mtec wiki

Cephasonics Ultrasound

System

The Cephasonics Ultrasound system consists of 2 systems (4x 64 channel box Griffin system + 2x 64 channel box Cicada system), which can be used simulteanously or separately from each other. The second system (2x 64 Cicada) will be delivered in summer 2019.

TODO - Specs - Grafik System Komponenten

Starting US System

First, turn on the PCiE Switch (if you are using the PC inside the rack), the computer and after the PC has booted turn on the Griffin/Cicada system. Otherwise the PCIe connection won´t be setup correctly.

PCIe connection has to be enabled at each startup or mode change of the system (PCIe connection is necessary for high speed imaging in CC-Mode):

1) turn on all system components (PCIe-switch in case of the PC inside the rack, PC, Cicada/Griffin)

2) Execute

```
pcie_routine
```

optional 3) Execute

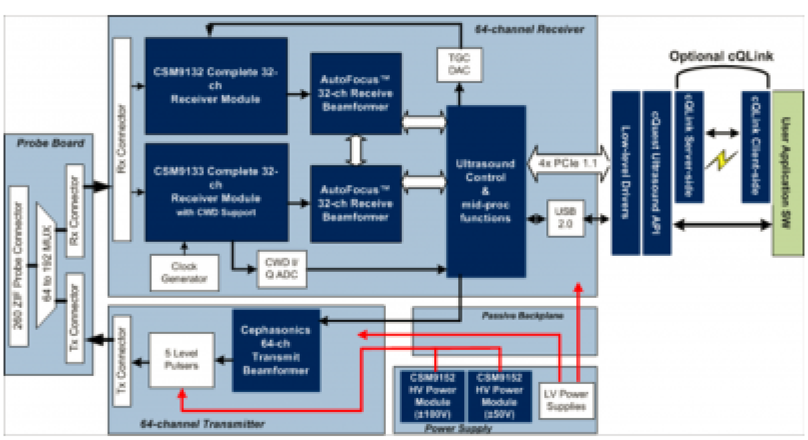
```
test_throughput
```

(PCIe throughput should be > 140 Mb/s)

If you don't see that throughput, refere to pcie troubleshooting

Components

Cephasonics System Architecture Overview



The architecture of each Cephasonics 64-channel box is sketched in the figure above. The elements of the connected probes are physically connected to a switching element which enables a connection either to the TX or RX part of the 64-channel box (Probe Board).

64-channel Transmitter Within the transmit part of the box the transmit beamformer can be adapted using the functions in the Cephasonics CUSDK. Time delays per channel or focus depth are examplary parameters to be specified in the functions.

64-channel Receiver

B-Mode Imaging

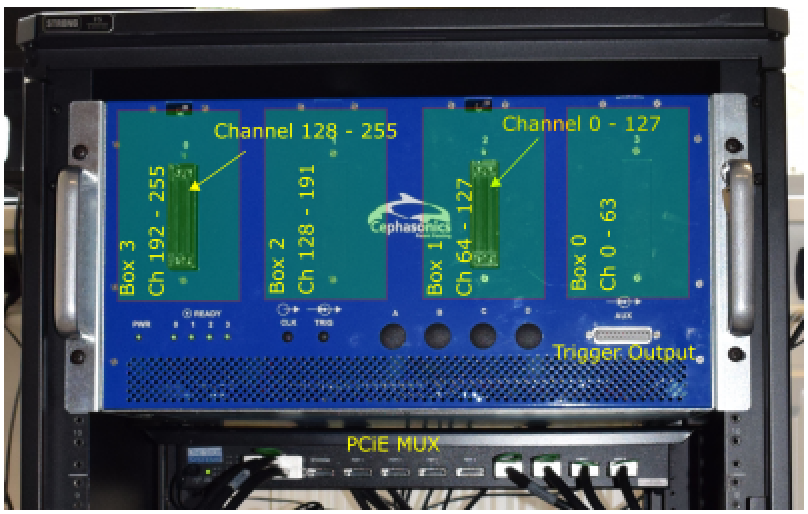
Ultrasound channel data is delayed, summed, filtered to form an ultrasound beam.

Channel Capture Mode Imaging

Raw ultrasonic RF channel data is captured.

Power Supply It is possible to decide between two different power supply modes (+-50 or +-100 V) for TX/RX of the ultrasound waves.

Griffin System

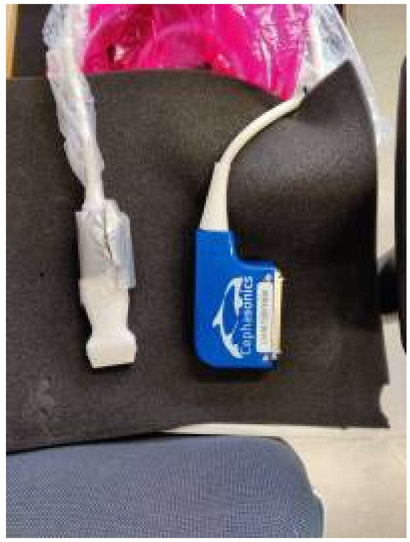


Cicada System

Ultrasound Transducers

Ultrasound Transducers are very expensive and very fragile. Therefore they need to be handled very carefully as if they were made out of glass. The following transducers are available (* - probes will be delivered in summer 2019):

Name	CPPA8027	CPLA12810A	Vermon 2D Probe
Type	Phased Array	Linear Array Probe	Matrix Transducer
Adaptor	Cephasonics	Cephasonics	Cephasonics
Center Frequency	2.8MHz	10MHz	3MHz
Elements	80	128	16×16
Pitch	0.254mm	0.21mm	300um

