PORTABLE ELECTROPHYSIOLOGIC MONITORING BASED ON THE OMAP-FAMILY PROCESSOR FROM A BEGINNERS' PROSPECTIVE

**K. Mankodiya[1], *S. Vogt[1], A. Kundu[1,4], M. Klostermann[1], J. Pohl[1], A. Ayoub[2], H. Gehring[3], U.G. Hofmann[1]*

[1] Institute for Signal Processing, University of Luebeck, Luebeck, Germany,
[2] Institute of Neuroendocrinology, University of Luebeck, Luebeck, Germany
[3] Department of Anesthesiology, University Medical Center Schleswig-Holstein, Campus Luebeck
[4] Center for Sensory-Motor Interaction (SMI), Department of Health Science and Technology, Aalborg University, Aalborg, Denmark
Email: hofmann@isip.uni-luebeck.de

[*] Authors contributed equally

ABSTRACT

We present here the experiences made on taking some first steps into the world of dual-core embedded processor technology. The processors we consider here consist of one DSP core and one general-purpose ARM core as are commonly used today by large mobile phone manufacturers. We describe some beginner's problems we encountered with Texas Instruments' OMAP platform and suggest some ways to avoid these common pitfalls for others. As an example application, we also introduce a new portable biosignal monitoring device that was the motivation for using a dual-core embedded processor in the first place.

Keywords— *OMAP, DSP, ARM, Embedded Linux, Biopotnetial Recoding, BeagleBoard .*

1. INTRODUCTION

Aging societies in today's world are beginning to face the severe challenge of sustaining their health maintenance level despite increasing costs of state-of-the-art treatments for a growing number of old patients. Providing affordable and high-level care is one of the biggest issues in public health care systems today [Braecklein, Dehm et al. 2007]. At the same time, the older generation has never before had such huge amounts of personal income at its disposal, willing to track their personal well being and health beyond governmental support. Also, a growing number of life-style oriented individuals is beginning to use medical monitoring devices for recreational use in sports and daily life.

Both aspects of the current societal situation funnel into one big and exciting technological conclusion: a clear demand for mobile, yet powerful monitoring systems exists, serving both the patient in need by maintaining qualified but remote supervision ("Telemedicine"), as well as the life-style and fitness oriented, financially independent person.

Both aspects particularly require the acquisition, fusion and analysis of a large number of physiological parameters by a light-weight, portable and power-saving device that is able to give qualified and autonomous warnings or suggestions to improve the level of fitness of its bearer. In other words, personalized health monitoring systems might in the future become as ubiquitous as current cell phones are today.

The following study reports on some of the steps we have taken to develop a less general, but nevertheless promising predecessor of mentioned future systems with the clear-cut target towards perioperative monitoring. In fact, the system's aim is to monitor several physiologic parameters, to ensure cerebral viability during open-heart surgery and cardiopulmonary–cerebral resuscitation [Roach, Kanchuger et al. 1996; Bokesch, Izykenova et al. 2006].

The project is to develop a compact system to record, analyze and display multiple electrophysiological channels like ECG, EMG and EEG and integrate them with available other monitoring sources, like NIR sensors, pulsoximeter or capnometers. In addition, this multimodal cerebral monitor (MCM) should not only passively gather data from its sources, but instead should analyze these signals and provide the anaesthesiologist in charge with a helpful interpretation of the data on the brain integrity of the patient monitored (see Fig. 1).

All these requirements, together with the requested miniaturisation and power economy called for computing technologies found in existing telecommunications applications.

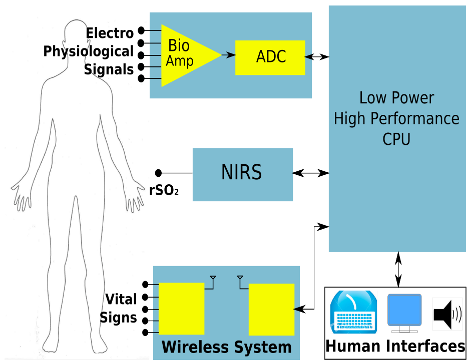

Fig. 1: A conceptual diagram of the data acquisition system of our Multimodal Cerebral Monitor.

We consequently turned to Texas Instrument's (TI) dual core Open Multimedia Architecture Platform (OMAP), found in high-end cell-phones as we were intrigued by their specified performance. The following gives a short report on the development of the MCM and the particular challenges we faced despite using commercially available components.

