Digitalna šahovska ploča

Sušac, Filip

Master's thesis / Diplomski 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://urn.nsk.hr/urn:nbn:hr:200:854045

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

zir.nsk.hr

Image not found or type unknown

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

Diplomski sveučilišni studij

DIGITALNA ŠAHOVSKA PLOČA

Diplomski rad

Filip Sušac

Osijek, 2016.

Sadržaj

1. UVOD 1

1.1. Zadatak diplomskog rada 1

2. PRIMJENJENJE TEHNOLOGIJE I ALATI 2

2.1. Konstrukcija šahovske ploče 2

2.2.Arduino razvojno okruženje 3

2.3.Arduino Nano razvojna maketa 5

2.4. Magnet i senzor magnetnog polja 6

3.REALIZACIJA SUSTAVA 8

3.1.Električna shema sustava 8

3.2.Izrađena maketa sustava 11

3.4. Arduino program 15

3.3.Grafičko sučelje 15

3.4. Testiranje i rad digitalne šahovske ploče 17

3.5. Troškovnik 18

4. Zaključak 19

LITERATURA 20

SAŽETAK 21

ABSTRACT 22

ŽIVOTOPIS 23

PRILOZI 24

1. UVOD

Cilj ovog diplomskog rada je izraditi maketu digitalne šahovske ploče. Maketa se može podijeliti na dvije cjeline: sklopovsku i programsku. Prisutnost šahovske figure koja u sebi ima ugrađeni magnet detektirati će senzor Hallovog efekta koji se nalazi pričvršćen ispod svakog šahovskog polja. Prikupljanje informacija od senzora Hallovog efekta obavljati će se pomoću Arduino platforme koja će u konačnici pomoću serijske komunikacije slati 64-bitni zapis Visual Basic-u u kojemu je izgrađeno grafičko sučelje s algoritmom koji prikazuje trenutno stanje ploče.

.

1.1. Zadatak diplomskog rada

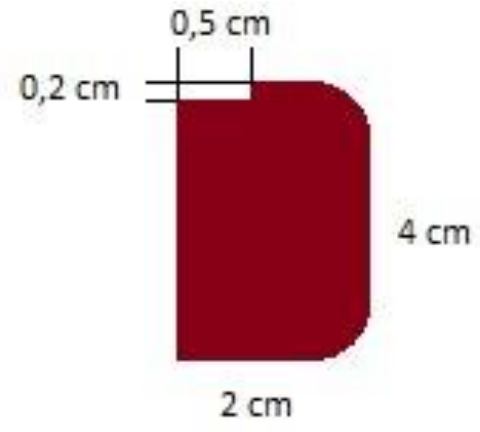
Šahovskim figurama potrebno je ugraditi permanentni magnet. Senzor Hallovog efekta ugrađen je ispod pojedinog polja šahovske ploče. Pomoću magneta i senzora magnetnog polja potrebno je detektirati zauzeće šahovskog polja. Na temelju promjena zauzeća polja potrebno je napisati program koji prepoznaje poteze i prikazuje poziciju na ekranu računala. U ovom radu koristi se mikrokontroler sa Arduino ili Atmel studio programskim okruženjem.

2. PRIMJENJENJE TEHNOLOGIJE I ALATI

2.1. Konstrukcija šahovske ploče

Najprije je bilo potrebno odrediti veličinu jednog šahovskog polja, a samim time i ukupnu veličinu makete. S obzirom na šahovske standarde maketa će imati standardnu veličinu šahovskog polja dimenzija 5 x 5 cm, što u konačnici rezultira šahovskom pločom veličine 50 x 50 cm, od čega je 40 x 40 cm šahovsko polje a dodatnih 5cm obruba sa svake strane.

S obzirom da prilikom izrade makete koriste se senzori Hallovog efekta za materijal ploče korištena je akrilna (pleksiglas) ploča kroz koju neometano prolazi magnetno polje potpuno nedeformirano. Završna izrada okvira digitalne šahovske ploče biti će izrađena od drvenih letvica širine 2 cm i visine 4 cm koje je potrebno dodatno obraditi kako bih akrilna ploča bez dodatnog učvršćivanja mogla biti postavljena u okvir. Na slici 2.1. prikazana je dimenzija jedne letvice s obradom. Podnožje digitalne šahovske ploče je lesonit ploča ukupnih dimenzija 52 x 52 cm debljine 0,4cm.



Slika 2.1. Dimenzije letvice

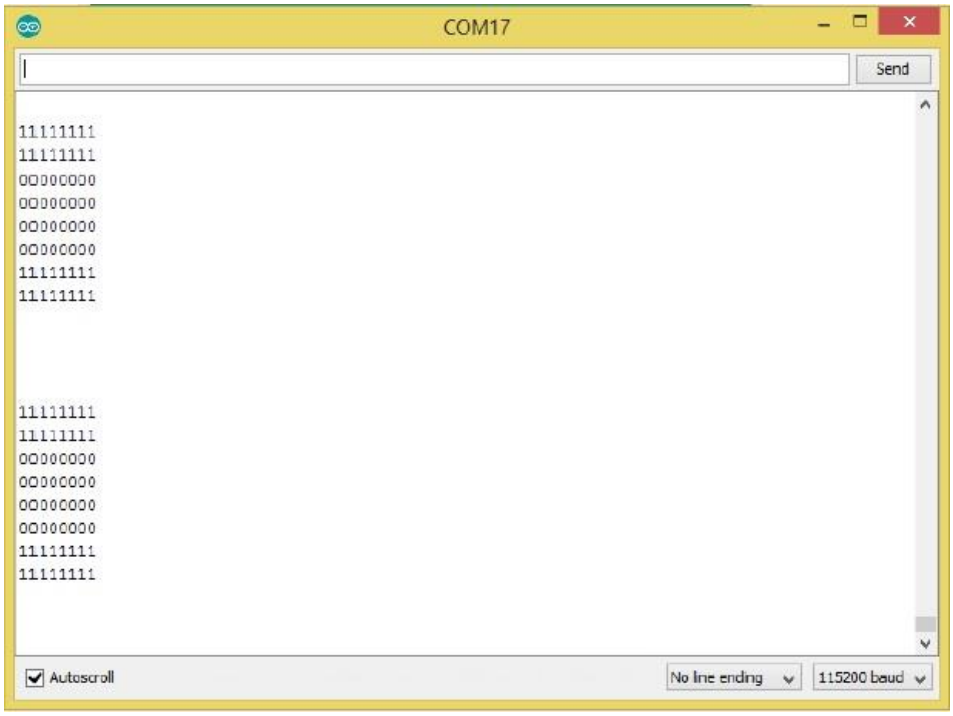
2.2.Arduino razvojno okruženje

Arduino je "open-source" platforma za kreiranje elektroničkih prototipova bazirana na sklopovlju i programskom paketu koji je fleksibilan i jednostavan za korištenje, odnosno Arduino platforma je skup elektroničkih i programskih dijelova koji se mogu jednostavno povezati u složenije cjeline. Arduino kod može se podijeliti u dva dijela: *setup* i *loop*. Unutar *setup* dijela upisani su dijelovi koda koji se izvršavaju samo jednom, prilikom inicijalizacije programa. U tom dijelu definiraju se varijable i konstante. *Loop* dio nam služi za dio koda koji se neprestano izvršava. U ovome dijelu smo definirane su naredbe koje dohvaćaju stanja sa senzora Hallovog efekta te ih šalje serijskom komunikacijom. Izgled Arduino sučelja prikazan je na slici 2.2.



