

Dekoder Morse-ove abecede

Birtić, Fran

Undergraduate thesis / Završni rad

2021

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://um.nsk.hr/um:nbn:hr:200:533152>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-07-13**

Repository / Repozitorij:

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



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

Stručni studij

DEKODER MORSE-OVE ABECEDA

Završni rad

Fran Birtić

Osijek, 2021. godina.

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac Z1S: Obrazac za imenovanje Povjerenstva za završni ispit na preddiplomskom stručnom studiju

Osijek, 31.08.2021.

Odboru za završne i diplomske ispite

**Imenovanje Povjerenstva za završni ispit
na preddiplomskom stručnom studiju**

| | |
|--|---|
| Ime i prezime studenta: | Fran Birtić |
| Studij, smjer: | Preddiplomski stručni studij Računarstvo |
| Mat. br. studenta, godina upisa: | AI 4592, 09.10.2017. |
| OIB studenta: | 05508047008 |
| Mentor: | Izv. prof. dr. sc. Davor Vinko |
| Sumentor: | |
| Sumentor iz tvrtke: | |
| Predsjednik Povjerenstva: | Izv.prof.dr.sc. Tomislav Matić |
| Član Povjerenstva 1: | Izv. prof. dr. sc. Davor Vinko |
| Član Povjerenstva 2: | Luka Filipović |
| Naslov završnog rada: | Dekoder Morse-ove abecede |
| Znanstvena grana rada: | Elektronika (zn. polje elektrotehnika) |
| Zadatak završnog rada | Zadatak završnog rada je razvoj i izrada dekodera Morse-ove abecede korištenjem mikroupravljača. Za više informacija javiti se mentoru: davor.vinko@ferit.hr |
| Prijedlog ocjene pismenog dijela ispita (završnog rada): | Vrlo dobar (4) |
| Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova: | Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 2 razina |
| Datum prijedloga ocjene mentora: | 31.08.2021. |
| <i>Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:</i> | Potpis: |
| | Datum: |

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**IZJAVA O ORIGINALNOSTI RADA**

Osijek, 04.10.2021.

Ime i prezime studenta:

Fran Birtić

Studij:

Preddiplomski stručni studij Računarstvo

Mat. br. studenta, godina upisa:

AI 4592, 09.10.2017.

Turnitin podudaranje [%]:

9

Ovom izjavom izjavljujem da je rad pod nazivom: **Dekoder Morse-ove abecede**

izrađen pod vodstvom mentora Izv. prof. dr. sc. Davor Vinko

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

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

IZJAVA

o odobrenju za pohranu i objavu ocjenskog rada

kojom ja Fran Birtić, OIB: 05508047008, student/ica Fakulteta elektrotehnike, računarstva i informacijskih tehnologija Osijek na studiju Preddiplomski stručni studij Računarstvo, kao autor/ica ocjenskog rada pod naslovom: Dekoder Morse-ove abecede,

dajem odobrenje da se, bez naknade, trajno pohrani moj ocjenski rad u javno dostupnom digitalnom repozitoriju ustanove Fakulteta elektrotehnike, računarstva i informacijskih tehnologija Osijek i Sveučilišta te u javnoj internetskoj bazi radova Nacionalne i sveučilišne knjižnice u Zagrebu, sukladno obvezi iz odredbe članka 83. stavka 11. *Zakona o znanstvenoj djelatnosti i visokom obrazovanju* (NN 123/03, 198/03, 105/04, 174/04, 02/07, 46/07, 45/09, 63/11, 94/13, 139/13, 101/14, 60/15).

Potvrđujem da je za pohranu dostavljena završna verzija obranjenog i dovršenog ocjenskog rada. Ovom izjavom, kao autor/ica ocjenskog rada dajem odobrenje i da se moj ocjenski rad, bez naknade, trajno javno objavi i besplatno učini dostupnim:

- a) široj javnosti
- b) studentima/icama i djelatnicima/ama ustanove
- c) široj javnosti, ali nakon proteka 6 / 12 / 24 mjeseci (zaokružite odgovarajući broj mjeseci).

**U slučaju potrebe dodatnog ograničavanja pristupa Vašem ocjenskom radu, podnosi se obrazloženi zahtjev nadležnom tijelu Ustanove.*

Osijek, 01.10.2021.

(mjesto i datum)

(vlastoručni potpis studenta/ice)

SADRŽAJ

| | |
|---|----|
| 1. UVOD | 1 |
| 2. RAZVOJ I PRIMJENA MORSE-OVE ABECEDA | 2 |
| 2.1. Prvi električni telegrafi | 2 |
| 2.2. Morse-ov kod | 2 |
| 2.3. Primjena Morse-ovog koda | 3 |
| 3. PROGRAMSKO RJEŠENJE, FIZIČKE KOMPONENTE I REZULTATI | 5 |
| 3.1. Opis platformi, tehnologija i alata | 5 |
| 3.1.1. Arduino platforma i razvojna pločica | 5 |
| 3.1.2. Arduino razvojno okruženje | 5 |
| 3.2. Izvod rješenja problema | 5 |
| 3.2.1. Unos i algoritam dekodiranja Morse-ove abecede | 5 |
| 3.2.2. Opis fizičkih komponenti uređaja | 10 |
| 3.2.3. Shema spajanja fizičkih komponenti uređaja | 11 |
| 3.3. Postignuti rezultati | 13 |
| 4. ZAKLJUČAK | 15 |
| LITERATURA | 16 |
| SAŽETAK (Arduino, dekoder, Morse-ova abeceda) | 17 |
| MORSE CODE DECODER – SUMMARY (Arduino, decoder, Morse code) | 18 |
| ŽIVOTOPIS | 19 |
| PRILOG | 20 |

1. UVOD

Morse-ova abeceda je način kodiranja koji je omogućavao komunikaciju na velikim udaljenostima prije razvoja danas poznatih telekomunikacijskih sustava. Nakon otkrića mogućnosti slanja signala kroz vodič, razvijene su brojne metode za prijenos poruke, ali najefikasniji i najjeftiniji način osmislili su Samuel F.B. Morse i Alfred Vail. Statističkom analizom engleskog jezika odredili su vjerojatnost pojave svakog slova te ih kodirali koristeći dva simbola koja se razlikuju u trajanju. Prvi, kraći simbol označava se kao točka (“.”, eng. *dot*), dok drugi simbol označava se kao crtica (“-”, eng. *dash*). Trajanje crtice trostruko je dulje od trajanja točke. Kratko vrijeme između točke i/ili crtice označava razmak između slova, a dulje vrijeme između signala označava razmak između riječi.

