Analysis of the impact of data resolution on the speaker recognition effectiveness in embedded fixed-point systems

Radosław Weychan, Agnieszka Stankiewicz, Tomasz Marciniak, Adam Dąbrowski
Division of Signal Processing and Electronic Systems
Chair of Control and Systems Engineering
Poznań University of Technology
Poznań, Poland
tomasz.marciniak@put.poznan.pl

Abstract—**This paper presents an analysis of issues related to the resolution of the biometric data of the speech signal processing implemented using fixed-point digital signal processor. Implementation calculations were made using the module with the commissioning of the TMS320C5515 processor, which can be programmed with the use of Code Composer Studio (CCS) environment. We shown features of C5515 digital signal processor that can be used for efficient implementation of algorithms for speech processing. Finally, the influence of data resolution on the accuracy of the calculations and the efficiency of identification is presented.**

Keywords- speaker identification, FFT, fixed point

I. Introduction

Speaker recognition is an interesting feature in human-machine interfaces, where a voice communication is used [1]. Important elements of a realization of speaker recognition with the use of embedded systems are [2,3,4]:

- quality of the input signal and noise reduction
- data resolution during acquisition and processing
- size of memory system
- availability of ready-to-use libraries.

Embedded systems can be based on low cost microcontrollers which have many features that have been characteristic properties of digital signal processors. Even 8-bit microcontrollers produced by, for example, Atmel company have a Harvard architecture and a hardware multiplier that can perform multiplication operations on fixed-point fractional numbers. For example ATmega328 microcontroller, used in the popular Arduino systems, is able to multiply the fractional numbers in two cycles using FMULS instruction that performs 8-bit × 8-bit → 16-bit signed multiplication and shifts the result one bit left [5, 6].

However, it should be noted that the speech signal processing supported by dedicated hardware, is available only in digital signal processors. Examples of such solutions are multiply-accumulate (MAC) instructions (in one cycle), multiple execution units, bit-reversed addressing, and even hardware acceleration of the FFT (fast Fourier transformation) calculations [7, 8]. Another advantage of modern digital signal processors is also optimized power consumption, allowing these systems to work in mobile systems, powered by small batteries.

In this paper the authors have focused on the impact of data resolution. During experimental studies we have used low cost TMS320C5515 eZDSP USB Stick module [9], briefly described in Chapter II. Chapter III examines the FFT calculation, while Chapter IV shows an influence of data resolution on the speaker recognition process.

II. DSP Module with TMS320C5515

The C5000 family of low-cost microprocessors from Texas Instruments is characterized by low power consumption, a relatively large internal memory and a rich set of communication interfaces. Because of the high data processing speed (clocked to 300 MHz) electronic modules based on C5000 processors are ideal for applications such as voice processing.

The TMS320C5515 is a 16-bit architecture, fixed-point microprocessor. It is clocked with a maximum clock rate of 120 MHz, as in the case of instructions executed during one and two clock cycles, giving a minimum execution time of 8.33 ns. This chip is equipped with two arithmetic logic unit (ALU) and two multiply-accumulate (MAC) units, allowing for up to 240 million multiplication and accumulation in 1 second (MMAC). The memory consists of 320 kB RAM and 128 KB of ROM. Internal memory allows for the acquisition of up to 20 seconds of speech on the assumption that all the RAM will be used to collect data with a resolution of 16 bits at a rate of 8000 samples per second. In the case of the module TMS320C5515 USB Stick eZDSP, 4 MB non-volatile flash memory is available [9].

The microprocessor has a built-in 4-channel analog-to-digital successive approximation converter (SAR ADC) with a resolution of 10 bits, working at a rate of 62.5 ksps (32 clock cycles clocked at 2 MHz to convert a single input voltage). Typically 10 bit resolution is too small for audio applications, hence the module TMS320C5515 eZDSP USB stick has an integrated audio encoder TLV320AIC3204 communicating with the microprocessor using a serial interface I2S (Inter-IC-Sound). The system operates at a rate of from 8 to 192 ksps, sending a mono or stereo data with a resolution of 16 bits [10].

