



Innovative Teaching Approaches in development of Software
Designed Instrumentation and its application in real-time
systems

Praktikum iz merno-akvizicionih sistema

Co-funded by the
Erasmus+ Programme
of the European Union



Innovative Teaching Approaches in development of Software Designed Instrumentation and its
application in real-time systems

Faculty of Technical
Sciences



Ss. Cyril and Methodius
University
Faculty of Electrical Engineering
and Information Technologies



Zagreb University of
Applied Sciences



School of Electrical
Engineering
University of Belgrade



Faculty of Physics
Warsaw University of Technology



Co-funded by the
Erasmus+ Programme
of the European Union



The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Lekcija 10 – *Arduino* platforma. Akvizicija analognog napona i korišćenje digitalnih portova

Cilj

Cilj vežbe je da studente:

- upozna sa osnovnim funkcionalnostima *Arduino* platforme, prednostima koje ona pruža, kao i ograničenjima koja postavlja.
- osposobi za samostalno kreiranje osnovnih *Arduino* aplikacija koje uključuju akviziciju analognog napona sa senzora i komunikaciju platforme sa korisnikom korišćenjem računara ili eksternih tastera i displeja.
- uputi u korišćenje otvorenog hardvera, na taj način podižući svest o *Open Source* rešenjima u modernoj nauci i inženjerstvu.

Oprema

- Računar sa instaliranim *Arduino IDE* softverskim okruženjem.
- *Arduino UNO R3* ploča.
- Protobord, kratkospojnici, otpornici (150 Ω , 220 Ω , 10 K Ω , 100 K Ω), NTC termistor, svetleća dioda, TCRT5000 reflektivni optički senzor, kapacitivni senzor dodira, četvorocifreni sedmosegmentni displej sa TM1637 čipom.

NAPOMENA: Najnoviju verziju *Arduino IDE* okruženja, kao i dokumentaciju vezanu za platformu je moguće naći na zvaničnom *Arduino* sajtu:

<https://www.arduino.cc/>

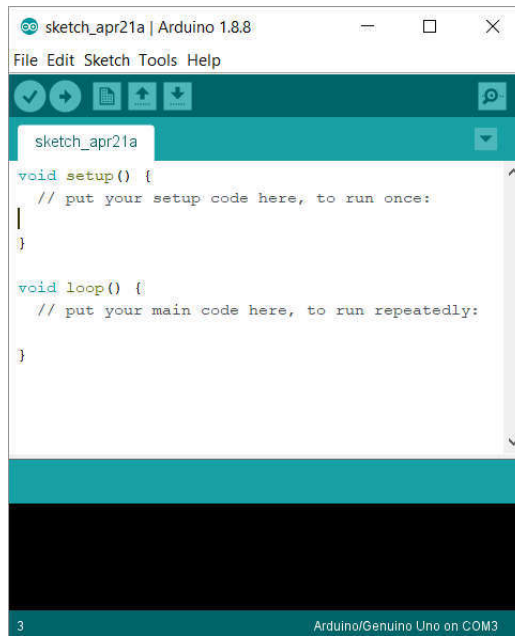
10.1 *Arduino IDE* okruženje i *Arduino UNO R3*



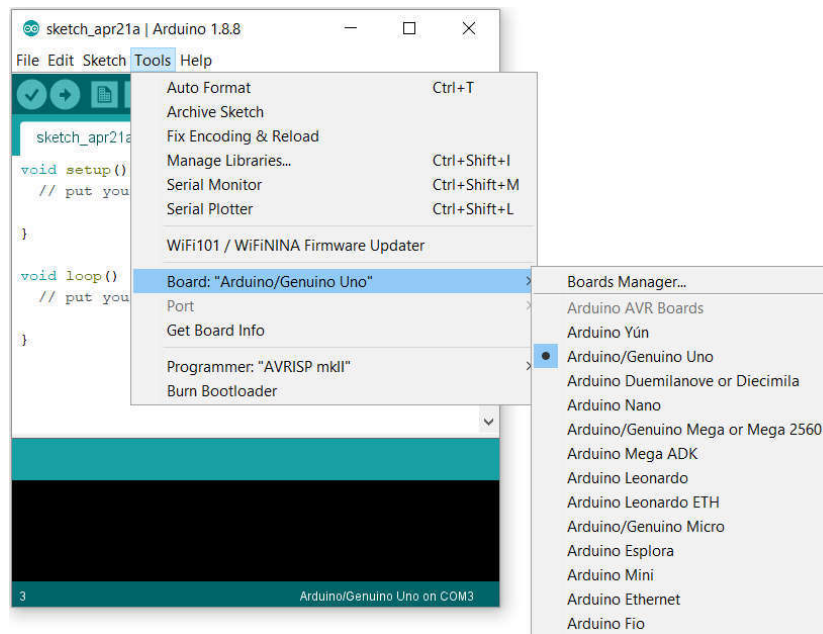
Arduino IDE okruženje se pokreće izborom opcije: **Start»All programs»Arduino** ili klikom na **Arduino** prečicu na *Desktop*-u. Inicijalni *Arduino* prozor je prikazan na Sl. 10.1.1.

NAPOMENA: Uočiti da svaki *Arduino* program sadrži dve celine:

- **setup:** celina u kojoj se upisuje kod za inicijalizaciju;
 - **loop:** glavni deo koda se piše u ovoj beskonačnoj petlji.
1. Povezati *Arduino UNO* sa računarom korišćenjem USB kabla. Kako *UNO* nije jedina ploča iz *Arduino* porodice, potrebno je u programu definisati za koju od dostupnih ploča se piše kod. Izbor ploče je omogućen unutar *Tools* padajućeg menija kao na Sl.10.1.2 gde je potrebno definisati da se kod spušta na *Arduino/Genuino UNO* ploču.

