

# **KETRON SDK docking instructions**

## **一、引言 introduction**

对接前，可以先了解科强手环的基本操作 Before docking, you need to learn the basic operation of KETRON Bracelet:

<https://v.qq.com/x/page/n13371v0ml5.html>

## **二、开发情况说明 Development description**

手环是通过 TCP 协议发送数据，需要服务器搭建 TCP 服务端。手环为 TCP 客户端，会自动往预设的服务端 IP 地址(或域名地址) 上传数据 The bracelet sends data through TCP protocol, which requires the server to build the TCP server. The bracelet is a TCP client and will automatically upload data to the preset IP address (or domain name address) of the server.

### **1. 开发前的基本环境搭建流程 Basic environment building process before development**

1. 手环插上当地运营商 SIM 卡 换手机卡后要关机再重启，确保手环通讯，开通流量上网 The bracelet should be inserted with the SIM card of the local operator. After changing the mobile phone card, it should be turned off and then restarted to ensure the communication of the bracelet .the SIM card should open the flow of Internet.

2. 设置手环上传地址：发送手环 SIM 卡短信服务器地址，修改手环服务器地址命令格式：

SETIP:URL 举例： SETIP:120.24.58.119:6100(注意格式不能有空格,换行等特殊字符.)

该 IP 地址就是服务器搭建的 TCP 地址。 Set the upload address of the bracelet: send the SMS server address of the SIM card of the bracelet, and modify the command format

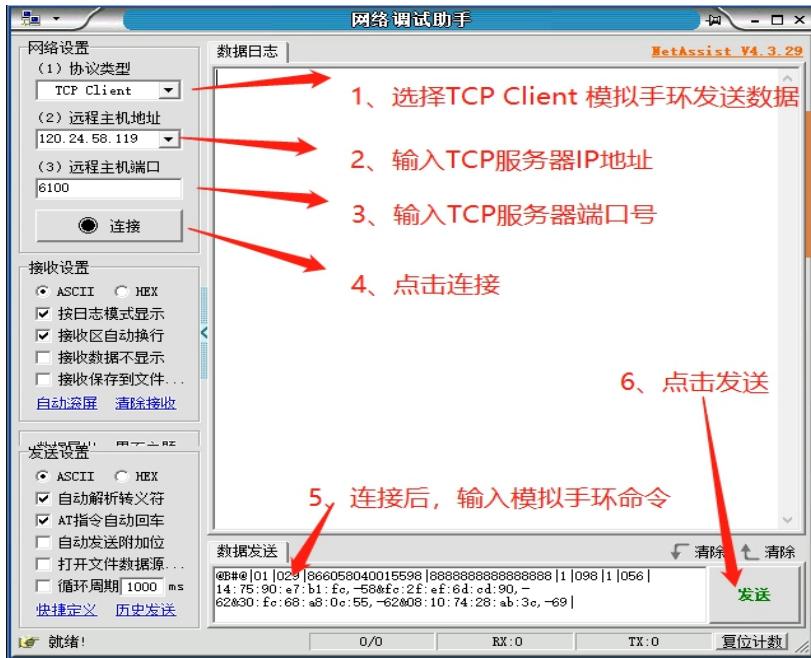
of the server address of the bracelet:SETIP:URL For example: SETIP:120.24.58.119 : 6100 (**note that the format can't have special characters such as space, newline, etc.,The IP address is the TCP address set up by the server**)



3 )通过以上设置，必须佩戴手环，手环才会自动上传健康数据。Through the above settings, you must wear a bracelet to upload health data automatically.

### 三.设置服务端 Set the server

- 1) 确保 TCP 服务器无误运行，**打开** TCP 服务器相关的地址端口 **防火墙** Ensure that the TCP server is running correctly, and **open the firewall** of address port related to TCP server .
2. ) 可以使用我们提供的 **TCP 测试小工具.exe** 进行测试，确保能够连通访问。you can test with the TCP test gadget.Exe provided by us to ensure access is connected.



3)通过以上设置服务器就能够接收手环的命令，然后通过命令解析进行采集有效数据进行业务处理，同时下发回复对应的命令，命令具体看 Through the above settings, the server can receive the command of the bracelet, and then collect the effective data for business processing through command analysis, and issue the corresponding command to reply. See the details of the command.

## **开发代码简短说明 Brief description of development code**

我司有两份 SDK 协议包用于设备对接 , 分别为 Java 和 .Net(C#) 版本 Our company has two SDK protocol packages for device docking, namely Java and. Net (C #) version.

接口API > 手环对接平台 > SDK协议示例 仅供参考			
名称	修改日期	类型	大小
Java1.1	2020/4/22 15:22	文件夹	
NET1.1	2020/4/22 15:22	文件夹	

在拿到我司提供的文档后 , 可以先对应 After getting the documents provided by our company, you can first map



这份文档规定协议内容查看 SDK 包 Demo 代码 , 清楚设备功能及代码协议 , 再进行开发。接下来就 C# 版本 Demo 做简单介绍。 This document specifies the protocol content, check the SDK package demo code, clear the device function and code protocol, and then carry out the development. Next, we will give a brief introduction to C # version demo.

