Hướng dẫn Test Driver với Virtual Sensor

Tổng quan

Driver này tạo **sensor ảo** để test DMA và char device interface mà **KHÔNG CẦN hardware thật**:

✅ **Không cần camera IMX219**
✅ **Không cần I2C**
✅ **Không cần GPIO/Power**
✅ **Tạo frame test pattern 30 FPS**
✅ **Hoàn toàn software-based**

Các file cần thiết

```
project/
├── unicam-virtual-sensor.c    # Driver với sensor giả lập
├── bcm2835-unicam-regs.h      # Register definitions
├── test-virtual.c             # Test program
├── Makefile                   # Build configuration
```

Makefile

makefile

```makefile
# Makefile for Virtual Unicam Driver

obj-m := unicam-virtual-sensor.o

KDIR := /lib/modules/$(shell uname -r)/build
PWD := $(shell pwd)

all: driver test

driver:
	make -C $(KDIR) M=$(PWD) modules

test: test-virtual.c
	gcc -o test-virtual test-virtual.c -Wall

clean:
	make -C $(KDIR) M=$(PWD) clean
	rm -f test-virtual *.pgm

install:
	sudo insmod unicam-virtual-sensor.ko

uninstall:
	sudo rmmod unicam-virtual-sensor

dmesg:
	dmesg | tail -50

log:
	dmesg -w | grep -E "virtual-unicam|unicam"
```

Build và Cài đặt

1. Biên dịch

bash

```bash
# Build driver module
make driver

# Build test program
make test

# Hoặc build tất cả
make all
```

2. Load driver module

```bash
# Load module
sudo insmod unicam-virtual-sensor.ko

# Kiểm tra module đã load
lsmod | grep unicam

# Xem log
dmesg | tail -30
```

Bạn sẽ thấy:

```
[  123.456] virtual-unicam: loading out-of-tree module
[  123.457] vendor,manual-unicam fe801000.csi: === Probing Virtual Unicam Driver ===
[  123.458] vendor,manual-unicam fe801000.csi: ✓ DMA buffer: 16777216 bytes at
0x3ea00000
[  123.459] vendor,manual-unicam fe801000.csi: ✓ Virtual sensor thread created
[  123.460] virtual-unicam fe801000.csi: Virtual sensor thread started
[  123.461] ╔═══════════════════════════════════════════╗
[  123.462] ║  Virtual Unicam Driver Ready!             ║
[  123.463] ╠═══════════════════════════════════════════╣
[  123.464] ║  Device: /dev/unicam0                     ║
[  123.465] ║  Mode:   SOFTWARE SIMULATION              ║
[  123.466] ║  Frame:  640x480 @ 30 FPS                 ║
[  123.467] ║  Buffer: 16 MB (25 frames)                ║
[  123.468] ║  Type:   Test pattern generator           ║
[  123.469] ╚═══════════════════════════════════════════╝
```

3. Kiểm tra device

```bash
# Xem device node
ls -l /dev/unicam0

# Output:
crw------- 1 root root 244, 0 Jan  8 15:30 /dev/unicam0
```

Chạy Test Program

Test cơ bản (10 frames)

```bash
sudo ./test-virtual
```

Output:

```
=== Virtual Unicam Test Program ===
Capturing 10 frames from virtual sensor...

[1] Opening /dev/unicam0...
✓ Device opened

[2] Getting buffer information...
✓ Buffer size: 16777216 bytes (16.00 MB)
  Frame size: 614400 bytes
  Max frames: 27

[3] Mapping buffer to user space...
✓ Buffer mapped at 0x7f8a3b000000

[4] Starting virtual sensor stream...
✓ Streaming started

[5] Waiting for frames...
────────────────────────────────────────────────────────────
Frame #  1 | Time:   0.05s | Offset: 0x00000000 | Brightness:  512 | CHANGED
✓ Saved: frame_000.pgm
Frame #  2 | Time:   0.08s | Offset: 0x00096000 | Brightness:  513 | CHANGED
✓ Saved: frame_001.pgm
Frame #  3 | Time:   0.11s | Offset: 0x0012c000 | Brightness:  514 | CHANGED
✓ Saved: frame_002.pgm
Frame #  4 | Time:   0.14s | Offset: 0x001c2000 | Brightness:  515 | CHANGED
Frame #  5 | Time:   0.17s | Offset: 0x00258000 | Brightness:  516 | CHANGED
...
────────────────────────────────────────────────────────────

[6] Statistics:
  Total frames: 10
  Total time: 0.33 seconds
  Average FPS: 30.30
  Expected FPS: 30.0

[7] Stopping stream...
✓ Streaming stopped

[8] Cleanup...
✓ Resources released

═══════════════════════════════════════════════════════════
Test completed successfully!
Check frame_000.pgm, frame_001.pgm, frame_002.pgm
View with: display frame_000.pgm
═══════════════════════════════════════════════════════════
```