Slika 2.2. Arduino sučelje

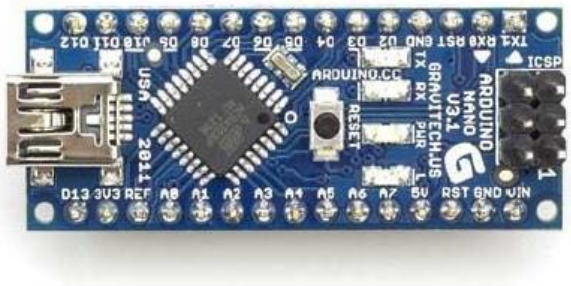
U prilogu A. nalazi se programski kod izrađen u Arduino programskom okruženju. Zadatak i cilj programa je da trenutna stanja očitana sa senzora Hallovog efekta dohvati i pomoću serijske komunikacije pošalje u Visual Basic razvojno okruženje. Na slici 2.3. prikazan je "Serial Monitor". Serial Monitor je zasebni skočni prozor Arduino razvojnog okruženja koji se ponaša kao odvojeni terminal koji komunicira primajući i šaljući serijske podatke, u ovome slučaju pokazuje matricu oblika 8x8 koja predstavlja šahovsko polje.



Slika 2.3. Arduino sučelje

2.3.Arduino Nano razvojna maketa

Prilikom odabira Arduino uređaja, kao dva najvažnija kriterija promatrani su cijena i veličina sklopa i na temelju toga odlučeno je koristiti Arduino Nano uređaje verzije 3.0 prikazani na slici 2.4.



Slika 2.4. Arduino Nano verzija 3.0[1]

Arduino Nano je razvojna maketa s ugrađenim USB priključkom. Trenutno je najmanja i potpuna razvojna maketa. U odnosnu na prošle verzije Arduino razvojne maketa fizički nedostatak Nano verzije leži u tome da nema dodatni ulaz za napajanje, dok u električnom smislu ima prednosti u vidu veće memorije i više analognih ulaza. Izrađena je u dva sloja.

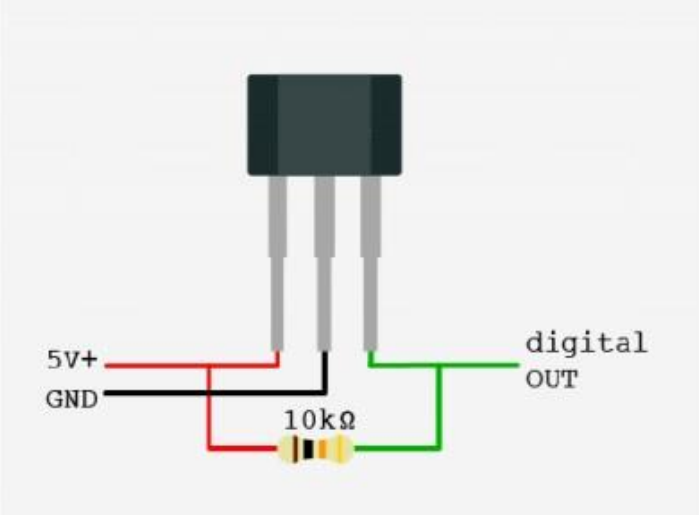
Karakteristike Arduino Nano v3.0 razvojne makete:

- Mikrokontroler: Atmel ATmega328
- Radni napon (logička razina) : 5V
- Ulazni napon (preporučen): 7-12 V
- Digitalni ulazno/izlazni pinovi: 14
- Analogni ulazni pinovi: 8
- Istosmjerna maksimalna struja po pinu: 40mA
- *Flash* memorija: 32KB
- SRAM: 2KB
- EEPROM: 1KB
- Frekvencija takta: 16MHz
- Dimenzije 18 x 45 mm
- Težina: 5g

[1] Izvor: https://www.arduino.cc/en/uploads/Main/ArduinoNanoFront_3_sm.jpg

2.4. Magnet i senzor magnetnog polja

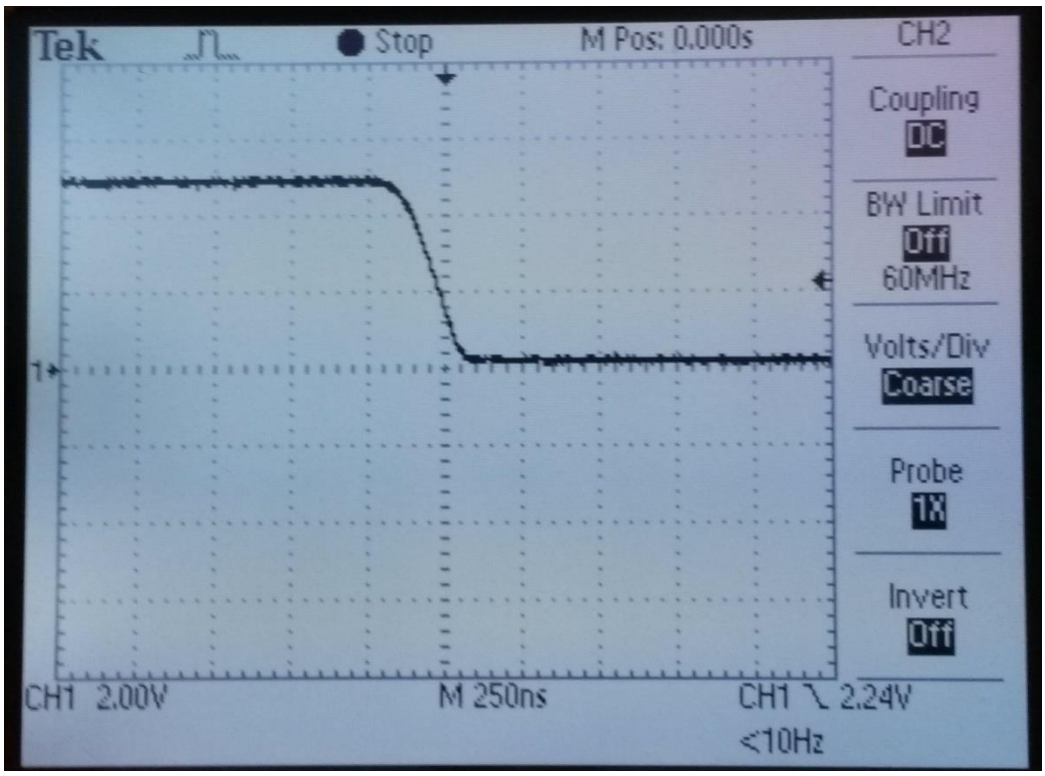
Kao senzor magnetnog polja korišten je senzor Hallovog efekta model A3141. Kompletne karakteristike senzora nalaze se u *datasheet*-u priloženom uz ovaj rad. Osnovni princip rada senzora Hallovog efekta je registriranje prisutnosti magnetskog polja. S obzirom na primjenu u ovome diplomskom radu nije potrebno razlučivati jakost izlaza senzora Hallovog efekta jer u radu radimo sa stanjima „postoji figura na polju“ i „ne postoji figura na polju“, odnosno dovoljna nam je informacija o prisutnosti magneta ali ne i o njegovom intenzitetu. Na slici 2.5. prikazan je senzor Hallovog efekta sa načinom spajanja.



Slika 2.5 Senzor Hallovog efekta A3141[2]

Eksperimentalni podatci pokazuju da vrijeme potrebno za promjenu stanja senzora Hallovog efekta je u okviru 250 nanosekundi, što je prikazano na slici 2.6. Informacija o potrebnom vremenu izmjene koristi nam prilikom pisanja programskog koda.

[2] Izvor: http://bildr.org/blog/wp-content/uploads/2011/04/us1881-400x295.png?width=500



Slika 2.6. Vrijeme promjene stanja senzora Hallovog efekta

Unutar šahovskih figura potrebno je implementirati permanentni magnet dovoljno snažan da bi senzor Hallovog efekta mogao prepoznati njegovo magnetsko polje. Eksperimentalni rezultati permanentnog magneta oblika stošca promjera 10mm i 2mm visine potvrđuju da dva magneta stvaraju dovoljno jako magnetsko polje potrebno za detekciju na senzoru Hallovog efekta. Na slici 2.7. prikazan je korišteni magnet.

