



LED pad

Tehnička dokumentacija



Autorica: Marina Vlahović

OŠ Malešnica
Zagreb

Sadržaj

1. Uvod.....	2
1.1. O autorici.....	2
1.2. Što je LED pad?.....	2
1.3. Ideja projekta	3
1.4. Razvoj ideje projekta	4
2. Korisnička dokumentacija	6
2.1. Korisničko sučelje	6
2.2. Uključivanje i značajke gumba za postavke	6
2.2.1. Gumb AD	7
2.2.2. Gumb TIME.....	8
2.2.3. Gumb GAME.....	9
2.2.4. Gumb START.....	9
2.3. Opis igara	10
2.3.1. Igra 1 – Razgibaj se	10
2.3.2. Igra 2 – Provjeri reflekse.....	10
2.3.3. Igra 3 – Vježba za jednu ruku.....	11
2.4. Baza pohranjenih rezultata	12
3. Tehnička dokumentacija	13
3.1. Hardverski dio	13
3.2. Softverski dio.....	15
3.2.1. Postavka vremena trajanja igre	15
3.2.2. Odabir igre.....	16
3.2.2. Prikaz reklamnog teksta	16
3.2.3. Igra 1	17
3.2.4. Igra 2	17
3.2.5. Igra 3	18
3.2.5. Spremanje rezultata u bazu.....	18
3.2.6. Rubni slučajevi.....	19
4. Mogućnosti primjene projekta	20

1. Uvod

1.1. O autorici

Ja sam Marina Vlahović iz Osnovne škole Malešnica u Zagrebu i imam 14 godina. Programiranjem se bavim od 2018. god. Jako me zanimaju robotika i IoT tehnologija. Jedne godine, moja se škola odlučila prijaviti na donacijski natječaj Generacija Now kada sam prvi put upoznala IoT tehnologiju. Arduino mi se jako svidio i volim programirati u njegovom razvojnom okruženju. Pritom sam naučila programirati i u C++-u i C-u.

Ovo je samo jedan od mojih radova s IoT tehnologijom, ali ujedno i moj najdraži rad pa sam ga odlučila prijaviti ove godine i na Razvoj softvera. Inače redovito sudjelujem na raznim izložbama za inovatore i makere. Bavim se još i sviranjem klavira i atletikom, ali izrada projekata mi je jedan od najdražih hobija i vrlo rado ih izrađujem. 😊

1.2. Što je LED pad?

Razvoj reakcije, fokusa i koncentracije te razgibavanje i vježbanje važni su čimbenici u razvoju djece, ali i ključne odlike vrhunskih sportaša te spadaju i u poželjne svakodnevne aktivnosti starijih osoba.

LED pad je zabavna igra koja **pomaže u razgibavanju i razvijanju refleksa.**

Izrađena je u obliku okomite interaktivne ploče na kojoj se nalazi 30 gumba i 30 pripadajućih LED dioda.



Slika 1 Interaktivna ploča igre LED pad

LED diode se nasumično pale te ih je potrebno što prije ugasiti pritiskom na pripadajući gumb. Na kraju svake igre prikazuje se broj uspješno ugašenih LED dioda u zadanom vremenu, a nakon toga i pohranjeni najbolji rezultat (high score).

To je samo jedna od opcija ove igre. Više o mogućnostima igre i budućem razvoju pročitajte u sljedećim poglavljima.

Igrom upravlja mikrokontroler **Arduino Mega 2560** za koji sam napisala program u Arduinovom integriranom razvojnom okruženju koristeći funkcije iz C i C ++ programskog jezika.

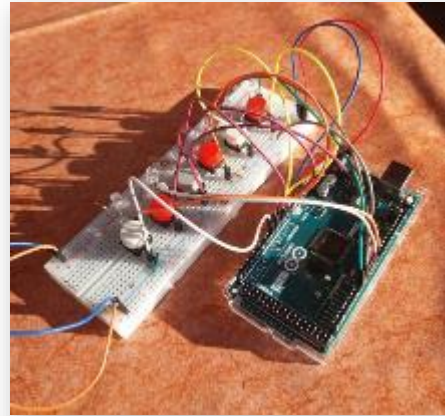
1.3. Ideja projekta

Već se duže vrijeme bavim izradom projekata s Arduino mikrokontrolerima i općenito s IoT tehnologijom. Jedan dan mi se baka požalila kako ju jako bole ramena i da joj je liječnik rekao da treba vježbati. I tako mi je pala na pamet ideja – mogla bih baki napraviti **uređaj za razgibavanje**, ali da pritom bude i u obliku igre kako bi se mogla i zabaviti dok svakodnevno vježba sa svojim prijateljicama.

Nakon izrade prototipa projekta uočila sam da ovaj uređaj mogu koristiti i djeca za **razvoj svojih motoričkih sposobnosti** kao i sportaši. Tada sam odlučila ovaj projekt prezentirati na izložbama i natjecanjima kako bi se što više ljudi upoznalo s njim.

