UTIA evBoard v1.0 Beamforming Demo

Zdeněk Pohl, Lukáš Kohout
zdenek.pohl@utia.cas.cz, *kohoutl@utia.cas.cz*

Revision history

Rev.	Date	Author	Description
01	20.5.2019	Z.P.	Document creation
02	2.7.2019	Z.P.	Finalization

Contents

Acknowledgement

This work has been supported from project SILENSE, project number ECSEL 737487 and MSMT 8A17006.

1 Description

This document provides documentation to hardware evaluation board developed in UTIA (UTIA evBoard v1.0), hardware interfaces in FPGA and accelerated DSP chain for implementation of beamforming application. To view measured data, the PC application was also developed and its operation is described in this document. The hardware platform consists of three basic components: TE0720 FPGA SoM module, TE0706 carrier board and UTIA evBoard v1.0, see Figure 1.

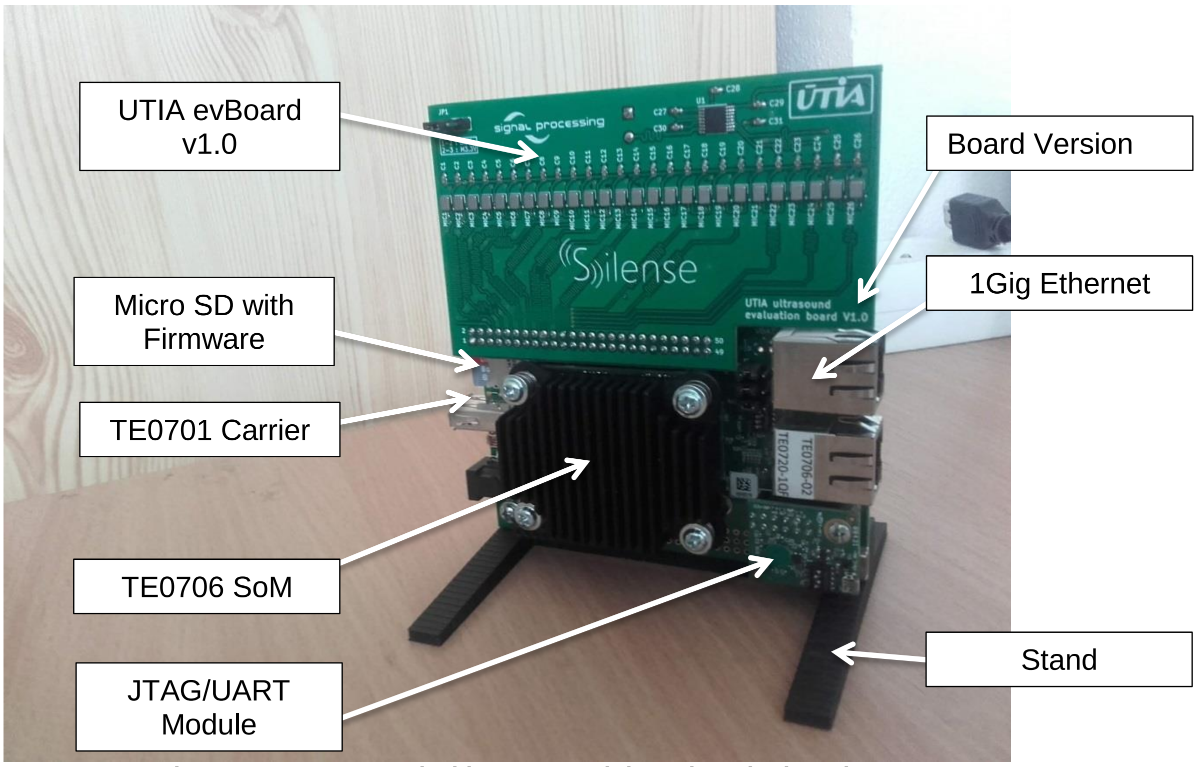


Figure 1: UTIA evBoard with FPGA module and carrier board.

2 Board Features

The evaluation board version 1.0 implements following features:

a) Linear microphone array consisting of 26 digital microphones.
b) Microphone frequency range 0-40kHz
c) Array maximal frequency 40kHz. Distance between microphone ports = 3.8 mm
d) 40 kHz piezo US speaker with wide output beam to cover maximal area
e) Common clock distribution to all microphones
f) US speaker driving circuit
g) Connector for mating with TE0706 carrier board J5 connector.

