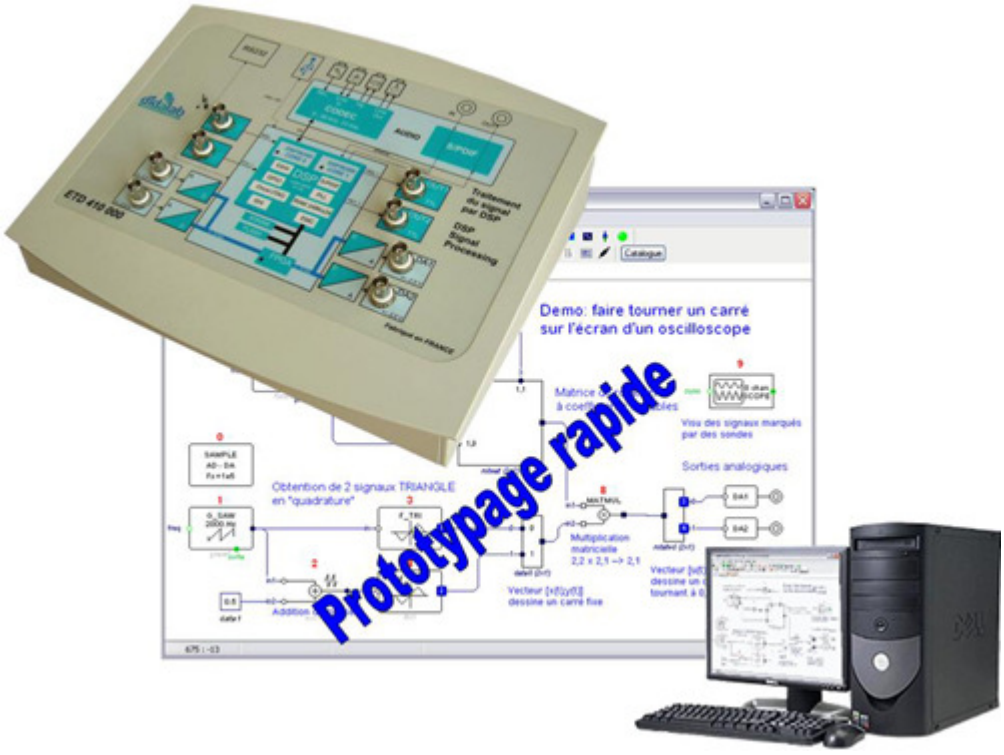
TRAITEMENT DU SIGNAL PAR D.S.P.



FIBULA G : TRAITEMENT DU SIGNAL EN TEMPS REEL

Bibliothèque
Exemple d'utilisation
Manuel de référence

DIDALAB SARL au capital de 1.000.000 € RCS VERSAILLES 494764269 SIRET 49476426900028 N° TVA : FR80494764269
Zac La clé St Pierre 5 rue du groupe Manoukian 78990 ELANCOURT
Tél. : +33 (0)1 30 66 08 88 Fax : +33 (0)1 30 66 72 20

FIBULA-G

Block Library

Reference Manual

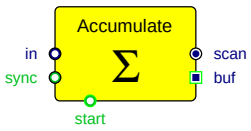
08/10/2010

FIBULA-G Block Library

ACCUM

Accumulate random signals

ACCUM



CATEGORY: Stat

DESCRIPTION:
Accumulate random signals
Sync resets buffer pointer to 0. Start clears buffer.
Fract output is a continuous scan of accumulate buffer

PARAMETERS:

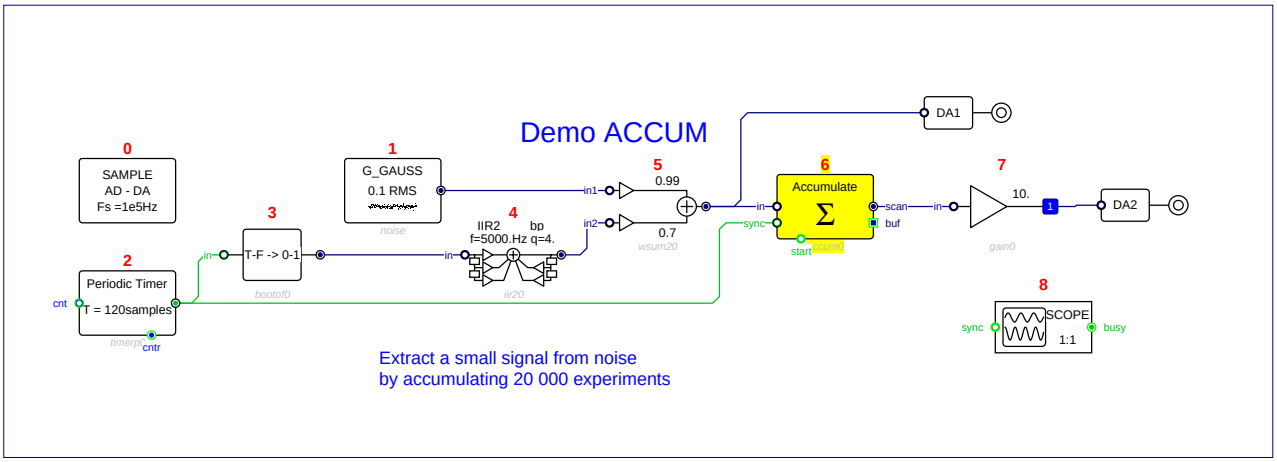
Parameter:	*Default values:*
Points	500
Number of adds	10000

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory
name_sync	BOOL	BIT	mandatory
name_start	BOOL	BIT	optional

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_scan	FRACT	WORD	normal
name_buf	FRACT	Matrix of WORD	optional

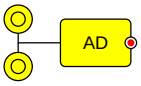


ACCUM test program

AD

Analog to Digital Converters 1:2 complex result

AD



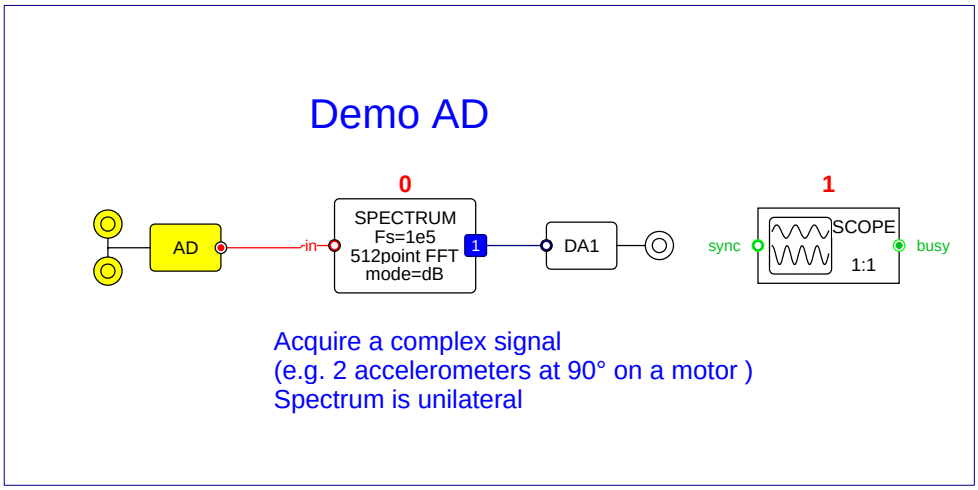
CATEGORY: Analog InOut

DESCRIPTION:
Analog to Digital Converters 1:2 complex result

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	COMPLEX	WORD	normal

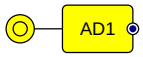
ATTRIBUTES
Non executable, Unique,



AD test program

Result of Analog to Digital Converter 1

AD1



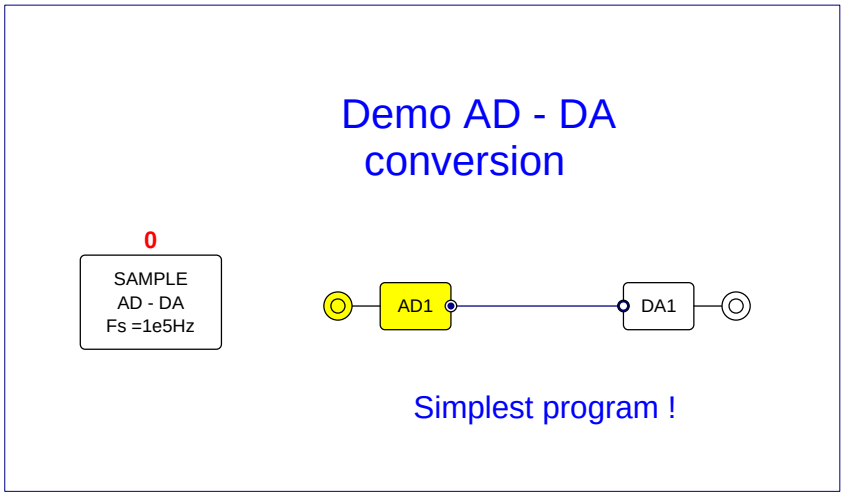
CATEGORY: Analog InOut

DESCRIPTION:
Result of Analog to Digital Converter 1

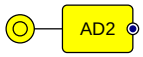
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

ATTRIBUTES
Non executable, Unique,



AD1 test program



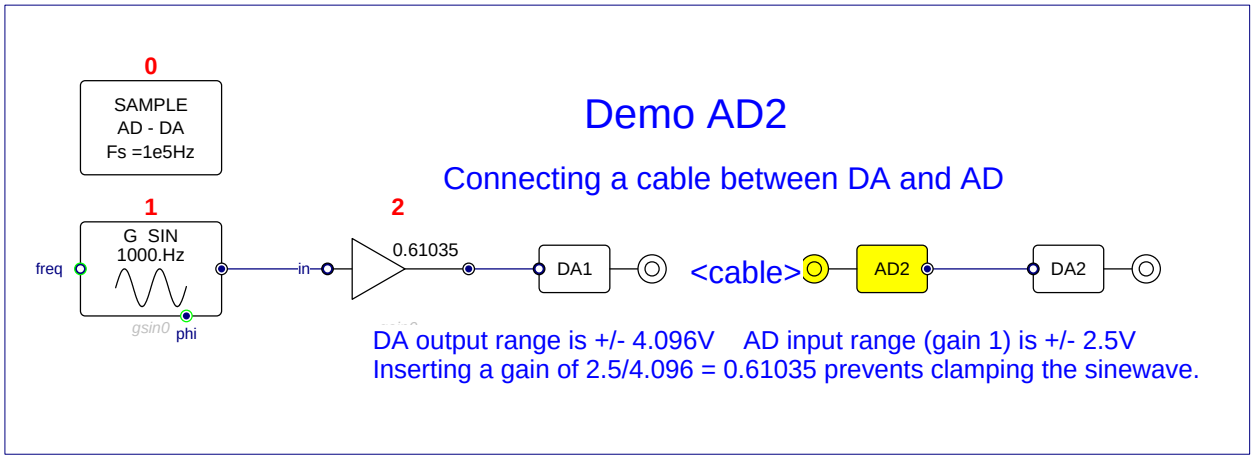
CATEGORY: Analog InOut

DESCRIPTION:
Result of Analog to Digital Converter 2

OUTPUTS

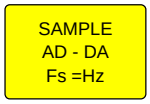
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

ATTRIBUTES
Non executable, Unique,



AD2 test program

Wait sample, read ADs, write DAs



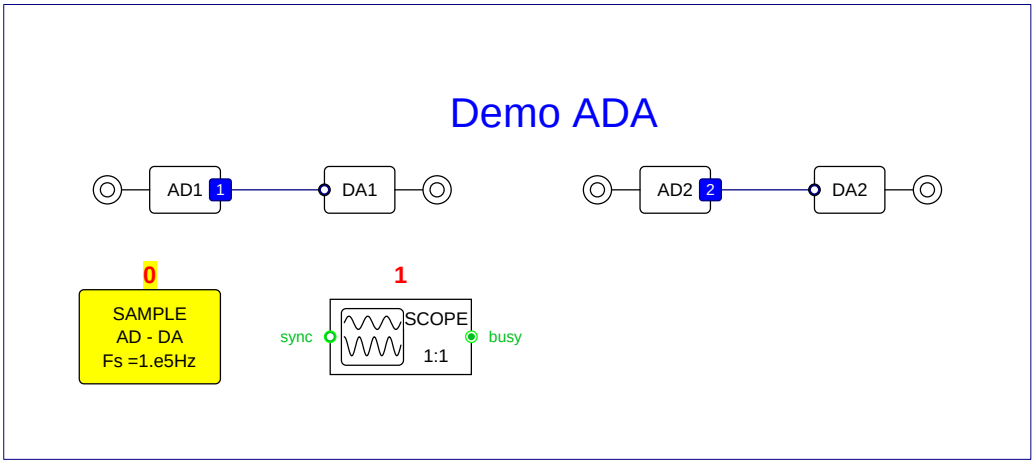
CATEGORY: Analog InOut

DESCRIPTION:
Wait sample, read ADs, write DAs
Defines actual_fs

PARAMETERS:

Parameter:	*Default values:*
Frequency (Hz)	1e5

ATTRIBUTES
Unique, Execute First, Defines: actual_fs



ADA test program

ADC

12 bit AD conversion

ADC

DESCRIPTION:
12 bit AD conversion
LTC1403 connected on SDI2_1 SCKR,FSR

PARAMETERS:

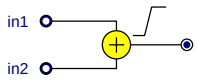
Parameter:	*Default values:*
Sampling frequency:	1E5

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

ATTRIBUTES
Unique,

Addition with saturation



CATEGORY: Arithmetic

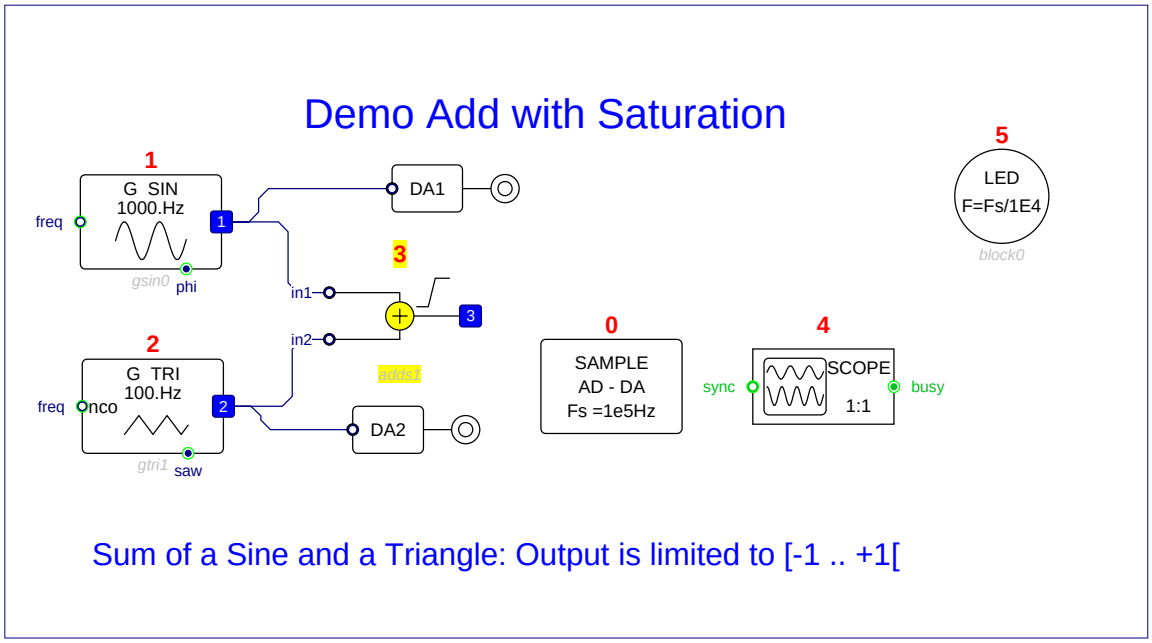
DESCRIPTION:
Addition with saturation

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in1	FRACT	WORD	mandatory
name_in2	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

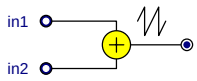


ADDS test program

ADDV

Add with overflow

ADDV



CATEGORY: Arithmetic

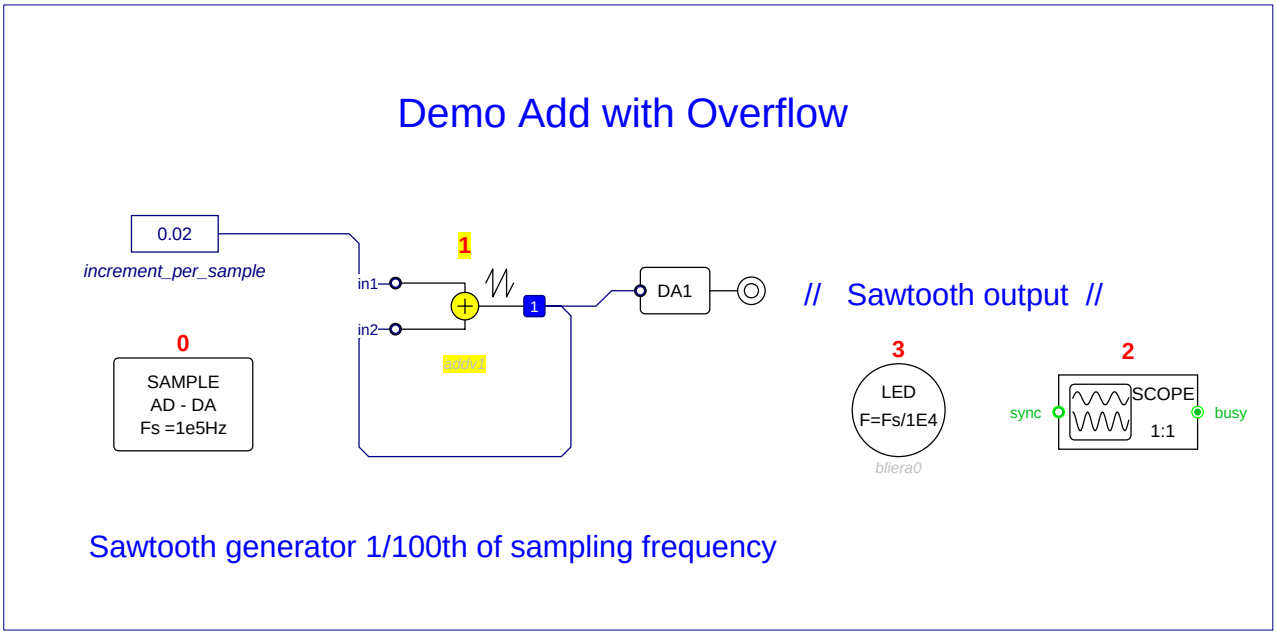
DESCRIPTION:
Add with overflow
y = in1 + in2;
if y>=1 then y=y-2; if y<-1 then y=y+2;

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in1	FRACT	WORD	mandatory
name_in2	FRACT	WORD	mandatory

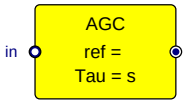
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



ADDV test program

Automatic Gain Control



CATEGORY: Audio

DESCRIPTION:
Automatic Gain Control

PARAMETERS:

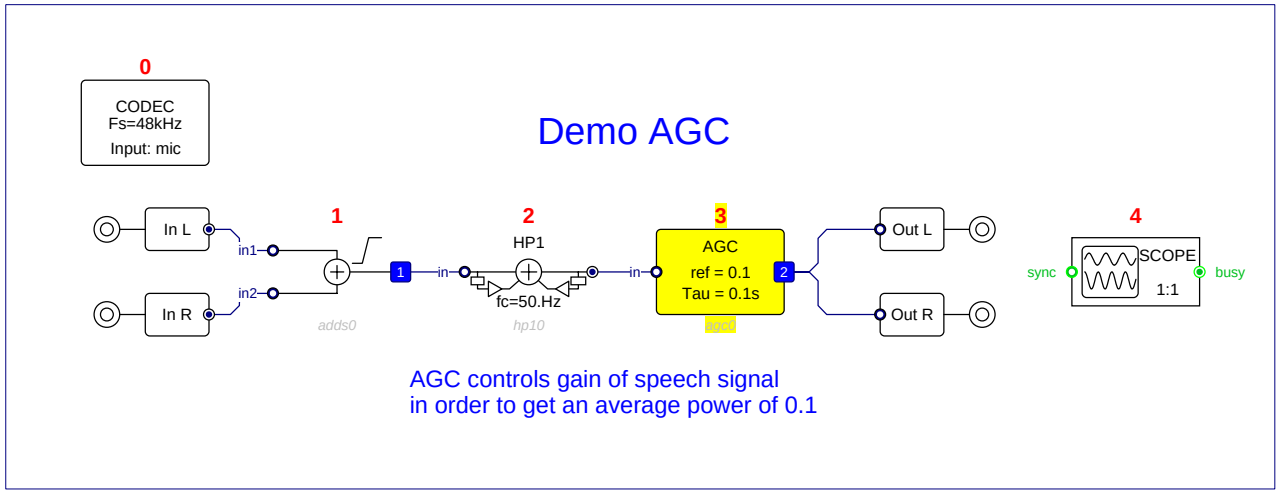
Parameter:	*Default values:*
Reference	0.75
Time Ct	5.

INPUTS

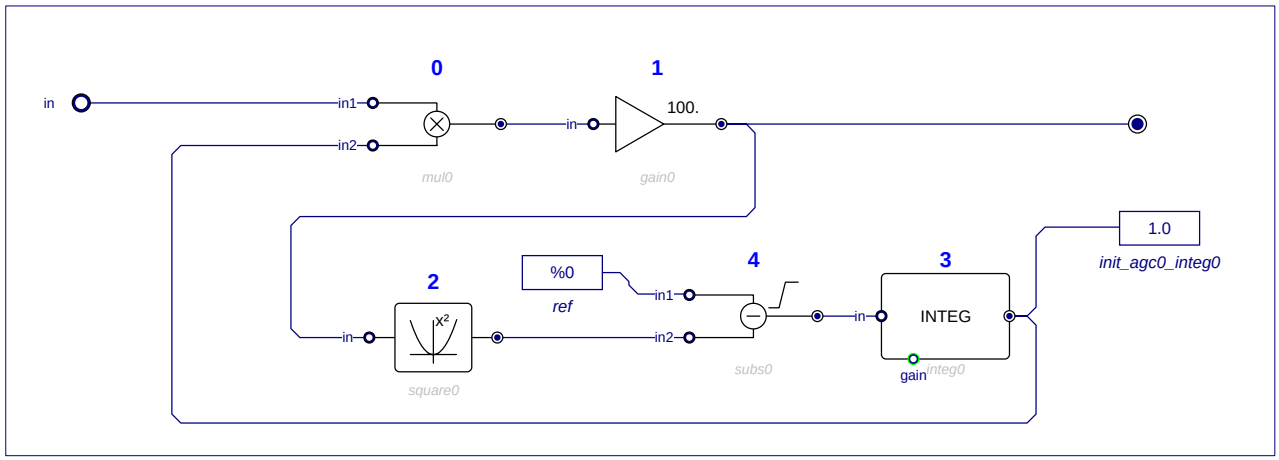
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

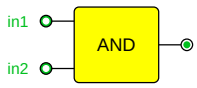


AGC test program



AGC internal schema

ANDGATE Logic AND function y = in1 & in2 ANDGATE



CATEGORY: Logic

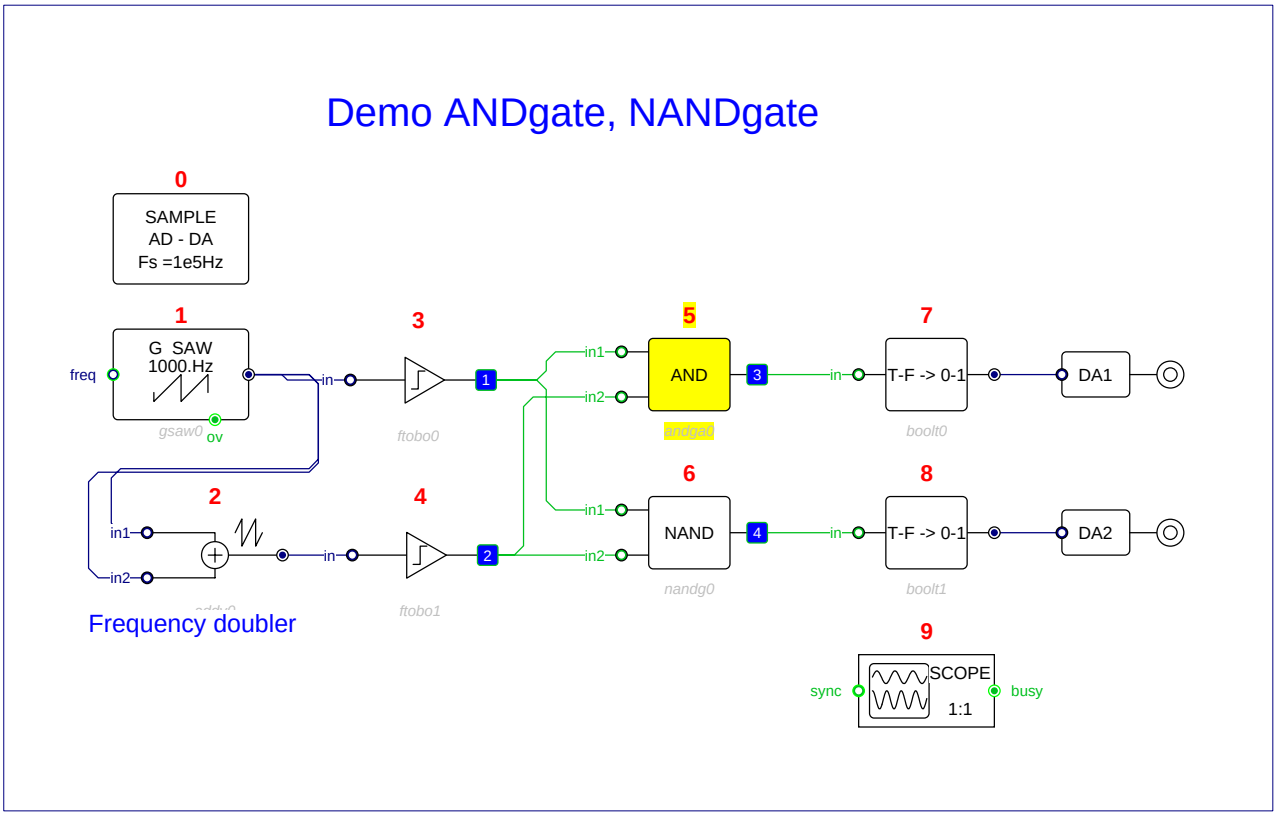
DESCRIPTION:
Logic AND function y = in1 & in2

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in1	BOOL	BIT	mandatory
name_in2	BOOL	BIT	mandatory

OUTPUTS

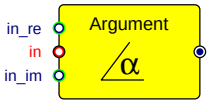
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	BOOL	BIT	normal



ANDGATE test program

Argument of a complex input

ARG



CATEGORY: Functions

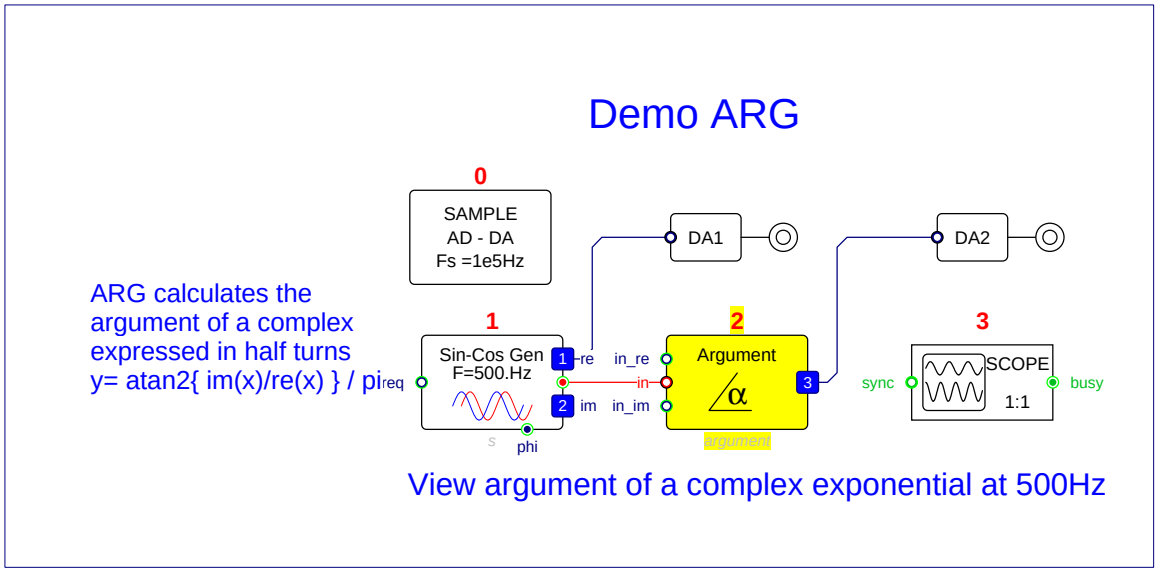
DESCRIPTION:
Argument of a complex input
y = 1/pi * arctan2(Im(x),Re(x))

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	COMPLEX	WORD	mandatory
name_in_re	FRACT	WORD	optional
name_in_im	FRACT	WORD	optional

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

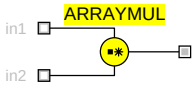


ARG test program

ARRAYMUL

Array Multiply

ARRAYMUL



CATEGORY: Matrix

DESCRIPTION:
Array Multiply
Out(i) = in1(i)*in2(i)

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in1	defined by cn	Matrix of	mandatory
name_in2	defined by cn	Matrix of	mandatory

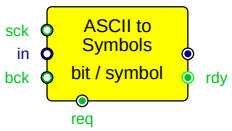
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	defined by cn	Matrix of	normal

ASCTOSYM

ASCII to symbols

ASCTOSYM



CATEGORY: Telecom

DESCRIPTION:
ASCII to symbols
Outputs an n-bit symbol on sck true, then negates sck.
Reads ascii on bck true then negates bck.
Asserts req when shift register < n bit

PARAMETERS:

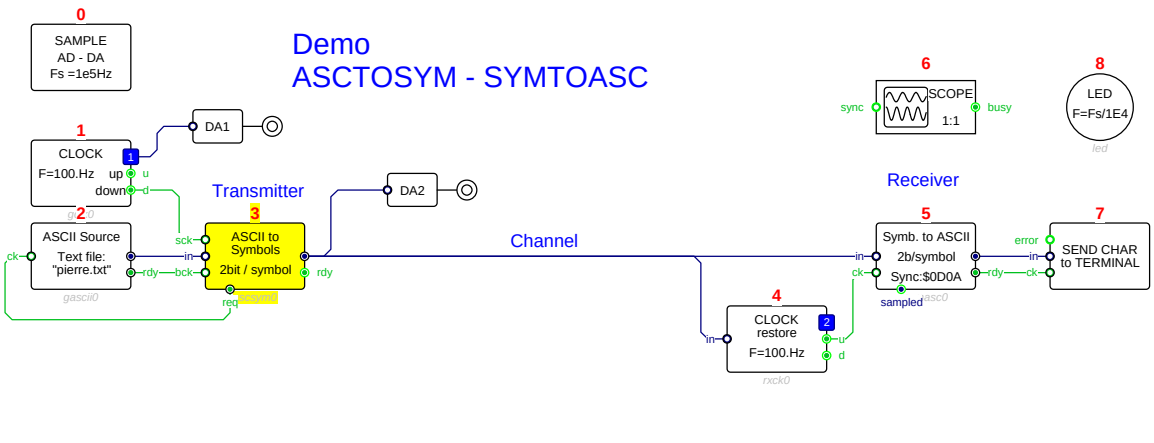
Parameter:	*Default values:*
Bits per symbol	1

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory
name_bck	BOOL	BIT	mandatory
name_sck	BOOL	BIT	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal
name_rdy	BOOL	BIT	optional
name_req	BOOL	BIT	normal



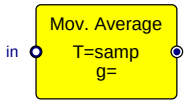
Transmitter: Convert a text to a string of 1-bit symbols sent at 100 Bauds.
Receiver: concatenate symbols to ASCII characters and send characters to serial port.

ASCTOSYM test program

AVERAGE

Moving average

AVERAGE



CATEGORY: Filters

DESCRIPTION:
Moving average
y(k) = (g/n) sum(xi), i= [k-n+1 ... k]

PARAMETERS:

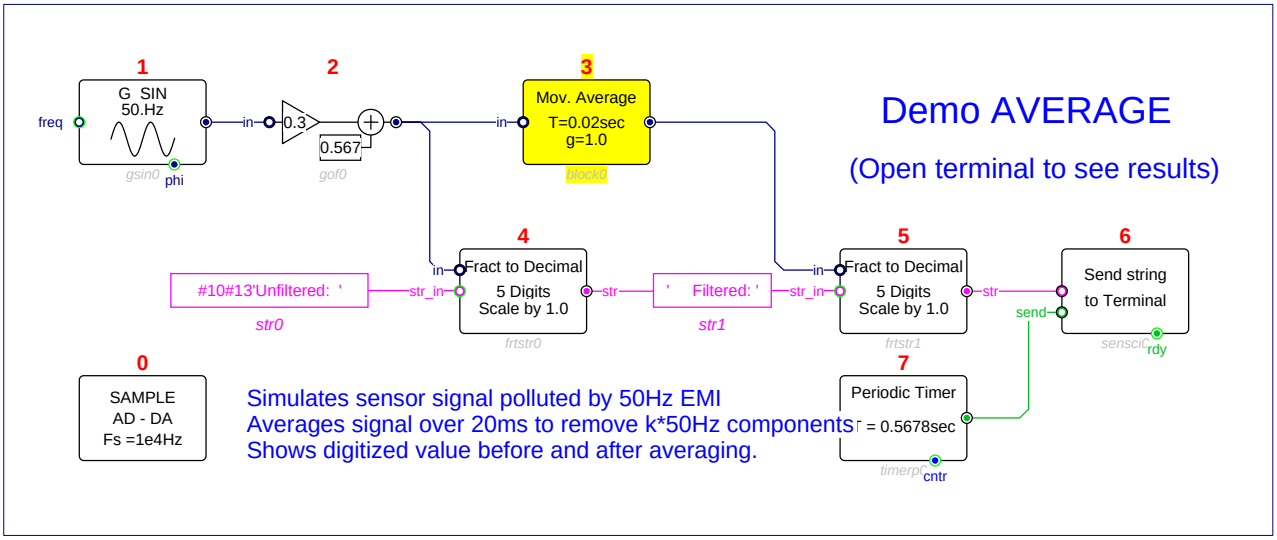
Parameter:	*Default values:*
gain	1.0
time	0.02
Unit	sec,samp

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



AVERAGE test program

BAD_CHAN

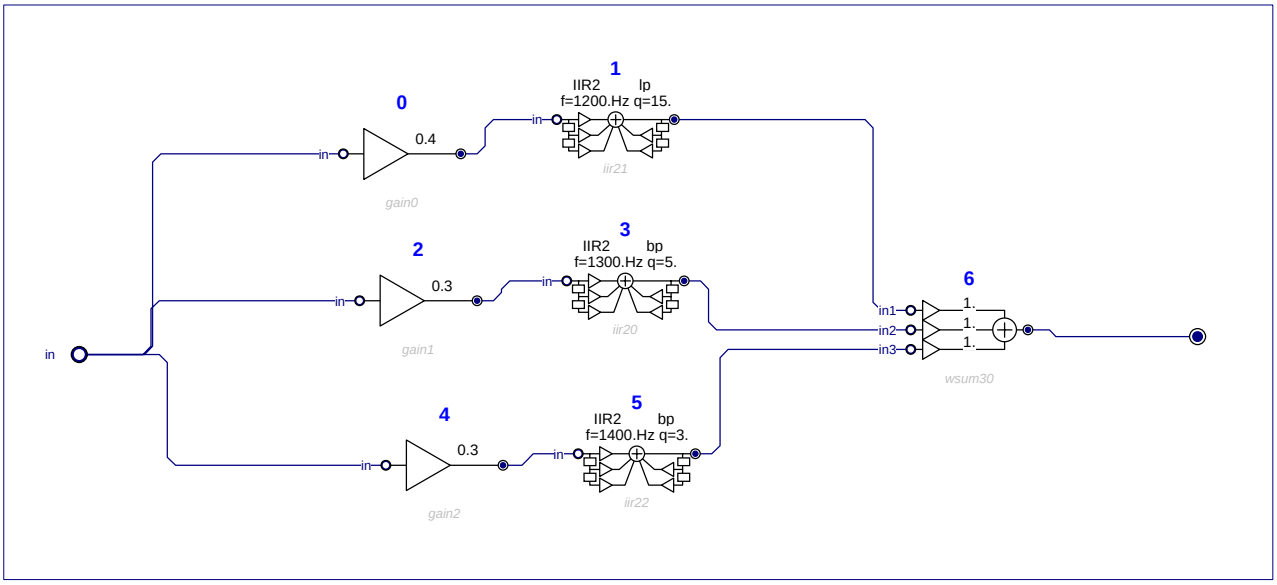
BAD_CHAN

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

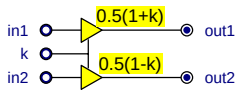
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



BAD_CHAN internal schema

Balance

BAL



CATEGORY: Audio

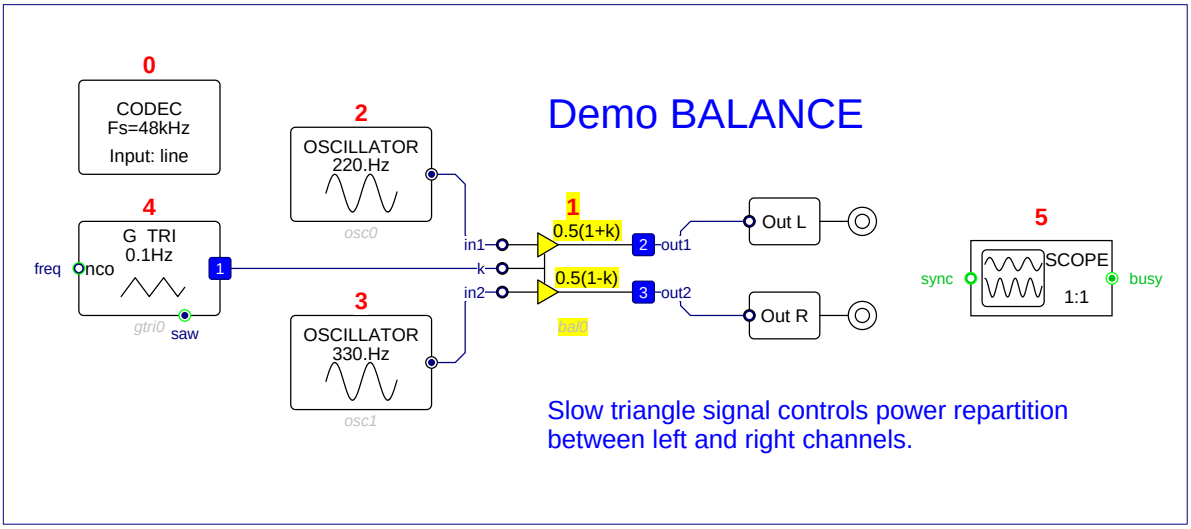
DESCRIPTION:
Balance
out1 = in1*(1+k)/2
out2 = in2*(1-k)/2

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in1	FRACT	WORD	mandatory
name_k	FRACT	WORD	mandatory
name_in2	FRACT	WORD	mandatory

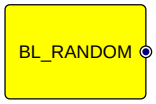
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_out1	FRACT	WORD	normal
name_out2	FRACT	WORD	normal



BAL test program

BL_RANDOM Band limited random BL_RANDOM

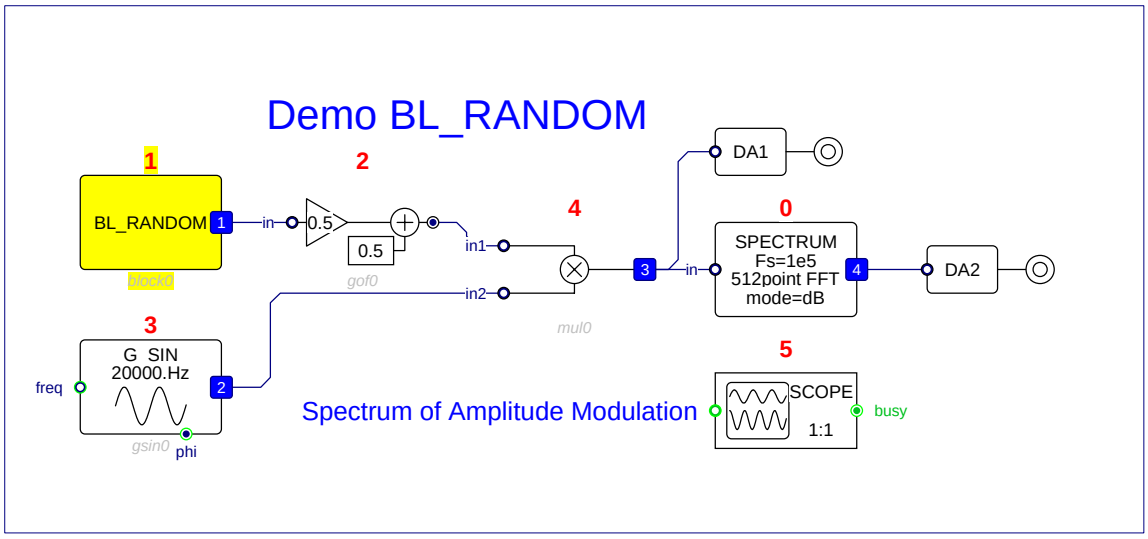


CATEGORY: Telecom

DESCRIPTION:
Band limited random
Filter 200-4500Hz. Simulates voice spectrum

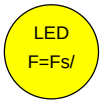
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



BL_RANDOM test program

BLINKER_D Led clignotante indiquant l'activité d'un coeur BLINKER_D

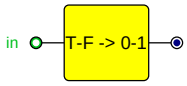


DESCRIPTION:
Led clignotante indiquant l'activité d'un coeur

PARAMETERS:

Parameter:	*Default values:*
Period (*Ts)	1E5

BOOLTOF Boolean to fractional conversion BOOLTOF



CATEGORY: Control

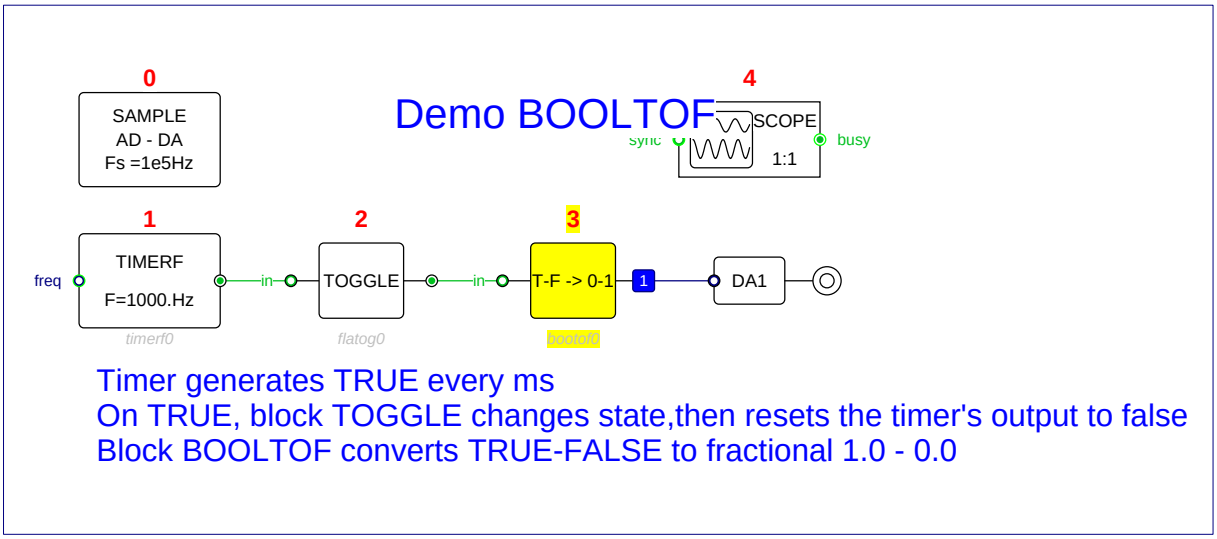
DESCRIPTION:
Boolean to fractional conversion
T - F --> 1.0 - 0.0

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	BOOL	BIT	mandatory

OUTPUTS

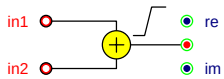
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



BOOLTOF test program

Complex Addition

CADD



DESCRIPTION:
Complex Addition
with saturation

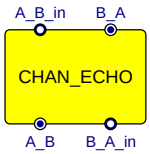
INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in1	COMPLEX	WORD	mandatory
name_in2	COMPLEX	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	COMPLEX	WORD	optional
name_re	FRACT	WORD	optional
name_im	FRACT	WORD	optional

CHAN_ECHO CHAN_ECHO



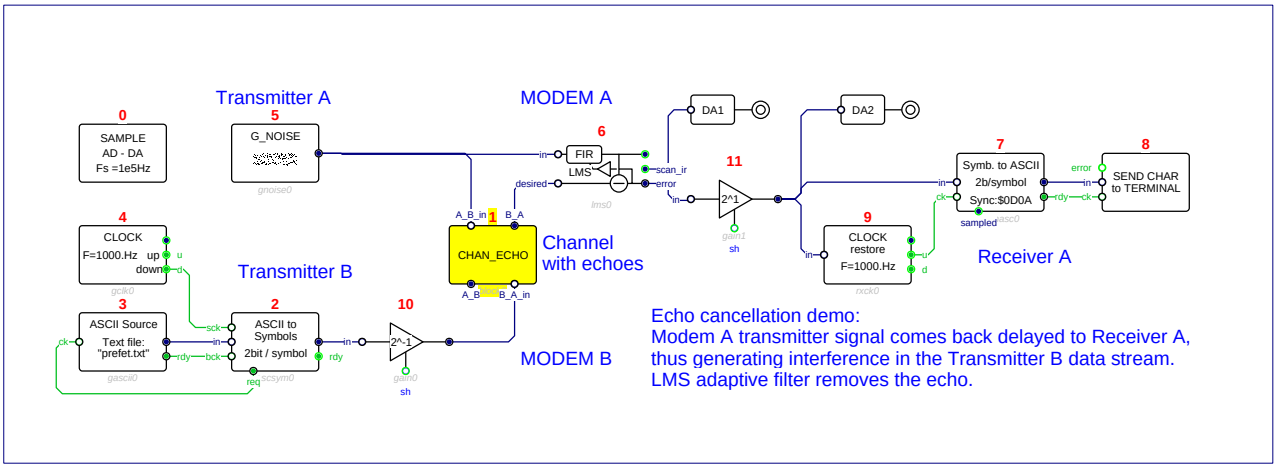
CATEGORY: Telecom

INPUTS

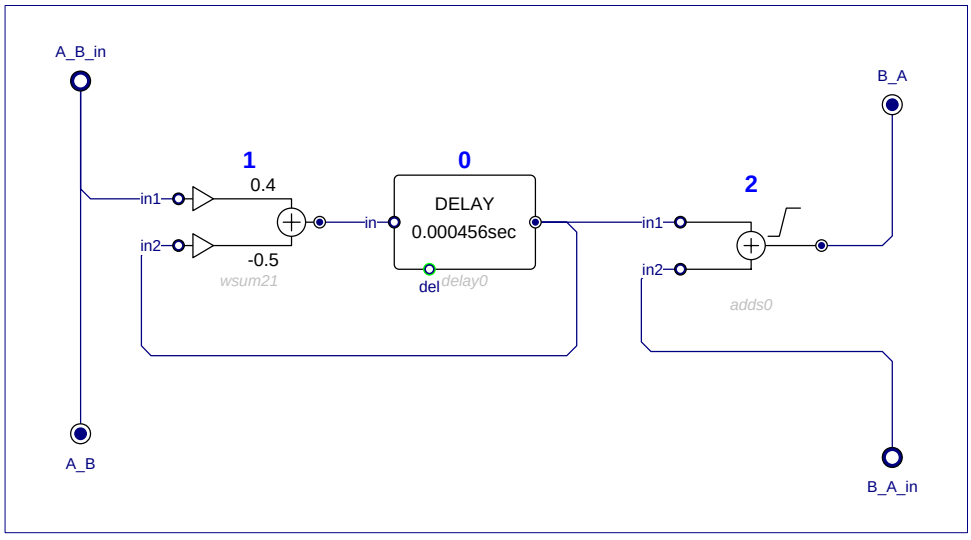
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_A_B_in	FRACT	WORD	mandatory
name_B_A_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_B_A	FRACT	WORD	normal
name_A_B	FRACT	WORD	normal



CHAN_ECHO test program



CHAN_ECHO internal schema

CHAN_RINGING LP channel w. resonance CHAN_RINGING

DESCRIPTION:
LP channel w. resonance

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

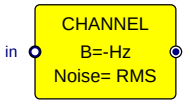
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

CHANNEL

Channel simulation

CHANNEL



CATEGORY: Telecom

DESCRIPTION:
Channel simulation
Bandpass and noisy transmission channel

PARAMETERS:

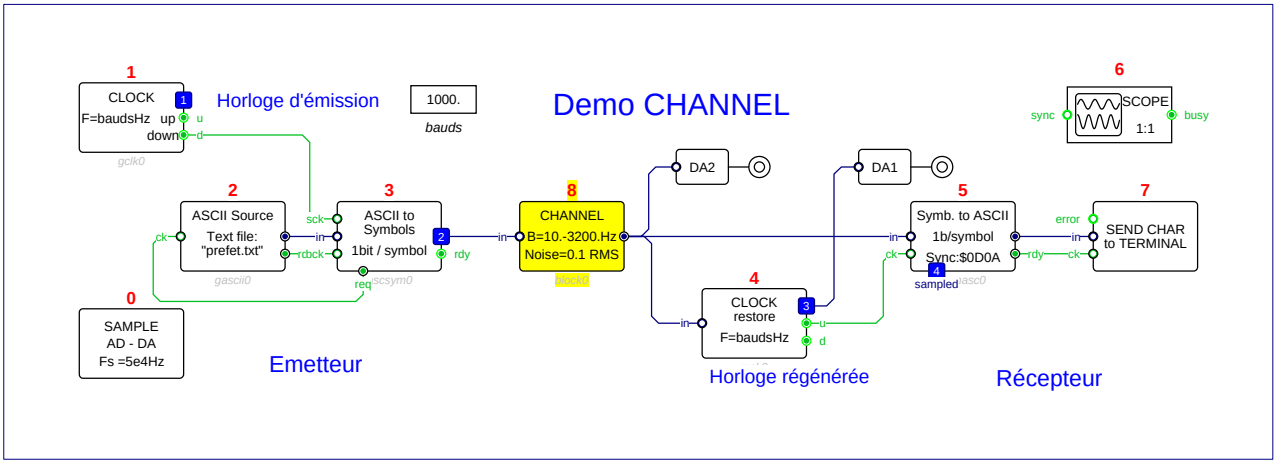
Parameter:	*Default values:*
Freq min:	200.
Freq max:	3200
Noise RMS:	0.

