

Last Generation Mechatronics: a Two-level Platform-based Reconfigurable Technology Approach

Csaba SZÁSZ

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

Abstract—As is well known mechatronic systems currently being designed and developed are often difficult multidisciplinary undertakings. Based on the intrinsic coupling of different implementation technologies, efficient design of mechatronic systems is of primordial importance for development of next generation industrial products. This paper is focused to current and future technological trends aimed to improve the design and implementation processes of mechatronic systems in an increasingly harsh industrial environment. Special attention is dedicated to introduction of the two-level platform-based reconfigurable technology approach. This strategy efficiently combines major advantages of both the hardware and software platform-based development trends in modern mechatronic systems. In order to support the unfolded theoretical arguments a last generation and versatile mechatronic system development is presented and discussed in the paper. The mentioned trends can be used as rough orientation for future mechatronic systems research and implementation activities.

Keywords—*mechatronic system; reconfigurable technology; two-level platform; software platform; hardware system.*

I. INTRODUCTION

In the related scientific literature mechatronics is defined as a multidisciplinary science domain that overlaps high level knowledge from very different areas such as mechanics, hydraulics, electrical engineering, electrical machines and actuators, electronics, computer science, digital hardware, systems theory, or software technologies. However, mechatronic systems play an ever more important role in the modern industry, with a wide range of applications in automotive industry, consumer goods, or a large scale of industrial products. The emergence of mechatronics with its growing importance in nowadays industry is boosted by two main factors [1]. One is linked to the market pull that means the modern market requirements imposed in a harsh industrial environment. The other is the technological push that embeds all the technological trends aimed to design and development of next generation mechatronic products. These two drivers imposes a lot of high level requirements that guide the most important trends in last generation mechatronic systems design and implementation. Among these may be evidenced the followings ranked in reference [1]:

- a new mechatronic product shall always be smaller and cheaper than its previous counterpart;
- a new mechatronic product shall always provide additional functionality than its previous counterpart;
- customers require more and more individualized products for their wishes;
- human machine interface becomes a key differentiation factor among the new products;
- electronic components will gradually replace mechanical components in mechatronic systems;
- electromechanical systems will increasingly be used as intelligent sensors;
- the microelectronic circuits double its complexity in each 1.5-2 years (Moore's law);
- a powerful wish for physical mobility becomes dominant (connection to Internet, connection to Virtual Reality Applications, etc.);
- the software layer will increasingly dominate the functions, behaviors, and overall quality of next generation mechatronic products.

Of course, the above mentioned trends are still not enough to ensure a comprehensive overview of all the changes that comprise recent year's mechatronics and is also not the main scope of this paper. However, they may be near sufficient to set up an appropriate evaluation of the main evolutions that comprise this challenging science domain. Once possessing a well fitting "diagnosis" becomes possible to establish an effective as well as efficient design strategy of mechatronic systems in order to reach more competitiveness in nowadays industrial and economical environment.

Obviously, it is not difficult also to assume that modern mechatronic systems structure are so complex and sophisticated that their design become intricate at all. This bottleneck claims new approaches being able to manage this complexity and new paradigms for mechatronic systems design are needed. Modularization and structuring functions may be one way in this endeavor. Generally speaking,

modularization and structuring are two important aids for solution of the complexity problem in electrical engineering. This means that the entire system is divided into independent modules or sub-units with its own well defined functions. According to this scenario each module will be handled independently during the entire design and development process but at the final end they will be aggregated back (following a well defined hierarchical structure) into the main system architecture. A similar development strategy that has been specially conceived for mechatronic systems laid on last generation reconfigurable hardware technology is unfolded next in the paper.

