

Mikroupravljački sustav za prepoznavanje položaja ruku

Tomić, Matej

Undergraduate thesis / Završni rad

2016

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:200:031646>

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

Download date / Datum preuzimanja: **2024-12-23**

Repository / Repozitorij:

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



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

Preddiplomski studij računarstva

**MIKROUPRAVLJAČKU SUSTAV ZA
PREPOZNAVANJE POLOŽAJA RUKU**

Završni rad

Matej Tomić

Osijek, 2016.godina

SADRŽAJ

| | |
|---|-----------|
| 1. UVOD..... | 1 |
| 1.1. Zadatak završnog rada..... | 1 |
| 2. ALATI KORIŠTENI PRILIKOM IZRADE | 2 |
| 2.1. Alati potrebni za izradu sklopa za izradu fotografije..... | 2 |
| 2.2. Alati potrebni za obradu fotografije | 4 |
| 3. REALIZACIJA SUSTAVA..... | 5 |
| 3.1. Izrada fotografije | 5 |
| 3.2. Obrada fotografije..... | 7 |
| 3.3. Upravljanje uređajem na temelju fotografije..... | 11 |
| 4. ZAKLJUČAK..... | 14 |
| LITERATURA | 15 |
| SAŽETAK | 16 |
| ABSTRACT..... | 17 |
| ŽIVOTOPIS | 18 |
| PRILOG A. Program za Arduino Uno razvojni sustav | 19 |
| PRILOG B. Program za MATLAB razvojni sustav – funkcija snapshot | 19 |
| PRILOG C. Program za MATLAB razvojni sustav – funkcija fingercount | 20 |

1. UVOD

Današnja tehnologija se razvija vrlo velikom brzinom i gotovo je nemoguće ostati u koraku s njenim napretkom. Sve je više primjera u kojima se standardan unos pomoću tipkovnice ili drugih fizičkih uređaja pokušava zamijeniti glasovnim upravljanjem i drugim sličnim metodama. S ciljem detaljnijeg upoznavanja za temu rada odabrana je jedna od tih metoda, točnije „Mikroupravljački sustav za prepoznavanje položaja ruku“. Cilj ovog rada je izraditi sustav koji će na temelju položaja ruku upravljati određenim uređajem. Sustav se može podijeliti na tri dijela: dio za izradu slike, obrada napravljene slike i upravljanje uređajem na temelju slike, odnosno položaja ruke. Pomoću kamere će se zabilježiti određeni položaj ruke. Zatim će se zabilježena fotografija obraditi na način da ju sustav svaki idući put prepozna i napravi određenu akciju, u ovom slučaju upravlja s uređajem.

1.1.Zadatak završnog rada

U ovom završnom radu potrebno je napraviti mikroupravljački sustav s kamerom za prepoznavanje položaja ruku. Potrebno je prepoznati određene položaje tijela te na temelju slike identificirane položaje ruku potrebno je koristiti za daljinski upravljanje uređaje, npr. paljenje gašenje svjetla.

2. ALATI KORIŠTENI PRILIKOM IZRADE

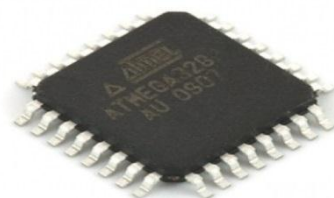
2.1. Alati potrebni za izradu sklopa za izradu fotografije

Prije početka izrade samog sklopa potrebno je odabrati modul pomoću kojeg će se sklop ostvariti. Detaljnijom analizom cijene i kvalitete, ali i jednostavnosti korištenja odabrana je Arduino platforma. Arduino je open-source platforma temeljena na hardware-u i software-u koji su jednostavni za korištenje. Za izradu sklopa potrebna je uporaba Arduino modula, Arduino programskog jezika (Arduino podržava C i C++) i Arduino software-a. Za izradu ovog rada odabran je ArduinoUNO modul prikazan na slici Sl.2.1..



Slika 2.1. Prikaz ArduinoUNO modula

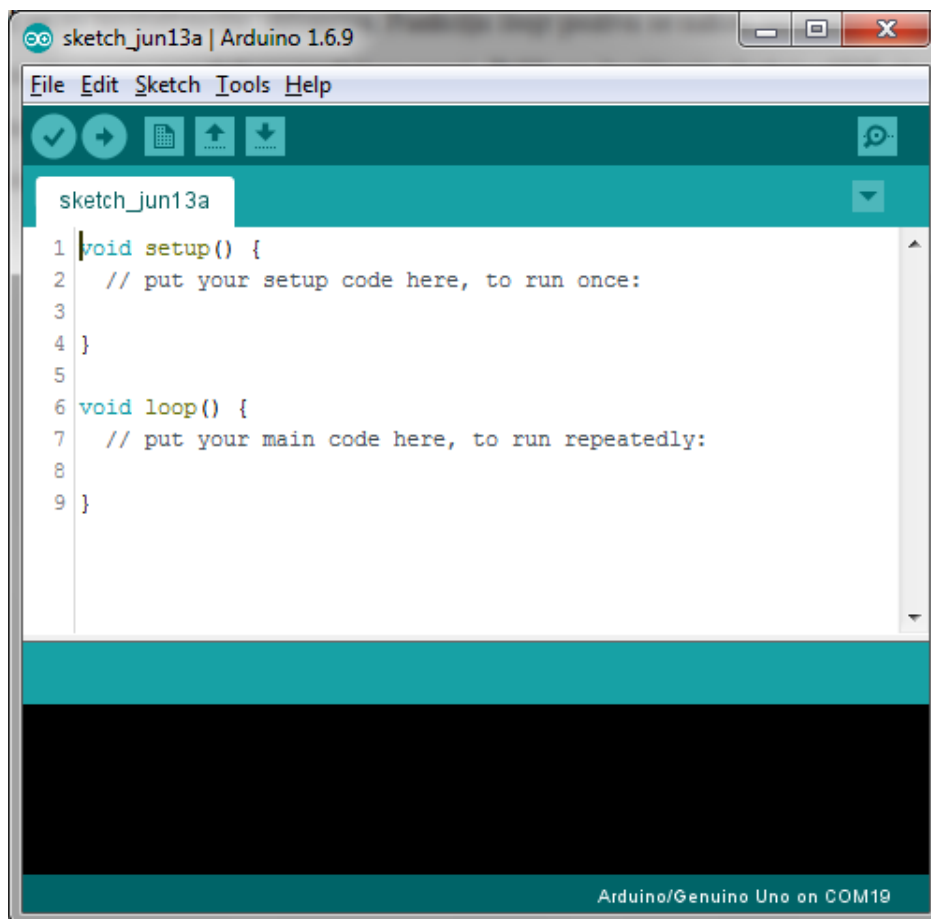
ArduinoUNO modul temelji se na Atmel 8-bitnom mikrokontroleru skrćenog imena ATmega328P prikazanog na slici Sl.2.2..