U vrijeme kada je Morse-ova abeceda bila u upotrebi dekodiranje poruke obavljali su obučeni operateri. Konstrukcija dekodera zamijenila bi operatera, automatizirala postupak dekodiranja poruka i smanjila mogućnost pogreške u prijenosu poruke. Sustav za dekodiranje poruke sastoji se od ulaza poruke (eng. “input”), jedinice za dekodiranje i izlaza za prikaz poruke u razumljivom obliku.

Dekoder Morse-ove abecede u 21. stoljeću moguće je simulirati koristeći Arduino razvojnu pločicu za dekodiranje poruke u obliku električnog signala, tipku za slanje poruke, računalo sa odgovarajućim programom za prikaz dekodirane poruke te vodiča za spajanje navedenih komponenti.

U nastavku završnog rada bit će objašnjen razvoj Morse-ove abecede, primjena abecede, konstrukcija i rad dekodera.

2. RAZVOJ I PRIMJENA MORSE-OVE ABECEDA

2.1. Prvi električni telegrafi

Prvi električni telekomunikacijski sustav namijenjen za prijenos tekstnih poruka od točke do točke bio je električni telegraf. Ovakav sustav omogućio je komunikaciju bez fizičkog prijevoza od 1840.-ih do sredine 20. stoljeća [1,2]. Prije električnog telegrafa koristili su se semafori sustavi koji uključuju svjetionike, dimne signale, semafor s zastavama i optičke telegrafe za vizualne signale za komunikaciju na udaljenostima od kopna [3, str. 37].

Električni telegraf sastojao se od dvije ili više zemljopisno odvojenih postaja (često nazivanih telegrafskim uredima) povezanih žicama, obično poduprtih na stupovima komunalnih usluga. Korišteni su za prijenos kodiranih impulsa električne struje kroz namjenske žice na velike udaljenosti.

Patentiran je čitav niz različitih električnih telegrafskih sustava, ali najšire rasprostranjeni uklapaju se u dvije široke kategorije. Prva kategorija sastoji se od iglenih brzjava u kojima se iglični pokazivač elektromagnetski kreće s pulsom električne struje iz baterije ili dinama niz telegrafsku liniju. Prvi komercijalni sustav i najkorišteniji telegraf s iglom, bio je telegraf Cooke i Wheatstone, izumljen 1837. Rani sustavi koristili su više igala koje su trebale više žica, ali trošak instalacije žica bio je ekonomski značajniji od troškova obuke rukovatelja, stoga je sustav s jednom iglom i kodom koji je trebalo naučiti postao norma. Drugu kategoriju čine armaturni sustavi u kojima impuls aktivira telegrafski zvučnik koji klikne. Morseov sustav, koji je izumio Samuel Morse 1838. godine koristeći jednu žicu, bio je armaturni sustav. Na stanici slanja, operater bi dodirnuo prekidač zvan telegrafski ključ, izgovarajući tekstualne poruke u Morseovom kodu. Izvorno je armatura bila namijenjena označavanju na papirnoj vrpici, međutim operateri su naučili tumačiti zvukove, stoga je postalo učinkovitije izravno zapisivati poruku.

2.2. Morse-ov kod

Morseov kod je metoda korištena u telekomunikacijama za kodiranje tekstualnih znakova kao standardiziranih sekvenci dva različita trajanja signala [4, str. 71]. Signali su nazvani točkice i crtice ili „dits“ i „dahs“ [5, str. 72.]. Svaki simbol Morseove abecede oblikovan je nizom točaka i crtica. Trajanje točke osnovna je mjerna jedinica vremena u prijenosu Morseova koda. Morseova

abeceda dobila je ime po Samuelu Morseu, jednom od izumitelja i pionira telegrafске komunikacije. Međunarodni Morseov kod, također poznat kao Kontinentalni Morseov kod, kodira 26 slova latinice od a do z, jedno nelatinično slovo, arapske brojeve te mali skup interpunkcijskih i proceduralnih signala bez razlike u velikim i malim slovima.

Morseov kod se uobičajeno prenosi uključivanjem i isključivanjem medija za prijenos informacija kao što su električna struja ili radiovalovi, međutim Morseov se kod može zapamtiti i poslati u obliku koji je uočljiv ljudskim osjetilima, npr. zvučnim valovima ili vidljivom svjetlošću [6, str. 6.–7.]. Budući da mnogi jezični sustavi koriste više od 26 slova latinske abecede, transliteracijom postojećih kodova razvijeni su Morseovi kodovi prilagođeni takvim jezičnim sustavima [7, str. 68–72]. S obzirom da su elementi Morseove abecede specificirani proporcijom, a ne vremenskim trajanjem, kod se prenosi najvećom brzinom koju prijemnik može dekodirati. Brzina prijenosa Morseova koda određena je skupinama po minuti, obično riječima u minuti.

