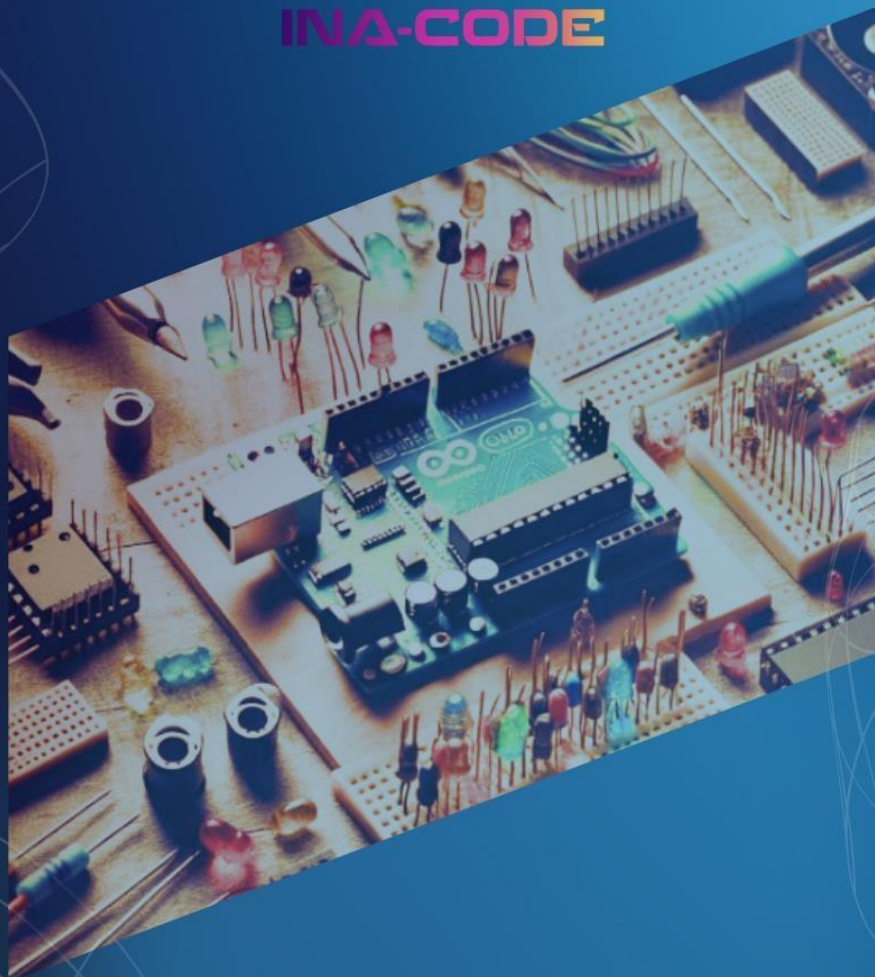


EXPERIMENT BOOKLET



INA-CODE



INNOVATIVE APPROACH FOR CODING IN DIGITAL ERA



EXPERIMENT BOOKLET



INNOVATIVE APPROACH FOR CODING IN DIGITAL ERA

2021-DE03-KA220-SCH-000024558

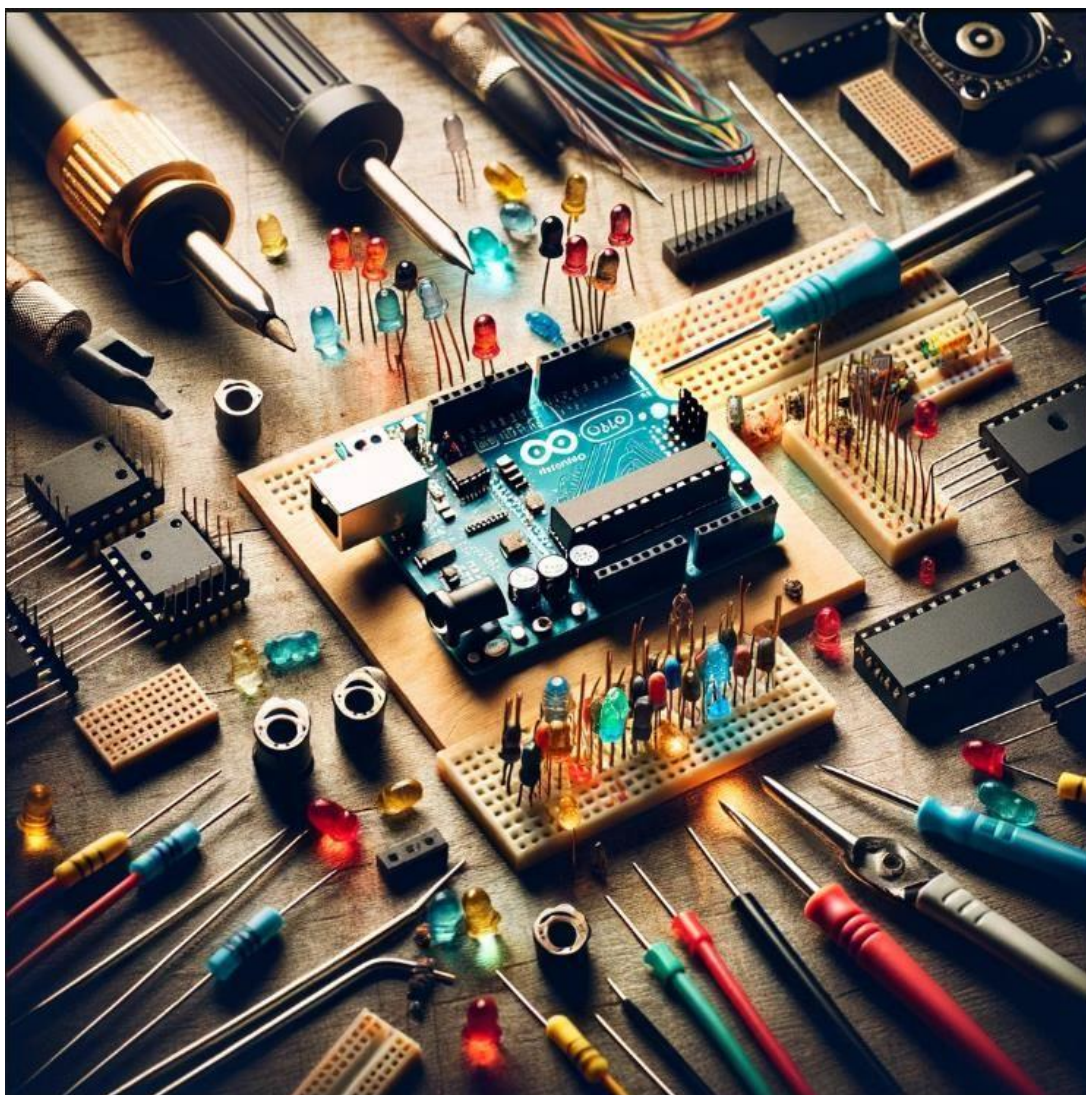


**Co-funded by
the European Union**

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them



Knjižica pokusa



Sadržaj

Predgovor	1-
Baterije.....	1
2-Uvod u elektroniku: otpornici, LED, fotootpornici i flex senzor	4
3-Uvod u elektroniku: kapacitet, induktivitet, relej, tranzistor i H-most sa istosmjernim motorom	7
4- Uvod u Arduino	10
5-Arduino platforme	13
6-Arduino programski jezik i editor-1	15
7-Arduino programski jezik i editor-2	18
8- Naredbe matematičkih operacija u Arduino IDE: Kako koristiti eksperiment s aritmetičkim operatorima.....	20
9- Naredbe matematičkih operacija u Arduino IDE: Kako koristiti aritmetičke operatore eksperiment 2.....	22
10- Kontrolne strukture u Arduino IDE: Kako koristiti kontrolne strukture-1.....	25
11- Kontrolne strukture u Arduino IDE: Kako koristiti kontrolne strukture-2.....	28
12- Kontrolne strukture u Arduino IDE: Kako koristiti kontrolne strukture-3.....	31
14- Stringovi u Arduino IDE: Kako koristiti string literale.....	37
15- Stringovi u Arduino IDE: Kako koristiti strukturu podataka String.....	40
16- Digitalni I/O operacije	43
17- Analogne I/O operacije	45

Predgovor

U današnjem svijetu, brzina i opseg digitalne transformacije zahtijevaju da se naši obrazovni sustavi prilagode tim promjenama. U tom kontekstu, naš rad, "Knjižica s eksperimentima: knjiga za učitelje i knjige za učenike koja uključuje 25 eksperimenata," predstavlja značajan korak prema poboljšanju digitalnih vještina i nastavnika i učenika. Ova je knjižica dio veće inicijative koju podupiru Europska komisija i nacionalne agencije, a čiji je cilj premošćivanje digitalnog jaza i poticanje inkluzivnijeg digitalnog obrazovnog krajolika diljem Europe.

