



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У НОВОМ САДУ



Немања Ђекић

USB-ом контролисан релејни прекидач

ДИПЛОМСКИ РАД
- Основне академске студије -

Нови Сад, 2016



КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, РБР:			
Идентификациони број, ИБР:			
Тип документације, ТД:	Монографска документација		
Тип записа, ТЗ:	Текстуални штампани материјал		
Врста рада, ВР:	Завршни (Bachelor) рад		
Аутор, АУ:	Немања Ђекић		
Ментор, МН:	Проф. др. Стеван Станковски		
Наслов рада, НР:	USB-ом контролисан релејни прекидач		
Језик публикације, ЈП:	Српски / латиница		
Језик извода, ЈИ:	Српски		
Земља публиковања, ЗП:	Република Србија		
Уже географско подручје, УГП:	Војводина		
Година, ГО:	2016		
Издавач, ИЗ:	Ауторски репринт		
Место и адреса, МА:	Нови Сад; трг Доситеја Обрадовића 6		
Физички опис рада, ФО: (поглавља/страница/цитата/табела/слика/графика/прилога)	7/24/0/4/18/0/0		
Научна област, НО:	Мехатроника		
Научна дисциплина, НД:	Мехатроника		
Предметна одредница/Кључне речи, ПО:	Програмски језик C, програмски језик C#, USB комуникација		
УДК			
Чува се, ЧУ:	У библиотеци Факултета техничких наука, Нови Сад		
Важна напомена, ВН:	Нема		
Извод, ИЗ:	USB-ом контролисан релејни прекидач. Било је потребно реализовати физичку архитектуру засновану на развојном окружењу NXP компаније. Реализација програмске подршке за физичку архитектуру у развојном окружењу LPCXpresso. Реализација графичке корисничке спрете у развојном окружењу Visual Studio 2010.		
Датум прихватања теме, ДП:			
Датум одбране, ДО:			
Чланови комисије, КО:	Председник:	др Гордана Остојић, ванред. проф.	
	Члан:	др Небојша Ђевалица, доцент	Потпис ментора
	Члан, ментор:	др Стеван Станковски, ред. проф.	



KEY WORDS DOCUMENTATION

Accession number, ANO:			
Identification number, INO:			
Document type, DT:	Monographic publication		
Type of record, TR:	Textual printed material		
Contents code, CC:	Bachelor Thesis		
Author, AU:	Nemanja Đekić		
Mentor, MN:	Stevan Stankovski, Prof. PhD		
Title, TI:	USB controlled relay switch		
Language of text, LT:	Serbian		
Language of abstract, LA:	Serbian		
Country of publication, CP:	Republic of Serbia		
Locality of publication, LP:	Vojvodina		
Publication year, PY:	2016		
Publisher, PB:	Author's reprint		
Publication place, PP:	Novi Sad, Dositeja Obradovica sq. 6		
Physical description, PD: (chapters/pages/ref./tables/pictures/graphs/appendices)	7/24/0/4/18/0/0		
Scientific field, SF:	Mechatronics		
Scientific discipline, SD:	Mechatronics		
Subject/Key words, S/KW:	Programming language C, programming language C#, USB communication		
UC			
Holding data, HD:	The Library of Faculty of Technical Sciences, Novi Sad, Serbia		
Note, N:	None		
Abstract, AB:	USB controlled relay switch The hardver was based on NXP development tool. The software was realased in LPCExpresso. GUI was realased in Visual Studio 2010.		
Accepted by the Scientific Board on, ASB:			
Defended on, DE:			
Defended Board, DB:	President:	Ph. D. Gordana Ostojić, professor	
	Member:	Ph. D. Nebojša Pjevalica, docent	Menthor's sign
	Member, Mentor:	Ph. D. Stevan Stankovski, full professor	

Zahvalnost

Zahvaljujem se RT-RK institutu na pruženoj mogućnosti za realizaciju ovog rada, kao i kolegama iz *Cirrus Logic* tima na stručnoj pomoći i savetima prilikom izrade ovog rada.

Posebnu zahvalnost dugujem svojim roditeljima i bratu zbog pružene moralne, emotivne i finansijske podrške tokom školovanja.

Takođe bih se zahvalio svima onima koji su pratili moje ideje, vizije i na bilo koji način pružali podršku.



SADRŽAJ

1.	Uvod	1
2.	Teorijske osnove.....	2
2.1	ARM arhitektura	2
2.2	USB komunikacija	3
2.2.1	Fizička arhitektura USB-a	3
2.3	Relej	4
2.3.1	Princip rada releja.....	4
2.3.2	Radni napon i struja.....	5
2.3.3	Prednosti i mane releja	6
3.	Koncept rešenja	7
4.	Programsko rešenje.....	11
4.1	Programska podrška	12
4.1.1	USB komunikacija.....	12
4.1.2	Upravljanje relejima	14
4.2	Grafička korisnička sprega (GUI).....	14
5.	Rezultati.....	17
5.1	Ispitivanje i verifikacija uređaja.....	17
5.2	Ispitivanje i verifikacija grafičke korisničke sprege	19
6.	Zaključak	22
7.	Literatura	23

SPISAK SLIKA

Slika 2.1 ARM Cortex M3 arhitektura	2
Slika 2.2 Šema mini USB-a tipa B	3
Slika 2.3 Delovi releja	4
Slika 2.4 a) Isključen relaj i b) Uključen relaj	4
Slika 2.5 Kontakti releja	5
Slika 3.1 LPC1769Xpresso razvojno okruženje	7
Slika 3.2 Relejna ploča	8
Slika 3.3 USB napajanje i naponski razdelnik.....	8
Slika 3.4 Raspored konektora na LPC1769Xpresso razvojnom okruženju.....	9
Slika 3.5 <i>Arduino</i> relejna ploča.....	9
Slika 3.6 <i>PULL UP</i> otpornik.....	10
Slika 4.1 Komunikacija PC-a sa relejom	11
Slika 4.2 Izgled grafičke korisničke sprege (Prvi i drugi dijalog prozor).....	15
Slika 5.1 Prikaz <i>PutTY</i> aplikacije.....	18
Slika 5.2 Ispitivanje uređaja uz pomoć <i>PutTY</i> aplikacije.....	18
Slika 5.3 Scenario 1	19
Slika 5.4 Scenario 2	20
Slika 5.5 Scenario 3	21

SPISAK TABELA

Tabela 2.1 Brzina USB ^[4] komunikacije	3
Tabela 4.1 Funkcije USB biblioteke.....	13
Tabela 4.2 Funkcije za upravljanje relejima.....	14
Tabela 4.3 Funkcije grafičke korisničke sprege	16

SKRAĆENICE

USB – *Universal Serial Bus*, Jedinstvena serijska magistrala

PC – *Personal Computer*, Računar za ličnu upotrebu

RISC – *Reduced Instruction Set Computing*, Procesor sa smanjenim skupom instrukcija

ARM – *Advanced RISC Machine*, Napredne mašine sa procesorima sa smanjenim skupom instrukcija

SoC – *System on a Chip*, Integisano kolo koje sadrži sve elektronske komponente nekog elektronskog sistema na jednom čipu

GSM – *Global System for Mobile Communication*, Globalni sistem za mobilnu komunikaciju

STP – *Shielded Twisted Pair*, Par izolovanih bakarnih žica

UTP – *Unshielded Twisted Pair*, Par neizolovanih bakarnih žica (za podatke kod USB)

DC – *Direct Current*, Jednosmerna sturja

AC – *Alternating Current*, Naizmenična struja

JTAG – *Joint Test Action Group*, Standard za ispitivanje i otklanjanje grešaka namenskih sistema

UART – *Universal Asynchronous Receiver/Transmitter*, Univerzalni asihroni prijemnik/predajnik

SPI - *Serial Peripheral Interface*, Serijski sprežni sistem za periferije

I2C – *Inter Integrated Circuit*, Serijska komunikaciona sprega

SSP – *Synchronous Serial Port*, Sihroni serijski port

CAN – *Controller Area Network*, Standardna magistrala kontrolera

PWM – *Pulse Width Modulation*,

RTC – *Real Time Clock*, Tajmer u realnom vremenu

I2S – *Integrated Interchip Sound*, Serijska komunikaciona sprega

GPIO – *General Purpose Input/Output*, Konektori (izvodi) opšte namene

CPU – *Central Processing Unit*, Centralna procesorska jedinica

COM Port – *Communication Port*, Otvor za komunikaciju

ASIO – *Audio Stream Input/Output*,

DSP – *Digital Signal Processing*, Digitalna obrada signala ili

Digital Signal Processor, Digitalni signalni procesor

1. Uvod

Procesi proizvodnje se često automatizuju, jer se javlja potreba za bržom i kvalitetnijom proizvodnjom. Ovo je jako značajno i za razvoj multimedijalnih uređaja.