Bitno je napomenuti da je originalna Morse-ova abeceda prilagođena engleskom jeziku. Kako bi se smanjila duljina slučajne poruke korišteno je Huffmanovo kodiranje. Statističkom analizom engleskog jezika utvrđena je učestalost svakog slova te je po pravilu Huffmanovog kodiranja češćim znakovima dodjeljen manji broj bitova. Primjerice „E“ kao najčešće slovo engleskog jezika je predstavljeno sa jednom točkom, dok je „Q“ predstavljeno kao --.-. Isti bismo postupak mogli provesti i za hrvatsku abecedu, kreirajući novu inačicu Morse-ove abecede.

2.3.Primjena Morse-ovog koda

Najraniji telegrafski sustavi koristili su sustav s jednom iglom koji je činio vrlo jednostavan i robustan instrument [3, str. 10]. Međutim, ovakav način prijenosa bio je spor jer je primatelj morao izmjenjivati gledanje igle i zapisivanje poruke. Ovaj problem riješen je Morseovim kodom u kojem otklon igle ulijevo odgovara točki, a ugib udesno crtici. Naime, otklonom u lijevo igla bi proizvela zvuk koji zvuči drugačije od onoga koji nastaje otklonom u desno. Obzirom da dva zvuka zvuče različito, ovakav instrument s jednom iglom je postao zvučni instrument, što je na koncu dovelo do razvoja zvučnog sistema s dvije pločice (eng. Double Plate Sounder System). Takav je uređaj pronašao komercijalnu primjenu diljem svijeta.

Morseov sustav za telegrafiju, koji se prvi put koristio oko 1844. godine, dizajniran je za izradu udubljenja na papirnoj traci primjenom električne struje. Morseov originalni telegrafski

prijamnik koristio je mehanički satni mehanizam za pomicanje papirnate vrpce [3, str. 47]. Kad je primljena električna struja, elektromagnet je uključio armaturu koja je gurnula olovku na pokretnu papirnatu traku, čineći udubljenje na traci. Kad je struja prekinuta, opruga je povukla olovku i taj dio pokretne trake ostao je neoznačen. Morseova abeceda razvijena je tako da operateri mogu prevesti udubljenja označena na papirnatj vrpci u tekstualne poruke. U najranijem dizajnu, Morseov kod se sastojao samo od brojeva i za dekodiranje je bila potrebna knjiga s popisom šifri i odgovarajućih riječi koje te šifre kodiraju. Međutim, inženjer strojarstva Alfred Vail je 1840. godine analizom učestalosti upotrebe slova u engleskom jeziku proširio originalni kod na slova i posebne znakove, što je omogućilo njegovu širu primjenu. Ovakav kod, prvi put korišten 1844. godine, postao je poznat kao Morseov fiksni kod ili „Railroad Morse“, a korišten je sve do kraja željezničke telegrafije u SAD-u 1970.-ih godina.

U drugoj polovici 19. stoljeća električni telegraf zamijenio je optički telegrafski sustav, i tako postao standardni način slanja hitnih poruka. Većina razvijenih država stvorila je i komercijalne telegrafske mreže s lokalnim telegrafskim uredima. U takvim uredima u gradovima i selima diljem zemlje, javnosti je bilo omogućeno slanje telegrama upućenih bilo kojoj osobi u zemlji, uz određenu naknadu. Podmorski telegrafski kabeli su omogućili prvu brzu komunikaciju između kontinenata.

Nakon tehnoloških otkrića poput telefona, interneta, mobilne telefonije i dr., telegraf je izbačen iz upotrebe, a najduže se zadržao i Indiji i Australiji.

3. PROGRAMSKO RJEŠENJE, FIZIČKE KOMPONENTE I REZULTATI

3.1. Opis platformi, tehnologija i alata

3.1.1. Arduino platforma i razvojna pločica

Arduino je open-source platforma za izradu uređaja koji reagiraju na promjene iz okoline. Za rad uređaja najbitnija je Arduino razvojna pločica na kojoj je moguće upisivati i mijenjati programski kod koji upravlja ulazno - izlaznim jedinicama koje su spojene na mikroupravljač. Za razvoj dekodera ulazni signal generirat ćemo pomoću tipke, a poruka će biti prikazana u Arduino razvojnom okruženju.

3.1.2. Arduino razvojno okruženje

Arduino razvojno okruženje omogućuje zapis programskog koda u C ili C++ programskom jeziku u memoriju mikroupravljača. Zapisivanje programskog koda omogućeno je klikom na tipku u sučelju, a prethodno je potrebno odabrati model korištenog mikroupravljača i spojiti razvojnu pločicu sa računalom koristeći USB priključak. Nakon zapisa koda u mikroupravljač, uređaj je spreman za korištenje. Isti USB priključak kao za zapis koda koristimo za prijenos dekodirane poruke do računala, a prikaz poruke je omogućen servisom unutar Arduino razvojnog okruženja pod nazivom Serial monitor.

3.2. Izvod rješenja problema

3.2.1. Unos i algoritam dekodiranja Morse-ove abecede

Sustav za slanje poruke u obliku Morse-ove abecede sadrži jednu tipku. Kraćim ili duljim držanjem tipke unosimo točku ili crticu u slovo koje trenutno upisujemo. Provjera pojedinog utipkanog simbola (točke i crtice) ovisi o trajanju pritiska tipke. Ukoliko držimo tipku dulje od 300 ms upisujemo crticu, a pritisak tipke kraći od 300 ms interpretira se kao točka. Vremena potrebna za unos točaka i crtica prikazana su u tablici 1.

| x = trajanje pritiska tipke (ms) | Tumačenje vremena pritiska tipke |
|---|---|
| $x < 300$ | Upis točke |
| $300 \leq x$ | Upis crtice |

