Software Interface ALSA-DSP MADI Driver

(translated from German, so no good English ;-),
2004 - winfried ritsch

Full functionality has been added to the driver. Since some of the Controls and startup-options are ALSA-Standard and only the special Controls are described and discussed below.

hardware functionality:

Audio transmission:

number of channels -- depends on transmission mode

The number of channels chosen is from 1..Nmax. The reason to use for a lower number of channels is only resource allocation, since unused DMA channels are disabled and less memory is allocated. So also the throughput of the PCI system can be scaled. (Only important for low performance boards).

Single Speed -- 1..64 channels

(Note: Choosing the 56channel mode for transmission or as receiver, only 56 are transmitted/received over the MADI, but all 64 channels are available for the mixer, so channel count for the driver)

Double Speed -- 1..32 channels

Note: Choosing the 56-channel mode for transmission/receive-mode , only 28 are transmitted/received over the MADI, but all 32 channels are available for the mixer, so channel count for the driver

Quad Speed -- 1..16 channels

Note: Choosing the 56-channel mode for transmission/receive-mode , only 14 are transmitted/received over the MADI, but all 16 channels are available for the mixer, so channel count for the driver

Format -- signed 32 Bit Little Endian (SNDRV_PCM_FMTBIT_S32_LE)

Sample Rates --

Single Speed -- 32000, 44100, 48000

Double Speed -- 64000, 88200, 96000 (untested)

Quad Speed -- 128000, 176400, 192000 (untested)

access-mode -- MMAP (memory mapped), Not interleaved
(PCM_NON-INTERLEAVED)

buffer-sizes -- 64,128,256,512,1024,2048,8192 Samples

fragments -- 2

Hardware-pointer -- 2 Modi

The Card supports the readout of the actual Buffer-pointer, where DMA reads/writes. Since of the bulk mode of PCI it is only
64 Byte accurate. SO it is not really usable for the ALSA-mid-level functions (here the buffer-ID gives a better result), but if MMAP is used by the application. Therefore it can be configured at load-time with the parameter precise-pointer.

(Hint: Experimenting I found that the pointer is maximum 64 to large never to small. So if you subtract 64 you always have a safe pointer for writing, which is used on this mode inside ALSA. In theory now you can get now a latency as low as 16 Samples, which is a quarter of the interrupt possibilities.)

Precise Pointer -- off
interrupt used for pointer-calculation

Precise Pointer -- on
hardware pointer used.

Controller:

Since DSP-MADI-Mixer has 8152 Fader, it does not make sense to use the standard mixer-controls, since this would break most of (especially graphic) ALSA-Mixer GUIs. So Mixer control has be provided by a 2-dimensional controller using the hwdep-interface.

Also all 128+256 Peak and RMS-Meter can be accessed via the hwdep-interface. Since it could be a performance problem always copying and converting Peak and RMS-Levels even if you just need one, I decided to export the hardware structure, so that of needed some driver-guru can implement a memory-mapping of mixer or peak-meters over ioctl, or also to do only copying and no conversion. A test-application shows the usage of the controller.

Latency Controls --- not implemented !!!

Note: Within the windows-driver the latency is accessible of a control-panel, but buffer-sizes are controlled with ALSA from hwparams-calls and should not be changed in run-state, I did not

implement it here.

System Clock -- suspended !!!!

Name -- "System Clock Mode"

Access -- Read Write

Values -- "Master" "Slave"

!!!! This is a hardware-function but is in conflict with the Clock-source controller, which is a kind of ALSA-standard. I makes sense to set the card to a special mode (master at some frequency or slave), since even not using an Audio-application a studio should have working synchronisations setup. So use Clock-source-controller instead !!!!

Clock Source

Name -- "Sample Clock Source"

Access -- Read Write

Values -- "AutoSync", "Internal 32.0 kHz", "Internal 44.1 kHz", "Internal 48.0 kHz", "Internal 64.0 kHz", "Internal 88.2 kHz", "Internal 96.0 kHz"

Choose between Master at a specific Frequency and so also the Speed-mode or Slave (Autosync). Also see "Preferred Sync Ref"

!!!! This is no pure hardware function but was implemented by ALSA by some ALSA-drivers before, so I use it also. !!!

