*2011 International Conference on Software and Computer Applications*
*IPCSIT vol.9 (2011) © (2011) IACSIT Press, Singapore*

# Testing the Battery Life of Mobile Phones and PDAs

Ondrej Krejcar [1,2]

[1] Department of Measurement and Control, Faculty of Electrical Engineering and Computer Science, VSB Technical University of Ostrava, Ostrava-Poruba, Czech Republic, ondrej.krejcar@remoteworld.net

[2] Department of Information Technologies, Faculty of Informatics and Management, University of Hradec Kralove, Hradec Kralove, Czech Republic, ondrej.krejcar@asjournal.eu

**Abstract.** This paper deals with the battery life of mobile devices and PDAs at various load levels of the battery. For such testing of PDA battery life, there are many programs in ensuring the current battery status. Our developed appplications can determine the battery at defined time intervals and display its status on the device display in numeric format XX%, where the measured value is stored in an external file in XLS format. Addressing problem is designed to store the measured data of the appplication in tabular form, for better evaluation performance and the subsequent effect on the battery. Testing is focused on the impact of various device appplications, that means from low-end appplications to very demanding appplications with Wi-Fi, Bluetooth, GPS, for the battery status.

**Keywords:** PDA, cell phone, battery status, measurement, Endurance

## 1. Introduction

Mobile and PDA devices are for end users, in the most cases, an integral part of everyday life. Such devices are equipped with a wide range of appplications for both work and play, and for communicating with smart networks, which in its use has, unfortunately, negative impact on the battery power of device.

It is known that the battery, during their working life, loses during the time the most valuable, its capacity. Taking into account that each running appplication is for its running more or less energy intensive, the result is revealed by the battery life, especially that of accelerated graphics-intensive appplications, particularly messaging appplications using the module for external communications with smart devices and networks as Wi-Fi, Bluetooth or IrDA, is falling rapidly. An extreme example of an appplication challenging the energy consumption of the battery device is GPS smart phones and PDAs, which is to discharge a fully charged battery to fully discharge, in a relatively short time. In such a situation it leads to rapid loss of electrical power from the battery device, which is needed to supply energy-intensive GPS module.

Battery life has become one of the most frequently occurring issues of mobile systems. Although much effort has been devoted to improving battery life, it did not reach the expected satisfaction. [1].

There is a wide variety of programs which can measure the current battery status and estimate the relative time of usability of equipment, before a battery is discharged. One of them is our program Battery Life Test designed for the measurement of the battery and following display of the current value of the battery on the device display. Application Battery life is created in Visual Studio NET Compact Framework, using C # language. In this environment the application will be further developed into another form. . NET Compact Framework is a subset of the full, desktop .NET framework and therefore the developer working with. NET technology on the desktop does not have to learn a completely new procedures [2].

While the communication device itself consumes a relatively small portion of overall performance, the impact on the overall performance of communication systems is important due to the non-deterministic nature of network traffic, which is busy trying to keep the platform more than is necessary [3].

Measurement of open circuit voltage can be applied to state estimation of rechargeable batteries based on lithium, alkali metal or lead. Unfortunately, this method cannot be used for battery based on the principle of a nickel. On Li-ion cells, 3.8V/cell the charge status of about 50% is indicated. It should be noted that the use of stress as a function of measured variables, is inaccurate, because cells from different manufacturers produce a slightly different voltage profile. This is due to the electrochemistry of electrodes and electrolyte. Temperature also affects the voltage. The higher the temperature is, the lower the voltage is [4].

The remaining time can be detected in several ways. It is possible to observe how quickly percent capacity is diminishing. When the total battery capacity (Wh) and the time period in which changes is known, the remaining time and average consumption can be determined in this timeframe. The second method allows the simple implementation of the meter. When divided by the battery consumption the time remaining will come, if the consumption of the device doesn't change. On this principle the measurement works, which can also be found under the battery icon in the Windows operating systems [5].

Measurement of the actual battery life using the Batery life test is done in real time, the measurement is continuous. Current battery status is indicated by a percentage value displayed on the display of the mobile device. The result table includes the results of individual measurements, showing how the batteries were stressed in these situations and how quickly the battery was discharged at the various levels of testing. Table of results with the measured values is stored in XLS outside the mobile device.

Defining the issues in this chapter can be divided into several points. As a basic principle governing the appplication Battery Life has officially been developed and publicly available appplication, the development is directed to its improvement and tuning, for better cooperation with the end user. Tuning refers to the visual, functional and communication areas. Battery Life program allows you to measure the actual battery life, and then view it on the device display. To measure the actual battery life using software libraries with Visual Studio 2008 is used to access and control the internal periphery of mobile devices and mutual communication between devices and PCs.