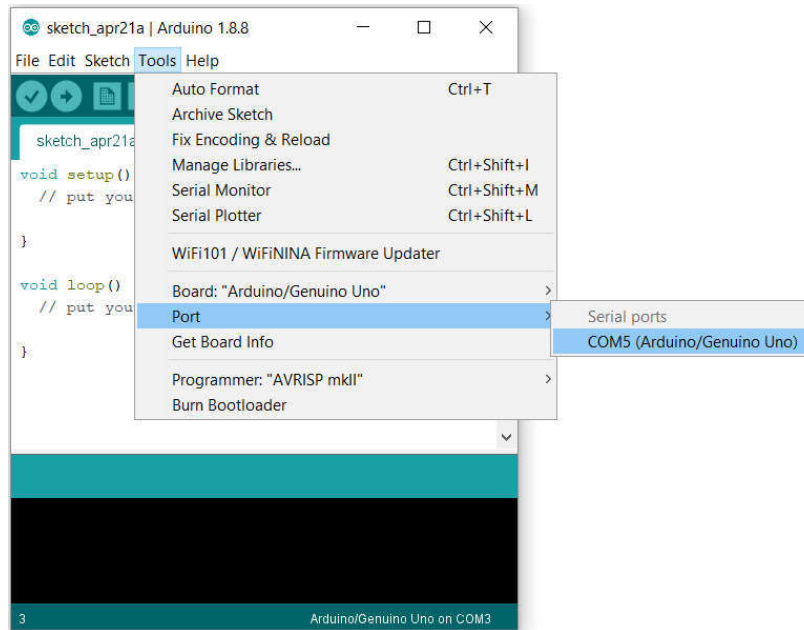


Sl. 10.1.1. Početni ekran *Arudino IDE* okruženja



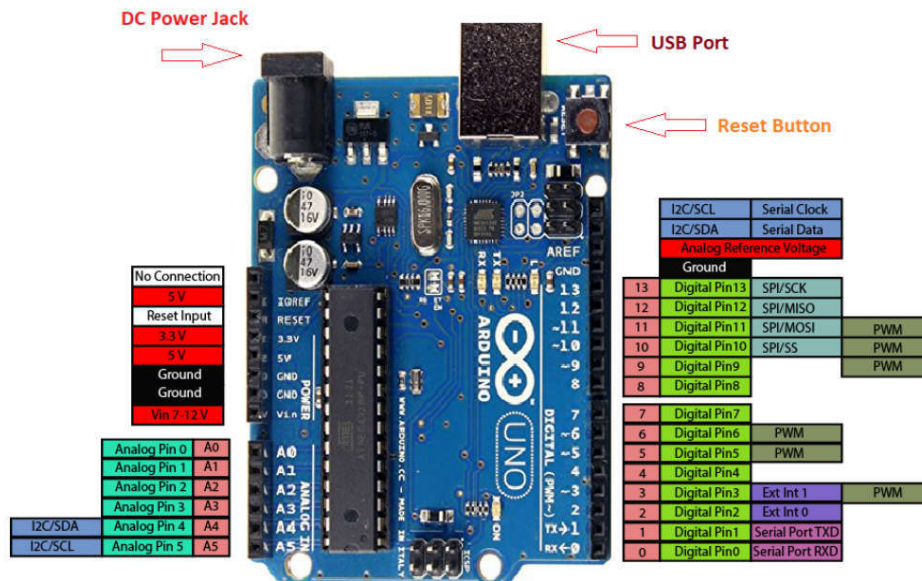
Sl. 10.1.2. Izbor ploče na koju se spušta kod

2. Takođe je potrebno definisati na kom COM portu je priključen *Arduino*. Najpre je potrebno u *Device manager*-u proveriti broj COM porta na koji je *Arduino* priključen, nakon čega je potrebno izvršiti selekciju porta u *Arduino IDE* programu kao na Sl. 10.1.3 čime se završava inicijalno podešavanje programa.



Sl. 10.1.3. Izbor COM porta računara na koji je povezan *Arduino*

NAPOMENA: Pinovi *Arduino UNO* ploče se mogu razvrstati u tri sekcije: POWER, ANALOG IN i DIGITAL prema funkciji koju obavljaju. *Pinout Arduino UNO R3* ploče je prikazan na Sl. 10.1.4.



Sl. 10.1.4. *Pinout* dijagram *Arduino UNO R3* ploče

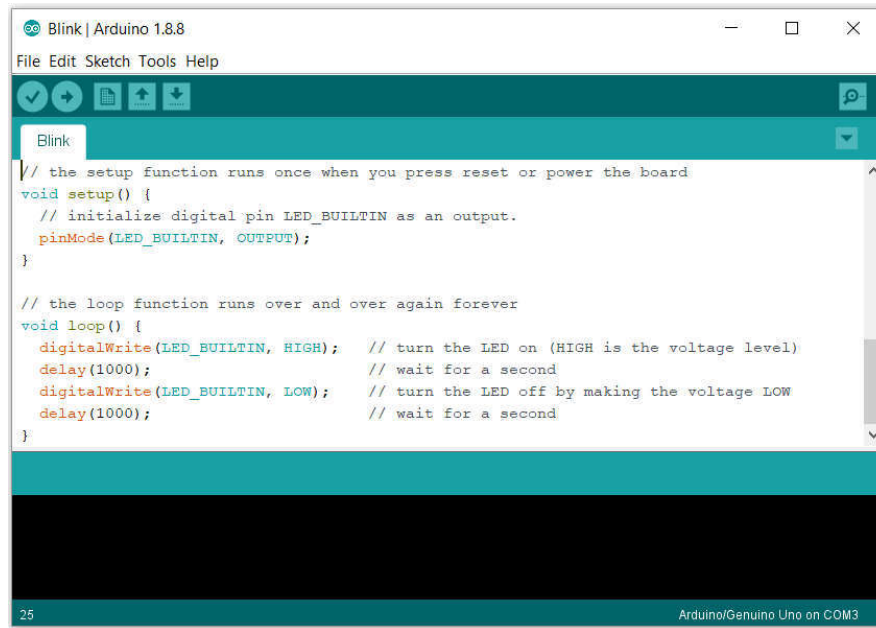
POWER sekciju čine pinovi koji se koriste za napajanje projekata (GND, 5 V i 3.3 V), kao i za dovođenje eksternog napajanja u opsegu 7-12 V (VIN + GND ili DC Power Jack).