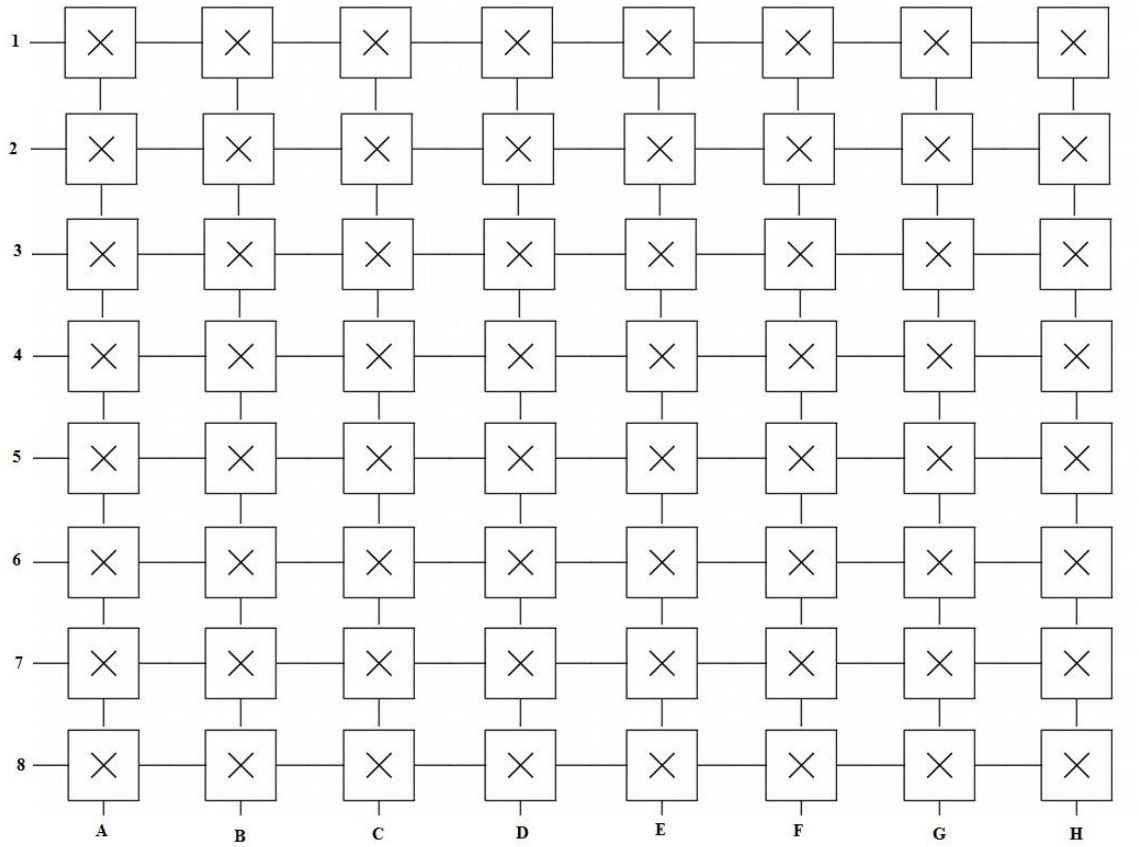


Slika 2.7. Permanentni magnet

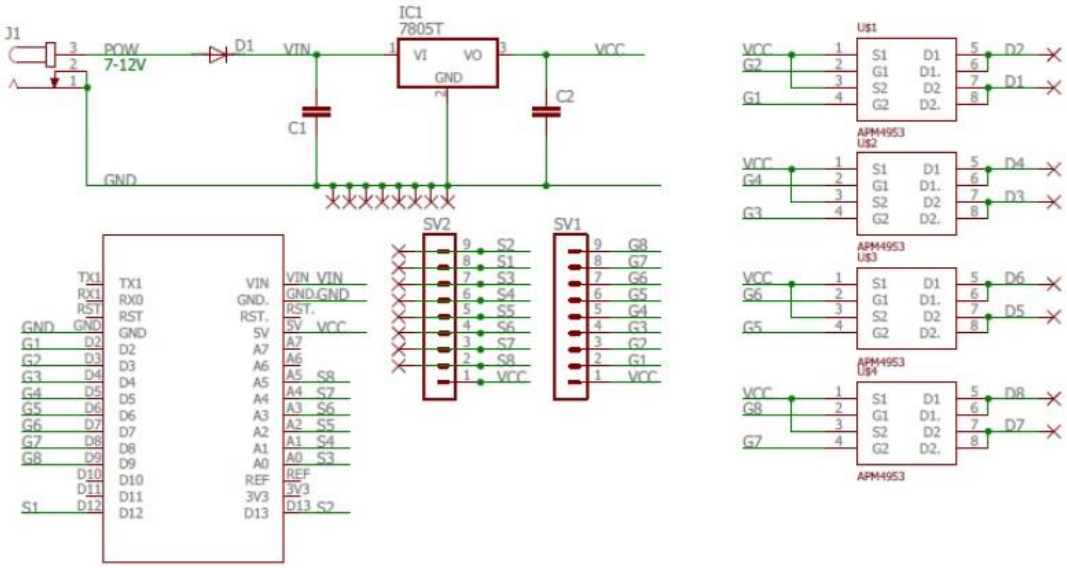
3.REALIZACIJA SUSTAVA

3.1.Električna shema sustava

Ispod svakog šahovskog polja nalazi se po jedan senzor Hallovog efekta. S obzirom na broj šahovskih polja korišteno je ukupno 64 senzora. Ograničenost ulaznih/izlaznih pinova Arduino uređaja onemogućuje spajanje svakoga senzora posebno te je potrebno spojiti senzore kreirajući matricu 8 x 8, na taj način smo ukupan broj ulaza i izlaza smanjili sa 64 na 16. Osam ulaznih linija i osam izlaznih linija, dok uzemljenje („GND“) je povezano između svih 64 senzora te dovedeno na uzemljenje Arduino uređaja. Prikaz sheme spajanja senzora Hallovog efekta prikazan je na slici 3.1 .

