

Audio Language Compiler Documentation

Stefan Jaritz

July 18, 2013

Contents

1	Introduction	1
2	the program body	2
2.1	the "main" function	3
2.2	global and local variable declarations	3
3	arithmetic and boolean operations on variables	4
3.1	\geq	4
3.2	$==$	5
3.3	\leq	5
3.4	$!=$	5
3.5	$\%$	5
3.6	$+$	5
3.7	$*$	6
3.8	$-$	6
3.9	$/$	6
3.10	$=$	7
3.11	$<$	7
3.12	$+=$	7
3.13	$>$	7
4	special operations on variables	8
4.1	delay	8
4.2	compressor	9
4.3	biquad	9
4.3.1	highShelv	9
4.3.2	coeff	10
4.3.3	lowShelv	10
4.3.4	Peak	11
4.3.5	LP	11
4.3.6	HP	11

4.4	sampleIO	12
4.4.1	read	12
4.4.2	write	12
4.5	noisegate	12
4.6	limiter	13
4.7	ui	13
4.7.1	dim	13
4.7.2	initDisplay	14
4.7.3	initButton	14
4.7.4	print	14
4.7.5	initPanel	15
4.7.6	LED	15
4.7.7	initLED	15
4.7.8	checkButton	16
4.8	expander	16

Chapter 1

Introduction

The Audio Language is designed as a imperative textual domain specific language. It contains structures for sequences, selections, repetitions and Procedural programming. In the following chapters the different aspects of programming with the Audio Language are explained.

Parts of this handbook are generated automatic.

Chapter 2

the program body

Describing an audio device with the Audio Language can be structured into tree parts. The first parts explains what piece of hardware should run what part of the description. The second part defines so called "global variables". These variables are accessible in every part of the program and can be exchanged between parts of hardware running parts of the description. The third part are functions which are describing the functionality of the audio device. The following listing gives a short example of the "frame" an Audio Language program:

```
1 global {
2     rational r;
3 }
4
5 main (myAudioDevPart1) {
6     local {
7     }
8     code {
9         r = 1;
10    }
11 }
12
13 main (myAudioDevPart2) {
14     local {
15     }
16     code {
17         r = 21;
18    }
19 }
```

This program divides the description of the audio device into two parts. There is one global variable "r" defined. Putting this into the Audio Language Compiler it will produces two files "myAudioDevPart1.alasm" and "myAudioDevPart2.alasm". The following sections will describe the "main" Function syntax and the variable declaration syntax.

2.1 the "main" function

2.2 global and local variable declarations

Chapter 3

arithmetic and boolean operations on variables

Two variables can be connected via operator. Sometimes a third variable can receive the result. To simplify the use of the operators the order of the variables connected by operators can be varied.

Syntax for 3 variables: $\langle c \rangle = \langle a \rangle \langle \text{operator} \rangle \langle b \rangle$;

Example: $c = a + b$;

Syntax for 2 variables: $\langle a \rangle \langle \text{operator} \rangle \langle b \rangle$

Example: $\text{if } (a < b) \{ \dots \}$

Legend:

symbol	explanation
v	variable
i	index of the variable
c	constant value
c/v	constant value or variable

3.1 $>=$

a	b	HAL function
integer	integer	compareIntegerMoreEqual (v/c) a, (i) a, (v/c) b, (i) b;
rational	rational	compareRationalMoreEqual (v/c) a, (i) a, (v/c) b, (i) b;

3.2 ==

a	b	HAL function
integer	integer	compareIntegerEqual (v/c) a, (i) a, (v/c) b, (i) b;
rational	rational	compareRationalEqual (v/c) a, (i) a, (v/c) b, (i) b;

3.3 <=

a	b	HAL function
integer	integer	compareIntegerLessEqual (v/c) a, (i) a, (v/c) b, (i) b;
rational	rational	compareRationalLessEqual (v/c) a, (i) a, (v/c) b, (i) b;

3.4 !=

a	b	HAL function
integer	integer	compareIntegerNEqual (v/c) a, (i) a, (v/c) b, (i) b;
rational	rational	compareRationalNEqual (v/c) a, (i) a, (v/c) b, (i) b;

3.5 %

c	a	b	HAL function
integer	integer	integer	modInteger (v) c, (v/c) a, (v/c) b;
rational	rational	rational	modRational (v) c, (v/c) a, (v/c) b;

