

Vladimir Simić, Vladimir Ćirić, Teufik Tokić, Ivan Milentijević, Oliver Vojinović, "Univerzalna hardversko-softverska platforma za realizaciju korisničkog i upravljačkog interfejsa uređaja za fizikalnu medicinu", Tehničko rešenje, Elektronski fakultet u Nišu, 2013.

TEHNIČKO REŠENJE

Univerzalna hardversko-softverska platforma za realizaciju korisničkog i upravljačkog interfejsa uređaja za fizikalnu medicinu

M-85: Prototip, nova metoda, softver, standardizovan ili atestiran instrument, nova genetska proba, mikroorganizmi

Autori:

Vladimir Simić⁽¹⁾, Vladimir Ćirić⁽¹⁾, Teufik Tokić⁽¹⁾, Ivan Milentijević⁽¹⁾, Oliver Vojinović⁽¹⁾

⁽¹⁾ Elektronski fakultet, Univerzitet u Nišu



Наставно-научно веће

Број: 07/05-010/14-001

Дана: 09.01.2014. год.

На основу члана 38. Статута Електронског факултета у Нишу, у складу са одредбама Правилника о поступку и начину вредновања, и квантитавном исказивању научноистраживачких резултата истраживача («Службени гласник РС», бр. 38/2008), Наставно-научно веће Електронског факултета у Нишу донело је на седници одржаној 09.01.2014. године следећу

ОДЛУКУ

1. Прихвата се техничко решење под називом «Универзална хардверско-софтверска платформа за реализацију корисничког и управљачког интерфејса уређаја за физикалну медицину», чији су аутори Владимир Симић, Владимир Ћирић, Теуфик Токић, Иван Милентијевић, Оливер Војиновић.
2. Признато техничко решење спада у категорију: софтвер – М85.
3. Одлуку доставити ауторима техничког решења и архиви Факултета.

Образложење

Комисија рецензента доставила је Наставно-научном већу Мишљење о испуњености услова за признање својства техничког решења резултату научноистраживачког рада под називом «Универзална хардверско-софтверска платформа за реализацију корисничког и управљачког интерфејса уређаја за физикалну медицину» који је реализован у оквиру пројекта ТР 32012 за потребе Електромедицине д.о.о. Ниш чији су аутори Владимир Симић, Владимир Ћирић, Теуфик Токић, Иван Милентијевић, Оливер Војиновић.

Наиме, рецензенти проф. др Вељко Малбаша, Факултет техничких наука у Новом Саду и проф. др Мирослав Трајановић, Машински факултет у Нишу оценили су да предложено техничко решење представља научни резултат који поред стручне компоненте пружа оригинални теоријски и научноистраживачки допринос. У том смислу рецензенти су предложили Наставно-научном већу Факултета да прихвати наведени резултат научноистраживачког рада као техничко решење.

На основу позитивног мишљења два рецензента-експерта из области техничког решења Наставно-научно веће је донело одлуку као у диспозитиву.

Председник
Наставно-научног већа,
З Д е к а н
Проф. др Драган Јанковић

Izjava korisnika o tehničkom rešenju

„Univerzalna hardversko-softverska platforma za realizaciju korisničkog i upravljačkog interfejsa uređaja za fizikalnu medicinu“

Autora: Vladimira Simića, Vladimira Ćirića, Teufika Tokića, Ivana Milentijevića, Olivera Vojinovića sa Elektronskog fakulteta, Univerziteta u Nišu

Yumis d.o.o. Niš je naručilac i korisnik rezultata projekta „Inteligentni KAbinet za Fizikalnu Medicinu – IKAFIM“ finansiranog sredstvima Ministarstva prosvete, nauke i tehnološkog razvoja Republike Srbije u okviru programa istraživanja u oblasti tehnološkog razvoja za period 2011-2014. godine (evidencioni broj projekta TR-32012).

Uzimajući u obzir rezultate sa ovog projekta, izjavljujemo da je Elektronski fakultet u Nišu kao realizator projekta u potpunosti ispunio očekivanja vezana za poboljšavanje karakteristika i unapređenje serije uređaja za fizikalnu terapiju. Implementirano rešenje predstavlja platformu koja će omogućiti realizaciju modernijeg korisničkog i upravljačkog interfejsa uređaja za fizikalnu terapiju uz zadržavanje postojećih funkcionalnosti. Nakon centralizacije upravljanja i povezivanja uređaja u mrežu, u ovoj fazi realizacije projekta uveden je i implemetiran korisnički i upravljački interfejs sa ekranom osetljivim na dodir. Od platforme se očekuje da će ubrzati dalji razvoj uređaja i omogućiti proširenje funkcionalnosti u budućnosti.

Izjavljujemo da tehničko rešenje pod nazivom **Univerzalna hardversko-softverska platforma za realizaciju korisničkog i upravljačkog interfejsa uređaja za fizikalnu medicinu** predstavlja novu platformu za nadgradnju uređaja za fizikalnu terapiju koju karakteriše moderniji korisnički i upravljački interfejs.

U Nišu, 30.12.2013.

Yumis d.o.o. Niš
Direktor



Jagoda Nikolić

Tehničko rešenje:

Univerzalna hardversko-softverska platforma za realizaciju korisničkog i upravljačkog interfejsa uređaja za fizikalnu medicinu

Rukovodilac projekta: prof. dr Ivan Milentijević

Autori: Vladimir Simić⁽¹⁾, Vladimir Ćirić⁽¹⁾, Teufik Tokić⁽¹⁾, Ivan Milentijević⁽¹⁾, Oliver Vojinović⁽¹⁾

⁽¹⁾ Elektronski fakultet u Nišu

Razvijeno: u okviru projekta tehnološkog razvoja TR-32012

Kratak opis

Ponovna upotrebljivost komponenata (engl. reusability) je veoma važan koncept projektovanja složenijih sistema. Kako bi se olakšala nadgradnja postojećih uređaja za fizikalnu medicinu uz minimalne troškove i napore a zatim obavila integracija u jedinstven sistem, razvijena je univerzalna hardversko-softverska platforma kao polazna osnova za realizaciju upravljačkih i korisničkih interfejsa različitih uređaja. Platforma je razvijena na Elektronskom fakultetu u Nišu za potrebe nadgradnje i međusobnog povezivanja medicinskih uređaja koji su predmet istraživanja na projektu "Inteligentni kabinet za fizikalnu medicinu", finansiran od strane Ministarstva prosvete, nauke i tehnološkog razvoja Republike Srbije pod brojem TR32012.

Tehničke karakteristike:

Hardverski deo platforme omogućuje da se podrže postojeće funkcionalnosti uređaja i doda mogućnost umrežavanja i realizacija modernijeg korisničkog interfejsa. Softverski deo platforme omogućuje da se projektuje upravljački i korisnički interfejs uređaja za fizikalnu medicinu uz stepen slobode pri izboru operativnog sistema, familije kontrolera i memorijskog sistema za trajno čuvanje podataka. U osnovi softverskog dela platforme je sistem za navigaciju kroz hijerarhijski organizovanu strukturu menija i konfiguracionih parametara sa podrškom za ekrane različitih veličina. Rešenjem je omogućeno iscrtavanje menija, interakcija sa korisnikom preko touch ekrana i podrška za sve identifikovane i buduće načine podešavanja i tipove parametara. Testiranje je obavljeno na touchscreen panelu rezolucije 320x240.

Tehničke mogućnosti:

U sklopu ovog rešenja napravljene su rutine za iscrtavanje različitih menija, uz obezbeđenu podršku za ekrane različite rezolucije. Omogućen je prikaz, unos i izmena konfiguracionih parametara uređaja. Tu spadaju znakovni podaci, brojčani podaci kao i parametri kod kojih se bira jedna iz skupa predefinisanih vrednosti. Kako su funkcije izdvojene u module obezbeđeno je da biblioteke funkcija budu nezavisne od operativnog sistema na kontroleru kao i memorijskog sistema u koji se konfiguracioni podaci zapisuju i iz koga se učitavaju po potrebi.

Realizator:

Elektronski fakultet u Nišu

Korisnik:

Elektromedicina d.o.o. Niš

Podtip rešenja:

Softver – M85

Mišljenje

Tehničko rešenje „Univerzalna hardversko-softverska platforma za realizaciju korisničkog i upravljačkog interfejsa uređaja za fizikalnu medicinu“ realizovano na Elektronskom fakultetu u Nišu, opisano je u okviru sledećih poglavlja:

1. Opis problema koji se rešava i prednosti tehničkog rešenja,
2. Stanje rešenosti problema u svetu
3. Arhitektura platforme,
4. Detaljan opis arhitekture i implementacije tehničkog rešenja,
5. Analiza performansi,

kojima su pridodate reference.

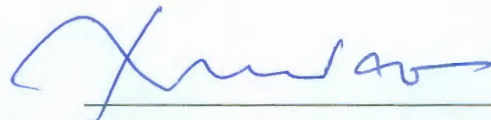
Tehničko rešenje je realizovano u okviru projekta "Inteligentni kabinet za fizikalnu medicinu ", broj projekta TR 32012. Program istraživanja projekta je u oblasti tehnološkog razvoja za period 2011-2014. godina, tehnološka oblast Elektronika, telekomunikacije i informacione tehnologije.

Tehničko rešenje pripada polju tehničko-tehnoloških nauka i oblasti elektrotehničkog inženjerstva. Korisnik rešenja je Elektromedicina d.o.o. Niš.

Na osnovu analize tehničkog rešenja "Univerzalna platforma za realizaciju korisničkog upravljačkog interfejsa uređaja za fizikalnu medicinu", mogu se izneti najbitniji zaključci:

1. Dokumentacija tehničkog rešenja kroz četiri poglavlja prikazuje pogodnosti koje ovakav sistem ima kod nadogradnje i implementacije upravljačkih interfejsa uređaja za fizikalnu medicinu.
2. U tehničkom rešenju je predstavljena hardverska i softverska platforma kojom se omogućava implementacija osnovnog hardvera konkretnih uređaja i realizuje korisnički interfejs na ekranima osetljivim na dodir.
3. Tehničko rešenje ima svoju praktičnu dimenziju s obzirom na to da implementiran univerzalni interfejs sa neophodnim grafičkim elementima za upravljanje i nadgledanje procesa terapije na uređaju za fizikalnu terapiju. Takođe, predstavljeno tehničko rešenje može poslužiti kao sistem za realizaciju kompletnih interfejsa sistema slične namene.

Na osnovu priložene dokumentacije, dajem predlog da se tehničko rešenje pod nazivom "Univerzalna hardversko-softverska platforma za realizaciju korisničkog i upravljačkog interfejsa uređaja za fizikalnu medicinu" navedenih autora prihvati kao novo tehničko rešenje i u skladu sa Pravilnikom o postupku i načinu vrednovanja, i kvantitativnom iskazivanju naučnoistraživačkih rezultata istraživača ("Službeni glasnik RS", broj 38/2008) klasifikuje kao rezultat "M85 Prototip, nova metoda, softver, standardizovan ili atestiran instrument, nova genska proba, mikroorganizmi".



Prof. dr Veljko Malbaša

Fakultet tehničkih nauka, Novi Sad

Tehničko rešenje:

Univerzalna hardversko-softverska platforma za realizaciju korisničkog i upravljačkog interfejsa uređaja za fizikalnu medicinu

Rukovodilac projekta: prof. dr Ivan Milentijević

Autori: Vladimir Simić⁽¹⁾, Vladimir Ćirić⁽¹⁾, Teufik Tokić⁽¹⁾, Ivan Milentijević⁽¹⁾, Oliver Vojinović⁽¹⁾

⁽¹⁾ Elektronski fakultet, Univerzitet u Nišu,

Razvijeno: u okviru projekta tehnološkog razvoja „Inteligentni KAbinet za Fizikalnu Medicinu - IKAFIM“ TR-32012

Kratak opis

Tehničko rešenje obuhvata projektovanje i implementaciju hardversko-softverske platforme koja je polazna osnova kod nadgradnje uređaja za fizikalnu medicinu. Hardverski deo sistema pruža interfejs ka postojećim analognim delovima uređaja a softverski deo sistema obezbeđuje skup biblioteka za implementaciju sistema menija karakterističnih za pojedine uređaje. Na taj način je smanjena cena i vreme razvoja novih prototipova i omogućena realizacija kompletnih korisničkih u upravljačkih interfejsa za sve uređaje.

