www.czasopisma.pan.pl PAN POLSKA AKADEMIA NAUK www.journals.pan.pl

Metrol. Meas. Syst., Vol. XVII (2010), No. 4, pp. 589–598

PAN POLSKA AKADEMIA NAUK

METROLOGY AND MEASUREMENT SYSTEMS

Index 330930, ISSN 0860-8229

www.metrology.pg.gda.pl

M&MS

MANAGEMENT OF MEMORY IN A REAL-TIME MEASUREMENT SYSTEM BASED ON A SIGNAL PROCESSOR

Łukasz Ćwikliński, Wiesław Kiciński

Nicolaus Copernicus University, Faculty of Physics, Astronomy and Informatics, Grudziądzka 5, 87-100 Torun, Poland (lukasz.cwiklinski@fizyka.umk.pl, ✉ w.kicinski@fizyka.umk.pl, +48 56 611 3343)

Abstract

The problem of management of memory in a signal processor has been discussed on the example of time parameters measurement system of transient signals. General rules of memory management and allocation in TMS320C6713 DSK have been described.

Keywords: memory management, signal processor, measurement system.

1. Introduction

Modern measurement systems with advanced algorithms of processing a signal in real-time can be successfully designed using signal processors thanks to their functional features, such as a superscalar architecture, pipelining, rich instructions list executable in one time cycle and the embedded floating-point arithmetic. These features allow an easy and time-efficient implementation of typical operations for digital signal processing like: the Fast Fourier Transform, filtration, convolution, correlation *etc*. Moreover, their architecture enables a fast exchange of data in which specialized blocks play an important role by enabling a creation of cyclic buffers, which are useful in analog-to-digital conversion [2, 4].

The key problem in designing systems that work in real-time is memory management. The organisation and the method of memory use determine the speed of exchange of data between functional blocks of a system and, therefore, they have an influence on its dynamic properties.

It is commonly accepted that a minimal number of operations of data exchanges and exchanges of modules of program's code between functional blocks of system is the basic optimization criterion in the problem of memory allocation and organisation. It is worth to notice that the longer the blocks are the shorter the total time of data exchange between functional blocks of a system (due to smaller number of control operations). However, the implementation of large buffers is limited by available resources of fast memories in a system.

The problem of memory allocation in a real-time measurement system is discussed on an example of a system detecting and measuring parameters of hydroacoustic transient signals. Sources of such signals in an underwater environment include biological (*e.g.* diver) and technical objects (*e.g.* underwater weapons). The measurement system is implemented on the TMS320C6713 signal processor that processes signals in real time using the wavelet transform.

The organization of this paper is as follows. Section 2 introduces a general justification of memory management in digital signal processors (DSP). In Section 3 we describe the main functional features of the TMS320C6713 Development System Kit. An example of memory management in a real-time measurement system is given in Section 4. Section 5 contains the concluding remarks.

Article history: received on Aug. 19, 2010; accepted on Oct. 20, 2010; available online on Nov. 26, 2010; DOI: 10.2478/v10178-010-0048-y.

2. General justification of memory management in DSP

The organisation of memory and the method of its management depend on hardware resources of the designed system and the degree of complexity of tasks which the system performs. The method of the allocation and the way of using a memory depend on the structure and the capacity of internal memory of the signal processor. Memory access time has a crucial influence on a system's dynamics.

There are two possible options of management of memory resources:

- the use of a default memory model resulting from the architecture of the signal processor;
- setting up a user-specified memory model when processor's memory resources are extended.

In the second case, memory resources of a system have to be declared in the compiler, using directives of a linker in a command file. These directives, besides a declaration of the memory size and address of its location, also contain a unique memory block's name and the attributes: read/write, initialization and content, *e.g.* the program execution code.

During a compilation of the source file, memory resources can be allocated statically or dynamically. Moreover, memory area for variables can be allocated dynamically as well as during the program execution. Software delivered by hardware manufacturers makes the management of memory resources simpler.

The so-called sections are basic objects used in the management of memory resources. They are assigned to constants, variables, arrays, to a stack, a code of the executed program and also to dynamic memory management [2, 3].

The sections are created by default in the compilation process (taking into account the physical memory resources of the designed system) or they can be user-defined by linker directives. Defined sections do not need to be initialized.

The possibility to define the sections by a user simplifies the process of memory management. The created sections are assigned to processor's memory resources declared in the linker command file.