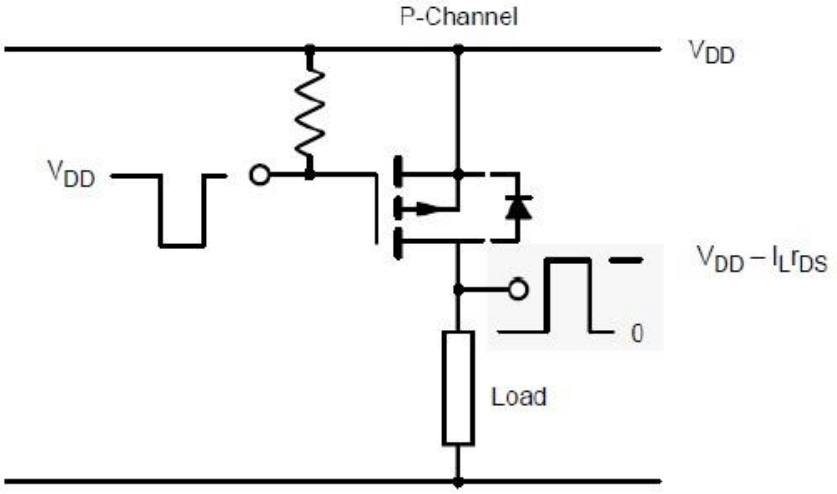


Slika 3.1. Prikaz načina spajanja Hallovih senzora

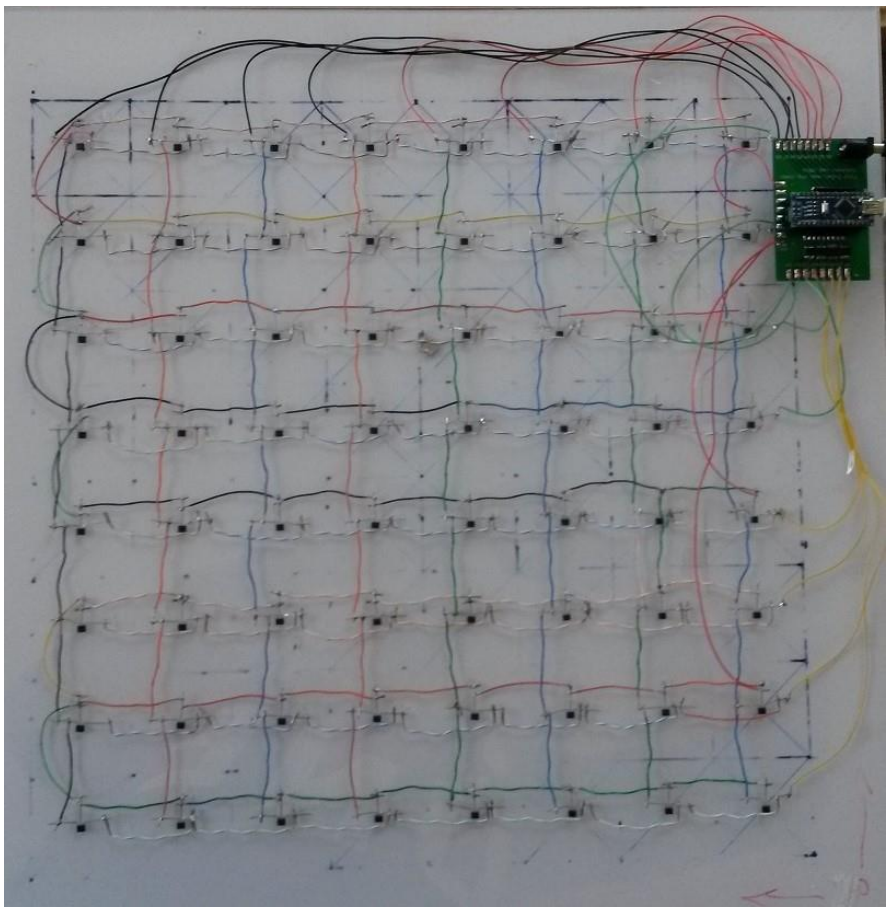


Slika 3.2. Shema sklopa za detekciju šahovskih figura

Na slici 3.2. prikazana je shema sklopa za detekciju šahovskih figura. Shema se sastoji od Arduino uređaja, dodatnog sklopa za napajanje koji za ulaz prima 7-12 V napon i regulira ga na izlazni napon VCC od 5 V, Pull-up otpornika koji su neophodni za pravilno spajanje senzora Hallovog efekta i četiri P-kanalna MOSFET tranzistora APM4953 koji upravljaju dovođenjem napona na redove. Shema rada P-kanalnog MOSFET tranzistora prikazana je na slici 3.3. Dobiveni rezultati mjerenja jakosti struje potrebnog za nesmetan rad svih senzora pokazuju da minimalna jakost struje po jednoj liniji (pinu) treba iznositi 100 mA. S obzirom na karakteristike Arduino Uno uređaja, navedene u drugom poglavlju ovoga rada, zaključujemo da Arduino Uno samostalno ne može napajati sve senzore i samim time samostalno ne može upravljati maketom. Iz tog razloga uvođenje dodatnog napajanja je neophodno za pravilan rad sklopa. S tiskane pločice pomoću P- kanalnih MOSFET tranzistora na redove dovodimo napajanje od 5 V dok izlazna stanja senzora Hallovog efekta snimamo po stupcima.



Slika 3.3. Način rada P-kanalnog MOSFET tranzistora[3]



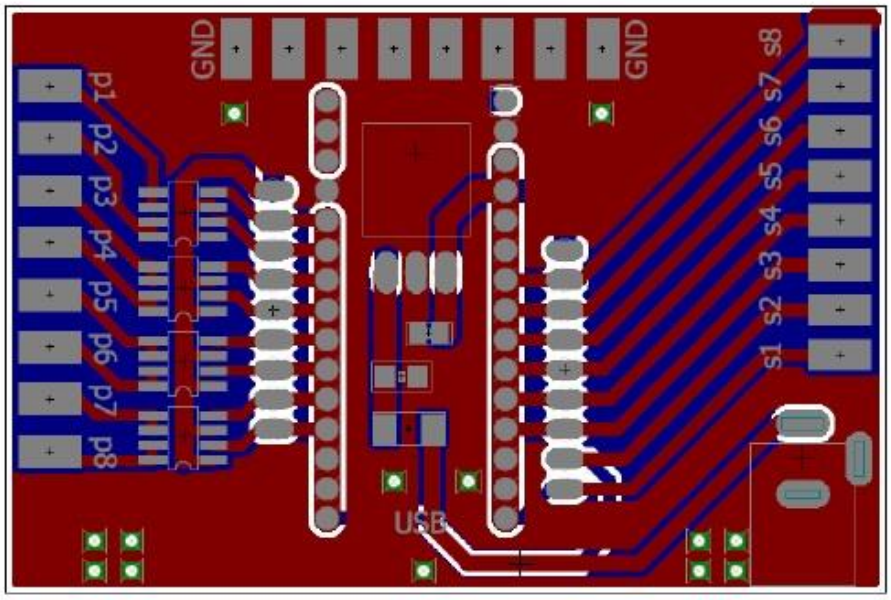
Slika 3.4. Unutarnja strana šahovske ploče

[3] Izvor: http://i.stack.imgur.com/bo9P2.png

Na slici 3.4 je prikazana unutarnja strana makete, na kojoj se jasno vidi shema spajanja senzora i mjesto na kojemu se nalazi upravljačka ploča.

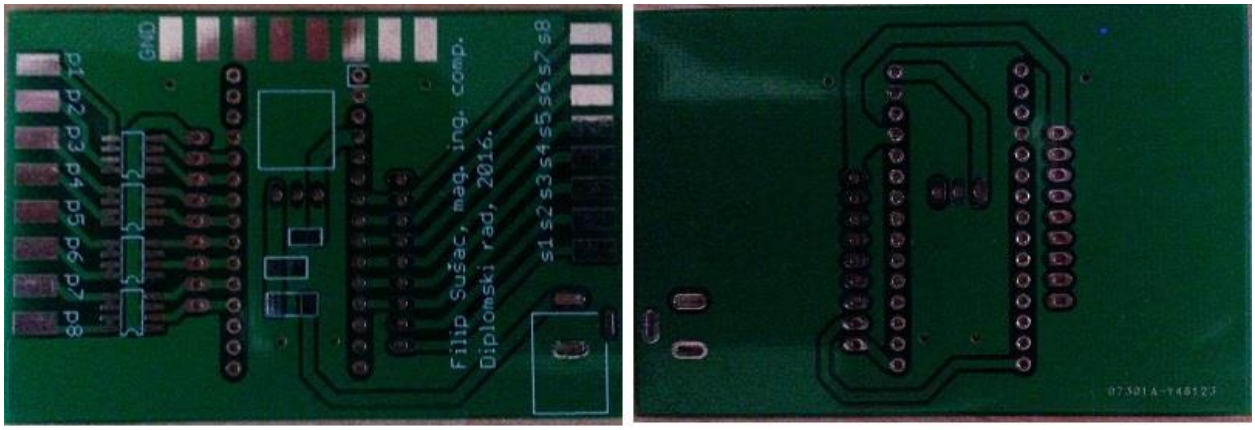
3.2.Izrađena maketa sustava

Pri odabiru programskog paketa za izradu sheme tiskane ploče razmotrene su dvije mogućnosti: CadSoft EAGLE i KiCad. Nakon detaljnijeg upoznavanja s programskom podrškom, iako su programski paketi vrlo slični, odlučeno je raditi u CadSoft EAGLE-u zbog većeg broja korisnika te detaljnije i veće baze podataka komponenti. Programski paket CadSoft EAGLE omogućuje brzu i jednostavnu izradu elektroničkih shema i tiskanih pločica. Projektirana shema tiskane pločice naručena je i izrađena od strane ITEAD[4] kompanije. Model tiskane pločice dimenzija 5 x 10 cm prikazan je na slici 3.5., dok na slici 3.6 je prikazana izrađena tiskana pločica korištena u ovome radu.



