IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 11, Issue 1 (May. - Jun. 2013), PP 87-94
www.iosrjournals.org

Linux-Based Data Acquisition and Processing On Palmtop Computer

[1]Mr. Yogesh P Murumkar, [2]Prof. N. D. Kale,
PVPIT College ,Bavdhan, Pune
Asst. Professor PVPIT, Bavdhan, Pune

***Abstract**: This paper presents an original implementation of a Personal Computer Memory Card International Association data acquisition card on a palmtop computer. The acquisition system is based on the Linux operating system and free drivers for the*
Acquisition board. The system has been used to develop a demo application
Consisting in a phonometer able to sample 1024 samples at 100 ksamples/s and compute the fast Fourier transform of the signal at a maximum rate of 6 frame/s.

I. Introduction:

THE RECENT increase in performance and flexibility of palmtop computers opens new opportunities for application in fields where portability is the main requirement. Many researchers have proposed to use the palmtop computer as the cornerstone for wearable computing [1]. Wearable computers can provide connection to wireless LAN for people that move within a limited space, allowing untethered access to data through the network. An example of such an application is represented by the MANAS project [2], [3], where visitors can access a museum database to obtain multimedia information about the collections. However, in most cases requiring data acquisition
(DAQ) and transmission from the environment around the user to a remote system, palmtop computers offer limited

I/O flexibility and, in particular, only digital interfaces. In many cases, it is desirable to wear a system able to acquire signals from sensors and transducers, process data, and transmit results and/or samples to a remote system [4]. An example is represented by patients that must be constantly monitored during everyday life for which a wearable monitoring system could significantly improve the quality of life. In many cases, the development of a custom acquisition system is necessary; hence,the cost of the system is high, especially if local data processingis required. An example of this type of approach is the AMON project [5], where complete monitoring and complex data processing are performed by a wearable system applied to the patient's arm. The main drawbacks of this approach are cost and limited flexibility.A significant improvement is therefore represented by the possibility to use standard hardware (a palmtop computer) to develop complex wearable acquisition systems. This new Manuscript rece approach dramatically reduces development times and costs (largely due to software) and improves flexibility. Recently,a personal digital assistant (PDA) module for LabVIEW 7.0 has been introduced by National Instruments [6], which allows the development of virtual instruments (VIs) using a subset of LabVIEW libraries. Programs run on Palmtop operating system (OS) [Windows Consumer Electronics (CE)] that is not real time.The aim of this paper is to present a Linux-based implementation of virtual instrumentation concepts [7] on palmtop computers using a commercial Personal Computer Memory Card International Association (PCMCIA) DAQ card. This approach allows developing real-time acquisition systems based on free OS and libraries.In the following sections, a detailed description of the software and hardware implementation will be provided. Furthermore,a demo implementation of the system will be presented,and experimental results on system performance will be discussed.

A. Hardware and OS Selection

PDAs can be grouped in families characterized by the supported OS. In particular, three main families can be found:1) Pocket PC using Windows CE, 2) Palm OS using Palm OS
proprietary OS, and Linux-based PDAs.The selection of the hardware/software platform must satisfy several constraints such as possibility of custom software development,availability of interfaces and drivers for acquisition board hardware, capability of high-speed processing, and realtime operation. None of the products presently on the market fulfills all these requirements. Therefore, in this paper, the selection has been based on the possibility to customize the system to obtain the desired characteristics. In particular, the Pocket PC has been found to offer the possibility to mount the Linux OS and provide PCMCIA interface. Furthermore, a
High-performance processor provides adequate support for data processing algorithms.
B. OS Configuration Pocket PCs normally mount the Windows CE OS, which does not allow custom software development or use of PCMCIA DAQ boards. Furthermore, real-time operation is not allowed; thus, many applications cannot be developed. For these considerations in this paper, we selected a special version