Focus	60mm	30mm	-
Image			

Name	L14-5/38 #1	L14-5/38 #2	L14-5/38 #3
Type	Linear Array Probe	Linear Array Probe	Linear Array Probe
Adaptor	Ultrasonix	Ultrasonix	Ultrasonix
Center Frequency	5-14MHz	5-14MHz	5-14MHz
Elements	128	128	128
Pitch	0.3mm	0.3mm	0.3mm
Focus	16mm (geo.)	16mm (geo.)	16mm (geo.)
Image			

Name	C5-2/60	MC9-4/12	EC9-5\10
Type	GPS Convex Transducer	Microconvex Transducer	Endovaginal Microconvex Transducer
Adaptor	Ultrasonix	Ultrasonix	Ultrasonix
Center Frequency	2-5MHz	4-9MHz	5-9MHz

Elements	128	128	128
Pitch	0.47mm	0.23mm	0.21mm
Focus	50mm (geo.)	35mm (geo.)	23mm (geo.)
Image			

Name	CPLA12875	xxx	xxx
Type	Linear Array	xxx	xxx
Adaptor	Cephasonics	xxx	xxx
Center Frequency	7.5MHz	xxx	xxx
Elements	128	xxx	xxx
Pitch	0.29mm	xxx	xxx
Focus	50mm	xxx	xxx
Image			

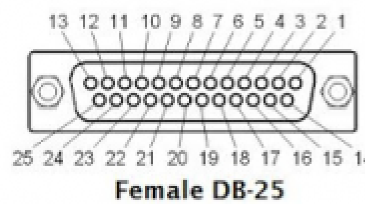
The transducers/probes has to be connected to the US system with their cable facing downwards. A mechanical switch on top of the connectors provides a physical rigid connection of the probes (left - lock, right - unlock). Important! Do not leave the switch in locked position when the connector is not connected - electrical connection PINs might get dirty.

Probehunter Connector

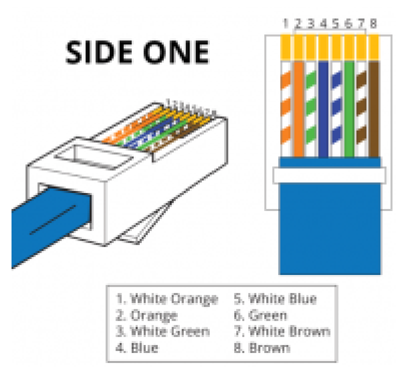
Custom made connector to connect Ultrasonix Probes to the Cephasonics system. Simply connect the adapter to the cephasonics and the Ultrasonix probe into the adaptor.

Triggers

Trigger connection is located at front on DB25 pin board (Griffin).

Pin Combination	Properties	Description	Image
1(GND) [brown] & 14 [red]	Out 0	Beam start 3.3V 50Ω, High = Beam Start	
2(GND) [orange] & 15 [yellow]	-	-	
3(GND) [green] & 16 [blue]	-	-	
4(GND) [purple] & 17 [gray]	Out 1	Trigger Out to Cicada System	
5(GND) [white] & 18 [black]	-	-	
6(GND) [brown] & 19 [red]	-	-	

Cicada (Sync-Port)

Pin Combination	Properties	Description	Image
Pin 1	Out 0	Software programmable	
Pin 2	GND		
Pin 3	Out 1	Beam start 3.3V 50Ω, High = Beam Start	
Pin 4	GND		
Pin 5	In 0	Trigger in from Griffin System	
Pin 6	GND		
Pin 7	In 1		
Pin 8	GND		

Bug Fixes

PCIe Troubleshooting

PCIe connection has to be enabled at each startup or mode change of the system if you need fast imaging but can be difficult to setup:

1) turn on all system components (PCIe-switch in case of the PC inside the rack, PC, Cicada/Griffin)

2) Execute

```
pcie_routine
```

optional 3)

```
test_throughput
```

(PCIe throughput should be > 140 Mb/s)

In case of lower throughput:

- there was probably a switch between CC-Mode and B-Mode without first removing the dmadriver → a full restart of all components is necessary (redo steps 1)-3))
- check

```
lspci -tv
```

- look for device names (Device 19aa:e004)
- if you don't see these device names

```
sudo ~/fix-pci
```

- if there is still no device, redo steps 1)-3)
- otherwise check

```
lspci -tvs Device 19aa:e004
```

- should have an [entry ≠ 00] in the 4th and 8th column in row 10
- otherwise: check if the dma driver is active

```
lsmod | egrep dma
```

- if it is not

```
sudo insmod ~/src/DMADriver/Linux/src/dmadriver.ko
```

- check

```
lspci -tvs Device 19aa:e004
```

- should have an [entry ≠ 00] in the 4th and 8th column in row 10
- otherwise: redo steps 1)-3)

Software Examples

Example Codes can be found in: cd /usr/local/cusdk/src/ex/native

Example programs are further categorized into two major operation modes:

- Beamforming – Ultrasound channel data is delayed, summed, filtered to form an ultrasound beam

- Channel Capture – Raw ultrasonic RF channel data

Finally, two primary interfaces to the CUSDK are available, these are:

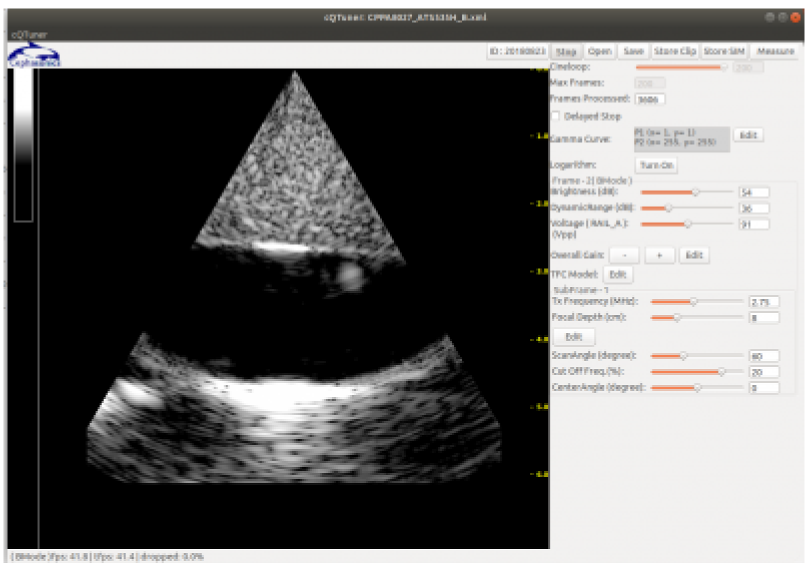
- Core Interface - Used for higher level access to beamforming and scan setup parameters.
- Advanced Interface – Low level access, closer to hardware and can be used for customization of transmit and receive sequences

CQTuner

CQTuner operates in B-mode imaging. Hence, the DMA driver has to be removed if the system is currently running in CC-Mode. CS_BITFILE_FORCE is only necessary if the system is currently running in CC-Mode:

```
sudo rmmod dmadriver   #Remove DMA Driver
cd /usr/local/cusdk/bin/x86_64/
CS_LOG_FILE= CS_BITFILE_FORCE= ./cQTuner   # Start cQTuner
```

This user interface has a real time display and reads in an xScan (XML based scan definition) file to create, configure and collect scan data. cQTuner is used as a starting point to view all the different types of scans and image quality available. The functionalities of the UI buttons on the top are introduced below.



ID: XXXX Submenu opens where information of the sample under test can be listed (e.g. Patient ID, Prefix, First Name…).

Open Loading probe file (.xml → /usr/local/cusdk/data/xscan/) with Probe configurations. The cQTuner user interface parameters will be set to the parameters stated in the probe file. New Buttons will appear in the cQTuner UI.

Start Start previewing images with the set parameters in the cQTuner UI.

Start Server Starts a server for recording images via a TCP trigger. Images are saved locally.

Save Saves the parameters adjusted in the user interface in a new probe file (.xml). Attention! Do not overwrite the probe files delivered from Cephasonics located in /usr/local/cusdk/data/xscan/.

Store Clip Stores .avi File of allocated data with size defined in the user interface (max frames, cine loop)

Store SIM Stores the RAW IQ data of the acquisition (standard path: /usr/local/cusdk/bin/x86_64/Date_Time/RAW)

Datatypes

.dat : IQ (RAW) data after digital downsampling, data is listed as follows (I Q I Q I Q), number of data points = numberBeams*2 (I + Q), size of data = SamplePerBeam*2*numberBeams

.info : header file listing the parameters of the imaging, beam angles

Measure Measuring tool to measure distances positions in images displayed live in the UI.

The program has many controls to change all aspects of the scan. cQTuner controls include:

- Open any xScan definition file
- Save any tuned scan as xScan definition file
- Cineloop capabilities
- Image, Raw Data, dump capabilities
- Movie (avi clip) creation
- Imaging modality based standard controls
- Imaging modality based advanced controls
- Simulation mode (allows data playback for post processing without need for hw)

Currently, cQTuner supports the following imaging modes:

- BMode, Harmonic Imaging (Simplex, Image Enhancement, Spatial compounding, Multi focal zone, Standard 2D Imaging controls, Advanced 2D Imaging controls)
- CFM (Duplex, Standard CFM controls, Advanced CFM controls)

cQTuner is not a clinical interface, it is an engineering interface that has controls that clinicians, engineers,and ultrasound researchers can use to tune ultrasound scans. The release comes packaged with several xScan files showing scans for different probes, phantoms, anatomies, and modes of operation. With these xScan files, sample images and movie clips are provided, so users can have an idea of what to expect with their hardware.

Blink Test

This is a practical example of the advanced interface used to display transmit and recieve on one element at a time with a constant monotonically increasing delay on the transmit and recieve firings. This simple program can be used to check probe connectivity to the system. Especially useful for those who have custom probes. This program has several command line options allowing the manipulation of selected channels and other options.

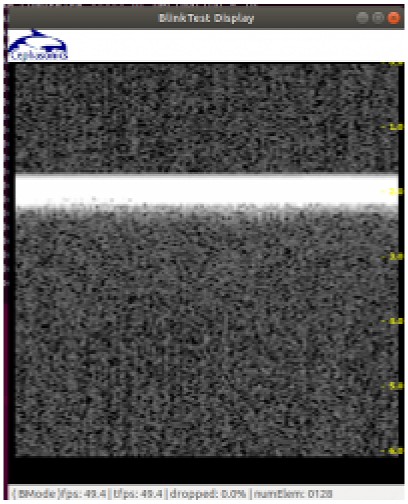
To execute the blink test to check for system and probe defects, you can run the following command:

```
/usr/local/cusdk/bin/x86_64/bt -i -n <no_of_elements>
```

where <no of elements> is the number of channels of the system you are evaluating (128 for Cicada, 256 for Griffin).

