The bttv driver

bttv and sound mini howto

There are a lot of different bt848/849/878/879 based boards available. Making video work often is not a big deal, because this is handled completely by the bt8xx chip, which is common on all boards. But sound is handled in slightly different ways on each board.

To handle the grabber boards correctly, there is a array tvcards[] in bttv-cards.c, which holds the information required for each board. Sound will work only, if the correct entry is used (for video it often makes no difference). The bttv driver prints a line to the kernel log, telling which card type is used. Like this one:

```
bttv0: model: BT848(Hauppauge old) [autodetected]
```

You should verify this is correct. If it isn't, you have to pass the correct board type as insmod argument, `insmod bttv card=2` for example. The file Documentation/admin-guide/media/bttv-cardlist.rst has a list of valid arguments for card.

If your card isn't listed there, you might check the source code for new entries which are not listed yet. If there isn't one for your card, you can check if one of the existing entries does work for you (just trial and error...).

Some boards have an extra processor for sound to do stereo decoding and other nice features. The msp34xx chips are used by Hauppauge for example. If your board has one, you might have to load a helper module like `msp3400` to make sound work. If there isn't one for the chip used on your board: Bad luck. Start writing a new one. Well, you might want to check the video4linux mailing list archive first...

Of course you need a correctly installed soundcard unless you have the speakers connected directly to the grabber board. Hint: check the mixer settings too. ALSA for example has everything muted by default.

How sound works in detail

Still doesn't work? Looks like some driver hacking is required. Below is a do-it-yourself description for you.

The bt8xx chips have 32 general purpose pins, and registers to control these pins. One register is the output enable register (`BT848_GPIO_OUT_EN`), it says which pins are actively driven by the bt848 chip. Another one is the data register (`BT848_GPIO_DATA`), where you can get/set the status if these pins. They can be used for input and output.

Most grabber board vendors use these pins to control an external chip which does the sound routing. But every board is a little different. These pins are also used by some companies to drive remote control receiver chips. Some boards use the i2c bus instead of the gpio pins to connect the mux chip.

As mentioned above, there is a array which holds the required information for each known board. You basically have to create a new line for your board. The important fields are these two:

```
struct tvcard
{
      [ ... ]
      u32 gpiomask;
      u32 audiomux[6]; /* Tuner, Radio, external, internal, mute, stereo */
};
```

gpiomask specifies which pins are used to control the audio mux chip. The corresponding bits in the output enable register (`BT848_GPIO_OUT_EN`) will be set as these pins must be driven by the bt848 chip.

The `audiomux[]` array holds the data values for the different inputs (i.e. which pins must be high/low for tuner/mute/...). This will be written to the data register (`BT848_GPIO_DATA`) to switch the audio mux.

What you have to do is figure out the correct values for gpiomask and the audiomux array. If you have Windows and the drivers four your card installed, you might to check out if you can read these registers values used by the windows driver. A tool to do this is available from http://btwincap.sourceforge.net/download.html.

You might also dig around in the `*.ini` files of the Windows applications. You can have a look at the board to see which of the gpio pins are connected at all and then start trial-and-error ...

Starting with release 0.7.41 bttv has a number of insmod options to make the gpio debugging easier:

bttv_gpio=0/1	enable/disable gpio debug messages
gpiomask=n	set the gpiomask value
audiomux=i,j,...	set the values of the audiomux array
audioall=a	set the values of the audiomux array (one value for all array elements, useful to check out which effect the particular value has).

The messages printed with `bttv_gpio=1` look like this:

```
bttv0: gpio: en=00000027, out=00000024 in=00ffffd8 [audio: off]
```

```
en  =   output _en_able register (BT848_GPIO_OUT_EN)
out =   _out_put bits of the data register (BT848_GPIO_DATA),
        i.e. BT848_GPIO_DATA & BT848_GPIO_OUT_EN
in  =   _in_put bits of the data register,
        i.e. BT848_GPIO_DATA & ~BT848_GPIO_OUT_EN
```