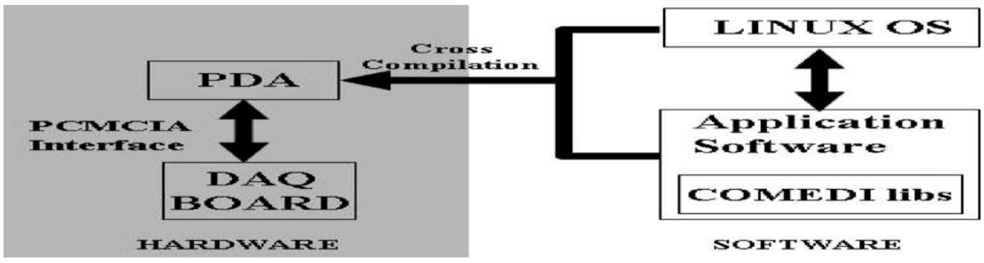

Fig 1. Block diagram of the proposed system. Large part of the development and debugging of the software is performed on a desktop computer.

Final measurement debugging has to be performed on the PDA. of the Linux OS suitable for Pocket PC [8]. Linux is an opensource freeware OS that allows using drivers for PCMCIA DAQ boards provided by the COMEDI project [9]. COMEDI libraries are freeware and provide efficient interfacing with PCMCIA DAQ boards.To obtain the integration of Linux OS, COMEDI libraries,and hardware resources, a custom kernel has to be used. Given the open-source modular nature of Linux, a user can recompile

The kernel to obtain the required functionality set. This approach is mandatory for embedded systems where hardware resources are limited [10]. In this way, only the necessary components are included into the OS, and the user can change the low-level system configuration according to the characteristics of the target application. To implement the custom kernel in the case of the system described in this paper, a special software called cross toolchain is necessary. In our case, the target platform is represented by the Pocket PC (XScale Architecture), and the procedure of cross compilation makes it possible to develop the user–kernel on a desktop PC and later install it on the PDA. The cross-compilation procedure is used also to install COMEDI software and to create the executable files for the DAQ application software.COMEDI libraries contain drivers that allow to manage the PCMCIA slot in the expansion jacket of the Pocket PC. In particular, automatic DAQ card recognition and custom setting is possible by means of suitable configuration files.In our system, an effective integration of OS, libraries, and hardware has been obtained.

C. Measurement Software Development

Fig. 1 shows a schematic representation of the system. As can be shown, part of the development and debugging of the measurement system can be carried out on a desktop computer.In particular, the C-code of the measurement program can be developed using the COMEDI libraries [9], calling library procedures to set up the acquisition board, and acquiring data.Successive data analysis and visualization can be done using standard math and TCL/Tk libraries, respectively.

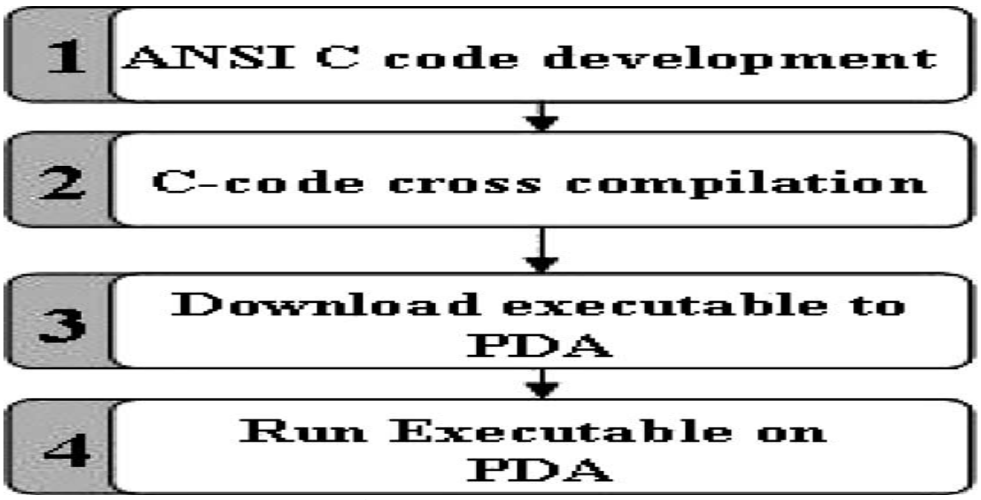
1 ANSI C code development
2 C-code cross compilation
3 Download executable to PDA
4 Run Executable on PDA

Fig. 2. Flowchart of the steps needed to develop the measurement software.

