

FIBULA-G

Block Library

Reference Manual

10/01/2013

FIBULA-G Block Library

ARITHMETIC

ADDS Addition with saturation
ADDV Add with overflow
CADD Complex or mixed Addition
COPY Copy data
CSUB Complex or mixed Subtraction
DIVIDE Fractional division num/den
DOUBLE Gain by 2
FULLSCALE Stretch to [-1..+1]
GAIN Fixed real gain
HALF Gain by 0.5
HALFSUM Half sum of inputs
MADD Multiply and Add
MADD2 Multiply and add 2 inputs
MADDS Multiply, Add, Shift
MUL Real multiplier
MULCC Multiply with conjugate
MULT Complex, mixed, or real multiplier
NEGATE Sign inversion $y = -x$
OFFGAIN Offset and gain:
GOF Gain followed by offset
SHIFT Gain by 2^N
SUBS Subtraction with saturation
SUBV Subtraction modulo +/- 1
WSUM2 Weighted sum of 2 inputs:
WSUM3 Weighted sum of 3 inputs:

AUDIO

BAL Balance
CODEC Audio CODEC
MIDI_BACH MIDI File
MIDI_CHEVAL MIDI File
MIDI_ENTRAINR MIDI File
MIDI_FELICITY MIDI File
IN_L Codec input Left
IN_R Codec input Right
LOGPOT Log potentiometer
OUT_L Codec output Left
OUT_R Codec output Right
AGC Automatic Gain Control
PAN Panoramic
MIDIPLAY MIDI Sequencer
PIANO

REVERB Add reverberation to sound
TRANSPOSE Transpose

CONTINUOUS

G_RAMP Slope generator
G_STEP Step generator
INTEGA Analog Integrator
LP1A 1st order lowpass
LP2A 2nd order lowpass

CONTROL

BOOLTOF Boolean to Fractional
CASE Case condition
CLEAR Clear
CNORM Norm a complex variable
COMPARE Relais function
CPLL Complex PLL
CPLXFREQ Instantaneous frequency
DECIM Decimation
DELAY Real, complex, fixed, variable
DEMUX 1 to 2 Demultiplexer
DERIV Instant Derivator
DERIVATE Derivator FIR Filter
EDGE Generate flags on zero crossing
EXCHG Exchange
FORCEC Force complex
FORCEF Force to value
FRTOBOOL Comparator
INTEG Integrator with input gain
KEY_EVENT Key Press Event
LOOKUP Read data in memory
MAXPOS Position of maximum
MULTIFLAG Set Multiple Flags
MUX 2 input multiplexer
NOP No operation
ONEOF8 Activate selected output
PEAK Get peak value of input
PISO Parallel In Serial Out
REST_SGN Restore Sign
SEQUENCE Sequence generator
SIPO Serial In Parallel Out
STOP Stop processor
SAMPHOLD Sample and Hold
SCAN Scan buffer
SNSD Send string to RS232 port
SWITCH Switch
TRAP Hang here (infinite loop)
UDELAY Unit delay z^{-1}
WAITFLAG Wait and clear flag

ETD410K

AD1 Analog to Digital Converter 1
AD2 Analog to Digital Converter 2
AD3 Analog to Digital Converter 3
AD4 Analog to Digital Converter 4
AD Complex ADC
ADA Wait sample, read ADs, write DAs
BLINKLED Core activity LED
DA1 Digital to Analog Converter 1
DA2 Digital to Analog Converter 2
DA3 Digital to Analog Converter 3
DA4 Digital to Analog Converter 4
DA Complex DAC
CLOCK Change DSP clock frequency
CODEC Audio CODEC

IN_L Codec input Left
IN_R Codec input Right
LED LED
OUT_L Codec output Left
OUT_R Codec output Right
TTL_IN1 Digital Input 1
TTL_IN2 Digital Input 2
TTL_OUT1 Digital output 1
TTL_OUT2 Digital output 2
UART standard UART at 115KBauds

FILTERS

AVERAGE Moving average
DECIM_FIR Decimation FIR
FILTERBANK Bandpass Filter Bank
FIR Finite Impulse Response filter
FIR1 Half sized FIR
FIR2 Bandpass FIR filter
FIRG Gaussian FIR filter
GOERTZEL Goertzel Algorithm
HILBERT Hilbert transform
HP1 First order High-Pass filter
IIR 2nd order IIR filter
IIR2 2nd order recursive filter
IIR6 6th order IIR
IIRCT 1st order Complex IIR filter
IIRT 2nd order IIR Transposed IIR
LMS Auto Adaptive FIR filter
LP1 1st order recursive lowpass filter
LPABS Lowpass of abs value
SLOPELIM Slope limiting filter

FUNCTIONS

ARG Argument of a complex input
CPLX_POW Complex Power
DECIBEL Decibel function
F_ATAN Arc Tangent
F_COS Cosine function $y = \cos(\pi x)$
F_EXP Real exponential function
F_EXPABS Exponential of abs
F_GAUSS Gaussian function
F_SIN Sine function $y = \sin(\pi x)$
F_SINCOS Sine-Cosine function
F_SINUS Sine Function
F_TRI Triangle function
FRDBL
INTERPOL 1D or 2D Table Interpolate
POLY Real Polynomial function
RDTABLE Read interpolate table
SOROOT Square root of input
SQUARE Square of input
TBLR2D 2-D Table read and interpolate

GENERATORS

G_BPR Binary Random Generator
G_CHIRP Chirp Generator
G_GAUSS Gaussian Noise
G_NOISE Random generator
G_PULSE Pulse generator
G_RECT Rectangle generator
G_SAW Sawtooth generator
G_SIN Sine wave generator
G_SINCOS Sine-Cosine Generator
G_SLOPE Triggered Slope Generator
G_SQUARE Square wave generator
G_TRI Triangle generator
OSC High purity sine oscillator
OSQIO Phase quadrature oscillator
TRIGD_PULSE Triggered pulse

INSTRUMENTS

HISTO Buffer switching histogram.
KEY_EVENT Key Press Event
LOGG Data Logger
PLOTTER Slow signal plotter
READMEM Read Memory
SCOPE Multi Channel Scope
SPECAN Spectrum Analyser

INTEGER

CASE Case condition
FITF2I Fract to Integer
FITI2F Integer to Fract
IADDS Integer addition with saturation
IADDV Integer addition modulo 2^{24}
IMUL Integer multiplier
INVT Inverse of an integer
ISUB Integer Subtraction
ITOBOL Comparator, boolean output

INTERRUPT

I_ADA Install AD-DA Interrupt
I_CODEC Install Codec Interrupt
I_TIMER0 Periodic Interrupt

ICC_SEND Send Message
ICC_SND_NMI Send Message
ICC_NMI NMI on message
ICC_SND_IRQ Send Message
ICC_IRQ IRQ on message
ICC_WAIT Wait for Message

LOGIC

ANDGATE Logic AND
BIT Bit Test

BLINKLED Core activity LED
BOOLTOF Boolean to Fractional
CHANGED Check if input has changed
FLAGSET Set bool variable to TRUE
FLAGCLR Set bool variable to FALSE
FLAGTOG Toggle boolean variable
FRCOMP Comparator
FRTOBOOL Comparator
INTCOMP Integer Comparator
INTTOBOOL Comparator of Integers
IQ_DECODER Incremental decoder
LED LED

NANDGATE Logic NAND
NORGATE Logic NOR function
NOTGATE
NXORGATE Logic NXOR function
ORGATE Logic OR function
PISO Parallel In Serial Out
SIPO Serial In Parallel Out
RS_FLOPFLIP RS flip flop
TTL_IN1 Digital Input 1
TTL_IN2 Digital Input 2
TTL_OUT1 Digital output 1
TTL_OUT2 Digital output 2
XORGATE Logic Exclusive OR

MATRIX

ARRAYMUL Array Multiply
COMPRESS Suppress small values
DCT Discrete Cosine Transform
DFTSWAP Swap Input Buffer halves
EXPAND Restore compressed data
IDCT Inverse DCT
DFT Discrete Fourier Transform
IDFT Inverse DFT
FFT Fast Fourier Transform
IFFT Inverse FFT
FFTCOEFF FFT coefficient table
FLOWTAPK Flow to Packets
FLOWTOVECT Data flow to vector.
GETOVERLAP Get overlap buffer
ISNOTNULL Test bool vector
MAKE_ERR Inject errors within vector
MATEOR XOR between matrices
MATMUL Matrix product
MATMULB Boolean Matrix product
MATSUM Sum of matrices

PAKTOFLOW Packets to Flow

VECTTOFLOW Vector to dataflow
VECT_POW Vector Power
V_CPLXPOW Vector Complex Mean Power
WINDOW Weighting Window
WR_BUFFER Write to Circular Buffer
ZEROPAD Zero Padding
MUSIC
MIDI_SEQ MIDI Sequencer
MIDI_CMD Midi Command interface
GENERIC_INST
GUITAR
HARP
ORGAN
MIDI_BACH MIDI File
MIDI_CHEVAL MIDI File
MIDI_ENTRAINR MIDI File
MIDI_FELICITY MIDI File
REQUIEM MIDI File
PAVANE MIDI File
TOLLITE_HOSTIAS MIDI File
DEBUSSY1 MIDI File
DEBUSSY2 MIDI File
COMEDIENS MIDI File
CONTRAPUNCT MIDI File
TOURDION MIDI File
ALTA_TRINITA MIDI File
NON LINEAR
COMPARE Relais function
MAGN Magnitude of a real or complex
POS Diode function
QUANT Quantize data to n bits
SGN Sign function
F_STEP Step function

SDR

ACCUM Accumulate random signals
AVERAGE Moving average
CORREL Cross correlation
G_NOISE Random generator
G_GAUSS Gaussian Noise
HISTO Buffer switching histogram.
STAT
ACCUM Accumulate random signals
AVERAGE Moving average
CORREL Cross correlation
G_NOISE Random generator
G_GAUSS Gaussian Noise
HISTO Buffer switching histogram.

STRING

FRTOHEX Fract to Hex-String

FRTOSTR Fract to String
INTTOSTR Integer to String
KBD Get ASCII from keyboard
WORDTOHEX Word to hexadecimal
WORDTOBIN Word to Binary
SEND_STR Send string

TABLES

SINC Sinc Table
SINUS Sine Table
SINCOS Sine-cosine Table
WINDOW Weighting Window

TELECOM

AGC Automatic Gain Control
BAD_CHAN
CHAN_ECHO
DELTAPHI Argument difference
DEMFSK FSK Demodulator
DPHI Phase differentiator
ENCODE GF(2) encoder
FADING Simulate fading channel
FIR_RC Raised Cosine FIR
FIR_RRC Root Raised Cosine
G_CLK Clock Generator
G_ASCII Triggered ASCII source
G_RNDSYM Random symbol generator
ASCTOSYM ASCII to symbols
BL_RANDOM Band limited random
COD_R3D 1-3 Repetition coder
DEC_R3D 3-1 Repetition decoder
DIFFCOD Differential coder
DIFFDEC Differential decoder
DIFFSCOD Differential sign coder
DIFFSECD Differential sign decoder
MAP Map symbol to complex
MODUL I-Q modulator
MODFSK FSK Modulator
MODMSK MSK Modulator
MUTINFO Mutual Information
MUXF I-domain multiplexer
CHANNEL Channel simulation
CHAN_RINGING Channel w. resonance
COSTAS COSTAS loop
DEMODO QAM demodulator
UNMAP Complex to symbol
RCPUULSE Raised Cosine Pulse shaper
RXCK Clock restoration
SCRAMBLE N-bit scrambler
SIGMAPHI Phase accumulator
SYMTOASC Symbols to ASCII
SNDC Send char to serial port
RX_AMI AMI decoder
RX_MAN Manchester line decoder
RX_MAND D-Manchester decoder
RX_MLT3 MLT3 line decoder
RX_NRZ NRZ decoder
RX_NRZI NRZI line decoder
RX_RZ Return to Zero line decoder
TX_AMI AMI Coder
TX_MAN Manchester line coder
TX_MAND D-Manchester Coder
TX_MLT3 MLT3 line coder
TX_NRZ Non Return to Zero line coder
TX_NRZI NRZI line coder
TX_RZ Return to Zero line coder
UNSCRAMBLE Unscrambler

TIMING

ADA Wait sample, read ADs, write DAs
BLINKLED Core activity LED
FS_TIMER Waits for sample time
CLOCK Change DSP clock frequency
COUNTER Event counter
HWTIMER HW Periodic Timer
LED LED
SWTEMPO Time waiting tempo
TIC Start H/W Cycle Counter
TOC Stop Chrono
TIMERF Frequency Timer
TIMERP Periodic Timer
TIMERS One shoot Timer.
TRIGD_PULSE Triggered pulse
WAITFLAG Wait and clear flag

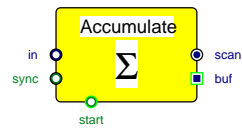
UNCLASSIFIED

ALAWTOLIN A-Law to linear conversion
AVEVERUM MIDI File
COD_STEREO
DEC_STEREO
DEM_FSK FSK Demodulator
F_TRIABS Abs value of Triangle function
FP_ABS Floating Point absolute value
FP_ADD Floating Point Addition
FP_CMP Floating Point Compare
FP_DIV Floating Point division num/den
FP_MAC Floating Point MAC
FP_MPY Floating Point multiply
FP_NEG Floating Point Negate
FP_SCALE Floating Point scaling
FP_SORT Square root of input
FP_SUB Floating Point subtraction
FP_WMAG2C Float $y = x_0 + g_1 x_1 + g_2 x_2$
FP_WSUM2 Float weighted sum
EPTOPR Float to Fract
FRTOFP Fract to Float
FRTOBOOL Comparator
IXOR Exclusive OR
LINTOALAW Linear to A-Law conversion
LSHIFT Logic N bit Shift
LULLY MIDI File
MOD_FSK FSK Modulator
RONDON MIDI File
SDRAM Install SDRAM PortA interface
SHIFTV Logic Shift
STEREO_COD
STEREO_DEC
VIOL
WAITKEY Wait until key pressed

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:

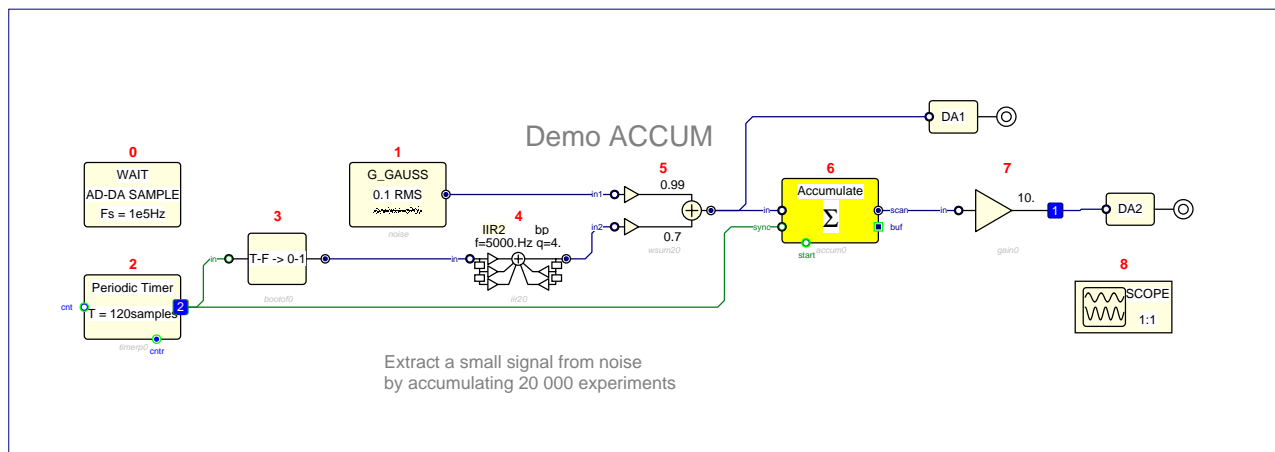
Parameter:	Default values:
Points	500
Number of adds	10000

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

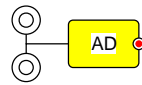


ACCUM test program

AD

Complex ADC

AD



CATEGORY: ETD410K

DESCRIPTION:
Complex ADC
AD1=Real AD2=Imag

OUTPUTS

Name:
name

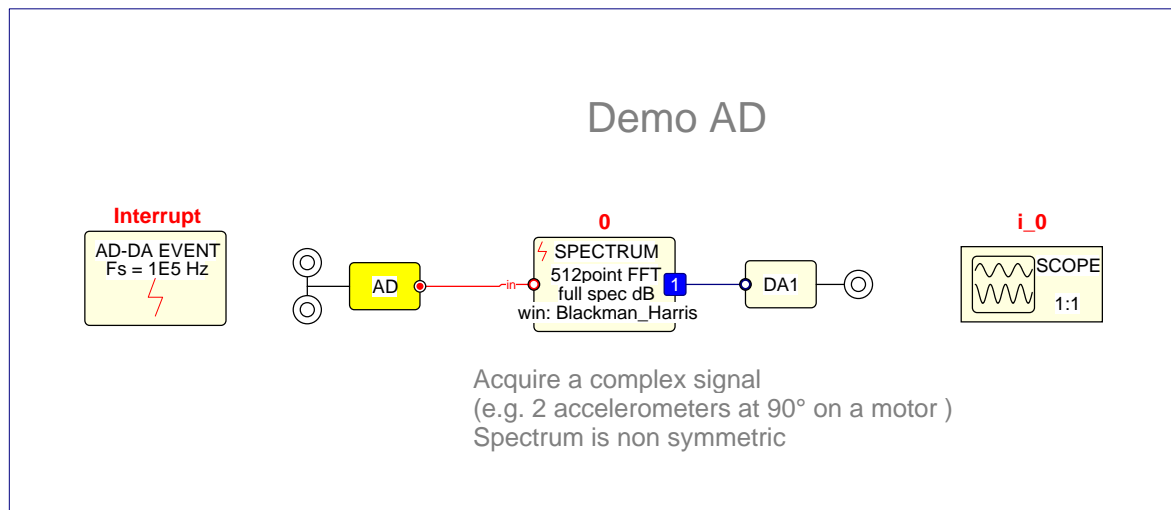
Data Type:
COMPLEX

Data Struct:
WORD

Connection:
normal

ATTRIBUTES

Non executable, Unique,



AD test program

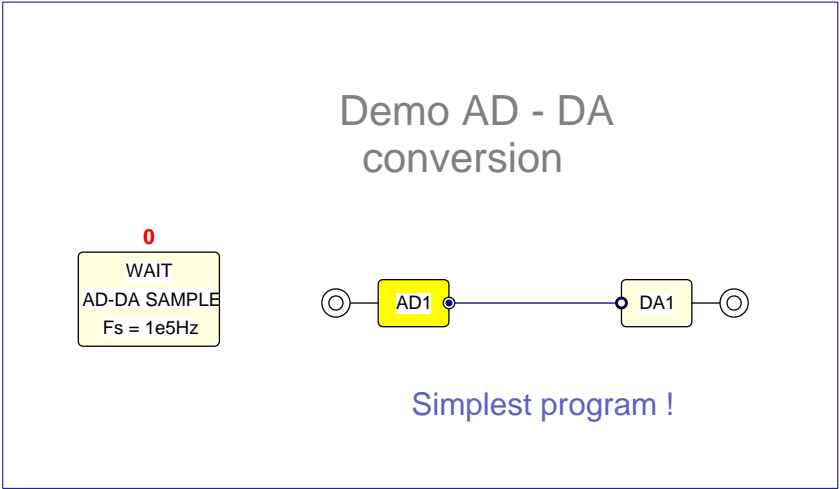


CATEGORY: ETD410K

DESCRIPTION:
Analog to Digital Converter 1

OUTPUTS			
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name	FRACT	WORD	normal

ATTRIBUTES
Non executable, Unique,



AD1 test program



CATEGORY: ETD410K

DESCRIPTION:
Analog to Digital Converter 2

OUTPUTS

Name:
name

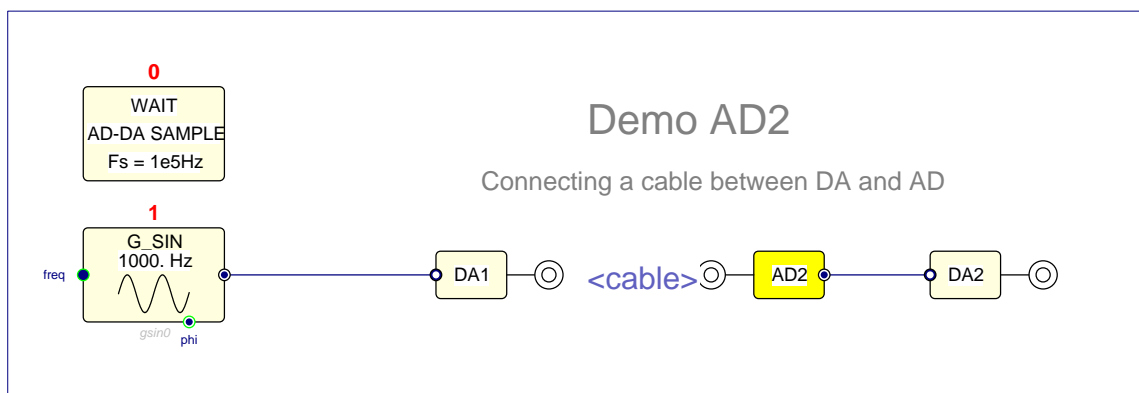
Data Type:
FRACT

Data Struct:
WORD

Connection:
normal

ATTRIBUTES

Non executable, Unique,



AD2 test program



CATEGORY: ETD410K

DESCRIPTION:
Analog to Digital Converter 3

OUTPUTS

<i>Name:</i> name	<i>Data Type:</i> FRACT	<i>Data Struct:</i> WORD	<i>Connection:</i> normal
----------------------	----------------------------	-----------------------------	------------------------------

ATTRIBUTES

Non executable, Unique,



CATEGORY: ETD410K

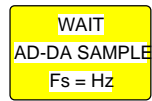
DESCRIPTION:
Analog to Digital Converter 4

OUTPUTS

<i>Name:</i> name	<i>Data Type:</i> FRACT	<i>Data Struct:</i> WORD	<i>Connection:</i> normal
----------------------	----------------------------	-----------------------------	------------------------------

ATTRIBUTES

Non executable, Unique,



CATEGORY: ETD410K

DESCRIPTION:

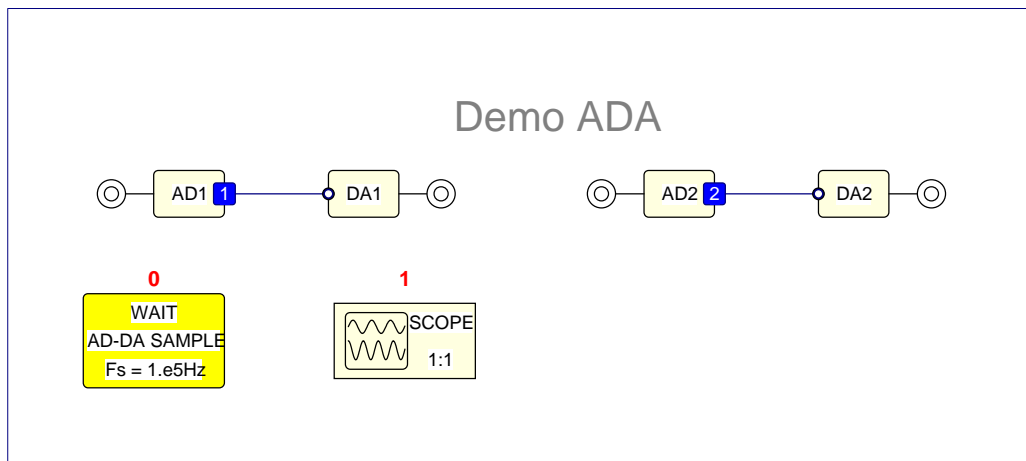
Wait sample, read ADs, write DAs
Defines actual_fs

PARAMETERS:

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

ATTRIBUTES

Unique, Execute First, Defines: actual_fs

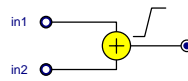


ADA test program

ADDS

Addition with saturation

ADDS



CATEGORY: ARITHMETIC

DESCRIPTION:
Addition with saturation

INPUTS

Name:
name_in1
name_in2

Data Type:
FRACT
FRACT

Data Struct:
WORD
WORD

Connection:
mandatory
mandatory

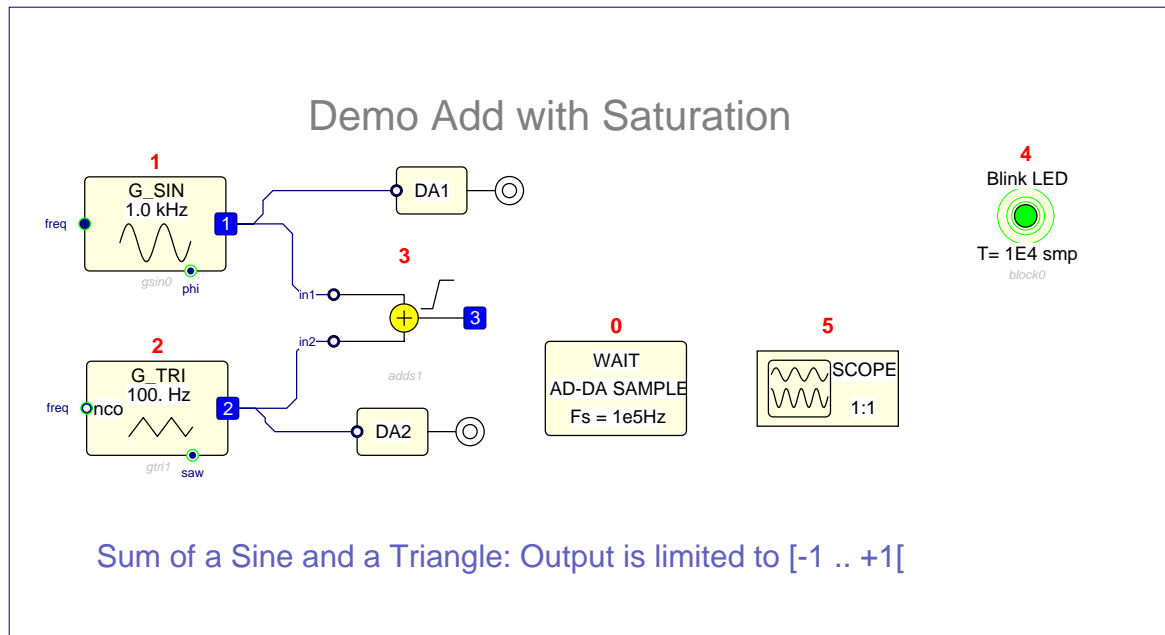
OUTPUTS

Name:
name

Data Type:
FRACT

Data Struct:
WORD

Connection:
normal

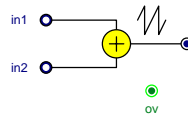


ADDS test program

ADDV

Add with overflow

ADDV



CATEGORY: ARITHMETIC

DESCRIPTION:

Add with overflow

$y = in1 + in2$;

if $y \geq 1$ then $y = y - 2$; if $y < -1$ then $y = y + 2$;

INPUTS

Name:

name_in1
name_in2

Data Type:

FRACT
FRACT

Data Struct:

WORD
WORD

Connection:

mandatory
mandatory

OUTPUTS

Name:

name
name_ov

Data Type:

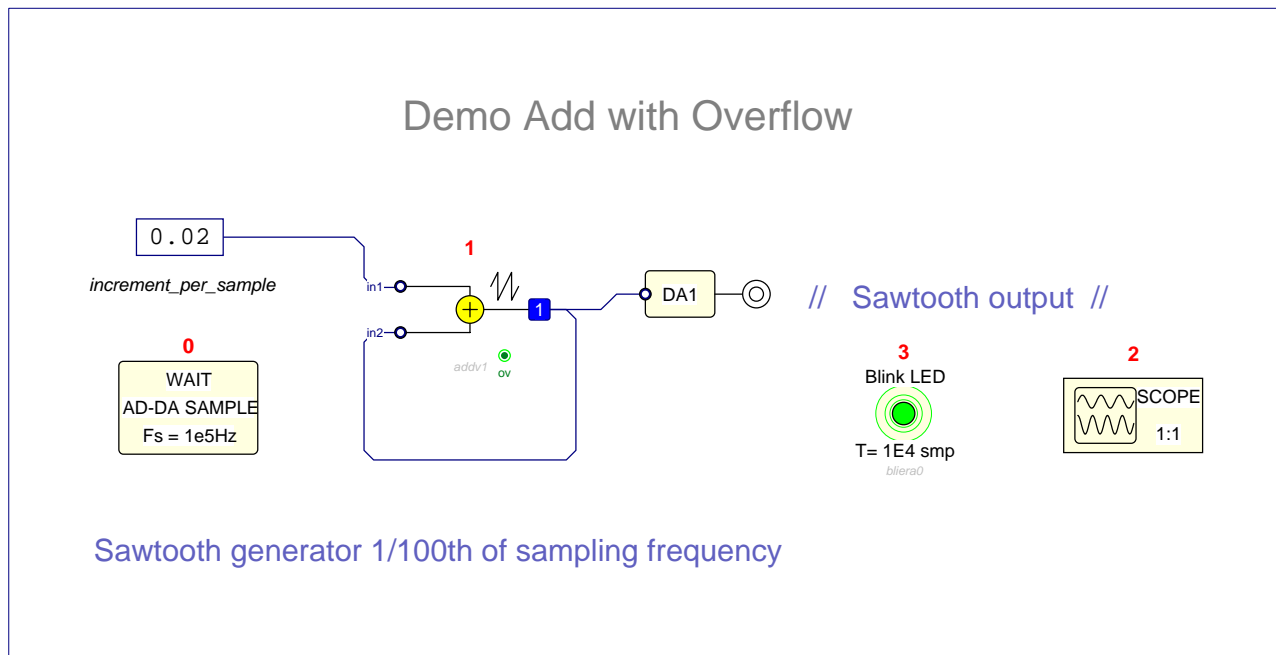
FRACT
BOOL

Data Struct:

WORD
BIT

Connection:

normal
optional

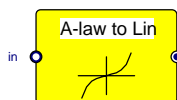


ADDV test program

AGC

ALAWTOLIN A-Law to linear conversion

ALAWTOLIN



DESCRIPTION:
A-Law to linear conversion

INPUTS
Name:
name_in

Data Type:
FRACT

Data Struct:
WORD

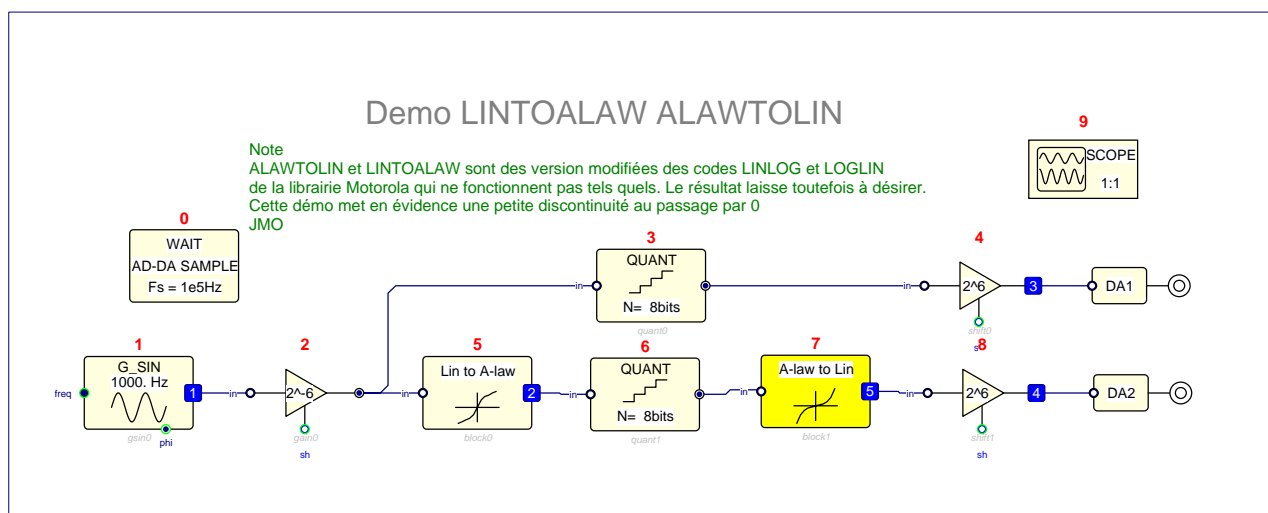
Connection:
mandatory

OUTPUTS
Name:
name

Data Type:
FRACT

Data Struct:
WORD

Connection:
normal



ALAWTOLIN test program



CATEGORY: MUSIC

DESCRIPTION:

MIDI File
Transcribed in asm format

OUTPUTS

<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name	INTEGER	Matrix of WORD	normal

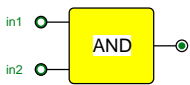
ATTRIBUTES

Non executable, Unique, Data Table

ANDGATE

Logic AND

ANDGATE



CATEGORY: LOGIC

DESCRIPTION:

Logic AND
 $\text{BOOL } y = \text{in1} \ \& \ \text{in2}$

INPUTS

Name:
name_in1
name_in2

Data Type:
BOOL
BOOL

Data Struct:
BIT
BIT

Connection:
mandatory
mandatory

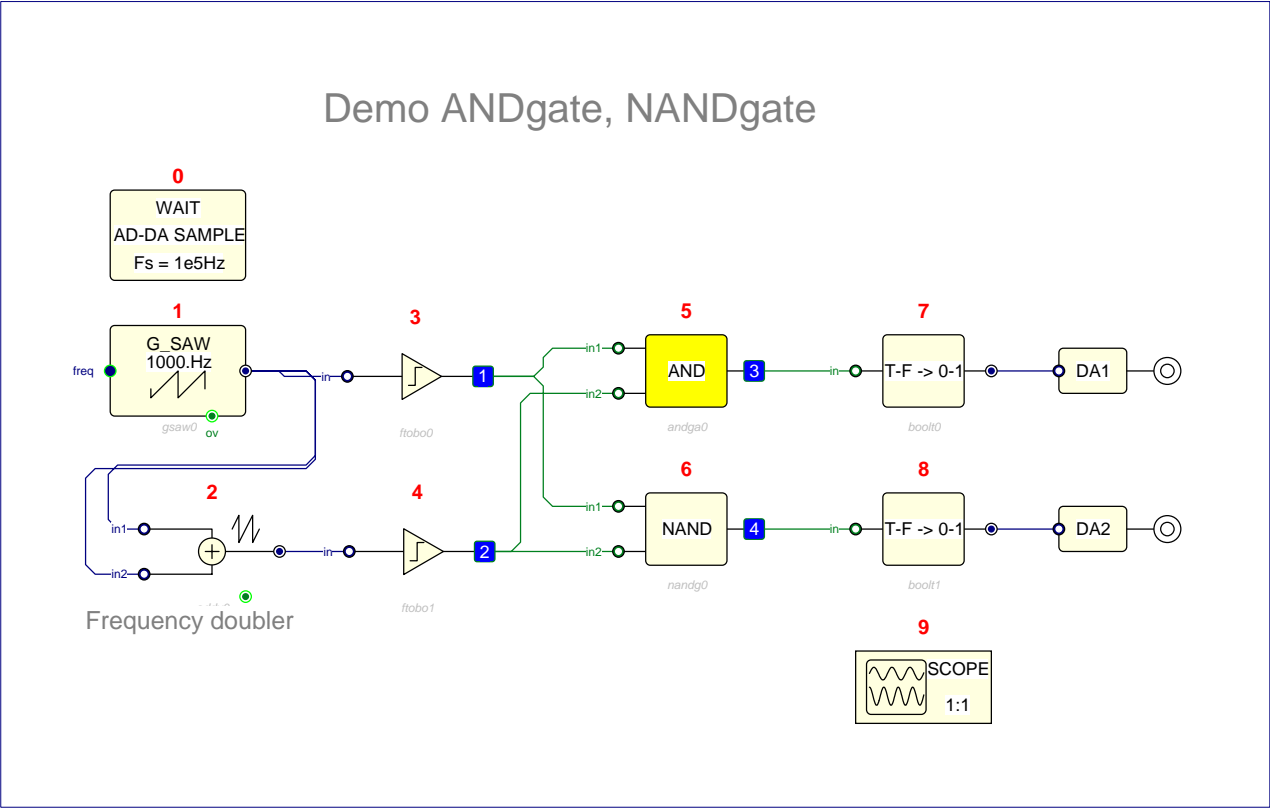
OUTPUTS

Name:
name

Data Type:
BOOL

Data Struct:
BIT

Connection:
normal

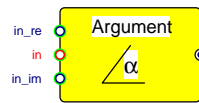


ANDGATE test program

ARG

Argument of a complex input

ARG



CATEGORY: FUNCTIONS

DESCRIPTION:

Argument of a complex input
 $y = 1/\pi * \arctan2(\text{Im}(x), \text{Re}(x))$

INPUTS

Name:
 name_in
 name_in_re
 name_in_im

Data Type:
 COMPLEX
 FRACT
 FRACT

Data Struct:
 WORD
 WORD
 WORD

Connection:
 optional
 optional
 optional

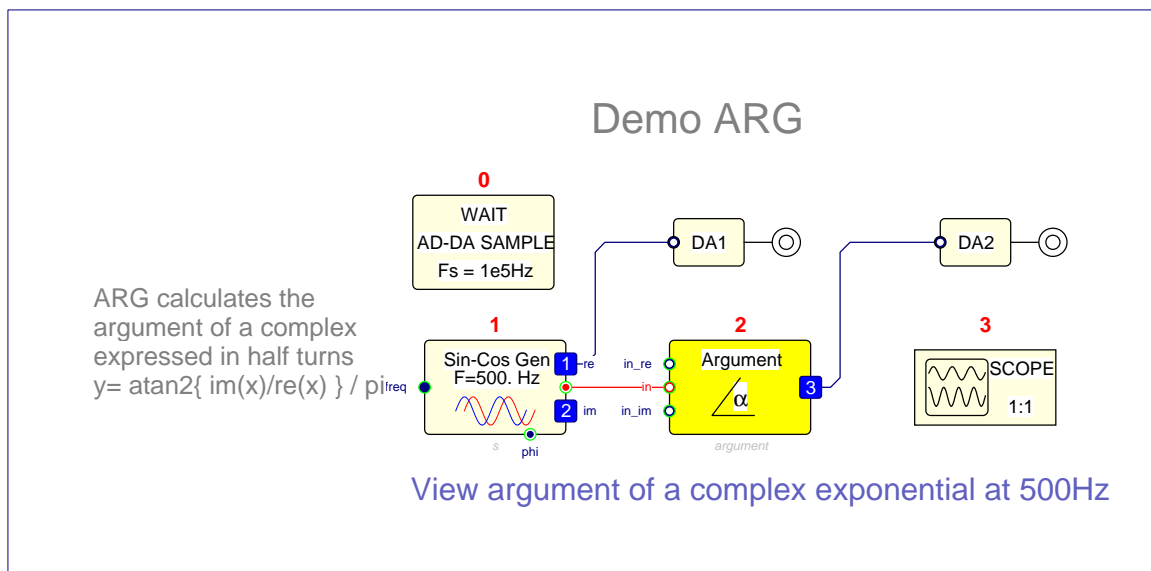
OUTPUTS

Name:
 name

Data Type:
 FRACT

Data Struct:
 WORD

Connection:
 normal

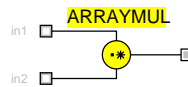


ARG test program

ARRAYMUL

Array Multiply

ARRAYMUL



CATEGORY: MATRIX

DESCRIPTION:

Array Multiply
 $Out(i) = in1(i) * in2(i)$

INPUTS

Name:
 name_in1
 name_in2

Data Type:
 defined by cn
 defined by cn

Data Struct:
 Matrix of
 Matrix of

Connection:
 mandatory
 mandatory

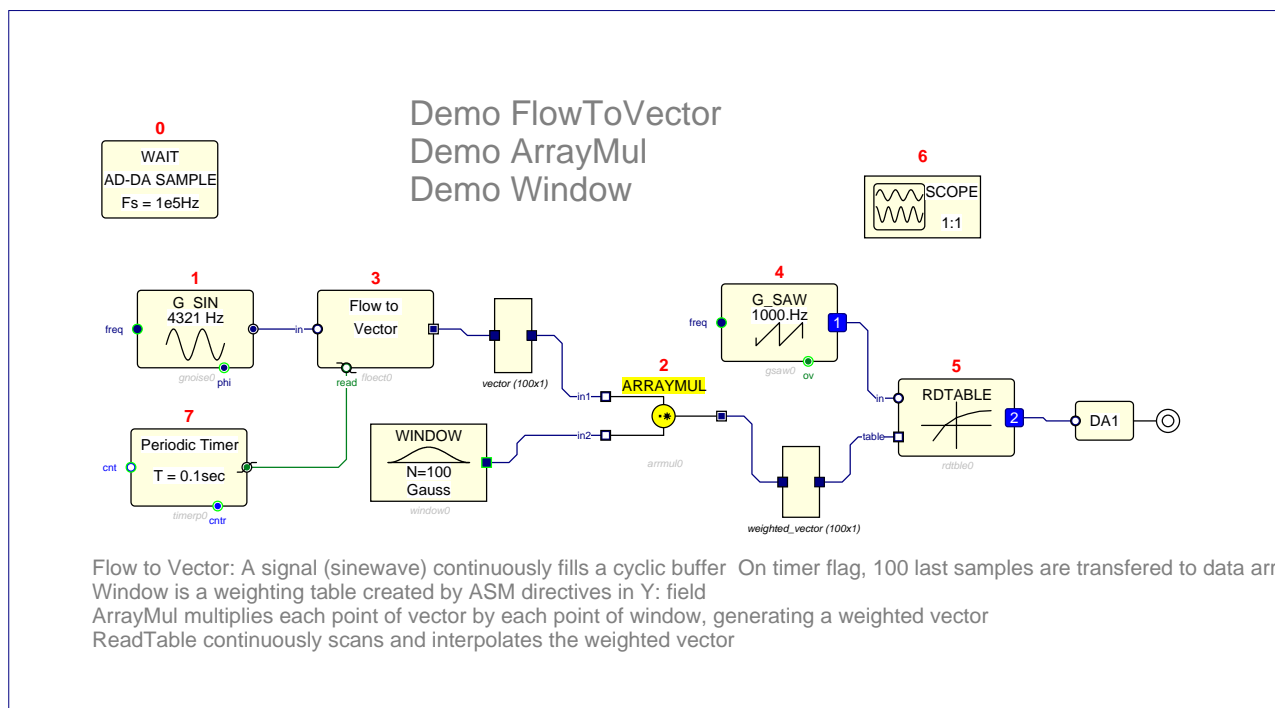
OUTPUTS

Name:
 name

Data Type:
 defined by cn

Data Struct:
 Matrix of

Connection:
 normal

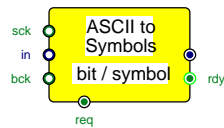


ARRAYMUL test program

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:

Bits per symbol

Default values:

1

INPUTS

Name:

name_in

name_bck

name_sck

Data Type:

FRACT

BOOL

BOOL

Data Struct:

WORD

BIT

BIT

Connection:

mandatory

mandatory

mandatory

OUTPUTS

Name:

name

name_rdy

name_req

Data Type:

FRACT

BOOL

BOOL

Data Struct:

WORD

BIT

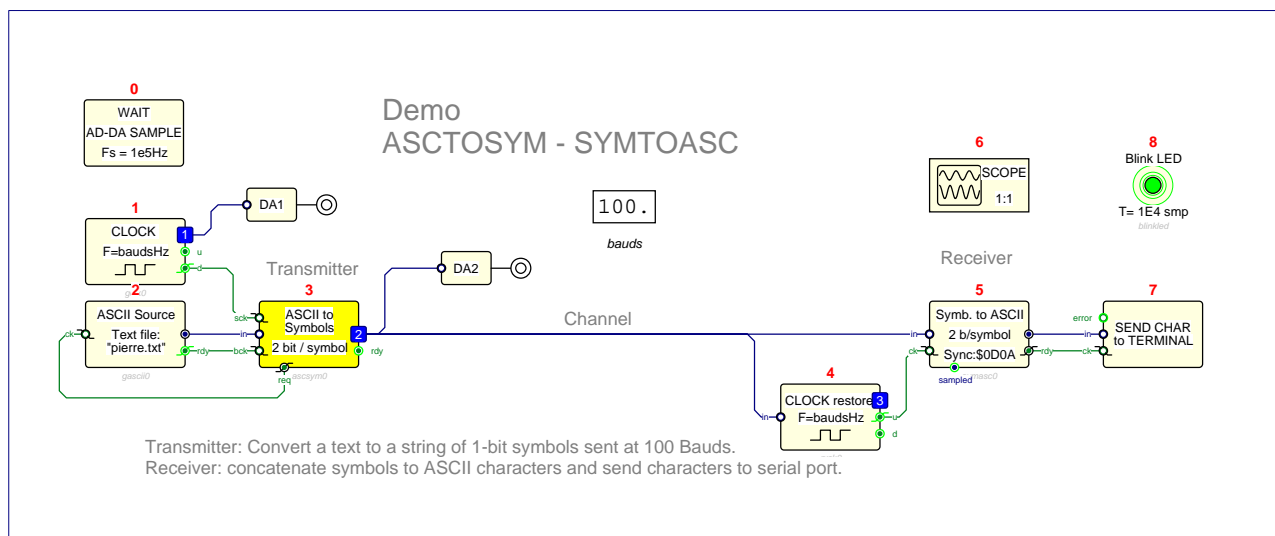
BIT

Connection:

normal

optional

normal

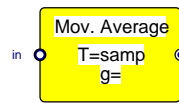


ASCTOSYM test program

AVERAGE

Moving average

AVERAGE



CATEGORY: FILTERS

DESCRIPTION:

Moving average

$$y(k) = (g/n) \sum(x_i), i = [k-n+1 \dots k]$$

PARAMETERS:

Parameter:

gain

time

Unit

Default values:

1.0

0.02

sec,samp

INPUTS

Name:

name_in

Data Type:

FRACT

Data Struct:

WORD

Connection:

mandatory

OUTPUTS

Name:

name

Data Type:

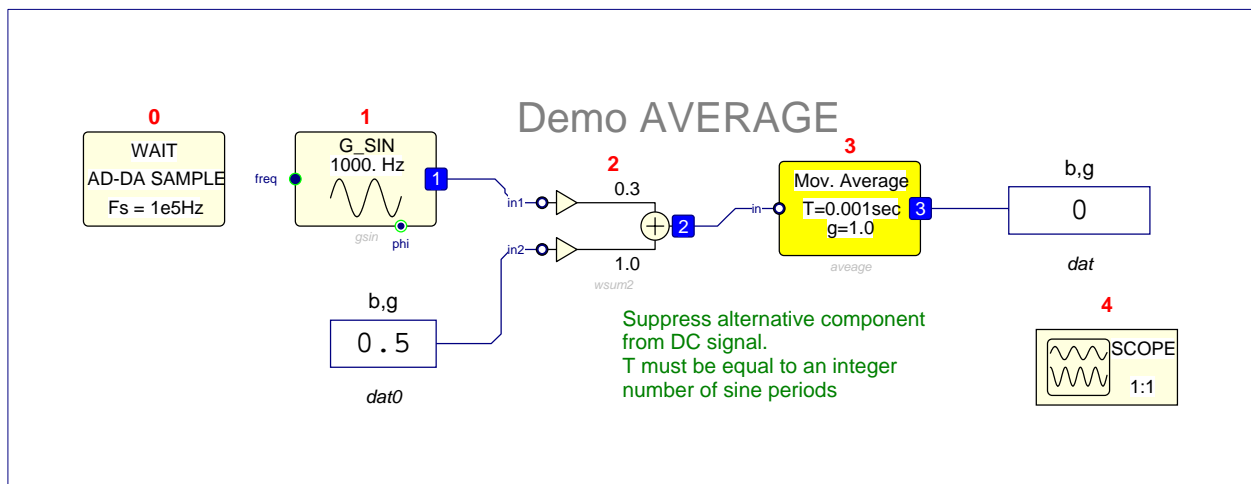
FRACT

Data Struct:

WORD

Connection:

normal



AVERAGE test program



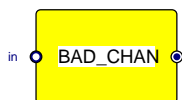
DESCRIPTION:
MIDI File
Transcripted in asm format

OUTPUTS			
<i>Name:</i> name	<i>Data Type:</i> INTEGER	<i>Data Struct:</i> Matrix of WORD	<i>Connection:</i> normal

ATTRIBUTES
Non executable, Unique, Data Table

BAD_CHAN

BAD_CHAN



CATEGORY: TELECOM

INPUTS

Name:
name_in

Data Type:
FRACT

Data Struct:
WORD

Connection:
mandatory

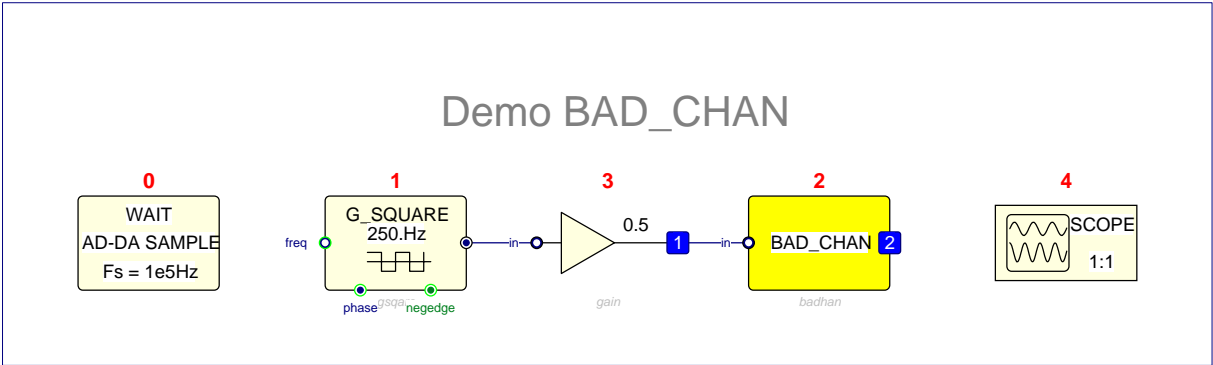
OUTPUTS

Name:
name

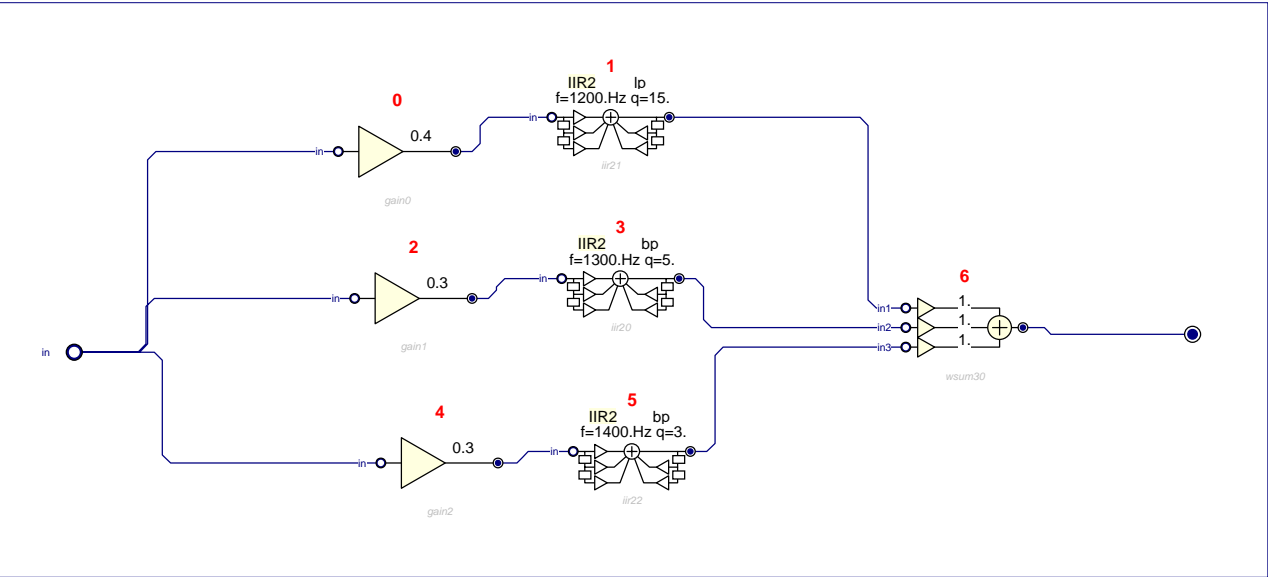
Data Type:
FRACT

Data Struct:
WORD

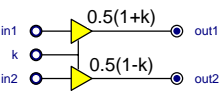
Connection:
normal



BAD_CHAN test program



BAD_CHAN internal schema



CATEGORY: AUDIO

DESCRIPTION:

Balance
 $out1 = in1 \cdot (1+k)/2$
 $out2 = in2 \cdot (1-k)/2$

INPUTS

Name:
name_in1
name_k
name_in2

Data Type:
FRACT
FRACT
FRACT

Data Struct:
WORD
WORD
WORD

Connection:
mandatory
mandatory
mandatory

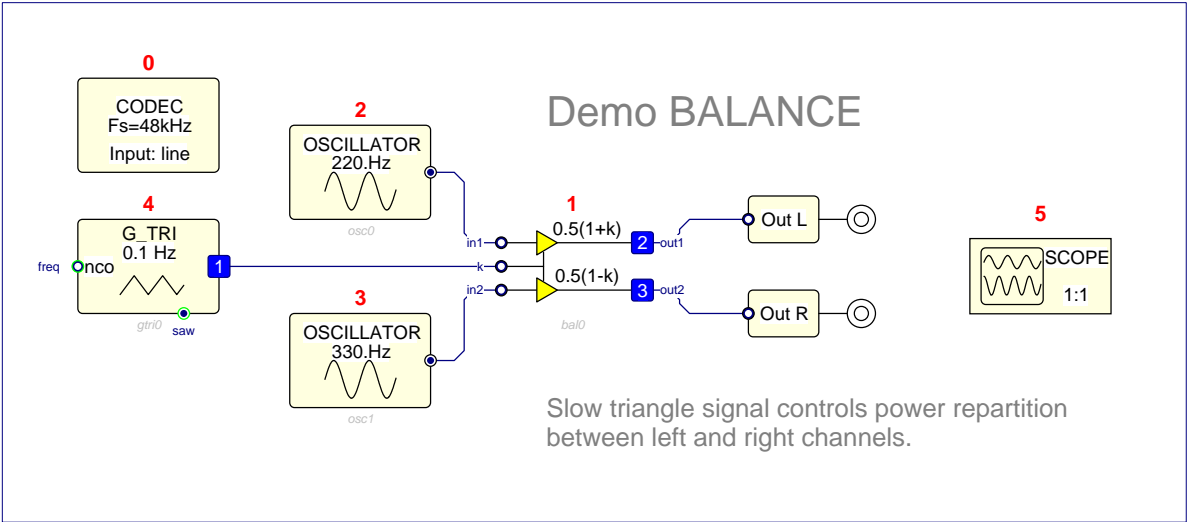
OUTPUTS

Name:
name_out1
name_out2

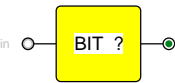
Data Type:
FRACT
FRACT

Data Struct:
WORD
WORD

Connection:
normal
normal



BAL test program



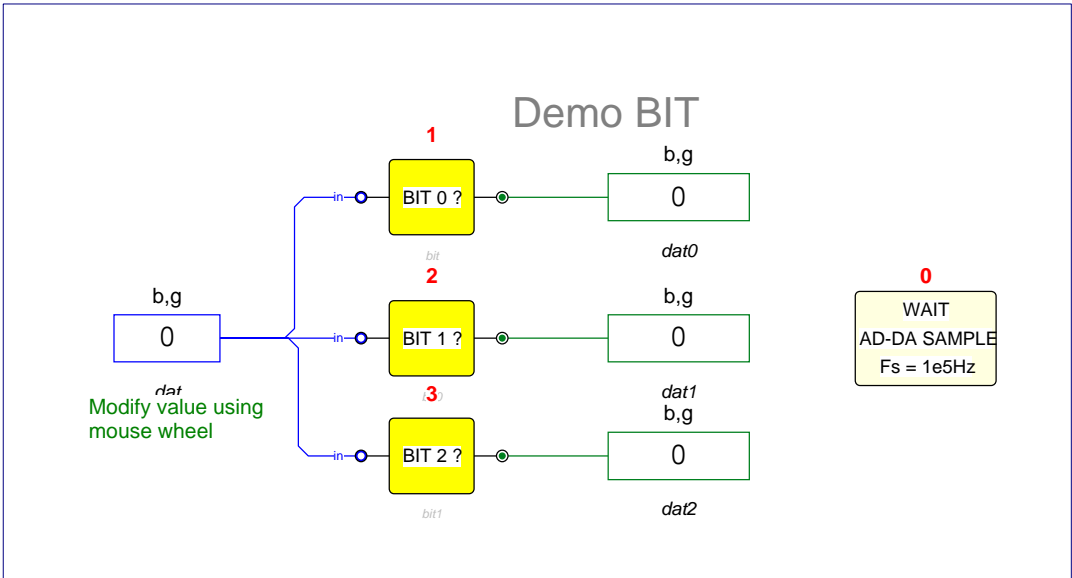
CATEGORY: LOGIC

DESCRIPTION:
Bit Test

PARAMETERS:
Parameter:
Bit Nr
Default values:
0

INPUTS			
<i>Name:</i> name_in	<i>Data Type:</i> defined by cn	<i>Data Struct:</i>	<i>Connection:</i> mandatory

OUTPUTS			
<i>Name:</i> name	<i>Data Type:</i> BOOL	<i>Data Struct:</i> BIT	<i>Connection:</i> normal

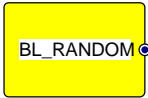


BIT test program

BL_RANDOM

Band limited random

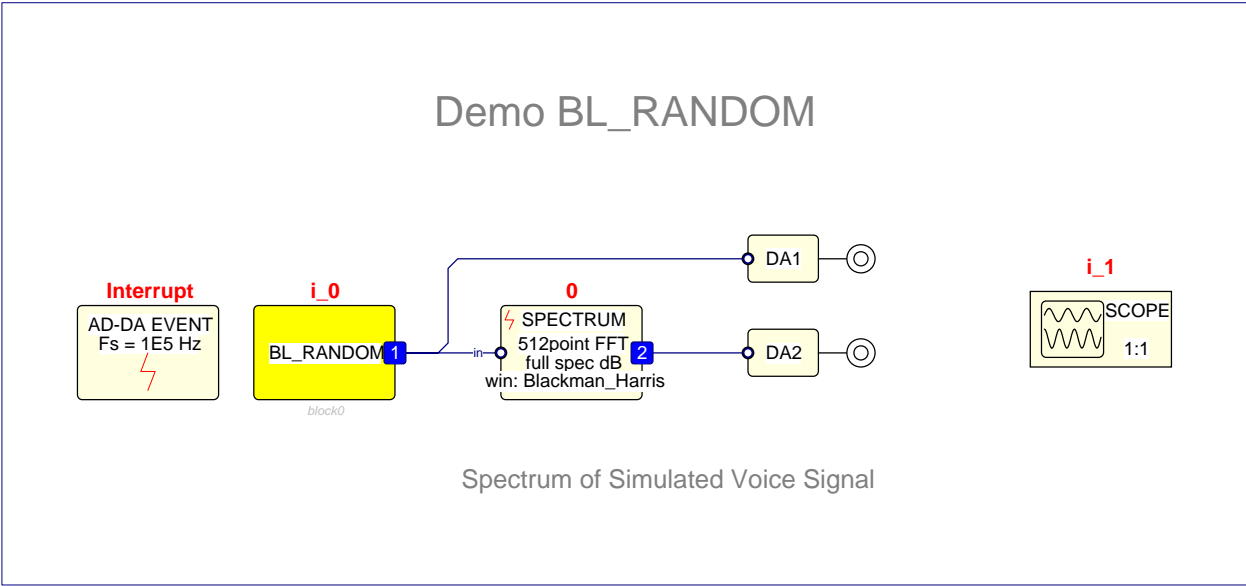
BL_RANDOM



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

BLINKLED

Core activity LED

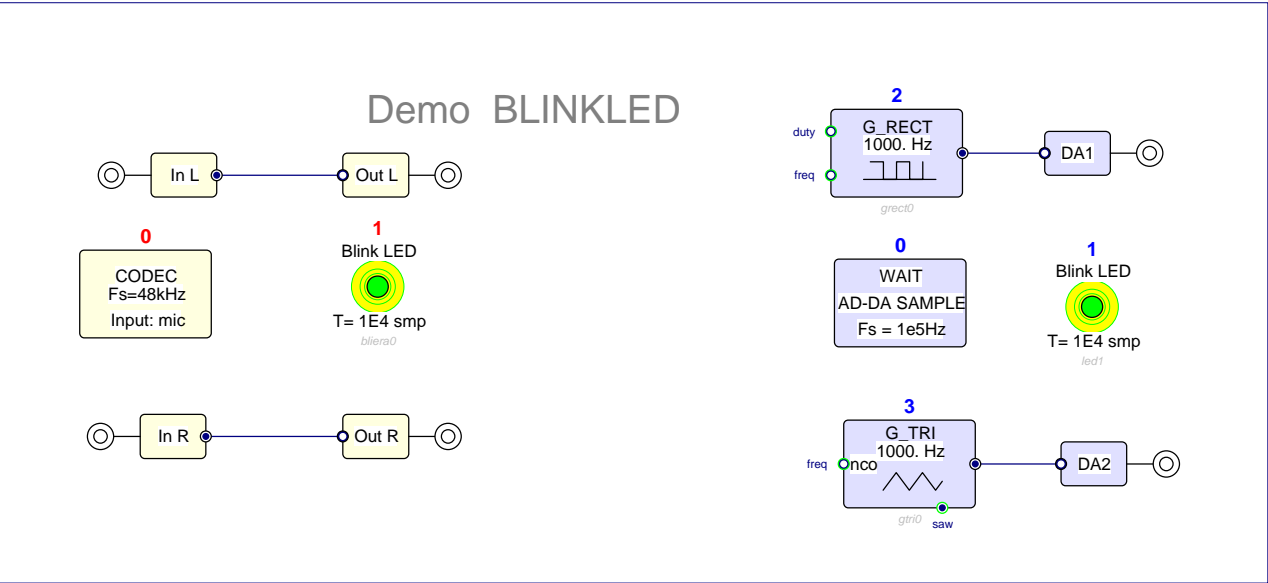
BLINKLED



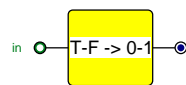
CATEGORY: ETD410K

DESCRIPTION:
Core activity LED
Twinkle core LED at given period

PARAMETERS:
Parameter: *Default values:*
Period (samples) 1E4



BLINKLED test program



CATEGORY: CONTROL

DESCRIPTION:
Boolean to Fractional
T - F --> 1.0 - 0.0

INPUTS
Name:
name_in

Data Type:
BOOL

Data Struct:
BIT

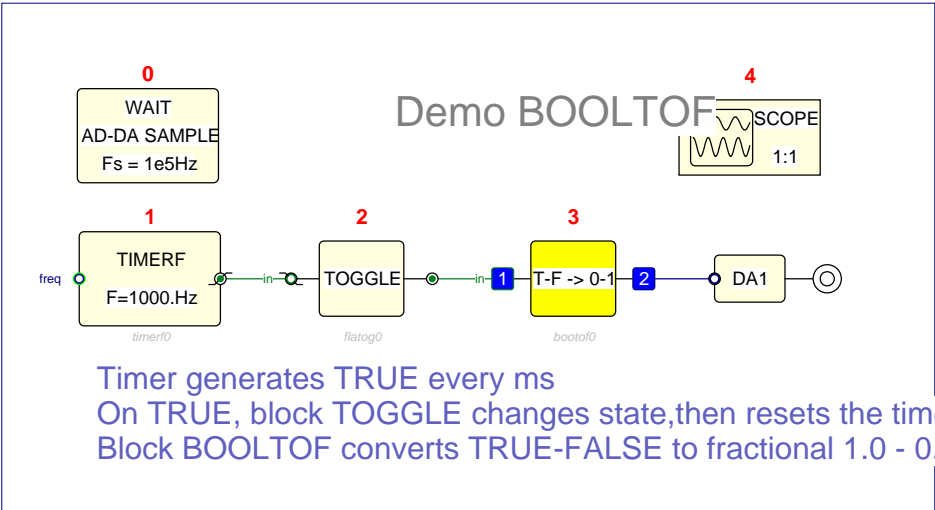
Connection:
mandatory

OUTPUTS
Name:
name

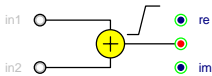
Data Type:
FRACT

Data Struct:
WORD

Connection:
normal



BOOLTOF test program



CATEGORY: ARITHMETIC

DESCRIPTION:
Complex or mixed Addition
with saturation

INPUTS

Name:
name_in1
name_in2

Data Type:
defined by cn
defined by cn

Data Struct:

Connection:
mandatory
mandatory

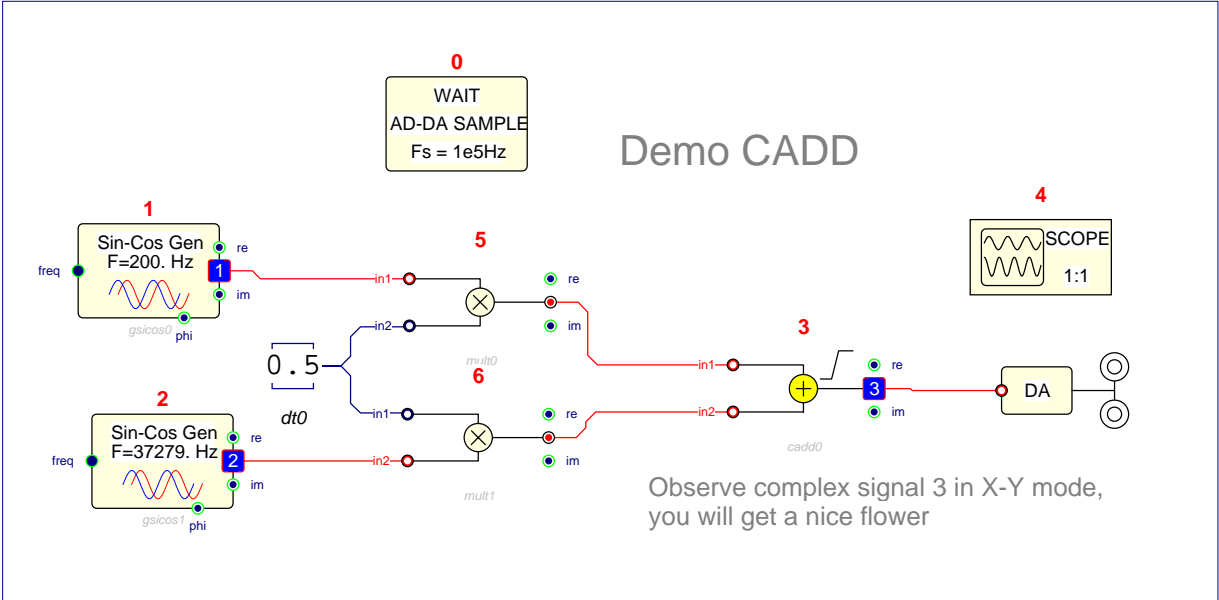
OUTPUTS

Name:
name
name_re
name_im

Data Type:
COMPLEX
FRACT
FRACT

Data Struct:
WORD
WORD
WORD

Connection:
optional
optional
optional

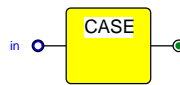


CADD test program

CASE

Case condition

CASE



CATEGORY: CONTROL

DESCRIPTION:

Case condition

Output is true if (in = Case_value), false otherwise

PARAMETERS:

Parameter:
CASE value

Default values:
\$20

INPUTS

Name:
name_in

Data Type:
INTEGER

Data Struct:
WORD

Connection:
mandatory

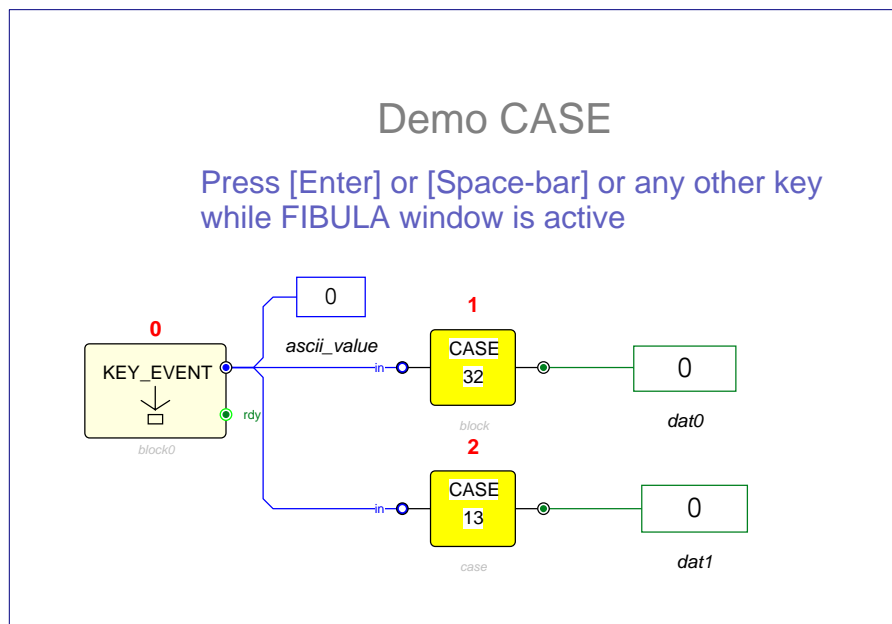
OUTPUTS

Name:
name

Data Type:
BOOL

Data Struct:
BIT

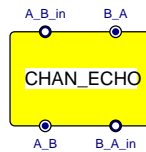
Connection:
normal



CASE test program

CHAN_ECHO

CHAN_ECHO



CATEGORY: TELECOM

INPUTS

Name:

name_A_B_in
name_B_A_in

Data Type:

FRACT
FRACT

Data Struct:

WORD
WORD

Connection:

mandatory
mandatory

OUTPUTS

Name:

name_B_A
name_A_B

Data Type:

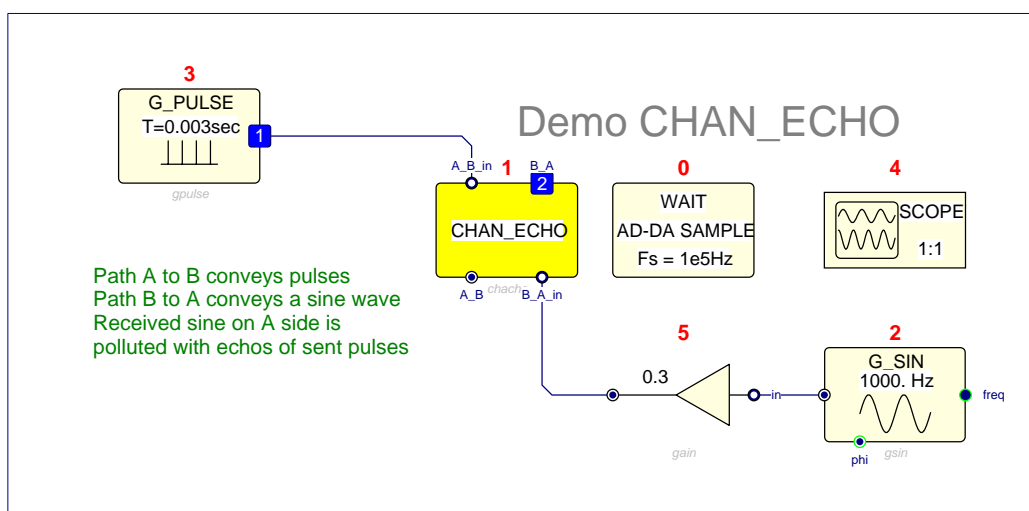
FRACT
FRACT

Data Struct:

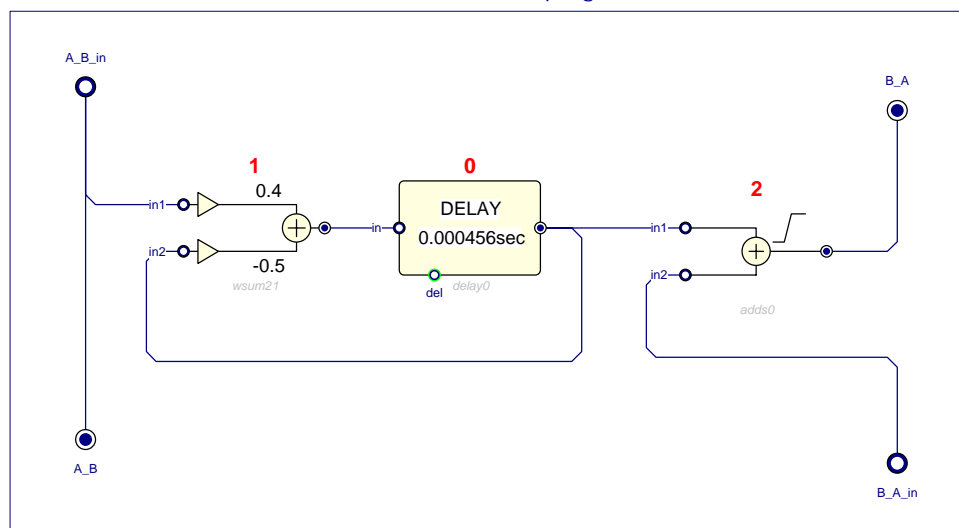
WORD
WORD

Connection:

normal
normal



CHAN_ECHO test program



CHAN_ECHO internal schema

CHAN_RINGING

Channel w. resonance



CATEGORY: TELECOM

DESCRIPTION:
Channel w. resonance

INPUTS

Name:
name_in

Data Type:
FRACT

Data Struct:
WORD

Connection:
mandatory

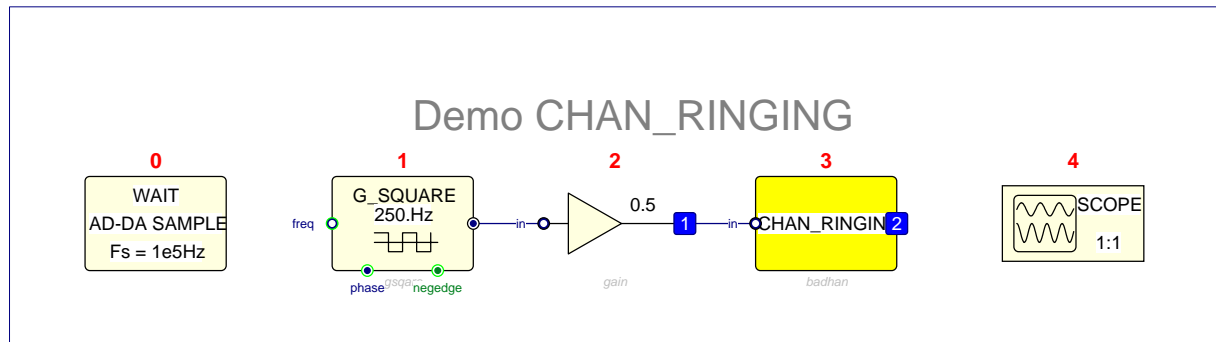
OUTPUTS

Name:
name

Data Type:
FRACT

Data Struct:
WORD

Connection:
normal

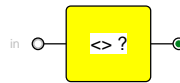


CHAN_RINGING test program

CHANGED

Check if input has changed

CHANGED



CATEGORY: LOGIC

DESCRIPTION:

Check if input has changed
Outputs TRUE if input has changed

INPUTS

Name:
name_in

Data Type:
defined by cn

Data Struct:

Connection:
mandatory

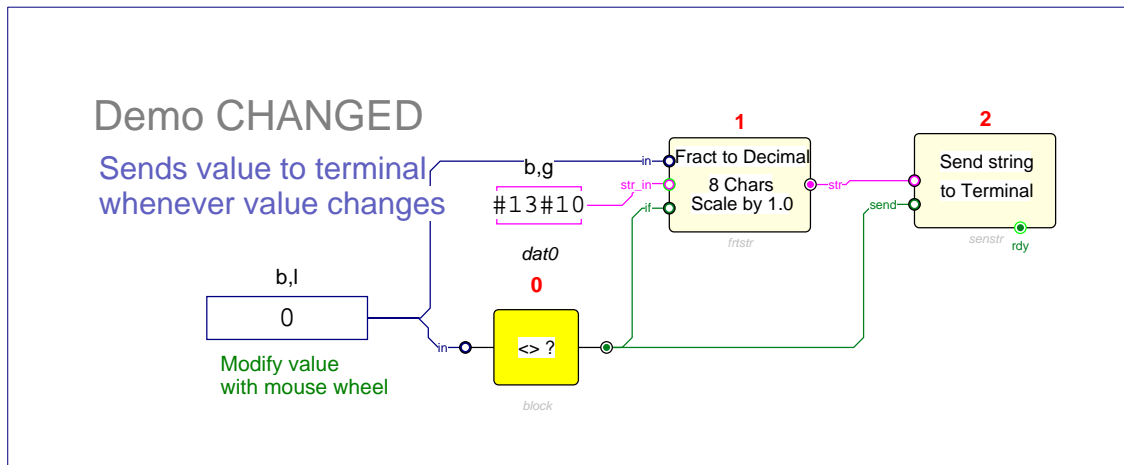
OUTPUTS

Name:
name

Data Type:
BOOL

Data Struct:
BIT

Connection:
normal

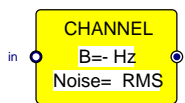


CHANGED test program

CHANNEL

Channel simulation

CHANNEL



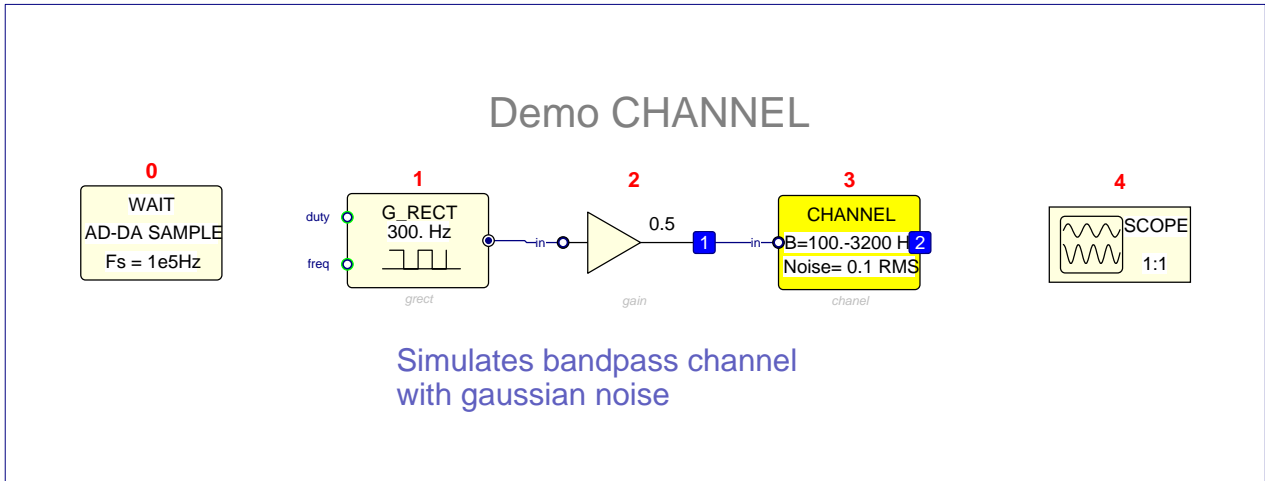
CATEGORY: TELECOM

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

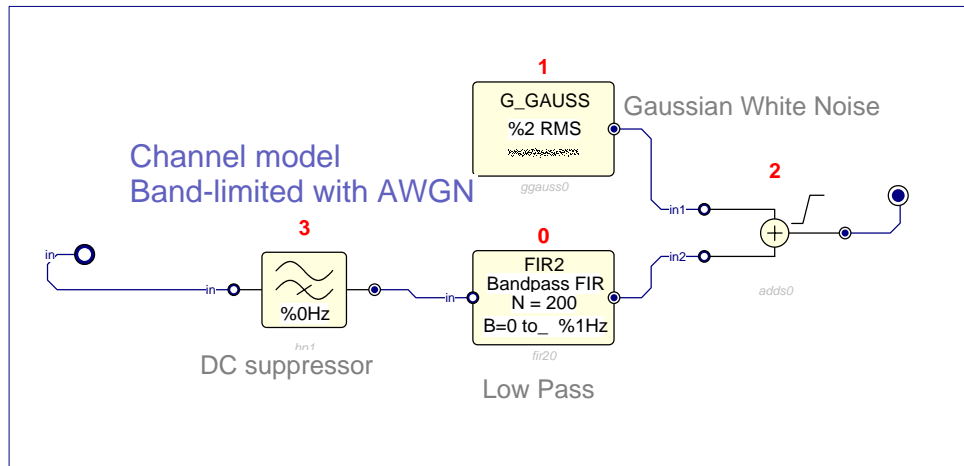
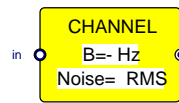


CHANNEL test program

CHANNEL

Channel simulation

CHANNEL

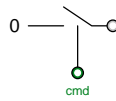


CHANNEL internal schema

CLEAR

Clear

CLEAR



CATEGORY: CONTROL

DESCRIPTION:

Clear

If cmd true then force connected output to 0

INPUTS

Name:

name

name_cmd

Data Type:

defined by cn

BOOL

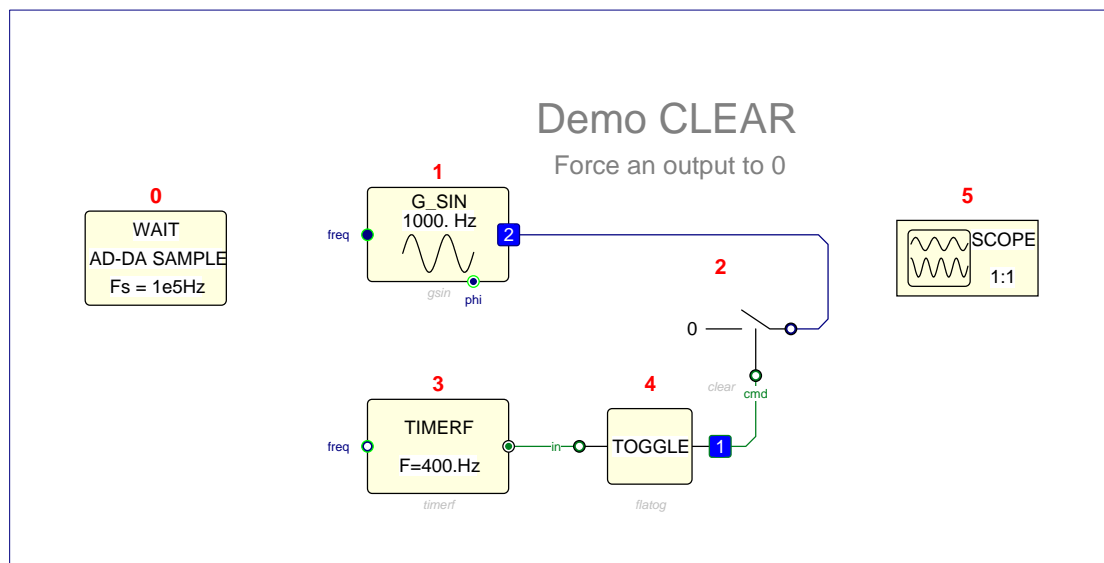
Data Struct:

BIT

Connection:

mandatory

mandatory

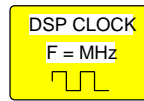


CLEAR test program

CLOCK

Change DSP clock frequency

CLOCK



CATEGORY: ETD410K

DESCRIPTION:

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

PARAMETERS:

Parameter:

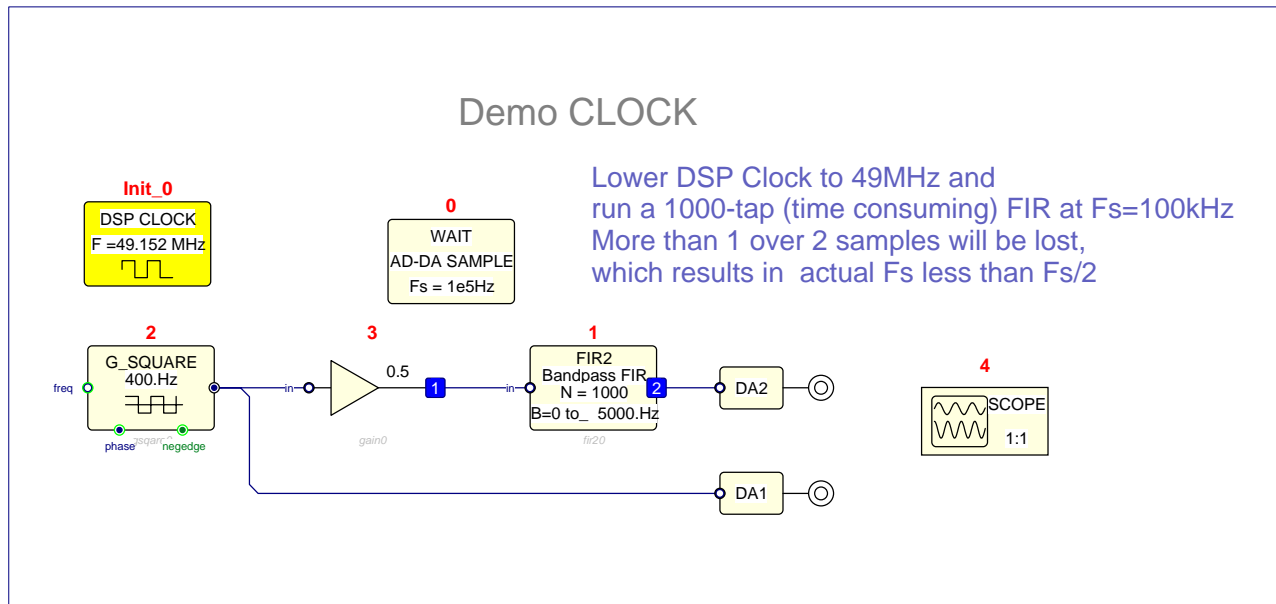
Frequency (MHz)

Default values:

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,

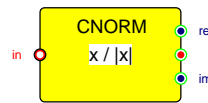


CLOCK test program

CNORM

Norm a complex variable

CNORM



CATEGORY: CONTROL

DESCRIPTION:

Norm a complex variable
 $z = x / |x|$

INPUTS

Name:
name_in

Data Type:
COMPLEX

Data Struct:
WORD

Connection:
mandatory

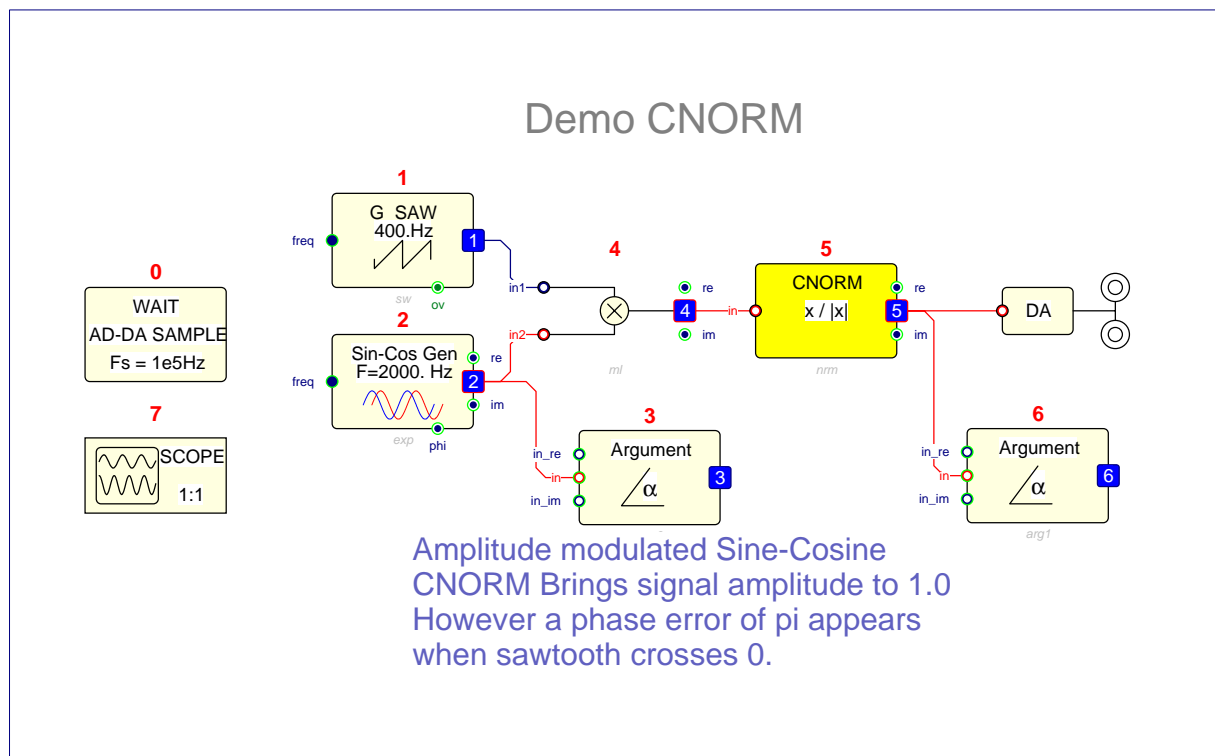
OUTPUTS

Name:
name
name_re
name_im

Data Type:
COMPLEX
FRACT
FRACT

Data Struct:
WORD
WORD
WORD

Connection:
optional
optional
optional

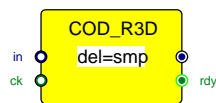


CNORM test program

COD_R3D

1-3 Repetition coder

COD_R3D



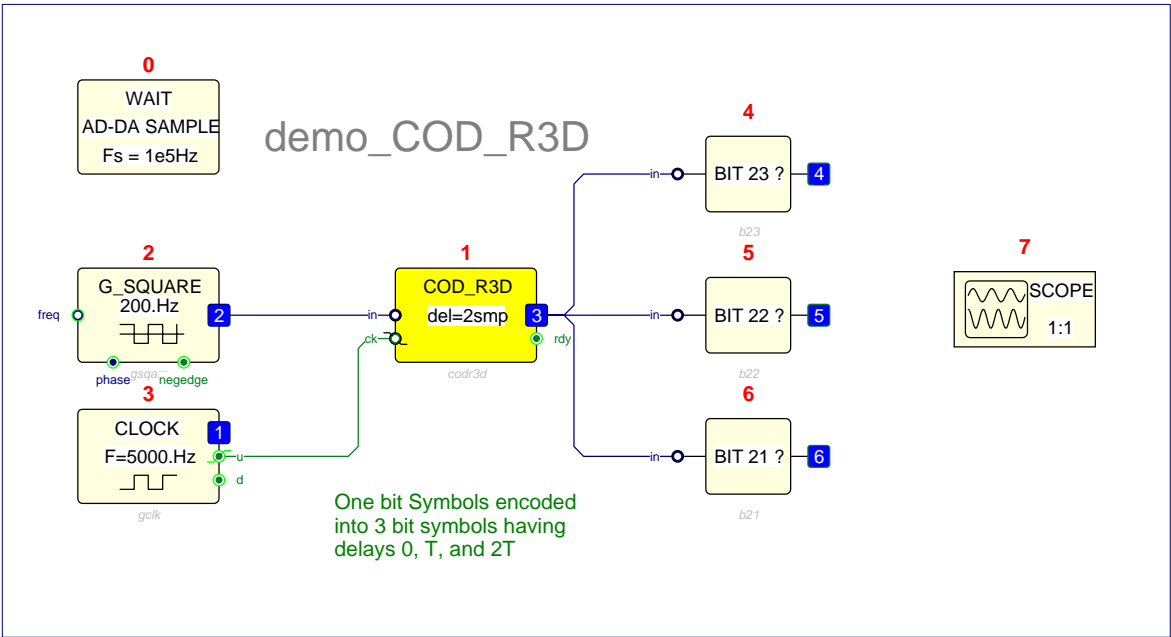
CATEGORY: TELECOM

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

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

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

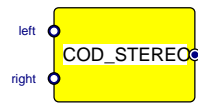
OUTPUTS			
Name:	Data Type:	Data Struct:	Connection:
name	FRACT	WORD	normal
name_rdy	BOOL	BIT	optional



COD_R3D test program

COD_STEREO

COD_STEREO



INPUTS

Name:
name_left
name_right

Data Type:
FRACT
FRACT

Data Struct:
WORD
WORD

Connection:
mandatory
mandatory

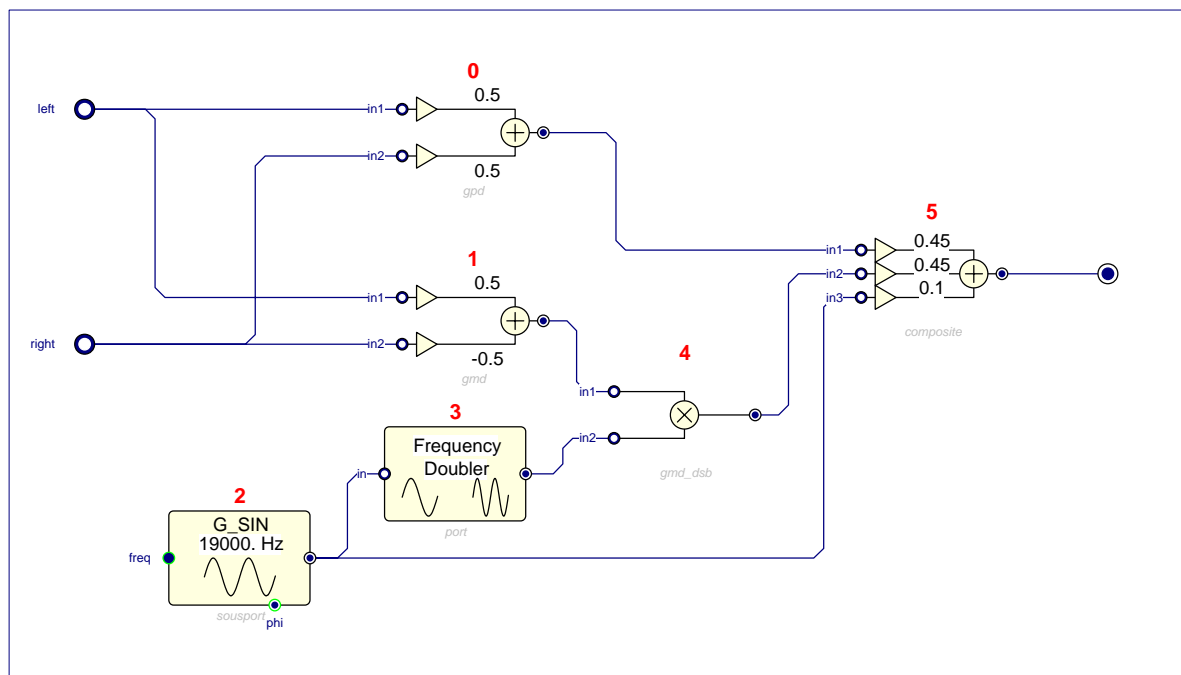
OUTPUTS

Name:
name

Data Type:
FRACT

Data Struct:
WORD

Connection:
normal

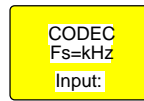


COD_STEREO internal schema

CODEC

Audio CODEC

CODEC



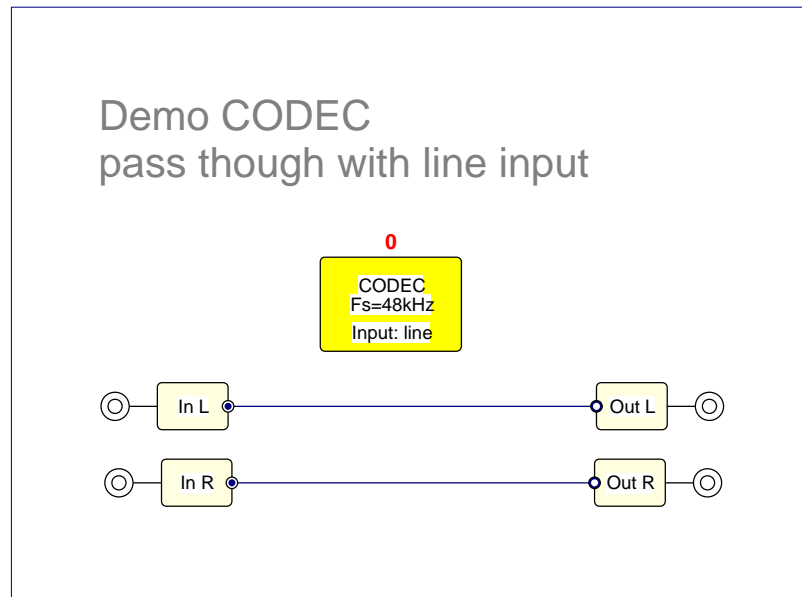
CATEGORY: AUDIO

DESCRIPTION:
Audio CODEC
Type tlv320aic23

PARAMETERS:
Parameter:
Fs (kHz)
Input

Default values:
8,32,48,96
line,mic

ATTRIBUTES
Unique, Execute First, Defines: actual_fs



CODEC test program



CATEGORY: MUSIC

DESCRIPTION:

MIDI File

Transcribed in asm format

OUTPUTS

Name:

name

Data Type:

INTEGER

Data Struct:

Matrix of WORD

Connection:

normal

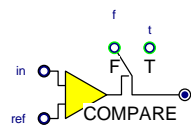
ATTRIBUTES

Non executable, Unique, Data Table

COMPARE

Relais function

COMPARE



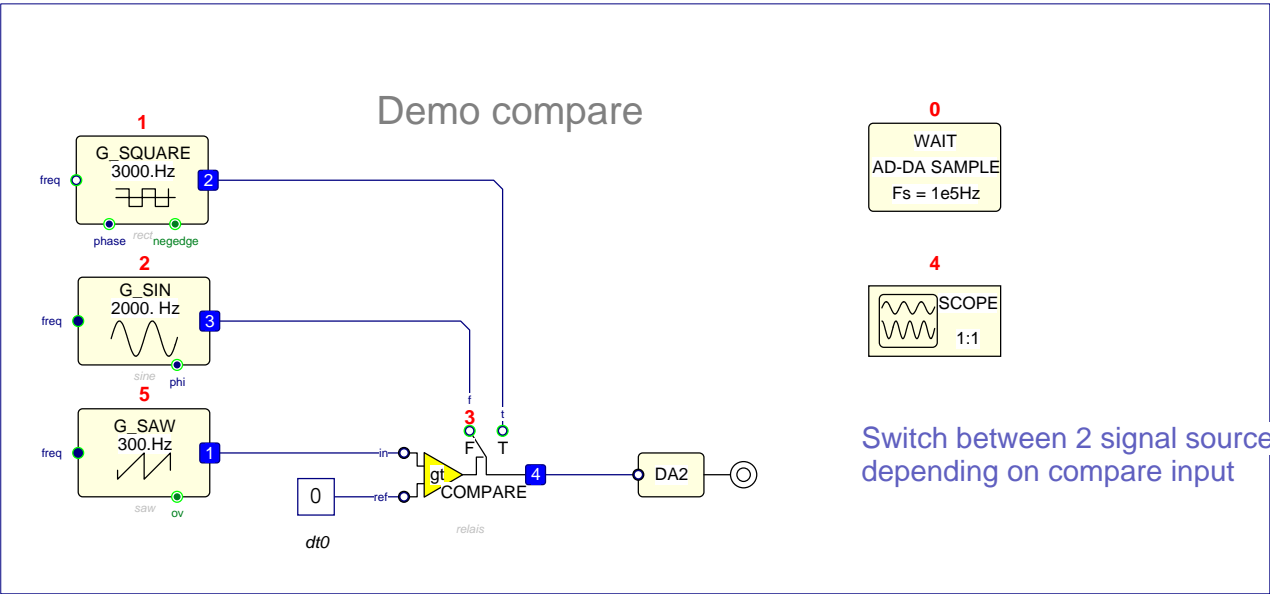
CATEGORY: CONTROL

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

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

INPUTS			
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name_in	FRACT	WORD	mandatory
name_ref	FRACT	WORD	mandatory
name_t	FRACT	WORD	optional
name_f	FRACT	WORD	optional

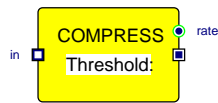
OUTPUTS			
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name	FRACT	WORD	normal



COMPARE test program

Suppress small values

COMPRESS



CATEGORY: MATRIX

DESCRIPTION:

Suppress small values

Values $<$ Threshold are replaced by 0

chains of zeros are replaced by code \$80nnnn

where $nnnn$ is the number of contiguous zeros

PARAMETERS:

Parameter:

Threshold

Default values:

0.01

INPUTS

Name:

name_in

Data Type:

FRACT

Data Struct:

Matrix of WORD

Connection:

mandatory

OUTPUTS

Name:

name

name_rate

Data Type:

FRACT

FRACT

Data Struct:

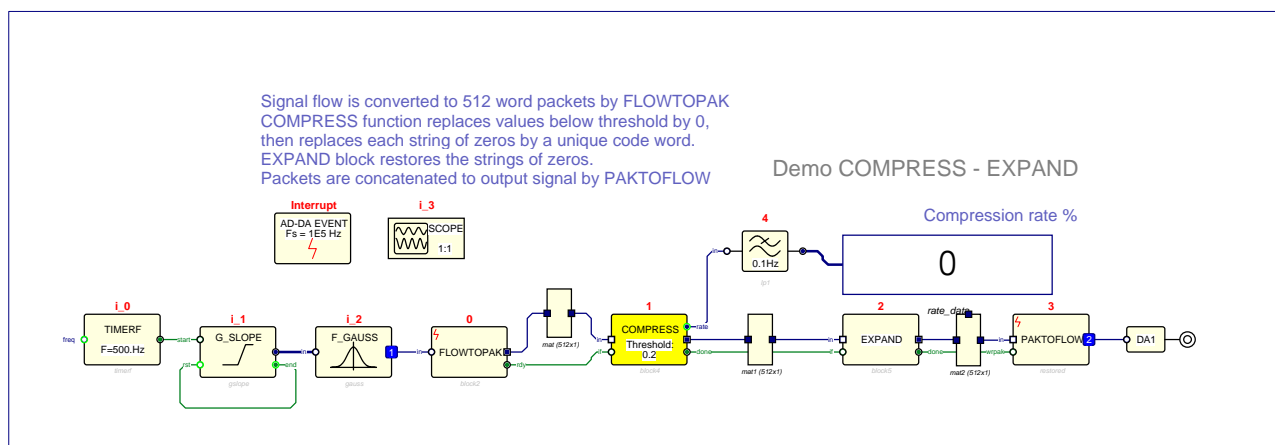
Matrix of WORD

WORD

Connection:

normal

optional



COMPRESS test program

CONTRAPUNCT

MIDI File

CONTRAPUNCT



CATEGORY: MUSIC

DESCRIPTION:

MIDI File

Transcribed in asm format

OUTPUTS

Name:

name

Data Type:

INTEGER

Data Struct:

Matrix of WORD

Connection:

normal

ATTRIBUTES

Non executable, Unique, Data Table

COPY

Copy data

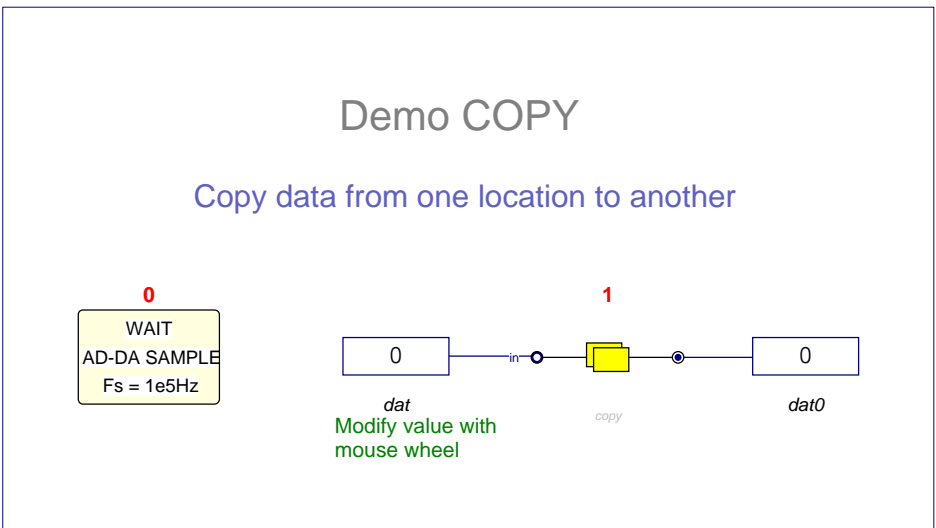
COPY



CATEGORY: ARITHMETIC

DESCRIPTION:
Copy data
Copy any data to different address

INPUTS			
Name:	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name_in	defined by cn		mandatory
OUTPUTS			
Name:	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name	defined by cn		normal

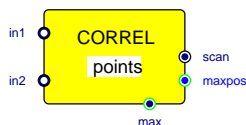


COPY test program

CORREL

Cross correlation

CORREL



CATEGORY: STAT

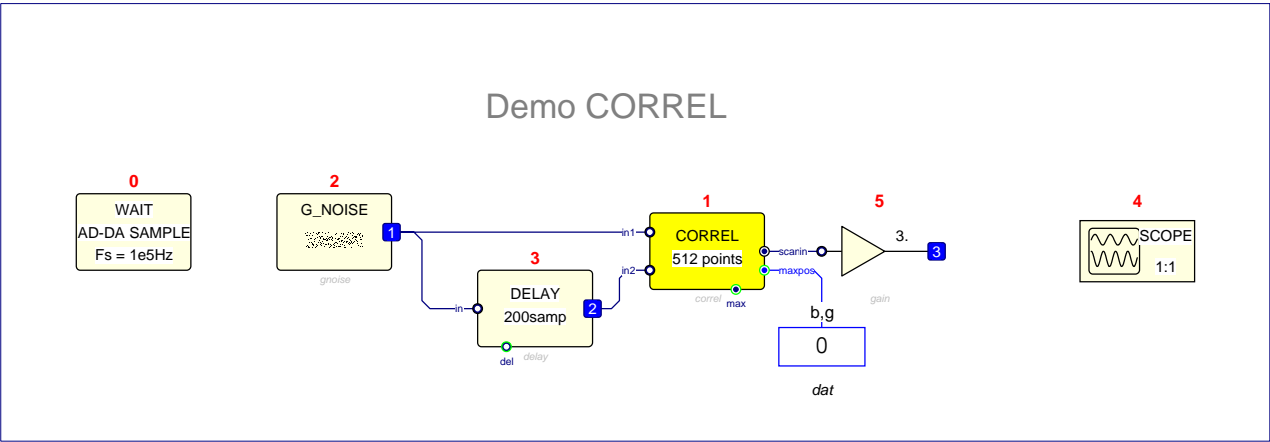
DESCRIPTION:
Cross correlation
between Input1 and Input 2
Fract output is a continuous scan of correlation function
Causal: $C(n,k)=(1-\text{epsi})\cdot C(n,k-1)+\text{epsi}\cdot(\text{in1}(k-n)\cdot\text{in2}(k)) \quad n=0..N-1$
Non_causal: $C(n,k)=(1-\text{epsi})\cdot C(n,k-1)+\text{epsi}\cdot(\text{in1}(k-n)\cdot\text{in2}(k-N/2)) \quad n=0..N-1$

PARAMETERS:
Parameter:
Points
epsi
Causal / Non Causal

Default values:
100
1e-4
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

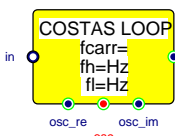


CORREL test program

COSTAS

COSTAS loop

COSTAS

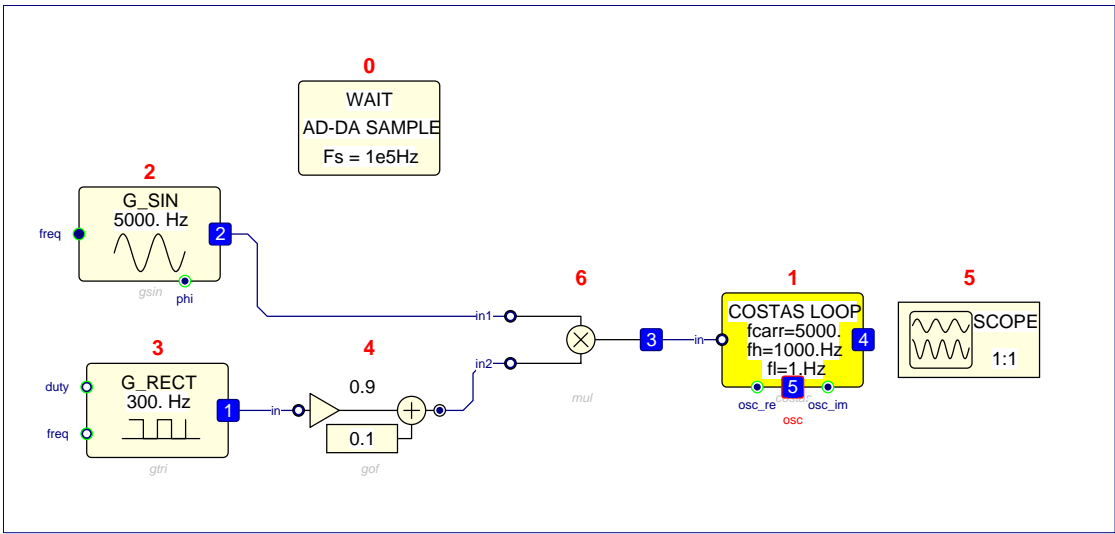


CATEGORY: TELECOM

DESCRIPTION:
COSTAS loop
for ASK demodulation and carrier recovery.

