CONNECTION OF MICROCONTROLLER AND MICROCOMPUTER TO DISTRIBUTED SIMULATION

Martin Jakeš [a], Josef Brožek [b]

[a],[b] Department of Software technologies
University of Pardubice
Pardubice, Czech Republic

[a]jakesmar@gmail.com, [b]mail@jobro.czl

ABSTRACT
The paper addresses a complementary part of the problem about making of a sturdy trainer which emulates the dispatching center. A specific problem that is solved, is the need for create new controls for interactive simulation. Controls that physically look like a device from real dispatch center must be connected with a computer simulation. The article is focused on a collection of data from various sensors through a microcontroller, or alone microcomputer. After like further processing, either directly in a microcontroller or through a traditional computer as IBM PC. Both of these methods are compared with each other and with focus on regard to the advantage and disadvantages of each.
The simulation is implemented by using online simulation techniques and interactive simulations. The entire simulator is built on the HLA architecture, because it is a modern and widely used standard (IEEE1516: 2010), with elements of Distributed interactive simulation (IEEE1278).

Keywords: simulation, HLA, microcontroller, distributed simulation, IEEE1516, IEEE1278, HLA convertor, signal processing, interactive simulation.

1. INTRODUCTION
When you are creating simulation systems of some classes it is necessary to take into account user preferences; this is especially true for applications such as trainer. For this type of application it is assumed, that the user effectively handles control of the input device. That simultaneously serves as the input/output means of simulation. That is the reason, why the success of the simulator depends on its appearance. Virtualization environments on a computer is often insufficient, due its low user-friendly operation. In a real situation, there is necessary to create a realistic looking environment, in which it is possible to emulate, through the simulator, various scenarios.

2. STATE OF THE ART
The extensive exploration of solutions to similar issues, we could not find similarly oriented study describing the formation of the sensor as a separate federate care only to provide information about the real world or receiving states of the simulated system and screening of these changes into the real world. One of the exceptions was very outdated publications (3) that it does not capture the current state of the HLA software or hardware that are available today. Another work was found linking minicomputers Raspberry Pi with a mobile device using Android via the RTI DDS technology (2). In the vast majority of publications any real-world sensors are connected to a computer which already represents a functional unit in which the simulation corresponds to a motor vehicle, aircraft or railway station. Or, see some publications performs simulating sensors (4).

The above-mentioned DDS middleware is a competitor HLA. Abbreviation for RTI with DDS means (Real-Time Innovations), which is the name of the company. In HLA RTI means (Run-time infrastructure is a middleware that is required when implementing the High Level Architecture) therefore, these shortcuts do not have nothing in common (for more information and comparisons these technologies resource (1))

2.1. Connecting hardware to computer
For connecting sensors to a computer are commonly used dedicated chips that connect via USB. These chips provide easy connection between computer and hardware without having to know the implementation of the USB communication. Probably the best-known manufacturer is company FTDI. Communication between these chips and hardware (possibly sensors) are mainly serial but chips with a parallel communications also exists (William J. 2002). Using them for interconnection with microcomputer Raspberry Pi is possible but unlike conventional computers, this microcomputer has also dedicated pins witch provide communication directly without the required chip.

3. HLA
High Level Architecture (hereinafter referred as HLA) is a very complex methodology for the creation and operation of distributed simulations. It does not limit the application domain or modelling method (i.e. it is

possible to model through the logical processes, using the principle of decomposition of logical processes and it is possible to create monolithic models, agent-based models, etc.). In addition to the architectural level it specifies many methods at lower levels. There are specified: method of communication between nodes of distributed simulation and the requirements placed only on the actual transmission format (in terms of communication via XML). The major benefit makes possibilities to create heterogeneous simulation models (e.g. using of JAVA and C++ in one simulator). One of the most valuable benefits of HLA is the possibility to adapt an existing simulator for cooperation in HLAs distributed simulation.
A very large standard with a lot of partial documents (IEEE1516: 2010 IEEE1516.1: 2010 IEEE1516.2: 2010 IEEE1516.3: 2010 IEEE1516.4 : 2010) makes it difficult to enter into the HLA problematics.
HLA was originally created for the army of the United States, specifically it has been developed for the Defense Modeling and Simulation Office, on order from the United States Department of Defense. There was a motivation for the interconnection of different simulators, respectively distributed simulation models to more complex simulator clusters. Today HLA it used also in the academic and commercial sector. Most of the non-traditional input and output devices are still the domain of the army only. This is reason, why we focused on developing them.