Memory management in signal processors produced by most leading manufacturers is similar. Some differences include file names, the organization of sections and the method of referring to it.

Manufacturers of signal processors, besides instructions of a dynamic allocation and of freeing memory resources coming from the standard C language (malloc, free), also offer additional instructions. They are extensions of the standard C language, in which it is possible both to declare a variables' size and to allocate memory. However, these instructions have some restrictions, *e.g.* they cannot be used in the interrupt service routine.

3. Management of memory in the TMS320C6713 DSK

The Texas Instruments TMS320C6713 Development System Kit (DSK) is provided with an external SDRAM memory module, the AIC23 audio codec, audio inputs and outputs, and connections for the so-called "daughter cards". The Code Composer Studio software is equipped with a graphical interface DSP/BIOS and contains among others [2, 3]:

- editor of program's source code;
- tools for generation of program's execution code: C/C++ compiler, assembler, linker;
- tools for optimization of application;
- digital signal processing libraries;
- real-time operating system that aids management of interrupts, configuration of input/output systems and running software in real-time.

Metrol. Meas. Syst., Vol. XVII (2010), No. 4, pp. 00–00

The program code and data can be located in any area of memory resources, accordingly to the constraints resulting from different memory access time. The processor's memory map and the system development kit are shown in Fig. 1.

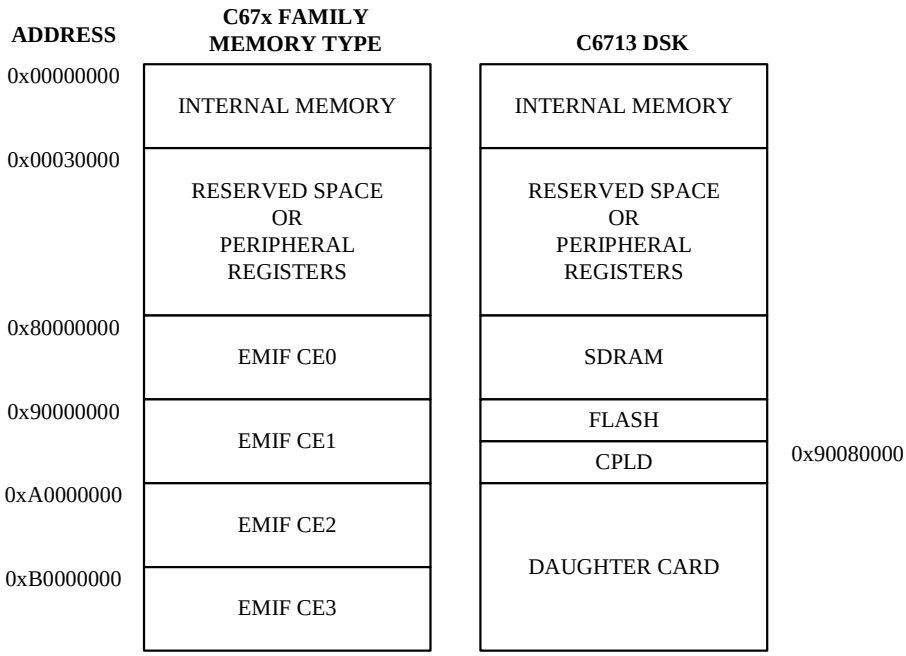

Fig. 1. Memory map for C67x processors family and for TMS320C6713 DSK.

There are following memory resources in TMS320C6713 DSK:

- internal processor's memories (4 KB I level program memory cache, 4 KB I level data memory cache, 64/192 KB II level program/data memory cache/SRAM);
- external memories (512 KB FLASH, 16 MB SDRAM).

The following linker directives can be used for memory management: MEMORY (memory organization) and SECTIONS (sections organization). The declaration of a section has to be preceded by a declaration of available memory resources. The linker configuration file (.cmd) can be created automatically by the DSP/BIOS graphical interface or manually in the text editor. In the first case, sections are created by default and the compiler allocates available memory resources to them. The minimal, default set of sections (Fig. 2) contains [12]:

- .text section – usually contains the executable program code;
- .data section – usually contains program initialization data;
- .bss section – memory space for uninitialized variables.

In case of user defined sections, it is recommended to create a separate .cmd file (without editing of the .cmd file generated by the DSP/BIOS interface). Only when a user declares his own sections, it is recommended to create an additional .cmd file.

