Embedded Systems
Laboratory Instruction

Exercises 18 & 18B

DSP – Digital Signal Processing

Supervisor: Grzegorz Baron

Hardware

There are two types of the evaluation boards available during the laboratory, BF533-EZ-KIT Lite, and BF537-EZ-KIT Lite. Both are based on the Blackfin microprocessors developed by Analog Devices company.

The short description of that devices is as follows:

- Blackfin Processors are 16-32-bit embedded microprocessor designed specifically to meet the computational demands and power constraints of today's embedded audio, video and communications applications.
- High Performance Processor Core
- High Bandwidth DMA Capability
- Video Instructions
- Efficient Control Processing
- Hierarchical Memory
- Superior Code Density
- Dynamic Power Management
- Easy to Use

All the family shares the same core presented in Fig. 1.

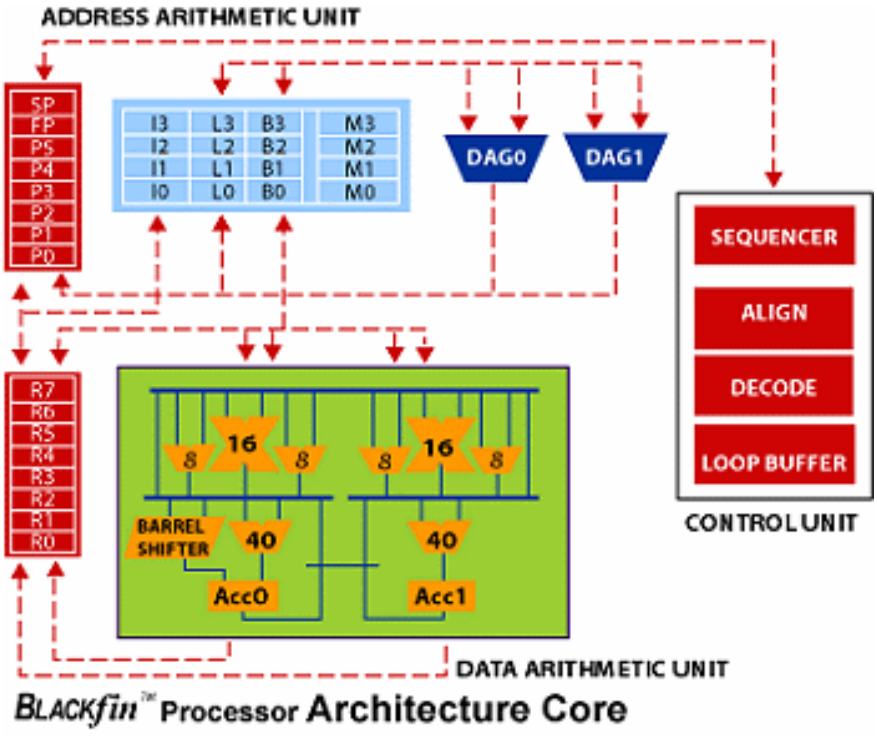


Figure 1 Blackfin processors Architecture Core

- General-purpose register files
    - Data register file

    - Data types include 8-, 16-, or 32-bit signed or unsigned integer and 16- or 32-bit signed fractional
    - two 32-bit reads AND two 32-bit writes (every clock cycle)
    - Address register file
    - Stack pointer
    - Frame pointer
- Data arithmetic unit
    - Two 16-bit MACs
    - Two 40-bit ALUs
    - Four 8-bit video ALUs
    - Single barrel shifter
- Address arithmetic unit
    - Memory fetches
    - Index, length, base, and modify registers
    - Circular buffering
- Program sequencer unit
    - Conditional jumps and subroutine calls
    - Nested zero-overhead looping
    - Code density

The figures 2 and 3 present the pictures of the evaluation boards.

Figure 2 BF533-EZ-KIT Lite evaluation board

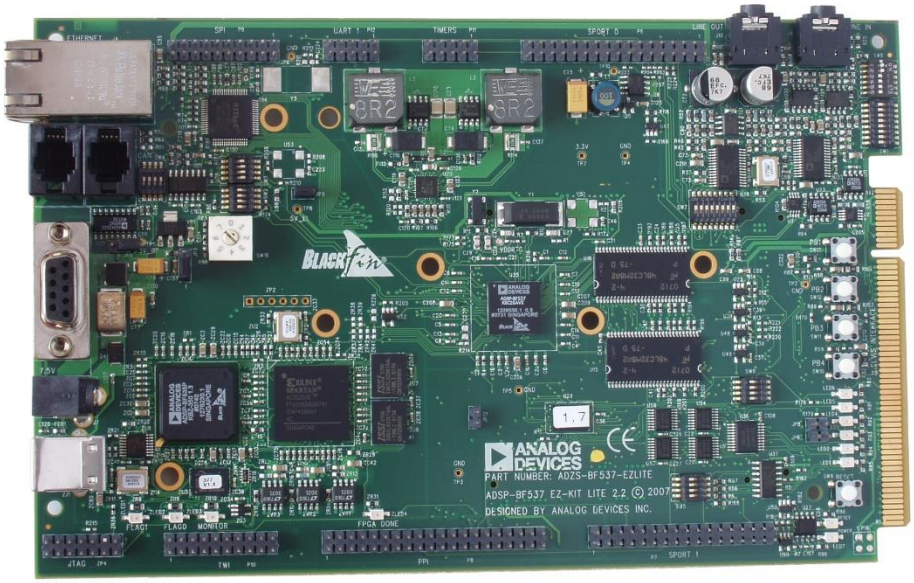


Figure 3 BF537-EZ-KIT Lite evaluation board

Figures 4 and 5 present the block diagrams of the evaluation boards.