4. SENSOR / DRIVER CONNECTION PRINCIPLE

Existing IT solutions (e.g. USB signal converter) are not appropriate. As one purpose it could serve to connect the sensors to the computer, but the connection to the computer is not a sufficient condition for the proper functioning of the input device in a distributed interactive simulation.

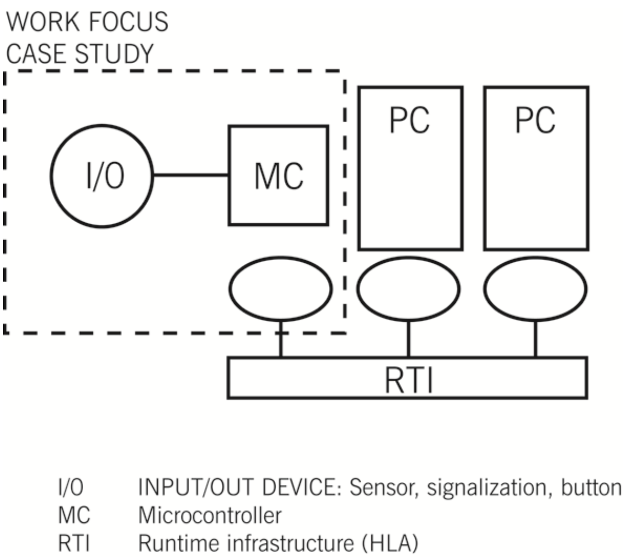

Figure 1: First alternative of situation

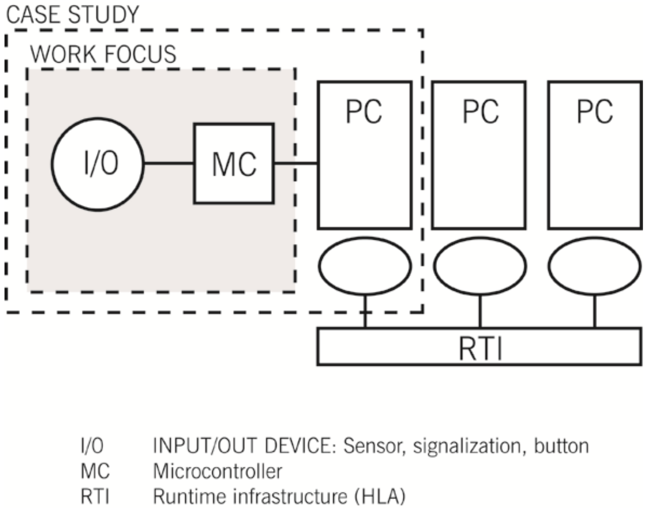

Figure 2: Second alternative of situation

A microcontroller itself is capable of processing various signals from input devices and sensors. The main part of developing is creation of bridge between distributed simulation and microcontroller. The various types of input and output devices are taken into scope in final paper, and in particular to account for their speed and complexity of data processing from those. The text is, rather focused on techniques how to get signals from the microcontroller and awards after-processing of data to possibilities to use them in a distributed interactive simulation.

Comparison of optimality of direct connection (a microcomputer connected to the LAN as an element of distributed simulation), with solution, which is realized by connecting microcomputer to another computer, will be part of the full text of the article. It will also include focus in regards to the advantage of each of those solutions and their measurement.

5. HARDWARE LAYER

For the needs of this publication, hardware layer is understood as a computing hardware serving as a pre-processor or an input-output data processor. Solution to be generally described as the use of a computer is too simplistic, because except of those complying with IBM-PC standard, there are numerous other devices considered to be a computer.
To connect external hardware (input-output device) with the simulator, there exist two basic ways. The first method is taking the input-output device and its encapsulating hardware as one object, while this solution has a certain advantage especially in the event of its engagement in distributed simulation. Alternatively, an ideal solution (namely in the programming point of view) is to separate both devices from each other followed by the formation of independent logic with the option to connect the given device and also its analogues.

