HJ281-ETH(以太网）振动温度传感器规格说明书

版本 V1.0

上海衡简智能技术有限公司
Shanghai HengJian Intelligent Co.Ltd

产品简介

HJ281-ETH 振动传感器是新一代直接采用网口传输，可实时连续不间断传输三轴加速度波形数据的传感器。

采样速率默认 4000HZ(可支持 8000HZ，16000 或者 32000HZ)；

采样分辨率 16 位 AD

测量通道： 3 轴

9~24V 供电。

接口方式：10/100M 网口

通讯协议:TCP 协议

防护等级：IP67，表面耐腐蚀；

防爆等级：EXia Ⅱ BT4；

外形尺寸：26（直径）×71（高度）mm；

安装方式：磁吸座/M10x1.5 螺纹孔/胶粘可选；

传感器输出：

三轴加速度值（±16G）；

三轴速度值(0-100mm/s)；

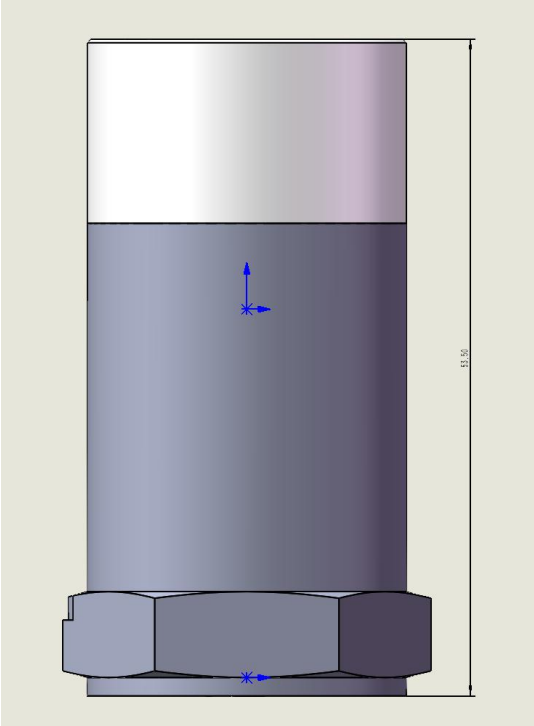
三轴位移值(0-8mm)；

三轴频谱最高频率以及幅度值

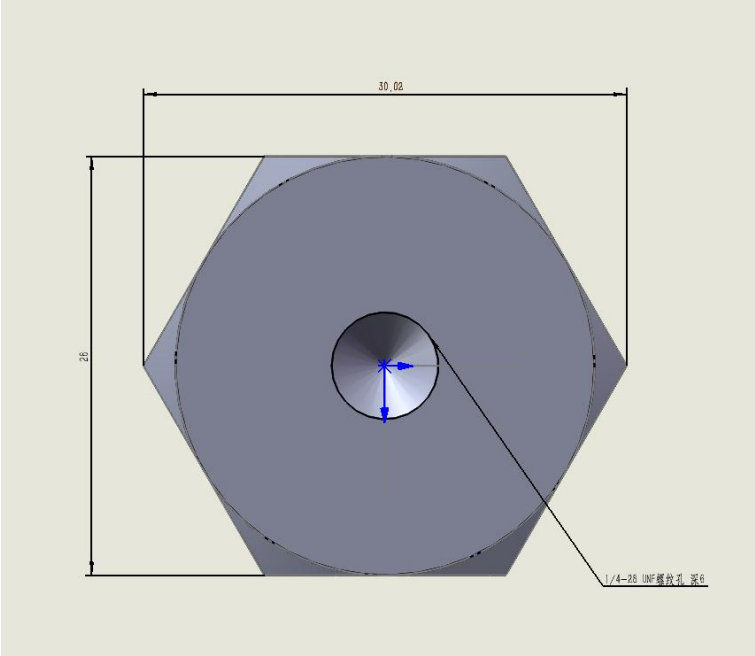
三轴重力矢量或倾斜度（可判断传感器位置是否有移动）；

免费提供数据采集工具以及 C#源码（数据采集并记录为 csv 格式）

尺寸图



六边形缺角指向为 X 轴方向。



硬件接线

DC 接头规格 DC5.5*2.1
网口规格 常规水晶头

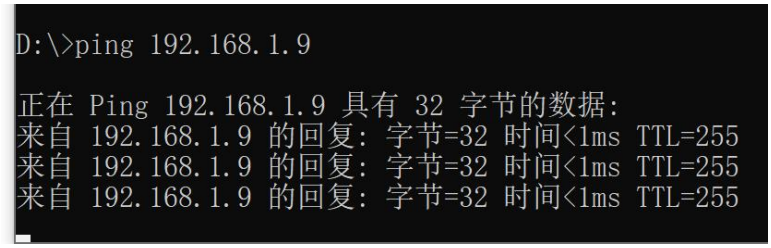
缺省设置

设备 IP 地址： 192.168.1.9 网关 192.168.1.1 子网掩码 255.255.255.0
工作方式 TCP 服务器端 TCP 端口号： 5001