3.6 +

c	a	b	HAL function
integer	integer	integer	addInteger (v) c, (v/c) a, (v/c) b;
rational	rational	rational	addRational (v) c, (v/c) a, (v/c) b;

3.7 *

c	a	b	HAL function
rational	expander	rational	calcExpander (v) c, (v) a, (v/c) b;
rational	rational	compressor	calcCompressor (v) c, (v) b, (v/c) a;
rational	delay	rational	calcDelay (v) c, (v) a, (v/c) b;
rational	rational	expander	calcExpander (v) c, (v) b, (v/c) a;
rational	noisegate	rational	calcNoisegate (v) c, (v) a, (v/c) b;
rational	rational	noisegate	calcNoisegate (v) c, (v) b, (v/c) a;
rational	rational	limiter	calcLimiter (v) c, (v) b, (v/c) a;
rational	biquad	rational	convoluteBiquad (v) c, (v) a, (v/c) b;
integer	integer	integer	mulInteger (v) c, (v/c) a, (v/c) b;
rational	limiter	rational	calcLimiter (v) c, (v) a, (v/c) b;
rational	rational	delay	calcDelay (v) c, (v) b, (v/c) a;
rational	compressor	rational	calcCompressor (v) c, (v) a, (v/c) b;
rational	rational	biquad	convoluteBiquad (v) c, (v) b, (v/c) a;
rational	rational	rational	mulRational (v) c, (v/c) a, (v/c) b;

3.8 -

c	a	b	HAL function
integer	integer	integer	subInteger (v) c, (v/c) a, (v/c) b;
rational	rational	rational	subRational (v) c, (v/c) a, (v/c) b;

3.9 /

c	a	b	HAL function
integer	integer	integer	divInteger (v) c, (v/c) a, (v/c) b;
rational	rational	rational	divRational (v) c, (v/c) a, (v/c) b;

3.10 =

a	b	HAL function
string	rational	rationalToString (v) a, (i) a, (v/c) b, (i) b;
string	integer	integerToString (v) a, (i) a, (v/c) b, (i) b;
integer	integer	assignInteger (v) a, (v/c) b;
string	string	assignString (v) a, (i) a, (v/c) b, (i) b;
rational	rational	assignRational (v) a, (v/c) b;

3.11 <

a	b	HAL function
integer	integer	compareIntegerLess (v/c) a, (i) a, (v/c) b, (i) b;
rational	rational	compareRationalLess (v/c) a, (i) a, (v/c) b, (i) b;

3.12 +=

a	b	HAL function
rational	rational	addRational (v) a, (v/c) b, (v/c) a;
string	string	concatStrings (v) a, (i) a, (v/c) b, (i) b;
integer	integer	addInteger (v) a, (v/c) b, (v/c) a;

3.13 >

a	b	HAL function
integer	integer	compareIntegerMore (v/c) a, (i) a, (v/c) b, (i) b;
rational	rational	compareRationalMore (v/c) a, (i) a, (v/c) b, (i) b;

Chapter 4

special operations on variables

Special operations are used to map HAL functions directly into the Audio Language. For better useability they are grouped into "operation classes". Some parameters are optional and have default values.

Syntax: <operation class>[<parameter 1>=<value 1>,<parameter 1>=<value 1>, ...]

Example: sampleIO[operation=write ,src=out, channel=1];

Explanation: From the sample class "sampleIO" the operation "write" is chosen. As source "src" the variable out is used and the channel is set to 1.

4.1 delay

name	HAL type	I/O type	default
obj	delay	reference to variable	—
N	integer	reference to variable	—
Nindx	integer	constant value	0
offset	integer	constant value	—

example:

delay[obj=<ref obj>, N=<ref N>, Nindx=0, offset=constant];

mapped to Audio Language Assembler:

initDelay obj, N, Nindx, offset;

4.2 compressor

name	HAL type	I/O type	default
obj	compressor	reference to variable	—
TAV	rational	constant value	—
AT	rational	constant value	—
RT	rational	constant value	—
CT	rational	constant value	—
CS	rational	constant value	—

example:

compressor[obj=<ref obj>, TAV=constant, AT=constant, RT=constant, CT=constant, CS=constant];

mapped to Audio Language Assembler:

initCompressor obj, TAV, AT, RT, CT, CS;