Fig. 2 shows a flow diagram of the steps followed for the development of the instrument presented in this paper.

1) Development of ANSI C-Code for the Application:

In this first step, the developer writes the ANSI C-code program that implements the target application. This phase is carried out on a desktop PC taking advantage of the user-friendly environment.A direct implementation of an application program on the PDA could be difficult and not easily realizable. During this

step, the developer performs a first phase of debugging of the C-code program to eliminate syntactic errors.In the Appendix, a simplified list of the program is reported.In particular, the instructions needed to include COMEDI libraries as well as TCL/Tk libraries have been outlined.As can be shown, the application program makes several call to COMEDI libraries to set the acquisition board channel list,sampling rate, range, etc., and several call to TCL/Tk libraries (not shown) to set up the graphic user interface (GUI). The core of the measurement program is a call to the COMEDI library procedure that starts a buffered acquisition of a user-defined number of samples. After the acquisition, data are processed using
math libraries to evaluate the fast Fourier transform (FFT).Finally, the resulting vector of data (in the frequency domain) is displayed on an amplitude–frequency graph by means of a TCL/Tk library procedure

.2) Cross Compilation of the C-Code:

This step is necessary to create the executable file to be downloaded on the PDA (see next step). In this phase, the developer uses existing tools of cross compilation (cross toolchain) to obtain an executable file for the particular target platform where the application will run. During this procedure, special attention is needed to include all software libraries used by the C-code algorithm. In some cases, static compilation is requested, and the dimension of the executable file significantly increases compared with a dynamically compiled one.

3) Executable Downloading to the PDA and Software Test:

The next of the procedure consists of the download of the executable file obtained from the previous steps to the PDA by a standard communication program utility like Minicom or HyperTerminal.An important role is played by the GUI developed for easy and immediate interaction with the user. To this purpose, there are two possibilities: 1) the implementation of the GUI directly in the executable file by a dedicated C software Library or 2) GUI realization by another development environment. A good choice is to implement the GUI by a TCL/Tk language
[11] since, in this way, a simple and fast development of the GUI can be made directly on the PDA

