Recent Innovations in Mechatronics (RIiM) Vol. 3. (2016). No. 1-2.
DOI: 10.17667/riim.2016.1-2/16.

Reconfigurable Hardware Technology Application in Buildings Supervising and Monitoring Systems

Szász Csaba
Department of Electrical Machines and Drives
Technical University of Cluj
Csaba.Szasz@emd.utcluj.ro

Miklós Attila
Department of Electrical Machines and Drives
Technical University of Cluj

Abstract— **Due to the continuously increasing demand for more comfortable residential or commercial buildings in the last decades the researches in the so called "intelligent buildings" topic has emerged as one of the most challenging and high ranked engineering task. Consumers also require a higher level of security, supervision, and control of the buildings according to a large scale of user needs. These expectations face building automation and supervising system developers with a challenging problem, difficult to approach with classical methods or strategies. Therefore, this paper is focused to outline novel facilities and solutions offered by the current level microelectronic technologies in building automation. In the first step of this endeavor the benefits and advantages of the reconfigurable hardware systems is highlighted and outlined. Then a concrete building automation and supervising system implementation in hardware reconfigurable technology is presented and detailed. The main unit of this system has been built upon a Genesys Virtex-5 FPGA-based development board, as a high speed, parallel, and distributed computing reconfigurable unit. The software modules for building supervising and monitoring purposes embed last generation MicroBlaze technology which allows fast and convenient implementation of sophisticated control algorithms. The result of the entire development is a powerful and versatile system representing a well suitable solution for the most sophisticated and demanding customer needs in building supervising and monitoring applications.**

Keywords— building automation, reconfigurable hardware, FPGA processor, supervising and monitoring, parallel and distributed computing;

I. INTRODUCTION

It is well known that with the continuously increasing trend in human standards of life a greater demand for more comfortable, convenient, and safer living places or residences is expressed. This requirement is also facilitated by the rapid advances in microelectronic technologies which strongly impact the human life-style, influencing the people's daily life and habits. However, the increased demand for buildings which are healthy, energy efficient and safer usually is expressed via the term "intelligent building" [1, 2]. According to a general rule definition, an intelligent building or residence embeds various features such as modern lighting technologies, energy efficient heating, ventilation and air conditioning systems, climate control, renewable energy resources utilization, high performance computer networks and information technologies, data acquisition systems and sensors, clever building security and supervising systems, remote control systems, but monitoring systems for the doors, windows, and persons entrance, as well [3, 4].

Of course, all the above mentioned complex systems requires the last generation information and microelectronic technologies, capable of implementing real-time control methods and strategies. This immense challenge pushes researchers and engineers involved in the building automation topic into a difficult situation. At first, it is necessary to determine the most adequate microelectronic technologies for buildings automation systems development and implementation. On the other hand, such digital technology should be integrated into the general control system to support of the entire heating, ventilation, and air-conditioning systems, respectively the supervising and events monitoring networks.

Obviously, in each particular case it is required a detailed knowledge regarding the main architectural features and particularities of the residential or commercial building investigated. This refers to the number of building rooms and their location, spacing's distribution and size, number of floors, annex buildings, existing basements or other architectural elements. Toward, it is important to map and count the number of doors and windows and their arrangement, the placement of lighting elements, location and number of heating elements, fan coils, or air-conditioning systems. By possessing the above information it is possible to overview the potential events that can be happen or should be monitored, or the processes amount that must be monitored. However, it is no doubt that all these should be balanced with the real-time implementation requirement completed by parallel and distributed execution of a high number of control, monitoring, supervising, or network communication task, as function of the building's structure and internal functionality. Hence, the above theoretical remarks lead to the main idea that perhaps it would be more efficient and practical to use a general-purpose hardware system architecture that possess the ability to "learn and adapt to" each building's specific mechatronics system structure or automation level implementation requirement. In other words, to use a well fitted hardware framework being able to fulfill a wide range of building automation applications in a large scale of different type residential or commercial buildings.