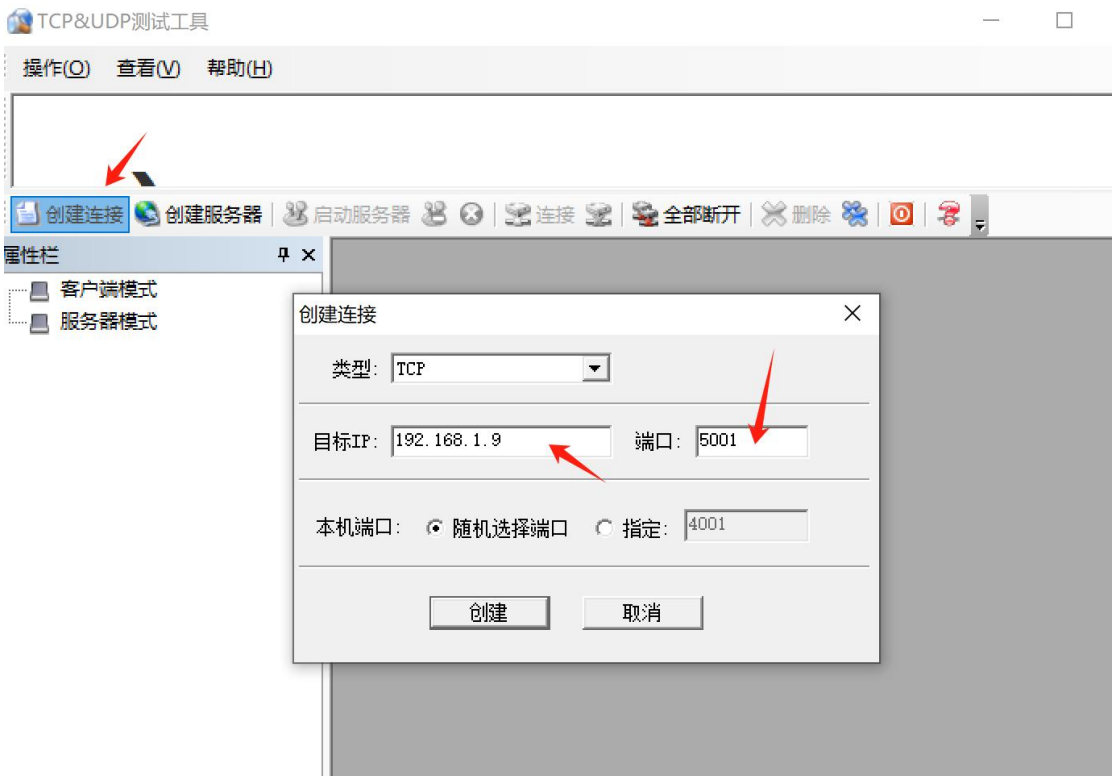
测试过程

- 将传感器通过网线与电脑网口直接相连接，并给传感器上电
- 将电脑的网口 ip 设置如下图

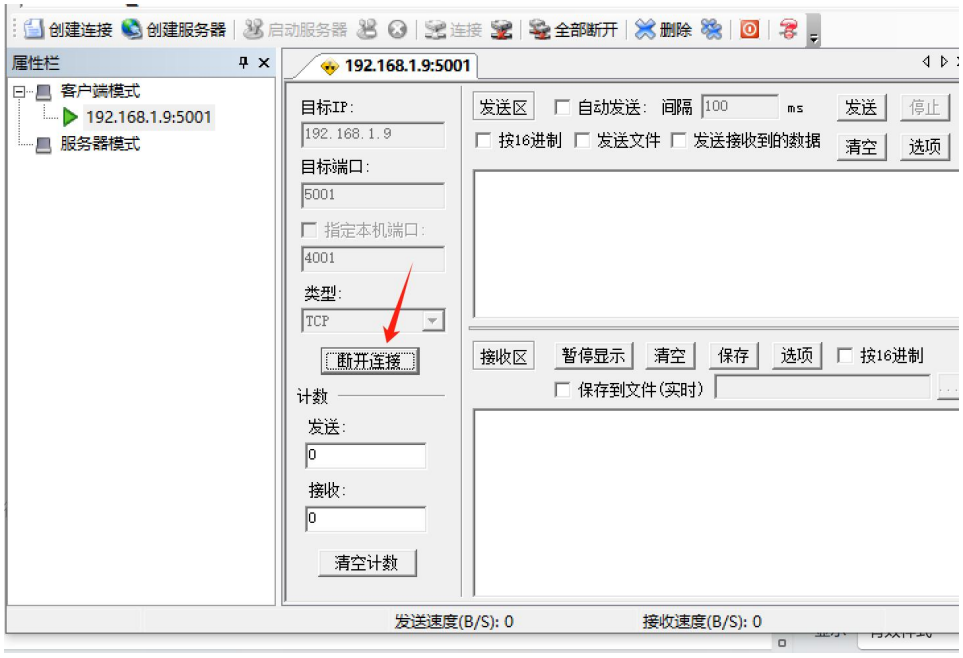
- 通过电脑 ping 一下，看看是否与传感器相通。如果不通，可考虑给传感器重新上电。



