

Socket 网络编程 -- UDP 虚谷与掌控

--- 2019.12.15 yywudao

一、网络通信

人生苦短，我用Python。

虚谷号和掌控板都可以通过 Wi-Fi 连接到网络，有网络的地方，就有数据，有数据的地方，皆可互相通信。

1. 服务器、客户端

网络中的两台设备，提供资源的一方称为服务器 (server)，取用或接收资源的一方称为客户端 (client)。

2. IP 地址

网络上的设备都有一个地址，这个地址称为 IP 地址 (*Internet Protocol address*)，即网际协议地址。

- IPv4 (*Internet Protocol version 4*)，网际协议第4版，IPv4 的地址长度32位元，全球的 IPv4 地址已耗尽。

xxx . xxx . xxx . xxx

- IPv6 (*Internet Protocol version 6*)，网际协议第6版，IPv6 的地址长度128位元。

xxxx : xxxx : xxxx : xxxx

- LAN (*Local Area Network*)，局域网。俗称内网，内网的 IP 地址，通常以 192.168 开头。

比如电脑、虚谷号连接到家里路由器分配到的 IP 地址：192.168.3.xxx、192.168.1.xxx

掌控板开启 AP 模式下的 IP 地址：192.168.4.1

3. Port 端口号

如果把网络通信比作收发信件，设备的 IP 地址相当于收件地址。设备上不同的程序处理不同的服务，端口号就是用来区分这一个个服务的收件人。

- 范围： $2^{16} = 65536$ (0~~65535)
- 1 ~ 1023 的端口号有特定的意义，知名端口 (Well-known ports)

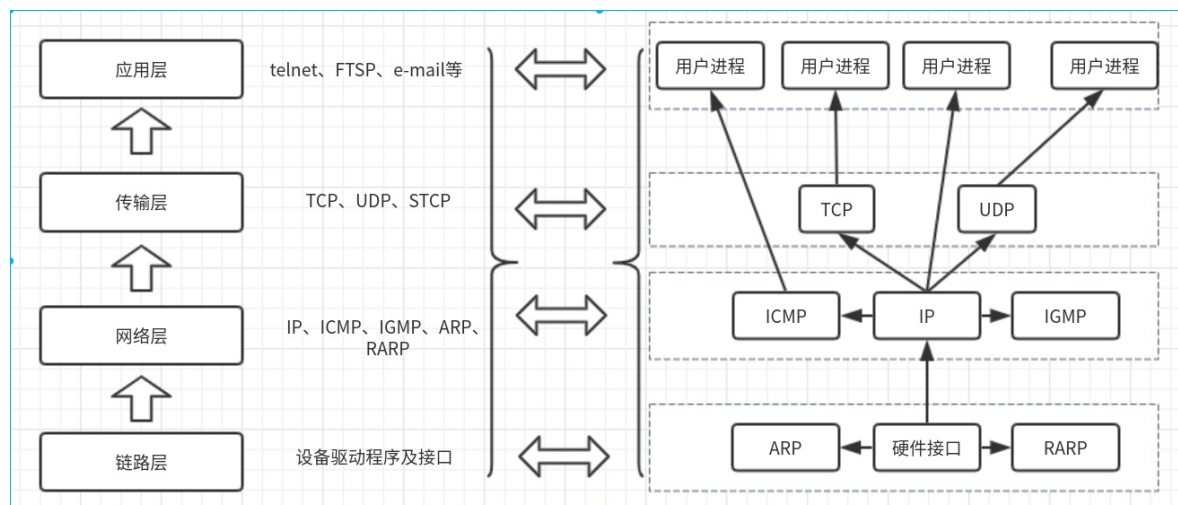
名称	端口号	说明
HTTP	80	超文本传输协议
HTTPS	443	超文本传输安全协议
FTP	21	文件传输协议
SSH	22	
TELNET	23	用于远程登录
SMTP	25	用于发送电子邮件

- 1024 ~ 65535，动态端口。自定义的可用端口

4. TCP/IP

网络上数据通信要遵守相应的协议，**TCP/IP** (*Transmission Control Protocol/Internet Protocol*)，即传输控制协议/网际协议，是常见的网络通信协议。

对于 **TCP/IP**，有四层模型之说：



名称	内容
应用层	应用程序的相互沟通
传输层	TCP、UDP、STCP等协议。
网络层	IP、ARP、RARP等协议。建立封包，设定IP地址和传输路径
链路层	由驱动程序负责，解析和还原网络上的0/1的信号

