# Bezdodirni interaktivni uređaj

**Hmura, Antonio**

**Master's thesis / Diplomski rad**

**2019**

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

*Permanent link / Trajna poveznica:* https://urn.nsk.hr/urn:nbn:hr:200:775449

*Rights / Prava:* In copyright / Zaštićeno autorskim pravom.

*Download date / Datum preuzimanja:* **2025-03-21**

*Repository / Repozitorij:*

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

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

**Sveučilišni studij**

# BEZDODIRNI INTERAKTIVNI UREĐAJ

**Diplomski rad**

**Antonio Hmura**

**Osijek, 2019.**

# SADRŽAJ


1. UVOD ..... 1
2. ELEKTRIČNI KAPACITET ..... 2
   2.1. Definicija i kondenzatori ..... 2
3. VRSTE ZASLONA ..... 4
   3.1. Zasloni osjetljivi na dodir ..... 4
   3.2. Bezdodirni zasloni ..... 6
4. PROTOTIP UREĐAJA ZA BEZDODIRNO UPRAVLJANJE ..... 7
   4.1. Opis i funkcionalnost uređaja ..... 7
   4.2. Sklop za detekciju promjene električnog polja ..... 13
   4.3. Rad sklopa za detekciju promjene električnog polja i utjecaj frekvencije ..... 16
   4.4. Programski kod i logika izvođenja ..... 19
5. FUNKCIONALNOST I ANALIZA RADA ..... 23
   5.1. Funkcionalnost sustava i alternativni način razvoja ..... 23
   5.2. Analiza rada ..... 24
6. ZAKLJUČAK ..... 27

LITERATURA ..... 28

SAŽETAK ..... 29

ABSTRACT ..... 30

ŽIVOTOPIS ..... 31

PRILOZI ..... 32

# 1. UVOD

Interakcija ljudi sa zaslonima raznih računalnih sustava u današnjem modernom svijetu postala je svakodnevica. Neovisno radi li se o provjeravanju novosti na društvenim mrežama na zaslonu osjetljivom na dodir svog pametnog telefona ili o upravljanju načina vožnje u modernom automobilu koristeći bezdodirni hologram, čovjek se navikao da pokretom svoga prsta ili ruke obavlja određene funkcije nekog tehnološkog sustava.

U ovom diplomskom radu riješen je zadatak razvoja bezdodirnog interaktivnog zaslona temeljenog na kapacitivnim senzorima položaja. Izrađen je prototip uređaja kojim korisnik upravlja osnovnim funkcijama računala pomakom svoje ruke kroz kućište prototipa. Smještanjem svoje ruke unutar uređaja određuje se pomak miša u ovisnosti o položaju ruke unutar kućišta.

Rad je strukturiran na način da su prvo opisane elektroničke i fizikalne pojave koje se odvijaju unutar cijelog sustava, te na kojim se principima temelji razvijani uređaj. Nakon navedenih dosadašnjih rješenja problema interakcije korisnika s računalnim sustavom objašnjena je konstrukcija uređaja za bezdodirnu interakciju, te korištena Arduino razvojna pločica. Obrađen je i opisan cijeli proces prepoznavanja pokreta unutar uređaja, te obrada signala pomoću sklopova na elektrodama uređaja.

Za bolje razumijevanje osnovnog koncepta na kojemu je zasnovan cijeli sustav bezdodirnog interaktivnog uređaja preporučuje se proučavanje završnog rada na temu „Dizajn elektroda za kapacitivni senzor položaja“. Navedeni rad istog autora služio je kao temelj za razvoj ovoga sustava, te je služio u početnoj fazi izrade bezdodirnog interaktivnog uređaja [1].

# 2. ELEKTRIČNI KAPACITET

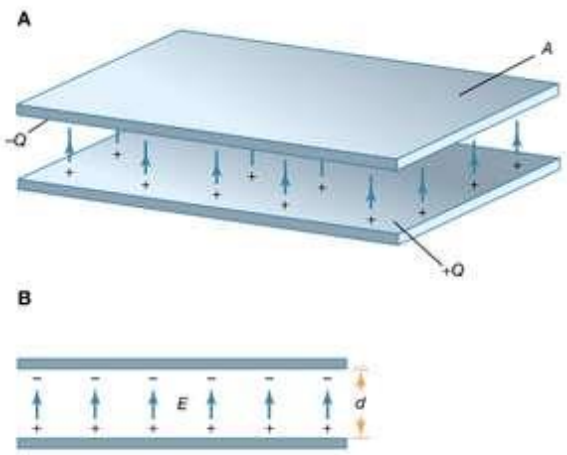
## 2.1. Definicija i kondenzatori

Električni kapacitet je fizikalna veličina kojom se opisuje sposobnost pohranjivanja naboja, a time i energije električnog polja. To je svojstvo električnog vodiča koje se mjeri količinom odvojenog električnog naboja koje se može pohraniti na njega po jedinici promjene električnog potencijala. Kapacitet također podrazumijeva i skladištenje električne energije. Ako se električni naboj prenosi između dva početno nenabijena vodiča, oba postaju jednako nabijena, jedan s pozitivnim nabojem, drugi s negativnim nabojem, te se između njih stvara razlika potencijala. Električni kapacitet kondenzatora određen je kvocijentom količine naboja koji se može pohraniti u kondenzatoru određenog primijenjenog napona. Mjerna jedinica za električni kapacitet vodiča po SI sustavu je Farad (kulon po voltu), naziv dobiva po engleskom fizičaru Michael Faraday-u. Kako jedan farad predstavlja veliku količinu kapaciteta, u praksi se najčešće pojavljuju iznosi koji predstavljaju milijunti dio farada, mikrofarad μF, te milijunti dio mikrofarada, pikofarad pF. Općenito vrijedi matematički izraz:

$$C = \frac{Q}{U}\ [F] \tag{2-1}$$

gdje je $C$ električni kapacitet u faradima (F), $Q$ je naboj vodiča u kulonima (C) i $U$ je razlika potencijala u voltima (V).

Kondenzator je poluvodička komponenta namijenjena za spremanje električne energije koja se sastoji od dva vodiča koji se međusobno nalaze u malom razmaku jedan od drugoga, te je između njih smješten izolator. Jednostavan primjer takve komponente za skladištenje električne energije je kondenzator s paralelnim pločama (Sl. 2.1.). Ako na jednoj ploči postoji pozitivni naboj određenog iznosa, te na drugoj ploči postoji negativni naboj istog iznosa, kaže se da kondenzator ima toliki iznos naboja koji se nalazi na svakoj ploči. Kondenzatori imaju široku primjenu. Na primjer, koriste se u digitalnim sklopovima za skladištenje električne energije i njeno oslobađanje u slučaju trenutnog nestanka električne energije u cijelom sustavu. Također služe i kao filtri raznih signala, sprječavanju oštećenja osjetljivih električnih komponenti i krugova itd. [2,3].

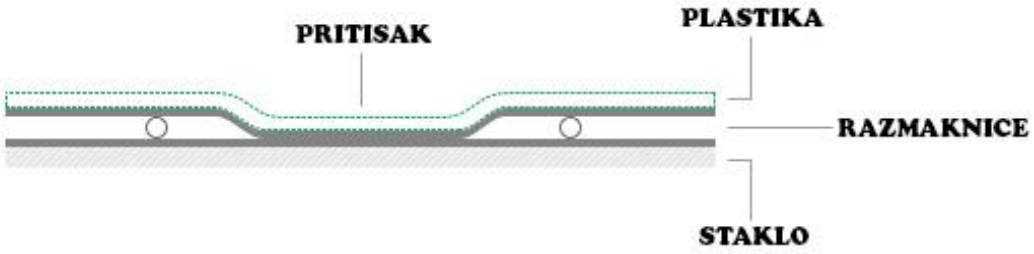



Sl. 2.1. Primjer jednostavnog kondenzatora s dvije paralelne ploče, gdje je *A* površina ($m^2$), *Q* naboj (C), *E* električno polje (V/m), *d* razmak između ploča (m) [3].

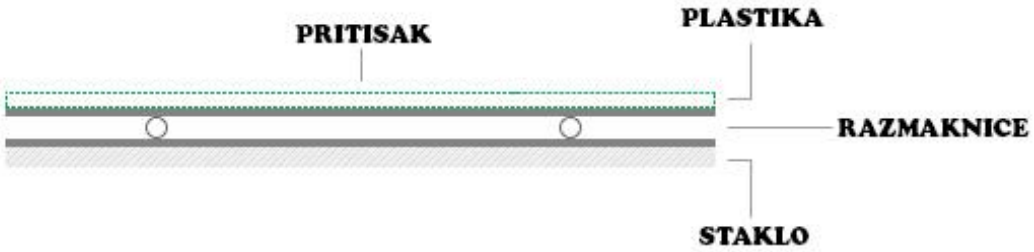
# 3. VRSTE ZASLONA

## 3.1. Zasloni osjetljivi na dodir

Postoje različite vrste zaslona osjetljivih na dodir, a razlikuju se po tome na koji način obavljaju prepoznavanje dodira na površini. Otpornički, kapacitivni, infracrveni, zasloni s površinskim akustičnim valom su samo neki od zaslona osjetljivih na dodir. Otpornički zasloni osjetljivi na dodir su danas najčešća vrsta tehnologije zaslona osjetljivih na dodir. Na vrhu zaslona se postavlja sloj provodne poliesterske plastike koji je od samog zaslona odvojen izolacijskom membranom. Pritiskom na zaslon zatvara se strujni krug spajanjem sloja poliestera i samog zaslona uređaja, te se podatak o položaju pritiska obrađuje u samom uređaju. Kapacitivni zasloni osjetljivi na dodir su izrađeni od nekoliko slojeva stakla. Unutarnji i vanjski slojevi provode električnu energiju, između kojih se nalazi izolator, staklo, pa se tako ponaša kao kondenzator. Dodirom zaslona se pobuđuje električno polje, te se iznosom promjene određuju koordinate dodira zaslona. Infracrveni zasloni osjetljivi na dodir se sastoje od mreže dioda koje šalju infracrvene valove, odnosno fotoćelija za detekciju signala sa suprotne strane. Dodirom unutar te mreže na površini zaslona prekida se određeni broj infracrvenih signala, te se na temelju toga dobiva podatak o točnom položaju dodir. Velika prednost korištenja tehnologije zaslona osjetljivih na dodir u odnosu na tipkovnice je što ju ljudi vrlo lako koriste. Zasloni osjetljivi na dodir uglavnom prikazuju točno onoliko informacija koliko je potrebno za obavljanje nekakvog zadatka, vodeći korisnike kroz komplicirane procese na vrlo jednostavan i sistematičan način [4]. Primjeri otporničkih (Sl. 3.1.), kapacitivnih (Sl. 3.2.), te infracrvenih (Sl. 3.3.) zaslona osjetljivih na dodir dani su na sljedećim slikama [5].




Sl. 3.1. Primjer presjeka otporničkog zaslona osjetljivog na dodir [5].




Sl. 3.2. Primjer presjeka kapacitivnog zaslona osjetljivog na dodir [5].

Sl. 3.3. Primjer presjeka infracrvenog zaslona osjetljivog na dodir [5].