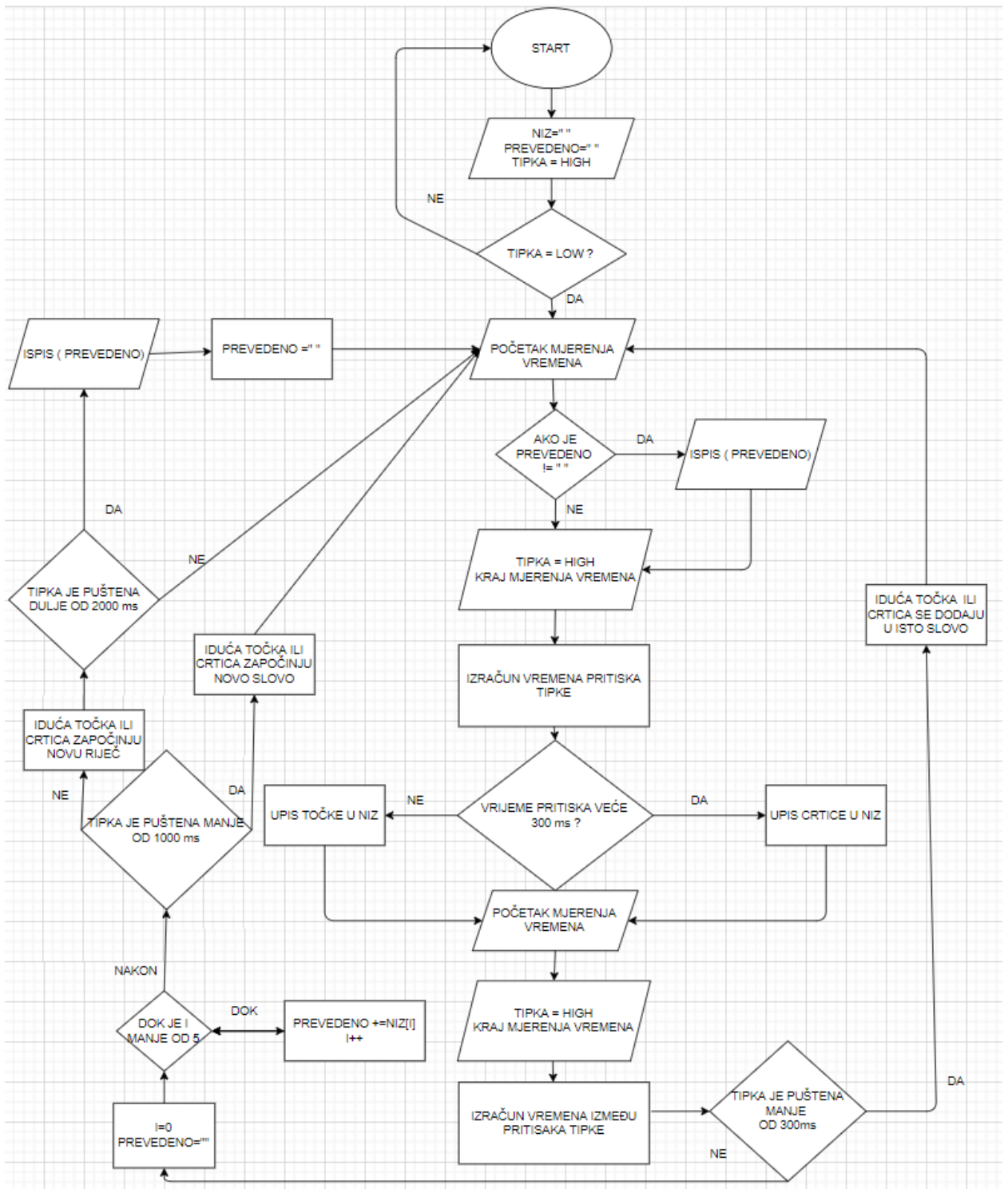
Tablica 1. tumačenje vremena pritiska tipke

Telekomunikacijski sustavi koji su koristili Morse-ovu abecedu u obzir su uzimali i vrijeme koje je prošlo između točaka i crtica. Razmak između točaka i crtica unutar istog slova jednak je duljini točke, razmak između dva slova iste riječi jednak je tri duljine točke, a nova riječ se označava sa sedam duljina točke.[6, str. 8] Nakon svakog puštanja tipke mjerimo vrijeme do idućeg pritiska na tipku. Ukoliko je tipka ponovno pritisnuta za manje od 300 ms, dodajemo točku ili crticu u prethodno slovo. Ukoliko je razmak između dva pritiska tipke između 300 i 1000 ms, prevodimo slovo koje smo do sada upisali i dodajemo novo slovo u istu riječ. Gornja granica vremena potrebnog za upis novog slova u istu riječ je podignuta sa 700 na 1000 ms kako bismo jednostavnije upisivali riječi. Ukoliko je vrijeme između pritisaka između 1000 i 2000 ms započinjemo novu riječ. Ako je od zadnjeg pritiska prošlo više od 2000 ms, ispisujemo prethodni upisani znak kako bismo osigurali ispis zadnjeg slova. Ovisnost vremena između pritisaka tipke prikazana je u tablici 2.

| y = trajanje između pritisaka tipke (ms) | Tumačenje vremena između pritisaka tipke |
|---|---|
| $y < 300$ | Iduću crticu ili točku upisujemo u isto slovo |
| $300 < y \leq 1000$ | Početak novog slova u istoj riječi. |
| $1000 < y \leq 2000$ | Početak nove riječi. |
| $2000 < y$ | Ispis zadnjeg slova poruke. |

Tablica 2. tumačenje vremena između pritisaka tipke

Svako slovo će se ispisati na ekranu u latiničnom i Morseovom obliku. Obzirom da nam je poznat konačan broj slova, u polje znakova možemo abecedno upisati sve kodirane znakove, te za svaku kombinaciju točkica i crtica koju pošaljemo dekođeru provjeriti nalazi li se u popisu. Ukoliko je operater pogriješio, na ekran će se ispisati serija točaka i crtica koja je poslana. Shematski prikaz algoritma nalazi se na slici 1.