The module with fixed-point processor TMS320C5515 includes FFT hardware acceleration. This unit supports DFT calculation with the length from 8 to 1024 points. With the use of external DSP core computation of greater than 1024-points FFTs can be performed [11]. It also has to be noted, that programmer should have a knowledge about hardware – memory allocation, data acquisition requirements, system configuration (DSP, interfaces and audio registers). In this case it is important to proper set PLL for audio data sampling frequency and memory allocation for transform coefficients. They have to be allocated in RAM (DARAM or SARAM) memory as Int32 variables, where 16 most significant bits represent real part, and 16 least significant bits are imaginary part of complex coefficient. Hardware acceleration is a result of using the look-up table, which contains set of 512 complex twiddle factors $W_N = e^{-j\frac{2\pi}{N}}$ (N-point FFT). For 512 and less-point transform a decimated subset of values are used.

In order to compute FFT coefficients, an input vector of time-domain values has to be properly prepared by grouping them in bit-reversed order as preprocessing to butterfly operation. Texas Instruments provides optimized assembly function `_hwafft_br`, which copies the values to bit-reversed addresses in memory. According to the transform resolution N, address of first input sample in memory has to be allocated in RAM such that $\log_2(4 \times N)$ zeros appear in the least significant bits. Moreover, input data vector has to be allocated as Int32 where 16 most significant bits represent real part, and 16 least significant bits imaginary part, which are zeros in this case. Proper memory allocation can be done for example in CMD file. In MEMORY section address of the block beginning and it's length have to be declared, while in SECTIONS structure actual variable label is being allocated at the beginning of the memory block. Label has to be declared in .c file as follows:

```
#pragma DATA_SECTION(data_br, "data_br");
Int32 data_br[128]={0};
```

Similar restrictions concern input data for the FFT computation. It is recommended to use pointer to the same address where bit-reversed address data are placed to overwrite them. Similar to bit-reverse operation, there is dedicated function `hwafft_Npts` (N-point transform) provided by Texas Instruments which computes the FFT coefficients. It is an assembly function calling bit-reversed data addresses and look-up table in coprocessor with twiddle factors to make butterfly operation done. It is possible also to call `hwafft_Npts` function from static ROM memory in order to save approximately 4 kB RAM memory. Output of the FFT function has about 5 clock cycles additional latency and provides failed data for first coefficient in first computation cycle.

In the case of 16-bit multiplication, the `hwafft_Npts` function allows to scale down output of every butterfly operation by factor of 2, which gives much better results (in case of noises and dynamic range) than scaling down time-domain input signal.

III. IMPACT OF DATA RESOLUTION ON FFT CALCULATION WITH THE USE OF HARDWARE ACCELERATOR

During experimental tests, two random voice signal frames have been used. They have been recorded with the rate of 8000 samples per second frequency and 16-bit resolution. Due to the destination of algorithm to speaker identification, where length of acquired frames is an equivalent of 16-20 ms, 128 samples of voice signal have been chosen (16 ms). The vector of input samples has been used to compute 128-point FFT.

In the first step the FFT coefficients have been calculated for signal with the resolution of 16 bits with the use of fixed-point DSP unit and then compared with result for floating-point representation in Matlab environment. It is presented in Fig. 1. To make two of plots comparable, second one have been scaled down by factor of 128 in case of scaling flag used on fixed-point DSP C5515. In the case of no scaling flag, the same computations are presented in Fig. 2.

In the second step of experiments, two various sets of 10-bit resolution data have been used. They have been processed with the use 16 bits fixed-point arithmetic (C5515) and compared with output of floating-point FFT implementation in Matlab environment. Computation on DSP have been done without scaling factor. Result of the comparison is presented in Fig. 3 and Fig. 4.

