

Razvoj tipkovnice pokretane Arduino mikrokontrolerom

Bićanić, Ana

Undergraduate thesis / Završni rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:235:476982>

Rights / Prava: [Attribution-NonCommercial-ShareAlike 4.0 International](#)/[Imenovanje-Nekomercijalno-Dijeli pod istim uvjetima 4.0 međunarodna](#)

Download date / Datum preuzimanja: **2024-07-22**

Repository / Repozitorij:

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Ana Bićanić

Zagreb, 2022.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Mentor:

Doc. dr. sc. Branimir Škugor

Student:

Ana Bićanić

Zagreb, 2022.

Izjavljujem da sam ovaj radi izradila samostalno koristeći znanja stečena tijekom studija i navedenu literaturu.

Zahvaljujem se mentoru, doc. dr. sc. Branimiru Škugoru, na pruženoj prilici te ukazanoj pomoći pri izradi ovoga rada.

Zahvaljujem se dr. sc. Frani Pamukoviću, na pruženoj pomoći i utrošenom vremenu tijekom izrade ovoga rada.

Zahvaljujem se svojim roditeljima na neizmjerne podršci i pomoći tijekom studiranja.

Također se zahvaljujem svojim prijateljima na pruženoj podršci tijekom studija.

Ana Bićanić



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE



Središnje povjerenstvo za završne i diplomske ispite
Povjerenstvo za završne i diplomske ispite studija strojarstva za smjerove:
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment, inženjerstvo
materijala i mehatronika i robotika

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa: 602 – 04 / 22 – 6 / 1	
Ur.broj: 15 - 1703 - 22 -	

ZAVRŠNI ZADATAK

Student: **Ana Bičanić** JMBAG: **0035215772**

Naslov rada na hrvatskom jeziku: **Razvoj tipkovnice pokretane Arduino mikrokontrolerom**

Naslov rada na engleskom jeziku: **Development of a keyboard equipped with Arduino microcontroller**

Opis zadatka:

Prema procjenama UN-a na svijetu je gotovo milijarda ljudi, odnosno 15% svjetske populacije, s nekom vrstom invaliditeta. U isto vrijeme, većina računalne periferije poput miševa i tipkovnica nisu izrađeni imajući na umu ovaj segment populacije. Cilj ovoga rada je razviti tipkovnicu namijenjenu osobi kojoj je omogućeno korištenje samo jedne ruke. Tijekom razvoja tipkovnice posebna pažnja će se posvetiti izgledu tipkovnice te rasporedu i obliku tipki kako bi se olakšalo korištenje osobi s invaliditetom. Također, tipkovnicu će pokretati Arduino mikrokontroler čijim će se programiranjem omogućiti izmjena tipki prema potrebi korisnika.

Tijekom rada potrebno je:

1. korištenjem programa za 3D modeliranje, Solidworks 2020, kreirati 3D model odnosno izgled tipkovnice i svih njenih dijelova,
2. pripremiti i isprintati model na 3D printeru (Artillery Genius) FDM metodom,
3. upoznat se s elektroničkim dijelovima (prekidači, diode, mikrokontroleri, itd.) potrebnim za osposobljavanje tipkovnice, te napraviti shematski prikaz budućeg „ožičenja“,
4. programirati Arduino Uno mikrokontroler kako bi tipkovnica bila osposobljena za komunikaciju s osobnim računalom USB metodom.

U radu je potrebno navesti korištenu literaturu i eventualno dobivenu pomoć.

Zadatak zadan:

30. 11. 2021.

Datum predaje rada:

1. rok: 24. 2. 2022.
2. rok (izvanredni): 6. 7. 2022.
3. rok: 22. 9. 2022.

Predviđeni datumi obrane:

1. rok: 28. 2. – 4. 3. 2022.
2. rok (izvanredni): 8. 7. 2022.
3. rok: 26. 9. – 30. 9. 2022.

Zadatak zadao:

Doc. dr. sc. Branimir Škugor

Predsjednik Povjerenstva:

Prof. dr. sc. Branko Bauer

SADRŽAJ

SADRŽAJ	5
POPIS SLIKA	6
POPIS OZNAKA	7
SAŽETAK.....	8
SUMMARY	9
1. UVOD.....	10
2. ZAHTJEVI KORISNIKA ZA IZRADU TIPKOVNICE	11
2.1. Veličina tipki	11
2.2. Raspored i broj tipki	11
2.3. Veličina i oblik tipkovnice	12
3. 3D MODEL	13
3.1. Veličina tipki prototipne tipkovnice.....	13
3.2. Mehanizam unutar tipke.....	14
3.2.1. Proračun opruge	14
3.3. Raspored i broj tipki prototipne tipkovnice.....	15
3.4. Veličina i oblik prototipne tipkovnice.....	16
4. 3D ISPIS	17
4.1. FDM metoda	17
4.1.1. Postupak FDM ispisa	18
4.2. Softveri i postavke FDM printera.....	19
4.3. Ispis tipkovnice	19
5. SPAJANJE KOMPONENTI NA ARDUINO UNO	21
5.1. Arduino uno ploča	21
5.2. Komponente korištene za spajanje tipkovnice	22
5.3. Shema spajanja dijelova za tipkovnicu	23
6. PROGRAMIRANJE ARDUINO MIKROKONTROLERA.....	26
6.1. Struktura Arduino programskog jezika	26
6.2. Arduino kod za tipkovnicu.....	27
7. SPAJANJE HID TIPKOVNICE KAO USB PRIKLJUČKA.....	29
8. ZAKLJUČAK.....	33
LITERATURA.....	34
PRILOZI.....	35

POPIS SLIKA

Slika 2.1 Standardna veličina tipke [3]	11
Slika 2.2 Standardna udaljenost tipki [4]	12
Slika 2.3 Veličina standardne tipkovnice [5]	12
Slika 3.1 Izometrija tipke	13
Slika 3.2 Mehanizam tipke, ispružena opruga	15
Slika 3.3 Raspored tipki	15
Slika 3.4 Gabaritne dimenzije	16
Slika 4.1 Dijelovi FDM printera [11]	18
Slika 4.2 Postupak FDM ispisa, spuštanjem podloge [12]	19
Slika 5.1 Dijelovi Arduino Uno ploče [13]	21
Slika 5.2 Matična ploča [13]	22
Slika 5.3 Kratkospojne žice [13]	22
Slika 5.4 Tipka [13]	22
Slika 5.5 Otpornici [13]	23
Slika 5.6 Matična ploča i priključci s donje strane [13]	23
Slika 5.7 Vodljive trake unutar matične ploče [13]	24
Slika 5.8 Funkcioniranje tipke [13]	24
Slika 5.9 Shema napajanja tipkovnice	25
Slika 6.1 Šifre za programiranje tipke pomoću buffera [16]	28
Slika 7.1 Resetiranje Arduino Uno ploče	29
Slika 7.2 Odabir uređaja	30
Slika 7.3 Odabir ulaza	30
Slika 7.4 Učitavanje HEX datoteke	31
Slika 7.5 Vidljivost Arduino Uno Porta na popisu uređaja	31
Slika 7.6 Prikaz HID tipkovnice u Upravitelju uređaja	32

POPIS OZNAKA

Oznaka	Jedinica	Opis
<i>F</i>	N	sila
<i>G</i>	N/mm ²	modul klizanja
<i>d</i>	mm	promjer
<i>f</i>	mm	hod opruge
<i>D</i>	mm	srednji promjer
<i>i</i>	-	broj radnih navoja
<i>D_u</i>	mm	unutarnji promjer
<i>R</i>	Ω	električni otpor
<i>U</i>	V	napon

SAŽETAK

Tema ovoga rada je izrada prototipa tipkovnice prilagođene osobi s invaliditetom, kako bi joj se omogućilo nesmetano korištenje računala u svrhu igranja specifične video igre pri čemu je ostavljena mogućnost prenamjene tipkovnice za neku drugu svrhu. 3D model prototipne tipkovnice izrađen je u programskom paketu SolidWorks, prema zahtjevima korisnika. Model je ispisan pomoću 3D printera FDM metodom. Nakon konstrukcijskog dijela, osmišljen je i prikazan način upravljanja tipkovnicom pomoću Arduino Uno mikrokontrolera te popratnih dijelova poput kratkospojnih žica, otpornika itd. U skladu s primarnom namjenom, napisan je kod kojim je programirana tipkovnica. Trenutno je ona programirana za igranje određene igrice, ali se ista tipkovnica može naknadno reprogramirati u bilo koju drugu svrhu jednostavnom izmjenom upravljačkog koda Arduina. Na posljetku je još izvršen prijenos napisanog koda s računala na samu tipkovnicu. Taj postupak naziva se *flashanje* i izvršen je pomoću softvera Flip 3.4.7, kako bi računalo putem USB priključka prepoznalo Arduino Uno kao HID tipkovnicu.

Ključne riječi: tipkovnica, 3D ispis, FDM, Arduino Uno, HID tipkovnica

SUMMARY

The topic of this paper is the development of a prototype keyboard adapted for a person with a disability, in order to enable uninterrupted use of computer for the purpose of playing a specific video game, leaving the possibility of reprogramming the keyboard for another purpose. The 3D model of the prototype keyboard was created in the SolidWorks software package, according to user requirements. The model was printed using a 3D printer with the FDM method. After the construction part, the way of running the keyboard with the Arduino Uno microcontroller and the accompanying parts such as short-circuited wires, resistors, etc. was designed and shown. Currently, it is programmed to play a particular game, but the same keyboard can be reprogrammed for any other purpose by simply changing the Arduino control code. Finally, it is necessary to transfer the written code from the computer to the keyboard itself. This procedure is called flashing and was performed using Flip 3.4.7 software, so that the computer can recognize the Arduino Uno as a HID keyboard via a USB port.

Keywords: keyboard, 3D printing, FDM, Arduino Uno, HID keyboard

1. UVOD

Oko 15% svjetske populacije ima neki oblik invaliditeta što bi značilo da više od milijardu ljudi u svijetu živi s nekim oblikom invaliditeta, a od čega gotovo 200 milijuna ima značajnih poteškoća u funkcioniranju. U budućnosti će te brojke sve više rasti. Taj porast je posljedica starenja stanovništva, globalnog povećanja kroničnih zdravstvenih stanja poput dijabetesa, kardiovaskularnih bolesti, raka i poremećaja mentalnog zdravlja.

U cijelom svijetu osobe s invaliditetom imaju lošije zdravstvene ishode, niža obrazovna postignuća i veće stope siromaštva od ljudi bez invaliditeta. To je dijelom i zato što osobe s invaliditetom nailaze na prepreke u pristupu uslugama koje mnogi od nas već odavno uzimaju zdravo za gotovo, uključujući zdravstvenu skrb, obrazovanje, zapošljavanje, mobilnost kao i dostupnost informacija. [1]

Osim navedenih poteškoća, osobe s invaliditetom su na neki način zakinite i u ostalim aspektima svakodnevnog života. Upravo zbog toga, neke od posljedica i učinaka invaliditeta su i problemi s mentalnim zdravljem, uključujući anksioznost i depresiju, gubitak slobode i neovisnosti te frustracija i ljutnja zbog oslanjanja na druge ljude.

Od navedenih milijardu ljudi koji živi s nekim oblikom invaliditeta, mnogo je onih koji intenzivno ili iz zabave igraju video igrice. Prema jednom istraživanju, 20,5% igrača ima mentalni, fizički ili razvojni poremećaj. S obzirom da je broj igrača u svijetu porastao na 2,5 milijarde u 2019., broj igrača s invaliditetom ne može se zanemariti. Stoga je motivacija ovoga rada bila olakšati igranje videoigara upravo tom dijelu populacije koja je na neki način zanemarena u svjetskoj industriji videoigara. [2]

Konkretno, u ovome radu pomoći će se korisniku koji je rođen bez podlaktice. Korisnik je veliki zaljubljenik u video igricu League of Legends (LoL) te mu standardna tipkovnica otežava igranje igrice, o čemu će više biti govora u idućem poglavlju.

Razvoj cijele tipkovnice sastojat će se od sljedećih koraka: 3D modeliranja i 3D ispisa tipkovnice, spajanja elektronskih dijelova te programiranja Arduino mikrokontrolera.

2. ZAHTJEVI KORISNIKA ZA IZRADU TIPKOVNICE

Prilikom izrade tipkovnice korisnik je istaknuo najvažnije zahtjeve te potrebne specifikacije koje tipkovnica mora sadržavati. Konkretno specifikacije moraju se uzeti u obzir prije same konstrukcije i programiranja tipkovnice. Navedeni zahtjevi detaljnije su opisani u idućim odlomcima.

2.1. Veličina tipki

Veličina tipki mora biti veća nego kod standardnih tipkovnica zato što korisnik tipke pritišće svojom podlakticom, koja je manje precizna i „nezgrapnija“ od prstiju kojima se koriste ljudi bez konkretnog invaliditeta. Stoga dolazimo do zaključka da je jedan od bitnijih faktora za određivanje veličine tipke razlika u preciznosti koju nam pruža pritiskanje podlakticom u odnosu na onu koju imamo prilikom pritiskanja tipke prstima.

