Trần Hoàng Phong - 22146192

Phạm Nguyên Đạt - 22146103

Đặng Ngọc Huy - 22146126

Hướng Dẫn Sử Dụng Driver và Ứng Dụng Giao Tiếp Cảm Biến Radar LD2420

1. Giới thiệu về LD2420

Cảm biến LD2420 là cảm biến radar vi sóng 24GHz do HiLink sản xuất. Thiết bị này có thể phát hiện chuyển động và sự hiện diện của con người trong phạm vi tối đa khoảng 8 mét, với khả năng phân biệt giữa chuyển đông và đứng yên.

Đặc điểm nổi bật:

- Nhận diện có người hoặc không có người
- Phân biệt người đứng yên và di chuyển
- Chia vùng phát hiện từ 1 đến 8
- Giao tiếp UART tốc độ cao 256000 bps

Mô tả chân và chức năng của từng chân

Tên chân	Mô tả	Ghi chú
VCC	Nuồn cấp	3.3V hoặc 5V (tùy module)
GND	Chân đất	
OT2 (TXD)	Truyền dữ liệu	
RXD	Nhận dữ liệu	
OT1	Ngỗ ra của digital	Mức HIGH khi phát hiện chuyển động (không bắt buộc sử dụng chân này)

Hướng dẫn kết nối HLK_LD2420 với Raspberry bằng kết nối UART

HLK_LD2420	Raspberry Pi (GPIO)
VCC	3.3 hoặc 5V tùy module
GND	GND
OT2(TXD)	GPIO15(RXD0)
RXD	GPIO14(TXD0)

OT1	Bất kì chân GPIO nào để đọc mức	
	HIGT/LOW	

2. Kiến trúc phần mềm

Dự án bao gồm 3 thành phần chính:

- ld2420_driver.c: Tạo device node /dev/ld2420
- user_ld2420.c: Giao tiếp UART và xử lý dữ liệu từ cảm biến
- Makefile: Hỗ trợ biên dịch driver kernel

3. Cài đặt driver

Biên dịch driver:

make

Nap driver vào kernel:

sudo insmod ld2420_driver.ko

Gö driver:

sudo rmmod ld2420_driver

4. Chi tiết driver

Driver tạo node ký tự và không xử lý read/write/ioctl, mục đích để ứng dụng user-space thao tác trực tiếp. Ví dụ:

```
static int __init ld_init(void) {
   major = register_chrdev(0, DEVICE_NAME, &fops);
   ld_class = class_create(THIS_MODULE, DEVICE_NAME);
   ld_device = device_create(ld_class, NULL, MKDEV(major, 0), NULL, DEVICE_NAME);
   printk(KERN_INFO "[LD2420] Simple character device loaded\n");
   return 0;
}
```

5. Ứng dụng người dùng

Cách sử dụng:

```
sudo ./user_ld2420 <vùng_bắt_đầu> <vùng_kết_thúc> [độ_nhạy]
```

6. Cấu hình UART

Thiết lập UART với tốc độ đặc biệt 256000 bps, xử lý bằng ioctl và cấu trúc termios2 (nếu sử dụng kết nối UART trên Raspberry mặc định sẽ là ttyS0, ttyS0 chỉ hổ trợ baudrate là 115200) nếu muốn sử dụng 256000 thì sử dụng ttyAMA0)

7. Cấu hình cảm biến

Một số lệnh điều khiển chính:

- Vào chế độ cấu hình
 {0xFD, 0xFC, 0xFB, 0xFA, ...}
- Thiết lập khoảng phát hiện
- Thiết lập độ nhạy

8. Theo dõi dữ liệu cảm biến

Ứng dụng sẽ in ra trạng thái mỗi khi nhận được frame từ cảm biến:

- 0x00: Không có ai
- 0x01: Có người đứng yên
- 0x02: Có người di chuyển

9. Gợi ý kiểm thử

- 1. Không có người → 'Không có ai'
- 2. Có người đi qua → 'Có người di chuyển'
- 3. Có người đứng trước cảm biến → 'Có người đứng yên'

10. Ghi chú

Bạn có thể thay đổi port UART, mở rộng chức năng để lập trình cấu hình nâng cao cho cảm biến LD2420.

