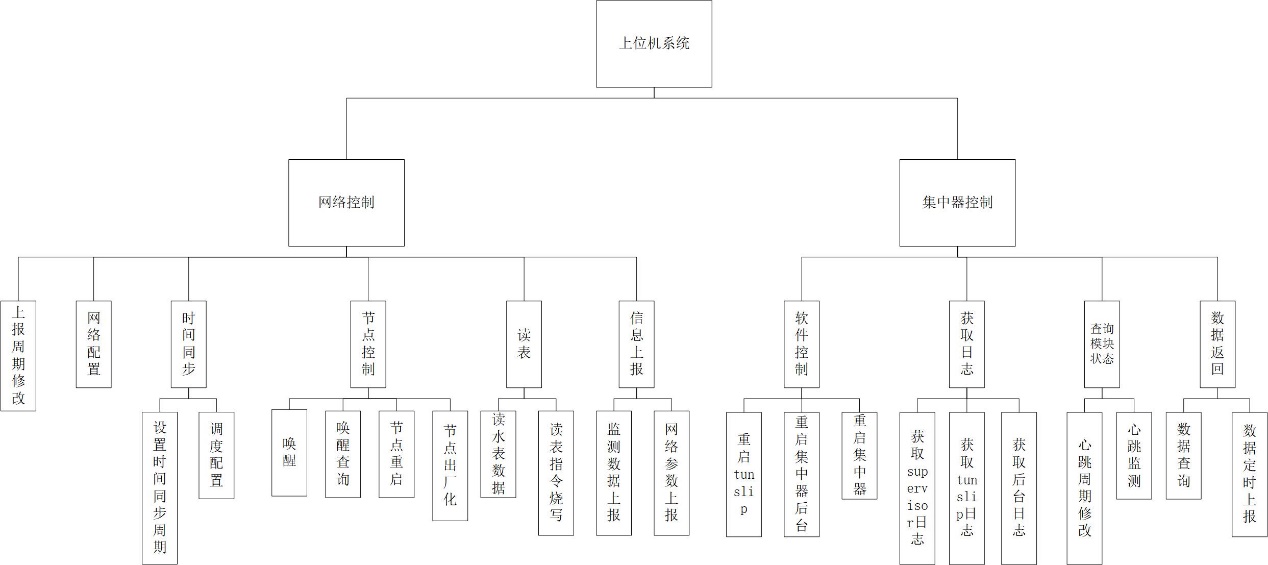
## 上位机简介

上位机使用python语言实现，具体分为发送到网络的数据和发送到集中器的数据。上位机系统是6LoWPAN无线传感器网络中，用于远程控制传感器网络和集中器的系统。该系统设计并实现固定的指令格式，利用GPRS无线网络通讯技术，与集中器后台通信，获取集中器收集的网络信息或者通过集中器后台下发指令到网络，从而达到对整个网络进行监测和管理的目的。