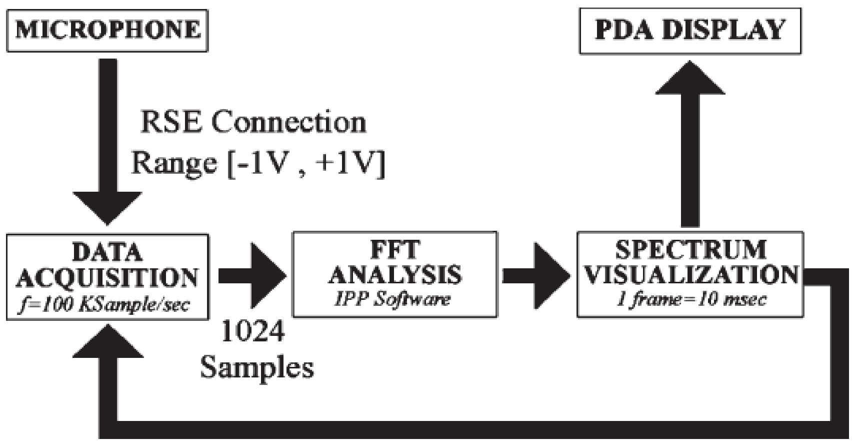

Fig. 1. Block diagram of the proposed system. Large part of the development and debugging of the software is performed on a desktop computer. Final measurement debugging has to be performed on the PDA.of the Linux OS suitable for Pocket PC [8]. Linux is an opensource freeware OS that allows using drivers for PCMCIA DAQ boards provided by the COMEDI project [9]. COMEDI libraries are freeware and provide efficient interfacing with PCMCIA DAQ boards.To obtain the integration of Linux OS, COMEDI libraries,and hardware resources, a custom kernel has to be used. Given the open-source modular nature of Linux, a user can recompile the kernel to obtain the required functionality set. This approach is mandatory for embedded systems where hardware resources are limited [10]. In this way, only the necessary components are included into the OS, and the user can change the low-level system configuration according to the characteristics of the target application. To implement the custom kernel in the case of the system described in this paper, a special software called cross toolchain is necessary. In our case, the target platform is represented by the Pocket PC (XScale Architecture), and the procedure of cross compilation makes it possible to develop the user–kernel on a desktop PC and later install it on the PDA. The cross-compilation procedure is used also to install COMEDI software and to create the executable files for the DAQ application software.COMEDI libraries contain drivers that allow to manage the PCMCIA slot in the expansion jacket of the Pocket PC. In particular, automatic DAQ card recognition and custom setting is possible by means of suitable configuration files.In our system, an effective integration of OS, libraries, and hardware has been obtained.

C. Measurement Software Development

Fig. 1 shows a schematic representation of the system. As can be shown, part of the development and debugging of the measurement system can be carried out on a desktop computer.In particular, the C-code of the measurement program can be developed using the COMEDI libraries [9], calling library procedures to set up the acquisition board, and acquiring data.Successive data analysis and visualization can be done using standard math and TCL/Tk libraries, respectively.Fig. 2 shows a flow diagram of the steps followed for the development of the instrument presented in this paper.*1) Development of ANSI C-Code for the Application:* In this first step, the developer writes the ANSI C-code program that

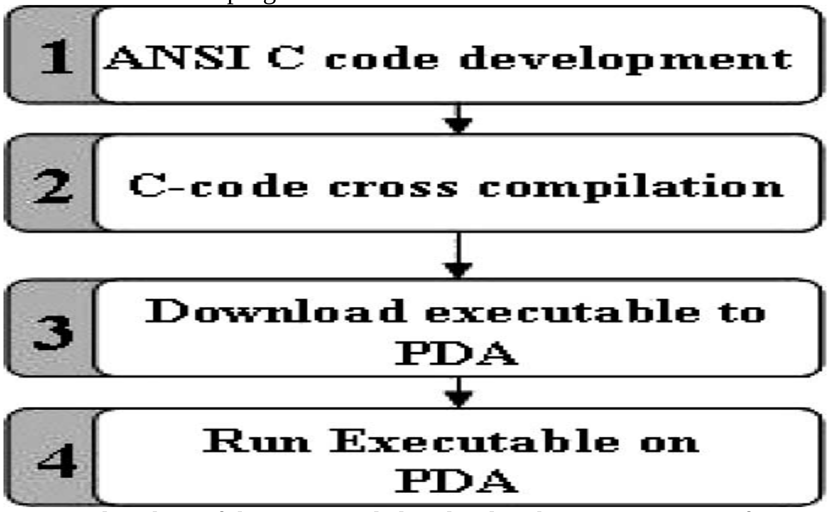

Fig. 2. Flowchart of the steps needed to develop the measurement software.

implements the target application. This phase is carried out on a desktop PC taking advantage of the user-friendly environment.A direct implementation of an application program on the PDA could be difficult and not easily realizable. During this step, the developer performs a first phase of debugging of the C-code program to eliminate syntactic errors.In the Appendix, a simplified list of the program is reported.In particular, the instructions needed to include COMEDI libraries as well as TCL/Tk libraries have been outlined.As can be shown, the application program makes several call to COMEDI libraries to set the acquisition board channel list,sampling rate, range, etc., and several call to TCL/Tk libraries (not shown) to set up the graphic user interface (GUI). The core of the measurement program is a call to the COMEDI library procedure that starts a buffered acquisition of a user-defined number of samples. After the acquisition, data are processed using math libraries to evaluate the fast Fourier transform (FFT).Finally, the resulting vector of data (in the frequency domain) is displayed on an amplitude–frequency graph by means of a TCL/Tk library procedure.

2) Cross Compilation of the C-Code: This step is necessary to create the executable file to be downloaded on the PDA (see next step). In this phase, the developer uses existing tools of cross compilation (cross toolchain) to obtain an executable file for the particular target platform where the application will run. During this procedure, special attention is needed to include all software libraries used by the C-code algorithm. In some cases, static compilation is requested, and the dimension of the executable file significantly increases compared with a dynamically compiled one.

