

BÀI 17: LẬP TRÌNH MẠNG SOCKET

TCP/IP Socket Programming

Mục tiêu bài học

1. **Mô hình Client-Server:** Ai phục vụ, ai yêu cầu?
2. **TCP vs UDP:** Tin cậy vs Tốc độ.
3. **Socket Flow:** `socket` -> `bind` -> `listen` -> `accept` .
4. **Thực hành:** Viết IoT Gateway gửi dữ liệu cảm biến lên PC.

1. Socket là gì?

Là "ổ cắm" kết nối mạng. Một socket được định danh bởi cặp: **IP Address : Port Number** .

Ví dụ: **192.168.1.10:8080**

- **Server:** Mở port, lắng nghe (Listen).
- **Client:** Chủ động kết nối (Connect).

2. Quy trình TCP Server

1. `socket()` : Tạo socket.
2. `bind()` : Gắn vào cổng (VD: 8080).
3. `listen()` : Chuyển sang chế độ đợi.
4. `accept()` : **Chặn (Block)** cho đến khi có khách (Client) kết nối. Trả về một `client_fd` mới.
5. `read()/write()` : Giao tiếp qua `client_fd` .
6. `close()` : Ngắt kết nối.

3. Quy trình TCP Client

1. `socket()` : Tạo socket.
2. `connect()` : Kết nối đến IP và Port của Server.
3. `read()/write()` : Gửi/Nhận dữ liệu.
4. `close()` : Nghỉ chơi.

PHẦN THỰC HÀNH (LAB 17)

Remote Control LED qua Mạng LAN

Yêu cầu

1. Server (Chạy trên Board):

- Lắng nghe Port 8888.
- Nếu nhận chuỗi "ON" → Bật LED (ghi vào sysfs).
- Nếu nhận chuỗi "OFF" → Tắt LED.
- Gửi phản hồi "OK" về Client.

2. Client (Chạy trên PC hoặc điện thoại dùng Netcat):

- Gửi lệnh điều khiển.
- *Test nhanh:* `echo "ON" | nc 192.168.1.xx 8888`

Q & A

Hẹn gặp lại ở Bài 18: Giao tiếp Phần cứng!