Slika 2.2. Prikaz Atmega328P mikrokontrolera

Sastoji se od 14 digitalnih ulazno/izlazni pinova, 6 analognih ulaza, USB (eng. universal serial bus) priključka kao i priključka za napajanje putem adaptera, ICMP (eng. internet control message protocol) zaglavlja i tipke za reset. Spajanjem ArduinoUNO modula i računala putem USB kabela ili adaptera omogućeno je korištenje i programiranje istoga. Više informacija o Arduino i ArduinoUNO modulu te ATmega328P mikrokontroleru nalazi se na [1], odnosno [2].

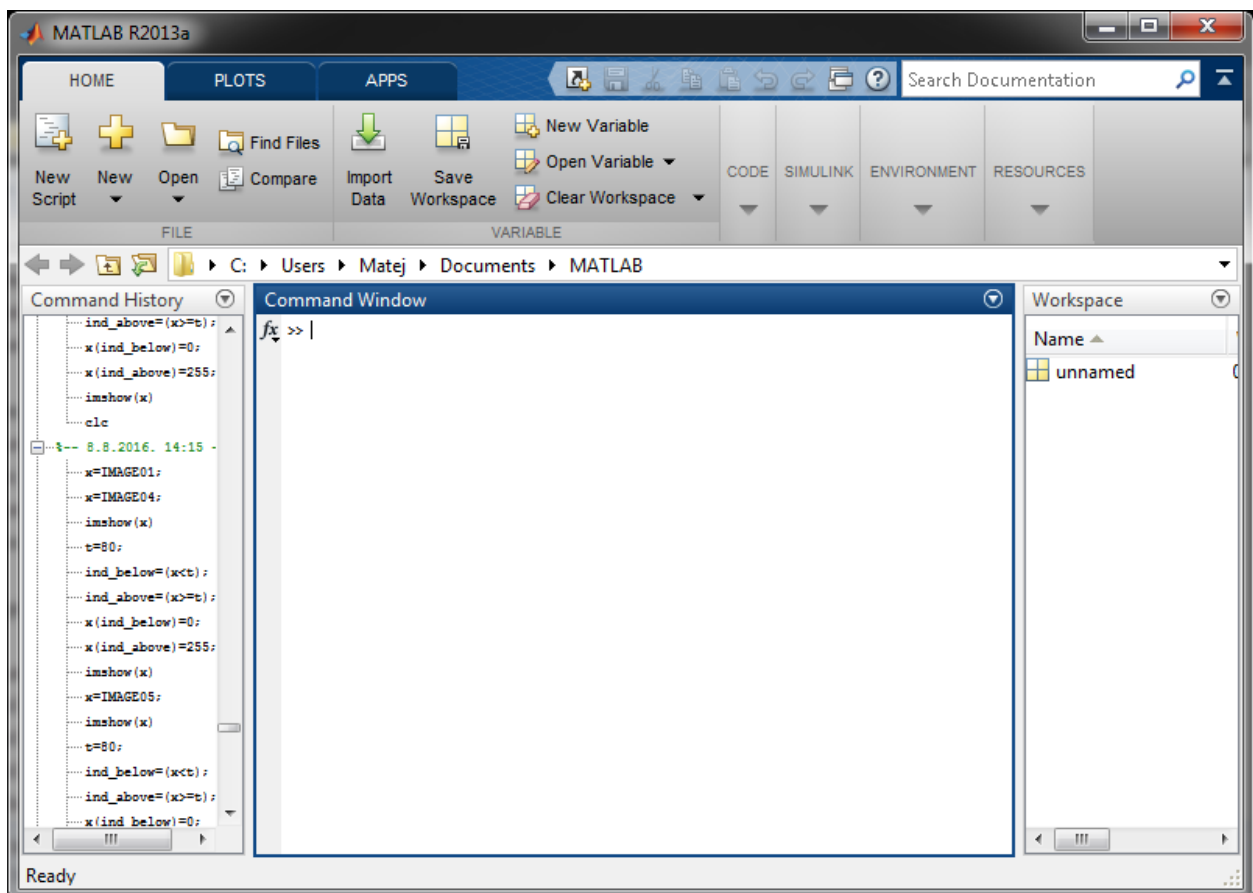
Za programiranje modula odabrano je Arduino IDE 1.6.9 (eng. integrated development environment) razvojno okruženje (slika S1.2.3.). Kompletan program napisan u Arduino IDE okruženju naziva se sketch. Arduino sketch program sastoji se od dva glavna dijela: loop i setup funkcije. Setup funkcija se izvršava samo jedan puta, prilikom prvog pokretanja i koristi se za inicijalizaciju sklopova. Funkcija loop poziva se nakon inicijalizacije i to beskonačno puta, sve dok se modul ne ugasi. Prilikom korištenja ArduinoIDE okruženja vrlo je bitno da je odabran točan modul koji se koristi i točan port preko kojega modul komunicira s računalom.



Slika 2.3. Prikaz ArduinoIDE 1.6.9 razvojnog okruženja

2.2. Alati potrebni za obradu fotografije

Nakon odabira modula za izradu fotografije potrebno je odabrati i alat pomoću kojeg će se ta fotografija obraditi. S obzirom na korištenje Arduino platforme odabrano je MATLAB (eng. Matrix Laboratory) razvojno okruženje. MATLAB razvojno okruženje pruža velike mogućnosti prilikom projektiranja sustava, simulacije i vizualizacije rada sustava, rješavanje matematičkih problema, prikaz grafova i funkcija, obrade slike i drugo. Podržava rad na Windows, Linux i OS X operacijskim sustavima te je jednostavan za korištenje. Dva osnovna dijela sučelja su: Command window i Workspace. Command window je dio koji služi za upisivanje naredbi koje će se kasnije izvršiti, dok se u Workspace dijelu nalaze varijable i njihove vrijednosti. Izgled MATLAB razvojnog okruženja može se vidjeti na slici Sl.2.4., a više informacija o MATLAB-u na [3].