Stvaranje ove knjižice motivirano je hitnom potrebom da se edukatori opremi potrebnim alatima i znanjem za snalaženje u digitalnoj eri, osobito u svjetlu izazova koje je istaknula pandemija COVID-19. Služi kao izvor za nastavnike da integriraju kodiranje i robotiku u svoj kurikulum, čime ne samo poboljšavaju svoje profesionalne profile nego i pripremaju učenike za zahtjeve radne snage 21. stoljeća.

Naši partneri u ovom nastojanju uključuju konzorcij škola, institucija za strukovno osposobljavanje i sveučilišta diljem Europe, a sve ujediniuje zajednički cilj unaprjeđenja digitalnog obrazovanja. Knjižica je osmišljena da bude praktična, pristupačna i prilagodljiva različitim obrazovnim okruženjima, osiguravajući da nastavnici i učenici iz različitih sredina mogu imati koristi od nje.

Svrha ove knjižice nadilazi puko podučavanje kodiranja i robotike; ima za cilj poticanje kritičkog mišljenja, kreativnosti i vještina rješavanja problema među učenicima. Uključivanjem ovih eksperimenata u svoju nastavu, nastavnici mogu pružiti zanimljivije i interaktivnije iskustvo učenja, potičući učenike da istražuju goleme mogućnosti digitalne tehnologije.

Zaključno, ova knjižica nije samo pomoć u podučavanju; to je katalizator promjena u našim obrazovnim sustavima, promičući digitalno kompetentnije i otpornije društvo. Ponosni smo što pridonosimo ovom transformativnom putovanju i nadamo se da će ono nadahnuti nastavnike i studente da prihvate izazove i prilike digitalnog doba.

Broj pokusa: 1

1-Baterije

Cilj eksperimenta

Cilj ovog eksperimenta je objasniti kako baterije rade i koje vrste baterija postoje. Učenici će naučiti kako prepoznati različite vrste baterija.

Teorijska pozadina

Prilikom odabira baterije treba voditi računa o naponu i kapacitetu. Kako bismo bolje razumjeli ove pojmove, možemo napraviti analogiju s vodom.

Zamislimo da imamo spremnik vode, napunjen vodom.

Napon je tlak u spremniku vode koji potiskuje vodu prema cijevi. Visina vode povećava tlak i ekvivalentan je naponu u električnim krugovima.

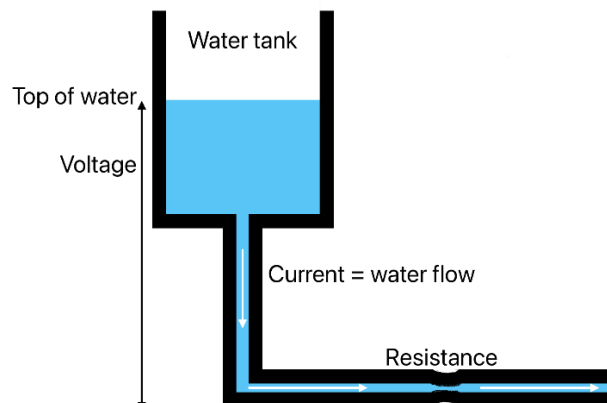
Na dnu spremnika nalazi se cijev. Voda koja teče kroz cijev predstavlja električnu struju. Što je veći tlak, to ćemo dobiti više vode, tj. što je veći napon, to je veća struja.

Prepreka se može pronaći na kraju cijevi. Predstavlja otpor jer je protok vode smanjen. Što je manji promjer cijevi (veći otpor), protok vode će biti manji. To je prikazano u električnom krugu: što je veći otpor, uz isti napon, struja će biti manja.

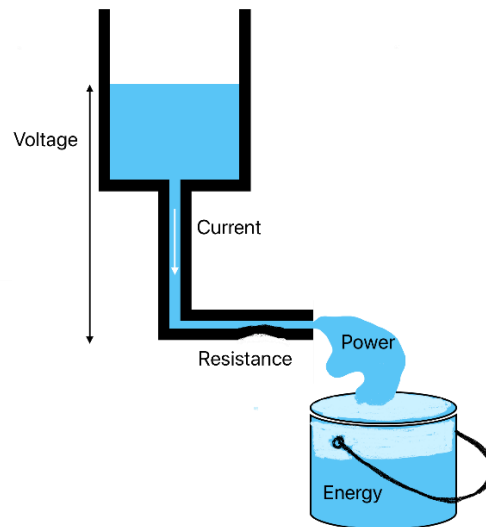
Matematičkim riječima:

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

gdje U predstavlja napon, I struju i R otpor.



Analogija se može proširiti i na moć i na energiju. Snaga je analogna protoku vode. Energija je količina vode koja završi u kanti. Snaga se mjeri u vatima (W) ili kilovatima (kW), a energija u vat-satima (Wh), milivat satima (mWh) ili u kilovatsatima (kWh).



Neki primjeri baterija prikazani su na slici.



Potrebni materijali

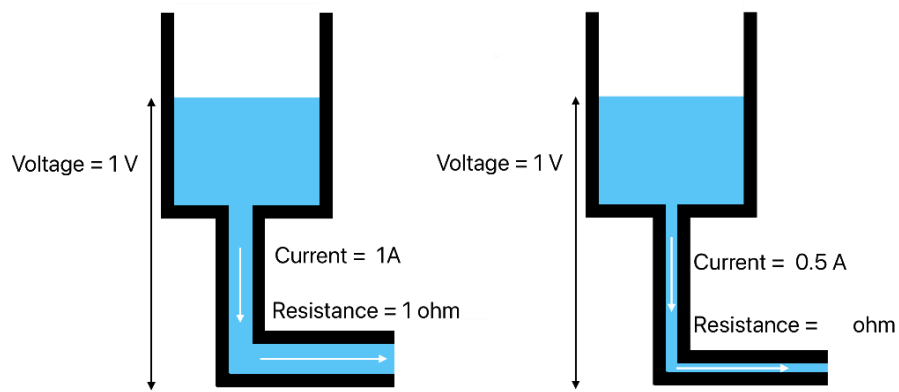
Različite baterije.

Zadatak 1.

Pronađite različite baterije u svojoj učionici i odredite naziv modela (npr. AA ili AAA) i napon.

Zadatak 2.

Za zadani napon i struju na slici odredite otpor.



Rješenje 1.

Na lijevoj slici otpor je

$$R = \frac{U}{I} = \frac{1}{1} = 1\Omega$$

Na desnoj slici otpor je

$$R = \frac{U}{I} = \frac{1}{0,5} = 2\Omega$$

Broj pokusa: 2

2-Uvod u elektroniku: otpornici, LED, fotootpornici i flex senzor

Cilj eksperimenta

Cilj ovog eksperimenta je objasniti kako spojiti LED, otpornike, fotootpornike i senzore savitljivosti na električne krugove. Učenici će naučiti koristiti te elemente na siguran način kako ne bi izgorjeli tijekom korištenja.

Teorijska pozadina

Svjetleća dioda ili LED je poluvodički elektronički element koji emitira svjetlost kada struja prolazi kroz njega.

Kraći krak diode naziva se katoda i predstavlja – pol. Priključuje se na – baterije, odnosno napajanje.

Duži krak diode naziva se anoda i predstavlja + pol. Spaja se na + bateriju, odnosno napajanje.

Ako je dioda pogrešno spojena, kroz nju neće teći struja i neće svijetliti.

Kada povezujemo LED u krug, moramo uzeti u obzir struju koja prolazi kroz tu LED. Ako kroz diodu teče prevelika struja, dioda će pregorjeti. Preporučena najveća struja kroz LED je 20 mA.

Da biste spriječili pregorijevanje, potrebno je spojiti otpornik na električni krug s LED-om. Koji otpornik koristiti određuje se prema Ohmovom zakonu:

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

gdje U predstavlja napon, I struju i R otpor.

