Energy Profiling of DSP Applications, A Case Study of an Intelligent ECG Monitor

*Dejan Raskovic, Emil Jovanov, Thomas Martin, Shuaib Hanief, Pedro Gelabert **

The University of Alabama in Huntsville
Electrical and Computer Engineering Department
Huntsville, Alabama 35899

[*] Texas Instruments
New Business Development
12203 SW Freeway, MS 701, Stafford, TX 77477

Contact address: raskovd@ece.uah.edu, phone: (256) 890 3492

ABSTRACT

Proper balance of power and performance for optimum system organization requires precise profiling of the power consumption of different hardware subsystems as well as software functions. Moreover, power consumption of mobile systems is even more important, since the battery is a large portion of the overall size and weight of the system. Average power consumption is only a crude estimate of power requirements and battery life; a much better estimate can be made using dynamic power consumption. Dynamic power consumption is a function of the execution profile of the given application running on specific hardware platform. In this paper we introduce a new environment for energy profiling of DSP applications. The environment consists of a JTAG emulator, a high-resolution HP 3583A multimeter and a workstation that controls devices and stores the traces. We use Texas Instruments' Real Time Data Exchange mechanism (RTDX™) to generate an execution profile and custom procedures for energy profile data acquisition using GPIB interface. We developed custom procedures to correlate and analyze both energy and execution profiles. The environment allows us to improve the system power consumption through changes in software organization and to measure real battery life for the given hardware, software and battery configuration. As a case study, we present the analysis of a real-time portable ECG monitor implemented using a Texas Instruments TMS320C5410-100 processor board, and a Del Mar PWA ECG Amplifier.

1. Introduction

Proper balance of power and performance for optimum system organization requires precise profiling of the power consumption of different hardware subsystems as well as software functions. Moreover, power consumption of mobile systems is even more important, since the battery is a large portion of the overall size and weight of the system. Average power consumption is only a crude estimate of power requirements and battery life; a much better estimate can be made using dynamic power consumption. Dynamic power consumption is a function of the execution profile of the given application running on specific hardware platform. Under some conditions the total energy available from the battery will be determined by the peak power of the system. Therefore, it is necessary to look at both the instantaneous and average power of the system to accurately estimate battery life.

In this paper we introduce a new environment for energy profiling of DSP applications. The environment consists of a JTAG emulator, a high-resolution HP 3583A multimeter and a workstation that controls devices and stores the traces. We use Texas Instruments' Real Time Data Exchange mechanism (RTDX™) to generate an execution profile and custom procedures for energy profile data acquisition using GPIB interface. We developed custom procedures to correlate and analyze both energy and execution profiles. The environment allows us to improve the system power consumption through changes in software organization and measure real battery life for the given hardware, software and battery configuration.

Proposed environment correlates the application execution profile with power consumption profile. Consequently, we are able to identify the specific hardware and software components whose power consumption has the most critical impact on battery life. Once these components are identified, we are able to optimize for the proper balance between performance and power consumption. For instance, power consumption of the compact flash storage subsystem may be reduced by software data compression, thereby reducing the number of write cycles to compact flash. This comes at the cost of increased CPU power consumption.

As a case study, we present the analysis of a real-time portable ECG monitor. A real time portable ECG development environment is implemented using a Texas Instruments TMS320C5410-100 processor board, and a Del Mar PWA ECG Amplifier. The amplifier is a dedicated three-channel ECG Holter amplifier with low power consumption. A set of electrodes and their leads are optimized for Holter applications.

Our environment for energy profiling of DSP systems allows us to explore the system design space and find optimum solutions for the given application. Moreover, we can measure system operation time for a particular battery family. Therefore, we can more accurately predict battery life in the presence of non-ideal battery behavior.

2. Power Efficient DSP Processing

For battery-powered DSP systems, power consumption is a critical design parameter. Power consumption is a determining factor in battery life and overall system size and weight. However tempting it may be to run a system at minimum power, this often leads to unacceptable performance. To properly trade-off power and performance requires that the work completed per battery discharge be maximized, once the real-time constraints are met (or in non-real-time systems, once the user's performance expectations are met).

Non-ideal battery properties require that the peak power of a system be measured as well as average power, if battery life is to be accurately estimated [25]. Under certain conditions, peak power consumption of a system determines battery capacity. Thus two systems with the same average power may have vastly different battery lives. As a consequence, to increase the amount of work that can be completed per battery discharge, designers must measure the dynamic power of the total system and be able to correlate the power measurements with an execution profile of the system.