- 打开 tcp 测试工具，并建立一个新链接，按图示，最后按创建



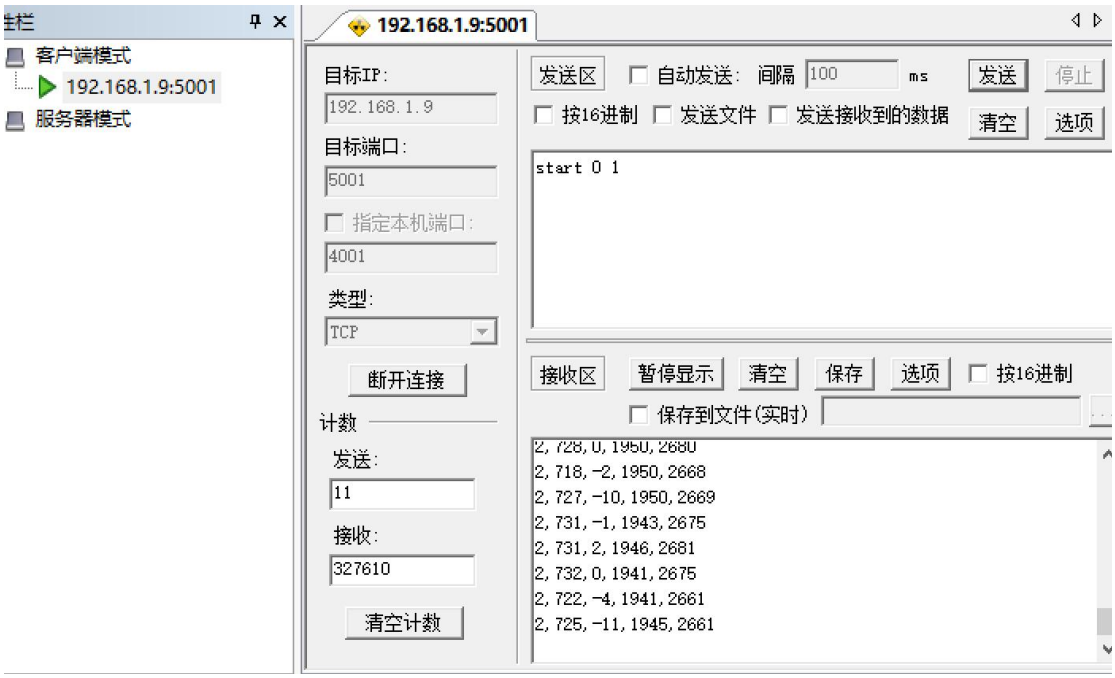
- 建立 tcp 链接，如果传感器正常，则可以看到图示链接成功



- 输入命令

输入 start 0 1， 并且最后按住 ctrl 键同时按回车键（该软件工具不支持直接输入回车，需要按住 ctrl 再按回车才能输入回车符）

然后按发送，就可以看到接收区中有传感器输出



以上步骤实现的话，即表明传感器工作正常。

传感器命令格式说明

传感器命令使用常规字符串，以\n\r 结束。下面命令说明省略\n\r

- 采集命令 start

命令格式 start x y， 其中 x 代表一次采集的数目（数据实际采集量为该值 x1024），y 代表数据上传的格式。

当 x 是 0 的时候，表示连续采集不要停止。

（注：若要使用采集时长的命令，可参考 start_sec 命令）

y 取值范围为 0，1，2，3

0 - 二进制格式包

1- csv 格式文件（方便复制后直接用 excel 或者 wps 打开分析）

2- 特征值文本 json 格式

3- 特征值二进制格式

举例，例如要采集 4096 个采样点的数据，并且以 csv 格式输出（对于 4000HZ 采样率而言，接近 1 秒时间），命令为

start 4 1

收到数据为

若要连续不停持续输出，则命令为

start 0 1

传感器收到采集命令之后即开始采集并按照指定的格式立即发送数据。

- 停止采集命令

stop

这个命令发给传感器之后，传感器即停止采集。

- **格式 1-csv 格式文件输出例子：**

1, 728, -9, 1934, 2654

1, 725, -8, 1935, 2653

1, 727, -6, 1941, 2663

1, 723, -1, 1945, 2668

1, 723, 2, 1931, 2657

1, 723, 1, 1926, 2651

