

# Healthcare Multimedia Application for Multi-modal Mobile Device Interaction

Penhaker Marek, Kijonka Jan

Department of Cybernetics and Biomedical Engineering  
VSB – Technical University of Ostrava, FEI, K450  
Ostrava, Czech Republic  
marek.penhaker@vsb.cz jan.kijonka@vsb.cz

**Abstract**—This study presents the healthcare multimedia application for multimodal device interaction based on mobile cell phone styles. The application provides visualization and user interaction of health care status information. The data are processed from a user interface and from sensors embedded in mobile devices and surrounding personal health care devices. The multimedia application is ready to use for personal health care and also for elderly people self monitoring.

**Keywords**-mobile; multimedia; application; interaction

## I. INTRODUCTION

A human can be monitored by user adaptive visualization system in home ambient. The information of measured data could be provided to the patient by several ways like visualization on a TV screen. The user would select the kind of vital function data to visualize on the TV screen by wireless driver. The disadvantages are worse portability of TV, necessity of creating a special data channel for connecting on TV video input, special wireless TV driver designing.

The other visualization possibility is on a PC monitor where the user would examine the measured data with a desktop computer or on a notebook screen. The disadvantages are the required computer and the cumbersome PC. The advantages are large screen area and simple handling of PC application by mouse and keyboard. Data would be transmitted by WiFi.

Visualization on PDA is the advantages are compactness and device mobility. It would be necessary to make an application for PDA. The data would be transmitted by WiFi or BlueTooth. Disadvantage is battery consumption and handling the smart phone by elderly people. [9]

Visualization by use of a mobile phone is the new measured values would be transmitted by SMS. The disadvantage is the reduced visualization option.

The best one way is developing a special mobile device (mobile unit for data measuring and visualization) with a user interface (LCD and keyboard) which is suitable to the visualization requirements. The device allows wireless transmission of vital functions data. The advantage is great stability in comparison to PC or PDA as the application doesn't work on any operation system. The device is useful for emergency calls (emergency medical service calling), for example after the given keyboard button push. In the

following chapters we describe the mobile unit and user communication interface. [10]

## II. MOBILE UNIT

The mobile unit is a device, which can monitor human basic vital functions and provide communication with surroundings by wireless nets.

The mobile unit integrates all the measuring and visualization functions to one compact mobile device, which is capable of ZigBee wireless communication. The device is embedded to a case of mobile phone size. The LCD, keyboard and Power button are used to create the user interface. The output connector is used for measuring and programming wires connection and for the battery charger connection. The device is powered by Li-Ion 3.3V battery. All components of the device are designed for this voltage.

### A. Block Diagram

The block diagram of the mobile unit is in Fig. 1. There is symbolized its inner structure, function of individual blocks and communication between them. An individual description is given to the blocks of the user communication interface.

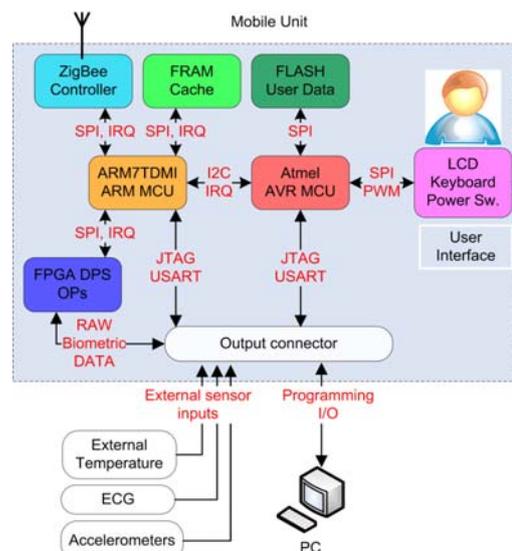


Figure 1. The mobile unit block diagram

**B. Mobile Unit Control**

The control microcontroller of the mobile unit is an MCU based on the ARM architecture. It transmits all the visualization data to the user interface control unit MCU based on the AVR architecture. The user interface control unit processes the received data.

The communication between ARM and AVR MCUs proceeds through the I2C serial interface. The I2C protocol is described in chapter I2C Communication Protocol. The components of the communication protocol include also an interrupted line allowing for the duplex communication.

