Interaktivna igra s arduino platformom

Špoljar, Domagoj

Undergraduate thesis / Završni rad

2017

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:073386

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-28**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

zir.nsk.hr

Image not found or type unknown

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij računarstva

Interaktivna igra s Arduino platformom

Završni rad

Domagoj Špoljar

Osijek, 2017. godina.

SADRŽAJ

1. UVOD 1

1.1. Zadatak završnog rada 1

2. MIKROUPRAVLJAČKI SUSTAVI 2

2.1. Prvi integrirani računalni sustav 2

2.2. Arduino 4

3. INTERAKTIVNE IGRE 7

3.1. Igra „Toplo-Hladno“ 7

3.2. Igra „Zaustavi me“ 8

3.3. Igra „Simon“ 8

3.4. Igra „Pronađi broj“ 9

3.5. Prikaz preostalog vremena igranja 10

3.6. Zvučni efekti 10

4. IMPLEMENTACIJA I TESTIRANJE 11

4.1. Sklopovlje 11

4.2. Programska podrška 16

4.2.1. Program igre „Toplo-Hladno“ 16

4.2.2. Program igre „Zaustavi me“ 18

4.2.3. Program igre „Simon“ 19

4.2.4. Program igre „Pronađi broj“ 20

4.3. „Setup()“ i „Loop()“ programski kod 21

4.4. Testiranje 22

5. ZAKLJUČAK 23

LITERATURA 24

SAŽETAK 25

ABSTRACT 26

ŽIVOTOPIS 27

PRILOZI 28

1. UVOD

U ovom je završnom radu potrebno na temelju Arduino platforme izraditi interaktivnu igru s najmanje dva načina rada. Interaktivna igra zamišljena je kao spoj četiri jednostavnije igre koje se istovremeno pokreću te korisnik može sam birati redoslijed rješavanja. Prilikom pokretanja interaktivne igre pokreće se i brojač koji pokazuje preostalo vrijeme za rješavanje igara. Cilj igre jest u određenom vremenskom roku uspješno riješiti sve jednostavnije igre i time zaustaviti brojač. Igra je realizirana pomoću elektroničkih komponenti (LE dioda, tipkala, potenciometra itd.) spojenih na Arduino Mega platformu. Time se omogućava interaktivnost korisnika s igrom. Cijela je igra isprogramirana pomoću Arduino programskog okruženja u C++ programskom jeziku.

1.1. Zadatak završnog rada

U ovom završnom radu potrebno je na temelju Arduino platforme izraditi interaktivnu igru s najmanje dva načina rada. U praktičnom dijelu rada potrebno je sklopovski izraditi igru pomoću raznih elektroničkih komponenti spojenih na Arduino platformu.

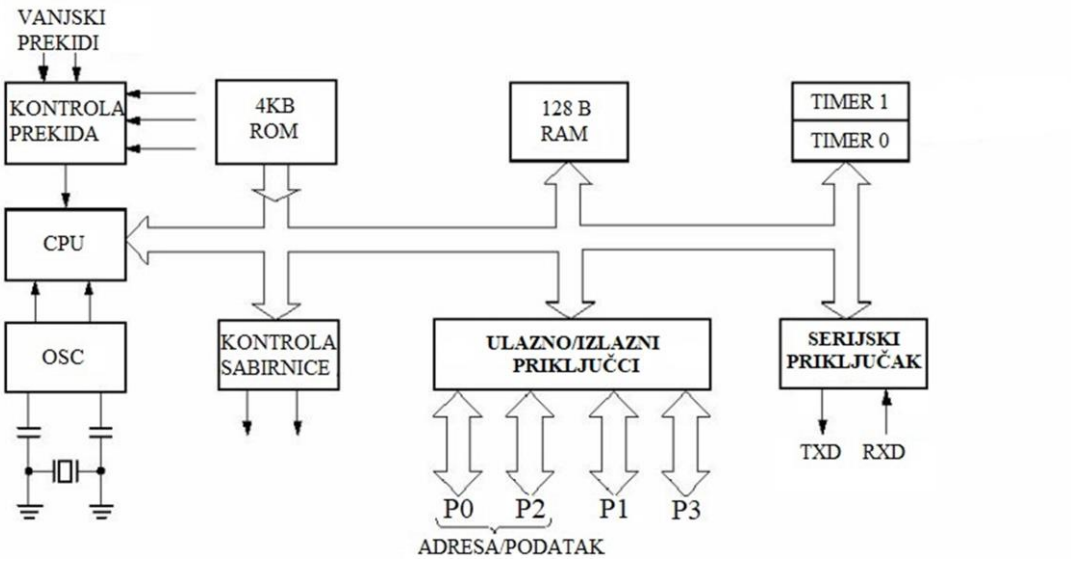
2. MIKROUPRAVLJAČKI SUSTAVI

Mikroupravljač (*eng. Microcontroller*) je kompletni računalni sustav koji se nalazi u jednom integriranom sklopu. Mikroupravljač se sastoji od mikroprocesora, memorije, digitalnih i analognih ulaza i izlaza, digitalnih satova (*eng. timer*), brojača (*eng. counter*), oscilatora, komunikacijskih sklopova i ostalih dodataka [1]. Njegovu namjenu moguće je programirati po želji tj. ona nije unaprijed određena. Razlog je tome to što mikroupravljač nema unaprijed definirane ulaze i izlaze. Ovisno o potrebi, moguće je spojiti razne elektroničke komponente, senzore i module na mikroupravljač poput LE dioda, potenciometara, fotootpornika, tipkala, senzora pokreta i sl. Uz to, mikroupravljač nije unaprijed programiran da izvodi određenu operaciju, nego ga se može isprogramirati po želji i potrebi. Mikroupravljači su danas široko raspostranjeni u raznim modernim uređajima; od robota, automobila i mobilnih telefona do satelita, medicinskih uređaja i igračaka. Zbog toga postoji veliki broj različitih vrsta mikroupravljača koji se razlikuju po veličini i rasponu memorije, brzini rada mikroupravljača, broju ulazno-izlaznih jedinica itd. Ovisno o potrebi i namjeni, dostupni su mikroupravljači sa 4-bitnim mikroprocesorima pa sve do 32-bitnih mikroprocesora. Što se tiče memorije, mikroupravljači uglavnom koriste RAM, flash, EPROM ili EEPROM memoriju. Svaka se od tih memorija koristi za različite funkcije unutar mikroupravljača. [2]

Potrebno je naglasiti kako mikroupravljač i mikroprocesor nisu isti sklopovi. Mikroprocesor je samo „mozak“ sustava te mu je potrebno dodati i druge komponente (npr. RAM i ROM memoriju) kako bi bio funkcionalan. Mikroupravljač sadrži sve potrebne komponente za rad unutar integriranog sklopa. Mikroupravljači su optimizirani za ugrađene primjene, suprotno mikroprocesorima koji se koriste u osobnim računalima [3].

2.1. Prvi integrirani računalni sustav

Na slici 2.1. prikazana je unutrašnja struktura Intel 8051 mikroupravljača. Intel 8051 (poznat kao i Intel MCS-51) je prvi integrirani računalni sustav koji je optimiziran za upravljanje procesima. Proizveden je 1980. godine te je ubrzo nakon toga postao jedan od najkorištenijih mikroupravljača u svijetu. Dijelovi koje ovaj mikroupravljač sadrži uglavnom se nalaze i u drugim, modernijim, mikroupravljačima. Unutrašnja struktura Intel 8051 mikroupravljača je detaljnije objašnjena u tekstu ispod.



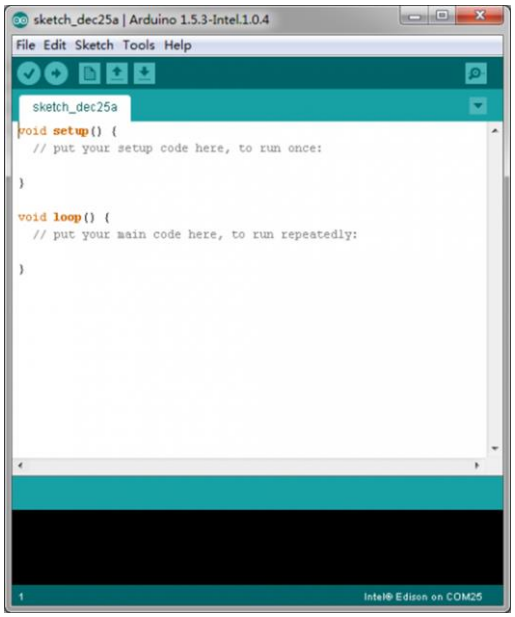
Sl. 2.1. Struktura INTEL 8051 mikroupravljača.

- CPU (Central processing unit) – Središnja procesorska jedinica, zaslužna za nadzor i upravljanje radom mikroupravljača. Sastoji se od aritmetičko-logičke jedinice (ALU), upravljačke jedinice i registara (priručne memorije). U Intel 8051 CPU je 8-bitni.
- ROM memorija – Statična memorija koja se koristi za čuvanje programskog koda i pohranu nepromjenjivih informacija. Intel 8051 sadrži 4KB ROM memorije.
- RAM memorija – Dinamička memorija namijenjena za upisivanje i čitanje podataka. U ovoj memoriji se nalaze četiri registarske banke po osam registara, kao i stack memorija. Sve je skupa prisutno 128 B RAM memorije za korištenje.
- OSCILATOR – Sklop koji koristi mehaničku rezonanciju vibracije kristala (obično kvarca) kako bi generirao električni signal vrlo precizne frekvencije. Služi za precizno generiranje taktova pomoću kojeg mikroupravljač može izvoditi programski kod. Brzina rada mikroupravljača ovisi o oscilatoru i njegovoj frekvenciji.
- TIMER (Brojač) – Brojači su sklopovi koji su zaslužni za mjerenje vremena unutar mikroupravljača. Pomoću brojača može se točno odrediti kada je potrebno izvesti neki dio programskog koda. Intel 8051 sadrži dva timera koja su neovisna jedan o drugome te se mogu posebno kontrolirati, postavljati i konfigurirati.
- SERIJSKI PRIKLJUČAK - Služi kao komunikacijsko sučelje za komunikaciju računala i mikroupravljača. Intel 8051 koristi UART serijsku komunikaciju. Isto tako, pomoću serijskog priključka moguće je programirati sami mikroupravljač.
- ULAZNO/IZLAZNI PRIKLJUČCI – Intel 8051 sadrži četiri različita priključka za spajanje raznih senzora, uređaja i komponenata koje je moguće programirati.