Tehničke karakteristike:

Softverski sistem platforme realizovan je u MikroC programskom jeziku. Hardverska platforma bazirana je na Microchip PIC32 familiji kontrolera. Korisnički u upravljački interfejs realizuje se preko touchscreen ekrana rezolucije 320x240 piksela.

Tehničke mogućnosti:

Softver na kontroleru omogućava realizaciju korisničkih interfejsa na touchscreen ekranu. Nakon kalibracije, omogućen je postupak pravljenja hijerarhijske strukture menija koji se zatim prikazuje na ekranu. Podržani su različiti formati podataka i konfiguracionih parametara i u skladu sa tim moguće je promeniti izgled svakog dela menija.

Realizator:

Elektronski fakultet u Nišu

Korisnik:

Elektromedicina d.o.o. Niš

Podtip rešenja:

Softver – M85

Mišljenje

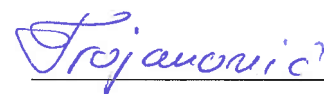
Tehničko rešenje „Univerzalna hardversko-softverska platforma za realizaciju korisničkog i upravljačkog interfejsa uređaja za fizikalnu medicinu“ realizovano je u okviru projekta „Inteligentni KAbinet za Fizikalnu Medicinu – IKAFIM“.

U uvodnom delu dokumenta dat je prikaz stanja i trendova u oblasti proizvodnje medicinskih uređaja s akcentom na primeni savremenih informaciono-komunikacionih tehnologija i razvoju platformi za implementaciju medicinskih uređaja. Hardverske platforme medicinskih uređaja su vrlo različite pa je veliki pomak učinjen u drugom smeru, ka razvoju univerzalnih softverskih platformi sa podrškom za različite familije procesora. Ovim rešenjem se postavlja razvojna platforma za implementaciju različitih uređaja za fizikalnu medicinu. Softverska platforma realizuje korisnički interfejs a hardverska platforma pruža podršku za postojeće uređaje i dodaje funkcije umrežavanja.

Centralni deo dokumenta detaljno opisuje strukturu predloženog rešenja. Rešenje je razdvojeno na dva podsistema. Softverski sistem omogućava projektovanje korisničkog interfejsa kao hijerarhijske strukture menija i obezbeđuje biblioteke za iscrtavanje, navigaciju i interakciju sa delovima menija. Razvijeni skup biblioteka je nezavistan od ostatka softverskog dela sistema pa može da podrži različite operativne sisteme. Hardverski deo sistema sadrži interfejs ka kontroleru, sistemske tastere i mogućnost dodavanja spoljnih hardverskih modula.

Rešenje je iskorišćeno kao polazna osnova pri implementaciji nadgradnje uređaja za fizikalnu medicinu s ciljem da se smanje troškovi i vreme izrade. Rešenje je primenljivo kod razvoja sličnih sistema baziranih na ekranima veće rezolucije kada je potrebno implementirati hijerarhijsku strukturu menija i grafički korisnički interfejs koji obezbeđuje funkcije navigacije, prikaza i izmene konfiguracionih parametara uređaja.

U skladu sa Pravilnikom o postupku i načinu vrednovanja, i kvantitativnom iskazivanju naučnoistraživačkih rezultata istraživača ("Službeni glasnik RS", broj 38/2008), a na osnovu analize tehničkog rešenja, predlažem da se tehničko rešenje pod nazivom „Univerzalna hardversko-softverska platforma za realizaciju korisničkog i upravljačkog interfejsa uređaja za fizikalnu medicinu“ usvoji kao novo tehničko rešenje iz kategorije "M85 Prototip, nova metoda, softver, standardizovan ili atestiran instrument, nova genska proba, mikroorganizmi".



prof. dr Miroslav Trajanović,
Mašinski fakultet Niš

1. Kratak opis

Tehničke karakteristike:

Kompanija Elektromedicina d.o.o. Niš nudi paletu terapijskih uređaja za fizikalnu medicinu: Eksposan, Intermed, Sonoton, Magnemed, Vakumed i Diaton. U okviru projekta "Inteligentni kabinet za fizikalnu medicinu" na Elektronskom fakultetu u Nišu razvijena je nova generacija ovih uređaja sa mogućnošću umrežavanja i daljinske kontrole i akvizicije podataka preko Ethernet mreže. Prva unapređena generacija uređaja je bazirana na mikrokontroleru i ima korisnički interfejs na LCD ekranu veličine 2x16 karaktera. Ovo tehničko rešenje daje implementaciju uređaja na mikrokontroleru novije generacije, uz uvođenje modernijeg korisničkog interfejsa sa ekranom osetljivim na dodir. Razvijen je skup biblioteka koje realizuju iscrtavanje korisničkog interfejsa uređaja za fizikalnu terapiju sa podrškom za ekrane osetljive na dodir. S obzirom na to da različiti uređaji imaju različite parametre terapije (frekvencija ultrazvuka kod Sonotona, širina i broj impulsa kod Eksposana, jačina magnetnog polja kod Magnameda i sl.) korisničkim interfejsom, datim u ovom tehničkom rešenju, obezbeđena je univerzalnost i mogućnost korišćenja istog hardvera za implementaciju različitih uređaja. Izbor tipa uređaja koji implementira jedinstveni hardver, uz korišćenje zajedničkih grafičkih elemenata, omogućen je izmenama u konfiguracionim fajlovima. Razvijen je prototip uređaja sa kontrolerom PIC32MX460F512L i ekranom osetljivim na dodir rezolucije 320x240 sa mogućnošću prikaza 262 hiljade različitih boja.

Tehničke mogućnosti:

Prototip uređaja baziran je na kontroleru PIC32MX460F512L sa ekranom osetljivim na dodir, rezolucije 320x240 i sa mogućnošću prikaza 262 hiljade različitih boja. U sklopu ovog rešenja kreirane su funkcije za iscrtavanje korisničkog interfejsa na grafičkom ekranu. Identifikovani su konfiguracioni parametri različitih uređaja za fizikalnu terapiju i omogućen je njihov prikaz, unos i izmena. Konfiguracioni i radni parametri uređaja mogu biti znakovni podaci, brojni podaci kao i parametri kod kojih se bira jedna iz skupa predefinisanih vrednosti. Realizovanim skupom funkcija omogućeni su navigacija kroz hijerarhijski opisan sistem menija i podešavanje uređaja preko ekrana osetljivog na dodir.

Realizator:

Elektronski fakultet u Nišu

Korisnik:

Elektromedicina d.o.o. Niš

Podtip rešenja:

Softver - M85

Projekat u okviru koga je realizovano tehničko rešenje:

Program istraživanja u oblasti tehnološkog razvoja za period 2011.-2014.

Tehnološka oblast:	Elektronika, telekomunikacije i informacione tehnologije
Rukovodilac projekta:	Prof. dr Ivan Milentijević
Naziv projekta:	Inteligentni KAbinet za Fizikalnu Medicinu - IKAFIM
Broj projekta:	TR 32012

2. Stanje u svetu

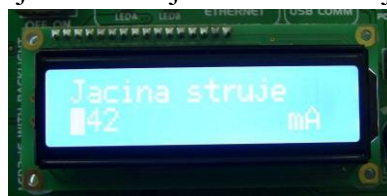
Proizvođači medicinskih uređaja teže ka razvoju pouzdanih, bezbednih, inteligentnih i povezanih uređaja. Jednom proizveden, uređaj ostaje u vlasništvu korisnika desetinu godina ili duže. Česta potreba u ovakvim slučajevima je mogućnost hardverske ili softverske nadgradnje postojećih uređaja, kako bi se iskoristile pogodnosti aktuelnih tehnologija i standarda, ili mogućnost dodavanja novih funkcionalnosti. To nije uvek izvodljivo i zavisi od načina projektovanja i implementacije sistema u celini, uključujući hardversku strukturu samog uređaja, hardversku platformu, korišćene komponente, korišćene analogne delove i protokole za povezivanje i razmenu podataka. Ako su uređaji kontrolisani regulatorima fizičkih veličina ili mehaničkim delovima dodatni izazov predstavlja prelazak na neki od savremenijih korisničkih i upravljačkih interfejsa, kao što su ekrani osetljivi na dodir.

Polazna osnova ovog tehničkog rešenja su uređaji kompanije Elektromedicina, koji su predmet istraživanja projekta "Inteligentni kabinet za fizikalnu medicinu – IKAFIM", koji se realizuje na Elektronskom fakultetu u Nišu, a većim delom finansira sredstvima Ministarstva prosvete, nauke i tehnološkog razvoja Republike Srbije. To su uređaji Eksposan, Intermed, Sonoton, Magnemed, Vakumed i Diaton. Uređaji za fizikalnu terapiju koji su predmet projekta u originalnoj verziji sadrže ekran za prikaz statusnih informacija, a kompletna kontrola i signalizacija na uređajima obavlja se mehaničkim delovima i svetlosnim indikatorima lokalno na samom uređaju. Na slici 2.1 prikazan je uređaj Intermed.



Slika 2.1. Uređaj Intermed

U okviru pomenutog projekta implementirana je nova generacija uređaja, bazirana na mikrokontroleru, sa korisničkim interfejsom koji koristi ekran sa 2x16 karaktera za prikaz statusnih informacija. Podešavanje parametara uređaja obavlja se preko tastera na samom uređaju. Takođe, novorazvijena generacija uređaja ima mogućnost povezivanja na centralni server preko Ethernet mreže, u cilju centralizacije upravljanja i akvizicije podataka. Razvijen je mrežni protokoli za komunikaciju medicinskih uređaja za fizikalnu terapiju sa centralnim računarom radi prikupljanja informacija i centralizacije upravljanja. Na slikama 2.2 i 2.3 može se videti deo korisničkog interfejsa na uređaju Intermed 2 koji koristi 2x16 ekran.



Slika 2.2. Izgled displeja prototipa uređaja Intermed 2 za odabir jačine struje



Slika 2.3. Izgled displeja prototipa uređaja Intermed 2 za odabir frekvencije

Predmet ovog tehničkog rešenja je dalje unapređenje korisničkog interfejsa na uređajima uvođenjem ekrana osjetljivog na dodir, čime se korisniku pruža moderniji način kontrole uređaja i prikaza informacija. Na taj način, uređaj se oslobađa mehaničkih tastera za navigaciju kroz korisnički interfejs. Tehničko rešenje opisano u nastavku predstavlja univerzalnu hardversku platformu koja je osnov za nadgradnju postojećih uređaja za fizikalnu terapiju. Projektovana hardverska platforma čini osnovu pomenutih medicinskih uređaja za fizikalnu terapiju, i pored ostalih funkcija implementira korisnički interfejs.

Kod rešenja predstavljenog u ovom dokumentu korišćen je ekran osjetljiv na dodir. Razvoj medicinskih platformi i tehnologija ekrana osjetljivih na dodir je u stalnom razvoju i aktuelne su teme istraživanja [1]. Grafički sistemi su danas sveprisutni u ljudskom životu, od pametnih telefona do medicinskih uređaja. Ekran osjetljiv na dodir je posebna vrsta uređaja koja reaguje na pritisak ili dodir. Takav uređaj pored prikazivanja informacija na ekranu može detektovati i poziciju tačke dodira. Zbog takvih svojstava ekrana osjetljivog na dodir omogućava se direktna interakcija sa sistemom koji se kontroliše. Interakcijom sa ekranom korisniku se pružaju brojne mogućnosti, pa se smatra da će ekrani osjetljivi na dodir u bliskoj budućnosti u velikoj meri zameniti tastaturu i miša. Jedna od prednosti ekrana na dodir je dostupnost programskog interfejsa i korišćenja njihovih funkcija na skoro svim elektronskim uređajima, ali i mogućnost korišćenja na komponentama malih dimenzija kao što su mikrokontroleri [2].

Razvoj tehnologija ekrana osjetljivih na dodir je u stalnom napretku. Petožični otporni sistem (engl. 5-Wire Resistive) je jedna od tehnologija koja se i danas često upotrebljava [3]. Osim ove tehnologije danas su u upotrebi kapacitivni sistemi (engl. capacitive), sistemi koji koriste površinske talase (engl. Surface Acoustic Wave), sistemi koji koriste infracrveni spektar (engl. Infrared), tehnologiju disperzivnog signala (engl. Dispersive Signal Technology (DST)), optičku tehnologiju (engl. Optical touch ekran technology) i dr. [4]. Za potrebe testiranja platforme korišćen je ekran sa četvoro-žičnom tehnologijom.

