|  |  |  |  |
| --- | --- | --- | --- |
| **版本**  **Version** | **日期**  **Date** | **说明**  **Reversion history** | **作者**  **Author** |
|  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | **Viatom产品设备通信协议 Viatom products communication protocol** | | |
| 深圳源动创新科技有限公司  Shenzhen Viatom Technology Co., Ltd. | | 文档编码 Document Number | | 版本 Revision |
|  | |  |
| 编写 Author | 审核 Review | | 批准 Approval | |
|  |  | |  | |
| © 深圳源动创新科技有限公司 版权声明  本文件仅供深圳源动创新科技内部使用。文件中的任何部分未经允许，不得在公司之外传播、引用或者复制。  ViATOM TECHNOLOGY (UNPUBLISHED) ALL RIGHTS RESERVED  This material is the proprietary information of ViATOM Technology. And any use, reproduction, or distribution of the material contained herein, unless specifically authorized by ViATOM Technology, is expressly forbidden and would violate the rights and interests of ViATOM Technology. | | | | |

目录Contents

[1 概述OverView 4](#_Toc27731)

[2 协议Protocol 4](#_Toc17257)

[2.1 Head 4](#_Toc529)

[2.2 CMD 4](#_Toc2765)

[2.3 ~CMD 4](#_Toc12589)

[2.4 Pkg.Type 5](#_Toc4162)

[2.4.1 Table Pkg.Type 5](#_Toc153)

[2.5 Pkg.No. 5](#_Toc14912)

[2.6 Length 5](#_Toc4850)

[2.7 Data 6](#_Toc18135)

[2.8 CRC 6](#_Toc17513)

[2.8.1 CRC8 function 6](#_Toc18021)

[3 Command 6](#_Toc15622)

[3.1 Universal command(0xE0-0xFF) 7](#_Toc23891)

[3.1.1 0xE0:回显Echo 7](#_Toc16913)

[3.1.2 0xE1:获取设备信息Get device information 7](#_Toc3163)

[3.1.3 0xE2:复位Reset 8](#_Toc10140)

[3.1.4 0xE3:恢复出厂设置Restore Factory settings(sihua) 8](#_Toc31849)

[3.1.5 0xE4:获取电池状态Get battery status 8](#_Toc2949)

[3.1.6 0xE5:开始固件升级Start firmware upgrade 8](#_Toc29204)

[3.1.7 0xE6:固件升级数据Firmware data upgrade 9](#_Toc3438)

[3.1.8 0xE7:固件升级结束End of firmware upgrade 9](#_Toc28220)

[3.1.9 0xE8:开始语言包升级Start language pack upgrade 9](#_Toc4979)

[3.1.10 0xE8:语言包升级Start language pack upgrade 10](#_Toc27737)

[3.1.11 0xE9:语言包升级结束Language pack upgrade is over 10](#_Toc27057)

[3.1.12 0xEA:烧录出厂信息Burn factory information 10](#_Toc29820)

[3.1.13 0xEB:加密Flash Encrypted Flash 11](#_Toc15060)

[3.1.14 0xEC:设置时间Time setting 11](#_Toc9743)

[3.1.15 0xED:获取主机温度Get host temperature 11](#_Toc4680)

[3.1.16 0xEE:恢复生产出厂状态Restore factory state 12](#_Toc18898)

[3.1.17 0xF1:获取文件列表Get file list 12](#_Toc23746)

[3.1.18 0xF2:读文件开始Start reading file 12](#_Toc11448)

[3.1.19 0xF3:读文件数据Read file data 12](#_Toc16952)

[3.1.20 0xF4:读文件结束End of reading file 13](#_Toc21119)

[3.1.21 0xF5:写文件开始Start writing file 13](#_Toc19593)

[3.1.22 0xF6:写文件内容Write file content 13](#_Toc17981)

[3.1.23 0xF7:写文件结束End of writing file 14](#_Toc10332)