## 2. New Solution

To create a new and in many ways better and more sophisticated version of the already mentioned appplications Battery Life Test, it is necessary to use better controls and graphics to the appplication in some way to clarify and add a graphic touch. Another innovation of this version is a combination of appplications running SQL database into which data are entered from the measurement of battery current value.

Battery Life program for an appplication will include several classes, each class will take care of some business. One of these classes will be in charge of the report of the SQL database with measured values from the battery. Another class will be responsible for data transfer via USB from the device to your computer, and it must contain all information necessary for communication.

Video page of this appplication is upgraded, so users get an overview of current battery information. Graphic elements are used as an indicator of battery with a progress bar and text box. The appplication includes a display of current time and date. Measurement of the current values of battery is done automatically spontaneously, without having to run the measurement. Recording of the measured values into the SQL database is run by button. Indication of the registration data is performed by checkbox.

This appplication is used for a public class Form1 system, which runs throughout our program. To perform the measurement of the battery inside class OpenNETCF.Windows.Forms.BatteryLife battery OpenNETCF.Windows.Forms.BatteryLife = new (); is used.

Thanks to that the program obtains information from the system on the current battery and can transmit and re-use them by other components. The value of battery is using the call method battery.UpdateBatteryLife (), updated and then sent to the display layer of numeric values and also to the progress-bar. In order to write the value received as the number you need to convert this value label2.Text battery.BatteryLifePercent.ToString = () + "%" Refresh ();.

The user can control the three elements of this appplication. The first element is running Battery Life to launch the automatic evaluation of the battery and then display the current value of equipment and graphic elements. Part of this step is also evaluating the current system date and time, which is also shown on the

device display. When Battery Life started the button Measurement is displayed on the device display, this button causes that the measured data will write into the SQL database. This is indicated by the checkbox which can also controll the writting. Wherever possible, users should have the ability to export data from SQL database to XLS file.

## 3. Solution Implementation

Real appplication program Battery life is written in object-oriented language C # (C-sharp). To use this language there is for his writing and decoding used a programming environment created by Visual Studio 2008 by Microsoft. For the programming of mobile appplication we need to use .NET Compact Framework (. NETCF) in Visual Studio 2008. This platform is designed for programming with a focus on mobile devices and Pocket PCs exclusively with mobile operating systems: Windows Mobile, which has many limitations.

Programming platform. NETCF, as stated above, is aimed at mobile devices and handhelds. Programmer in this case is given a graphical component representing a virtual mobile devices running Windows Mobile, which is essentially a basic desktop feature (basic class program), like the platform. NETF which is intended only for personal computers with Windows operating systems. The virtual devices are also used for practical use as an emulator of real equipment on which the created program is tested immediately. This emulator has the same features as mobile devices, the same response to user queries and up to some extent it can be worked with as a real device. The main drawback is the impossibility of rendering values of system components (battery, temperature, CPU load, etc..). If available, the mobile device or handheld computer can actually simulate and test a software appplication directly on the device.

To use the Battery Life appplication there are used several graphic elements which function as an informative and switching. Basic class of the program in which all the graphics operations are made, is shown in the program header Battery Life launched in an emulator called Battery Life Test 0.1 (the actual name of the class Form1 is). In the desktop program the layout of individual layers and components, which in this application are used, can be seen.

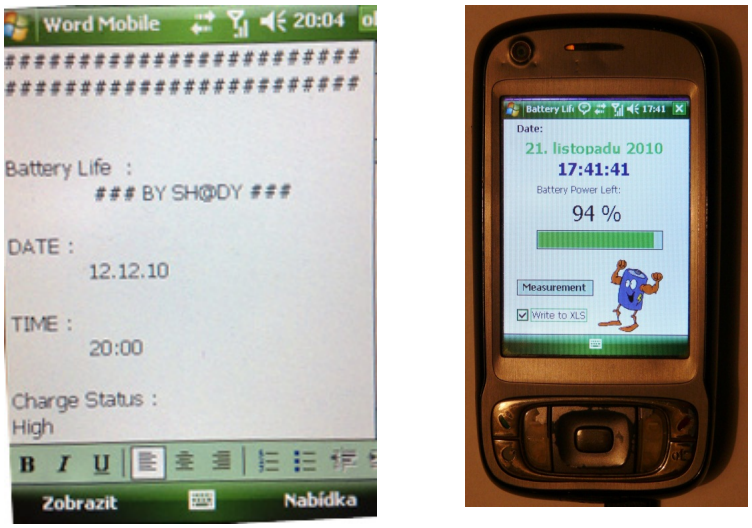


Fig. 1: Sample test programs.