Bez obzira na područje primene, osnovne prednosti koje tehnologija osjetljivih ekrana pruža su te da ekran osjetljiv na dodir

- zbog intuitivnosti interfejsa skraćuje vreme potrebno za obučavanje;
- smanjuje mogućnost greške operatera, jer nudi izbor između jasno definisanih opcija;
- eliminiše potrebu za tastaturom i mišem;
- takvi terminali su izdržljiviji od miša i tastature;
- pružaju mogućnost direktne interakcije sa bilo kojim tipom multimedije;
- smanjuju zauzeće prostora na radnim površinama.

Od nedostataka mogu se identifikovati:

- Prsti nisu precizni kao miš, tako da elementi prikazani na ekranu moraju zadovoljiti uslov za minimalnom veličinom, kako bi se omogućilo efikasno korišćenje uređaja, jer specijalna olovka nije uvek praktična za upotrebu;
- Ukucavanje većeg broja slova ili brojeva je naporno, pa takvi ekrani nemaju veliku primenu tamo gde se unosi veća količina teksta ili brojeva, mada postoje rešenja koja kombinuju multi-touch tehnologiju i tzv. virtualne tastature koje se iscertavaju preko postojećih elemenata interfejsa;
- Povlačenje (engl. drag & drop) nije lako izvršiti već se više preferira pokazivačka interakcija.

3. Hardversko-softverska platforma medicinskih uređaja za fizikalnu terapiju sa grafičkim displejem osjetljivim na dodir

U okviru projekta „Inteligentnog kabineta za fizikalnu medicinu“ izvršena je analiza načina upravljanja i administracije medicinskih uređaja da bi se kreirao efikasan i intuitivni korisnički interfejs, koji će korisnicima omogućiti da putem ekrana osjetljivog na dodir u najmanjem broju koraka podese uređaj i zadaju terapiju. Ovo tehničko rešenje opisuje arhitekturu platforme koja predstavlja univerzalnu osnovu za implementaciju medicinskih uređaja za fizikalnu terapiju sa grafičkim displejem osjetljivim na dodir.

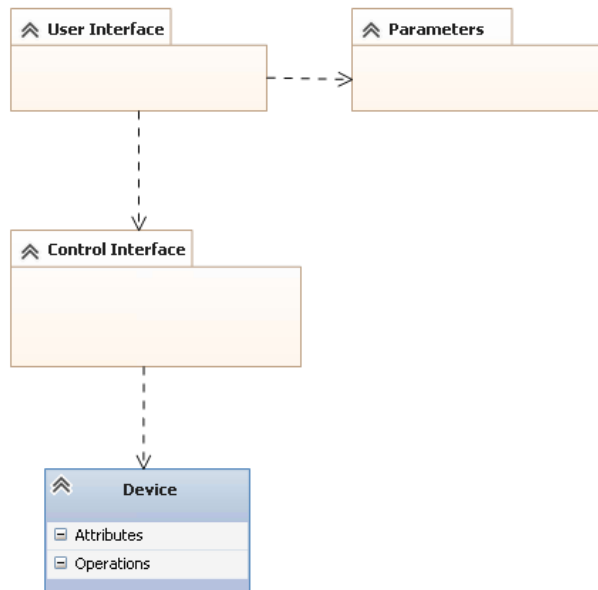
U arhitekturi je mogće izdvojiti dva logička dela. Prvi deo je hardverski, i čini ga štampana ploča sa mikrokontrolerom i ekranom osjetljivim na dodir. Drugi deo je softverski, koga čini skup biblioteka i rutina za iscertavanje korisničkog interfejsa u vidu menija, rutina za navigaciju kroz menije preko ekrana osjetljivog na dodir, i za izmenu i snimanje konfiguracionih parametara. Univerzalnost se pre svega odnosi na mogućnost da se sa realizovanim skupom funkcija može organizovati proizvoljna struktura menija i podržati različit skup tipova konfiguracionih parametara. Implementacija dozvoljava dodavanje novih tipova podataka i odgovarajućih rutina za iscertavanje. Implementacija je nezavisna od memorijskog sistema koji se koristi za čuvanje konfiguracionih parametara.

3.1 Arhitektura platforme

Dve glavne prednosti kod realizacije ovakve platforme ogledaju se u obezbeđivanju podrške za sve popularnije grafičke interfejse bazirane na ekranima osjetljivim na dodir, kao i mogućnosti da se interfejs i hardverska struktura prilagodi konkretnim uređajima koji se implementiraju na platformi. Da bi ti bilo moguće, posebna pažnja posvećena je projektovanju arhitekture softvera i parametrizaciji najbitnijih delova, kako bi se samo promenom tipa uređaja u konfiguraciji softvera implementirao drugi tip uređaja.

Funkcije za unos, pregled i izmenu konfiguracionih parametara, koje obezbeđuje upravljački interfejs, razvijene su tako da obezbede prilagođavanje tipu ekrana koji se koristi. Grafički interfejs obezbeđuje navigaciju kroz hijerarhijske nivoe menija, pregled i izmenu parametara. Posebna pažnja posvećena je obezbeđivanju podrške za različite tipove podataka i parametara koji se mogu sresti kod uređaja za fizikalnu terapiju. Uglavnom se vrši unos brojevanih podataka, vrši se izbor jedne od više ponuđenih opcija, a u retkim slučajevima potrebno je unositi tekstualni sadržaj. Kako je kompletan interfejs prebačen sa mehaničkih tastera na ekran, platforma je zadržala samo mali broj mehaničkih tastera za obavljanje servisnih procedura kao što su resetovanje uređaja, brisanje sadržaja memorije, resetovanje mrežne komunikacije i slično. Platforma se povezuje na Ethernet mrežu pa je na samoj ploči obezbeđen odgovarajući priključak.

Arhitektura se sastoji iz tri sloja: korisničkog interfejsa, podsistema za definisanje konfiguracionih parametara i kontrolnog interfejsa (slika 3.1). Korisnički interfejs sadrži rutine za iscertavanje i navigaciju kroz različite tipove menija i grafičkih elemenata na bazi parametara definisanih u okviru komponente Parameters. Kontrolni interfejs definiše način kojim se akcije korisnika iz grafičkog interfejsa povezuju sa konkretnom komandom na uređaju predstavljenim komponentom Device. Skup funkcija realizovan je u programskom jeziku MikroC/C. Delovi korisničkog interfejsa realizovani su korišćenjem Visual TFT alata za grafičko projektovanje.

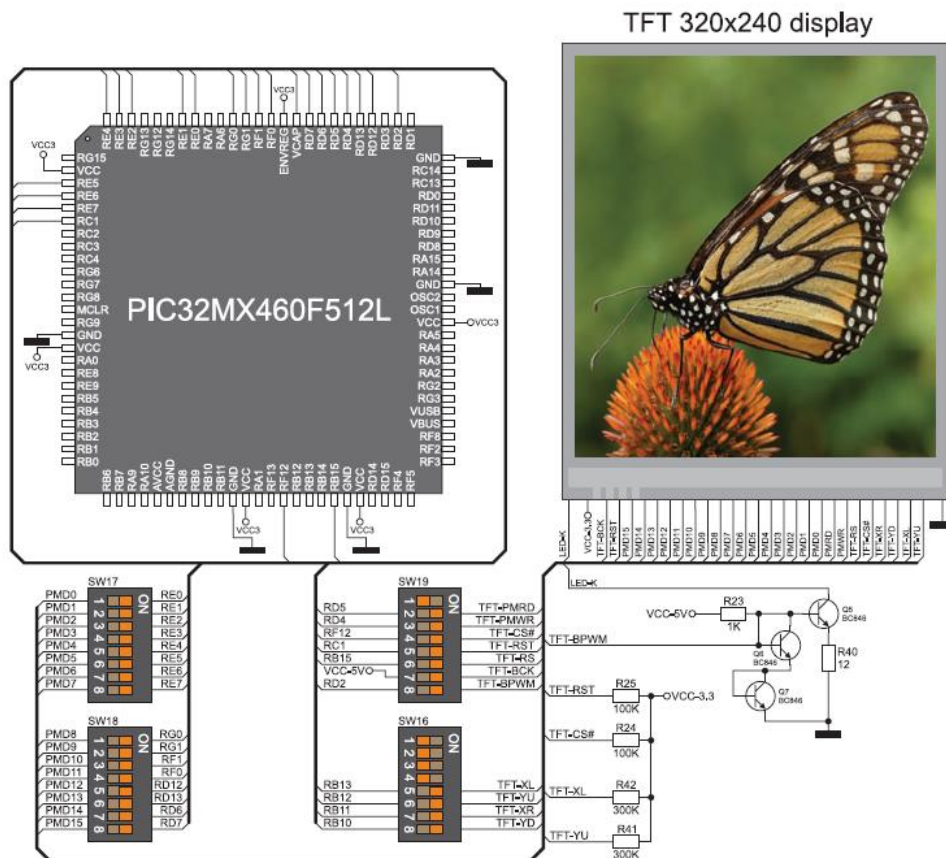


Slika 3.1 Arhitektura korisničkog i upravljačkog interfejsa

3.2 Hardverski deo platforme

Za opis ovog tehničkog rešenja od interesa je u kratkim crtama predstaviti hardverski deo sistema koji realizuje korisnički upravljački interfejs. Korišćeni model 32-bitnog mikrokontrolera nudi visoke performanse i dolazi sa bogatim skupom alata potrebnih za razvoj sistema. Sa slobodnim softverom, jeftinim razvojnim alatima, njegovom kompatibilnošću sa 16-bitnim proizvodima, potrebnom procesorskom snagom, memorijom i vezom sa perifernim uređajima za dizajnerske potrebe PIC32 daje dobre rezultate. Za razvoj je korišćen mikrokontroler PIC32MX460F512L na MCU kartici firme Mikroelektronika d.o.o. Ovaj mikrokontroler pripada familiji PIC32MX4xx, ima radni takt do 80Mhz, programsku memoriju do 512KB, memoriju za podatke do 32KB, RAM memoriju do 32KB, naponski opseg od 2.3V do 3.6V, temperaturni opseg od -40°C do 125°C i kompatibilnost sa 16-bitnim PIC MCU karticama. Za razvoj hardverskog dela platforme osim kontrolera iskorišćen je i TFT ekran osetljiv na dodir rezolucije 320x240 tačaka koji može da prikaže 262 hiljade boja.

Šema povezivanja prikazana je na slici 3.2.



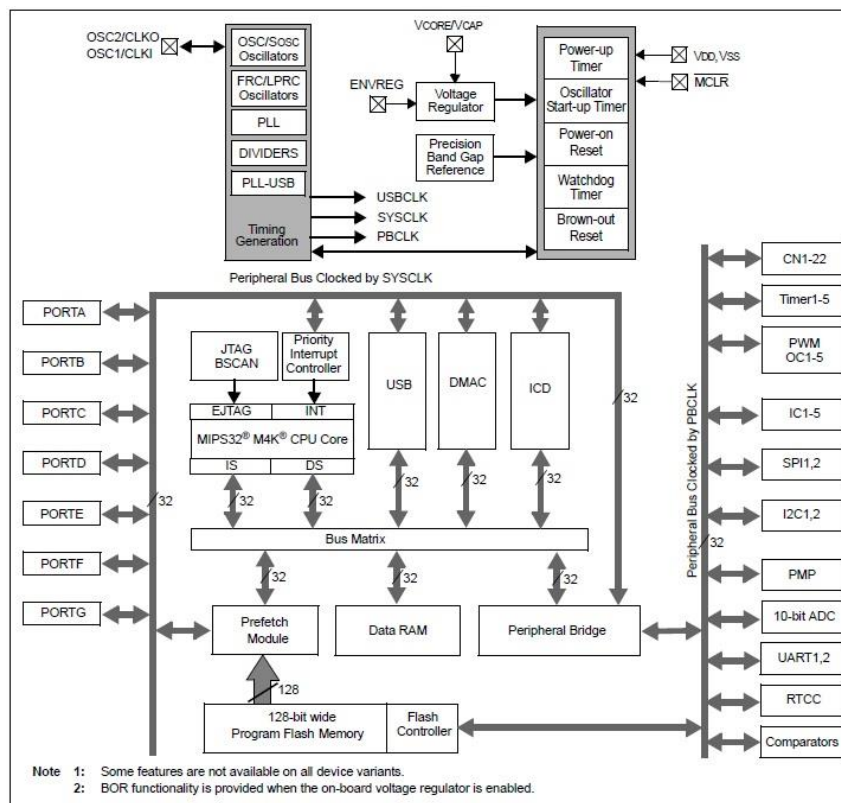
Slika 3.2 Šema povezivanja ekrana osjetljivog na dodir

3.3 Korisnički i upravljački interfejs

U cilju kreiranja efikasnog i intuitivnog korisničkog interfejsa, koji će korisnicima omogućiti da putem ekrana osjetljivog na dodir u najmanjem broju koraka podese uređaj i zadaju terapiju, izvršena je analiza načina rukovanja i administracije medicinskih uređaja za fizikalnu terapiju. U nastavku je dat prikaz načina na koji je interfejs projektovan.