Allocation of constants, variables and modules of the program's source code can be performed by the compiler's directives belonging to the #pragma set. These directives allow an allocation of program modules to sections other than the default one from the source code level.

For example, the syntax that allows to assign a variable to a section in memory has the form [2]:

#pragma DATA_SECTION (symbol,"section_name"), where symbol means the name of a variable;

whereas the syntax of a directive that allows to define a variable's size in memory:

#pragma DATA_ALIGN (symbol, constant), where symbol means the name of a variable and $2^{constant}$ means data size. A section's name has to be assigned beforehand to memory in the linker command file.

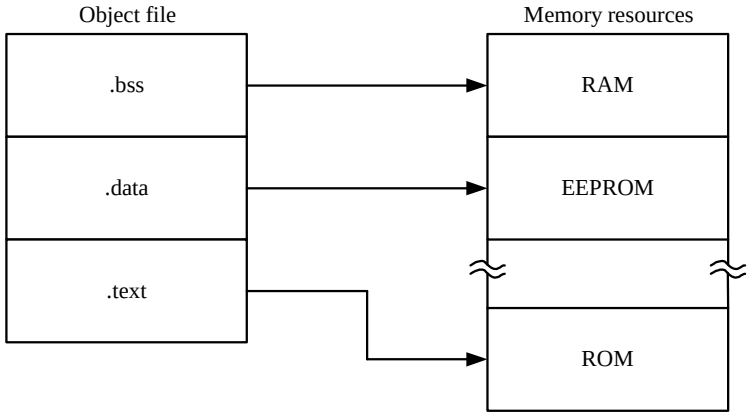

Fig. 2. An example of mapping of sections in processor memory.

4. An example of memory management in a measurement system

The need of a simultaneous service of the signal acquisition module and realization of often time-consuming procedures of signal analysis, imposes requirements on the method of system's memory management. A boundary condition for management of memory resources is the duration of analysis of the measuring signal that has to be shorter than the data acquisition time.

4.1. Procedure of signal processing

The presented measurement system was designed for the measurement of parameters of broadband transient signals in real-time. An example of a broadband transient signal and the

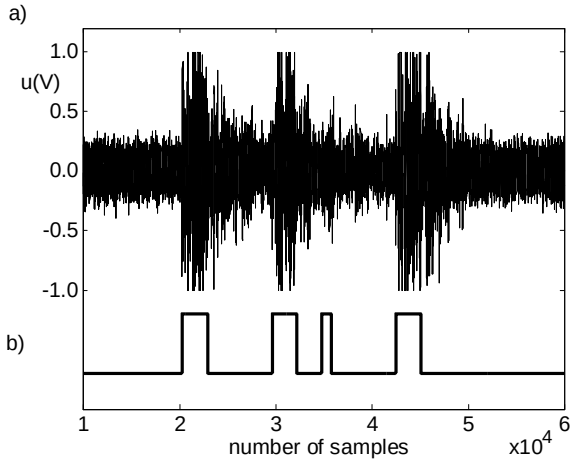

Fig. 3. An example of broadband transient signal: a) time waveform; b) detection curve.

detection curve of nonstationary segments are shown in Fig. 3. The processing of measuring signal includes:

– signal acquisition;
– detection of nonstationary segments;

Metrol. Meas. Syst., Vol. XVII (2010), No. 4, pp. 00–00

– computation of the average period between nonstationary segments of signal and average time of these segments.