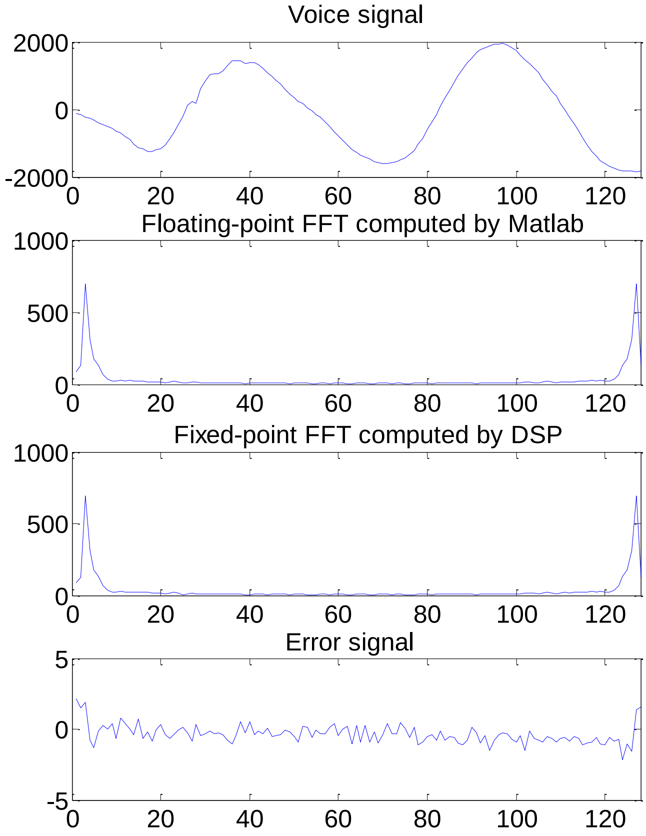

Figure 1. 16-bit representation (with scaling)

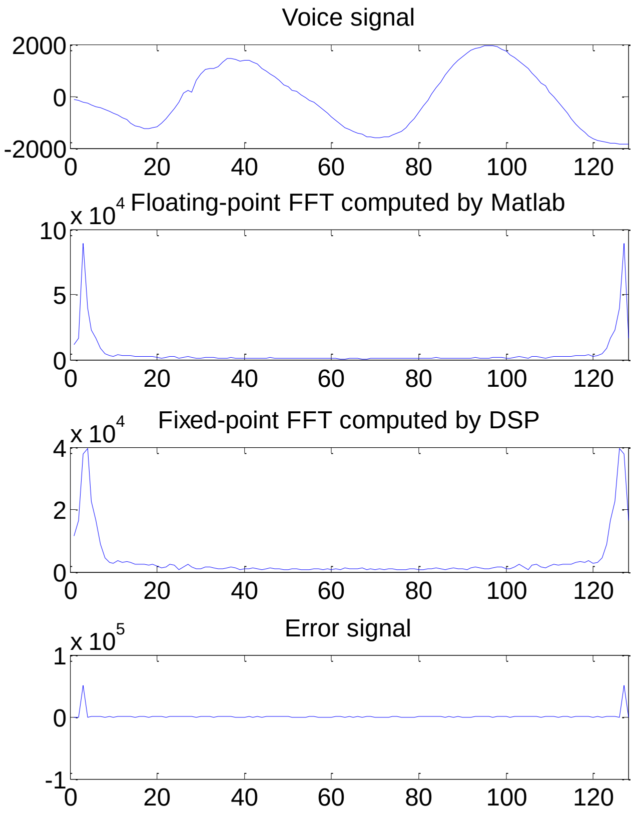

Figure 2. 16-bit representation (without scaling)

