

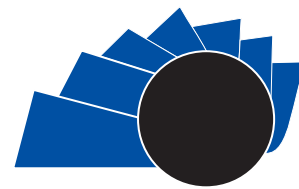


UNIVERSIDAD DISTRITAL  
FRANCISCO JOSÉ DE CALDAS

## Visión Electrónica

### Más que un estado sólido

<https://doi.org/10.14483/issn.2248-4728>



VISIÓN ELECTRONICA

A Research Vision

## Integration of hardware and web artefacts in embedded systems

### *Integración de hardware y artefactos web en sistemas embebidos*

Duván Arley Pardo-Cadena <sup>1</sup>, Julián Rolando Camargo-López <sup>2</sup>,  
Cesar Andrey Perdomo-Charry <sup>3</sup>

#### INFORMACIÓN DEL ARTICULO

Historia del artículo

Enviado: 25/02/2018

Recibido: 17/03/2018

Aceptado: 12/04/2018

#### Keywords:

Embedded System,  
Integration Method,  
Internet of Things,  
Web Artifact,  
Web of Things.

#### Palabras clave:

Sistema integrado,  
Método de integración,  
Internet de las cosas,  
Artefacto web,  
Web of Things.

#### ABSTRACT

This paper is a guide for the realization of integration processes between Web artifacts and the Hardware that supports them in Embedded Systems oriented to the internet of things. This suggested method is based on tests and hardware and software architecture design that contemplates the support of Web transactions taking into account the limitations of an Embedded System. The result of the study is a guide for working groups with a structured systematic and coherent orientation of process of integration between software and hardware, aimed to Embedded Systems that interact on the Web.

#### RESUMEN:

Este documento es una guía para la realización de procesos de integración entre artefactos Web y el Hardware que los soporta en Sistemas Embebidos orientados a Internet de las cosas. Este método sugerido se basa en pruebas y diseño de arquitectura de hardware y software que contempla el soporte de transacciones web teniendo en cuenta las limitaciones de un sistema integrado. El resultado del estudio es una guía para grupos de trabajo con una orientación estructurada y coherente del proceso de integración entre software y hardware, dirigido a sistemas integrados que interactúan en la Web.

<sup>1</sup>BSc. In Electronic Engineering, Universidad Distrital Francisco José de Caldas, Colombia. MSc. In Information and Communication Sciences, Universidad Distrital Francisco José de Caldas, Colombia. Current position: DIAN, Colombia. E-mail: [dpardoc@dian.gov.co](mailto:dpardoc@dian.gov.co) ORCID: <https://orcid.org/0000-0002-0909-0108>

<sup>2</sup>BSc. In Electronic Engineering, Universidad Distrital Francisco José de Caldas, Colombia. MSc. In Information and Communication Sciences, Universidad Distrital Francisco José de Caldas, Colombia. Current position: Professor at Universidad Distrital Francisco José de Caldas, Colombia. E-mail: [jcamargo@udistrital.edu.co](mailto:jcamargo@udistrital.edu.co) ORCID: <https://orcid.org/0000-0003-3483-1884>

<sup>3</sup>BSc. In Electronic Engineering, Universidad Surcolombiana, Colombia. MSc. In Information and Communications, Universidad Distrital Francisco José de Caldas, Colombia. Current position: Professor at Universidad Distrital Francisco José de Caldas, Colombia. E-mail: [cperdomo@udistrital.edu.co](mailto:cperdomo@udistrital.edu.co) ORCID: <https://orcid.org/0000-0001-7310-4618> Cite this article as: D. A. Pardo-Cadena, J. R. Camargo-López and C. A. Perdomo-Charry, "Integration of hardware and web artefacts in embedded systems", *Visión electrónica, algo más que un estado sólido*, vol. 1, no. 2, Special edition, july-december 2018. DOI revista: <https://doi.org/10.14483/issn.2248-4728>