2. SYSTEM DEVELOPMENT

2.1. PHASE I: OSK5912 - AN OMAP2 BASED SYSTEM

As can be seen in Fig.1, the hardware of such an MCM system consists of several wired electrophysiological recording channels, some wireless sensor recipients and a central processing unit, incorporating software for smart analysis of data. In order to reach a short time to market, we started to design our target application on off-the-shelf development boards.

For the projected electrophysiological channels, a combination of custom made active electrodes and a commercial biopotential amplifier array serve as a signal conditioning unit and convey the processed electrophysiological signals to an ADS1258 analog to digital converter (Texas Instruments, USA). After digitization, the data enters into the dual-core OSK5912 evaluation module (OMAP Starter Kit, Texas Instruments, USA), which can be seen as the heart of the system. The OSK comes with an OMAP5912 processor and a sufficient number of fast I/O channels (see Fig. 4).

2.1.1 Front-end Electronics

Biopotential Amplifier – RHA1016
Our chosen biopotential amplifier RHA1016 (Intan Technologies, LLC, Salt Lake City, USA) contains 16 differential amplifiers with programmable bandwidths intended for bio-instrumentation monitoring and recording applications. A multiplexer in the RHA1016 permits all 16 channels to be sampled at rates as high as 30 kilosamples per second per channel (see Fig. 3) [RHA1016 datasheet]. All channel signals are individually amplified by a factor of 200 and funneled internally into a 16-to-1 multiplexer (MUX) stage. The multiplexer can output differentially amplified and multiplexed signals at the rate of up to 500kHz, controlled by an external 4-bit Channel Select signal.

Electrophysiology recording was performed with the RHA1016 evaluation board (Intan Technologies, USA) to validate its use for biopotential recordings. Exemplary, ECG recording was conducted by placing two electrodes on the chest and a third one on the elbow for reference. The evaluation board of RHA1016 received the ECG signals at its input and gave a 200 times amplified signal at output pins. The output signal was passed through an INA114 instrumentation amplifier to avoid grounding problems. The ECG obtained in this experiment is depicted in Fig. 2 as an oscilloscope output.

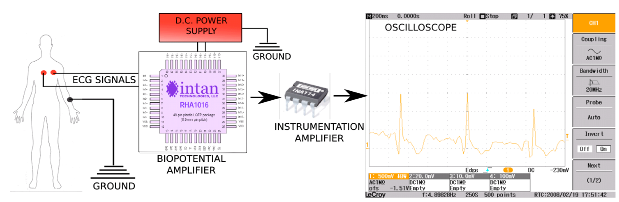

Fig.2: An experiment to see the credibility of the RHA1016 biopotential amplifier for ECG signals

Analog to Digital Converter - ADS1258
Amplified and multiplexed biosignals enter into an ADS1258 analog-to-digital converter. We chose an ADS1258 evaluation board to provide single-cycle settled data at channel scan rates from 1.8k to 23.7ksamples per second (SPS) per channel [ADS1258 datasheet]. Since the ADS1258 features its own 16-to-1 input multiplexer, we avoid double multiplexing by feeding the RHA1016-Out signal directly to the input at ADCIN (see Fig. 3).

As can be seen in Fig. 3, the ADS1258 is controlled by an industry standard serial peripheral interface (SPI) from an external source, which in our case was the OSK5912.

2.1.2 OMAP Starter Kit - OSK5912

The OMAP5912 is a highly integrated hardware platform containing two processor cores. This dual-core architecture consists of one TMS320C55X DSP core and one reduced instruction set computer (RISC) technology ARM926EJ-S ARM core.

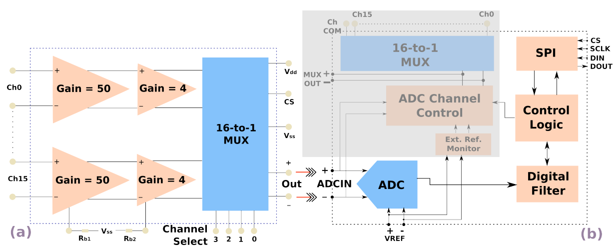

Fig. 3: A block diagram of the RHA amplifier [RHA1016 datasheet] (a) and the A/D ADS1258 (b). 16:1 Multiplexed signals from RHA1016 bypass the multiplexer of ADS1258 and are directly connected to the ADC circuit via ADCIN pins [ADS1258 datasheet]

The DSP achieves high performance by a high amount of parallelism at low power dissipation [OMAP5912 Manual]. It has an addressing range of 24–bits, data memory of 80K-Bytes, Memory Management Unit (MMU) and Translation Look Aside Buffers (TLB). A central 40-bit arithmetic/logic unit (ALU) is supported by an additional 16-bit ALU. The ALU is split into two 16 bit modules and is able to execute two instructions per cycle. The DSP core also contains a 24K-Byte instruction cache to minimize external memory accesses, improving data throughput and conserving system power. The CPU supports an internal bus structure composed of one program bus, three data read buses, two data write buses, and additional buses dedicated to peripheral and direct memory access (DMA) activity [OMAP5912 Manual].