上位机实现的指令分为两种。一种发送到网络，另外一种发送到集中器。

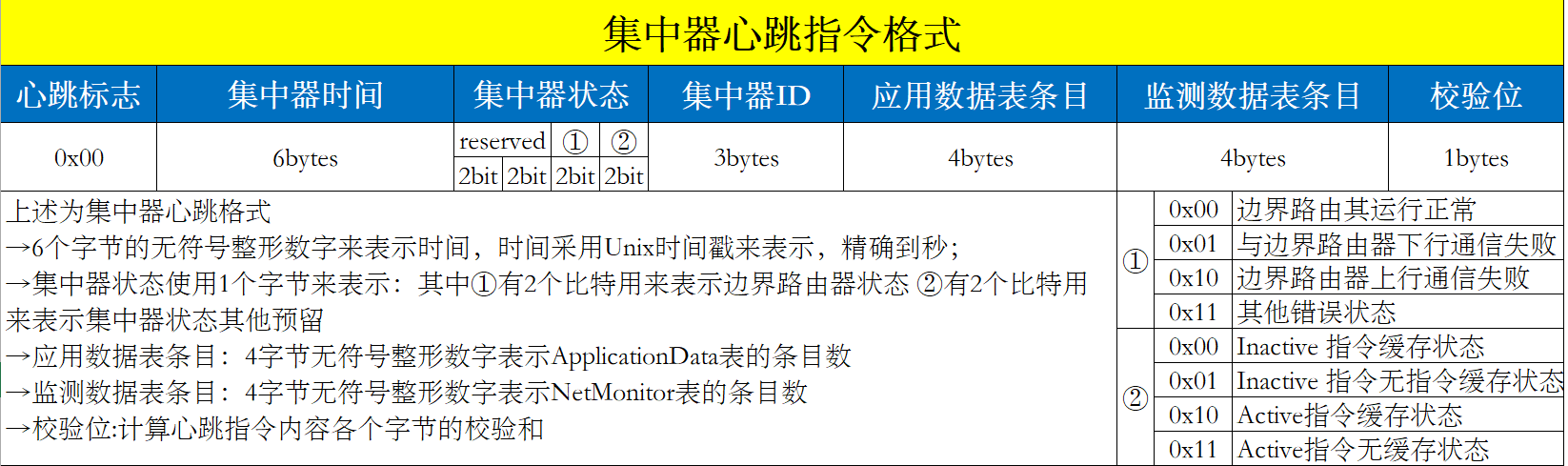
## 上位机功能划分



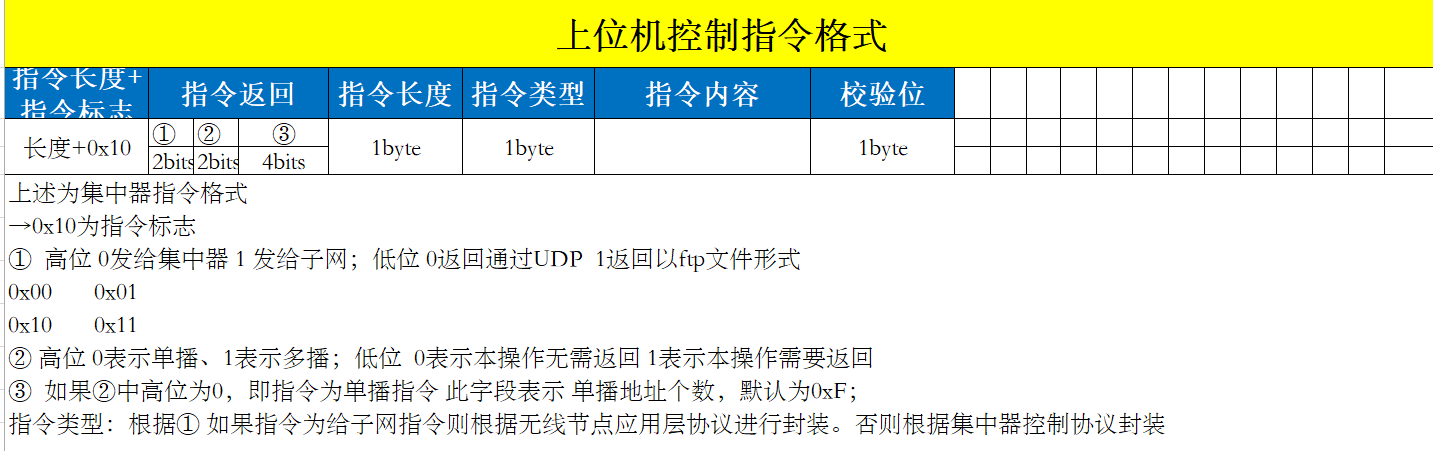
## 上位机指令格式

格式分两种，一种是心跳的格式，另一种是指令的格式

心跳格式：



目前校验位并未使用，返回的消息无法直接查看应用数据表条目和监测数据表条目。由于上位机只能开启一个终端，所以获取心跳程序在下发指令的时候不能同时运行，主要的作用还是获取集中器地址。