Slika 2.4. Prikaz MATLAB razvojnog okruženja

3. REALIZACIJA SUSTAVA

Sustav je, kako je u uvodu već navedeno, podijeljen na tri dijela. Prvi dio je izrada fotografije, drugi njena obrada i treći upravljanje uređajem na temelju fotografije. Kompletno rješenje trebalo bi biti u Arduino razvojnom sustavu. Slika koja bi se izradila nekom od kamera bila bi pohranjena u *.jpeg* formatu i zatim prebaciti u *.bmp* format radi njene obrade. Međutim, kako je zbog pristupačnosti i cijene odabran ArduinoUNO modul to nije moguće. Općenito, Arduino temeljen na AVR mikrokontroleru ima ograničen prostor za pohranu podataka te nema dovoljno snage za računanje kompleksnih operacija. Većina Arduino modula ima vrlo malo SRAM (eng. Static Random Access Memory) memorije u koju bi se slika pohranila. Naime, da bi slika bila upotrebljiva za daljnu obradu potrebne je veličina 320x240. ArduinoUNO [4] ima samo 2KB(eng. Kilobyte) SRAM memorije i slika veličine samo 64x32 zauzela bi svu tu memoriju. Arduino Mega [5] ima 8KB SRAM memorije i slika veličine 128x64 zauzela bi također svu memoriju. Čak niti Arduino Zero sa 32KB SRAM memorije nije dovoljan. Jedino rješenje bi bili moduli poput *Raspberry Pi* i *BeagleBoard* čija SRAM memorija seže od nekoliko MB (eng. Megabyte) pa sve do GB (eng. Gigabyte), ali njihova cijena je mnogo veća. Prikaz odnosa memorije nalazi se u tablici Tab. 3.1.. Stoga je kompletno rješenje sustava podijeljeno na programski dio u MATLAB razvojnom okruženju i programski dio u Arduino razvojnom okruženju. MATLAB razvojno okruženje upotrijebljeno je za izradu i obradu slike te slanje instrukcije na ArduinoUNO modul. Arduino razvojno okruženje upotrijebljeno je za upravljanje i izvršavanje instrukcija poslanih putem MATLAB-a.

| Naziv modula | SRAM memorija |
|------------------|---------------|
| ArduinoUNO | 2KB |
| Arduino Mega | 8KB |
| Arduino Zero | 32KB |
| Raspberry Pi [6] | 128MB – 1GB |
| BeagleBoard [7] | 512MB – 2GB |

Tablica 3.1. Prikaz SRAM memorije

3.1. Izrada fotografije

Nakon što je, uz navedene probleme, odlučeno kojim će se alatima služiti, kreće se u realizaciju sustava. Programski dio za izradu i obradu sastoji se od dvije funkcije, *snapshot* i *fingercount*. Funkcija *snapshot* služi za izradu slike i unutar nje se nakon izrade slike poziva funkcija za obradu - *fingercount*. Za izradu slike odabrana je web kamera koja je ugrađena na Toshiba

prijenosnom računalu. Odabrana kamera pruža mogućnost vrlo visoke kvalitete slike kao i mogućnost odabira različitih veličina slike potrebne za daljnji rad. Kako bi se slika kasnije mogla obraditi kameru je potrebno povezati sa Matlab razvojnim okruženjem. Prvo se naredbom *imaqhwinfo* pronađu svi mogući uređaji pomoću kojih se može dobiti slika. Daljnjim dodavanjem parametara naredbi *imaqhwinfo* prikazu se informacije o odabranom uređaju, u ovom slučaju web kameri. Prikaz informacija na slici Sl.3.1..

```
>> imaqhwinfo('winvideo',1)

ans =

    DefaultFormat: 'MJPG_1280x720'
    DeviceFileSupported: 0
    DeviceName: 'TOSHIBA Web Camera - HD'
    DeviceID: 1
    VideoInputConstructor: 'videoinput('winvideo', 1)'
    VideoDeviceConstructor: 'imaq.VideoDevice('winvideo', 1)'
    SupportedFormats: {1x14 cell}
```

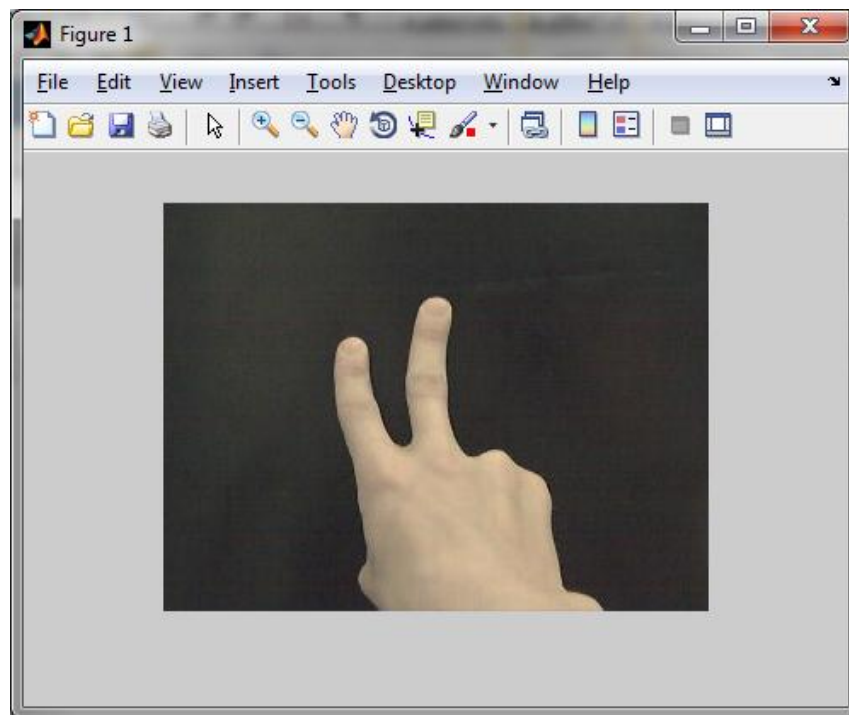
Slika 3.1. Prikaz informacija o Toshiba web kameri

Nakon što se odabere željena veličina slike, sve je spremno za izradu slike. Kako bi se dobio uvid u ono što kamera trenutno snima potrebno je upisati naredbu *preview* sa odgovarajućim parametrom prikazano na slici Sl.3.2.. Ukoliko se želi snimati koriste se naredbe *start* i *stop* za pokretanje, odnosno zasustavljanje snimanja. U ovom slučaju one se ne koriste jer je potrebna slika trenutnog stanja na kameri.



Slika 3.2. Prozor koji prikazuje što kamera snima

Sada kada se točno zna što kamera prikazuje može se izraditi fotografija i to na način da se prvo definira nekakav objekt s prikladnim imenom. Nakon toga, stvorenom objektu se pridoda naredba *getsnapshot* s odgovarajućim parametrom i trenutna slika onoga što kamera prikazuje će biti izrađena. Kako je sliku potrebno obraditi mora se pripaziti na pozadinu. U ovome slučaju odabrana je crna pozadina kako bi se ruka bila u što boljem kontrastu. Postizanjem kontrasta omogućuje se lakša obrada slike. Prikaz napravljene fotografije na slici S1.3.3..