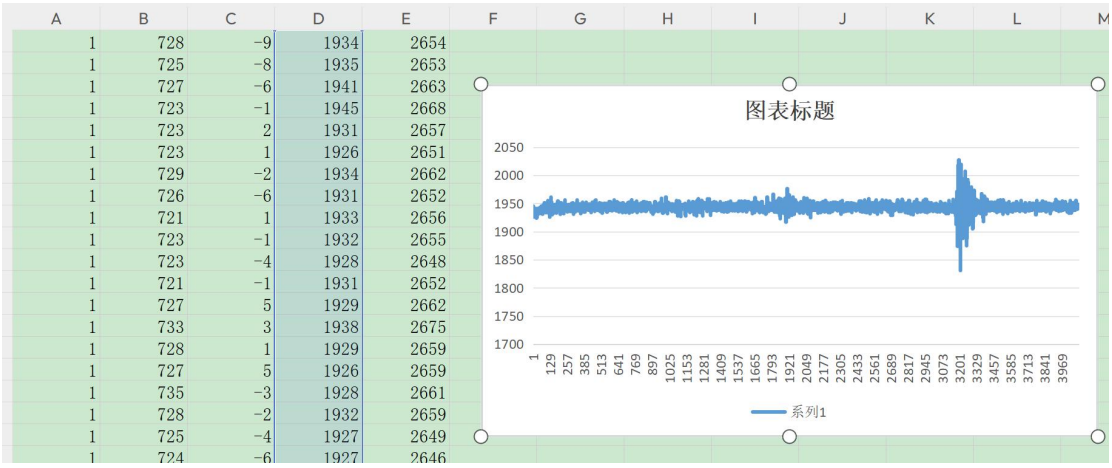
1, 729, -2, 1934, 2662

1, 726, -6, 1931, 2652

1, 726, -6, 1931, 2652

第一列为包序列号（若正常则同一个值重复 1024 次），从 1 到 4 循环，若中间有断开，则表示可能有数据包丢失。后面分别是 x，y，z 加速度数据，最后一列为前四者数字之后，用于校验作用。

将数据从 tcp 工具中复制到文本文件编辑器并另存为 csv 格式，则可以直接用 wps 或者 excel 打开。如图示。这种格式方便客户直接查看振动数据。



格式 0-二进制格式包说明：

格式 0 为二进制包格式。这种格式数据量最小，在 8K 或者更高采样率情况下需要采用。

单个包定义如下：

```
#define SAMPLES_PER_PACKET 1024
#define AXIS 3
struct ACC_PACKET
{
    uint32_t header_tag;   //包头，固定为 0xfefefefe
    uint32_t pk;
    uint32_t size;
    int16_t xyz[SAMPLES_PER_PACKET*AXIS];
```

```
    uint32_t crc;          //crc 校验值，包含 header_tag 到 xyz 结束
    uint32_t tail_tag;     //包尾部，固定为 0x12121212
};
```

即每个包以包头 0xfefefefe 开头，之后是包序列号，每次累加。size 为固定的 0x1800 字节，即为 xyz 数组的数据量 6144（1024*2*3）字节。xyz 为数据量，共 1024 个 3 轴采样。以 int16 格式，每个占用 2 个字节。按照 x,y,z 排列。crc 校验算法参考本文件附录部分。另外，可申请数据采集工具以及 C#源代码。