**C. User Interface Control**

The ATmega644P is a low power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. It contains enough capacity of Flash program memory (64K Bytes). The clock frequency is set to 12MHz, maximal for 3.3V MCU operating voltage. The power consumption is 7mA in active mode and 0.2µA in power-down mode (used during the device down). Used peripherals are PWM (LED Backlight intensity adjustment), timers/counters, I2C, USART, JTG and SPI interfaces. [1]

The AVR MCU allows for communication with other components of the user communication interface (LCD, keyboard, external Flash memory) and external devices (MCU ARM, PC). The AVR MCU contains the main program in Flash program memory. The program allows for receiving vital function data,, storing and reading from an external data Flash memory, communication with the LCD driver, keyboard keys status reading and communication with PC (MCU programming, data Flash uploading).

**D. User Data Flash**

The AT45DB081D with serial SPI interface is used as the external Flash memory. Its 8Mb of memory are organized as 4096 pages of 256 or 264 bytes each. It allows intelligent memory operations, page programming and flexible erase options (page, block, sector, and chip erase). 100 000 program/erase cycles per page are guaranteed. [3]

Content of external user data Flash:

- Video data (icons, user menu panels, battery and signal strength states, the map of monitoring area) is stored in RGB format suitable for LCD writing, or compressed (RLE, JPEG).
- Fonts (3 font styles 6x8, 8x8 and 8x16 pixels)
- Trend data (temperature, pressure, oxygen saturation, weight, and position)
- Settings (back light, curves' speed rate, actual indexes of trends data arrays)

**E. LCD**

The LCD has relatively the simple 9-bit serial interface, running on 6.6MHz maximum frequency. It uses an Epson S1D15G10 or a Phillips PCF8833 controller driver, which allows driving and data imaging on LCD. LCD driving with the Phillips PCF8833 controller and driver is described in chapter LCD Driving. LCD has powerful white LED backlight - 7V @ 40-50mA (very bright). [5]

It's necessary to use the DC-DC step-up converter for LED backlight power supply. The used step-up converter MC34063 is a switching regulator. Back light intensity is regulated by PWM signal generating. [7]

**F. Keyboard**

It is used several push buttons of matrix keyboard for user menu controlling. The user menu is described in chapter User Menu.

**G. Output Connector Programming I/O**

MCU AVR and external Flash memory programming is possible by the USART and JTAG interfaces. USART and JTAG are connected to the output mobile unit connector.

There was the MATLAB GUI used to create an application, which enables uploading the external user data Flash. This application is described in the chapter Interactive User Data Flash Uploading. The mobile unit is connected with PC through the RS232 serial interface (COM port PC). It is possible to use a chip based on MAX232 for RS232 to USART conversion.

The AVR JTAGICE mkII programmer serves for programming the AVR MCU. It supports all types of AVR MCUs. For programming and debugging the AVR MCU we can use the AVR Studio development platform.

**III. ADAPTABILITY USER INTERFACE**

The mobile unit for measuring and measured data visualization is a compact device, which is able to be configured by the user to present well-arranged results. The visualization system consists of several construct parts described below.

**A. User Interface Parts**

- LCD
- Matrix keyboard
- Power button
- User interface control unit and external user data Flash
- Output connector programming I/O

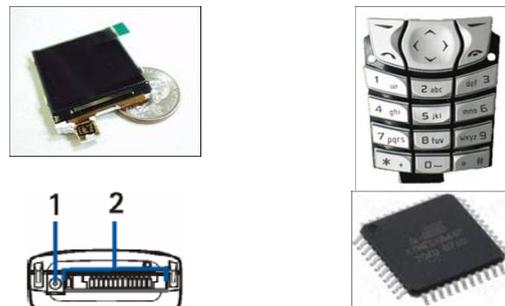


Figure 2. User interface parts a) LCD b) matrix keyboard c) output connector d) MCU and Flash

**B. User Interface LCD Imaging**

For the user of a mobile unit, the user menu displayed on LCD was created.. Moving and options in the user menu are

achieved by several keys of the keyboard. The user has the possibility to display vital functions and additional data. He can also execute an emergency call, upload user data Flash, or launch the demonstration mode of visualization. Additional options are settings of the parameters as waveform sweep speed, LCD back light intensity.