[3.1.24 0xF8:删除文件Delete files 14](#_Toc16448)

[3.1.25 0xF9:获取用户列表Get user list 14](#_Toc31232)

[3.1.26 0xFA:进入DFU升级模式Enter DFU upgrade mode 14](#_Toc21482)

[3.2 Private command(0x00-0xDF) 15](#_Toc20058)

[3.2.1 血压计Blood pressure monitor(AirBP Plus) 15](#_Toc11636)

# **概述Overview**

此通信协议支持蓝牙透传、RS232串口、USB HID等，对应蓝牙透传UUID、USB设备的PID VID等设备特有参数在私有协议中说明。

This communication protocol supports Bluetooth transparent transmission, RS232 serial port, USB HID and so on. Bluetooth transparent transmission UUID, PID VID of USB device and other device-specific parameters are described in private protocol.

**Notes：**

1. 协议传输字节序都采用小端模式
2. The sequence of bytes in this protocol is little-endian.
3. 当为USB HID通信时，每帧64Byte数据格式为：
4. When it is USB HID communication, the data format of each frame of 64Byte is:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Length** | **有效内容**  **Effective content** | **无效内容**  **Invalid content** |
| Size | 1 | length | 63- length |
| Description | 有效长度(小于64)  Effective length (less than 64) | 协议有效数据  Protocol valid data | 无效数据,0x00  Invalid data |

# **协议Protocol**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Head** | **CMD** | **~CMD** | **Pkg.Type** | **Pkg.No.** | **Lenght** | **Data** | **CRC** |
| Size | 1 | 1 | 1 | 1 | 1 | 2 | Length | 1 |
| Description | 0xA5 | 指令  Command | 指令取反  Inverted command | 包类型  (**[Pkg.Type表](#_Pkg.Type表)**)  Package type  (Table Pkg.Type) | 流水包号  Serial number  (0x00-0xFE) | 数据长度  Data length | 数据  Data  (object) | CRC8 |

## **Head**

固定值Fixed value 0xA5

## **CMD**

指令, 详见[Command](#_Command)章节

Command, see the chapter [Command](#_Command) for details.

## **~CMD**

指令值取反

Inverted command value.

## **Pkg.Type**

通信包类型

Communication packet type.

### **Table Pkg.Type**

|  |  |
| --- | --- |
| **Pkg.Type[[1]](#footnote-0)** | **Description** |
| **正常数据包(0x00-0x0F)**  **Normal packet (0x00-0x0F)** | |
| 0x00 | 指令请求  Command request |
| 0x01 | 正常应答  Normal response |
| … |  |
| **私有异常(0xA0-0xDF)**  **Private exception (0xA0-0xDF)** | |
| … | 见[Private command](#_Private_command(0x00-0xDF))章节  See chapter [Private command](#_Private_command(0x00-0xDF)) |
| … |  |
| **通用异常(0xE0-0xFF)**  **General exception (0xE0-0xFF)** | |
| 0xE0 | 找不到文件  File not found |
| 0xE1 | 读文件失败  Failed to read file |
| 0xE2 | 写文件失败  Failed to write file |
| 0xE3 | 固件升级失败  Firmware upgrade failed |
| 0xE4 | 语言包升级失败  Language pack upgrade failed |
| … |  |
| 0xFB | 设备资源被占用/设备忙  Device resources are occupied / Device is busy |
| 0xFC | 指令格式错误  Command format error |
| 0xFD | 不支持指令  Unsupported command |
| 0xFE | … |
| 0xFF | 通用错误  General error |

## **Pkg.No.**

数据包流水包号，主动发送时递增, 0xFF预留

Data packet serial number. It is incremented when sending actively, and 0xFF is reserved.

## **Length**

数据长度，当数据长度为0时无Data字段

Data length. No data field when the data length is 0.

## **Data**

数据内容，长度为Length字段表示

Data content. Its length is indicated by the Length field.