II. RECONFIGURABLE HARDWARE TECHNOLOGY IN BUILDINGS SUPERVISING AND MONITORING SYSTEMS

Scientists and researchers involved in microelectronics generally agree that one of the most challenging paradigms of the current level digital technology is linked with the

reconfigurable hardware technology. Reconfigurable computing combines some of the flexibility of software with the high performance hardware, by possessing the ability to rapidly change the functionality of their internal structure and wiring configuration between its modules after customer needs. Such structures also have the potential to exploit coarse-grained functional parallelism as well fine-grained instruction level parallelism using custom computing technologies. Additionally, reconfigurable hardware technology offers the huge advantages of speed, flexibility, adaptability and easy of scale, representing the ideal solution to many computationally intensive tasks in electrical engineering. All the above features are achieved by seamless and efficient mapping of system functions and re-routing abilities supported by a programming flow that abstracts the hardware implementation details [5, 6].

The most important very large scale integrated circuit that represents the embedded reconfigurable technology is the FPGA (Field Programmable Gate Array) processor. FPGA chips allow hardware level wiring of decision logic by multi-grid computation technique, providing a cost-effective flexible platform for fine-grained parallel computing applications. All those are completed immense computational efficiency matched by rich on-chip interconnectivity, high bandwidth concurrent memory access and complete programming technologies. Therefore, it looks a feasible approach to combine reconfigurable technology with parallelization and to use it in buildings supervising and monitoring systems development and implementation where flexibility and hardware reconfiguration are first importance issues. Hence, instead of using various hardware architectures for different residential or commercial buildings, in this paper a FPGA-based hardware configuration will be presented, which possess the ability of rapidly change its internal structure and wiring configuration between modules. There the key concept is that by using reconfigurable technology, the physical hardware support should be able to “learn and adapt to” each building’s specific configuration and automation level [5, 6].

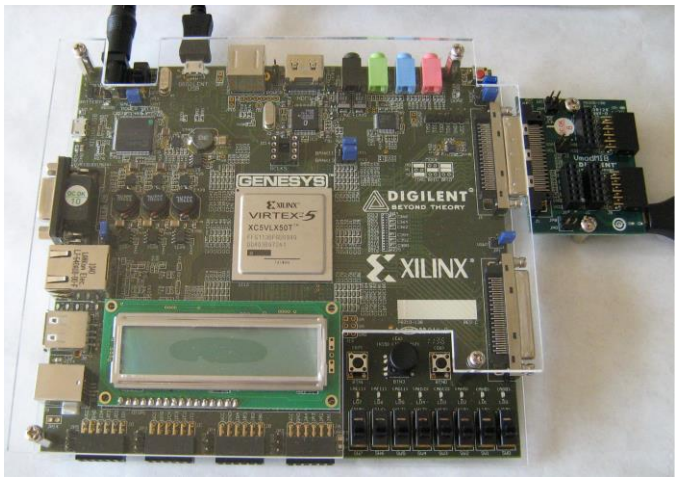

Fig. 1. The Genesys Virtex-5 development board

Considering the above presented theoretical remarks, an adequate chooses to implement high complexity buildings supervising and monitoring systems is the Digilent Inc. manufactured Genesys Virtex-5 FPGA development board (figure 1) [7]. This is one of the latest generation Xilinx FPGA-based development kits, with the following hardware resources: Xilinx Virtex-5 LX50T FPGA, 256Mbyte DDR2 memory, 32Mbyte Numonyx StrataFlash, 100/1000 Ethernet, RS232 port, multiple USB2 ports, HDMI video, AC-97 in-out lines, 100MHz fixed oscillator and (up to) 400MHz programmable clock generator, 112 FPGA I/O, 8 LEDs, two-axis navigation switch, 8 slide switches, and a 16x2 character LCD display [7]. In the followings a concrete building supervising and monitoring system implementation example it is presented, where the listed hardware resources have been shared to develop a versatile hardware framework also well suited to a wide range of very similar applications.

III. BUILDING SUPERVISING AND MONITORING SYSTEM IMPLEMENTATION EXAMPLE IN RECONFIGURABLE HARDWARE TECHNOLOGY

For the implementation example let’s consider the building’s cadastral plan (in a .jpeg or .bmp format) of an arbitrary chosen building as it is presented in figure 2. This plan indicates the building rooms and their arrangement, sizes of the rooms, the placement of doors and windows, the location of lighting elements, fan coils, or other heating units.