- KONTROLA PREKIDA – Sklop koji prekida normalnu operaciju mikroupravljača te izvodi druge operacije višeg prioriteta. Intel 8051 sadrži dvije vanjske nožice pomoću kojih je moguće izvesti drugu operaciju od normalne operacije mikroupravljača. Uz to postoji još dva interna vremenska okidača i jedan okidač na serijskom priključku.
- KONTROLA SABIRNICE – Sklop zaslužan za prijenos električnih impulsa iz mikroprocesora u ostale dijelove mikroupravljača i obrnuto. Pomoću njega se odvija komunikacija svih unutrašnjih dijelova mikroupravljača. [4]

2.2. Arduino

Arduino je naziv za računarsku i programsku platformu pomoću koje razni umjetnici, dizajneri i elektroničari mogu napraviti uređaje koji omogućuju spajanje računala s fizičkim svijetom tj. kreirati interaktivne objekte i okruženja. Arduino je razvila talijanska tvrtka SmartProjects 2005. koristeći 8-bitne mikroupravljače Atmel AVR. Cilj im je bio razviti jednostavnu, malu i jeftinu platformu pomoću koje bi se računalo moglo lakše povezati s fizičkim svijetom. Arduino se sastoji od sklopovskog dijela koji je zapravo mala razvojna ploča koja sadrži mikroupravljač i ostale elektroničke komponente potrebne za rad te od programskog dijela poznatijeg kao Arduino IDE (*eng. Integrated Development Environment*) (Sl. 2.2.) pomoću kojeg se programira i upravlja samom pločom. [5]



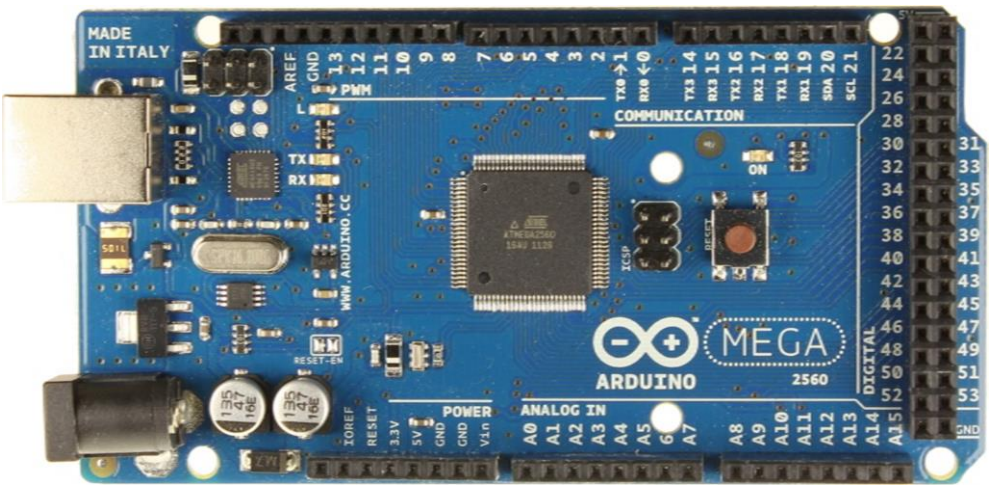
Sl. 2.2. Izgled Arduino IDE programskog okruženja.

Službene Arduino ploče najčešće koriste megaAvr seriju integriranih sklopova, točnije ATmega8, ATmega168, ATmega328, ATmega1280 i ATmega2560. Arduino ploče uglavnom sadrže 5V linearni naponski regulator i 16MHz kristalni oscilator. Mikroupravljači koji se koriste

u Arduino ploči sadrže predinstalirani program za samoučitavanje (*eng Bootloader*) koji pojednostavljuje postupak prebacivanja prevedenog koda u flash memoriju na mikroupravljaču. [6]. Zbog toga je moguće programirati Arduino u pojednostavljenoj inačici C++ programskog jezika koja početnicima olakšava ulazak u svijet sklopovskog programiranja.

Arduino platforma je namijenjena svim učenicima, studentima i ostalim početnicima koji nisu upoznati s načinom razvoja programske podrške. Na internetu postoji velika zajednica ljudi koja dijeli svoje znanje o Arduinu, programiranju i elektronici. Postoji ogroman broj uputa i projekata koje je moguće napraviti pomoću Arduino platformom. Zbog svoje jednostavnosti, pristupačnosti, niske cijene i široke zajednice, Arduino platforma postala je vrlo popularan izbor za ulazak u svijet programiranja i elektronike.

Za realizaciju završnog rada odabrana je Arduino Mega (Sl. 2.3.) razvojna ploča. Arduino Mega sadrži Atmega2560-16AU mikroupravljač koji ima 54 digitalnih ulaza/izlaza (od kojih je 15 PWM izlaza) i 16 analognih ulaza. Napaja se pomoću USB priključka na računalu ili vanjskog napajanja (AC/DC adapter ili baterija). Detaljnije tehničke specifikacije (*eng. datasheet*) prikazane su u tablici 2.1.



Sl. 2.3. Izgled Arduino Mega razvojne ploče.

Tab 2.1. Tehničke specifikacije Arduino Mega ploče.

Mikroupravljač	*Atmega2560-16AU*
Radni napon	*5V*
Ulazni napon (preporučeno)	*7-12V*
Ulazni napon (limit)	*6-20v*
Digitalne IU/I nožice	*54 (15 se može koristiti za PWM)*
Analogni ulazi	*16*
DC struja po jednoj U/I nožici	*40mA*
DC struja za 3.3 V nožicu	*50 mA*
Flash memorija	*128 KB*
SRAM	*8 KB*
EEPROM	*4 KB*
Takt rada	*16 MHz*
Dužina	*2100 mm*
Širina	*4000 mm*

3. INTERAKTIVNE IGRE

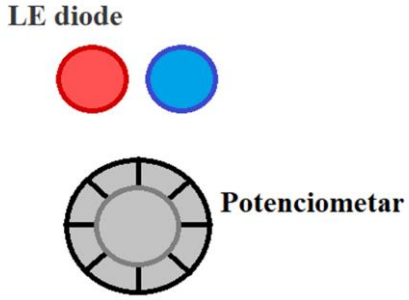
Završni rad sastoji se od četiri neovisne interaktivne igre koje se pokreću istovremeno. Postavljeno je vremensko ograničenje rješavanja igara te je cilj uspješno riješiti sve igre prije isteka vremena. Slijed igranja igara nije bitan nego je ostavljen igraču na izbor. Igre su namijenjene djeci i osnovnoškolcima kojima se na privlačan i interaktivan način potiče vježbanje određenih intelektualnih vještina. Kod izbora igara bilo je bitno da svaka igra potiče i razvija različite kognitivne sposobnosti poput memorije, refleksa, preciznosti i sl. Nakon provedenog istraživanja, osmišljene su sljedeće četiri igre:

1. „Toplo-Hladno“,
2. „Zaustavi me“,
3. „Simon“,
4. „Pronađi broj“.

Uz osmišljene igre, za kompletan završni rad potrebno je osmisliti pokazivač preostalog vremena u igrama te zvučne efekte koji dodatno osiguravaju interakciju s igrama. U daljnjem će se tekstu detaljno razraditi i objasniti svaka igra kao i ostali dijelovi kompletne igre.

3.1. Igra „Toplo-Hladno“

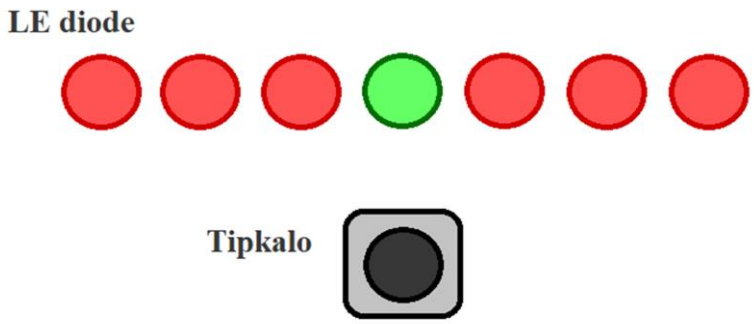
„Toplo-Hladno“ je kratka igra u kojoj igrač treba pronaći određen položaj potenciometra koristeći LE diode kao smjernice. Igra se sastoji od dvije LE diode od kojih je jedna crvene a druga plave boje te jedan potenciometar (Sl. 3.1.). Okretanjem potenciometra mijenja se intenzitet osvjetljavanja crvene i plave LE diode. Plava LE dioda je pokazatelj „hladne temperature“ koju treba izbjegavati a crvena boja pokazatelj „tople temperature“ koju je potrebno pronaći. Cilj igre je pronaći položaj potenciometra u kojem crvena LE dioda svijetli a plava ne svijetli. Postoji samo jedna takva vrijednost te ju igrač treba pronaći preciznim okretanjem potenciometra. Ova je igra osmišljena s ciljem vježbanja vještine preciznosti kod igrača.



Sl. 3.1. Simbolički prikaz igre „Toplo-Hladno“.

