2. Arduino razvojna platforma

Arduino predstavlja *open source* [1] platformu za kreiranje projekata iz oblasti elektronike. Arduino razvojna platforma se sastoji iz dva dela. Jedan deo je hardverski, koji se odnosi na Arduino razvojnu ploču na kojoj se nalazi mikrokontroler i prateće hardverske komponente, dok drugi deo predstavlja softver ili *IDE* (*Integrated Development Environment*), koji služi za pisanje programskog kôda, prevođenje kôda na mašinski jezik (kompajliranje) i spuštanje koda na hardversku platformu.

Arduino je brzo postao popularan, kako u svetu profesionalne elektronike, tako i kod ljudi kojima elektronika predstavlja hobi. Jedan od glavnih razloga je činjenica da je programiranje mikrokontrolera u okviru Arduino platforme dosta jednostavno i zahteva samo USB kabl i PC računar. Pored toga, Arduino softver koristi pojednostavljenu verziju C++ programskog jezika što dodatno olakšava učenje programiranja mikrokontrolera [2]. Zahvaljujući svojoj popularnosti i otvorenosti, oko Arduino platforme je nastala čitava jedna zajednica entuzijasta koji slobodno razmenjuju iskustva i dele svoje projekte. Na internetu se može naći veliki broj zanimljivih projekata [3] ali i softverskih biblioteka koje olakšavaju pisanje složenih programa i interakciju Arduino platforme sa raznim drugim hardverskim uređajima.

Arduino projekat je započeo 2003 godine i razvili su ga studenti iz Italije, grada *Ivrea (Interection Desing Institute Ivrea)* koji su za cilj imali da razviju što jednostavniju platformu koja će omogućiti interakciju sa senzorima, aktuatorima i drugim razvojnim pločama. Naziv Arduino potiče od imena Arduin koji zapravo predstavlja ime bara u kome su se studenti okupljali.Zapravo, to je staro rimsko ime jednog slavnog vojskovođe iz *Ivrea*-e [4].

Postoje razne Arduino hardverske platforme. Svaka od njih ima drugačija svojstva tj. različit broj pinova, različite mikrokontrolere ili mikroprocesore što za sobom povlači različite brzine rada. U ovoj knjizi se obrađuje Arduino UNO platforma koja predstavlja najčešće korišćenu platformu. Međutim kada se steknu osnovna znanja pri radu sa jednom platformom vrlo lako se može preći na rad sa nekim drugim platformama. Spisak trenutno aktivnih patformi dat je u tabeli 2.1. [5].

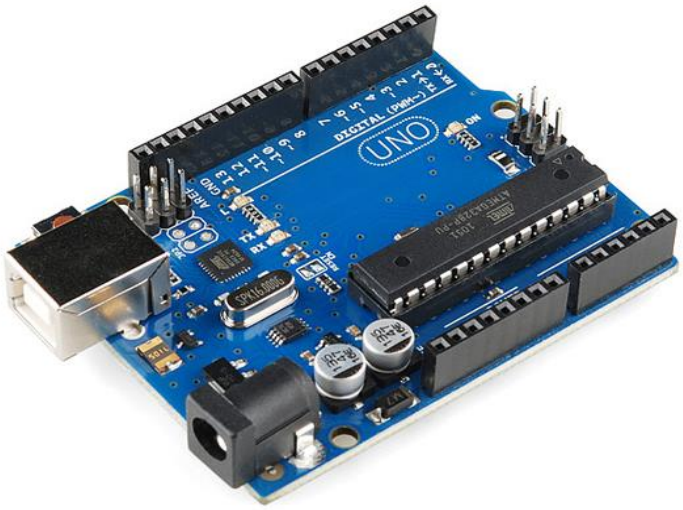
Tabela 2.1 - Neke od Arduino platformi

Naziv platforme	Korišćeni mikrokontroler – bitska širina	Brzina rada	Broj pinova	Radni napon
Arduino UNO	**ATmega328P – 8 bit**	**16MHz**	**14 – Digitalnih 6 – Analognih**	**5V**
Arduino LEONARDO	**ATmega32u4 – 8bit**	**16MHz**	**20 – Digitalnih 7- Analognih**	**5V**
Arduino MICO	**ATmega32u4 – 8 bit**	**16MHz**	**20 – Digitalnih 7 - Analognih**	**5V**
Arduino NANO	**ATmega328 – 8bit**	**16MHz**	**22 – Digitalnih**	**5V**

			8 - Analognih	
Arduino MEGA 2560	**ATmega2560 – 8bit**	**16MHz**	**54 – Digitalnih** **16 - Analognih**	**5V**
Arduino DUE	**AT91SAM3X8E – 32bit**	**84MHz**	**54 – Digitalnih** **12 - Analognih**	**5V**

2.1. Arduino kao hardverska platforma

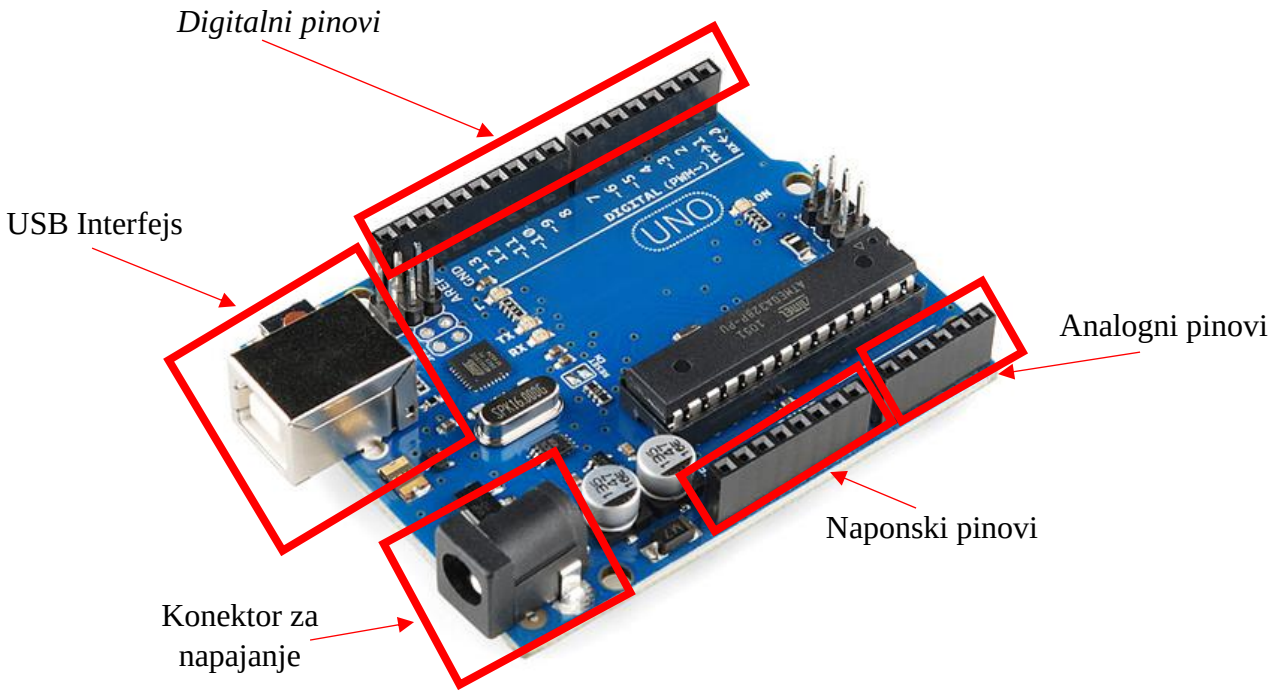
Za vežbe u ovom priručniku biće korišćena Arduino UNO hardverska platforma prikazana na slici 2.1.. Osnovni Interfejsi ove platforme prikazani su na slici 2.2.. Detalji osnovnih hardverskih platformi mogu se naći na zvaničnom sajtu Arduino projekta [6].



Slika 2.1 - Arduino Uno hardverska platforma

Konektor za napajanje služi za napajanje Arduino hardverske platforme. Po preporukama proizvođača napon napajanja može biti u ospegu od 5V do 20V, dok optimalan napon napajanja iznosi 7V.

USB interfejs služi za povezivanje razvojne ploče sa PC računarom. Preko ovog USB linka se vrši programiranje Arduina. USB interfejs se takođe može koristiti i kao serijski interfejs za razmenu podataka između Arduina i PC računara. Takođe putem ovog porta se može napajati ceo sistem jer USB port pruža mogućnost napajanja sistema naponom od 5V.



Slika 2.2. – Arduino uno hardverska razvojna platforma sa naznačenim pojedinim delovima

Naponski pinovi predstavljaju izvedene napone napajanja od 5V i 3.3V koji se koriste kada je neophodno dostaviti napajanje ostatku sistema koji se povezuje sa Arduino hardverskom platformom. Ova dva napona predstavljaju dva najčešće korišćena napona pri radu sa elektronskim komponentama. Pored napona od 5V i 3.3V tu se nalaze i 2 GND pina koji predstavljaju masu sistema. Masa sistema predstavlja tačku nultog potencijala. Tako da kada pričamo o naponu nekog pina na Arduino hardverskoj platformi, onda se podrazumeva napon, odnosno razlika potencijala između datog pina i GND pinova. Pinovi koji se nalaze u ovom delu kao i njihova svrha izslistani su u tabeli 2.2.