II. TWO-LEVEL PLATFORM-BASED RECONFIGURABLE TECHNOLOGY APPROACH

Scientists on the computer side of mechatronics predicts that the future of mechatronic systems move towards to a high degree of adaptability and self-organization [2]. According to this viewpoint mechatronic systems also will exhibit a high degree of concurrency by embedding a set of individual units running in parallel [2]. On the other side, it is well known that by increasing the microprocessors frequency rate may be covered the increasing scale requirement and complexity of

high performance mechatronic systems [3]. Unfortunately, the clock scale even of last generation microprocessors cannot be increased unlimited. There are important technological reasons that limit this steady performance increase and since several years this frequency scale nearly has stagnated. Hence, in order to match the timing scale constraints of last-generation mechatronic systems, the platform-based design strategy has been introduced [3]. According to this approach the structure and heterogeneity of the application should be exploited by an adequate platform that exploits the available intrinsic parallelism with the right granularity. This theoretical concept has been defined and explained in detail in reference [3].

In the followings let's assume as theoretical background the above mentioned platform-based approach. This development strategy may be extended and completed toward to an original design flow embedding more integration levels, increased modularity, and improved high-level synthesis. Such platform will operate as a high level integrated design environment (IDE) improving both the hardware and software design flow of complex mechatronic systems. The difference expressed by this proposal consist on the basic idea that there a two-level platform-based reconfigurable technology approach is introduced, as shown next in figure 1.

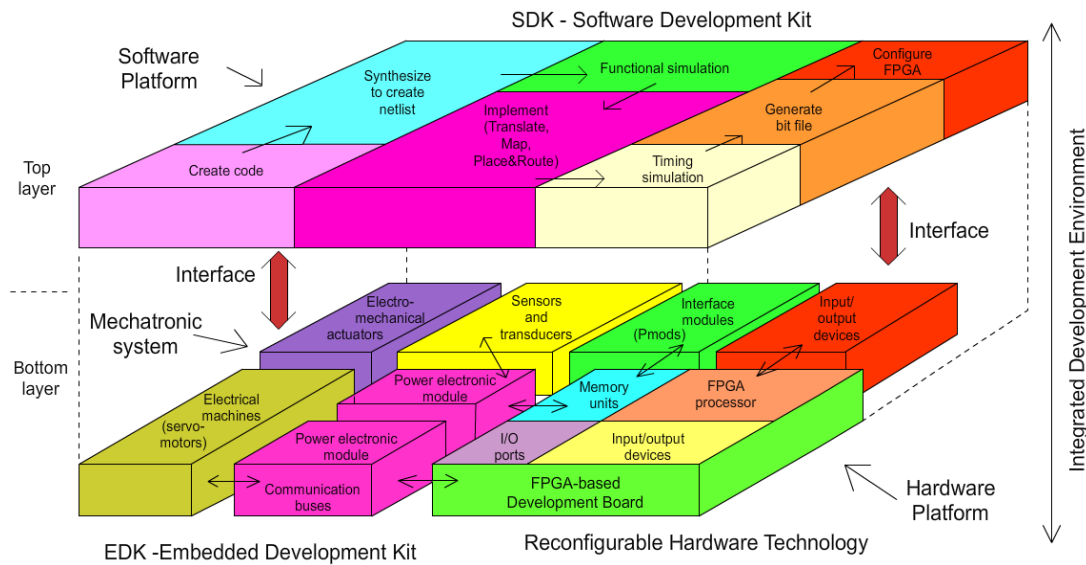


Fig. 1. The two-level platform-based reconfigurable technology approach

The hardware platform embeds all the main physical components and hardware units of a complex mechatronic system and forms the bottom layer of the presented model. The upper layer represents the software development platform that covers all the software technologies and program units required for the application control strategy implementation. Of course, the two layers communicate with each other via an adequate interface which includes data and control busses and the necessary communication protocols. Because of their complexity the two above mentioned layers will be discussed separately in the next paragraphs.