Example for Cycada:

```
cd /usr/local/cusdk/bin/x86_64/
CS_LOG_FILE= CS_BITFILE_FORCE= ./bt -n 128 -i
```



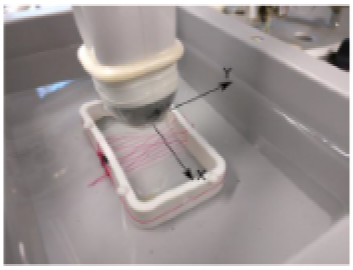
Salea

Plug Salea in Griffin Computer and use following commands to start the Logic Software:

```
salea
```

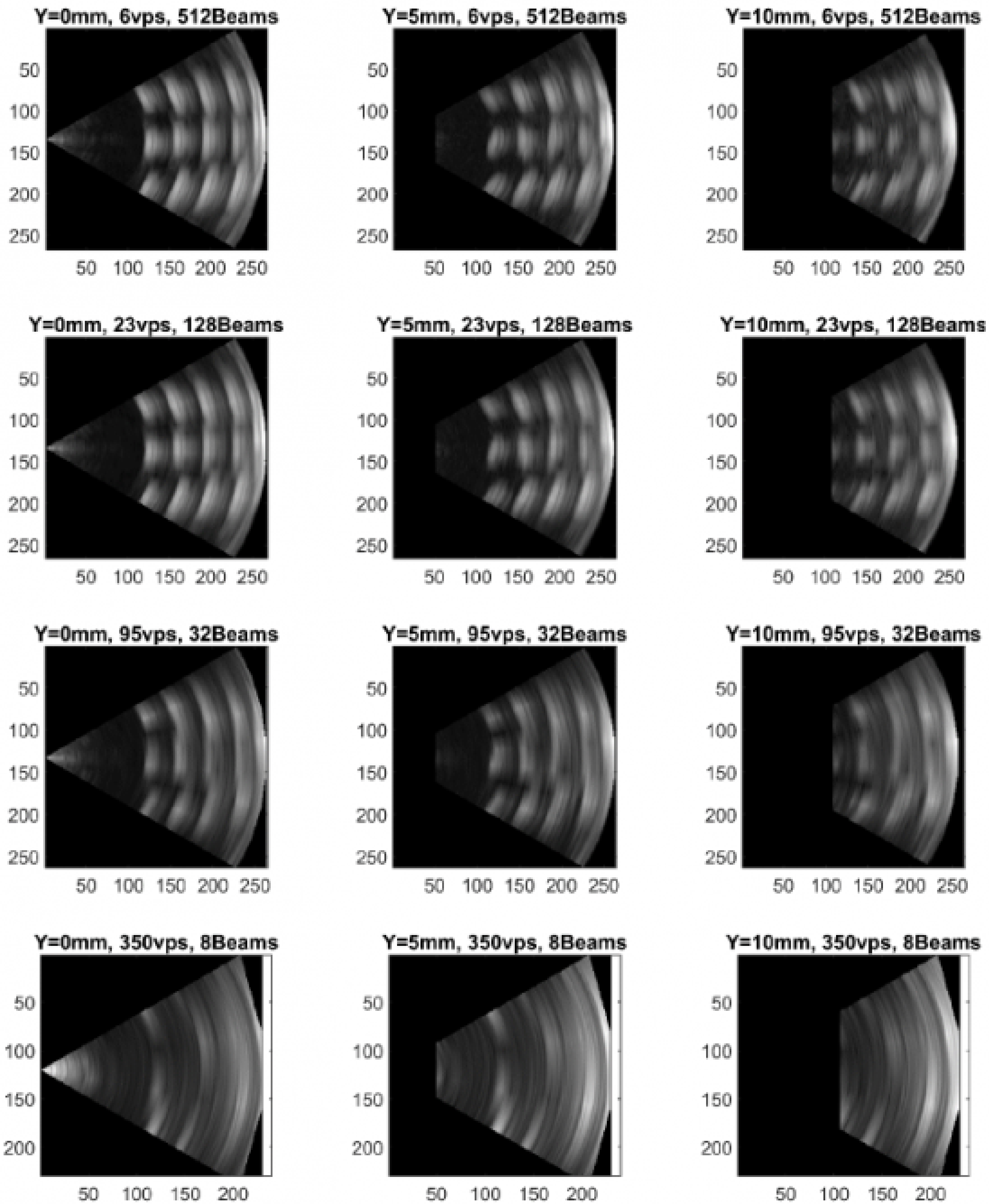
Volume Probe

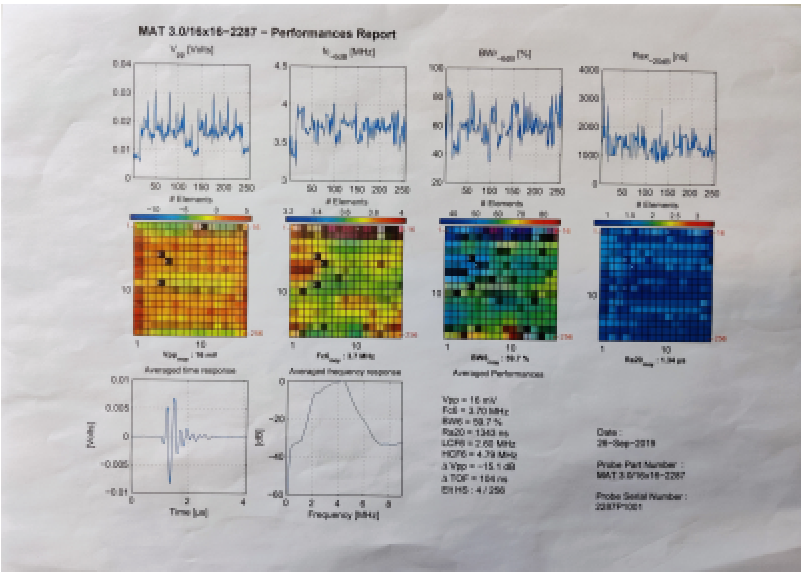
The Volume Probe consistsof 16×16 elements (total of 256). It needs to be connected to the Griffin System (Connector 0 ↔ Griffin 1, Connector 0 ↔ Griffin 1). Center frequency is 3.7MHz. The element pitch is 0.3×0.3mm.