5. TCP、UDP

TCP (*Transmission Control Protocol*)，即传输控制协议。

UDP (*User Datagram Protocol*)，即用户数据报协议。

- TCP 相当于挂号信，UDP 相当于普通信件
- 采用 TCP 传输，设备会在收到封包 (数据) 时，回复讯息给发送端，确认资料接收无误。若发送端在一段时间内没有收到回复讯息，它会认为封包在传输过程中丢失了，会重发一次该封包。
- 采用 UDP 传输，不会确认封包是否到达目的地。不与对方建立连接，直接把数据报发送给对方。所以称，UDP 是面向非连接的协议。

二、Socket 网络编程 UDP

1. Socket 套接字

socket 套接字。使用类似于文件：创建文件、读写文件、关闭文件。创建套接字、读写套接字、关闭套接字。

- 创建 socket (套接字)

```
import socket # 导入 socket 模块
socket.socket(AddressFamily, Type)
```

函数 `socket.socket` 创建一个 socket，该函数有两个参数

- `AddressFamily`：(协议族)

AF_INET：用于 Internet 进程间通信，IPv4

AF_INET6：用于 Internet 进程间通信，IPv6

AF_UNIX：用于同一台设备进程间通信

◦ Type：（套接字类型）

SOCK_STREAM：流式套接字，用于 TCP

SOCK_DGRAM：数据报套接字，用于 UDP

- TCP socket (TCP 套接字)

```
import socket

# 创建 TCP 套接字
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

# ...使用套接字收发数据

# 关闭套接字
s.close()
```

- UDP socket (UDP 套接字)

```
import socket

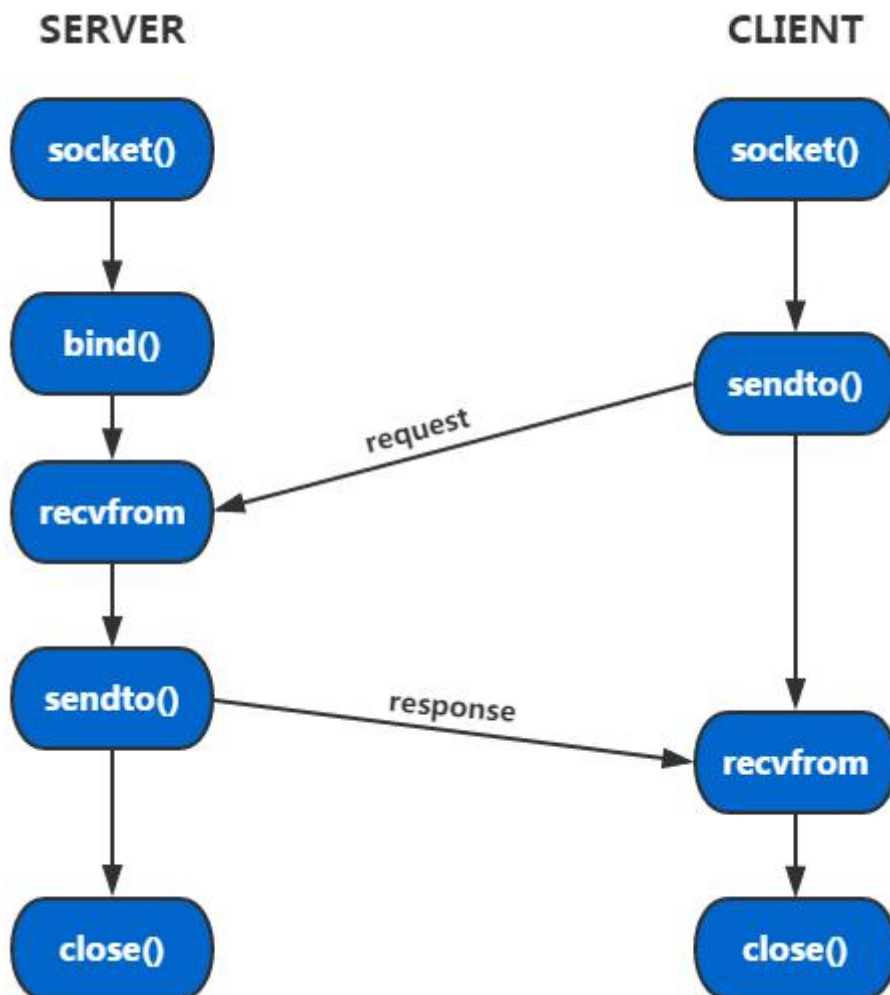
# 创建 UDP 套接字
s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

# ...使用套接字收发数据

# 关闭套接字
s.close()
```

