LECOEUR ELECTRONIQUE

19 route de Courtenay - 45220 – CHUELLES –
☎: 02 38 94 27 30– 🖷: 02 38 94 29 67
💻: info@lecoeur-electronique.com

USER'S GUIDE FOR MATALB INTERFACE

OPEN SYSTEM

Contents

1) Matlab functions description

↥

Init (With transmitter's init,Nb of channels,Nb of sequences);

Description: *Makes OPEN system general initialization. Must be done at each power-on of the system.*

Parameters:

*With transmitter's init: when 0 inits the system without loading transmitter's curves, when 1 loads transmitter's curves. (Refer to **Excel Global set-up file description** to locate curves files on the hard disk)*

Number of channels: selects the number of channels to be initialized (starting from first channel).

Number of sequences: selects the number of sequences to be uploaded to OPEN (0 to 256).

Example:

```
LecoeurUS('Init',1,32,64);
```

Inits 32 channels with 64 sequences.

↥

Init_USB ();

Description: *Makes OPEN USB link initialization. Must be done each time you enter in Matlab application. This function could be used when the system is previously initialized, instead of using the Init function which could be time consuming.*

Example:

Init_USB ();

↥

Gain (Gain in dB (0 to 79.9), Channel);

Description: Sets the receiver gain of the selected channel.

Parameters:

Gain in dB: The value of the receiver's gain (0 to 79.9 dB, by step of 0.1 dB).

Channel: Channel number (1 to n).

Example:

```
LecoeurUS('Gain',60,16);
```

Sets the gain of channel 16 to 60 dB

↥

Retard (Delay in µs (0 to 160 µs), Channel);

Description: *For the selected channel, sets the delay between the beginning of emission and the digitalization. Works only for Ascan, Bscan, Brascan, Brcscan functions. This feature has no effect on the micro-sequencer.*

Parameters:

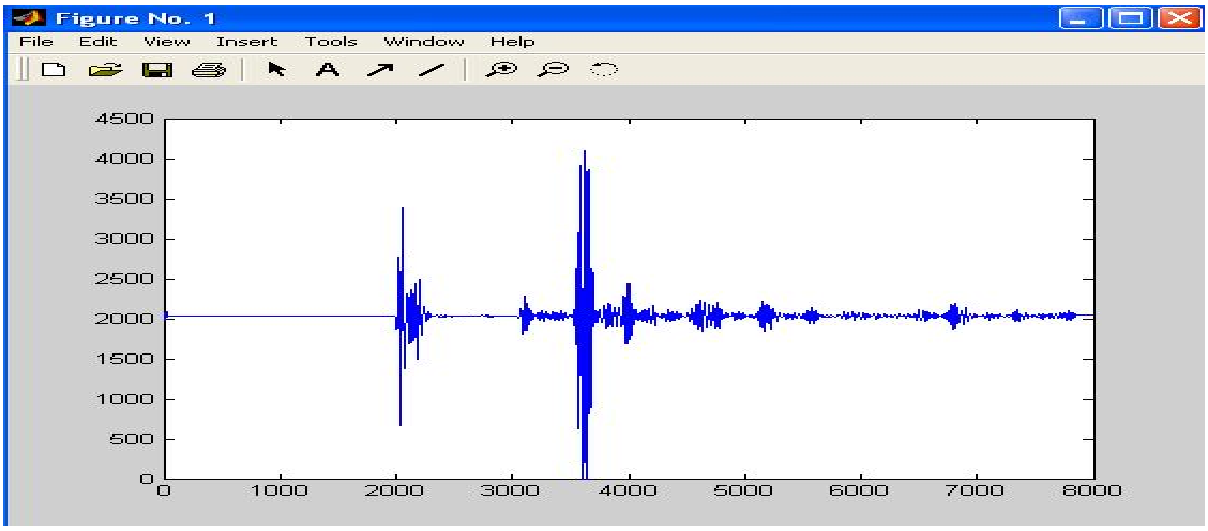
Delay in µs: The value of the delay before digitalization (µs). Maximum value is 160 µs.

Channel: Channel number (1 to n).

Example:

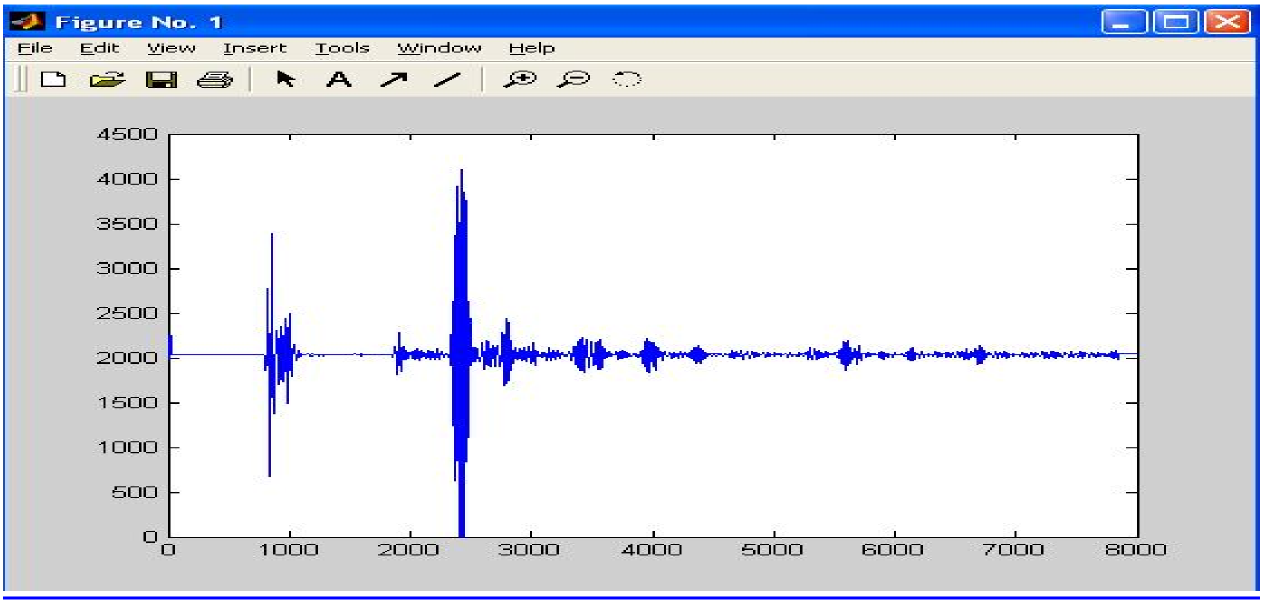
LecoeurUS('Retard',50,1);

Sets the delay of channel 1 to 50 µs



LecoeurUS('Retard',80,1);

Sets the delay of channel 1 to 80µs.