3) Executable Downloading to the PDA and Software Test:
The next of the procedure consists of the download of the executable file obtained from the previous steps to the PDA by a standard communication program utility like Minicom or HyperTerminal.An important role is played by the GUI developed for easy and immediate interaction with the user. To this purpose, there are two possibilities:
1) the implementation of the GUI directly in the executable file by a dedicated C software Library or
2) GUI realization by another development environment. A good choice is to implement the GUI by a TCL/Tk language[11] since, in this way, a simple and fast development of the GUI can be made directly on the PDA.

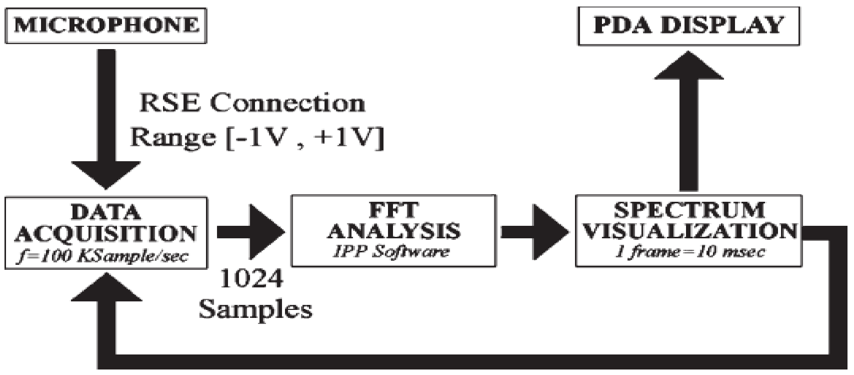

Fig. 3. Structure of the phonometer VI. The output signal of the microphone is connected to the DAQ board using referenced single-ended connection. DAQ is performed using buffered acquisition mode. FFT evaluation exploits a fixedpoint algorithm.

4) Execution of the Measurement Program*:* After executable downloading, the program can be started by means of Linux instructions on the PDA.

D. ANSI C-Code and DAQ Procedures

COMEDI software provides a set of functions implementing various types of DAQ modes so that the developer can choose the best acquisition procedure for the target application.COMEDI procedures allow acquisition of data in single, limited buffered, and continuous buffered modes.Single mode performs an asynchronous sampling of the input signal. In this mode, the sampling procedure is repeated every time a new value of the input signal is needed. In limited buffered DAQ mode, a number of samples are acquired from the target input(s). The developer can choose the sampling frequency and the number of samples to be acquired by setting software parameters in the C-code.The last DAQ mode allowed by COMEDI libraries is the continuous buffered mode. This procedure is similar to the limited buffered mode, but it is for unlimited number of acquired samples. DAQ processes continuously and stores samples in a circular buffer. At the same time, the application algorithm must extract samples from the buffer and process them. The operation stops when a developer-specified event occurs (i.e.,trigger signal).For the implementation of the phonometer application discussed in the next section, we have chosen the limited buffered

mode (Fig. 3) that allows to acquire, process, and display data at a maximum of 6 frames/s rate without circular buffer overflow.In our application, execution has been slowed to 1 frame/s for better readability. Microphone output signal is sampled at a sample rate of 100 ksamples (KS)/s to cover audioband analysis. Antialiasing is performed on the DAQ board; thus, no successive filtering is needed. The acquired samples are then processed using a fixedpoint algorithm for FFT evaluation. Results are then displayed using TCL/Tk procedures.

III. System Experimental Validation

To validate the described measurement platform, we have realized a simple phonometer that analyzes the output signal