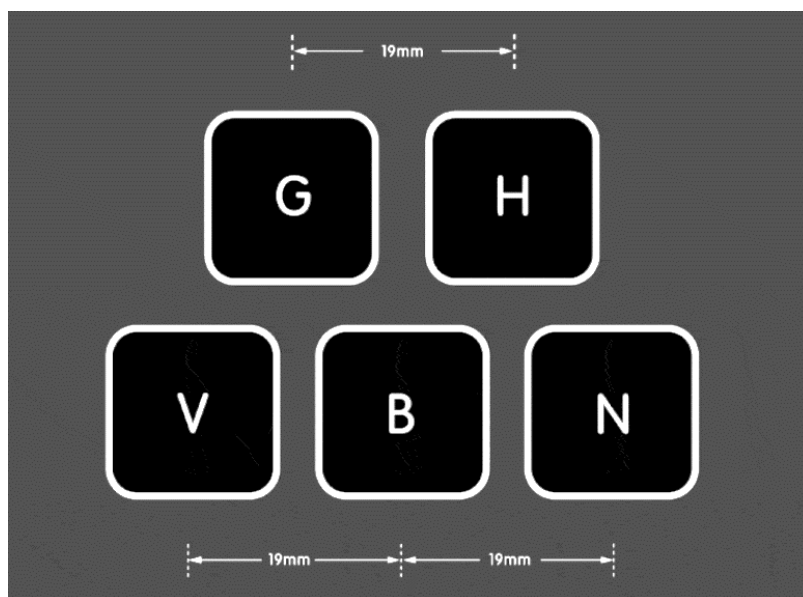


Slika 2.1 Standardna veličina tipke [3]

Na slici 2.1 prikazana je standardna veličina tipke, koja je prilagođena veličini jagodice ljudskog prsta (u prosjeku 12,5 mm x 8 mm).

2.2. Raspored i broj tipki

Razmak središta tipki na standardnoj tipkovnici iznosi u prosjeku 19 mm [Slika 2.2]. Raspored tipki na ovoj tipkovnici bit će znatno rjeđi i podijeljen po kategorijama odnosno funkcionalnostima koje određene tipke imaju. Tipke će biti raspoređene tako da omogućе što brže i lakše pritiskanje.



Slika 2.2 Standardna udaljenost tipki [4]

Standardna tipkovnica ima 101 tipku, dok će za potrebe konkretnog korisnika biti dovoljno 13 tipki. Tih 13 tipki koristit će se za igranje igrice, ali će one isto tako biti korisne u mogućoj prenamjeni tipkovnice u druge svrhe što je moguće jednostavnom izmjenom upravljačkog koda Arduina, a što je detaljno opisano u [poglavlju 6](#).

2.3. Veličina i oblik tipkovnice

Veličina i oblik tipkovnice bit će prilagođeni pokretima korisnika prilikom igranja. U obzir će se uzeti raspon okreta nadlaktice te smanjeni brojevi stupnjeva slobode što znači da će tipkovnica biti manjih dimenzija od standardnih.



Slika 2.3 Veličina standardne tipkovnice [5]

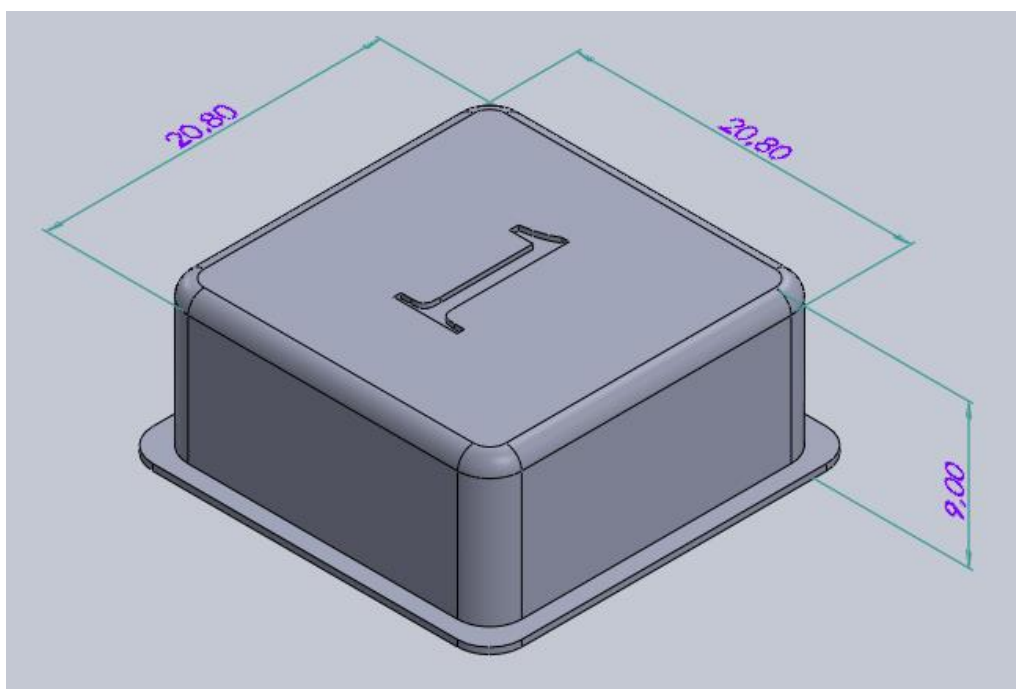
Način uključivanja svih navedenih zahtjeva u razvoj ove tipkovnice bit će detaljnije razrađen i prikazan u fazama 3D modeliranja, 3D ispisa te samog programiranja tipkovnice.

3. 3D MODEL

3D model jedna je od bitnijih stavki za izradu tipkovnice. Na temelju modela stvara se jasnija vizualizacija o tome kako će tipkovnica izgledati te je konkretan model baza za 3D ispis, tehnologiju pomoću koje će se izraditi prototip tipkovnice.

Pomoću 3D modeliranja pokušalo se, u najvećoj mogućoj mjeri, zadovoljiti potrebe korisnika što se tiče samog rasporeda i veličine tipkovnice. Kao što je već prikazano u [poglavlju 2](#), postoje određeni standardi prilikom konstruiranja tipkovnice, koji su u ovom slučaju izmijenjeni. Detaljnija razrada bit će prikazana pomoću više pogleda na 3D model koji je izrađen u programskom paketu SolidWorks.

3.1. Veličina tipki prototipne tipkovnice



Slika 3.1 Izometrija tipke

Prototipna tipka značajno je veća od standardne, što se vidi iz priložene slike [[Slika 3.1](#)]. Njezine vanjske dimenzije su 20,80 mm x 20,80 mm te je ukupne visine 9 mm. Svojom visinom ona djelomično ulazi u konstrukciju tipkovnice tako da samo 5 milimetara ostaje iznad gornje površine tipkovnice. Razlog tome je mehanizam pomoću kojeg sama tipka funkcionira, a on će biti objašnjen u [točki 3.2](#).

3.2. Mehanizam unutar tipke

Unutar same tipke ukomponiran je i mehanizam pomoću kojega se pritišće gumb koji je smješten na donjoj površini tipkovnice te pomoću žica povezan s Arduino mikrotkontrolerom. Mehanizam je napravljen po uzoru na standardne tipkovnice s lineranim prekidačem koji funkcionira pomoću tlačne opruge. Prikazan je kao jednostavan cilindar koji će se zajedno s tipkom ispisati pomoću 3D printera. Oko cilindra se još dodaje opruga koja apsorbira silu pritiskanja te omogućava praktičnije pritiskanje tipke.

3.2.1. Proračun opruge

Proračun opruge računa se prema sljedećem izrazu:

$$F = \frac{G \cdot d^4 \cdot f}{8 \cdot D^3 \cdot i}, \text{ N} \quad (3.1)$$

gdje je:

F [N] – sila koja opterećuje oprugu

G [N/mm²] – modul klizanja

d [mm] – promjer žice

f [mm] – hod opruge

D [mm] – srednji promjer opruge

i - broj radnih navoja.

Istraživanjem linearnih mehanizama te najpovoljnije sile pritiskanja tipke, došlo se do podatka kako igrači videoigara preferiraju tipke sa pritiskom silom $F = 0,45$ N. [6]

Nadalje, hod opruge izračunat je mjerenjem udaljenosti od dna cilindra tipke do površine gumba te on iznosi $f = 3,2$ mm.

Broj radnih navoja određen je proizvoljno te on iznosi $i = 15$.

Modul klizanja ovisi o vrsti materijala od koje je opruga napravljen. Za potrebe izrade tipkovnice uzeta je vrijednost za patentiranu žicu koja iznosi $G = 83\,000$ N/mm². [7]

Sada preostaje izračunati promjer žice. Srednji promjer D možemo izraziti pomoću promjera žice na sljedeći način:

$$D = d + D_u, \text{ mm} \quad (3.2)$$

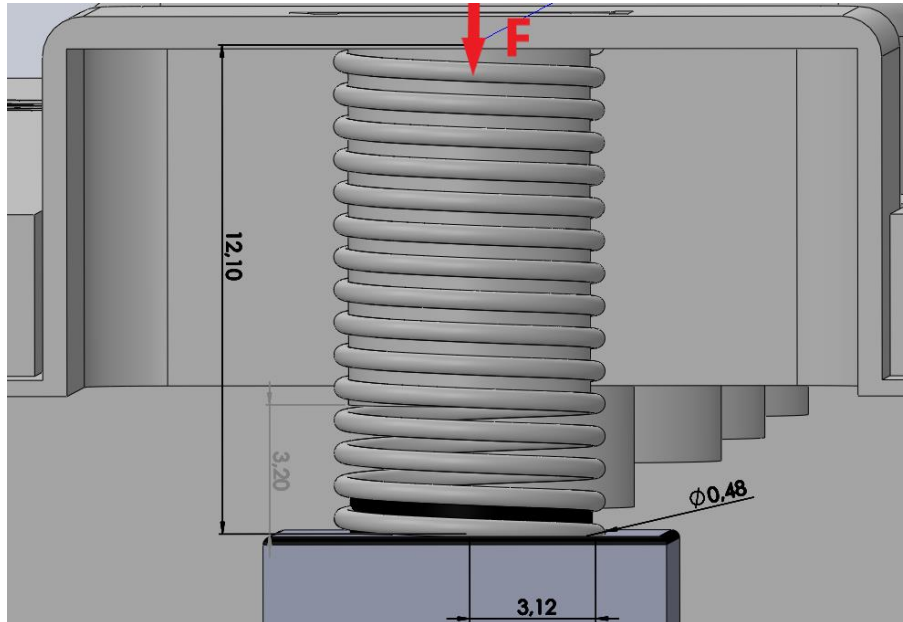
D_u [mm] – unutarnji promjer opruge

U našem slučaju D_u predstavlja promjer cilindra koji iznosi $D_u = 6$ mm.

Uvrštavanjem poznatih vrijednosti u formulu, dobijemo sljedeći izraz:

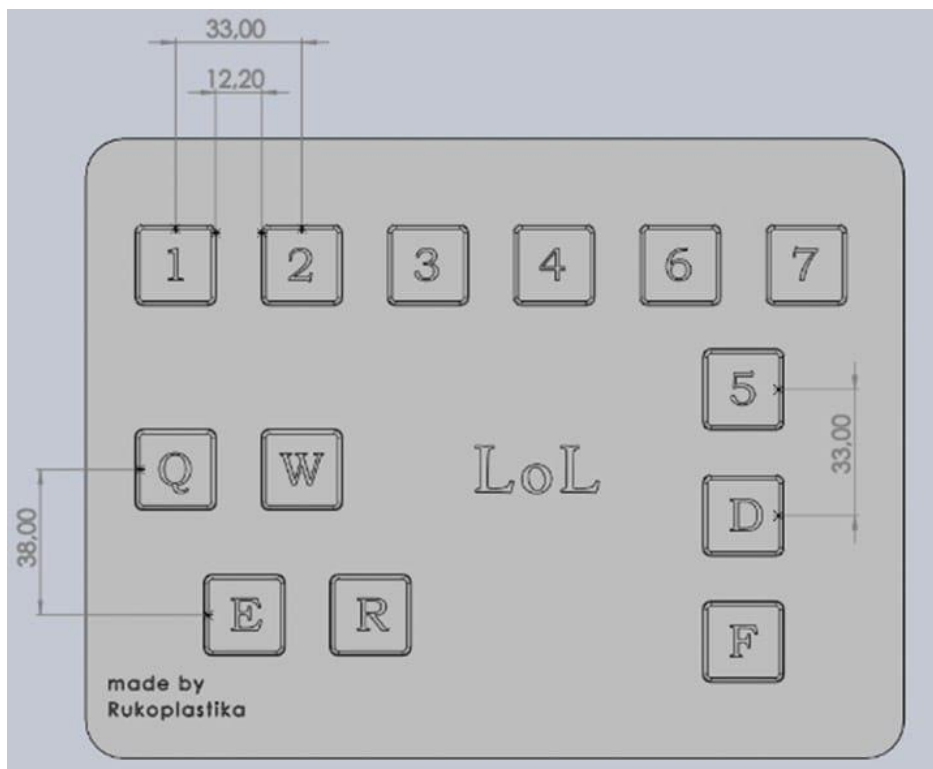
$$0,45 = \frac{83000 \cdot d^4 \cdot 3,2}{8 \cdot (d+6)^3 \cdot 15} \quad (3.3)$$

Iz dobivenog izraza vrijednost d iznosi 0,48 mm.



