



# socket-TCP/UDP 模块

版本：1.0

日期：2026-01-12

状态：受控/临时文件

文件保密级别：(请勾选 ■ )

绝密

保密

公开

# 文件管控表

上海移远通信技术股份有限公司（以下简称“移远通信”）始终以为客户提供最及时、最全面的服务为宗旨。如需任何帮助，请随时联系我司上海总部，联系方式如下：

上海移远通信技术股份有限公司

上海市闵行区田林路 1016 号科技绿洲 3 期（B 区）5 号楼 邮编：200233

电话：+86 21 5108 6236 邮箱：[info@quectel.com](mailto:info@quectel.com)

或联系我司当地办事处，详情请登录：<http://www.quectel.com/cn/support/sales.htm>。

如需技术支持或反馈我司技术文档中的问题，请随时登录网址：

<http://www.quectel.com/cn/support/technical.htm> 或发送邮件至：[support@quectel.com](mailto:support@quectel.com)。

## 前言

移远通信提供该文档内容以支持客户的产品设计。客户须按照文档中提供的规范、参数来设计产品。同时，您理解并同意，移远通信提供的参考设计仅作为示例。您同意在设计您目标产品时使用您独立的分析、评估和判断。在使用本文档所指导的任何硬软件或服务之前，请仔细阅读本声明。您在此承认并同意，尽管移远通信采取了商业范围内的合理努力来提供尽可能好的体验，但本文档和其所涉及服务是在“可用”基础上提供给您的。移远通信可在未事先通知的情况下，自行决定随时增加、修改或重述本文档。

## 使用和披露限制

### 许可协议

除非移远通信特别授权，否则我司所提供硬软件、材料和文档的接收方须对接收的内容保密，不得将其用于除本项目的实施与开展以外的任何其他目的。

### 版权声明

移远通信产品和本协议项下的第三方产品可能包含受移远通信或第三方材料、硬软件和文档版权保护的相关资料。除非事先得到书面同意，否则您不得获取、使用、向第三方披露我司所提供的文档和信息，或对此类受版权保护的资料进行复制、转载、抄袭、出版、展示、翻译、分发、合并、修改，或创造其衍生作品。移远通信或第三方对受版权保护的资料拥有专有权，不授予或转让任何专利、版权、商标或服务商标权的许可。为避免歧义，除了正常的非独家、免版税的产品使用许可，任何形式的购买都不可被视为授予许可。对于任何违反保密义务、未经授权使用或以其他非法形式恶意使用所述文档和信息的违法侵权行为，移远通信有权追究法律责任。

### 商标

除另行规定，本文档中的任何内容均不授予在广告、宣传或其他方面使用移远通信或第三方的任何商标、商号及名称，或其缩略语，或其仿冒品的权利。

### 第三方权利

您理解本文档可能涉及一个或多个属于第三方的硬软件和文档（“第三方材料”）。您对此类第三方材料的使用应受本文档的所有限制和义务约束。

移远通信针对第三方材料不做任何明示或暗示的保证或陈述，包括但不限于任何暗示或法定的适销性或特定用途的适用性、平静受益权、系统集成、信息准确性以及与许可技术或被许可人使用许可技术相关的不侵犯任何第三方知识产权的保证。本协议中的任何内容都不构成移远通信对任何移远通信产品或任何其他硬软件、设备、工具、信息或产品的开发、增强、修改、分销、营销、销售、提供销售或以其他方式维持生产的陈述或保证。此外，移远通信免除因交易过程、使用或贸易而产生的任何和所有保证。

## 隐私声明

为实现移远通信产品功能，特定设备数据将会上传至移远通信或第三方服务器（包括运营商、芯片供应商或您指定的服务器）。移远通信严格遵守相关法律法规，仅为实现产品功能之目的或在适用法律允许的情况下保留、使用、披露或以其他方式处理相关数据。当您与第三方进行数据交互前，请自行了解其隐私保护和数据安全政策。

## 免责声明

- 1) 移远通信不承担任何因未能遵守有关操作或设计规范而造成损害的责任。
- 2) 移远通信不承担因本文档中的任何因不准确、遗漏、或使用本文档中的信息而产生的任何责任。
- 3) 移远通信尽力确保开发中功能的完整性、准确性、及时性，但不排除上述功能错误或遗漏的可能。除非另有协议规定，否则移远通信对开发中功能的使用不做任何暗示或法定的保证。在适用法律允许的最大范围内，移远通信不对任何因使用开发中功能而遭受的损害承担责任，无论此类损害是否可以预见。
- 4) 移远通信对第三方网站及第三方资源的信息、内容、广告、商业报价、产品、服务和材料的可访问性、安全性、准确性、可用性、合法性和完整性不承担任何法律责任。

