# 脉诊模块数据通讯协议V0.7

## 简介

本数据通讯协议用于脉诊模块与音箱或者PC通讯之用。通讯的物理通道基于USB或者蓝牙（扩展）。协议第一部分阐明了基本数据帧格式与用法。第二部分介绍了其他数据帧的格式与用途。

USB线缆/蓝牙（扩展）

USB虚拟串口/蓝牙模块

USB虚拟串口/蓝牙模块

FW Protocol

SW Protocol



## 基本通讯帧格式与应用规则

### 基本通讯帧（Frame）格式

本部分内容主要介绍了通讯过程中，数据与命令通过物理层传输时使用的传输格式。采用USB转串口的方式，或者蓝牙无线传输的方式（数字接口蓝牙模块），如下描述的每一帧代表USB数据帧或者蓝牙数据包的数据负载。

一系列有序排列的数据位共同组成数据帧，数据位从左至右依次编号，从K-1（左侧最高有效位）至0（右侧最低有效位），最高位在左），位域长度为K。每一帧(Frame)的数据长度小于等于20字节。帧格式包含5个部分，分别是：标志包头域（Frame Header），长度域（Length），信息标识域（Message ID），负载域（Payload）和CRC校验。示意图如下，其中M（Mandatory）代表必备，O（Optional）代表可选。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bytes：  4 | 1 | 1 | <=12 | 1 |
| Frame Header  (O) | Length  (O) | Message ID  (O) | Payload  (M) | CRC8  (O) |

#### Frame Header域

用途：用于数据传输时标记数据包身份，防止数据混淆。

1.脉诊设备向上位机发送脉诊数据包头DataHeader4（4 byte）：0xAA，0x55，0xAA，0x55。

2.脉诊设备向上位机发送ACK，NACK或者Ready的RespHeader4（4byte）：0xAA，0xAA，0x55，0x55。

3.上位机向脉诊设备发送命令，加入标识包头CMDHeader4（4 byte）：0x55，0x55，0xAA，0xAA。

#### Length域

Length域长度为1个byte（8 bites），其内容是一个8位的无符号整数，其含义是该帧数据Length域之后的数据长度，Length域的取值范围是1~13。

#### Message ID域

Message ID域的长度是1个byte，其内容是用户应用信息标识，即通讯帧具体含义识别符。详细的定义见后续章节。

#### Payload域

Payload域是一个可选的域段，既可以包含一个或多个byte，也可以是0个byte。具体情况需要根据不同Message ID的上下文对话情况确定，详细定义见后续章节。

#### CRC8域

CRC8域是一个可选的域段，占用0个或者1个byte。

### 协议应用规则

通讯过程中主要存在以下几种传输形式：1无应答传输。2肯定应答ACK传输。3否定应答NACK传输。

#### 2.1无应答传输

该帧发出后无需对方应答，即该帧的发出者（MCU或者音箱PC）默认数据发出是成功的，无需确认。信息流序框图如下：

DATA or CONTROL Frame

Originator

Recipient

Originator

Recipient

#### 2.2肯定应答ACK传输

如果接收方正确的收到了数据，它将发出ACK应答包。信息流序框图如下：

ACK Frame with or without payload

DATA or CONTROL Frame

Originator

Recipient

#### 2.3否定应答NACK传输

如果接收到的帧不符合标识过滤规则，例如：格式有错，校验未通过（可选）或者无法解析，则接收方发出NACK帧，该帧payload域可以包含error code错误代码。信息流序框图如下：

NACK Frame with error code

“BAD” DATA or CONTROL

Originator

Recipient

## 具体帧格式编码定义

按用途的可以将帧分为如下几类：

1. 信息获取命令。发起者：PC或音箱。接收者：MCU
2. 系统设置命令。发起者：PC或音箱。接收者：MCU
3. 数据Ready包。发起者：MCU。接收者：PC或音箱。
4. 数据传输包。发起者：MCU。接收者：PC或音箱。

