JUST ENOUGH COMPUTING

by

Johan DAMS

Genesi U.S.A., Inc.
200 Patterson Ave.
San Antonio, TX 78209
USA
Phone: +1 210 829 5500
Fax: +1 210 829 5426

Email: jd@puv.fi
URL: http://www.genesippc.com

Acknowledgments

I would like to take this opportunity to thank Mr. Bill Buck and Ms. Raquel Velasco for the opportunity they have given me to work on several exciting projects, and to have introduced me to some truly wonderful people whom I can now call my friends. I hope we can continue this co-operation and friendship for a long time to come.

I especially want to express my gratitude to my fiancé for putting up with my work schedule, which sometimes seems to take more time than there are hours in a day. Her encouragement and understanding truly keep me going, and without her I would never be where I am now.

Abstract

"Why not use an embedded processor as a general purpose solution if the targeted requirements are met by the capability available. First, unlocking opportunity has everything to do with price. Last, you don't need a rocket ship to travel to the grocery store."

"This is not about a processor or a "desktop" solution, it is about developing an interface that works anywhere for many things and many people. The objective is to help people do what they do already, better and more conveniently for less – that ought to be just enough."

Quotes by Mr. Bill Buck, CEO of Genesi.

If we look at the educational landscape, we see that many schools, universities and organisations world wide struggle with finding a budget to equip themselves with the needed infrastructure to allow students to experience things we in the Western world often take for granted: Internet access, infrastructure for programming courses, laboratories for embedded systems and robotics, etc.
In this paper we will discuss some real and existing implementations which can not only help those institutions which have financial issues, but also those who are willing to experiment with a new environment for certain aspects of IT education. The focus here will be on projects based around the EFIKA single board computer.
The EFIKA system, developed by a company called Genesi [1], is based around a Freescale MPC5200B System on Chip (SOC), originally developed exclusively for the automotive industry, running at 400MHz. The board has 128 MB Ram, PCI interface, 2 USB ports, AC97 audio, an IDE interface and Ethernet controller. The whole board consumes around 1.2 Watts of power, is completely passively cooled, and costs 99 USD.
The low power consumption yet high performance combined with the low price make it an attractive base for some very exciting applications which can benefit both the educational institutes, as well as corporations seeking a competitive edge. The goal is not only to provide tools for people to use, but also to give young entrepreneurs a solid base and easily accessible system on which to build their ideas and dreams. A wide range of applications are already being developed by many developers world wide. Open Source Software is one of the key pillars to this success, but we will see that closed source applications are important just as well.
The findings in this paper are real applications in use today, tried and tested developments in several different parts of the world, and are ongoing and expanding to this day.

Keywords: Computing, Power efficient, education, Open source, EFIKA

Contents

1 Introduction

Recent developments in the embedded world have made it possible to have the same amount of computing power available on a single chip than personal computers had just a few years ago. Whats more, these embedded devices consume only a fraction of the power than this personal computer did. The main idea is thus: we managed fine with those computers to do word processing, browse the web, email, etc - why can't we do that again - only much more power efficient?

Open source software, and the communities of like-minded people there, have been one of the driving forces in this direction. A simple example would be running Linux on your toaster. Why would you do that - it's completely useless? Because you can, and because it's fun! The same principle applies here, but is a little more focused on being actually useful, while not taking away the fun out of the equation. The openness of both the software and the people is key here.

During the course of this paper, a selection of applications will be presented which have implementations today, all over the world. As you will see, there is a social relevant theme which helps among others to develop those areas in this world which have until now been relatively cut off from the digital world we take for granted. The same ideas can be retrofitted to apply to the Western societies and the educational landscape specifically.

2 The Basis

Every IT project needs a couple of basic ingredients to get started. These are an idea, imagination, maybe some programming skills, and a platform to develop for. The EFIKA, based around the MPC5200B, is the platform, as depicted in Figure 1. The MPC5200B itself is a true system on chip, featuring a 400 MHz MPC603e core with an integrated double precision Floating Point Unit (FPU) and incorporates a hardware-based memory management unit (MMU). The MPC5200B was designed for fast data throughput and processing. The integrated BestComm DMA controller offloads the main MPC603e core from I/O intensive data transfers. An integrated Double Data Rate (DDR) memory controller accelerates data access with an effective memory bus speed of 266 MHz. A high-speed PCI interface backed by the BestComm DMA controller and DDR memory support enables high-speed data transfers in and out of the MPC5200B.