3. Energy and Execution Profiling

3.1. Real Time Data Exchange (RTDX)

Monitoring and energy profiling of our target application is implemented using Texas Instruments Real-Time Data Exchange mechanism (RTDX) [27]. RTDX consist of two components. A small RTDX software library runs on the target. Using calls to this API library target program is able to pass data to or from it. JTAG interface is used to move data to or from the host platform. Those data transfers occur in real-time, with little overhead, while the DSP application is running.

Another component of RTDX is an *RTDX Host Library*, which operates in conjunction with *Code Composer Studio*. COM API is provided for displays and analysis tools to communicate with RTDX. This is the way to obtain the data from the target, or to send data to the running DSP application. Any standard software packet that makes use of COM APIs can be used, such as Microsoft Excel, Visual Basic, or Visual C++.

To establish a bi-directional connection between host and target, both input and output channels should be created. In our case, only the target will send data to the host, so only output channel was created and enabled for communication. After that, data can be simply written to that channel from the target side.

Real-Time Data Exchange can work in two different modes — continuous and non-continuous. First mode was used to measure battery data life, where the data (number of seconds elapsed) was sent from the target and captured by a Visual Basic application written for that purpose. This application reads data from RTDX, using COM interface, and displays it. When batteries run out, there will be no more data from the target, so the application will be displaying the last data received, i.e., the number of seconds until the target runs out of batteries.

Second mode was used for energy profiling and analysis. The data record (including time stamp, peripheral status, and program status) is captured and written into a log file. This file (.rpt file) was dumped into a text file and matched with multimeter readings. Both log file and multimeter readings are used by another Visual Basic application to analyze energy consumption and produce reports.

For the time stamp we use the low-resolution timer that is available on the DSP. A call to the function `CLK_getltime` returns the number of timer interrupts that have occurred. Since the interval between two interrupts is adjustable (with microsecond resolution), this can be used as a relatively precise timer. In the case of battery life measurements *Periodic Function Manager* is used to periodically execute a function that sends the same time-stamp. In that case, the timer registers were set to have this function executed ones per second, giving us more than accurate measurements of battery life.

3.2. DSP Application Profiling

Data sent to our Visual Basic Energy Profiling Application consist of two parts — the information about the procedure currently being executed and the information about current state of peripherals. Those two parts combined create the Status Word. That way, we are able to match energy consumption with the state of our application and our target board.

Generally, there are two ways to perform the profiling. One can periodically send the Status Word, and match this (on-line or off-line) with readings of a multimeter. There is a possibility, of course, that some procedures are “skipped” if the status word is sent infrequently, but generally, this is a good approach.

Another way is to send a status word at the beginning and the end of each procedure. In this case, time stamp should also be sent. That way, it is easier to calculate exact energy consumption, since the exact execution time for each procedure is known. The improvement of this approach would be block-level resolution, which would be justified in the case of long procedures with particularly important or energy-consuming blocks. Resolution could be even more improved if the program counter is sent instead of the information about the current procedure. We prefer to use this approach.

3.3. Energy Profile Analyzer

The DSP Energy Profile Analyzer is a Visual Basic application for analysis and visualization of measured data. Set of measurements could be represented using different views from the main application menu.

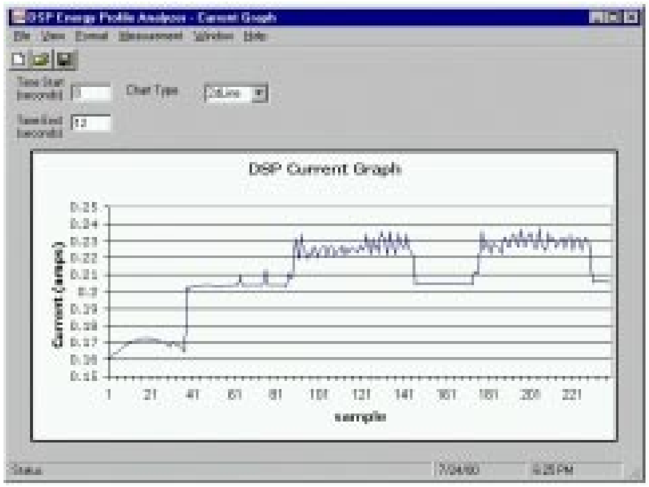

***Figure 1**: Current profile graph*

The current profile of a typical audio application is presented in Figure 1, peripheral activity graph is given in Figure 2, and distribution of the power budget between procedures is represented in Figure 3.

