

51CTO | HarmonyOS技术社区

HARMONY OS

网络编程

许思维

这次课讲什么？

- **Harmony OS IoT硬件Wi-Fi编程**

- STA模式
 - 连接其他热点
- AP模式
 - 创建热点

- **TCP/IP协议和LwIP简介**

- **TCP编程**

- TCP服务端
- TCP客户端

- **UDP编程**

- UDP服务端
- UDP客户端

- **一套代码兼容多个平台**

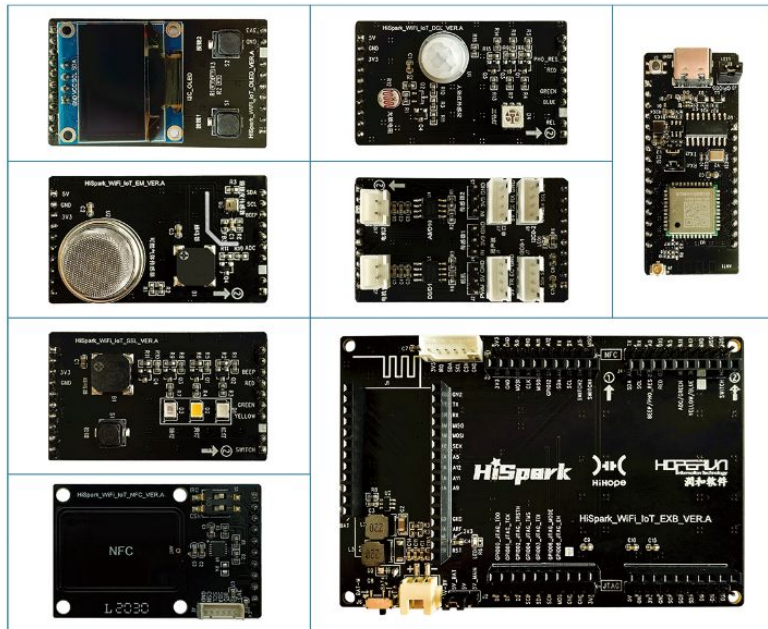
- Hi3861/Hi3516/Hi3518
- Harmony OS(Liteos-m)/Harmony OS(Liteos-a)/Unix like
- RISC-V/ARM/X86

我们的示例代码可以兼容
这些平台

平台	OS	CPU架构
Hi3861	Harmony OS (liteos-m)	RISC-V
Hi3516/Hi3518	Harmony OS (liteos-a)	ARM Cortex A7
PC	类Unix (Linux/Mac)	x86_64

学完可以干什么？

- 突破边界
 - 不再是“单机游戏”
- 通过TCP/UDP功能实现各种创意
 - 通过网络控制开发板上的器件
 - 控制LED灯
 - 控制蜂鸣器
 - 读取传感器
 - 控制小车
- 更高级别的网络应用
 - HTTP服务器
 - CoAP协议移植
 - MQTT协议移植
 - 流媒体协议移植



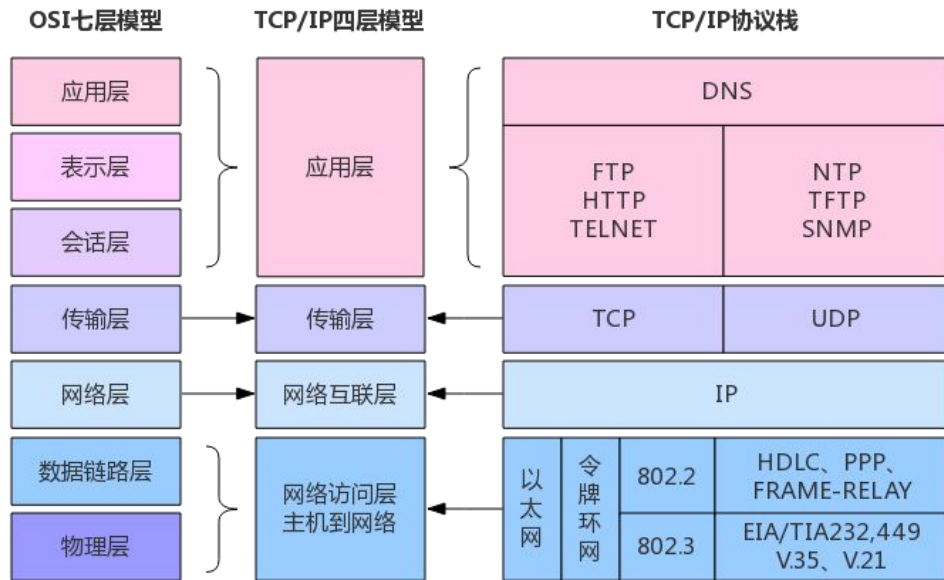
- WiFi的工作模式
 - STA
 - AP
 - Mesh/Any
 - P2P*
- WiFi STA模式
 - 扫描其他设备
 - 连接到其他热点
 - DHCP获取IP
- WiFi AP模式
 - 开启热点
 - 提供DHCP服务

