

# Arduino pinger

Strokovno poročilo za četrti predmet poklicne mature



Mentorji:

Mag. Milan Setničar  
Alenka Zorko, prof.  
Matevž Ževart, inž.

Avtor:

Domen Žukovec, R 4. B

Ljubljana, maj 2019

## Povzetek

V poročilu je predstavljen tester omrežij Arduino pinger. Na začetku so opisana računalniška omrežja, natančneje osebna in javna. Nato pa so predstavljena orodja in okolja, ki sem jih uporabljal za izdelavo testerja. Sledi podrobna predstavitev delovanja in sestave Arduino pingerja, za tem pa še težave, s katerimi sem se soočil pri izdelavi.

**Ključne besede:** Arduino, Arduino pinger, usmerjevalnik, testiranje omrežja

## Abstract

In this report a network tester named is presented. First, we talk about computer networks, specifically public and home networks. Then tools and environments that I used for the creation of the tester are introduced. That follows with a detailed presentation on how the Arduino pinger works and how it's made. Next we talk about some problems I had to work around and in the end I present a conclusion.

**Key words:** Arduino, Arduino pinger, router, network testing

# Kazalo

1.	Uvod.....	5
2.	Računalniška omrežja .....	5
2.1	Domače omrežje .....	5
2.2	Javno omrežje .....	5
3.	Orodja in okolja.....	6
3.1	Arduino .....	6
3.1.1	Zgodovina.....	6
3.1.2	Arduino Mega 2560 .....	6
3.1.3	Arduinovo integrirano razvojno okolje (IDE).....	7
3.2	C++ .....	7
3.3	Notepad++ .....	8
3.4	3D-tiskanje.....	9
3.4.1	Delovanje 3D-tiskanja.....	9
3.4.2	Zgodovina 3D-tiskalnikov.....	9
3.4.3	Tinkercad.....	10
4.	Delovanje .....	10
4.1	Program in funkcije .....	10
4.1.1	Osnovne deklaracije .....	11
4.1.2	Funkcija rebootEthShield .....	11
4.1.3	Funkcije googleKratek .....	12
4.1.4	Funkcija localKratek .....	12
4.1.5	Funkcija videoKratek .....	13
4.1.6	Funkcija voiceKratek .....	13
4.1.7	Funkcija googleDolgi .....	14
4.1.8	Funkcija localDolgi .....	14
4.1.9	Funkcija videoDolg .....	15
4.1.10	Funkcija voiceDolg .....	15
4.1.11	Funkciji loop in setup.....	16
4.1.12	Preverjanje funkcij gumba.....	16
4.2	Sestavni deli.....	17
5.	Težave pri razvoju .....	18
6.	Zaključek.....	18
7.	Viri in literatura.....	19
7.1	Viri slik .....	19

## Kazalo slik

Slika 1: Arduinovo IDE-okolje .....	7
Slika 2: Primer kode v jeziku C++ .....	8
Slika 3: Ohišje Arduino pingerja v programu Tinkercad .....	10
Slika 4: Osnovni meni Arduino pingerja.....	10
Slika 5: Globalne spremenljivke programa Arduino pingerja .....	11
Slika 6: Primer funkcije Arduino pingerja .....	11
Slika 7: Prikaz funkcije googleKratek na zaslonu Arduino pingerja .....	12
Slika 8: Prikaz funkcije localKratek na zaslonu Arduino pingerja .....	12
Slika 9: Prikaz funkcije videoKratek na zaslonu Arduino pingerja .....	13
Slika 10: Prikaz funkcije voiceKratek na zaslonu Arduino pingerja .....	13
Slika 11: Prikaz funkcije googleDolgi na zaslonu Arduino pingerja.....	14
Slika 12: Prikaz funkcije localDolgi na zaslonu Arduino pingerja .....	14
Slika 13: Prikaz funkcije videoDolg na zaslonu Arduino pingerja .....	15
Slika 14: Prikaz funkcije voiceDolg na zaslonu Arduino pingerja .....	15
Slika 15: Odsek kode, ki preverja koliko časa je bil gumb kliknjen .....	16
Slika 16: Odsek kode, kliče ustrezno funkcijo .....	16
Slika 17: Sestavni deli Arduino pingerja.....	17
Slika 18: Vezje Arduino pingerja.....	17

# 1. Uvod

Za izdelavo Arduino pingerja sem se odločil, saj za testiranje omrežja z računalnikom tehnik na terenu porabi preveč časa in usmerjevalnik brez kakršnegakoli testiranja velikokrat kar zamenja za novega. Delujoči usmerjevalniki so zato velikokrat brez potrebe poslani na servis, kar podjetjem, sploh večjim kot so Telekom d. d., povzroča velike denarne izgube. Spodbudo za izdelavo naprave mi je dal mentor Matevž Ževart med prakso na Telekomu. Arduino pinger je kompaktna in lahka naprava, ki jo tehnik lahko brez težav nosi po terenu in z njo v sekundah preveri delovanje povezave z usmerjevalnikom, glavnim strežnikom Googla (torej internetom) in s telefonsko ter televizijsko linijo (1).