3.3.1 Razvojno okruženje

Od razvojnih okruženja korišćeni su programski alati MikroC PRO za familiju kontrolera PIC32 i VisualTFT kompanije Mikroelektronika [5]. Projektovanje je usmereno ka implementaciji interfejsa koji zadovoljava principe HCI dizajna. U realizaciji sistema je korišćen razvojni sistem LV32MXv6 [6], napredni razvojni sistem za PIC32MX mikrokontrolere. Za razvoj i implementaciju korisničkog i upravljačkog interfejsa izabran je mikrokontroler PIC32MX460F512L [7] i TFT ekran u boji rezolucije 320x240. Na slici 3.3 prikazana je blok šema ovog mikrokontrolera.

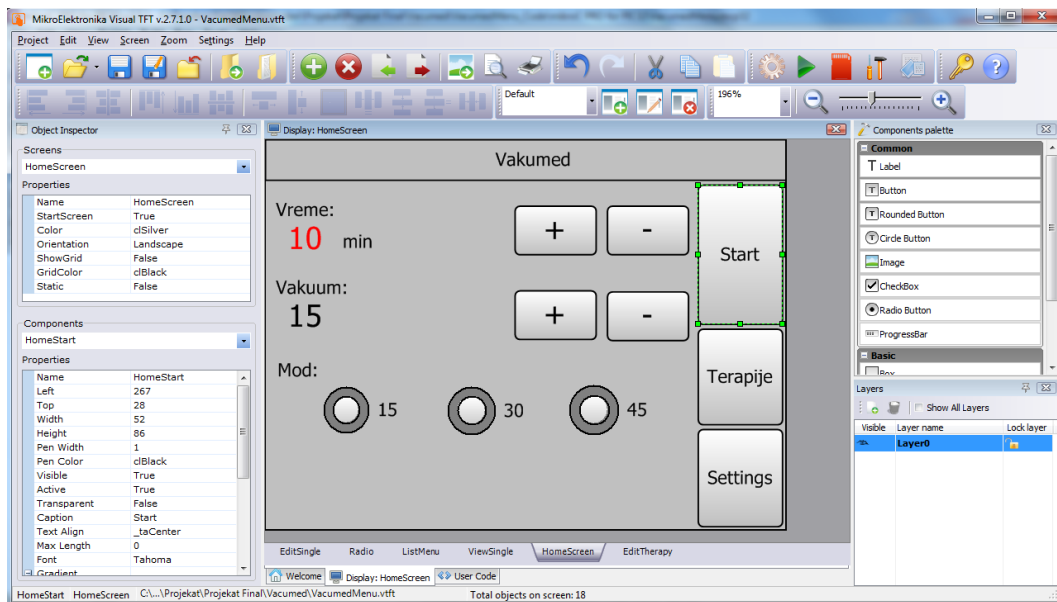


Slika 3.3 Blok dijagram mikrokontrolera

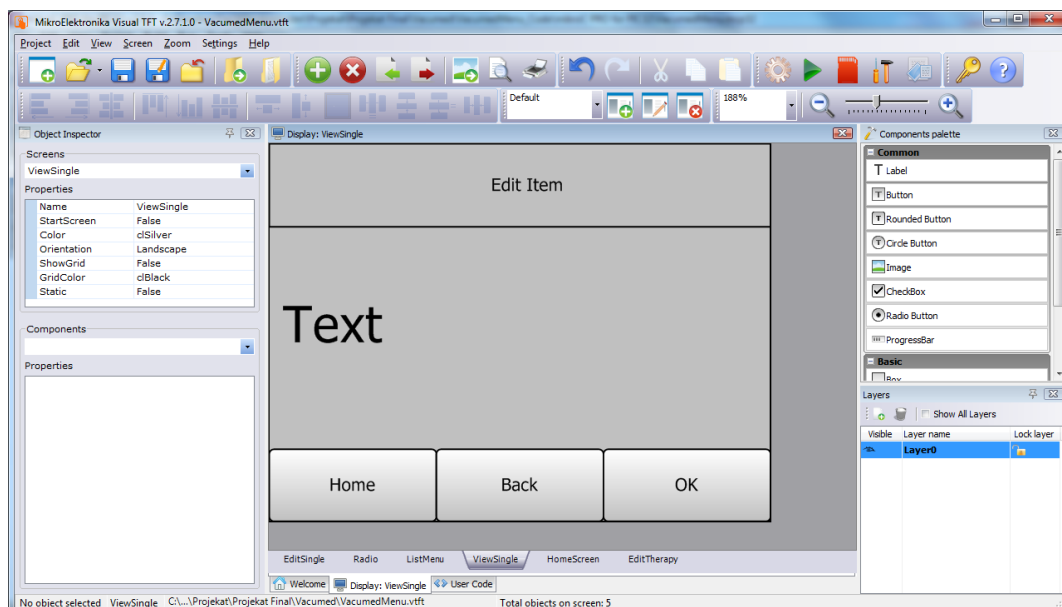
3.3.2 Izrada interfejsa

Kako bi se olakšala implementacija korisničkog interfejsa na mikrokontroleru korišćen je alat VisualTFT. VisualTFT je programski alat za kreiranje grafičkog interfejsa na PIC32MX mikrokontrolerima. Omogućava definisanje neograničenog broja ekrana i nudi vizuelno okruženje za postavljanje objekata na ekrane i izmenu njihovih svojstava, čime se rad sa TFT ekranom umnogome pojednostavljuje.

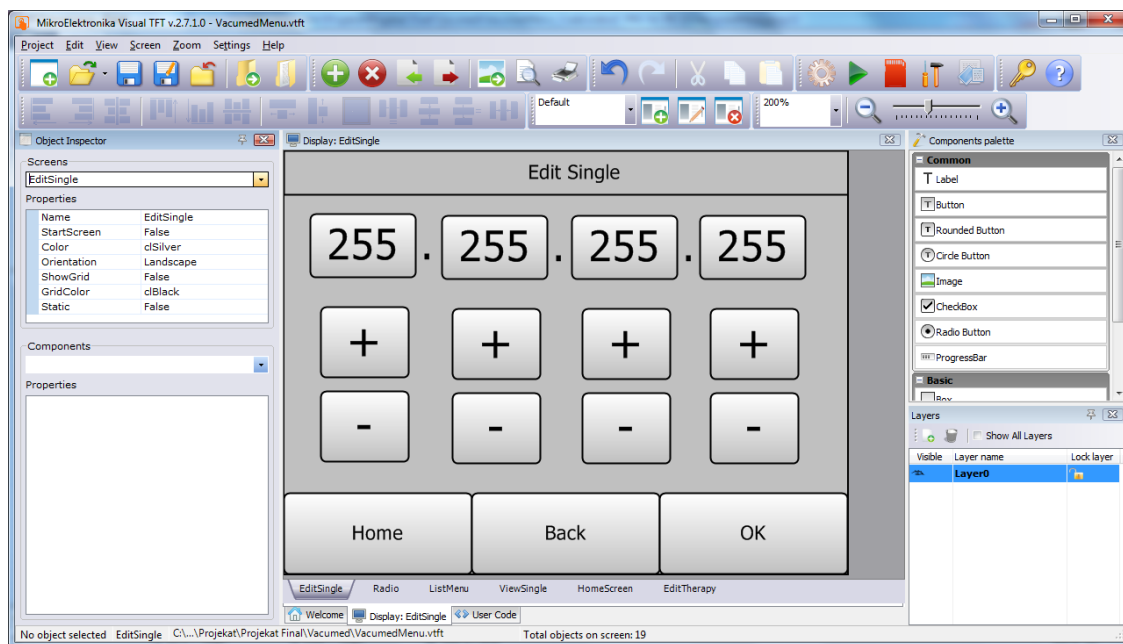
U nastavku je prikazano nekoliko različitih ekrana koji su deo korisničkog interfejsa uređaja Vakumed (slike 3.4 – 3.10). Vakumed je uređaj koji radi na mrežnom naponu od 220V i mrežne frekvencije od 50Hz sa maksimalnom potrošnjom struje od 300mA. Uređaj podržava impulsni i konstantni režim rada. Podešavanje napona se vrši ručno (linearno) i postoje 3 načina rada za impulsni režim. Na samom uređaju se može podešavati impulsni režim u opsegu 15, 30 ili 45 i jačina vakuum-a (od 0 - 6000 mbar). Vakuum se postiže vakumskim pumpama koje se priključuju na uređaj i predstavljaju analogni deo uređaja kojim upravljaju elektronski sklopovi uređaja.



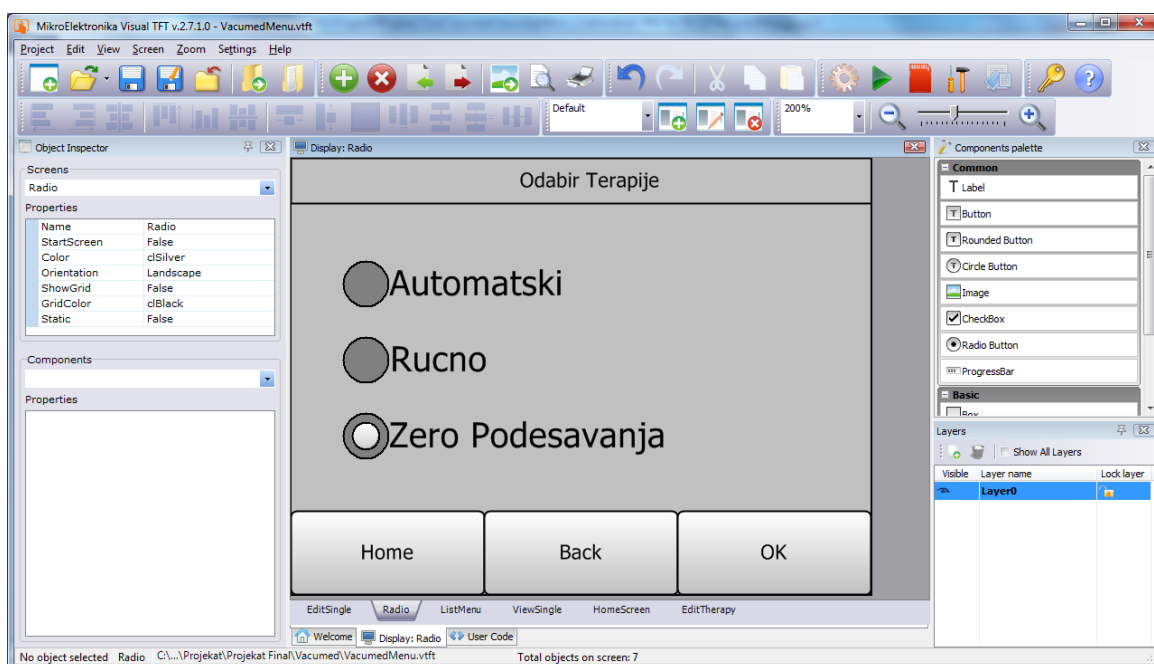
Slika 3.4 Dizajn interfejsa za izmenu terapija na uređaju



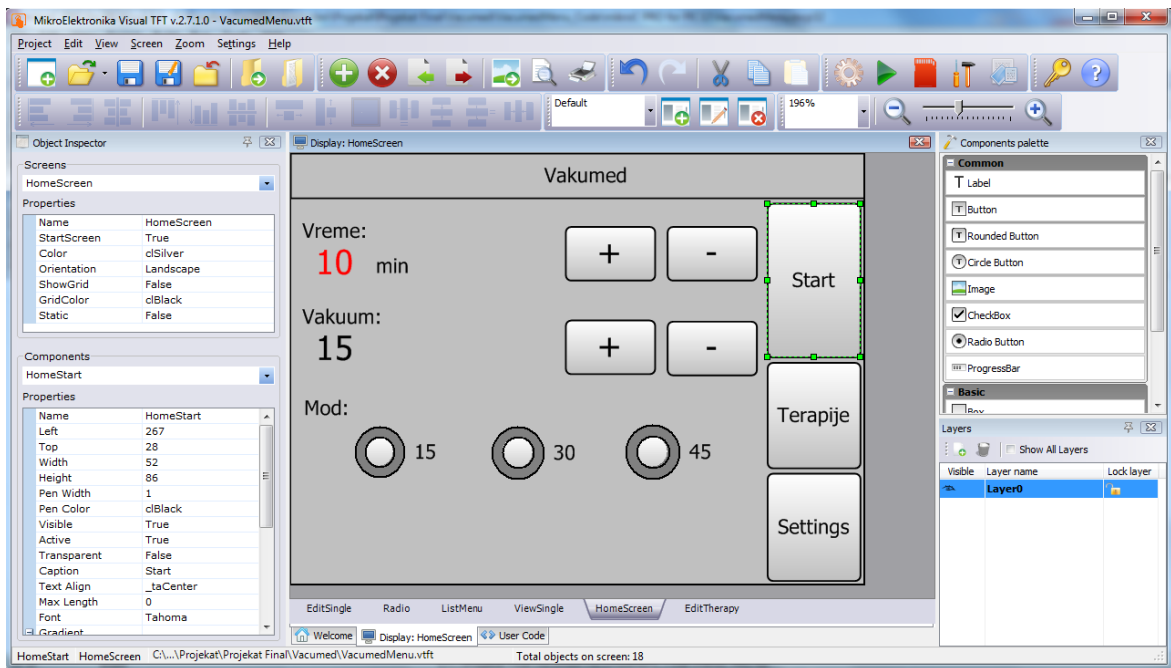
Slika 3.5 Ekran za prikaz nepromenljivog teksta poput MAC adrese, ukupnog broja startovanih terapija, serijskog broja uređaja itd.



