**Noah电源板CAN通讯协议**

**R0.0**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Versions | Revisions | Authors | Release Date |
| R00 | Initial Version | Kaka | 2017.09.30 |

**Table of Contents**

[1 通信机制 4](#_Toc484694469)

[2 Func ID 4](#_Toc484694470)

[3 SrcMac ID 4](#_Toc484694471)

[4 Source ID 4](#_Toc484694472)

[5 通用通信报文 5](#_Toc484694473)

[5.1 查询版本 5](#_Toc484694474)

[6 通信报文 6](#_Toc484694475)

[6.1 设置读取各模块开关状态 6](#_Toc484694476)

[6.2 读取故障状态 9](#_Toc484694477)

[6.3 读取电流信息 11](#_Toc484694478)

[6.4 获取RK板状态 13](#_Toc484694479)

[6.5 读取设置灯光亮度 13](#_Toc484694480)

# 通信机制

1. 使用CAN通信
2. 扩展ID，500kbps

# Func ID

FUNC\_ID 协议主功能码，4 位组合定义如下

|  |  |  |
| --- | --- | --- |
| FuncID | 功能 | 描述 |
| 0x00 | Reserve | 保留功能码 |
| 0x01 | 连续写端口 | 用于对单个或者多个资源节点的数据写入 |
| 0x02 | 连续读端口 | 用于读取单个或者多个资源节点的数据 |
| 0x03 | 事件触发传送 | 用于从站主动向主站传送数据 |
| 0x04 | Reserve | Reserve |
| 0x05 | Reserve | Reserve |
| 0x06 | 复位控制 | 控制节点复位 |
| … | Reserve | Reserve |
| 0xff | 异常响应 | 用于响应通信相关的异常 |

# SrcMac ID

Noah电源板的 SrcMacID 为0x50。

# Source ID

Noah电源板source ID分配：

节点内部资源的地址，共256个地址;

其中，00 – 1F区间为通用功能地址；

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 功能分类 | SourceID | 功能 | 描述 | 属性 |
| 通用功能地址 | 0X00 | Reserve | Reserve |  |
| 0x01 | 版本查询 | 查询模块的当前软件、硬件等版本信息 | R |
| 0x02 |  |  |  |
| 0x03 |  |  |  |
| … … |  |  |  |
| 0X10 | 升级准备 | 写入校验和包长度 | W |
| 0X11 | 升级传输 | 写入升级数据包 | W |
| 0X12 | 校验 | 读取升级包校验 | R |
| … … |  |  |  |
| …-0X1F |  |  |  |
| 模块功能地址 | 0x80 | 读取各模块开关状态 | 读取各个模块的开关状态 | RO |
|  | 0x80 | 设置各模块开关状态 | 设置各个模块的开关状态 | RO |
|  | 0x81 | 设置模块开关 | 设置各个模块的开关 | W |
|  | 0x82 | 读取故障状态 | 读取各个模块的电流或电压异常情况 | RO |
|  | 0x83 | 读取电池电压 | 读取电池电压或电量百分比 | R |
|  | 0x83 | 读取各电流电压温度数据 | 读取各个模块的电流电压温度数据 | RO |
|  | 0x84 | 读取设置当前灯带效果 | 读取和设置当前灯带的效果 | WR |
|  |  |  |  |  |

# 通用通信报文

## 查询版本

Source ID：0x01

发送：

数据长度：1

数据内容：

|  |
| --- |
| byte1 |
| 查询内容 |

查询内容：

0x01：软件版本

0X02：协议版本

0x03 :硬件版本

**应答：**

**数据长度：不定**

**数据内容：**

|  |  |
| --- | --- |
| byte1 | byte2-byte N |
| 查询内容 | 协议内容 |

查询内容：

同上

协议内容：

长度不定，软件版本，硬件版本，协议版本的数据长度不一定一样。例如：

软件版本：“NOAHC001M08B102 “长度为15 bytes，那么N为16，从byte2到byte18分别为(数据内容格式为字符)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| byte 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| N | O | A | H | C | 0 | 0 | 1 | M | 0 | 8 | B | 1 | 0 | 2 |

# 通信报文

设置各模块开关状态

Source ID：0x81

**数据长度**：6

**数据内容：**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **byte 1** | **byte2** | **byte3** | **byte4** | **byte5** | **byte6** | **byte7** |
| **预留** | **组号** | **选择模块** | | | | **操作** |