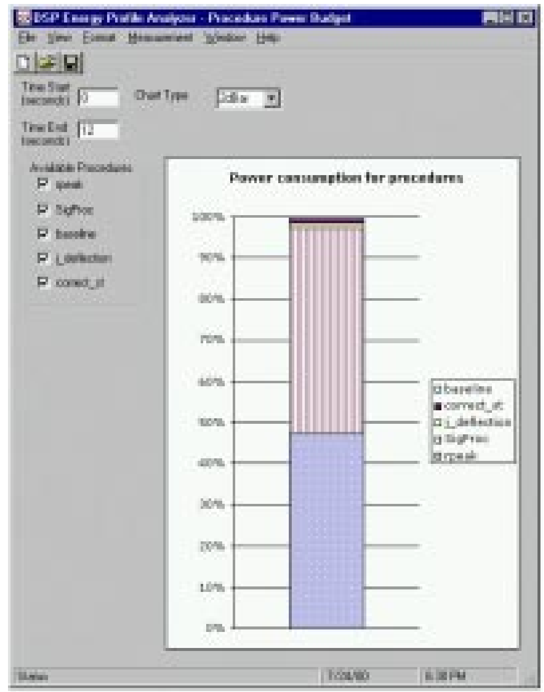

***Figure 3:** Procedure power consumption*

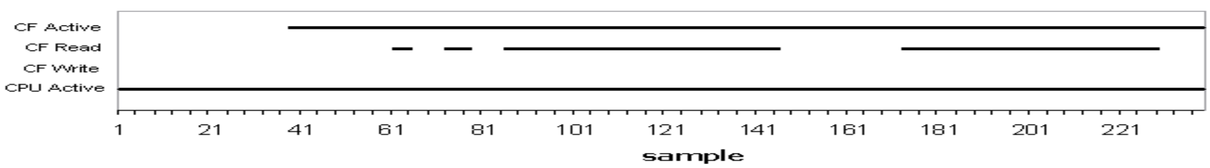

***Figure 2:** Peripheral activity graph for the current profile graph given in Figure 1*

4. DSP Holter Application

The existence of silent myocardial ischemia creates the need for Holter monitoring of the asymptotic patient [2]. Holters are portable, battery operated, devices for monitoring of physiological signals [2][3][4][5]. Lack of processing power has limited the function of Holter devices primarily to data acquisition. Traditionally, Holters have been used to monitor ECG or EEG (brain electrical activity) and record 24-hour activity on cassette tape. Recorded signals are then analyzed off-line using dedicated diagnostic systems. Recently, new generations of Holter devices use solid state memory instead of magnetic tape [5][6].

With the current state of processor technology, standard processing of biomedical signals (such as filtering, spectral and statistical analysis) does not require significant processing power. This is particularly the case for the new generation of DSP processors with processing power in excess of 100 MIPS, even in a portable environment. The main reasons for this are low sampling frequency (typically less than 1KHz) and a relatively small number of channels (usually three for ECG Holters). On the other side, new sophisticated signal processing algorithms require significantly larger processing power (for example, non-linear dynamics [19], wavelets [20], etc.).

In the first period of computerized electrocardiography, real time processing was used to detect and document arrhythmias. Improved performance of monitoring devices allowed morphological analysis and pattern recognition. The most important application of ECG pattern recognition is detection of transient ischemic event [3][4]. Ischemia is considered a primary cause of cardiac infarction and life threatening cardiac arrhythmia.

Real time processing of heart electrical activity is a relatively simple task for the current generation of digital signal processors. With the sampling rates of 250 Hz (long term monitoring) to 1 KHz (heart rate variability analysis), most signal processing procedures could be performed in a fraction of the sampling period. However, morphological analysis of the signal is much more time-consuming.

Table 1: Characteristics of wearable ECG monitor prototype

Processor	TI C5410-100 DSP
Program Memory	128 KB
Secondary Storage	30 MB (compact Flash)
Codec sampling freq.	44.1 KHz
Power consumption	0.5 W (150 mA @3.3V)
Battery weight	100 g
Estimated battery life	10 hours

4.1. Description of wearable ECG monitor prototype

We have implemented a wearable, DSP-based, real-time ECG holter prototype, shown in Figure 3. A real time portable ECG development environment is implemented using a Texas Instruments TMS320C5410-100 processor board [11][24], and a Del Mar PWA ECG Amplifier [5]. The amplifier is a dedicated three-channel ECG Holter amplifier with low power consumption. A set of electrodes and their leads are optimized for Holter applications. We use on board flash memory as program and small-scale data storage.