The ARM926EJ-S is a 32-bit little-endian processor core that performs 32-bit or 16-bit instructions and processes 32-bit, 16-bit, or 8-bit data. It has an instruction cache, data cache, and a write buffer, which minimizes the external memory access time. In general, these features are transparent to program execution [OMAP5912 Manual].

In our application, the DSP core is responsible for collecting digitized data from the A/D converter via SPI and analyzing it, while the ARM processor is used for getting the data from the DSP core, storing it, and sending it to a connected LCD screen for display. Fig. 4 shows an overview of the actual OSK and its sub-components. Four expansion connectors are provided for connecting the external hardware that we need.
On the software side, TI has developed a universal framework for portability of algorithms called the eXpressDSPTM Algorithm Interoperability Standard (XDIAS) and an accompanying software architecture to support it, which is called Reference Framework 6 (RF6). The Reference Framework is a suite of source code examples to demonstrate DSP usage for typical applications. Reference Frameworks 6 (RF6) leverages DSP/BIOS, DSP/BIOS Link, the Chip Select Library (CSL), and XDAIS. RF6 is intended to enable designers to create applications on systems containing a DSP connected to a General Purpose Processor (GPP) i.e. the ARM [Mullanix, T. et al., 2004].

Above features made the OSK quite appealing for our needs, so we began development using this platform.
Unfortunately, despite all our efforts and short of subscribing to a commercial support contract, we were not able to install or recompile the only recommended Monta Vista Linux OS for the ARM on any halfway current desktop linux installation (MontaVista Linux 3.0). Thus we were not able to harvest the OSK and its promising dual-core features. As it turned out, the OSK was considered a

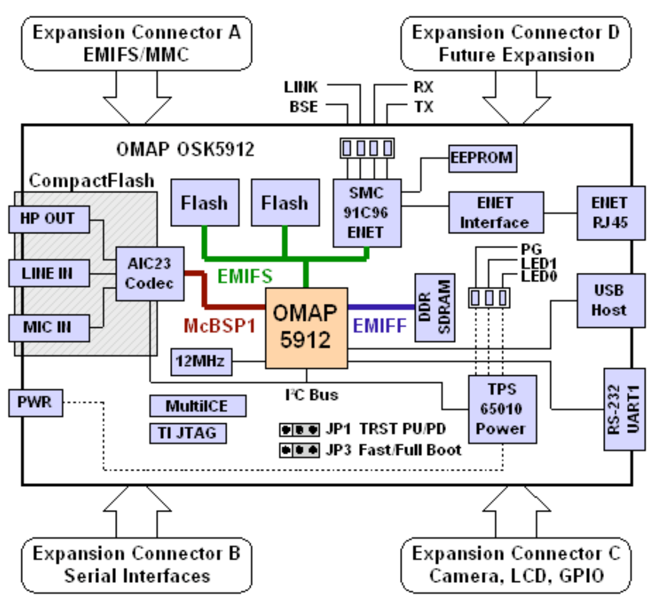

Fig.4: OSK5912 layout [OMAP5912 Manual]

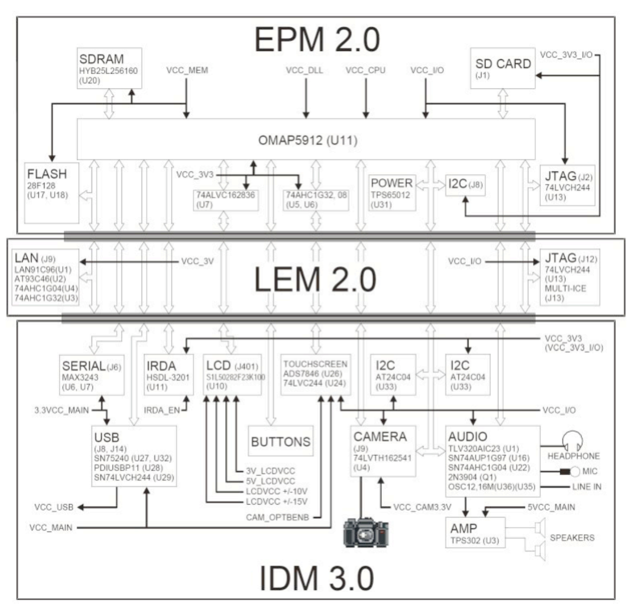

Fig.5: LDK5912 block diagram [LEOs DSP Manual]

„non-catalogued“ TI product and was thus only supported by external, commercial consultants.