1.4. Razvoj ideje projekta

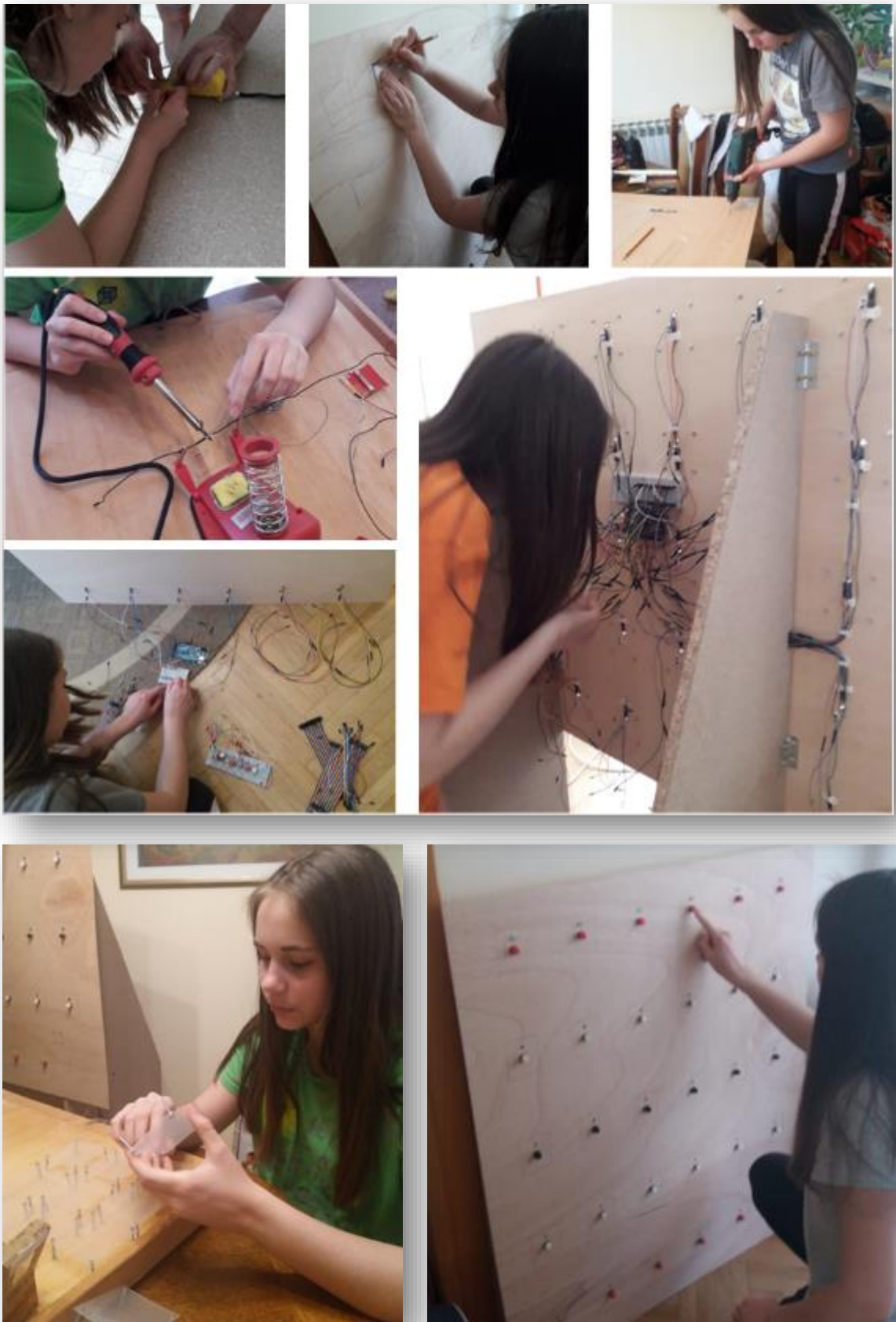
Svoju ideju projekta igre za razgibavanje najprije sam skicirala na papir, a nakon toga sam izradila umanjenu verziju igre – da vidim radi li to i sviđa li se i baki ideja. Ta umanjena verzija se sastojala od nekoliko gumba i LED-ica na eksperimentalnoj pločici. Baki se svidjela ideja pa sam odlučila napraviti i veću verziju – trenutnu s kojom sam se prijavila na natjecanje.