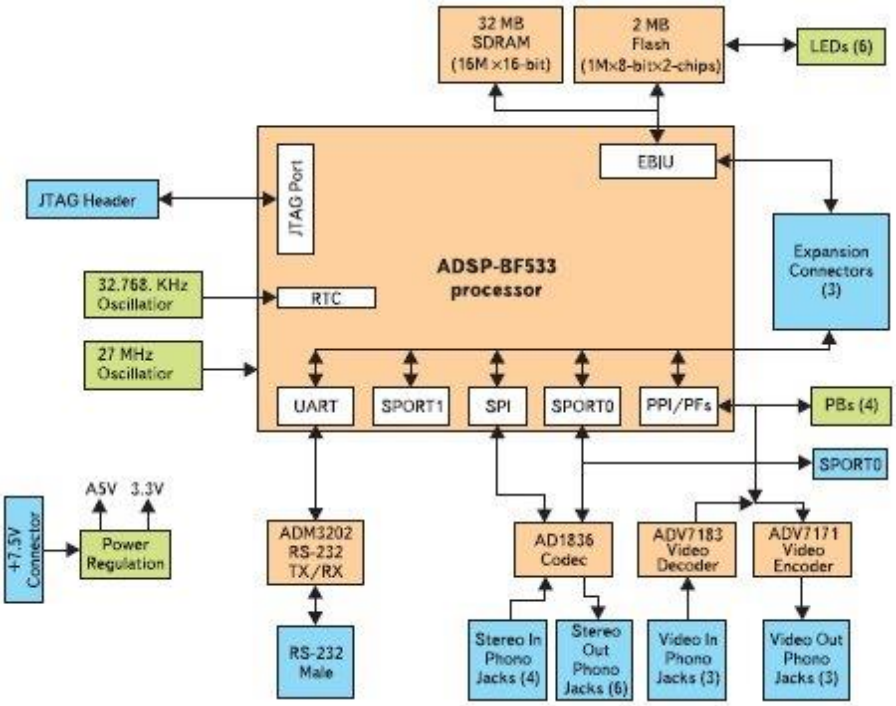


Figure 4 BF533-EZ-KIT Lite block diagram

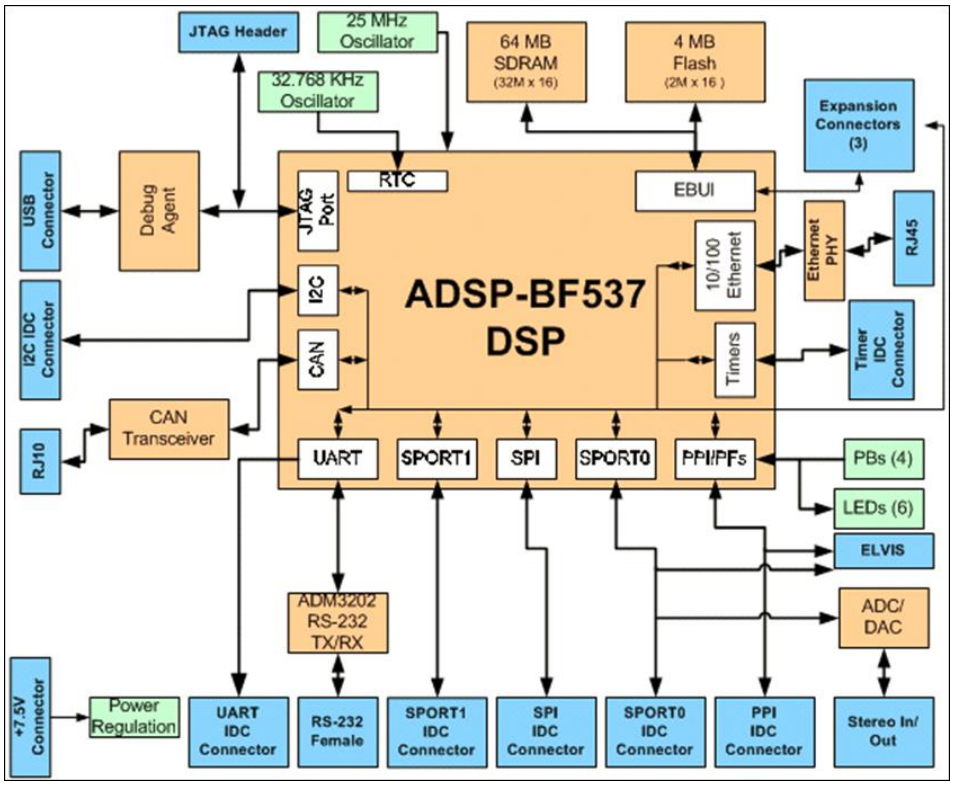


Figure 5 BF537-EZ-KIT Lite block diagram

Full documentation of the evaluation boards is available on the Analog Devices web site:

http://www.analog.com/media/en/dsp-documentation/evaluation-kit-manuals/ADSP-BF533_ezkit_man_rev.3.2.pdf

http://www.analog.com/media/en/dsp-documentation/evaluation-kit-manuals/ADSP-BF537_ezkit_man_rev.2.52.pdf

Integration of the IDE and hardware

To execute and debug the software on the evaluation boards some preliminary steps must be performed to integrate the IDE and hardware.

1. Switch on the board power supply
2. Connect USB cable to the PC
3. Start the Crosscore Embedded Studio (Figure 6)
4. Open the selected project (Figure 7 - Figure 11)
5. Select the menu**: Run->Debug configurations** (Figure 12)
    a. Add new item to *Debug with Crosscore debugger* (Figure 13)
    b. In the wizard select the proper processor and debug target (Figure 14 - Figure 16)
    c. Press **Debug** button (Figure 17)
6. Start development of the program using typical tools like: *Resume, Step,* breakpoints, expression and variables watching etc. (Figure 18)