PARAMETERS:
Parameter: *Default values:*
Fcar 5000.
Fhi 5000.
Flo 1.

INPUTS			
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name_in	FRACT	WORD	mandatory
OUTPUTS			
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name	FRACT	WORD	optional
name_osc	COMPLEX	WORD	optional
name_osc_re	FRACT	WORD	optional
name_osc_im	FRACT	WORD	optional

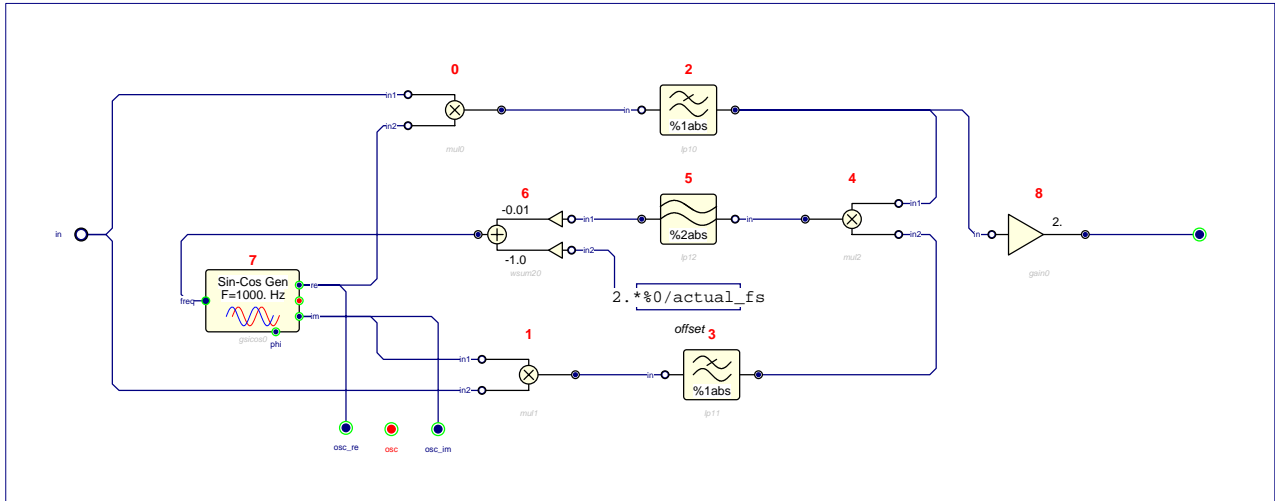
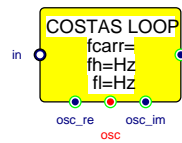


COSTAS test program

COSTAS

COSTAS loop

COSTAS

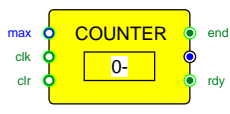


COSTAS internal schema

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:

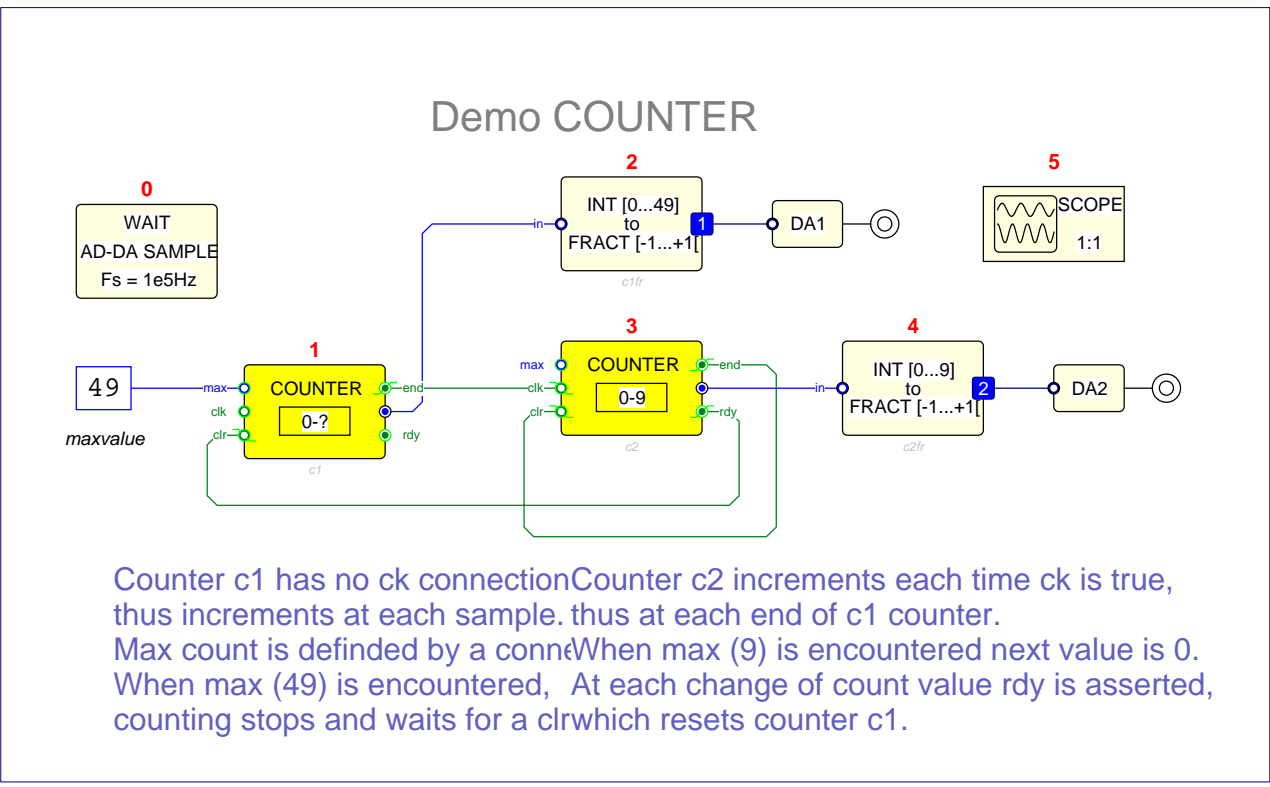
Parameter: maxi
Default values: 10

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

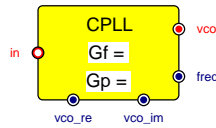


COUNTER test program

CPLL

Complex PLL

CPLL



CATEGORY: CONTROL

DESCRIPTION:
Complex PLL

PARAMETERS:

Parameter:
frequency_gain
phase_gain

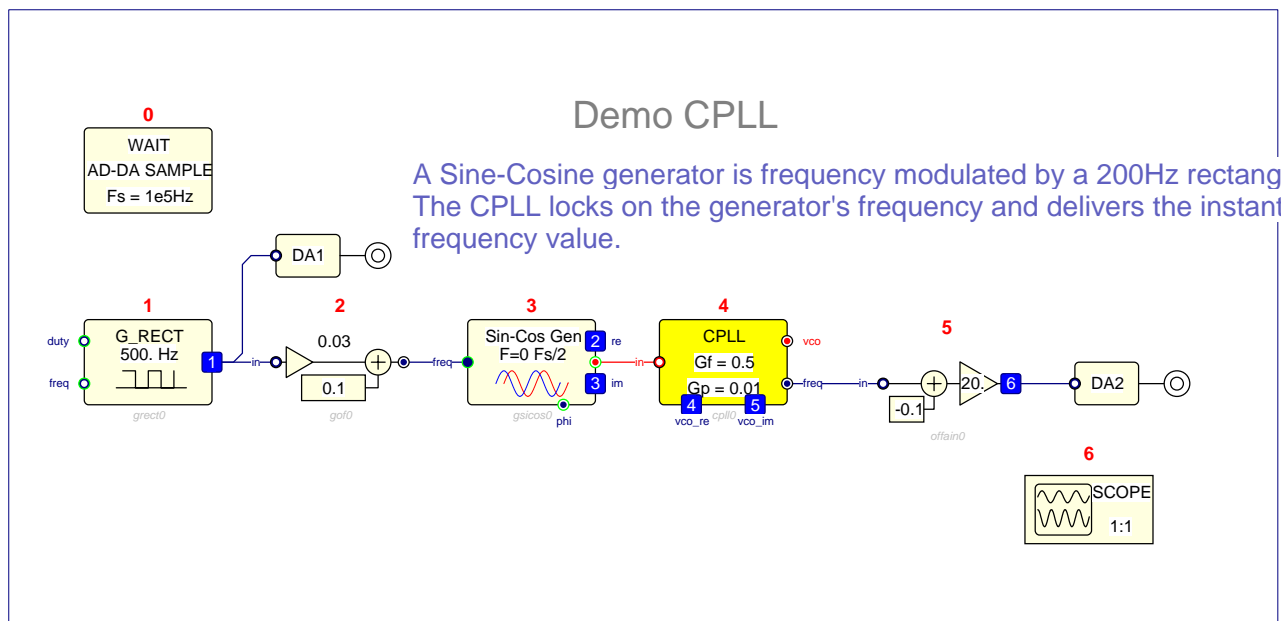
Default values:
0.01
0.01

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

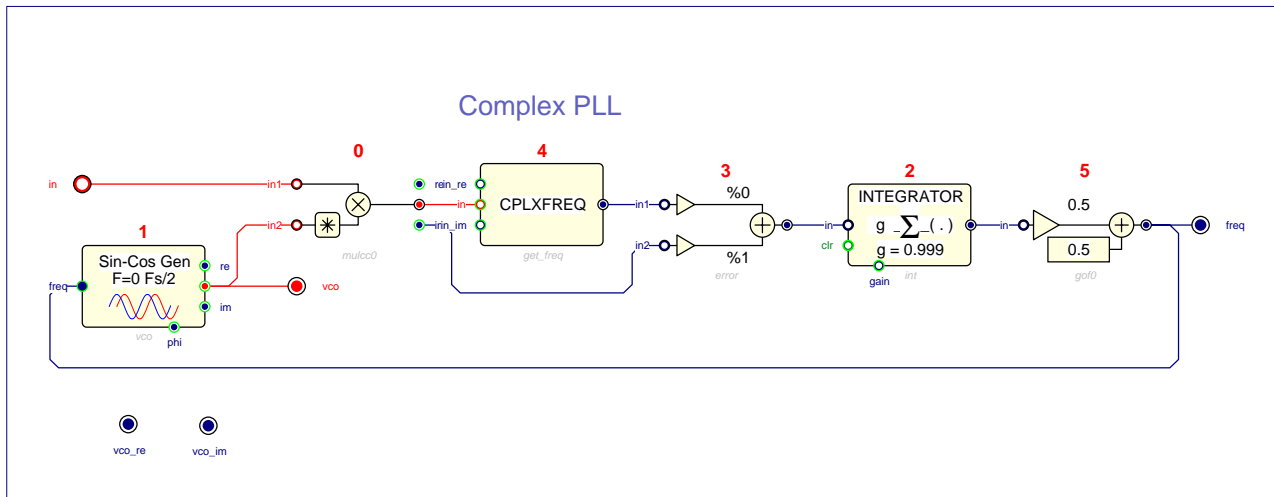
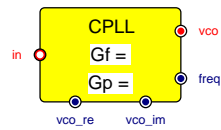


CPLL test program

CPLL

Complex PLL

CPLL



CPLL internal schema

CPLX_POW

Complex Power

CPLX_POW



CATEGORY: FUNCTIONS

DESCRIPTION:

Complex Power
Computes $\text{Re}^2 + \text{Im}^2$

INPUTS

Name:
name_in

Data Type:
COMPLEX

Data Struct:
WORD

Connection:
mandatory

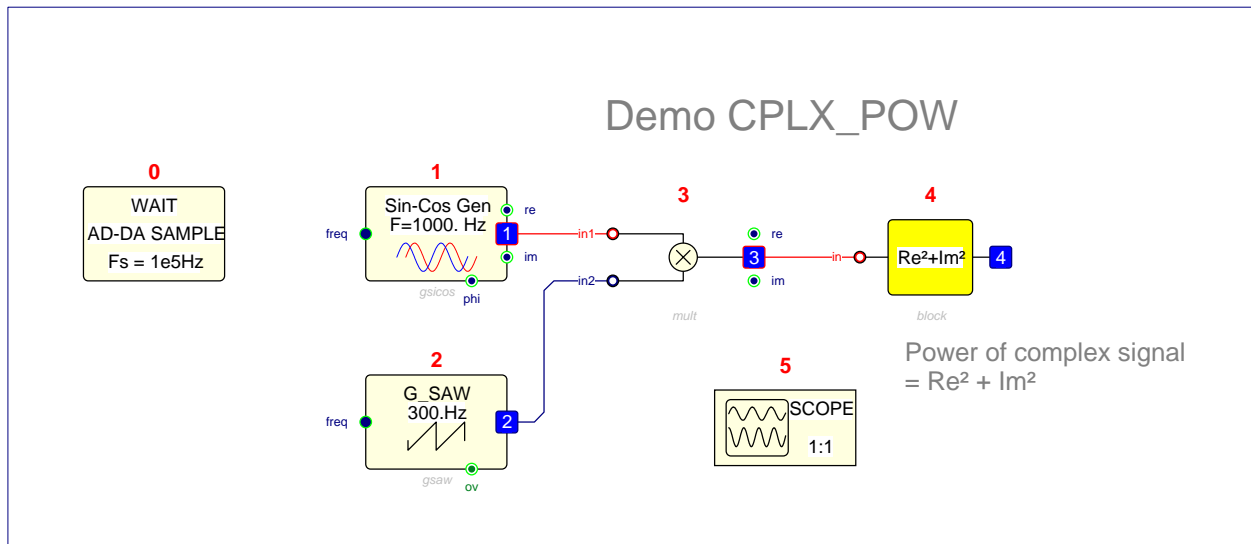
OUTPUTS

Name:
name

Data Type:
FRACT

Data Struct:
WORD

Connection:
normal

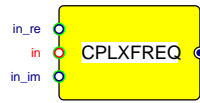


CPLX_POW test program

CPLXFREQ

Instantaneous frequency

CPLXFREQ



CATEGORY: CONTROL

DESCRIPTION:

Instantaneous frequency

of complex signal

$$y = f/(F_s/2) = 1/\pi * \text{Arg}(x(k).x^*(k-1))$$

INPUTS

Name:

name_in
name_in_re
name_in_im

Data Type:

COMPLEX
FRACT
FRACT

Data Struct:

WORD
WORD
WORD

Connection:

optional
optional
optional

OUTPUTS

Name:

name

Data Type:

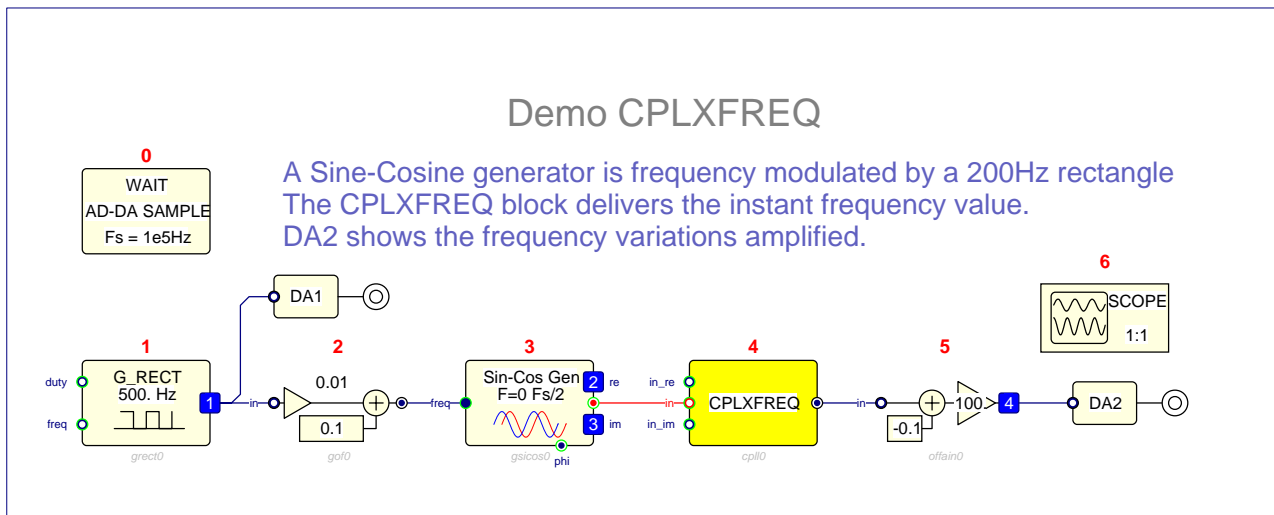
FRACT

Data Struct:

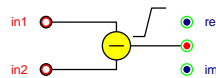
WORD

Connection:

normal



CPLXFREQ test program



CATEGORY: ARITHMETIC

DESCRIPTION:

Complex or mixed Subtraction
with saturation

INPUTS

Name:
name_in1
name_in2

Data Type:
COMPLEX
COMPLEX

Data Struct:
WORD
WORD

Connection:
mandatory
mandatory

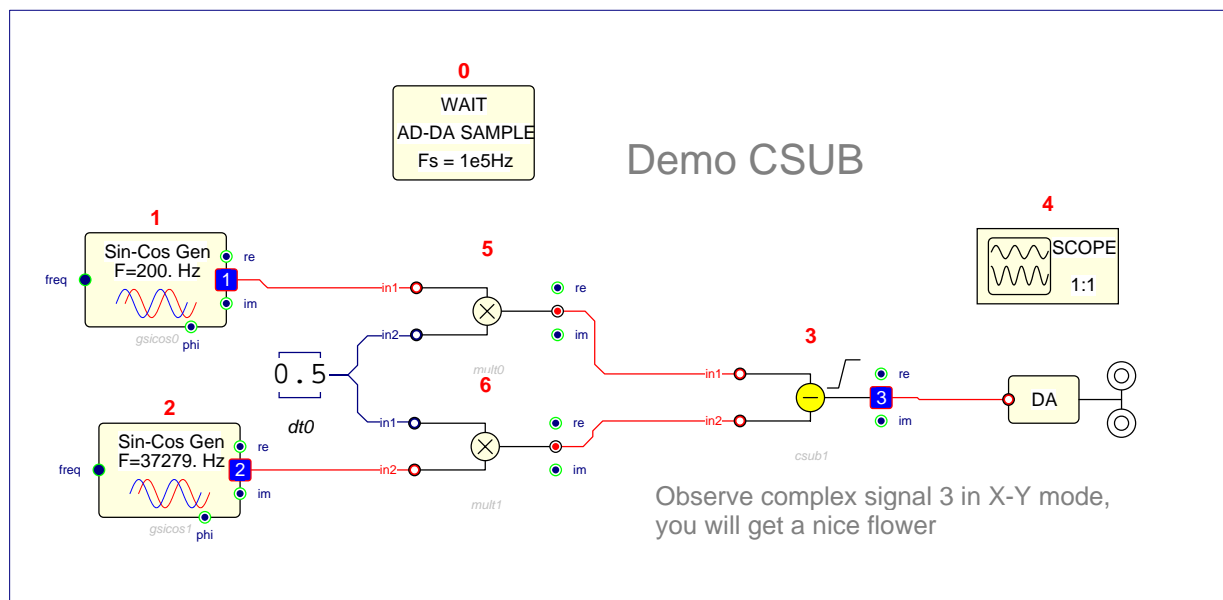
OUTPUTS

Name:
name
name_re
name_im

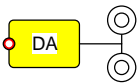
Data Type:
COMPLEX
FRACT
FRACT

Data Struct:
WORD
WORD
WORD

Connection:
optional
optional
optional



CSUB test program

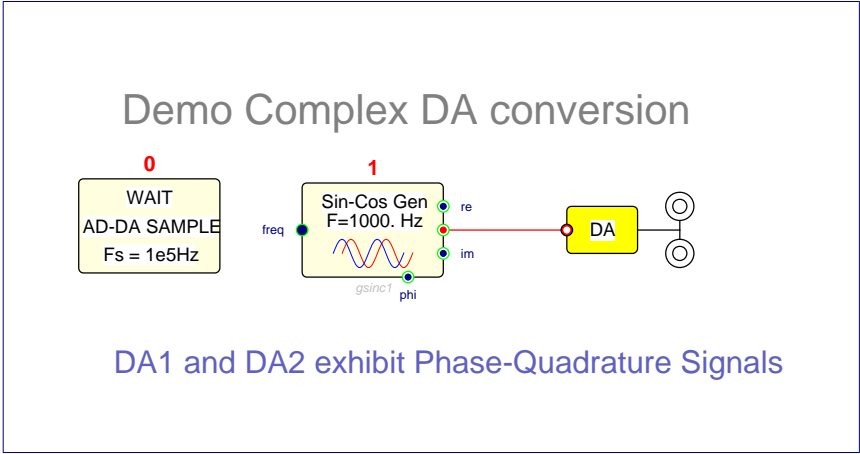


CATEGORY: ETD410K

DESCRIPTION:
Complex DAC
DA1=Re DA2=Im

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

ATTRIBUTES
Non executable, Unique,



DA test program

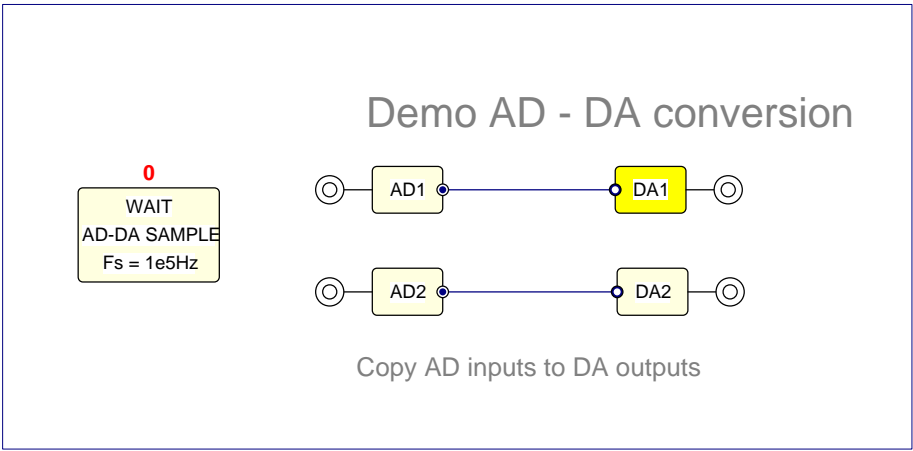


CATEGORY: ETD410K

DESCRIPTION:
Digital to Analog Converter 1

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

ATTRIBUTES
Non executable, Unique,



DA1 test program

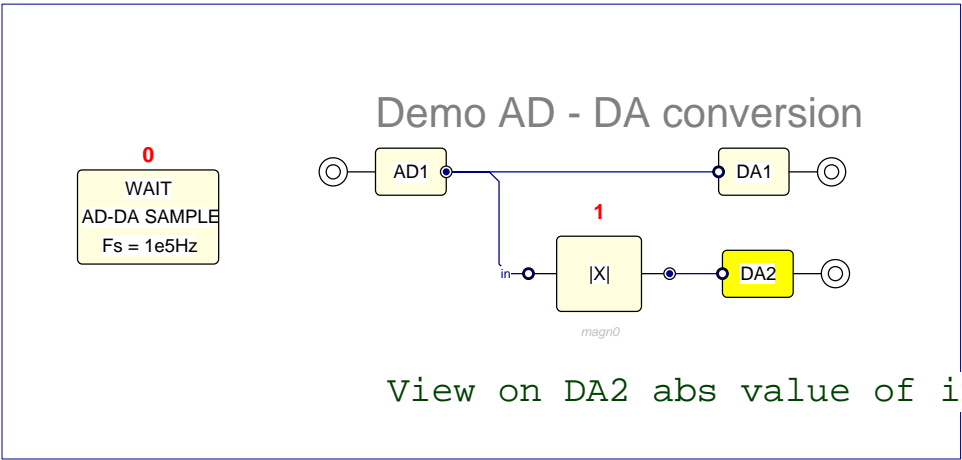


CATEGORY: ETD410K

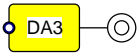
DESCRIPTION:
Digital to Analog Converter 2

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

ATTRIBUTES
Non executable, Unique,



DA2 test program



CATEGORY: ETD410K

DESCRIPTION:
Digital to Analog Converter 3

INPUTS			
<i>Name:</i> name	<i>Data Type:</i> FRACT	<i>Data Struct:</i> WORD	<i>Connection:</i> mandatory

ATTRIBUTES
Non executable, Unique,



CATEGORY: ETD410K

DESCRIPTION:
Digital to Analog Converter 4

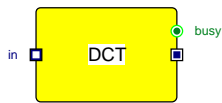
INPUTS			
<i>Name:</i> name	<i>Data Type:</i> FRACT	<i>Data Struct:</i> WORD	<i>Connection:</i> mandatory

ATTRIBUTES
Non executable, Unique,

DCT

Discrete Cosine Transform

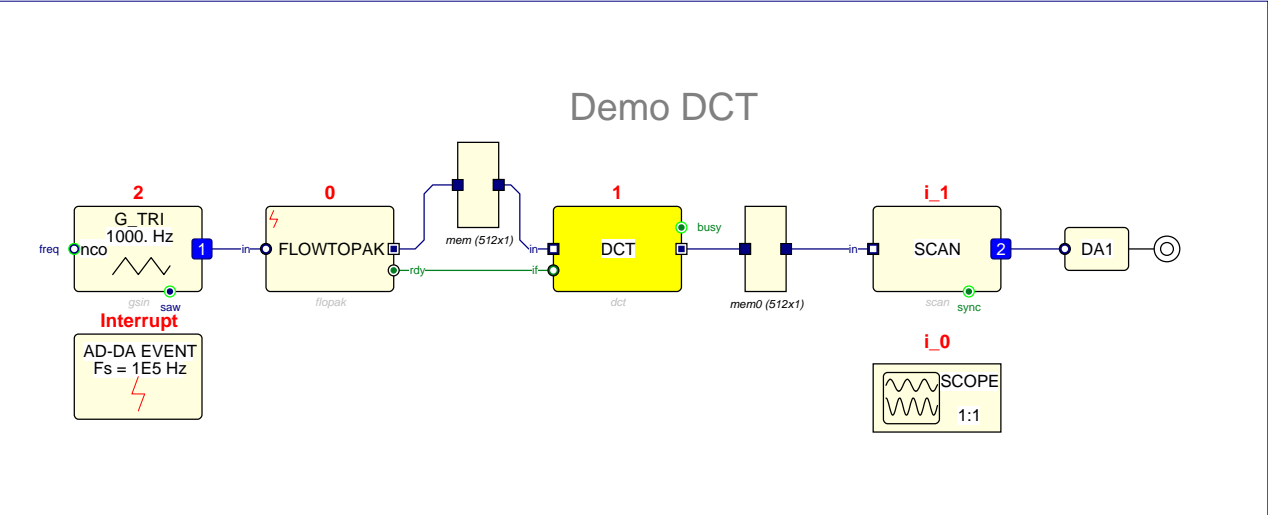
DCT



CATEGORY: MATRIX

DESCRIPTION:
Discrete Cosine Transform

INPUTS			
Name:	Data Type:	Data Struct:	Connection:
name_in	FRACT	Matrix of WORD	mandatory
OUTPUTS			
Name:	Data Type:	Data Struct:	Connection:
name	FRACT	Matrix of WORD	normal
name_busy	BOOL	BIT	optional



DCT test program

DEBUSSY1

MIDI File

DEBUSSY1



CATEGORY: MUSIC

DESCRIPTION:
MIDI File
Transcribed in asm format

OUTPUTS

<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name	INTEGER	Matrix of WORD	normal

ATTRIBUTES
Non executable, Unique, Data Table

DEBUSSY2

MIDI File

DEBUSSY2



CATEGORY: MUSIC

DESCRIPTION:
MIDI File
Transcripted in asm format

OUTPUTS

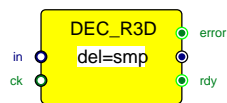
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name	INTEGER	Matrix of WORD	normal

ATTRIBUTES
Non executable, Unique, Data Table

DEC_R3D

3-1 Repetition decoder

DEC_R3D



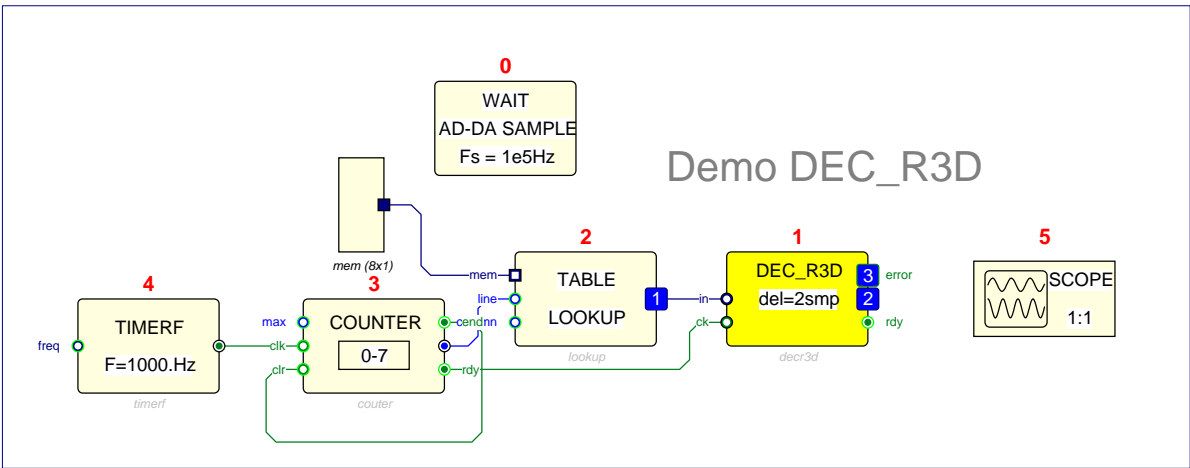
CATEGORY: TELECOM

DESCRIPTION:
3-1 Repetition decoder
with delay compensation

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

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

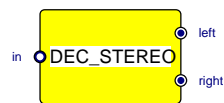
OUTPUTS	Data Type:	Data Struct:	Connection:
Name:			
name	FRACT	WORD	normal
name_rdy	BOOL	BIT	optional
name_error	BOOL	BIT	optional



DEC_R3D test program

DEC_STEREO

DEC_STEREO



INPUTS

Name:
name_in

Data Type:
FRACT

Data Struct:
WORD

Connection:
mandatory

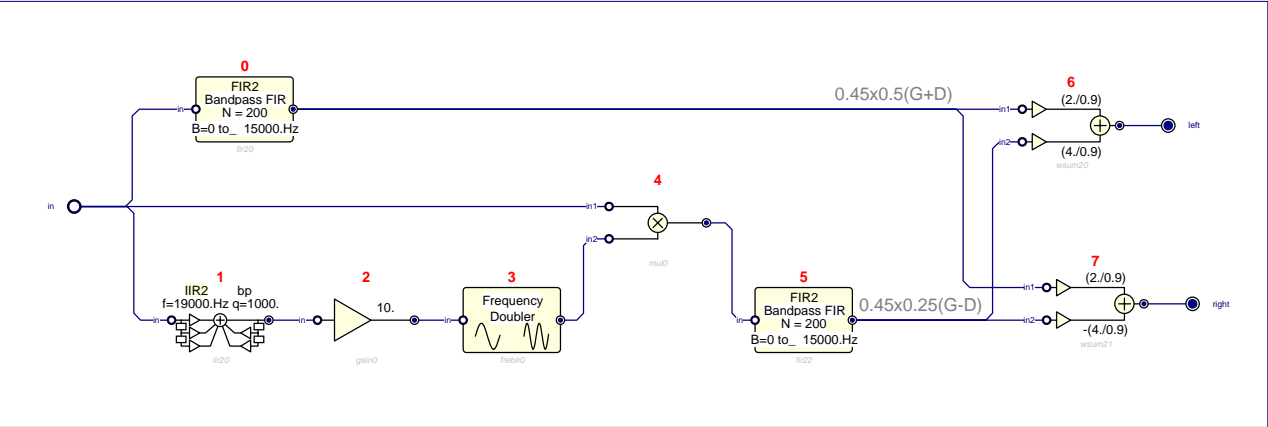
OUTPUTS

Name:
name_left
name_right

Data Type:
FRACT
FRACT

Data Struct:
WORD
WORD

Connection:
normal
normal

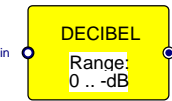


DEC_STEREO internal schema

DECIBEL

Decibel function

DECIBEL



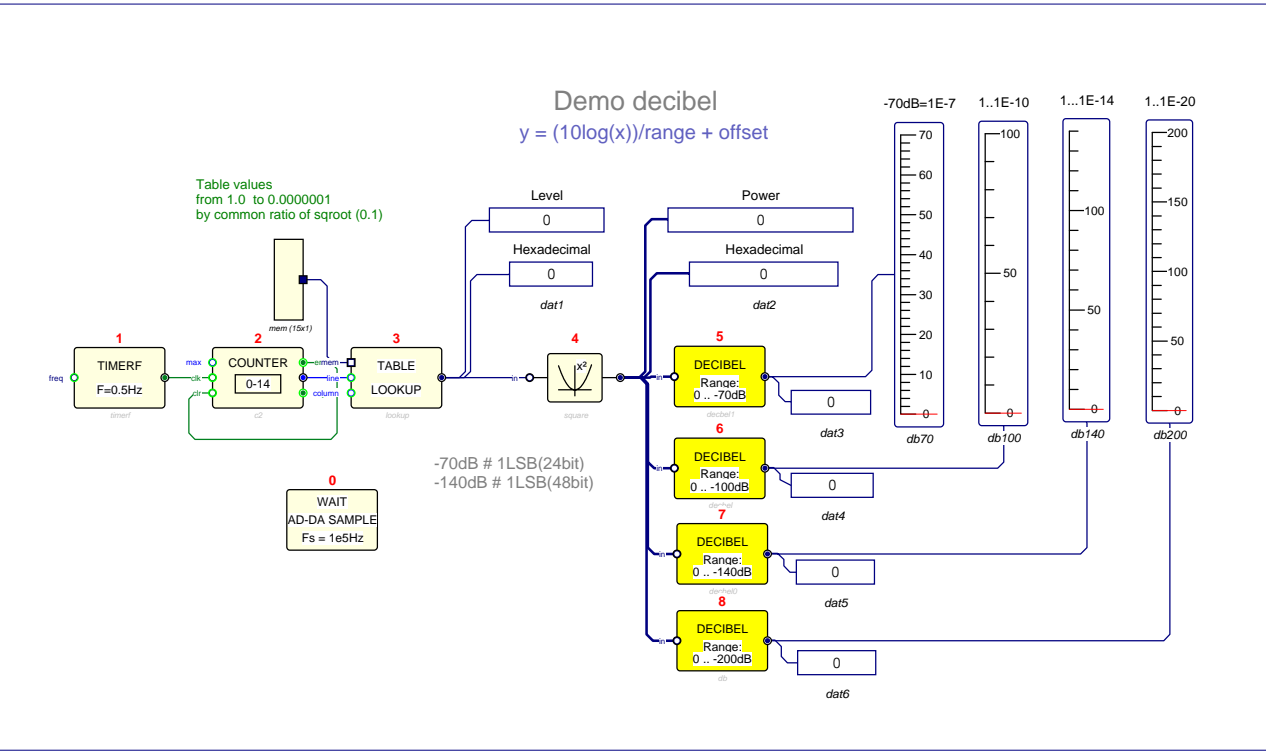
CATEGORY: FUNCTIONS

DESCRIPTION:
Decibel function
Output 0..1 corresponds to 0 ..-Range dB
 $y = 10 \cdot \log_{10}(x) / \text{Range} + \text{Offset}$

PARAMETERS:
Parameter: *Default values:*
Range 70,100,140,200
Offset 0

INPUTS
Name: *Data Type:* *Data Struct:* *Connection:*
name_in FRAC T WORD mandatory

OUTPUTS
Name: *Data Type:* *Data Struct:* *Connection:*
name FRAC T WORD normal

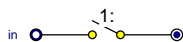


DECIBEL test program

DECIM

Decimation

DECIM



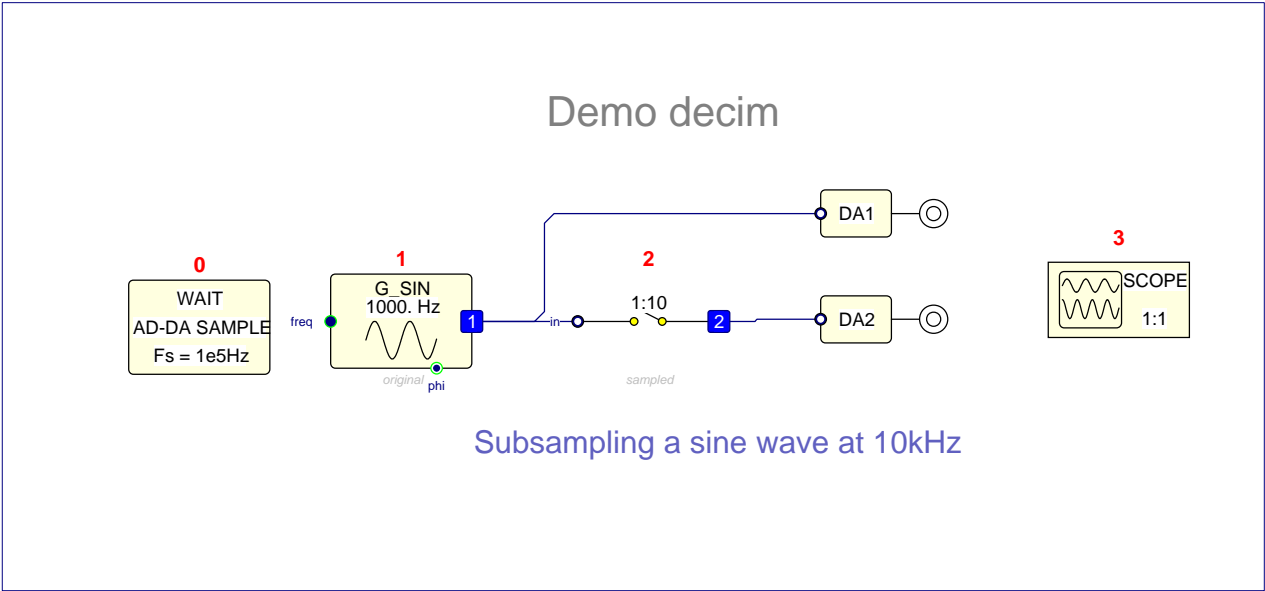
CATEGORY: CONTROL

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

PARAMETERS:
Parameter: N Default values: 10

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

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

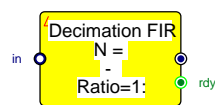


DECIM test program

DECIM_FIR

Decimation FIR

DECIM_FIR



CATEGORY: FILTERS

DESCRIPTION:

Decimation FIR
Updates delay line every sample in Interrupt
Calculates output every <ratio> samples in mainloop

PARAMETERS:

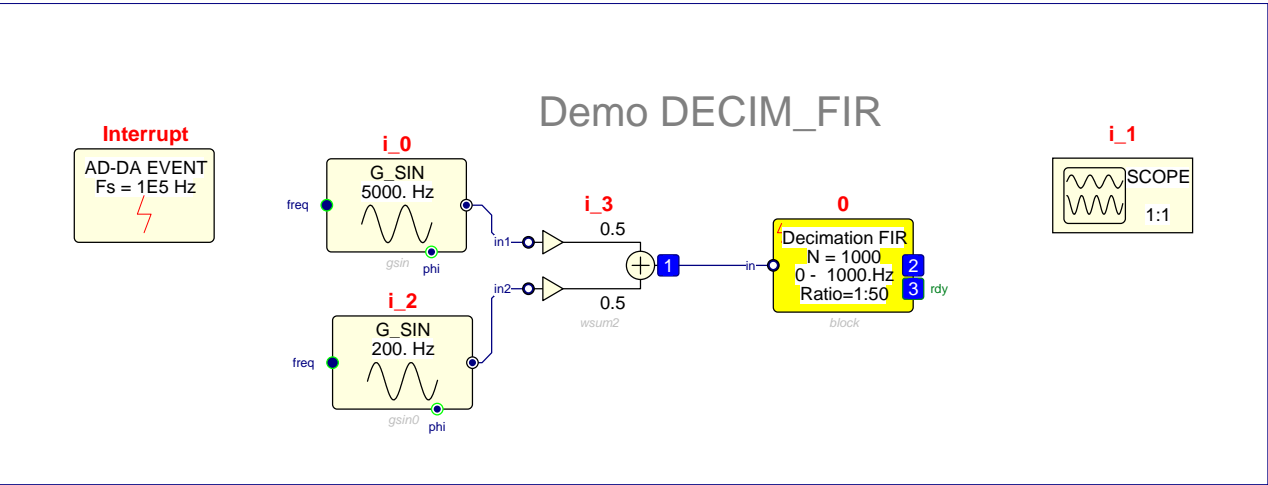
Parameter:	Default values:
Size	100
Freq low	0
Freq high	1E4
Unit	Hz,Fs/2
Ratio	10

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	optional

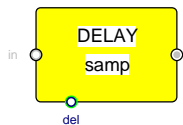


DECIM_FIR test program

DELAY

Real, complex, fixed, variable

DELAY



CATEGORY: CONTROL

DESCRIPTION:

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

PARAMETERS:

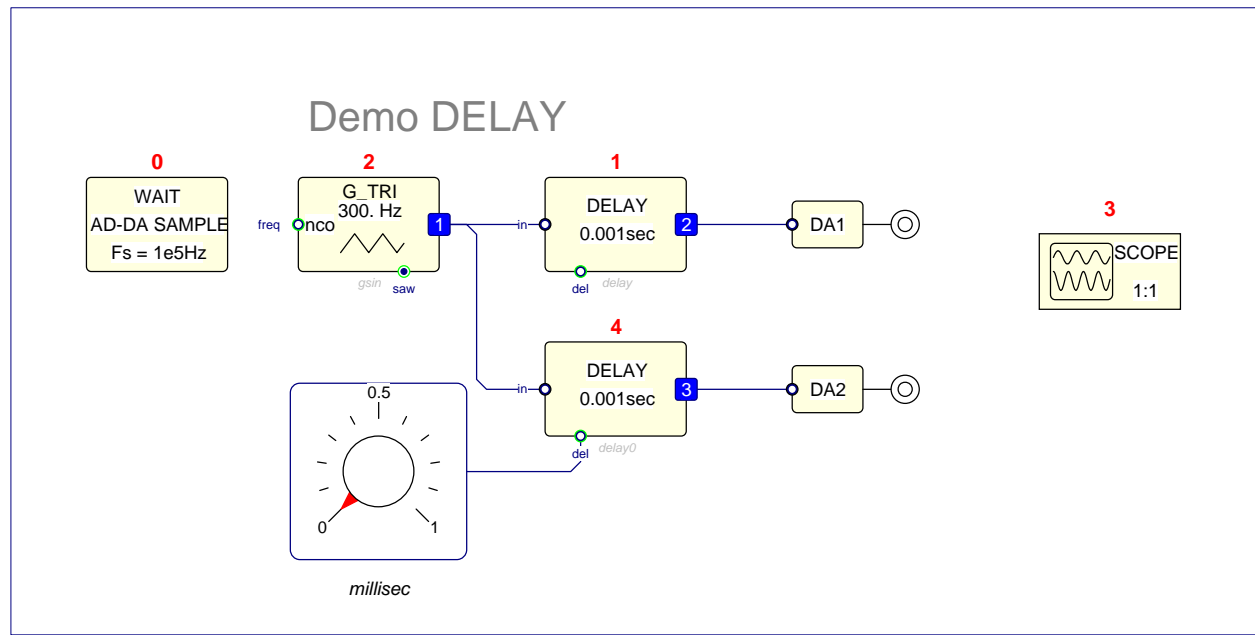
<i>Parameter:</i>	<i>Default values:</i>
Value	0.001
Unit	sec, samp

INPUTS

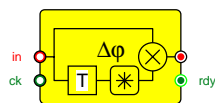
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name_in	defined by cn	WORD	mandatory
name_del	FRACT		optional

OUTPUTS

<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name	defined by cn		normal



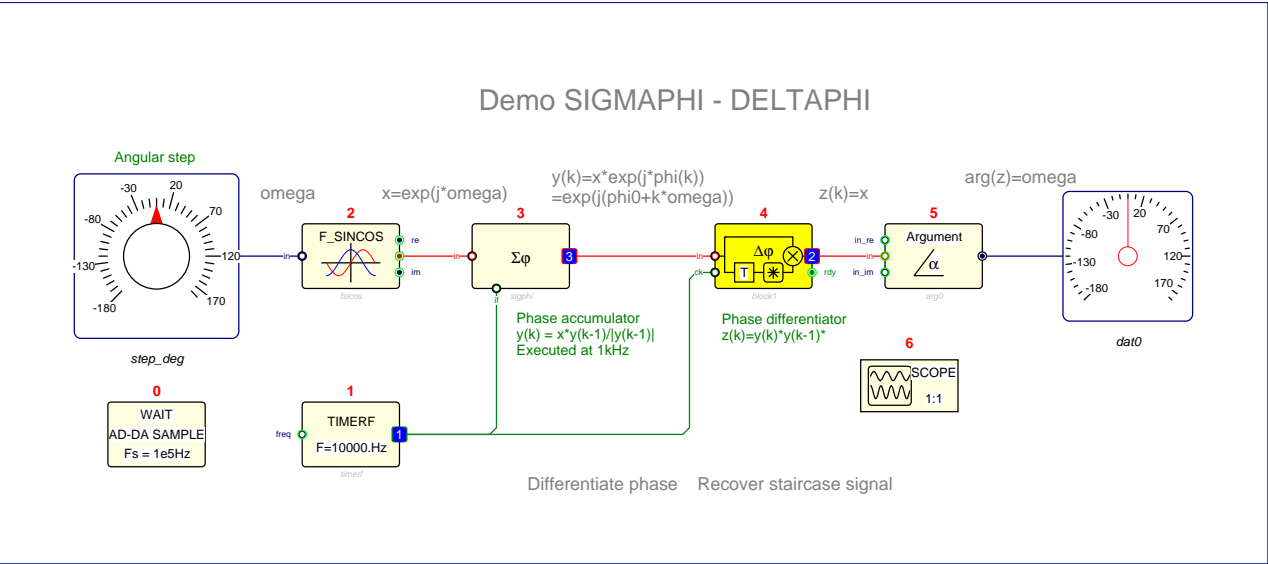
DELAY test program



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	optional

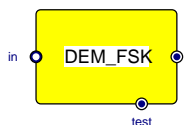


DELTAPHI test program

DEM_FSK

FSK Demodulator

DEM_FSK

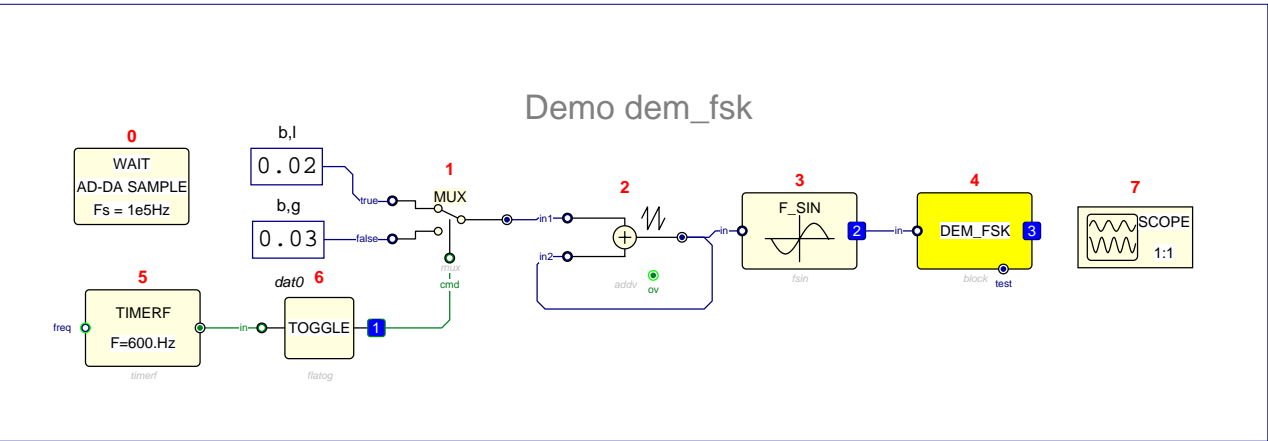


DESCRIPTION:
FSK Demodulator

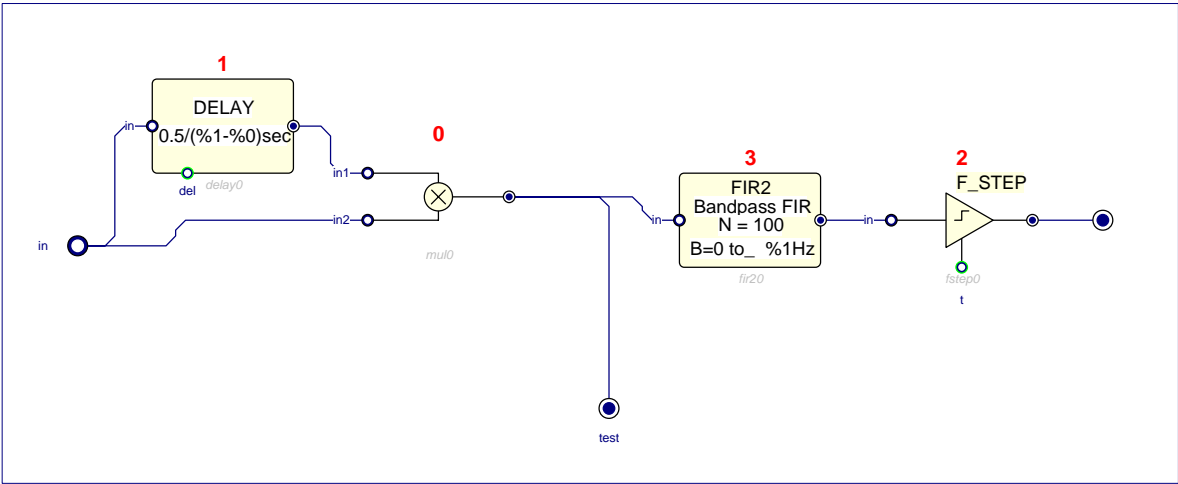
PARAMETERS:
Parameter: Default values:
f0 1000.
f1 1500.

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

OUTPUTS	Data Type:	Data Struct:	Connection:
Name: name_test	FRACT	WORD	normal



DEM_FSK test program

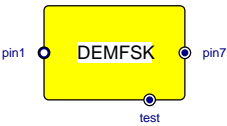


DEM_FSK internal schema

DEMFSK

FSK Demodulator

DEMFSK



CATEGORY: TELECOM

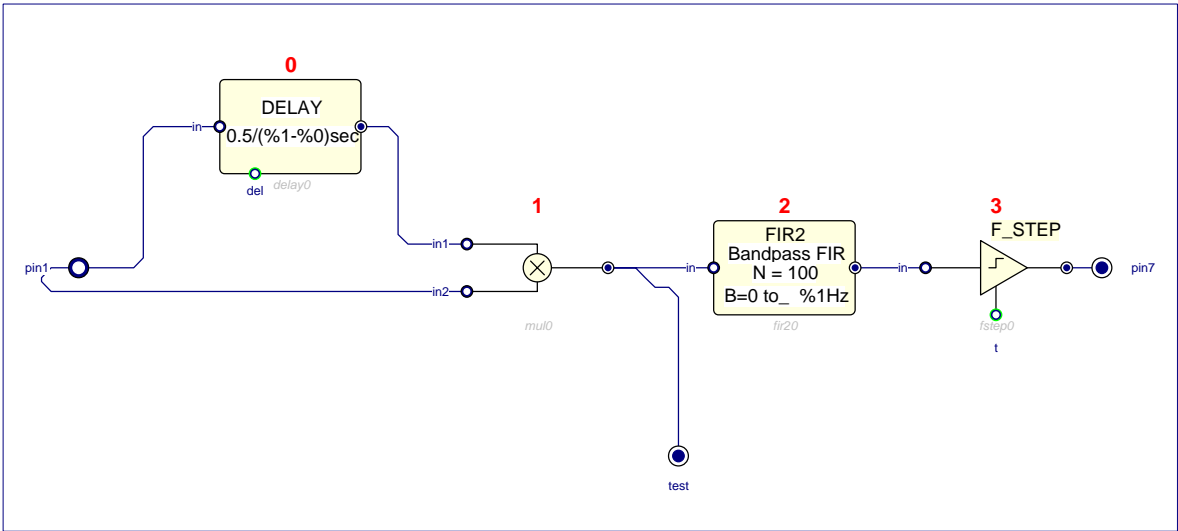
DESCRIPTION:
FSK Demodulator

PARAMETERS:
Parameter:
f0
f1

Default values:
1000.
1500.

INPUTS			
Name:	Data Type:	Data Struct:	Connection:
name_pin1	FRACT	WORD	mandatory

OUTPUTS			
Name:	Data Type:	Data Struct:	Connection:
name_pin7	FRACT	WORD	normal
name_test	FRACT	WORD	normal

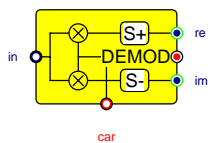


DEMFSK internal schema

DEMOD

QAM demodulator

DEMOD



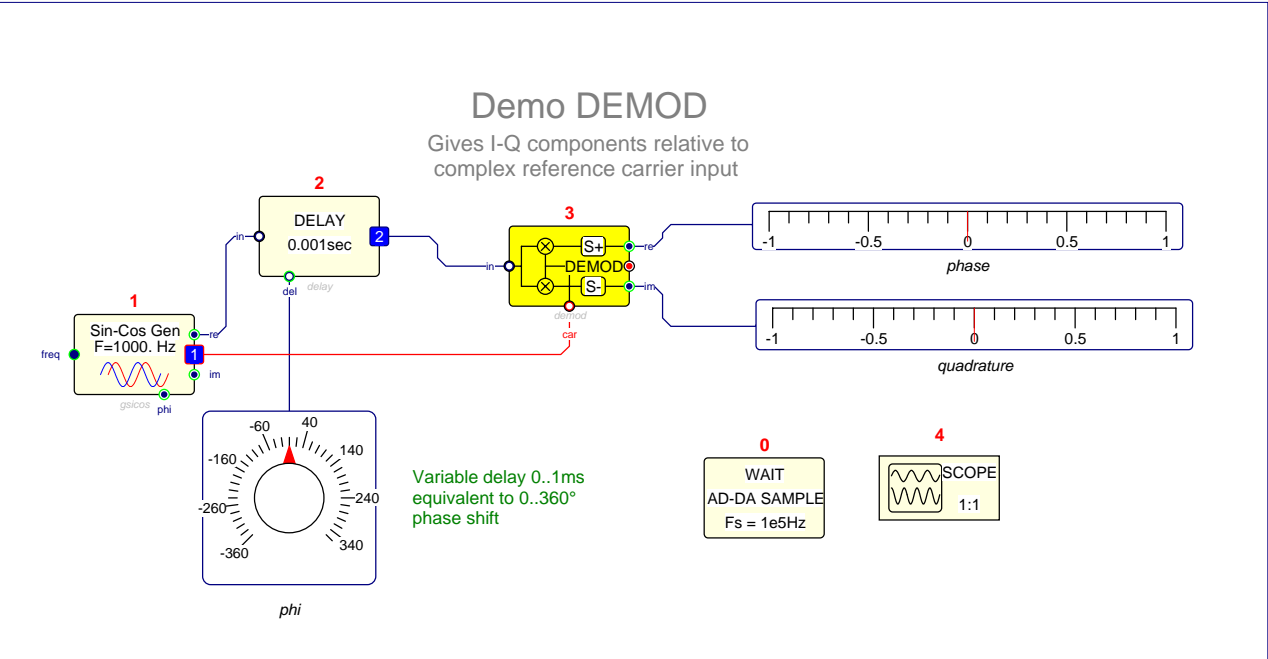
CATEGORY: TELECOM

DESCRIPTION:
QAM demodulator

PARAMETERS:
Parameter: Taccum
Default values: 0.001

INPUTS	Data Type:	Data Struct:	Connection:
<i>Name:</i> name_in	FRACT	WORD	mandatory
name_car	COMPLEX	WORD	mandatory

OUTPUTS	Data Type:	Data Struct:	Connection:
<i>Name:</i> name	COMPLEX	WORD	normal
name_re	FRACT	WORD	optional
name_im	FRACT	WORD	optional

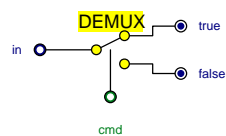


DEMOM test program

DEMUX

1 to 2 Demultiplexer

DEMUX



CATEGORY: CONTROL

DESCRIPTION:
1 to 2 Demultiplexer

INPUTS

Name:
name_cmd
name_in

Data Type:
BOOL
FRACT

Data Struct:
BIT
WORD

Connection:
mandatory
mandatory

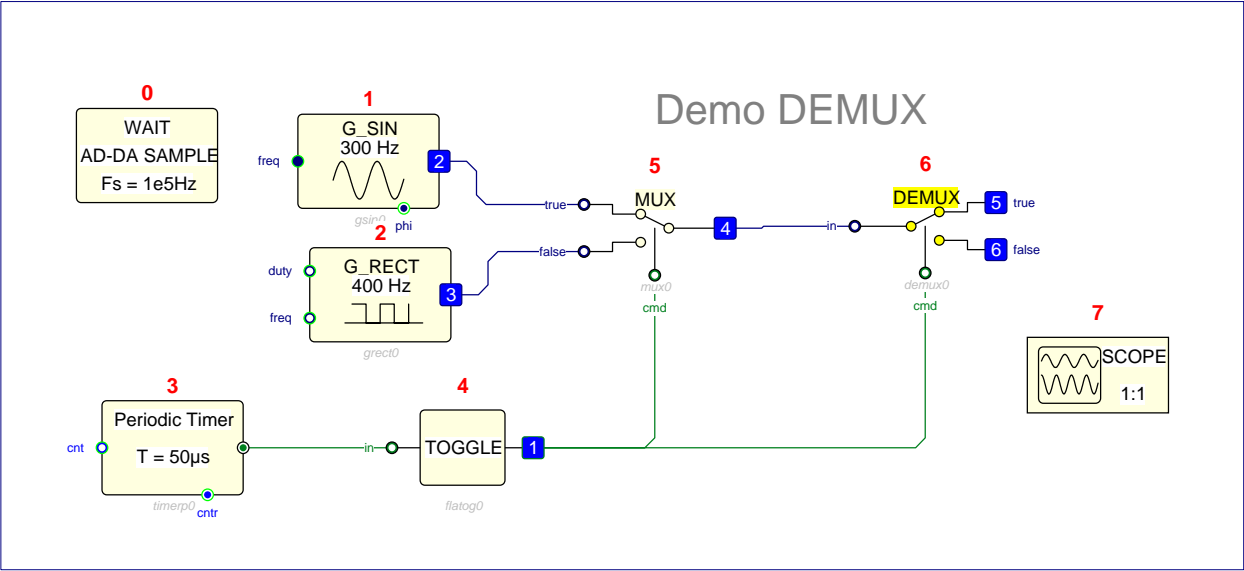
OUTPUTS

Name:
name_true
name_false

Data Type:
FRACT
FRACT

Data Struct:
WORD
WORD

Connection:
normal
normal



DEMUX test program

DERIV

Instant Derivator

DERIV



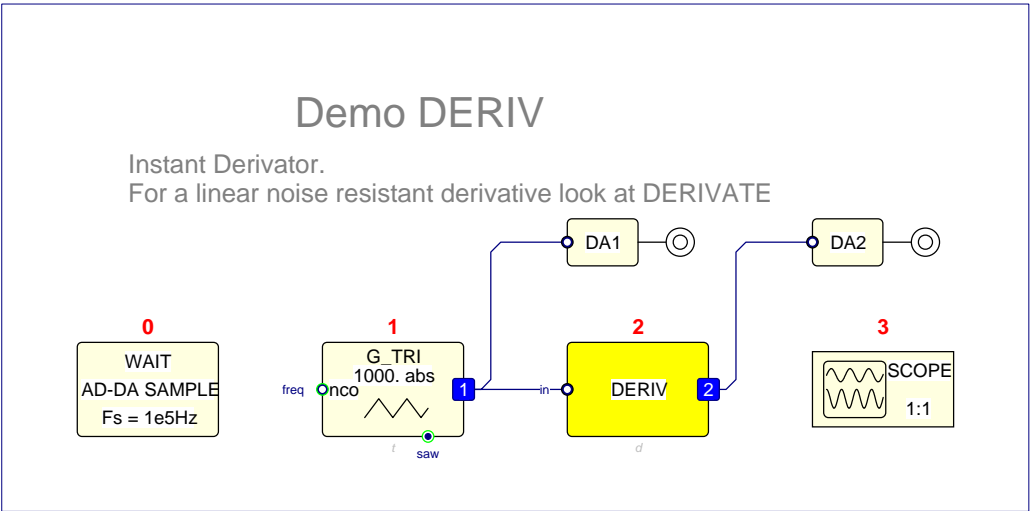
CATEGORY: CONTROL

DESCRIPTION:
Instant Derivator
with input gain
 $y(k) = g \cdot (x(k) - x(k-1))$

PARAMETERS:
Parameter: gain
Default values: 0.001

INPUTS			
<i>Name:</i> name_in	<i>Data Type:</i> FRACT	<i>Data Struct:</i> WORD	<i>Connection:</i> mandatory

OUTPUTS			
<i>Name:</i> name	<i>Data Type:</i> FRACT	<i>Data Struct:</i> WORD	<i>Connection:</i> normal

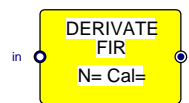


DERIV test program

DERIVATE

Derivator FIR Filter

DERIVATE



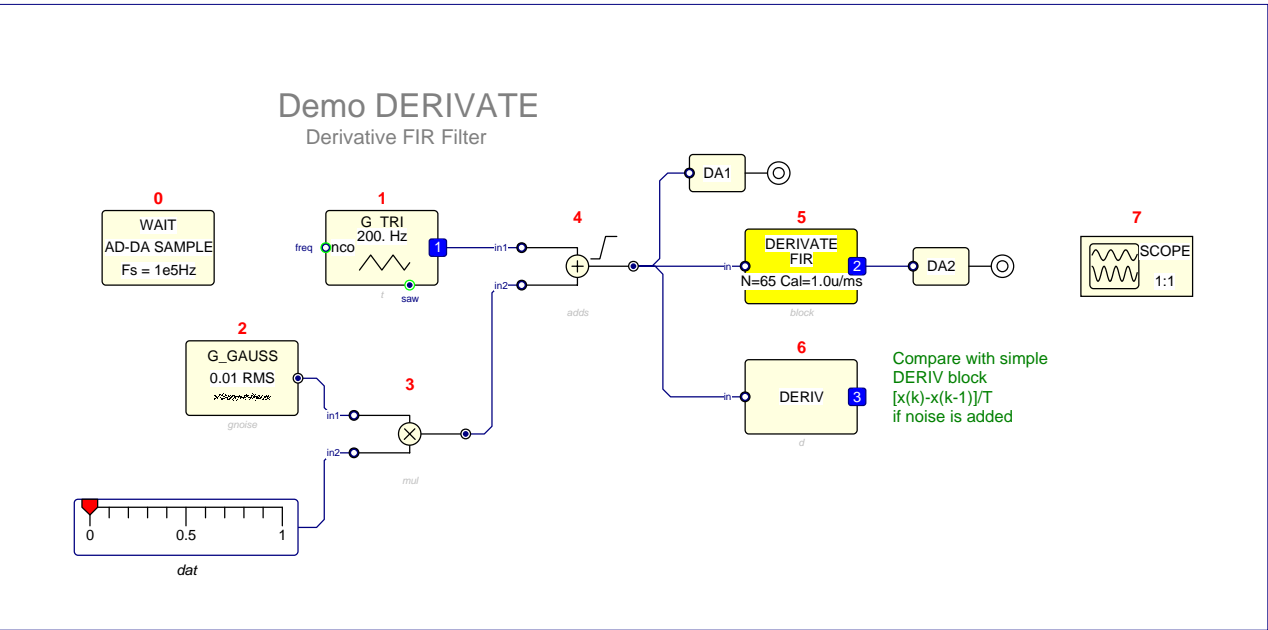
CATEGORY: CONTROL

DESCRIPTION:
Derivator FIR Filter
 $h(n) = a[(N-1)/2-n]$, $n=0..N-1$

PARAMETERS:
Parameter: Default values:
Size 101
Calibration: 1.0 is 1.0
Unit u/s,u/ms,u/smp

INPUTS
Name: Data Type: Data Struct: Connection:
name_in FRAC T WORD mandatory

OUTPUTS
Name: Data Type: Data Struct: Connection:
name FRAC T WORD normal

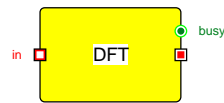


DERIVATE test program

DFT

Discrete Fourier Transform

DFT



CATEGORY: MATRIX

DESCRIPTION:
Discrete Fourier Transform

INPUTS

Name:
name_in

Data Type:
COMPLEX

Data Struct:
Matrix of DWORD

Connection:
mandatory

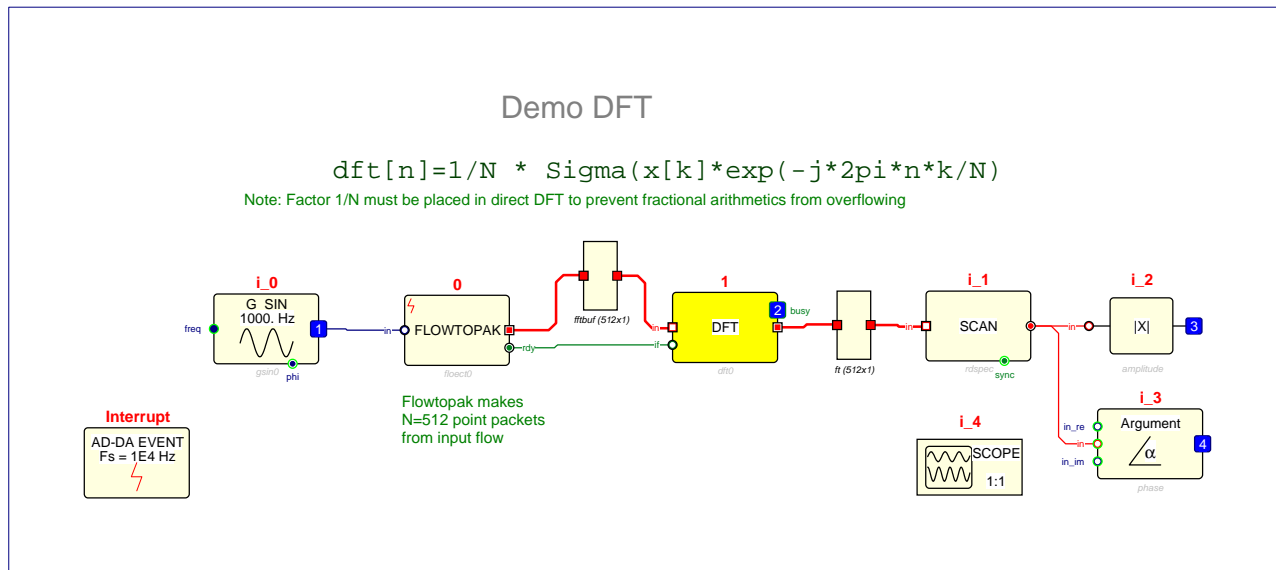
OUTPUTS

Name:
name
name_busy

Data Type:
COMPLEX
BOOL

Data Struct:
Matrix of DWORD
BIT

Connection:
normal
optional

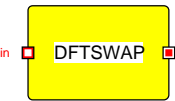


DFT test program

DFTSWAP

Swap Input Buffer halves

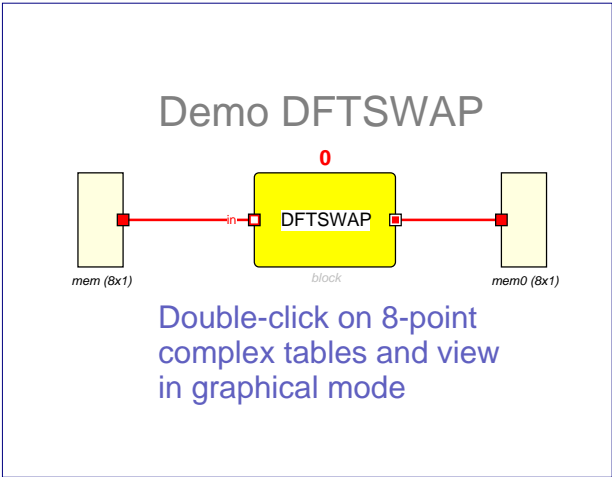
DFTSWAP



CATEGORY: MATRIX

DESCRIPTION:
Swap Input Buffer halves
Shows FFT and DFT with 0 centered

INPUTS			
Name:			
name_in	Data Type: COMPLEX	Data Struct: Matrix of DWORD	Connection: mandatory
OUTPUTS			
Name:			
name	Data Type: COMPLEX	Data Struct: Matrix of DWORD	Connection: normal

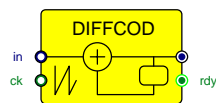


DFTSWAP test program

DIFFCOD

Differential coder

DIFFCOD



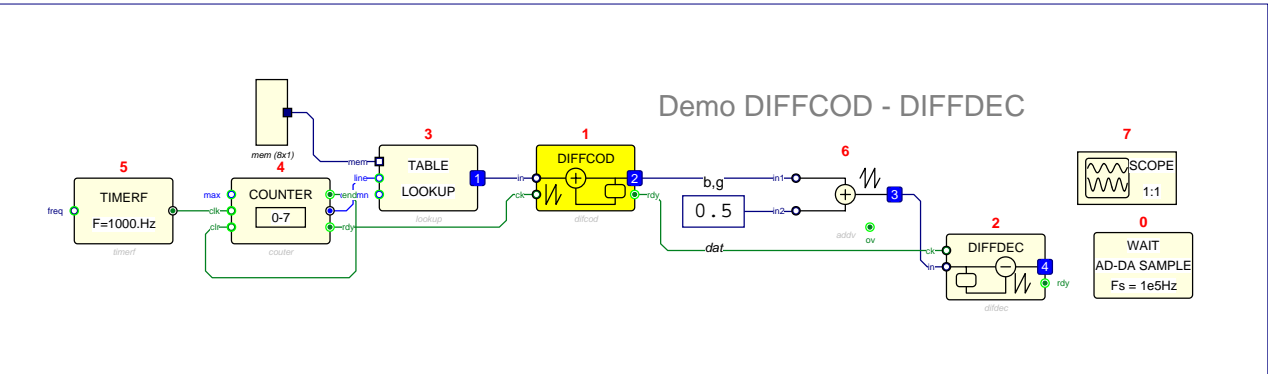
CATEGORY: TELECOM

DESCRIPTION:
Differential coder

PARAMETERS:
Parameter: Bit per symbol
Default values: 1

INPUTS	Data Type:	Data Struct:	Connection:
<i>Name:</i> name_ck	BOOL	BIT	mandatory
<i>Name:</i> name_in	FRACT	WORD	mandatory

OUTPUTS	Data Type:	Data Struct:	Connection:
<i>Name:</i> name_rdy	FRACT	WORD	normal
<i>Name:</i> name_rdy	BOOL	BIT	optional

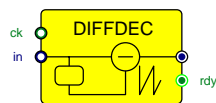


DIFFCOD test program

DIFFDEC

Differential decoder

DIFFDEC



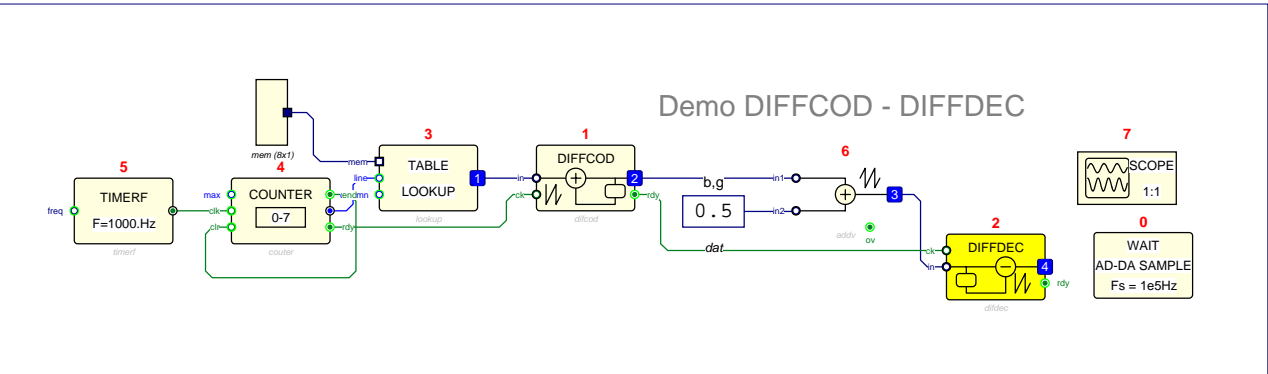
CATEGORY: TELECOM

DESCRIPTION:
Differential decoder

PARAMETERS:
Parameter: Bit per symbol
Default values: 1

INPUTS	Data Type:	Data Struct:	Connection:
<i>Name:</i> name_ck	BOOL	BIT	mandatory
<i>Name:</i> name_in	FRACT	WORD	mandatory

OUTPUTS	Data Type:	Data Struct:	Connection:
<i>Name:</i> name_rdy	FRACT	WORD	normal
<i>Name:</i> name_rdy	BOOL	BIT	optional



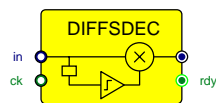
DIFFDEC test program

DIFFSCOD

DIFFSDEC

Differential sign decoder

DIFFSDEC



CATEGORY: TELECOM

DESCRIPTION:
Differential sign decoder
 $y(k) = x(k) * \text{sgn}(x(k-1))$

INPUTS

Name:
name_ck
name_in

Data Type:
BOOL
FRACT

Data Struct:
BIT
WORD

Connection:
mandatory
mandatory

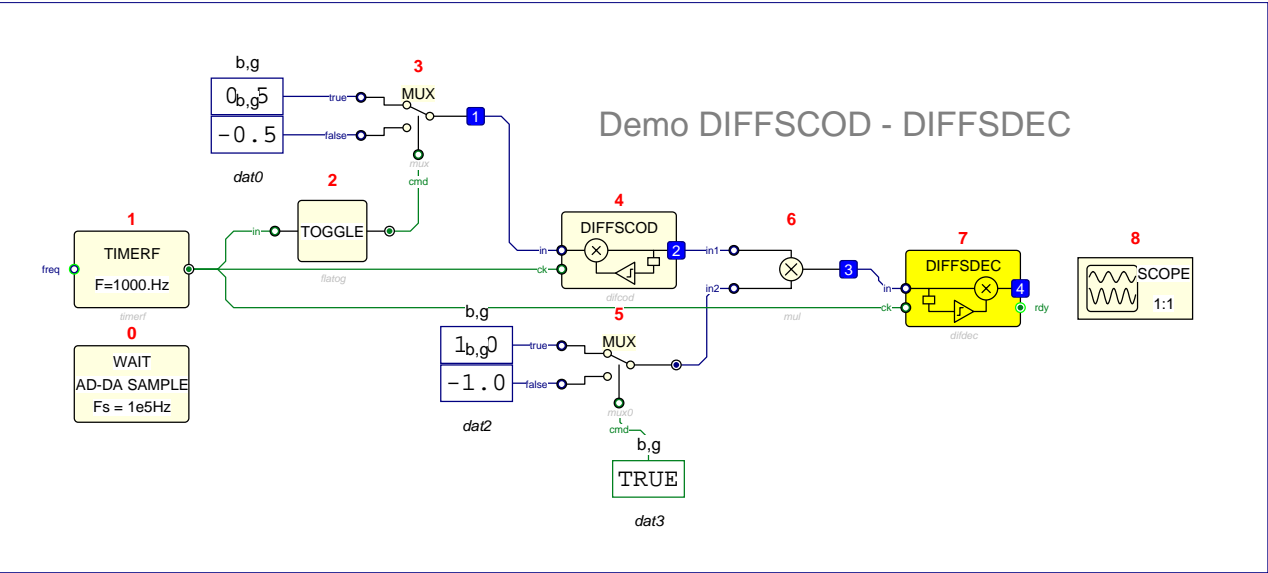
OUTPUTS

Name:
name
name_rdy

Data Type:
FRACT
BOOL

Data Struct:
WORD
BIT

Connection:
normal
optional

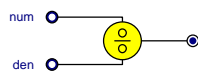


DIFFSDEC test program

DIVIDE

Fractionnal division num/den

DIVIDE



CATEGORY: ARITHMETIC

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

INPUTS

Name:
name_num
name_den

Data Type:
FRACT
FRACT

Data Struct:
WORD
WORD

Connection:
mandatory
mandatory

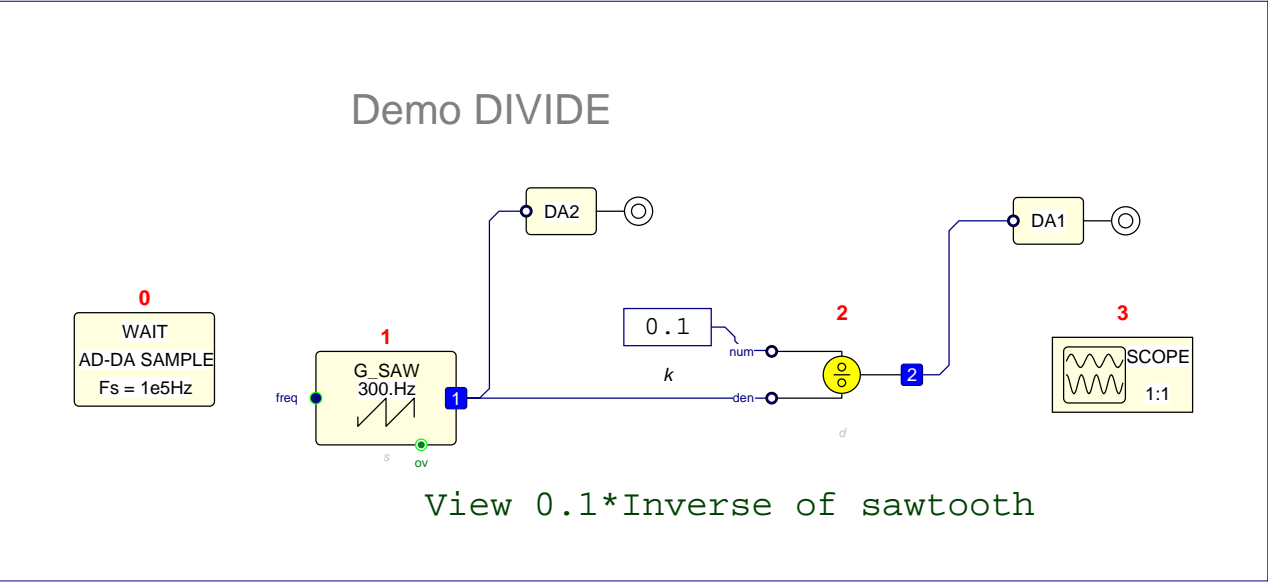
OUTPUTS

Name:
name

Data Type:
FRACT

Data Struct:
WORD

Connection:
normal

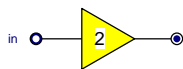


DIVIDE test program

DOUBLE

Gain by 2

DOUBLE



CATEGORY: ARITHMETIC

DESCRIPTION:
Gain by 2

INPUTS
Name:
name_in

Data Type:
FRACT

Data Struct:
WORD

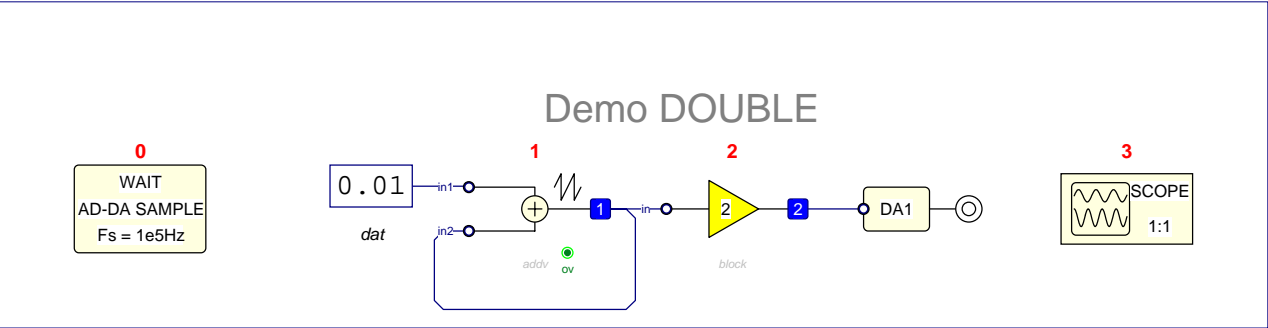
Connection:
mandatory

OUTPUTS
Name:
name

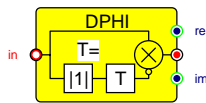
Data Type:
FRACT

Data Struct:
WORD

Connection:
normal



DOUBLE test program



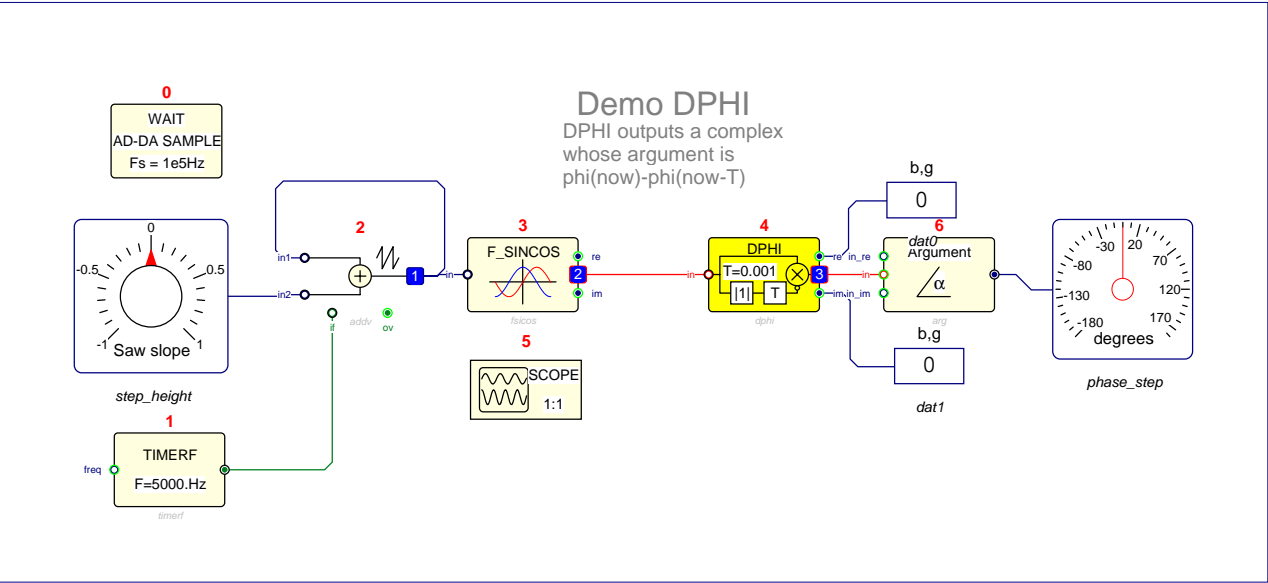
CATEGORY: TELECOM

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

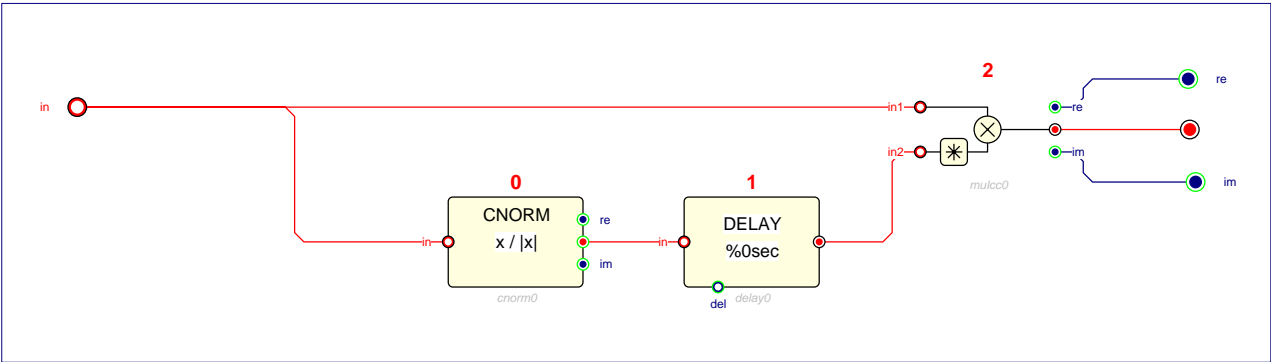
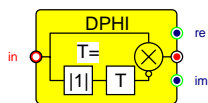
PARAMETERS:
Parameter: Delay
Default values: 0.001

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

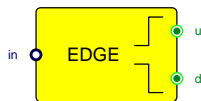


DPHI internal schema

EDGE

Generate flags on zero crossing

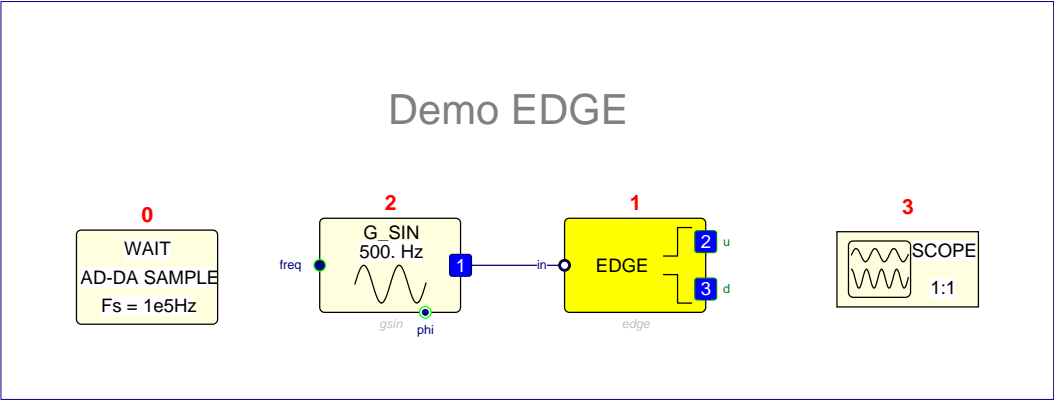
EDGE



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

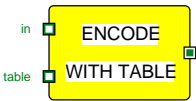


EDGE test program

ENCODE

GF(2) encoder

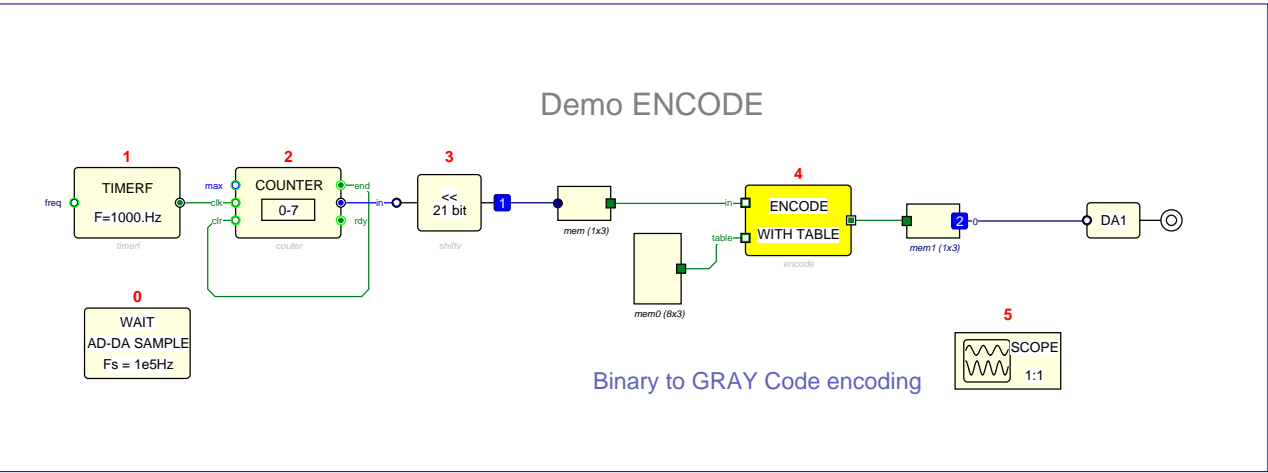
ENCODE



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

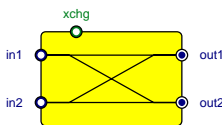


ENCODE test program

EXCHG

Exchange

EXCHG



CATEGORY: CONTROL

DESCRIPTION:

Exchange
If xchg TRUE then Out1=In2 and Out2=In1
else Out1=In1 and Out2=In2

INPUTS

Name:
name_in1
name_in2
name_xchg

Data Type:
FRACT
FRACT
BOOL

Data Struct:
WORD
WORD
BIT

Connection:
mandatory
mandatory
mandatory

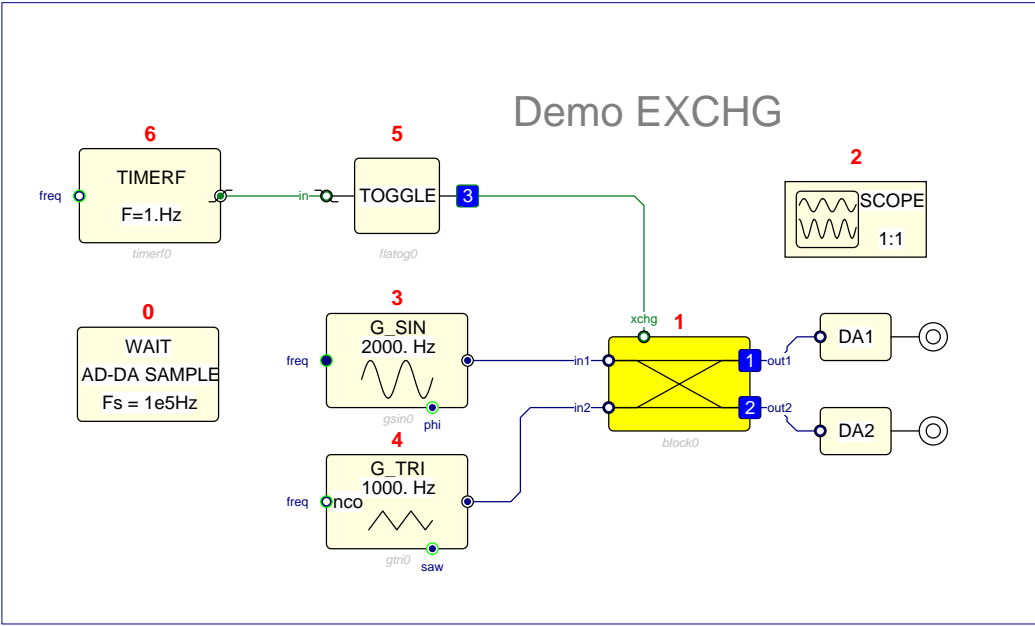
OUTPUTS

Name:
name_out1
name_out2

Data Type:
FRACT
FRACT

Data Struct:
WORD
WORD

Connection:
normal
normal



EXCHG test program

EXPAND

Restore compressed data

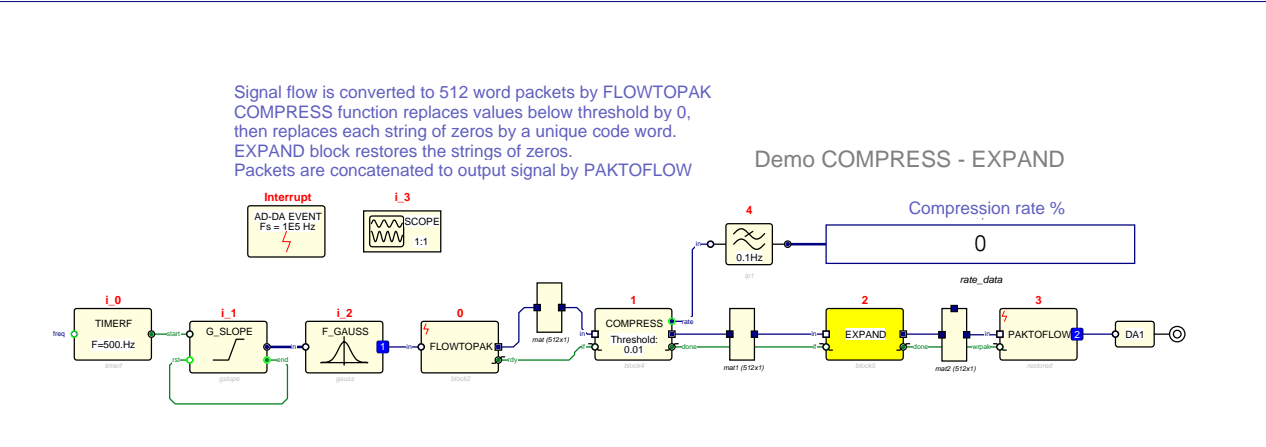
EXPAND



CATEGORY: MATRIX

DESCRIPTION:
Restore compressed data
Codes in the form of \$80nnnn are replaced by a string of nnnn zeros