Some additional MPC5200B features include:

- 16K Instruction and 16K Data Caches
- 10/100 Ethernet MAC
- ATA/IDE Interface
- USB 1.1 Host (two each. USB 2.0 compatible)
- Programmable Serial Controllers (six)
- Serial Peripheral Interface (SPI)
- I2C (two) and I2S (up to three)
- CAN 2.0 A/B (two)
- GPIO (up to 56)
- 8 Timers

The EFIKA has no AGP slot, but AGP graphics cards can be used with a right angle PCI to AGP 1x convertor, which is included with the board. If graphical output is not required, this can be left off to save both power and space. Power consumption of an EFIKA with an ATI9250 graphics card and hard-drive attached is around 10Watt. Without the graphics card, but with hard-drive, power consumption below 4 Watt can easily be achieved. Use of a flash drive can reduce this further to 1.2 to 1.5 Watt.

Developers and users of this platform from all over the world get together on powerdeveloper.org, a forum and knowledge base, where they discuss problems, announce recent developments, and keep a log of their project in the form of a blog. Over 400 EFIKA boards have been

sponsored by Genesi in co-operation with Freescale to the developers in order to turn project ideas into working implementations. Many of the developers there have a solid Linux background, with kernel maintainers and testers providing the latest improvements in the kernel to other developers to use. Various Linux distributions are supported by now, including Ubuntu, Gentoo, OpenSUSE, Crux, and many more. This is possible because of the uniqueness of the platform. The fact that the MPC5200B is PowerPC based probably makes for better ties between developers across distributions than if the EFIKA would have been x86 based. As we will see later, it's not just Linux that is supported. There are, among others, OpenSolaris[1] , QNX[2] and MorphOS[3] developments as well.

Figure 1: EFIKA mainboard

The main point here is that even though every developer has his own ideas and own project, they all find common ground at some point due to the fact that they all use the same base. A solid base is what is needed to make a community succeed, which in turn is needed to make these projects succeed. The goal of powerdeveloper.org is to bring like minded people together and develop communities, both of users and developers, in which ideas are exchanged and imagination turned into reality. This same principle can be applied to education: get a solid base on which students can fall back onto, and they will find each other eventually in order to start from common ground in project activities.

Education in the context of this paper goes far beyond the traditional classroom lectures and laboratories. Education also means bringing technology to places where the Internet and computers are not common tools yet, and provide the training for those people in order to give them opportunities to improve their lives or change societies. This is a very demanding and some-

[1] An open development based on the source code for the Solaris Operating System, http://www.opensolaris.org
[2] Commercial real time operating system, http://www.qnx.com
[3] Non-UNIX based, multimedia oriented operating system, http://www.morphosppc.com

times even dangerous activity, which brings however a lot of satisfaction when these goals can be achieved.

The ideal would be to bring all those people together, both those in the Western world as those in the developing world, and get them to work together on common goals. The best of both worlds, to improve the situation of people in developing countries, but also to increase awareness in the West that third world does not necessarily mean without food or water. That way, communities can be build based on mutual understanding, not hindered by feelings of superiority or fear, but stimulated by productive input, and brought together across boundaries by technology.

3 Intermezzo: Communities

Before continuing, a little should be said about how a relatively small company can get so many developers and people from the industry interested in their product.

The key to success in any project involving a small or even start-up company in the IT sector today depends on two major factors: the first one is money, the second one is people. Money is nice to have, but not always readily available for marketing purposes. Most of these companies spend the majority of their budget on actually developing their product. The Internet however provides an ideal and powerful marketing tool which is relatively cheap and very effective if done right.

When working with Open Source software, the actual code is only of secondary importance. The main focus point should be the people who make the software in the first place. Getting developers interested in your product can get a chain reaction going. Word of mouth is extremely effective within developer communities. Therefor, providing a resource in the form of a simple forum can get a long way to establish and maintain good relationships with said developers. Often, they are working on the software in their free time; motivation to work on the implementation on your device can be as simple as just giving some key developers the hardware and documentation. It helps if the hardware is completely ”open”; not many Open Source developers like the idea of signing a non-disclosure agreement.

Of course for this to work out, it implies that good, personal communication is set up between the company leaders and the people in the open source community. Many developers hate waiting for a piece of documentation which they need in order to continue their work. Swift responses to emails, being active on the forum, and a repository of documentation on a website, which can be part of the forum, with all relevant documentation should definitely be considered.

To market the company itself to the outside world, the power of search engines should not be underestimated. Maintaining a blog about your company activities, using relevant keywords, can get you a very high page rank. This will get a lot of people to your blog, and hence increase the public awareness of your company name and product. Genesi uses both to great success, the forum and resources are located at http://www.powerdeveloper.org, the blog at http://bbrv.blogspot.com. Besides getting in touch with people from the outside, these tools help to maintain good contacts with people and developers on the inside whom can be widely separated geographically.