Slika 3.5. Model tiskane pločice

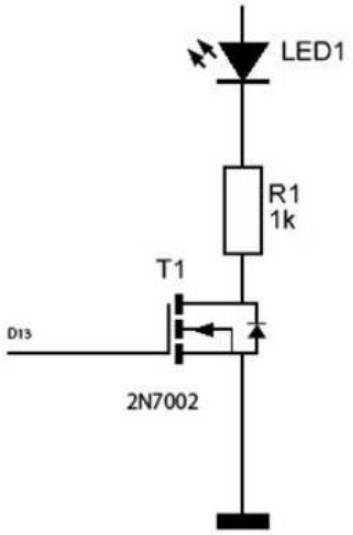
[4] https://www.itead.cc/



Slika 3.6. Tiskana pločica: a) gornja strana b) donja strana

Na Arduino uređaju korišteno je osam pinova za naponske, odnosno izlazne, signale i osam za ulazne signale. Pinovi D2,D3,D4,D5,D6,D7,D8,D9 korišteni su kao naponski signali koji služe za dovođenje napona u redove dok su pinovi D12,D13,A0,A1,A2,A3,A4,A5 korišteni kao ulazni pinovi odnosno služe za snimanje stanja izlaza senzora Hallovog efekta. Prilikom korištenja D13 kao ulaznog pina napravljena je modifikacija Arduino Nano razvojne pločice verzije 3.0. Uklonjena je dioda D13, te je time prekinut strujni krug koji je ometao ulazni signal stavljajući fiksni napon ulaza na logičku nulu. Alternativno rješenje prilikom izrade pločice koje bi uklonilo ovaj neželjeni efekt prikazano je na slici 3.7[5]

[5] Detaljno objašnjenje: https://arduinodiy.wordpress.com/2012/04/28/solving-the-d13-problem-on-your-arduino/



Slika 3.7. Otklanjanje neželjenog efekta sa pina D13[6]

Pomoću tokarskog stroja na dnu svake figure napravljena je centrirana udubina oblika stošca promjera 11 mm i dubine 4,5 mm, koja je dovoljna za smještaj dva permanentna magneta i tanki sloj silikonskog lijepila kako bi isti bili trajno pričvršćeni za šahovsku figuru. Na slici 3.8. prikazana je šahovska figura nakon implementacije dva permanentna magneta.



Slika 3.8. Šahovska figura s permanentnim magnetom

[6] Izvor: https://arduinodiy.files.wordpress.com/2012/04/d13_thumb.jpg?w=281&h=397

Sama maketa sastavljena je od četiri drvene letvice, prethodno obrađene kako je navedeno u poglavlju 2.1., sastavljanje pomoću profesionalnog lijepila za drvo. Nakon sastavljanja okvira bilo je potrebno izbušiti dvije rupe, jednu za USB konekciju te jednu za napajanje sklopa. Dizajn šahovske ploče izrađen je u studio za obradu slike te naručen i izrađen u tiskarskoj radnji „BorovoGraf Vukovar d.o.o.“ . Naljepnica je PVC kvalitete trajno zalijepljena na akrilnu ploču korištenu pri izradi makete. Na slici 3.9. prikazana je ploča prije završne obrade i izgradnje okvira, postolja i naljepnice dok je na slici 3.10. prikazana gotova maketa digitalne šahovske ploče napravljene u ovome radu.



Slika 3.9. Izgled makete prije završne obrade



Slika 3.10. Konačni izgled digitalne šahovske ploče

3.4. Arduino program

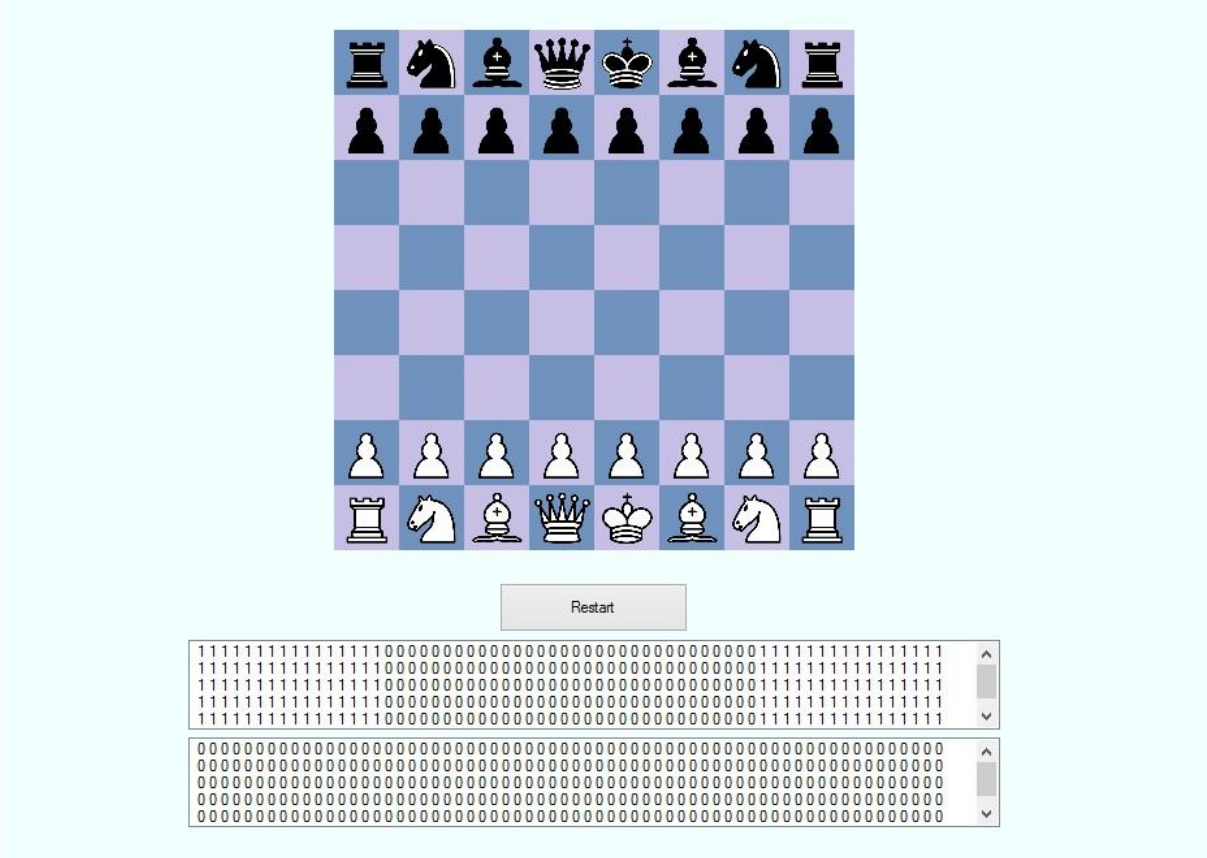
U prilogu A. priloženom uz ovaj rad nalazi se izvorni kod koji je učitan na Arduino Uno uređaj unutar makete. U *setup* dijelu definirani su ulazni i izlazni pinovi te *Baud Rate. Baud Rate* je postavljen na 115200, što predstavlja maksimalnu brzinu slanja podataka u mjernoj jedinici vremena bit u sekundi [bits/s]. Unutar *loop* dijela koda imamo petlju *while* koja provjerava dostupnost serijske komunikacije, ukoliko je komunikacija uspostavljena program čeka na znak „*2*“ koji šalje računalo, tek nakon primljenog znaka, postavlja vrijednost svih polja matrice *b* na „0“, zatim po redovima dovodi napajanje a po stupcima dohvaća stanja ploče. Nakon dohvaćanja svih vrijednosti sa ploče imamo matricu b koja sadrži vrijednost *„1“* na mjestima gdje je prisutna šahovska figura, odnosno na tim mjestima je senzor Hallovog efekta detektirao prisutnost magnetskog polja, a vrijednost *„0“* nam označava prazno polje. Pomoću naredbe *Serial.print* šaljemo matricu *b* putem serijske komunikacije.