Slika 3.2 Mehanizam tipke, ispružena opruga

3.3. Raspored i broj tipki prototipne tipkovnice

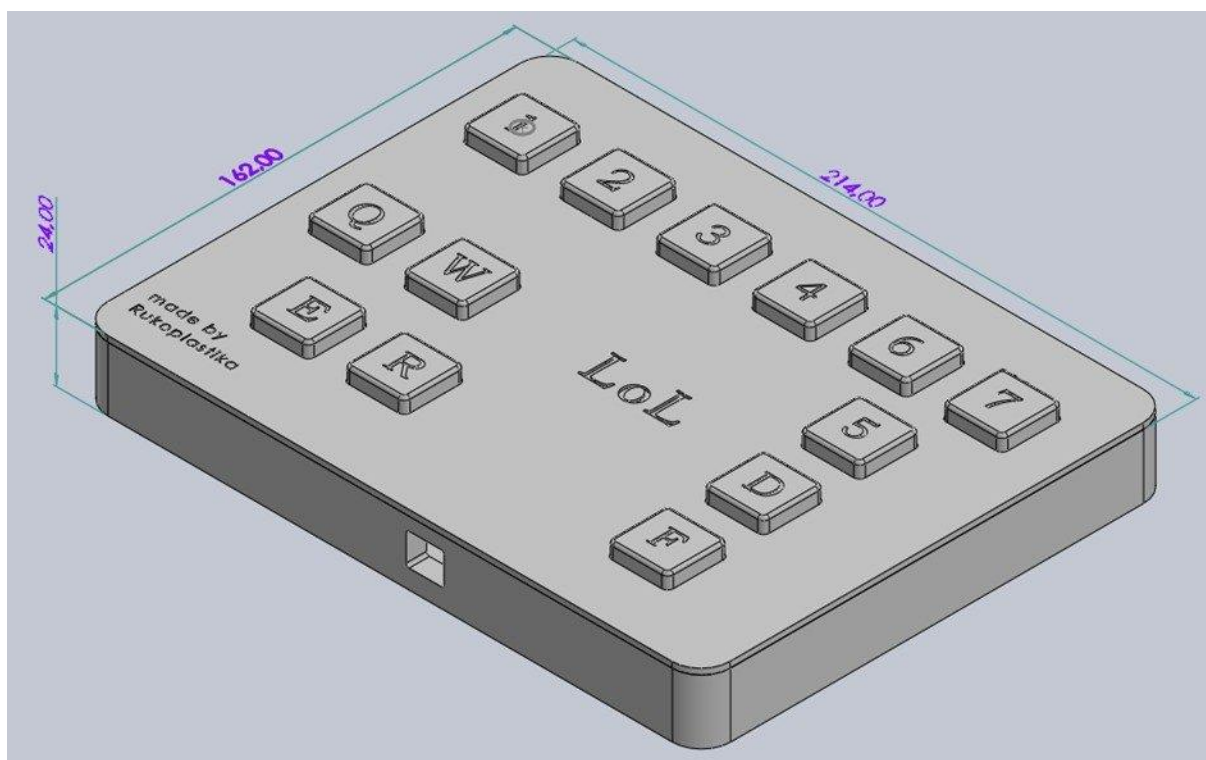


Slika 3.3 Raspored tipki

Prilikom izrade tipkovnice bilo je potrebno osigurati tipke brojeva od 1 do 7 te slova Q, W, E, R, D i F. Raspored prikazan na [slici 3.3](#) je takav upravo zbog potrebe korisnika za upravljanje konkretne video igre. Razmak između tipki koje se redaju jedna za drugom je 33 mm, znatno veća nego kod standardnih tipkovnica, a temelji se na zahtjevu samog korisnika.

3.4. Veličina i oblik prototipne tipkovnice

Gabaritne mjere prototipne tipkovnice, koja će biti ispisana pomoću 3D printera, prikazane su na slici ispod.



Slika 3.4 Gabaritne dimenzije

4. 3D ISPIS

Prilikom izrade ovog prototipa koristila se metoda 3D ispisa, koja je idealna za potreban omjer kvalitete i brzine izrade modela. To je postupak brze izrade prototipa kojim se uređajima zasnovanima na 3D tehnologiji ispisuju fizički predmeti. 3D ispis daje mogućnost jasnog uvida u tok postupka dizajniranja, mogućnost isticanja raznih parametara, mogućnost lakog i ranog uočavanja mogućih grešaka te njihovog brzog i efikasnog ispravljanja. Postoji ukupno 7 kategorija procesa 3D ispisa: polimerizacija u spremniku, ekstrudiranje materijala, očvršćivanje praškastog sloja, visokotlačno srašćivanje materijala, visokotlačno srašćivanje pomoću veziva, taloženje usmjerenom energijom i laminacija slojeva. Svaka od ovih vrsta procesa ima različite tehnologije. Najčešće tehnologije su SLA (stereolitografija koja je tehnologija fotopolimerizacije u spremniku), FDM (modeliranje fuzije taloženja koja je tehnologija ekstrudiranja materijala) te SLS (selektivno lasersko sinteriranje, jedna od tehnologija očvršćivanja praškastim slojem) [8]. Za ispis tipkovnice koristila se FDM metoda, čije su postavke i parametri opisani u [točkama 4.1. i 4.2.](#)

4.1. FDM metoda

FFF, kratica od “Fused Filament Fabrication” je najraširenija i najjeftinija tehnologija 3D ispisa. Prvi puta se pojavila 1992. godine. FFF tehnologiju 3D printanja se često naziva FDM („*Fused Deposition Modeling*” hrv. „tehnologija taložnog očvršćivanja“), a to je naziv kojeg je popularizirala tvrtka Stratasys koja je ujedno i zaslužna za nastanak FDM/FFF tehnologije 3D ispisa. [9]

Prednosti ove metode su te što je:

- povoljna: povoljan je sam uređaj, a i filament koji se koriste,
- filament su višekratno upotrebivi: većina filamenata je na bazi plastike stoga se mogu lako preoblikovati u oblik filameta zagrijavanjem i ravnanjem,
- raznolikost izbora materijala: postoji više od 9 vrsta filamenata,
- printer lako prenosiv: zbog jednostavnog dizajna, njegove se komponente lako odvoje što ga čini vrlo praktičnim i lako se može premjestiti s jednog mjesta na drugo,
- kompaktnog dizajna: ne zauzima previše prostora. Iako vanjski volumen pisača u potpunosti ovisi o volumenu ispisa poput vanjskog okvira, pogonske jedinice itd. ostali dijelovi ne zauzimaju puno prostora. [10]

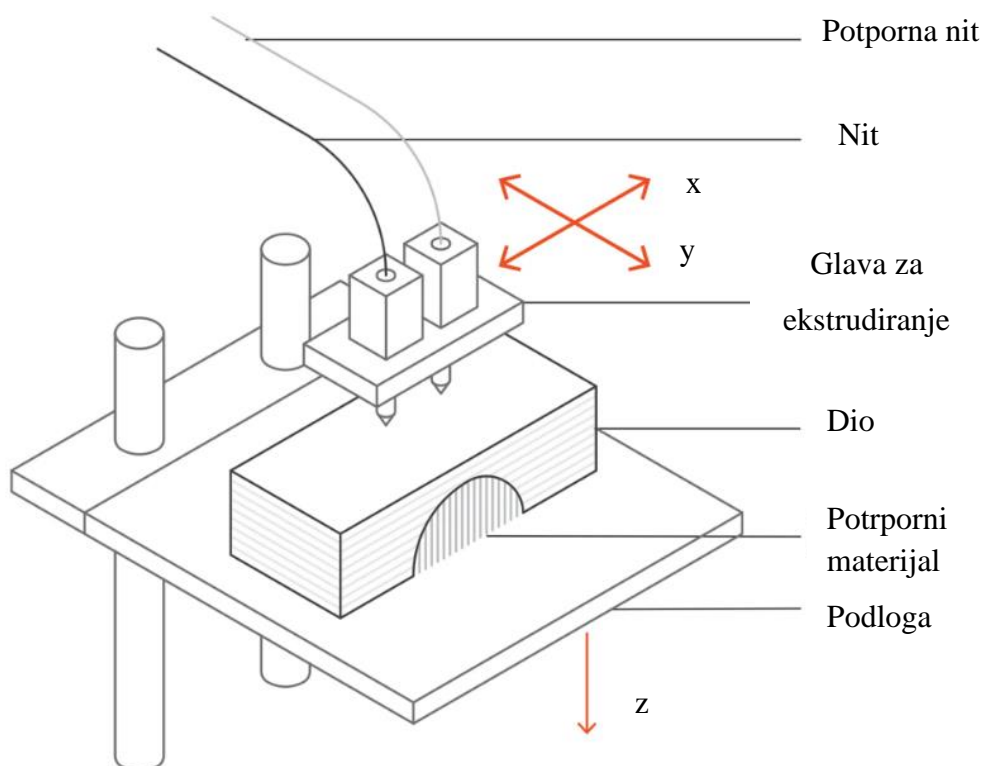
4.1.1. Postupak FDM ispisa

Proces izrade 3D modela FDM metodom je sljedeći: prvo se ubaci kolut termoplastičnog filameta u pisač. Filament je plastična nit, a neke od najčešćih koji se danas koriste su PLA, PETG, ABS i ASA. Nakon što mlaznica postigne željenu temperaturu, filament se dovodi u ekstruzijsku glavu i mlaznicu, gdje se topi.

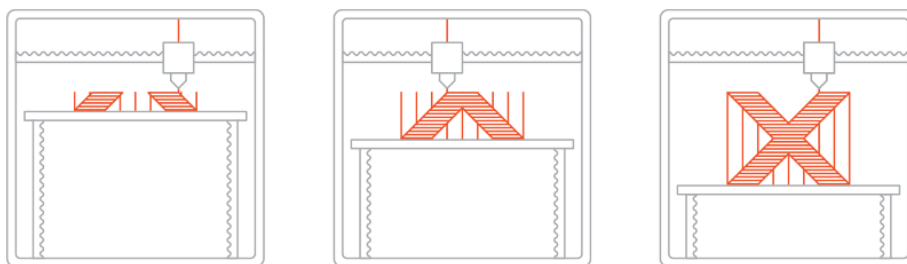
Glava za ekstruziju pričvršćena je na troosni sustav koji joj omogućuje kretanje u smjeru x, y i z osi. Otopljeni materijal se ekstrudira u tanke niti i taloži sloj po sloj na unaprijed određenim mjestima, gdje se hladi i skrućuje. Na ekstruzijsku glavu mogu se pričvrstiti ventilatori, kako bi se ubrzalo hlađenje.

Za popunjavanje područja potrebno je više prolaza. Kada je sloj gotov, platforma za izradu pomiče se prema dolje (ili u nekim postavkama stroja, glava za ekstruziju se pomiče gore) i nanosi se novi sloj. Ovaj postupak se ponavlja sve dok dio nije gotov. [8]

Svaki FDM/FFF 3D printer mora imati mogućnost micanja glave u smjeru X, Y i Z osi. Ti pomaci omogućeni su pomoću vertikalnih navoja koji pomiču glavu gore-dolje, i horizontalnih “vodilica” koje služe za pomicanje glave u smjeru X i Y osi. [9]



Slika 4.1 Dijelovi FDM printera [11]



Slika 4.2 Postupak FDM ispisa, spuštanjem podloge [12]

4.2. Softveri i postavke FDM printera

Glavni parametri FDM/FFF 3D printera su:

- temperatura 3D ispisa (za odabrani materijal),
- brzina ispisa,
- visina slojeva (rezolucija 3D ispisa),
- brzina uzimanja filameta,
- oplošje 3D modela („*shell*“, „*top*“, „*bottom*“),
- količina i oblik ispune,
- temperatura mlaznice i radne podloge,
- priprema i poravnavanje (niveliranje) radne površine.

Svi navedeni parametri utječu na kvalitetu ispisa, a oni pak ovise o tipu materijala koji se koristi te očekivanjima korisnika što se tiče kvalitete ispisa. Ukoliko je cilj što brže dobiti željeni objekt, povisuje se brzina ispisa čime se smanjuje kvaliteta. Što je brzina ispisa i visina sloja manja, to je sprintani objekt „finiji“.

Gotovo svaki FDM/FFF 3D printer korisnicima nudi podešavanje navedenih parametara uporabom softvera kojim se kontrolira 3D printer, dok se ostale postavke poput niveliranja moraju provesti ručno. [9]

4.3. Ispis tipkovnice

Nakon općenitih informacija o načinu i metodama 3D ispisa, u ovoj točki bit će detaljnije opisane postavke 3D printera koje su se koristile prilikom izrade konkretne tipkovnice. Također, bit će prikazan konačan rezultat 3D ispisa. Softver korišten za pretvorbu STL datoteka u G-kod te pripremu koda za ispis je Repetier Host, a printer korišten za 3D ispis tipkovnice je Artillery Genius.

