# Nanopi2与stm32 通信协议

注：本说明只是初步定义，待完善。

通信协议格式

1. 帧的格式图



2.帧内容的具体含义和位数

**起始标记**：使用0XDADC作为一帧数据的起始标记，占用2个字节 。

**Type ：**帧的功能含义，占用1个字节。

|  |  |
| --- | --- |
| value | 功能 |
| 1 | 传输stm32检测的电压值 |
| 2 | 传输设备的设备型号、硬件版本、软件版本 |
| 3 | 传输设备认证的公钥 |
| 4 | 传输程序更新协商及程序包 |
| 5 | 主心跳 |
| 6 | 次心跳 |
| 7 | 升级处理 |
| 8 | 获取湿度和温度 |

**Subtype：**扩展Type的功能根据实际需求定义，占用1个字节。默认：0x01

**数据长度：**记录所发送数据的字节总数，占用4个字节（int）。

**预留 ：**用于以后的扩展 3个字节。

**校验位：**采用crc8方式校验 CRC-8 x8+x5+x4+1 0x31（0x131）

**数据：**用户发送的有效数据，具体的格式如下

1. 传输stm32检测的电压值（单位：mv）

定义各个电压值占用2个字节。

3.3V 5V 12V 48V



1. 设备版本号

# 硬件版本号： 占用16个字节（示例模板：GAC-300 v2.0）

# Stm32软件包版本号：占用16个字节（示例模板：GAC-300 v1.0）

# 以上示例模板根据实际需要进行编号。

硬件版本号 stm32软件版本号

# 

1. 传输设备认证的公钥

与“ 4.传输stm32更新程序包“的方式相同 。

4．传输stm32更新程序包

传输程序分为4个步骤 ：

第一步：请求发送程序数据 (nanopi -> stm32)



# 版本号：将要传输的更新程序的版本号 占用16个字节（示例模板：GAC-300 v1.0）。

# Block\_amount: 传输设备的总块数 占用4个字节。

# data\_size :将要传输的更新程序的总字节数 占用4个字节 。（stm32是否有足够的空间）

# 第二步：stm32接收预备返回状态值 （stm32->nanopi）

# 

# ack: stm32返回状态值 （stm32判断将要更新的程序比当前程序版本高并且有足够的空间保存将要传输的程序时 ack=1 否则返回ack=0）

# 第三步：发送更新程序的分块数据 (nanopi->stm32)

# 

# block\_index : 发送更新的程序的分块序列号 占用4个字节 。

# len :所发块的data的大小占用4个字节 。

# data：发送的对应的分块序列号的块数据 。

# 第四步：返回接收状态值 (stm32->nanopi)

# 

# ack：ack=1 接收成功 ，ack=0 接收失败 。

# 发送一个较大的程序时，在第一步和第二步验证成功后，数据发送的过程重复第三步和第四步，直到成程序发送完毕 。

5．主心跳

# 主心跳的没有数据部分，用帧头的type来唯一标志。

7．升级处理

# Stm32\_client升级通知，用帧头的type来唯一标志。

# Stm32 返回ack 。

1. 获取温度和湿度 单位（温度：℃ 湿度：%rh）

温度 湿度 预留 预留

