

Načrtovanje in ovrednotenje kakovosti programske opreme grafičnega uporabniškega vmesnika visokonapetostnega signalnega generatorja

Denis Pavliha, Matej Reberšek in Damijan Miklavčič

Univerza v Ljubljani, Fakulteta za elektrotehniko, Tržaška 25, 1000 Ljubljana, Slovenija
E-pošta: denis.pavliha@fe.uni-lj.si

Povzetek. Članek opisuje načrtovanje, izdelavo in preizkus elektroprotorja, tj. naprave za elektroporacijo bioloških celic, ki zmore proizvesti nanosekundne (tj. dolžine med 40 in 200 ns) visokonapetostne (do 1 kV) električne pulze. Elektroprotor je krmiljen z uporabo zanesljivega, hitro odzivnega in uporabniku prijaznega grafičnega uporabniškega vmesnika, ki temelji na arhitekturi osebnih računalnikov. Grafični uporabniški vmesnik je načrtovan za krmiljenje z uporabo dotika, saj je kot vmesnik za interakcijo človek-stroj uporabljen zaslon, občutljiv na dotik. Ker je varnost pomemben dejavnik v načrtovanju biomedicinskih naprav, smo izvedli temeljito ovrednotenje kakovosti programske opreme. Vsi vhodni parametri (tj. programski vmesniki za zaslonke kontrole) grafičnega uporabniškega vmesnika so bili temeljito preizkušeni z uporabo postopkov *črne škatlice* in s pomočjo načrtovanja preizkusnih primerov (angl. *test-case design approach*) ter z uporabo postopkov *bele škatlice* in s pomočjo pregleda izvorne kode programske opreme (angl. *code inspection*). Našli smo večje in tudi manjše napake in jih izločili že v fazi razvoja naprave, zato je izdelani elektroprotor varen.

Ključne besede: elektroporacija, pulzna močnostna elektronika, grafični uporabniški vmesnik, preizkušanje vgrajene programske opreme, pregled izvorne kode

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

In this study we present design, construction and evaluation of an electroprotor 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 electroprotor.

1 UVOD

Elektroporacija je pojav, ki nastane, če biološko celico izpostavimo električnemu polju, ki ga povzročijo visokonapetostni električni pulzi s trajanjem v območju od nano- do milisekund [1]. Prisotnost zunanega električnega polja povzroči poracijo bioloških celic: prepustnost celične membrane se začasno poveča, kar molekulam iz okolice omogoča vstop v celico [2]. V zadnjem času je posebna pozornost namenjena

nanosekundnim pulzom (tj. nanopulzom), saj njihovo delovanje povzroči poracijo ne le plazmaleme, temveč tudi membran notranjih celičnih organelov[3]. Za proizvodnjo nanopulzov so potrebne nove naprave (elektroprotorji), saj izhodne stopnje uveljavljenih elektroprotorjev ne zmorejo proizvesti pulzov, krajših od 1 μ s [4].

Ker mikroelektrodni nanopulzni elektroprotorji še niso dostopni na trgu, smo načrtovali in izdelali elektroprotor, ki lahko proizvaja električne pulze s parametri (amplituda med 250 in 1000 V, dolžina med 40 in 200 ns, število med 1 in 100 in ponavljalna frekvenca med 1 in 100.000 Hz) za eksperimentalno uporabo pod mikroskopom z mikroelektrodami. Čeprav se tovrstni poskusi izvajajo le v laboratoriju, smo težili k načrtovanju robustne in varne naprave, ki bi bila obenem preprosta za uporabo. Pri načrtovanju biomedicinskih naprav so kritični naslednji vmesniki: vmesnik med strojno in programsko opremo, vmesnik med programsko opremo in uporabnikom ter vmesnik med strojno opremo in tkivom.