## 上位机实现原理

上位机是通过python程序实现的socket编程，通过GPRS与集中器通信，都是一些基本的逻辑。主要是集中器后台对指令的处理略有难度。

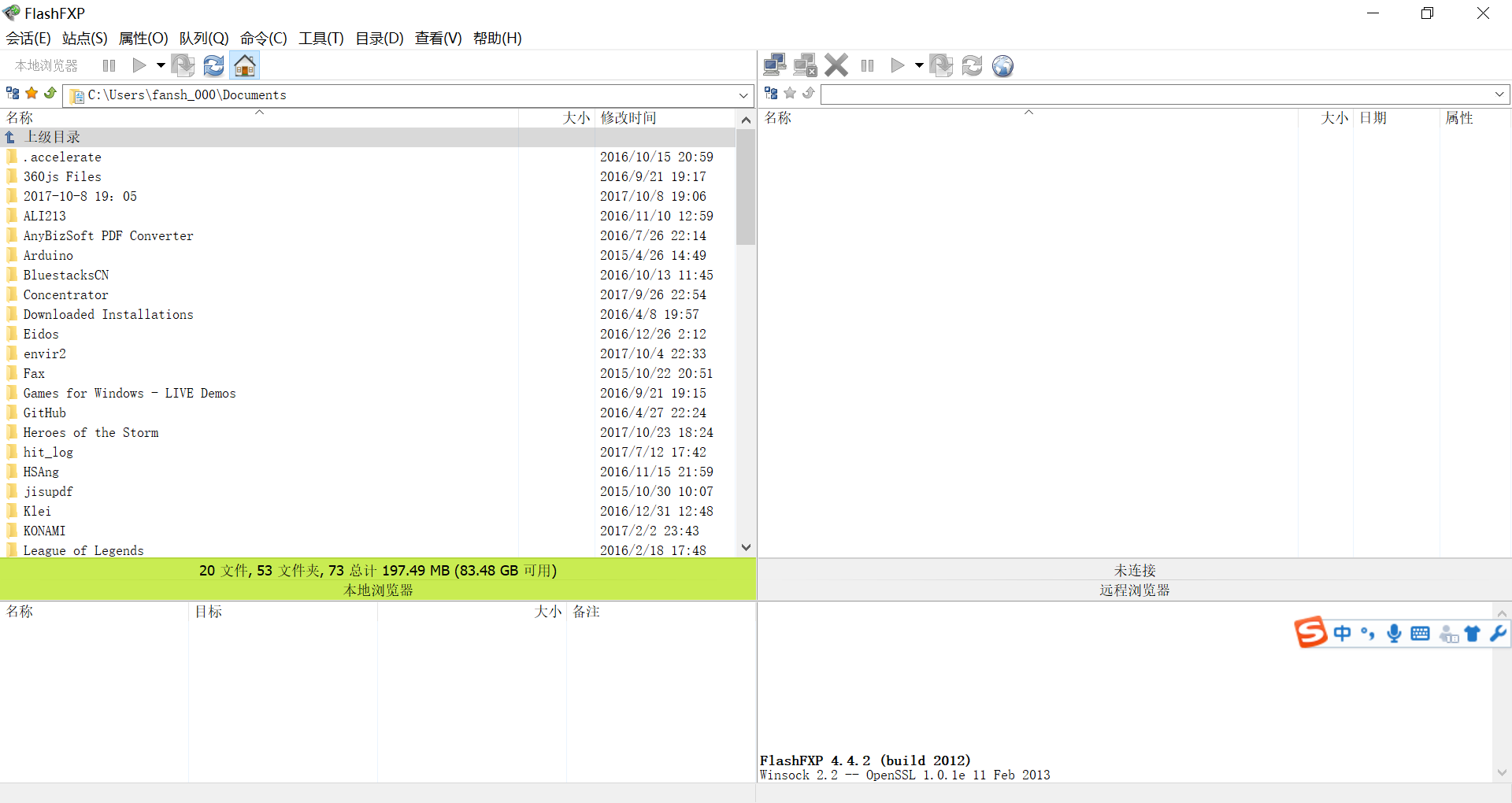
## 上位机使用方法及技巧

在上位机上使用vim查看文件相对麻烦，推荐使用flashfxp软件连接上位机，从上位机下载文件到本机再进行查看。

Flashfxp软件使用方法：

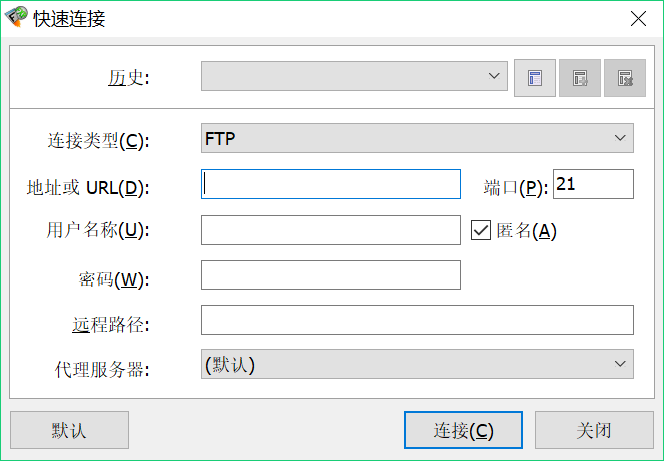
软件下载：链接：http://pan.baidu.com/s/1nuNiFg9 密码：gbxx

