据	校验位	结束符
:	01	01	响应说明	LRC	\r\n

系统默认 23 种咖啡

23 种咖啡	索引值	控制内容
意式咖啡	1	0001
美式咖啡	2	0002
咖啡（意式咖啡）	9	0009
卡布奇诺	4	0004
玛琪雅朵	5	0005
拿铁	6	0006
白咖啡	14	000e
芮斯崔朵	10	000a

意式玛奇朵	13	000d
奶油咖啡	12	000c
双杯意式咖啡	51	0033
双杯卡布奇诺	55	0037
双杯拿铁	57	0039
双杯玛琪雅朵	56	0038
双杯白咖啡	59	003b
双杯咖啡	52	0034
双杯芮斯崔朵	53	0035
双杯意式玛奇朵	58	003a
双杯奶油咖啡	54	0036
热水	3	0003
牛奶	100	0064
奶沫	101	0065
大壶咖啡	11	000b

配方咖啡制作

控制报文：

帧标识	地址	控制命令	控制内容	校验位	结束符
:	01	02	咖啡配方	LRC	\r\n

响应报文：

帧标识	地址	控制命令	响应数据	校验位	结束符
:	01	02	响应说明	LRC	\r\n

K95 配方定义

协议名	备注	字符数
配方类型	类型有 8 种， 01-08	2
咖啡量	毫升 0-240	4
咖啡温度	低/中/高 0/1/2	2
咖啡浓度	清淡/适中/浓郁 0/1/2	2
热水量	毫升 0~450	4
热水温度	低/中/高 0/1/2	2
牛奶时间	秒 0~120	2
奶沫时间	秒 0~120	2
是否预煮	1 是 0 否	2
咖啡增强	1 是 0 否	2
咖啡奶沫同时出	1 同时出， 0 不同时	2
出品顺序	0:先奶后咖啡 1:先咖啡后奶	2



K95配方.xlsx

配方指令详情见：

取消制作

控制报文：

帧标识	地址	控制命令	控制内容	校验位	结束符
:	01	03	0001000C	LRC	\r\n

响应报文：

帧标识	地址	控制命令	响应数据	校验位	结束符
:	01	03	响应说明	LRC	\r\n

放杯确认

控制报文：准备要放杯了

帧标识	地址	控制命令	控制内容	校验位	结束符
:	01	06	0001	LRC	\r\n

响应报文：

帧标识	地址	控制命令	响应数据	校验位	结束符
:	01	06	响应说明	LRC	\r\n

清洗冲泡器

控制报文：

帧标识	地址	控制命令	控制内容	校验位	结束符
:	01	04	0001010A	LRC	\r\n

响应报文：

帧标识	地址	控制命令	响应数据	校验位	结束符
:	01	04	响应说明	LRC	\r\n

清洗奶沫器

控制报文：

帧标识	地址	控制命令	控制内容	校验位	结束符
:	01	04	0001040A	LRC	\r\n

响应报文：

帧标识	地址	控制命令	响应数据	校验位	结束符
:	01	04	响应说明	LRC	\r\n

状态查询

控制报文：

帧标识	地址	控制命令	控制内容	校验位	结束符
:	01	05	0000000D	LRC	\r\n

响应报文:

帧标识	地址	控制命令	响应数据	校验位	结束符
:	01	05	系统状态	LRC	\r\n

系统状态:

示例: :0105 1000(系统状态) 0000 (制作状态)

0000 (制作进度) 0000 (制作进度)

0000 (故障信息) B (保养信息) 3A\r\n

数据	定义	索引	值
0-3	系统状态 (16 进制表示 如: AD10)	[0]	0; 初始化状态 1; 空闲状态 2; 运行状态 3; 关机状态 8;//正在制作咖啡 9;//冲泡器清洗和奶沫器清洗, 目前没法区分 A;//冲泡器药片清洗 B;//奶沫器药片清洗 C;//除垢清洗 D;//清空管路
		[1]	应用状态, Application state;
		[2]	应用子状态, Sub state;
		[3]	
4-7	制作状态	[4]	饮品类型: 1-意式咖啡、2-美式咖啡、3-热水、4-卡布奇诺、 5-玛琪雅朵、6-拿铁咖啡、7-热牛奶、8-奶沫
		[5]	饮品制作状态: 0-正在制作、1-暂停、2-取消、3-完成
		[6]	当前任务进度索引: 0-null 1-出咖啡 2-出热水 3-出牛奶 4-出奶沫 5-同时出, 即水、牛奶、奶沫、咖啡灯两种以上同时出 6-磨豆 * 特别注意, 当制作进度为磨豆时候, 目前不能提