↥

RF(t)=Ascan (Nb of samples+10, Channel, Log/Lin(0/1));

Description: *Returns the A-SCAN signal of the selected channel.*

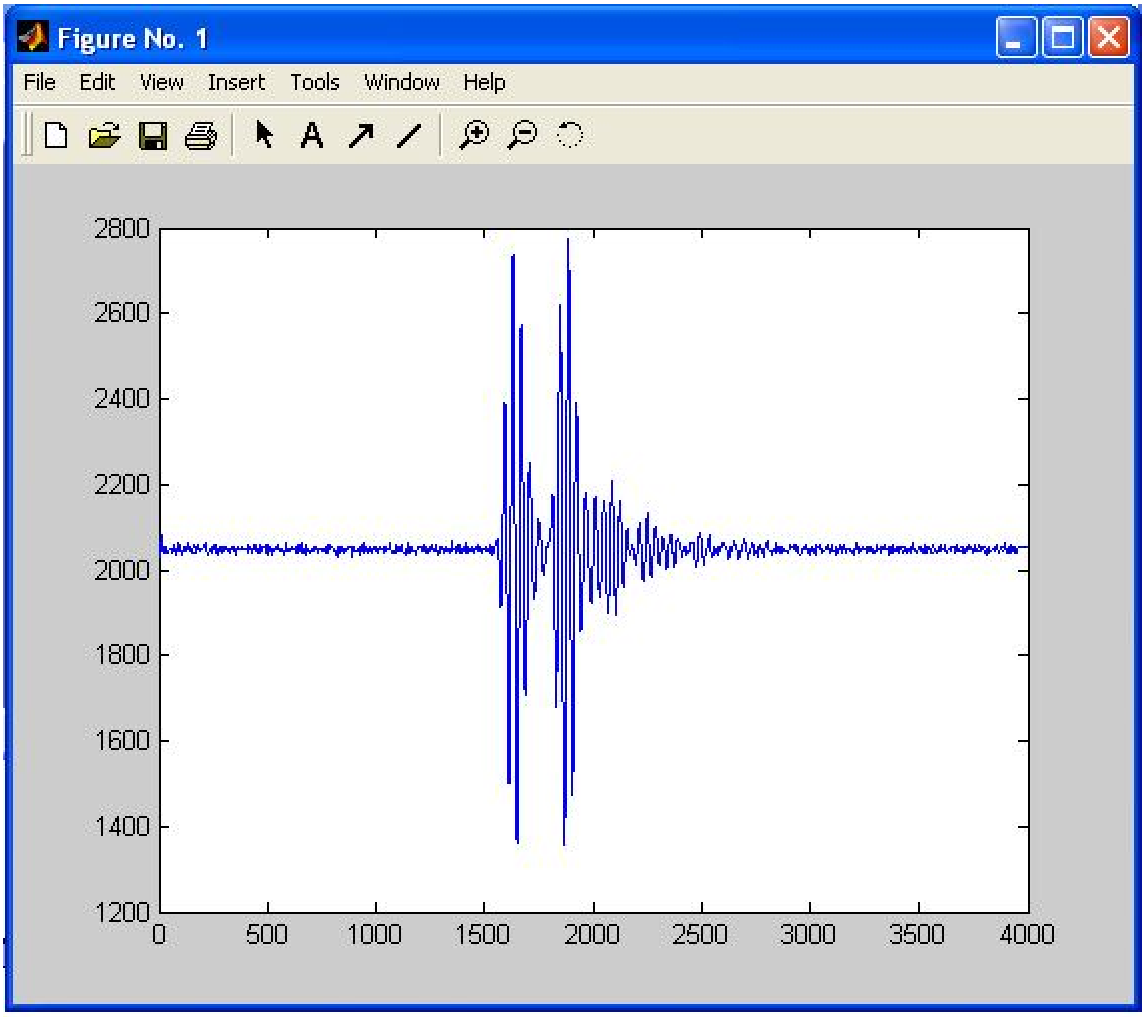
Parameters:

Nb of samples: Number of samples on time axis.

Channel: Channel number (1 to n).

Log=0/Lin=1: This parameter is set to 1 to get a linear A-SCAN, passing 0 returns a logarithmic A-SCAN.

Linear A-SCAN get with Ascan function.



Example:

plot(LecoeurUS('Ascan',8000,1,1));

Displays A-scan of channel 1, sampled with 8000 samples.

↥

RF(ch,t)=Bscan (Nb of samples+10, Axial resolution, Log/Lin(0/1));

Description: Creates a B-SCAN composed with all channels. The returned value is a matrix (nb of channels, nb of samples/decimation).

Parameters:

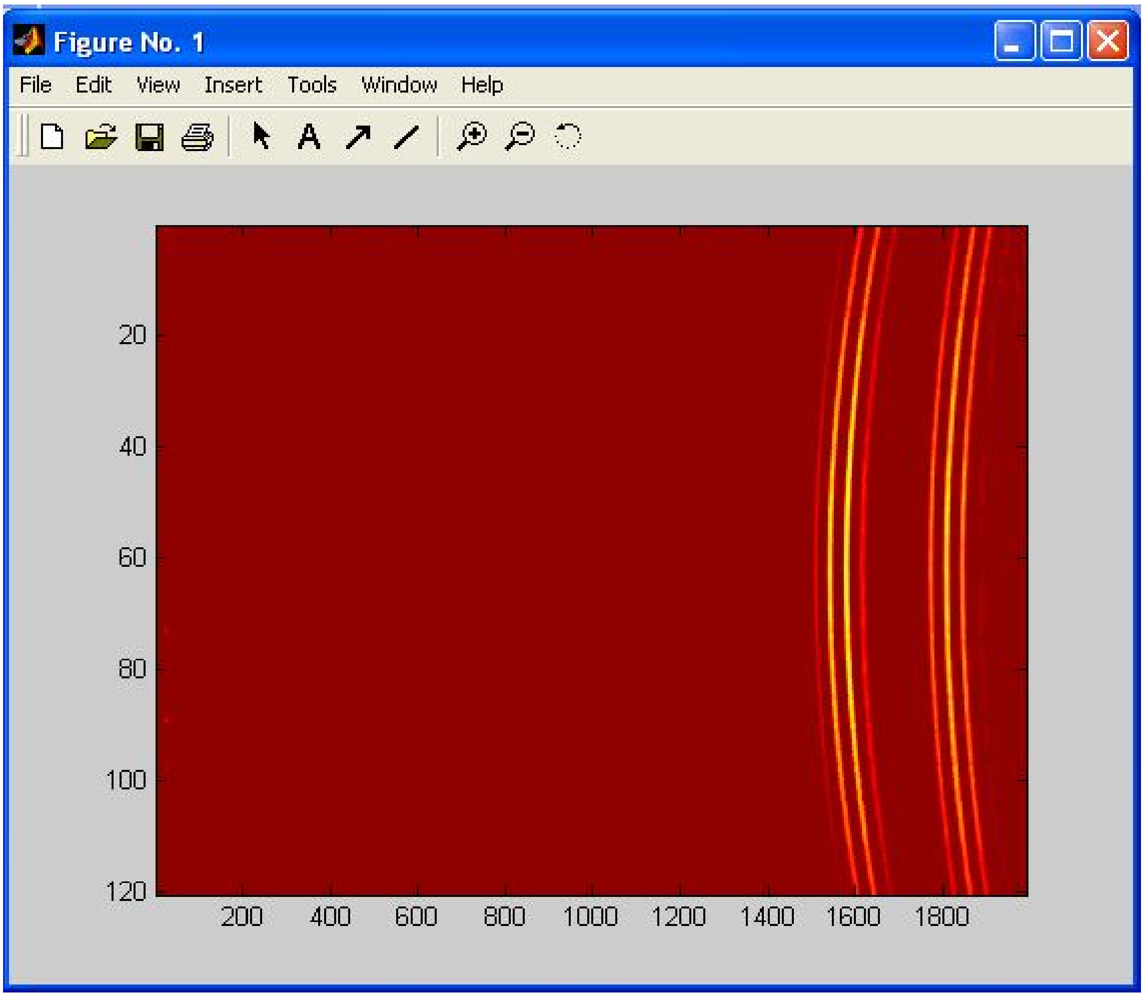
Nb of samples: Number of samples on time axis.