Filament korišten za ispis tipkovnice je PLA (termoplast *Polylactic Acid* tj. polilaktična kiselina).

Postavke glavnih parametara navedenih u [točki 4.2](#) navedeni su u ovom odlomku, dok će ostatak biti prikazan u [prilogu](#) na kraju dokumenta.

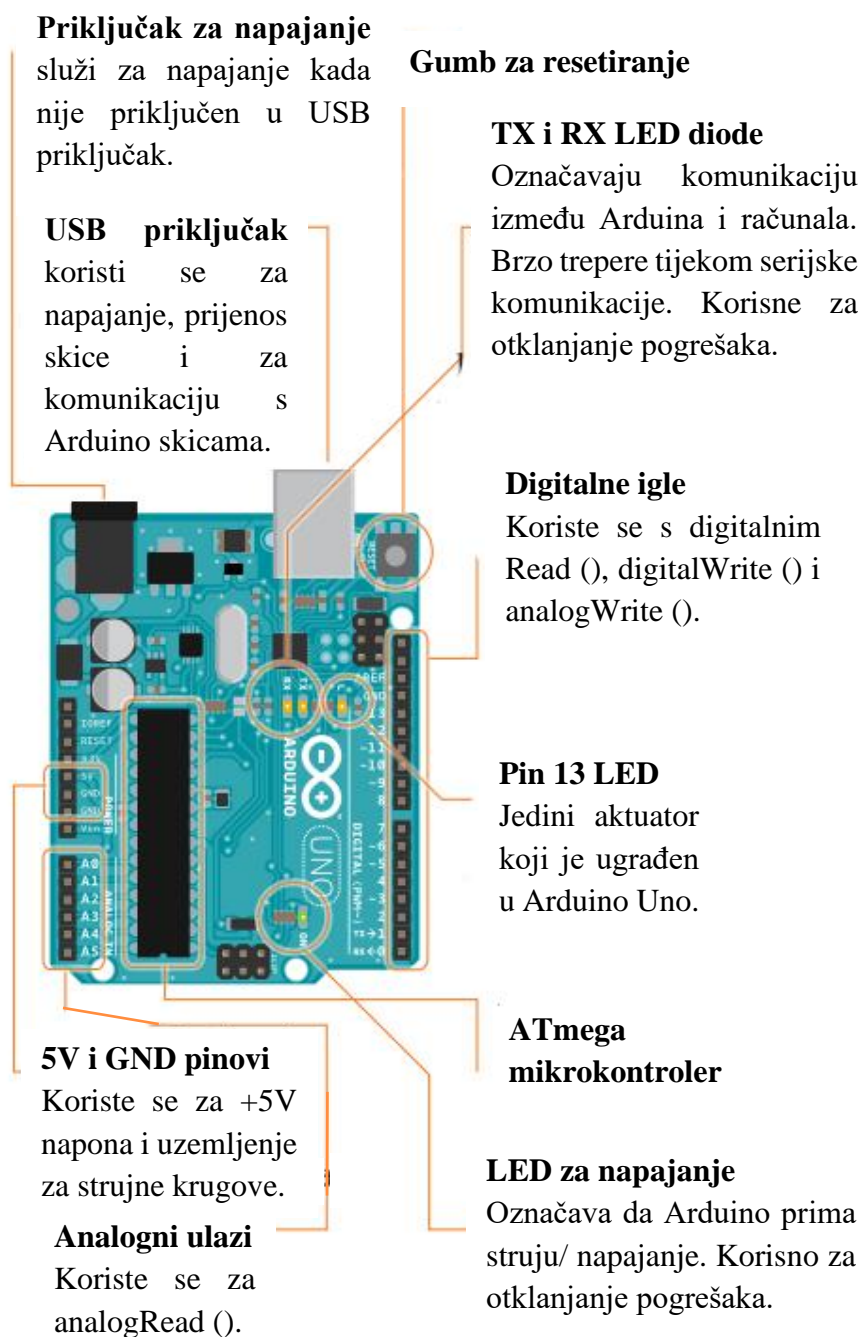
Glavni parametri korišteni za 3D ispis tipkovnice:

- brzina ispisa: 50 mm/s,
- visina sloja: 0,2 mm,
- oblik ispiune: zvijezda uzorak,
- ispuna: 20%,
- temperatura ekstruzijske glave: 195 °C,
- temperatura podloge: 70°C.

5. SPAJANJE KOMPONENTI NA ARDUINO UNO

5.1. Arduino uno ploča

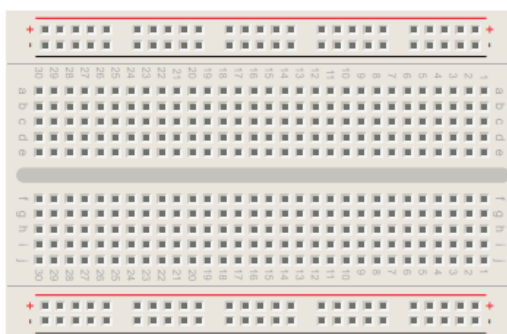
Arduino Uno je ploča mikrokontrolera otvorenog koda koja se temelji na mikrokontroleru ATmega328P. Ploča je opremljena skupovima digitalnih i analognih ulazno/izlaznih (I/O) pinova koji se mogu povezati s raznim pločama za proširenje (eng. *shieldovima*) i drugim krugovima. Na slici ispod [[Slika 5.1](#)] dodatno su opisani dijelovi Arduino Uno ploče.



Slika 5.1 Dijelovi Arduino Uno ploče [13]

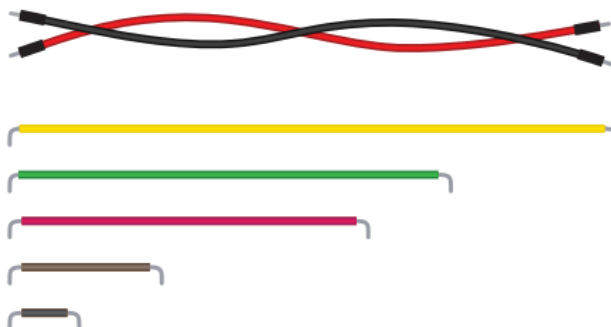
5.2. Komponente korištene za spajanje tipkovnice

- **MATIČNA PLOČA:** ploča na kojoj se grade električni sklopovi, izgleda poput panela s redovima rupa koji omogućuju spajanje žica i komponenti zajedno. Ovaj tip ploče ne zahtijeva lemljenje.



Slika 5.2 Matična ploča [13]

- **KRATKOSPOJNE ŽICE:** žice koje se koriste za međusobno povezivanje komponenti na matičnoj ploči s Arduinoom.



Slika 5.3 Kratkospojne žice [13]

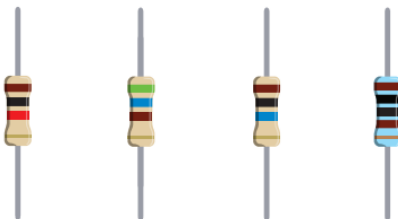
- **TIPKA:** momentni prekidači koji zatvaraju strujni krug kada se pritisnu. Vrlo lako se postavljaju na matičnu ploču pomoću „pinova“. Dobri su za otkrivanje ON/OFF signala.



Slika 5.4 Tipka [13]

- **USB (TIP A-B):** omogućuje povezivanje Arduina na računalo za programiranje te osigurava napajanje Arduina.

- **OTPORNICI:** odupiru se protoku električne energije u strujnom krugu, mijenjajući napon i struju kao rezultat. Vrijednosti otpornika se mjere u ohmima (Ω). Obojene pruge označavaju njihovu vrijednost.



Slika 5.5 Otpornici [13]

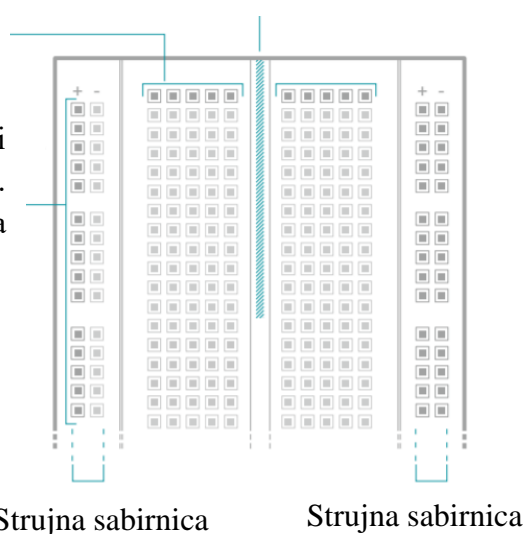
5.3. Shema spajanja dijelova za tipkovnicu

Raspored matičnih ploča postavljen je u skladu s oblikom tipkovnice i rasporedom tipki prikazanim u 3D modeliranju [Slika 3.1]. Za optimalno funkcioniranje ove tipkovnice, potrebne su 3 male matične ploče [Slika 5.2] dimenzija 5,5 cm x 8,5 cm te 2 mini matične ploče dimenzija 3,6 cm x 4,6 cm. One su međusobno povezane kratkospojnim žicama [slika 5.3] pomoću kojih se zatvara strujni krug. Crvene žice služe za stvaranje napona kroz sve matične ploče i spojene su na napon U iznosa 5 volta (V), a crne žice spojene su na uzemljenje (GND) na Arduino Uno ploči. Princip napajanja matične ploče te protoka električne energije kroz samu ploču prikazan je na slikama u nastavku [Slike 5.6 i 5.7].

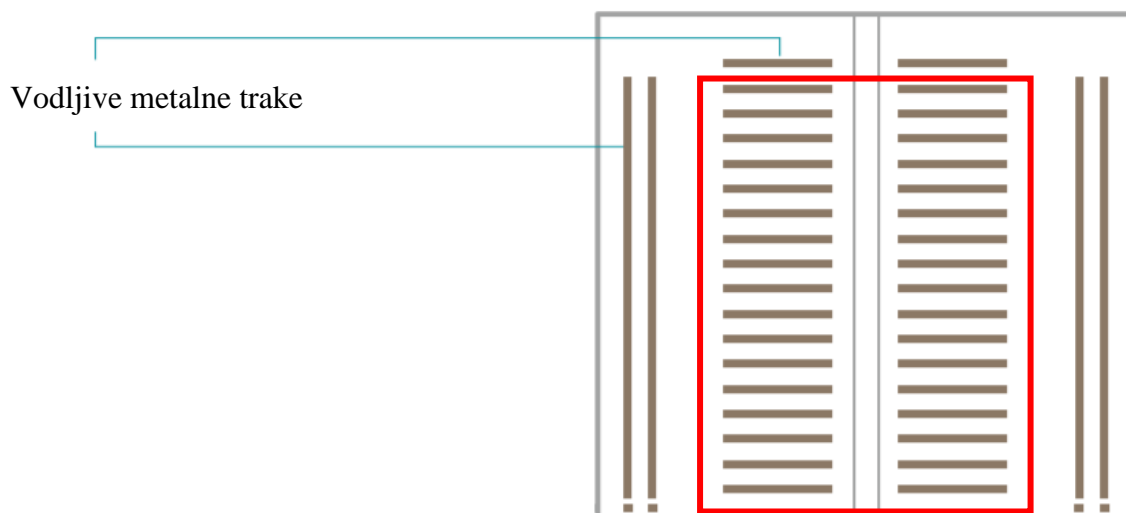
5 rupa u svakom horizontalnom redu su spojene električnim putem, pomoću metalne trake unutar matične ploče.

Okomite trake koje se protežu po dužini matične ploče su električno povezane. Obično se koriste za priključke za napajanje i uzemljenje.

Srednji red prekida povezanost između dviju strana ploča.



Slika 5.6 Matična ploča i priključci s donje strane [13]



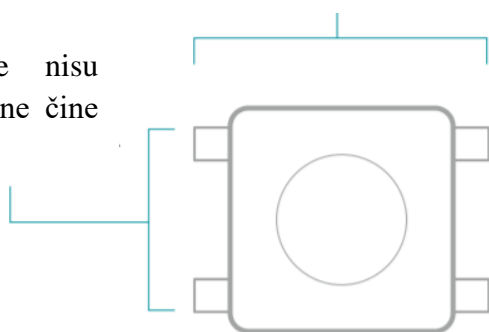
Slika 5.7 Vodljive trake unutar matične ploče [13]

Nakon povezivanja matičnih ploča na dotok električne energije, te njihovo međusobno povezivanje, slijedi spajanje tipki [Slika 5.4]. One su postavljene na samim sredinama matičnih ploča, tako da su pomoću svojih „nožica“ umetnute u rupe na matičnoj ploči. Način na koji se električna energija prenosi unutar same tipke prikazan je na slici 5.8. Mini matične ploče, koje su također korištene prilikom izrade ove tipkovnice, funkcioniraju kao središnji dio matične ploče prikazane na slici 5.7 (dio uokviren crvenom bojom).