5.1. Personal computer

In order to a comparison was good enough and had sufficient informative value personal computer was also included to it. Given that the development of sensors connected to the simulation is usually performed using a PC, it's another reason why incorporate computers.

To simplify the comparison of the computer will perform calculations solely on the CPU (computing on graphics card are not considered) Core2Duo. Fitted with memory in dual-channel RAM and has a size of four gigabytes. For connecting the input output devices it was used only USB port. However, many devices at the hardware level using only the USB emulation through a serial communications port. The reader should therefore always be careful whether (if necessary) device uses USB full capacity.

5.2. Mobile device

Besides standard computer, we can choose to use a mobile device (tablet, mobile phone) as a hardware interlayer. Despite it does not seem quite useful at first sight, it is essential to mention the advantages of mobile devices: mobility and price, low overhead cost, simple programming and especially large number of interfaces. Mobile device have USB ports and often also Wi-Fi, Bluetooth and other technologies. The development may be as easy as that of PC, however, thanks to the portability and price, or due to the implicit network support, solution in certain events appears to be much more effective.

Another advantage, not yet mentioned, which can be used mainly in online situations, is the permanent battery operation. If a device, for example a pre-processor for sensor network data processing, is connected we can collect data even during a power failure because mobile devices have their back up source.

A drawback of this solution is a lower computation output. Despite the price is lower than that of the previous solution, it is far from being ideal.

5.3. Raspberry PI

Raspberry PI (Figure 3) is an interesting solution that, for the price around €39, gives options similar to those of a standard computer or mobile device. Raspberry has standard USB ports (for connecting commercial devices) and ports enabling specific programming options. Feeding is carried out via USB, optionally from a mobile phone Power Bank. Certain drawback of Raspberries is a limited choice of operating system selection: the inherent system is implicitly used. In any case, such hardware platform presents a great potential for the formation of simulator input-output devices. Its small size, low consumption, a relatively high computation output and a wide selection of different ports makes Raspberry a perfect interlink between any (commercial or inherent) hardware device and computer simulation.

Additional Information: After making a comparison test it was released a new market model Raspberry PI, which has all the advantages mentioned in this article, but there has been a significant increase in computing power. Comparison can be found for example on the (Benchoff Brian 2015). But modern equipment for the purposes of this paper was not included for reasons of state of the article.

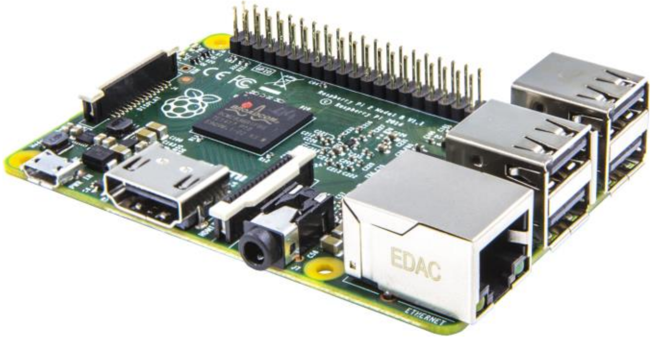

Figure 3 Raspberry PI

5.4. Microcontroller

It is only a few years ago that non-standard devices were connected to a PC via microcontrollers.

Original requirement for programming in specific programming language which is different for various chips (e.g. Assembler), the programming language C became a standard method that opened micro-controllers to the community not specializing in the lowest solution levels.

In any event, this device becomes obsolete and even though there exist classes of tasks, where it still can be used, they are not suitable for applications in modern simulators. You can see (Richardson 2012, Shawn, 2012) for more information.

5.5. Comparison of solutions

Comparison is based on own research in (Brozek, Jakes and Gago, 2014) and a few prestige sources as (Huddleston, 2007; Schön, 2013; Schönborn 2013).

5.5.1. Price

Taking in view sufficiently subsidized simulations such as Hi-tech segment that do not have to consider the price differences within percent units, it will be quite interesting to look at Table 1 which illustrates current prices, and what is more, the price is compared with the computation output.

Computational efficiency of microcontroller is under focused scope.