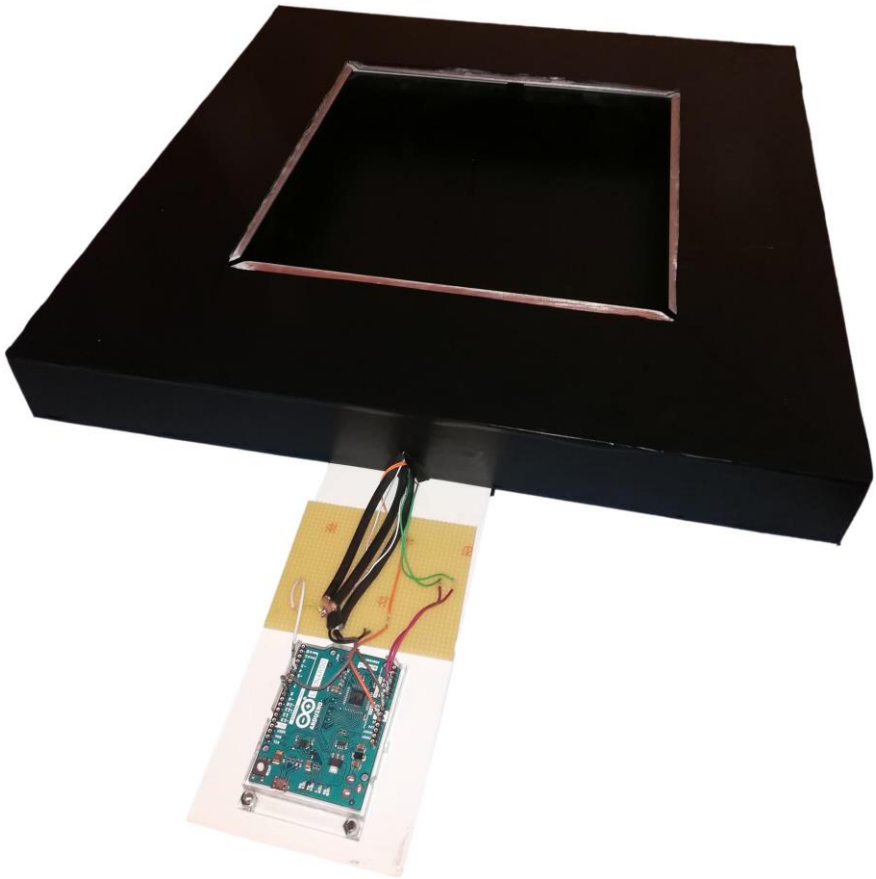
## 3.2. Bezdodirni zasloni

Tehnologija zaslona bez dodira koristi prepoznavanje pokreta ruke ili prsta u blizini zaslona bez kontakta s njim. Današnja rješenja ove tehnologije podrazumijevaju korištenje jednog ili više senzora u blizini zaslona koji prepoznaje pokrete korisnika koji želi njime upravljati. Tehnologija bezdodirnih zaslona ne koristi nikakvu dodatnu opremu ili senzore koje bi korisnik morao nositi ili držati u svojoj ruci. Općenito, uređaj s ovakvom tehnologijom se temelji na optičkom prepoznavanju uzoraka pomoću senzora optičke matrice s lećom za detektiranje pokreta ruke. Taj je senzor spoje na procesor digitalne slike, koji interpretira određene uzorke kretanja, te kao izlaz daje signale za kontrolu uređaja, strojeva, različitih sučelja itd. Takav uređaj prepoznaje pokrete bez dodirivanja zaslona. Senzori se nalaze oko zaslona. Ruka ili prst stvaraju sliku položaja unutar polja senzora. Senzor prepoznaje sliku, svjetlo dolazi do senzora, reflektira se u matricu piksela, te nakon toga pikseli pretvaraju ulazno svjetlo u električni naboj pomoću fotodiode. Senzori zatim generiraju električne signale koji se obrađuju unutar samog sustava kako bi korisnik dobio željeni izlaz ovisno o pokretu [6]. U ovom radu nije korišten skup senzora, već četiri kapacitivne elektrode postavljene na rubove kućišta koji imaju svoje pripadajuće elektroničke sklopove za prepoznavanje pobude unutar kućišta.

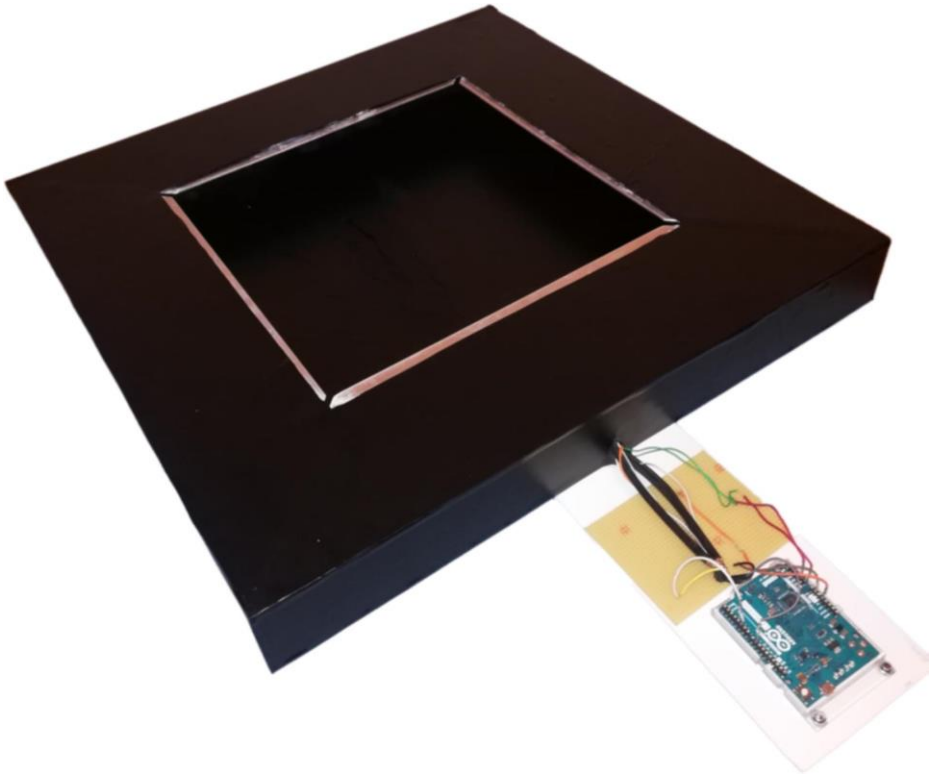
# 4. PROTOTIP UREĐAJA ZA BEZDODIRNO UPRAVLJANJE

## 4.1. Opis i funkcionalnost uređaja

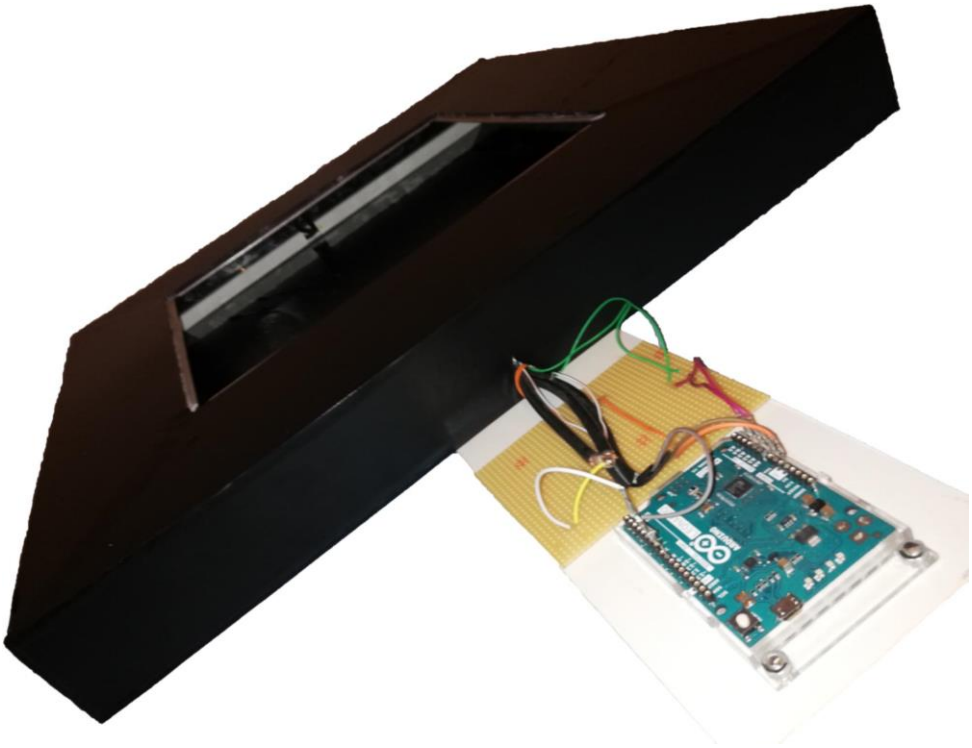
Uređaj kao cjelina je izrađen u obliku kvadratnog plastičnog kućišta koje na svakom rubu ima ugrađenu elektrodu. Vanjska širina i duljina kućišta iznose 50 cm. Dio okvira koji na sebi sadrži elektrode je od podnožja odvojen plastičnim profilima duljine 5 cm. Unutarnji dio koji je odvojen od podnožja ima otvor širine i duljine 30 cm. Na pojedinoj strani tog otvora su smještene elektrode koje služe za stvaranje električnog polja. Elektrode imaju oblike trapeza čija duljina veće stranice iznosi 38 cm, a duljina manje stranice iznosi 28 cm. Šiljasti kutovi tog trapeza iznose 45°, odnosno tupi kutovi iznose 135°. Kao materijal za kućište je odabrana plastika odnosno poli(vinil-klorid) debljine 3 mm, a materijal elektroda je aluminij debljine 2 mm. Na slikama (Sl. 4.1. – Sl. 4.5.) je prikazana konstrukcija prototipa uređaja za bezdodirnu interakciju.



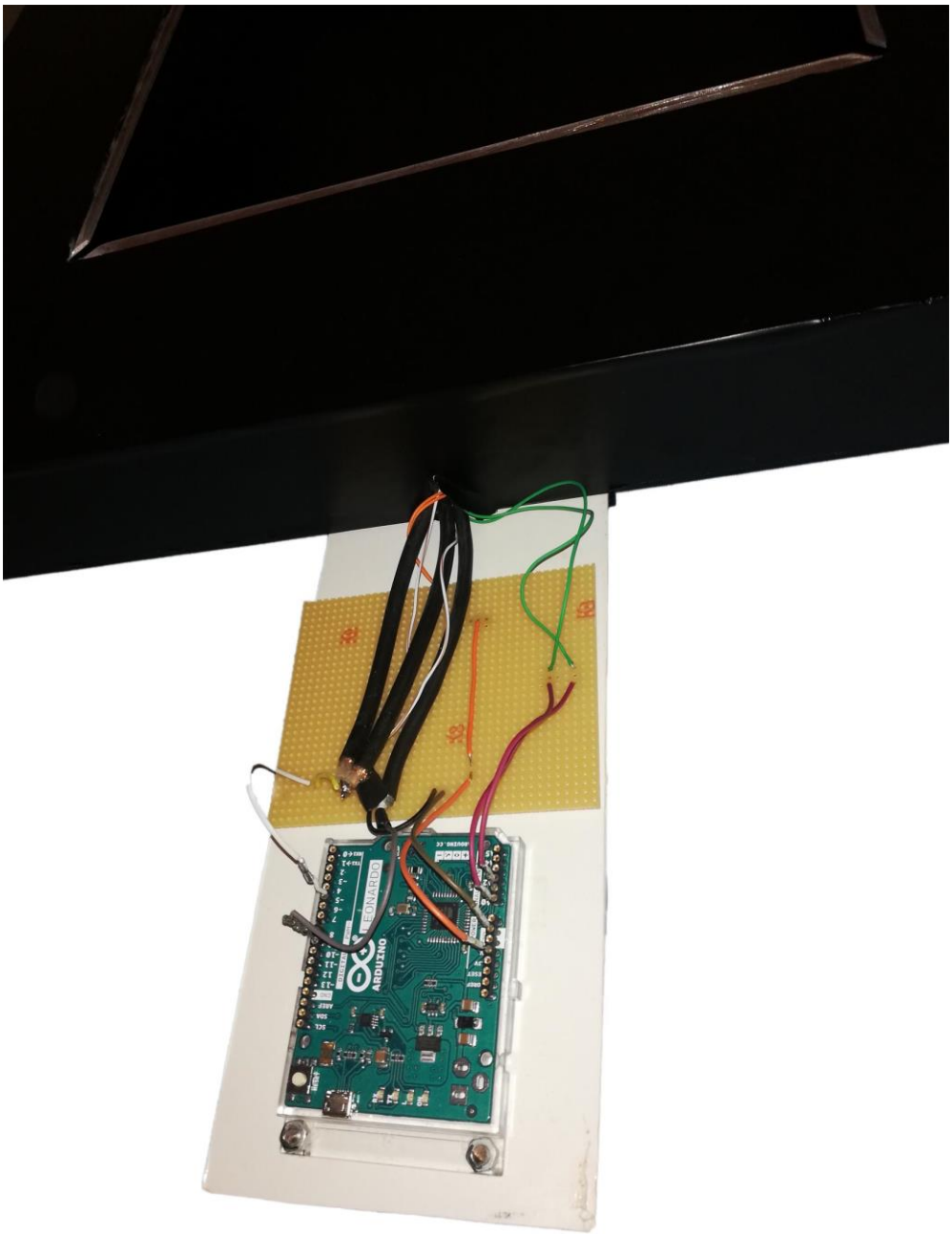
Sl. 4.1. Izrađen prototip uređaja za bezdodirno upravljanje