Due to the lack of an active online community for OSK development, we finally signed a costly support contract, which did help us a lot with many linux problems such as setting up a TFTP server and configuring a u-Boot kernel. Also, the new development environment supplied by MontaVista finally did run on current versions of Ubuntu and SUSE desktop linux, but was geared more towards ARM application development and complete system design.

It should be noted here, that it seems, contrary to normal desktop development platforms, embedded development software is still often just put together once for a particular piece of development hardware and then never touched again during the whole life cycle of the development board product. When we were looking for updates to the supplied support software of newly purchased OSKs, the newest updates we could find were already 3 years old and had not been changed since. Some software such as DSP/BIOS Link (now called just DSPLink) has indeed seen continued development, but all support for the OSK platform has been taken out of the current code a long time ago.

We therefore strongly recommend a person-to-person enquiry about any considered TI product before purchasing it for academic use. TI now keeps a list of development hardware that is said to be fully supported and recommended for universities at http://www.ti.com/europe/university .

Because of the OSK's status as non-supported development platform and also its lack of a built-in method for graphical user interface display, we abandoned the original TI OSK and switched to a third-party OMAP5912 product by Empower Technologies (Richmond, Canada) as we were hoping to re-use our OMAP5912-specific code for communication with the ADS1258 A/D converter.

2.2. PHASE II: LDK5912 - AN OMAP2 BASED SYSTEM

The LEOs Development Kit (LDK5912) by Empower Technologies consists of the dual core OMAP5912 along with a small colour LCD touch screen and the same number of input–output peripherals as the OSK. Dwarfing TI's original OSK, the LDK already runs a complete Linux kernel–2.6 along with TI's DSPLink (version 1.30) for OMAP5912. So, it is a lot closer to real useable hardware, than the OSK. It is officially supposed to be used for ARM-side application development only, with DSP programming being considered an "advanced use" by the manufacturer.

Fig. 5 depicts the three hardware modules of LDK5912. The top module, called EPM is mounted at the back containing a flash storage device, power supplies and the OMAP5912 processor. The middle module, dubbed LEM contains network connections, while the bottom module IDM contains human interfacing peripherals, like the touch screen and audio codecs.

In our application, software has to run on both the ARM and the DSP cores and is intended to distribute the work done by each single core of the OMAP5912. Empowers LEOs was used to develop an application program running on the ARM core while TI's DSP/BIOS Link and RF6 were used to develop routines running on the DSP core. Thus, the ARM core serves as a host providing human interaction and connecting to the DSP [Mullanix, T. et al. 2004].

2.2.1 ARM development environment

The development of the ARM is done on a desktop linux platform using Ubuntu 7.10. The LDK5912 ships with a pre-compiled LEOs kernel and JFFS2 file system preinstalled in the onboard flash so that the system can run out of the box. The LEOs Development Environment can be installed in Ubuntu 7.10 Linux workstation using precompiled binaries [LEOs SDK Manual].

LEOs is comprised of a Linux 2.6 Kernel, drivers for the LDK5912 peripherals, hardware power management, TI DSP/BIOS Link support, LEOs Message Queue System with support for the LEOs proprietary GUI, software development tools, software applications (including soft keypad and handwriting recognition), system applications and multimedia applications [LEOs API Manual].

An application is launched on the LDK when the user taps the specific icon or presses the necessary button. The launch command initialises the GUI of that application. The button tap generates a GUI message (GUI_MSG). Information about the occurrence of any event and its context is recorded in the GUI_MSG data structure and then added to the message queue. The messages are checked all the time before they get processed. If the message fails to deliver properly it is sent back to the structure and then processed again. The processed messages are then passed through another logic loop which checks whether the user has sent a quit message or not. If there is a quit message, the GUI closes, ending the application. The user can also close an application without going through the above process. If the exit button of the GUI is pressed after launching the application, the GUI closes immediately and hence halts the application.