File Name	Description
station_info.h	Defines the structure and macro of the station information contained in the event information.
wifi_device.h	Provides capabilities to enable and disable the station mode, connect to and disconnect from a station, query the station status, and listen for events.
wifi_device_config.h	Defines the Wi-Fi station configuration.
wifi_error_code.h	Defines error codes of the Wi-Fi service.
wifi_event.h	Defines callbacks and structure of Wi-Fi events.
wifi_hotspot.h	Provides capabilities to enable and disable the hotspot mode, connect to and disconnect from a hotspot, query the hotspot status, and listen for events.
wifi_hotspot_config.h	Defines the Wi-Fi hotspot configuration.
wifi_linked_info.h	Defines the data structure and macro of the Wi-Fi connection information.
wifi_scan_info.h	Defines the data structure and macro of the Wi-Fi scan result information.

TCP/IP协议简介

- TCP/IP协议栈

- IP
 - ping
- TCP
 - HTTP/MQTT
- UDP
 - DNS/DHCP
 - NTP/TFTP
 - CoAP

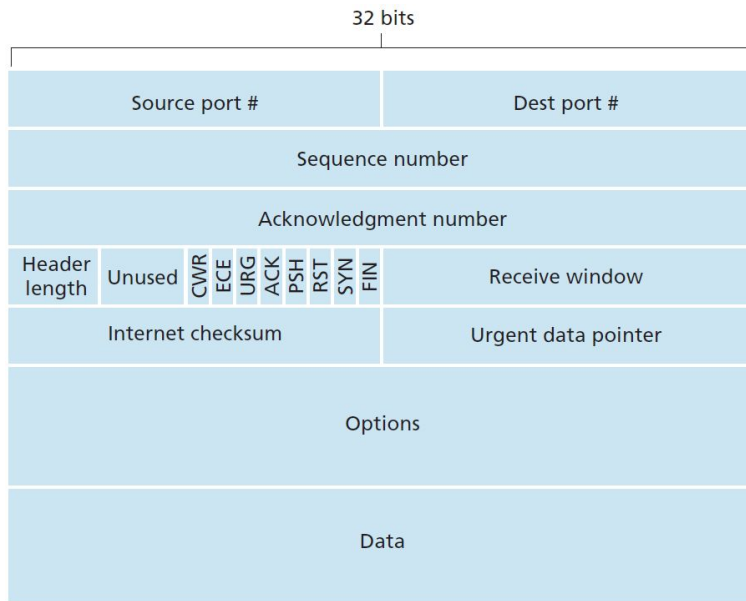


- A Lightweight TCP/IP stack
 - 特性
 - IP
 - TCP
 - HTTP/MQTT
 - UDP
 - DNS/DHCP
 - ICMP
 - IGMP
 - ...
 - 应用
 - HTTP client/server
 - MQTT client
 - SNTP
 - ...
 - Network interface (NETIF) API
 - Netconn API
 - 可以避免数据拷贝（优点）
 - 需要配合netbuf系列接口使用
 - Socket API
 - 和BSD Socket接口一致
 - 辅助API
 - IP地址格式转换
- <https://savannah.nongnu.org/projects/lwip/>
https://www.nongnu.org/lwip/2_1_x/index.html
https://lwip.fandom.com/wiki/LwIP_Wiki
https://lwip.fandom.com/wiki/LwIP_Application_Developers_Manual

- 两份LwIP
 - third_party/lwip
 - 源码形式编译
 - 供liteos-a内核使用
 - 还有一部分代码在kernel/liteos_a中，一起编译
 - vendor/hisi/hi3861/hi3861/third_party/lwip_sack
 - hi3861-sdk的一部分
 - 静态库形式编译
 - 不可修改配置
 - 可以查看当前配置
 - vendor/hisi/hi3861/hi3861/third_party/lwip_sack/include/lwip/lwipopts.h