INPUTS

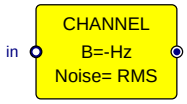
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

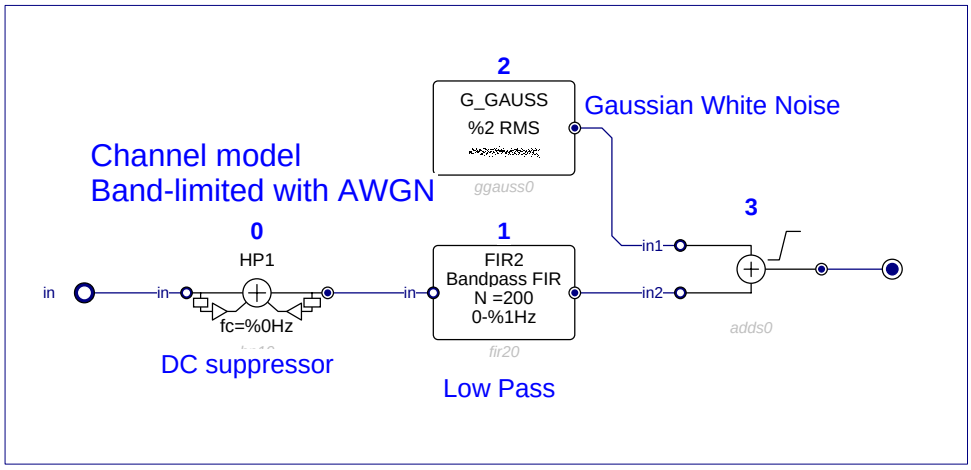
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



CHANNEL test program



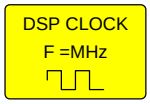


CHANNEL internal schema

CLOCK

Change DSP clock frequency

CLOCK



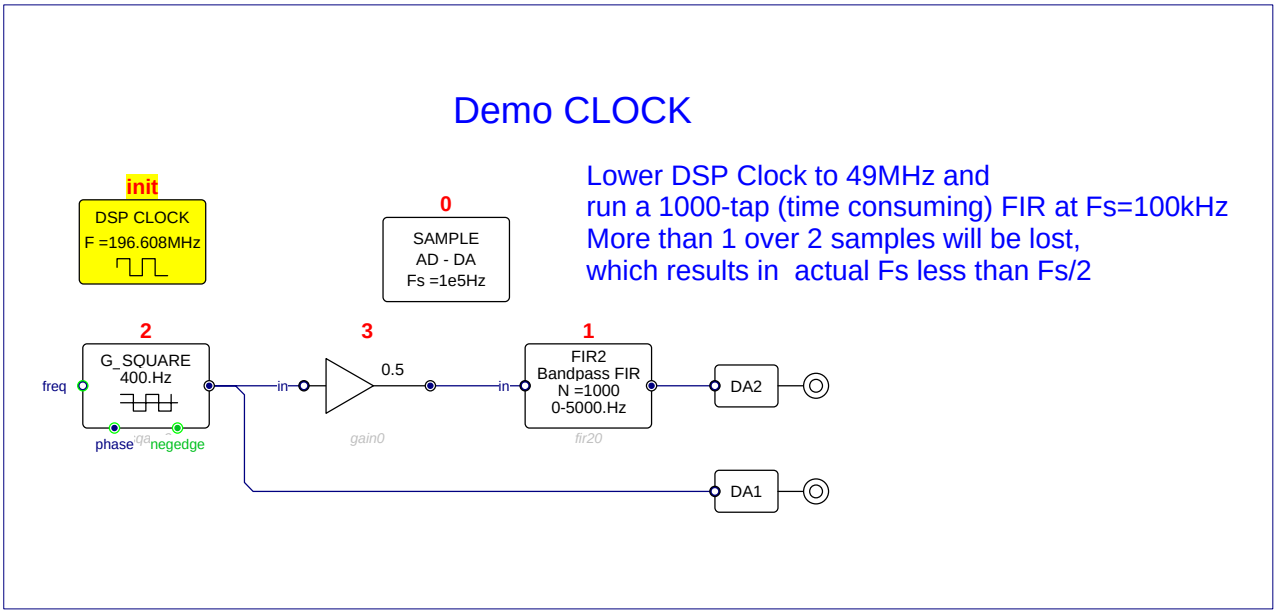
CATEGORY: Timing

DESCRIPTION:
Change DSP clock frequency
192kHz --> 200.7MHz
by reprogramming the PLL

PARAMETERS:

Parameter:	*Default values:*
Frequency (MHz)	196.608,200.704,175.104,150.528,98.304,49.152,24.576,12.288,6.144,3.072,1.536,0.768,0.384,0.192

ATTRIBUTES
Execute at Init, Unique,

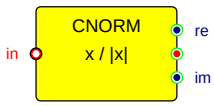


CLOCK test program

CNORM

Norm a complex variable

CNORM



CATEGORY: Control

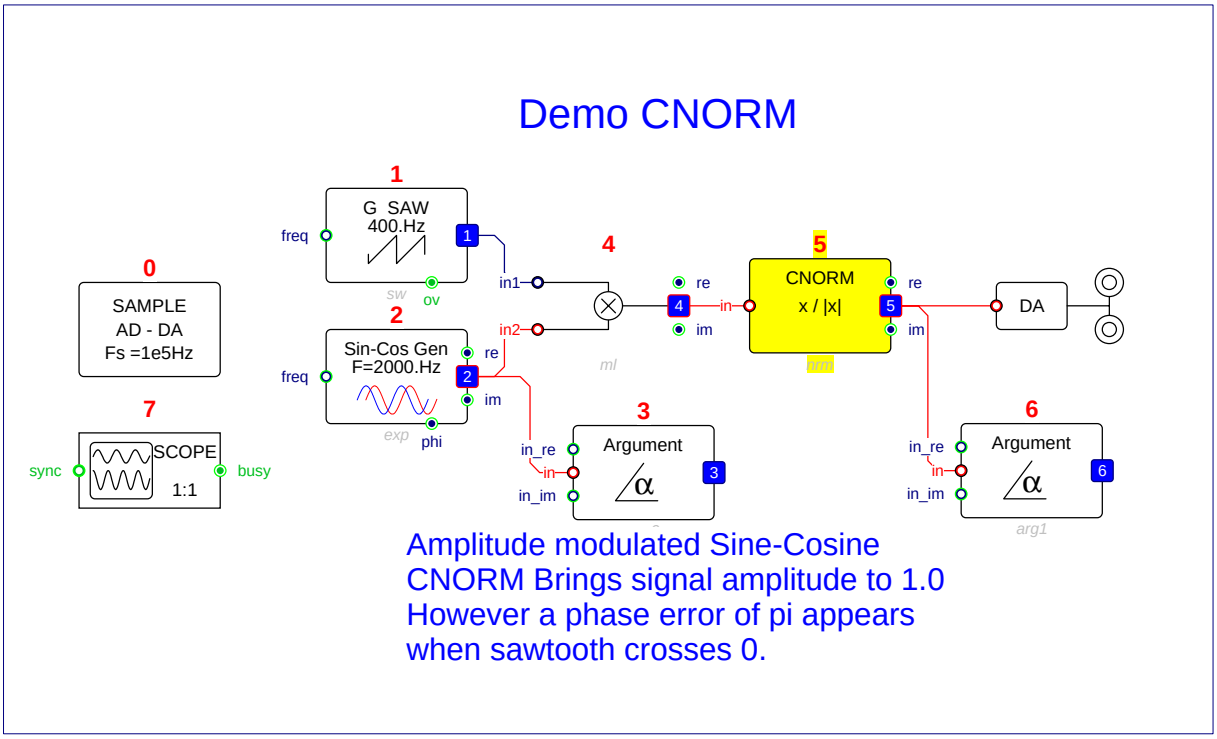
DESCRIPTION:
Norm a complex variable
z= x / |x|

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	COMPLEX	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	COMPLEX	WORD	optional
name_re	FRACT	WORD	optional
name_im	FRACT	WORD	optional



CNORM test program

1-3 Repetition coder

COD_R3D

CATEGORY: Telecom

DESCRIPTION:
1-3 Repetition coder
y(k) = x(k):x(k-n):x(k-2n)

PARAMETERS:

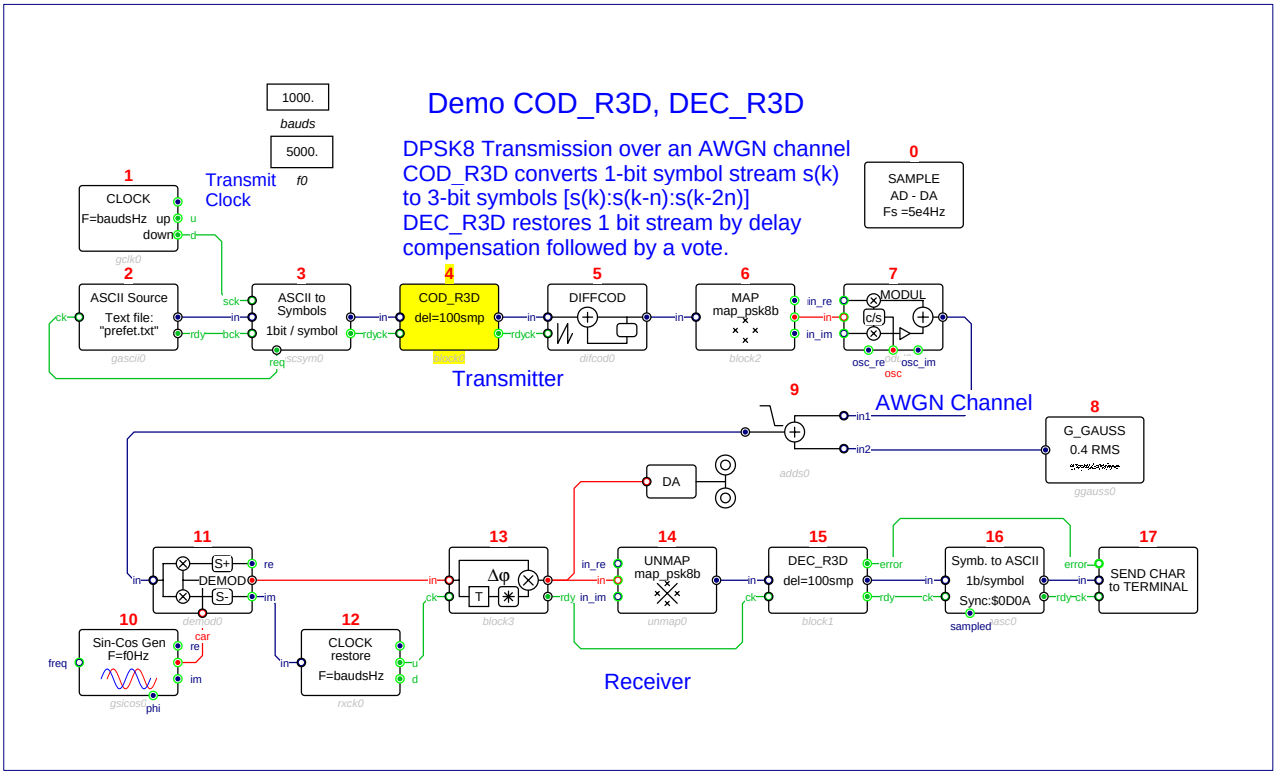
Parameter:	*Default values:*
Delay (samples)	100

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory
name_ck	BOOL	BIT	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal
name_rdy	BOOL	BIT	optional

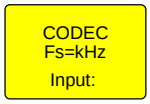


COD_R3D test program

CODEC

Audio CODEC

CODEC



CATEGORY: Audio

DESCRIPTION:
Audio CODEC
Type tlv320aic23

PARAMETERS:

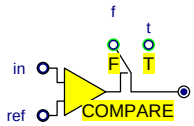
Parameter:	*Default values:*
Fs (kHz)	8,32,48,96
Input	line,mic

ATTRIBUTES
Unique, Execute First, Defines: actual_fs

COMPARE

Relais function

COMPARE



CATEGORY: Non linear

DESCRIPTION:
Relais function
Output is oTrue or oFalse, wether
(name_in - name_ref) is >, >=,=,<=,< to 0

PARAMETERS:

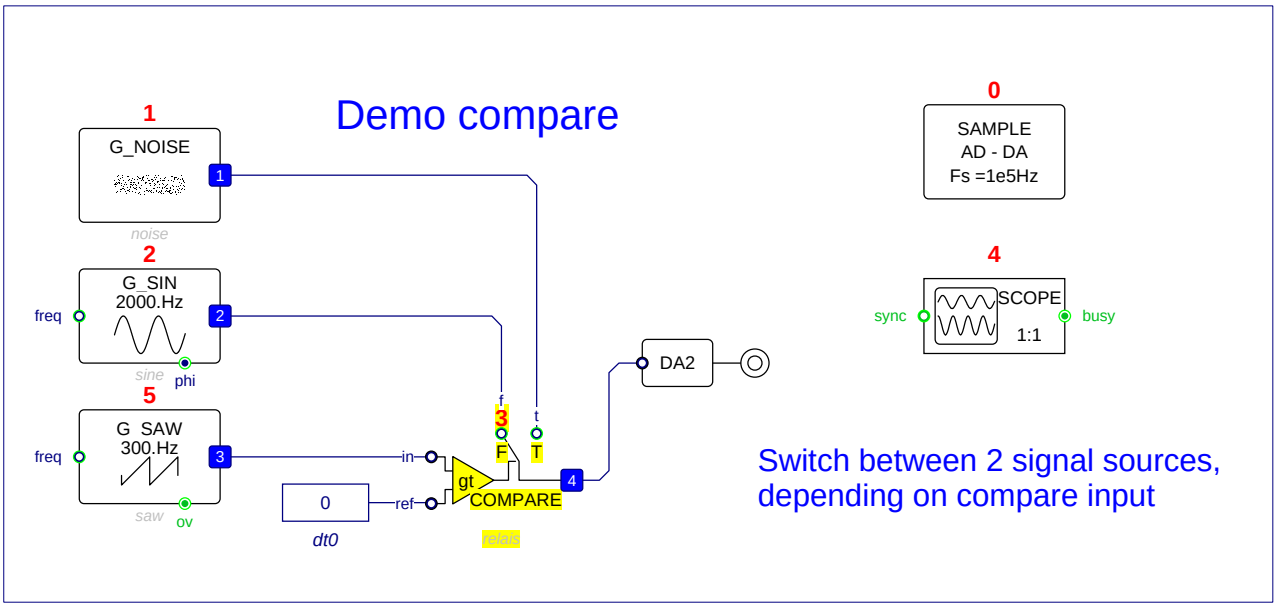
Parameter:	*Default values:*
condition	gt,ge,eq,ne,le,lt
oTrue	1.
oFalse	0

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory
name_ref	FRACT	WORD	mandatory
name_t	FRACT	WORD	optional
name_f	FRACT	WORD	optional

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

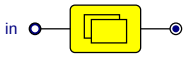


COMPARE test program

COPY

Copy data to different address

COPY



CATEGORY: Arithmetic

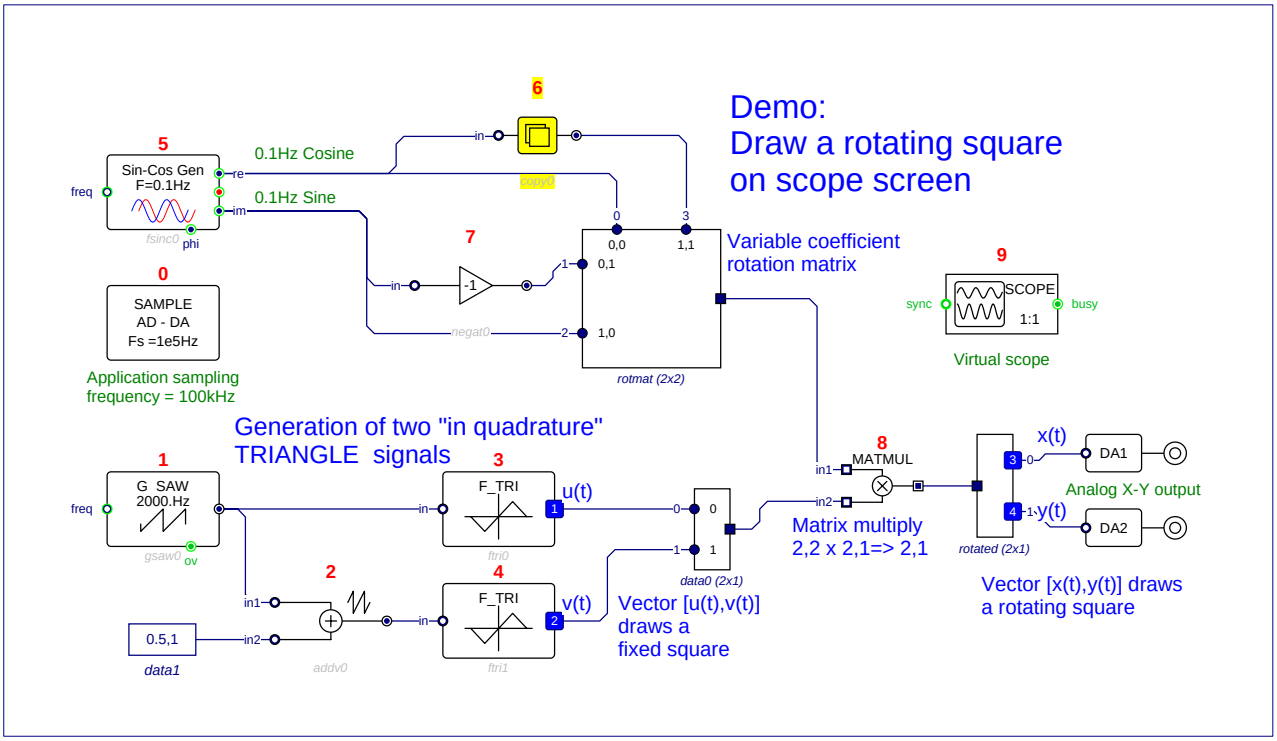
DESCRIPTION:
Copy data to different address

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

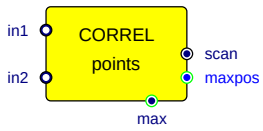
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



COPY test program

Cross correlation

CORREL



CATEGORY: Stat

DESCRIPTION:
Cross correlation
between Input1 and Input 2
Fract output is a continuous scan of correlation function

PARAMETERS:

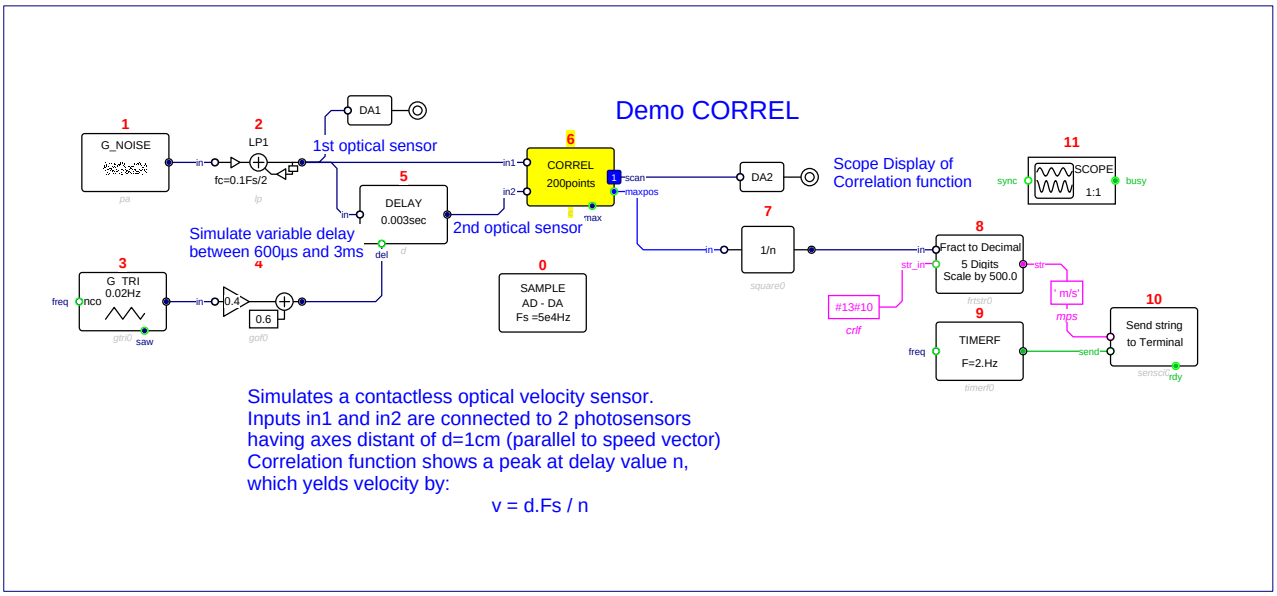
Parameter:	*Default values:*
Points	100
epsi	1e-4
Causal / Non Causal	c,nc

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in1	FRACT	WORD	mandatory
name_in2	FRACT	WORD	mandatory

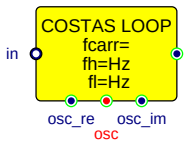
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_scan	FRACT	WORD	normal
name_max	FRACT	WORD	optional
name_maxpos	INTEGER	WORD	optional



CORREL test program

COSTAS loop



CATEGORY: Telecom

DESCRIPTION:
COSTAS loop
for ASK demodulation and carrier recovery.

PARAMETERS:

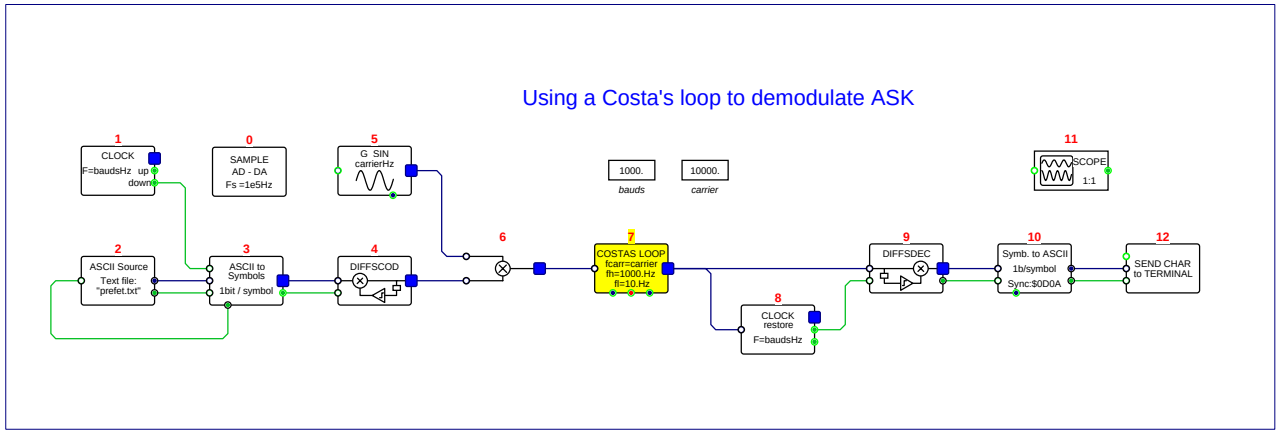
Parameter:	*Default values:*
Fcarr	5000.
Fhi	5000.
Flo	1.

INPUTS

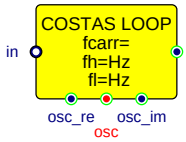
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

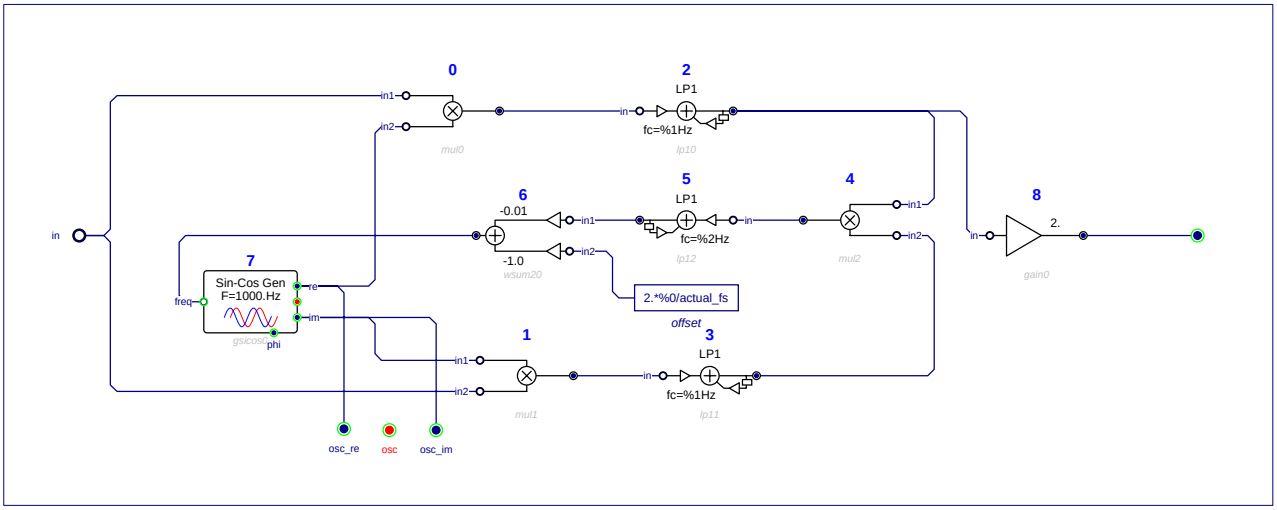
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	optional
name_osc	COMPLEX	WORD	optional
name_osc_re	FRACT	WORD	optional
name_osc_im	FRACT	WORD	optional



COSTAS test program



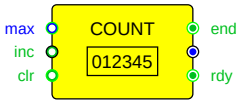


COSTAS internal schema

COUNT

Event counter

COUNT



CATEGORY: Integer

DESCRIPTION:
Event counter
Increment output on inc true, reset inc
Clear output on clr true, reset clr
Set end if output = max

PARAMETERS:

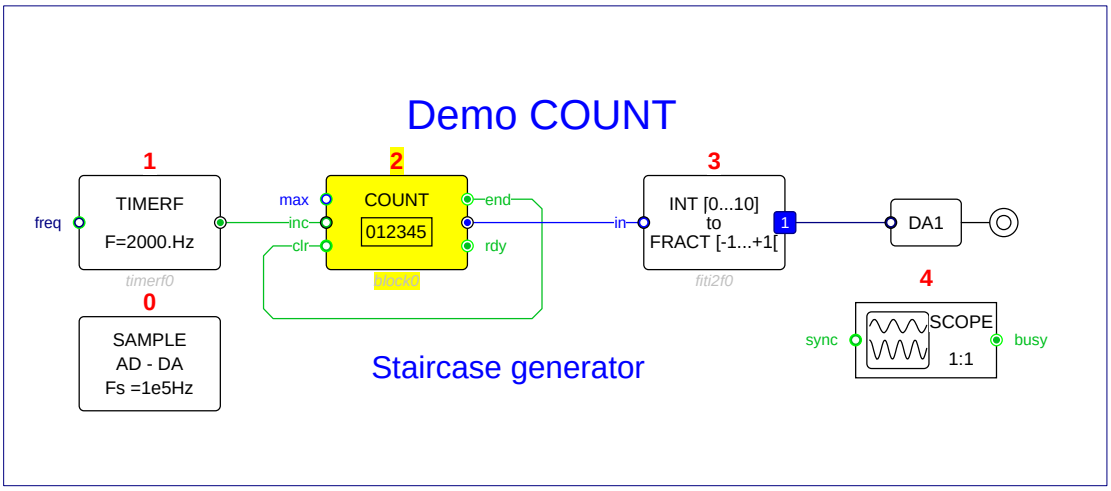
Parameter:	*Default values:*
Maximum	10

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_inc	BOOL	BIT	mandatory
name_max	INTEGER	WORD	optional
name_clr	BOOL	BIT	optional

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_end	BOOL	BIT	optional
name_rdy	BOOL	BIT	optional
name	INTEGER	WORD	normal

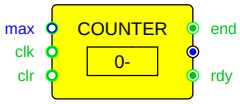


COUNT test program

COUNTER

Event counter

COUNTER



CATEGORY: Timing

DESCRIPTION:
Event counter
Increments on name_clk TRUE or on each sample if name_clk unconnected
name_end is asserted whenever count reaches maxi (or connected max value if connection exists)
Counter will not increment beyond maxi
When name_clr is true, the counter is reset to 0 on next clk
If connected, output rdy is asserted each time the counter output has been modified

PARAMETERS:

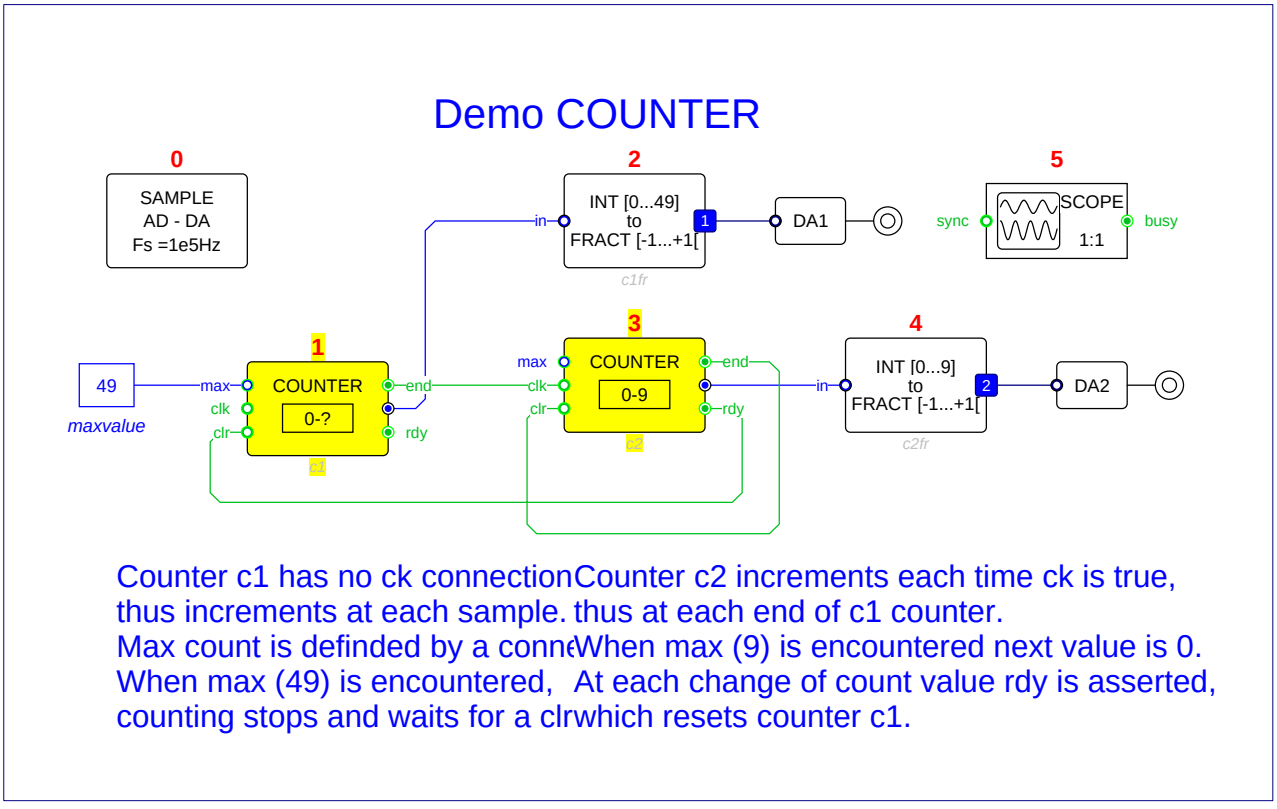
Parameter:	*Default values:*
maxi	10

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_max	INTEGER	WORD	optional
name_clk	BOOL	BIT	optional
name_clr	BOOL	BIT	optional

OUTPUTS

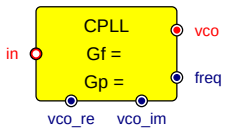
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_end	BOOL	BIT	optional
name	INTEGER	WORD	normal
name_rdy	BOOL	BIT	optional



COUNTER test program

Complex PLL

CPLL



CATEGORY: Control

DESCRIPTION:
Complex PLL

PARAMETERS:

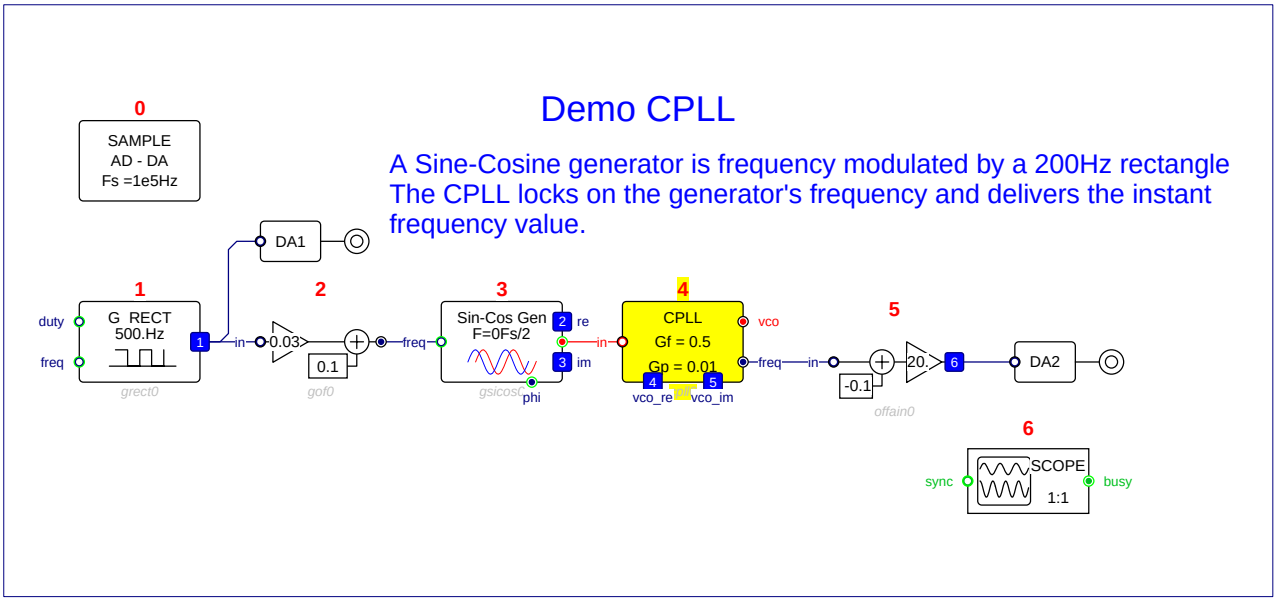
Parameter:	*Default values:*
frequency_gain	0.01
phase_gain	0.01

INPUTS

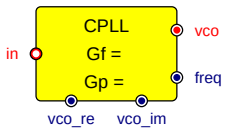
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	COMPLEX	WORD	mandatory

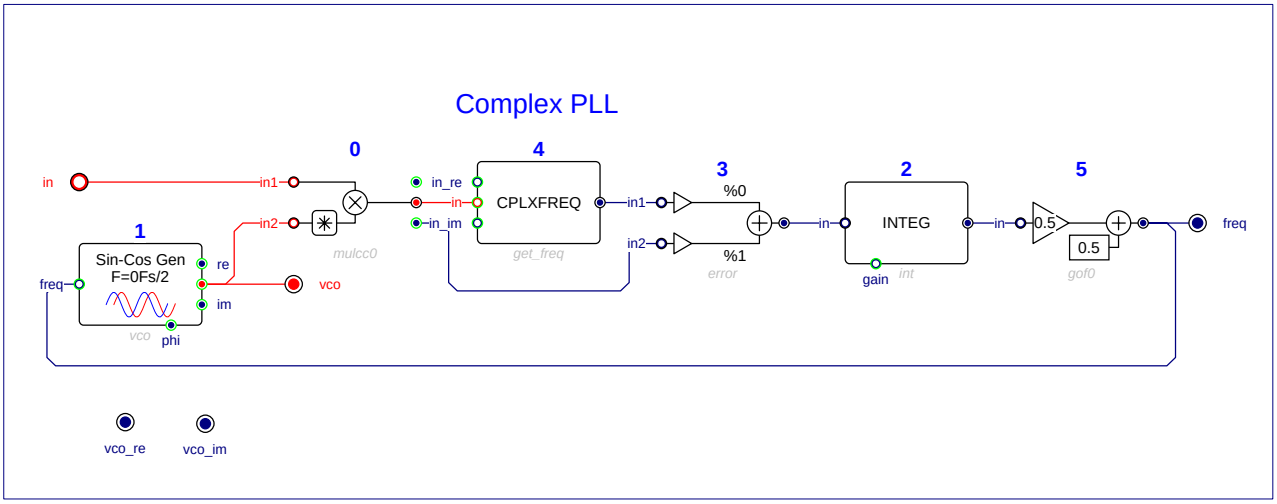
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_vco	COMPLEX	WORD	normal
name_freq	FRACT	WORD	normal
name_vco_re	FRACT	WORD	normal
name_vco_im	FRACT	WORD	normal



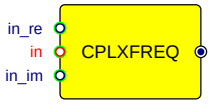
CPLL test program





CPLL internal schema

CPLXFREQ Instantaneous frequency CPLXFREQ



CATEGORY: Control

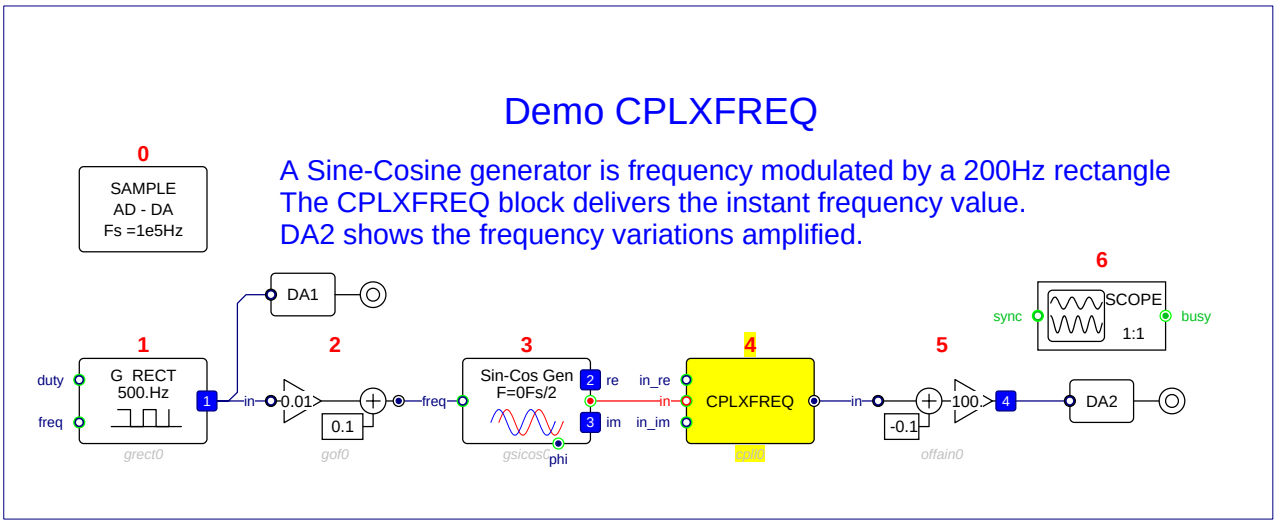
DESCRIPTION:
Instantaneous frequency
of complex signal
y = f/(Fs/2)= 1/pi * Arg(x(k).x*(k-1))

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	COMPLEX	WORD	optional
name_in_re	FRACT	WORD	optional
name_in_im	FRACT	WORD	optional

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

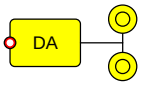


CPLXFREQ test program

DA

Digital to Analog Converters 1:2 complex input

DA



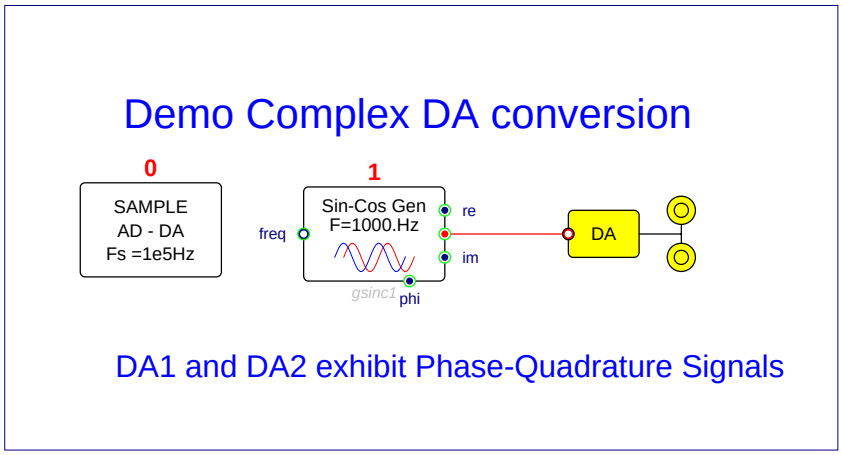
CATEGORY: Analog InOut

DESCRIPTION:
Digital to Analog Converters 1:2 complex input

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	COMPLEX	WORD	mandatory

ATTRIBUTES
Non executable, Unique,



DA test program

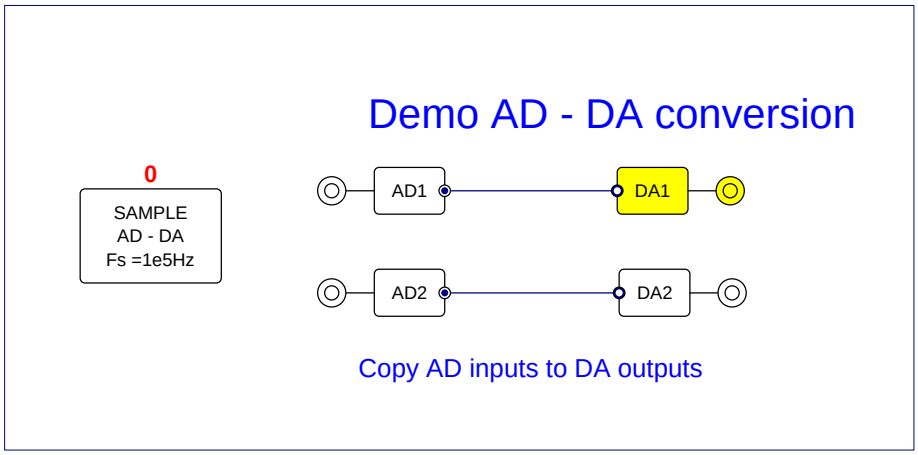
CATEGORY: Analog InOut

DESCRIPTION:
Digital to Analog Converter 1

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	mandatory

ATTRIBUTES
Non executable, Unique,

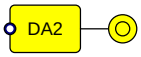


DA1 test program

DA2

Digital to Analog Converter 2 input

DA2



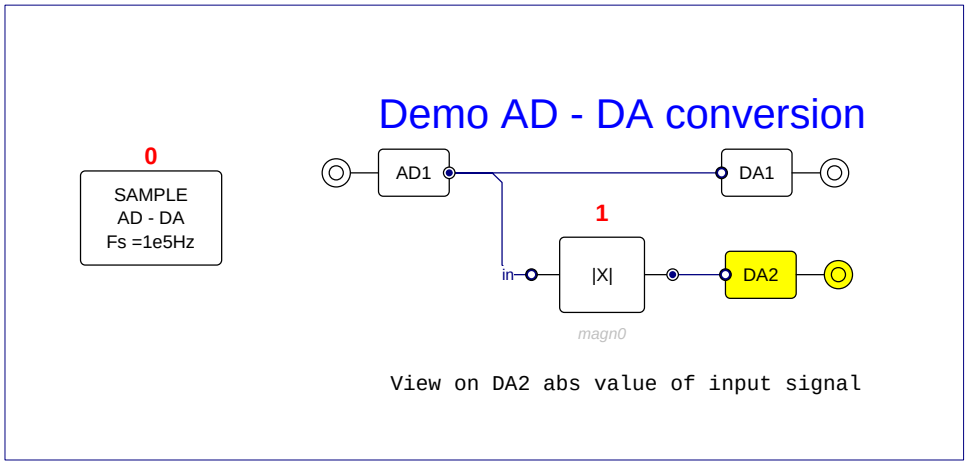
CATEGORY: Analog InOut

DESCRIPTION:
Digital to Analog Converter 2 input

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	mandatory

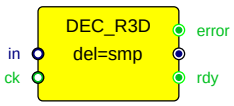
ATTRIBUTES
Non executable, Unique,



DA2 test program

3-1 Repetition decoder

DEC_R3D



CATEGORY: Telecom

DESCRIPTION:
3-1 Repetition decoder
with delay compensation

PARAMETERS:

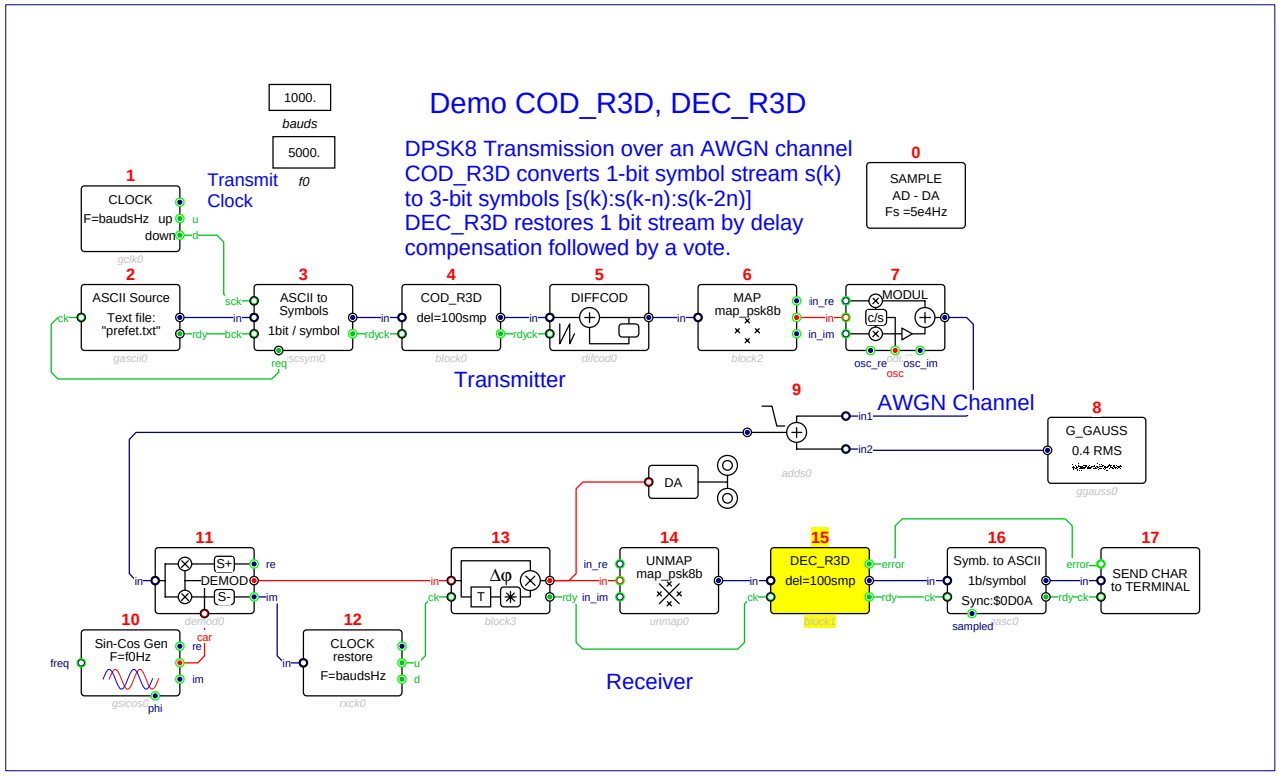
Parameter:	*Default values:*
Delay (samples)	100

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory
name_ck	BOOL	BIT	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal
name_rdy	BOOL	BIT	optional
name_error	BOOL	BIT	optional

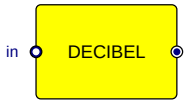


DEC_R3D test program

DECIBEL

Decibel/100 function

DECIBEL



CATEGORY: Functions

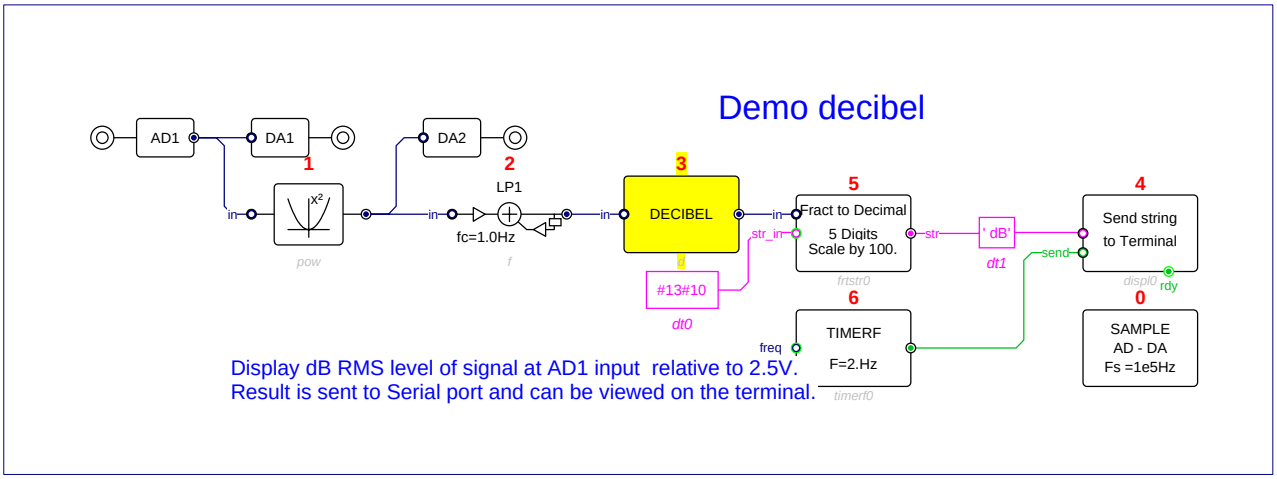
DESCRIPTION:
Decibel/100 function
y=0.1*Log10(x)

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

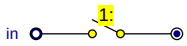
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



DECIBEL test program

Decimation

DECIM



CATEGORY: Control

DESCRIPTION:
Decimation
Every N samples copy input to output otherwise copy 0 to output

PARAMETERS:

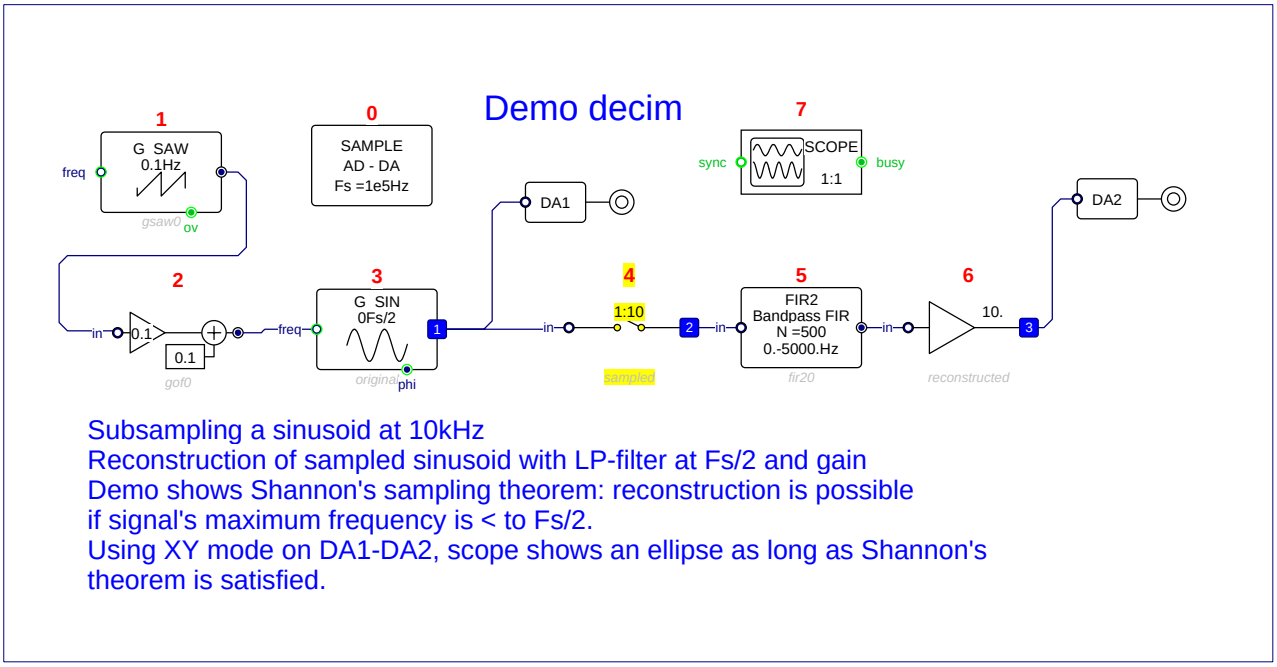
Parameter:	*Default values:*
N	10

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

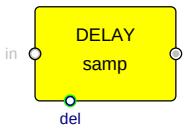


DECIM test program

DELAY

Real or complex, fixed or variable delay

DELAY



CATEGORY: Control

DESCRIPTION:
Real or complex, fixed or variable delay
abs: delay in seconds, del 0..1 = 0..delaymax
rel:delay in samples, del 0..N = 0..delaymax

PARAMETERS:

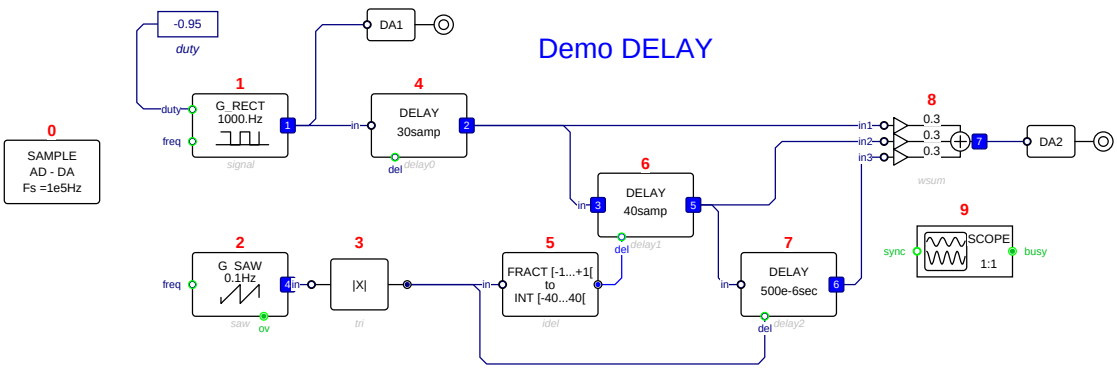
Parameter:	*Default values:*
Value	0.001
Unit	sec,samp

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	defined by cn		mandatory
name_del	FRACT	WORD	optional

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	defined by cn		normal

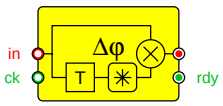


Scope animation showing moving pulses.
Uses constant, variable (samples) and variable (seconds) delay blocks
Use H/W scope to view signals, not S/W scope

DELAY test program

Argument difference

DELTAPHI



CATEGORY: Telecom

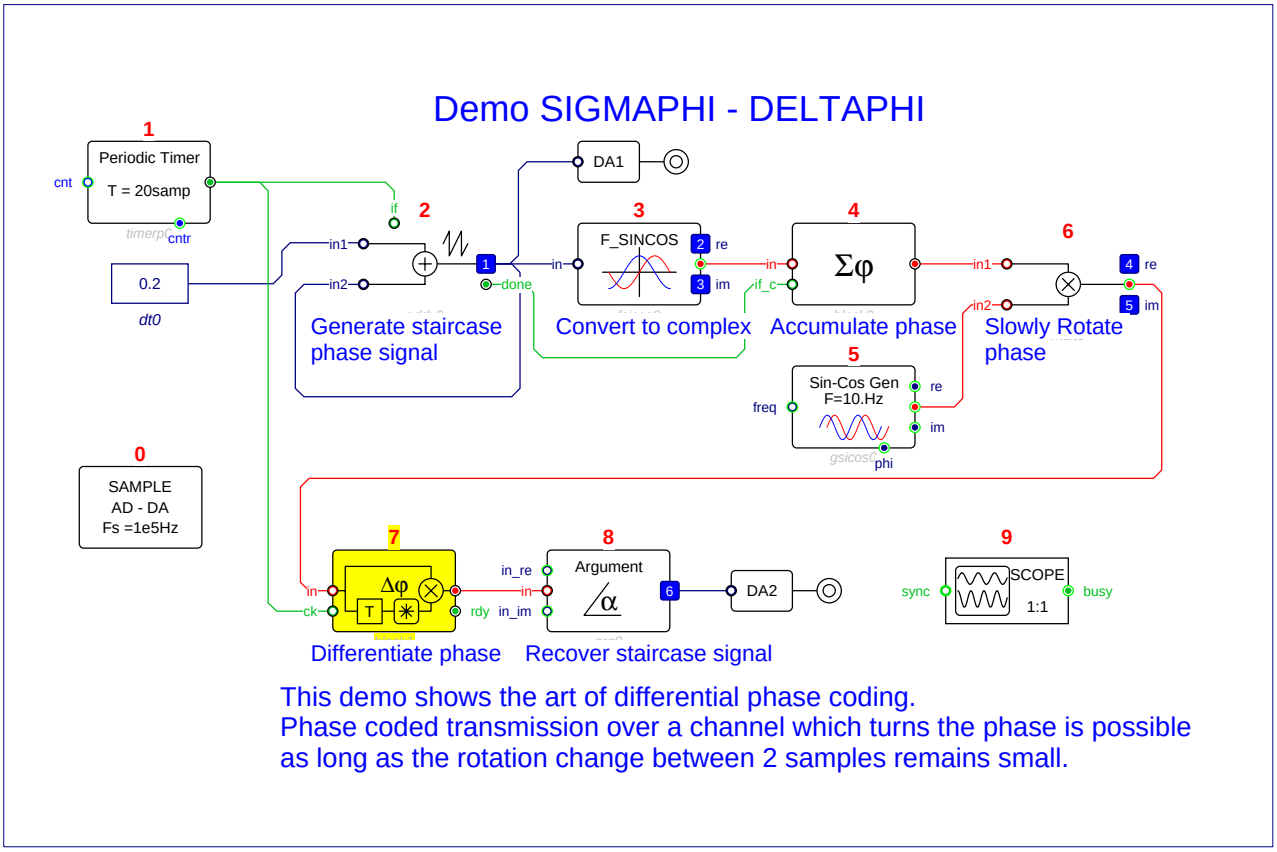
DESCRIPTION:
Argument difference
Get phase jump from previous complex sample

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	COMPLEX	WORD	mandatory
name_ck	BOOL	BIT	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	COMPLEX	WORD	normal
name_rdy	BOOL	BIT	normal

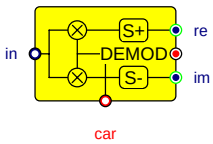


DELTAPHI test program

DEMOD

Phase-Quadrature demodulator

DEMOD



CATEGORY: Telecom

DESCRIPTION:
Phase-Quadrature demodulator

PARAMETERS:

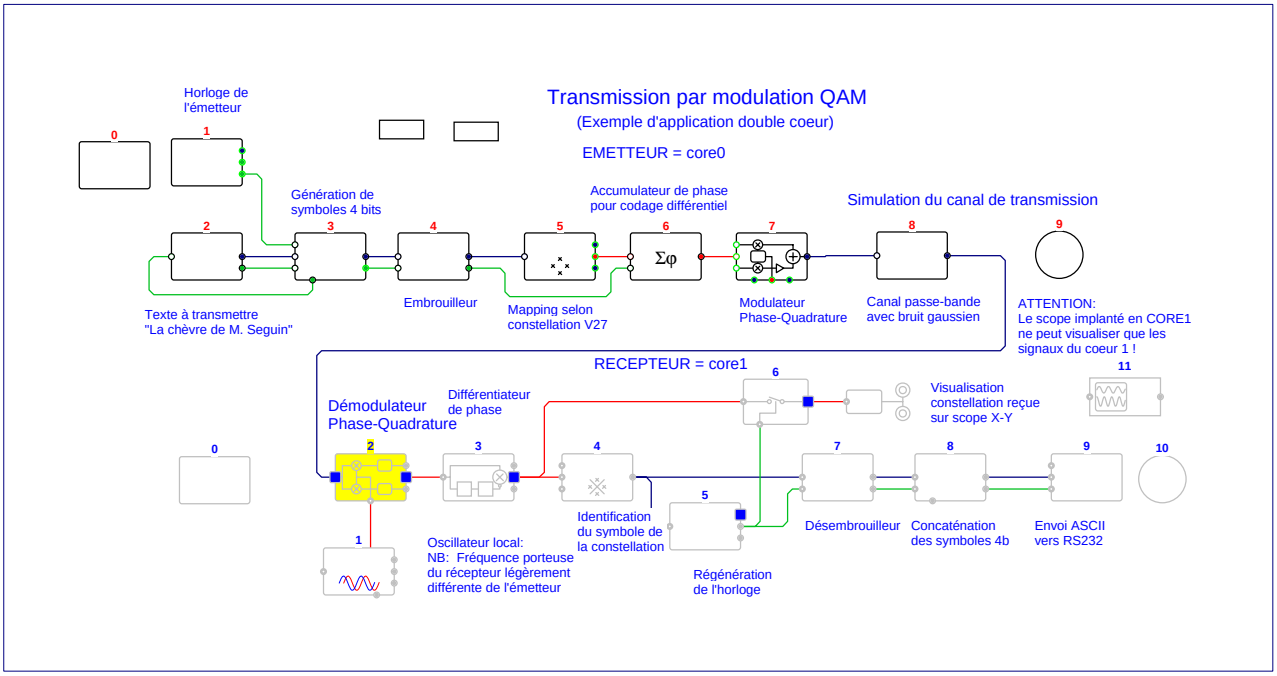
Parameter:	*Default values:*
Taccum	0.001

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory
name_car	COMPLEX	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	COMPLEX	WORD	normal
name_re	FRACT	WORD	optional
name_im	FRACT	WORD	optional

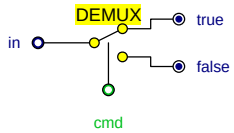


DEMOD test program

DEMUX

1 to 2 Demultiplexer

DEMUX



CATEGORY: Control

DESCRIPTION:
1 to 2 Demultiplexer

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_cmd	BOOL	BIT	mandatory
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_true	FRACT	WORD	normal
name_false	FRACT	WORD	normal

DERIV

Numerical derivator with input gain

DERIV

CATEGORY: Control

DESCRIPTION:
Numerical derivator with input gain
y(k) = g*(x(k) - x(k-1))

PARAMETERS:

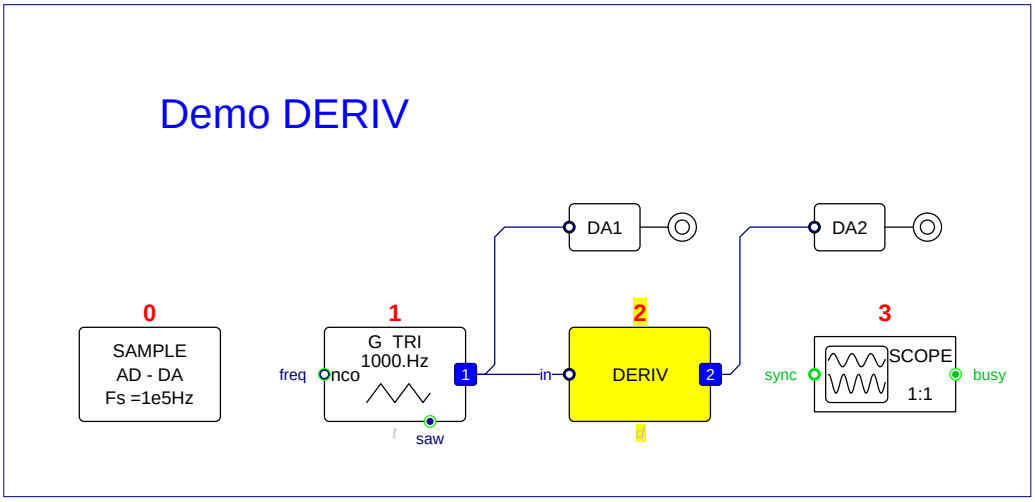
Parameter:	*Default values:*
gain	0.001

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

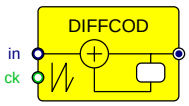


DERIV test program

DIFFCOD

Differential coder

DIFFCOD



CATEGORY: Telecom

DESCRIPTION:
Differential coder

PARAMETERS:

Parameter:	*Default values:*
Bit per symbol	1

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_ck	BOOL	BIT	mandatory
name_in	FRACT	WORD	mandatory

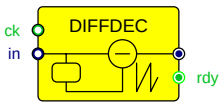
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

DIFFDEC

Differential decoder

DIFFDEC



CATEGORY: Telecom

DESCRIPTION:
Differential decoder

PARAMETERS:

Parameter:	*Default values:*
Bit per symbol	1

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_ck	BOOL	BIT	mandatory
name_in	FRACT	WORD	mandatory

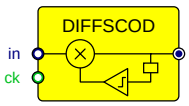
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal
name_rdy	BOOL	BIT	optional

DIFFSCOD

Differential sign coder

DIFFSCOD



CATEGORY: Telecom

DESCRIPTION:
Differential sign coder
y(k) = x(k) * sgn(y(k-1))

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_ck	BOOL	BIT	mandatory
name_in	FRACT	WORD	mandatory

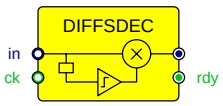
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

DIFFSDEC

Differential sign decoder

DIFFSDEC



CATEGORY: Telecom

DESCRIPTION:
Differential sign decoder
y(k) = x(k) * sgn(x(k-1))

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_ck	BOOL	BIT	mandatory
name_in	FRACT	WORD	mandatory

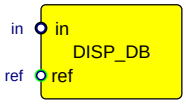
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal
name_rdy	BOOL	BIT	normal

DISP_DB

Display ratio in dB

DISP_DB



CATEGORY: String

DESCRIPTION:
Display ratio in dB
Converts ratio (in/ref) to dB, converts dB value to string,
sends string to serial port every 0.5sec .
Note: 0 < in < ref <= 1.0
if ref not connected, then ref assumed to be 1.0

PARAMETERS:

Parameter:	*Default values:*
Prologue	'Ratio = '
Epilogue	' dB'

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory
name_ref	FRACT	WORD	optional

DISP_FR

Display fract value

DISP_FR

CATEGORY: String

DESCRIPTION:
Display fract value
sends string to serial port every 0.5sec
Scaling factor: what to display if input = 1

PARAMETERS:

Parameter:	*Default values:*
Prologue	Voltage=
Digits	5
Epilogue	Volt
Scaling factor	2.5

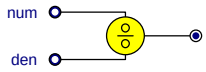
INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