11. Hướng dẫn thay đổi thông số để đạt kết quả mong muốn

Cảm biến LD2420 cho phép người dùng điều chỉnh các tham số để phù hợp với mục đích sử dụng cụ thể như: phát hiện xa hơn, nhạy hơn, hoặc loại bỏ nhiễu. Dưới đây là các hướng dẫn chi tiết.

• Thay đổi khoảng phát hiện (Vùng phát hiện):

Tham số này xác định phạm vi hoạt động của cảm biến. Mỗi vùng tương ứng với khoảng 1m (tối đa 8 vùng).

Trong chương trình, vùng bắt đầu và vùng kết thúc được truyền vào dòng lệnh:

Ví du:

sudo ./user_ld2420 1 3

Câu lệnh trên cấu hình cảm biến để phát hiện chuyển động từ 1m đến 3m tính từ cảm biến.

Thay đổi độ nhạy:

Tham số này điều chỉnh độ nhạy trong phát hiện, với giá trị từ 0 (rất thấp) đến 255 (rất nhay).

Ví du:

sudo ./user_ld2420 1 5 220

Câu lệnh trên giúp cảm biến nhạy hơn với chuyển động nhỏ trong vùng 1 đến 5m. Giảm giá trị nếu cảm biến phát hiện sai (nhiễu hoặc phản ứng với vật không mong muốn).

12. Cách thay đổi dữ liệu cấu hình trong mã nguồn

Trong mã nguồn `user_ld2420.c`, các hàm như `set_distance()` và `set_sensitivity()` là nơi thay đổi cấu hình gửi tới cảm biến.

Thay đổi vùng phát hiện:

```
void set_distance(int fd, unsigned char start, unsigned char end) {
    ...
}
    Thay đổi giá trị 'start' và 'end' sẽ điều chỉnh phạm vi cảm biến.
    Thay đổi độ nhạy:

void set_sensitivity(int fd, unsigned char sens) {
    ...
}
    Truyền giá tri 'sens' trong khoảng 0-255 để điều chỉnh đô nhay của cảm biến.
```

Bạn có thể chỉnh sửa các giá trị mặc định trong hàm `main()` nếu muốn sử dụng thông số cố định mà không cần nhập từ dòng lệnh.

13. Công dụng chính của LD2420

• Nhát hiện chuyển động (Motion Detection):

Phát hiện người di chuyển trong vùng quét như đi lai, cử động tay, v.v.

Phân biệt giữa có người đứng yên và không có ai - điều mà cảm biến PIR (hồng ngoại) không làm tốt.

• Yham vi hoat đông linh hoat:

Điều chỉnh khoảng cách phát hiện (tối đa ~6m), góc quét rộng.

Chống nhiễu ánh sáng & thời tiết:

Không bi ảnh hưởng bởi ánh sáng, bui, hoặc nhiệt đô như cảm biến hồng ngoại (PIR).

• Kết nối UART hoặc GPIO:

Giao tiếp dễ dàng với các vi điều khiển như Arduino, STM32, Raspberry Pi...

14. Ứng dụng thực tế của LD2420

- Nhà thông minh (Smart Home):
- Tự động bật/tắt đèn khi có người trong phòng.
- Bật máy lạnh hoặc quat khi có người.
- Điều khiển đèn ngủ theo sự hiện diện (phát hiện cả khi người không di chuyển).
- Văn phòng và công sở:
- Tiết kiệm năng lương bằng cách tắt thiết bị khi không có người.
- Kiểm tra mật độ người sử dụng để phân tích tối ưu không gian.
- Thiết bị an ninh:
- Cảnh báo khi có người xâm nhập, kể cả khi đứng yên.
- Giám sát vùng cấm như kho hàng hoặc khu vực nhạy cảm.
- Thiết bị chăm sóc sức khỏe / người già:

- Theo dõi hoạt động, phát hiện người ngã hoặc đứng yên quá lâu.
- Cảnh báo sớm cho người nhà hoặc nhân viên y tế.
- Thiết bị IoT và robot:
- Tích hợp vào robot để phát hiện người dùng xung quanh.
- Dùng trong hệ thống cảm biến thông minh ngoài trời, như trạm xe buýt hoặc bãi đỗ xe.