Vmesnik med strojno in programsko opremo je operacijski sistem (OS), ki gosti programsko opremo grafičnega uporabniškega vmesnika (GUI). GUI vpeljuje v proces načrtovanja naprav t. i. *strojne abstrakcije* (angl. *hardware abstraction layer – HAL*); HAL omogoča načrtovalcu ločitev platforme programske opreme od strojne platforme naprave. Tovrstna ločitev govori v prid uporabniku, saj ta lahko

uporablja kompleksno napravo, kot je elektroporator, učinkoviteje in brez podrobnega znanja o lastnostih ali notranji sestavi te naprave. Z vpeljavo HAL lahko namreč izdelamo popolnoma delujoč GUI tako, da uporabnika ne izpostavimo strojnemu parametrom naprave. Da bi dosegli abstrakcijo strojne opreme, smo krmilnik GUI [5] izdelali na podlagi arhitekture osebnega računalnika (angl. *Personal Computer* – PC). V preteklosti je bil eden od načinov konstrukcije GUI izdelava računalniškega programa, ki je tekkel na namiznem operacijskem sistemu [6]; sami smo se odločili za uporabo osebnega računalnika kot izključno krmilnika GUI. Tako je lahko vsa programska oprema, ki teče na krmilniku, vključno z operacijskim sistemom, namenjena izključno upravljanju interakcije človek-stroj, za upravljanje pa ni potreben dodatni PC, kar pripomore h kompaktnosti same naprave.

Uporabniku prijazen GUI, ki smo ga vgradili v izdelani nanopulzni elektroporator [7], uporabniku omogoča popoln nadzor nad napravo (tj. vmesnik med programsko opremo in uporabnikom). Garmer je poudarila, da »lahko z izboljšano uporabnostjo vmesnikov zmanjšamo tako število nesreč in tudi skrajšamo čas, ki je potreben za učenje uporabe nove opreme« [8]. Teh smernic smo se držali pri izdelavi krmilnika GUI, ki deluje zanesljivo in se odziva uporabniku čim hitreje. Delovanje nekega sistema je namreč odvisno od operacijskega sistema in programske opreme, ki upravlja interakcijo človek-stroj. Mnogokrat je največji izziv pri razvoju naprav, kot so elektroporatorji, ravno izbira ustrezne platforme in nato izdelava programske opreme zanjo. Dobro izdelana programska oprema lahko namreč izboljša uporabniško izkušnjo pri ravnanju z napravo. Po drugi strani pa lahko vodi slabo izdelana programska oprema do slabe uporabniške izkušnje (tj. uporabnik doživlja napor pri uporabi naprave) in v najhujšem primeru celo do nesreč. Po izdelavi programske opreme je torej ključnega pomena, da ovrednotimo programsko opremo in ugotovimo, ali obstaja možnost napak, ki bi lahko privedle do kritičnih situacij in posledično poslabšale uporabniško izkušnjo.

Pri strojni platformi je zelo pomembno, da poleg krmilnika GUI namenimo pozornost tudi varnosti izhodne močnostne stopnje naprave, saj je izhodna stopnja podsklop naprave, kateremu je uporabnik neposredno izpostavljen (tj. vmesnik med strojno opremo in tkivom). Za varno uporabo v medicinskem okolju morajo medicinske naprave dosegati varnostne zahteve, ki jih opredeljujejo tehnični standardi (npr. *International Electrotechnical Commission* IEC-60601, medicinska elektronska oprema). Medicinske naprave morajo dosegati predvsem zahteve v zvezi z uhajavimi tokovi, galvansko ločitvijo, pogoji redne in enojne napake, elektromagnetnimi interferencami (EMI) in elektromagnetno kompatibilnostjo (angl. *Electromagnetic Compatibility* – EMC). Čeprav je izdelani elektroporator namenjen laboratorijski uporabi

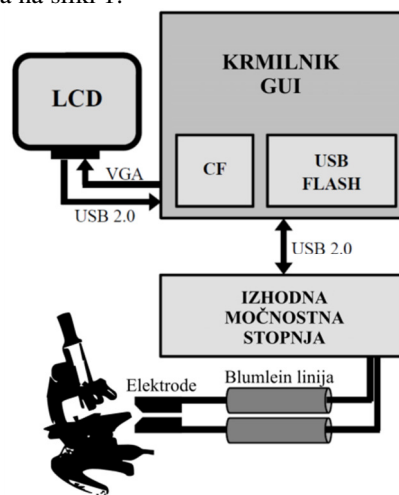
in ne v medicini, pa je bila varnost naprave eden glavnih ciljev med razvojem.