版权所有 ©上海移远通信技术股份有限公司 2024，保留一切权利。

**Copyright © Quectel Wireless Solutions Co., Ltd. 2024.**

## 目录

<b>1 模块概述</b>	5
<b>2 构造函数</b>	5
2.1. usocket.socket	5
2.1.1. 接口概述	5
2.1.2. 函数原型	5
2.1.3. 参数描述	5
2.1.4. 返回值描述	5
2.1.5. 示例	6
2.2. usocket.getaddrinfo	6
2.2.1. 接口概述	6
2.2.2. 函数原型	6
2.2.3. 参数描述	6
2.2.4. 返回值描述	6
2.2.5. 示例	6
<b>3 socket 方法</b>	7
3.1. bind	7
3.1.1. 接口概述	7
3.1.2. 函数原型	7
3.1.3. 参数描述	7
3.1.4. 返回值描述	7
3.1.5. 示例	7
3.2. listen	7
3.2.1. 接口概述	7
3.2.2. 函数原型	7
3.2.3. 参数描述	7
3.2.4. 返回值描述	7
3.2.5. 示例	8
3.3. accept	8
3.3.1. 接口概述	8
3.3.2. 函数原型	8
3.3.3. 参数描述	8
3.3.4. 返回值描述	8
3.3.5. 示例	8
3.4. connect	8
3.4.1. 接口概述	8
3.4.2. 函数原型	8
3.4.3. 参数描述	8
3.4.4. 返回值描述	9
3.4.5. 示例	9
3.5. recv	9
3.5.1. 接口概述	9

3.5.2. 函数原型 .....	9
3.5.3. 参数描述 .....	9
3.5.4. 返回值描述 .....	9
3.5.5. 示例 .....	9
3.6. recvfrom .....	9
3.6.1. 接口概述 .....	9
3.6.2. 函数原型 .....	9
3.6.3. 参数描述 .....	10
3.6.4. 返回值描述 .....	10
3.6.5. 示例 .....	10
3.7. send .....	10
3.7.1. 接口概述 .....	10
3.7.2. 函数原型 .....	10
3.7.3. 参数描述 .....	10
3.7.4. 返回值描述 .....	10
3.7.5. 示例 .....	10
3.8. sendto .....	10
3.8.1. 接口概述 .....	10
3.8.2. 函数原型 .....	11
3.8.3. 参数描述 .....	11
3.8.4. 返回值描述 .....	11
3.8.5. 示例 .....	11
3.9. setblocking .....	11
3.9.1. 接口概述 .....	11
3.9.2. 函数原型 .....	11
3.9.3. 参数描述 .....	11
3.9.4. 返回值描述 .....	11
3.9.5. 示例 .....	11
3.10. settimeout .....	12
3.10.1. 接口概述 .....	12
3.10.2. 函数原型 .....	12
3.10.3. 参数描述 .....	12
3.10.4. 返回值描述 .....	12
3.10.5. 示例 .....	12
3.11. close .....	12
3.11.1. 接口概述 .....	12
3.11.2. 函数原型 .....	12
3.11.3. 参数描述 .....	12
3.11.4. 返回值描述 .....	12
3.11.5. 示例 .....	13

# 1 模块概述

usocket 模块提供对 BSD 套接字接口的访问，支持 TCP/UDP 网络通信。该模块实现了 CPython socket 模块的子集，专为嵌入式系统优化设计。

# 2 构造函数

## 2.1. usocket.socket

### 2.1.1. 接口概述

创建新的套接字对象。

### 2.1.2. 函数原型

```
class usocket.socket(af=AF_INET, type=SOCK_STREAM)
```

### 2.1.3. 参数描述

参数名	是否必填	数据类型	默认值	说明
af	否	int	AF_INET	地址族
type	否	int	SOCK_STREAM	socket 类型

- af 仅支持 AF\_INET

### 2.1.4. 返回值描述

返回套接字对象。

### 2.1.5. 示例

```
import usocket

# 创建基于 TCP 的流式套接字
socket = usocket.socket(usocket.AF_INET, usocket.SOCK_STREAM)

# 创建基于 UDP 的数据报套接字
socket = usocket.socket(usocket.AF_INET, usocket.SOCK_DGRAM)
```