Tabela 2.2 - Naponski pinovi

NAZIV PINA	OPIS	NAPOMENA
IOREF	**Ulazno/izlazna naponska referenca koja daje napon/uzima napon na kome radi mikrokontroler**	
RESET	**Signal koji resetuje mikrokontroler**	
3.3V	**Daje izlaznu naponsku referncu od 3.3V**	
5V	**Daje izlaznu naponsku referencu od 5V**	
2xGND	**Pin koji obezbeđuje masu (0V) ostatku sistema**	
Vin	**Služi za napajanje arduino hardverske platforme nekim spoljašnjim naponom napajanja**	

Na **analogne pinove** se dovodi neki analogni napon u opsegu od 0V do 5V. Ovi pinovi se kasnije vode na Analogno-Digitalne konvertore, unutar mikrokontrolera, koji imaju zadatak da konvertuju analogni napon u odgovarajuću digitalnu vrednost. Kada govorimo o Arduino UNO platformi, analogna vrednost u ospegu od 0-5V se konvertuje u vrednost 0-1023 u digitalnom domenu. To znači da će analognoj vrednosti od 0V odgovarati digitalna vrednost 0 dok će analognoj vrednosti 5V odgovarati digitalna vrednost 1023. Pinovi koji se nalaze u ovom delu kao i njihova svrha izslistani su u tabeli 2.3. Više o analognim pinovima u poglavlju **(broj poglavlja)**

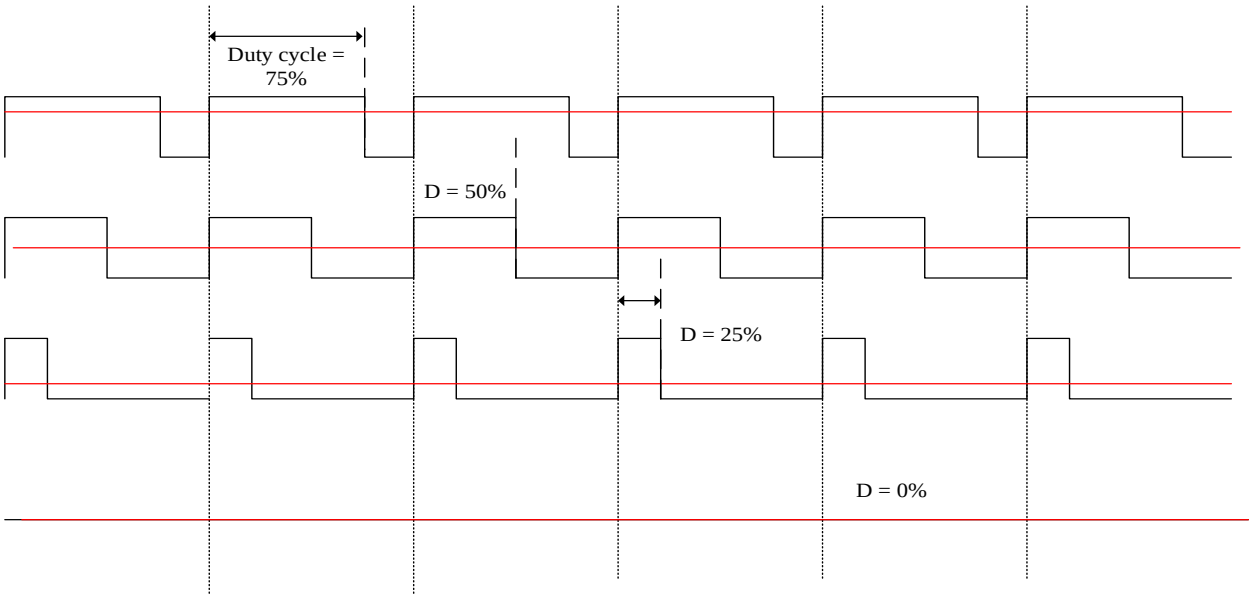
Tabela 2.3- Analogni pinovi

NAZIV PINA	OPIS	NAPOMENA
A0..A5	**Analogni Ulazi**	**A4 i A5 se koriste kao SDA i SCL linije za I2C serijsku komunikaciju (detaljnije u poglavlju** (broj poglavlja)**)**

Na ovoj platformi postoji 14 **digitalnih pinovova.** Pod digitalnim pinom se podrazumeva da se na tom pinu mogu softverski upisati dve vrednosti „0“ (logička nula) i „1“ (logička jedinica). Svi digitalni pinovi imaju dva režima rada ulazni i izlazni. Ukoliko se digitalni pin konfiguriše kao izlazni nakon upisa logičke nule ili logičke jedinice na neki pin, na tom pinu se može izmeriti vrednost od 0V ili 5V, respektivno. Dakle na digitalnom pinu je moguće očitati samo dva napona i to 0V ili 5V što je veoma bitno za dalji rad. Više o digitalnim pinovima se može saznati u poglavlju ***(broj poglavlja).***

Ukoliko se pin proglasi ulaznim, to znači da se sa tog pina može očitati jedna od dve moguće binarne vrednosti („0“ ili „1“). Koja vrednost će biti očitana zavisi od toga kako je pin povezan na ostatak sistema. Ukoliko je pin u ostatak sistema povean tako da na pin dolazi vrednost manja od 1.5V, tada ćemo na pinu očitati „0“ dok ukoliko se na pinu nalazi vrednost veća od 3V na pinu ćemo očitati „1“. Ukoliko se napon na digitalnom pinu nalazi između 1.5V i 3V nije poznato koja će vrednost biti pročitana što znači da ne smemo dozvoliti da se tu nalazi napon u ovim granicama [7]. Neki od pinova imaju mogućnost generisanja PWM signala.

PWM (eng. *Pulse-width modulation* -srp. Impulsno-širinski modulisan) signal je digitalni periodični signal kod koga je za vreme jedne periode moguće menjati dužinu trajanja logičke jedinice ili logičke nule. Obično se menja dužina trajanja logičke jedinice. Izgled PWM Signala prikazan je na slici 2.3. PWM signali se obično koriste za kontrolu brzine DC motora, za promenu intenziteta osvetljaja diode, itd.



Slika 2.3 - Impulsno širinski modulisan signal

Detaljan opis digitalnih pinova nalazi se u tabeli 2.3.

Tabela 2.4 - Digitalni pinovi

NAZIV PINA	OPIS	NAPOMENA
		Pinovi:
		0,1 – Koriste se od strane USB porta za programiranje mikrokontrolera
0..13	**Digitalni ulazi izlazi**	**3,5,6,9,10,11 – Imaju mogućnost generisanja PWM signala**
		2,3 – Imaju mogućnost generisanja prekida
		10,11,12,13 – Koriste se za SPI serijsku komunikaciju

Serijska komunikacija predstavlja razmenu podataka između Arduino platforme i nekog modula. Podaci se razmenjuju po postojećem protokolu. Koji protokol se koristi i koliko pinova se za koji protokol koristi zavisi od tipa serijske komunikacije. Najčešće su korišćena 3 tipa serijske komunikacije:

- ***I2C komunikacija*** – Serijska sinhrona komunikacija koristi dve linije
- ***UART komunikacija*** - Serijska asinhrona komunikacija koja koristi 2 linije
- ***SPI komunikacija*** – Serijska sinhrona komunikacija koja koristi četiri linije

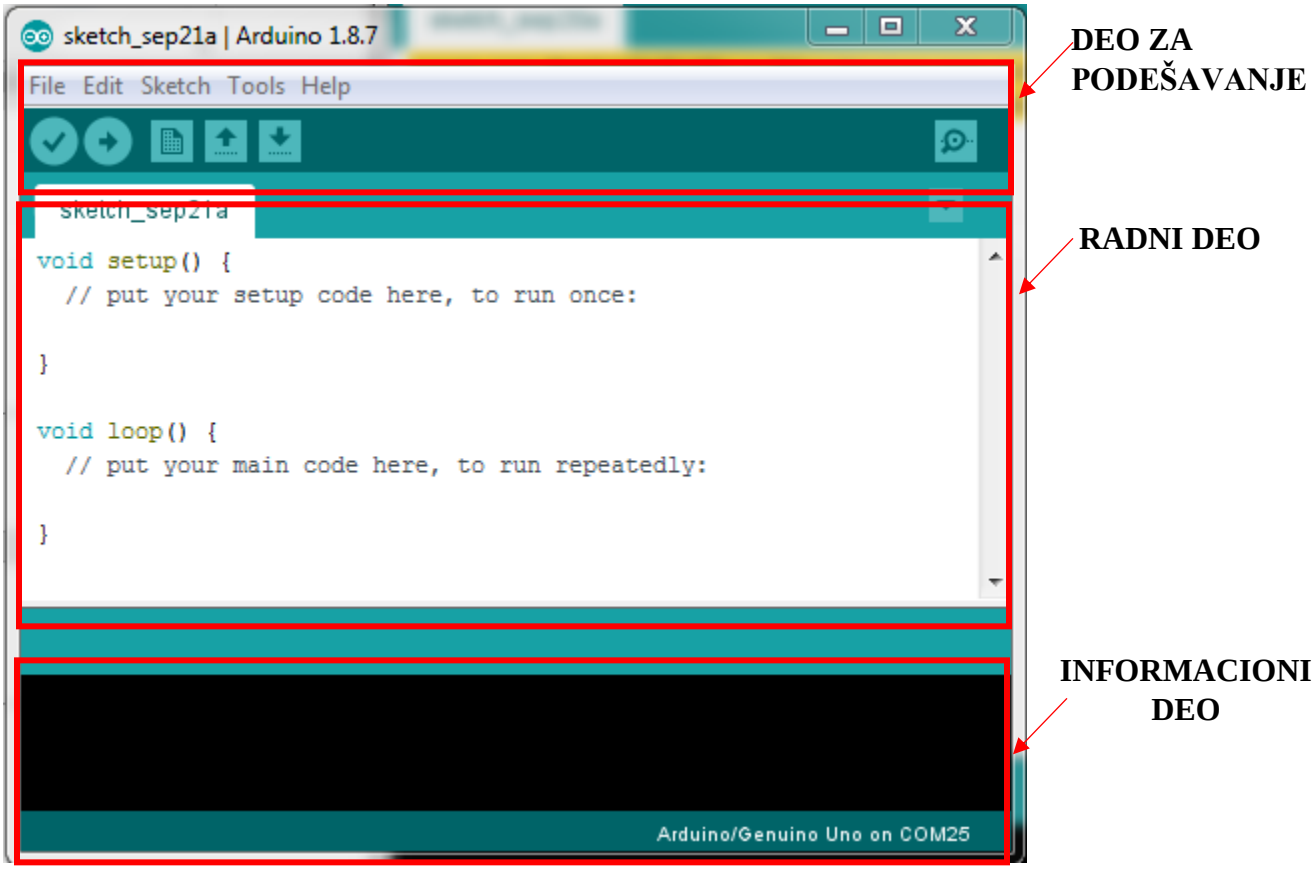
Detaljnija analiza serijske komunikacije biće sprovedena u poglavlju ***(broj poglavlja).***

