17th Symposium IMEKO TC 4, 3rd Symposium IMEKO TC 19 and 15th IWADC Workshop
Instrumentation for the ICT Era
Sept. 8-10, 2010, Kosice, Slovakia

Matlab Software for High-Resolution Multichannel ECG Measurement

Slavomir Karas, Milan Tysler

Institute of Measurement Science, Slovak Academy of Sciences,
Dúbravská cesta 9, 841 04 Bratislava, Slovakia

Abstract - High resolution multichannel ECG measurement is necessary prerequisite for advanced diagnostics of various cardiovascular diseases. Special multichannel electrocardiograph ProCardio-8 was developed to obtain ECG signals with desired quality. The system is conceived as a virtual instrument with an external intelligent measuring unit connected to a host PC running dedicated application software. In this paper, the data acquisition software developed in MATLAB version 7 and using the Signal Processing Toolbox is introduced. In the first part of the paper, applicability of MATLAB for fast data acquisition was verified and data throughput between the measuring unit and the PC was estimated using a data source with adjustable data rate. Second part of the paper is devoted to development and optimization of a measuring software module that includes control of the measuring unit, testing of electrode contacts and real time data acquisition, visualization and storage of measured data.

I. Introduction

In many cases, especially in so sensitive field as measurement in medicine, the development, debugging and consecutive testing of a real-time measuring software can be a very difficult and time consuming process [1]. The classical approach is to write an application source code in C++ language that certainly takes a large amount of programmer's time and labour. Another possible way that can simplify and accelerate this process is to use MATLAB, its supporting environment and extension functions to compose the code for entire control and communication with the external hardware, for processing of the data stream during the data acquisition and for subsequent off-line processing and analysis of measured signals [2].

The aim of this work was to use Matlab version 7 and its tools to create data acquisition software for a battery-powered ECG mapping system ProCardio-8 [3] developed in our laboratory (Fig.1). The system enables high-resolution multichannel measurement of ECG signals needed for cardiac diagnostics based on analysis of surface potential maps [4]. It consists of a set of active electrodes connected to an external measuring unit and a personal computer that controls the unit and evaluates the measured data. Measuring unit is placed in a patient terminal box and is connected to the USB port of the host PC through an USB FIFO chip (FT245R from FTDI) and an optical USB cable (Opticis M2-110). In full configuration, the system enables recording of ECG signals from up to 128+4 unipolar leads measured relatively to a CMS (common mode sense) electrode. To minimize a common mode disturbance signal, an active neutralization of the patient using a DRL (driven right leg) electrode is employed [5].

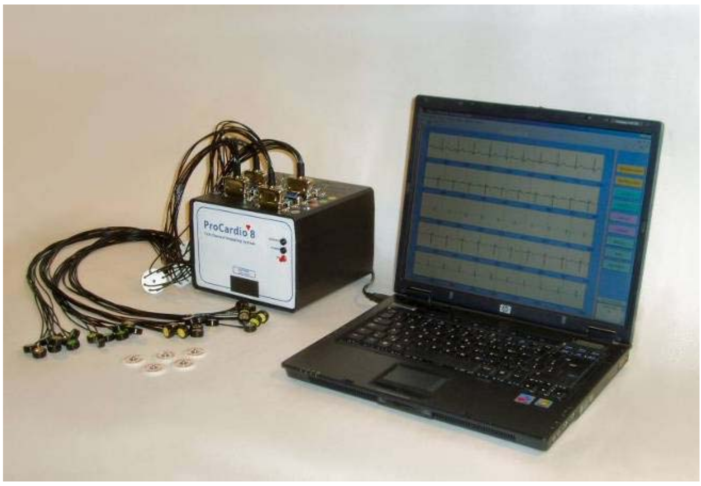


Figure 1. Multichannel high resolution ECG mapping system ProCardio-8.