### 1. Introduction

Firstly, embedded systems that have the capacity of connecting to a net (usually Ethernet or Wi-Fi) are generally complex systems that require a permanent use of integration protocol. There is a wide variety of tools that allow controlling and handling integration tasks. In some cases, standard tools like UML MARTE allow to change the integration process. However, it does not have a formal structuration when it comes to coordinate specifically the integration of Software and Hardware. [1-4]

Secondly, it is unusual to find technology capable of fulfilling every single application requirements for the most complex embedded systems. In this case individual elements are selected having into account their interoperability in order to avoiding posterior problems in the integration. In some cases, integrated packages of compatible elements are available in the market, ready to be used in embedded systems from different technologies like integrated Java solutions from Sun, NET Compact Framework from Microsoft, and CORBA from Object Management Group (OMG), etc. [3], [5][8]

#### 1.1. Technologies Integration

There is a wide variety of integration models that have been proved in Software and can be used for the integration in embedded systems having into account parameters of Hardware as if they were components and artifacts of Software. These integration models' objective is to reduce time, expenses, and other elements of the project in order to have a more efficient work. In architectonic design the model or integration protocol must be taken into account; there are examples like the architecture oriented to SOA services or the point-to-point SAS-based architecture. Some other architectures that can be seen from macro to micro (top-down) are the EAI (Enterprise Application Integration) and the Integration Oriented to Services. [9-10]

There are technological integration models that can be used transversally for systems that contain software and hardware. For example, Kerschberg model has an approach in the heterogeneity of knowledge sources and it establishes integration components (layer-based model). See Figure 1. This model has a relationship between potentiated architecture and technologies oriented to support knowledge management. It has a strong technological approach that differentiates actions, presentation, knowledge management and data bases. [11-12]

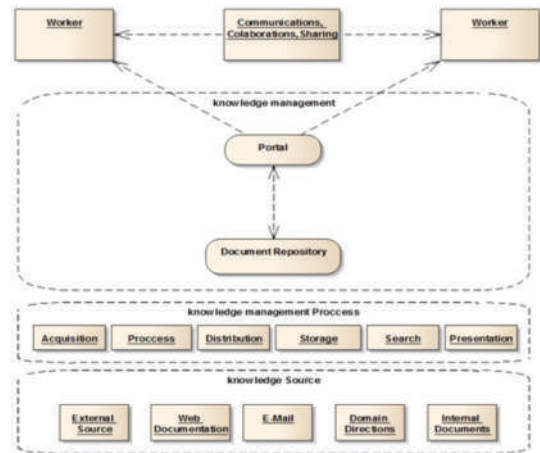


Figure 1. Technology integration model, [13].

#### 1.2. Hardware Architecture and Software Artifacts

Having into account the context of the development, different authors have a perspective of the management of every single stage of the project. When it comes to specific elements of the design of software and hardware, authors like Oliveira mentions that the development of embedded systems does not only consists of the embedded software, but the hardware is also important, see Figure 2. In Design Architecture and Architecture controls, the author makes a description of the architecture of the design and the architecture for elements management in the project. He highlights things like risk factors, critical aspects and expenses changes. [14-15]

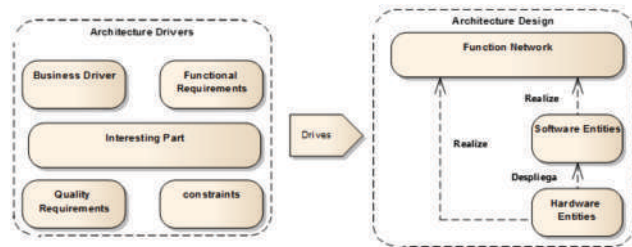


Figure 2. Architecture design and architecture controls, [15].

As a matter of fact, good software development practices (through the utilization of operative systems for small devices: iOS, Android, Java based Firmware, among others) raise different methodologies and organizational structures for execution processes. This evolution shows the importance of having into account the hardware in which apps will operate.

