The cx88 driver

Author: Gerd Hoffmann

Documentation missing at the cx88 datasheet

MO_OUTPUT_FORMAT (0x310164)

System Message: WARNING/2 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\media\drivers\[linux-master][Documentation][driver-api][media][drivers]cx88-devel.rst, line 13)

Cannot analyze code. No Pygments lexer found for "none".

```
.. code-block:: none

  Previous default from DScaler: 0x1c1f0008
  Digit 8: 31-28
  28: PREVREMOD = 1

  Digit 7: 27-24 (0xc = 12 = b1100 )
  27: COMBALT = 1
  26: PAL_INV_PHASE
    (DScaler apparently set this to 1, resulted in sucky picture)

  Digits 6,5: 23-16
  25-16: COMB_RANGE = 0x1f [default] (9 bits -> max 512)

  Digit 4: 15-12
  15: DISIFX = 0
  14: INVCBF = 0
  13: DISADAPT = 0
  12: NARROWADAPT = 0

  Digit 3: 11-8
  11: FORCE2H
  10: FORCEREMD
  9: NCHROMAEN
  8: NREMODEN

  Digit 2: 7-4
  7-6: YCORE
  5-4: CCORE

  Digit 1: 3-0
  3: RANGE = 1
  2: HACTEXT
  1: HSFMT
```

0x47 is the sync byte for MPEG-2 transport stream packets. Datasheet incorrectly states to use 47 decimal. 188 is the length. All DVB compliant frontends output packets with this start code.

Hauppauge WinTV cx88 IR information

The controls for the mux are GPIO [0,1] for source, and GPIO 2 for muting.

GPIO0	GPIO1	
0	0	TV Audio
1	0	FM radio
0	1	Line-In
1	1	Mono tuner bypass or CD passthru (tuner specific)

GPIO 16(I believe) is tied to the IR port (if present).

From the data sheet:

- Register 24'h20004 PCI Interrupt Status
  - bit [18] IR_SMP_INT Set when 32 input samples have been collected over
  - gpio[16] pin into GP_SAMPLE register.

What's missing from the data sheet:

- Setup 4KHz sampling rate (roughly 2x oversampled; good enough for our RC5 compat remote)
- set register 0x35C050 to 0xa80a80
- enable sampling
- set register 0x35C054 to 0x5
- enable the IRQ bit 18 in the interrupt mask register (and provide for a handler)

GP_SAMPLE register is at 0x35C058

Bits are then right shifted into the GP_SAMPLE register at the specified rate; you get an interrupt when a full DWORD is received. You need to recover the actual RC5 bits out of the (oversampled) IR sensor bits. (Hint: look for the 0/1and 1/0 crossings of the RC5 bi-phase data) An actual raw RC5 code will span 2-3 DWORDS, depending on the actual alignment.

I'm pretty sure when no IR signal is present the receiver is always in a marking state(1); but stray light, etc can cause intermittent noise values as well. Remember, this is a free running sample of the IR receiver state over time, so don't assume any sample starts at any particular place.

Additional info

This data sheet (google search) seems to have a lovely description of the RC5 basics:
http://www.atmel.com/dyn/resources/prod_documents/doc2817.pdf

This document has more data: http://www.nenya.be/beor/electronics/rc5.htm

This document has a how to decode a bi-phase data stream: http://www.ee.washington.edu/circuit_archive/text/ir_decode.txt

This document has still more info: http://www.xs4all.nl/~sbp/knowledge/ir/rc5.htm