ELEKTROTEHNIŠKI VESTNIK 78(5): 281-286, 2011
ENGLISH EDITION

Design and Quality Assessment of the Graphical User Interface Software of a High-voltage Signal Generator

Denis Pavliha, Matej Reberšek, and Damijan Miklavčič

University of Ljubljana, Faculty of Electrical Engineering, Tržaška 25, 1000 Ljubljana, Slovenia
E-mail: denis.pavliha@fe.uni-lj.si

Abstract. In this study we present design, construction and evaluation of an electroporator to be used in biological cell electroporation for generating nanosecond (i.e. of duration between 40 and 200 ns) high-voltage (up to 1 kV) electric pulses. The device is controlled by a reliable, fast-responsive, and user-friendly Graphical User Interface based on the Personal Computer architecture. The Graphical User Interface enables touch-screen control by using a touch-screen display for human-machine interaction. Since safety is an important issue in biomedical engineering, a comprehensive quality assessment was made involving thoroughly testing all the input parameters (i.e. software handlers for on-screen controls) of the Graphical User Interface software with black-box methods, using the test-case design approach, and with white-box methods, using the code-inspection approach. Several minor and major errors were found and eliminated in the development phase, thus resulting in a safe operation of the developed electroporator.

Keywords: electroporation, pulse power electronics, graphical user interface, embedded software testing, code inspection.

1 INTRODUCTION

Electroporation is a phenomenon that occurs when a biological cell is exposed to an electric field caused by high-voltage electric pulses with durations in the range from nano- to milliseconds [1]. In the presence of an external electric field, biological cells get porated: their membrane becomes permeabilized, which allows molecules from outside to enter the cell [2]. Lately, however, special attention has been given to nanosecond pulses (i.e. nanopulses), since their effect on biological cells causes not only external, but also internal cell membrane poration [3]. In order to generate nanopulses, new devices (i.e. electroporators) are needed because output power stages of regular electroporators cannot generate pulses with durations shorter than 1 µs [4].

Since commercial microelectrode nanopulse electroporators are not available yet, we designed and constructed an electroporator that can generate electric pulses with parameters (amplitude between 250 and 1000 V, duration between 40 and 200 ns, number of pulses between 1 and 100, and repetition rate between 1 and 100.000 Hz) for experiments using microelectrodes under a microscope. Although such experiments are conducted within a laboratory environment, we opted towards designing a robust and safe device that would be at the same time easy-to-use. When designing biomedical devices, their critical interfaces are: hardware/software, software/user and hardware/tissue.

The hardware/software interface is handled by the operating system (OS) which hosts the graphical user interface (GUI) software. GUI actually introduces to the process of device design the so-called hardware abstraction layer (HAL). HAL allows the designer to separate the software platform from the device hardware structure. Such separation allows the user to use a complex device like an electroporator more effectively and without advanced knowledge of the device properties or inner structure since with HAL, a fully functional GUI can be designed without presenting the end-user any hardware parameters at all. To achieve hardware abstraction, we built a GUI controller [5] based on the Personal Computer (PC) architecture. In the past, a possible way of designing a GUI would have simply been by writing a computer program that runs on a desktop operating system [6]; however, we find transforming the whole PC into an embedded GUI controller a better approach. In this way, all the software that runs on the PC including OS can be dedicated to provide human-machine interaction only and no external PC is needed to operate the device, which adds to the device compactness.

Our user-friendly GUI implemented in the nanopulse electroporator [7] enables the user to fully control the device (i.e. the software/user interface). In designing our GUI controller which performs reliably and allows for an optimally fast user response, we followed the guidelines of Garmer stating that by improving the user

Received November 10, 2011
Accepted November 28, 2011

interface usability, incidents and accidents would be reduced and so would be the time needed to learn how to use new equipment [8]. Namely, functioning of a system is defined by the operating system and software applications used to provide human-machine interaction. Therefore, choosing an appropriate software platform and then developing a software application is in many cases the most challenging task in developing a device such as an electroporator. While well-designed software can, indeed, improve the user experience when interacting with a device, poor software design can only lead to a degraded user experience (i.e. the user does not feel comfortable using the device) and, in the worst case, can give rise to accidents. It is therefore most important that designing the software platform is followed by the software quality assessment to determine if there are any errors that may lead to erroneous situations and consequently degrade the user experience.