## **CRC**

Head~Data段的CRC8校验，CRC8计算函数如下

CRC8 check of Head~Data segment. CRC8 calculation function is as follows.

### **CRC8函数CRC8 function**

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// 函数名称Function name: crc8\_compute

// 函数说明Function description: crc8计算crc8 calculation

// 输入变量: pdata:数据地址 data\_size:数据长度 crc\_in:上一个crc值,初值赋0

// Input varible: pdata:data address data\_size:data length crc\_in:last crc value, and initial value is 0.

// 输出变量: crc值Output variable: crc value

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

unsigned char crc8\_compute(unsigned char \*pdata, unsigned data\_size, unsigned char crc\_in)

{

uint8\_t cnt;

uint8\_t crc\_poly = 0x07;

uint8\_t data\_tmp = 0;

while (data\_size--){

data\_tmp = \*(pdata++);

crc\_in ^= (data\_tmp << 0);

for(cnt = 0; cnt < 8; cnt++){

if(crc\_in & 0x80){

crc\_in = (crc\_in << 1) ^ crc\_poly;

}else{

crc\_in = crc\_in << 1;

}

}

}

return crc\_in;

}

# Command

本章返回只做正常应答（**Pkg.Type = 0x01**）和带数据异常(带**Pkg.Type**描述)说明，，其他异常返回通用如下表，其中私有异常可在各自章节查阅：

The returns in this chapter only normal responses(**Pkg.Type = 0x01**) and data exceptions(with **Pkg.Type** description). Other exceptions are returned as shown in the following table. Private exceptions reference to respective chapters.

|  |  |  |
| --- | --- | --- |
| **Receive** | | **Description** |
| **Pkg.Type** | 0xXX | Table [Pkg.Type](#_Pkg.Type表) |
| **Length** | 0x0000 |  |
| **Data** | N/A |  |

**Send：往设备端发送数据 Pkg.Type = 0x00**

**Send：Send data to the device Pkg.Type = 0x00**

**Receive: Receive data from device Pkg.Type = 0x01**

## Universal command(0xE0-0xFF)

### 0xE1:获取设备信息Get device information

|  |  |  |
| --- | --- | --- |
| **Send** | | **Description** |
| **Length** | 0x0000 |  |
| **Data** | N/A |  |
| **Receive** | | **Description** |
| **Length** | Sizeof(DeviceInfo) | DeviceInfo对象长度  DeviceInfo object length |
| **Data** | DeviceInfo | DeviceInfo成员如下  DeviceInfo members are as follows |

DeviceInfo {

Unsigned char hw\_verson; //硬件版本 Hardware version e.g. ‘A’ : A版 version

Unsigned int fw\_version; //固件版本 Firmware version e.g. 0x00010200 : V1.2.0

Unsigned int bl\_version; //引导版本 Bootloader version e.g. 0x00010200 : V1.2.0

Unsigned char branch\_code [8]; //Branch编码 Coding e.g. “40020000” : Ezcardio Plus

Unsigned char fs\_version; //文件系统版本 File system version e.g. 0x01:V1

Unsigned char reserved0[2]; //预留 Reserved

Unsigned short device\_type; //设备类型 device type e.g. 0x8611: 血压计

Unsigned short protocol\_version; //协议版本 Protocol version e.g.0x0100:V1.0

Unsigned char cur\_time[7]; //time e.g.0xE1070301090000:2017-03-01 09:00:00

Unsigned short protocol\_data\_max\_len; //通信协议数据段最大长度，不包括固定字节

Unsigned short protocol\_data\_max\_len; // Data segment maximum length of communication protocol, which exclude fixed bytes.