As Figure 4 shows, the prototype board has several functions that are unnecessary for our ECG application, but are desirable from a prototyping standpoint. For example, the board has LCD output and keyboard input as well as a serial interface. The codec samples at 44.1 KHz, a much higher frequency than required for an ECG. This added functionality burdens the prototype in size, weight, and power consumption, although it considerably eases debugging and allows us to develop a wide variety of applications on a single platform. Ongoing experiments with the prototype platform will allow us to tune both the hardware and the software to the ECG application, especially its power consumption and size. Table 1 shows several features of the prototype, including the measured power consumption and the estimated battery life.

4.2. Improving battery life and system weight by energy profiling of applications

Holter applications typically require battery life of 24-48 hours, so power consumption of the ECG monitor is a critical issue. As Table 1 shows, our expected battery life is much less. Removing some of the unnecessary functionality described in Section 4.1 will decrease the power consumption markedly, but not enough to meet the target battery life. Since batteries are large fraction of the system weight, we cannot simply add more batteries to extend battery life because the system would become too heavy.

One possibility to improve battery life is to find optimum trade-off between energy spent on storing raw data versus energy to compress data. Long-term monitoring applications require significant storage space. As an example 24-hour ECG record with 250 Hz sampling frequency requires approximately 42 MB of memory. This can be easily accommodated using standard compact flash memory card as secondary storage for long term monitoring. This amount of storage could be significantly reduced by data compression. However, data compression will consume considerable CPU energy. This trade-off involves a number of variables including the relative power consumption of the storage subsystem and CPU, level of compression, possibility to use lossy compression, and the quality of compression that can be achieved for the given type of data. In some applications whether or not data would be decompressed by the same battery operated system is also an issue.

The proper balance of power and performance for optimum system organization requires precise profiling of the power consumption of different hardware subsystems as well as software functions.

