

Development and Application of Remote Laboratory for Embedded Systems Design

Anzhelika Parkhomenko, Olga Gladkova, Eugene Ivanov, Aleksandr Sokolyanskii, Sergey Kurson

Zaporizhzhya National Technical University, Software Tools Department, Zaporizhzhya, Ukraine

Abstract—This paper devoted to Embedded Systems' Hardware-Software CoDesign and an overview of approach based on using ready platforms. Comparable analysis of well-known platforms from different vendors is given. Application of remote laboratories of the embedded system rapid prototyping was offered. Remote lab architecture, set of experiments and use case scenario are described. Proposed approach improves the current state of the art in the area of embedded systems design allowing to accelerate stages of hardware-software integration and testing.

Index Terms—Embedded systems' design; Hardware and software platform; Remote laboratory; Web-interface; Experiment set; Program code; Control system of the mobile objects; Support design decision.

I. INTRODUCTION

Today Remote Laboratories are successfully developed and implemented worldwide. According to observations the vast majority of already implemented experiments and installations with remote access are used as pure educational resources and that the possibilities for the professional researcher or designer are not fully explored. On the other hand the Embedded Systems market is permanently evolving and requires the creation of more complex systems in shortening periods of time. Under these conditions, the development of systems "from scratch" is not effective. Therefore, one of the key areas to improve the efficiency of Embedded Systems design, is the accumulation of technical solutions for reuse. This (reuse) of knowledge can be illustrated and implemented in remote laboratories. The disunity of descriptions of already developed components (hardware units, programs, algorithms, implementations, etc.) hinders their reuse, but the application of remote experiments will allow the designer to obtain information about ready hardware/software platforms and components for making decisions on the Embedded Systems realization.

II. INVESTIGATION OF THE EMBEDDED SYSTEM DESIGN FEATURES

At present experts represent and analyze various height-level design methodologies for the embedded systems. Design flow of embedded system presented in various publications, e.g. in [1-4]. The main directions shown in [1]:

- Object-oriented design.
- Hardware/software CoDesign.
- Platform-based design.
- Actor-oriented design.

- Mixed-language design.
- Aspect design.

Hardware-Software CoDesign is considered as one of perspective approaches in design of the ESs. CoDesign techniques has both advantages and disadvantages. On the one hand, complication of the embedded systems' design, and on the other hand significant improvement of the final product characteristics in comparison with alternative versions of design solution. Generally the embedded systems design flow looks like in Fig. 1.

The established practices of embedded systems design show that hardware decisions are initially selected. Further, based on it the software superstructure will be made. This also has to great risk of delaying the whole product design chain [2]. After the hardware and software components have been implemented, they must be separately verified, integrated, and whole system tested again.

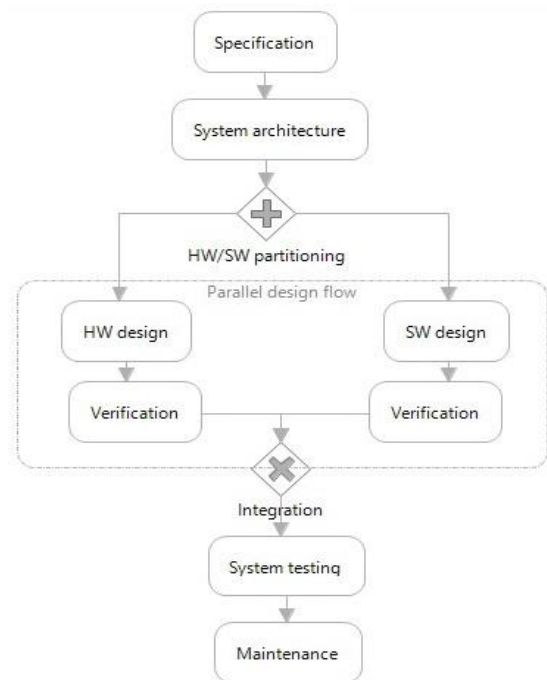


Figure 1. Classical design flow of embedded systems

Since the problem of reduction of the design time remains relevant, this approach of embedded system design has not been successful and requires improvements. Today, in the field of embedded systems (ES) design is very popular use of ready hardware and

software platforms allowing to accelerate development of products so to reduce the time to market.