2 MATERIALI IN METODE

2.1 Strojna platforma

Strojna platforma izdelanega nanopulznega elektroporatorja [5] se sestoji iz dveh podsklopov: krmilnika GUI in izhodne močnostne stopnje.

Krmilnik GUI temelji na arhitekturi osebnih računalnikov (PC). Kot glavno strojno platformo krmilnika GUI smo uporabili matično ploščo formata mini-ITX, in sicer VIA Epia EN12000EG (VIA Technologies, Tajvan, 2001). Komunikacija med krmilnikom GUI in izhodno stopnjo je izvedena prek vodila Universal Serial Bus 2.0 (USB 2.0). Podatki se nahajajo na kartici CompactFlash (CF), ki je povezana z matično ploščo prek krmilnika Integrated Drive Electronics (IDE). Uporabljen je zaslon LCD, občutljiv na dotik; z matično ploščo je povezan prek povezav Video Graphics Array (VGA, za prikaz) in USB 2.0 (za krmiljenje z dotikom). Shema strojne platforme je prikazana na sliki 1.



Slika 1: Strojna platforma izdelanega nanopulznega elektroporatorja. (angl. *LCD* - *Liquid Crystal Display*, *VGA* - *Video Graphics Array*, *USB* - *Universal Serial Bus*, *GUI* - *Graphical User Interface*, *CF* - *CompactFlash*)

Naprava je nameščena v kompaktno elektromagnetno kompatibilno ohišje dimenzij 54 x 34 x 54 cm (širina krat višina krat globina) in tehta približno 30 kg. Uporabljeni zaslon je 15-palčni LCD, občutljiv na dotik (1537L, EloTouch, ZDA, 2005). Na prednji plošči se poleg zaslona LCD nahajajo še par priključkov *SubMiniature version A* (SMA), s katerimi na nanopulzni elektroporator priklopimo *linijo Blumlein* z elektrodami. Na prednji plošči je tudi tipka za vklop naprave. *Linija Blumlein* je nadgradnja pulzno-formirne linije: amplituda izhodne napetosti ustreza napajalni napetosti, impedanca bremena pa je dvakratnik impedance linije. *generator Blumlein* sestoji iz visokonapetostnega stikala, dveh koaksialnih linij in bremena med linijama. Nanopulzni elektroporator, ki smo ga izdelali, je modificiran *generator Blumlein* in

proizvaja nanosekundne visokonapetostne pulze spremenljivega trajanja in visoke ponavljalne frekvence s pomočjo štirih sinhroniziranih stikal.

2.2 Programska platforma

2.2.1 Operacijski sistem

Vsa programska oprema nanopulznega elektroporatorja temelji na operacijskem sistemu Microsoft Windows CE 5.0 .NET (Microsoft, ZDA, 2004). Za zagon operacijskega sistema je potreben sistemski zaganjalnik (angl. *bootloader*). Pri izdelavi operacijskega sistema na razvojni delovni postaji proizvede razvojno okolje Windows CE (imenovano Microsoft Platform Studio) datoteko, ki je zagonska koda izvornega sistema zaganjalnika Windows CE. Ta je izdelan za operacijski sistem MS-DOS (Microsoft, ZDA, 1981). Takšen postopek vpeljuje v zagon dodatno zakasnitev, saj mora biti torej pred sistemskim zaganjalnikom zagnan tudi MS-DOS. Zakasnitev smo zmanjšali z uporabo alternativnega sistema zaganjalnika: FastBoot (VIA Technologies, Tajvan, 2004) namreč omogoča nalaganje zagonske kode operacijskega sistema Windows CE brez posredovanja operacijskega sistema MS-DOS (tj. neposredno iz zapisa *Master Boot Record* – MBR kartice CF). Tako je zagonski postopek hitrejši od izvornega sistema zaganjalnika Windows CE.

Konfiguracija operacijskega sistema Microsoft Windows CE 5.0 .NET je bila izdelana po meri tako, da vključuje le komponente, ki so ključne za delovanje nanopulznega elektroporatorja. Vključene so naslednje komponente: osnovne komponente za omrežje (tj. *Network Driver Interface Specification - NDIS Support, TCP/IP, and Network Utilities*), komunikacijo prek USB (tj. *Enhanced Host Controller Interface - EHCI Support, USB Host Support*), shranjevanje podatkov (tj. *Storage Manager, FAT File System, USB Storage Driver*) in podporo ogrodju .NET (tj. *.NET Compact Framework 2.0*). Operacijski sistem podpira strojno platformo prek podpornega paketa *Board Support Package - BSP* (VIA Technologies, Tajvan, 2008); ta je dobavljiv pri izdelovalcu strojne platforme krmilnika GUI.