Slika 2 Probna verzija igre

Tijek rada:

- ▶ Crtanje tehničkog crteža svih pozicija konstrukcije
- ▶ Ocrtavanje mjesta za bušenje na šperploči
- ▶ Bušenje rupa na šperploči
- ▶ Piljenje iverica i sastavljanje konstrukcije
- ▶ Lemljenje žica, svjetlećih dioda i tipki, omotavanje izolir – trakom i sastavljanje na šperploču
- ▶ Rezanje i bušenje rupa na plastici
- ▶ Stavljanje i pričvršćivanje plastike
- ▶ Pisanje programa u programskom sučelju Arduino IDE
- ▶ Testiranje i popravci
- ▶ Finaliziranje projekta u reprezentativnu verziju



Slika 3 Tijek izrade interaktivne ploče

2. Korisnička dokumentacija

2.1. Korisničko sučelje

Uređaj je spoj hardvera i softvera. Izrađen je u obliku okomite interaktivne ploče na kojoj se nalazi 30 gumba i 30 pripadajućih LED dioda kojima upravlja softver u mikrokontroleru Arduino Mega 2560.

Označeni su gumbi za odabir vremena igre, odabir vrste igre, pokretanje igre te prikaz reklamnog teksta.

Sučelje je u potpunosti prilagođeno korisnicima i jednostavno je za korištenje.

2.2. Uključivanje i značajke gumba za postavke

Kada se igra uključi na električni napon, prvo se izvede početna animacija. Ta animacija se prikazuje dok se ne pritisne neki od gumba za postavke.

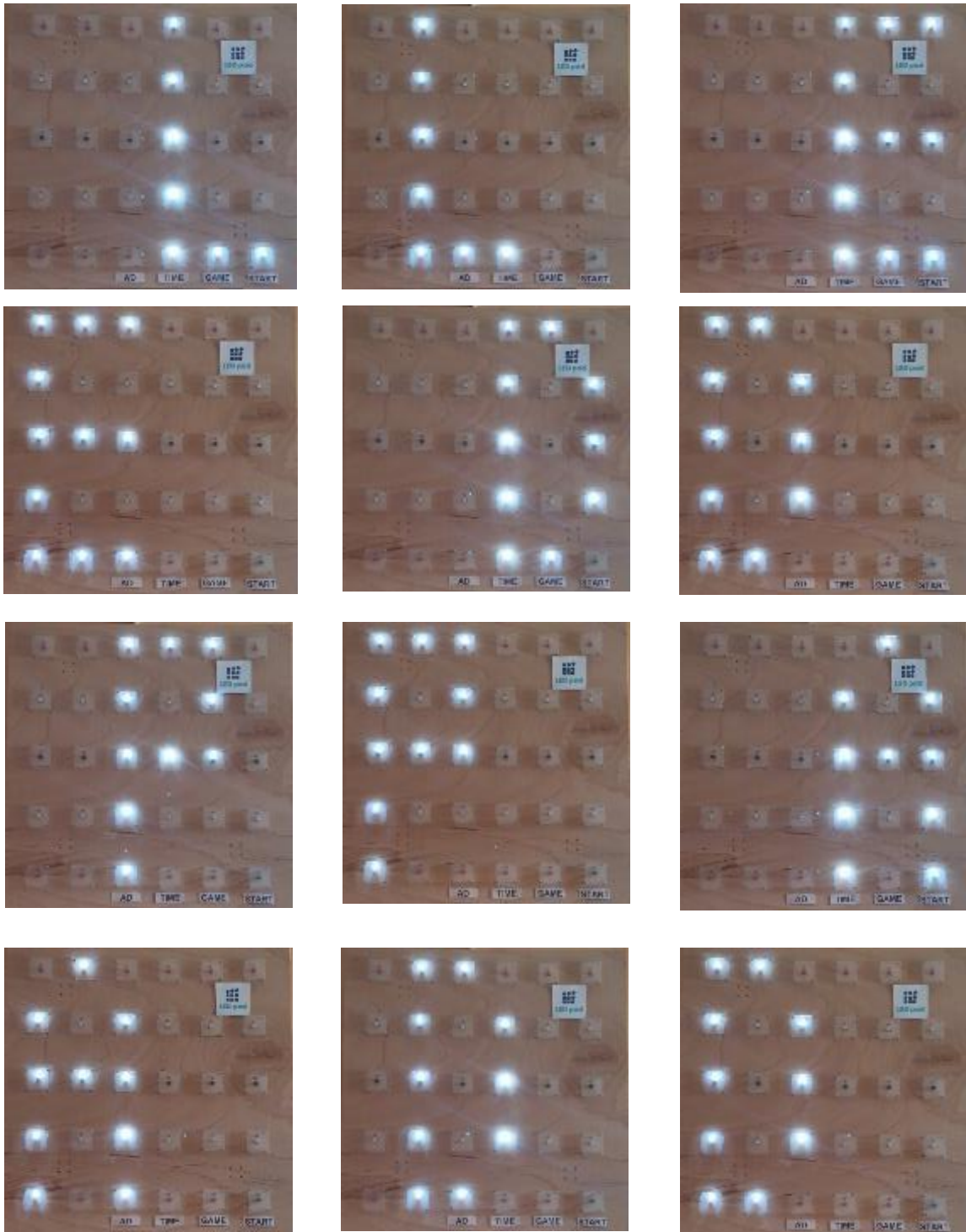


Slika 4 Gumbi za postavke

- gumb AD – prikazuje reklamni tekst tehnologijom „trčeće svjetlo“ sa svjetlećim diodama (LED).
- gumb TIME – služi za odabir vremena trajanja pojedine igre
- gumb GAME – služi za izbor igre
- gumb START – pokreće izabranu igru