DIVIDE

Fractionnal division num/den

DIVIDE



CATEGORY: Arithmetic

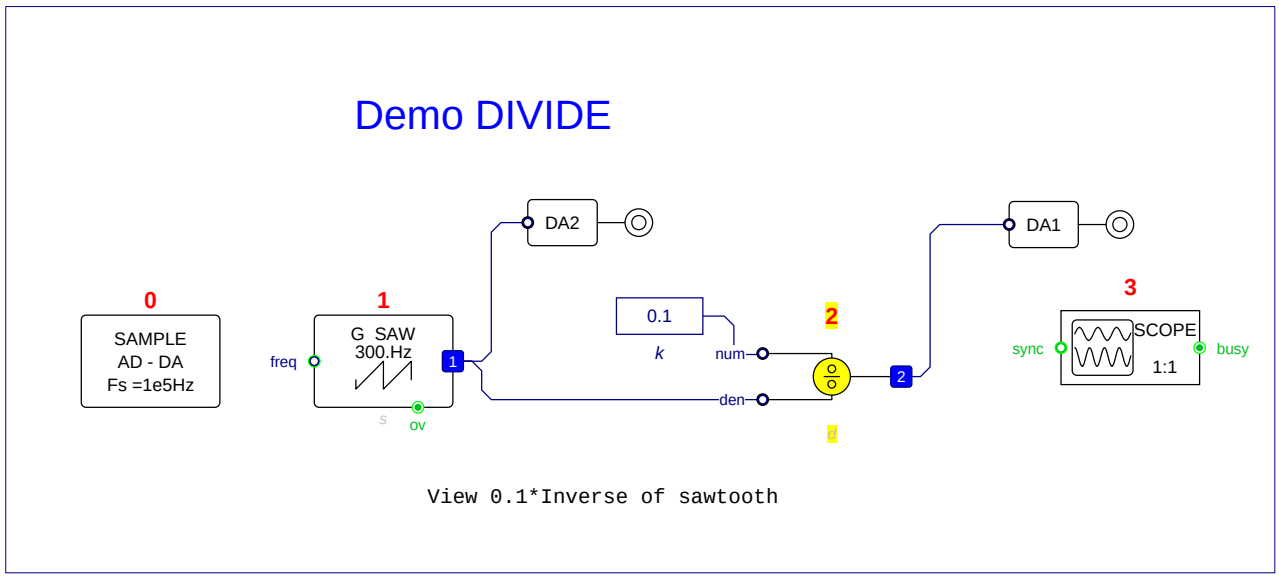
DESCRIPTION:
Fractionnal division num/den
|den| must be > to |num|

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_num	FRACT	WORD	mandatory
name_den	FRACT	WORD	mandatory

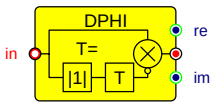
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



DIVIDE test program

Phase differentiator



CATEGORY: Telecom

DESCRIPTION:
Phase differentiator
Output argument = arg[in(t)] - arg[in(t-T)]
Output module = input module

PARAMETERS:

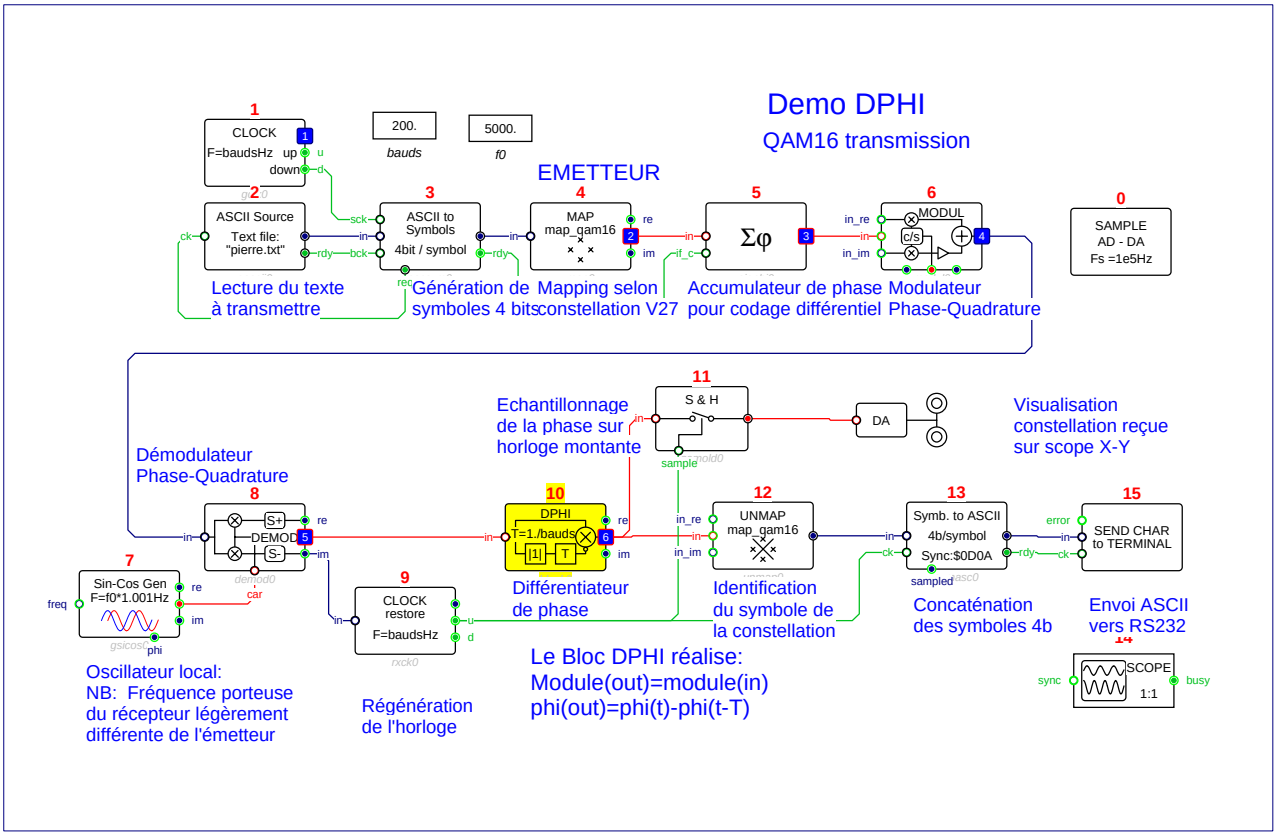
Parameter:	*Default values:*
Delay	0.001

INPUTS

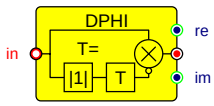
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	COMPLEX	WORD	mandatory

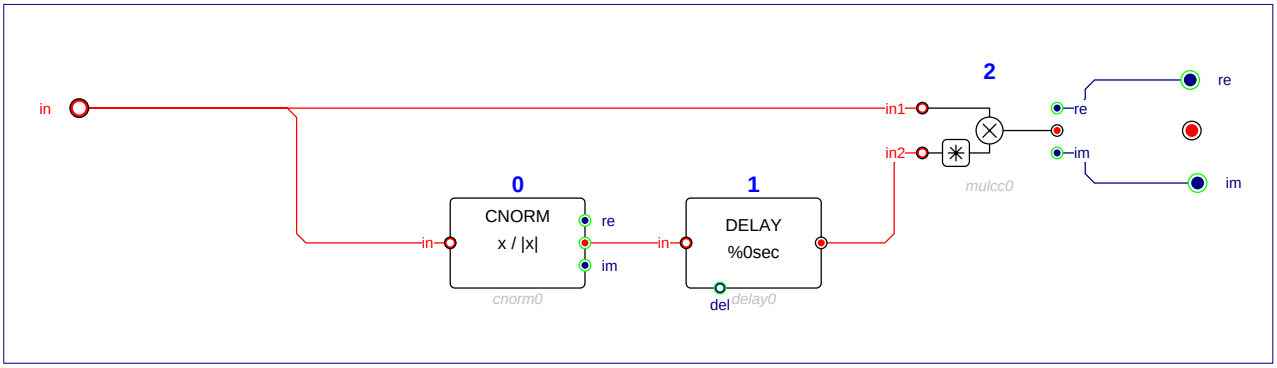
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	COMPLEX	WORD	normal
name_re	FRACT	WORD	optional
name_im	FRACT	WORD	optional



DPHI test program

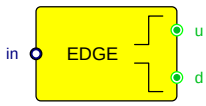




DPHI internal schema

Generate flags on zero crossing

EDGE



CATEGORY: Control

DESCRIPTION:
Generate flags on zero crossing
name_u = rising edge name_d = falling edge

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

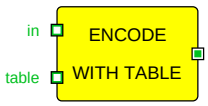
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_u	BOOL	BIT	optional
name_d	BOOL	BIT	optional

ENCODE

GF(2) encoder

ENCODE



CATEGORY: Telecom

DESCRIPTION:
GF(2) encoder
Bool line matrix encode to bool line matrix
Input= address in table (left justified)
Output= table[input]

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	BOOL	Matrix of BIT	mandatory
name_table	BOOL	Matrix of BIT	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	BOOL	Matrix of BIT	normal

F_ATAN

Arc Tangent between -1 and +1

F_ATAN

CATEGORY: Functions

DESCRIPTION:
Arc Tangent between -1 and +1
y = 1/pi * arctan(x)

PARAMETERS:

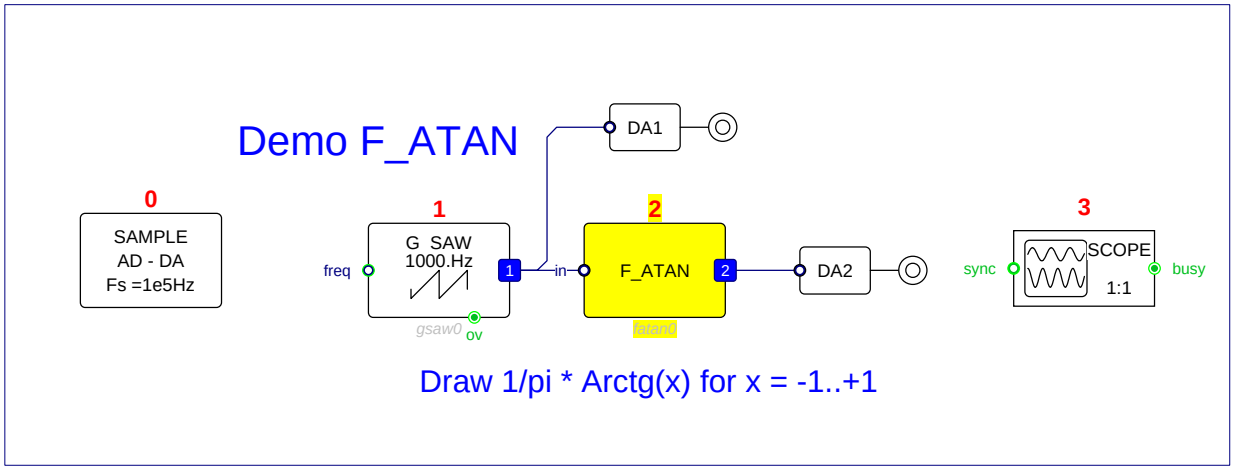
Parameter:	*Default values:*
points	15

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

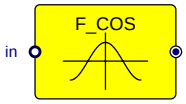


F_ATAN test program

F_COS

Cosine function y = Cos(pi*x)

F_COS



CATEGORY: Functions

DESCRIPTION:
Cosine function y = Cos(pi*x)
Input in: angle in half turns (-1..+1 -> -pi ..+pi rad)
Parameter "Points" defines the number of points of the 0..pi/2 Sine table

PARAMETERS:

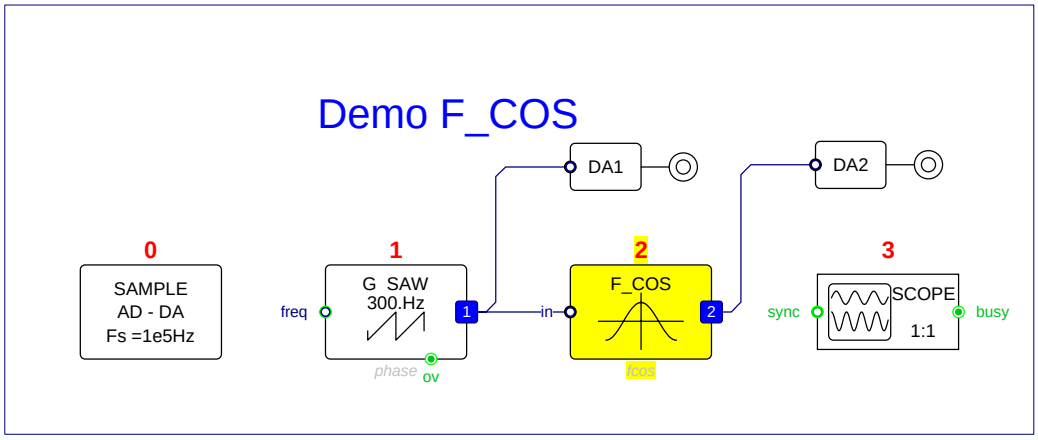
Parameter:	*Default values:*
Points	15

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

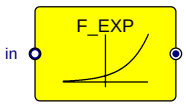


F_COS test program

F_EXP

Real exponential function y = 2^k*x / 2^k

F_EXP



CATEGORY: Functions

DESCRIPTION:
Real exponential function y = 2^k*x / 2^k

PARAMETERS:

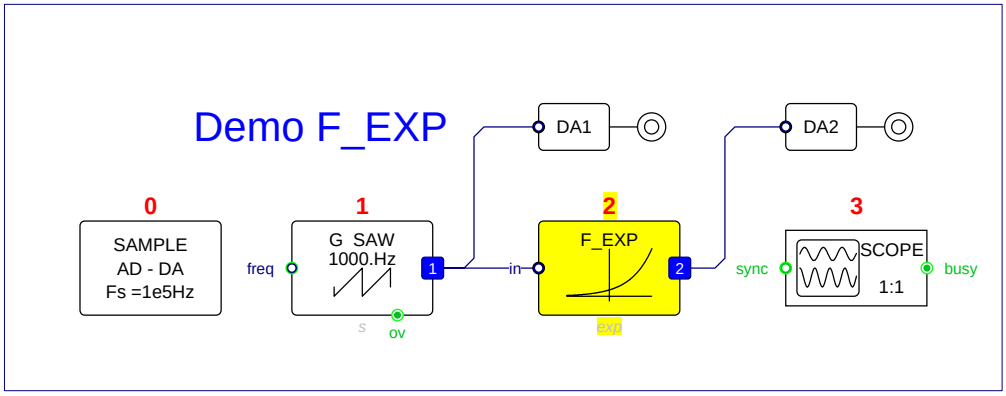
Parameter:	*Default values:*
k	3.5
Points	21

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

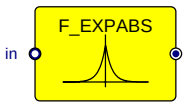


F_EXP test program

F_EXPABS

Exponential of abs

F_EXPABS



CATEGORY: Functions

DESCRIPTION:
Exponential of abs
Real exponential of -abs(input) function y = 2^-|k*x|

PARAMETERS:

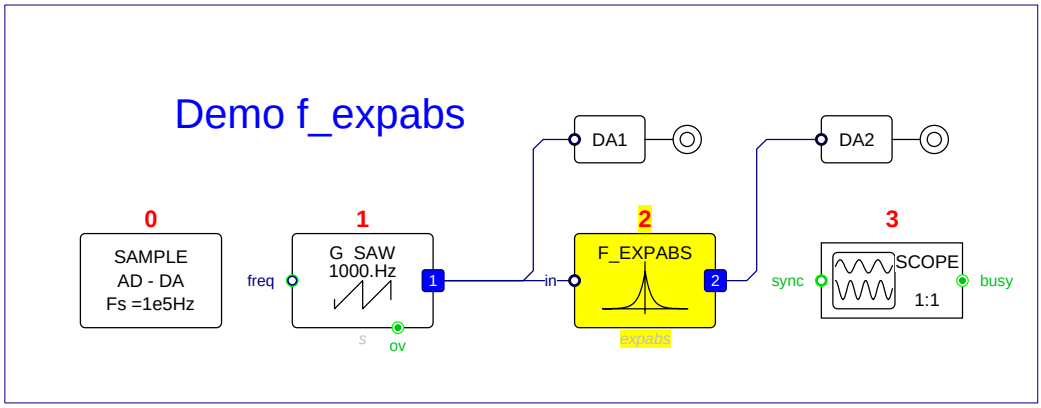
Parameter:	*Default values:*
k	3.5
Points	21

INPUTS

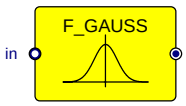
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



F_EXPABS test program



CATEGORY: Functions

DESCRIPTION:
Gaussian function
$y=2^{-kx^2}$

PARAMETERS:

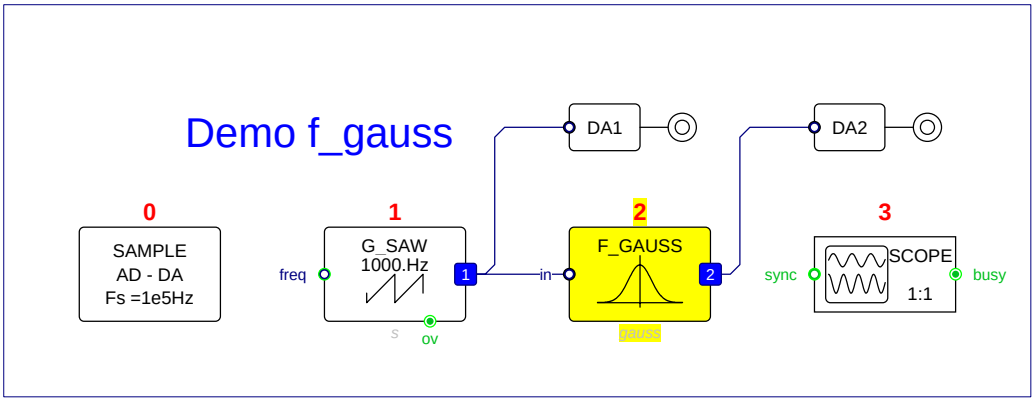
Parameter:	*Default values:*
k	15.
Points	21

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

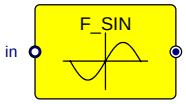


F_GAUSS test program

F_SIN

Sine function y = Sin(pi*x)

F_SIN



CATEGORY: Functions

DESCRIPTION:
Sine function y = Sin(pi*x)
Input in: angle in half turns (-1..+1 -> -pi ..+pi rad)
Parameter "Points" defines the number of points of the 0..pi/2 Sine table

PARAMETERS:

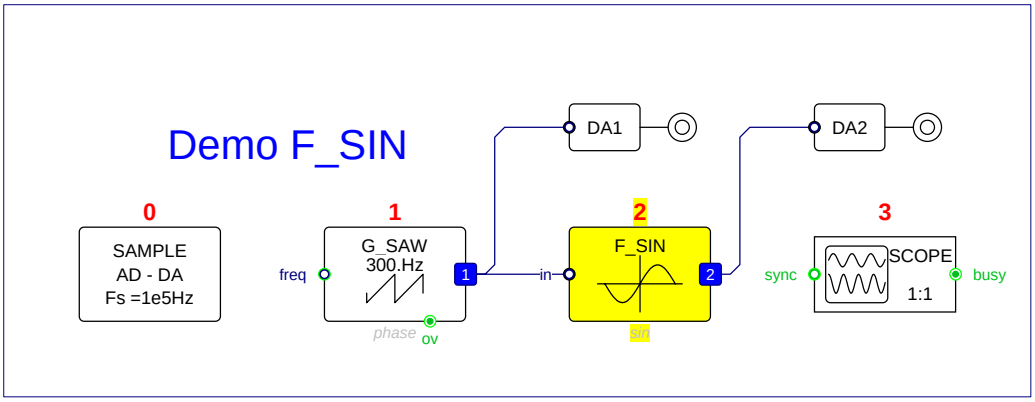
Parameter:	*Default values:*
Points	15

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

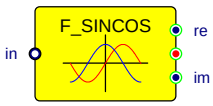


F_SIN test program

F_SINCOS

Sine-Cosine function

F_SINCOS



CATEGORY: Functions

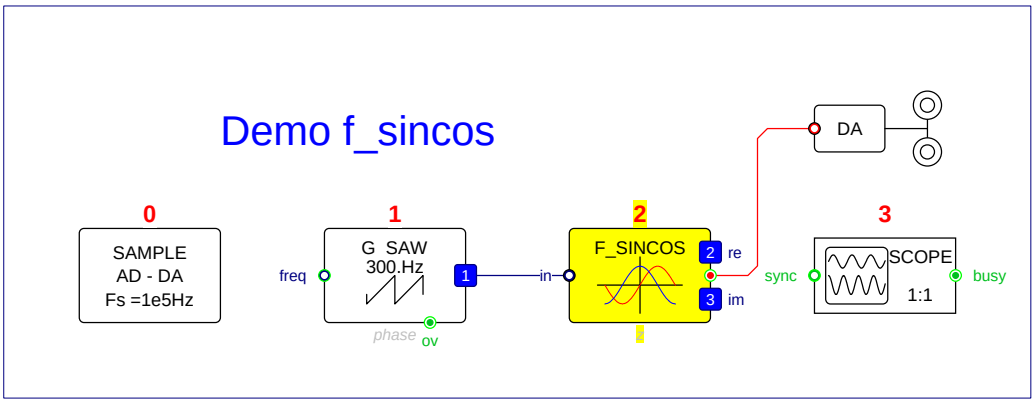
DESCRIPTION:
Sine-Cosine function
Complex output = exp(j.pi.x)
x represents the argument expressed in half turns

INPUTS

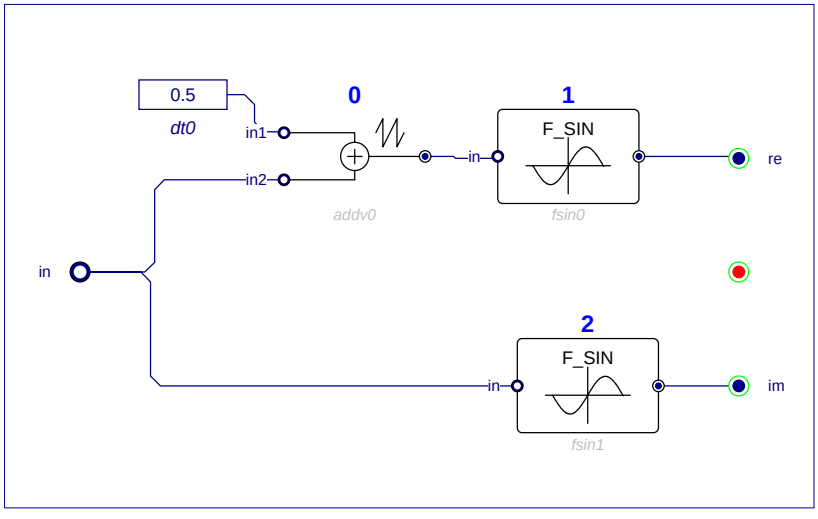
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_re	FRACT	WORD	optional
name	COMPLEX	WORD	optional
name_im	FRACT	WORD	optional



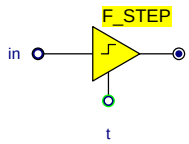
F_SINCOS test program



F_SINCOS internal schema

Step function

F_STEP



CATEGORY: Non linear

DESCRIPTION:
Step function
y= value left if x < threshold;
y= value right if x >= threshold;

PARAMETERS:

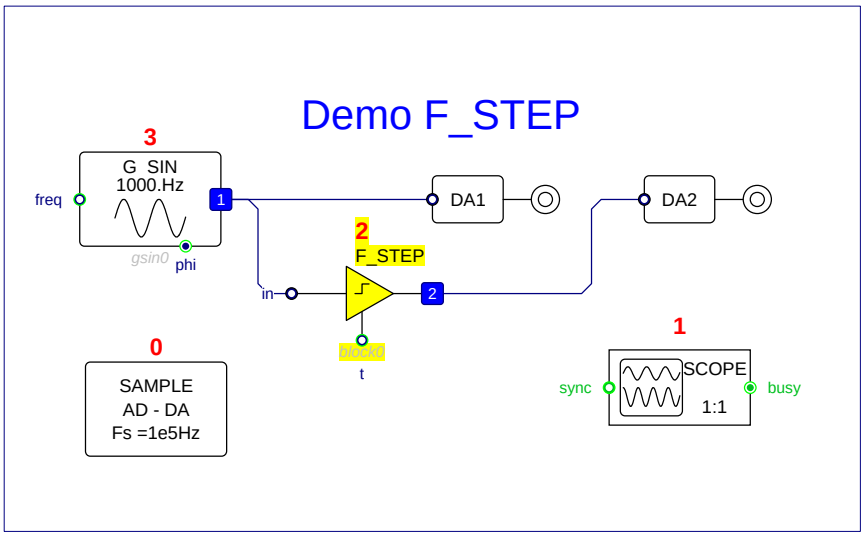
Parameter:	*Default values:*
Threshold	0
Value left	0
Value right	1.0

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory
name_t	FRACT	WORD	optional

OUTPUTS

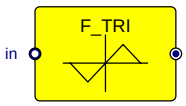
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



F_STEP test program

Triangle function

F_TRI



CATEGORY: Functions

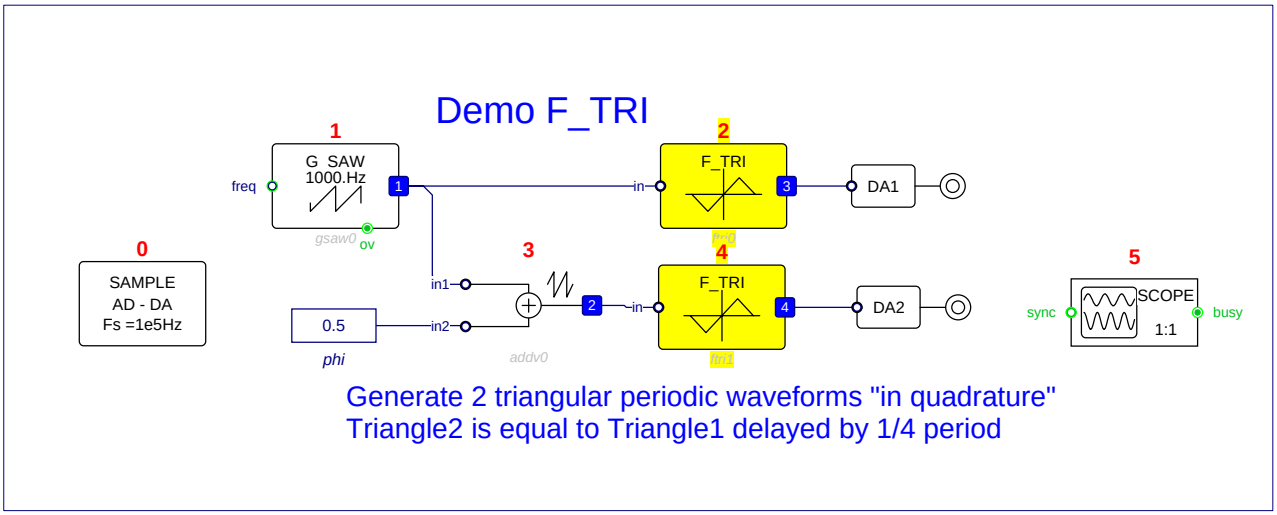
DESCRIPTION:
Triangle function
y=-2x-2, x=[-1..-0.5[y = 2x, x=[-0.5..0.5[y=-2x+2, x=[0.5..1[

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

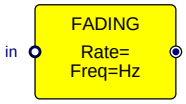


F_TRI test program

FADING

Simulate fading channel

FADING



CATEGORY: Telecom

DESCRIPTION:
Simulate fading channel

PARAMETERS:

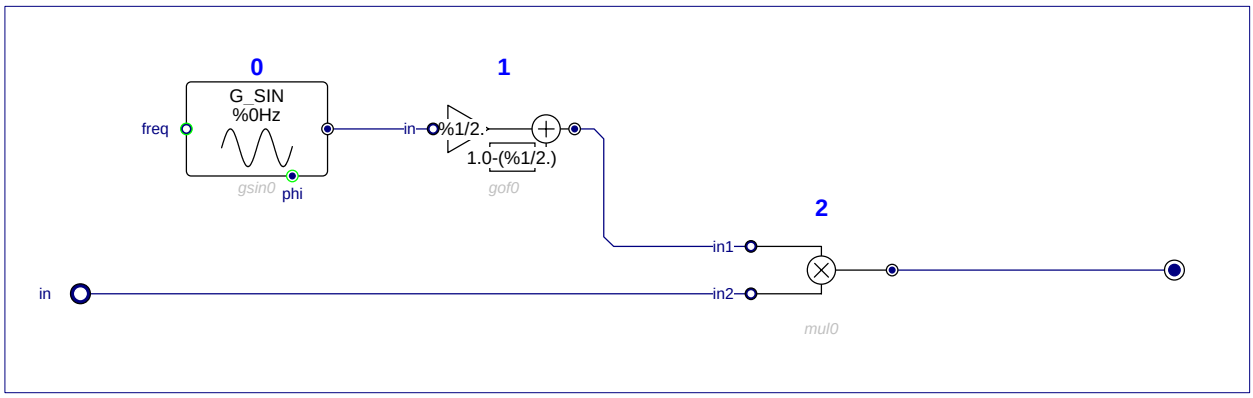
Parameter:	*Default values:*
Frequency	0.1
Proportion	0.5

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



FADING internal schema

Discrete Fast Fourier Transform

CATEGORY: Matrix

DESCRIPTION:
Discrete Fast Fourier Transform
Size of input vector must be power of 2
If input has col=1 then Real input assumed.
If input has col=2 then Complex input assumed.
Output matrix is always complex (2 columns).

PARAMETERS:

Parameter:	*Default values:*
Size	16,32,64,128,256,512,1024

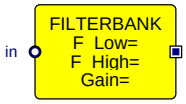
INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	COMPLEX	Matrix of WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	COMPLEX	Matrix of WORD	normal

FILTERBANK Bandpass Filter Bank FILTERBANK



CATEGORY: Filters

DESCRIPTION:
Bandpass Filter Bank
Bank of 2nd order filters with equal Q and equal gain at resonance
Resonant frequencies are regularly spaced in
geometric sequence between Freq_low and Freq_High
Number of filters is determined by output array size

PARAMETERS:

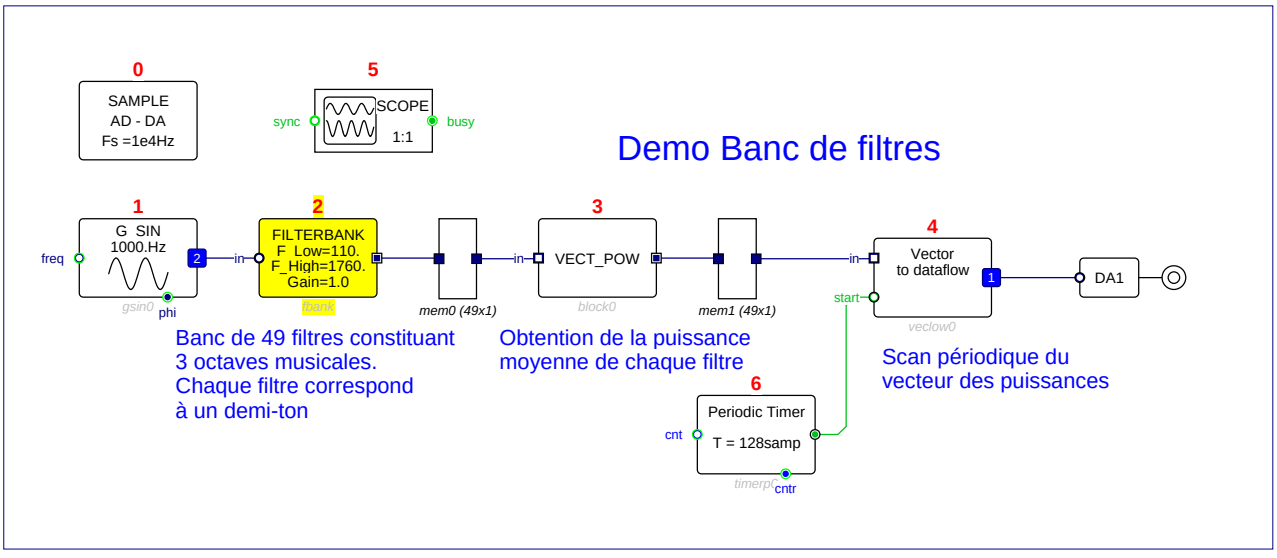
Parameter:	*Default values:*
Freq_Low	10.
Freq_High	1E4.
Gain	1.0
Unit	Hz,Fs/2

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	Matrix of WORD	normal



FILTERBANK test program

FIR

Finite Impulse Response filter

FIR

CATEGORY: Filters

DESCRIPTION:
Finite Impulse Response filter

PARAMETERS:

Parameter:	*Default values:*
impulse response	coeffs

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

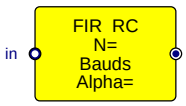
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

FIR_RC

Raised Cosine FIR

FIR_RC



CATEGORY: Telecom

DESCRIPTION:
Raised Cosine FIR
eliminates ISI in communications
Alpha = rollof factor 0: (Sinc) BW=Bauds/2 1: BW=bauds

PARAMETERS:

Parameter:	*Default values:*
Size	200
Bauds	1000.
Alpha	0.5

INPUTS

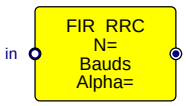
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

Root Raised Cosine

FIR_RRC



CATEGORY: Telecom

DESCRIPTION:
Root Raised Cosine
FIR filter which eliminates ISI in communications
Alpha = rollof factor 0: (Sinc) BW=Bauds/2 1: BW=bauds

PARAMETERS:

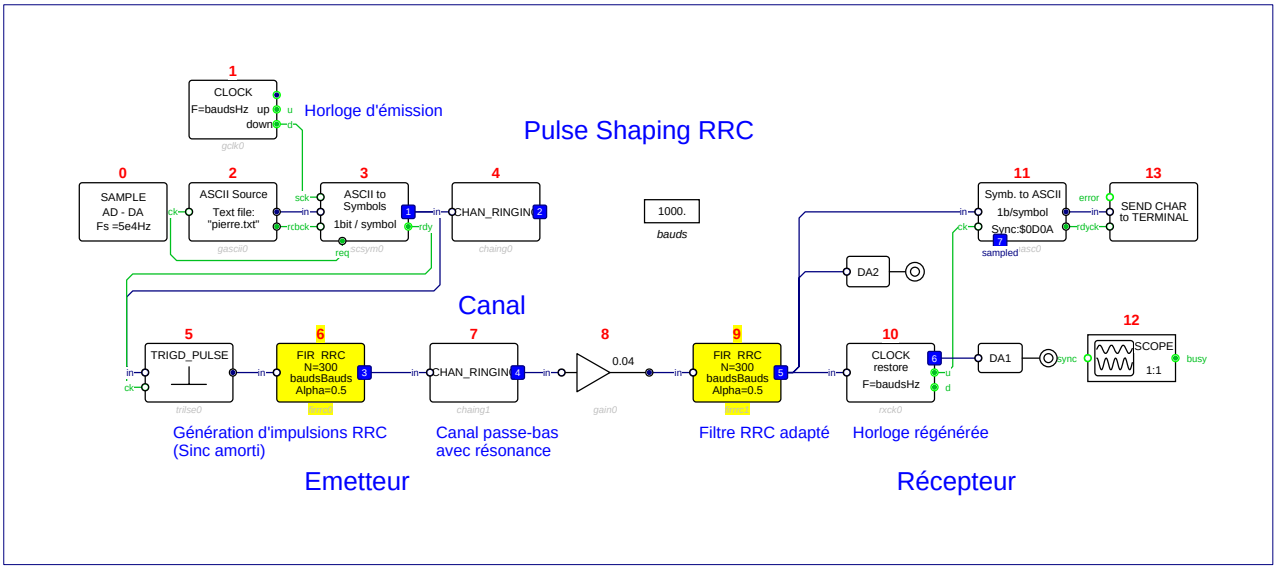
Parameter:	*Default values:*
Size	200
Bauds	1000.
Alpha	0.5

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

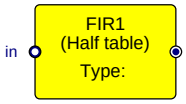
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



FIR_RRC test program

Half sized FIR



CATEGORY: Filters

DESCRIPTION:
Half sized FIR
Finite Impulse Response filter with half sized table
Impulse response should be symmetric or antisymmetric (s,a)
Full size can be odd or even (o,e)

PARAMETERS:

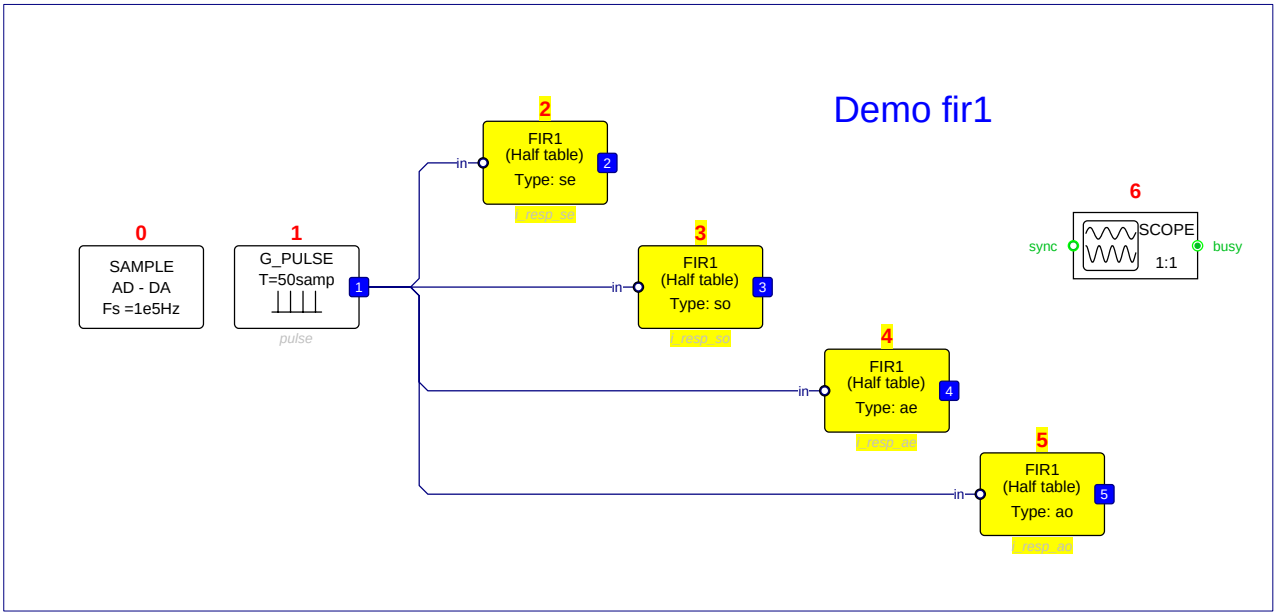
Parameter:	*Default values:*
table	coeffs
symmetry	se,so,ae,ao

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

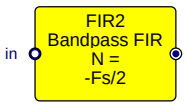
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



FIR1 test program

Bandpass Finite Impulse Response filter



CATEGORY: Filters

DESCRIPTION:
Bandpass Finite Impulse Response filter
For lowpass, do Freq low = 0
For highpass, do Freq high = Fs/2

PARAMETERS:

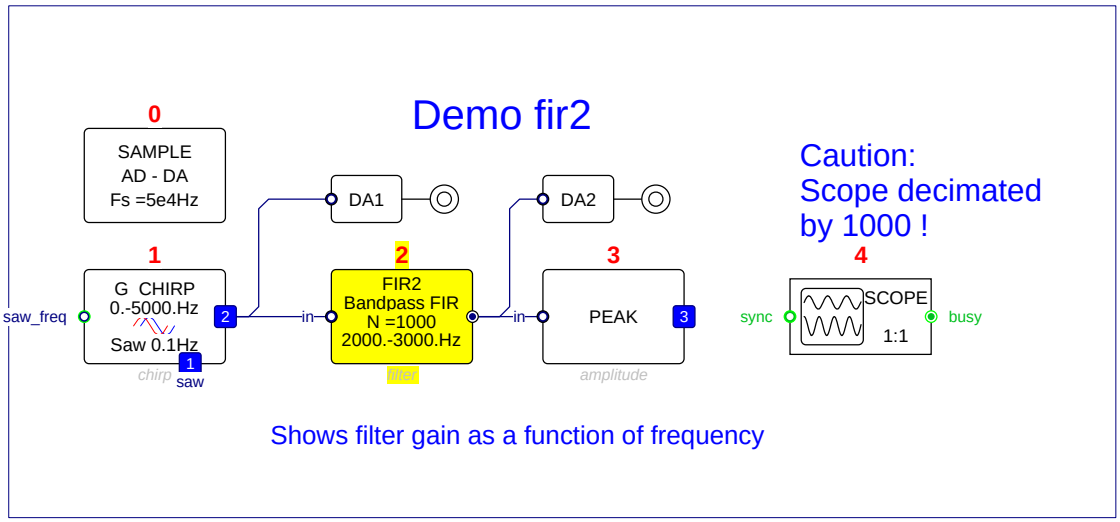
Parameter:	*Default values:*
size	500
freq low	1000.
freq high	2000.
Unit	Hz,Fs/2

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



FIR2 test program

FIRG

Gaussian FIR filter; size represents 6 sigma

FIRG

CATEGORY: Filters

DESCRIPTION:
Gaussian FIR filter; size represents 6 sigma

PARAMETERS:

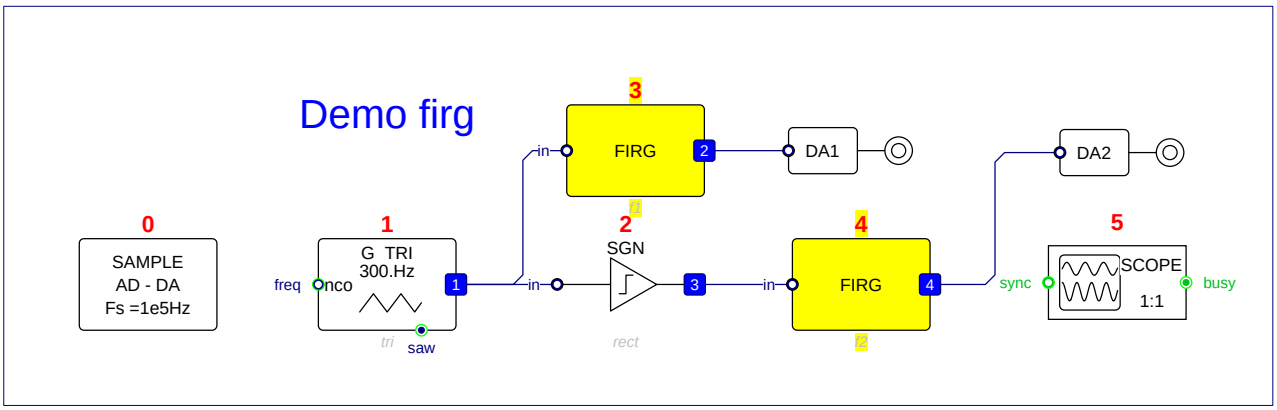
Parameter:	*Default values:*
Size	50

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

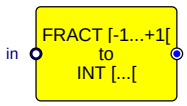
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



FIRG test program

Fract to Integer

FITF2I



CATEGORY: Integer

DESCRIPTION:
Fract to Integer
Converts Fract [-1.0..+1.0[--> Integer [min ...max[

PARAMETERS:

Parameter:	*Default values:*
min	0
max	100

INPUTS

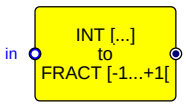
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	INTEGER	WORD	normal

Integer to Fract

FITI2F



CATEGORY: Integer

DESCRIPTION:
Integer to Fract
Converts Integer[min ...max] --> Fract[-1.0..+1.0[

PARAMETERS:

Parameter:	*Default values:*
min	0
max	100

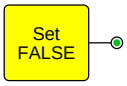
INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	INTEGER	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

FLAGCLR Set boolean variable to FALSE FLAGCLR



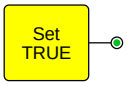
CATEGORY: Logic

DESCRIPTION:
Set boolean variable to FALSE

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	BOOL	BIT	normal

FLAGSET Set boolean variable to TRUE FLAGSET



CATEGORY: Logic

DESCRIPTION:
Set boolean variable to TRUE

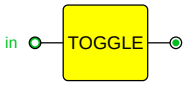
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	BOOL	BIT	normal

FLAGTOG

Toggle boolean variable

FLAGTOG



CATEGORY: Logic

DESCRIPTION:
Toggle boolean variable
On input TRUE, toggle output, then reset input to false.

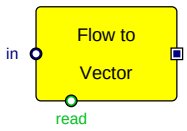
INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	BOOL	BIT	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	BOOL	BIT	normal

FLOWTOVECT Data flow to vector. FLOWTOVECT



CATEGORY: Matrix

DESCRIPTION:
Data flow to vector.
Cyclic buffer data shift register
On boolean read command, buffer data are copied to output vector
without deleting input buffer. This allows overlapping FFT.

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory
name_read	BOOL	BIT	mandatory

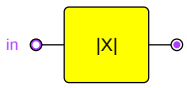
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	Matrix of WORD	normal

FP_ABS

Floating Point absolute value

FP_ABS



CATEGORY: Floating_Point

DESCRIPTION:
Floating Point absolute value

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FLOAT	DWORD	mandatory

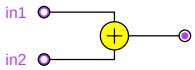
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FLOAT	DWORD	normal

FP_ADD

Floating Point Addition

FP_ADD



CATEGORY: Floating_Point

DESCRIPTION:
Floating Point Addition

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in1	FLOAT	DWORD	mandatory
name_in2	FLOAT	DWORD	mandatory

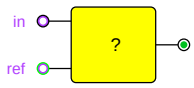
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FLOAT	DWORD	normal

FP_CMP

Comparator with boolean output

FP_CMP



CATEGORY: Floating_Point

DESCRIPTION:
Comparator with boolean output
Result is True if condition met
Reference level is ref input if connected, parameter otherwise

PARAMETERS:

Parameter:	*Default values:*
Ref level	0
Condition	in>ref,in>=ref,in=ref,in<>ref,in<=ref,in<ref

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FLOAT	DWORD	mandatory
name_ref	FLOAT	DWORD	optional

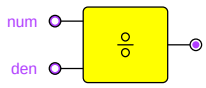
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	BOOL	BIT	normal

FP_DIV

Floating Point division num/den

FP_DIV



CATEGORY: Floating_Point

DESCRIPTION:
Floating Point division num/den

INPUTS

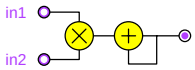
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_num	FLOAT	DWORD	mandatory
name_den	FLOAT	DWORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FLOAT	DWORD	normal

Floating Point multiply-accumulate

FP_MAC



CATEGORY: Floating_Point

DESCRIPTION:
Floating Point multiply-accumulate
y(k)=y(k-1) +/- x1(k)*x2(k)

PARAMETERS:

Parameter:	*Default values:*
Sign of accumulation	pos,neg

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in1	FLOAT	DWORD	mandatory
name_in2	FLOAT	DWORD	mandatory

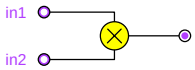
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FLOAT	DWORD	normal

FP_MPY

Floating Point multiply

FP_MPY



CATEGORY: Floating_Point

DESCRIPTION:
Floating Point multiply

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in1	FLOAT	DWORD	mandatory
name_in2	FLOAT	DWORD	mandatory

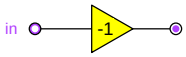
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FLOAT	DWORD	normal

FP_NEG

Floating Point Sign inversion y = -x

FP_NEG



CATEGORY: Floating_Point

DESCRIPTION:
Floating Point Sign inversion y = -x

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FLOAT	DWORD	mandatory

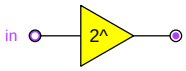
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FLOAT	DWORD	normal

FP_SCALE

Floating Point scaling

FP_SCALE



CATEGORY: Floating_Point

DESCRIPTION:
Floating Point scaling
multiply by 2^N

PARAMETERS:

Parameter:	*Default values:*
N	1

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FLOAT	DWORD	mandatory

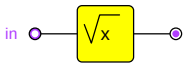
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FLOAT	DWORD	normal

FP_SQRT

Square root of input

FP_SQRT



CATEGORY: Floating_Point

DESCRIPTION:
Square root of input

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FLOAT	DWORD	mandatory

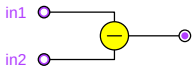
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FLOAT	DWORD	normal

FP_SUB

Floating Point subtraction

FP_SUB



CATEGORY: Floating_Point

DESCRIPTION:
Floating Point subtraction

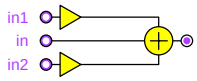
INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in1	FLOAT	DWORD	mandatory
name_in2	FLOAT	DWORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FLOAT	DWORD	normal

FP_WMAC2 Float y= x0+g1*x1+g2*x2 FP_WMAC2



CATEGORY: Floating_Point

DESCRIPTION:
Float y= x0+g1*x1+g2*x2
Floating point sum of one input and 2 weighted inputs
y = in + g1*in1 + g2*in2

PARAMETERS:

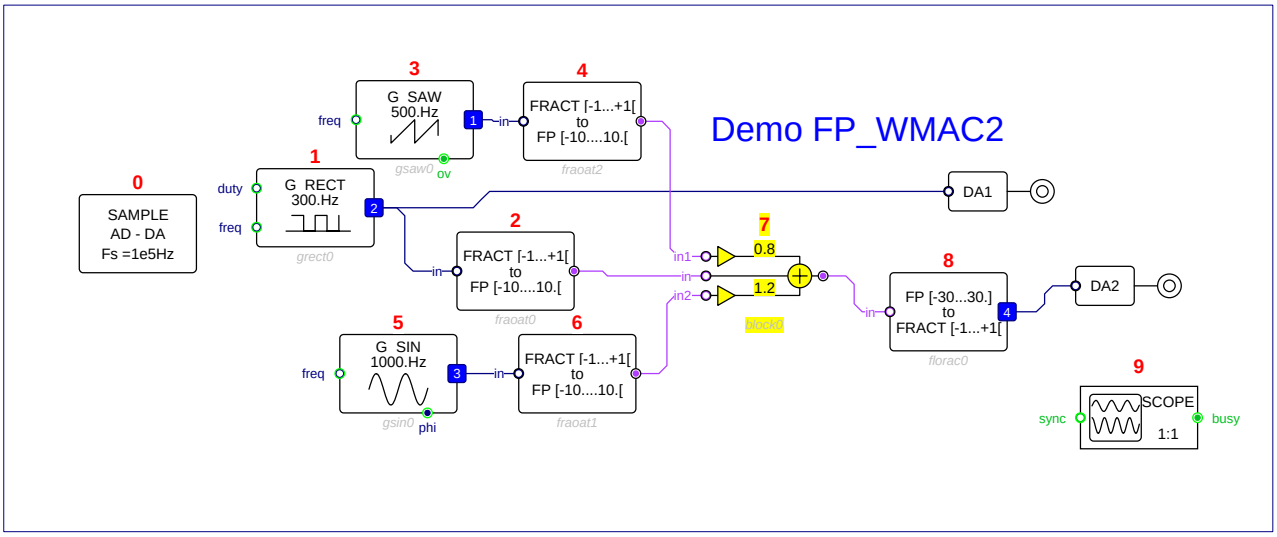
Parameter:	*Default values:*
Gain1	1.0
Gain2	1.0

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in1	FLOAT	DWORD	mandatory
name_in2	FLOAT	DWORD	mandatory
name_in	FLOAT	DWORD	mandatory

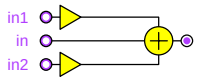
OUTPUTS

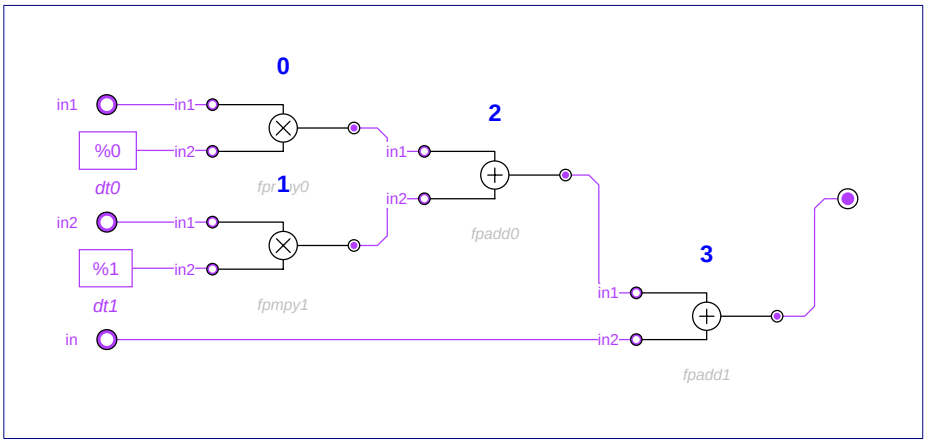
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FLOAT	DWORD	normal



FP_WMAC2 test program

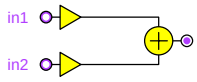
FP_WMAC2 Float y= x0+g1*x1+g2*x2 FP_WMAC2





FP_WMAC2 internal schema

FP_WSUM2 Float weighted sum FP_WSUM2



CATEGORY: Floating_Point

DESCRIPTION:
Float weighted sum
y = g1*in1 + g2*in2

PARAMETERS:

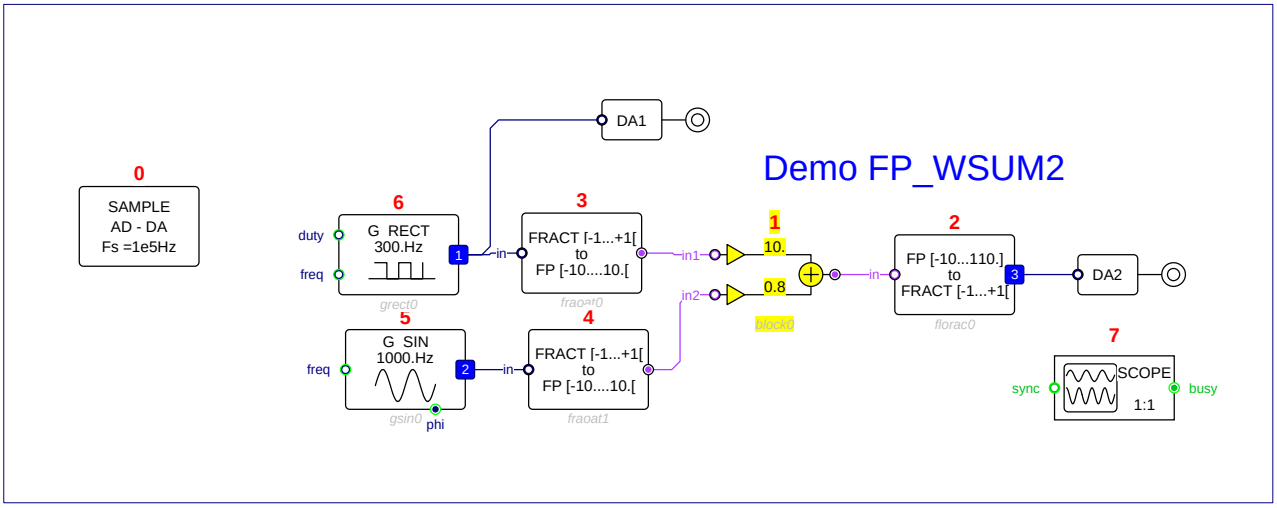
Parameter:	*Default values:*
Gain1	1.0
Gain2	1.0

INPUTS

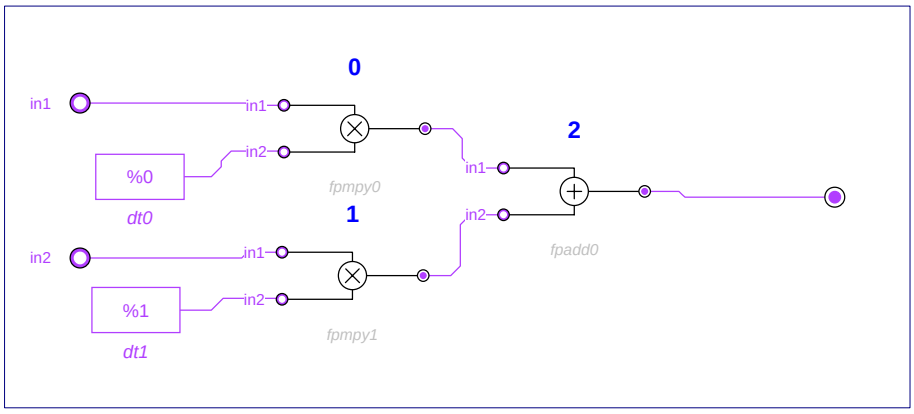
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in1	FLOAT	DWORD	mandatory
name_in2	FLOAT	DWORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FLOAT	DWORD	normal



FP_WSUM2 test program

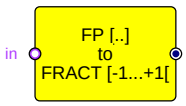


FP_WSUM2 internal schema

FPTOFR

Float to Fract

FPTOFR



CATEGORY: Floating_Point

DESCRIPTION:
Float to Fract
Converts Float [min ...max] --> Fract [-1.0..+1.0[

PARAMETERS:

Parameter:	*Default values:*
min	-10.
max	10.

INPUTS

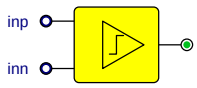
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FLOAT	DWORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

Comparator

FRCOMP



CATEGORY: Logic

DESCRIPTION:
Comparator
Result is TRUE if in is > ref
ref is either defined by connection or by parameter

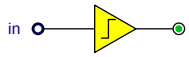
INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_inp	FRACT	WORD	mandatory
name_inn	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	BOOL	BIT	normal

FRTOBOOL Comparator FRTOBOOL



CATEGORY: Logic

DESCRIPTION:
Comparator
Result is TRUE if in is > ref
ref is either defined by connection or by parameter

PARAMETERS:

Parameter:	*Default values:*
Ref level	0

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

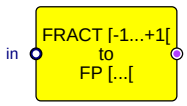
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	BOOL	BIT	normal

FRTOFP

Fract to Float

FRTOFP



CATEGORY: Floating_Point

DESCRIPTION:
Fract to Float
Converts Fract[-1.0..+1.0[--> Float [min ...max[

PARAMETERS:

Parameter:	*Default values:*
min	-10.
max	10.