Besides the GUI hardware part, it is also important to design a robust and safe device output power stage for being a component the user is directly exposed to (i.e. the hardware/tissue interface). In order to safely use them in a medical environment, medical devices must meet specific safety specifications laid down in technical standards (e.g. International Electrotechnical Commission IEC-60601, Medical Electrical Equipment) mostly imposed on the leakage currents, galvanic insulation, normal and single-fault states, electromagnetic interference (EMI), and electromagnetic compatibility (EMC). Even though the designed prototype was not meant to be used in medicine, safety was our primary objective throughout its development.

2 MATERIALS AND METHODS

2.1 Hardware Platform

The hardware platform of our custom-built nanopulse electroporator [5] consists of two parts: the GUI controller and the output power stage.

The GUI controller is based on the Personal Computer (PC) architecture. A mini-ITX mainboard VIA Epia EN12000EG (VIA Technologies, Taiwan, 2001) was used as the main hardware platform for the GUI controller. Communication between the GUI controller and the output stage is done via the Universal Serial Bus 2.0 (USB 2.0) connection. Data resides on a CompactFlash (CF) card connected to the mainboard Integrated Drive Electronics (IDE) controller. The touch-screen Liquid Crystal Display (LCD) is connected to the mainboard Video Graphics Array (VGA) controller (for display) and also via a USB 2.0 connection (for touch-screen controls). The schematics of the hardware platform are presented in Fig. 1.

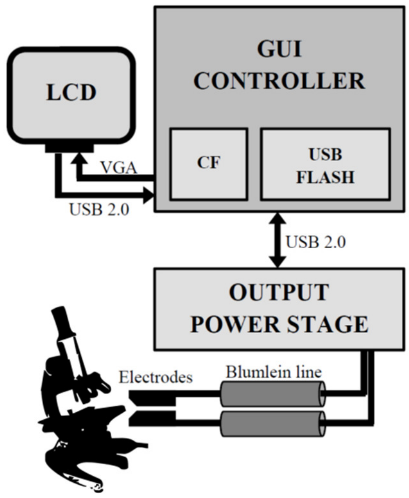

Figure 1: Hardware platform of our custom-built nanopulse electroporator. (LCD - Liquid Crystal Display, VGA - Video Graphics Array, USB - Universal Serial Bus, GUI - Graphical User Interface, CF - CompactFlash)

The device is enclosed in a compact electro-magnetic compatible case with dimensions of 54 x 34 x 54 cm (width *x* height *x* depth) and it weighs approx. 30 kilograms. The used display is a 15" LCD touch screen (1537L, EloTouch, USA, 2005). Besides embedding the LCD, the front panel also includes a pair of SubMiniature version A (SMA) output connectors for connecting the Blumlein line with electrodes to the device, and a power-on hardware button. Namely, a Blumlein generator represents a pulse-forming network where the output amplitude is the same as the supplied voltage and the impedance of the load is twice the impedance of the line. The Blumlein generator comprises a high-voltage switch, two coaxial lines, and a load between these two lines. Our device is designed as a modified Blumlein generator that generates variable-duration and high repetition rate high-voltage nanosecond pulses by four synchronized switches.

2.2 Software Platform

2.2.1 Operating system

All the software of our custom-built nanopulse electroporator is based on the Microsoft Windows CE 5.0 .NET operating system (Microsoft, USA, 2004). First, a bootloader is used to load the operating system. When the operating system is built on the development workstation, the Windows CE development environment (named Microsoft Platform Studio) generates a binary image that can serve as a payload for the native Windows CE bootloader; the latter is designed for the MS-DOS operating system (Microsoft, USA, 1981). Nevertheless, since MS-DOS needs to be loaded before the Windows CE bootloader, such a procedure imposes an additional delay when booting the nanopulse electroporator. To minimize the delay at the boot time, an alternative bootloader is used: the

FastBoot bootloader (VIA Technologies, Taiwan, 2004) allows loading a binary operating system image without MS-DOS interaction (i.e. from the Master Boot Record of the CF card). In that way, the boot procedure is faster compared to the native Windows CE bootloader.