ANALOG IN sekciju čine pinovi koji su povezani na A/D konvertor koji se nalazi u samom mikrokontroleru (naziv mikrokontrolera: *Atmel Atmega 328p*) na kojima je omogućeno čitanje analognog napona.

DIGITAL sekciju čine pinovi koji se mogu ponašati kao digitalni ulazi ili izlazi.

10.2 Digitalni ulazi i izlazi (Digital I/O)

1. *Arduino IDE* okruženje dolazi sa velikim brojem ugrađenih primera kojima se pristupa pritiskom na *File»Examples*. Pokrenuti *Blink* primer pritiskom na *File»Examples»01.Basics»Blink*. Otvoriće se prozor sa kodom kao na Sl. 10.2.1. Ovaj primer demonstrira kontrolu svetleće diode koja je ugrađena na ploči i po potrebi se može koristiti kao indikator. Ova svetleća dioda je povezana na digitalni pin D13 *Arduino UNO* ploče koji se može pozvati i korišćenjem definisane konstante `LED_BUILTIN`.



```
Arduino IDE - Blink | Arduino 1.8.8
File Edit Sketch Tools Help
Blink
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}


// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
25 Arduino/Genuino Uno on COM3
```


Sl. 10.2.1. *Blink* primer za kontrolu ugrađene svetleće diode

`pinMode(pin,mode)`: služi za podešavanje digitalnog pina (u ovom primeru je pin podešen kao izlazni - OUTPUT).

`digitalWrite(pin,value)`: služi za ispisivanje visoke (HIGH tj. 5V) ili niske (LOW tj. 0V) vrednosti na digitalnom izlazu.

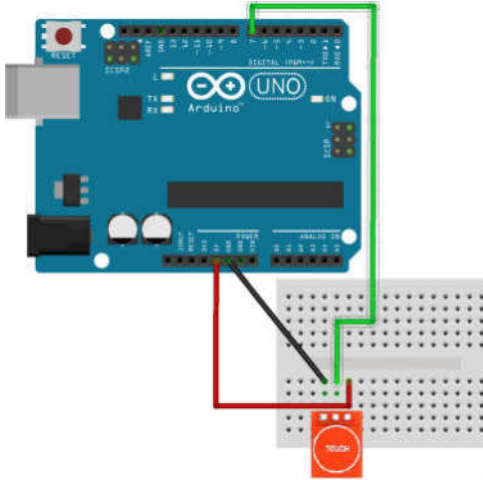
`delay(ms)`: zaustavlja izvršavanje programa za definisani broj milisekundi.

2. Spustiti kod na ploču pritiskom na dugme *Upload* . Posmatrati ploču i uočiti ponašanje svetleće diode obeležene sa L.

NAPOMENA: Nekada je potrebno izvršiti *Compiling* koda bez spuštanja koda na ploču. Ova opcija je omogućena pritiskom na dugme *Verify* .

3. Sačuvati prethodni program pritiskom na *File»Save as* pod nazivom *Blink_test.ino*, a zatim taj, novosačuvani program modifikovati tako svetleća dioda svake sekunde bude uključena 200 ms.
4. **Isključiti** *Arduino* iz računara i povezati kapacitivni senzor dodira korišćenjem kratkospojnika i protoborda kao na Sl.10.2.2. Zadatak je napraviti program koji će detektovati dodir na senzoru i dok je senzor dodirnut držati uključenu ugrađenu svetleću diodu na ploči. **Voditi računa** o

povezivanju napajanja kako se senzor ne bi inverzno polarisao i na taj način spalio.



Sl. 10.2.2. Šema povezivanja kapacitivnog senzora dodira

5. Otvoriti novi *Sketch* pritiskom na *File»New* i sačuvati ga pod nazivom *TouchLED.ino*. U programu najpre definisati dve nove promenljive *int* tipa pod nazivom *touchPin* i *touchVal* pre *setup* sekcije koda. Njihove vrednosti inicijalizovati na 7 i 0 respektivno. Promenljiva *touchPin* označava broj digitalnog pina na koji je povezan senzor dodira, dok će promenljiva *touchVal* služiti za upisivanje pročitane vrednosti na *touchPin* digitalnom pinu. Definisane promenljive: tip naziv = vrednost; (`int touchPin = 7;`)
6. U okviru *setup* sekcije korišćenjem funkcije `pinMode()` inicijalizovati pin `LED_BUILTIN` na kome se nalazi ugrađena svetleća dioda kao izlazni (OUTPUT) i *touchPin* kao ulazni pin (INPUT).
7. Unutar beskonačne petlje *loop* je najpre potrebno pročitati vrednost na digitalnom pinu *touchPin* i upisati je u promenljivu *touchVal*. Nakon toga je potrebno, korišćenjem *if* funkcije proveriti da li je vrednost jednaka 1 (senzor je dodirnut) ili 0 (senzor nije dodirnut). Potrebno je obezbediti da ugrađena svetleća dioda svetli samo ukoliko je senzor dodirnut. Korišćenjem `delay()` funkcije se obezbeđuje da se provera vrednosti pina odvija na **približno** 50 ms. Gotov kod je prikazan na Sl. 10.2.3.

```
int touchPin = 7;
int touchVal = 0;

void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(touchPin, INPUT);
}

void loop() {
  touchVal = digitalRead(touchPin);
  if (touchVal == 1) {
    digitalWrite(LED_BUILTIN, HIGH);
  }
  else {
    digitalWrite(LED_BUILTIN, LOW);
  }
  delay(50);
}
```