Operacijski sistem (OS) smo še dodatno prilagodili. Odstranili smo privzeto uporabniško lupino OS (angl. *shell*), kar pomeni, da v OS ni privzetega uporabniškega vmesnika (npr. namizja). Namesto tega smo izdelali program, ki deluje kot zaganjalnik programskega vmesnika GUI, napisanega z uporabo ogrodja .NET. Programski zaganjalnik GUI je zagnan takoj po zagonu grafičnega okolja (*Graphic Windowing and Events Subsystem - GWES*); napisan je v programskem jeziku C in zgrajen z uporabo Windows CE Platform Builder, ki je razvojno okolje sistema Microsoft Windows CE .NET 5.0. Programski zaganjalnik je potreben zato, ker neposredni klici programov, napisanih z uporabo ogrodja .NET, niso mogoči iz okolij, ki niso del tehnologije .NET; programski zaganjalnik je torej most za posredne klice programske opreme.

2.2.2 Programska oprema grafičnega uporabniškega vmesnika

Grafični uporabniški vmesnik (GUI) je glavni vmesnik nanopulznega elektroporatorja za interakcijo človek-stroj. GUI je aplikacija, izdelana z uporabo ogrodja Microsoft .NET, napisana v programskem jeziku Visual Basic .NET in zgrajena z uporabo Visual Studio 2005 Professional (Microsoft, ZDA, 2005). GUI je izdelan za krmiljenje z uporabo dotika in uporabniku omogoča nastavljanje pulznih parametrov (amplitude, časa trajanja, števila in ponavljalne frekvence) s pomočjo zaslonkih drsnikov ali zaslonke številčne tipkovnice. Izdelali smo dva načina vnosa, hitrega (z uporabo zaslonkih drsnikov) in natančnega (z uporabo zaslonke številčne tipkovnice) zaradi različnih uporabniških potreb glede na situacijo. Dejanja (tj. proženje naprave, generiranje pulzov in razbremenitev izhodne stopnje) so izvedena s pritiskom ustreznih zaslonkih tipk. Uporabnik lahko vse parametre naprave shrani na zunanje priklopljeni USB pomnilniški medij oz. jih z medija naloži v napravo. Parametri so shranjeni v obliki tekstovne datoteke, kar omogoča preprosto urejanje z uporabo osebnega računalnika.

Programska oprema, ki pomeni GUI, sestoji iz treh glavnih programskih podsklopov: uporabniškega vmesnika, jedra programske opreme in komunikacijskega vmesnika. Podsklop uporabniškega vmesnika vključuje kodo za delo z vidnimi zaslonkimi komponentami (npr. številčna tipkovnica, tipke za dejanja, drsniki parametrov) in zgolj beleži uporabnikova dejanja. Slednja nato obravnava jedro programske opreme, ki je izdelano kot t. i. sistem *polling*: uporabnikovo dejanje sproži izhodno dejanje naprave, torej so dogodki izvedeni asinhrono. Komunikacijski vmesnik skrbi za komunikacijo med GUI in izhodno močnostno stopnjo nanopulznega elektroporatorja z uporabo po meri izdelanega gonilnika izhodne močnostne stopnje prek vodila Universal Serial Bus 2.0 (USB 2.0). Vsi strojni ukazni (npr. za generiranje pulzov) so proizvedeni v izhodni močnostni stopnji; GUI zgolj odpošlje zelene parametre pulzov (tj. amplitudo, trajanje, število in ponavljalno frekvenco) skupaj z ukazom, medtem ko so vsi nadaljnji izračuni izvedeni v izhodni močnostni stopnji. Tako smo dosegli abstrakcijo strojne opreme.