SKLOPNE VEZE

Ove dvije „nožice“ prekidača su spojene jedna s drugom.

Ove dvije nisu spojene. One čine tipku.



SHEMATSKI IZGLED

SKLOPKE



A - Simbol prekidača

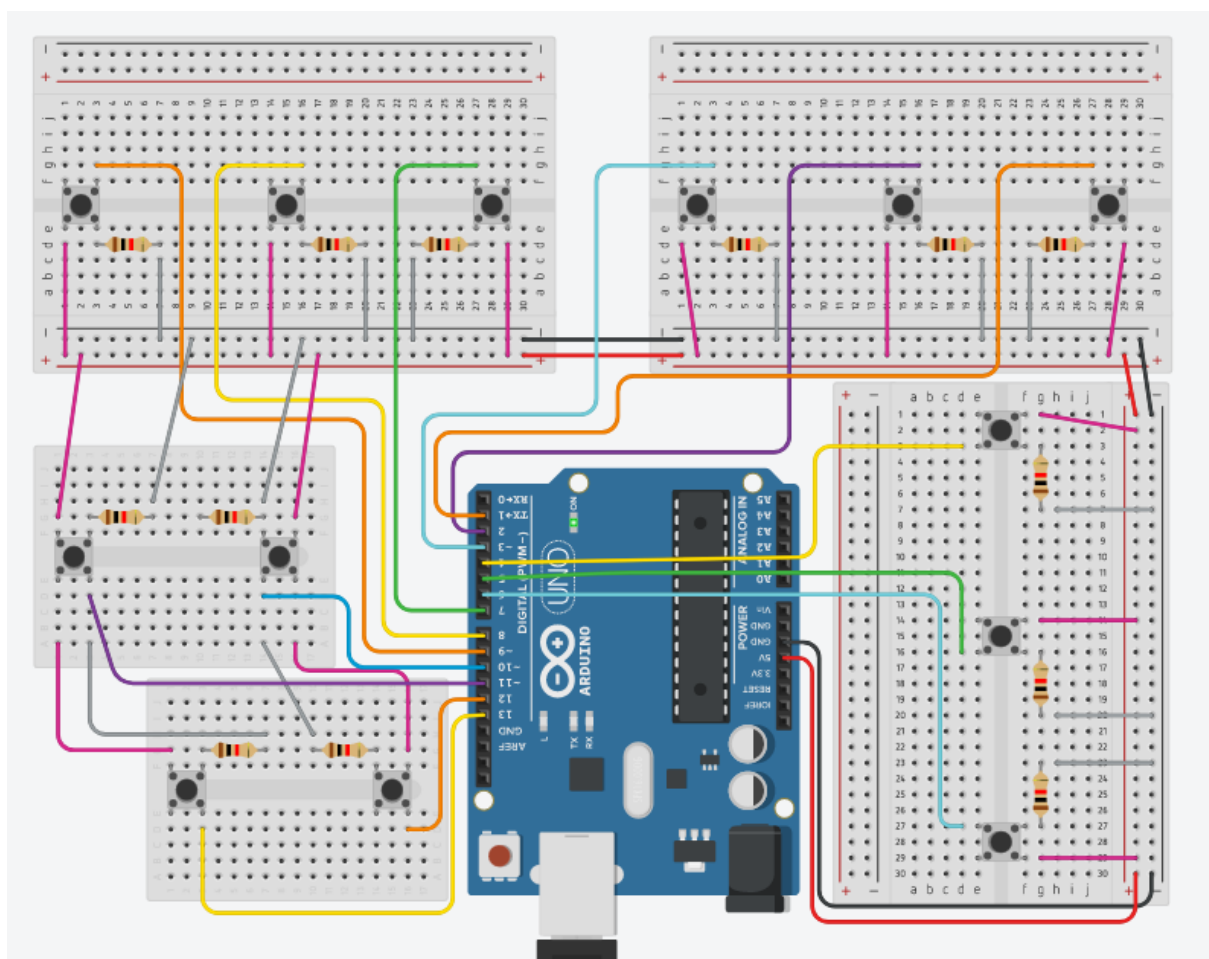


B - Simbol tipke

Slika 5.8 Funkcioniranje tipke [13]

Nakon umetanja tipki, potrebno je same tipke povezati na dotok električne energije te na određene digitalne pinove. Pomoću njih će se kasnije programirati svaka tipka zasebno, ovisno o tome što određena tipka predstavlja. Arduino digitalni (binarni) pinovi mogu čitati samo dva stanja: kada postoji napon na pinu i kada ga nema. Digitalni pinovi mogu djelovati kao ulazi ili izlazi, ovisno o tome za što nam trebaju. U slučaju izrade ove tipkovnice, oni služe kao inputi (ulazi). Igle prekidača koje su međusobno povezane jedna s drugom, s jedne strane se spajaju na digitalni pin preko kratkospojne žice, a s druge strane se preko otpornika spajaju na uzemljenje. Otpornici korišteni za izradu ove tipkovnice su otpora R iznosa $1000\ \Omega$. Nasuprotna igla, koja nije povezana s iglom za uzemljenje, spaja se na napon (redak označen oznakom +).

Shema spajanja dijelova prikazana je pomoću Tinkercad-a, besplatne online zbirke softverskih alata za 3D dizajn, elektroniku i kodiranje. Shema je prikazana na slici 5.9.



Slika 5.9 Shema napajanja tipkovnice

6. PROGRAMIRANJE ARDUINO MIKROKONTROLERA

Integrirano razvojno okruženje Arduino IDE (Integrated Development Environment) je poseban softver odnosno glavni program za uređivanje teksta koji se koristi za Arduino programiranje. To je mjesto gdje se upisuje kod (skica) prije nego što se učita na Arduino ploču koju želite programirati. Kreirana „skica“ (naziv koji je dat datotekama Arduino koda, eng. *sketch*) se obrađuje i prevodi u strojni jezik.

Programski jezik Arduino temelji se na vrlo jednostavnom hardverskom programskom jeziku zvanom *processing* (hrv. obrada), koji je sličan jeziku C.

Arduino IDE podržava jezike C i C++ koristeći posebna pravila strukturiranja koda uz dodatak posebnih metoda i funkcija.

U Arduino, poput ostalih vodećih programskih platformi, postoje ugrađene biblioteke koje pružaju osnovnu funkcionalnost. Osim toga, moguće je uvesti druge biblioteke i proširiti mogućnosti i značajke Arduino ploče. Te su knjižnice grubo podijeljene na biblioteke koje su u interakciji s određenom komponentom ili one koje implementiraju nove funkcije.

6.1. Struktura Arduino programskog jezika

Arduino programski jezik može se podijeliti u tri glavna dijela: funkcije, vrijednosti (varijable i konstante) i struktura. Funkcije se koriste za upravljanje Arduino pločom i izvođenje računanja (npr. `DigitalRead ()`, `digitalWrite ()`, `pinMode ()`, `delay ()`). Vrijednosti su Arduino tipovi podataka i konstante (npr. `HIGH | LOW`, `INPUT | OUTPUT`, `const`, `int ()`). Struktura su elementi Arduino (C++) koda kao što su `setup ()`, `loop ()`, `if`, `while`. [14]

Osnovna logika koda Arduino je struktura "if - then" i može se podijeliti u 4 bloka:

- **SETUP** - obično se zapisuje u odjeljku za postavljanje Arduino koda i izvodi stvari koje je potrebno učiniti samo jednom.
- **INPUT** - na početku petlje čitaju se ulazi. Ove vrijednosti će se koristiti kao uvjeti ("if").
- **MANIPULIRANJE PODACIMA** - ovaj se odjeljak koristi za pretvaranje podataka u prikladniji oblik ili za izračune.
- **OUTPUT** - ovaj odjeljak definira konačni ishod logike ("then") prema podacima izračunatim u prethodnom koraku. [15]

6.2. Arduino kod za tipkovnicu

- `uint8_t buf[8] = { 0 }; // Keyboard report buffer //`

Programiranje tipkovnice započinje korištenjem naredbe `uint8_t buf [8]` čime se inicijalizira „međuspremnik“ tipkovnice koji je potreban da bi Arduino USB HID tipkovnica bila implementirana putem USB-a koji povezuje računalo i Arduino.

Buffer tj međuspremnik zapravo označava niz koji ima 8 elemenata (članova); svaki član sadrži 8-bitnu neoznačenu količinu. Format *buffera* je sljedeći:

[0 = modifier, 1 = reserved, 2 = Key1, 3 = Key2, 4 = Key3, 5 = Key4, 6 = Key5, 7 = Key6]

- `#define PIN_1 9`
`#define PIN_2 8`

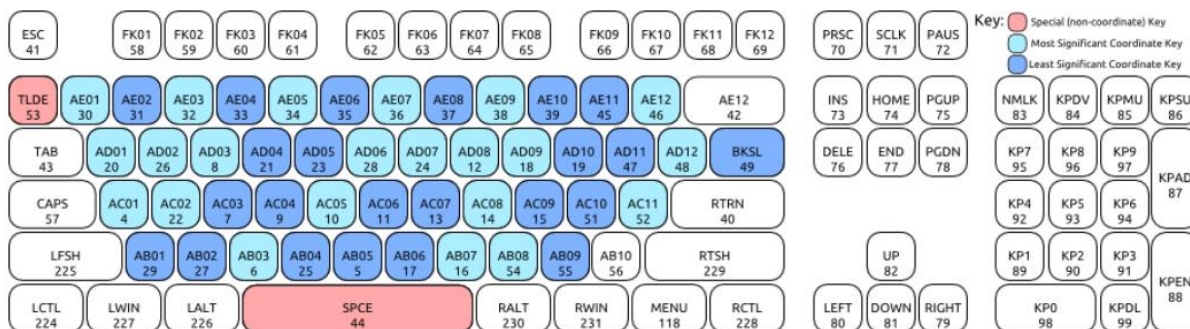
`#define` je komponenta C++ pomoću koje se dodjeljuje naziv konstantnoj vrijednosti prije nego što se program prevede te je definiran (digitalni) pin (brojevi 9, 8 iz primjera) koji će pripasti svakoj od vrijednosti `PIN_X`.

- `void setup()`
`{`
`Serial.begin(9600);`
`pinMode(PIN_1,INPUT);`
`pinMode(PIN_2, INPUT);`
`}`

Funkcija `setup()` se poziva kada skica započne. Koristi se za inicijalizaciju varijabli, načina rada pin-a itd. Funkcija `setup()` će se pokrenuti samo jednom, nakon svakog uključivanja ili resetiranja Arduino ploče. `Serial.begin (9600)` postavlja brzinu prijenosa podataka u bitovima u sekundi za serijski prijenos podataka između Arduina i računala. Funkcija `pinMode` konfigurira navedeni pin da se ponaša kao ulaz.

- `void loop()`
`{`
`if(digitalRead(PIN_1) == HIGH) {`
`buf[2] = 30;`
`Serial.write(buf,8);`
`releaseKey();`
`delay (1000);`
`}`
`}`

Funkcija `loop()` koristi se za aktivnu kontrolu Arduino ploče. Ovaj korak se obavlja u funkciji `Loop()` kako bi se ažurirao svaki ciklus kroz koji Arduino prolazi provjeravajući je li tipka pritisnuta. Pomoću funkcije `digitalRead` čita se vrijednost sa specificiranog pina (high/low). `Buf[2] = 30` označava tipku 1 na tipkovnici, dakle, svaka tipka ima svoju šifru pomoću koje se definira. Sve šifre prikazane su na slici 6.1.



Slika 6.1 Šifre za programiranje tipke pomoću buffera [16]

Na isti način je definirano preostalih 12 tipki.

- `void releaseKey()`

```

{
  buf[0] = 0;
  buf[2] = 0;
  Serial.write(buf, 8);
}

```

Ovaj korak je potreban za završetak protoka podataka HID tipkovnice. Za ovaj primjer šaljem šifru ključa putem `buf[2]` pa ih moramo resetirati na 0 kada se otpusti tipka na Arduino. Bez ovog koraka HID tipkovnica može početi slati unose gumba, ali nikada neće prestati slati bitove dok se USB HID tipkovnica (Arduino) ne isključi.

7. SPAJANJE HID TIPKOVNICE KAO USB PRIKLJUČKA

Human Interface Devices (ili HID-ovi) su klasa perifernih uređaja pomoću kojih se unose naredbe. To je izraz koji se obično koristi za miševe i tipkovnice.

HID tipkovnice se ne razlikuju od standardnih tipkovnica u prijenosnim računalima, osim što su spojene preko USB-a, a ne ugrađene.

