Volume Control

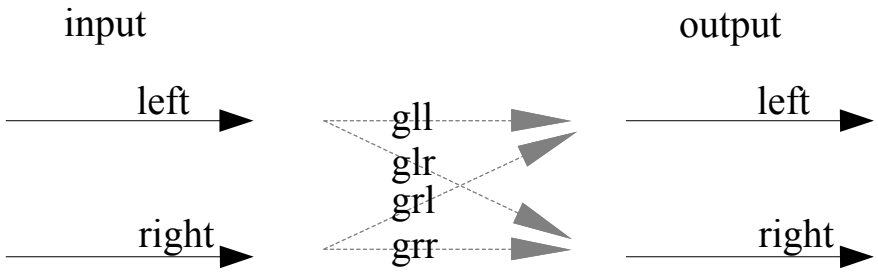
V1.0

Date	Version	Description	Authors
11/03/05	1	Description of the volume control EAP component	

Table of contents

1 Description

The aim of this component is to control the volume of each channel which indirectly implies pan and balance control. All this is controlled by 4 gains as shown in illustration 1:



output(left) = gll * input(left) + grl * input(right)
output(right) = glr * input(left) + grr * input(right)

Moreover the component has to handles mono/stereo conversion. Thus if the input signal is mono grl and grr aren't used and if the output signal is mono glr and grr aren't used.

2 Implementation

2.1 gain conversion

The four gains are given in dB in Q8 representation by the user. It means that gains belong to the interval [-128dB,128dB[. Since the user can't give negative parameters (because of hamaca implementation) the values given by the user are translated of 128dB. The gains are thus given in Q8 unsigned representation ([0db,256dB[).
For example:
We have Q8_ONE = 1<<8 : one in Q8 representation
128dB = 0x8000 in Q8 unsigned representation
if we want a gain of -3.2db the command will be : command = (unsigned int)((-3.2 + 128) * Q8_ONE)
Note : a command of 0 won't be interpreted as a gain of -128dB but as a gain of -infinity dB eg a linear gain of 0.

the conversion between dB and linear gain is done as fallow:
the conversion rule is :

$$20\log(gainl) = gaindB$$
$$gainl = 10^{\frac{gaindB}{20}} = LinearGain(gaindB)$$

we have : $LinearGain(6.02) = 2$
so $LinearGain(x) = LinearGain(x' + y \times 6.02) = LinearGain(x') \times 2^{y} \qquad with\ x' \in [0\,;6.02](dB)$

2^{y} is only a left or right shift according *y* sign, so the dB to linear conversion is confined in the definition of the*LinearGain* function between 0 and 6.02dB. This is done with a conversion table. This table is computed at compilation time according the value of NB_STEP which determines the number of entries. To find the index of the table entry associated to x', x' is multiplied by $STEPFACT = \frac{(NBSTEP-1)}{6.02}$

Since *gainl* may be greater than 1, linear gains are stored with a fractional value and a shift associated.

2.2 gain smoothing

In order to avoid zip noise, the volume command can't be applied immediately it has to be smoothed. To do so, a first order recursive filter is used :

$$gain_{applied}(n) = alpha \times gain_{applied}(n-1) + (1-alpha) \times gain_{asked}$$

where alpha is the response time and is set by the user. The gains are smoothed at every samples until the differences between the gains applied and the gains asked are lower than VOLCTRL_DIFF_MAX % (VOLCTRL_DIFF_MAX = 2).

2.3 sample length

The volctrl component can handle input of 16 or 24 bits. It always give 24 bits samples.

3 EAP parameters

There are no static parameters:

```
typedef struct {
        t_ha_uint16 iReserved0;
        t_ha_uint16 iReserved1;
} t_eap_volctrl_static_params;
```

There are 6 dynamic parameters for the volume control component :

```
typedef struct{
        t_ha_uint16 inb_output_chan;
        t_ha_uint16 igll;
        t_ha_uint16 iglr;
        t_ha_uint16 igrl;
        t_ha_uint16 igrr;
        t_ha_uint16 ialpha;
        t_ha_uint16 iReserved0;
        t_ha_uint16 iReserved1;
} t_eap_volctrl_dynamic_params;
```

- **inb_output_chan**: determines the number of channels of the output signal (1 = mono, 2 = stereo).
  This parameter is used with the field *nb_chan* of EAP packet so that we can select the running mode between the four possibles : mono -> mono / mono -> stereo / stereo ->mono / stereo -> stereo. (default is stereo -> stereo)
- **igll**: gain of the input left channel onto the output left channel. The value of the gain must be give in dB translated of 128dB in Q8 unsigned representation. It means that to give a gain of 2.3dB the command is (2.3 + 128)*(1<<8). A value of 0 means a command of -infinity dB. (default is 0dB)
- **iglr**: gain of the input left channel onto the output right channel (if it exists). The value fallow the same rules as igll. (default is –infinity dB)
- **igrl**: gain of the input right channel (if it exists) onto the output left channel. The value fallow the same rules as igll. (default is –infinity dB)
- **igrr**: gain of the input right channel (if it exists) onto the output left channel (if it exists). The value fallow the same rules as igll. (default is 0 dB)
- **ialpha**: Value of the time constant for gain smoothing. (default value is 0x7F00 = 0.992188)

4 Standalone executable

To compile the standalone executable do make unix (or dsp) in*audioeffect/libvolctrl/standalone/ directory*.
The arguments of standalone executable are : <input_file> <output_file> <param_file>.
The param_file looks like :

nb_bit_in: 16;	size of input samples (if nb_bit_in = 24 the 16 bit samples read from the input file are put in 24 bits before the call to volctrl)
nb_channel_in: 2;	number of channels of input file
nb_channel_out: 2;	number of channels of output file
gll: 3.0;	value of gll
glr: -128.0;	value of glr NB: -128 <=> -infinity dB
grl: -128.0;	value of grl
grr: -3.0;	value of grr
alpha: 0x7F00;	value of time constant for smoothing
block_size: 64;	EAP block size