3 How to Run

Prerequisites:

1. UTIA evBoard v1.0 assembled with TE0706 carrier with TE0720-1CF/1QF/2IF Zynq SoM.
2. 5V power source for TE0706.
3. (optional) Mini USB cable for Virtual UART connection to PC via JTAG/UART module.
4. UTP Cable for connection to PC data visualization. (board uses UDP multicast to pass beamformer output)
5. (optional) Board stand mounted to support assembled array in upright position. It features also possibility to be mounted on tripod.
6. Carrier board jumpers J10,J11,J12 all shorts placed to 2-3 position.
7. The DHCP server must be running on local network where the board is connected. Alternatively, the UART connection to board can be used to setup Ethernet interface manually.
8. Micro SD card with firmware.

Updating/Writing firmware to Micro SD card:

1. Plug the Micro SD card to PC reader (it may be needed to use SD card adapter)
2. Format the card to FAT32 filesystem. Alternatively, the FAT32 can be only first partition of the card with size of 256 MByte.
3. Copy content of firmware package to the SD card root.
4. Disconnect card and remove it from reader.

Running the application:

1. Populate Micro SD card with firmware to J4 slot on TE0706.
2. Plug the power source. Board should start booting automatically. I the boot process does not start automatically, the reset button S2 on carrier board TE0706 can be used to initiate booting.
3. The beamformer application should start automatically after linux boot. In case of need it can be also started manually by following command:

   ```
   > /run/media/mmcblk0p1/evBoard_ffbf.elf
   ```

   IMPORTANT NOTE: If another instance of the application is running, it must be terminated before executing command. Otherwise the application will crash.

4. The board features FTP server which can be used for accessing captured data files or beamformer configuration file:
   a. Configuration is stored in `evboard_ffbf.cfg` file at location:

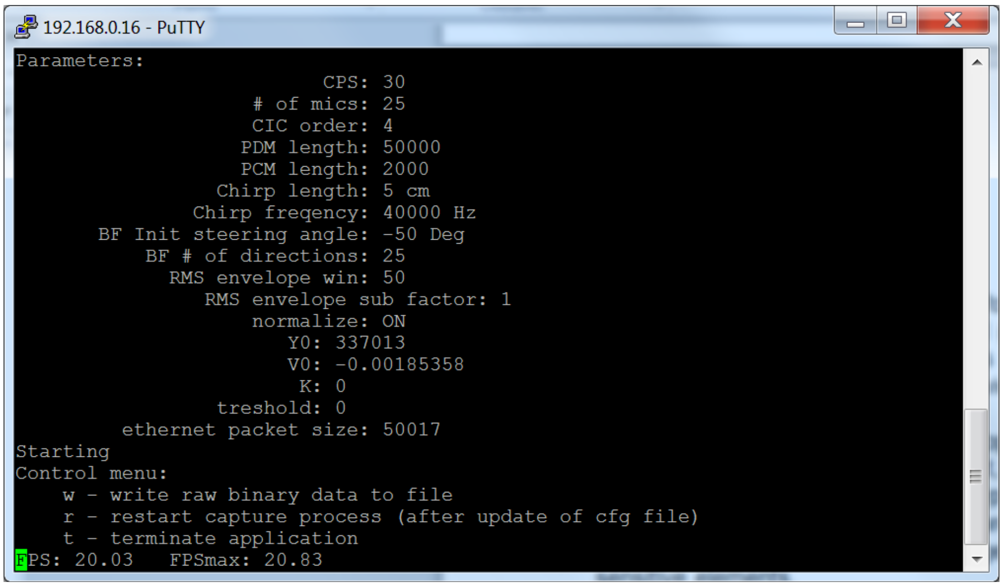
      ```
      > /run/media/mmcblk0p1/evboard_ffbf.cfg
      ```

   b. Captured data can be found in the following file after they are written (must be manually initiated while the application is running):

      ```
      > /run/media/mmcblk0p1/test.bin
      ```

5. While the application is running:

   a. The actual parameters of beamformer are shown in terminal:



```
Parameters:
                          CPS: 30
                    # of mics: 25
                    CIC order: 4
                   PDM length: 50000
                   PCM length: 2000
                 Chirp length: 5 cm
               Chirp freqency: 40000 Hz
       BF Init steering angle: -50 Deg
           BF # of directions: 25
             RMS envelope win: 50
                RMS envelope sub factor: 1
                    normalize: ON
                       Y0: 337013
                       V0: -0.00185358
                        K: 0
                 treshold: 0
         ethernet packet size: 50017
Starting
Control menu:
    w - write raw binary data to file
    r - restart capture process (after update of cfg file)
    t - terminate application
FPS: 20.03   FPSmax: 20.83
```