Sl. 4.2. Izrađen prototip uređaja za bezdodirno upravljanje



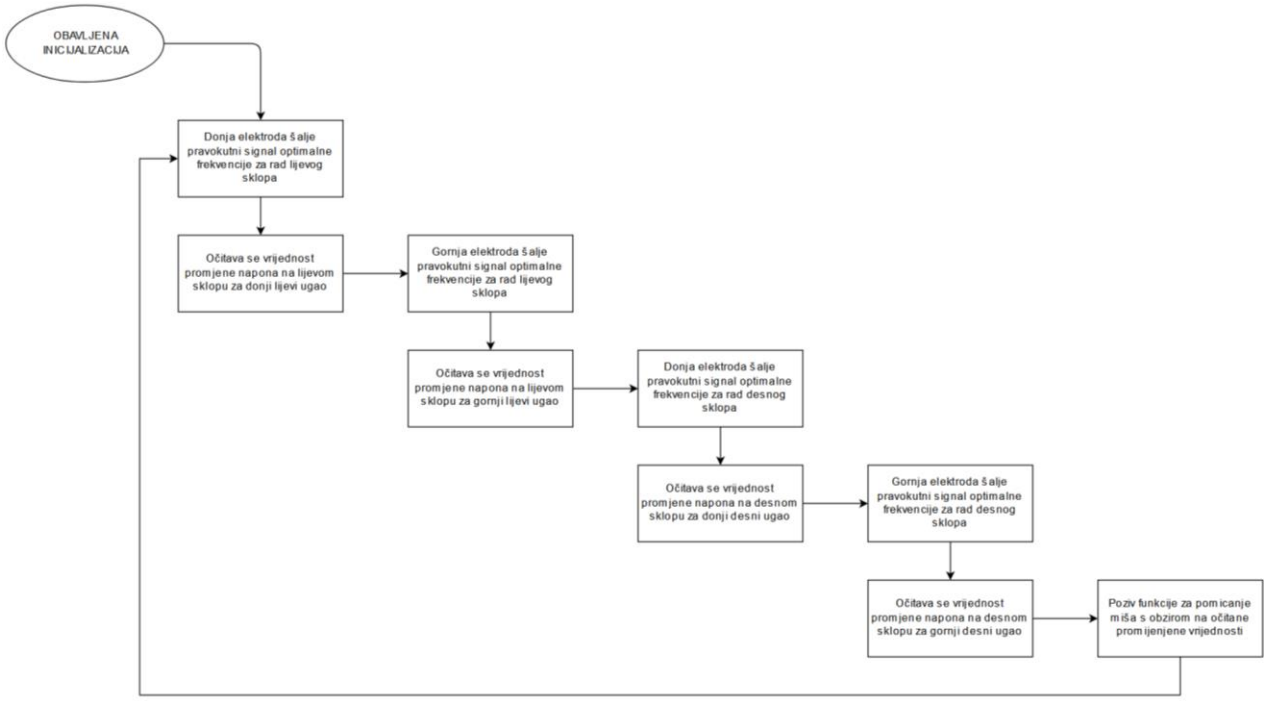
Sl. 4.3. Izrađen prototip uređaja za bezdodirno upravljanje




Sl. 4.4. Prikaz korištene Arduino platforme s izvodima iz uređaja

Zbog velike osjetljivosti cijelog sustava na uzemljenje tj. masu, morala se osigurati što manja interferencija elektroda koje daju referentni ulazni signal i elektroda koje daju izlazni signal promjene električnog polja. Iz toga razloga su dvije elektrode (smještene na gornju i donju poziciju kućišta) spojene na upravljački sklop odnosno mikrokontroler koaksijalnim kabelom oko unutrašnjosti kućišta kako ne bi došlo do ometanja izlaznih signala ostalih elektroda. Preostale dvije elektrode su spojene svaka na svoj odgovarajući sklop koji služi i kao pojačalo signala kojemu je glavna funkcija određivanje promjene električnog polja između elektrode na koju je spojen i elektrode koja šalje ulazni signal. Oba sklopa imaju svoje uzemljenje, izvod napajanja od 5 V, te izvod izlaznog signala za očitavanje promjene električnog polja. Ulazni signal za stvaranje električnog polja između elektroda je pravokutni signal određene frekvencije dobiven pronalaskom najbolje frekvencije rada sklopova u samom kodu mikrokontrolera. Unutar kućišta je cjelokupno ožičenje dovoljno odvojeno od elektroda kako ne bi došlo do interferencija.

Cijeli sustav je osmišljen na konceptu rada kondenzatora, tako što se između dviju ploča smještenih okomito jedna s obzirom na drugu nalazi dielektrik, zrak. Sustav funkcionira tako što se na gornju (odnosno donju) elektrodu šalje pravokutni signal amplitude 5 V određene frekvencije te se stvara električno polje između lijeve (odnosno desne) elektrode. Unošenjem ruke ili prsta unutar kućišta korisnik predstavlja određenu promjenu dielektrika u cijelom sustavu te svojom interakcijom mijenja električno polje među pločama. Ta promjena koju je korisnik pobudio unutar sustava svojom interakcijom se prepoznaje pomoću sklopova na elektrodama. Pobudom unutar tog sustava te prepoznavanjem između kojih elektroda je došlo do promjene može se odrediti položaj ruke. Na izlaznim signalima pojedine elektrode očitava se rast ili pad vrijednosti napona ovisno o tome prilazi li korisnik rukom ili se udaljava od pojedine elektrode. Elektrode koje daju ulazni signal rade naizmjenično kako ne bi došlo do potpunog punjenja električnog polja unutar cijelog sustava i nemogućnosti očitavanja promjene. Dijagram toka rada cijelog sustava prikazana je na slici (Sl. 4.5.) na kojoj su prikazani osnovni dijelovi izvršavanja koda koji su detaljno opisani u nastavku rada.




Slika 4.5. Dijagram toka rada cijelog sustava

Svi signali, te cijela funkcionalnost sustava obrađuje se korištenjem Arduino Leonardo razvojnog sustava zasnovanog na Atmega32u4. Odabran je navedeni mikrokontroler zbog jednostavnosti korištenja, programiranja i implementacije koristeći C programski jezik, te dovoljnoj moći i brzini rada za zadovoljavanje svih potreba ovog sustava. Arduino Leonardo se sastoji od 20 digitalnih ulaznih ili izlaznih izvoda, od kojih se 7 može koristiti kao izlaz s pulsno-širinskom modulacijom što je za ulazne signale sustava koji se opisuje u ovom diplomskom radu iznimno potrebno. Također se sastoji i od 12 analognih ulaza, od kojih su neki korišteni za dobivanje izlaznog signala sa bezdodirnog sustava, 16 MHz kvarcni kristal, USB priključak, priključak za dodatno napajanje, priključak za programator ATmega32u4 mikrokontrolera, te gumb za reset. Arduino Leonarno razvojni sustav je prikazan na slici (Sl. 4.6.) [7].

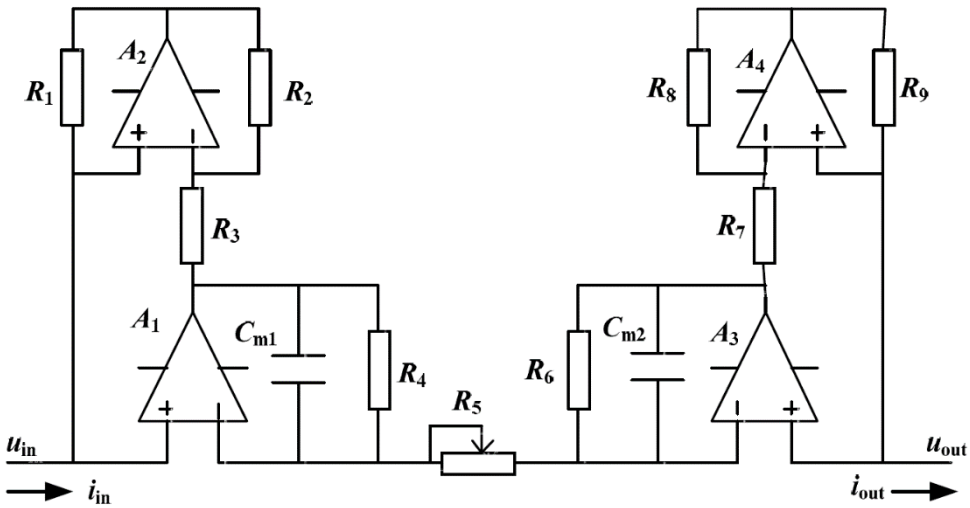
Sl. 4.6. Slika prikazuje korišteno Arduino Leonardo sučelje [7].

Odabrana Arduino platforma, Leonardo, temeljena na Atmega32u4 mikroprocesoru ima vrlo važnu funkcionalnost zbog koje se i koristi za bezdodirni interaktivni uređaj. Naime, uz Arduino Esplora, Zero, Due i MKR Family ploče, Arduino Leonardo ima mogućnost da ga računalo prepozna kao ulaznu jedinicu odnosno konkretno miš pri samom spajanju. Spajanjem navedene verzije Arduino pločice s računalnim sustavom dolazi do pripreme uređaja unutar računalnog sustava, te se on nakon toga može ponašati kao miš i koristiti sve funkcionalnosti miša pomoću „Mouse.h“ Arduino biblioteke. Biblioteka se sastoji od osnovnih funkcija za inicijalizaciju miša, prekida komunikacije, pomicanje pokazivača miša, lijevog klika, povlačenja itd. [8].