Neka LED ima napon od 3 V. Ako koristimo bateriju od 9 V, potreban nam je otpornik:

$$R = \frac{U - U_{LED}}{I} = \frac{9 - 3}{0,02} = \frac{6}{0,02} = 300\Omega$$

Obično se koristi otpornik sa standardnim vrijednostima koje su najbliže izračunatoj vrijednosti. U ovom primjeru, to je otpornik od 330Ω.

Zadatak 1

Odredite koji otpornik trebamo koristiti ako LED spojimo na bateriju od 4,5 V.

Rješenje 1

Moramo koristiti:

$$R = \frac{U - U_{LED}}{I} = \frac{4,5 - 3}{0,02} = \frac{1,5}{0,02} = 75\Omega$$

Zadatak 2

Istražite na internetu pad napona na LED diodama drugih boja i ispunite tablicu.

Boja LED-a	napon
Žuta boja	
naranča	
Crvena	
Plava	
zelena	
ljubičica	

Rješenje 2.

Boja LED-a	napon
Žuta boja	1,9-2,1 V
naranča	2,0-2,1 V
Crvena	1,6-2,0 V
Plava	2,7-3,2V
zelena	1,9-2,2 V
ljubičica	2,8-4,0 V

Zadatak 3

Isprobajte simulator elektroničkih sklopova: <https://www.falstad.com/circuit/e-voltdivide.html>

Prikazani krug ima dvije paralelne grane. Napon baterije je 10V.

Dva 10Ω otpornici su u prvoj grani, što je ukupno 20Ω . Struja kroz ove otpornike je $I_1 = 0,5 \text{ A}$.

U drugoj grani su četiri otpornika od $10 \text{ k}\Omega$, što je ukupno 40Ω . Struja kroz ove otpornike je $I_2 = 0,25 \text{ A}$.

U animaciji je grana s većom strujom prikazana točkama koje putuju brže, dok točke u grani s manjom strujom putuju sporije.

Odredite vrijednosti otpornika u drugoj grani tako da vrijednosti struje budu jednake. Testirajte u simulatoru.

Rješenje 3

Četiri otpornika od 5Ω mora biti u drugoj grani, što je ukupno 20Ω . Struja kroz te otpornike tada bi bila $I_2 = 0,5 \text{ A}$.

Zadatak 4

Kad su dva 10Ω otpornika u prvoj grani, tada je napon između njih $U_1 = 5 \text{ V}$.

Kad se jedan od otpora promijeni, promijenit će se i napon između njih. Ako je gornji otpornik 10Ω a donji je 30Ω , napon između njih bit će $7,5 \text{ V}$.

Odredite vrijednost donjeg otpornika tako da napon između njih bude 8 V .

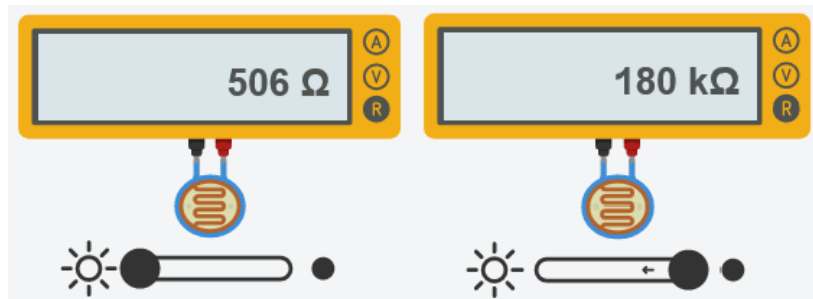
Otpornik koji mijenja svoju vrijednost naziva se **apotenciometar**.

Rješenje 4

Zajedno s gornjim otpornikom 10Ω , donja bi trebala biti 40Ω .

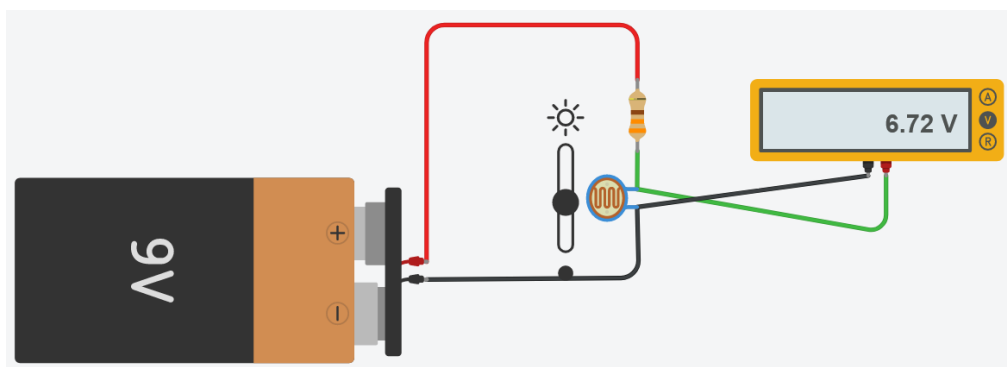
Zadatak 5

Fotootpornik je otpornik koji se koristi za mjerenje svjetlosti. Što više svjetla pada na njega, njegov otpor je manji. Što manje svjetla pada na njega, to je njegov otpor veći. Fotootpornik se ponaša kao da je promjenjivi otpornik, tj. potenciometar. Jedina razlika je u tome što ga ne podešavamo vlastitim rukama, već osvjetljenje sobe.



Kako se mijenja otpor fotootpornika, mijenja se i njegov napon. Što je veći otpor fotootpornika (nema svjetla), veći mu je i napon.

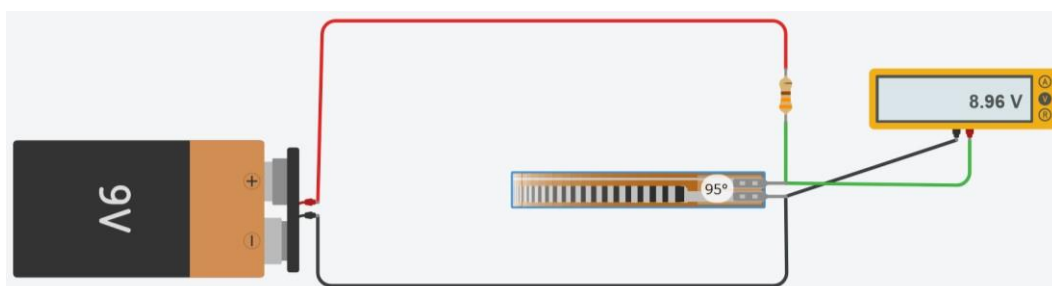
U TINKERCAD-u spojite 9V bateriju, otpornik, fotootpornik i voltmetar, prema primjeru na slici. Promijenite svjetlo na fotootporniku i promatrajte kako se mijenjaju njegov otpor i napon.



Zadatak 5

Senzor savijanja je senzor koji mjeri količinu otklona ili savijanja. Obično se koristi u biotehnoškim nosivim proizvodima za praćenje položaja i pokretljivosti ljudskih zglobova.

Spojite flex senzor prema shemi i izmjerite otpor i napon.



Broj pokusa: 3

3-Uvod u elektroniku: kapacitet, induktivitet, relej, tranzistor i H-most s istosmjernim motorom

Cilj eksperimenta

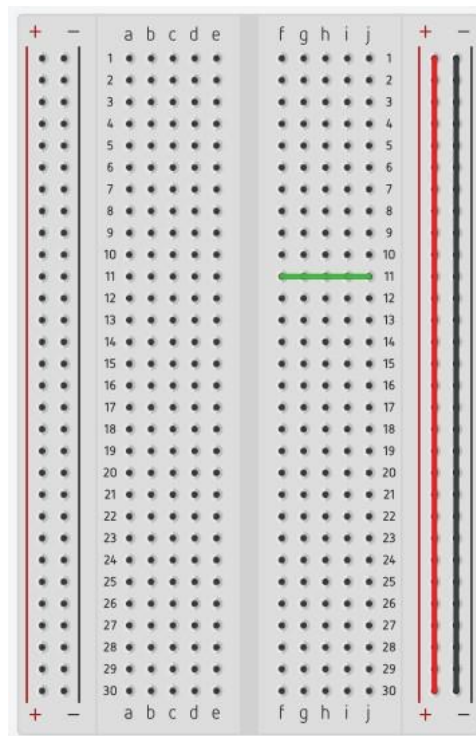
Cilj ovog eksperimenta je objasniti kako spojiti kapacitet, induktivitet, relej, tranzistore i istosmjerne motore na električne krugove. Učenici će naučiti koristiti te elemente na siguran način kako ne bi izgorjeli tijekom korištenja.