3.2. Igra „Zaustavi me“

„Zaustavi me“ je jednostavna i praktična igra osmišljena s ciljem vježbanja refleksa i brzine reakcije. Sastoji se od sedam LE dioda od kojih su šest crvene i jedna zelene boje te jednog tipkala (Sl. 3.2.). LE diode su poredane u niz s time da je zelena dioda u sredini (četvrta po redu). Kada se igra pokrene, LE diode počinju sekvencijalno svijetliti. Prvo se osvijetli prva LE dioda, pa druga i tako sve do zadnje nakon čega ponovno počinju svijetliti u suprotnom smjeru. Samo je jedna LE dioda upaljena u određenom trenutku. Pored njih se nalazi spomenuto tipkalo koje služi za zaustavljanje pokreta osvjetljenja LE dioda. Igrač može u bilo kojem trenutku pritisnuti tipkalo. Nakon pritiska, LE dioda koja je u tom trenutku bila osvijetljena ostaje svijetliti. Cilj je igre pritisnuti tipkalo u trenutku kada svijetli zelena LE dioda i time se prelazi na sljedeću razinu. Ako se tipkalo pritisne kada svijetli jedna od 6 crvenih LE dioda, to se prepoznaje kao pogreška te se igra ponovno postavlja na prvu razinu. Sveukupno postoji pet razina koje je potrebno uspješno riješiti. Svaka sljedeća razina ima veću brzinu osvjetljavanja LE dioda čime se povećava složenost rješavanja igre. Nakon riješene pete razine, igra prestaje s radom te je igrač uspješno riješio cijelu igru.

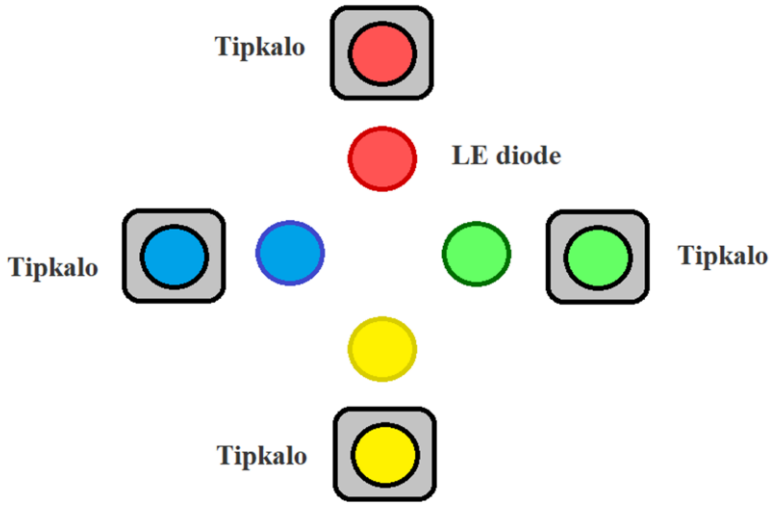


Sl. 3.2. Simbolički prikaz igre „Zaustavi me“.

3.3. Igra „Simon“

Igra „Simon“ je rekreacija poznate dječje igre Simon kaže (*eng. Simon says*). Simon je jednostavna igra koja od igrača zahtjeva pažnju i koncentraciju kako bi se uspješno riješila. Igra se sastoji od četiri LE diode različitih boja (crvena, plava, zelena i žuta) i četiri tipkala pripadajućih boja (Sl. 3.3.). Igra započinje osvjetljavanjem jedne od LE dioda. Nakon osvjetljenja igrač mora pritisnuti tipkalo te iste boje. Nakon svakog uspješnog pritiska tipkala, igra ponavlja dosadašnji niz boja, a zatim prikazuje novu boju. Igrač mora pratiti redoslijed osvjetljavanja LE dioda te ga rekreirati pritiskom na tipkala. Cilj je igre uspješno ponoviti cijeli

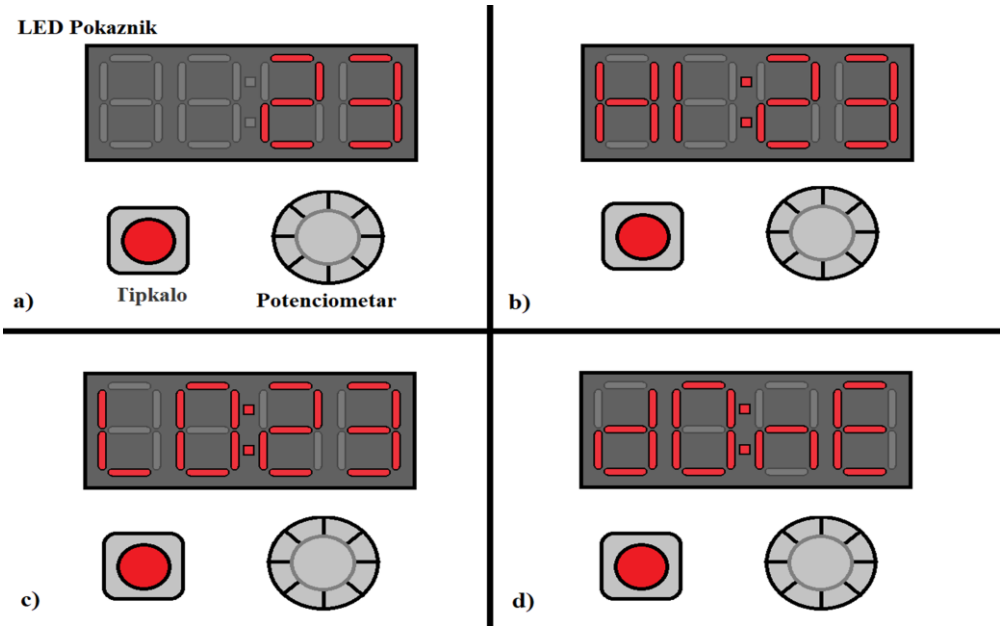
prikazani niz. Ako se pritisne pogrešna tipka koja ne odgovara traženoj boji, igra počinje ispočetka te se generira novi nasumični niz boja koji je potrebno rekreirati. Za uspješno rješavanje ove igre potrebno je pamtiti redoslijed koji se prikaže. Tako se vježba kratkoročno pamćenje i koncentracija kod igrača.



Sl. 3.3. Simbolički prikaz igre „Simon“.

3.4. Igra „Pronađi broj“

Ova igra traži od igrača vrlo jednostavnu stvar – pronaći traženi broj. Prilikom pokretanja igre generira se nasumični broj od 0 do 50 te igrač treba pogoditi generirani broj. Igra se sastoji od potenciometra, tipkala i LED pokaznika (Sl. 3.4.a). Okretanjem potenciometra, na LED pokazniku se sekvencijalno prikazuju brojevi od 0 do 50. Kada se igrač odluči za jedan broj koji je prikazan na LED pokazniku, treba pritisnuti tipkalo te time odabire taj broj. Igrač je uspješno riješio igru ako je odabrani broj isti kao i traženi. Ako odabrani broj nije isti kao traženi, na pokazniku će se prikazati je li odabrani broj veći (slova „HI“ ispred odabranog broja, Sl. 3.4.b) ili manji (slova „LO“ ispred odabranog broja, Sl. 3.4.c) od traženog. Time se igrača navodi prema traženom broju. Nakon toga moguće je opet pogađati broj sve dok se ne pogodi traženi broj. Kada igrač pogodi traženi broj, na pokazniku će se prikazati slova „done“ (Sl. 3.4.d). Postoje mnoge različite metode pogađanja traženog broja. Igraču se prepušta da sam odabere kako će ga pronaći. Najbolji rezultat je onaj s najmanjim brojem pokušaja jer je potrebno najmanje vremena za uspješno rješavanje.



*Sl. 3.4. Simbolički prikaz igre „Pronađi broj“. **a)** način prikaza brojeva na LED pokazniku, **b)** prikaz na LED pokazniku kada je traženi broj veći od odabranog, **c)** prikaz na LED pokazniku kada je traženi broj manji od odabranog, **d)** prikaz na LED pokazniku kada je odabran traženi broj.*

Rješavanju ove igre može se pristupiti i na drugačiji način. Koristeći neke od postojećih algoritama za pretraživanje podataka (npr. linearno, binarno ili interpolacijsko pretraživanje [7]), moguće je uspješno odabrati traženi broj. Igraču se na taj način približavaju pojmovi algoritama i pretraživanja podataka te prikazuju prednosti i nedostatci određenih vrsta pretraživanja.

3.5. Prikaz preostalog vremena igranja

Uz prethodno navedene igre potrebno je igraču prikazati preostalo vrijeme. Kao rješenje je osmišljen pokazivač preostalog vremena. Kada se igre pokrenu, kreće odbrojavanje koje pokazuje koliko je vremena ostalo igraču kako bi uspješno riješio sve igre. Svake sekunde vrijeme se umanjuje i prikazuje na pokazivaču. Vrijeme koje igrač ima na raspolaganju nakon što se igre pokrenu je četiri minute.