Sl. 10.2.3. Program za čitanje sa kapacitivnog senzora dodira i uključivanje svetleće diode

`digitalRead(pin)`: služi za čitanje sa digitalnog ulaza. Može vratiti vrednosti 1 ili 0 u zavisnosti od toga da li je ulazu pročitana visoka ili niska vrednost napona respektivno.

Povezati *Arduino* sa računarom i prebaciti kod na ploču pritiskom na dugme *Upload*. Dodirnuti kapacitivni senzor dodira i posmatrati svetleću diodu L na *Arduino UNO* ploči. Zatim bez dodirivanja približiti prst blizu senzora. Objasniti pojavu koja se uočava!

8. **Isključiti** *Arduino* iz računara i razvezati kolo.

10.3 Analogna akvizicija napona i serijska komunikacija sa računarom

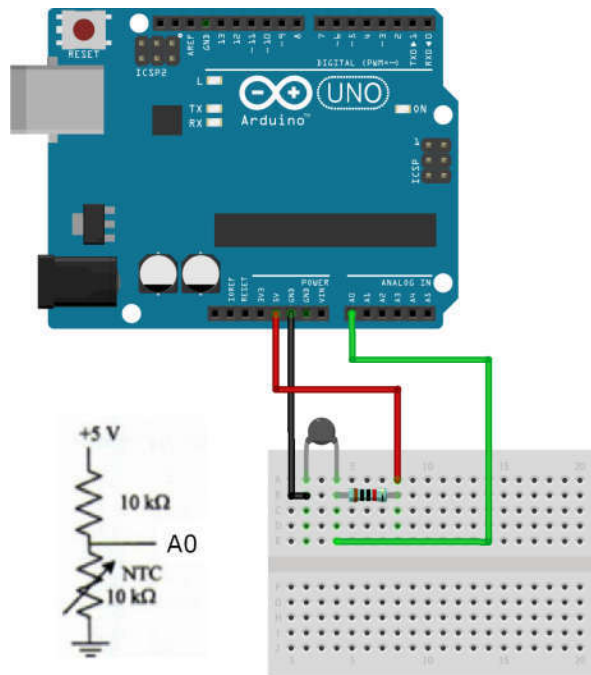
1. Povezati razdelnik napona sa NTC termistorom na protobordu prema šemi sa Sl. 10.3.1.
2. Otvoriti primer *AnalogReadSerial* koji se nalazi u sekciji *File»Examples»01.Basics* i sačuvati ga kao novi *Sketch* pod nazivom *AnalogReadSerial_NTC.ino*.

Serial.begin(): inicijalizuje komunikaciju između *Arduino* ploče i računara (ili nekog drugog uređaja koji je sposoban za serijsku komunikaciju sa *Arduino* pločom). Argument funkcije je tzv. *baud rate* – broj bita po sekundi (najčešće 9600).

analogRead(pin): vraća rezultat A/D konverzije sa preciziranog analognog ulaznog pina. Kako A/D konvertor na *Arduino* UNO R3 ploči ima rezoluciju od 10 bit-a ova funkcija vraća *integer* vrednosti od 0 do 1023 koje predstavljaju nivoe celokupnog ulaznog naponskog opsega. U opštem slučaju, 1024 nivoa se raspoređuju na opseg od 0 do 5V. Kolika je tada rezolucija A/D konverzije? Analogni pin nije potrebno inicijalizovati već je njegovo podrazumevano stanje da je ulazni.

Serial.print(): ispisuje zadatu vrednost na serijskom portu kako bi je računar pročitao. Dodavanjem sufixa **ln** se obezbeđuje i prelazak u novi red nakon poslate vrednosti.

Za one koji žele da znaju više: Opseg napona funkcije **analogRead()** je moguće i smanjiti kako bi se povećala rezolucija u manjem opsegu ulaznog napona korišćenjem funkcije **analogReference(type)**.



Sl. 10.3.1. Povezivanje naponskog razdelnika sa NTC termistorom na *Arduino*

3. Modifikovati kod tako da *delay()* funkcija zaustavlja izvršavanje programa na 10 ms.
4. U petlji dodati i promenljivu voltage tipa *float* u kojoj će se upisivati vrednost sa A/D konvertora saklirana na opseg od 0 do 5V na taj način obezbeđujući ispisivanje realnih vrednosti napona, a ne naponskih nivoa od 0 do 1023.