2.2.1. Gumb AD

Pritiskom na gumb AD, trčecim slovima ispisuju se riječi LED i PAD jedna za drugom s desna na lijevo u razmaku od jedne sekunde. U budućnosti bi se umjesto riječi LED PAD ispisivao reklamni tekst na taj način.



Slika 5 Prikaz „trčecih svjetala“ za ispis reklamnog teksta

QR CODE

-za video prikaz ispisa reklamnog teksta



2.2.2. Gumb TIME

Pritiskom na gumb TIME određuje se vrijeme trajanja igre koje igrač želi u tom trenutku. Igrač može odabrati između 10, 20 i 30 sekundi. Svakim pritiskom na gumb vrijeme igre se povećava za 10 sekundi, a kada pređe 30, vraća se na 10 sekundi (30 sekundi je trenutni maksimum trajanja igre). Nakon što se igrač odlučio za vrijeme igre, pritiskom na gumb GAME potvrđuje to vrijeme i više ga ne može promijeniti dokle god ne završi igra.



Slika 6 Prikaz vrijednosti izabranog vremena igre

QR CODE

-za video prikaz ispisa postavljanja vremena za igru



2.2.3. Gumb GAME

Pritiskom na gumb GAME određuje se vrsta igre koju korisnik želi igrati. Postoji mogućnost odabira između 3 igre. Svakim pritiskom na gumb vrsta igre povećava se za 1, a kada pređe 3, vraća se na 1. Više o svakoj igri slijedi u nastavku. Pritiskom na gumb START, odabrana vrsta igre se potvrđuje i ne može se promijeniti dokle god igra ne završi.



Slika 7 Prikaz broja odabrane igre

QR CODE

-za video prikaz izbora broja igre



2.2.4. Gumb START

Pritiskom na gumb START odabrana igra s izabranim vremenom započinje. Prvo se izvodi animacija za početak igre. Nakon 500 milisekundi počinje igra i odbrojanje zadanog vremena. Igra završava nakon što odabrano vrijeme istekne.

2.3. Opis igara

2.3.1. Igra 1 – Razgibaj se

Prva igra, Razgibaj se, radi na način da nakon što počne igra, random se bira LEDica koja će se upaliti (može biti bilo koja s ploče) te je igrač pritiskom na pripadajući gumb treba što prije ugasi kako bi se upalila iduća i čim jednu LEDicu ugasi dobiva jedan bod. Na kraju igre se ispiše rezultat te nakon rezultata se ispiše highscore za tu igru u odabranom vremenu.



Slika 8 Prikaz igranja prve igre

QR CODE

-za video prikaz igranja prve igre



2.3.2. Igra 2 – Provjeri reflekse

Druga igra, Provjeri reflekse, radi na način da nakon što počne igra, random se bira LEDica koja će se upaliti, ali samo u prvom redu interaktivne ploče te je igrač pritiskom na pripadajući gumb treba što prije ugasi kako bi se upalila iduća. Čim jednu LEDicu ugasi dobiva jedan bod. Na kraju igre se ispiše rezultat te nakon rezultata se ispiše highscore za tu igru u odabranom vremenu.



Slika 9 Prikaz igranja druge igre

QR CODE

-za video prikaz igranja druge igre



2.3.3. Igra 3 – Vježba za jednu ruku

Treća igra, Vježba za jednu ruku, radi na način da nakon što počne igra, pale se dvije LEDice po kutovima ploče te se tako upali prvo jedna LEDica u jednom kutu te druga u suprotnom kutu (po dijagonali ili u istom redu, ali na drugom kraju toga reda) i tako se ponavlja 5 puta za redom radi razgibavanja i rastezanja. Nakon isteka vremena, ispisuje se rezultat koliko puta je igrač uspio ugaziti po dvije LEDice na ploči.



Slika 10 Prikaz igranja treće igre

QR CODE

-za video prikaz igranja treće igre



2.4. Baza pohranjenih rezultata

Jedna od novih mogućnosti LED pada je i ta da se highscore pohranjuje na Arduinovu ugrađenu memoriju pod nazivom EEPROM.

EEPROM (također E2PROM) označava električno izbrisivu programabilnu memoriju samo za čitanje i vrsta je trajne memorije koja se koristi u računalima, integrirana u mikrokontrolere za pametne kartice i udaljene sustave bez ključa te druge elektroničke uređaje za pohranu relativno male količine podataka omogućujući brisanje i reprogramiranje pojedinih bajtova.