III. THE HARDWARE PLATFORM

As it has been mentioned before, this layer is composed only by physical modules, in our case the main units of a classical mechatronic system. In a most common situation let's suppose that the considered application also requires the use of electrical machines (or various servomotors) for motion control implementation or other electromechanical actuators such as electric valves, electric drive supapes, relays-based actuators, etc. They are driven by the adequate power electronic modules as in expressed in the figure. To these actuators may be interconnected a large set of various type sensors to measure physical magnitudes (position, velocity,

torque, current, voltage, pressure, light intensity, liquid or gas flow, humidity, etc.). With this information desired closed-loop control strategies may be implemented. The digital control unit of the entire system is a Field Programmable Gate Array (FPGA) processor-based development board. This is in fact a sophisticated and high performance ready-to-use manufacturer development platform embedding hardware reconfigurable technology. This technology represents perhaps one of the most challenging paradigms for hardware systems development and implementation. The central unit of this module is the FPGA processor that is the most important general purpose reconfigurable processor with a leading role in reconfigurable computing strategies implementation. These chips combine the high performance of hardware with some of the flexibility of software by using the ability of hardware architectures to rapidly change its internal structure and functionality regarding the various user needs. Such high performance reconfigurable hardware systems also exploit the advantages of the fine-grained instruction level parallelism as well coarse-grained functional parallelism, by using custom computing technologies [4]. Therefore, FPGAs possess the ability of parallelization and parallel computing, being ideally suited for distributed task solving or network computing applications. They also provide a cost-effective solution for complex real-time and computationally intensive multitasking processing purposes in mechatronics [4]. More over FPGAs allow multi-grid computation to be an ideal platform for fine-grained parallel computing, representing the perfect solution for the implementation of highly concurrent control, communication, monitoring, and supervising tasks required in mechatronic systems development and implementation. Additionally, reconfigurable hardware technology offers the huge advantages of flexibility, low-power consumption, speed, adaptability and ease of scale, represents the complete solution to a wide range of computationally intensive developments and applications [4].

Of course, the FPGA chip usually is interfaced with a wide range of input/output devices and interfaces, such as push buttons, switches, LED displays, 7 segment displays, A/D and D/A converters, timers/counters, PWM outputs, Pmod connectors, or LCD displays. Furthermore, via the Pmod connectors the system may be extended with additional A/D and D/A converter units, displays, various communication modules (R232, Bluetooth, GPS, RS485, etc.), light sensor units, temperature sensor modules, or even power electronic converters for motor control applications. Such devices are generically named Pmods and during the entire hardware development process they can be handled and interfaced individually in a very convenient manner. By using the EDK (Xilinx Embedded Development Kit) system that allows the implementation of the MicroBlaze processor technology [5], all the above components may be integrated together in a compact and powerful hardware development platform well suited for complex mechatronic systems implementation.

IV. THE SOFTWARE PLATFORM

It is no doubt that among the implementation technologies for mechatronic systems software has a distinctive or specific

role. This because of software comprises the entire system complexity being used as an integration platform. On the other side software it has no physical border conditions therefore at technological level software is easier to deal with [1].

However, the hardware platform implementation mandatory is followed by the software layer development of the considered application. In case of the FPGA-based development systems that are built upon hardware reconfigurable technology dedicated software development kits are used. These are complex software platforms that allow user tailored design via a set of well defined steps and programming operations that form together a versatile and powerful implementation technology. Of course, at the beginning of this algorithm some input data must be specified that covers information about specification of the platform, timing constraints, or application and mapping instances. This initial step is followed by the concrete code source elaboration for the application. Then starts a synthesis processes flow that converts the above mentioned input specifications into an FPGA implementation. This is composed by two different trajectories: one is a hardware synthesis that generates HDL (Hardware Description Language) code and the other is the software synthesis that generates application binaries [3]. To this end, in order to transform a platform specification to a VHDL (Very High Speed HDL) code the platform generator performs five main steps: Model parsing, Component initiation, Component connection, Top-level VHDL file creation, and Switch configuration. Without the claim to enter here into more details, there it is only mentioned that model parsing means that specifications are parsed into a data structure (a list of the connections and platform resources). The component installation refers to a VHDL code that initiates a component including its generic map corresponding to the platform resource attributes from the platform instance model. Component connection also means a VHDL code that specifies the component port map. The top-level VHDL file is generated by the platform generator including among others all the relevant external connections. In a last stage for each switch a specific file is generated that contains the data for the static switch routing table. Finally, the generated whole VHDL code can be synthesized into an FPGA bitstream file that is uploaded to the FPGA core via a configuration interface (generic vendor tools) [3]. All these software components are tight integrated into a versatile, high performance, and user-friendly compact software development platform generically named SDK (Xilinx Software Development Kit) [6].