4 Open Client

Thin clients are nothing new. In fact, they can be seen as a re-emergence of the old terminal/mainframe set-up, only graphical this time. Increased server and network performance allow for the clients to connect to the server and run applications on this server, while the display of said application is send over the network to the user using the thin client. The EFIKA can be used as a thin client, but this does not really fit into the just enough computing paradigm. The CPU of the EFIKA is perfectly capable of running an entire Operating System, and some applications. Yet, it is not powerful enough to run heavy applications such as Open Office[4]. With a combination of a server in the background to run these heavy applications, together with the EFIKA running the Graphical User Interface and the lightweight applications, we can make sure that the use of the total available computing power is used the most efficiently. This means for example that a less powerfull server is needed to support a greater number of clients, or, that with the same server, more clients can be used compared to traditional thin client set-ups. The EFIKA in this case is referred to as Open Client, and can be bought from Genesi, equipped with a XGI graphics card and mounted in a black steel case, all passively cooled of course. A possible set-up like this is shown in Figure 2, where the server in this case is a Sun T1000 running Solaris.

Several different Linux images can be made available to the EFIKA clients in order to allow different flavours of Linux to be offered to users, or even allow for a separate Linux image per user. These Linux images are shared using Network File System (NFS). The server can act as a DHCP server for the Open Clients if needed, or just integrate with an existing network. Since multiple users could make use of the same image, and to make user management easier, all the information regarding the users are stored in a MySQL database running on the server. After the EFIKA has booted, users can log in onto the system, with their credentials being verified against this database. This allows for dynamic user creation and removal via a web interface which can add and delete database entries as well as file system entries, for instance creating and deleting home directories, and copying default configuration files to this directory.

The original set-up like this was intended for a laboratory environment at VAMK, Vaasa University of Applied Sciences, in Vaasa, Finland. This lab offers support for embedded programming courses, computer architecture and operating systems. It gives the opportunity for every student to play around a Linux environment while being able to program 'dangerous' applications which would not be practical on a dedicated server many people rely on. This kind of programs, such as kernel modules and device drivers to name a few, have the ability to crash a system. Also, creating heavy loads, zombie processes, lots of system calls, etc. and study their impact on a system can be done in this environment without the possibility of interfering with other people's work. Why not have a classroom or two with dedicated computers running