The application updates actual frame rate (FPS) for beamformer output, at the same time it presents maximal framerate (FPSmax) which describes the maximal FPS which can be reached in this configuration of the beamformer (real FPS may be lower because of lower chirp per second (CPS) request in configuration file.

b. From terminal, the beamformer application can be controlled by entering following command characters:
   i. **w –** writes raw binary data of one captured chirp to file `test.bin.`
   ii. **r –** restarts the application. It is intended to be used after change in configuration file.
   iii. **t –** terminates the beamformer application

4 Beamformer Application Configuration File

The beamformer application was implemented to load its configuration from file in the host filesystem. The individual parameter values must be present in the file in given order and within the correct ranges. Writing other values than listed in the following table may crash the application.

Row #	Default Value	Value Range	Description
1	30	1-60	Desired number of chirps per second generated and captured, if too high it limits maximal PCM length
2	25	1-26	Number of mics to be used for beamforming. It always starts from MIC1. Value 5 in this field means mic1-5 will be used (see PCB labels at mics)
3	4	1-4	Order of CIC decimation filter, decimation factor is fixed to 25, decimation factor determines sampling frequency 192 kHz
4	2000	1-192000	Number of PCM samples to be captured – will be limited by application if requested CPS is too high
5	5	1-30	Chirp length in cm. Length of generated chirp will be rounded to full periods of given chirp frequency value. Example: at 40 kHz one period wave length is 8mm, 5 cm chirp request will generate 6 periods. More chirp length creates more blurry output. Lower value creates weaker chirp.
6	40000	unsigned int	Desired chirp frequency. wavegenerator creates rectangular chirp signal at desired frequency. Keep in mind that resonant frequency of used speaker is peaking around 40 kHz.
7	0	0-3	Number of speaker active braking periods to cancel ringing. Number must be lower than number of periods in chirp. We do recommend to use value 2 as value 3 has no effect and value 4 is already making reverse phase response of the membrane.
8	-50.0	float 0…-90.0	Beamformer angle from array axis. When -50.0 is set, the beamformer assumes its view angle to be from -50 to 50 degrees with respect to array axis.
9	25	1-100	Number of directions computed by the beamformer. The view angle will be divided to given number of sectors. For example value 3 for angle -50 to 50 says three beams

			will be computed using beam directions: [-50,0,50] Deg
10	50	1-100	Window length for RMS envelope computation
11	1	see decription	Subsampling of RMS envelope output, value 1 = subsampling OFF, value 2 = drop every odd sample, etc. We do recommend to use lower value than 1/10th of window length
12	0	unsigned int	Minimal value for conversion to CV_8UC1, used when normalize flag is off. Output is computed as min-max saturation and after that scaled to unsigned char range.
13	24000	unsigned int	Maximal value for conversion to CV_8UC1, used when normalize is off
14	1	0-1	Normalization: 0 – off, 1 – on
15	337013	float	First coefficient of normalization function fitted by calibration process: The calibration output curve: Y = 667013.e^-0.00185358 Used normalization curve is: Y = 337013.e^-0.00185358
16	-0.00185358	float	Second coefficient of normalization function fitted by calibration process: The calibration output curve: Y = 667013.e^-0.00185358 Used normalization curve is: Y = 337013.e^-0.00185358
17	0.0	float	Reserved value
18	0.0	float	Reserved value

5 Beamformer UDP Multicasting

UTIA evBoard multicasts its output to multicast group 239.0.01:6001. Every beamformer output is sent as one packet with header. Zynq ARM core is using the same endianness as Intel architecture – no conversions to network byte order are performed.

Packet structure:

Address	Size	Type	Name	Description
0x0000	2	short	height	number of rows, rows represent beamformer directions
0x0002	2	short	width	Width, number of columns, length of RMS envelope output
0x0004	4	float	fi0	initial angle of

				beamformer
0x0008	1	unsigned char	fps	frames per second
0x0009	4	unsigned	max_val	Value of global maximum, computed in HW before normalization, used for calibration
0x000D	2	unsigned short	max_row	Row index of global maximum, computed before normalization, used for calibration
0x000F	2	unsigned short	max_col	Column index of global maximum, computed before normalization, used for calibration
0x0010	height*width	unsigned char	data	Beamformer output array, pixel format CV_8UC1