Za pravilno delovanje nanopulznega elektroporatorja je potrebna kalibracija izhodne močnostne stopnje, saj moramo le-to prilagoditi krmilniku GUI. Kalibracija je mogoča prek nastavitvenega vmesnika, ki je del GUI in je zaščiten z geslom. Zaščita je izvedena zato, ker je kalibracija enkratni postopek, ki ga izvedejo razvojni inženirji naprave in ni mišljeno, da bi ga ponovno izvajali uporabniki.

2.2.3 Gonilnik za USB-povezavo izhodne stopnje

Gonilnike potrebujemo za podporo napravam, ki izvorno niso podprte v operacijskem sistemu Microsoft Windows CE 5.0 .NET. Izhodna močnostna stopnja nanopulznega elektroprotoratorja je s krmilnikom GUI povezana prek povezave Universal Serial Bus 2.0 (USB 2.0) in integriranega vezja Cypress EZ-USB FX2LP (CY7C68013A, Cypress, ZDA, 2002), ki nima podpore za operacijski sistem Microsoft Windows CE .NET 5.0, saj proizvajalec ne ponuja tovrstnih gonilnikov. Ker je pisanje gonilnikov zahteven postopek, smo napisali gonilnik s pomočjo razvojnega programskega paketa Windriver (Jungo Ltd, ZDA, 2008). Windriver namreč omogoča razvoj gonilnikov naprav za operacijski sistem Microsoft Windows CE .NET 5.0 glede na lastnosti naprave; ob priklopu naprave na razvojno delovno postajo Windriver pregleda napravo in izdela vsestransko uporabni gonilnik v številnih programskih jezikih.

Z Windriverjem izdelani gonilnik za Cypress EZ-USB FX2LP za operacijski sistem Microsoft Windows CE .NET 5.0 je bil razvit v programskem jeziku C z uporabo okolja Microsoft Embedded Visual C++ 4.0 (Microsoft, ZDA, 2004). Gonilnik smo še dodatno prilagodili tako, da omogoča zunanje klice funkcij, kar je nujno za delovanje nanopulznega elektroprotoratorja (npr. klici tipa *Vendor Request*, *GPIF Single Read* ipd). Gonilnik je v obliki knjižnice DLL (angl. *Dynamically-Linked Library*) in je krovni gonilnik jedrnih komponent orodja Windriver. Gonilnik DLL (ki je napisan v programskem jeziku C, torej ni .NET aplikacija) je dostopen v GUI (ki je .NET aplikacija) prek postopka, imenovanega *data marshaling* [9]. Ta omogoča programom v okviru ogrodja .NET (angl. *managed code*) dostop do strukturiranih podatkov in parametrov v kodi, ki ni napisana v okviru ogrodja .NET (angl. *unmanaged code*) in zatorej opravlja vlogo mostu med GUI in gonilnikom DLL, ki omogoča komunikacijo prek USB.

2.3 Ovrednotenje kakovosti programske opreme

Programsko opremo grafičnega uporabniškega vmesnika (GUI) smo za potrebe ovrednotenja kakovosti navidezno razdelili v strukturo, ki jo prikazuje tabela 1.

Tabela 1: Struktura programske opreme GUI za potrebe ovrednotenja kakovosti

Zaslon	Del zaslona
1. Glavni zaslon	1.1. Vhodni parametri
	1.2. Dejanja
	1.3. Delo z datotekami
	1.4. Izklop naprave
2. Nastavitveni zaslon	2.1. Vhodni parametri
	2.2. Dejanja

Vhodni parametri (tj. zaslonska številčna tipkovnica in zaslonski drsniki) tako glavnega kot nastavitvenega zaslona (1.1. in 2.1. v tabeli 1) so bili preizkušeni z