2. UDP 程序流程

UDP套接字使用流程图：



- 流程总览

```
import socket # 导入 socket 模块

s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM) # 创建 UDP Socket 对象
s.bind((Host, Port)) # 绑定 IP 地址和端口

s.sendto(string, ("IP address", Port)) # 发送数据
s.recvfrom(1024) # 接收 1024 字节的数据

s.close() # 关闭套接字
```

1.) UDP 服务器端配置

- a 创建 UDP 套接字

```
import socket # 导入 socket 模块
s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM) # 创建 UDP Socket 对象
```

- b 以元组的形式将套接字绑定到地址

```
s.bind((Host, Port)) # 绑定本机地址和端口(端口号需大于1023)
```

```
s.bind(('', 10000))
s.bind(('localhost', 10000))
s.bind(('127.0.0.1', 10000))
```

- c 接收数据

```
s.recvfrom(bufsize) # 接收数据, bufsize 为接收的最大数据量
```

```
r = s.recvfrom(1024) # 接收1024字节的数据量, 类型为 tuple
data = r[0] # 接收信息的数据内容, 类型为 bytes
addr = r[1] # 信息发送方的地址和端口, 类型为 tuple

data, addr = s.recvfrom(124)
data = data.decode('utf-8') # 处理数据内容, 进行解码
```

- d 回复数据

```
s.sendto(b'something', (IP address, Port))
# 两个参数, 发送的数据类型为 bytes, 发送的地址和端口

data = xxxxx # 定义发送的数据内容
s.sendto(data.encode('utf-8'), addr) # 数据内容进行编码, addr 为之前接收数据的地址和端口
```

- e 关闭套接字

```
s.close()
```

2.) UDP 客户端配置

- a 创建套接字

```
import socket # 导入 socket 模块
s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM) # 创建 UDP Socket 对象
```

- b 发送数据

```
s.sendto(b'something', (IP address, Port))
# 两个参数, 发送的数据类型为 bytes, 服务器端的地址和端口

data = xxxxx # 定义发送的数据内容
s.sendto(data.encode('utf-8'), (IP address, Port)) # 数据内容进行编码, 服务器端的地址和端口
```

- c 接收数据

```
s.recvfrom(bufsize) # 接收数据, bufsize 为接收的最大数据量
```

```

r = s.recvfrom(1024) # 接收1024字节的数据量，类型为 tuple
data = r[0] # 接收信息的数据内容，类型为 bytes
addr = r[1] # 信息发送方的地址和端口，类型为 tuple

data, addr = s.recvfrom(124)
data = data.decode('utf-8') # 处理数据内容，进行解码