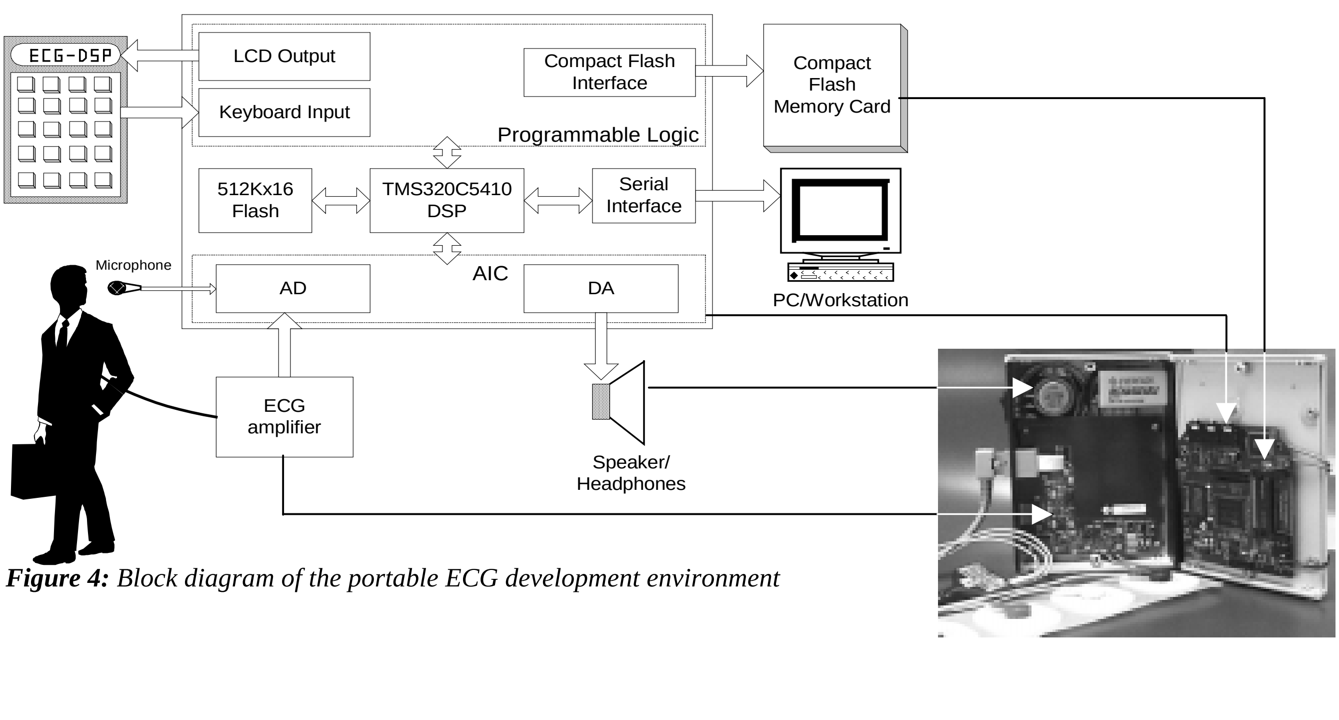

Figure 4: *Block diagram of the portable ECG development environment*

Average power consumption is only a crude estimate of power requirements and battery life; a much better estimate can be made using dynamic power consumption [25]. Dynamic power consumption is a function of the execution profile of the given application running on specific hardware platform.

We are developing a new environment for energy profiling of DSP applications, similar to that described by Flinn for profiling general purpose systems [26]. The environment consists of a JTAG emulator, a high-resolution HP 3583A multimeter and a workstation that controls devices and stores the traces. We use a standard Real Time Data Exchange mechanism (RTDX) [27] to generate an execution profile and custom procedures for energy profile data acquisition using GPIB interface. By correlating power consumption with application execution profile, the environment allows us to improve the system power consumption through changes in software organization and measure real battery life for the given hardware, software and battery configuration.

The proposed environment correlates the application execution profile with power consumption profile. Consequently, we will be able to identify the specific hardware and software components whose power consumption has the most critical impact on battery life.

Once these components are identified, we will be able to optimize for the proper balance between performance and power consumption. As mentioned above, data compression may reduce power consumption due to the secondary storage, although at the cost of increased CPU power consumption. In addition to data compression, we have also identified filtering and encryption of data as primary areas for reducing power consumption of the wearable ECG prototype. Both areas appear to have a large number of trade-offs at the algorithmic level that should be explored to finely tune power consumption and performance.

5. Conclusion

Dynamic power consumption is a critical design issue in estimation of power requirements and battery life. Our environment for energy profiling of DSP systems facilitates exploration of the system design space and finding optimum solutions for the given application. Moreover, we can measure precisely real system operation time for a particular battery family. Therefore, we can more accurately predict battery life in the presence of non-ideal battery behavior.

References

[1] Holzman, T., "Computer-human interface solutions for emergency medical care," Interactions, Vol. 6, no. 3, May 1999, pp. 13-24.

[2] Levin R.I., Cohen D., Frisbie W., Selwyn A.P., Barry J., Deanfield J.E., Keller B., Campbell D.Q., "Potential for Real-Time Processing of the Continuosly Monitored Electrocardiogram in the Detection, Quantitation, and Intervention of Silent Myocardical Ischemia", *Cardiology Clinics*, Vol. 4, No. 4, Nov 1986, pages 735-745.

[3] Deanfield J., "Holter monitoring in assessment of angina pectoris", *The American Journal of Cardiology*, Vol. 59, No. 7, 1987, pages 18C-22C.

[4] Subramanian V., "Clinical and research applications of ambulatory Holter ST-segment and heart rate monitoring", *The American Journal of Cardiology*, Vol. 58, No. 4, 1986, pages 11B-20B.

[5] DEL MAR Medical Systems, http://www.dma.com/Pages/medicsys.htm

[6] Krahn, A., Klein, G., Yee, R., Skanes, C., "Recording that elusive rhythm," CMAJ, November 30, 1999 Vol. 161, pp. 1424-5.

[7] Cole, C., Blackstone, E., Pashkow, F., Snader, C., Lauer, M. "Heart-Rate Recovery Immediately after Exercise as a Predictor of Mortality," The New England Journal of Medicine, October 28, 1999, Vol. 341, No. 18. pp. ??.

[8] Heidenreich, P., Ruggerio, C., Massie, B. "Effect of a Home Monitoring System on Hospitalization and Resource Use for Patients with Heart Failure," American Heart Journal, 1999, Vol. 138, No. 4, pp. 633-640.

[9] Wheelock B., *Autonomous real-time detection of silent ischemia*, M.S. thesis, University of Alabama in Huntsville, Huntsville, 1999.

[10] Maglaveras N., Stamkopoulos T., Pappas C., Strintzis M.G., "An adaptive backpropagation neural network for real-time ischemia episodes detection: development and performance analysis using the European ST-T database", *IEEE Transactions on Biomedical Engineering*, 45:7, pages 805-813, July 1998.