**组号：**

**0x01：组号1**

**0x02：组号2**

**选择模块号：（组号1）**

命令组号**：0x01**

**各模块开关状态（低16位）**：（0 –关闭；1-开启 ）

BIT0（0x0001）：一键开机键状态

BIT1（0x0002）：DLP开关状态

BIT2（0x0004）：X86开关状态

BIT3（0x0008）：PAD开关状态

BIT4（0x0010）：DC-DC-EN-5V开关状态

BIT5（0x0020）：DC-DC-EN-12V开关状态

BIT6（0x0040）：DC-DC-EN-24V开关状态

BIT7（0x0080）: 电机板开关状态

BIT8（0x0100）： 壁障板开关状态

BIT9（0x0200）: 灯带开关状态

BIT10（0x0400）: 预留5V开关状态

BIT11（0x0800）: 12V路由器开关状态

BIT12（0x1000）: 音箱开关状态

BIT13（0x2000）: NV 板开关状态

BIT14（0x4000）: 打印机开关状态

BIT15 ( 0x8000 ): 低音炮开关状态

**各模块开关状态（高16位）**：（0 –关闭；1-开启 ）

BIT0（0x0001）：预留12V开关状态

BIT1（0x0002）：预留 24V 开关状态

BIT2（0x0004）：电池直连 NV 开关状态

BIT3( 0x0008 ): AIUI 开关状态

BIT4( 0x0010 ): 5V 路由器开关状态

。。。

**命令组号**：0x02 (待扩展)

**操作：**

**0x00：关闭**

**0x01：打开**

**应答**

**数据长度：10**

**数据内容:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **byte1** | **byte2** | **byte3-byte6** | **byte7-byte10** | **byte11** |
| **预留** | **组号** | **选择模块** | **开关状态** | **操作** |

选择模块号：

同上述

组号：

同上述

开关状态：

（对应的bit 位：0 –关闭；1-开启 ）

BIT0（0x0001）：一键开机键状态

BIT1（0x0002）：DLP开关状态

BIT2（0x0004）：X86开关状态

BIT3（0x0008）：PAD开关状态

BIT4（0x0010）：DC-DC-EN-5V开关状态

BIT5（0x0020）：DC-DC-EN-12V开关状态

BIT6（0x0040）：DC-DC-EN-24V开关状态

BIT7（0x0080）: 电机板开关状态

BIT8（0x0100）： 壁障板开关状态

BIT9（0x0200）: 灯带开关状态

BIT10（0x0400）: 预留5V开关状态

BIT11（0x0800）: 12V路由器开关状态

BIT12（0x1000）: 音箱开关状态

BIT13（0x2000）: NV 板开关状态

BIT14（0x4000）: 打印机开关状态

BIT15 ( 0x8000 ): 低音炮开关状态

**各模块开关状态（高16位）**：（0 –关闭；1-开启 ）

BIT0（0x0001）：预留12V开关状态

BIT1（0x0002）：预留 24V 开关状态

BIT2（0x0004）：电池直连 NV 开关状态

BIT3( 0x0008 ): AIUI 开关状态

BIT4( 0x0010 ): 5V 路由器开关状态

。。。

读取各模块开关状态

Source ID：0x82

**数据长度**：2

**数据内容：**

|  |  |
| --- | --- |
| **byte 1** | **byte2** |
| **预留** | **组号** |

**组号：**

**0x01：组号1**

**0x02：组号2**

**选择模块号：（组号1）**

**应答**

**数据长度：6**

**数据内容:**

|  |  |  |
| --- | --- | --- |
| **byte1** | **byte2** | **byte3-byte6** |
| **预留** | **组号** | **开关状态** |

选择模块号：

同上述

组号：

同上述

开关状态：

（对应的bit 位：0 –关闭；1-开启 ）

BIT0（0x0001）：一键开机键状态

BIT1（0x0002）：DLP开关状态

BIT2（0x0004）：X86开关状态

BIT3（0x0008）：PAD开关状态

BIT4（0x0010）：DC-DC-EN-5V开关状态

BIT5（0x0020）：DC-DC-EN-12V开关状态

BIT6（0x0040）：DC-DC-EN-24V开关状态

BIT7（0x0080）: 电机板开关状态

BIT8（0x0100）： 壁障板开关状态

BIT9（0x0200）: 灯带开关状态

BIT10（0x0400）: 预留5V开关状态

BIT11（0x0800）: 12V路由器开关状态

BIT12（0x1000）: 音箱开关状态

BIT13（0x2000）: NV 板开关状态

BIT14（0x4000）: 打印机开关状态

BIT15 ( 0x8000 ): 低音炮开关状态

**各模块开关状态（高16位）**：（0 –关闭；1-开启 ）

BIT0（0x0001）：预留12V开关状态

BIT1（0x0002）：预留 24V 开关状态

BIT2（0x0004）：电池直连 NV 开关状态

BIT3( 0x0008 ): AIUI 开关状态

BIT4( 0x0010 ): 5V 路由器开关状态

。。。

读取系统状态

**发送：**

**Source ID：0x83**

**数据长度**：1

|  |
| --- |
| **byte1** |
| **读取命令** |

**读取命令：**

* 0x00: 读取系统状态
* 。。。

**应答：**

**数据长度:5**

**数据内容：**

|  |  |  |
| --- | --- | --- |
| **byte1** | **byte2** | **byte3** |
| **读取命令** | **系统状态** | |