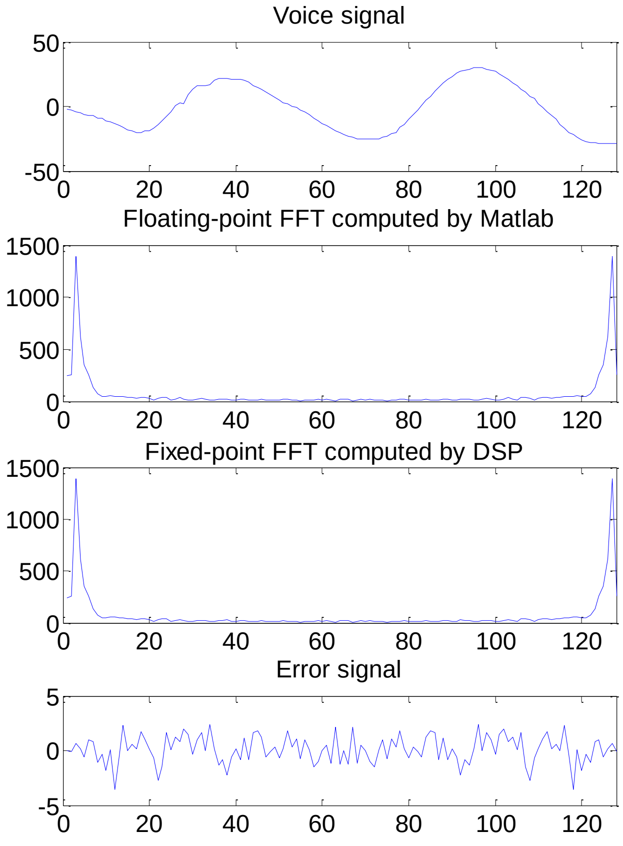

Figure 3. 10-bit representation (without scaling)

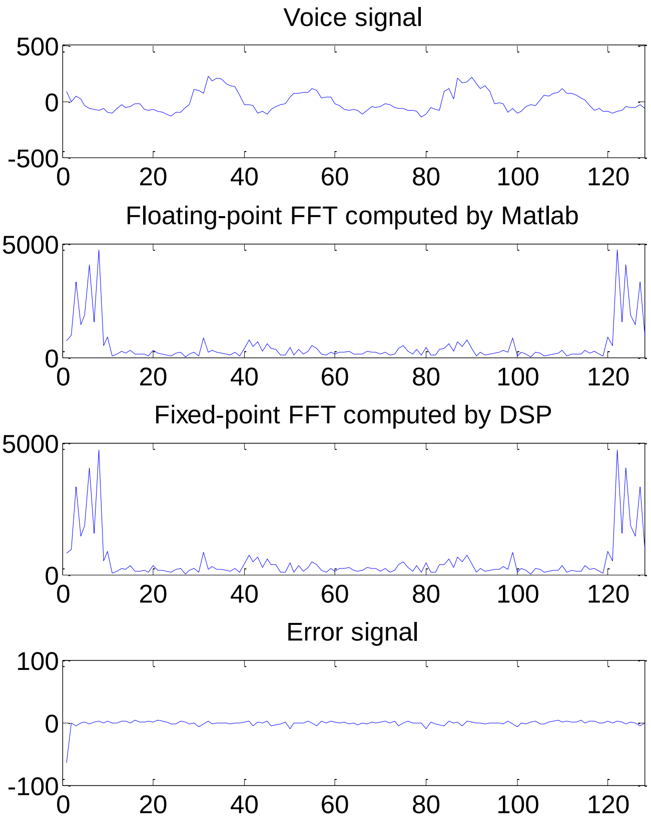

Figure 4. 10-bit representation (without scaling)

Figure 1 presents correctness and high precision of computing the FFT with the use of fixed-point arithmetic. Calculated error signal is a result of different arithmetic and is two orders of magnitude lower than useful information of the FFT. It shows also necessity of using the scaling operation for butterflies. When this operation is not provided, useful information is being lost in case of maximum data range. It is presented in Fig. 2.