## 2. Računalniška omrežja

### 2.1 Domače omrežje

Je proces povezovanja vseh naprav v domu skupaj z internetom. Domače omrežje se uporablja za vzpostavitev povezave med napravami, ki jih uporabljamo doma. Podobno je standardnemu omrežju LAN ali WLAN, ki je krajevno omejeno. Predvsem ga uporabljamo za povezovanje potrošnikov in njihovih naprav. Ti vključujejo računalnike, prenosnike, mobilne telefone, tablice in televizorje. Na splošno se domače omrežje lahko doseže s pomočjo usmerjevalnika, ki ga nudi ponudnik internetnih storitev, ki ima lahko podporo za več žičnih ali brezžičnih vrst povezav. Vse naprave se lahko neposredno povežejo z modemom, da lahko dostopajo do interneta (2).

### 2.2 Javno omrežje

Je vrsta omrežja do katerega ima lahko dostop vsakdo in se prek njega lahko poveže z drugimi omrežji ali internetom. To je v nasprotju z zasebnim (domačim) omrežjem, kjer se določijo omejitve in pravila dostopa. Ker ima javno omrežje malo omejitev ali pa jih sploh nima, morajo uporabniki pri dostopanju paziti na morebitna varnostna tveganja. Javno omrežje je označba uporabe in ne topologija ali drugo tehnično povezano načelo. Tehnične razlike med zasebnim in javnim omrežjem glede strojno opremo in infrastrukturo ni, razen varnostnih sistemov ter sistemov za nastavljanje in preverjanje prisotnosti (3).

## 3. Orodja in okolja

### 3.1 Arduino

Arduino je podjetje, ki temelji na odprtokodni strojni in programski opremi. Poleg podjetja je tudi skupnost, kateri lahko prispeva vsak. Glavna funkcija podjetja je oblikovanje in izdelovanje enoploščnih mikrokrmilnikov ter mikrokrmilniških kompletov za izgradnjo digitalnih naprav in interaktivnih objektov. Njihovi izdelki so licencirani v skladu z *GNU General Public License (GPL)* oziroma z splošno javno licenco, kar omogoča izdelavo Arduino plošč in distribucijo programske opreme z strani vseh. Vezja so komercialno na voljo v že vnaprej sestavljeni obliki ali pa kot kompleti za samo izdelavo.

Modeli plošč uporabljajo vrsto mikroprocesorjev in krmilnikov. Vezja so opremljena z digitalnimi in analognimi vhodi ali izhodi, na katere lahko priklopimo različne razširitvene plošče (ščite) ali pa druga vezja. Nekateri modeli imajo serijske komunikacijske vmesnike, vključno z univerzalnim serijskim vodilom (USB), ki se uporablja tudi za nalaganje programov iz osebnih računalnikov. Mikrokontrolerje običajno sprogramiramo z pomočjo funkcij iz programskih jezikov C in C++. Poleg tradicionalnih orodij za programiranje, projekt Arduino zagotavlja svoje integrirano razvojno okolje (IDE) (4).

#### 3.1.1 Zgodovina

Projekt Arduino se je začel leta 2003 kot program za študente na Interactive Design Institute Ivrea v Italiji, s ciljem začetnikom in strokovnjakom zagotoviti poceni in preprost način za ustvarjanje naprav. Primeri takšnih naprav, namenjenih začetnikom, so enostavni roboti, termostati in detektorji gibanja. Ime Arduino izhaja iz bara v Ivrea v Italiji, kjer so se srečali ustanovitelji projekta.