4.3 biquad

4.3.1 highShelv

name	HAL type	I/O type	default
obj	biquad	reference to variable	—
index	integer	constant value	0
fs	rational	reference to variable	—
fc	rational	reference to variable	—
q	rational	reference to variable	—
g	rational	reference to variable	—

example:

biquad[type=highShelv, obj=<ref obj>, index=0, fs=<ref fs>, fc=<ref fc>, q=<ref q>, g=<ref g>];

mapped to Audio Language Assembler:

initBiquadAsHighFreqShelvFilter obj, index, fs, fc, q, g;

4.3.2 coeff

name	HAL type	I/O type	default
obj	biquad	reference to variable	—
index	integer	constant value	0
a0	rational	reference to variable	0.0
a1	rational	reference to variable	1.0
a2	rational	reference to variable	0.0
b1	rational	reference to variable	0.0
b2	rational	reference to variable	0.0

example:

biquad[type=coeff, obj=<ref obj>, index=0, a0=<ref a0>, a1=<ref a1>, a2=<ref a2>, b1=<ref b1>, b2=<ref b2>];

mapped to Audio Language Assembler:

initBiquad obj, index, a0, a1, a2, b1, b2;

4.3.3 lowShelv

name	HAL type	I/O type	default
obj	biquad	reference to variable	—
index	integer	constant value	0
fs	rational	reference to variable	—
fc	rational	reference to variable	—
q	rational	reference to variable	—
g	rational	reference to variable	—

example:

biquad[type=lowShelv, obj=<ref obj>, index=0, fs=<ref fs>, fc=<ref fc>, q=<ref q>, g=<ref g>];

mapped to Audio Language Assembler:

initBiquadAsLowFreqShelvFilter obj, index, fs, fc, q, g;

4.3.4 Peak

name	HAL type	I/O type	default
obj	biquad	reference to variable	—
index	integer	constant value	0
fs	rational	reference to variable	—
fc	rational	reference to variable	—
q	rational	reference to variable	—
g	rational	reference to variable	—

example:

biquad[type=Peak, obj=<ref obj>, index=0, fs=<ref fs>, fc=<ref fc>, q=<ref q>, g=<ref g>];

mapped to Audio Language Assembler:

initBiquadAsPeakFilter obj, index, fs, fc, q, g;

4.3.5 LP

name	HAL type	I/O type	default
obj	biquad	reference to variable	—
index	integer	constant value	0
fs	rational	reference to variable	—
fc	rational	reference to variable	—

example:

biquad[type=LP, obj=<ref obj>, index=0, fs=<ref fs>, fc=<ref fc>];

mapped to Audio Language Assembler:

initBiquadAsLP obj, index, fs, fc;

4.3.6 HP

name	HAL type	I/O type	default
obj	biquad	reference to variable	—
index	integer	constant value	0
fs	rational	reference to variable	—
fc	rational	reference to variable	—

example:

biquad[type=HP, obj=<ref obj>, index=0, fs=<ref fs>, fc=<ref fc>];

mapped to Audio Language Assembler:

initBiquadAsHP obj, index, fs, fc;

4.4 sampleIO

4.4.1 read

name	HAL type	I/O type	default
channel	integer	constant value	—
dest	rational	reference to variable	—
wait	integer	constant value	1

example:

sampleIO[operation=read, channel=constant, dest=<ref dest>, wait=1];

mapped to Audio Language Assembler:

readSampleFrame channel, dest, wait;

4.4.2 write

name	HAL type	I/O type	default
channel	integer	constant value	—
src	rational	reference to variable	—

example:

sampleIO[operation=write, channel=constant, src=<ref src>];

mapped to Audio Language Assembler:

writeSampleFrame channel, src;

4.5 noisegate

name	HAL type	I/O type	default
obj	noisegate	reference to variable	—
TAV	rational	constant value	—
AT	rational	constant value	—
RT	rational	constant value	—
NT	rational	constant value	—
NS	rational	constant value	—

example:

noisegate[obj=<ref obj>, TAV=constant, AT=constant, RT=constant, NT=constant, NS=constant];

mapped to Audio Language Assembler:
initNoisegate obj, TAV, AT, RT, NT, NS;

