

BÀI 18: GIAO TIẾP PHẦN CỨNG USER SPACE

UART, I2C, SPI without Kernel Driver



Mục tiêu bài học

1. **Vấn đề:** Không phải lúc nào cũng cần viết Kernel Driver. User Space Driver nhanh và an toàn hơn.
2. **UART:** Cấu hình Baudrate, Parity dùng `termios`.
3. **I2C:** Giao tiếp cảm biến qua `/dev/i2c-x`.
4. **SPI:** Giao tiếp tốc độ cao qua `/dev/spidev`.

1. UART (Serial Port)

Trong Linux, cổng Serial là `/dev/ttyS0` , `/dev/ttyUSB0` ...

Cấu hình với thư viện `termios` :

- Cờ `c_cflag` : Baudrate (`B115200`), Data bits (`CS8`), Stop bit.
- Cờ `c_lflag` : Chế độ **Raw** (tắt echo, tắt xử lý tín hiệu đặc biệt) → Quan trọng cho truyền dữ liệu nhị phân.

“ Code mẫu: `open("/dev/ttyS1", O_RDWR | O_NOCTTY)` ”

2. I2C (Inter-Integrated Circuit)

Sử dụng driver `i2c-dev`. Kernel tạo ra file `/dev/i2c-1`.

Quy trình:

1. `open("/dev/i2c-1")`.
2. `ioctl(fd, I2C_SLAVE, 0x48)` : Chọn địa chỉ Slave (Ví dụ cảm biến nhiệt độ).
3. `read()/write()` : Đọc ghi thanh ghi cảm biến.

“ Tool: `i2cdetect -y -r 1` để quét địa chỉ I2C trên bus.”

3. SPI (Serial Peripheral Interface)

Sử dụng driver `spidev`. File `/dev/spidevX.Y` (Bus X, Chip Select Y).

Quy trình:

1. `open`.
2. `ioctl(SPI_IOC_WR_MODE)` : Chọn Mode (0, 1, 2, 3).
3. `ioctl(SPI_IOC_WR_MAX_SPEED_HZ)` : Chọn tốc độ.
4. `ioctl(SPI_IOC_MESSAGE)` : Gửi/Nhận dữ liệu full-duplex.



PHẦN THỰC HÀNH (LAB 18)

Đọc cảm biến nhiệt độ/gia tốc (I2C)

Yêu cầu

1. Kết nối cảm biến (ví dụ MPU6050 hoặc BH1750) vào bus I2C của Board.
2. Dùng **i2cdetect** để tìm địa chỉ.
3. Viết chương trình C:
 - Ghi vào thanh ghi cấu hình (Power On).
 - Đọc giá trị đo được.
 - In ra màn hình liên tục.

Q & A

Chúc mừng bạn hoàn thành Giai đoạn 3!

Chuẩn bị sang "Vùng đất cấm": Kernel Driver (Bài 19)!