Table 1 Price comparison

Device	Price [EUR]	Power [GFLOPS]	Power / price [GFLOP / EUR]
PC (2x AMD Opteron 6344, 64 GB RAM, GTX 660)	2 000	10 000	5
Raspberry PI B+ (ARM 900 MHz, 1 GB RAM)	40	1	0,025
Dell Tablet (Intel Atom 1,6 GHz, 2 core, 512 MB RAM)	56	2	0,036

5.5.2. Computation output

However, computation output (Figure 4) may not play its role. Dependence of the whole solution on the computation output relates the way the unit is used. Therefore, if the task is a mere collection of interactive data or providing interactive outputs, all the compared solutions have a sufficient computation output. If we take in consideration the fact that besides a I/O message, a unit may be the main node of a particular simulation, or one of the computation nodes for a distributed simulation, then the absolute value becomes important. Major effect of computation output is monitored in extensive online simulator characterized by a large number of very fast processes.

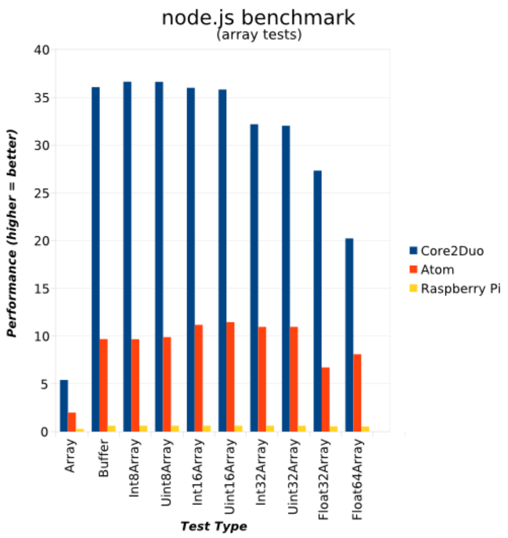

Figure 4 Computation output [Henniger, 2014]

5.5.3. Energy requirements

Comparison of energy requirements is illustrated in Table 2. Microcontroller is a clear winner. On the other hand, when returning to computation output, it is obvious that computation output conversion into an electric energy output of Watt is not as favourable as it seemed (the value is poor), and micro-controller excels only optically. The worst values appear at computers, which is not a surprise.

Table 2 Energy consumption

Device	Run on battery 5AH [hours]	Power [GFLOPS]
PC (2x AMD Opteron 6344, 64 GB RAM, GTX 660)	<1	10 000
Raspberry PI B+ (ARM 900 MHz, 1 GB RAM)	17	1
Dell Tablet (Intel Atom 1,6 GHz, 2 core, 512 MB RAM)	6	2
Micocontroller (AT Mega 16 series)	48+	< 0.001

5.5.4. Results of comparison

Based on several independent measurements and also on the team´s experience, we can demonstrate that Raspberry can be considered an optimal device for the formation of interlinks for input-output device. On the other hand, in terms of a simple solution, especially if mobility is not required, users prefer a more conservative way – connection to a PC.

6. OUR OWN SOLUTIONS

To test different variants, has been created demonstrator simulator with an intuitive rules. Distributed simulation has three key parts: The first part is the software driver for the entire solution, the second part visualizer and archival core. The last, third part of the Federation, is the one, who in various ways connected hardware solution.

6.1. Issue conception

The simulator is intended to reflect operation of the simple logistics system: A test rail circuit. The principle is that, there is a railroad track with two different circuits. Track is used to test the basic handling characteristics of the train. Demonstrator software can test only characteristic as speed, stability. It also can show the possibility of inclination in dependence on the speed or electrical properties, as speed to input electrical power (more specifically, on set of input voltage and supply current).

In the simulator is a sequence of simulation calculation evolution. First is practical management (interactive Federate). The orders are send to second part: i/o federate, which operates physically at model railways and at the same time collect data (eg. A journey time of the train, information on how the error occurred, or report time of crashes the train). A demonstrator is so interesting example of simulation that is not only software, but it is a combination of software and physical model.

6.2. Electric signal / federate simulation

Physical Model Railroad (trains size TT) is controlled by voltage. The basic source of tension is a certified laboratory source, which is connected via other electronics to railway. The electronic varying according to the method of controlling: a computer / mobile devices / rawsberry / Microcontroller with the service circuits.

The most complex development required the involvement of microcontroller. It was difficult especially due needs of very complex wiring, realized by circuits on a non-solder field. Simultaneously, programming was very non trivial due necessity to work with a low level instructions. Sad finding was the fact, that implementation at a low level does not improve performance. Hardware of Microcontroller has been generally insufficient neither for basic data processing in distributed simulation HLA (ie the translation of key data in XML format).

