# EMBEDDED SYSTEMS: DEFINITIONS - TAXONOMIES - FIELD

## PROF. DR. MARTIN TIMMERMAN

## PART I: DEFINITIONS - TAXONOMIES



Martin Timmerman is also professor at the Royal Military Academy Brussels and at the Vrije Universiteit Brussels.

m.timmerman@dedicated-systems.com; martin.timmerman@rma.ac.be; martin.timmerman@vub.ac.be

# 1 Preface

There are lots of conceptions, even more misconceptions and certain confusions about what embedded systems are and the characteristics they have. As expected, this has been confirmed during this project.

The initial version of this document has been produced beginning 2007 under the contract 150524-2006F1ED BE from the EC for a project named “Embedded Systems: Definitions, Taxonomies and Field”. This document covers the definitions and the taxonomies. A second document covers the “field”.

The objective of the documents is to establish a solid conceptual background in this field.

The content of the first document is:

- An accurate detailed definition, description and characterisation of Embedded Systems (EmS) and its borders from both technological and functional point of view, bearing in mind the need to revise and re-evaluate current ICT indicators and statistics.

The content of the second document is:

- A picture of classifications and categorizations of the EmS’ landscape relevant for their measurement and evaluation within the i2010 conceptual and policy framework

To better understand the actual and future situation of *embedded systems*, we will start with a short overview of embedded systems history.

We will then discuss the definitions of embedded systems and its technical and functional characteristics one can find in literature.

EmS are finally subdivided in 3 categories depending on the most important driver:

- cost
- safety critical
- high availability

Today, research is oriented towards these main areas and we are convinced this is also an excellent criterion for subdivision.

The initial versions of these documents are a result of a discussion between the author and IPTS representatives, the discussion during a meeting organised in February 2007 with key representatives of the EmS world and of a survey conducted by the author. The documents are in constant evolution taking into account changing technologies and markets and comments and suggestions from readers.

Enjoy the reading and don’t hesitate to contact me with remarks and suggestions.

Martin

The author Martin Timmerman can be contacted at Dedicated Systems, Bergensesteenweg 421 B12,
B 1600 Sint-Pieters-Leeuw, Belgium, m.timmerman@dedicated-systems.com

# 2 History of embedded systems.

## 2.1 A historical definition

***An embedded system is a subsystem of a larger system which is capable of controlling the larger system through measurement and control.***

However, a lot of systems considered as embedded systems today are not anymore covered by this definition as we will see.

The next paragraphs are a compilation of findings on the history of Embedded Systems (EmS) in different places on the Internet and contributions of the author. It will show the elements contributing to the evolution of EmS.

## 2.2 Early embedded systems examples

### 2.2.1 Before electronics: Jacquard's loom

Historically (mechanical) EmS were first used for helping men controlling a system. This was for replacing repetitive jobs, or for creating a mechanism that could do the job faster. A typical historical example was a contribution to computer methods from the weaving industry. Indeed, in 1801 Joseph Marie Jacquard developed an attachment for weaving looms that used punched cards to "program" a loom to a specific pattern. Although no electronics are involved in this machine, the control part of the loom can certainly be considered an EmS. In 1801, the machine was human-powered. Later it was motorized, and current versions replace the punch cards with electronic computer control showing the classical evolution over the last 200 years of a lot of machinery.

### 2.2.2 The OTIS elevator story

Another typical example is an elevator is a typical example. [OTIS]

From ancient times through the middle Ages, and into the 13th century, man, animal, wind and water power was the driving force behind hoisting devices.

By 1850 steam and hydraulic elevators had been introduced and in 1852 the world's first safety elevator was introduces by Elisha Graves Otis called the "parachute". The first passenger elevator was installed by Otis in New York in 1857. By 1873 over 2,000 OTIS elevators were in use in office buildings, hotels and department stores across America, and five years later the first OTIS hydraulic passenger elevator was installed. The Era of the Skyscraper followed.... and in 1889 OTIS revealed the first successful direct-connected geared electric elevator machines. In 1898 overseas business had added to the company's growth, and Otis Brothers merged with 14 other elevator entities to form the Otis Elevator Company.

In 1903: the gearless traction electric elevator was introduced. To control the elevators functions, primitive EmS were used which were totally based on cabled relay logic complemented by the intelligence of an operator who was needed to make the elevator work.

The OTIS "Autronic Systems", a more sophisticated EmS, was introduced in 1948 making it possible to use the elevator without operators. The intelligence of the operator was shifted towards the control systems.

Otis unveils Elevonic 101, the first completely microprocessor-based elevator control system in 1979. After the introduction of the microprocessor, elevator control functions got much more sophisticated and modern embedded systems are used today to control the elevators. Each elevator in a building or group of buildings may also be networked via Intranet or even Internet in an overall management, monitoring and maintenance system approach.

### 2.2.3 History of telephone switching systems

The telephone switching system started as a manual switching system where human intelligence through operators was used to make it happen. Rapidly electro-mechanical switching replaced the operators' interventions. The first electronic switching central office was the 1_ESS in 1965. Phones at that time used analog technology. The electronic switching systems itself used cabled logic functions to get the switching done.

The first digital switching systems came on the market in 1980. A typical example is "system 12" from Alcatel. Massive amounts of simple digital logic functions are replaced by software control. Microprocessors are used in large amounts. Due to the flexibility of the software control, a lot new services are invented. However, this creates the problem of how to quickly adapt large amounts of software used in this EmS. Software starts to play a major role. After the introduction of the Internet, a switching system became part of it and again new services are now supported by these large EmS. Telecom operators had to redefine their business.

### 2.2.4 Electronic computer technology used in space

The Autonetics D-17 is recognized today as the first mass-produced embedded system guidance computer for the Minuteman missile, released in 1961. It was built from discrete transistor logic and had a hard disk for main memory. When the Minuteman II went into production in 1966, the D-17 was replaced with a new computer that was the first high-volume use of integrated circuits. This program alone reduced prices on quad NAND gate ICs from $1000/each to $3/each, permitting their use in commercial products.

The crucial design features of the Minuteman computer were that its guidance algorithm could be *reprogrammed* later in the program, to make the missile more accurate. The computer could also test the missile, saving cable and connector weight.

In 1964, the Apollo Guidance Computer was also considered a first recognizably modern embedded system. It was developed by Charles Stark Draper at the MIT Instrumentation Laboratory. Each flight to the moon had two. They ran the inertial guidance systems of both the command module and LEM. At the project's inception, the Apollo guidance computer was considered the riskiest item in the Apollo project. The use of the then new monolithic integrated circuits, to reduce the size and weight, increased this risk.

### 2.2.5 Factory automation

In 1960 Digital Equipment produced the PDP family of computers which were intended to be used in factory automation. The PDP11 was most popular and the QBus was introduced to permit flexibility on the I/O control side.

One of the first Real-Time Operating Systems (RTOS) was introduced: RT-11.

Digital Equipment has in the mean time been absorbed by other companies; however, both the hardware and software are still in use today. Mentec Inc for example is continuing the support of the RTOS. Some EmS have indeed a long lasting life of several decades.

## 2.3 Help from technology

### 2.3.1 Introduction

The last 3 decades have witnessed a remarkable evolution of EmS electronic components to be used in EmS. This chapter gives an overview of these technology evolution.

### 2.3.2 Special purpose systems need special purpose processors

Special purpose processors and techniques were developed to support the ever increasing complexity of EmS. EmS are bridging the gap between the “analog” external world and the digital processing world. Therefore early attempts were made to “group” these 2 technologies in a single chip or package.