II. Methods

A. Testing Matlab software for fast data acquisition

The software for communication with the USB FIFO hardware is based on D2XX driver from FTDI and its architecture consists of a Windows WDM driver that communicates with the FIFO device via Windows USB stack and a DLL library of functions that interfaces the application software written in Matlab to the WDM driver. The interface has four layers:

- application interface (written in Maltab),
- FTDI WDM driver interface,
- Windows USB interface,
- USB physical layer.

The ProCardio-8 device is projected for recording of ECG signals with sampling frequencies selectable from 125 Hz to 2 kHz and with resolution of 16 to 22 bits (2 or 3 bytes) per sample. So the required data throughput for simultaneous recording of all 128+4 channels is 0.792 MB/s.

Because the optical USB cable supports transfer rates up to 1.5 MB/s, the limiting factor of the physical layer is the USB FIFO chip with its data transfer rate of 1 MB/s. The question therefore is what transfer rate can be achieved with the Matlab software running under Windows. The main limiting aspects of the software are:

- the ability of Matlab to call a DLL function of the device driver,
- the speed of emptying the Windows USB driver stack.

The aim of our study was to obtain answers to following questions:

- what is the maximum continuous data throughput through the USB channel that can be reached by the Matlab software,
- what is the optimal block size of the data read from the Windows stack into the PC memory to reach maximal data throughput without data overrun in the USB channel.

To obtain these parameters, pulse generator with adjustable frequency was used to generate a write signal for the USB FIFO buffer and simultaneously to drive an 8-bit counter producing 1 byte wide testing data. If the data stream from the USB FIFO buffer to the host PC was not sufficient to transmit all data coming from the counter, a “buffer full” status was reported by the USB FIFO chip and pulses coming from the generator were blocked until free space in the transmit buffer was available again. By gradual increase of the generator frequency until the USB FIFO reported the “buffer full” status we could estimate the maximal uninterrupted data stream through the USB channel under particular conditions in the host PC.

B. ProCardio measuring software in Matlab

Software architecture.

The general scheme of the measuring software for the ProCardio-8 system is shown in Fig. 2. It was designed as Matlab application controlled by a set of parameters stored in global variables. These variables are loaded immediately after the start of the application from system configuration files and can be updated by each user during modification of the system settings. User specific configuration can be saved on disk for future use.

The two main modules of the measuring software are *testing of electrode contacts and signal analysis* and *measuring module*. The first module is executed before each recording of ECG signals from a particular patient. The module measures and displays polarization voltages of electrode-skin interfaces during placing of electrodes on the patient thorax. Simultaneously, DC offsets and amplitudes of ECG signals from all channels are analysed behind the scene and this information is used in subsequent setting of individual measuring channels. After sufficiently long and successful electrode and signal testing the module can be terminated and measuring module can be started.

The measuring module uses parameters obtained from the system configuration files and from information acquired by the testing module. The module includes functions for communication over the USB port and transferring data between the measuring unit and host PC, functions for low-level control of the microprocessor in the measuring unit, functions for reading of the data stream from the USB port buffer and their conversion to ECG samples in proper format, functions that dynamically modify the graphical user interface and visualize the measured signals in desired format on the PC screen and functions for storing of the recorded data on the hard disk.

Electrode testing and signal analysis

The ECG signals measured from human thorax have amplitude approximately from 0.5 to 5 mV [8]. If considering typical signal amplitude of 1 mV, fixed gain of the amplifiers set to 40 and the input range of measuring channels -2.5 to 2.5 V, the amplitude of the useful ECG signal is 0.8 % of the full input range of the measuring channel. As the measuring system is equipped with DC-coupled ECG amplifiers the software must take into account that also all low frequency artefacts from the patient (movement artefacts, breathing activity in