Preferred Sync Ref

Name -- "Preferred Sync Reference"

Access -- Read Write

Values -- "Word" "MADI"

Within the Auto-sync-Mode the preferred Sync Source can be chosen. If it is not available another is used if possible.

Note: Since MADI has a much higher bit-rate than word-clock, the
card should synchronise better in MADI Mode. But since the RME-PLL is very good, there are almost no problems with word-clock too. I never found a difference.

TX 64 channel ---

Name -- "TX 64 channels mode"

Access -- Read Write

Values -- 0 1

Using 64-channel-modus (1) or 56-channel-modus for MADI-transmission (0).

Note: This control is for output only. Input-mode is detected automatically from hardware sending MADI.

Clear TMS ---

Name -- "Clear Track Marker"

Access -- Read Write

Values -- 0 1

Don't use to lower 5 Audio-bits on AES as additional Bits.

Safe Mode oder Auto Input ---

Name -- "Safe Mode"

Access -- Read Write

Values -- 0 1

(default on)

If on (1), then if either the optical or coaxial connection has a failure, there is a takeover to the working one, with no sample failure. Its only useful if you use the second as a backup connection.

Input ---

Name -- "Input Select"

Access -- Read Write

Values -- optical coaxial

Choosing the Input, optical or coaxial. If Safe-mode is active, this is the preferred Input.

---------- Mixer ----------

Mixer

Name -- "Mixer"

Access -- Read Write

Values - <channel-number 0-127> <Value 0-65535>

Here as a first value the channel-index is taken to get/set the corresponding mixer channel, where 0-63 are the input to output fader and 64-127 the playback to outputs fader. Value 0 is channel muted 0 and 32768 an amplification of 1.

Chn 1-64

fast mixer for the ALSA-mixer utils. The diagonal of the mixer-matrix is implemented from playback to output.

Line Out

Name -- "Line Out"

Access -- Read Write

Values -- 0 1

Switching on and off the analog out, which has nothing to do with mixing or routing. the analog outs reflects channel 63,64.

--- information (only read access):

Sample Rate

Name -- "System Sample Rate"

Access -- Read-only

getting the sample rate.

External Rate measured

Name -- "External Rate"

Access -- Read only

Should be "Autosync Rate", but Name used is ALSA-Scheme. External Sample frequency liked used on Autosync is reported.

MADI Sync Status

Name -- "MADI Sync Lock Status"

Access -- Read

Values -- 0,1,2

MADI-Input is 0=Unlocked, 1=Locked, or 2=Synced.

Word Clock Sync Status

Name -- "Word Clock Lock Status"

Access -- Read

Values -- 0,1,2

Word Clock Input is 0=Unlocked, 1=Locked, or 2=Synced.

AutoSync

Name -- "AutoSync Reference"

Access -- Read

Values -- "WordClock", "MADI", "None"

Sync-Reference is either "WordClock", "MADI" or none.

RX 64ch --- noch nicht implementiert

MADI-Receiver is in 64 channel mode oder 56 channel mode.

AB_inp --- not tested

Used input for Auto-Input.

actual Buffer Position --- not implemented

!!! this is a ALSA internal function, so no control is used !!!

Calling Parameter:

index int array (min = 1, max = 8),
"Index value for RME HDSPM interface." card-index within ALSA

note: ALSA-standard

id string array (min = 1, max = 8),

"ID string for RME HDSPM interface."

note: ALSA-standard

enable int array (min = 1, max = 8),
"Enable/disable specific HDSPM sound-cards."

note: ALSA-standard

precise_ptr int array (min = 1, max = 8),
"Enable precise pointer, or disable."

note: Use only when the application supports this (which is a special case).

line_outs_monitor int array (min = 1, max = 8),
"Send playback streams to analog outs by default."

note: each playback channel is mixed to the same numbered output channel (routed). This is against the ALSA-convention, where all channels have to be muted on after loading the driver, but was used before on other cards, so i historically use it again)

enable_monitor int array (min = 1, max = 8),
"Enable Analog Out on Channel 63/64 by default."

note: here the analog output is enabled (but not routed).