INPUTS			
Name:	Data Type:	Data Struct:	Connection:
name_in	FRACT	Matrix of WORD	mandatory
OUTPUTS			
Name:	Data Type:	Data Struct:	Connection:
name	FRACT	Matrix of WORD	normal

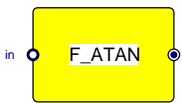


EXPAND test program

F_ATAN

Arc Tangent

F_ATAN



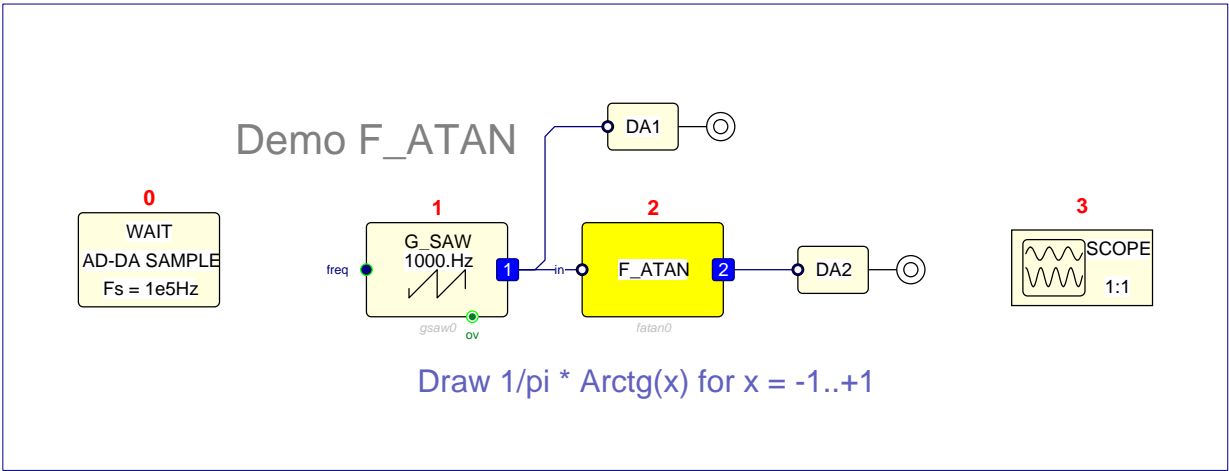
CATEGORY: FUNCTIONS

DESCRIPTION:
Arc Tangent
Only for input between -1 and +1
 $y = 1/\pi * \arctan(x)$

PARAMETERS:
Parameter: points
Default values: 15

INPUTS			
<i>Name:</i> name_in	<i>Data Type:</i> FRACT	<i>Data Struct:</i> WORD	<i>Connection:</i> mandatory

OUTPUTS			
<i>Name:</i> name	<i>Data Type:</i> FRACT	<i>Data Struct:</i> WORD	<i>Connection:</i> normal

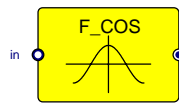


F_ATAN test program

F_COS

Cosine function $y = \text{Cos}(\pi \cdot x)$

F_COS



CATEGORY: FUNCTIONS

DESCRIPTION:

Cosine function $y = \text{Cos}(\pi \cdot x)$

Input in: angle in half turns ($-1..+1 \rightarrow -\pi ..+\pi$ rad)

INPUTS

Name:
name_in

Data Type:
FRACT

Data Struct:
WORD

Connection:
mandatory

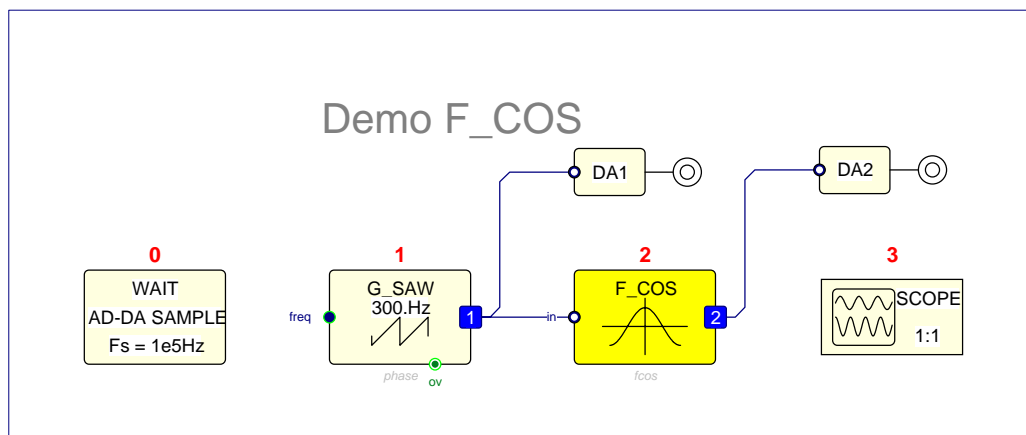
OUTPUTS

Name:
name

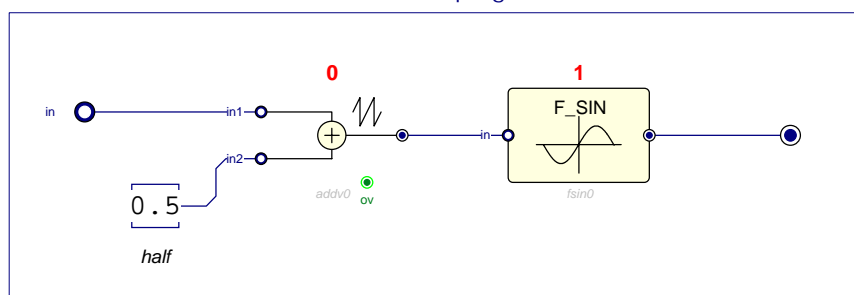
Data Type:
FRACT

Data Struct:
WORD

Connection:
normal



F_COS test program

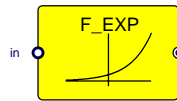


F_COS internal schema

F_EXP

Real exponential function

F_EXP



CATEGORY: FUNCTIONS

DESCRIPTION:
Real exponential function
 $y = 2^{k \cdot x} / 2^k$

PARAMETERS:

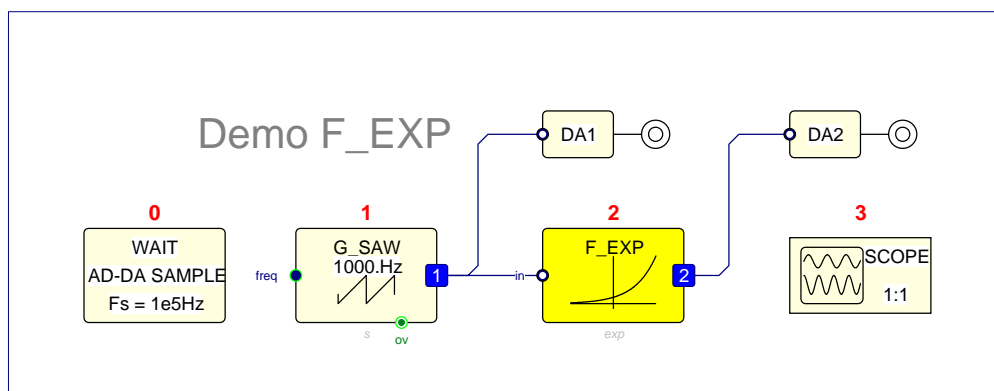
<i>Parameter:</i>	<i>Default values:</i>
k	3.5
Points	21

INPUTS

<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name_in	FRACT	WORD	mandatory

OUTPUTS

<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name	FRACT	WORD	normal

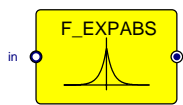


F_EXP test program

F_EXPABS

Exponential of abs

F_EXPABS



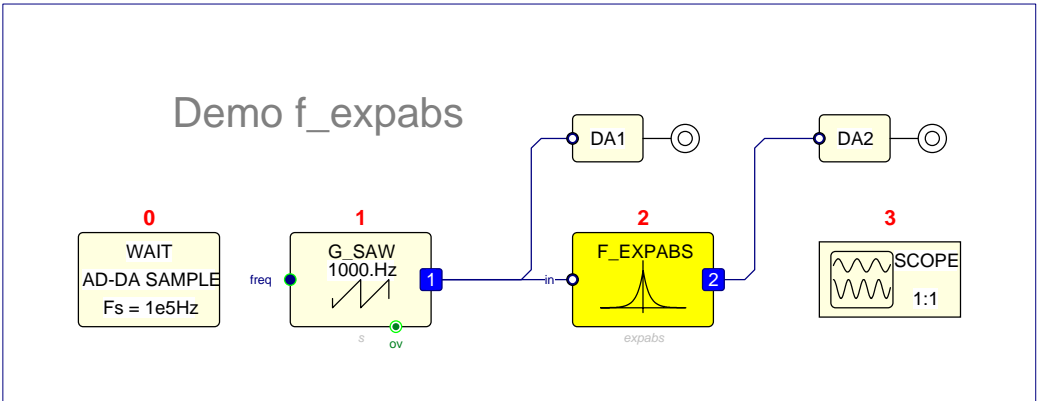
CATEGORY: FUNCTIONS

DESCRIPTION:
Exponential of abs
Real exponential of -abs(input) function $y = 2^{-|k \cdot x|}$

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

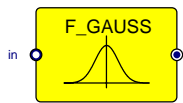


F_EXPABS test program

F_GAUSS

Gaussian function

F_GAUSS



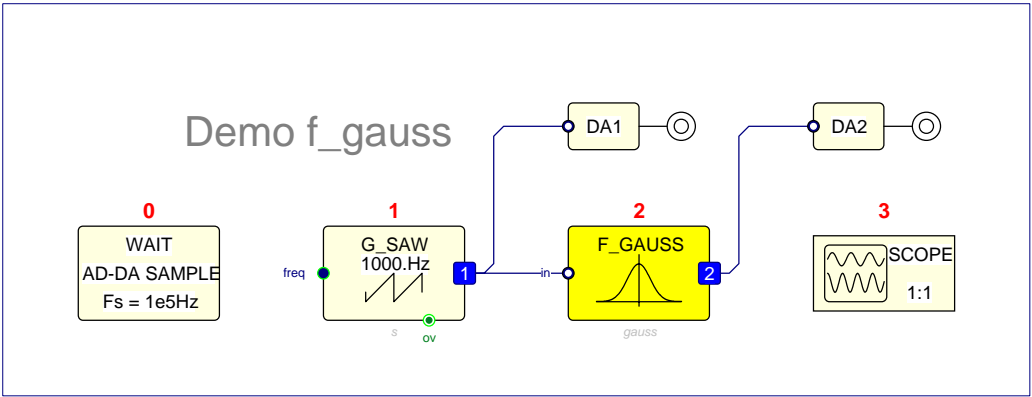
CATEGORY: FUNCTIONS

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

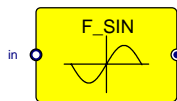


F_GAUSS test program

F_SIN

Sine function $y = \sin(\pi \cdot x)$

F_SIN



CATEGORY: FUNCTIONS

DESCRIPTION:

Sine function $y = \sin(\pi \cdot x)$

Input in: angle in half turns (-1..+1 -> -pi ..+pi rad)

INPUTS

Name:
name_in

Data Type:
FRACT

Data Struct:
WORD

Connection:
mandatory

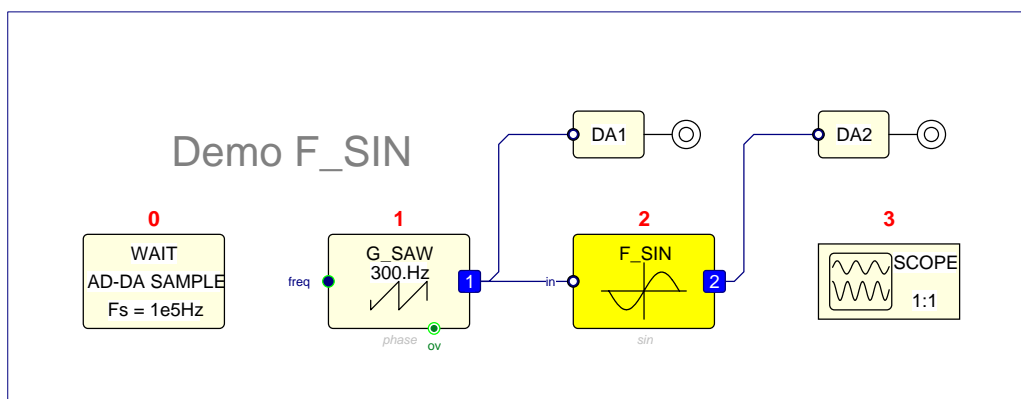
OUTPUTS

Name:
name

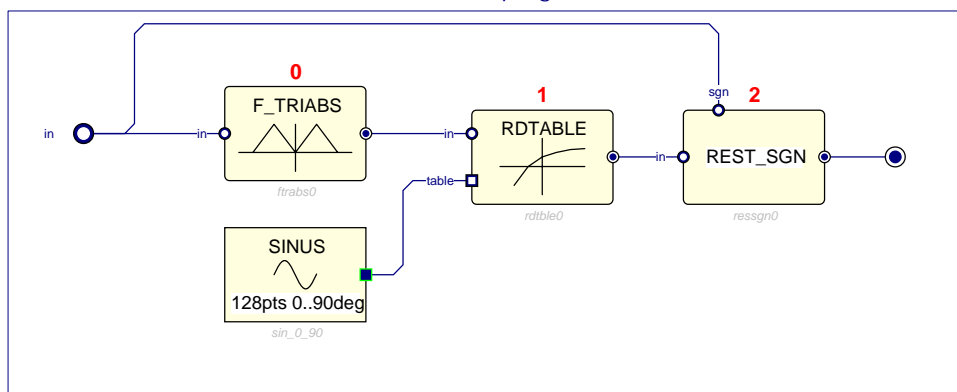
Data Type:
FRACT

Data Struct:
WORD

Connection:
normal



F_SIN test program

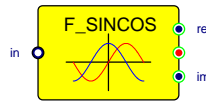


F_SIN internal schema

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:
name_in

Data Type:
FRACT

Data Struct:
WORD

Connection:
mandatory

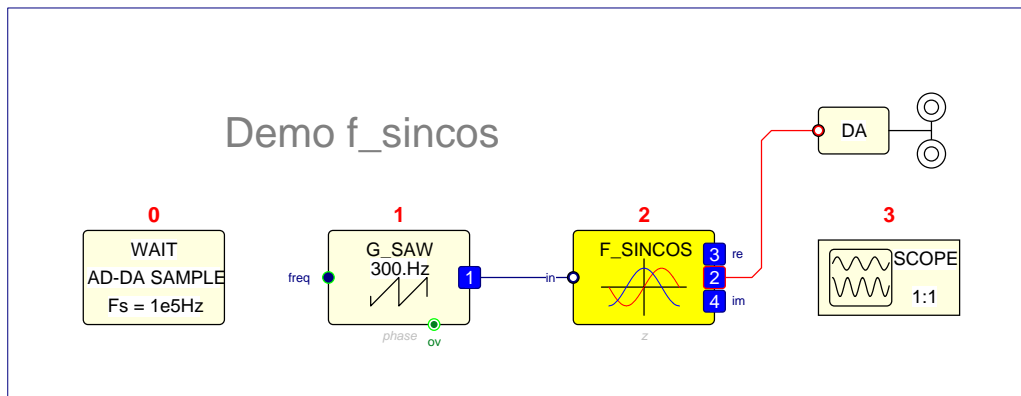
OUTPUTS

Name:
name_re
name
name_im

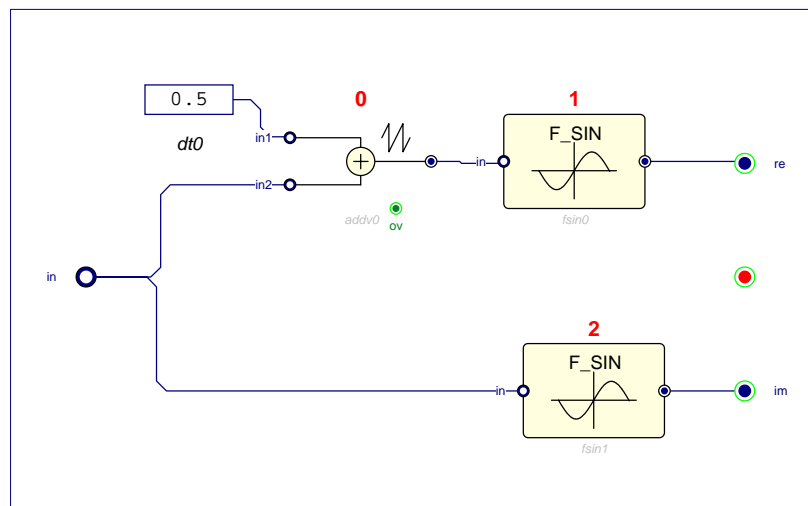
Data Type:
FRACT
COMPLEX
FRACT

Data Struct:
WORD
WORD
WORD

Connection:
optional
optional
optional



F_SINCOS test program

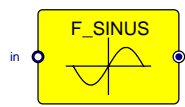


F_SINCOS internal schema

F_SINUS

Sine Function

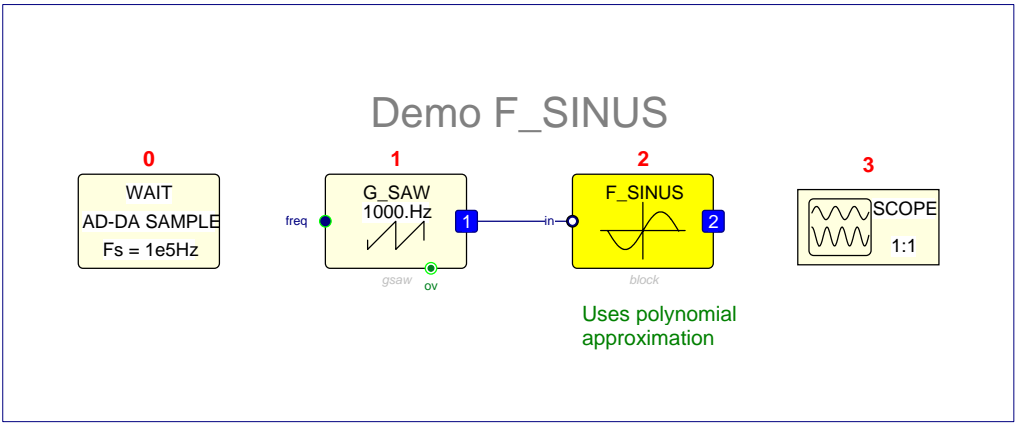
F_SINUS



CATEGORY: FUNCTIONS

DESCRIPTION:
Sine Function
 $y = \sin(\pi \cdot x)$
Polynomial approximation

INPUTS			
Name:	Data Type:	Data Struct:	Connection:
name_in	FRACT	WORD	mandatory
OUTPUTS			
Name:	Data Type:	Data Struct:	Connection:
name	FRACT	WORD	normal

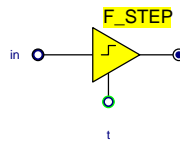


F_SINUS test program

F_STEP

Step function

F_STEP



CATEGORY: NON LINEAR

DESCRIPTION:

Step function

$y = \text{value left if } x < \text{threshold};$

$y = \text{value right if } x \geq \text{threshold};$

PARAMETERS:

Parameter:

Threshold

Value left

Value right

Default values:

0

0

1.0

INPUTS

Name:

name_in

name_t

Data Type:

FRACT

FRACT

Data Struct:

WORD

WORD

Connection:

mandatory

optional

OUTPUTS

Name:

name

Data Type:

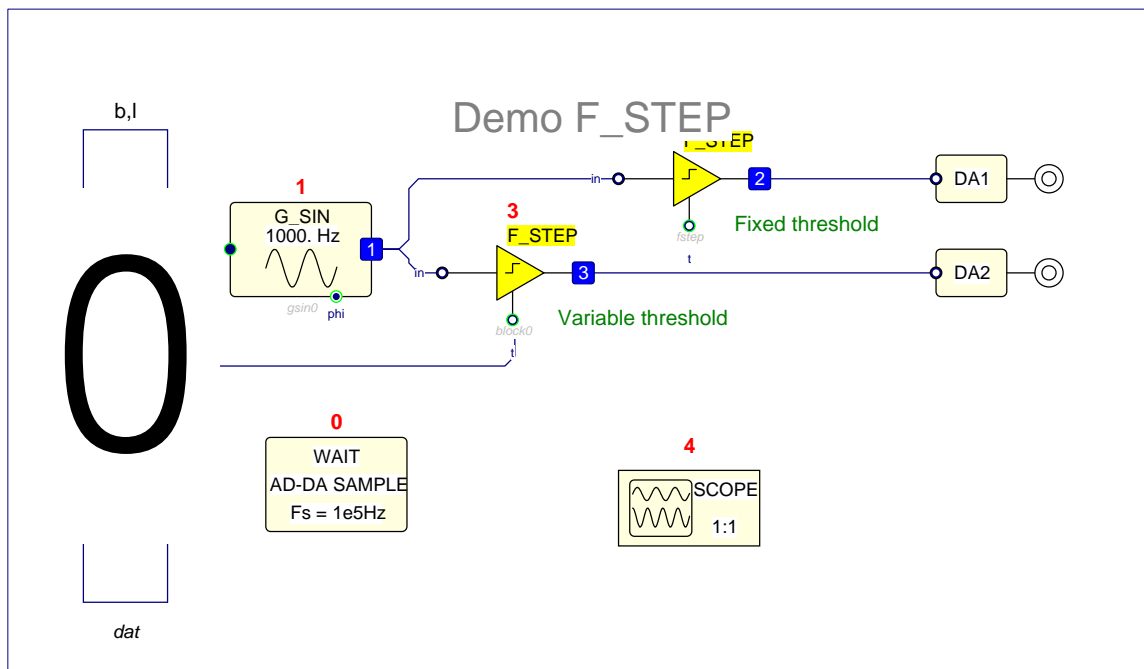
FRACT

Data Struct:

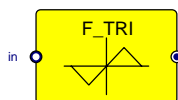
WORD

Connection:

normal



F_STEP test program



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:
name_in

Data Type:
FRACT

Data Struct:
WORD

Connection:
mandatory

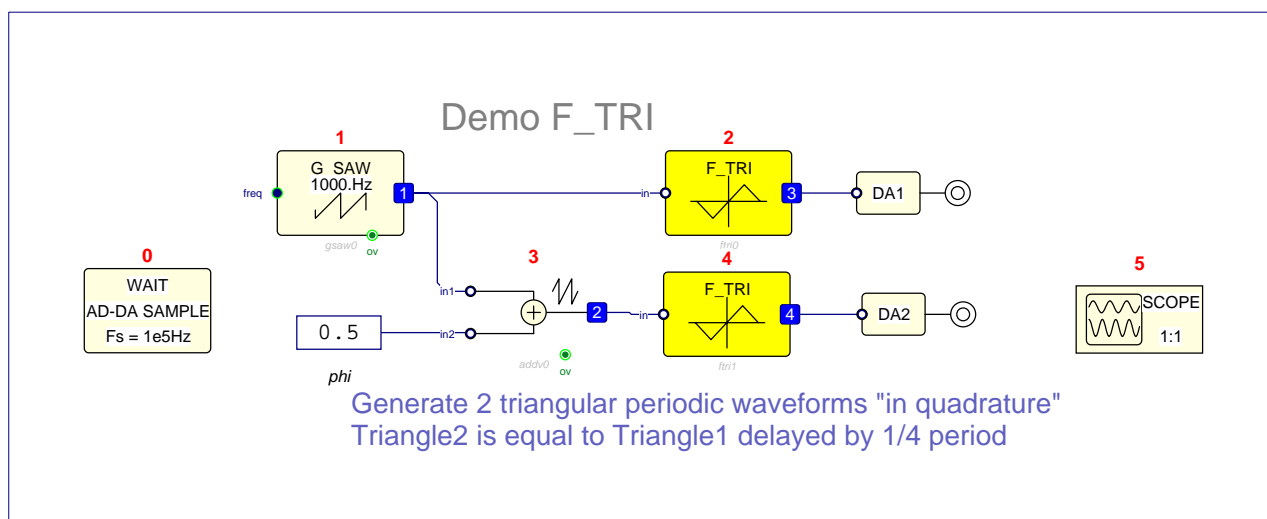
OUTPUTS

Name:
name

Data Type:
FRACT

Data Struct:
WORD

Connection:
normal

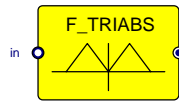


F_TRI test program

F_TRIABS

Abs value of Triangle function

F_TRIABS



DESCRIPTION:

Abs value of Triangle function

INPUTS

Name:
name_in

Data Type:
FRACT

Data Struct:
WORD

Connection:
mandatory

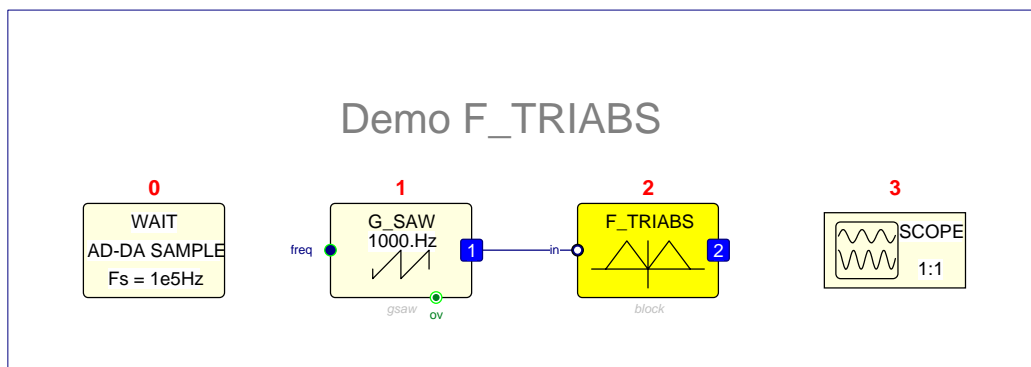
OUTPUTS

Name:
name

Data Type:
FRACT

Data Struct:
WORD

Connection:
normal

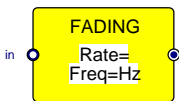


F_TRIABS test program

FADING

Simulate fading channel

FADING



CATEGORY: TELECOM

DESCRIPTION:
Simulate fading channel

PARAMETERS:
Parameter:
Frequency
Proportion

Default values:
0.1
0.5

INPUTS
Name:
name_in

Data Type:
FRACT

Data Struct:
WORD

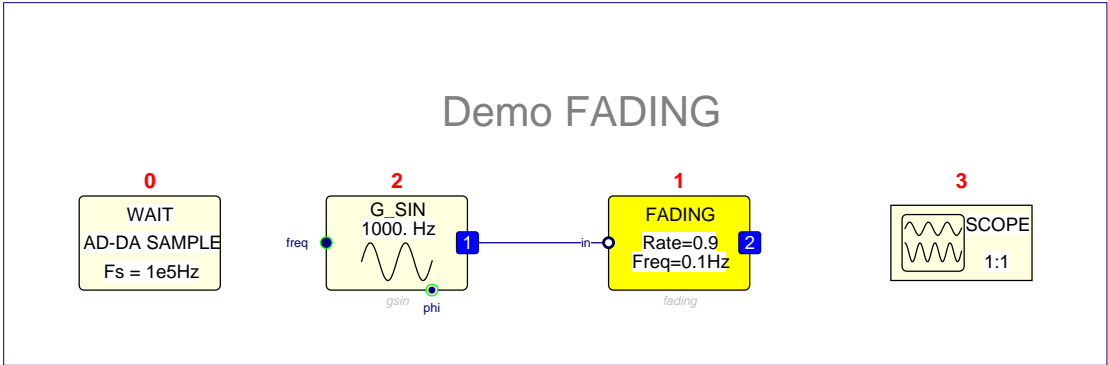
Connection:
mandatory

OUTPUTS
Name:
name

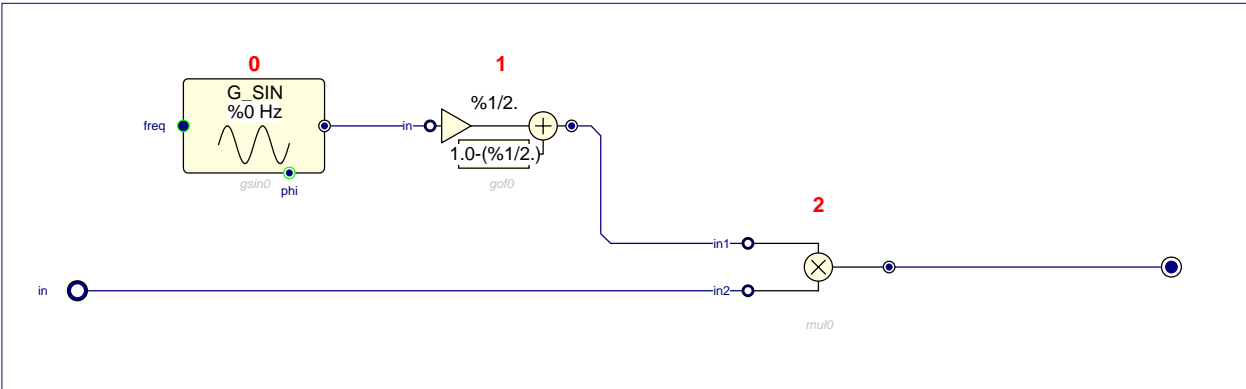
Data Type:
FRACT

Data Struct:
WORD

Connection:
normal



FADING test program



FADING internal schema

FFT

Fast Fourier Transform

FFT



CATEGORY: MATRIX

DESCRIPTION:

Fast Fourier Transform
Size is given by output vector

INPUTS

Name:
name_in

Data Type:
COMPLEX

Data Struct:
Matrix of DWORD

Connection:
mandatory

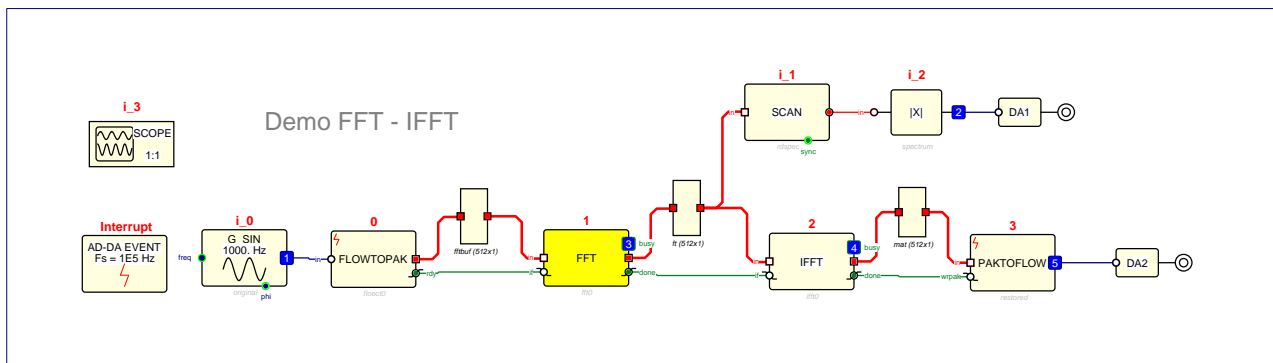
OUTPUTS

Name:
name
name_busy

Data Type:
COMPLEX
BOOL

Data Struct:
Matrix of DWORD
BIT

Connection:
normal
optional

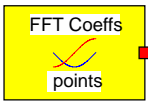


FFT test program

FFTCOEFF

FFT coefficient table

FFTCOEFF



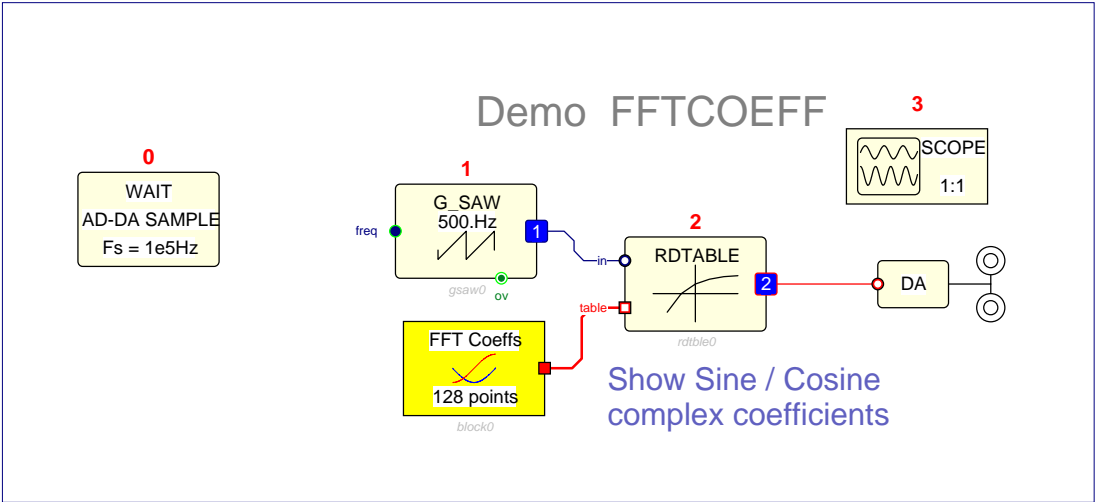
CATEGORY: MATRIX

DESCRIPTION:
FFT coefficient table
Sin Cos [0 ... -pi]
Size is 1/2 of FFT size

PARAMETERS:
Parameter: Size
Default values: 8,16,32,64,128,256,512,1024,2048,4096

OUTPUTS
Name: name
Data Type: COMPLEX
Data Struct: Matrix of DWORD
Connection: normal

ATTRIBUTES
Non executable, Data Table

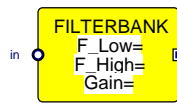


FFTCOEFF test program

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:

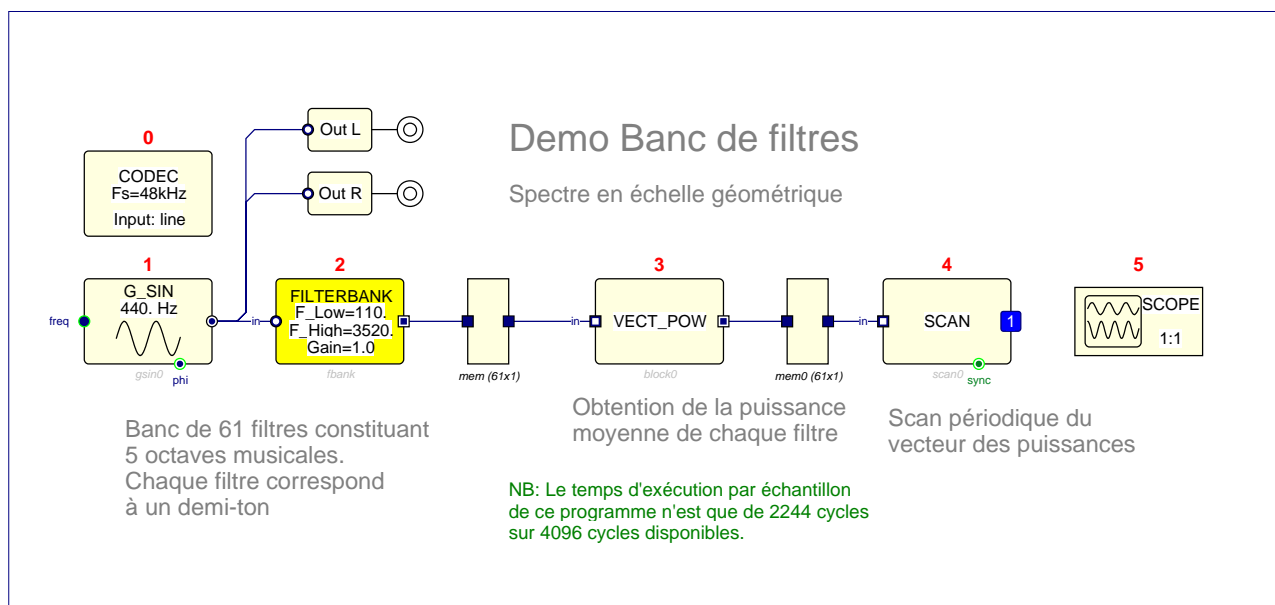
<i>Parameter:</i>	<i>Default values:</i>
Freq_Low	10.
Freq_High	1E4.
Gain	1.0

INPUTS

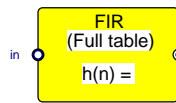
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name_in	FRACT	WORD	mandatory

OUTPUTS

<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name	FRACT	Matrix of WORD	normal



FILTERBANK test program



CATEGORY: FILTERS

DESCRIPTION:
Finite Impulse Response filter

PARAMETERS:

Parameter:
impulse response

Default values:
coeffs

INPUTS

Name:
name_in

Data Type:
FRACT

Data Struct:
WORD

Connection:
mandatory

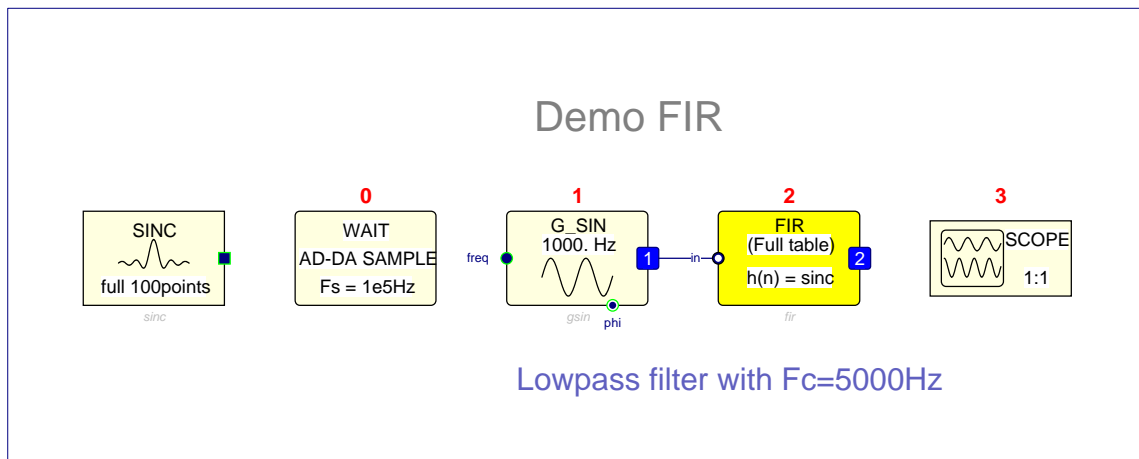
OUTPUTS

Name:
name

Data Type:
FRACT

Data Struct:
WORD

Connection:
normal

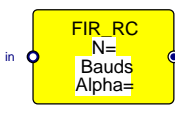


FIR test program

FIR_RC

Raised Cosine FIR

FIR_RC



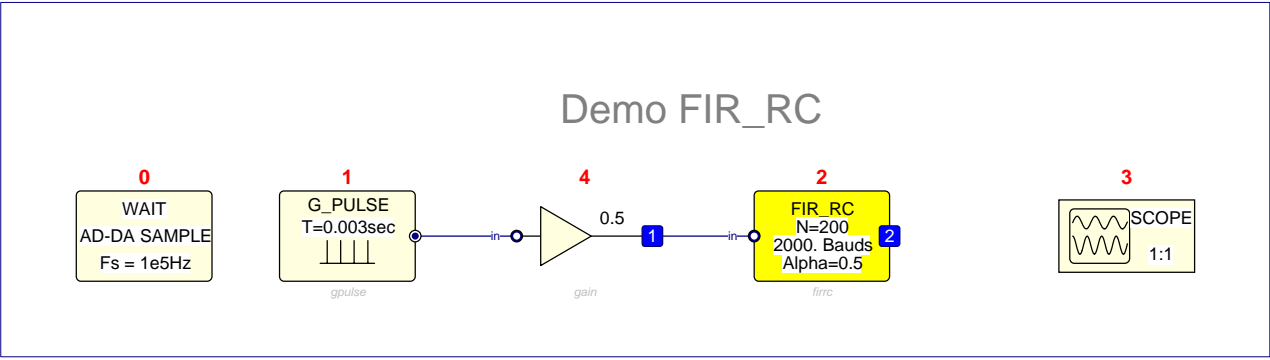
CATEGORY: TELECOM

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

PARAMETERS:	
<i>Parameter:</i>	<i>Default values:</i>
Size	200
Bauds	1000.
Alpha	0.5

INPUTS			
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name_in	FRACT	WORD	mandatory

OUTPUTS			
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name	FRACT	WORD	normal

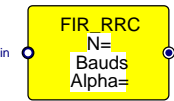


FIR_RC test program

FIR_RRC

Root Raised Cosine

FIR_RRC



CATEGORY: TELECOM

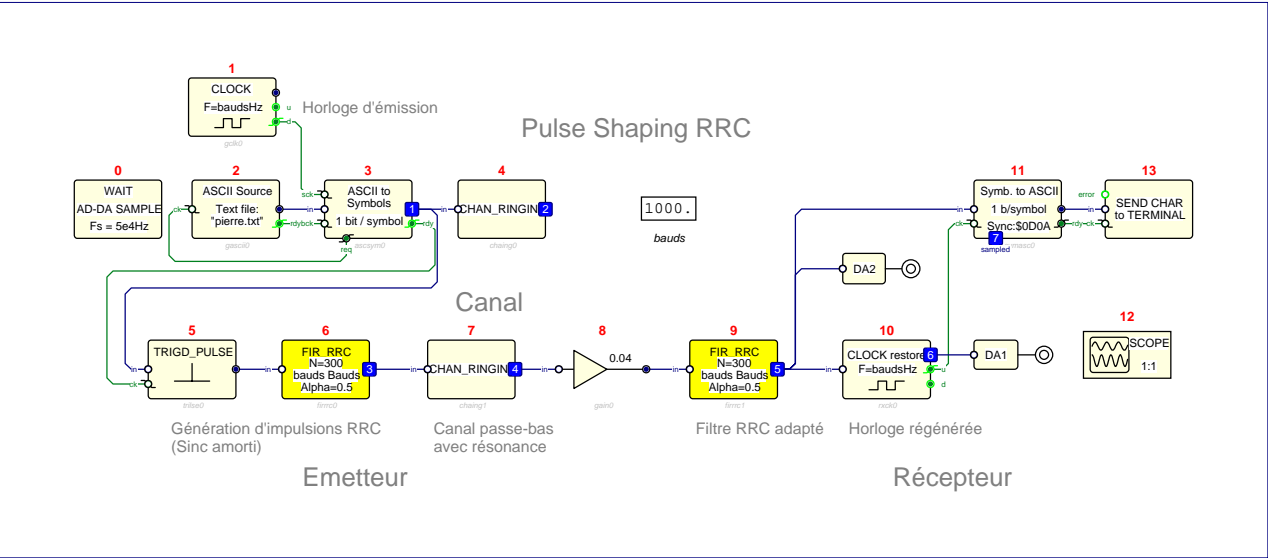
DESCRIPTION:
Root Raised Cosine
FIR filter which eliminates ISI in communications
Alpha = rolloff factor 0: (Sinc) BW=Bauds/2 1: BW=bauds

PARAMETERS:

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

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

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

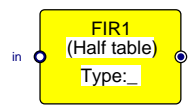


FIR_RRC test program

FIR1

Half sized FIR

FIR1



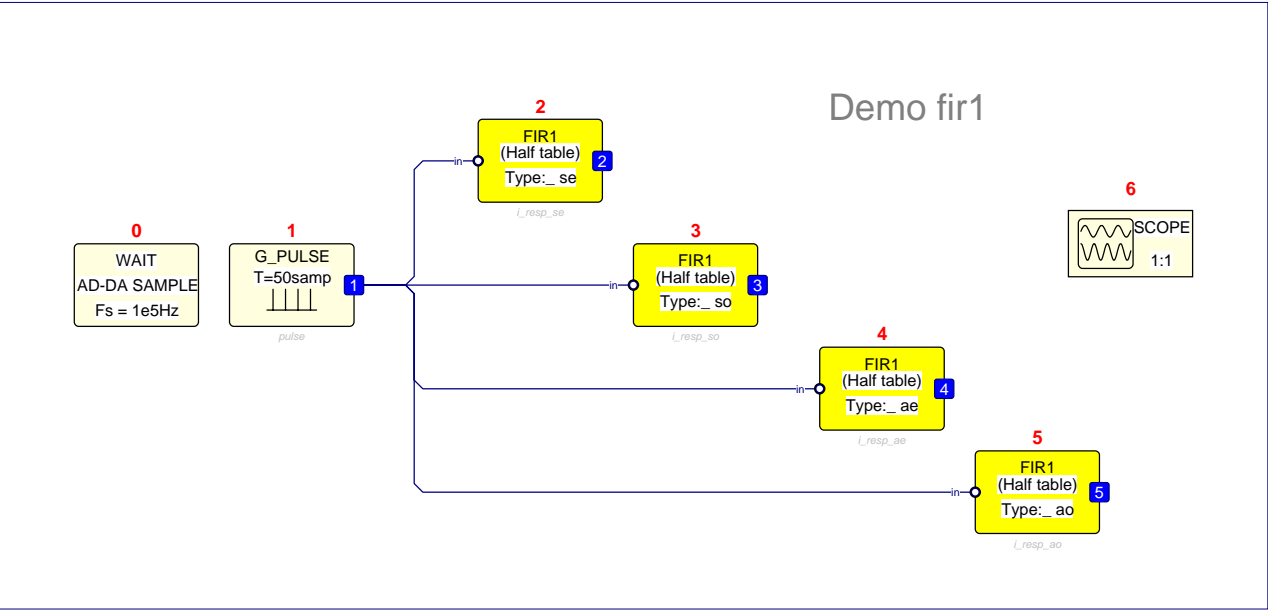
CATEGORY: FILTERS

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

INPUTS			
<i>Name:</i> name_in	<i>Data Type:</i> FRACT	<i>Data Struct:</i> WORD	<i>Connection:</i> mandatory

OUTPUTS			
<i>Name:</i> name	<i>Data Type:</i> FRACT	<i>Data Struct:</i> WORD	<i>Connection:</i> normal

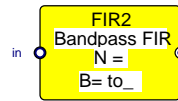


FIR1 test program

FIR2

Bandpass FIR filter

FIR2



CATEGORY: FILTERS

DESCRIPTION:

Bandpass FIR filter

For lowpass, do Freq low = 0

For highpass, do Freq high = $F_s/2$

PARAMETERS:

Parameter:

size
freq low
freq high
Unit

Default values:

500
1000.
2000.
Hz,kHz,MHz,Fs/2

INPUTS

Name:
name_in

Data Type:
FRACT

Data Struct:
WORD

Connection:
mandatory

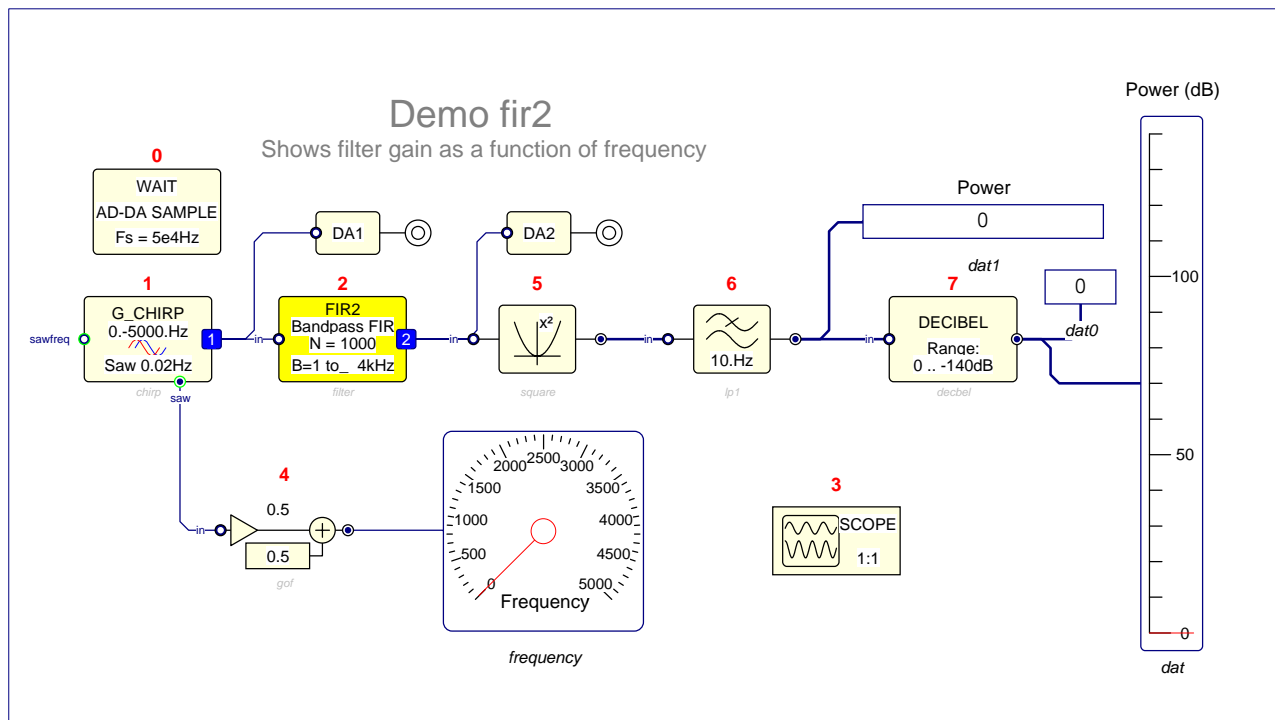
OUTPUTS

Name:
name

Data Type:
FRACT

Data Struct:
WORD

Connection:
normal



FIR2 test program

FIRG

Gaussian FIR filter

FIRG



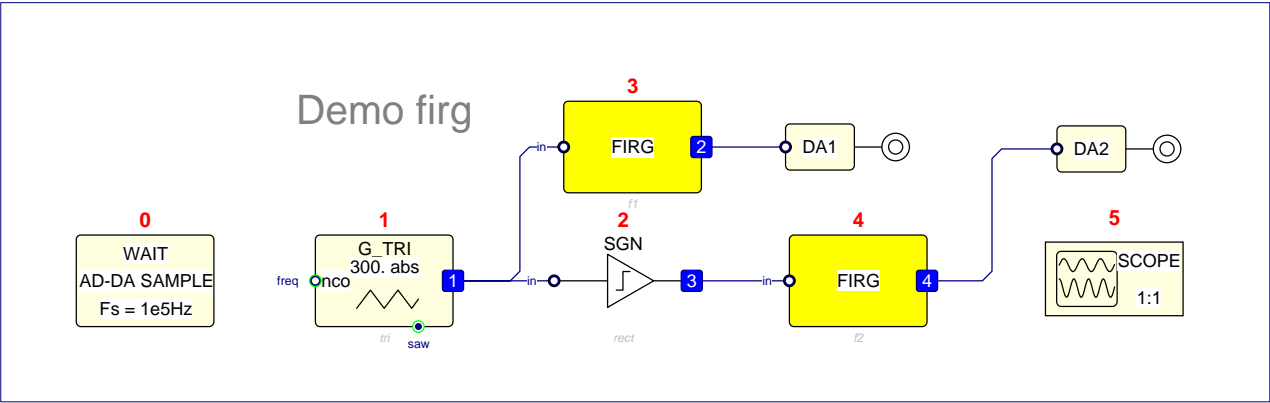
CATEGORY: FILTERS

DESCRIPTION:
Gaussian FIR filter
size represents 6 sigma

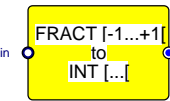
PARAMETERS:
Parameter: *Default values:*
Size 50

INPUTS			
Name:	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name_in	FRACT	WORD	mandatory

OUTPUTS			
Name:	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name	FRACT	WORD	normal



FIRG test program



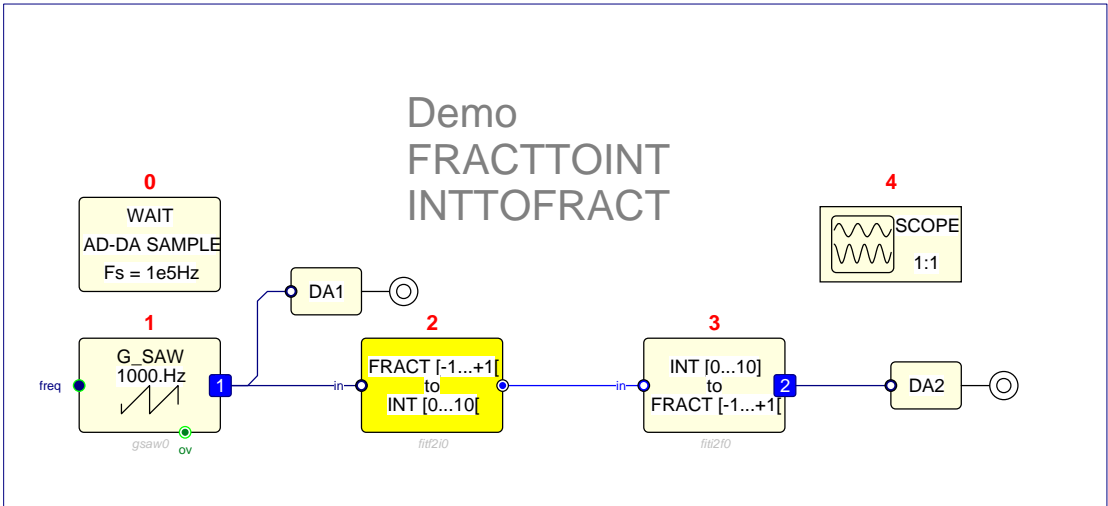
CATEGORY: INTEGER

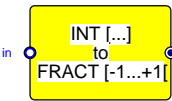
DESCRIPTION:
Fract to Integer
Converts Fract [-1.0..+1.0[--> Integer [min ...max[

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





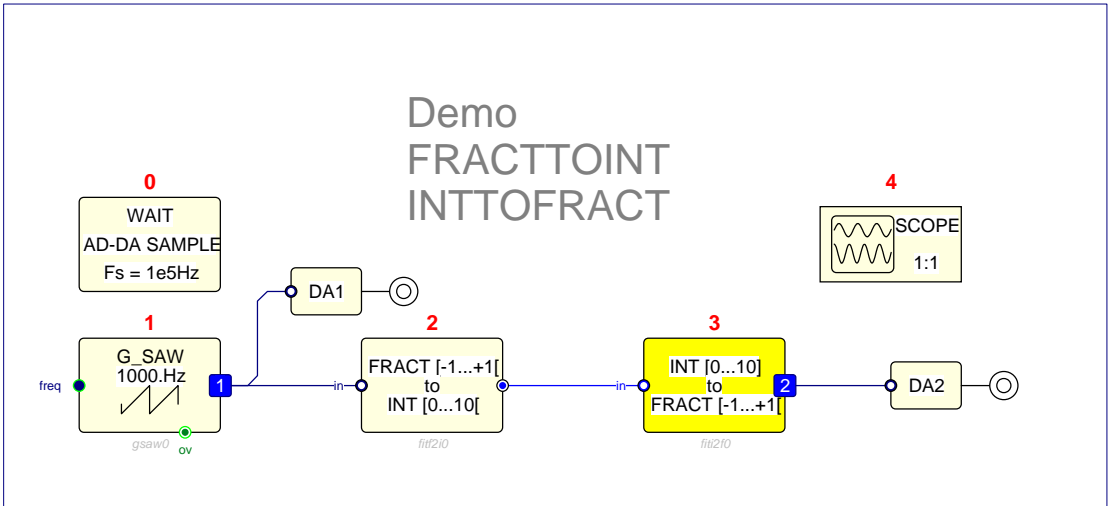
CATEGORY: INTEGER

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

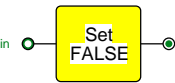


FITI2F test program

FLAGCLR

Set bool variable to FALSE

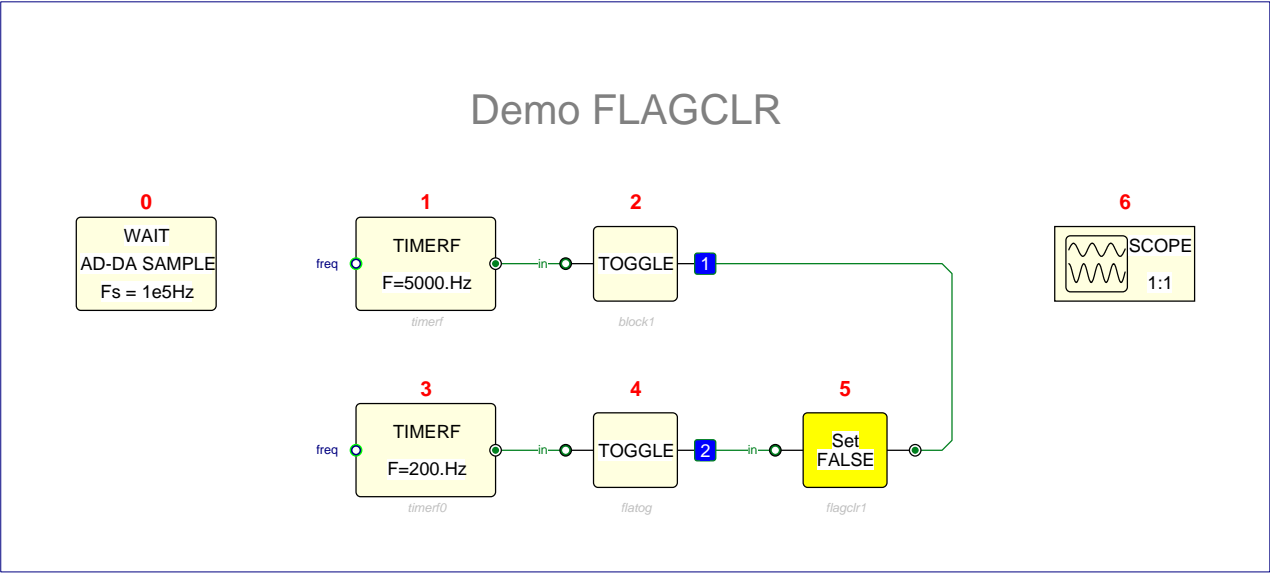
FLAGCLR



CATEGORY: LOGIC

DESCRIPTION:
Set bool variable 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

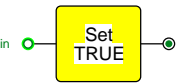


FLAGCLR test program

FLAGSET

Set bool variable to TRUE

FLAGSET



CATEGORY: LOGIC

DESCRIPTION:
Set bool variable to TRUE

INPUTS

Name:
name_in

Data Type:
BOOL

Data Struct:
BIT

Connection:
optional

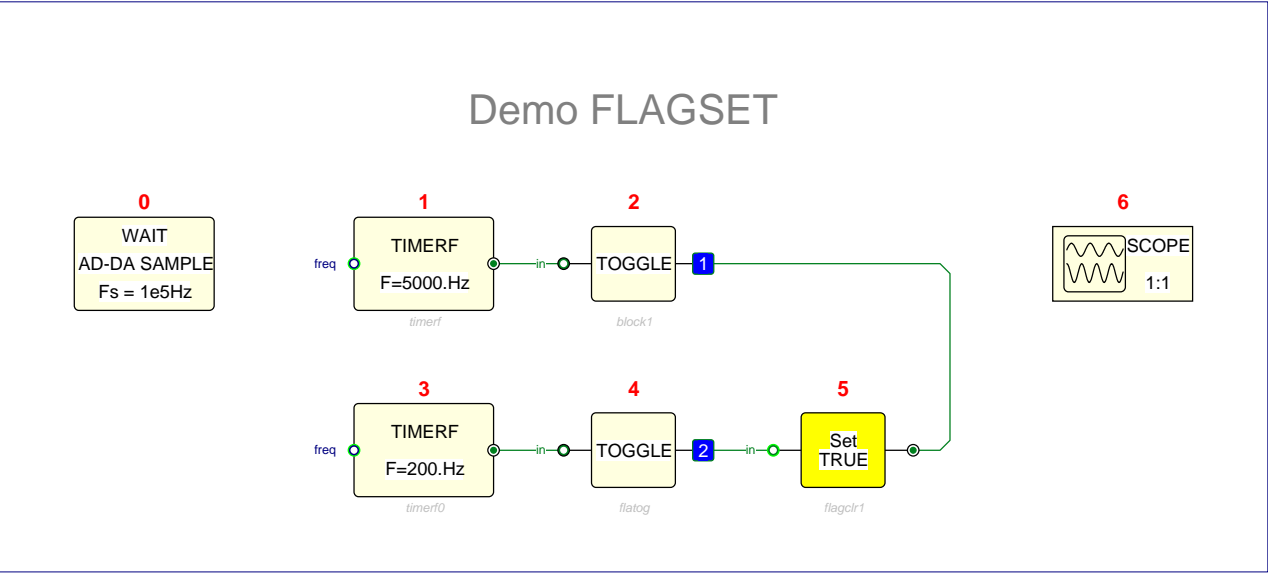
OUTPUTS

Name:
name

Data Type:
BOOL

Data Struct:
BIT

Connection:
normal

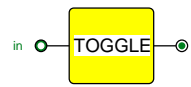


FLAGSET test program

FLAGTOG

Toggle boolean variable

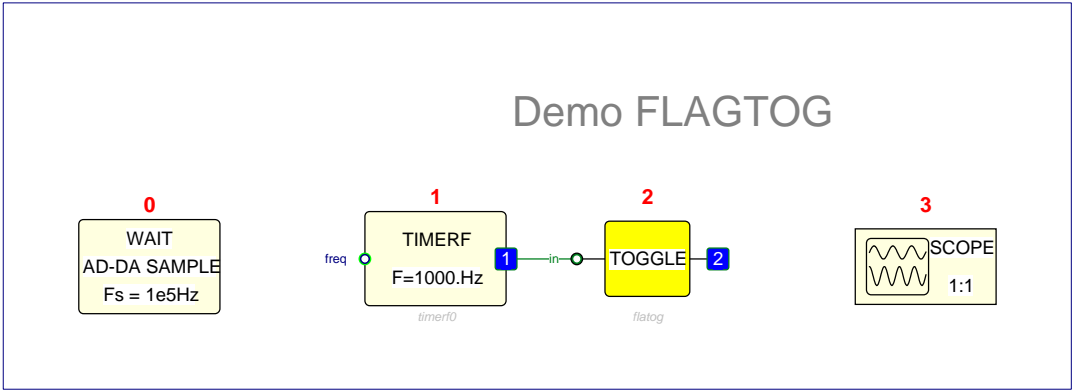
FLAGTOG



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



FLAGTOG test program

FLOWTOPAK

Flow to Packets

FLOWTOPAK



CATEGORY: MATRIX

DESCRIPTION:
Flow to Packets

PARAMETERS:
Parameter:
Buffer size
Packet size

Default values:
1024
512

INPUTS
Name:
name_in

Data Type:
defined by cn

Data Struct:

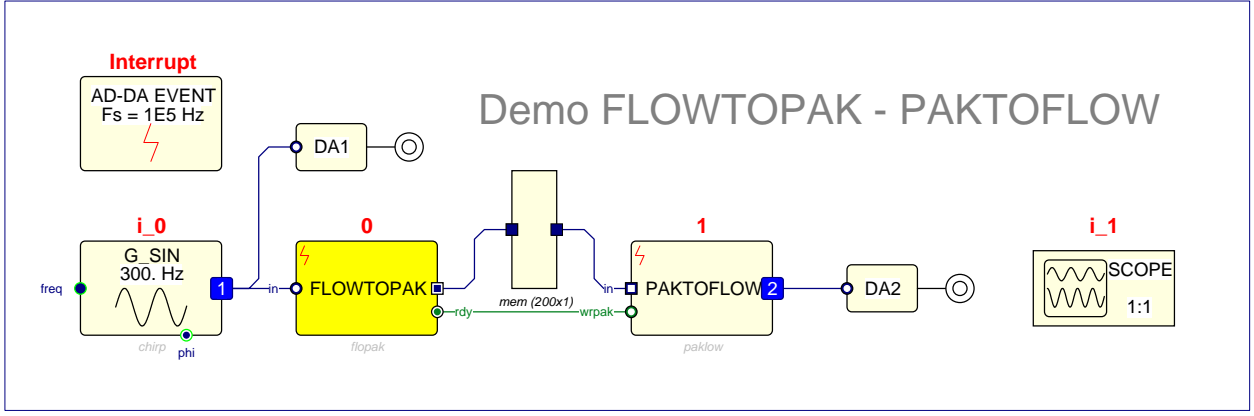
Connection:
mandatory

OUTPUTS
Name:
name_rdy

Data Type:
defined by cn
BOOL

Data Struct:
Matrix of
BIT

Connection:
normal
normal

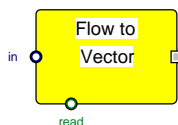


FLOWTOPAK test program

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:
name_in
name_read

Data Type:
FRACT
BOOL

Data Struct:
WORD
BIT

Connection:
mandatory
mandatory

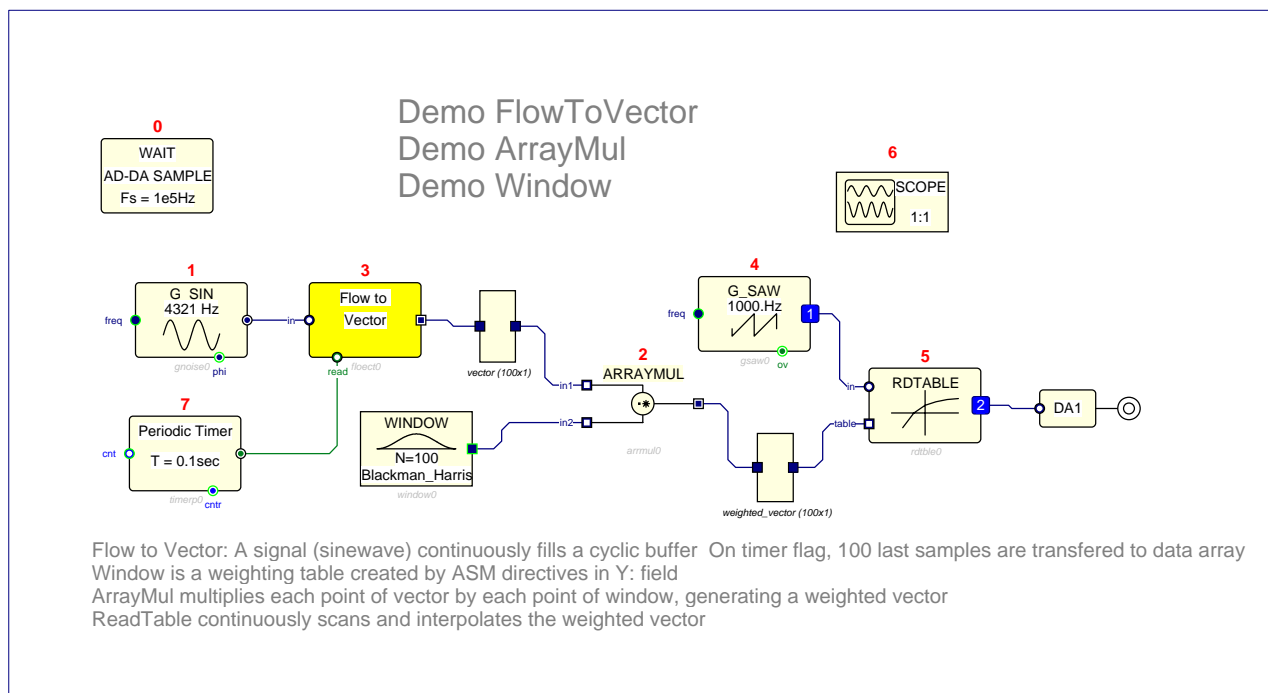
OUTPUTS

Name:
name

Data Type:
defined by cn

Data Struct:
Matrix of

Connection:
normal

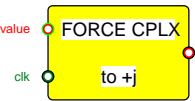


FLOWTOVECT test program

FORCEC

Force complex

FORCEC



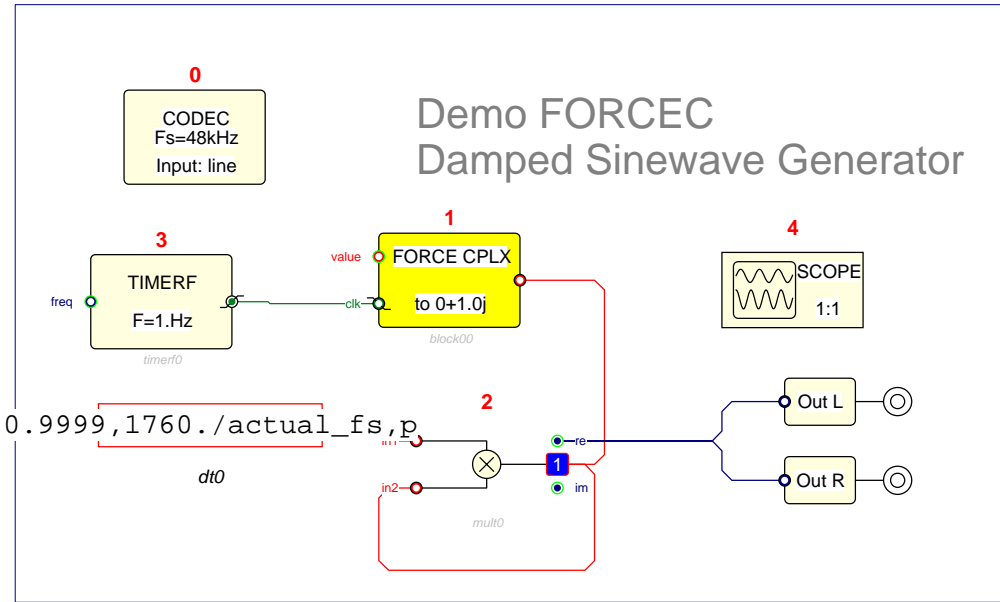
CATEGORY: CONTROL

DESCRIPTION:
Force complex
On clk TRUE, set connected complex output to VALUE
then reset clk to FALSE.

PARAMETERS:

Parameter:	Default values:
Value_Real	0
Value_Imaginary	1.0

INPUTS	Data Type:	Data Struct:	Connection:
Name:	COMPLEX	WORD	optional
name_value	BOOL	BIT	mandatory
name_clk	COMPLEX	WORD	mandatory

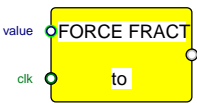


FORCEC test program

FORCEF

Force to value

FORCEF

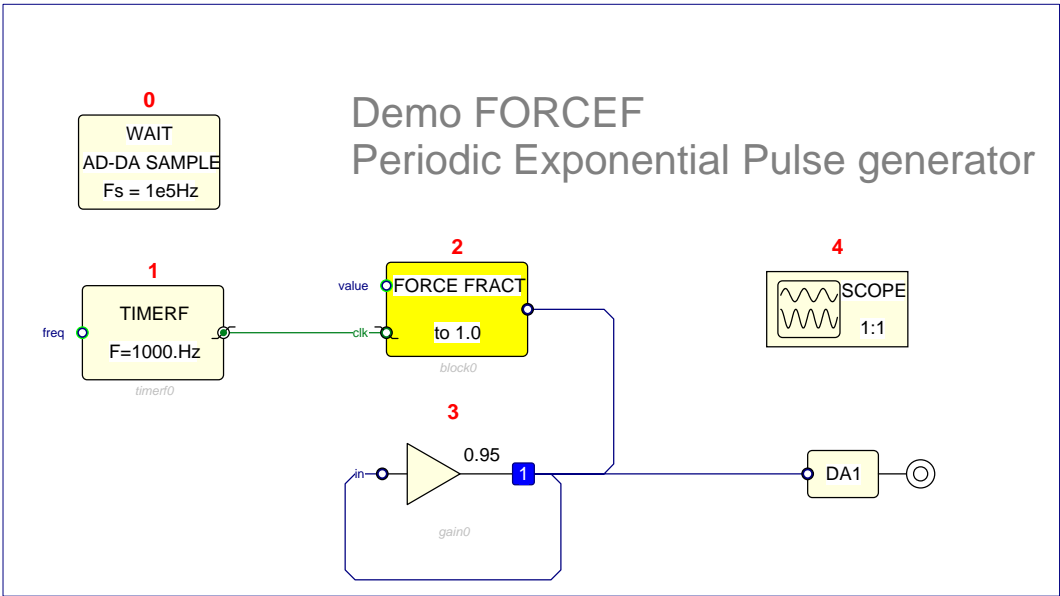


CATEGORY: CONTROL

DESCRIPTION:
Force to value
On clk TRUE, set connected fractional output to VALUE
then reset clk to FALSE

PARAMETERS:
Parameter: Value
Default values: 1.0

INPUTS	Data Type:	Data Struct:	Connection:
Name:	BOOL	BIT	mandatory
name_clk	FRACT	WORD	optional
name_value			mandatory
name	defined by cn		

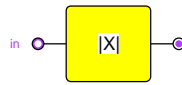


FORCEF test program

FP_ABS

Floating Point absolute value

FP_ABS



DESCRIPTION:
Floating Point absolute value

INPUTS
Name:
name_in

Data Type:
FLOAT

Data Struct:
DWORD

Connection:
mandatory

OUTPUTS
Name:
name

Data Type:
FLOAT

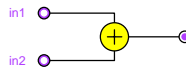
Data Struct:
DWORD

Connection:
normal

FP_ADD

Floating Point Addition

FP_ADD



DESCRIPTION:
Floating Point Addition

INPUTS

Name:
name_in1
name_in2

Data Type:
FLOAT
FLOAT

Data Struct:
DWORD
DWORD

Connection:
mandatory
mandatory

OUTPUTS

Name:
name

Data Type:
FLOAT

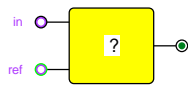
Data Struct:
DWORD

Connection:
normal

FP_CMP

Floating Point Compare

FP_CMP



DESCRIPTION:
Floating Point Compare
Result is True if condition met
Reference level is ref input if connected, parameter otherwise

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

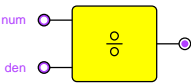
INPUTS			
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name_in	FLOAT	DWORD	mandatory
name_ref	FLOAT	DWORD	optional

OUTPUTS			
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name	BOOL	BIT	normal

FP_DIV

Floating Point division num/den

FP_DIV



DESCRIPTION:
Floating Point division num/den

INPUTS

Name:
name_num
name_den

Data Type:
FLOAT
FLOAT

Data Struct:
DWORD
DWORD

Connection:
mandatory
mandatory

OUTPUTS

Name:
name

Data Type:
FLOAT

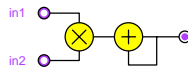
Data Struct:
DWORD

Connection:
normal

FP_MAC

Floating Point MAC

FP_MAC



DESCRIPTION:
Floating Point MAC
Multiply - Accumulate
 $y(k)=y(k-1) +/- x1(k)*x2(k)$

PARAMETERS:
Parameter:
Sign of accumulation

Default values:
pos,neg

INPUTS

Name:
name_in1
name_in2

Data Type:
FLOAT
FLOAT

Data Struct:
DWORD
DWORD

Connection:
mandatory
mandatory

OUTPUTS

Name:
name

Data Type:
FLOAT

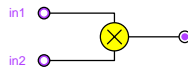
Data Struct:
DWORD

Connection:
normal

FP_MPY

Floating Point multiply

FP_MPY



DESCRIPTION:
Floating Point multiply

INPUTS

Name:
name_in1
name_in2

Data Type:
FLOAT
FLOAT

Data Struct:
DWORD
DWORD

Connection:
mandatory
mandatory

OUTPUTS

Name:
name

Data Type:
FLOAT

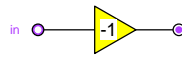
Data Struct:
DWORD

Connection:
normal

FP_NEG

Floating Point Negate

FP_NEG



DESCRIPTION:
Floating Point Negate
Sign inversion $y = -x$

INPUTS
Name:
name_in

Data Type:
FLOAT

Data Struct:
DWORD

Connection:
mandatory

OUTPUTS
Name:
name

Data Type:
FLOAT

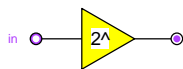
Data Struct:
DWORD

Connection:
normal

FP_SCALE

Floating Point scaling

FP_SCALE



DESCRIPTION:
Floating Point scaling
multiply by 2^N

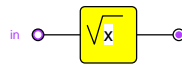
PARAMETERS:
Parameter:
N
Default values:
1

INPUTS			
<i>Name:</i> name_in	<i>Data Type:</i> FLOAT	<i>Data Struct:</i> DWORD	<i>Connection:</i> mandatory
OUTPUTS			
<i>Name:</i> name	<i>Data Type:</i> FLOAT	<i>Data Struct:</i> DWORD	<i>Connection:</i> normal

FP_SQRT

Square root of input

FP_SQRT



DESCRIPTION:
Square root of input

INPUTS
Name:
name_in

Data Type:
FLOAT

Data Struct:
DWORD

Connection:
mandatory

OUTPUTS
Name:
name

Data Type:
FLOAT

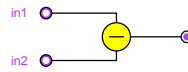
Data Struct:
DWORD

Connection:
normal

FP_SUB

Floating Point subtraction

FP_SUB



DESCRIPTION:
Floating Point subtraction

INPUTS

Name:
name_in1
name_in2

Data Type:
FLOAT
FLOAT

Data Struct:
DWORD
DWORD

Connection:
mandatory
mandatory

OUTPUTS

Name:
name

Data Type:
FLOAT

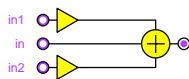
Data Struct:
DWORD

Connection:
normal

FP_WMAC2

Float $y = x_0 + g_1 \cdot x_1 + g_2 \cdot x_2$

FP_WMAC2



DESCRIPTION:

Float $y = x_0 + g_1 \cdot x_1 + g_2 \cdot x_2$

Floating point sum of one input and 2 weighted inputs

$y = in + g_1 \cdot in_1 + g_2 \cdot in_2$

PARAMETERS:

Parameter:

Gain1

Gain2

Default values:

1.0

1.0

INPUTS

Name:

name_in1

name_in2

name_in

Data Type:

FLOAT

FLOAT

FLOAT

Data Struct:

DWORD

DWORD

DWORD

Connection:

mandatory

mandatory

mandatory

OUTPUTS

Name:

name

Data Type:

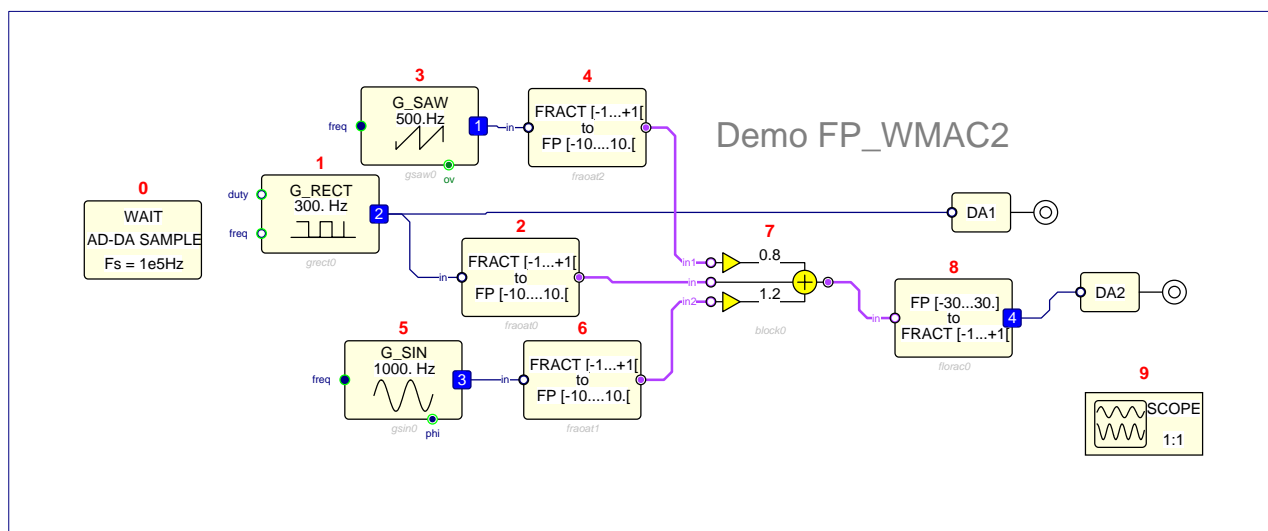
FLOAT

Data Struct:

DWORD

Connection:

normal

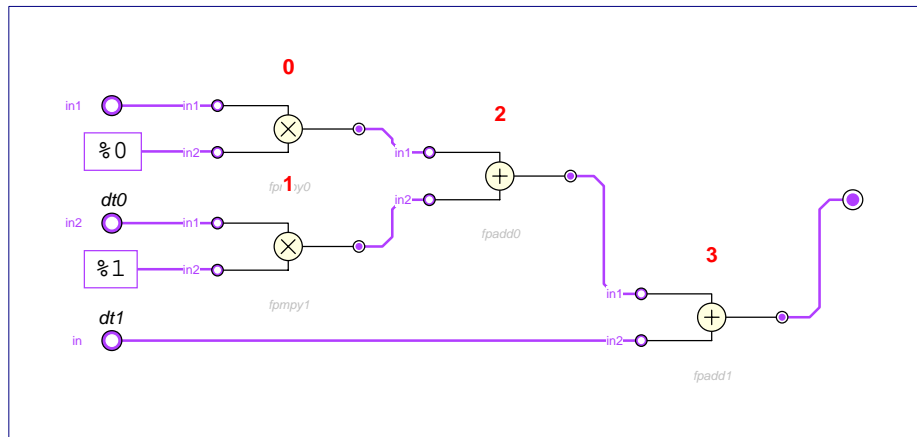
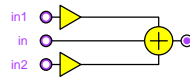


FP_WMAC2 test program

FP_WMAC2

Float $y = x_0 + g_1 \cdot x_1 + g_2 \cdot x_2$

FP_WMAC2

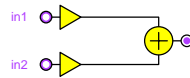


FP_WMAC2 internal schema

FP_WSUM2

Float weighted sum

FP_WSUM2



DESCRIPTION:
Float weighted sum
 $y = g1 \cdot in1 + g2 \cdot in2$

PARAMETERS:

Parameter:

Gain1

Gain2

Default values:

1.0

1.0

INPUTS

Name:

name_in1

name_in2

Data Type:

FLOAT

FLOAT

Data Struct:

DWORD

DWORD

Connection:

mandatory

mandatory

OUTPUTS

Name:

name

Data Type:

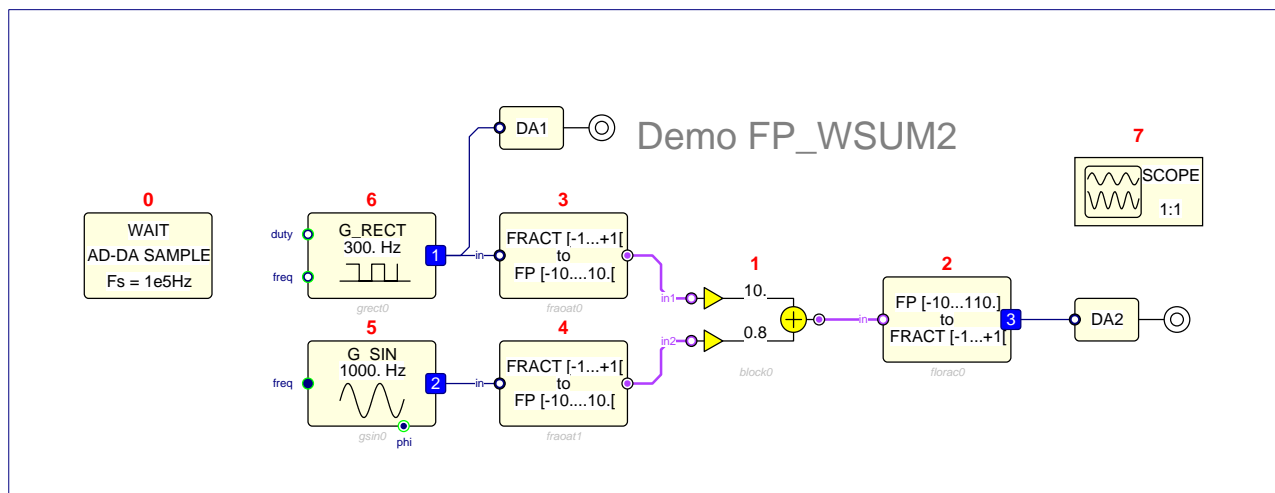
FLOAT

Data Struct:

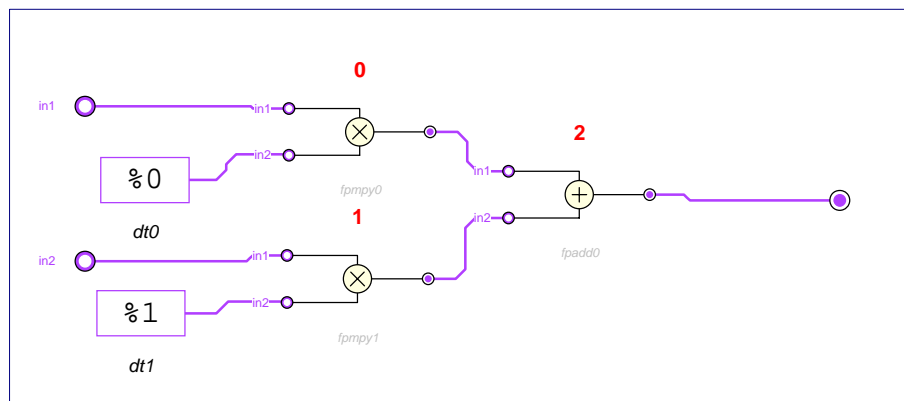
DWORD

Connection:

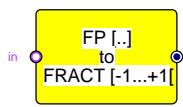
normal



FP_WSUM2 test program



FP_WSUM2 internal schema



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

PARAMETERS:

<i>Parameter:</i>	<i>Default values:</i>
min	-10.
max	10.

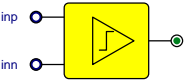
INPUTS			
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name_in	FLOAT	DWORD	mandatory

OUTPUTS			
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name	FRACT	WORD	normal

FRCOMP

Comparator

FRCOMP



CATEGORY: LOGIC

DESCRIPTION:

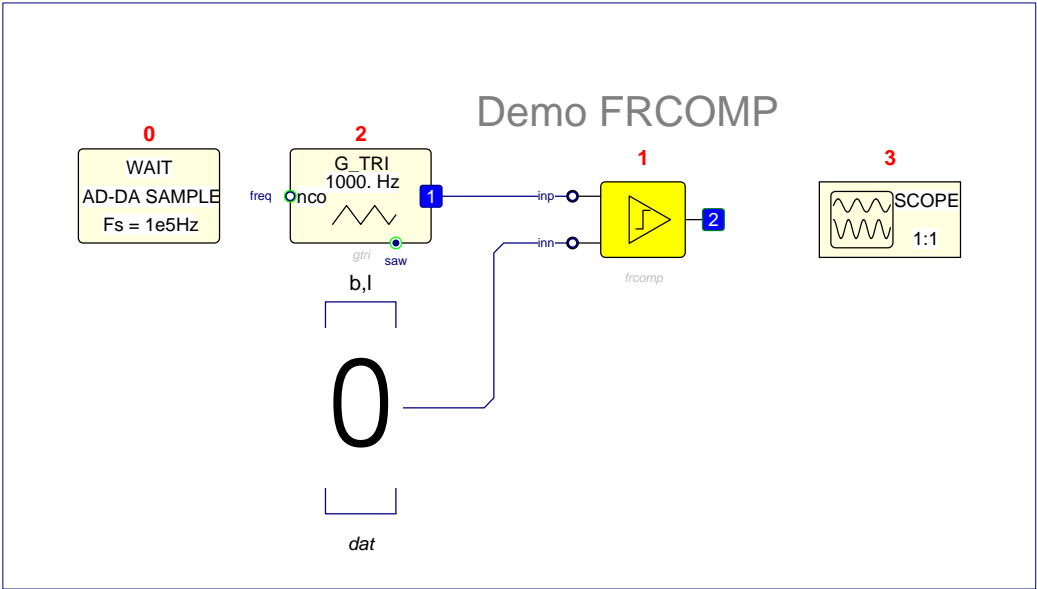
Comparator
Result is TRUE if in is > ref
ref is either defined by connection or by parameter

INPUTS

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

OUTPUTS

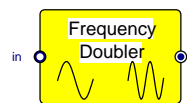
Name:	Data Type:	Data Struct:	Connection:
name	BOOL	BIT	normal



FRCOMP test program

FRDBL

FRDBL



CATEGORY: FUNCTIONS

INPUTS

Name:
name_in

Data Type:
FRACT

Data Struct:
WORD

Connection:
mandatory

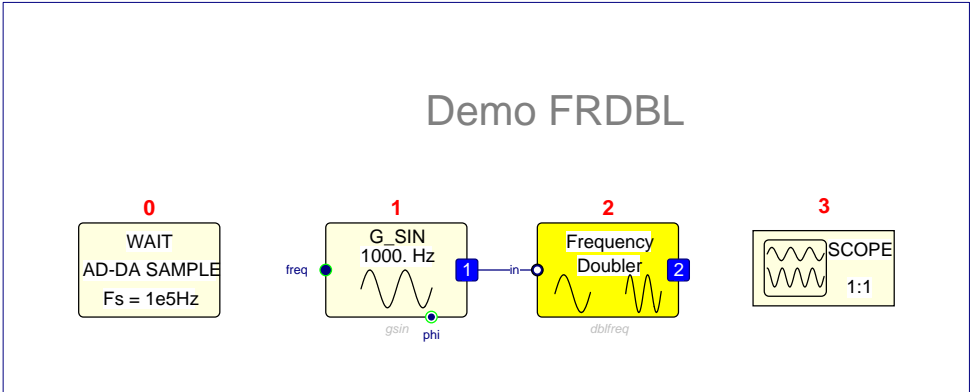
OUTPUTS

Name:
name

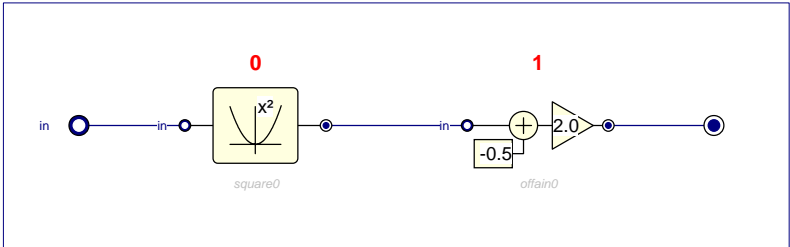
Data Type:
FRACT

Data Struct:
WORD

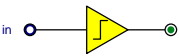
Connection:
normal



FRDBL test program



FRDBL internal schema



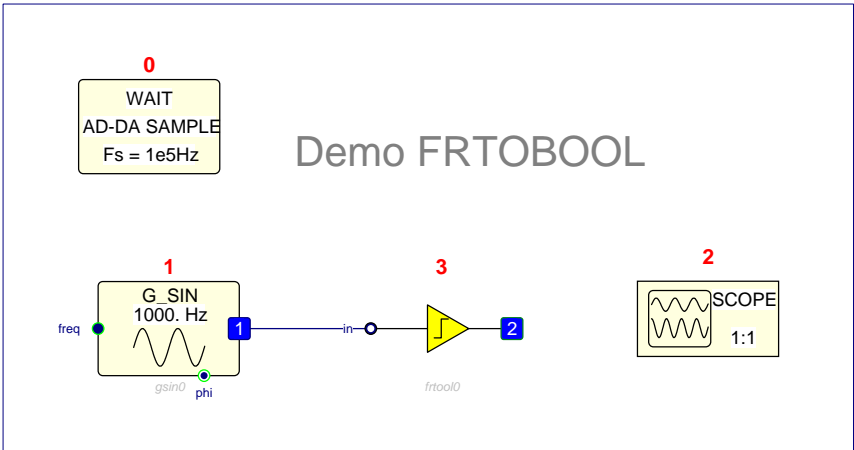
CATEGORY: CONTROL

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

PARAMETERS:
Parameter: *Default values:*
Ref level 0

INPUTS			
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name_in	FRACT	WORD	mandatory

OUTPUTS			
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name	BOOL	BIT	normal

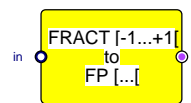


FRTOBOOL test program

FRTOFP

Fract to Float

FRTOFP

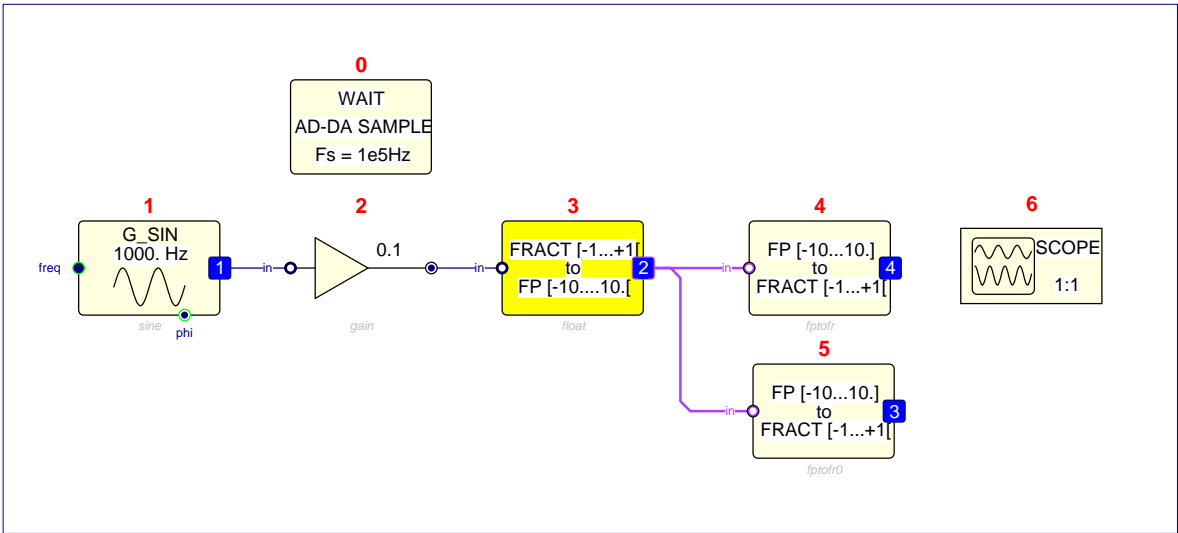


DESCRIPTION:
Fract to Float
Converts Fract[-1.0..+1.0[--> Float [min ...max[

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

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

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

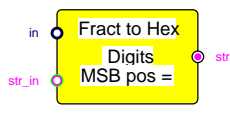


FRTOFP test program

FRTOHEX

Fract to Hex-String

FRTOHEX



CATEGORY: STRING

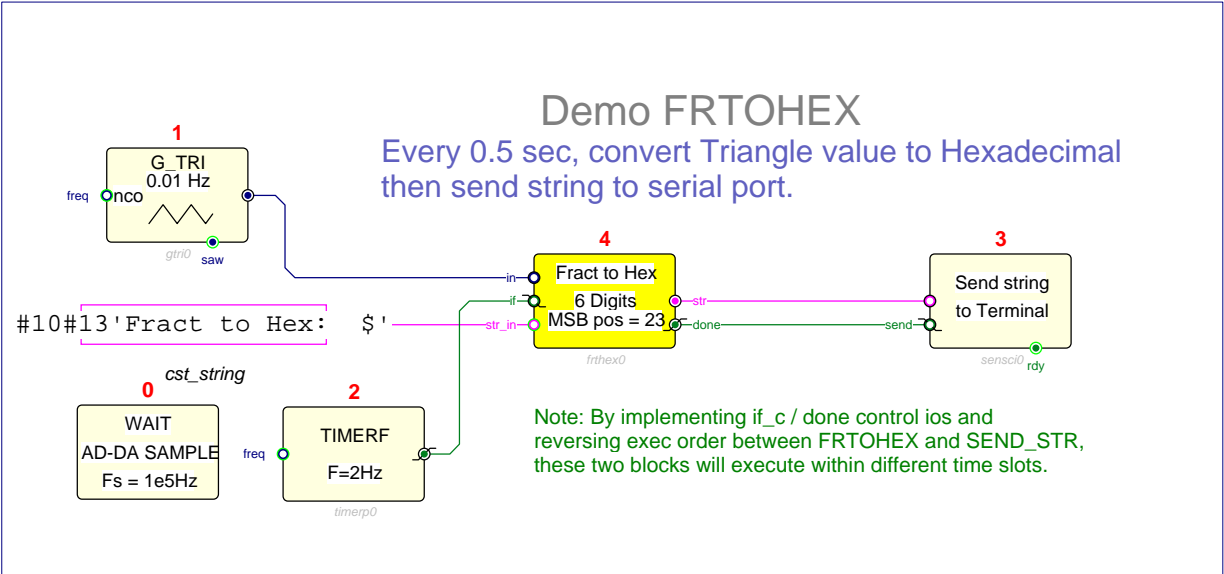
DESCRIPTION:
Fract to Hex-String
Convert Fractional input to Hexadecimal String

PARAMETERS:

<i>Parameter:</i>	<i>Default values:</i>
Nb digits	6
MSB position	23

INPUTS			
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name_in	FRACT	WORD	mandatory
name_str_in	STRING	WORD	optional

OUTPUTS			
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name_str	STRING	WORD	normal

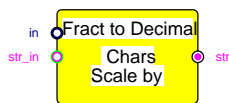


FRTOHEX test program

FRTOSTR

Fract to String

FRTOSTR



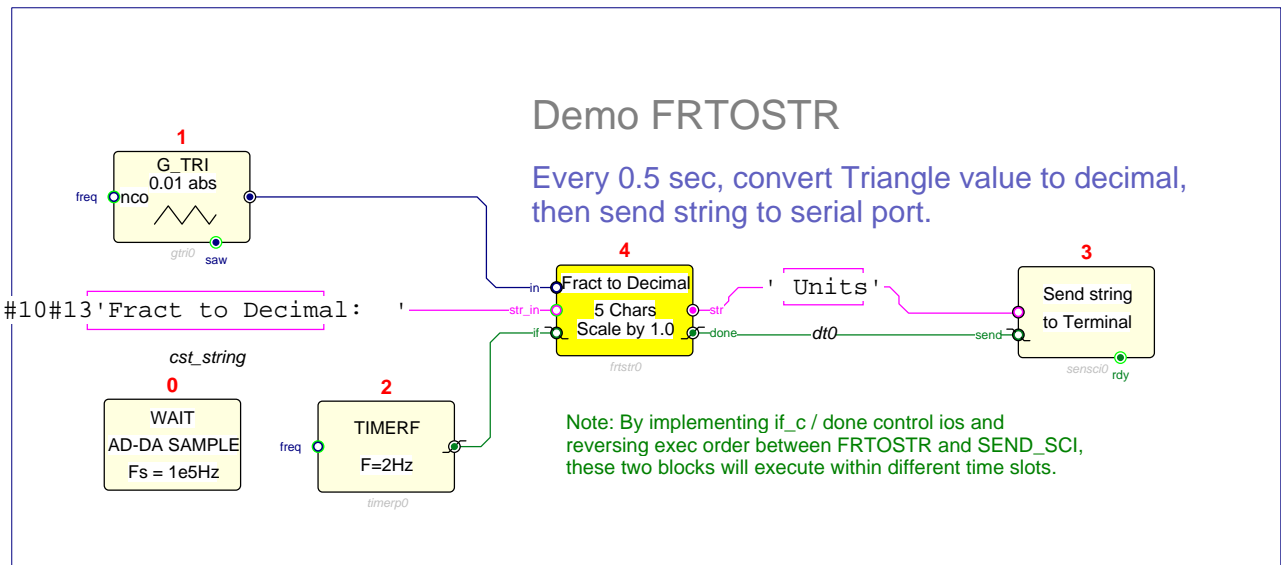
CATEGORY: STRING

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

PARAMETERS:
Parameter: Characters
Default values: 7
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

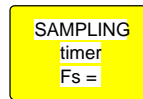


FRTOSTR test program

FS_TIMER

Waits for sample time

FS_TIMER



CATEGORY: TIMING

DESCRIPTION:

Waits for sample time
Defines actual_fs

PARAMETERS:

Parameter:

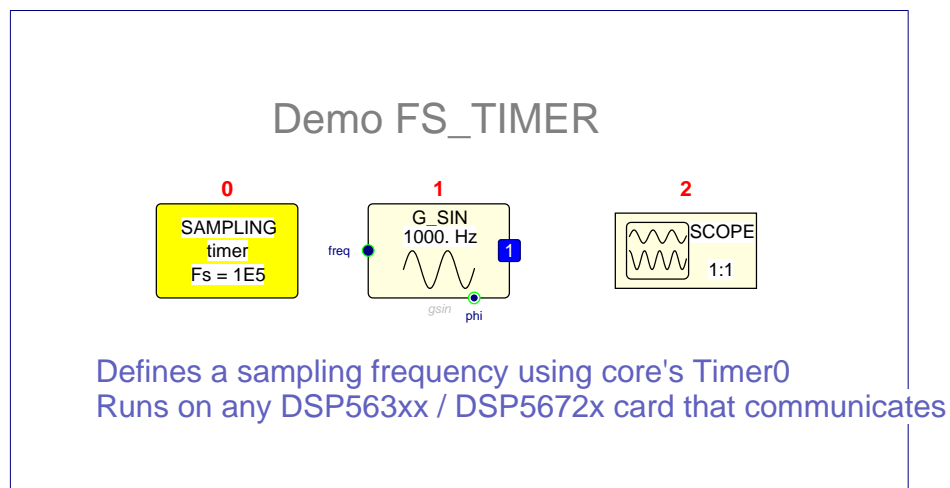
Fs

Default values:

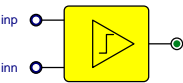
1E5

ATTRIBUTES

Unique, Execute First, Defines: actual_fs

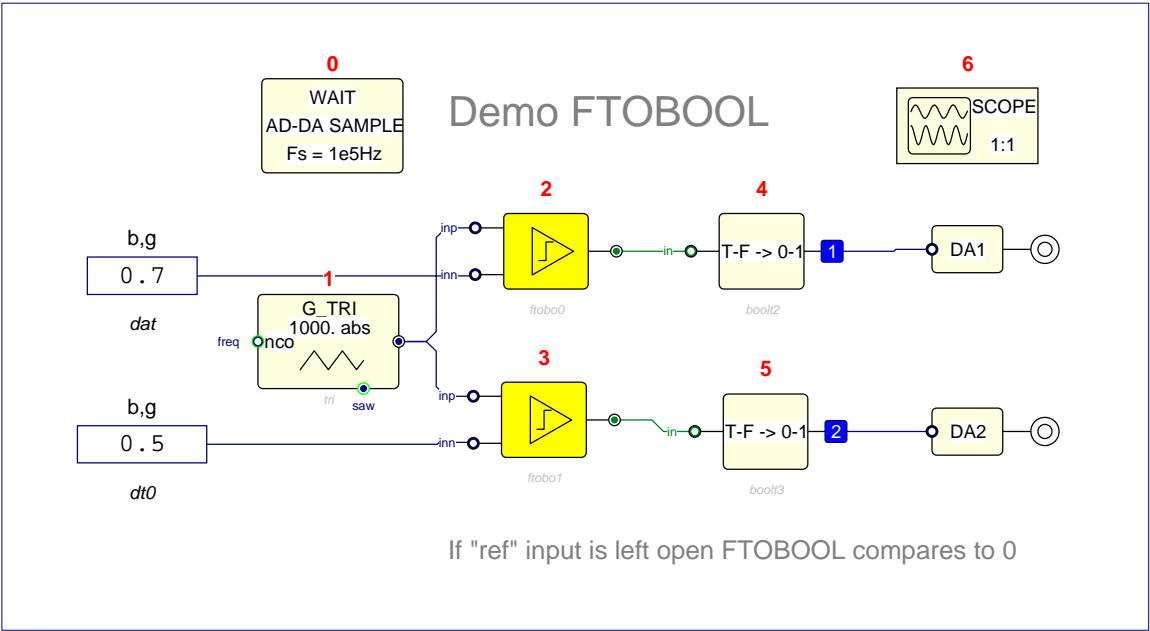


FS_TIMER test program



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

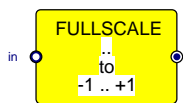


FTOBOOL test program

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:

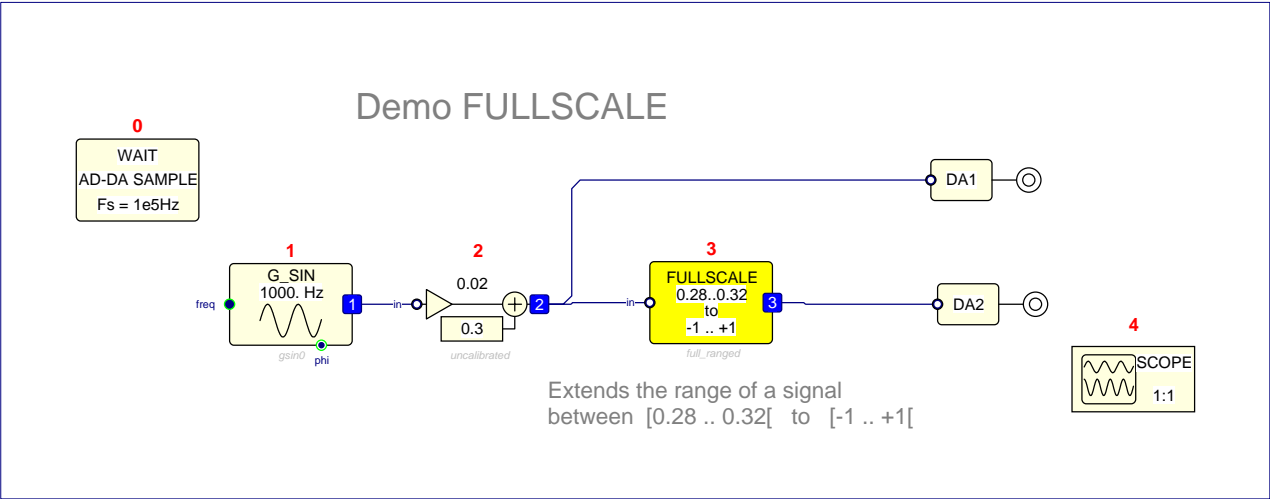
Parameter:	Default values:
Min input	0.1
Max input	0.2

INPUTS

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

OUTPUTS

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

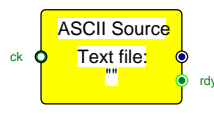


FULLSCALE test program

G_ASCII

Triggered ASCII source

G_ASCII



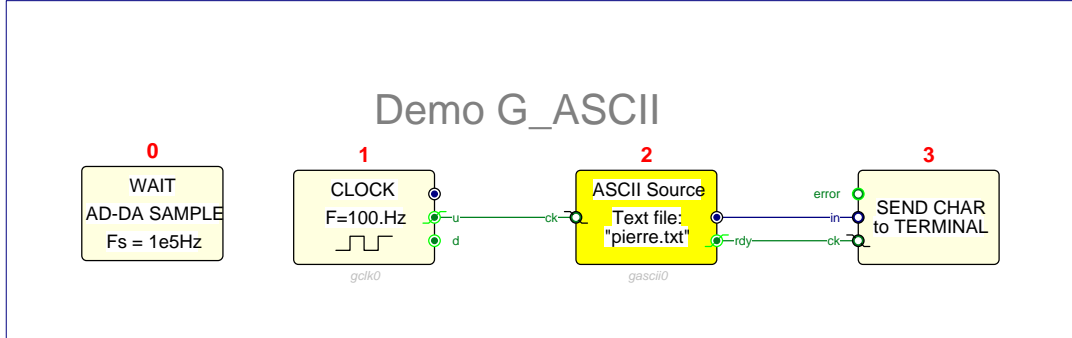
CATEGORY: TELECOM

DESCRIPTION:
Triggered ASCII source

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

INPUTS			
<i>Name:</i> name_ck	<i>Data Type:</i> BOOL	<i>Data Struct:</i> BIT	<i>Connection:</i> mandatory

OUTPUTS			
<i>Name:</i> name	<i>Data Type:</i> FRACT	<i>Data Struct:</i> WORD	<i>Connection:</i> normal
<i>Name:</i> name_rdy	<i>Data Type:</i> BOOL	<i>Data Struct:</i> BIT	<i>Connection:</i> optional

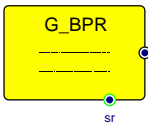


G_ASCII test program

G_BPR

Binary Random Generator

G_BPR

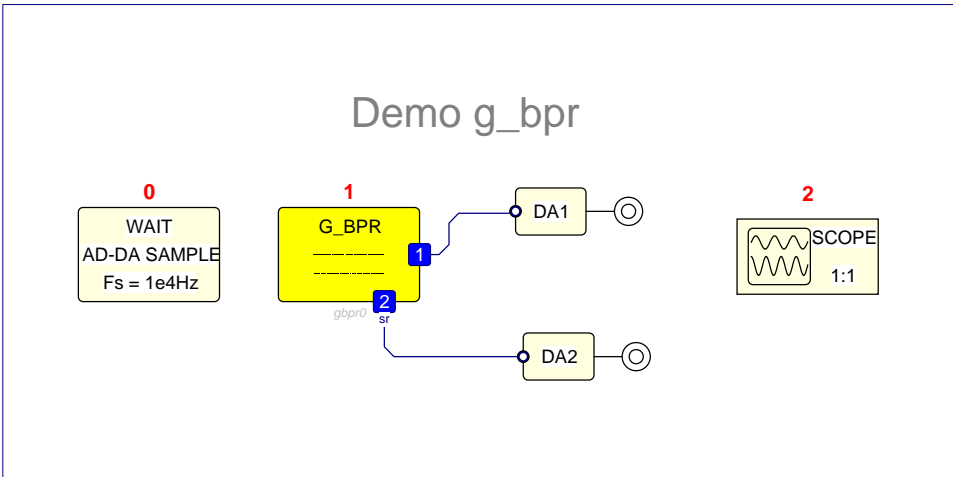


CATEGORY: GENERATORS

DESCRIPTION:
Binary Random Generator
Pseudo Random Sequence length = $2^{\text{bits}} - 1$

PARAMETERS:
Parameter: Bits
Default values: 10

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

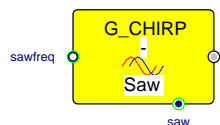


G_BPR test program

G_CHIRP

Chirp Generator

G_CHIRP



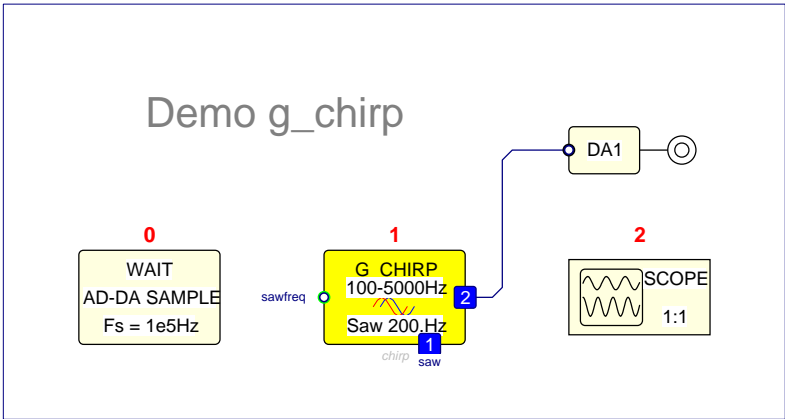
CATEGORY: GENERATORS

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	Data Type:	Data Struct:	Connection:
<i>Name:</i> name_sawfreq	FRACT	WORD	optional

OUTPUTS	Data Type:	Data Struct:	Connection:
<i>Name:</i> name_saw	defined by cn FRACT	WORD	normal optional

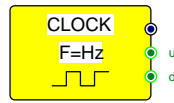


G_CHIRP test program

G_CLK

Clock Generator

G_CLK



CATEGORY: TELECOM

DESCRIPTION:

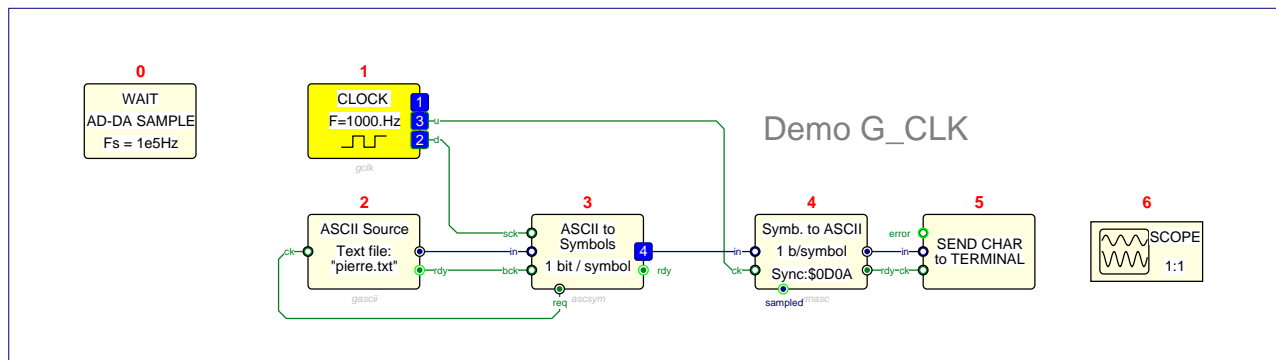
Clock Generator
Boolean Clock generator for transmission Baud Rate
u = rising edge; d = falling edge

PARAMETERS:

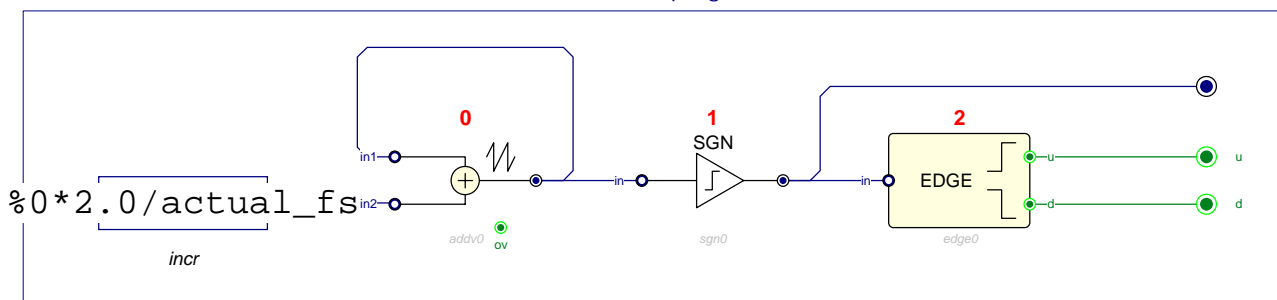
Parameter: Frequency (Hz)
Default values: 1000.