### 1.3. Test Design

This methodology's approach is to build an embedded system that provides web services having into account restrictions, boundaries and the necessities of the client. Three types of final tests for the system must be specified: software-oriented tests, hardware-oriented tests, and full tests that contain the two of them.

- Embedded Software tests are structured around the functionality of the system. These must show every single functional requirement associated with the web service. In addition, non-functional tests must be established having into account service quality, security and stability.
- Hardware tests must contemplate every restrictive aspect from non-functional requirements. Generally, these are more complex (a processing test or a storage tests is harder to evaluate in the hardware). Some of these tests can be seen from tests in the software (if it is operating in the developed platform).
- Full tests are based in hardware-software integration tests. It is possible that software tests are done as in simulations or emulations, depending of the tools that have been selected. Having this into account it is necessary that full tests are made using the developed hardware.

In order to establish a structured way of making the integration process between hardware elements and software artifacts oriented to web connectivity, an approximation to general architecture oriented to fulfilling tests based in functional and non-functional requirements was made. To do this we studied the different modules involved in the hardware that support the necessities of the business. Since tests are oriented to requirements, they can be seen in single tests for each software artifact and its corresponding support from the hardware.

### 2. Integration of Software IoT – Hardware artifacts Method

Having into account the strong interaction between Web artifacts and the supporting Hardware the sustaining for them is done in the operative system or in manufacturer's libraries. The compatibility between the architecture of the processor (CISC, RISC, x86, ARM, among others) and the code executed depends on things like the compiler, codification practices, third party's libraries, recycling of codes, etc.

Having this into account and the main objective of this paper (to formalize a procedure that allows the integration between both development elements) it can be established a parallel structure that allows them to interact with common objectives (in this case, tests).

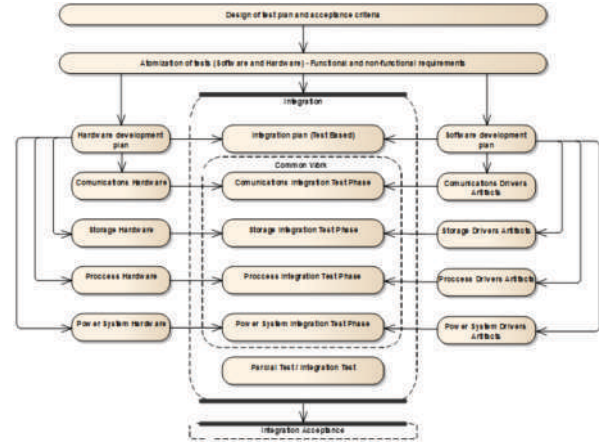


Figure 3. Transversal scheme of proposed integration. Source: own.

The integration process that the authors propose raises an initial point the design of tests and acceptance criteria. It finishes with the acceptance of criteria given in the first stage (see Figure 3 and Figure 4). In these there is a graphic representation of the different stages of the process and its main components.

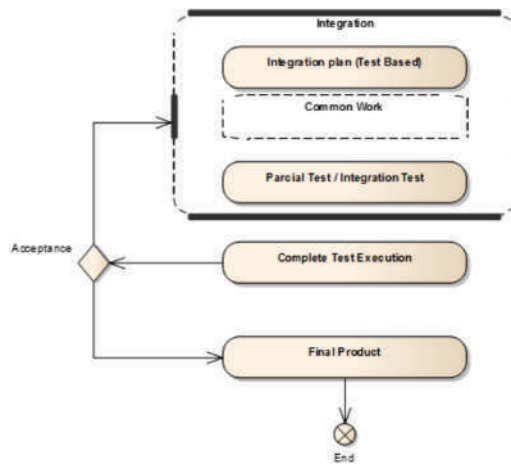
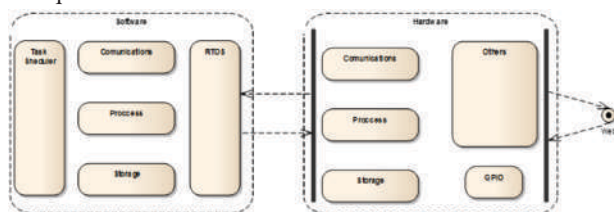


Figure 4. Generalized scheme of Integration and Acceptance. Source: own.