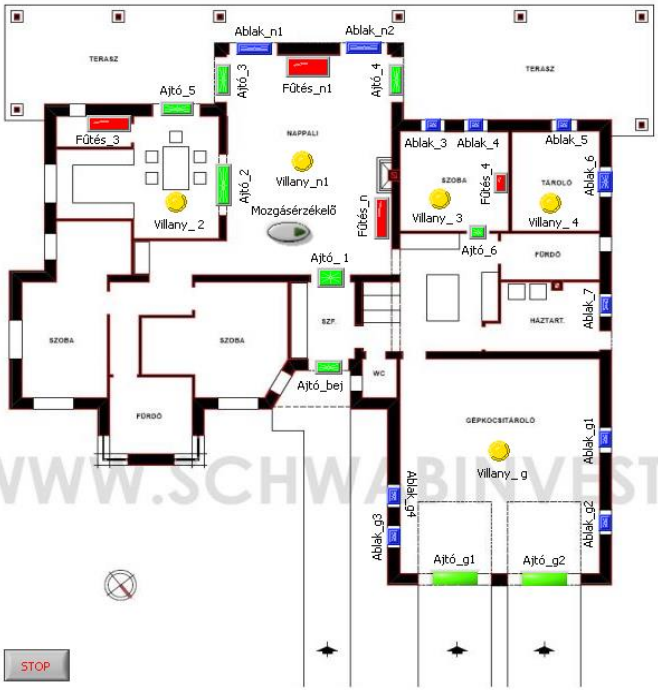

Fig. 2. The cadastral plan of an arbitrary chosen building configuration [8]

The graphical elements placed on the figure are arranged according to the following rules:

- the doors are marked with green squares;
- the windows are marked with blue squares;

- the heating elements are marked with red squares;
- the lighting elements are marked with yellow circles;
- the motion sensors are marked with gray oval circles.

The motion detection sensors were placed to detect if there are persons inside the building (or enters during the night time), and they can be programmed with arbitrary time intervals to turn off lighting elements or close doors. In the front panel of the LabView program presented in figure 3 it is indicated that the entire building is supervised and monitored by a special program that continuously checks the off/on state (open or closed switches) of each door, window, lighting element, fan coil, or heating element [8]. This monitoring system is interconnected to the alarm system of the building.

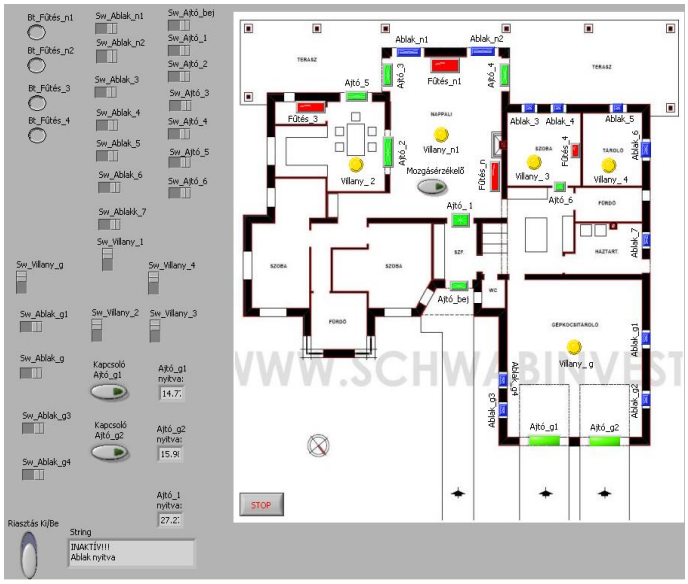

Fig. 3. The Front Panel of the LabView software-based building supervising and monitoring system [8]

In case of any unwanted events or persons entrance the building supervising and monitoring system can generate voice signals, phone calls, or display alerting messages.