Razvoj multimedijalnih uređaja se odvija u nekoliko faza, od kojih je najznačajnija faza ispitivanja (*engl. Testing*). U fazi ispitivanja uređaja se utroši najviše vremena, pa ne čudi da se i u ovu fazu uključuju sve više i više uređaji za automatizaciju ispitivanja. Neretko se dešava da se ispitivanja koja se obavljaju zaustave pre nego što se izvrše svi zadani slučajevi, zbog greške u sistemskoj podršci, pa je potrebno fizički ponovo pokrenuti (*engl. Restart*) ili isključiti uređaj koji se testira i ponoviti ispitivanja.

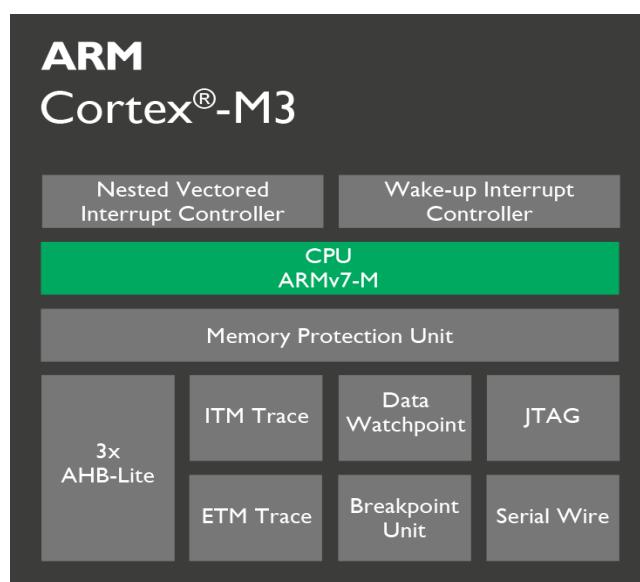
Ovaj rad nudi jedno od rešenja za ovaj problem, pomoću kojeg je moguće uključivanje i isključivanje eksternih uređaja, u ovom slučaju uređaja koji se ispituju, a da korisnik ne mora fizički biti prisutan na mestu gde se ispitivanja odvijaju. U ovom radu realizovano je rešenje koje omogućava uključivanje i isključivanje eksternih uređaja pomoću PC-a putem USB komunikacije. Cilj zadatka je realizacija uređaja koji kontroliše uređaje koji rade na viskom naponu/struji pomoću uređaja male snage i niskog napona, kao i aplikacije za dati uređaj. Za realizaciju ovog uređaja korišten je NXP LPC1769Xpresso mikrokontroler, koji je baziran na ARM tehnologiji.

2. Teorijske osnove

2.1 ARM arhitektura

ARM^{[1][2]} je porodica arhitektura skupa instrukcija za procesore bazirane na *RISC* arhitekturi razvijena od strane britanske kompanije ARM Holdings.

RISC bazirani pristup računarskog dizajna znači da ARM procesori zahtevaju znatno manje tranzistora od tipičnih procesora kod prosečnih računara. Ovaj pristup smanjuje cenu, zagrevanje i potrošnju energije. Ovo su željene osobine kod lakih, prenosnih, uređaja koji rade na bateriju uključujući pametne telefone, prenosne računare, tablete i druge ugrađene računarske sisteme. Jednostavniji dizajn omogućava efikasnije višejezgrane procesore i veći broj jezgara po manjoj ceni, što daje veću računarsku moć i višu energetsku efikasnost za servere i super računare.



Slika 2.1 ARM Cortex M3 arhitektura

2.2 USB komunikacija

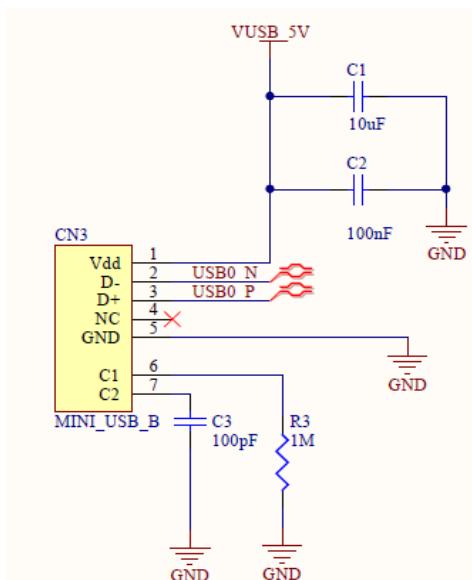
USB(engl. *Universal Serial Bus*^[3] – univerzalna serijska magistrala ili opšta posledovna sabirnica) je spoljašnji priključak za razne periferne uređaje (štampač, miš, tastatura, digitalna kamera, modem, itd.) . Karakteriše ga velika brzina i jednostavnost priključenja (engl. *Plug & Play* princip). Moguće je uključiti i do 127 uređaja na USB magistralu. ž

Vrsta	Brzina
USB 1.1 (Low Speed)	1.5Mbit/s
USB 1.1 (High Speed)	12 Mbit/s
USB 2.0 (High Speed)	480 Mbit/s
USB 3.0 (High Speed)	5 Gbit/s

Tabela 2.1 Brzina USB^[4] komunikacije

2.2.1 Fizička arhitektura USB-a

Standardi^[5] USB1.x i 2.0 koriste kabal sa dva para bakarnih žica, od kojih se jedan par koristi za napajanje, a drugi za podatke. Par žica koji je zadužen za podatke smešten je tako da bi se smanjio uticaj interferencije i šuma. Dužina kabla je ograničena na 5 m zbog maksimalnog kašnjenja odgovora koje iznosi 1500ns što uz maksimalnu brzinu prenosa preko bakarne žice ograničava dužinu. Ukoliko se ne dobije odgovor primaoca unutar 1500 ns, takav podatak se smatra izgubljenim i zahteva ponovno slanje.



Slika 2.2 Šema mini USB-a tipa B

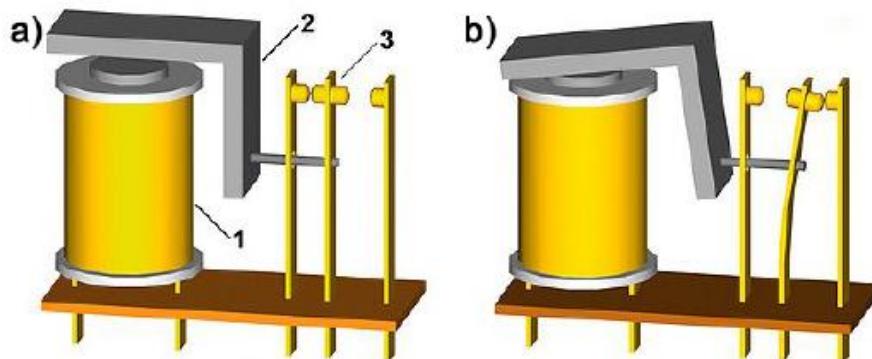
Prilikom rešavanja datog zadatka korišten je USB2.0 sa USB mini B konektorom, da bi za komunikaciju mogao da se koristi kabal, koji se koristi i za programiranje mikrokontrolera. USB 2.0 na izlazu daje napon od 5V i maksimalnu struju od 500mA. U paralelu sa napajanjem vežu se dva "decoupling" kondenzatora da bi osigurali stabilno napajanje pri preklapanju logičkih stanja kada čip povuče više struje nego što je potrebno.

2.3 Relej

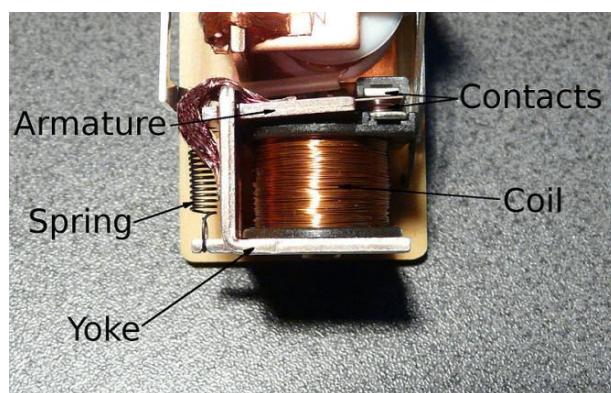
Relej^[6] je jedna od najčešće korištenih elektro komponenti koje se koriste u savremenoj automatizaciji. Imaju niz pogodnosti koje utiču na čestu primenu, kao što je rad u opsegu temperatura od -40°C do +80°C, kao i lako održavanje. Postoji nekoliko vrsta releja: releji snage, step releji, industrijski releji, vremenski releji, itd.