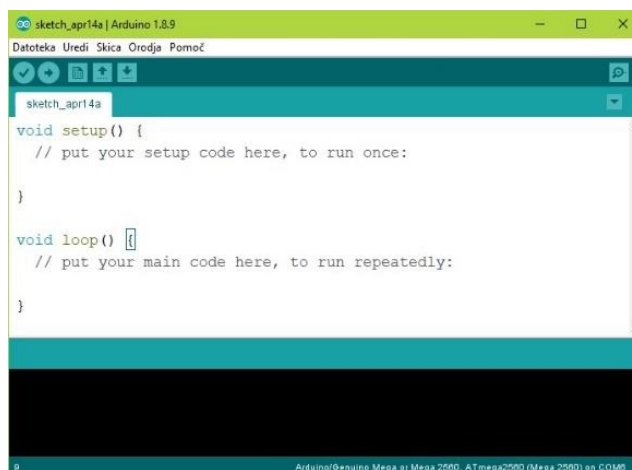
#### 3.1.2 Arduino Mega 2560

Mega 2560 je zasnovan za bolj komplekse projekte. Ima 54 digitalnih vhodno/izhodnih zatičev (od tega 15 lahko uporabimo kot PWM-izhode), 16 analognih vhodov, 4 UART(strojna serijska vrata), 16 MHz-kristalni oscilator, USB-povezljivost, vtičnico, ICSP-glavo in gumb za ponastavitev. Vsebuje vse, kar je potrebno za podporo mikrokrmilnika. Plošča Mega 2560 je združljiva z večino ščitov za Uno.

### 3.1.3 Arduinovo integrirano razvojno okolje (IDE)

Je aplikacija za Windows, MacOS, Linux, ki je napisana v programskem jeziku Java. Izvira iz integriranega razvojnega okolja za programska jezika Processing in Wiring. Vključuje urejevalnik kod s funkcijami, kot so izrezovanje in lepljenje besedila, iskanje in zamenjava besedila, samodejno zamikanje, ujemanje spojev, avtomatsko urejanje kode in poudarjanje sintaks. Omogoča prevajanje in nalaganje skripte na Arduino vezje z enim klikom. Vsebuje tudi območje za sporočila, besedilno konzolo, orodno vrstico gumbi za skupne funkcije in hierarhijo menijev.

Arduino IDE podpira jetike C in C++ s posebnimi pravili strukturiranja kode. Programsko knjižnico dobavlja iz projekta Wiring, ki ponuja številčne skupne vhodne in izhodne postopke. Uporabniško napisana koda zahteva samo dve osnovni funkciji, za zagon skice in glavne programske zanke, ki sta *main()* in *loop()*. Arduino IDE uporablja program Avrdude za pretvorbo izvršljive kode v besedilno datoteko v šestnajstiskem kodiranju, ki se naloži na Arduino ploščo s programom za nalaganje.



Slika 1: Arduinovo IDE-okolje

## 3.2 C++

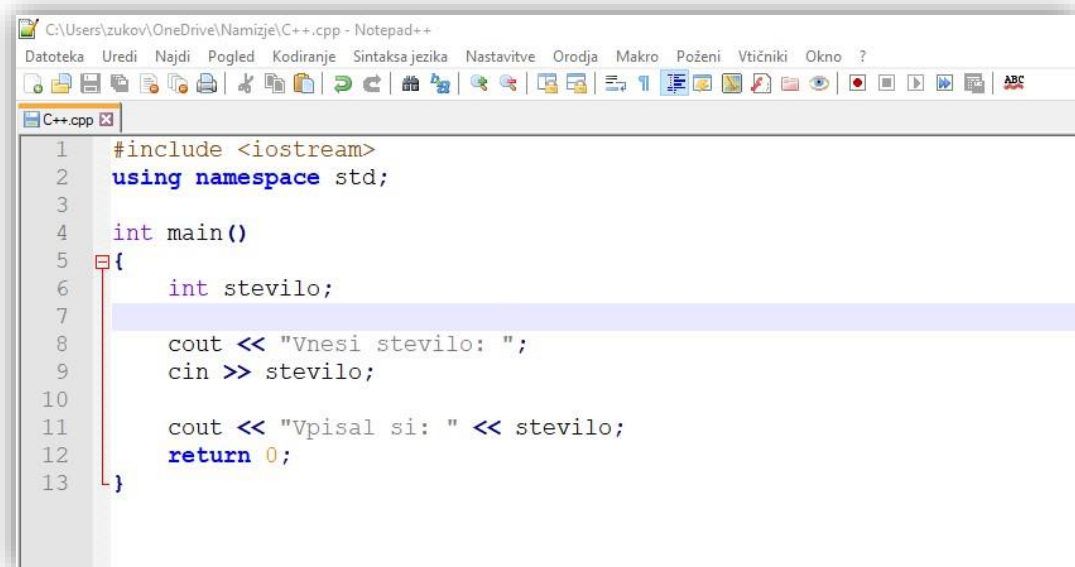
Programski jezik C++ je izboljšana verzija jezika C, ki se običajno uporablja za objektno usmerjeno programiranje. Razvija se že več kot trideset let. Čeprav je to daleč od najstarejšega računalniškega jezika, je to eden starejših, ki je danes v splošni rabi. Zato lahko rečemo da je dobil petko ta sposobnost prilagajanja spreminjajočim se tehnološkim časom.