## 2.2. usocket.getaddrinfo

### 2.2.1. 接口概述

DNS 域名解析。

### 2.2.2. 函数原型

```
usocket.getaddrinfo(host, port)
```

### 2.2.3. 参数描述

参数名	是否必填	数据类型	默认值	说明
host	是	string	-	主机域名
port	是	int	-	端口号

### 2.2.4. 返回值描述

返回 5 元组列表，每个元组结构为：(family, type, proto, canonname, sockaddr)。

### 2.2.5. 示例

```
import usocket
result = usocket.getaddrinfo('www.baidu.com', 443)
```

# 3 socket 方法

## 3.1. bind

### 3.1.1. 接口概述

将套接字绑定到指定 address。

### 3.1.2. 函数原型

```
socket.bind(address)
```

### 3.1.3. 参数描述

参数名	是否必填	数据类型	默认值	说明
address	是	tuple	-	包含地址和端口号的元组

### 3.1.4. 返回值描述

无返回值。

### 3.1.5. 示例

```
# 绑定指定 address
socket.bind(("192.168.0.1", 80))
```

## 3.2. listen

### 3.2.1. 接口概述

开启监听客户端连接，可指定最大客户端连接数。

### 3.2.2. 函数原型

```
socket.listen(backlog)
```

### 3.2.3. 参数描述

参数名	是否必填	数据类型	默认值	说明
backlog	是	int	-	接受套接字的最大个数

### 3.2.4. 返回值描述

无返回值。

### 3.2.5. 示例

```
socket.listen(5)
```

## 3.3. accept

### 3.3.1. 接口概述

接受连接请求，返回新的套接字和客户端地址信息。

### 3.3.2. 函数原型

```
socket.accept()
```

### 3.3.3. 参数描述

无参数。

### 3.3.4. 返回值描述

返回元组 (conn, (address, port))。

### 3.3.5. 示例

```
# socket.accept() 返回的是一个二元组（2个元素）
client_socket, client_addr = socket.accept()
```

```
# client_addr 本身就是一个包含（地址，端口）的二元组
ip, port = client_addr
```

```
print(f"客户端连接来自：IP={ip}, 端口={port}")
```

## 3.4. connect

### 3.4.1. 接口概述

连接到指定 address 的服务器。

### 3.4.2. 函数原型

```
socket.connect(address)
```

### 3.4.3. 参数描述

参数名	是否必填	数据类型	默认值	说明
-----	------	------	-----	----

address	是	tuple	-	包含地址和端口号的元组
---------	---	-------	---	-------------

### 3.4.4. 返回值描述

无返回值。

### 3.4.5. 示例

```
# 连接指定 address  
socket.connect(("192.168.0.1", 80))
```

## 3.5. recv

### 3.5.1. 接口概述

从套接字接收数据。

### 3.5.2. 函数原型

```
socket.recv(size)
```

### 3.5.3. 参数描述

参数名	是否必填	数据类型	默认值	说明
size	是	int	-	一次接收的最大数据量读数据

### 3.5.4. 返回值描述

返回字节对象，表示接收到的数据。

### 3.5.5. 示例

```
buf = socket.recv(1024)
```

## 3.6. recvfrom

### 3.6.1. 接口概述

从套接字接收数据(适用于 UDP 套接字)。

### 3.6.2. 函数原型

```
socket.recvfrom(size)
```

### 3.6.3. 参数描述

参数名	是否必填	数据类型	默认值	说明
size	是	int	-	一次接收的最大数据量读数据

### 3.6.4. 返回值描述

返回元组 (bytes, address)。

### 3.6.5. 示例

```
buf = socket.recvfrom(1024)
```

## 3.7. send

### 3.7.1. 接口概述

发送数据到套接字。

### 3.7.2. 函数原型

```
socket.send(bytes)
```

### 3.7.3. 参数描述

参数名	是否必填	数据类型	默认值	说明
bytes	是	bytes	-	待发送的数据

### 3.7.4. 返回值描述

返回实际发送的字节数。

### 3.7.5. 示例

```
socket.send('123'.encode())
```

## 3.8. sendto

### 3.8.1. 接口概述

发送数据到指定的 address (用于 UDP 套接字)。

### 3.8.2. 函数原型

```
socket.sendto(bytes, address)
```

### 3.8.3. 参数描述

参数名	是否必填	数据类型	默认值	说明
bytes	是	bytes	-	待发送的数据
address	是	tuple		目标地址和端口号

### 3.8.4. 返回值描述

返回实际发送的字节数。

### 3.8.5. 示例

```
socket.sendto('123'.encode(), ('192.168.1.10',10000))
```

## 3.9. setblocking

### 3.9.1. 接口概述

设置套接字为阻塞模式或非阻塞模式。

### 3.9.2. 函数原型

```
socket.setblocking(flag)
```

### 3.9.3. 参数描述

参数名	是否必填	数据类型	默认值	说明
flag	是	bool	-	True: 阻塞模式 False: 非阻塞模式

### 3.9.4. 返回值描述

无返回值。

### 3.9.5. 示例

```
import usocket

# 创建套接字
sock = usocket.socket(usocket.AF_INET, usocket.SOCK_STREAM)
```

```
# 设置为非阻塞模式  
sock.setblocking(False)
```

## 3.10. `settimeout`

### 3.10.1. 接口概述

设置套接字的接收数据超时时间，只有阻塞模式下才有意义。

### 3.10.2. 函数原型

```
socket.settimeout(value)
```

### 3.10.3. 参数描述

参数名	是否必填	数据类型	默认值	说明
value	是	int	-	超时时间（秒）

### 3.10.4. 返回值描述

无返回值。

### 3.10.5. 示例

```
socket.settimeout(30)
```

## 3.11. `close`

### 3.11.1. 接口概述

关闭套接字并释放所有资源。

### 3.11.2. 函数原型

```
socket.close()
```

### 3.11.3. 参数描述

无参数。

### 3.11.4. 返回值描述

无返回值。

### 3.11.5. 示例

```
socket.close()
```