2.2. Softversko okruženje za pisanje Arduino programa

Svi programi koji se kreiraju za Arduino platforme pišu se u Arduino IDE softverskom razvojnom okruženju. Potrebni softver je moguće skinuti sa linka https://www.arduino.cc/en/Main/Software . Proizvođač nudi više različitih softvera koji su pisani za različite operativne sisteme. U zavisnosti od operativnog sistema, treba skinuti odgovarajući fajl. Što se tiče Windows OS moguće su dva tipa fajla i to **.zip** fajl u slučaju da program skidamo za računar na kome nemamo privilegije administratora. Kao druga opcija nudi se Windows Installer fajl koji skidamo u slučaju da računar na kome instaliramo ovaj softver ima privilegije administratora.

Nakon što se program preuzme sa datog linka, instalira (ukoliko je to potrebno) i pokrene program, dobija se prozor prikazan na slici 2.4. Radno okruženje ovog softvera se sastoji od tri celine: radni deo, informacioni deo i deo u kome se vrše podešavanja. .

Pri radu sa Arduino platformom neizostavni deo predstavlja korišćenje određenih biblioteka. Svrha biblioteka (*library*) jeste da korisniku olakšaju pristup određenim hardverskim elementima poput analognih pinova, digitalnih PWM portova, pinova za serijsku komunikaciju itd. Biblioteke se mogu naći na internetu i nakon njihovog preuzimanja,

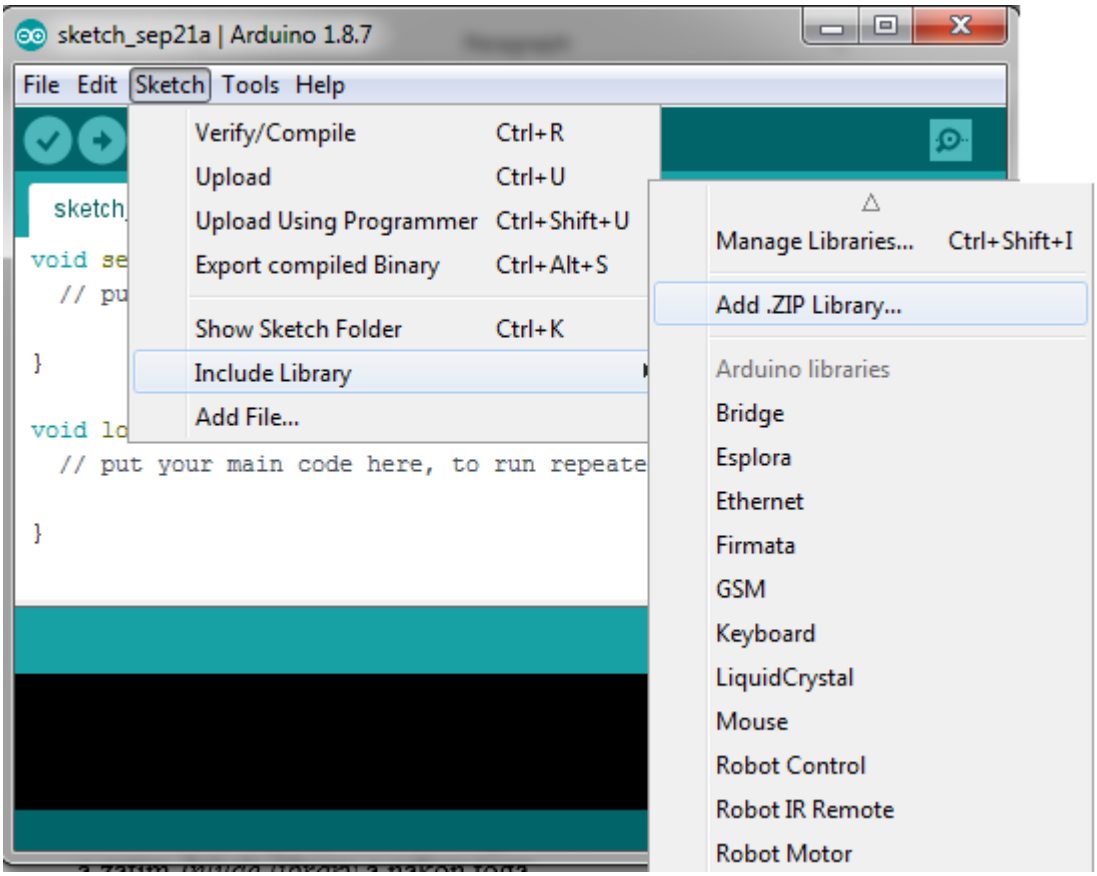


Slika 2.4 - Arduino IDE softversko okruženje

najverovatnije da će tip fajla biti .zip. Da bi bilo moguće koristiti neku biblioteku, najpre je

neophodno instalirati je u Arduino okruženje. Instaliranje biblioteke biće ilustrovano na primeru instaliranja biblioteke za rad sa tajmerima a detaljan opis same biblbioteke je urađen u poglavlju ***(broj poglavlja)***

Sa link https://github.com/PaulStoffregen/TimerOne preuzimamo biblioteku kao .**zip** fajl. Nakon toga se u Arduino softverskom okruženju selektuje meni *Sketch* a zatim *Inlude library* a nakon toga *Add .Zip file*. Ovaj postupak je ilustrovan na slici 2.5. Nakon ovoga se dobija prozor u kome treba da napravimo putanju do prethodno skinutog .zip fajla.



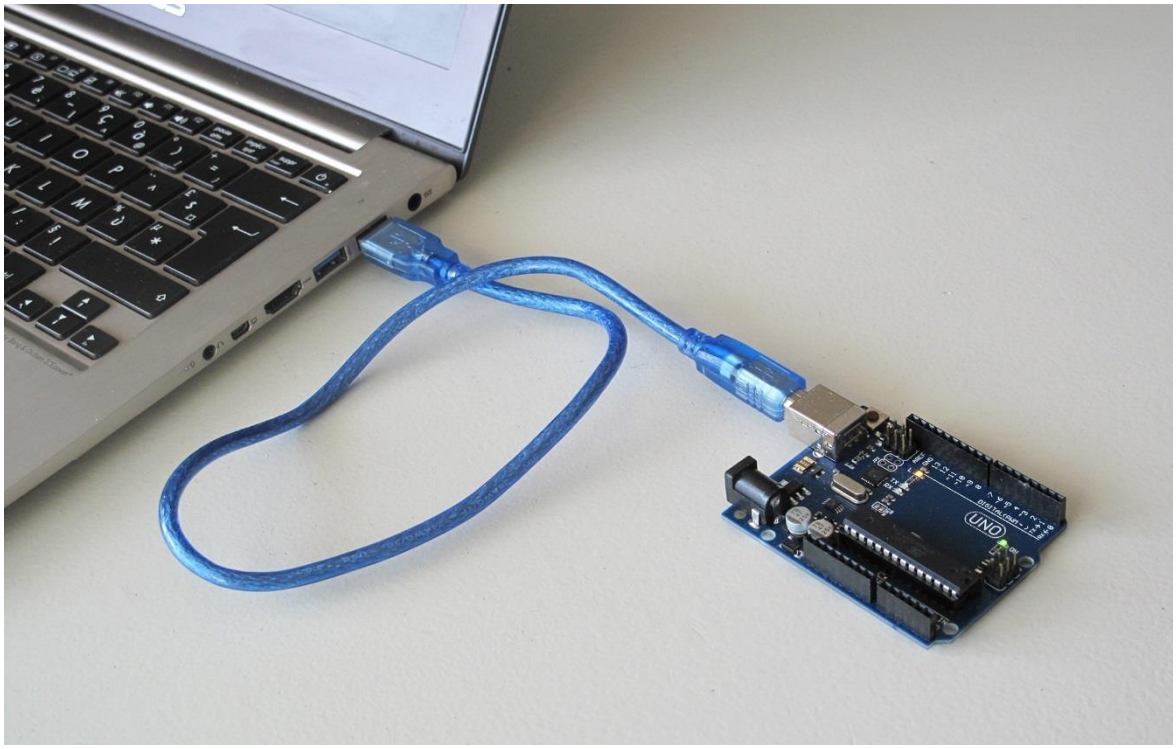
Slika 2.5 - Uključivanje biblioteke u projekat