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

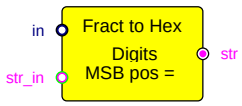
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FLOAT	DWORD	normal

FRTOHEX

Fract to Hex-String

FRTOHEX



CATEGORY: String

DESCRIPTION:
Fract to Hex-String
Convert Fractional input to Hexadecimal String

PARAMETERS:

Parameter:	*Default values:*
Nb digits	6
MSB position	23

INPUTS

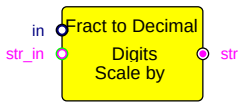
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory
name_str_in	STRING	WORD	optional

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_str	STRING	WORD	normal

Fract to String

FRTOSTR



CATEGORY: String

DESCRIPTION:
Fract to String
Convert Fractional input to Decimal String
Scaling factor gives displayed value for input 1.0

PARAMETERS:

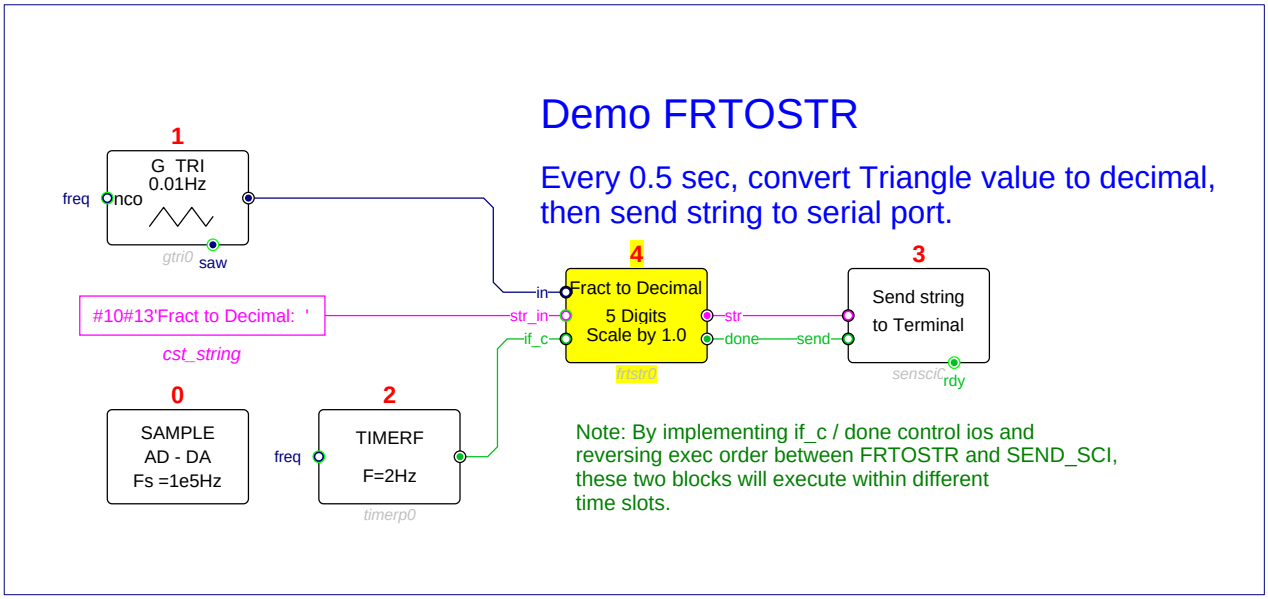
Parameter:	*Default values:*
Nb digits	5
Scaling factor	1.0

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory
name_str_in	STRING	WORD	optional

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_str	STRING	WORD	normal

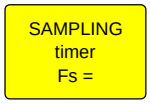


FRTOSTR test program

FS_TIMER

Waits for sample time

FS_TIMER



CATEGORY: Timing

DESCRIPTION:
Waits for sample time
Defines actual_fs

PARAMETERS:

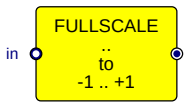
Parameter:	*Default values:*
Fs	1E5

ATTRIBUTES
Unique, Execute First, Defines: actual_fs

FULLSCALE

Stretch to [-1..+1[

FULLSCALE



CATEGORY: Arithmetic

DESCRIPTION:
Stretch to [-1..+1[
Extend signal range from [min .. max[to [-1 .. +1[
y = (2x-max-min)/(max-min)

PARAMETERS:

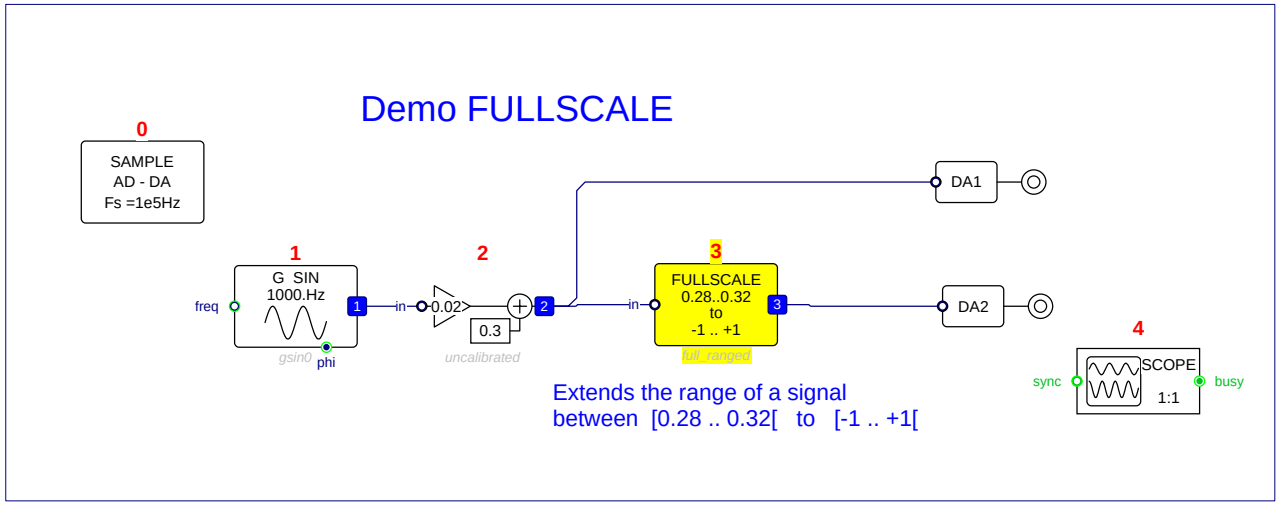
Parameter:	*Default values:*
Min input	0.1
Max input	0.2

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

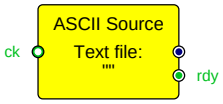
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



FULLSCALE test program

Triggered ASCII source

G_ASCII



CATEGORY: Telecom

DESCRIPTION:
Triggered ASCII source

PARAMETERS:

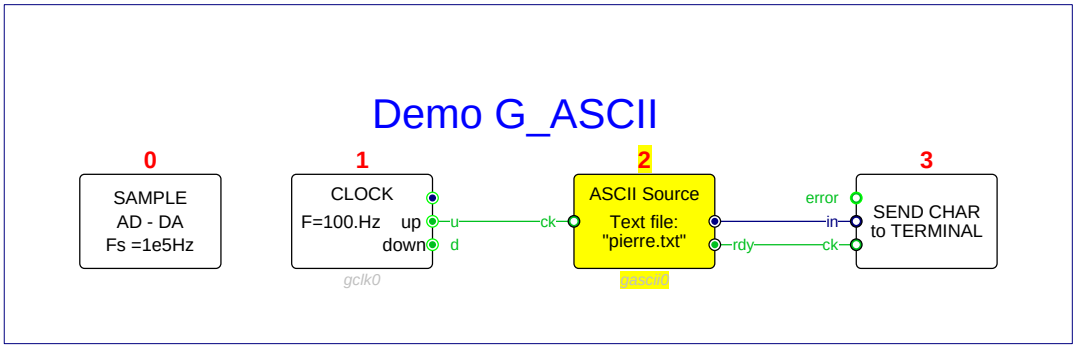
Parameter:	*Default values:*
Text file	pierre.txt,prefet.txt,chevre.txt

INPUTS

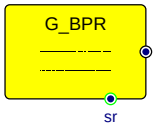
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_ck	BOOL	BIT	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal
name_rdy	BOOL	BIT	normal



G_ASCII test program



CATEGORY: Generators

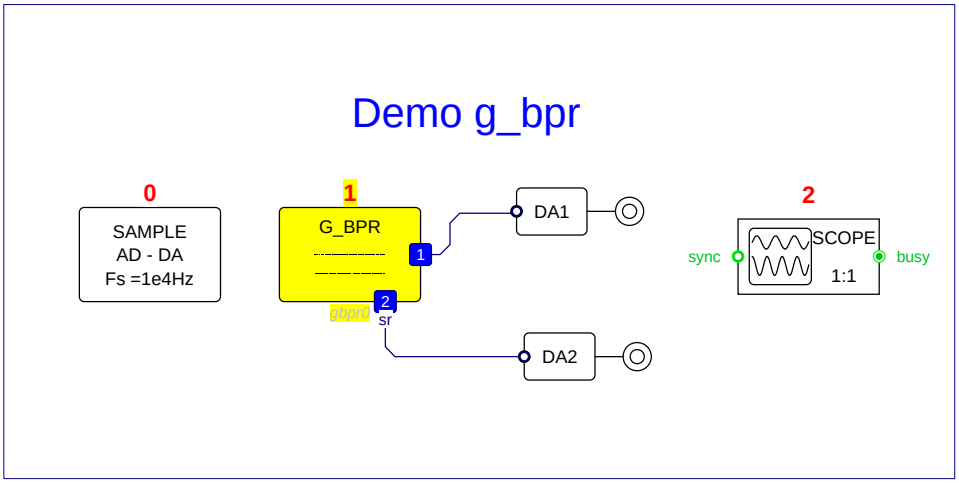
DESCRIPTION:
Binary Random Generator
Pseudo Random Sequence length = 2^bits - 1

PARAMETERS:

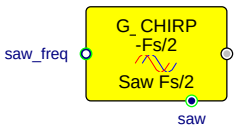
Parameter:	*Default values:*
Bits	10

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal
name_sr	FRACT	WORD	optional



G_BPR test program



CATEGORY: Generators

DESCRIPTION:
Chirp Generator
Real or Complex (linearly frequency modulated Sine or Sine-Cosine)

PARAMETERS:

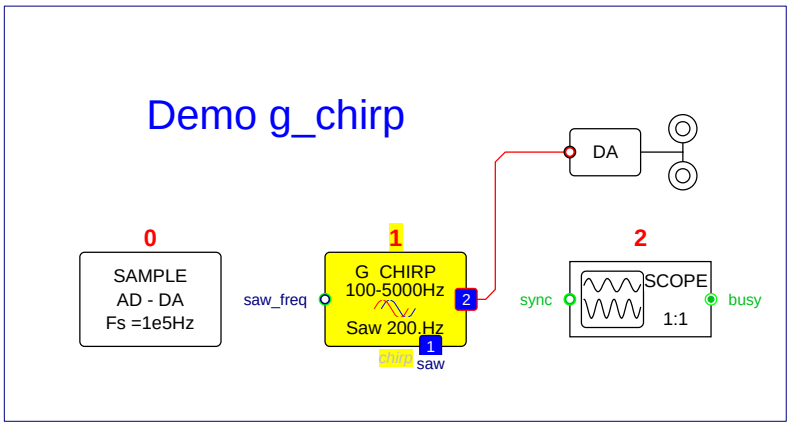
Parameter:	*Default values:*
Fmin	0
Fmax	1000.
Fsaw	10.
Unit	Hz,Fs/2

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_saw_freq	FRACT	WORD	optional

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	defined by cn		normal
name_saw	FRACT	WORD	optional

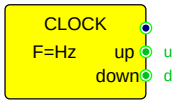


G_CHIRP test program

G_CLK

Clock Generator

G_CLK



CATEGORY: Telecom

DESCRIPTION:
Clock Generator
Boolean Clock generator for transmission Baud Rate
u = rising edge; d = falling edge

PARAMETERS:

Parameter:	*Default values:*
Frequency (Hz)	1000.

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_d	BOOL	BIT	optional
name_u	BOOL	BIT	optional
name	FRACT	WORD	optional

CATEGORY: Generators

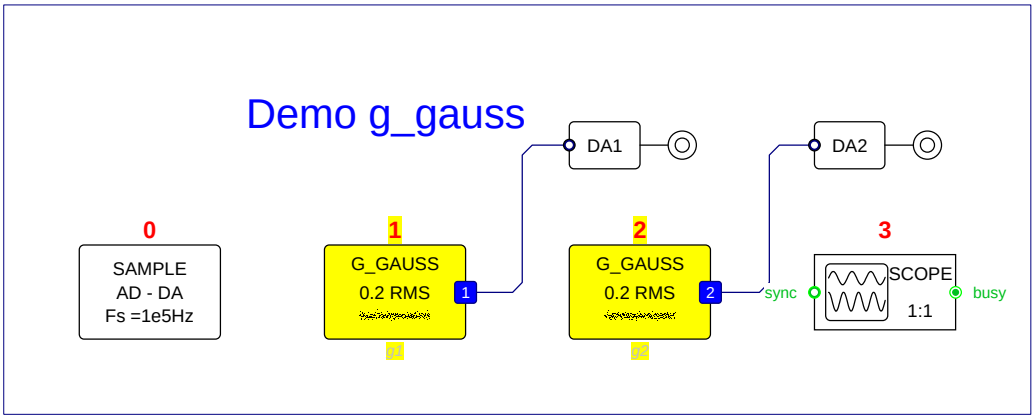
DESCRIPTION:
Gaussian Noise
Gaussian generator with Standart Deviation Sigma.
Nb acc determines generator precision. Choose 4 for power calculation, >= 25 for error rate calculation

PARAMETERS:

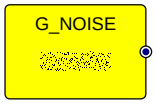
Parameter:	*Default values:*
sigma	0.2
Nb acc	4

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



G_GAUSS test program

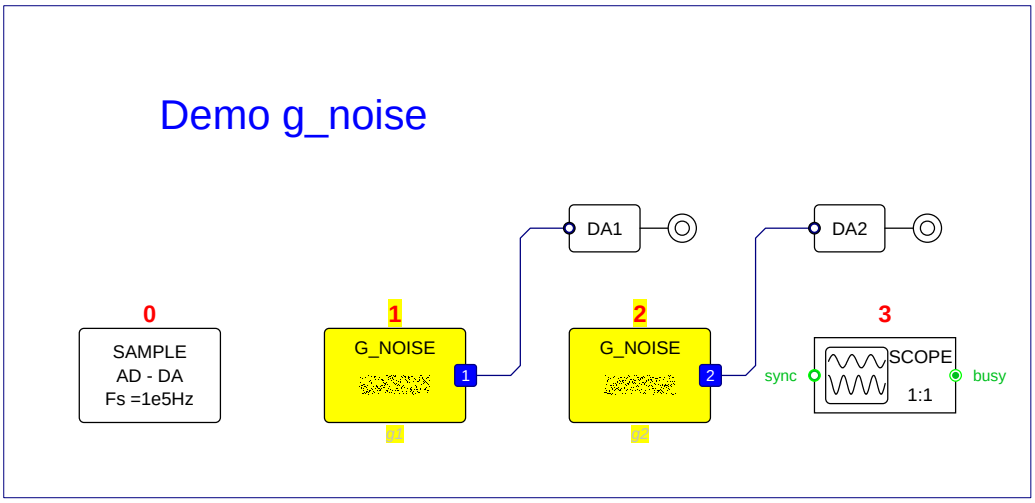


CATEGORY: Generators

DESCRIPTION:
Random generator
Uniformly distributed between -1.0 and +1.0

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

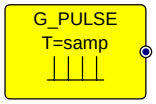


G_NOISE test program

G_PULSE

Pulse generator

G_PULSE



CATEGORY: Generators

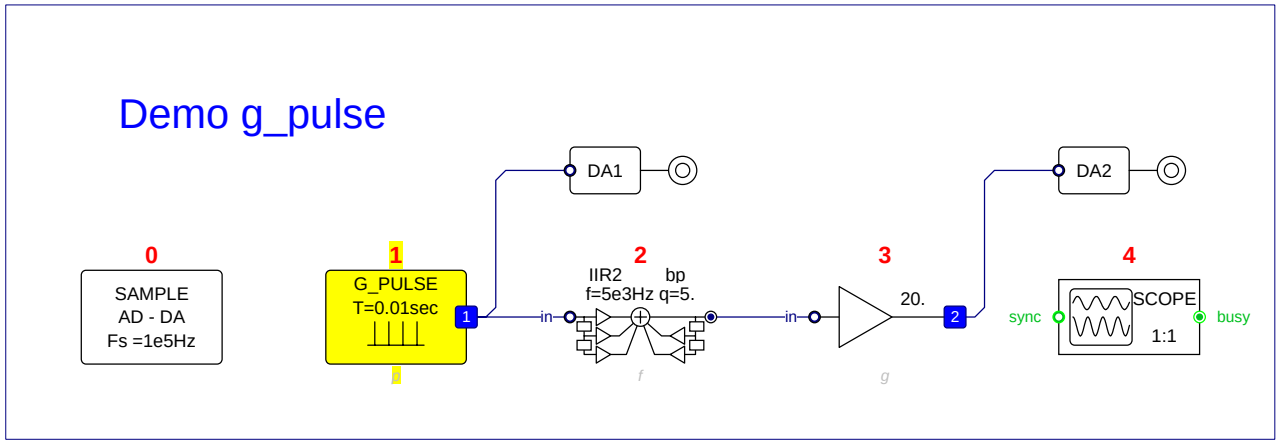
DESCRIPTION:
Pulse generator
Generates single or periodic pulses with amplitude 1.0
Period is expressed in samples. Single pulse if period = 0 .

PARAMETERS:

Parameter:	*Default values:*
period	0.001
Unit	sec,samp

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

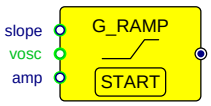


G_PULSE test program

G_RAMP

Slope generator

G_RAMP



CATEGORY: Continuous

DESCRIPTION:
Slope generator
Starts on SCOPE START command or on boolean VOSC signal
Slope and amplitude controlled byparameter values or by connecting optional inputs.

PARAMETERS:

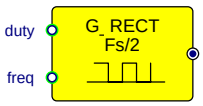
Parameter:	*Default values:*
Amplitude	1.0
Slope	1e-4
Abs=U/sec Rel=U/smp	abs,rel

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_vosc	BOOL	BIT	optional
name_slope	FRACT	WORD	optional
name_amp	FRACT	WORD	optional

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



CATEGORY: Generators

DESCRIPTION:
Rectangle generator
Modulable in frequency and duty cycle by connecting optional inputs.

PARAMETERS:

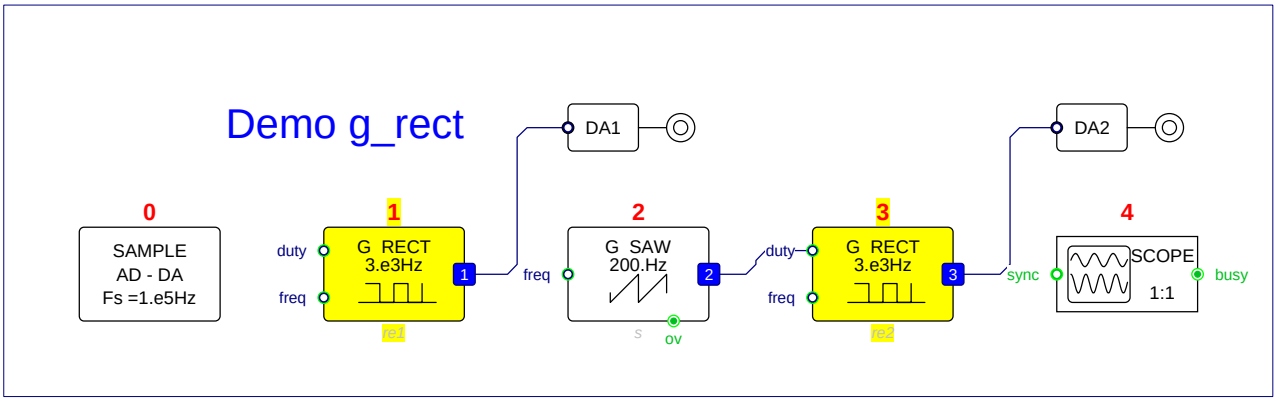
Parameter:	*Default values:*
Freq	1000.
Unit	Hz,Fs/2

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_duty	FRACT	WORD	optional
name_freq	FRACT	WORD	optional

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



G_RECT test program

G_RNDSYM Random symbols generator. G_RNDSYM

CATEGORY: Telecom

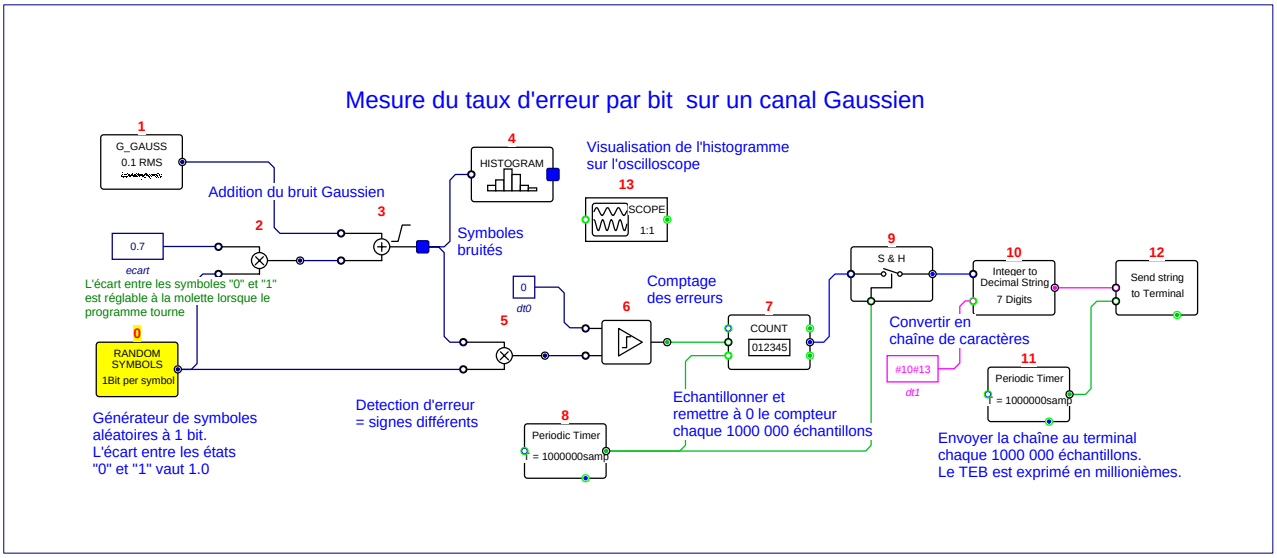
DESCRIPTION:
Random symbols generator.

PARAMETERS:

Parameter:	*Default values:*
Bits per symbol	1

OUTPUTS

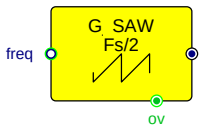
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



G_RNDSYM test program

Sawtooth generator

G_SAW



CATEGORY: Generators

DESCRIPTION:
Sawtooth generator
Opt input freq= 0..1 -> 0..Fs/2 overrides param if connected
Opt output ov= bool true at each discont.

PARAMETERS:

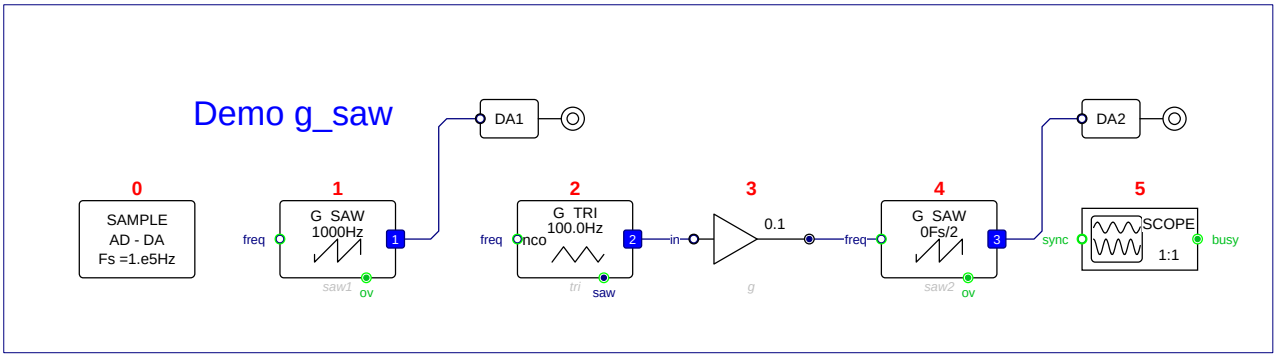
Parameter:	*Default values:*
Frequency	1000.
Unit	Hz,Fs/2

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_freq	FRACT	WORD	optional

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal
name_ov	BOOL	BIT	optional

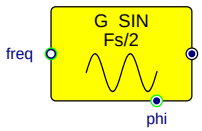


G_SAW test program

G_SIN

Sine wave generator

G_SIN



CATEGORY: Generators

DESCRIPTION:
Sine wave generator
Optional input: variable frequency 0..1 -> 0..Fs/2
Optional input if connected overrides parameter

PARAMETERS:

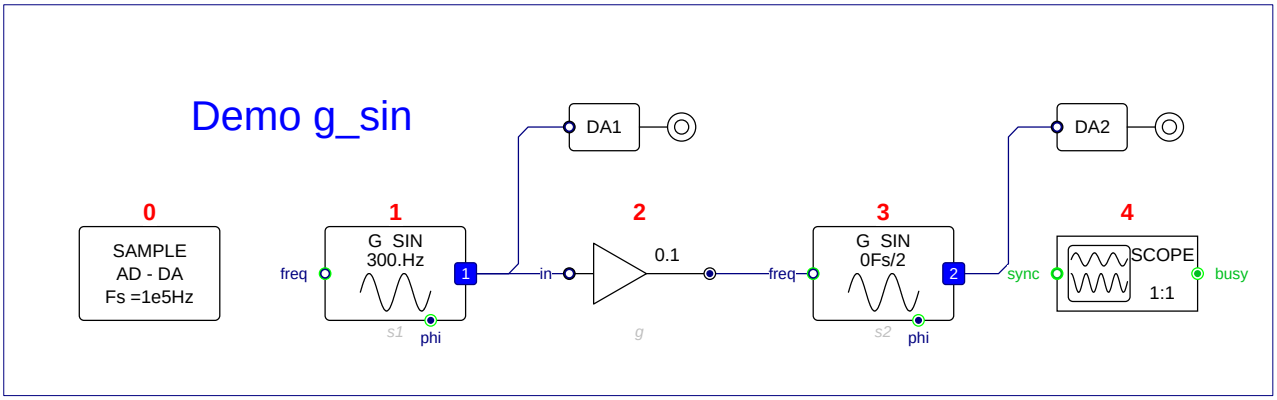
Parameter:	*Default values:*
Frequency	1000.
Unit	Hz,Fs/2

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_freq	FRACT	WORD	optional

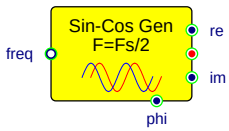
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal
name_phi	FRACT	DWORD	optional



G_SIN test program

G_SINCOS Sine-Cosine complex generator G_SINCOS



CATEGORY: Generators

DESCRIPTION:
Sine-Cosine complex generator
y(k) = exp(j2pikf/Fs)

PARAMETERS:

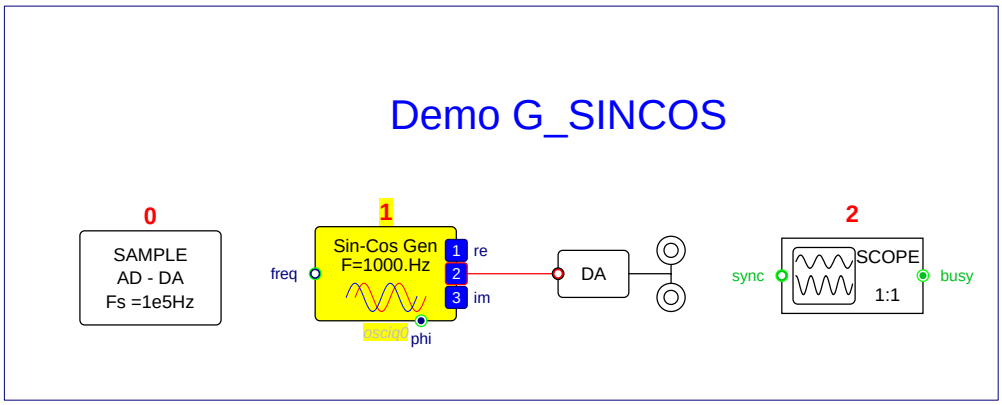
Parameter:	*Default values:*
freq	1000.
abs or rel	abs,rel

INPUTS

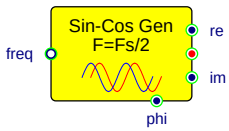
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_freq	FRACT	WORD	optional

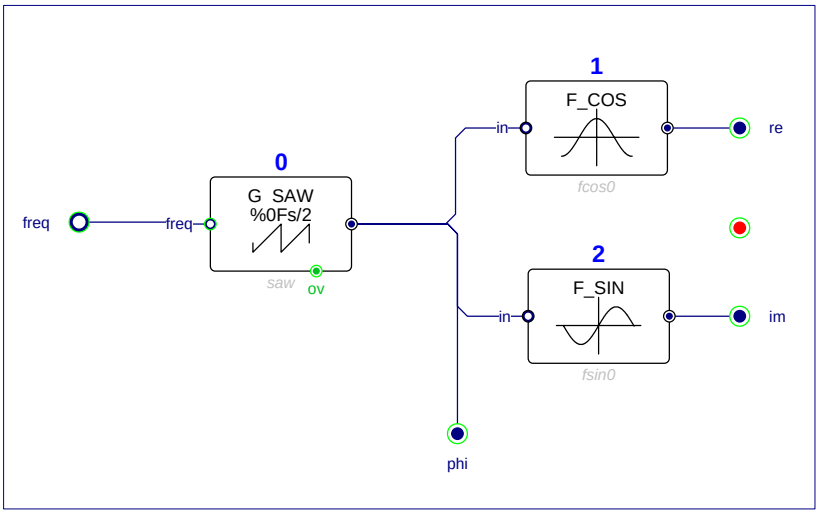
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_re	FRACT	WORD	optional
name_im	FRACT	WORD	optional
name_phi	FRACT	WORD	optional
name	COMPLEX	WORD	optional



G_SINCOS test program



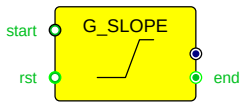


G_SINCOS internal schema

G_SLOPE

Triggered Slope Generator

G_SLOPE



CATEGORY: Generators

DESCRIPTION:
Triggered Slope Generator
Output is preset to Start Value on rst command
Output begins to ramp upwards or downwards on start command
Optional boolean output "end" is true after output has reached End Value

PARAMETERS:

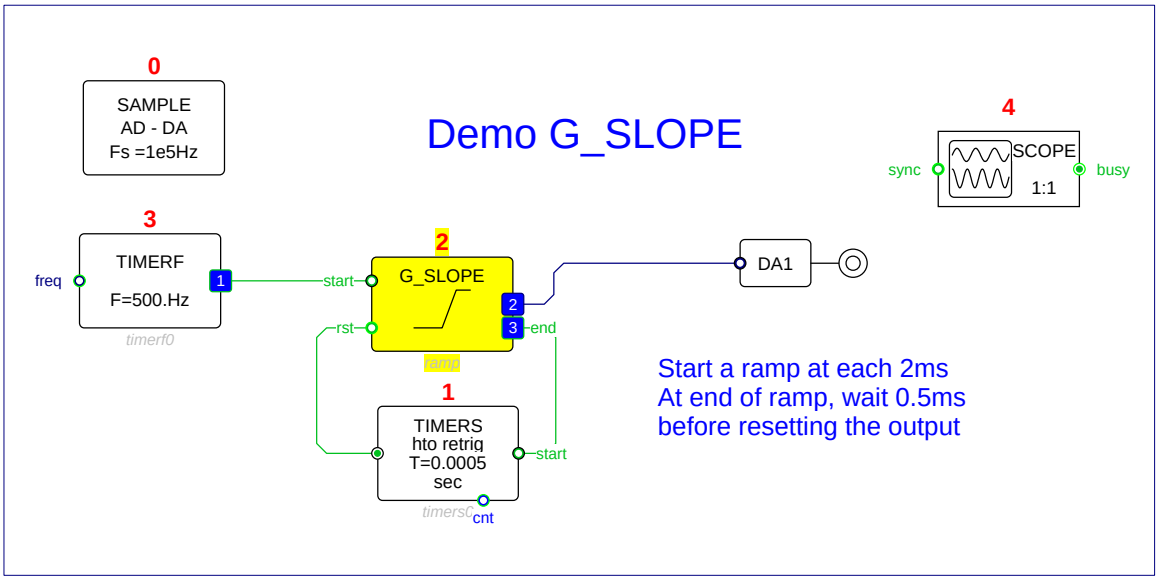
Parameter:	*Default values:*
Start value	-1.0
End value	1.0
Duration	0.001
Unit	sec,samp

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_rst	BOOL	BIT	optional
name_start	BOOL	BIT	mandatory

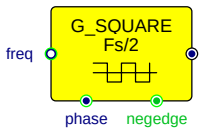
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	DWORD	normal
name_end	BOOL	BIT	optional



G_SLOPE test program

G_SQUARE Square wave generator G_SQUARE



CATEGORY: Generators

DESCRIPTION:
Square wave generator
Symmetric +1.0 / -1.0 output
Modulable in frequency (0.0 to 1.0 at freq input --> 0-Fs/2)
Optional fractional phase output
Optional negative edge boolean output

PARAMETERS:

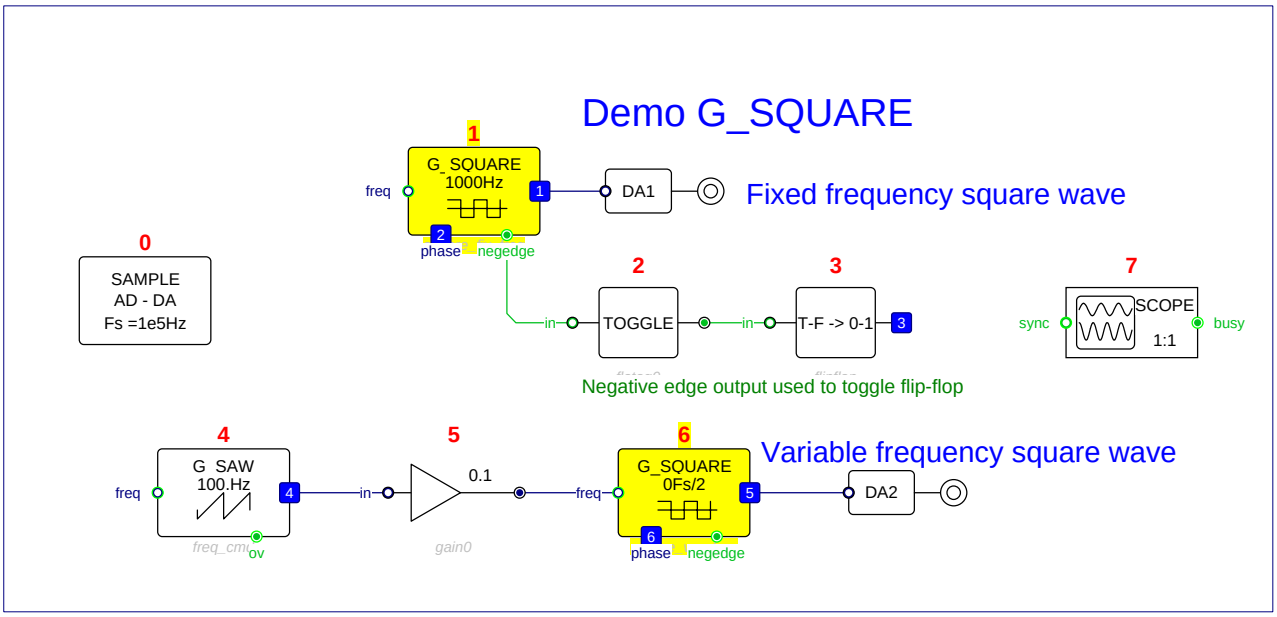
Parameter:	*Default values:*
Frequency	1000.
Unit	Hz,Fs/2

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_freq	FRACT	WORD	optional

OUTPUTS

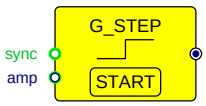
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal
name_phase	FRACT	WORD	optional
name_negedge	BOOL	BIT	optional



G_SQUARE test program

Step generator

G_STEP



CATEGORY: Continuous

DESCRIPTION:
Step generator
Starts on SCOPE START command or on boolean VOSC signal
Amplitude controlled by parameter value or by connecting optional input.

PARAMETERS:

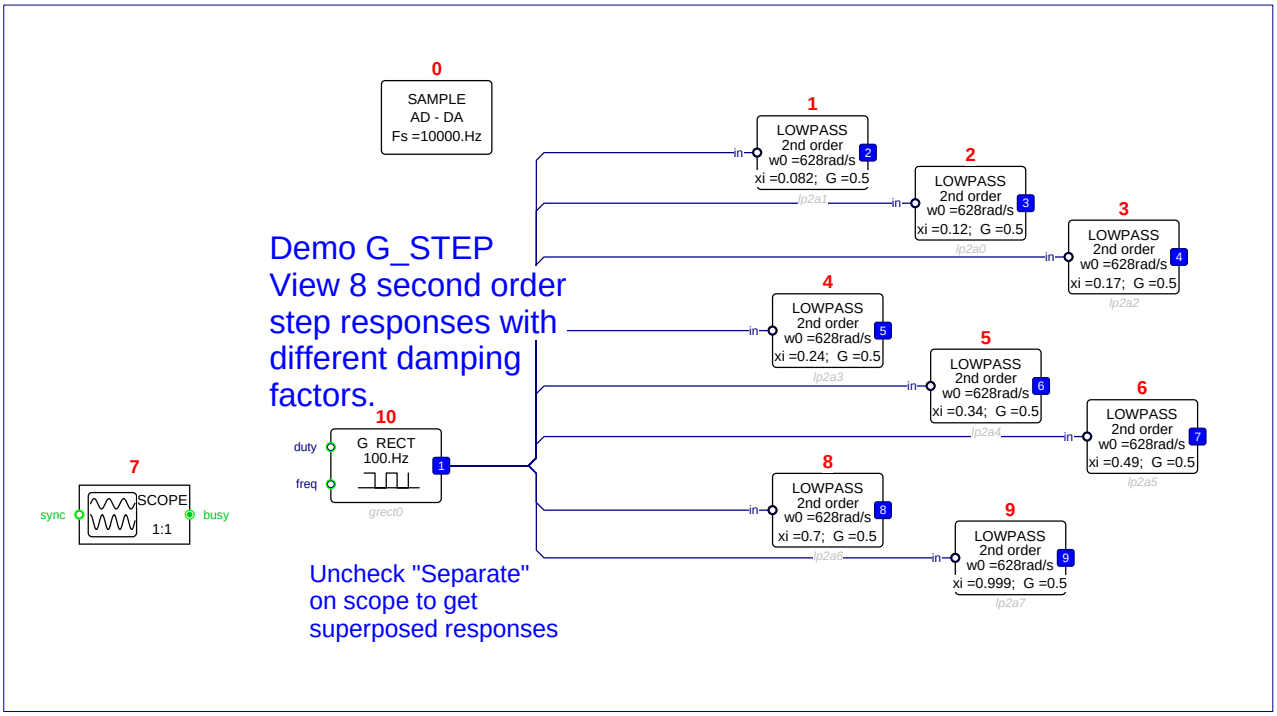
Parameter:	*Default values:*
Amplitude	1.0

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_sync	BOOL	BIT	optional
name_amp	FRACT	WORD	optional

OUTPUTS

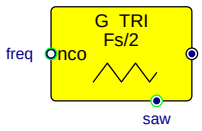
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



G_STEP test program

Triangle generator

G_TRI



CATEGORY: Generators

DESCRIPTION:
Triangle generator
Frequency modulable by connecting optional input.

PARAMETERS:

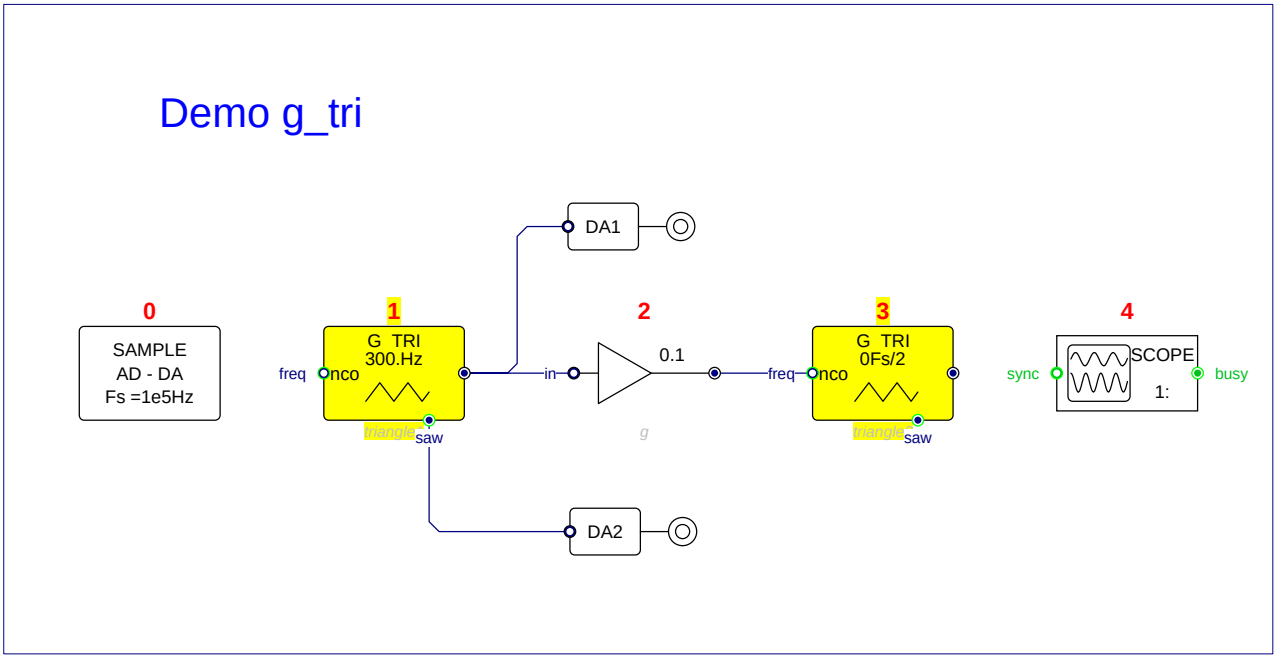
Parameter:	*Default values:*
Frequency	1000.
Unit	Hz,Fs/2

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_freq	FRACT	WORD	optional

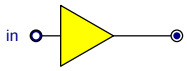
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal
name_saw	FRACT	WORD	optional



G_TRI test program

Fixed real gain



CATEGORY: Arithmetic

DESCRIPTION:
Fixed real gain
y(k) = g * x(k)

PARAMETERS:

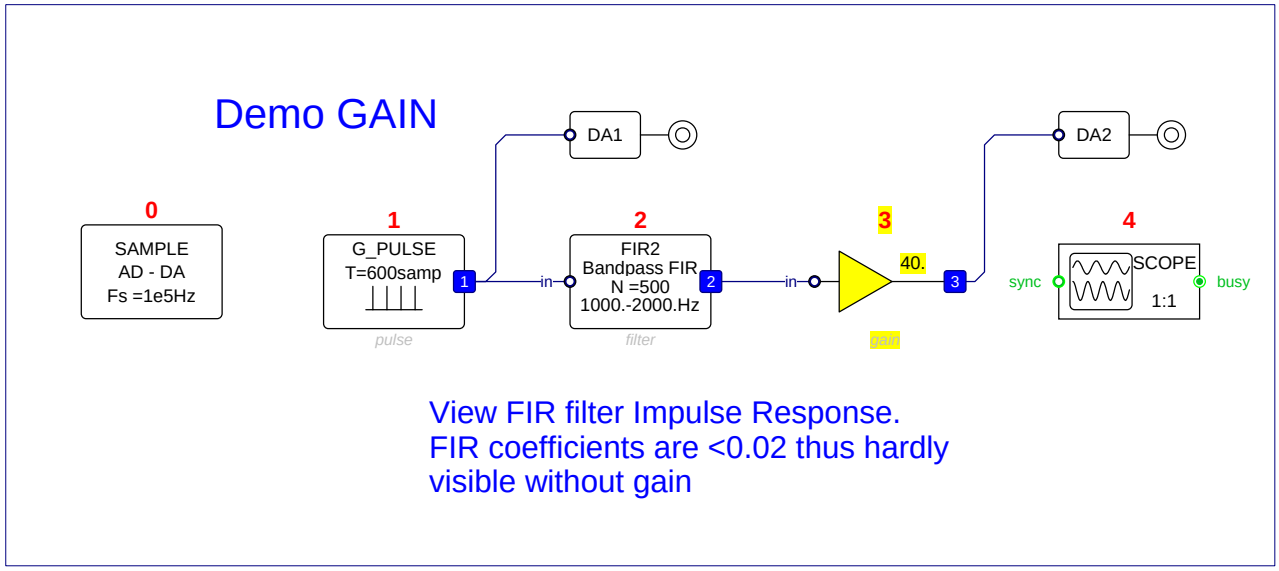
Parameter:	*Default values:*
Gain	10.

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

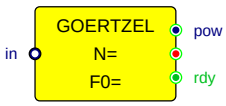


GAIN test program

GOERTZEL

Goertzel Algorithm

GOERTZEL



CATEGORY: Filters

DESCRIPTION:
Goertzel Algorithm
Gives one complex point of the N point DFT of input signal at desired frequency.
Output pow gives 4 * Power of complex output (used for tone detection)
a= 2pif0/Fs; w(0)=0 w(-1)=0
w(k) = x(k)/N + 2cos(a)w(k-1)-w(k-2) k=1..N
Complex out y = w(N)-exp(ja)w(N-1)
pow out = $4(w^2(N)+w^2(N-1)-2\cos(a)w(N)w(N-1))$

PARAMETERS:

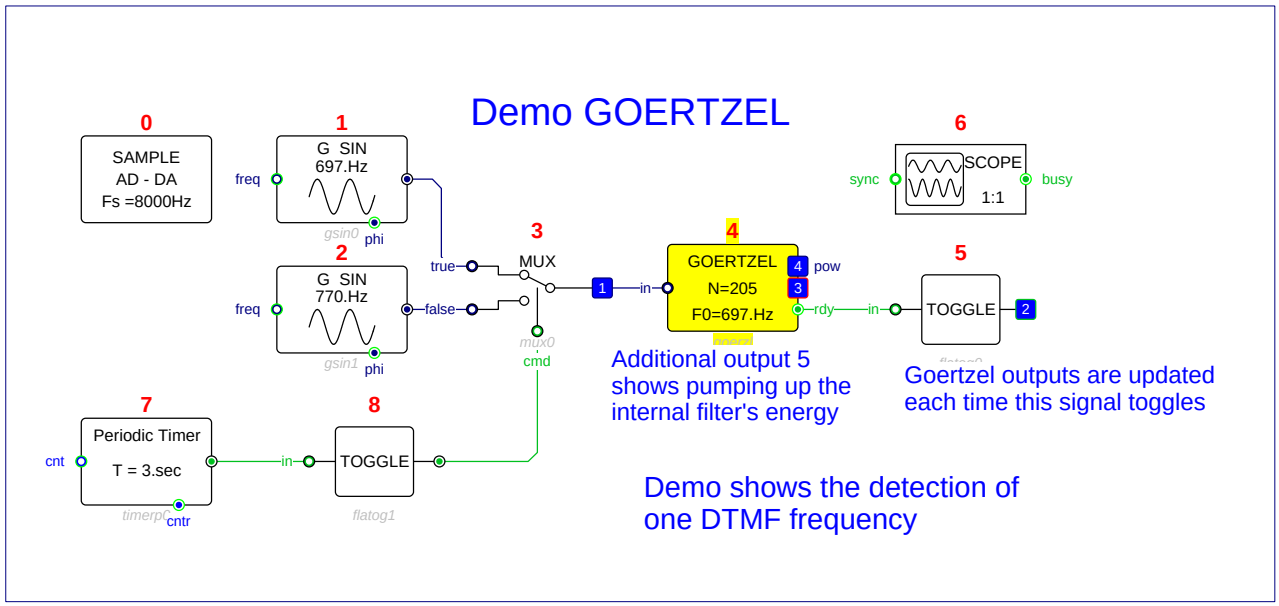
Parameter:	*Default values:*
Samples	1000
Frequency	1000.
Unit	Hz,Fs/2

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

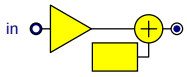
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_rdy	BOOL	BIT	optional
name	COMPLEX	WORD	optional
name_pow	FRACT	WORD	optional



GOERTZEL test program

Gain followed by offset



CATEGORY: Arithmetic

DESCRIPTION:
Gain followed by offset

PARAMETERS:

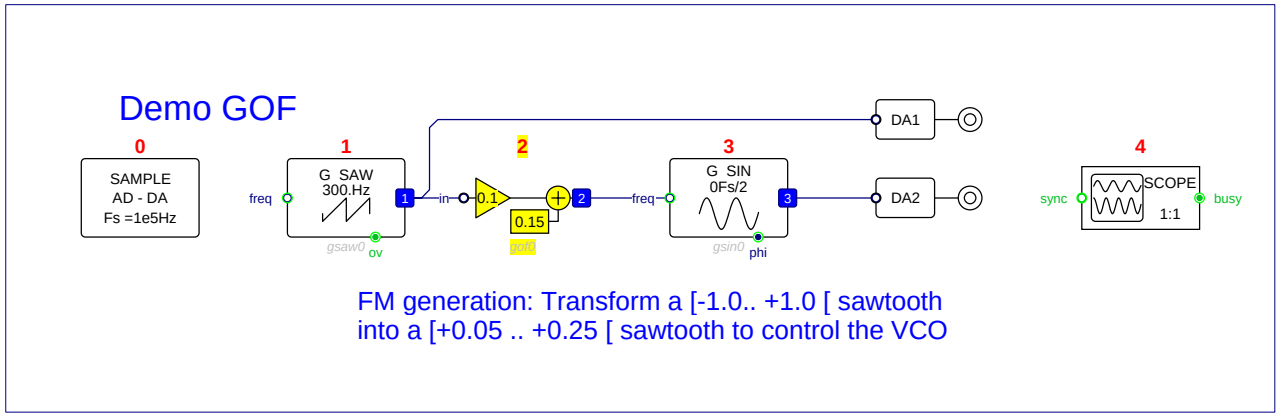
Parameter:	*Default values:*
gain	0.1
offset	0.1

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



GOF test program

HILBERT

Hilbert transform

HILBERT

CATEGORY: Filters

DESCRIPTION:
Hilbert transform
Transforms real signal into complex signal

PARAMETERS:

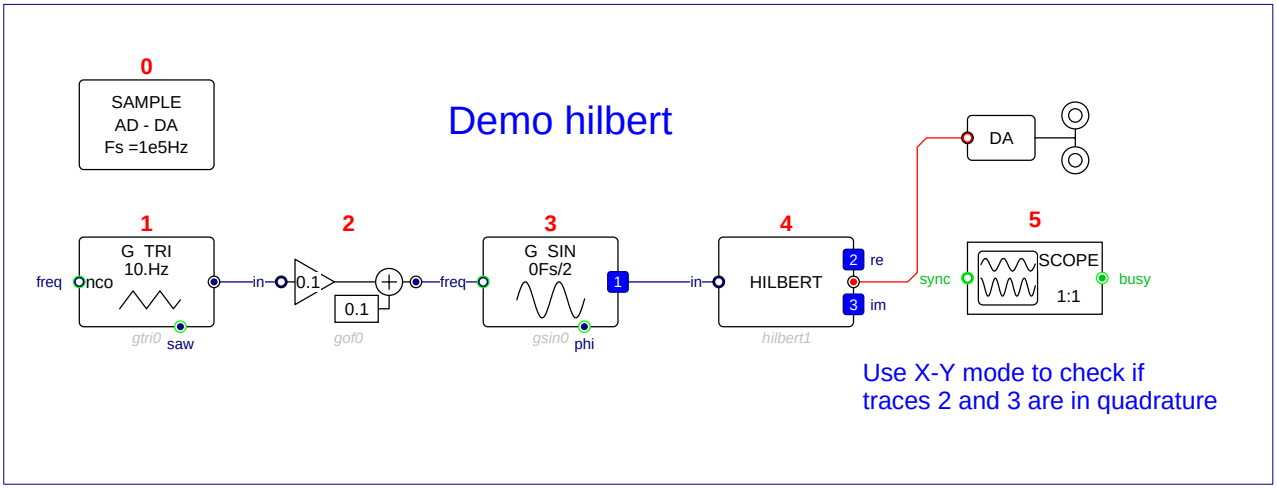
Parameter:	*Default values:*
number of taps	101

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	COMPLEX	WORD	normal
name_re	FRACT	WORD	optional
name_im	FRACT	WORD	optional

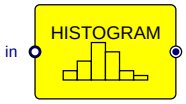


HILBERT test program

HISTO

Buffer switching histogram.

HISTO



CATEGORY: Instruments

DESCRIPTION:
Buffer switching histogram.
Output = histogram periodic scan with negative Sync pulse

PARAMETERS:

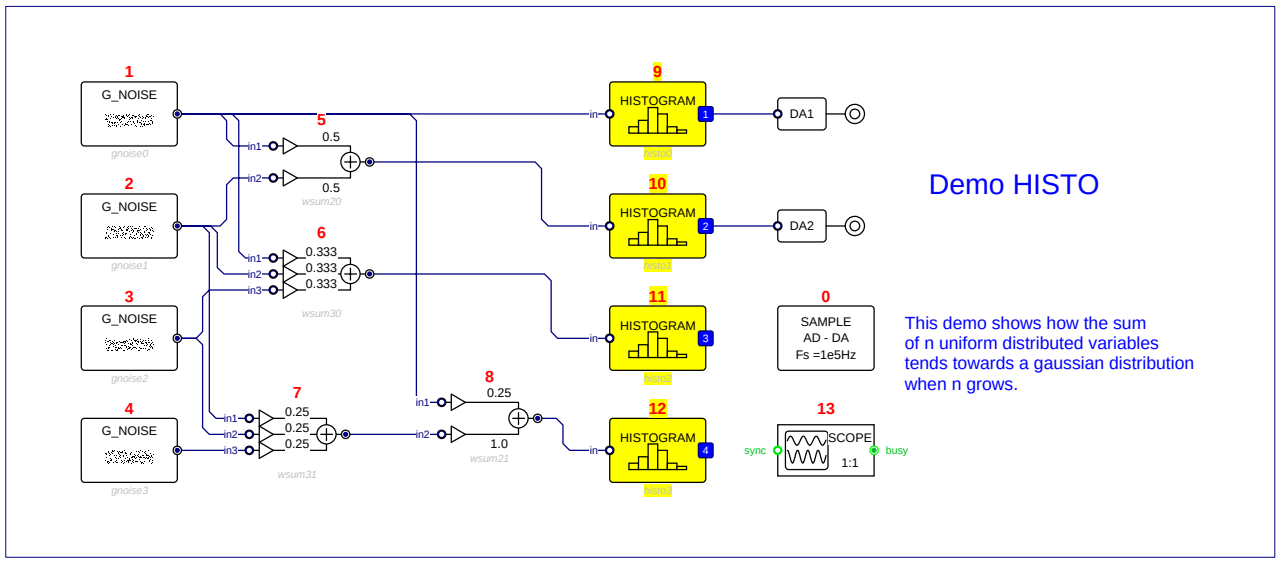
Parameter:	*Default values:*
Nb of classes	100
Nb samples total	100000
Gain	1.0

INPUTS

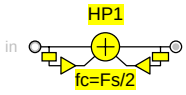
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



HISTO test program



CATEGORY: Filters

DESCRIPTION:
First order High-Pass filter

PARAMETERS:

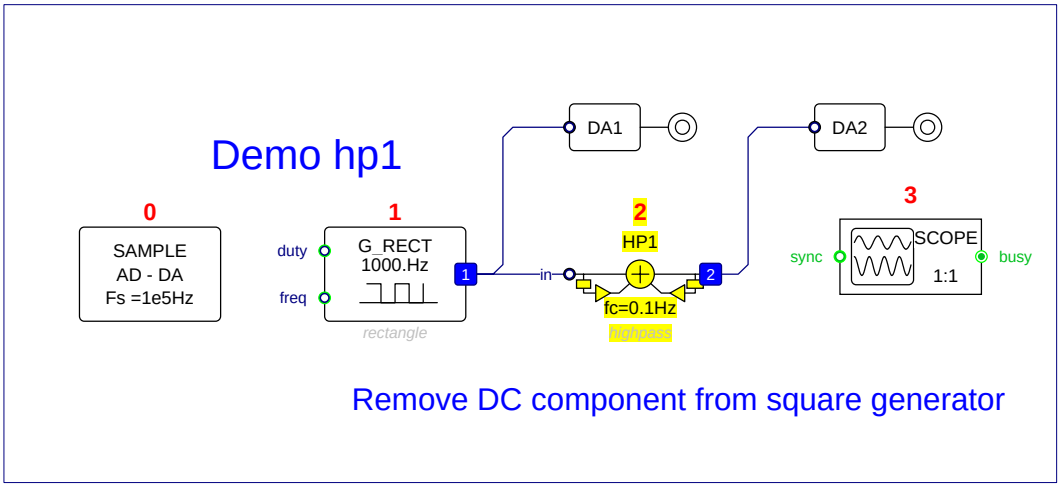
Parameter:	*Default values:*
Cutoff frequency	10.
Unit	Hz,Fs/2

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	defined by cn		mandatory

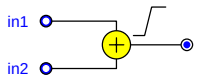
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	defined by cn		normal



HP1 test program

Integer addition with saturation



CATEGORY: Integer

DESCRIPTION:
Integer addition with saturation

INPUTS

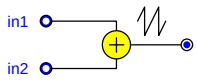
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in1	INTEGER	WORD	mandatory
name_in2	INTEGER	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	INTEGER	WORD	normal

Integer addition modulo 2^24

IADDV



CATEGORY: Integer

DESCRIPTION:
Integer addition modulo 2^24

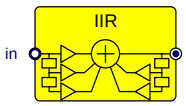
INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in1	INTEGER	WORD	mandatory
name_in2	INTEGER	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	INTEGER	WORD	normal

2nd order IIR filter



CATEGORY: Filters

DESCRIPTION:
2nd order IIR filter
Transfer function:
$(b0 + b1/z + b2/z^2)/(1 - a1/z - a2/z^2)$

PARAMETERS:

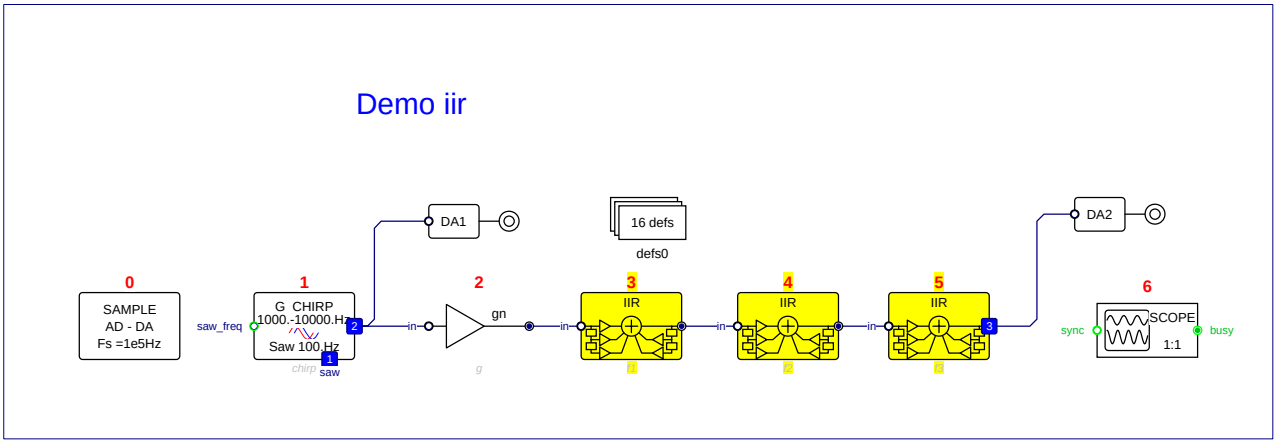
Parameter:	*Default values:*
b0	0.109
b1	0.218
b2	0.109
a1	1.056
a2	-0.493

INPUTS

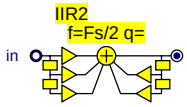
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



IIR test program



CATEGORY: Filters

DESCRIPTION:
2nd order recursive filter

PARAMETERS:

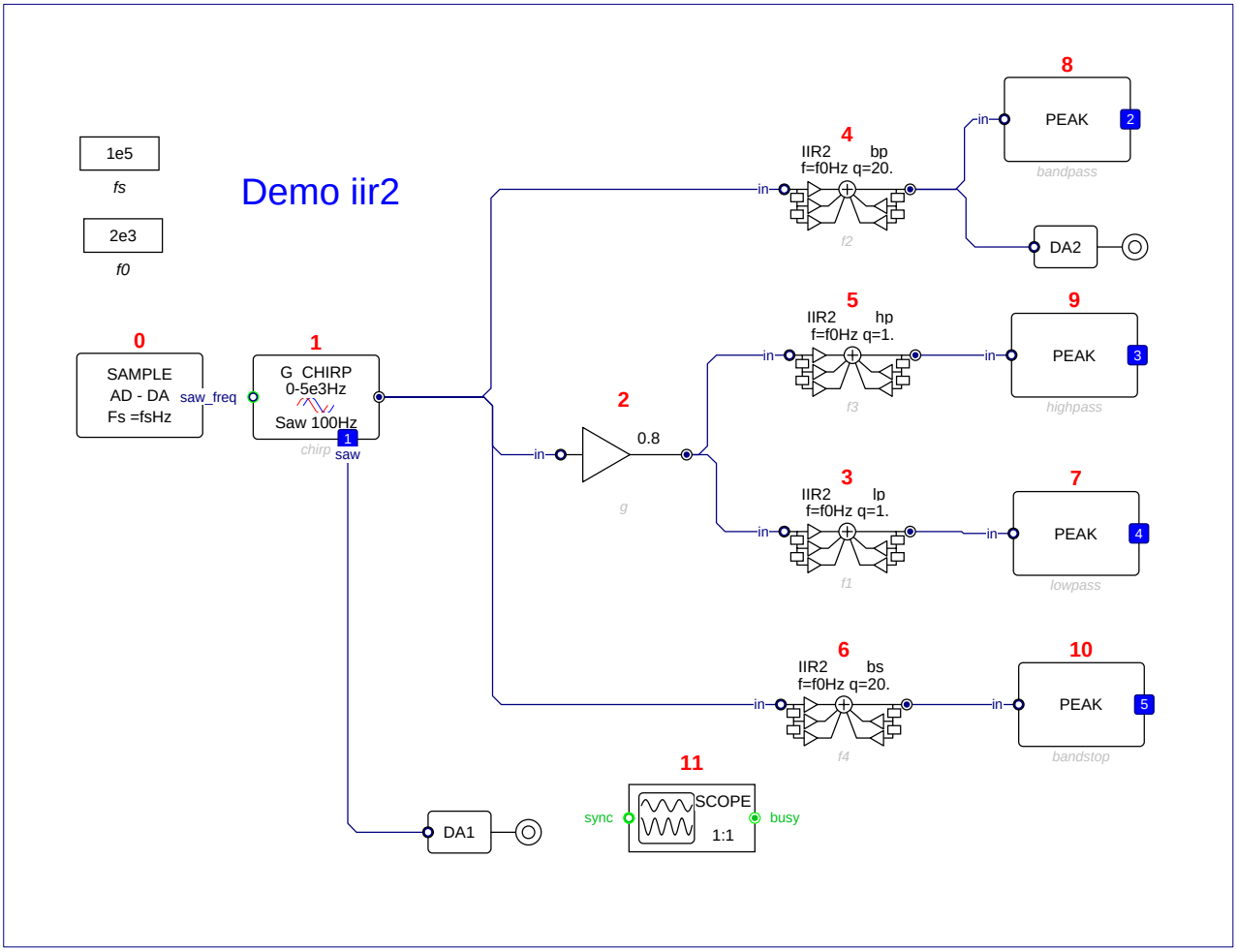
Parameter:	*Default values:*
Filter type	lp,bp,hp,bs
Frequency	1000.
Q factor	1.2
Unit	Hz,Fs/2

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

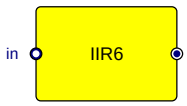
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



IIR2 test program

2nd order IIR filter

IIR6



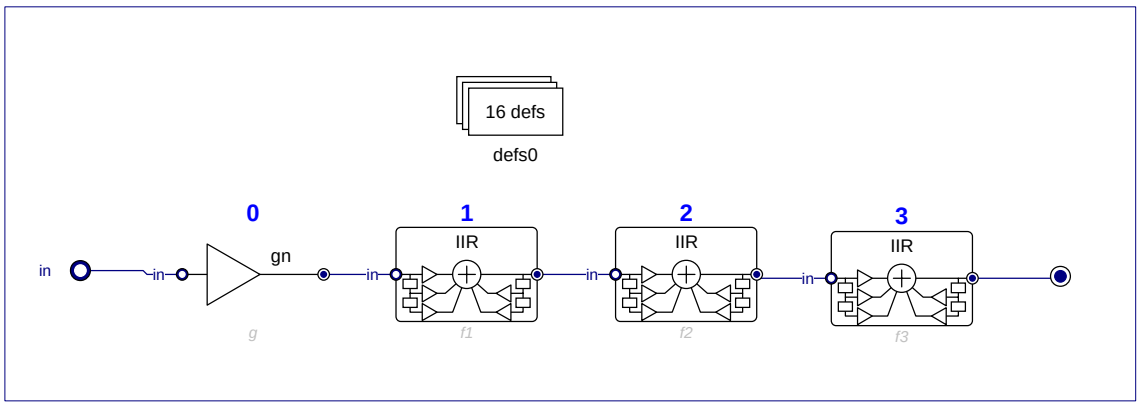
DESCRIPTION:
2nd order IIR filter
Transfer function:
$(b0 + b1/z + b2/z^2)/(1 - a1/z - a2/z^2)$

INPUTS

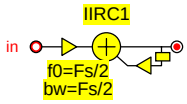
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



IIR6 internal schema



CATEGORY: Filters

DESCRIPTION:
1st order Bandpass complex IIR filter.