软件界面：

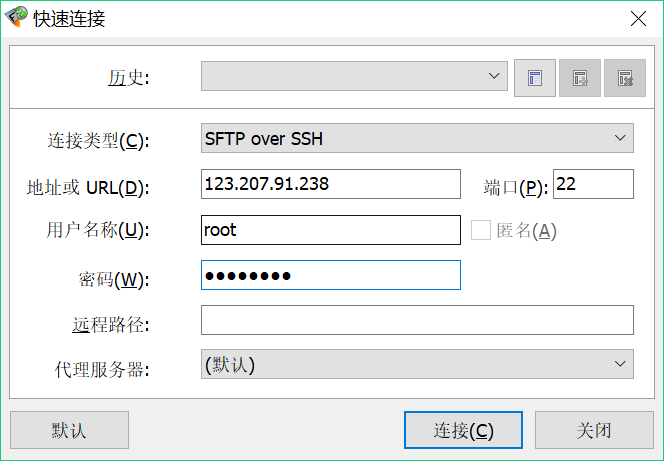


一般左侧为电脑中“文档”目录内容，右侧为将要连接设备的目录。

连接方法：点击左上角会话，第一项：快速连接，之后弹出界面：



连接上位机：

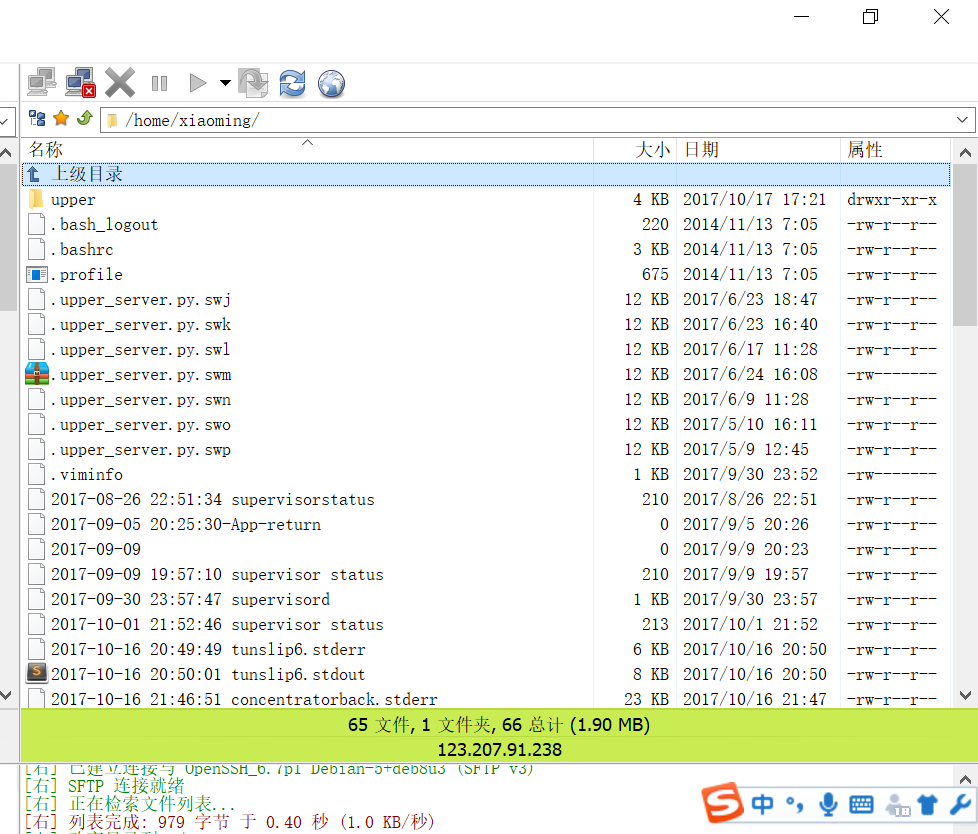


注意，连接类型选择SFTP，地址为123。207.91.238，端口自动选定为22，用户名为root，密码为xiaoming。之后点击连接即可。另外，如果以后申请到了新的云服务器，则有可能需要修改地址，用户名称以及密码。

连接后界面：



出于保险考虑，尽量每次都选择接受一次。



然后去找到home/xiaoming。就找到了存储文件的目录。一般的返回文件的文件名中都包含的改文件发送的时间，根据发送的时间可以判断对应的哪个指令下发得到的。

主文件夹下是返回的消息或者数据，主文件夹下的upper文件夹存放的是上位机程序文件。右键左侧或者右侧的文件可以传输，查看等，拷贝文件也可以通过该软件进行。如果连接中断，可以点击右侧界面上方的小电脑进行快速选择重连。

在主文件夹下：

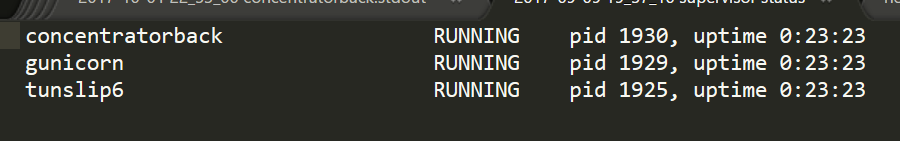
concentratorback.stdout:集中器后台输出日志文件