In the case of 10-bit resolution data, results are presented in Fig. 3 and 4. It shows the validity of using 10-bit ADC. In this case error signal is more than two orders of magnitude lower than useful information of the FFT. It is also not necessary a scaling operation for butterfly in the case of dynamic range of input signal, which can be written using 8 bits for speech. Error for the first FFT coefficient, which can be seen in Fig. 4 is a result of processing data first time. In this case it does not have any information.

Another part of experiments was an analysis of required FFT calculation time for fixed-point representation. In the case of 128 input samples, bit-reverse operations took 679 clock cycles, while FFT calculation 1032 cycles in debugging mode under Code Composer Studio environment. For PLL set up to 100 MHz. 1711 clock cycles gives 17.11 ms to calculate 128-point FFT. However, in the documentation [7] there is a benchmark of 912 clock cycles (279 for bit-reverse and 633 for the FFT) while PLL was also set up to 100 MHz. In this case full FFT computation should take 9.22 ms.

Texas Instruments have also provided the library DSPLIB for C55X devices [12]. This library includes over 50 C-callable assembly-optimized functions for digital signal processing. One of them is the FFT, which can be used for real and complex data input. It should be noted, that the output of this function is in bit-reversed order. The FFT function corresponding to our case is 16-bit, 128 point `CFFT`. It takes 2516 clock cycles with scaling and 2211 without scaling factor. The bit-reverse function `CBREV` takes 310 clock cycles. Comparing the values to hardware accelerated FFT, it is up to about 3 times more time consuming (according to benchmark from [7]). It can be also noticed, that scaling operation does not influence time of operations in case of hardware acceleration. It is a very important issue for real-time processing in embedded systems.

IV. Speaker recognition using data resolution of 10 and 8 bits

The influence of presented processing was tested in speaker recognition application. Characteristic features extracted from sound wave were encoded with MFCC (mel-scale cepstral coefficients). Finally to create a speaker model Gaussian Mixture Models technique was used. Details of this application and its features were described in [1]. The database used for this purpose includes 40 speakers repeating 6 phrases 30 times, while every utterance lasts about 1 s.

Figure 5 shows the DET (detection error tradeoff) plot of the results. It can be inferred, that reducing the number of bits representing the input signal from 16 to 10, has a significant impact on the accuracy of speaker recognition algorithm. The EER (equal error rate) for 16-bit representation (of the input signal) is equal to about 34%, while for 10-bit representation exceeds 38%. Furthermore computations made using 8-bit processor and 10-bit input signal gives 56% EER. The difference in the result of 16-bit and 32-bit processing is almost indistinguishable. Figure 5b shows enlarged fragment of the plots. The designation 'raw input' stands for speech unprocessed with any silence removing algorithm.

During the experiments, application of silence removal algorithms were tested, as a mean to improve the results of speaker recognition with decreased data and processing resolution. Two EPD (end-point detection) methods ("middle energy" and "middle Jang HOD") were used to detect silence in the input utterance. They were described in detail in [13,14].

Figures 6 and 7 show the results of speaker recognition application for input data with removed silence by respectively middle energy and middle Jang HOD algorithms.

From Figures 5, 6 and 7, it can be inferred, that application of any silence removal method can significantly improve the accuracy of speaker recognition algorithm. Although 10-bit input data gives initially worse results, by using silence removal the outcome of processing is as if 16-bit input data was used. Thus, the use of 10-bit A/D converter for the input data is applicable without diminished correctness of speaker recognition system.