**系统状态**：

BI0 – BIT3

- 关机待机 00

- 开机中 01

- 开机完成运行中 02

- 关机中 03（发送该信号后，X86和PAD进行自主关机）

- 故障 04

BIT4

* 充电状态
  + Bit4 = 0 不在充电中；Bit4 = 1 充电中

BIT5

* 电池电压
  + Bit5 = 0 电池电压正常；Bit5 = 1 电池电压低

BIT6

* 电流上报开启状态
* Bit6 = 0 未开启自动上报；Bit6 = 1 已开启自动上报

BIT7

* 充电器插入状态
* Bit7=0 充电器未插入；Bit7=1 充电器已插入

BIT8 ~ BIT15 (保留为0，待扩展)

读取故障状态

**发送：**

**Source ID：0x84**

**数据长度**：0

|  |
| --- |
| **byte1** |
| **命令组号** |

**命令组号：**

* 0x01: 读取第一组（每组32bit）
* 。。。

**应答：**

**数据长度:5**

**数据内容：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **byte1** | **byte2** | **byte3** | **byte4** | **byte5** |
| **命令组号** | **故障**（byte2） | **故障**（byte3） | **故障**（byte4） | **故障**（byte5） |

命令组号：0x01

**故障位**（byte1）：（0正常 1异常）

|  |  |
| --- | --- |
| 环境温度检测 | **故障位** -Bit0 |
| 电池总电流检测 | **故障位** -Bit1 |
| 电机驱动器24V动力电流检测 | **故障位** -Bit2 |
| 底盘驱动控制器5V电流检测 | **故障位** -Bit3 |
| 导航Camer 5V电流检测 | **故障位** -Bit4 |
| 避障系统5V电流检测 | **故障位** -Bit5 |
| DLP 5V电流检测 | **故障位** -Bit6 |
| 灯带系统 5V电流检测 | **故障位** -Bit7 |
| 胸口PAD 12V电流检测 | **故障位** -Bit8 |
| X86工控板 12V电流检测 | **故障位** –Bit9 |
| 2.1声道音频功放 12V电流检测 | **故障位** –Bit10 |
| 导航Camer 12V电流检测 | **故障位** –Bit11 |
| 打印机 24V电流检测 | **故障位** –Bit12 |
| 预留5V电流检测 | **故障位** –Bit13 |
| 预留12V电流检测 | **故障位** –Bit14 |
| 预留24V电流检测 | **故障位** –Bit15 |

**故障位**（高2Bytes）（0正常 1异常）

|  |  |
| --- | --- |
| 电池直连NV电流检测 | **故障位** -Bit0 |
| 12V 连NV 电流检测 | **故障位** -Bit1 |
| 路由器电流检测 | **故障位** -Bit2 |
| 低音炮电流检测 | **故障位** -Bit3 |
| 红外板电流检测 | **故障位** -Bit4 |
| 充电器电流检测 | **故障位** -Bit5 |
| 充电电流检测 | **故障位** -Bit6 |
| 24V电压检测 | **故障位** -Bit7 |
| 12V电压检测 | **故障位** -Bit8 |
| 5V电压检测 | **故障位** –Bit9 |
| 5V总电流检测 | **故障位** –Bit10 |
| 12V总电流检测 | **故障位** –Bit11 |
| 24V总电流检测 | **故障位** –Bit12 |
| 5V 稳压温度检测 | **故障位** –Bit13 |
| 12V 稳压温度检测 | **故障位** –Bit14 |
| 24V稳压温度检测 | **故障位** –Bit15 |

**故障位2**（1Bytes）（0正常 1异常）

|  |  |
| --- | --- |
| AIUI电流检测 | **故障位** -Bit0 |
| 5V路由器电流检测 | **故障位** -Bit1 |
|  |  |
|  |  |
|  |  |

命令组号：0x02(待扩展)

读取电池电压

发送：

Source ID：0x85

数据长度：1

数据内容：

|  |
| --- |
| byte1 |
| 预留 |

**应答：**

**数据长度：0**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| byte1 | byte2 | byte3 | byte4 | byte5 |
| 预留 | 电池电压 | | 电量百分比 | |