3.3.Grafičko sučelje

Grafičko sučelje i program za obradu stanja ploča izrađen je u programskom okruženju Visual Studio 2015 Enterprise. Izvorni kod grafičkog sučelja (GUI-a) preuzet je sa izbornog kolegija „Šah i računala“[7] te nadograđen kako bi odgovarao zahtjevima ovoga rada. Na slici 3.9. Prikazan je konačni izgled grafičkog sučelja. Sučelje se sastoji od šahovske ploče, dva *listboxa* te dodatnog dugmeta koji služi za ponovno pokretanje aplikacije. U prvom *listbox-u* (drugi gledano od dolje prema gore) prikazuju se trenutna stanja pozicija figura u zapisu oblika niza od 64 znaka. Dok se u drugome *listbox-u* prikazuje matrica razlike pomoću koje se određuje promjena pozicije figure. Algoritam matrice razlike realiziran je na način da ukoliko je na jednome polju postojala figura (program dohvaća podatak „1“) koja je nakon toga pomaknuta (pogram dohvaća podatak „0“) na tome mjestu (odgovarajućem indexu) u matrici razlike se zapisuje vrijednost „-1“. Zatim nakon što tu istu figuru stavimo na polje koje je bilo prazno, odnosno imalo vrijednost „0“ prelazi u vrijednost „1“, u matrici razlike se zapisuje vrijednost „1“. U suštini

[7] Izborni kolegij na drugoj godini diplomskog studija procesnog računarstva.

unutar matrice razlike pronalazimo ukupno tri vrijednosti: „-1“ uzimanje figure sa polja, „1“ označava stavljanje figure na prazno polje i „0“ nam govori da na tome polju nema promjene. Na temelju matrice razlike i na temelju postavljene početne pozicije figura u mogućnosti smo u realnom vremenu prikazivati odgovarajuća pozicije šahovskih figura. Prije pokretanja programa potrebno je uspostaviti i sinkronizirati serijsku komunikaciju. Parametri koji su nam potrebni za uspješnu serijsku komunikaciju su: *COM port*, *BaudRate*, *Parity*, *Data bits i Stop bits*. *BaudRate* i *COM port* mora biti jednako postavljen kao i u Arduino kodu. U podrutini *Timer1_Tick* definiran je vremenski brojač koji šalje znak „2“ i prima niz od 64 znaka. Primljeni niz se sprema u funkciju *NovaPozicija()* i prikazuje u *ListBox1*. U podrutini *NovaPozicija()* definirano je spremanje primljenog niza kao nova pozicija šahovskih figura. Svaki dio spremanja novog stanja riješen je pomoću funkcija: *TrenutnaPozicija(), TrenutniIndexi(), CrtajPraznoPolje(), CrtajPolje().* Nazivi funkcija govore dijelu koda koje funkcija obavlja. Pomoćna funkcija *PrilagodiZaIspis()* nam služi za pregledniji ispis stanja ploče i matrice razlike zbog ispisivanja predznaka kod matrice razlike.



Slika 3.9. Grafičko sučelje

3.4. Testiranje i rad digitalne šahovske ploče

Nakon uspješne implementacije Arduino koda na konačni model makete digitalne šahovske ploče i završene aplikacije za grafičko sučelje testiran je rad ukupnog sustava. Kroz tri stanja prikazan je rad ploče na slikama 3.10, 3.11 i 3.12.



Slika 3.10. Početna pozicija



Slika 3.11. Stanje 1



Slika 3.12. Stanje 2

3.5. Troškovnik

U tablici 3.1. prikazani su ukupni troškovi izrade makete.

Naziv proizvoda	Cijena po komadu	Količina	Ukupna cijena
Arduino Uno	0,98 $	1	0,98 $ = 6,60 kn[8]
Senzor Hallovog efekta	0,057 $	64	3,65 $ = 24,57 kn
Izrada tiskane pločice	2,15 $	1	2,15 $ = 14,47 kn[9]
Pleksiglas ploča dimenzija 50 x 50 cm	50,00 kn	1	50,00 kn
Šahovske figure	70,00 kn	1	70,00 kn
Permanentni magnet	1,15 kn	64	73,60 kn
Izrada PVC naljepnice	80,00 kn	1	80,00 kn
Drvene letvice	8,00 kn	4	32,00 kn
Potrošni materijal	20,00 kn	1	20,00 kn
UKUPNO:		**140**	**371,24 kn**

Tablica 3.1. Troškovnik

[8] Konverzija valuta napravljena na dan 19.9.2016 koristeći srednji tečaj HNB-a.
[9] Minimalna narudžba tiskanih pločica od strane ITEAD studia je 10 komada.

4. Zaključak

U ovome radu napravljena je maketa digitalne šahovske ploče. Maketa je podijeljena na dvije velike cjeline: sklopovsku i programsku. Maketa je uspješno realizirana i obavlja očekivane funkcije. Prednosti izrade ovakve makete su dostupnosti materijala, niska cijena realizacije i jednostavnost korištenja. Kod sklopovskog dijela najveća mana je nemogućnost sklopovskog prepoznavanja figure, s obzirom na korištenje senzora Hallovog efekta za detekciju magnetnog polja u nemogućnosti smo točno odrediti koja je figura nad kojim poljem, uporabom RFID senzora ovaj problem bi bio uklonjen, no primjena istih bi u konačnici rezultirala višestrukim troškovima realizacije. Sklopovske nedostatke bilo je potrebno programski riješiti, s toga šahovska partija mora početi iz standardne početne pozicije. Smetnje na senzorima također mogu prouzročiti pogreške pri prikazu na računalu. Dodatnom optimizacijom koda i ugradnjom tipkala na šahovsku ploču mogli bi se kontrolirati potezi, te programski implementirati digitalni sat koji je dio standardne šahovske opreme.

LITERATURA

[1] Serijska komunikacija, http://www.martyncurrey.com/arduino-and-visual-basic-part-1-receiving-data-from-the-arduino/, rujan 2016.

[2] B. Newsome, Beginning Visual Basic 2015, John Wiley & Sons

[3] M. McRoberts, Beginning Arduino 2nd Ed, Technology in action

[4] Hallov efekt, http://hyperphysics.phy-astr.gsu.edu/hbase/magnetic/hall.html, lipanj 2016

[5] Izborni kolegij Šah i računala, https://loomen.carnet.hr/course/view.php?id=3721 , lipanj 2016

SAŽETAK

U ovome radu izrađena je maketa digitalne šahovske ploče. Maketa je podijeljena u dvije cjeline: sklopovsku i programsku. Materijal korišten za izradu okvira ploče i šahovskih figura je drvo, a sama ploča, dimenzija 50 x 50 cm, je od akrila. Unutar svake figure ugrađena su dva magneta koji stvaraju dovoljno jako magnetsko polje za aktivaciju senzora Hallovog efekta, pričvršćenog ispod akrilne ploče. Za prikupljanje informacija od senzora korišteno je Arduino razvojno okruženje koje pomoću serijske komunikacije prosljeđuje podatke grafičkom sučelju koje je izrađeno u Visual Basic razvojnom okruženju.

Ključne riječi: šah, Arduino, Hall-ov efekt, VB.Net, Visual Basic

ABSTRACT

Title: Digital chess board

This thesis represents a model of digital chess board. The model is divided into two parts: hardware and software. The material used to make the frame panel and chess pieces is a wood, panel dimensions are 50 x 50 cm made of acrylic. Within each figure are incorporated two magnets that generate a magnetic field strong enough to activate the Hall sensor attached underneath an acrylic panel. To collect information from sensors we used the Arduino development environment. With serial communication data is transmitted to a graphical interface that was made in the Visual Basic development environment.