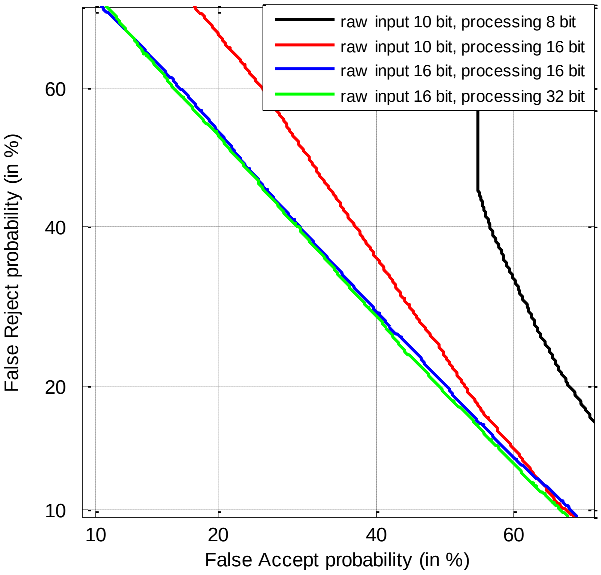

Figure 5. Results of speaker recognition for selected input signal and processing representation

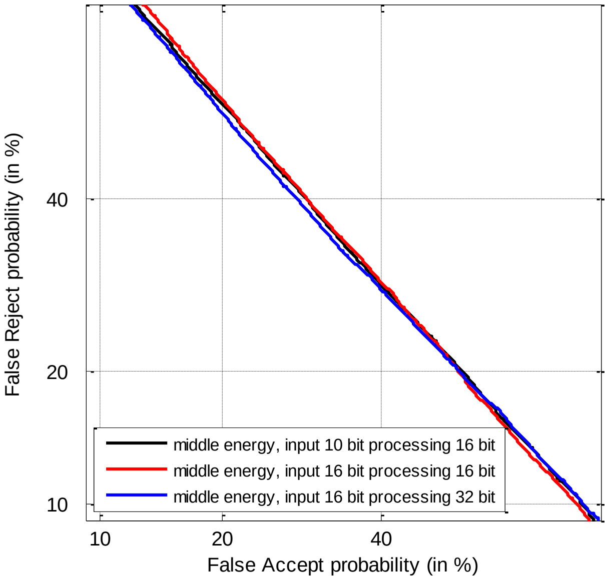

Figure 6. Results of speaker recognition for selected input signal and processing representation with silence removal using middle energy algorithm – enlarged

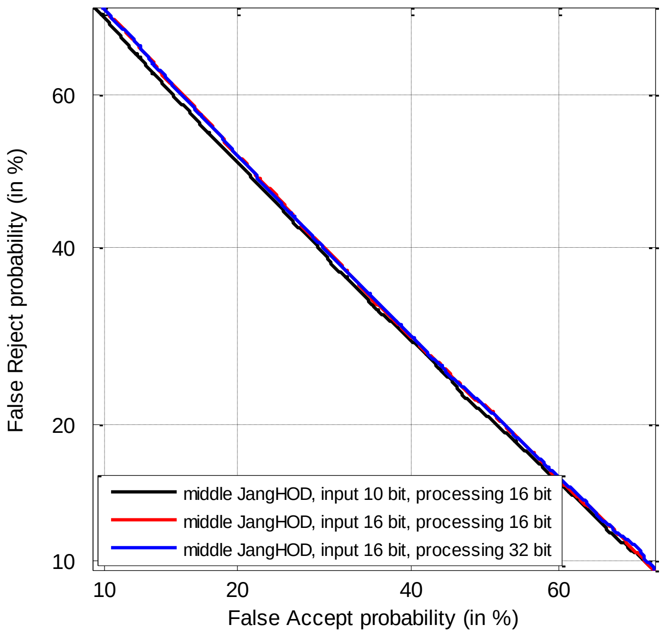

Figure 7. Results of speaker recognition for selected input signal and processing representation with silence removal using middle Jang HOD algorithm

V. Conclusions

Modern microcontrollers and digital signal processors, operating with low power consumption are attractive elements for the construction of control and signal processing systems.