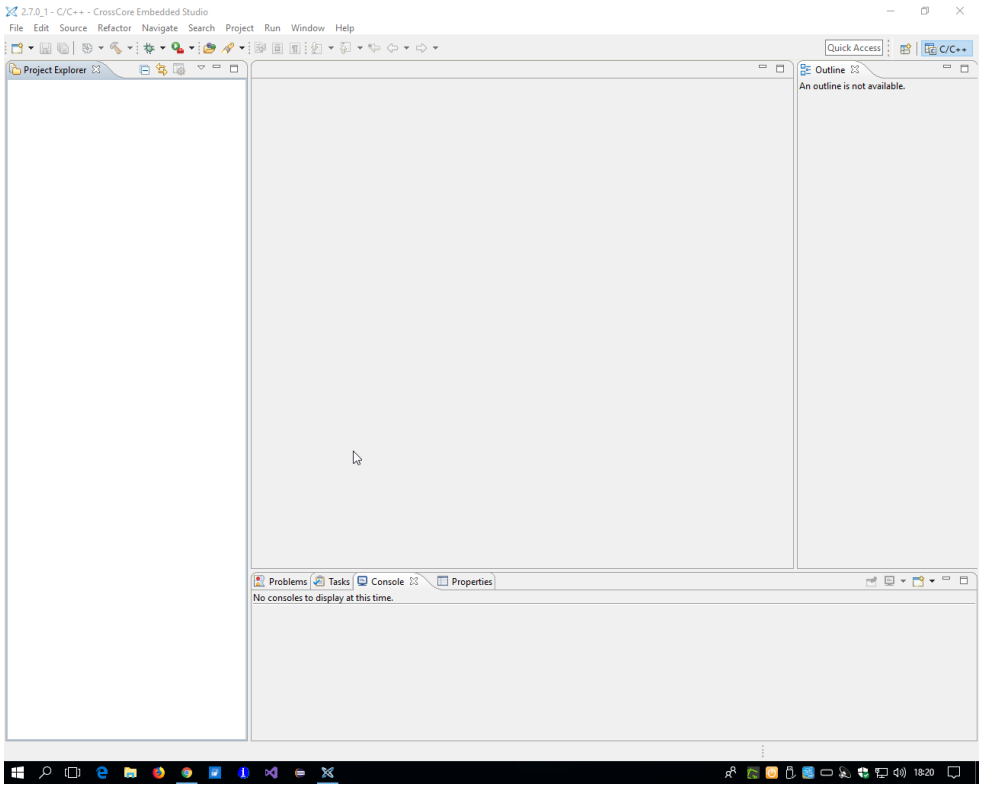


Figure 6 CrossCore Embedded Studio

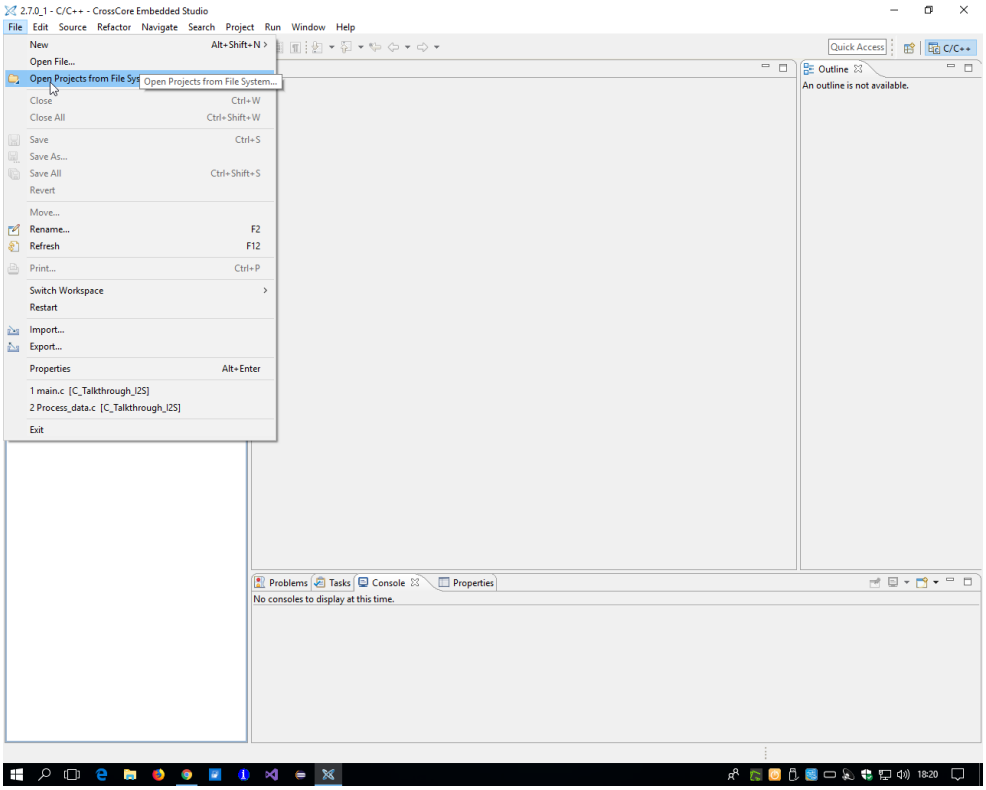


Figure 7 Opening the project - Step 1

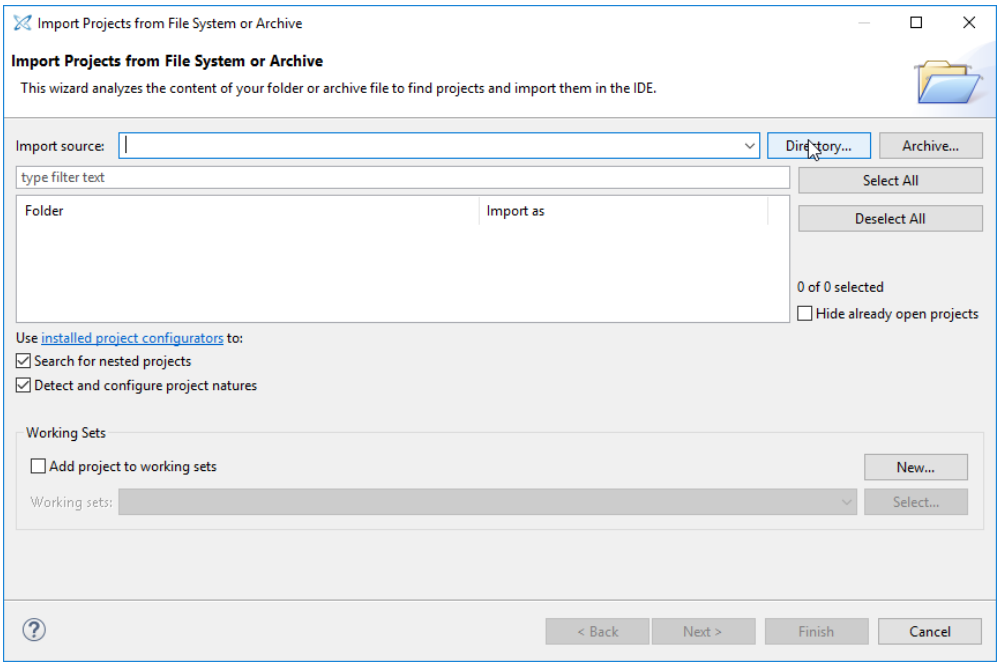


Figure 8 Opening the project - Step 2

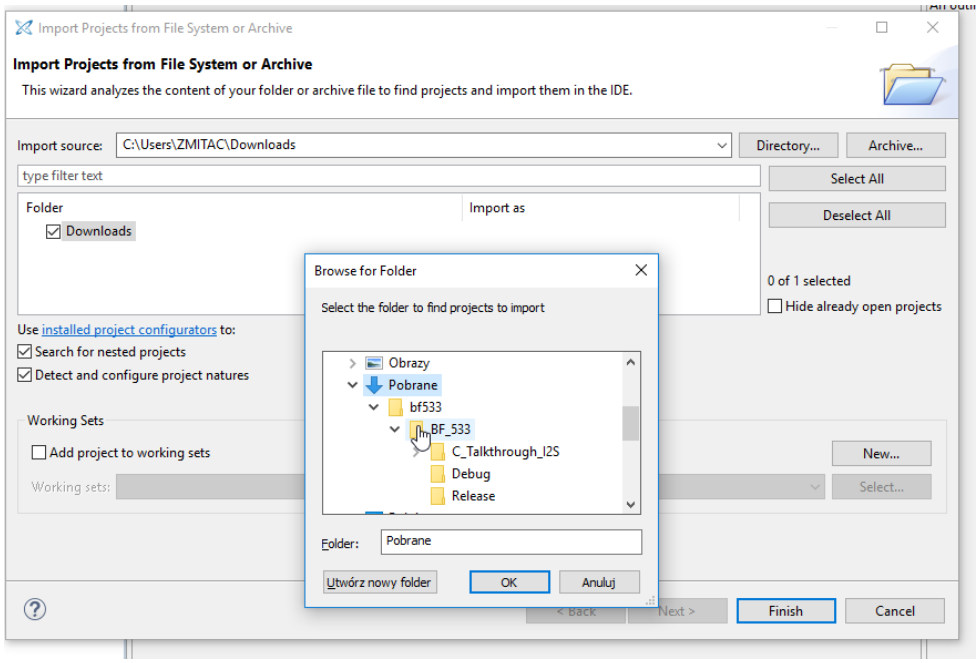


Figure 9 Opening the project - Step 3

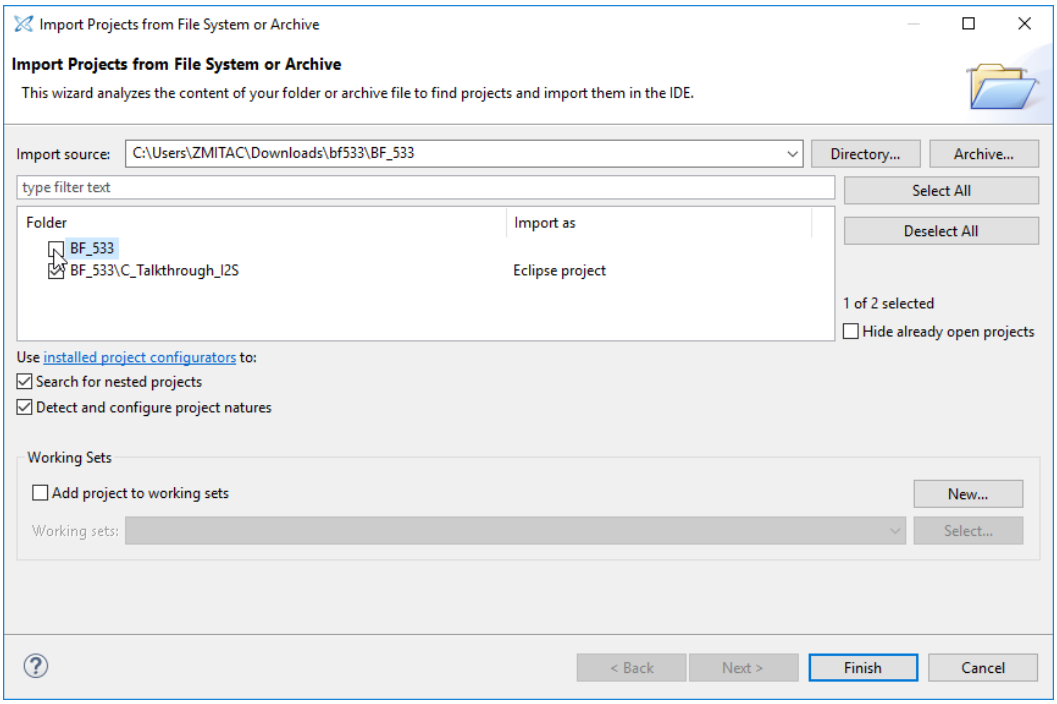


Figure 10 Opening the project - Step 4

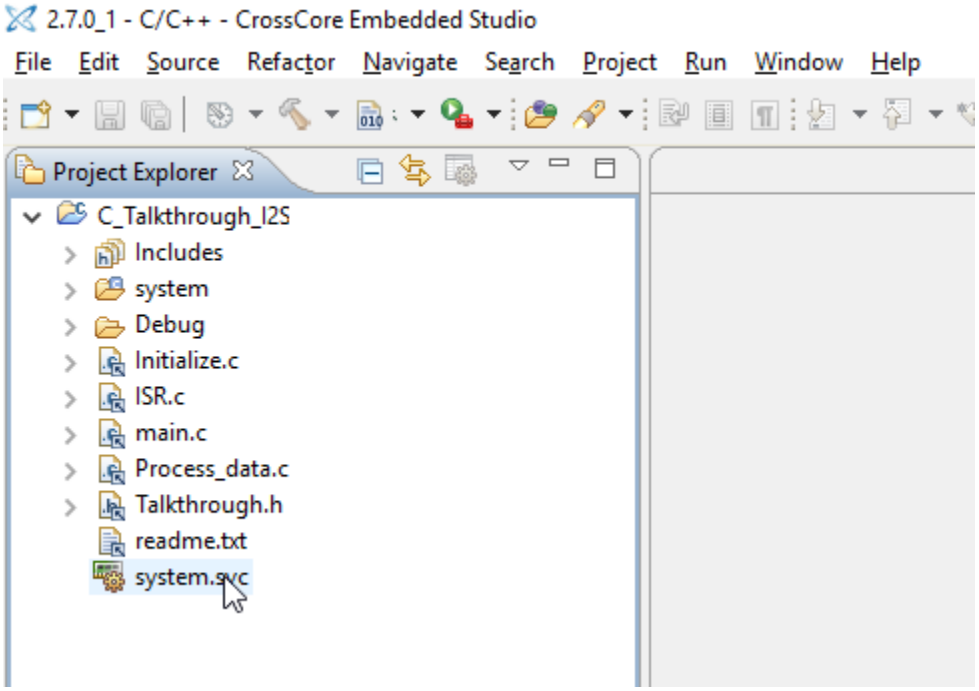