U ovom slučaju, EEPROM sam koristila kako bi na njega pohranila highscore za svaku igru i svako moguće vrijeme u pojedinoj igri. Pohranila sam highscoreove na EEPROM tako da ostanu pohranjeni od prošlih igara i nakon što je ploča isključena iz struje. Svaki od highscoreova je pohranjen na različitoj adresi na EEPROM-u i svaki put kada igrač ostvari novi highscore, pohranit će se na njegovu adresu (ovisno o vrsti igre i trajanju igre) te se nakon odigrane igre u varijablu prema vrijednost pročitana sa EEPROM-a s točno određene adrese.

Adrese se označavaju brojevima (0, 1, 2, 3...) te podaci spremljeni na EEPROM imaju tip podataka byte.



Slika 11 Prikaz rezultata



Slika 12 Prikaz oznake HS (highscore)



Slika 13 Prikaz highscore-a

3. Tehnička dokumentacija

3.1. Hardverski dio

Ova je igra temeljena na softveru koji upravlja hardverom – mikrokontrolerom Arduino MEGA 2560.

Arduino je platforma bazirana na hardveru i softveru. Sastoji se od fizičkih komponenti – programibilne mikrokontrolerske razvojne pločice i softverske aplikacije Arduino IDE (en. *Integrated Development Environment*) koja se koristi za pisanje koda i prebacivanje istog na razvojnu pločicu pomoću USB kabela.

Tehničke specifikacije za Arduino MEGA 2560

Mikrokontroler	ATmega2560
Radni napon	5V
Ulazni napon (preporučeno)	7-12V
Ulazni napon (granica)	6-20V
Digitalni I/O izvodi	54 (od kojih 15 osigurava PWM izlaz)
Analogni ulazni izvodi	16
DC struja po I/O izvodu	20 mA
DC struja za 3.3V izvod	50 mA
Flash memorija	256 KB od čega 8 KB koristi bootloader
SRAM	8 KB
EEPROM	4 KB
Brzina sata	16 MHz
Ugrađene svjetleće diode	13
Duljina	101,52 mm
Širina	53,3 mm
Težina	37 g



Slika 14 ATmega2560

Projekt se sastoji od interaktivne ploče, na kojoj se nalaze gumbi i LED-ice i na kojoj se izvršava igra. Sastoji se još i od drvene konstrukcije kako bi ploča mogla stajati u okomitom položaju. Gumbi i LED-ice su s Arduinoom povezani kablčićima i žicama.

Za izradu konstrukcije i elektroničkog dijela koristila sam sljedeće materijale i alate:

Materijali:

- Arduino MEGA 2560
- 30 svjetlećih dioda
- Žice raznih boja
- Šperploča debljine 4 mm, veličine 1000x1000mm, 1 komad
- Iverica debljine 16 mm, veličine 1500x500 mm, 2 komada
- Iverica debljine 16 mm, veličine 1000x500 mm, 1 komad
- Plastična ploča debljine 1 mm
- Eksperimentalna pločica
- 30 otpornika
- Vijci i matice
- 8 velikih i 8 malih kutnih profila
- Izolir – traka
- Patafix

Alat i pribor:

- Ubodna pila
- Škare
- Pribor za tehničko crtanje (olovke, trokuti, ravnalo...)
- Odvijač
- Turpija
- Bušilica
- Lemilica i lem žica

3.2. Softverski dio

Softver sam napisala u Arduinovom integriranom razvojnom okruženju, koristeći funkcije iz C i C++ programskog jezika.

Arduino IDE otvorenog je koda i potpuno besplatan, može se preuzeti na web stranici [Arduino.cc](https://www.arduino.cc/en/Main/Software), odnosno na sljedećem linku:

<https://www.arduino.cc/en/Main/Software> .

3.2.1. Postavka vremena trajanja igre

```
if (stanje4 == LOW) //ako je pritisnut gumb za odabir vremena trajanja igre...
{
  vrijeme_igre = vrijeme_igre+10; //svaki put kada se pritisne gumb vrijeme trajanja igre se poveca za 10 sekundi
  desetica_vremena = vrijeme_igre / 10 % 10; //pohranjivanje vrijednosti desetica vremena u varijablu

  ugasi_sve();
  digitalWrite(LED30, LOW); //ugasi sve LEDice
  delay(500); //ceka 500 milisekundi

  //ispis vremena na ploči na koje je igrač trenutno prebacio
  desetica_resultata(desetica_vremena);
  jedinica_resultata(0);

  while((stanje4 == HIGH and stanje3 == HIGH and stanje2 == HIGH and stanje == HIGH); //stoji to odredeno ispisano vrijeme
}

if (vrijeme_igre >= 30){ //ako je gumb za odabir vrijeme igre stisnut vise od 3 puta (igrač može staviti da igra traje
  vrijeme_igre = 0; //vrijeme igre se vraća na 0
}
```

Slika 15 Dio koda za postavljanje vremena igre.