3.6. Zvučni efekti

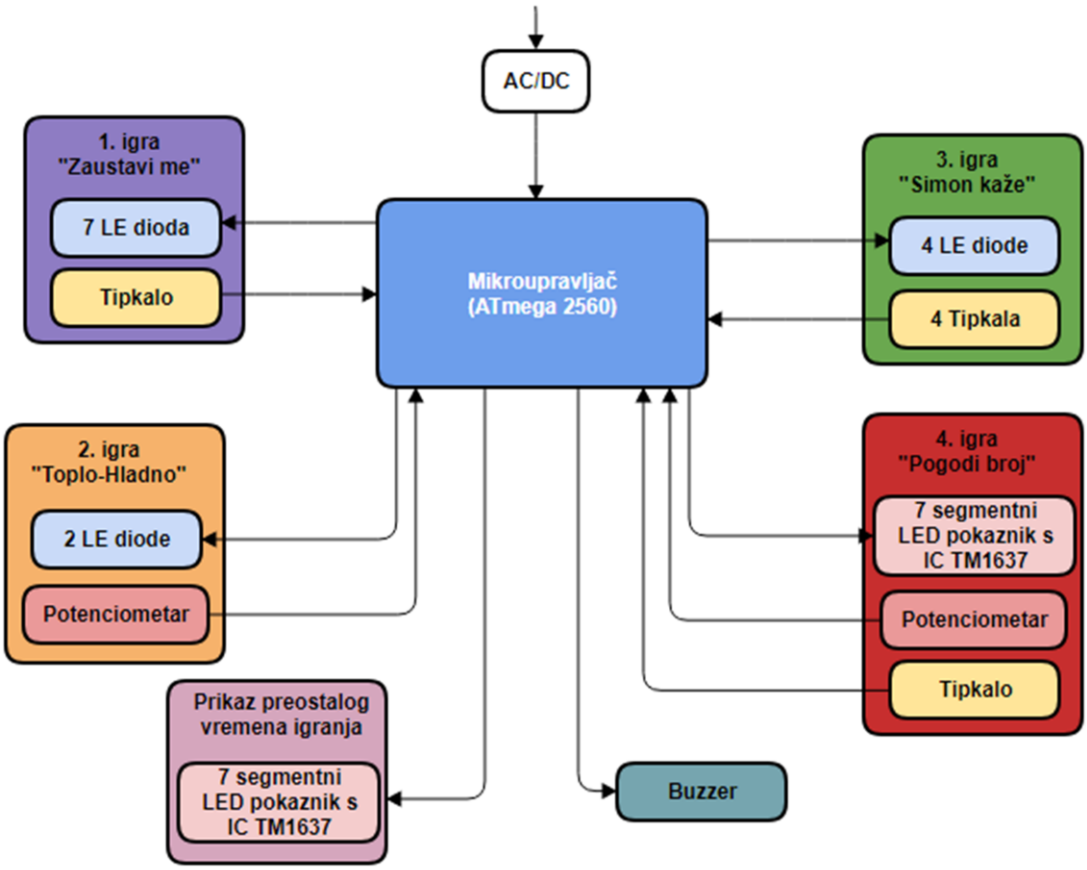
Kako bi igre osigurale dodatnu interakciju s igračem, osmišljeno je implementiranje zvučnih efekata u svaku igru pomoću buzzera. Zvučni efekti su jednostavan način davanja povratne informacije igraču nakon njegove interakcije s igrama. Ako je pritisnuo točnu tipku, prešao na sljedeću razinu ili uspješno riješio jednu od igara, igrač će čuti zvuk koji predstavlja uspješno obavljenu radnju. Ako pak pogriješi u nečemu, oglasit će se zvuk koji predstavlja pogrešku te će igrač znati da je napravio nešto pogrešno.

4. IMPLEMENTACIJA I TESTIRANJE

U daljnjem tekstu detaljno će se objasniti implementacija navedenih igara. Prvo je potrebno definirati sklopovlje kompletne igre tj. navesti i objasniti koje će se elektroničke komponente koristiti za njihov rad. Uz to će se objasniti ispravno spajanje pojedinih komponenta s mikroupravljačem. Nakon što je sklopovlje definirano objasnit će se programska podrška u kojoj će se prikazati kako je mikroupravljač isprogramiran za izvođenje osmišljenih igara.

4.1. Sklopovlje

Na slici 4.1. prikazana je blokovska shema spajanja razvijenih igara i ostalog potrebnog sklopovlja s mikroupravljačem. Sve se igre sastoje od određenih elektroničkih komponenti koje su neovisno spojene na mikroupravljač. Isto tako, pokazatelj preostalog vremena i buzzer sa zvučnim efektima predstavljaju zasebne cjeline koje komuniciraju s mikroupravljačem. Mikroupravljač upravlja svim cjelinama istovremeno. Sve su elektroničke komponente spojene na mikroupravljač pomoću definiranih ulaza/izlaza. Točan popis korištenih elektroničkih komponenata kao i broj potrebnih ulaza/izlaza prikazan je u tablici 4.1..



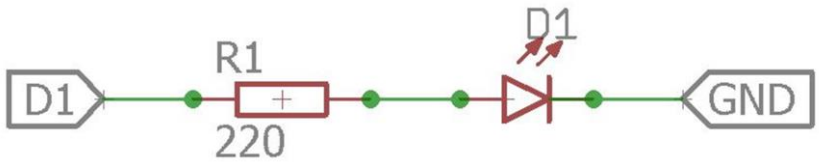
Sl. 4.1. Blokovski prikaz sheme spajanja.

Tab. 4.1. Popis potrebnih elektroničkih komponenti za realizaciju sklopovlja.

Naziv komponente	*Broj potrebnih ulaza/izlaza*	*Količina*	*Ukupan broj ulaza/izlaza za određenu komponentu*
LE dioda	*1*	*13*	*13*
Tipkalo	*1*	*6*	*6*
Potenciometar	*1*	*2*	*2*
Buzzer	*1*	*1*	*1*
7 segmentni LED pokaznik s IC TM1637	*2*	*2*	*4*
Ukupan broj potrebnih ulaza/izlaza			*26*

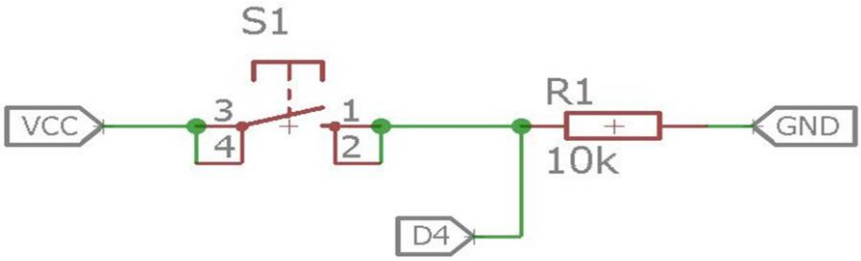
Nakon što su potrebne elektroničke komponente definirane, potrebno ih je ispravno spojiti s mikroupravljačem. Svaka se komponenta spaja s mikroupravljačem na određeni način. U nastavku je detaljnije opisana svaka komponenta i objašnjen način na koji je spojena s mikroupravljačem.

- LE dioda - Za implementaciju su odabrane difuzne (*eng. diffused*) LE diode promjera pet milimetara. Sveukupno ima 13 LE dioda od kojih je osam crvene, dvije plave, dvije zelene i jedna žute boje. Sve se LE diode posebno spajaju na digitalne izlaze mikroupravljača. Zbog razlike u naponu propusne polarizacije diode i izlaznog napona mikroupravljača, svakoj LE diodi se serijski dodaje otpornik kako bi se smanjio napon na određenu vrijednost potrebnu za ispravan rad LE diode. Vrijednosti otpornika koji su spojeni s LE diodama su 220 Ω (za crvenu, zelenu i žutu boju) i 100Ω (za plavu boju). Na slici 4.2. prikazan je ispravan način spajanja LE diode i otpornika s mikroupravljačem.



Sl. 4.2. Shematski prikaz spajanja LE diode i otpornika s mikroupravljačem.

- Tipkalo – Sveukupno je potrebno šest različitih tipkala koja su zasebno spojena na digitalne ulaze mikroupravljača. Odabrana su tipkala veličine 12 milimetara s kapicama različite boje. Kapicama se osigurava lakše razlikovanje tipkala. Svako je tipkalo spojeno s mikroupravljačem preko pull-down otpornika. Razlog tomu je uklanjanje visoke impendancije na ulazu/izlazu mikroupravljača spojenog s tipkalom [8]. Zbog toga je između uzemljenja i ulaza u mikroupravljač potrebno dodati otpornik vrijednosti 10kΩ (Sl. 4.3.). Time se osigurava ispravan rad mikroupravljača te precizno očitavanje trenutnog stanja tipkala.



Sl. 4.3. Shematski prikaz spajanja tipkala i otpornika s mikroupravljačem.

- Potenciometar – Elektronička komponenta koja je idealna za unos većeg raspona stanja je potenciometar. Okrečući klizač potenciometra, mijenja se vrijednost otpora samog potenciometra. Ako se potenciometar spoji na analogni ulaz mikroupravljača, može se regulirati napon koji mikroupravljač očitava s navedenog ulaza. Pomoću toga mikroupravljač se može programirati kako bi prepoznao točan položaj zaokreta klizača u potenciometru. Time je igraču omogućena veća sloboda interakcije s igrama. Postoji mnogo različitih vrsta potenciometara. Za potrebe završnog rada odabrana su dva linearna jednookretajna THT (*eng. Through-hole technology*) potenciometra vrijednosti 10kΩ.
- 7 segmentni LED pokaznik s IC TM1637 - Za prikaz preostalog vremena i prikaz brojeva na pokazniku igre „Pronađi broj“ odabran je modul koji sadrži četiri 7 segmentna LED pokaznika spojenih na TM1637 integrirani sklop (Sl. 4.4.). TM1637 je sklop koji je namjenjen upravljanju 7 segmentnim LED pokaznicima. On olakšava prikazivanje i kontroliranje LED pokaznika te ujedno smanjuje broj potrebnih priključaka za korištenje. Modul sadrži četiri nožice: VCC, GND, CLK i DIO. VCC služi za spajanje na izvor napajanja, a GND za spajanje na uzemljenje. CLK je ulaz kojim integrirani sklop dobiva signal takta, dok DIO služi za prijenos podataka između integriranog sklopa i mikroupravljača. Navedeni integrirani sklop u Arduino programskom okruženju sadrži vlastitu biblioteku koja omogućuje jednostavnije korištenje sklopa.