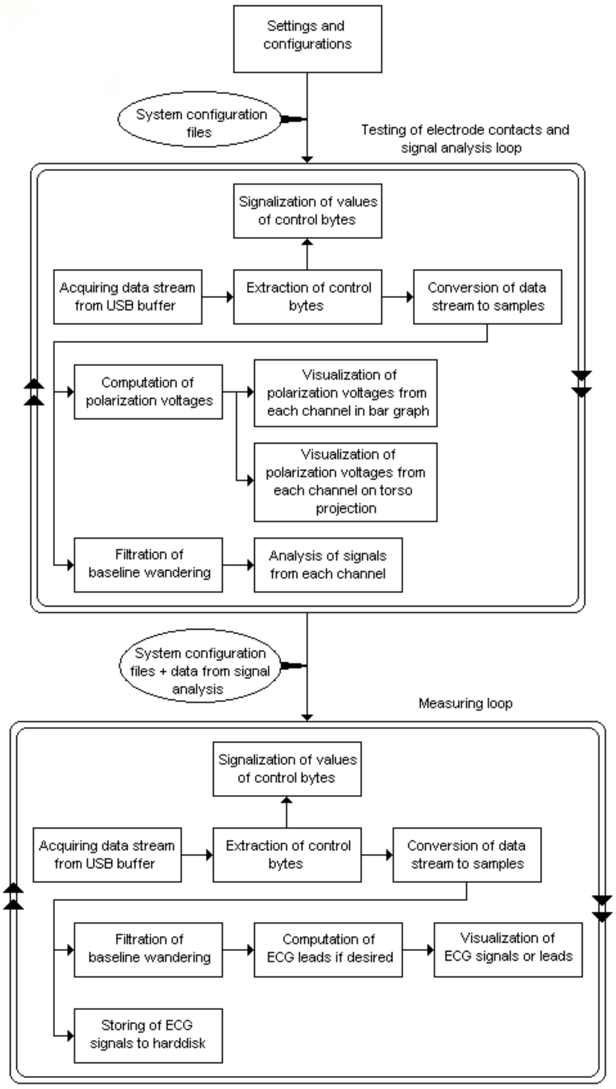

Settings and configurations
System configuration files
Testing of electrode contacts and signal analysis loop
Signalization of values of control bytes
Acquiring data stream from USB buffer
Extraction of control bytes
Conversion of data stream to samples
Computation of polarization voltages
Visualization of polarization voltages from each channel in bar graph
Visualization of polarization voltages from each channel on torso projection
Filtration of baseline wandering
Analysis of signals from each channel
System configuration files + data from signal analysis
Measuring loop
Signalization of values of control bytes
Acquiring data stream from USB buffer
Extraction of control bytes
Conversion of data stream to samples
Filtration of baseline wandering
Computation of ECG leads if desired
Visualization of ECG signals or leads
Storing of ECG signals to harddisk

Figure 2. Block diagram of the measuring software.

relaxed state with basic frequency of 0.17 – 0.3 Hz [9]) and skin-electrode interface (differences in polarization voltages between electrodes with normal values up to several mV) are superposed on the useful ECG signals. Moreover, the software has to compensate or report to the user DC offsets of measuring channels due to non-ideal electronic parts (mainly the amplifiers) or missing or bad electrode contacts. Before the signal measurement and visualization the software has to "find" the ECG signals within the input range and dynamically adjust the signal offset and amplitude. From this it is clear that to ensure optimal signal quality and visibility of all channels during the measurement with sufficient amplitude resolution is not a trivial task.

Real-time visualization of ECG signals

The used sampling frequencies yielding high number of signal samples together with resolution of real PC monitors result in more acquired points from sampled signals than it is possible to display on the screen. To ensure proper reconstruction of the signal shape in real-time without repeated plotting of redundant points or discontinuities in signal tracings an appropriate method has to be used. The best results could be obtained by using the sinx/x interpolation [6], however this method is applicable only in situations where the time is not critical (mainly off-line processing or real-time applications with a signal processor).