It all started in 1978 by Intel introducing in 2920 the first "analogue signal processor" and in 1979 by AMI with the S2811. Bell Labs introduced the first single chip **Digital Signal Processor (DSP):** the Mac 4 Microprocessor. In 1980, the first stand-alone, complete DSP -- the NEC µPD7720 and AT&T DSP1 were presented at the IEEE International Solid-State Circuits Conference '80.

Major players are Texas Instrument and Analog Devices. The first generation DSP produced by Texas Instruments (TI), the TMS32010 presented in 1983, proved to be the first great success. Since, today signal processors yield much greater performance.

Of course, not all DSP provide the same speed and many-many kind of signal processors exist, each one of them being better suited for a specific task, ranging in price from about 1.50 to 300 EUR.

Most DSP use fixed-point arithmetic, because in real world signal processing, the additional range provided by floating point is not needed, and there is a large speed benefit; however, floating point DSP are common for scientific and other applications where additional range or precision may be required. Today general purpose CPU's have ideas and influences from digital signal processors and both worlds are melting together.

Generally, DSP are dedicated integrated circuits, however DSP functionality can also be realized using Field Programmable Gate Array (FPGA) chips. The historical roots of FPGA are in complex programmable logic devices (CPLD) of the early to mid 1980s. CPLD and FPGA include a relatively large number of programmable logic elements. CPLD logic gate densities range from the equivalent of several thousand to tens of thousands of logic gates, while FPGA typically range from tens of thousands to several million.

Today’s FPGA support full or partial in-system reconfiguration. This allows their designs to be changed "on the fly" either for system upgrades or for dynamic reconfiguration as a normal part of system operation. Some FPGA have the capability of partial re-configuration that lets one portion of the device be re-programmed while other portions continue running. These characteristics tend to be used in the EmS to come.

### 2.3.3 Need for a modular approach: board based systems

Due to the wide number of fields (see document 2) where EmS are used, each EmS is different in nature especially when it comes to interfacing through the I/O structure. In the 60ties and early 70ties, most EmS were build from dedicated hardware. However as chip integration was very low, systems were then composed of different electronic boards connected together in a proprietary cabled way. This induced a serious cost, reliability was low and time to market for such systems was long.

An interesting next step was the use of Bus based systems which introduced some interconnection standardisation. Initially, each manufacturer invented his own bus based interconnect. As this interconnect is realised through a backplane, these are then called a backplane bus.

A backplane is a printed circuit board that connects several connectors in parallel to each other, so that each pin of each connector is linked to the same relative pin of all the other connectors. It is used as a backbone to connect several printed circuit board cards together. A PC motherboards integrates in most cases an internal backplane for expansion cards. The PCI bus is an example recently replaced by PCI-Express technology.

As mentioned before, Digital introduced the QBus as his backplane bus for the PDP11 machines. This permitted to other vendors than Digital to produce QBus based I/O boards. Digital could then concentrate on the processor part of the machine, and other companies on the I/O boards. Most of the busses at that time were a prolongation of the processor bus and were therefore specific (and different) for each family of processors.

However the idea came that much more manufacturers should be capable of contributing to an arsenal of available electronic boards both processor and I/O boards. Also, the bus interconnect should be independent of the processor in use. As a result, a major contribution came in the beginning of the 80ties with the VMEbus which was a European version of the VERSAbus used by Motorola for the 68000 processor platforms. Initially the VMEbus was used as a single processor bus, of course for the 68000 processor, but rapidly some companies provided processor boards with other processors like the Intel x86-family, the SPARC processors from SUN, and a series of DSP processors. However, from the very beginning the VMEbus was designed to be used as a multiprocessor bus and in that way could be considered as a parallel networking bus. Although it took some time for people to understand this, and for the market to have the need for multiprocessor systems, in the 90ties, the VMEbus got intensively used in such multiprocessor architectures. This characteristic and the fact a lot of extensions to the VMEbus have been adopted through VITA (VMEbus International Trade Association – http://www.vita.org ) are the major reasons for the VMEbus to still be in use today.

PCI bus technology was introduced for the PC world in the beginning of the 90ties as an integrated motherboard backplane. Very rapidly there was an interest in using this technology in board based EmS. As a consequence the CPCI bus (Compact PCI) became a competitor for the VMEbus. Also, the CPCI bus is constantly enhanced and extended through PICMG (the PCI Industrial Computer Manufacturers Group) http://www.picmg.org ). Both have been enhanced over the years and are still in use today.

An important evolution in technology was the exponential increase of the number of transistors one could integrate on a single chip. As a consequence, on one electronic board (with VMEbus or CPCI size), one could put much more complex circuits and single board computers (SBC) became possible. However, as EmS are characterised by the different kind of (classical and special) I/O functions, mezzanine board technology came along. Mezzanines boards are smaller electronic boards which can be plugged onto a SBC. Different standards exists like M-Modules, IP (Industry Pack), PMC (PCI Mezzanine Card), and more recently AdvancedMC (Advanced Mezzanine Card) making it possible to smaller manufacturers to specialise in I/O functions, where larger manufacturers specialises in the SBC. Other solutions using small stackable boards are for example based on the PC-104.

These board based technologies are still largely used today in low-quantity high-complex EmS. Examples will be given in the field section of this document. As you will see, the variety of application fields is high and therefore these boards come in different versions: wide temperature ranges, ruggedized versions etc. Also the chassis used to accommodate these boards, have been sophisticated over the years. The biggest problem here is the cooling and therefore one sees different kinds of cooling (by conduction, convection, by air and by water).

It should be notices that power consumption introducing the cooling issue is also an important issue for these large and complex systems. One should therefore understand that using techniques to reducing power consumption is not limited to mobile systems.

An overview of different bus technologies can be found at

http://www.interfacebus.com/Interface_BackPlane_Buses.html

### 2.3.4 High integration and packaging for high volume EmS

For large quantities and compact (portable) systems, special techniques needed to be developed. New packaging combining display with analogue and digital components were introduced. Also the higher densities used in chip fabrication permitted for **System-on-a-chip or system on chip (SoC or SOC)** approaches. The idea is indeed to integrate all components of a computer or other electronic system into a single chip. As a result, a small system (like a cellphone for example) may contain digital, analog, mixed-signal, and often radio-frequency functions, important in our wireless society – all on one chip or in one package. Indeed, if it is not feasible to construct a SoC for a particular application, an alternative is a **System in Package (SiP)** comprising y number of chips in a single package. SoC is believed to be more cost effective since it increases the yield of the fabrication and also its packaging is less complicated compared to a SiP.

## 2.4 Towards networked EmS

### 2.4.1 Introduction

Another track in this evolution is the emergence of distributed EmS frequently termed as networked EmS indicating the importance of the networking infrastructure and communication protocol. Different technologies are used to create networks of EmS of to network components in order to have a distributed EmS.

### 2.4.2 Parallel bus based multiprocessor approach

As seen in paragraph 2.3.3, a parallel bus like the VMEbus or cPCI bus can be used in a multiprocessor approach. In fact, these were the first networks of computer components teaming together to produce an

EmS. However there is a practical limit to the bandwidth and the distance between components one can achieve with such parallel interconnect approaches.

### 2.4.3 Serial backplane bus based system approach

In the mean time, in business computing, serial networks like Ethernet were used to interconnect PC and servers. Switches replaced rapidly the hubs for networking interconnects. The telecom industry drives the use of this switching technology to be used in the board based systems and switching backplanes have been developed realising high speed serial interconnects between different blades in a chassis. A nice example is the AdvancedTCA solution provided through PICMG coming along with MicroTCA and AdvancedMC, the mezzanine boards to be used in this environment.