2.3. Povezivanje Arduina sa PC računarom

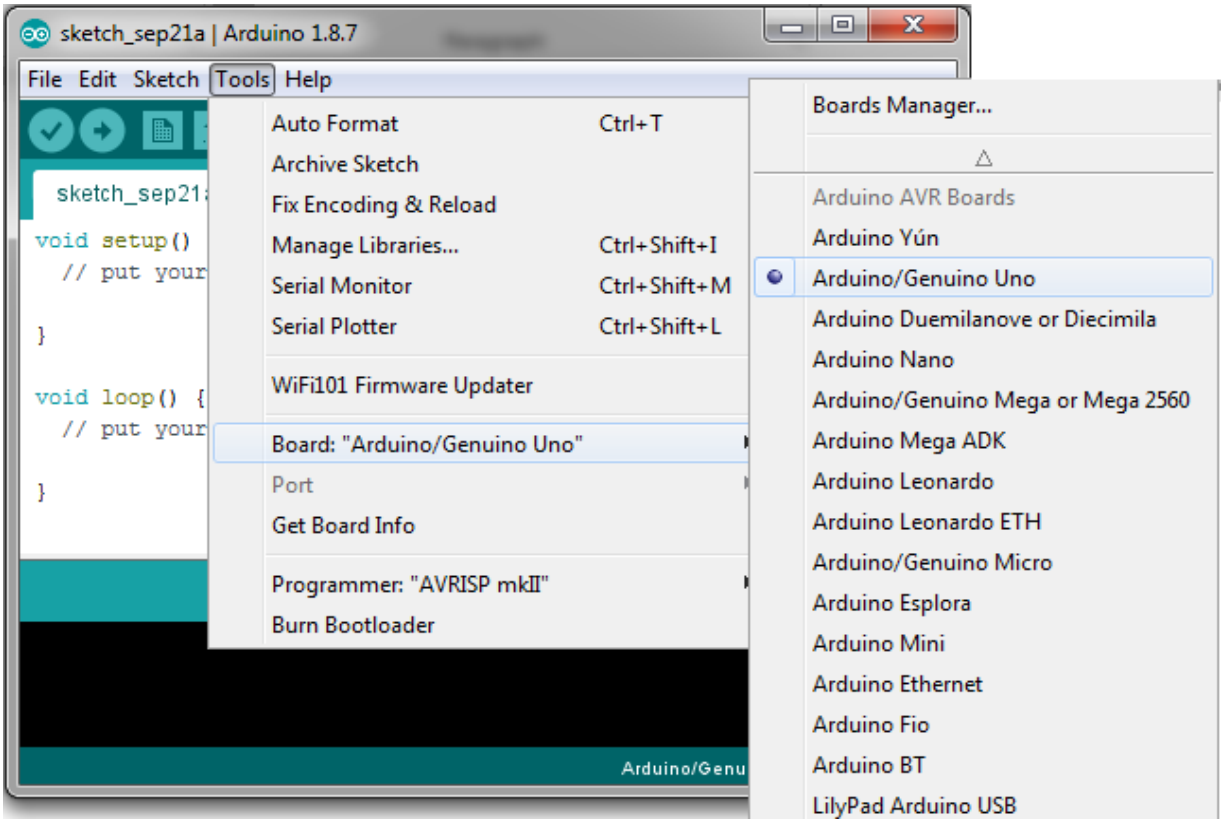
Pre nego što se započne sa pisanjem programa neophodno je pravilno povezati ceo sistem. Pod povezivanjem sistema se podrazumeva hardversko povezivanje elemenata i softverska konfiguraciju. Hardversko povezivanje sistema podrazumeva ispravno fizičko povezivanje Arduino hardverske platforme sa PC računarom. Primer ispravno povezane Arduino platforme i PC računara prikazan je na slici 2.6.

Softverska konfiguracija sistema podrazumeva određenu konfiguraciju unutar Arduino IDe softvera. Ova kofniguracija podrazumeva selekciju platforme za koju pišemo kod, a nakon

toga neophodno je izvršiti selekciju USB porta PC računara putem koga se odbija komunikacija sa Arduinom.

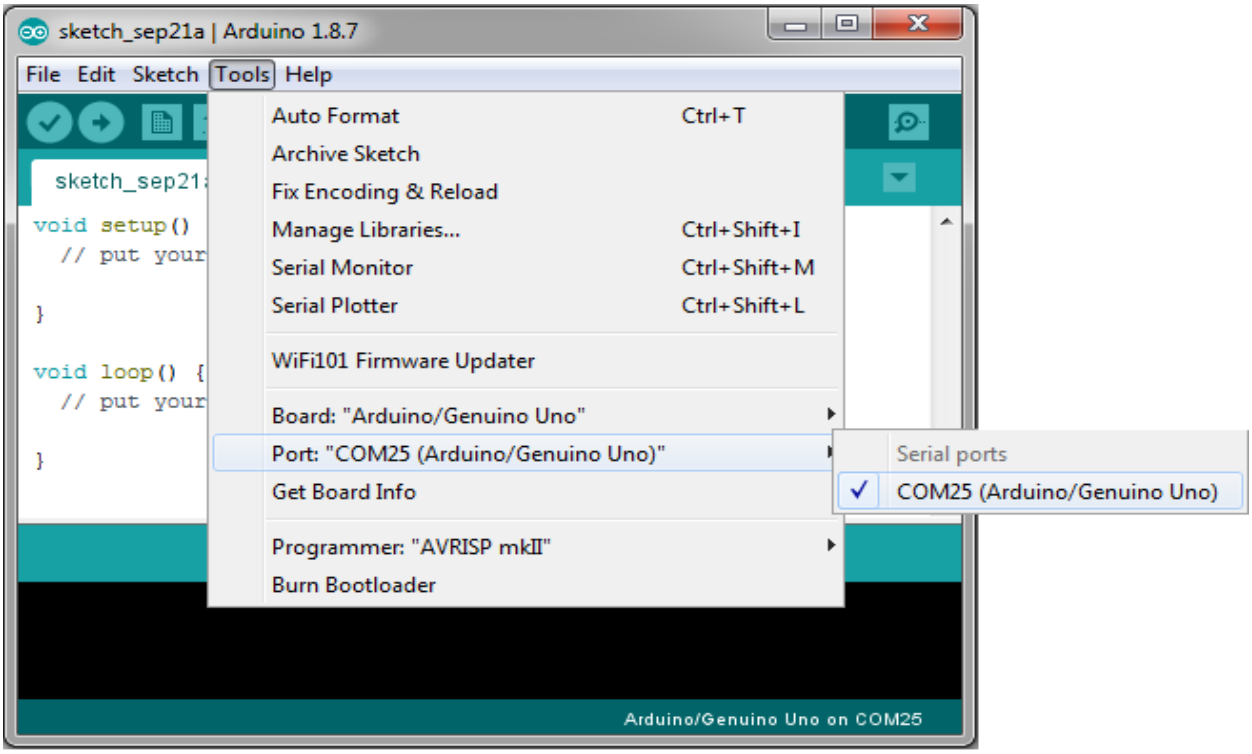


Slika 2.7 - Povezivanje Arduino UNO hardverske platforme sa PC računarom



Slika 2.6- Ispravno konfigursano Arduino IDE softversko okruženje

Prvo se vrši izbor platforme za koju pišemo program i to se radi tako što se u meniju *tools* izabere sekcija *Board* a zatim se iz izlistanih platformi izabere platforma za koju pišemo program (u našem slučaju Arduino UNO). Ovo podešavanje je ilustrovano na slici 2.7.



Slika 2.8 - Podešavanje porta

Nakon odabira platforme neophodno je izvršiti selekciju porta posredstvom koga se odvija komunikacija sa Arduino modulom. To se vrši tako što se u meniju *Tools*, sekciji *Port*, izabere onaj port koji u nazivu ima ime platforme za koju pišemo kodove. Ovaj postupak je ilustrovan na slici 2.8.

2.4. Struktura Arduino programa

Svaki program pisan za Arduino sadrži tri glavne programske celine i to:

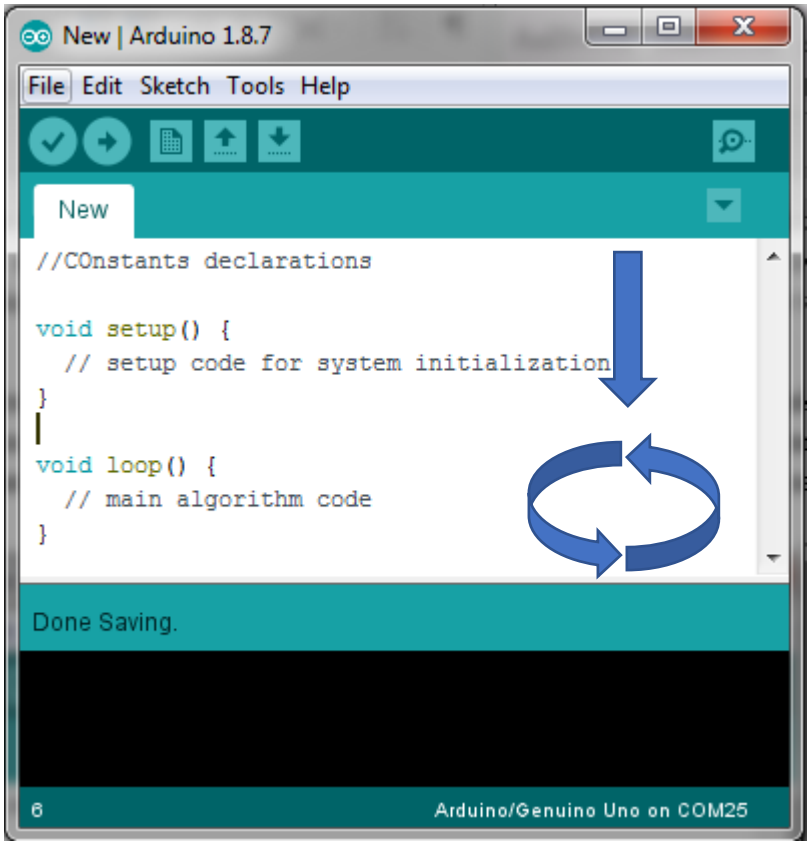
- Deklaracija promenljivih i korišćenih portova
- Inicijalizacija sistema
- Glavni program

Deklaracija promenljivih se vrši na samom početku programa. Pod deklaracijom promenljivih se podrazumeva deklaracija konstanti koje označavaju broj korišćenog pina, deklaracija promenljivih koje nose neku informaciju, itd.

