I have a C/GTK program that creates the following pipeline:-

| Camera | | Video | | Caps | | Queue | | Video | | Video |

| v4l2src |->| Rate |->| Filter |->| (blk) |->| convert |->| sink |-> Screen

which gets changed dynamically, adding 2 elements, when a menu option is selected (there is also a reversing option) into this:-

| Camera | | Video | | Caps | | Queue | | Video | | Cairo | | Video | | Video |

| v4l2src |->| Rate |->| Filter |->| (blk) |->| convert |->| overlay |->| convert |->| sink |-> Screen

It works anywhere from 0 to n times before the stream will freeze sometimes with the following error

(GST\_DEBUG=1).

libv4l2: error turning on stream: Invalid argument

0:00:07.395996377 3849 0x8856150 ERROR v4l2 gstv4l2bufferpool.c:566:gst\_v4l2\_buffer\_pool\_streamon:<v4l2:pool:src> error with STREAMON 22 (Invalid argument)

The approach I take when the menu option is selected is:-

probe\_id = gst\_pad\_add\_probe (blockpad, GST\_PAD\_PROBE\_TYPE\_BLOCK\_DOWNSTREAM,

OnPadProbe, user\_data pointer , NULL);

and the OnPadProbe routine:-

unlinks the elements in the pipeline

creates the 'cairooverlay' element and the second 'videoconvert' adapter

sets up callbacks

adds the new elements to the pipeline, links them and sets their state to Playing

return GST\_PAD\_PROBE\_REMOVE;

The cairooverlay Callbacks for 'caps-changed' and 'draw' are pretty straightforward.

Is this a valid approach for changing a pipeline on the fly? I've used debug code and found that the pad is still blocked and that the pipeline is still in playing state after the video freezes, but I'm unsure what to try next.

I've found that there is no problem when I switch video sinks from 'xvimagesink' to 'ximagesink', but no idea why.

I've also wondered if there is much penalty in just including the 2 additional elements anyway and connecting / disconnecting the Callbacks as required.

Any advice or pointers would be gladly received.

Regards.

Tony