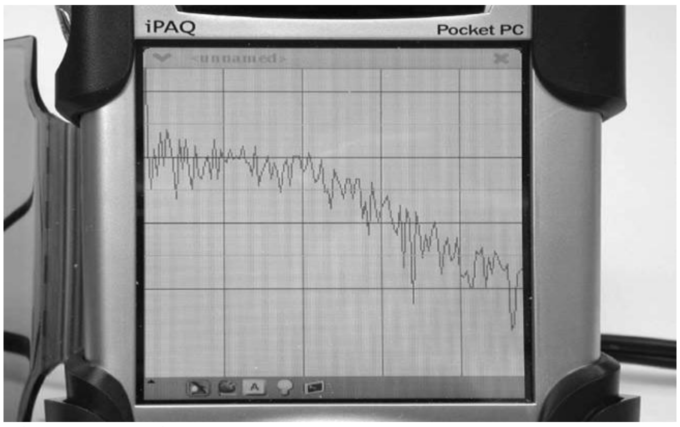

Fig. 4. Photograph of the front panel of the phonometer. Scales have been dropped to better exploit the limited display resolution. Scales range can be manually set by means of menus.of a microphone. This application needs

both signal acquisition and data processing by FFT; thus, it is a good benchmark to evaluate both measurement and processing capabilities of the hardware.Acquisition of the input signal has been performed by setting the acquisition buffer length to 1024 samples and the sample rate to 100 KS/s. The algorithm used to calculate the FFT is optimized to match PDA processor characteristics. In particular,only fixed-point operations are performed [12]. This allows to increase processing speed since the Intel PXA250 processor hasno floating-point unit [13]. The program GUI is represented by a window showing the signal power spectrum.

In the case of our phonometer, visualization of the spectrum on the display of the PDA is realized by means of an X Library Function, which is a set of software functions used by Linux to implement graphic objects [14].

Fig. 4 shows a typical GUI realized for the phonometer developed in TCL/Tk language. With the approach described above, development of the GUI has been simple and fast.

IV. System Performance Analysis

The proposed system has been validated with respect to the following main parameters:
1) execution time;
2) power consumption;
3) accuracy (dc);
4) weight and dimensions.

A. Execution Time

Table I shows the performance of various hardware/OS platforms in terms of execution time of an FFT algorithm working on 1024 samples. In particular, the processor, the software used to implement the FFT algorithm, the OS, and the execution time are indicated. In the case of the Intel PXA250 processor used in our experiments, the library used performs calculations in fixed point, whereas in the other cases, calculations are in floating point.

Measurements have been done also for other sample buffer dimensions (512, 256, and 128 samples), and the dependence of the execution time on the buffer size has been found to be linear.

TABLE I
FFT TIME PERFORMANCE

PROCESSOR	LIBRARY	OS	EXECUTION TIME
INTELPXA250 400 MHZ	IPP	FAMILIAR LINUX	350 usec
Intel PIII 800MHZ	FFTW	Red Hat Linux	90 usec
Intel PIII 800MHZ	Lab VIEW	Red Hat Linux	220 usec
Intel PIII 800MHZ	Lab VIEW	Windows NT	170 usec

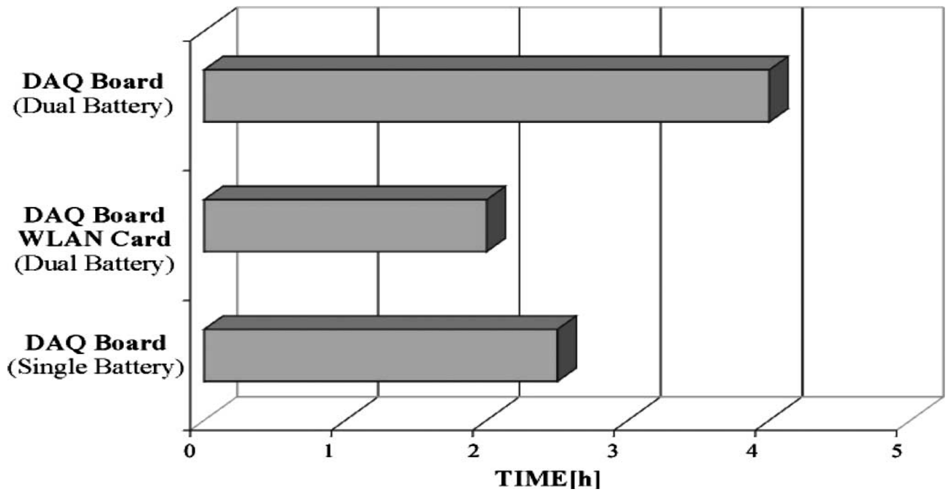

Fig. 5. Battery charge duration as a function of the various operating modes and expansion jackets used. The PDA internal battery duration is of about 14 h of PDA operation without extension boards and WLAN disabled.