Researches had shown there are many different manufacturers of hardware and software platforms: Arduino, Texas Instruments, Parallax Inc., Netmedia, Microchip, Digilent, MBED, Raspberry PI, Cyclone. Each platform has the form factor and functionality, and designer's choice depends on the task. Thus, the designer of ES has to know possibilities of the ready platforms existing in the market and be able to make the responsible decision concerning application of a certain hardware-software platform. Unfortunately, the information which is offered on the website not always allows to make the right reasonable choice and optimally implement the project. Acquisition of a huge number of various platforms by small and medium-sized enterprises is unacceptable for the analysis of their opportunities, despite the relatively low prices [5].

Due to growth of popularity of ready hardware-software platforms, a number of the companies (e.g., Autodesk) offer applications for virtual simulation of their work. For example, it is possible to allocate such simulators for Arduino as: 123D Circuits, Virtualbreadbord, Simulino, Virtronics, Simuino, Arduino Simulator and others. However, the simulation doesn't replace real work with the hardware and software. In fact, instead of real physical process the simulator allows to study only its mathematical model. Many software simulators are not free, requires time for their studying, have limited functionality and incomplete element base. There are also powerful packages of circuitry design, such as: ISIS PROTEUS, Altium Designer and others. However, the use of their full functionality isn't always required, and the cost is quite high.

Therefore, creation of specialized laboratory for research of ready hardware-software platforms based on remote experiment is an actual task.

Application of remote laboratories will make changes to the embedded systems design flow (Fig. 2).

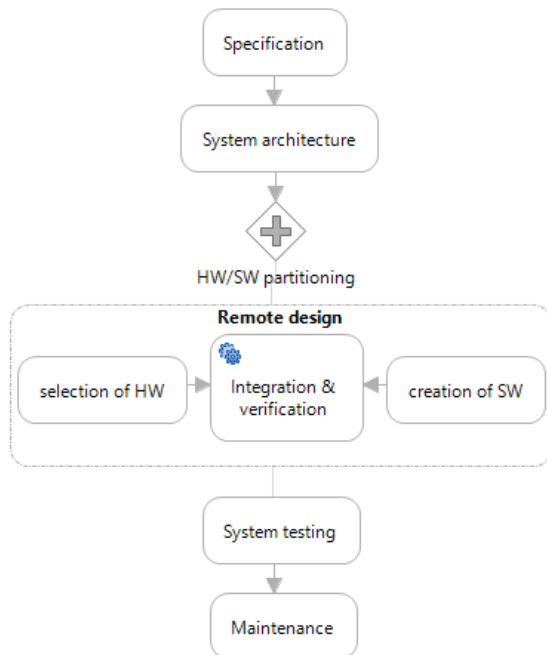


Figure 2. Modified design flow

Introduction in the design process of remote lab will allow users to:

- Extract and analyze information about specifications of ready hardware and software components for their reuse.
- Perform selection of hardware platforms.
- Develop a program code.
- Test compatibility and working capacity of hardware and software parts of the projectable system.
- Check of the devices functions to the specification requirements.
- Observe the experiment performance.

Remote testing of components allows quick assess the feasibility of their application and decision on their integration into the project, thereby reducing the risks and the time expenses for design.

III. DEVELOPMENT AND APPLICATION OF REMOTE LABORATORY FOR THE SUPPORT OF ADOPTION OF DESIGN DECISIONS

The specialized remote laboratory RELED (REmote Laboratory for Embedded Systems Design) is focused on the providing help to developers in a fast and effective choice of standard design decisions on the basis of ready hardware-software platforms.

Fig. 3 gives an overview about RELED infrastructure. The laboratory provides remote controlled access to Arduino based experimental equipment. Arduino is the powerful microcontroller with large volume of memory that allows to create complex electronic devices. Besides, the hardware-computing Arduino platform is one of the most popular and simple in studying.

For comparison the following well-known boards were chosen: Arduino Uno, LaunchPad, PCduino, Raspberry PI, BeagleBone Black. There are already many articles comparing various development boards [6-8]. The comparative Table 1 is given below.