2.3.1 Princip rada releja

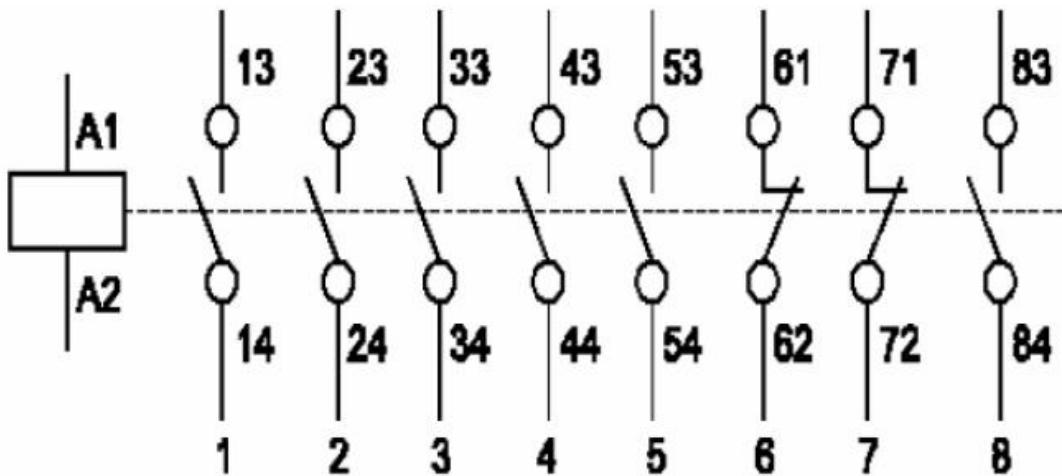
Relej je naprava koja se koristi za prekidanje ili uspostavljanje strujnog kola putem elektromagneta koji otvara i zatvara strujne kontakte.



Slika 2.4 a) Isključen relej i b) Uključen relej



Slika 2.3 Delovi releja



Slika 2.5 Kontakti releja

2.3.1.1 Struja kontakata

Prilikom prekidanja jednosmernih struja usled razdvajanja kontakata dolazi do većeg varničenja nego kod naizmeničnih struja. Kod naizmeničnih stруја (na 50 ili 60 Hz) dolazi do minimalne struje 100 ili 120 puta u sekundi, zato je i varničenje ređe i manje i kontakti imaju duži vek trajanja. Kontakti su metalni, često presvučeni slojem platine ili srebra radi bolje provodnosti.

2.3.2 Radni napon i struja

Na relejima su obično dati podaci slični ovima: 30 V_{DC}, 250 V_{AC}, 10 A. To znači da je radni napon elektromagneta 30 V jednosmernog napona (DC), a kontakti su predviđeni za prekidanje naizmeničnog napona (AC) od 250 V, pri čemu je struja ne veća od 10 A. Česti radni naponi elektromagneta (zavojnice, primarnog kola) releja su 5, 12, 24 V. Česti radni naponi kontakata su 12 i 24 V jednosmernog napona (DC), i 120 V i 220 V naizmeničnog napona (AC).

Relej koji je korišten za realizaciju uređaja, koji dat u ovom radu, ima identične karakteristike kao što je navedeno u opštim podacima s tim da je minimalan napon radnih kontakti jednosmernog napona 5V. Napajanje releja, kao i mikrokontrolera je obezbeđeno pomoću jednog USB kabla.

2.3.3 Prednosti i mane releja

2.3.3.1 Prednosti releja

1. U odnosu na poluprovodničke komponente daje idealan kratak spoj kada je aktiviran, bez obzira na polaritet.
2. Lako prilagođavanje različitim naponima
3. Temperaturna nezavisnost (od -40° do 80°C)
4. Visok otpor između isključenih kontakata
5. Moguće uključivanje/isključivanje većeg broja nezavisnih električnih kola
6. Prisutno galvansko razdvajanje između upravljačkog i glavnog (radnog) električnog kola
7. Jednostavno održavanje

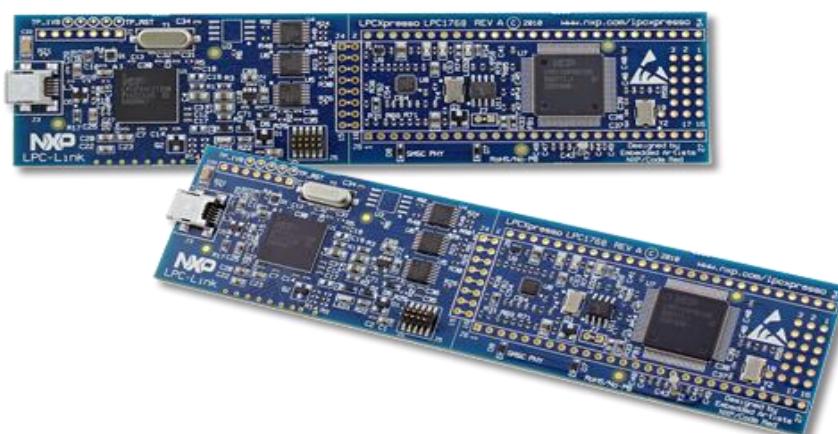
2.3.3.2 Mane releja

1. Zahtevaju dosta prostora
2. Javljuju se šumovi pri reagovanju (električno zagađenje mreže)
3. Ograničena brzina reagovanja (od 3 do 17 ms)
4. Osetljivi na uticaj prljavštine i prašine

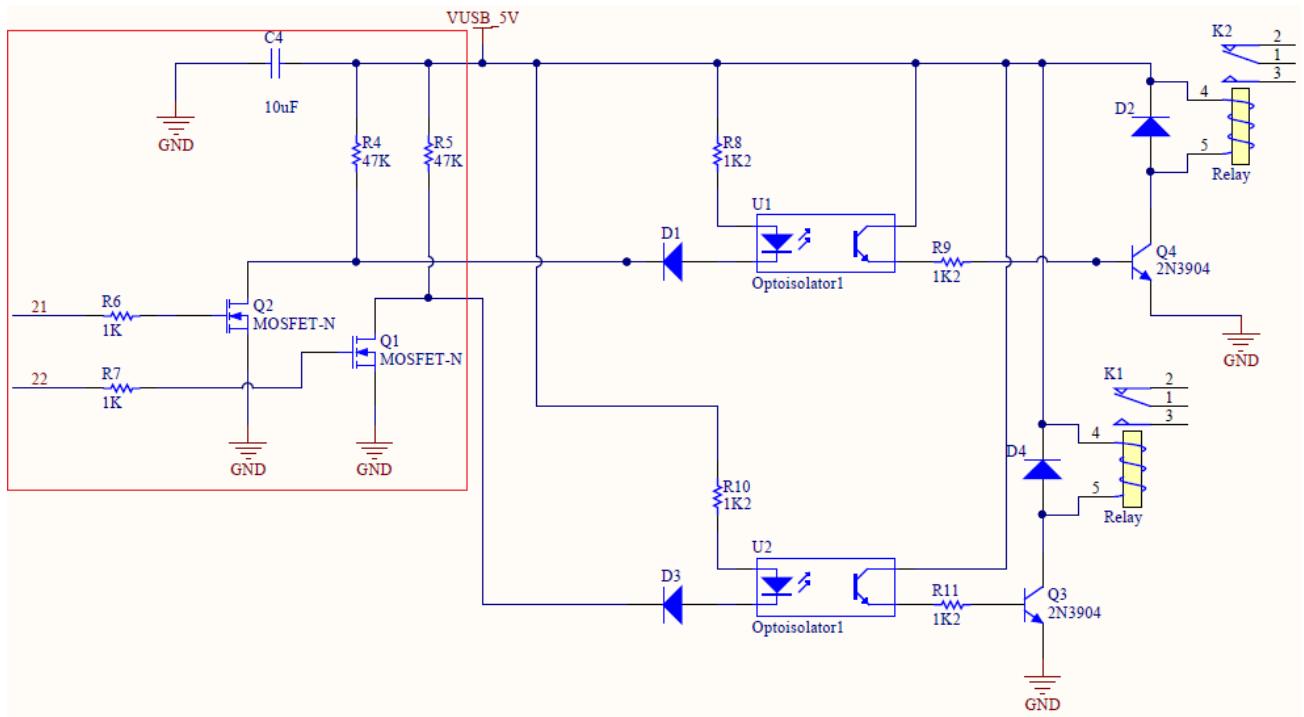
3. Koncept rešenja

Rešenje je realizovano povezivanjem razvojne ploče i ploče sa relejima:

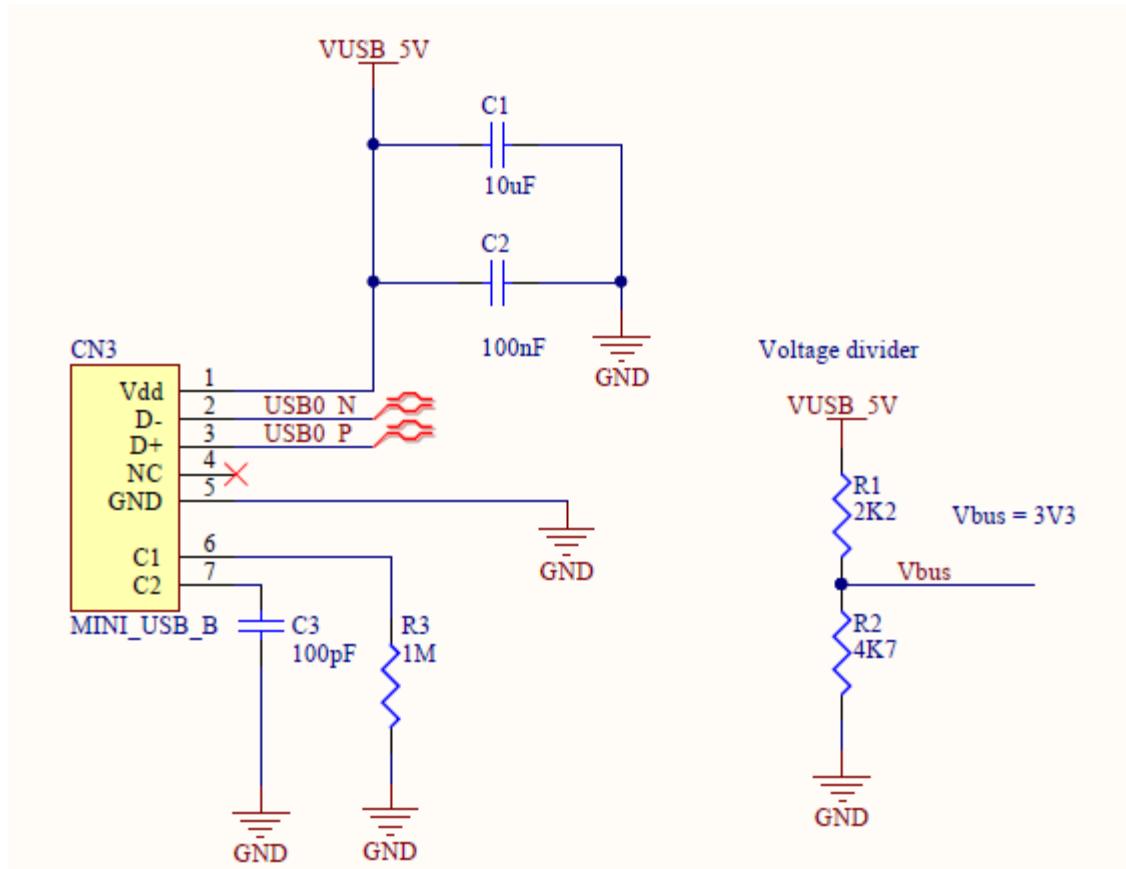
- LPC1769Xpresso (slika 3.1) razvojna ploča kompanije *NXP Semiconductors* poseduje ARM Cortex-M3 mikrokontroler, integriran programator i JTAG debugger. LPC1769 ima 64kB SRAM-a, 512kB “Flash” memorije, 4 UART-a, 3 I2C-a, SPI, 2 SSP-a, 2 CAN-a, PWM, USB2.0 Device/Host/OTG, RTC, Ethernet, I2S, itd. Pored svega navedenog dostupno je i besplatno razvojno okruženje bazirano na *Eclipse* razvojnom okruženju.
- Relejna „click“ ploča je isprojektovana, tako da bi se odgovorilo na dati zadatak. S obzirom da je već na tržištu postojala relejna ploča za *Arduino* razvojna okruženja, odlučeno je da se prepravi ista i prilagodi razvojnom okruženju LPC1769 Xpresso. Na slikama 3.2, 3.3 i 3.4 prikazana je konačna šema relejne „click“ ploče.



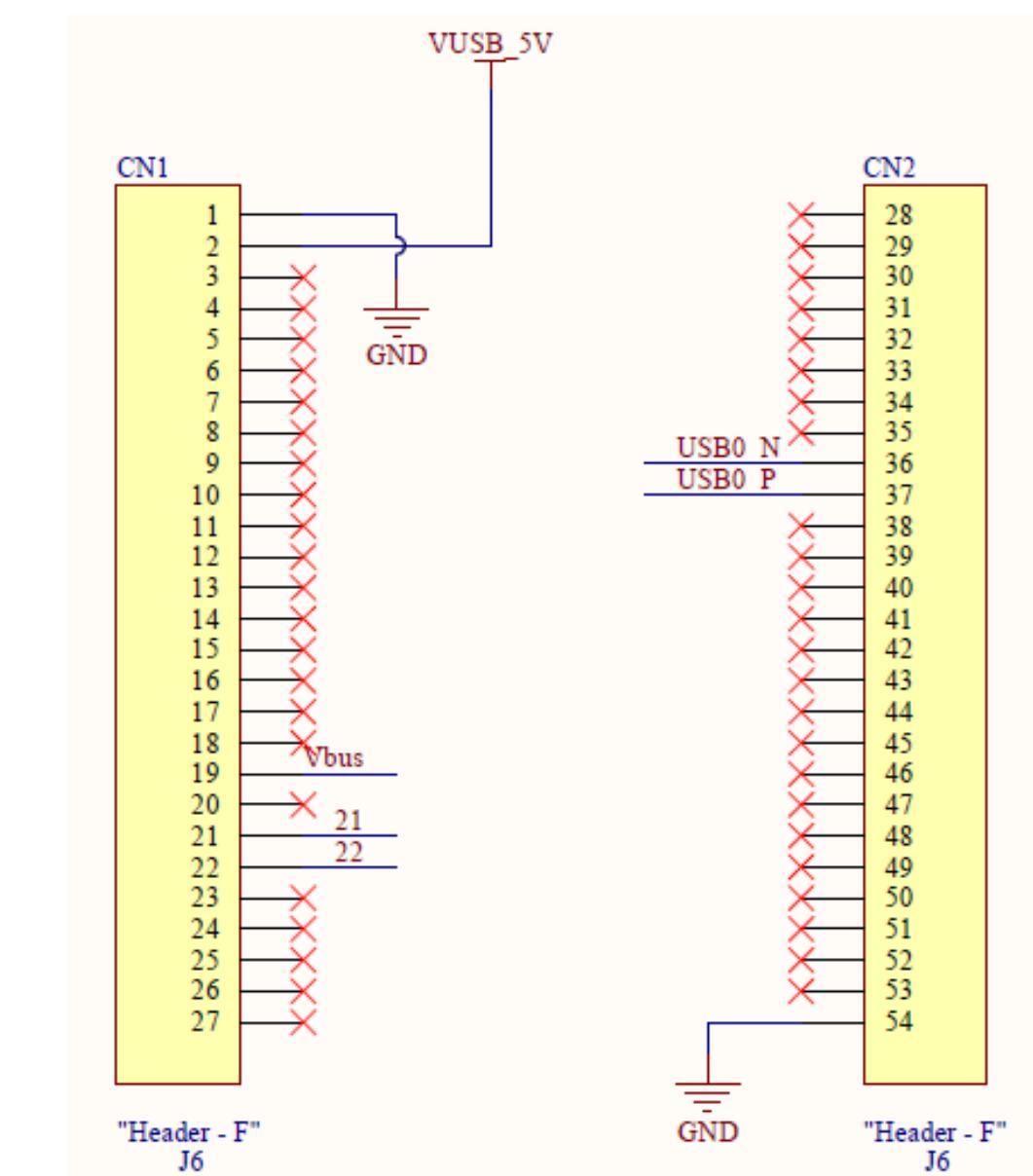
Slika 3.1 LPC1769Xpresso razvojno okruženje