4.6 limiter

name	HAL type	I/O type	default
obj	limiter	reference to variable	—
ATpeak	rational	constant value	—
RTpeak	rational	constant value	—
ATsmooth	rational	constant value	—
RTsmooth	rational	constant value	—
LT	rational	constant value	—
LS	rational	constant value	—

example:

limiter[obj=<ref obj>, ATpeak=constant, RTpeak=constant, ATsmooth=constant, RTsmooth=constant, LT=constant, LS=constant];

mapped to Audio Language Assembler:

initLimiter obj, ATpeak, RTpeak, ATsmooth, RTsmooth, LT, LS;

4.7 ui

4.7.1 dim

name	HAL type	I/O type	default
x	integer	constant value	—
y	integer	constant value	—
xle	integer	constant value	—
yle	integer	constant value	—
obj	None	reference to variable	—
index	integer	constant value	—

example:

ui[func=dim, x=constant, y=constant, xle=constant, yle=constant, obj=<ref obj>, index=constant];

mapped to Audio Language Assembler:

uiSetDim x, y, xle, yle, obj, index;

4.7.2 initDisplay

name	HAL type	I/O type	default
uuid	integer	constant value	—
obj	display	reference to variable	—
index	integer	constant value	0
parent	panel	reference to variable	—
pIndex	integer	constant value	0

example:

```
ui[func=initDisplay, uuid=constant, obj=<ref obj>, index=0, parent=<ref parent>, pIndex=0];
```

mapped to Audio Language Assembler:

```
uiInitDisplay uuid, obj, index, parent, pIndex;
```

4.7.3 initButton

name	HAL type	I/O type	default
uuid	integer	constant value	—
obj	button	reference to variable	—
index	integer	constant value	0
parent	panel	reference to variable	—
pIndex	integer	constant value	0

example:

```
ui[func=initButton, uuid=constant, obj=<ref obj>, index=0, parent=<ref parent>, pIndex=0];
```

mapped to Audio Language Assembler:

```
uiInitButton uuid, obj, index, parent, pIndex;
```

4.7.4 print

name	HAL type	I/O type	default
obj	display	reference to variable	—
index	integer	constant value	0
text	string	reference to variable	—
tindex	integer	constant value	0

example:

```
ui[func=print, obj=<ref obj>, index=0, text=<ref text>, tindex=0];
```

mapped to Audio Language Assembler:

uiSetDisplay obj, index, text, tindex;

4.7.5 initPanel

name	HAL type	I/O type	default
uuid	integer	constant value	—
obj	panel	reference to variable	—
index	integer	constant value	0

example:

ui[func=initPanel, uuid=constant, obj=<ref obj>, index=0];

mapped to Audio Language Assembler:

uiInitPanel uuid, obj, index;

4.7.6 LED

name	HAL type	I/O type	default
obj	LED	reference to variable	—
index	integer	constant value	0
on	integer	constant value	1

example:

ui[func=LED, obj=<ref obj>, index=0, on=1];

mapped to Audio Language Assembler:

uiSetLED obj, index, on;

4.7.7 initLED

name	HAL type	I/O type	default
uuid	integer	constant value	—
obj	LED	reference to variable	—
index	integer	constant value	0
parent	panel	reference to variable	—
pIndex	integer	constant value	0

example:

ui[func=initLED, uuid=constant, obj=<ref obj>, index=0, parent=<ref parent>, pIndex=0];

mapped to Audio Language Assembler:

uiInitLED uuid, obj, index, parent, pIndex;

4.7.8 checkButton

name	HAL type	I/O type	default
obj	button	reference to variable	—
index	integer	constant value	0

example:

ui[func=checkButton, obj=<ref obj>, index=0];

mapped to Audio Language Assembler:

uiCheckButtonPressed obj, index;

4.8 expander

name	HAL type	I/O type	default
obj	expander	reference to variable	—
TAV	rational	constant value	—
AT	rational	constant value	—
RT	rational	constant value	—
ET	rational	constant value	—
ES	rational	constant value	—

example:

expander[obj=<ref obj>, TAV=constant, AT=constant, RT=constant, ET=constant, ES=constant];

mapped to Audio Language Assembler:

initExpander obj, TAV, AT, RT, ET, ES;