uporabo postopkov *črne škatlice* (angl. *black-box methods*) s pomočjo načrtovanja preizkusnih primerov (angl. *test-case design approach*) [10]. Vsak parameter je bil dodeljen vrednostni skupini, le-ta pa je bila opredeljena kot bodisi pravilen bodisi nepravilen razred vrednosti (angl. *valid or invalid equivalence class*), tj. skupina vrednosti, ki so lahko oz. ne smejo biti sprejete kot vhodni podatek. Vrednosti iz obeh razredov so bile nato uporabljene kot vhodni podatki vhodnih parametrov naprave in nato smo zabeležili odziv programskih podsklopov, ki obravnavajo te parametre: vrednosti iz pravih oz. nepravilnih razredov je programska oprema GUI potrdila oz. zavrnila. Dodatno pozornost smo namenili vrednostim na meji med razredoma, saj smo želeli preveriti odziv naprave pri robnih pogojih. Poleg tega smo temeljito pregledali tudi izvorno kodo podsklopov, ki obravnavajo vhodne parametre tako glavnega kot nastavitvenega zaslona, in sicer z uporabo postopkov *bele skrinjice* (angl. *white-box methods*) s pomočjo pregleda izvorne kode (angl. *code inspection*). Pregled smo izvedli zaradi odpravljanja težav, povezanih predvsem z dodeljevanjem pomnilnika.

Zaslonske tipke dejanj tako glavnega kot nastavitvenega zaslona (1.2. in 2.2. v tabeli 1) smo preizkusili le s postopki *bele skrinjice* s pomočjo pregleda izvorne kode [10]. Vsa relevantna izvorna koda je bila temeljito preverjena in nepravilna ravnanja (npr. napačno zavrženi pomnilniški kazalci, kar bi lahko povzročilo puščanje pomnilnika) so bila označena kot potencialno nevarne situacije. Izvorna koda, povezana z delom z datotekami in izklopom naprave, je bila prav tako preizkušena s pregledom izvorne kode.

3 REZULTATI

3.1 Strojna platforma

Izdelani visokonapetostni signalni generator [7] je prikazan na sliki 2.



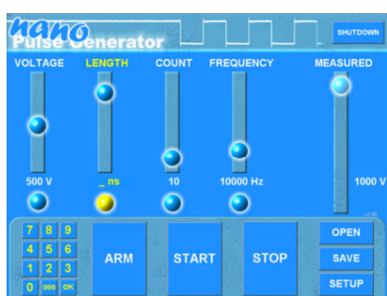
Slika 2: Izdelani nanopulzni elektroprotorator

Izvedli smo meritve strojne opreme [5]; meritve potrjujejo, da naprava proizvaja pulze z želenimi vrednostmi parametrov (amplituda, trajanje, število in ponavljalna frekvenca). Ocenili smo tudi največje mogoče tokove: največji tok čez breme je ocenjen na 1 A, medtem ko je največji tok čez soupore (angl. *shunt*) ocenjen na 4 A. Izhod nanopulznega elektroprotoratorja je

galvansko ločen od omrežne napetosti in lahko prenese 4 kVDC v časovnem obdobju 60 s. Uhajavi tokovi znašajo manj kot 10 μ A. Ovrednotene so vse mogoče enojne in dvojne napake naprave in ne pomenijo nevarnosti za uporabnika naprave ali breme. Nanopolzni elektroporator je vgrajen v elektromagnetno kompatibilno ohišje; naprava je torej zaščitena pred zunanji elektromagnetnimi viri, sama pa prav tako ne povzroča motenj zunanjim napravam oz. ne more poškodovati uporabnika ali bremena.

3.2 Programska platforma

Slika 3 prikazuje zaslon programske opreme GUI, ki je glavni uporabniški vmesnik izdelanega visokonapetostnega signalnega generatorja.



Slika 3: Izdelani GUI visokonapetostnega signalnega generatorja (glavni zaslon). Slika prikazuje vnos parametra dolžine pulza, kar nakazuje prazna vrednost pred enoto ns in osvetljen izbirni gumb pod zaslonkim drsnikom

Ob vklopu naprave se najprej zažene programski zaganjalnik (angl. *bootloader*) operacijskega sistema. Nato je zagnan operacijski sistem brez uporabniške lupine (angl. *shell*) in programska aplikacija GUI. Zaradi manjkajoče privzete uporabniške lupine in avtomatskega zagona GUI uporabnik ne more dostopati do nastavitvev operacijskega sistema.