[4]Open Office is an Open Source alternative to Microsoft Office

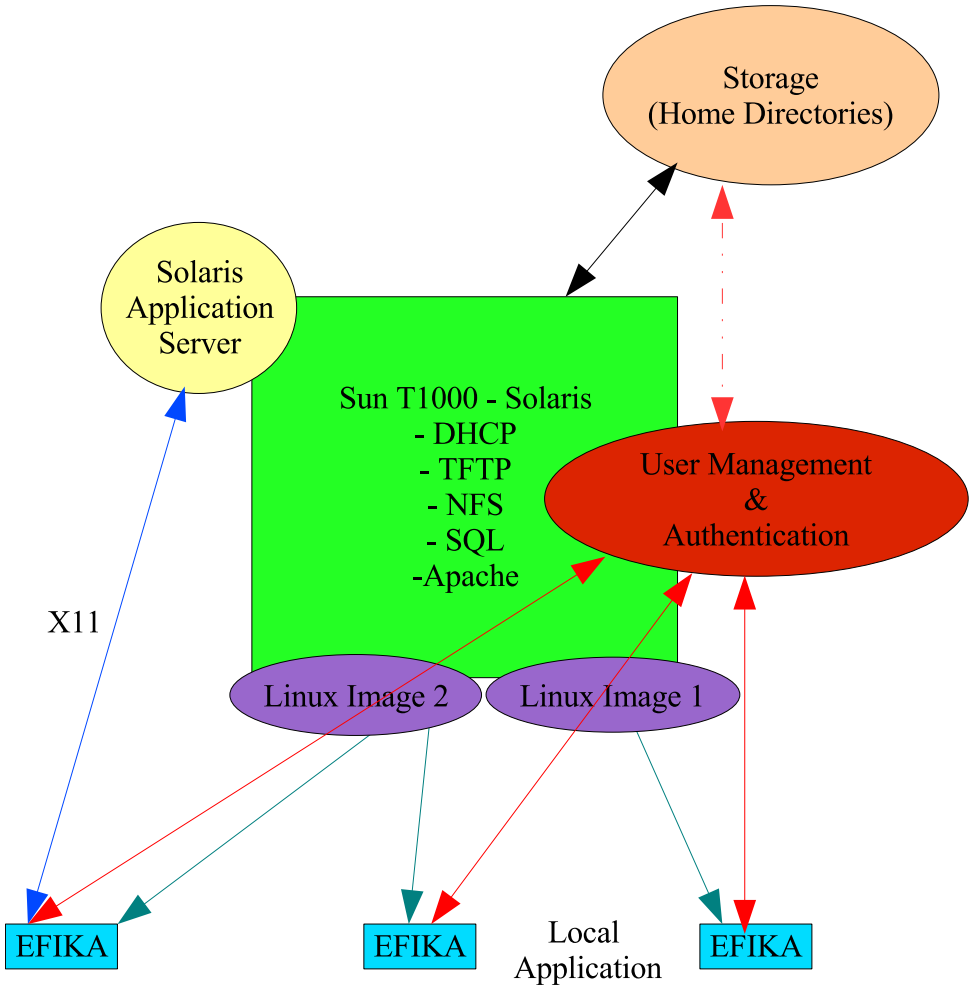

Figure 2: EFIKA clients connected to T1000 server

Linux? Many schools do not have the technical staff available to maintain extra classes with different operating systems. The maintenance on the system presented here is minimal, as the EFIKA itself has no hard-drive. For classroom sessions, a laptop could act as the server, and the existing mouse, keyboard and monitor in any PC room can be shared with the EFIKA if a separate room is unavailable. The dynamic user creation allows to assign different defaults to different students. Hence, exercises can be created and depending on who logs into the system, different assignment, tasks, or programs can be made available.

A different set-up could be made for a team of developers working with the EFIKA as a reference platform for their own custom implementations. Figure 3 below gives an idea of how this can work. As a side note, even though every set-up of this type discussed in this document has a SUN T1000 server running Solaris at the core, one is of course not limited to using this server or Solaris. A similar system can easily be made using a Linux set-up on different hardware. The T1000 however fits in with the EFIKA rather nicely, in that it is also low power, yet high performance.

The two cases below, show two different sides of the same coin. On one hand, we're dealing

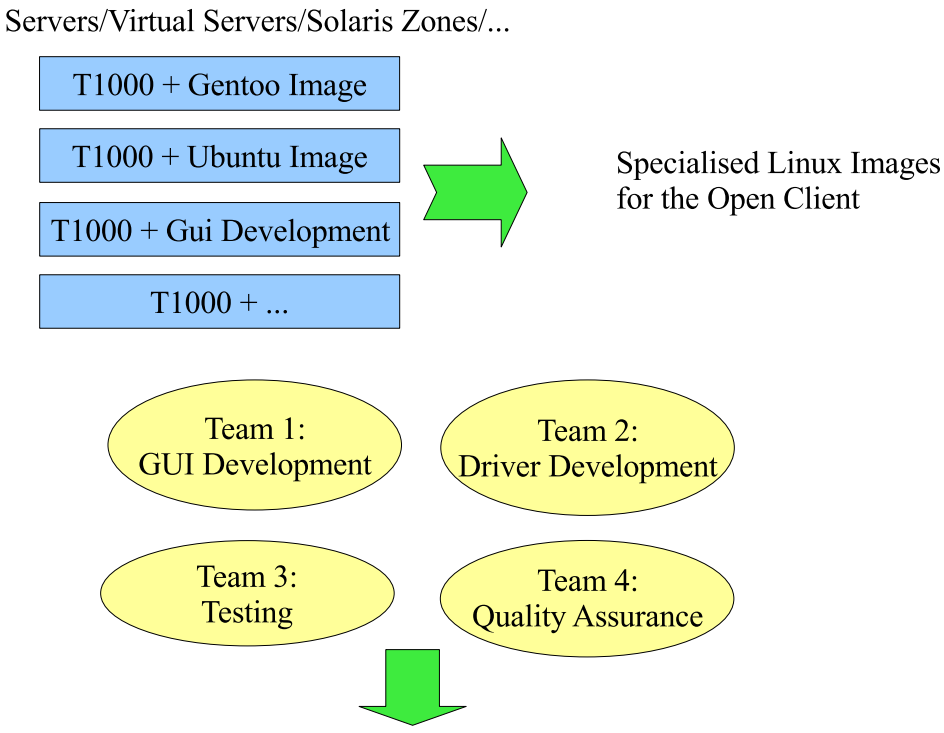

Figure 3: The developers' desktop

with a developing country, where Internet access is scarce and high tech equipment not readily and cheaply available like in Western countries. The other situation deals with a world renowned university which basically has an unlimited budget for high tech equipment and infrastructure. Yet, both use the same hardware and the same set-up for different purposes. This ability to create one system, which can adapt to a multitude of applications is a very powerful tool which saves development time and money, while reaching a wide install base with very specific custom requirements.