concentratorback.stderr：集中器后台报错日志文件

tunslip6.stdout：tunslip输出

tunslip6.stderr：tunslip报错

supervisorstatus：supervisor状态报告，里面有concentratorback，tunslip6以及gunicorn的运行时间等报告，当三个都显示running的时候表示运行正常。如下图所示



upper/address：存放的是上一次通过运行接收心跳指令获取的集中器的地址。

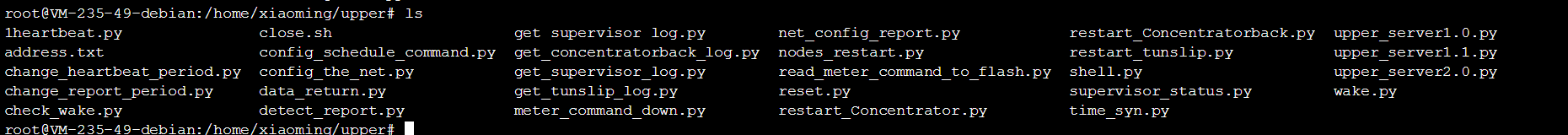
return\_message：集中器后台返回的指令下发成功消息，里面包含了当前时间，下发指令的指令码。

wait\_time.txt返回上次下发指令需要等待多久能够下发，单位是秒，格式为：当前时间，指令码，等待时间。

文件名以app或topo结尾的文件：定时上报的数据，文件名中包含的时间就是获取的时间，里面是字符串形式的数据。

topo4：通过指令返回的数据。

在upper文件夹下：



1heartbeat.py:获取集中器后台心跳，主要是获得集中器的地址，集中器的地址和端口号都保存在address.txt文件中，每次发送指令的时候，都是先从address.txt文件中获取。

