

## **EMBEDDED CYBERPHYSICAL SYSTEMS DESIGN CHALLENGES AND MODEL-BASED DESIGN APPROACH**

Larchenko Bogdan

Scientific Supervisor – Dr. of Technical Science, prof. Svetlana Chumachenko

Kharkiv National University of Radio Electronics

(61166, Kharkiv, Nauky ave, 14, Design Automation Department,

(057) 702-13-26)

e-mail: svetlanachumachenko@nure.ua, fax: (057) 702-13-26

This paper provides a review of commonly faced challenges in embedded cyberphysical systems design and a Model-Based Design Approach which helps address various difficulties and complexities in the process of embedded systems development.

When it comes to developing embedded IoT devices, the hardware design is viewed as a critical component for the success of the IoT product. In order to ensure the embedded IoT product meets the required function, consumes low power, and is secure and reliable, a lot of challenges are faced by the embedded IoT device manufacturers during the hardware designing phase of these devices.

The Internet of Things (IoT) is defined as a process in which objects are equipped with sensors, actuators, and processors that involve hardware board design and development, software systems, web APIs, and protocols, which together create a connected environment of embedded systems. When it comes to designing of these embedded IoT systems, they need to be designed for specific functions, possessing qualities of a good product design like low power consumption, secured architecture, reliable processor, etc.

With the rising demand for connected devices, embedded systems need to work with heterogeneous devices and adapt to different networking architectures to cope-up with new functionalities and performances in the real-time environment. Due to this situation of increasing technology adoption and deployment of new applications, embedded system designers face several problems in terms of flexibility while developing embedded IoT systems such as:

1. Ensuring smooth integration of new services;
2. Difficulty in adapting to new environments;
3. Frequent changes in hardware and software facilities;
4. Integration of small size chip with low weight and lesser power consumption;
5. Carrying out energy awareness operations.

All the IoT hardware products need to perform securely in the real-time embedded environment. Since all the embedded components operate in a highly resource-constrained and in physically insecure situations, engineers often face problems in ensuring the security of these embedded components.

Model-based design (MDB) approach proves to be an effective and efficient means of understanding the product parts such as commercial microcontrollers and processors as well as algorithms and code for the working of both microelectronic and embedded devices. When MBD is used effectively, it provides a single design platform to optimize overall system design. The main advantage of this approach is that it helps embedded software developers to understand the difference between simulator and software development tool in order to create simulation models and check whether algorithms will work before the embedded code is written. Through virtual prototyping, system engineers can easily see whether the whole system will work as intended, even before the hardware is manufactured and available for testing.

The model-based design framework typically unfolds in the following way. System modelling activities involve creating a mathematical and behavioural representation of the embedded system under consideration. It refers to a visual method for designing complex control systems, communication systems, and signal processing systems within an MBD framework. With MBD continuous testing as algorithms and real-time computational models are created and refined. During simulation, continuous-time systems are solved using numerical integration. MBD provides a rapid prototyping method of a product. It is a fast and cost-effective way for engineers to control signal processing, verify design at the early stage, and evaluate design trade-offs. After rapid prototyping, a detailed software design activity is performed to convert the controller model to a detailed, executable software specification. Embedded code (often highly optimized) is then generated from the model for the detailed controller model and downloaded to the actual embedded microprocessor or ECU as part of the production software build. To combine hardware and production code into model-based testing, one can compare dynamic outputs of models with data collected through software-in-the-loop and processor-in-the-loop test or with data measured in the test lab, using the data inspector or logging tools.

The main benefit of MBD is the auto-generation of code, which can eliminate human errors and allow code reusability. In addition, model-based design usage consistently provides the benefits of faster time to the first demonstration, faster time to market with a qualitative product, quick turnaround of iterations without the need of hardware, design and continuous testing in order to improve development effectiveness, reusable models which can improve development time and cost.

#### References.

1. Susanna Pantsar-Syvaniemi. (2012). Architecting Embedded Software for Context-Aware Systems. Embedded Systems – Theory and Design Methodology. Rijeka, Croatia.
2. Komal Chauchan, (2018). Why Model-Based Design is important in embedded systems. “eInfochips” . Austin, Texas.