[11] Jovanov E., Gelabert P., Adhami R., Wheelock B., Adams R., "Real Time Holter Monitoring of Biomedical Signals", *DSP Technology and Education Conference DSPS'99*, Aug 1999, Houston, Texas.

[12] Qu, H. and Gotman, J. "A Patient-Specific Algorithm for Detection of Seizures Onset in Long-Term EEG Monitoring: Possible Use as a Warning Device, IEEE Transactions on Biomedical Engineering, 1997, 44, pp. 15-22.

[13] Jung, T., Makeig, S., Stensmo, M., and Sejnowski, T. "Estimating alertness from the EEG power spectrum", IEEE Transactions on Biomedical Engineering, 44, No. 1, 1997.

[14] Streitberg, B., Rohmel, J., Herrmann, W., and Kubicki, S., "COMSTAT rule for vigilance classification based on spontaneous EEG activity," Neuropsychobiology 17, 1987, pp. 105-117.

[15] Taub, E., Pidikiti, R., Chatterjee, A., Uswatte, G., King, D., Bryson, C., Willcutt, C., Jannett, T., Yakely, and S., Spear, M. "CI Therapy extended from upper to lower extremity in stroke patients," Neuroscience Abstracts, 1999, vol. 25, pp. 320.

[16] Jannett, T.C. and DeFalque, R.J. "Integrated instrumentation for closed-loop feedback control of muscle relaxation: initial clinical trials." Proceedings of the 12th Annual IEEE EMBS Conference, Philadelphia PA, November, 1990.

[17] Strubb, H., Johnson, K., Allen, A., Bellotti, V., Starner, T., "Privacy, Wearable Computers, and Recording Technology," Panel discussion, The Second International Symposium on Wearable Computers, October 19-20, 1998, Pittsburgh, PA.

[18] Bell, G. "The Body Electric", Communications of the ACM, February 1997, Vol. 40, no. 2, pp. 31-32.

[19] Hoyer, D., Pompe, B., Herzel, H., Zwiener, U., "Nonlinear Cooridantion of Cardiovascular Autonomic Control", *IEEE Engineering in Medicine and Biology*, Vol 17, No. 6, pages 17-21, 1998.

[20] Akay M., Ed., Time Frequency and Wavelets in Biomedical Signal Processing, IEEE Press, 1999.

[21] Vaage-Nilsen M., Rasmussen V., Sorum C., Jensen G., "ST-Segment Deviation During 24-Hour Ambulatory Electrocardiographic Monitoring and Exercise Stress Test in Healthy Male Subjects 51 to 75 Years of Age: The Copenhagen City Heart Study", *Am Heart J*, 137(6):1070-1074, 1999.

[22] Pan, J., Tompkins, W.J., "A Real Time QRS Detection Algorithm", *IEEE Transactions on Biomedical Engineering*, Vol. BME-32, No. 3, March 1985, pages 230-236.

[23] Van Alste J.A., Schilder T.S., "Removal of Base-Line Wander and Power-Line Inerference from the ECG by an Efficient FIR Filter with a Reduced Number of Taps", *IEEE Transactions on Biomedical Engineering*, Vol. BME-32, No. 12, December 1985, pages 1052-1060.

[24] TMS320C54X DSP: CPU and Peripherals, Reference Set, Volume 1 (SPRU131C), Texas Instruments, 1996, http://www.ti.com

[25] T. Martin and D. Siewiorek, “Non-ideal Battery Properties and Low Power Operation in Wearable Computing," Proceedings of the Third International Symposium on Wearable Computers, San Francisco, CA, October, 1999, pp. 101-106.

[26] Flinn, J. and Satyanarayanan, M., "PowerScope: A Tool for Profiling the Energy Usage of Mobile Applications," Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, Louisiana, February, 1999.

[27] Real-Time Data Exchange (RTDX™), Texas Instruments, http://www.ti.com/sc/docs/tools/dsp/rtdx/index.htm

[28] Jovanov E., Starcevic D., Radivojevic V., Samardzic A., Simeunovic V., "Perceptualization of biomedical data", *IEEE Engineering in Medicine and Biology*, Vol 18, No. 1, pages 50-55, 1999.

[29] Jovanov, E., Starcevic, D., Samardzic, A., Marsh, A., and Obrenovic, Z., "EEG analysis in a telemedical virtual world," Future Generation Computer Systems 15, pp. 255-263, 1999.