OUTPUTS

<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name_u	BOOL	BIT	optional
name_d	BOOL	BIT	optional
name	FRACT	WORD	normal



G_CLK test program

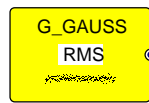


G_CLK internal schema

G_GAUSS

Gaussian Noise

G_GAUSS



CATEGORY: GENERATORS

DESCRIPTION:

Gaussian Noise

Gaussian generator with Standard Deviation Sigma.

Nb acc determines generator precision. Choose 4 for power calculation, >= 25 for error rate calculation

PARAMETERS:

Parameter:

sigma
Nb acc

Default values:

0.2
4

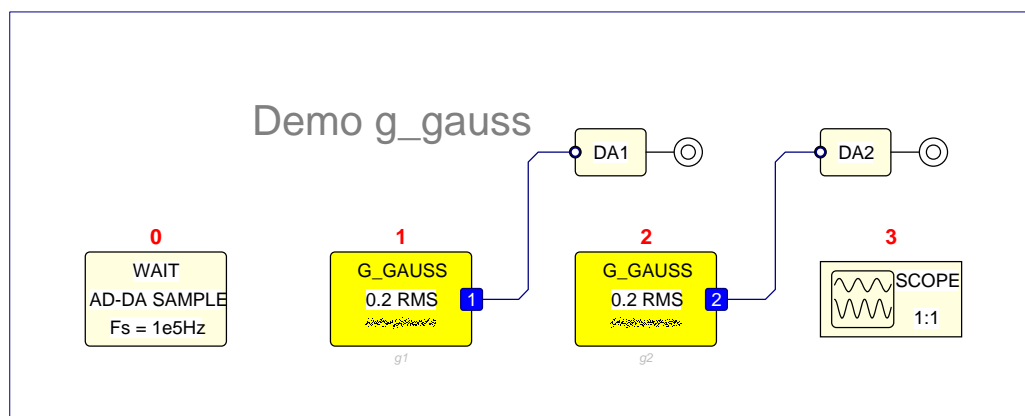
OUTPUTS

Name:
name

Data Type:
FRACT

Data Struct:
WORD

Connection:
normal

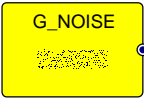


G_GAUSS test program

G_NOISE

Random generator

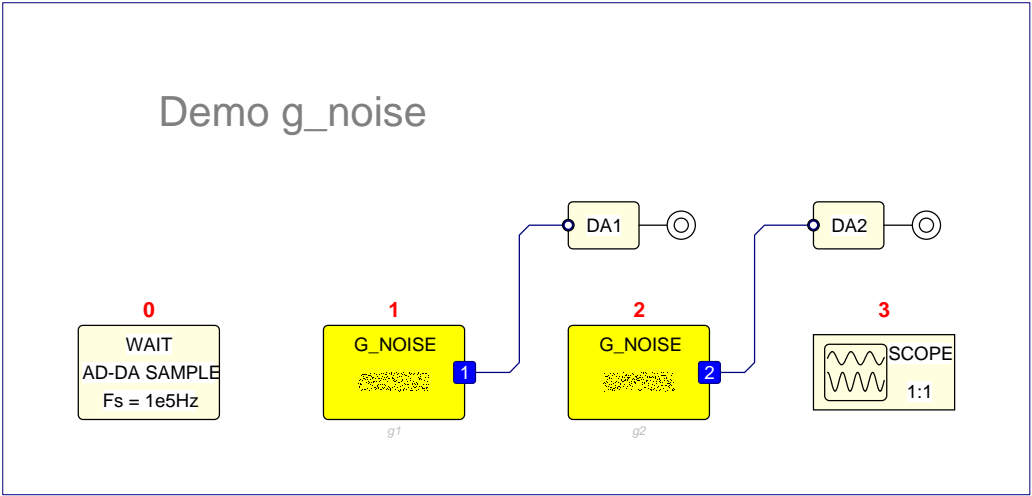
G_NOISE



CATEGORY: GENERATORS

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

OUTPUTS			
Name:	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name	FRACT	WORD	normal

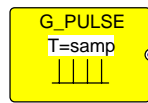


G_NOISE test program

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:

period

Unit

Default values:

0.001

sec,samp

OUTPUTS

Name:

name

Data Type:

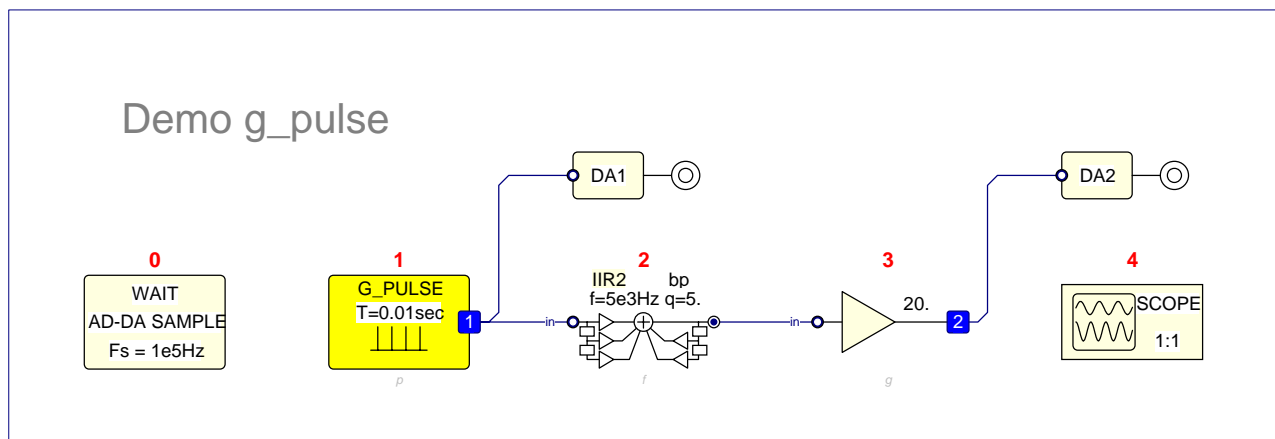
FRACT

Data Struct:

WORD

Connection:

normal

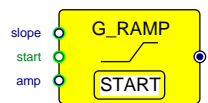


G_PULSE test program

G_RAMP

Slope generator

G_RAMP



CATEGORY: CONTINUOUS

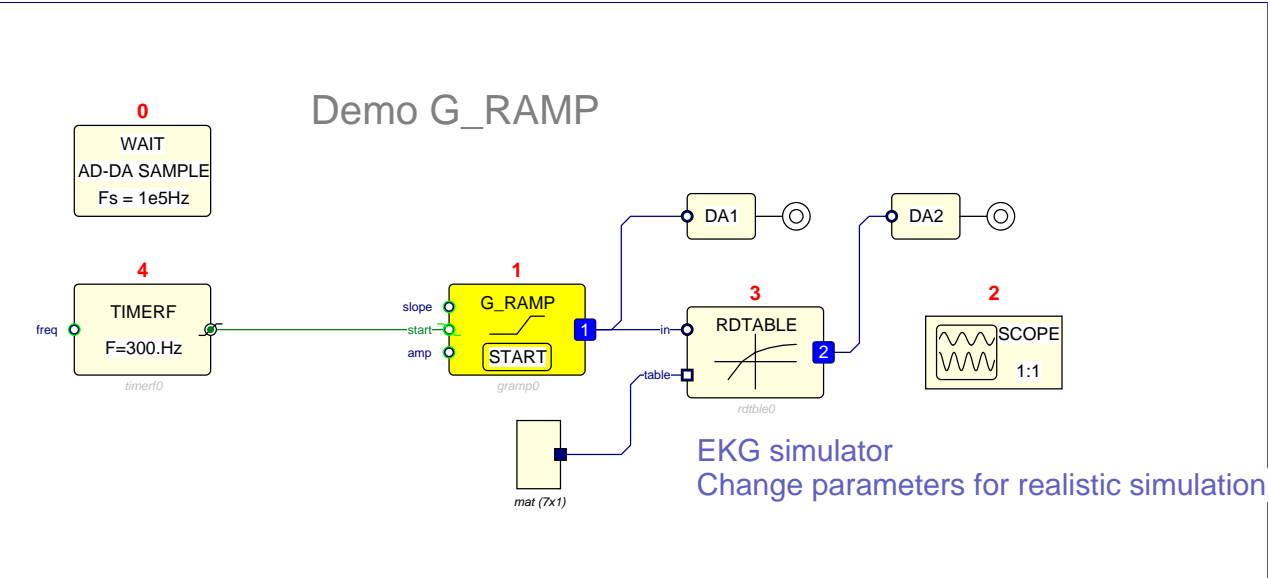
DESCRIPTION:
Slope generator
Starts on SCOPE START command or on boolean VOSC signal
Slope and amplitude controlled by parameter values or by connecting optional inputs.

PARAMETERS:

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

INPUTS	Data Type:	Data Struct:	Connection:
Name:			
name_start	BOOL	BIT	optional
name_slope	FRACT	WORD	optional
name_amp	FRACT	WORD	optional

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

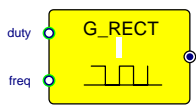


G_RAMP test program

G_RECT

Rectangle generator

G_RECT



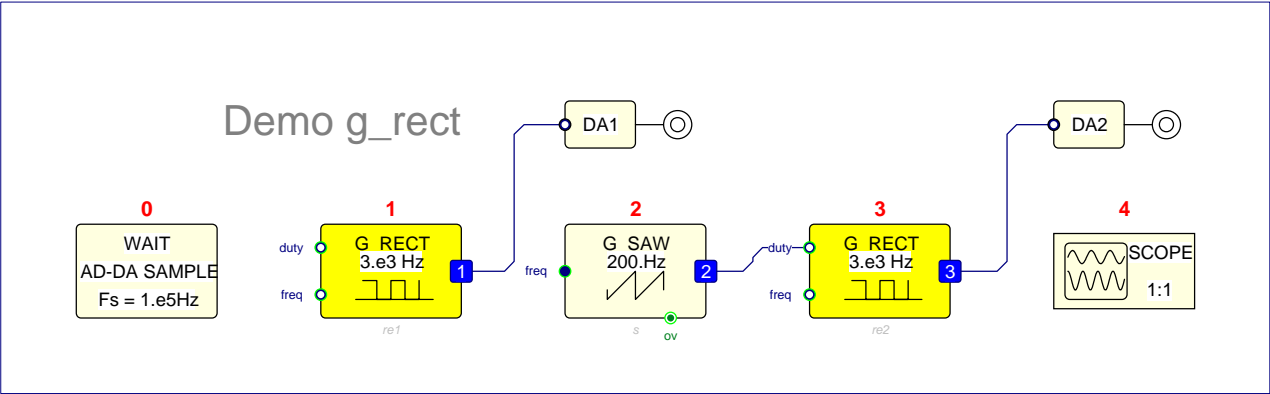
CATEGORY: GENERATORS

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

PARAMETERS:
Parameter:
Freq
Unit
Default values:
1000.
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

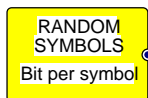


G_RECT test program

G_RNDSYM

Random symbol generator

G_RNDSYM



CATEGORY: TELECOM

DESCRIPTION:
Random symbol generator

PARAMETERS:

Parameter:
Bits per symbol

Default values:
1

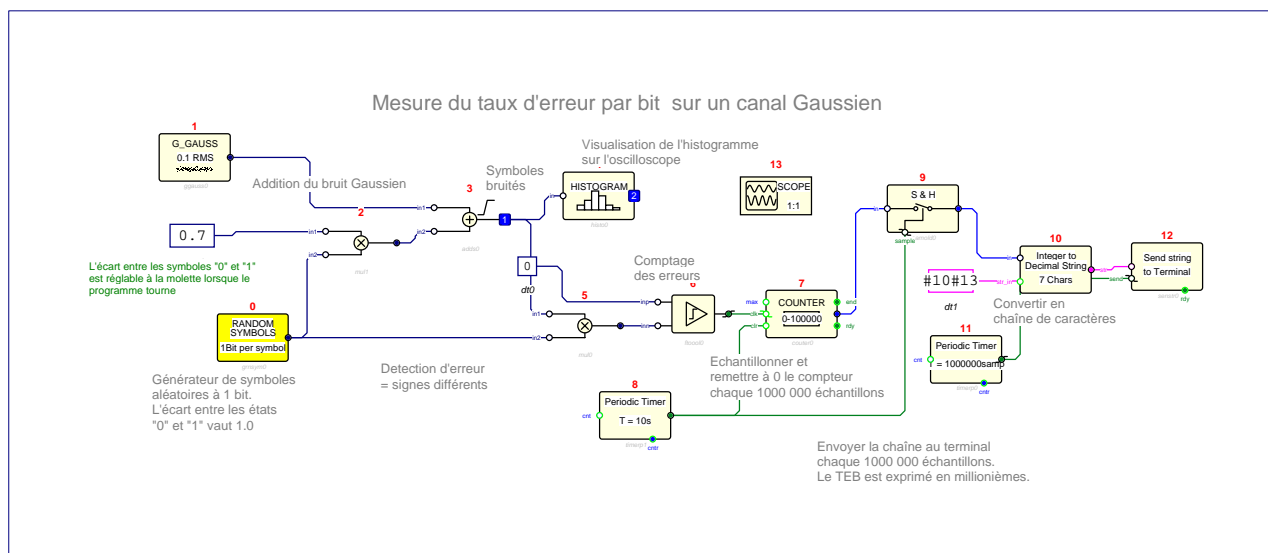
OUTPUTS

Name:
name

Data Type:
FRACT

Data Struct:
WORD

Connection:
normal

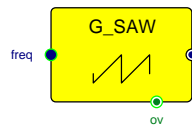


G_RNDSYM test program

G_SAW

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:

frequency
unit

Default values:

1000.
Hz,fs/2

OUTPUTS

Name:

name_freq
name
name_ov

Data Type:

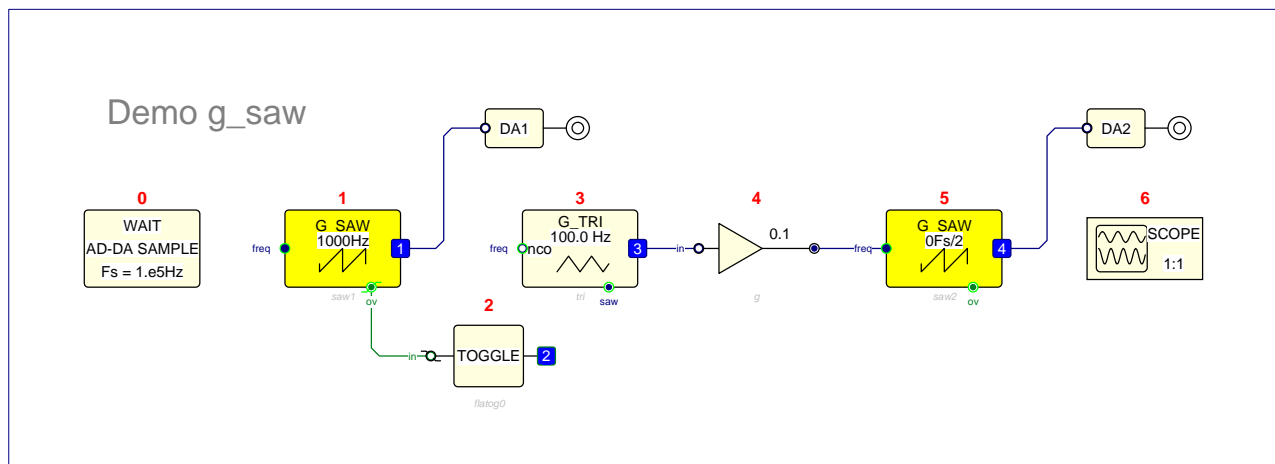
FRACT
FRACT
BOOL

Data Struct:

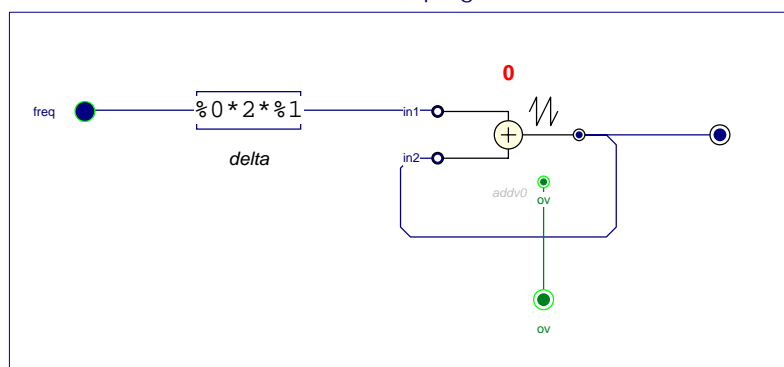
WORD
WORD
BIT

Connection:

optional
normal
optional



G_SAW test program

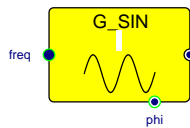


G_SAW internal schema

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:

Frequency

Unit

Default values:

1000.

Hz,kHz,MHz,Fs,Fs/2

OUTPUTS

Name:

name_freq

name_phi

name

Data Type:

FRACT

FRACT

FRACT

Data Struct:

WORD

WORD

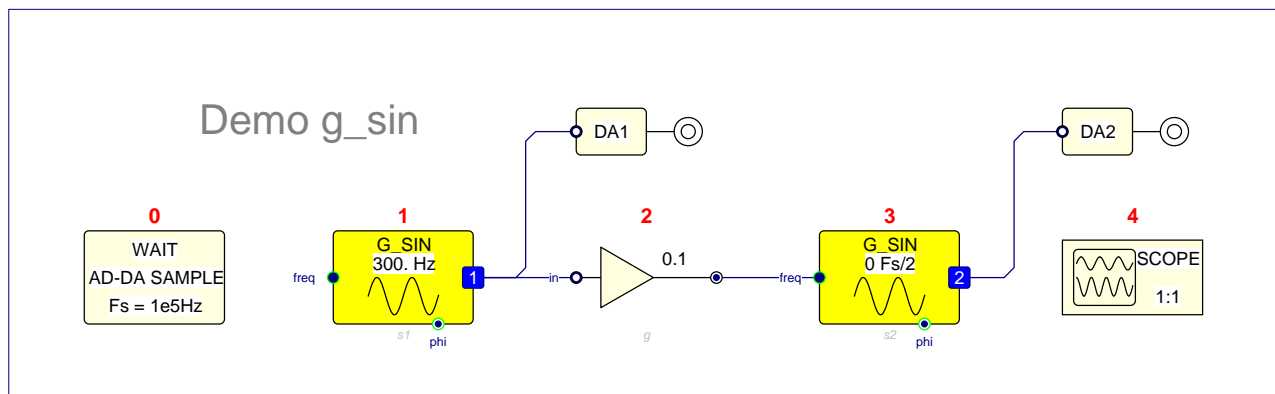
WORD

Connection:

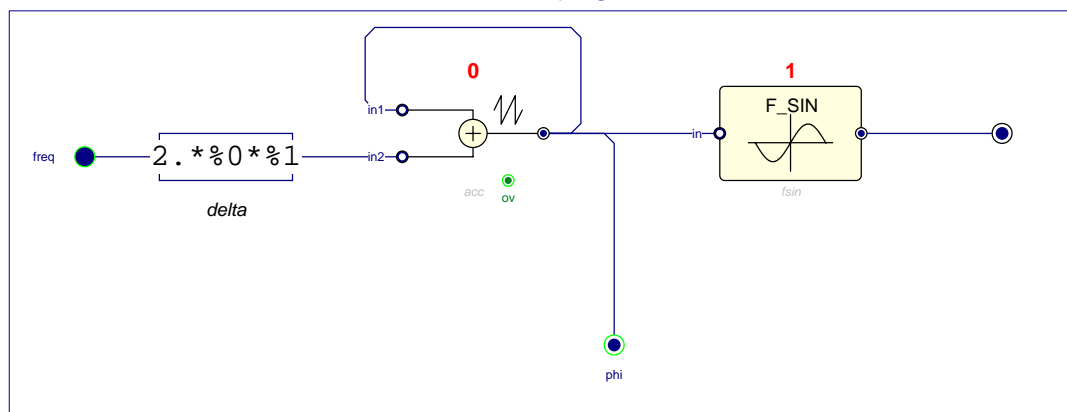
optional

optional

normal



G_SIN test program

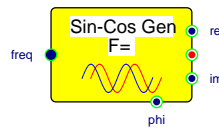


G_SIN internal schema

G_SINCOS

Sine-Cosine Generator

G_SINCOS



CATEGORY: GENERATORS

DESCRIPTION:

Sine-Cosine Generator
Complex or real outputs
 $z(k) = \exp(j2\pi k f / F_s)$

PARAMETERS:

Parameter:

freq
Unit

Default values:

1000.
Hz, Fs/2

OUTPUTS

Name:

name_re
name_freq
name_im
name_phi
name

Data Type:

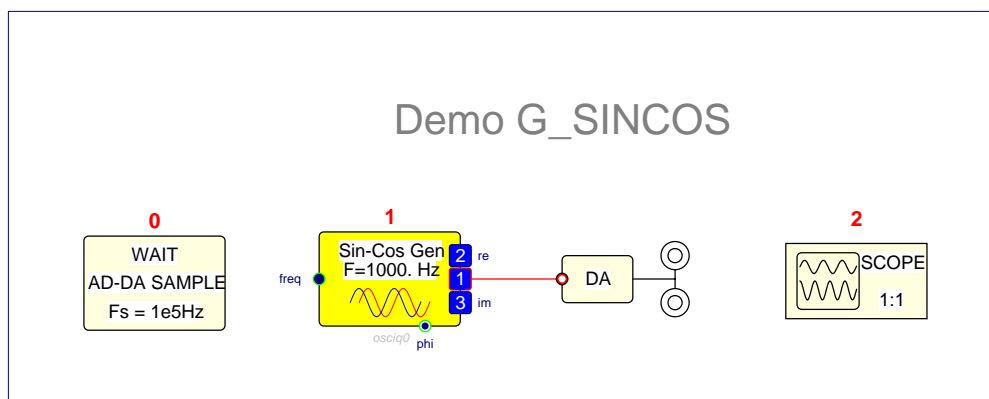
FRACT
FRACT
FRACT
FRACT
COMPLEX

Data Struct:

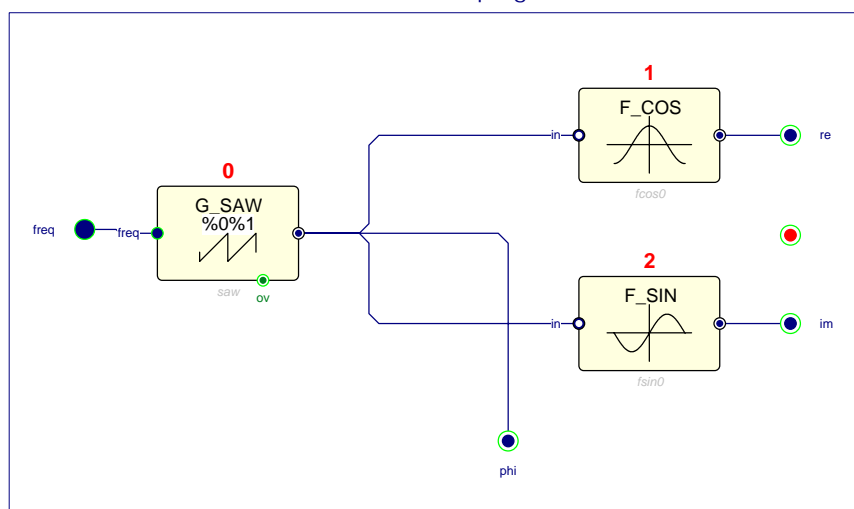
WORD
WORD
WORD
WORD
WORD

Connection:

optional
optional
optional
optional
optional



G_SINCOS test program

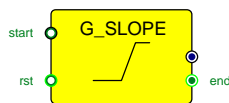


G_SINCOS internal schema

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:

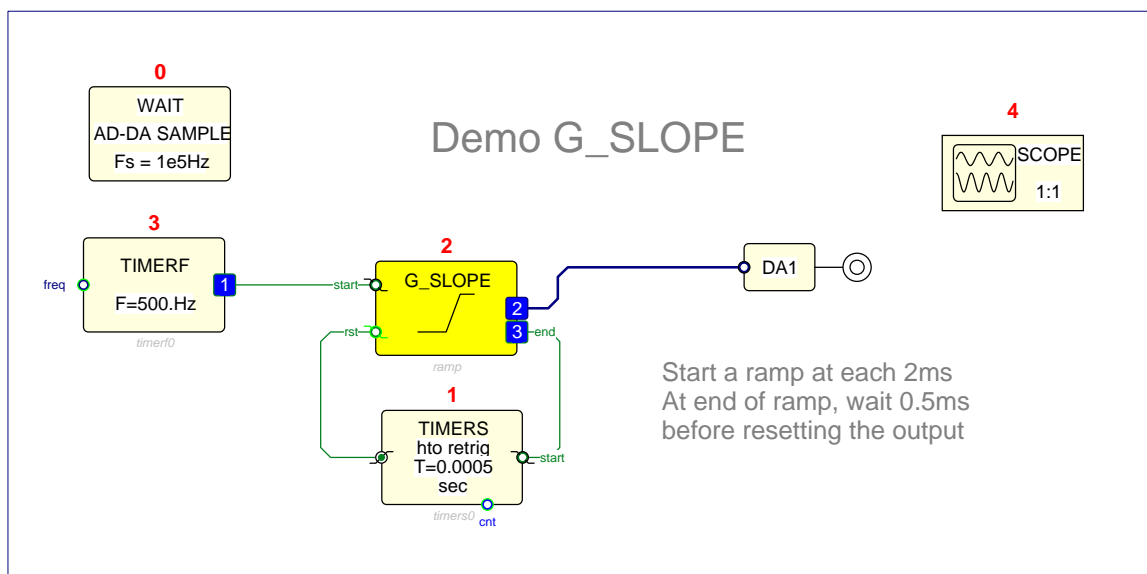
Parameter:	Default values:
Start value	-1.0
End value	1.0
Duration	0.001
Unit	sec,samp

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



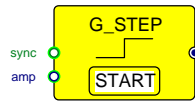
G_SLOPE test program

G_SQUARE

G_STEP

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:

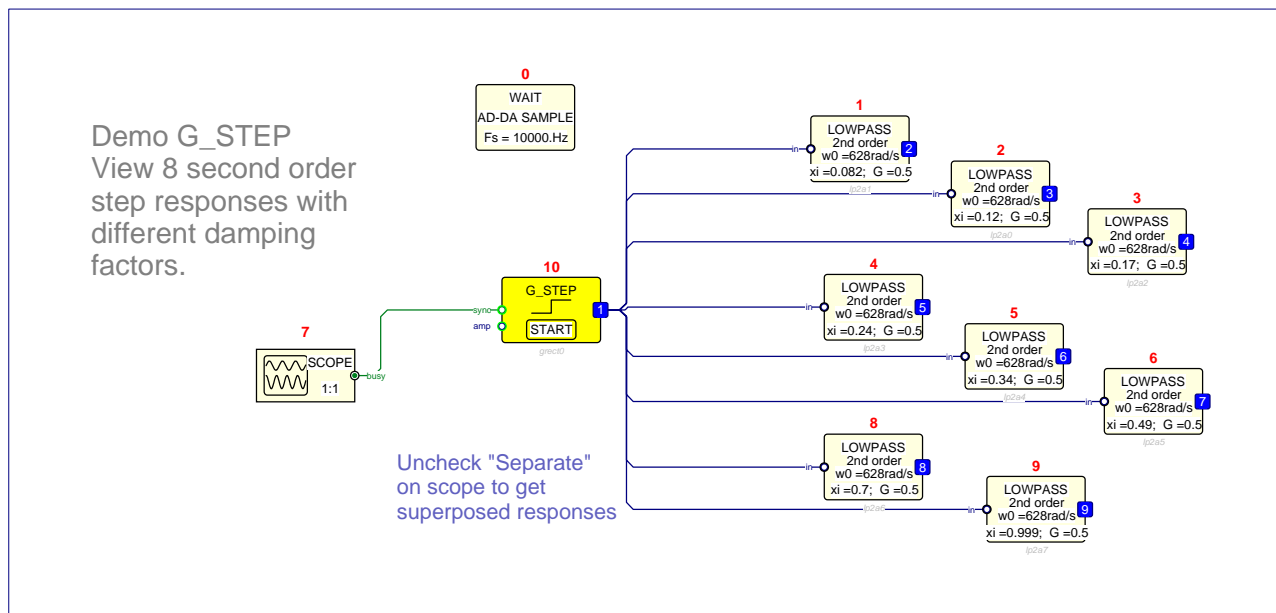
Connection:

name

FRACT

WORD

normal

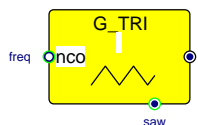


G_STEP test program

G_TRI

Triangle generator

G_TRI



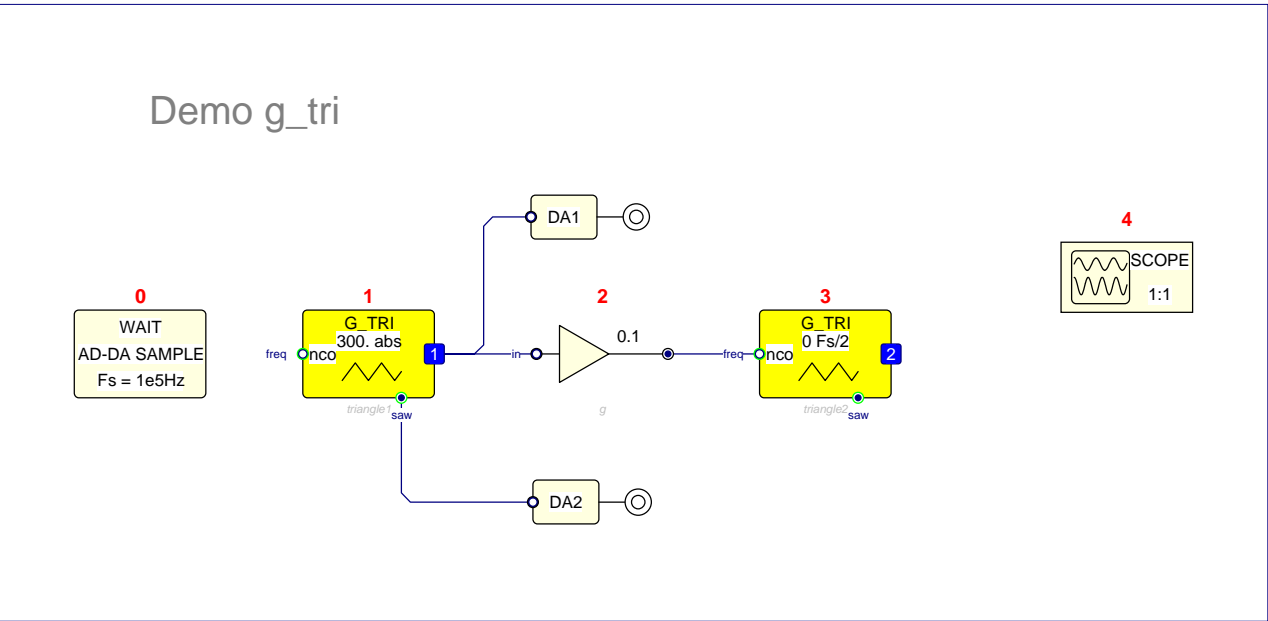
CATEGORY: GENERATORS

DESCRIPTION:
Triangle generator
Frequency modulable by connecting optional input.

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

INPUTS			
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name_freq	FRACT	WORD	optional

OUTPUTS			
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name	FRACT	WORD	normal
name_saw	FRACT	WORD	optional

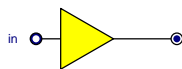


G_TRI test program

GAIN

Fixed real gain

GAIN



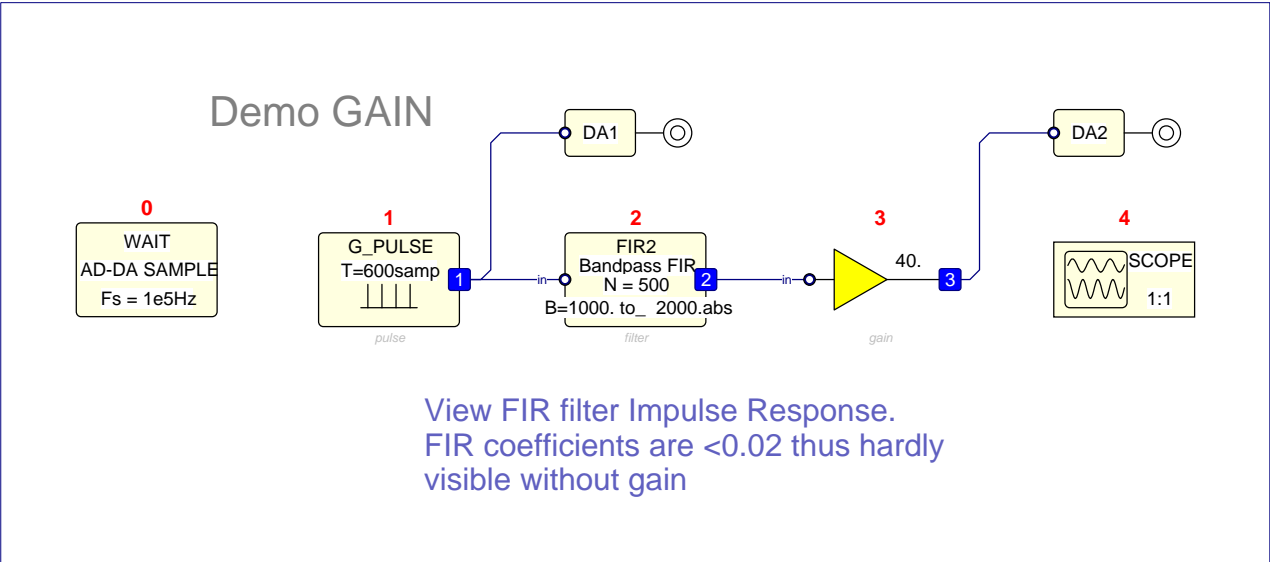
CATEGORY: ARITHMETIC

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

PARAMETERS:
Parameter: Gain
Default values: 10.

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

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



GAIN test program

GENERIC_INST

GENERIC_INST



CATEGORY: MUSIC

INPUTS

Name:
name_start
name_freq
name_vol

Data Type:
BOOL
FRACT
FRACT

Data Struct:
BIT
WORD
WORD

Connection:
mandatory
mandatory
optional

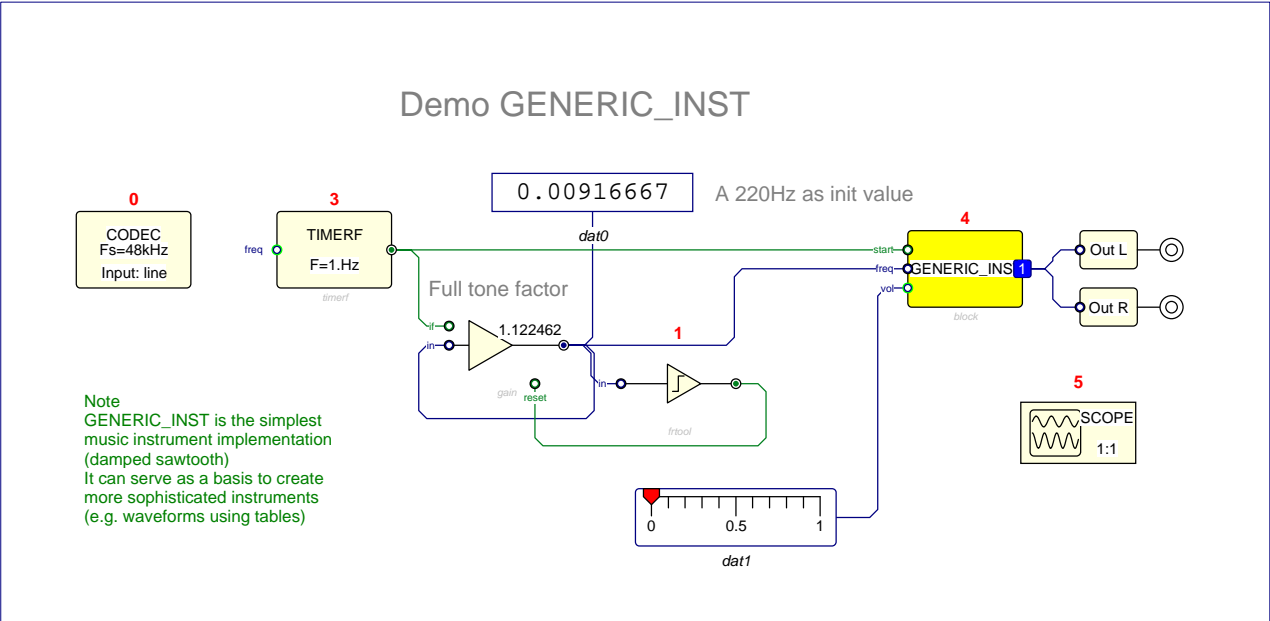
OUTPUTS

Name:
name

Data Type:
FRACT

Data Struct:
WORD

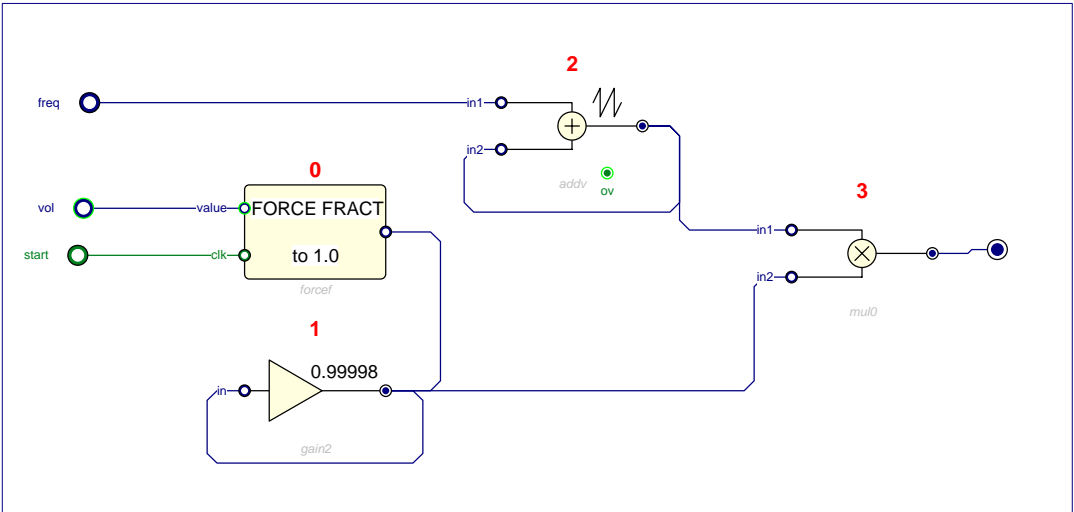
Connection:
normal



GENERIC_INST test program

GENERIC_INST

GENERIC_INST

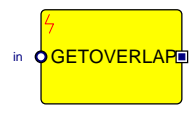


GENERIC_INST internal schema

GETOVERLAP

Get overlap buffer

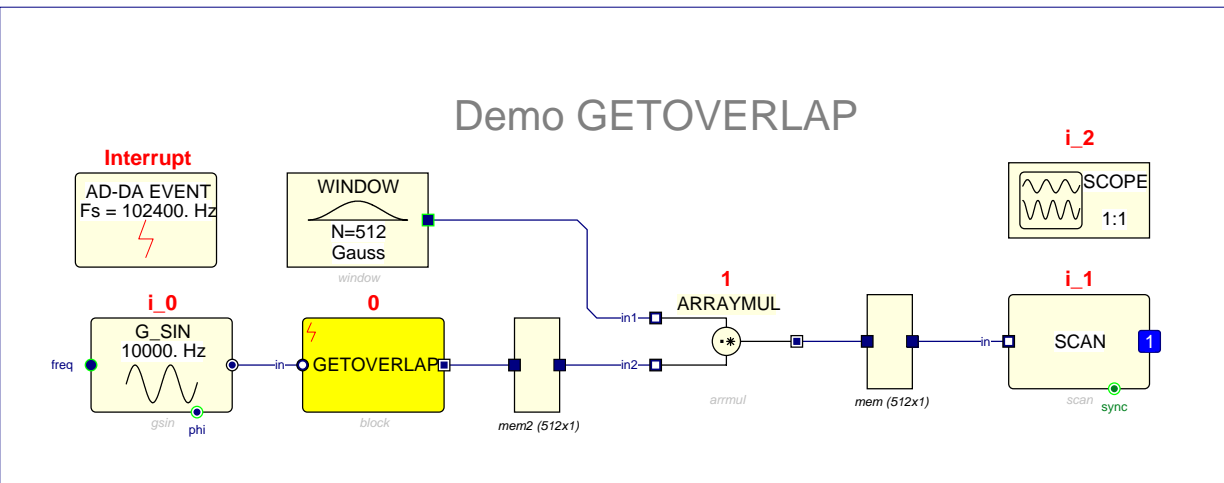
GETOVERLAP



CATEGORY: MATRIX

DESCRIPTION:
Get overlap buffer
Fills output buffer with N last samples

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

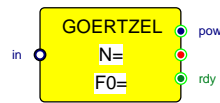


GETOVERLAP test program

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 $4 * \text{Power of complex output}$ (used for tone detection)

$a = 2\pi f_0 / F_s$; $w(0)=0$ $w(-1)=0$

$w(k) = x(k)/N + 2\cos(a)w(k-1) - w(k-2)$ $k=1..N$

Complex out $y = w(N) - \exp(ja)w(N-1)$

$\text{pow out} = 4(w^2(N) + w^2(N-1) - 2\cos(a)w(N)w(N-1))$

PARAMETERS:

Parameter:

Samples
Frequency
Unit

Default values:

1000
1000.
Hz, $F_s/2$

INPUTS

Name:
name_in

Data Type:
FRACT

Data Struct:
WORD

Connection:
mandatory

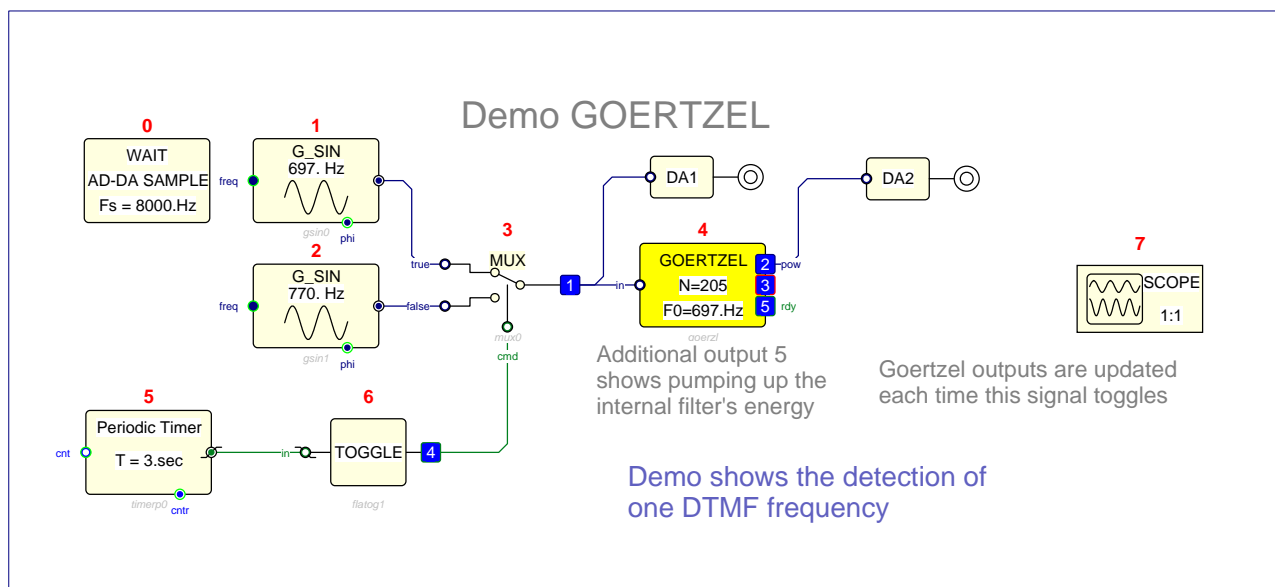
OUTPUTS

Name:
name_rdy
name
name_pow

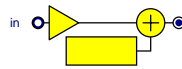
Data Type:
BOOL
COMPLEX
FRACT

Data Struct:
BIT
WORD
WORD

Connection:
optional
optional
optional



GOERTZEL test program



CATEGORY: ARITHMETIC

DESCRIPTION:
Gain followed by offset

PARAMETERS:

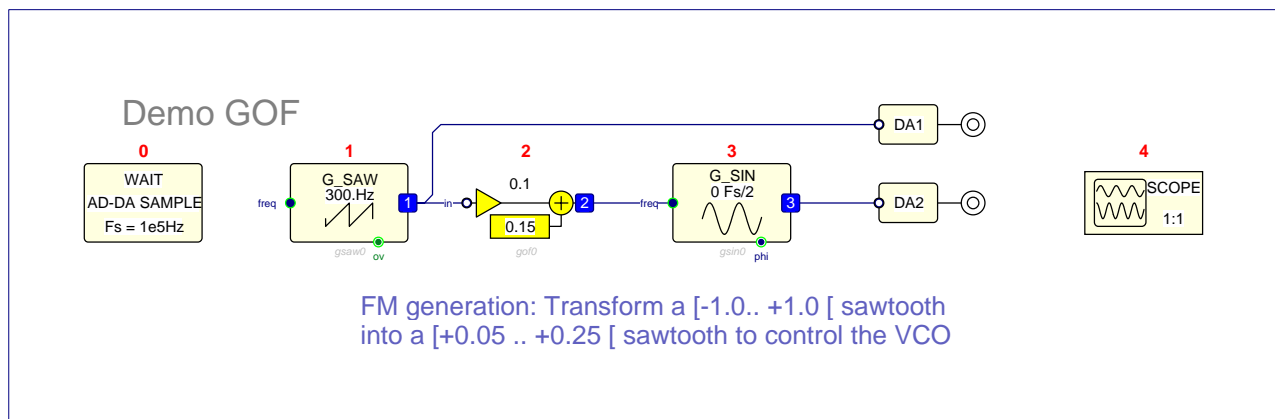
<i>Parameter:</i>	<i>Default values:</i>
gain	0.1
offset	0.1

INPUTS

<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name_in	FRACT	WORD	mandatory

OUTPUTS

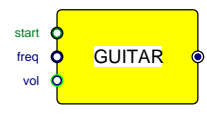
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name	FRACT	WORD	normal



GOF test program

GUITAR

GUITAR



CATEGORY: MUSIC

INPUTS

Name:
name_start
name_freq
name_vol

Data Type:
BOOL
FRACT
FRACT

Data Struct:
BIT
WORD
WORD

Connection:
mandatory
mandatory
optional

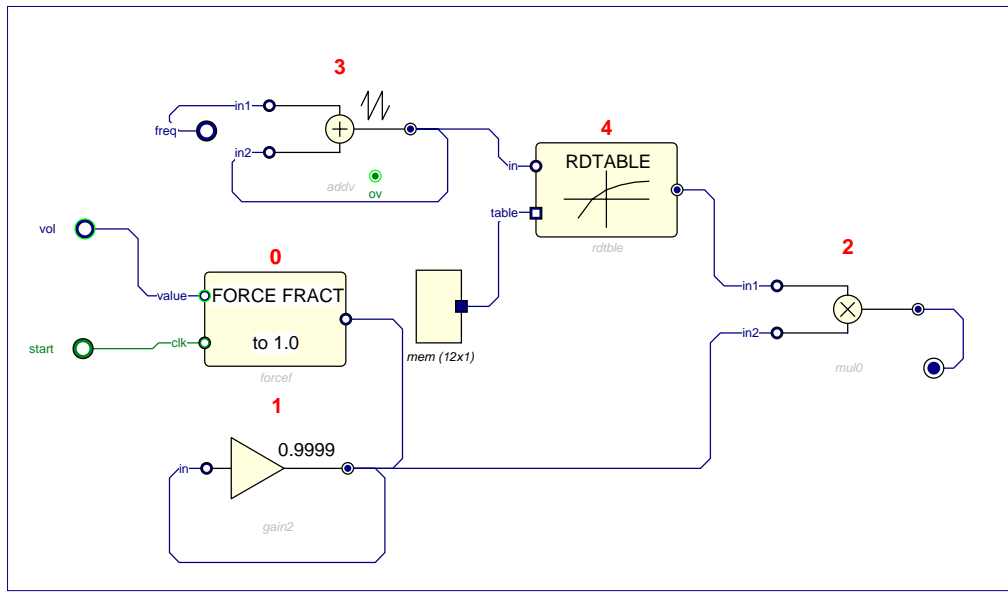
OUTPUTS

Name:
name

Data Type:
FRACT

Data Struct:
WORD

Connection:
normal

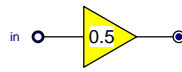


GUITAR internal schema

HALF

Gain by 0.5

HALF



CATEGORY: ARITHMETIC

DESCRIPTION:
Gain by 0.5

INPUTS

Name:
name_in

Data Type:
FRACT

Data Struct:
WORD

Connection:
mandatory

OUTPUTS

Name:
name

Data Type:
FRACT

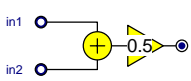
Data Struct:
WORD

Connection:
normal

HALFSUM

Half sum of inputs

HALFSUM



CATEGORY: ARITHMETIC

DESCRIPTION:
Half sum of inputs

INPUTS

Name:
name_in1
name_in2

Data Type:
FRACT
FRACT

Data Struct:
WORD
WORD

Connection:
mandatory
mandatory

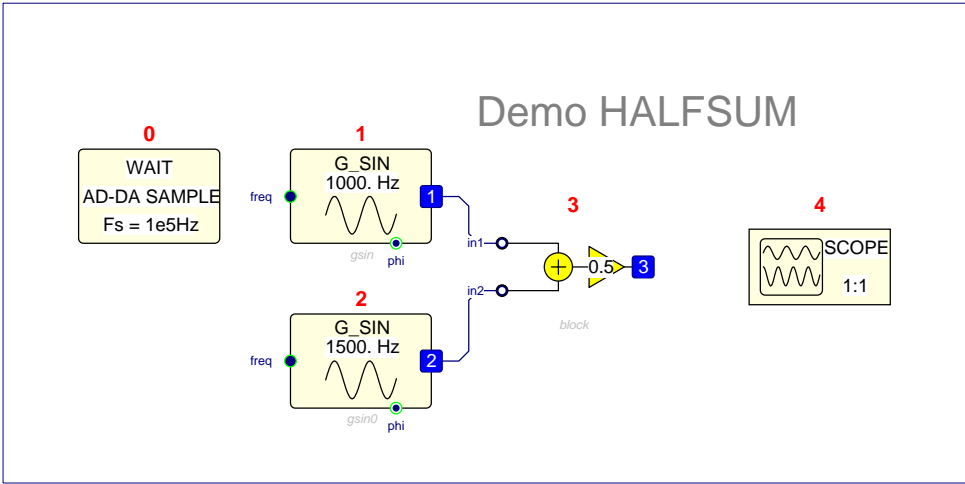
OUTPUTS

Name:
name

Data Type:
FRACT

Data Struct:
WORD

Connection:
normal



HALFSUM test program



CATEGORY: MUSIC

INPUTS

Name:
name_start
name_vol

Data Type:
BOOL
FRACT

Data Struct:
BIT
WORD

Connection:
mandatory
optional

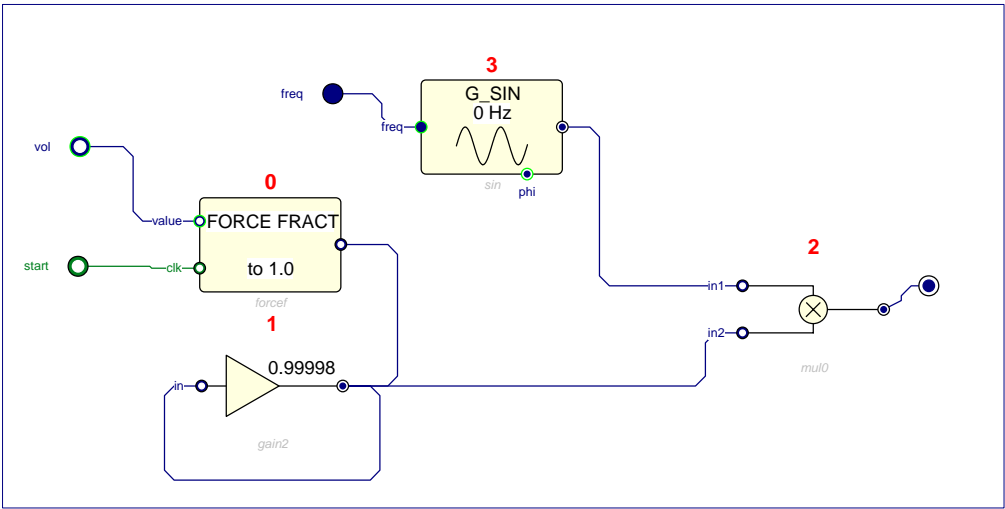
OUTPUTS

Name:
name_freq
name

Data Type:
FRACT
FRACT

Data Struct:
WORD
WORD

Connection:
normal
normal



HARP internal schema

HILBERT

Hilbert transform

HILBERT



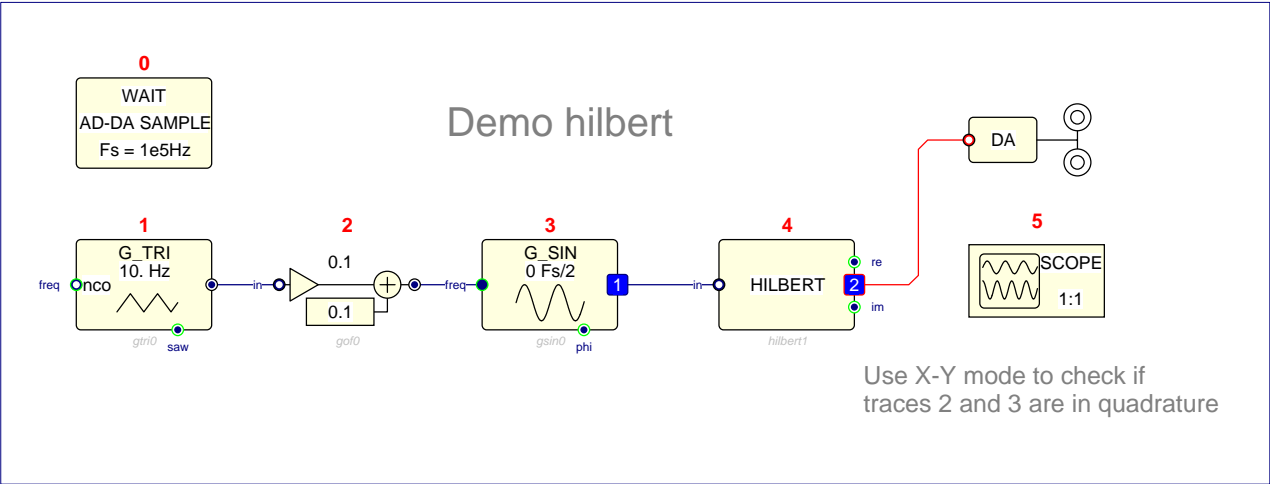
CATEGORY: FILTERS

DESCRIPTION:
Hilbert transform
Transforms real signal into complex signal

PARAMETERS:
Parameter:
number of taps
Default values:
101

INPUTS			
<i>Name:</i> name_in	<i>Data Type:</i> FRACT	<i>Data Struct:</i> WORD	<i>Connection:</i> mandatory

OUTPUTS			
<i>Name:</i> name	<i>Data Type:</i> COMPLEX	<i>Data Struct:</i> WORD	<i>Connection:</i> normal
name_re	FRACT	WORD	optional
name_im	FRACT	WORD	optional



HILBERT test program

HISTO

Buffer switching histogram.

HISTO



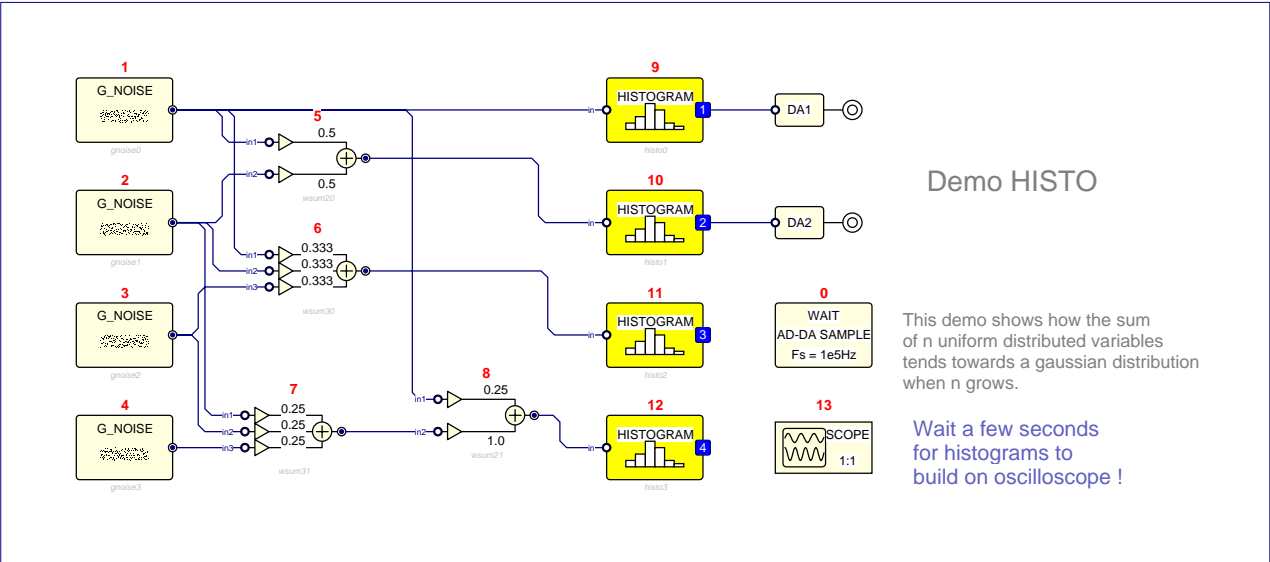
CATEGORY: INSTRUMENTS

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

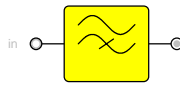
PARAMETERS:
Parameter: *Default values:*
Nb of classes 512
Nb samples total 100000
Gain 1.0

INPUTS
Name: *Data Type:* *Data Struct:* *Connection:*
name_in FRAC T WORD mandatory

OUTPUTS
Name: *Data Type:* *Data Struct:* *Connection:*
name FRAC T WORD normal



HISTO test program



CATEGORY: FILTERS

DESCRIPTION:
First order High-Pass filter

PARAMETERS:
Parameter:
Cutoff frequency
Unit

Default values:
10.
Hz,Fs/2

INPUTS
Name:
name_in

Data Type:
defined by cn

Data Struct:

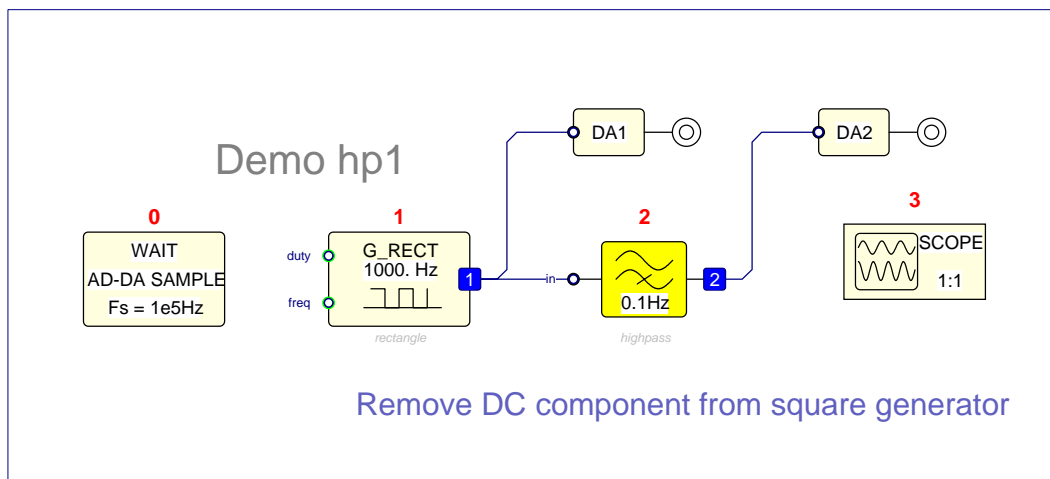
Connection:
mandatory

OUTPUTS
Name:
name

Data Type:
defined by cn

Data Struct:

Connection:
normal

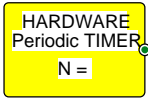


HP1 test program

HWTIMER

HW Periodic Timer

HWTIMER



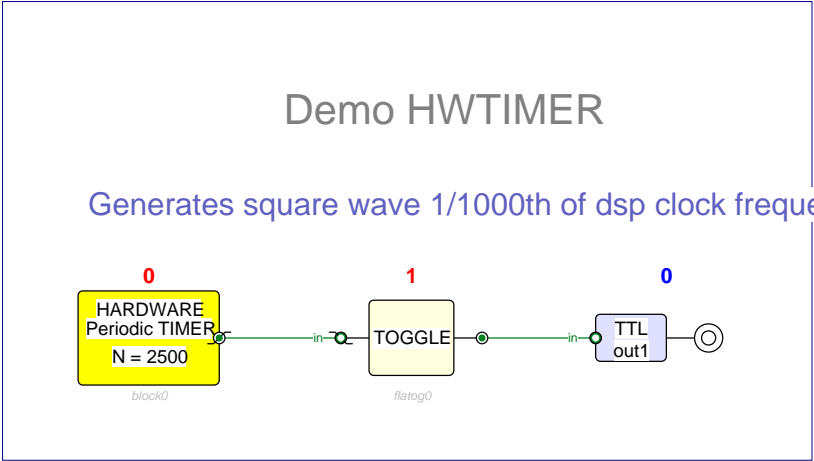
CATEGORY: TIMING

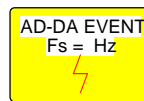
DESCRIPTION:
HW Periodic Timer
Program one of the three 56300 core timers for setting a flag every 2N clock cycles

PARAMETERS:

<i>Parameter:</i>	<i>Default values:</i>
Timer Nr	0,1,2
Count	100

OUTPUTS			
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name	BOOL	BIT	normal





CATEGORY: INTERRUPT

DESCRIPTION:
Install AD-DA Interrupt

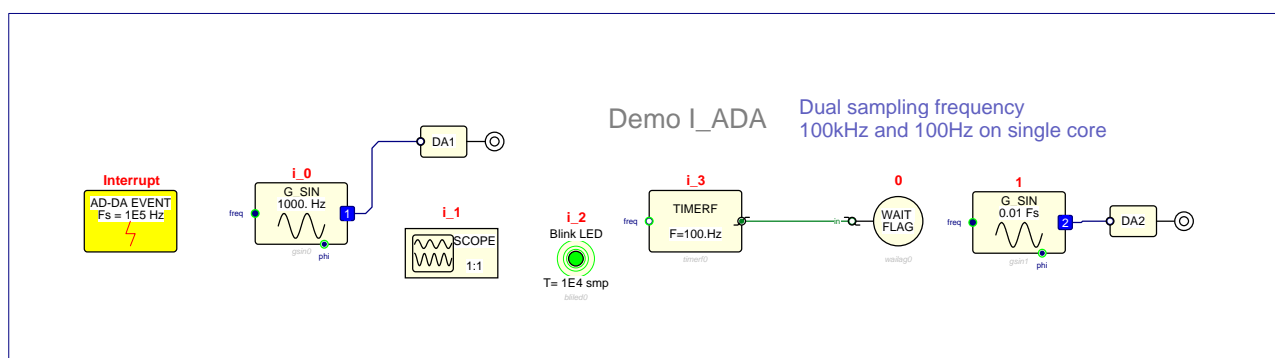
PARAMETERS:

Parameter:
Frequency

Default values:
1E5

ATTRIBUTES

Unique, Execute First, Interrupt Installation, Defines: actual_fs



I_ADA test program



CATEGORY: INTERRUPT

DESCRIPTION:
Install Codec Interrupt

PARAMETERS:

<i>Parameter:</i>	<i>Default values:</i>
Frequency (kHz)	8,32,48,96
Input	line,micro

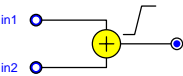
ATTRIBUTES
Unique, Execute First, Interrupt Installation, Defines: actual_fs

I_TIMER0

IADDS

Integer addition with saturation

IADDS



CATEGORY: INTEGER

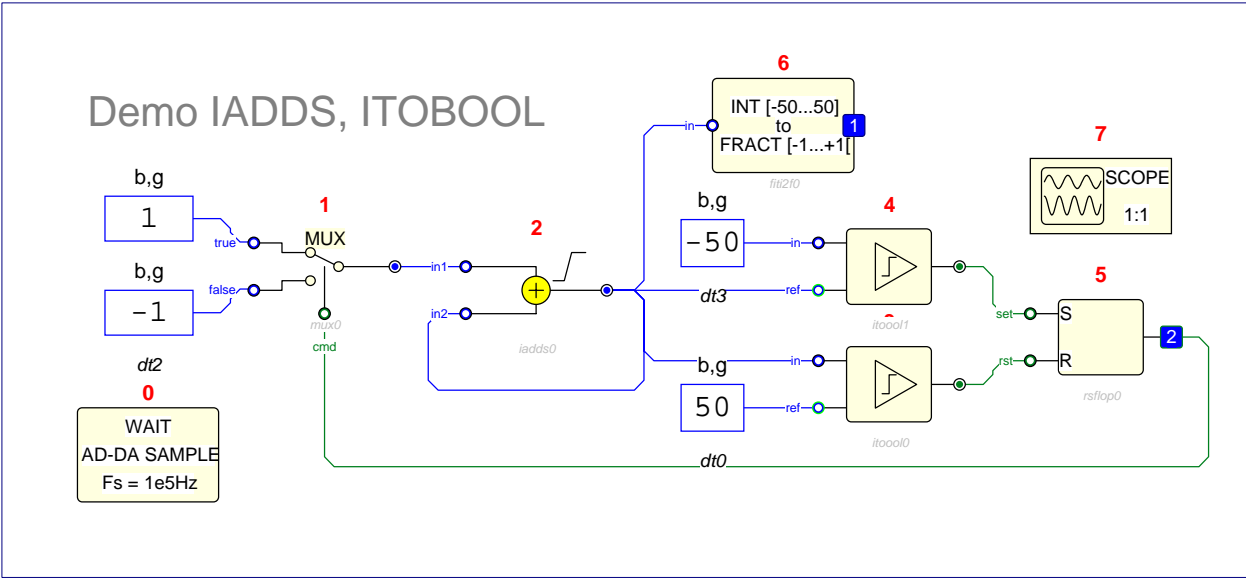
DESCRIPTION:
Integer addition with saturation

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

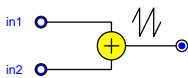


IADDS test program

IADDV

Integer addition modulo 2^{24}

IADDV



CATEGORY: INTEGER

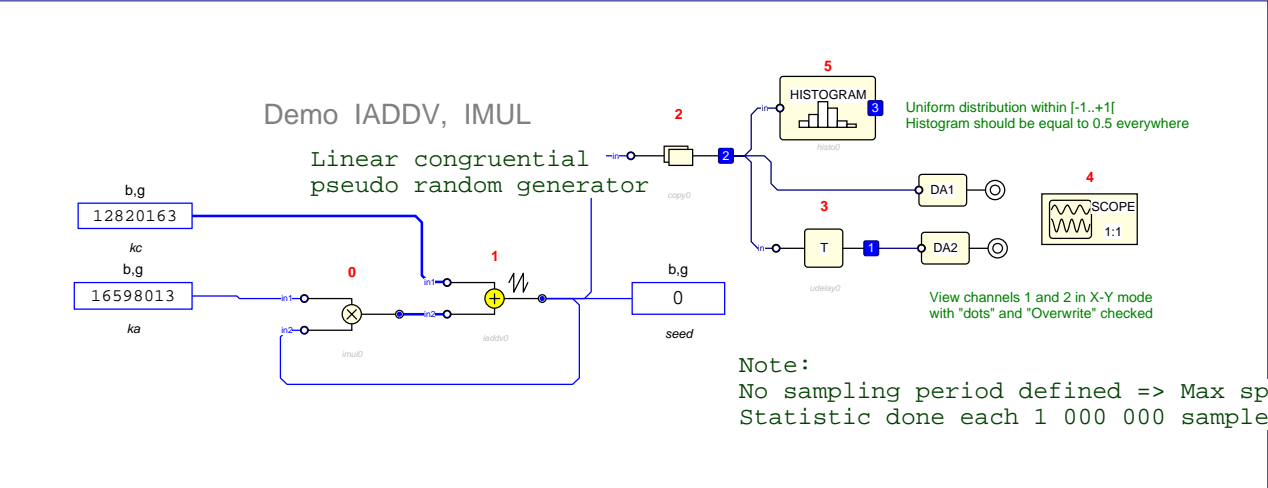
DESCRIPTION:
Integer addition modulo 2^{24}

INPUTS

<i>Name:</i> name_in1	<i>Data Type:</i> INTEGER	<i>Data Struct:</i> WORD	<i>Connection:</i> mandatory
<i>Name:</i> name_in2	<i>Data Type:</i> INTEGER	<i>Data Struct:</i> WORD	<i>Connection:</i> mandatory

OUTPUTS

<i>Name:</i> name	<i>Data Type:</i> INTEGER	<i>Data Struct:</i> WORD	<i>Connection:</i> normal
----------------------	------------------------------	-----------------------------	------------------------------

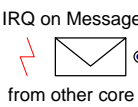


IADDV test program

ICC_IRQ

IRQ on message

ICC_IRQ

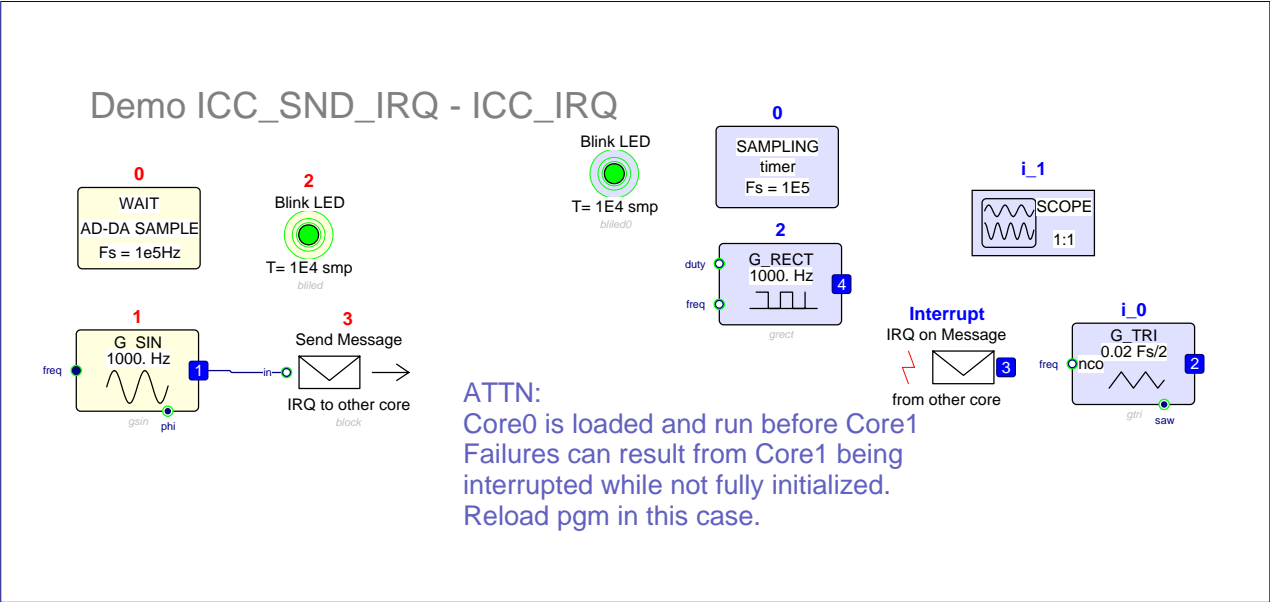


CATEGORY: INTERRUPT

DESCRIPTION:
IRQ on message
Install interrupt generated by message from other core

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

ATTRIBUTES
Unique, Execute First, Interrupt Installation,

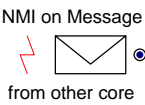


ICC_IRQ test program

ICC_NMI

NMI on message

ICC_NMI

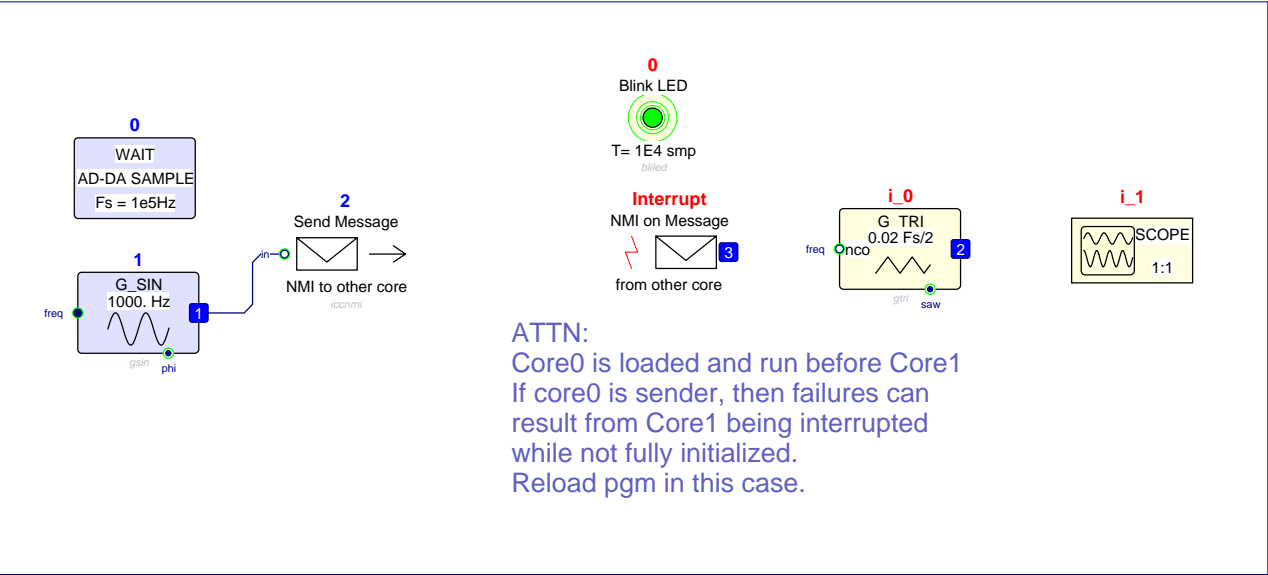


CATEGORY: INTERRUPT

DESCRIPTION:
NMI on message
Install non maskable interrupt generated by message from other core

OUTPUTS			
<i>Name:</i> name	<i>Data Type:</i> FRACT	<i>Data Struct:</i> WORD	<i>Connection:</i> normal

ATTRIBUTES
Unique, Execute First, Interrupt Installation,

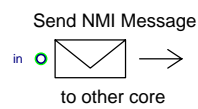


ICC_NMI test program

ICC_SEND

Send Message

ICC_SEND



CATEGORY: INTERRUPT

DESCRIPTION:

Send Message
Transmit a message to other core, without interrupt
Connection to opt input overrides message parameter

PARAMETERS:

<i>Parameter:</i>	<i>Default values:</i>
Message	0

INPUTS

<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name_in	FRACT	WORD	optional

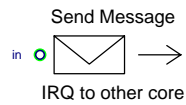
ATTRIBUTES

Unique,

ICC_SND_IRQ

Send Message

ICC_SND_IRQ



CATEGORY: INTERRUPT

DESCRIPTION:

Send Message

Do Interrupt to other core and transmit message

Connection to opt input overrides message parameter

PARAMETERS:

Parameter:

Message

Default values:

0

INPUTS

Name:

name_in

Data Type:

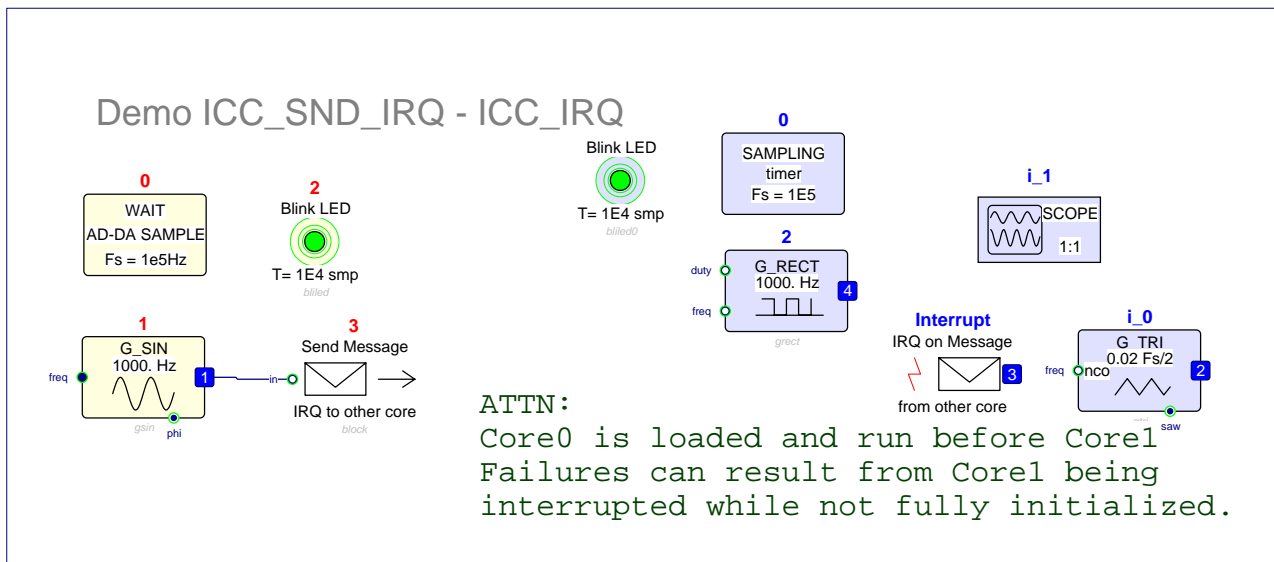
FRACT

Data Struct:

WORD

Connection:

optional

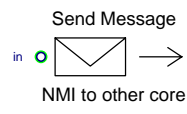


ICC_SND_IRQ test program

ICC_SND_NMI

Send Message

ICC_SND_NMI

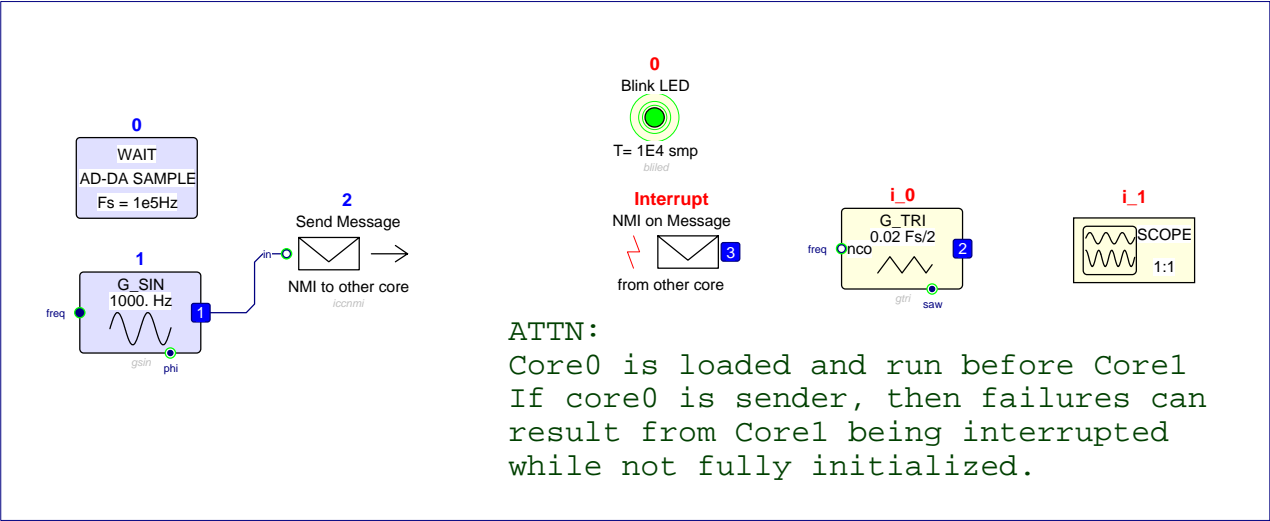


CATEGORY: INTERRUPT

DESCRIPTION:
Send Message
Do Non Masquable Interrupt to other core and transmit message
Connection to opt input overrides message parameter

PARAMETERS:
Parameter: Message
Default values: 0

INPUTS			
<i>Name:</i> name_in	<i>Data Type:</i> FRACT	<i>Data Struct:</i> WORD	<i>Connection:</i> optional

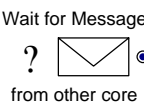


ICC_SND_NMI test program

ICC_WAIT

Wait for Message

ICC_WAIT

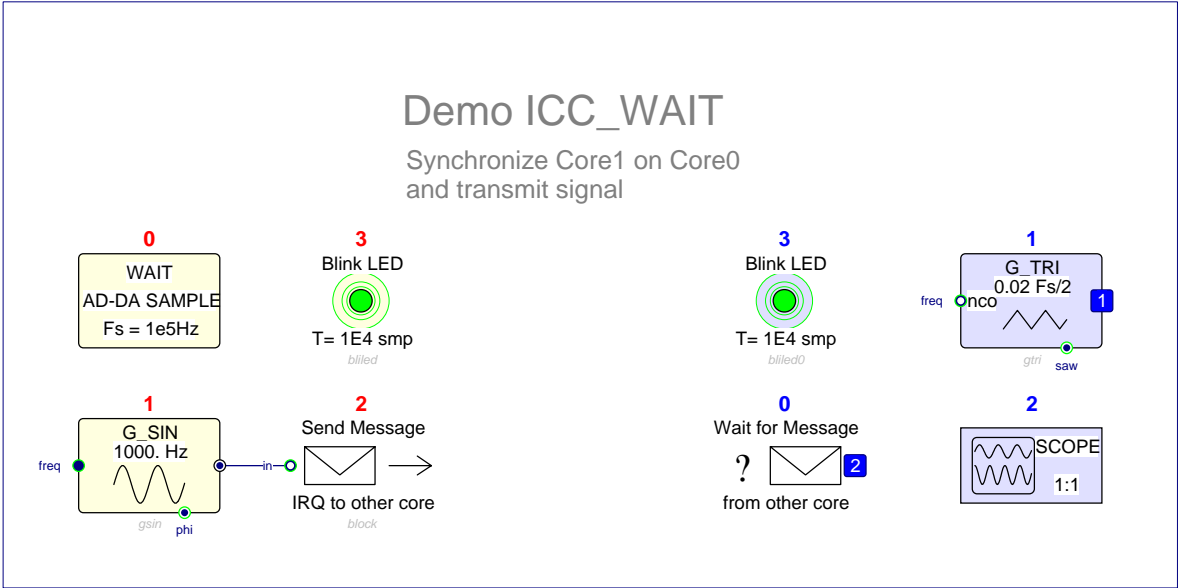


CATEGORY: INTERRUPT

DESCRIPTION:
Wait for Message
Wait for interrupt and message generated by other core

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

ATTRIBUTES
Unique, Execute First,

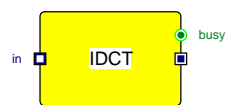


ICC_WAIT test program

IDCT

Inverse DCT

IDCT



CATEGORY: MATRIX

DESCRIPTION:
Inverse DCT

INPUTS

Name:
name_in

Data Type:
FRACT

Data Struct:
Matrix of WORD

Connection:
mandatory

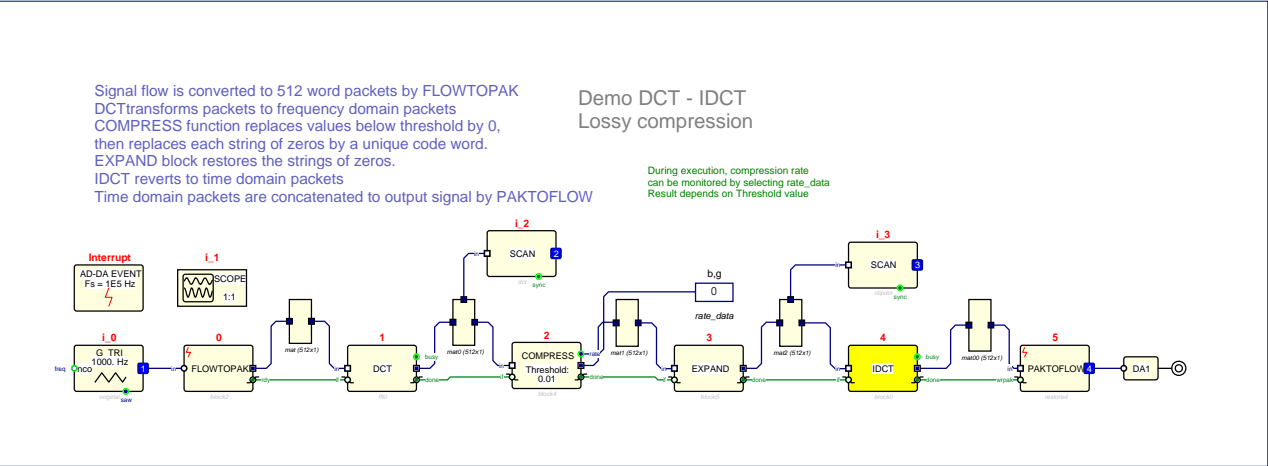
OUTPUTS

Name:
name
name_busy

Data Type:
FRACT
BOOL

Data Struct:
Matrix of WORD
BIT

Connection:
normal
optional



IDCT test program

IDFT

Inverse DFT

IDFT



CATEGORY: MATRIX

DESCRIPTION:
Inverse DFT
slow

INPUTS

Name:
name_in

Data Type:
COMPLEX

Data Struct:
Matrix of DWORD

Connection:
mandatory

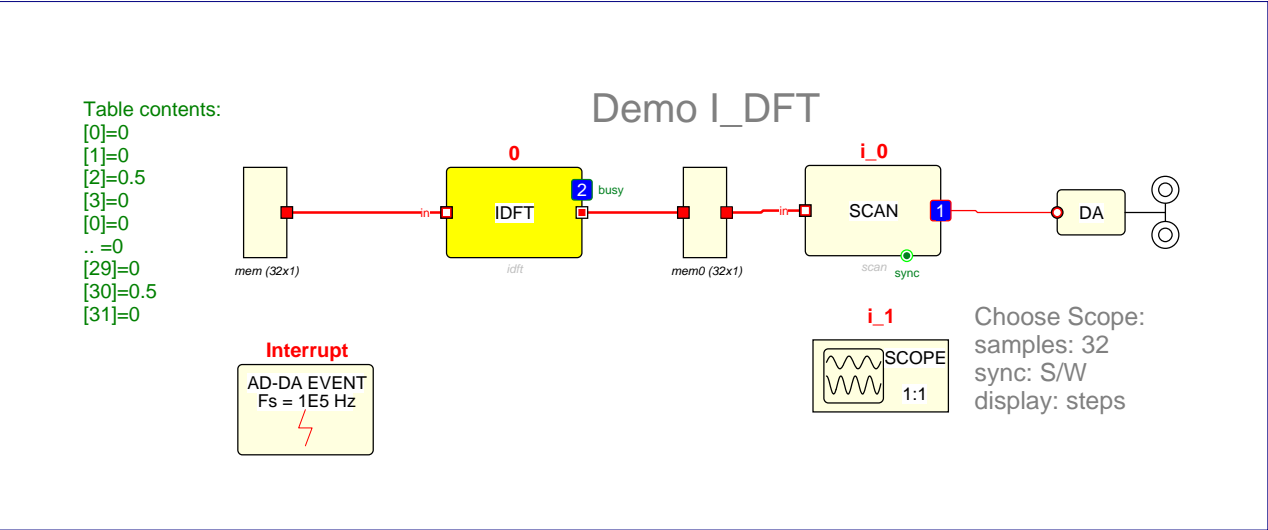
OUTPUTS

Name:
name
name_busy

Data Type:
COMPLEX
BOOL

Data Struct:
Matrix of DWORD
BIT

Connection:
normal
optional



IDFT test program

IFFT

Inverse FFT

IFFT



CATEGORY: MATRIX

DESCRIPTION:
Inverse FFT

INPUTS

Name:
name_in

Data Type:
COMPLEX

Data Struct:
Matrix of DWORD

Connection:
mandatory

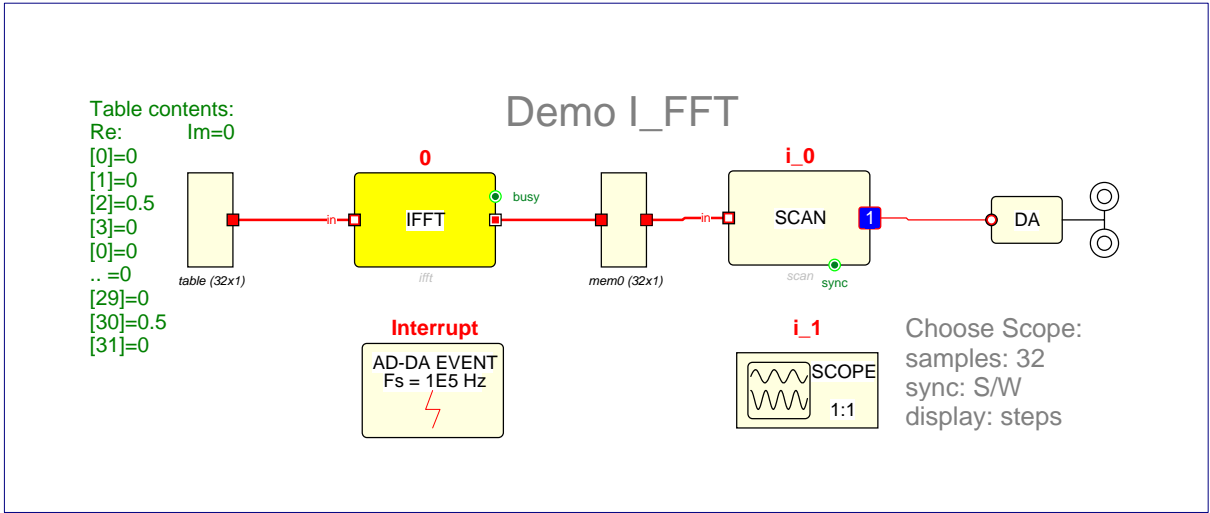
OUTPUTS

Name:
name
name_busy

Data Type:
COMPLEX
BOOL

Data Struct:
Matrix of DWORD
BIT

Connection:
normal
optional

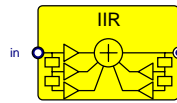


IFFT test program

IIR

2nd order IIR filter

IIR



CATEGORY: FILTERS

DESCRIPTION:

2nd order IIR filter

Transfer function:

$$(b_0 + b_1/z + b_2/z^2)/(1 - a_1/z - a_2/z^2)$$

PARAMETERS:

Parameter:

Default values:

b0	0.109
b1	0.218
b2	0.109
a1	1.056
a2	-0.493

INPUTS

Name:
name_in

Data Type:
FRACT

Data Struct:
WORD

Connection:
mandatory

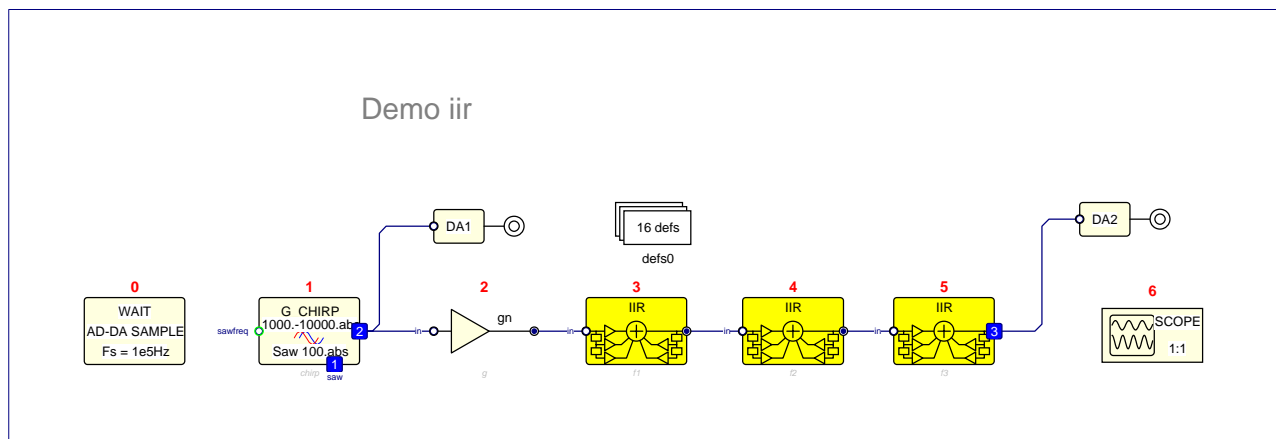
OUTPUTS

Name:
name

Data Type:
FRACT

Data Struct:
WORD

Connection:
normal

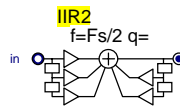


IIR test program

IIR2

2nd order recursive filter

IIR2



CATEGORY: FILTERS

DESCRIPTION:

2nd order recursive filter

PARAMETERS:

Parameter:

Filter type
Frequency
Q factor
Unit

Default values:

lp,bp,hp,bs
1000.
1.2
Hz,Fs/2

INPUTS

Name:
name_in

Data Type:
FRACT

Data Struct:
WORD

Connection:
mandatory

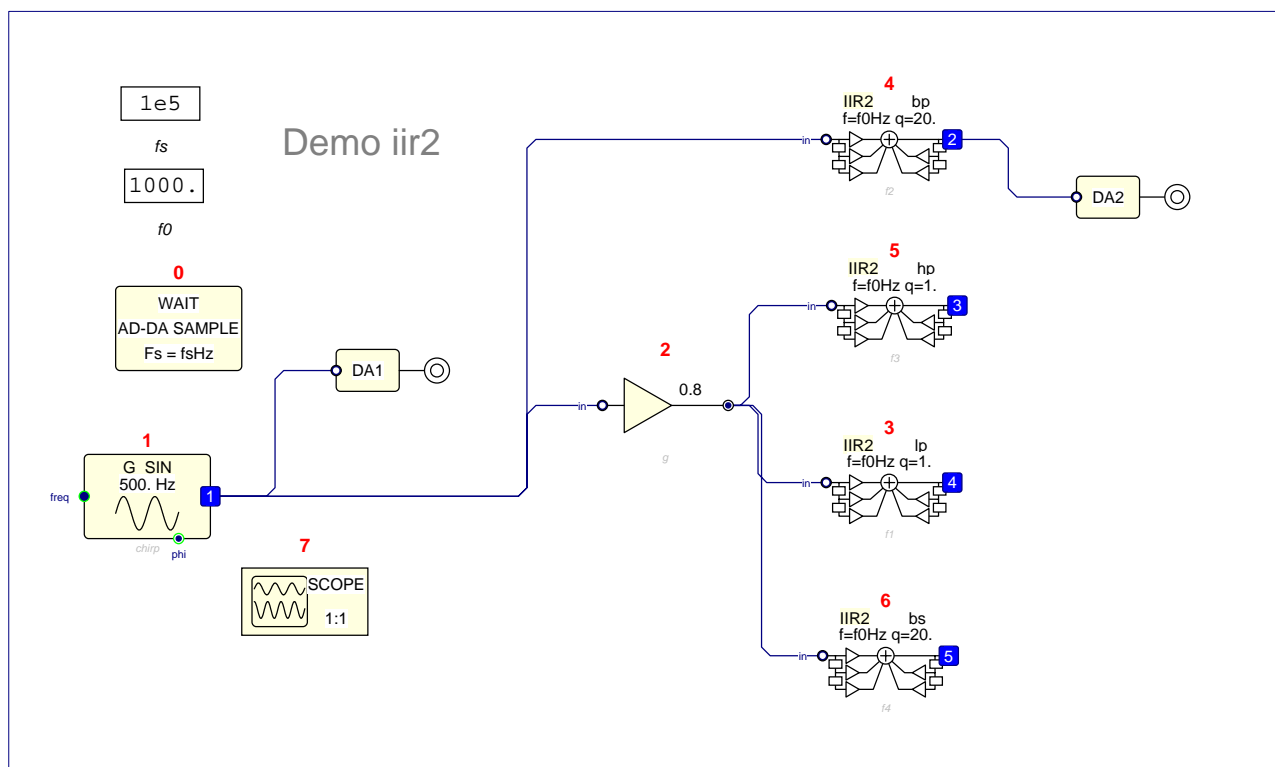
OUTPUTS

Name:
name

Data Type:
FRACT

Data Struct:
WORD

Connection:
normal



IIR2 test program

IIR6

6th order IIR

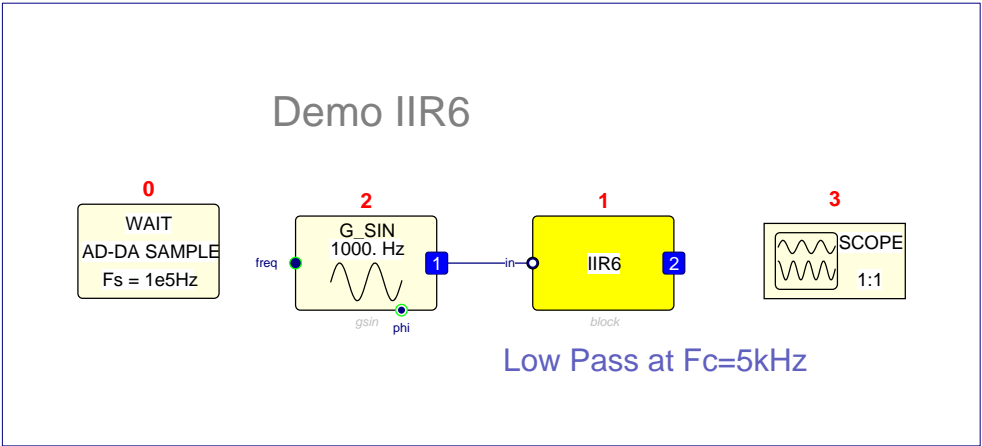
IIR6



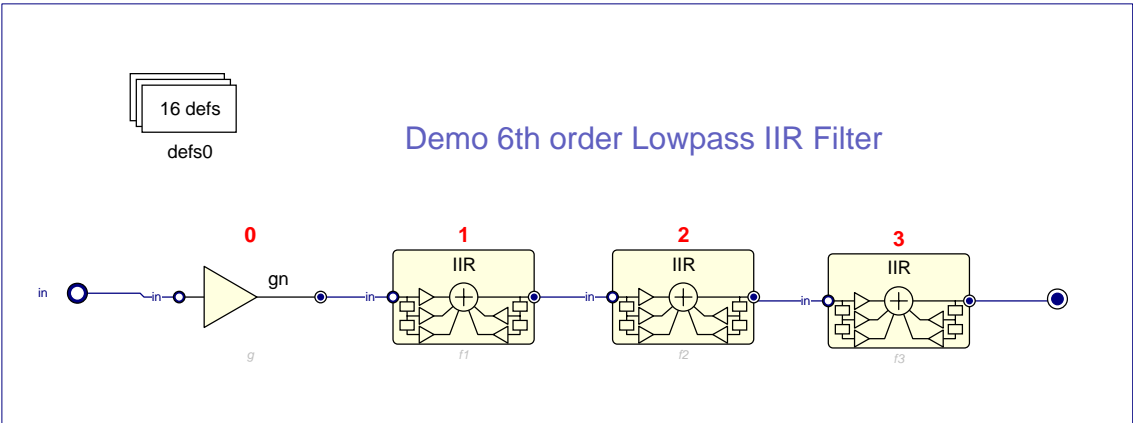
CATEGORY: FILTERS

DESCRIPTION:
6th order IIR
Lowpass with $F_c=5\text{kHz}$ at $F_s=100\text{kHz}$

INPUTS			
Name:	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name_in	FRACT	WORD	mandatory
OUTPUTS			
Name:	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name	FRACT	WORD	normal



IIR6 test program

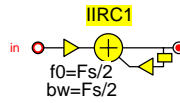


IIR6 internal schema

IIRC1

1st order Complex IIR filter.