下面数据为 start 1 0 之后收到的数据（总共 6144 字节）

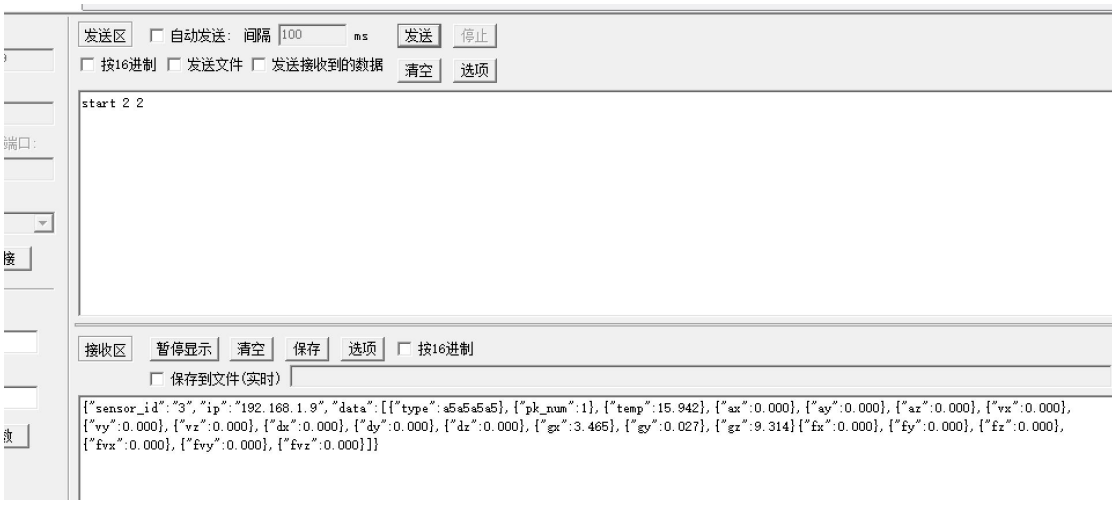
fe fe fe fe 01 00 00 00 00 18 00 00 d4 02 fc ff 86 07 d8 02 ff ff 95 07 d7 02 02 00 95 07 d9 02 07 00 9a 07 d4 02 01 00 a1 07 d7 02 f6 ff a6 07 d4 02 f9 ff 97 07 d4 02 f8 ff 9d 07 d5 02 f1 ff

（此处省略）.....ff

94 07 db 02 00 00 97 07 d6 02 fe ff 95 07 d9 02 fe ff 9a 07 d9 02 04 00 99 07 d6 02 00 00 97 07 d9 02 fc ff 91 07 d1 02 00 00 91 07 db 02 07 00 93 07 d9 02 09 00 8f 07 db 02 fd ff 9a 07 d3 02 02 00 97 07 d4 02 03 00 96 07 d7 02 04 00 98 07 d8 02 fd ff 98 07 e7 63 d6 b3 12 12 12 12

- **格式 2-特征值文本 json**

格式 2 为特征值格式，即振动综合总量的方式，以 json 格式提供，末尾以\n\r 结束



{"sensor_id":"3","ip":"192.168.1.9","data":[{"type":a5a5a5a5},{"pk_num":1},{"temp":15.942},{"ax":0.000},{"ay":0.000},{"az":0.000},{"vx":0.000},{"vy":0.000},{"vz":0.000},{"dx":0.000},{"dy":0.000},{"dz":0.000},{"gx":3.465},{"gy":0.027},{"gz":9.314}{"fx":0.000},{"fy":0.000},{"fz":0.000},{"fvx":0.000},{"fvy":0.000},{"fvz":0.000}]}

下面为具体解析说明：

{"sensor_id":"3",　　-传感器编号，客户可修改

"ip":"192.168.1.9",　-传感器 ip 地址

"data":[{"type":a5a5a5a5},　-传感器类型，对此型号固定为 a5a5a5a5

{"pk_num":1},　　-数据包序列号，每次传输累加

{"temp":15.942},　　-温度（此温度为大概温度，精度较低）
{"ax":0.000},{"ay":0.000},{"az":0.000},　-振动加速度量，单位 m/s²
{"vx":0.000},{"vy":0.000},{"vz":0.000},　-振动速度量，单位 mm/s
{"dx":0.000},{"dy":0.000},{"dz":0.000},　-振动位移量，单位 um
{"gx":3.465},{"gy":0.027},{"gz":9.314}　-重力作用分量，单位 m/s²
{"fx":0.000},{"fy":0.000},{"fz":0.000},　-三个方向的振动频率，单位为 Hz
{"fvx":0.000},{"fvy":0.000},{"fvz":0.000]} -三个方向上加速度 FFT 之后的最大幅值，单位 m/s²