The image of the Microsoft Windows CE 5.0 .NET operating system was custom-built to include only components required for the functioning of the nanopulse electroporator. Our custom build of the operating system includes basic components for networking (i.e. Network Driver Interface Specification - NDIS Support, TCP/IP, and Network Utilities), USB communication (i.e. Enhanced Host Controller Interface - EHCI Support, USB Host Support), data storage (i.e. Storage Manager, FAT File System, USB Storage Driver) and .NET programmability (i.e. .NET Compact Framework 2.0). The operating system supports the hardware platform via a Board Support Package - BSP (VIA Technologies, Taiwan, 2008) provided by the hardware board manufacturer.

Besides the custom configuration of the operating system components, some other modifications were also made to the operating system. Namely, in the operating system configured there is no default shell meaning that no default user interface is available (e.g. the Windows desktop). Instead, a custom program application which serves as a .NET-invoking application is executed after the Graphic Windowing and Events Subsystem (GWES) is loaded. The invoking application is written in the C programming language and built within the Windows CE Platform Builder which is the development environment of the Microsoft Windows CE .NET 5.0. The invoking application is required because a direct invoke of the managed (i.e. written in the .NET Framework) applications from the Microsoft Windows CE .NET 5.0 operating system is not possible; therefore, such an invoking application serves as a bridge for indirect application calls.

2.2.2 *Graphical User Interface software*

The Graphical User Interface (GUI) is the main interface between the end-user and the nanopulse electroporator. The GUI is namely a managed application written in the Visual Basic .NET programming language and built in Visual Studio 2005 Professional (Microsoft, USA, 2005). The GUI is designed for touch-screen control and allows the user to set pulse parameters (amplitude, duration, number and repetition rate) with on-screen sliders or an on-screen numerical keypad. Two different input methods were implemented as the user may need a fast input mode (i.e. the on-screen sliders) or an accurate input mode (i.e. the on-screen numerical keypad). Actions (i.e. arming the device, triggering pulses and discharging the output) can be executed by pressing appropriate on-screen buttons. Moreover, all the device parameters can be both saved or loaded as presets residing on an externally attached USB flash disk, so a later modification of presets (i.e. easily examinable text files) is possible on a workstation.

The managed application that represents the GUI consists of three major software sections: the user interface, the core of the software, and the communication interface. The user interface software section includes handlers for visible on-screen components (e.g. numerical keypad, action buttons, parameter sliders, etc.) and only registers the user actions. The latter are examined by the core of the software which is designed as a polling system: the user input namely triggers output actions, therefore all the actions are polled by the user asynchronously. Finally, the communication interface handles the communication between the GUI and the output stage of the nanopulse electroporator using a custom Universal Serial Bus 2.0 (USB 2.0) driver for the output power stage USB client integrated circuit. Nevertheless, the hardware commands (e.g. for pulse generation) are generated in the output stage of the nanopulse electroporator; the GUI only sends the desired pulse parameters (i.e. amplitude, duration, number and repetition rate) together with the actual command, while all further hardware calculations are done in the output stage itself. In that way, hardware abstraction is achieved.

However, calibration of the output stage is necessary to match the GUI with the hardware (i.e. the output stage of the nanopulse electroporator) and is possible through a password-protected setup screen within the GUI. The setup screen protection is implemented because calibration is a one-time step preferably done by the developers, and is not intended to be additionally executed by the user.

2.2.3 *Driver for the output stage USB connection*

To achieve support for devices that are not natively compatible with the Microsoft Windows CE 5.0 .NET, software drivers are needed. The output stage of the electroporator is connected to the GUI controller via the Universal Serial Bus 2.0 (USB 2.0) connection using the integrated circuit Cypress EZ-USB FX2LP (CY7C68013A, Cypress, USA, 2002) which lacks driver support for the Microsoft Windows CE .NET 5.0 operating system since these drivers are not provided by the manufacturer. As driver writing is a complex procedure, we developed a custom driver using the Windriver development software (Jungo Ltd, USA, 2008). The Windriver namely allows development of the device drivers for the Microsoft Windows CE .NET 5.0 operating system based on the device properties that are evaluated upon connecting the target device to the development workstation. Then, a wide-applicable driver is generated in several programming languages.