5. Umesto ispisivanja promenljive `sensorValue`, za argument funkcije `Serial.println()` uzeti promenljivu `voltage`. Konačan izgled programskog koda je prikazan na Sl. 10.3.2.

```
// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

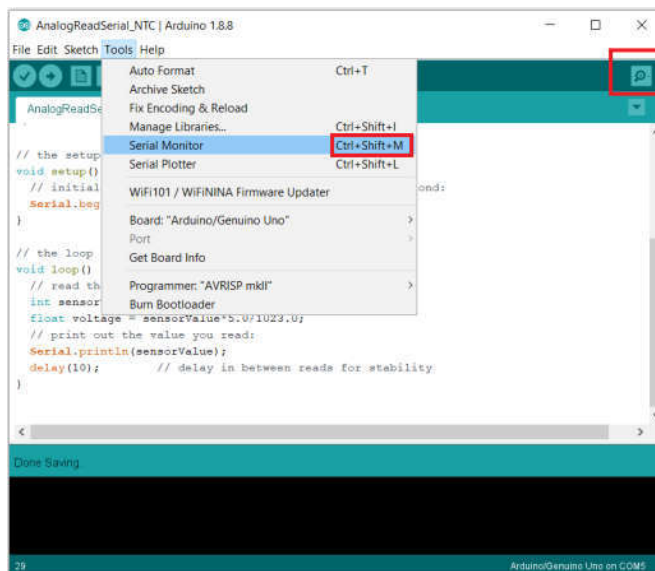
// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  float voltage = sensorValue*5.0/1023.0;
  // print out the value you read:
  Serial.println(voltage);
  delay(10);      // delay in between reads for stability
}
```

Sl. 10.3.2. Kod za analognu akviziciju i ispisivanje vrednosti na serijskom portu

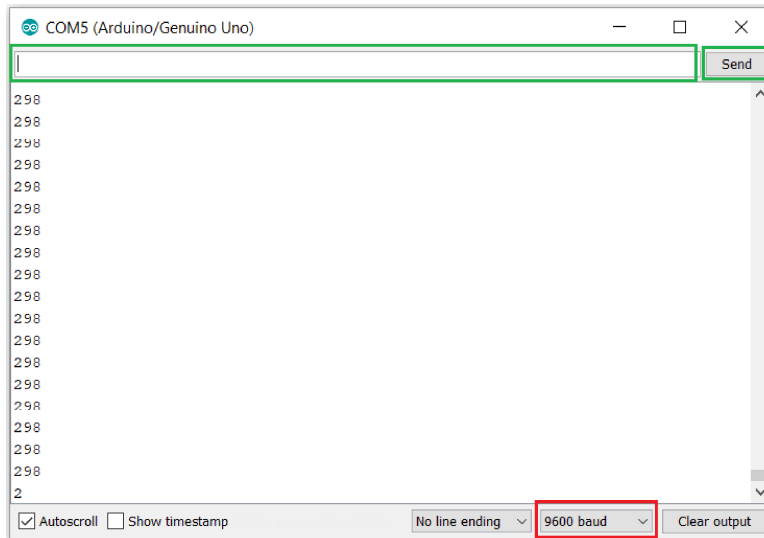
6. Sačuvati promene u programu.
7. Povezati *Arduino* sa računarom i spustiti kod na ploču. Na ploči obratiti pažnju na svetleće diode obeležene sa RX i TX koje signaliziraju serijsku komunikaciju *Arduino* ploče sa računarom. Objasniti njihovo trenutno stanje!

NAPOMENA: Različiti programski paketi kao što su *LabVIEW*, *Python* ili *MATLAB* poseduju set funkcija koje omogućavaju računaru da prikuplja nadolazeće podatke na serijskom portu i prikazuje ih. *Arduino IDE* takođe omogućava ispisivanje podataka poslanih sa *Arduino* ploče korišćenjem *Serial Monitor* opcije.

8. U prozoru *Arduino IDE* okruženja pokrenuti *Serial Monitor* izborom *Tools»Serial Monitor*, prečicom `Ctrl+Shift+M` ili pritiskom na ikonicu u gornjem desnom uglu prozora, kao na Sl. 10.3.3. Otvoriće se prozor u kome će se ispisivati vrednosti napona koje se šalju sa *Arduino* ploče kao na Sl. 10.3.4.



Sl. 10.3.3. Uključivanje *Serial Monitor* prozora



Sl. 10.3.4. Izgled Serial Monitor prozora. Opcije za slanje poruka su uokvirene zelenim pravougaonikom. *Baud rate* podešavanje je uokvireno crvenim pravougaonikom.

U ovom prozoru je takođe moguće slati podatke *Arduino* ploči korišćenjem *text box*-a u vrhu prozora i pritiskom na dugme *Send*. Kako trenutni kod koji se nalazi na ploči ne vrši čitanje na serijskom portu, slanje poruka ploči nije od interesa za rad programa. Ipak, interesantno je poslati poruku proizvoljne sadržine *Arduino* ploči i tokom slanja posmatrati RX i TX svetleće diode. Šta je sada moguće uočiti?

NAPOMENA: Obratiti pažnju na *Baud rate* podešavanje u okviru *Serial Monitor* prozora. Kako bi poruke bile protumačene ispravno, izabrana vrednost se mora poklopiti sa vrednošću koja je pozvana kao argument funkcije *Serial.begin()*.

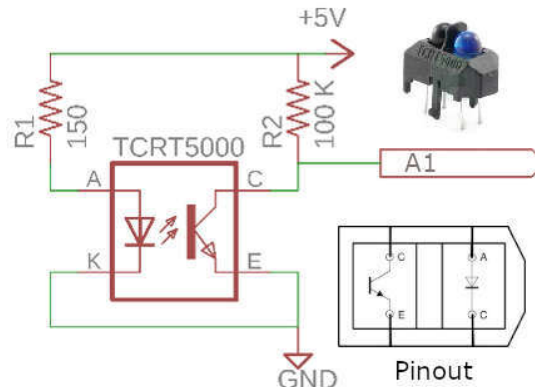
9. Promeniti vrednost za *Baud rate* u prozoru *Serial Monitor*-a i uočiti šta se dešava sa sadržajem poruke. Vratiti vrednost na 9600.