User environment of application Battery Life in real device is shown on figures bellow. User control in this environment is not difficult or complicated. Controls in this appplication are only button to start writing data to the table XLS and checkbox indicating the status of registration and can be used also to change this situation. To disable the appplication the red power button serves.

## 4. Testing of Developed Application

Appplication Batrtery Life is used for measuring the battery status of mobile devices. To assess battery state, reading the system of values is used in the appplication, these are graphically displayed on the display

of the mobile device. When running the application Battery Life, the user has option to decide whether he wants to store the measured data on the condition of batteries in a data file or not. This data file is saved in TXT and XLS also in memory of a mobile device. These files contain data showing the measured parameters of the mobile device. These parameters are expressed successively by a measurement date and time at which the measurements were made, battery charge, battery size on the device display using a progress bar, the battery obtained from system.

As registration and export of data into XLS format, compared to TXT, is technically more complicated, export of measured data to the XLS format is not fully correct, because of that the data in this format are poorly formatted. Testing of program was carried on a mobile PDA device specifically made by HTC. This is a model-TYTNII Kaiser with Windows Mobile 6.1 Professional. The parameters of the device QUALCOM7200 400MHz CPU, 128MB RAM, 256MB Flash, 32-bit data bus. Appplication Battery life is saved in the phone so it can run without being connected to the PC and the programming environment of Visual Studio. The appplication is able to run in the background of the phone with other parallel appplications. On Battery life appplication there has been tested showing of the current battery status reffering to the actual battery status from the readings made by the system. Using real-time display and data display was also tested. These components can be implemented in several ways, which are listed in the source code of this appplication. One way of carrying out the date and time in a real-time,is direct use of system components windows date. Another solution is to use programming communication Windows Mobile APIs (WinAPI hereinafter), when the entire process of registration date and time is created manually and takes only the values of the current time. WinAPI is the possibility for using more precise measurement of time compared with Windows.Forms components. To use the WinAPI it is necessary to use special DLLs that enables access and communication with the processor (Windows systems use the principles of the microcontroller). The disadvantage of using WinAPI's is time display in UTC format, which does not shift in summer and winter time, it's not consistent with the shape of the date and time displayed in the main menu and differs with value of the time -1 hour. Therefore, the source code of the solution for both appplications using the WinAPI and classical use of Windows.Forms (see the source code of appplication - conditional use). For more complicated ways, and very precise use of timers and time measurement it is also possible to use timers using the communication between the fibers and their interruptions to work. For the timer using the fiber components Windows.Threading are used. The use of such timers achieves very high precision, access time of 3-5 ms, which is, compared to using a timer with Windows.Forms with access time around 15ms, almost 3x faster and more accurate. However, a timer using threads is very complicated.

Other tested components the export and registratiton of measured values into the TXT and XLS documents was. Entry is controlled by activating the button Measurement or checkbox. Exported documents are stored in the directory memory cell of mobile device.

The last component tested is the icon of the power key, which consists of icon using PictureBox component, which is set so that when you press shut down the whole appplication is shut down and activities of all running processes in the appplication BatteryLife is stopped.

## 5. Conclusions

In this project, I tried the appplication and use of different types of timers and time measurement using the Windows.Forms API and Windows Mobile. Also, I tried using various system components and obtaining data from the instrument. Interesting but complicated part was the work with export and entry of the measured data into different types of document formats, namely TXT and XLS. Writing to TXT, however, is technically simpler, and when exporting to XLS format there was not achieved a full format correctness, or the correct transfer to XLS. The tested device was able to open XLS file using Notepad, but with the loss of correct data format. I gained some experience in timing components and their precision, but the use of timer working with fibers have not been used because of difficulty of implementation. I've tried working with graphic elements, but I was very limited by the ability to work with .NET Copmact Framework, which does not allows full working with graphic elements. For this reason, the graphic aspects of this appplication are very simple at the expense of performance graphics programming adjustments to individual components of the Framework. Certainly there is a possibility of graphical enhancements of background, using vector

modified graphics components, or programming of graphical and animation capabilities for the buttons and especially progressbar, but the intensity of such changes is on a much higher level of difficulty.