C++ je razvil Bjarne Stroustrup, ki je opravil prvi prototip kot del svojega doktorskega projekta. V zgodnjih letih se je imenoval jezik »C z razredi«. Začel je razvijati nov jezik, ker je menil, da noben obstoječ ni bil idealen za velike projekte. Kasneje, ko je delal v AT&T Bell Labs, se je spet počutil omejeno. Zavrgel je svoj »C z razredi« in upeljal še druge jezike. Na koncu je poleg razredov dodal še virtualne funkcije, predloge (templates) in preobremenitev operatorja (operator overloading).

C++ je precej presegel delo enega človeka. Ime se je dejansko spomnil drug razvijalec Rick Mascitti. Delno je bila igra besed na ime operaterja »++« in delno sklicevanje na izboljšavo (dva plusa sta morda tudi šala).

Jezik je bil prvič standardiziran leta 1998. Standardi si bili ponovno obnovljeni leta 2003, 2007 in 2011. Jezik C++ zdaj vzdržuje International Organization for Standardization (ISO), ki je velika organizacija za standarde. Trenutna različica programskega jezika je C++20. Glavni cilji za zdaj so, da postane boljši jezik za vgrajene sisteme in da bolje podpira novince.

C++ se pogosto uporablja v programskem inženirstvu vgrajenih sistemov. Priljubljen je tudi v komunikacijskih aplikacijah in igrah. Uporablja se v mnogih različnih panogah, kot so zdravstvo, finance... Eden od razlogov, da se programerji odločijo za C++, je, da se dobro povezuje z drugimi programskimi jeziki. Še en plus je njegova visoka zmogljivost. C++ je vplival tudi na kasnejše jezike, kot so PHP, Java, C#... (5)



```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int stevilo;
7
8      cout << "Vnesi stevilo: ";
9      cin >> stevilo;
10
11     cout << "Vpisal si: " << stevilo;
12     return 0;
13 }
```

Slika 2: Primer kode v jeziku C++

### 3.3 Notepad++

Notepad++ je brezplačen urejevalnik izvorne kode in nadomestek za beležnico, ki podpira veliko več programskih jezikov. Temelji na zmogljivi komponenti za urejanje Scintilla, napisan je v jeziku C++ in uporablja Win32 API in STL, ki zagotavlja višjo hitrost izvajanja in manjšo velikost programa. Z dobro optimizacijo poskuša Notepad++ uporabniku ponuditi najboljšo možno storitev z minimalno procesorsko močjo (6).



## 3.4 3D-tiskanje

### 3.4.1 Delovanje 3D-tiskanja

Vsak 3D-tiskalnik deluje na isti glavni princip. Digitalni model se spremeni v fizični tridimenzionalni objekt z dodajanjem materiala (plastike) plast na plast. 3D-tiskanje je bistveno drugačen način izdelave delov v primerjavi s tradicionalnimi tehnologijami izrezovanja (CNC obdelava) ali proizvodnimi tehnologijami (injection molding). Pri 3D tiskanju niso potrebna posebna orodja (npr. rezalno orodje z določeno geometrijo ali kalup). Namesto tega je izdelan neposredno na vgrajeno platformo, po slojih, kar vodi do edinstvenih nizov prednosti in omejitev.

Proces se vedno začne z digitalnim 3D-modelom (načrtom fizičnega objekta). Model s programsko opremo tiskalnika razdelimo v tanke, dvodimenzionalne plasti, ki jih nato pretvori v niz navodil, da jih lahko tiskalnik tiska.

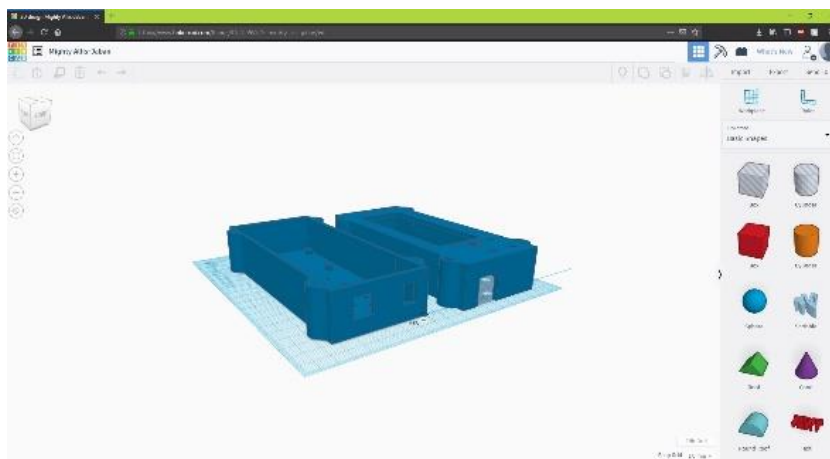
3D-tiskanje se lahko razlikuje odvisno od procesa, s katerim tiskamo. Na primer, namizni računalniki raztapljajo plastične filamente in jih položijo na tiskalno platformo skozi šobo. Veliki industrijski SLS stroji pa uporabljajo laser za taljenje (ali sintranje) tankih plasti kovinskih ali plastičnih prahov. Razpoložljivi materiali se razlikujejo tudi glede na proces. Plastika je daleč najpogostejša, uporabljamo pa lahko tudi kovino. Proizvedeni deli imajo lahko tudi širok spekter specifičnih lastnosti, od optično čistih, do gumastih predmetov. Odvisno od velikosti dela in vrste tiskalnika lahko traja tiskanje od štiri do osemnajst ur. Tiskani deli pa so redko pripravljeni za uporabo takoj po tisku. Velikokrat zahtevajo še nekaj naknadne obdelave, da se doseže želena raven kvalitete. Ti koraki zahtevajo dodaten čas in običajno ročen napor (7).