Key words: chess, Arduino, Hall-ov efect, VB.Net, Visual Basic

ŽIVOTOPIS

Filip Sušac rođen je 03. prosinca 1991. godine u Zagreb, u kojem živi sve do početka mirne reintegracije, kada sa svojom obitelji dolazi u Vukovar. Osnovnoškolsko obrazovanje stječe u VII Osnovnoj školi u Vukovaru. U 2007. upisuje opći smjer Gimnazije Vukovar, koji redovno završava 2010. godine. Iste godine upisuje prediplomski studij računarstva na Elektrotehničkom fakultetu u Osijeku koji završava u redovnom roku, s temom završnog rada: "Snimanje značajki vožnje radi optimiranja potrošnje" (mentor: prof.dr.sc. Željko Hocenski). Sa statusom redovnog studenta, 2014. godine upisuje diplomski studij na Elektrotehničkom fakultetu, smjer procesno računarstvo. Tijekom diplomskog studija odlazi na stručnu praksu u Indiju, pod mentorstvom profesora P.C.Siddalingaswamy, s mjestom rada na Tehnološkom institutu u Manipalu (MIT-Manipal Institute of Tehnology) na odjelu računalne znanosti. Izvrsno se služi engleskim jezikom, a mogućnost usavršavanja stekao je radom u Europskom parlamentu u Bruxellesu u veljači 2015., i ožujku 2016. U siječnju 2014. izabran za člana Stegovnog suda za studente na mandat od dvije godine. Od 2010. aktivni je član počasne sokolske garde "Hrvatski sokol Osječko-baranjske županije".

Filip Sušac

PRILOZI

Kompletan izvorni kod Visual Basic i Arduino aplikacije, kao i svi dodatni materijali (elektronička shema, skice, datasheetovi, itd.) nalaze se na DVD-u priloženom uz ovaj rad. U nastavku slijede dijelovi koda.

PRILOG A. PROGRAM ZA ARDUINO NANO MAKETU

```
#define SERIAL_MONITOR
#define  ON_OFF_DELAY   20
#define  READ_DELAY     20

const int OutputPin[] = {2, 3, 4, 5, 6, 7, 8, 9};
const int InputPin[] = {13, 12, A0, A1, A2, A3, A4, A5};

char b[8][8];

void setup()
{
  for (int i = 0; i < 8; i++)
  {
    pinMode(InputPin[i], INPUT_PULLUP);
    pinMode(OutputPin[i], OUTPUT);
    digitalWrite(OutputPin[i], HIGH);
  }
  Serial.begin(115200);
}

void loop()
{
  while(Serial.available())
  {
    char c = Serial.read();
    if(c != '2')
    {
      break;
    }


    for (int i = 0; i < 8; i++)
    {
      for (int j = 0; j < 8; j++)
      {
        b[i][j] = '0';
      }
    }

    for (int j = 0; j < 8; j++)
    {
      digitalWrite(OutputPin[j], LOW);
      delay(READ_DELAY);
```

```
      for (int i = 0; i < 8; i++)
      {
        if (digitalRead(InputPin[i]) == 0)
        {
          b[i][j] = '1';
        }
      }
      digitalWrite(OutputPin[j], HIGH);
      delay(ON_OFF_DELAY);
    }

      for (int j = 7; j > -1; j--)
    {
      for (int i = 0; i < 8; i++)
      {
        Serial.print(b[i][j]);
      }
    }
    Serial.println();
  }
}
```