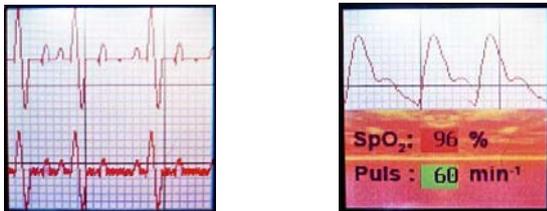


Figure 3. Actual ECG and pletysmography record demonstration on LCD  
a) ECG channels 1 and 2 b) pletysmography

Actual waveform measurement imaging ECG (channels 1 and 2) and pletysmography (including SpO2 and pulse values). SpO2 values are stored in the user data Flash memory to examine it in the saturation trend.

Trends imaging – the body and ambient temperatures, systolic and diastolic pressure, a pulse, SpO2, weight, the position (the room ID). It is possible to set the common or separate scales for the body temperature and ambient trend, on/off line with particular value displaying, on/off minimum and maximum value of graph displaying.

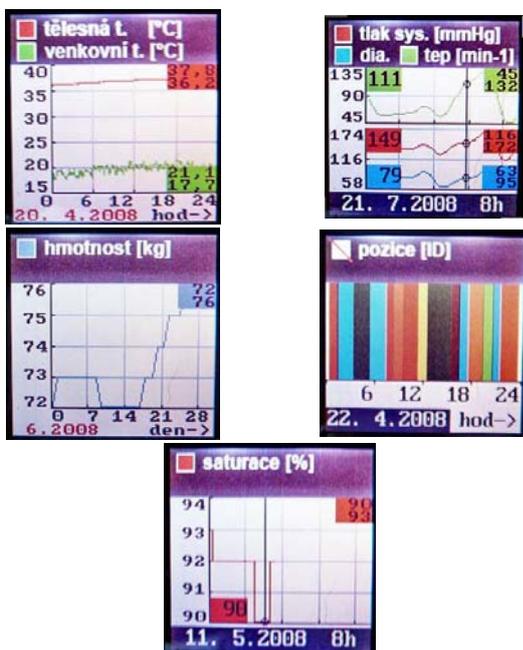


Figure 4. Trends imaging on LCD screen demonstration a) body and ambient temperature curves b) pulse, systolic and diastolic pressure curves c) saturation curve d) body weight curve e) position

There is also other information as actual position displaying, emergency calling, event displaying, DEMO mode, user data Flash uploading, settings - waveform sweep speed, LCD back light intensity.

#### IV. LCD DRIVING

The used LCD type is driven by Phillips PCF8833 controller driver. The PCF8833 is a single chip low power CMOS LCD controller driver, designed to drive color Super-Twisted Nematic (STN) displays of 132 rows and 132 RGB columns. All necessary functions for the display are provided in a single chip, including RAM which has a capacity of 209 kbit (132 x 12-bit x 132). The PCF8833 offers the 3-line serial interface.

The PCF8833 has 2 access types; the one defining the operating mode of the device (instruction) and the other filling the display RAM (data). Efficient data transfer is achieved by auto-incrementing the RAM address pointers.

##### A. Write mode – the serial interface

SPI 9-bit communication. Each data packet contains a control bit D/C and a transmission byte (instruction/data). The logic value of control bit D/C interprets the following byte as instruction (D/C is logic 0) or data (D/C is logic 1).

##### B. Instructions

There are 3 types of instructions; the one defining the display configuration (data format, color inversion, partial mode, rolling scroll mode, etc.), the one setting X and Y addresses, and miscellaneous data.

Different display data formats are available because different color depths are supported by PCF8833. The color depths supported are: 4 Kbyte colors (12-bit/pixel), RGB 4 : 4 : 4 bits input (4 bits for red, 4 bits for green and 4 bits for blue color resolution). The data coming from the interface is directly stored in RAM, or with better color depth, for maximal use affording a realistic imaging, as flat scene imaging it supports 65 Kbyte colors (16-bit/pixel), RGB 5 : 6 : 5 bits input. The 16-bit data coming from the interface is mapped by means of dithering to 12-bit data. Then, the dithered 12-bit data is stored in the RAM.