### 3.4.2 Zgodovina 3D-tiskalnikov

Arthur C. Clarke je prvič opisal osnovne funkcije 3D-tiskalnika leta 1964. Prvi 3D tiskalnik je leta 1987 izdelal Chuck Hull, uporabljal pa je sistem »stereolitografije«. V devetdesetih in dva tisočih so bile izdelane druge tehnologije 3D-tiskanja (FDM in SLS). Ti tiskalniki so bili dragi in se v glavnem uporabljajo za industrijsko izdelavo prototipov. Odbor za *ASTM F42* je leta 2009 objavil dokument s standardno terminologijo o aditivni proizvodnji. To je vzpostavilo 3D tiskanje kot industrijsko proizvodnjo tehnologijo. V istem letu so se patenti za *FDM* iztekli in prvi poceni 3D namizni tiskalniki so začeli nastajati pod okvirjem projekta RepRap. Kar je nekoč stalo 200.000€, je nenadoma postalo dostopno za manj kot 2.000€. V obdobju 2015-2017 je bilo globalno prodanih več kot 1 milijon 3D-tiskalnikov, prodaja industrijskih kovinskih tiskalnikov pa se je leta 2017 v primerjavi s prejšnjim letom skoraj podvojila.

### 3.4.3 Tinkercad

Ustanovljen je bil leta 2011. Je največja skupnost 3D-oblikovalcev in navdušencev 3D-tiskanja. Do zdaj je privabil več kot sedem milijonov uporabnikov, vsako minuto pa se pridružijo novi. Aktivna skupnost je skupaj ustvarila že več kot petdeset milijonov 3D modelov, vsak dan pa je dodanih še na tisoče novih. Konec leta 2016 je prvotni Tinkercad urednik predstavil popolnoma prenovljen urejevalnik v2.0, ki deluje 800 % hitreje kot prej in nam omogoča, da povečavo aplikacije na načine, ki jih stari ne more. Junija 2013 je Tinkercad postal del družine aplikacij Autodesk (8).

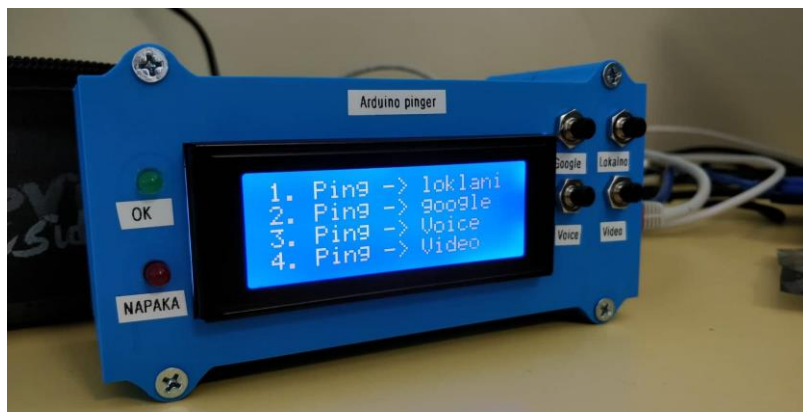


Slika 3: Ohišje Arduino pingerja v programu Tinkercad

## 4. Delovanje

### 4.1 Program in funkcije

Program je napisan v Arduinovem IDE-okolju. Glavne funkcije so enkraten ali konstanten *ping* lokalnega usmerjevalnika, Googlevega strežnika, telefonske ali video mreže Telekoma Slovenije. Rezultat se izpiše na LCD-zaslonu, za hitrejše razumevanje pa besedilo na zaslonu podpirata se ustrezno prižgani rdeča ali zelena dioda.