IIRC1



CATEGORY: FILTERS

DESCRIPTION:
1st order Complex IIR filter.

PARAMETERS:

Parameter:
Resonance frequency
Bandwidth
Unit

Default values:
1000.
10.
Hz, $F_s/2$

INPUTS

Name:
name_in

Data Type:
COMPLEX

Data Struct:
WORD

Connection:
mandatory

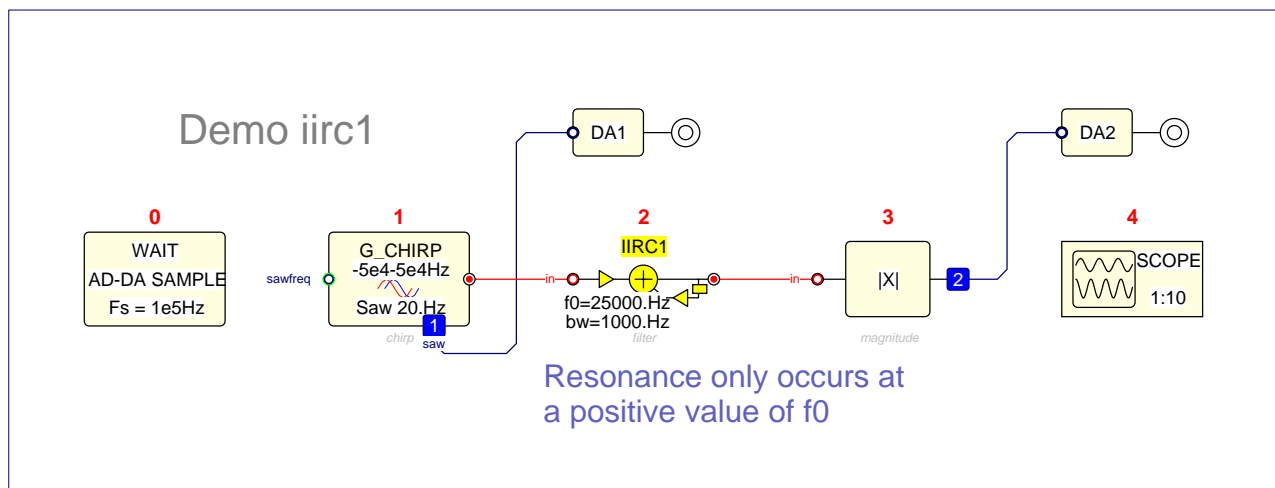
OUTPUTS

Name:
name

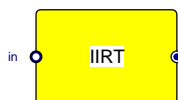
Data Type:
COMPLEX

Data Struct:
WORD

Connection:
normal



IIRC1 test program



CATEGORY: FILTERS

DESCRIPTION:

2nd order IIR Transposed IIR
 $Y(z)=[b_0 + b_1/z + b_2/z^2]/[1 + a_1/z + a_2/z^2]$

PARAMETERS:

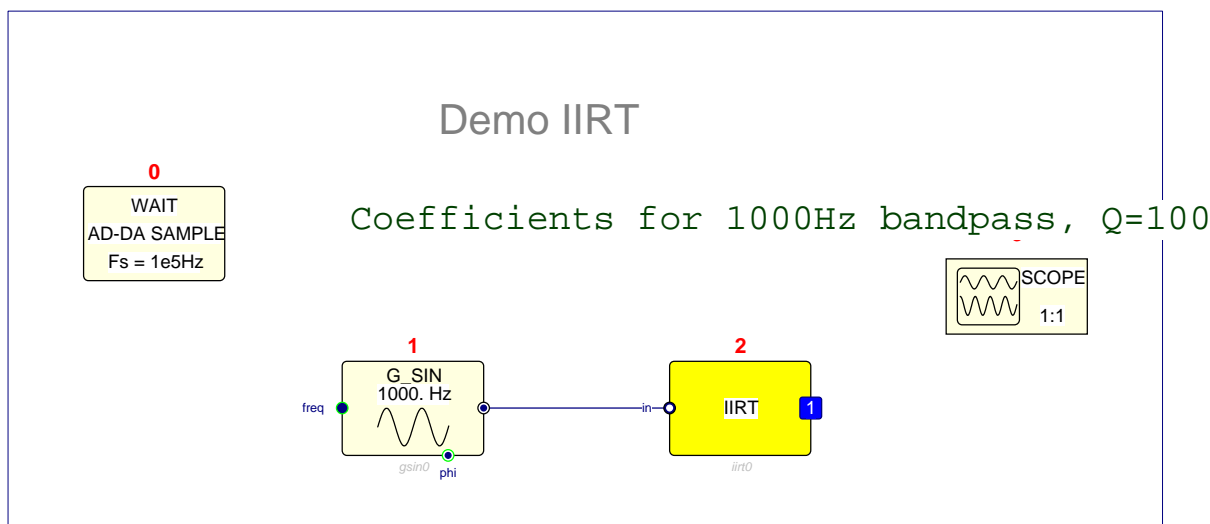
<i>Parameter:</i>	<i>Default values:</i>
b0	
b1	
b2	
a1	
a2	

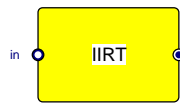
INPUTS

<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name_in	FRACT	WORD	mandatory

OUTPUTS

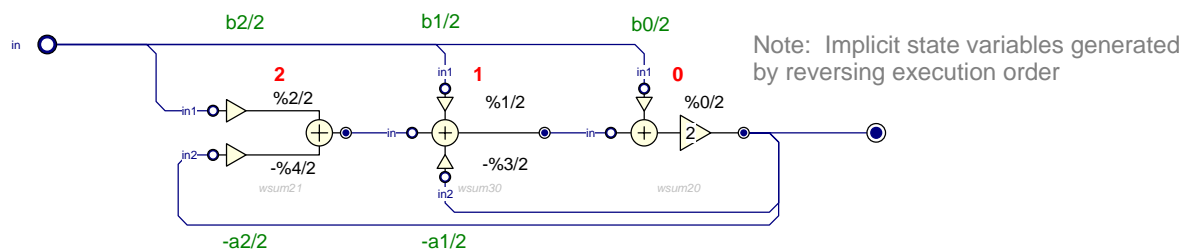
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name	FRACT	WORD	normal



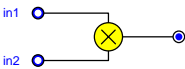


2nd order IIR Transposed canonic form

$$H(z) = (b_0 + b_1/z + b^2/z^2) / (1 + a_1/z + a_2/z^2)$$



IIRT internal schema



CATEGORY: INTEGER

DESCRIPTION:
Integer multiplier

INPUTS

Name:
name_in1
name_in2

Data Type:
INTEGER
INTEGER

Data Struct:
WORD
WORD

Connection:
mandatory
mandatory

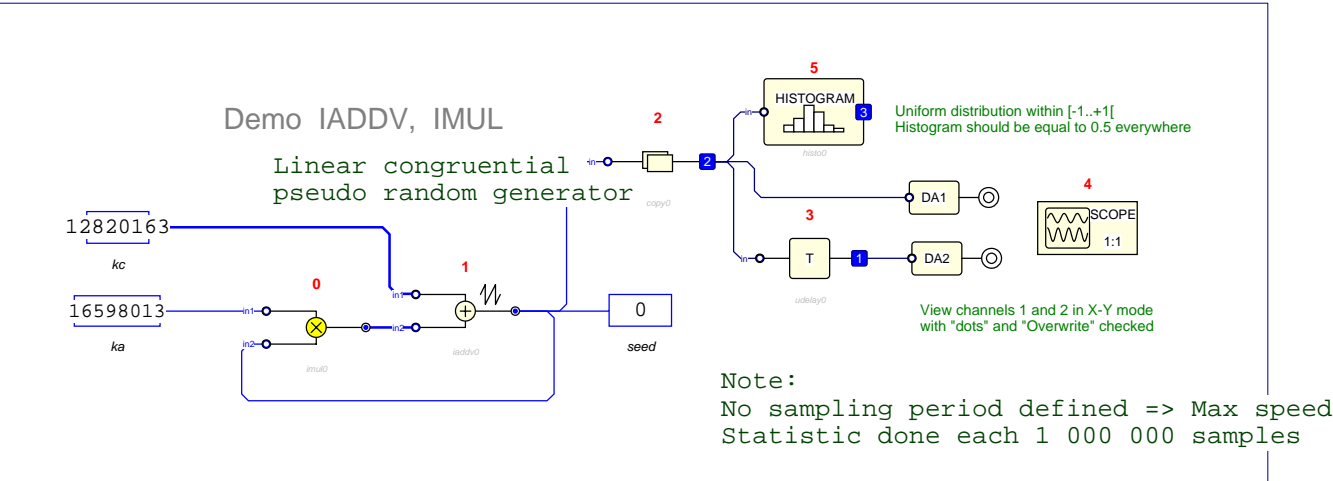
OUTPUTS

Name:
name

Data Type:
INTEGER

Data Struct:
WORD

Connection:
normal



IMUL test program

IN_L

Codec input Left

IN_L



CATEGORY: AUDIO

DESCRIPTION:
Codec input Left
Result of A to D conversion

OUTPUTS

Name:
name

Data Type:
FRACT

Data Struct:
WORD

Connection:
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:
name

Data Type:
FRACT

Data Struct:
WORD

Connection:
normal

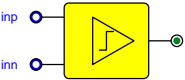
ATTRIBUTES

Non executable, Unique,

INTCOMP

Integer Comparator

INTCOMP



CATEGORY: LOGIC

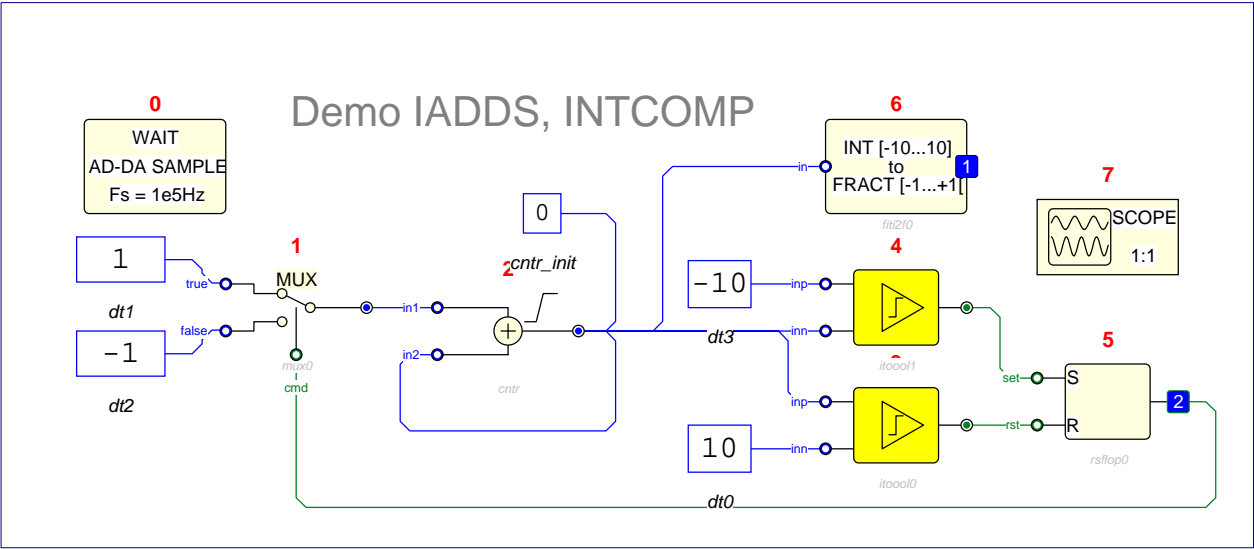
DESCRIPTION:
Integer Comparator
boolean output
Result is True if $inp > inn$

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

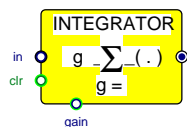


INTCOMP test program

INTEG

Integrator with input gain

INTEG



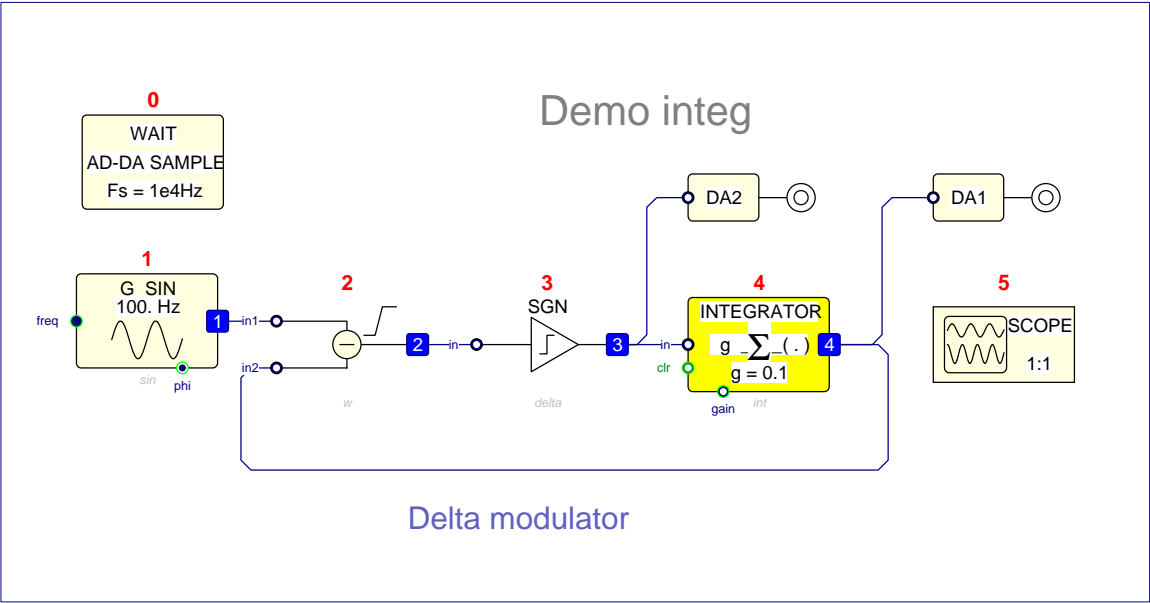
CATEGORY: CONTROL

DESCRIPTION:
Integrator with input gain
 $y(k) = y(k-1) + g \cdot x(k)$

PARAMETERS:
Parameter: gain
Default values: 0.001

INPUTS	Data Type:	Data Struct:	Connection:
Name:			
name_in	FRACT	WORD	mandatory
name_clr	BOOL	BIT	optional
name_gain	FRACT	WORD	optional

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

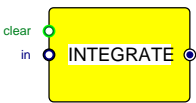


INTEG test program

INTEGA

Analog Integrator

INTEGA



CATEGORY: CONTINUOUS

DESCRIPTION:

Analog Integrator
 $y = \text{Integral}[\text{gain} \cdot \text{in}(t) dt]$
Gain 1.0 with input 1.0 generates slope 1.0 U/sec

PARAMETERS:

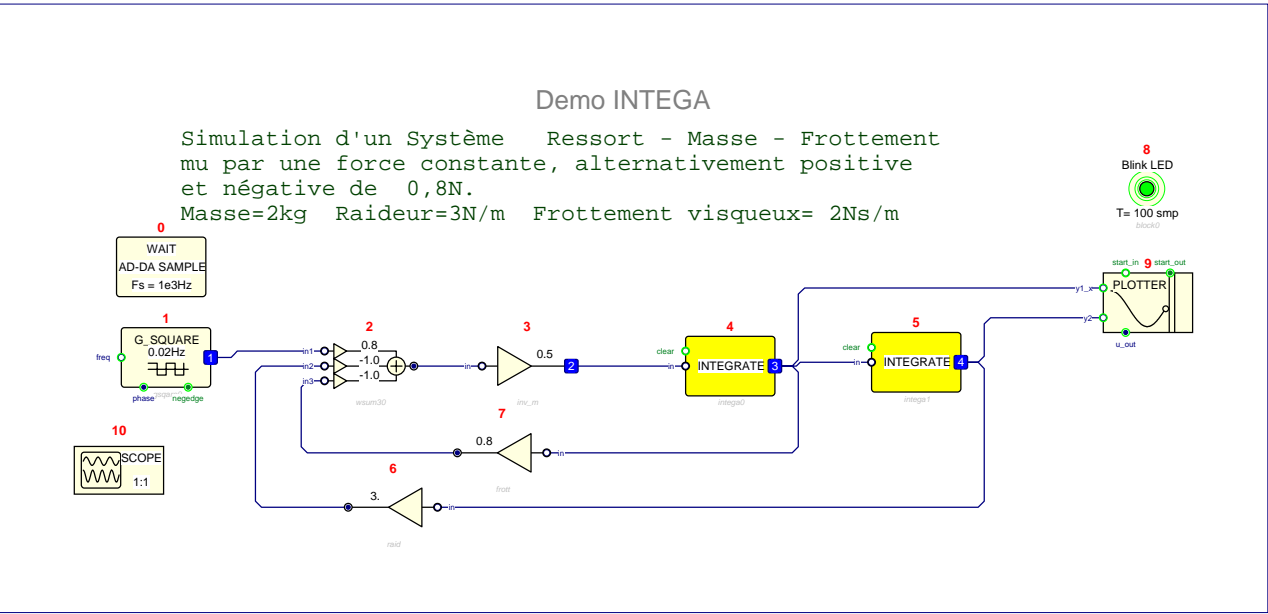
<i>Parameter:</i>	<i>Default values:</i>
Gain U/sec	1.0

INPUTS

<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name_clear	BOOL	BIT	optional
name_in	FRACT	WORD	mandatory

OUTPUTS

<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name	FRACT	WORD	normal

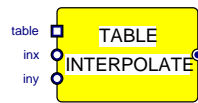


INTEGA test program

INTERPOL

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:

name_inx
name_table
name_iny

Data Type:

FRACT
FRACT
FRACT

Data Struct:

WORD
Matrix of WORD
WORD

Connection:

mandatory
mandatory
mandatory

OUTPUTS

Name:

name

Data Type:

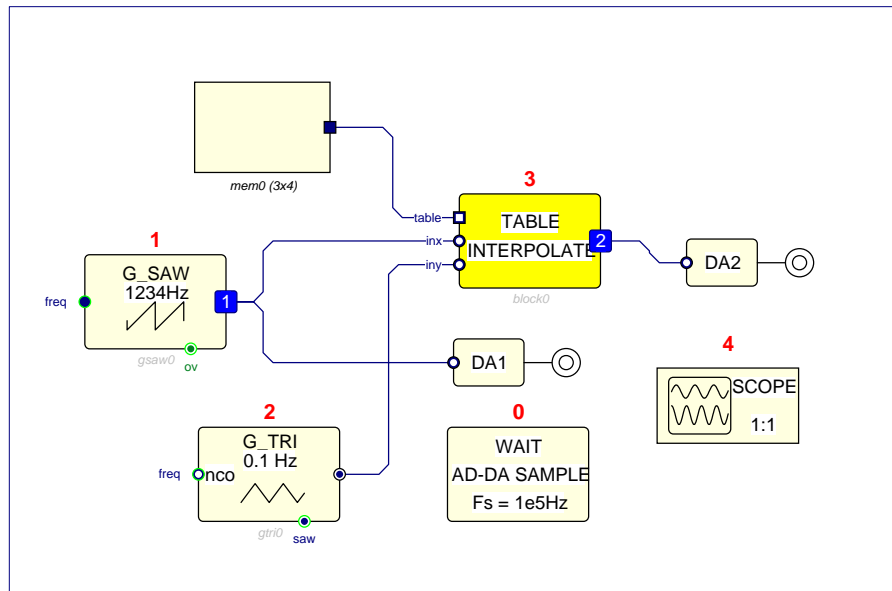
FRACT

Data Struct:

WORD

Connection:

normal

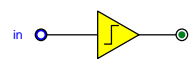


INTERPOL test program

INTTOBOOL

Comparator of Integers

INTTOBOOL



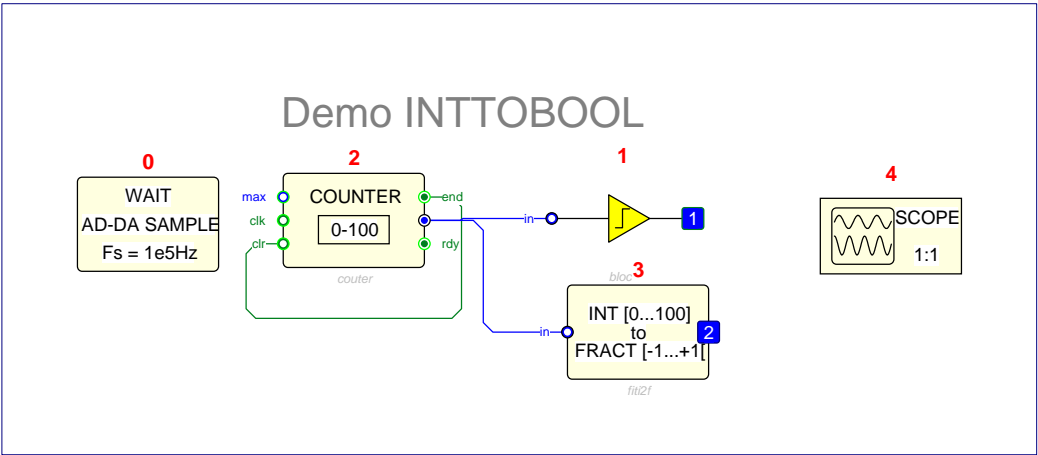
CATEGORY: LOGIC

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

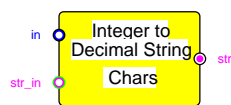


INTTOBOOL test program

INTTOSTR

Integer to String

INTTOSTR

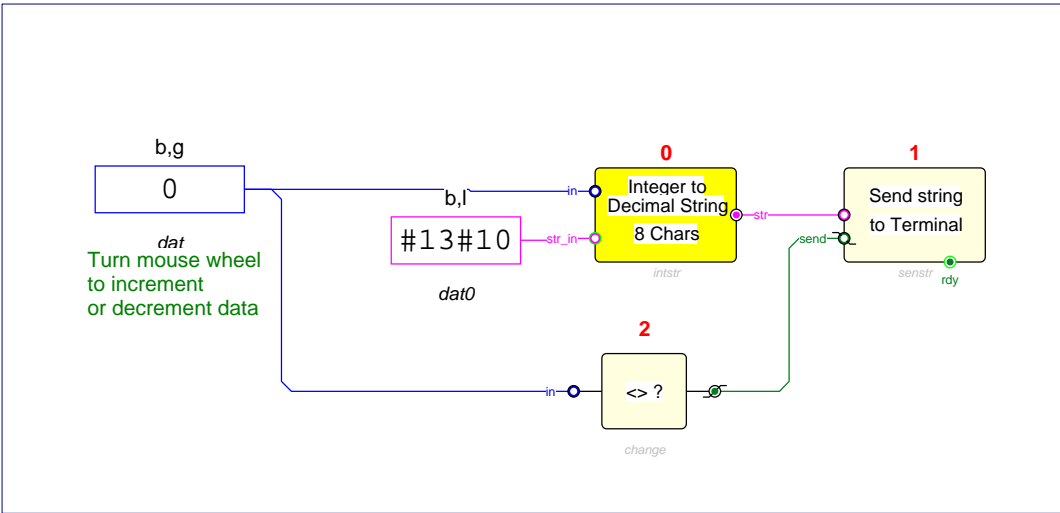


CATEGORY: STRING

DESCRIPTION:
Integer to String
Convert Signed Integer input to decimal string

PARAMETERS:
Parameter: Nb chars (2..8) Default values: 8

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

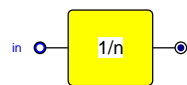


INTTOSTR test program

INVINT

Inverse of an integer

INVINT



CATEGORY: INTEGER

DESCRIPTION:
Inverse of an integer
Result is fractional.

INPUTS

Name:
name_in

Data Type:
INTEGER

Data Struct:
WORD

Connection:
mandatory

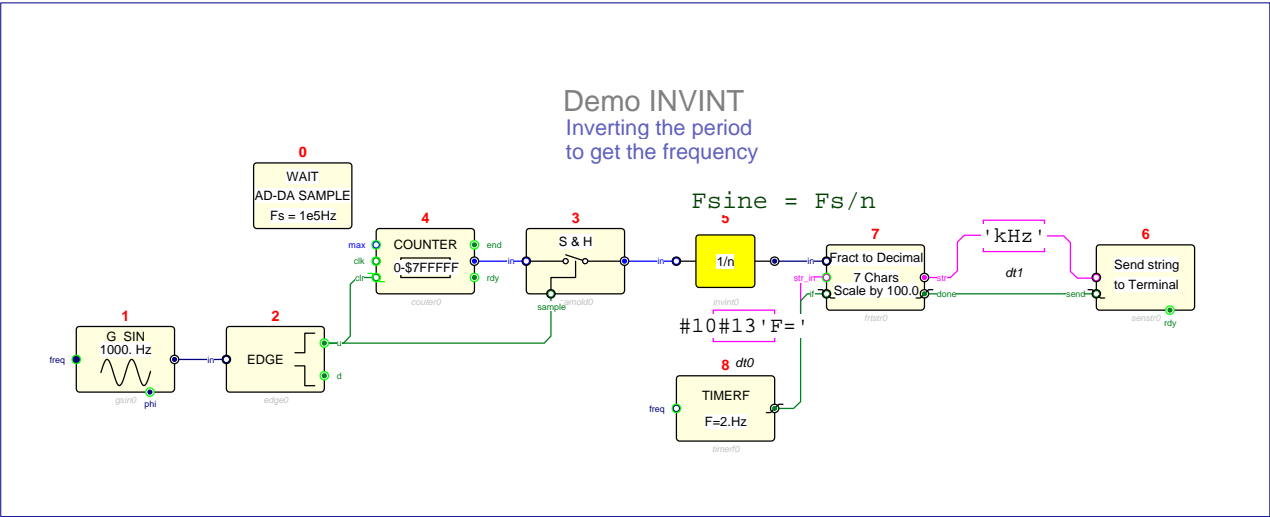
OUTPUTS

Name:
name

Data Type:
FRACT

Data Struct:
WORD

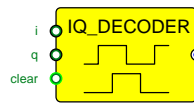
Connection:
normal



INVINT test program

IQ_DECODER Incremental decoder

IQ_DECODER



CATEGORY: LOGIC

DESCRIPTION:
Incremental decoder
for phase-quadrature logic signals

INPUTS

Name:
name_i
name_q
name_clear

Data Type:
BOOL
BOOL
BOOL

Data Struct:
BIT
BIT
BIT

Connection:
mandatory
mandatory
optional

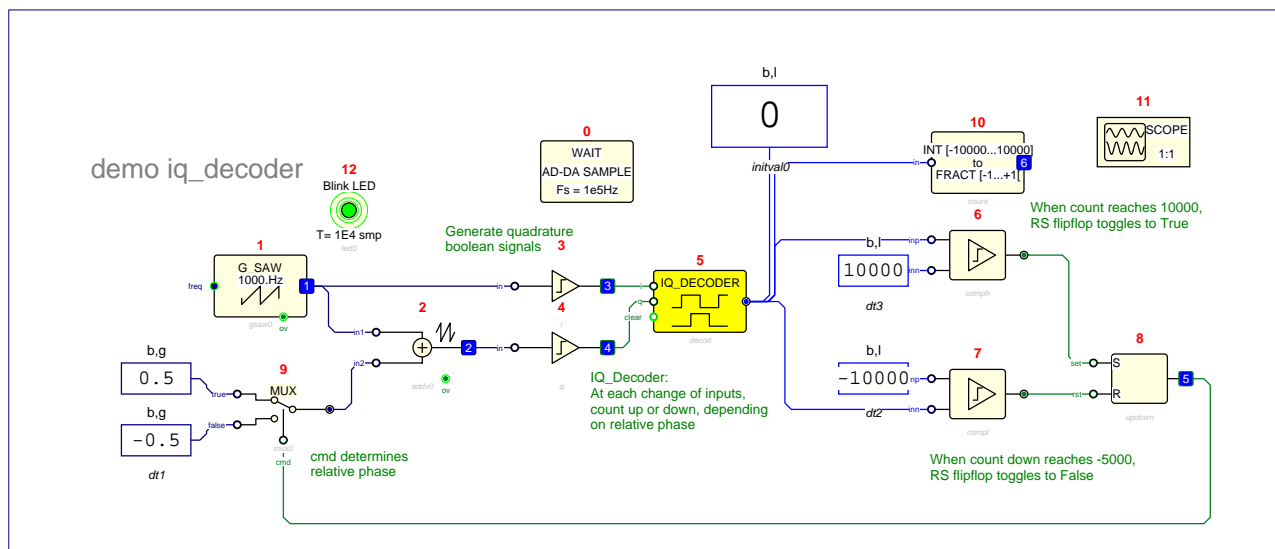
OUTPUTS

Name:
name

Data Type:
INTEGER

Data Struct:
WORD

Connection:
normal

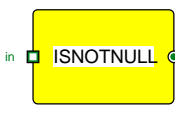


IQ_DECODER test program

ISNOTNULL

Test bool vector

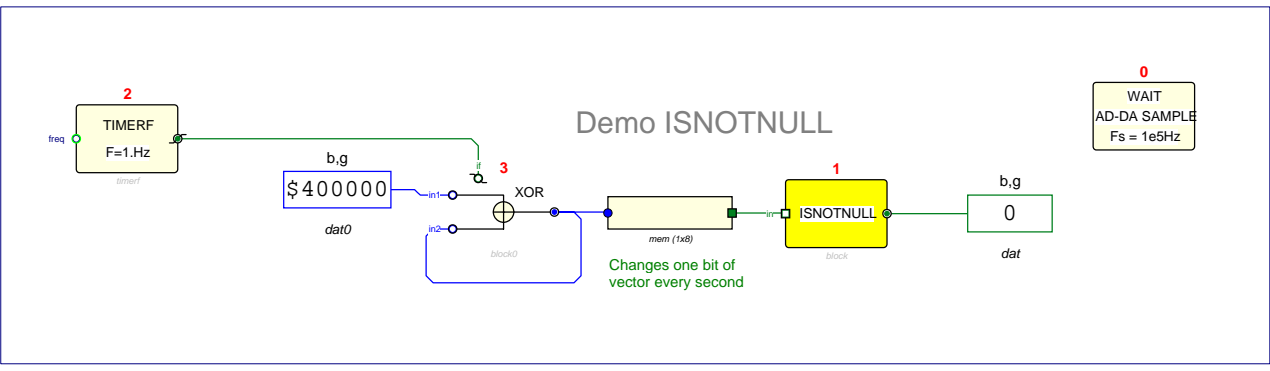
ISNOTNULL



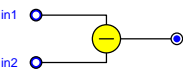
CATEGORY: MATRIX

DESCRIPTION:
Test bool vector
Test line boolean vector for non nul

INPUTS			
Name:			
name_in	Data Type: BOOL	Data Struct: Matrix of BIT	Connection: mandatory
OUTPUTS			
Name:			
name	Data Type: BOOL	Data Struct: BIT	Connection: normal



ISNOTNULL test program



CATEGORY: INTEGER

DESCRIPTION:
Integer Subtraction

INPUTS

Name:
name_in1
name_in2

Data Type:
INTEGER
INTEGER

Data Struct:
WORD
WORD

Connection:
mandatory
mandatory

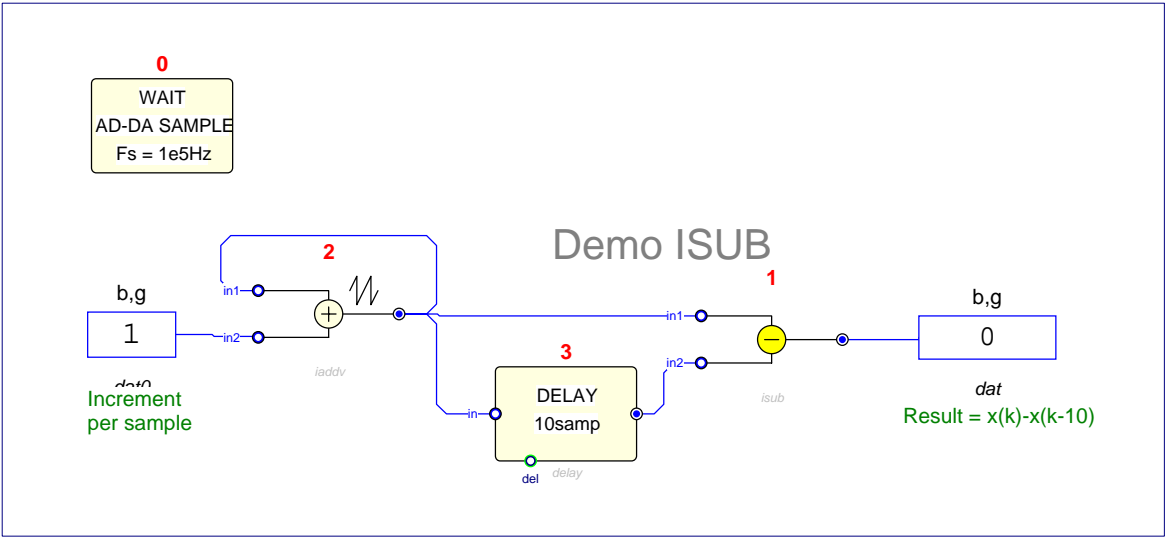
OUTPUTS

Name:
name

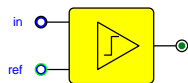
Data Type:
INTEGER

Data Struct:
WORD

Connection:
normal



ISUB test program



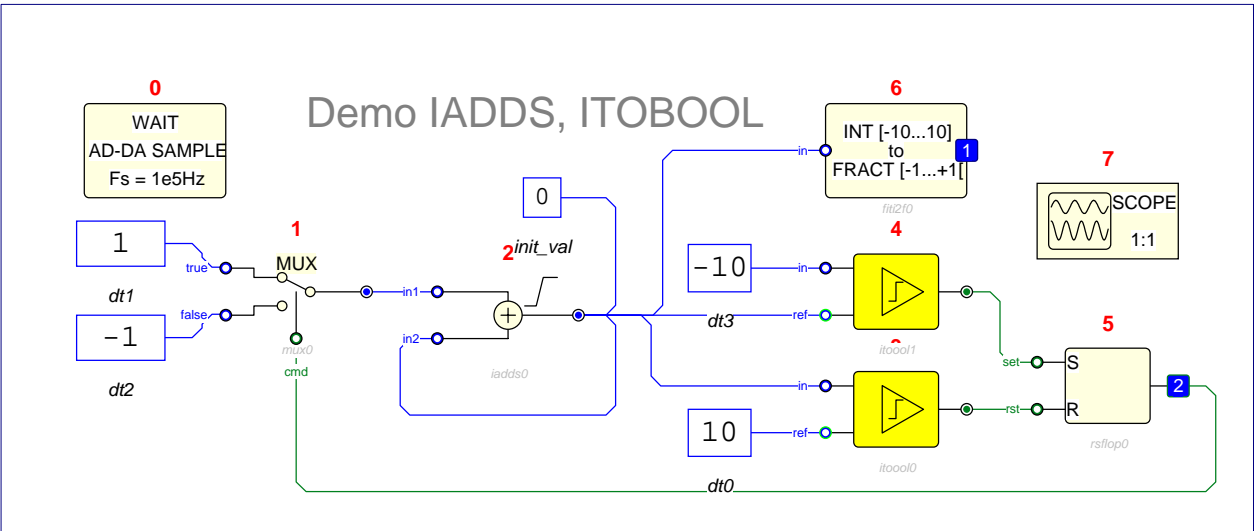
CATEGORY: INTEGER

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

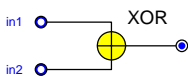


ITOBOOL test program

IXOR

Exclusive OR

IXOR



DESCRIPTION:
Exclusive OR
between 2 words

INPUTS

Name:
name_in1
name_in2

Data Type:
INTEGER
INTEGER

Data Struct:
WORD
WORD

Connection:
mandatory
mandatory

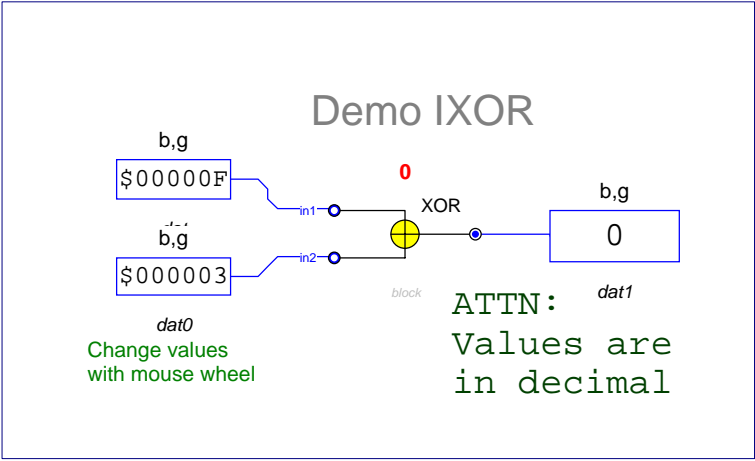
OUTPUTS

Name:
name

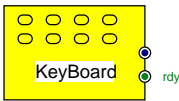
Data Type:
INTEGER

Data Struct:
WORD

Connection:
normal



IXOR test program



CATEGORY: STRING

DESCRIPTION:
Get ASCII from keyboard

OUTPUTS

Name:
name
name_rdy

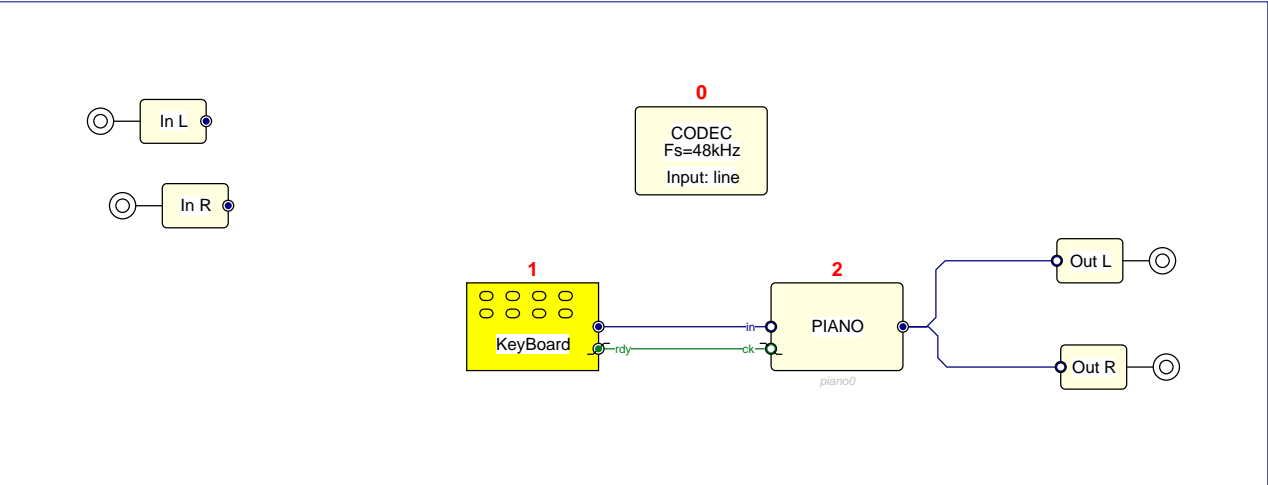
Data Type:
FRACT
BOOL

Data Struct:
WORD
BIT

Connection:
normal
normal

ATTRIBUTES

Unique,



KBD test program

KEY_EVENT

Key Press Event

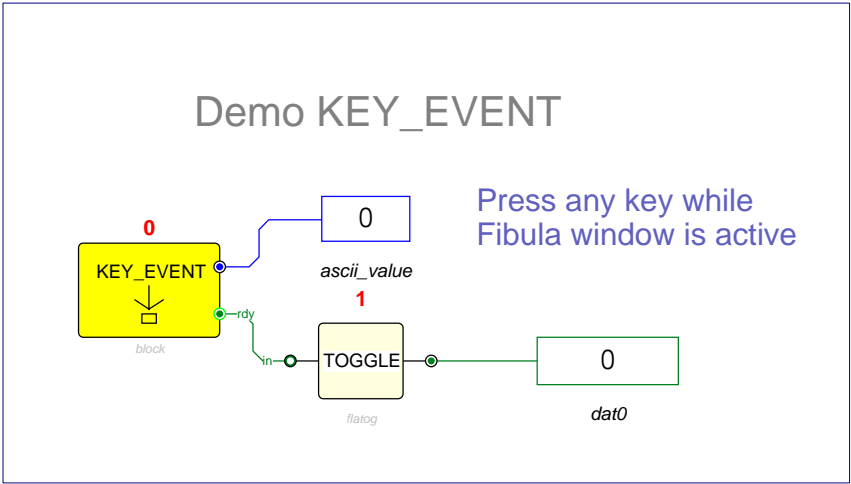
KEY_EVENT



CATEGORY: CONTROL

DESCRIPTION:
Key Press Event
Returns key ascii code and optional Bool rdy=True when key is pressed

OUTPUTS			
<i>Name:</i> name name_rdy	<i>Data Type:</i> INTEGER BOOL	<i>Data Struct:</i> WORD BIT	<i>Connection:</i> normal optional

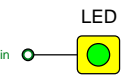


KEY_EVENT test program

LED

LED

LED



CATEGORY: ETD410K

DESCRIPTION:

LED
Set LED ON if input is TRUE
Set LED off otherwise

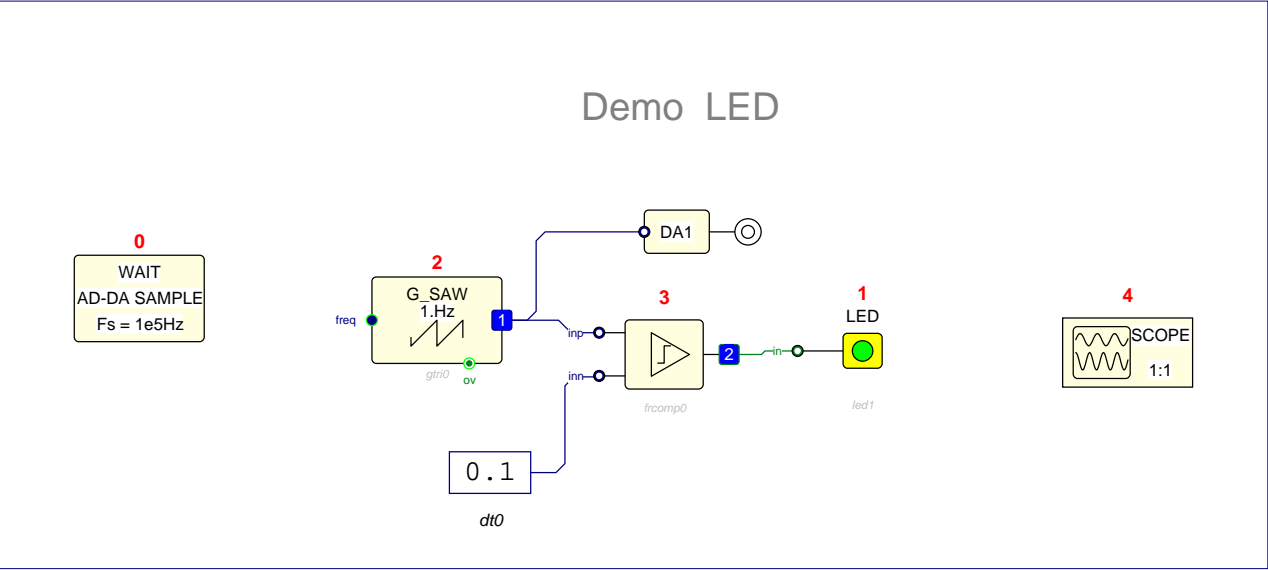
INPUTS

Name:
name_in

Data Type:
BOOL

Data Struct:
BIT

Connection:
mandatory

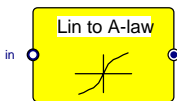


LED test program

LINTOALAW

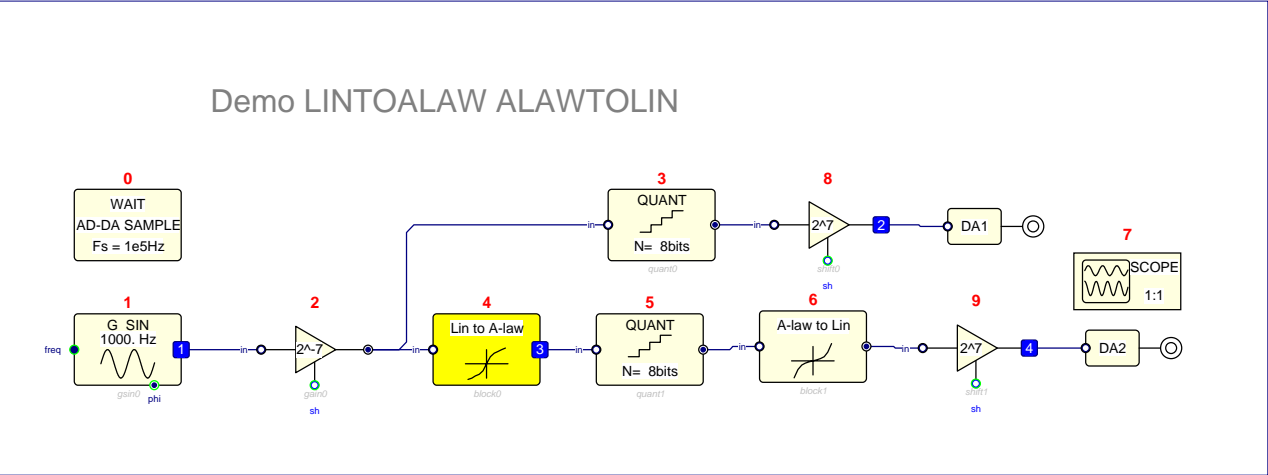
Linear to A-Law conversion

LINTOALAW

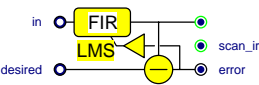


DESCRIPTION:
Linear to A-Law conversion

INPUTS			
Name:	Data Type:	Data Struct:	Connection:
name_in	FRACT	WORD	mandatory
OUTPUTS			
Name:	Data Type:	Data Struct:	Connection:
name	FRACT	WORD	normal



LINTOALAW test program



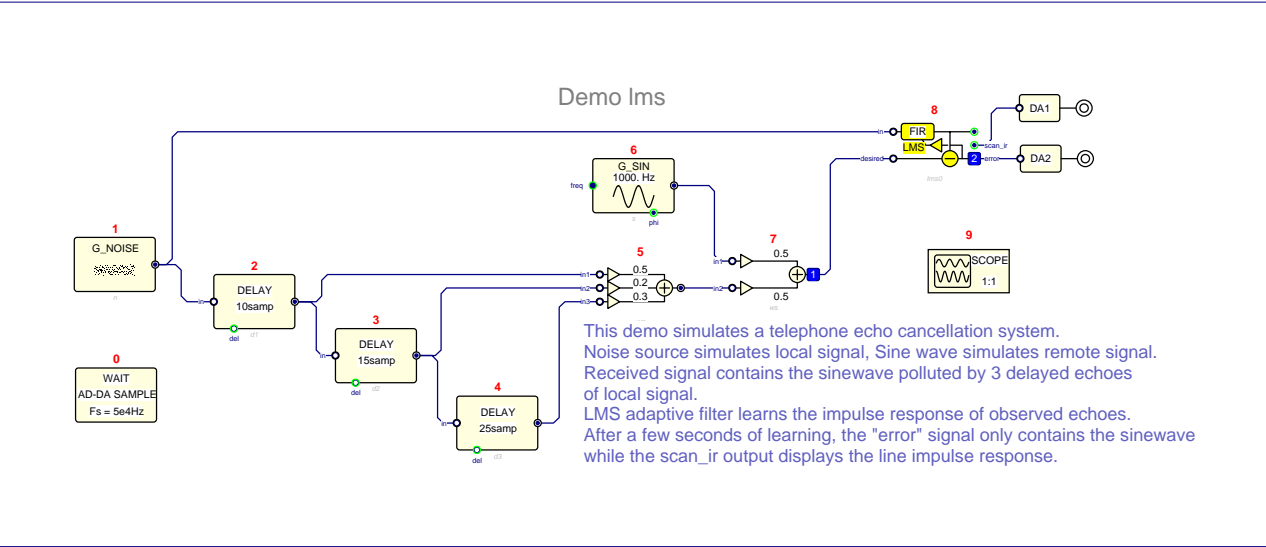
CATEGORY: FILTERS

DESCRIPTION:
Auto Adaptive FIR filter.
Impulse response evolves 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

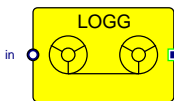


LMS test program

LOGG

Data Logger.

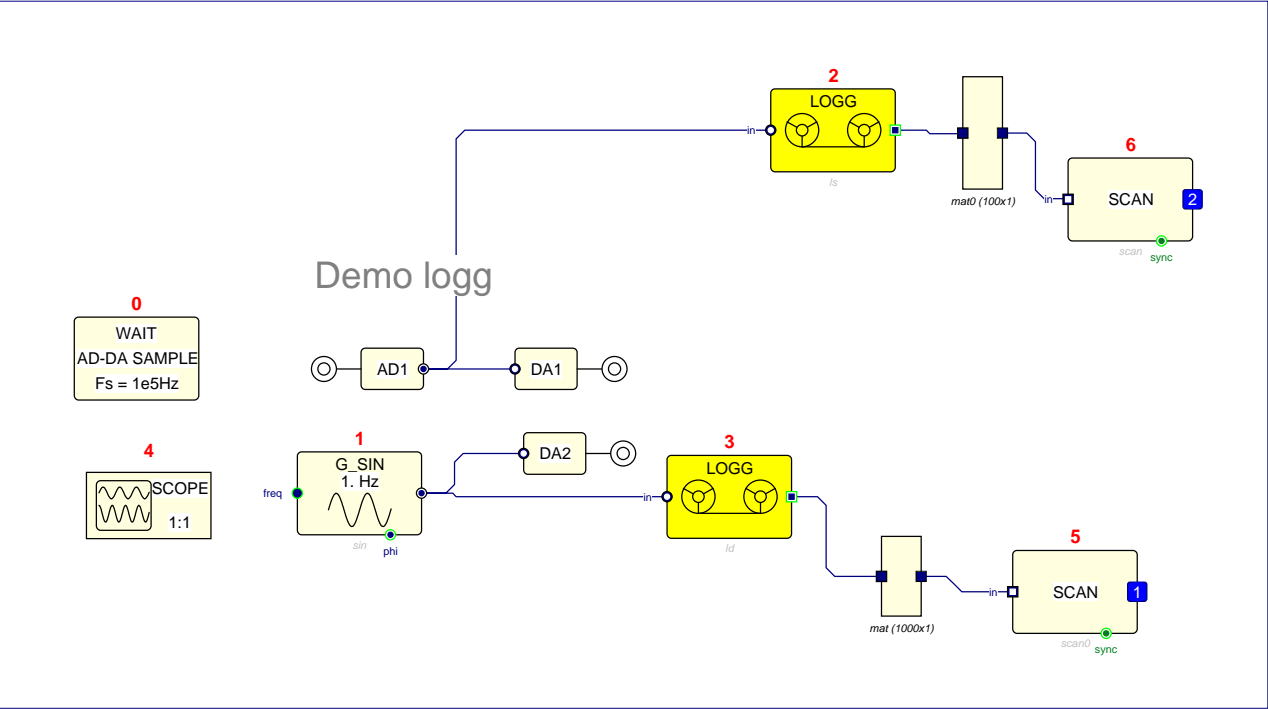
LOGG



CATEGORY: INSTRUMENTS

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

INPUTS			
Name:	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name_in	FRACT	WORD	mandatory
OUTPUTS			
Name:	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name	FRACT	Matrix of WORD	optional

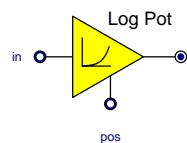


LOGG test program

LOGPOT

Log potentiometer

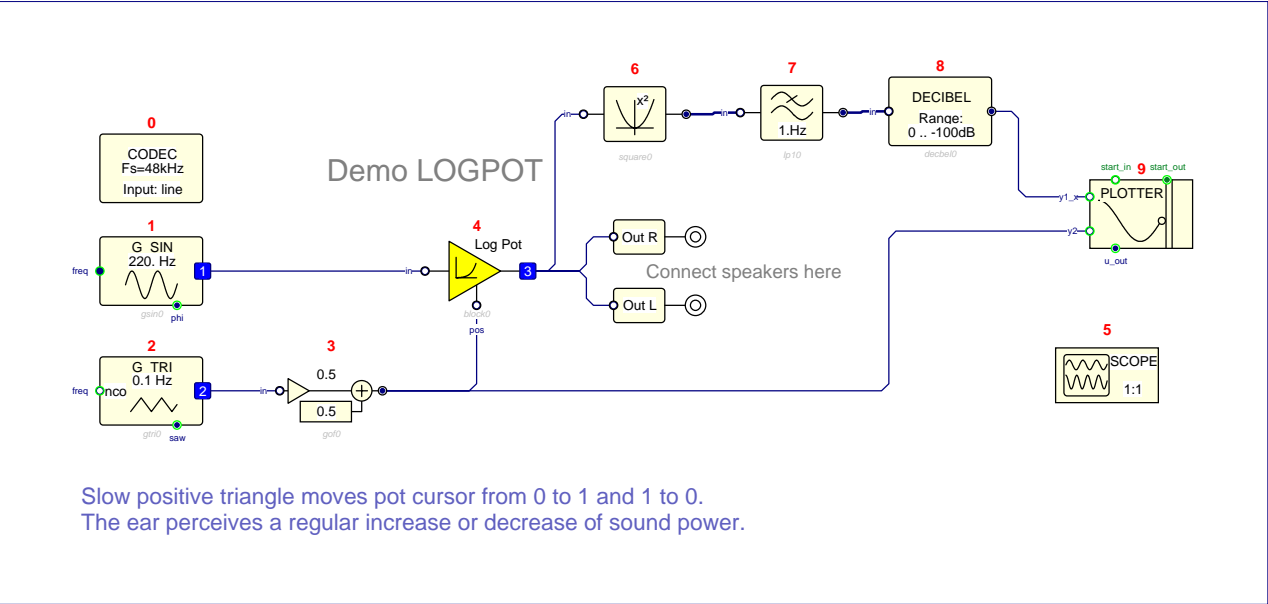
LOGPOT



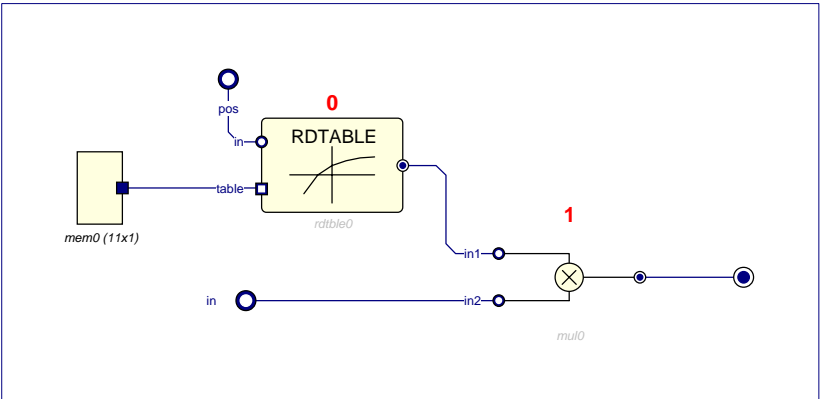
CATEGORY: AUDIO

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



LOGPOT test program

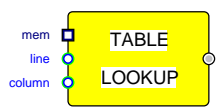


LOGPOT internal schema

LOOKUP

Read data in memory

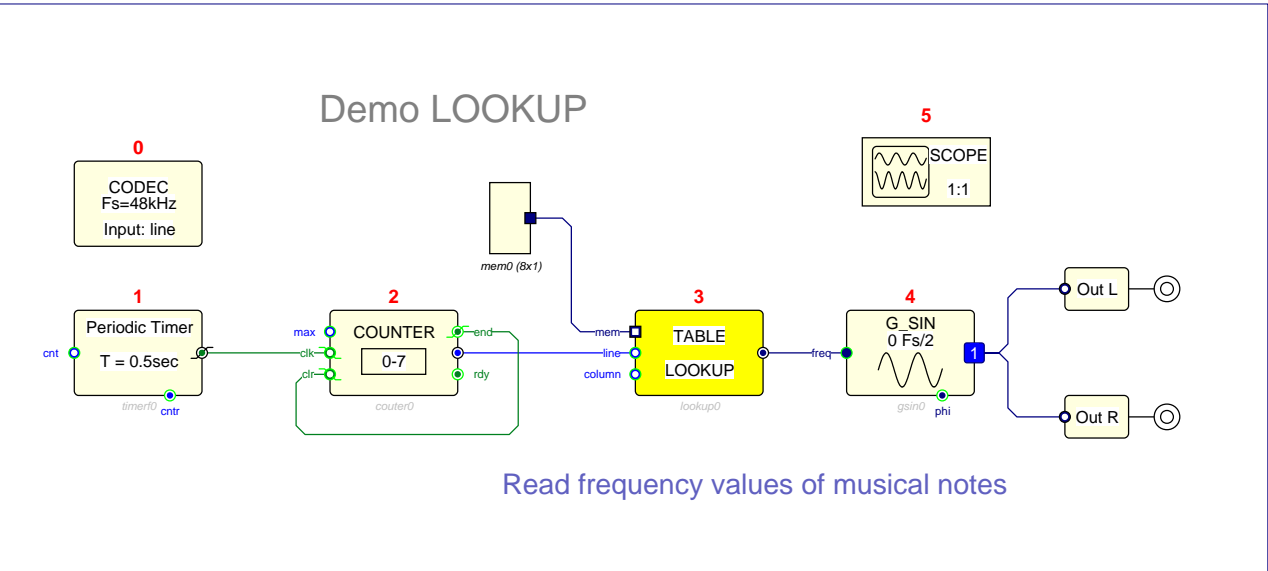
LOOKUP



CATEGORY: CONTROL

DESCRIPTION:
Read data in memory
Field can be X:, Y:, L:, or P:
Input gives address; Output may be single or double.

INPUTS			
<i>Name:</i> name_mem name_line name_column	<i>Data Type:</i> FRACT INTEGER INTEGER	<i>Data Struct:</i> Matrix of WORD WORD WORD	<i>Connection:</i> mandatory optional optional
<i>OUTPUTS</i> <i>Name:</i> name	<i>Data Type:</i> defined by cn	<i>Data Struct:</i>	<i>Connection:</i> normal

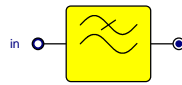


LOOKUP test program

LP1

1st order recursive lowpass filter

LP1



CATEGORY: FILTERS

DESCRIPTION:

1st order recursive lowpass filter

PARAMETERS:

Parameter:

Cutoff frequency
Unit

Default values:

1.
Hz, Fs/2

INPUTS

Name:
name_in

Data Type:
FRACT

Data Struct:
WORD

Connection:
mandatory

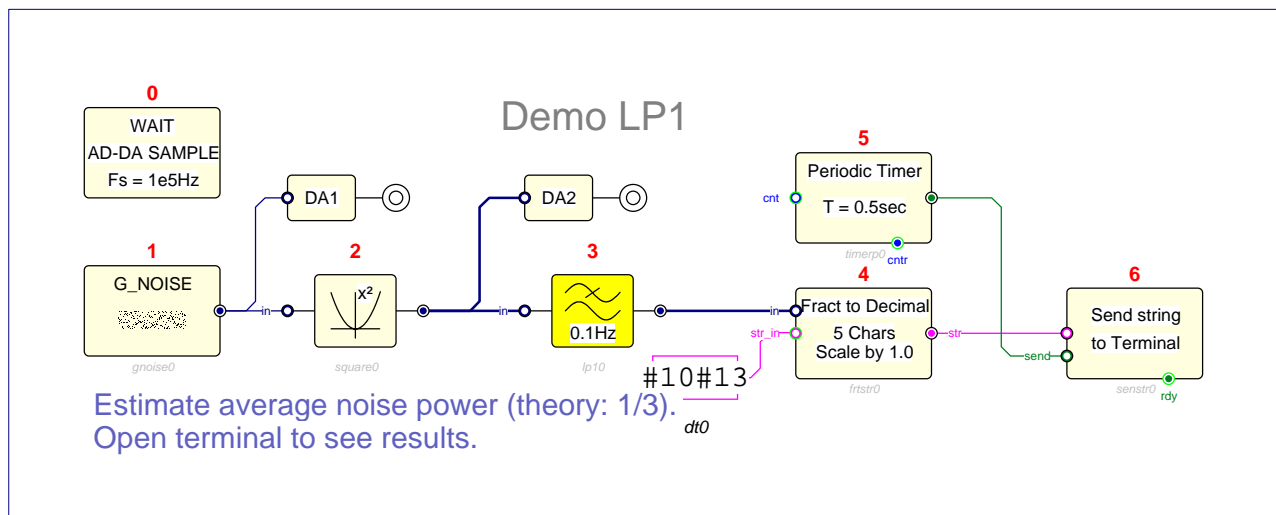
OUTPUTS

Name:
name

Data Type:
FRACT

Data Struct:
DWORD

Connection:
normal

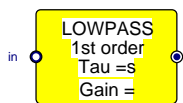


LP1 test program

LP1A

1st order lowpass

LP1A



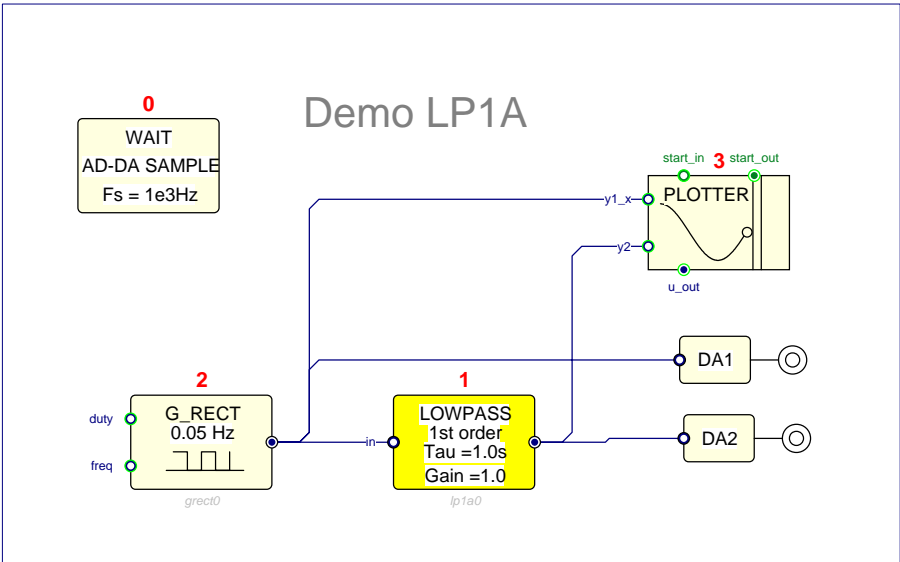
CATEGORY: CONTINUOUS

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

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

INPUTS			
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name_in	FRACT	WORD	mandatory

OUTPUTS			
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name	FRACT	WORD	normal

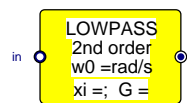


LP1A test program

LP2A

2nd order lowpass

LP2A



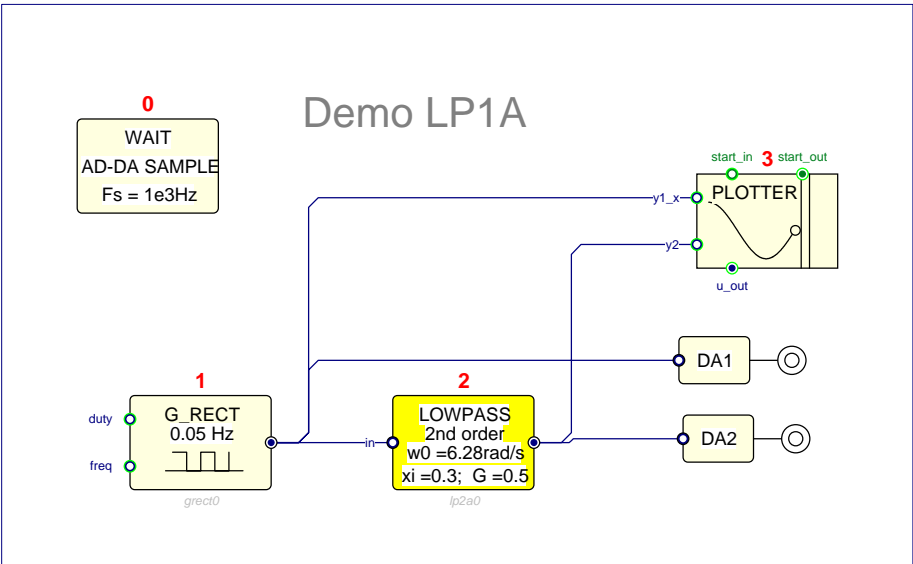
CATEGORY: CONTINUOUS

DESCRIPTION:
2nd order lowpass
 $Y(p)/X(p) = \frac{w_0^2}{(p^2 + 2\xi w_0 p + w_0^2)}$

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

INPUTS
Name: *Data Type:* *Data Struct:* *Connection:*
name_in FRAC T WORD mandatory

OUTPUTS
Name: *Data Type:* *Data Struct:* *Connection:*
name FRAC T WORD normal

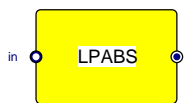


LP2A test program

LPABS

Lowpass of abs value

LPABS



CATEGORY: FILTERS

DESCRIPTION:
Lowpass of abs value

PARAMETERS:
Parameter:
Cutoff frequency
Unit

Default values:
1.
Hz,Fs/2

INPUTS
Name:
name_in

Data Type:
FRACT

Data Struct:
WORD

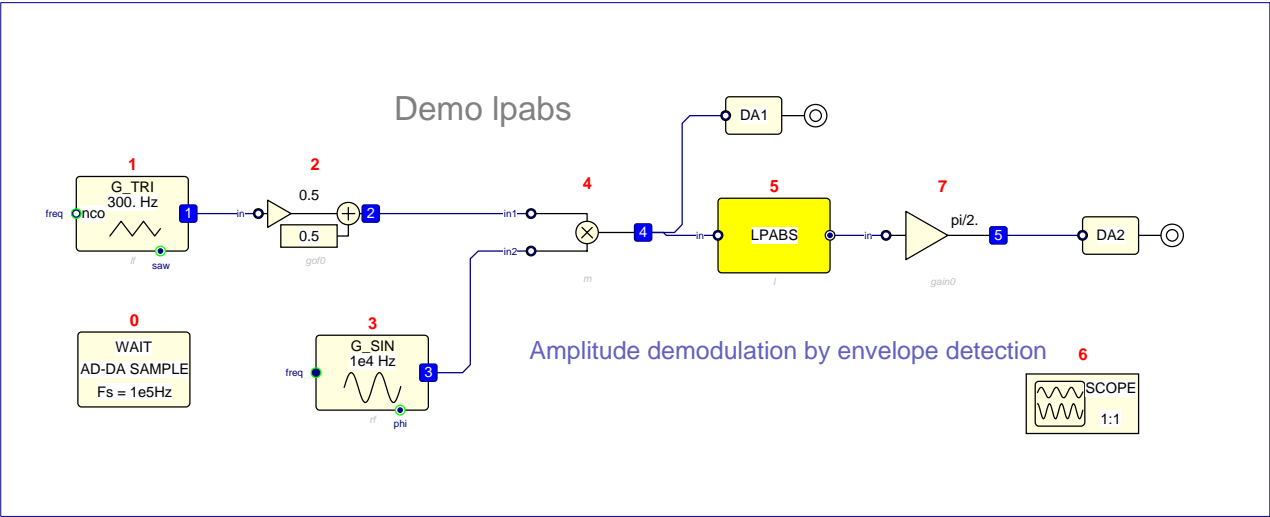
Connection:
mandatory

OUTPUTS
Name:
name

Data Type:
FRACT

Data Struct:
WORD

Connection:
normal

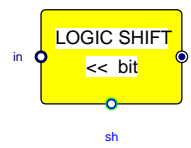


LPABS test program

LSHIFT

Logic N bit Shift

LSHIFT



DESCRIPTION:
Logic N bit Shift
if N>0 then Logic Shift Left
else Logic Shift Right

PARAMETERS:
Parameter:
N
Default values:
1

INPUTS			
<i>Name:</i> name_in name_sh	<i>Data Type:</i> FRACT INTEGER	<i>Data Struct:</i> WORD WORD	<i>Connection:</i> mandatory optional
OUTPUTS			
<i>Name:</i> name	<i>Data Type:</i> FRACT	<i>Data Struct:</i> WORD	<i>Connection:</i> normal



DESCRIPTION:
MIDI File
Transcripted in asm format

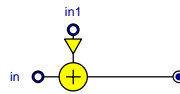
OUTPUTS			
<i>Name:</i> name	<i>Data Type:</i> INTEGER	<i>Data Struct:</i> Matrix of WORD	<i>Connection:</i> normal

ATTRIBUTES
Non executable, Unique, Data Table

MADD

Multiply and Add

MADD



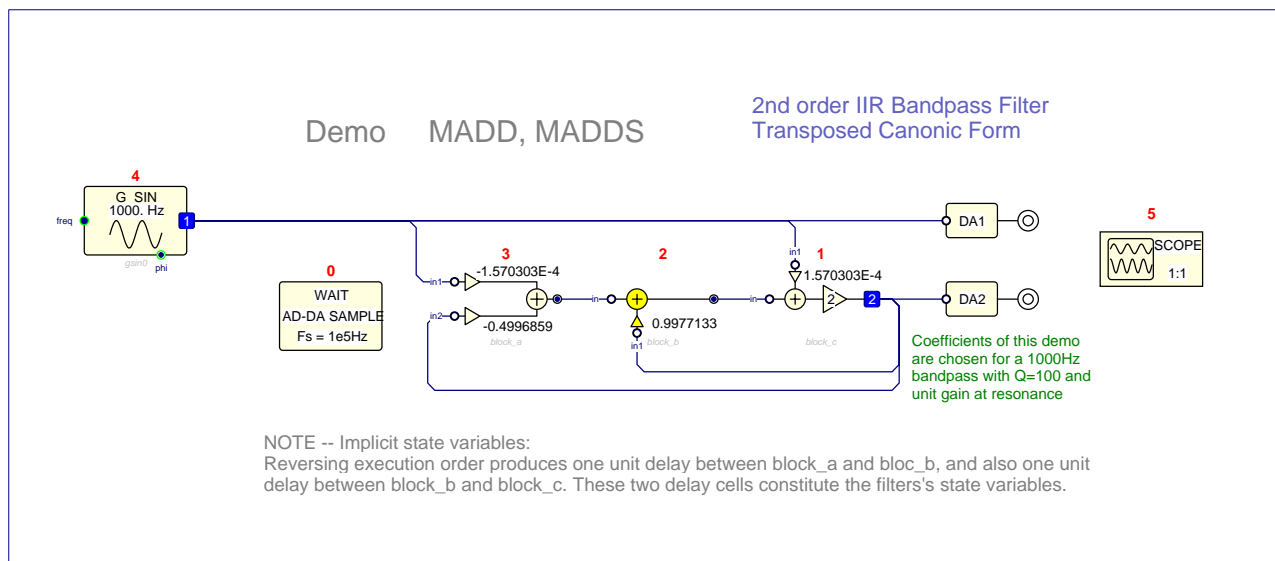
CATEGORY: ARITHMETIC

DESCRIPTION:
Multiply and Add
Add weighted input
 $y = in + G_i * in1$
This is a filter primitive

PARAMETERS:
Parameter: G_i *Default values:* 1.0

INPUTS			
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name_in1	FRACT	WORD	mandatory
name_in	FRACT	WORD	mandatory

OUTPUTS			
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name	FRACT	WORD	normal

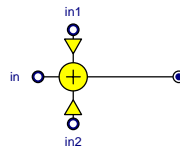


MADD test program

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:

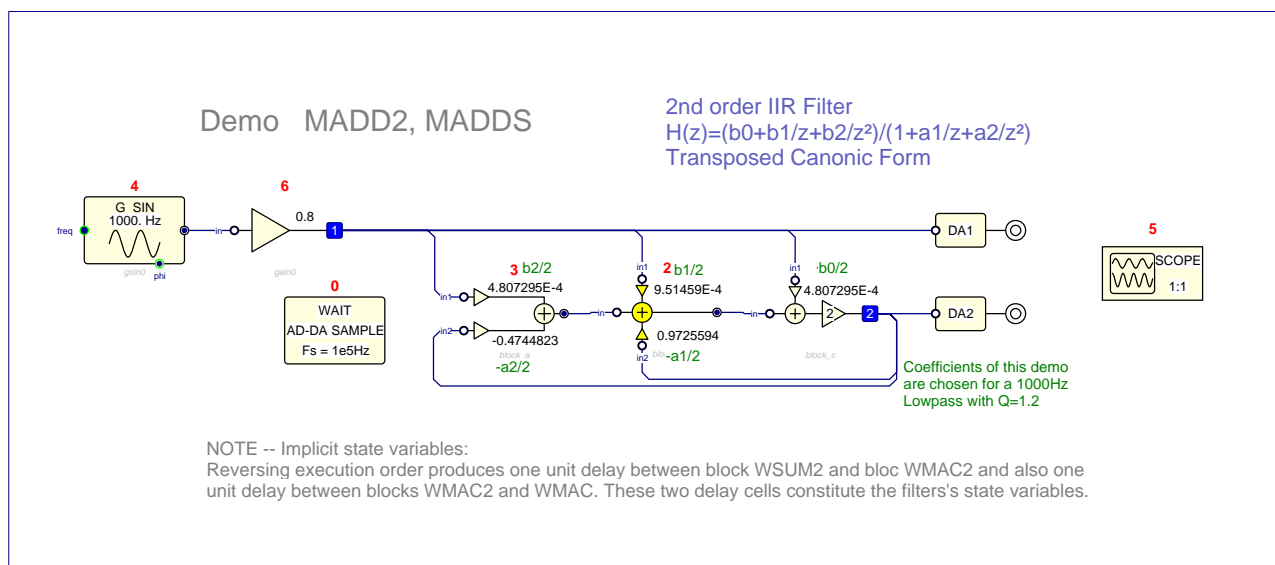
Parameter:	Default values:
G1	1.0
G2	1.0

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

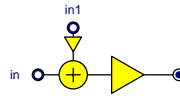


MADD2 test program

MADDS

Multiply, Add, Shift

MADDS



CATEGORY: ARITHMETIC

DESCRIPTION:

Multiply, Add, Shift
Add weighted input, optionally shift result
 $y = (in + G_i * in1) * G_s$
This is an IIR filter primitive

PARAMETERS:

<i>Parameter:</i>	<i>Default values:</i>
G_i	1.0
G_s	2,1,0.5

INPUTS

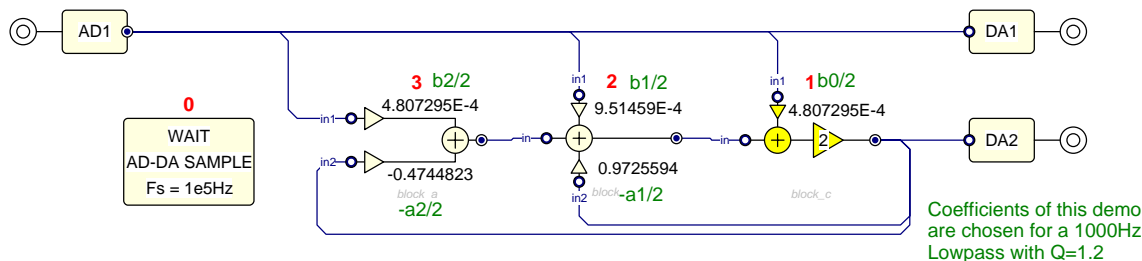
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name_in1	FRACT	WORD	mandatory
name_in	FRACT	WORD	mandatory

OUTPUTS

<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name	FRACT	WORD	normal

Demo MADD2, MADDS

2nd order IIR Filter
 $H(z) = (b_0 + b_1/z + b_2/z^2) / (1 + a_1/z + a_2/z^2)$
Transposed Canonic Form



NOTE -- Implicit state variables:

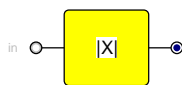
Reversing execution order produces one unit delay between block WSUM2 and bloc WMAC2 and also one unit delay between blocks WMAC2 and WMAC. These two delay cells constitute the filters's state variables.