Axial resolution: Makes a decimation on time axis. Value is between 1 and 7:
1-> No decimation
2-> Decimation of 3
3-> Decimation of 7
4-> Decimation of 21
5-> Decimation of 27
6-> Decimation of 49
7-> Decimation of 63

Log=0/Lin=1: This parameter is set to 1 to get a linear B-SCAN, passing 0 returns a logarithmic B-SCAN.

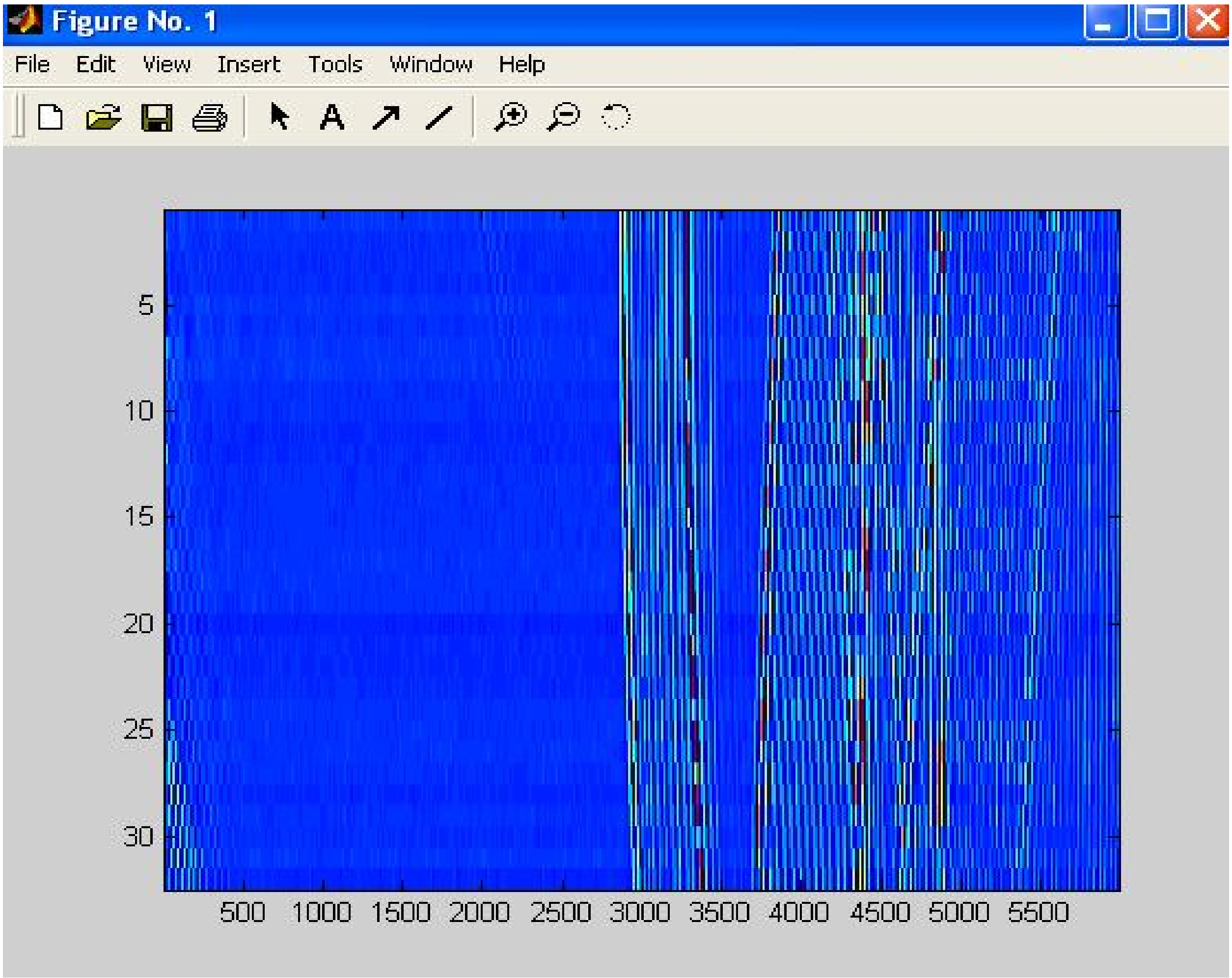
Example:

imagesc(LecoeurUS('Bscan',2000,1,1) ,[2000 2200]);



```
imagesc(LecoeurUS('Bscan',6000,1,1),[2000 2200]);
```

Display the B-scan of all channels without decimation.



↥

Setseq (Active sequence(0 to 255));

Description: *Sets the active sequence (transmitter waveform and receiver delay law) to sequence n. Works only for Ascan, Bscan, Brascan, Brcscan functions. Micro-sequencer is not concerned by this feature.*

Parameters:

Active sequence: *Active sequence number (0 to 255). This value activates a new transmitter curve and a new delay law for receivers.*

Example:

LecoeurUS('Setseq',32);

Activates the Sequence number 32.

↥

BF(t)=Brascan (Nb of samples+10);

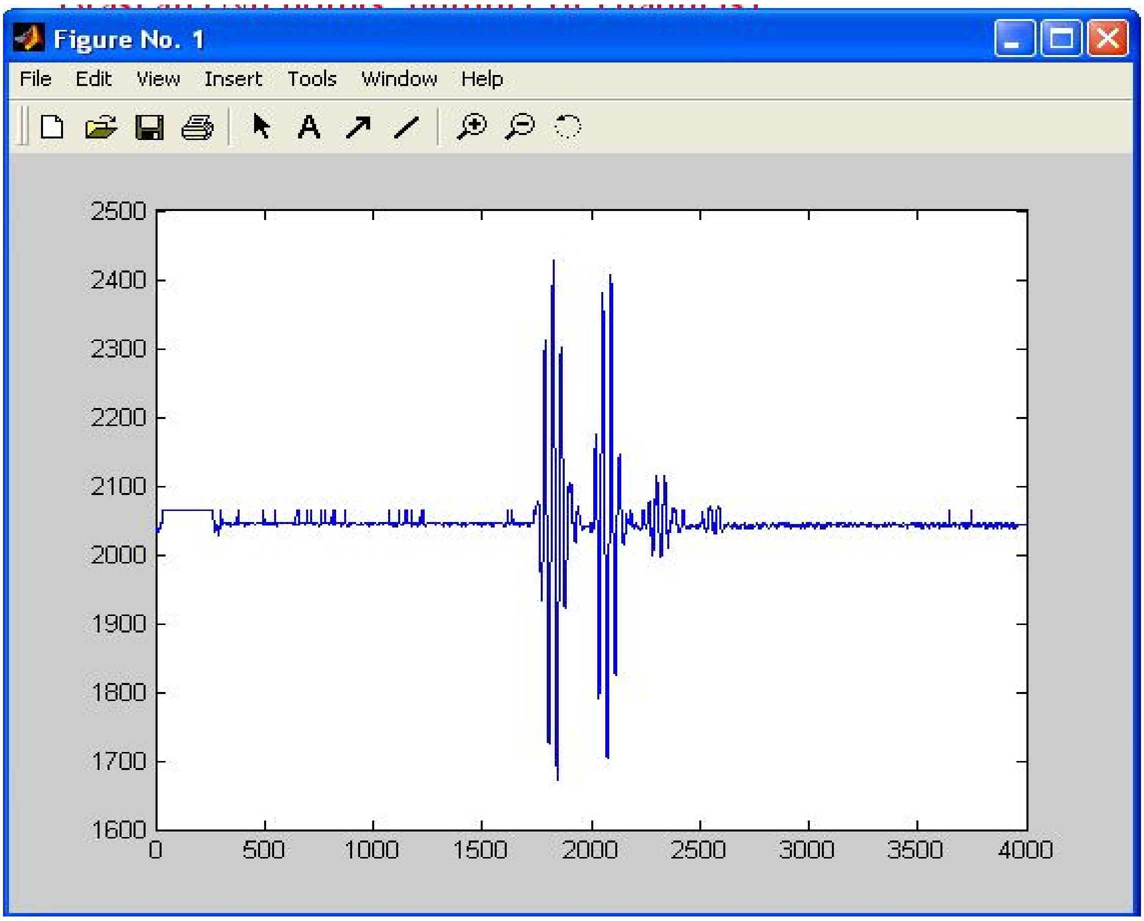
Description: *Returns one A-SCAN which is the result of sum of all A-SCANs for the selected sequence, applying the delay law before summation.*