- **格式 3-特征值二进制格式**

格式 3 也是特征值的输出，不过直接用二进制格式输出。数据解析参考格式 2.
数据包格式为；

```
 struct RMS_BUF
{
    uint32_t type;
    uint32_t ts; //固定为 0
    uint16_t pk_num;
    uint16_t voltage;
    float temp;
    float  xrms;
    float  yrms;
    float  zrms;
    float xvrms;
    float yvrms;
    float zvrms;
    float xdrms;
    float ydrms;
    float zdrms;
    float xaver;
    float yaver;
    float zaver;
    float xfreq;
     float yfreq;
    float zfreq;
    float xvalue;
    float yvalue;
    float zvalue;
    uint32_t crc;  //crc 为从头到 zvalue
    uint32_t tail; //固定为 0x13131313

};
```

其他命令

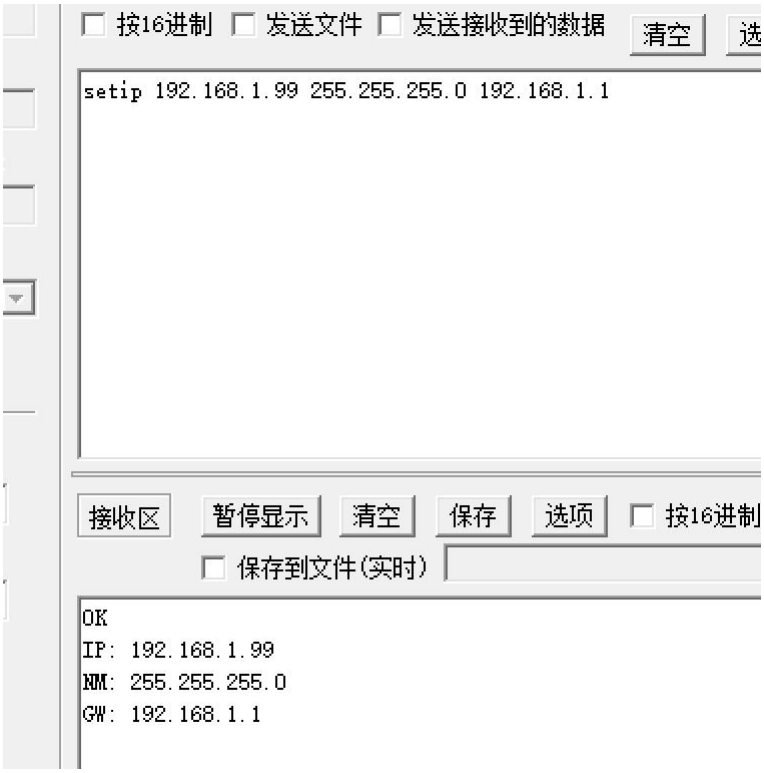
- 设定传感器 IP 地址（需要将传感器接入网关或者路由器）

setip xxx yyy zzz

xxx 为 ip 地址，yyy 为掩码，zzz 为网关

举例设定传感器 ip 为 192.168.1.99

传感器返回 OK 并输出设定好的 IP 地址



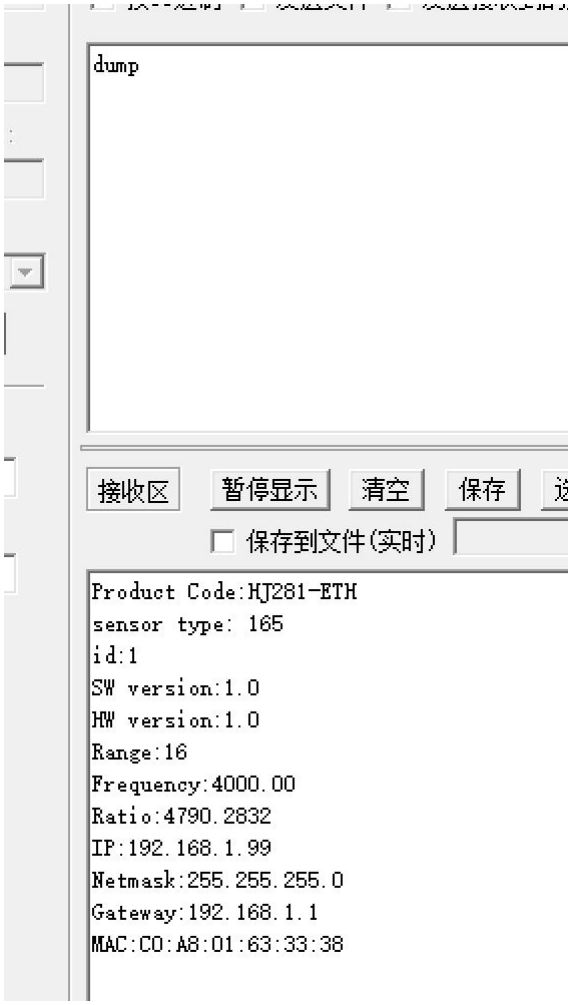
若要设置生效，则需要下发重启传感器命令

- 重启传感器命令

reset

- 查看传感器信息

dump



Product Code:HJ281-ETH

sensor type: 165 -类型，振动类型

id:1 -传感器编号

SW version:1.0

HW version:1.0

Range:16 -传感器振动量程 16g

Frequency:4000.00 - 传感器采样频率

Ratio:4790.2832 -转换因子，即传感器获得的加速度原始数据乘以该值后得到 um/s^2 为单位

IP:192.168.1.99 -传感器 IP

Netmask:255.255.255.0 -传感器网络掩码

Gateway:192.168.1.1 -传感器网关

MAC:C0:A8:01:63:33:38 -传感器 MAC 地址

- start_sec

请求采集固定时长数据。命令类似 start x y
不同点是 start_sec 参数 x 代表是采集时长，单位为秒。 y 同 start 一致，表示上传格式。

附录一
CRC 算法

```
static const unsigned int CRC32_Table[256] =
{
    0x00000000, 0x77073096, 0xEE0E612C, 0x990951BA,
    0x076DC419, 0x706AF48F, 0xE963A535, 0x9E6495A3,
    0x0EDB8832, 0x79DCB8A4, 0xE0D5E91E, 0x97D2D988,
    0x09B64C2B, 0x7EB17CBD, 0xE7B82D07, 0x90BF1D91,
    0x1DB71064, 0x6AB020F2, 0xF3B97148, 0x84BE41DE,
    0x1ADAD47D, 0x6DDDE4EB, 0xF4D4B551, 0x83D385C7,
    0x136C9856, 0x646BA8C0, 0xFD62F97A, 0x8A65C9EC,
    0x14015C4F, 0x63066CD9, 0xFA0F3D63, 0x8D080DF5,