Obviously, all the above pointed design steps and methods cover a software implementation technology which seems to be very sophisticated. Fortunately, a concrete mechatronic system design and implementation example can facilitate a better understanding of all the mentioned design processes. In this way becomes possible also to understand the entire two-level platform-based reconfigurable technology approach expressed before in figure 1.

V. TWO-LEVEL PLATFORM-BASED IMPLEMENTATION EXAMPLE OF A MECHATRONIC SYSTEM

In order to present a full two-level platform-based implementation example let's consider the block diagram of a mechatronic system as shown in figure 2. The main module of this system is the Zybo Zynq-7000 ARM/FPGA Development board [7]. This is a high performance FPGA processor-based ready-to-use manufacturer board well suited for sophisticated control algorithms implementation in demanding application. Zybo is a feature-rich embedded software and digital circuit development platform built around the Xilinx Zynq-7000 family [7]. In detail, the board integrates a dual-core ARM Cortex-A9 processor with a Xilinx 7-series FPGA circuit that is interfaced with a rich set of peripherals (Pmod-type modules, on-board memories, audio/video inputs/outputs).

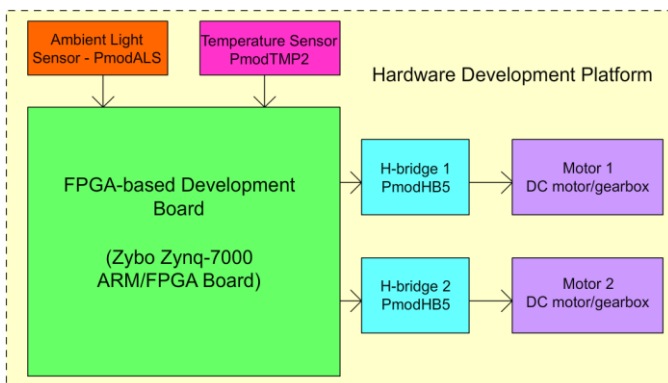


Fig. 2. Block diagram of the mechatronic system

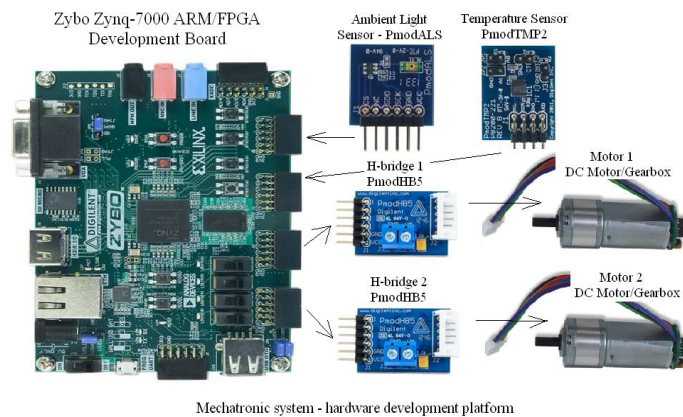


Fig. 3. General view of the mechatronic system - the hardware development platform

Among the most important features of the Zybo Zynq-7000 Development board may be ranked the followings: 249 kB Block RAM, 80 DSP (Digital Signal Processor) slices, on-chip dual channel 12-bit 1 MSPS analog-to-digital converter (XADC), on-board JTAG programming, UART to USB converter, 512 Mb x 32 DDR3 memory block (1050 Mbps bandwidth), 650 MHz dual-core Cortex-A9 processor, 1G Ethernet and USB2.0 communication ports, VGA output port, 128 Mb Serial Flash interface, audio codec with headphone out, microphone and line in jacks, and general

purpose input/output (GPIO) interfaces embedding 6 pushbuttons, 4 slide switches, 5 LED diodes, and 6 Pmod-type ports with 6 pin communication bus [7]. The above mentioned facilities and resources provide an ultra-low-cost alternative for designers who require massive parallel processing power and wide hardware flexibility.