unsigned char reserved1[4]; //预留 Reserved

SN{

Unsigned char len; //SN 长度(小于18) e.g. 10

Unsigned char len; //SN length(less than 18) e.g. 10

Unsigned char serial\_num[18]; //SN号 e.g. “2017022211” : 2017022211

Unsigned char serial\_num[18]; //SN number e.g. “2017022211” : 2017022211

}

unsigned char reserved2[4]; //预留 Reserved

}

### 0xE4:获取电池状态Get battery status

|  |  |  |
| --- | --- | --- |
| **Send** | | **Description** |
| **Length** | 0x0000 |  |
| **Data** | N/A |  |
| **Receive** | | **Description** |
| **Length** | Sizeof(BatteryInfo) | BatteryInfo对象长度  BatteryInfo object length |
| **Data** | BatteryInfo | BatteryInfo成员如下  BatteryInfo members are as follows |

BatteryInfo {

Unsigned char state; //电池状态 Battery status e.g. 0:正常使用Normal 1:充电中Charging 2:充满Full 3:低电量Low power

Unsigned char percent; //电池状态 Battery status e.g. 电池电量百分比Battery charge percentage

Unsigned short voltage; //电池电压 Battery voltage(mV) e.g. 3950 : 3.95V

}

## Private command(0x00-0xDF)

### 血压计Blood pressure monitor (AirBP Plus)

蓝牙透传服务Bluetooth transparent transmission service

Service UUID : 14839ac47d7e415c9a42167340cf2339

Character TX UUID: 8b00ace7eb0b49b0bbe99aee0a26e1a3

Character RX UUID: 0734594aa8e74b1aa6b1cd5243059a57

#### 0x00:获取配置参数 Get configuration parameters

|  |  |  |
| --- | --- | --- |
| **Send** | | **Description** |
| **Length** | 0x0000 |  |
| **Data** | N/A |  |
| **Receive** | | **Description** |
| **Length** | Sizeof(Configuartion) | Configuartion对象长度  Configuartion object length |
| **Data** | Configuartion | Configuartion成员如下  Configuartion members are as follows |

Configuartion {

unsigned int prev\_calib\_zero; //上一次校零adc值 The adc value of zero calibration last time. e.g. 489335<=zero<=1607816

unsigned int last\_calib\_zero; //最后一次校零adc值 The last time the adc value is zeroed. e.g. 489335<=zero<=1607816

unsigned int calib\_slope; //校准斜率值 Calibration slope value\*100 e.g. 5368<=slope<=7000 136.3LSB/mmHg-170.4LSB/mmHg

unsigned short slope\_pressure; //校准斜率时用的压力值 Pressure value for calibrating slope

unsigned short stop\_pressure; //停止打气压力值 Stop pumping pressure value. unit:mmHg

time\_t sleep\_ticks; //上次进休眠时间 Sleep time last time.

time\_t calib\_ticks; //最后一次校准时间 The last time of calibrating time.

time\_t reset\_ticks; //上次重启进低功耗模式时间 Last restart into low energy mode time

unsigned char sleep\_flag; //是否进入低功耗模式标志 The flag that whether enter low energy mode.

unsigned char beep\_switch; //0 打开蜂鸣器Turn on the buzzer. 1关闭蜂鸣器Turn off the buzzer.

unsigned char reserved[3]; //预留 Reserved.

}

#### 0x04: 启动测量Start measurement

双向通信,当主机往上位机发送时,为通知主机已进入测量

（测量结束后静态压降至15mmHg发送**停止测量**通知）

Use duplex port for communication . When the host sends to the master computer, it is notify that the host has entered into measurement.