4.1 Case: Centre d'Accès Communautaire

The "Centre d'Accs Communautaire", or Community Access Centre in Sal, Morocco, is a space of communication, exchange and debate on the various aspects of the local development. Computers and other services are placed at the disposal of the public for various activities, in particular training in data processing and the services on line on Internet.

What makes this place so special, and what has it got to do with this project? Multiple EFIKA Open Clients, and a server, are now sitting next to the more conventional PC's at this centre. The goal is formation and education of youth, expanding open source software initiatives, reduce operation costs of the centre to make it self sustained and of course to let the kids have fun!

The centre is a space of communication, exchange and debate on the various aspects of the local development. Computers and other services are placed at the disposal of the public for various activities, in particular training in data processing and the services on line on the Internet.

This center was set up by Tanmia [2] and MTDS [3] (Morocco Trade and Development Services) which is a telecommunications consulting firm located in Rabat, Morocco. It is the premier Internet Service Provider in Morocco providing dial-up, leased line and DSL service throughout the Kingdom. The Tanmia development portal is a national Internet portal designed to increase the capabilities of Moroccan associations by utilising information technology and communication tools. It is a participation site where associations connect and exchange information. The EFIKA clients were donated to this project by Genesi. The server, a T1000, was sponsored by SUN. The whole project is funded in part by USAID with involvement of UNESCO.

There is a great interest from organisations such as UNESCO in that region because this region especially, but Morocco in general is a place where a common language is not so evident. The majority of the people speak their own Arab dialect, while Standard Arabic and French are the official languages in government and media. The problem is that many people do not read or speak Standard Arabic, which means the newspaper is useless to them. Indeed, Morocco has probably the lowest newspaper circulation in the area. To address this situation, efforts are underway by for instance UNESCO and Tanmia to give the people a voice in their own language.
Thanks to technology, so called podcasts are made in the native language of the communities. The communities there create the content for others to hear across the Internet. The EFIKA project there is helping to lower the technological cost to create and listen to podcasts, and, as the picture below shows the kids are having fun!

Currently, the second round of funding is in progress, with the intention to set up a similar CAC in Temara, Morocco, where the same set-up will be placed, again, next to conventional PC's.

Of course, there are issues when introducing a brand new system based on open source in a community which has, albeit limited, experience with Windows systems and might never even have heard of Linux or Open Source. Due this fact, it is imperative that a good presentation has to be made regarding the difficulties from switching to a somewhat familiar environment (Windows) to a completely (in their eyes anyway) alien environment. This is not only true for the people using the system, but also for those people managing the system at the CAC. Some suggestions can be made:

- It is important to make clear product specifications and a description about EFIKA with pictures of e.g. connectors used. The connection between T1000 and EFIKA has to be described very explicitly with all the details and limitations.
- It is also very be beneficial to include some pictures of the set-up as a whole (can be several options to demonstrate what is possible) as well as some pictures of the desktop (showing the applications that are used).
- Because this set-up is based on Open Source Software (OSS) it is also good to give a short presentation of the most important programs in use and how they compare to their Windows counterparts. Screen shots are a good idea in order to show this visually (making sure that as much and as precise information as possible is communicated across making the process/project transparent and efficient), but an even more useful tool in this aspect are screen capture movies with narration explaining what goes on.
- Teaching and explaining the system in person is probably the best way however, as this brings trust and transparency into the relationship. Some examples of personalised set-ups with description of the maintenance of the system is important to provide. Having these specifications makes the process more explicit and transparent which lowers the possibility for misunderstandings and gives more confidence to the customer.

4.2 Case: Center for Hellenic Studies

Imagine doing your studies in the field of Ancient Greek culture. You have a large library of Greek texts - naturally. However, you might also haves everal databases to consult? Online documents to print, such as publications? Save some important notes to a flash drive for safe keeping. While doing that, send some email to colleagues requesting information about some subject you are researching?

That's exactly what happens at the Center for Hellenic Studies (CHS), a research institute affiliated with Harvard University located in Washington D.C. A T1000/EFIKA Open Client set-up

is in place there right now, taking care of the tasks described above. Flash drives can be connected to the open client, CUPS takes care of the printing facilities and multiple printers are available to each Open Client. Open Office, Firefox, AbiWord, Gnumeric, Gftp for file transfer and Gaim for instant messaging are available. Even a little applet to keep track of the weather forecast, just in case you wonder if you will get wet later that day. These EFIKA Open Clients are installed at two floors of the library (where silence is very much appreciated, so passive cooling comes in handy!). Visitors can also use them to check email and browse the web. An Open Client is shown below installed at the library at CHS.