TABLE I.
WELL-KNOWN PLATFORMS' COMPARATIVE TABLE

	Microcontroller boards		Computer boards		
	Arduino Uno	Launchpad MSP430	PCduino 3	Raspberry PI	Beagle Bone Black
Price	20\$	10\$	59\$	35\$	45\$
MC/CPU	16MHz, ATmega 328	16MHz, MSP430 G2553	1GHz ARM Cortex A8	700 MHz ARM 1176JZFS	1GHz TI Sitara AM3359 ARM Cortex A8
RAM	2KB	512B	1GB	512MB, SDRAM	512MB, DDR3
Pins	14, 6	20, 3	32	2x13, 8	2x46
Overall size (mm)	68.6x53.4	66x51	121x65	86x54	86.4x53.3

It is clear that Arduino and LaunchPad are in a different league than the PCduino, Raspberry PI, Beagle Bone Black. The Arduino and LaunchPad are the

microcontrollers. A microcontroller is just one tiny part of a computer. The Arduino can be programmed in C, but can't run an operating system. Arduino and LaunchPad are just perfect for electronics projects and prototyping. They allow rapid, cheap, prototyping of embedded systems.

On the other hand, the Raspberry Pi and PCduino are computers. They need an operating system to work. Those devices can run the operating system alone [6]. The major differences between Arduino and LaunchPad are: cost and memory. A lot of the power of the Arduino is in its community code libraries. Apart from the LaunchPad shields are practically absent.

So, in comparison with other similar platforms, Arduino has a number of advantages:

- The project was developed and develops as the project with an open code which works as the network project/community, allowing participants to exchange experience and ready applied practices, further accelerating the process of development and debugging.
- The low cost of the microcontroller and expansions for it.
- The simplicity and cross-platform of the programming environment (OS Windows, Macintosh OSX and Linux: 32/64bit).

The stand with experiments is connected to the laboratory server via the serial interface. The computer with Linux Debian OS acts as the server of remote

laboratory. The server provides access to experiment programming and video stream of the laboratory. Video stream is implemented with use of "ffmpeg". The server processes web client requests and carries out the following actions depending on the obtained data:

- Compilation of received initial code (Arduino, with use of the console utility Ino).
- Returning the compilation results (using HTTP protocol, as well as the sends request for compilation).
- Uploading binaries to the microcontroller (provided that the controller is free).
- Client queuing (if experiment is occupied).

The RELDES process diagram is shown in Fig. 4. To get access to the experiments user needs to be registered. Access to the experiments page is closed for unregistered users. Users can upload their program code to a microcontroller on the Arduino board via the Web-client. With use of the Web-interface the user is able to create own code or upload file with initial code. The Lab server compiles received initial code. Being compiled, the generated code is uploaded to the microcontroller on the Arduino board. If the program has errors the server generates and returns the compilation results. Therefore, the user has the possibility to create and edit their program code directly in the client's Web interface without any need in development tools on the user's PC.