Regardless the tool that is used to coordinate, structure, determine, and keep or execute the diverse integration tasks, these are the basic elements in the integration: According to architecture decisions in the integration of the base Software, it refers to the Operative System

or to the code that operates in the system. It begins with the communication tasks since the aim is to keep the web communication supported by the embedded system. Integration tasks between the hardware that will present the physical interface towards the net and the software that keeps all the computational logic for its correct functioning must be carried out.

Storage tasks must be structured having into account the chosen architecture. This is necessary due to the complexity of each case. Although, there is usually a physical support in the hardware using well known storage technologies and different layers supporting the embedded software. It is possible that the operative system “layer” takes care of the file system and the application layer takes care of handling them. Processing tasks must be studied from both software and hardware. Depending of the developing necessities it is usual that each element has associated requirements. For example, in some cases, the hardware can take care of encryption tasks and the software of the rest of the tasks. Mathematical calculations and others may be associated to elements inside of the stablished architecture of the hardware, simplifying the development level to a software level.



**Figure 5.** General scheme of Architecture S.E. – Web. Source: own.

Service-quality elements definition is structured around non-functional requirements. Depending on its nature, integration area must determine in which way it affects the software and the hardware. For example, there are cases in which the encryption of files can be done by the software, just like they can be done by the hardware.

In general, the approach of the embedded system development is to separate abruptly the layer of the software from the hardware, see Figure 5. When carrying out tests it must be had into account that these layers must coexist regardless the architecture since each element of the software is supported by one in the hardware for its operation. Additionally, the hardware cannot work without the control of the software. Stablished tests must have this relationship into account. For example, for specific tests, the right control of hardware must be evaluated in the software and the

interactional element (buffers or storage spaces in general) must be contemplated in the hardware tests. It must also be homologated by hardware and software for its right operation. In case that the two environments agree in order to fulfill the functionality evaluated by a test there is an integration milestone.

In another instance, it can be analyzed the different transversal aspects like non-desired cases. For example, if the systems lacks electric energy, how can the software and hardware handle this inconvenient? There are cases in which absolute control is given to the software. However, in other cases in which the hardware's nature allows controlling or taking decisions over different eventualities in the system, the focus must be analyzed in the aspect of the architecture in the development.

Each element previously mentioned is usually associated to a group of partial tests that represent the control of the state in the general development. Approving these implies a milestone in the development. Generally, these tests are associated to a tests plan for each session. For example, in communications tests of communication between the system and the net in general can be done: ICMP data and similar.

Moreover, when software and hardware need to be integrated and have finished meaningful tasks integration tests are done. They can be of integration between software and software, software and hardware, or hardware and hardware. Each item defined inside the tests must be part of an individual tests plan for each module. The objective is that all the tests are made documenting the results in order to be taken into account in case of needing adjustment in any of the modules.

In order to finish the integration process and the development and encrypting process in general conceptual tests must be done. In these the tests plan is associated to each of the requirements (functional and non-functional). In theory all of the tests must be approved since these are virtually the partial tests combined with the integration tests.

### 3. Discussions

Taking into account, the test-based approach raised in the development of this article as a method for technologies integration of hardware and software elements web-interactive oriented is profitable not only if web-oriented, but also as a more general development inside developments that contemplate hardware and web artifacts.