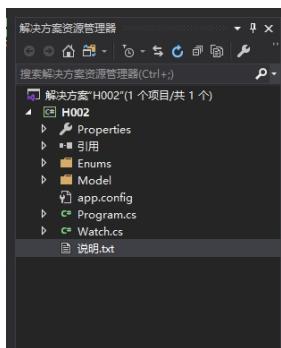
首先 , 打开 SDK 文件 .Net1.1, 将会看到如下内容 : First, open the SDK file. Net1.1, and you will see the following:

接口API > 手环对接平台 > SDK协议示例 仅供参考 > NET1.1

名称	修改日期	类型	大小
.vs	2020/4/22 15:22	文件夹	
H002	2020/4/22 15:22	文件夹	
C#的配代码_仅供参考.cs	2018/5/31 9:30	Visual C# Sourc...	90 KB
H002.sln	2018/3/26 14:30	Visual Studio Sol...	1 KB



其次，将.Net Demo 在 VS 中打开，你将会看到如下内容 Next, open the. Net demo in VS and you will see the following:



最后，将 **说明.txt** 打开，可以了解到几个文档内容含义 Finally, open the description. Txt to understand the meaning of several documents,

Enums/ --- Possible enumeration types

Model/ --- Command message for model processing and transformation

SDK/Watch.cs --- A few examples of the development of the bracelet docking

their own platform, refer to the development of the bracelet access platform to see

the main self-development

```

    //</summary>
    /// <param name="cmdStr">命令代码</param>
    public C001(string cmdStr)
    {
        var data = cmdStr.Split('|');
        var dataLength = data.Length;
        //正确格式下 最小长度为12
        if (data[0] == "@E#@" && data[2] == data[dataLength - 1] == "@E#@")
        {
            HeadCode = data[0];
            Protocol = data[1];
            IMEI = data[3];
            IMSI = data[4];
            WearState = int.Parse(data[5]);
            Power = int.Parse(data[6]);
            TerminalTime = data[7];
            LbsType = int.Parse(data[dataLength - 2]);
            FootCode = data[dataLength - 1];
            //GPS数据
            if (LbsType == 1)
            {
                var Lat = Convert.ToDecimal(data[8]);
                var Lng = Convert.ToDecimal(data[9]);
                GpsString = Lat + "|" + Lng;
            }
            //基站数据
            if (LbsType == 2)
            {
                ; //LBS解析
            }
            //wifi数据+基站数据
            if (LbsType == 3)
            {
                WifiCellString = data[8] + "|" + data[9];
            }
        }
        else {
            Protocol = null;
        }
    }

    public override string ToString()
    {
        var rString = HeadCode + "|";
    }
}

```

每个 model 都对应协议的指令 code , 可以根据代码逻辑进行数据对接测试 Each model corresponds to the protocol instruction code, which can be used for data docking test according to the code logic.

## 注意事项 Precautions (h002)

- 定位数据优先 wifi ( wifi 版 ) > GPS > 基站 ; GPS 还是 Wifi ( 仅仅支持 wifi 版 ) 可以通过刷固件 设置那个优先: The first priority of positioning data is WiFi > GPS > base station; GPS or WiFi (WiFi version only) can set that priority through firmware brushing.

- 红黄绿红红 依次按 这三个键可以进入工厂测试模式 , 最后一步是测试 GPS 的 , 可以进入

工厂测试模式测试一下: Red, yellow, green, red and red press these three keys in turn to enter the factory test mode. The last step is to test GPS. You can enter the factory test mode to test.

3、基站定位接口问题：对于接入自己平台的注意，手环定位上传 有 wifi 数据，基站数据，这 2 类数据是需要 第三方 lbs 数据平台服务商进行经纬度解析获取位置。 推荐高德 LBS 服务 (可百度相关接口) Base station positioning interface problem: for the attention of accessing your own platform, there are WiFi data and base station data uploaded by the bracelet positioning. These two types of data require the third-party lbs data platform service provider to analyze the longitude and latitude to obtain the location. Recommend Gaode LBS service (Baidu related interface is available).

4、手环命令上传模式：手环主动发送命令码到服务器，服务器接收后要有下发结束命名，否则重复收到上一条命令: smart bracelet command upload mode: the bracelet sends command code to the server actively. After receiving, the server shall have the name of issuing end, otherwise, the last command will be received repeatedly.

5. 手环命令下发模式：手环主动连接后，在下发对应的回复命令，同时拼接你需要下发的命令一起下发 (一般建立下发队列，进行离线命令)。Bracelet command distribution mode: after the active connection of the bracelet, the corresponding reply command is issued, and the commands you need to issue are spliced together (generally, the distribution queue is established for offline command).

## **下发命令部分示例展示 Example of issuing command**

### **1. GPS 数据上传场景 Scene of uploading GPS data :**

1.1 . 上行当前位置数据包 Uplink current location packet

示例 Examples:

@B#@|01|001|11111222223333|88888888888888|1|55|20160715150323|125.482  
76|37.615124|1|@E#@

1.2 . 下行确认位置数据包 Downlink confirm location packet :

示例 Examples:

@B#@|01|002|11111222223333|0|20160729173850|@E#@

### **2. 开机场景 Startup scenario**

2.1 . 上行开机数据包 Uplink boot packet

示例 Examples:

@B#@|01|003|11111222223333|88888888888888|1.0.1|1|55|20160715150323|12  
5.48276|37.615124|1|@E#@

2.2 . 下行开机确认数据包 Downlink boot confirmation packet

示例 Examples:

@B#@|01|004|11111222223333|0|20160729174051|@E#@

### 3 关机场景 Shutdown scenario

#### 3.1. 上行关机数据包 Uplink shutdown packet

示例 Examples:

```
@B#@|01|005|11111222223333|88888888888888|55|20160715150323|125.4827  
6|37.615124|1|@E#@
```