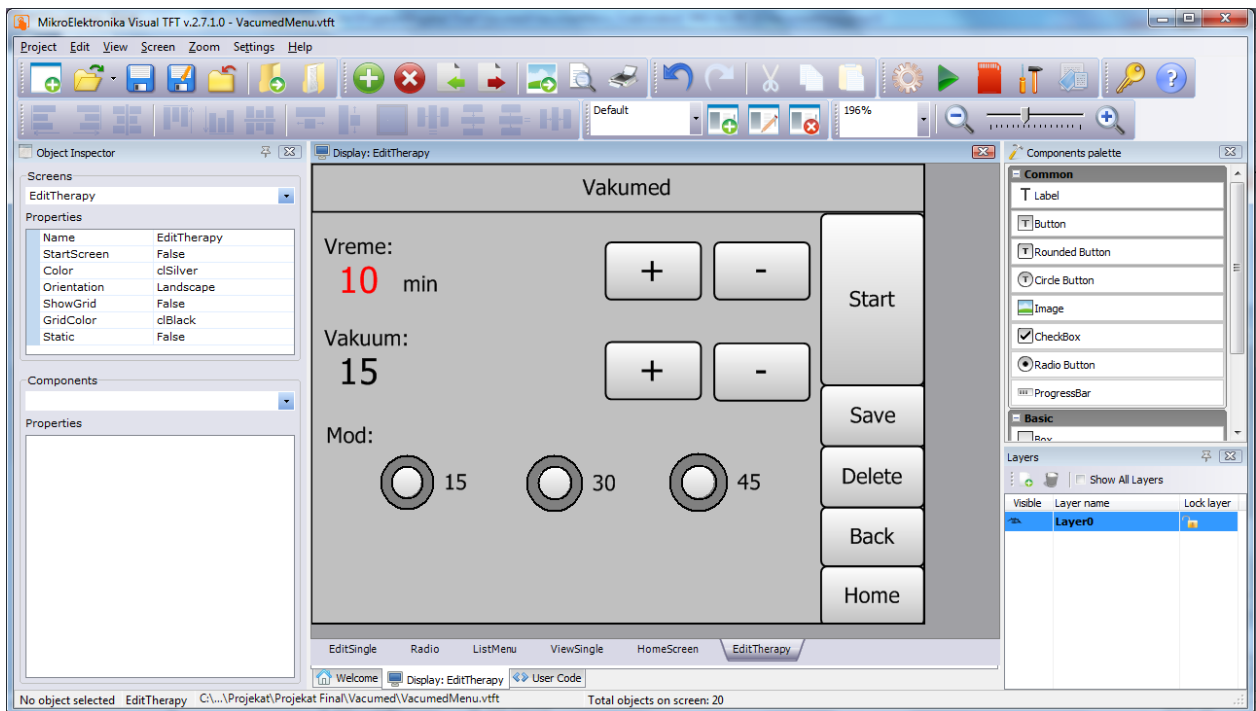
Slika 3.6 Ekran za izmenu aktivne IP adrese na uređaju. Ovaj korisnički interfejs iskorišćen je za podešavanje IP adrese, mrežne maske, adrese gejtveja i adrese DNS servera na uređaju



Slika 3.7 Ekran za izbor jedne od predefinisanih opcija. Ovaj interfejs iskorišćen je za odabir načina podešavanja IP adrese na uređaju, korišćenjem ručno podešene IP adrese, DHCP-a ili zeroconf-a

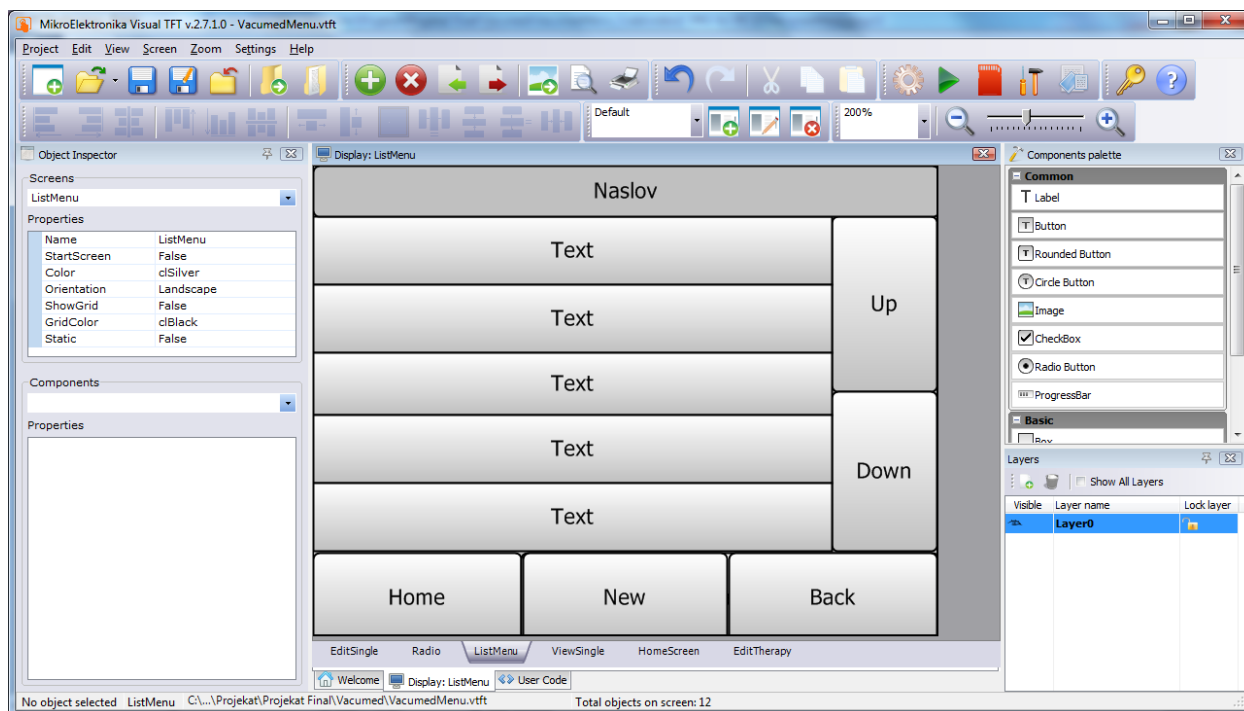


Slika 3.8 Početni ekran koji prikazuje poslednju korišćenu terapiju ili trenutno aktivnu terapiju



Slika 3.9 Ekran za izmenu terapije zapamćene na samom uređaju

Prethodna dva ekrana poseduju tastere kojima se podešavaju radni parametri trenutno aktivne terapije, kao i taster za startovanje terapije. Startni ekran (Slika 3.8) poseduje tastere koji vode do dodatnih opcija uređaja (Terapije, Settings), dok ekran za izmenu terapije poseduje tastere kojima se pamte izmene u terapiji, briše terapija ili vraća na prethodnu stavku u navigaciji.



Slika 3.10 Interfejs za prikaz listi iz koje treba izabrati jednu od opcija ili dodati novu

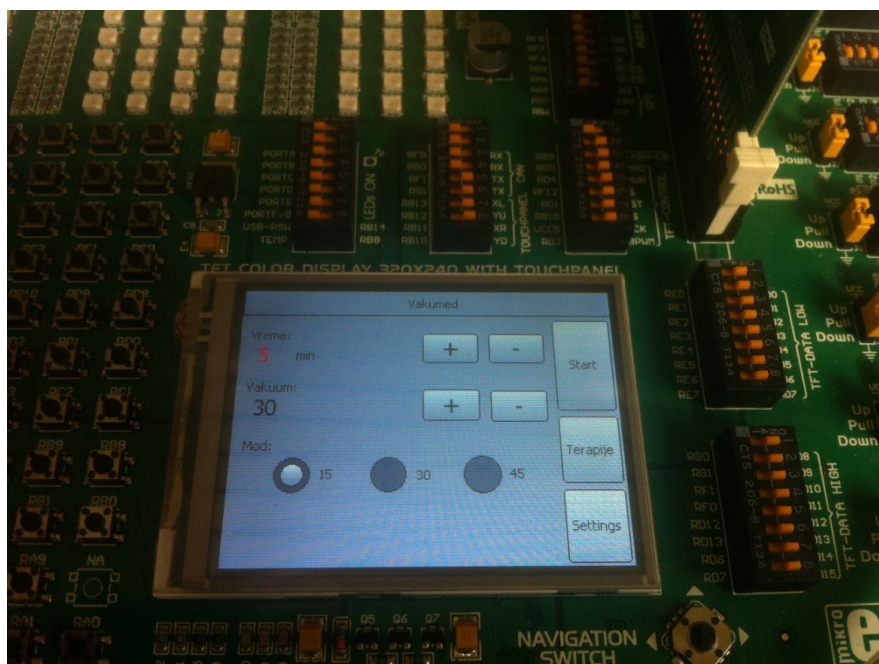
Ekran za prikaz liste je deo koji je zahtevao postojanje dodatne programerske logike. Ovaj ekran predstavlja listu stavki u meniju, koja se programski popunjava odgovarajućim podacima. Tasteri Up i Down sa desne strane treba da omoguće rotiranje liste za slučaj da sadrži više od pet stavki, koliki je maksimum koji može biti prikazan na ekranu u jednom trenutku i zavisi od veličine ekrana.

3.3.3 Izgled korisničkog i upravljačkog interfejsa

S obzirom na to da tehničko rešenje obezbeđuje univerzalnost platforme za implementaciju raličitih uređaja za fizikalnu terapiju firme Elektromedicina d.o.o. iz Niša, kreiran je jedinstveni interfejs koji služi za administraciju uređaja tipa Vakumed, Eksposan, Intermed, Sonoton, Magnemed i Diaton. Administracija podrazumeva, pored osnovnog kretanja kroz menije, i funkcije kao što je na primer zadavanje IP adrese useredaja u cilju povezivanja na centralizovani server za upravljanje i akviziciju. U cilju obezbeđivanja univerzalnosti analizirane su opcije svakog od uređaja, kao i najčešći slučajevi korišćenja kako bi se projektovao optimalan raspored opcija. Nakon faze projektovanja korisničkog interfejsa i dodatne programske logike, sistem je implementiran i testiran na razvojnom okruženju. U nastavku su prikazani najbitniji delovi korisničkog interfejsa.