Inicijalizacija sistema podrazumeva dovođenje sistema u početno stanje. To znači da je neophodno podesiti direkciju pinova, početne vrednosti koje će biti na pinovima ukoliko se pinovi proglase kao izlazni, dodela početnih vrednosti globalnim promenljivim, konfiguracija prekida, itd. Inicijalizacija sistema se vrši unutar funkcije *setup* koja se poziva samo jednom i to na samom početku izvršavanja sistema .

Nakon inicijalizacije sistema neophodno je napisati algoritme koji realizuju neke funkcionalnosti. Ovi algoritmi se većinom pišu u glavnoj programskoj petlji *loop*. *Loop* predstavlja funkciju koja se poziva svaki put**, tj. može se reći da se program stalno vrti u ovoj petlji.**

Dakle, svaki program pisan za Arduino platformu mora da sadrži *setup* i *loop* funkcije. One mogu biti prazne ali neophodno je da postoje u programu. Zbog toga nakon otvaranja novog Arduino fajla dobijamo situaciju prikazanu na slici 2.9. gde se inicijalno nalaze dve neophodne funkcije.



Slika 2.9- Struktura Arduino programa

2.5. Proces instalacije programa na Arduino platformu

Kao što je već rečeno, programi za Arduino se pišu u Arduino softverskom okruženju. Međutim sada se postavlja pitanje kako se zapravo odvija proces programiranja Arduina, tj. kako se zapravo vrši prevođenje nekog programa koji pišemo u nešto što Arduino platforma sprovodi? Na ovo pitanje pokušaćemo da odgovorimo u nastavku ovog poglavlja.

Sa jedne strane sistema imamo Arduino hardversku platformu koja predstavlja digitalni sistem. Glavna osobina digitalnog sistema jeste u tome što on radi sa digitalnim signalima, tj. sa logičkim nulama(„0“) i logičkim jedinicama(„1“). Sa druge strane se nalazi Arduino

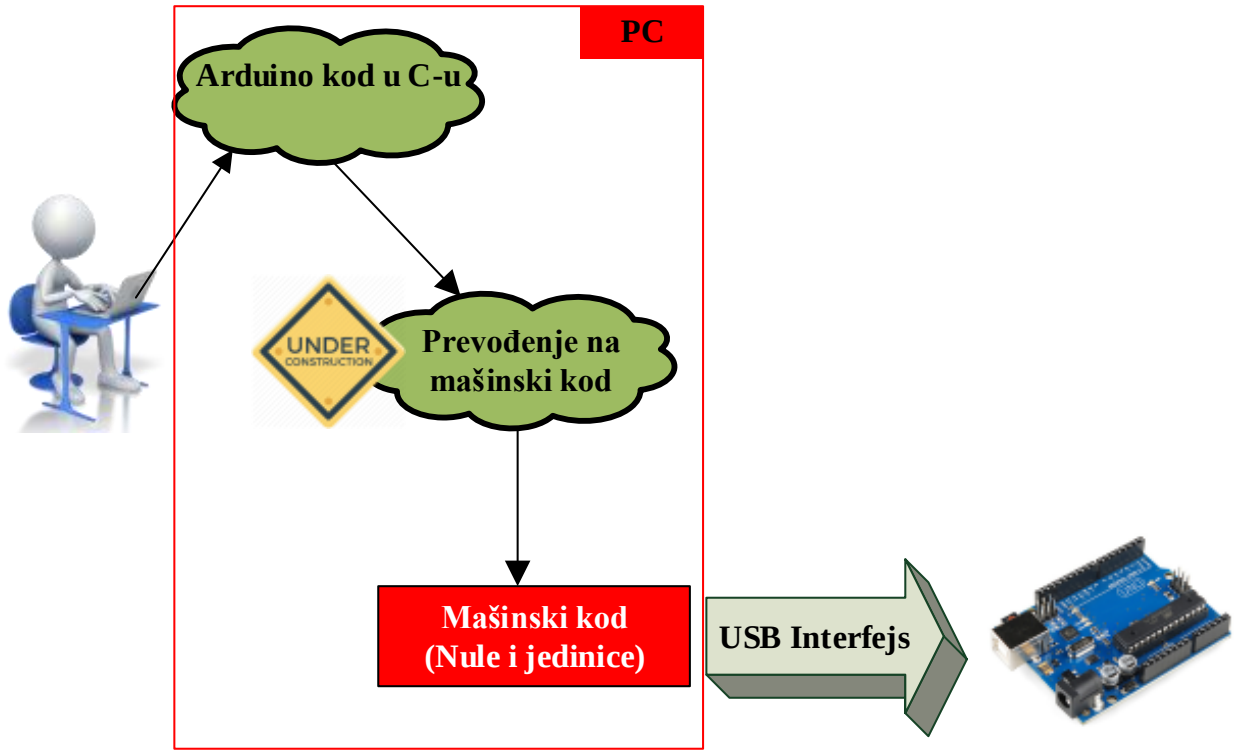
softversko okruženje u kome se piše kôd u tekstualnom formatu. Taj kôd se piše po sintaksnim pravilima programskog jezika C. Ovaj programski jezik spada u više programske jezike što znači da mikrokontroleri/računari ne mogu direktno da ga razumeju. Zbog toga je neophodno da ovaj kôd bude preved na niži programski jezik.

Unutar Arduino softverskog okruženja postoji deo koji se naziva kompajler koji ima zadatak da kôd napisan u C-u prevede na mašinski kod. Mašinski kôd predstavlja niz nula i jedinica koje predstavljaju instrukcije koje mikrokontroler izvršava jednu po jednu. Pokretanje

Slika 2.10- Primer uspešno kompajliranog Arduino programa

procesa kompajliranj se vrši klikom na dugme u obliku znaka za tačno (štikle) koji se nalazi u gornjem levom uglu kao što je prikazano na slici 2.10. Nakon toga, ukoliko je kôd pravilno napisan i nema sintaksnih grešaka, na sredni donjeg dela prozora se ispisuje *Done Compiling* što potvrđuje da je kod uspešno kompajliran.

Kada se kompajliranje završi to znači da je uspešno generisan mašinski kôd koji se još uvek nalazi u memoriji PC računara, pa je zatim neophodno taj kôd preneti u memoriju Arduina gde će se on izvršavati. Prenošenje ispravno kopajlirang koda vrši se klikom na dugme u obliku strelice na desno u gornjem uglu IDE prozora što je prikazano na slici 2.10. Jednom kada se kôd prenese na mikrokontroler on ostaje u mikrokontroleru sve dok se ponovo ne unese neki novi kôd. Ovaj mehanizam programiranja mikrokontroler ilustrovan je na slici 2.11.



Slika 2.11 - Ilustracija mehanizma instaliranja kôda na mikrokontroler

Treba znati da kada se jednom mašinski kôd uspešno upiše u memoriju Arduina, napisani program se izvršava na Arduino hardverskoj platformi potpuno nezavisno od PC računara na kome je pisan. Pri testiranju napisanog kôda, čest je slučaj da i po programiranju Arduino kontroler i PC računar ostaju povezani pomoću USB kabla. Ovakva veza PC računara i Arduina služi samo za napajanje hardverske platforme. Ukoliko bi se ova veza prekinula, a Arduino napajo iz nekog drugog izvora napajanja, program bi se na njemu izvršavao na potpuno isti način.

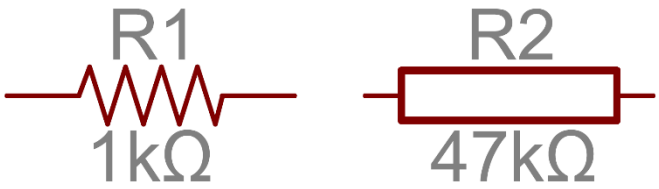
3. Sklapanje električnih kola

Tema ovog poglavlja jes formiranje električnih kola i upoznavanje čitaoca sa osnovnim zakonima koji određuju njihov rad. Cilj poglavlja je da osposobi čitaoca da samostalno sklapa i proverava ispravnost električnih kola koja će biti predstavljena u ovoj knjizi.

Pošto ovaj priručnik obrađuje temu Arduina, električna kola su znatno pojednostavljenja jer se kao izvor napajanja u tim kolima uglavnom koriste digitalni izlazi koji imaju jednosmerni- DC napon, pa su kao takva jednostava za rešavanje.

