

CIS5930 Final Project Report

Team Members:

De Yu Liu
Kannan Sankaran
Antony Pathrose

The project primarily consists of the following features:

- 1) Probing PCI devices
- 2) Live mode
- 3) Freeze mode
- 4) Mmap implementation
- 5) Streaming implementation

The init and cleanup functions

The `frgrab_init_module` function registers the device after probing for the correct base address, which varies according to the jumper settings. The technique used for probing is as follows:

1. Freeze the device first. (To avoid values changing between read/writes.)
2. Assume one of the known base addresses to be the base address for the current jumper setting.
2. Write an arbitrary value at an address that is offset from the assumed base address by a known value.
3. Read back the value from the same address.
4. If the written and read values match then the assumed address is the correct base address. Otherwise iterate with the other known addresses until success or all known addresses are probed whichever is earlier.
5. If still not success it implies jumper A is on. Use the `pci_resource_start` function to determine the base address in this case.

The `frgrab_cleanup_module` does the opposite i.e. unregisters the device.

The main video device structure

The video device information is stored in the structure `frgrab`. This contains the various parameters for the video device.

Functionality of the device driver

The functionality of the video device driver is contained in the file operations structure that contains the mapping of the user-defined calls to the actual driver implementation(methods). Accordingly, there are the following important functions:

- 1) `frgrab_open` – This function simply increments the usage count of the device.
- 2) `frgrab_close` – This function simply decrements the usage count of the device.
- 3) `frgrab_read` - This is the main function that implements the functionality to capture the pictures in the LIVE and FREEZE modes.

- 4) `frgrab_ioctl` – This is an important function that contains a whole range of options that the user can implement in order to change the image properties--options such as Brightness, Contrast etc.

Various modes

The following are the main modes that were implemented:

- 1) Live mode – This is implemented by the driver in the following manner:

```
writeb(frgrab_polarity, pstv-  
frgrab_mem+PIXELSMART_POLARITY_CONTROL_REG);  
wmb();  
writeb(0x0091, pstv->frgrab_mem+0x2000);  
wmb();
```

- 3) Freeze mode:

```
writeb(0x0099, pstv->frgrab_mem+0x2000);  
wmb();
```

- 2) Adjust Brightness case `SETBRIGHTNESS`:

```
__GET_USER(p, (int *)arg);  
I2C_start();  
I2C_send_byte(HRT_AD_DEVICE_ID);  
I2C_send_byte(25);  
I2C_send_byte(p);  
I2C_stop();  
return 0;
```

- 3) Adjust Contrast

```
case SETCONTRAST:  
__GET_USER(p, (int *)arg);  
I2C_start();  
I2C_send_byte(HRT_AD_DEVICE_ID);  
I2C_send_byte(19);  
I2C_send_byte(p);  
I2C_stop();  
return 0;
```

- 6) Read: we implemented the read by kernel timer and `tq_timer`.

- 4) Mmap

The `mmap` function does the mapping of the device memory given by the probing device onto the virtual memory assigned by the machine.

5) Streaming implementation – This was attempted but was not completely implemented.

Problems faced

In order to work on streaming we had installed videodevX ,which rewrote the default videodev.h with its own videodev.h. And it claimed that on uninstalling it , the old videodev.h would be re-instated. Apparently that is not true. Because of our lack of prior experience in patching kernels, we were naïve to believe the claim and wasted the entire weekend trying to find out where the problem was, in our code, when actually the problem was somewhere else .

There is a good lesson in it for us—Treat all claims as disclaimers!

Test Files

V4lgrab.c and test.c– to test ioctl's and grab one frame.

V4lgrabmmap2.c – to test mmap

Alex Rudnick's hrtdemo program– To get a continuous flow of pictures(not streaming.)