Slika 1. shematski prikaz izvođenja programa

U nastavku slijede ključni dijelovi koda. U loop petlji se provjerava stanje tipke. Ukoliko je tipka pritisnuta pozivamo funkciju koja očitava poruku. Poruka se očitava mjerenjem vremena za koje je tipka pritisnuta te mjerenjem vremena za koje tipka nije pritisnuta. Prilikom detekcije pritiska tipke bilježimo vrijeme, a za to koristimo funkciju millis. Prilikom definiranja varijabli

bitno je pripaziti da su varijable u koje spremamo vrijeme u mogućnosti spremiti dovoljno velike brojeve za dulje izvođenje programa. Spomenuta funkcija `millis` vraća broj koji predstavlja broj ms od početka izvođenja programa tako da varijable `startPressed`, `endPressed`, `idleTime` i `holdTime` trebaju biti tipa `long`. Programski kod koji se izvodi pritiskom tipke prikazan je slikom 2. Puštanjem tipke izvršava se kod sa slike 3. U trenutku puštanja tipke možemo izračunati vrijeme trajanja držanja tipke te upisati točku ili crticu u skladu s trajanjem pritiska. Prilikom idućeg pritiska na tipku ponovno se izvršava kod sa Slike 2. Ukoliko vrijeme puštene tipke upućuje da je znak upisan do kraja, tj. ukoliko je vrijeme od zadnjeg puštanja tipke veće ili jednako 300, a manje od 1000 ms, poziva se funkcija `convert` koja u popisu slova traži utipani niz točaka i crtica te ih ispisuje na ekran. Ukoliko je vrijeme od puštanja tipke do ponovnog pritiska između 1000 i 2000 ms, između slova će se upisati razmak. Kako bismo osigurali ispis zadnjeg znaka, u loop petlji pozivamo funkciju `convert` ukoliko je prošlo 2 sekunde od puštanja tipke bez novog pritiska i ukoliko već postoji utipkano i neispisano slovo. Programski kod ove provjere prikazan je slikom 4.

```
if (buttonState == LOW) {
    startPressed = millis();
    idleTime = startPressed - endPressed;

    if (idleTime >= 300 && idleTime < 1000) {
        convert();
    }

    if (idleTime >= 1000 && idleTime <= 2000) {
        convert();
        Serial.print(" ");
    }
}
```

Slika 2. programski kod koji se izvršava pritiskom na tipku, računanje vremena od zadnjeg puštanja tipke i odvajanje sljedećeg znaka ili riječi

```

endPressed = millis();
holdTime = endPressed - startPressed;

if (holdTime >= 0 && holdTime < 300) {
    m_in[m_inc] = '.';
    m_inc++;
}

if (holdTime >= 300) {
    m_in[m_inc] = '-';
    m_inc++;
}

```

Slika 3. programski kod koji se izvršava puštanjem tipke, računanje vremena od pritiska do puštanja tipke određuje upis crtice ili točke

```

if(millis()-endPressed>2000 && m_in[0] != ' '){
    convert();
}

```

Slika 4. programski kod koji se izvršava ukoliko u memoriji postoji znak unesen prije 2 sekunde, osigurava ispis zadnjeg znaka.

Ranije spomenuta funkcija convert poziva funkciju findMorse koja koja niz točaka i crtica traži u Morse-ovoj abecedi i pronalazi latiničnu protuvrijednost. Funkcija findMorse prikazana je slikom 5.

```

String findMorse(String s){
    int v = 36;
    for (int x = 0; x<36;x++)
    {
        if (morsecode[x].equals(s))
            v = x;
    }
    return alpha[v];
}

```

Slika 5 programski kod koji uspoređuje utipkanu kombinaciju točaka i crtica te ukoliko ista postoji u Morse-ovoj abecedi, vraća slovo u latiničnom obliku

Za unos točaka i crtica definirano je globalno polje znakova `m_in` u koje je moguće unijeti 5 elemenata. Unutar `convert` funkcije pomoću `Serial.print` naredbe ispisuje se slovo u Morse-ovoj abecedi i u latiničnom obliku. Nakon ispisa svakog slova ili broja, svaki element polja `m_in` se postavlja na prazan znak kako bi polje bilo spremno za unos sljedećeg znaka.

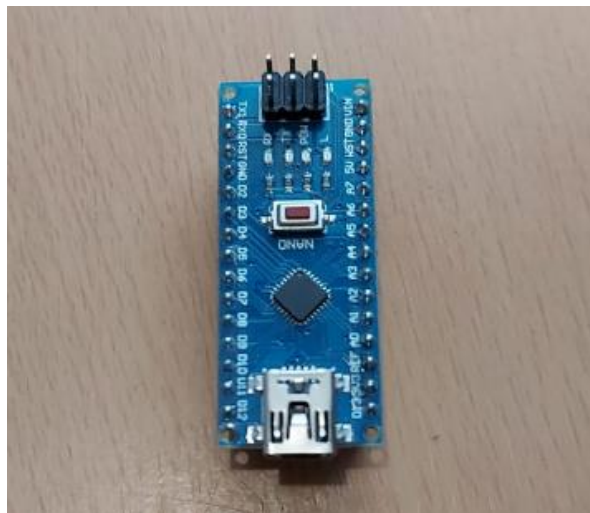
3.2.2. Opis fizičkih komponenti uređaja

Za izradu dekodera korištene komponente su :

- Razvojna pločica Arduino Nano

Obzirom da u današnje vrijeme imamo vrlo velik izbor mikroupravljača te da za dani problem nije potrebno puno procesorske snage, odabrana je komponenta sa sljedećim tehničkim specifikacijama:

- Mikroupravljač : ATmega 328
- Radni napon : 5 V
- Flash memorija : 32 kB
- Takt : 16 MHz



Slika 6. Arduino Nano Razvojna pločica

- Tipkom definiramo trajanje pojedine crtice ili točke i razmak između slova i riječi. Obzirom da je za potrebe slanja kraćih i duljih signala potrebna preciznost u tipkanju, kao tipka korišten je endstop za 3d pisač. Za pritisak ove tipke potreban

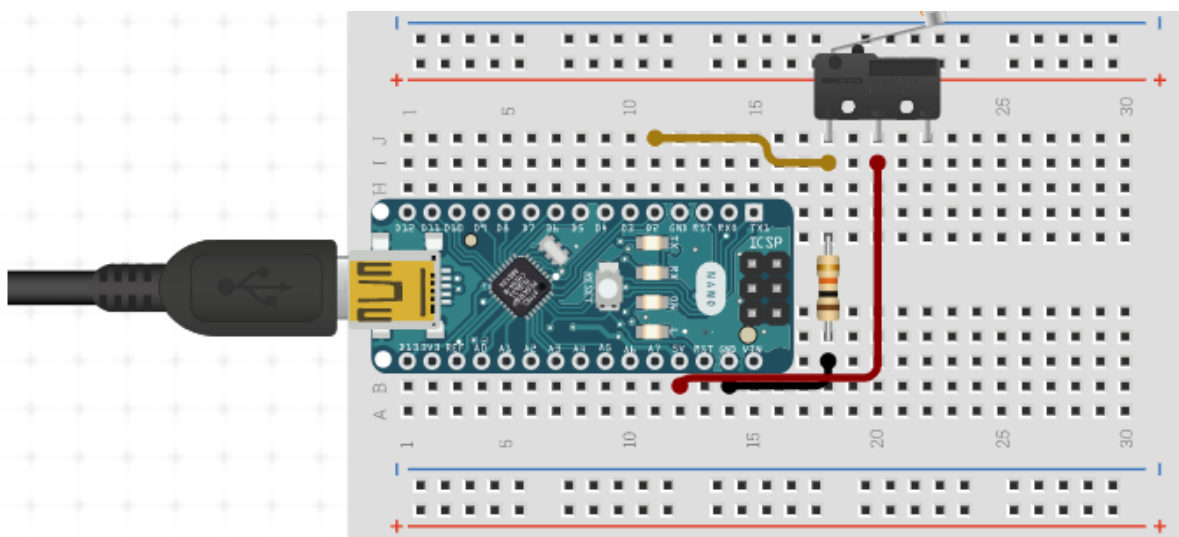
je slabiji pritisak nego uobičajena tipkala za Arduino, pa je slanje poruka puno efikasnije. Tipka je prikazana na slici 7.



Slika 7. tipka , endstop za 3D pisač

- Vodiči za spajanje navedenih komponenti.