#### V. I2C COMMUNICATION PROTOCOL

For communication between ARM and AVR MCUs we define the I2C communication protocol.

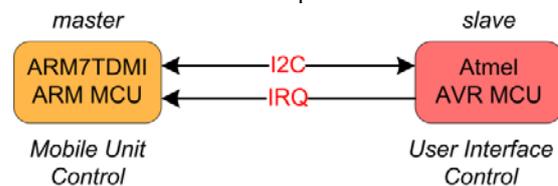


Figure 5. I2C communication protocol block diagram

Mobile Unit Control MCU – it transmits a continual data stream, which contains a vital measured data function, actual time, and actual events' data.

There is defined sampling frequency for ECG, pletysmography, temperature and time; a single-shot value for weight, pressure, position, and events' data.

User Interface Control MCU – it receives a continual data stream; transmits the events' data (emergency calling, power down request). Transmitter mode is activated by the interrupted line.

Data stream – the communication protocol consists of 1 byte opcode, which interprets following data type, or it defines some event in itself. The opcode is followed by several data bytes or by next opcode.

VI. PROGRAMMING AND THE APPLICATION DEVELOPMENT

For programming the microcontroller ATmega644P, we can use the AVR STK500 Flash Microcontroller Starter Kit. The AVR STK500 presents a complete tool kit for the Atmel AVR microcontrollers. It allows development, debugging and testing prototypes applications. [8]

The application was programmed in AVR Studio 4 with AVR GCC programming language. The compiler is included in AVR Studio.

A. Libraries and Functions

The program is built-up of considerable functions' libraries, which are necessary to communication with LCD controller driver, primitive graphic objects, text and numeric values rendering, image data rendering (images, flat maps of monitored area), vital functions data receiving and the user interface. The program configures and uses microcontroller peripherals as I/O, SPI, USART, I2C, TC0, TC2, etc.

The video data are stored in the user Flash memory. We access the data by memory page address listed in the fat.h library.

The draw.h library contains the functions for graphic objects on LCD imaging (pixel, line, rectangle, circle, text, and number).

The functions for Flash memory handling, reading image data and displaying images on LCD are in the flash\_lcd.h library.

The functions for time handling, trends, ECG and pletysmography waves imaging and user interface options are stored in particulars libraries time.h, trends.h, waves.h, TIMER0\_okna.h.

In Figure 6, there is an image stored in the Flash memory with illustrated coordinates (x0, y0) and (x1, y1), which are stored in initialization bytes of data file. Also, this illustrates the width and height of an image in pixels. The part of the image actually displayed on the LCD screen is marked in the red frame with the width x1\_LCD0 - x0\_LCD0 pixels and the height y1\_LCD0 - y0\_LCD0 pixels. The parameters x0\_LCD0, x1\_LCD0, y0\_LCD0 and y1\_LCD0 determine the part of the LCD display screen used for image displaying.

The appropriate part of the image is displayed on LCD according to the received position coordinate (x\_poizce, y\_poizce). (x\_poizce, y\_poizce) is actual position of patient in monitored area. x\_poizce has range <x0, x1>, and y\_poizce has range <y0, y1>. By mapa\_bod argument selection we can on/off displaying of the point of position coordinate. [10]

The appropriate function for image data LCD displaying is executed by the compression type of the stored image data.