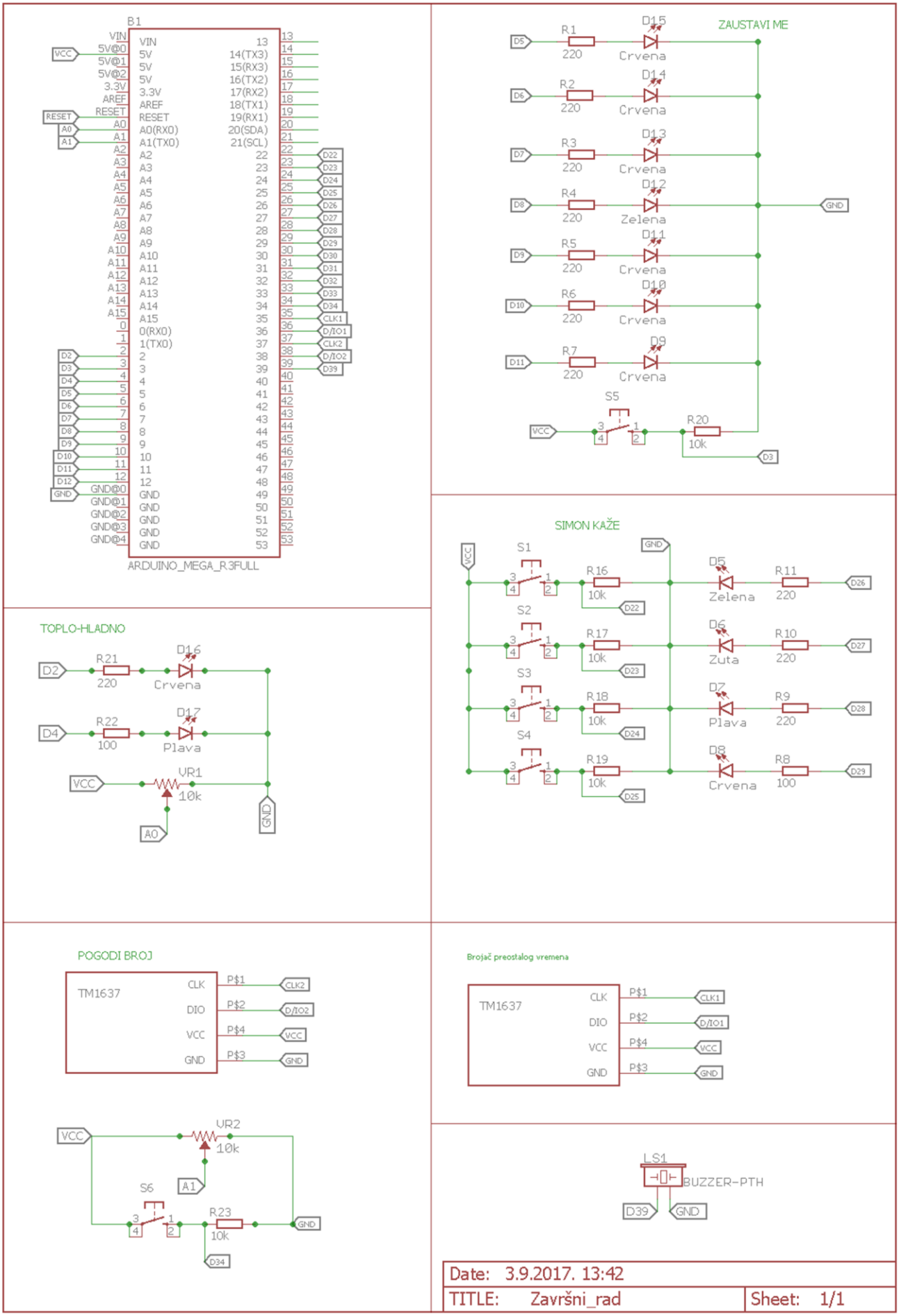
Sl. 4.4. Prikaz TM1637 7 segmentnog modula.

- Buzzer - Za emitiranje zvuka korišten je buzzer (Sl. 4.5.). Buzzer ili beeper je zvučni uređaj koji pretvara električnu energiju u zvučne valove. Može biti mehanički, elektromehanički ili piezoelektrični [9]. Za potrebe završnog rada odabran je piezoelektični buzzer. On u sebi sadrži malu zavojnicu i magnet. Kada kroz zavojnicu prođe struja, ona se magnetizira i povlači prema magnetu. Tim mehaničkim pokretom moguće je reproducirati zvuk različitih frekvencija i jačina. Pomoću mikroupravljača moguće je generirati točne tonove te naposlijetku i melodije. Arduino programsko okruženje sadrži posebnu biblioteku namijenjenu za buzzer kojom se olakšava korištenje same komponente.



Sl. 4.5. Prikaz Piezoelektričnog buzzera.

Na slici 4.6. prikazana je kompletna elektronička shema sklopovlja. Shema je izrađena u Cadsoft Eagle programu. Na shemi je vidljivo koji su ulazi/izlazi mikroupravljača korišteni.



Sl. 4.6. Shematski prikaz sklopovlja izrađen u Cadsoft Eagle programu.

4.2. Programska podrška

Programska podrška je razvijena u C++ programskom jeziku upotrebom klasa i objekata. Kod je pisan u Arduino IDE programskom okruženju. Svaka igra, pokazivač preostalog vremena i buzzer sadrže svoju posebnu klasu. Unutar klasa nalaze se varijable i metode koje su potrebne za ispravan rad svih komponenti. Objekti svih navedenih klasa su instancirani prije izvođenja glavnog dijela programa.

Korištena je biblioteka pod nazivom *TM1637Display.h* koja olakšava korištenje 7 segmentnog LED pokaznika s IC TM1637. Pomoću nje možemo kontrolirati kompletan prikaz, pojedinačne znamenke i jačinu osvjetljenja na LED pokazniku.

Tipkalo u igri „Zaustavi me“ je spojeno na ulaz koji može prepoznati vanjski prekid (*eng externall interrupt*). Kada igrač pritisne tipkalo, neovisno koji dio programskog koda se izvodi, mikroupravljač će uvijek prepoznati pritisak i izvesti pripadnu funkciju kojom provjerava koja je LE dioda u tom trenutku bila osvijetljena.

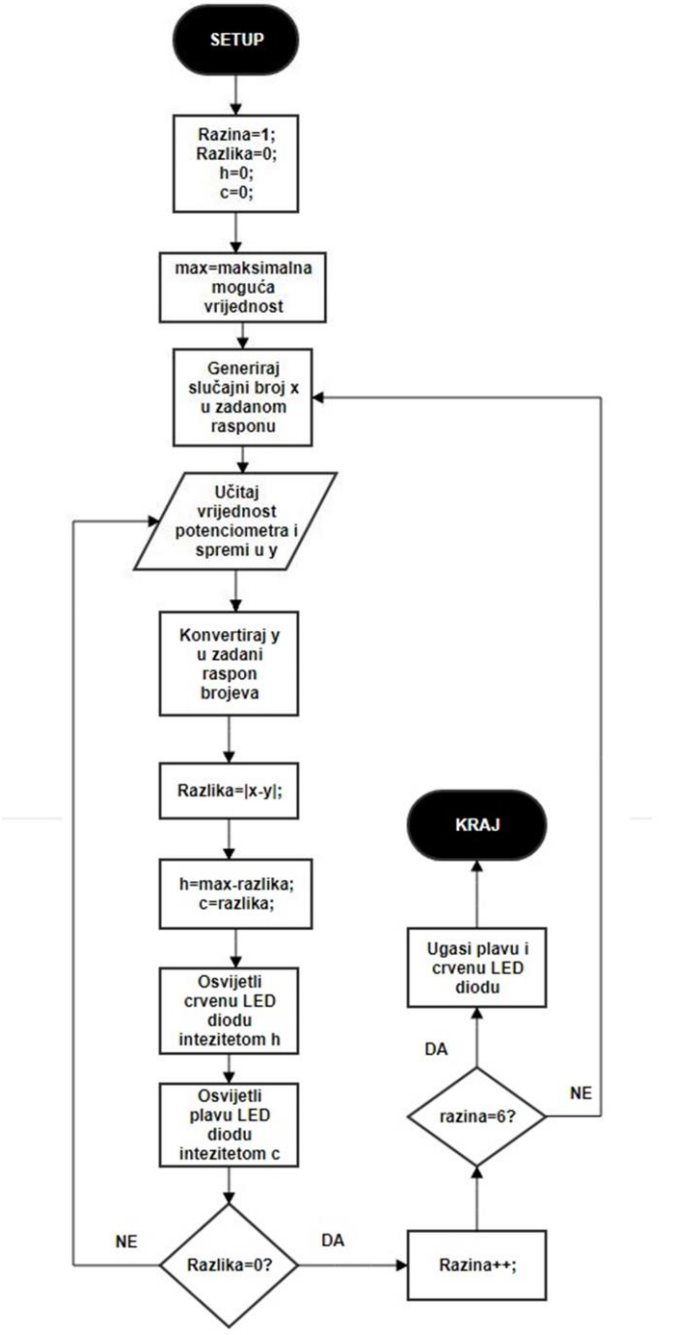
Programski kod se izvodi sekvencijalno, ali igrač ima privid da se svaka igra paralelno izvodi. Cijeli programski kod ne sadrži ni jedan poziv na *delay()* funkciju ([10]) jer ona zaustavlja izvođenje programskog koda na određeno vrijeme i time onemogućuje izvođenje metoda klasa koji zahtjevaju različita vremenska trajanja pojedinih dijelova koda. Umjesto toga, za mjerenje vremena korištena je *millis()* funkcija koja ne zaustavlja izvođenje programskog koda nego vraća broj koliko je milisekundi prošlo od pokretanja mikroupravljača. Time se ostvaruje privid paralelnog izvođenja programskog koda.

Svako generiranje slučajnih brojeva realizirano je pomoću *random()* funkcije. Ona koristi funkciju *randomseed()* koja inicijalizira pseudoslučajni broj koji se nalazi u nizu brojeva koji je uvijek isti. Nakon svakog pozivanja *random()* funkcije, funkcija vrati sljedeći broj koji se nalazi u tom nizu. Parametar koji je predan *randomseed()* funkciji je očitanje analognog ulaza u koji je spojen potenciometar igre „Toplo-Hladno“. Time se osigurava da se prilikom svakog pokretanja mikroupravljača generira drugačiji slučajni broj.