Slika 3.3. Prikaz napravljene fotografije

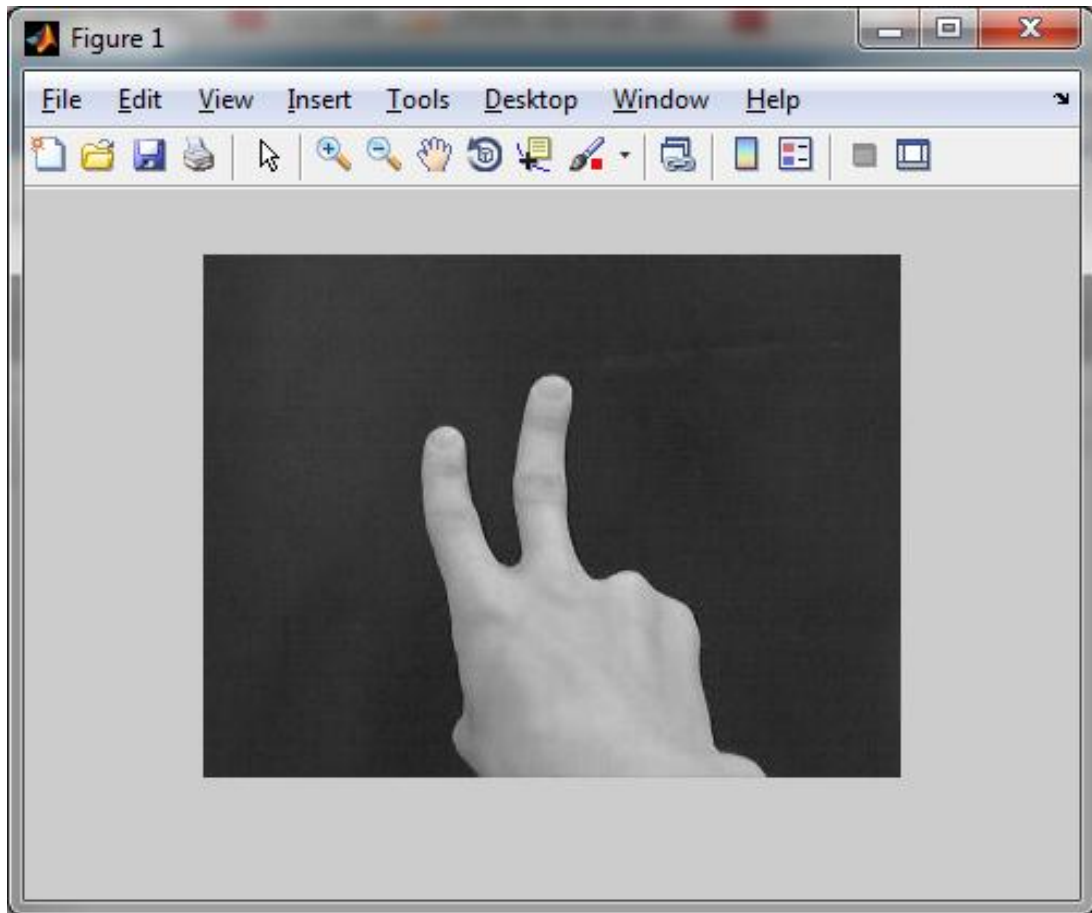
3.2.Obrada fotografije

Kada je slika napravljena može se početi s njenom obradom. Prije obrade potrebno je odlučiti na koji će se način prepoznati koliko je prstiju na slici. U ovom slučaju odabire se samo jedan redak slike i na temelju padajućih i rastućih bridova prepoznaje se broj prikazanih prstiju. Nakon što je određena metoda detekcije može se početi sa obradom slike. Za obradu se koriste brojne mogućnosti koje pruža Matlab okruženje. Prvo je izvornu sliku potrebno pretvoriti u sliku koja se sastoji samo od sivih tonova. Takav postupak izvršava se naredbom *rgb2gray* [8]. Spomenuta naredba prebacuje RGB (red,green,blue) vrijednosti u vrijednosti sivih tonova na temelju formule (3-1).