Teorijska pozadina

Akondenzator je elektronički element koji može pohraniti električni naboj. Fizička veličina koja opisuje sposobnost pohranjivanja naboja je električni kapacitet. **Kapacitet** mjeri se u **Farad (F)**. Povećanje napona također povećava količinu pohranjenog naboja. Istodobno, kapacitet se ne mijenja. Najčešće kondenzator ima dvije paralelne vodljive ploče koje se elektrificiraju spajanjem na izvor istosmjernog napona. Jedna ploča spojena je na pozitivni, a druga na negativni pol izvora.

Kada električna struja teče kroz zavojnicu, stvara se magnetsko polje. **Azavojnica** je elektronički element koji pohranjuje energiju elektromagnetskog polja. Koliko će energije biti pohranjeno ovisi o induktivitetu i struji koja prolazi kroz samu zavojnicu. **Theinduktivnost** zavojnice je veličina koja se koristi za opisivanje sposobnosti pohranjivanja energije magnetskog polja. Što je veća struja kroz zavojnicu, veća je magnetska indukcija zavojnice. Jedinica mjere za induktivitet je **Henri (H)**.

Protoboard je konstrukcijska ploča koja se koristi za spajanje prototipa električnih krugova. Pet rupa u jednom redu su u kratkom spoju. Spojimo li električni element ili žicu na dva od njih, oni će se ponašati kao da su kontakti izravno spojeni. Isto vrijedi i za stupce ispod oznaka + i - na krajnjoj lijevoj i krajnjoj desnoj strani protoboarda.



Zadatak 1

Razmotrimo jednostavan električni krug koji sadrži samo izvor, jedan otpornik, a **kondenzator**, i prekidač. Proučite što se događa sa strujom kada se prekidač uključi i isključi. Dodajte žarulju u krug i promatrajte kada će zasvijetliti.

Link na simulator: <https://www.falstad.com/circuit/e-cap.html>

Zadatak 2

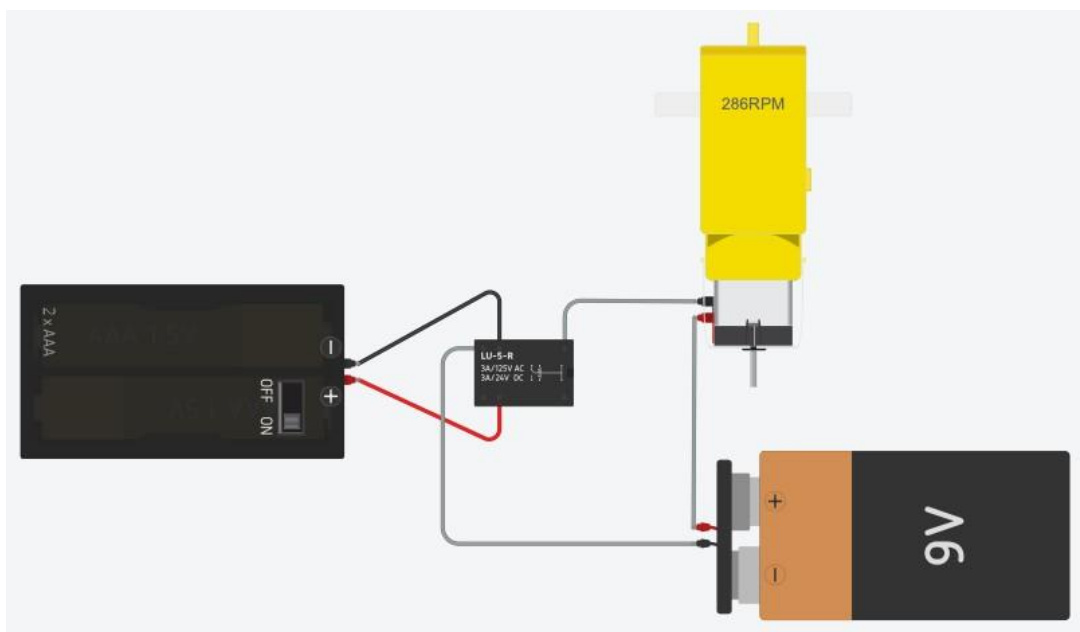
Razmotrimo jednostavan električni krug koji sadrži samo izvor, jedan otpornik, a **induktivnost**, i prekidač. Proučite što se događa sa strujom kada se prekidač uključi i isključi. Dodajte žarulju u krug i promatrajte kada će zasvijetliti.

Link na simulator: <https://www.falstad.com/circuit/e-induct.html>

Zadatak 3

Arelejje električni prekidač. Aktivira se strujom iz prvog električnog kruga kako bi se uključio ili isključio drugi električni krug. Kod rada s relejima obavezna je uporaba dva odvojena napajanja za dva odvojena strujna kruga.

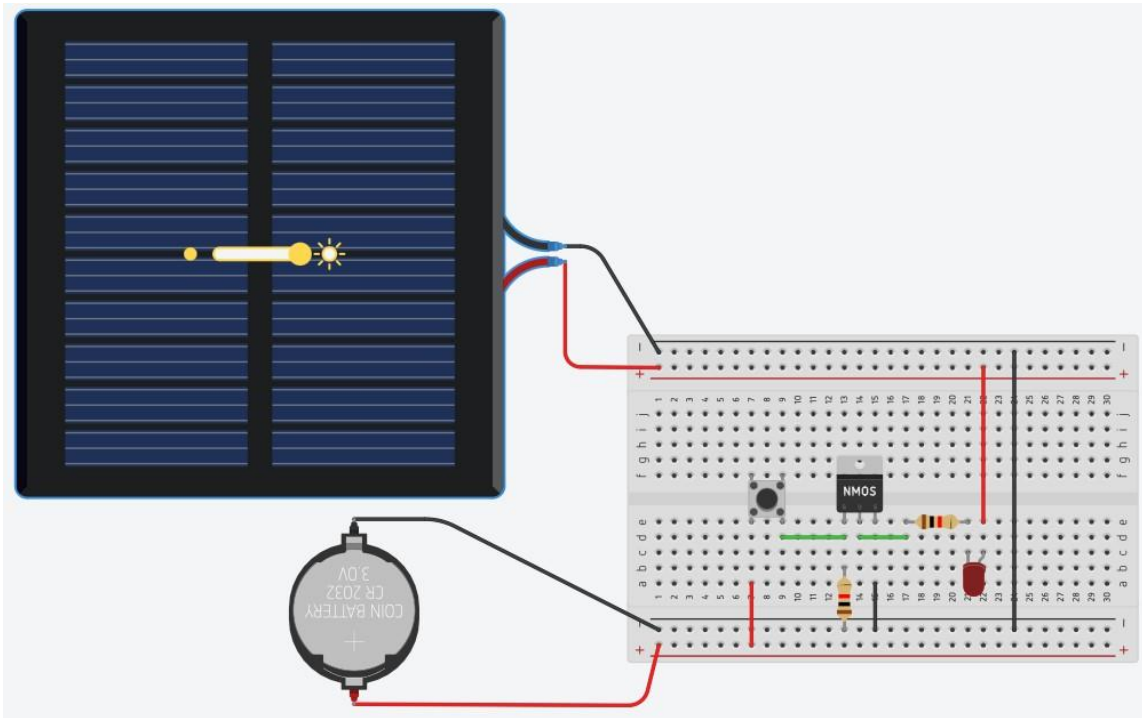
Spojite shemu s relejem u TINKERCAD-u prema slici i isprobajte kako relej radi.



Zadatak 4

An **NMOS tranzistorje** poluvodič koji se koristi kao elektronički prekidač.