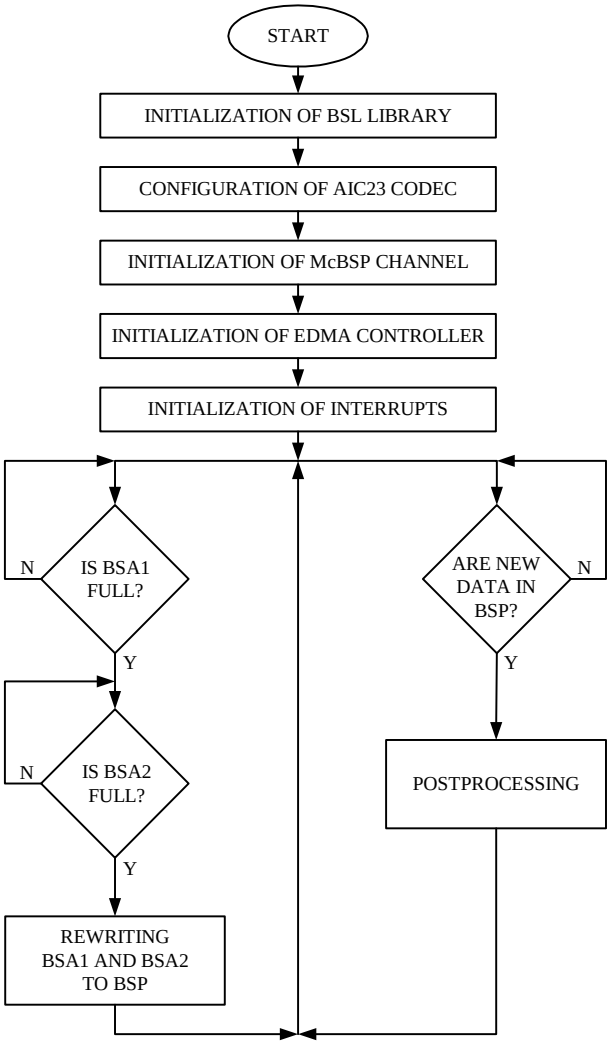

Fig. 4. Block diagram of program implemented in signal processor:
BSL – board support library, BSA – buffer of signal acquisition, BSP – buffer of signal processing.

The program implemented in the signal processor (Fig. 4) is based on the algorithm of nonstationary segments detection described in [6]. The procedure of the wavelet analysis, based on Malvar wavelet, has been implemented in the module of signal analysis [5]. The coefficients of the wavelet transform have been computed using the Fast Fourier Transform (FFT) algorithm.

4.2. Implementation of the algorithm of signal processing

The algorithm of signal processing has been implemented in the TMS320C6713 DSK. The measuring signal has been loaded to the input of the AIC23 Codec and then, after analog-digital conversion, has been processed accordingly to the algorithm described in 4.1.

The series of signal samples (the so-called audio data) from the output of the AIC23 Codec were transferred to the processor through the bidirectional serial port McBSP1 accordingly to EDMA channel service protocol.

To assure continuity of the signal acquisition process, three buffers have been implemented:

– two signal acquisition buffers, each with 4096 samples capacity;
– a signal analysis buffer with 8192 samples capacity.

Audio data in signal acquisition buffers are written alternately. Since the measuring signal is monophonic and the codec has a stereophonic input, every second signal sample in acquisition buffer is redundant.

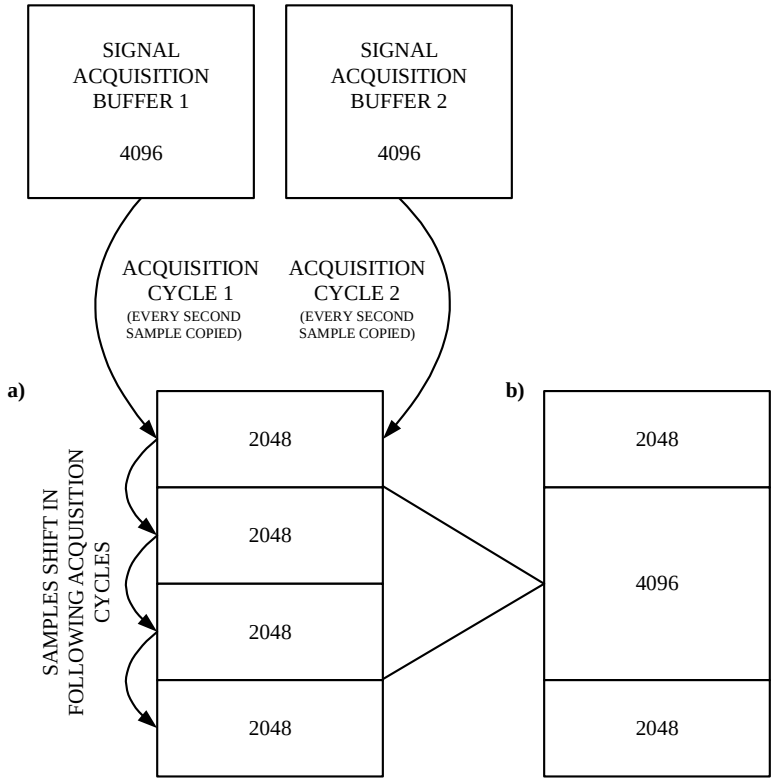

Fig. 5. The relationship between acquisition buffers (a) and signal analysis buffer (b).

