International Journal of Computer Applications (0975 – 8887)
Volume 106 – No.11, November 2014

A Comprehensive Review of Embedded System Design Aspects for Rural Application Platform

Pallab Dutta
Team Leader
C-DOT, EC
Bangalore

Upendra G
SRE
C-DOT, EC
Bangalore

Giribabu E
RE
C-DOT, EC
Bangalore

Sridharan B
Group Leader
C-DOT, EC
Bangalore

Vipin Tyagi
Executive Director
C-DOT, Mehrauli
New Delhi

ABSTRACT
Abstract: There has been an exponential growth in the penetration of embedded systems in almost all domains ranging from automobiles to home appliances to ICT. In the recent times, there has been lot of emphasis towards using embedded systems and embedded computing processors even in desktop kind of application environment. This trend is due to the intrinsic benefits offered by the embedded systems over traditional desktop systems and also due to rampant increase in the computational power of embedded processors. Rural application platforms are meant to be used for executing applications required for providing different e-services, self-help services relevant in the rustic areas. The eco system prevailing in rural areas, especially in the developing countries are largely challenging in terms of power outages, erratic grid power conditions, high temperature ranges, varying humidity, dusty environment etc. Although embedded systems offer many advantages, the embedded systems have some serious limitations as well, particularly in the contexts of application platform. In this paper, we discuss and analyze different aspects and challenges towards design of embedded systems for rural application platform. Also we discuss on approaches towards mitigating those challenges pertinent to design of an embedded system suited for rural application platform.

General Terms
Embedded System, Microprocessor, Operating System

Keywords
Embedded system, Embedded software, System Design, Application platform

1. INTRODUCTION
In last couple of decades there had been unprecedented expansion and growth in almost all facets of ICT (Information Communication Technology). Many initiatives have been taken towards e-services by both governmental and private agencies across the world and many of them are successful. After a detail and deeper look, it can be seen that majority of these successful initiatives are restricted to urban areas and success rates in rural areas are much lower. This is due to many technical and commercial reasons and constrains. One of the major technical challenges to overcome for a successful e-service initiative is towards design of an efficient application platform that is suitable to rural eco-system. A pertinently optimized embedded system can be designed to rural application platform requirements.

Embedded system: There are many definitions of embedded system; in most generic terms, an embedded system is one that has computer-hardware with software embedded in it as one of its most important component [4]. It is a dedicated computer-based system for an application(s) or product. It may be either an independent system or a part of a larger system. As its software usually embeds in ROM (Read Only Memory) it does not need secondary memories as in a computer. Embedded systems are classified based on complexities, functionalities etc. The following are the three broad classifications of embedded systems [4]: small scale (or low end) embedded systems, medium scale embedded system, high end embedded system. Small scale systems are designed with a single 8- or 16-bit microcontroller; these have simplified hardware & software, mostly have internal memory (both ROM and RAM) and do not possess any operating system, works on single executing thread. These are typically low power, may even be battery operated. Medium scale embedded systems are usually designed with a single or few 16- or 32-bit microcontrollers or DSPs or Microprocessors. These have both hardware and software complexities. These mostly have some amount of external memory. Sophisticated high end embedded systems have enormous hardware and software complexities and may need cluster of processors or configurable processors and programmable logic arrays. These are used for complex applications that need hardware and software co-design and integration in the final system. These highend systems will have several Megabytes of external high speed RAMs (like DDRs) and solid stage storage memory also. Some functionalities such as cryptographic algorithms, graphic processing algorithms, video decoders, discrete cosine transformation and inverse transformation algorithms, Network protocols and network drivers functions are implemented in the hardware (through specialized co-processing units) to obtain better speed performance. These have full-fledged OS (in some cases Real Time OS) running and in many cases can have high end displays subsystems connected also. In recent times there has been lot of focus and research interest in this category of embedded system as the applications are becoming more and more computationally rich and sophisticated.

Application platform: By definition, an application platform provides services to applications. Whatever is required to successfully run/execute an application is made available by the application platform. This typically includes hardware, device drives, operating systems etc. The exact services offered by the application platform depend on specific application. Some of the applications such as e-services require display support, network connectivity, graphic and video capabilities. The success of a system also depends on the suitability of application platform for a given set of applications in a given eco system of deployment scenario. There are many parameters of an application platform and there are conflicting requirements, hence optimization is huge challenge.