### 信息获取命令

用途：PC或音箱通过发送命令，获取MCU端传感器的相关参数和信息。

实现方式：PC或音箱发命令给MCU，MCU根据情况做出应答。帧定义见下表：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Commands and Responds | Header Type | Length | Message  ID | Payload  Length(未使用) | Payload | Originator | Chinese  Explanation |
| MCU\_Get\_PID | CMDHeader4 | 1 | 0x22 | 0 |  | PC | 获取传感器ID |
| MCU\_Get\_PID\_Response\_ACK | RespHeader4 | 6 | 0x24 | 6 | PID | MCU | 肯定应答 |
| MCU\_Get\_PID\_Response\_NACK | RespHeader4 | 2 | 0x28 | 1 | Err code | 否定应答 |
| MCU\_Get\_ID | CMDHeader4 | 1 | 0x32 | 0 |  | PC | 获取传感器ID |
| MCU\_Get\_ID\_Response\_ACK | RespHeader4 | 13 | 0x34 | 12 | UID | MCU | 肯定应答 |
| MCU\_Get\_ID\_Response\_NACK | RespHeader4 | 2 | 0x38 | 1 | Err code | 否定应答 |
| MCU\_Get\_Version | CMDHeader4 | 1 | 0x42 | 0 |  | PC | 获取版本号 |
| MCU\_Get\_Version\_Response\_ACK | RespHeader4 | 3 | 0x44 | 2 | Version | MCU | 肯定应答 |
| MCU\_Get\_Version\_Response\_NACK | RespHeader4 | 2 | 0x48 | 1 | Err code | 否定应答 |

其中 MCU PID是HardwareVersion+4字节SN号

其中MCU UID是芯片内部的device ID，是每个芯片唯一的标识，长度为12个字节。

Version字段为2个字节，第一个字节代表硬件版本，第二个字节代表固件版本。

### 系统设置命令

用途：PC或音箱通过发送命令，设置MCU 的工作状态和参数。

实现方式：PC或音箱发命令给MCU，MCU根据情况做出应答。帧定义见下表：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Commands and Responds | Header Type | Length | Message  ID | Payload  Length | Payload | Originator | Chinese  Explanation |
| MCU\_Set\_Led | CMDHeader4 | 2 | 0x52 | 1 | Led State | PC | 设置LED状态 |
| MCU\_Set\_Led\_Response\_ACK | RespHeader4 | 1 | 0x54 | 0 |  | MCU | 肯定应答 |
| MCU\_Set\_Led\_Response\_NACK | RespHeader4 | 2 | 0x58 | 1 | Err code | 否定应答 |
| MCU\_Start\_Acquisition | CMDHeader4 | 1 | 0x62 | 0 |  | PC | 开始数据采集 |
| MCU\_Start\_Acquisition\_Response\_ACK | RespHeader4 | 1 | 0x64 | 0 |  | MCU | 肯定应答 |
| MCU\_Start\_Acquisition\_Response\_NACK | RespHeader4 | 2 | 0x68 | 1 | Err code | 否定应答 |
| MCU\_Stop\_Acquisition | CMDHeader4 | 1 | 0x72 | 0 |  | PC | 停止数据采集 |
| MCU\_Stop\_Acquisition\_Response\_ACK | RespHeader4 | 1 | 0x74 | 0 |  | MCU | 肯定应答 |
| MCU\_Stop\_Acquisition\_Response\_NACK | RespHeader4 | 2 | 0x78 | 1 | Err code | 否定应答 |
| MCU\_Enter\_DFU\_Mode | CMDHeader4 | 1 | 0x82 | 0 |  | PC | 令MCU进入DFU模式 |
| MCU\_Enter\_DFU\_Mode\_Response\_ACK | RespHeader4 | 1 | 0x84 | 0 |  | MCU | 肯定应答 |
| MCU\_Enter\_DFU\_Mode\_Response\_NACK | RespHeader4 | 2 | 0x88 | 1 | Err code | 否定应答 |
| MCU\_Set\_Motor | CMDHeader4 | 3 | 0x92 | 2 | Motor Param | PC | 设置步进电机参数 |
| MCU\_Set\_Motor\_Response\_ACK | RespHeader4 | 1 | 0x94 | 0 |  | MCU | 肯定应答 |
| MCU\_Set\_Motor\_Response\_NACK | RespHeader4 | 2 | 0x98 | 1 | Err code | 否定应答 |
| MCU\_Set\_Serial\_Number | RespHeader4 | 5 | 0xC2 | 4 | SN | PC | 设置SerialNum |
| MCU\_Set\_SN\_Response\_ACK | RespHeader4 | 1 | 0xC4 | 0 |  | MCU | 肯定应答 |
| MCU\_Set\_SN\_Response\_NACK | RespHeader4 | 2 | 0xC8 | 1 | Err code | 否定应答 |