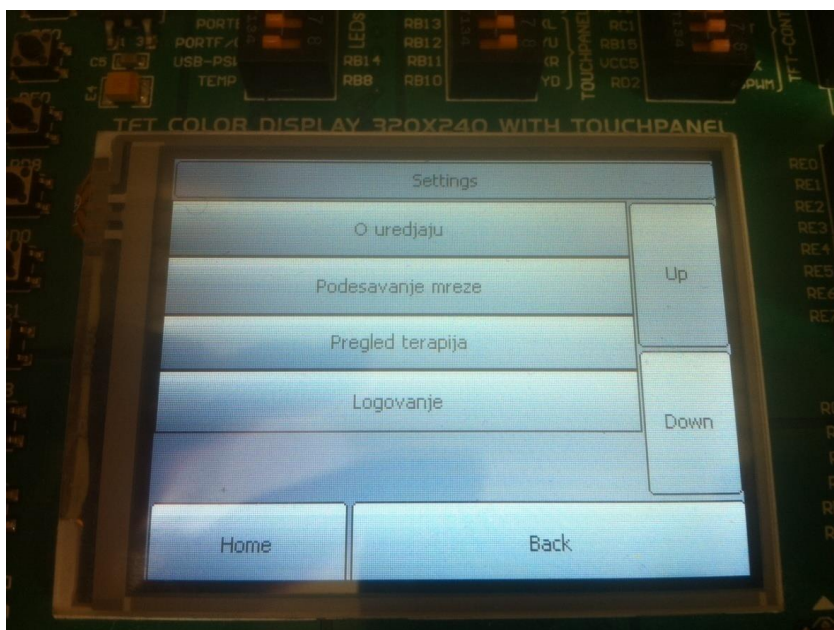
Na slici 3.11 prikazan je izgled startnog ekrana koji se pojavljuje nakon što se uređaj uključi. U centralnom delu ekrana nalaze se kontrole za izmenu radnih parametara čije su vrednosti inicijalno postavljene na podrazumevana podešavanja. U desnom delu postoje tasteri:

- **Start** za startovanje/stopiranje terapije sa radnim parametrima,
- **Terapije** za pregled svih terapija koje postoje definisane u sistemu,
- **Settings** za pristup glavnom meniju.



Slika 3.11 Prikaz ekrana uređaja na početku rada

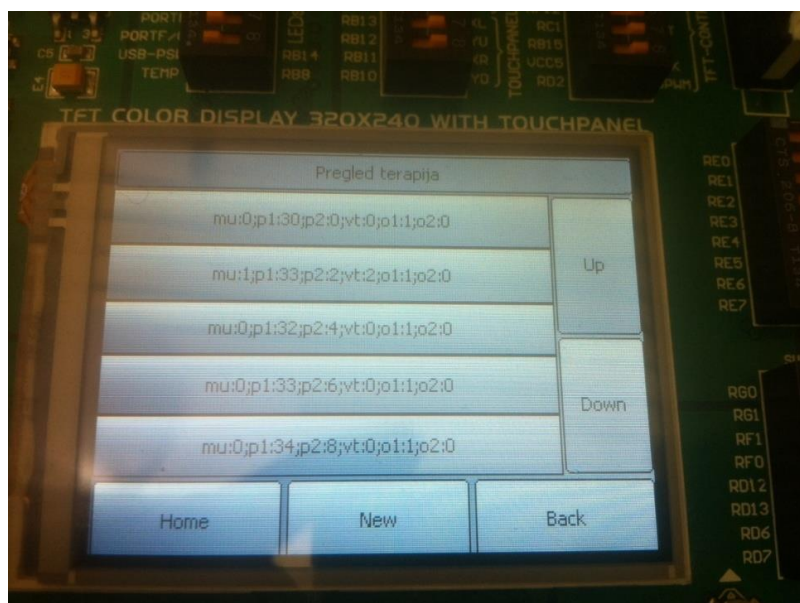
Na slici 3.12 prikazan je izgled menija čija je struktura i funkcionalnost identična za svaki meni. U ovom slučaju u centralnom delu se nalaze stavke glavnog menija od kojih svi osim menija Logovanja vode na druge podmenije. U desnom delu se nalaze tasteri za ciklično skrolovanje menija. Ova funkcija je aktivna samo u slučaju da postoji više od pet stavki u meniju. U donjem delu ekrana postoje tasteri za vraćanje na glavni ekran i iscertavanje prethodnog ekrana.



Slika 3.12 Prikaz dela menija koji može da sadrži veći broj stavki i koje se prikazuju ciklično na ekranu

Klikom na opcije “O uređaju” i “Podešavanje mreže” otvaraju se istovetni ekrani samo sa novim opcijama u meniju. Stavka “Pregled terapija” otvara pregled svih terapija koje postoje snimljene na uređaju (slika 3.13). Kada je na ekranu potrebno iscertati više od pet stavki menija, dodaje se mogućnost cikličnog rotiranja stavki u meniju preko tastera Up i Down. Na ovom ekranu postoji dodatna opcija “New” koja služi za kreiranje nove terapije.

Klikom na ovaj taster kreira se nova terapija u sistemu i prikaže se ekran za podešavanje parametara terapije koji je objašnjen u nastavku.



Slika 3.13 Pregled terapija na Vakumed uređaju

Da bi realizovani sistem menija bio dovoljno uopšten da bi podržao različite formate konfiguracionih parametara urađena je analiza postojećeg skupa uređaja za fizikalnu terapiju. Rezultat analize je skup konfiguracionih parametara i njihovih tipova. U skladu sa tim, u okviru menija predviđene su kontrole za iscertavanje potrebnih komponenti i prihvatanje unetih parametara. U nastavku su prikazani neki od tipova konfiguracionih parametara i izgled odgovarajućeg menija kojim se pristupa podešavanju ili prikazu ovog tipa.

3.3.3.1 Prikaz “read only” podataka

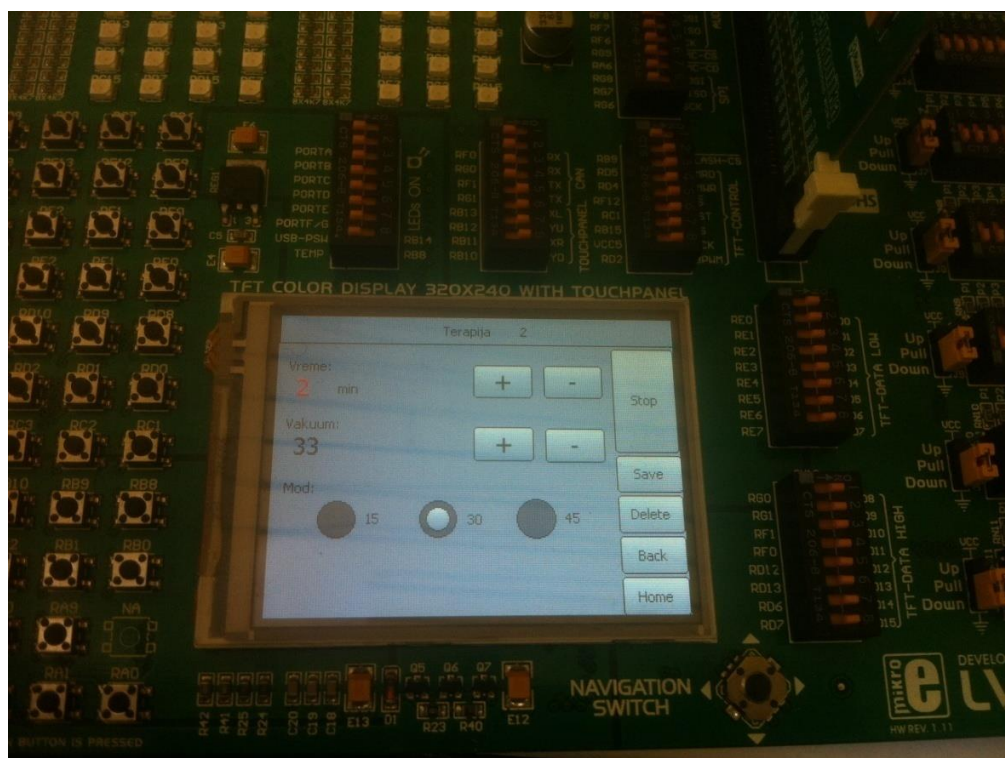
Slika 3.14 prikazuje ekran za podatke koje nije moguće izmeniti. U ovom primeru vidljiv je ekran do koga se dolazi klikom na stavku “Serijski broj” iz podmenija “O Uređaju”. Stavka OK vraća na prethodni ekran.



Slika 3.14 Prikaz serijskog broja uređaja kao primera read-only podatka

3.3.3.2 Ažuriranje terapije

Do ekrana za izmenu terapije (slika 3.15) dolazi se sa podmenija “Pregled terapija”, klikom na određenu terapiju iz liste ili opciju “New” (slika 3.13). Kao i na startnom ekranu, u centralnom delu postoje opcije za podešavanje parametara. U desnom delu, ukoliko uređaj nije aktivan, postoji opcija za startovanje terapije sa parametrima koji su trenutno prikazani na ekranu. Klikom na tu opciju, automatski se prikazuje startni ekran i na njemu se kao radni parametri prikazuju parametri sa stranice za izmenu terapije. U slučaju da je uređaj već aktivan, umesto starta postoji taster za stopiranje. Tasteri “Save” i “Delete” se koriste za snimanje podataka odnosno brisanje terapije sa uređaja.



Slika 3.15 Prikaz ekrana za izmenu terapije na uređaju

3.3.4 Komponente sistema

Osnovu implementacije softvera sistema čine 4 fajla:

- final_main.c
- final_events_code.c
- final_driver.c
- final_objects.h

final_main.c je glavna datoteka. U njoj se nalaze pozivi funkcija za inicijalizaciju ekrana. final_events_code.c datoteka se koristi za pisanje korisničkog programa MicroC PRO za PIC32. U okviru ovog fajla kreirana je logička veza između korisničkog i kontrolnog interfejsa sistema (Slika 3.1).

Pored ovih fajlova koji sadrže logiku za iscertavanje i navigaciju kroz menije, logika koja upravlja konfiguracionim parametrima različitih uređaja nalazi se u okviru fajlova:

- parametric.h
- parametric.c

Ova dva fajla sadrže deklaracije i definicije parametara i funkcija uređaja. Funkcije definisane u okviru ovih fajlova povezuju korisnički interfejs u sistemu sa sistemom koji upravlja konfiguracionim i radnim parametrima uređaja (Slika 3.1). Detalji implementacije ovih funkcija prikazani su u nastavku.

4. Implementacioni detalji

U nastavku će biti prikazani najbitniji delovi u implementaciji sa kratkim pratećim opisom. Struktura za pamćenje parametara terapije prikazana je na slici 4.1. Slika 4.2 prikazuje sadržaj strukture koja čuva opseg dozvoljenih radnih parametara u toku terapija.

```
typedef struct parametri
{
    unsigned char mu;
    //Mod uređaja, vidi tabelu (0-prvi mod, 1-drugi mod, 2-treci mod)

    REZOLUCIJA_PARAMETAR p1;//paramater1 iz tabele
    REZOLUCIJA_PARAMETAR p2; //paramater2 iz tabele
    REZOLUCIJA_VREME vt;//vreme terapije u minutima
    REZOLUCIJA_OPCIJA o1;//dodatna opcija 1 iz tabele
    REZOLUCIJA_OPCIJA o2;//dodatna opcija 2 iz tabele

    #if UREDJAJ==MAGNEMED
        //Samo kod magnemeda, parametri za drugi kanal

        REZOLUCIJA_PARAMETAR p12;
        //paramater1 iz tabele za drugi kanal

        REZOLUCIJA_PARAMETAR p22;
        //paramater2 iz tabele za drugi kanal

        REZOLUCIJA_VREME vt2;
        //vreme terapije u minutima za drugi kanal
    #endif
} TParametri;
```

Slika 4.1 Struktura za pamćenje parametara terapije. Preprocesorskim direktivama dodata je podrška da se u fazi kompajliranja bira tip uređaja.

```

typedef struct opsegvrednosti
{
    struct parametri pocetnevrednosti; //pocetne vrednosti parametara
    struct parametri minimum; //minimalne vrednosti parametara
    struct parametri maksimum; //maksimalne vrednosti parametara
    struct parametri izmena;
    signed char mnozilacp1;
    signed char mnozilacp2;
    #if UREDJAJ==MAGNEMED
        signed char mnozilacp12;
        signed char mnozilacp22;
    #endif
} TOpsegvrednosti;