In this paper, we analyze the optimization challenges pertaining to embedded systems designed for rural application platform. In section 2, we discuss and analyze the requirements to be met by an embedded system to be suitable as rural application platform, in section 3, we discuss the evolution of embedded system from past to the present. Design challenges in the contexts of conflicting requirements and optimization challenges w.r.t embedded systems for rural application platform are discussed in section 4, the support required from eco system and our approaches towards meeting the design challenges are described in sections 5 and 6. In section 7, the future trends in embedded system are also discussed.

2. REQUIREMENTS SPECIFICATIONS OF RURAL APPLICATION PLATFORM

There are various requirements those are to be met by a system in order to use it as a rural application platform. The rural eco system scenarios are very demanding and these makes the application platform requirements very rigid and also conflicting in many cases. If we try to optimize one parameter, this can cause other parameters shooting out of the optimized range. So in-order to optimize the system as a whole, all the parameters needs to be balanced in such a way that overall peak system performance can be achieved. This scenario can be achieved by weighing the parameters and balancing them appropriately, this need not necessarily means that all the individual parameters are at their optimum values when analyzed in isolation. The generic requirements to be met by the application platform are given in Table 1. All these requirements are expressed at broader terms in order to satisfy the overall system performance. If the requirements are expressed in finer granularity, this leads to higher conflicting scenario and overall performance objectives suffers. As an example, in general computing power is expressed in terms of MIPS (Million Instructions Per Second) or MFLOPS (Million Floating Point Operations Per Second), but such low level parameters do not guaranty the required functional requirements are satisfied. This is because the embedded system under consideration may consist of a SoC (System on Chip) and other building blocks that have dedicated video decoding engines, Ethernet MAC engines. In such cases, even if CPU row horse power is less, this would be able to play youtube video. On the contrary, if the embedded system does not contain the dedicated engines for video decoding and Ethernet MAC, a comparatively higher CPU MIPS also would not suffice and the high level functional requirements cannot be met. Some of the requirements are shown in Table 1 are conflicting to others. E.g.; Higher computing power is directly proportional to power consumption. The high computational CPUs consume more power than comparatively low computational CPUs (Pentium4 is around 60W vs Atom D510 is around 13 W). In the table1 point 8 (Software compatibility / Porting effort) conflicts with 4,5. x86 architecture based CPUs (typically desktops) easily satisfies point 8 but does not satisfy points 4,5. A system with Atom D510 satisfies 1&8 but does not satisfy some of the other parameters.

Table 1: Generic System Requirement Specifications of Rural Application Platform

Serial Number	Parameter	Requirements
1	Computing Power	Shall be capable of running Youtube video at 640X480 resolutions.
2	Power Supply	Single DC power supply, so that it is easier to interface with Solar Panel etc.
3	Fan	Shall be Fanless (Mechanically moving parts to be avoided)
4	Mechanical/Electrical Ruggedness	Shall be rugged, able to operate from 0 C to 50 C, shall have only solid state storage memory (no mechanically rotating parts).
5	Power Consumption	< 10W
6	Cost	Low as compared to a standard Desktop
7	Size	Considerably Smaller than a standard Desktop
8	Software compatibility / Porting effort	Shall have maximum compatibility and shall require minimum porting effort.

3. EVOLUTION OF EMBEDDED SYSTEM FROM PAST TO THE PRESENT

The principles of operation of system components and design methodologies are essentially the same in all embedded systems although there are an infinite variety of embedded systems.

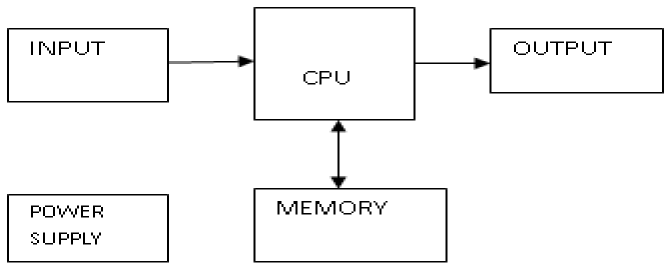

Figure 1: Generic Block Diagram of Embedded System

For instance, every embedded system contains a processor and software that runs on the processor [2]. The processor may be a microcontroller or a microprocessor. Also, in order to have software there must be a place to store the executable code and temporary storage for run-time data manipulations which will take the form of ROM and RAM respectively. In case of small memories they may be contained in the same chip as the processor. Generally microcontrollers will have such arrangements. Otherwise one or both types of memory will reside in external memory chips. Furthermore, all embedded systems also contain some type of inputs and outputs. Inputs to the system are generally sensors and probes, communication signals, or control knobs and buttons. Outputs are generally displays, communication signals, or changes to the physical world. This is illustrated in Fig.1.

