

Lab. vežba br. 1: Uvod u Arduino I deo

Cilj vežbe

Cilj vežbe je da se studenti i studentkinje upoznaju sa Arduino slobodnim softverom tj. razvojnim okruženjem (eng. *Integrated Development Environment* IDE, <https://www.arduino.cc/en/Main/Software>, pristupljeno 8. novembra 2020) i sa UNO R3 hardverom. Osnovni principu koje bi studenti/kinje trebalo da savladaju su upravljanje digitalnim izlazima i analognim ulazima, ali i da realizuju relativno jednostavnu metodu za merenje nepoznate otpornosti.

Oprema

Na raspolaganju su UNO R3 mikrokontrolerska pločica sa USB kablom za povezivanje sa računarom (tip A na tip B), linearni potenciometar (10 k Ω), protobord, kratkospojnice, dve LE diode (zeleno i crveno ili neke druge boje koje su na raspolaganju), otpornici raznih otpornosti.

Zadaci

Potrebno je testirati rad *Blink.ino* i *AnalogReadSerial.ino* primere iz Arduino softvera i upoznati se sa Arduino okruženjem. Takođe, realizovati jednostavnu metodu za merenje otpornosti. U obrazac koji je posebno namenjen za ovu laboratorijsku vežbu, potrebno je uneti rezultate merenja i računanja. Takođe, osim obrasca, dežurni/a će pregledati i kod u Arduino okruženju.

Zadatak #1: Otvoriti Arduino razvojno okruženje. Iz padajućeg menija odabrati *File/Examples/01.Basics/Blink* i otvoriće se primer *Blink*. Snimiti ovaj primer pod novim nazivom *Primer1.ino*. Primetiti da je *.ino* ekstenzija Arduino programa i da je potrebno, da bi se pravilno pokrenuo ili sačuvao neki program, da bude smešten u folderu sa istim imenom kao ime fajla. Prilikom snimanja programa pod novim imenom ekstenzija se podrazumeva pa je nije neophodno eksplicitno navoditi u imenu fajla, jer će, u tom slučaju, biti deo imena programa. Arduino programi se nazivaju skice tj. na eng. *sketches*. Testirati rad *Blink.ino* programa pomoću *Verify* opcije za kompajliranje koda i proveriti da li ima grešaka u kodu. Potom, proučiti strukturu koda:

- 1) U funkciji "setup()" se vrši inicijalizacija i ova funkcija se izvršava samo jednom u kodu¹, definiše se sa ključnom reči "void" i oznaka za početak i kraj funkcije su vitičaste zagrade "{}".
- 2) Kako digitalni pinovi mogu biti definisani kao ulazi i kao izlazi² potrebno je definisati da li se radi o ulaznom ili izlaznom pinu i za to služi funkcija "pinMode()". Pogledati uputstvo za korišćenje ove funkcije na: <https://www.arduino.cc/en/Reference/PinMode> (pristupljeno 8. novembra 2020). Primetiti da je prvi argument ove funkcije broj pina koji se koristi kao digitalni ulaz ili izlaz, a drugi argument definiše da li će se pin koristiti kao ulazni ili izlazni. Umesto broja pina može se koristiti i reč "LED_BUILTIN" za ugrađeni LED indikator na pinu 13.
- 3) Funkcija "loop()" tj. beskonačna petlja počinje, kao i funkcija "setup()", ključnom reči "void" i za početak i kraj se koriste vitičaste zagrade "{}".

¹ Redosled izvršavanja programskog Arduino koda je sekvencijalan tj. linija po linija koda.

² Pored INPUT i OUTPUT, postoji i INPUT_PULLUP tip ulaza, a o tome će biti više reči kasnije na predmetu.

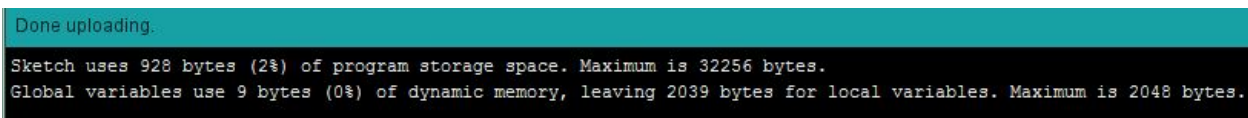
- 4) Unutar beskonačne petlje se nalaze dve funkcije "digitalWrite()" i "delay()". Uputstva i pomoć (eng. *help*) za ove dve funkcije, studenti/kinje mogu pogledati na sledećim internet stranicama: <https://www.arduino.cc/en/Reference/DigitalWrite> (pristupljeno 8. novembra 2020) i <https://www.arduino.cc/en/Reference/Delay> (pristupljeno 8. novembra 2020). Primititi da je prvi argument "digitalWrite()" funkcije broj pina na kome se upisuje određena vrednost, a da je drugi argument napon koji se dovodi na digitalni izlaz ("HIGH" i "LOW", odnosno 5 V i 0 V, respektivno).

