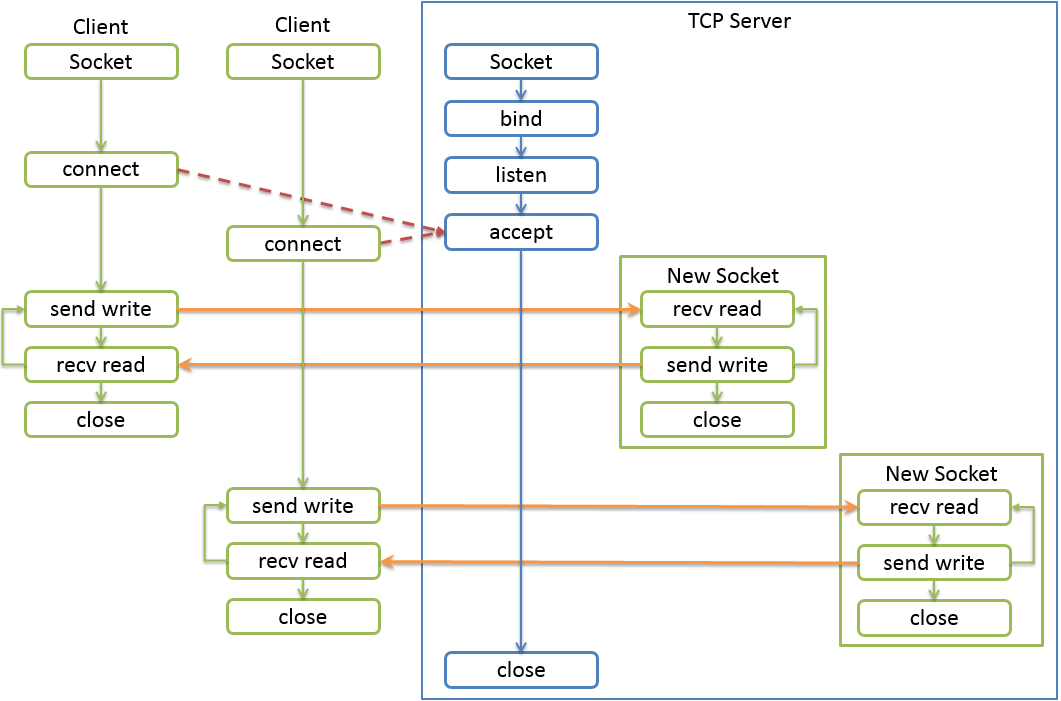
服务器端网络编程



在4G模组与服务器端，采用TCP/IP通信协议来进行通信传输数据。Socket变成，需要两端，一方作为服务器端，另一方就作为客户端。服务器端称为Server，客户端称为Client。

4G模组作为客户端的角色以TCP/IP协议通过透传的方式不断地向服务器端发送采集到的传感器数据。而服务器端，则要建立起监听客户端上传的数据的模式。在客户端向服务器端发送数据之前，服务器端就应该已经做好了开启监听模式的工作，等待客户端的连接并能接收处理之前双方定义好的数据帧格式。

TCP编程

Socket是网络编程的一个抽象的概念。通常我们使用一个Socket表示打开了一个网络链接，而打开一个Socket需要知道目标计算机的IP地址和端口号，再指定通信协议类型就可以了。

1. Socket介绍

Socket套接字，Python中提供了socket.py标准库，非常底层的接口库用来进行网络编程。Socket是一种通用的网络编程接口，和网络层次没有一一对应的关系。在调用一个socket()函数中，就涉及到了两个参数。

协议族，AF表示Address Family,用于socket()函数的第一个参数。

|  |  |
| --- | --- |
| 名称 | 含义 |
| AF\_INET | IPV4 |
| AF\_INET6 | IPV6 |
| AF\_UNIX | Unix Domain Socket,windows没有 |

Socket类型

|  |  |
| --- | --- |
| 名称 | 含义 |
| SOCK\_STREAM | 面向连接的流套接字。默认值，TCP协议 |
| SOCK\_DGRAM | 无连接的数据报文套接字。UDP协议 |

1. Socket启动过程

首先，创建一个Socket对象，绑定服务器的IP地址和唯一的未被其他程序占用的端口Port。调用bind()方法，IPV4地址为一个二元组(‘IP地址字符串’，Port)。绑定好之后，服务器就开始监听，将在指定的IP的端口上监听。调用listen（）方法。之后，若有客户端通过TCP/IP协议连接服务器端，进行通信。服务器端通过调用socket.accept()方法获取用于传送数据的Socket对象。Accept方法阻塞等待客户端建立连接，返回一个新的Socket对象和客户端地址的二元组地址是远程客户端的地址，IPV4中它是一个二元组（clientaddr,port）。之后，就可以调用recv()方法来接收数据了。在测试中，可以知道accept和recv方法都是阻塞的。这样一来，主线程经常被阻塞住而不能工作了。所以，可以想到这里就需要引入多线程来处理服务器socket的监听和接收。

1. SocketServer模块编程

Socket编程过于底层，编程虽然方法步骤固定简单，但是很难写出功能贴近实际生产使用的代码，所以Python语言就对socket底层API进行了封装。在Python库中就有了socketserver模块，它是网络服务编程框架，便于企业级快速开发。

BaseServer

UnixDatagramServer

UnixStreamServer

UDPServer

TCPServer

SocketServer简化了网络服务器的编写。它有4个同步类：TCPServer,UDPServer,UnixSteamServer,UnixDatagramServer。

Socketserver模块提供了Socketserver.BaseServer(server\_address,RequestHandlerClass)编程接口，需要提供服务器绑定的地址信息，和用于处理请求的RequestHandlerClass类。RequestHandlerClass类必须是BaseRequestHandler类的子类。