As shown in figure 2, the Zybo Zynq-7000 system has been interconnected with several interface modules. Two ready-to-use sensor modules will provide accurate information from the development system's near environment. The first is a compact and small PCB size ambient light sensor that converts light to digital data with 8-bit resolution [8]. The next is a 16-bit high accuracy temperature sensor (*PmodTMP3*) using the I²C communication protocol for data transmission [8]. Therefore are measured the light intensity and temperature of the environment and as function of these information will be controlled the two DC motors also interfaced to the mechatronic system. These are two motor/gearbox (1:19 gear ration) with rugged, heavy duty construction that include encoder for sensing rotation and speed [8]. They are interfaced to the Zybo Zynq-7000 system via two compact *PmodHB5* modules. These are in fact a 2A H-bridge converter units specially manufactured to drive small to medium sized DC motors with operating voltage up to 12V. As is possible to observe from figure 3, they all forms together a versatile mechatronic system and represents the hardware development platform (the hardware layer) of the model shown in figure 1.

The next issue is how to integrate all these individual or stand-alone components and modules into a powerful and flexible hardware platform of a mechatronic system. By the way, for this purpose has been developed the already mentioned EDK system specially targeted to solve such kind of design and development tasks. In this paper the EDK system implemented under the Vivado Design Suite (VDS) will be used [9]. VDS represents a new approach for broadly deploying system platforms using next generation C/C++ and IP-based (Intellectual Property) high productivity design. Such technology allows reduced design and implementation cycles, short analysis and verification convergence. Let's try it all these remarkable facilities in case of the mechatronic system plotted in figure 3. For this purpose is started the *Vivado HLX Edition* toolkit to create a new Vivado project. Among the first steps of this development should be mentioned that the project will be implemented on hardware frame of the Zybo Zynq board, as is marked in figure 4.

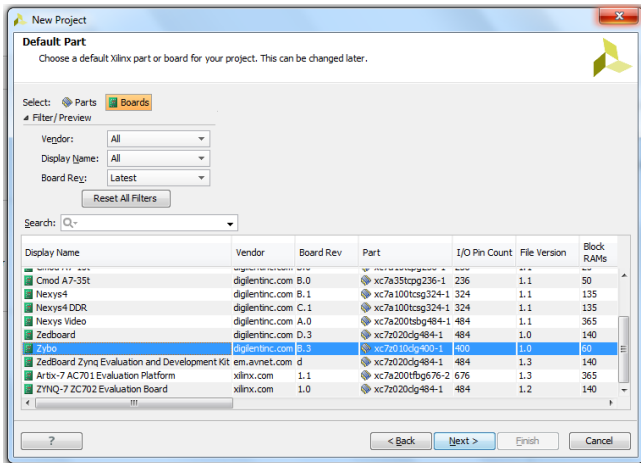


Fig. 4. Choosing the Zybo board hardware frame for development and implementation

Then, should be interfaced all the components of the mechatronic system to the Zybo Zynq-7000 development board. In this endeavor it is chosen the adequate Pmod connector for each module in order to ensure their physical communication with the FPGA processor. For example, let's configure the *JB* connector of the board to communicate with the module *PmodHB5 (H-bridge_1)* that drives the *Motor_1*, and *JD* with *H-bridge_2* driving *Motor_2*. In a similar way connector *JC* will be interfaced to the ambient light sensor *PmodALS*, and connector *JE* to the temperature sensor *PmodTMP3*. All these configuration steps are evidenced on the menu panel plotted next in figure 5.