Povezati UNO R3 sa računarnom, a potom odabrati tip mikrokontrolerske pločice i COM port na kome je povezana UNO R3 mikrokontrolerska pločica prema Uputstvu za Arduino instalaciju, Zenodo DOI [10.5281/zenodo.3534376](https://doi.org/10.5281/zenodo.3534376), pristupljeno 8. novembra 2020.

Pritiskom na dugme *Upload*, pokrenuti *upload*-ovanje programa u mikrokontrolersku pločicu. Proveriti da li program radi tj. da li se ugrađena LED pali i gasi u ritmu koji je zadat programom. Ako je sve u redu, dobiće se kratka poruka *Done uploading.* kao na Sl. 1.

Primititi donji prozor Arduino programskog okruženja u kome je nakon *upload*-ovanja ispisana poruka kao na Sl. 1. U trenutku kada se kod *upload*-uje u memoriju UNO R3 pločice, program zauzima određeni memorijski prostor i korisnik/ca dobija informaciju o tome. Na Sl. 1 to je 2% memorijskog prostora.

Kada je program jednom *upload*-ovan u memoriju računara, može se priključiti preko USB porta na bilo koji računar (koji čak i nema Arduino sofver instaliran). Konekcija preko USB porta se, u tom slučaju, koristi samo za napajanje. Da bi se program mikrokontrolera "vratio" na početne uslove *upload*-ovanog programa, potrebno je pritisnuti prekidač "reset" na UNO R3 pločici. Da li bi se pritiskom ovog tastera nešto promenilo u izvršavanju *Blink.ino*?



```
Done uploading.
Sketch uses 928 bytes (2%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum is 2048 bytes.
```

Slika 1, Izveštaj nakon *upload*-ovanja *Blink.ino* primera u UNO R3 mikrokontrolersku pločicu.

Promeniti kašnjenja (unutar "delay()" funkcija) i posmatrati šta se tada dešava sa ugrađenom LED. U "setup()" delu upisati umesto "LED_BUILTIN" broj 13 i proveriti da li kod i dalje radi.

Neposredno pre poziva i definisanja "setup()" funkcije moguće je definisati globalne promenljive. Definirati celobrojnu (eng. *integer*) promenljivu pod nazivom "ledPin" pomoću ključne reči "int" (<https://www.arduino.cc/en/Reference/Int>, pristupljeno 8. novembra 2020) i zameniti u "digitalWrite()" i u "pinMode()" funkcijama naziv izlaznog digitalnog pina. Testirati rad programa. Primititi da je prilikom izvršavanja *upload*-ovanog programa LED sa nazivom "ON" uvek uključena na UNO R3 pločici. Dodatno, primititi da su "HIGH" i "LOW" tipa *Boolean* i da ih je moguće zameniti numeričkim vrednostima, gde *Boolean* promenljivoj "LOW" odgovara numerička 0, a *Boolean* promenljivoj "HIGH" bilo koja vrednost koja je različita od 0.

Sad je potrebno promeniti kod tako da se umesto digitalnog pina 13 koristi digitalni pin 8 i preko otpornika od 220 Ω dovesti jednu crvenu LED na digitalni pin 8. Otpornost otpornika koji su

studentima/kinjama na raspolaganju proveriti preko *color code*-a koji se može naći u udžbeniku N. Miljković "Metode i instrumentacija za električna merenja", 2016, ETF Beograd, (https://zenodo.org/record/1335250#.XaCOgGZS_IV, pristupljeno 8. novembra 2020) na strani 26. Testirati rad programa. Po izboru, koristiti *color code* sa interneta.

Zadatak #2: Crvenu diodu povezati na digitalni pin 8 ("ledPin1"), a zelenu diodu povezati na digitalni pin 13 ("ledPin2") i uvesti ista kašnjenja za obe diode: "ledON" i "ledOFF" u programu *Primer1.ino* i definisati nazive dioda i kašnjenja kao globalne promenljive u programu. Omogućiti da se crvena i zelena LED uključuju i isključuju nazimeno i snimiti kod u fajl pod nazivom *Primer2.ino*.

Proveriti otpornosti i ostalih otpornika koji su na raspolaganju u ovoj vežbi primenom *color code* šeme. Zameniti otpornik otpornosti 220Ω u kolu drugim otpornikom u dogovoru sa dežurnim. Koja se promena uočava?