Blades, being a SBC including a disk sitting on such a switched backplane, are now widely used as an alternative to 19inch racks full of 1U height servers, especially in the telecom industry.

### 2.4.4 Cabled networking

Of course not all applications permit the use of a backplane approach. In a car for example, different subsystems need to be interconnected with cables or optical fibers in order to be capable of collaborative working. In a “brake by wire system”, the different wheels have each an embedded subsystem working together through a (real-time) network in order to brake the car correctly.

### 2.4.5 Wireless networking

The wired technologies are already well used today and a lot of special purpose industrial network technologies have been developed. However, in some applications, wireless technologies are showing up to be even more important and all kinds of radio technologies are starting to be integrated in EmS.

## 2.5 An overview as conclusion

Today the vast majority of computing systems are embedded systems using the above technologies. Factors of 40 to 1 compared to geneneral purpose computing are shown in recent statistics. (see http://cordis.europa.eu/ist/embedded/hardware.htm )

The 60ties were characterized by dedicated cabled systems. These were very expensive, missed reliability and had a time to market which was very long.

The 70ties were characterized by the introduction of the microprocessor. Time to market was shortened by using standard boards based on one or another parallel bus. Better reliability was also achieved.

Processor performance being limited, the 80ties have been characterized by the use of multiprocessor approaches in order to achieve more complex systems needing more computer power.

In the 90ties, individual processors got sufficient performance and most applications came back to the use of one processor. Individual portable devices became possible and this opened the consumer market. New techniques of packaging a variety of chips helped reducing production costs and helped introducing the use of EmS (like cell phones and MP3 players) in our daily lives.

The actual decade is characterized by an evolution on 2 tracks: the smaller (portable) networked devices using wireless technology and the larger complex systems being backplane based. A typical marriage of these 2 kinds of approaches can found in the telecom industry.

We observe also booming multimedia business. Take a car as a typical example. Today business is about putting multimedia or similar devices into the car.

The next decade however will be different in that EmS will be more used to “drive” the car. This means that safety and security issues used in large volume – low cost devices will become major challenges. Also during the next decade emphasis will have to be put on reducing power consumption. Science shows that a multiprocessor system is using less power than a single processor one having the same features. Therefore there is no way back to single processor systems.

All these combined aspects will put serious challenges on research and development and therefore the major investment herein from the European Commission is more than justified.

# 3 Looking for a definition

We will first start by collecting some definitions one can find in very recent papers, books published and organisations promoting R&D activities in this field.

A project to conduct studies to develop valuation factors for non-production computers has commenced in for the STATE BOARD OF EQUALIZATION, PROPERTY AND SPECIAL TAXES DEPARTMENT in the state of California, USA. A definition of what they call **"non-production computers"** was needed for that project. This definition is very interesting in the framework of our study. It is in fact the opposite of what we are looking at. The definition tries to make the boundary between general purpose systems and special purpose systems.

- [CALI2006] *Non-production computers consist of: (1) general purpose computers; (2) general purpose computer peripherals; and (3) local area network (LAN) devices. General purpose computers contain a central processing unit and memory (be it volatile, fixed, on chips, on a disk, or a diskette), and run a stored program (software). General purpose computers can be programmed to do different kinds of tasks, rather than special purpose computers that are limited by design to a specific task. General purpose computers consist of mainframes, servers and microcomputers (desktops and laptops).* ***General purpose computer peripherals*** *consist of the auxiliary equipment which is designed to be placed under the control of a general purpose computer. General purpose computer peripherals include equipment such as monitors, keyboards, mouses, docking stations, printers, scanners, disk drives, tape drives, modems, wireless cards and web cameras. LAN devices are used to connect two or more general purpose computers, to store data and to facilitate data traffic in a network. LANs are usually contained in a single building (but equipment which is part of a LAN is not excluded merely because it is also part of a wide area network). LAN devices include equipment such as routers, computer network switches, hubs, virus protection equipment, and storage devices. Non-production computers do not include telecommunication equipment or lines (wire, fiber or other) used to connect LANs, computers embedded in machinery, and equipment or computers specifically designed for use in any other application directly related to manufacturing*.

They include computer peripherals and LAN devices in their definition of "non-production computers". It is immediately clear from the reference if "non-productive computers" mean "non-automation-systems" or something else.

We will see later that computer peripherals are in most case embedded devices.

- [BOSI2005] refers to EmS as ***electronic programmable*** *sub-systems that are generally* ***an integral part of a larger heterogeneous system****. Embedded Systems play an increasingly important role in the* ***added value of advanced products****….*

**Electronic programmable**: this is an important characteristic of an embedded system. However, it should be more explicit and one should talk about both software and hardware re-programmability.

**An integral part of a larger heterogeneous system:** this was probably true until low cost consumer products like PDA, iPod and cell phones were build. Indeed an embedded device is not necessarily a subsystem and so the definition does not cover all embedded systems or devices. A patient (medical)

http://www.dedicated-systems.com
Email: m.timmerman@dedicated-systems.com

monitoring system for example is widely accepted as being an EmS, however it is not really part of a larger system, except if you consider the larger system the hospital where it is operated in.

**Added value of advanced products:** the added value produced through the electronics and the software is an important issue for EmS. One should think about what the system is capable of doing without this hard- and software.

- [IEEE1992] defines embedded systems as An Embedded Computer System: *A computer system that **is part of a larger system** and performs some of the requirements of that system; for example, a computer system used in an aircraft or rapid transit system.*

In this definition, the "part of a larger system" excludes cell phones, PDA and iPod like devices except if you consider the "larger system" as the system they are connecting to in the case of a cell phone, and the connection to another device for a PDA and iPod in order to load or unload files (agenda, music, etc.).

Although the Wikipedia quality standards are not met for the following definition, we include it here:

- *An embedded system is a **special-purpose** system in which the computer is completely **encapsulated** by the device it controls. Unlike a general-purpose computer, such as a personal computer, an embedded system performs one or a few **pre-defined tasks**, usually with very **specific requirements**. Since the system is **dedicated to specific tasks**, design engineers can **optimize** it, **reducing the size and cost** of the product. Embedded systems are often mass-produced, so the cost savings may be multiplied by millions of items.*
  *Handheld computers or PDAs are generally considered embedded devices because of the nature of their hardware design, even though they are more expandable in software terms. This line of definition continues to blur as devices expand.*
  *Physically, embedded systems range from portable devices such as MP3 players, to large stationary installations like traffic lights or factory controllers.*

**Special –purpose:** is an important word to make the difference with general purpose computing.

**Encapsulated by the device it controls** limits the definition to devices that are really embedded and that have a controller function. Again a cell-phone has a communication function rather than a controlling function. A few sentences later, they state themselves the limitation of their definition.

**Pre-defined tasks** is important in the definition. Due to this "pre-definition" of tasks, the system is "**dedicated to specific tasks**", it acts in the framework of a predefined purpose. That's the reason why we personally prefer the use of the word **dedicated systems** instead of **embedded systems**. The word "dedicated systems" is not yet widely spread, but different companies and organisations are already using it instead of embedded systems especially in the US.

Another word used for cell phones, PDA, set top boxes etc.. in the US is a **smart device**.

ITEA (http://www.itea-office.org/ ), is a strategic pan-European programme for advanced pre-competitive R&D in software for Software-intensive Systems and Services. Part of Software-intensive Systems is embedded systems and in this way, ITEA deals with them. They use this definition from Wikipedia.

- Following [QING2003] *Embedded Systems are computing systems with tightly coupled hardware and software integration that are designed to perform a dedicated function. The word embedded* *reflects the fact that these systems are usually an integral part of a larger system, known as the **embedding system**. Multiple embedded systems can coexist in an embedding system.*

He adds in the discussion following his definition where he gives even counter examples of this definition:

*A single comprehensive definition does not exist. Therefore, we need to focus on the characteristics of embedded systems from many different perspectives to gain a real understanding of what embedded systems are and what makes embedded systems special*.