Parameters:

Nb of samples: Number of samples to transfer for the BRA-SCAN.

Examples:

plot(LecoeurUS('Brcscan',4000,1,1) ,[2000 2200]);



LecoeurUS('Setseq',32) ;

plot(LecoeurUS('Brascan',8000));

Displays the A-SCAN which is the sum of all channels for the sequence 32.

↥

BF(ch,t)=Brcscan (Nb of samples+10, Axial resolution, Log/Lin(0/1));

Description: Creates a B-SCAN composed with n channels, each line is the sum of all channels. First line is a BRA-SCAN obtained using sequence 0, last line using sequence n-1. The returned value is a matrix (nb of channels, nb of samples/decimation)

Parameters:

Nb of samples: Number of samples on time axis.

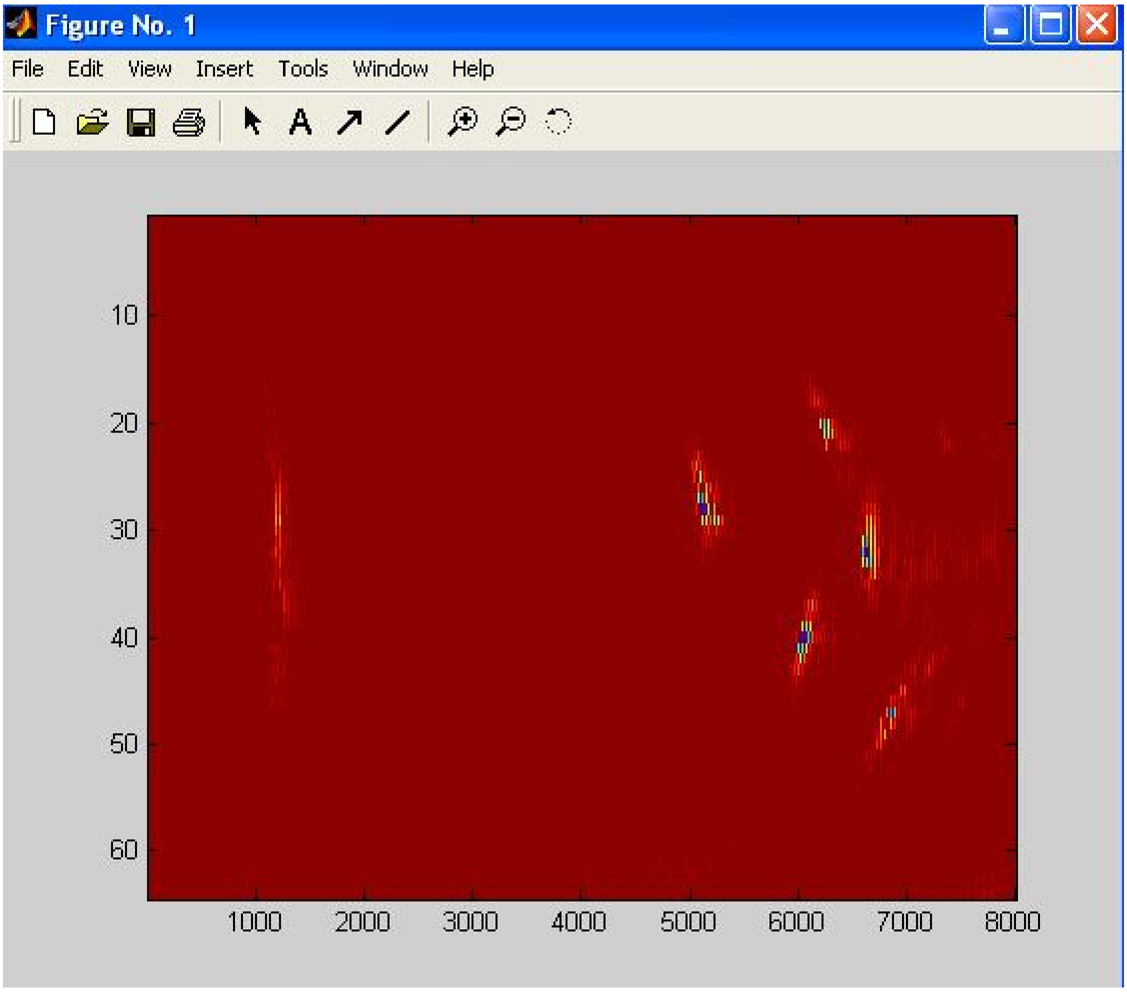
Axial resolution: Makes a decimation on time axis. Value is between 1 and 7:
1-> No decimation
2-> Decimation of 3
3-> Decimation of 7
4-> Decimation of 21
5-> Decimation of 27
6-> Decimation of 49
7-> Decimation of 63

Log=0/Lin=1: This parameter is set to 1 to get a linear BRC-SCAN, passing 0 returns a logarithmic BRC-SCAN.

Example:

```
imagesc(LecoeurUS('Brcscan',8000,1,1),[0 2000]);
```

Displays the B-scan of all channels for all sequences.



↥

FrequenceRecurrence (PRF(kHz));

Description: Sets the Pulse Repetition Frequency (PRF) of the OPEN system. Passing 0 stops all fights.

Parameters:

PRF: PRF in kHz from 0.1 kHz to 15 kHz. Value 0 stops all fights.

Example:

```
LecoeurUS('FrequenceRecurrence',1);
```

Sets the PRF to 1 kHz.

↥

LoadEmSeq (Sequence, Nb of channels, Wave(t,ch));

Description: Loads a transmitter curve in the selected sequence on all active channels. The curve is a matrix (nb of samples, nb of active channels) and must be is normalized between –1 and +1. The sample (1,1) of the matrix is the first emitted sample on channel 1. It is recommended that the first and the last samples of the curve have the value 0.

Parameters:

Sequence: Sequence number(0 to 255).

Nb of channels: Number of active channels of OPEN system.