It is important to highlight that applying the right group of tests can be seen in complete and clear results. In order to do this some conditions must be fulfilled inside each test:

- Each requirement must be functional and non-functional
- Be as specific as possible
- It must be measurable
- It must be justifiable according its functionality

In the schedule chosen to determine architecture elements (from software or hardware), we must carry out the study of all the elements that are transversal in two levels: software-hardware and interoperability of system artifacts. In this stage it must be clear which tools must be used in the equipment for development of software and hardware. On the top of that, elements like programming language Operative system, real-time operative system, and whether the solution will be based upon Bare Metal. In other cases, we can wait for designs that contemplate the usage of technologies like FPGA or CPLD. In this stage we must have into account elements from software architecture that contemplate basic tools (libraries, platforms, services, among others) In addition to the benefits previously mentioned, the test approach allows us to sustain a base of specific knowledge like a variety of combinations with the interactions of elements like hardware architecture, manufacturer elements, team's contribution, and architecture in general. It also allows to generate knowledge that can be consulted as an experience source in future projects simplifying the decision making for critical elements in the development. On the other hand, this can be consulted in order to structure future tests and as a base for architecture design of projects that are oriented to embedded software, [16-21].

#### 4. Conclusions

- The integration of software artifacts and hardware elements process around a group of tests approach fulfills the needs of the solution that requires it. It guarantees to meet the established requirements in initial stages of the project.
- The organization of the group of tests has a

direct impact in the structure of artifacts and in the architecture of the hardware when a test-passing oriented method is used. The milestones proposed by each test about development identify each of the stages that will be overcome during the integration process.

- The usage of simple tests that encompass the interaction between software and hardware allow the tester to have an account of the processes that have been done correctly or incorrectly inside a project. This information can be part of the knowledge management in the work group, which will allow to avoid delays in the future thanks to previous experiences.

#### Acknowledgements

This result was obtained with the support of the members of the investigation group LASER (Laboratorio de Automatización Sistemas Embebidos y Robótica) as an investigation product of the postgraduate thesis called "Metodología para el desarrollo de Sistemas Embebidos con prestación de servicios Web" for the Universidad Distrital Francisco José de Caldas Postgraduate studies on Sciences of the information and communication.

#### References

- [1] F. Herrera, H. Posadas, P. Peñil, E. Villar, F. Ferrero, R. Valencia and G. Palermo, "The COMPLEX methodology for UML/MARTE Modeling and design space exploration of embedded systems", *Journal of Systems Architecture*, vol. 60, no. 1, 2014, pp. 55-78. <https://doi.org/10.1016/j.sysarc.2013.10.003>
- [2] B. Seli and S. Gérard, "Modeling and analysis of real-time and embedded systems with UML and MARTE", Morgan Kaufmann, 2014. <https://doi.org/10.1016/C2012-0-13536-5>
- [3] OMG, "A UML Profile for MARTE: Modeling and Analysis of Real-Time Embedded Systems", 2008. [Online]. Available at: <https://www.omg.org/omgmarte/Documentations/Specifications/08-06-09.pdf>
- [4] R. Sanchez, "Estado del Arte del Desarrollo de Sistemas Embebidos desde una Perspectiva