MADDS test program

MAGN

Magnitude of a real or complex

MAGN



CATEGORY: NON LINEAR

DESCRIPTION:
Magnitude of a real or complex

INPUTS

Name:
name_in

Data Type:
defined by cn

Data Struct:

Connection:
mandatory

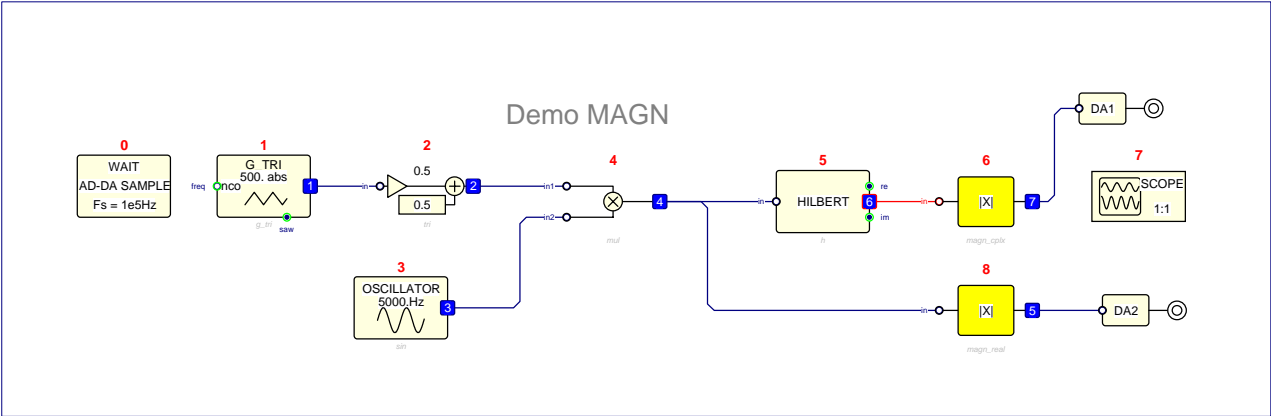
OUTPUTS

Name:
name

Data Type:
FRACT

Data Struct:
WORD

Connection:
normal

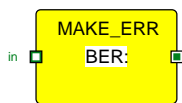


MAGN test program

MAKE_ERR

Inject errors within vector

MAKE_ERR



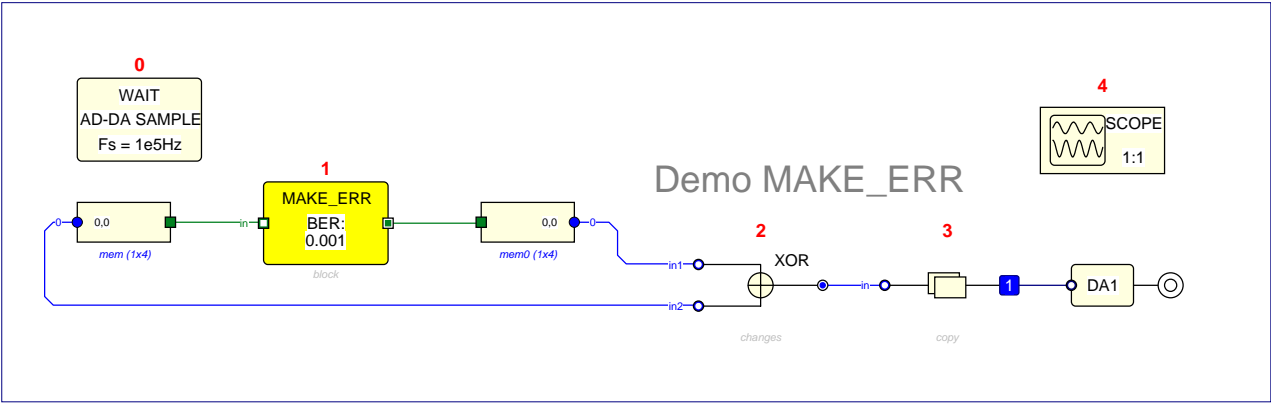
CATEGORY: MATRIX

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

PARAMETERS:
Parameter: BER: Default values: 0.0001

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

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

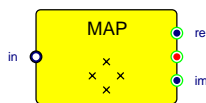


MAKE_ERR test program

MAP

Map symbol to complex

MAP



CATEGORY: TELECOM

DESCRIPTION:

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

PARAMETERS:

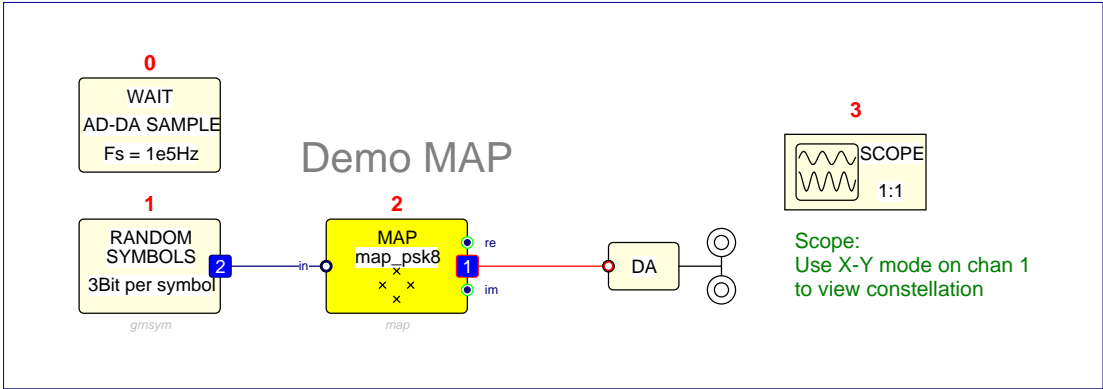
<i>Parameter:</i>	<i>Default values:</i>
Bits per symbol	1
Constellation	map_ook,map_bpsk,map_ask4,map_ask8,map_psk4,map_psk8,map_qam8,map_qam16

INPUTS

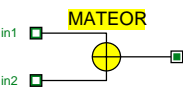
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name_in	FRACT	WORD	mandatory

OUTPUTS

<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name	COMPLEX	WORD	optional
name_re	FRACT	WORD	optional
name_im	FRACT	WORD	optional



MAP test program



CATEGORY: MATRIX

DESCRIPTION:
XOR between matrixes

INPUTS

Name:
name_in1
name_in2

Data Type:
BOOL
BOOL

Data Struct:
Matrix of BIT
Matrix of BIT

Connection:
mandatory
mandatory

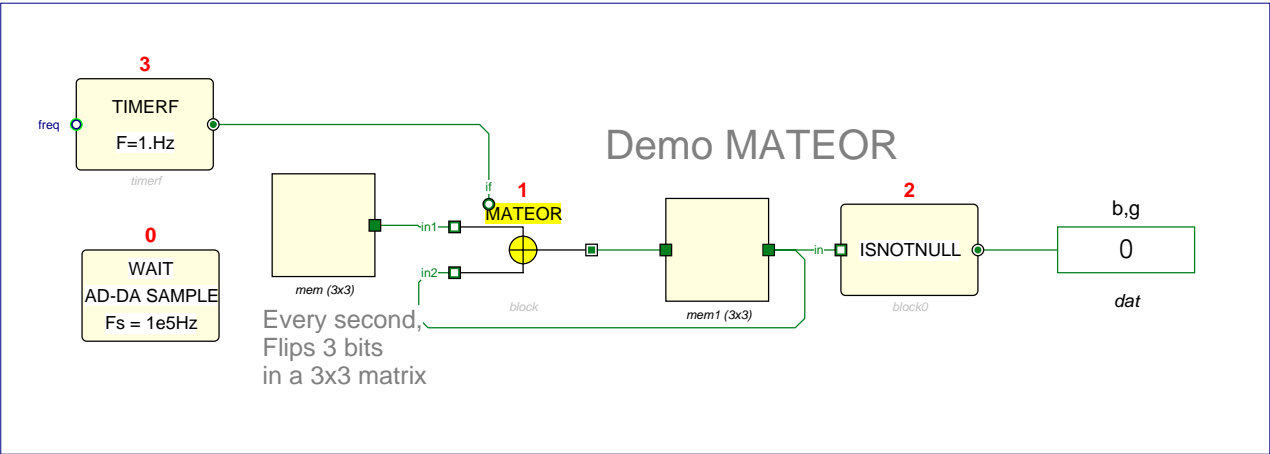
OUTPUTS

Name:
name

Data Type:
BOOL

Data Struct:
Matrix of BIT

Connection:
normal

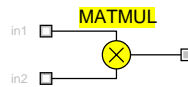


MATEOR test program

MATMUL

Matrix product

MATMUL



CATEGORY: MATRIX

DESCRIPTION:
Matrix product

INPUTS

Name:
name_in1
name_in2

Data Type:
defined by cn
defined by cn

Data Struct:
Matrix of
Matrix of

Connection:
mandatory
mandatory

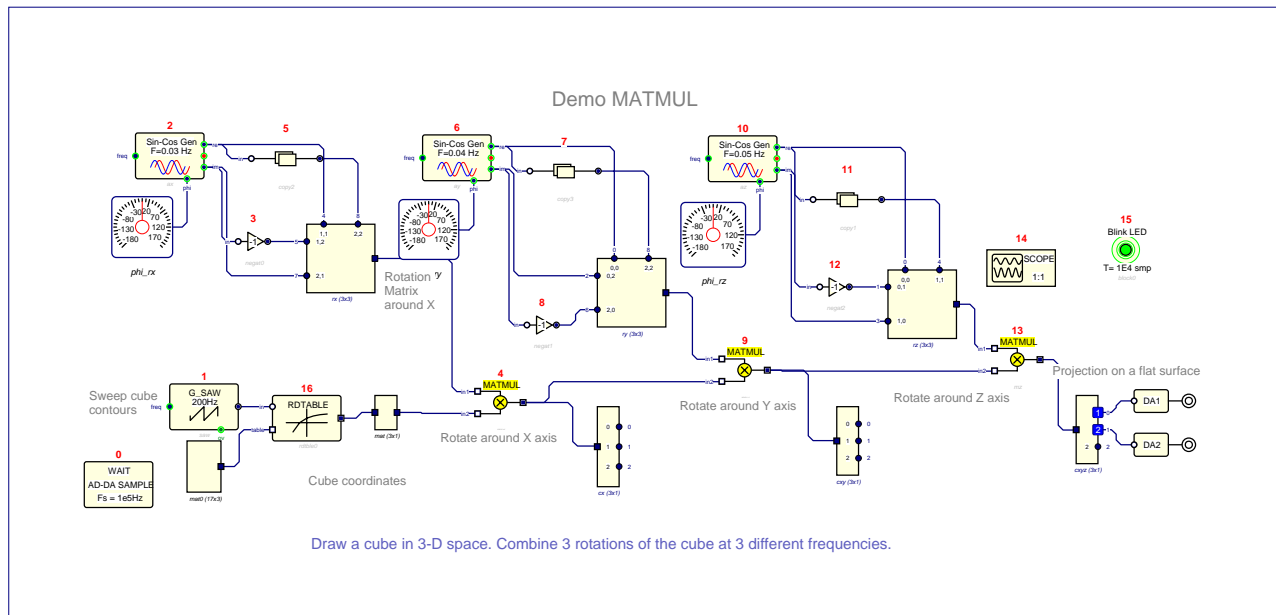
OUTPUTS

Name:
name

Data Type:
defined by cn

Data Struct:
Matrix of

Connection:
normal

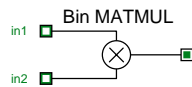


MATMUL test program

MATMULB

Boolean Matrix product

MATMULB



CATEGORY: MATRIX

DESCRIPTION:

Boolean Matrix product

In Galois Field(2)

$M[i,k] = \text{Sigma}\{M1[i,j]M2[j,k], j=0..\text{cols}(M1)\} \text{ mod } 2$

INPUTS

Name:

name_in1

name_in2

Data Type:

BOOL

BOOL

Data Struct:

Matrix of BIT

Matrix of BIT

Connection:

mandatory

mandatory

OUTPUTS

Name:

name

Data Type:

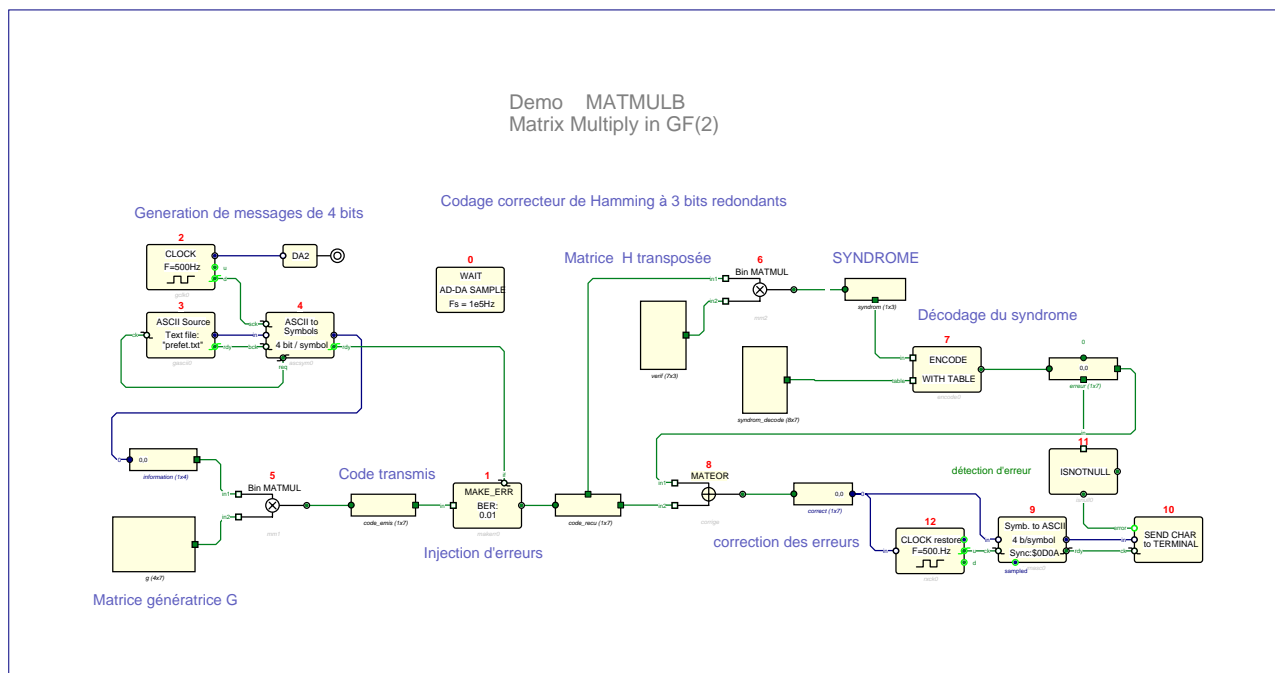
BOOL

Data Struct:

Matrix of BIT

Connection:

normal

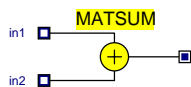


MATMULB test program

MATSUM

Sum of matrixes

MATSUM



CATEGORY: MATRIX

DESCRIPTION:
Sum of matrixes

INPUTS

Name:
name_in1
name_in2

Data Type:
FRACT
FRACT

Data Struct:
Matrix of WORD
Matrix of WORD

Connection:
mandatory
mandatory

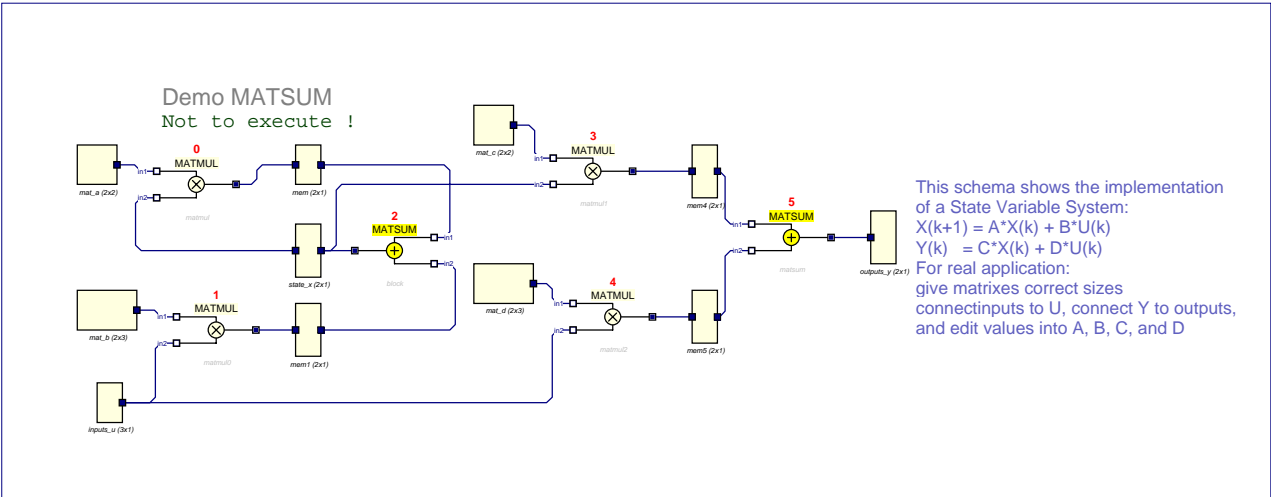
OUTPUTS

Name:
name

Data Type:
FRACT

Data Struct:
Matrix of WORD

Connection:
normal

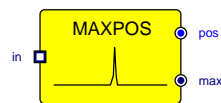


MATSUM test program

MAXPOS

Position of maximum

MAXPOS



CATEGORY: CONTROL

DESCRIPTION:

Position of maximum
among elements of vector between min pos and max pos
Pos is an integer relative to vector start
Max gives the peak value

PARAMETERS:

Parameter:

Min pos
Max pos

Default values:

0
100

INPUTS

Name:
name_in

Data Type:
FRACT

Data Struct:
Matrix of WORD

Connection:
mandatory

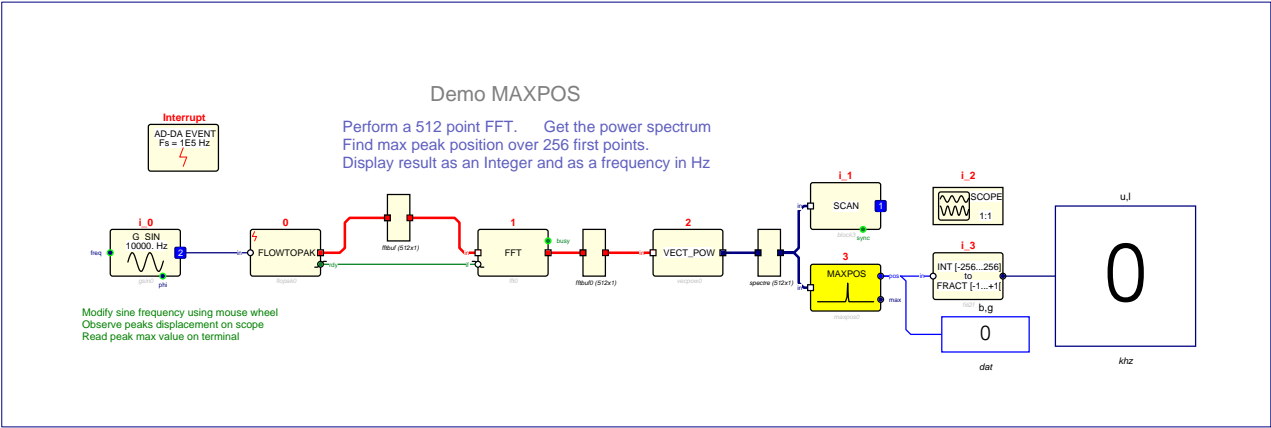
OUTPUTS

Name:
name_pos
name_max

Data Type:
INTEGER
FRACT

Data Struct:
WORD
WORD

Connection:
normal
normal



MAXPOS test program

MIDI_BACH

MIDI File

MIDI_BACH



CATEGORY: AUDIO

DESCRIPTION:
MIDI File
Transcripted in asm format

OUTPUTS

<i>Name:</i> name	<i>Data Type:</i> INTEGER	<i>Data Struct:</i> Matrix of WORD	<i>Connection:</i> normal
----------------------	------------------------------	---------------------------------------	------------------------------

ATTRIBUTES
Non executable, Unique, Data Table



CATEGORY: AUDIO

DESCRIPTION:
MIDI File
Transcripted in asm format

OUTPUTS

<i>Name:</i> name	<i>Data Type:</i> INTEGER	<i>Data Struct:</i> Matrix of WORD	<i>Connection:</i> normal
----------------------	------------------------------	---------------------------------------	------------------------------

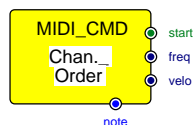
ATTRIBUTES

Non executable, Unique, Data Table

MIDI_CMD

Midi Command interface

MIDI_CMD



CATEGORY: MUSIC

DESCRIPTION:

Midi Command interface
Generates tone data for a given channel
and a given note order

PARAMETERS:

Parameter:

Channel
Note

Default values:

0
0

OUTPUTS

Name:

name_start
name_freq
name_velo
name_note

Data Type:

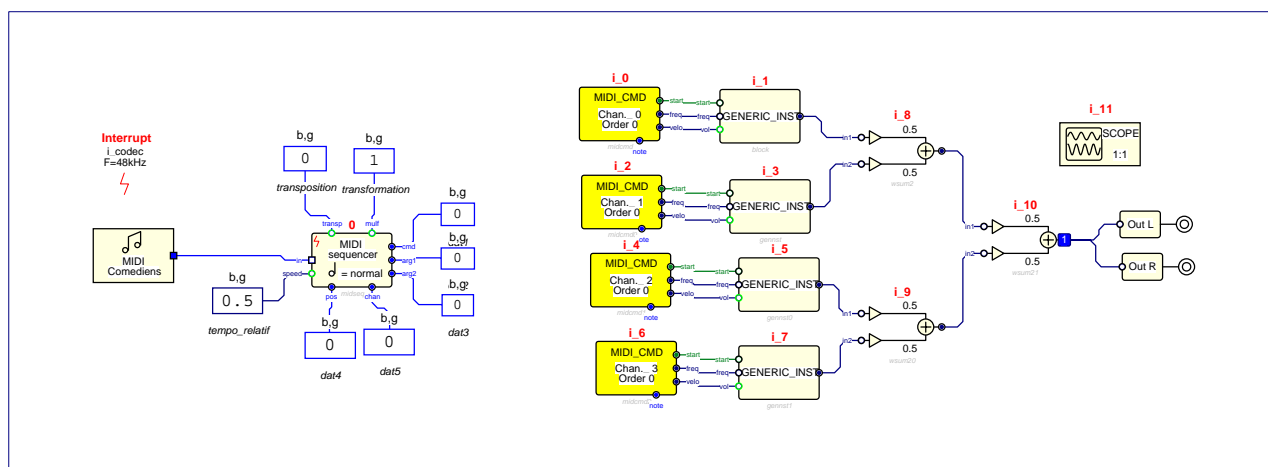
BOOL
FRACT
FRACT
INTEGER

Data Struct:

BIT
WORD
WORD
WORD

Connection:

normal
normal
normal
normal



MIDI_CMD test program

MIDI_ENTTAINR

MIDI File

MIDI_ENTTAINR



CATEGORY: AUDIO

DESCRIPTION:

MIDI File

Transcripted in asm format

OUTPUTS

Name:

name

Data Type:

INTEGER

Data Struct:

Matrix of WORD

Connection:

normal

ATTRIBUTES

Non executable, Unique, Data Table

MIDI_FELICITY

MIDI File

MIDI_FELICITY



CATEGORY: AUDIO

DESCRIPTION:
MIDI File
Transcripted in asm format

OUTPUTS

<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name	INTEGER	Matrix of WORD	normal

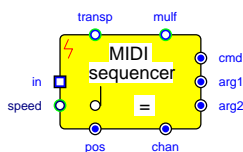
ATTRIBUTES

Non executable, Unique, Data Table

MIDI_SEQ

MIDI Sequencer

MIDI_SEQ



CATEGORY: MUSIC

DESCRIPTION:
MIDI Sequencer

PARAMETERS:

Parameter:

Tempo

Default values:

normal,120

INPUTS

Name:

name_in
name_transp
name_mulf
name_speed

Data Type:

INTEGER
INTEGER
INTEGER
FRACT

Data Struct:

Matrix of WORD
WORD
WORD
WORD

Connection:

mandatory
optional
optional
optional

OUTPUTS

Name:

name_chan
name_arg1
name_cmd
name_arg2
name_pos

Data Type:

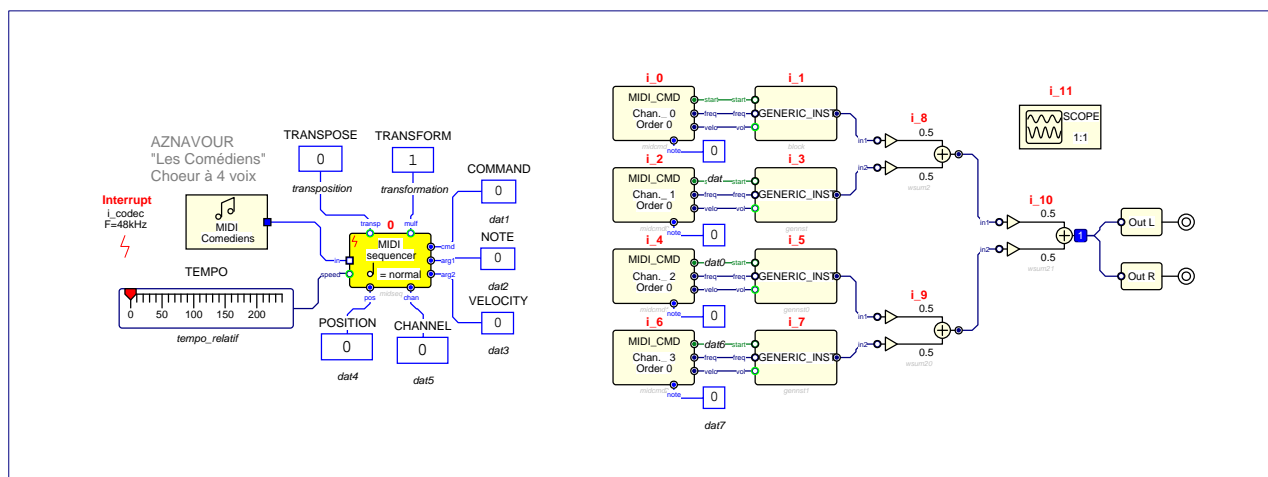
INTEGER
INTEGER
INTEGER
INTEGER
INTEGER

Data Struct:

WORD
WORD
WORD
WORD
WORD

Connection:

normal
normal
normal
normal
normal

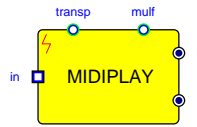


MIDI_SEQ test program

MIDIPLAY

MIDI Sequencer

MIDIPLAY



CATEGORY: AUDIO

DESCRIPTION:

MIDI Sequencer

Plays .MID files in ASM format

Melody can be transposed by n half tones

Melody can be transformed by multiplying note number by m mod 12

INPUTS

Name:

name_in
name_transp
name_mulf

Data Type:

INTEGER
INTEGER
INTEGER

Data Struct:

Matrix of WORD
WORD
WORD

Connection:

mandatory
optional
optional

OUTPUTS

Name:

name_l
name_r

Data Type:

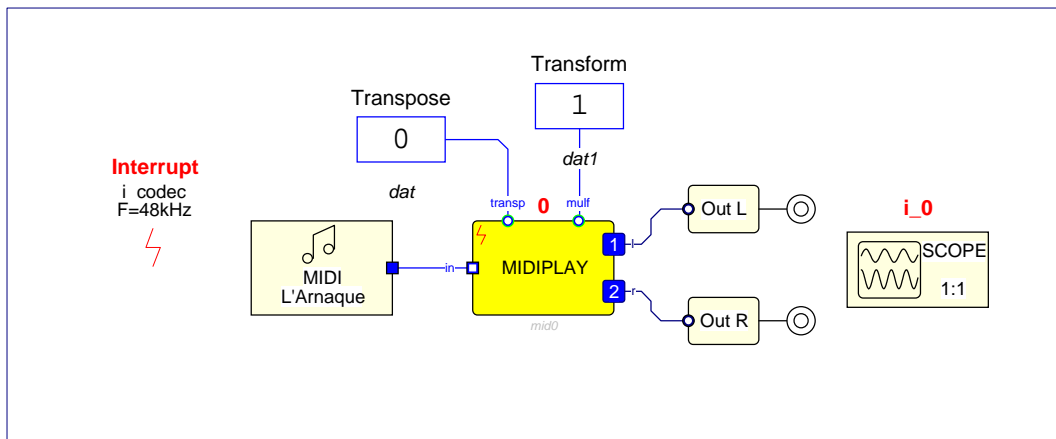
FRACT
FRACT

Data Struct:

WORD
WORD

Connection:

normal
normal

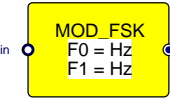


MIDIPLAY test program

MOD_FSK

FSK Modulator

MOD_FSK

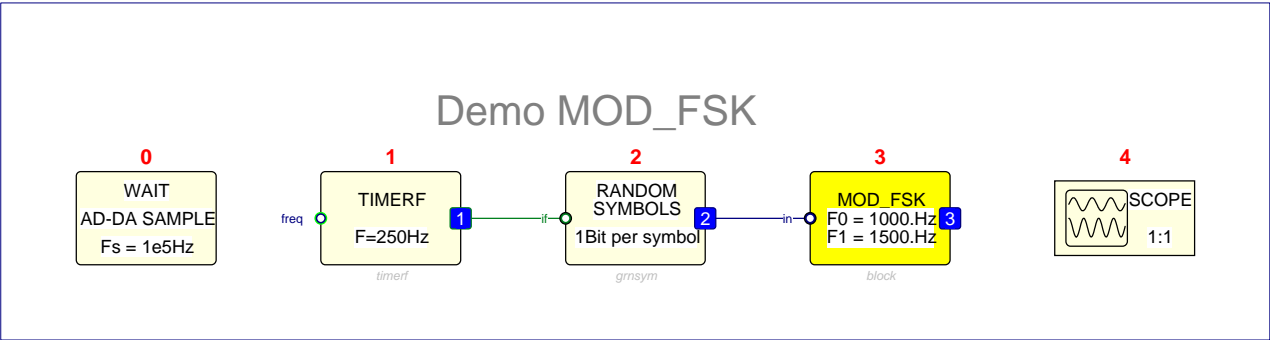


DESCRIPTION:
FSK Modulator

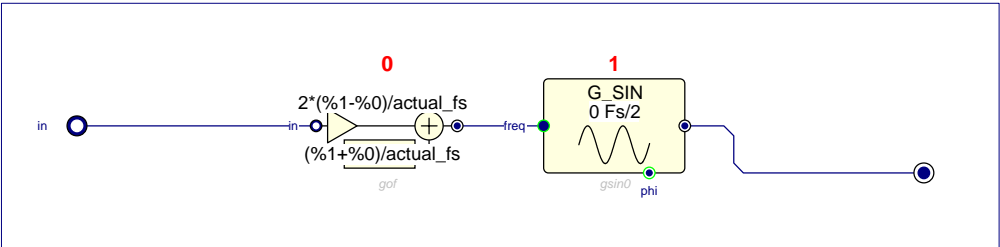
PARAMETERS:
Parameter: Default values:
F0 1000.
F1 1500.

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

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



MOD_FSK test program

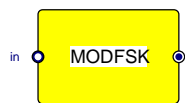


MOD_FSK internal schema

MODFSK

FSK Modulator

MODFSK



CATEGORY: TELECOM

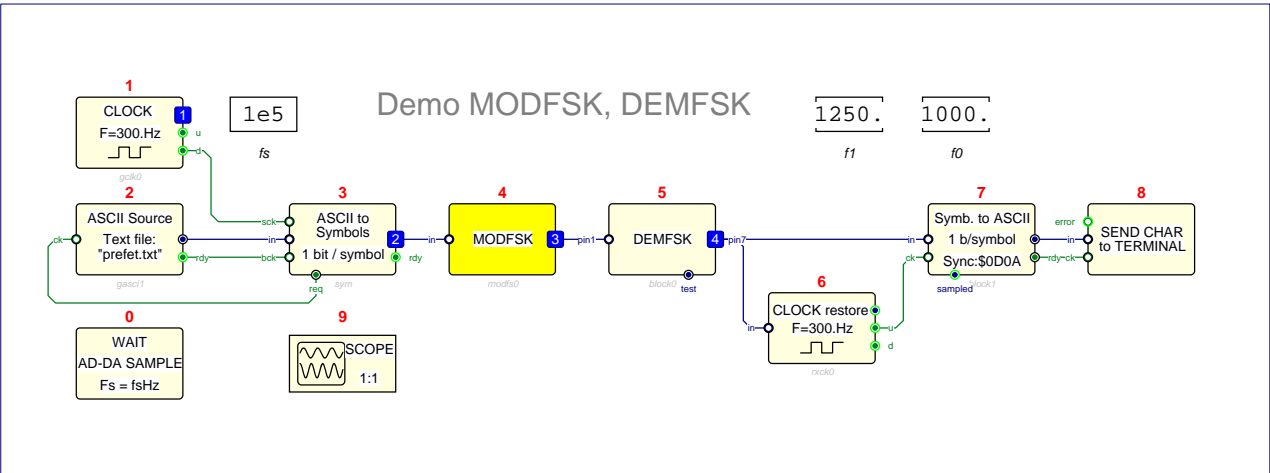
DESCRIPTION:
FSK Modulator

PARAMETERS:

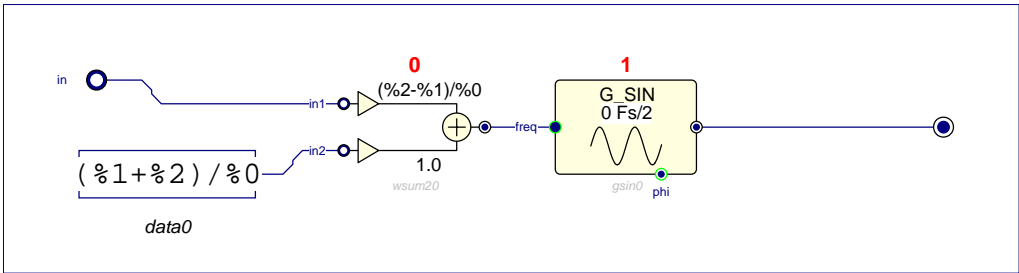
Parameter:	Default values:
fs	1e5
f0	1000.
f1	1500.

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

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



MODFSK test program

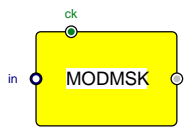


MODFSK internal schema

MODMSK

MSK modulator

MODMSK



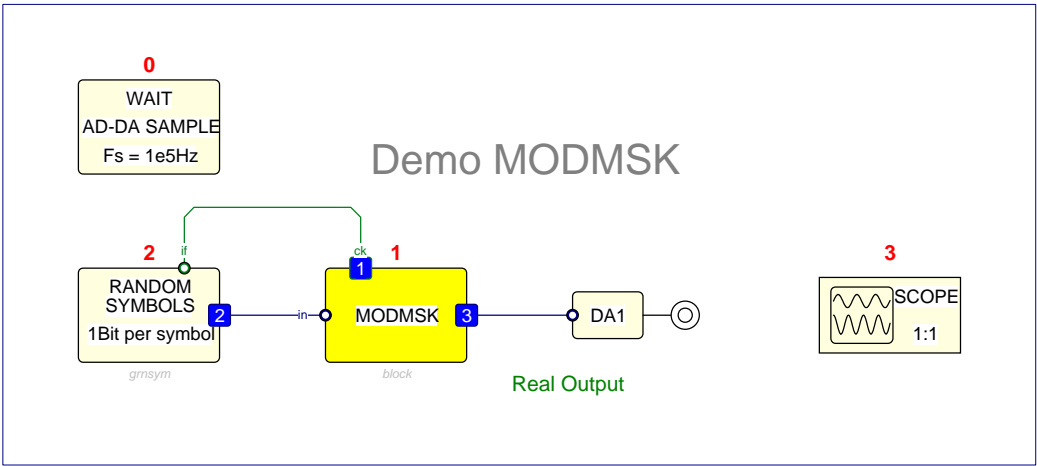
CATEGORY: TELECOM

DESCRIPTION:
MSK modulator
Minimum Shift Keying
Out can be FRACT or COMPLEX

PARAMETERS:
Parameter: Bauds
Default values: 1000.

INPUTS			
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name_in	FRACT	WORD	mandatory

OUTPUTS			
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name	defined by cn		normal
name_ck	BOOL	BIT	normal

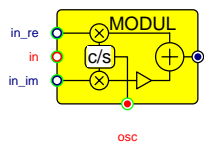


MODMSK test program

MODUL

I-Q modulator

MODUL



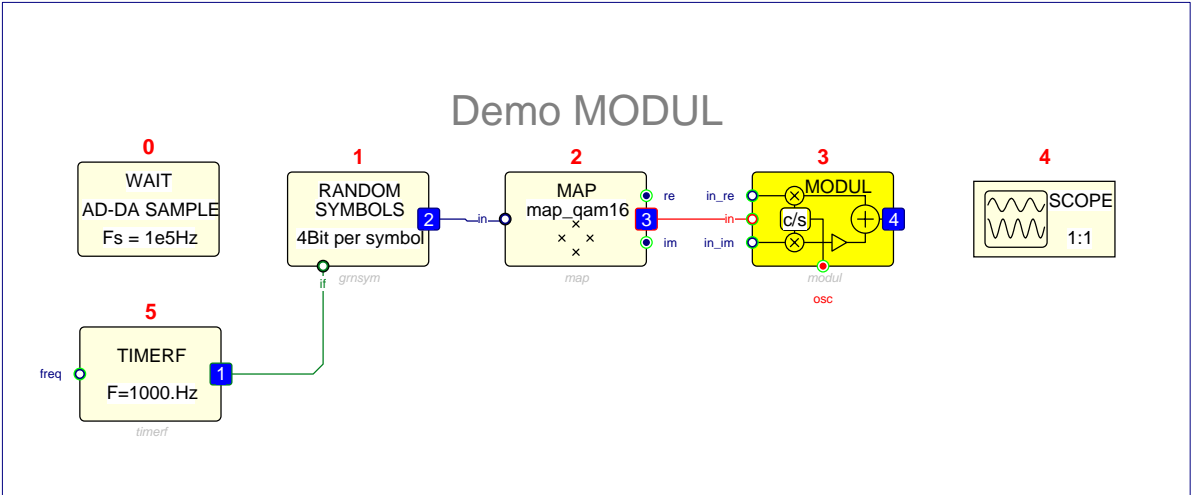
CATEGORY: TELECOM

DESCRIPTION:
I-Q modulator

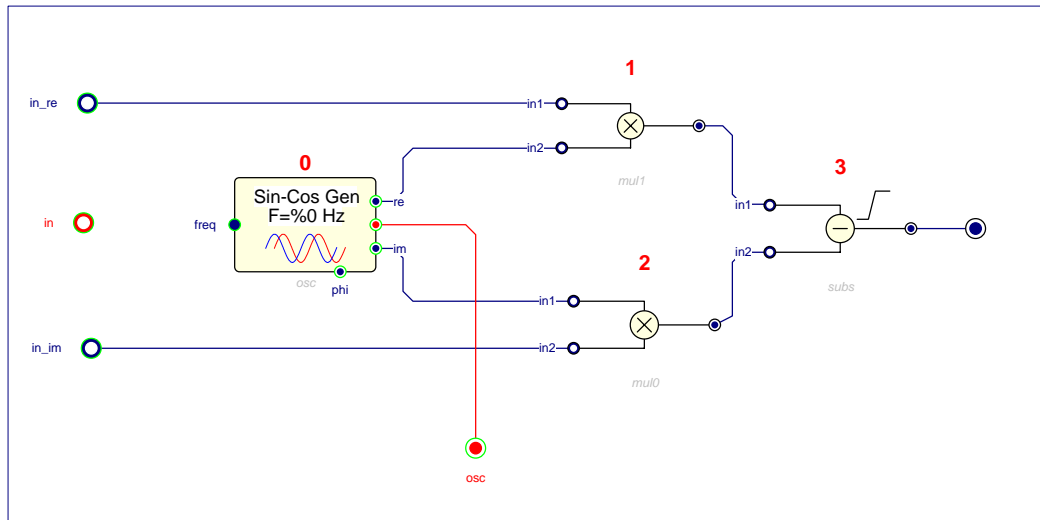
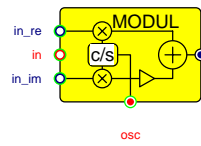
PARAMETERS:
Parameter: Frequency
Default values: 100.

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



MODUL test program

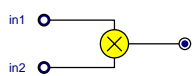


MODUL internal schema

MUL

Real multiplier

MUL



CATEGORY: ARITHMETIC

DESCRIPTION:
Real multiplier

INPUTS

Name:
name_in1
name_in2

Data Type:
FRACT
FRACT

Data Struct:
WORD
WORD

Connection:
mandatory
mandatory

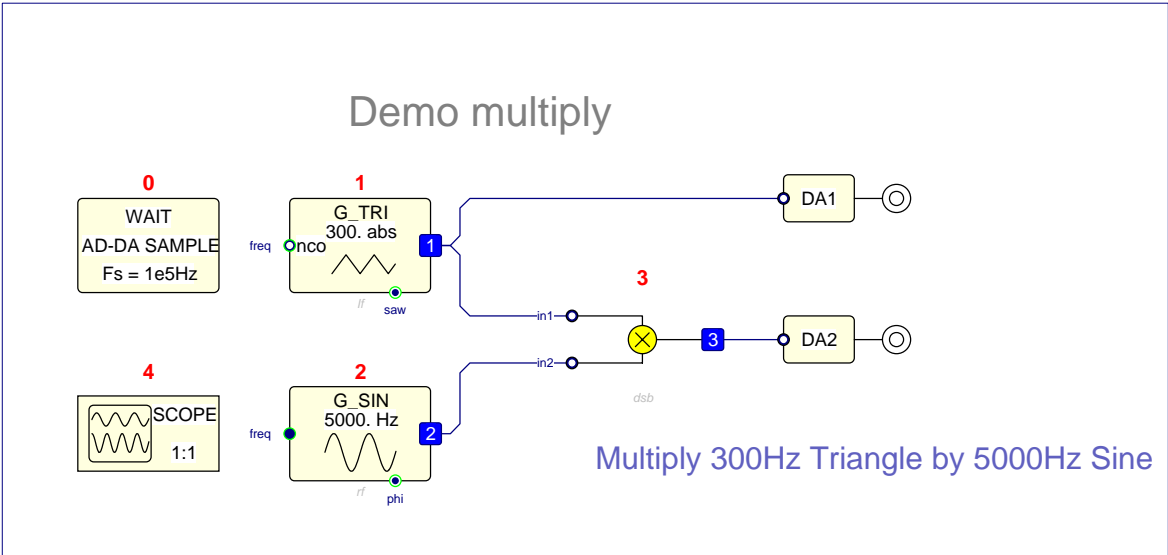
OUTPUTS

Name:
name

Data Type:
FRACT

Data Struct:
WORD

Connection:
normal

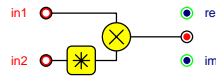


MUL test program

MULCC

Multiply with conjugate

MULCC



CATEGORY: ARITHMETIC

DESCRIPTION:

Multiply with conjugate
Complex product of in1 by conjugate of in2

INPUTS

Name:
name_in1
name_in2

Data Type:
COMPLEX
COMPLEX

Data Struct:
WORD
WORD

Connection:
mandatory
mandatory

OUTPUTS

Name:
name
name_re
name_im

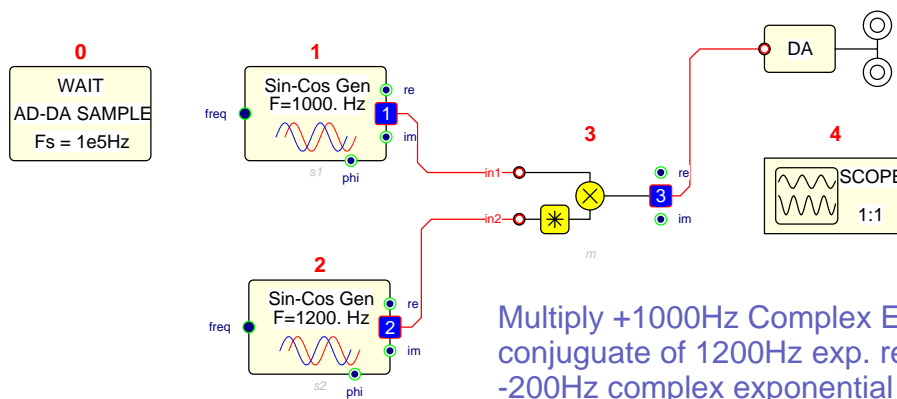
Data Type:
COMPLEX
FRACT
FRACT

Data Struct:
WORD
WORD
WORD

Connection:
normal
optional
optional

Demo mulcc

Multiply by complex conjugate
Phase of result is the difference of inputs phases

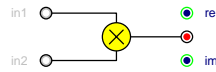


MULCC test program

MULT

Complex, mixed, or real multiplier

MULT



CATEGORY: ARITHMETIC

DESCRIPTION:

Complex, mixed, or real multiplier

INPUTS

Name:

name_in1

name_in2

Data Type:

defined by cn

defined by cn

Data Struct:

Connection:

mandatory

mandatory

OUTPUTS

Name:

name

name_re

name_im

Data Type:

COMPLEX

FRACT

FRACT

Data Struct:

WORD

WORD

WORD

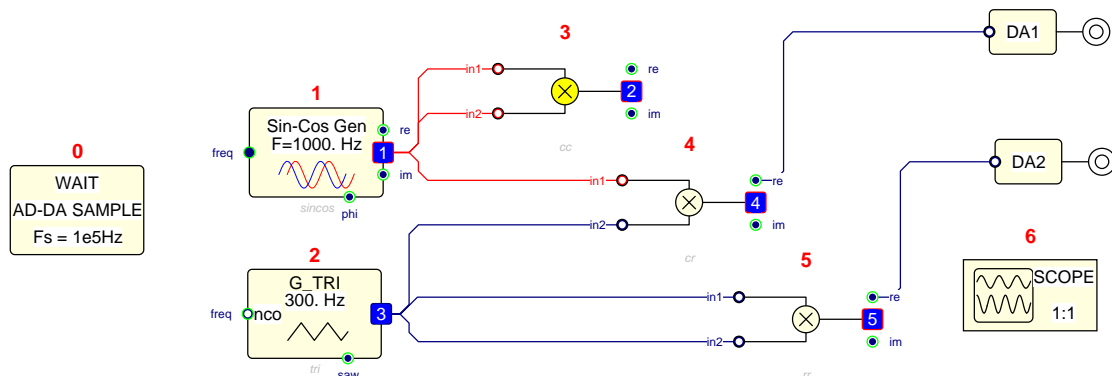
Connection:

normal

optional

optional

Demo mult



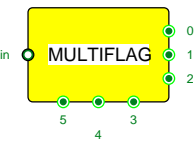
Block MULT accepts real or complex inputs
CxC, CxR, RxC, and RxR products are possible

MULT test program

MULTIFLAG

Set Multiple Flags

MULTIFLAG



CATEGORY: CONTROL

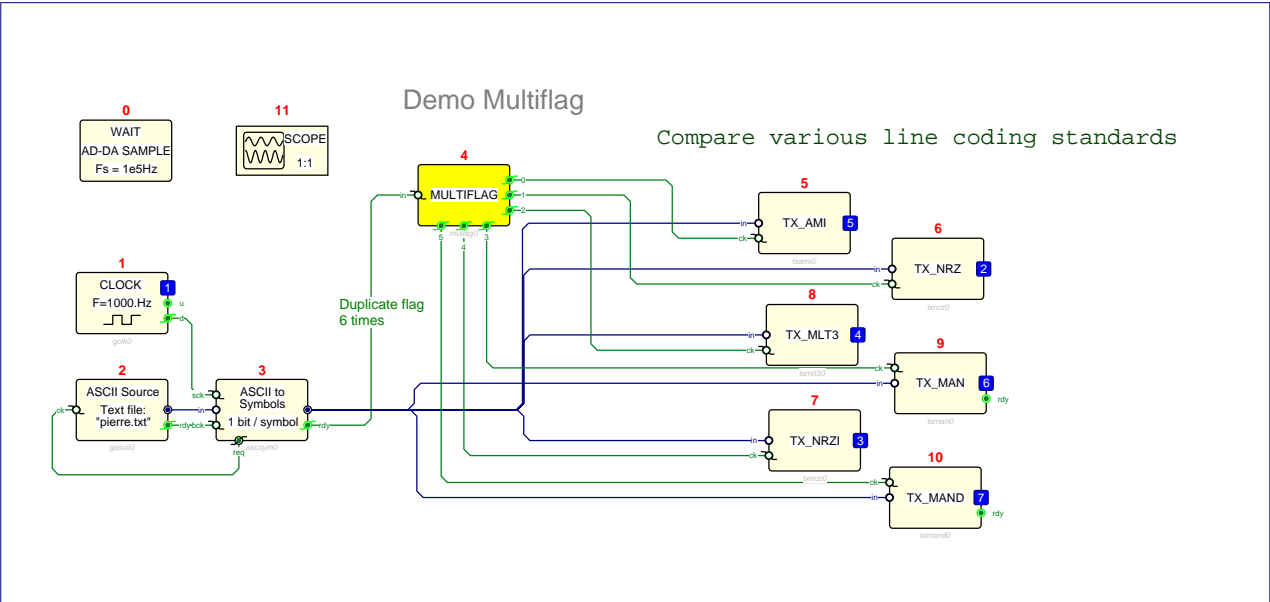
DESCRIPTION:
Set Multiple Flags

INPUTS

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

OUTPUTS

Name:	Data Type:	Data Struct:	Connection:
name_0	BOOL	BIT	optional
name_1	BOOL	BIT	optional
name_2	BOOL	BIT	optional
name_3	BOOL	BIT	optional
name_4	BOOL	BIT	optional
name_5	BOOL	BIT	optional

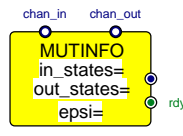


MULTIFLAG test program

MUTINFO

Mutual Information

MUTINFO



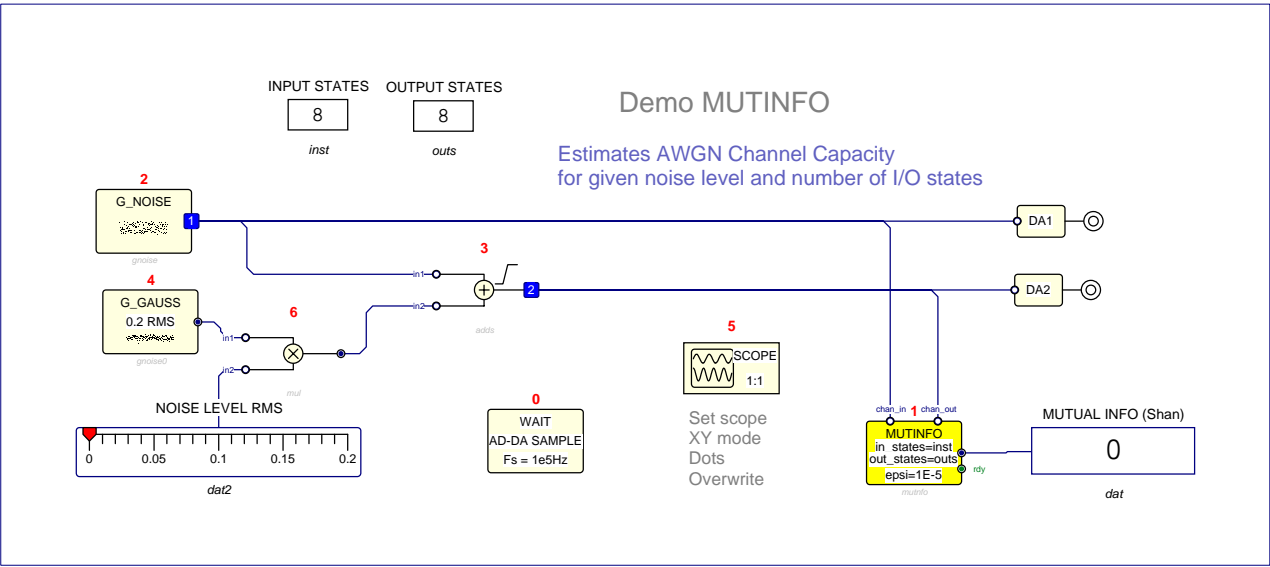
CATEGORY: TELECOM

DESCRIPTION:
Mutual Information
Output calibrated by 1/64

PARAMETERS:
Parameter: *Default values:*
Input states 8
Output states 8
Epsilon 1E-5

INPUTS	Data Type:	Data Struct:	Connection:
<i>Name:</i> name_chan_in	FRACT	WORD	mandatory
name_chan_out	FRACT	WORD	mandatory

OUTPUTS	Data Type:	Data Struct:	Connection:
<i>Name:</i> name	FRACT	WORD	normal
name_rdy	BOOL	BIT	normal

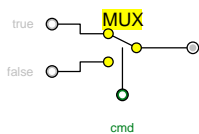


MUTINFO test program

MUX

2 input multiplexer

MUX



CATEGORY: CONTROL

DESCRIPTION:
2 input multiplexer

INPUTS

Name:
name_true
name_false
name_cmd

Data Type:
defined by cn
defined by cn
BOOL

Data Struct:

BIT

Connection:
mandatory
mandatory
mandatory

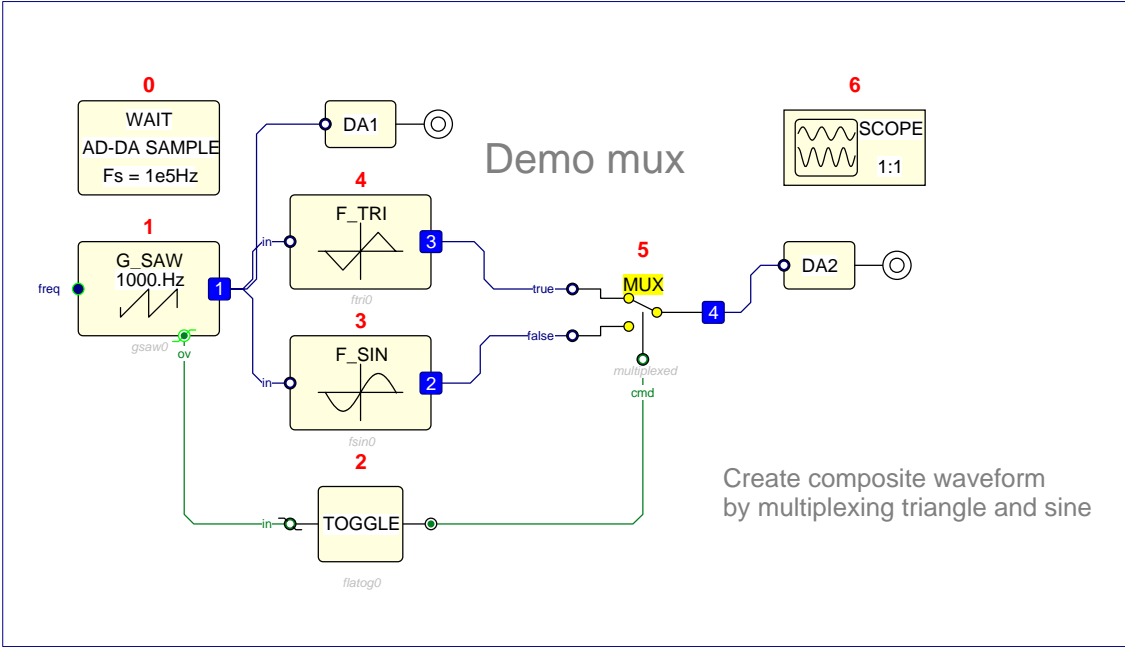
OUTPUTS

Name:
name

Data Type:
defined by cn

Data Struct:

Connection:
normal

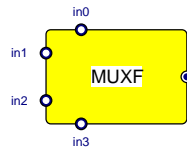


MUX test program

MUXF

f-domain multiplexer

MUXF



CATEGORY: TELECOM

DESCRIPTION:
f-domain multiplexer

INPUTS

Name:

name_in1
name_in0
name_in2
name_in3

Data Type:

FRACT
FRACT
FRACT
FRACT

Data Struct:

WORD
WORD
WORD
WORD

Connection:

mandatory
mandatory
mandatory
mandatory

OUTPUTS

Name:

name

Data Type:

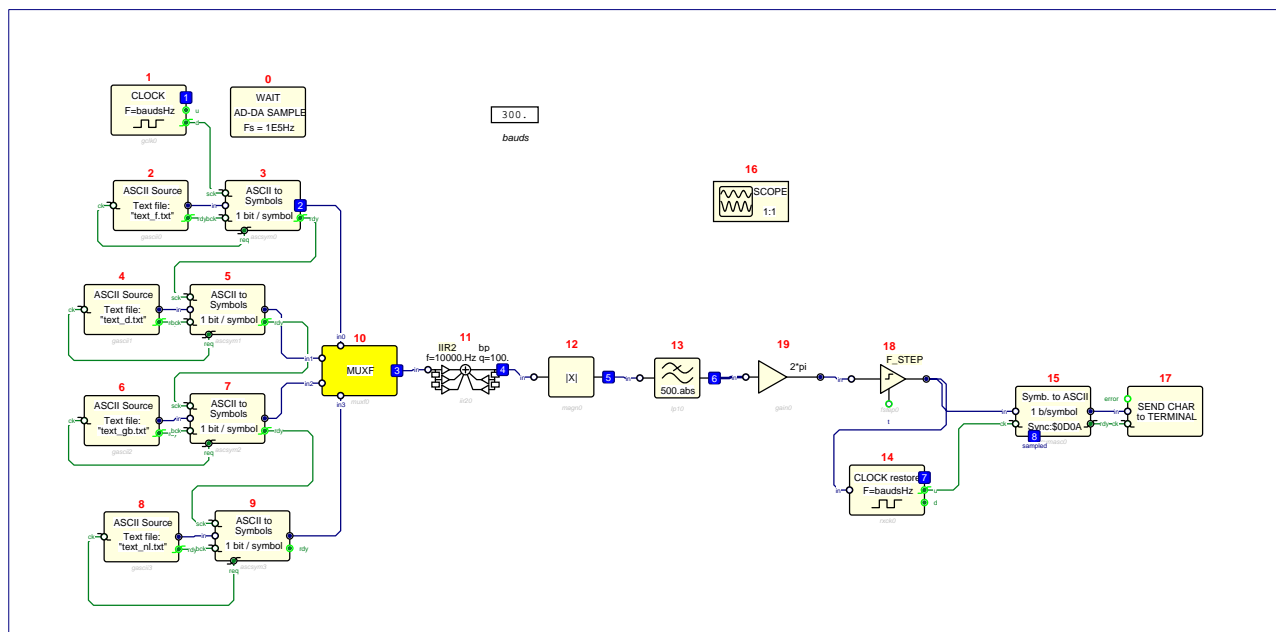
FRACT

Data Struct:

WORD

Connection:

normal

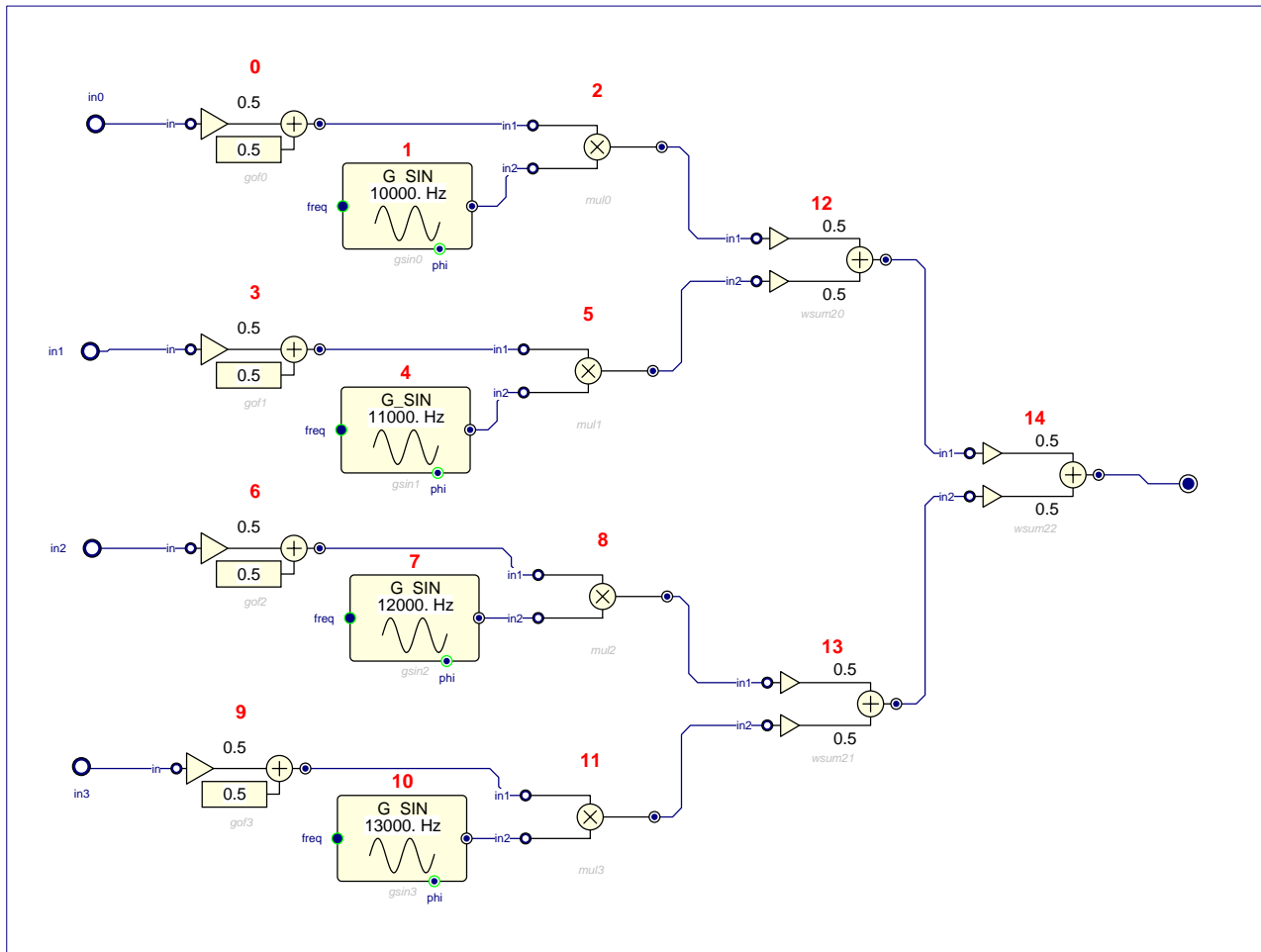
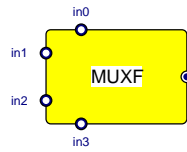


MUXF test program

MUXF

f-domain multiplexer

MUXF

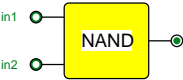


MUXF internal schema

NANDGATE

Logic NAND

NANDGATE



CATEGORY: LOGIC

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

INPUTS

Name:
name_in1
name_in2

Data Type:
BOOL
BOOL

Data Struct:
BIT
BIT

Connection:
mandatory
mandatory

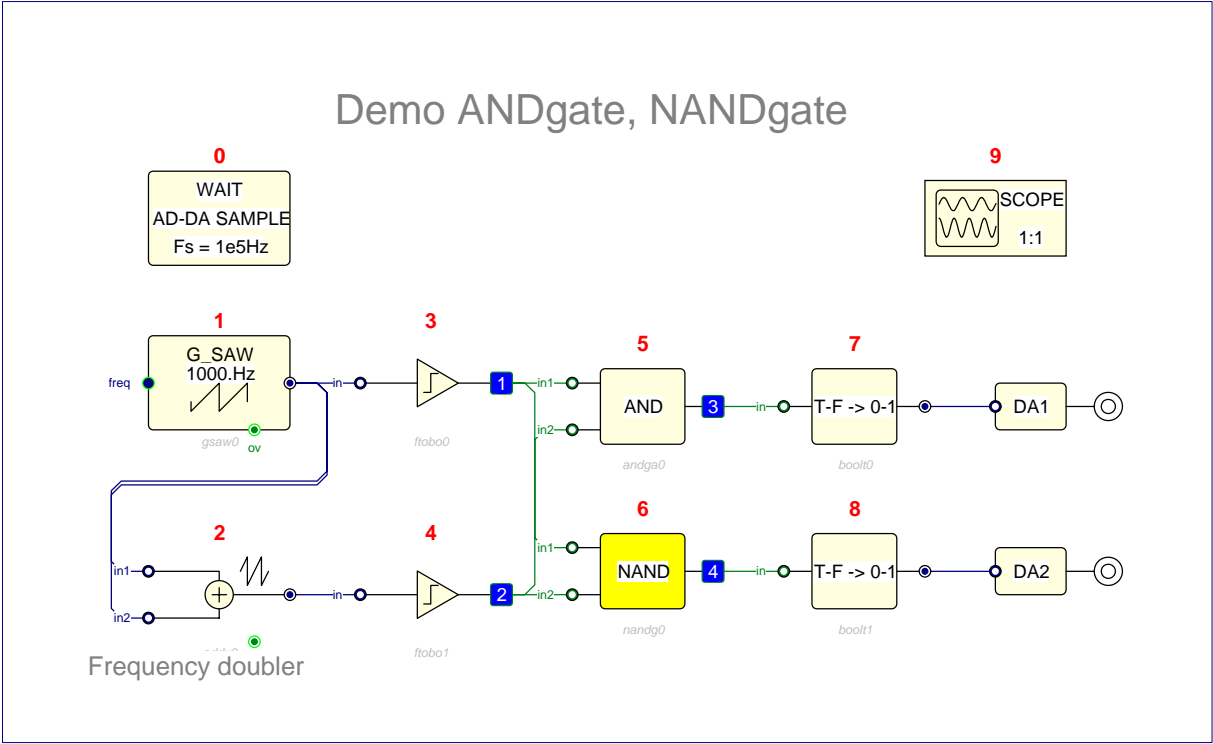
OUTPUTS

Name:
name

Data Type:
BOOL

Data Struct:
BIT

Connection:
normal

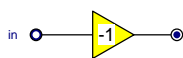


NANDGATE test program

NEGATE

Sign inversion $y = -x$

NEGATE



CATEGORY: ARITHMETIC

DESCRIPTION:
Sign inversion $y = -x$

INPUTS

Name:
name_in

Data Type:
FRACT

Data Struct:
WORD

Connection:
mandatory

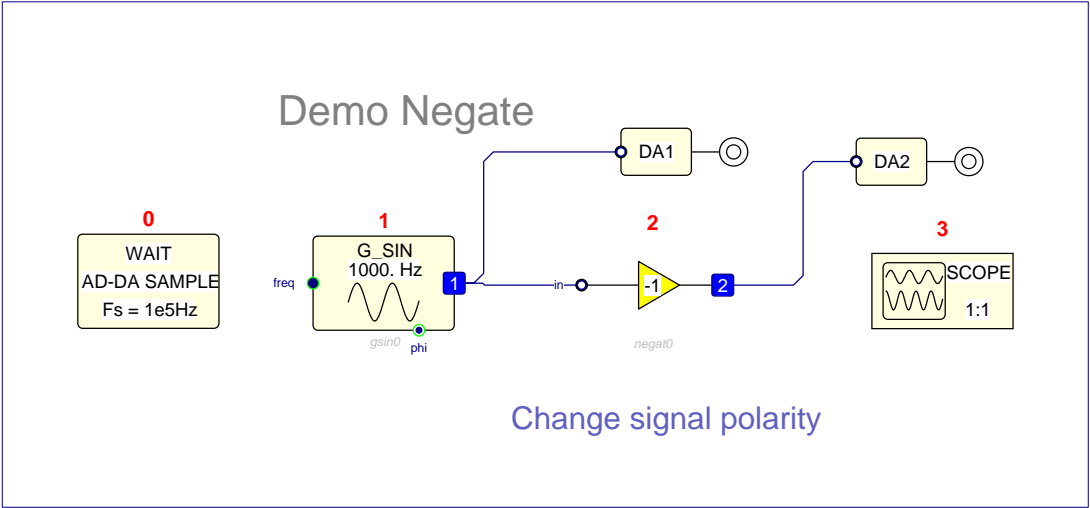
OUTPUTS

Name:
name

Data Type:
FRACT

Data Struct:
WORD

Connection:
normal



NEGATE test program

NOP

No operation

NOP



CATEGORY: CONTROL

DESCRIPTION:
No operation

Demo NOP

0



nop0

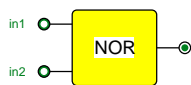
This program does nothing
(infinite loop with NOP instruction inside)

NOP test program

NORGATE

Logic NOR function

NORGATE



CATEGORY: LOGIC

DESCRIPTION:
Logic NOR function
 $y = (in1 \vee in2) \setminus$

INPUTS

Name:
name_in1
name_in2

Data Type:
BOOL
BOOL

Data Struct:
BIT
BIT

Connection:
mandatory
mandatory

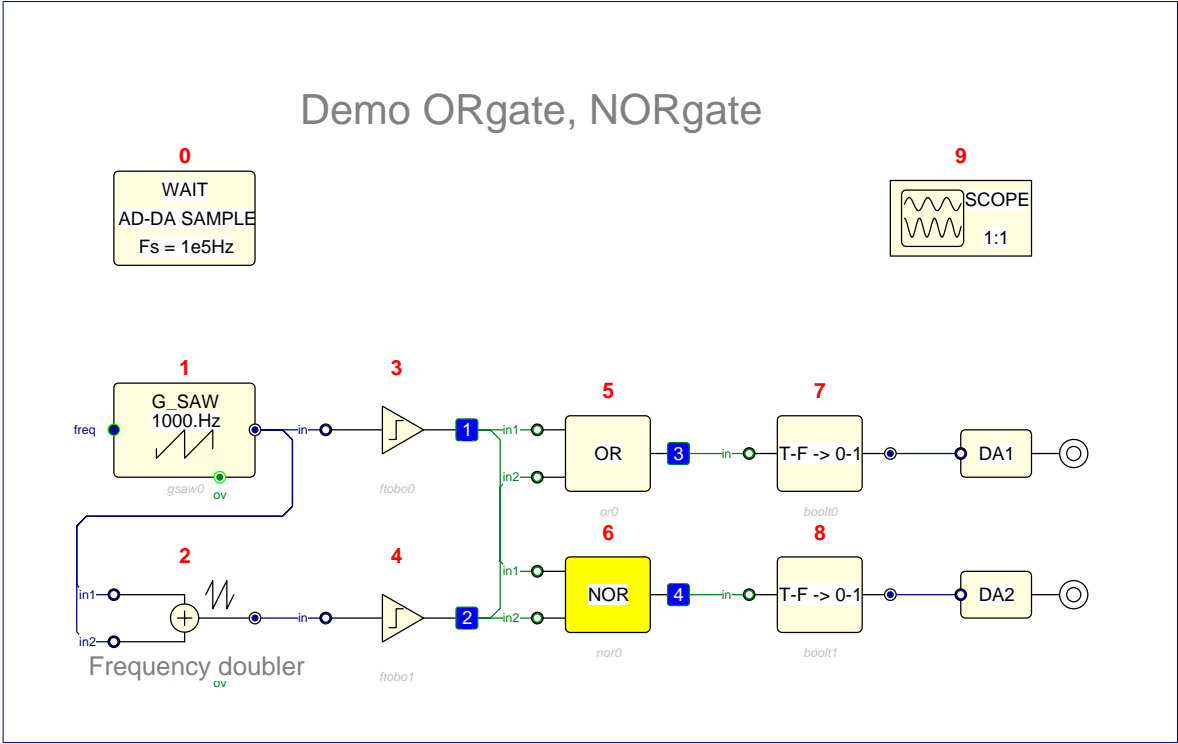
OUTPUTS

Name:
name

Data Type:
BOOL

Data Struct:
BIT

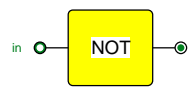
Connection:
normal



NORGATE test program

NOTGATE

NOTGATE



CATEGORY: LOGIC

INPUTS

Name:
name_in

Data Type:
BOOL

Data Struct:
BIT

Connection:
mandatory

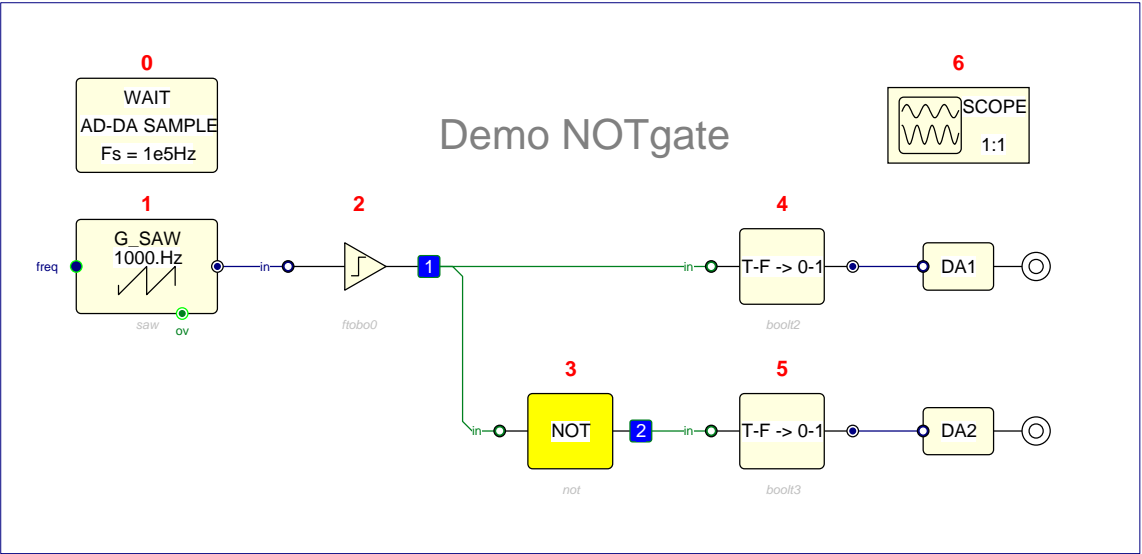
OUTPUTS

Name:
name

Data Type:
BOOL

Data Struct:
BIT

Connection:
normal

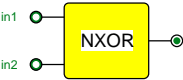


NOTGATE test program

NXORGATE

Logic NXOR function

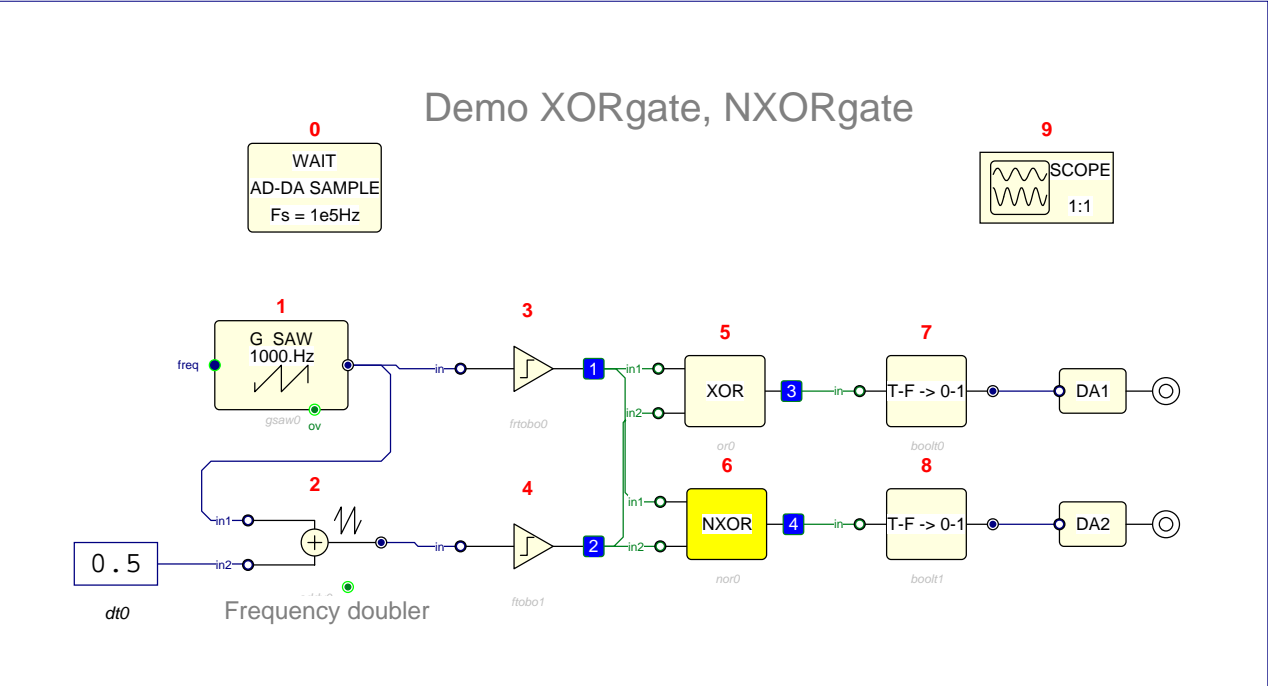
NXORGATE



CATEGORY: LOGIC

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

INPUTS			
Name:			
name_in1	Data Type: BOOL	Data Struct: BIT	Connection: mandatory
name_in2	BOOL	BIT	mandatory
OUTPUTS			
Name:			
name	Data Type: BOOL	Data Struct: BIT	Connection: normal

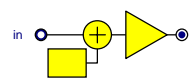


NXORGATE test program

OFFGAIN

Offset and gain:

OFFGAIN



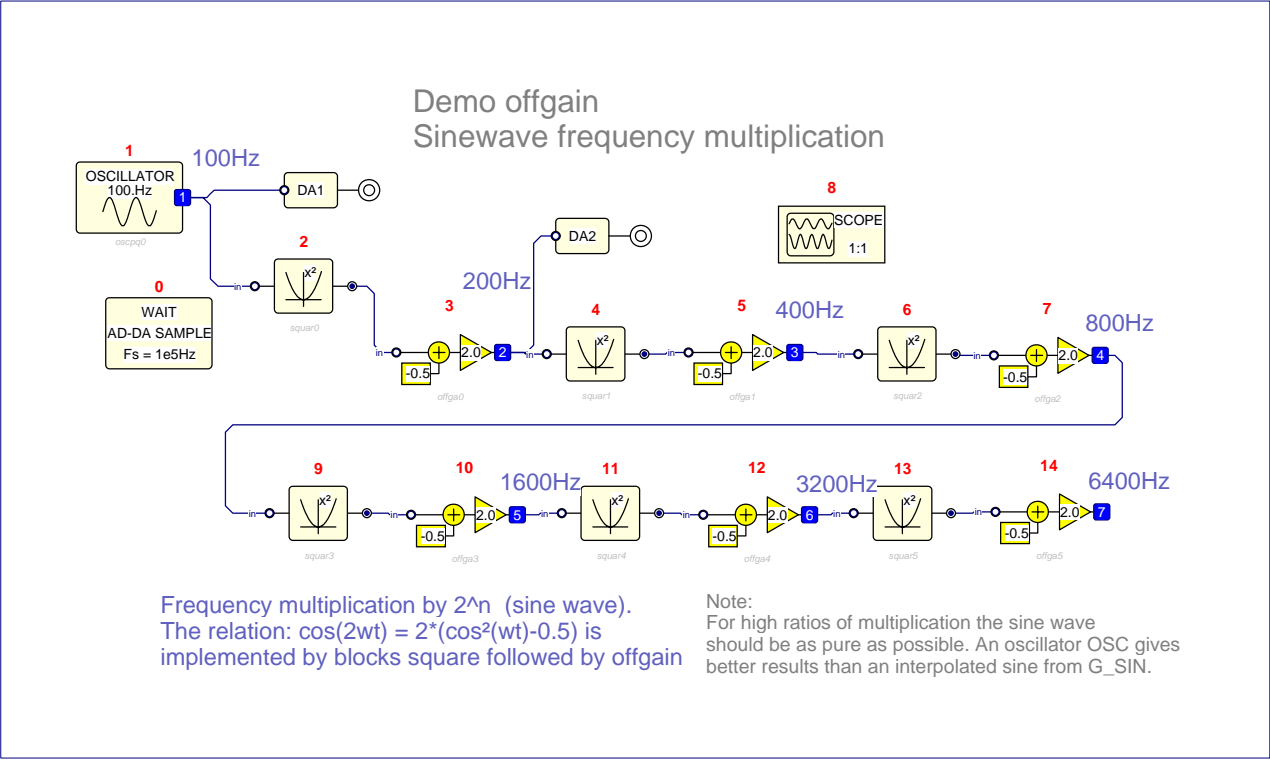
CATEGORY: ARITHMETIC

DESCRIPTION:
Offset and gain:
 $y = \text{gain} * (\text{in} + \text{offset})$

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

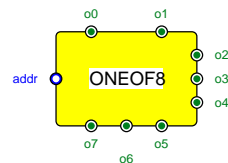


OFFGAIN test program

ONEOF8

Activate selected output

ONEOF8



CATEGORY: CONTROL

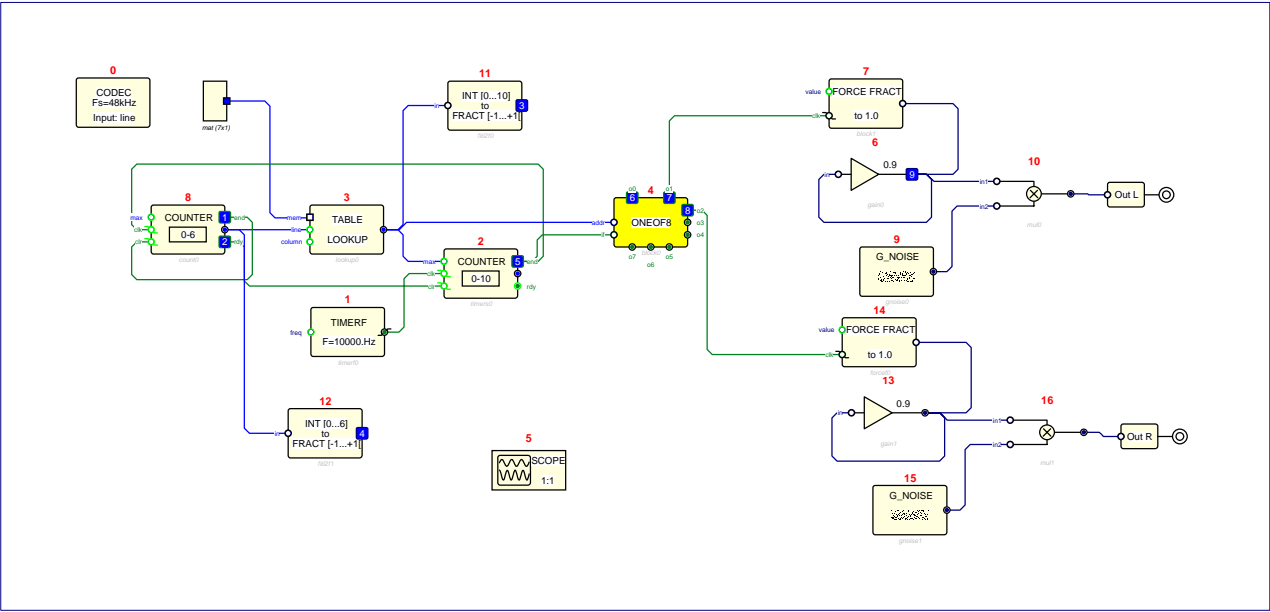
DESCRIPTION:
Activate selected output
To be used with if or if_c condition

INPUTS

Name:	Data Type:	Data Struct:	Connection:
name_addr	INTEGER	WORD	mandatory

OUTPUTS

Name:	Data Type:	Data Struct:	Connection:
name_o2	BOOL	BIT	normal
name_o3	BOOL	BIT	normal
name_o4	BOOL	BIT	normal
name_o5	BOOL	BIT	normal
name_o6	BOOL	BIT	normal
name_o7	BOOL	BIT	normal
name_o1	BOOL	BIT	normal
name_o0	BOOL	BIT	normal



ONEOF8 test program

ORGAN

ORGAN



CATEGORY: MUSIC

INPUTS

Name:
name_start
name_freq
name_vol

Data Type:
BOOL
FRACT
FRACT

Data Struct:
BIT
WORD
WORD

Connection:
mandatory
mandatory
optional

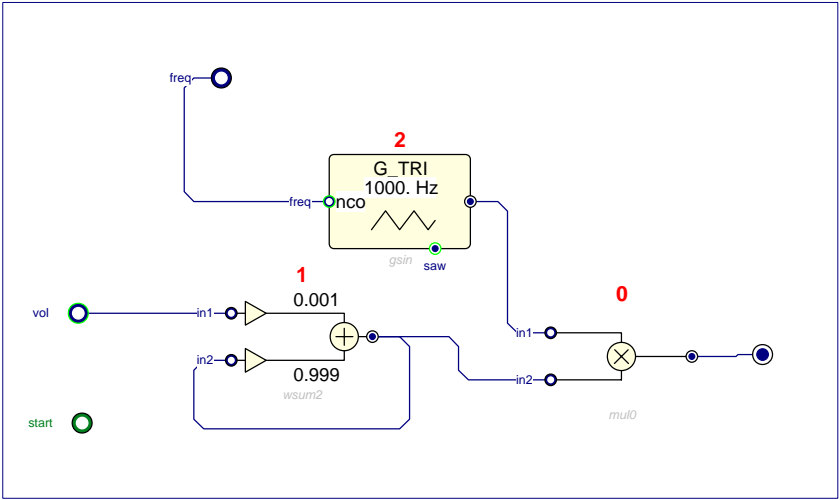
OUTPUTS

Name:
name

Data Type:
FRACT

Data Struct:
WORD

Connection:
normal

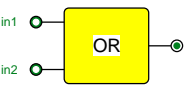


ORGAN internal schema

ORGATE

Logic OR function

ORGATE



CATEGORY: LOGIC

DESCRIPTION:
Logic OR function
 $y = in1 \vee in2$

INPUTS

Name:
name_in1
name_in2

Data Type:
BOOL
BOOL

Data Struct:
BIT
BIT

Connection:
mandatory
mandatory

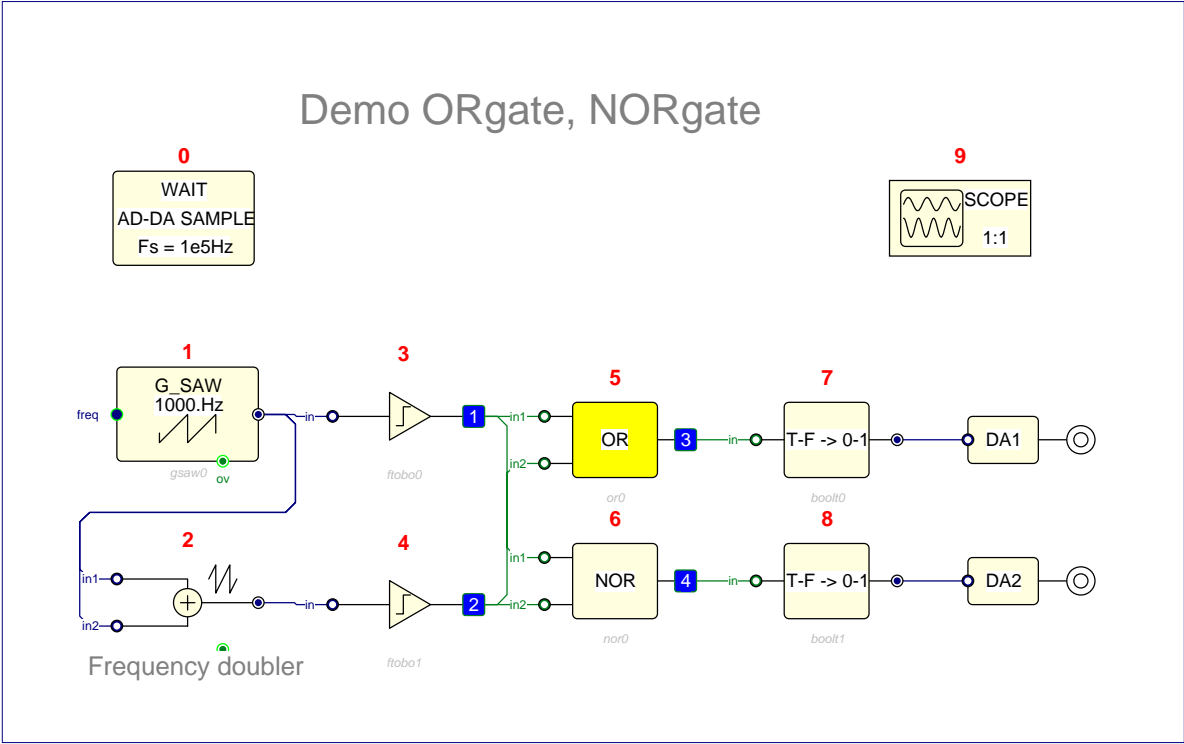
OUTPUTS

Name:
name

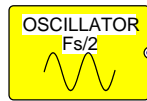
Data Type:
BOOL

Data Struct:
BIT

Connection:
normal



ORGATE test program



CATEGORY: GENERATORS

DESCRIPTION:
High purity sine oscillator

PARAMETERS:

Parameter:
Frequency
Unit

Default values:
1000.
Hz,Fs/2

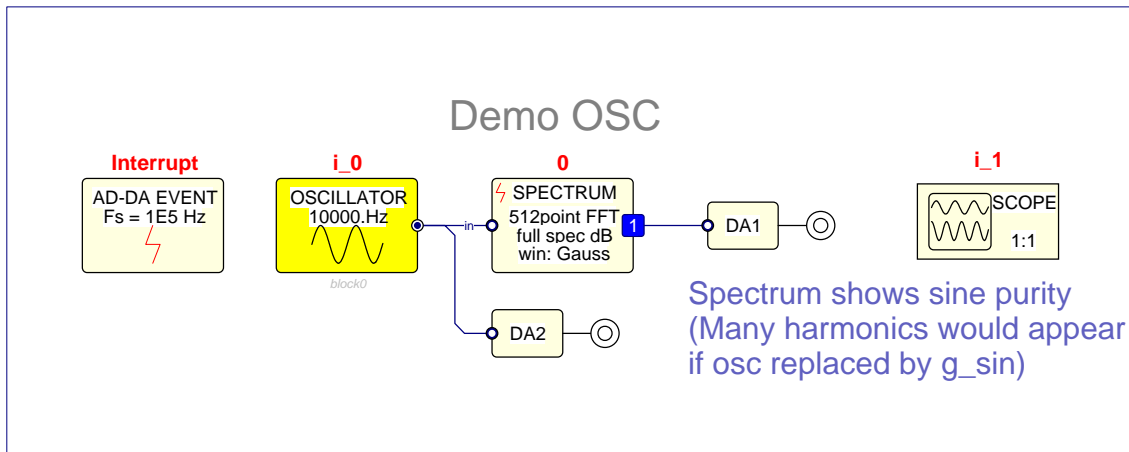
OUTPUTS

Name:
name

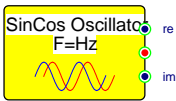
Data Type:
FRACT

Data Struct:
WORD

Connection:
normal



OSC test program

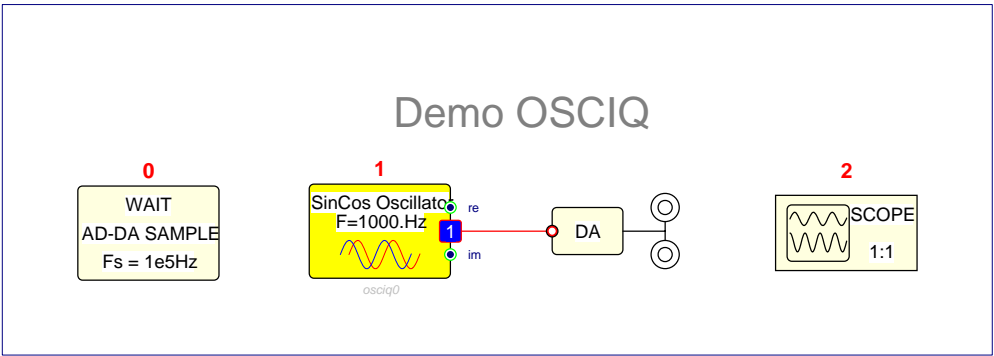


CATEGORY: GENERATORS

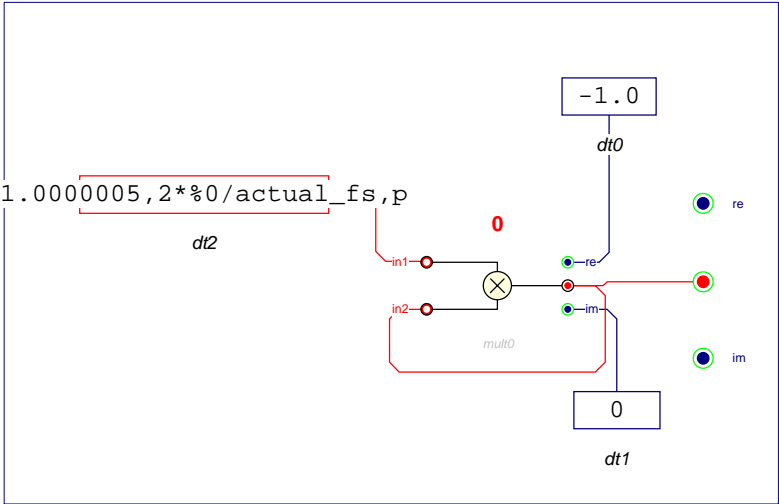
DESCRIPTION:
Phase quadrature oscillator

PARAMETERS:
Parameter: Frequency
Default values: 1000.

OUTPUTS			
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name	COMPLEX	WORD	optional
name_re	FRACT	WORD	optional
name_im	FRACT	WORD	optional



OSCIQ test program

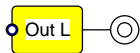


OSCIQ internal schema

OUT_L

Codec output Left

OUT_L



CATEGORY: AUDIO

DESCRIPTION:
Codec output Left
Digital to Analog Converter input

INPUTS

<i>Name:</i> name	<i>Data Type:</i> FRACT	<i>Data Struct:</i> WORD	<i>Connection:</i> mandatory
----------------------	----------------------------	-----------------------------	---------------------------------

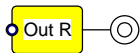
ATTRIBUTES

Non executable, Unique,

OUT_R

Codec output Right

OUT_R



CATEGORY: AUDIO

DESCRIPTION:
Codec output Right
Digital to Analog Converter input

INPUTS

<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name	FRACT	WORD	mandatory

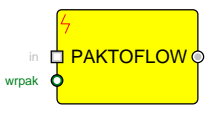
ATTRIBUTES

Non executable, Unique,

PAKTOFLOW

Packets to Flow

PAKTOFLOW



CATEGORY: MATRIX

DESCRIPTION:
Packets to Flow

PARAMETERS:
Parameter:
Buffer size
Packet size

Default values:
1024
512

INPUTS
Name:
name_in
name_wrpak

Data Type:
defined by cn
BOOL

Data Struct:
Matrix of
BIT

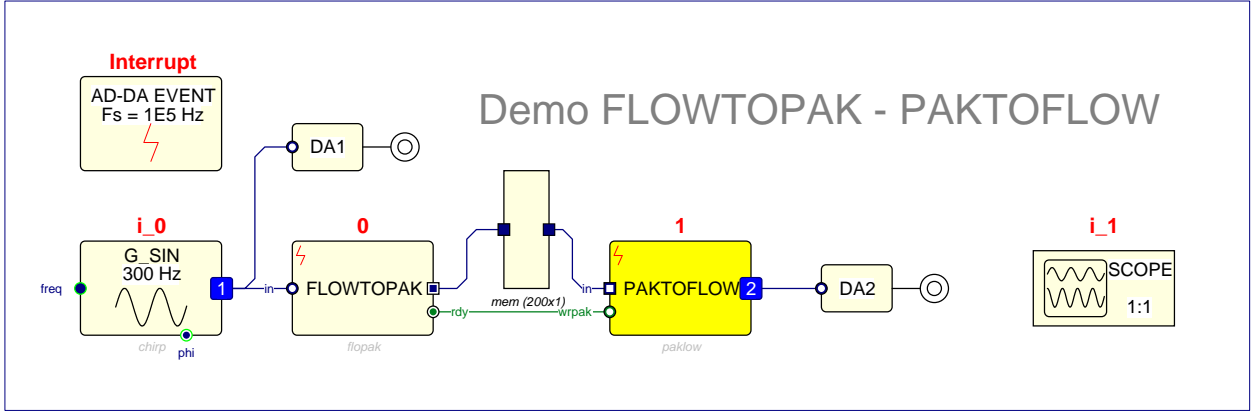
Connection:
mandatory
mandatory

OUTPUTS
Name:
name

Data Type:
defined by cn

Data Struct:

Connection:
normal

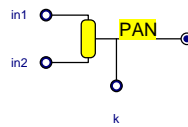


PAKTOFLOW test program

PAN

Panoramic

PAN



CATEGORY: AUDIO

DESCRIPTION:

Panoramic

$$y = (x1+x2)/2 + k(x1-x2)/2$$

$$y = \text{in1 if } k=1 \quad y = (\text{in1}+\text{in2})/2 \text{ if } k=0 \quad y = \text{in2 if } k=-1$$

INPUTS

Name:

name_in1

name_in2

name_k

Data Type:

FRACT

FRACT

FRACT

Data Struct:

WORD

WORD

WORD

Connection:

mandatory

mandatory

mandatory

OUTPUTS

Name:

name

Data Type:

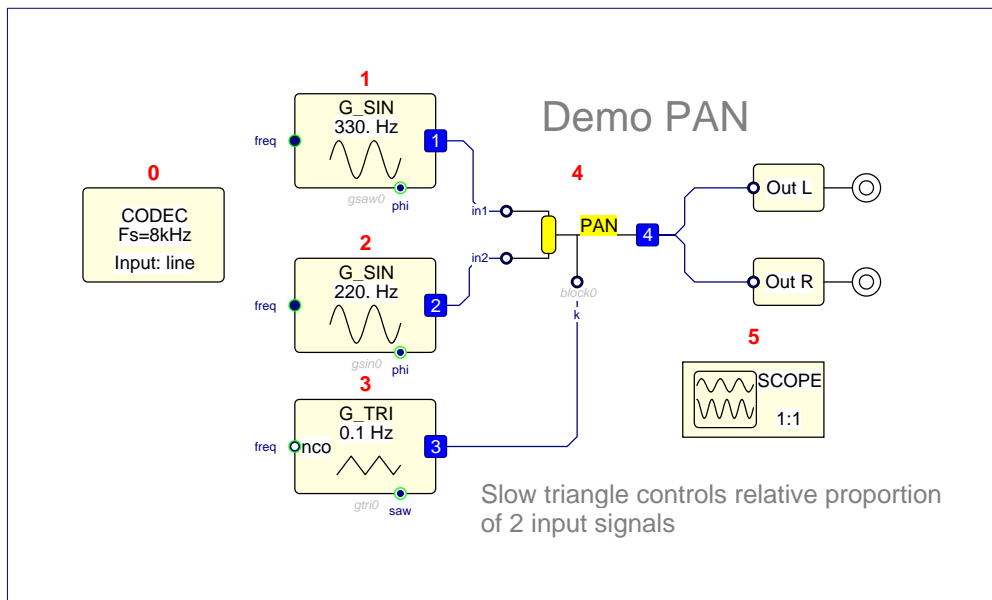
FRACT

Data Struct:

WORD

Connection:

normal



PAN test program



CATEGORY: MUSIC

DESCRIPTION:
MIDI File
Transcribed in asm format

OUTPUTS

<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name	INTEGER	Matrix of WORD	normal

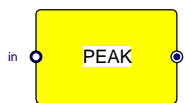
ATTRIBUTES

Non executable, Unique, Data Table

PEAK

Get peak value of input

PEAK



CATEGORY: CONTROL

DESCRIPTION:
Get peak value of input

INPUTS

Name:
name_in

Data Type:
FRACT

Data Struct:
WORD

Connection:
mandatory

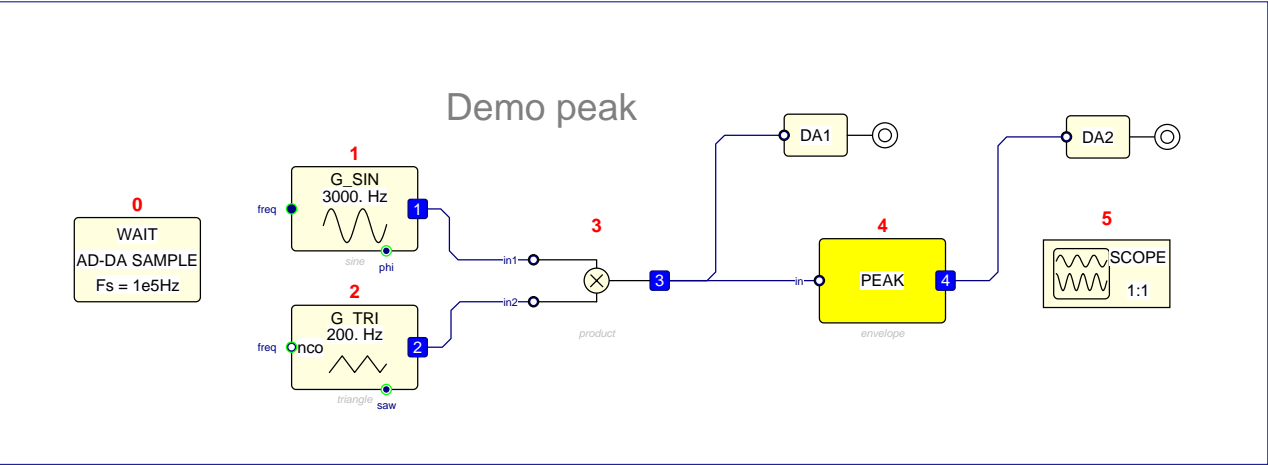
OUTPUTS

Name:
name

Data Type:
FRACT

Data Struct:
WORD

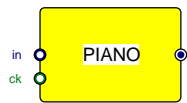
Connection:
normal



PEAK test program

PIANO

PIANO



CATEGORY: AUDIO

INPUTS

Name:
name_in
name_ck

Data Type:
FRACT
BOOL

Data Struct:
WORD
BIT

Connection:
mandatory
mandatory

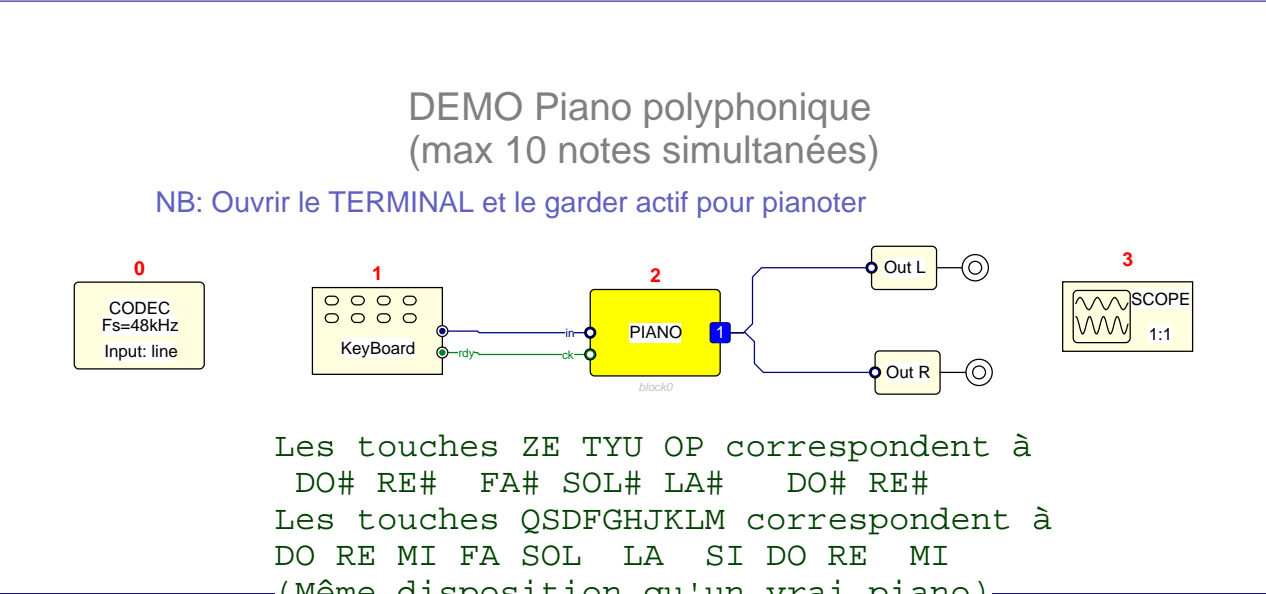
OUTPUTS

Name:
name

Data Type:
FRACT

Data Struct:
WORD

Connection:
normal



PIANO test program

PISO

Parallel In Serial Out

PISO



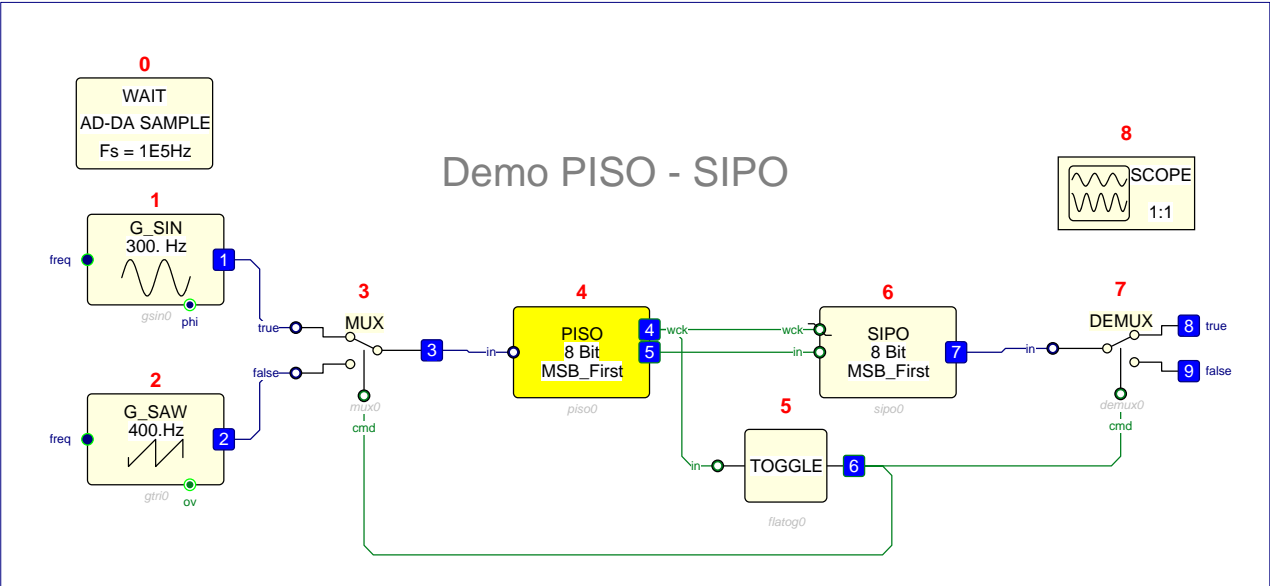
CATEGORY: CONTROL

DESCRIPTION:
Parallel In Serial Out
Serializing Shift Register function

PARAMETERS:
Parameter:
Word Length
Shift Direction
Default values:
8,12,16
MSB_First,LSB_First

INPUTS			
<i>Name:</i> name_in	<i>Data Type:</i> FRACT	<i>Data Struct:</i> WORD	<i>Connection:</i> mandatory

OUTPUTS			
<i>Name:</i> name	<i>Data Type:</i> BOOL	<i>Data Struct:</i> BIT	<i>Connection:</i> normal
<i>Name:</i> name_wck	<i>Data Type:</i> BOOL	<i>Data Struct:</i> BIT	<i>Connection:</i> normal

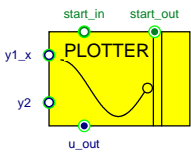


PISO test program

PLOTTER

Slow signal plotter

PLOTTER



CATEGORY: INSTRUMENTS

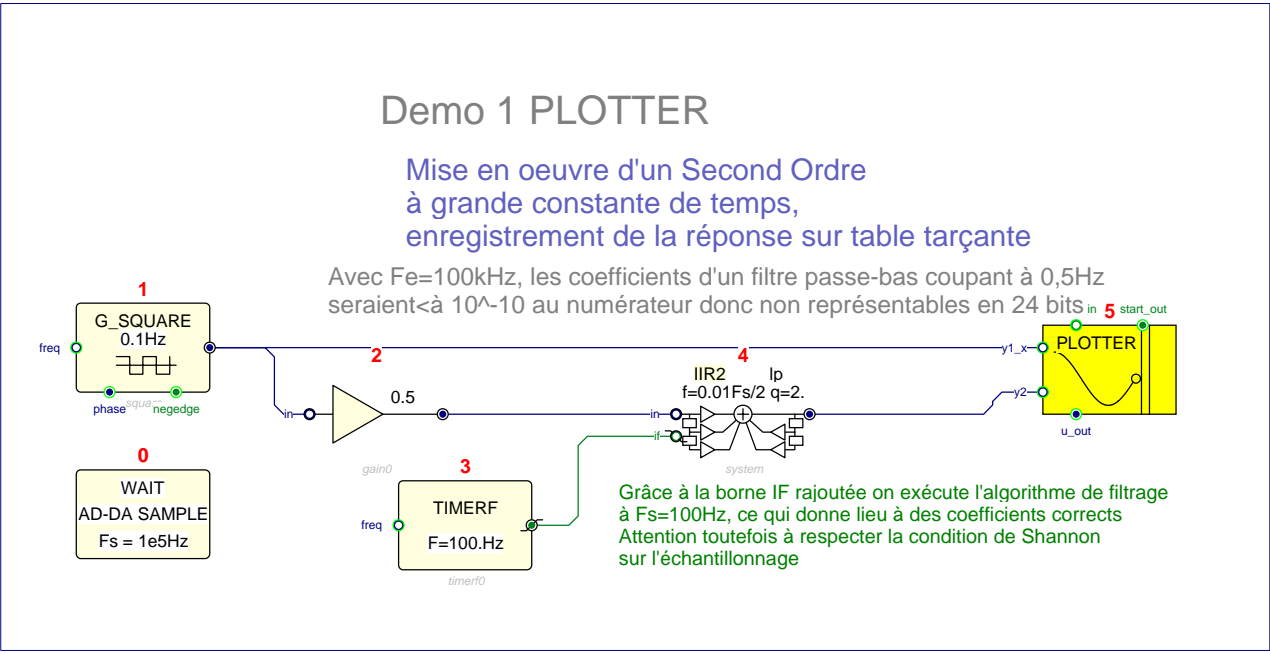
DESCRIPTION:
Slow signal plotter

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
Y1 max or R max	1.0
Y1 min or R min	-1.0
Y1 or R unit	—
Y2 max	1.0
Y2 min	-1.0
Y2 unit	—
Uout_Mode	Full_range,Linear,Geometric,Frequency_Linear,Frequency_Geometric
Uout_max	1.0
Uout_min	-1.0
Uout_Unit	—

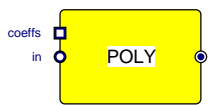
INPUTS		Data Type:	Data Struct:	Connection:
Name:				
name_y1_x		FRACT	WORD	optional
name_y2		FRACT	WORD	optional
name_start_in		BOOL	BIT	optional

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

ATTRIBUTES
Unique,



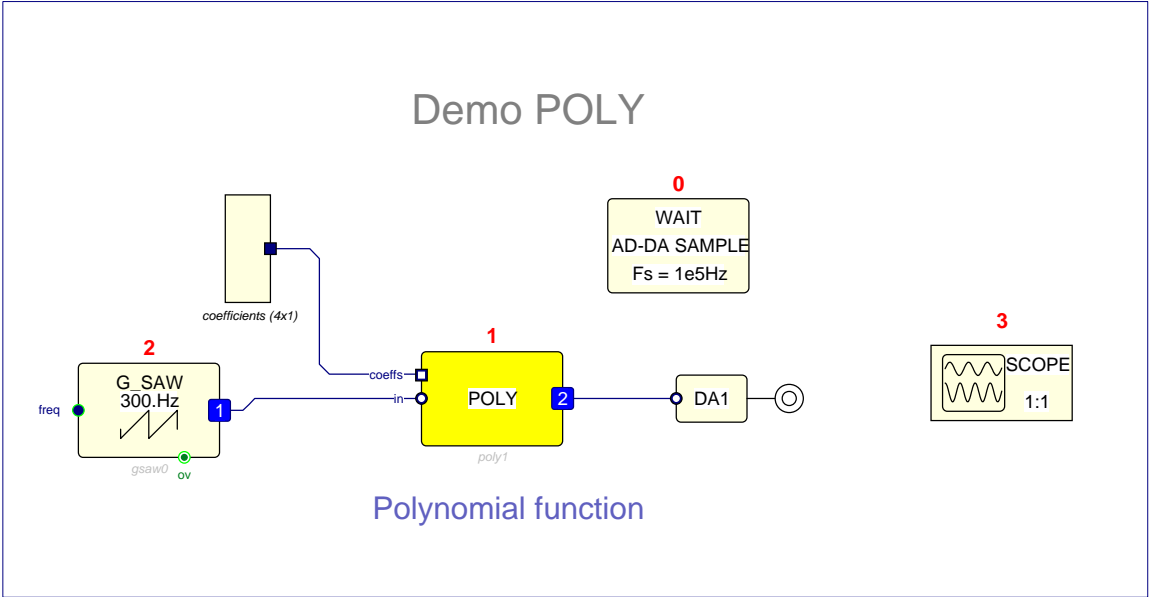
PLOTTER test program



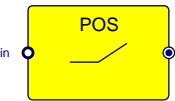
CATEGORY: FUNCTIONS

DESCRIPTION:
Real Polynomial function
Table contains coefficients in order 1, x, x², ...