Hence, every second sample from the acquisition buffer has been copied to the signal analysis buffer. The synchronization of data exchange between buffers is assured by configuring the EDMA channel. The size of the signal analysis buffer is equal to the width of the signal analysis window increased by the signal samples records, indispensable for folding and unfolding operations accompanying calculations of the Malvar wavelet coefficients [4, 5, 7]. The relationship between the buffers is shown in Fig. 5.

The following procedures are run after filling of the signal analysis buffer: the wavelet analysis, the determination of detection curve and the calculation of the time parameters of the transient segments. The above-mentioned procedures are run parallel with the data written alternately to the signal acquisition buffers. The program implemented in the signal processor consists of the following main modules:

– module of AIC23 Codec configuration;
– module of EDMA (Enhanced Direct Memory Access) configuration;
– module of EDMA interrupt service;
– module of data exchange between buffers;
– module of wavelet transform;
– module of the analyzed signal.

Codec configuration (through serial port McBSP0) includes:

– setting the signal level;
– setting the signal sampling frequency;

– defining the audio data format.

The serial port McBSP1 and one DMA channel have been used for communication between the codec and the signal acquisition buffers. The EDMA channel configuration includes:

– setting the "linking" mode;
– setting priorities of the EDMA channel events service;
– setting addresses of data source (McBSP1 port), its destination (I and II acquisition buffer);
– setting parameters of data exchange blocks (16 bit format, length of block equal to the size of the acquisition buffer).

In the process of program compilation, the linker creates proper sections for execution of code's modules, constants, variables, tables *etc*. and then assigns the system memory resources.

4.3. Memory management

The access time and the memory capacity determine the section assignment in the TMS320C6713 DSK memory resources. Taking this into consideration, the signal acquisition buffers are organized in the processor's SRAM II level internal memory, and the signal analysis buffer in the SDRAM external memory. The program code is located in the external memory.

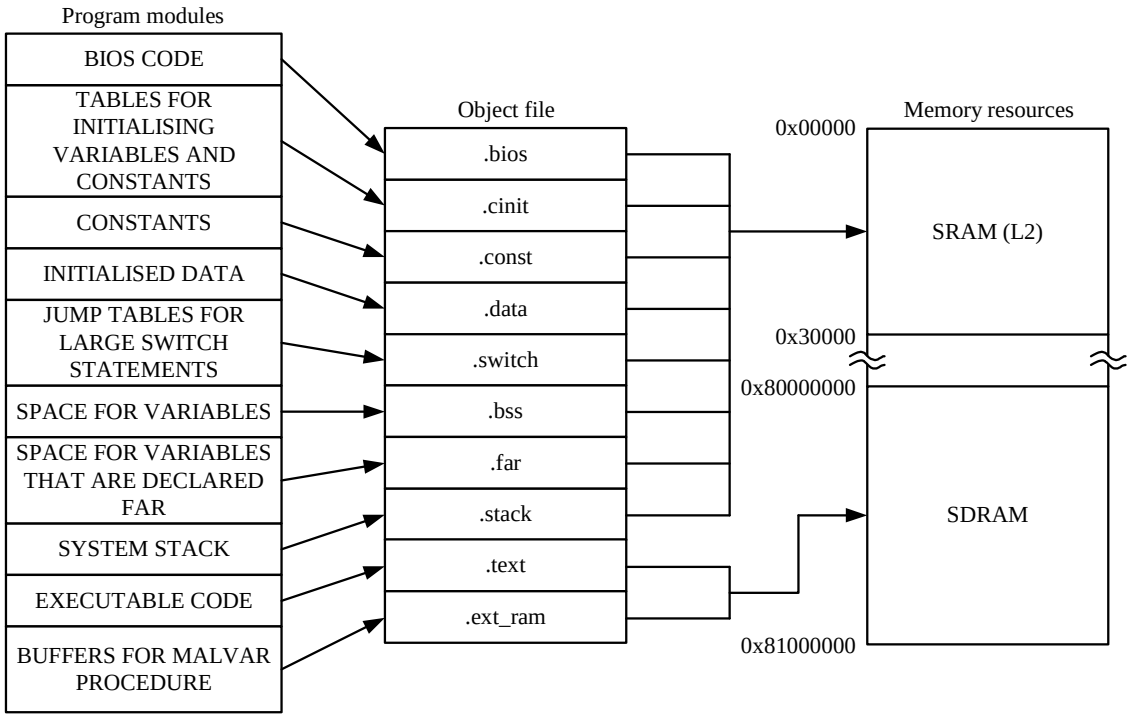

Fig. 6. Program modules and their mapping in sections and memory of development kit.