PARAMETERS:

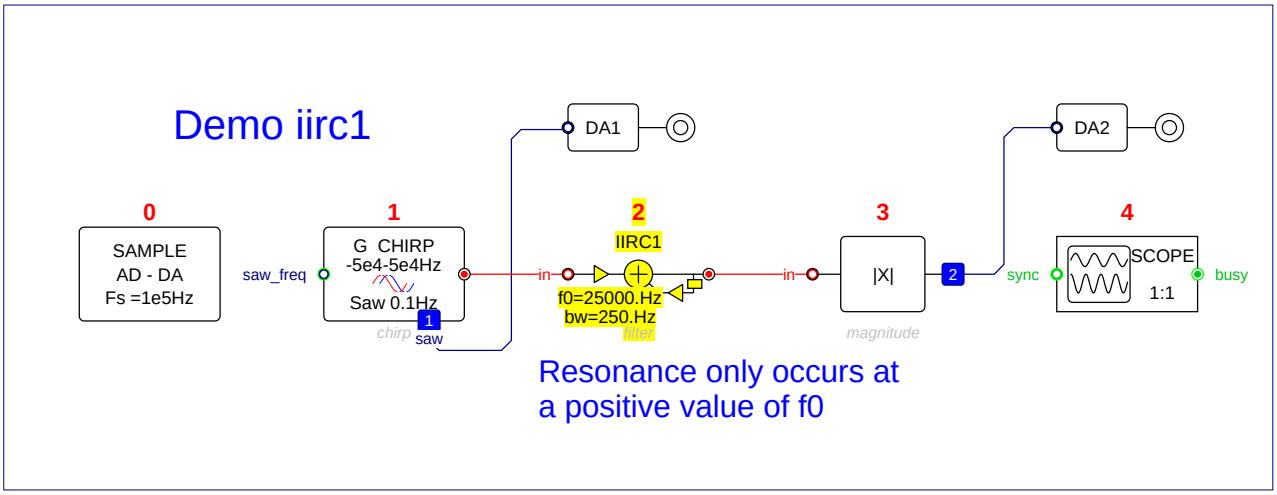
Parameter:	*Default values:*
Resonance frequency	1000.
Bandwidth	10.
Unit	Hz,Fs/2

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	COMPLEX	WORD	mandatory

OUTPUTS

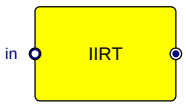
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	COMPLEX	WORD	normal



IIRC1 test program

2nd order IIR Transposed canonic form

IIRT



CATEGORY: Filters

DESCRIPTION:
2nd order IIR Transposed canonic form
$Y(z)=[b0 + b1/z + b2/z^2]/[1 + a1/z + a2/z^2]$

PARAMETERS:

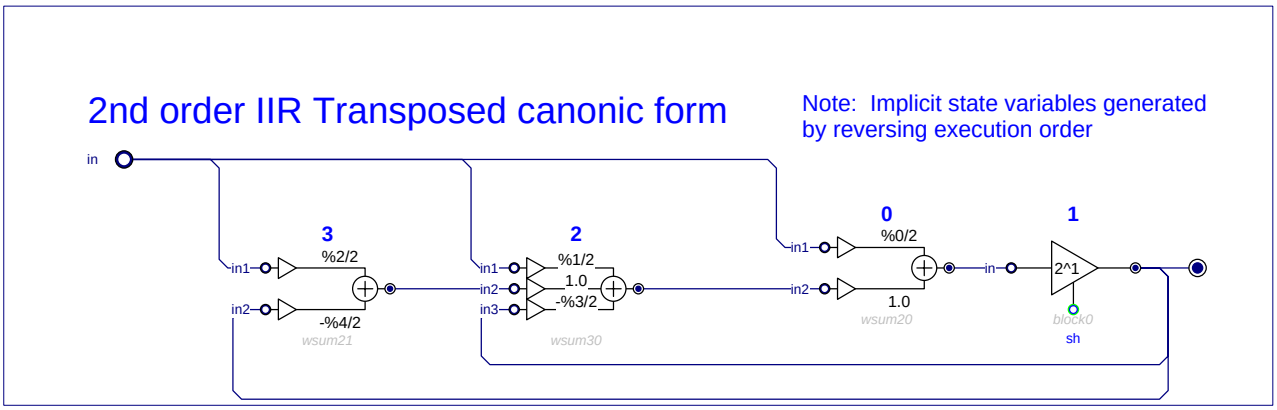
Parameter:	*Default values:*
b0	
b1	
b2	
a1	
a2	

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

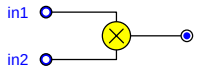


IIRT internal schema

IMUL

Integer multiplier

IMUL



CATEGORY: Integer

DESCRIPTION:
Integer multiplier

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in1	INTEGER	WORD	mandatory
name_in2	INTEGER	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	INTEGER	WORD	normal

IN_L

Codec input Left

IN_L

CATEGORY: Audio

DESCRIPTION:
Codec input Left
Result of A to D conversion

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

ATTRIBUTES
Non executable, Unique,

IN_R

Codec input Right

IN_R

CATEGORY: Audio

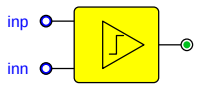
DESCRIPTION:
Codec input Right
Result of A to D conversion

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

ATTRIBUTES
Non executable, Unique,

INTCOMP Integer Comparator, boolean output INTCOMP



CATEGORY: Logic

DESCRIPTION:
Integer Comparator, boolean output
Result is True if inp > inn

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_inp	INTEGER	WORD	mandatory
name_inn	INTEGER	WORD	mandatory

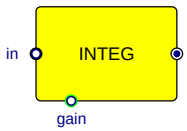
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	BOOL	BIT	normal

INTEG

Numerical integrator with input gain

INTEG



CATEGORY: Control

DESCRIPTION:
Numerical integrator with input gain
y(k) = y(k-1)+ g*x(k)

PARAMETERS:

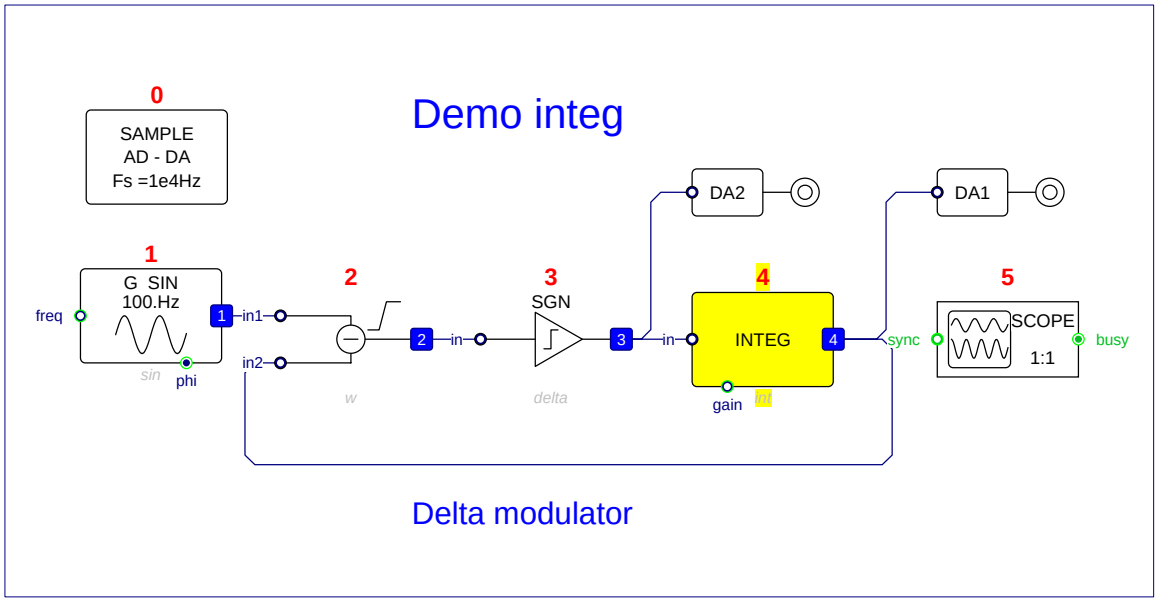
Parameter:	*Default values:*
gain	0.001

INPUTS

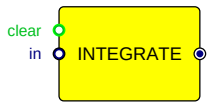
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory
name_gain	FRACT	WORD	optional

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



INTEG test program



CATEGORY: Continuous

DESCRIPTION:
Analog Integrator
y = Integral[gain*in(t)dt]
Gain 1.0 with input 1.0 generates slope 1.0 U/sec

PARAMETERS:

Parameter:	*Default values:*
Gain U/sec	1.0

INPUTS

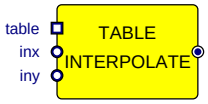
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_clear	BOOL	BIT	optional
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

1D or 2D Table Interpolate

INTERPOL



CATEGORY: Functions

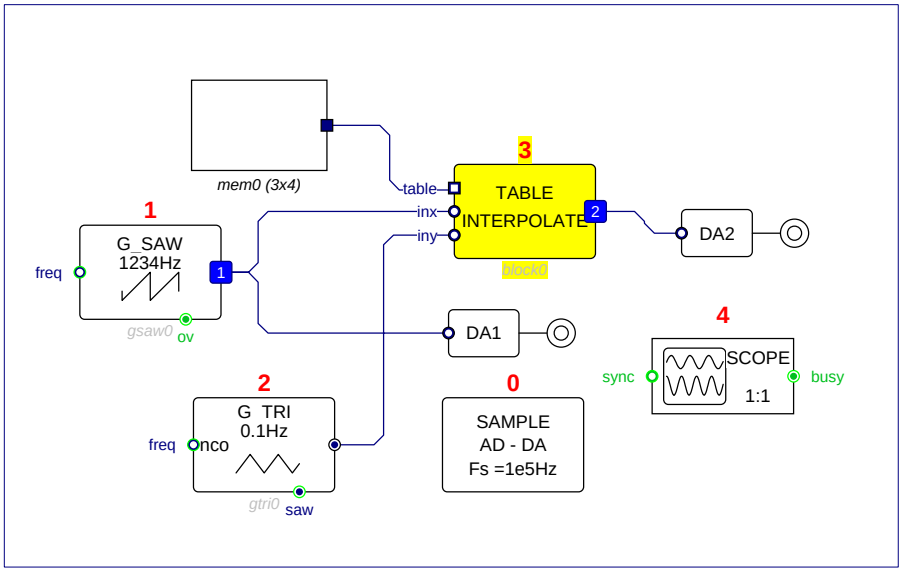
DESCRIPTION:
1D or 2D Table Interpolate
Inputs inx and iny define a continuous position in the table
Parameter "Signed" for X means input inx varies from -1.0 to +1.0 to scan one line of the table
Parameter "Unsigned" corresponds to a 0 .. +1.0 interval for input.

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_inx	FRACT	WORD	mandatory
name_table	FRACT	Matrix of WORD	mandatory
name_iny	FRACT	WORD	mandatory

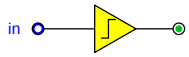
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



INTERPOL test program

INTTOBOOL Comparator of Integers INTTOBOOL



CATEGORY: Logic

DESCRIPTION:
Comparator of Integers
Result is TRUE if in is > ref
ref is defined by parameter

PARAMETERS:

Parameter:	*Default values:*
Ref level	0

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	INTEGER	WORD	mandatory

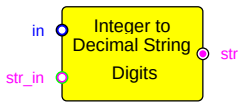
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	BOOL	BIT	normal

INTTOSTR

Integer to String

INTTOSTR



CATEGORY: String

DESCRIPTION:
Integer to String
Convert Integer input to decimal string

PARAMETERS:

Parameter:	*Default values:*
Nb digits (1-7)	7

INPUTS

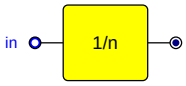
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	INTEGER	WORD	mandatory
name_str_in	STRING	WORD	optional

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_str	STRING	WORD	normal

Inverse of an integer

INVINT



CATEGORY: Integer

DESCRIPTION:
Inverse of an integer
Result is fractionnal.

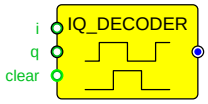
INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	INTEGER	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

IQ_DECODER Incremental decoder IQ_DECODER



CATEGORY: Logic

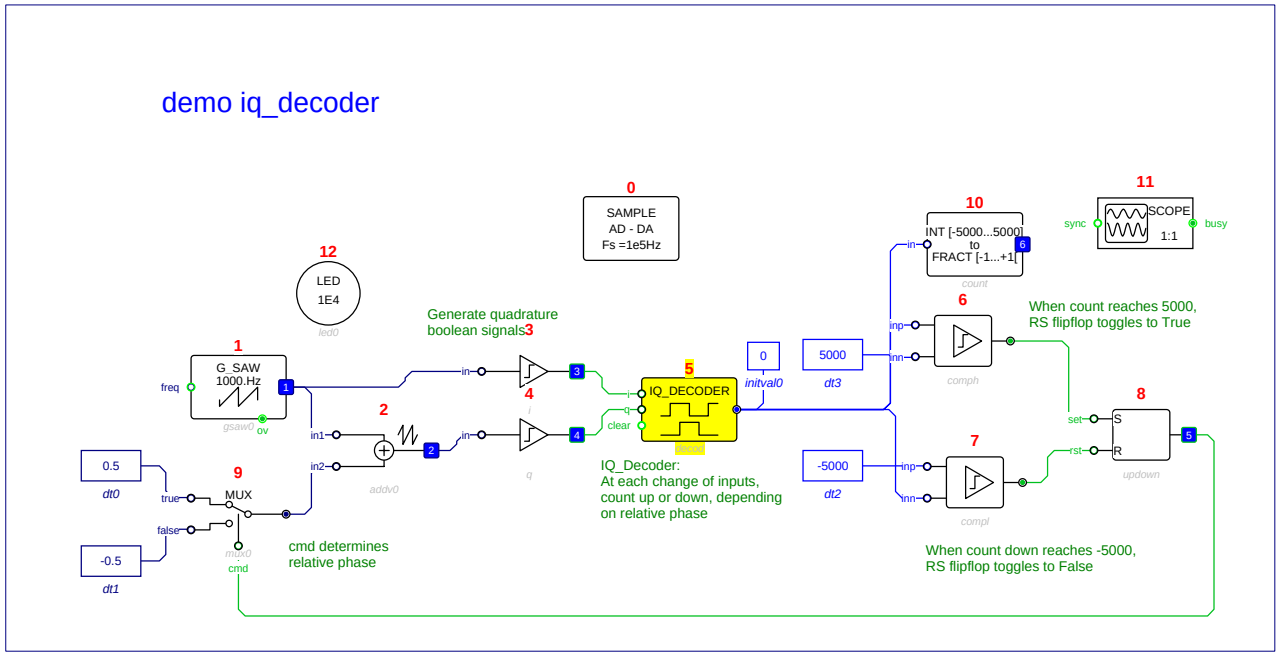
DESCRIPTION:
Incremental decoder
for phase-quadrature logic signals

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_i	BOOL	BIT	mandatory
name_q	BOOL	BIT	mandatory
name_clear	BOOL	BIT	optional

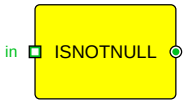
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	INTEGER	WORD	normal



IQ_DECODER test program

ISNOTNULL Test line boolean vector for non nul ISNOTNULL



DESCRIPTION:
Test line boolean vector for non nul

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	BOOL	Matrix of BIT	mandatory

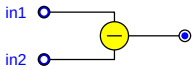
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	BOOL	BIT	normal

ISUB

Integer Subtraction

ISUB



CATEGORY: Integer

DESCRIPTION:
Integer Subtraction

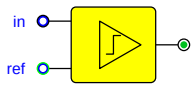
INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in1	INTEGER	WORD	mandatory
name_in2	INTEGER	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	INTEGER	WORD	normal

ITOBOOL Comparator, boolean output ITOBOOL



CATEGORY: Integer

DESCRIPTION:
Comparator, boolean output
Result is True if in is > ref (ref connected)
Result is True if in is > 0 (ref unconnected)

PARAMETERS:

Parameter:	*Default values:*
Ref level	0

INPUTS

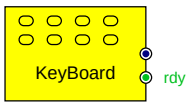
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	INTEGER	WORD	mandatory
name_ref	INTEGER	WORD	optional

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	BOOL	BIT	normal

Get ASCII from keyboard

KBD



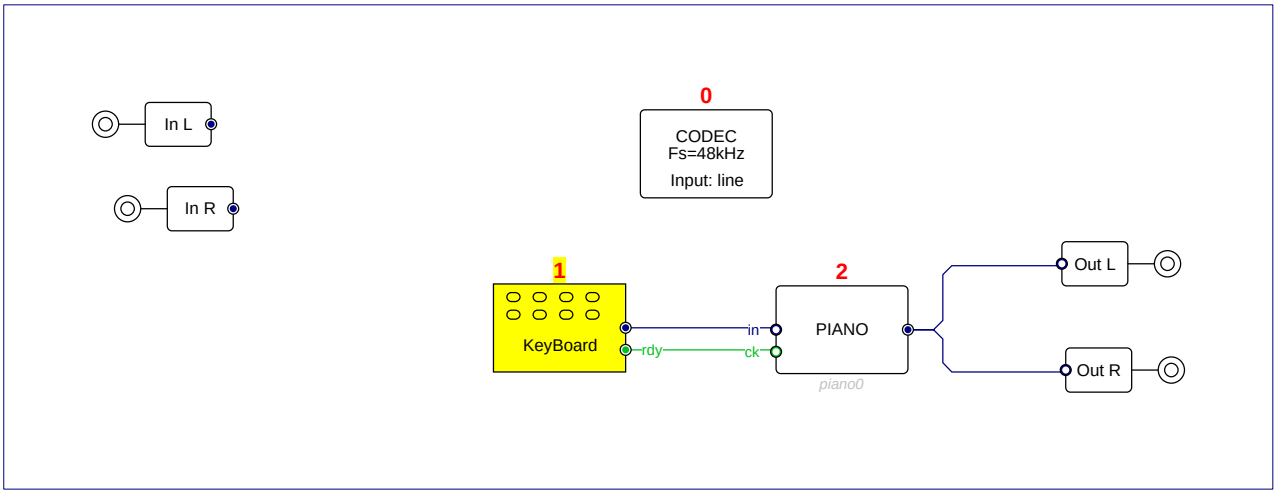
CATEGORY: String

DESCRIPTION:
Get ASCII from keyboard

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal
name_rdy	BOOL	BIT	normal

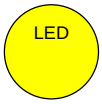
ATTRIBUTES
Unique,



KBD test program

Core activity LED

LED

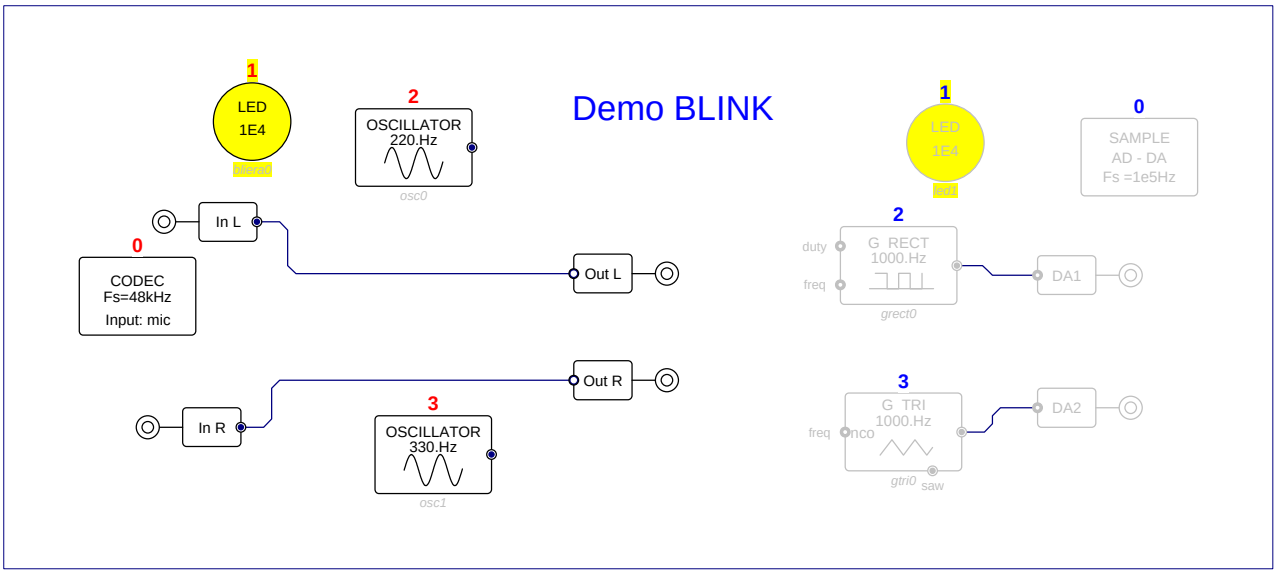


CATEGORY: Timing

DESCRIPTION:
Core activity LED
Set LED ON or OFF or twinkling at given frequency

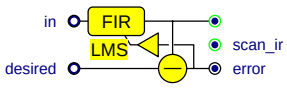
PARAMETERS:

Parameter:	*Default values:*
ON OFF or Period (*Ts)	ON,OFF,1E5



LED test program

Auto Adaptive FIR filter.



CATEGORY: Filters

DESCRIPTION:
Auto Adaptive FIR filter.
Impulse response evolutes in order to minimize error (Least Mean Square algorithm).

PARAMETERS:

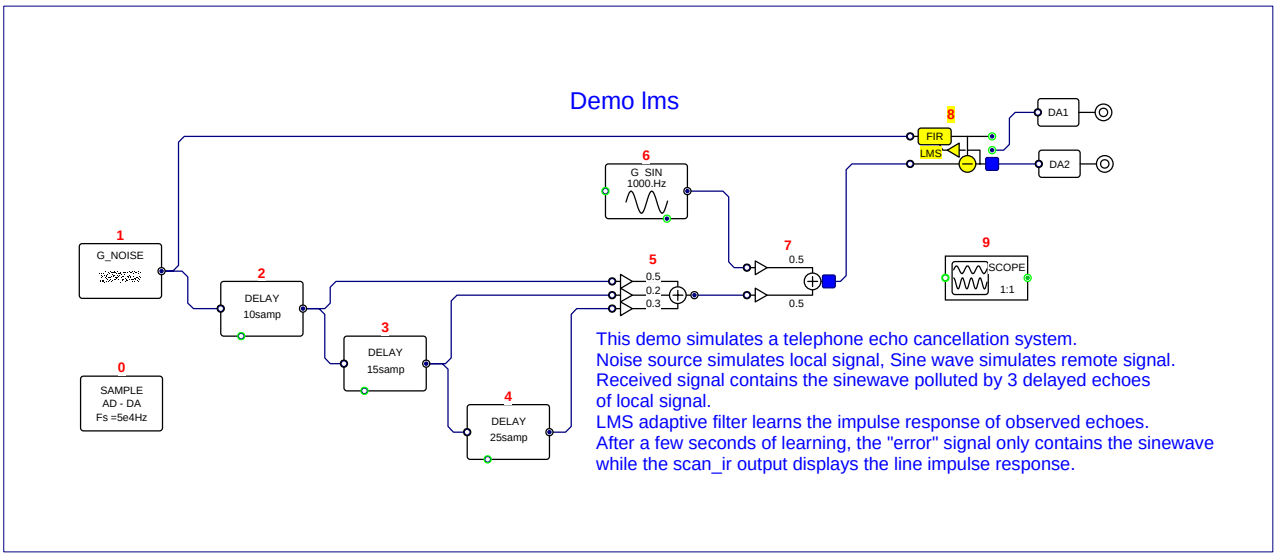
Parameter:	*Default values:*
size	100
gain	1e-5

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory
name_desired	FRACT	WORD	mandatory

OUTPUTS

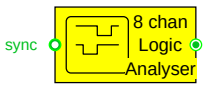
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	optional
name_scan_ir	FRACT	WORD	optional
name_error	FRACT	WORD	normal



LMS test program

1-8 Channel Logic analyser

LOGAN



CATEGORY: Instruments

DESCRIPTION:
1-8 Channel Logic analyser
Inputs are bits 0..7 of port H

PARAMETERS:

Parameter:	*Default values:*
Samples	500
Inputs	8

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_sync	BOOL	BIT	optional

OUTPUTS

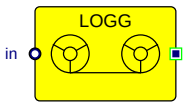
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	BOOL	BIT	optional

ATTRIBUTES
Unique,

LOGG

Data Logger.

LOGG



CATEGORY: Instruments

DESCRIPTION:
Data Logger.
Stores any data into a cyclic buffer for test purpose.

PARAMETERS:

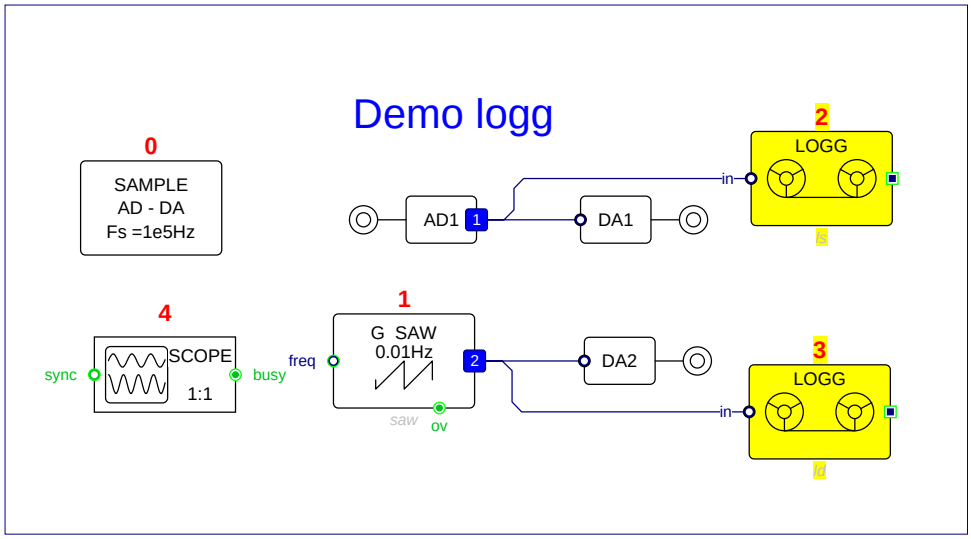
Parameter:	*Default values:*
Memory field	x,y,l,p
Size of buffer	1000

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	Matrix of WORD	optional

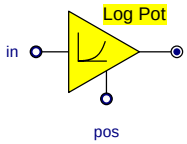


LOGG test program

LOGPOT

Log potentiometer

LOGPOT



CATEGORY: Audio

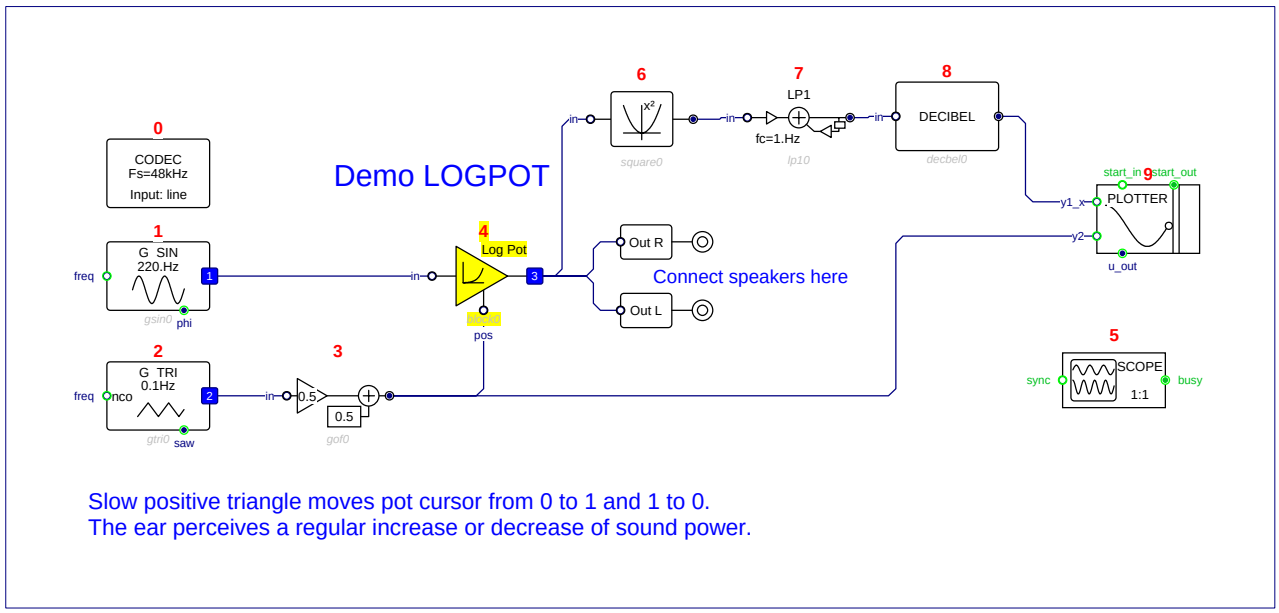
DESCRIPTION:
Log potentiometer
Audio perceived power is proportional to position

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory
name_pos	FRACT	WORD	mandatory

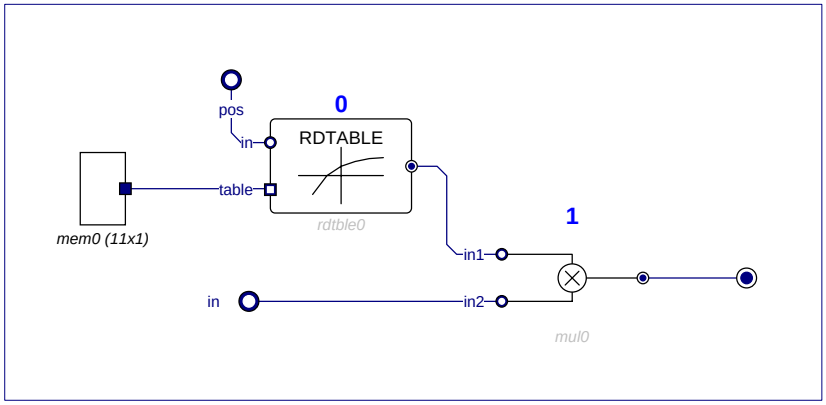
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



Slow positive triangle moves pot cursor from 0 to 1 and 1 to 0.
The ear perceives a regular increase or decrease of sound power.

LOGPOT test program

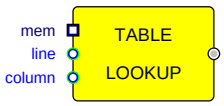


LOGPOT internal schema

LOOKUP

Read data in x:, y:, l:, or p: memory

LOOKUP



CATEGORY: Control

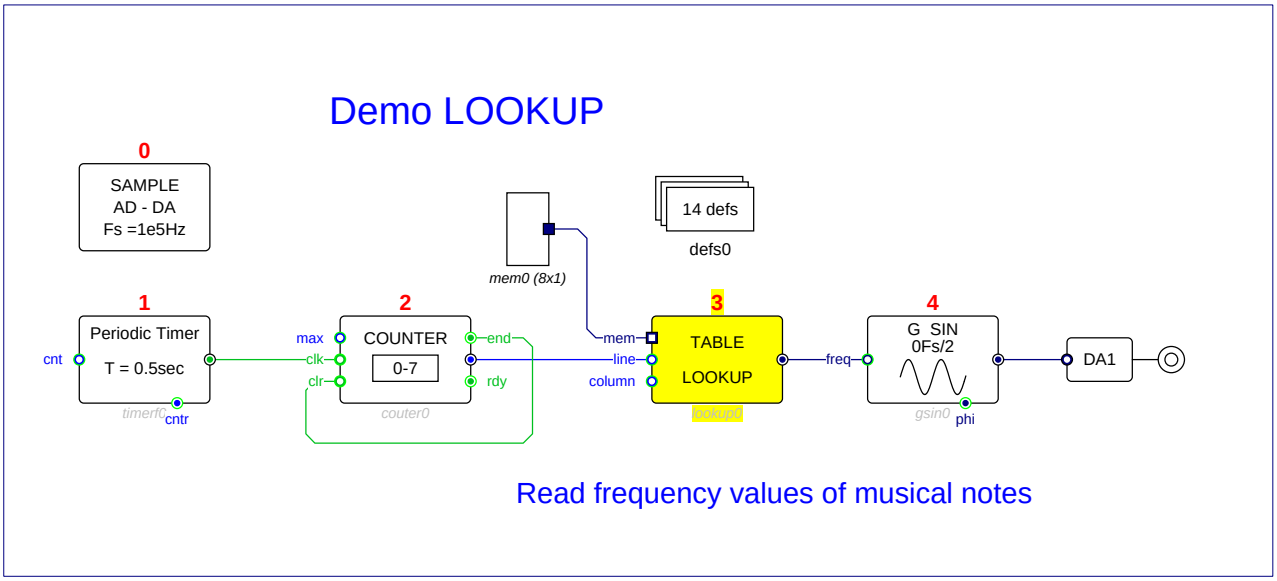
DESCRIPTION:
Read data in x:, y:, l:, or p: memory
Input gives address; Output may be single or double.

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_mem	FRACT	Matrix of WORD	mandatory
name_line	INTEGER	WORD	optional
name_column	INTEGER	WORD	optional

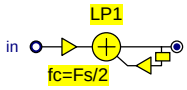
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	defined by cn		normal



LOOKUP test program

1st order recursive lowpass filter



CATEGORY: Filters

DESCRIPTION:
1st order recursive lowpass filter

PARAMETERS:

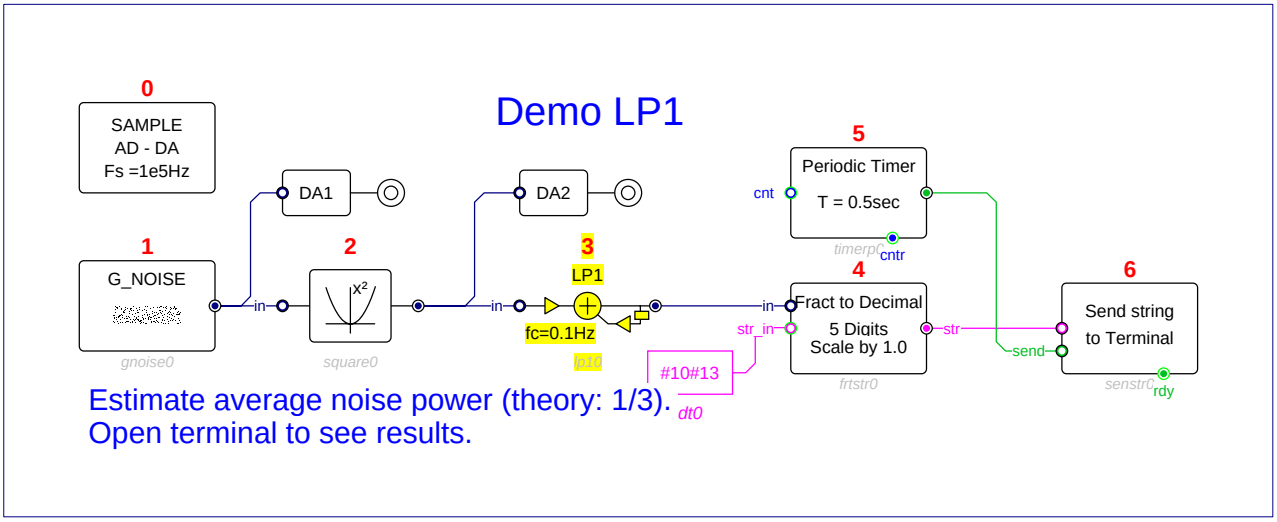
Parameter:	*Default values:*
Cutoff frequency	1.
Unit	Hz,Fs/2

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	DWORD	normal

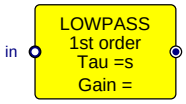


LP1 test program

LP1A

1st order lowpass

LP1A



CATEGORY: Continuous

DESCRIPTION:
1st order lowpass
Y(p)/X(p)= G / (1+Tp)

PARAMETERS:

Parameter:	*Default values:*
Time cst. (sec)	1.0
Gain	1.0

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

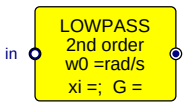
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

LP2A

2nd order lowpass

LP2A



CATEGORY: Continuous

DESCRIPTION:
2nd order lowpass
$Y(p)/X(p)= w0^2/(p^2 +2xiw0p +w0^2)$

PARAMETERS:

Parameter:	*Default values:*
w0 rad/s	1.0
xi	0.3
Gain	0.5

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

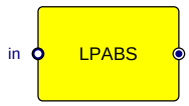
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

LPABS

Lowpass of abs value

LPABS



CATEGORY: Filters

DESCRIPTION:
Lowpass of abs value

PARAMETERS:

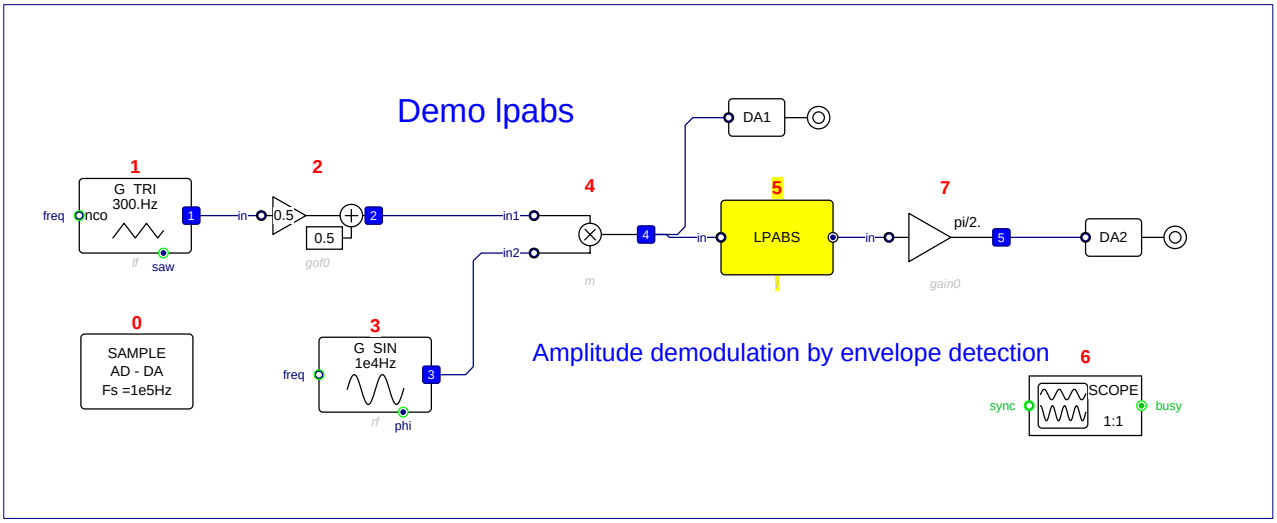
Parameter:	*Default values:*
Cutoff frequency	1.
Unit	Hz,Fs/2

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

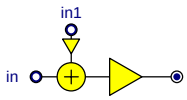


LPABS test program

MADD

Multiply and Add

MADD



CATEGORY: Arithmetic

DESCRIPTION:
Multiply and Add
Add weighted input, optionally shift result
y=(in+Gi*in1)*Gs
This is an IIR filter primitive

PARAMETERS:

Parameter:	*Default values:*
Gi	1.0
Gs:	2,1,0.5

INPUTS

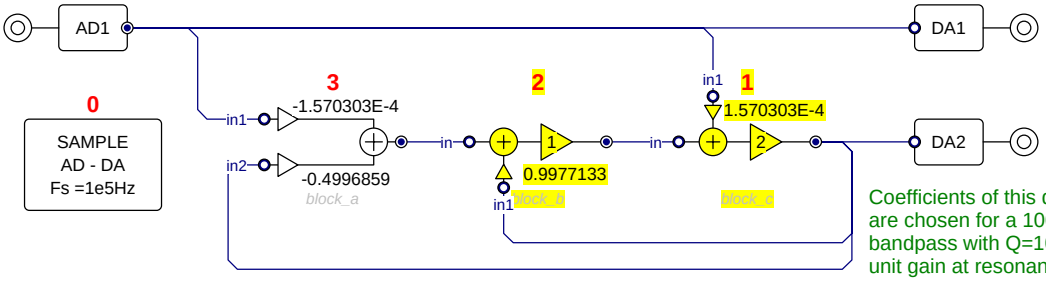
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in1	FRACT	WORD	mandatory
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

Demo WMAC

2nd order IIR Bandpass Filter
Transposed Canonic Form



Coefficients of this demo are chosen for a 1000Hz bandpass with Q=100 and unit gain at resonance

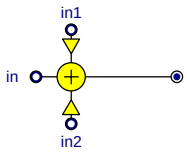
NOTE -- Implicit state variables:
Reversing execution order produces one unit delay between block_a and bloc_b, and also one unit delay between block_b and block_c. These two delay cells constitute the filters's state variables.

MADD test program

MADD2

Multiply and add 2 inputs

MADD2



CATEGORY: Arithmetic

DESCRIPTION:
Multiply and add 2 inputs
Add 2 weighted inputs to input
y = in+in1*g1+in2*g2
This is an IIR filter primitive

PARAMETERS:

Parameter:	*Default values:*
G1	1.0
G2	1.0

INPUTS

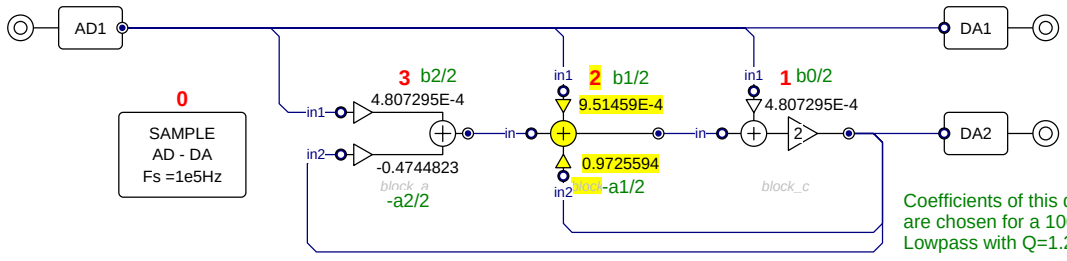
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in1	FRACT	WORD	mandatory
name_in	FRACT	WORD	mandatory
name_in2	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

Demo WMAC2,WMAC

2nd order IIR Filter
H(z)=(b0+b1/z+b2/z²)/(1+a1/z+a2/z²)
Transposed Canonic Form



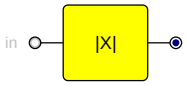
NOTE -- Implicit state variables:
Reversing execution order produces one unit delay between block WSUM2 and bloc WMAC2 and also one unit delay between blocks WMAC2 and WMAC. These two delay cells constitute the filters's state variables.

MADD2 test program

MAGN

Magnitude of a real or complex input

MAGN



CATEGORY: Non linear

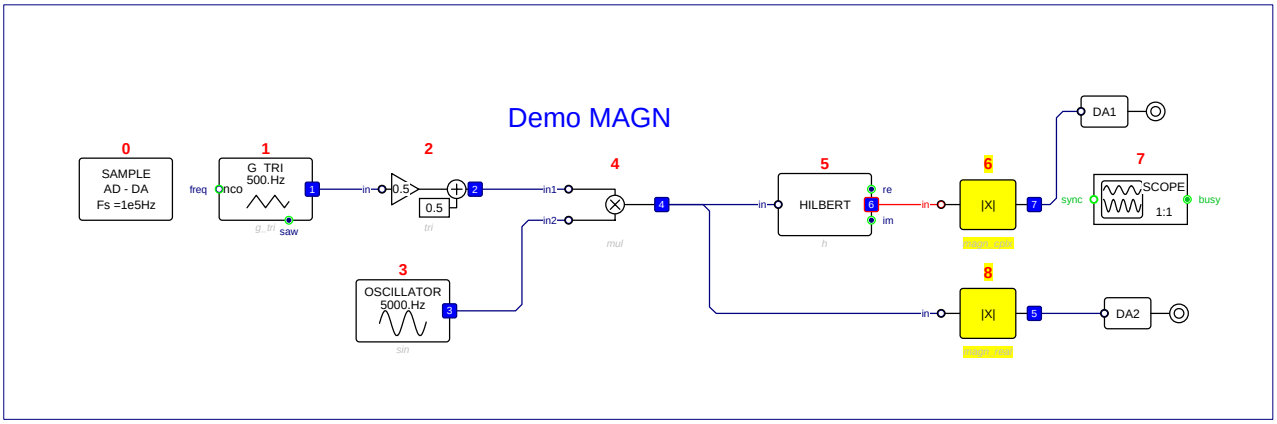
DESCRIPTION:
Magnitude of a real or complex input

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	defined by cn		mandatory

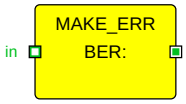
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



MAGN test program

MAKE_ERR Inject errors within vector MAKE_ERR



DESCRIPTION:
Inject errors within vector
with proba BER for each bit

PARAMETERS:

Parameter:	*Default values:*
BER:	0.0001

INPUTS

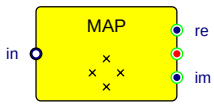
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	BOOL	Matrix of BIT	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	BOOL	Matrix of BIT	normal

Map symbol to complex

MAP



CATEGORY: Telecom

DESCRIPTION:
Map symbol to complex
The set of complex values forms a constellation

PARAMETERS:

Parameter:	*Default values:*
Bits per symbol	1
Constellation	map_ook,map_bpsk,map_ask4,map_ask8,map_psk4,map_psk8,map_qam8,map_qam16

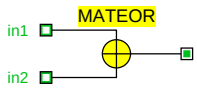
INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	COMPLEX	WORD	optional
name_re	FRACT	WORD	optional
name_im	FRACT	WORD	optional

XOR between matrixes



CATEGORY: Matrix

DESCRIPTION:
XOR between matrixes

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in1	BOOL	Matrix of BIT	mandatory
name_in2	BOOL	Matrix of BIT	mandatory

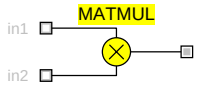
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	BOOL	Matrix of BIT	normal

MATMUL

Matrix product

MATMUL



CATEGORY: Matrix

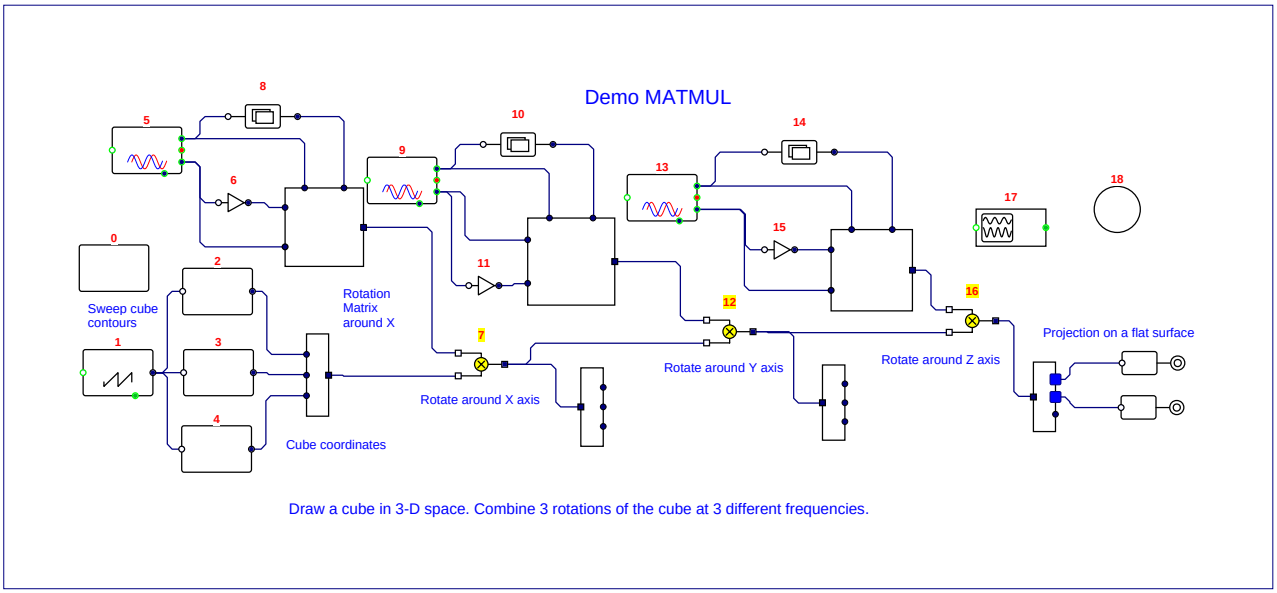
DESCRIPTION:
Matrix product

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in1	defined by cn	Matrix of	mandatory
name_in2	defined by cn	Matrix of	mandatory

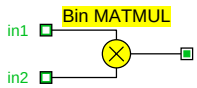
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	defined by cn	Matrix of	normal



MATMUL test program

MATMULB Boolean Matrix product in GF(2) MATMULB



CATEGORY: Matrix

DESCRIPTION:
Boolean Matrix product in GF(2)
M[i,k]=Sigma{M1[i,j]M2[j,k], j=0..cols(M1)} mod 2

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in1	BOOL	Matrix of BIT	mandatory
name_in2	BOOL	Matrix of BIT	mandatory

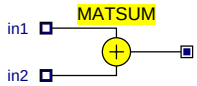
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	BOOL	Matrix of BIT	normal

MATSUM

Sum of matrixes

MATSUM



CATEGORY: Matrix

DESCRIPTION:
Sum of matrixes

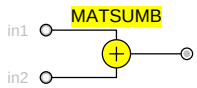
INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in1	FRACT	Matrix of WORD	mandatory
name_in2	FRACT	Matrix of WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	Matrix of WORD	normal

MATSUMB GF(2) sum of matrixes MATSUMB



CATEGORY: Matrix

DESCRIPTION:
GF(2) sum of matrixes
out(i,j) = in1(i,j) eor in2(i,j)

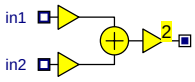
INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in1	BMAT	WORD	mandatory
name_in2	BMAT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	BMAT	WORD	normal

MATWSUM2 Weighted Sum of Matrixes MATWSUM2



CATEGORY: Matrix

DESCRIPTION:
Weighted Sum of Matrixes
C = 2^n *(k1*A + k2*B)

PARAMETERS:

Parameter:	*Default values:*
k1	0.5
k2	0.5
n	0

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in1	FRACT	Matrix of WORD	mandatory
name_in2	FRACT	Matrix of WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	Matrix of WORD	normal

MINISCOPE View signal at cursor position MINISCOPE

CATEGORY: Instruments

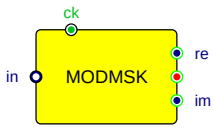
DESCRIPTION:
View signal at cursor position

PARAMETERS:

Parameter:	*Default values:*
points	100

ATTRIBUTES
Unique,

MODMSK Minimum Shift Keying modulator MODMSK



CATEGORY: Telecom

DESCRIPTION:
Minimum Shift Keying modulator

PARAMETERS:

Parameter:	*Default values:*
Bauds	1000.

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

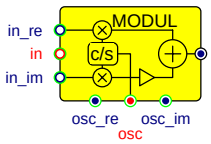
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	COMPLEX	WORD	optional
name_re	FRACT	WORD	optional
name_im	FRACT	WORD	optional
name_ck	BOOL	BIT	normal

MODUL

I-Q modulator

MODUL



CATEGORY: Telecom

DESCRIPTION:
I-Q modulator

PARAMETERS:

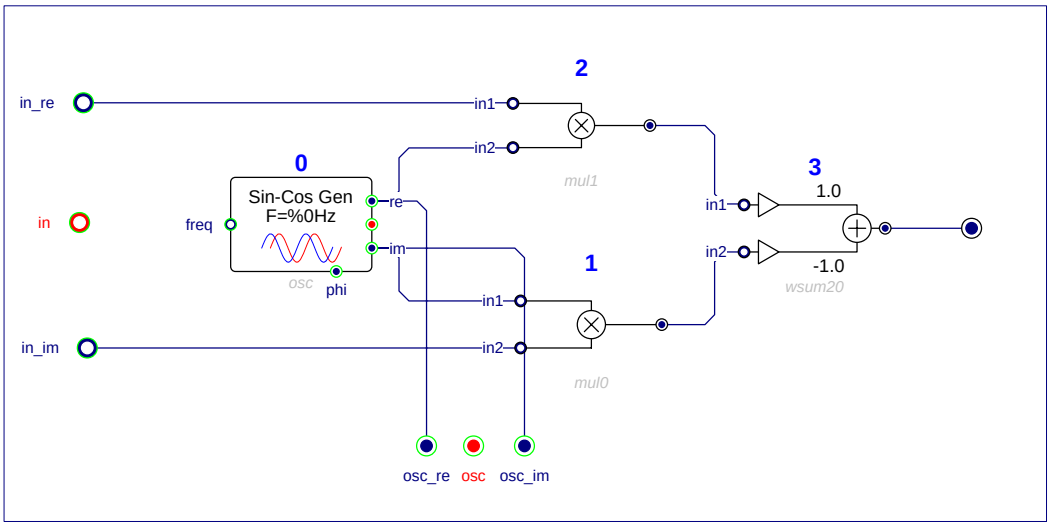
Parameter:	*Default values:*
Frequency	100.