- Following [ZURA2006] *A networked embedded system is a collection of spatially and functionally distributed embedded nodes interconnected by means of wireline or wireless communication infrastructure and protocols, **interacting with the environment (via sensor/actuator elements) and each other**, and, possibly a master node performing some control and coordination functions, to coordinate computing and communications in order to achieve certain goal(s).*
  *They appear in a variety of application domains such as, automotive, train, aircraft, office building, industrial monitoring and control and environmental monitoring, and, in future, control as well.*

This definition is very complete and correct for networked embedded systems. However, not all systems are networked (yet).

The wording **interacting** is interesting here. There is interaction with the environment but also with each other. Some elements today are just interacting with each other and not with the environment. I would have preferred **and/or each other** instead of **and each other.**

ARTEMIS (http://www.artemis-office.org ) a program for EmS R&D in EUROPE uses the following definition:

*The term "Embedded Systems" describes **electronic products, equipment or more complex systems,** where the embedded computing devices are **not visible from the outside** and are generally **inaccessible by the user**.*

It is indeed already clear from all these definitions that **electronics** are important. The computing devices are indeed not visible from the outside world except if you use a trendy transparent case. The inaccessibility by the users has probably to do with the fact the user cannot change the behaviour of the computing devices. However, more and more systems permit for downloading extra functionality and some people even go that far that they say that the embedded system could be just a downloadable software component to be run on a multitude of (standard) platforms.





http://www.dedicated-systems.com
Email: m.timmerman@dedicated-systems.com

# 4 Definitions

## 4.1 Basic Definition

Based on the above definitions and their discussion, we propose the following one:

**An Embedded system (EmS) is an electronic system with dedicated functionality build into its hardware and software. The hardware is microprocessor based and uses some memory to keep the software and data and provides an interface to the world or system it is part of.**

**An EmS may or may not be networked and has the following attributes in some degree: real-time, safety, security and dependability. In addition, it can be characterised by networking capabilities, complexity, life cycle length, system quality and environmental factors such like cost, weight, size and power consumption.**

**A human interface ((touch) display, (limited) keyboard, and others) may be present to permit the use of the device via human interaction, but in other cases there is no such (direct) interaction and in that case, no human interface.**

**In most cases, it is a part of a larger heterogeneous system where it plays a computing, measuring, controlling and or monitoring role.**

**They differ from general purpose systems because (for cost and size reasons) they mostly have less processing capability, use limited memory resources and are much more power aware.**

**Some examples are: an aircraft flight control system, car cruise control, traffic light control, factory automation systems, car entertainment system, cell Phones, iPod, tablets and kiosks.**

We feel that this definition is more exact than the others while keeping the capacity to adapt to new or very specific forms of EmS. We tried to maximise the probability to cover all kinds of relevant EmS, whereas the risk to pre-exclude EmS by developing a too narrow description appears minimised.

## 4.2 Definitions refined: dedicated functionality

All EmS have dedicated functionality. It means that the system has been designed for a specific purpose and pre-defined tasks. This system functionality is predefined in the hardware and software.

This functionality can be modified or adjusted through software changes. In most cases, uploading new versions of the software today is for bug fixing reasons. However, it is possible to add functionality to the devices via a software upgrade as long as the hardware doesn't need a modification, or the available memory space is large enough to accommodate the changes. Techniques are now already available to add functionalities on-the-fly. Some people consider these downloadable software components as software EmS without knowing on what hardware it will run in the end.

On the other hand, new hardware technology has been introduced such as FPGA for example. This allows to modify (part of) the hardware functionality.

For time being these modifications are limited and can only be done when the device is not performing its normal tasks. However in the future, this will change and devices will be reprogrammed continuing to perform (part or all) of its functions.

In exceptional cases where reaction times are extremely small, microprocessor technology cannot always deal with the timing constraints and the microprocessor technology is then replaced by hard cabled electronic logic devices. In the future one might also expect the introduction of nanotechnology in the EmS components.

**What are the dedicated functions**? There is an enormous variety of EmS. Some may be extremely small such as intelligent sensors; others are vast systems such as aircraft control systems and vehicle simulators. As a consequence the functions performed by these different systems may vary a lot but can be put in the following categories:

- Computation (giving the intelligence to the overall system or make it “think”)

- Measurement via sensors

- Control of the environment via actuators after decisions made during the computation

- Communication (including data, music, video etc..)

- Human interface (via a display and some buttons or a (limited) keyboard or touch screen or an audio interface)

Not all these functions are necessary. A robot will not always have a human interface. A medical monitoring device will not always control the patient, but limit the output to some display and alarming function.

## 4.3 Attributes of EmS

As there is no real consensus about what Embedded Systems are, different wordings are used for characterizing EmS in different publications. In this chapter we will define these characteristics. As you will see some are complementary and others are overlapping.

All EmS will have these characteristics in some degree going from 0 to 100% depending on the type of application. As a consequence the development methodology of these systems with different combinations and degrees of these characteristics might seriously differ.

### 4.3.1 Real-Time

A lot of embedded systems need to be **real-time systems.**

**Real-time computing is computing where the system correctness depends not only on the correct logical result of the computation but also on the result delivery time called *deadline.***

It means that **the system has a predictable behaviour for some or all of its features or functions.**

The timing constraints may range from seconds to milliseconds and a real-time system should not be confused with a "fast" system. As this characteristic is very important, it will be further developed below in 4.5.

### 4.3.2 Safe

A **safe EmS does not harm the user or in other terms avoids physical or economic damage to persons or property.**

This is straightforward for medical devices, but the characteristic is also valid for an aircraft, a copy machine, a cell phone, elevator control etc.. in different degree.

Safety comes with a cost and one will find systems with very limited safety degree going up to the maximum possible called safety critical, where human lives are at stake if the system fails.

Safety aspects are mostly covered by one or another regulation.

It is now accepted that it is technological impossible to make systems which are 100% safe and safety is now more defined in terms of risk factors and maximum allowed damage.

This notion should not be intermixed with the following one. In English there are 2 different words: safe and secure, however in other languages, there is mostly only one word for both and this introduces serious problems of comprehension.

### 4.3.3 Secure

A secure system is one where **only intended use of the system will be permitted.** This also means **avoiding un-permitted access or modification.**

A first problem might be the definition of the "intended use" boundaries. Indeed, there may be contingencies arising that designers have not considered where legitimate use is not "intended". On the other hand, merely defining in terms of malevolent intentions misses out some kinds of unauthorised use of the system that may be potentially damaging. As a consequence, there are quite different approaches to defining and implementing security.

Security comes at a cost. In today's practice, it is therefore limited or nonexistent in standalone systems with un-modifiable hardware and software (in ROM). However, for systems permitting for uploading new software, re-programming of hardware and for systems that are interconnected, the security aspect becomes very important and will be a major challenge in the next decade.

Some examples of reasons to hack an EmS are

- Influencing (reducing) billing in telecommunication systems
- Changing engine characteristics in a car in order to escape from more taxes and insurance fees

The methods to deal with security in EmS are not (yet) very different from the ones used in General Purpose systems, except that more security features are (partly) implemented in hardware.