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



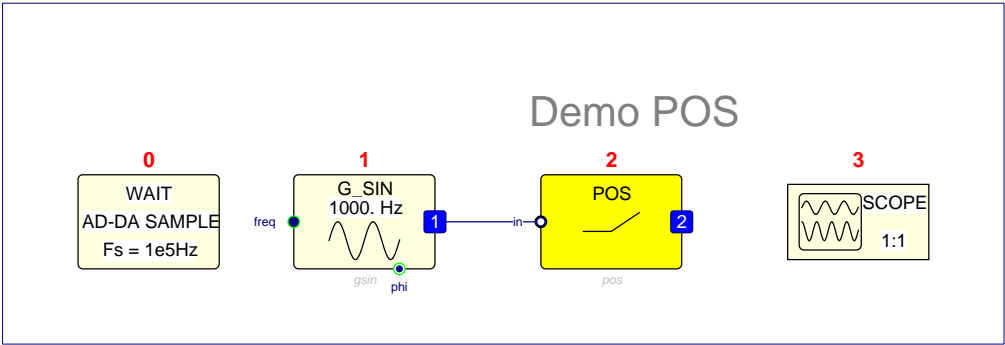
POLY test program



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

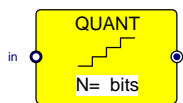


POS test program

QUANT

Quantize data to n bits

QUANT



CATEGORY: NON LINEAR

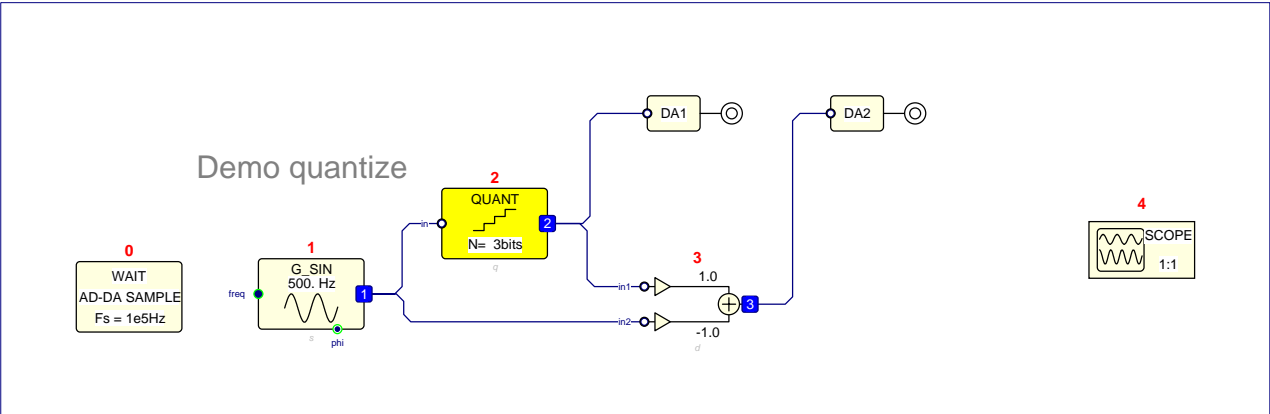
DESCRIPTION:
Quantize data to n bits

PARAMETERS:
Parameter:
bits
Approx: rnd/exss/deflt

Default values:
3
r,e,d

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

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



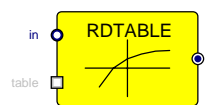
QUANT test program

RCPULSE

RDTABLE

Read interpolate table

RDTABLE



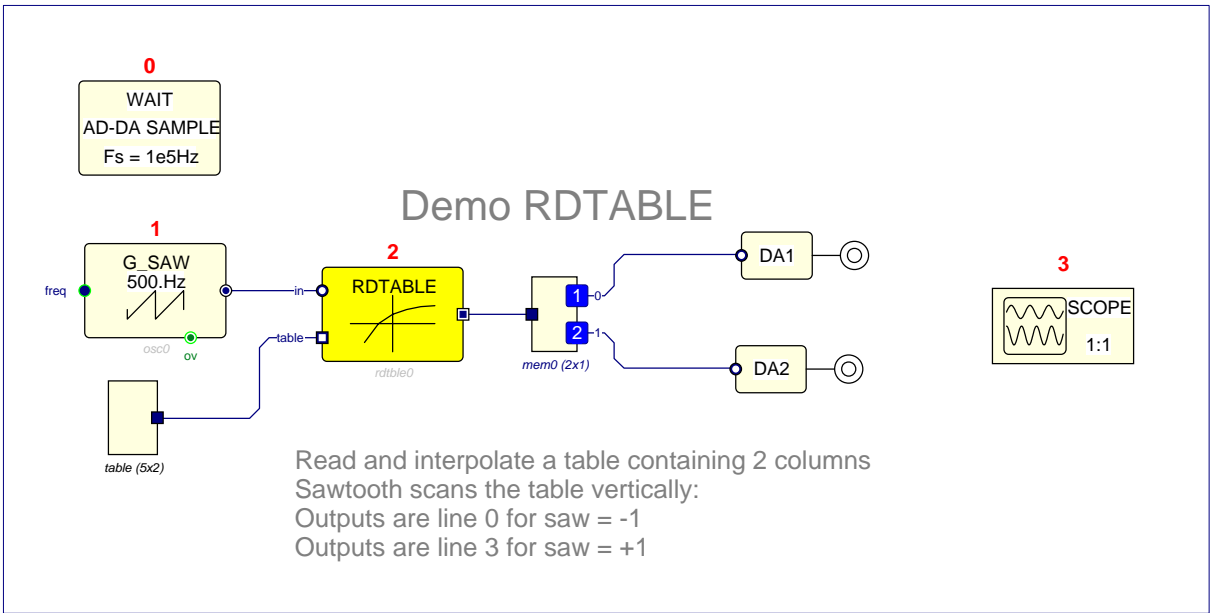
CATEGORY: FUNCTIONS

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 column (0, 1, 2 ..)

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

INPUTS			
Name:	Data Type:	Data Struct:	Connection:
name_in	FRACT	WORD	mandatory
name_table	defined by cn	Matrix of	mandatory

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

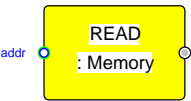


RDTABLE test program

READMEM

Read Memory

READMEM



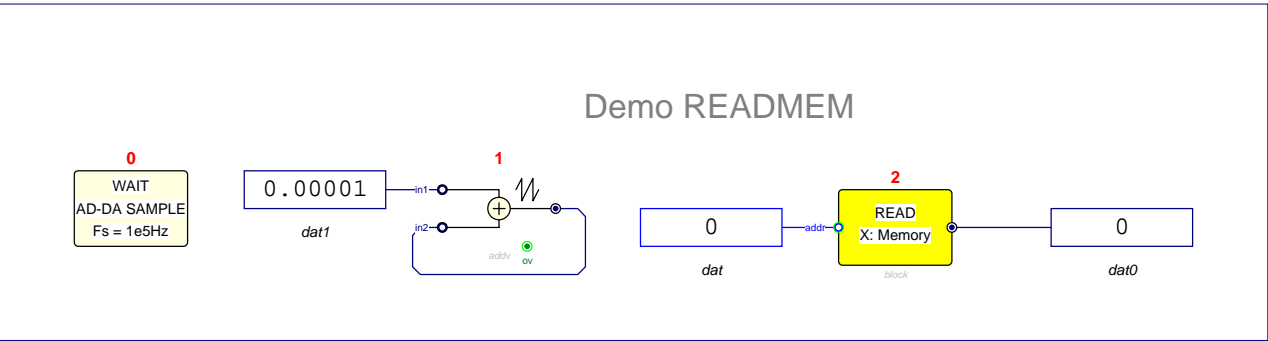
CATEGORY: INSTRUMENTS

DESCRIPTION:
Read Memory
Gives memory content at specified address

PARAMETERS:
Parameter: *Default values:*
Address 0
Memory field X,Y,L,P

INPUTS			
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name_addr	INTEGER	WORD	optional

OUTPUTS			
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name	defined by cn		normal



READMEM test program

REQUIEM

MIDI File

REQUIEM



CATEGORY: MUSIC

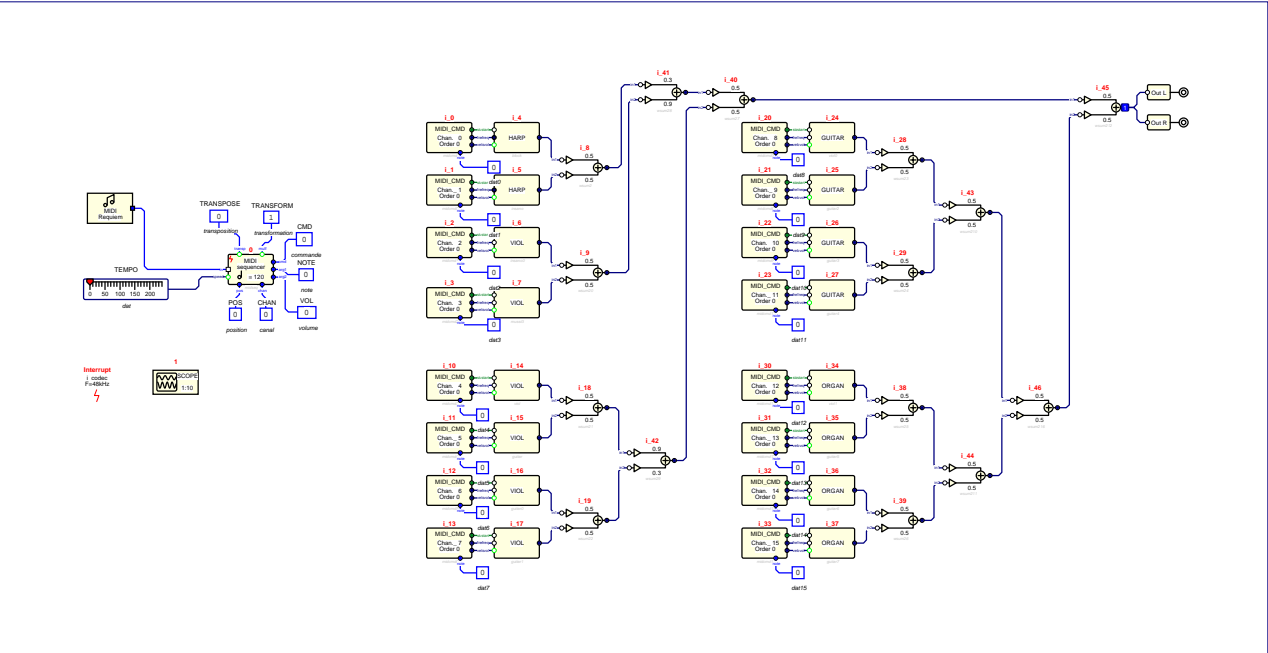
DESCRIPTION:
MIDI File
Transcripted in asm format

OUTPUTS

Name:	Data Type:	Data Struct:	Connection:
name	INTEGER	Matrix of WORD	normal

ATTRIBUTES

Non executable, Unique, Data Table

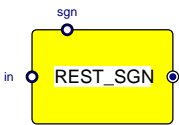


REQUIEM test program

REST_SGN

Restore Sign

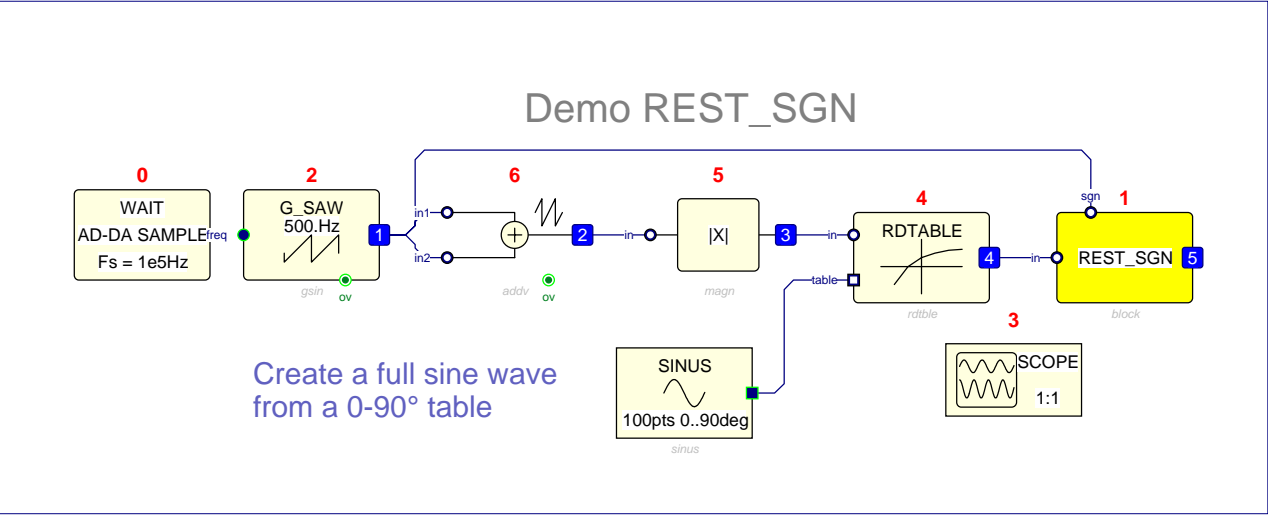
REST_SGN



CATEGORY: CONTROL

DESCRIPTION:
Restore Sign
Output = Input if sgn>0 else output=-Input

INPUTS			
Name:	Data Type:	Data Struct:	Connection:
name_in	FRACT	WORD	mandatory
name_sgn	FRACT	WORD	mandatory
OUTPUTS			
Name:	Data Type:	Data Struct:	Connection:
name	FRACT	WORD	normal



REST_SGN test program

REVERB

Add reverberation to sound

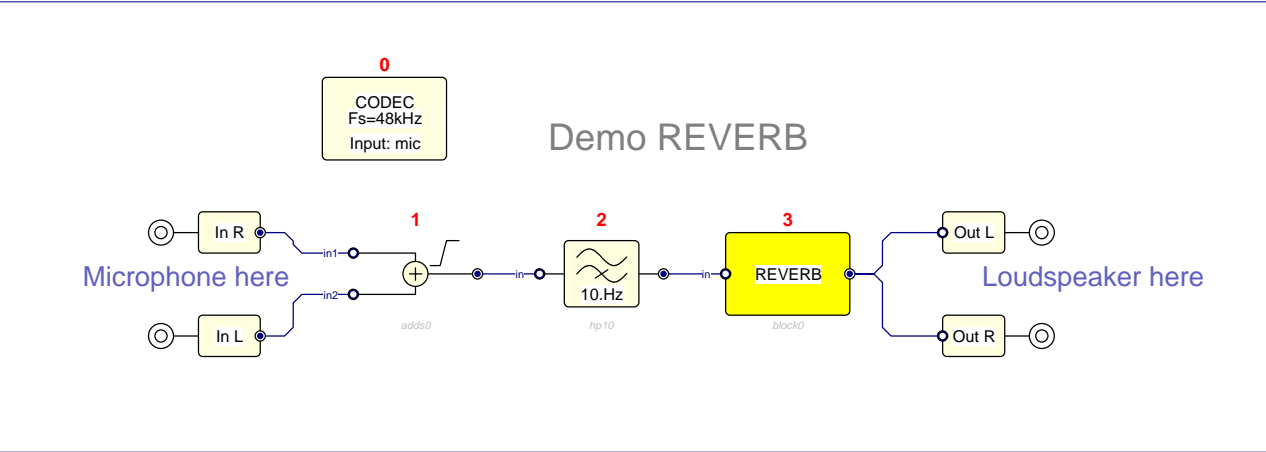
REVERB



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



DESCRIPTION:
MIDI File
Transcripted in asm format

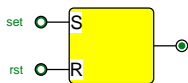
OUTPUTS			
<i>Name:</i> name	<i>Data Type:</i> INTEGER	<i>Data Struct:</i> Matrix of WORD	<i>Connection:</i> normal

ATTRIBUTES
Non executable, Unique, Data Table

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:

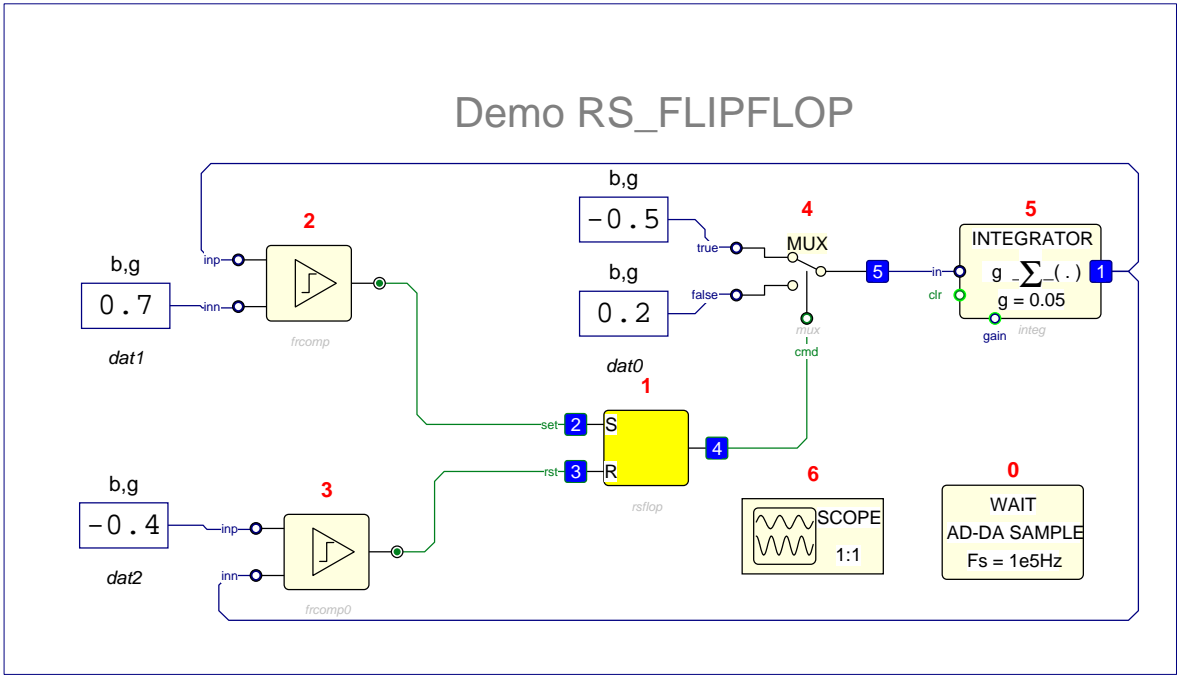
Parameter: Dominant state: Default values: SET,RESET

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

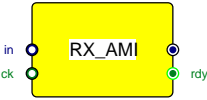


RS_FLIPFLOP test program

RX_AMI

AMI decoder

RX_AMI



CATEGORY: TELECOM

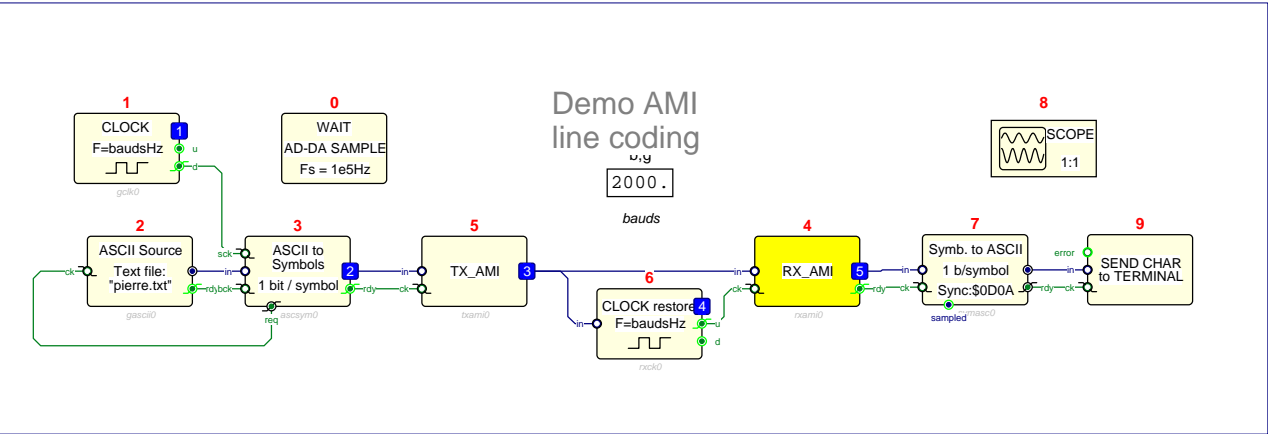
DESCRIPTION:
AMI decoder
Alternate Mark Inversion line decoder

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	optional

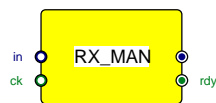


RX_AMI test program

RX_MAN

Manchester line decoder

RX_MAN



CATEGORY: TELECOM

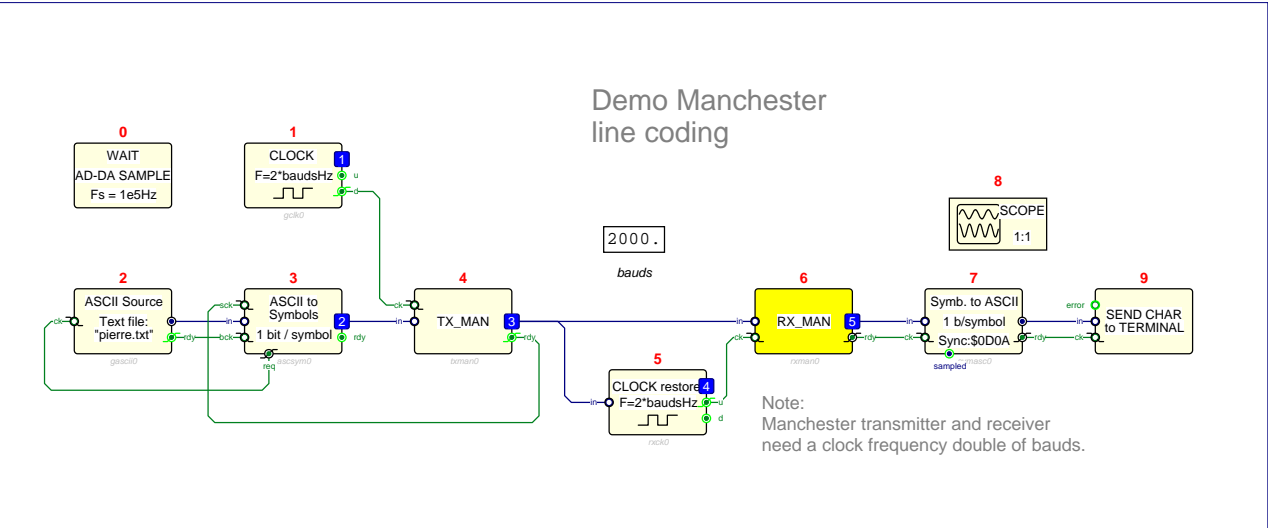
DESCRIPTION:
Manchester line decoder
Input clock is 2 x Bauds

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

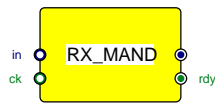


RX_MAN test program

RX_MAND

D-Manchester decoder

RX_MAND



CATEGORY: TELECOM

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

INPUTS

Name:
name_in
name_ck

Data Type:
FRACT
BOOL

Data Struct:
WORD
BIT

Connection:
mandatory
mandatory

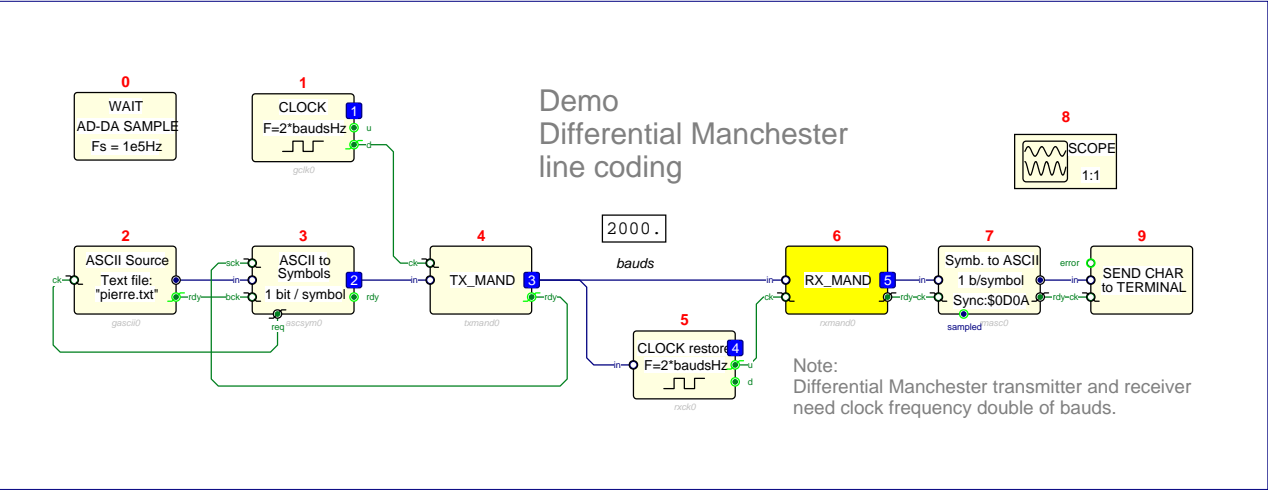
OUTPUTS

Name:
name
name_rdy

Data Type:
FRACT
BOOL

Data Struct:
WORD
BIT

Connection:
normal
normal

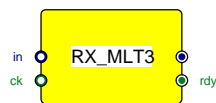


RX_MAND test program

RX_MLT3

MLT3 line decoder

RX_MLT3



CATEGORY: TELECOM

DESCRIPTION:
MLT3 line decoder

INPUTS

Name:
name_in
name_ck

Data Type:
FRACT
BOOL

Data Struct:
WORD
BIT

Connection:
mandatory
mandatory

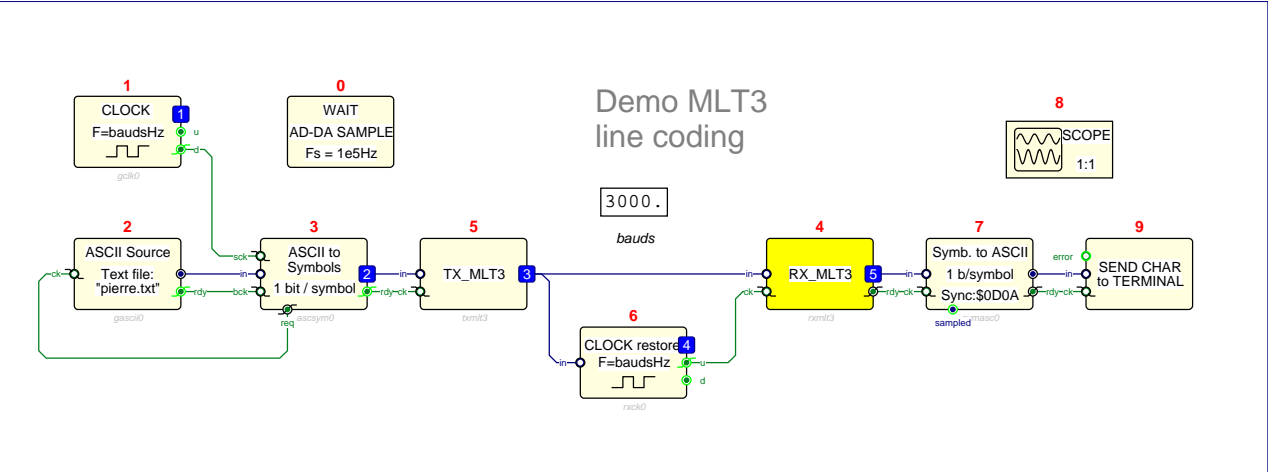
OUTPUTS

Name:
name
name_rdy

Data Type:
FRACT
BOOL

Data Struct:
WORD
BIT

Connection:
normal
normal

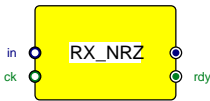


RX_MLT3 test program

RX_NRZ

NRZ decoder

RX_NRZ



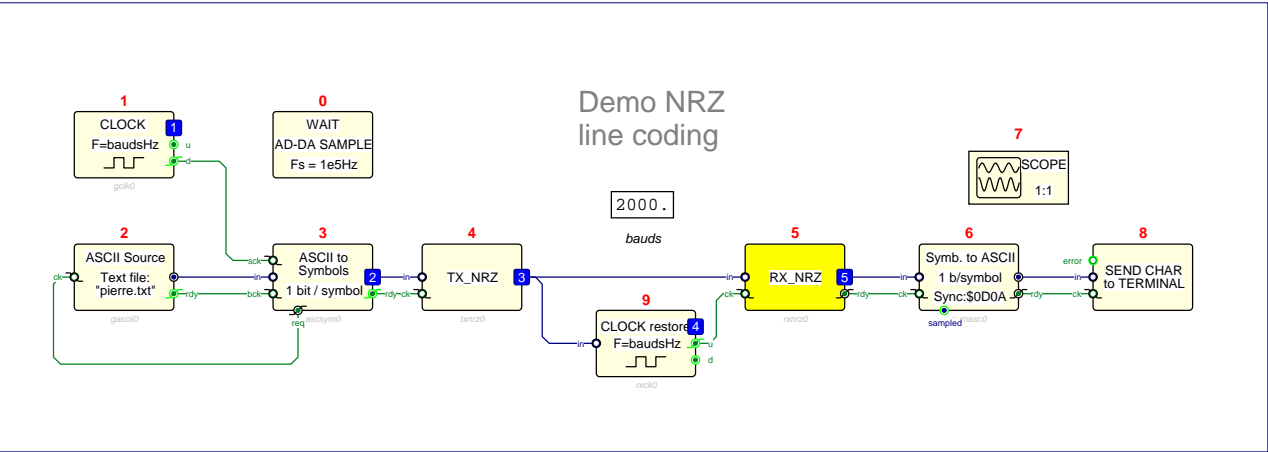
CATEGORY: TELECOM

DESCRIPTION:
NRZ decoder
Non Return to Zero line decoder

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

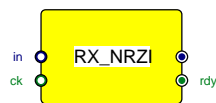


RX_NRZ test program

RX_NRZI

NRZI line decoder

RX_NRZI



CATEGORY: TELECOM

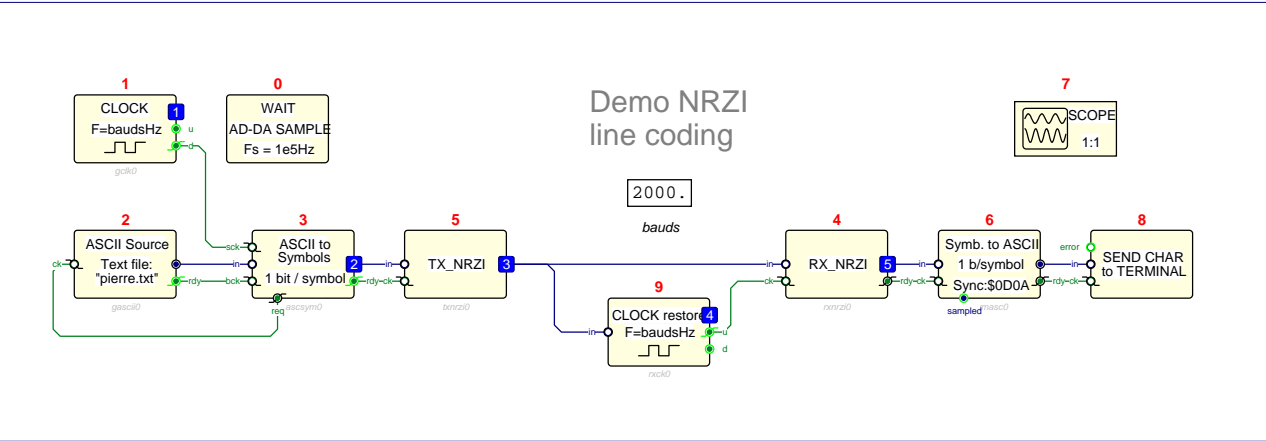
DESCRIPTION:
NRZI line decoder

PARAMETERS:

<i>Parameter:</i>	<i>Default values:</i>
Level Space	-1.0
Level Mark	1.0

INPUTS			
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name_in	FRACT	WORD	mandatory
name_ck	BOOL	BIT	mandatory

OUTPUTS			
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name	FRACT	WORD	normal
name_rdy	BOOL	BIT	normal

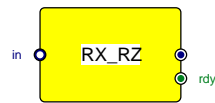


RX_NRZI test program

RX_RZ

Return to Zero line decoder

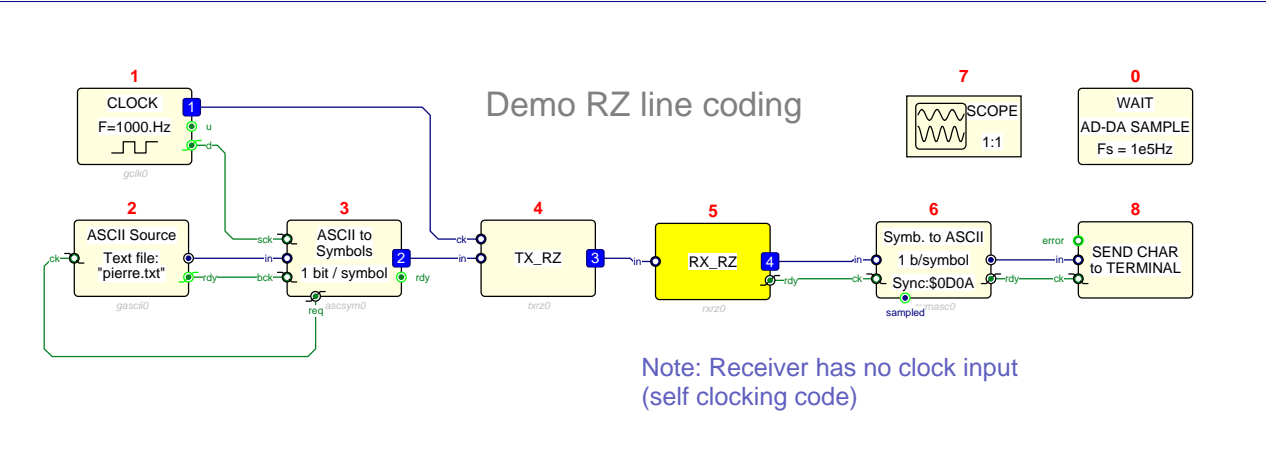
RX_RZ



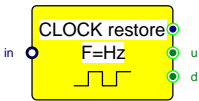
CATEGORY: TELECOM

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



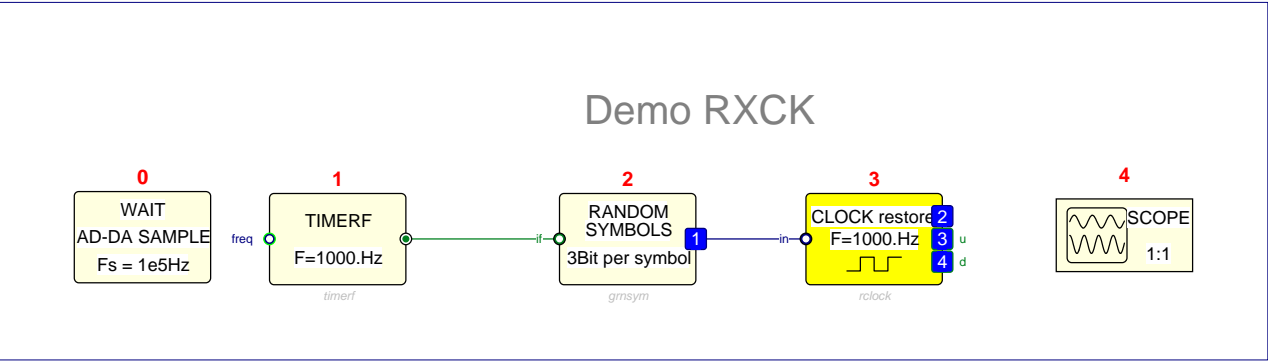
CATEGORY: TELECOM

DESCRIPTION:
Clock restoration
Retrives optimal clock phase at receiver end

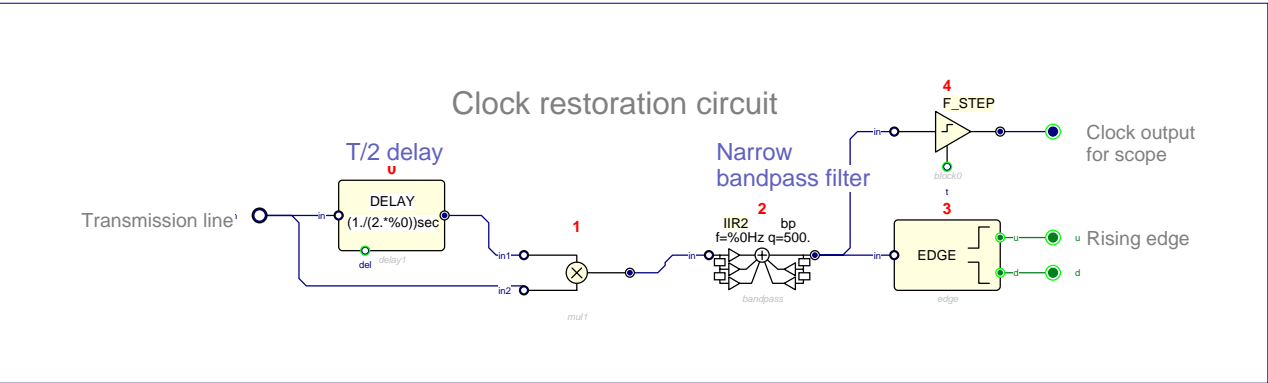
PARAMETERS:
Parameter: Frequency
Default values: 100.

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 test program

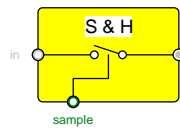


RXCK internal schema

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:

Sample on ..

Default values:

pos_edge,neg_edge,true

INPUTS

Name:

name_in

name_sample

Data Type:

defined by cn

BOOL

Data Struct:

BIT

Connection:

mandatory

mandatory

OUTPUTS

Name:

name

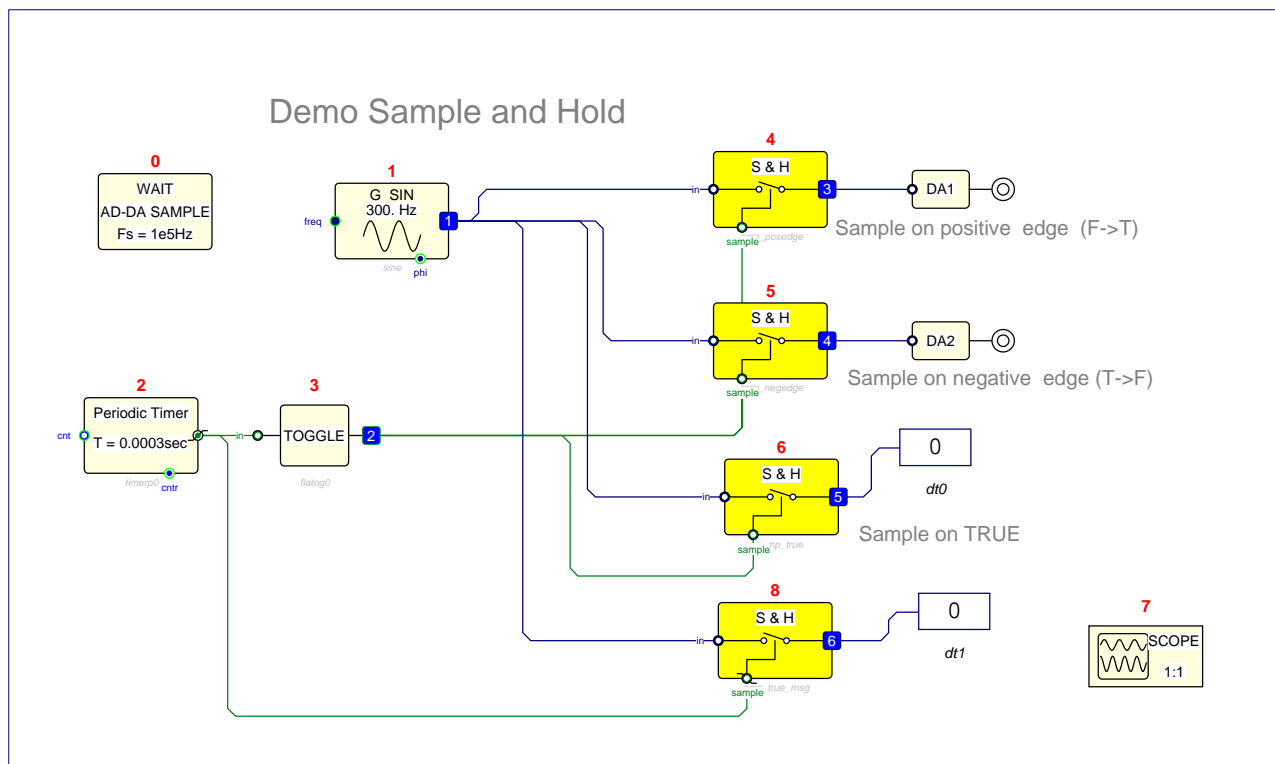
Data Type:

defined by cn

Data Struct:

Connection:

normal

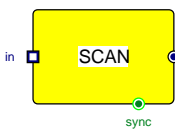


SAMPHOLD test program

SCAN

Scan buffer

SCAN



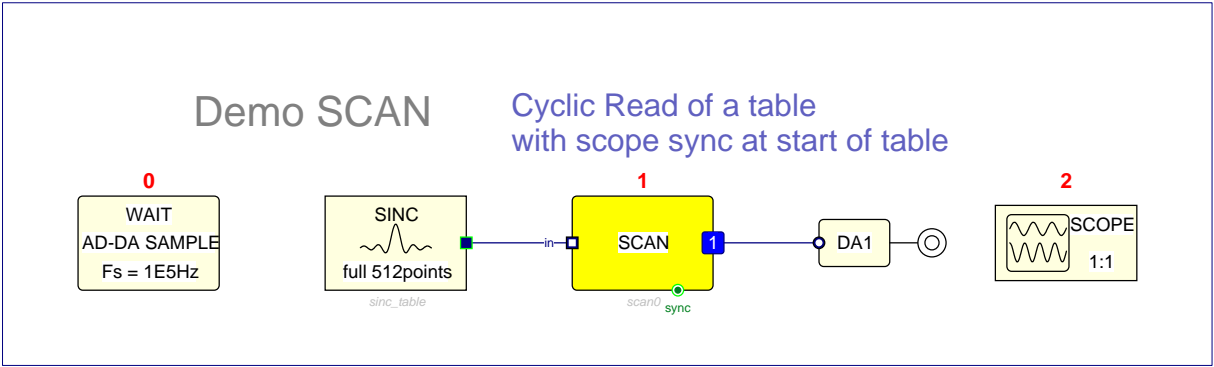
CATEGORY: CONTROL

DESCRIPTION:
Scan buffer
Read circular buffer one point at each sample
S/W synchronize scope when pointer is at begin of buffer

PARAMETERS:
Parameter: Gain
Default values: 1.0

INPUTS	Data Type:	Data Struct:	Connection:
<i>Name:</i> name_in	FRACT	Matrix of WORD	mandatory

OUTPUTS	Data Type:	Data Struct:	Connection:
<i>Name:</i> name_sync	FRACT BOOL	WORD BIT	normal optional

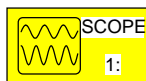


SCAN test program

SCOPE

Multi Channel Scope

SCOPE



CATEGORY: INSTRUMENTS

DESCRIPTION:

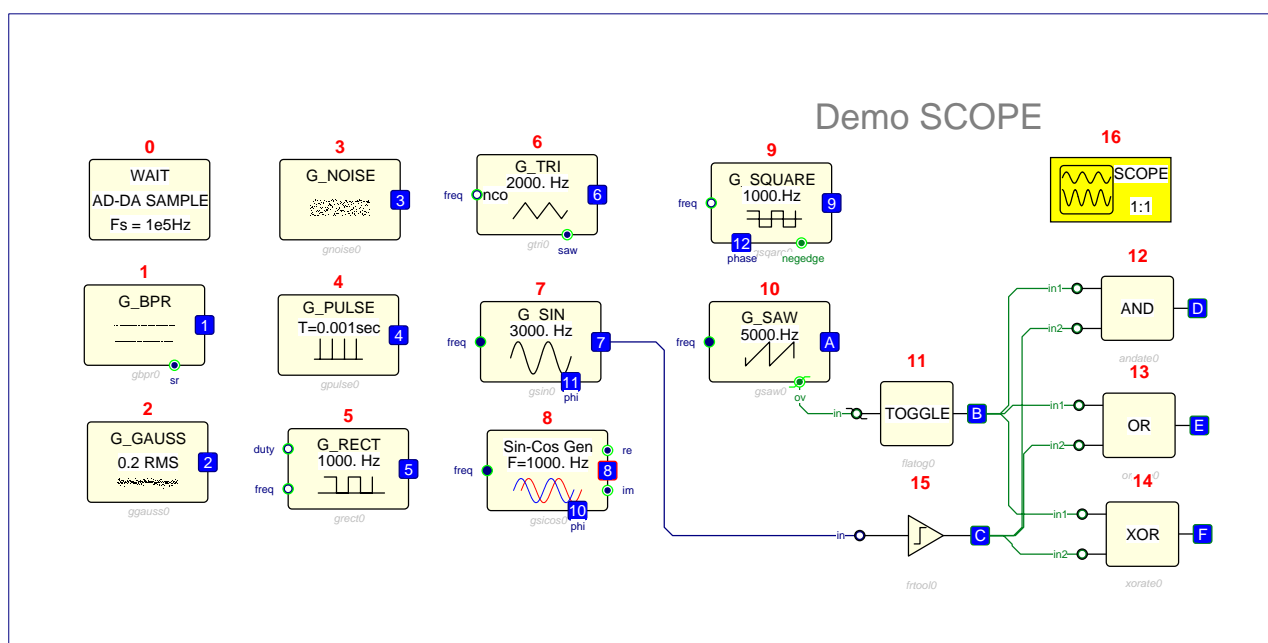
Multi Channel Scope
Signals to view are defined by PROBES

PARAMETERS:

Parameter: Decimation factor
Default values: 1

ATTRIBUTES

Unique,

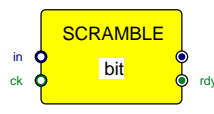


SCOPE test program

SCRAMBLE

N-bit scrambler

SCRAMBLE



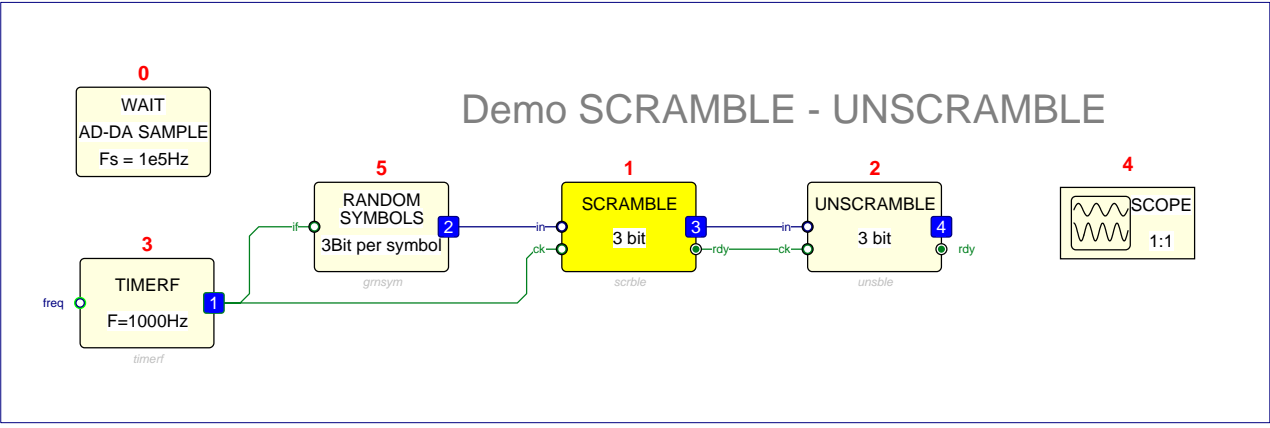
CATEGORY: TELECOM

DESCRIPTION:
N-bit scrambler
for equalizing symbol distribution

PARAMETERS:
Parameter: Bits
Default values: 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



SCRAMBLE test program

SDRAM

Install SDRAM PortA interface

SDRAM



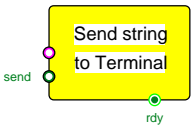
DESCRIPTION:
Install SDRAM PortA interface

ATTRIBUTES
Execute at Init, Unique,

SEND_STR

Send string

SEND_STR



CATEGORY: STRING

DESCRIPTION:
Send string
to text Terminal

INPUTS

Name:
name
name_send

Data Type:
STRING
BOOL

Data Struct:
WORD
BIT

Connection:
mandatory
mandatory

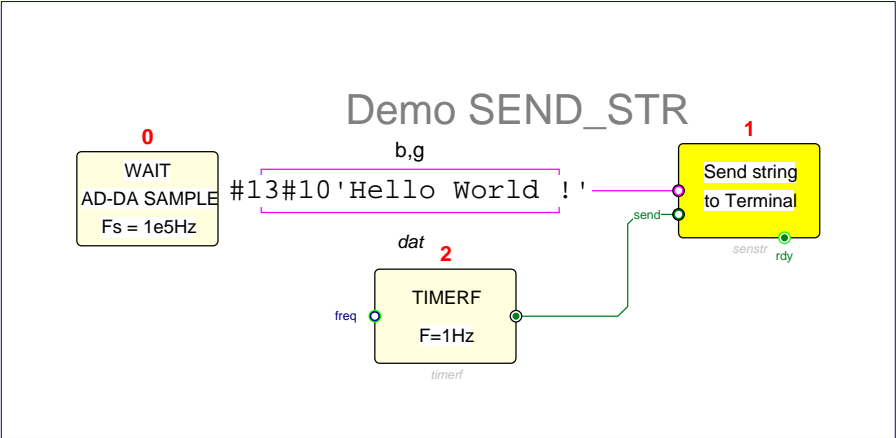
OUTPUTS

Name:
name_rdy

Data Type:
BOOL

Data Struct:
BIT

Connection:
optional

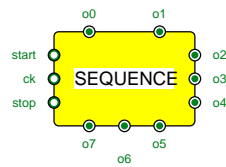


SEND_STR test program

SEQUENCE

Sequence generator

SEQUENCE



CATEGORY: CONTROL

DESCRIPTION:

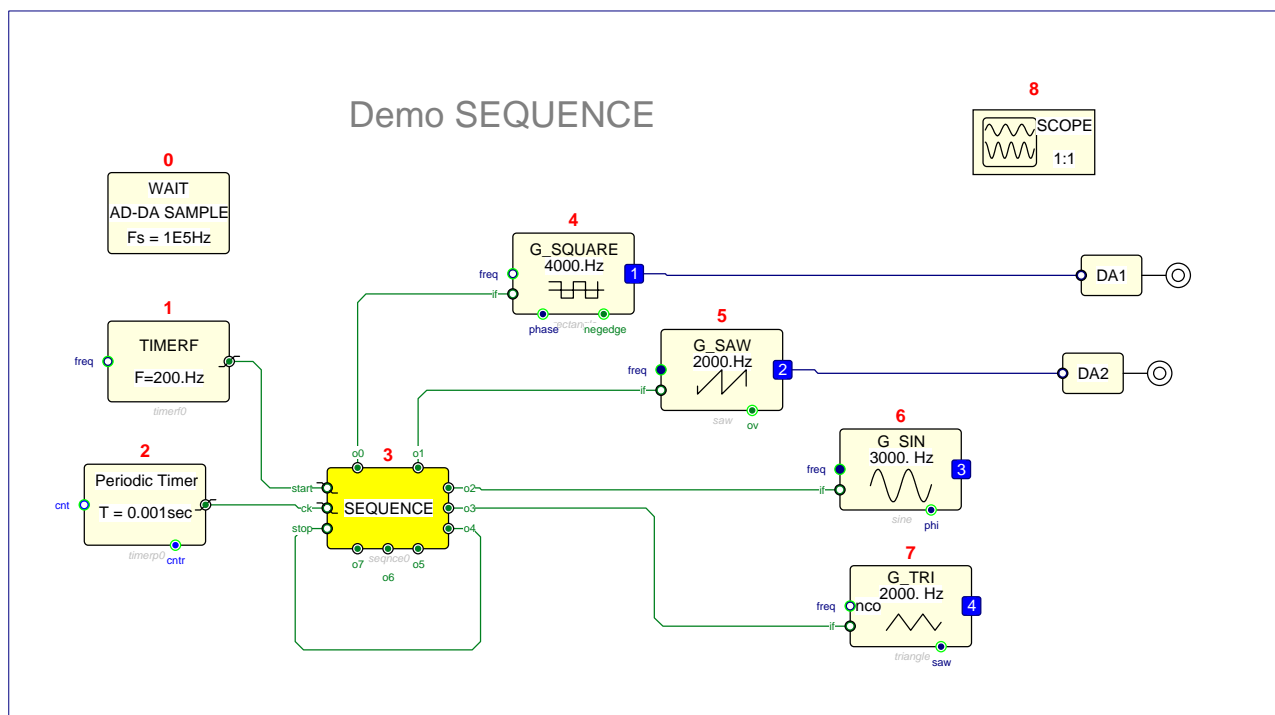
Sequence generator
On clock, next output becomes active
Pulse= active for 1 sample
State= active until next clock
If start is true, then next active output is o0
Stop inhibits clock. Start negates Stop.

INPUTS

Name:	Data Type:	Data Struct:	Connection:
name_start	BOOL	BIT	mandatory
name_ck	BOOL	BIT	mandatory
name_stop	BOOL	BIT	mandatory

OUTPUTS

Name:	Data Type:	Data Struct:	Connection:
name_o0	BOOL	BIT	normal
name_o1	BOOL	BIT	normal
name_o2	BOOL	BIT	normal
name_o3	BOOL	BIT	normal
name_o4	BOOL	BIT	normal
name_o5	BOOL	BIT	normal
name_o6	BOOL	BIT	normal
name_o7	BOOL	BIT	normal

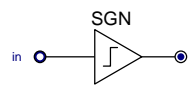


SEQUENCE test program

SGN

Sign function

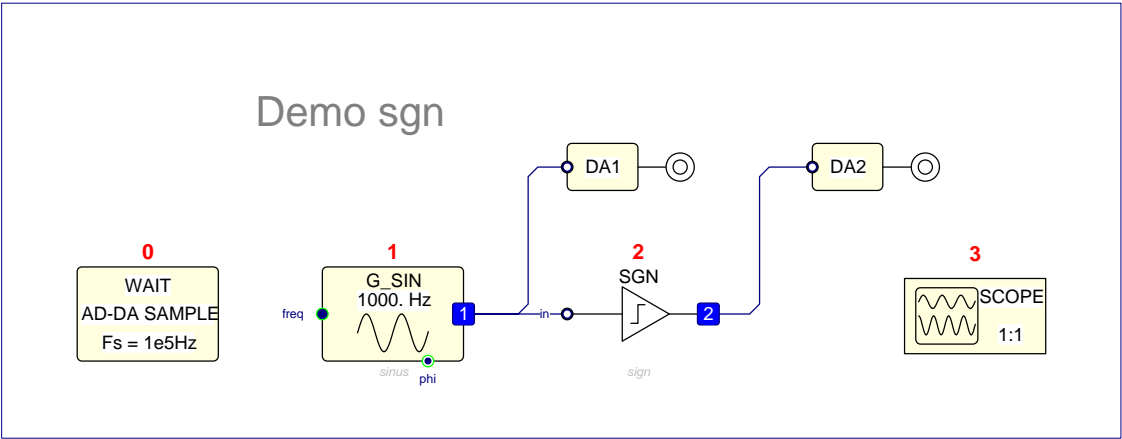
SGN



CATEGORY: NON LINEAR

DESCRIPTION:
Sign function
 $y=+1$ if $x \geq 0$; $y=-1$ if $x < 0$

INPUTS			
Name:			
name_in	Data Type: FRACT	Data Struct: WORD	Connection: mandatory
OUTPUTS			
Name:			
name	Data Type: FRACT	Data Struct: WORD	Connection: normal

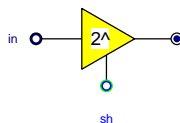


SGN test program

SHIFT

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:

N

Default values:

1

INPUTS

Name:

name_in
name_sh

Data Type:

FRACT
INTEGER

Data Struct:

WORD
WORD

Connection:

mandatory
optional

OUTPUTS

Name:

name

Data Type:

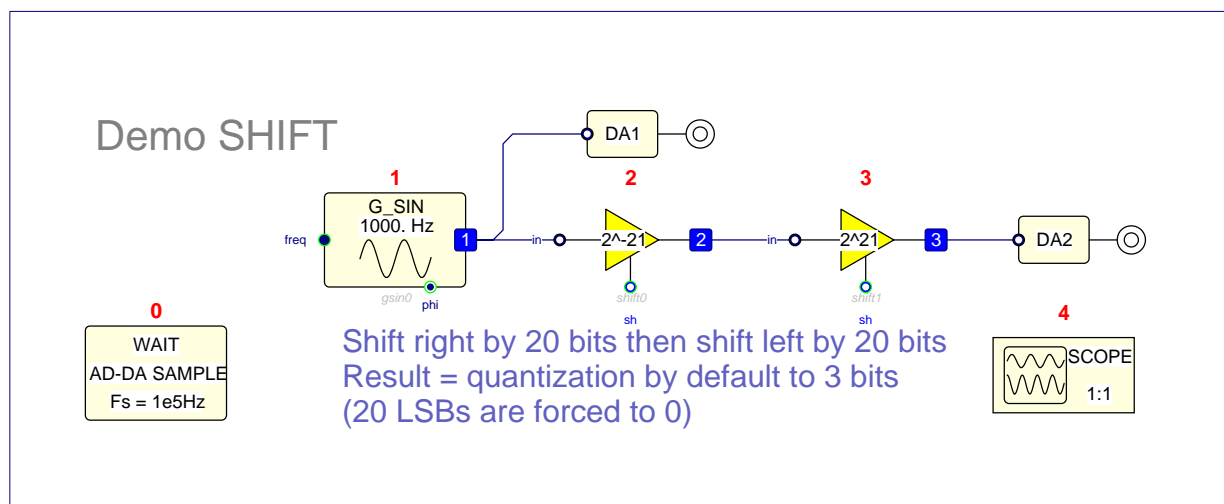
FRACT

Data Struct:

WORD

Connection:

normal

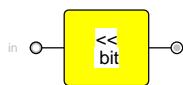


SHIFT test program

SHIFTV

Logic Shift

SHIFTV

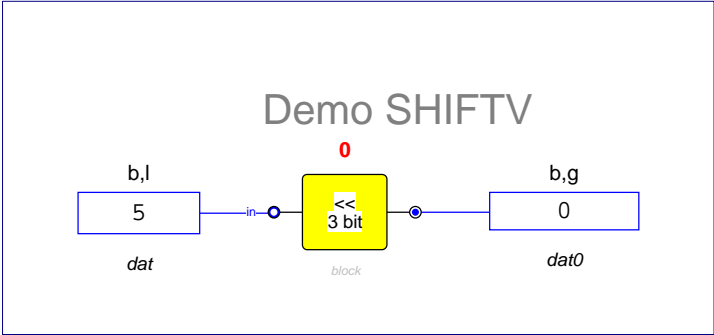


DESCRIPTION:
Logic Shift
if N>0 then left logic shift by N
else right logic shift by N

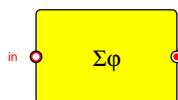
PARAMETERS:
Parameter:
Bits to Left
Default values:
1

INPUTS			
<i>Name:</i> name_in	<i>Data Type:</i> defined by cn	<i>Data Struct:</i>	<i>Connection:</i> mandatory

OUTPUTS			
<i>Name:</i> name	<i>Data Type:</i> defined by cn	<i>Data Struct:</i>	<i>Connection:</i> normal



SHIFTV test program



CATEGORY: TELECOM

DESCRIPTION:
Phase accumulator

INPUTS

Name:
name_in

Data Type:
COMPLEX

Data Struct:
WORD

Connection:
mandatory

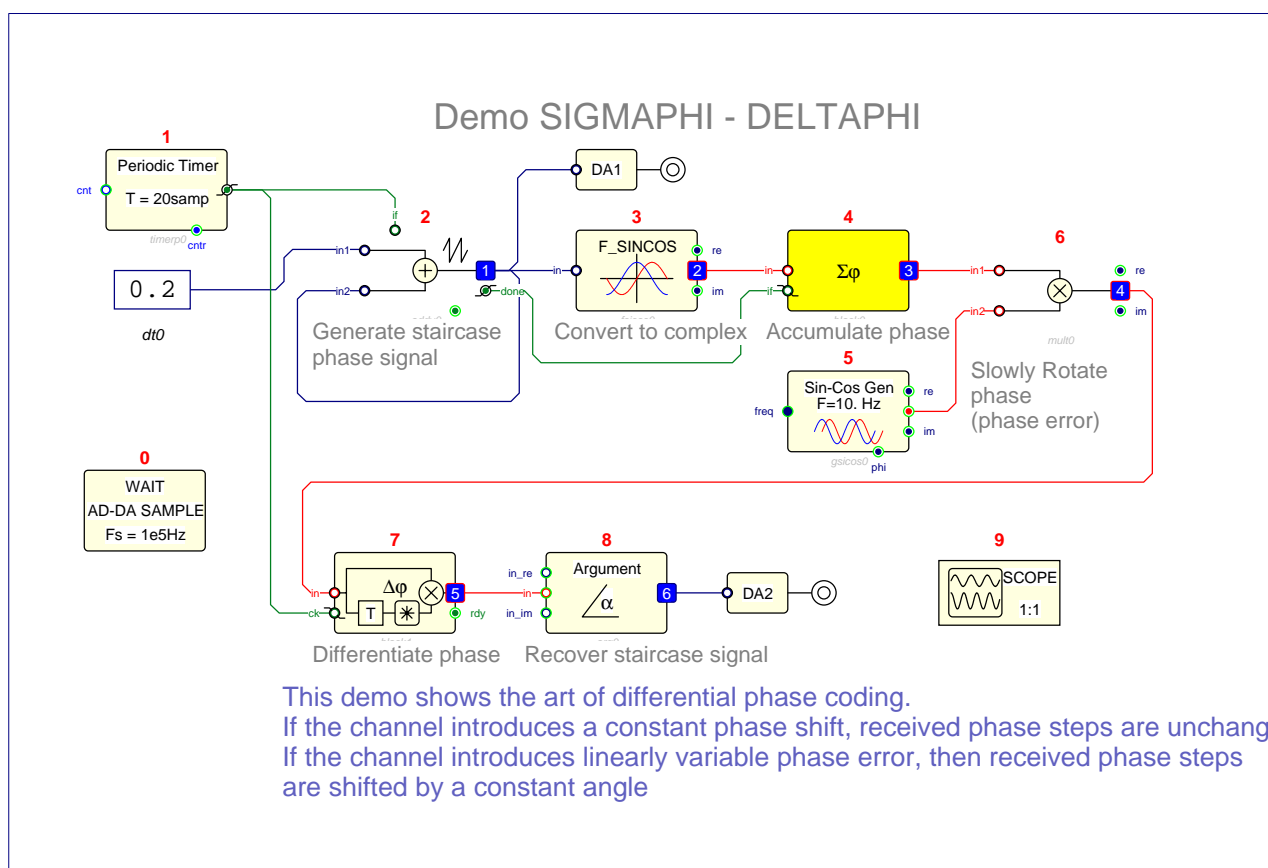
OUTPUTS

Name:
name

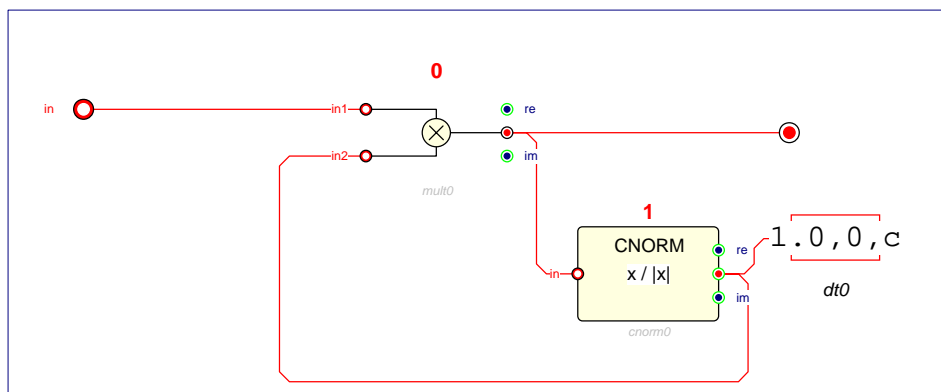
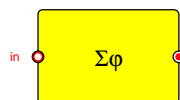
Data Type:
COMPLEX

Data Struct:
WORD

Connection:
normal



SIGMAPHI test program

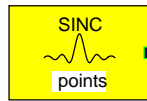


SIGMAPHI internal schema

SINC

Sinc Table

SINC



CATEGORY: TABLES

DESCRIPTION:

Sinc Table

Generates a Sinc table or a half Sinc table of Size points

$y(n) = w(n) \cdot \text{Sin}(k \cdot \pi \cdot (n - n_0)) / (k \cdot \pi \cdot (n - n_0))$ $n = 0 \dots \text{size} - 1$ Full: $n_0 = (\text{size} - 1) / 2$ Half: $n_0 = 0$

When calibration is off, Sinc maximum is 1.0, else $\text{Sum}(\text{full table})$ of $y(n) = \text{calibration value}$

When window = rectangle, $w(n)$ is a constant, else $w(n)$ is a Blackman-Harris weighting function

PARAMETERS:

Parameter:

Full or Half

Size

K factor

Window

Calibration

Default values:

full, half

100

0.1

rectangle, blackman-harris

off, 1.0

OUTPUTS

Name:

name

Data Type:

FRACT

Data Struct:

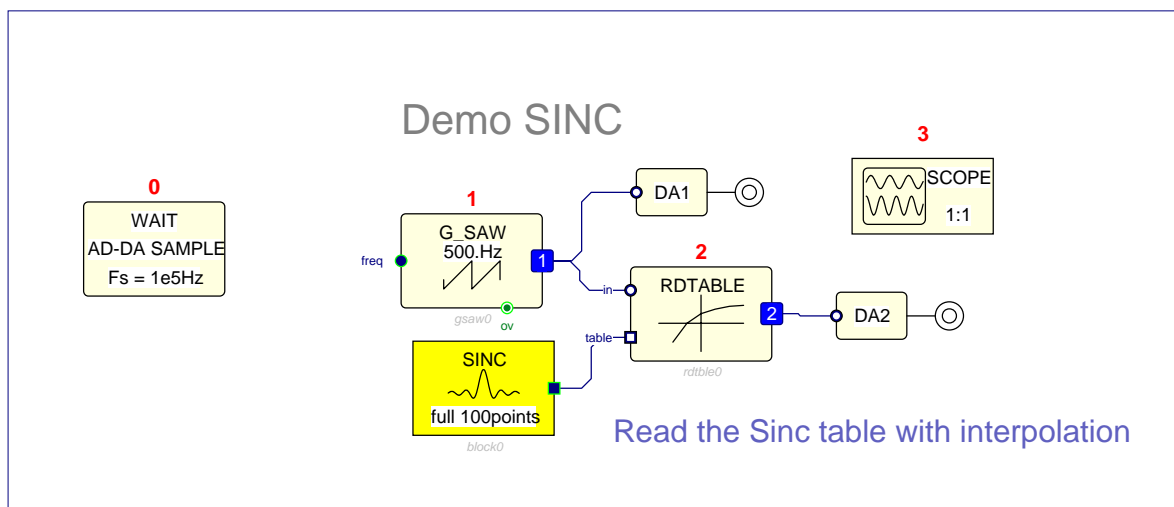
Matrix of WORD

Connection:

optional

ATTRIBUTES

Non executable, Data Table

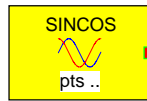


SINC test program

SINCOS

Sine-cosine Table

SINCOS



CATEGORY: TABLES

DESCRIPTION:

Sine-cosine Table

Generates an N point sampled Sine-Cosine table starting at phase phi0 and ending at phi1

Sine-Cosine pairs are considered as Imag-Real parts of a complex table stored in L:

$y_{re}(n) = \cos(\phi_0 + n \cdot (\phi_1 - \phi_0) / (N-1))$

$y_{im}(n) = \sin(\phi_0 + n \cdot (\phi_1 - \phi_0) / (N-1))$ when phases are expressed in radians

PARAMETERS:

Parameter:

Size

Start phase

End phase

Unit

Alignment

Default values:

100

0

360

deg,rad,half_turns

linear,module,fft

OUTPUTS

Name:

name

Data Type:

COMPLEX

Data Struct:

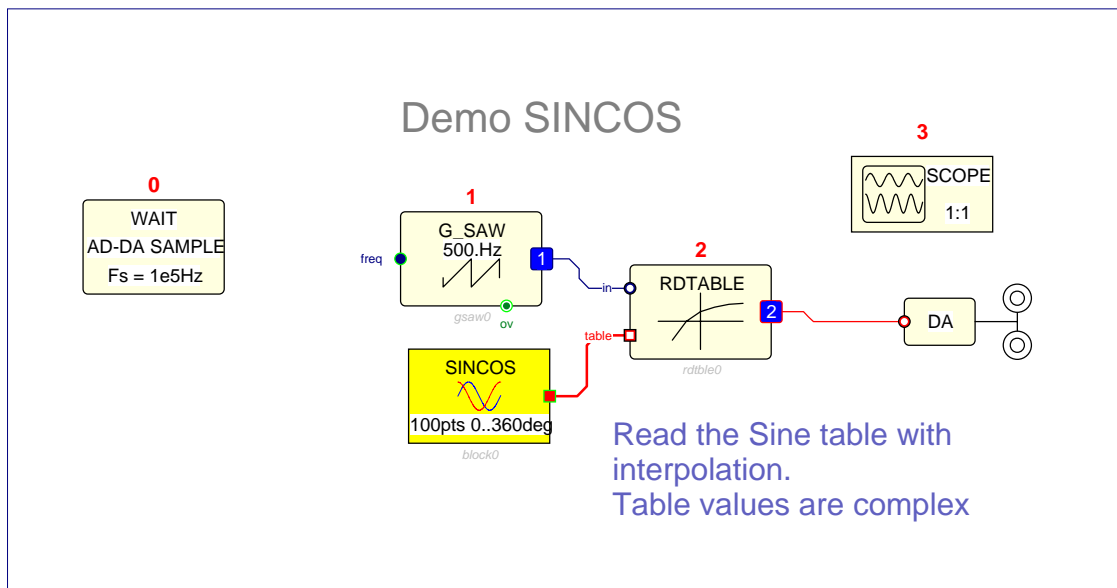
Matrix of DWORD

Connection:

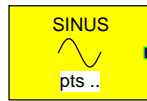
optional

ATTRIBUTES

Non executable, Data Table



SINCOS test program



CATEGORY: TABLES

DESCRIPTION:

Sine Table

Generates an N point sampled Sine table starting at phase phi0 and ending at phi1
 $y(n) = \sin(\phi_0 + n \cdot (\phi_1 - \phi_0) / (N - 1))$ when phases are expressed in radians

PARAMETERS:

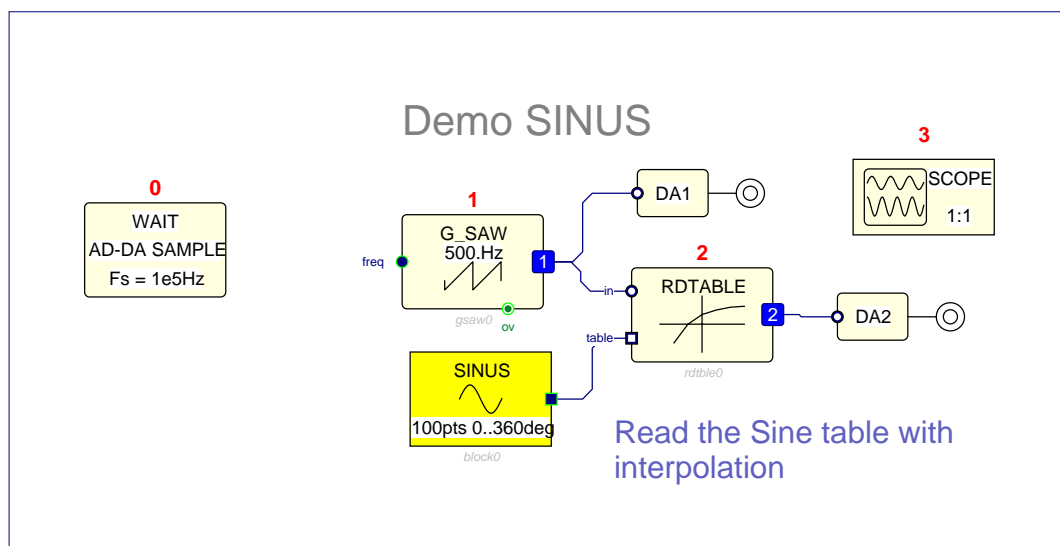
Parameter:	Default values:
Size	100
Start phase	0
End phase	360
Unit	deg, rad, half_turns

OUTPUTS

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

ATTRIBUTES

Non executable, Data Table

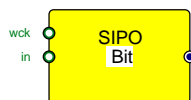


SINUS test program

SIPO

Serial In Parallel Out

SIPO



CATEGORY: CONTROL

DESCRIPTION:

Serial In Parallel Out
Serial to Parallel Shift Register function

PARAMETERS:

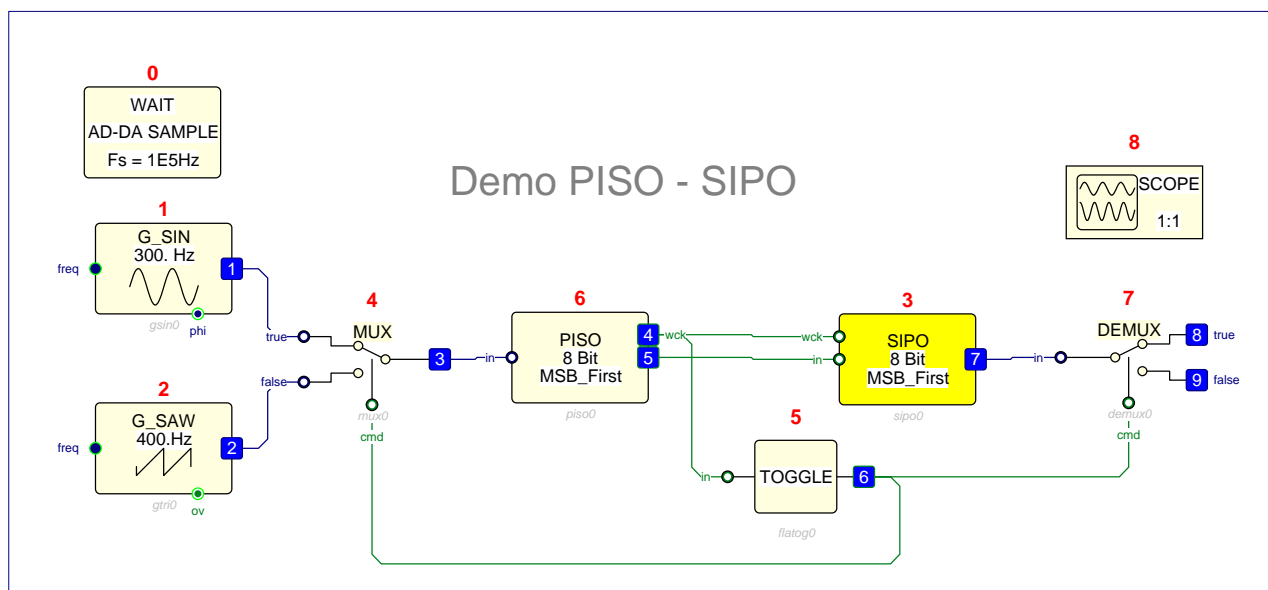
<i>Parameter:</i>	<i>Default values:</i>
Word Length	8,12,16
Shift Direction	MSB_First,LSB_First

INPUTS

<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name_in	BOOL	BIT	mandatory
name_wck	BOOL	BIT	mandatory

OUTPUTS

<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name	FRACT	WORD	normal



SIPO test program

SLOPELIM

Slope limiting filter

SLOPELIM



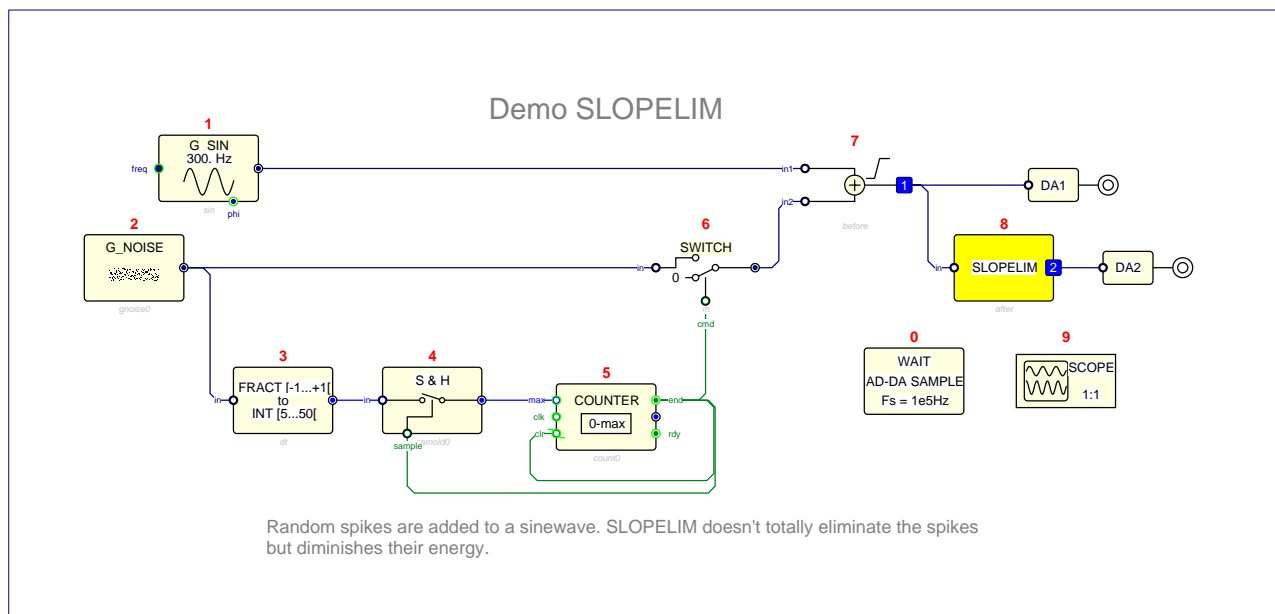
CATEGORY: FILTERS

DESCRIPTION:
Slope limiting filter
for EMI spikes suppression.

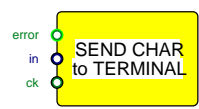
PARAMETERS:
Parameter: Default values:
slope 0.01
slopeneg 0.01

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



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:
name_in
name_ck
name_error

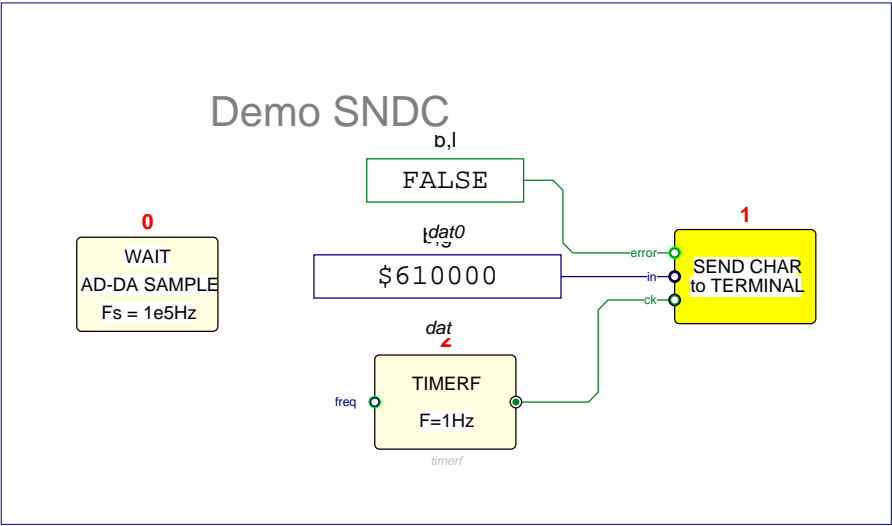
Data Type:
FRACT
BOOL
BOOL

Data Struct:
WORD
BIT
BIT

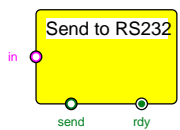
Connection:
mandatory
mandatory
optional

ATTRIBUTES

Unique,



SNDC test program



CATEGORY: CONTROL

DESCRIPTION:
Send string to RS232 port

PARAMETERS:
Parameter:
String name

Default values:
s0

INPUTS
Name:
name_in
name_send

Data Type:
STRING
BOOL

Data Struct:
WORD
BIT

Connection:
mandatory
mandatory

OUTPUTS
Name:
name_rdy

Data Type:
BOOL

Data Struct:
BIT

Connection:
normal



CATEGORY: INSTRUMENTS

DESCRIPTION:

Spectrum Analyser

Real or complex input.

Modes Amp, Power and DB. Max FFT points: 1024 (full display) or 2048 (half display)

Outputs a periodic scan of spectrum with S/W sync

Specan runs in main loop. Sampled application must run inside an interrupt.

if Decimate > 1 then input signal is subsampled every n samples, and scope frequency scale adapted

PARAMETERS:

Parameter:

FFT Size

Display

Mode

Window

Decimate by

Default values:

1024,512,256,128,64,32,16,8

full,half

dB,Amp,Pow

Rectangle,Triangle,Hann,Hamming,Nuttall,Gauss,Blackman_Harris,Flat_Top

1

INPUTS

Name:

name_in

Data Type:

defined by cn

Data Struct:

Connection:

mandatory

OUTPUTS

Name:

name

Data Type:

FRACT

Data Struct:

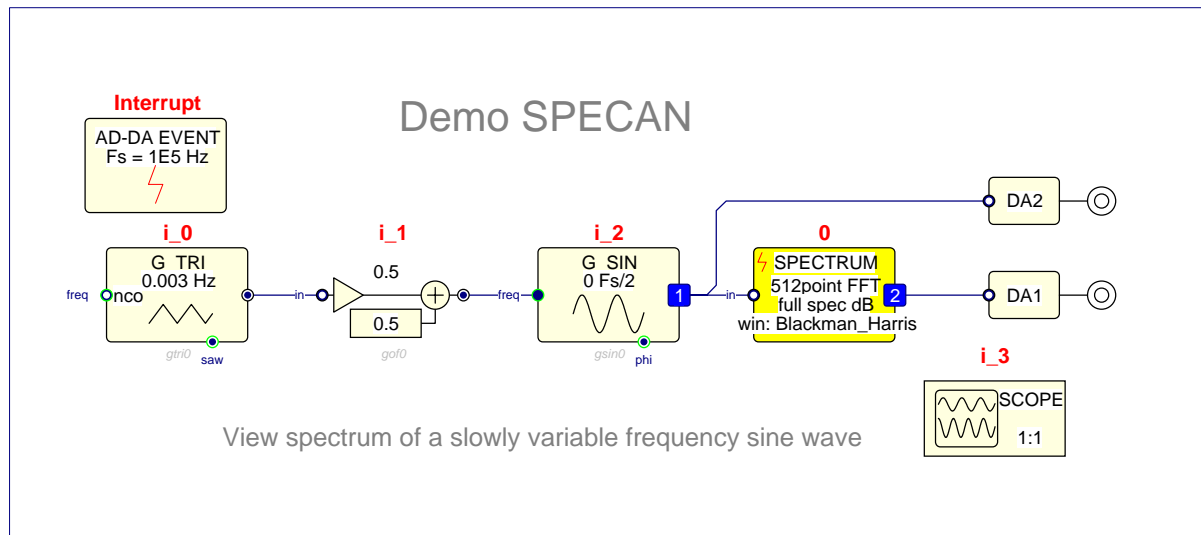
WORD

Connection:

normal

ATTRIBUTES

Unique,

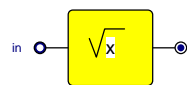


SPECAN test program

SQROOT

Square root of input

SQROOT



CATEGORY: FUNCTIONS

DESCRIPTION:
Square root of input

INPUTS

Name:
name_in

Data Type:
FRACT

Data Struct:
WORD

Connection:
mandatory

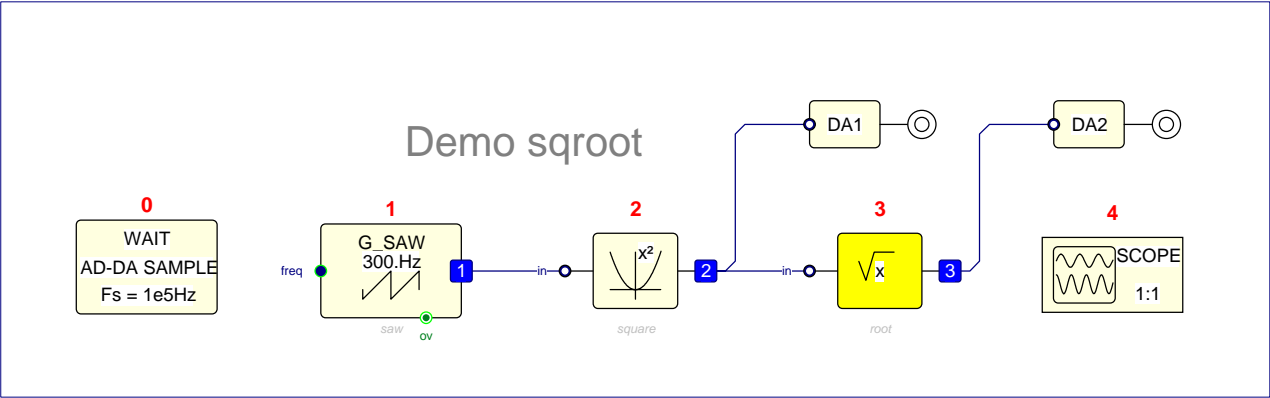
OUTPUTS

Name:
name

Data Type:
FRACT

Data Struct:
WORD

Connection:
normal

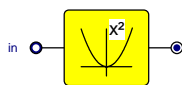


SQROOT test program

SQUARE

Square of input

SQUARE



CATEGORY: FUNCTIONS

DESCRIPTION:
Square of input

INPUTS

Name:
name_in

Data Type:
FRACT

Data Struct:
WORD

Connection:
mandatory

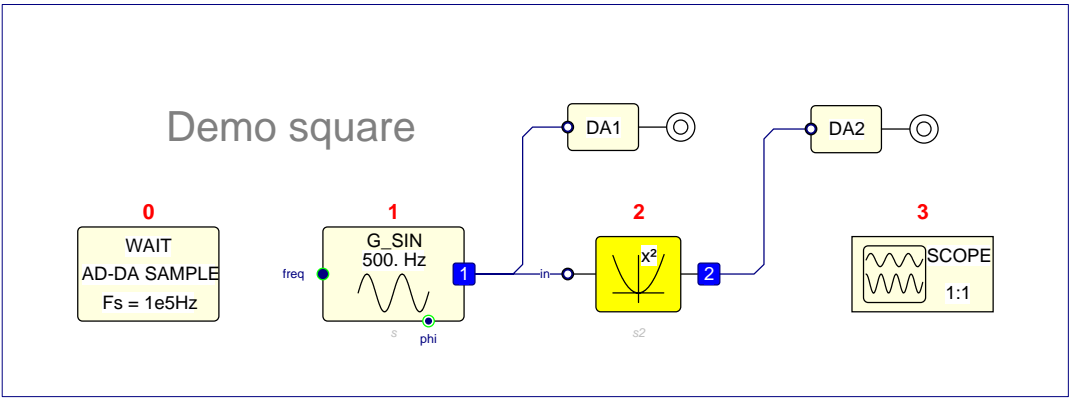
OUTPUTS

Name:
name

Data Type:
FRACT

Data Struct:
DWORD

Connection:
normal



SQUARE test program

STEREO_COD

STEREO_COD



INPUTS

Name:
name_G
name_D

Data Type:
FRACT
FRACT

Data Struct:
WORD
WORD

Connection:
mandatory
mandatory

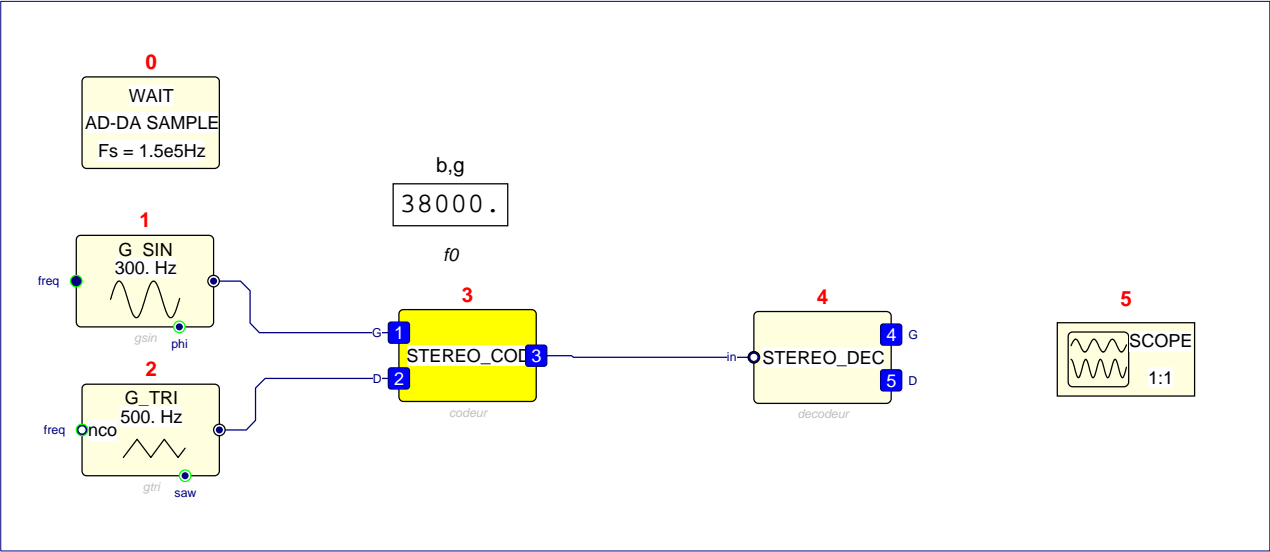
OUTPUTS

Name:
name

Data Type:
FRACT

Data Struct:
WORD

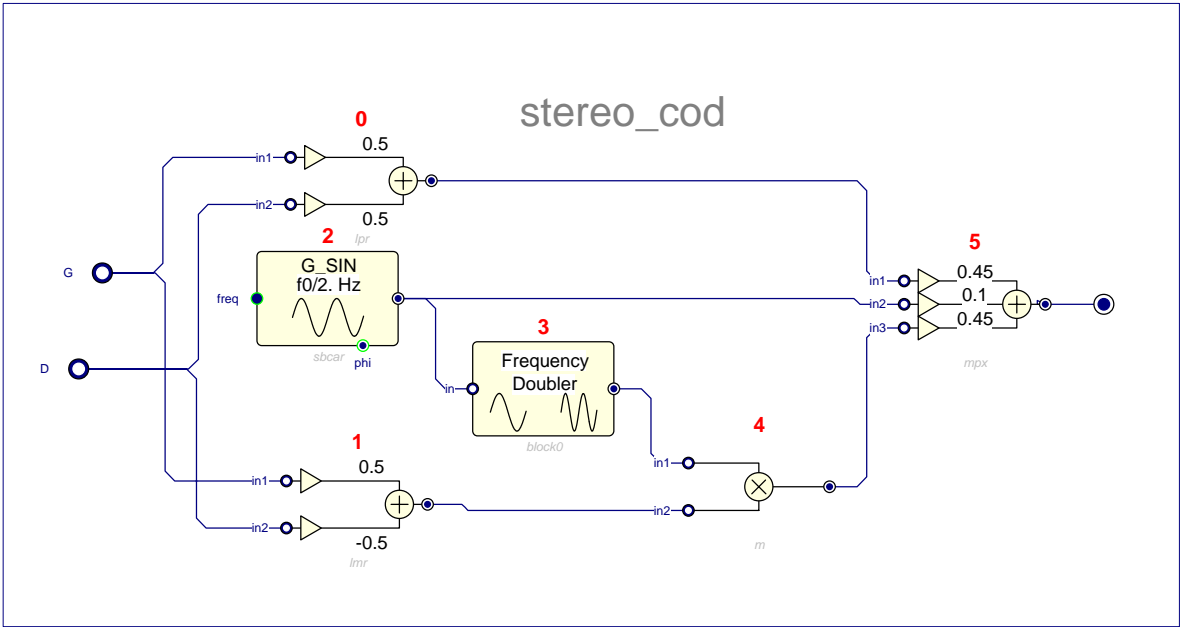
Connection:
normal



STEREO_COD test program

STEREO_COD

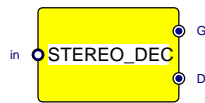
STEREO_COD



STEREO_COD internal schema

STEREO_DEC

STEREO_DEC



INPUTS

Name:
name_in

Data Type:
FRACT

Data Struct:
WORD

Connection:
mandatory

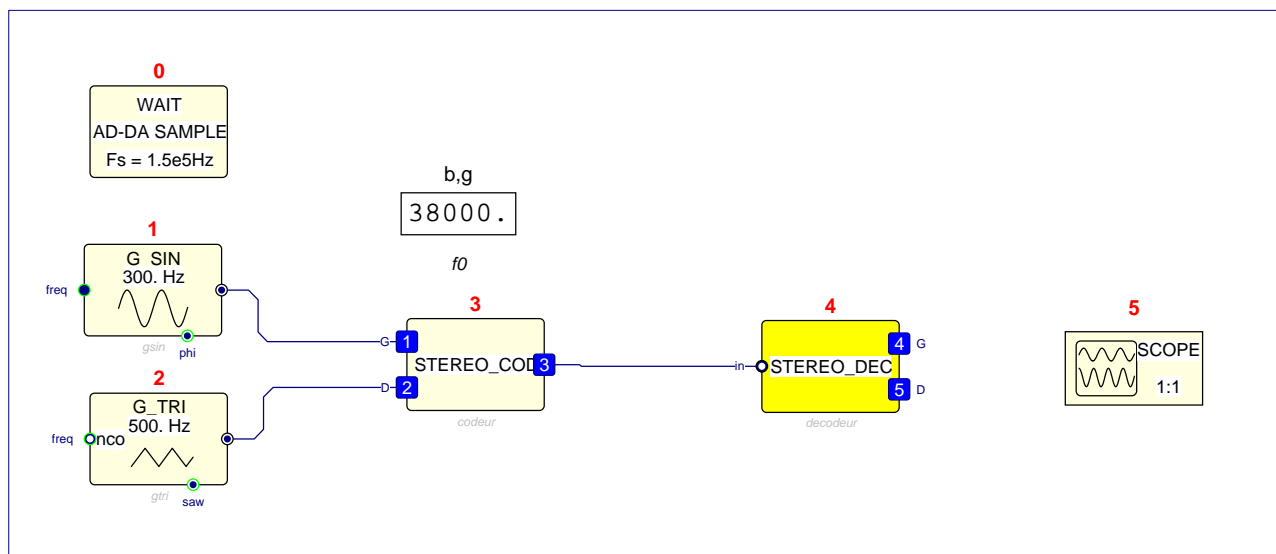
OUTPUTS

Name:
name_G
name_D

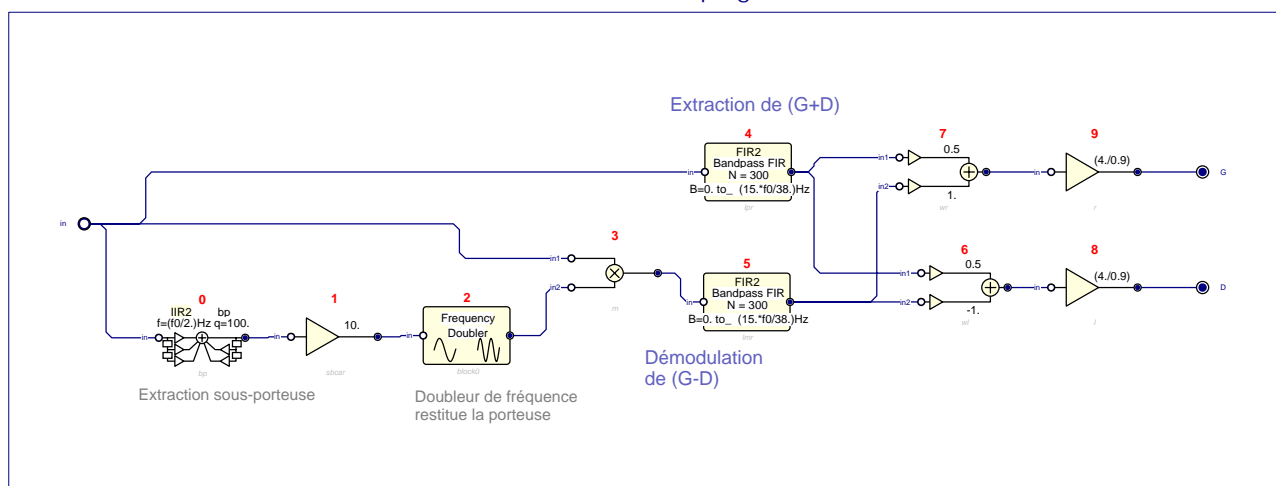
Data Type:
FRACT
FRACT

Data Struct:
WORD
WORD

Connection:
normal
normal



STEREO_DEC test program



STEREO_DEC internal schema

STOP

Stop processor

STOP

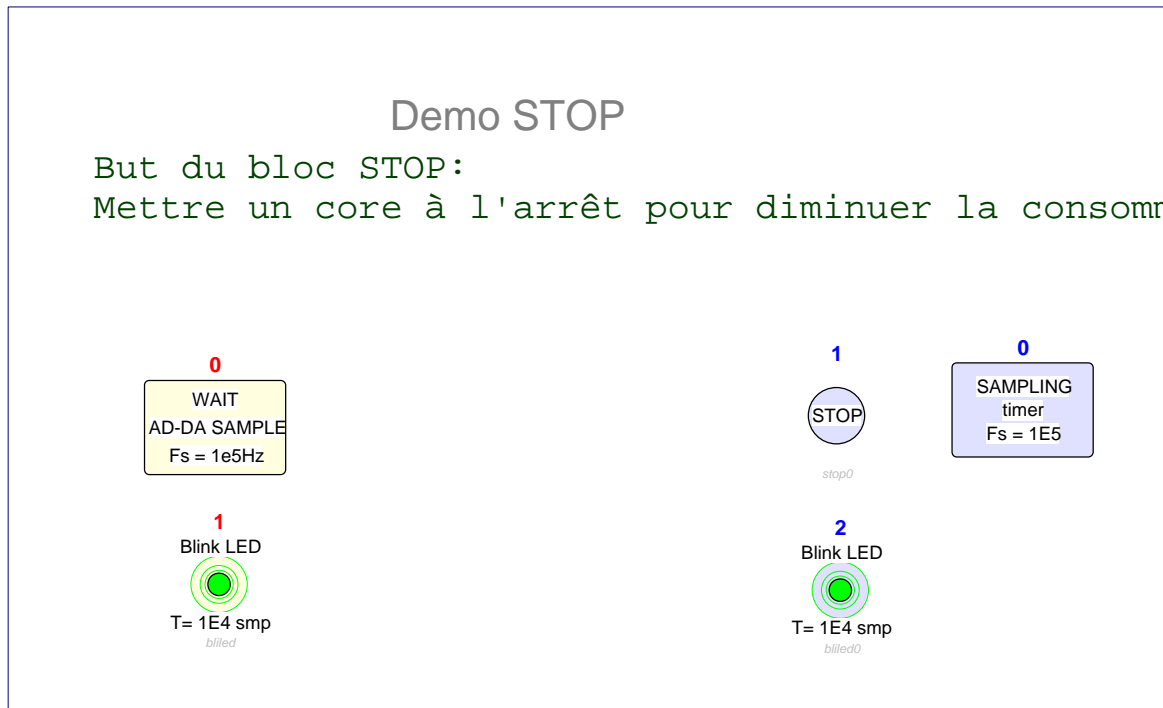


CATEGORY: CONTROL

DESCRIPTION:

Stop processor

Usage: inhibition of one core

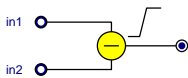


STOP test program

SUBS

Subtraction with saturation

SUBS



CATEGORY: ARITHMETIC

DESCRIPTION:
Subtraction with saturation
Output = in1 - in2

INPUTS

Name:
name_in1
name_in2

Data Type:
FRACT
FRACT

Data Struct:
WORD
WORD

Connection:
mandatory
mandatory

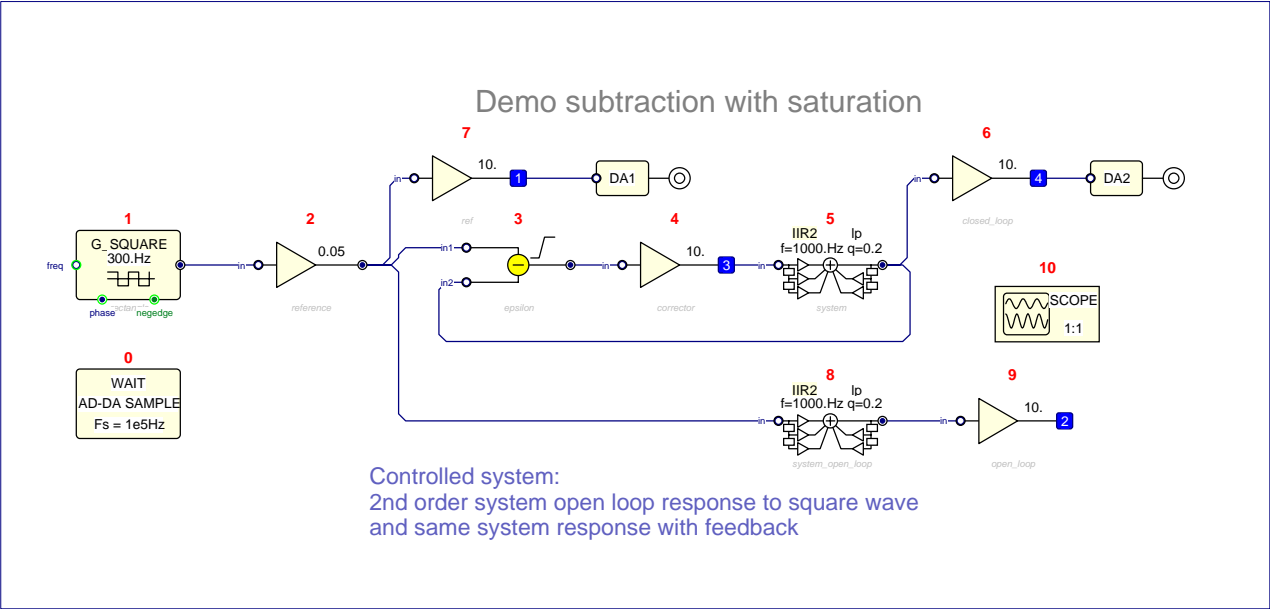
OUTPUTS

Name:
name

Data Type:
FRACT

Data Struct:
WORD

Connection:
normal

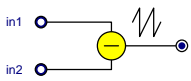


SUBS test program

SUBV

Subtraction modulo +/- 1

SUBV



CATEGORY: ARITHMETIC

DESCRIPTION:
Subtraction modulo +/- 1

INPUTS

Name:
name_in1
name_in2

Data Type:
FRACT
FRACT

Data Struct:
WORD
WORD

Connection:
mandatory
mandatory

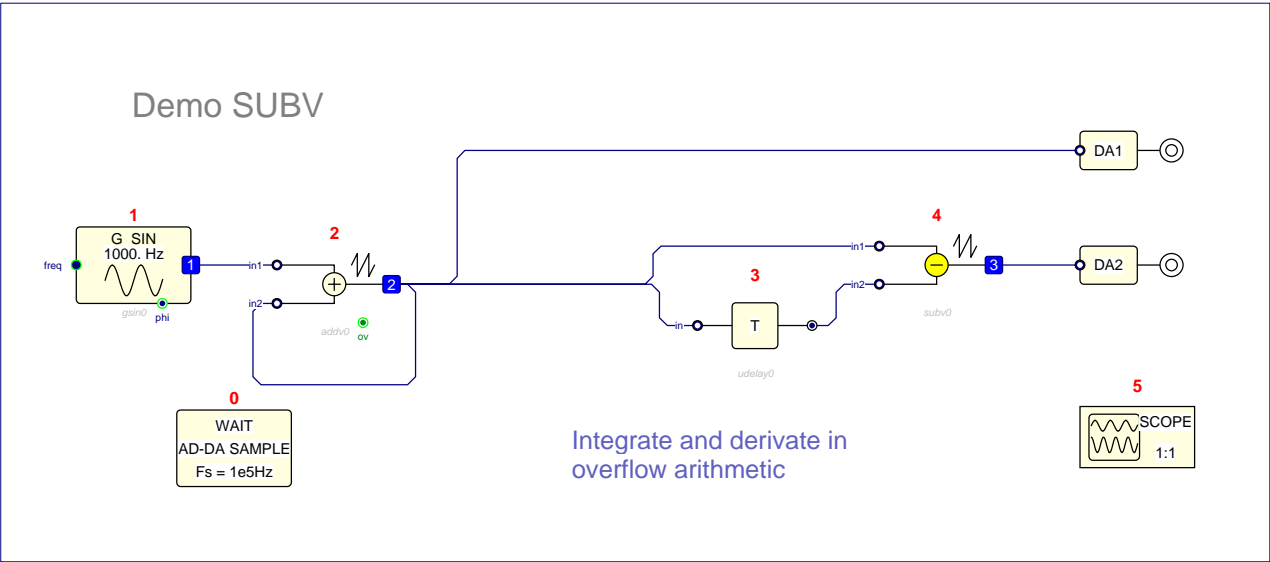
OUTPUTS

Name:
name

Data Type:
FRACT

Data Struct:
WORD

Connection:
normal

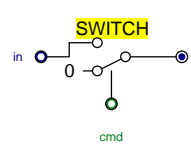


SUBV test program

SWITCH

Switch.