Često se *Arduino* koristi kao uređaj za akviziciju naponskih signala te je od interesa iscertati te signale u realnom vremenu. U okviru *Arduino IDE* okruženja je iscertavanje vrednosti koje *Arduino* šalje na serijski port omogućeno korišćenjem *Serial Plotter* opcije.

10. Zatvoriti *Serial Monitor* prozor. Otvoriti *Serial Plotter* izborom *Tools»Serial Plotter* ili prečicom *Ctrl+Shift+L*. Obratiti pažnju da *Serial Monitor* i *Serial Plotter* ne mogu biti otvoreni u isto vreme!
11. Posmatrati promenu dobijenog signala na ekranu sa promenom temperature NTC termistora.
12. **Isključiti *Arduino* iz računara. Kolo ostaviti sastavljeno!**

10.4 Analogni akvizicija i prikazivanje signala sa više senzora

1. Na protobordu **sastaviti** šemu sa Sl. 10.4.1.



Sl. 10.4.1. Šema za povezivanje TCRT5000 reflektivnog optičkog senzora

TCRT5000 je jednostavan primer reflektivnog optičkog senzora rastojanja. Čine ga svetleća dioda koja emituje i fototranzistor koji detektuje svetlost u IC delu spektra (svetlost se ne vidi golim okom). Ukoliko se prepreka nalazi ispred senzora, deo svetlosti koji se emituje sa svetleće diode se reflektuje nazad ka fototranzistoru gde biva detektovan. Što je prepreka bliža, više svetlosti se reflektuje ka fototranzistoru, i njegova struja raste!

2. Kolektor fototranzistora povezati na A1 analogni ulaz *Arduino* ploče.
3. Modifikovati program iz prethodne tačke ponavljanjem odgovarajućih funkcija kako bi se obezbedilo čitanje dva napona sa analognih ulaza A0 i A1. Vrednosti sa analognih ulaza ispisivati u istom redu, razdvojene *TAB* konstantnom.

Modifikovani program je prikazan na Sl. 10.4.2.

```
// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  float voltage = sensorValue*5.0/1023.0;

  // read the input on analog pin 1:
  int sensorValue1 = analogRead(A1);
  float voltage1 = sensorValue1*5.0/1023.0;

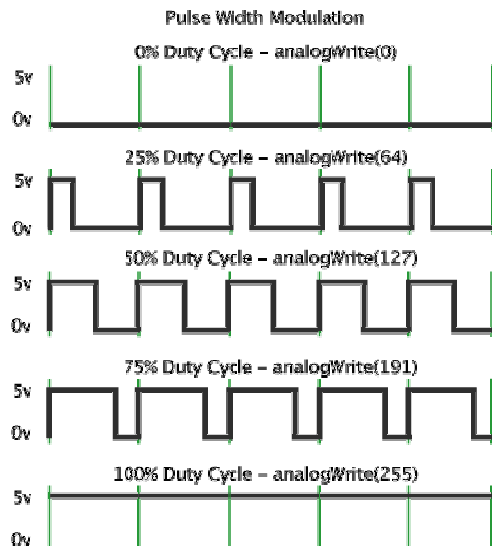
  // print out the values you read:
  Serial.print(voltage);
  Serial.print(" ");
  Serial.println(voltage1);
  delay(10);      // delay in between reads for stability
}
```

Sl. 10.4.2. Kod koji omogućava iscrtavanje vrednosti sa dva analogni ulaza istovremeno

4. Sačuvati ovaj kod pod novim imenom *AnalogRead_NTC_TCRT5000.ino*
5. Povezati *Arduino* sa računarom i spustiti kod na ploču. Pokrenuti *Serial Monitor* kako bi se uočio format u kome se šalju podaci.
6. Zatvoriti *Serial Monitor* i pokrenuti *Serial Plotter*. Postaviti neki vid prepreke i ispred TCRT5000 senzora i posmatrati kako se dobijeni signal menja sa promenom rastojanja prepreke od senzora. Odgovoriti na sledeća pitanja:
 - Koliki je opseg rastojanja koje ovaj senzor može detektovati?
 - Ukoliko se uočava izražen uticaj šuma kada nema predmeta ispred senzora objasniti njegovo poreklo!
 - Ukoliko bi se izvršila kalibracija ovog senzora u trenutnom okruženju, da li bi se isti rezultati mogli očekivati i ukoliko se senzor iznese napolje?
7. **Isključiti** *Arduino* iz računara. Rastaviti kolo.