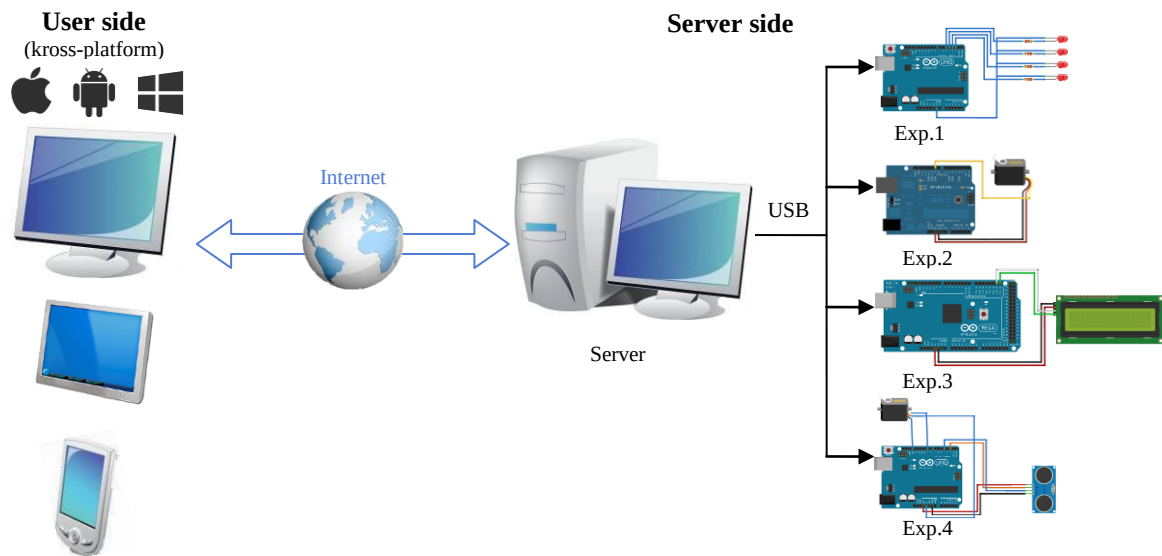


Figure 3. Overview of the RELDES infrastructure

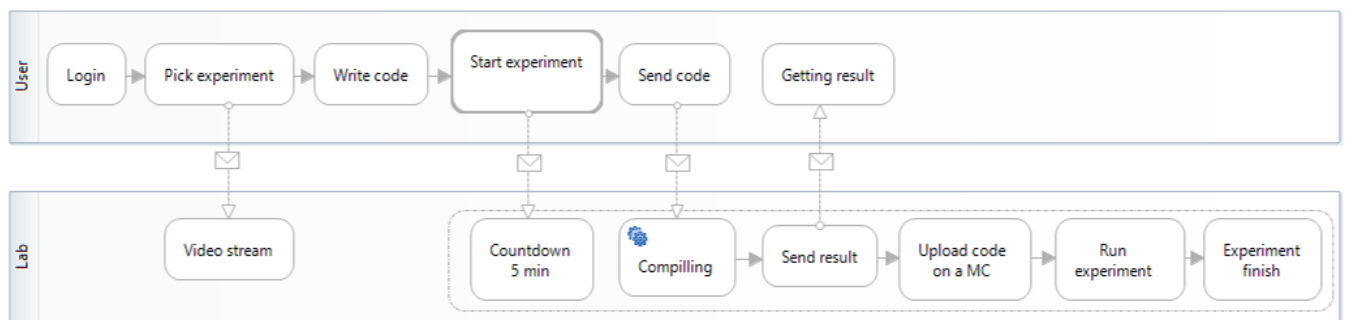


Figure 4. Laboratory process diagram

Client-server communication is carried out via web-interface implemented with HTML+CSS+JS. The service is running on the Apache server with use of the relational MySQL database. Fig. 5 illustrates an impression of the experiment page Web-interface.

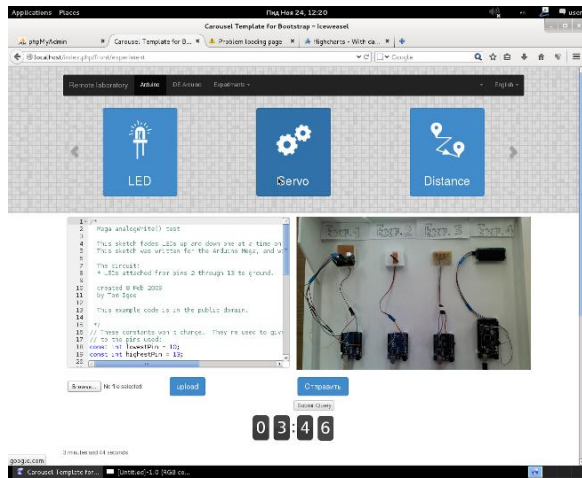


Figure 5. RELDES web-interface

The user friendly slider for fast switching between experiments is located at the top of the page. The remaining area is divided for the displaying video and code editing. At start of work with remote experiment user can observe time remained to the end of experiment.

Besides web browser no additional software is required to access the remote laboratory

IV. THE LIST OF REMOTE EXPERIMENTS

The set of experiments depends on specifics of the designed ES. For example, as standard decisions for design of mobile objects control system [9,10], experiments based on the servos, the liquid crystal display, the LEDs, the distance sensor, etc. are offered. The proposed set of experiments can be expanded for other hardware-software platforms, and tasks connected with realization of various classes of ES.

Experiment 1. Traffic Lights. This experiment allows users to control set of LED with Arduino board. LEDs are connected to digital pins of the board. Resistors are used to restrict the current flowing through the diodes. Diodes have common ground. (Fig.6). Scenarios of experiment: Sequential lights, Running lights, Brightness change.

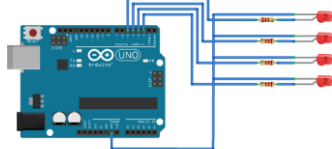


Figure 6. Model of the experiment Traffic Lights

The scenario of experiment allows to set and control such parameters of the traffic light as: pin number and pin mode. Function `pinMode` is used for that. Also, to control diodes' brightness `analogWrite` call is used. Its arguments are pin number and voltage value. Voltage value is not used directly as parameter of `analogWrite`, value ranged from 0 to 255 is used instead. To make it possible for user to see diodes brightness change function `delay` is called. It

stops board's work for amount of time (in ms) stated as argument of the function.