From the development perspective, microcomputers were a very pleasant surprise, thanks to its easy setup. Raspberry PI has hardware pins, which are very easy to address and operate with them. Programming interface for these input / outputs is very easy.

6.3. Other federates

The other two federate types are standard implemented as standard full software solution. They are created by programmatically in Java. Operation (communication and synchronization) is provided with standard library classes of HLA. RTI is used from the Pitch Company (Pitch RTI).

Management federate is implemented a standard interactive simulator. User entry is written to the calendar of events by introduced NextEvent. Events calendar make synchronization with the rest of the federation.

7. DISCUSION

The results of our study reflects the scientific trends in the world. Based on testing it is necessary to declare that the involvement of microcontroller in distributed simulation provides no added value. At the same time, the development of microcontroller is the most challenging of all tested technology. Ten years ago, it might be interesting to use microcontrollers for economic reasons. The current market situation, however, offers a much better solution (raspberry) with similar price. In further discussion and microcontrollers are taken into consideration only marginally.

Chapter expands the comparison of the various technologies about results of our own research and testing.

7.1. Sophistication

An important parameter when considering whether to include a technology into our own solutions is difficulty of implementation (Respectively, difficulty of implementation in relation to the added value for a solution. But since in this paper are examined only solution with the same added value that we can remove this influence and focus only at difficulty of implementation).

Implementation time was examined according to the same solution implementation tasks for all examined systems. The measured values were man hour.

All four solutions that were tested are characterized by distinct complexity implementation. Among the simplest solution are a standard computer programming and programming for Rawsberry PI. Difference between these two technologies is very small. This is due to very good support for high-level programming languages, and a large variety of supported drivers.

The third solution in difficulty is in our comparison of programming for mobile devices (mobile telephony or tablets). Moderate disadvantage is the requirement for a limited number of environments and difficult access to the drivers. Overall, the programming for mobile devices is about 2 times more difficult than the case of computers or microcomputers.

Far more complex (10x implementation demanding than solutions for mobile devices and therefore 20x implementations demanding than for computer and microcomputer) became implementations for microcontrollers. In addition, it was found that there are some problem sets that cannot be solved through Microcontroller because of hardware limitations.

7.2. Mobility

One of the key characteristics in simulators may be mobility solutions. For example: In the simulator real environment, it is necessary to attach some interface directly to the soldier in the simulator is completely inappropriate to connect a standard PC with a weight of several kilograms, and the requirement for permanent power connection.

In the tests were the key requirements for ease of installation, size and method of power.

Although at first sight could be natural that those tests prevails mobile devices, it is not. The real winner was the microcomputer, which (in conjunction with power pack) managed to work much longer than mobile devices. Weight and dimensions should take better character than with mobile device.

In second place could be placed Microcontroller. But, because it is necessary to take into account also ancillary circuits and specific power requirements and extreme demands on the communication scatters, was very high discomfort when trying to build an "island simulator".

Mobile devices have the disadvantage of required imaging unit. Calculations in sleep mode, or turn off the display is very difficult to implement and that are characteristics that strongly supported Discomfort in solving.

Through actual miniaturization of computers with a standard computer is totally unsuitable as a competitive candidate in this discipline.

7.3. Usability and potential

As mentioned in the introduction, the potential for the Microcontroller is currently very small. Old solutions can still deploy, but the development of innovations already lacks effectiveness.

Other technologies are advised mobile technology. Although all tests achieved average results, general this is a solution that lacks the potential (except for their use in distributed interactive simulations, as explained in Brozek, Jakes, Gago (2014).

The computer has a big advantage if it is used in stationary solutions. A great advantage in this case is the number of inputs that reaches many dozens. While potential in processing speed and ease of maintenance is very high.

Microcomputers excel in this type of task, where the requirements are very similar to those on a standard PC, but also require mobility solution of. Likewise, it is win solutions where we get by with limited computational power but is paramount for us economical solution. Then they are an ideal example of emulation, live simulation and input output units simulators.

8. CONCLUSION

A key objective in this work was to test different ways of microcontroller participate in distributed simulation. The second primary objective was to compare different methods of involvement with alternative solutions (computer, mobile device, I microcomputer).