The main purpose of the paper is to implement a real-time control strategy for the above introduced building supervising and monitoring system by using the Genesys Virtex-5 FPGA development board hardware facilities. First of all, it is important to mention here that as it has been presented in the previous paragraph, the board provides 112 FPGA digital input/output bits to communicate with external devices or processes. This high number of communication lines looks full sufficient in the vast majority of buildings considered for expertise and implementation. Toward, in order to ensure a rapid and efficient development, instead of the VHDL (Very High Speed Hardware Description Language) technology, it has been preferred the most evaluate MicroBlaze technology embedded in the Xilinx Platform Studio EDK (Embedded Development Kit) software toolkit [xx]. This platform allows the implementation of a single-processor based system in MicroBlaze technology easy to be programmed in a *C/C++* language environment. Basically, the MicroBlaze is a 32 bit Wishbone compatible full-featured and FPGA optimized RISC (Reduced Instruction Set Computer) soft processor for use in FPGA designs applications. It's all instructions are 32-bits wide and executable in a single clock cycle, with fast register access, zero-wait state block RAM, and true dual port access possibility. Many aspects of the MicroBlaze can be user configured: cache size, pipeline depth (3-stage or 5-stage), embedded peripherals, memory management unit, and bus-interfaces can be customized. Other notable features include floating point unit, instruction and data caches, hardware exception support, error correction codes, or low-latency interrupt mode. The MicroBlaze has a versatile interconnect system to support a large scale of embedded applications. To communicate with these applications uses as a primary I/O bus the core connected PLB (Processor Local Bus) bus with master/slave capability, and the LMB (Local Memory Bus) bus to access the local memory. In figure 4 it is presented a MicroBlaze single-processor configuration settings example in the Xilinx Platform Studio EDK software environment.

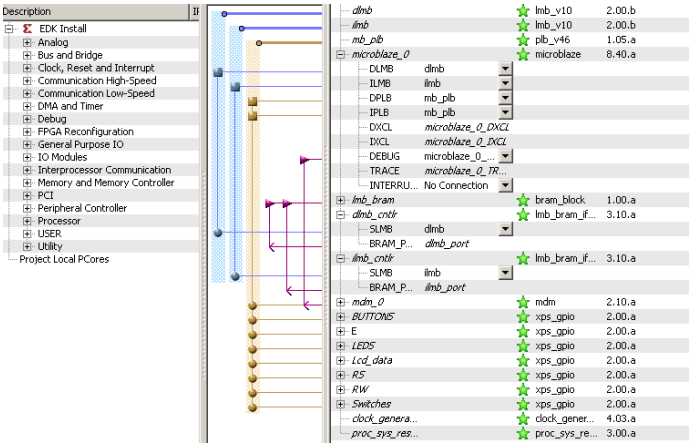

Fig. 4. The MicroBlaze single-processor configuration settings in the Xilinx Platform Studio EDK environment

There it can be observed that in order to increase the communication efficiency the MicroBlaze processor also uses two additional buses: the *dlmb* data local memory bus, and the *ilmb* instruction local memory bus.

In fact, designers use the XPS (Xilinx Platform Studio) module to configure all the hardware specification of their embedded system, including the processor core, the memory controller, I/O peripherals, etc. Therefore, in the specific case of the building supervising and monitoring system expressed by the LabView Front Panel diagram from figure 3 it is necessary to specify all the communication lines with the added interfaces, peripherals, sensors, or other devices. Considering that the open/closed state of the doors, windows, lighting elements, heating elements, or motion sensors inside the entire building is monitored continuously by adequate sensors which emits on/off (logical 1/0 signals), it means that the MicroBlaze processor should be interconnected with simple digital I/O ports. Obviously, for this purposes a maximal number of 112 FPGA digital input/output bits are available which must be individually wired to the above

mentioned sensors. In order to do not complicate excessively the FPGA-based control system implementation example, let's consider a simple situation when the 8 slide switches available on the development board will be used as signal generators, indicating the doors, windows, lighting elements, or heating elements open/closed state. At the same time, the 8 LED diodes can be used to indicate these states via light signals, and the available 2x16 character LCD display to print text messages or alerting messages in case of any event is happen.