    0x3B6E20C8, 0x4C69105E, 0xD56041E4, 0xA2677172,
    0x3C03E4D1, 0x4B04D447, 0xD20D85FD, 0xA50AB56B,
    0x35B5A8FA, 0x42B2986C, 0xDBBBC9D6, 0xACBCF940,
    0x32D86CE3, 0x45DF5C75, 0xDCD60DCF, 0xABD13D59,
    0x26D930AC, 0x51DE003A, 0xC8D75180, 0xBFD06116,
    0x21B4F4B5, 0x56B3C423, 0xCFBA9599, 0xB8BDA50F,
    0x2802B89E, 0x5F058808, 0xC60CD9B2, 0xB10BE924,
    0x2F6F7C87, 0x58684C11, 0xC1611DAB, 0xB6662D3D,
    0x76DC4190, 0x01DB7106, 0x98D220BC, 0xEFD5102A,
    0x71B18589, 0x06B6B51F, 0x9FBFE4A5, 0xE8B8D433,
    0x7807C9A2, 0x0F00F934, 0x9609A88E, 0xE10E9818,
    0x7F6A0DBB, 0x086D3D2D, 0x91646C97, 0xE6635C01,
    0x6B6B51F4, 0x1C6C6162, 0x856530D8, 0xF262004E,
    0x6C0695ED, 0x1B01A57B, 0x8208F4C1, 0xF50FC457,
    0x65B0D9C6, 0x12B7E950, 0x8BBEB8EA, 0xFCB9887C,
    0x62DD1DDF, 0x15DA2D49, 0x8CD37CF3, 0xFBD44C65,
    0x4DB26158, 0x3AB551CE, 0xA3BC0074, 0xD4BB30E2,
    0x4ADFA541, 0x3DD895D7, 0xA4D1C46D, 0xD3D6F4FB,
    0x4369E96A, 0x346ED9FC, 0xAD678846, 0xDA60B8D0,
    0x44042D73, 0x33031DE5, 0xAA0A4C5F, 0xDD0D7CC9,
    0x5005713C, 0x270241AA, 0xBE0B1010, 0xC90C2086,
    0x5768B525, 0x206F85B3, 0xB966D409, 0xCE61E49F,
    0x5EDEF90E, 0x29D9C998, 0xB0D09822, 0xC7D7A8B4,
    0x59B33D17, 0x2EB40D81, 0xB7BD5C3B, 0xC0BA6CAD,
    0xEDB88320, 0x9ABFB3B6, 0x03B6E20C, 0x74B1D29A,
    0xEAD54739, 0x9DD277AF, 0x04DB2615, 0x73DC1683,
    0xE3630B12, 0x94643B84, 0x0D6D6A3E, 0x7A6A5AA8,
    0xE40ECF0B, 0x9309FF9D, 0x0A00AE27, 0x7D079EB1,
```

```
    0xF00F9344, 0x8708A3D2, 0x1E01F268, 0x6906C2FE,
    0xF762575D, 0x806567CB, 0x196C3671, 0x6E6B06E7,
    0xFED41B76, 0x89D32BE0, 0x10DA7A5A, 0x67DD4ACC,
    0xF9B9DF6F, 0x8EBEEFF9, 0x17B7BE43, 0x60B08ED5,
    0xD6D6A3E8, 0xA1D1937E, 0x38D8C2C4, 0x4FDFF252,
    0xD1BB67F1, 0xA6BC5767, 0x3FB506DD, 0x48B2364B,