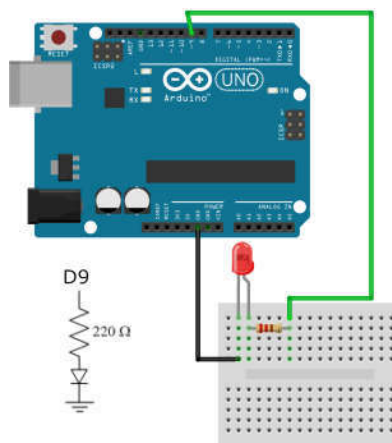
10.5 Impulsno širinska modulacija

Za razliku od *Arduino DUE* ploče, *UNO* ne poseduje D/A konvertor te nije moguće ispisivati analogne vrednosti napona na nekom od njegovih pinova. Međutim, *Arduino UNO* poseduje mogućnost generisanja Impulsno Širinski Modulisanih pulseva (*Pulse Width Modulation - PWM*) na pinovima koji pored svog broja poseduju oznaku \sim . Impulsno širinski modulisani pulsevi predstavljaju povorku četvrtki definisane frekvencije kod kojih je moguće menjati procenat vremena jedne periode tokom koga izlazni napon ima visoku vrednost. Ovaj procenat se naziva faktor ispune (*Duty Cycle - DC*). Impulsno širinska modulacija je neretko dovoljna i poželjna za kontrolu određenih uređaja kao što su neki tipovi motora, promenu intenziteta svetlosti svetleće diode itd. Izgled PWM signala sa različitim faktorom ispune je prikazan na Sl. 10.5.1.



Sl. 10.5.1. Ilustracija impulsno širinske modulacije

1. U *Arduino IDE* okruženju otvoriti primer *Fading* izborom *File»Examples»03.Analog»Fading*.
2. Sačuvati program pod novim imenom *Fading_test.ino*.
3. Povezati kolo prema Sl. 10.5.2 Kod iz primera je prikazan na Sl. 10.5.3



Sl. 10.5.2. Povezivanje eksterne svetleće diode na pin 9 *Arduino UNO* ploče

4. Povezati *Arduino* sa računarem i spustiti kod na ploču. Uočiti šta se dešava sa svetlećom diodom. Objasniti zašto iako je svetleća dioda uvek u samo dva stanja, uključenom i isključenom, naše oko registruje kontinualnu promenu intenziteta svetlosti od minimalnog do maksimalnog!
5. **Isključiti** *Arduino* iz računara.

```
void setup() {
  // nothing happens in setup
}

void loop() {
  // fade in from min to max in increments of 5 points:
  for (int fadeValue = 0 ; fadeValue <= 255; fadeValue += 5) {
    // sets the value (range from 0 to 255):
    analogWrite(ledPin, fadeValue);
    // wait for 30 milliseconds to see the dimming effect
    delay(30);
  }

  // fade out from max to min in increments of 5 points:
  for (int fadeValue = 255 ; fadeValue >= 0; fadeValue -= 5) {
    // sets the value (range from 0 to 255):
    analogWrite(ledPin, fadeValue);
    // wait for 30 milliseconds to see the dimming effect
    delay(30);
  }
}
```

Sl. 10.5.3. Kod iz primera *Fading* dostupnog u *Arduino IDE* okruženju

`analogWrite(pin, value)`: omogućava generisanje PWM signala na odgovarajućim pinovima obeleženim sa ~ na ploči. Vrednost koju ova funkcija prima kao drugi argument može biti u opsegu od 0 do 255 (8 bita je na raspolaganju) gde vrednosti iz ovog opsega odgovaraju ovrednostima od 0 do 100% faktora ispunje. Ukoliko se za PWM pošalje vrednost 0, vrednost napona će biti uvek 0V, dok u slučaju slanja vrednosti 255 napon kontinualno zadržava vrednost 5V. Kako *Arduino DUE* ploča poseduje i D/A konvertor povezan na pinove DAC0 i DAC1, funkcija `analogWrite()` za ove pinove na *DUE* ploči omogućava generisanje pravog analognog napona.

Za razmišljanje: Ukoliko je ipak neophodno dobiti analogni naponski izlaz, a na raspolaganju je *Arduino UNO* ploča, razmisliti koji jednostavan elektronski sklop bi mogao poslužiti kao konvertor PWM signala u jednosmerni napon!

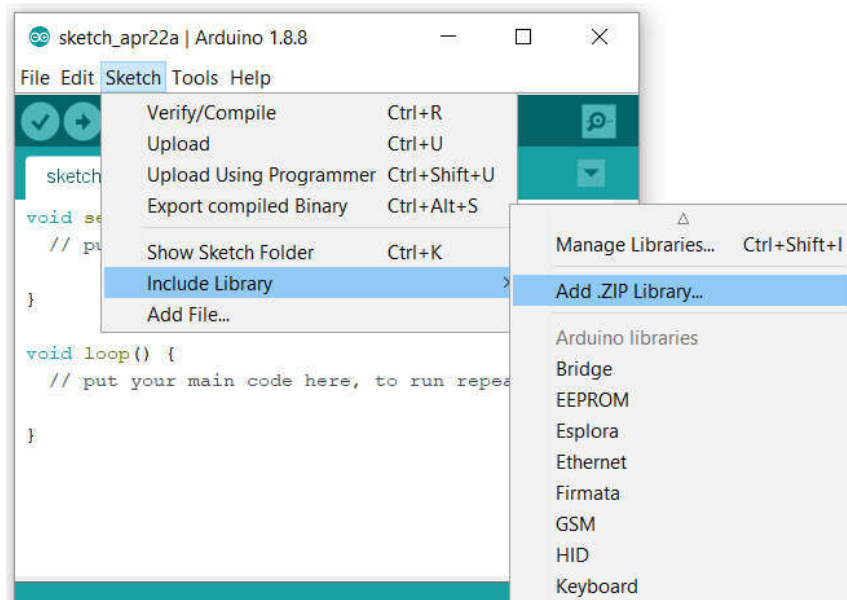
Zadatak 10.5.1. – za samostalni rad

Korišćenjem znanja iz prethodnih poglavlja realizovati *Arduino* program koji u zavisnosti od blizine prepreke TCRT5000 senzoru menja osvetljaj eksterne crvene svetleće diode. TCRT5000 senzor i crvenu svetleću diodu povezati u kolo u skladu sa Sl. 10.4.1 i Sl. 10.5.2. Obezbediti da se provera blizine prepreke izvršava na približno 50 ms.