Evolution:

Embedded system consists of CPU (that may be microprocessor/microcontroller/DSP or combinations), memory and input / output subsystems. There had been exponential upgrade in all components of embedded system.

Apollo Guidance Computer was one of the first publicly recognized embedded systems Developed by Charles Stark Draper at the MIT Instrumentation Laboratory in 1966 [5]. Intel 4004 was the first microprocessors used in calculators. Since that time there has been lot of change. Table 2 shows the evolution of embedded system over last 4-5 decades.

Table 2: Embedded Microprocessor Evolution

Parameter / Year	**Microprocessor semiconductor Technology**	**Performance**	**Clock speed**	**Memory Size**	**Technology trend**	**Application areas**
1970s	10 micron	0.6 MIPS	1 Mhz	64 KB	4 bit / 8 bit microprocessor	Military/ Aerospace
1980s	1 micron	20 MIPS	10 Mhz	1 MB	8 bit embedded microcontroller/ microprocessor	Consumer /Wireless Telecom
1990s	0.3 micron	300 MIPS	100 Mhz	32 MB	32 bit microprocessor / embedded microcontroller	Entertainment/Internet/ Games/Toys/Auto
2000s	0.1 micron	1500 MIPS	800 Mhz	512 MB	System on chip	RFID/Appliances
Present	0.035 micron	2500 MIPS	1.2 Ghz	2 GB	Distributed system on chip with multi core	Sensors/Ubiquitous

4. DESIGN CHALLENGES

In this section, we discuss design challenges, conflicting requirements and optimization challenges w.r.t embedded systems for rural application platform. Satisfying the requirements as given in Table 1 poses major design challenges. Some of the requirements as shown in Table 1 are conflicting to others. E.g.; Higher computing power is mostly directly proportional to power consumption. The high computational CPUs consume more power than comparatively low computational CPUs. Pentium 4 is around 60W offering 9700 MIPS. Atom D510 is around 13 W offering 8400 MIPS and ARM Cortex A8 with less than a watt of power offers 2000 MIPS [13,14]. Desktop computers are not mechanically and electrically rugged. It contains moving parts in Hard disk, upgradeable RAM slots etc. In an embedded system, such parts and design are not acceptable. Harddisk contains mechanically moving parts and thus reduces the mechanical ruggedness, RAM slots are provided in desktop systems in-order to support future up-gradation of memory, but this cannot be implemented in an embedded system as putting the RAM in slots is a not mechanically strong arrangement and reduces the mechanical strength of the system. Hence the memory requirements in an embedded system need to be finalized beforehand as field up-gradation is not possible here. In the table1, point 8 (Software compatibility / Porting effort) conflicts with 4,5. x86 architecture based CPUs (typically desktops) easily satisfies point 8 but does not satisfy points 4,5. As we can analyze that, there is hardly a practical system possible that satisfies all the requirements in totality. Hence it is imperative that the parameters / requirements are to be weighted appropriately. The weighing of each parameter depends on the specific application and product/project contexts. In some cases, the power consumption may not be as critical as cost and software compatibility etc. Hence there cannot be any fixed weighing factors and these are context specific. The challenge lies in judiciously weighing the parameters and there by properly performing engineering tradeoffs.

5. DEPENDENCY ON ECO SYSTEM

Building a complex embedded system, as for rural application scenario is definitely a challenging task w.r.t both hardware and software design aspects. Here, the key design challenge is to build an extremely powerful system (in terms of features those provided by the system so as to make it ease of use for rural people) at very low power consumption (~ 10w), low NRE cost and at the same time meeting all the requirements as listed in Table 1. In fact both the hardware and software are to be co-designed in any embedded system. Evaluation of overall system cost, power consumption, performance, size, components availability, flexibility, software tools, time-to market etc are the key factors which drive the SoC selection which is very important in meeting the required system specifications. Also the eco-system support along with SoC vendor support for both hardware and software components are very much required to build a highly optimized system and play a crucial role in life-cycle of the system design and development. Support from eco system is one of the most important aspects towards design of a successful rural application platform satisfying the challenging and many a time, conflicting requirements. We discuss below the support required from eco system towards design of an embedded system for rural application platform:

- Dependency on eco system for Hardware design aspects: Various components within an embedded system including processor or microcontroller, memory (ROM and RAM), hardware and software interfaces (or environment peripherals) are the important elements which drive the design approach of the whole system [3,6]. The current trend in the embedded industry is that the processors are evolving towards a fully featured system on chips (SoC) which are customizable for a specific application scenario. These SoCs are customised for specific requirements to build complex systems with required interfacing modules. Design of an embedded system using these SoCs require comprehensive support from the eco system including most importantly the SoC vendor. The peripherals compatible with such SoCs plays a very important role in the successful design. All the hardware interfacing details, routing guidelines for high speed critical signals, packaging details, assembly guidelines are required to be provided by the vendor with sufficient detailing. As these SoC are having high level of integration, hence these guidelines are extremely important towards design of a working fault free system. Also the PCB design tools required to support the SoCs are very important and plays a pivotal role in the overall hardware design flow for an embedded system. The availability of Hardware interfaces specifically required for an application platform is to be verified.

- Dependency on eco system for OS/Device Drivers/ BSP: whether the system is microprocessor based or a SOC based, embedded system components influence and dictate how the embedded software alias firmware is designed and developed [6-8]. In an embedded system environment, one important aspect is that the Kernel does the startup & initialization of the board, memory and I/O resource management, necessary drivers for both on-board hardware devices and external peripherals. Root File-system provides all the required run-time user-space as well as kernel-space libraries, system binaries, start-up scripts (including DDR initialization), feature specific configuration files for the whole embedded system. Most of the silicon vendors' provide board support packages (BSPs) i.e boot-loaders, kernel and rootfs supported for their SoC. In an embedded system, it is required to customize and port these BSPs as per their customized hardware design towards meeting their specific requirements. Open source OS - Linux kernel supports most of the processor architectures (x86, PowerPC, MIPS, ARM, SPARC, SuperH etc.) available in the industry for diversified applications, of which ARM architecture has been significantly penetrated in the embedded industry. Linux kernel sources have architecture-independent, highly optimized device drivers, resource and power management modules for achieving better system performance while consuming low power. Linux provides the eco system support, hence reusing of the above software components is highly beneficial for designing an optimised system. Only the architecture and board specific codes are required to be modified to suit as per their design requirements[8]. Proprietary drivers are required to be integrated to the firmware; proprietary vendor support is crucial and is provided by the vendor is most cases.
- Dependency on eco system for efficient tool-chain /Development environment and applications: Software development tool-chains (include compilers, linker, assembler and debuggers) either provided from SoC vendor or GNU tools should be used to build the libraries and binaries required for the target. It is equally crucial to ensure that open source generic application blocks are available in open domain.

6. APPROACHES TOWARDS MEETING THE DESIGN CHALLENGES

There are many challenges towards design of a rural application platform. These are discussed in section 4. There are various approaches towards meeting the challenges, most of the time an engineering tradeoff is required to be made. The selection of the computing module (microprocessor/SoC) is very important. The RISC based SoC/microprocessors are power efficient by architectural design. ARM, PowerPC are common architectural choice with ARM being most prominent one in terms of penetration as well as eco system support. The selection of computing module is also driven by availability of required built-in interfaces. For rural application platform, display interfaces are mandatory. Hence SoCs with built in display interfaces (VGA/HDMI) becomes a natural choice. However, the processing power of microprocessor/SoC needs to be properly verified. As discussed in section 1, the row CPU computing power interms of MIPS/MFLOPS are not always a clear indicator and does not guaranty the performance requirements. The built in hardware modules (e.g.; video decoders) plays a very important role and enhances the overall performance. Hence while selecting the microprocessor/SoC, the availability of built-in hardware engines/co-processing modules needs to be verified against the required functionalities. Also, it is important to ensure the required driver support for such co-processing modules are available and easily integrateable. Design of memory subsection is equally crucial. The faster RAMs (SRAM) are costly and power hungry and hence the usage of this must be limited in the design. Low power DDR is a good choice for optimized performance and power consumption. Power section design plays a decisive role in the overall power efficiency of the system. Design of regulators with higher efficiency improves the overall power efficiency of the system. Designs with support of sleep modes/power saving modes also gives advantages in terms of power consumption, however in such designs, system response times needs to be verified and ensured to be within the required specifications. In an embedded system, there are design challenges in operating systems as well. Linux kernel has evolved significantly in terms of the features it provide and the architectures it supports [8]. Though as per architecture, Linux is monolithic, all the drivers can be made dynamically loadable. It's system call interface (kernel-to-user space API) enables the application sources portable across the systems. Hence using linux kernel in any embedded system architecture, the kernel requires heavy tailoring in terms of the features it provides so as to remove the resources for unnecessary processes those may affect the speed of the system. Kernel has to be trimmed/customized by tuning the entire Software interface and their modules within and outside of the SoC by keeping the HW design in context. According to the requirements of HW design, the appropriate driver or resource management modules within the kernel have to be replaced by the proprietary modules or the open-source modules need to be ported for the target SoC and functionalities. All the modules like interrupt handler, GPIO and board-specific initialization sources for the busses and peripheral interfaces like USB, I2C, SPI and device drivers need to be customized and this poses a significant challenge in successful porting of drivers and low level software. The overall performance of the embedded system depends heavily on efficiency of these modules. There is architecture dependent codes for memory management, CPU related, these are key parts which decide the system functionalities as well as performance. Also as per the HW environment, user-space applications and even the libraries have to be customized and built for the required target[6-8]. As described in [8], appropriate protocols and networks/bus-types must be chosen carefully to reduce communication overhead. This enables the designed system to satisfy speed requirements and also limit power dissipation.