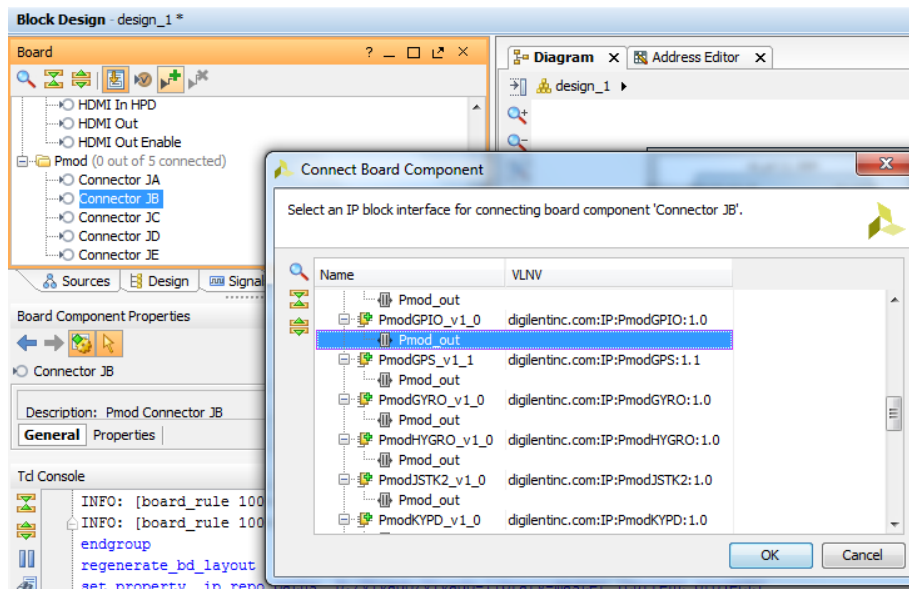


Fig. 5. The interface modules connection to the Zybo Zynq-7000 development board

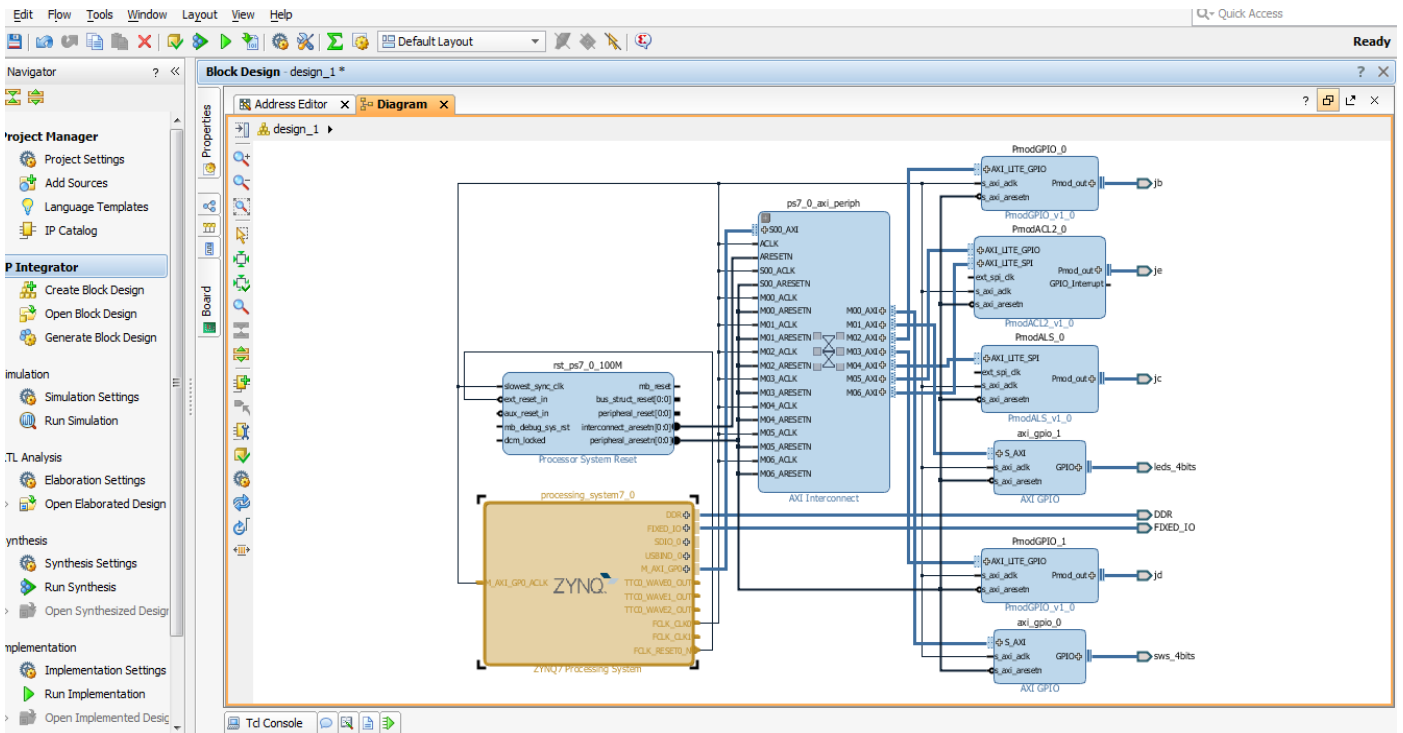


Fig. 6. The interface modules connection to the Zybo Zynq-7000 development board

After several very similar setting operations results the block diagram presented in figure 6. This digital system expresses the final configuration of the implemented hardware platform that represents the bottom layer of the two-level platform-based approach introduced in the first paragraph. What is important to observe here is the efficiency and high flexibility of the implemented digital structure. This technology allows rapid generation of the platform connectivity design along with the necessary software stack. In this way are dramatically shortened all the necessary design and verification times. The entire development flow provide for a high level of synthesis and analysis of complex VHDL designs, rapid configuration, separation of platform development and differentiated logic, and focusing designers on the application.

Obviously, the hardware platform development should be followed by the software platform implementation that represents the top-layer of the two-level platform-based approach. For this purpose the created hardware platform configuration is uploaded into the SDK environment.