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	COMPLEX	WORD	optional
name_in_re	FRACT	WORD	optional
name_in_im	FRACT	WORD	optional

OUTPUTS

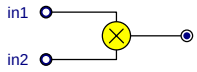
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal
name_osc	COMPLEX	WORD	optional
name_osc_re	FRACT	WORD	optional
name_osc_im	FRACT	WORD	optional



MODUL internal schema

Real multiplier

MUL



CATEGORY: Arithmetic

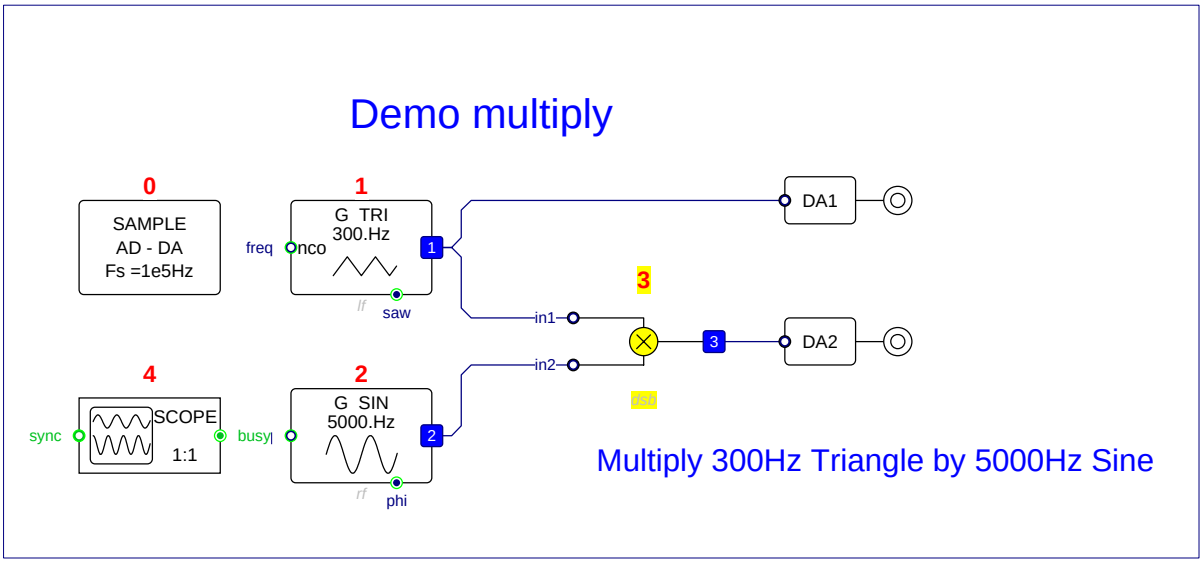
DESCRIPTION:
Real multiplier

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in1	FRACT	WORD	mandatory
name_in2	FRACT	WORD	mandatory

OUTPUTS

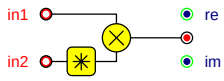
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



MUL test program

Multiply with conjugate

MULCC



CATEGORY: Arithmetic

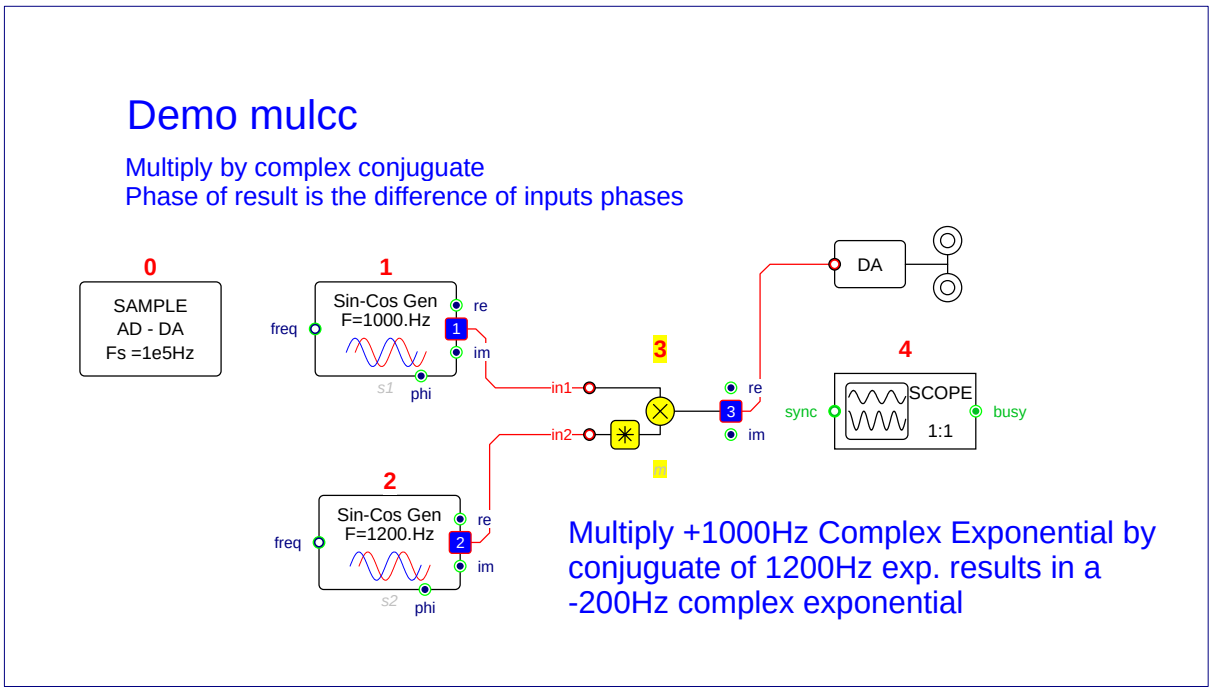
DESCRIPTION:
Multiply with conjugate
Complex product of in1 by conjuguate of in2

INPUTS

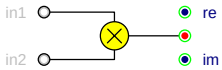
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in1	COMPLEX	WORD	mandatory
name_in2	COMPLEX	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	COMPLEX	WORD	normal
name_re	FRACT	WORD	optional
name_im	FRACT	WORD	optional



MULCC test program



CATEGORY: Arithmetic

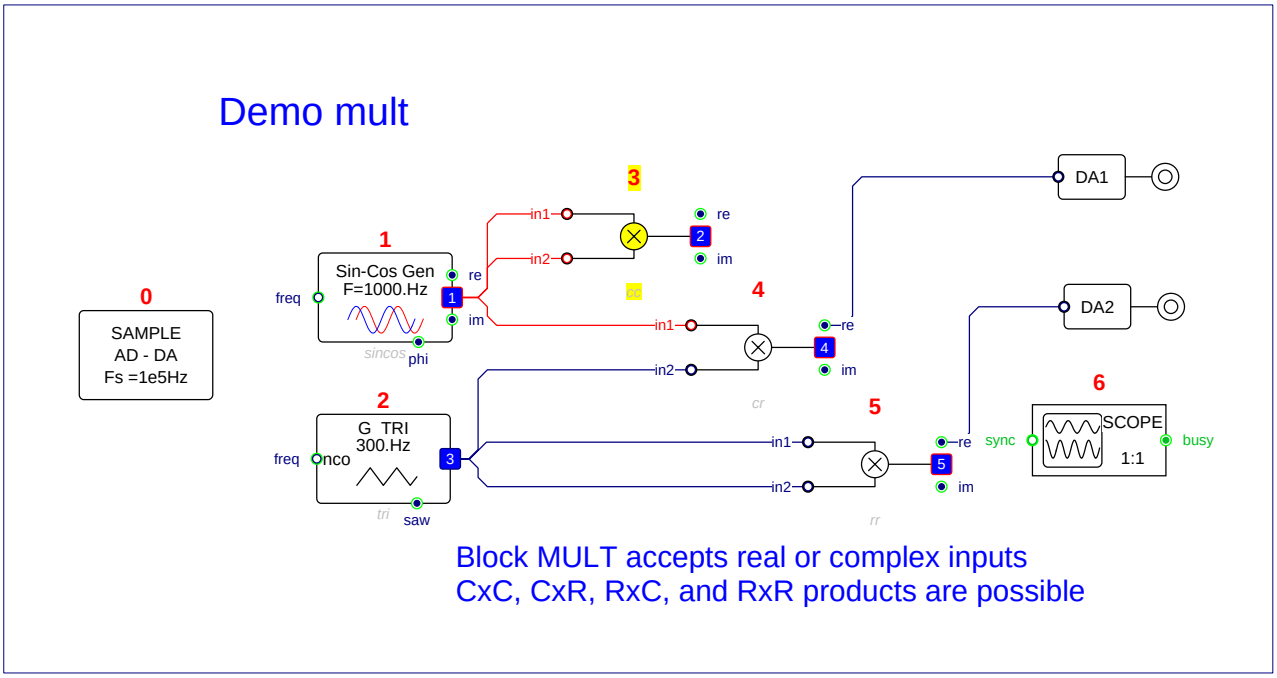
DESCRIPTION:
Complex, mixed, or real multiplier

INPUTS

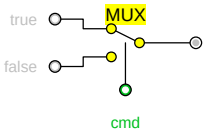
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in1	defined by cn		mandatory
name_in2	defined by cn		mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	COMPLEX	WORD	optional
name_re	FRACT	WORD	optional
name_im	FRACT	WORD	optional



MULT test program



CATEGORY: Control

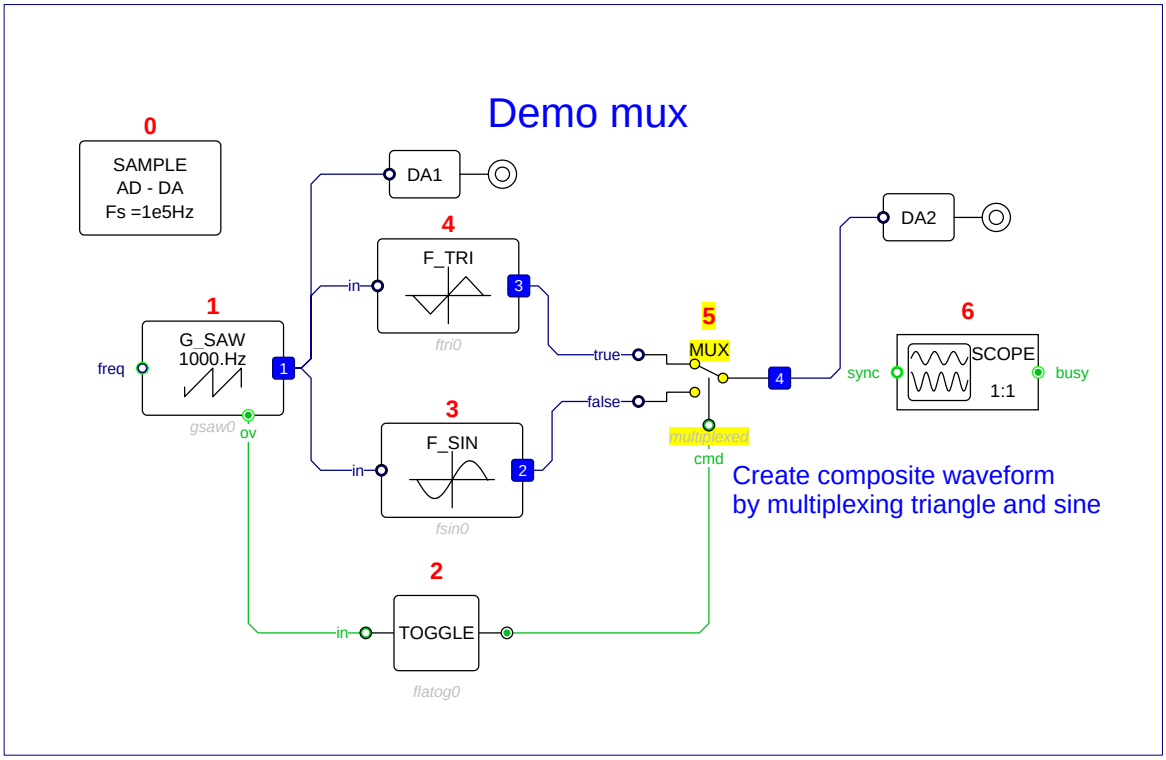
DESCRIPTION:
2 input multiplexer

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_true	defined by cn		mandatory
name_false	defined by cn		mandatory
name_cmd	BOOL	BIT	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	defined by cn		normal

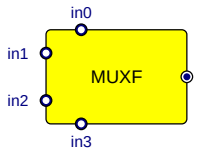


MUX test program

MUXF

f-domain multiplexer

MUXF



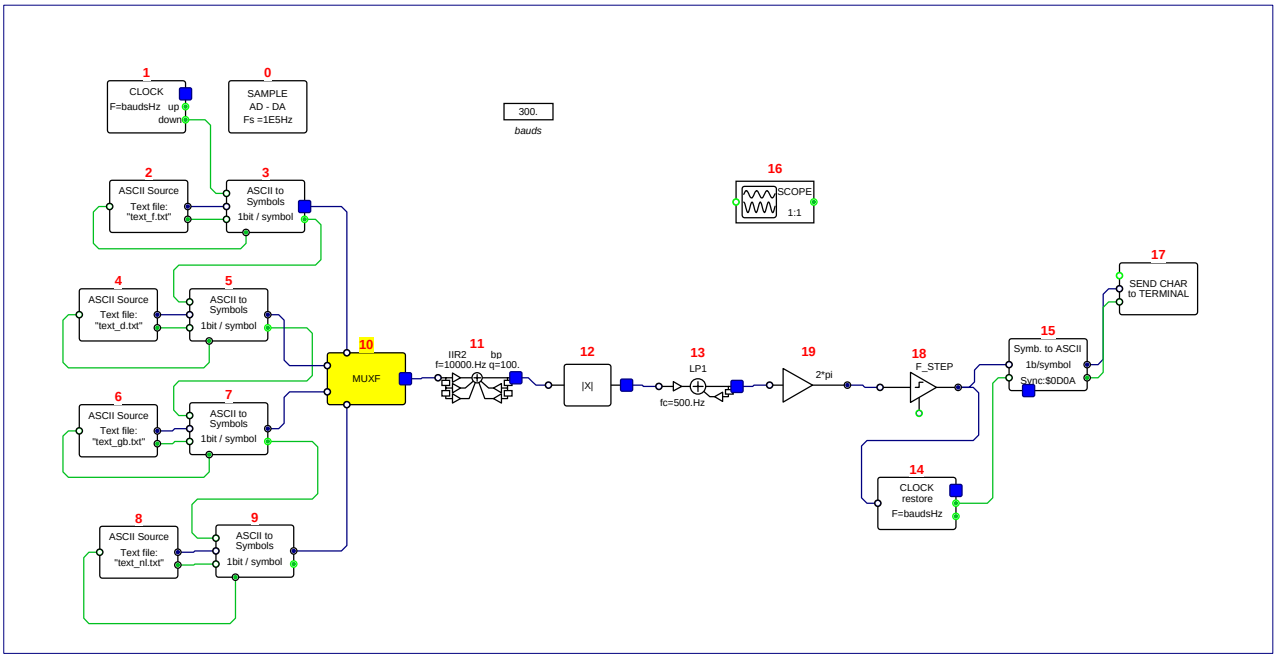
DESCRIPTION:
f-domain multiplexer

INPUTS

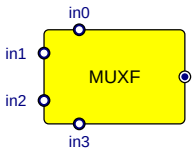
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in1	FRACT	WORD	mandatory
name_in0	FRACT	WORD	mandatory
name_in2	FRACT	WORD	mandatory
name_in3	FRACT	WORD	mandatory

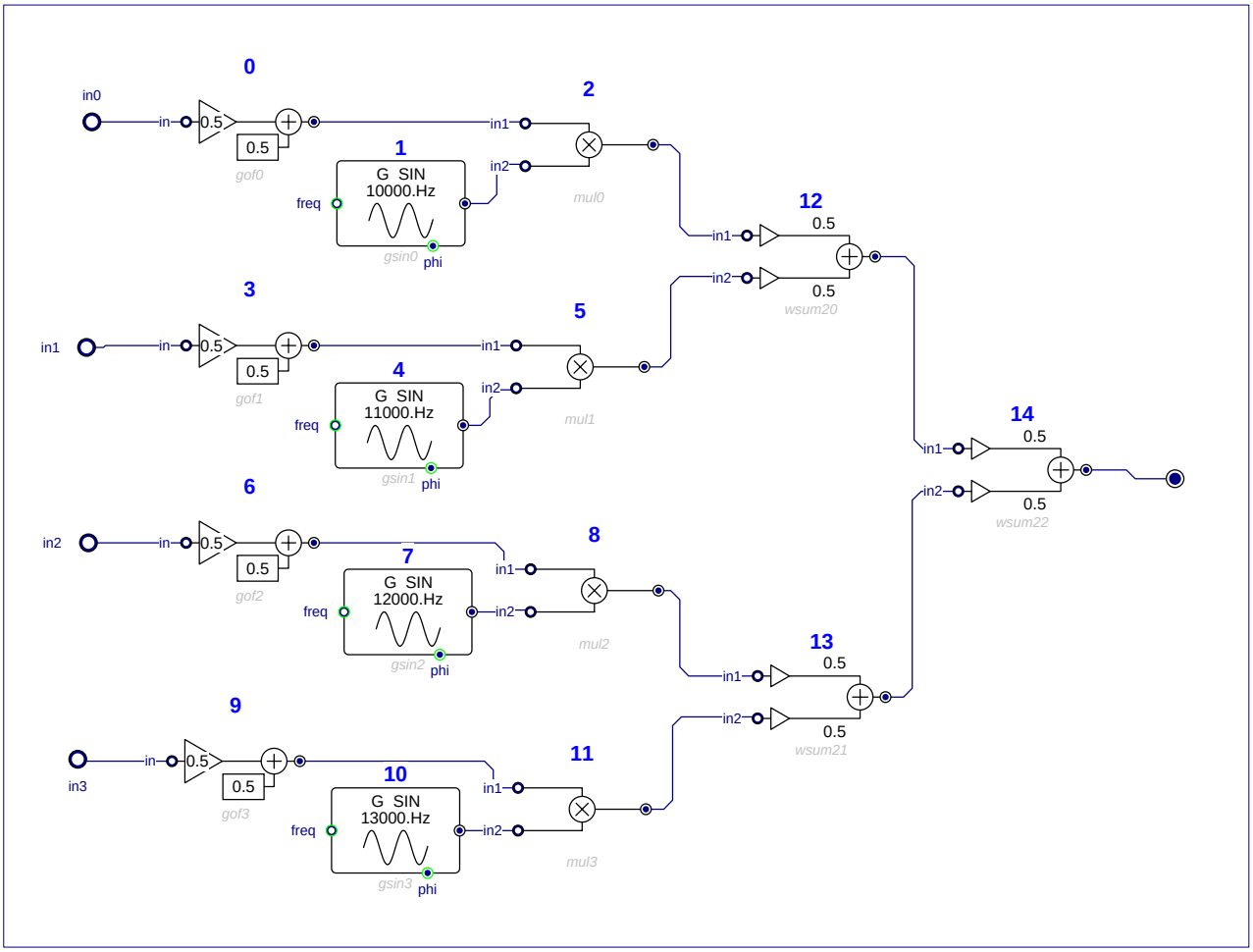
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



MUXF test program



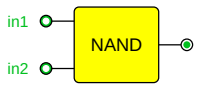


MUXF internal schema

NANDGATE

Logic NAND

NANDGATE



CATEGORY: Logic

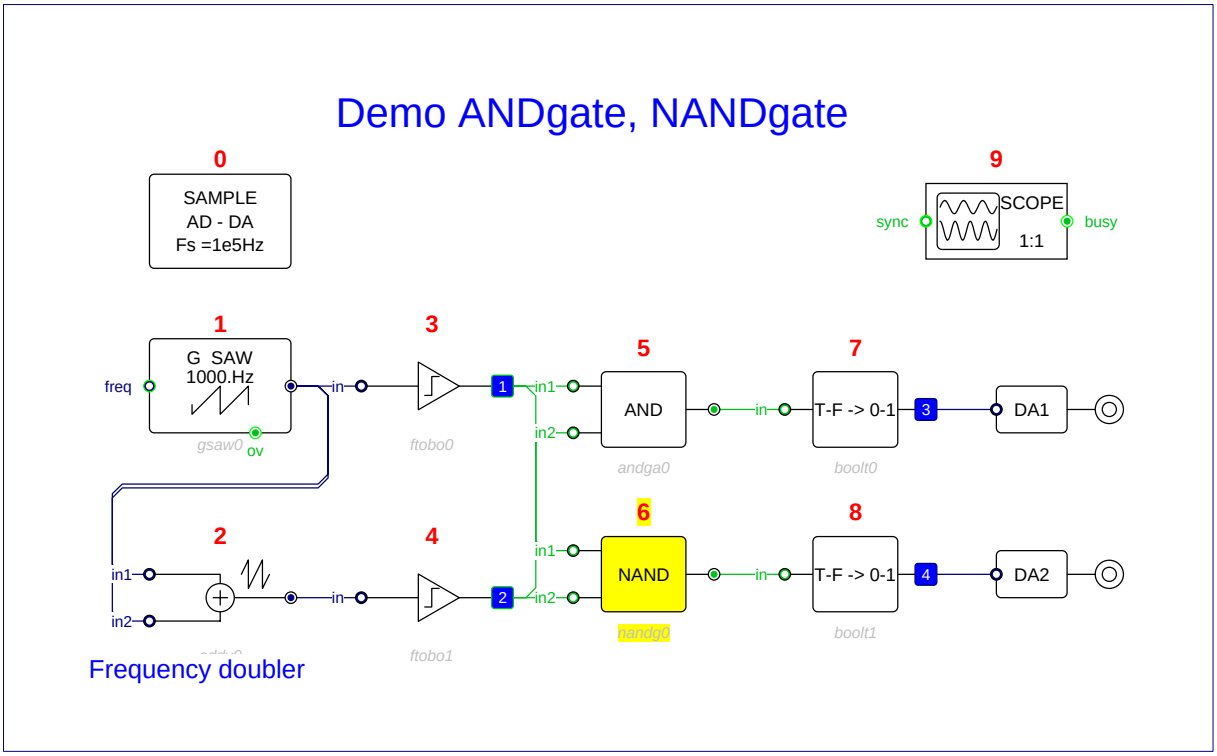
DESCRIPTION:
Logic NAND
y = (in1 & in2)\

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in1	BOOL	BIT	mandatory
name_in2	BOOL	BIT	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	BOOL	BIT	normal

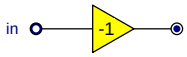


NANDGATE test program

NEGATE

Sign inversion y = -x

NEGATE



CATEGORY: Arithmetic

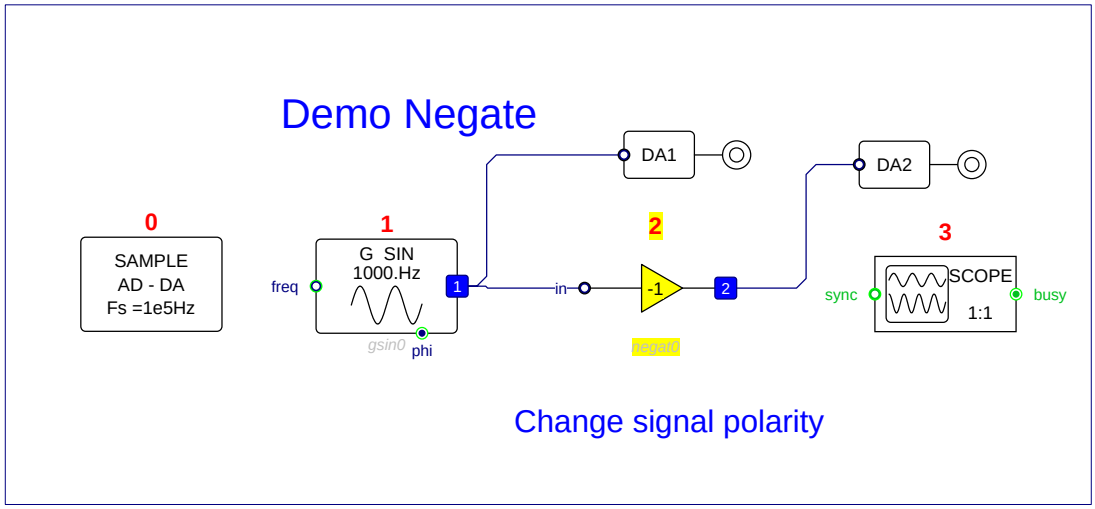
DESCRIPTION:
Sign inversion y = -x

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



NEGATE test program

NOP

No operation

NOP

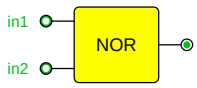
CATEGORY: Control

DESCRIPTION:
No operation

NORGATE

Logic NOR function

NORGATE



CATEGORY: Logic

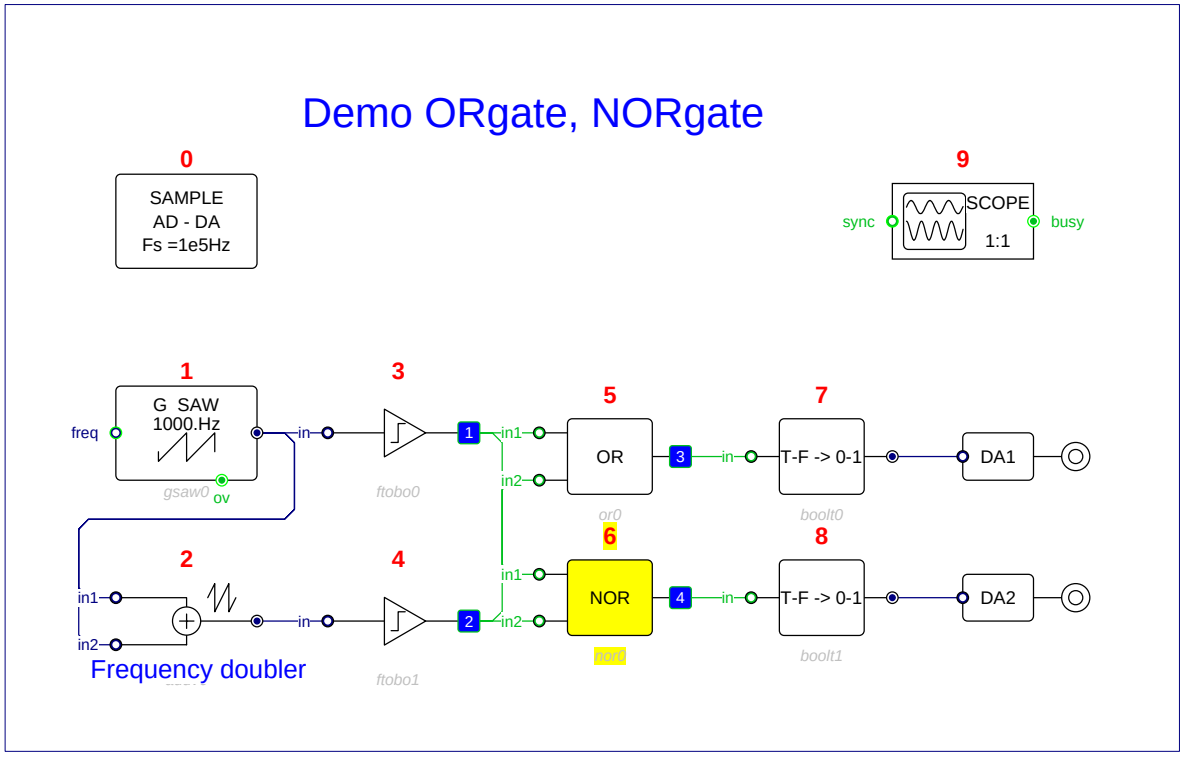
DESCRIPTION:
Logic NOR function
y = (in1 v in2)\

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in1	BOOL	BIT	mandatory
name_in2	BOOL	BIT	mandatory

OUTPUTS

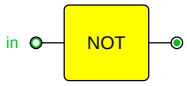
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	BOOL	BIT	normal



NORGATE test program

NOTGATE

NOTGATE



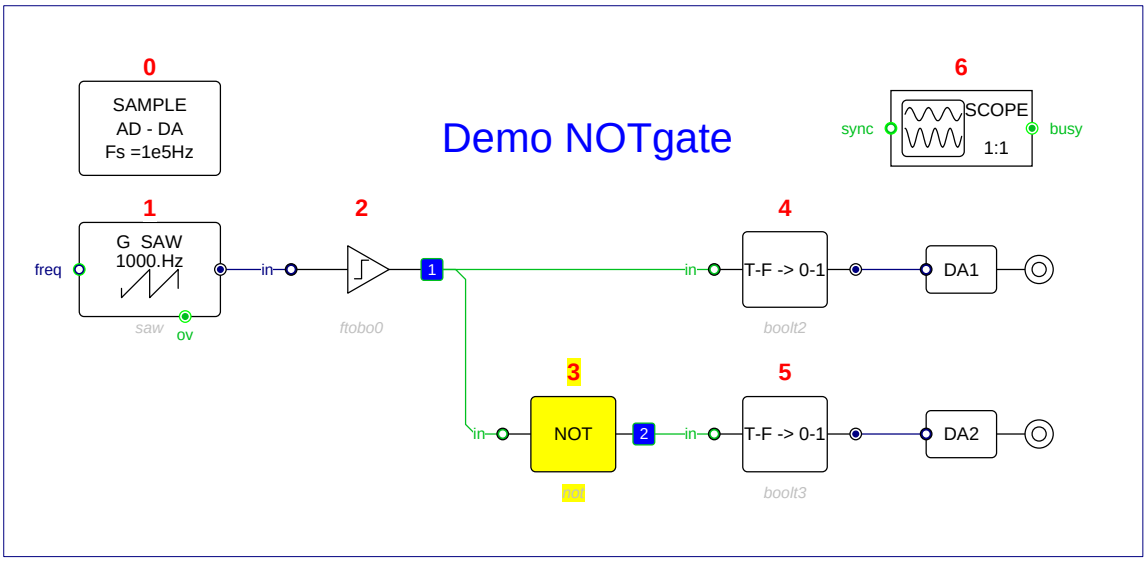
CATEGORY: Logic

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	BOOL	BIT	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	BOOL	BIT	normal

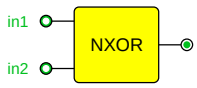


NOTGATE test program

NXORGATE

Logic NXOR function

NXORGATE



CATEGORY: Logic

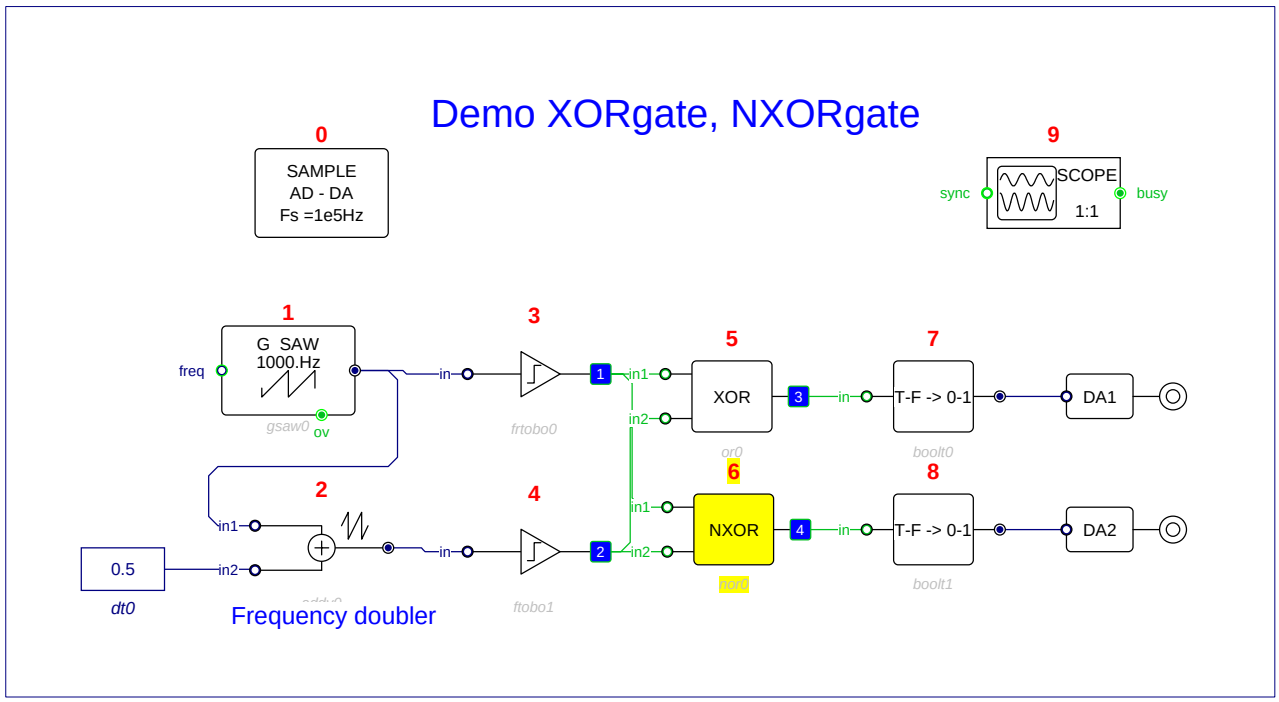
DESCRIPTION:
Logic NXOR function
y = (in1\ v in2) & (in1 v in2\)

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in1	BOOL	BIT	mandatory
name_in2	BOOL	BIT	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	BOOL	BIT	normal

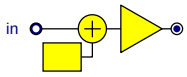


NXORGATE test program

OFFGAIN

Offset and gain:

OFFGAIN



CATEGORY: Arithmetic

DESCRIPTION:
Offset and gain:
y = gain*(in + offset)

PARAMETERS:

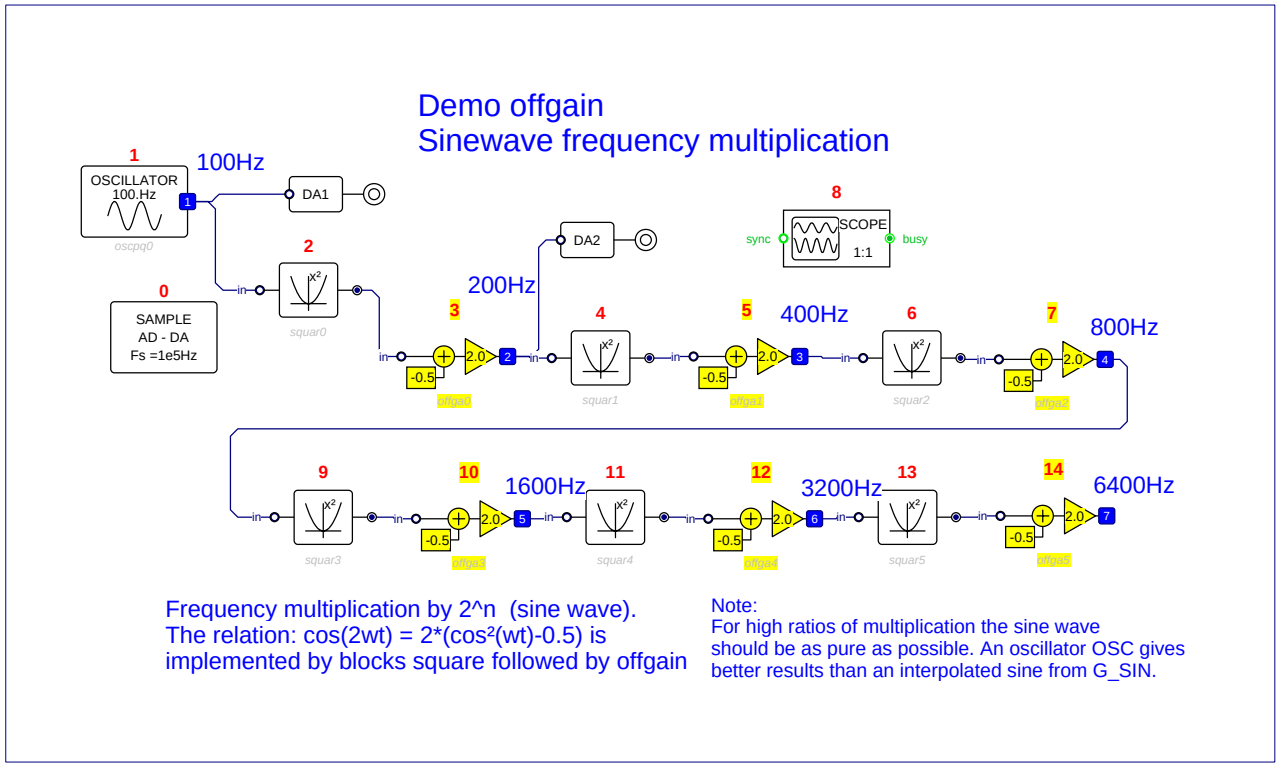
Parameter:	*Default values:*
offset	0.5
gain	2.0

INPUTS

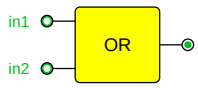
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



OFFGAIN test program



CATEGORY: Logic

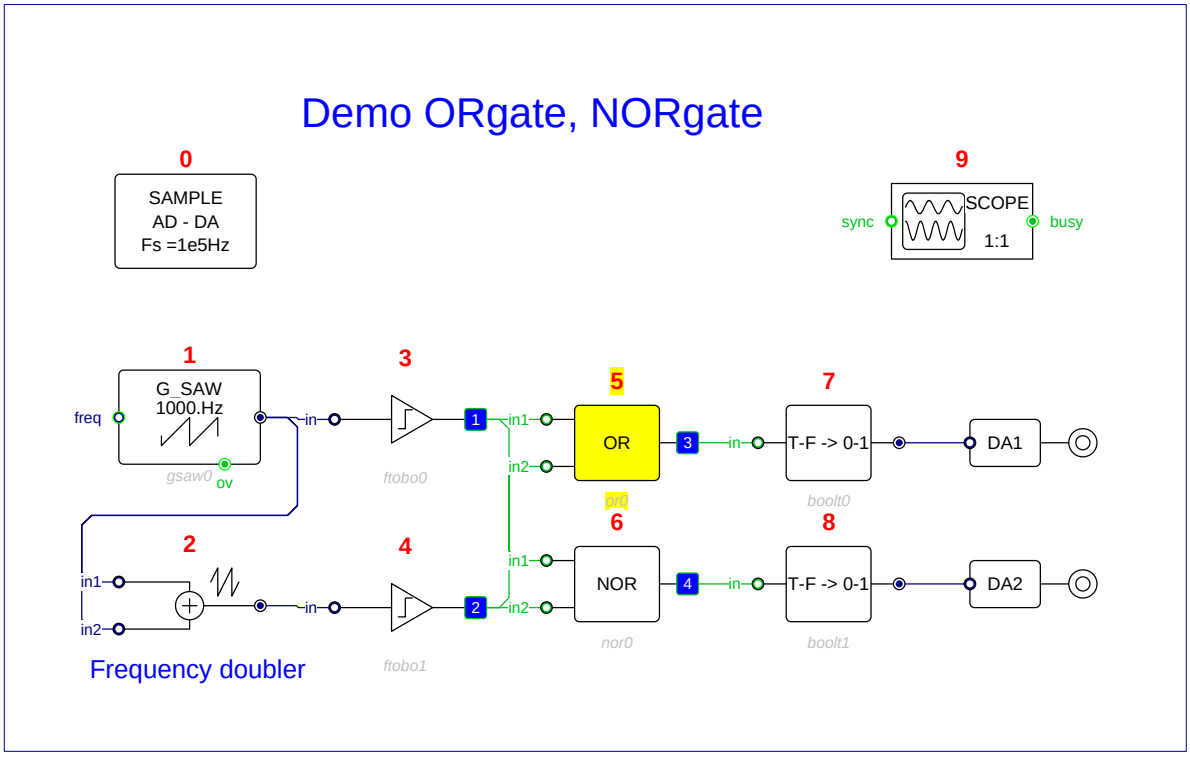
DESCRIPTION:
Logic OR function
y = in1 v in2

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in1	BOOL	BIT	mandatory
name_in2	BOOL	BIT	mandatory

OUTPUTS

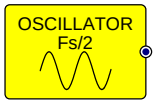
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	BOOL	BIT	normal



ORGATE test program

High purity sine oscillator

OSC



CATEGORY: Generators

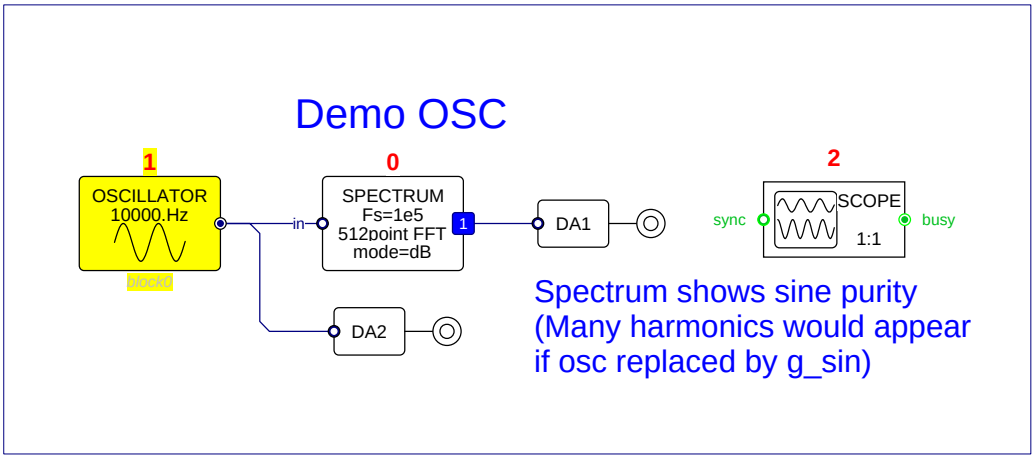
DESCRIPTION:
High purity sine oscillator

PARAMETERS:

Parameter:	*Default values:*
Frequency	1000.
Unit	Hz,Fs/2

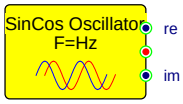
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



OSC test program

Sinusoidal phase quadrature oscillator



CATEGORY: Generators

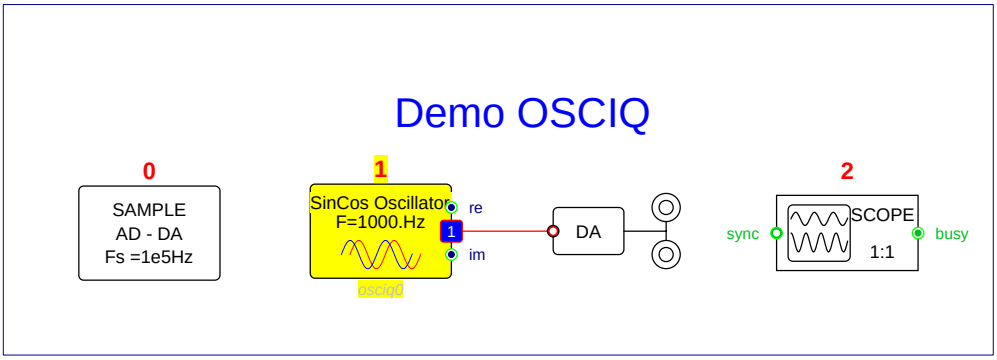
DESCRIPTION:
Sinusoidal phase quadrature oscillator

PARAMETERS:

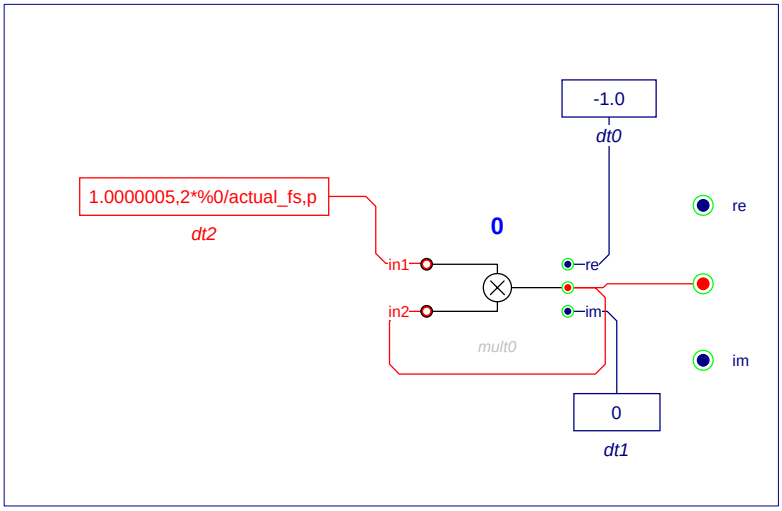
Parameter:	*Default values:*
Frequency	1000.

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	COMPLEX	WORD	optional
name_re	FRACT	WORD	optional
name_im	FRACT	WORD	optional



OSCIQ test program

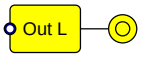


OSCIQ internal schema

OUT_L

Codec output Left

OUT_L



CATEGORY: Audio

DESCRIPTION:
Codec output Left
Digital to Analog Converter input

INPUTS

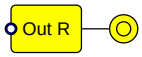
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	mandatory

ATTRIBUTES
Non executable, Unique,

OUT_R

Codec output Right

OUT_R



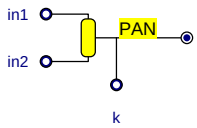
CATEGORY: Audio

DESCRIPTION:
Codec output Right
Digital to Analog Converter input

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	mandatory

ATTRIBUTES
Non executable, Unique,



CATEGORY: Audio

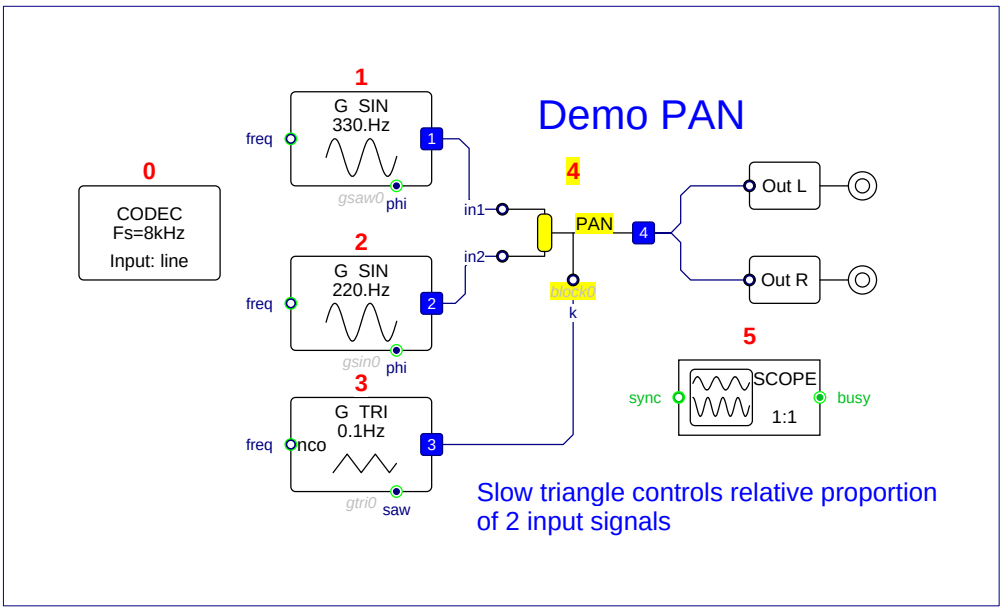
DESCRIPTION:
Panoramic
y = (x1+x2)/2 + k(x1-x2)/2
y=in1 if k=1 y=(in1+in2)/2 if k=0 y=in2 if k=-1

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in1	FRACT	WORD	mandatory
name_in2	FRACT	WORD	mandatory
name_k	FRACT	WORD	mandatory

OUTPUTS

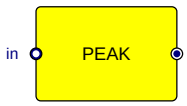
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



PAN test program

Get peak value of input

PEAK



CATEGORY: Control

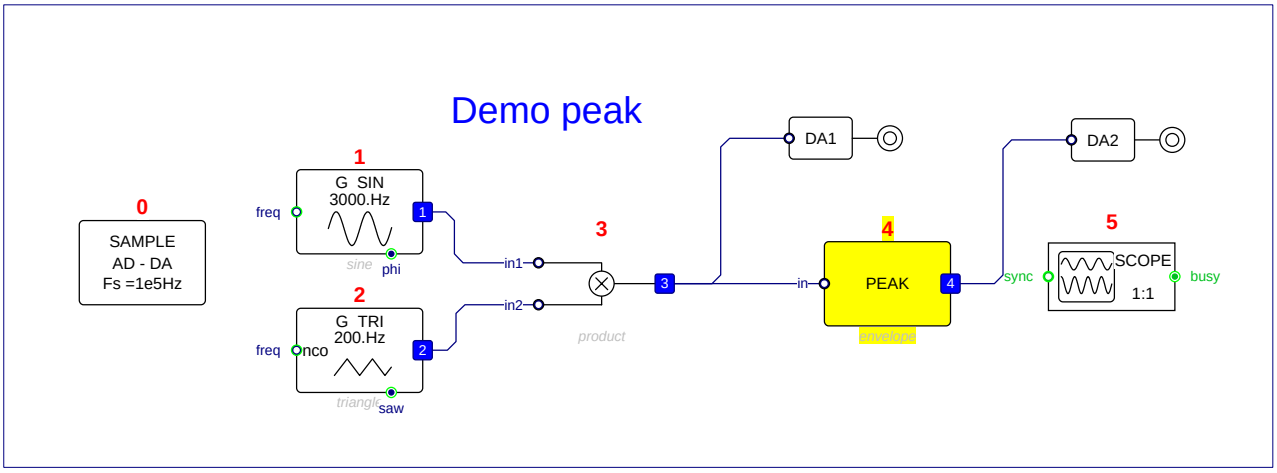
DESCRIPTION:
Get peak value of input

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

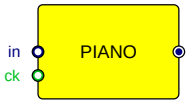
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



PEAK test program

PIANO

PIANO



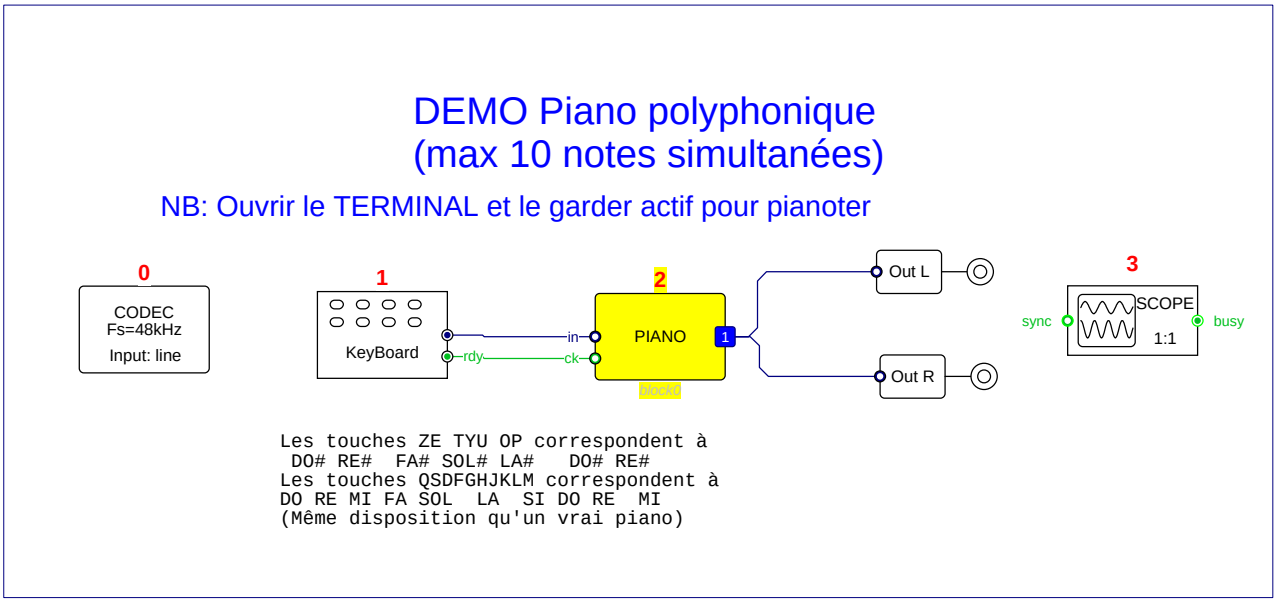
CATEGORY: Audio

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory
name_ck	BOOL	BIT	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

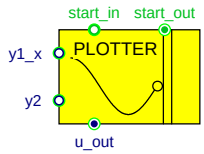


PIANO test program

PLOTTER

Slow signal plotter

PLOTTER



CATEGORY: Instruments

DESCRIPTION:
Slow signal plotter

PARAMETERS:

Parameter:	*Default values:*
Mode	T_Y1_Y2,X_Y
Scan Time	5s,10s,20s,1mn,2mn,5mn,10mn,30mn,1h,2h,6h,12h,24h
X_Y1 max	1.0
X_Y1 min	-1.0
X_Y1 unit	_
Y2 max	1.0
Y2 min	-1.0
Y2 unit	_
Uout_Mode	Fractional,Frequency_Linear,Frequency_Geometric
Uout_max	1.0
Uout_min	-1.0
Uout_Unit	_

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_y1_x	FRACT	WORD	optional
name_y2	FRACT	WORD	optional
name_start_in	BOOL	BIT	optional

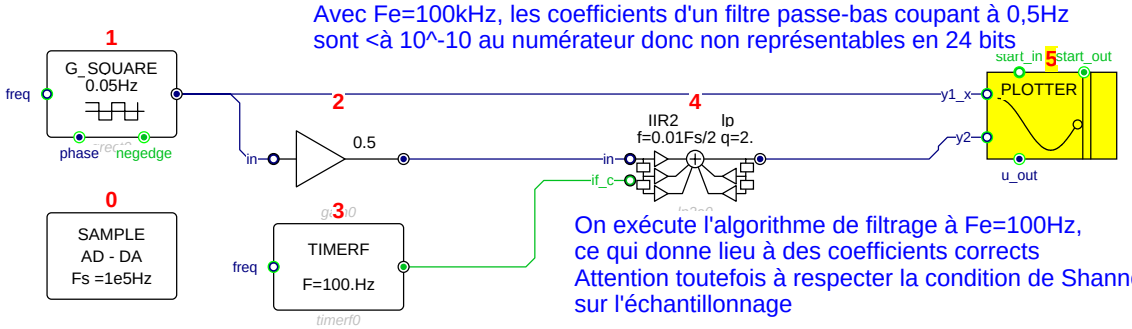
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_u_out	FRACT	WORD	optional
name_start_out	BOOL	BIT	optional

ATTRIBUTES
Unique,

DEMO1 PLOTTER
Mise en oeuvre d'un Second Ordre
à grande constante de temps,
enregistrement de la réponse sur table tarçante

Avec Fe=100kHz, les coefficients d'un filtre passe-bas coupant à 0,5Hz
sont <à 10^-10 au numérateur donc non représentables en 24 bits

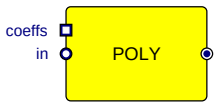


On exécute l'algorithme de filtrage à Fe=100Hz,
ce qui donne lieu à des coefficients corrects
Attention toutefois à respecter la condition de Shannon
sur l'échantillonnage

PLOTTER test program

Real Polynomial function

POLY



CATEGORY: Functions

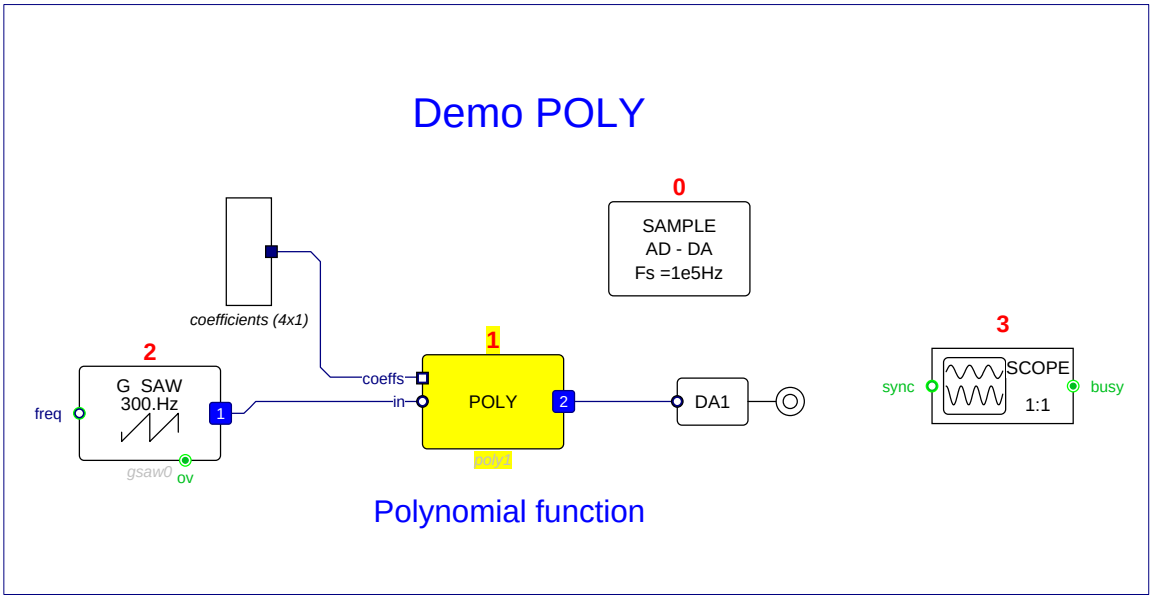
DESCRIPTION:
Real Polynomial function
Table contains coefficients in order 1, x, x^2, ...

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_coeffs	FRACT	Matrix of WORD	mandatory
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

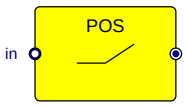


POLY test program

POS

Diode function: if x > 0 then y = x else y = 0

POS



CATEGORY: Non linear

DESCRIPTION:
Diode function: if x > 0 then y = x else y = 0

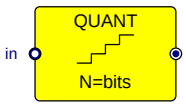
INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

Quantize data to n bits



CATEGORY: Non linear

DESCRIPTION:
Quantize data to n bits

PARAMETERS:

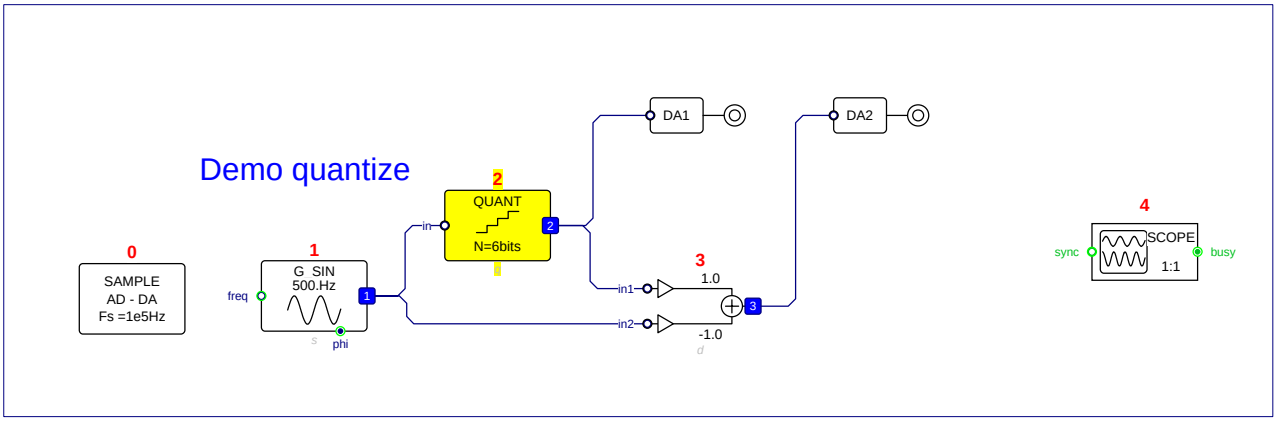
Parameter:	*Default values:*
bits	3
Approx: rnd/exss/deflt	r,e,d

INPUTS

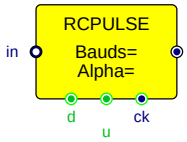
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



QUANT test program



CATEGORY: Telecom

DESCRIPTION:
Raised Cosine Pulse shaper
Alpha=0 -> rect spectrum -> Sinc pulse
Alpha=1 -> Hann spectrum -> shortest pulse
Length= pulse duration (symbol periods)

PARAMETERS:

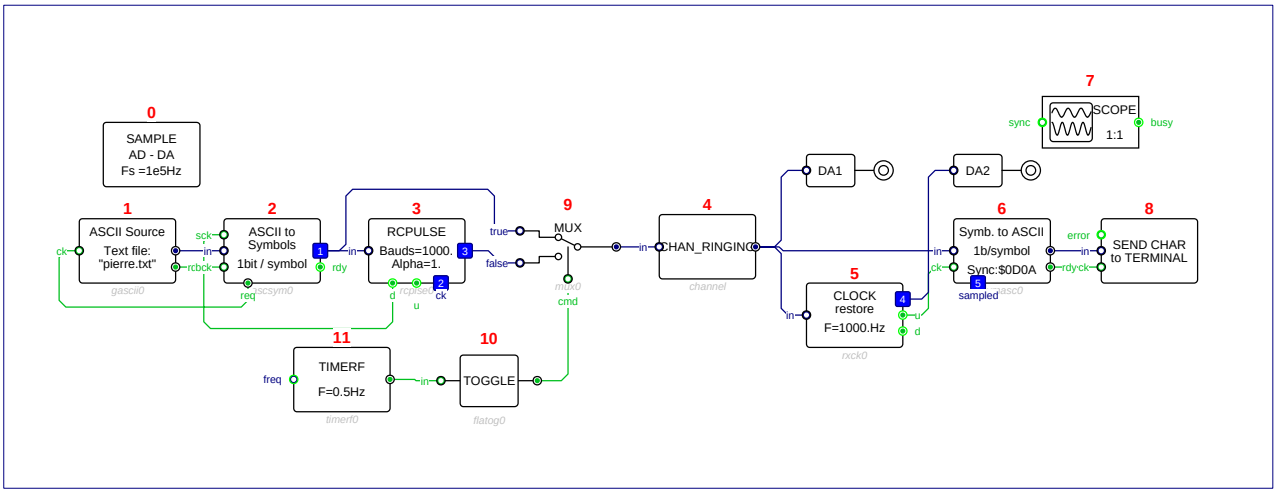
Parameter:	*Default values:*
Bauds:	1000.
Alpha:	0.5
Length:	2

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_ck	FRACT	WORD	optional
name_d	BOOL	BIT	optional
name_u	BOOL	BIT	optional
name	FRACT	WORD	normal

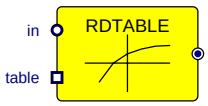


RCPULSE test program

RDTABLE

Read interpolate table

RDTABLE



CATEGORY: Functions

DESCRIPTION:
Read interpolate table
Table is defined by matrix connection
If table columns>1 then more outputs can be added
Output name is the Nr of corresponding culumn (0, 1, 2 ..)