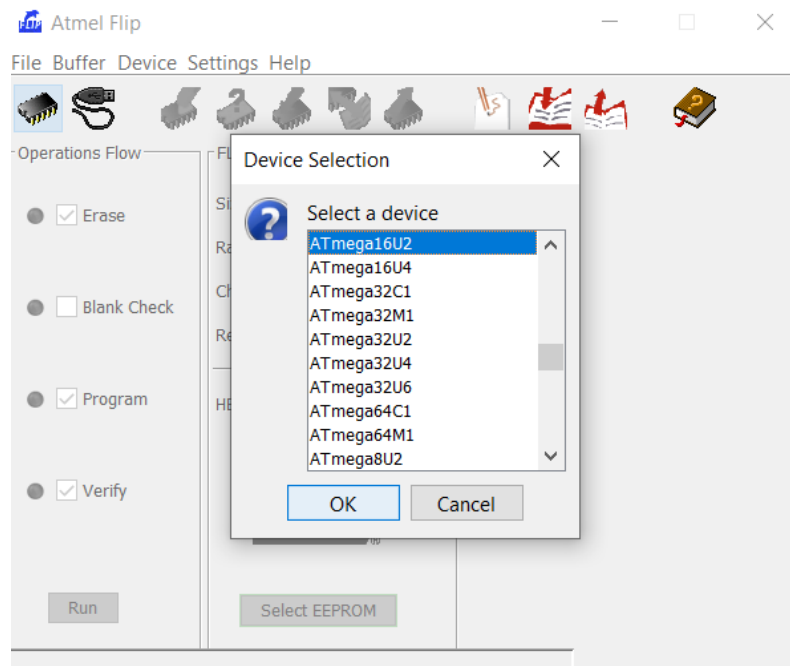
Nakon svih odrađenih koraka, potrebno je još objasniti kako pretvoriti Arduino čip u HID uređaj s tipkovnicom tj. kako da naše računalo putem USB priključka prepozna Arduino Uno kao HID tipkovnicu.

- Postupak kreće od instaliranja Flip 3.4.7, softvera koji podržava programiranje tzv. *flashanje* uređaja u sustavu putem RS232, USB-a ili CAN-a. Dostupne su dvije instalacijske datoteke: jedna s integriranim Java Run-time Environment (JRE) i jedna bez.
- Nakon instalacije Flip 3.4.7, Arduino se spaja pomoću USB kabela na računalo te se zatim resetira Arduino Uno ploča. Resetiranje se vrši spajanjem pinova označenih crvenom bojom [slika 7.1]. Arduino se resetira te prelazi u oblik za *fleshanje* čime nestaje iz popisa uređaja u Device Manageru i u njegovom nativnom softveru tj. IDE-u – Arduino IDE 1.8.16, ali je sada vidljiv FLIP-u.

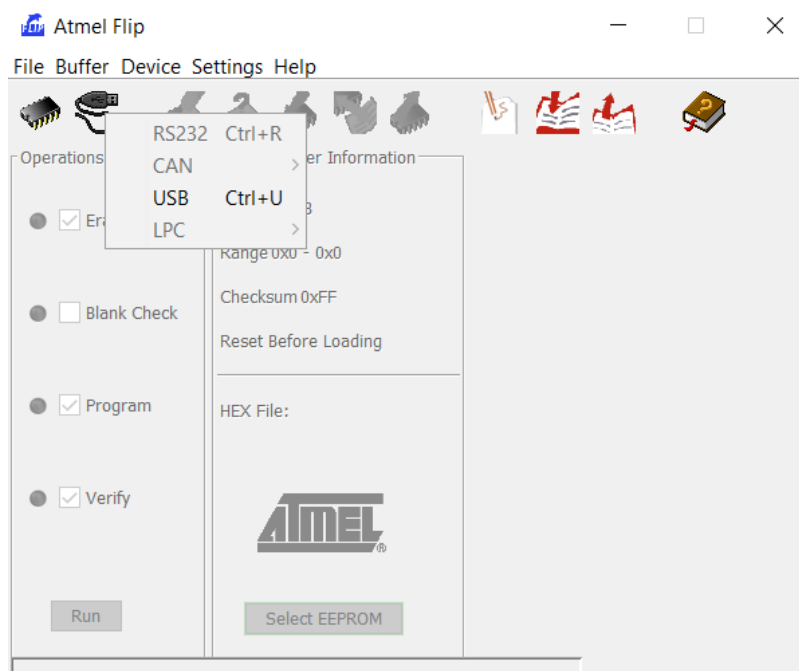


Slika 7.1 Resetiranje Arduino Uno ploče

- U FLIP-u se odabire uređaj Atmega16U2 te se za sredstvo komunikacije odabire USB ulaz.

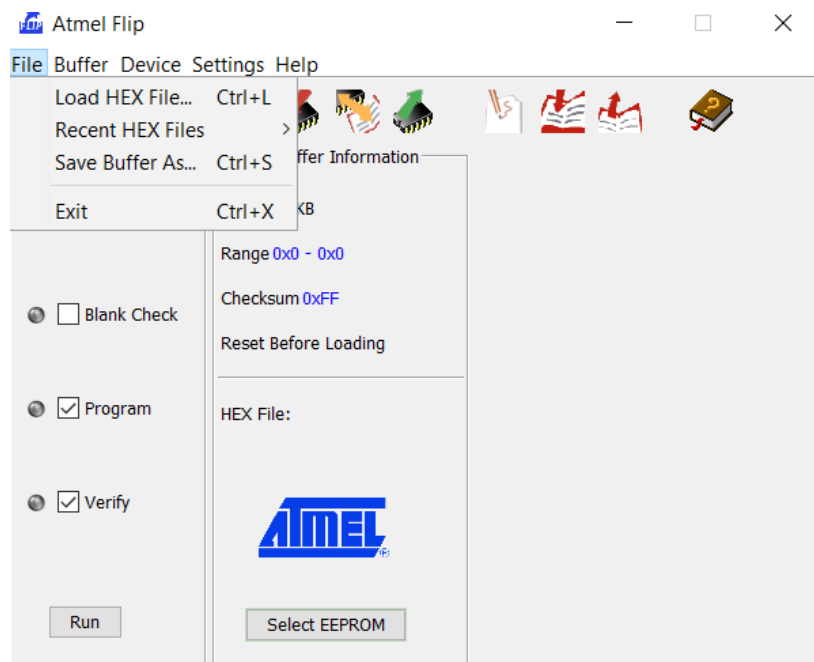


Slika 7.2 Odabir uređaja



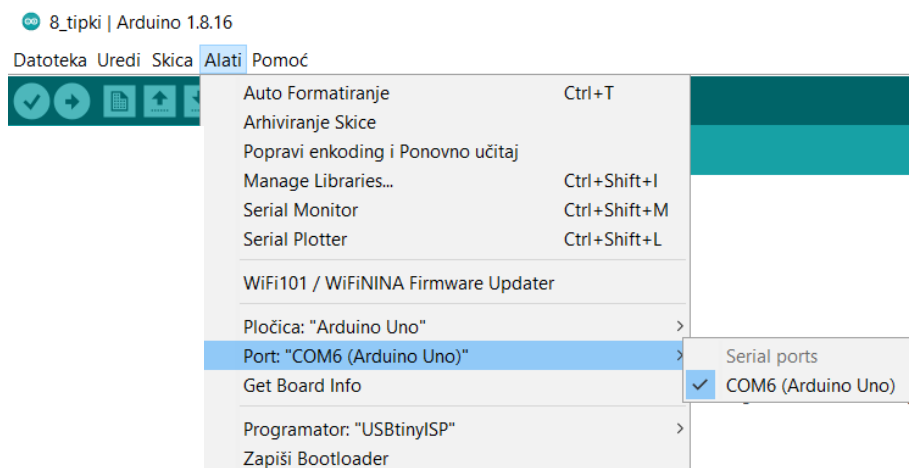
Slika 7.3 Odabir ulaza

- U FLIP-u se zatim učita (File- Load HEX file) Arduino-usbserial-uno.hex datoteka te se pokreće (Run) *flashanje*.



Slika 7.4 Učitavanje HEX datoteke

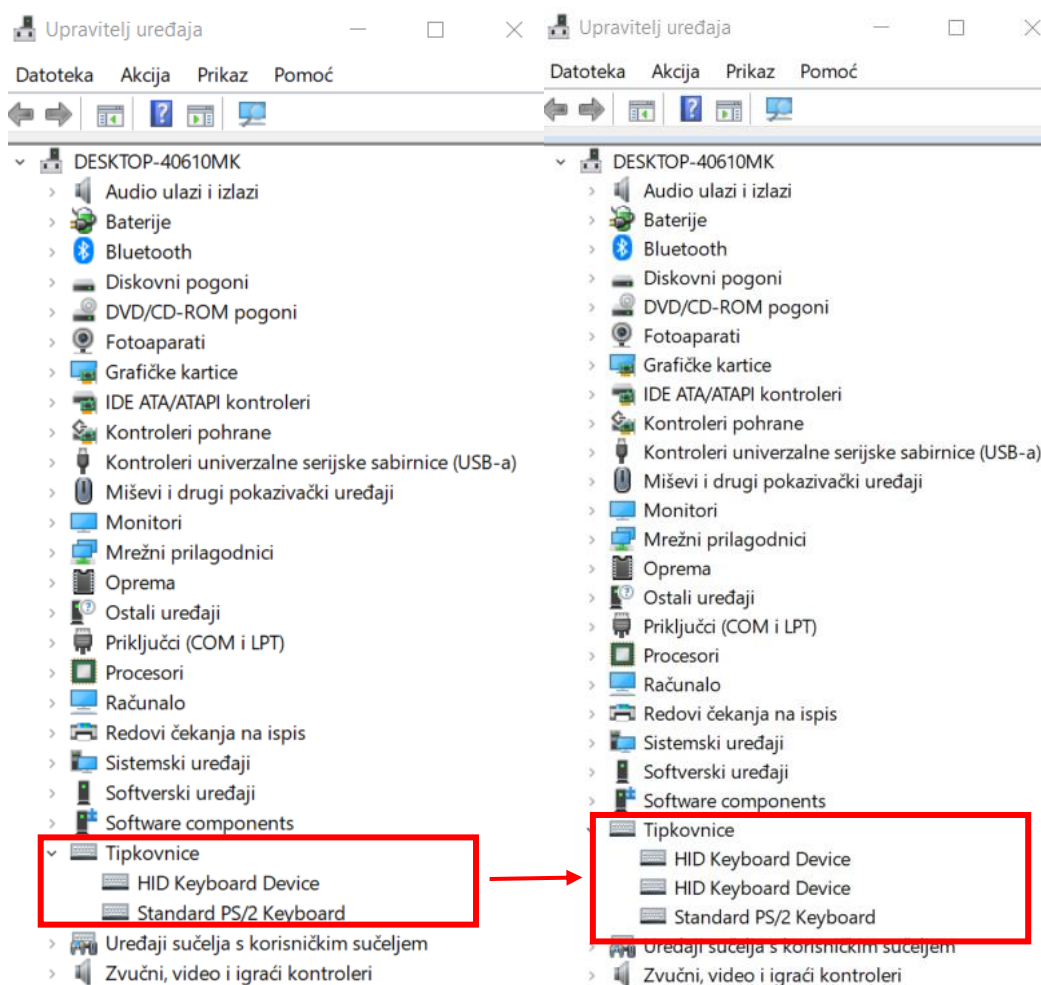
- Nakon ovog koraka Arduino se isključuje iz računala te se priključi nazad nakon par sekundi. Sada bi Arduino Uno trebao ponovo biti vidljiv u popisu uređaja u svom IDE-u.



Slika 7.5 Vidljivost Arduino Uno Porta na popisu uređaja

- U Arduino IDE-u ispisuje se kod kojim se želi programirati tipkovnica (detaljno objašnjen u [poglavlju 6](#) ovoga rada) te se učita (prenese) na željenu Arduino ploču koja je u ovom slučaju Arduino Uno.

- Idući korak je ponovo resetiranje Arduino ploče pomoću već prethodno navedena dva pina [slika 7.1].
- Zatim se ponovo ulazi u Flip te učitava Arduino-keyboard-0.3.hex datoteku i ponovo pokreće (Run) *flashanje*.
- Na kraju bi Arduino trebao biti prepoznat kao HID USB Keyboard te je kao takav spreman za korištenje.



Slika 7.6 Prikaz HID tipkovnice u Upravitelju uređaja

Na slici 7.6. prikazan je Upravitelj uređaja te se vidi kako nakon uključivanja tipkovnice pomoću USB kabela, računalo prepoznaje tipkovnicu kako HID tipkovnicu te je ona spremna za korištenje.

8. ZAKLJUČAK

U ovom radu u potpunosti je prikazan postupak razvoja tipkovnice za pomoć osobi s invaliditetom. Temelji se na jednostavnoj metodi 3D modeliranja, 3D ispisa te praktičnoj i jeftinoj platformi za programiranje i napajanje – Arduino.

U današnje vrijeme, kada sve veći dio populacije uživa u igranju video igara, potrebno je ostvariti ravnopravnost i jednaku pristupačnost kako za osobe bez invaliditeta, tako i onima sa određenim invaliditetom.

Tvorci igračih konzola dizajniraju proizvode za masovnu proizvodnju što predstavlja problem za osobe s invaliditetom. Obzirom da se osobama s invaliditetom uglavnom pristupa individualno te postoji mnogo vrsta invaliditeta, jasno je da je ovaj dio industrije tek u razvojnoj fazi. Isto tako, u današnje vrijeme postoji i velik opseg videoigara na tržištu što dodatno komplicira razvoj ovakve vrste proizvoda.

