Device控制器通信规范

|  |  |  |
| --- | --- | --- |
| 版本 | 修订内容 | 日期 |
| 1.0 |  | 2020/8/18 |
| 1.1 | 1. 增加指令的详细说明 2. 耗时指令直接返回忙碌状态 | 2020/8/18 |

Chiv

# 概述

举凡我司生产的各类可控制器件都应遵循本规范。本规范是一种协议层规范，规定主从设备之间的通信方式。可控类器件一般都由低端单片机实现，性能相对较弱，因此本规范的主要目的是提供一致性的通信协议，其它如加密、压缩等，都不在本规范的考虑范围内。

# 特性

规范的目的是制定统一的通信协议，降低设备维护难度，提高稳定性。鉴于此，本规范支持以下特性。

* 纯文本
* 主从式，强应答

规定，使用本规范时必须存在主机，只有主机能够发起通信。强应答，表示从机收到数据后必须给与应答。

* 支持特性查询

主机能够查询从机的状态，取得从机的特性。

# 规范

## 定义

在介绍规范之前，我们定义几个术语，以方便描述。

1. 指令

一个指令由以下ASCII控制字符和可打印字符构成。尖括号内位必须按行，中括号内为可选项。

<**STX**>[part]<**US**><action><**US**>[parameters,delim is **US**]<**ETX**>[chksum]<**EOT**>

chksum为校验和，校验和只针对STX和ETX之间的字符，包括US。校验和为8位，它由如下方式计算而得：

1. STX和ETX之间的所有字符异或，得到S0
2. S0二进制值中含1的个数若为偶数，则将S0的最高位置为1，得到S1，即校验和中始终含奇数个1。

[part]：用于制定指令操作的对象，例如LED、ADC等

<action>：用于制定行为，例如，对于LED，可以是On，Off；对于彩色LED，可以是SetColor

[parameters]：参数，以**US**隔开，例如对于彩色LED，可以是0,0,0，Green等。

这些设计是类的抽象，例如，我们可以把原型

Led::Set(enum ColorName cr)；

Led::Set(const Color& rgb);

设计为如下指令

<**STX**>Led6<**US**>Set<**US**>[it depends]<**ETX**>[chksum]<**EOT**>，直接通过颜色名字设置可以发送<**STX**>Led6<**US**>Set<**US**>Green<**ETX**>[chksum]<**EOT**>，通过RGB设置可以发送

<**STX**>Led6<**US**>Set<**US**>[it depends]<**ETX**>0,255,0<**EOT**>

在设计设备的指令时应当灵活应对，最佳的设计即等效设计，在从机中设计好某个设备的控制模块，然后将此模块按照以上的方式抽象成指令。例如：

<**STX**>ColorLed<**US**>Set<**US**>6, Green<**ETX**>[chksum]<**EOT**>，表示将第6个彩色Led设置为绿色。

1. 指令包

指令包中包括一组指令。

<SOH>[指令]<EOB>

1. 应答

一条应答由以下ASCII控制字符和可打印字符构成。

<**STX**><JSON><**ETX**>[chksum]<**EOT**>

校验和计算方式见“见上方”。应答的主体部分为一个JSON格式的字符串，UTF-8编码。

应答也可以是以下形式

<**ACK**>

这个应答表示指令已被接受，它可用于一些不需要返回值的情形。

## 强制指令

所谓强制指令，是指必须实现的指令；只有实现了必须实现指令的设备才能声称符合本规范，也只有符合本规范的设备才允许出厂。

### 设备ID请求

指令

<**STX**>RequestID<**ETX**>[chksum]<**EOT**>

应答的JSON必须包含”Name"和”ID”键，举例如下：

<**STX**>

{

“Name”: “V8 Interface/Switch”

“ID”: “QN9ST8U001”

}

<**ETX**>[chksum]<**EOT**>

### 请求设备特性

指令

<**STX**>RequestFeatures<**ETX**>[chksum]<**EOT**>

应答，没有具体要求，设备自行提供即可。

### 查询指令

指令

<**STX**>RequestStatus<**ETX**>[chksum]<**EOT**>

应答

若从设备处于Standby状态，则返回<ACK>，否则返回JSON字串，并在Status键值中描述状态。

### 异常应答

对于任何异常（无论是错误的指令还是从机本身的异常），都返回一个JSON字串，并在键值Status中描述异常状态。如：

{

“Status”: “Unsupported”

}

### 耗时指令处理

若某条指令需要消耗较长时间，则从机应遵循如下方式执行：

1. 指令错误，返回异常应答；
2. 返回忙碌状态，
3. 进入忙碌状态，此时对所有的指令返回异常应答。

## 设备特有指令

设备特有指令不能与3.2节中规定的指令重复，除此之外，只需要遵循指令和应答的规范即可。