3.1. Osnovne električne komponente

Osnovnu električnu komponentu koja se koristi u gotovo svim elektronskim kolima predstavlja otpornik. Njegov najčešći zadatak u električnim kolima, koja se sreću u ovoj knjizi, jeste da ograniči protok struje. Kao posledica protoka struje kroz otpornik, na njegovim krajevima se javlja određeni napon. Ako se posmatra iz drugog ugla, tada se može reći da se struja kroz otpornik javlja kao posledica napona na njegovim krajevima. Najčešće korišćeni električni simboli otpornika prikazani su na slici 3.1. dok je realan otpornik prikazan na slici 3.2.



Slika 3.2 - Električni simboli otpornika

Slika 3.1 - Realan Otpornik

Napon i struja otpornika međusobno su povezani relacijom koja se naziva Omov zakon, a koja matematički ima sledeću formulu:

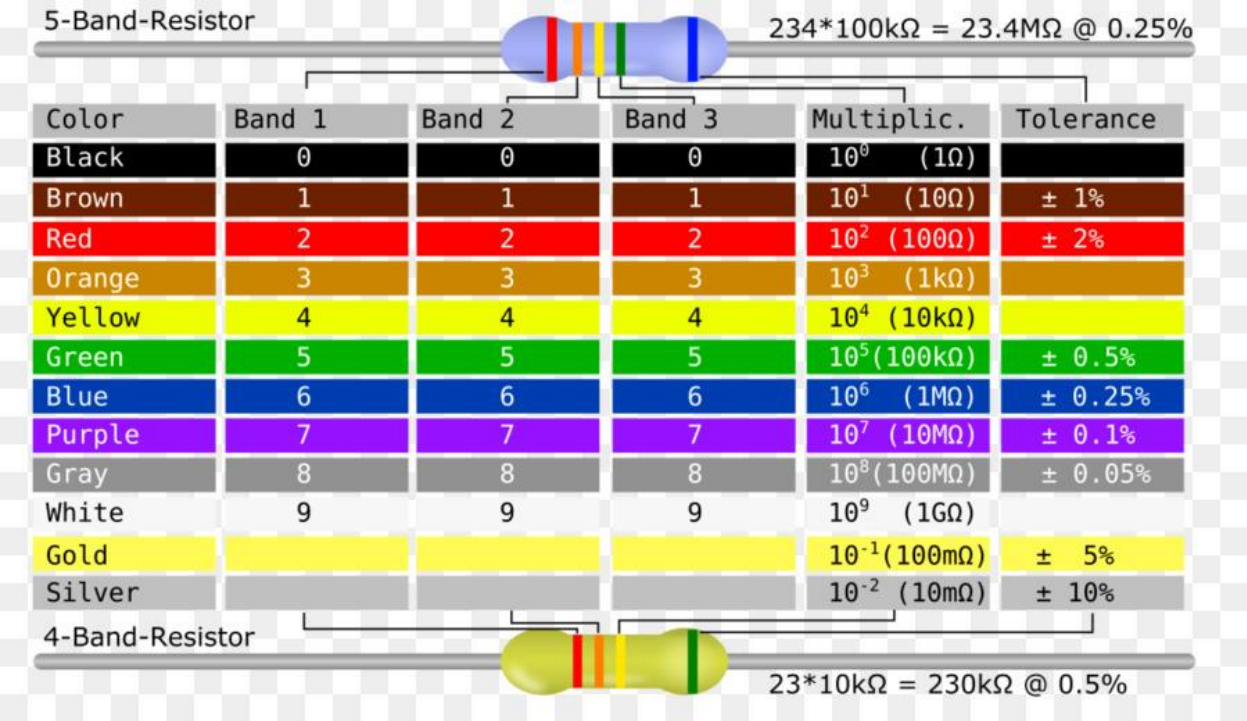
$$U = I \cdot R$$

ili

$$I = \frac{U}{R}$$

Sa U je označen napon na otporniku koji se izažava u voltima (V) dok je sa I označena struja kroz otponik koja se izražava u amperima (A). Zaključak je da je napon proporcionalan struju i da je konstanta te proporcionalnosti označena sa R što predstavlja otpornost otpornika koja se izražava u omima (Ω).

Prilikom rada sa otpornicima često postoji potreba da se pronađe otpornik neke otpornosti ili da se pročita otpornost nekog otpronika. Otpornost otpornika se može pročitati sa kućista otpornika na osnovu redosleda boja koje se na njemu nalaze. U tu svrhu se koristi tabela otpornosti prikazana na slici 3.3.



Color	Band 1	Band 2	Band 3	Multiplic.	Tolerance
Black	0	0	0	10^0 (1Ω)	
Brown	1	1	1	10^1 (10Ω)	± 1%
Red	2	2	2	10^2 (100Ω)	± 2%
Orange	3	3	3	10^3 (1kΩ)	
Yellow	4	4	4	10^4 (10kΩ)	
Green	5	5	5	10^5(100kΩ)	± 0.5%
Blue	6	6	6	10^6 (1MΩ)	± 0.25%
Purple	7	7	7	10^7 (10MΩ)	± 0.1%
Gray	8	8	8	10^8(100MΩ)	± 0.05%
White	9	9	9	10^9 (1GΩ)	
Gold				10^{-1}(100mΩ)	± 5%
Silver				10^{-2} (10mΩ)	± 10%

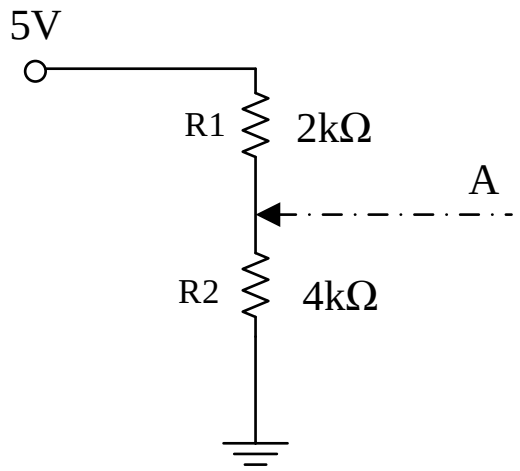
Slika 3.3 - Tabela za određivanje otpornosti

Postoje otpornici sa 4 ili 5 linija koje određuju njihovo otpornost. Boja svake linije predstavlja neki broj. Struktura broja koji predstavlja otpornost se može podeliti na 2 dela i to osnovni deo i množilac. U slučaju otpornika sa 5 linija (5-band-resistor) osnovni deo je trocifreni broj. Prvu cifru određuje boja prve linije, drugu cifru određuje boja druge linije i treću cifru određuje boja treće liniji. Četvrta linija predstavlja vrednost množioca. Da bi dobili tačnu otpornost neophodno je pomnožiti osnovni deo sa množiocem. U slučaju otpornosti koja ima 4 linije (4-band-resistor) osnovni deo je dvocifren broj koji određuju prve dve linije dok treća linija određuje množilac. Poslednje linije u nizu kod oba tipa otpornika određuju toleranciju koja nije bitna za priemre u ovom praktikumu.

Proces određivanja otpornosti biće detaljnije objašnjen na primeru otpornika sa slike 3.2. Otpornik ima 4 boje. Prva je crvena a druga je crna što nakon prevođenja u brojeve na osnovu tabele 3.3. predstavlja osnovni broj 20. Treća linija je crvena i ona predstavlja množilac koji iznosi $10^2 = 100$. Nakon množenja osnovnog dela i množioca dobija se vrednost 20 * 100 = 2000Ω ili 2kΩ.

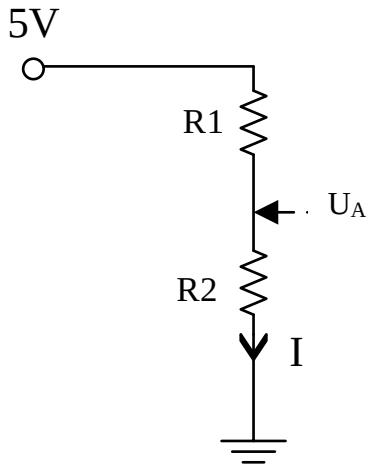
Nakon objašnjenja procesa određivanja otpornosti, koji predstavlja veoma bitan proces pri radu sa električnim kolima, sledi primer jednog tipičnog električnog kola koje se sastoji od otpornika.

Jedno prosto električno kolo koje se sastoji od dva otpornika i jednog izvora jednosmernog napona prikazan je na slici 3.4.



Slika 3.4. – Razdelnik napona

Ovakvo kolo se često naziva razdelnik napona i služi da se u tački A dobije proizvoljna željena vrednost napona kada je napon napajanja dat i fiksan (ne može se proizvoljno birati). Sada je neophodno da odredimo formulu za izračunavanje napona u tački A.



Struja *I* koja protiče kroz granu sa otpornicima jednaka je:

$$I = \frac{5V}{R_1 + R_2}$$

Na osnovu Omovog zakona i poznatog izraza za vrednost struje može se odrediti napon na otporniku R_2 koji zapravo predstavlja traženi napon u tački A. Dakle, izraz za napon u ovoj tački glasi:

$$U_A = I \cdot R_2$$

Nakon zamene izraza za struju u kolu u izraz za napon u tački A dobijamo sledeću formulu:

$$U_A = \frac{R_2}{R_1 + R_2} 5V$$

Iz priložene formule se vidi zašto ovo kolo nosi naziv razdelnik napona. Napona u tački A predstavlja podeljen napon od 5V koji je proporcionalan odnosu otpornosti R1 i R2. Napon od 5V predstavlja čest napon kada radimo sa Arduino platformom i predstavlja vrednost napona koji se nalazi na digitalnom pinu kada je on u stanju logičke jednice. Što je otpornost R2 veća napon u tački A je bliži naponu od 5V dok ukoliko je veća otpornost R1 napon je bliži naponu GND odnosno 0V. U slučaju da su u kolu jednake otpornosti, napon od 5V se deli i u tački A imamo napon od 2.5V što možemo zaključiti ukoliko se u formulu ubace vrednosti od po 2kΩ.