### 4.3.4 Dependability

A single term often used to characterise an EmS is dependability

Following the IFIP WG10.4 on Dependable Computing and Fault Tolerance it is **the trustworthiness of a computing system which allows reliance to be justifiably placed on the service it delivers.**

Dependability includes the following attributes of a computing system [AVLA2001]:

- *Availability: **readiness** for **correct** service;*
- *Reliability: **continuity** of correct service;* (see **Error! Reference source not found.**)
- *Safety: absence of catastrophic consequences on the user(s) and the environment;* (see 4.3.2)
- *Security: the concurrent existence of (a) availability for authorized users only, (b) confidentiality, and (c) integrity.* (see 4.3.3)

We discover the words readiness and correctness

- **Correctness** means that the functional behavior should satisfy the specifications and that producing the right result at the wrong time is not enough. The latter is expressed via the real-time behavior expressed in 4.3.1.
- **Availability** or **permanent readiness** means that non-terminating processes provide for an ongoing interaction with the environment..
- We prefer not to include Safety and Security as being part of dependability and that's the reason why they have been treated before.

Dependability includes for us the following overlapping attributes:

**Fault tolerance:**

Fault tolerance **is the ability of a system or component to continue normal operation despite the presence of hardware or software faults.** The design of the system should **avoid that the systems stops working or fails.**

Until now, technology and science have focussed for the last 30 years on hardware fault tolerance. This is achieved by some sort of redundancy in the system and might be costly to implement. However with the advent of software use, this characteristic is now much more difficult to achieve.

In aircraft or space systems, fault tolerance is achieved by incorporating multiple times the same embedded system. However, this solution is far too expensive for cars, and alternative approaches are under development. (see EC-project BRAKE as an example)

In low cost EmS, fault tolerance is mostly not an issue at all. If your cell phone dies, you just buy another one.

**Robustness:**

Robustness means **the ability of a system to maintain performance or degrade gracefully when exposed to conditions not well represented in the data used to develop the system or when the system is inadequately handled.**

Put in another way [Wikipedia]

*Robustness is the resilience of the system under stress or when confronted with invalid input. It is the ability of the software system to maintain function even with the changes in internal structure or external environment. For example, an operating system is considered robust if it operates correctly when it is starved of memory or storage space, or when confronted with an application that has bugs or is behaving in an illegal fashion such as trying to access memory or storage belonging to other tasks in a multitasking system.*

Indeed, it is sometime very hard or impossible to foresee all possible requirements when one develops an EmS. Especially, the environmental circumstances, (in what environment the system will be used, and how), are never easy to foreseen.

Also hardware may fail and provide for example wrong inputs to the system. Some systems need to be capable to continue “a best effort” functionality despite these hardware failures or environmental issues.

**Reliability:**

*It is the ability of a system or component to perform its required functions under stated conditions for a specified period of time.* [IEEE1990]

Another definition from from McGraw-Hill’s Science and Technology Dictionary is:

*The probability that a component part, equipment, or system will satisfactorily perform its intended function under given circumstances, such as environmental conditions, limitations as to operating time, and frequency and thoroughness of maintenance for a specified period of time.*

One sees that both definitions mean the same.

There is a range from highly reliable or robust to low reliable or robust systems depending on the application.

**Availability:**

Availability is readiness or uptime of a system. A product, system or service with three nines (99,9 %) or better uptime is referred to as “highly available”.

## 4.4 Characteristics of EmS

### 4.4.1 Networked

There is a clear evolution towards networked embedded systems or networks of embedded systems. Simultaneously, EmS become smaller and smaller. Indeed, while previously sensors were directly connected to the central computing elements (mostly in an analogue way), today the intelligence is pushed towards the “ends”. Sensors become in themselves EmS. They have a processor included and do some preprocessing of the measured physical property (like temperature, displacement, pressure, etc..) sending the results of this preprocessing to a central management subsystem via a digital network. The same is valid for the actuator part of a system if any.

Although there are still a lot of EmS which are not networked yet, there are serious reasons to make them networked. The first one, introduced years ago was maintenance. Indeed, if a bug was discovered in the software of EmS 20 years ago, one was then obliged to replace the ROM with the system software. This was a serious burden which took time and obliged a shutdown of the system or the whole factory for a serious amount of time.

More recently, systems are made of systems and they also get distributed. This distribution goes from very short to very large distances. Networking is an important issue simply to make all these systems work together. Again the real-time aspect might become an important issue here. Some networks needs to have a predictable behaviour and others are not that critical.

Both wired and wireless networks are used today. The importance of the latter is increasing rapidly, even in factory automation. However, the increasing number of wireless technologies available today combined with a large number of different protocols in use is a serious challenge for the EmS world.

### 4.4.2 Life cycle length

Low cost like an MP3 player commercial EmS have a very short lifecycle of a couple of years. However, going from a car, towards and aircraft and ending up in factory automation, systems are expected to run for one to three decades. This is a serious problem with the ever changing electronic technology we use today and it is a major challenge to maintain an EmS system alive (with spare parts) for up to 3 decades.

### 4.4.3 Complexity

In terms of complexity, EmS range from very simple devices with a single microcontroller chip, to very complex systems with multiple units, peripherals and network connections. This complexity is not simply proportional to the number of lines of code but relates also to the variety present in the application. This complexity changes continuously as the needs and activities in the real world continuously change. Complexity is also introduced due to the need of constant maintenance without interrupting the system which in a lot of cases should be permanently ready as stated above.

### 4.4.4 System Quality – Quality of Service (QoS)

Another term one can start using in EmS is system Quality.

Following ISO9000 quality is the **Degree to which a set of inherent characteristic fulfils requirements.**

These requirements might be expectations by the users and are not necessary what the designer wants. A couple of examples will make this clear:

- A large laser copy machines or mopiers (multipurpose copiers also called All-in-Ones or aio) are doing thousands of copies an hour. Today these machines are not insuring that all these copies will be OK. Indeed for cost reasons these machines do not necessary meet real-time requirements in order to deal with ALL the deadlines (= each copy made in time). The customer discovers this "white" pages (because the machine did not print it correctly due to a missed deadline) but thinks this is normal. There is lack of quality here.
- How many times did you need to reset your cell phone by unplugging the battery? This is a question of quality. The system should be permanently ready. Again her, poor design explained for cost reasons are at the basis of this lack of quality.

QoS is actually a term used in the telecom industry. We suggest this term should also to used to indicate the Quality of Service provided by an EmS outside the telecom industry.

### 4.4.5 Environmental factors: cost, size, weight & power consumption

There is a considerable spectrum of cost and sizes in EmS. Portable consumer devices should be very low cost, have a limited size and weight and consume as less power as possible. On the other side of the spectrum, an airplane simulator is costly, has a considerable size and weight and power consumption is not an issue at all.

Between these 2 extremes, there are all sorts of combinations possible.

One way to reduce power consumption is to use a multi-processor approach. This is the reason why more and more EmS are de factor build on a multi-processor architecture. The same tendency is observed in the general purpose application world. Vice versa, some low power general purpose processors become more interesting to be used in some EmS applications (like telecom).

### 4.4.6 Implementation strategy

As we discovered in the history of EmS, 2 types of EmS remain today if we look to the implementation:

- Board based systems with a backplane of a sort (parallel bus passive backplane, or a switching system active backplane). All boards are going into 19inch racks or similar chassis.
- Systems based on SoC or SiP going inside a purpose build housing

## 4.5 Real-time EmS explored

### 4.5.1 Definition

Different definitions of real-time systems exist. These definitions are mostly given in the framework of a specific application field. We therefore tried to make one which is independent of an application domain:

**A real-time system responds in a (timely) predictable way to all individual (unpredictable) external stimuli arrivals.**

The most important word is **PREDICTABILITY.** The system should respond to each individual external event in a predictable way, this means before the deadline defined in the system requirement. It is important to note that ***average performance is NOT the issue!***

To clarify this, let us take an example.