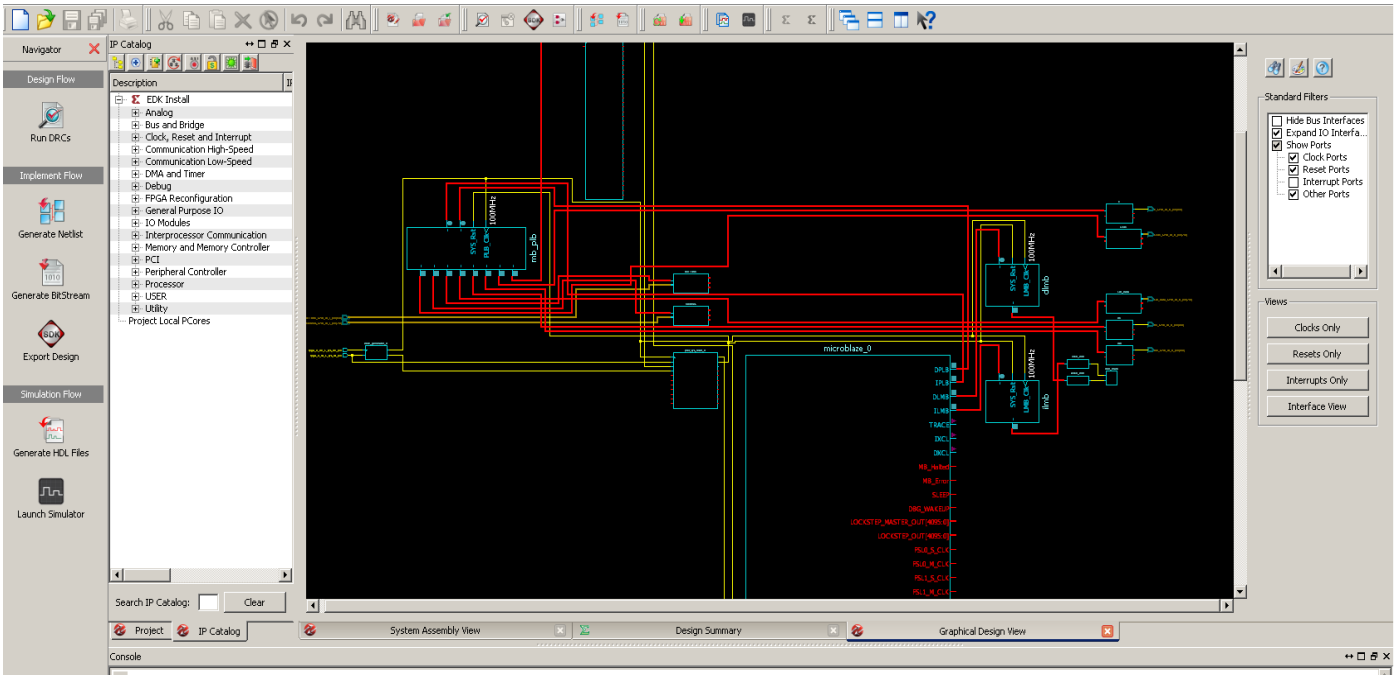

Fig. 5. The block diagram of the implemented hardware system (Graphical Block Design) in the Xilinx Platform Studio EDK environment