Figure 11 Opening the project – Final

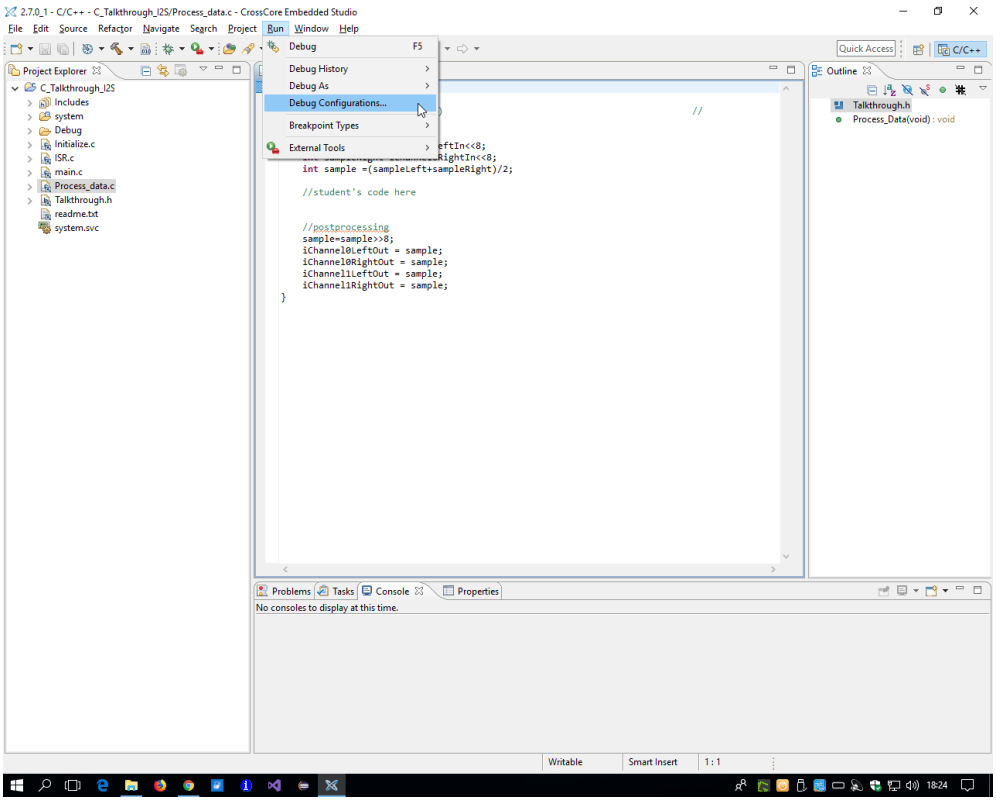


Figure 12 Configuring the debugging environment

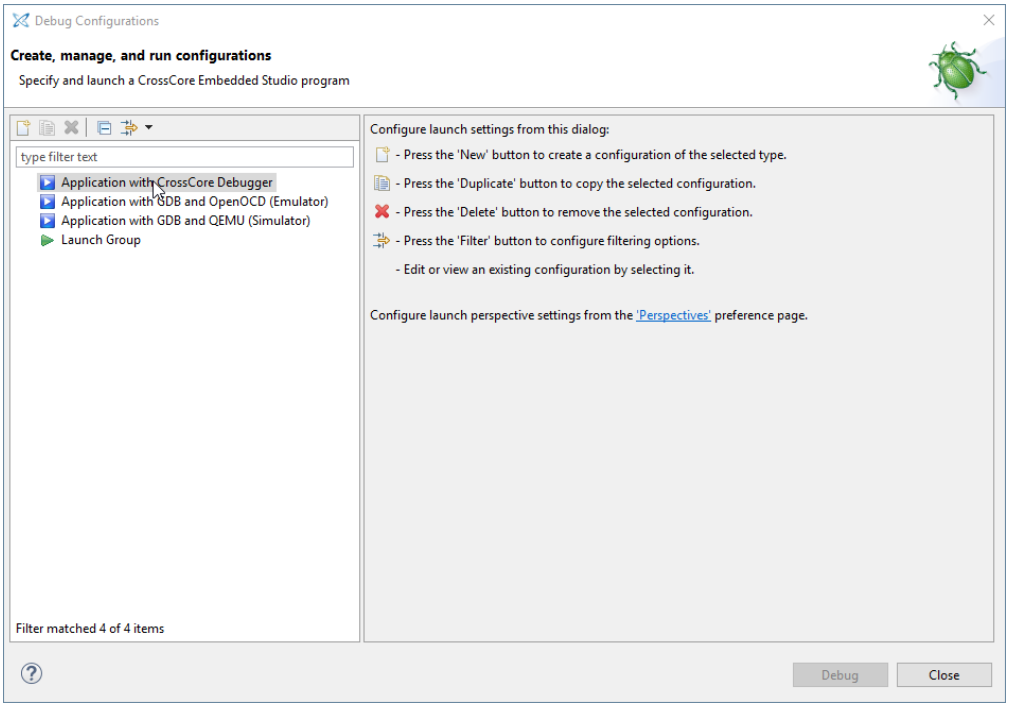


Figure 13 Adding the new debug session

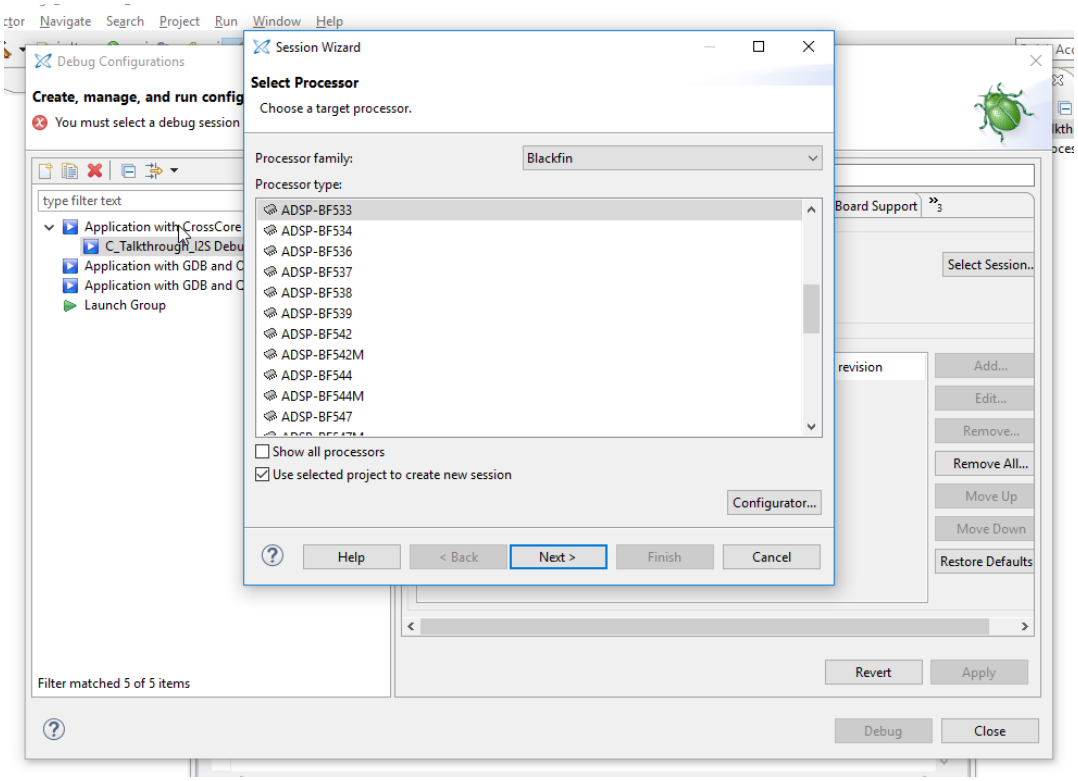


Figure 14 Selecting the proper processor

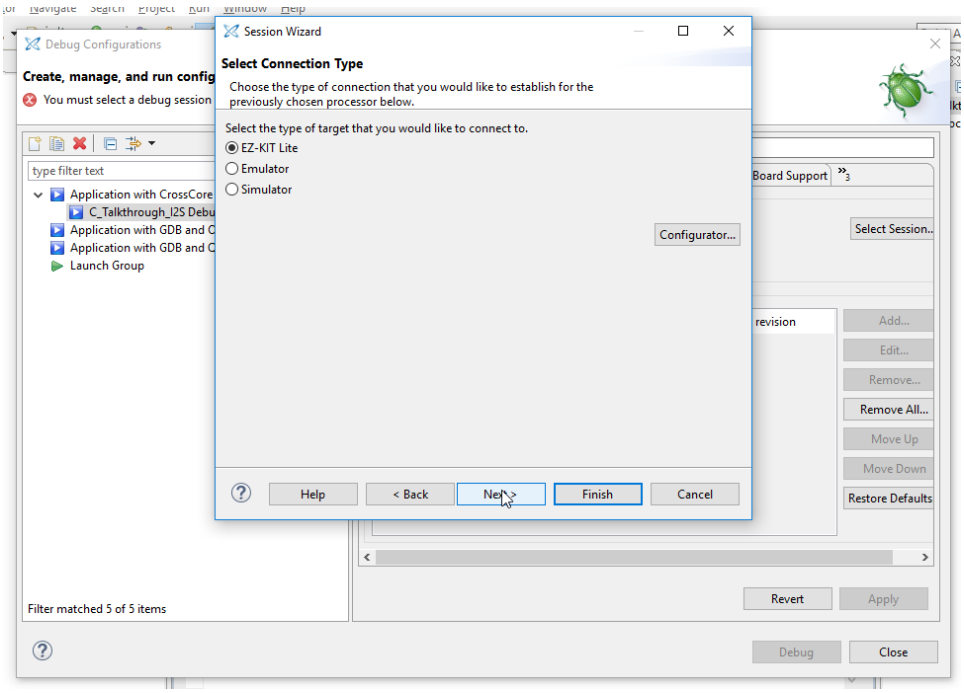


Figure 15 Selecting the proper target

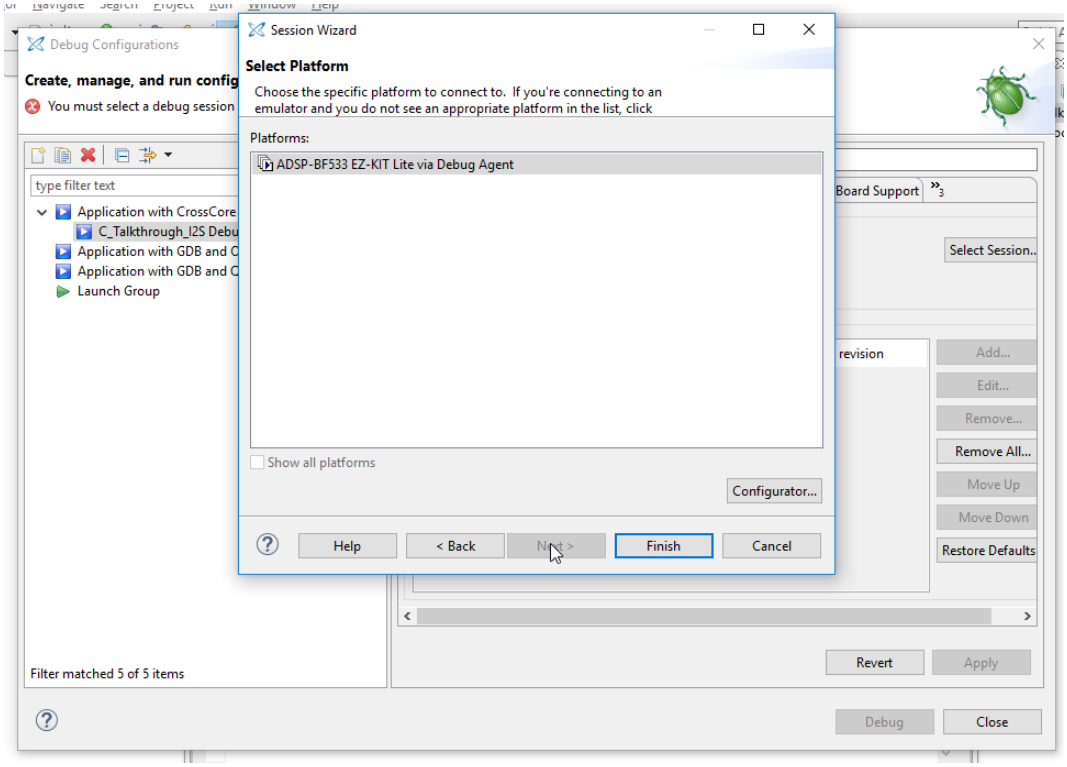


Figure 16 Finalizing the procedure