Slika 3.2 Relejna ploča



Slika 3.3 USB napajanje i naponski razdelnik



Slika 3.4 Raspored konektora na LPC1769Xpresso razvojnom okruženju

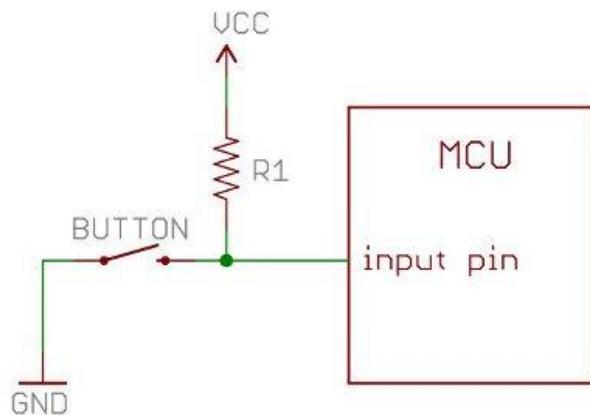


Slika 3.5 Arduino relejna ploča

Arduino relejna ploča je prikazana na slici 3.2 i 3.5. Na slici 3.2 označen je deo koji je dodan na navedenu relejnu ploču. Ulazi relejne ploče na ovom primeru povezani su pomoću konektora 21 i 22 sa razvojnim okruženjem *LPC1769Xpresso*. Konektori 21 i 22 su GPIO konektori.

Prepravka relejne ploče podrazumeva ubacivanje MOSFET-ova između GPIO konektora na razvojnog okruženju i ulaznih pinova relejne ploče. MOSFET-ovi u ovom slučaju služe kao prekidači. Pored MOSFET-ova ubaćeni su i PULL UP^[7] otpornici. PULL UP otpornici su spojeni sa napajanjem i *Drain*-om MOSFET-a, što omogućava da na izlazu konstantno bude ili nizak logički nivo „0“ ili visok logički nivo „1“.

Arduino relejna ploča radi na inverznoj logici, što znači kada je na ulazu doveden nizak logički nivo relej će se uključiti, i obrnuto. Kada se sa izlaza mikrokontrolera pošalje visok logički nivo „1“ na *Gate* MOSFET-a, MOSFET se aktivira i samim tim ulazni pin *Arduino* relejne ploče direktno povezuje sa uzemljenjem (GND). Trenutni protok struje kroz otpornik ide direktno na uzemljenje, pa na ulazu relejne ploče dobija se nizak logički nivo „0“, što u ovom slučaju aktivira relej. Isto važi i za obrnutu logiku, kada je na *Gate*-u MOSFET-a nizak logički nivo „0“.



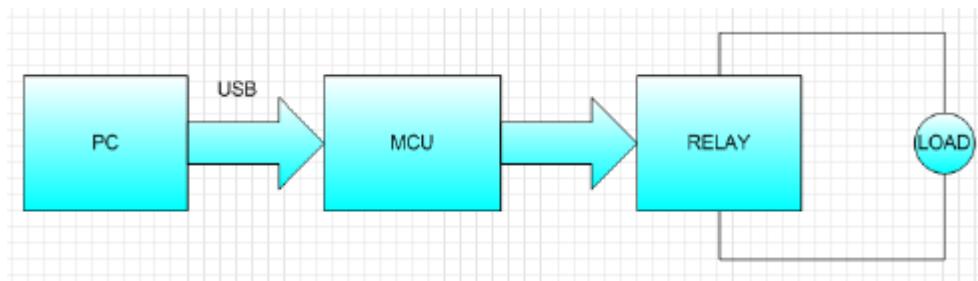
Slika 3.6 PULL UP otpornik

Napajanje relejne „click“ ploče realizovano je pomoću USB konektora. USB, pored komunikacije sa računarom i napajanja relejne „click“ ploče, omogućava i napajanje razvojnog okruženja.

4. Programsko rešenje

U ovom poglavlju predstavljena su programska rešenja za kontrolu releja putem USB-a. Rešenje je realizovano u programskom jeziku C i u programskom jeziku C#. Radna okruženja koja su korištena prilikom izrade su *LPCXpresso (Eclipse IDE)* i *Microsoft Visual Studio 2010*.

Relejna „click“ ploča realizovana je kao USB uređaj koji je moguće povezati sa personalnim računarom (PC). Da bi slao komande za paljenje i gašenje releja, potrebno je da korisnik poveže navedeni uređaj sa računarcem putem USB kabla i da pokrene aplikaciju koja je napravljena za ovaj uređaj. Aplikacija je realizovana tako da korištenje iste bude što jednostavnije.



Slika 4.1 Komunikacija PC-a sa relejom

4.1 Programska podrška

Programska podrška za uređaj pisana je u *C* programskom jeziku. Sastoji se od dva dela, a to su USB modul i reljefni modul.

4.1.1 USB komunikacija

Modul za USB komunikaciju realizovan je pomoću *USB_CDC.lib* biblioteke. Ova biblioteka se može pronaći na sajtu proizvođača razvojnog okruženja. Ono što je karakteristično za ovu biblioteku jeste da ona USB prolaz (*engl. Port*) emulira kao *UART*^[8] prolaz. *UART* (univerzalni asihroni prijemnik/predajnik) je dvožična asinhrona serijska komunikaciona sprega. *UART* pruža podršku za komunikaciju između *CPU* i drugih asinhronih periferija koje koriste standardni *NRZ* (*non return to zero*) format.

UART karakteristike:

- Dva spoljašnja konektora:
 - TxD – (*Transmit Data*) – Slanje podataka
 - RxD – (*Recieve Data*) – Prijem podataka

- Brzina (*Baud Rate = bits per second*):

Kreće se od 9600 do 115200. U ovom rešenju koristi se 115200.

- Half/Full Duplex:

Kada sistem mora istovremeno slati i primati podatke, traži se odvojena fizička arhitektura za ovo. Full Duplex ima odvojenu fizičku arhitekturu za prijem i slanje.

- *Non return to zero* format:

Standard za kodiranje (nivo, ivica i sl.). *LPC1769Xpresso* koristi *non return to zero* standard, što znači :

 - Logička 1 -> Visok logički nivo
 - Logička 0 -> Nizak logički nivo

- Paritet:

Za dodatno poboljšanje integriteta podataka koriste se tehnike pariteta – dodatni bit-ovi koji se šalju sa podacima.

 - *Even parity* = 1, kada je paran broj jedinica u byte-u
 - *Odd parity* = 1, kada je neparan broj jedinica u byte-u

USB modul se sastoji iz sledećih funkcija:

Naziv funkcije	Opis funkcije
vcom_init()	Postavlja određene vrednosti u registar modula za USB komunikaciju, koji su neophodni za ispravnu komunikaciju.
vcom_bread()	Prekidna rutina za prijem podataka, koja isčitava podatke iz prijemnog registra, i upisuje ih u kružni bafer. Vraća bajtove koji su primljeni.
vcom_read_req()	Prekidna rutina za prijem podataka, koja isčitava podatke iz prijemnog registra, i upisuje ih u kružni bafer. Vraća <i>boolean</i> vrednost (<i>true</i> ili <i>false</i>), tj. da li su podaci primljeni ili ne.
vcom_read_count(void)	Proverava kolika je dužina primljenih podataka (<i>bit length</i>).
vcom_connected(void)	Proverava da li je uređaj povezan, ako je povezan vraća non-zero vrednost.
vcom_write()	Prekidna rutina za slanje podataka, ispisuje iz kružnog bafera, i prazni ga.

Tabela 4.1 Funkcije USB biblioteke

4.1.2 Upravljanje relejima

Relejni modul se sastoji iz sledećih funkcija:

Naziv funkcije	Opis funkcije
<code>extern int Set_Relay_Pins_As_Output(void)</code>	Podešava konektore koji se koriste za kontrolu releja kao izlazne konektore.
<code>extern int Random_Buttons_Protection()</code>	Filtrira unos karaktera,tj. svi karakteri koji nisu definisani da obavljaju neku funkciju na releju, nisu validni. Dok god korisnik ne unese validan karakter, relej neće biti moguće promeniti trenutno stanje releja.
<code>int Status_function(void)</code>	Ispisuje status za svaki relej, tj. koji od releja je uključen, a koji isključen.
<code>int Turn_Relay_1_on(void)</code>	Uključuje prvi relej.
<code>int Turn_Relay_1_off(void)</code>	Isključuje prvi relej
<code>int Turn_Relay_2_on(void)</code>	Uključuje drugi relej.
<code>int Turn_Relay_2_off(void)</code>	Isključuje drugi relej.