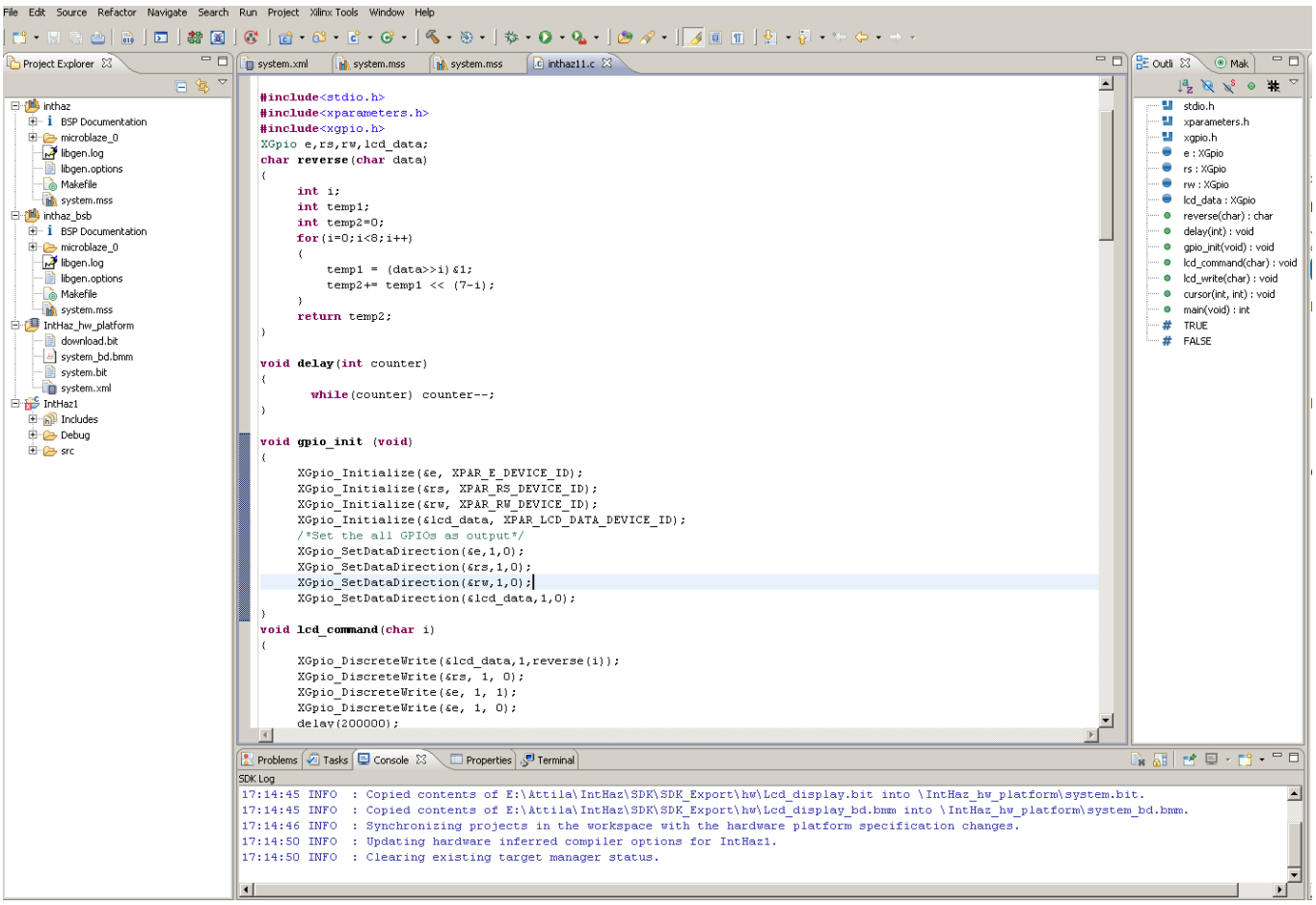

Fig. 6. Piece of the implemented C/C++ language control algorithm in the SDK component