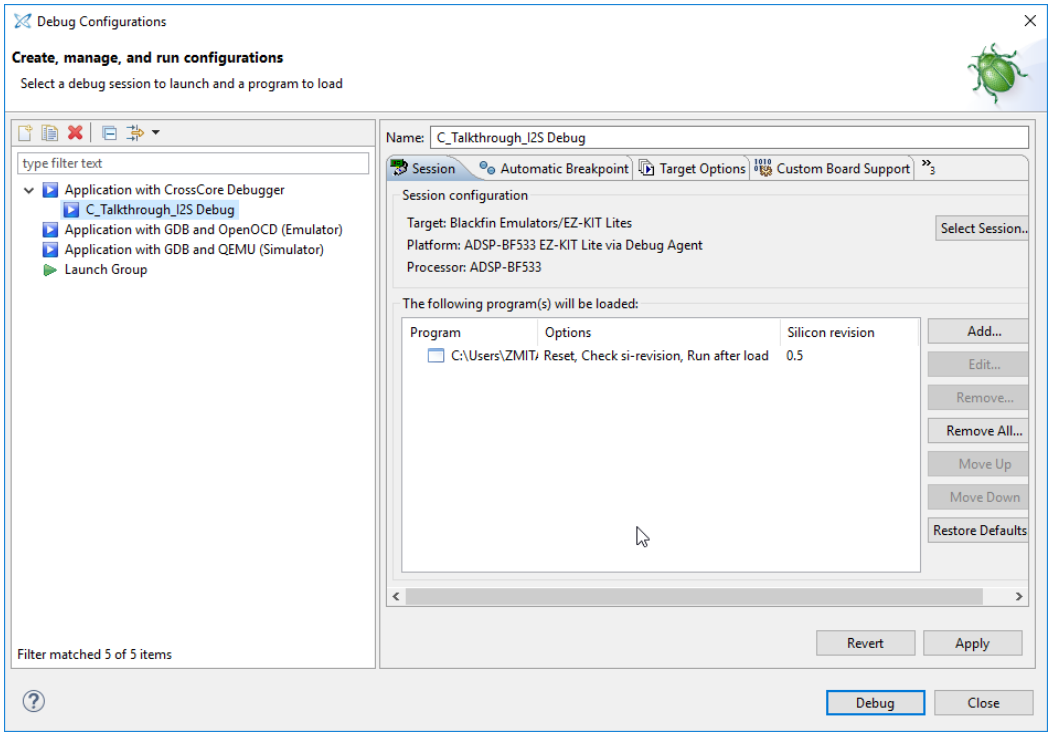


Figure 17 Starting the debugging

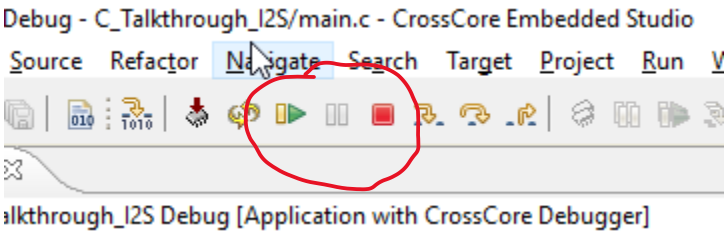


Figure 18 Starting the program

Plot tool

The plot tool allows to graphically visualize content of the physical memory of the processor. If there is a buffer declared and filled with signal samples, it is possible to observe the shape of the signal. In order to configure the plot tool some steps presented in Figure 19 to Figure 25 must be performed.