Wave[samples,cha]: Matrix (nb of samples, nb of active channels) containing the wave (between –1 and +1).

↥

LoadRetSeq (Nb of channels, Delays[]);

Description: Loads the receivers delays for all channels and all sequences. The unit is 1/num frequency. The table is composed with NbOfChannels^2 delays.

The first nb channels delays are sent to channel one sequence 0 to n channels -1, the second nb channels delays are sent to channel 2 sequence 0 to n chanel-1, and so on.

Parameters:

Nb of channels: Number of active channels of OPEN system.

Delays[]: Matrix (NbOfChannels, NbOfChannels) containing the delays (in steps of 1/numfrequency).

↥

Dac (Dac on,Default gain,Smooth on,Channel,Positions[](µs), Gains[](dB));

Description: Loads a DAC curve on the selected channel. In fact all 8 channels of the board corresponding to selected channel are affected by this command. (board=1+floor((channel-1)/8)).

Parameters:

Dac on: When 0 sets the DAC curve off, when 1 sets it on.

Default gain: Default gain (0 to 79.9 dB) before position 0.

Smooth on: Enables (1) or disables (0) the smoothing of the DAC curve.

Channel: Channel number (1 to n).

Positions[],Gains[]: List of values containing the DAC curve [Position1, Pos2,...,PosN, Gain1, Gain2,...,GainN]. Position unit is µs. Position origin is the begining of the acquisition.
Max size of the table is 256 values
Max position value is 169 µS

Example:

attenuation=0.06; % typical attenuation value (dB/mm/MHz) (medical)
c=1.54; % P waves celerity (mm/us)
rsf=80; % receiver sampling frequency (MHz)
f0=3.5; % emitter central frequency (MHz)

```
N=8; % nb of subdivisions
Gain=40; % default gain (dB)
nbsamples=4000; % nb of samples of the acquisition
pos0=0; gain0=Gain; % initial position (us) and gain (dB)
Nbchannels=64; % nb of channels
posN=nbpts/rsf; %final position (us)
depth=posN*c/2; % depth (mm) (echography)
gainN=gain0+ attenuation*f0*depth; %final gain (dB)
pos=pos0+[0:N-1]*(posN-pos0)/N; % positions list (us)
gain_tgc=gain0+[0:N-1]*(gainN-gain0)/N; % gains list (dB)
for j=1: Nbchannels /8;
   LecoeurUS('Dac',1,Gain,1,j,pos,gain_tgc);
end;
```

↥

FIR (Coefficients[8]);

Description: Loads 8 new coefficients in the FIRs for all channels.

Parameters:

Coefficients[]: Table in which are the 8 coefficients. Clock frequency is the sampling frequency for receivers.

Example:

L'exemple ci-dessous retourne une erreur à l'appel de FIR. Pourquoi ?

```
% Raised cosine lowpass FIR filter
N=8; % nb of coefficients
cutoff=10; % cutoff freq (MHz)
bsf=80; % base sampling frequency (MHz)
rolloff=0;
b = firrcos(N-1, cutoff, rolloff, bsf, 'rolloff'); freqz(b,1);
LecoeurUS('FIR', b)
```

↥

SetTransDelay (Channel, Sequence, Delay in ns);

Description: Sets the delay of the selected channel pulser. The delay is only active on the selected channel.

Parameters:

Channel: Channel number (1 to n).

Sequence: Sequence number (0 to 255).

Delay value: Delay in ns.

Example:

```
LecoeurUS('SetTransDelay',1,0,100);
```

Sets the delay of channel 1 pulser to 100 ns.

↥

GlobalSendDelays (Nb of sequences);

Description: Sends to all boards the pulsers and receivers delays of all the sequences (Refer to **Excel Global set-up file description** to locate curves files on the hard disk).

Parameters:

Nb of sequences: Number of sequences to load (1 to 256).

Example:

LecoeurUS('GlobalSendDelays',64);

Sends delays of pulsers and receivers in all channels for 64 sequences.

↥

FightOnOff (On/Off(1/0),Channel);

Description: Switches On/Off the selected pulser.

Parameters:

On/Off: 1 to set on the pulser, 0 to switch it off.

Channel: Channel number (1 to n).

Example:

```
LecoeurUS('FightOnOff',0,1);
```

Switches off the pulser of channel one.

↥

SoftTrigger (On/Off(1/0));

Description: Sets the trigger controlled by software. All pulsers are off until a digitalization is requested.

Parameters:

On/Off: 1 to set on the Software trigger, 0 to switch it off.

Example:

```
LecoeurUS('SoftTrigger',1);
```

Switches on the software trigger.

↥

SetNbChannel (Nb of channels);

Description: Sets the number of channels in use in OPEN.

Parameters:

Nb of channels: Number of channels in use (1 to 256) beginning by the first channel.

Example:

```
LecoeurUS('SetNbChannel',16);
```

The number of channels in use is now 16.

↥

ExternTrigger (On/Off(1/0));

Description: Sets the trigger input source to internal or external. In external mode; a TTL pulse must be input in the socket named Trigger IN at the back side of OPEN box.

Parameters:

On/Off: 1 or 0 to select or disable external trigger input source.

Example:

```
LecoeurUS('ExternTrigger',1);
```

Open system is currently waiting for an external signal.

↥

Calibrate (Channel(all=0));

Description: Makes a calibration of a designated channel or of all channels.

Parameters:

Channel: 1 to n; passing 0 calibrates all channels.

Example:

LecoeurUS('Calibrate',0);

Makes a calibration for all channels.

↥

Help ('Function name');

Description: Gives information about a specific function.

Parameters:

Help('Function Name'): The list of available functions is obtained by passing no Function Name.

Examples:

LecoeurUS('Help');

Returns a list of the available functions.

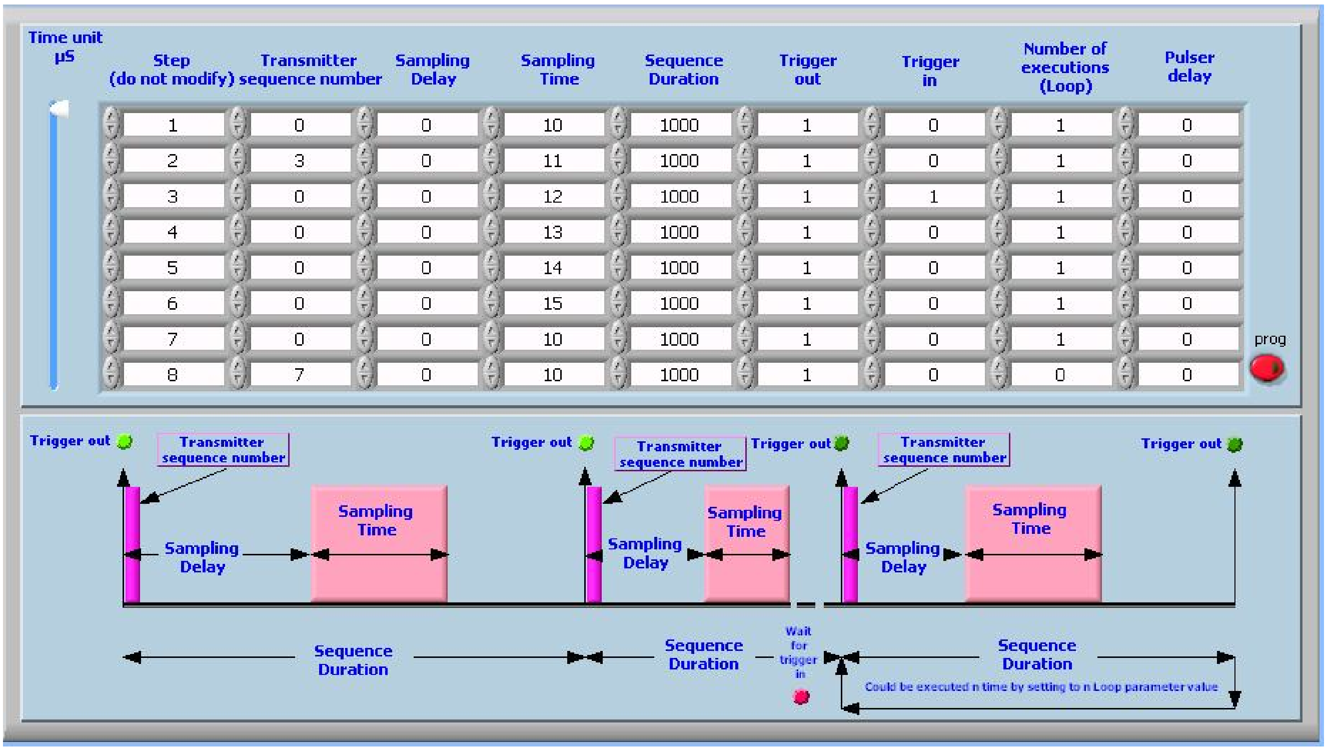
LecoeurUS('Help','Gain');