Ovaj dio programa služi za odabir vremena igre pritiskom na gumb TIME. Svaki put kada se pritisne taj gumb, vrijeme igre se poveća za 10 sekundi te se na ploči ispiše trenutno odabrano vrijeme i to vrijeme stoji ispisano na ploči dokle god netko ne pritisne ponovo taj gumb (tada se ponovo poveća za 10 sekundi) ili dok netko ne pritisne gumb za odabir igre. Ako je varijabla vrijeme_igre veća od 30, tada se vraća na 0 jer je 30 sekundi trenutno maksimalno postavljeno vrijeme trajanja igre.

3.2.2. Odabir igre

```
if (stanje == LOW) //ako je pritisnut gumb za promjenu levela
{
  vrsta_igre++; //varijablina vrijednost se povecava za 1 zbog odabira igre
  delay(500); //ceka 500 milisekundi
  upali = 0; //pohranjivanje vrijednosti pomocnoj varijabli koja oznacava da igra traje (1) ili dok ne traje igra (0)
}

if (vrsta_igre == 1){ //ako je odabran prvi level

  zgasi_sve(); //gasi sve LEDice
  jedinica rezultata(1); //prikaz broja 1 na ploci
  upali = 0; //pohranjivanje vrijednosti pomocnoj varijabli koja oznacava da igra traje (1) ili dok ne traje igra (0)

  if (stanje2 == LOW){ //ako je pokrenut prvi level

    zgasi_sve(); //gasi sve LEDice

    pocetno(); //izvravanje pocetne animacije
    delay(1000); //prickeka 1 sekundu (1000 milisekundi) prije pocetka igre

    rezultat = 0; //rezultat se postavlja na 0 bodova
    odbrojavanje = 0; //odbrojavanje se postavlja na 0 jer jos nije krenulo odbrojavat
    pocetno_vrijeme = millis(); //pocetno vrijeme se oznacava u milisekundama
```

Slika 16 Dio koda za odabir igre.

Ovaj dio programa služi za odabir vrste igre. Pritiskom na gumb GAME varijabla vrsta_igre povećava se za 1 i ispisuje se broj te igre (1, 2 ili 3). Pritiskom na gumb START počinje odabrana igra nakon animacije. Odabrana igra traje onoliko sekundi koliko je igrač postavio na gumbu TIME.

3.2.2. Prikaz reklamnog teksta

```
if (stanje3 == LOW) //ako je pritisnut gumb za ispis LED PAD...
{
  //pozivanje funkcije za ispis svakog slova
  slovoL();
  slovoE();
  slovoD();

  delay(1000); //prickeka jednu sekundu (1000 milisekundi) između dvije riječi (LED i PAD)

  slovoP();
  slovoA();
  slovoB();
}
```

Slika 17 Dio koda za prikaz teksta.

Ovaj dio programa služi da, kada je pritisnut gumb AD, trčecim se slovima ispišu riječi LED i PAD na način da je svako slovo smješteno u posebnu funkciju te se te funkcije pozivaju redosljedom tako da ispisuju te dvije riječi. Između dvije riječi je pauza od 1 sekundu (1000 milisekundi).

3.2.3. Igra 1

```
void igra1(int qum1, int LED) //funkcija za prvu igru - svaki put kad jednu LEDicu igrac ugasi, druga ce se upaliti
{
    sqasi_sve();
    digitalWrite(LED, HIGH); //gasi sve i pali jednu odabranu random LEDicu

    while (digitalRead(qum1) != LOW) // ceka da se tipka pritisne
        delay(100);

    while (digitalRead(qum1) == LOW) // ceka da se tipka otpusti
        ;

}

void igra_biranje1() //funkcija za biranje random LEDice koja ce se upaliti u prvoj igri
{
    biranje = random(0,20); //bira se random broja od 0 do 20

    for (int y = 0; y < brojPinova; y++) //provjerava ako je odabrana y-ta random broj, poziva funkciju i y-tom LEDicom i y-tim qumom i tom slučaju
    if (ledPinovi[y] == biranje){
        digitalWritePinovi[y], ledPinovi[y]; //pozivanje funkcije za i jek prve igre
        rezultat++; //povećavanje varijable rezultata za 1
    }
}
;
```

Slika 18 Dio koda za igru1.

Ovo su dvije funkcije koje služe za biranje random LED-ice (donja funkcija) i tada se ta odabrana LED-ica upali (gornja funkcija). Gornja funkcija služi i da random odabrana LED-ica ostane upaljena, a sve ostale ugašene, dokle god nije pritisnut pripadajući gumb odabrane LED-ice.