## 6. Acknowledgements

This work was supported in part by (1) „Centre for Applied Cybernetics“, Ministry of Education of the Czech Republic under project 1M0567, (2) „SMEW – Smart Environments at Workplaces“, Grant Agency of the Czech Republic, GACR P403/10/1310, (3) „SCADA system for control and monitoring of processes in Real Time“, Technology Agency of the Czech Republic, TACR, TA01010632 and (4) "User Adaptive Systems", VSB - Technical University of Ostrava under project SP/2011/22. We also acknowledge a support of Jan Smid in developing of test application and Lukas Cernohorsky in consultation support.

## 7. References

[1] Banerjee N (Banerjee, Nilanjan), Rahmati (Rahmati, Ahmad), Corner MD (Corner, Mark D.), Rollins S (Rollins, Sami), L Zhong (Zhong, Lin).: Users and batteries: Interactions and adaptive energy management in mobile systems. Lecture notes in komputer science, Volume: 4717, Pages: 217- 234, 2007

[2] O. Krejcar, D. Janckulik, L. Motalova, Complex Biomedical System with Biotelemetric Monitoring of Life Functions. In Proceedings of the IEEE Eurocon 2009, May 18-23, 2009, St. Petersburg, Russia. pp. 138-141. DOI 10.1109/EURCON.2009.5167618

[3] P. Tucnik, Optimization of Automated Trading System's Interaction with Market Environment, 9th International Conference on Business Informatics Research, Univ. Rostock, Rostock, Germany, Lecture Notes in Business Information Processing, 2010, Vol. 64, pp. 55-61

[4] P. Mikulecky, Remarks on Ubiquitous Intelligent Supportive Spaces, 15th American Conference on Applied Mathematics/International Conference on Computational and Information Science, Univ Houston, Houston, TX, 2009, pp. 523-528, ISBN: 978-960-474-071-0

[5] M. Augustynek, M. Penhaker, D. Korpas, Controlling Peacemakers by Accelerometers. In 2010 The 2nd International Conference on Telecom Technology and Applications, ICTTA 2010. March 19-21, 2010, Bali Island, Indonesia, Volume2, 2010, p. 161–163. ISBN 978-0-7695-3982-9, DOI: 10.1109/ICCEA.2010.288

[6] J. Krawiec, M. Penhaker, O. Krejcar, V. Novak, R. Bridzik, Web System for Electrophysiological Data Management In Proceedings of 2010 Second International Conference on Computer Engineering and Application,s ICCEA 2010, 19. – 21. March 2010, Bali Island, Indonesia, Volume 1, 2010 p. 404-407.

[7] O. Krejcar, Problem Solving of Low Data Throughput on Mobile Devices by Artefacts Prebuffering. *EURASIP Journal on Wireless Communications and Networking*, 2009, Article ID 802523, 8 pages. Hindawi publishing corp., New York, USA, DOI 10.1155/2009/802523

[8] O. Krejcar, R. Frischer, Non Destructive Defects Detection by Performance Spectral Density Analysis, Journal Sensors, MDPI Basel, Vol. 11, No. 3., pp. 2334-2346. (2011)

[9] O. Krejcar, R. Frischer, Detection of Internal Defects of Material on the Basis of Performance Spectral Density Analysis, Journal of Vibroengineering, 2010 – Vol. 12, No. 4 - P. 541-551.

[10] J. Pindor, M. Penhaker, M. Augustynek, D. Korpas, Detection of ECG Significant Waves for Biventricular Pacing Treatment. In 2010 The 2nd International Conference on Telecom Technology and Applications, ICTTA 2010. March 19-21, 2010, Bali Island, Indonesia, Volume 2, 2010, p. 164–167. DOI: 10.1109/ICCEA.2010.186

[11] Z. Labza, M. Penhaker, M. Augustynek, D. Korpas, Verification of Set Up Dual-Chamber Pacemaker Electrical Parameters. In 2010 The 2nd International Conference on Telecom Technology and Applications, ICTTA 2010. March 19-21, 2010, Bali Island, Indonesia, Volume 2, 2010, p. 168–172. DOI: 10.1109/ICCEA.2010.187

[12] P. Brida, J. Machaj, J. Duha, A Novel Optimizing Algorithm for DV based Positioning Methods in ad hoc Networks, Electronics and Electrical Engineering. 2010. – No. 1(97). – P. 33–38.

[13] Kulkarni A (Kulkarni, Ajay), Wang R (Wang, Ren), Maciocco C (Maciocco, Christian), Bakshi S (Bakshi, Sanjay), Tsai J (Tsai, James).: IDC: An Energy Efficient Communication Scheme for Connected Mobile Platforms. IEEE International Conference on Communications, Pages: 5069-5074, 2009