The cpia2 driver

Authors: Peter Pregler <Peter_Pregler@email.com>, Scott J. Bertin <scottbertin@yahoo.com>, and Jarl Totland <Jarl.Totland@bdc.no> for the original cpia driver, which this one was modelled from.

Introduction

This is a driver for STMicroelectronics's CPiA2 (second generation Colour Processor Interface ASIC) based cameras. This camera outputs an MJPEG stream at up to vga size. It implements the Video4Linux interface as much as possible. Since the V4L interface does not support compressed formats, only an mjpeg enabled application can be used with the camera. We have modified the gqcam application to view this stream.

The driver is implemented as two kernel modules. The cpia2 module contains the camera functions and the V4L interface. The cpia2_usb module contains usb specific functions. The main reason for this was the size of the module was getting out of hand, so I separated them. It is not likely that there will be a parallel port version.

Features

- Supports cameras with the Vision stv6410 (CIF) and stv6500 (VGA) cmos sensors. I only have the vga sensor, so can't test the other.
- Image formats: VGA, QVGA, CIF, QCIF, and a number of sizes in between. VGA and QVGA are the native image sizes for the VGA camera. CIF is done in the coprocessor by scaling QVGA. All other sizes are done by clipping.
- Palette: YCrCb, compressed with MJPEG.
- Some compression parameters are settable.
- Sensor framerate is adjustable (up to 30 fps CIF, 15 fps VGA).
- Adjust brightness, color, contrast while streaming.
- Flicker control settable for 50 or 60 Hz mains frequency.

Making and installing the stv672 driver modules

Requirements

Video4Linux must be either compiled into the kernel or available as a module. Video4Linux2 is automatically detected and made available at compile time.

Setup

Use `modprobe cpia2` to load and `modprobe -r cpia2` to unload. This may be done automatically by your distribution.

Driver options

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\admin-guide\media\(linux-master)(Documentation)(admin-guide)(media)cpia2.rst, line 60)

Unknown directive type "tabularcolumns".

```
.. tabularcolumns:: |p{13ex}|L|
```

Option	Description
video_nr	video device to register (0=/dev/video0, etc) range -1 to 64. default is -1 (first available) If you have more than 1 camera, this MUST be -1.
buffer_size	Size for each frame buffer in bytes (default 68k)
num_buffers	Number of frame buffers (1-32, default 3)
alternate	USB Alternate (2-7, default 7)
flicker_freq	Frequency for flicker reduction(50 or 60, default 60)
flicker_mode	0 to disable, or 1 to enable flicker reduction. (default 0). This is only effective if the camera uses a stv0672 coprocessor.

Setting the options

If you are using modules, edit /etc/modules.conf and add an options line like this:

```
options cpia2 num_buffers=3 buffer_size=65535
```

If the driver is compiled into the kernel, at boot time specify them like this:

```
cpia2.num_buffers=3 cpia2.buffer_size=65535
```

What buffer size should I use?

The maximum image size depends on the alternate you choose, and the frame rate achieved by the camera. If the compression engine is able to keep up with the frame rate, the maximum image size is given by the table below.

The compression engine starts out at maximum compression, and will increase image quality until it is close to the size in the table. As long as the compression engine can keep up with the frame rate, after a short time the images will all be about the size in the table, regardless of resolution.

At low alternate settings, the compression engine may not be able to compress the image enough and will reduce the frame rate by producing larger images.

The default of 68k should be good for most users. This will handle any alternate at frame rates down to 15fps. For lower frame rates, it may be necessary to increase the buffer size to avoid having frames dropped due to insufficient space.

Alternate	bytes/ms	15fps	30fps
2	128	8533	4267
3	384	25600	12800
4	640	42667	21333
5	768	51200	25600
6	896	59733	29867
7	1023	68200	34100

Table: Image size(bytes)

How many buffers should I use?

For normal streaming, 3 should give the best results. With only 2, it is possible for the camera to finish sending one image just after a program has started reading the other. If this happens, the driver must drop a frame. The exception to this is if you have a heavily loaded machine. In this case use 2 buffers. You are probably not reading at the full frame rate. If the camera can send multiple images before a read finishes, it could overwrite the third buffer before the read finishes, leading to a corrupt image. Single and double buffering have extra checks to avoid overwriting.

Using the camera

We are providing a modified gqcam application to view the output. In order to avoid confusion, here it is called mview. There is also the qx5view program which can also control the lights on the qx5 microscope. MJPEG Tools (http://mjpeg.sourceforge.net) can also be used to record from the camera.