其中Led State是1个字节，每一位的位置对应指示灯编号，每一位的值对应指示灯的状态，0代表关，1代表开，因此最多可以对应8个Led指示灯,当前硬件最多包含3个指示灯，bit 0对应LED1，bit 1位对应LED2，bit 2位对应LED3。Motor Step是2个字节，其数值对应步进电机的选装方向和步数，具体含义待定义。上图中紫色部分涉及的两项功能：MCU\_Enter\_DFU\_Mode和MCU\_Set\_Motor作为扩展接口，留待后续进行功能实现。

### 数据READY包

用途：用于按下脉诊按键之后的消息，通知上位机开始采集数据。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Commands and Responds | Header Type | Length | Message  ID | Payload  Length | Payload | Originator | Chinese  Explanation |
| MCU\_Send\_Ready | RespHeader4 | 1 | 0xA2 | 0 |  | MCU | 数据Ready通知 |

通常状况下，上位机接收到Ready通知后开始脉诊数据采集。

### 数据传输包

4.1电通传感器数据格式：

用途：用于MCU向音箱或者PC传输脉搏测量数据，数据格式为16位无符号数。

实现方式：MCU连续发出数据帧，无需对方应答。帧定义见下表：

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Header | Length | Message  ID | DY1L | DY1H | DY2L | DY2H | DY3L | DY3H | JY1L | JY1H | JY2L | JY2H | JY3L | JY3H |
| DataHeader4  (0xAA，0x55，0xAA，0x55) | 无 | 无 | 动压1低字节 | 动压1高字节 | 动压2低字节 | 动压2高字节 | 动压3低字节 | 动压3高字节 | 静压1低字节 | 静压1高字节 | 静压2低字节 | 静压2高字节 | 静压3低字节 | 静压3高字节 |

其中H1，H2，H3，H4是数据包的特异性标志头，为提高数据传输效率，数据包不含Length和Message ID域，长度固定。

4.2华普传感器数据格式：

用途：用于MCU向音箱或者PC传输脉搏测量数据，数据格式为24位无符号数。

实现方式：MCU连续发出数据帧，无需对方应答。帧定义见下表：

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Header | Length | Message  ID | YL1H | YL1M | YL1L |  | YL2H | YL2M | YL2L |  | YL3H | YL3M | YL3L |  |
| DataHeader4  (0xAA，0x55，0xAA，0x55) | 无 | 无 | 压力1高字节 | 压力1中字节 | 压力1低字节 | 0x00 | 压力2高字节 | 压力2中字节 | 压力2低字节 | 0x00 | 压力3高字节 | 压力3中字节 | 压力3低字节 | 0x00 |

其中H1，H2，H3，H4是数据包的特异性标志头，为提高数据传输效率，数据包不含Length和Message ID域，长度固定。

### Error Code 定义

|  |  |
| --- | --- |
| **“Error code” field** | **Error** |
| 0x01 | Unsupported command |
| 0x02 | Out-of-range value |
| 0x03 | Not executable command |
| 0x04 | Wrong syntax |
| 0x05 | MCU board not connected |
| 0x06 | Command data corrupt |
| 0x07 – 0xFF | RFU（备用） |

### 应用举例

形如下图，待前述定义确定后补充完整，如需CRC8校验，则在每帧数据的帧尾加入CRC8数据段。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MCU\_Get\_ID | | | | |
| Header | Length | Message ID | Frame Type:CONTROL, Ack: required, I/O: Input, Version: USB, Qos: Normal | |
| 0x55,0x55,0xAA,0xAA | 0x01 | 0x32 |
| MCU\_Get\_ID\_Response (ACK case) | | | | |
| Header | Length | Message ID | MCU UID | Frame Type:ACK with Payload, Ack: not required, I/O: Output, Version: USB, Qos: Normal |
| 0xAA,0xAA,0x55,0x55 | 0x0D | 0x34 | 12 byte UID Data |
| MCU\_Get\_ID\_Response (NACK case) | | | | |
| Header | Length | Message ID | Error Code | Frame Type:NACK with Error code, Ack: not required, I/O: Output, Version: USB, Qos: Normal |
| 0xAA,0xAA,0x55,0x55 | 0x02 | 0x38 | 1 byte Error Code Data |