$$0.2989*R + 0.5870*G + 0.1140*B$$

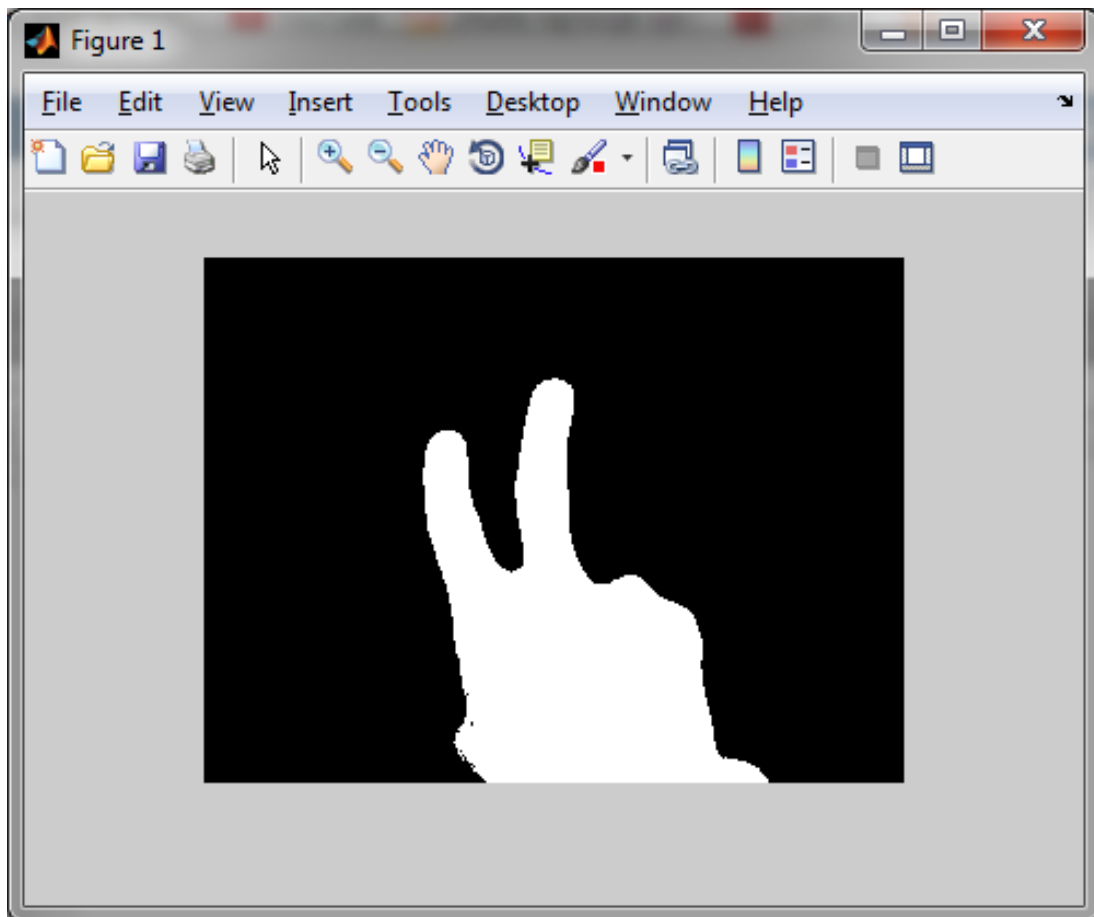
3-1

Nakon ovoga postupka pretvorbe izvorna slika sastoji se samo od sivih tonova, prikazano na slici Sl.3.4..



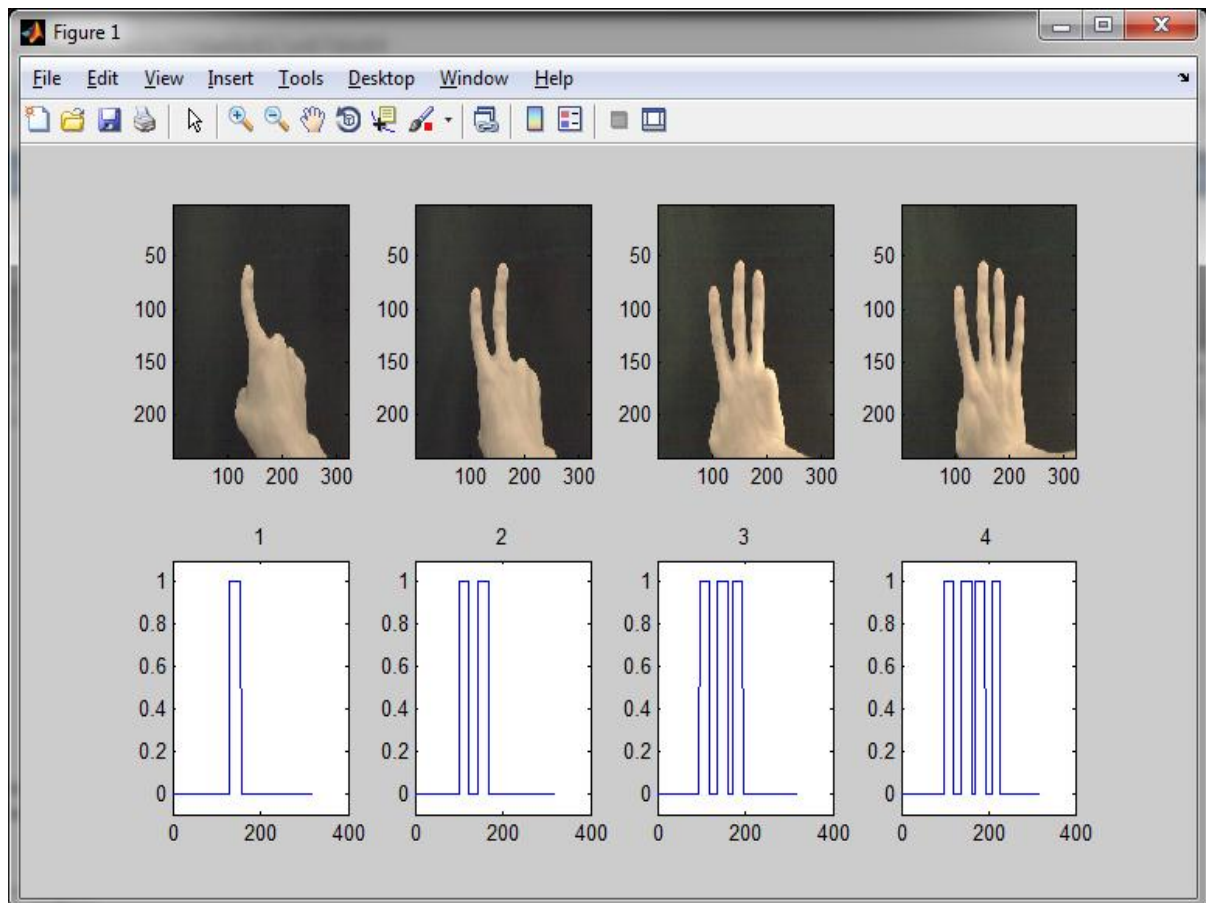
Slika 3.4. Prikaz slike nakon rgb2gray pretvorbe

Nakon pretvorbe, sliku je potrebno prebaciti u crno-bijelu. Sivi tonovi koji sačinjavaju sliku tvore skalu od 0 do 255, gdje nula predstavlja crnu boju, a 255 bijelu. Kako bi se slika prebacila u crno-bijelu odnosno binarnu sliku potrebno prvo definirati prag na temelju kojeg će se to ostvariti. Ukoliko je vrijednost određenog dijela slike veća od definiranog praga, taj će se dio prebaciti u bijelu boju. Također, ako je vrijednost određenog dijela slike manja od definiranog praga, taj će se dio prebaciti u crnu boju. U ovom slučaju odabran je prag 80 i svaki pojedini dio slike veći od 80 pretvoren je u bijelo, a manji od 80 u crno. Prikaz binarne slike na Sl.3.5..



Slika 3.5. Prikaz binarne (crno-bijele) slike

Nakon što je binarna slika uspješno dobivena može se odrediti broj prstiju pokazanih na slici. Detekcija se vrši već navedenom metodom padajućih i rastućih bridova i to na način da se prate dvije susjedne vrijednosti bridova za zadani red slike. Kako bi se navedena funkcionirala potrebno je definirati dvije varijable $x1$ i $x2$ za vrijednosti bridova, varijablu d koja predstavlja razliku između dviju vrijednosti bridova te varijablu n u koju će se zbrajati samo rastući bridovi. Imena varijabli nisu određena, odnosno zadaju se po volji. Dakle, ukoliko je razlika između vrijednosti dvaju susjednih vrijednosti bridova veća od nule, tada se pojavio rastući brid i vrijednost pohranjena u varijabli d se povećava za jedan. Ako razlika dvaju susjednih vrijednosti bridova nije veća od nule vrijednost pohranjena u varijabli d ostaje nepromijenjena. Ovaj postupak provjere vrijednosti dvaju susjednih vrijednosti bridova ponavlja se sve dok se ne dođe do kraja reda slike za koji se provjera izvršava. Nakon završetka provjere kao rezultat se dobije slika sa rastućim i padajućim bridovima te detektiranim brojem koji je bio prikazan na slici, prikazano na Sl.3.6..