```

Slika 4.2 Struktura za pamćenje opsega vrednosti parametara terapije

Slika 4.3 prikazuje skup konfiguracionih parametara na uređajima.

```

typedef struct struktura
{
    unsigned char myMacAddr[6]; //MAC adresa uređaja
    unsigned char IpAdresa[4]; //IP adresa uređaja
    unsigned char GwAdresa[4]; //Gateway adresa uređaja
    unsigned char IpMask[4]; //IP maska uređaja
    unsigned char DnsAdresa[4]; //DNS adresa uređaja
    unsigned char IdUredjaja[6]; //ID uređaja
    unsigned char tipUredjaja; //prvo slovo imena uređaja, za Ekspozan
E, itd.
    unsigned char IdMreze[6]; //ID mreže
    unsigned short status; //0-neaktivan, 1 aktivan, 2 greska
    unsigned long int vremeRada; //ukupno vreme rada uređaja u minutima
    unsigned long int zapoceteterapije; //ukupno terapija zapoceto
    unsigned short IdTerapije; //id terapije u bazi uređaja
    struct parametri parametri; //posmatramo je kao trenutnu terapiju
koja se izvrsava
    unsigned short tipGreske; //greska koja se desila na uređaju
    unsigned short vezaSaServerom; //Da li uređaj ima logicku vezu sa
serverom, 0-ne, 1-da
    unsigned short Konfigurisi; //0 DHCP, 1 staticki, 2 zeroconfig
    unsigned char sifra[4];
    unsigned short pocetnoVremeTerapije;
    unsigned short ukupnovt;
    unsigned short bitovi;
    unsigned short predhodnaterapija;
} TStruktura;

```

Slika 4.3 Struktura konfiguracionih parametara uređaja

Za isrcrtavanje nekog ekrana poziva se funkcija DrawScreen, pri čemu joj se prenosi ekran koji treba prikazati (DrawScreen(&ViewSingle)). Da bi se implementirao mehanizam navigacije tj. praćenja istorije navigacije kroz meni i prethodno prikazanih ekrana, uvedena je struktura koja čuva prethodno prikazane ekrane (Slika 4.4).

```
typedef struct History
{
    //pokaziac na prethodni screen
    int current_index;
    //niz pokazivaca na fje koje se pozivaju za isrcrtavanje trenutnog
    screena -> maksimalna dubina je 5.
    void (*function[5]) (void);
} THistory;
```

Slika 4.4 Struktura za pamćenje navigacije kroz menije

Pri svakom isrcrtavanju novog ekrana prethodni se dodaje u ovu strukturu pozivom funkcije addHistory() prikazane na slici 4.5.

```
void addHistory(void (*function) (void))
{
    history.current_index = history.current_index++;
    history.function[history.current_index] = function;
}
```

Slika 4.5 Funkcija za pamćenje prethodnog ekrana u toku navigacije kroz sistem menija

Prilikom klika na taster Back sa grafičkog interfejsa, poziva se funkcija čiji je listing prikazan na slici 4.6 i čiji je zadatak uklanjanje poslednjeg ekrana iz THistory strukture.

```
void backOneHistory()
{
    //poziv fje za isrcrtavanje prethodnog ekrana
    // -2 ide jer svaka fja koja se poziva u sebi poziva addHisotry, sto
    automatski inkrementira index.
    //Zato ide -2, da bi ga pri pozivu povecao na current_index-1 koliko
    bi trebalo da bude
    history.current_index = history.current_index - 2;
    history.function[history.current_index+1] ();
}
```

Slika 4.6 Funkcija koja realizuje kretanje za jedan korak unazad kroz sistem menija

Pritiskom na taster Home sa grafičkog interfejsa, poziva se funkcija koja vrši brisanje kompletne strukture i prikazana je na slici 4.7.


```

void removeHistory()
{
    history.current_index = 0;

    //postavljamo parametre na home Screen-u da budu kao pocetni
parametri

    ShortToStr(strukt.parametri.vt, CTvt);

    HomeVreme.Caption = TrimLeft(CTvt);

    IntToStr(strukt.parametri.p1, CTp1);

    HomeVakuum.Caption = TrimLeft(CTp1);

    //...ukolonjen kod

    DrawScreen(&HomeScreen);
}

```

Slika 4.7 Funkcija koja realizuje povratak na vrh sistema menija i realizuje se pritiskom na taster Home

Navigacija je implementirana korišćenjem tastera preko strukture koja čuva podatke za dinamičko iscrtavanje menija i listi a prikazana je na slici 4.8.

```

typedef struct ButtonData
{
    //naziv

    char* caption;

    char* paramName;

    //pokazivac na fju koja se poziva klikom

    void (*function) (void);

} TButtonData;

```

Slika 4.8 Struktura koja čuva pokazivač na funkciju za iscrtavanje

Obzirom da se za očitavanje i izmenu parametara uređaja koriste iste funkcije za čitanje i izmenu svih parametara uređaja, pre otvaranja ekrana, promenljiva paramName dobija vrednost koja definiše koji parametar treba prikazati ili izmeniti pri pozivu funkcija getParam() ili setParam(). Parametar function je pokazivač na funkciju kojom se iscrtava odgovarajući ekran. Niz podataka TButtonData structure se koristi za implementiranje meni liste, definisane na ListMenu ekranu. Svaki element liste, definisan je svojim nazivom *caption-om* koji će biti tekst na element liste i funkcijom koja će biti pozvana pri kliku na taj taster.

Prilikom startovanja aplikacije, u okviru glavnog programa poziva se funkcija za inicijalizaciju svih parametara, gde se popunjavaju odgovarajuće strukture vrednostima. Slika 4.9 prikazuje funkciju za inicijalizaciju.

```

void Initialization()
{
    zeroconf();

    //postavljanje historija da pocetni Screen bude uvek home Screen.
    history.current_index = 0;
    history.function[0] = HomeButton;
    strukt.status = 0;

    //funkcije koje ovde upisujemo se ne pozivaju direktno klikom na dugme

    //vec njih pozivaju f-je List_1_button,... i pozivaju ih sa
    odgovarajucim indexom
    {
        menu_list_data[0].caption = "O uredjaju";
        menu_list_data[0].function = OUredjajuButton;
        menu_list_data[1].caption = "Podesavanje mreze";
        menu_list_data[1].function = PodesavanjeMrezeButton;
        //izostavljen kod za inicijalizaciju osnovnih stavki menija
        // O Uredjaju
        menu_o_uredjaju[0].caption = "Serijski Broj";
        menu_o_uredjaju[0].paramName = "serijskibroj";
        menu_o_uredjaju[0].function = DrawViewSingle;
        //izostavljen kod za inicijalizaciju stavke menija "O uredjaju"
        // Podesavanja mreze
        menu_podesavanje_mreze[0].caption = "IP Uredjaja";
        menu_podesavanje_mreze[0].paramName = "IpAdresa";
        menu_podesavanje_mreze[0].function = DrawEditSingle;
        //izostavljen kod za inicijalizaciju stavke menija "Podesavanje
        mreze"
    }
    popuniTestParametre();
}

```

Slika 4.9 Funkcija za globalnu inicijalizaciju sistema menija

Na kraju ove funkcije, poziva se funkcija kojom se popunjavaju parametar terapija. U realnom okruženju, ovom funkcijom će se vrednosti parametara terapija postaviti na definisane vrednosti (Slika 4.10).

```

void popuniTestParametre ()
{
    int ii;

    for (ii=0; ii < test_terapija_count; ii++)
    {
        test_terapije[ii].mu = opseg.pocetnevrednosti.mu;
        test_terapije[ii].p1 = opseg.pocetnevrednosti.p1 + ii;
        test_terapije[ii].p2 = opseg.pocetnevrednosti.p2 + (2*ii);
        test_terapije[ii].vt = opseg.pocetnevrednosti.vt;
        test_terapije[ii].o1 = opseg.pocetnevrednosti.o1;
        test_terapije[ii].o2 = opseg.pocetnevrednosti.o2;

        strcpy(buff[ii], "");
        strcpy(buff[ii], UcitajPromenljiveuString(ii));

        test_terapije_data[ii].caption = buff[ii];
        test_terapije_data[ii].function = DrawEditTherapy;
    }

    //pozivamo kako bi popunili parametre na home Screen-u
    //postavljamo parametre na home Screen-u da budu kao pocetni
    parametri

    strukt.parametri.vt = opseg.pocetnevrednosti.vt;
    ShortToStr(strukt.parametri.vt, CTvt);

    // izostavljen kod

    DrawScreen (&HomeScreen);
}

```

Slika 4.10 Funkcija karakteristična za svaki uređaj, vrši inicijalizaciju svih struktura koje sadrže parametre. Moguće je podatke direktno upisati ili ih učitati iz memorijskog sistema

Klikom na tastere Settings ili Terapije iz grafičkog interfejsa, MenuList ekran se popunjava podacima iz odgovarajućeg niza. Primer funkcije koja se poziva klikom na Settings taster prikazana je na slici 4.11.

```

void SettingsButton()
{
    curr_list = &menu_list_data;
    list_length = menu_list_count;
    list_index = 0;
    addHistory(SettingsButton);
    listCaption.Caption = "Settings";
    ListDisplay();
}

```

Slika 4.11 Funkcija za iscrtavanje menija za podešavanje uređaja

Opis elemenata koji se pojavljuju u telu funkcije prikazan je u nastavku:

`curr_list` je promenljiva koja definiše koja je trenutna lista koja treba da se prikaze u meniju.

`list_length` - definiše broj elemenata liste.

`list_indeks` - postavlja od kog indeksa se kreće sa iscrtavanjem – vrednost je 0 kada se dolazi preko tastera a taj parametar se menja klikom na taster UP/DOWN.

`addHistory(SettingsButton)` - postavlja se u istoriju koja je trenutna stranica.

`listCaption.Caption = "Settings"` - postavlja se naziv menija i poziva se iscrtavanje menija.

`ListDisplay()` - Funkcija koja vrši iscrtavanje liste (prikazuje meni). Radi nezavisno od menija koji treba da se prikaže. Odvojena je funkcionalnost od podataka tako sto se uvek pre poziva prvo postave promenljive koja je trenutna lista sa kojom se radi, njena dužina i početni indeks prve stavke u meniju.

Na slici 4.12 prikazan je listing funkcije koja u zavisnosti od toga da li se radi o iscrtavanju menija sa terapijama iscrtava taster za novu terapiju ili, u suprotnom, iscrtava Back taster koji se nalazi u svim ostalim slučajevima.

```

void ListDisplay() {
    updateListButtons(list_index);
    if (curr_list == &test_terapije_data)
    {
        TherapyBackOriginal.Active = 0;
        TherapyNew.Active = 1;
        TherapyBack.Active = 1;
        TherapyBackOriginal.Visible = 0;
        TherapyNew.Visible = 1;
        TherapyBack.Visible = 1;
    }
    else
    {
        TherapyBackOriginal.Active = 1;
        TherapyNew.Active = 0;
        TherapyBack.Active = 0;
        TherapyBackOriginal.Visible = 1;
        TherapyNew.Visible = 0;
        TherapyBack.Visible = 0;
    }
    DrawScreen(&ListMenu);
}

```

Slika 4.12 Listing funkcije koja u zavisnosti od toga da li se radi o iscrtavanju menija sa terapijama iscertava taster za novu terapiju ili Back taster

Funkcija `updateListButtons` (`int startIndeks`) prikazana na slici 4.13 i u okviru nje se pre iscertavanja liste vrši inicijalizacija svojstava `Caption` tastera i vrši iscertavanje pet vidljivih stavki menija. Takođe, u okviru ove funkcije reguliše se slučaj kada ima manje od pet elemenata u listi.

```
void updateListButtons(int startIndeks)
{
    int counter;
    for ( counter = 0 ; counter < 5 && counter < list_length; counter++ ) {
        ListMenu.Buttons[counter]->Caption = curr_list[(counter +
startIndeks + list_length) % list_length].caption;
        ListMenu.Buttons[counter]->Visible = 1;
    }
    for(;counter < 5; counter++)
    {
        ListMenu.Buttons[counter]->Visible = 0;
    }
}
```

Slika 4.13 Pomoćna funkcija za inicijalizaciju prikaza listi

Kako bi se omogućilo ciklično kretanje po meniju (kada lista ima više od pet stavki) implementirane su funkcije `List_UpOnClick()` i `List_DownOnClick()`, tako što je u okviru njih postavljen `list_indeks` – promenljiva koja čuva indeks prve stavke koju treba iscertati. Listing ovih funkcija prikazan je na slici 4.14.

```
void List_UpOnClick() {
    if (list_length > 5)
    {
        list_index = (list_index - 1 + list_length) % list_length;
        ListDisplay();
    }
}
void List_DownOnClick() {
    if (list_length > 5)
    {
        list_index = (list_index + 1) % list_length;
        ListDisplay();
    }
}
```