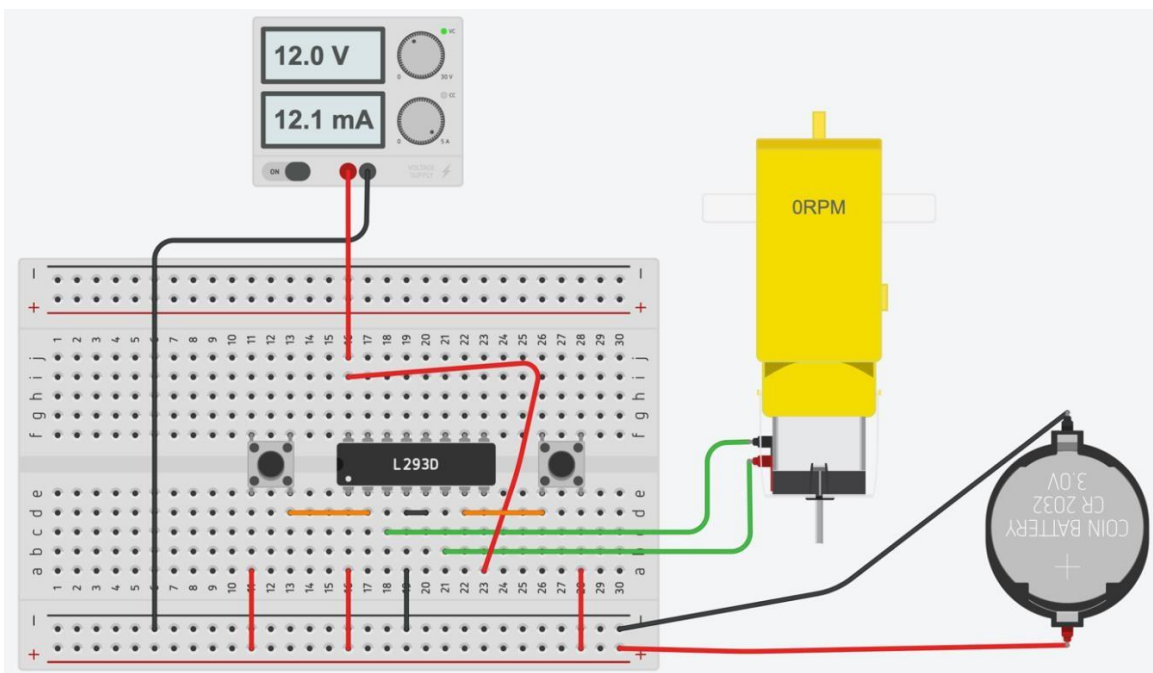
Spojite shemu s NMOS-om u TINKERCAD-u prema slici i isprobajte kako radi. Promijenite vrijednosti solarne ćelije i testirajte je. Promijenite vrijednosti otpornika i testirajte ti.



Zadatak 5

DC motori imaju dvije žice, negativnu (crnu) i pozitivnu (crvenu). Kada je pozitivni pol spojen na napajanje, a negativni na GND, osovina motora se okreće. Kada su žice spojene obrnuto, pozitivno na GND i negativno na napajanje, osovina motora će se ponovno okretati, ali u suprotnom smjeru. Kako ne bi morali ručno spajati žice pri svakoj promjeni smjera, koristi se H-bridge čip.

Spojite shemu s H-mostom i motorom u TINKERCAD-u prema slici i testirajte je. Pritisnite tipke i promatrajte u kojem se smjeru okreće osovina motora.



Broj pokusa: 4

4- Uvod u Arduino

Cilj eksperimenta

Cilj ovog eksperimenta je objasniti što je mikrokontroler i čemu služi. Polaznici će naučiti koristiti simulator za spajanje elektronike na Arduino.

Teorijska pozadina

Mikrokontroler je malo i jeftino računalo na jednom čipu koje se može koristiti za upravljanje funkcijama kućanskih i uredskih uređaja, robota, vozila, potrošačke elektronike itd. Najpoznatiji mikrokontroler je Arduino UNO.

Zadatak 1

U Tinkercad izborniku Starters odaberite Arduino. Otvorit će se preko 30 demo shema i programa s Arduinom i elektronikom.

Odaberite jedan demo program i odgovorite na pitanja.

1. Koji se elektronički element koristi?
2. Na koji pin Arduina je spojen element?
3. Pokrenite simulaciju. Što je program učinio?
4. Kopirajte shemu.
5. Otvorite kod. Kopirajte korišteni kod. Pokušajte povezati iste naredbe iz Blocks i Text verzija programa.

Zadatak 2

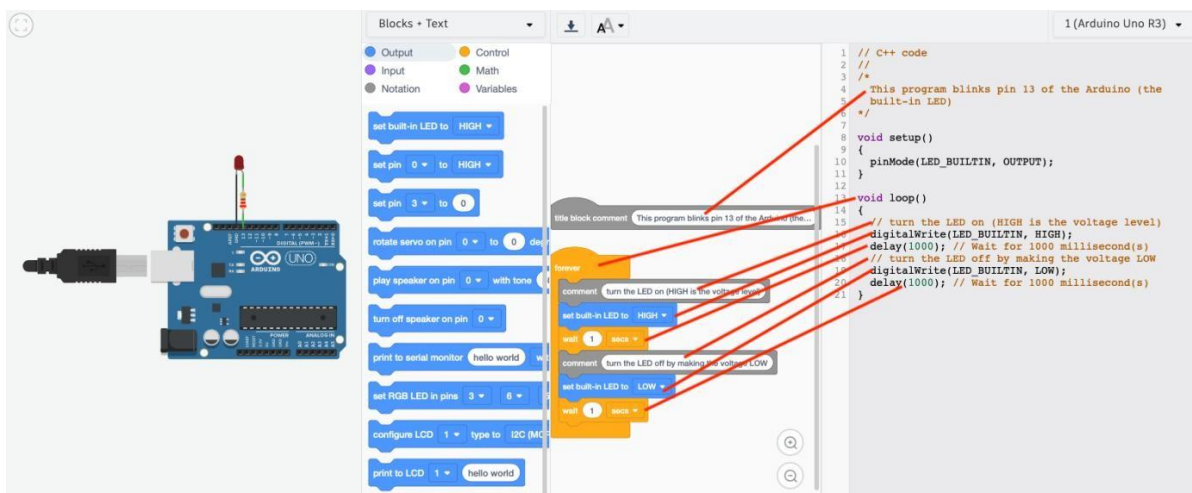
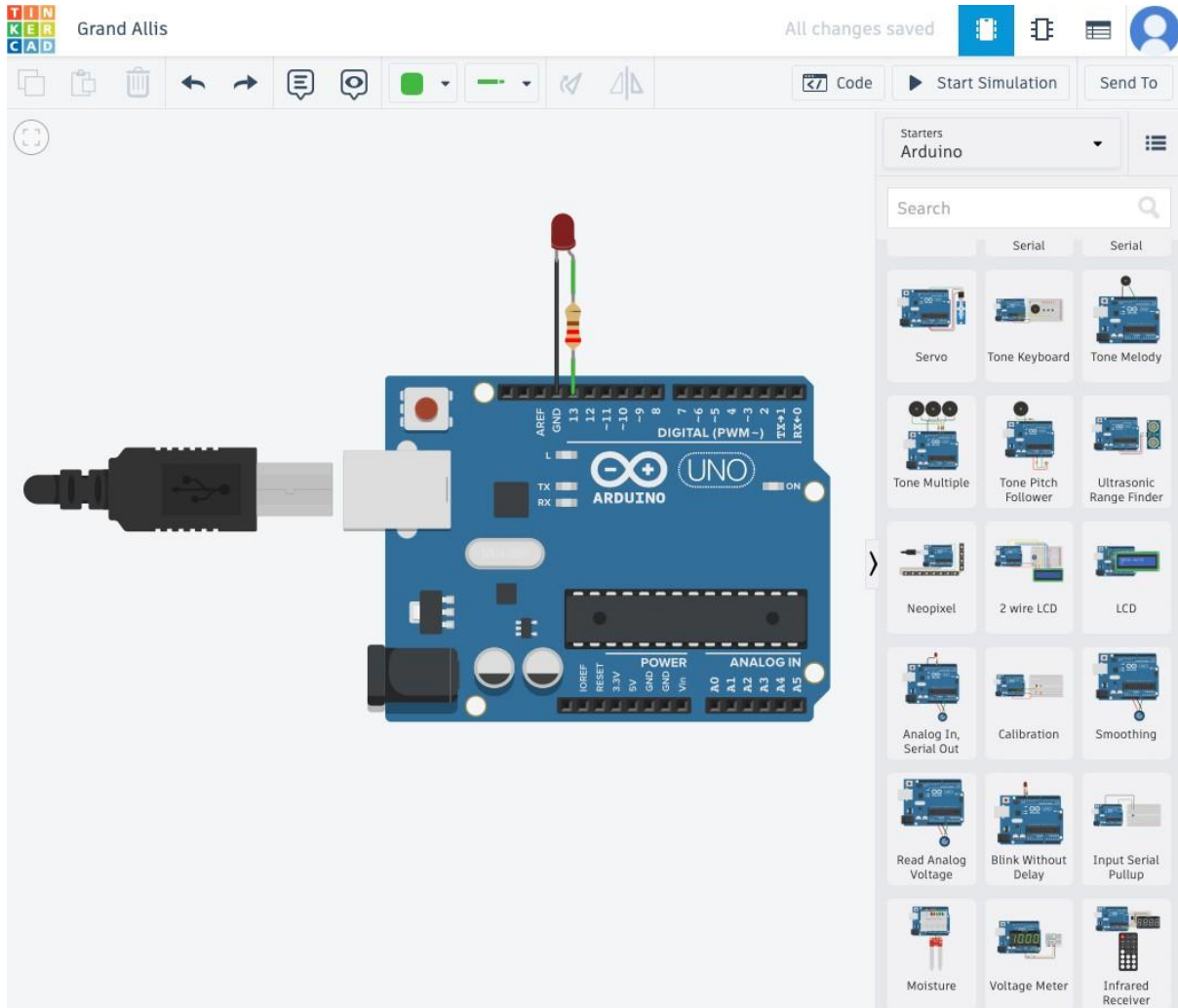
U Tinkercad izborniku Starters odaberite Arduino. Otvorit će se preko 30 demo shema i programa s Arduinom i elektronikom.