GUI vključuje tudi nastavitveni zaslon, ki omogoča kalibracijo strojne opreme (tj. izhodne močnostne stopnje naprave); nastavitveni zaslon je prikazan na sliki 4. Kalibracija izhodne močnostne stopnje je preprosta: uporabnik z osciloskopom izmeri izhodne pulze na elektrodah. Za vsak pulzni parameter (napetost ali trajanje), ki je nastavljen v nastavitvenem zaslonu (slika 4), mora uporabnik zagnati generiranje kalibracijskega signala s pritiskom na zaslonko tipko »Arm and Test«. Nato mora uporabnik vnesti izmerjeno vrednost (napetost ali trajanje) v ustrezno vnosno okno na zaslonu z uporabo zaslonke številčne tipkovnice. Na koncu programski algoritmi samodejno izračunajo parametre kalibracije z uporabo vnaprej definiranih odsekovnih linearnih enačb in ustrezno prilagodijo programsko opremo.

3.3 Ovrednotenje kakovosti programske opreme

Ovrednotenje kakovosti vhodnih parametrov (tj. zaslonke številčne tipkovnice in zaslonkih drsnikov) tako glavnega kot nastavitvenega zaslona (1.1. in 2.1. v

tabeli 1) z uporabo postopkov *črne skrinjice* (angl. *black-box methods*) ni razkrilo napak pri vnosu omenjenih parametrov. Postopki *pregleda izvorne kode* (angl. *code inspection*) pa so razkrili napake v povezavi z dodeljevanjem pomnilnika: zaznali smo puščanje pomnilnika zaradi nepravilno zavrženih pomnilniških kazalcev. Napako smo nato odpravili, kar se odraža v izboljšani kakovosti programske opreme GUI.



Slika 4: Nastavitveni zaslon GUI

Naprej so postopki pregleda izvorne kode tako glavnega kot nastavitvenega zaslona (1.1. in 2.1. v tabeli 1) opozorili na površno zaznavanje napak pri komunikaciji prek vodila USB: pri nizu prenosa ukazov je bila morebitna napaka pri prenosu zaznana le pri zadnjem ukazu iz niza. Napako programske opreme smo odpravili tako, da smo izboljšali algoritem za zaznavo napak pri prenosu niza ukazov prek USB.

V izvorni kodi podsklopov za delo z datotekami smo zaznali in odpravili kritično napako, ki je nastala pri shranjevanju parametrov na zunanje priključeni pomnilniški medij USB: napaka je nastala, če je uporabnik poskusil počistiti ime datoteke potem, ko je to bilo že počiščeno. V tem primeru se je GUI nehal odzivati in edini način za ponovno pridobitev nadzora nad napravo je bil ponovni zagon. Napako smo tako zaznali in odpravili prav zaradi temeljitih postopkov ovrednotenja kakovosti programske opreme.

Poleg tega smo še dodatno izboljšali varnost nanopolznega elektroporatorja z vpeljavo postopkov analize in upravljanja tveganja (angl. *risk analysis and risk management*). Ta postopka sta uporabljena za identifikacijo situacij, ki pomenijo nevarnost pri uporabi naprave [11]. Zaznane nevarnosti smo ocenili v tabeli 2.

Tabela 2: Zaznane nevarnosti v analizi in upravljanju tveganja

Nevarnost	Verjetnost	Območje
Izhodna napetost je višja od pričakovane.	Manj verjetna	ALARP ¹
Uporabnik namenoma poskusi dvigniti napetost nad omejitvev.	Manj verjetna	ALARP ¹
Napetost je prisotna na ohišju naprave.	Neverjetna	Sprejemljivo
Pulzi so generirani brez uporabnikovega ukaza.	Neverjetna	Sprejemljivo
Ukaz za razbremenitev ne deluje, ko je nujno potreben.	Neverjetna	Sprejemljivo

¹ ALARP – majhno tveganje, katerega odprava ni smiselna (angl. *as low as reasonably practicable*)

Vse nevarnosti, naštetje v tabeli 2, lahko opredelimo kot resno kritične, saj lahko povzročijo resne poškodbe zaradi električnega udara. Ker pa je to manj verjetno ali neverjetno, jih lahko razvrstimo v območje sprejemljivih tveganj oz. tveganj, katerih odprava ni smiselna (angl. *as low as reasonably practicable* – ALARP). Povzamemo lahko, da ni nevarnosti, ki bi ogrožale varnost pri delu z napravo. Postopki ovrednotenja nevarnosti so bili izvedeni v skladu s standardnimi smernicami [12].