Test với nhiều frames

```bash
# Capture 100 frames
sudo ./test-virtual 100

# Capture 300 frames (10 seconds @ 30 FPS)
sudo ./test-virtual 300
```

Xem Frame Images

```bash
# Cài ImageMagick (nếu chưa có)
sudo apt install imagemagick

# Xem frame
display frame_000.pgm

# Xem tất cả frames dạng slideshow
display frame_*.pgm

# Convert sang PNG
convert frame_000.pgm frame_000.png

# Tạo GIF animation từ nhiều frames
convert -delay 33 frame_*.pgm animation.gif
```

Test Pattern Types

Driver tạo 4 loại pattern khác nhau theo vùng:

1. **Top 1/4**: Horizontal gradient + frame counter
2. **Upper middle 1/4**: Vertical gradient
3. **Lower middle 1/4**: Checkerboard pattern
4. **Bottom 1/4**: Frame number display

Mỗi frame khác nhau một chút để dễ verify frame rate.

Advanced Testing

Test performance

```python
#!/usr/bin/env python3
import time
import os

def test_fps(duration=10):
    """Test actual FPS over duration seconds"""
    os.system('sudo ./test-virtual %d > test_output.txt' % (duration * 30))

    with open('test_output.txt') as f:
        for line in f:
            if 'Average FPS:' in line:
                fps = float(line.split(':')[1].strip())
                print(f"Achieved FPS: {fps:.2f}")

                if abs(fps - 30.0) < 1.0:
                    print("✓ Frame rate is accurate!")
                else:
                    print("⚠ Frame rate deviation detected")

test_fps()
```

Monitor memory usage

```bash
# Watch memory in real-time
watch -n 1 'cat /proc/meminfo | grep -E "MemFree|Cached"'

# Check driver memory allocation
cat /proc/buddyinfo
```

Check circular buffer behavior

```bash
# Capture more frames than buffer can hold
sudo ./test-virtual 100

# Buffer should wrap around correctly
# Check dmesg for any errors
dmesg | grep -i error
```

Debug Commands

```bash
# Monitor kernel log in real-time
sudo dmesg -w | grep -E "virtual|unicam"

# Check device info
cat /proc/devices | grep unicam

# Check interrupt stats (should show 0 for virtual)
cat /proc/interrupts | grep unicam

# Memory mapped info
cat /proc/$(pgrep test-virtual)/maps | grep unicam
```

Troubleshooting

Lỗi: "No such device"

```bash
# Kiểm tra module loaded
lsmod | grep unicam

# Load lại module
sudo rmmod unicam-virtual-sensor
sudo insmod unicam-virtual-sensor.ko
```

Lỗi: "Permission denied"

```bash
# Chạy với sudo
sudo ./test-virtual

# Hoặc thay đổi permissions (không khuyến khích)
sudo chmod 666 /dev/unicam0
```

Lỗi: "mmap failed"

```bash
# Kiểm tra DMA buffer allocation
dmesg | grep "DMA buffer"

# Có thể cần tăng CMA memory trong /boot/config.txt
# Thêm dòng: cma=256M
```

Frame rate không đúng 30 FPS

```bash
# Kiểm tra CPU throttling
vcgencmd measure_clock arm

# Kiểm tra nhiệt độ
vcgencmd measure_temp

# Tắt power saving
sudo cpupower frequency-set -g performance
```

Unload Driver

```bash
# Stop test program trước (nếu đang chạy)
sudo killall test-virtual

# Unload module
sudo rmmod unicam-virtual-sensor

# Verify removed
lsmod | grep unicam
```

So sánh với Real Hardware

Feature	Virtual Sensor	Real IMX219
Hardware required	❌ None	✅ Camera + cables
Power management	❌ Simulated	✅ Required
I2C configuration	❌ Not needed	✅ Required
CSI-2 timing	❌ Simulated	✅ Must be correct
Frame generation	✅ Software	✅ Hardware
Frame rate	✅ Precise 30 FPS	⚠ May vary
Test patterns	✅ Programmable	❌ Limited
Debugging	✅ Easy	⚠ Complex

Khi nào dùng Virtual Sensor?

✅ **Phát triển driver interface**
✅ **Test DMA buffer management**
✅ **Debug char device operations**
✅ **Test userspace applications**
✅ **CI/CD testing không cần hardware**
✅ **Teaching/Learning driver development**

❌ **Không thay thế được:**

- CSI-2 PHY timing verification
- Real camera sensor behavior
- Power management testing
- I2C communication testing

Chuyển sang Real Hardware

Khi driver hoạt động tốt với virtual sensor, chuyển sang hardware:

1. Thay `unicam-virtual-sensor.c` → original driver
2. Add I2C sensor configuration
3. Add power management
4. Configure CSI-2 lanes
5. Test với camera thật

Driver virtual giúp bạn verify:

- ✅ DMA buffer allocation works
- ✅ mmap to userspace works
- ✅ IOCTL interface works
- ✅ Frame tracking works
- ✅ Circular buffer logic works

Giờ chỉ cần tập trung vào hardware-specific code!