The Radiotrack radio driver

Author: Stephen M. Benoit <benoits@servicepro.com>

Date: Dec 14, 1996

ACKNOWLEDGMENTS

This document was made based on 'C' code for Linux from Gideon le Grange (legrang@active.co.za or legrang@cs.sun.ac.za) in 1994, and elaborations from Frans Brinkman (brinkman@esd.nl) in 1996. The results reported here are from experiments that the author performed on his own setup, so your mileage may vary... I make no guarantees, claims or warranties to the suitability or validity of this information. No other documentation on the AIMS Lab (http://www.aimslab.com/) RadioTrack card was made available to the author. This document is offered in the hopes that it might help users who want to use the RadioTrack card in an environment other than MS Windows.

WHY THIS DOCUMENT?

I have a RadioTrack card from back when I ran an MS-Windows platform. After converting to Linux, I found Gideon le Grange's command-line software for running the card, and found that it was good! Frans Brinkman made a comfortable X-windows interface, and added a scanning feature. For hack value, I wanted to see if the tuner could be tuned beyond the usual FM radio broadcast band, so I could pick up the audio carriers from North American broadcast TV channels, situated just below and above the 87.0-109.0 MHz range. I did not get much success, but I learned about programming ioports under Linux and gained some insights about the hardware design used for the card.

So, without further delay, here are the details.

PHYSICAL DESCRIPTION

The RadioTrack card is an ISA 8-bit FM radio card. The radio frequency (RF) input is simply an antenna lead, and the output is a power audio signal available through a miniature phone plug. Its RF frequencies of operation are more or less limited from 87.0 to 109.0 MHz (the commercial FM broadcast band). Although the registers can be programmed to request frequencies beyond these limits, experiments did not give promising results. The variable frequency oscillator (VFO) that demodulates the intermediate frequency (IF) signal probably has a small range of useful frequencies, and wraps around or gets clipped beyond the limits mentioned above.

CONTROLLING THE CARD WITH IOPORT

The RadioTrack (base) ioport is configurable for 0x30c or 0x20c. Only one ioport seems to be involved. The ioport decoding circuitry must be pretty simple, as individual ioport bits are directly matched to specific functions (or blocks) of the radio card. This way, many functions can be changed in parallel with one write to the ioport. The only feedback available through the ioports appears to be the "Stereo Detect" bit.

The bits of the ioport are arranged as follows:

System Message: WARNING/2 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\media\drivers\[linux-master][Documentation][driver-api][media][drivers]radiotrack.rst, line 65)

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

```
.. code-block:: none

        MSb                                                     LSb
        +------+------+------+--------+--------+-------+---------+--------+
        | VolA | VolB | ???? | Stereo | Radio  | TuneA | TuneB   | Tune   |
        |  (+) |  (-) |      | Detect | Audio  | (bit) | (latch) | Update |
        |      |      |      | Enable | Enable |       |         | Enable |
        +------+------+------+--------+--------+-------+---------+--------+
```

VolA	VolB	Description
0	0	audio mute
0	1	volume + (some delay required)
1	0	volume - (some delay required)
1	1	stay at present volume

Stereo Detect Enable	Description
0	No Detect
1	Detect

Results available by reading ioport >60 msec after last port write.

0xff ==> no stereo detected, 0xfd ==> stereo detected.

Radio to Audio (path) Enable	Description
0	Disable path (silence)
1	Enable path (audio produced)

TuneA	TuneB	Description
0	0	"zero" bit phase 1
0	1	"zero" bit phase 2
1	0	"one" bit phase 1
1	1	"one" bit phase 2

24-bit code, where bits = (freq*40) + 10486188. The Most Significant 11 bits must be 1010 xxxx 0x0 to be valid. The bits are shifted in LSb first.

Tune Update Enable	Description
0	Tuner held constant
1	Tuner updating in progress

PROGRAMMING EXAMPLES

System Message: WARNING/2 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\media\drivers\[linux-master][Documentation][driver-api][media][drivers]radiotrack.rst, line 127)

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

```
.. code-block:: none

        Default:        BASE <-- 0xc8  (current volume, no stereo detect,
                                        radio enable, tuner adjust disable)

        Card Off:       BASE <-- 0x00  (audio mute, no stereo detect,
                                        radio disable, tuner adjust disable)

        Card On:        BASE <-- 0x00  (see "Card Off", clears any unfinished business)
                        BASE <-- 0xc8  (see "Default")

        Volume Down:    BASE <-- 0x48  (volume down, no stereo detect,
                                        radio enable, tuner adjust disable)
                        wait 10 msec
                        BASE <-- 0xc8  (see "Default")

        Volume Up:      BASE <-- 0x88  (volume up, no stereo detect,
                                        radio enable, tuner adjust disable)
                        wait 10 msec
                        BASE <-- 0xc8  (see "Default")

        Check Stereo:   BASE <-- 0xd8  (current volume, stereo detect,
                                        radio enable, tuner adjust disable)
                        wait 100 msec
                        x <-- BASE     (read ioport)
                        BASE <-- 0xc8  (see "Default")

                        x=0xff ==> "not stereo", x=0xfd ==> "stereo detected"

        Set Frequency:  code = (freq*40) + 10486188
                        foreach of the 24 bits in code,
                        (from Least to Most Significant):
                        to write a "zero" bit,
                        BASE <-- 0x01  (audio mute, no stereo detect, radio
                                        disable, "zero" bit phase 1, tuner adjust)
                        BASE <-- 0x03  (audio mute, no stereo detect, radio
                                        disable, "zero" bit phase 2, tuner adjust)
                        to write a "one" bit,
                        BASE <-- 0x05  (audio mute, no stereo detect, radio
                                        disable, "one" bit phase 1, tuner adjust)
                        BASE <-- 0x07  (audio mute, no stereo detect, radio
                                        disable, "one" bit phase 2, tuner adjust)
```