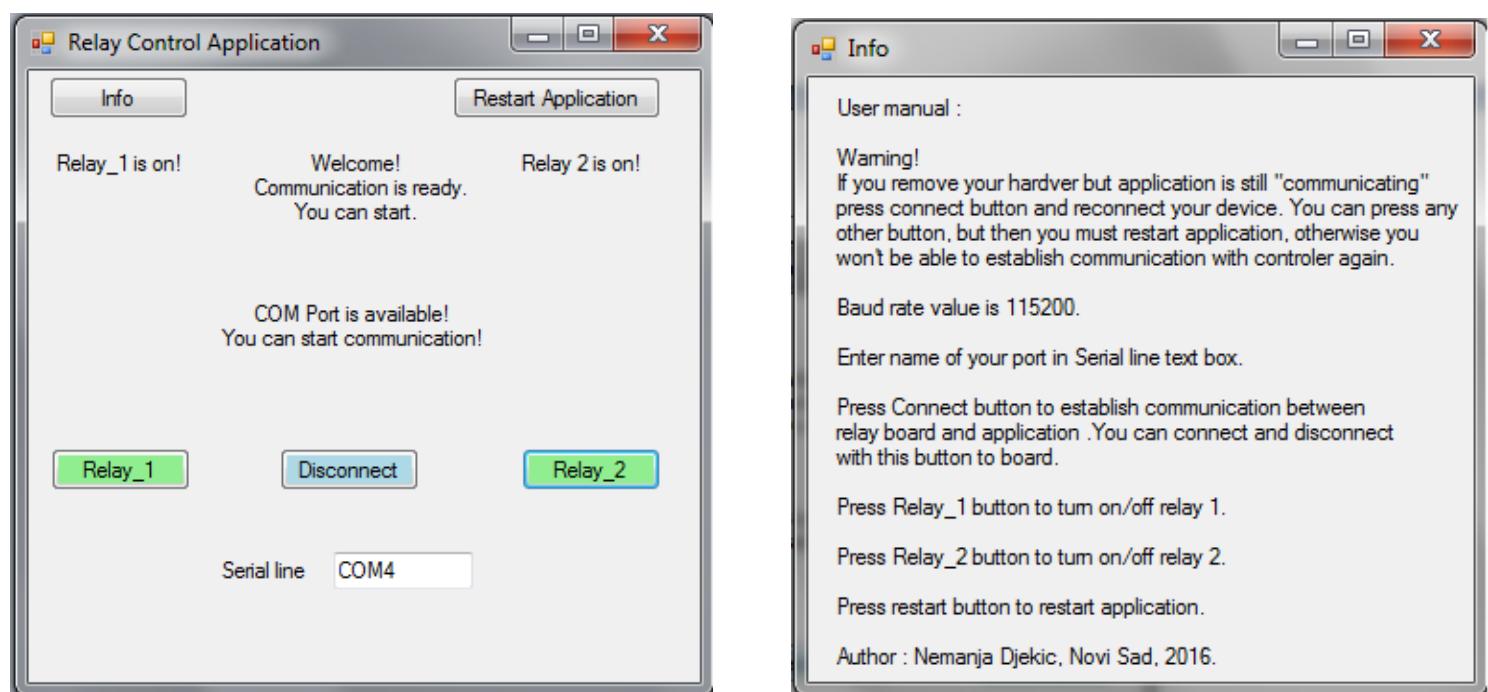
Tabela 4.2 Funkcije za upravljanje relejima

4.2 Grafička korisnička sprega (GUI)

Grafička korisnička sprega omogućava korisniku povezivanje sa uređajem (dugme *connect*) i upravljanje relejima (paljenje/gašenje) pomoću dugmeta *Relay_1* i dugmeta *Relay_2*. U polje *Serial_line* korisnik unosi ime COM prolaza (*engl. Port*) koji uređaj koristi, sve dok nije unesen aktivan prolaz, dugmad neće biti moguće koristiti.

Grafička korisnička sprega realizovana je u C# programskom jeziku koristeći WFD (*Windows Form Designer*). WFD koristi klase iz biblioteke .NET *Framework class library*.

Grafička korisnička sprega se sastoji iz dva dijalog prozora. Prvi dijalog prozor sadrži dugmad za kontrolu releja i povezivanje. Drugi dijalog prozor se otvara iz prvog dijalog prozora pritiskom na dugme info. Ovaj dijalog prozor prikazuje uputstva za korištenje aplikacije.



Slika 4.2 Izgled grafičke korisničke sprege (Prvi i drugi dijalog prozor)

Za realizovanje serijske komunikacije u C#-u korištena je klasa *SerialPort Class* iz .NET Framework biblioteke, koja uproštava korištenje Windows-ovih komandi za rad sa serijskim prolazom.

Pritiskom na dugme „Connect/Disconnect“ otvara/zatvara se izabrani prolaz i PC se povezuje za uređajem.

Da bi se proverio status uređaja, tj. da li je uređaj povezan ili ne, korišten je tajmer, uz pomoć kojeg se na svakih 10ms šalje neki podatak na uređaj i čeka se povratna informacija („PING-PONG“). U slučaju da je komunikacija sa uređajem fizički (nasilno) prekinuta, prilikom rada sa uređajem, bilo koja sledeća komanda iz GUI-a rezultiraće isključivanjem (blokiranjem) dugmadi, sve dok se uređaj ponovo ne poveže. U C#-u postoji već definisana klasa *Timer Class* u .NET Framework biblioteci, koja je korištena u ovom slučaju.

Naziv funkcije	Opis funkcije
<code>void relay1_Click()</code>	Funkcija za paljenje i gašenje prvog releja.
<code>void relay2_Click()</code>	Funkcija za paljenje i gašenje drugog releja.
<code>void connect_Click()</code>	Funkcija za povezivanje aplikacije sa uređajem. Otvara/zatvara izabrani prolaz.
<code>void Restart_btn_Click()</code>	Funkcija za ponovno pokretanje (engl. <i>restart</i>) aplikacije.
<code>void Info_btn_Click()</code>	Otvara drugi dijalog prozor u kome se nalazi uputstvo za korištenje aplikacije.
<code>void Connection_timer()</code>	Funkcija koja proverava da li je uređaj povezan ili ne.
<code>void Stat_Rly_1()</code>	Ispisuje status prvog releja (uključen/isključen).
<code>void Stat_Rly_2()</code>	Ispisuje status drugog releja (uključen/isključen).
<code>void general_Status()</code>	Ispisuje da li je uređaj povezan sa PC-em.
<code>void connection_Status()</code>	Ispisuje da li je izabrani prolaz otvoren/zatvoren.
<code>void textBox1_TextChanged()</code>	Učitava ime prolaza(engl. <i>port</i>) koje korisnik unosi.