Slika 4: Osnovni meni Arduino pingerja

### 4.1.1 Osnovne deklaracije



Izven vseh funkcij se deklarirajo globalne spremenljivke, ki se bodo uporabljale v več kot eni funkciji. Z *#include* pa se navežemo na knjižnice, ki jih Arduino IDE v osnovi še nima vključenih. Navezati sem se moral na knjižnice za delovanje internetnega ščita in LCD-zaslona.

### 4.1.2 Funkcija rebootEthShield

```
int rebootEthShield()
{
  lcd.clear();
  pinMode(7, OUTPUT);
  digitalWrite(7, HIGH);
  delay(50);
  digitalWrite(7, LOW);
  delay(50);
  digitalWrite(7, HIGH);
  delay(100);
  byte macData[] = {0xaa, 0xbb, 0xcc, 0x12, 0x34, mac};
  lcd.setCursor(0, 0);
  lcd.print("-----");
  lcd.setCursor(0, 3);
  lcd.print("-----");
  lcd.setCursor(0, 2);
  lcd.print("|           |");
  lcd.setCursor(0, 1);
  lcd.print("|   Pridobivam IP   |"); // <-- Fill in your Ethernet shield's MAC address here.
  Ethernet.begin(macData);
  lcd.clear();
  IPAddress Gateway = Ethernet.gatewayIP();
  if (Gateway[0] == 192) {
    lcd.setCursor(0, 0);
    lcd.print("-----");
    lcd.setCursor(0, 3);
    lcd.print("-----");
  }
}
```

Slika 6: Primer funkcije Arduino pingerja

Funkcija se sproži, ko v tester omrežja (Arduino pinger) vključimo internetni kabel. V funkciji resetiramo internetni ščit (ethernet shield), da dobi nov logični naslov (IP) s katerim bo lahko dostopal na internet oziroma glasovni ali pa video strežnik Telekoma Slovenije.

### 4.1.3 Funkcije googleKratek



Slika 7: Prikaz funkcije googleKratek na zaslonu Arduino pingerja

S kratkim klikom na gumb Google bo tester omrežja *pingal* strežnik Googla z logičnim naslovom 8.8.8.8 in izpisal ustrezno besedilo ter prižgal rdečo ali zeleno LED-diodo.

### 4.1.4 Funkcija localKratek



Slika 8: Prikaz funkcije localKratek na zaslonu Arduino pingerja

S kratkim klikom na gumb Lokalno bo tester omrežja *pingal* lokalni usmerjevalnik z logičnim naslovom 192.168.1.1 in izpisal ustrezno besedilo ter prižgal rdečo ali zeleno LED-diodo.

### 4.1.5 Funkcija videoKratek



Slika 9: Prikaz funkcije videoKratek na zaslonu Arduino pingerja

S kratkim klikom na gumb video bo tester omrežja *pingal* video server Telekoma Slovenije z logičnim naslovom 10.253.3.120 in izpisal ustrezno besedilo ter prižgal rdečo ali zeleno LED-diodo.

### 4.1.6 Funkcija voiceKratek

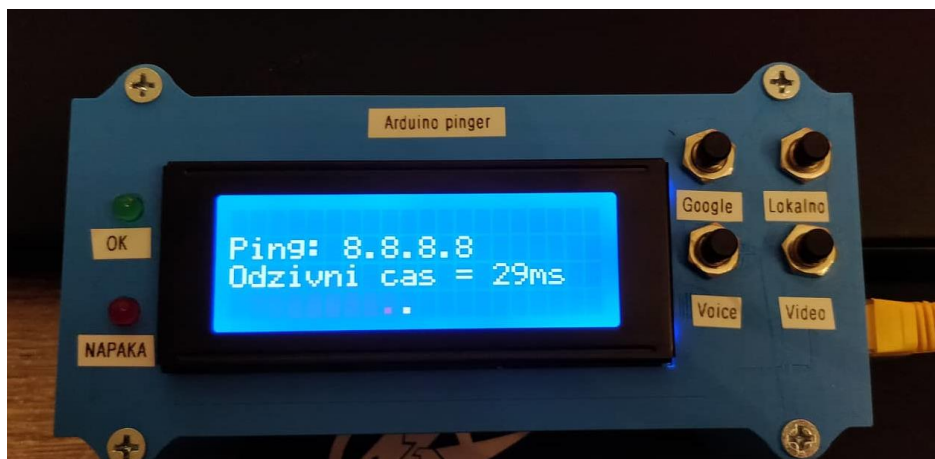


Slika 10: Prikaz funkcije voiceKratek na zaslonu Arduino pingerja

S kratkim klikom na gumb video bo tester omrežja *pingal* voice server Telekoma Slovenije z logičnim naslovom 10.253.0.1 in izpisal ustrezno besedilo ter prižgal rdečo ali zeleno LED-diodo.