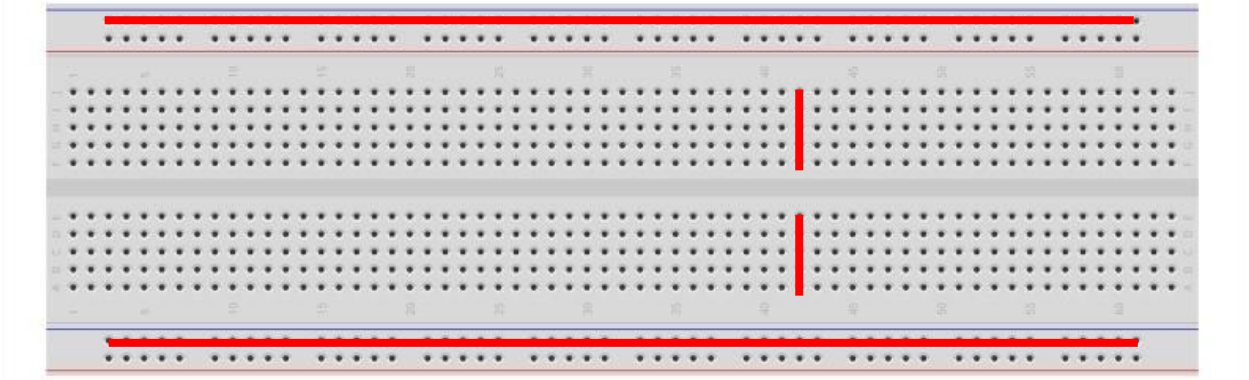
Često je otpornost otpornika konstanta. Međutim, postoje otpornici čija se otpornost menja usled raznih spoljašnjih faktora (temperatura, pritisak, razni mehanički faktori). Međutim, u električnim kolima se najčešće koristi otpornik čija se otpornost menja mehanički delovanjem, tj pomeranjem mehaničkog dela otpornika (najčešće obrtanjem). Takav otpornik se naziva potenciometar. Električni simbol kao i realan izgled potenciometra su prikazani na slici Slika 3.4 i sliciSlika 3.5

Slika 3.4 - Električni simbol potenciometra

Slika 3.5 - Realan potenciometar

3.2. Povezivanje komponenti pomoću protoborda

Protoboard predstavlja alat za kratkotrajno prototipsko sklapanje električnih kola. Njeogov zadatak je da napravi trenutne veze između pojedinih elektronskih komponenti unutar sistema. Tačnije, zadatak protoboard-a jeste da napravi kratke spojeve između krajeva električnih komponenti i da na taj način formira električno kolo. Igled protoborda je prikazan na slici Slika



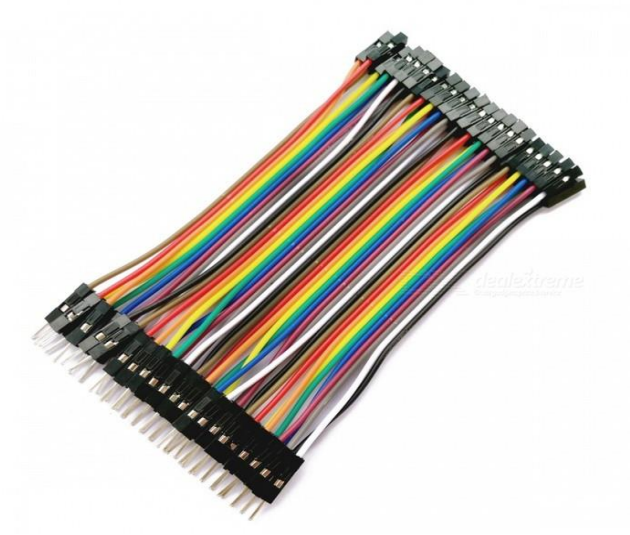
Slika 3.6 - Protobord

3.6

Nizovi tačkica koji se nalaze paralelno crvenim linijama su kratko spojeni. Spajanje susednih tačkica koje nisu spojene na protobordu vrši se pomoći žica pod nazivom kratkospajači. Zadatak kratkospajača jeste da obezbede prespajanje određenih tačkica na protobordu koje nisu kratkospojene. Kratkospajači su prikazani na slici Slika 3.7

U zavisnosti od krajeva kratkospajača postoje tri tipa kratkospajača:

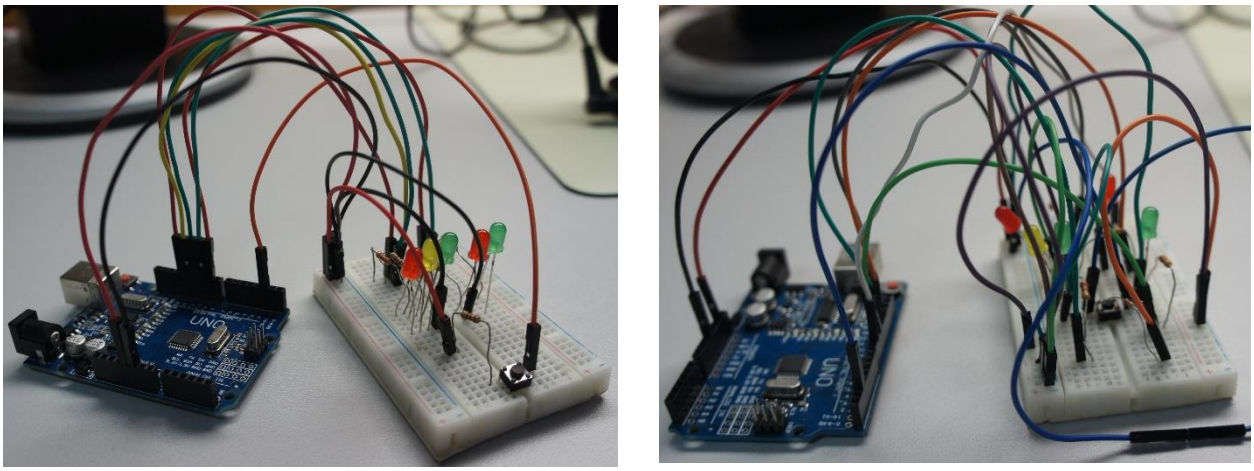
1. Muško-muški
2. Muško-ženski
3. Žensko-ženski



Slika 3.7- Kratkospajači

Postoje različite dužine kratkospajača. Lepa praksa jeste da se kraći kratkospajači koriste za povezivanje komponenti na protobordu dok se duži kratkospajači koriste za dovođenje napajanja i signala sa drugih platformi.Takođe, treba smisleno koristiti boje kratkospajača. Usvojiti neki standard korišćenja boja kratkospajača i njime se voditi kroz projektovanje sistema. Na primer crvena boja se koristi za napajanje, crna za masu, zelena za zelene diode, itd.

Korišćenjem protoborda i kratkospajača, jedno isto električno kolo može se fizički realizovati na veći broj različitih načina. Međutim pri povzivanju treba voditi računa o preglednosti kola. Realizacije kola treba da budu takve da kolo bude pregledno i da se lako mogu uočiti eventualne greške i napraviti neophodna merenja. U ovoj knjizi, za sve primere



Slika 3.8 - a- jasno, smisleno i pregledno realizovano kolo, b- nepregledna realizacija

kola date su realizacije sa protobordom koje se pregledne i vizuelno jasne. Slika Slika 3.8. daje uporedni pregled istog električnog kola koje je pregledno i nepregledno realizovano.

3.3. Merenje napona i struja u električnom kolu

Pored matematičke analize električnih kola često je neophodno izvršiti određene provere električnim merenjima. U tu svrhu se koristi multimetar koji je prikazan na slici 3.10. U ovoj knjizi su obrađene osnovne metode rada sa ovim mernim instrumentom, dok se detaljnije uputstvo za kroišćenje može naći na https://www.fluke.com/en-us/learn/best-practices/measurement-basics

Slika 3.9 - Prednja strana unimera

Ovaj multimetar pruža različite mogućnosti merenja i testiranja. Obrtnim selektorom koji se nalazi u centralnom delu ovog mernog instrumenta biraju se različiti modovi rada u zavisnosti od toga šta se meri ili provera. Neki od mogućih modova rada su:

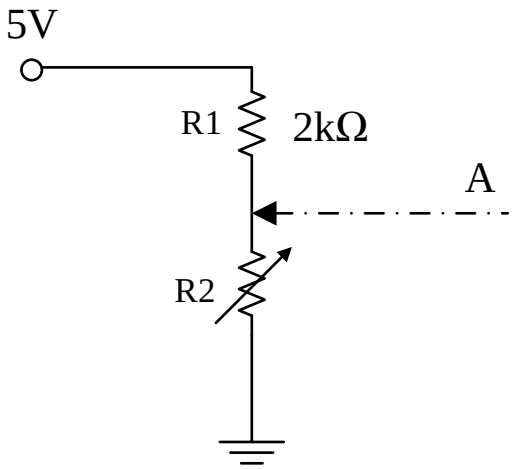
- Merenje naizmeničnih napona
- Merenje jednosmernih napona
- Merenje otpornosti
- Testiranje dioda
- Testiranje kratkog spoja

U zavinsosti od odabranog režima rada vrši se povezivanje kablova multimetra na odgovarajuće ulaze tako da simbol pored izabranog režima rada odgovara simbolu na koji povezujemo kablove.