The Windriver-based Cypress EZ-USB FX2LP device driver for the Microsoft Windows CE .NET 5.0 operating system was developed in the C programming language and built using the Microsoft Embedded Visual C++ 4.0 (Microsoft, USA, 2004). Further customizations were made to the Windriver-generated driver in order to allow external function calls that are necessary for the functioning of the nanopulse electroporator (e.g. Vendor Requests, GPIF Single Read, etc.). The driver is a Dynamically-Linked Library (DLL) and acts as a wrapper for the Windriver core components. The DLL driver (which is unmanaged program code) exports custom-written USB-related functions and is invoked from the GUI application (which is managed program code) by performing data marshaling [9]. The latter is a procedure that allows managed programs to access structured data and parameters from unmanaged code, thus it acts as a bridge between the managed GUI application and the unmanaged DLL driver to enable USB communication.

2.3 Software Quality Assessment

To allow for a software quality assessment, the GUI software application was fictitiously partitioned into the structure shown in Table 1.

Table 1: Structure of the GUI software used in our software quality assessment

Screen	Part of screen
1. Main screen	1.1. Input parameters
	1.2. Actions
	1.3. File management
	1.4. Shutdown
2. Setup screen	2.1. Input parameters
	2.2. Actions

The input parameters (i.e. the on-screen numerical keypad and the on-screen sliders) of the main and the setup screens (1.1. and 2.1. in Table 1) were tested with black-box methods using the test-case design approach [10]. Each parameter was assigned its value group defined as a valid or invalid equivalence class, i.e. a group of values that may or may not be used as input data. Then, the values from the valid and the invalid equivalence classes were used as input parameters and responses of the input-parameter handlers were determined: the values from the valid and the invalid equivalence classes were accepted or rejected by the GUI software. Also, the values near the edge between the valid and the invalid equivalence classes were additionally tested in order to determine responses of the input-parameter handlers at the boundary conditions. Moreover, the source code of the input parameters of both the main and the setup screens was examined with the white-box methods using the code-inspection approach in order to reveal memory-related (i.e. regarding memory allocation and data pointer management) error-prone situations.

The action on-screen buttons of the main and the setup screens (1.2. and 2.2. in Table 1) were tested with solely the white-box methods using the code-inspection approach [10]. All the relevant source code was examined and illegal actions (e.g. undisposed memory pointers that may lead to memory leaking) were marked as error-prone situations. The file management-related source code and the shutdown-related source code were examined using the code-inspection approach, as well.

3 Results

3.1 Hardware Platform

The developed high-voltage signal generator [7] is shown in Fig. 2.

Figure 2: Developed high-voltage signal generator (nanopulse electroporator)

Measurements of the used hardware were performed to confirm that pulses of the desired pulse parameters (amplitude, duration, number and repetition rate) are successfully generated [5]. Moreover, the maximal permissible currents were determined: the maximal current through the load is estimated at 1 A, while that through the shunt is estimated at 4 A. The output of the nanopulse electroporator is galvanically insulated from the grid voltage and can withstand 4 kVDC for a time period of 60 s. The leakage current is below 10 µA. Single and double faults of the device are evaluated and cannot hazard the load or the operator. The nanopulse electroporator resides in an electromagnetically-compatible enclosure; therefore, the device is both protected from the outer electromagnetic sources and neither interferes with other devices nor harms the load or the operator.

3.2 Software Platform

Fig. 3 shows a screenshot of the GUI application, which is the main application screen of the developed high-voltage signal generator.

Powering-on the device is followed first by invoking the operating system bootloader, then by loading the shellless operating system and finally by executing the GUI application. The absence of the operating system shell and automatic execution of the GUI application prevent the end-user from accessing the operating system features.