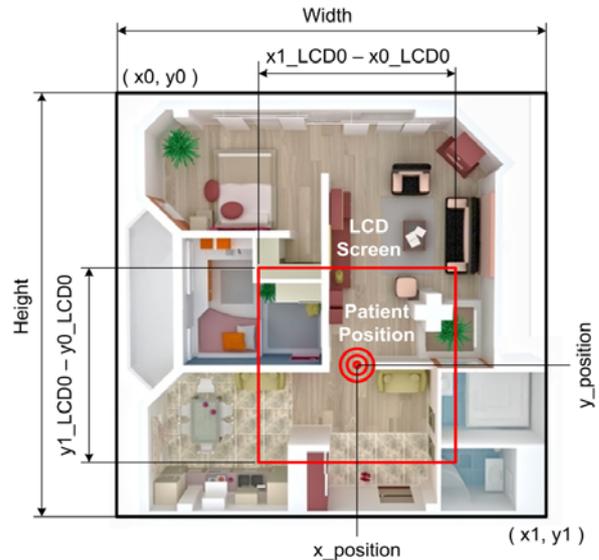


Figure 6. Map of monitored area and image parameters stored in flash memory

VII. PROGRAM FUNCTION DESCRIPTION

The program starts to execute in the int main (void) function, just through the AVR MCU connected to the battery voltage. In the int main (void) function, there are executed the following tasks: the Watchdog reset, the power push button configuration as interruptible input, enable all interrupts, and the power consumption reduction (the ADC shutdown). Finally, the AVR MCU is switched to the power-down mode for the minimal power consumption. From this power mode, the AVR can wake-up only by external interrupt (the power push button). At the end of the int main (void) function, there is the infinite loop placed while (1); to continue the program after MCU wake-up.

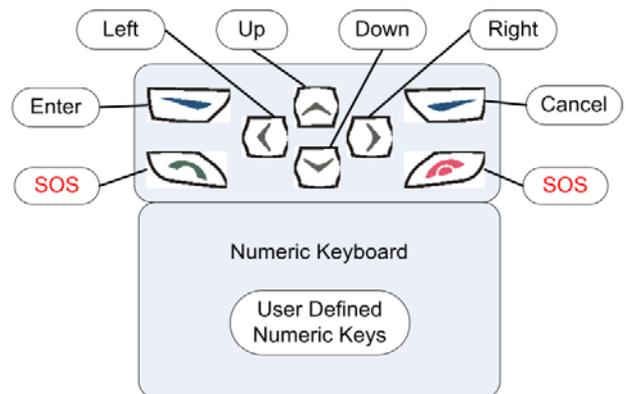


Figure 7. Keyboard use

After the device is switched-on, the void INIT() function is executed. After the device initialization the push-buttons status is periodically read in timer 0 interrupt. There are particular functions executed in the interrupt code of the user

interface menu. The I2C interface is also active for communication between ARM and AVR MCUs.

VIII. USER MENU

The user can move in the user menu displayed on a LCD screen by several push buttons.

A. Main Menu

After switching on the mobile device, a logo is displayed on LCD (figure 8).

The ZigBee signal strength status (Figure 9), battery status (Figure 10), and the actual time formatted to HH:MM are displayed in the upper part of the LCD screen. These data are called as "Initiative Line". Below this line this is space for one line message called as "Event Line". It is the last event concerning data receiving: a new measured pressure value begins next and the end of EKG or plethysmography measuring, ALARM – smoke in the room. Below the "Event Line" there is displayed the main menu (Figure 11) namely one of the mentioned option (a, b, c, and d). Moving in the main menu is realized by arrow buttons "Left" and "Right". The selection is approved by the key "Enter".

The similar style of moving in the user menu is in other menus: waves, trends and settings. [13]

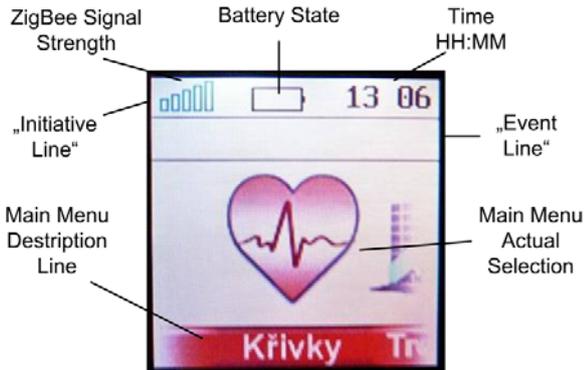


Figure 8. The main menu window



Figure 9. The ZigBee signal strength status



Figure 10. The battery statuses



Figure 11. Main menu

IX. INTERACTIVE USER DATA FLASH UPLOADING

There was MATLAB GUI used to create an application, which enables uploading the external user data Flash. You can see the main window of the application in the Figure 12. We use a serial COM port PC for programming. An image with \*.bmp, \*.ico, \*.jpeg, \*.png, \*.tiff extensions, a data file and others can be stored into the Flash memory.