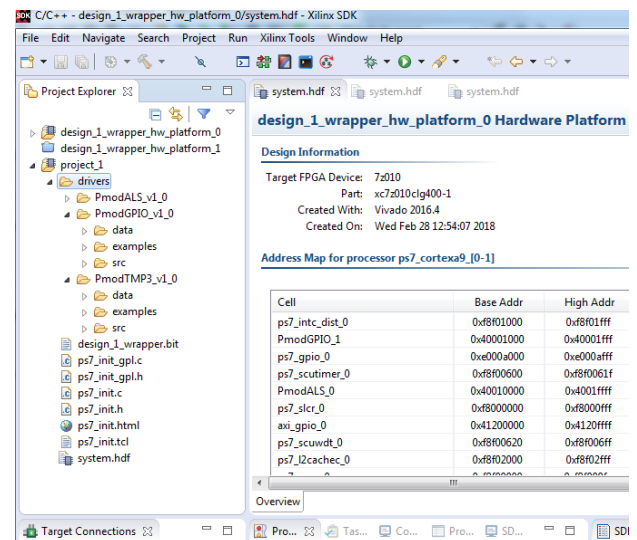


Fig. 7. The hardware platform uploaded into the SDK environment

Figure 7 plots the main menu where in left side are listed all the hardware components that embeds the previously created hardware platform. This include in order the *PmodALS* light sensor, the *PmodGPIOs* (4 switches and 4 LEDs), and two connectors for the *PmodHB5* bridges), respectively the temperature sensor *PmodALS* as basic parts of the software driver components. A more detailed view of the entire project development in the SDK toolkit is shown next in figure 8. There are listed under the driver modules all the required C source code files that make up the integrated software platform. Additionally, in right side of the figure is also listed

a piece of the main C program source code for the mechatronic system control. As is possible to observe, *Vivado HLX* highly accelerates the entire design and development implementation of the mechatronic system by enabling C/C++ source codes to be directly synthesized into VHDL. Therefore remarks as both IP integrator for rapid implementation and system integration directly from C to silicon technology. In this way generates a coherent and highly efficient software package for desired applications.