7. FUTURE TRENDS IN EMBEDDED SYSTEM DESIGN

Embedded system has undergone in last four decades many fold enhancements in computational power, complexities and other nonfunctional features. The design approaches for embedded system has also seen quite a few quantum shifts. The requirements to be met by the embedded systems are becoming increasingly complex and intricate. To cater to future requirements, new embedded system design approaches are getting evolved. Below we discuss some of the most prevailing ones [9, 11].

- Higher Integration
  - Complex System-on-Chip (SOC) : A microprocessor plus additional peripheral support devices integrated into a single chip. Integration of CPU, DSP and other specific engines (GPU, Crypto engines) are in the trends towards complex SoC realization with higher level of integration.
  - Core based SOC : Reusable Intellectual Property (IP) circuits or cores are pre-designed and pre-verified functional units (E.g. ARM, PowerPC, DSP). Both Hard and soft IP cores are in the trend.
- Hardware-Software Co-design

The latest trend in embedded system design is Hardware-Software co-design. Traditional design approach calls for early separation of Hardware and Software. This traditional approach results in long design time/high cost/ hard to maintain and debug. Concurrent design of hardware & software components, evaluate the effect of a design decision at early stage by “virtual prototyping” to enable systematic design space exploration [9]. Sometimes, even “rapid prototyping” approach is taken to mitigate design and architectural risks. Advantages of Hardware-software co-design are significant performance improvement for embedded system design, earlier architecture closure, reducing risk, earlier HW/SW integration, reducing design cycle, developing HW/SW in parallel.

- Advanced Design flow

Advanced design flow is key towards developing a bug free complex embedded system. To implement advanced design flow, better tools for high level language translation are required. System level HW-SW Co-simulation helps designers towards logical partitioning of functionalities in hardware and software modules also helps in optimized design partitioning.

- Extreme miniaturization

The world is moving towards Machine To Machine (M2M) communication, ubiquitous computing and internet of Things (IoT). In such scenarios, every small device will have minimum communication and computing capabilities; that is embedded systems will be built into enormous number of small items. The trend is towards extreme miniaturization of such embedded systems to facilitate M2M, Ubiquitous Computing. Nano technology will surely play an important role towards physical realization of those very tiny embedded computing elements.

- Ultra low power design

Energy is one of the most important nonfunctional requirements for embedded system design. Not only hardware platform but also software contributes to the energy cost of embedded system. As given in [11], low power design for embedded systems is important whether the power is coming from a dedicated conventional energy source, a battery or a non-conventional energy source. All systems benefit from low power design with reduced cost, increased reliability and compliance with energy standards. At a system level, there are many trade-offs which can be made depending on the application, usage and environment where the embedded system intend to operate. In general, the strategy to reduce energy consumption is to [11]:

- minimize the power required at run-time
- maximize time spent by the system in lowest power mode without compromising on the functionalities and requirements.
- minimize conversion losses by using efficient voltage regulators (switching regulators) and efficient design. Where ever linear regulators are used, the difference voltage shall be minimum to reduce power loss.
- minimize CPU idle time and CPU usage in non-functional tasks.
- maximize use of power efficient peripherals
- Design interrupt driven system rather than polling
- minimize duration of usage of power hungry subsystems such as displays, display needs to be in power save mode whenever it is not in use.