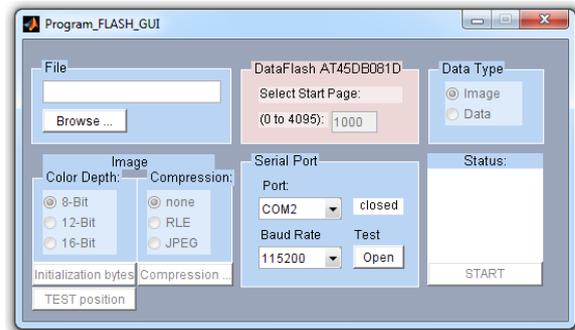


Figure 12. Program\_FLASH\_GUI

Images must be stored into the Flash memory in a format, which is suitable for displaying on the mobile unit LCD screen. The LCD – PCF8833 driver supports 3 data formats (for 8-bit, 12-bit or 16-bit colors) as stated in the chapter LCD Driving. It is also convenient to compress images in order for them to take as least space as possible in the Flash memory with 8Mbit capacity. [12]

The data file of each image contains 24 initializing bytes in its beginning. We cannot change the first 6 bytes as those contain necessary information about the image (the width and height of the picture in pixels, the compression kind and colors quality). It is not possible to identify an image without this information, to use an appropriate decompression function, to set the format and so on. The byte 7 – 10 serve for entering the scale of the image. These bytes must be entered if the image is a map. Other initializing bytes are not used in this application. They can be configured additionally.

X. GUI DESCRIPTION

The main window of the user interface for programming the Flash memory is shown in the Figure 12.

After clicking on the "Browse" button you can choose the required file containing a picture or data, which is to be loaded into the Flash memory.

You can choose the first memory page, which will be used for storing data, in the "DataFlash AT45DB081D" item in the GUI. If you select a page, which already contains data, these will be lost by loading new data. It is possible to load an image in a selected format or data like types of a font or trends. The loaded are saved into the Flash exactly the same way as it is saved in the initial file.

A picture can be compressed into JPEG or RLE (Run Length Encoding), which is a compression with no losses. It is also possible to select a quality of colors. When loading images you can set the initializing bytes after clicking on the dedicated button in the GUI.

The data is sent through a serial COM port. It is necessary to set a free port and a data rate of transition. The data rate of transition is preset to 15200 Bd/s. The "Open" button checks the availability of the selected port. [13]

When all the parameters are chosen, the data transfer from PC to Flash memory can begin by clicking on the "START" button on the right bottom corner of the GUI. The "Status" window displays the data file size [byte], the page end of the memory, and additional information. When the data type "Image" is chosen, the original and compressed image is displayed in new MATLAB figures.

After that runs the time 10s, during that the mobile unit should be connected. The item "Flash" should be selected in the mobile unit's user menu. The data transfer begins after pressing the key "Enter". After the transfer termination, the status message is displayed both on mobile LCD unit and on the PC screen.

## XI. CONCLUSIONS

The objective was to create a multimedia user interface for the mobile unit in the form of applications. The generated application works on previously realized hardware. Programming and STK500 communication ports are connected to COM ports on PCs. AVR  $\mu$ C is programmed in the AVR Studio 4. The communication port is used to program the external FLASH memory of the MATLAB GUI. With socket for external development, the board is managing AVR  $\mu$ C connected to an external flash memory, LCD and keypad Nokia. The external field is also involved DC converter generating DC voltage suitable for backlighting LCD and keypad.

Simulation of receiving and sending vital data functions are performed in two ways. The first method is implemented by running the DEMO mode in the user menu. The LCD displays are fictitious measured data. In this case, however, the function of the communication interface is not verified. The second way is by the communication protocol. For this purpose  $\mu$ C AVR Atmega168, which simulates the communication between AVR and ARM AVR Atmega168 is programmed to broadcast a fictitious measured data and that all options according to the protocol. It also has activated external interruption simulating the ARM interrupt. Thus, the functional and messages such as "Off" and "Request for Assistance".