3.2.3. Shema spajanja fizičkih komponenti uređaja

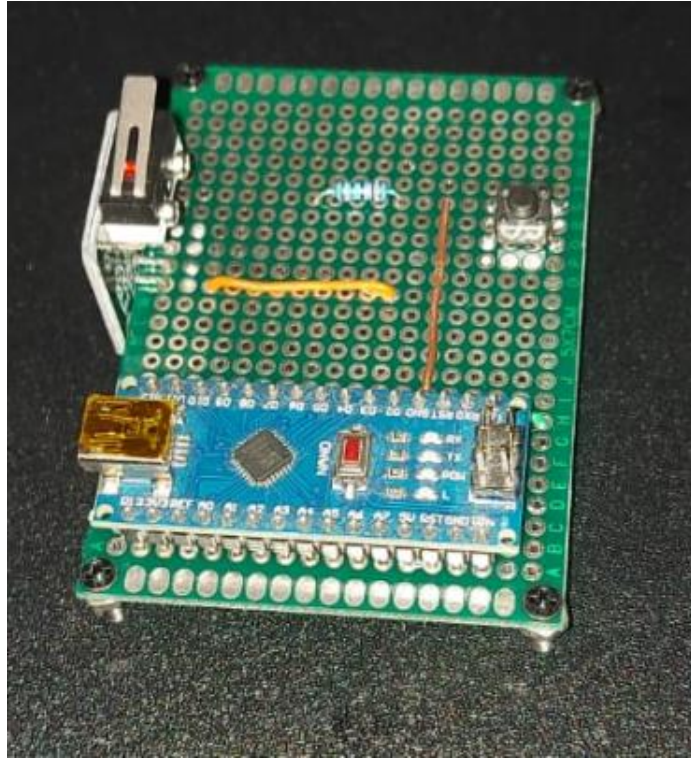


Slika 8. shematski prikaz spojenog dekodera

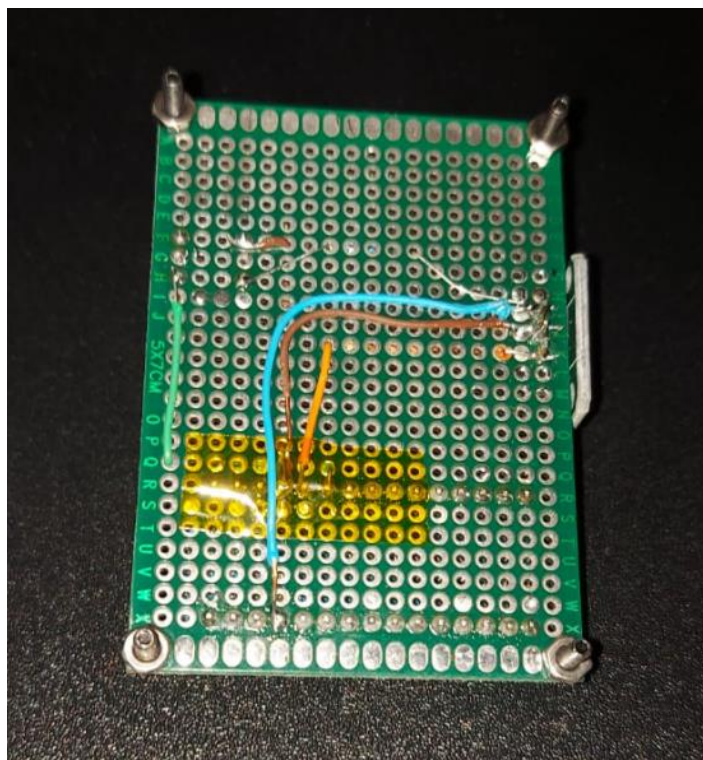
Shematski prikaz spoja predstavljen je slikom 8. Tipku vodičem spajamo na uzemljenje i izvor napajanja te ju spajamo sa razvojnom pločicom na pin D2. Tipka na pločici već ima ugrađen otpornik od 10kohm-a. Razvojna pločica je spojena s računalom putem MINI-B USB kabela. Kako

bismo prikazali poslane poruke, na računalu je nužno pokrenuti Serial Monitor unutar Arduino razvojnog okruženja.

Nakon što je rad dekodera testiran, komponente su zalemljene na pločicu za lemljenje. Uređaj je prikazan Slikama 9 i 10.



Slika 9. Fotografija dekodera sa tipkom za unos – prednja strana



Slika 10. Fotografija dekodera sa tipkom za unos – stražnja strana

3.3. Postignuti rezultati

Ovako konstruiran uređaj na Serial monitor ispisivat će utipkane poruke. Ispis poruka prikazan je na Slikama 11, 12 i 13 :

Nova Poruka:

--M---O.-.R...S.E---O...-V.-A .-A-...B.E--.C.E-...D.-A

Slika 11. Ispis “MORSEOVA ABECEDA”

Nova Poruka:

--M.E-.N-T---O.-.R ...D.-A...-V---O.-.R ...-V..I-.N--K---O

Slika 12. Ispis “MENTOR DAVOR VINKO”

Nova Poruka:

..I--..Z.-.R.-A--D..I---O ..-.F.-.R.-A-.N -...B..I.-.R-T..I-.-.C

Slika 13. Ispis "IZRADIO FRAN BIRTIC"

Ovaj uređaj je napravljen po međunarodnim pravilima Morse-ove abecede, uz modifikaciju gornje granice za upis slova u istu riječ, tako da je trajanje točki, crtica i vremenskog razmaka između tih simbola definirano kao što je ranije u tekstu navedeno. Utipkavanje na ovaj način zahtjeva potpunu koncentraciju operatera te je potrebno određeno vrijeme kako bi se operater uvježbao. Kao pomoć novim korisnicima ovoga uređaja, moguće je dodati još jednu tipku u sklop kojom bi se za vrijeme izvođenja programa linearno produljilo vrijeme unosa točki, crtica i razmaka. Tako bi primjerice bilo moguće vrijeme jedne točke povećati sa 100 na 200 ms, vrijeme crtice sa 300 na 600 ms te isto primijeniti na razmake između slova i riječi. Na taj način bi način bio omogućen unos znakova originalnom i sporijom brzinom te bi bio izbjegnuta problem kratkih vremenskih intervala koji bi eventualno otežao korištenje novim korisnicima.

4. ZAKLJUČAK

Cilj ovoga završnog rada je napraviti dekoder Morse-ove abecede. Za proces dekodiranja potrebna je poruka čiji je unos omogućen pomoću tipke, tako da konačan sklop zapravo predstavlja model telegrafskog sustava sa automatiziranim dekodiranjem. Za vrijeme razvoja ovoga uređaja koncept unosa poruke i prevođenja promijenjen je nekoliko puta kako bi se olakšao unos poruke korisnicima, a došlo je i do promjena programskog koda u fazi testiranja uređaja. Primjerice, bitno je pripaziti da ne dođe do prelijeva kod brojanja vremena pomoću funkcije millis jer se rad sklopa temelji na računanju vremenske razlike između pritisaka tipke i računanju vremena za koje tipka nije pritisnuta. Kako bi unos rečenica bio olakšan, vrijeme između pritisaka tipke potrebno za upis novog slova u istoj riječi produljeno je sa 700 ms na 1000 ms zbog čestih grešaka kod unosa. Ovakav uređaj bio bi koristan pri učenju i uvježbavanju Morse-ove abecede. Uz kratko upoznavanje sklopa osoba koja se nikada nije susrela sa Morse-ovom abecedom za nekoliko minuta vježbe može samostalno upisivati rečenice.

U moderno vrijeme se Morse-ova abeceda vrlo rijetko koristi, međutim primjena ove abecede u telegrafiji je bila veliki korak u razvoju telekomunikacijskih sustava koje danas svakodnevno koristimo.