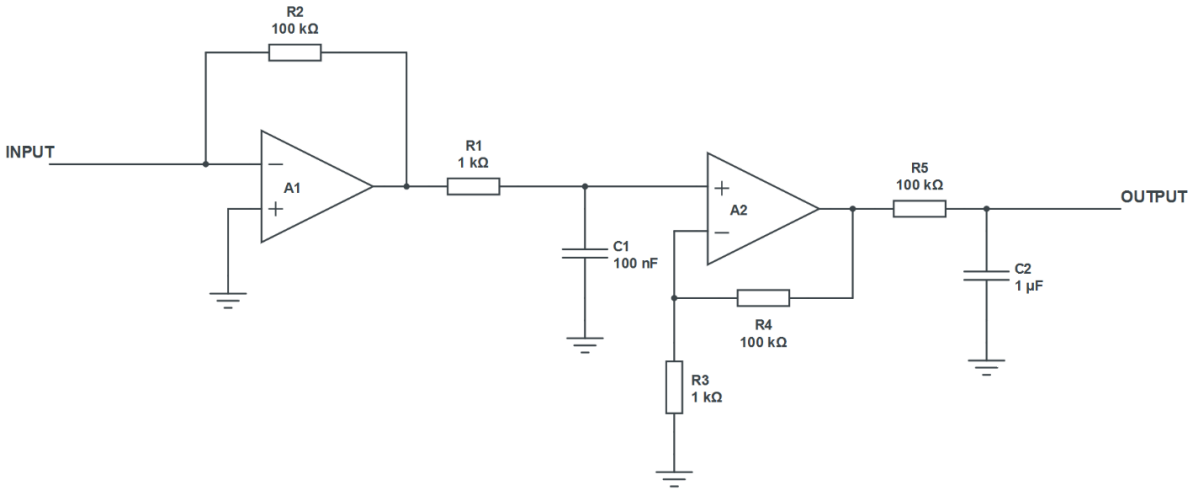
## 4.2. Sklop za detekciju promjene električnog polja

Sklop koji je spojen na dvije elektrode, te služi za prepoznavanje pojavljivanja mase odnosno uzemljenja u električnom polju zasniva se na načinu rada $C^4D$ senzora koji se razvija kako bi bio korišten za mjerenja električne provodljivosti u elektrolitskim otopinama. Metoda mjerenja električnih svojstava pomoću $C^4D$ senzora je beskontaktna, te radi na način da se na cijev izolatorskog materijala kroz koje protječe elektrolit postave dvije elektrode međusobno odvojene jedna od druge. Na prvu elektrodu se spaja izmjenični izvor napona, te ona služi kao elektroda za stvaranje pobude u sustavu. Druga elektroda kao izlaz daje struju stvorenu u tom sustavu, te se taj signal obrađuje različitim pretvorničkim sklopovima kako bi ta vrijednost mogla služiti u mjerenjima [9]. Sličan, ali manje precizan koncept korišten je i u ovom diplomskom radu, ali ne za određivanje svojstava stranog tijela koji pravi pobudu u sustavu, već samo za očitavanje njegovog položaja.

Shema $C^4D$ senzora (Sl. 4.7.) je puno kompleksnija zbog svoje preciznosti, te uz dodatno uzemljenje na drugoj elektrodi koristi četiri operacijska pojačala, dok su na shemi korištenoj za izradu sklopa za prepoznavanje položaja stranog tijela unutar sustava opisanog u ovom diplomskom radu (Sl. 4.8.) potrebna i korištena samo dva operacijska pojačala.

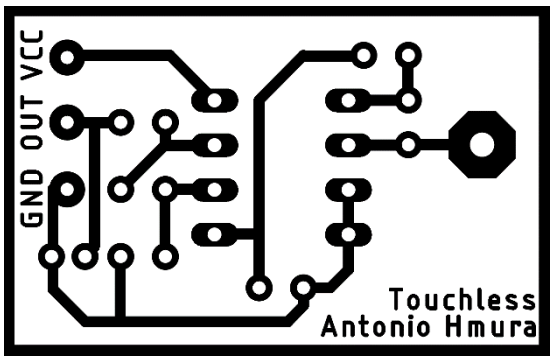



Sl. 4.7. Shema poboljšanog $C^4D$ senzora za mjerenje električne provodljivosti u elektrolitskim otopinama [9].

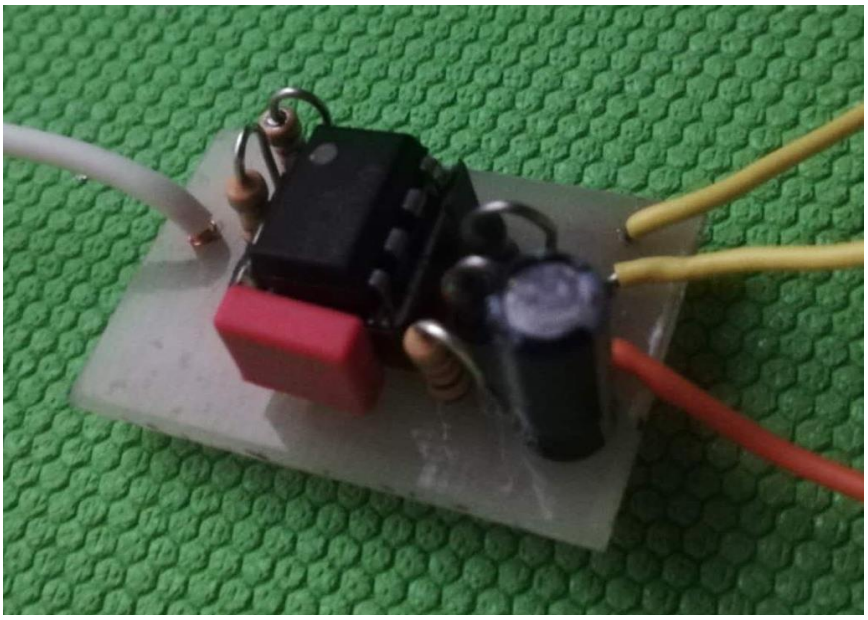



Sl. 4.8. Shema sklopa korištenog za prepoznavanje stranog tijela u uređaju za bezdodirno upravljanje.

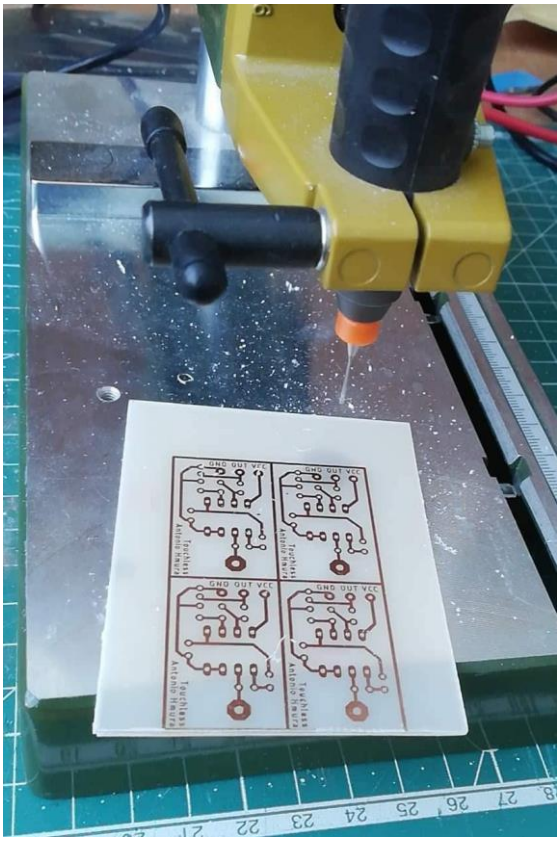
Pomoću Eagle programskog alata izmodelirane su tiskane pločice na kojima su nakon njihove izrade smještene elektroničke komponente navedenoj shemi (Sl. 4.9.). U sklopu se koristi operacijsko pojačalo LM2904P, pet otpornika snage do 1 W odgovarajućih vrijednosti, jedan elektrolitski kondenzator kapaciteta 1 µF, te jedan film kondenzator kapaciteta 100 nF. Izvod napajanja (VCC) spojen je na konstantnih 5 V istosmjernog napona, izlazni izvod sklopa (OUT) je spojen na analogni ulaz Arduino Leonardo mikrokontrolera, te izvod uzemljenja (GND) na zajedničko uzemljenje na Arduino Leonardo mikrokontroleru. Izrađena pločica dimenzija 3 cm duljine i 2 cm širine sa smještenim komponentama je prikazana na slici (Sl. 4.10.), a dio procesa same izrade pločice prikazan je na slici (Sl. 4.11.).




Sl. 4.9. Model tiskane pločice potreban za kemijsku izradu.



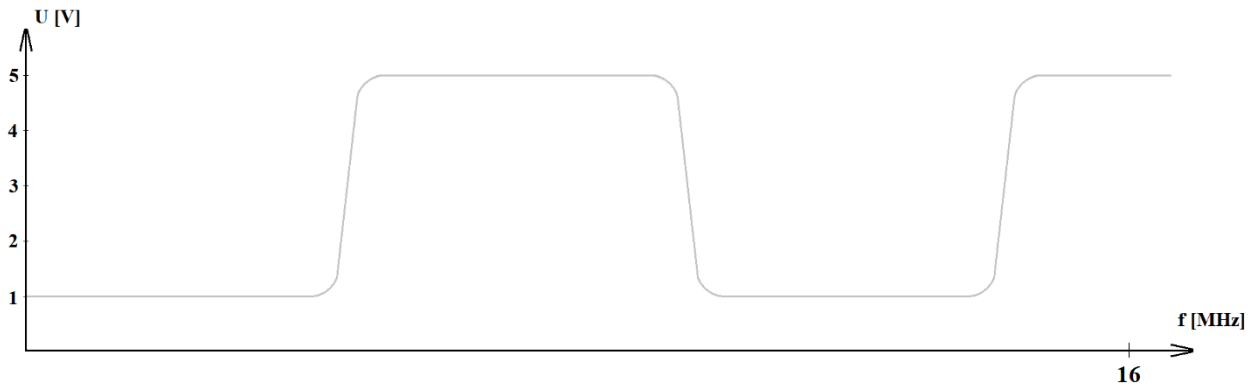
Sl. 4.10. Fotografija izrađene pločice s ugrađenim komponentama.