10.6 Primena biblioteka i prikazivanje vrednosti na displeju

U praktičnim primenama, često je neophodno obezbediti komunikaciju između *Arduino* ploče i korisnika, tj. slanje poruka korisniku, bez posredstva računara. Vizualne informacije složenije prirode, kao što je pročitana vrednost napona na analognom ulazu, nije moguće na jednostavan način predložiti korisniku korišćenjem jednostavnih indikatora poput proste svetleće diode. Rešenje ovog problema se postiže korišćenjem displeja. Jedan od najčešćih tipova displeja danas je tzv. sedmosegmentni displej – displej sa sedam segmenata koji mogu biti uključivani u različitim kombinacijama kako bi prikazivali brojeve od 0 do 9, kao i određena slova. U zavisnosti od realizacije, pored osnovnih sedam segmenata se mogu naći i displeji koji poseduju tačku u donjem desnom uglu.

1. Da bi se omogućila kontrola četvorocifrenog sedmosegmentnog displeja sa *Arduino UNO* ploče, neophodno je najpre izvršiti ubacivanje, tj. *import* eksterne biblioteke u *IDE*. Biblioteka za kontrolu displeja je preuzeta sa linka: <https://github.com/avishorp/TM1637> u formi .zip datoteke. Import biblioteke u *Arduino IDE* okruženju je moguć izborom opcije *Sketch»Include Library»Add .ZIP Library...* kao na Sl. 10.6.1. Pored nove biblioteke, *Arduino IDE* okruženje dolazi sa velikim brojem već dostupnih biblioteka o kojima se više informacija može naći na njihovoj zvaničnoj *web* stranici.



Sl. 10.6.1. Postupak *Import*-a biblioteke u *Arduino IDE* okruženje

2. Nakon uspešnog *Import*-ovanja biblioteke u okruženje, napisati program prema Sl. 10.6.2. Ovaj program demonstrira korišćenje *TM1637Display* biblioteke na jednostavnom primeru u kome se unutar *for* petlje brojač *NumStep* inkrementira na svakih 500 ms, nakon što se njegova vrednost ispiše na displeju.

```

#include <TM1637Display.h>

const int CLK = 9; //Set the CLK pin connection to the display
const int DIO = 8; //Set the DIO pin connection to the display

int NumStep = 0; //Variable to interate

TM1637Display display(CLK, DIO); //set up the 4-Digit Display.

void setup()
{
    display.setBrightness(7); //set the diplay to maximum brightness
}

void loop()
{
    for(NumStep = 0; NumStep < 9999; NumStep++) //Interrate NumStep
    {
        display.showNumberDec(NumStep); //Display the Variable value;
        delay(500); //A half second delay between steps.
    }
}

```

Sl. 10.6.2. Kod za kontrolu četvorocifrenog sedmosegmentnog displeja

`#include <TM1637Display.h>` : obezbeđuje uključivanje biblioteke u program.

`TM1637Display display(CLK, DIO)`: inicijalizacija displeja.

`display.setBrightness()`: podešava intenzitet svetlosti kojim svetli displej. Argument funkcije može uzeti vrednosti od 0 do 7 gde je sa 0 označen najslabiji intenzitet, dok u slučaju argumenta 7 displej najintenzivnije svetli.

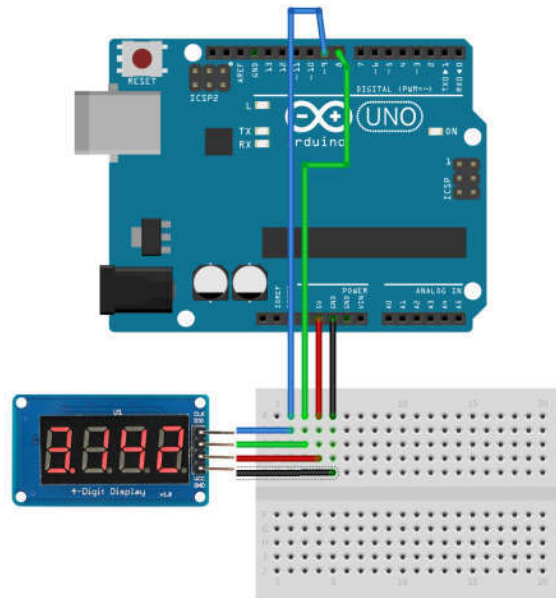
`display.showNumberDec()`: prikazuje argument funkcije na displeju.

Sačuvati kod pod nazivom *tm1637Display.ino*. Povezati samo *Arduino* sa računarom i spustiti kod na ploču. **Isključiti** *Arduino* iz računara.

3. Povezati modul sa četvorocifrenim sedmosegmentnim displejem sa **isključenom** *Arduino* pločom prema Sl. 10.6.3.
4. Ponovo povezati *Arduino* sa računarom i uočiti promenu brojeva na displeju.
5. Promeniti argument funkcije `display.setBrightness()` na 0 i spustiti kod na ploču. Uočiti razliku u intenzitetu svetlosti sa displeja.
6. Isključiti *Arduino* iz računara.

Zadatak 10.6.1 – za samostalni rad

Realizovati program koji će svake sekunde na četvorocifrenom sedmosegmentnom displeju prikazivati vrednost napona termistora povezanog u naponski razdelnik prema Sl. 10.3.1. Prikazani napon je potrebno da bude izražen u mV.



Sl. 10.6.3. Šema prema kojoj je potrebno povezati displej sa *Arduino* pločom