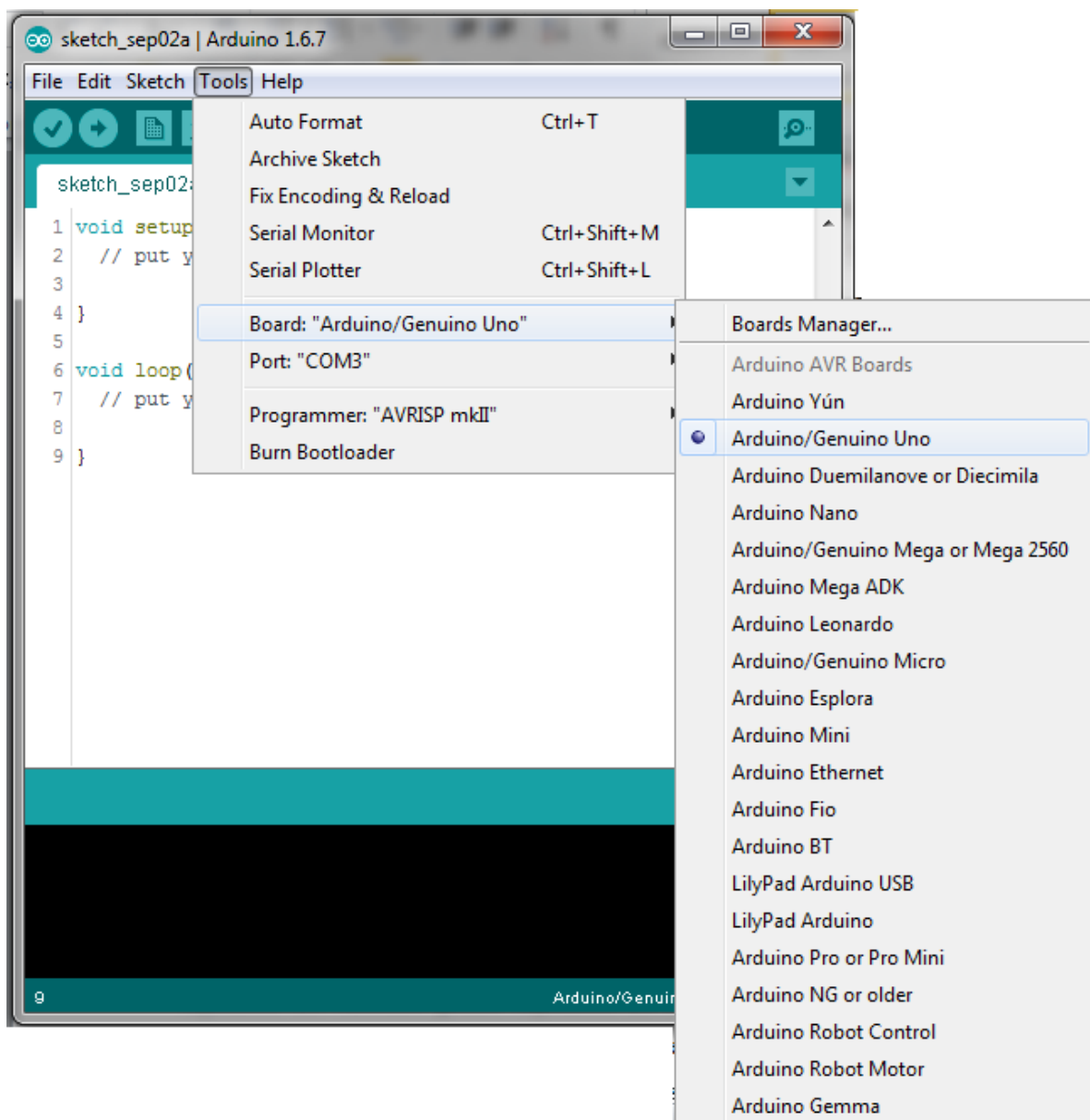


Slika 3.6. Prikaz rezultata detekcije broja prstiju metodom rastućih bridova

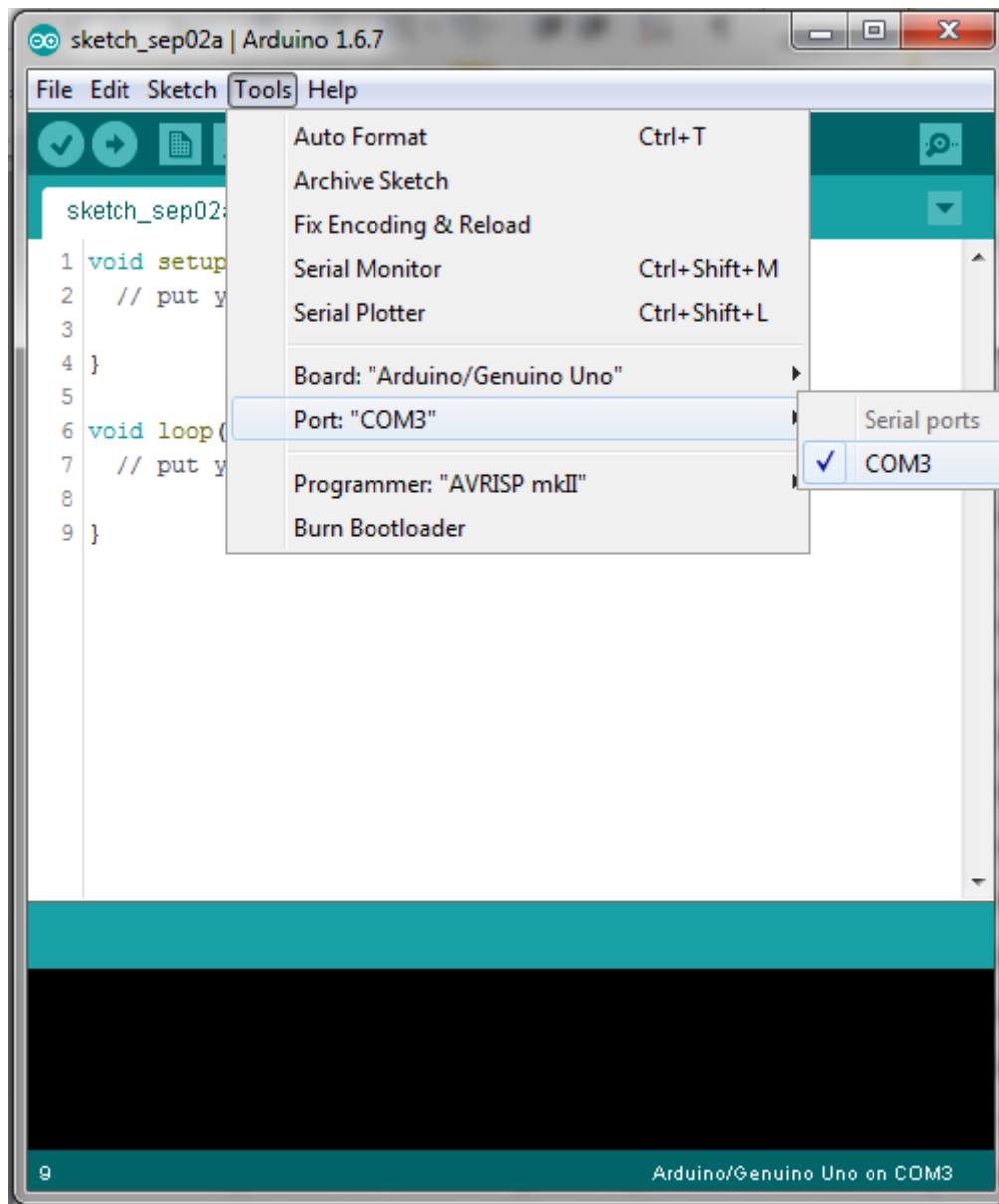
Nakon uspješne detekcije broja prstiju pruža se mogućnost upravljanja određenim uređajem. Kako bi se to u potpunosti ostvarilo, potrebno je prvo omogućiti komunikaciju između Matlab razvojnog okruženja i Arduino razvojnog okruženja.