Sl. 4.11. Fotografija dijela procesa izrade pločice

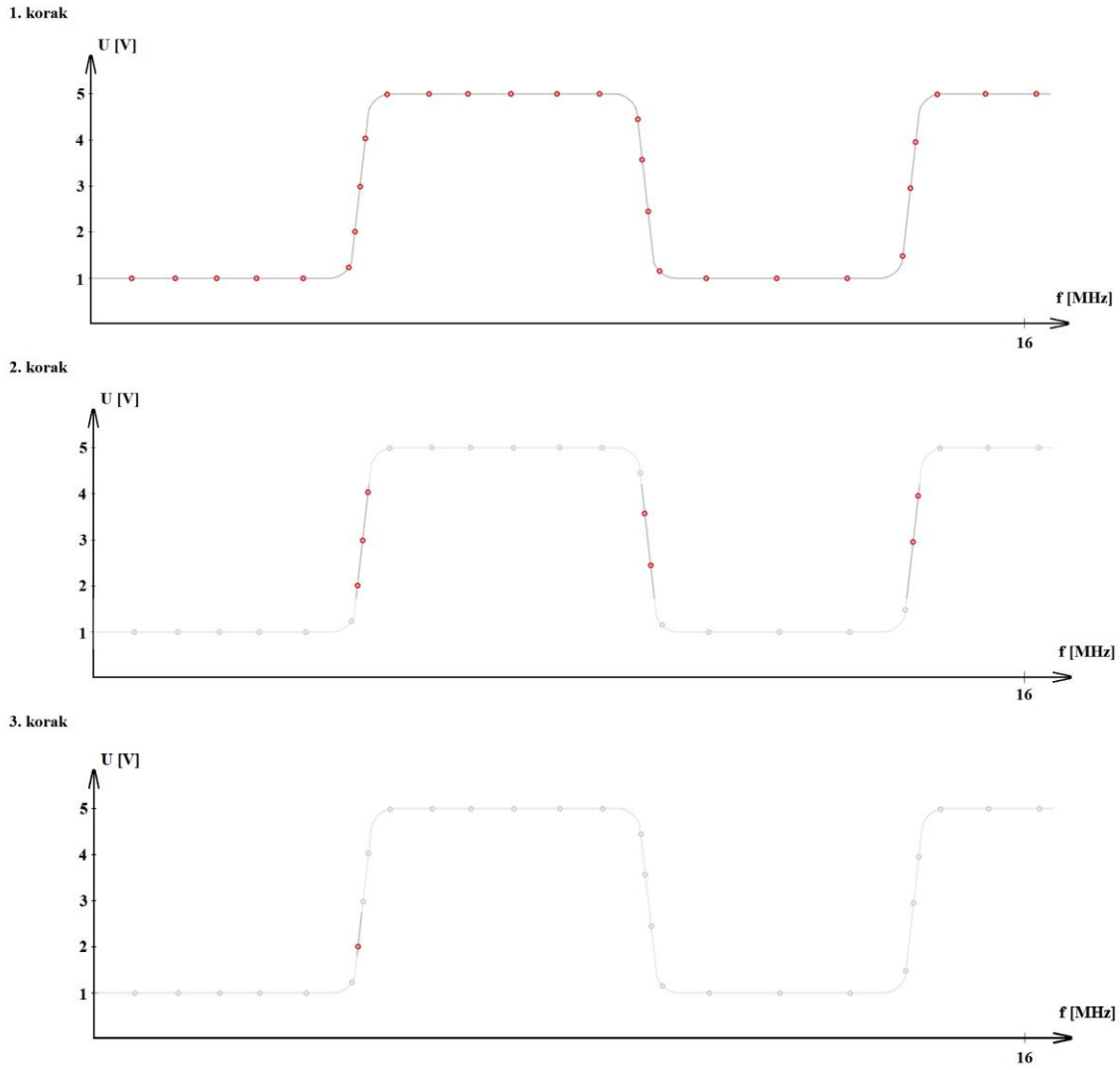
## 4.3. Rad sklopa za detekciju promjene električnog polja i utjecaj frekvencije

Za što točniju detekciju promjene električnog polja, pojedinom sklopu je potrebno pronaći optimalnu frekvenciju rada. Kako su ti sklopovi osjetljivi na svoju okolinu te na uzemljenja u njihovoj blizini, proces pronalaska optimalne frekvencije obavlja se svakim uključivanjem cijelog sustava u rad. Pri samoj početnoj inicijalizaciji cijelog sustava radna frekvencija kvarcnog kristala koji daje radni takt Arduinu se dijeli kako bi se pronašla optimalna frekvencija pojedinog sklopa. Sustav prolazi kroz petlju od 100 ponavljanja te dijeli radni takt s brojem trenutnog koraka kroz petlju. Radni takt se tako podijeli 100 puta kako bi se dobila vrijednost na kojoj će pobuda električnog polja izazvati najveću promjenu napona. Svakim korakom kroz petlju u polje se upisuje očitana vrijednost na analognom ulazu koja predstavlja napon pod frekvencijom u tom koraku. Nakon što su spremljene sve vrijednosti za pojedini sklop, potrebno je odabrati matematički odrediti najbolju frekvenciju za rad. Naime, sklop za detekciju promjene električnog polja najbolje radi pri frekvencijama na kojima je sklop u određenom obliku prijelazne pojave. Odabire se vrijednost frekvencije na samom početku te prijelazne pojave kako bi pri svakoj pobudi sklop dao najveću promjenu u odnosu na minimalnu odnosno referentnu vrijednost. Takvu je pobudu jednostavno za obrađivati i odrediti položaj mase (npr. ruke korisnika) unutar cijelog sustava. Nakon što je pronađena optimalna frekvencija rada sprema se vrijednost koraka u petlji kako bi se znalo s kojom je vrijednošću potrebno dijeliti frekvenciju pri normalnom radu sustava. Ostale vrijednosti se zanemaruju jer nisu važne niti relevantne za rad sustava. Unutar koda se nikada ne sprema vrijednost same frekvencije jer je potrebna samo vrijednost djelitelja kako bi se dobila najbolja frekvencija rada. Primjer radne karakteristika odnosno grafa ovisnosti napona o frekvenciji je prikazan na slici (Sl. 4.12.). Navedeni primjer ne predstavlja realnu radnu karakteristiku svakog sklopa zbog toga što izgled radne karakteristike ovisi o komponentama na sklopu, te njegovoj tehnologiji izrade. Kako vrijednost frekvencije nije poznata unutar koda nije ni prikazana na danom primjeru.




Sl. 4.12. Primjer idealne radne karakteristike sklopa

Postupak filtriranja frekvencija je vrlo jednostavan. Prvi korak filtriranja je da se unutar jednodimenzionalnog polja spremaju vrijednosti analognog ulaza za pojedinu frekvenciju. Prilikom puštanja pojedine frekvencije za mjerenje postavljena je vremenska odgoda od 100 ms kako bi se cijeli sustav ustalio s novom frekvencijom i dao odgovarajuću vrijednost na sklopu. U idućem koraku se stalne odnosno minimalne i maksimalne vrijednosti uklanjaju iz polja vrijednosti. Nakon toga se unutar preostalih vrijednosti pronalazi minimalna vrijednost na krivulji prijelazne pojave te se ona koristi za rad tog sklopa u svom normalnom načinu radu. Isti se postupak ponavlja i za sklop na drugoj elektrodi. Primjer određivanja najbolje frekvencije rada prikazan je po koracima na slici (Sl. 4.13.). Navedeni primjer ne predstvalja realni postupak već njegovu ilustraciju kroz nekoliko ponavljanja unutar petlje. Crvene točke prestavljaju spremljene vrijednosti unutar polja za pojedinu frekvenciju unutar petlje.




Sl. 4.13. Primjer određivanja najbolje frekvencije rada po koracima.

## 4.4. Programski kod i logika izvođenja

Svaki program pisan za Arduino platformu sastoji se od dva osnovna dijela, odnosno funkcije, *setup* (eng. funkcija postavljanja) i *loop* (eng. funkcija petlje). Funkcija postavljanja se izvodi prva i izvršava se samo jednom prije izvršavanja osnovne zadaće programa koja se izvršava unutar beskonačne petlje. Nakon postavljanja slijedi ponavljajuća funkcija u kojoj se izvršava glavni dio programa. Naravno, nužno je deklarirati i inicijalizirati potrebne globalne varijable i funkcije izvan osnovna dva dijela Arduino koda.

Funkcija postavljanja (*Setup*) se u programskom kodu za bezdodirni interaktivni uređaj može podijeliti na dvije cjeline postavljanja. Prva cjelina služi za frekvencijsku analizu te se sastoji od funkcija za određivanje najbolje frekvencije rada za pojedini ugao svake elektrode. U ovom dijelu koda se za svaku elektrodu odvija dijeljenje frekvencije, očitavanje vrijednosti pada napona na njoj, pronalazak najveće i najmanje vrijednosti pada napona, te na kraju spremanje faktora najboljeg djelitelja frekvencije za njegovu daljnju upotrebu. Nakon spremanja svih vrijednosti očitanih za pojedinu frekvenciju, uklanjaju se sve vrijednosti čiji je iznos manji ili veći za 10% od maksimalne vrijednosti, te sve vrijednosti čiji je iznos manji ili veći za 33.33% od minimalne vrijednosti. Nakon tog filtriranja pronalazi se minimalna vrijednost od onih koje su ostale, odnosno njen indeks unutar polja kako bi se pronašao djelitelj frekvencije. Dio koda za filtriranje spremljenih vrijednosti prikazan je na slici (Sl. 4.14.).

```
int getBestFreqIndex(int *voltagesArray, int Max, int Min)
{
  int N = 0;
  int tmpMin = 1024;
  voltagesArray[0] = 0;
  voltagesArray[1] = 0;
  for(i=0; i<100; i++)
  {
    if(((voltagesArray[i])<(Max + (Max/10))) && ((voltagesArray[i])>(Max - (Max/10)))) voltagesArray[i] = 0;
    if(((voltagesArray[i])<(Min + (Min/3))) && ((voltagesArray[i])>(Min - (Min/3)))) voltagesArray[i] = 0;
  }
  for(i=0; i<100; i++)
  {
    if(voltagesArray[i] > 0)
    {
      if(tmpMin > voltagesArray[i])
      {
        tmpMin = voltagesArray[i];
        N = i;
      }
    }
  }
  i = 0;
  return N;
}
```

Sl. 4.14. Funkcija za pronalazak optimalnog djelitelja frekvencije.

Druga cjelina u funkciji postavljanja je cjelina koja služi za određivanje referentnih vrijednosti dok je cijeli sustav u stanju mirovanja. Navedeni dio funkcionira tako što Arduino daje pravokutni impuls na jednu od elektroda optimalnom frekvencijom koju je pronašao u prethodnom koraku, te računa aritmetičku sredinu vrijednosti dok u sustavu nema promjene dielektrika odnosno ruke unutar kućišta u 50 koraka prolaska kroz petlju. Isti se postupak ponavlja i za drugu elektrodu kako bi se dobila referentna vrijednost. Dobivena referentna vrijednost služi za usporedbu s promijenjenom vrijednosti kada postoji pobuda unutar sustava kako bi se dobila pozicija uzemljenja unutar samog sustava.

Potpunom inicijalizacijom, koja se sastoji od frekvencijske analize i analize referentnih vrijednosti, zaključuje se postupak postavljanja sustava koji zatim inicijalizira komunikaciju računala i sustava kao ulazne jedinice tj. miša. Potrebno je napomenuti da za vrijeme postupka inicijalizacije sustav mora biti bez uzemljenja u svojoj blizini kako bi precizno odredio sve vrijednosti za normalan rad u svojoj okolini. Završavanjem funkcija postavljanja ulazi se u glavni dio koda, beskonačnu funkciju petlje koja se konstantno ponavlja.

Funkcija petlje svojim ponavljanjem omogućava vrlo čestu provjeru promjene padova napona na sklopovima elektroda. Unutar funkcije u rad se pušta jedna od elektroda frekvencijom određenom u inicijalizacijskom postupku na početku izvođenja programa. U varijable se spremaju očitane vrijednosti na sklopovima pojedine elektrode koje se uspoređuju s referentnom vrijednošću. Idući korak je pozivanje funkcije za pomicanje miša. Navedena funkcija uspoređuje trenutno očitane vrijednosti s referentnim vrijednostima iz postupka inicijalizacije te ovisno o tome kod kojih vrijednosti je došlo do povećanja, prepoznaje se položaj ruke unutar sustava te stvara vektor pomaka. Taj se vektor računa kao suma promjene vrijednosti na pojedinom uglu cijelog sustava, te ovisno o kvadrantu ugla unutar koordinatnog sustava vrijednost dobiva pozitivni odnosno negativni predznak. Funkcija koja se poziva za pomicanje miša je prikazana na slici (Sl. 4.15.). Kako sustav ne bi reagirao na male poremećaje unutar električnog polja, promjena vrijednosti mora biti barem 35% veća od referentne.

```
void moveMouse()
{
  float xValueDelta = 0;
  float yValueDelta = 0;
  if((voltageDownLeft > (1.35*defaultDownLeft)) || (voltageDownRight > (1.35*defaultDownRight)) || (voltageTopLeft > (1.35*defaultTopLeft)) || (voltageTopRight > (1.35*defaultTopRight)))
  {
    xValueDelta = (voltageDownRight + voltageTopRight - voltageDownLeft - voltageTopLeft)/2;
    yValueDelta = (voltageTopLeft + voltageTopRight - voltageDownLeft - voltageDownRight)/2;
  }
  Mouse.move(xValueDelta, yValueDelta, 0);
}
```

Sl. 4.15. Dio koda koji prikazuje funkciju za pomicanje pokazivača

Kada se postupak rada, provjere i mogućeg pomaka obavi za jedan ugao elektrode, isti se postupak ponavlja za svaki idući. Unutar ponavljajuće funkcije Arduina dolazi do naizmjeničnog paljenja jedne elektrode izvora impulsa, očitanja ugla te njenog gašenja. Zatim se impuls šalje na drugu elektrodu te se mjeri promjena vrijednosti na uglu na istoj strani kao u prethodnom koraku. Navedeni postupak se radi za obje strane kako bi se dobila promjena vrijednost na svim uglovima sustava.