Tabela 4.3 Funkcije grafičke korisničke sprege

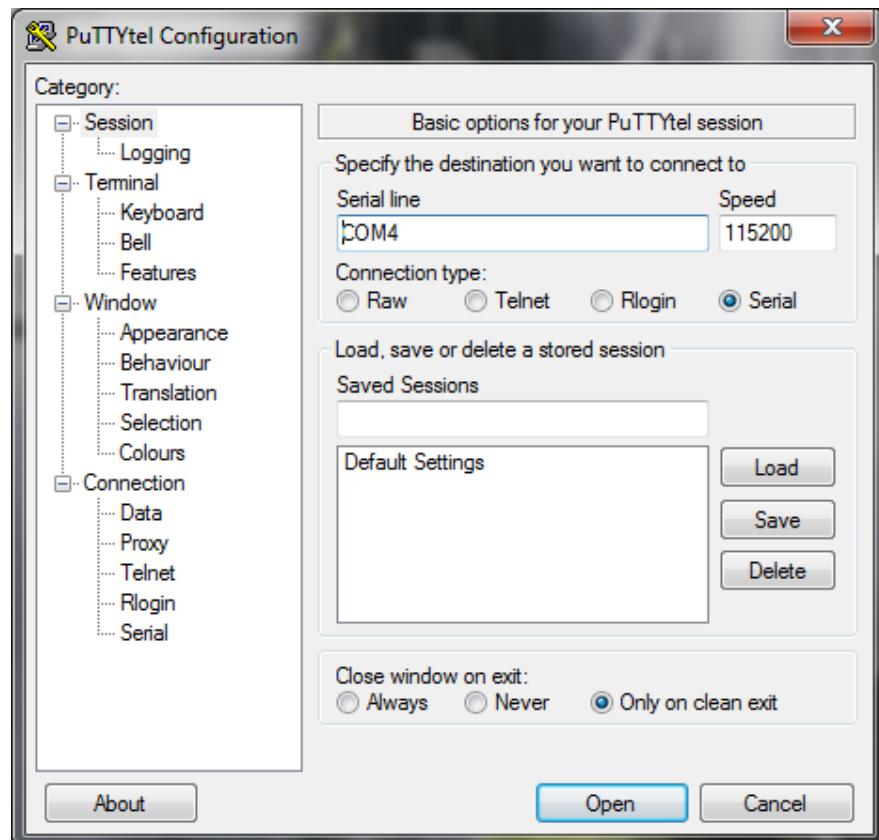
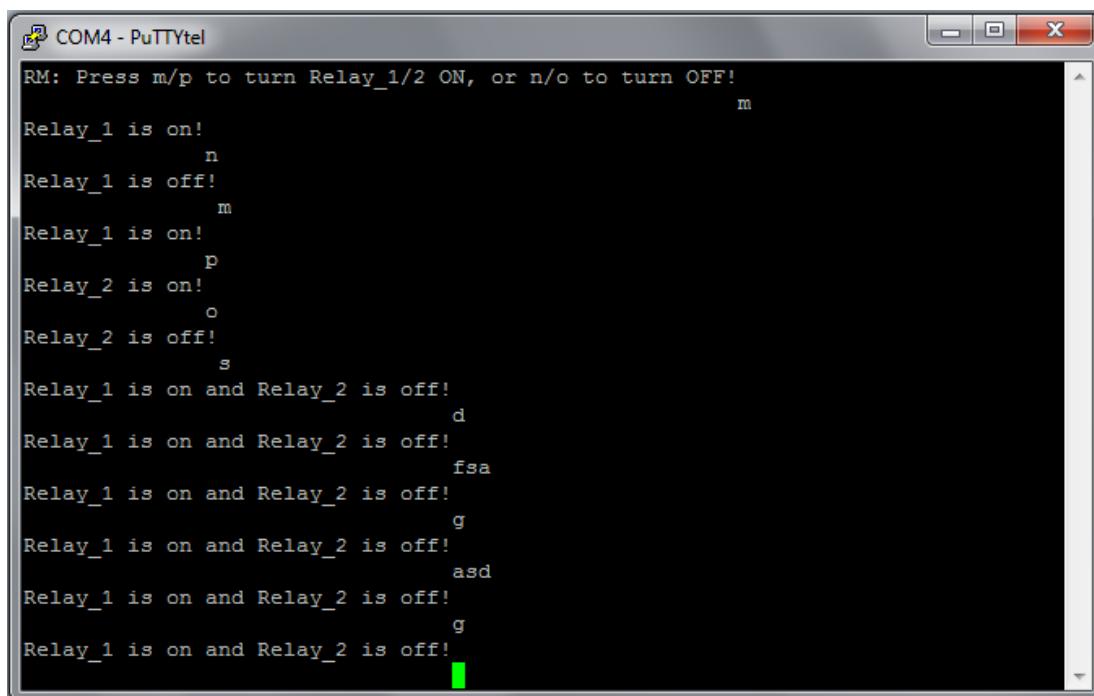
5. Rezultati

U ovom poglavlju opisano je ispitivanje ispravnosti i funkcionalnosti uređaja, kao i aplikacije za krajnje korisnike. Da bi se utvrdila ispravnost, uređaj je podvrsgnut raznim vrstama provera. Ispitivanje su podeljana u dve faze. Prva faza obuhvata ispitivanje ispravnosti i funkcionalnosti uređaja, kao i USB komunikacije. Druga faza obuhvata ispitivanje od strane krajnjeg korisnika

5.1 Ispitivanje i verifikacija uređaja

U cilju ispitivanja ispravnosti uređaja u ovoj fazi korišten je multimetar (unimer) i *PuTTy* (Slika 5.1). aplikacija. Prvo je izvršena provera fizičke arhitekture, tj. da li su sve komponente ispravne i dobro povezane. Proveravani su kratki spojevi, napon i struja na određenim mestima. Kada je utvrđeno da je uređaj ispravan i spreman za dalje ispitivanje, povezan je sa PC-em i započelo je testiranje sa *PuTTy* aplikacijom.

Uz pomoć *PuTTy* aplikacije proveravana je USB komunikacija, kao i paljenje i gašenje releja uz pomoć određenih karaktera, koji su uneseni sa tastature. Na slici 5.2 moguće je videti primer ispitivanja.

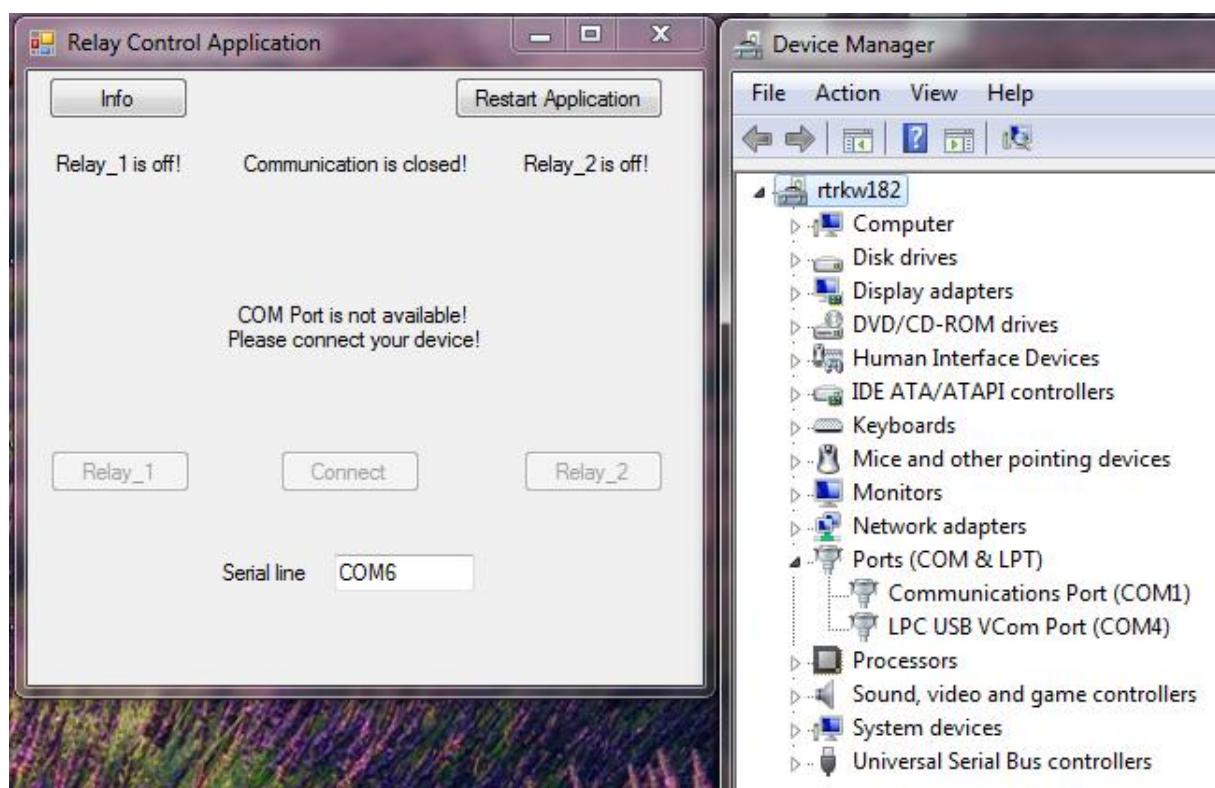
Slika 5.1 Prikaz *PuTTy* aplikacijeSlika 5.2 Ispitivanje uređaja uz pomoć *PuTTy* aplikacije

5.2 Ispitivanje i verifikacija grafičke korisničke sprega

U ovom delu poglavlja ispitivana je grafička korisnička sprega (GUI). Izvršeno je nekoliko vrsta testova, koji su podeljeni po scenarijima.