Experiment 2. Servomotor. In this experiment user is allowed to control servo. It is controlled through digital pin. Servo got 3 pins: vcc, ground and control pin (Fig.7). Scenarios of experiment: Two position turns, Smooth turns.

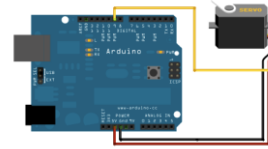


Figure 7. Model of the experiment Servomotor

Program is using `Servo.h` library, which contains `Servo` class implementation. It is used to communicate with motor. To set position of the servo method `write` is used. Angle is passed to the method as parameter. Function `delay` is halt board's work for desirable time in milliseconds. It is necessary to give the servo time to perform turn before next command is given. Not all servos got turning range of 0-180 degrees. It may vary depending on model.

Experiment 3. LCD display. In this experiment user is allowed to control liquid crystal display. Display has integrated controller and not accessed by Arduino directly. Communication is done via I2C serial protocol (Fig.8). Scenarios of experiment: Static text, Scrolling text.

Firstly `LiquidCrystal_I2C.h` and `Wire.h` libraries are included. `Wire.h` implements I2C protocol and `LiquidCrystal_I2C.h` implements communication with LCD via I2C. `LiquidCrystal_I2C` class object is created on the beginning of the code with parameters specific to given LCD controller. In this case in the function `LCD` parameters are set in `begin` call. Its parameters are numbers of rows and columns. `backlight()/noBacklight()` function is called to enable/disable LCD's back light.

Cursor position may is set by calling `setCursor` (column, row) function. `clear()` function is called to remove all the symbols from LCD. `delay(milliseconds)` function is called to delay activity of the board for stated time in milliseconds. Function `print(string)` is called to print symbols starting from cursor position.

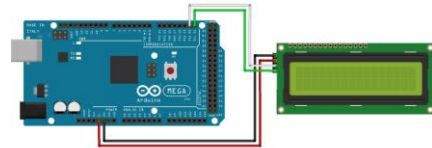


Figure 8. Model of the experiment LCD display

In this case it is done setting the parameters of the LCD screen with a call to `begin` c parameters, means the number of columns and rows of characters. Next is called `f-tion print`, which would display a specified set of characters. The second function of loop is executed in an infinite loop while the rest of the board. In this case it is performed setting the cursor to the beginning of the second line (numbered from zero produced) by calling `setCursor` and printing time of the board in seconds.

Experiment 4. Sonic distance sensor. In this experiment user works with sonic distance sensor. Sensor has 4 pins:

vcc, ground, trigger and signal. (Fig.9). Scenarios of experiment: Distance measurement, Distance sensor with servo.

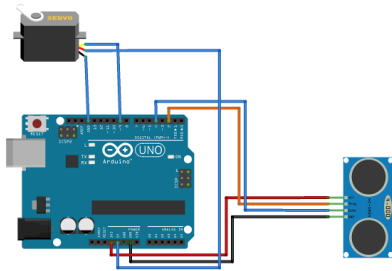


Figure 9. Model of the experiment Ultrasonic sensor distance

User receives data from sensor through the serial interface. Serial class is used for that. In this function serial interface is initialized with baud rate of 9600 and sensor pins modes are set with pinMode call. Variables with sensor pins are declared before setup function. Distance sensor has two modes: measuring and sending data. To set it in measuring mode pulse with 2 microseconds length must be sent to the sensor (trig pin). It is done by calling digitalWrite function. Its parameters are pin number and pin value. To form the pulse function called multiple times with LOW and HIGH values. Delays between calls are made with delayMicroseconds function. To read data from the sensor pulseIn is called. This function's arguments are pin number and expected value (HIGH or LOW). Returned value will represent travel time of sonic signal measured by the sensor. To convert that value to distance functions microsecondsToCentimeters and microsecondsToInches are used. Results are returned to user by print method of serial interface.