The work was for me a great benefit in terms of programming the AVR  $\mu$ C. It has been used many AVR function and all communication interfaces that offer AVR. Also, capacity SRAM and programming FLASH memory have been largely used. Due to the scale of the program was necessary to abandon the original type AVR Atmega168 and go to ATmega644P with 64kB flash memory / 4 KB SRAM.

Another direction of development and improving the user interface of the communication could be to the user menu of new features and use the whole keyboard for entering numbers of parameters. To view the trends would be appropriate to move the LCD larger. This can select a more readable font size for most users.

## ACKNOWLEDGMENT

The work and the contributions were supported by the project SV SP 2012/114 "Biomedical engineering systems VIII" and TACR TA01010632 "SCADA system for control and measurement of process in real time". Also supported by project MSM6198910027 Consuming Computer Simulation and Optimization. The paper has been elaborated in the framework of the IT4Innovations Centre of Excellence project, reg. no. CZ.1.05/1.1.00/02.0070 supported by Operational Programme 'Research and Development for Innovations' funded by Structural Funds of the European Union and state budget of the Czech Republic.

## REFERENCES

- [1] ATmega164P/324P/644P [online]. Last revision on the 1st of February 2009 [cit. 2009-04-20].
- [2] [http://atmel.com/dyn/resources/prod\\_documents/doc8011.pdf](http://atmel.com/dyn/resources/prod_documents/doc8011.pdf)
- [3] AT45DB081D [online]. last revision 1st of February 2009 [cit. 2009-04-20].
- [4] [http://www.atmel.com/dyn/resources/prod\\_documents/doc3596.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc3596.pdf)
- [5] Color LCD 128x128 Nokia Knock-Off [online]. last revision 5th of April 2009 [cit. 2009-04-05]. [http://www.sparkfun.com/commerce/product\\_info.php?products\\_id=569](http://www.sparkfun.com/commerce/product_info.php?products_id=569)
- [6] PCF8833 [online]. last revision 20th of April 2009 [cit. 2009-04-20]. [www.nxp.com/acrobat\\_download/datasheets/PCF8833\\_1.pdf](http://www.nxp.com/acrobat_download/datasheets/PCF8833_1.pdf)
- [7] MC34063 [online]. last revision 20th of April 2009 [cit. 2009-04-20]. [http://www.datasheetcatalog.com/datasheets\\_pdf/M/C/3/4/MC34063.shtml](http://www.datasheetcatalog.com/datasheets_pdf/M/C/3/4/MC34063.shtml)
- [8] STK500 User Guide. [online]. last revision 1st of March 2003 [cit. 2009-04-10] [http://www.atmel.com/dyn/resources/prod\\_documents/doc1925.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc1925.pdf)
- [9] Dvorak, J., Havlik, J., Data Synchronization for Independent USB Devices. In 2011 International Conference on Applied Electronics. Plzeň: Západočeská univerzita v Plzni, 2011, p. 111-113. ISBN 978-80-7043-987-6.
- [10] O. Krejcar, D. Janckulik, L. Motalova, and J. Kufel, "Mobile Monitoring Stations and Web Visualization of Biotelemetric System - Guardian II". Proc. EuropeComm 2009. LNICST vol. 16, pp. 284-291. R. Mehmood, et al. (Eds). Springer, Heidelberg (2009).
- [11] Krajcuskova Z., Kukucka M.: Time-Frequency Representation and Unconventional Reliability Growth Model, In TSP 2011 : Proceedings of 34th International Conference on Telecommunications and Signal Processing, August 18th-20th, Budapest, Hungary, 2011, s. 518-520, ISBN 978-1-14577-1409-2
- [12] Kasik, V.: Acceleration of Backtracking Algorithm with FPGA. In 2010 International Conference on Applied Electronics, pp. 149-152, Pilsen, Czech Republic, 2010, ISBN 978-80-7043-865-7, ISSN 1803-7232.
- [13] Kijonka, J., Penhaker, M., Kasik, V., Stankus, M. User Adaptive System for Data Management in Home Care Maintenance Systems, In Proceedings of 3rd Asian Conference on Intelligent Information and Database Systems, Springer, Lecture Notes in Computer Science, 2011, pp. 492-501, 20. - 22. April, 2011 Volume 6592/2011, DOI: 10.1007/978-3-642-20042-7\_50