The findings showed that the trends that were still in the last decade obvious are overcome. Using of microcontrollers in complex systems can no longer be recommended. Claims for developers and the extreme need for efficient optimization function in a complex system cannot compensate for the benefits of microcontrollers.

The second set goal (compared with alternative technologies) revealed the extreme potential of microcomputers in distributed simulation. And not just as potential candidates for compute nodes, but also as signal converters - input and output.

The paper describes the development in IT industry. Previously, cheap and efficient, Microcontroller is thanks similar affordability, but far greater user-friendliness with programming is replaced by the microcomputer.

Paper resume is clear: microcontrollers are overcome. Let's concentrate on the microcomputer.

REFERENCES

BENCHOFF, Brian. BENCHMARKING THE RASPBERRY PI 2. Hackaday [online]. 2015 [accessed 2015-07-22]. Available from http://hackaday.com/2015/02/05/benchmarking-the-raspberry-pi-2/

Brozek J., Jakes M., Gago L. Using tablets in distributed simulation. Pardubice, 2014. ISBN 978-889799932-4. Conference Paper. EMSS 2014 Proceedings, University of Bordeaux.

BY TERO KARVINEN, Kimmo Karvinen. Make sensors: a hands-on primer for monitoring the real world with Arduino and Raspberry Pi. 1. Aufl. MakerMedia/KM/Feb: O'Reilly, 2014. ISBN 978-144-9368-104.

Chemical sensors: simulation and modeling. Editor Ghenadii Korotcenkov. New York: Momentum Press, c2012, xxiv, 440 s. Sensor technology series. ISBN 978-1-60650-309-6.

FRADEN, Jacob. Handbook of modern sensors: physics, designs, and applications. 4th ed. New York: Springer, 2010, xv, 663 s. ISBN 978-1-4419-6465-6.

Free software foundation, Inc, 2002, CERTI - Summary [Savannah]. Available from: http://savannah.nongnu.org/projects/certi [accessed 22 April 2014].

FTDI [online]. [accessed 2015-07-22]. Available from: http://www.ftdichip.com/

Fujimoto, Richard M., c2000, Parallel and distributed simulation systems. New York; John Wiley & Sons. ISBN 04-711-8383-0.

Geeknet, Inc., 2005, SourceForge.net: Open HLA - Project Web Hosting - Open Source Software. Available from:http://ohla.sourceforge.net/ [accessed 22 April 2014].

Huddleston C. Intelligent sensor design using the microchip dsPIC. Amsterdam: Elsevier/Newnes, c2007. xix, 283 p. Embedded technology series.

IYENGAR, S, L PRASAD a H MIN. Advances in distributed sensor integration: application and theory. Upper Saddle River, N.J.: Prentice Hall PTR, c1995, xvii, 273 p. ISBN 01-336-0033-5.

KARLSSON, Mikael; NÄSSTRÖM, Fredrik; HAAPALAHTI, Gustav. Building a modular multi-sensor simulation framework using COTS software and HLA. In:Proceedings of the IASTED International Conference. 2009. p. 025.

Knight, Pamela, Corder, Aaron, Liedel Ron, 2001, Evaluation of Run Time Infrastructure (RTI) Implementations. U.S. Army SMDS. Available from: https://www.scs.org/confernc/hsc/hsc02/hsc/papers/hsc017.pdf [accessed 15 July 2014].