Banking uses transactional systems. One wants to do an average number of transactions (for examples wired bank transfers) per hour. Some transactions might take very short time, others will take much more. The user of the system is interested in performance figures like the average number of transaction per time unit. This system is not a real-time system. On average a transaction might take for example 1 second. Nobody guaranties that a single transaction will not take more than a for example 5 seconds to do..

If however the system requirements state that a single transaction will never take longer than 5 seconds, then we have an individual deadline imposed on each transaction. We then need to guaranty that each transaction will be finished within 5 seconds (with an average transaction time of 1 second). This system with this requirement is a real-time system.

The biggest confusion is that people consider real-time as “fast computing” and that one can resolve each deadline problem by using a “faster” processor or system. Science shows that this is not true at all. Special techniques need to be used to design and implement real-time systems in order to guaranty the maximum response time limit called deadline. As this is not an easy task the real-time aspect is further divided in hard and soft real-time as we will now discover.

### 4.5.2 Real-time system types

An embedded system does not necessarily need to have a predictable behaviour, and in that case it is not a real-time system. However, a quick overview of all possible embedded systems shows that you will rapidly find the need for some predictable behaviour and therefore most embedded systems need to be real-time for at least some of their functionality.

In a well-designed RT system, each individual deadline should be met. With the actual state of the practice, it is sometimes hard and also costly to achieve this requirement. Therefore people invented different ***types*** of real-time systems.

- **Hard real-time (1):** missing an individual deadline results in catastrophic failure of the system (and people will hopefully invest sufficient money in this project in order to avoid this catastrophic failure). It also means that the cost of the failure is very high. (Here hard real-time is related to safety)

- **Hard real-time (2):** missing a deadline entails an unacceptable **quality reduction**, (see 4.4.4). Technically there is no difference with the previous definition, however economically, the "disaster risk" and associated cost is limited compared to the previous case.
- **Soft real-time:** deadlines may be missed and can be recovered from. The reduction in system quality and performance is acceptable and does not introduce another than technical or feature cost. For hard real-time systems one can say that the deadline MUST be met. For soft real-time systems one will then say that the deadline SHOULD be met.
- **General purpose computing:** no specific deadlines have to be met. The system requirements are expressed in terms of **average** performance. A transaction system is a general purpose system as long as the requirements are expressed in terms of "number of transactions per time unit". If however a maximum time limit is imposed on each transaction, then it falls in the category of a real-time systems, probably soft-real time, because missing a deadline will just reduce the quality of the system.

**Quality of service (QoS):** as stated above (4.4.4), this term is also applicable to an EmS in a whole. The EmS should provide sufficient QoS. To realize this, some hard real-time functionality is needed in the system.

Today a lot of EmS are lacking this QoS for budget reasons (cost for design and production) One might expect that in the next decade, EmS requirements will focus more on QoS requirements, introducing the need for more hard real-time system instead of soft real-time.

**Deadlines:** a RT system has a lot of things to do simultaneously. For each event from the external world or environment to be dealt with, the requirements may impose a maximum time limit for dealing with this event. This maximum time limit is called the deadline for that event.

### 4.5.3 Real-time system components

To build a predictable system, ALL of its components, hardware & software, plus a good design are contributing to this predictability. Having both good hardware and a good RTOS is a minimal but not sufficient requirement for building a correct real-time system. A wrongly designed system with excellent hardware and software building blocks may still lead to disaster.

### 4.5.4 Multitasking or multithreading

Even the simplest EmS system will probably deal with more than one event coming from the environment. One can deal with these multiple events in various ways, but a multitasking or multithreading approach on one processor is a very common solution. RT engineers are used to multithreading an application since the very beginning of real-time EmS. Only more recently, general purpose software is evolving in the same direction.

Using an RTOS as basic component with many tasks or threads working closely together to deal with the application, is not just another flavour of software writing. It is for time being completely different from business software writing. A real-time software engineer is constantly busy designing the collaborative mechanism between threads and tasks and will daily deal with device drivers and interrupt handling what most software engineers hate doing.

### 4.5.5 Deadline spectrum

The deadlines to respect in RT systems might range from Pico seconds to seconds and even hours. Today's processors might be fast and deal with external events (interrupts) in the microseconds range, but power consumption may then be a serious problem. Portable EmS will have limited processor power for that reason.

Therefore there is a spectrum of technologies to be used to deal with the different deadline requirements.

The shortest deadlines (less than a microsecond) will have to be handled by hardware without software being involved. Less short deadlines (1 to 10 microseconds) can be dealt with one processor and just one program and some interrupt routines. Using an RTOS will help a lot in designing complex EmS but only deadlines in the range of .01 to 100 millisecond's and more can be reasonably handled. For very long deadlines, one might even think using humans if reaction times are specified in the hours range.

The fire brigade is considered a real-time system, because one expects them to be there within some 10 minutes or so after a call. This simple example shows also that sufficient resources need to be present in idle condition to realize this constraint. Indeed, if all fire brigade cars left for multiple fires, you will have no response if you call them.

### 4.5.6 Some RT conclusion

RT is important because some (and certainly not all) of the functionality of your system will require a predictable response on one or more events. That specific part of your system will need a real-time or predictable behaviour.

In that way, systems might lack quality of service. One might expect that in the coming years, this quality of service will become a major requirement, pushing designers to use more hard RT solutions and requesting for a real RTOS instead of just one or another non real-time OS.

Vice-versa, a 100% hard real-time system, never missing a deadline in all circumstances, is utopia. Therefore risk assessment will become a major issue in the use of EmS and design methodologies to deal with these issues will have to be developed. The introduction of the notion of QoS for EmS will become mandatory in the next decade.

# 5 Embedded systems components

## 5.1 Introduction

This chapter wants to help in the classification of EmS by means of the components used for it.

## 5.2 Hardware

### 5.2.1 Introduction

**COTS**: or Commercial Off the Shelf

EmS fall today in 2 large categories if it comes to the hardware used to implement the system. Board based systems use ready available hardware boards based on one or another bus like VMEbus, CPCI etc. In most cases this approach is used if the number of systems to be build is limited.

For very large quantities and especially in consumer devices, design techniques are more oriented towards the use of Intellectual Property (IP) and a lot of these IP components will be found into a limited number of chips used in the device.

See also the historical introduction: 2.3

### 5.2.2 The processor

One or more microprocessors are used in an EmS.

Battery operated (mobile) equipment cannot be run with the fastest processors used today due to power consumption issues. A cell phone runs not longer than 3 minutes if a Pentium IV processor should be used.

In larger non battery operated systems - like a telephone switching system - where a tremendous number of processors are used, power consumption should also be limited. Indeed, other issues show then up like heating (how to keep such high density racks cool) and energy costs (both power consumption and cooling).

As a net consequence one observes the use of processors coming from notebooks in this kind of installations.

You will therefore find a very wide range of microprocessors used in EmS, ranging from 8 up to 64 bit. Also very special processors are built for very special applications (like the hearing device for deaf born people).

The issue of “power aware computing” is one important topic in EmS research today.

### 5.2.3 Memory

For small EmS devices, memory is still limited to the strict minimum, for cost reasons or for power consumption reasons. The memory is used to store the program and the data. As in most cases, the device will not have a mass storage device such as a disk; the software should reside in memory which keeps the information when the system is powered down. Previously, Read Only Memory (ROM) was used for this function. However, as one wants to change in an easy way the software for bug fixing reasons or to modify or add to the functionality of the device through the lifetime of that device, more and more flash is used part of the basic system memory.