PARAMETERS:

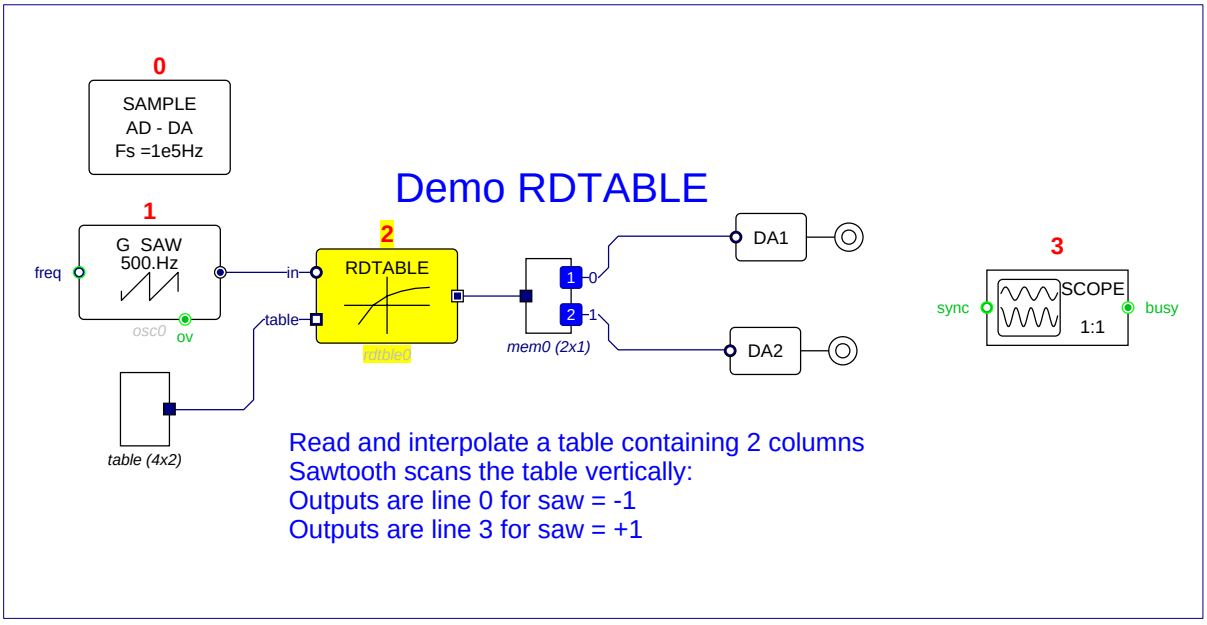
Parameter:	*Default values:*
Signed or unsigned	s,u

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory
name_table	FRACT	Matrix of WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



RDTABLE test program

Add reverberation to sound

REVERB

CATEGORY: Audio

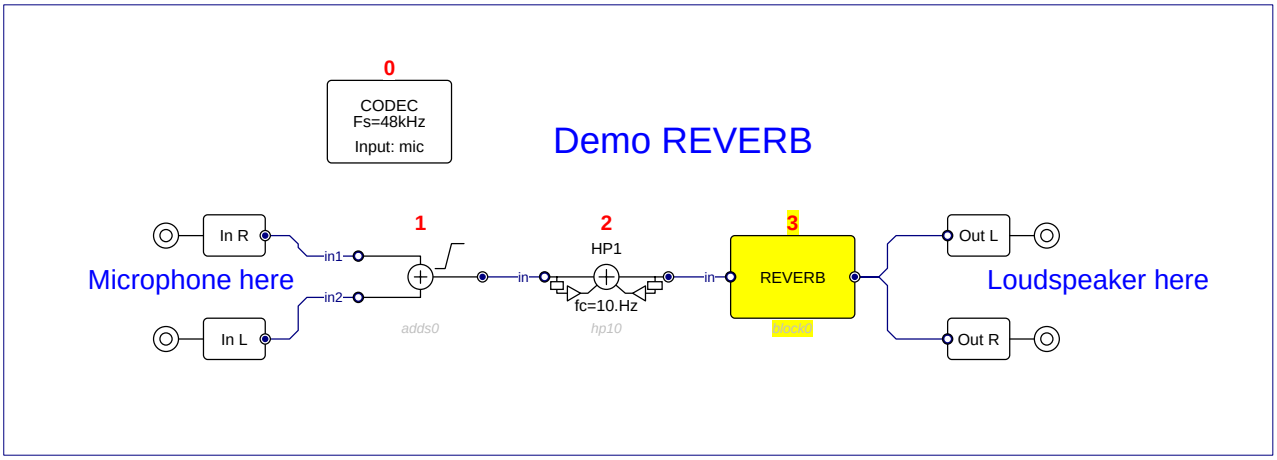
DESCRIPTION:
Add reverberation to sound

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

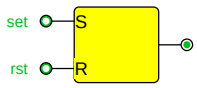
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



REVERB test program

RS_FLIPFLOP RS flip flop RS_FLIPFLOP



CATEGORY: Logic

DESCRIPTION:
RS flip flop
Applying TRUE to SET input results in output TRUE
Applying TRUE to RESET input results in output FALSE
If parameter "Dominant state" is "SET" then applying TRUE to both inputs results in TRUE
Otherwise applying TRUE to both inputs results in FALSE

PARAMETERS:

Parameter:	*Default values:*
Dominant state:	SET,RESET

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_set	BOOL	BIT	mandatory
name_rst	BOOL	BIT	mandatory

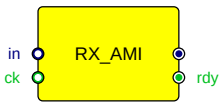
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	BOOL	BIT	normal

RX_AMI

Alternate Mark Inversion line decoder

RX_AMI



CATEGORY: Telecom

DESCRIPTION:
Alternate Mark Inversion line decoder

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory
name_ck	BOOL	BIT	mandatory

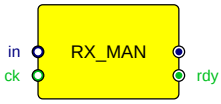
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal
name_rdy	BOOL	BIT	normal

RX_MAN

Manchester line decoder

RX_MAN



CATEGORY: Telecom

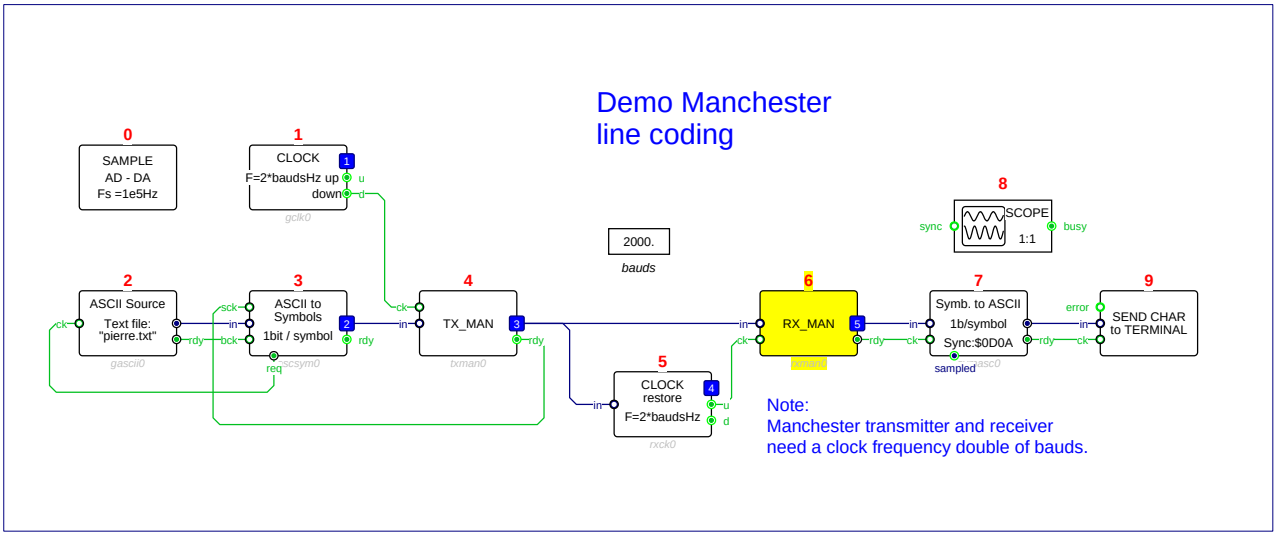
DESCRIPTION:
Manchester line decoder
Input clock is 2 x Bauds

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory
name_ck	BOOL	BIT	mandatory

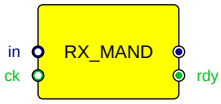
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal
name_rdy	BOOL	BIT	normal



RX_MAN test program

RX_MAND Differential Manchester line decoder RX_MAND



CATEGORY: Telecom

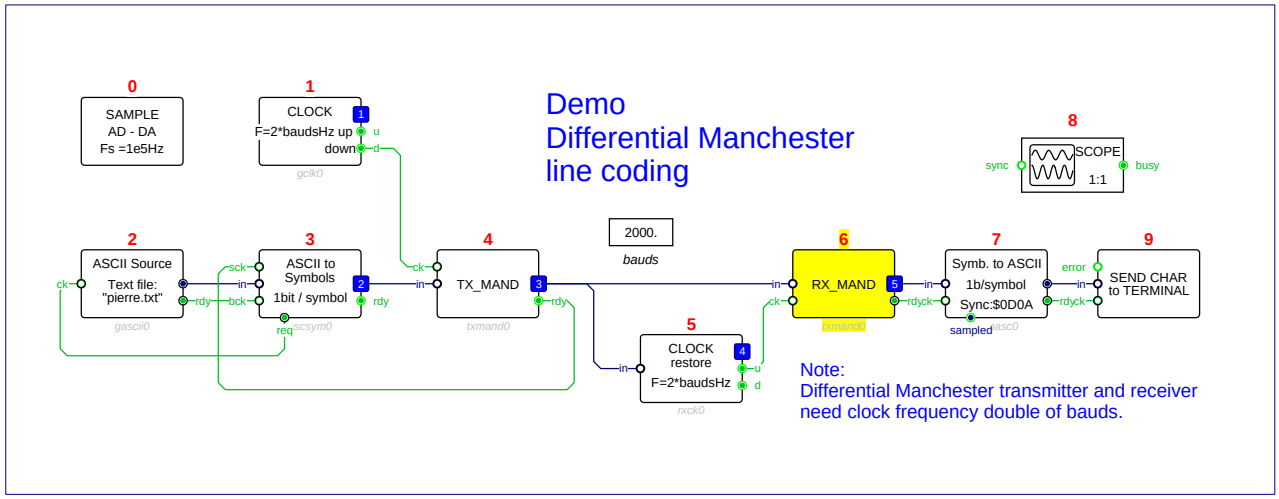
DESCRIPTION:
Differential Manchester line decoder
Input clock is 2 x Bauds

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory
name_ck	BOOL	BIT	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal
name_rdy	BOOL	BIT	normal



RX_MAND test program

CATEGORY: Telecom

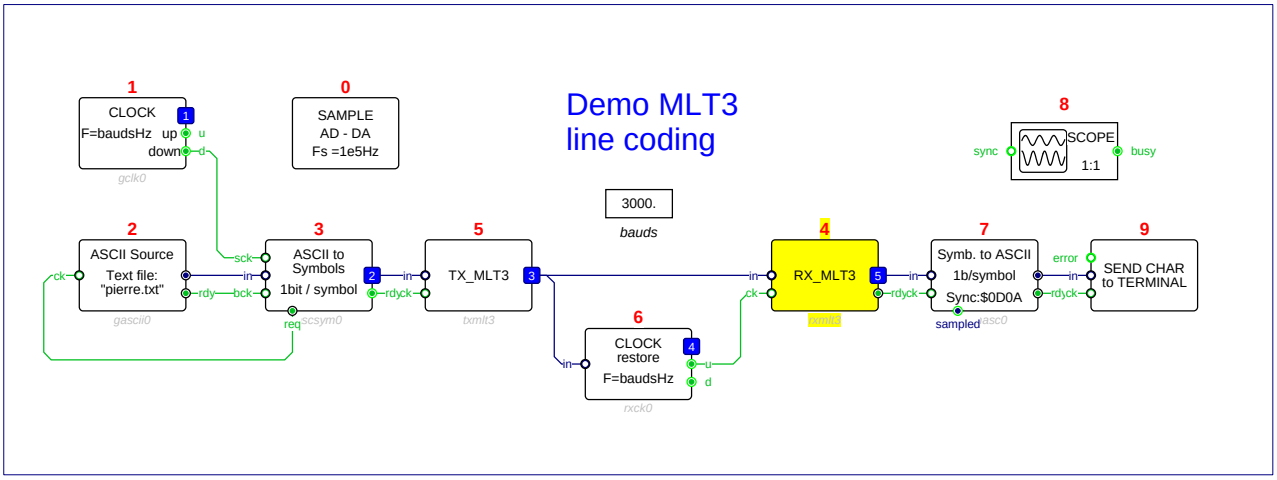
DESCRIPTION:
MLT3 line decoder

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory
name_ck	BOOL	BIT	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal
name_rdy	BOOL	BIT	normal



RX_MLT3 test program

CATEGORY: Telecom

DESCRIPTION:
Non Return to Zero line decoder

PARAMETERS:

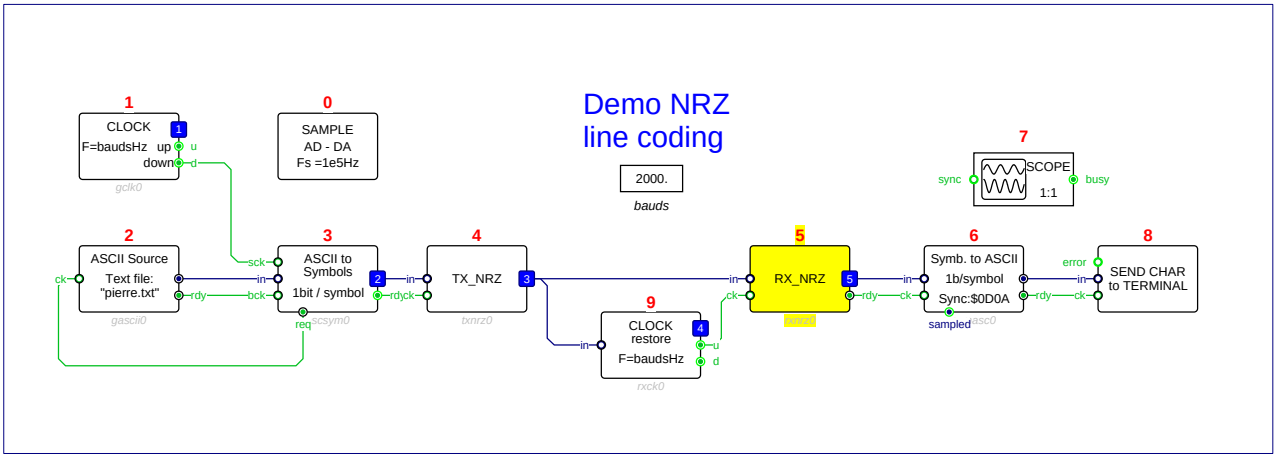
Parameter:	*Default values:*
Level Space	-1.0
Level Mark	1.0

INPUTS

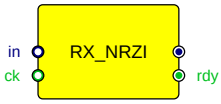
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory
name_ck	BOOL	BIT	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal
name_rdy	BOOL	BIT	normal



RX_NRZ test program



CATEGORY: Telecom

DESCRIPTION:
NRZI line decoder

PARAMETERS:

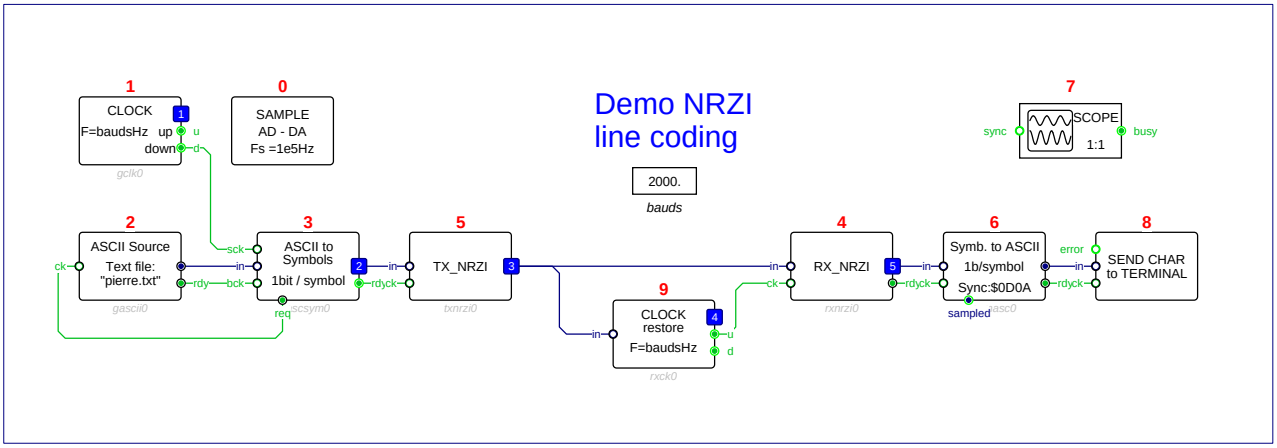
Parameter:	*Default values:*
Level Space	-1.0
Level Mark	1.0

INPUTS

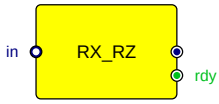
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory
name_ck	BOOL	BIT	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal
name_rdy	BOOL	BIT	normal



RX_NRZI test program



CATEGORY: Telecom

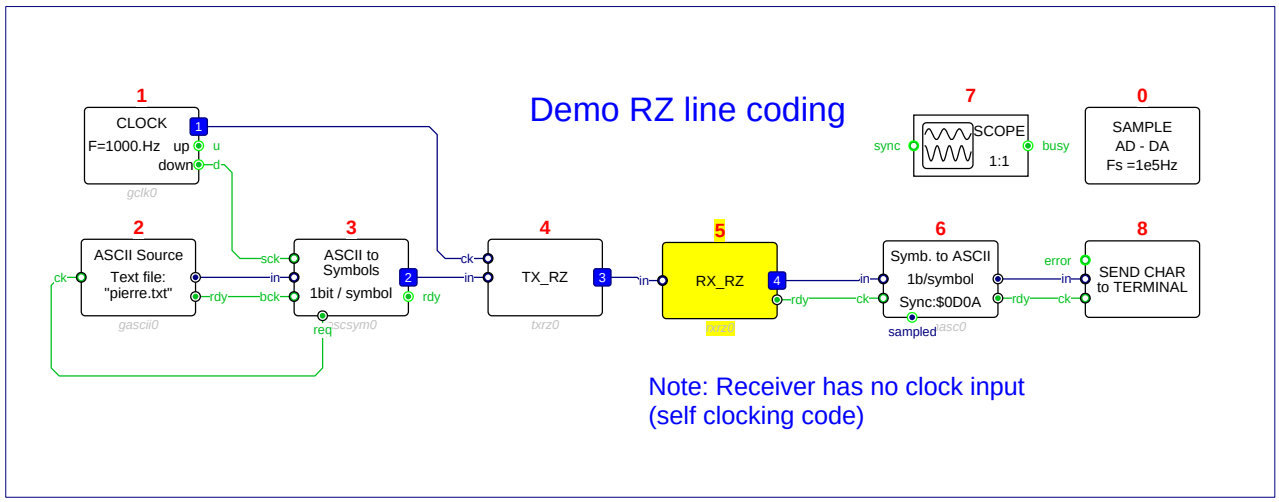
DESCRIPTION:
Return to Zero line decoder
(Self clocking)

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal
name_rdy	BOOL	BIT	normal



RX_RZ test program

Clock restoration

RXCK

CATEGORY: Telecom

DESCRIPTION:
Clock restoration
Retrives optimal clock phase at receiver end

PARAMETERS:

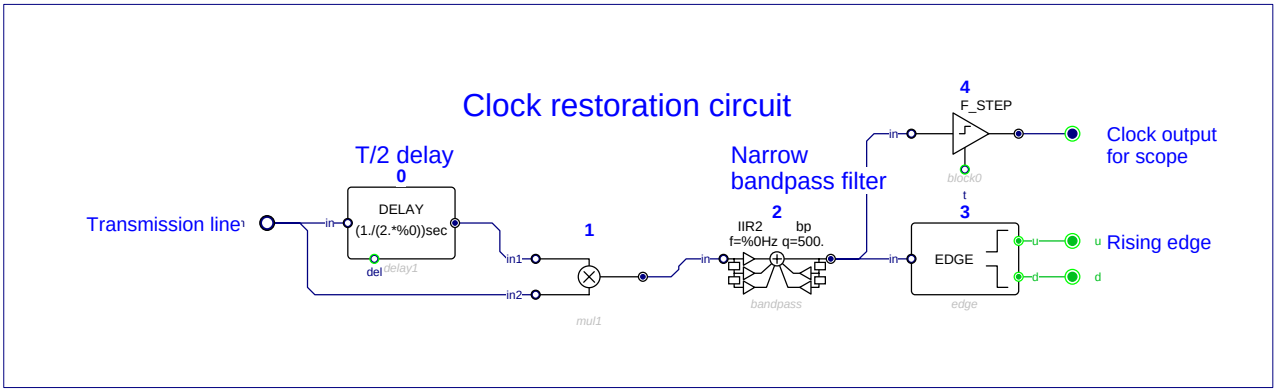
Parameter:	*Default values:*
Frequency	100.

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_d	BOOL	BIT	optional
name	FRACT	WORD	optional
name_u	BOOL	BIT	optional

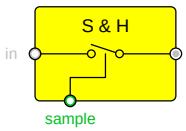


RXCK internal schema

SAMPHOLD

Sample and Hold

SAMPHOLD



CATEGORY: Control

DESCRIPTION:
Sample and Hold
Input may be real or complex
pos_edge, neg_edge = sample on L to H or H to L;
true = Sample on H, then set to L

PARAMETERS:

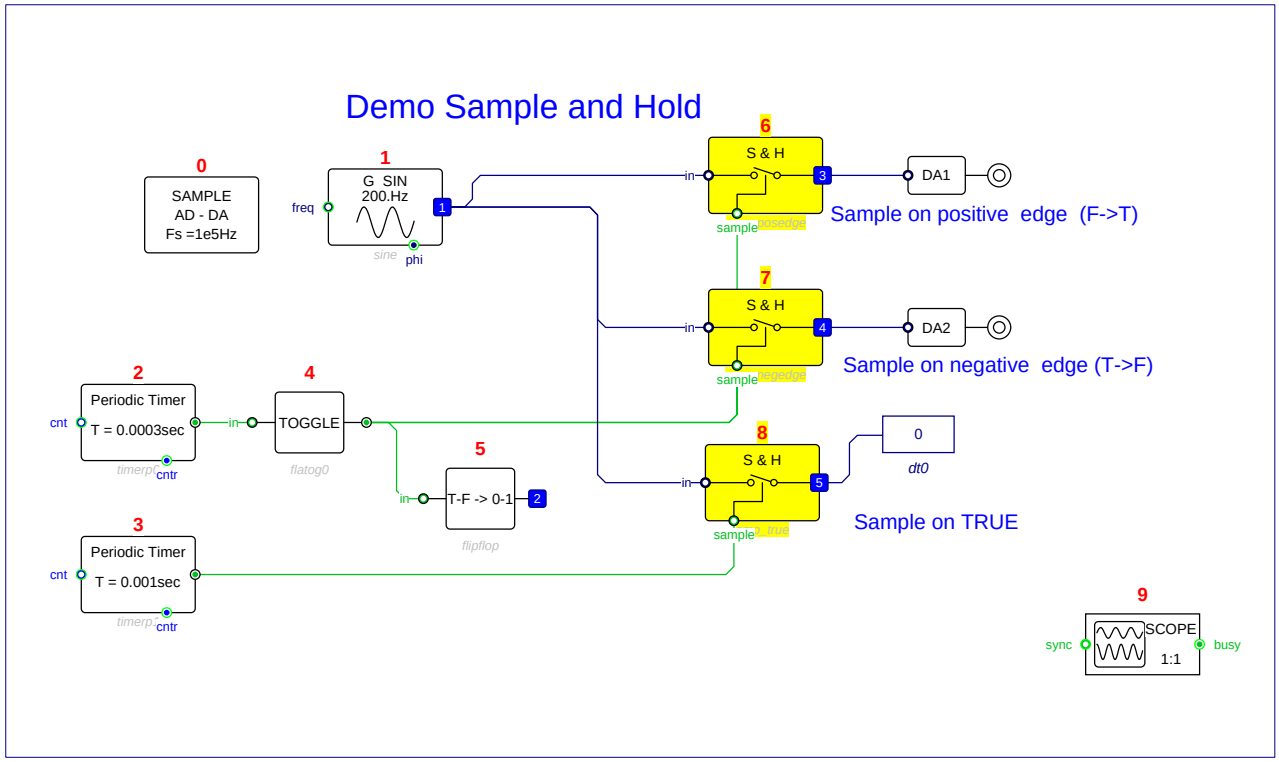
Parameter:	*Default values:*
Sample on ..	pos_edge,neg_edge,true

INPUTS

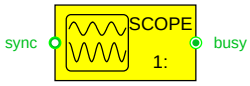
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	defined by cn		mandatory
name_sample	BOOL	BIT	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	defined by cn		normal



SAMPHOLD test program



CATEGORY: Instruments

DESCRIPTION:
Multi Channel Scope

PARAMETERS:

Parameter:	*Default values:*
Decimation factor	1

INPUTS

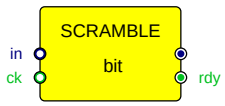
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_sync	BOOL	BIT	optional

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_busy	BOOL	BIT	optional

ATTRIBUTES
Unique,

SCRAMBLE N-bit scrambler SCRAMBLE



CATEGORY: Telecom

DESCRIPTION:
N-bit scrambler
for equalizing symbol distribution

PARAMETERS:

Parameter:	*Default values:*
Bits	1

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory
name_ck	BOOL	BIT	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal
name_rdy	BOOL	BIT	normal

SDRAM

Install SDRAM PortA interface

SDRAM

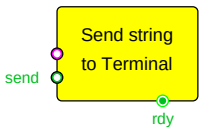
DESCRIPTION:
Install SDRAM PortA interface

ATTRIBUTES
Execute at Init, Unique,

SEND_STR

Send string

SEND_STR



CATEGORY: String

DESCRIPTION:
Send string
to text Terminal

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	STRING	WORD	mandatory
name_send	BOOL	BIT	mandatory

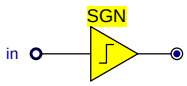
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_rdy	BOOL	BIT	optional

SGN

Sign function y=+1 if x>=0; y=-1 if x<0

SGN



CATEGORY: Non linear

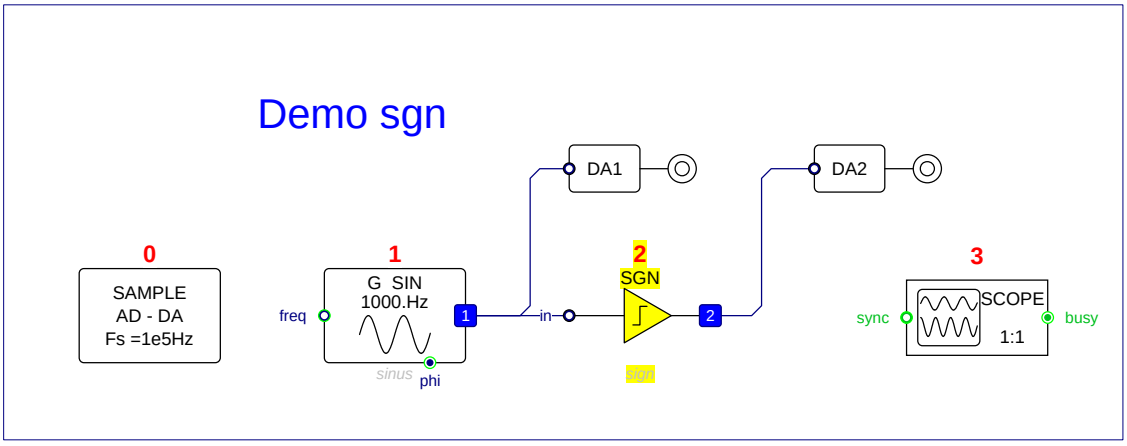
DESCRIPTION:
Sign function y=+1 if x>=0; y=-1 if x<0

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

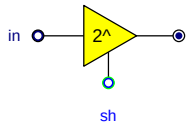
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



SGN test program

Gain by 2^N

SHIFT



CATEGORY: Arithmetic

DESCRIPTION:
Gain by 2^N
if N>0 then left arithmetic shift by N
else right arithmetic shift by N

PARAMETERS:

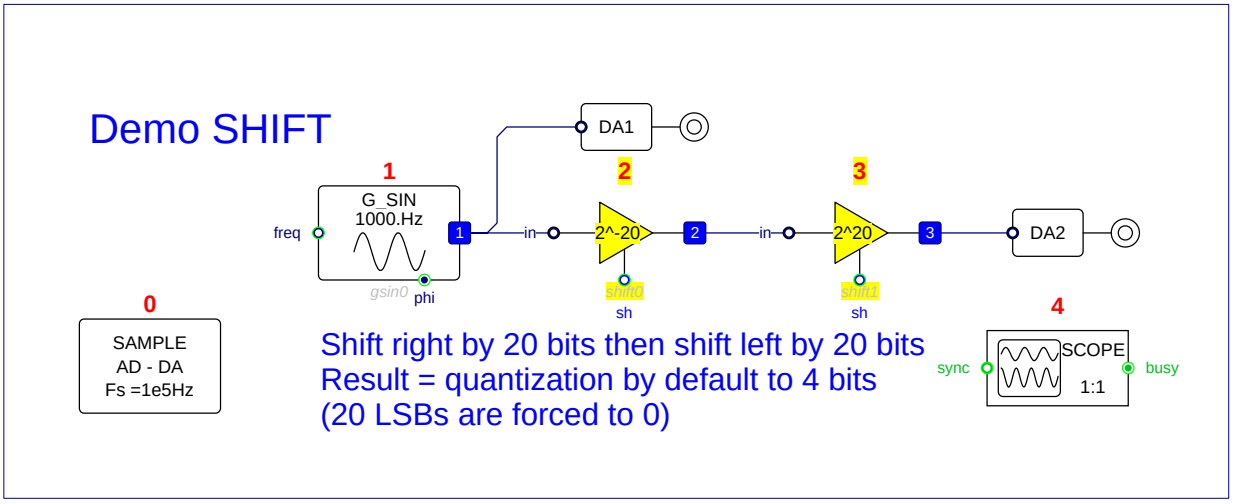
Parameter:	*Default values:*
N	1

INPUTS

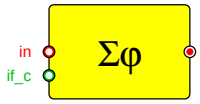
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory
name_sh	INTEGER	WORD	optional

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



SHIFT test program



CATEGORY: Telecom

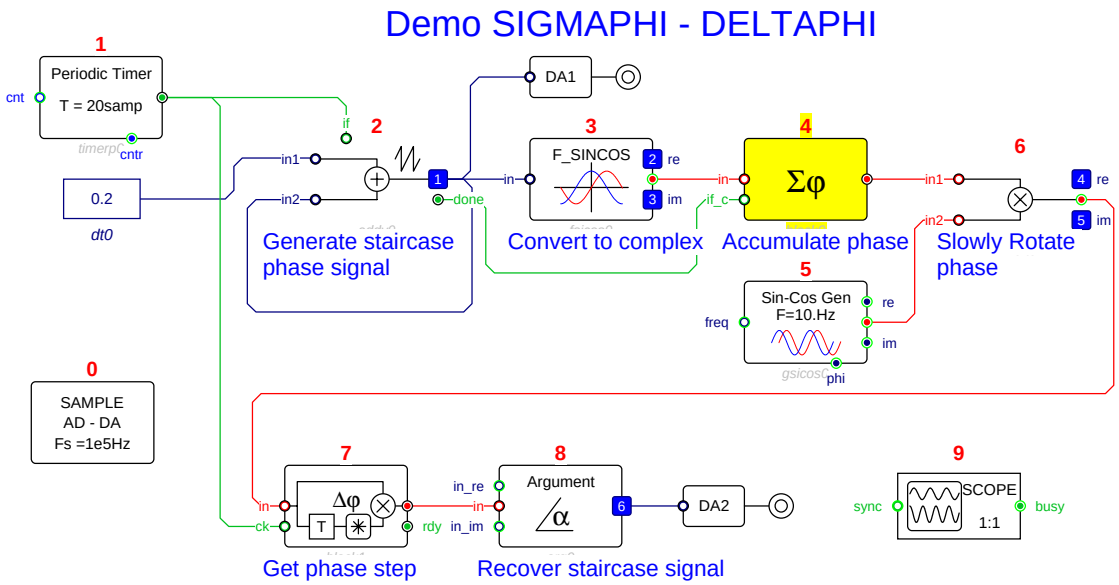
DESCRIPTION:
Phase accumulator

INPUTS

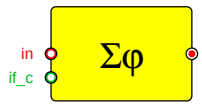
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	COMPLEX	WORD	mandatory
name_if_c	BOOL	BIT	mandatory

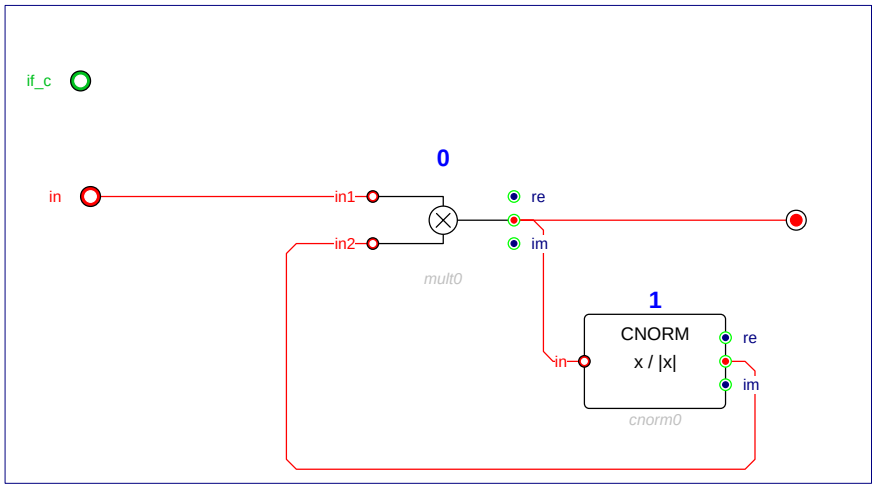
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	COMPLEX	WORD	normal



SIGMAPHI test program





SIGMAPHI internal schema

SLOPELIM

Slope limiting filter

SLOPELIM

CATEGORY: Filters

DESCRIPTION:
Slope limiting filter
for EMI spikes suppression.

PARAMETERS:

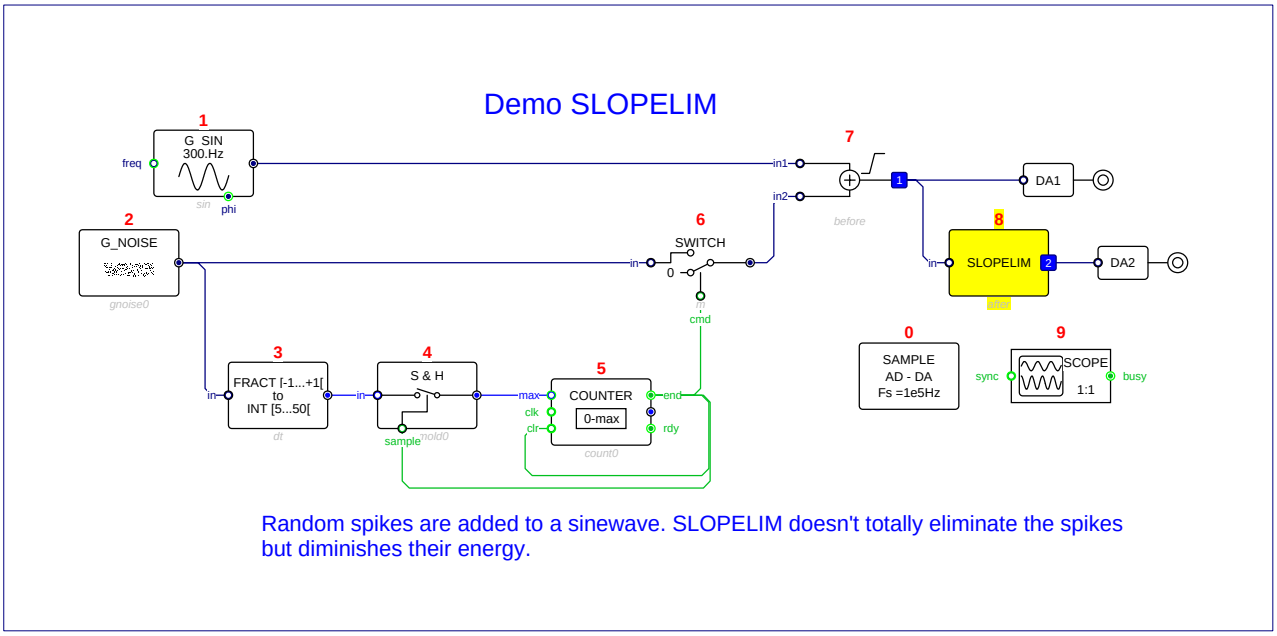
Parameter:	*Default values:*
slope	0.01
slopeneg	0.01

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

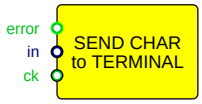
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



SLOPELIM test program

Send char to serial port

SNDC



CATEGORY: Telecom

DESCRIPTION:
Send char to serial port
On sndc_ck, send single character to Serial port
If sndc_error TRUE, terminal color = red else color=yellow

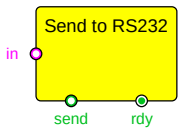
INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory
name_ck	BOOL	BIT	mandatory
name_error	BOOL	BIT	optional

ATTRIBUTES
Unique,

Send string to RS232 port

SNDS



CATEGORY: Control

DESCRIPTION:
Send string to RS232 port

PARAMETERS:

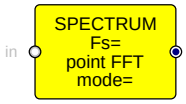
Parameter:	*Default values:*
String name	s0

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	STRING	WORD	mandatory
name_send	BOOL	BIT	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_rdy	BOOL	BIT	normal



CATEGORY: Instruments

DESCRIPTION:
Spectrum Analyser
Real or complex input. Modes Amp, Power and DB
Outputs a periodic scan of spectrum with negative sync pulses
Specan runs in main loop.
Sampled application runs inside Timer0 interrupt.

PARAMETERS:

Parameter:	*Default values:*
Sampling frequency	1e5
FFT Size	1024,512,256,128,64,32,16,8
Display	full,half
Mode	dB,Amp,Pow
Window	Rectangle,Triangle,Hann,Hamming,Nuttall,Gauss,Blackman_Harris,Flat_Top
Highpass ?	ac,dc

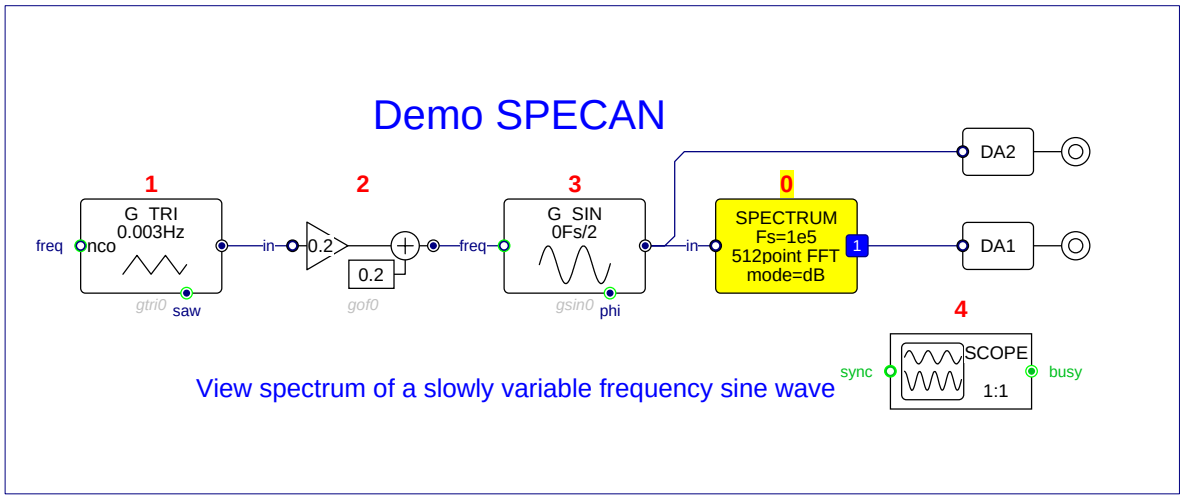
INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	defined by cn		mandatory

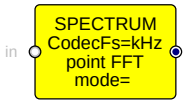
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

ATTRIBUTES
Unique, Execute First, Defines: actual_fs



SPECAN test program



CATEGORY: Audio

DESCRIPTION:
Spectrum Analyser
Includes CODEC I/Os
Real or complex input. Modes Amp, Power and DB
Outputs a periodic scan of spectrum with negative sync pulses
Specan runs in main loop.
Sampled application runs inside CODEC interrupt.

PARAMETERS:

Parameter:	*Default values:*
Fs (kHz)	8,32,48,96
FFT Size	1024,512,256,128,64,32,16,8
Display	full,half
Mode	db,amp,pow
Windowing ?	win,nowin
Highpass ?	ac,dc
Input	line,mike

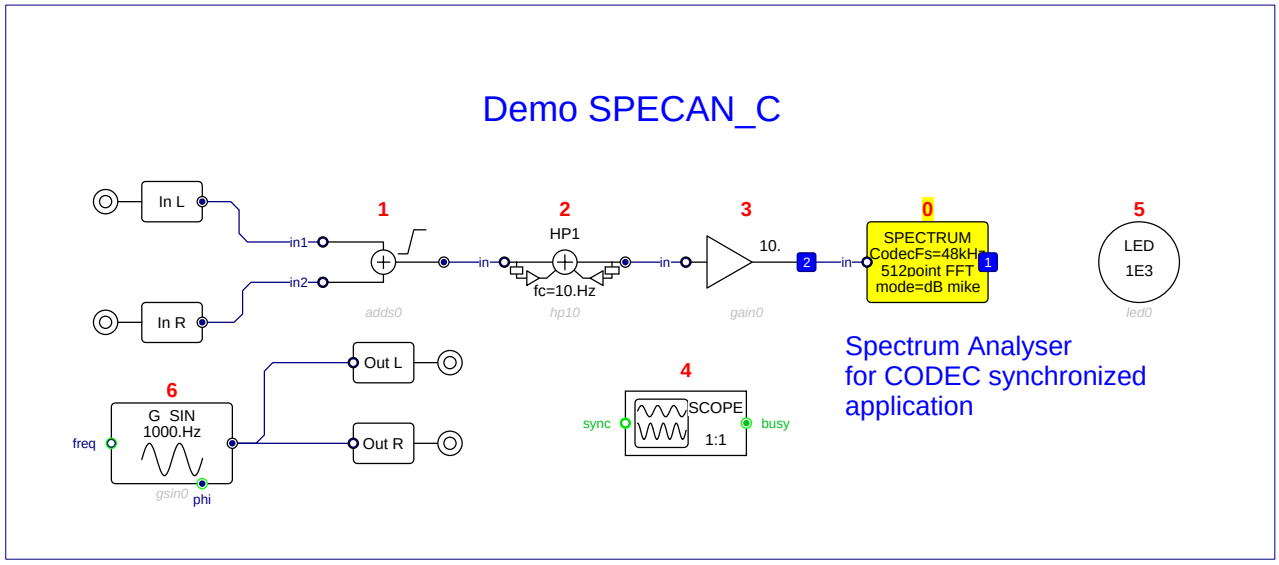
INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	defined by cn		mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

ATTRIBUTES
Unique, Execute First, Defines: actual_fs

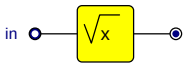


SPECAN_C test program

SQROOT

Square root of input

SQROOT



CATEGORY: Functions

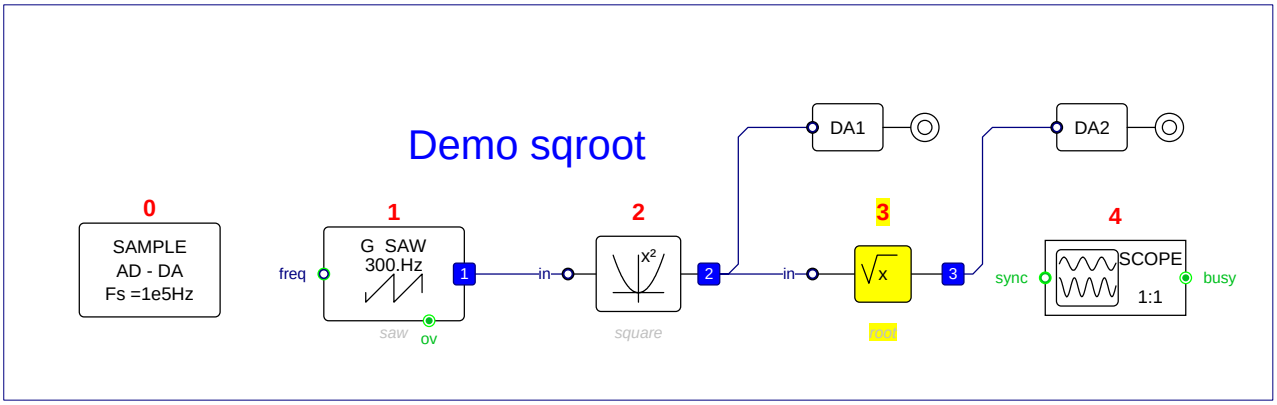
DESCRIPTION:
Square root of input

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

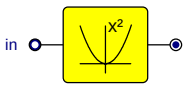
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



SQROOT test program

Square of input

SQUARE



CATEGORY: Functions

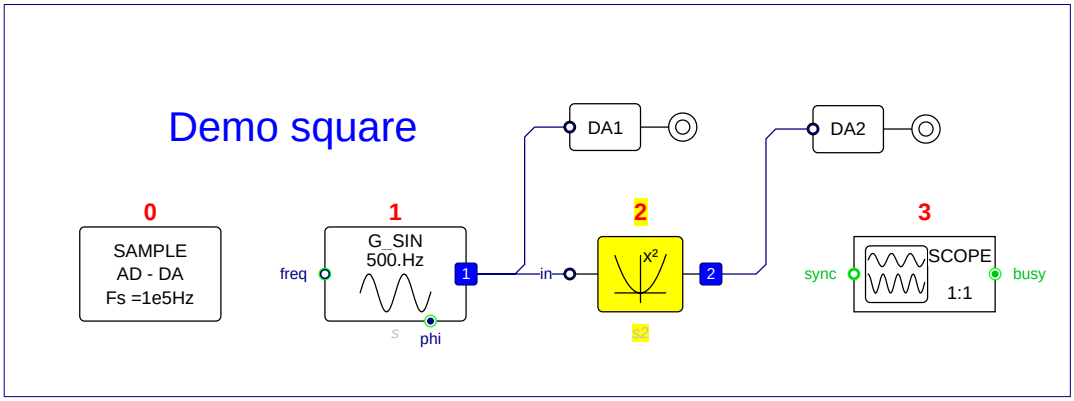
DESCRIPTION:
Square of input

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	DWORD	normal



SQUARE test program

STOP Stop program and return to debugger STOP

CATEGORY: Control

DESCRIPTION:
Stop program and return to debugger

Matrix to Hex

STRNH

CATEGORY: String

DESCRIPTION:
Matrix to Hex
Convert matrix data to a string of hex numbers

PARAMETERS:

Parameter:	*Default values:*
name	
1	digits,1,6,msb_pos,1,23,separator,3,spc,cr
ln	

INPUTS

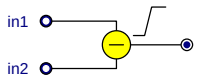
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	ATRIX	Matrix of WORD	mandatory
name_strin	STRING	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_strout	STRING	WORD	normal

Subtraction with saturation

SUBS



CATEGORY: Arithmetic

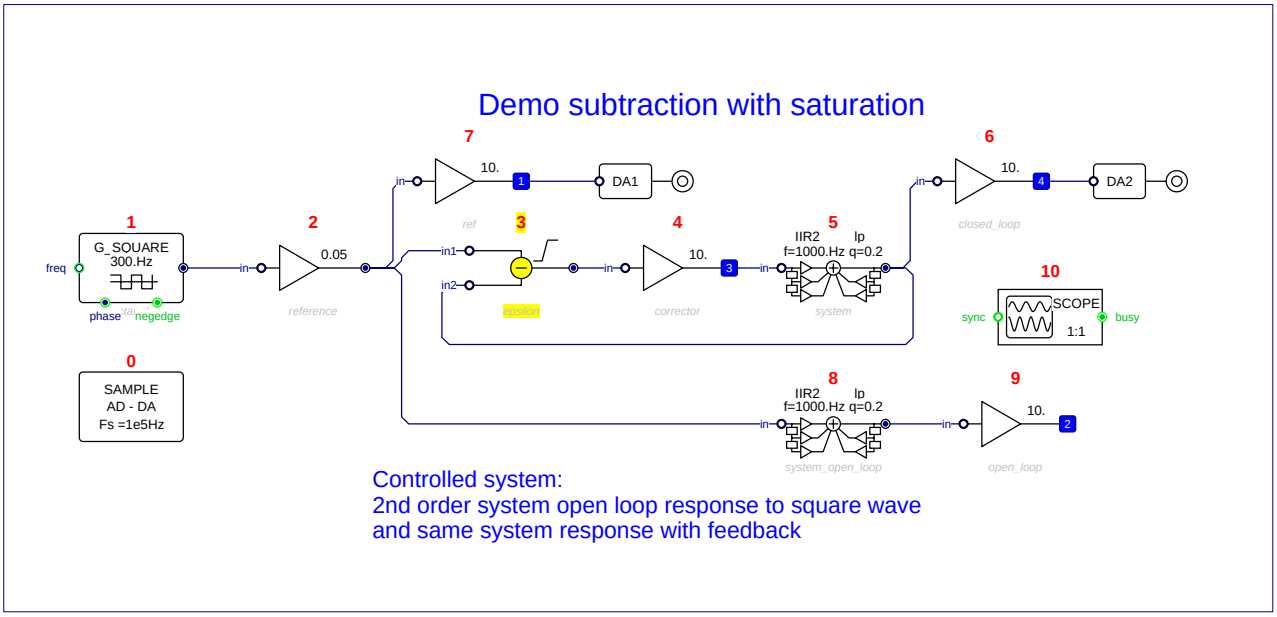
DESCRIPTION:
Subtraction with saturation
Output = in1 - in2

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in1	FRACT	WORD	mandatory
name_in2	FRACT	WORD	mandatory

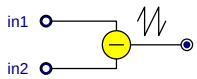
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



SUBS test program

Subtraction modulo +/- 1



CATEGORY: Arithmetic

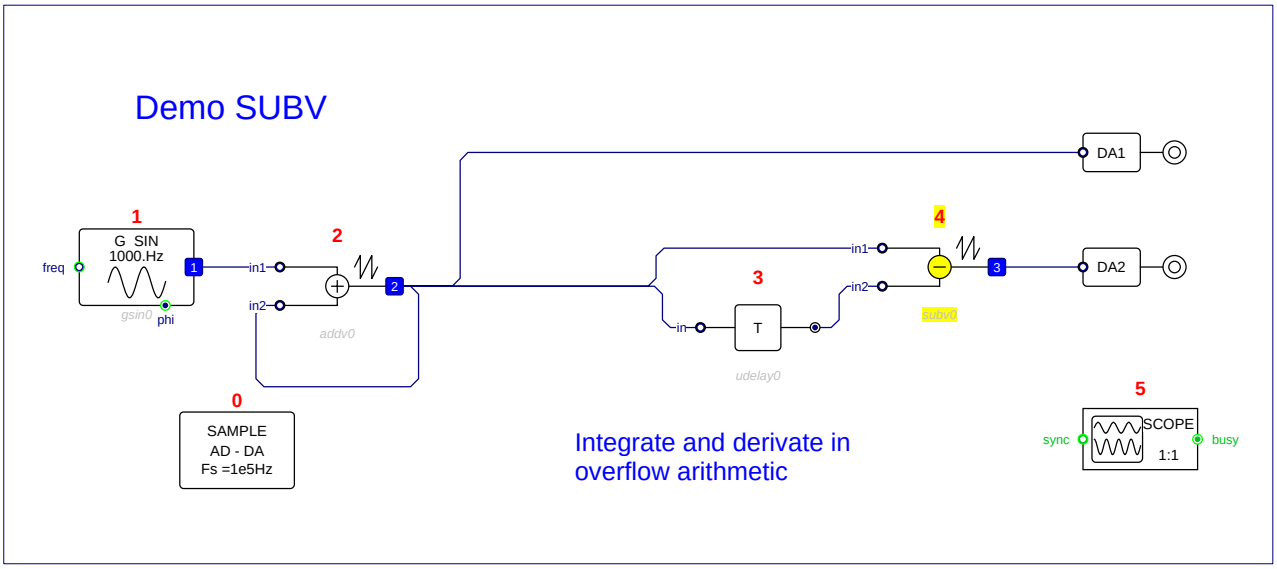
DESCRIPTION:
Subtraction modulo +/- 1

INPUTS

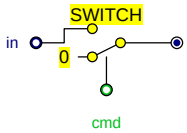
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in1	FRACT	WORD	mandatory
name_in2	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



SUBV test program



CATEGORY: Control

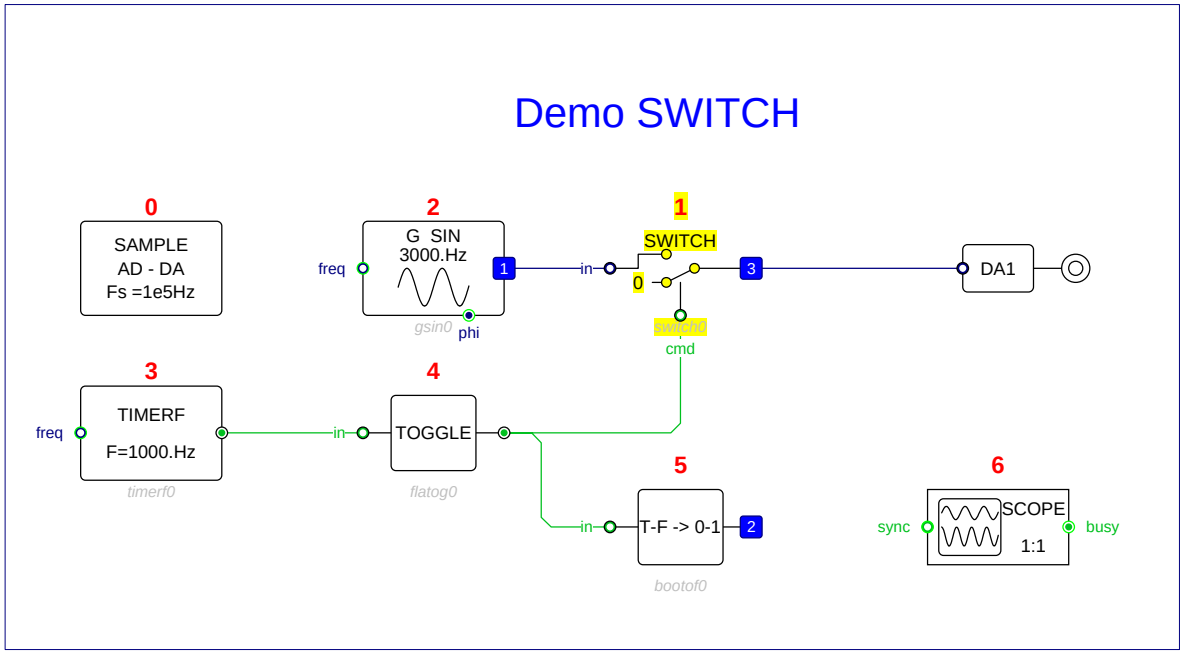
DESCRIPTION:
Switch.
Output= input if cmd=true, else output=0

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory
name_cmd	BOOL	BIT	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

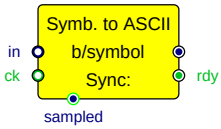


SWITCH test program

SYMTOASC

Symbols to ASCII

SYMTOASC



CATEGORY: Telecom

DESCRIPTION:
Symbols to ASCII
Concatenate symbols to ASCII characters.
Executes on ck true, then resets ck
Synchronize on 16 bit pattern.