In figure 5 it is presented the block diagram of the implemented hardware system (Graphical Block Design) in the Xilinx Platform Studio EDK environment. There it can be observed the software configured MicroBlaze processor, connected I/O port and the LCD display module. The SDK (Software Development Kit) module handles the software that will execute on the embedded system and enables to write, compile, and debug C/C++ applications for the embedded hardware system. A piece of the algorithm implemented in SDK for the building supervising and monitoring application is shown above in figure 6. In order to illustrate the advantages of the implementation in the followings it is presented a short section from the implemented program *C* language source code.

```
void gpio_init (void)
{
    XGpio_Initialize(&e, XPAR_E_DEVICE_ID);
    XGpio_Initialize(&rs, XPAR_RS_DEVICE_ID);
    XGpio_Initialize(&rw, XPAR_RW_DEVICE_ID);
    XGpio_Initialize(&lcd_data, XPAR_LCD_DATA_DEVICE_ID);
    /*Set the all GPIOs as output*/
    XGpio_SetDataDirection(&e,1,0);
    XGpio_SetDataDirection(&rs,1,0);
    XGpio_SetDataDirection(&rw,1,0);
    XGpio_SetDataDirection(&lcd_data,1,0);
}
void lcd_command(char i)
{
    XGpio_DiscreteWrite(&lcd_data,1,reverse(i));
    XGpio_DiscreteWrite(&rs, 1, 0);
    XGpio_DiscreteWrite(&e, 1, 1);
    XGpio_DiscreteWrite(&e, 1, 0);
    delay(200000);
}

void lcd_write(char i)
{
     XGpio_DiscreteWrite(&lcd_data, 1, reverse(i));
     XGpio_DiscreteWrite(&rs, 1, 1);
     XGpio_DiscreteWrite(&e, 1, 1);
     XGpio_DiscreteWrite(&e, 1, 0);
     delay(200000);
}

void cursor(int row, int column)
{
    if(row==1)
        lcd_command(0x80|(column-1));
    if (row==2)
       lcd_command(0x80|(0x40+(column-1)));
}

int main(void)
{
                XGpio Leds, Switches;
                int data,n,c1,c2,c3;
                char c4,c5,c6;
                typedef int bool;
                #define TRUE  1
                #define FALSE 0
                bool f = FALSE;
     gpio_init();
     XGpio_DiscreteWrite(&rw,1,0);
     delay(20000);
     lcd_command(0x38);
     lcd_command(0x01);
     lcd_command(0x06);
     lcd_command(0x0C);
     XGpio_Initialize(&Leds,XPAR_LEDS_DEVICE_ID);
     XGpio_SetDataDirection(&Leds, 1, 0);
     XGpio_Initialize(&Switches,XPAR_SWITCHES_DEVICE_ID);
     XGpio_SetDataDirection(&Switches, 1, 1);
     data=0;
     while(1)
     {
               data = XGpio_DiscreteRead(&Switches, 1);
       XGpio_DiscreteWrite(&Leds, 1, data);
       n=data;
       c1=n%10;
       c2=(n%100-c1)/10;
       c3=(n-c1)/100;
       if (data > 0  && data <= 3  &&  f == FALSE)
       {
                lcd_write('A');
                lcd_write('j');
                lcd_write('t');
                lcd_write('o');
                c6=c3+'0';
                lcd_write(c6);
                c5=c2+'0';
                lcd_write(c5);
                c4=c1+'0';
                lcd_write(c4);
                cursor(2,1);
                lcd_write('K');
                lcd_write('i');
                lcd_write('n');
                lcd_write('y');
                lcd_write('i');
                lcd_write('t');
                lcd_write('v');
                lcd_write('a');
                f = TRUE;
                lcd_command(0x02);
            }
         if (data > 3  && data <= 8  && f == FALSE)
             {
                lcd_write('A');
                lcd_write('b');
                lcd_write('l');
                lcd_write('a');
                lcd_write('k');
                c6=c3+'0';
                lcd_write(c6);
                c5=c2+'0';
                lcd_write(c5);
                c4=c1+'0';
                lcd_write(c4);
                cursor(2,1);
                lcd_write('K');
                lcd_write('i');
                lcd_write('n');
                lcd_write('y');
                lcd_write('i');
                lcd_write('t');
                lcd_write('v');
                lcd_write('a');
                f = TRUE;
                lcd_command(0x02);
                   }
```

The first procedures of the above listed source code refers to the general purpose inputs/outputs (GPIO) initialization, to set the LCD device communication lines and to set up its operation mode. Then it is programmed the character write procedure to the 2x16 position LCD display. The main *C* procedure contains instructions to continuously check the state of the 8 slide switches, and as function of their on/off state position the available 8 LEDs on the board will be programmed to display clipping-light alerting signals. At the same time the LCD device displays the state of the switch on which has monitored an external event, respectively indicates that a door, window, lighting element, or a heating element it has turn into the *on* or *off* state. This information is completed

then by alerting or other user warning messages. However, the main idea of this software development is to prove that the FPGA-based reconfigurable hardware technology can be adapted very conveniently for a wide range of building supervising and monitoring applications, and is well suited to fulfill on all the most demanding user requirements.