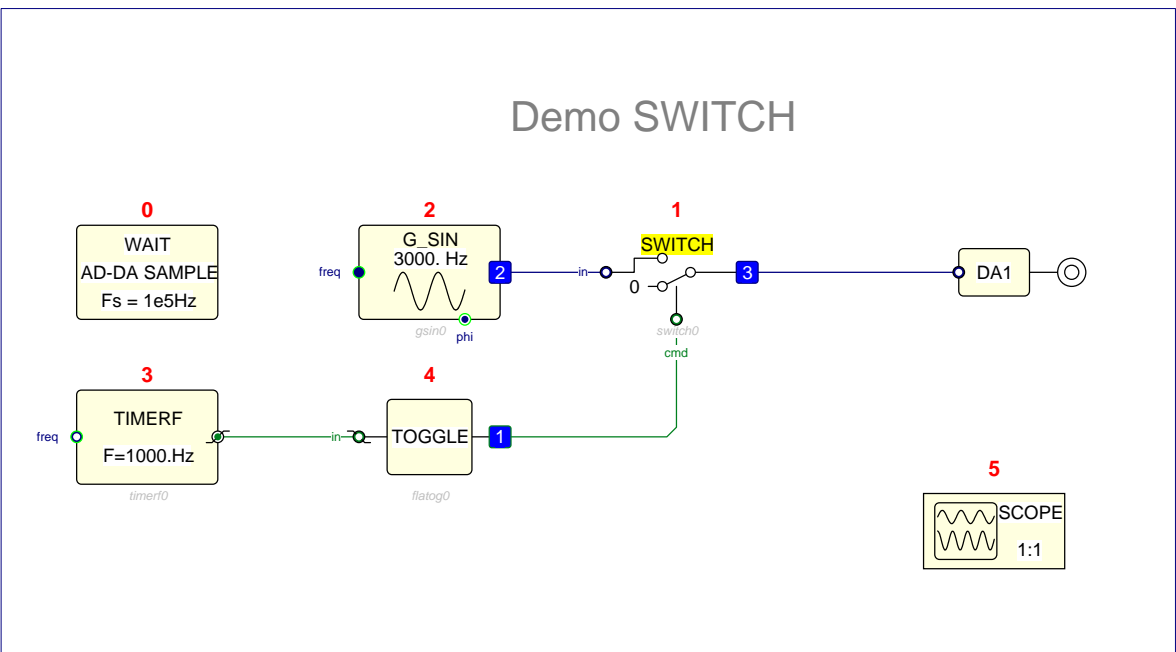
SWITCH



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

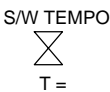


SWITCH test program

SWTEMPO

Time waisting tempo

SWTEMPO



CATEGORY: TIMING

DESCRIPTION:

Time waisting tempo
by s/w counting

PARAMETERS:

Parameter:

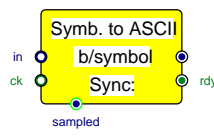
Time

Unit

Default values:

1

µs,ms,s



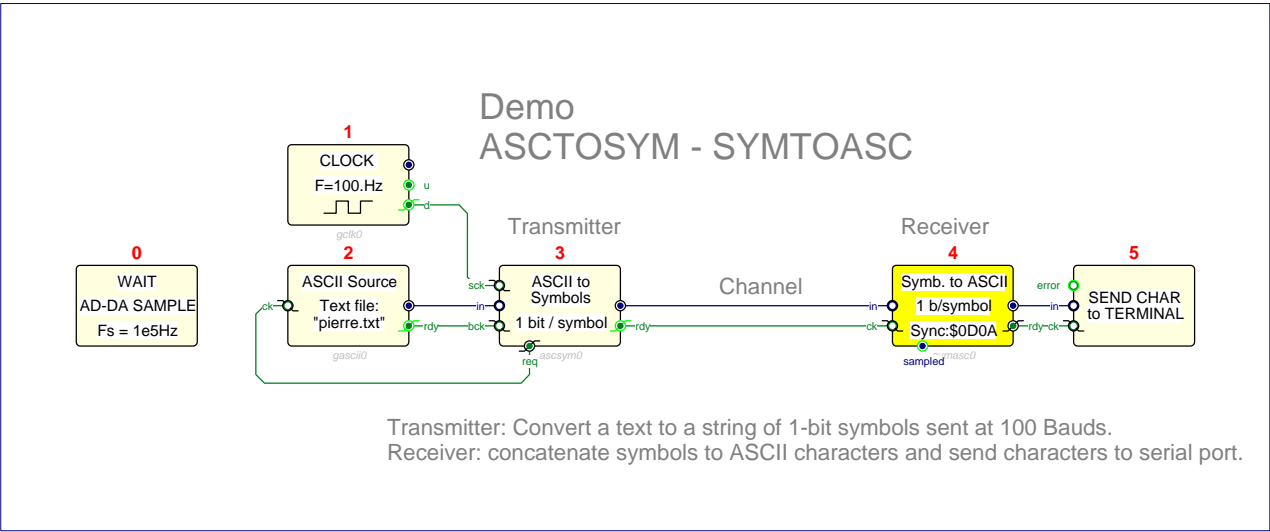
CATEGORY: TELECOM

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

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

INPUTS	Data Type:	Data Struct:	Connection:
<i>Name:</i> name_in	FRACT	WORD	mandatory
name_ck	BOOL	BIT	mandatory

OUTPUTS	Data Type:	Data Struct:	Connection:
<i>Name:</i> name_sampled	FRACT	WORD	optional
name	FRACT	WORD	normal
name_rdy	BOOL	BIT	normal



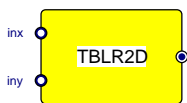
Transmitter: Convert a text to a string of 1-bit symbols sent at 100 Bauds.
Receiver: concatenate symbols to ASCII characters and send characters to serial port.

SYMTOASC test program

TBLR2D

2-D Table read and interpolate

TBLR2D



CATEGORY: FUNCTIONS

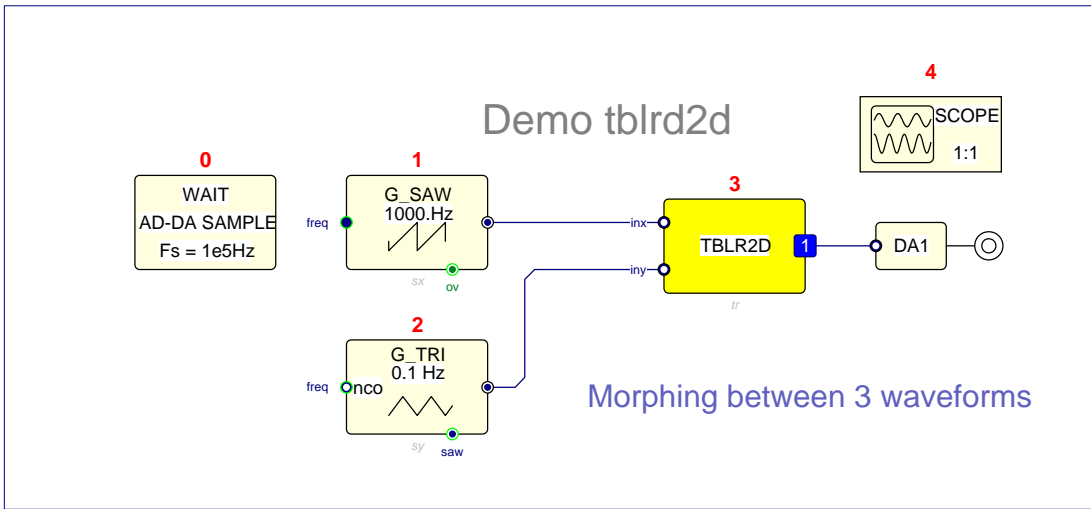
DESCRIPTION:
2-D Table read and interpolate

PARAMETERS:

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

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

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



TBLR2D test program



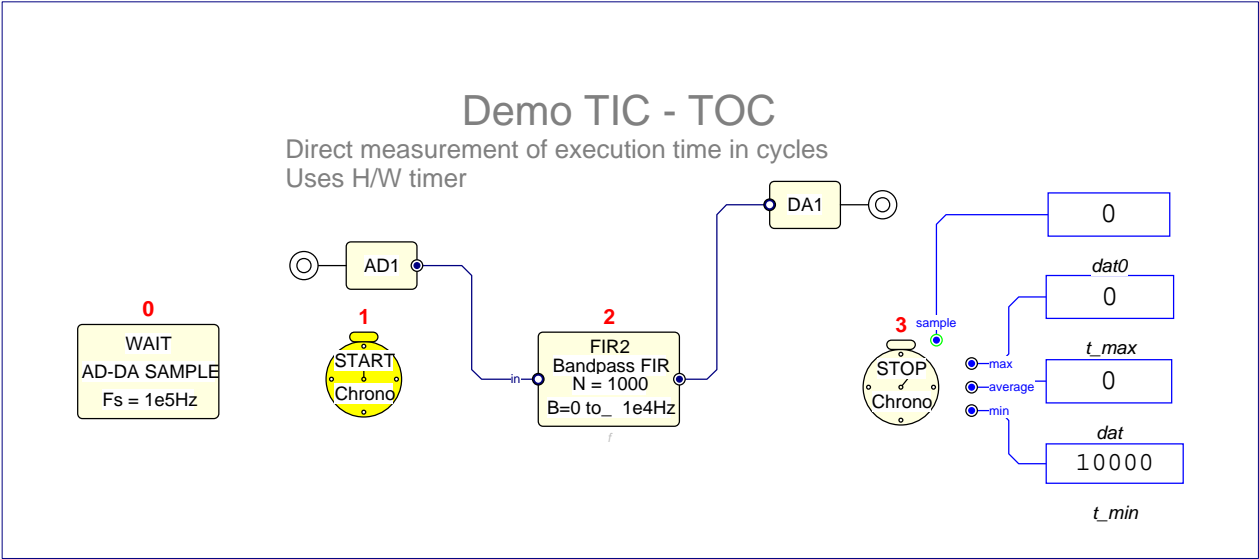
CATEGORY: TIMING

DESCRIPTION:
Start H/W Cycle Counter

PARAMETERS:
Parameter:
Timer #
Interrupts

Default values:
1,2,0
allowed,masked

ATTRIBUTES
Unique,

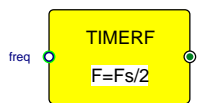


TIC test program

TIMERF

Frequency Timer

TIMERF



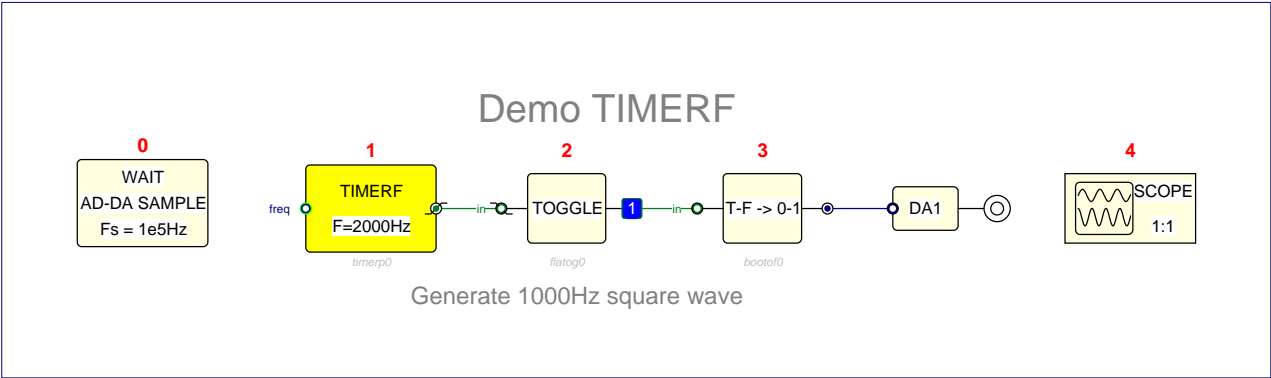
CATEGORY: TIMING

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

PARAMETERS:
Parameter: Frequency unit
Default values: 1000. Hz, $F_s/2$

INPUTS	Data Type:	Data Struct:	Connection:
<i>Name:</i> name_freq	FRACT	WORD	optional

OUTPUTS	Data Type:	Data Struct:	Connection:
<i>Name:</i> name	BOOL	BIT	normal

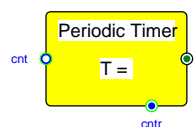


TIMERF test program

TIMERP

Periodic Timer

TIMERP



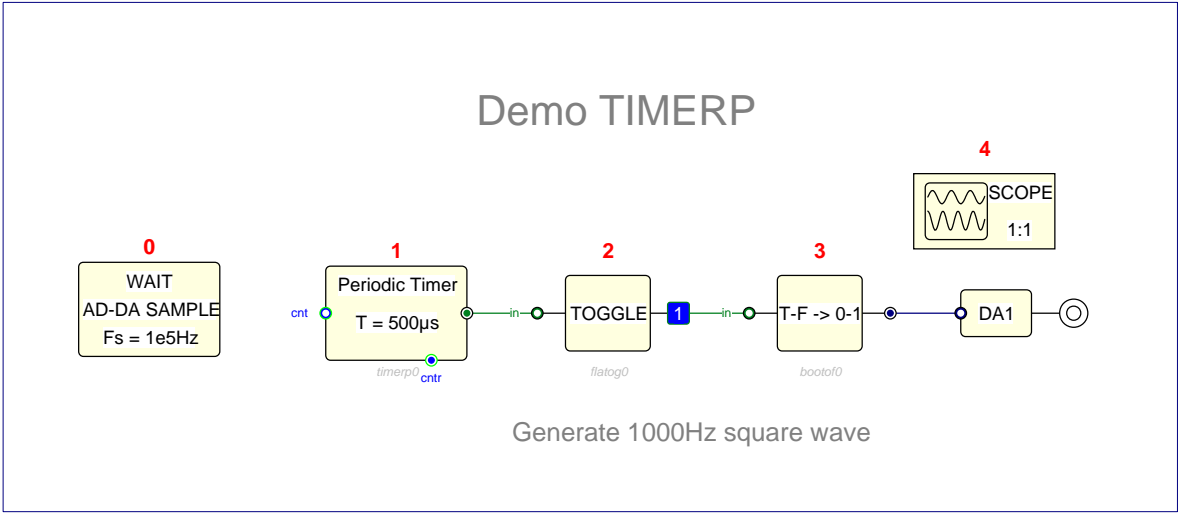
CATEGORY: TIMING

DESCRIPTION:
Periodic Timer
Sets a flag every N sampling periods

PARAMETERS:
Parameter:
Period
Unit
Default values:
0.01
smp,s,ms,μs

INPUTS			
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name_cnt	INTEGER	WORD	optional

OUTPUTS			
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name_cntr	BOOL INTEGER	BIT WORD	normal optional

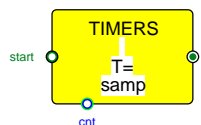


TIMERP test program

TIMERS

One shoot Timer.

TIMERS

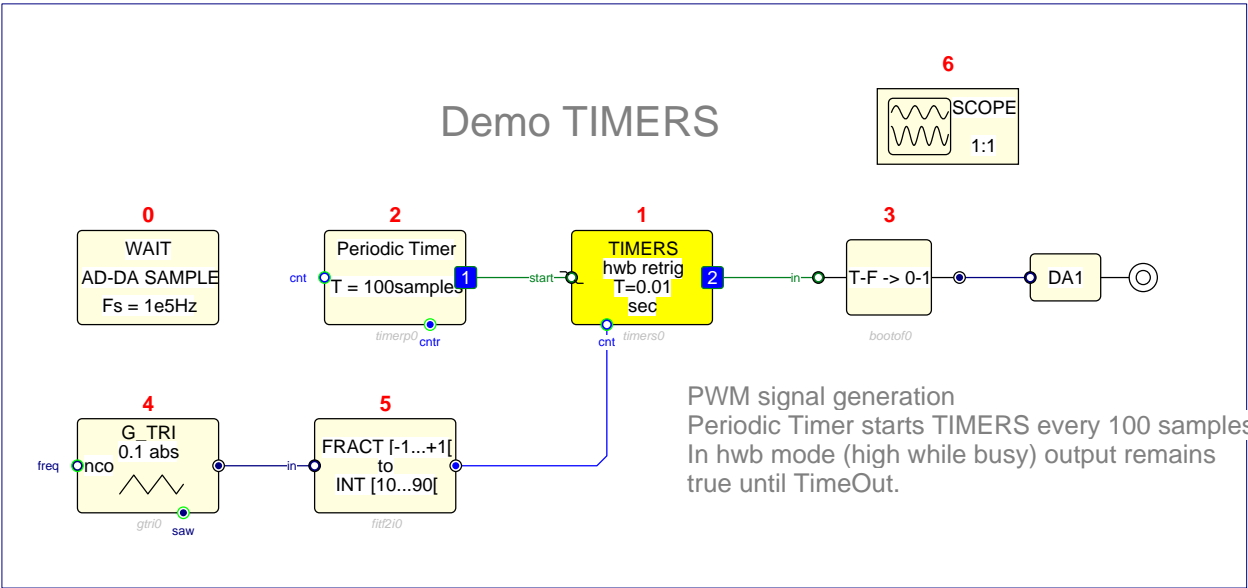


CATEGORY: TIMING

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:
High busy or high at TO ?
Retriggerable ?
Time
Unit
Default values:
hwb,hto
retrig,non_retrig
0.01
smp,s,ms,µs

INPUTS			
<i>Name:</i> name_start name_cnt	<i>Data Type:</i> BOOL INTEGER	<i>Data Struct:</i> BIT WORD	<i>Connection:</i> mandatory optional
<i>OUTPUTS</i> <i>Name:</i> name	<i>Data Type:</i> BOOL	<i>Data Struct:</i> BIT	<i>Connection:</i> normal



TIMERS test program



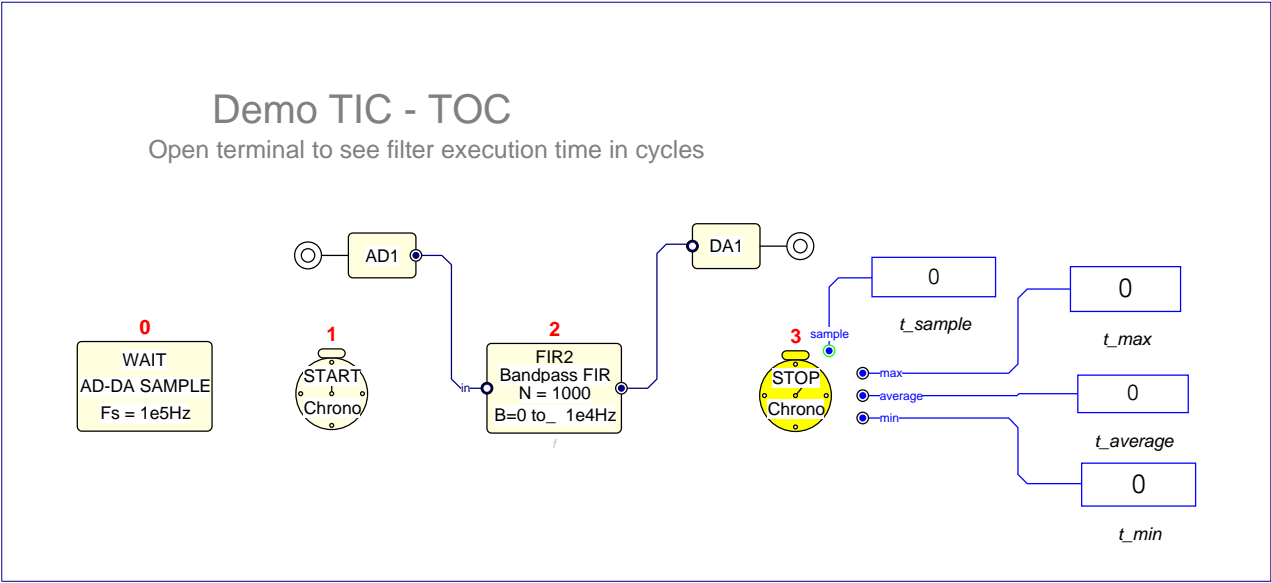
CATEGORY: TIMING

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
In single mode, chrono is inhibited after occurences measurements, and Min and Max are displayed
In refreshed mode, Min and Max are refreshed after each occurences measurements

PARAMETERS:	
<i>Parameter:</i>	<i>Default values:</i>
Timer #	1,2,0
Occurences	10
Single or Refreshed	single,refreshed

OUTPUTS			
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name_max	INTEGER	WORD	normal
name_average	INTEGER	WORD	normal
name_min	INTEGER	WORD	normal
name_sample	INTEGER	WORD	optional

ATTRIBUTES
Unique,



TOC test program

TOLLITE_HOSTIAS

MIDI File

TOLLITE_HOSTIAS



CATEGORY: MUSIC

DESCRIPTION:

MIDI File
Transcribed in asm format

OUTPUTS

<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name	INTEGER	Matrix of WORD	normal

ATTRIBUTES

Non executable, Unique, Data Table



CATEGORY: MUSIC

DESCRIPTION:
MIDI File
Transcribed in asm format

OUTPUTS

<i>Name:</i> name	<i>Data Type:</i> INTEGER	<i>Data Struct:</i> Matrix of WORD	<i>Connection:</i> normal
----------------------	------------------------------	---------------------------------------	------------------------------

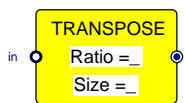
ATTRIBUTES

Non executable, Unique, Data Table

TRANPOSE

Transpose

TRANPOSE



CATEGORY: AUDIO

DESCRIPTION:

Transpose
Change pitch of melody or voice by multiplying frequencies by given ratio
 $f_i(\text{out}) = f_i(\text{in}) * k$
Size = rotating buffer size (samples)

PARAMETERS:

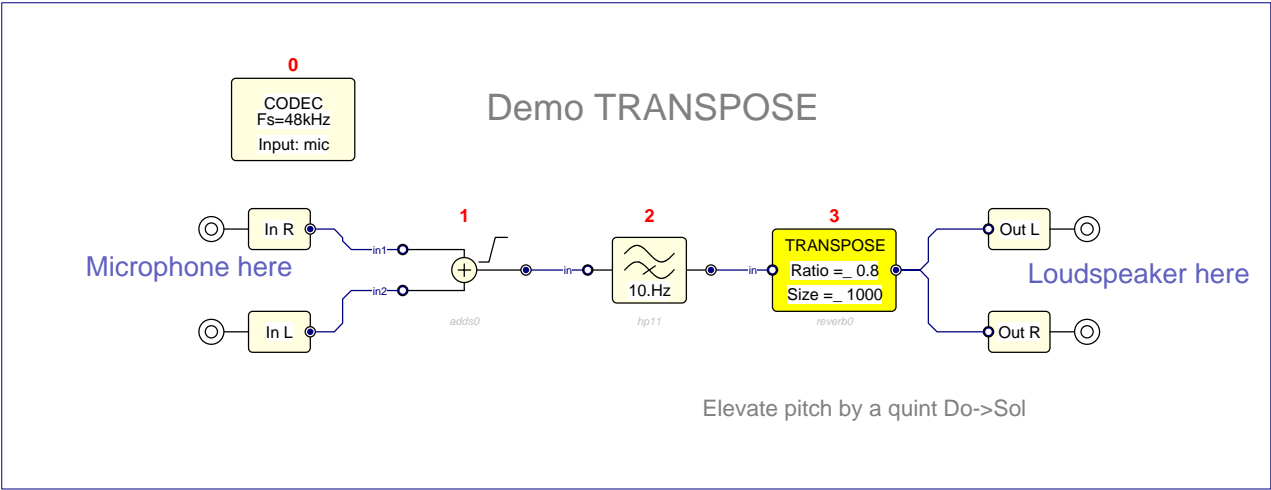
Parameter:	Default values:
Size	1000
Ratio	1.5

INPUTS

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

OUTPUTS

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



TRANPOSE test program

TRAP

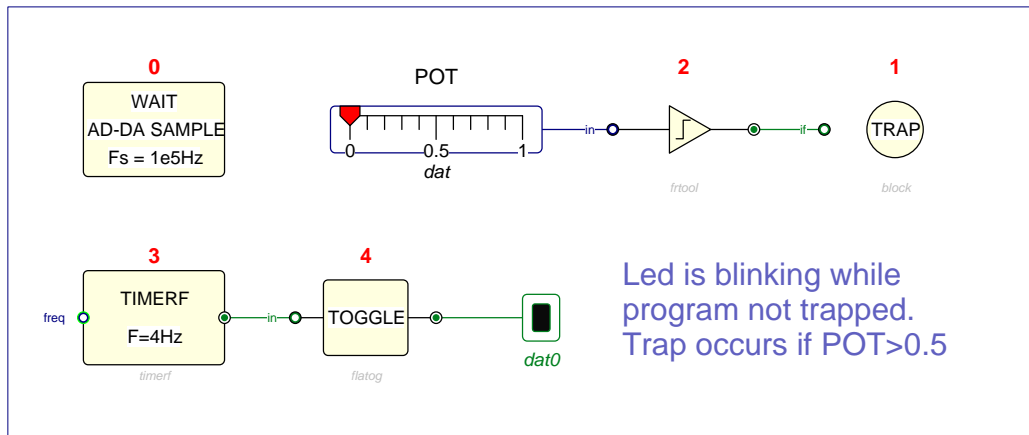
Hang here (infinite loop)

TRAP



CATEGORY: CONTROL

DESCRIPTION:
Hang here (infinite loop)

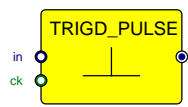


TRAP test program

TRIGD_PULSE

Triggered pulse

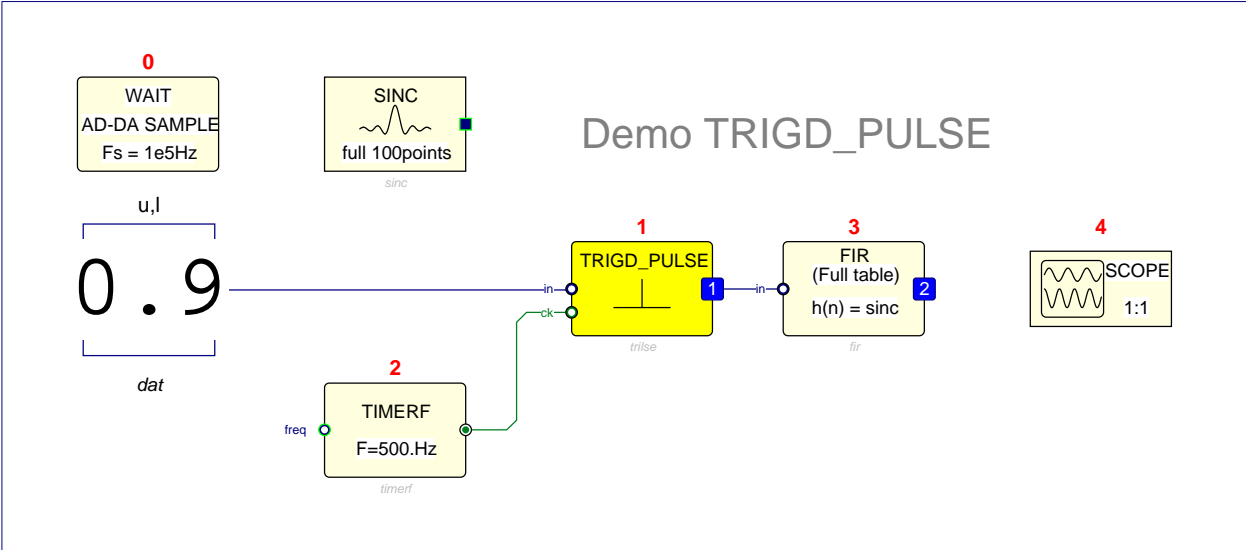
TRIGD_PULSE



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

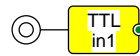


TRIGD_PULSE test program

TTL_IN1

Digital Input 1

TTL_IN1



CATEGORY: ETD410K

DESCRIPTION:

Digital Input 1

Bool Output reflects state of IRQC pin

OUTPUTS

Name:

name

Data Type:

BOOL

Data Struct:

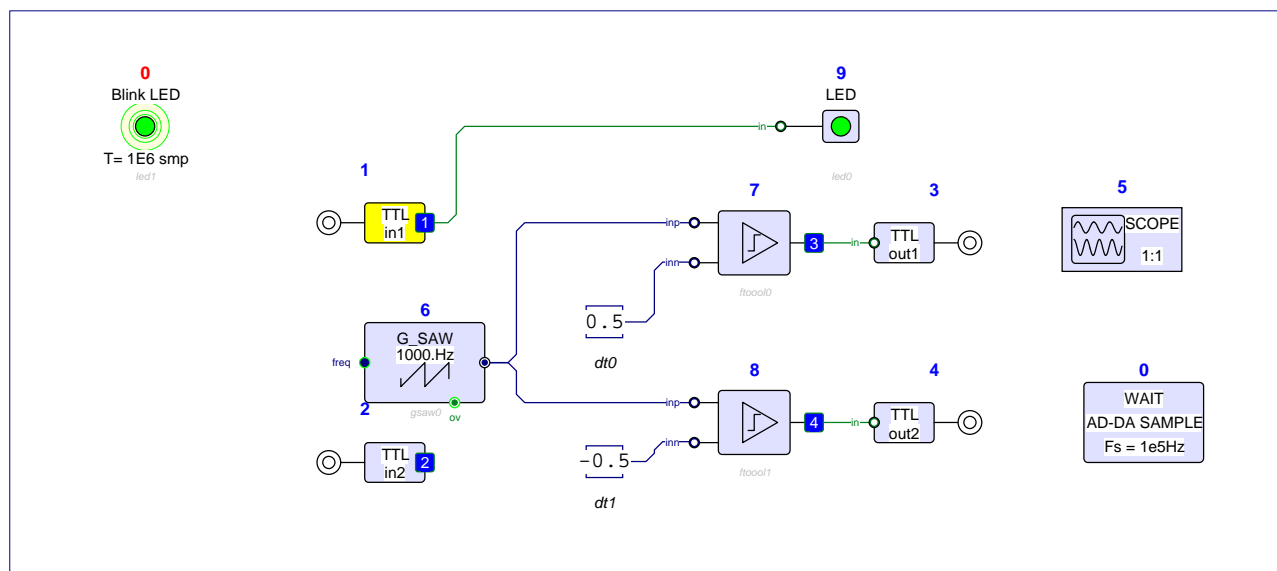
BIT

Connection:

normal

ATTRIBUTES

Unique,



TTL_IN1 test program

TTL_IN2

Digital Input 2

TTL_IN2

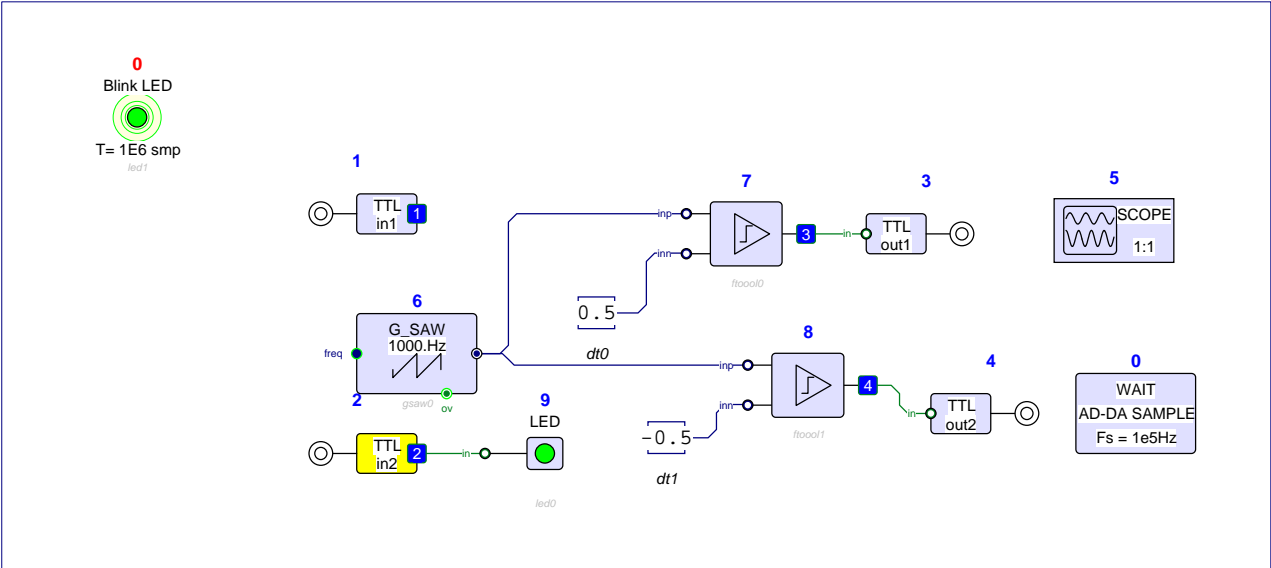


CATEGORY: ETD410K

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: ETD410K

DESCRIPTION:

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

INPUTS

Name:
name_in

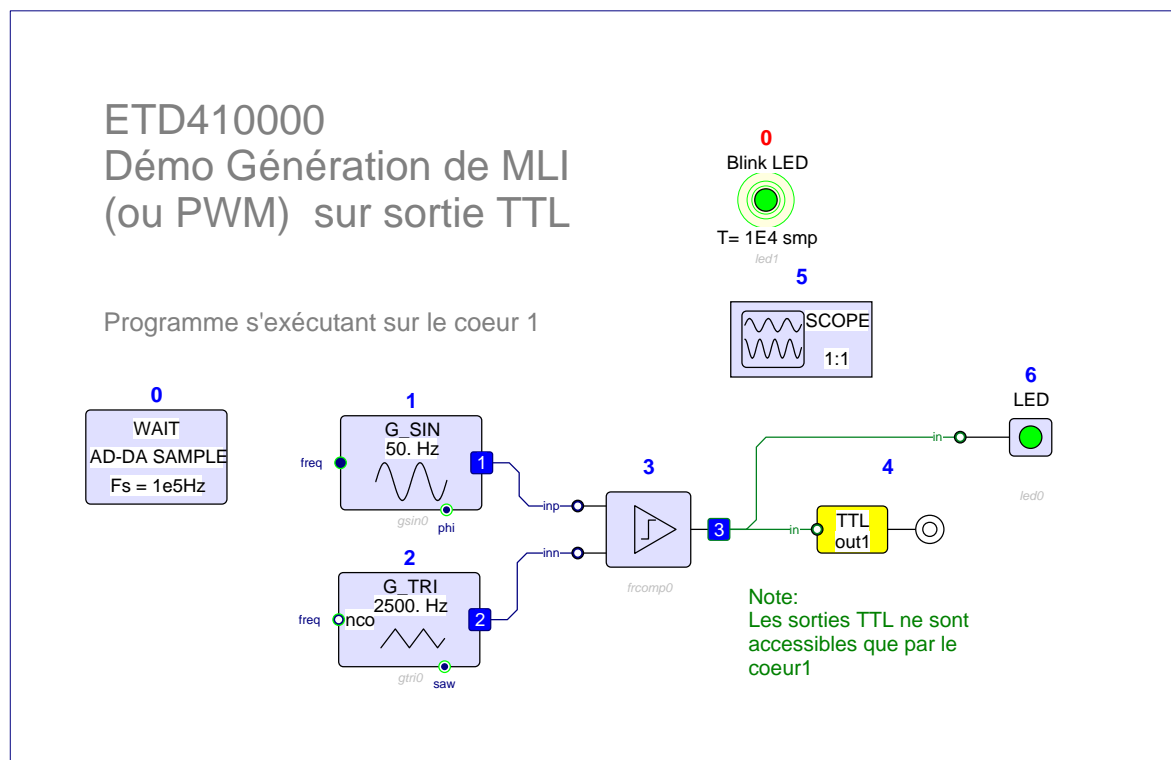
Data Type:
BOOL

Data Struct:
BIT

Connection:
mandatory

ATTRIBUTES

Unique,



TTL_OUT1 test program

TTL_OUT2

Digital output 1

TTL_OUT2



CATEGORY: ETD410K

DESCRIPTION:

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

INPUTS

Name:
name_in

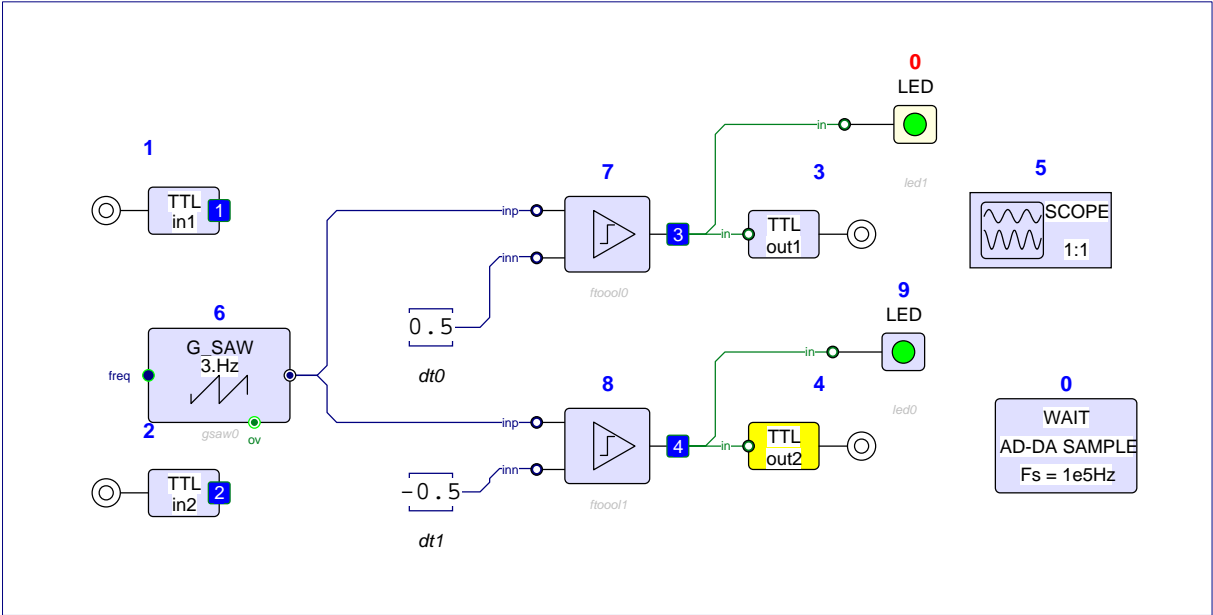
Data Type:
BOOL

Data Struct:
BIT

Connection:
mandatory

ATTRIBUTES

Unique,



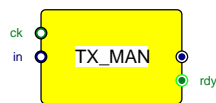
TTL_OUT2 test program

TX_AMI

TX_MAN

Manchester line coder

TX_MAN



CATEGORY: TELECOM

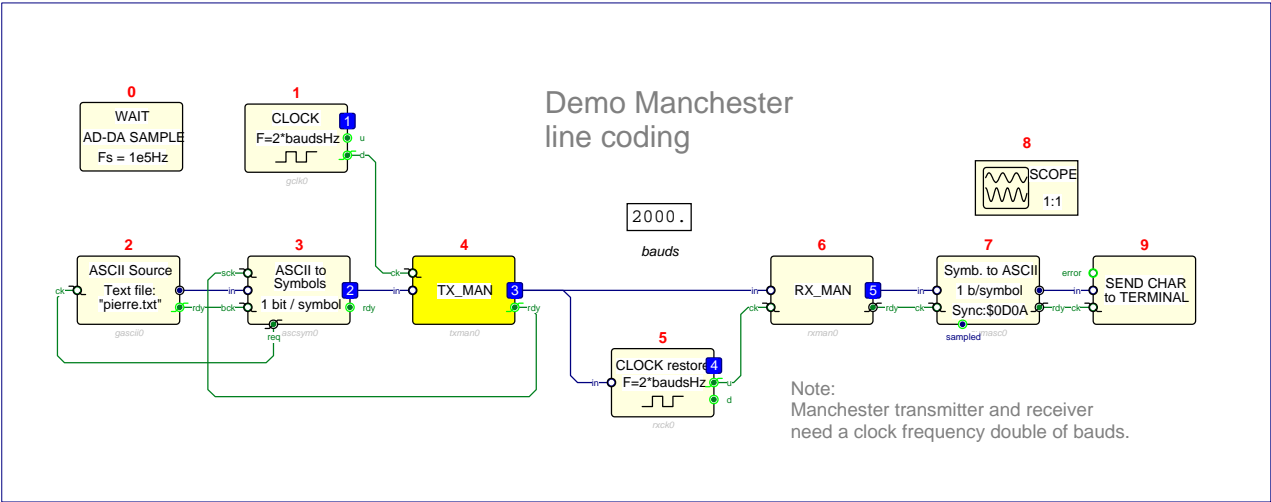
DESCRIPTION:
Manchester line coder
Input clock is 2 x Bauds

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

D-Manchester Coder

TX_MAND



CATEGORY: TELECOM

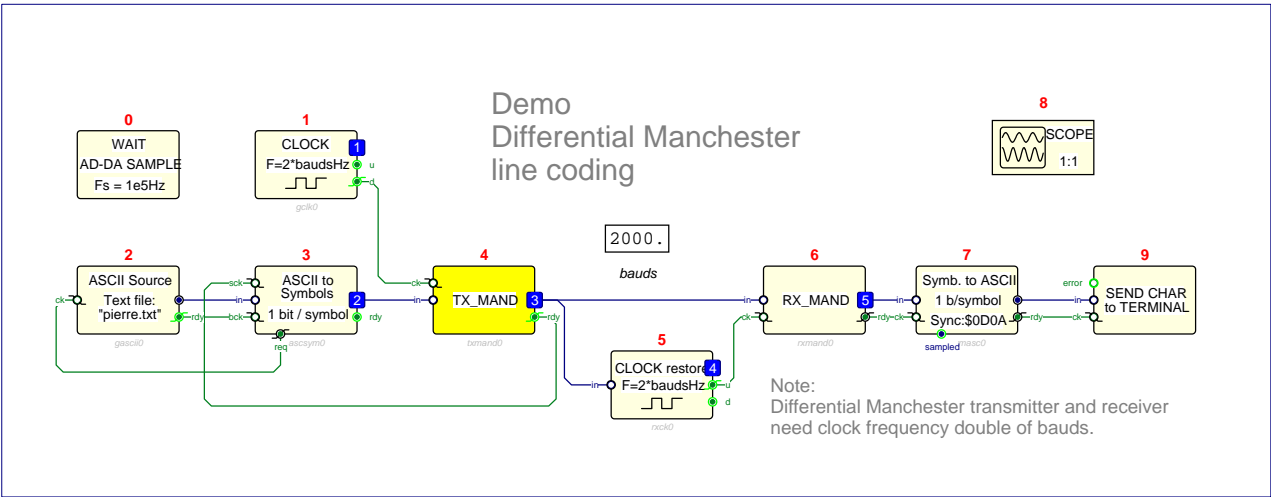
DESCRIPTION:
D-Manchester Coder
Differential Manchester line coder
Input clock is 2 x Bauds

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

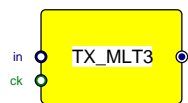


TX_MAND test program

TX_MLT3

MLT3 line coder

TX_MLT3



CATEGORY: TELECOM

DESCRIPTION:
MLT3 line coder

INPUTS

Name:
name_in
name_ck

Data Type:
FRACT
BOOL

Data Struct:
WORD
BIT

Connection:
mandatory
mandatory

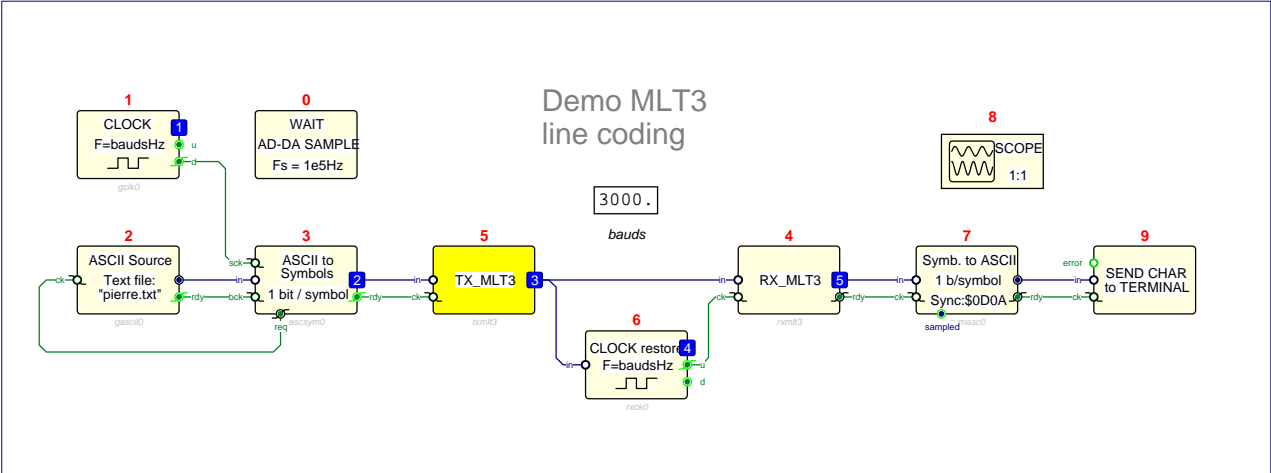
OUTPUTS

Name:
name

Data Type:
FRACT

Data Struct:
WORD

Connection:
normal

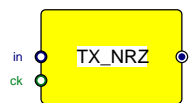


TX_MLT3 test program

TX_NRZ

Non Return to Zero line coder

TX_NRZ



CATEGORY: TELECOM

DESCRIPTION:
Non Return to Zero line coder

PARAMETERS:
Parameter:
Level Space
Level Mark

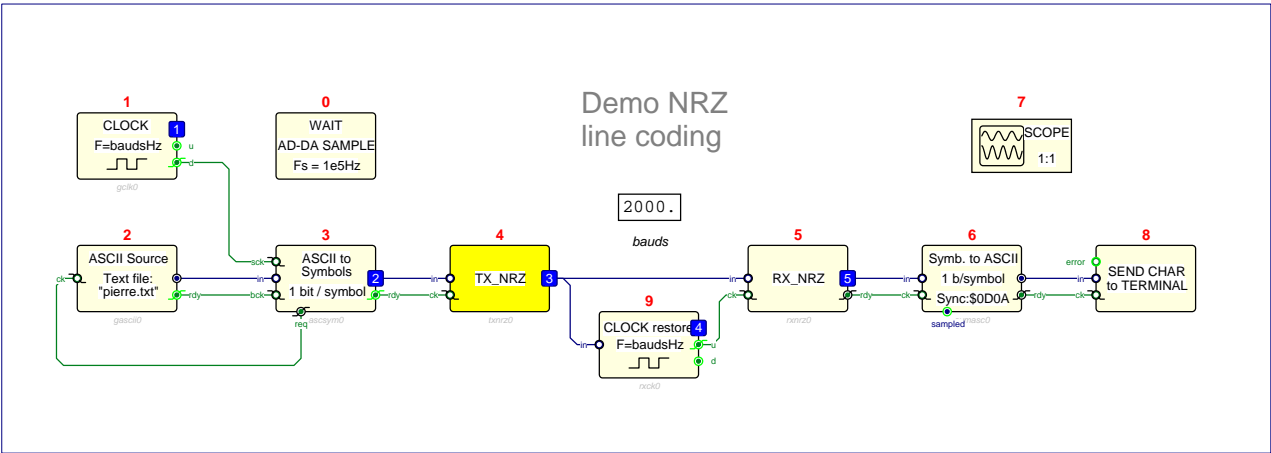
Default values:
-1.0
1.0

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

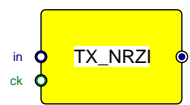


TX_NRZ test program

TX_NRZI

NRZI line coder

TX_NRZI



CATEGORY: TELECOM

DESCRIPTION:
NRZI line coder

PARAMETERS:
Parameter:
Level Space
Level Mark

Default values:
-1.0
1.0

INPUTS
Name:
name_in
name_ck

Data Type:
FRACT
BOOL

Data Struct:
WORD
BIT

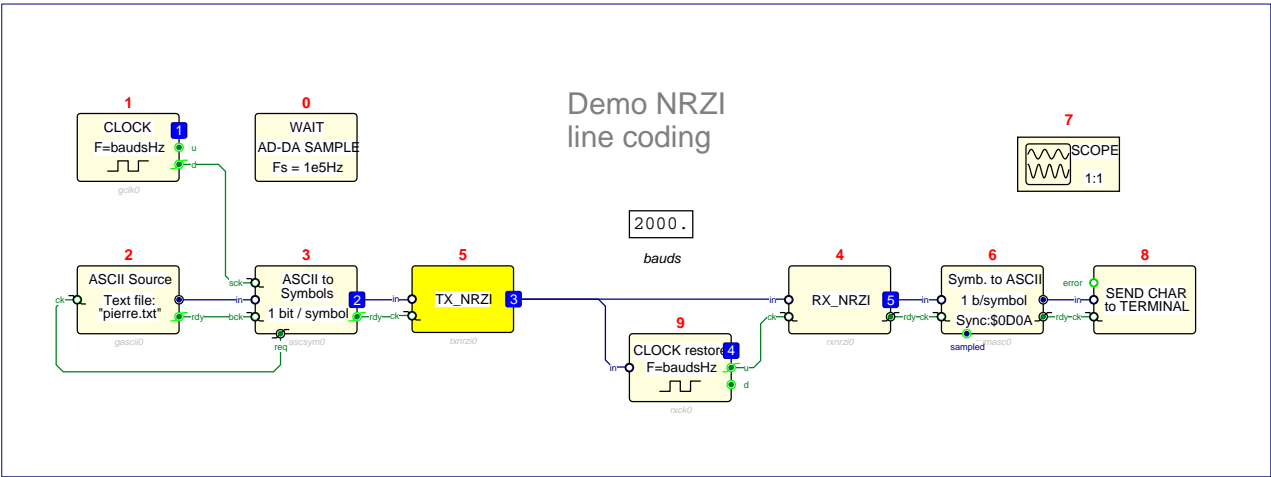
Connection:
mandatory
mandatory

OUTPUTS
Name:
name

Data Type:
FRACT

Data Struct:
WORD

Connection:
normal

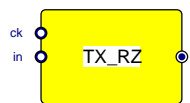


TX_NRZI test program

TX_RZ

Return to Zero line coder

TX_RZ



CATEGORY: TELECOM

DESCRIPTION:
Return to Zero line coder

INPUTS

Name:
name_in
name_ck

Data Type:
FRACT
FRACT

Data Struct:
WORD
WORD

Connection:
mandatory
mandatory

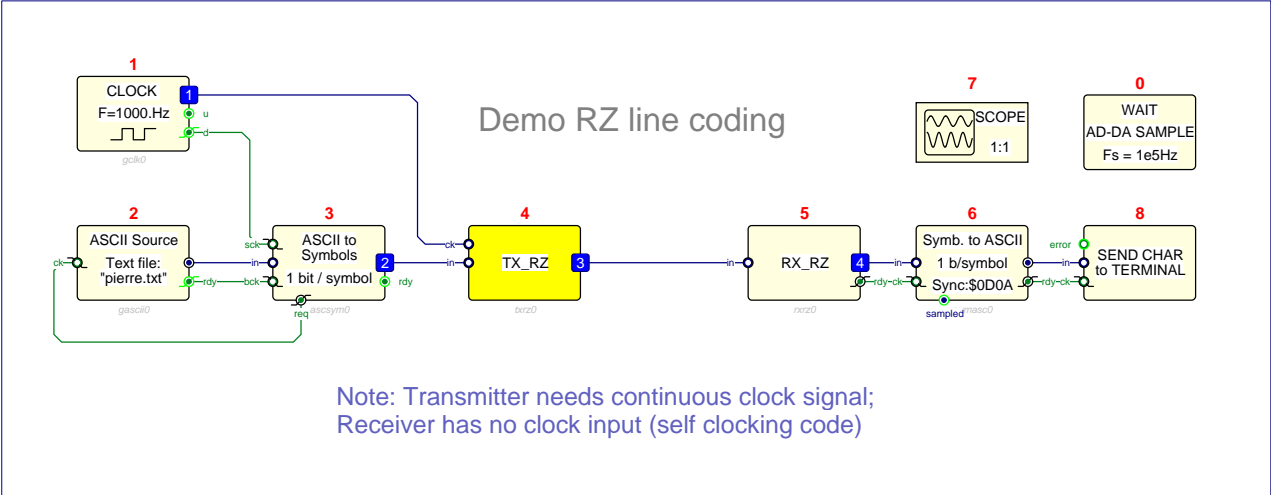
OUTPUTS

Name:
name

Data Type:
FRACT

Data Struct:
WORD

Connection:
normal



TX_RZ test program



CATEGORY: ETD410K

DESCRIPTION:
standart UART at 115KBauds

INPUTS

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

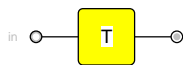
OUTPUTS

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

UDELAY

Unit delay z^{-1}

UDELAY



CATEGORY: CONTROL

DESCRIPTION:
Unit delay z^{-1}
Complex or real

INPUTS
Name:
name_in

Data Type:
defined by cn

Data Struct:

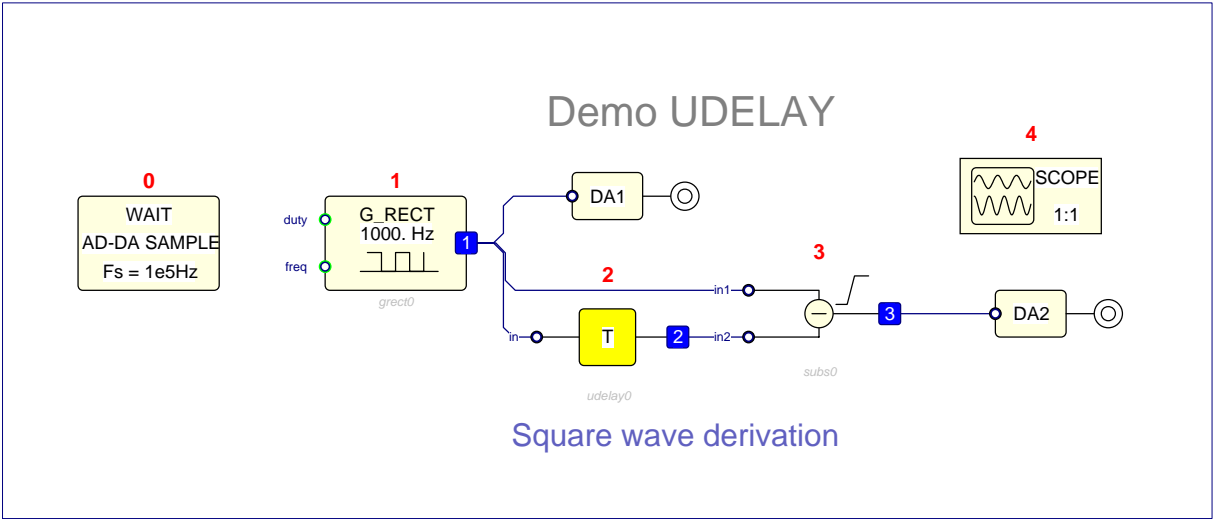
Connection:
mandatory

OUTPUTS
Name:
name

Data Type:
defined by cn

Data Struct:

Connection:
normal

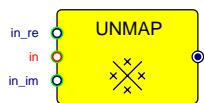


UDELAY test program

UNMAP

Complex to symbol

UNMAP



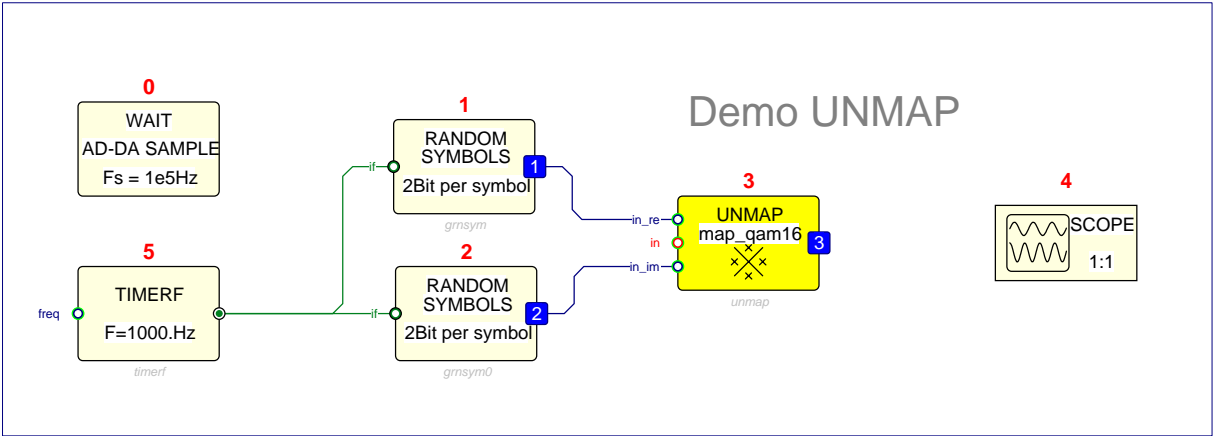
CATEGORY: TELECOM

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

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

INPUTS			
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name_in	COMPLEX	WORD	optional
name_in_re	FRACT	WORD	optional
name_in_im	FRACT	WORD	optional

OUTPUTS			
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name	FRACT	WORD	normal

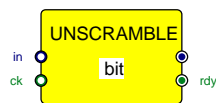


UNMAP test program

UNSCRAMBLE

Unscrambler

UNSCRAMBLE



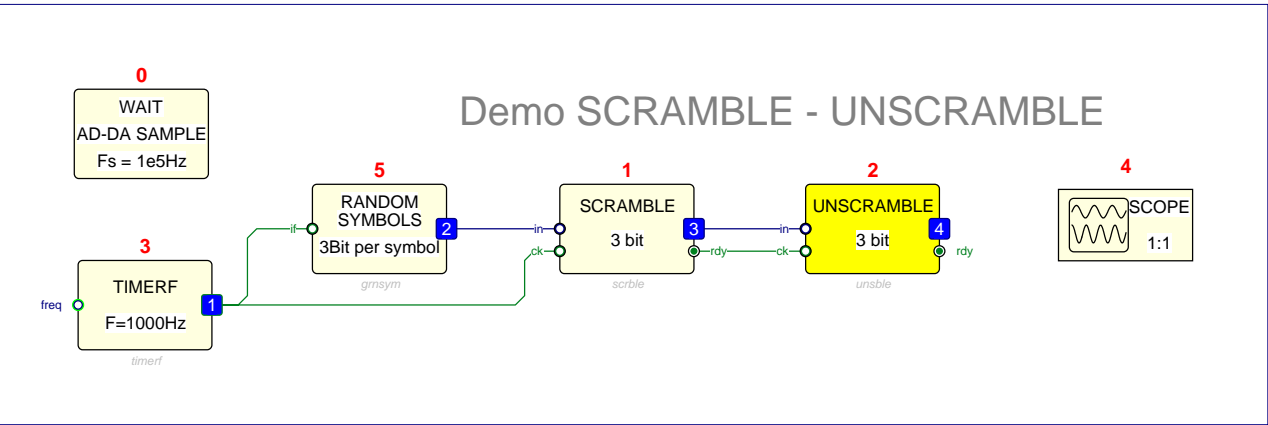
CATEGORY: TELECOM

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

PARAMETERS:
Parameter: Bits
Default values: 1

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

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



UNSCRAMBLE test program

V_CPLXPOW

Vector Complex Mean Power

V_CPLXPOW



CATEGORY: MATRIX

DESCRIPTION:
Vector Complex Mean Power
for each element x calculate y that is a LP filtered value of x
 $y = y + ((x_{re}^2 + x_{im}^2) < <g - y > > p$

PARAMETERS:

<i>Parameter:</i>	<i>Default values:</i>
Time Ct 2spower	5
Gain 2spower	2

INPUTS			
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name_in	COMPLEX	Matrix of DWORD	mandatory
OUTPUTS			
<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name	FRACT	Matrix of DWORD	normal

VECT_POW

Vector Power

VECT_POW



CATEGORY: MATRIX

DESCRIPTION:

Vector Power
 $y[i] = \text{average}(|x[i]|^2)$

INPUTS

Name:
name_in

Data Type:
defined by cn

Data Struct:
Matrix of

Connection:
mandatory

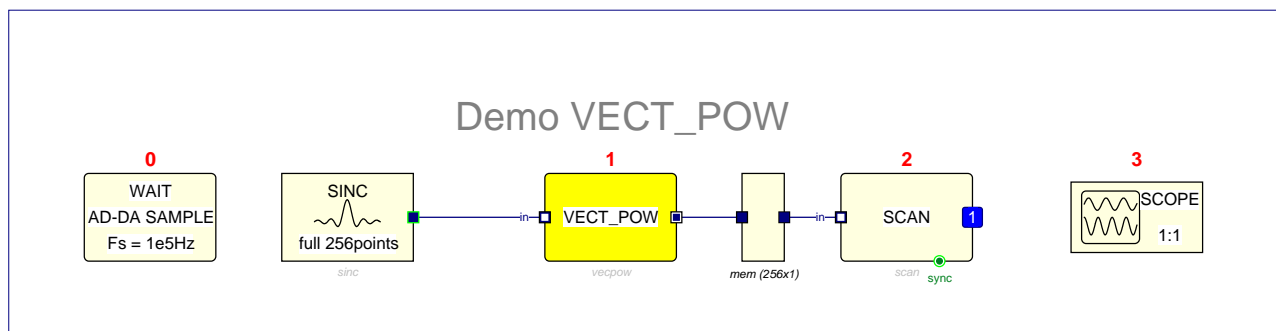
OUTPUTS

Name:
name

Data Type:
FRACT

Data Struct:
Matrix of DWORD

Connection:
normal

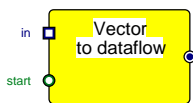


VECT_POW test program

VECTTOFLOW

Vector to dataflow

VECTTOFLOW



CATEGORY: MATRIX

DESCRIPTION:
Vector to dataflow

INPUTS
Name:
name_in
name_start

Data Type:
FRACT
BOOL

Data Struct:
Matrix of WORD
BIT

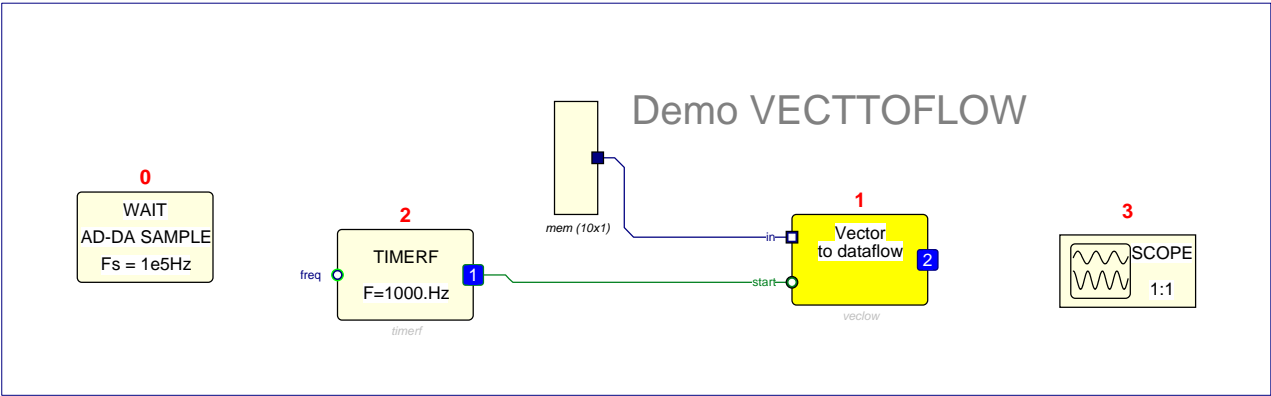
Connection:
mandatory
mandatory

OUTPUTS
Name:
name

Data Type:
FRACT

Data Struct:
WORD

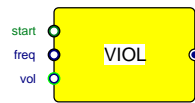
Connection:
normal



VECTTOFLOW test program

VIOL

VIOL



INPUTS

Name:
name_start
name_freq
name_vol

Data Type:
BOOL
FRACT
FRACT

Data Struct:
BIT
WORD
WORD

Connection:
mandatory
mandatory
optional

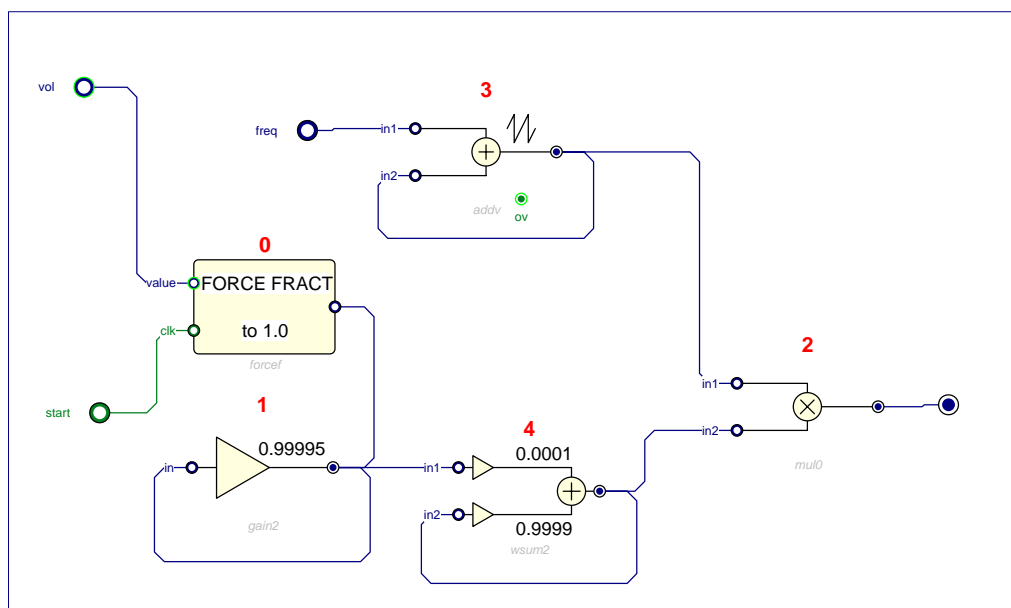
OUTPUTS

Name:
name

Data Type:
FRACT

Data Struct:
WORD

Connection:
normal



VIOL internal schema

WAITFLAG

Wait and clear flag

WAITFLAG



CATEGORY: CONTROL

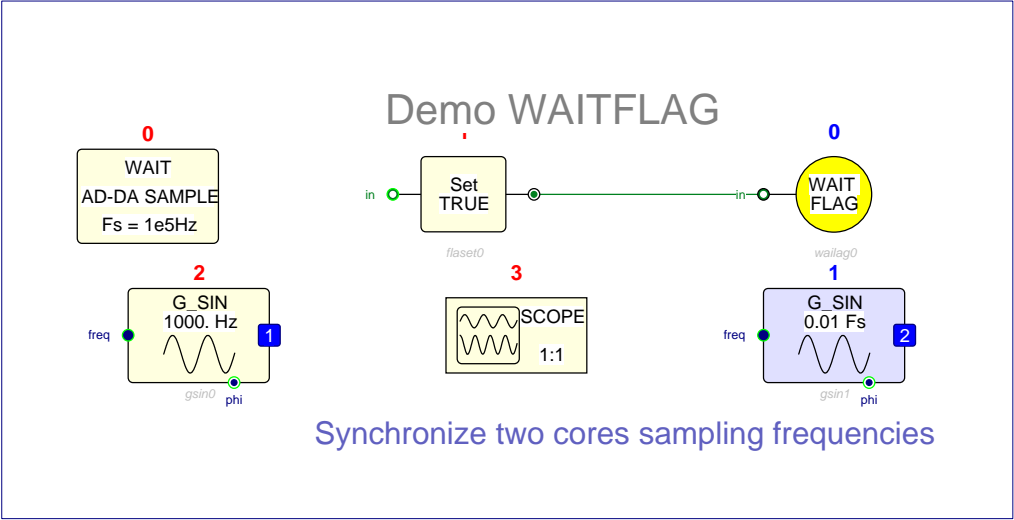
DESCRIPTION:
Wait and clear flag

INPUTS
Name:
name_in

Data Type:
BOOL

Data Struct:
BIT

Connection:
mandatory



WAITFLAG test program

WAITKEY

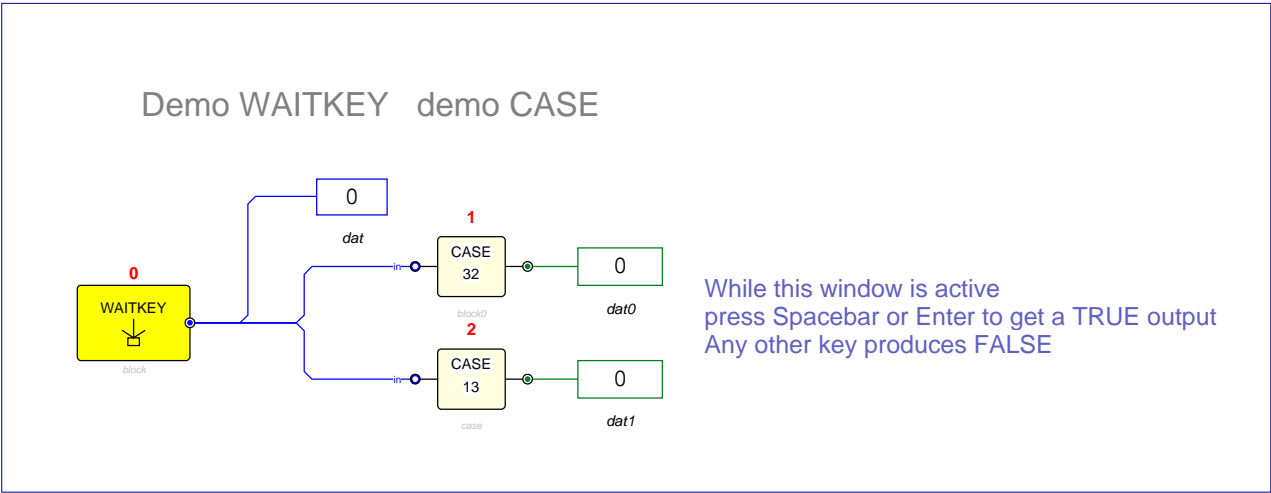
Wait until key pressed

WAITKEY



DESCRIPTION:
Wait until key pressed
Returns key value as INTEGER
Allows executing algorithm once on keyboard action

OUTPUTS			
<i>Name:</i> name	<i>Data Type:</i> INTEGER	<i>Data Struct:</i> WORD	<i>Connection:</i> normal

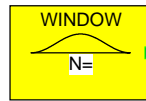


WAITKEY test program

WINDOW

Weighting Window

WINDOW



CATEGORY: MATRIX

DESCRIPTION:

Weighting Window

(Y: memory)

If Gain = Max, max of window is 1.0, mean is window dependant

If Gain = 1/8, mean value of window is 1/8

If Gain = 1/N, mean value of window is 1/size

PARAMETERS:

Parameter:

Window type

Size

Gain

Default values:

Triangle,Hann,Hamming,Gauss,Blackman_Harris,Nuttal,Flat_Top

512

Max,1/8,1/N

OUTPUTS

Name:

name

Data Type:

FRACT

Data Struct:

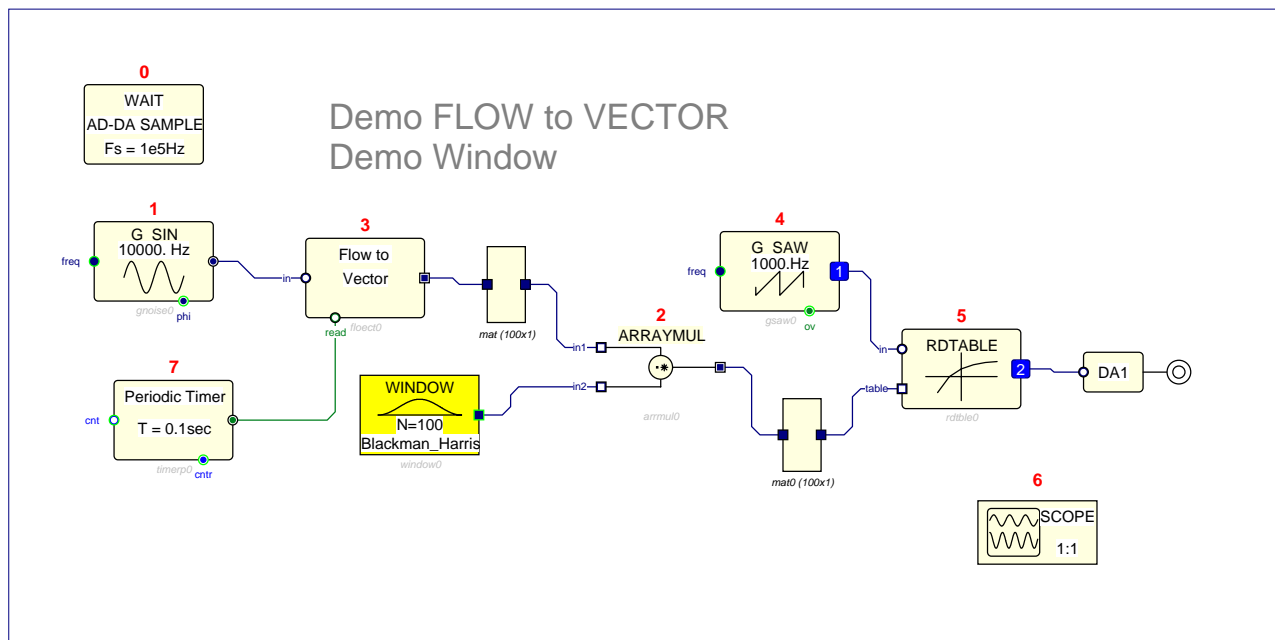
Matrix of WORD

Connection:

optional

ATTRIBUTES

Non executable, Data Table

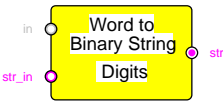


WINDOW test program

WORDTOBIN

Word to Binary

WORDTOBIN



CATEGORY: STRING

DESCRIPTION:

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

PARAMETERS:

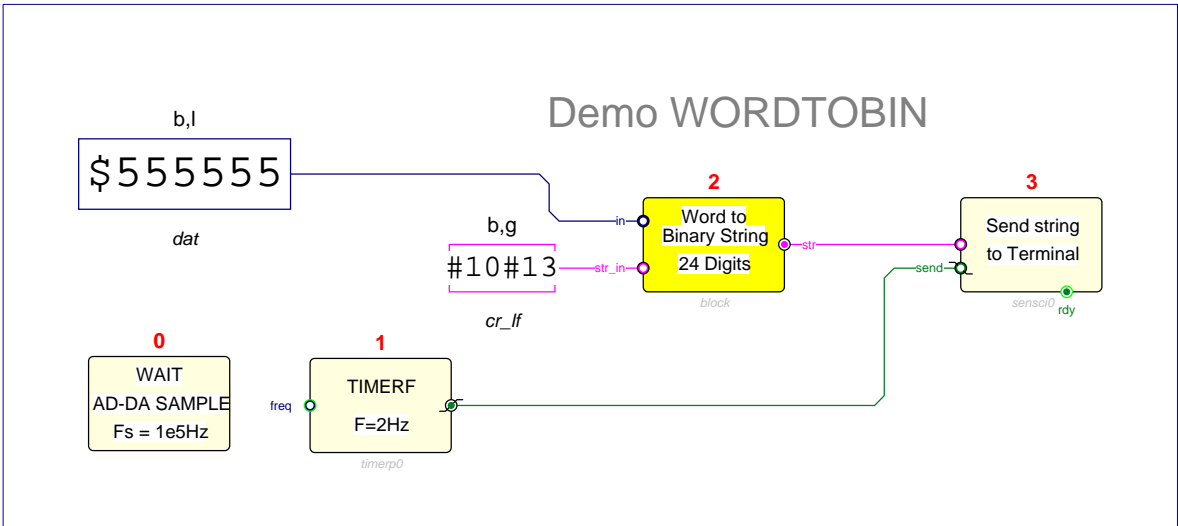
<i>Parameter:</i>	<i>Default values:</i>
Nb digits	24
Bit pos of MSB	23

INPUTS

<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name_in	defined by cn	WORD	mandatory
name_str_in	STRING		mandatory

OUTPUTS

<i>Name:</i>	<i>Data Type:</i>	<i>Data Struct:</i>	<i>Connection:</i>
name_str	STRING	WORD	normal

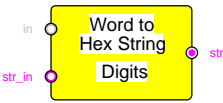


WORDTOBIN test program

WORDTOHEX

Word to hexadecimal

WORDTOHEX



CATEGORY: STRING

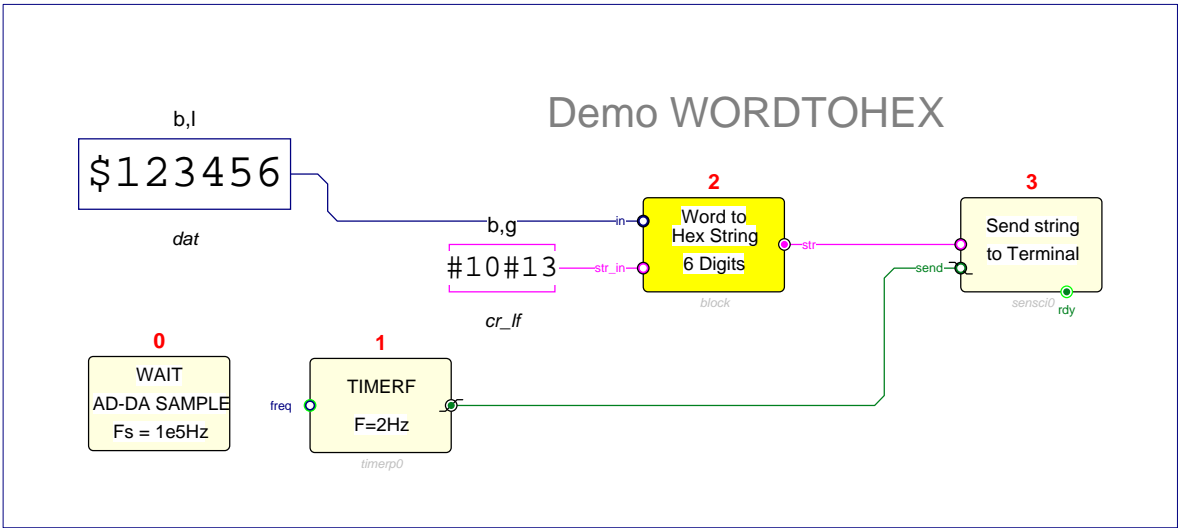
DESCRIPTION:
Word to hexadecimal
Convert word input to hex string

PARAMETERS:

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

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

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

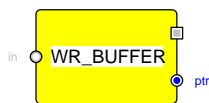


WORDTOHEX test program

WR_BUFFER

Write to Circular Buffer

WR_BUFFER



CATEGORY: MATRIX

DESCRIPTION:
Write to Circular Buffer

INPUTS

Name:
name_in

Data Type:
defined by cn

Data Struct:

Connection:
mandatory

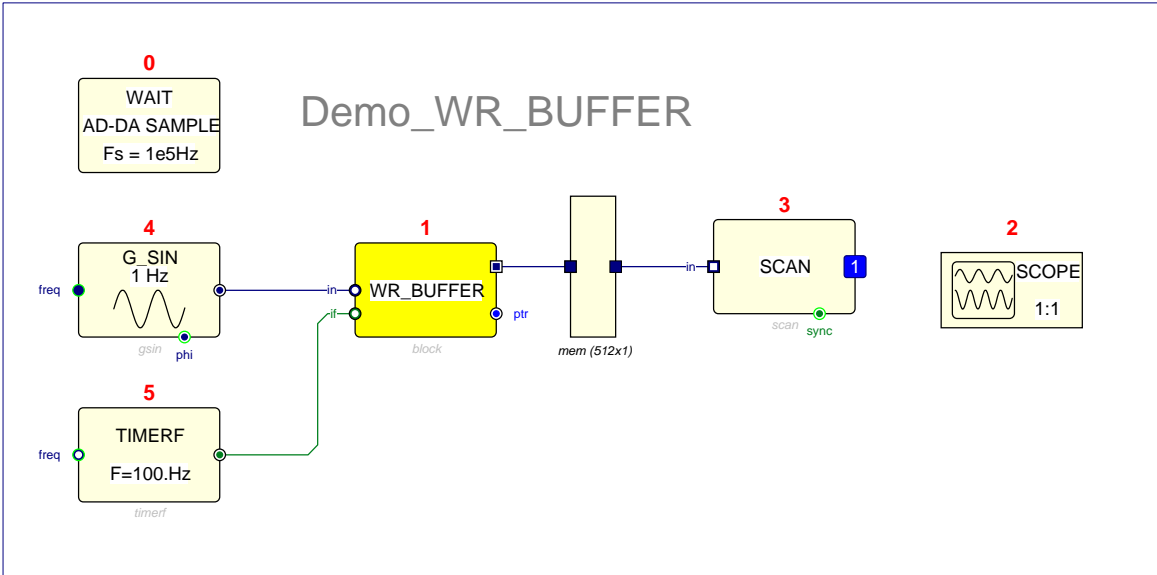
OUTPUTS

Name:
name_ptr

Data Type:
INTEGER

Data Struct:
Matrix of
WORD

Connection:
normal
normal

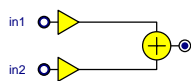


WR_BUFFER test program

WSUM2

Weighted sum of 2 inputs:

WSUM2



CATEGORY: ARITHMETIC

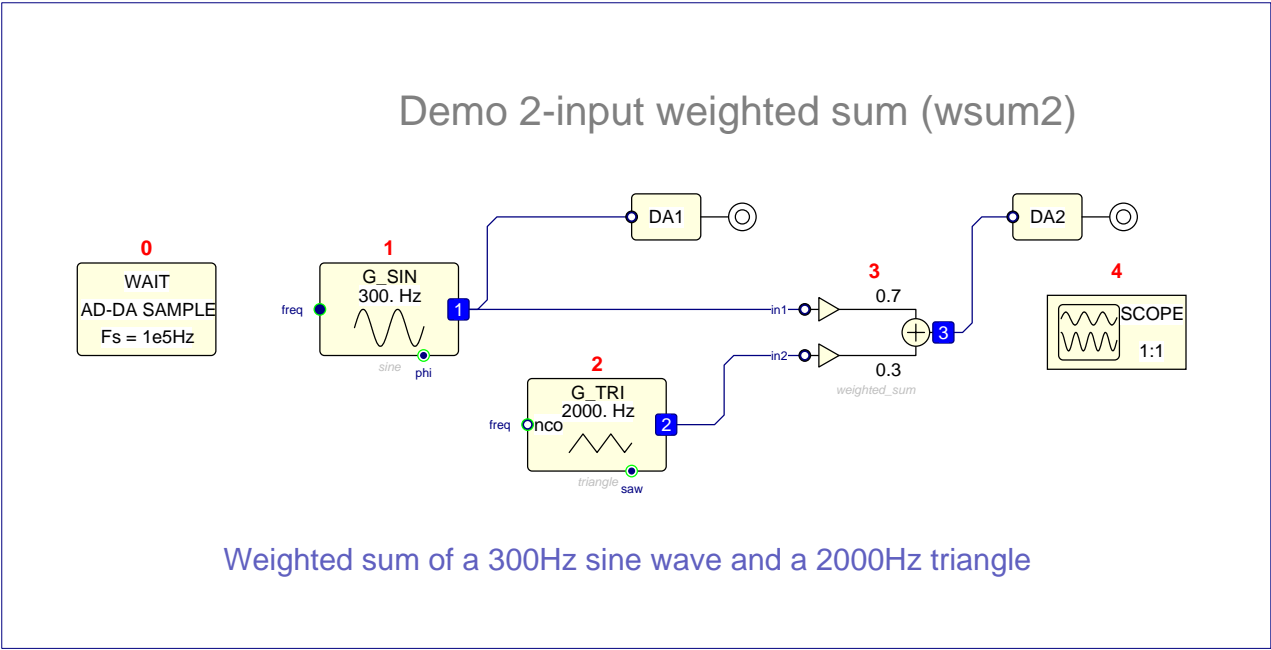
DESCRIPTION:
Weighted sum of 2 inputs:
 $y = g1 \cdot in1 + g2 \cdot in2$
 $-1.0 \leq g1, g2 \leq +1.0$

PARAMETERS:

Parameter:	Default values:
gain1	1.0
gain2	1.0

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

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

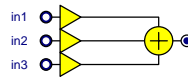


WSUM2 test program

WSUM3

Weighted sum of 3 inputs:

WSUM3



CATEGORY: ARITHMETIC

DESCRIPTION:

Weighted sum of 3 inputs:
 $y = g1 \cdot in1 + g2 \cdot in2 + g3 \cdot in3$
 $-1 \leq g1, g2, g3 \leq +1$

PARAMETERS:

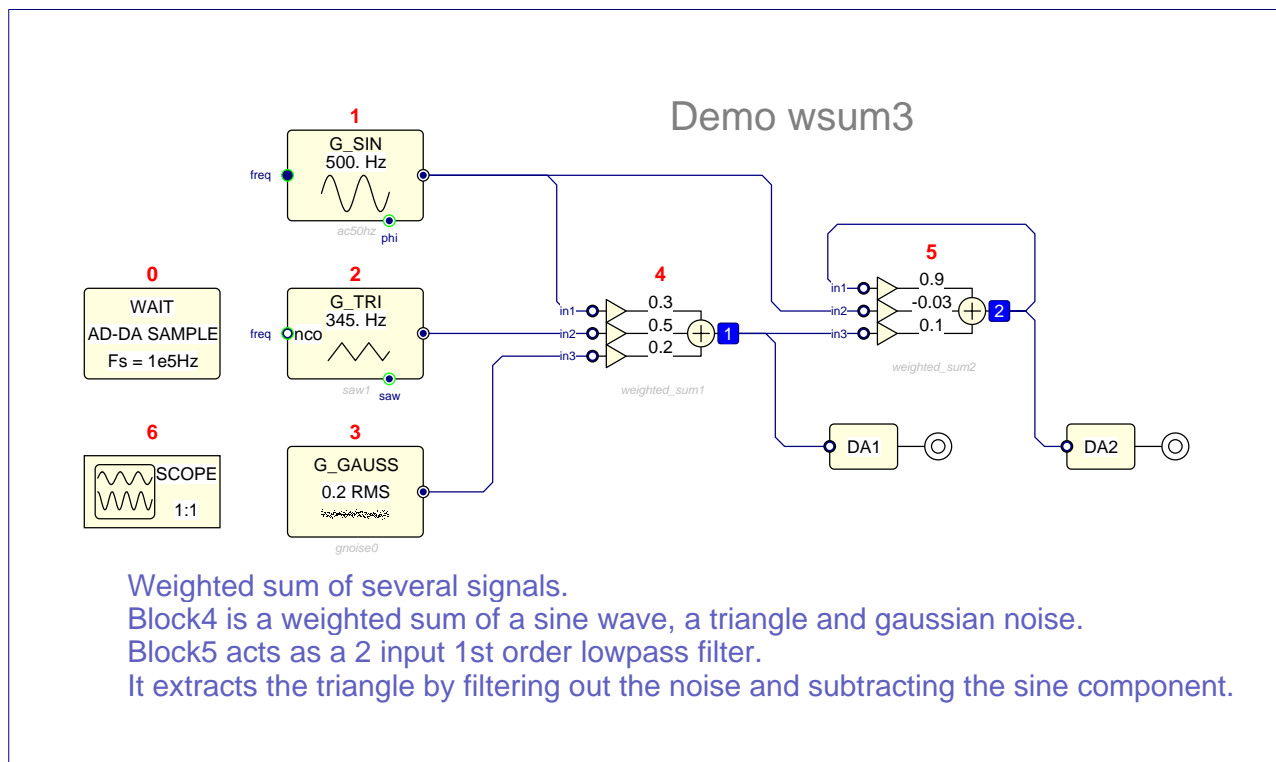
Parameter:	Default values:
gain1	1.0
gain2	1.0
gain3	1.0

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

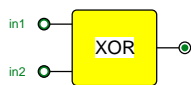


WSUM3 test program

XORGATE

Logic Exclusive OR

XORGATE



CATEGORY: LOGIC

DESCRIPTION:
Logic Exclusive OR
 $y = (in1 \& in2) \vee (in1 \& \neg in2)$

INPUTS

Name:
name_in1
name_in2

Data Type:
BOOL
BOOL

Data Struct:
BIT
BIT

Connection:
mandatory
mandatory

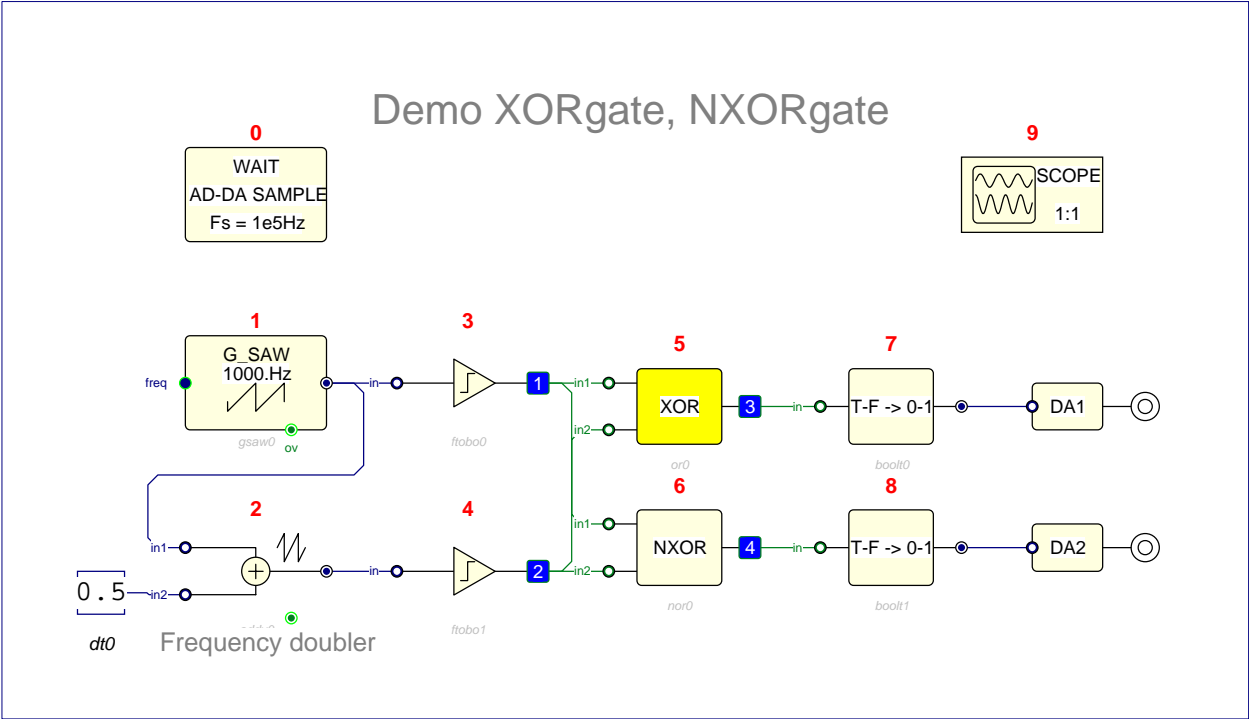
OUTPUTS

Name:
name

Data Type:
BOOL

Data Struct:
BIT

Connection:
normal

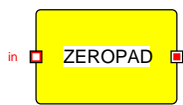


XORGATE test program

ZEROPAD

Zero Padding

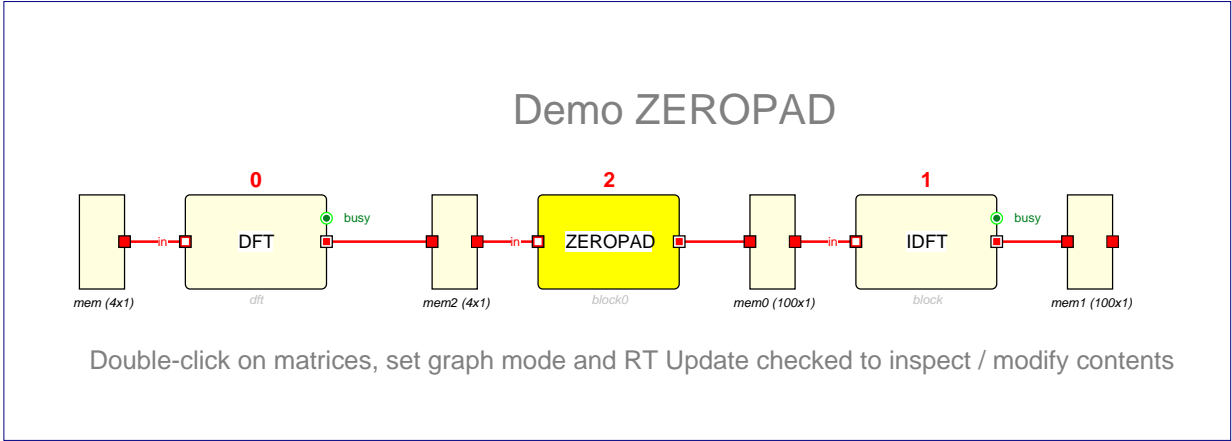
ZEROPAD



CATEGORY: MATRIX

DESCRIPTION:
Zero Padding
Transforms size N to size M with $M > N$
such as points around 0 (mod N) remain around 0 (mod M)

INPUTS			
Name:	Data Type:	Data Struct:	Connection:
name_in	COMPLEX	Matrix of DWORD	mandatory
OUTPUTS			
Name:	Data Type:	Data Struct:	Connection:
name	COMPLEX	Matrix of DWORD	normal



ZEROPAD test program