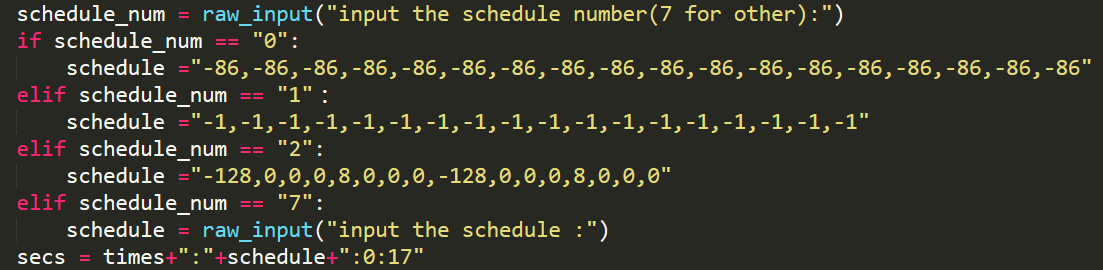
close.sh：关机文件

get\_config\_report.py获取配置信息，无需输入内容

restart\_Concentratorback.py重启集中器后台，无需输入内容

address.txt存放集中器的地址和端口号

config\_schedule\_command.py下发调度指令，如图，输入不同的数字会设置不同的调度，输入数字7的时候可以自己配置调度，按照下图的格式输入即可。



get\_concentratorback\_log.py获取集中器后台的日志，会返回stdout和stderr两个文件，需要输入想要的日志文件的行数。

nodes\_restart.py节点重启指令

restart\_tunslip.py重启tunslip

change\_heartbeat\_period.py修改心跳周期，需要输入心跳间隔

config\_the\_net.py网络配置指令

get\_supervisor\_log.py获取supervisor日志，

read\_meter\_command\_to\_flash.py读表指令写flash

shell.py重新连接网络

change\_report\_period.py修改数据上报周期，单位为天

data\_return.py数据返回，返回类型为.db可以直接放入本机的集中器前台进行查看，需要输入想看数据的起始时间，集中器会返回从输入时间到当前时间的去哪不数据。格式为2017-10-26 21:48:00。

get\_tunslip\_log.py获取tunslip日志，需要输入所需要的行数

resart.py重启集中器指令

supervisor\_status.py获取supervisor中三个软件的状态

wake.py:唤醒指令

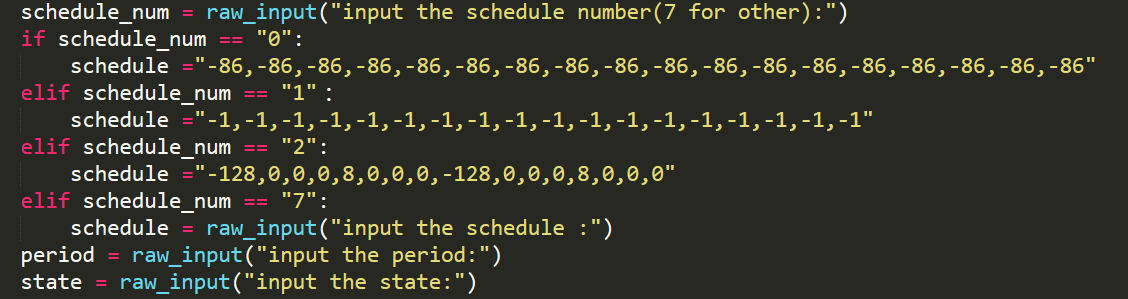
check\_wake.py:唤醒查询指令

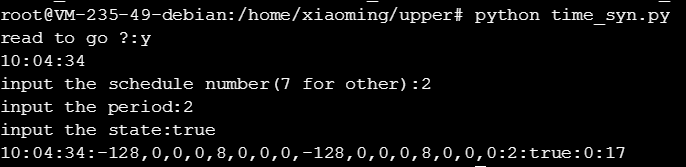
detect\_report.py:监测上报指令

meter\_command\_down.py读表指令下发指令

restart\_Concentrator.py重启集中器前台指令

time\_syn.py:时间同步指令





## 返回消息查看

返回消息分两种，一种是指令等待时间，另一种是指令下发成功消息

发送到网络的指令可以在任何时候下发，集中器后台会计算距离指令下发的时间，将等待的时间返回到上位机，并等待相应的时间再下发。指令下发到根节点后，会返回下发成功消息到上位机。

文件传输通过FTP形式，所有信息通过文件的形式传输。等待时间在文件wait\_time.txt中，格式为：；下发成功信息在文件return\_message中，格式为：。当需要返回消息时候，集中器后台会生成一个同样名称的文件，然后通过FTP上传到上位机，覆盖上位机中的同名称的文件，通过查看文件中信息的时间和对应指令的编号，即可了解信息是否被更新。