The program modules and their mapping in sections (Table 1) and memory of development kit are shown in Fig. 6. The section with the program code (.text) and the section containing variables of the procedure of the Malvar wavelet calculation (.ext_ram) are located in the SDRAM memory. The .ext_ram section is created by a user in an additional .cmd file. Other sections automatically created by the linker (in program compilation process), are situated in the 2nd level of the SRAM memory.

Table 1. Main section of program.

Section	Resources
.bios	section contains BIOS code
.cinit	section contains tables for initializing variables and constants
.const	section contains constants of program
.data	section contains initialized data
.switch	section contains tables for large switch statements
.text	section contains all the executable code
.bss	section contains reserved space for uninitialised variables
.far	section contains reserved space for variables that are declared far
.stack	section contains reserved space for system stack
.ext_ram	section contains input and output variables for procedure of Malvar coefficients calculation

5. Conclusions

The problem of management of signal processor's memory in a real-time measurement system has been discussed on the example of a system applied for measurement of time parameters of a transient signal. The organization and the method of memory use determine the speed of data transfer between the system functional blocks, resulting in its dynamic properties. The criterion of the optimization problem of memory allocation and organization is the acquisition time of the measurement system and minimization of the number of data exchange operations and program code modules between system functional blocks is general rules of memory management and allocation in digital signal processors are discussed. Next, the organization and method of memory management in a system based on the TMS320C6713 DSK is described.

In the presented example, from available memory resources of the TMS320c6713 DSK (192 kB SRAM II level processor internal memory and 1.6 MB SDRAM external memory) 130 kB of SRAM II level memory and 177 kB of SDRAM memory are used.

Under the described conditions of memory organization, the following times of procedure realization are acquired:

– procedures of signal analysis – max. 280 ms;
– procedure of Malvar coefficients calculation (8192 point FFT) – 45 ms.

For comparison, if the program code is located in the processors internal memory, the realisation time of Malvar coefficients calculation is shorter by about 9 ms than the time of its realization in the SDRAM external memory [4]. The total time of procedures execution of the signal analysis process is considerably shorter than the acquisition time of 4096 signal samples, which equals 512 ms at an 8 kHz sampling frequency.

Acknowledgments

This work was supported by the Ministry of Science and Higher Education, Poland (grant # N N505 360736).

References

[1] Burnos, P., Gajda, J., Maj, P. (2010). Digital system for detection and location of miners trapped in hard coalmines – glop2. *Metrol. Meas. Syst.*, 17(2), pp. 245-254.

[2] Chassaing, R. (2005). *Digital signal processing and applications with the C6713 and C6416 DSK*. New Jersey: John Wiley & Sons, Inc., Hoboken.

[3] *Code Composer Studio IDE v3 White Papers*. (2004). Texas Instruments, Dallas.

[4] Ćwikliński, Ł., Kiciński, W. (2009). Computer-aided designing of signal processing systems on digital signal processors. *Biuletyn WAT*, (2), 217-227. (in Polish).

[5] Ćwikliński, Ł. (2009). Influence of parameters of Malvar wavelet on boundary effects of multiresolusion analysis. *Electrical Review*, (2), 41-44. (in Polish).

[6] Kiciński, W. (2010). Malvar wavelet in transient detection. *European Conference on Underwater Acoustic*. Istanbul, Turkey, 1280-1286.

[7] Mallat, S. (1999). *A wavelet tour of signal processing*. Academic Press.

[8] Ravier, P., Amblard, P.O. (1996). Using Malvar Wavelet for transient detection. In *Proceedings of the IEEE-SP International Symposium*. Paris, 229-232.

[9] Kotarski, M., Smulko, J. (2009). Noise measurement set-ups for fluctuations-enhanced gas sensing. *Metrol. Meas. Syst.*, 16(3), 457-464.

[10] TMS320 DSP/BIOS User's Guide. (2004). Texas Instruments, Dallas.

[11] TMS320C6713B Floating-Point Digital Signal Processor Data Sheet. (2006). Texas Instruments, Dallas.

[12] TMS320C6713 DSK Technical Reference. (2003).Texas Instruments, Dallas.

[13] TMS320C6000 Assembly Language Tools, v 6.1. (2008). Texas Instruments, Dallas.