Harvard University is opening its first European campus, it will be located in Nafplion, Greece. There too, the EFIKA Open Client will be present to perform similar tasks, but also to give visitors and local communities Internet access.

Surprisingly, the same issues arise in this case, as in the CAC case. Even though the people have a more thorough understanding of technology, the Windows and Macintosh based systems are so ingrained, that the switch to a Linux and Solaris based system proved to be much more difficult than anticipated. Therefore, the same information required in Morocco goes for the U.S.A. as well.

4.3 Other Developments

The two cases mentioned above are just a small selection of what can be achieved with imagination, creativity and a cheap hardware/software platform. There are other uses however. Below,

a selection of applications currently in use or development is made.

A portable version, equipped with a TFT screen could make a great data logging device. Solar power could be used to power the EFIKA, for use during excavations, or desert explorations and geographical mapping. In fact, there could be many of them powered by wind and solar energy in several regions of the world where schools lack equipment, or where power outages are common. Then again, it could be turned into a portable media player, or even a very low power laptop. A working prototye is shown in Figure 4.

Figure 4: Portable EFIKA

Closer to home, consumer Network address Translation (NAT) and Firewall devices could be assembled and set up in order to decrease possible malware. The use of flash based storage would make for a completely silent and very power efficient device, consuming around 2 to 4 watts total. You could add some storage and turn it into a small home server to store files, music, pictures, or even host a website - you could leave it on 24 hours a day and not notice it on your power bill. The same principle could work to create an access point for Internet services, an educational podcast repository or educational audio and video streaming system in a developing country.

In combination with MorphOS, a very lightweight, multitasking, multimedia oriented operating system, a media player can be build which even allows for high definition video (720p) playback. This turns it into a nice player which connects to a TV, and streams content from a server from another room. It could be a device for art galleries to play back video art.

National Instruments is making a nice package [4] with LabVIEW together with the EFIKA and QNX. This system could be used for educational purposes, but it also could be the base for

a home nursing system whereby the EFIKA can send medical data straight to the doctor, shown in Figure 5.

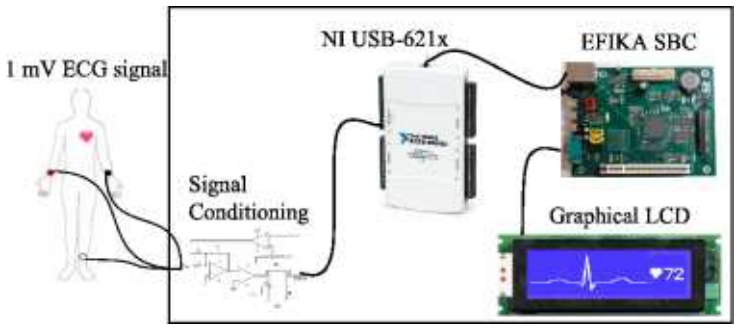

Figure 5: LabVIEW, QNX and EFIKA

Think about setting up a network of EFIKA computers to create a peer-to-peer network to establish a free speech and press movement within a dictatorship where Internet might not be available, or would be highly censored and monitored.

The list goes on: high-performance transparent network filters, advanced Linux based routers, gateway/hub for intelligent home wireless sensor networks, disaster relief network infrastructures (ad-hoc wireless mesh deployment), etc.

All the systems mentioned just now are in development, or have already been deployed, at this very moment.

5 Firmware

Besides Linux, the EFIKA supports operating systems such as OpenSolaris, QNX and MorphOS. A wide variety of other operating systems can be supported rather easily thanks to an advanced firmware which takes care of hardware initialisation. The support of as much operating systems as possible can vastly increase the user and developper base of the platform, and therefor increase the brand recognition and of course chances for success.

One of the key points is hardware support; you want to use off the shelf components wherever possible. Back in the days when Apple was using PowerPC, they had to make agreements with graphics card manufacturers in order to make compatible hardware. The reason is that every graphics card contains a piece software used for its initialisation. Of course, the PC's x86 is the most common architecture for these cards, so this code is written for x86. A PowerPC chip cannot run this, and thus cannot initialise the graphics card. The cards that were build for Apple were more expensive then their counterparts on the PC, and were at times not up-to-date with the latest and greatest on the PC.
The MPC5200B is PowerPC, yet we can use a standard PC graphics card with it. Since a PC graphics card requires a piece of internal x86 code to be executed in order to initialise the card, how can the PowerPC run the x86 code? This is where the Firmware steps in, as it contains a little x86 emulator which makes this possible. This in itself can greatly reduce cost and increase lifetime and expand application field. The same principle works for keyboards, mice, USB flash drives with a variety of filesystem supprt, etc.