Odaberite jedan demo program i odgovorite na pitanja.

1. Koji se elektronički element koristi?
2. Na koji pin Arduina je spojen element?
3. Pokrenite simulaciju. Što je program učinio?
4. Kopirajte shemu.
5. Otvorite kod. Kopirajte korišteni kod. Pokušajte povezati iste naredbe iz Blocks i Text verzija programa.

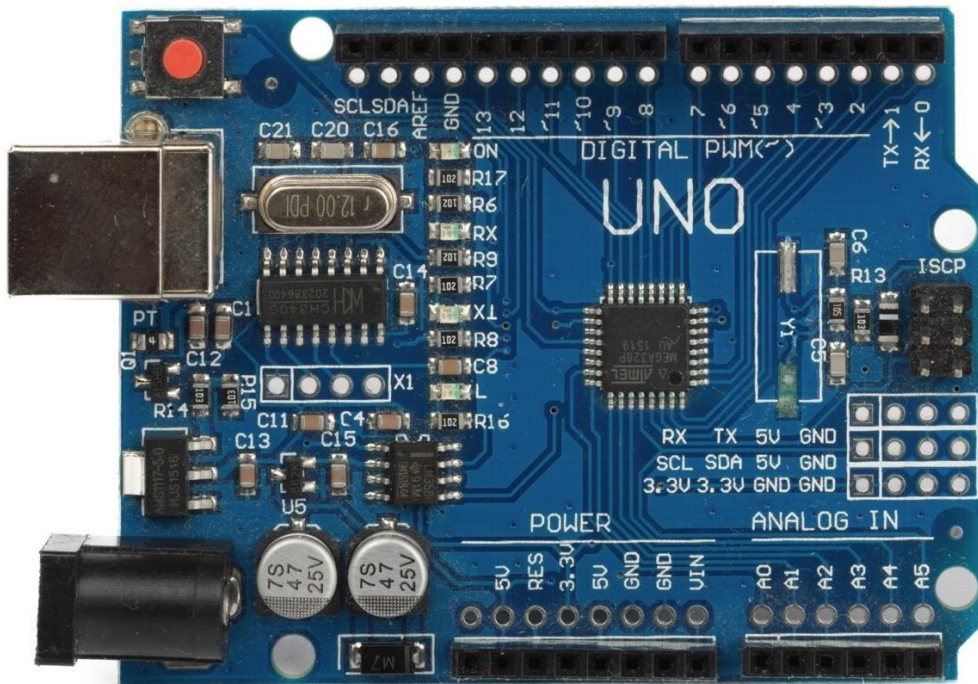
Moguće rješenje za zadatak 1 i zadatak 2

Jedan primjer je demo Blink. LED i otpornik spojeni su na pin 13. LED trepće svake 1 s.



Zadatak 3

Na fotografiji Arduino UNO označite najvažnije dijelove.



5-Arduino platforme

. Što je Arduino

Cilj eksperimenta

Cilj je koristiti Arduino ili sličnu tehnologiju za prikaz praktične primjene koja uključuje senzore, aktuatora ili druge komponente. Ovaj projekt ima za cilj riješiti problem ili izvršiti određeni zadatak, pokazujući kako se ove tehnologije mogu primijeniti u scenarijima stvarnog svijeta.

Teorijska pozadina

Temeljni koncepti uključuju elektroniku, programiranje i načela projektiranja sustava relevantna za eksperiment. Teme mogu pokrivati teoriju sklopova, digitalnu logiku, arhitekturu mikrokontrolera i korištenje C/C++ u razvoju Arduina, pružajući sveobuhvatno razumijevanje potrebno za uspješnu implementaciju projekta.

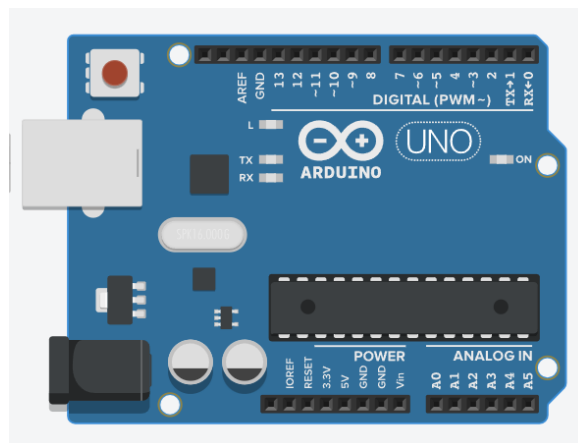
Zadatak 1

Napravite jednostavan sustav za nadzor temperature koristeći Arduino ploču kako biste bolje razumjeli njegovu funkcionalnost.

Potrebni materijali

Arduino UNO R3, Arduino IDE softver ili Tinkercad simulacijski portal.

Eksperimentalni krug



Slika 1 Arduino UNO R3 razvojna ploča

Spojte Arduino ploču na računalo putem USB A/B kabela i otvorite Arduino IDE. Odaberite priključak (Alati→Priključak) na koji je spojen Arduino. Otvorite zaslon serijskog monitora iz izbornika Alati.

Riješenje:

Koraci postupka

- 1-Postavite krug kako je opisano, osiguravajući da su svi spojevi sigurni.
- 2-Napišite i prenesite kod na Arduino.
- 3-Gledajte zaslon serijskog priključka.
- 4-Promatrajte promjene na ekranu.

Kodovi

// Definirajte ulazni pin na koji je spojen senzor temperature

```
int senzorPin = A0;
```

```
void setup() {
```

```
  Serial.begin(9600); // Pokretanje serijske komunikacije
```

```
}
```

```
void loop() {
```

```
  int čitanje = analogRead(senzorPin); // Očitajte vrijednost senzora
```

```
  napon plovka = očitanje * 5,0;
```

```
  napon /= 1024.0; // Pretvori to očitanje u napon
```

```
  temperatura plovka C = (napon - 0,5) * 100; // Pretvorite napon u temperaturu u Celzijusima
```

```
  Serial.print("Temperatura: ");
```

```
  Serial.print(temperatureC);
```

```
  Serial.println(" C");
```

```
  kašnjenje (1000); // Pričekajte sekundu prije ponavljanja petlje
```

```
}
```

Objašnjenje Kodeksa

Kod očitava izlaz temperaturnog senzora na Arduino, pretvara ga u napon, izračunava temperaturu u Celzijevim stupnjevima i neprekidno je prikazuje na serijskom monitoru.

6-Arduino programski jezik i editor-1

Cilj eksperimenta

Cilj eksperimenta s Arduinoom je napisati jednostavan program za kontrolu različitih LED dioda, paljenje i gašenje u nizu u određenim intervalima. Ovaj eksperiment ima za cilj podučiti osnovne programske konstrukcije Arduinoa i kako komunicirati s fizičkim svijetom kroz programiranje.