4.2.1. Program igre „Toplo-Hladno“

Na slici 4.7. prikazan je dijagram toka igre „Toplo-Hladno“. Pri pokretanju igre prvo se generiraju pomoćne varijable i slučajni broj *x* kojeg je potrebno pronaći okretajući klizač potenciometra. Slučajni broj *x* je generiran u zadanom rasponu brojeva kako bi se olakšalo kasnije rukovanje tom varijablom. Zatim se očitava vrijednost potenciometra i sprema u varijablu *y*. Varijabla *y* se konvertira u zadani raspon brojeva isto kao i varijabla *x*. Koristeći

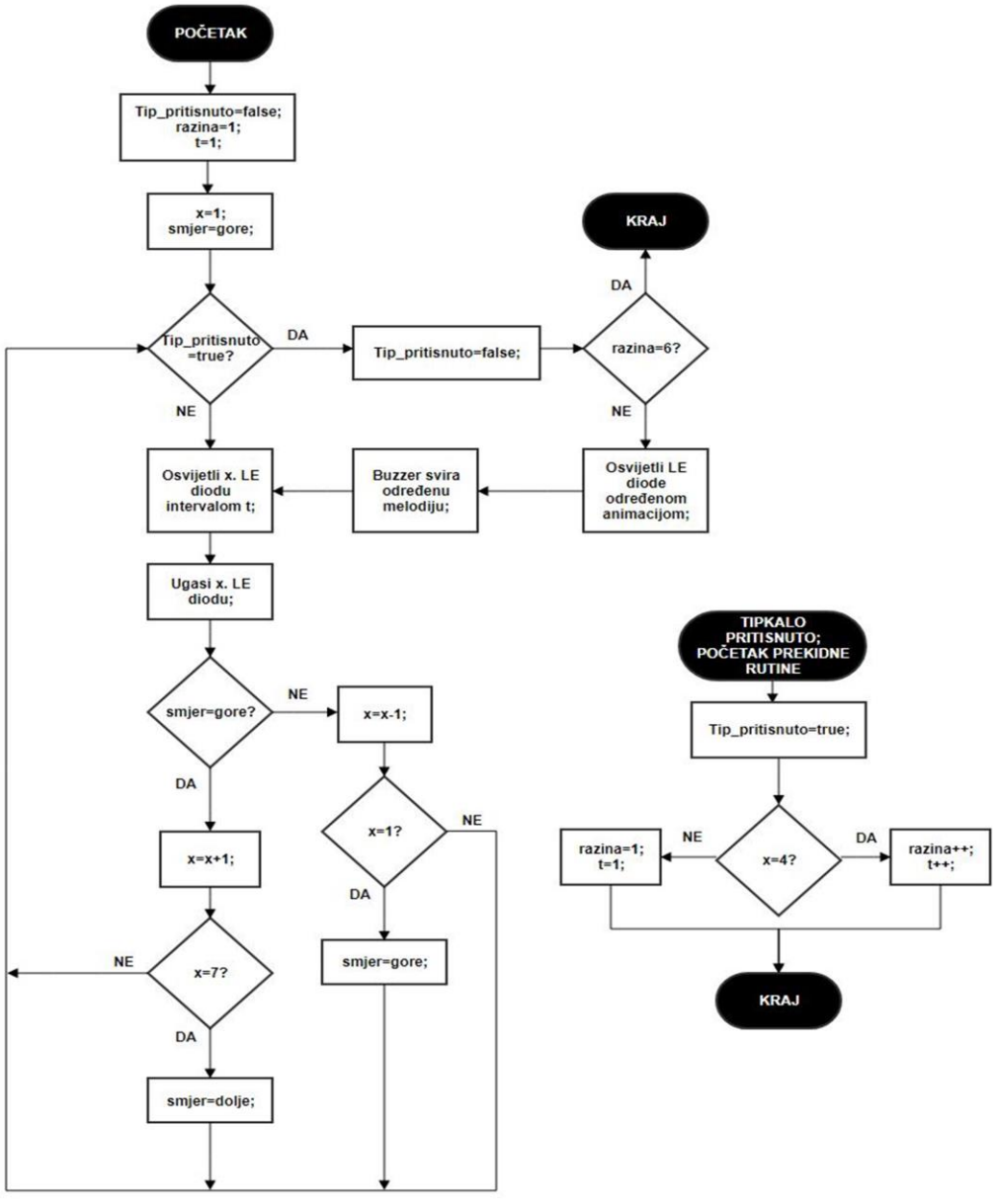
apsolutnu razliku *x* i *y* varijable, moguće je utvrditi koliko je odstupanje vrijednosti potenciometra od tražene vrijednosti *x*. Ako je razlika jednaka nuli, točan položaj klizača potenciometra je pronađen te se prelazi na sljedeću razinu tj. ponovo se generira slučajni broj koji je potrebno pronaći okretanjem klizača. Crvena i plava LE dioda se osvjetljuju različitim intezitetom kako bi igraču signalizirale kojim smjerom je potrebno okretati klizač potenciometra da se pronađe traženi položaj. Igra je završena nakon što se točno pronađe peti položaj klizača potenciometra.



Sl. 4.7. Prikaz dijagrama toka „Toplo-Hladno“ igre.

4.2.2. Program igre „Zaustavi me“

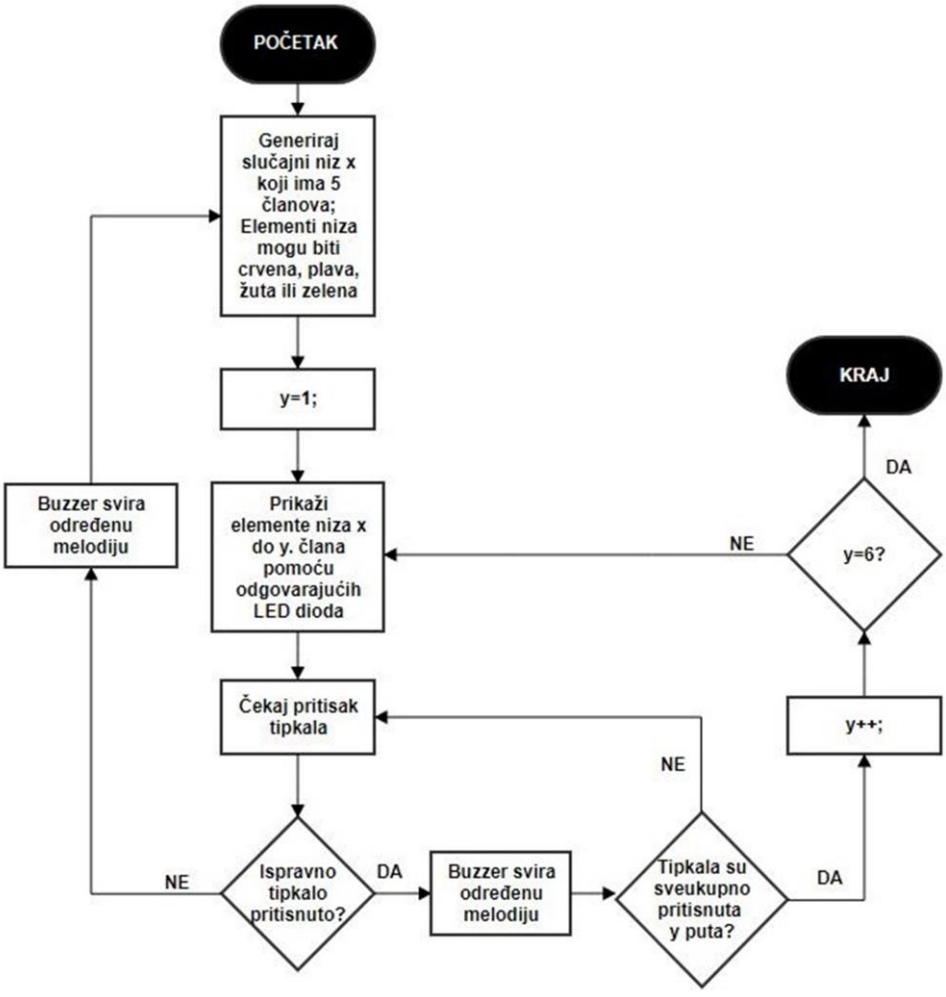
Na slici 4.8. prikazan je dijagram toka igre „Zaustavi me“. Pri pokretanju igre LE diode se sekvencijalno osvjetljavaju sve dok se tipkalo ne pritisne. Način na koji LE diode svijetle je razrađen tako da prvo svijetli prva LE dioda, zatim druga i tako dalje sve do zadnje sedme diode nakon čega svijetle ponovo u suprotnom smjeru do prve diode. Kada je tipkalo pritisnuto, aktivira se prekidna rutina koja provjerava je li četvrta LE dioda (ona je jedina zelene boje) u tom trenutku svijetlila. Ako je svijetlila četvrta LE dioda, igrač prelazi na sljedeću razinu. Ako je svijetlila bilo koja druga LE dioda, igra se ponovno pokreće od prve razine. Nakon što se uspješno prijeđe peta razina, igra je uspješno riješena te prestaje s radom.



Sl. 4.8. Prikaz dijagrama toka „Zaustavi me“ igre.

