# 全自动生化仪串口通信协议

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 描述 | 版本 | 修改人 |
| 2016-06-19 | 创建文档 | 1.0 |  |

## 概述

## 规则

## 协议内容

### 协议格式

协议头+消息内容+校验

协议头：

|  |  |  |
| --- | --- | --- |
| 名称 | 字节 | 解释 |
| 标志位 | 1 | 0xbc，一组消息的开始 |
| 地址 | 1 | 对应控制板的地址，0xff为广播 |
| 长度 | 1 | 消息内容的长度 |
| 序列号 | 1 | 表示消息发送的顺序，每次加一 |
| 消息号 | 1 | [7bit] 应答位，接收方是否需要应答 |
| [0-6bit] 消息号，详见**消息内容** |
| 消息内容 | - | 详见**消息内容** |
| 校验位 | 1 | 以上内容累加和 |

**约定：**所有大于一个字节的数值内容，均以**大端**形式传输。

### 消息内容

广播(0x00)

**功能：**上位机查询控制板是否已经工作正常。下位机收到该消息后根据自身地址进行延时回复应答消息。

**格式：**

**上位机发送：**

|  |  |  |
| --- | --- | --- |
| 名称 | 字节 | 内容 |
| 消息号 | 1 | 0x00 |
| 消息内容 | - | 空 |

自检(0x01)

**功能：**上位机发送该指令后，对应控制板进行自检操作，操作完成后返回结果。

**格式：**

**上位机发送：**

|  |  |  |
| --- | --- | --- |
| 名称 | 字节 | 内容 |
| 消息号 | 1 | 0x01 |
| 消息内容 | - | 空 |

**控制板回复：**

|  |  |  |
| --- | --- | --- |
| 名称 | 字节 | 内容 |
| 消息号 | 1 | 0x01 |
| 自检结果 | 1 | 0：成功  >0：失败，并对应为失败号 |

电机控制(0x02)

**功能：**上位机发送指令通知控制板控制电机进行相应操作，控制板操作完成后回复操作结果。

**格式：**

**上位机发送：**

|  |  |  |
| --- | --- | --- |
| 名称 | 字节 | 内容 |
| 消息号 | 1 | 0x02 |
| 序号 | 1 | 标识，用于回复控制结果 |
| 总步骤 | 1 | 0x00-0xff |
| 控制类型 | 1 | 该控制的类型 |
| 是否重复执行 | 1 | 0：不重复  其他：重复次数 |
| 步骤1 | - | 详见**步骤**格式 |
| 步骤2 | - | 详见**步骤**格式 |
| - | - | - |

**步骤：**

|  |  |  |
| --- | --- | --- |
| 名称 | 字节 | 内容 |
| 电机序号 | 1 | 由控制板定义 |
| 旋转方向 | 1 | 0：顺时针  1：逆时针 |
| 停止条件 | 1 | 0：旋转到指定步数  1：传感器触发 |
| 步数 | 2 | 0：永久旋转  其他：旋转步数 |

**控制板回复：**

|  |  |  |
| --- | --- | --- |
| 名称 | 字节 | 内容 |
| 消息号 | 1 | 0x02 |
| 序号 | 1 | 同上位机发送的序号 |
| 结果 | 1 | 0：成功  其他：失败号 |

状态查询(0x03)

**功能：**上位机查询控制板的状态，控制板收到后回复自身状态

**格式：**

**上位机发送：**

|  |  |  |
| --- | --- | --- |
| 名称 | 字节 | 内容 |
| 消息号 | 1 | 0x03 |
| 消息内容 | - | 空 |

**控制板回复：**

|  |  |  |
| --- | --- | --- |
| 名称 | 字节 | 内容 |
| 消息号 | 1 | 0x03 |
| 状态 | 1 | 0：空闲  1：忙碌 2：故障 |
| 故障码 | 1 | 0x00-0xff |

应答(0x04)

**功能：**当接收的消息需要确认时，需要回复该消息，用来确认通信成功。

**格式：**

**上位机发送：**

|  |  |  |
| --- | --- | --- |
| 名称 | 字节 | 内容 |
| 消息号 | 1 | 0x04 |
| 序列号 | 1 | 接收到的消息的序列号 |

## 附录

校验程序

|  |
| --- |
| unsigned char sumCheck(const unsigned char \*dat, unsigned char len)  {  unsigned char count = 0, i;  for(i = 0; i < len; i++)  {  count += dat[i];  }  return count;  } |