### 4.1.7 Funkcija googleDolgi



Slika 11: Prikaz funkcije googleDolgi na zaslonu Arduino pingerja

Z dolгим pritiskom na gumb Google (vsaj dve sekundi) bo tester omrežja začel konstantno *pingati* strežnik Google z logičnim naslovom 8.8.8.8, dokler gumba spet ne pridržiimo za vsaj dve sekundi. Med *pinganjem* izpisuje IP, ki ga *pingamo*, odzivni čas ter indikator s pomočjo katerega lahko izvemo, da *pinganje* še poteka. V primeru napake v povezavi na LCD-zaslon izpiše »...Ping neuspesen...«

### 4.1.8 Funkcija localDolgi



Slika 12: Prikaz funkcije localDolgi na zaslonu Arduino pingerja

Z dolгим pritiskom na gumb Lokalno (vsaj dve sekundi) bo tester omrežja začel konstantno *pingati* lokalni usmerjevalnik z logičnim naslovom 192.167.1.1, dokler gumba spet ne pridržiimo za vsaj dve sekundi. Med *pinganjem* izpisuje IP, ki ga *pingamo*, odzivni čas ter indikator s pomočjo katerega lahko izvemo, da *pinganje* še poteka. V primeru napake v povezavi na LCD-zaslon izpiše »...Ping neuspesen...«

#### 4.1.9 Funkcija videoDolg



Slika 13: Prikaz funkcije videoDolg na zaslonu Arduino pingerja

Z dolгим pritiskom na gumb Video (vsaj dve sekundi) bo tester omrežja začel konstantno *pingati* video strežnik Telekoma Slovenije z logičnim naslovom 10.253.3.120, dokler gumba spet ne pridržimo za vsaj dve sekundi. Med *pinganjem* izpisuje IP, ki ga *pingamo*, odzivni čas ter indikator s pomočjo katerega lahko izvemo, da *pinganje* še poteka. V primeru napake v povezavi na LCD-zaslon izpiše »...Ping neuspesen...«

#### 4.1.10 Funkcija voiceDolg



Slika 14: Prikaz funkcije voiceDolg na zaslonu Arduino pingerja

Z dolгим pritiskom na gumb Video (vsaj dve sekundi) bo tester omrežja začel konstantno *pingati* voice strežnik Telekoma Slovenije z logičnim naslovom 10.253.0.1, dokler gumba spet ne pridržimo za vsaj dve sekundi. Med *pinganjem* izpisuje IP, ki ga *pingamo*, odzivni čas ter indikator s pomočjo katerega lahko izvemo, da *pinganje* še poteka. V primeru napake v povezavi na LCD-zaslon izpiše »...Ping neuspesen...«

### 4.1.11 Funkciji loop in setup

V funkciji *loop* čakamo na klik gumba in posredno kličemo ustrezno funkcijo. V *setupu* pa deklariramo spremenljivke in kličemo vse kar potrebujemo za delovanje ostalih funkcij.

### 4.1.12 Preverjanje funkcij gumba

```
buttonState = 0;
if (digitalRead(YOUR_RESET_BUTTON_PIN)) {
    prev = millis();
    buttonState = 1;
    while ((millis() - prev) <= DURATION_IN_MILLIS) {
        if (!(digitalRead(YOUR_RESET_BUTTON_PIN))) {
            buttonState = 2;
            break;
        }
    }
}
```

Slika 15: Odsek kode, ki preverja koliko časa je bil gumb kliknjen

```
//klice ustezno funkcijo
if (!buttonState)
{
    //NIČ NI PRITISNENO;
}
else if (buttonState == 1)
{
    if (Gateway[0] == 192)
    {LocalDolg();}
}
else if (buttonState == 2)
{
    if (Gateway[0] == 192)
    {LocalKratek();}
}
```

Slika 16: Odsek kode, kliče ustrezno funkcijo

Na prvi sliki je prikazana koda, s pomočjo katere preverjamo, koliko časa je določen gumb pridržen. To dosežemo z uporabo funkcije *millis()*. Na drugi sliki pa lahko vidimo, kako se po izvedbi kode iz prve slike izvede if-stavek s katerim se kliče ustrezna funkcija.