Solution proposed for our Matlab software is shown in Figure 3 and is similar to the method for peak detection used in some digital oscilloscopes (LeCroy or Tektronics, [7]). The method displays the signal amplitude (from minimal to maximal sample value) in predefined time interval Δt to one vertical line that has the width of one pixel on the display unit. In this way compression of the time axis is achieved without missing the signal peaks. The value of the time interval Δt has to be selected with respect to the character of the displayed signal. For ECG signals, Δt should be smaller than 30 ms, (i.e. the minimal ECG wave duration shown in the Table 1 [8]) to be sure that some ECG waves will not be merged. For our software we have selected Δt = 16 ms for all possible sampling frequencies. Value of Δt thus defines the time base for visualization of ECG signals during the measurement.

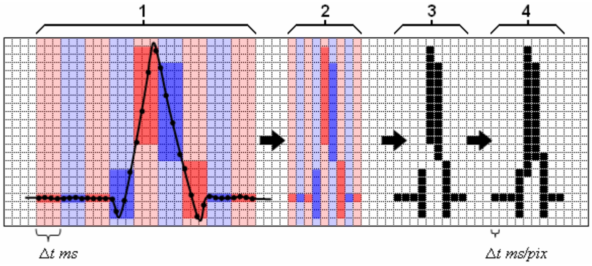
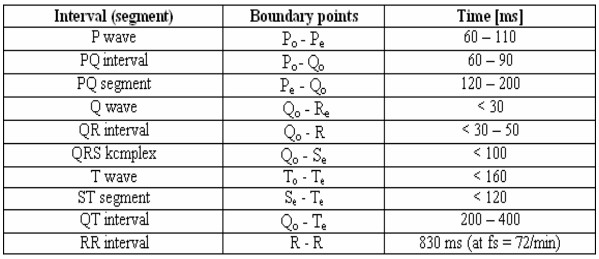


Figure 3. Fast detection of signal envelope.



Interval (segment)	Boundary points	Time [ms]
P wave	P_o - P_e	60 – 110
PQ interval	P_o- Q_o	60 – 90
PQ segment	P_e - Q_o	120 – 200
Q wave	Q_o - R_e	< 30
QR interval	Q_o - R	< 30 – 50
QRS kcmplex	Q_o - S_e	< 100
T wave	T_o - T_e	< 160
ST segment	S_e - T_e	< 120
QT interval	Q_o - T_e	200 – 400
RR interval	R - R	830 ms (at fs = 72/min)

Table 1: Durations of ECG waves.

From Figure 3 another problem of the visualization is apparent: if the amplitude change of acquired ECG signal between two samples is bigger than corresponding screen resolution, the signal trace may be broken. This occurs usually on the onset and offset of the R wave, especially with low sampling frequencies as can be seen in Figure

3, segment 3. To solve this problem, artificial connections between vertical lines of previous and following processed intervals of the visualized signal need to be added as shown in the 4th segment in the same figure.

III. Results

It is obvious that the data throughput of the data acquisition software may depend on the power of the host computer. During our testing we used a notebook with a single core 32-bit 1.8 GHz Intel processor and 1 GB of RAM. We could confirm that maximum data throughput, determined by the condition that the USB FIFO transmit buffer in the measuring unit must not overflow, was increasing with increasing size of the read buffer defined in the Matlab software. Maximal data flow was reached with read buffer sizes above 50 bytes (Figure 4 left) and remained constant even if the read buffer was further increased. This result suggests that the data flow of about 0.85 MB/s is the highest data throughput that can be reached with the used computer configuration to guarantee reliable and continuous transfer of the data stream. This value was obtained only with compiled Matlab code, while without compilation lower data flow of about 0.81 MB/s was achieved. In Figure 4 right, corresponding times for emptying the whole Windows input buffer can be seen.