3.2.4. Igra 2

```
void igra2(int qum2, int LED2) //funkcija za drugu igru - svaki put kad jednu LEDicu igrac ugasi, druga ce se upaliti. ali samo u prvom redu
{
    sqasi_sve();
    digitalWrite(LED2, HIGH); //gasi sve i pali jednu odabranu random LEDicu

    while (digitalRead(qum2) != LOW) // ceka da se tipka pritisne
        delay(100);

    while (digitalRead(qum2) == LOW) // ceka da se tipka otpusti
        ;

}

void igra_biranje2() //funkcija za biranje random LED ice koja ce se upaliti u drugoj igri
{
    biranje = random(0,6); //bira se random broja od 0 do 6

    for (int y = 0; y < brojPinova; y++) //provjerava ako je odabrana y-ta random broj, poziva funkciju s y-tom LEDicom i y-tim qumom u tom slučaju
    if (ledPinovi[y] == biranje){
        igra2(qum2Pinovi[y], ledPinovi[y]); //pozivanje funkcije za tipku druge igre
        rezultat--; //povećavanje varijable rezultata za 1
    }
}
;
```

Slika 19 Dio koda za igru2.

Ovo su dvije funkcije koje služe za biranje random LED-ice (donja funkcija) i tada se ta odabrana LED-ica upali (gornja funkcija) samo u prvom redu. Gornja funkcija služi i da random odabrana LED-ica ostane upaljena, a sve ostale ugašene, dokle god nije pritisnut pripadajući gumb odabrane LED-ice.

3.2.5. Igra 3

```
void igra3(int gumb_1, int LED_3, int gumb_2_2, int LED_2_2) //funkcija za igru - postavljanje 2 para random odabranih dviju LEDica
{
    for(int i = 0; i < 5; i++)
    {
        digitalWrite(LED_3, HIGH); //gumb 1 pali jednu odabrano random LEDicu
        digitalWrite(LED_2_2, HIGH); //gumb 2 pali drugu odabrano random LEDicu
        while (digitalRead(gumb_1) != LOW); //veka da se tipka pritisne
        digitalWrite(LED_3, LOW);

        digitalWrite(LED_2_2, HIGH); //gumb 2 pali drugu odabrano random LEDicu
        digitalWrite(LED_1_1, HIGH); //gumb 1 pali jednu odabrano random LEDicu
        while (digitalRead(gumb_2_2) != LOW); //veka da se tipka pritisne
        digitalWrite(LED_2_2, LOW);
    }
}

void igra_biraenje() //funkcija za biranje dvije random LEDice na njihovo postavljanje 5 puta
{
    biranje = random(0,3); //biranje random broja od 0 do 2

    if(biranje == 0) //ako je random odabran broj 0
        igra3(gumb3, LED3, gumb1, LED1); //izvršavanje funkcije za treću igru s prvom i 3. LEDicom
    else if(biranje == 1) //ako je random odabran broj 1
        igra3(gumb1, LED1, gumb2, LED2); //izvršavanje funkcije za drugu igru s prvom i 2. LEDicom
    else //ako je random odabran broj 2
        igra3(gumb2, LED2, gumb4, LED4); //izvršavanje funkcije za treću igru s 2. i 4. LEDicom
}
}
```

Slika 20 Dio koda za igru3.

Ovo su dvije funkcije koje pale dvije LEDice po kutovima ploče te se tako upali prvo jedna LEDica u jednom kutu te druga u suprotnom kutu (po dijagonali ili u istom redu, ali na drugom kraju toga reda) i tako se ponavlja 5 puta za redom.

3.2.5. Spremanje rezultata u bazu

```
trenutni_rezultat = rezultat; //u varijablu trenutni rezultat se pohranjuje vrijednost rezultata nakon odigrane igre

if(trenutni_rezultat1 > high_1) //ako je trenutni rezultat veći od dosadašnjega highscorea
{
    if(vrijeme_igre == 10000) //ako je vrijeme igre bilo postavljeno na 10 sekundi
        EEPROM.write(0, trenutni_rezultat1); //na adresi 0 se na EEPROM ispisuje vrijednost novog highscorea
    else if(vrijeme_igre == 20000) //ako je vrijeme igre bilo postavljeno na 20 sekundi
        EEPROM.write(1, trenutni_rezultat1); //na adresi 1 se na EEPROM ispisuje vrijednost novog highscorea
    else //ako je vrijeme igre bilo postavljeno na 30 sekundi
        EEPROM.write(2, trenutni_rezultat1); //na adresi 2 se na EEPROM ispisuje vrijednost novog highscorea
}
}
```

Slika 21 Dio koda za spremanje rezultata u bazu.

Pohranjuje highscore za svaku igru i svako moguće vrijeme u pojedinoj igri.