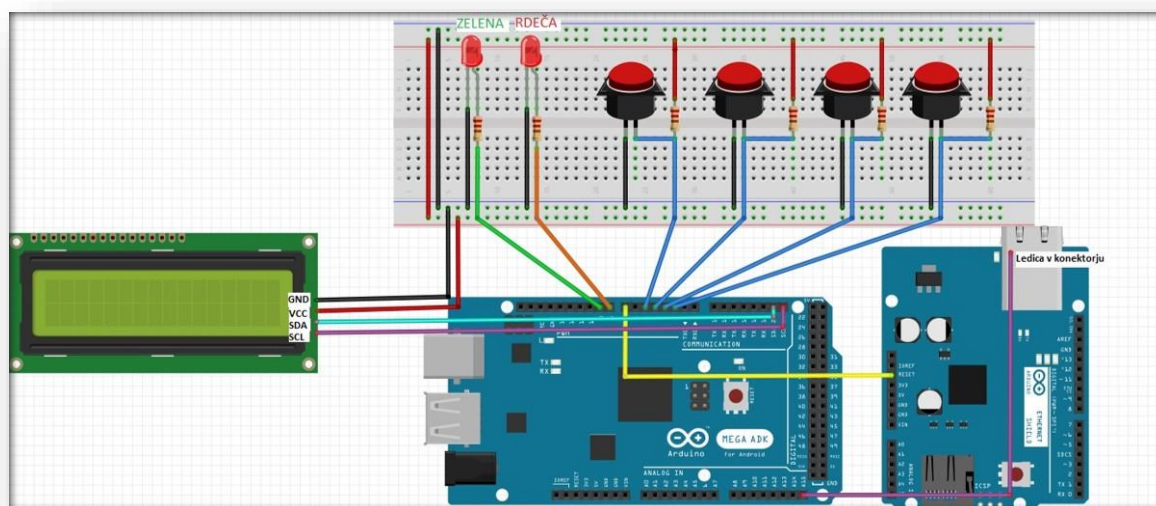


## 4.2 Sestavni deli

Arduino pinger je sestavljen iz štirih gumbov, dveh LED-diod, LCD-zaslona, Arduino atmege, internetnega ščita in spajkalne plošče.



Slika 17: Sestavni deli Arduino pingerja



Slika 18: Vezje Arduino pingerja

## 5. Težave pri razvoju

Pri razvoju sem naletel na le nekaj večjih težav. Prva od teh je bila, da sem imel starejšo različico internetnega ščita, ki ni podpirala ponastavitve preko programske kode, kar pomeni, da sem moral nogico za *resetiranje* zakriviti, tako da se ni usedla na Arduino, ampak je bila prosta. Nato sem nanjo spajkal kabel, ki sem ga vključil v prost vtič Arduinota. Če bi nogico pustil na prvotnem mestu, bi se namreč ščit ponastavil samo skupaj z Arduinom, tako pa sem imel popolno kontrolo nad ponastavitvijo. Drugo večjo težavo pa so mi povzročali gumbi, saj nisem imel dovolj znanja, da bi z enim gumbom zagotovil izvajanje dveh funkcij. S težavo sem se spopadal en teden, preden sem dobil zadovoljivo rešitev. Med izdelavo sem naletel še na obilico majhnih težav, ki sem jih skupaj z mentorjem hitro odpravil.

## 6. Zaključek

Kljub možnostim izboljšave sem z Arduino pingerjem zadovoljen. Lahko bi dodal še številčnico za poljubno vpisovanje logičnih naslovov, a bi bila taka naprava mogoče za nekatere preveč zapletena za uporabo. Namen projekta je bil, da sestavim tester omrežij, ki je preprost za uporabo, a vseeno dovolj zmogljiv, da lahko tehnikom na terenu pomaga pri iskanju napak ali rešitev. Poskušal sem ga narediti čimbolj uporabniku prijaznega in zdi se mi, da mi je to tudi uspelo. Nima odvečnih animacij, uporaba je preprosta in prikaz je lahko razumljiv. V prihodnosti bi rad dodal še notranjo baterijo in izboljšal ohišje oziroma ga poskušal narediti tanjšega.

## 7. Viri in literatura

Vsi internetni viri so bili dosegljivi aprila 2019.

- (1) [http://wiki.fmf.uni-lj.si/wiki/Vrste\\_omre%C5%BEja](http://wiki.fmf.uni-lj.si/wiki/Vrste_omre%C5%BEja)
- (2) <https://www.techopedia.com/definition/23525/home-networking>
- (3) <https://www.techopedia.com/definition/26424/public-network>
- (4) <https://www.arduino.cc/>
- (5) <https://www.softwareengineerinsider.com/programming-languages/cplusplus.htm>
- (6) <https://notepad-plus-plus.org/>
- (7) <https://www.3dhubs.com/guides/3d-printing/#basics>
- (8) <https://www.tinkercad.com/about/>

### 7.1 Viri slik

- Slika na naslovnici: lastna slika, zajeta aprila 2019
- Slike 1–16: lastne slike, zajete aprila 2019