3.3. Upravljanje uređajem na temelju fotografije

Nakon uspješno odrađene obrade slike i dobivenog broja prstiju koji je pokazan na slici može se relizirati i posljednji dio sustava, upravljanje uređajem. Na samom početku je naveden ArduinoUNO modul koji će se koristiti za upravljanje. Kako bi to bilo moguće, potrebno je ostvariti komunikaciju, u ovom slučaju serijsku [9]. Program uspostavlja serijsku komunikaciju na serijskom komunikacijskom sučelju (engl. COM port) između Arduino razvojnog okruženja i MATLAB razvojnog okruženja. Prvo je u Arduino okruženju potrebno odabrati ArduinoUNO modul i port na kojem se ArduinoUNO modul nalazi. Prikaz odabira modula i porta na slikama SI.3.7. i SI.3.8..



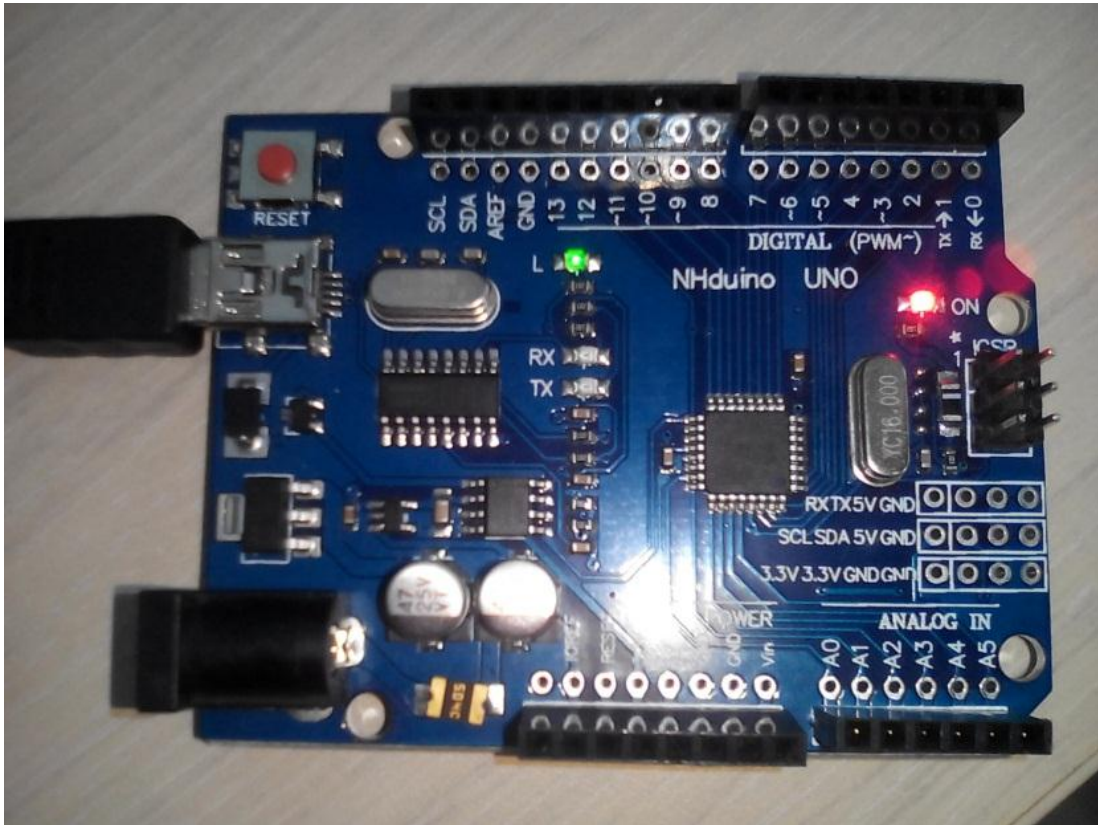
Slika 3.7. Prikaz odabira modula koji se koristi



Slika 3.8. Prikaz odabira porta

Nakon odabira porta i modula, potrebno je odrediti brzinu prijenosa podataka. Za ovaj sustav brzina potrebna za prijenos podataka iznosi 9600 bauda. Kako bi komunikacija bila uspješna, port i brzinu prijenosa potrebno je definirati i u MATLAB razvojnom okruženju. Vrijednosti porta i brzine moraju biti identične onima u Arduino razvojnom okruženju. Za pokretanje serijske komunikacije koristi se naredba *fopen* koja kao parametar ima deklariran objekt serijske komunikacije sa odabranim portom i brzinom prijenosa podataka. Nakon što se pokrene komunikacija, sve vrijednosti dobivene obradom slike mogu se koristiti za upravljanje uređajem. Kako bi se nekakva vrijednost poslala iz MATLAB-a na Arduino koristi se naredba *fprintf* koja za parametre ima objekt serijske komunikacije i nekakvu vrijednost koja se šalje na Arduino. U

ovom slučaju se pomoću jednostavne *if* petlje ispituje broj pokazanih prstiju te se u slučaju broja dva na Arduino šalje vrijednost 20. Arduino prima vrijednost 20 te također pomoći *if* petlje pali „ledicu“. U slučaju broja tri na Arduino se šalje vrijednost 30 i „ledica“ se neće upaliti, odnosno ugasiti će se. Nakon što se izvrši jedna od naredbi komunikacija se zatvara pomoću naredbe *fclose*. Prikaz upaljene ledice na slici Sl.3.9..