Setup with a 4 layer Z-wire phantom (red rods) and a volume US-probe

Probe Specs:
16x16 Elements, 3.7MHz, Pitch 0.3mm
Volume Specs:
Volume Size in Pixel = 268 x 267 x 276 pixel (x,y,z)
Volume Size in mm = 40.2 x 40.1 x 41.4 mm
Each volume has a size of ca. 80MB





Shear Waves

Example for US Shear Wave push and imaging.

General Information

Phantoms: If you would like to use gelatin phantoms with TiO2 particles inside and also in case of graphit - make sure, that no air bubbles are at the surface. Otherwise, a wave wont be coupled properly into the phantom.

Systems: The Cicada system has a higher operating voltage output and is best used for pushing. Imaging can be done either with Cicada or the Griffin System.

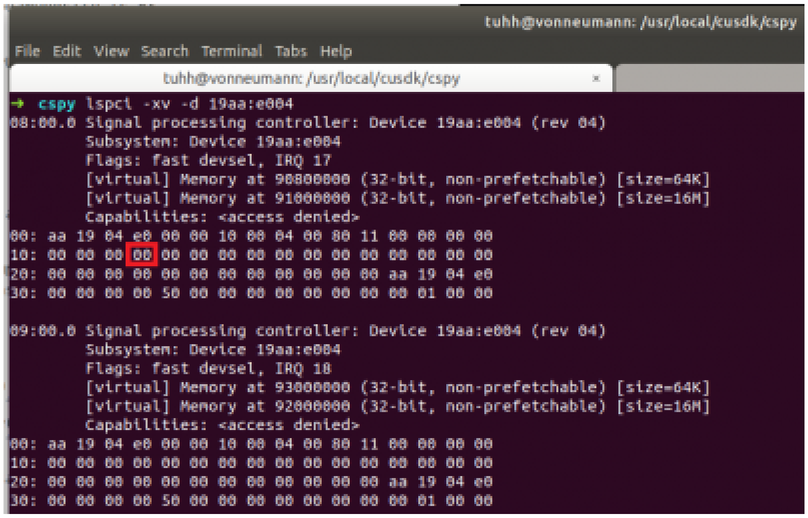
For switching from CC-Mode (shear wave elastography) to B-Mode imaging (cQTuner):

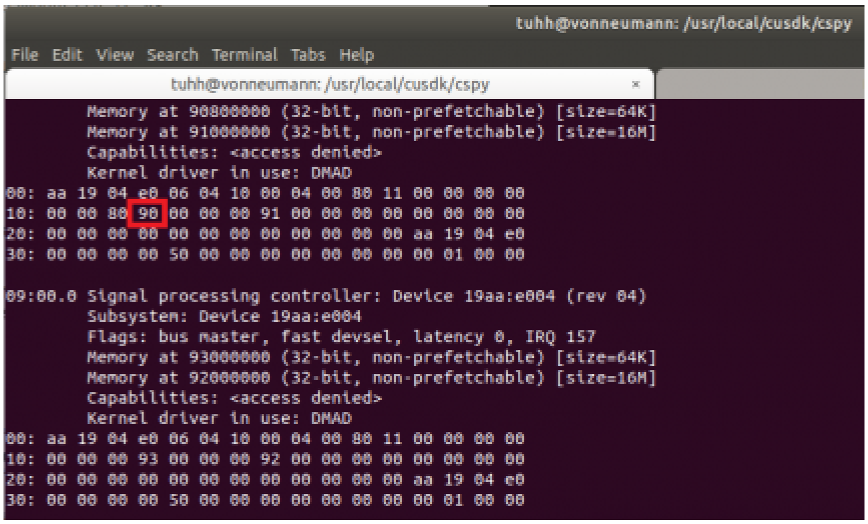
```
sudo rmmod dmadriver   #Remove DMA Driver
cd /usr/local/cusdk/bin/x86_64/
CS_LOG_FILE= CS_BITFILE_FORCE= ./cQTuner   # Start cQTuner
```

General Setup Cicada

Check if PCiE is active:

```
lspci -xvd 19aa:e004
```





if the entry in the red box is '00' then no PCiE is active. Left image no PCiE right image PCiE active. If not active then start CC Mode:

```
cd /usr/local/cusdk/bin/x86_64/
CS_LOG_FILE= CS_BITFILE_FORCE= ./btcc
```

```
~/fix-pci
```

```
sudo insmod ~/src/DMADriver/Linux/src/dmadriver.ko
```

Check if PCiE is now active if FPS > 160 (with Cicada >200):

```
test_throughput
```