Returns the description of 'Gain' function.

Real-Time Micro-Sequencer functions

This part of the OPEN system is dedicated to real time applications. Using the micro sequencer allows to program your own emission/ reception/ digitalization/ timings. After setting-up the micro sequencer the ‘RUN’ command executes the program, outputs external trigger, wait for an input trigger, change the sequences and store in internal memory the samples that are collected during the execution. After a 'RUN' the memory can be read and data downloaded from OPEN system.

DESCRIPTION OF THE MICRO SEQUENCER



Step (do not modify)	Transmitter sequence number	Sampling Delay	Sampling Time	Sequence Duration	Trigger out	Trigger in	Number of executions (Loop)	Pulser delay
1	0	0	10	1000	1	0	1	0
2	3	0	11	1000	1	0	1	0
3	0	0	12	1000	1	1	1	0
4	0	0	13	1000	1	0	1	0
5	0	0	14	1000	1	0	1	0
6	0	0	15	1000	1	0	1	0
7	0	0	10	1000	1	0	1	0
8	7	0	10	1000	1	0	0	0

The micro sequencer works as a processor, a program is up-loaded in OPEN and a run time executes this program. Each line of the program is an ***instruction*** *which describes the action to do.*

Transmitter sequence number

This parameter indicates the number of the sequence use when executing the instruction. If the pulser is analog the sequence indicates a wave form, if the pulser is pulsed it set-up a delay. This parameter has the same value for the 8 channels of the same board.

Sampling delay

This parameter is the delay between the begining of transmission and the begining of digitalization. This parameter has the same value for the 8 channels of the same board.

Sampling duration

This duration is the digitalization duration. During this, the signal of all channels is digitalized and stored in memory. This parameter has the same value for the 8 channels of the same board.

Sequence duration

The sequence duration is the time to wait before executing the next instruction. It is generally called PRF period.

Trigger Out

If this bit is set to one, OPEN system will generate a pulse on the 'trigger-out' output, at the beginning of the instruction. This pulse is a TTL one the duration of this pulse is 50 ns.

Trigger In

If this bit is set to one, OPEN system will wait for an input pulse on the 'Trigger IN' input before it executes the instruction.

Number of Executions

This field of bits is a loop generator. The instruction will be executed N times. N is a value between 1 and 255.

Pulser delay

This parameter is only active with the option ‘analog pulser’ of OPEN. The waveform is shifted right in the time base. The value of this shift is the parameter value. This parameter has the same value for the 8 channels of the same board.

↥

Upload_Seq_Duration (Durations[](12.5ns));

Description: Up-loads the table of Sequence duration to the micro sequencer. The first line of this table is the first instruction of the micro-sequencer program.

Parameters:

Durations[]: Table of Sequence duration to up-load in steps of 12.5 ns.
Max value for the duration is 2^32 * 12.5 nS : 53 687 091,22 µS ~53 s

Example:

LecoeurUS('Upload_Seq_Duration',20000,20000,20000,20000,20000,20000,2000,2000);

Up-load the table of 'sequence duration'. In this example the micro sequencer program has 8 instructions. The duration of the 6 first sequences is 250 µs, then 25 µs for the two last.

↥

Upload_Seq_Num_Delay (Card nb(1 to n),Delays[](12.5ns));

Description: *Up-load the table of the delays between beginning of transmission and beginning of digitalization to the micro sequencer. The first line of this table is the first instruction of the micro-sequencer program and so on.*

Parameters:

Card nb: number of the board in which the table will be up-loaded. First card is nb 1.

*Delays[]: table of delays to up-load in steps of 12.5 ns. Minimal value is 0. Max value for delay is 2^16 * 12.5 nS :* $819.2\ \mu S$

Example:

LecoeurUS('Upload_Seq_Num_Delay',1,20,200,200,200,200,200,200,200);

Up-loads the table of delays. In this example the micro sequencer program has 8 instructions. The first sequence has a delay of 25 ns, the following delays are 250 ns. The board number in which the table is up-loaded is 1.

↥

Upload_Seq_Num_Width (Card nb(1 to n),Widths[](12.5ns));

Description: Up-loads the table of the 'sampling duration' to the micro sequencer. The first line of this table is the first instruction of the micro-sequencer program and so on.

Parameters:

Card nb: number of the board in which the table will be up-loaded. First card is nb 1.

Widths[]: table of 'sampling duration' to up-load in steps of 12.5 ns. Minimal value is 0. Valeur Max ?

Example:

LecoeurUS('Upload_Seq_Num_Width',1,200,200,200,200,200,20,20,20);

Up-loads the table of 'sampling duration'. In this example the micro sequencer program has 8 instructions. The first sequence has a sampling duration of 2.5 µs. The board number in which the table is up-loaded is 1.

↥

Upload_Seq_Pulser_Delay (Card nb(1 to n),Delays[](12.5ns));

Description: *Up-loads the table of the 'pulser delay' to the micro sequencer. The first line of this table is the first instruction of the micro-sequencer program and so on.*

Parameters:

Card nb: number of the board in which the table will be up-loaded. First card is nb 1.

Delays[]: table of 'pulser delay' to up-load in steps of 12.5 ns. Minimal value is 0. Valeur Max ?

Example:

LecoeurUS('Upload_Seq_Pulser_Delay',1,200,0,0,0,0,0,0,0,0);

Up-loads the table of 'pulser delay'. In this example the micro sequencer program has 8 instructions. The first sequence has a pulser delay of 2.5 µs. The board number in which the table is up-loaded is 1.

↥

Upload_Seq_Pulser_Seq (Card nb(1 to n),Sequences[]);

Description: Up-loads the table of the 'pulser sequence' to the micro sequencer. The first line of this table is the first instruction of the micro-sequencer program and so on.

Parameters:

Card nb: number of the board in which the table will be up-loaded. First card is nb 1.