BaseRequestHandler类是和用户连接的用户请求处理类的基类，定义为BaseRequestHandler(request,client\_address,server)。服务器Server实例接收到用户请求后，最后会实例化这个类。它被初始化时，就送入了3个构造函数：request,client\_address,server本身。以后就可以在BaseRequestHandler类的实例上使用以下属性：self.request是和客户端的连接的socket对象，self.server是TCPServer实例本身，self.client\_address是客户端地址。这个类在初始化的时候，就会依次调用3个方法。定义的子类可以覆盖掉这些方法。

#BaseRequestHandler要自定义的子类覆盖的方法

**def** \_\_init\_\_(self, request, client\_address, server):  
 self.request = request  
 self.client\_address = client\_address  
 self.server = server  
 self.setup()  
 **try**:  
 self.handle()  
 **finally**:  
 self.finish()  
  
**def setup**(self): #每一个连接初始化

**pass  
  
def handle**(self): #每一次请求处理  
 **pass  
  
def finish**(self): #每一个连接清理  
 **pass**

在其中的handler()方法相当于socket的recv()方法。每个不同的连接上的请求过来之后，生成这个连接的socket对象即self.request，客户端地址是self.client\_address。

由于上面说到了BaseRequestHandler是会依次调用上面的三个方法的，当调用到finish()方法时，可以看到连接之后就立即断开了。所以，在handler方法中加入**while not** self.event.is\_set():在handler里面循环。

server = socketserver.ThreadingTCPServer(('0.0.0.0', 9999), MyHandler)。这里用的是ThreadingTCPServer。ThreadingTCPServer是多线程异步的，可以同时处理多个连接。TCPServer是同步的，一个连接处理完了，即一个连接的handler方法执行完了，才能处理另一个连接，且只有主线程。同步库，多个客户端都可以accept，但是相当于进入了队列中等待，先来先处理的原则，来处理recv,处理完第一个socket的recv后，继续之后的socket操作recv方法，尽管其他socket在等待，但是数据都是被缓存起来的。

在网络通信中，客户端会有断开连接，造成网络异常断开的情况发生。如果连接的线程中handler方法中抛出异常，，例如客户端主动断开导致的异常，线程崩溃。应添加执行self.clients的pop方法，捕获客户端的各种异常。通过测试打印可以看到，客户端主动断开，会导致recv方法立即返回一个空bytes，并没有同时抛出异常。当循环回到recv这一句的时候就会抛出异常。所以，下列语句

**if not** data:  
 print("Broken pipe")  
 **break**

可以通过判断data数据是否为空来判断客户端是否断开。若断开，则break退出循环。

所以，使用SocketServer创建服务器的过程如下。

3.1 从BaseRequestHandler类派生出子类，并覆盖其handler()方法来创建请求处理程序类，此方法将处理传入请求。

3.2 实例化一个服务器类，传参服务器的地址和请求处理类。

3.3 调用服务器实例的handler\_request()或serve\_forever()方法。

3.4 调用server\_close()关闭套接字，完成网络通信。

数据库

1.数据库

按照数据结构来组织、存储、管理数据的仓库。

计算机的发生是为了做科学计算的，而科学计算需要大量的数据输入和输出。早期，可以使用打孔卡片的孔、灯泡的亮灭来表示数据输入、输出。后来，数据可以存储在磁带上，顺序的读取、写入磁带。1956年IBM发明了磁带驱动器这个革命性产品，支持随机访问。随着信息化时代的到来，有了硬件存储技术的发展，有大量的数据需要存储和管理，数据库管理系统DBMS就诞生了。不管使用什么存储介质，数据库的数据模型才是其核心和基础。

分类：按照数据模型分类：网状数据库、层次数据库、关系型数据库。

这里，我们需要着重了解关系数据库。使用行、列组成的二维表来组织数据和关系，表中行（记录）即可以描述数据实体，也可以描述实体间关系。关系模型比网状模型、层次模型更简单，不需要关系数存储的物理细节，专心于数据的逻辑构建，而且关系模型有论文的严格的数学理论基础支撑。

1970年，IBM的研究员E.F.Codd发表了名为“A Relational Model of Data for Large Shared Data Banks”的论文，提出了关系模型的概念，奠定了关系模型的理论基础。关系模型，有严格的数学基础，抽象级别较高，简单清晰，便于理解和使用。经过几十年的发展，关系数据库百花齐放，技术日臻成熟和完善。基于关系模型构建的数据库系统成为RDBMS(Relational DataBase System)。IBM DB2、Oracle的Oracle和Mysql、微软的MS SQL。以前的informix、Sybase等。

2.MySQL

MySQL是一种关系型数据库管理软件，支持网络访问，默认服务端口3306。MySQL使用mysql协议。

3.安装

使用yum安装rpm包，推荐安装MariaDB。在服务器上安装好MySQL，充当数据库服务器。yum install mariadb安装好数据库之后，service mysql\_start开启服务器程序。

1. SQL语句

SQL是结构化查询语言Structured Query Language。1987年被ISO组织标准化。所有主流的关系型数据库多支持SQL,NoSQL也有很大一部分支持SQL。

SQL语句分为：DDL数据定义语言，负责数据库定义、数据库对象定义，由CREATE、ALTER与DROP三个语法所组成。DML数据操作语言，负责对数据库对象的操作，CRUD增删改查。DCL数据控制语言，负责数据库权限访问控制，由GRANT和REVOKE两个指令组成。TCL事务控制语言，负责处理ACID事务，支持commit、rollback指令。SQL语句大小写不敏感。SQL语句末尾应该使用分号结束。

1. 数据库工具

Navicat Lite是一个多连接数据库开发工具。是一套快速、可靠的资料库管理工具。它是可以用图形界面来连接，操作自己的数据库。