Fig. 7. General view of the experimented laboratory setup

In figure 7 it is presented a picture capturing the experimenting operations of the building supervising and monitoring example real-time implementation on the laboratory setup. This development system represents a well fitted hardware solution for a large scale of different complexity building automation applications implementation. It can be utilized without any change in its hardware architecture, just the adaptation of the software algorithm to "learn and adapt to" the building's specific structure and automation level is required.

IV. CONCLUSION

The paper outlines novel facilities and solutions offered by the current level microelectronic technologies in building supervising and monitoring systems design and development. There are highlighted the benefits and advantages of the reconfigurable hardware systems application by presenting a concrete implementation example. The main unit of this system has been built upon a Genesys Virtex-5 FPGA-based development board, as a high speed, parallel, and distributed computing reconfigurable unit, which hardware module has been programmed with last generation MicroBlaze technology allowing fast and convenient implementation of sophisticated control algorithms. The basic conclusion is that it is more efficient and practical to use a general-purpose hardware system architecture that possess the ability to "learn and adapt to" each building's specific mechatronics system structure or automation level implementation requirement. In other words, it is recommended to use a well fitted hardware framework being able to fulfill a wide range of building supervising and monitoring applications in a large scale of different type residential or commercial buildings. The result of the entire development is a powerful and versatile system representing a well suitable solution for the most sophisticated and demanding customer needs in building supervising and monitoring applications.

REFERENCES

[1] M. Deru and P. Torcellini, "Improving Sustainability of Buildings through a Performance-based Design Approach" Preprint NREL Report No CP-550-36276, World Renewable Energy Congress VIII, Denver, CO: www.nrel.gov/docs/fy04osti/36276.pdf, 2004.

[2] M. Deru and P. Torcellini, "Improving Sustainability of Buildings through a Performance-based Design Approach" Technical Report NREL/TP-550-38617, Golden CO: National Renewable Energy Lab. www.nrel.gov/docs/fy04osti/38617.pdf, 2006.

[3] P. Torcellini, S. Pless, M. Deru, D. Crawley, "Zero Energy Buildings: A Critical look at the Definition" ACEEE Summer Study, Pacific Grove, California, NREL/CP-550-39833, 2006

[4] C. Barley, M. Deru, S. Pless, P. Torcellini, "Procedure for Measuring and Reporting Commercial Building Energy Performance" Technical Report NREL/TP-550-38601, Golden CO: National Renewable Energy Lab. www.nrel.gov/docs/fy06osti/38601.pdf, 2005.

[5] G. Husi, Cs. Szász, V. H. Hashimoto, "Application of reconfigurable hardware technology in the development and implementation of building automation systems", Environmental Engineering and Management Journal, November 2014, Vol. 13, No. 11, PrintISSN: 1582-9596, eISSN: 1843-3707, http://omicron.ch.tuiasi/EEMJ.

[6] G. Husi, Cs. Szász , "Building Automation Technology in Electrical and Mechatronics Department in Debrecen", 16 Ulusal Makina Teorisi Sempozyumu, Atatürk Üniversitesi, Mühendislik Fakültesi 12-13 Eylül, 2013, pp. 671-674.

[7] Digilent Co., Genesys Virtex-5 User's Giude, digilentinc.com/genesys-virtex-5-fpga-development-board.

[8] Cs. Szász, G. Husi, "LabView Software-based Building Supervising and Events Monitor System for Net Zero-energy Strategies Implementation", Recent Innovations in Mechatronics (RIiM), International Journal of Mechatronics Science, Electrical Engineering and Mechatronics Department, Faculty of Engineering, University of Debrecen, Hungary, Vol 1, 2014, No.1, ISSN:2064-9622. http://riim.lib.unideb.hu/index.php?oldal=aktualisszam&folyoirat_szam,