Sequences[]: table of 'pulser sequence' to used (each between 0 and 255).

Example:

LecoeurUS('Upload_Seq_Pulser_Seq',1,0,0,0,0,0,0,0,0);

Up-loads the table of 'pulser delay'. In this example the micro sequencer program has 8 instructions. All these instructions use the pulser sequence 0. The board number in which this table is loaded is 1.

Upload_Seq_Program (Loop&Triggs[]);

Description: *Up-loads the table in which the program for the micro sequencer is. Each line of the table contains:*

- *Loop parameter*
- *Trigger in*
- *Trigger out*
- *End program*

Each line is two bytes long. The less significant byte contains the loop parameter. Only the 3 most significant bits of most significant byte are used:

MOST SIGNIFICANT BYTE:

B7 -> Not End bit (if this bit is 0 the program stops)
B6 -> Trigger out (if this bit is 1 a pulse is generated on the 'trigger out' output of open just before the instruction is executed). When loop>1, trigger out is generated at each iteration of the instruction.
B5 -> Trigger in (if this bit is 1 the micro sequence program will wait for a pulse on the 'trigger in' input of OPEN before executing the instruction. When loop>1, trigger in is expected at each iteration of the instruction.
B4->B0: not used

LESS SIGNIFICANT BYTE:

B7 ->B0: loop value between 1 and 255. IE when loop is 2, instruction is executed twice.

Parameters:

Loop&Triggs[]: Program of the micro sequencer.

Example:

LecoeurUS('Upload_Seq_Program', hex2dec('C000')+1, hex2dec('D000')+2, 0);

Up-loads the table of ‘pulser delay’. In this example the micro sequencer program has 3 instructions:

First one: C001 in hexa -> ‘loop’ parameter is 1, ‘Not End’ parameter is 1, ‘Trigger out’ parameter is 1, ‘Trigger in’ parameter is 0.

Second one: D002 in hexa -> ‘loop’ parameter is 2, ‘Not End’ parameter is 1, ‘Trigger out’ parameter is 1, ‘Trigger in’ parameter is 1.

Third one: 0 is 0 in hexa -> ‘loop’ parameter is 0, ‘Not End’ parameter is 0, ‘Trigger out’ parameter is 0, ‘Trigger in’ parameter is 0 -> THE PROGRAM STOPS.

↥

Exe_Sequencer (Starting address);

Description: *Runs the micro sequencer program. Stores samples starting at the memory location 'starting address'. It is very important to wait for the completion of the execution before invoquinq any OPEN system command.*

Parameters:

Starting address: Starting memory location at which samples will be stored. This value is limited by the memory capacity available in your OPEN system.

Example:

```
LecoeurUS('Exe_Sequencer',0000); pause(whole_sequence_duration);
```

Runs the micro sequencer program previously up-loaded and store samples at the memory location starting at 0 (beginning of the memory), then wait for the completion of the execution.

↥

RF(t)=Read_Memory (Nb of samples+18, Channel, Starting address, Delay law);

Description: Reads the memory of the selected channel after a run is executed.

Parameters:

Number of samples: Size of the block to read. Maximum value for a read is 65535. Larger read must decomposed in smaller ones and starting address choosen consequently.

Channel: Number of the channel to read (1 to n)

Starting address: Beginning of the memory location to read. This value is limited by the memory capacity available in your OPEN system.

Delay law: When 0 returns the samples without delay law applied. When 1 to 255 returns the sum of the delayed samples of the 8 channels of the board. Value is the number of the delay law of the sequence.

Example:

```
plot(LecoeurUS('Read_Memory',15000+18,1,0,0));
```

Reads a bloc of 15000 samples from the memory of channel 1. The starting address is 0 , and no delay law is applied.

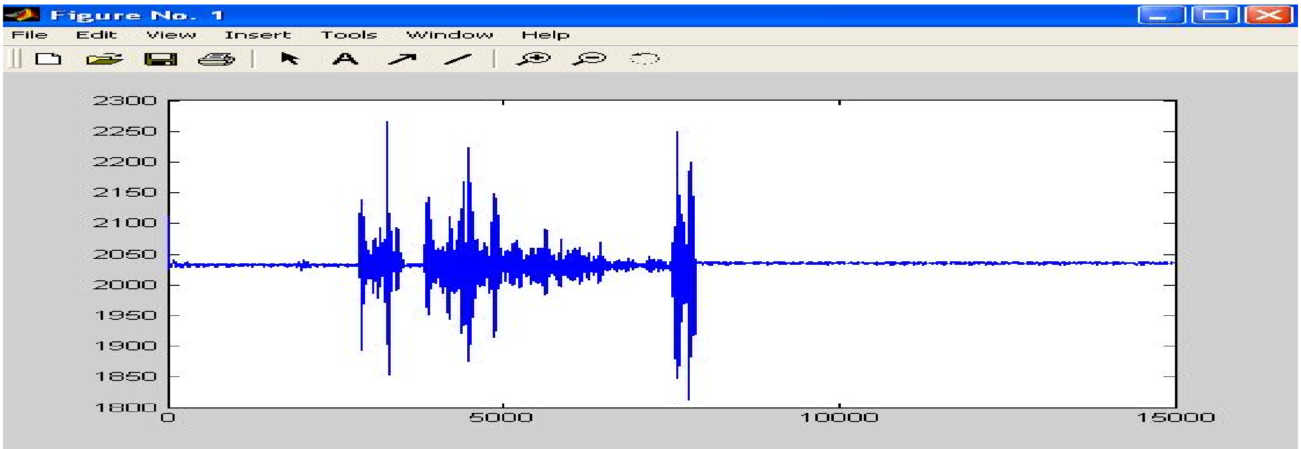


Figure No. 1
File Edit View Insert Tools Window Help
2300
2250
2200
2150
2100
2050
2000
1950
1900
1850
1800
0
5000
10000
15000

↥

2) Excel Global set-up file description

The complete set-up of your OPEN system is contained in an Excel file located in: c:\saphirp\brainscan.xls.
Here are some useful parameters description contained into this file. Each time the function 'Init' is used the set-up is up-loaded in the OPEN UT boards. This means effect of modifying brainscan.xls will only be applied after calling 'Init' function.

2.1) Managing sequences

Open system is able to store up to 128 sequences. Each sequence contained a transmitter wave for each channel, and a receiver delay for each channel.