In case of second scenario, sensor is attached to servo. It allows user to turn the sensor manually and measure distance to different objects. Servo position is set by write method of Servo object. Angle values are stored in the array. After servo position is set command to measure distance is called. To do so logical 1 pulse with duration of 10 microseconds is sent to the sensor with digitalWrite call. To read the measurements pulseIn is called. Results are sent to the user via serial interface by calling print method.

CONCLUSIONS

Development of remote laboratory for complex hardware-software design of the embedded systems is an actual task, because the problems existing in this area demand qualitatively new techniques, technologies and tools of ES design.

Benefits of our laboratory:

- Cross-platform (use any device and operation system).
- Accessibility (use our hardware instead of spending money).
- Configurability (write your own code or use code templates).

Practical application of the developed laboratory will allow the designer to make the right decision choice on expediency of application of ready platform, and possible options of the project development.

Future work will be focused on expanding the range of provided hardware-software platforms and set of experiments for solution of tasks from different areas of embedded systems design.

REFERENCES

- [1] Platunov, A.E., Postnikov, N.P., "High-level design of embedded systems", p.121, SPb.: NIU ITMO, 2011.
- [2] Teich, J., "Hardware/Software Codesign: The Past, the Present, and Predicting the Future", Proceedings of the IEEE, Vol.100, pp. 1411-1429, Germany, May 13, 2012.
- [3] Abdurrohman, M., Kuspriyanto, Sutikno, S., Sasongko, A., "The New Embedded System Design Methodology For Improving Design Process Performance", International Journal of Computer Science and Information Security, Vol. 8, No. 1, pp. 35-43, 2010.
- [4] Wolf H., Wayne, "Hardware-Software Co-Design of Embedded Systems", Proceedings of the IEEE, Vol.82, NO. 7, pp. 967-989, Germany, July 1994.
- [5] Parkhomenko, A., Gladkova, O., "Interactive remote laboratory for research of hardware-software platforms", Proceedings of the International Scientific-Practical Conference «Internet-Education-Science-2014», pp. 111-113, Vinnytsia: VNTU, October 14 - 17, 2014.
- [6] Santos, R., Arduino vs Raspberry PI vs Beagle Bone Black – 2013. [Electronic source] - <http://randomnerdtutorials.com/arduino-vs-raspberry-pi-vs-beaglebone-vs-pcduino/>
- [7] Allan, A., Which Board is Right for Me? – 2014. [Electronic source] <http://makezine.com/magazine/make-36-boards/which-board-is-right-for-me/>
- [8] Leonard, M. How to Choose the Right Platform: Raspberry Pi or BeagleBone Black? – 2014. . [Electronic source] - <http://michaelhleonard.com/raspberry-pi-or-beaglebone-black/>
- [9] Parkhomenko, A.V., Gladkova, O.N., "Virtual Tools and Collaborative Working Environment in Embedded System Design", Proceedings of XI International Conference on Remote Engineering and Virtual Instrumentation (REV2014), pp. 91-93, Porto: Polytechnic, Portugal, February 26-28, 2014.
- [10] Tsmots, I., Teslyuk, V., Vavruk, I., "Hardware and software tools for motion control of mobile robotic system", Proceedings of XII International Conference «The Experience of Designing and application of CAD Systems in Microelectronics (CADSM 2013)», p. 368, Lviv Polytechnic, Lviv, 2013.

AUTHORS

Anzhelika Parkhomenko is with the Zaporizhzhya National Technical University, Software Tools Department, 69063 Zaporizhzhya, Ukraine, Zhukovskogo 64, (e-mail: parhom@zntu.edu.ua).

Olga Gladkova is with the Zaporizhzhya National Technical University, Software Tools Department, 69063 Zaporizhzhya, Ukraine, Zhukovskogo 64, (e-mail: gladkovaolga9@mail.ru).

Eugene Ivanov is with the Zaporizhzhya National Technical University, Software Tools Department, 69063 Zaporizhzhya, Ukraine, Zhukovskogo 64, (e-mail: skycat4best@gmail.com).

Aleksandr Sokolyanskii is with the Zaporizhzhya National Technical University, Software Tools Department, 69063 Zaporizhzhya, Ukraine, Zhukovskogo 64, (e-mail: sanya_sokol1992@mail.ru).

Sergey Kurson is with the Zaporizhzhya National Technical University, Software Tools Department, 69063 Zaporizhzhya, Ukraine, Zhukovskogo 64, (e-mail: kurson-serg@yandex.ru).