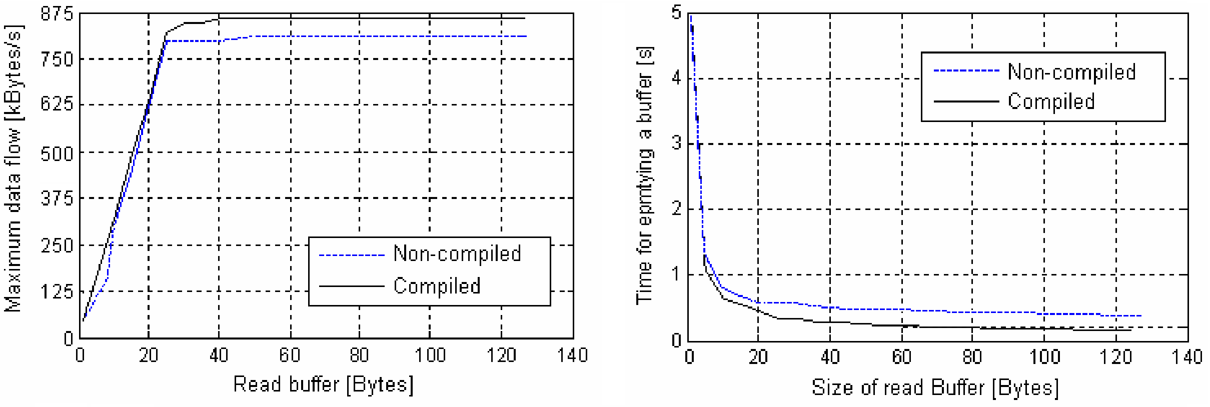


Figure 4. Left: Dependence of the maximum data flow through the USB channel on the size of the read buffer. Right: Dependence of the emptying time of the Windows USB buffer (~64 kB) on the size of the read buffer.

From the results achieved in this part of the study it could be concluded that the fast multichannel data acquisition necessary in the ProCardio-8 system can be accomplished even with a measuring program fully written in Matlab.

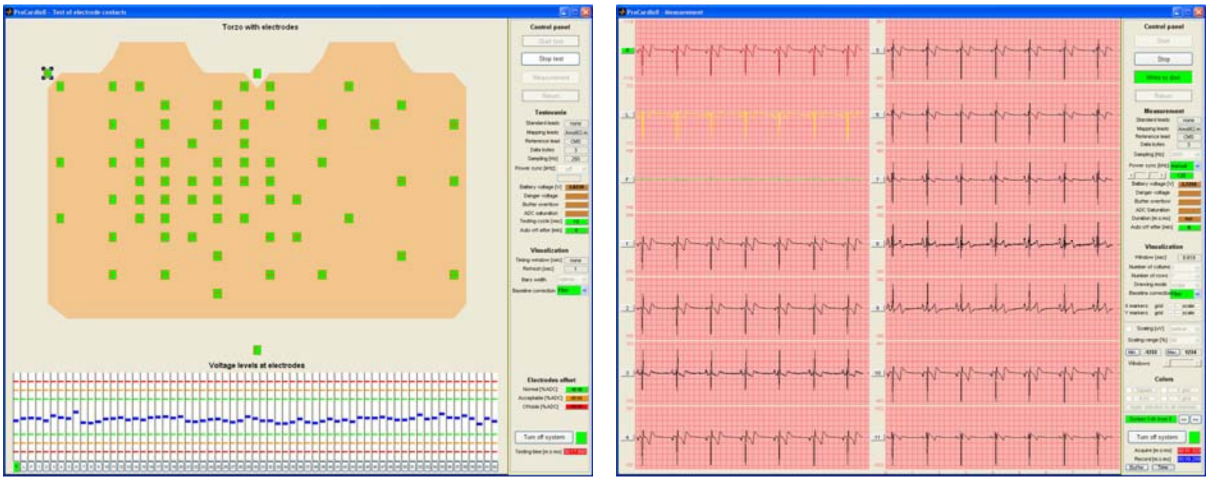


Figure 5. Left: User interface during the testing of electrode contacts and signals analysis.
Right: Application screen of the measuring module during recording of 62 signals of the Amsterdam ECG lead set s of the Amsterdam lead system.

Typical screens of the developed ProCardio-8 measuring program described above and written in Matlab are shown in Figure 5.