2.2.2 DSP development environment

Programming a DSP application to run on LEOs is considered an advanced use of Linux Digital Appliance (LinuxDA, Empower Technologies Inc., Canada) and LEOs. Code for the DSP is built using TI's Code Composer Studio (CCS) Integrated Development Environment (IDE) and associated DSP/BIOS real–time operating system. CCS connects to the LDK5912 EVM through a JTAG emulator. Programs can be written in either "C", "C++"or in assembly language. For development of the DSP, RF6 and DSP/BIOS Link are used to build programs. Using these packages is

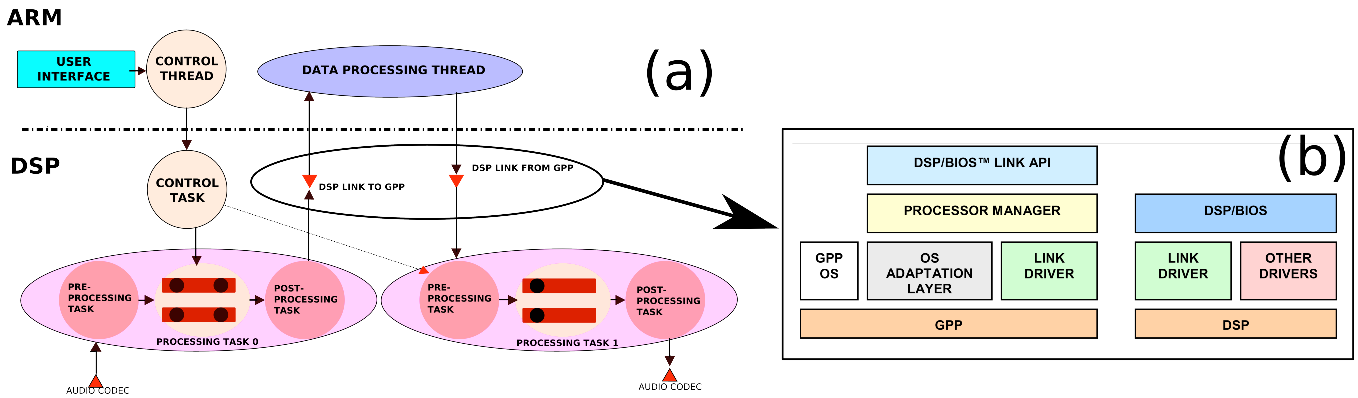

Fig. 6: (a) The diagram shows how code following the reference framework 6 (RF6) processes an incoming stereo audio signal on the DSP, sends the data to the ARM, where some processing can be done and then sent back to the DSP for further processing. [Mullanix, T., et al., 2004] (b) Software Architecture of DSP/BIOS Link, which is the foundation software for the interprocessor communication [TI DSP/BIOs Link 2005]

believed to minimize code development complexity [LEOs DSP Manual].

2.2.3 ARM-DSP Communication

The LEOs provides users and developers the option to utilize both ARM *and* DSP of the OMAP architecture. In general, the all important DSP–ARM communication can be established with three modes: (1) *DSP Gateway*, an open source software originally developed by Nokia, that uses Linux "device files" to send data to the DSP from the ARM side. (2) A *DSP Bridge* is often a highly specialized, high-level library used for DSP–ARM communication which provides very high level transfer functions and makes it difficult to reuse it in different applications (3) *DSP/BIOS Link* provides a more general layer of abstraction from the information transfer between the two CPU cores, making it easy to reuse across different platforms [TI DSP/BIOs Link 2005].

To understand the working behavior of an RF6 application, we take a look at an example for processing audio input in Fig. 6 a). The user initiates the ARM to boot the DSP via a user interface e.g. a GUI command. The ARM then commands the DSP to start acquiring data from the audio codec. The DSP itself processes this acquired data and passes them to the ARM. The ARM copies the data and sends it back to the DSP. Before sending this data back to the DSP, the ARM has the option to process this data if required. The ARM can again set some parameters in the DSP like filtering etc. So after further processing of the data, the DSP sends the signal out through the codec output.

The data processing elements in RF6 are tasks, channels, cells and XDAIS algorithms. XDAIS algorithms such as volume algorithm are wrapped by a cell in order to be used by the RF6. A group of cells makes up a channel. Furthermore, a collection of channels is processed by a task.

Software architecture of DSP/BIOs Link

DSP/BIOS Link is the foundation software for the inter–processor communication between ARM–DSP. It provides a generic API that abstracts the characteristics of the physical link connecting ARM and DSP from the applications. [TI DSP/BIOS Link 2005]

On the ARM side (see Fig. 6 b)), the OS ADAPTION LAYER module sums up the generic OS services that are required by the other components of DSP/BIOS Link. During DSP–ARM communication, this LINK DRIVER is responsible for controlling the execution of the DSP and data transfer. The PROCESSOR MANAGER maintains a book–keeping record for all components. The DSP/BIOS Link API is interface for all clients on the ARM side [TI DSP/BIOs Link 2005].