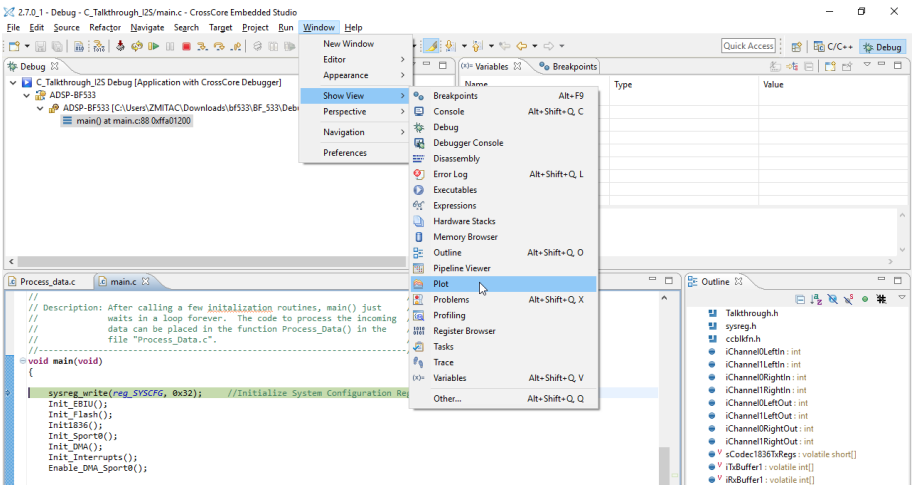


Figure 19 Selecting the plot tool

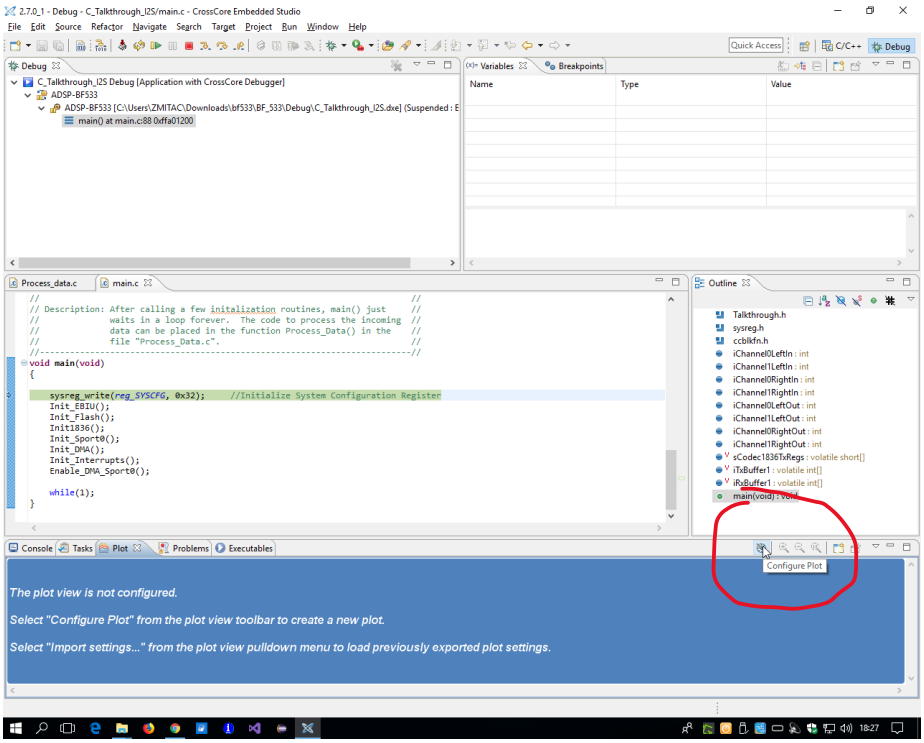


Figure 20 Starting the configuration wizard

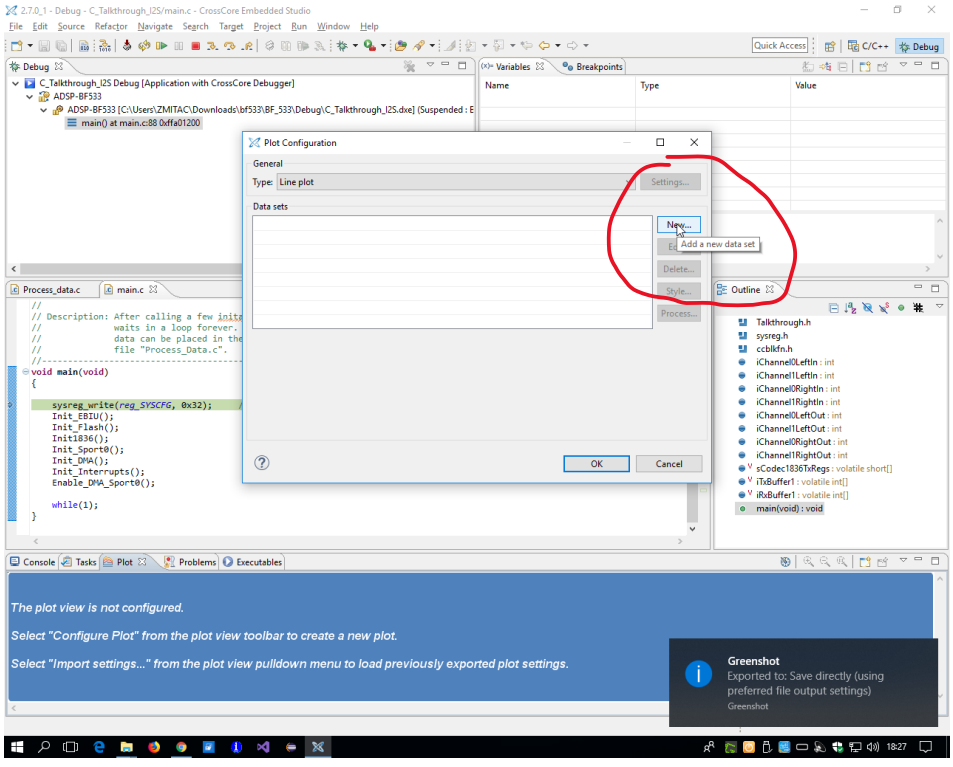


Figure 21 Adding new diagram

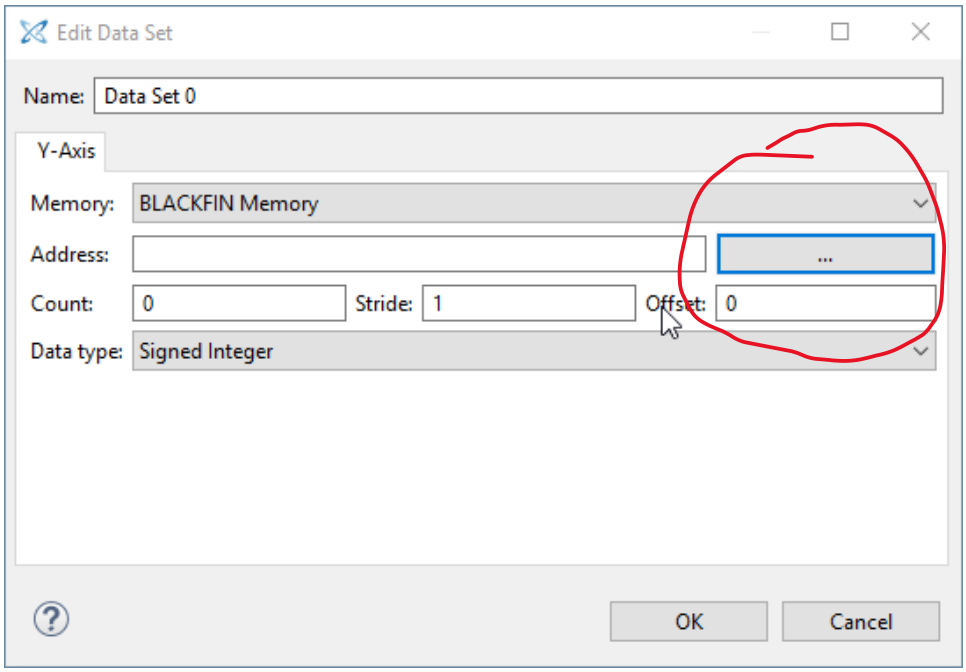


Figure 22 Selecting the address

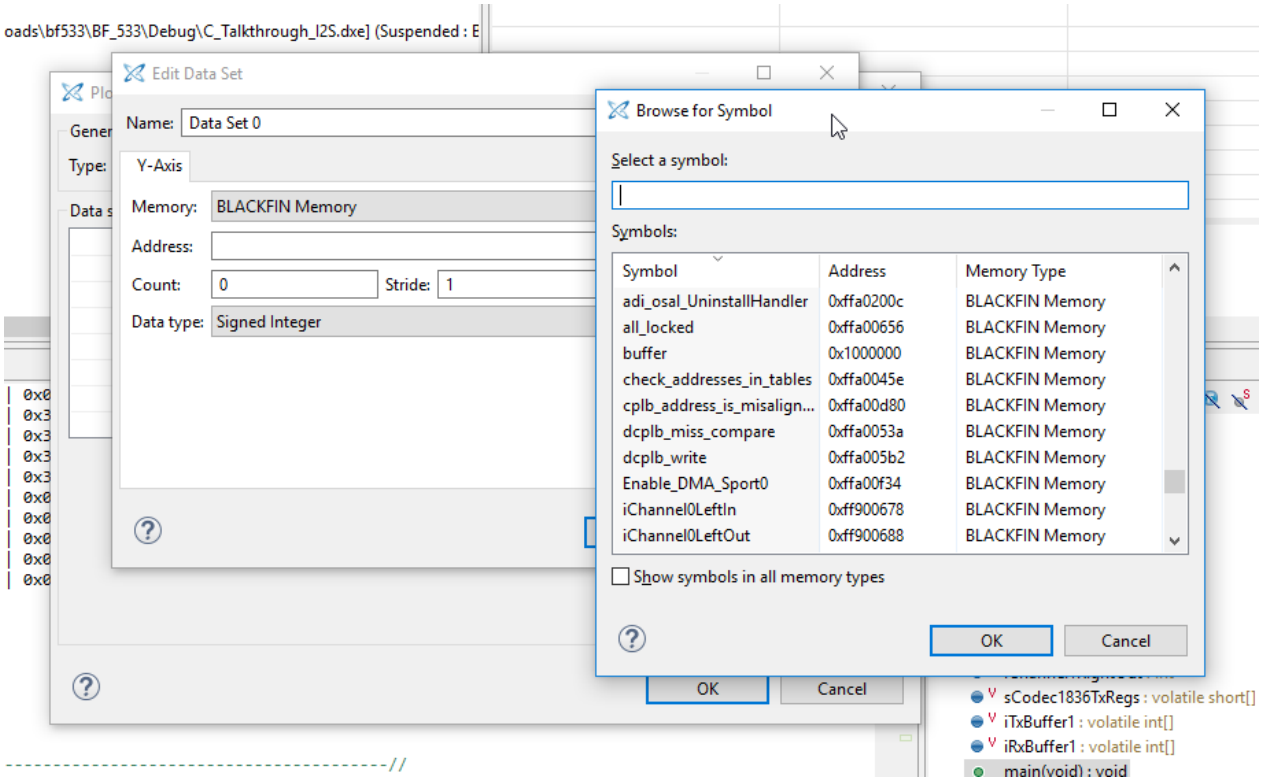


Figure 23 Selecting the address using the name of the buffer

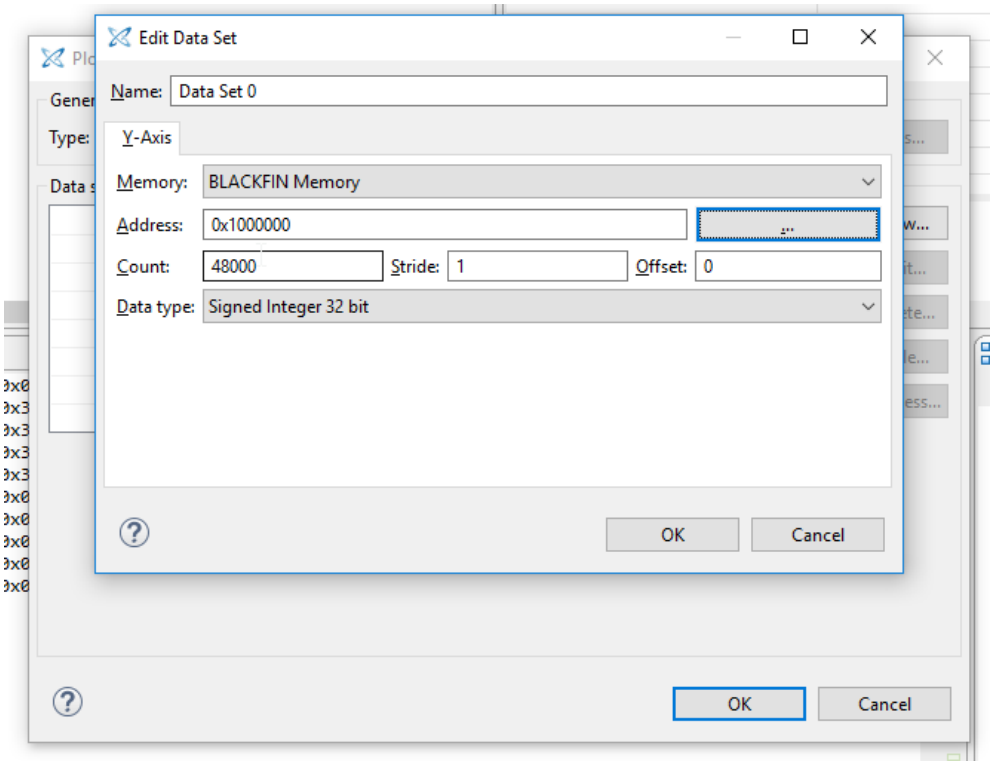


Figure 24 Finalizing the configuration

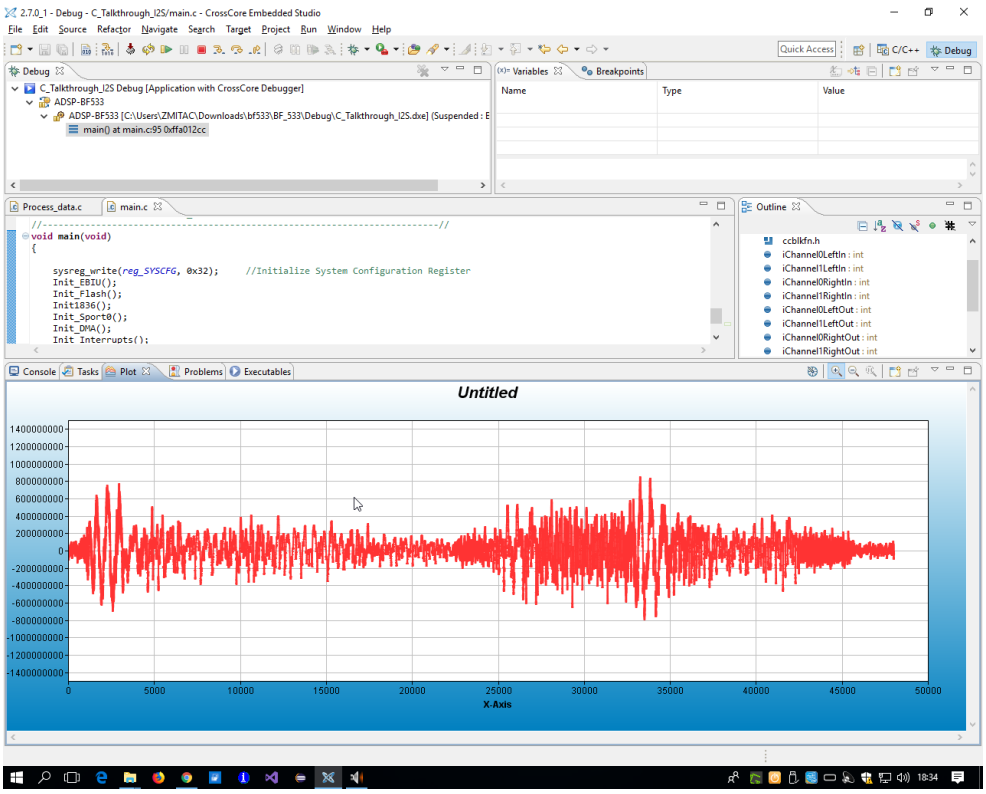


Figure 25 The exemplary signal

The basic project

As the base for the development of laboratory tasks the “Audiocodec Talkthrough” must be downloaded from: http://157.158.56.13/GB. It is important to select the proper project to the given development board. Downloaded archive should be unpacked and imported to the Crosscore Embedded Studio using **File->Open project from file system** menu item.

When the software is executed on the board the following sequence of steps is performed:

1. A/D conversion of the analog audio input signal
2. Transfer of samples to the processor
3. Nothing is done with the samples, they are sent to the D/A converter
4. Analog output signal is sent to the LineOut socket and to the speakers placed on the laboratory stands.