B. Power Consumption

An important characteristic of the measurement system is battery duration. A particular problem at this regard is that PCMCIA DAQ boards have a nonnegligible power consumption (for example, the National Instruments DAQCard-6024E requires a peak supply current of 250 mA). In the case of the iPAQ PDA, the extension jacket needed to mount the PCMCIA DAQ board has an internal battery that extends system operation time. As can be shown in Fig. 5, depending on the jacket used, the operation time with the DAQ board mounted and acquiring at full speed is between 2 and 4 h. Additional power is required by the wireless LAN interface, if activated. In the case of dual battery jacket, the lifetime is reduced to less than 2 h.Extending battery lifetime is particularly important in the case of applications where data have to be transmitted to a remote system during acquisition.

C. Accuracy (DC)

The accuracy of the system has been measured, and Fig. 6 shows the error expressed in millivolts occurred, measuring voltages up to 10 V. The signal used for this characterization is a dc level generated by a Ketley 236 SMU. The voltage level obtained by averaging 1000 samples acquired by the DAQ card has been compared with that measured by means of an Agilent 34401A multimeter. The accuracy of this latter instrument is ±190 μV in the worst case (+10 V). The error reported in the graph is the difference between the DAQ card acquisition value

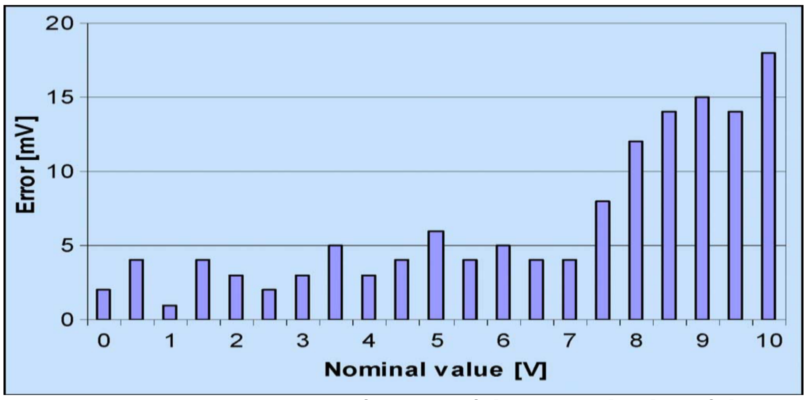

Fig. 6. System voltage measurement accuracy as a function of the nominal value of the test voltage applied to the input of the DAQ board. (Color version available online at http://ieeexplore.ieee.org.) and the multimeter reading. A marked discontinuity has been found, measuring voltages above 7 V. This is due to the change in the acquisition range. National Instruments DAQCard-6024 is a 12 bit/200 KS/s acquisition board, and accuracy specifications show that, in any case, the error is within DAQ board specifications, confirming the correct operation of the system.

D. Weight and Dimensions

In portable systems, weight and dimensions play a key role.In the case of the proposed system, weight is 430 g for the measurement system, including PDA, double slot jacket, DAQ card, and 0.45-m cable. In addition, conditioning circuits and connectors have to be considered. The weight of this latter part of the instrument depends on the particular application.In the case of our demo, 35 g must be added. Dimensions of the system are 140 × 85 × 40 mm (not including external conditioning and connectors).

V. Future Developments: Real-Time Acquisition

An important future development of the system presented above will be the use of a real-time OS, such as RTLinux or RTAI.Some applications, in fact, have strict real-time requirements that nonreal-time versions of the Linux OS cannot satisfy.Linux is designed to be a nonpreemptable kernel, which is not suitable for hard real-time applications. Moreover, the scheduler type (time slicing) may cause unpredictable delays on hardware access and low-resolution timing control.To obtain better real-time performance in Linux, we can take the **following steps:**

1) reduce the complexity and the size of the Linux kernel,removing the critical sections that do not have a deterministic behavior (i.e., uCLinux [15]);
2) modify the Linux kernel standard by specific patches (i.e.,Montavista HardHat Linux);
3) run Linux such as a low priority process in a minimal real-time kernel (i.e., RTLinux [16] and RTAI [17]).