Example code for parsing header from packet which works on Intel architecture, it is assumed that packet data are in `msgbuf` array of `unsigned char` type:

```
int rows = (msgbuf[1] << 8) | msgbuf[0];
int cols = (msgbuf[3] << 8) | msgbuf[2];
float fi0 = *((float*)&msgbuf[4]);
unsigned char fps = msgbuf[8];
unsigned max_val = (msgbuf[12] << 24) | (msgbuf[11] << 16) | (msgbuf[10] << 8) | msgbuf[9];
unsigned max_row = (msgbuf[14] << 8)  | msgbuf[13];
unsigned max_col = (msgbuf[16] << 8)  | msgbuf[15];
```

6 PC Application bf_view

For visualization of multicasted beamformer data, the application for PC was developed using OpenCV 3.4.1 library. The application implements these functionalities:

1. Opens and reads UDP multicast from the board.
2. Computes beamformer data transformation to interpolated polar image.
3. (optionally) Captures data from USB webcam to add also visual information.
4. Implements recoding of displayed data to video file.
5. Support flipping of USB webcam image.
6. Implements calibration mode – captures calibration data set (point of global maxima) and computes exponential fit curve.
7. Supports crop to hide chirp passing over microphones.

The application is parsing the UDP packets coming from the UTIA evBoard.

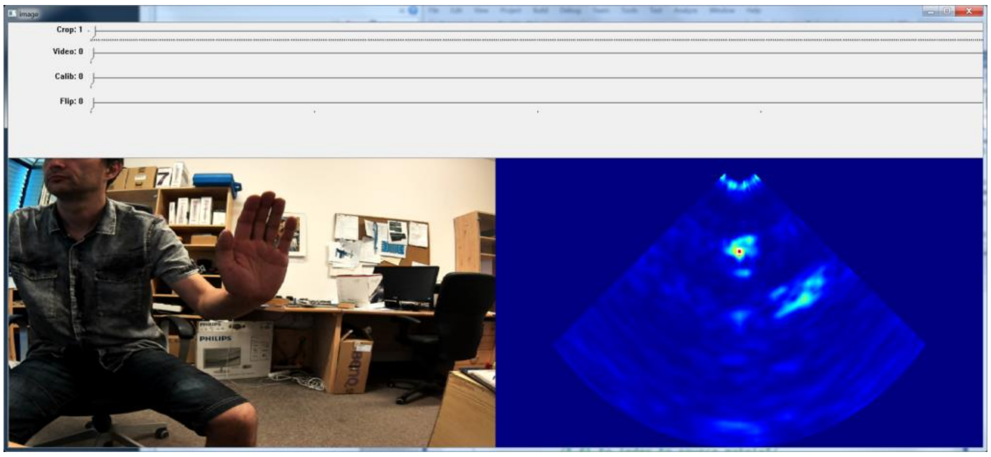

Crop: 1
Video: 0
Calib: 0
Flip: 0

Figure 2: PC application bf_view GUI.

7 Using Matlab Scripts to Analyze Raw Captured Data

The folder Matlab of the package contains scripts and workspace variables needed to show captured raw data from microphones as a waveforms. To show captured waveforms follow these steps:

1. We assume that raw captured data file ‘`test.bin`’ were copied using FTP from the UTIA evBoard to folder `o:\Matlab\testvec`
2. Open Matlab and change path to `o:\Matlab`. (path where Matlab is executing scripts can be checked by pwd command)
3. Call:

   ```
   >> load('utia_evb_25.mat')
   ```

   the command loads variable `p_utia_evb25` which contains parameters needed to process the raw data. The file sets the number of processed microphones to 25 (mic26 will be ignored).
4. Use command:

   ```
   >> [pcm chirp] =
   show_evboard_waveforms_b('testvec/test.bin',p_utia_evb25,f
   igure(1),[1]);
   ```

   After short time, 25 figures each containing captured data from microphones converted to PCM at 192 kHz are displayed.

5. The output variable `pcm` is array with PCM samples for every microphone and `chirp` contains chirp waveform subsampled to PCM sample time. Its waveform shape is affected by the subsampling. More precise result may be obtained from raw data without subsampling.

8 License

The package is provided by UTIA AV CR as is free of charge. For sources please contact authors.

9 Content of the package

evBoard_v1_0_files.zip	
- bf_view	Microsoft Visual Studio Community 2015 project for bf_view application, source included, requires OpenCV 3.4.1 library
- firmware	Contains ZIP archive with SD card content.
- Matlab	Matlab scripts to process raw data which can be captured by evBoard

10 References

[1] Silense project web pages: http://www.silense.eu