PRILOG B. PROGRAM ZAVISUAL BASIC GRAFIČKO SUČELJE

```
Imports System.IO.Ports
Imports System.Drawing
Imports System.Drawing.Drawing2D
Imports System.Windows.Forms
Imports System.Math

Public Class Form1


    Dim W2 As Integer
    Dim VelicinaPolja As Integer = 50
    Dim PomakPocetkaPloce As Point = New Point(VelicinaPolja, VelicinaPolja)
    Dim TrenutnaPozicija(7, 7) As Char
    Dim TrenutniIndeksi(7, 7) As Integer
    Dim PrethodniIndeksi(7, 7) As Integer
    Dim figuraGore As Char


    Dim readThread As New System.Threading.Thread(AddressOf PortRead)
    Dim COM_port_name As String = "COM4"
    Dim p As SerialPort
    Dim TimerInterval_ms As Integer = 1000
    Dim Message As String


    Dim BijeloPolje As Image = Image.FromFile("Figure/1.png")
    Dim CrnoPolje As Image = Image.FromFile("Figure/2.png")
    Dim OznacenoPolje As Image = Image.FromFile("Figure/3.png")
    Dim P1 As Image = Image.FromFile("Figure/P1.png")
    Dim P2 As Image = Image.FromFile("Figure/P2.png")
    Dim D1 As Image = Image.FromFile("Figure/D1.png")
    Dim D2 As Image = Image.FromFile("Figure/D2.png")
    Dim K1 As Image = Image.FromFile("Figure/K1.png")
    Dim K2 As Image = Image.FromFile("Figure/K2.png")
    Dim T1 As Image = Image.FromFile("Figure/T1.png")
    Dim T2 As Image = Image.FromFile("Figure/T2.png")
    Dim S1 As Image = Image.FromFile("Figure/S1.png")
    Dim S2 As Image = Image.FromFile("Figure/S2.png")
    Dim L1 As Image = Image.FromFile("Figure/L1.png")
    Dim L2 As Image = Image.FromFile("Figure/L2.png")


    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load

        Me.WindowState = FormWindowState.Maximized
        Me.BackColor = Color.Azure

        W2 = Me.Width / 2

        p = New SerialPort(COM_port_name, 115200, Parity.None, 8, 1)
        p.WriteTimeout = 500
        p.ReadTimeout = 500
        p.Open()

        PomakPocetkaPloce.X = W2 - 4 * VelicinaPolja
        ListBox1.Left = W2 - ListBox1.Width / 2
        ListBox2.Left = W2 - ListBox2.Width / 2
        Button1.Left = W2 - Button1.Width / 2
```

```
        ListBox1.Items.Clear()
        ListBox2.Items.Clear()

        Message = ""

        TrenutniIndeksi = PocetniIndeksi()
        PrethodniIndeksi = TrenutniIndeksi

        readThread.Start()

        Timer1.Interval = TimerInterval_ms
        Timer1.Enabled = True
    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        readThread.Abort()
        p.Close()
        Application.Restart()
    End Sub


    Private Sub Form1_Leave(sender As Object, e As EventArgs) Handles MyBase.Leave
        readThread.Abort()
        p.Close()
    End Sub


    Private Sub Form1_KeyDown(sender As Object, e As KeyEventArgs) Handles
MyBase.KeyDown, MyBase.KeyUp
        If e.KeyCode = Keys.Escape Then
            Me.Close()
        End If
    End Sub
    Protected Overrides Sub OnPaint(ByVal e As PaintEventArgs)
        TrenutnaPozicija = PocetnaPozicija()
        CrtanjePozicije()
    End Sub


    Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
        PortRead()

        If Message.Length = 65 Then
           ListBox1.Items.Insert(0, PrilagodiZaIspis(Message))
           NovaPozicija()
        End If

    End Sub

    Private Sub PortRead()
        p.WriteLine("2")
        Try
            Message = p.ReadLine()
        Catch generatedExceptionName As TimeoutException
        End Try
    End Sub

    Private Sub NovaPozicija()
        Dim d(7, 7) As Integer

        PrethodniIndeksi = TrenutniIndeksi
        TrenutniIndeksi = IndeksiIzPoruke()
```

```
        For ix = 0 To 7
            For iy = 0 To 7
                d(ix, iy) = TrenutniIndeksi(ix, iy) - PrethodniIndeksi(ix, iy)
            Next
        Next
        ListBox2.Items.Insert(0, PrilagodiZaIspis(d))

        For ix = 0 To 7
            For iy = 0 To 7
                If d(ix, iy) = 0 Then
                    Continue For
                ElseIf d(ix, iy) < 0 Then
                    figuraGore = TrenutnaPozicija(ix, iy)
                    TrenutniIndeksi(ix, iy) = 0
                    TrenutnaPozicija(ix, iy) = "_"
                    CrtajPraznoPolje(ix, iy)
                    CrtajPolje(ix, iy)
                ElseIf d(ix, iy) > 0 Then
                    TrenutnaPozicija(ix, iy) = figuraGore
                    CrtajPraznoPolje(ix, iy)
                    CrtajPolje(ix, iy)
                End If
            Next
        Next
    End Sub
    Private Function PorukaIzIndeksa(indeksi As Integer(,)) As String
        Dim poruka As String
        poruka = ""
        For iy = 7 To 0 Step -1
            For ix = 0 To 7
                poruka = poruka + indeksi(ix, iy).ToString()
            Next
        Next
        Return poruka
    End Function
    Private Function IndeksiIzPoruke() As Integer(,)
        Dim Indeksi(7, 7) As Integer
        Dim i As Integer = 0
        For iy = 7 To 0 Step -1
            For ix = 0 To 7
                Indeksi(ix, iy) = Convert.ToInt32(Message(i)) - 48
                i = i + 1
            Next
        Next
        Return Indeksi
    End Function
    Private Function PocetniIndeksi() As Integer(,)
        Dim Indeksi(7, 7) As Integer
        For ix = 0 To 7
            For iy = 0 To 7
                If iy = 0 Then
                    If (ix = 0 Or ix = 7) Then
                        Indeksi(ix, iy) = 1
                    ElseIf (ix = 1 Or ix = 6) Then
                        Indeksi(ix, iy) = 1
                    ElseIf (ix = 2 Or ix = 5) Then
                        Indeksi(ix, iy) = 1
                    ElseIf (ix = 3) Then
                        Indeksi(ix, iy) = 1
                    ElseIf (ix = 4) Then
                        Indeksi(ix, iy) = 1
```

```
                End If
            ElseIf iy = 1 Then
                Indeksi(ix, iy) = 1
            ElseIf iy = 6 Then
                Indeksi(ix, iy) = 1
            ElseIf iy = 7 Then
                If (ix = 0 Or ix = 7) Then
                    Indeksi(ix, iy) = 1
                ElseIf (ix = 1 Or ix = 6) Then
                    Indeksi(ix, iy) = 1
                ElseIf (ix = 2 Or ix = 5) Then
                    Indeksi(ix, iy) = 1
                ElseIf (ix = 3) Then
                    Indeksi(ix, iy) = 1
                ElseIf (ix = 4) Then
                    Indeksi(ix, iy) = 1
                End If
            Else
                Indeksi(ix, iy) = 0
            End If
        Next
    Next
    Return Indeksi
End Function

Private Function PrilagodiZaIspis(s As String) As String
    Dim ss As String
    ss = ""
    For i = 0 To 63
        ss = ss + " " + s(i)
    Next
    Return ss
End Function
Private Function PrilagodiZaIspis(indeksi As Integer(,)) As String
    Dim ss As String
    Dim x_ As Integer
    ss = ""
    For iy = 7 To 0 Step -1
        For ix = 0 To 7
            x_ = indeksi(ix, iy)
            If x_ < 0 Then
                ss = ss + "-" + Math.Abs(x_).ToString()
            Else
                ss = ss + " " + x_.ToString()
            End If
        Next
    Next
    Return ss
End Function

Private Function PocetnaPozicija() As Char(,)
    Dim pozicija(7, 7) As Char
    For ix = 0 To 7
        For iy = 0 To 7
            If iy = 0 Then
                If (ix = 0 Or ix = 7) Then
                    pozicija(ix, iy) = "t"
                ElseIf (ix = 1 Or ix = 6) Then
                    pozicija(ix, iy) = "s"
                ElseIf (ix = 2 Or ix = 5) Then
                    pozicija(ix, iy) = "l"
                ElseIf (ix = 3) Then
```

```
                    pozicija(ix, iy) = "d"
                ElseIf (ix = 4) Then
                    pozicija(ix, iy) = "k"
                End If
            ElseIf iy = 1 Then
                pozicija(ix, iy) = "p"
            ElseIf iy = 6 Then
                pozicija(ix, iy) = "P"
            ElseIf iy = 7 Then
                If (ix = 0 Or ix = 7) Then
                    pozicija(ix, iy) = "T"
                ElseIf (ix = 1 Or ix = 6) Then
                    pozicija(ix, iy) = "S"
                ElseIf (ix = 2 Or ix = 5) Then
                    pozicija(ix, iy) = "L"
                ElseIf (ix = 3) Then
                    pozicija(ix, iy) = "D"
                ElseIf (ix = 4) Then
                    pozicija(ix, iy) = "K"
                End If
            Else
                pozicija(ix, iy) = "_"
            End If
        Next
    Next
    Return pozicija
End Function

Private Sub CrtanjePozicije()
    Dim g As Graphics = Me.CreateGraphics()
    Dim ix, iy As Integer
    For ix = 0 To 7
        For iy = 0 To 7
            CrtajPraznoPolje(ix, iy)
            CrtajPolje(ix, iy)
        Next
    Next
End Sub
Private Sub CrtajPolje(ix As Integer, iy As Integer)
    Dim x As Integer
    Dim y As Integer
    Dim figura As Char
    x = ix * VelicinaPolja + PomakPocetkaPloce.X
    y = iy * VelicinaPolja + PomakPocetkaPloce.Y
    figura = TrenutnaPozicija(ix, iy)
    Dim g As Graphics = Me.CreateGraphics()
    If (figura = "t") Then
        g.DrawImage(T2, x, y, VelicinaPolja, VelicinaPolja)
    ElseIf (figura = "s") Then
        g.DrawImage(S2, x, y, VelicinaPolja, VelicinaPolja)
    ElseIf (figura = "l") Then
        g.DrawImage(L2, x, y, VelicinaPolja, VelicinaPolja)
    ElseIf (figura = "d") Then
        g.DrawImage(D2, x, y, VelicinaPolja, VelicinaPolja)
    ElseIf (figura = "k") Then
        g.DrawImage(K2, x, y, VelicinaPolja, VelicinaPolja)
    ElseIf (figura = "p") Then
        g.DrawImage(P2, x, y, VelicinaPolja, VelicinaPolja)
    ElseIf (figura = "T") Then
        g.DrawImage(T1, x, y, VelicinaPolja, VelicinaPolja)
    ElseIf (figura = "S") Then
        g.DrawImage(S1, x, y, VelicinaPolja, VelicinaPolja)
```

```
        ElseIf (figura = "L") Then
            g.DrawImage(L1, x, y, VelicinaPolja, VelicinaPolja)
        ElseIf (figura = "D") Then
            g.DrawImage(D1, x, y, VelicinaPolja, VelicinaPolja)
        ElseIf (figura = "K") Then
            g.DrawImage(K1, x, y, VelicinaPolja, VelicinaPolja)
        ElseIf (figura = "P") Then
            g.DrawImage(P1, x, y, VelicinaPolja, VelicinaPolja)
        End If
    End Sub

    Private Sub CrtajPraznoPolje(ix As Integer, iy As Integer)
        Dim x As Integer
        Dim y As Integer
        Dim g As Graphics = Me.CreateGraphics()
        x = ix * VelicinaPolja + PomakPocetkaPloce.X
        y = iy * VelicinaPolja + PomakPocetkaPloce.Y
        If Not (((iy + ix * 7) Mod 2) = 0) Then
            g.DrawImage(CrnoPolje, x, y, VelicinaPolja, VelicinaPolja)
        Else
            g.DrawImage(BijeloPolje, x, y, VelicinaPolja, VelicinaPolja)
        End If
    End Sub

    Private Sub CrtajOznacenoPolje(ix As Integer, iy As Integer)
        Dim x As Integer
        Dim y As Integer
        Dim g As Graphics = Me.CreateGraphics()
        x = ix * VelicinaPolja + PomakPocetkaPloce.X
        y = iy * VelicinaPolja + PomakPocetkaPloce.Y
        g.DrawImage(OznacenoPolje, x, y, VelicinaPolja, VelicinaPolja)
    End Sub
End Class
```