These can be further refined and optimized based on the particular requirements of the embedded system. For example, the clocking speed of the processor may not required to be operated at its maximum speed for a given usage, hence by operating the microprocessor below the maximum speed will save power. At the core of nearly all embedded applications platform is a processing unit which can be either a custom ASIC, a FPGA or a microprocessor/SoC. This component is usually the critical piece in developing a low power system.

The extreme design objective of a low power energy efficient embedded system is to move towards energy harvesting system that can achieve perpetual operation without depending on external grid/battery power source. The most recent research trend is focused towards designing embedded systems those are extremely low power and do not require external source of energy. An energy harvesting system can achieve the objective of perpetual operation by ensuring that the harvested energy meets or exceeds the energy consumed by the system during operation. Energy management is a critical aspect of designing an energy harvesting system. The important part is to determine the available power output of the harvester. Generally the efficiency of such power harvester is low (efficiency of solar panel is around 15% to 20%, wind turbine is around 25% to 30%). There are many commercially available energy harvester, these convert solar, mechanical or thermal energy into electrical energy. The critical part of an efficient energy harvesting system is to maximize output power of the energy harvester. Equally important in designing an energy harvesting system is minimizing the power consumption of the embedded system without compromising on the functional requirements. Low power consumption can be achieved by selecting components with low leakage specifications and by using an ultra low power microcontroller (MCU) / Microprocessor/ SoCs and all other subsystems. Also on the software design aspects there is latest focus towards achieving high energy efficiency of the system.

8. CONCLUSION

There are various challenges towards design of embedded system, specifically for rural application platform. In this paper we have discussed the requirements to be met by the rural application platform, the challenges towards meeting those requirements, also we have discussed on the future trends in the embedded system design. In our analysis, we find that requirements are conflicting in nature and it is beneficial to express the requirements at higher level of functional abstraction rather than a very granular level. This will be useful in the design of embedded system and to narrow down to optimization approaches and analysis specifically in the context of rural application platform. In future, we plan to perform quantitative analysis of the optimization approaches for embedded systems using statistical methods and feature selection methods like Principal Component Analysis.

9. REFERENCES

[1] Berger, Arnold S. "*Embedded System Design: An Introduction to Processes, Tools, and Techniques*". CMP Books. 2002.

[2] P.I. Okwu and I.N. Onyeje "*Ubiquitous Embedded Systems Revolution: Applications and Emerging Trends*", International Journal of Engineering Research and Applications (IJERA) Vol. 3, Issue 4, Jul-Aug 2013, pp.610-616

[3] Steve Heath "*Embedded Systems design*" 2nd ed.

[4] *highered.mheducation.com/sites/dl/free/007340456x/167481/sample1.pdf* – *" Introduction to Embedded System"*

[5] *www.ece.uah.edu/~milenka/cpe32310S/lectures/cpe323intro.pdf* - Dr Aleksandar Milenkovic, "*Introduction to Computer Systems"*

[6] F Vahid and T Givargis "*Embedded Systems Design: A Unified Hardware/Software Introduction*"

[7] Michael Barr; Anthony J. Massa (2006). "*Programming embedded systems: with C and GNU development tools"*. O'Reilly

[8] Christopher Hallinan *"Embedded Linux Primer"* Prentice Hall

[9] http://www.secs.oakland.edu/~ganesan/old/courses/SYS595%20F06/Embedded%20System%20PPT%20Handout.pdf. Subra Ganeshan. "*Embedded System – Present and Future"*

[10] Dr. Daug Locke, "*Real-Time and Embedded Systems: Past, Present and Future*".

[11] Øystein Ra and Torbjørn Strøm "*Trends and Challenges in Embedded Systems – CoDeVer and HiBu Experiences"*.

[12] http://training.linuxfoundation.org/free-linux-training/linux-training-videos. Linux device drivers, internals, Embedded Linux.

[13] http://www.intel.com/content/www/us/en/intelligent-systems/previous-generation/embedded-pentium-iv.html. Pentium 4 power consumption.

[14] http://ark.intel.com/products/43098/Intel-Atom-Processor-D510-1M-Cache-1_66-GHz. Intel Atom processor D510.