- Scenario 1:

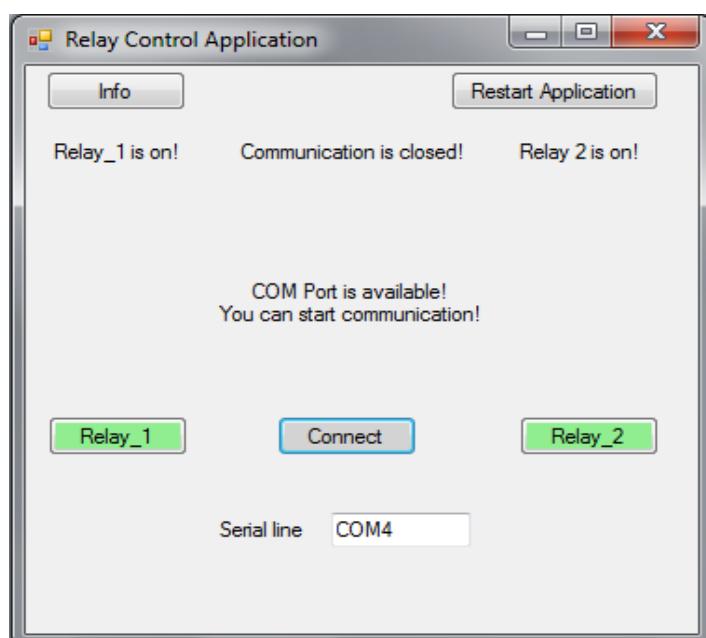
Ponašanje aplikacije u slučaju da korisnik unese ime COM prolaza koji ne postoji, a uređaj je povezan sa PC-em. Ukoliko korisnik uradi ovako nešto dugmad za kontrolu releja ostaju isključena, sve dok se ne unese ime postojećeg COM prolaza. Ovo važi takođe i za slučaj kada uređaj nije fizički povezan sa PC-em.



Slika 5.3 Scenario 1

- Scenario 2:

Ponašanje aplikacije u slučaju da korisnik uključi jedan ili oba releja i zatvori COM prolaz (port). Ukoliko se desi ovako nešto pamti se stanje u komu su releji ostali, pa kada korisnik ponovo otvorи prolaz moguće je nastaviti rad sa aplikacijom. Kada je prolaz zatvoren, a releji uključeni, nije moguće menjati trenutno stanje releja, kao ni položaj dugmadi koji služe za upravljanje istim. Ako korisnik npr. uključi oba releja i zatvori prolaz, releji će ostati uključeni. Kada korisnik ponovo otvorи prolaz, releji ostaju uključeni i moguće ih je isključiti.

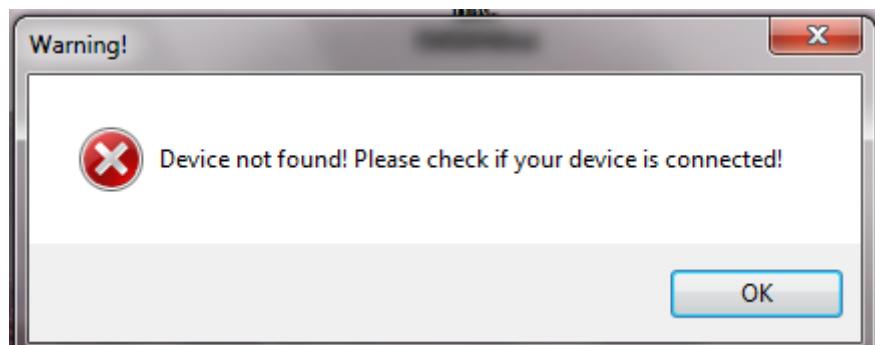


Slika 5.4 Scenario 2

- Scenario 3:

Poslednji scenario predstavlja ponašanje aplikacije u slučaju da je uređaj povezan sa PC-em i prolaz otvoren, međutim korisnik (nasilno) fizički prekine vezu između uređaja i PC-a. Ukoliko korisnik na ovaj način prekine vezu između aplikacije i uređaja, neće biti vidljivih promena u

grafičkoj korisničkoj sprezi, sve dok korisnik ne pritisne neki od dugmadi za relej ili prolaz. Kada korisnik pritisne bilo koji od navedenih dugmadi pojaviće se dijalog prozor sa greškom i sva dugmad momentalno će biti isključena (blokirana). Kada se uređaj ponovo fizički poveže, moguće je ponovo započeti komunikaciju. U slučaju da prilikom ponovnog povezivanja aplikacija ne detektuje uređaj, aplikaciju je moguće dugmetom „*Restart Application*“ ponovo pokrenuti.



Slika 5.5 Scenario 3

Svi prethodno navedeni scenariji su izvršeni, te je aplikacija spremna za korištenje.

6. Zaključak

U ovom radu prikazana je realizacija uređaja koji se koristi za prekidanje i uspostavljanje strujnog kola. Ovaj uređaj povezan je pomoću USB-a sa personalnim računarom, odakle je moguće upravljati njime uz pomoć grafičke korisničke sprege.

Ovaj uređaj omogućava unapređenje procesa automatizacije ispitivanja DUT-a (engl. *Device Under Testing*). Automatizovana ispitivanja DUT-a se često izvode tokom noći, jer se neretko dešava da je broj ispitnih slučajeva jako velik ili da je vreme trajanja ispitnih slučajeva relativno dugo. U slučaju da se ispitivanja prekinu iz bilo kog razloga, pre nego što se izvrše svi slučajevi, potrebno je ponovo pokrenuti (engl. *Restart*) ili ponovo incijalizovati (engl. *Hard Reset*) DUT i započeti ponovo ili nastaviti ispitivanje od mesta prekida. Ovaj uređaj kao deo automatizovanog ispitnog okruženja omogućava ponovno pokretanje DUT-a tako da osoba, koja je zadužena za kontrolu ispitivanja, ne mora biti fizički prisutna na mestu na kome se ispitivanja odvijaju.

Pored automatizovanog ispitnog okruženja, uređaj se može koristiti i u bilo kojoj drugoj nameni, gde je potrebno izvršiti kontrolisano prekidanje strujnog kola pomoću prekidača koji je galvanski odvojen.

Mogući pravci daljeg rada na unapređenju rešenja mogući su unapređenjem postojećih funkcionalnosti ili dodavanjem novih. Modularna struktura rešenja omogućava integraciju izmena i dopuna, bez velikih poteškoća.

7. Literatura

- [1] Web stranica ARM products, *ARM Cortex M3 Proccesor*, dostupno na: <http://www.arm.com/products/processors/cortex-m/cortex-m3.php>, učitana 15.6.2016.
- [2] Web stranica Wikipedia, *The Free Encyclopedia*, *ARM arhitektura*, dostupno na: https://sr.wikipedia.org/sr/ARM_%D0%B0%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0#32-D0.B1.D0.B8.D1.82.D0.BD.D0.B0.D0.B0.D1.80.D1.85.D0.B8.D1.82.D0.B5.D0.BA.D1.82.D1.83.D1.80.D0.B0, učitana 15.6.2016.
- [3] Web stranica Computer hope, *USB technology*, dostupno na: <http://www.computerhope.com/jargon/u/usb.htm>, učitana 16.6.2016.
- [4] Web stranica Pericom, “*Three things you should know about USB switches*”, dostupno na: <https://www.pericom.com/support/technical-articles/3-things-you-need-to-know-about-usb-switches/>, učitana 16.6.2016.
- [5] Web stranica Wikipedia, *The Free Encyclopedia*, *USB*, dostupno na: <https://hr.wikipedia.org/wiki/USB>, učitana 16.6.2016.
- [6] Web stranica Automatika, *Baza znanja*, dostupno na: <https://www.automatika.rs/baza-znanja/teorija-upravljanja/releji.html>, učitana 17.6.2016.
- [7] Web stranica Sparkfun, *Pull Up resistors*, dostupno na: <https://learn.sparkfun.com/tutorials/pull-up-resistors>, učitana 17.6.2016.

- [8] Web stranica prof.dr. Lejla Banjamović-Mehmedović, *Serijska komunikacija*, Univerzitet u Tuzli, Fakultet elektrotehnike, 2013. , dostupno na: http://www.lejla-bm.com.ba/SRV/SRV_LV6.pdf, učitano 17.6.2016.