### 5.2.4 Mass storage

Depending on the environment where the EmS is used, rotating mechanical devices such as disks are not desired; also size may play an important role. However disk technology is now rapidly changing and smaller and reliable disks are now available. One might expect more much more disk use in the future.

### 5.2.5 Sensors

Depending on what needs to be measured and in what circumstances, a tremendous number of sensors have been developed. A sensor is a device which by its nature is transforming one physical parameter (pressure, temperature, humidity, displacement, etc.) into an electrical analog signal.

However, propagating this electrical analog signal to a distant place for measurement raisons is not always possible. Therefore more and more sensors are becoming themselves small EmS transforming the physical property in a digital signal which can easily carried over a network (wired or wireless). These small intelligent sensors might even do some signal processing before making the results available.

These intelligent sensors are then part of a networked EmS as defined before.

### 5.2.6 Actuators

Actuator: a device responsible for actuating a mechanical device. Actuators exist in all dimensions. Just like with sensors, in older systems, a direct action is taken by the EmS on the actuator via analog signals. However, more and more actuators are now becoming systems by themselves and are controlled via a network. The existence of a processor and software to do the communication with the controlling device, allows also introducing supplementary intelligence in the actuator device.

### 5.2.7 Human Interfaces

Some EmS do not need any human interaction, but others do. This interaction can be done via classical computer input output devices, but in most cases, special input output devices are foreseen taking into account the sometimes harsh environment around the EmS. Input devices are for example: buttons, joy stick, touch screen, special keyboards, eye movement detection, speech, detection of signals in nerves of the human body, etc. Output devices are all sorts of displays, synthetic speech, brain stimulation etc..

## 5.3 Software

### 5.3.1 Introduction

Time to market for new devices is reduced every day. As a consequence, a lot of designs today and even more in the future will use Commercial off the Shelf (COTS) software modules. Different modules or building blocks are available today such as RTOS, middleware, Internet modules etc.

### 5.3.2 RTOS

As complexity increases, the use of an operating system on board of an EmS will increase. Another reason for choosing an OS (instead of building one) is the availability of complementary software building blocks and developments tools. A very large number of (RT) OS are available today and it is not always clear to designers what OS to choose.

In general: ***a good RTOS can be defined as one that has a bounded (predictable) behaviour under all system load scenarios (simultaneous interrupts and thread execution).***

Indeed, as stated before (see 4.5.3), in order to build a real-time EmS, ALL components used for that system need to have a predictable behaviour. This is therefore also valid for the important building block which is the OS bridging the gap between hardware and the application software.

### 5.3.3 Middleware

Embedded systems are more and more networked. The networked devices need not only to "talk" to each other, but the need to "understand" each other. Therefore different protocols may be used.

Middleware Software serves as an intermediary between systems software (the OS or RTOS) and the application software in a networked environment.

### 5.3.4 Internet modules

Networked EmS are not only using special industrial networks (like Profibus, CAN etc.) but also more classical IT networks such as Internet based on IP (Internet Protocol). Devices need to be configured and the Internet can be used to configure the device and capture information out of it. A web server software module is ideal for this. This is only one example of typical net software actually used in EmS.

### 5.3.5 GUI

The EmS who are using a display as an output device need to be programmed to do so. Software modules needed to do this in an easy way and as such reduce time to market is mandatory.

## 5.4 Design tools and methodologies

The way used to design and implement EmS may be quite different from one system to another (due to the serious mix of characteristics of each system or device – see 4.3). The research today in this field is important.

# 6 EmS evolution

## 6.1 Introduction

There are clear tendencies in the evolution of EmS. These will be described in this chapter.

## 6.2 Subsystems subdivided in subsystems

As we have seen in 4.5.5 that real-time is a matter of deadlines to meet and these deadlines may vary over different magnitudes (from nanoseconds to hours). Most systems therefore will be subdivided in subsystems which might be implemented in hardware OR a processor + one program OR a processor using an RTOS with a multitasking approach.

## 6.3 Intelligent sensors

An immediate consequence of the previous paragraph (6.2) is the approach where people start using intelligent sensors. Indeed, previously, most embedded systems did have some measurements to do like temperature, pressure etc. Sensors were used producing analogue signals and an analogue to digital converter sampled these analogue signals with the aid of a microprocessor. Such a program was very common. This approach has also a serious drawback because you are mixing analogue and digital signals in the same systems, which is not an easy thing to deal with.

Today, sensors are made as subsystems with all electronics (and optional software) to deliver digital signals (or messages) to the EmS. They become an EmS themselves and are part of a network of EmS. Indeed the sensor in itself is then a small EmS with possibly an analogue to digital converter and some software running on a 4, 8, 16 and exceptionally a 32 bit processor delivering data in one or another way (on an industrial network for example), mostly without a RTOS. However this could change in the future if the sensors become smarter and smarter and 32 bit processors become cheaper.

## 6.4 Predictable networks

As subsystems are now smart devices they communicate with each other via a network. Different networks can be used for that, but if the overall characteristic of the system needs to be real-time (= predictable) then the subsystems should be connected with networks having a predictable response time. In that way, efforts are now going in making IT networks predictable.

## 6.5 Power consumption

Power is a major concern in EmS, especially for portable devices. Long up-time requirements combined with low weight (battery) and enough CPU power is not an easy combination.

For non portable devices, power consumption is also an issue when it comes to bring a serious number of subsystems together. Indeed, cooling and power budget may then become an issue.

A general tendency in a “greener” society is anyhow to reduce power consumption as much as possible.

Power aware computing is a new computer science who wants to deal with this problem.

http://www.dedicated-systems.com
Email: m.timmerman@dedicated-systems.com

## 6.6 Smart, smarter, smartest

Systems are today much smarter. Indeed, instead of making simple control decision, you can add more decision making intelligence and add also much more monitoring functions to make the operation of the device safer.

The end user only starts to see now that smart systems are becoming available and that they might become smarter and smarter in the future. This imposes supplementary constrains to the (RT) behaviour of the system and to the operating system if one is used making the system as a whole much more complex.

## 6.7 Memory protection

20 years ago, when people started using software systems to control and supervise nuclear installations, one of the typical requirements for these systems was the use of a memory management unit to “protect” the data from being corrupted in a way or another. This was realized with the hard and software on the DEC PDP and VAX machines. Nobody saw the interest to have memory protection in small EmS that time. However as safety and security are now becoming major issues, memory protection becomes too, even in these small embedded devices and also in other than the nuclear industry.

As software complexity grows, memory management and the use of cache and virtual memory are becoming important issues for EmS. However, one should take care because these techniques are introducing unpredictable latencies which are contrary to the need for real-time EmS (needing a predictable behaviour).

# 7 EmS Classification

## 7.1 Introduction

As one can expect from the previous, there is a wide variety of EmS in different application domains. Until recently, every application domain was convinced that they were unique in what the EmS should do and especially how it should work.

With this idea or approach research and development is then extremely related to the application domains. More recently, in the framework of ARTEMIS (http://www.artemis-office.org ), where one wants to organise future research in this EmS field a need for defining categories of EmS was develop. These categories should structure the research and make it more efficient.

As a result, it was recognized that one could discover 3 main EmS categories: cost effective systems – safety driven systems and high available systems. This categorization will be used to focus the R&D effort.

## 7.2 Cost effective systems

The cost factor is the most important one. Most of these systems are produced in large quantities. These systems will be soft real-time and will be build using IP building blocks.

These systems tend to be soft real-time.

*Examples are: Cell phones, MP3 players and other portable consumer devices.*

## 7.3 Safety driven systems

Safety is the most important factor. Depending on the application field, the implementation of this characteristic may be different. In aircraft, fail safe safety is incorporated through the multiple implementation of the EmS. In cars however where the cost factor is more important, other safety strategies will be used.