Kovač, Pavel, 2012. Reflection: Where are we going tablets?. Available from: http://www.svethardware.cz/uvaha-kam-smerujitablety/34506-3 [accessed 15 July 2014].
Ku, Andrew, 2012. Lenovo's ThinkPad X230T Tablet PC, Tested And Reviewed. Available from: http://www.tomshardware.com/reviews/thinkpadx230t-review-benchmark,3229-2.html [accessed 15 July 2014].
Kuhl, Frederick, Dahmann, Judith, Weatherly Richard, The institute of electrical and electronics engineers, Inc, c2000, Upper Saddle River, NJ; Prentice Hall PTR. ISBN 01-302-2511-8.
Manlig, František, 1999. Computer simulation of discrete events. Available from: http://www2.humusoft.cz/www/archived/pub/witness/9910/manlig.htm [accessed 15 July 2014].
Mocný, Ondřej, 2009. Real-time physics simulation for mobile device. Bachelor thesis. CHARLES UNIVERSITY.
NGOC THANH NGUYEN ... [ET AL.], Ngoc Thanh Nguyen ... [et al.(eds. Agent and multi-agent systems technologies and applications: second KES international symposium, KES-AMSTA 2008 : Inch'ŏn, Korea, March 26-28, 2008 : proceedings. Berlin: Springer, 2008. ISBN 978-354-0785-828.
Pi4J [online]. 2012, 18.4.2015 [cit. 2015-07-22]. Available from: http://pi4j.com/
Pitch technologies ab, 1993, Products – Overview, Available from: http://www.pitch.se/products/products-overview [accessed 22 April 2014].
Rabelo, Luis, Sala-Diakanda, Serge, Pastrana, John, Marin, Mario, Bhide, Sayli, Joledo, Oloruntomi, Bardina, Jorge, 2013. Simulation Modeling of Space Missions Using the High Level Architecture. Available from: http://www.hindawi.com/journals/mse/2013/967483/[accessed 15 July 2014].
RAJIVE, Joshi a Castellote GERARDO-PARDO. A Comparison and Mapping of Data Distribution Service and High-Level Architecture [online]. 2006 [accessed 2015-07-21]. Available from: https://www.rti.com/docs/Comparison-Mapping-DDS-HLA.pdf
Richardson M., Wallace S.P. Getting started with Raspberry Pi. 1st ed. Sebastopol, CA: O'Reilly Media, 2012, xiii, 161 p. Make: projects. ISBN 14-493-4421-6.
RTI Connext DDS Android-Raspberry Pi Demos [online]. MEENA, Ashish. 2014 [accessed 2015-07-21]. Available from: https://community.rti.com/howto/rti-connext-dds-android-raspberry-pi-demos
Schön O., 2013. How powerful is your PC? New 3DMark test measures the ability of PC and tablets and phones. Available from: http://tech.ihned.cz/hry/c1-59252020-3dmark-pro-pc-i-iphone-a-android [accessed 15 July 2014].
Schön, Otakar, 2013. How powerful is your PC? New 3DMark test measures the ability of PC and tablets and phones. Available from: http://tech.ihned.cz/hry/c1-59252020-3dmark-propc-i-iphone-a-android [accessed 15 July 2014].
Schonborn T. SoC Shootout: x86 vs. ARM. In: [online]. 2013 [accessed 2015-03-15]. Available from: http://www.notebookcheck.net/SoC-Shootout-x86-vs-ARM.99496.0.html
Sensors. 2011, vol. 11, issue 12. ISSN 1424-8220. Available from: http://www.mdpi.com/1424-8220/11/4/4385/
The institute of electrical and electronics engineers, Inc, 2010, IEEE1516:2010: IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-Framework and Rules. New York; IEEE. ISBN 978-0-7381-6251-5.
The institute of electrical and electronics engineers, Inc, 2010, IEEE1516:2010: IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-Object Model Template (OMT) Specifications. New York; IEEE. 2010.ISBN 978-0-7381-6249-2.
The institute of electrical and electronics engineers, Inc, 2010, IEEE1516:2010: IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-Federate Interface Specification. New York;IEEE. ISBN 978-0-7381-6247-8.
The simulation interoperability standards organization,. Independent Throughput and Latency Benchmarking for the Evaluation of RTI Implementations, 2001, The Simulation Interoperability Standards Organization. Fall. DOI: SISO-01F-SIW-033.
Using HLA to Transmit Real-Time Sensor Imagery [online]. In: SILBERT, Mark, William SCHIBLER a Gordon CURRAN. 2000 [accessed 2015-07-21]. Available from: https://www.researchgate.net/publication/235185975_Using_HLA_to_Transmit_Real-Time_Sensor_Imagery
Vtmäk, 2014, HLA RTI – Run Time Infrastructure – MÄK RTI. Available from: http://www.mak.com/products/link/mak-rti.html [accessed 22 April 2014].
William J. Serial and Parallel Transmission. THE GALE GROUP INC. Encyclopedia.com [online]. 2002 [accessed 2015-07-22]. Available from: http://www.encyclopedia.com/doc/1G2-3401200276.html