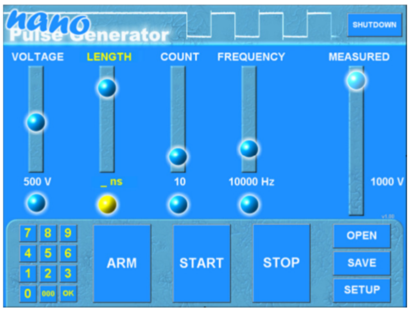

Figure 3: Developed GUI of the high-voltage signal generator. The figure shows entering the length pulse parameter indicated with no value before the ns label and the lightened selector button below the label

The GUI also includes a setup screen which allows calibration of the hardware (i.e. of the output power stage of the device); the setup screen is shown in Fig. 4. Calibration of the output stage is simple: an oscilloscope is used to measure the output pulses on the microelectrodes. For each pulse parameter (voltage or duration) needed to be set in the setup screen (Fig. 4) the user generates the output pulse by pressing the on-screen button "Arm and test". Then, the user inputs the measured value (voltage or duration) in the appropriate parameter input box using the on-screen numerical keypad. Finally, the software algorithms automatically calculate all the calibration parameters using the predefined multi-section linear equations and adjust the software accordingly.

3.3 *Software Quality Assessment*

Our black-box-based quality assessment of the input parameters (i.e. the on-screen numerical keypad and the on-screen sliders) of the main and the setup screens (1.1. and 2.1. in Table 1) revealed no errors present when entering these input parameters. However, the code inspection of the input-parameter handler source code showed serious memory-related errors: a memory leakage as a consequence of undisposed memory pointers was found and eliminated, thus resulting in an increased quality of the developed GUI software.

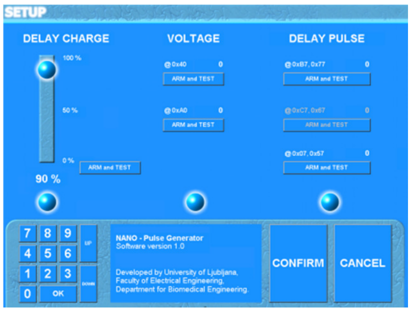

Figure 4: Setup screen of the GUI

Furthermore, inspecting the code of the action on-screen buttons of both main and setup screens (1.2. and 2.2. in Table 1) revealed weak error-handling algorithms developed for the USB communication; in a series of command transfers via USB, only the last command was positively tested for a successful acknowledgement. To avoid this deficiency, a more sophisticated acknowledge-detecting algorithm was applied.

In the file management-related source code there was a critical error found when saving configuration parameters in the externally-attached USB flash disk; the error situation was induced if the user cleared the filename after the filename had already been an empty string. In such a case, the GUI would cease to function with the hardware reset being the only option to recover. The error was successfully eliminated by thoroughly testing the software.

Moreover, procedures of risk-analysis and risk-management were applied to further improve safety of the nanopulse electroporator. By using these procedures, hazards representing security threats were identified [11]. Evaluation of the examined possible hazards is shown in Table 2.

Table 2: Possible hazards evaluated in the risk analysis procedure

Hazard	Category of likehood	Safety region
The output voltage is higher than set and expected.	Improbable	ALARP [1]
The user intentionally tries to raise voltage above the limits.	Improbable	ALARP [1]
Voltage is present on the device case.	Incredible	Acceptable
Pulses get generated without a user command executed.	Incredible	Acceptable
The discharge command does not function when needed.	Incredible	Acceptable

[1] *ALARP – as low as reasonably practicable.*

The hazards listed in Table 2 can be classified as of critical severity, since their occurrence can cause serious injury due to electric shock. However, as their category of likehood proves to be improbable or incredible, the safety regions they are included into are as low as reasonably practicable (ALARP) or acceptable, respectively. Therefore, there are no hazards that impose safety faultiness. Our risk-evaluation procedures were performed according to standard directives [12].

4 DISCUSSION

We developed a compact nanopulse electroporator to be used in microelectrode cell electroporation. Because of the embedded touch-screen display and especially because the Graphical User Interface (GUI) software does not include any hardware parameters, but only user-relevant parameters that are necessary for operation [13], the device is easy to use. Moreover, the layout of the GUI is designed intuitively, which means the device can be operated without knowledge of electronics or computer software. Hence, learning of how to operate the nanopulse electroporator is not necessary.

As the nanopulse electroporator is a biomedical device, its safety was an important consideration to be made. Though not designed to be used in medicine, the hardware of our nanopulse electroporator meets the imposed medical safety standards. Moreover, to assure its safe and risk-free use, a comprehensive software analysis was performed compliably with specifications of the IEC 601-1-4 standard [12].

