LECOEUR ELECTRONIQUE
300 Chemin des Comtois
45220 CHUELLES - FRANCE -
Tel : +33 (0)2 38 94 28 30
Fax : +33 (0)2 38 94 29 67

mailto:info@lecoeur-electronique.com
http://www.lecoeur-electronique.com

Rédacteur	*S.BERTOLOTTO*
Date	*17/09/2018*

Révision 2 le 17 Septembre 2018

1 MENU

2 I/O

INPUT	OUTPUT
	Code
ProductNumber	
ChannelNumber	
Function	
In1	
In2	
In3	
In4	
In5	
In6	
	Out1
	Out2
	Out3
	Out4
	Out5
	Out6
Array (tableau)	
Array lenght	

3 DECLARATION

```
long __cdecl ApintUsb(unsigned long ProductNumber,
        unsigned long ChannelNumber, char Function[], double In1, double In2,
            double In3, double In4, double In5, double In6, short int *Out1,
            short int *Out2, short int *Out3, short int *Out4, short int *Out5,
    short int *Out6, unsigned short int Array[], long *len);
```

Note: DLL was compiled in C standard call. For Visual Basic language, DLL must be recompiled in standard call.

4 DESCRIPTION

First of all: ProductNumber and ChannelNumber are 2 parameters which must be set each time to :

- 0 = USBox
- **1** = **US-Key**
- 0..7 = For an 8 multi channel system*

Note:

- Up case letter and space are important
- **Code** return 1 is function is recognized else 0. It's not an error code

Menu

Function	*Description*	*Input*	*Output*
"RunExeX32"	**BEFORE EVERYTHING You must call this function**	**To load the TSR for x32 to x64 conversion** **ONLY for x64 DLL**	
"KillExeX32"	**The last function to call before exit**	**To kill the TSR for x32 to x64 conversion** **ONLY for x64 DLL**	
"Init usb", "Usb init", "init usb", "usb init"	Initialize the USB2 connection	Attention, it is necessary to call this function as each times as there are ways passing the number of the way to initialize (0..7)	
"Load configuration", "load configuration"	Loads a default configuration contained in c: \ saphirp \ ustcad	In # 1 = Nb total of channels Attention, it is necessary to call once this function with the total number of lanes. Configurations come from 1V, 1D, 2V, 2D, 3V, 3D files, etc.	
~~"Channel", "channel"~~	~~Set the current channel~~	*	
"Id code", "id code"	Hardware's code reading		Out # 1
"Prf", "prf"	PRF adjustment	In # 1 = PRF (kHz)	
"Echo-start", "echo-start"	Echo-start inhibition	In # 1 = Echo-start position (µs) In # 2 = Echo-start width (µs) = In # 1 = 0 => echo-start OFF	
"Pulse delay", "pulse delay"	Delay	In # 1 = Pulse delay in µs (***n****1.6µs)	
"Filter/Mode", "filter/mode"	Filter setting and single/double crystal Mode	In # 1 = 0 =1.25MHz = 1 = 2.5MHz = 2 = 5MHz = 3 = 10MHz = 4 = Broad band In # 2 = 0 : emetter / receiver strapped = 1 : emetter / receiver disconnected	
"Gain", "gain"	Gain adjustment	In # 1 = gain (dB)	
"Voltage", "voltage"	Pulse voltage adjustment	10 ≤ In # 1 ≤ 230 (V)	
"Width", "width"	Pulse width adjustment	0 ≤ In # 1 ≤ 255 In#2 : Wave train ON(1) OFF(0) In#3 : Number of pulses	
"Echo-start mode", "echo-start mode"	echo-start on/off + polarity	In # 1 = 0 negative wave / 1 positive wave In # 2 = 0..100 (%) echo-start threshold	
"Scale delay", "scale delay"	Delay adjustment	In # 1 = delay (µs)	
"Wave", "wave"	Select Wave to control	In # 1 = 0 rectified / 1 negative / 2 positive	