Zbog preciznosti i smanjivanja kašnjenja rada cijelog sustava nije moguće koristiti gotove Arduino funkcije za stvaranje pravokutnog signala, već se direktno pristupa registrima vremenskih brojača unutar Atmega32u4 mikrokontrolera, te se njihove vrijednosti mijenjaju po potrebi za određenu frekvenciju. Na slici (Sl. 4.16.) je prikazan dio koda za stvaranje pravokutnog impulsa za vremenski brojač 1 unutar Atmega32u4 mikrokontrolera. Registri za vremenski brojač su postavljeni za rad koristeći se službenom dokumentacijom Atmega32u4 mikrokontrolera [10].

```
TCCR1A = _BV(COM1A0) | _BV(COM1B1) | _BV(WGM11) | _BV(WGM10);
TCCR1B = _BV(WGM13) | _BV(CS10);
OCR1A = bestFreqIndex;
OCR1B = 50;
delay(100);
```

Sl. 4.16. Dio koda za inicijalizaciju pravokutnog signala u vremenskom brojaču 1

# 5. FUNKCIONALNOST I ANALIZA RADA

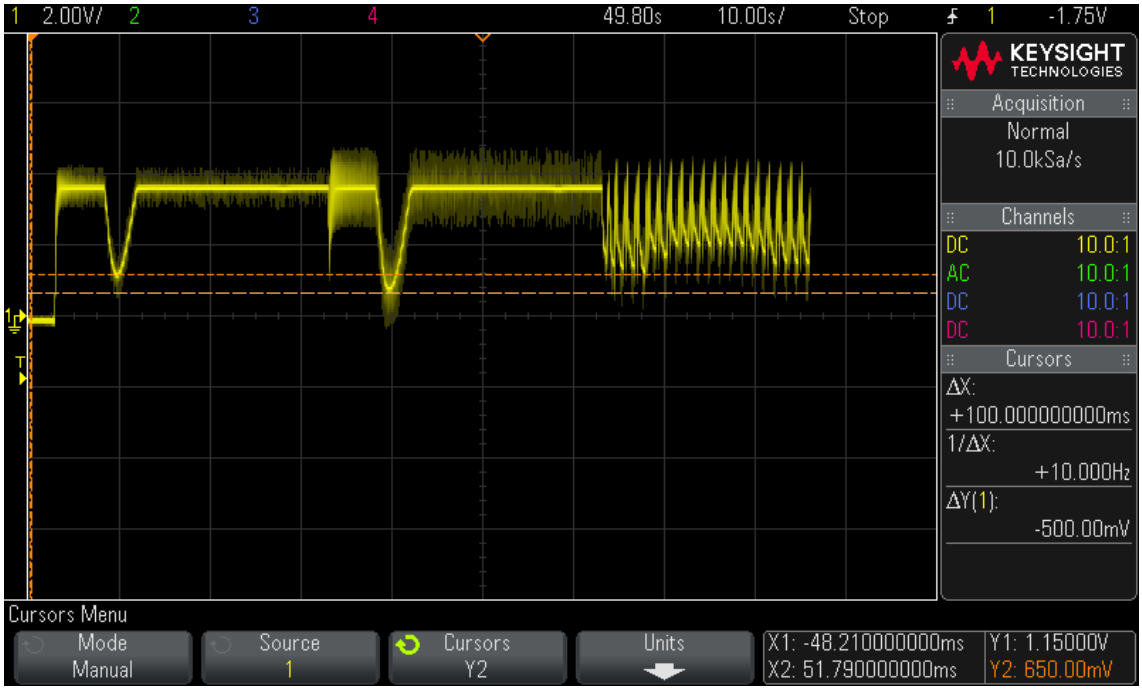
## 5.1. Funkcionalnost sustava i alternativni način razvoja

Uređaj se s računalom spaja USB sučeljem korištenjem *micro* USB kabla, te se izvođenjem koda prepoznaje kao ulazna jedinica odnosno miš. Uređaj je zamišljen da bude smješten polegnut vodoravno na nekoj radnoj površini. Nisu potrebne nikakve dodatne instalacije niti konfiguracije za normalan rad uređaja jer se pri samom uključivanju u računalo uređaj sam obavlja instalaciju, te pokretanjem koda obavlja inicijalizaciju i svoj rad. Funkcionalnost sustava se bazira na stvaranju vektora pomaka u ovisnosti o promjeni pobude na pojedinom uglu sustava. Postavljanjem ruke unutar sustava mijenja se električno polje te ovisno o položaju ruke računa se vrijednost vektor pomaka. Vrijednosti koje se računaju unutar vektora pomaka umanjene su za 90% kako bi se dobio što realniji pomak u odnosu na smještaj ruke s obzirom na pojedini ugao. Zbog svoje konstrukcije, elektroničkih komponenata i velike osjetljivosti cijelog sustava nije bilo moguće doći do najpreciznijeg rješenja problema.

Kao alternativno rješenje cijeli sustav može biti temeljen na potpuno drugačijoj tehnologiji izrade. Korištenjem laserskih ili nekih drugih svjetlosnih dioda te postavljanjem njihovih odgovarajućih receptora, uređaj se može u potpunosti razviti očitavanjem položaja ruke odnosno uzemljenja prekidanjem snopova svjetlosti unutar kućišta. Takav pristup bi zahtijevao postavljanje dioda odašiljača svjetlosti na dvije okomite strane uređaja i njihovih odgovarajućih receptora na preostale dvije okomite strane uređaja, pa bi sustav bio zasnovan na prekidu snopa svjetlosti. Tako razvijen sustav puno bi manje ovisio o utjecajima i interferencijama okoline jer nebi bio ometan drugim električnim poljima i različitim oblicima uzemljenja u blizini tog uređaja.

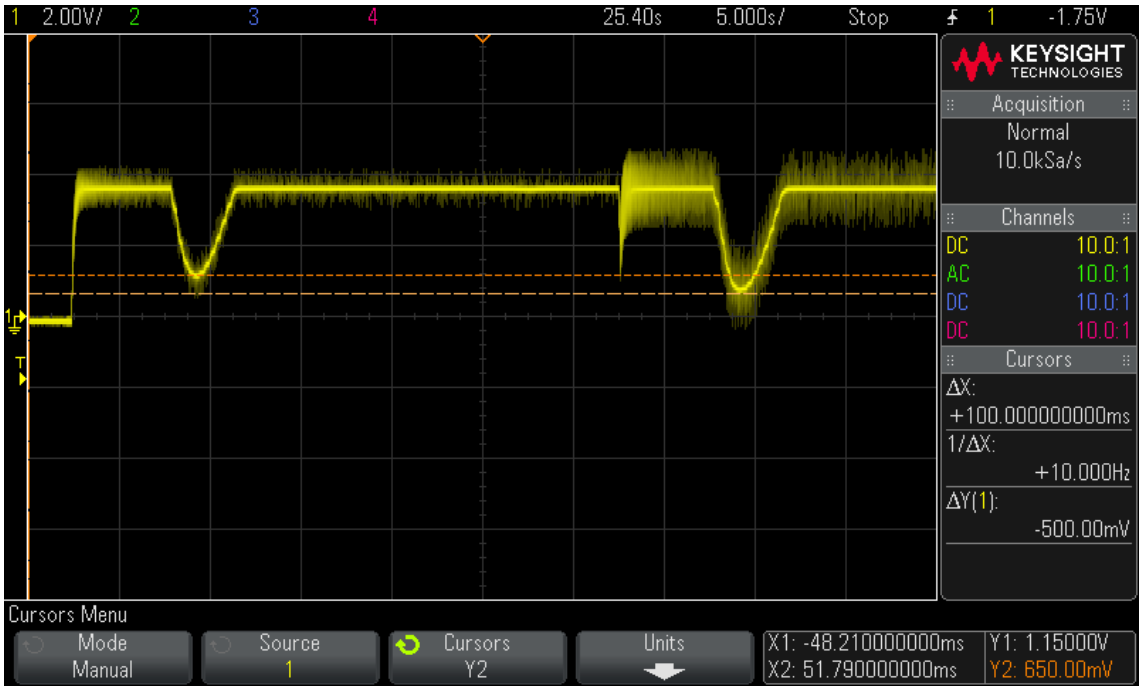
## 5.2. Analiza rada

Rad sustava, kao što je već opisano u radu, sadrži svoja dva osnovna dijela, početni dio odnosno inicijalizaciju uređaja i normalan način rada u kojemu se odvija pomicanje pokazivača miša. Pri obavljanju analize rada i testiranja sustava dobivene su radne karakteristike koje prikazuju promjene vrijednosti napona kroz vrijeme očitane osciloskopom. Priložene slike daju prikaze realnih odziva sustava kroz određeni vremenski period, te nisu idealan prikaz rada. Na slici (Sl. 5.1.) prikazana je karakteristika rada kroz obje cjeline (inicijalizacija i normalan način rada). Na slici su također naglašene različite amplitude pada napona za pojedinu elektrodu koristeći se pokazivačima vrijednosti (engl. *Cursors*).