Sam proces osmišljavanja, konstruiranja i realiziranja ovakvog proizvoda je znatno kompleksniji. Također, kada se takav proizvod i osmisli i na kraju plasira na tržište, najčešće je cjenovno nedostupan ciljanom tržištu. Danas postoji i nekoliko humanitarnih organizacija koje pomoću donacija potpomažu kupovinu takvih pomagala za osobe s invaliditetom.

Ovim radom pokazano je koji su sve koraci u razvitku ovakvoga proizvoda te da se pomoću nekoliko jednostavnih metoda može osmisliti i realizirati jedan takav proizvod.

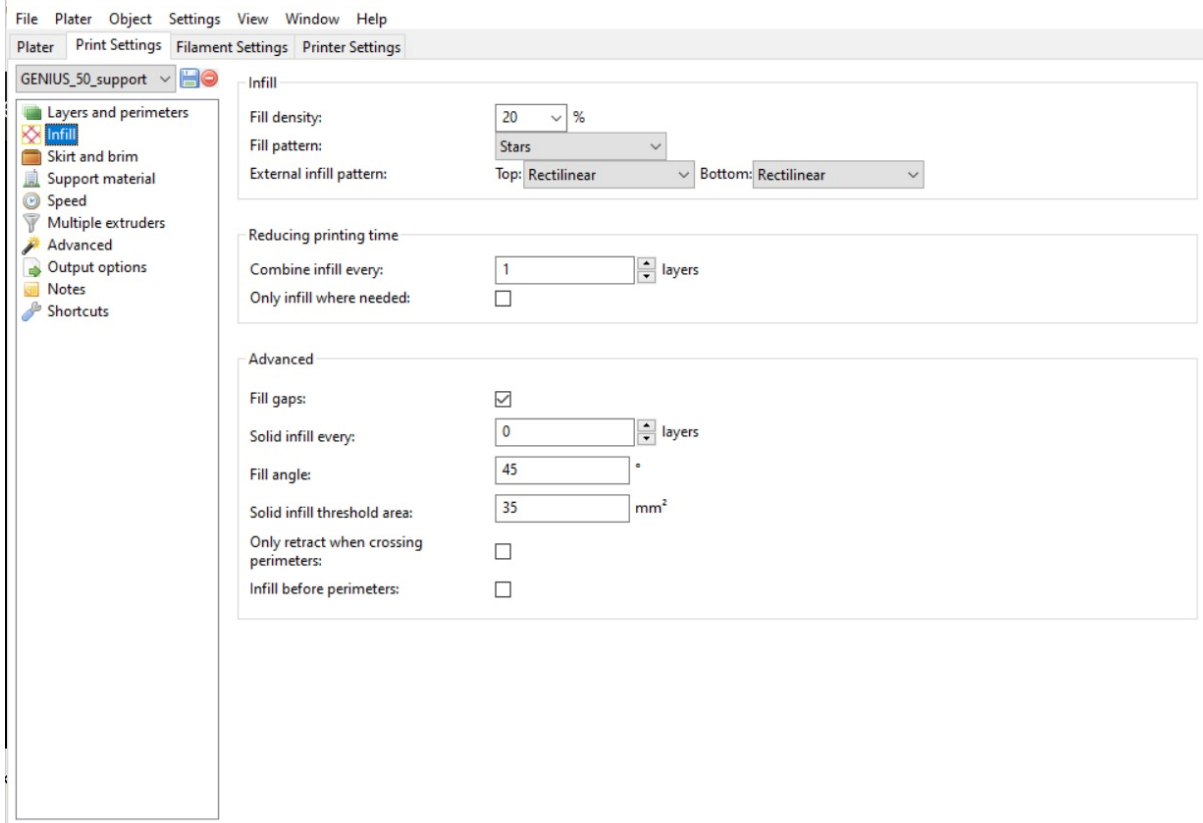
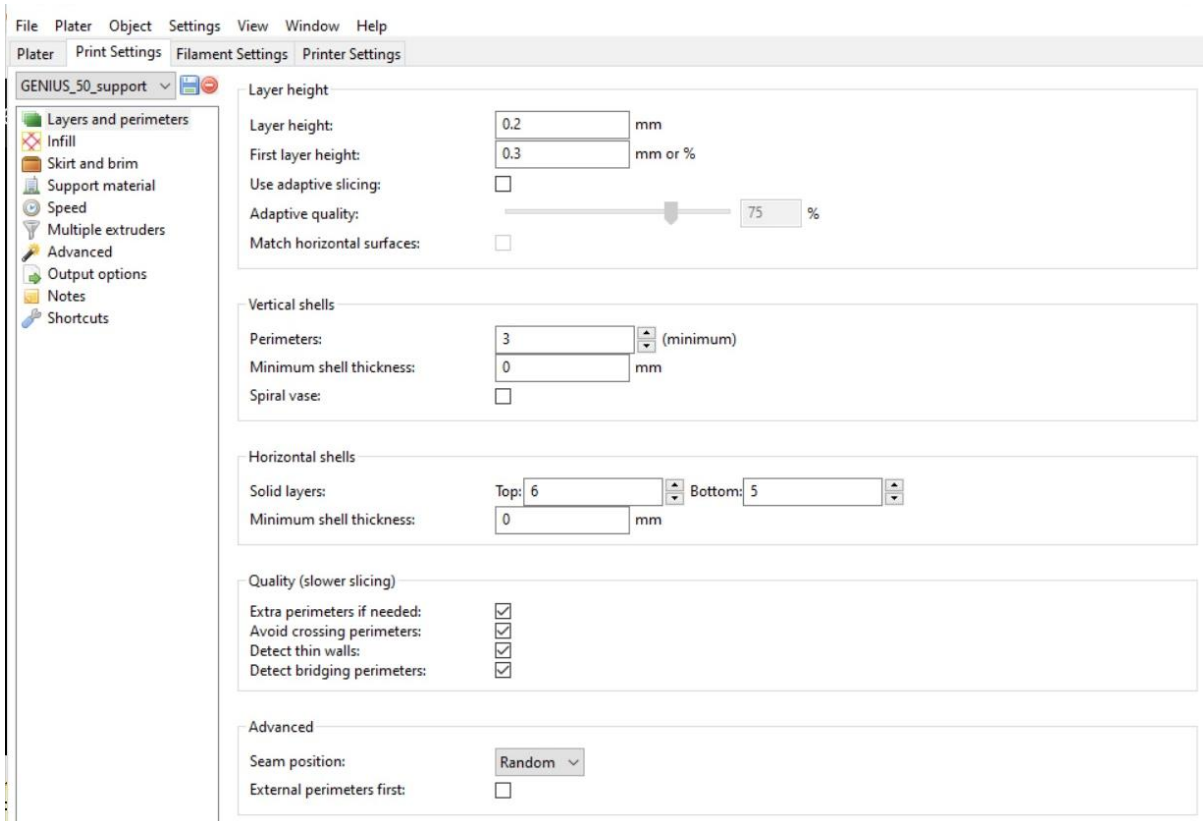
LITERATURA

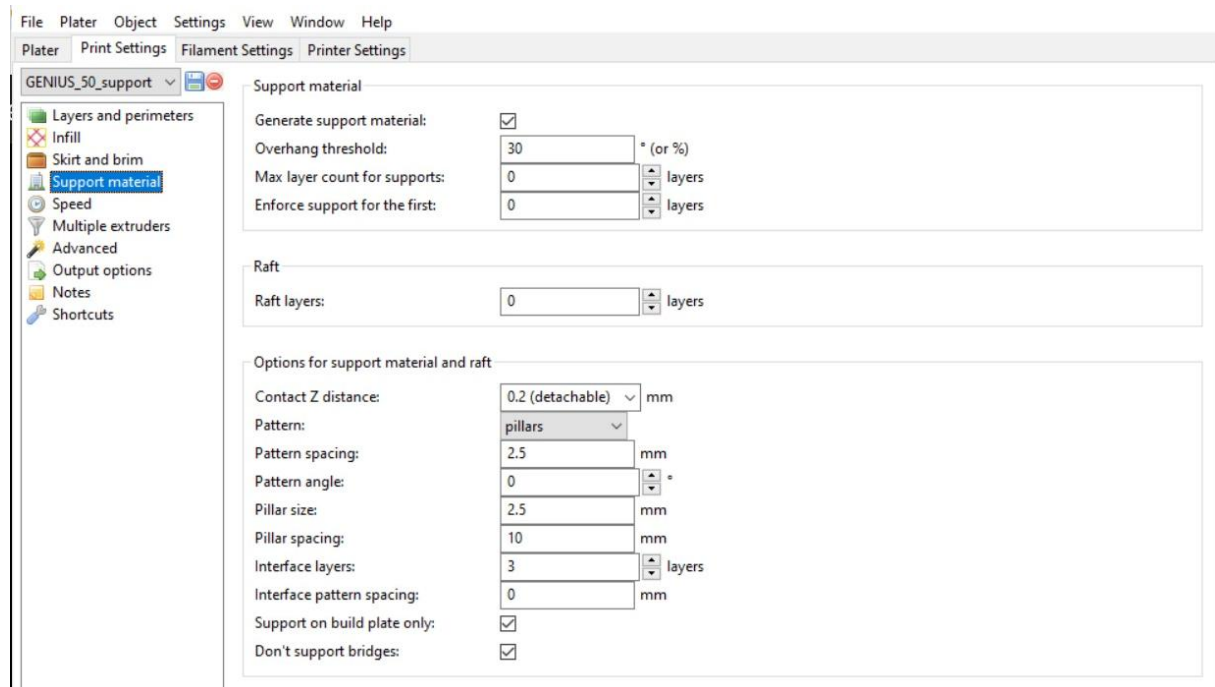
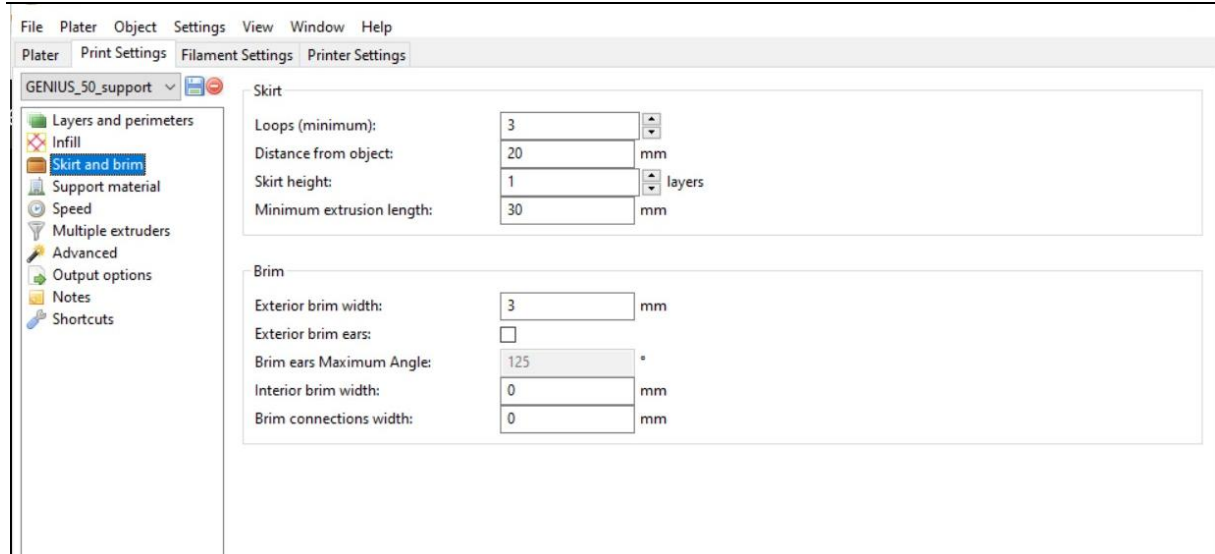
- [1] https://www.who.int/disabilities/world_report/2011/report.pdf
- [2] <https://medium.com/@noahlandonpaige/accessibility-in-video-games-63bcb25ef79f>
- [3] <https://hobgear.com/understand-keyboard-sizes/>
- [4] <https://www.popularmechanics.com/technology/design/a18550/the-science-of-making-a-great-keyboard/>
- [5] <https://cougargaming.com/products/keyboards/puri-rgb/>
- [6] <https://www.daskeyboard.com/blog/mechanical-keyboard-guide/#common-mechanical-keyboard-switches>
- [7] Karl – Heinz Decker: Elementi strojeva, Tehnička knjiga, Zagreb, 2006.
- [8] <https://www.hubs.com/knowledge-base/what-is-fdm-3d-printing/>
- [9] <https://www.3dglobe.net/fdmfff>
- [10] <https://themechninja.com/07/fdm-printing-advantages-disadvantages-detailed/>
- [11] <https://www.3dprintaj.com/fdm-fff-tehnologija-3d-printanja/>
- [12] <https://tobuya3dprinter.com/types-of-3d-printing-fdm-sla-and-dlp/>
- [13] <https://www.uio.no/studier/emner/matnat/ifi/IN1060/v21/arduino/arduino-projects-book.pdf>
- [14] <https://www.arduino.cc/reference/en/>
- [15] <https://www.circuito.io/blog/arduino-code/>
- [16] <https://github.com/Robotechnic/DigiKeyboardFr>