    0xD80D2BDA, 0xAF0A1B4C, 0x36034AF6, 0x41047A60,
    0xDF60EFC3, 0xA867DF55, 0x316E8EEF, 0x4669BE79,
    0xCB61B38C, 0xBC66831A, 0x256FD2A0, 0x5268E236,
    0xCC0C7795, 0xBB0B4703, 0x220216B9, 0x5505262F,
    0xC5BA3BBE, 0xB2BD0B28, 0x2BB45A92, 0x5CB36A04,
    0xC2D7FFA7, 0xB5D0CF31, 0x2CD99E8B, 0x5BDEAE1D,
    0x9B64C2B0, 0xEC63F226, 0x756AA39C, 0x026D930A,
    0x9C0906A9, 0xEB0E363F, 0x72076785, 0x05005713,
    0x95BF4A82, 0xE2B87A14, 0x7BB12BAE, 0x0CB61B38,
    0x92D28E9B, 0xE5D5BE0D, 0x7CDCEFB7, 0x0BDBDF21,
    0x86D3D2D4, 0xF1D4E242, 0x68DDB3F8, 0x1FDA836E,
    0x81BE16CD, 0xF6B9265B, 0x6FB077E1, 0x18B74777,
    0x88085AE6, 0xFF0F6A70, 0x66063BCA, 0x11010B5C,
    0x8F659EFF, 0xF862AE69, 0x616BFFD3, 0x166CCF45,
    0xA00AE278, 0xD70DD2EE, 0x4E048354, 0x3903B3C2,
    0xA7672661, 0xD06016F7, 0x4969474D, 0x3E6E77DB,
    0xAED16A4A, 0xD9D65ADC, 0x40DF0B66, 0x37D83BF0,
    0xA9BCAE53, 0xDEBB9EC5, 0x47B2CF7F, 0x30B5FFE9,
    0xBDBDF21C, 0xCABAC28A, 0x53B39330, 0x24B4A3A6,
    0xBAD03605, 0xCDD70693, 0x54DE5729, 0x23D967BF,
    0xB3667A2E, 0xC4614AB8, 0x5D681B02, 0x2A6F2B94,
    0xB40BBE37, 0xC30C8EA1, 0x5A05DF1B, 0x2D02EF8D
};

/**
 * [_getCRC32 使用查表法计算 CRC32]
 * @param   buffer      [缓存地址]
 * @param   bufferLen [缓存长度]
 * @return              [计算出的 CRC32 值]
 */
unsigned int _getCRC32(uint32_t *flash_addr, unsigned int bufferLen)
{
    unsigned int crc32Value = 0xffffffff;

    unsigned char *pTmpBuffer = (uint8_t*)flash_addr;

    while(bufferLen--)
    {
```

```
            crc32Value = CRC32_Table[(crc32Value^ *pTmpBuffer++) & 0xff] ^ (crc32Value >> 8);
        }

        return (crc32Value^0xffffffff);
}
```

附录二

采集工具用法

运行命令行窗口，三个参数分别是传感器 ip 地址，采集数量，要存储成的文件名称

```
D:\>collect.exe 192.168.1.9 10 data.csv
```

需要 C#源码的话在购买了传感器之后请联系厂家提供。