PARAMETERS:

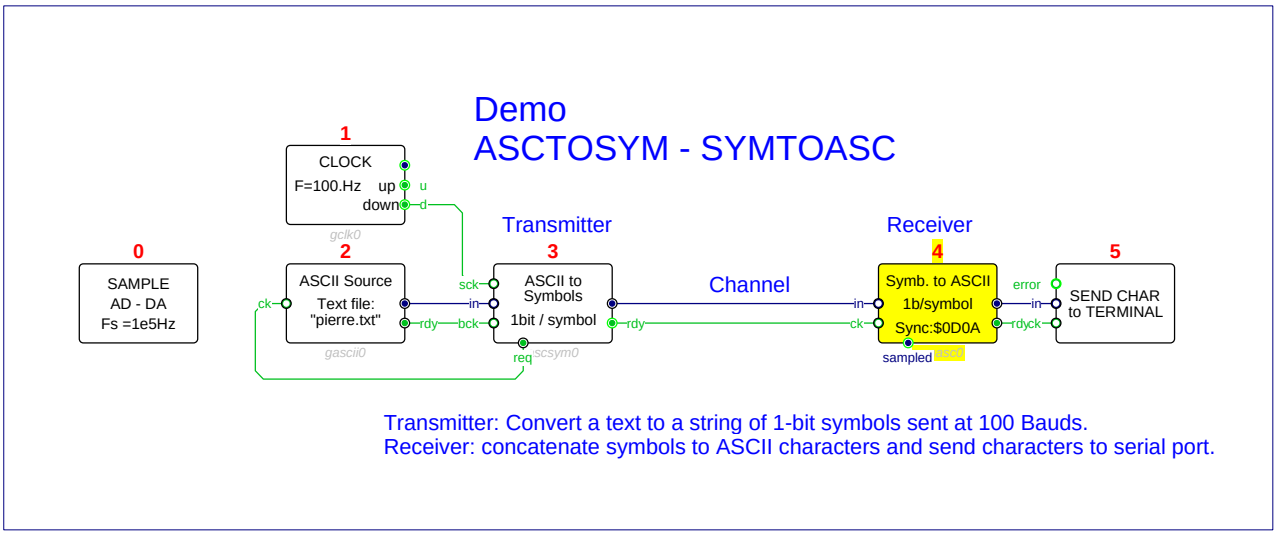
Parameter:	*Default values:*
Bits per Symbol	1
Sync Word	$0D0A

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory
name_ck	BOOL	BIT	mandatory

OUTPUTS

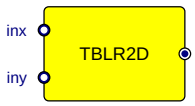
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_sampled	FRACT	WORD	optional
name	FRACT	WORD	normal
name_rdy	BOOL	BIT	normal



SYMTOASC test program

2-D Table read and interpolate

TBLR2D



CATEGORY: Functions

DESCRIPTION:
2-D Table read and interpolate

PARAMETERS:

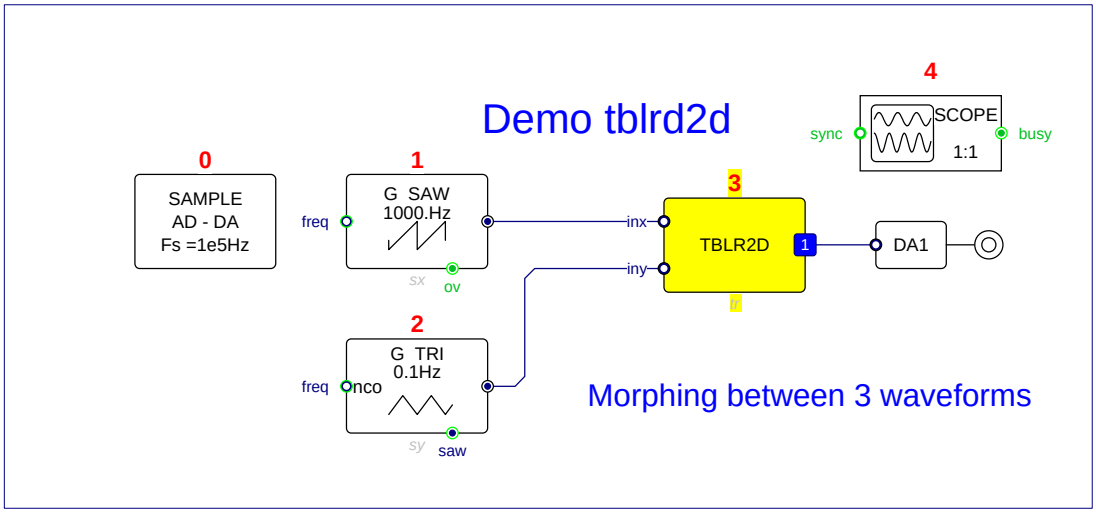
Parameter:	*Default values:*
table	atable
XSize	10
X signed/unsigned	s,u
Y signed/unsigned	s,u

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_inx	FRACT	WORD	mandatory
name_iny	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

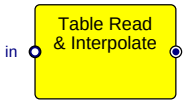


TBLR2D test program

TBLRD

Table read and interpolate

TBLRD



CATEGORY: Functions

DESCRIPTION:
Table read and interpolate

PARAMETERS:

Parameter:	*Default values:*
table	atable
Signed/Unsigned inp.	s,u

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

Start H/W Cycle Counter

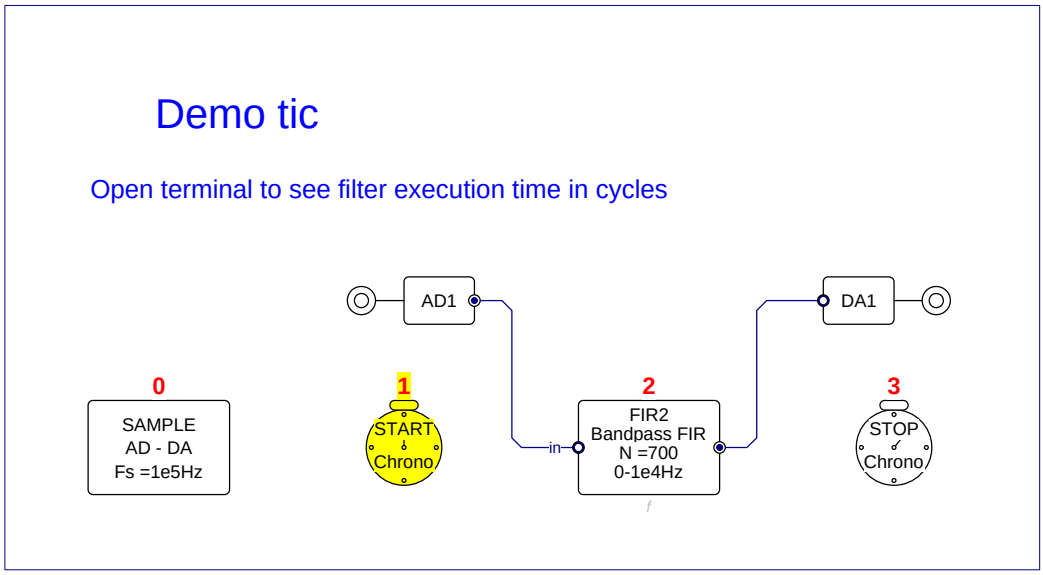
CATEGORY: Timing

DESCRIPTION:
Start H/W Cycle Counter

PARAMETERS:

Parameter:	*Default values:*
Timer #	1,2,0

ATTRIBUTES
Unique,

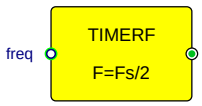


TIC test program

TIMERF

Periodic Timer defined by a frequency

TIMERF



CATEGORY: Timing

DESCRIPTION:
Periodic Timer defined by a frequency
Sets a flag at rate freq

PARAMETERS:

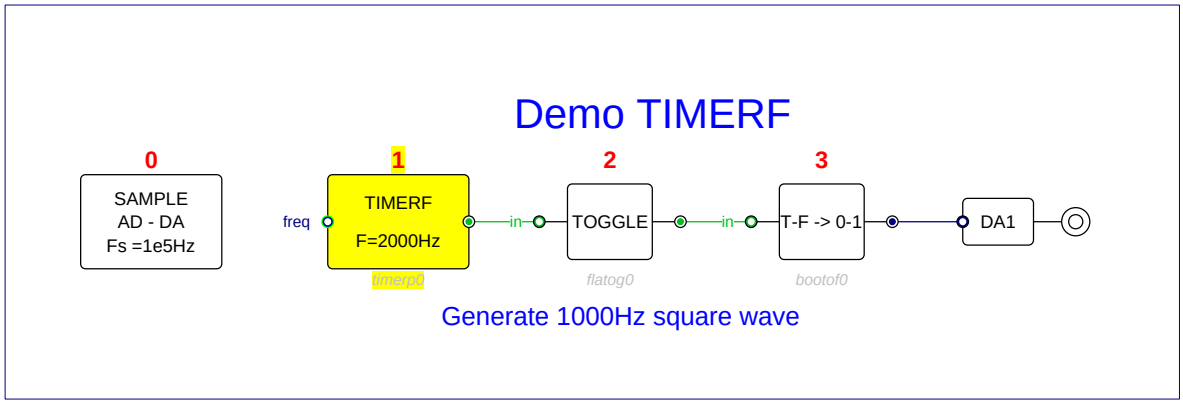
Parameter:	*Default values:*
Frequency	1000.
unit	Hz,Fs/2

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_freq	FRACT	WORD	optional

OUTPUTS

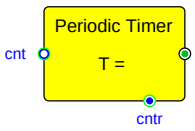
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	BOOL	BIT	normal



TIMERF test program

Periodic Timer

TIMERP



CATEGORY: Timing

DESCRIPTION:
Periodic Timer
Sets a flag every N sampling periods

PARAMETERS:

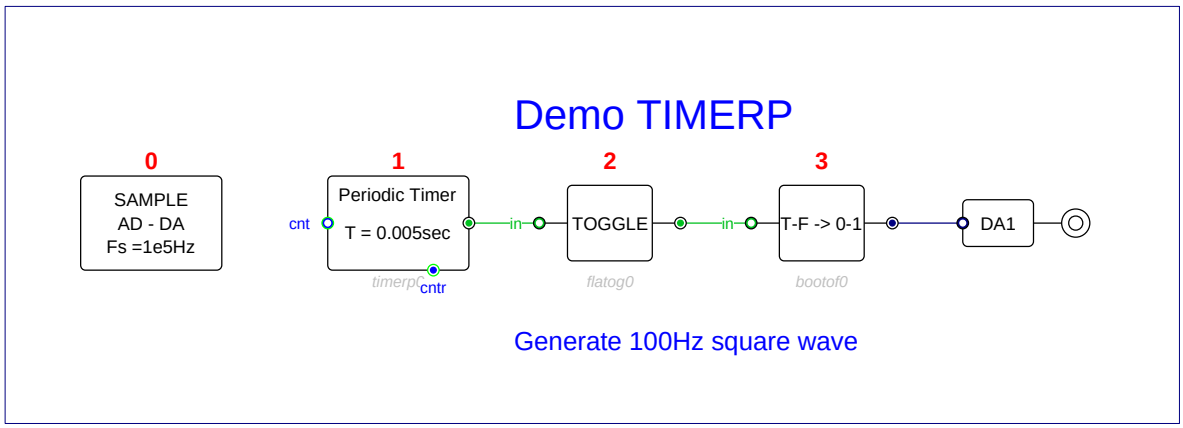
Parameter:	*Default values:*
Period	0.01
Unit	sec,samp

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_cnt	INTEGER	WORD	optional

OUTPUTS

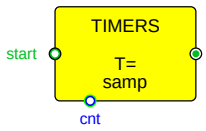
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	BOOL	BIT	normal
name_cntr	INTEGER	WORD	optional



TIMERP test program

One shoot Timer.

TIMERS



CATEGORY: Timing

DESCRIPTION:
One shoot Timer.
mode hwb sets a flag at start, clears it at timeout
mode hto only sets a flag at timeout.
retriggerable means the timer can be restarted while busy

PARAMETERS:

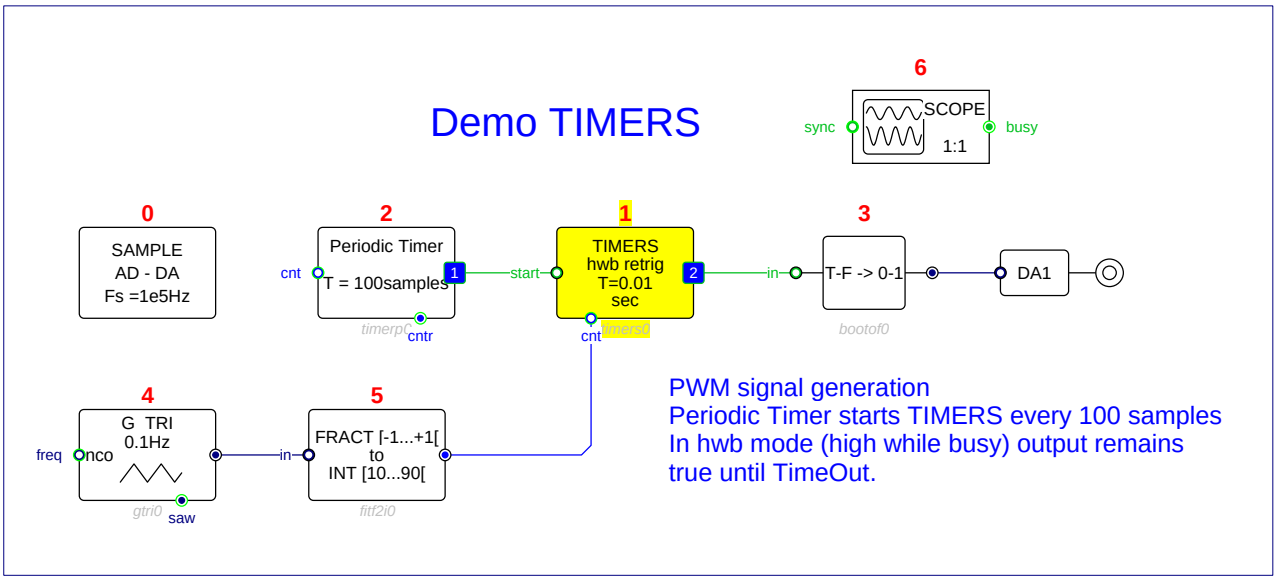
Parameter:	*Default values:*
High busy or high at TO ?	hwb,hto
Retriggerable ?	retrig,non_retrig
Time	0.01
Unit	sec,samp

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_start	BOOL	BIT	mandatory
name_cnt	INTEGER	WORD	optional

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	BOOL	BIT	normal



TIMERS test program

Stop Chrono

TOC

CATEGORY: Timing

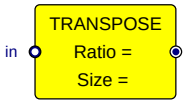
DESCRIPTION:
Stop Chrono
Stop H/W Cycle Counter and display on terminal Min and Max cycle counts
Occurences is the number of measurements before results are displayed

PARAMETERS:

Parameter:	*Default values:*
Timer #	1,2,0
Occurences	10

ATTRIBUTES
Unique,

TRANSPOSE Transpose TRANSPOSE



CATEGORY: Audio

DESCRIPTION:
Transpose
Change pitch of melody or voice by multiplying frequencies by given ratio
fi(out) = fi(in)*k
Size = rotating buffer size (samples)

PARAMETERS:

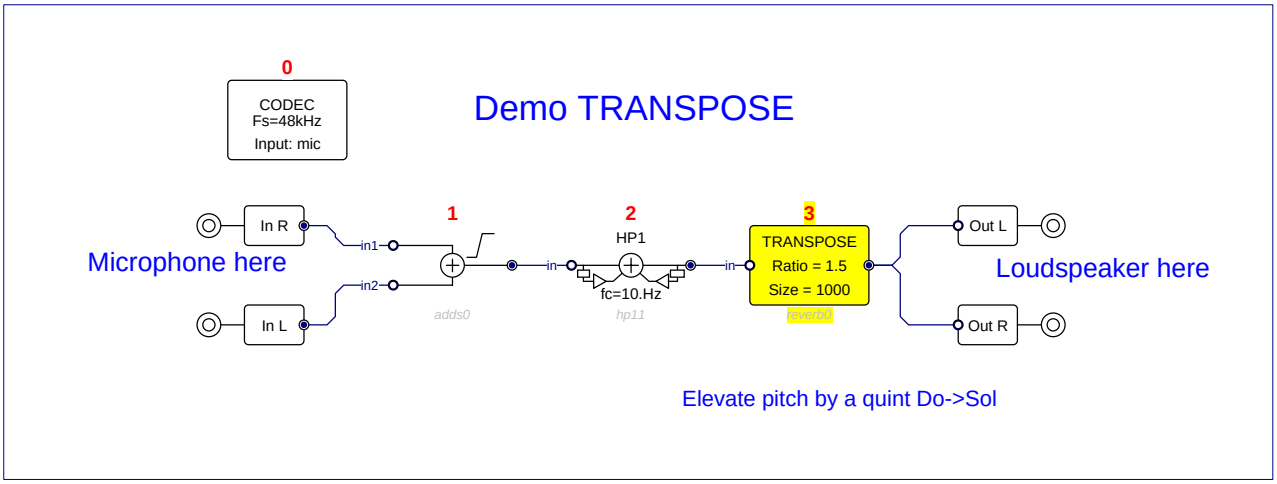
Parameter:	*Default values:*
Size	1000
Ratio	1.5

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



TRANSPOSE test program

TRAP

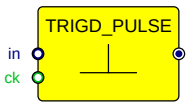
Hang here (infinite loop)

TRAP

CATEGORY: Control

DESCRIPTION:
Hang here (infinite loop)

TRIGD_PULSE Triggered pulse TRIGD_PULSE



CATEGORY: Generators

DESCRIPTION:
Triggered pulse
Generate pulse with amplitude given by input on clock true

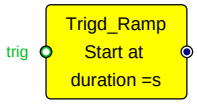
INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory
name_ck	BOOL	BIT	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

TRIGRAMP Software Triggered Ramp. TRIGRAMP



CATEGORY: Generators

DESCRIPTION:
Software Triggered Ramp.
While out not saturated do out=out+delta.
On trig, set output to defined value.

PARAMETERS:

Parameter:	*Default values:*
Start value	-1.0
Duration	0.3

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_trig	BOOL	BIT	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

TTL_IN1

Digital Input 1

TTL_IN1

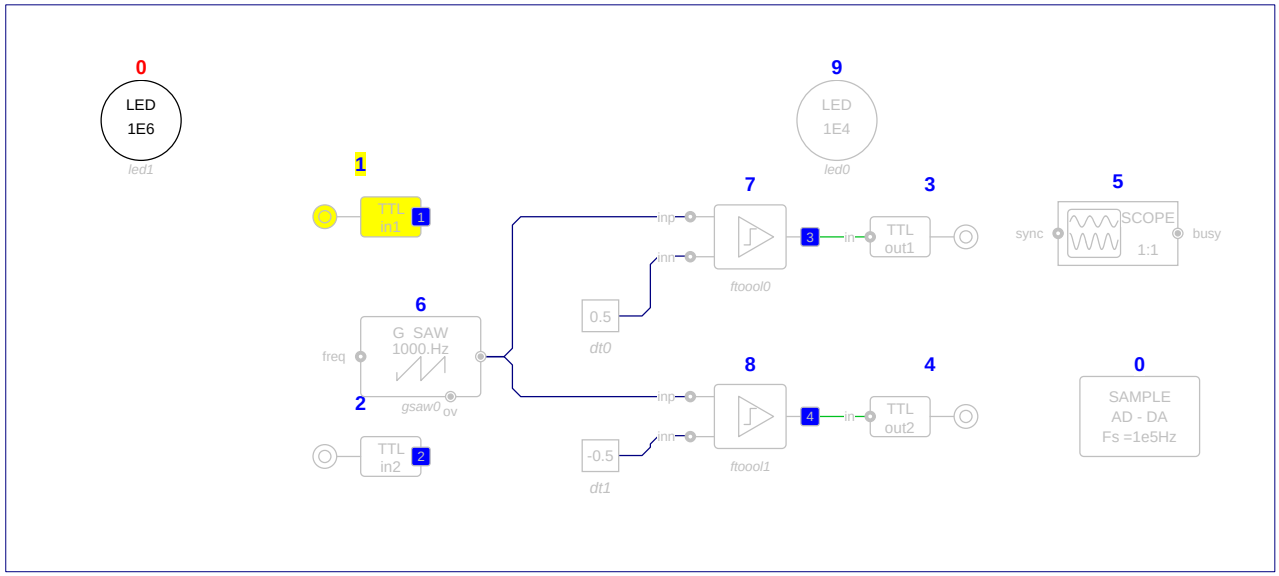
CATEGORY: Logic

DESCRIPTION:
Digital Input 1
Bool Output reflects state of IRQC pin

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	BOOL	BIT	normal

ATTRIBUTES
Unique,



TTL_IN1 test program

Digital Input 2

TTL_IN2

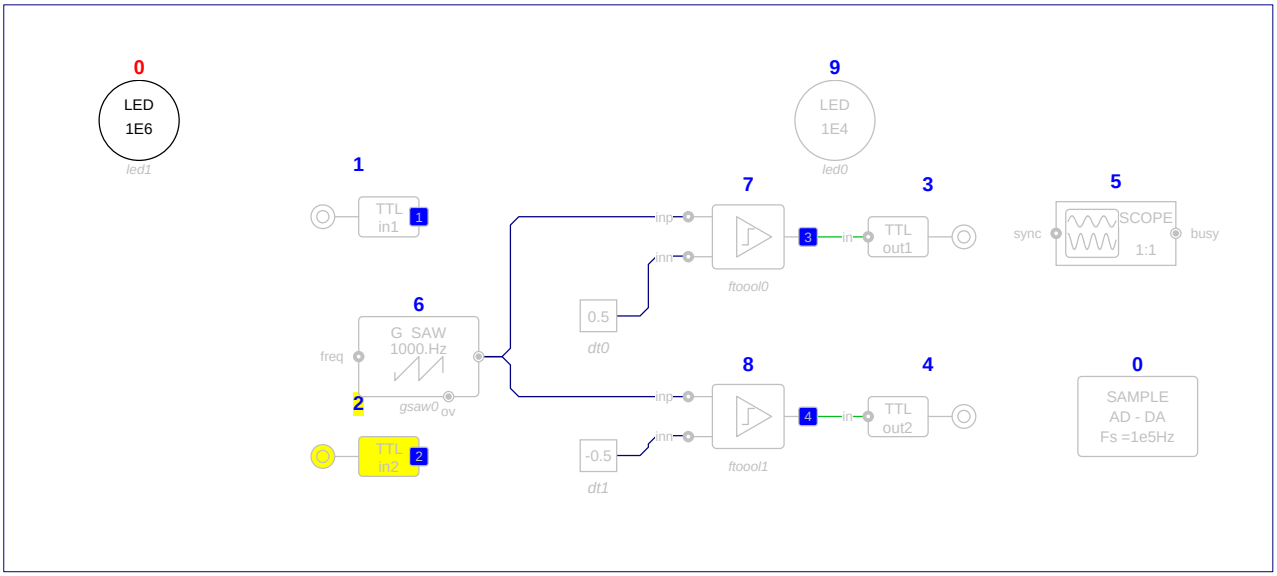
CATEGORY: Logic

DESCRIPTION:
Digital Input 2
Output reflects state of Core1 PC9 pin

OUTPUTS

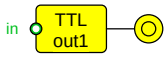
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	BOOL	BIT	normal

ATTRIBUTES
Unique,



TTL_IN2 test program

TTL_OUT1 Digital output 1 TTL_OUT1



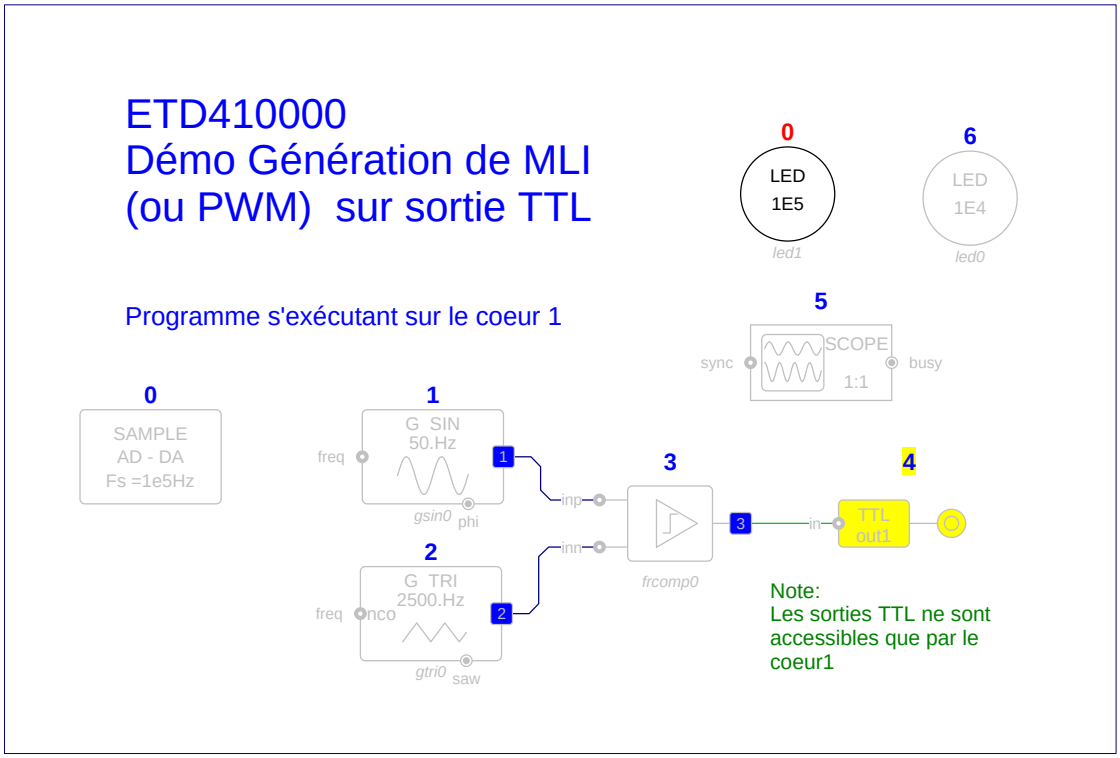
CATEGORY: Logic

DESCRIPTION:
Digital output 1
Output State (Core1, pin PE8) is given by boolean input

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	BOOL	BIT	mandatory

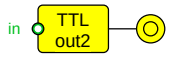
ATTRIBUTES
Unique,



TTL_OUT1 test program

Digital output 1

TTL_OUT2



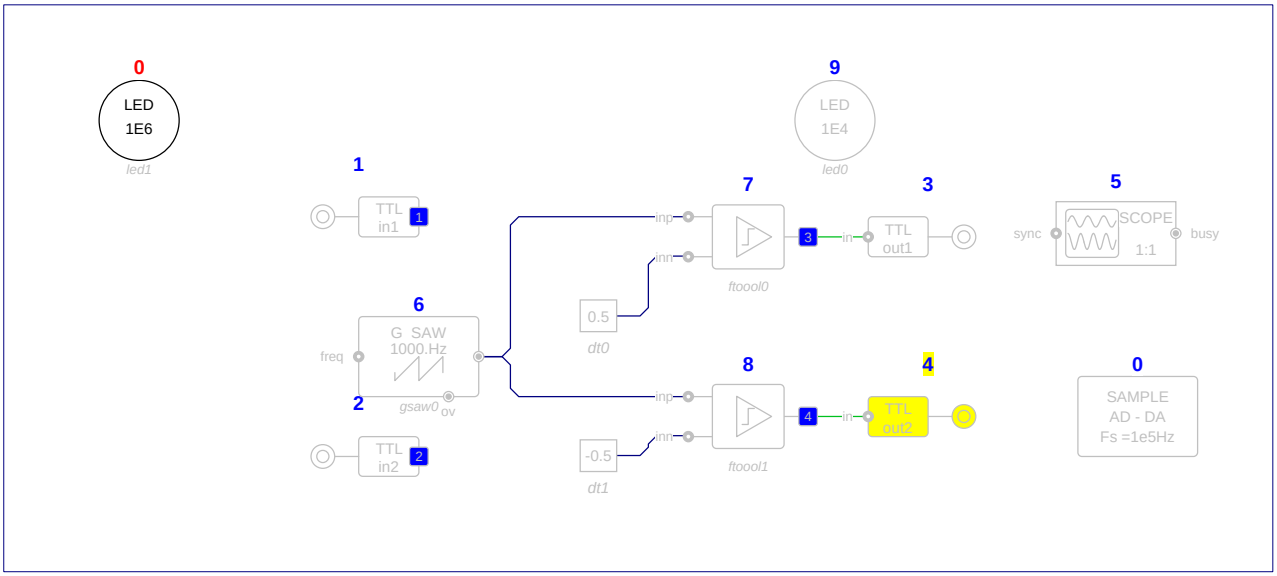
CATEGORY: Logic

DESCRIPTION:
Digital output 1
Output State (Core1, pin PE7) is given by boolean input

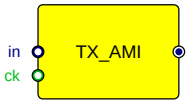
INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	BOOL	BIT	mandatory

ATTRIBUTES
Unique,



TTL_OUT2 test program



CATEGORY: Telecom

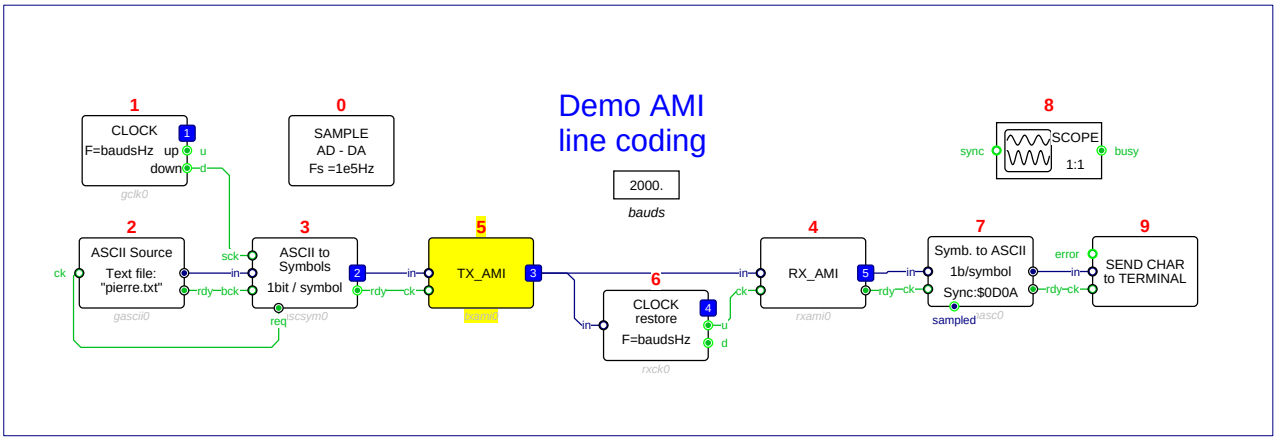
DESCRIPTION:
Alternate Mark Inversion line coder

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory
name_ck	BOOL	BIT	mandatory

OUTPUTS

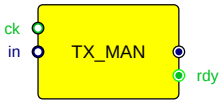
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



TX_AMI test program

Manchester line coder

TX_MAN



CATEGORY: Telecom

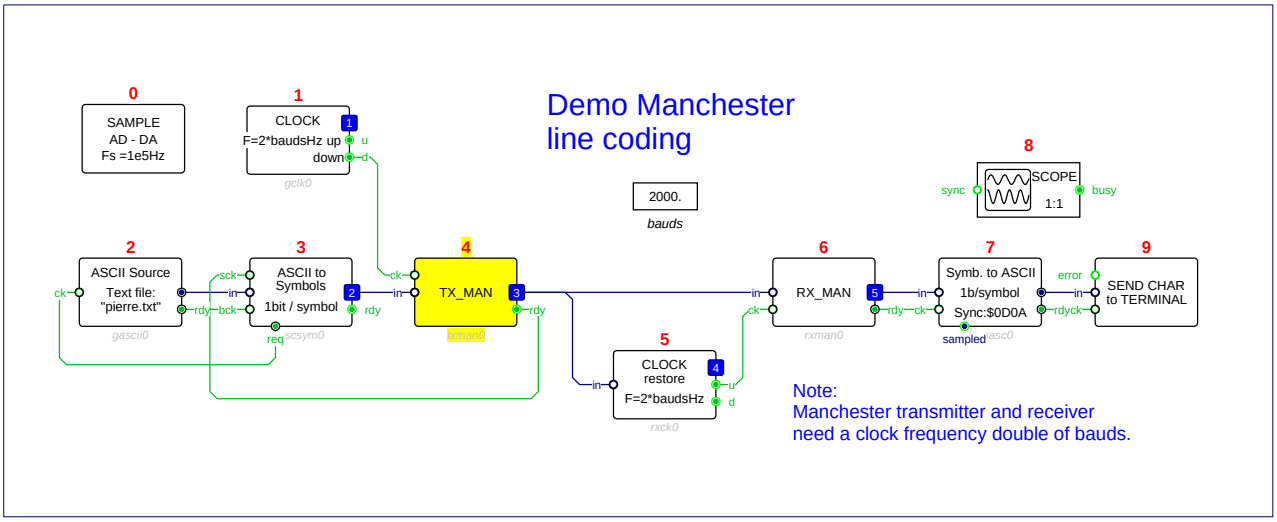
DESCRIPTION:
Manchester line coder
Input clock is 2 x Bauds

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory
name_ck	BOOL	BIT	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal
name_rdy	BOOL	BIT	optional



TX_MAN test program

TX_MAND Differential Manchester line coder TX_MAND

CATEGORY: Telecom

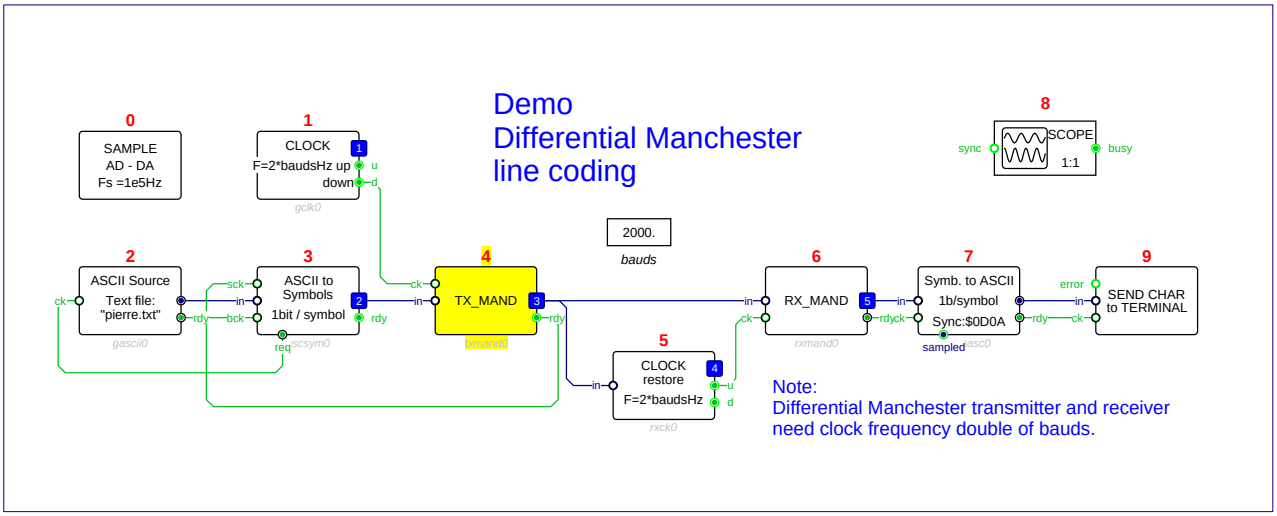
DESCRIPTION:
Differential Manchester line coder
Input clock is 2 x Bauds

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory
name_ck	BOOL	BIT	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal
name_rdy	BOOL	BIT	optional



TX_MAND test program

CATEGORY: Telecom

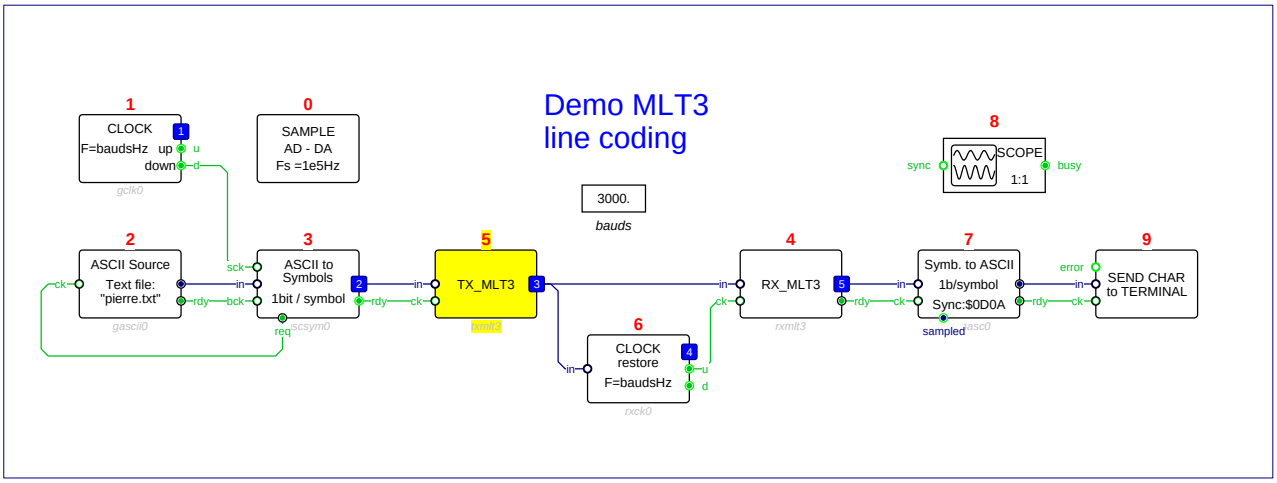
DESCRIPTION:
MLT3 line coder

INPUTS

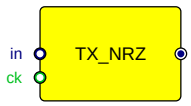
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory
name_ck	BOOL	BIT	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



TX_MLT3 test program



CATEGORY: Telecom

DESCRIPTION:
Non Return to Zero line coder

PARAMETERS:

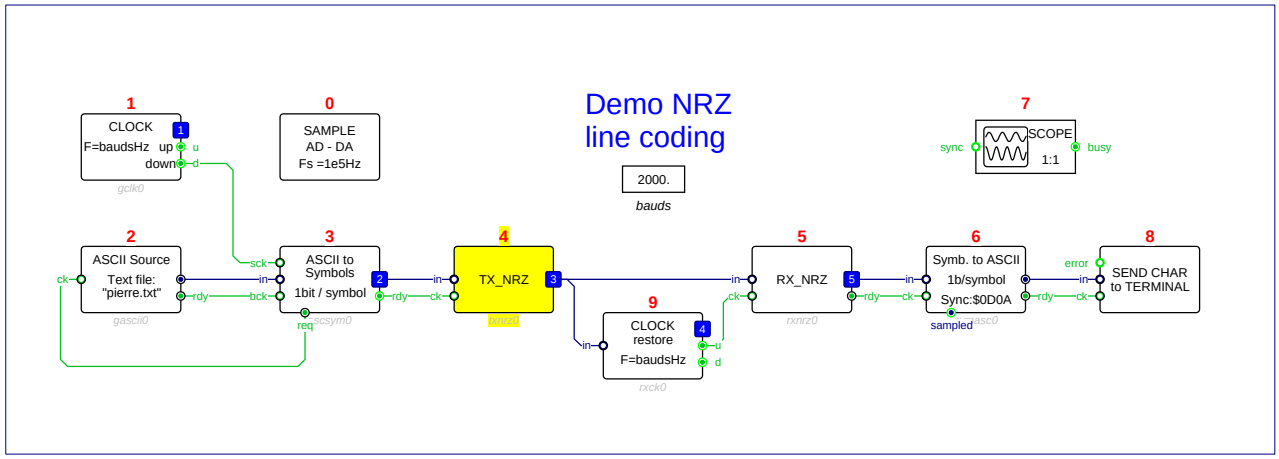
Parameter:	*Default values:*
Level Space	-1.0
Level Mark	1.0

INPUTS

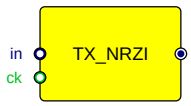
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory
name_ck	BOOL	BIT	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



TX_NRZ test program



CATEGORY: Telecom

DESCRIPTION:
NRZI line coder

PARAMETERS:

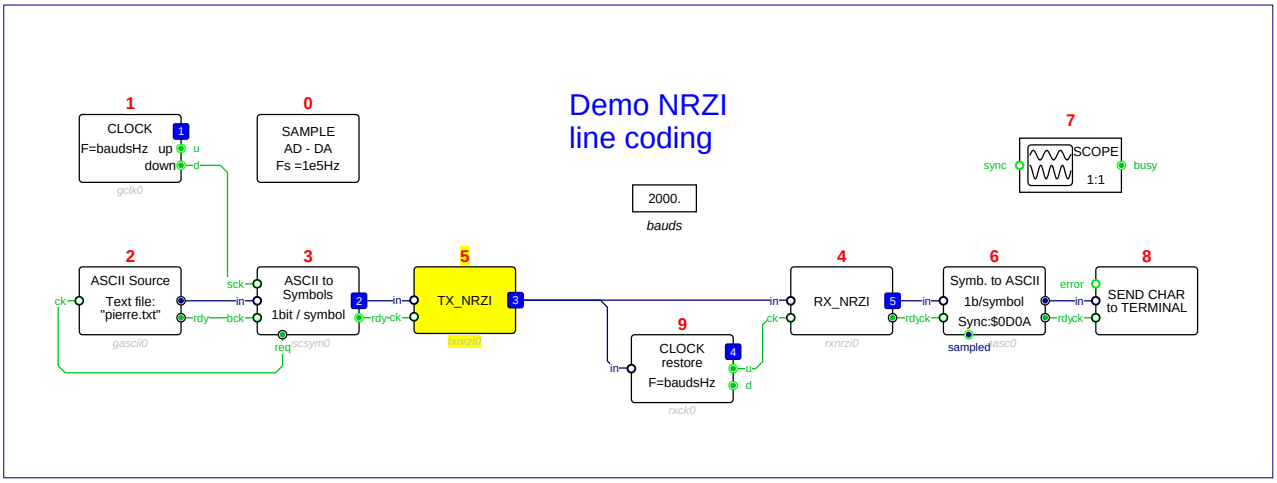
Parameter:	*Default values:*
Level Space	-1.0
Level Mark	1.0

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory
name_ck	BOOL	BIT	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



TX_NRZI test program

CATEGORY: Telecom

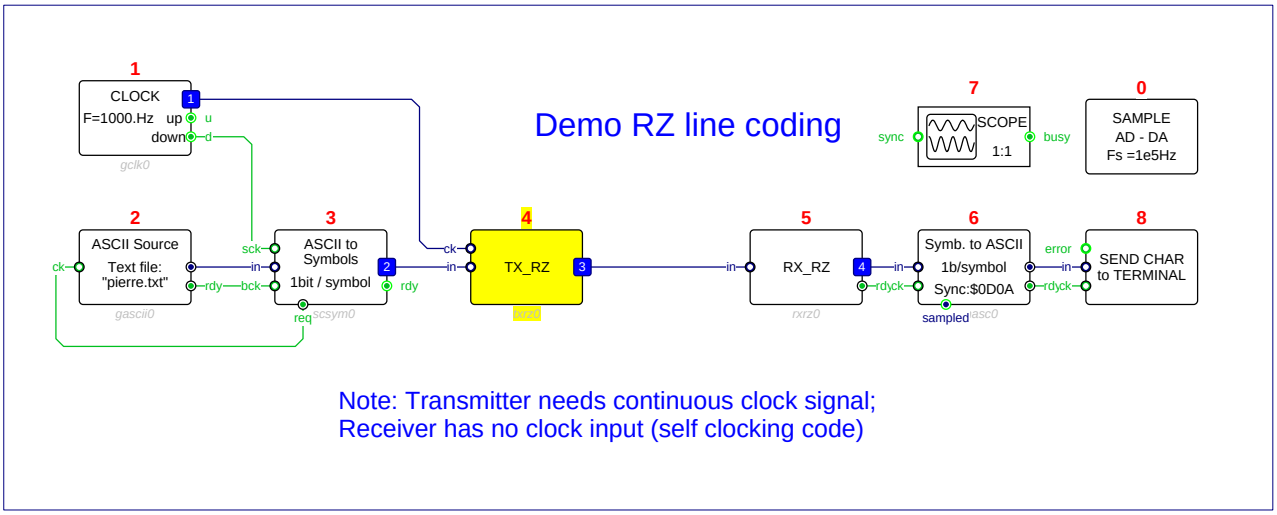
DESCRIPTION:
Return to Zero line coder

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory
name_ck	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

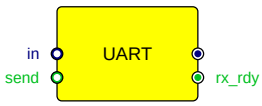


TX_RZ test program

UART

standart UART at 115KBauds

UART



CATEGORY: String

DESCRIPTION:
standart UART at 115KBauds

INPUTS

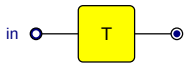
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory
name_send	BOOL	BIT	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal
name_rx_rdy	BOOL	BIT	normal

Unit delay z^-1

UDELAY



CATEGORY: Control

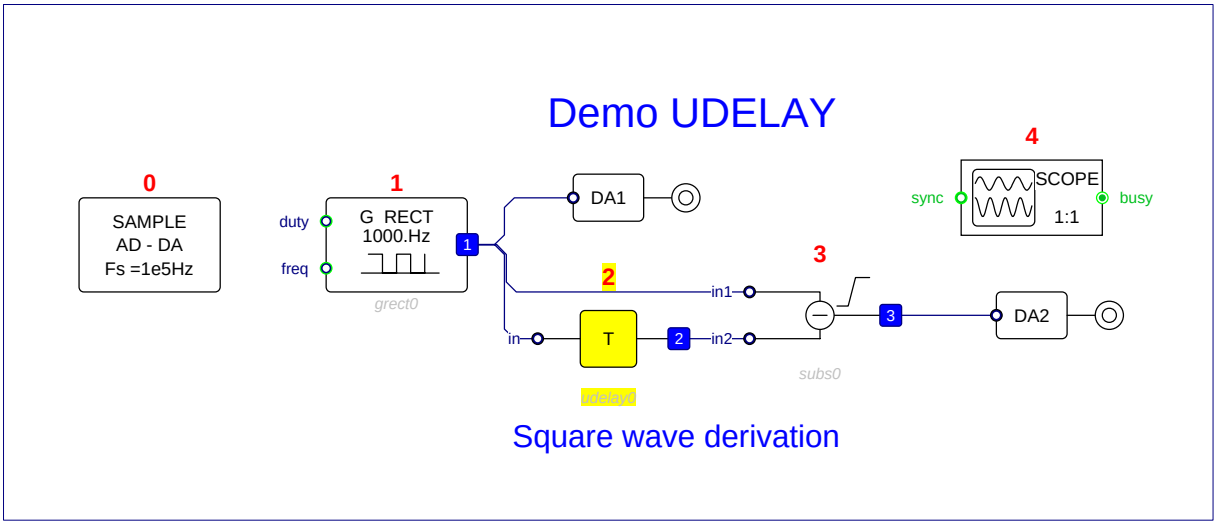
DESCRIPTION:
Unit delay z^-1

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

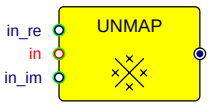


UDELAY test program

UNMAP

Complex to symbol

UNMAP



CATEGORY: Telecom

DESCRIPTION:
Complex to symbol
Retrieve symbol by searching nearest distance to constellation point

PARAMETERS:

Parameter:	*Default values:*
bits per symbol	1
Constellation	map_ook,map_bpsk,map_ask4,map_ask8,map_psk4,map_psk8,map_qam8,map_qam16

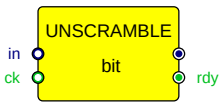
INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	COMPLEX	WORD	optional
name_in_re	FRACT	WORD	optional
name_in_im	FRACT	WORD	optional

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

UNSCRAMBLE Unscrambler UNSCRAMBLE



CATEGORY: Telecom

DESCRIPTION:
Unscrambler
N-bit unscrambler for decoding data generated by SCRAMBLE

PARAMETERS:

Parameter:	*Default values:*
Bits	1

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	WORD	mandatory
name_ck	BOOL	BIT	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal
name_rdy	BOOL	BIT	normal

VECT_POW

Vector Power

VECT_POW

CATEGORY: Matrix

DESCRIPTION:
Vector Power
$y[i] = average(|x[i]|^2)$

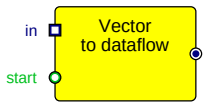
INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	defined by cn	Matrix of	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	Matrix of DWORD	normal

VECTTOFLOW Vector to dataflow VECTTOFLOW



CATEGORY: Matrix

DESCRIPTION:
Vector to dataflow

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	FRACT	Matrix of WORD	mandatory
name_start	BOOL	BIT	mandatory

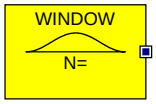
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal

WINDOW

Implement Window

WINDOW



CATEGORY: Matrix

DESCRIPTION:
Implement Window
in X: or Y: memory

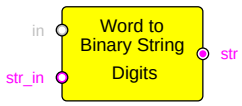
PARAMETERS:

Parameter:	*Default values:*
win	Triangle,Hann,Hamming,Gauss,Blackman_Harris,Nuttal,Flat_Top
Size	512
Integral	1.0

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	Matrix of WORD	normal

WORDTOBIN Word to Binary WORDTOBIN



CATEGORY: String

DESCRIPTION:
Word to Binary
Convert word input to binary string (zeros and ones)

PARAMETERS:

Parameter:	*Default values:*
Nb digits	24
Bit pos of MSB	23

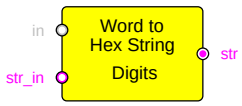
INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	defined by cn		mandatory
name_str_in	STRING	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_str	STRING	WORD	normal

WORDTOHEX Word to hexadecimal WORDTOHEX



CATEGORY: String

DESCRIPTION:
Word to hexadecimal
Convert word input to hex string

PARAMETERS:

Parameter:	*Default values:*
Nb digits	6
Bit pos of MSB	23

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in	defined by cn		mandatory
name_str_in	STRING	WORD	mandatory

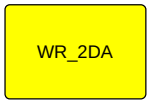
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_str	STRING	WORD	normal

WR_2DA

Write to Double DA

WR_2DA



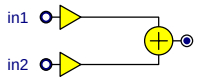
DESCRIPTION:
Write to Double DA

PARAMETERS:

Parameter:	*Default values:*
Fs (Hz)	1E5

ATTRIBUTES
Unique, Execute First, Defines: actual_fs

WSUM2 Weighted sum of 2 inputs: WSUM2



CATEGORY: Arithmetic

DESCRIPTION:
Weighted sum of 2 inputs:
y = g1*in1 + g2*in2
-1.0 <= g1,g2 <= +1.0

PARAMETERS:

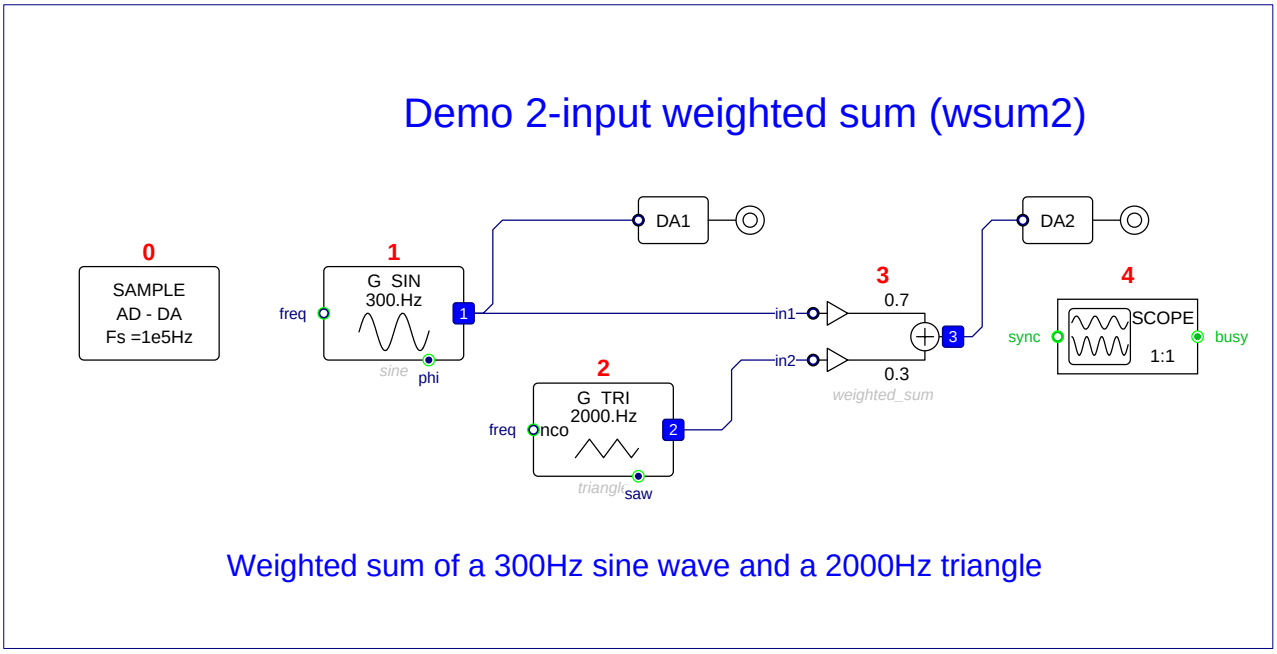
Parameter:	*Default values:*
gain1	1.0
gain2	1.0

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in1	FRACT	WORD	mandatory
name_in2	FRACT	WORD	mandatory

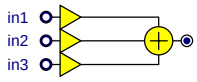
OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



WSUM2 test program

WSUM3 Weighted sum of 3 inputs: WSUM3



CATEGORY: Arithmetic

DESCRIPTION:
Weighted sum of 3 inputs:
y = g1*in1 + g2*in2 + g3*in3
-1 <= g1,g2,g3 <= +1

PARAMETERS:

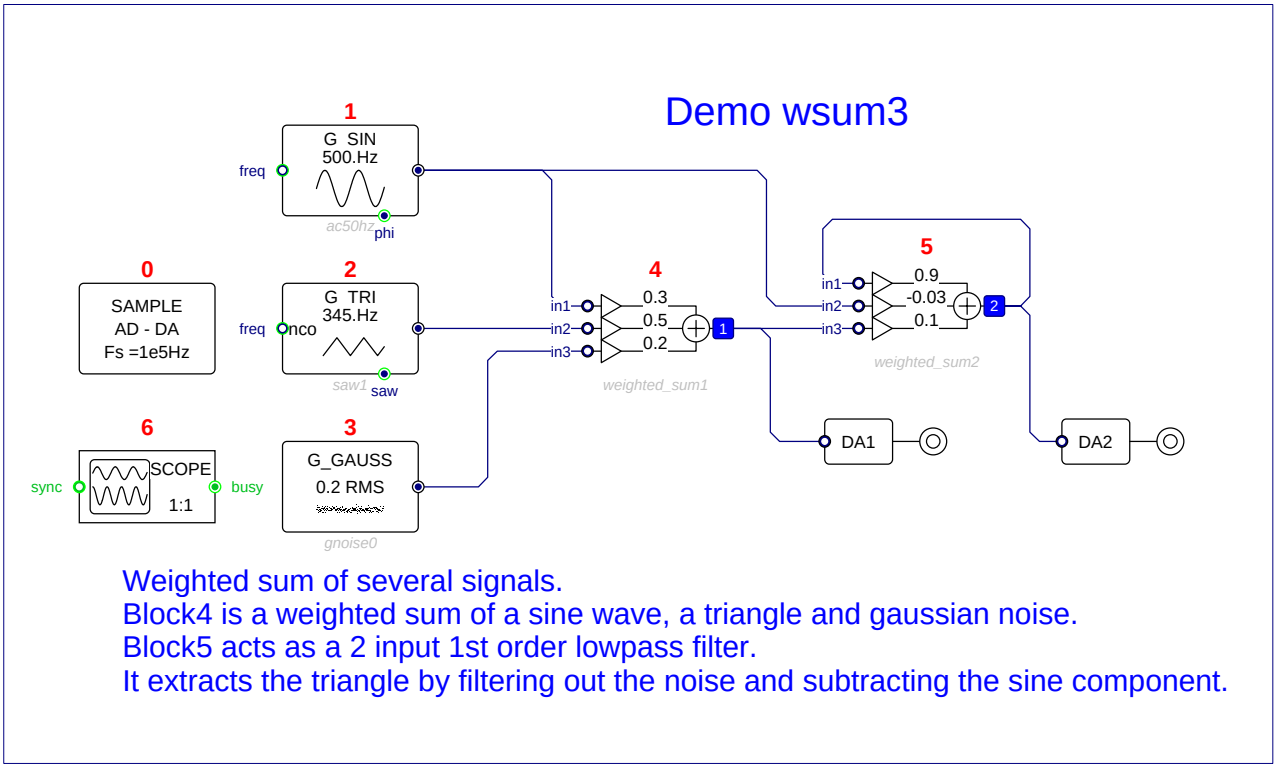
Parameter:	*Default values:*
gain1	1.0
gain2	1.0
gain3	1.0

INPUTS

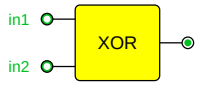
Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in1	FRACT	WORD	mandatory
name_in2	FRACT	WORD	mandatory
name_in3	FRACT	WORD	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	FRACT	WORD	normal



WSUM3 test program



CATEGORY: Logic

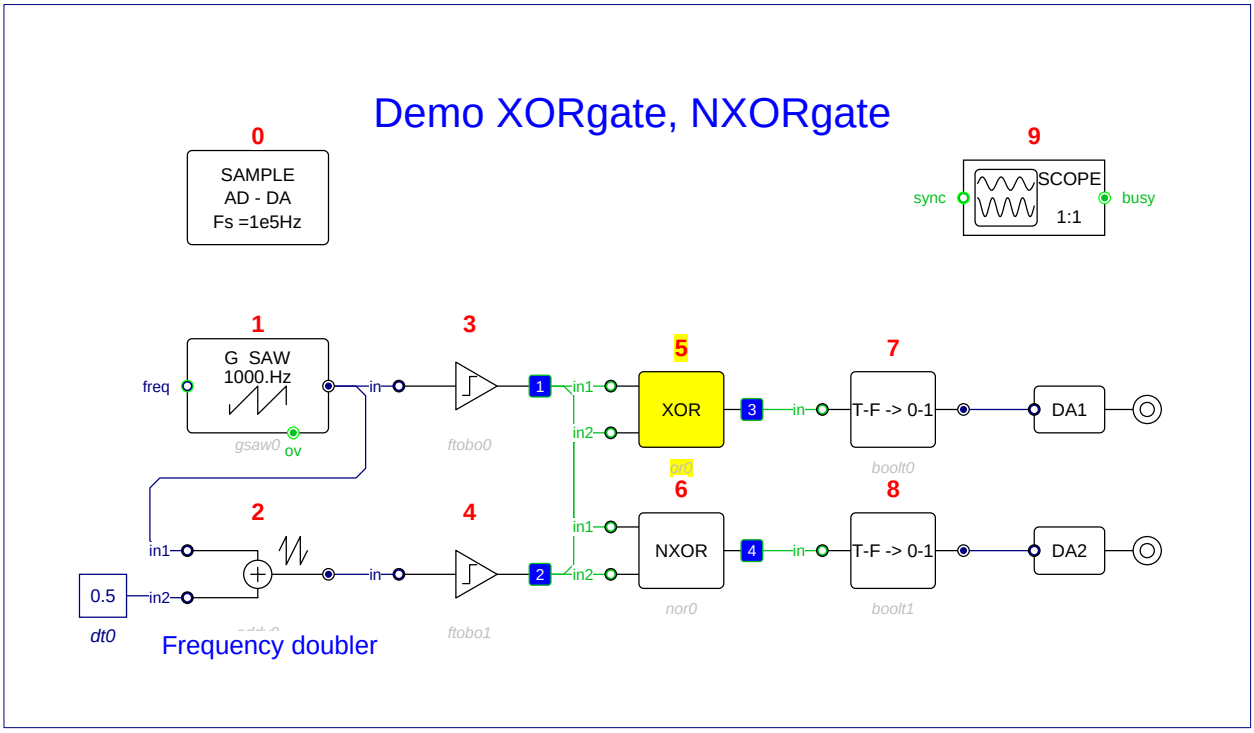
DESCRIPTION:
Logic Exclusive OR
y = (in1 & in2\) v (in1\ & in2)

INPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name_in1	BOOL	BIT	mandatory
name_in2	BOOL	BIT	mandatory

OUTPUTS

Name:	*Data Type:*	*Data Struct:*	*Connection:*
name	BOOL	BIT	normal



XORGATE test program