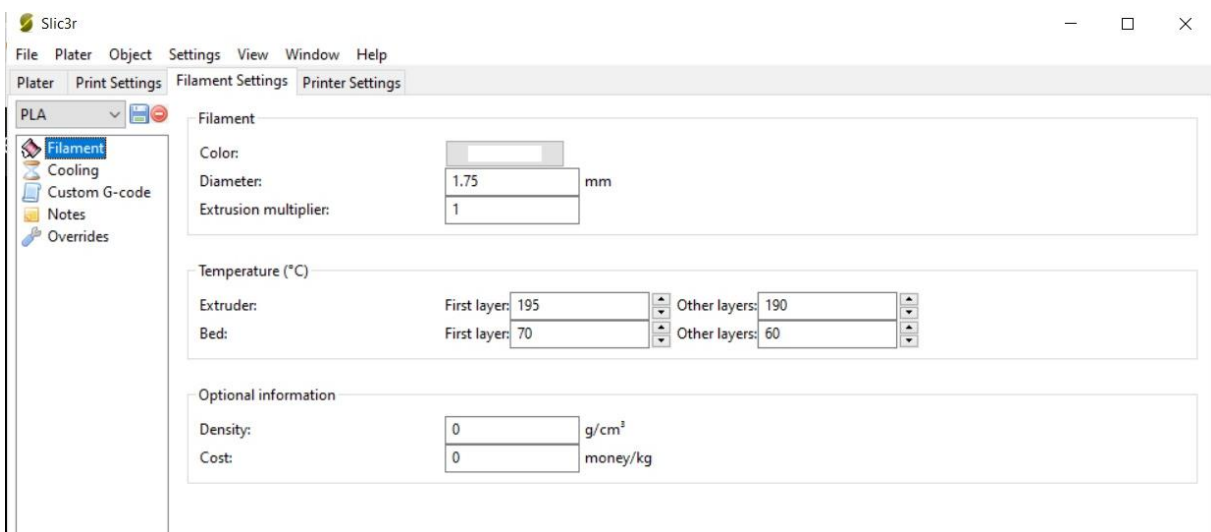
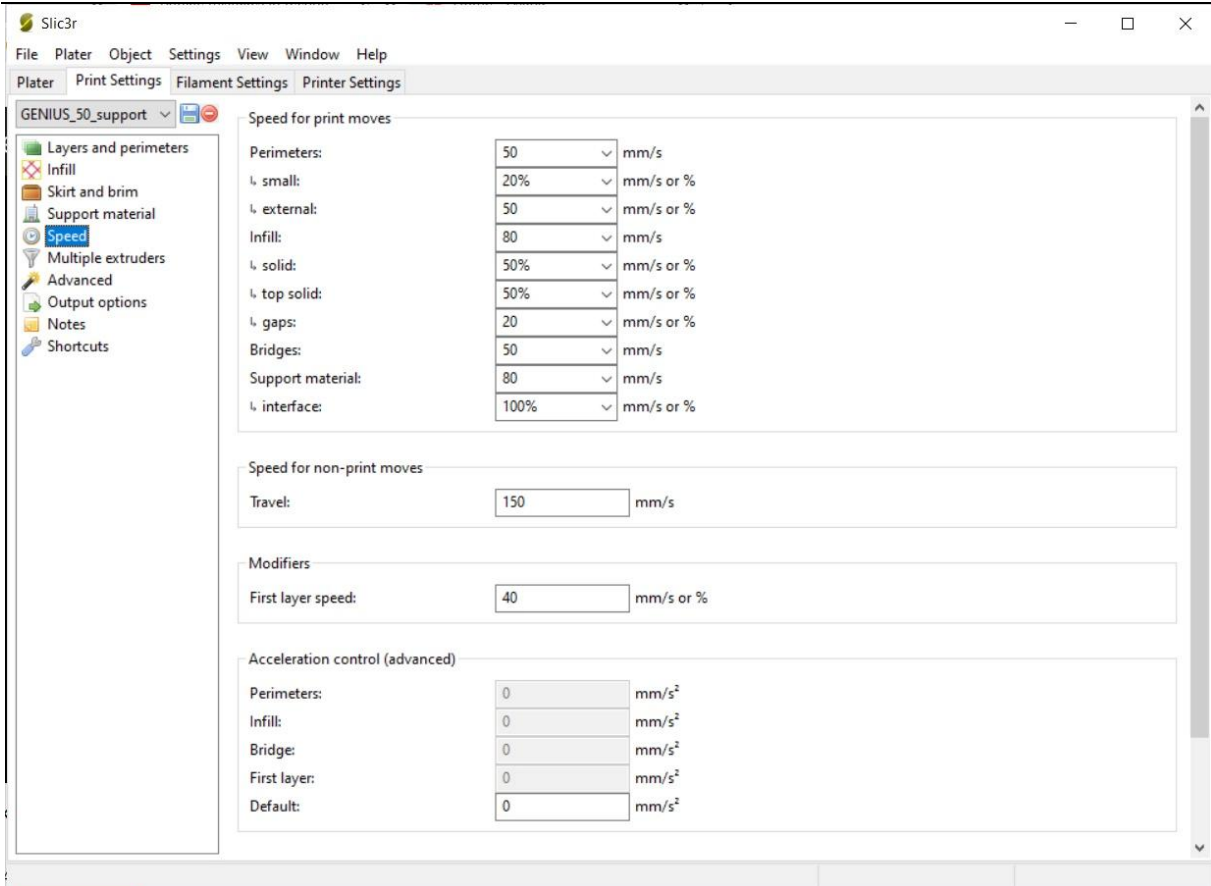
PRILOZI

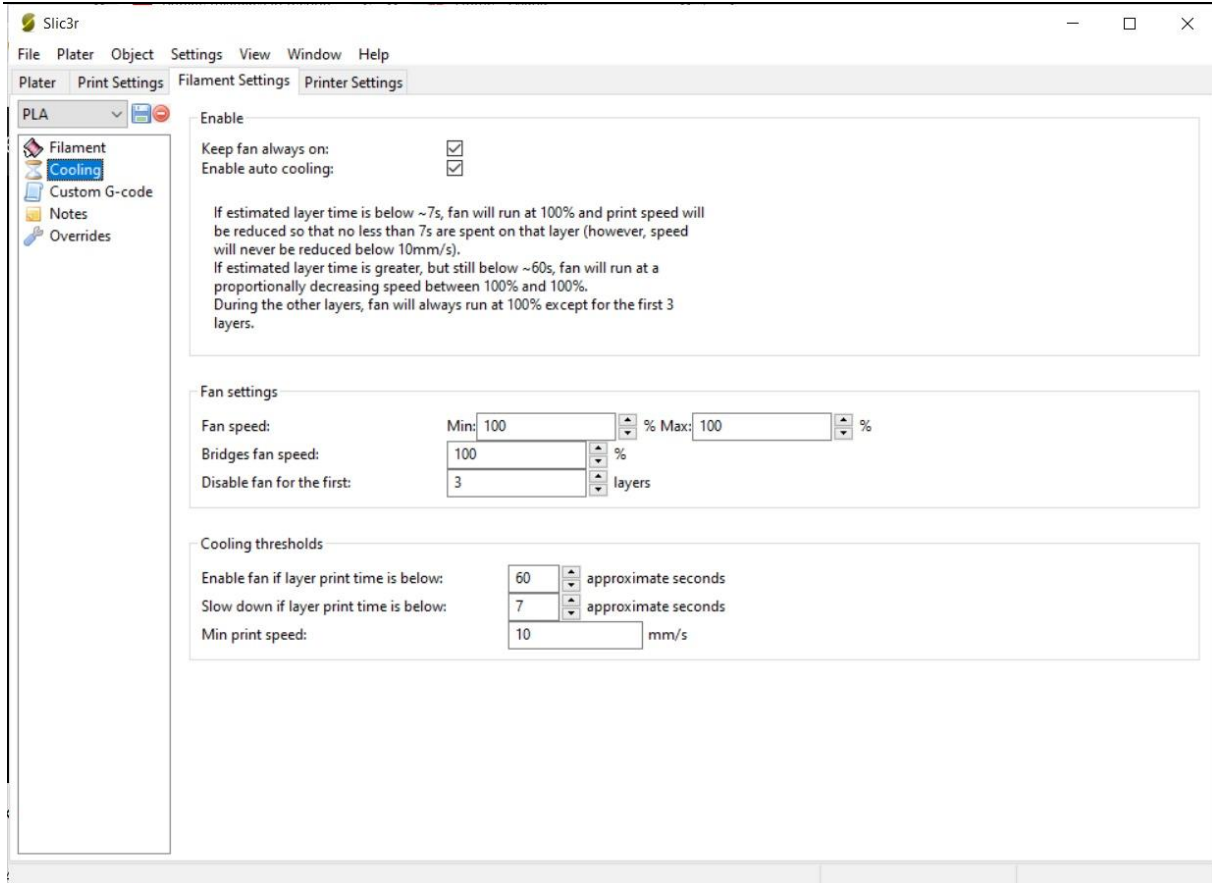
1. Parametri 3D ispisa
2. Arduino Uno programski kod

1. Parametri 3D ispisa









2. Arduino Uno programski kod

```
uint8_t buf[8] = { 0 };

#define PIN_1 9 // dodijeli naziv konstantnoj vrijednosti prije nego što se program prevede
#define PIN_2 8
#define PIN_3 7
#define PIN_4 3
#define PIN_6 2
#define PIN_7 1
#define PIN_5 4
#define PIN_D 5
#define PIN_F 6
#define PIN_Q 10
#define PIN_W 11
#define PIN_E 13
#define PIN_R 12

void setup()
{
    Serial.begin(9600); //postavljanje brzine prijenosa i serijske komunikacije s funkcijom
    pinMode(PIN_1, INPUT); // Konfigurira navedeni pin da se ponaša kao ulaz ili izlaz.
    pinMode(PIN_2, INPUT);
    pinMode(PIN_3, INPUT);
    pinMode(PIN_4, INPUT);
    pinMode(PIN_6, INPUT);
    pinMode(PIN_7, INPUT);
    pinMode(PIN_5, INPUT);
    pinMode(PIN_D, INPUT);
    pinMode(PIN_F, INPUT);
    pinMode(PIN_Q, INPUT);
    pinMode(PIN_W, INPUT);
    pinMode(PIN_E, INPUT);
    pinMode(PIN_R, INPUT);

    delay (1000);
}

void loop()
{
    if(digitalRead(PIN_1) == HIGH) { // čita vrijednost sa specificiranog pina (high/low)
        buf[2] = 30; // BROJ 1
        Serial.write(buf,8);
        releaseKey();
        delay (1000);
    }
    if(digitalRead(PIN_2) == HIGH) {
        buf[2] = 31; // BROJ 2
        Serial.write(buf,8);
        releaseKey();
        delay (1000);
    }
    if(digitalRead(PIN_3) == HIGH) {
        buf[2] = 32; // BROJ 3
        Serial.write(buf,8);
        releaseKey();
        delay (1000);
    }
}
```

```
if(digitalRead(PIN_4) == HIGH) {
    buf[2] = 33; // BROJ 4
    Serial.write(buf,8);
    releaseKey();
    delay (1000);
}
if(digitalRead(PIN_5) == HIGH) {
    buf[2] = 34; // BROJ 5
    Serial.write(buf,8);
    releaseKey();
    delay (1000);
}
if(digitalRead(PIN_6) == HIGH) {
    buf[2] = 35; // BROJ 6
    Serial.write(buf,8);
    releaseKey();
    delay (1000);
}
if(digitalRead(PIN_7) == HIGH) {
    buf[2] = 36; // BROJ 7
    Serial.write(buf,8);
    releaseKey();
    delay (1000);
}
if(digitalRead(PIN_D) == HIGH) {
    buf[2] = 7; // SLOVO d
    Serial.write(buf,8);
    releaseKey();
    delay (1000);
}
if(digitalRead(PIN_F) == HIGH) {
    buf[2] = 9; // SLOVO f
    Serial.write(buf,8);
    releaseKey();
    delay (1000);
}
if(digitalRead(PIN_Q) == HIGH) {
    buf[2] = 20; // SLOVO q
    Serial.write(buf,8);
    releaseKey();
    delay (1000);
}
if(digitalRead(PIN_W) == HIGH) {
    buf[2] = 26; // SLOVO w
    Serial.write(buf,8);
    releaseKey();
    delay (1000);
}
if(digitalRead(PIN_E) == HIGH) {
    buf[2] = 8; // SLOVO e
    Serial.write(buf,8);
    releaseKey();
    delay (1000);
}
```

```
if(digitalRead(PIN_R) == HIGH) {
    buf[2] = 21;           // SLOVO r
    Serial.write(buf, 8);
    releaseKey();
    delay (1000);
}

}

void releaseKey() {
    buf[0] = 0;
    buf[2] = 0;
    Serial.write(buf, 8);
}

}
```