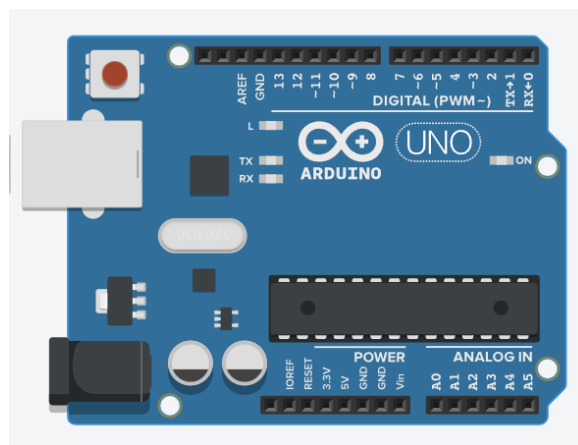
Teorijska pozadina

Programski jezik koji se koristi u Arduinou temelji se na Wiringu, prilagođenoj verziji C++ dizajniranoj za jednostavnu upotrebu u uobičajenim ulazno/izlaznim operacijama. Ključne točke o ovom programskom jeziku uključuju objektno orijentirano programiranje (OOP), prenosivost, visoke performanse, standardnu biblioteku, svestranost, programiranje s više paradigmi i upravljanje memorijom. Arduino programiranje vrti se oko funkcija `setup()` i `loop()`. `setup()` se poziva jednom na početku programa za postavljanje konfiguracija kao što su pin modovi, dok `loop()` sadrži glavnu programsku logiku koja se ponavlja tijekom vremena, kontrolirajući hardver poput LED dioda na temelju logike programa.

Potrebni materijali

Arduino UNO R3, Arduino IDE softver ili Tinkercad simulacijski portal.

Eksperimentalni krug



Slika 2 Arduino UNO R3 razvojna ploča

Spojite Arduino ploču na računalo putem USB A/B kabela i otvorite Arduino IDE. Odaberite priključak (Alati→Priključak) na koji je spojen Arduino. Otvorite zaslon serijskog monitora iz izbornika Alati.

Koraci postupka

- 1-Postavite krug kako je opisano, osiguravajući da su svi spojevi sigurni.
- 2-Napišite i prenesite kod na Arduino.
- 3-Gledajte zaslon serijskog priključka.
- 4-Promatrajte promjene na ekranu.

Kodovi

Prvi primjer koda: Led kontrola

```
int led1 = 4, led2 = 5, led3 = 6, led4 = 7;

void setup() {
  Serial.begin(9600);
  pinMode(led1, IZLAZ);
  pinMode(led2, IZLAZ);
  pinMode(led3, IZLAZ);
  pinMode(led4, IZLAZ);
}

void loop() {
  digitalWrite(led1, HIGH);

  kašnjenje (1000); // Pričekajte 1 sekundu
  digitalWrite(led1, LOW);

  kašnjenje (500); // Pričekajte 0,5 sekundi
  digitalWrite(led2, HIGH);

  kašnjenje (1000);
  digitalWrite(led2, LOW);

  kašnjenje (500);
  digitalWrite(led3, VISOKO);

  kašnjenje (1000);
  digitalWrite(led3, LOW);

  kašnjenje (500);
  digitalWrite(led4, VISOKO);

  kašnjenje (1000);
  digitalWrite(led4, LOW);

  kašnjenje (500);
}
```

Objašnjenje Kodeksa

Primjer koda pokazuje kako uzastopno uključiti i isključiti četiri LED diode spojene na Arduino, stvarajući jednostavan vizualni uzorak. Koristi osnovne funkcije kao što su pinMode(), digitalWrite() i delay() za kontrolu vremena i stanja svake LED diode.

Drugi primjer koda: Očitavanje vrijednosti senzora

```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  int SensorValue = analogRead(A0);  
  Serial.println(sensorValue);  
  kašnjenje(1); // Kratka odgoda za čitljivost  
}
```

Objašnjenje Kodeksa

Primjer koda fokusiran je na čitanje vrijednosti sa senzora spojenog na Arduino analogni pin A0 i ispis tih vrijednosti na serijski monitor. Prikazuje kako koristiti `analogRead()` i `Serial.println()` za prikupljanje podataka senzora i bilježenje.

7-Arduino programski jezik i editor-2

Cilj eksperimenta

Cilj je istražiti mogućnosti Arduina u sučelju s različitim sensorima i komponentama. To uključuje stvaranje projekata koji pokazuju kako se Arduino može koristiti za praktične primjene kao što su praćenje okoliša, robotika i interaktivne umjetničke instalacije.

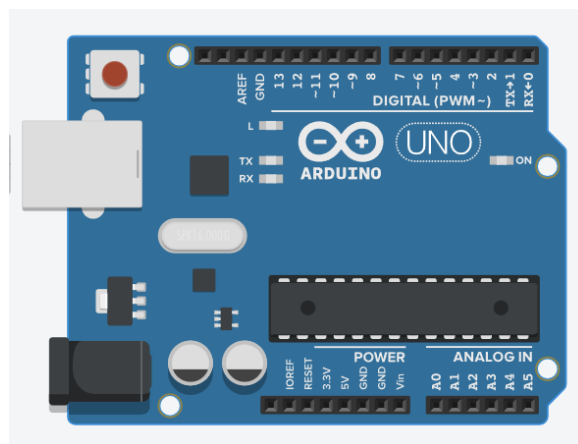
Teorijska pozadina

Teorijska osnova uključuje razumijevanje principa mikrokontrolera, posebno arhitekture i programiranja Arduina. Obuhvaća digitalni i analogni ulaz/izlaz, osnove električne energije i sklopova, programske konstrukcije (varijable, petlje, uvjete) i komunikacijske protokole kao što su Serial i I2C. Pozadina se također dotiče važnosti hardvera i softvera otvorenog koda u poticanju inovacija i obrazovanja u elektronici i programiranju.

Potrebni materijali

Arduino UNO R3, Arduino IDE softver ili Tinkercad simulacijski portal.

Eksperimentalni krug



Slika 3 Arduino UNO R3 razvojna ploča

Spojite Arduino ploču na računalo putem USB A/B kabela i otvorite Arduino IDE. Odaberite priključak (Alati→Priključak) na koji je spojen Arduino. Otvorite zaslon serijskog monitora iz izbornika Alati.

Zadatak 1.

Napravite projekt temeljen na Arduino koji demonstrira principe dizajna elektroničkih sklopova i programiranja pomoću Arduino IDE.

Rješenje 1.

Dizajniranje kruga koji je u interakciji s različitim sensorima i aktuatorima, programiranim s Arduinoom za obavljanje specifičnih funkcija, kao što je mjerenje uvjeta okoline ili upravljanje svjetlima ili motorima.

Broj pokusa: 8

8- Naredbe matematičkih operacija u Arduino IDE: Kako koristiti eksperiment s aritmetičkim operatorima

Cilj eksperimenta

Cilj ovog eksperimenta je pokazati kako se osnovni aritmetički operatori kao što su zbrajanje, oduzimanje, množenje, dijeljenje i ostatak koriste u Arduino integriranom razvojnom okruženju (IDE). Učenici će naučiti koristiti ove operatore i ispisati rezultat na zaslonu serijskog porta Arduino IDE.

Teorijska pozadina

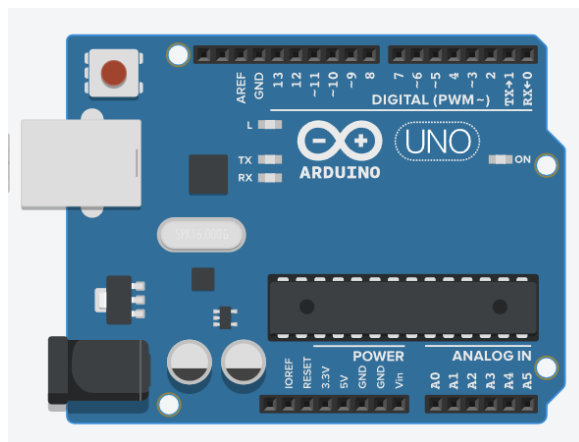
Svi programski jezici uključuju neke aritmetičke operatore, a Arduino IDE također uključuje osnovne aritmetičke operatore. Ovi se operatori mogu koristiti u Arduino za izvođenje nekih matematičkih operacija kao što je obrada nekih podataka senzora. Postoji 5 osnovnih aritmetičkih operacija koji se prikazuju u ovoj aplikaciji.