- Integrada entre el Hardware y Software”, *Revista colombiana de tecnologías de avanzada*, vol. 2, no. 22, 2013, pp. 98–105. <https://doi.org/10.24054/16927257.v22.n22.2013.416>
- [5] T. Noergaard, “Embedded Systems Architecture: A Comprehensive Guide for Engineers and Programmers”, Newnes, 2005.
- [6] M. Marques, E. Siegert and L. De Brisolara, “Uma Abordagem para Engenharia de Requisitos no Domínio de Software Embarcado”, 2013. [Online]. Available at: [http://wer.inf.puc-rio.br/WERpapers/artigos/artigos\\_WER13/Marques.pdf](http://wer.inf.puc-rio.br/WERpapers/artigos/artigos_WER13/Marques.pdf)
- [7] R. Hamouche and R. Kocik, “Component-based and aspect-oriented methodology and tool for Real-Time Embedded Control Systems Design”, Design, Automation & Test in Europe Conference & Exhibition (DATE), 2012. <https://doi.org/10.1109/DATE.2012.6176586>
- [8] J. F. Broenink, P. D. Vos, Z. Lu and M. Bezemer, “A co-design approach for embedded control software of cyber-physical systems”, 11th System of Systems Engineering Conference (SoSE), 2016. <https://doi.org/10.1109/SYSE.2016.7542927>
- [9] S. Mayer and D. Guinard, “An Extensible Discovery Service for Smart Things”, 2011. [Online]. Available at: <https://www.vs.inf.ethz.ch/publ/papers/mayer-extensiblediscovery-2011.pdf>
- [10] R. Kleinfeld, S. Steglich, L. Radziwonowicz and C. Doukas, “Glue.Things: A Mashup Platform for Wiring the Internet of Things with the Internet of Services”, Proceedings of the 5th International Workshop on Web of Thing, 2014, pp. 16–21. <https://doi.org/10.1145/2684432.2684436>
- [11] A. Pérez, O. Berreteaga, A. Ruiz, A. Urkidi, J. Pérez and I. Coop, “Una Metodología Para El Desarrollo De Hardware Y Software Embebidos En Sistemas Críticos De Seguridad”, *Sistemas, cibernética e informática*, 2006, pp. 70–75.
- [12] L. C. Cordeiro, “TXM: Uma Metodologia de Desenvolvimento de HW / SW Agil para Sistemas Embarcados”, thesis MSc., Universidade Federal do Amazonas, 2007.
- [13] A. Ahmad, M. M. Rathore and A. Paul, “Integration of Capillary Devices in the Smart Society based on Web of Things”, Proceedings of the 3rd International Conference on Human-Agent Interaction, 2015, pp. 269–272.
- [14] A. G. da Silva and R. de Oliveira, “Desenvolvimento de um sistema embarcado em dispositivos móveis para atender às especificações da ISSO 14001”, *Unoesc & Ciência - ACET*, 2014, pp. 49–56.
- [15] O. Trindade, R. T. Vaccare, L. de Oliveira and K. R. Castelo, “Uma Metodologia Para Desenvolvimento De Sistemas Embarcados Críticos Com Vistas a Certificação”, An. do IX Simpósio Automação Intel. - IX SBAI, 2009, pp. 1–6.
- [16] U. Eklund and J. Bosch, “Architecture for embedded open software ecosystems”, *Journal of Systems and Software*, vol. 92, 2014, pp. 128–142. <https://doi.org/10.1016/j.jss.2014.01.009>
- [17] D. Guinard, V. Trifa, F. Mattern and E. Wilde, “From the Internet of Things to the Web of Things: Resource Oriented Architecture and Best Practices”, *Architecting the Internet of Things*, 2011, pp. 97–129. [https://doi.org/10.1007/978-3-642-19157-2\\_5](https://doi.org/10.1007/978-3-642-19157-2_5)
- [18] M. A. Vega-Rodríguez, “Methodologies and tools for the design space exploration of embedded systems”, *Journal of Systems Architecture*, vol. 60, no. 1, 2014, pp. 53–54. <https://doi.org/10.1016/j.sysarc.2013.12.001>
- [19] A. Shen, M. Kuzlu, M. Pipattanasomporn, S. Rahman and L. Chen, “A Performance Testing Method for Embedded Software Platforms”, IEEE International Conference on Cyber Technology in Automation, Control, and

- Intelligent Systems (CYBER), 2016, pp. 135 – 140 .  
<https://doi.org/10.1109/CYBER.2016.7574810>
- [20] M. Bamakhrama, J. T. Zhai, H. Nikolov and T. Stefanov, "A methodology for automated design of hard-real-time embedded streaming systems", 2012. [Online]. Available at:  
<https://pdfs.semanticscholar.org/83ab/9a4ec6207ca1ac702da709fc5645f05ce854.pdf>
- [21] L. Kong, B. Wang, S. Zhou and Y. Yang, "A Novel Approach for Coverage Probing of Embedded Software in SoPC", International Conference on Dependable Systems and Their Applications (DSA), 2017, pp. 155-155.  
<https://doi.org/10.1109/DSA.2017.33>