On the DSP side (see Fig. 6b)), the LINK DRIVER specializes in communicating with the ARM over the physical link. There is no specific DSP/BIOS Link API for the DSP. The communication is done using the DSP/BIOS modules, namely SIO/GIO/MSGQ [TI DSP/BIOs Link 2005].

2.2.4 LDK5912–ADS1258 communication

Since our application needs to acquire signals from an A/D converter, we needed to replace the audio codecs in Fig. 6 a) with code for communicating with the ADS1258. Empower Technologies officially supports only UART serial communication with external devices and strongly discourages use of the existing samtec^tm connectors for chip-to-chip serial peripheral interface (SPI) connections. But as the UART standard is meant for longer range device-to-device connections via cable and is not designed to support fast chip-to-chip communication, and we had also already developed working code for SPI communication between our A/D converter and the OMAP5912 chip, we decided to try accessing the LDK's OMAP5912 hardware pins to use for SPI communication by creating a purpose-built samtec plug to connect to the existing samtec socket.

This communication between the LDK5912 board and the A/D converter was done from the DSP side using the Code Composer Studio (CCS) development environment. In our application, the A/D converter is used as Slave and the LDK5912 as Master during SPI communication via the LDK's Multi channel Buffered Serial Port (McBSP). This means that the A/D converter writes its bits to the LDK at a certain speed given by a clock pulse signal generated by the LDK and the LDK will receive each bit with the same frequency. The McBSP port is configured in "clock–stop" mode to provide compatibility with the SPI protocol. Fig. 7 shows the communication protocol between the SPI and McBSP. Communication between the master and the slave is determined by the presence or absence of the master clock. [TMS320c6000 2006]

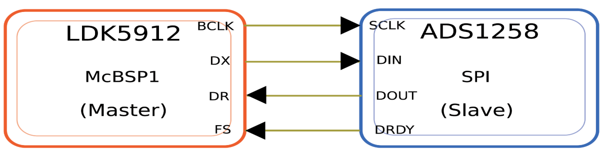

Fig.7: The LDK5912 is used as a "master" and A/D converter as "slave" during SPI over McBSP communication. [TMS320c6000 2006]

We were able to establish connection on this interface despite the discouraging advice of the manufacturer. Fig. 8 shows an oscilloscope plot, depicting a short communication episode on the SPI/McBSP interface between the LDK5912 and the ADS1258.

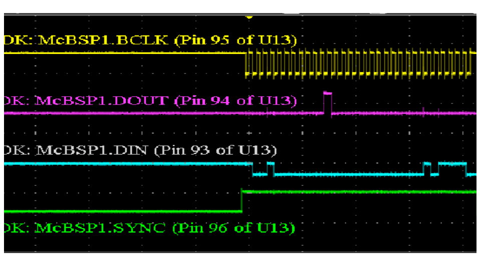

Fig.8: Scope plot of the communication between LDK5912 and ADS1258: 32 bit clock pulse generated (channel 1) . Channel 2 depicts the output sent by LDK5912 to ADS1258. Channel 3 represents the output generated by the ADS1258 and received by the LDK5912, while channel 4 is the synchronization for software interrupts.

We finally achieved the goal of bringing electrophysiologic analog signals into our portable monitoring device, as can be seen in Fig. 9, despite the fact that manufacturer support systems in microelectronics seem not to be geared up to support academic low-cost projects.

As by now it has become clear that the OMAP5912 is finally going out of production and neither the OSK nor the LDK will be continued to be shipped forever, we are currently once more transitioning our development efforts to a new state-of-the-art development hardware for future use. This time, we are not just changing to a new development board while keeping the same processor, but are transitioning to a new generation of mobile dual core processors of the third generation of OMAP chips.