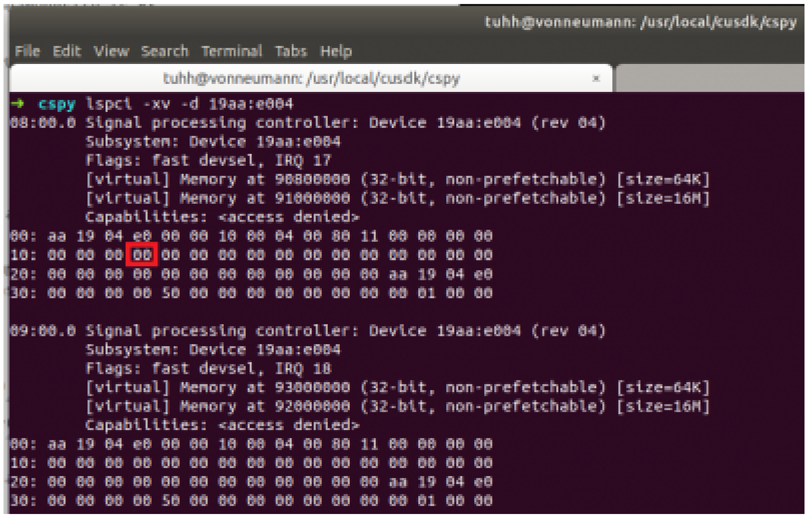
General Setup Griffin

PCiE is found if 'Device 19aa:e004' appears in the list with:

```
lspci -tv
```

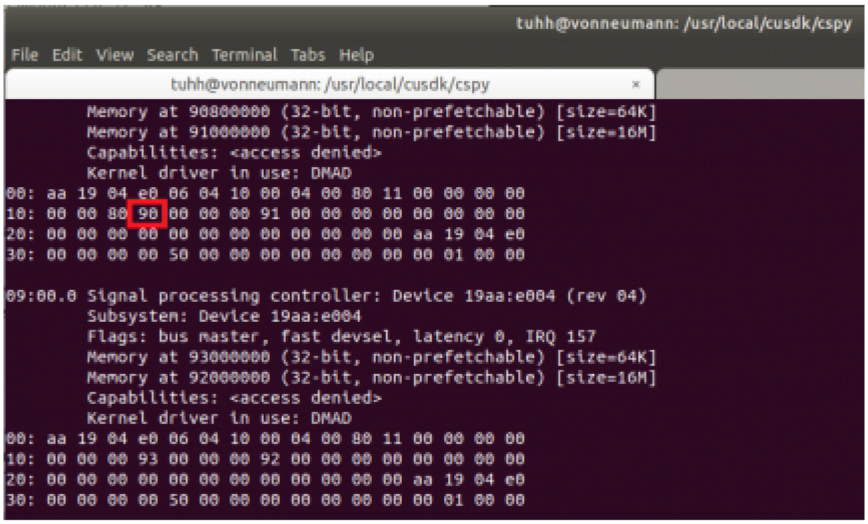
Also check with:

```
lspci -xvd 19aa:e004
```



```
tuhh@vonneumann: /usr/local/cusdk/cspy
File Edit View Search Terminal Tabs Help
tuhh@vonneumann: /usr/local/cusdk/cspy
➜ cspy lspci -xv -d 19aa:e004
08:00.0 Signal processing controller: Device 19aa:e004 (rev 04)
        Subsystem: Device 19aa:e004
        Flags: fast devsel, IRQ 17
        [virtual] Memory at 90800000 (32-bit, non-prefetchable) [size=64K]
        [virtual] Memory at 91000000 (32-bit, non-prefetchable) [size=16M]
        Capabilities: <access denied>
00: aa 19 04 e0 00 00 10 00 04 00 80 11 00 00 00 00
10: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
20: 00 00 00 00 00 00 00 00 00 00 00 00 aa 19 04 e0
30: 00 00 00 00 50 00 00 00 00 00 00 00 00 01 00 00

09:00.0 Signal processing controller: Device 19aa:e004 (rev 04)
        Subsystem: Device 19aa:e004
        Flags: fast devsel, IRQ 18
        [virtual] Memory at 93000000 (32-bit, non-prefetchable) [size=64K]
        [virtual] Memory at 92000000 (32-bit, non-prefetchable) [size=16M]
        Capabilities: <access denied>
00: aa 19 04 e0 00 00 10 00 04 00 80 11 00 00 00 00
10: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
20: 00 00 00 00 00 00 00 00 00 00 00 00 aa 19 04 e0
30: 00 00 00 00 50 00 00 00 00 00 00 00 00 01 00 00
```



```
tuhh@vonneumann: /usr/local/cusdk/cspy
File Edit View Search Terminal Tabs Help
tuhh@vonneumann: /usr/local/cusdk/cspy
        Memory at 90800000 (32-bit, non-prefetchable) [size=64K]
        Memory at 91000000 (32-bit, non-prefetchable) [size=16M]
        Capabilities: <access denied>
        Kernel driver in use: DMAD
00: aa 19 04 e0 06 04 10 00 04 00 80 11 00 00 00 00
10: 00 00 80 90 00 00 00 91 00 00 00 00 00 00 00 00
20: 00 00 00 00 00 00 00 00 00 00 00 00 aa 19 04 e0
30: 00 00 00 00 50 00 00 00 00 00 00 00 00 01 00 00

09:00.0 Signal processing controller: Device 19aa:e004 (rev 04)
        Subsystem: Device 19aa:e004
        Flags: bus master, fast devsel, latency 0, IRQ 157
        Memory at 93000000 (32-bit, non-prefetchable) [size=64K]
        Memory at 92000000 (32-bit, non-prefetchable) [size=16M]
        Capabilities: <access denied>
        Kernel driver in use: DMAD
00: aa 19 04 e0 06 04 10 00 04 00 80 11 00 00 00 00
10: 00 00 00 93 00 00 00 92 00 00 00 00 00 00 00 00
20: 00 00 00 00 00 00 00 00 00 00 00 00 aa 19 04 e0
30: 00 00 00 00 50 00 00 00 00 00 00 00 00 01 00 00
```

if the entry in the red box is '00' then no PCiE is active. Left image no PCiE right image PCiE active. If not active then double proof by checking that FPS is > 160:

```
test_throughput
```

If FPS < 160 then onyl USB is active. Force change to CC mode and insert DMA-Driver by running the following script:

```
pcie_routine
```

