libv4l

Hans de Goede
Red Hat
Contents

- Introduction
- Future (from last years presentation)
- Emulated controls
- When to enable?
- Video processing
- Future
Introduction

- Get gspca into the mainline kernel
- Remove format conversion from gspca
- Applications don't handle the new formats
- Solution: Write a conversion library
- And patch ALL applications to use it
The Future?

- Add emulated controls
- Better handle rotation
- Software image quality enhancements:
  - White balance
  - Normalize
- Software image quality enhancements have a separate measure / transform phase
- GET_WEBCAM_ATTR ioctl
- Emulated controls persistency
Emulated controls

- Persistent
- Shared

- Using shared memory:
  
  /dev/shm/libv4l-hans:usb-0000:00:1d.7-5:
  BisonCam NB Pro

- Disadvantage: settings are per user
When to enable?

- Enabling emulated controls, forces libv4l to go through the conversion path, even when not converting.
- libv4l has a device table for devices which benefit from software video processing.
- Also enabled when the device will always need conversion.
Video Processing

- White balancing, simple make image gray on average algorithm
- Gamma correction
- Manipulate per component (RGB) component lookup tables, so once sort of processing is enabled, further processing is essentially free.
- Only works in RGB color space, requires YUV -> RGB -> YUV conversion in certain cases
The Future

- Add plugable processing algorithms, so that we can have driver specific algorithms
- Move quirk table to a config file?
- Support for software auto focus?
- Support for video input device sharing?
- Get more out of tree drivers into the mainline, port mainline v4l1 drivers to v4l2
libv4l

Hans de Goede
Red Hat
Contents

- Introduction
- Future (from last years presentation)
- Emulated controls
- When to enable?
- Video processing
- Future
Introduction

- Get gspca into the mainline kernel
- Remove format conversion from gspca
- Applications don't handle the new formats
- Solution: Write a conversion library
- And patch ALL applications to use it
The Future?

- Add emulated controls
- Better handle rotation
- Software image quality enhancements:
  - White balance
  - Normalize
- Software image quality enhancements have a separate measure / transform phase
- GET_WEBCAM_ATTR ioctl
- Emulated controls persistency
Emulated controls

- Persistent
- Shared
- Using shared memory:
  /dev/shm/libv4l-hans:usb-0000:00:1d.7-5:
  BisonCam NB Pro
- Disadvantage: settings are per user
When to enable?

- Enabling emulated controls, forces libv4l to go through the conversion path, even when not converting.
- libv4l has a device table for devices which benefit from software video processing.
- Also enabled when the device will always need conversion.
Video Processing

- White balancing, simple make image gray on average algorithm
- Gamma correction
- Manipulate per component (RGB) component lookup tables, so once sort of processing is enabled, further processing is essentially free.
- Only works in RGB color space, requires YUV -> RGB -> YUV conversion in certain cases
The Future

- Add plugable processing algorithms, so that we can have driver specific algorithms
- Move quirk table to a config file?
- Support for software auto focus?
- Support for video input device sharing?
- Get more out of tree drivers into the mainline, port mainline v4l1 drivers to v4l2