C#调用咖啡机api实例流程，其他程序原理类似。

namespace coffee\_api

1、先实例化MegaCoffee\_api类，如下：

public MegaCoffee\_api coffee = new MegaCoffee\_api();

2、初始化函数

coffee.init\_serial("com8",115200); //串口初始化；；

3、获取产品列表

coffee.get\_product\_list();

由于是回调函数不能直接获取返回值，判断productlistAvailable 是否为true,

当productListAvailable==true及可读取id\_list，name\_list产品序列id及产品名称

4、开始制作

public void start\_product(string idStr);参数为id\_list中的某一项

5、请求响应状态：

public int ResponseCode;

定义如下：

**enum** ResponseCode {

UNKNOWN\_RESPONSE\_CODE =0；//未知响应代码

SUCCESS **=** 1；//请求成功

INVALID\_PARAMETER **=** 2；//给定参数之一无效

DBUS\_ADAPTER\_ERROR **=** 3;//远程api组件与应用程序之间的通信失败

GENERAL\_ERROR =4；//由于没有进一步指定的问题，请求失败

SYSTEM\_BUSY **=** 100; //应用程序当前正在完成另一个请求，无法执行此请求

UNKNOWN\_PRODUCT\_ID **=** 101; //请求中指定的产品ID未知

PRODUCT\_NOT\_AVAILABLE =102；//产品不可得

}

5、产品制作状态：

public int product\_status;

0x00开始制作；0x01表示制作完成,0x02制作错误

1. 咖啡机状态信息coffee\_machine\_system\_status

定义如下：

**enum** coffee\_machine\_system\_status{

UNKNOWN\_RESPONSE\_CODE =0；//未知响应代码

INVALID\_PARAMETER **=** 2；//给定参数之一无效

DBUS\_ADAPTER\_ERROR **=** 3;//远程api组件与应用程序之间的通信失败

GENERAL\_ERROR =4；//由于没有进一步指定的问题，请求失败

SYSTEM\_BUSY **=** 100; //应用程序当前正在完成另一个请求，无法执行此请求

UNKNOWN\_PRODUCT\_ID **=** 101; //请求中指定的产品ID未知

PRODUCT\_NOT\_AVAILABLE =102；//产品不可得

}

7、事件消息

public int msg\_code;

enum msg\_code

{//使用error-list.xlsx里的错误列表

}

主要事件状态划分类别：

public struct coffee\_machine\_status

{

public byte BoilerControllerOK;

public byte StatusManagerOK;

public byte BeanHopperPresent;

public byte BeanHopperLevelOK;

public byte GroundsDrawerOK;

public byte MilkTemperatureOK;

public byte MilkLevelOK;

public byte CleaningOK;

public byte PistonError;

public byte CommunicationError;

public byte ComponentError;

public byte ConfigurationError;

public byte FlowError;

public byte BrewChamberEmpty;

public byte BrewChamberTooFull;

public byte MilkEmptyDuringProduct;

public byte SafetyReedDuringProduct;

public byte ChassisFanFailure;

}

所有状态0为正常

BoilerControllerOK //锅炉状态

StatusManagerOK //状态管理器工作正常

BeanHopperPresent //豆子漏斗就位了

BeanHopperLevelOK //豆子漏斗高度ok

GroundsDrawerOK //抽屉Ok，不Ok

MilkTemperatureOK //牛奶温度Ok,高

MilkLevelOK //牛奶Ok，牛奶低

CleaningOK //需要清洁，不需要清洁

PistonError //检测到活塞错误

CommunicationError //通讯错误

ComponentError //检测到组件错误

ConfigurationError //配置错误

FlowError //水流错误

BrewChamberEmpty //研磨后，酿造室是空的。重新装满豆子漏斗确保豆子漏斗锁打开

BrewChamberTooFull //研磨后无法关闭酿造室。为产品指定的咖啡量可能太高，或者冲泡室电机出现问题

MilkEmptyDuringProduct //制作过程中牛奶低

SafetyReedDuringProduct //安全停止

ChassisFanFailure //机箱风扇工作正常，异常

详情如下：

BoilerControllerOK = 0;

BoilerControllerOK = 2;

BoilerControllerOK = 0;

BoilerControllerOK = 4;

BoilerControllerOK = 5;

BoilerControllerOK = 6;

BoilerControllerOK = 0;

StatusManagerOK = 0;

StatusManagerOK = 9;

BeanHopperPresent = 10;

BeanHopperPresent = 11;

BeanHopperLevelOK = 0;

BeanHopperLevelOK = 13;

BeanHopperLevelOK = 14;

GroundsDrawerOK = 16;

GroundsDrawerOK = 0;

GroundsDrawerOK = 18;

GroundsDrawerOK = 19;

MilkTemperatureOK = 0;

MilkTemperatureOK = 21;

MilkTemperatureOK = 22;

MilkLevelOK = 0;

MilkLevelOK = 24;

MilkLevelOK = 25;

CleaningOK = 0;

CleaningOK = 27;

CleaningOK = 28;

PistonError = 1;

CommunicationError = 1;

ComponentError = 1;

ConfigurationError = 1;

FlowError = 1;

BrewChamberEmpty = 1;

BrewChamberTooFull = 1;

MilkEmptyDuringProduct = 1;

SafetyReedDuringProduct = 1;

ChassisFanFailure = 1;