4.2.3. Program igre „Simon“

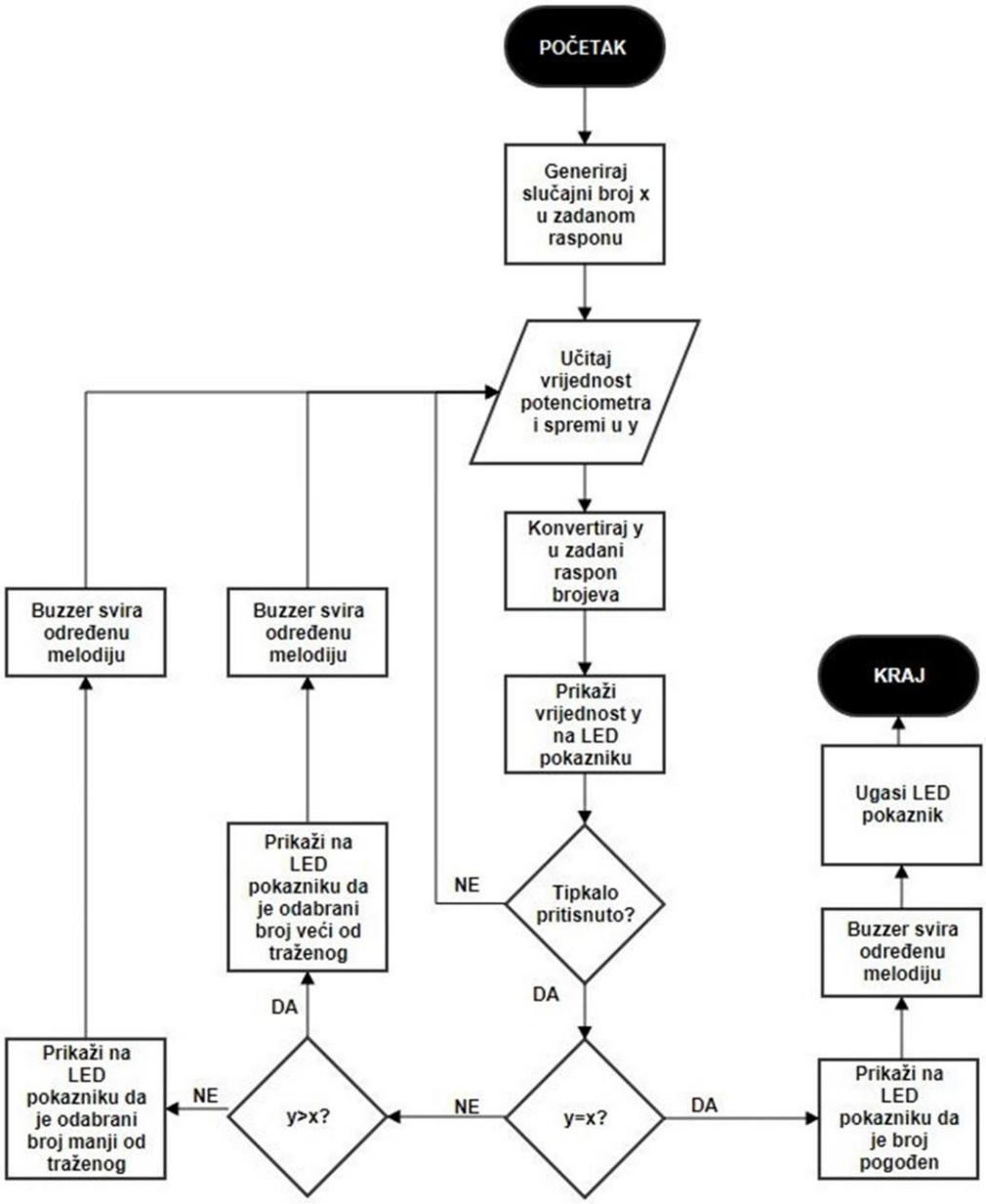
Na slici 4.9. prikazan je dijagram toka igre „Simon“. Prilikom pokretanja igre, generiran je nasumični niz od pet elemenata. Svaki element u nizu sadrži jednu od četiri navedene boje. Igra započinje osvjetljavanjem jedne od LE dioda koja predstavlja određenu boju prvog elementa u nizu. Igra čeka pritisak tipkala koji odgovara prikazanoj boji. Ako je pritisnuto traženo tipkalo, igrač prelazi na sljedeću razinu. Sada se osvjetljava ista LE dioda, no nakon nje i druga koja predstavlja drugi element u nizu. Tada je potrebno pritisnuti dva tipkala odgovarajućim redosljedom kojim su se LE diode osvjetljavale. Nakon svakog pritiska tipkala provjerava se je li odabrana ispravna boja. Ako je unesen ispravan slijed tipkala, igra ponavlja dosadašnji niz, a zatim prikazuje sljedeću boju u nizu. Ovisno koliko je boja prikazano, toliko je potrebno puta odabrati određenu boju tipkalom. Ako se pritisne pogrešna tipka koja ne odgovara traženoj boji, igra počinje ispočetka te se generira novi nasumični niz od pet elemenata. Igra završava nakon što se ispravno ponovi slijed od pet prikazanih boja.



Sl. 4.9. Prikaz dijagrama toka „Simon“ igre.

4.2.4. Program igre „Pronađi broj“

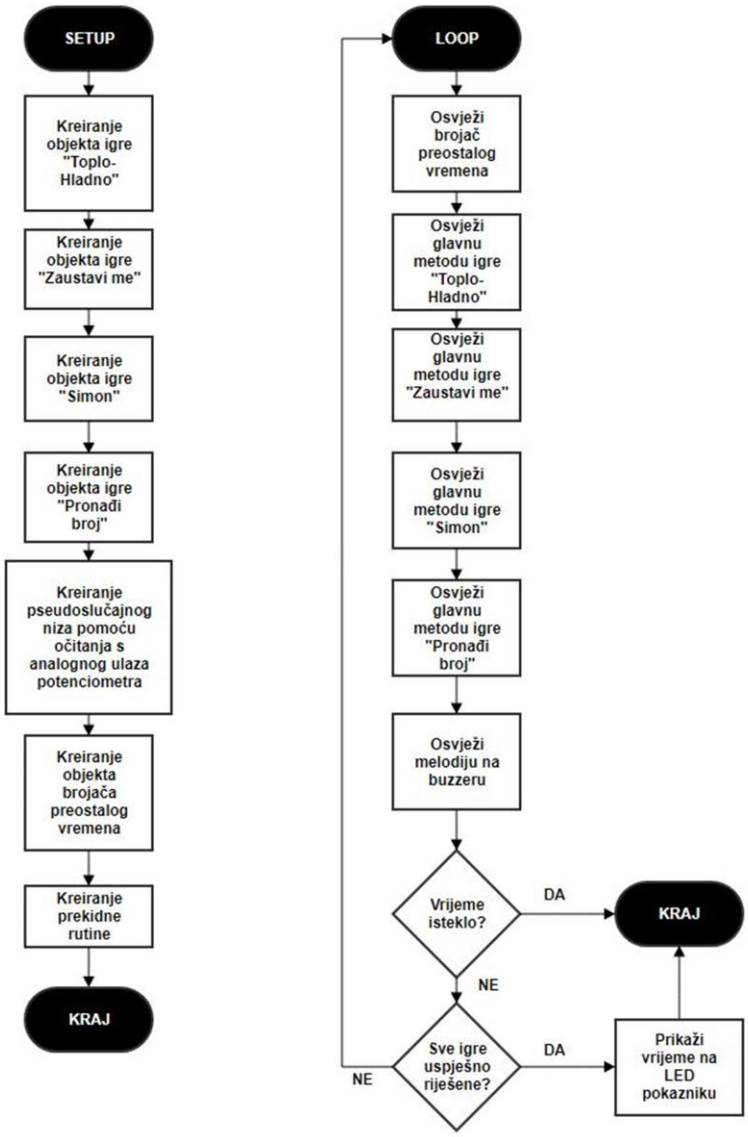
Na slici 4.10. prikazan je dijagram toka igre „Pronađi broj“. Prilikom pokretanja igre generira se slučajni broj *x* u rasponu od 0 do 50. Zatim se očitava vrijednost klizača na potenciometru i sprema u varijablu *y*. Varijabla *y* se konvertira u raspon od 0 do 50 i prikazuje na LED pokazniku. Program čeka pritisak tipkala kojim se odabire broj *y* i uspoređuje s varijablom *x*. Ako su *x* i *y* jednaki prikazuje se odgovarajuća poruka na LED pokazniku i igra završava. Ako odabrani broj *y* nije jednak kao *x*, provjerava se je li *y* veći ili manji od *x* i prikazuje se odgovarajuća poruka na LED pokazniku i ponovo se očitava vrijednost otpornika.



Sl. 4.10. Prikaz dijagrama toka „Pronađi broj“ igre.

4.3. „Setup()“ i „Loop()“ programski kod

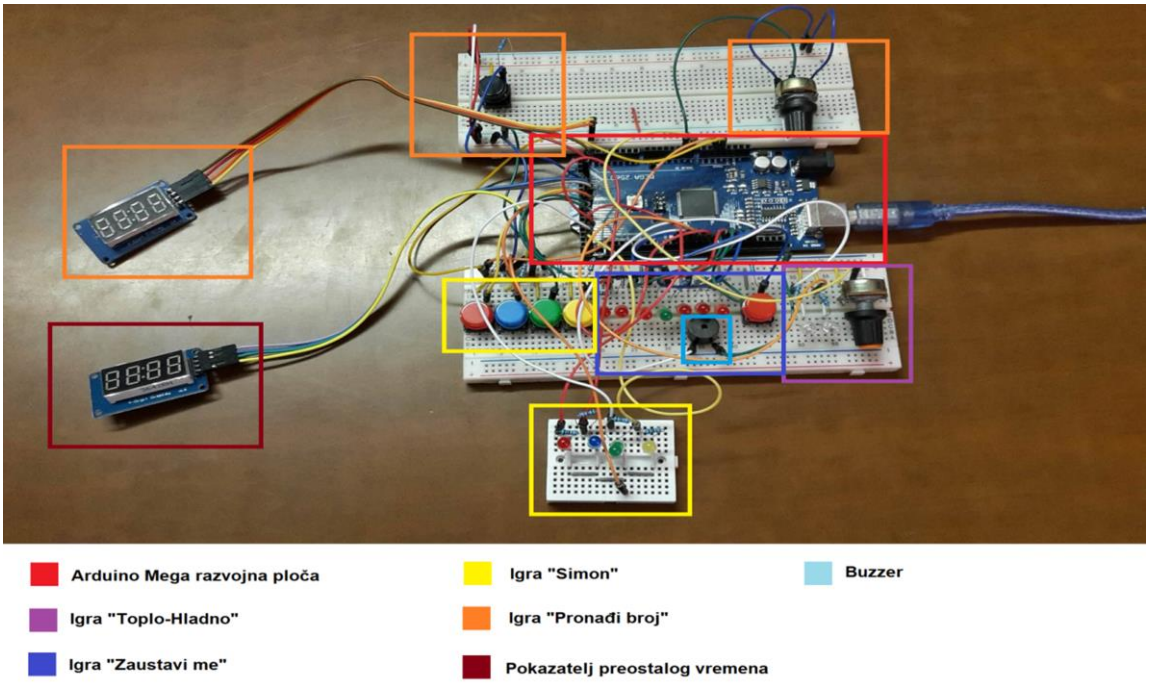
Na slici 4.11. prikazan je dijagram toka *Setup()* i *Loop()* dijela. Prilikom pokretanja mikroupravljača, s*etup()* funkcija se izvede samo jednom, nakon čega se *Loop()* funkcija izvodi beskonačno u petlji. U *Setup()* funkciji su instancirani objekti svih klasa igara i brojača preostalog vremena. Uz to kreirana je prekidna rutina i pseudoslučajni niz koji je potreban za generiranje pseudoslučajnih brojeva u ostatku programskog koda. U *Loop()* funkciji se osvježavaju sve glavne metode igara, melodija na buzzeru i brojača preostalog vremena koji izvode određene zadatke sa navedenim ulazima/izlazima. Nakon toga se provjerava stanje brojača i ako je vrijeme isteklo prekida se izvođenje ostatka programskog koda. Ako vrijeme brojača nije isteklo provjeravaju se stanja svih igara te ako su sve uspješno riješene, zapisuje se vrijeme u trenutku kada je zadnja igra riješena i prekida se izvođenje ostatka programskog koda.