During the testing of electrode contacts and signal analysis (Figure 5 left), signals from all used channels are filtered by a high-pass filter. This filter eliminates the artefacts caused by patient breathing activity and noise in the ECG. Resulting signals allow to judge about the quality and stability of electrode contacts. With increasing DC offsets and its variability (representing mainly differences of polarization potentials between the measuring and reference electrode) quality of electrode contact is decreasing. DC offsets of individual electrodes and acceptable offset limits are shown in the lower part of the screen. On a torso scheme, positions of all electrodes are shown and optimal, suboptimal or bad electrode contacts are marked by green, yellow or red colour. In this program module also amplitudes of all measured channels are analysed and this information is used in the measuring module to ensure optimal visualization of ECG signals in individual windows during the measurement.

Typical screen of the measuring module during the signal recording is shown in Figure 5 right. In this module, low frequency artefacts are suppressed and amplitude of the signals is optimized for signal visualization. On-screen controls allow to start and stop data acquisition and recording of raw signals and to configure the screen according to user needs.

In both screens, most controls for setting of the measurement parameters and adjusting screen layout and appearance are in the right part of the display.

IV. Discussion and conclusions

In this paper prototype of a measuring software for the ProCardio-8 multichannel high-resolution electrocardiograph fully written in Matlab was introduced. It was shown that even with high resolution 22 bit A/D conversion, sampling rates up to 2 kHz and 132 used measuring channels the solution is feasible. However, it may be limited by the power of the used computer and the mode of signal visualization during the measurement (number of visualized leads and difficulty of their real-time processing).

The developed software enables automated checking of the measuring conditions before and during the ECG measurement which is especially important if many signals are measured simultaneously.

The software also solves problems related to the visualization of measured ECG signals with sufficient amplitude resolution when the low-frequency artefacts are present. The solution is based on testing of the signal parameters before (and also during) the measurement. We can state that optimal processing and visualization of ECG signals during the measurement is guaranteed if following assumptions are valid:

- conditions during testing of electrode contacts and during the measurement are the same (including the patient and environment conditions),
- rate of change of the low-frequency artifacts is small enough (frequency by which the baseline of measured ECG signals is modulated is lower than 0.68Hz which is the cutoff frequency of the high-pass filter,
- measured signals are quasi-stationary (ECG signals are cyclo-stationary signals),
- testing of electrode contacts runs sufficiently long time (so the transients decayed).

Signal processing during real-time visualization does not affect the stored data as raw data are recorded on disk.

References

[1] BRONZINO, D. JOSEPH. Biomedical Engineering Handbook: Section VI Biomedical Signal Analysis. CRC Press, 1995

[2] HANSELMAN D., LITLLEFIELD B. Mastering Matlab 7. Prentice Hall, 2005

[3] TYSLER M., SVEHLIKOVA J., ROSIK V., KARAS S., HEBLAKOVA E., KNEPPO P., MUZIK J., KANIA M., ZACZEK R., KOBYLECKA M.: Model-based Method and Instrumentation for Noninvasive Identification of Local Ischemic Lesions in the Heart. *IFMBE Proceedings*, Vol.25/IV, pp.1403–1406, 2009

[4] ROSÍK V., KARAS S., HEBLÁKOVÁ E., TYŠLER M., FILIPOVÁ S.: Portable Device for High Resolution ECG Mapping. *Measurement Science Review*, Vol.7, Sec. 2, 4, pp. 57-61, 2007

[5] WINTER, B. B., WEBSTER, G. J.: Driven Right Leg Circuit Design. *IEEE Transaction On Biomedical Engineering*. Vol. BME-30, No. 1, 1983

[6] Tektronix Technical Brief: Real-Time Versus Equivalent-Time Sampling, 1992

[7] HAVLIK L.: Osciloscope and its using. Sdelovaci technika Praha, 2002

[8] ROZMAN J. at all.: Electronic devices in medicine. Academia Praha, 2006

[9] JAVORKA K. at all.: Medical Physiology. Osveta Martin. 2006.