Zadatak #3: Otvoriti Arduino primer *AnalogReadSerial.ino* iz *01.Basics* grupe ugrađenih primera i snimiti kao *Primer3.ino*. Primetiti da:

- 1) Za definisanje serijske komunikacije je potrebno pozvati funkciju "Serial.begin()". Pogledati uputstvo <https://www.arduino.cc/en/Serial/Begin> (pristupljeno 8. novembra 2020). Ulazni parametar ove funkcije je *boud rate* tj. broj bita po sekundi (bps) koje je moguće "preneti" serijskoj vezom (tj. brzina serijske konekcije). Neke od tipičnih vrednosti su 4800 bps, 9600 bps, 14400 bps, 19200 bps, ... U ovom primeru će ostati podrazumevana vrednost 9600 bps.
- 2) Primetiti da je unutar "loop()" funkcije definisano čitanje napona koji se vodi na analogni ulaz tj. na kanal A0 funkcijom "analogRead()" (<https://www.arduino.cc/en/Reference/AnalogRead>, pristupljeno 8. novembra 2020).
- 3) Za štampanje/prikaz na serijskom portu koriste se funkcije "Serial.print()" i "Serial.println()". Funkcija "Serial.println()" omogućava da se prilikom štampanja omogući prelazak u novi red tj. liniju.
- 4) Primetiti da je za svaku iteraciju dodato kašnjenje od 1 ms između očitavanja.

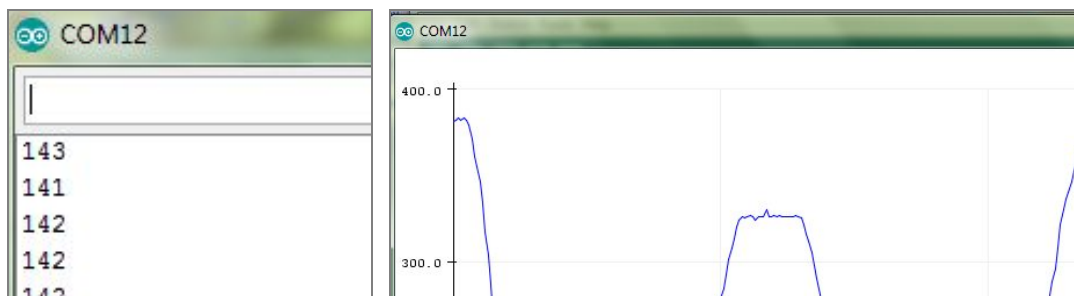
Povezati potencijometar na pin A0 i na napajanje sa UNO R3 pločice od 5 V i testirati rad aplikacije. Da bi se proverila ispravnost prikaza na serijskom portu, potrebno je otvoriti prozor za posmatranje serijskog porta. Postoje dve opcije tj. dva načina da se posmatraju podaci na serijskom portu (pogledati Sl. 2):

- 1) Iz padajućeg menija odabirom *Tools/Serial Monitor* vrši se numerički prikaz u novom prozoru.
- 2) Iz padajućeg menija odabirom *Tools/Serial Plotter* vrši se grafički prikaz u novom prozoru.

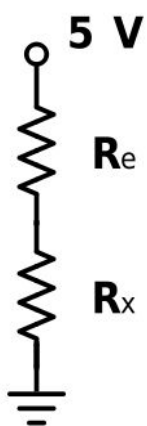
Na Sl. 2 je prikazan izgled oba prozora za prikaz podataka na serijskom portu³. Uneti prikaz u V na serijskom portu i snimiti ove izmene u postojećem fajlu pod nazivom *Primer3.ino*.

Zadatak #4: Realizovati naponski razdelinik kao na Sl. 3. Ako je $R_e = 220 \Omega$, izračunati nepoznatu otpornost R_x , ako je poznato napajanje u kolu $E = 5 V$ i napon na nepoznatom otporniku U_x . Povezati napon na nepoznatom otporniku na A4 analogni ulaz na UNO R3 mikrokontrolerskoj pločici i ispisati na serijskom portu vrednost otpornika u Ω . Snimiti ovaj kod pod nazivom *Primer4.ino*.

³ Primetiti da nije moguće istovremeno prikazivati podatke sa serijskog porta i na grafiku i na monitoru.



Slika 2, Na levom panelu je prikazan izgled numeričkog monitora, a na desnom grafičkom prozoru za prikaz podataka sa serijskog porta.



Slika 3, Naponski razdelnik za merenje nepoznate otpornosti R_x .

Ime i prezime studenta/studentkinje	Broj indeksa

Laboratorijska vežba br. 1 - Uvod u Arduino I deo

Zadaci #1, #2, #3 i #4: Uneti u tabelu koliko memorije u procentima zauzimaju sledeći programi:

	<i>Primer1.ino</i>	<i>Primer2.ino</i>	<i>Primer3.ino</i>	<i>Primer4.ino</i>
memorija [%]				

Zadatak #4: Testirati naponski razdelnik primenom tri nepoznate otpornosti. Koristiti program koji se nalazi u fajlu *Primer4.ino*. Rezultate poređenja otpornosti primenom *color code* šeme i metode naponskog razdelnika uneti u sledeću tabelu:

otpornici	otpornost iz <i>color code</i> -a [Ω]	tolerancija iz <i>color code</i> -a [%]	otpornost primenom naponskog razdelnika [Ω]
R_1			
R_2			
R_3			

Izračunati relativne greške merenih otpornosti primenom naponskog razdelnika i otpornosti određenih primenom *color code*-a i upisati u tabelu:

	Δ_1	Δ_2	Δ_3
relativna greška [%]			