(After the measurement, if the static pressure drops to 15mmHg, a notification of stopping measurement will be sent.)

|  |  |  |
| --- | --- | --- |
| **Send** | | **Description** |
| **Length** | Sizeof(SendRate) | 如无参数，默认为原始采样率  With no parameter, the default is original sampling rate with no parameter. |
| **Data** | SendRate |
| **Receive** | | **Description** |
| **Length** | Sizeof(RealTimePressure) | RealTimePressure对象长度  RealTimePressure object length |
| **Data** | RealTimePressure | RealTimePressure成员如下  RealTimePressure members are as follows |

SendRate {

Unsigned char rate; // 数据发送频率Data transmission frequency(Hz)

}

RealTimePressure {

Short pressure\_static; // 静态压通道实时压The real-time pressure of static pressure channel.（mmHg）\*100

Short pressure\_pulse; // 脉搏通道实时压The real-time pressure of pulse channel.（mmHg）\*100

}

#### 0x05: 停止测量Stop measurement

~~双向通信,当主机往上位机发送时,为通知主机已停止测量~~

~~Use duplex port for communication. When the host sends to the master computer, it is notify that the measurement of host has been stopped.~~

Only receive this command from the device and use it as a measurement state.

|  |  |  |
| --- | --- | --- |
| **Send** | | **Description** |
| **Length** | 0x0000 |  |
| **Data** | N/A |  |
| **Receive** | | **Description** |
| N/A | | 无需返回  No return |

#### 0x06: 当前运行状态(提示信息,当算法运行状态切换时发送) Current running status (Prompt message and sent when the algorithm running state is switched)

|  |  |  |
| --- | --- | --- |
| **Send** | | **Description** |
| **Length** | 0x0001 |  |
| **Data** | RunStatus | 当前运行状态  Current running status |
| **Receive** | | **Description** |
| **Length** | 0x0000 |  |
| **Data** | N/A |  |

enum RunStatus {

// 以算法测量状态为主 Mainly based on algorithm measurement status

}

#### 0x07:测量结果(测量结束时主机主动发送) Measurement result(The host takes the initiative to send at the end of the measurement)

|  |  |  |
| --- | --- | --- |
| **Send** | | **Description** |
| **Length** | 0x0000 |  |
| **Data** | N/A |  |
| **Receive** | | **Description** |
| **Length** | Sizeof(BloodPressureResult) | BloodPressureResult对象长度  BloodPressureResult object length |
| **Data** | BloodPressureResult | BloodPressureResult成员如下  BloodPressureResult members are as follows |

BloodPressureResult {

Unsigned char time[7]; // 时间 time e.g.0xE1070301090000:2017-03-01

Unsigned char state\_code; // 状态码Status code

Unsigned short [systolic](javascript:void(0);)\_[pressure](javascript:void(0);); // 收缩压Systolic blood pressure

Unsigned short [diastolic](javascript:void(0);)\_[pressure](javascript:void(0);); // 舒张压Diastolic blood pressure

Unsigned short mean\_pressure; // 平均压Mean blood pressure

Unsigned short pulse\_rate; // 脉率Pulse rate

}

#### 0x08: 工程启动测量(工程模式)Engineering start measurement (engineering mode)

|  |  |
| --- | --- |
| **Send** | **Description** |
| 同0x04:启动测量  Same as **0x04: Start measurement** | |
| **Receive** | **Description** |
| 见0x04:启动测量，返回指令为0x04  See **0x04: Start measurement**，and the return instruction is 0x04 | |

**Notes：和正常启动测量区别在于测量会持续到静态压为15mmHg以下才结束**

**Notes: The difference from normal start-up measurement is that the measurement will continue until the static pressure is below 15mmHg.**

#### 0x09: 蜂鸣器开关Buzzer switch

|  |  |  |
| --- | --- | --- |
| **Send** | | **Description** |
| **Length** | Sizeof(BeepStatus) | BeepStatus对象长度  BeepStatus object length |
| **Data** | BeepStatus | BeepStatus成员如下  BeepStatus members are as follows |
| **Receive** | | **Description** |
| **Length** | 0x0000 |  |
| **Data** | N/A |  |

CalibrationZero {

Unsigned short stus; // e.g. 0 打开蜂鸣器Turn on the buzzer. 1关闭蜂鸣器 Turn off the buzzer.

1. 制定协议时同一种指令协议包类型尽量保持一种数据格式 [↑](#footnote-ref-0)