The chip we recently started to use in a number of student development projects is the OMAP3530,

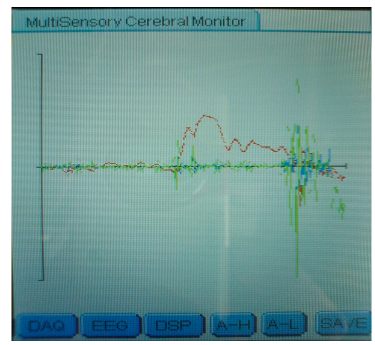

Fig.9: Screenshot of the LDK running our electrophysiological acquisition software for exemplary EEG data (red). The computing power of the OMAP processor is used to display realtime calculated wavelet coefficients by the à-trous algorithm [Dutilleux, P., 1989] (first detail blue, second detail green).

which contains a fast ARM core, a DSP core, and a graphics accelerator all on one die. This new generation of OMAP processors is officially supported by TI (it is "in the catalogue") and has a large and growing online community supporting it. As this generation of processors is still fairly new, it is probable that these chips can be used for another 3-5 years without loosing software support.

The OMAP3530 processor can be ordered as part of an official full-scale evaluation module (OMAP3530EVM) that provides most peripherals such as a numeric keypad and a TFT display for completeness.

A different interesting approach to development boards follows the idea of keeping the board as simple as possible and supplying only the most needed peripheral hardware mounted on the board itself. Such a system has been developed for the OMAP3530 by the open source community under the name the BeagleBoard (BeagleBoard.org, USA).

2.3. PHASE III: BEAGLEBOARD - AN OMAP3 BASED SYSTEM

Unlike OSK5912 and LDK5912, the BeagleBoard has an open source and freely supported operating system coming with a growing repository of working applications. The large and advantageous development community of BeagleBoard developers can often solve a problem faster than the technical support can do, although most community efforts are currently still aimed towards the embedded linux ARM part of the system. Questions about pure DSP programming may still be better guided at TIs official support.

On top of the lower price and larger and faster support availability, the BeagleBoard has another advantage: Its physical dimensions are just around 10x10 cm, making it more suitable for portable applications at a university.

The USB-powered BeagleBoard is a low-cost, fan-less single board computer based on a TI OMAP3530 applications processor that is said to reach laptop-like performance and integrates a 600MHz ARM Cortex-A8 core with a high-end 430MHz DSP-TMS320C64x core. The OMAP3530 supports high-level operating systems (OSs), such as Windows CE, Linux and others. [BB RM 2008]

As can be seen in Fig. 10, a USB OTG (On-the-Go) port exists on the BeagleBoard and has two modes: in client mode, the USB port lets the beagle board act as a common USB device that can be connected to a PC. In host mode, the beagle board becomes the main device and can use other USB devices as clients. USB mice, keyboards, network interfaces, and hard disks can be connected to the beagle board in this way. Besides new specialized fast SPI ports (McSPI) on the OMAP3530, there are also five McBSP ports present. McBSP2 provides a full-duplex, direct serial interface between an audio codec module inside the TWL4030 chipset. The video interface of the OMAP3530 is accessible through a DVI-D interface connector located on the board. The BeagleBoard contains JTAG, RS232, SVideo and expansion ports for different applications. Additional hardware can easily be connected via USB as mentioned above [BB RM 2008].

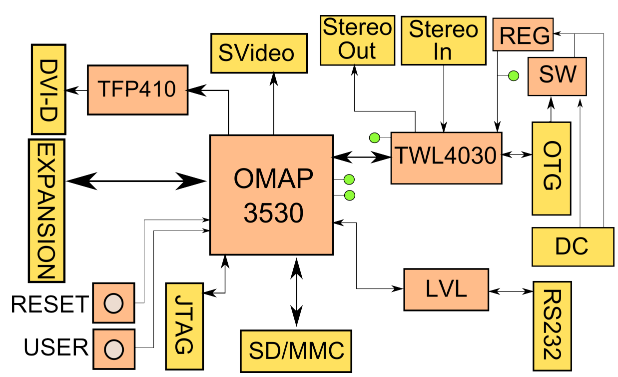

Fig.10: A BeagleBoard layout [BB RM 2008]

For our application, an expansion board to BeagleBoard (Fig. 11) is under development, which integrates the biopotential amplifier (RHA1016) with ADS1258 A/D on a single PCB. The main task of this expansion board is to perform the digitization and signal conditioning of many electrophysiological analog signals. A digitized form of these signals is then passed to the BeagleBoard as the heart of our Multimodal Cerebral Monitor through SPI communication.