			供实际进度，仅代表当前口味，0-清淡、1-适中、2-浓郁，如果进度索引值发生变化，代表磨豆完成
		[7]	
8-11	制作进度	[8]-[11]	当前实际进度值
12-15	制作进度	[12]-[15]	进度执行完成的目标值
16-19	故障信息	[16]-[19]	附件一：故障列表
20	保养信息 (16 进制表示 如：F)	[20]	F=>1111 第一位：1 冲泡器需要清洗， 第二位：1 奶沫器需要清洗， 第三位：1 咖啡机需要除垢， 第四位：1 需要更换滤芯

附件一：故障列表

如果返回的故障信息是 0003 的话，

转为二进制为 0000 0000 0000 0011

然后反向即 1100 0000 0000 0000，按照此索引值对应下表

(0：无故障，1 有故障)

索引	说明	描述
【0】	安装蓄水盘	蓄水盘没有安装
【1】	水箱缺水	水箱缺水

【2】	磨豆异常	磨豆异常，无豆检测有豆子，可能是磨豆少、磨豆机不转等问题。
【3】	蓄水盘满	蓄水盘废水满
【4】	咖啡渣满	咖啡机满渣
【5】	系统缺水	系统缺水
【6】	冲泡器故障	冲泡器下压位置异常
【7】		冲泡器复位位置异常
【8】	高温报警	咖啡机电热盘温度过高，高于设定上限温度
【9】	低温报警	咖啡机开机温度检测环境温度低于 0°。需要温度升到 1° C 以上才可以使用
【10】	ERROR 2	NTC 损坏，阻值无穷大
【11】	咖啡豆用尽	豆盒中没有豆了
【12】	ERROR 1	电热盘保险丝烧断，加热 NTC 阻值无变化
【13】	压力过大	制作咖啡时管路压力过大
【14】	ERROR 6	连续 3 次系统补水，仍补不上水
【15】	牛奶已用尽	A & B 中没有牛奶

附录二：附校验码生成代码

Java 代码

```
public static String callLrcCommon(String data) {
    try {
        if (data.length() % 2 != 0) {
            data = data + "0";
        }
        int total = 0;
```

```

        int len = data.length();
        int num = 0;
        while (num < len) {
            String s = data.substring(num, num + 2);
            total += Integer.parseInt(s, 16);
            num += 2;
        }
        total = ~total + 1;
        String checkSum = Integer.toHexString(total & 255).toUpperCase();

        while (checkSum.length() < 2) {
            checkSum = "0" + checkSum;
        }
        return checkSum;
    } catch (Exception e) {
    }
    return "";
}

```

C 语言代码

```

uint8 CalculateLrc(char *MsgBuf, uint8 DataLen)
{
    uint8 Lrc;

    Lrc = 0;
    DataLen = DataLen >> 1; //字符长度转换成数字长度
    while (DataLen-->0)
    {
        Lrc = Lrc + TwoChar2Hex(MsgBuf);
        MsgBuf = MsgBuf + 2;
    }
    Lrc = 0xff - Lrc; //求补码
    Lrc++;

    return Lrc;
}

```

```

uint8 TwoChar2Hex(char *pChar) //把 2 个字符转换成数字
{
    uint8 HexData;

    HexData = Char2Hex(*pChar) << 4;
    pChar++;
}

```

```
HexData = HexData + Char2Hex(*pChar);

return HexData;
}
```

```
uint8 Char2Hex(char temp)
{
    if (temp < 'A')
    {
        return (temp & 0x0f);
    }
    else
    {
        return (temp - 0x37);
    }
}
```