3.2.6. Rubni slučajevi

```
void animacija_za_rezultat_preko_100() //funkcija za izvođenje animacije LEDicama kada je igrač postigao 100 ili više bodova
{
    digitalWrite(LED1, HIGH);
    delay(100);
    digitalWrite(LED2, HIGH);
    digitalWrite(LED7, HIGH);
    delay(100);
    digitalWrite(LED3, HIGH);
    digitalWrite(LED8, HIGH);
    digitalWrite(LED10, HIGH);
    digitalWrite(LED4, HIGH);
    delay(100);
    digitalWrite(LED9, HIGH);
    digitalWrite(LED14, HIGH);
    digitalWrite(LED19, HIGH);
    delay(100);
    digitalWrite(LED5, HIGH);
    digitalWrite(LED16, HIGH);
    digitalWrite(LED18, HIGH);
    digitalWrite(LED20, HIGH);
    digitalWrite(LED25, HIGH);
    delay(100);
    digitalWrite(LED6, HIGH);
    digitalWrite(LED11, HIGH);
    digitalWrite(LED18, HIGH);
    digitalWrite(LED21, HIGH);
    digitalWrite(LED26, HIGH);
}
```

Slika 22 Dio koda za slučaj kada rezultat prelazi vrijednost 100.

Kada vrijednost rezultata prijeđe 100 tada se izvodi posebna animacija jer trenutno nema dovoljno LEDica za ispis troznamenkastog broja.

```
if (vrijeme_igre >= 30){ //ako je gumb za odabir vrijeme igre stisnut vise
    vrijeme_igre = 0; //vrijeme igre se vraca na 0
}

if (vrijeme_igre == 0){ //ako je vrijeme igre 0 (odnosno ako je bilo postavljeno na 30 ali
    vrijeme_igre = 30000; //varijabla dobiva vrijednost 30000 (zbog milisekundi n_je 30)
}
else{ //inace ako je vrijeme postavljeno na 10 ili 20 sekundi
    vrijeme_igre = vrijeme_igre*1000; //vrijeme igre se mnozi s tisuću zbog milisekundi
}
```

Slika 23 Dio koda za slučaj kada je igrač odabrao 30 sekundi kao vrijeme igre.

Kada je igrač odabrao 30 sekundi kao vrijeme igre, zbog gornjeg if-a vrijednost se odmah vrati na 0 jer kad bih stavila samo znak >, a ne >= onda bi igrač mogao odabrati i 40 sekundi, što je veće od maksimuma. Zato se vrijednost mora odmah vratiti na 0 pa je zbog toga potreban donji if da ako je vrijednost varijable 0 (odnosno ako je igrač odabrao 30 sekundi) tada se u varijablu pohranjuje vrijednost od 30 sekundi kako bi igra dobro radila (u programu piše 30000 i pod else se množi s 1000 jer se kod vremena koriste milisekunde, a ne sekunde, a jedna sekunda ima 1000 milisekundi).

4. Mogućnosti primjene projekta

S obzirom da su razvoj reakcija i kognitivnih sposobnosti, poput fokusa i koncentracije, važni čimbenici u razvoju djece, želja mi je unaprijediti ovaj uređaj i napraviti veću verziju.

Postavljanje uređaja u hol svoje škole

- kako bi se učenici mogli razgibavati tijekom odmora i razvijati svoje motoričke sposobnosti

Postavljanje uređaja u trgovačke centre

- posjetitelji se mogu zabaviti i razgibati tijekom posjeta i odmora od kupovine, a dok se uređaj ne koristi, može prikazivati reklamne poruke i na taj način sam sebe otplaćivati.
- pokrenuti Crowdfunding kampanju za prikupljanje sredstava za izradu komercijalne verzije uređaja
- sponzoriranje projekta od strane budućeg kupca

Postavljanje uređaja u sportske objekte

- uređaj je idealan za **sportske objekte** za vježbu refleksa i razgibavanje sportaša
- sklopiti ugovore o najmu ili kupnji uređaja sa sportskim klubovima ili njihovim osnivačima
- uvesti nove mogućnosti prema zahtjevima korisnika, npr.:
 - vježba za osnovnu razinu kognitivnih sposobnosti - da se pale lampice u crvenoj i plavoj boji te treba ugaziti samo plave
 - igra kojom se pamti redoslijed uključivanja lampica

Postavljanje uređaja u vrtićima i u ustanovama za djecu s poteškoćama u razvoju

- značajnu primjenu uređaj može imati i u ovim ustanovama zbog razvoja motoričke sposobnosti - djetetove sposobnosti voljnog i svrhovitog korištenja dijelova vlastitog tijela
- tražiti podršku osnivača ustanova i ravnatelja
- donirati nekoliko uređaja koji bi bili financirani kroz neki EU projekt

Postavljanje uređaja u ustanove za starije osobe

- ipak, glavna mi je ideja i želja ovaj uređaj postaviti u ustanove za starije osobe jer je prvenstveno njima i namijenjen za razgibavanje i vježbu koncentracije i pamćenja