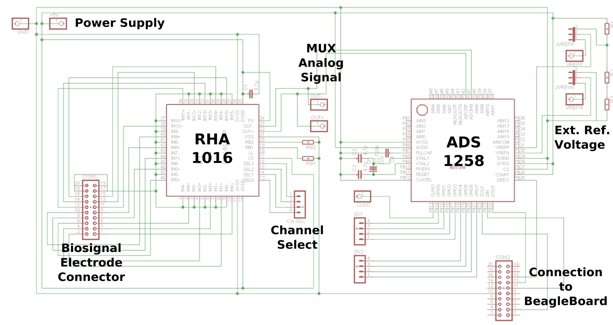

Fig.11: RHA and ADS on a single expansion board

3. RESULTS & DISCUSSIONS

Although the project presented here is work in progress, we are already able to perform signal processing tasks on a prototype platform of the BeagleBoard/ExpansionBoard platform.

Spike Detection Algorithm

A simple threshold-based detection algorithm for action potential like signals already implemented on the DSP as an example application. This application allows to detect online incoming spikes for later spike clustering and marks them by sending a rect signal. Fig. 12 shows a spike-like input signal, which is acted upon by generating a square pulse for as long as the incoming signal value is above a predefined threshold. This system is used to serve as a device-under-test in a Hardware-in-the-Loop test stand [Vogt, Klostermann et al. 2008].

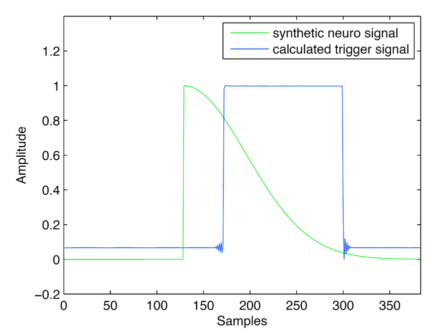

Fig.12: Simulated spike detection by online thresholding of incoming signals. Depicted is an output signal reacting on the simulated spike [Vogt, Klostermann et al. 2008].

4. CONCLUSIONS

We described our work on the way to implement a biosignal monitoring device in dual-core embedded hardware without profound knowledge of embedded linux or dual-core development. In spite of the many problems we encountered during our quest, we were able to use this hardware to our benefit and will continue to use it in the future.

We hope the experiences we learned the hard way will not be in vain as other beginners might avoid the mistakes we made.

ACKNOWLEDGEMENTS

This work was in part financed by the German Ministry of Research project BiCIRTS.

REFERENCES

- BB RM 2008: BeagleBoard System Reference Manual, Revision B.5, Sep. 2008
- Bokesch, P. M., G. A. Izykenova, et al. (2006). "NMDA Receptor Antibodies Predict Adverse Neurological Outcome After Cardiac Surgery in High-Risk Patients." Stroke 37: 1432-1436.
- Braecklein, M., J. Dehm, et al. (2007). VDE-Anwendungsempfehlungen für TeleMonitoring H. Korb. Frankfurt, VDE.
- Dutilleux, P. (1989). An implementation of the algorithm `a trous to compute the wavelet transform. Wavelets: Time-Frequency Methods and Phase Space, Berlin: Springer-Verlag, pages 298–304.
- LEOs API Manual: Empower Technologies, Inc. LEOs Software Development Manual Volume 2 : API, 2006.
-
- LEOs DSP Manual: Empower Technologies, Inc. LEOs Software Development Manual Volume 3 : DSP, 2006.
- LEOs SDK Manual: Empower Technologies, Inc. LEOs Software Development Manual Volume 1 : SDK, 2006.
- Mullanix, T., Vincent Wan, et al. Reference Frameworks for eXpressDSP Software: RF6, A DSP/BIOS Link-Based GPP-DSP System SPRA796A. Texas Instruments, April 2004.
- Thakor, Nitish V., Hyun-Chool Shin, et al. Quantative EEG assessment. hypothermic neuroprotection after cardiac arrest quantitative EEG assessment. IEEE Engineering in Medicine and Biology, 25(4):20–25, July/August 2006.
- OMAP5912 Manual: Texas Instruments. OMAP5912 Applications Processor data manual. Texas Instruments, December 2005. Literature Number : SPRS231E.
- RHA1016 Datasheet: Intan Technologies LLC, 2006, http://www.intantech.com/products.html .
- Roach, G., M. Kanchuger, et al. (1996). "Adverse cerebral outcomes after coronary bypass surgery. ." N Engl J Med. 335(25): 1857-63.
- TI DSP/BIOs Link 2005: Texas Instruments. DSP/BIOSTM LINK, November 22 2005.
- TMS320c6000 2006: TMS320C6000 DSP multichannel buffered serial port (McBSP) reference guide, Dec. 2006.
- Vogt, S., M. Klostermann, et al. (2008). Hardware-in-the-loop testbed for closed-loop brain stimulators. ECIFMBE 2008- Antwerp, IFMBE Proceedings ISSN