These systems tend to be hard real-time and factors such as fault tolerance and dependability are considered. As an extension, quality driven systems as described in 4.4.4 will join this category in the future.

*Examples are: medical devices, aircraft (fly –by-wire systems), cars (X-by wire systems)*

## 7.4 High availability systems

These systems should continue working at all times (and be permanently ready).

These systems tend to be soft real-time and factors such as reliability and robustness are important.

*Examples are: telecom systems (the consequence of a not working systems is less revenue for the telecom provider)*

## 7.5 Mixed systems

Of course the previous structuring does not mean that all systems just fit in these categories. A heart and lung machine for example should be both highly available and safe. Indeed, most EmS will have a flavour of all these. This explaining why EmS are a complex matter, but you probably got convinced by now.

# 8 Conclusions

## 8.1 Introduction

We finally want to give some correlation between the defined attributes and characteristics of EmS. We adopt the following scale.

Importance of the attribute or characteristic.

- -- not important at all
- - not very important
- 0 neutral
- + important
- ++ very important

The following paragraphs give an average correlation between the categories and the attributes or characteristics. Of course, by definition, most systems will fall outside these "averages", but that is what "averages" are about, isn't it? This first draft of this table is made from guts and feeling of the author and should be refined after numerous discussions and research.

## 8.2 Categories vs. attributes

| *attributes* | Cost driven | Safety driven | High availability |
|---|---|---|---|
| Real-Time (RT) | 0 (Soft RT) | ++ Hard RT | 0 (Soft RT) |
| Safety | -- | ++ Safety Critical | 0 |
| Secure | - | 0 | 0 |
| Dependability | -- | 0 | ++ |

## 8.3 Categories vs. some characteristics

| *characteristics* | Cost driven | Safety driven | High availability |
|---|---|---|---|
| Life cycle length | -- | + | 0 |
| Cost | -- | ++ | 0 |
| Low power | ++ | 0 | - |
| Backplane based | -- | 0 | + |
| SoC or SiP | ++ | 0 | - |

Examples are given in part 2 of this document and will be classified in the 3 categories synthesized in a combined table as shown below:

| | Cost driven | Safety driven | High availability |
|---|---|---|---|
| ***attributes*** | | | |
| Real-Time (RT) | 0 (Soft RT) | ++ Hard RT | 0 (Soft RT) |
| Safety | -- | ++ Safety Critical | 0 |
| Secure | - | 0 | 0 |
| Dependability | -- | 0 | ++ |
| ***characteristics*** | | | |
| Life cycle length | -- | + | 0 |
| Cost | -- | ++ | 0 |
| Low power | ++ | 0 | - |
| Backplane based | -- | 0 | + |
| SoC or SiP | ++ | 0 | - |

# 9 References

[BOSI2005] Bruno Bouyssounouse & Joseph Sifakis (Eds.), Embedded Systems Design. The ARTIST Roadmap for Research and Development, Springer, LNCS 3436, ISBN 3-540-25107-3, 2005

[ZURA2006] R. Zurawsky (Ed.), Embedded Systems Handbook, CRC Press Taylor & Francis Group, US, ISBN 0-8493-2824-1, 2006

[FATU2005] FAST – TUM: Study of Worldwide Trends and R&D Programmes in Embedded Systems in View of Maximising the Impact of a Technology Platform in the Area. Final report 18.11.2005 for the European Commission.

[TNOI2005] TNO/IDATE report on "Software intensive systems in the future". September 2005V5.

[QING2003] QING LI with Caroline Yao: Real-Time Concepts for Embedded Systems. CMPBOOKS, ISBN: 1-57820-124-1, 2003

[IEEE1990] nstitute of Electrical and Electronics Engineers. IEEE Standard Computer Dictionary A Compilation of IEEE Standard Computer Glossaries. New York, NY: 1990.

[AVLA2001] A. Avizienis, J.-C. Laprie and B. Randell: Fundamental Concepts of Dependability. Research Report No 1145, LAAS-CNRS, April 2001.

[OTIS]
http://www.otis.com;
http://www.utc.com/profile/facts/history.htm

[CALI2006] http://www.boe.ca.gov/proptaxes/pdf/lta06055.pdf

[MASC1998] Manfred Schlett, Hitachi Europe GmbH, Trends in Embedded-Microprocessor Design, IEEE-Computer Magazine, 1998

# 10 Glossary of terms

**ARTEMIS:** Organisation which wants to secure employment in Europe and reinforce European industry's leading position in Embedded Systems technology, thereby yielding both direct and indirect benefits to the European citizen. http://www.artemis-office.org

**cPCI:** compact PCI

**CPU:** CPU stands for Central Processing Unit and is often, simply called, the processor. The CPU is a microchip that is installed on a printed circuit board and acts as the computer's brain possessing instructions and manages the flow of information through a computer system. http://en.wikipedia.org/wiki/CPU

**DSP:** Digital Signal Processor http://en.wikipedia.org/wiki/Digital_signal_processor

**FPGA:** Field Programmable Gate Array. A class of integrated circuits pioneered by Xilinx for which the logic function is defined by the customer using development system software AFTER the IC has been manufactured and delivered to the end user. Mask programmed gate arrays are another type of IC whose logic is defined DURING the manufacturing process. http://en.wikipedia.org/wiki/FPGA

**I/O:** Input Output: is the collection of interfaces that different functional units (sub-systems) of an information processing system use to communicate with each other http://en.wikipedia.org/wiki/Input/output

**IP:** Intellectual Property: http://en.wikipedia.org/wiki/Intellectual_property

**IPTS:** The Institute for Prospective Technological Studies (IPTS) is one of the seven scientific institutes of the European Commission's Joint Research Centre (JRC). Its mission is to provide customer-driven support to the EU policy-making process by researching science-based responses to policy challenges that have both a socio-economic and a scientific or technological dimension. http:// www.jrc.es

**ITEA:** is a strategic pan-European programme for advanced pre-competitive R&D in software for Software-intensive Systems and Services (SiS). http://www.itea-office.org/

**LEM:** Lunar Excursion Module or the Apollo Lunar Module was the lander portion of the Apollo spacecraft built for the US Apollo program to achieve the transit from moon orbit to the surface and back. http://en.wikipedia.org/wiki/Apollo_Lunar_Module

**PDP:** Programmable Digital Processor: early range of mini-computers. http://hampage.hu/pdp-11/

**QBus:** is a bus used in Digital's PDP-11 and MicroVAX series of computers. A bus is a subsystem that transfers data and power between computer components inside a computer. Unlike a point-to-point connection, a bus can logically connect several peripherals over the same set of wires.

**RTOS:** Real-Time Operating System: An operating specifically designed to insure real-time behaviour to an (embedded) computer systems http://en.wikipedia.org/wiki/RTOS

**SoC:** System on a Chip: http://en.wikipedia.org/wiki/System-on-a-chip

**VME:** Versatile Modules Europe

http://www.dedicated-systems.com
Email: m.timmerman@dedicated-systems.com

# 11 Acknowledgement

This report should not have been possible without the comments and reactions of a lot of people who took the time to go through multiple versions of this document.

I want to explicitly mention Marc BOGDANOWICZ from IPTS - ICT Unit, Information Society, Competitiveness and Growth and Javid KHAN, Scientific Officer at DG Information Society and Media, for the numerous hours they spend to go through the drafts of this document generating suggestions for enhancements.

To finish, I want to quote Javid's definition of an EmS: *"embedded systems is a computer that does not look like a computer"* which is certainly a good synthesis of what we have been discussing.

# 12 Content table