A sequence is made active by calling function 'Setseq'.
The column 'Q' of branscan.xls contains the name and directory of a file for each channel

Example: c:\saphirp\Brain trans seq\Lec0.txt

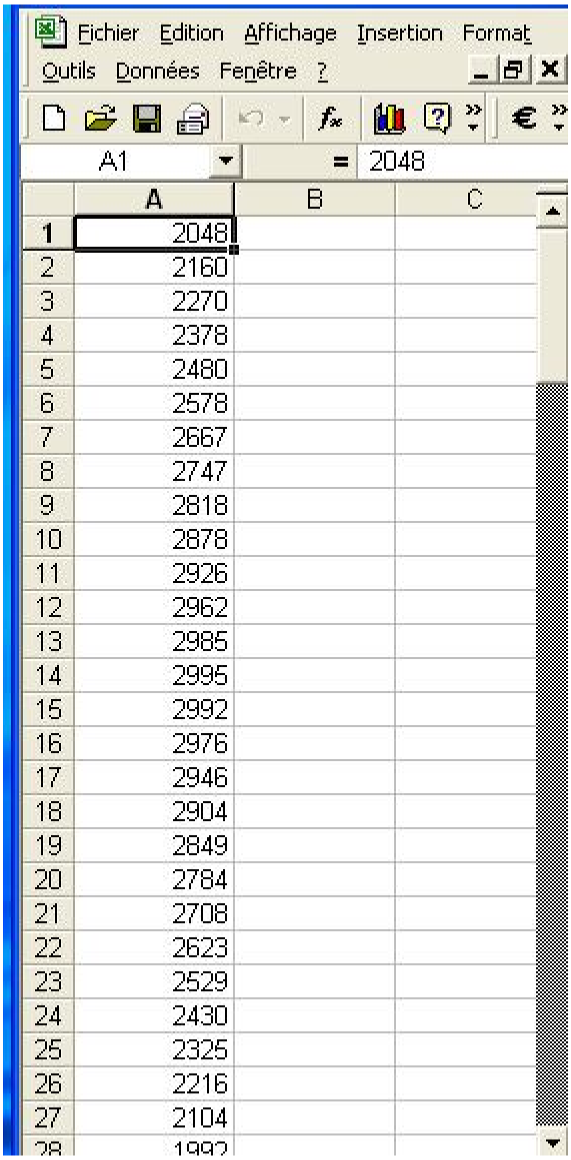
The file Lec0.txt contains names and directories of transmitter Waveform files.

Example (Lec0.txt):

c:\saphirp\analog_pulser\motif1.xls
c:\saphirp\analog_pulser\motif2.xls
c:\saphirp\analog_pulser\motif3.xls
c:\saphirp\analog_pulser\motif4.xls
c:\saphirp\analog_pulser\motif5.xls
…..

c:\saphirp\analog_pulser\motif64.xls
The first line of this file is for the sequence number 0
The second for sequence 1 and so on.

If we open the file Motif3 for example we find a list of numbers between 0 and 4096. This is the shape of the transmitter wave that will be activated when sequence 3 is selected.



A1 = 2048

	A	B	C
1	2048		
2	2160		
3	2270		
4	2378		
5	2480		
6	2578		
7	2667		
8	2747		
9	2818		
10	2878		
11	2926		
12	2962		
13	2985		
14	2995		
15	2992		
16	2976		
17	2946		
18	2904		
19	2849		
20	2784		
21	2708		
22	2623		
23	2529		
24	2430		
25	2325		
26	2216		
27	2104		
28	1992		

For the receiver delays the file which describes the sequences is contained in column ‘P’ of braiscan.xls. This file is a text file. It is composed with a list of *numbers that are the delays to apply for sequence 0 (first line) to sequence n (Line n+1)*

Example:

Column P channel 1: c:\saphirp\Brain delay seq\dloa001.txt

Delay sequences for listed in the file dloa001.txt

Listing of the file dloa001.txt:

```
5999
5993
5977
5949
5910
5860
5799
5728
5645
5552
5447
5332
5207
5071
4925
4768
4601
4424
4238
4041
3835
3620…
```

When sequence 0 will be activated by 'Setseq' function the delay of 5999 ns will be apply on channel 1, sequence 1-> 5993 ns will be applied on channel 1, and so on.

The function 'Brcscan' produces an image composed with the sum of all channels. The first line is sequence 0, the second sequence 1,….

3) Matlab program examples

A simple B-SCAN acquisition

```
% Simple B-Scan

nb_periods=2;  % nb of periods of the emitted sine wave
Ne=64; % nb of array elements
f0=3.5; % array central frequency (MHz)
deads=[]; % array dead channels
esf=80; % emitting sampling frequency (MHz)
amplitude=0.99; % amplitude of emission
Gain=70; % (dB) max 79.9 dB
Delay=10; % (us)
Nbsamples=4000;

%LecoeurUS('Init_USB'); % init USB
LecoeurUS('Init',1,64,1); % full init
Nt=round(nb_periods*esf/f0);
t=(0:1:Nt-1)/esf;
s0=amplitude*sin(2*pi*f0*t)';
s0=[zeros(1,1); s0; zeros(1,1);]; % first and last points set to 0
Nt=length(s0);
for i=1:Ne;
  wave(:,i)= amplitude*s0;
end;
wave (:,deads)=0;
figure(1); imagesc(wave);title('emitted wave');
LecoeurUS('LoadEmSeq',0,Ne,wave); % loads plane waveform in bank 0
LecoeurUS('Setseq', 0); % select bank 0

for j=1:Ne;
  LecoeurUS('Gain',Gain,j);
  LecoeurUS('Retard', Delay,j);
end; clear j;

RF=LecoeurUS('Bscan', Nbsamples,1,1);
figure(2);imagesc(abs(hilbert(RF)));title('simple bscan');
```

↥

A simple acquisition using the micro-sequencer

```
% A simple acquisition using the micro-sequencer

nb_periods=2;  % nb of periods of the emitted sine wave
Ne=64; % nb of array elements
f0=3.5; % array central frequency (MHz)
deads=[]; % array dead channels
esf=80; % emitting sampling frequency (MHz)
amplitude=0.99; % amplitude of emission
Gain=70; % (dB) max 79.9 dB
Delay=10; % (us)
Nbsamples=4000;

%LecoeurUS('Init_USB'); % init USB
LecoeurUS('Init',1,64,1); % full init
Nt=round(nb_periods*esf/f0);
t=(0:1:Nt-1)/esf;
s0=amplitude*sin(2*pi*f0*t)';
s0=[zeros(1,1); s0; zeros(1,1);]; % first and last points set to 0
Nt=length(s0);
for i=1:Ne;
  wave(:,i)= amplitude*s0;
end;
wave (:,deads)=0;
figure(1); imagesc(wave);title('emitted wave');
LecoeurUS('LoadEmSeq',0,Ne,wave); % loads plane waveform in bank 0

Nboards=Ne/8; % nb of boards
address=0; % acquisition starting address
bsf=80; % base sampling freq. (MHz)
rsf=80; % receiver sampling freq. (MHz)

for j=1:Ne; LecoeurUS('Gain',Gain,j); end; clear j;

LecoeurUS('Upload_Seq_Program',hex2dec('C001'),0); % trigger out, loop=1
LecoeurUS('Upload_Seq_Duration', Nbsamples*bsf/rsf+Delay*bsf+Nt*bsf/esf);
for i=1:Nboards;
  LecoeurUS('Upload_Seq_Num_Width',i,Nbsamples*bsf/rsf);
  LecoeurUS('Upload_Seq_Pulser_Delay',i,0);
  LecoeurUS('Upload_Seq_Pulser_Seq',i,0); % using bank 0
  LecoeurUS('Upload_Seq_Num_Delay',i,Delay*bsf);
end;
LecoeurUS('Exe_Sequencer',address); % execution
pause (0.015); % waiting for the end of the execution

for j=1:Ne;
  RF(:,j)=LecoeurUS('Read_Memory',Nbsamples,j,address,0);
end;
figure(2);imagesc(abs(hilbert(RF)));title('acquisition using micro-sequencer');
```

LECOEUR ELECTRONIQUE

19 route de Courtenay - 45220 – CHUELLES –
☎: 02 38 94 27 30– 🖷: 02 38 94 29 67
💻: info@lecoeur-electronique.com