LITERATURA

1. S. Moss , Final telegram to be sent. STOP, The Guardian: International Edition , 2013.
dostupno na: <https://www.theguardian.com/world/shortcuts/2013/jul/10/final-telegram-to-be-sent-india>, [13.7.2021.]
2. R. Wenzlhuemer The Development of Telegraphy, 1870–1900: A European Perspective on a World History Challenge, History Compass, br 1, svezak 5, 2007. dostupno na:
<https://onlinelibrary.wiley.com/doi/full/10.1111/j.1478-0542.2007.00461.x>, [13.7.2021.]
3. S. F. B. Morse, Examination of the Telegraphic Apparatus and the Processes in Telegraphy, U.S. Government Printing Office, Washington DC, 1869. dostupno na:
https://books.google.hr/books?id=rGgPAAAYAAJ&pg=PA7&redir_esc=y#v=onepage&q&f=false , [13.7.2021.]
4. F.S. Beechey, Electro-Telegraphy UK: E. & F.N. Spon. , London ,1876. , dostupno na:
<https://archive.org/details/electrotelegrap00beecgoog/page/n74/mode/2up>, [13.7.2021.]
5. F.J. Camm, Radio Engineers' Pocket Book 2nd edition. Chemical Publishing Co.,UK, 1941.,
dostupno na: <https://archive.org/details/RadioEngineersPocketBook/page/n71/mode/2up>,
[13.7.2021.]
6. Department of the Army (US), International Morse Code (Instructions). U.S. Government Printing Office. Washington, DC: Department of the Army. 1968., dostupno na:
<https://archive.org/details/us-army-11-459/page/n3/mode/2up> , [13.7.2021.]
7. Instructions for Learning International Morse Characters. War Department., Washington DC 1943., dostupno na:
https://archive.org/details/Tm11-459_676-nsia/page/n51/mode/2up , [13.7.2021.]

SAŽETAK (Arduino, dekodер, Morse-ova abeceda)

Cilj ovog završnog rada je izraditi dekodер Morse-ove abecede. Unos znakova u obliku Morse-ove abecede omogućen je pomoću tipke. Za dekodiranje unesenih podataka koristi se Arduino Nano razvojna pločica koja je, osim s tipkom, spojena sa računalom preko sučelja za serijsku komunikaciju. Dekodiranje se temelji na jednostavnom uspoređivanju znakova. Svako slovo latinične abecede ima Morse-ov niz od jednog ili više znakova. Tako svaku kombinaciju kratkih i dugih pritisaka tipke možemo usporediti sa vrijednostima koje su prethodno učitane na Arduino razvojnu pločicu. Dekodirani znakovi ispisuju se u Serial monitor unutar Arduino razvojnog okruženja. Rezultat projekta je sklop kojim se jednostavno upisuju simboli u obliku Morse-ove abecede te se na zaslon ispisuje latinični znakovi.

MORSE CODE DECODER – SUMMARY (Arduino, decoder, Morse code)

The goal of this paper is to develop a Morse code decoder. Entering characters in the Morse code form is enabled by use of a button. An Arduino Nano development board is used to decode the entered data. The board is connected to the computer via a serial communication interface. Decoding is based on comparison of strings of characters. Each letter of the Latin alphabet has a Morse string of one or more characters. Each combination of short and long key presses can be compared to the strings previously loaded on the Arduino development board. Decoded characters are printed to a Serial monitor within the Arduino integrated development environment. The result of the project is a circuit that translates symbols in the form of the Morse code to the Latin characters and prints them on the screen.

ŽIVOTOPIS

Fran Birtić je rođen 19. kolovoza 1997. godine u Osijeku. Slijedi kronološki pregled obrazovanja.

Završena Osnovna Škola Laslovo.

Završena I. Gimanzija Osijek.

Trenutno pohađa Preddiplomski stručni studij računarstva na Fakultetu Elektrotehnike, Računarstva i Informacijskih Tehnologija u Osijeku.

PRILOG

U nastavku se nalazi programski kod.

```
const int buttonPin = 2;
int buttonState = HIGH;
int lastButtonState = HIGH;
long startPressed = 0;
long endPressed = 0;
long holdTime = 0;
long idleTime = 0;
int m_inc = 0;
char m_in[6];
String translated = "";

String morsecode[] ={
    ".-", "-...", "-.-.", "-..", ".:", "..-.", "---", "....",
    "..", ".---", "-.-", "-..", "--", "-.", "---", ".---",
    "--.-", ".-.", "...", "-", ".-.", "....", "-.-", "-.-.",
    "-.-.", "-..", ".----", ".---", "....", ".....", "-.....", "-....", "-...", "-..", "-.-.", "-.-."
};

String alpha [] = {"A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N", "O", "P",
"Q", "R", "S", "T", "U", "V", "W", "X", "Y", "Z", "1", "2", "3", "4", "5", "6", "7", "8", "9", "0", "
"};

void setup() {
    pinMode(buttonPin, INPUT);
    Serial.begin(9600);
    Serial.println("\nNova Poruka:");
}
```

```

void loop() {

    buttonState = digitalRead(buttonPin);

    if (buttonState != lastButtonState) {
        updateState();
    }

    if(millis()-endPressed>2000 && m_in[0] != ' '){
        convert();

    }
    lastButtonState = buttonState;
}

void updateState() {

    if (buttonState == LOW) {
        startPressed = millis();
        idleTime = startPressed - endPressed;

        if (idleTime >= 300 && idleTime < 1000) {
            convert();
        }
        if (idleTime >= 1000 && idleTime<=2000) {
            convert();
            Serial.print(" ");
        }

    } else {

```

```

endPressed = millis();
holdTime = endPressed - startPressed;
if (holdTime >= 0 && holdTime <= 300) {
    m_in[m_inc] = '.';
    m_inc++;
}
if (holdTime >= 300) {
    m_in[m_inc] = '-';
    m_inc++;
}
}
}

```

```

void convert(){
    for (int i = 0; i<6; i++)
    {
        translated += m_in[i];
    }
    translated.trim();
    Serial.print(translated);
    translated = findMorse(translated);
    Serial.print(translated);
    translated = "";
    for (int i = 0; i<6; i++)
    {
        m_in[i] = ' ';
    }
    m_inc=0;
}

```

```

String findMorse(String s){
    int v = 36;

```

```
for (int x = 0; x<36;x++)  
{  
    if (morsecode[x].equals(s))  
        v = x;  
}  
return alpha[v];  
}
```