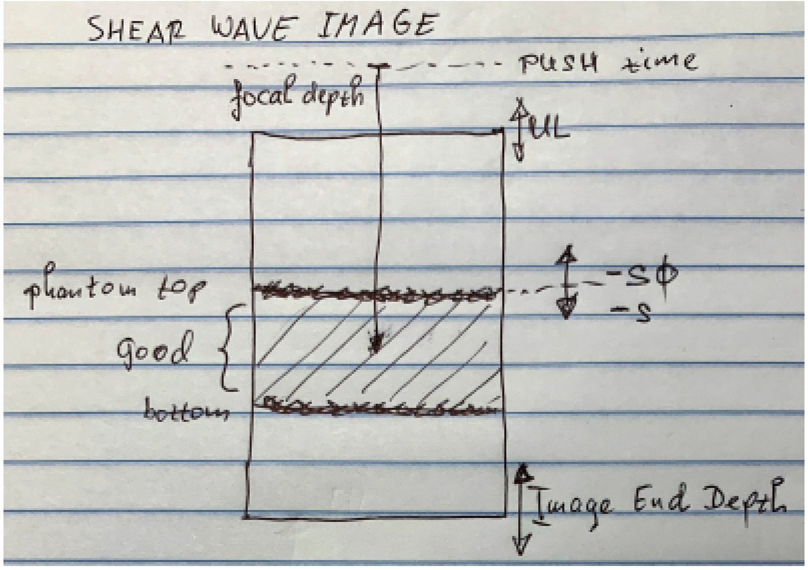
If this does not work then try restarting the computer and run ./pcie_routine again.

Parameters

Paramter	default value	description
START_DEPTH	0.01	Phantom depth [m] at which raw data acquisition starts
END_DEPTH	0.03	Phantom depth at which raw data acquisition end
IMAGE_END_DEPTH	0.025	Depth of reconstructed image
PIXELS	400	Width of image in Pixel after reconstruction
GAMMA	0.5	Image Brightness
IMAGE_DIR	images	Saved images

Paramter	**default value**	**description**
TX_FREQ	5000000	Frequency for pushing
PUSH_FOCAL_DEPTH	0.022	Depth [m] of push
PUSH_ORIGIN	32	All elements are mapped from 1-64. Center push = 32.
PUSH_CYCLES	2000	Number of cycles for puhsing. DO NOT CHANGE ABOVE 2000! DANGER!
NUM_PUSH_SEQ	1	Repititions of push cycles. DO NOT CHANGE ABOVE 1! DANGER!
GAP	0.5	Time delay between Repitions of push cycles.
CUST_PROBE_MAP	0	Mapping of Elements (0 or 1).
PROBE_LEN	0.0375	Lenght [m] of Probe. Can be derived from the XScan folder.
NUM_ELS	128	Elements of probe. Can be derived from the XScan folder.
ANGLE_DEGREES	0	Angle of plane wave (PW) imaging (old).
NUM_ANGLES	1	Number of angles of PW-imaging (old).
ENSEMBLE_REP	10	Number of images to acquire after pushing.
PRF	2000	Frequency [Hz] for image acquisition.
SHIFT_COUNT	0	Sequential imaging: Number of repetitions of imaging + pushing.
SHIFT_RATIO	0.99	Shift between repetitions in percentage (1=100%). Can be set to: SHIFT_RATIO=1/(SHIFT_COUNT+1).
EXTRA	0.9	Time delay between pushing and start imaging.
DECIMATION	2	Raw data downsampling from 40MHz.
SAMPLING_FREQ	20000000	Resulting Frequency: 40MHz/DECIMATION.
S0	-240	Sample of shooting. Parameter that strongly influences the focus. Good parameters are: -120,-240,-280. Will shift display up and down. Check e.g. a screw can be good seen.
UL	“(0.0,0.01)”	Fix display.
BURST_CYCLES_ON	150	Test from Cephasonics: divide push in x-cycles pushing (BURST_CYCLES_ON) and x-cycles cooling off (BURST_CYCLES_OFF).
BURST_CYCLES_OFF	0	Test from Cephasonics: divide push in x-cycles pushing (BURST_CYCLES_ON) and x-cycles cooling off (BURST_CYCLES_OFF)
SPEED_SOUND	1553	Speed of sound in phantom [m/s]. (Gelatine ca. 5%=1464, 7.5%=1644, 10%=1520, 15%=1539,17.5%=1590,20%=1615)

Image from Bob describing some of the parameters:



Push and Imaging Cicada

Check if PCiE is active. Start the following code with only -f as file parameter on the Cicada system:

```
cd /usr/local/cusdk/cspy
bin/egcc_run_UFI_PUSH -f params/ufiPUSH_7_5mhz.txt
```

For Imaging only on Griffin use:

```
cd /usr/local/cusdk/cspy
bin/egcc_run_UFI_PUSH -f params/ufiPUSH_10mhz.txt -L
```

Only if the correct ultrasound speed and S0 factor is chosen a good push is feasible. Demo Video left with PRF=2000Hz and 7.5MHz linear probe and 10% gelatine. Demo video right provided by Cephasonics with 10kHz image acquisition:

Push Cicada and Imaging Griffin

The Cicada system waits for a trigger from the master Griffin system. Start the following on the Cicada with the parameters -P (Push only) and -f (file with probe parameters). The program will now wait for a input trigger on pin 5 (red cable) and pin 6 (ground) from the Griffin system pin 4 (trigger) and pin 17 (ground).

```
cd /usr/local/cusdk/cspy
bin/egcc_run_UFI_PUSH -f ufiPUSH_7_5mhz.txt -P
```