VI. CONCLUSIONS

The paper presents a two-layer platform-based design and development approach for mechatronic systems. Such strategy can serve not only a rough orientation but also may represent a high efficiency and useful solution for last-generation mechatronic systems implementation.

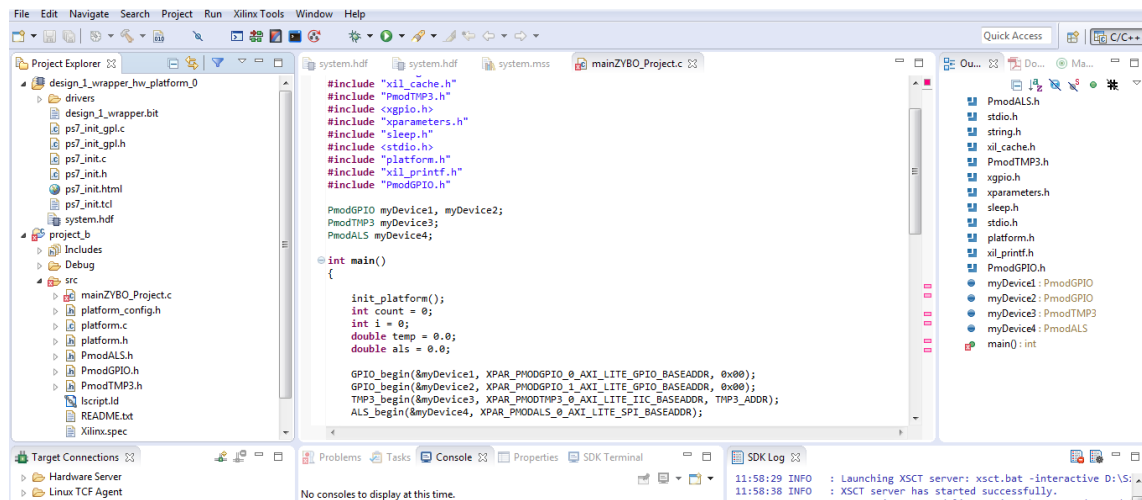


Fig. 8. General view of the software platform implemented in SDK and the main program C source code

References

- [1] W. Dieterle, "Mechatronic Systems: Industrial Applications and Modern Design Methodologies" Elsevier IFAC Publications, Sydney, 2004, www.elsevier.com/locate/ifac
- [2] W. Schaffer, H. Wehrheim, "The Challenges of Building Advanced Mechatronic Systems", IEEE Computer Society: Int. Conference on Future of Software Engineering, 0-7695-2829-5/07, 2007.
- [3] R. Frijns, "Platform-based Design for High-performance Mechatronic Systems" technische Universiteit Eindhoven, 2015, ISBN: 978-90-386-3826-3.
- [4] Cs. Szász and G. Husi, "The Intelligent Building Definition: A Central European Approach", 2014 IEEE/SCIE International Symposium on System Integration, December 13-15, Tokyo, Japan, ISBN: 978-1-4799-6942-5, DOI: 10.1109/SII.2014.7028040, pp. 216-221, 2014.
- [5] <https://www.xilinx.com/support/documentation-navigation/development-tools/mature-products/embedded-development-kit.html>
- [6] https://www.xilinx.com/support/documentation/sw_manuals/xilinx2015_1/SDK_Doc/index.html
- [7] <https://store.digilentinc.com/zybo-zynq-7000-arm-fpga-soc-trainer-board/>
- [8] <https://store.digilentinc.com/pmod-modules/?sort=bestselling&page=2>
- [9] <https://www.xilinx.com/products/design-tools/vivado.html>