When performing a software analysis, finding errors is actually considered to be a success [10]. In a detailed software-quality assessment, several minor and major errors were found and eliminated, thus making our GUI even more robust and safer. The device reliability was particularly improved by comprehensively analyzing its software and by reducing the remaining safety risks identified by using the risk-management approach [11]. The latter made us apply certain measures to reduce risk occurrence. These measures involved obligatory wearing of low-conductance gloves while operating the device, or even training in risk management, specifically in safety assurance measures.

ACKNOWLEDGMENTS

This work was supported by the Slovenian Research Agency (ARRS) and was conducted in the scope of the Electroporation in Biology and Medicine (EBAM) European Associated Laboratory (LEA). The authors thank Assoc. Prof. J. Bojkovski, Ph.D., from University of Ljubljana (Faculty of Electrical Engineering) for the discussion on software evaluation.

REFERENCES

[1] G. Pucihar *et al.*, Equivalent pulse parameters for electroporation, IEEE Transactions on Biomedical Engineering. 58 (2011) 3279-3288.

[2] T. Kotnik *et al.*, Induced Transmembrane Voltage and Its Correlation with Electroporation-Mediated Molecular Transport, J Membr Biol. 236 (2010) 3-13.

[3] K. Schoenbach *et al.*, The Effect of Intense Subnanosecond Electrical Pulses on Biological Cells, IEEE Trans Plasma Sci. 36 (2008) 414-422.

[4] M. Puc *et al.*, Techniques of signal generation required for electropermeabilization: Survey of electropermeabilization devices, Bioelectrochemistry. 64 (2004) 113-124.

[5] D. Pavliha *et al.*, A Graphical User-interface Controller for the Biomedical High-voltage Signal Generator, Electrotechnical Review. 78 (2011) 79-84.

[6] A. Kopustinskas *et al.*, A user-friendly PC-based data acquisition and analysis system for respirometric investigations, Comput Meth Prog Bio. 82 (2006) 231-237.

[7] M. Reberšek *et al.*, Blumlein Configuration for High-Repetition-Rate Pulse Generation of Variable Duration and Polarity Using Synchronized Switch Control, IEEE Trans Biomed Eng. 56 (2009) 2642-2648.

[8] K. Garmer *et al.*, User Participation in Requirements Elicitation Comparing Focus Group Interviews and Usability Tests for Eliciting Usability Requirements for Medical Equipment: A Case Study, Int J Ind Ergon. 33 (2004) 85-98.

[9] M. Suleman *et al.*, Data marshaling for multicore systems, IEEE Micro. 31 (2011) 56-64.

[10] G.J. Myers, The Art of Software Testing, John Wiley & Sons, Inc., 2004.

[11] D.S. Herrmann, Software Safety and Reliability: Techniques, Approaches, and Standards of Key Industrial Sectors, 1st ed., Wiley-IEEE Computer Society Pr, 2000.

[12] IEC, 601-1-4. Medical electrical equipment. Part 1: General requirements for safety 4. Collateral Standard: Programmable electrical medical systems, (1996).

[13] M. Heymann, A. Degani, Formal Analysis and Automatic Generation of User Interfaces: Approach, Methodology, and an Algorithm, Hum Factors. (2007) 311-330.

Denis Pavliha received his B.Sc. degree in electrical engineering from the University of Ljubljana, Slovenia, in 2009. He is currently working towards his Ph.D. degree in electrical engineering.

Matej Reberšek received his Ph.D. degree in electrical engineering from the University of Ljubljana, Slovenia, in 2008. He is currently a Research Associate in the Laboratory of Biocybernetics, Faculty of Electrical Engineering, University of Ljubljana. His current research interests include electroporation, especially design of electroporation devices and investigation of biological responses to nanosecond electrical pulses.

Damijan Miklavčič received his Ph.D. degree in electrical engineering from the University of Ljubljana, Slovenia, in 1993. He is currently a Professor at the Faculty of Electrical Engineering, University of Ljubljana, where he is also the Head of the Laboratory of Biocybernetics and Chair of the Department for Biomedical Engineering. His current research interests include electroporation-assisted drug and gene delivery, including cancer treatment by means of electrochemotherapy, and modeling.