TCP协议的特点

- 有连接
 - 必须先建立连接
 - 有断开连接的过程
- 一对一
 - 每个“连接”只允许有两个端点
 - 端点：主机（IP）+ 端口
- 可靠传输
 - 有一套机制：序号、确认号、重传
- 面向字节流
 - 虽然协议层面是按块传输数据的
 - 但提供的是一连串的无结构的字节流
 - 因此需要应用层协议，区分应用消息，俗称“拆包”
- 全双工
 - 连接建立后，允许双方同时发送数据



- TCP客户端套路

```
sockfd = socket(AF_INET, SOCK_STREAM, 0);
```

```
serverAddr; // 准备目标服务器的主机、端口信息
```

```
connect(sockfd, &serverAddr, sizeof(serverAddr));
```

```
while (1) {  
    send(sockfd, buffer, buffSize, flags);  
    recv(sockfd, buffer, buffSize, flags);  
}
```

示例代码: <https://gitee.com/hihopeorg/HarmonyOS-IoT-Application-Development>

- TCP服务端套路

```
sockfd = socket(AF_INET, SOCK_STREAM, 0);
```

```
servingAddr; // 准备服务端的主机、端口信息
```

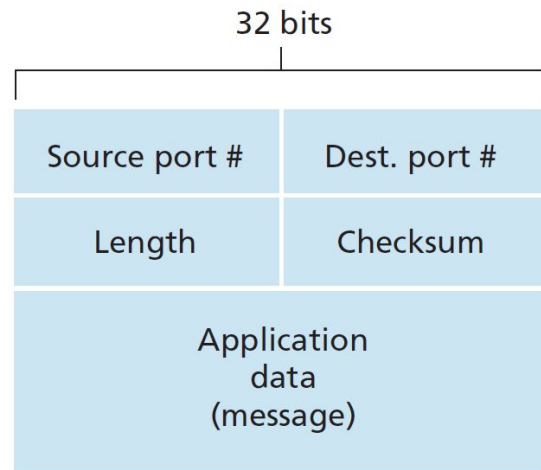
```
bind(sock, &servingAddr, srvAddrLen);
```

```
listen(sockfd, backlog);
```

```
while (1) {  
    connfd = accept(sockfd, &clientAddr, cliAddrLen);  
  
    recv(connfd, request, sizeof(request), 0);  
  
    send(connfd, buffer, sizeof(buffer), 0);  
}
```

示例代码: <https://gitee.com/hihopeorg/HarmonyOS-IoT-Application-Development>

- 无连接
 - 不需要建立连接、断开连接
- 支持一对一、一对多
 - 一个“连接”只允许有两个端点
- 不保证可靠传输
 - 没有序号、确认、重传机制
- 面向报文
 - 对应用层交下来的报文，既不合并，也不拆分
 - 加上UDP头之后交给下一层
- 更小的包头（相对TCP而言）



- UDP客户端套路

```
sockfd = socket(AF_INET, SOCK_STREAM, 0);
```

```
while (1) {  
    sendto(sockfd, buffer, buffSize, flags, &toAddr, toLen);  
    recvfrom(sockfd, buffer, buffSize, flags, &fromAddr, &fromLen);  
}
```

示例代码: <https://gitee.com/hihopeorg/HarmonyOS-IoT-Application-Development>

- UDP服务端套路

```
sockfd = socket(AF_INET, SOCK_STREAM, 0);
```

```
bind(sock, &servingAddr, srvAddrLen);
```

```
while (1) {  
    recvfrom(sockfd, buffer, buffSize, flags, );  
    sendto(sockfd, buffer, buffSize, flags, );  
}
```

示例代码: <https://gitee.com/hihopeorg/HarmonyOS-IoT-Application-Development>

- netcat——瑞士军刀
 - 模拟TCP/UDP服务端，调试客户端程序
 - 模拟TCP/UDP客户端，调试服务端程序
- tcpdump——CT扫描仪
 - 抓取TCP/IP协议包数据
 - 命令行分析协议数据
- wireshark——显微镜
 - 抓取TCP/IP协议包数据
 - 可视化分析协议数据

<https://manpages.debian.org/buster/netcat-traditional/nc.1.en.html>

<https://www.tcpdump.org/manpages/tcpdump.1.html>

<https://wiki.wireshark.org/>

Thanks



开发套件购买



开发套件交流群