Slika 3.9. Prikaz upaljene ledice nakon uspješne detekcije broja prstiju

4. ZAKLJUČAK

Tema ovog rada je bila „Mikroupravljački sustav za prepoznavanje položaja ruku“. Sustav je podijeljen na tri dijela te su sva tri, uz manje probleme, uspješno izvedena. Problem u nedostatku memorije i snage Arduino modula da se obradi slika uspješno je riješen na način da se programski dio koji je kompletan trebao biti u Arduino razvojnom okruženju, podijelio na dio u MATLAB razvojnom okruženju i dio u Arduino razvojnom okruženju. Dio u MATLAB razvojnom okruženju omogućuje izradu fotografije korištenjem kamere putem serijske komunikacije. Fotografija se zatim obrađuje algoritmom koji je razvijen u MATLAB razvojnom okruženju. Nakon obrade fotografije, ista se koristi kako bi se upravljalo određenim uređajem, u ovom slučaju „ledicom“ na Arduinu. Sustav bi se uz dodatno ulaganje i kupnju boljih komponenti mogao ubrzati i poboljšati, a samim time i koristiti za razne svrhe.

LITERATURA

[1] Arduino razvojno okruženje, 23.6.2016.

<https://www.arduino.cc/en/Guide/Introduction>

[2] Atmel 328P mikroprocesor, 23.6.2016.

http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf

[3] MATLAB razvojno okruženje, 24.6.2016.

<http://www.mathworks.com/products/matlab/>

[4] ArduinoUno modul, 20.8.2016.

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

[5] Arduino Mega modul

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

[6] Raspberry Pi moduli

<https://www.raspberrypi.org>

[7] BeagleBoard moduli, 20.8.2016.

<http://beagleboard.org/boards>

[8] Prikaz slike pomoću sivih tonova, 20.8.2016

<http://www.mathworks.com/help/matlab/ref/rgb2gray.html>

[9] Serijska komunikacija, 20.8.2016.

<http://www.mathworks.com/help/supportpkg/arduino/ug/use-serial-communications-with-arduino-hardware.html>

SAŽETAK

Naslov: Mikroupravljački sustav za prepoznavanje položaja ruku

U ovom radu izrađen je mikroupravljački sustav za prepoznavanje položaja ruku. Sustav se sastoji od tri glavna dijela: dio za izradu slike, obrada napravljene slike i upravljanje uređajem na temelju slike, odnosno položaja ruke. Tijekom rada navedeni su i objašnjeni alati i komponente koje su korištene za izradu sustava. Nakon opisa alata i komponenti, detaljnije je objašnjena realizacija sustava.

Ključne riječi: mikroupravljači sustav, položaj ruku, obrada slike

ABSTRACT

Title: Microprocessor system for hands gesture recognition

In this work microprocessor system for hands gesture recognition was developed. It consists of three main parts: system for making pictures, image processing and device managment based on image, respectively hand gesture. Firstly, the tools and components that are used in system development are described. After that, the system implementation is described.

Keywords: microprocessor system, hands gesture, image processing

ŽIVOTOPIS

Matej Tomić rođen je 10. studenoga 1994. godine u Našicama. Od svoga rođenja živi u Našicama. U 2009. godini upisuje u opću gimnaziju u Srednoj školi Isidora Kršnjavog u Našicama. U trećem razredu srednje škole sudjeluje na školskom, a zatim i na županijskom natjecanju iz informatike pod nazivom „Infokup“. Srednju školu završava redovno u 2013. godini te u istoj upisuje preddiplomski studij računarstva na Elektrotehničkom fakultetu u Osijeku.

PRILOG A. Program za Arduino Uno razvojni sustav

```
const int ledpin=13;
int recbroj;

void setup()
{
  Serial.begin(9600);
  pinMode(13, OUTPUT);
}

void loop()
{
  if(Serial.available()>0)
  {
    recbroj=Serial.read();
    if (recbroj == 20) // Ako se putem MATLAB-a pošalje vrijednost 20, onda upaliti
    ledicu

    { digitalWrite(ledpin, HIGH); }

    if(recbroj == 30) // Ako se putem MATLAB-a pošalje vrijednost 30, onda ugasiti ledicu

    { digitalWrite(ledpin, LOW); }

  }
}
```

PRILOG B. Program za MATLAB razvojni sustav – funkcija snapshot

```
%Deklaracija objekta serijske komunikacije
s=serial('COM5','BAUD',9600);

%otvaranje serijskog porta
fopen(s);

%inicijaliziranje objekta za snimanje kamerom
Video=videoinput('winvideo',1,'MJPG_352x288');
preview(Video)

% izrada slike
img=getsnapshot(Video);

%pozivanje funkcije za obradu slike i detekciju
%broja pokazanih prstiju
fingercount
```

PRILOG C. Program za MATLAB razvojni sustav – funkcija fingercount

```
% deklaracija konstante
redak    = 108;
prag     = 80;

% stvaranje slike za prikaz rezultata
figure(1);

% slika u boji
I=img;

% sivi tonovi jednog retka slike
x = rgb2gray(I(108,:,:));

% crno-bijela slika
x(x<=prag)=0;
x(x>prag) =1;
% rastući bridovi

x1 = x;
x2 = [x(2:end),x(1)]; % prvi element stavljen na zadnje mjesto
d = x2-x1;
n = sum(d>0);

% prikaz rezultata
plot(x);
title(num2str(n)); ylim([-0.1,1.1]);
broj=n

%slanje na Arduino vrijednost 20 ako je broj prstiju 2, odnosno
%vrijednost 30 ako je broj prstiju 3
    if broj==2
        fprintf(s,20);
    end;
    if broj==3
        fprintf(s,30);
    end;

%zatvaranje serijskog porta
fclose(s);
```