**电压(2Byte)：**十进制数（最大65536），单位为10mV。如2556 表示25.56V（若为0，表示硬件不支持）

**电量(2Byte)：**十进制数百分比（最大100），单位为%。如56 表示56%（若为0，表示硬件故障）

读取ADC信息

发送：

Source ID：0x86

数据长度：1

数据内容：

|  |
| --- |
| byte1 |
| 电流上报频率 |

**频率：**

* 0x00: 单次读取
* 0x01: 1Hz
* 0x02: 2Hz
* 0x03: 5HZ
* 0x04: 10Hz
* 0x05: 50Hz
* 0x06:100Hz
* 0x07: 0.5Hz
* 0x08: 0.2Hz
* 0x09: 0.1Hz
* 0xFF:停止上报
* 。。。

**应答：**

**数据长度：0**

**上报：**

**数据长度：66**

**数据内容：**

|  |  |
| --- | --- |
| **2 bytes** | **5V右手电机模组电流** |
| **2 bytes** | **24V高拍仪电机驱动电流** |
| **2 bytes** | **24V头部电机模组电流** |
| **2 bytes** | **24V camera电机电流** |
| **2 bytes** | **5V加密键盘电流** |
| **2 bytes** | **5V扫码器电流** |
| **2 bytes** | **5V左手电机模组电流** |
| **2 bytes** | **5V读卡器电流** |
| **2 bytes** | **5V头部电流** |
| **2 bytes** | **24V右手电机模组电流** |
| **2 bytes** | **12V头部RK电流** |
| **2 bytes** | **12V胸部RK电流** |
| **2 bytes** | **24V左手电机模组电流** |
| **2 bytes** | **12V功放电流** |
| **2 bytes** | **5V维修版电流** |
| **2 bytes** | **5V触摸板电流** |
| **2 bytes** | **保留** |
| **2 bytes** | **12V路由器电流** |
| **2 bytes** | **Vbat总电流** |
| **2 bytes** | **12V读卡器电流** |
| **2 bytes** | **12V交换机电流** |
| **2 bytes** | **12V电源温度** |
| **2 bytes** | **5V电源温度** |
| **2 bytes** | **环境温度** |
| **2 bytes** | **5V头部触摸板电流** |
| **2 bytes** | **12V总电流** |
| **2 bytes** | **5V总电流** |
| **2 bytes** | **5V头部灯板电流** |
| **2 bytes** | **12V电压** |
| **2 bytes** | **5V电压** |
| **2 bytes** | **电池电压** |
| **2 bytes** | **5V预留电压** |
| **2 bytes** | **5V 高拍仪LED电流** |
| **2 bytes** | **5V camera电流** |
| **2 bytes** | **5V hd\_camera 电流** |

设置红外补光灯亮度

Source ID：0x87

发送：

数据长度：2

数据内容：

|  |  |
| --- | --- |
| byte1 | byte2 |
| 预留 | 红外补光灯亮度 |

亮度值：

0—100：亮度值

**应答：**

**数据长度：3**

**数据内容：**

|  |  |
| --- | --- |
| byte1 | byte2 |
| 预留 | 红外补光灯亮度 |

亮度值：

0—100：亮度值

0xff：设置失败

读取红外补光灯亮度

Source ID：0x88

发送：

数据长度：1

数据内容：

|  |
| --- |
| byte1 |
| 预留 |

应答：

**数据长度：3**

**数据内容：**

|  |  |
| --- | --- |
| byte1 | byte2 |
| 预留 | 红外补光灯亮度 |

亮度值：

0—100：亮度值

0xff：获取失败

设置灯带效果

Source ID：0x89

发送：

数据长度：6

数据内容：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **byte1** | **byte2** | **byte3** | **byte4** | **byte5** | **byte6** |
| **预留** | 灯光模式 | 灯光颜色R | 灯光颜色G | 灯光颜色B | 灯光频率 |

**预留：0**

**灯光模式**：选择需要显示的模式状态：

- 不设置模式时默认配置 00

- 正常状态 01

- 故障 02

- 低电量 03

- 正在充电 04

- 左转 05

- 右转 06

- 通讯中断 07

灯光颜色

- 灯光颜色R 灯光颜色的R值

- 灯光颜色G 灯光颜色的G值

- 灯光颜色B 灯光颜色的B值

灯光闪烁间隔时间

* 单位：20ms

应答：

**数据长度：6**

**数据内容：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **byte1** | **byte2** | **byte3** | **byte4** | **byte5** | **byte6** |
| **预留** | 灯光模式 | 灯光颜色R | 灯光颜色G | 灯光颜色B | 灯光频率 |