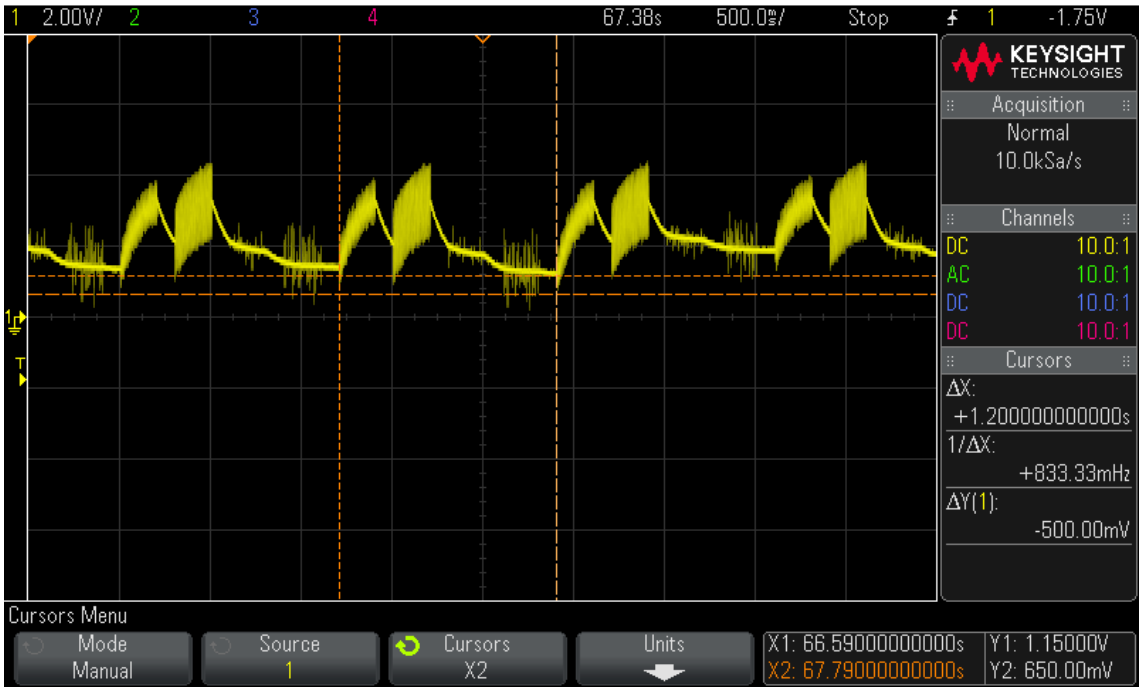

Sl. 5.1. Slika prikaza zaslona osciloskopa kroz obje cjeline rada sustava

Na slici (Sl. 5.2.) je prikazan uvećani prikaz zaslona osciloskopa samo inicijalizacijskog dijela rada sustava. Također su prikazani i pripadajući pokazivači vrijednosti za padove napona, te njihove vrijednosti.




Sl. 5.2. Slika prikaza zaslona osciloskopa samo za inicijalizaciju sustava

Na slici (Sl. 5.3.) je prikazan uvećani prikaz zaslona osciloskopa samo normalnog dijela rada sustava. U ovom slučaju važnije je prikazati vremensko trajanje jednog obilaska očitanja vrijednosti za pojedini ugao sustava koji ukupno traje 1.2 sekunde.




Sl. 5.3. Slika prikaza zaslona osciloskopa samo za normalan način rada sustava

Zbog različitih utjecaja okoline i ostalih nepoznatih faktora koji utječu na sam rad sustava i njegovo električno polje nisu dobivena očekivana mjerenja za desni sklop koji služi za očitavanje promjene električnog polja. Ostale funkcionalnosti sustava su radile u očekivanim granicama vrijednosti i promjena, ali je desna strana davala manji odziv s obzirom na rad sustava, te su se pojavili određeni šumovi u radu, kao što je prikazano na slikama. Zbog navedenih smetnji sustav je imao manje poteškoće s pomicanjem pokazivača prema dolje. Šumovi koji su se pojavili mogli su nastati zbog utjecaja okoline testiranja, te drugih čimbenika unutar prostorije, koja je u ovom slučaju bila elektronički laboratorij.

Od samog početka izrade cijelog uređaja obavljalo se testiranje pojedinog dijela sustava. Sklop za prepoznavanje promjene električnog polja je u svojoj testnoj verziji bio izrađen na bušenoj pločici sa smještenim komponentama koji se spajao na elektrode, te se njegov izlaz spajao na osciloskop. Testirane su promjene napona pri različitim frekvencijama rada sklopa, nakon potvrđene ispravnosti testnog sklopa, izrađuje se nekoliko pločica s potrebnim elektroničkim komponentama. Ponovno su testirani i navedeni izrađeni sklopovi pri različitim frekvencijama rada zbog potvrde ispravnosti kako bi ih se moglo koristiti u izradi bezdodirnog interaktivnog uređaja. Osim toga, u procesu izrade uređaja su testirane i vrijednosti promjene napona s ugrađenim elektrodama i sklopovima. Vremenska domena sustava testirana je na način da su se provjeravale odgovarajuće vrijednosti promjena frekvencija unutar dijela programskog koda za pronalazak najboljeg djelitelja frekvencije, te za vremenske brojače Arduino platforme. Testiranja promjena napona, te promjena frekvencija obavljana su u elektrotehničkom laboratoriju korištenjem osciloskopa i izvora napajanja. Sustav se u svim oblicima testiranja pokazao točan uz određene smetnje, te je s tim korakom prestao biti razvijan, te je izrađen do mjere prepoznavanja položaja ruke i pomaka s obzirom na taj položaj, s malim devijacijama u radu.

# 6. ZAKLJUČAK

Unaprijeđenjem tehnologije, zasloni osjetljivi na dodir su jako brzo i na dobar način postali svakodnevni spoj čovjeka s nekakvom vrstom tehnologije. Uz pravilan razvoj na određena mjesta uskoro se može očekivati i sve veća uporaba uređaja i zaslona koji će imati mogućnost upravljanja bez fizičkog dodira. Ovaj diplomski rad je dokaz kako uz pomoć već poznatih elektroničkih i fizikalnih pojava uz određenu primjenu znanja može doći do jednostavnog uređaja za bezdodirnu interakciju. Izrađen je uređaj koji pomoću aluminijskih elektroda i sklopova za prepoznavanje promjene električnog polja može pretvoriti ljudski pokret ruke unutar svog kućišta u pokret pokazivača miša na računalu. Kao procesor svih podataka i glavni uređaj za upravljanje rada korištena je Arduino Leonardo razvojna pločica. Promjene električnog polja se procesiraju unutar Leonardovog Atmega32u4 mikrokontrolera kao položaji ruke unutar razvijenog uređaja te se interpretiraju u računalu kao pomak pokazivača miša. Analizom rada navedenog mikrokontrolera napisan je C program koji korištenjem Arduino-ve „Mouse.h“ biblioteke funkcija uspješno pretvara ljudsku interakciju s uređajem u pomak pokazivača miša. Nakon potpune izrade uređaja i testiranja sustava u nekoliko različitih okolina (kao što su soba, ured i elektronički laboratorij) može se doći do zaključka kako je sustav uspješno razvijen uz manje smetnje u radu. Sustav je razvijan kao dokaz koncepta bezdodirnog interaktivnog uređaja, te se može nastaviti razvijati dodavanjem sklopova za prepoznavanje promjene električnog polja, dodavanjem ili promjenom elektroda, izradom pomoću CNC stroja ili nekom drugom preciznijom tehnikom, promjenom temelje tehnologije koncepta (korištenjem prekida snopova svjetlosti, ne pobudom električnog polja) itd.

# LITERATURA


[1] Antonio Hmura, završni rad na temu „Dizaj elektroda za kapacitivni senzor položaja“, https://urn.nsk.hr/urn:nbn:hr:200:016299, 2017., pristup ostvaren 11. kolovoza 2019.

[2] Nekoliko urednika enciklopedije Britannica na temu kapaciteta, https://www.britannica.com/science/capacitance, 3. rujan 2018., pristup ostvaren 24. lipnja 2019.

[3] Nekoliko urednika enciklopedije Britannica na temu kondenzatora, https://www.britannica.com/technology/capacitor, 15. studeni 2018., pristup ostvaren 24. lipnja 2019.

[4] Woodford, Chris – Touchscreens, https://www.explainthatstuff.com/touchscreens.html, 30. kolovoz 2018., pristup ostvaren 24. lipnja 2019.

[5] Urednici *„Cammax“* internetske stranice s opisom vrsta zaslona na dodir, https://www.cammaxlimited.co.uk/news/general/what-are-the-different-types-of-touchscreens/, 2018, pristup ostvaren 24. lipnja 2019.

[6] N.B. Navale College of Engineering, Solapur, Maharashtra, India, *„Study of touch less touch screen technology“* autora Nilofar E. Chanda, http://troindia.in/journal/ijcesr/vol4iss11part4/48-52.pdf, 2017., pristup ostvaren 24. lipnja 2019.

[7] Službena stranica s opisom korištenog Arduino Leonardo razvojnog sustava, https://store.arduino.cc/leonardo, 2019, pristup ostvaren 24. lipnja 2019.

[8] Službena dokumentacija „Mouse.h“ bilbioteke za rad s mišem na Arduino sučelju, https://www.arduino.cc/reference/en/language/functions/usb/mouse

[9] Znanstveni članak *„New C4D Sensor with a Simulated Inductor“* autora Yingchao Lyu, Haifeng Ji, Shijie Yang, Zhiyao Huang, Baoliang Wang and Haiqing Li, https://pdfs.semanticscholar.org./77ed/39b1cf26c90a5caca3c518f7f72a1654d73c.pdf, objavljen 27. siječnja 2016, pristu ostvaren 24. lipnja 2019.

[10] Službena dokumentacija (eng. *„datasheet“*). za Atmega32u4 mikrokontroler, http://ww1.microchip.com/downloads/en/devicedoc/atmel-7766-8-bit-avr-atmega16u4-32u4_datasheet.pdf, 2016., pristup ostvaren 6. kolovoza 2019.

# SAŽETAK

Cilj ovog diplomskog rada je razviti i izraditi uređaj koji položaj ljudske ruke unutar svoga kućišta predstavlja računalu kao pomak pokazivača u lijevu ili desnu stranu. Sustav se temelji na promjene električnog polja između četiri elektrode smještene unutar kućišta uređaja. Za obradu i slanje svih podataka i instrukcija u sustavu korištena je Arduino Leonardo platforma s ATmega32u4 mikrokontrolerom koja se povezana s računalom može ponašati kao ulazni uređaj odnosno miš. Korišten je C++ programski jezik za razvoj funkcionalnosti uređaja. Uz testiranja tijekom same izrade i testiranja gotovog uređaja donosi se odgovaraju zaključak o izrađenom bezdodirnom interaktivnom uređaju.

**Ključne riječi**: *električno polje, bezdodirno, Arduino, Leonardo, ATmega32u4,*

## ABSTRACT

The goal of this master thesis is to develop and make a device which moves the mouse cursor left or right in regards to position of the human hand inside its case. The system is based on the change of the electric field between four electrodes placed in the case of the device. Arduino Leonardo, an ATmega32u4 microprocessor based platform is used for processing and sending all data and instructions in the system which can act as an input device (mouse) when connected to a computer. Programming language used for developing device functionality is C++. By testing the finished device and running tests while it was in development process an appropriate conclusion is reached about the touchless interactive device.

**Keywords**: *electric field, touchless, Arduino, Leonardo, ATmega32u4*

# ŽIVOTOPIS

Antonio Hmura je rođen 31. siječnja 1996. u Osijeku. Nakon završene osnovne škole „Mladost“ u Osijeku 2010. godine upisuje Elektrotehničku i prometnu školu Osijek, smjer tehničar za računalsvo. Unutar svog srednjoškolskog obrazovanja svoja teorijska znanja nadopunjuje sudjelovanjem u natjecanjima iz osnova informatike na kojem 2013. godine ostvaruje prvo mjesto na županijskom natjecanju. Obavljanjem stručne prakse u HEP TE-TO Osijek i RTV servisu Madecki proširuje svoje praktične vještine. 2014. godine direktnim upisom započinje studij računarstva na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku. Od 2015./2016. do 2017./2018. akademske godine je demonstrator kolegija Elektronika I na smjerovima elektrotehnika i računarstvo. 2017. godine postaje sveučilišni prvostupnik inženjer računarstva. Također, nastavlja nadograđivati svoja znanja obavljanjem prakse u Ericsson NT Osijek na posljednjoj godini diplomskog studija.