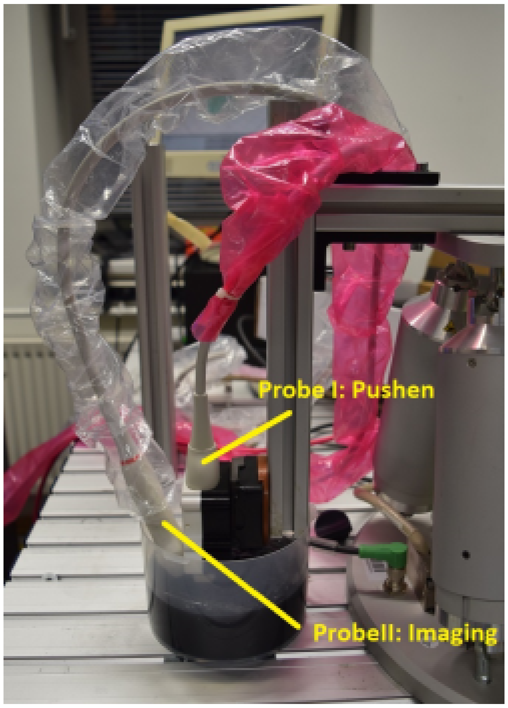
Setup the Griffin system for imaging with parameters -f (probe file), -L (Griffin System), -N (no pushing), -I (10MHz Probe for imaging), -U (if connected to left '1' adaptor). Start with 10 Images only and adapt S0 and speed of sound of phantom.

```
cd /home/tuhh/Desktop/cusdk/cspy
bin/egcc_run_UFI_PUSH -f ufiPUSH_10mhz.txt -L -N -I
```

Demo Video with PRF=2000Hz and 7.5MHz linear probe for pushing and 10Mhz probe for imaging. Phantom at 10% gelatine concentration.

Edit video with:

```
bin/showPWLoupasTk -L 0.02489 -D 0.03 -d images -g 0.5 -c 1550 -P 2000
```



Push Piezo and Imaging Griffin

```
cd /home/tuhh/Desktop/cusdk/cspy
bin/egcc_run_UFI_PUSH -f params/ufiPUSH_10mhz.txt -L -N -I
```

Save raw data

To save the raw US-data to cusdk/cspy/Raw/<date> you can add “-s” to your push command. E.g.:

```
cd /home/tuhh/Desktop/cusdk/cspy
bin/egcc_run_UFI_PUSH -f ufiPUSH_10mhz.txt -L -N -I -s
```

cbmde

This example program is the first example program users should start with. It uses the Core API interface to build a simple scan, collect data, scan convert and dump image files. The example program is heavily commented to explain the coding process and APIs.

MATLAB

Start Matlab with:

```
sudo Matlab/Matlab_2019a/bin/matlab
```

CQ Tuner

1. start CQ Tuner via terminal:

```
start_cQTuner
```

2. press “open” and choose .xml probe configuration file

3. press “start svr”

4. press “start”

5. run Matlab code:

```
%% connect to server
instrreset
US=tcpip('134.28.45.11', 3000); % IP and Port, where cQTuner is running (may vary!)
fopen(US);
set(US, 'TimeOut', 1);

%% set name of imagefolder
expName = 'exampleFolderName'; % Name of the experiment, string
fileNameUS = [expName];
fprintf(US, ['setFolder' fileNameUS]);

%% start/stop imaging
fprintf(US,'bmodeStart\n');
pause(0.5); % 500ms break for one image; multiple images also possible
fprintf(US,'bmodeEnd\n');
pause(0.5); % wait 500ms for US restart in case of looping;

%% disconnect from server
fprintf(US,'disconnect\n');
fclose(US);
delete(US);
clear US
```

6. image folder location: Desktop/cusdk/bin/x86_64/exampleFolderName

SUPRA

TODO: summarize commands

refer to home/git/supra_shearwave/MatlabClient.m for an example

SUPRA

SUPRA [http://gitlab.et8.tu-harburg.de/Gerlach/supra_shearwave/] is an open-source framework for ultrasound applications.

Set the system to CC-Mode (pw: tuhh):

```
pcie_routine
```

Check if PCIE is running correctly (FPS>140):

```
test_throughput
```

You can start an example for the imaging pipeline in CC-Mode by:

```
start_supra -c ../supra_shearwave/config/configCephasonics.xml
```

You can start the SUPRA gui with server support to execute commands over LAN by

```
start_supra -c ../supra_shearwave/config/configCephasonics.xml -p <PORT>
```

where <PORT> has to be replaced by a port of your choice. Available commands can be queried by sending “help\n” to the server. Note the “\n” at the end of each command.

How to use SUPRA

SUPRA offers a multitude of setting options for image capturing and processing. To get started you can visit the wiki [https://github.com/IFL-CAMP/supra/wiki] on SUPRA or watch the SUPRA-con playlist [https://www.youtube.com/watch?v=dQkUSpV2CtE&list=PLWMNh90FOFSiGOA_VilmuRVqD_5goBrJq].

Images are saved by including a MetaImageOutputDevice-Node in the graph at any point of the pipeline. When “Sequence Start” is triggered (either be pressing the button or using the server), images are saved until “Sequence Stop” is triggered or the specified number of images has been saved. Images are saved relative to ~/git/data. You can specify a folder and filename. E.g. “folder/filename” would save in ~/git/data/folder/filename_0.raw …

Issues

- The number of scan lines cannot be changed while imaging is active. You need to stop imaging, change the number of scanlines and start imaging again.

Software Installation

The installation files can be found on Pallando: \\pallando.et8.tu-harburg.de\data\software\Geräte\CephasonicsUltrasound\Software

Perform the following steps on a Linux Platform:

```
tar xvfz CUSDK_2.6.10_U18.04_SRV.tar.gz
sudo mv CUSDK_2.6.10_U18.04_SRV /usr/local
cd /usr/local/    cd /usr/local/cusdk
vi .envrc
direnv allow
cd cfg/install/
./packageSetup.sh
```

Check installation by building and testing with demo programs:

```
cd /usr/local/cusdk/src/ex/native/cqtuner
make
cd ../bt
make
cd /usr/local/cusdk/bin/x86_64/
CS_LOG_FILE = ./bt -n256 -i
CS_LOG_FILE = ./cQTuner
```

hardware/us/cephasonics.txt · Last modified: 2021/07/06 13:40 by 134.28.45.11