What follows is a short overview of the evolution of different firmware in general, up until the current edition of the Genesi firmware, called AURA.

Figure 6 is the old traditional way of providing a board support package for the operating system. This implies a lot of work if one wants to migrate to other hardware, even when some of these hardware upgrades are minimal.

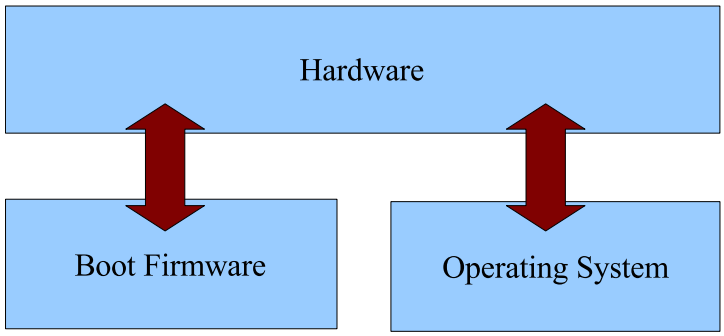

Figure 6: Board Support Package Firmware

Figure 7 shows the traditional UBoot method of setting up the hardware. Minimal information about the hardware in question can be read to make starting the actual operating system easier.

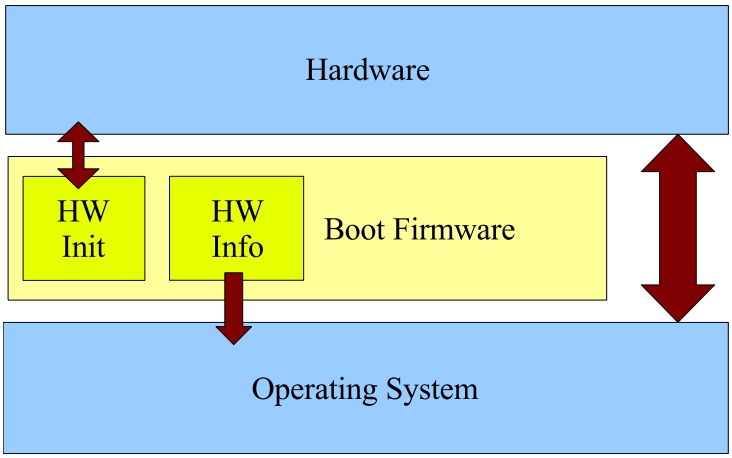

Figure 7: Traditional Firmware (e.g., U-Boot)

Figure 8 shows the Common Hardware Reference Platform firmware, as used by the PowerPC Macintosh clones before Steve Jobs returned to Apple and disallowed the creation of Macintosh Clones. Still, the IEEE 1275/CHRP firmware was a major step in the right direction to make it easier for operating system maintainers to migrate across hardware revisions and updates.

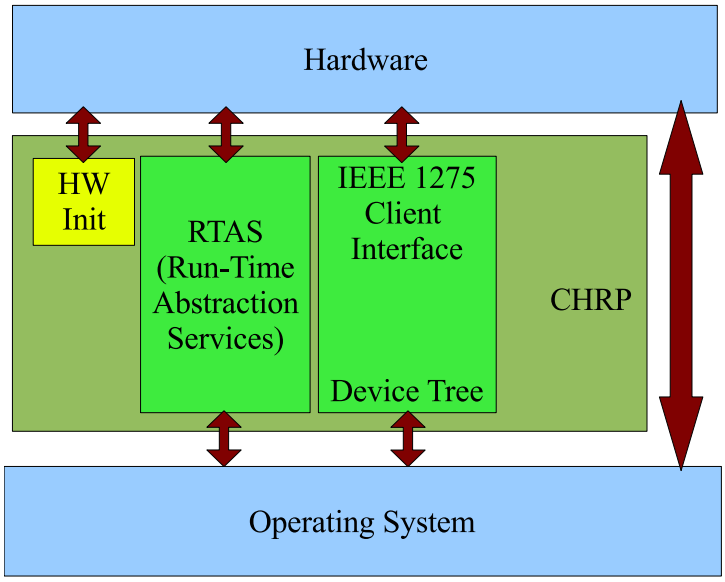

Figure 8: IEEE 1275/CHRP compliant Platforms

Finally, Figure 9 shows the Genesi firmware in its current implementation. Based on Open Firmware, developed by Sun Microsystems, this firmware allows for the easy migration of operating systems to completely different hardware architectures. Porting from one platform to another becomes a matter of days if not hours, instead of weeks and months, thanks to being based upon an interactive programming language, Forth.