```

- d 关闭套接字

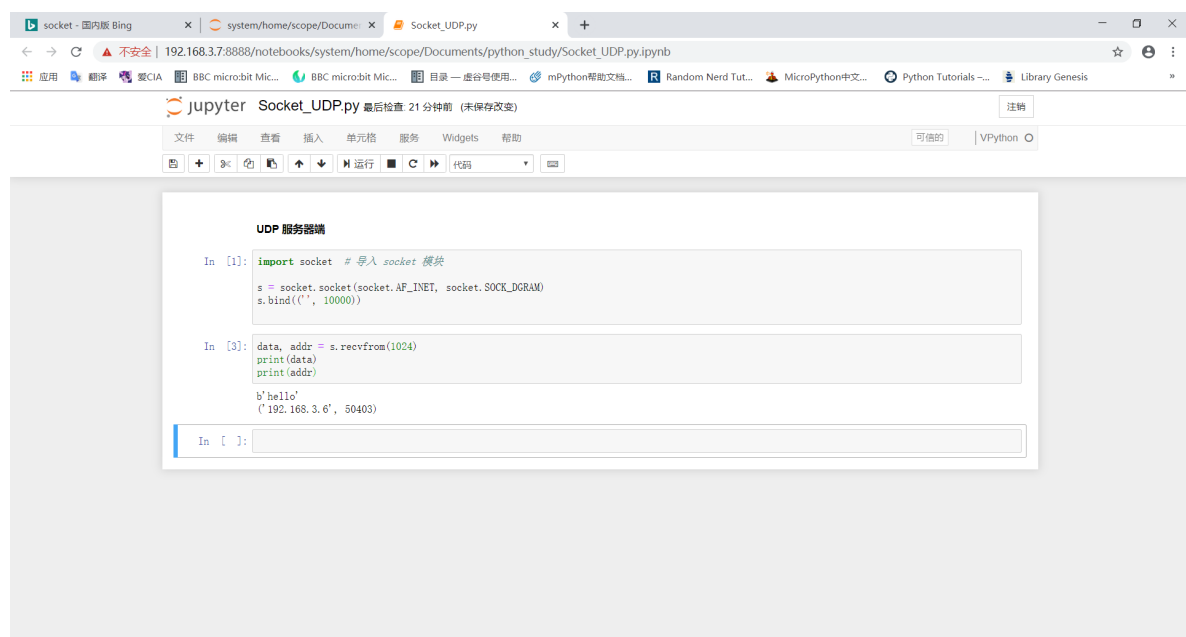
```
s.close()
```

三、虚谷号 Jupyter 应用示例

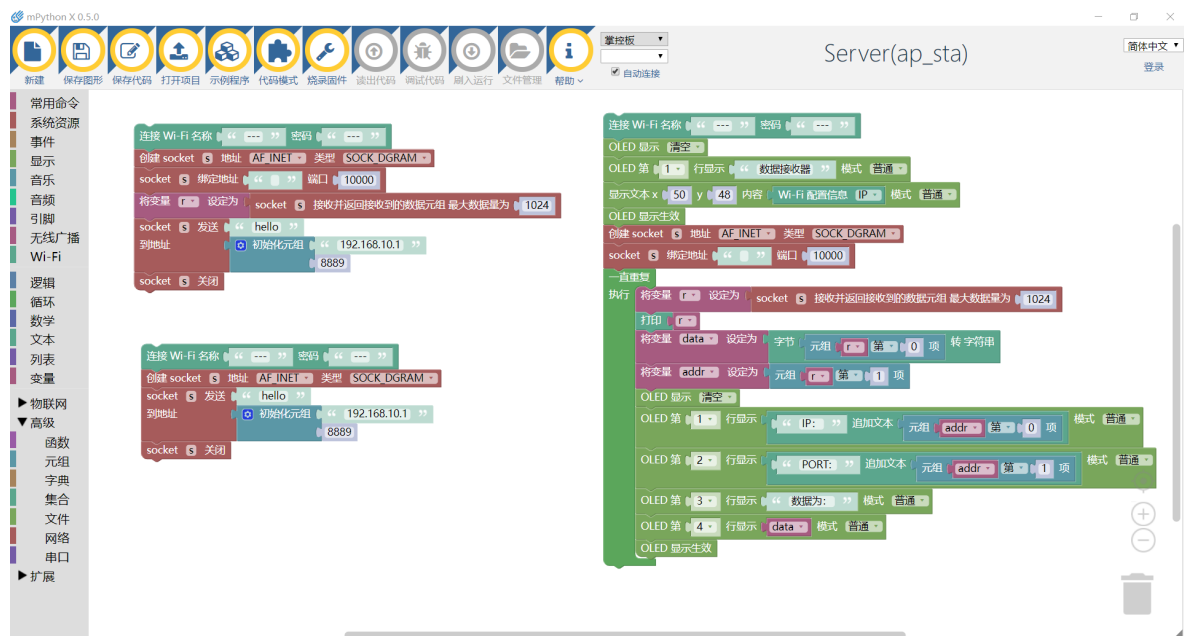
1. Thonny 客户端程序



2. Jupyter 服务器端程序



四、掌控板示例 (mPython X)



- 详见 DF 论坛：[项目分享]掌控板：UDP的那些事儿（二）-- 来自“远方的信息”

参考资料

- 《TCP/IP详解.卷1：协议》阅读总结
<https://zhuanlan.zhihu.com/p/86503961>
- 《Linux编程之UDP SOCKET全攻略》
<https://www.cnblogs.com/skyfsm/p/6287787.html>
- 《TCP/IP SOCKET 编程详解》
<https://www.cnblogs.com/wdy0505/p/10100285.html>
- 《TCP和UDP的区别 (socket)》
<https://www.cnblogs.com/LUO77/p/5801977.html>
- 《超图解 Python 物联网实作入门》赵英杰著
- mPython 在线文档：10.4. 套接字-UDP
<https://mpython.readthedocs.io/zh/master/tutorials/basics/network/udp.html>
- [python教程] socket 网络编程 (TCP/UDP)
<https://www.bilibili.com/video/av56893736?p=1>
- [入门教程]掌控板：UDP的那些事儿（一）-- 漫谈通“信”
<http://mc.dfrobot.com.cn/thread-300194-1-1.html>
- [项目分享]掌控板：UDP的那些事儿（二）-- 来自“远方的信息”
<http://mc.dfrobot.com.cn/thread-300199-1-1.html>
-