

hauppauge-wintv-cx88-ir.txt

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

Of course, 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.

[http://www.atmel.com/dyn/resources/prod\\_documents/doc2817.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc2817.pdf)

This data sheet (google search) seems to have a lovely description of the  
RC5 basics

<http://users.pandora.be/nenya/electronics/rc5/> and more data

[http://www.ee.washington.edu/circuit\\_archive/text/ir\\_decode.txt](http://www.ee.washington.edu/circuit_archive/text/ir_decode.txt)  
and even a reference to how to decode a bi-phase data stream.

<http://www.xs4all.nl/~sbp/knowledge/ir/rc5.htm>  
still more info