Because the Open Firmware Forth code is compiled into FCode (a bytecode) and not into the machine language of any particular computer architecture, Open Firmware code included in for example, an I/O card can be executed by any system that uses Open Firmware. In this way,

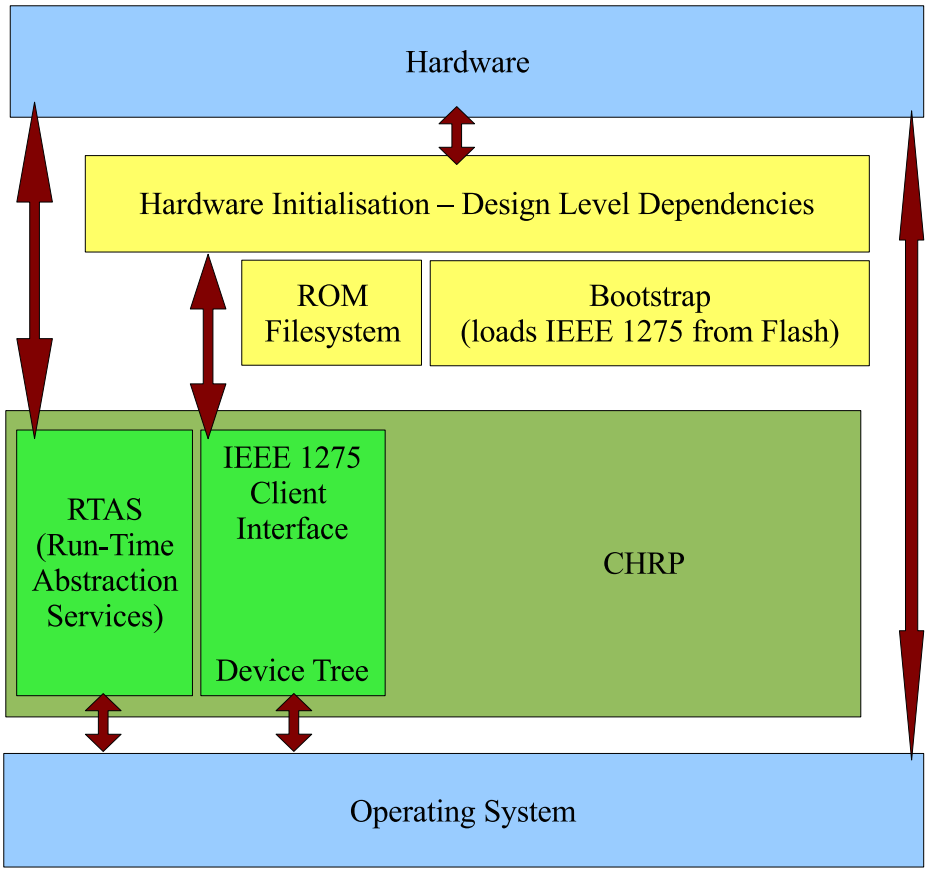

Figure 9: Genesi Firmware - AURA

an I/O card can provide boot-time diagnostics, configuration code, and device drivers that will be usable on any system running Open Firmware, allowing many of the same I/O cards to be used on say the EFIKA and SUN systems. Genesi licences this firmware to third parties as a way to decrease time to market, and to have and almost immediate support of several operating systems available.

6 Conclusion

This paper has shown a wide variety of applications of the MPC5200B based EFIKA single board computer. The goal was to show that low power consumption, high performance embedded chips can be used outside its initial target market, and into a wide range of devices and set-ups which can provide for general computing needs which were only recently still strictly the realm of desktop PC's. Thanks to advancements in embedded systems in general, these capabilities are available for a low price which makes these very attractive for educational purposes and infrastructure projects both in the West, as well as in developing countries. These are the core concepts behind Just Enough Computing.

The key to success however, is the community created around the EFIKA, with users and developers working together to improve and expand the possibilities of the system. Couple that with the fact that consumers are becoming more socially aware and concerned about the environment and are also becoming more economically savvy and also that high power, expensive computers that can run Quake 4 at 100 frames per second are not economical in developing countries, the EFIKA might be one of those products that can find a market both here in the West, as well as in developing countries. The next generation EFIKA is already on its way, and promises an even higher integration, more performance, yet the same power efficiency, which will increase its application field and will see even more exciting developments.

References

[1] “Genesi U.S.A., Inc.” http://www.genesi-usa.com .

[2] “Tanmia.” http://www.tanmia.ma .

[3] “MTDS.” http://www.mtds.com .

[4] “LabVIEW, QNX and EFIKA.” http://zone.ni.com/devzone/cda/tut/p/id/6208 .