There is the *ProcessData* function where the basic processing of data samples should be performed. This function is called as the interrupt routine with the frequency equal to the sampling frequency of hardware audio codecs placed on the evaluation boards. The content of the function is as follows:

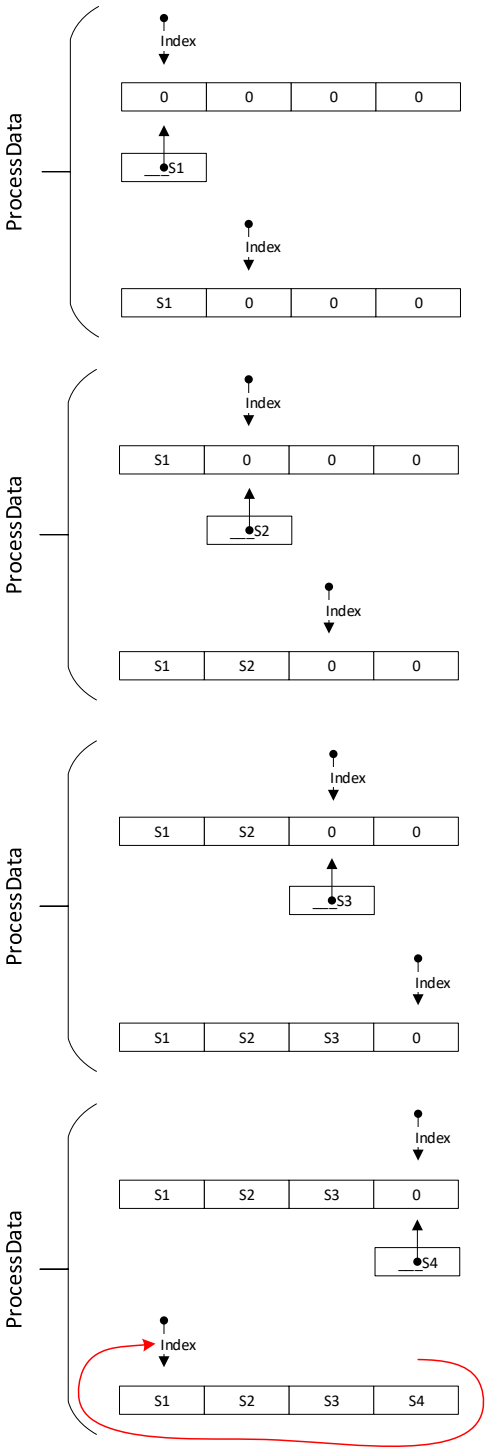
```
void Process_Data(void)
{
    int sampleLeft=iChannel0LeftIn<<8;
    int sampleRight=iChannel0RightIn<<8;
    int sample =(sampleLeft+sampleRight)/2;

    //student's code here


    //postprocessing
    sample=sample>>8;
    iChannel0LeftOut = sample;
    iChannel0RightOut = sample;
    iChannel1LeftOut = sample;
    iChannel1RightOut = sample;
}
```

As can be noticed, in the basic version samples are rewritten from input to the output without processing. Additional code can be places in the body of the *ProcessData* function to perform additional processing of the audio signal.

Laboratory tasks

1. Add the echo / delay effect to the input audio signal.

   Hint. To obtain the delay of signal in the digital domain the circular buffer must be implemented as the Figure 26 presents.


ProcessData
Index
0
0
0
0
S1
Index
S1
0
0
0
ProcessData
Index
S1
0
0
0
S2
Index
S1
S2
0
0
ProcessData
Index
S1
S2
0
0
S3
Index
S1
S2
S3
0
ProcessData
Index
S1
S2
S3
0
S4
Index
S1
S2
S3
S4

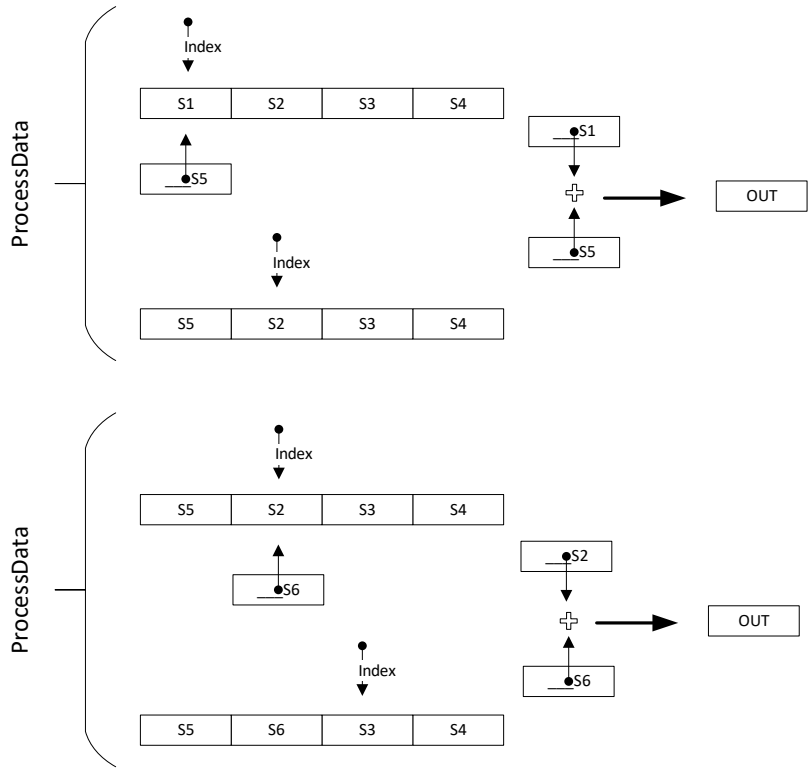
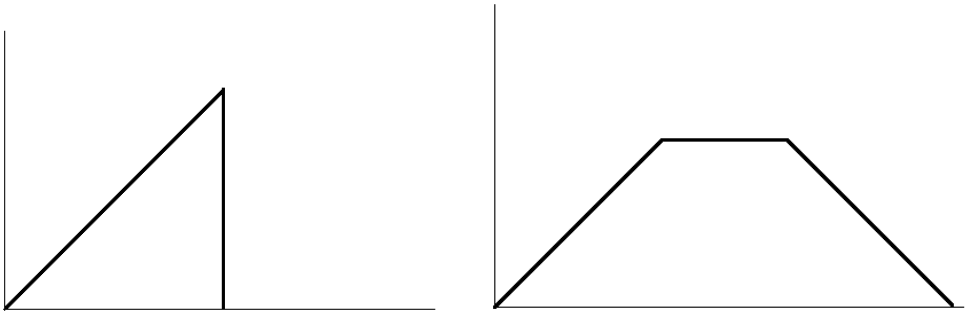
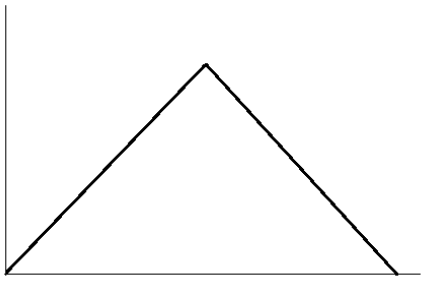


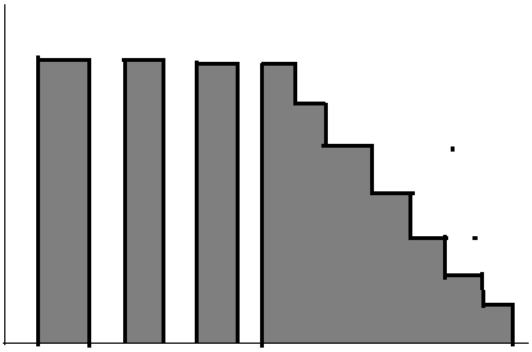
Figure 26 Circular buffer for the delaying of signal

2. Add multiple echo/delay to the input signal
3. Implement the Reverb effect
4. Implement the audio effect given by the supervisor.
5. Generate the sequence of audio signals with different frequencies. The shape of the signal should be one of the presented below





6. Add the envelope to the signal obtained during the task 4. Exemplary shape of the envelope can be as presented below:



7. Perform the task given by the exercise supervisor.

REPORT

The report must contain theoretical description of the implemented audio effect, the well-commented source code of the *ProcessData* functions and other changes made in the raw project, and conclusions.