Built-in analog-to-digital converter allows the implementation of an advanced system at no extra cost. For example, the cost of (previously mentioned) the processor TMS320C5515 is about $11 and the cost of TLV320AIC3204 audio codec is about $4.

Our analysis shows that in the case of 10-bit data recognition speaker is only slightly worse performance (about 4% of EER). It should be remembered that the advanced audio processing system requires a precise calculation of a minimum of 16-bit signal processors. Where precision is not required and the processing speed is not high the use of very inexpensive 8-bit microcontroller systems can be justifiable - cost of the ATmega 328P is only about $2.

References

[1] T. Marciniak, R. Weychan, Sz. Drgas, A. Dąbrowski, A. Krzykowska, “Speaker recognition based on short Polish sequences,” SPA 2010: Signal Processing Algorithms, Architectures, Arrangements, and Applications Conference Proceedings, IEEE, Poland Sect, 2010, pp. 95-98.

[2] Y. S. Moon, C. C. Leung, K. H. Pun, “Fixed-point GMM-based Speaker Verification over Mobile Embedded System,” Proceeding WBMA '03 Proceedings of the 2003 ACM SIGMM workshop on Biometrics methods and applications, pp. 53-57

[3] Zhenling Zhang; Yangli Jia; Guang Xie, “Design and implementation of speaker recognition system,” Software Engineering and Service Science (ICSESS), 2011 IEEE 2nd International Conference on , 2011, pp.559-562.

[4] M. Lizondo, P. D. Agüero, A. J. Uriz, J. C. Tulli, E. Lucio Gonzalez, “Embedded speaker verification in low cost microcontroller,” Congreso Argentino de Sistemas Embebidos 2012. Buenos Aires, Argentina. 15-17 Agosto, 2012.pp. 128-133.

[5] 8-bit AVR Microcontroller with 4/8/16/32K Bytes In-System Programmable Flash ATmega48PA / ATmega88PA / ATmega168PA / ATmega328P, Rev. 8161D–AVR–10/09.

[6] 8-bit AVR Instruction Set, Atmel, Rev. 0856I–AVR–07/10

[7] FFT Implementation on the TMS320VC5505, TMS320C5505, and TMS320C5515 DSPs (Rev. B) 09 Jan 2013. http://www.ti.com/lit/an/sprabb6b/sprabb6b.pdf

[8] Texas Instruments, *TMS320C5515 Fixed-Point Digital Signal Processor*, SPRS645E VIII 2010, REV I 2012, http://www.ti.com/lit/ds/symlink/tms320c5515.pdf .

[9] Spectrum Digital, *TMS320C5515 eZDSP USB Stick Technical Reference,* 512845-0001 Rev A II 2010 http://support.spectrumdigital.com/boards/usbstk5515/reva/files/usbstk5515_TechRef_RevA.pdf

[10] TLV320AIC3204 Application Reference Guide, Texas Instruments, SLAA557 – November 2012 http://www.ti.com/lit/ml/slaa557/slaa557.pdf

[11] Woon-Seng Gan; Seth, A.; Kuo, S.M., “Versatile and portable DSP platform for learning embedded signal processing,” IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2011, pp.2888-2891.

[12] TMS320C55x DSP Library Programmer’s Reference, Texas Instruments, SPRU422J - Rev. V 2013, http://www.ti.com/lit/ug/spru422j/spru422j.pdf

[13] T. Marciniak, R. Weychan, A. Dąbrowski, A. Krzykowska, “Influence of silence removal on speaker recognition based on short Polish sequences,” SPA 2011: Signal Processing Algorithms, Architectures, Arrangements, and Applications Conference Proceedings, IEEE, Poland Sect, 2011, pp. 159-163.

[14] T. Marciniak, R. Weychan, A. Krzykowska A., “Speaker recognition based on telephone quality short Polish sequences with removed silence,” Przeglad Elektrotechniczny 2012, vol. 88, no. 6, pp. 42-46.

This work was prepared within the INDECT and DS projects.