Slika 4.14 Listing funkcija koje realizuju navigaciju kroz sistem menija

Za iscrtavanje odgovarajućih ekrana, definisane su sledeće funkcije, koje služe kao wrapper za podešavanje parametara pre iscrtavanja odgovarajućih ekrana:

```
void DrawEditTherapy() {...}
void DrawEditSingle() {...}
void DrawViewSingle() {...}
void DrawRadio() {...}
```

Funkcije void List_1_Button() - void List_5_Button(), prikazane na slici 4.15, pozivaju se klikom na odgovarajuću stavku menija (liste). Redni broj odgovara redosledu u meniju na kome se nalazi. U ovim funkcijama računa se radni indeks koji predstavlja indeks stavke koja se trenutno menja ili prikazuje. Na osnovu tog indeksa se poziva odgovarajuća callback funkcija. Ona dalje može da poziva iscrtavanje novog menija, editora ili prikaza podataka. Koju će funkciju dalje da poziva definisano je u inicijalnoj strukturi.

```
void List_1_Button() {
    radni_index = (list_index + 0) % test_terapija_count;
    curr_list[radni_index].function();
}
void List_2_Button(){
    radni_index = (list_index + 1) % test_terapija_count;
    curr_list[radni_index].function();
}
void List_3_Button() {...}
void List_4_Button() {...}
void List_5_Button() {...}
```

Slika

4.15 Callback funkcije kojima se realizuje poziv funkcije za obradu pritiska na tastere

Za pribavljanje parametara koristi se funkcija char* getPar() koja pribavlja vrednost parametra kao string na osnovu stringa sa nazivom parametra i smešta ga u globalnu promenljivu pom (Slika 4.16).

```

char* getPar ()
{
    pom[0]='\0';
    if (strcmp(main_param, "GwAdresa")==0)
    {
        pom[0]=strukt.GwAdresa[0];
        pom[1]=strukt.GwAdresa[1];
        pom[2]=strukt.GwAdresa[2];
        pom[3]=strukt.GwAdresa[3];
        pom[4]='\0';
        return pom;
    }
    if (strcmp(main_param, "status")==0)
    {
        ShortToStr(strukt.status, pom);
        TrimLeft(pom);
        return pom;
    }
    //izostavljen kod
}

```

Slika 4.16 Pomoćna funkcija za pribavljanje vrednosti traženog parametra

Funkcija `char* setPar()` se na sličan način (samo u obrnutom smeru) koristi za skladištenje parametara (Slika 4.17).

```

short setPar( char *vrednost)
{
    //izostavljen kod
    if (strcmp(main_param, "GwAdresa")==0)
    {
        strukt.GwAdresa[0]=vrednost[0];
        strukt.GwAdresa[1]=vrednost[1];
        strukt.GwAdresa[2]=vrednost[2];
        strukt.GwAdresa[3]=vrednost[3];
        if (strukt.GwAdresa[3]==0)
            strukt.GwAdresa[3]=1;
        return 1;
    }
    //izostavljen kod
}

```

Slika 4.17 Pomoćna funkcija za promenu vrednosti konfiguracionih parametara

Na Edit Single ekranu koji se između ostalog koristi za podešavanje IP adrese, za svaki + i - taster je vezana odgovarajuća funkcija (Slika 4.18) koja povećava ili smanjuje vrednost datog parametra za jedan. Ove vrednosti koje se menjaju su tipa bajt tako da uzimaju vrednosti od 0-255.

```
void EditSingle1PlusOnClick() {
    ES1value++;
    ByteToStr(ES1value, EditSingle1.Caption);
    DrawScreen(&EditSingle);
}

void EditSingle1MinusOnClick() {
    ES1value--;
    ByteToStr(ES1value, EditSingle1.Caption);
    DrawScreen(&EditSingle);;
}

void EditSingle2PlusOnClick() {...}
void EditSingle2MinusOnClick() {...}
void EditSingle3PlusOnClick() {...}
void EditSingle3MinusOnClick() {...}
void EditSingle4PlusOnClick() {...}
void EditSingle4MinusOnClick() {...}
```

Slika 4.18 Funkcije za povećanje i smanjenje parametra na interfejsu korišćenog za podešavanje IP adrese uređaja

Funkcija `void OK()` se poziva sa svakog ekrana koji radi izmenu podataka kada se klikne taster OK radi snimanja podataka. Listing ove funkcije prikazan je na slici 4.19.

```

void OK() {
    if(history.function[history.current_index] == DrawEditSingle)
    { // podesavanje parametarka u formatu IP adrese ili subnet maske
        pom[0] = ES2value;
        pom[1] = ES2value;
        pom[2] = ES3value;
        pom[3] = ES4value;
        setPar(pom);
    }
    else if(history.function[history.current_index] == DrawEditTherapy)
    {
        //update terapije
        test_terapije[radni_index].vt = Tvt;
        test_terapije[radni_index].pl = Tpl;
        test_terapije[radni_index].mu = Tmu;

        //radimo refresh sumarnog prikaza terapije koji se koristi na
list terapija

        strcpy(buff[radni_index],"");
strcpy(buff[radni_index],UcitajPromenljiveuString(radni_index));

        test_terapije_data[radni_index].Caption = buff[radni_index];
        test_terapije_data[radni_index].function = DrawEditTherapy;
        backOne();
    }
}

```

Slika 4.19 Listing funkcije koja se poziva za potvrdu načinjenih izmena ili zatvaranje prikazanih delova menija

Funkcija void StartTherapy() (slika 4.20) poziva se klikom na taster za startovanje terapije. Ukoliko je poziv obavljen sa glavnog ekrana koriste se podaci koji su na njemu podešeni, a ukoliko je poziv obavljen sa ekrana za neku terapiju onda se podešeni parametri sa tog ekrana postavljaju kao radni parametri a zatim se prikazuje glavni ekran.

```

void StartTherapy()
{
    if (strukt.status == 0)
        strukt.status = 1;
        HomeStart.Caption = "Stop";
        TherapyStart.Caption = "Stop";
        if (history.current_index == 0)
            { DrawScreen(&HomeScreen);
            }
        else
            { strukt.parametri.vt = Tvt;
            strukt.parametri.pl = Tpl;
            strukt.parametri.mu = Tmu;
            removeHistory();
            }
    }
    else if (strukt.status == 1)
        { //stopiranje terapije
        strukt.status = 0;
        HomeStart.Caption = "Start";
        TherapyStart.Caption = "Start";
        if (history.current_index == 0)
            { DrawScreen(&HomeScreen);
            }else
            { DrawScreen(&EditTherapy); }
        }
}

```

Slika 4.20 Funkcija za pokretanje terapije na uređaju

Sledeće funkcije se pozivaju kada se vrši izmena parametara na glavnom ekranu ili na ekranu za izmenu terapije i to su:

```

void TherapyVremePlusOnClick() {...}
void TherapyVremeMinusOnClick() {...}
void TherapyVakuumPlusOnClick() {...}
void TherapyVakuumMinusOnClick() {...}
void TherapyMod1OnClick() {...}
void TherapyMod2OnClick() {...}
void TherapyMod3OnClick() {...}

```

Dodavanje nove terapije poziva se sa ekrana za pregled terapija kada se dodaje nova terapija. Terapija se dodaje na kraj liste i postavljaju joj se inicijalni parametri. Nakon toga otvara se editor za tu terapiju. Funkcija za dodavanje nove terapije prikazana je na slici 4.21.

```

void TherapyNewOnClick() {
    //max moze 20 terapija
    if (test_terapija_count < max_terapija_count)
    {
        test_terapije[test_terapija_count].mu = opseg.pocetnevrednosti.mu;
        test_terapije[test_terapija_count].p1 = opseg.pocetnevrednosti.p1;
        test_terapije[test_terapija_count].p2 = opseg.pocetnevrednosti.p2;
        test_terapije[test_terapija_count].vt = opseg.pocetnevrednosti.vt;
        test_terapije[test_terapija_count].o1 = opseg.pocetnevrednosti.o1;
        test_terapije[test_terapija_count].o2 = opseg.pocetnevrednosti.o2;

        strcpy(buff[test_terapija_count], "");
        strcpy(buff[test_terapija_count], UcitajPromenljiveuString(test_terapija_count));

        test_terapije_data[test_terapija_count].Caption =
        buff[test_terapija_count];

        test_terapije_data[test_terapija_count].function = DrawEditTherapy;
    }

    radni_index = test_terapija_count;
    test_terapija_count++;
    curr_list[radni_index].function();
}

```

Slika 4.21 Funkcija za dodavanje nove terapije

5. Zaključak

U ovom tehničkom rešenju data je implementacija uređaja na mikrokontroleru novije generacije, uz uvođenje modernijeg korisničkog interfejsa sa ekranom osjetljivim na dodir. Razvijen je skup biblioteka koje realizuju iscrtavanje korisničkog interfejsa uređaja za fizikalnu terapiju sa podrškom za ekrane osjetljive na dodir. S obzirom na to da različiti uređaji imaju različite parametre terapije, projektovanim korisničkim interfejsom obezbeđena je univerzalnost i mogućnost korišćenja istog hardvera za implementaciju različitih uređaja. Izbor tipa uređaja koji implementira jedinstveni hardver, uz korišćenje zajedničkih grafičkih elemenata, omogućen je izmenama u konfiguracionim fajlovima. Razvijen je prototip uređaja sa kontrolerom PIC32MX460F512L i ekranom osjetljivim na dodir rezolucije 320x240 sa mogućnošću prikaza 262 hiljade različitih boja.

Prototip uređaja baziran je na kontroleru PIC32MX460F512L sa ekranom osjetljivim na dodir, rezolucije 320x240 i sa mogućnošću prikaza 262 hiljade različitih boja. U sklopu ovog rešenja kreirane su funkcije za iscrtavanje korisničkog interfejsa na grafičkom ekranu. Identifikovani su konfiguracioni parametri različitih uređaja za fizikalnu terapiju i omogućen je njihov prikaz, unos i izmena. Konfiguracioni i radni parametri uređaja mogu biti znakovni podaci, brojčani podaci kao i parametri kod kojih se bira jedna iz skupa predefinisanih vrednosti. Realizovanim skupom funkcija omogućeni su navigacija kroz hijerarhijski opisan sistem menija i podešavanje uređaja preko ekrana osjetljivog na dodir.

Implementacijom hardverskog dela platforme i pratećeg softverskog dela u vidu datih funkcija, ovim tehničkim rešenjem uveden je novi kvalitet u uređaje za fizikalnu medicinu firme Elektromedicina d.o.o. S jedne strane razvijena hardverska platforma pruža jedinstvenu baznu platformu za implementaciju svih uređaja iz proizvodne palete Elektromedicine d.o.o. Sa druge strane softverski deo platforme pruža jedinstveni korisnički interfejs, a razlike koje postoje među uređajima u vidu tipova, naziva i vrednosti parametara kompenzovane su aktivacijom različitih konfiguracionih fajlova.

Problem koji je na posredni način rešen uvođenjem univerzalne platforme je problem navikavanja korisnika na različite interfejse prilikom korišćenja različitih tipova uređaja istog proizvođača. Na ovaj način, operater uređaja prilikom savladavanja načina korišćenja jednog uređaja ujedno ovladava tehnikom upotrebe svih uređaja iz proizvodnog programa proizvođača.

Reference

1. A.Garcia, M. Huizar, B.Baumgartner, U.Schreiber, A.Knoll, "Embedded Platform for Automation of Medical Devices", *Proceedings on Computing in Cardiology, Hangzhou, China*, 18-21 September 2011, pp. 829-832.
2. D. Ibrahim, "Using Touch Screen LCDs in Embedded Applications", *Electron. World*, September 2010, pp. 28-34.
3. Bhalla, Mudit Ratana, and Anand Vardhan Bhalla. "Comparative Study of Various Touchekran Technologies", *International Journal of Computer Applications*, No. 8, 2010, pp. 12-18.
4. <http://www.touchscreen-me.com/about-touch-screens.php>, dostupno decembra 2013.
5. <http://www.mikroe.com/>, dostupno decembra 2013.
6. <http://www.mikroe.com/products/view/462/lv-32mx-v6-development-system/>, dostupno decembra 2013.
7. <http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en534177>, dostupno decembra 2013.