Usled dužeg pauze u korišćenju instrumenta za vreme merenja, može doći do isključenja instrumenta. U tom slučaju je neophodno obrtni selektor vratiti u poziciju OFF a nakon toga ponovo vratiti u željeni režim rada.

Primer 3.1. - Merenje jednosmernog napona

Merenje jednosmernih napona u kolu biće objašnjeno na kolu prikazanom na slici 3.11. Zadatak je da se meri napon u tački A dok se pomera položaj potenciometra. Potrebno je uraditi sledeće:

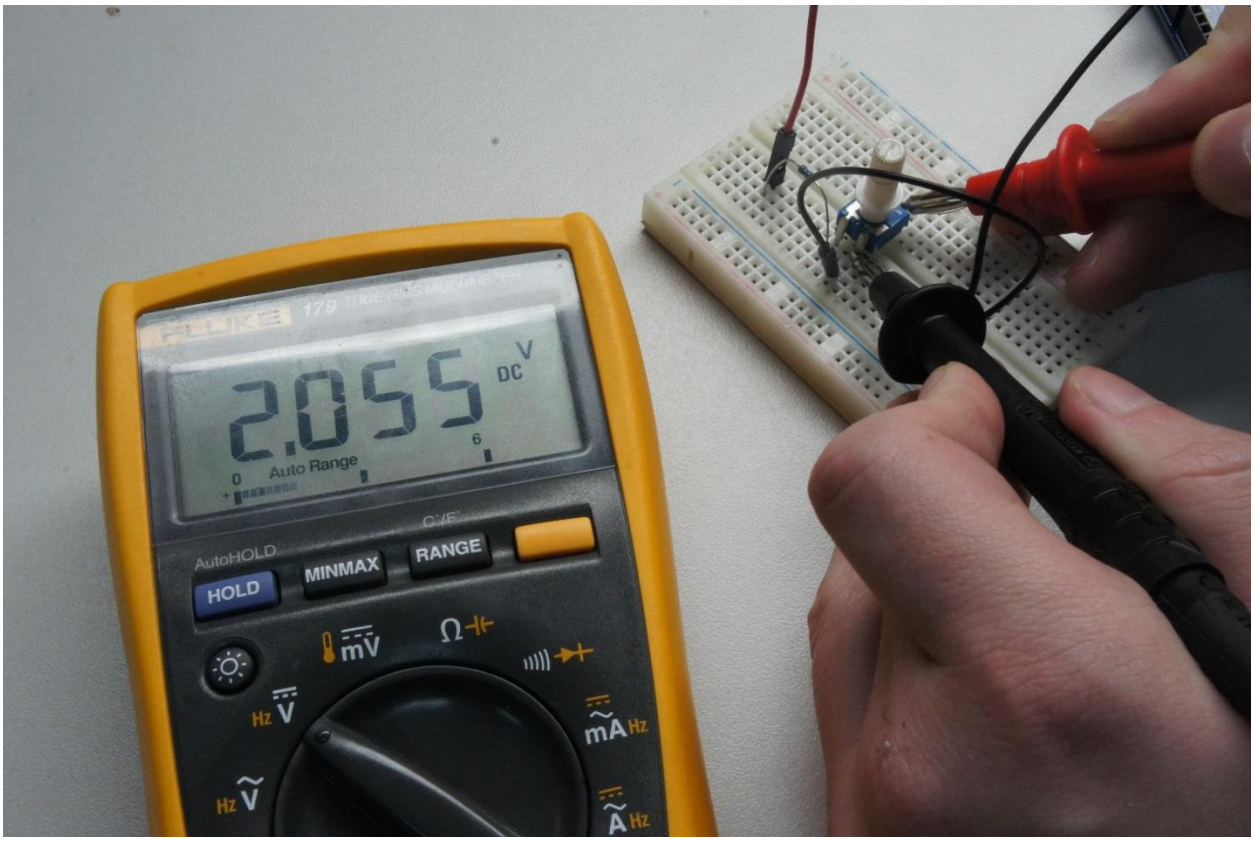


Slika 3.10 - Razdelnik napona sa potenciometrom

1. Podesiti mod rada (Slika 3.12 - Multimetar u traženom režimu rada)
2. Priključiti kablove(Slika 3.11)
3. Meriti napon (Slika 3.13)

Slika 3.12 - Multimetar u traženom režimu rada

Slika 3.11 - Način povezivanja kablova sa unimerom



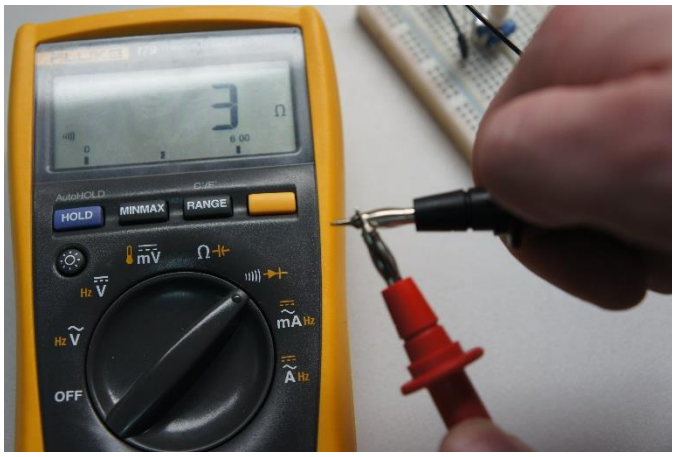
Slika 3.13 - Merenje traženog napona

Primer 3.2. - Testiranje kratkog spoja

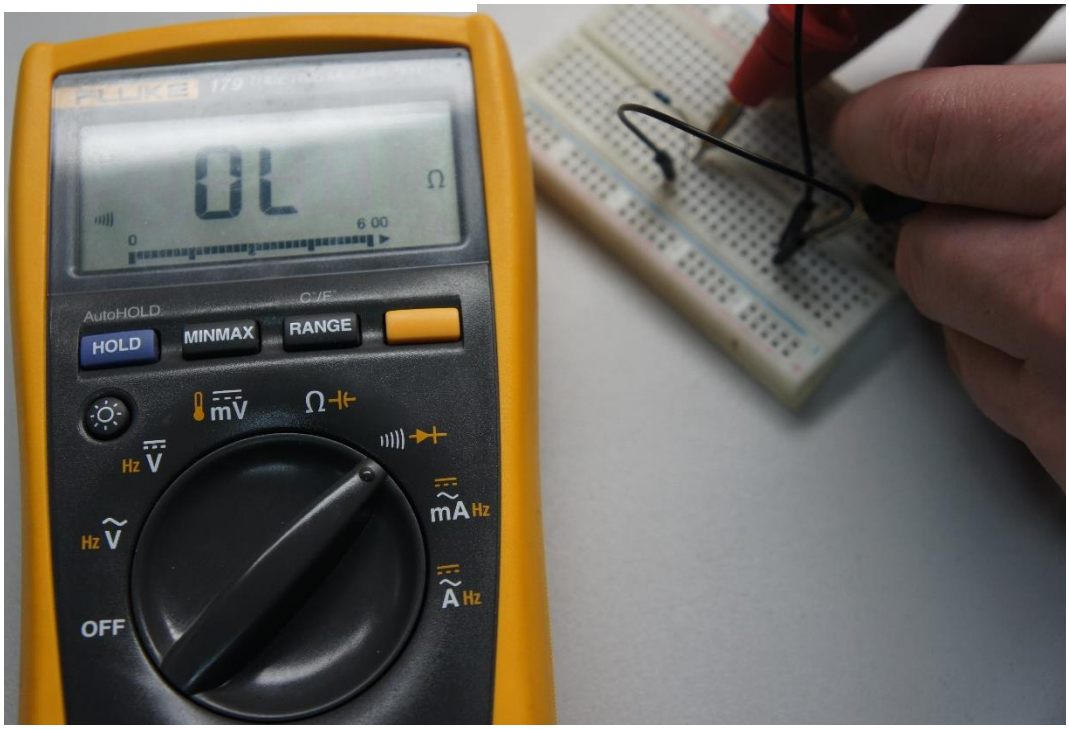
Ovaj primer ima za cilj da demonstrira mod testiranja kratkog spoja koji pruža ovaj merni instrument. Ovaj mod je veoma koristan u fazi sklapanja i testiranja električnog kola kada je neophodno proveriti da li se određene tačke, ili komponente, u kolu ispravno spojene ili da li je došlo do kratkog spajanja tačaka (ili komponenti) koje ne bi smele da budu kratko spojene.

1. Podesiti multimetar u mod koji proverava kratke spojeve (slika 3.15.)
2. Priljučiti kablove (slika 3.13.)
3. Međusobno spojiti krajeve kablova za merenje. Trebalo bi da se čuje pištanje
4. Instrument je spreman. Proveriti da li postoji kratak spoj (slika 3.17)

Slika 3.14 - Unimer u traženom režimu rada



Slika 3.15 - Provera da li se čuje pištanje



Slika 3.16 - Ispitivanje kratkog spoja