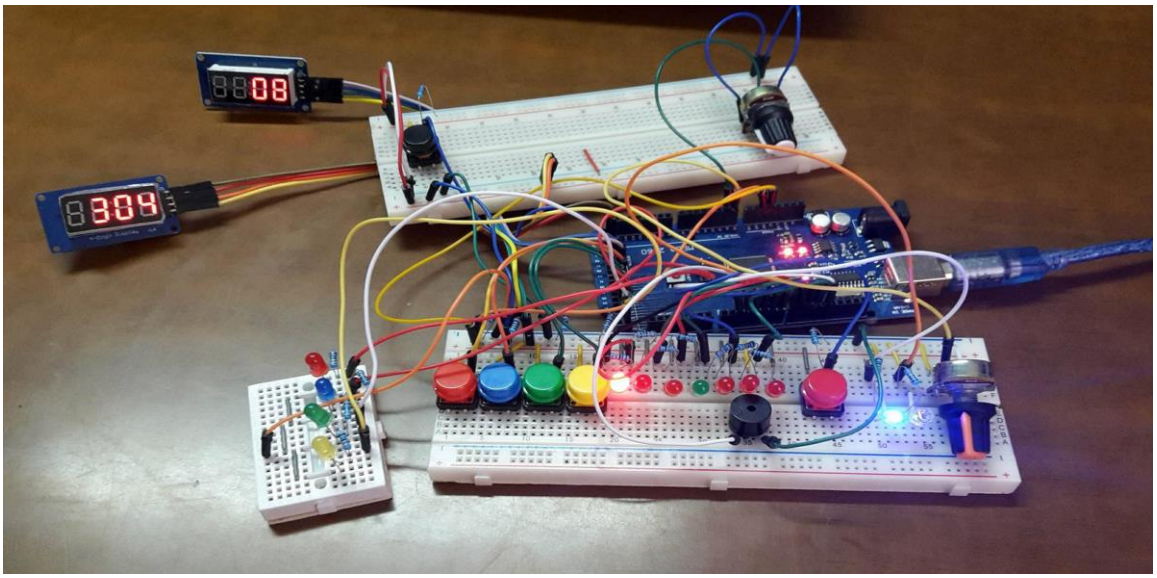
Sl. 4.11. Prikaz dijagrama toka Setup() i Loop() funkcije.

4.4. Testiranje

Testiranje je izvedeno spajanjem komponenti na eksperimentalnu pločicu. Prije samog spajanja, provjerena je svaka komponenta i potvrđena njezina ispravnost. Zatim su sve komponente spojene s mikroupravljačem prema navedenim uputama. Slika 4.12. prikazuje fizički izgled testiranja završnog rada. Označeni su različiti dijelovi igara koji nisu spojeni posebnim redosljedom. Nakon dodavanja programske podrške u mikroupravljač, testiran je svaki dio igre te potvrđen njen ispravan rad (Sl. 4.13.)



Sl. 4.12. Fizički izgled sklopovlja na eksperimentalnoj pločici.



Sl. 4.13. Prikaz rada sklopovlja na eksperimentalnoj pločici.

5. ZAKLJUČAK

Zadatak ovog završnog rada bio je osmisliti i izraditi interaktivnu igru na Arduino platformi s najmanje dva načina rada. Uz to je bilo potrebno sklopovski izraditi samu igru i pripadajuću programsku podršku kako bi igra ispravno radila. Osmišljena je i izrađena igra koja se sastoji od četiri jednostavnije igre koje je potrebno u zadanom vremenskom roku uspješno riješiti. U teorijskom dijelu rada objašnjeno je što je mikroupravljač, kako se programira, što je Arduino razvojna ploča te je na primjeru Intel 8051 mikroupravljača objašnjena unutrašnja struktura mikroupravljača. U praktičnom dijelu objašnjene su manje, jednostavnije igre te je detaljno razrađen način njihova rada. Nakon toga opisano je sklopovlje same igre i programska podrška kojom je mikroupravljač programiran. Sklopovlje se sastoji od elektroničkih komponenti spojenih na Arduino Mega razvojnu ploču. Korištene su osnovne elektroničke komponente koje se jednostavno povezuju s mikroupravljačem. Za izradu programske podrške potrebno je naprednije znanje C++ programskog jezika zbog korištenja klasa i objekata kojima su svi programski dijelovi objedinjeni u jednu cjelinu. Kompletna igra je testirana na eksperimentalnoj pločici te je potvrđen njen ispravan rad.

Jednostavnije su igre osmišljene s ciljem poticanja i razvoja kognitivnih sposobnosti kod igrača. Zbog toga je igra namjenjena primarno djeci i osnovnoškolcima. Odabrane su igre koje zahtjevaju od igrača različite intelektualne sposobnosti, no zbog modularnog pristupa izrade igara, moguće je dodati ili promjeniti postojeće igre s novim, drugačijim igrama.

LITERATURA

[1] Mikrokontroler, https://sh.wikipedia.org/wiki/Mikrokontroler, 25.8.2017

[2] L. Budin, Mikroračunala i mikroupravljači, Element, Zagreb, 1999

[3] Uvod u mikrokontrolere, https://www.sites.google.com/site/robotickragujevac034/mikrokontroleri/predavanja/01-uvod-u-mikrokontrolere, 25.8.2017

[4] M. Verle, Arhitektura i programiranje 8051 Mikrokontrolera, MikroElektronika, 2007

[5] Arduino, https://hr.wikipedia.org/wiki/Arduino, 26.8.2017

[6] Arduino, https://sr.wikipedia.org/wiki/Arduino, 26.8.2017

[7] Robert Sedgewick, Algorithms in C++, Addison-Wesley, 2009

[8] Pull-up/ Pull-down otpornik, https://e-radionica.com/hr/blog/2015/08/19/pull-up-pull-down-otpornik/, 29.8.2017

[9] Buzzer, https://e-radionica.com/hr/blog/2015/08/19/buzzer/, 30.8.2017

[10] Delay(), https://www.arduino.cc/en/Reference/Delay, 2.9.2017.

SAŽETAK

Cilj ovog završnog rada je bio osmisliti interaktivnu igru na Arduino platformi s najmanje dva načina rada. Uz to je bilo potrebno sklopovski izraditi igru pomoću raznih elektroničkih komponenti spojenih na Arduino platformu. Osmišljena je i izrađena igra koja se sastoji od četiri jednostavnije igre koje je potrebno u zadanom vremenskom roku uspješno riješiti. U teorijskom dijelu rada objašnjeno je što je mikroupravljač, kako se programira, što je Arduino razvojna ploča te je na primjeru Intel 8051 mikroupravljača objašnjena unutrašnja struktura mikroupravljača. U praktičnom dijelu objašnjene su osmišljene jednostavnije igre te je detaljno razrađen način njihova rada. Sklopovlje se sastoji od elektroničkih komponenti spojenih na Arduino Mega razvojnu ploču. Programska podrška je razvijena u C++ programskom jeziku upotrebom klasa i objekata. Kod je pisan u Arduino IDE programskom okruženju.

Ključne riječi: interaktivna igra, Arduino platforma, jednostavnije igre, mikroupravljač, Arduino Mega, C++, Arduino IDE

ABSTRACT

Title: Interactive game on Arduino platform

The goal of this final thesis was to design an interactive game on the Arduino platform with at least two modes of operation. In addition, it was necessary to build the game with various electronic components connected to the Arduino platform. The game was designed which consists of four simpler games that are needed to be solved within a given time. In the theoretical part of the work it is explained what the microcontroller is, how it is programmed, what is the Arduino development board and the example of Intel 8051 microcontroller explains the internal structure of the microcontroller. In the practical part, the simpler games were explained and the way of their work was elaborated in detail. The hardware consists of electronic components connected to the Arduino Mega Development Board. Software is developed in the C ++ programming language using classes and objects. The code is written in the Arduino IDE program environment.

Keywords: interactive game. Arduino platform, simpler games, microcontroller, Arduino Mega, C++, Arduino IDE

ŽIVOTOPIS

Domagoj Špoljar rođen je 26.3.1996. u Đakovu. Završio je opću gimnaziju A. G. Matoša u Đakovu s vrlo dobrim uspjehom. Nakon srednje škole upisuje Elektrotehnički fakultet u Osijeku, smjer Računarstvo. Posjeduje znanja i iskustva sa C, C++, C# i HTML programskim jezicima. Kroz dosadašnje obrazovanje, stekao je znanja u računalnim programima Matlab, AutoCAD, Cadsoft Eagle i Photoshop. Aktivno se služi engleskim jezikom u govoru i pismu.

PRILOZI

Priložen je CD s:

- programskim kodom izrađenim u Arduino IDE programskom okruženju,
- video isječkom ispravnog rada sklopovlja,
- dodatnim slikama izgleda sklopovlja,
- kompletne elektroničke sheme sklopovlja izrađenog u Cadsoft Eagle programu i
- svim slikama korištenim u završnom radu.