4 RAZPRAVA

Razvili smo kompaktni nanopulzni elektroporator za elektroporacijo celic z uporabo mikroelektrod. Uporaba naprave je preprosta, ker smo vgradili zaslon LCD, občutljiv na dotik, in izdelali grafični uporabniški vmesnik (GUI), ki uporabniku prikazuje le relevantne podatke, in ne strojnih parametrov naprave [13]. Zasnova GUI je intuitivno usmerjena, kar pomeni, da je z napravo mogoče rokovati brez predhodnega znanja elektronike ali računalništva. Zato učenje dela z nanopulznim elektroporatorjem ni potrebno.

Ker je nanopulzni elektroporator biomedicinska naprava, smo dali med razvojem poseben poudarek varnosti. Čeprav nanopulzni elektroporator ni namenjen uporabi v medicini, pa smo strojno platformo naprave vseeno načrtovali tako, da dosega standarde medicinske varnosti. Za zagotovitev varne uporabe naprave brez morebitnih tveganj smo nato izvedli temeljito ovrednotenje kakovosti vgrajene programske opreme v skladu s standardom IEC 601-1-4 [12].

Pri izvajanju analize programske opreme lahko najdene napake štejemo kot uspeh [10]. Zaradi temeljitega ovrednotenja programske opreme smo našli večje in manjše napake ter jih izločili že v fazi razvoja, zato je GUI robusten in varen. Zanesljivost naprave je bila v veliki meri izboljšana prav zaradi izvedene analize programske opreme; kljub temu pa lahko dodatna tveganja še zmanjšamo z uporabo postopka *upravljanja tveganja* (angl. *risk management*) [11]. Ta narekuje dodatna varnostna merila (npr. uporabnik nosi neprevodne zaščitne rokavice med rokovanjem z napravo) za zmanjšanje morebitnih tveganj. Kot zadnjo možnost lahko omenimo tudi učenje uporabnikov naprave glede varne uporabe naprave.

ZAHVALA

Delo je podprla Javna agencija za raziskovalno dejavnost Republike Slovenije (ARRS) in je bilo izvedeno v okviru evropskega združenega laboratorija (LEA) projekta Elektroporacija v biologiji in medicini (EBAM). Avtorji se zahvaljujejo izr. prof. dr. Jovanu Bojkovskemu (Univerza v Ljubljani, Fakulteta za elektrotehniko) za diskusijo o kakovosti programske opreme.

LITERATURA

[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.*, Krmilnik biomedicinskega visokonapetostnega signalnega generatorja z grafičnim uporabniškim vmesnikom, *Elektrotehniški vestnik*, 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 je diplomiral leta 2009 na Fakulteti za elektrotehniko Univerze v Ljubljani. Trenutno je zaposlen kot mladi raziskovalec na Fakulteti za elektrotehniko, kjer je habilitiran v naziv asistenta in vodi vaje s področja programiranja pri predmetih na dodiplomskem študiju. Trenutno je podiplomski študent (doktorski študij 3. bolonjske stopnje) in se raziskovalno ukvarja s programsko opremo za načrtovanje zdravljenja globlje ležečih tumorjev z elektrokemoterapijo.

Matej Reberšek je doktoriral leta 2008 na Fakulteti za elektrotehniko Univerze v Ljubljani, kjer je trenutno zaposlen kot raziskovalni sodelavec v Laboratoriju za biokibernetiko. Njegovo raziskovalno delo vključuje elektroporacijo, predvsem razvoj naprav, in proučevanje bioloških odzivov na nanosekundne električne pulze.

Damijan Miklavčič je doktoriral leta 1993 na Fakulteti za elektrotehniko Univerze v Ljubljani, kjer je trenutno zaposlen kot redni profesor in je predstojnik Laboratorija za biokibernetiko. Njegovo raziskovalno delo je v zadnjih letih povezano s prenosom genov in zdravilnih učinkovin pri zdravljenju raka z elektroporacijo, razvojem strojne opreme in numeričnim modeliranjem bioloških procesov. Je avtor več kot 150 znanstvenih člankov, urednik in pridružen urednik pri več znanstvenih revijah, član uredniških odborov in recenzent pri več kot 30 različnih znanstvenih revijah.