"Gate position", "gate position"	Gate position	In # 1 = Gate number 1..3 In # 2 = Gate position (µs)	
"Gate width", "gate width"	Gate width	In # 1 = Gate number 1..3 In # 2 = Gate width (µs)	
"Gate hight", "gate hight"	Gate height	In # 1 = Gate number 1..3 In # 2 = Gate height (%)	
"Relays", "relays"	Alarm on/off	In # 1 = bit0 : Gate1 / bit1 : Gate2 / bit2 : Gate3 bit value : 0 alarm on appear. / 1 disapp	
"Alarm filter", "alarm filter"	Strike before alarm	In # 1 = Gate number 1..3 In # 2 = Number of shots before alarm	
"Measures", "measures"	Gates measures No alarm = 0		Out # 2= alarme1 (MSB), amplitude1 (LSB) Out # 3 = alarme2 (MSB), amplitude2 (LSB) Out # 4 = alarme3 (MSB), amplitude3 (LSB) Out # 5 = distance1 (step of 12.5ns) Out # 6 = distance2 (step of 12.5ns) Out # 1 = distance3 (step of 12.5ns)
"A-scan", "Ascan", "a-scan", "ascan"	Get a-scans coming from the 12bit analog to digital converter	In # 1 = 0 HF / 1 A-scan In # 2 = Retentivity display (0..255) In # 3 = Number of samples In #4 = A-scan wave : 0 full rectified / 1 negative / 2 positive	Array Out#1≠ 0 → timeout, NO A-scan available
"Help", "help"	Function list		Array U8 (ASCII code)
"Version", "version"	DLL Version		Out#1
			Out#1 : LSB Out#2 : MSB
"SamplingFreq/Mode", "samplingfreq/mode", "samplingfreqmode"	Change the sampling frequency	In#1 : 0=160 1=80 2=40 3=20 ~~4=10MHz~~ In # 2 = 0 : emetter / receiver strapped = 1 : emetter / receiver disconnected ⓘ If frequency is different of 80MHz, filter is set to broadband and DAC is stopped	
"Scale","Scale A-scan counter"	Define the number of samples of the frame	In#1 : Step of ***25ns*** Note : It can be different of the A-scan *§ A-scan counter function*	
"Dac", ""dac"	Program the DAC curve	In#1 : 0=Dac OFF / 1=Dac ON Array=courbe DAC en 1/10dB !!!! *§Dac curve function*	
"Very Fast Ascan" (<4000samples, HF, PRF>2kHz)	Get a-scans coming from the 12bit analog to digital converter	In #1 = Number of samples In #2 = turbo 0=OFF / 1=ON (without any timeout)	Array Out#1=1 timeout no A-scan
"Start Sampling"	Ask a sampling		
"Wait Sampling"	Wait the end of sampling		***Code*** : 0=in process / 1=finished
"Read Samples"	Read samples	In #1 = Number of samples	

5 DAC CURVE FUNCTION

The USBoxS DAC curve is 166µs depth.
It programmable by 256 step of 650ns.

To program it, send 256 gain values (1/10dB) through "Array" and set 1 in "In#1"
Ex: 10.1dB => 101

Take care, if you want to adjust the gain after programming a Dac curve, don't use "Gain" function but reprogram a new Dac curve with an offset.
Using "Gain" function while a Dac curve will set it off !

Over the 166µs, the gain of "Array[0]" is reprogrammed.

The 1st point of the curve, "Array[0]", correspond to the emission or interface echo if you are working with "echo-start" (Original sync).
In this last case, "Array[0]" is the gain during the water path (between emission to interface echo)

Menu

6 ATTENTION

- ***Code*** *is 1 only if the string (function) is recognized. It's not an error code concerning a good processing*

- ***Array*** *is an array declared as Unsigned Integer (uInt16) and* ***Len*** *its size. Before calling a function, you must declare* ***Array*** *with a good size. For example if you want to read an A-scan of 1000 samples, you must declare* ***Array[1000] and Len=1000****.*

- ***Array*** *is an array declared as Unsigned Integer(uInt16) however "A-scan" return integer (uInt16) and* ***"A-scan counter" return unsigned short integer (uInt8) codded on uInt16.*** *It's always unsigned. As the pointer is always on uInt16 you must cast differently Array following functions*

- *"****Init usb"*** *If you use only one USBoxS, you must call "Init usb" with* ***ChannelNumber****=0. If you use 2 USBoxS, you must call one time "Init usb" with* ***ChannelNumber****=0 AND one time with* ***ChannelNumber****=1. 1st channel is the USB2 plug connected*

- *"****Filter/Mode"*** *set automatically the sampling frequency to 80 MHz*

↕

- ***"SamplingFreq/Mode"*** *set automatically filter to Broadband (No filter)*

- ***"Load configuration"*** *program completely 1,2,3,4… USBoxS based on default parameters files c:\saphirp\ustcad\(1,2,3,4…)V*

  In#1=2 → *USBoxS* ***0*** *is initialised with default parameters c:\saphirp\ustcad****1V***
  → *USBoxS* ***0*** *is initialised with default DAC curve c:\saphirp\ustcad****1D***
  → *USBoxS* ***1*** *is initialised with default parameters c:\saphirp\ustcad****2V***
  → *USBoxS* ***1*** *is initialised with default DAC curve c:\saphirp\ustcad****2D***

  These files are managed by US_Key.EXE