Potpis:

# PRILOZI

## 1) Programski kod razvijenog sustava na Arduino sučelju

```
#include "Mouse.h"

int i = 0;

int sweepVoltagesDownLeft[100];
int sweepVoltagesDownRight[100];
int sweepVoltagesTopLeft[100];
int sweepVoltagesTopRight[100];

int voltageDownLeft = analogRead(A1);
int voltageDownRight = analogRead(A0);
int voltageTopLeft = analogRead(A1);
int voltageTopRight = analogRead(A0);

int maxDownLeft = 0, minDownLeft = 0;
int maxDownRight = 0, minDownRight = 0;
int maxTopLeft = 0, minTopLeft = 0;
int maxTopRight = 0, minTopRight = 0;

int bestFreqIndexDownLeft = 0;
int bestFreqIndexDownRight = 0;
int bestFreqIndexTopLeft = 0;
int bestFreqIndexTopRight = 0;

int defaultDownLeft = 0;
int defaultDownRight = 0;
int defaultTopLeft = 0;
int defaultTopRight = 0;

void sweepTroughDown();
void sweepTroughTop();

float getDefaultDownValue(int BestFreqIndex, char side);
float getDefaultTopValue(int BestFreqIndex, char side);

void moveMouse();

int getMinVoltage(int *voltagesArray);
int getMaxVoltage(int *voltagesArray);
int getBestFreqIndex(int *voltagesArray, int Max, int Min);

void setup() {
  Serial.begin(9600);
  pinMode(9, OUTPUT);
  pinMode(5, OUTPUT);

  Serial.println("START");
  sweepTroughDown();
  sweepTroughTop();
  Serial.println("SWEEP DONE");

  maxDownLeft = getMaxVoltage(sweepVoltagesDownLeft);
  minDownLeft = getMinVoltage(sweepVoltagesDownLeft);
  maxDownRight = getMaxVoltage(sweepVoltagesDownRight);
  minDownRight = getMinVoltage(sweepVoltagesDownRight);
  maxTopLeft = getMaxVoltage(sweepVoltagesTopLeft);
  minTopLeft = getMinVoltage(sweepVoltagesTopLeft);
  maxTopRight = getMaxVoltage(sweepVoltagesTopRight);
  minTopRight = getMinVoltage(sweepVoltagesTopRight);
```

```
60.
61.   bestFreqIndexDownLeft = getBestFreqIndex(sweepVoltagesDownLeft, maxDownLeft, minDownL
    eft);
62.   Serial.println(bestFreqIndexDownLeft);
63.   bestFreqIndexDownRight = getBestFreqIndex(sweepVoltagesDownRight, maxDownRight, minDo
    wnRight);
64.   Serial.println(bestFreqIndexDownRight);
65.   bestFreqIndexTopLeft = getBestFreqIndex(sweepVoltagesTopLeft, maxTopLeft, minTopLeft)
    ;
66.   Serial.println(bestFreqIndexTopLeft);
67.   bestFreqIndexTopRight = getBestFreqIndex(sweepVoltagesTopRight, maxTopRight, minTopRi
    ght);
68.   Serial.println(bestFreqIndexTopRight);
69.
70.   defaultDownLeft = getDefaultDownValue(bestFreqIndexDownLeft, 'L');
71.   Serial.println(defaultDownLeft);
72.   defaultDownRight = getDefaultDownValue(bestFreqIndexDownRight, 'R');
73.   Serial.println(defaultDownRight);
74.   defaultTopLeft = getDefaultTopValue(bestFreqIndexTopLeft, 'L');
75.   Serial.println(defaultTopLeft);
76.   defaultTopRight = getDefaultTopValue(bestFreqIndexDownRight, 'R');
77.   Serial.println(defaultTopRight);
78.
79.   Mouse.begin();
80. }
81.
82. void loop() {
83.
84.   TCCR1A = _BV(COM1A0) | _BV(COM1B1) | _BV(WGM11) | _BV(WGM10);
85.   TCCR1B = _BV(WGM13) | _BV(CS10);
86.   OCR1A = bestFreqIndexDownLeft;
87.   OCR1B = 50;
88.   delay(200);
89.   voltageDownLeft = analogRead(A1);
90.   Serial.print("DL ");
91.   Serial.println(voltageDownLeft);
92.   TCCR1B = 0b00000000;
93.   delay(100);
94.
95.   TCCR3A = _BV(COM3A0) | _BV(COM3B1) | _BV(WGM31) | _BV(WGM30);
96.   TCCR3B = _BV(WGM33) | _BV(CS30);
97.   OCR3A = bestFreqIndexTopLeft;
98.   OCR3B = 50;
99.   delay(200);
100.          voltageTopLeft = analogRead(A1);
101.          Serial.print("TL ");
102.          Serial.println(voltageTopLeft);
103.          TCCR3B = 0b00000000;
104.          delay(100);
105.
106.          TCCR1A = _BV(COM1A0) | _BV(COM1B1) | _BV(WGM11) | _BV(WGM10);
107.          TCCR1B = _BV(WGM13) | _BV(CS10);
108.          OCR1A = bestFreqIndexDownRight;
109.          OCR1B = 50;
110.          delay(200);
111.          voltageDownRight = analogRead(A0);
112.          Serial.print("DR ");
113.          Serial.println(voltageDownRight);
114.          TCCR1B = 0b00000000;
115.          delay(100);
116.
117.          TCCR3A = _BV(COM3A0) | _BV(COM3B1) | _BV(WGM31) | _BV(WGM30);
118.          TCCR3B = _BV(WGM33) | _BV(CS30);
119.          OCR3A = bestFreqIndexTopRight;
120.          OCR3B = 50;
121.          delay(200);
```

```
122.         voltageTopRight = analogRead(A0);
123.         Serial.print("TR ");
124.         Serial.println(voltageTopRight);
125.         TCCR3B = 0b00000000;
126.         delay(100);
127.
128.         moveMouse();
129.       }
130.
131.
132.
133.       void sweepTroughDown()
134.       {
135.         for(i=0; i<100; i++)
136.         {
137.           TCCR1A = _BV(COM1A0) | _BV(COM1B1) | _BV(WGM11) | _BV(WGM10);
138.           TCCR1B = _BV(WGM13) | _BV(CS10);
139.           OCR1A = i;
140.           OCR1B = 50;
141.           delay(300);
142.           sweepVoltagesDownLeft[i] = analogRead(A1);
143.           sweepVoltagesDownRight[i] = analogRead(A0);
144.         }
145.         i = 0;
146.         TCCR1B = 0b00000000;
147.         delay(100);
148.       }
149.
150.       void sweepTroughTop()
151.       {
152.         for(i=0; i<100; i++)
153.         {
154.           TCCR3A = _BV(COM3A0) | _BV(COM3B1) | _BV(WGM31) | _BV(WGM30);
155.           TCCR3B = _BV(WGM33) | _BV(CS30);
156.           OCR3A = i;
157.           OCR3B = 50;
158.           delay(300);
159.           sweepVoltagesTopLeft[i] = analogRead(A1);
160.           sweepVoltagesTopRight[i] = analogRead(A0);
161.         }
162.         i = 0;
163.         TCCR3B = 0b00000000;
164.         delay(100);
165.       }
166.
167.       int getMinVoltage(int *voltagesArray)
168.       {
169.         int tmpMin = 1024;
170.         for(i=2; i<100; i++)
171.         {
172.           if(tmpMin > voltagesArray[i]) tmpMin = voltagesArray[i];
173.         }
174.         i = 0;
175.         return tmpMin;
176.       }
177.
178.       int getMaxVoltage(int *voltagesArray)
179.       {
180.         int tmpMax = 0;
181.         for(i=2; i<100; i++)
182.         {
183.           if(tmpMax < voltagesArray[i]) tmpMax = voltagesArray[i];
184.         }
185.         i = 0;
186.         return tmpMax;
187.       }
```

```
188.
189.        float getDefaultDownValue(int BestFreqIndex, char side)
190.        {
191.          float defaultValue = 0;
192.          TCCR1A = _BV(COM1A0) | _BV(COM1B1) | _BV(WGM11) | _BV(WGM10);
193.          TCCR1B = _BV(WGM13) | _BV(CS10);
194.          OCR1A = BestFreqIndex;
195.          OCR1B = 50;
196.          delay(300);
197.          if(side == 'L')
198.          {
199.            for(i=0; i<50; ++i)
200.            {
201.              defaultValue += analogRead(A1);
202.              if (i == 49)
203.              {
204.                i = 0;
205.                TCCR1B = 0b00000000;
206.                delay(100);
207.                return (defaultValue/50.0);
208.              }
209.            }
210.          }
211.          else if(side == 'R')
212.          {
213.            for(i=0; i<50; ++i)
214.            {
215.              defaultValue += analogRead(A0);
216.              if (i == 49)
217.              {
218.                i = 0;
219.                TCCR1B = 0b00000000;
220.                delay(100);
221.                return (defaultValue/50.0);
222.              }
223.            }
224.          }
225.        }
226.
227.        float getDefaultTopValue(int BestFreqIndex, char side)
228.        {
229.          float defaultValue = 0;
230.          TCCR3A = _BV(COM3A0) | _BV(COM3B1) | _BV(WGM31) | _BV(WGM30);
231.          TCCR3B = _BV(WGM33) | _BV(CS30);
232.          OCR3A = BestFreqIndex;
233.          OCR3B = 50;
234.          delay(300);
235.          if(side == 'L')
236.          {
237.            for(i=0; i<50; ++i)
238.            {
239.              defaultValue += analogRead(A1);
240.              if (i == 49)
241.              {
242.                i = 0;
243.                TCCR3B = 0b00000000;
244.                delay(100);
245.                return (defaultValue/50.0);
246.              }
247.            }
248.          }
249.          else if(side == 'R')
250.          {
251.            for(i=0; i<50; ++i)
252.            {
253.              defaultValue += analogRead(A0);
```

```
254.            if (i == 49)
255.            {
256.              i = 0;
257.              TCCR3B = 0b00000000;
258.              delay(100);
259.              return (defaultValue/50.0);
260.            }
261.          }
262.        }
263.      }
264.
265.      int getBestFreqIndex(int *voltagesArray, int Max, int Min)
266.      {
267.        int N = 0;
268.        int tmpMin = 1024;
269.        voltagesArray[0] = 0;
270.        voltagesArray[1] = 0;
271.        for(i=0; i<100; i++)
272.        {
273.          if(((voltagesArray[i])<(Max + (Max/10))) && ((voltagesArray[i])>(Max - (Max/
    10)))) voltagesArray[i] = 0;
274.          if(((voltagesArray[i])<(Min + (Min/3))) && ((voltagesArray[i])>(Min - (Min/3
    )))) voltagesArray[i] = 0;
275.        }
276.        for(i=0; i<100; i++)
277.        {
278.          if(voltagesArray[i] > 0)
279.          {
280.            if(tmpMin > voltagesArray[i])
281.            {
282.              tmpMin = voltagesArray[i];
283.              N = i;
284.            }
285.          }
286.        }
287.        i = 0;
288.        return N;
289.      }
290.
291.      void moveMouse()
292.      {
293.        float xValueDelta = 0;
294.        float yValueDelta = 0;
295.        Serial.println(" MOVE ");
296.        if((voltageDownLeft > (1.35*defaultDownLeft)) || (voltageDownRight > (1.35*def
    aultDownRight)) || (voltageTopLeft > (1.35*defaultTopLeft)) || (voltageTopRight > (1.35
    *defaultTopRight)))
297.        {
298.          xValueDelta = (voltageDownRight + voltageTopRight - voltageDownLeft - voltag
    eTopLeft)/2;
299.          yValueDelta = (voltageTopLeft + voltageTopRight - voltageDownLeft - voltageD
    ownRight)/2;
300.        }
301.        Serial.println(xValueDelta);
302.        Serial.println(yValueDelta);
303.        Mouse.move(xValueDelta, yValueDelta, 0);
304.      }
305.
```