Since many projects are aiming at a Linux real-time OS forhandheld devices (in particular, the FSMLABS and IMEC [18] for ARM processors), it can be envisaged that, in a short time, porting will be possible for the XScale architecture and for other PDA platforms.

VI. Conclusion

This paper has presented a complete implementation of a DAQ board-based acquisition and processing system in a PDA equipped with Linux OS. The system has been validated by means of a VI test vehicle performing FFT analysis of the acquired data. A Linux OS platform for interfacing with a DAQ board has been presented, and further improvements to extend operation to real-time acquisition have been proposed.The measurement system has been characterized. In particular,acceptable battery lifetime has been verified (especially in remote operation with dual battery extension), and overall acquisition performance has been found to respond to DAQ board specifications.

References

[1] Massachussets Institute of Technology. Research Home Page, Mass. Inst. Technol., Cambridge, MA. [Online]. Available: http://web.mit.edu/ research

[2] A. Hanef and A. Ganz, "MANAS—The soul of mobile," in *Proc. World Multi-Conf. Systemics, Cybern. and Informatics*, Jul. 2001.

[3] Sotto Voce, *Electronic Guidebooks*, Palo Alto Research Center, Palo Alto, CA. [Online]. Available: http://www2.parc.com/csl/projects/guidebooks

[4] A. M. Sabatini, V. Genovese, and E. S. Maini, "Portable system for data acquisition and transmission based on handheld PC technology," *Electron. Lett.*, vol. 38, no. 25, pp. 1635–1637, Dec. 2002.

[5] P. Lukowicz, U. Anliker, J. Ward, G. Troster, E. Hirt, and C. Neufelt, "AMON: A wearable medical computer for high risk patients," in *Proc. 6th ISWC*, Oct. 7-10, 2002, pp. 133–134.

[6] National Instruments Measurement and Automation. Research Home Page, Measurement and Automation, National Instruments Corp. Austin, TX. [Online]. Available: http://www.ni.com

[7] H. Goldberg, "What is virtual instrumentation?" *IEEE Instrum. Meas. Mag.*, vol. 3, no. 4, pp. 10–13, Dec. 2000.

[8] Home Page for handhelds.org, *Open Source Operating System for Handheld Devices*. [Online]. Available: http://www.handhelds.org

[9] *COMEDI—The Linux Control and Measurement Device Interface*. [Online]. Available: http://www.comedi.org

[10] A. Lennon, "Embedded systems—Embedding Linux," *IEE, Rev.*, vol. 47,no. 3, pp. 33–37, May 2001.

[11] TCL and Tk Developer site. [Online]. Available: http://www.tcl.tk

[12] *Intel Integrated Performance Primitives for the Intel PXA26x, PXA250, and PXA210 Family of Processor—Reference Manual*, Intel, Inc., Santa Clara, CA, Dec. 2002, Version 3.0. [Online]. Available: http://www. intel.com/software/products/ipp/ipp30

[13] *ARM Application Note 33—Fixed Point Arithmetic on the ARM*, Sep. 1996. Document number: ARM DAI 0033A.

[14] *Xlib—C Language X Interface Reference Manual. MIT X Consortium Standard*, X.Org Foundation, Brookline, MA, X version 11, Release 6. [Online]. Available: http://www.x.org

[15] *CLinux—Embedded Linux Microcontroller Project*. [Online]. Available: http://www.uClinux.org

[16] FSMLabs—The RTLinux Company. FSMLabs, Inc., Socorro, NM. [Online]. Available: http://www.fsmlabs.com

[17] *DIAPM RTAI—Realtime Application Interface for Linux*, Dipartimento di Ingegneria Aerospaziale—Politecnico di Milano, Milano, Italy. [Online]. Available: http://www.rtai.org

[18] *RTLinux for StrongARM*, IMEC, Leuven, Belgium. [Online]. Available: http://www.imec.be/rtlinux