Operatori	Operater Simbol	Varijable	Primjer	Proizlaziti
Dodatak	+	Sen1 = 21	Rezultat = Sen1 + Sen2	24
Oduzimanje	-	Sen2 = 3	Rezultat = Sen1 - Sen2	18
Množenje	*	Rezultat = 0	Rezultat = Sen1 * Sen2	63
Podjela	/		Rezultat = Sen1 / Sen2	7
Ostatak	%		Rezultat = Sen1 % Sen2	0

Potrebni materijali

Arduino UNO R3, Arduino IDE softver ili Tinkercad simulacijski portal.

Eksperimentalni krug



Slika 4 Arduino UNO R3 razvojna ploča

Spojite Arduino ploču na računalo putem USB A/B kabela i otvorite Arduino IDE. Odaberite priključak (Alati→Priključak) na koji je spojen Arduino. Otvorite zaslon serijskog monitora iz izbornika Alati.

Koraci postupka

- 1-Postavite krug kako je opisano, osiguravajući da su svi spojevi sigurni.
- 2-Napišite i prenesite kod na Arduino.
- 3-Gledajte zaslon serijskog priključka.
- 4-Promatrajte promjene na ekranu.

Kodovi

```
/* Globalna definicija varijable */
intsenzor1=21; intsenzor 2=3; int
Proizlaziti=0; poništitipostaviti(){

Serijski.početi(9600);
//Dodatak
Proizlaziti=senzor1+senzor 2;
Serijski.ispisati("Dodavanje senzora1 i senzora2 je "); Serijski.println
(Proizlaziti); odgoditi(2000. godine);

//Oduzimanje
Proizlaziti=senzor1-senzor 2;
Serijski.ispisati("Oduzimanje senzora1 i senzora2 je "); Serijski.
println(Proizlaziti); odgoditi(2000. godine);

//Množenje
Proizlaziti=senzor1*senzor 2;
Serijski.ispisati("Množenje senzora1 i senzora2 je "); Serijski.println(
Proizlaziti); odgoditi(2000. godine);

//Podjela
Proizlaziti=senzor1/senzor 2;
Serijski.ispisati("Podjela senzora1 i senzora2 je "); Serijski.
println(Proizlaziti); odgoditi(2000. godine);

//Ostatak
Proizlaziti=senzor1%senzor 2;
Serijski.ispisati("Ostatak senzora1 i senzora2 je "); Serijski.println(
Proizlaziti); odgoditi(2000. godine);

}

void loop() {
}
```

Objašnjenje Kodeksa

Postaviti:Serijski monitor se aktivira brzinom od 9600 bauda. Za prethodno definirane varijable relevantne matematičke operacije izvode se samo jednom.

Petlja:-

Broj pokusa: 9

9- Naredbe matematičkih operacija u Arduino IDE: Kako koristiti aritmetičke operatore eksperiment 2

Cilj eksperimenta

Cilj ovog eksperimenta je pokazati kako se osnovni aritmetički operatori kao što su zbrajanje, oduzimanje, množenje, dijeljenje i ostatak koriste u Arduino integriranom razvojnom okruženju (IDE). Učenici će naučiti koristiti ove operatore s broječanim varijablama s pomičnim zarezom.

Teorijska pozadina

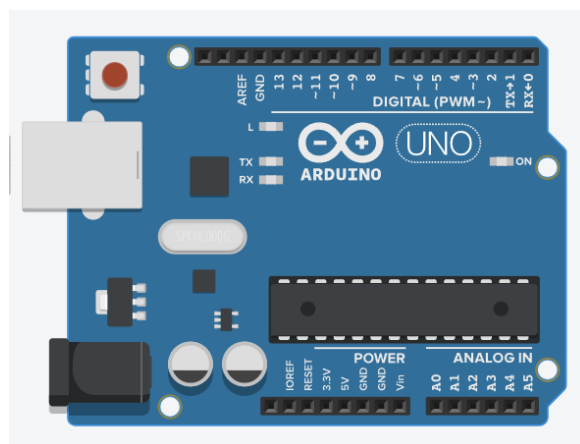
Ponašanje aritmetičkih operacija razlikuje se ovisno o vrsti korištene broječne varijable. Konkretno, operator dijeljenja može dati rezultat s pomičnim zarezom. Stoga, kada se operator dijeljenja koristi na cjelobrojnim varijablama, možda se neće dobiti očekivani rezultat. U senzorskim aplikacijama s Arduino pločama, obrađeni brojevi su uglavnom tipa s pomičnim zarezom. Osim toga, modulo operator (ostatak) je operator koji se može koristiti samo s cijelim brojevima.

Operatori	Operater Simbol	Varijable	Primjer	Proizlaziti
Dodatak	+	Sen1 = 24,5	Rezultat = Sen1 + Sen2	28.00 sati
Oduzimanje	-	Sen2 = 3,5	Rezultat = Sen1 - Sen2	21.00
Množenje	*	Rezultat = 0,0	Rezultat = Sen1 * Sen2	85.75
Podjela	/	Sen3 = 12	Rezultat = Sen1 / Sen2	7.00
Ostatak	%	Sen4 = 5	Rezultat = Sen3 % Sen4	2,00

Potrebni materijali

Arduino UNO R3, Arduino IDE softver ili Tinkercad simulacijski portal.

Eksperimentalni krug



Slika 5 Arduino UNO R3 razvojna ploča

Spojite Arduino ploču na računalo putem USB A/B kabela i otvorite Arduino IDE. Odaberite priključak (Alati→Priključak) na koji je spojen Arduino. Otvorite zaslon serijskog monitora iz izbornika Alati.

Koraci postupka

- 1-Postavite krug kako je opisano, osiguravajući da su svi spojevi sigurni.
- 2-Napišite i prenesite kod na Arduino.
- 3-Gledajte zaslon serijskog priključka.
- 4-Promatrajte promjene na ekranu.

Kodovi

```
//Globalna varijabla
plutatisenzor1=24.5;
plutatisenzor 2=3.5;
plutatiProizlaziti=0,0; int
sjeme=123; poništiti
postaviti()
{
  Serijski.početi(9600);

  //Operacija zbrajanja
  Proizlaziti=senzor1+senzor 2;
  Serijski.ispisati("Dodavanje senzora1 i senzora2 je "); Serijski.println
  (Proizlaziti); odgoditi(2000. godine);

  //Operacija oduzimanja
  Proizlaziti=senzor1-senzor 2;
  Serijski.ispisati("Oduzimanje senzora1 i senzora2 je "); Serijski.
  println(Proizlaziti); odgoditi(2000. godine);

  //Operacija množenja
  Proizlaziti=senzor1*senzor 2;
  Serijski.ispisati("Množenje senzora1 i senzora2 je "); Serijski.println(
  Proizlaziti); odgoditi(2000. godine);

  //Operacija dijeljenja
  Proizlaziti=senzor1/senzor 2;
  Serijski.ispisati("Podjela senzora1 i senzora2 je "); Serijski.
  println(Proizlaziti); odgoditi(2000. godine);

  intsenzor3=12,senzor 4=5; //
  Operacija ostatka (modulo).
  Proizlaziti=senzor3%senzor 4;
  Serijski.ispisati("Ostatak senzora3 i senzora4 je "); Serijski.println(
  Proizlaziti); odgoditi(2000. godine);

}

void petlja()
{
  randomSeed(seed);
  senzor1 = slučajan(1, 300);
```

```

senzor2 = slučajan(1, 300);
Serial.println(senzor1); Serial.println(senzor2);
sjeme = senzor1 * senzor2; Rezultat = senzor1 +
senzor2; Serial.print("Dodavanje podataka
senzora = "); Serial.println(Rezultat);

Rezultat = senzor1 - senzor2;
Serial.print("Oduzimanje podataka senzora = ");
Serial.println(Rezultat);
Rezultat = senzor1 * senzor2;
Serial.print("Umnožavanje podataka senzora = ");
Serial.println(Rezultat);
Rezultat = senzor1 / senzor2;
Serial.print("Podjela podataka senzora = ");
Serial.println(Rezultat);
Rezultat = (int)senzor1 % (int)senzor2;
Serial.print("Ostatak podataka senzora = ");
Serial.println(Rezultat);
}

```

Objašnjenje Kodeksa

Postaviti: Serijski monitor se aktivira brzinom od 9600 bauda. Za prethodno definirane varijable relevantne matematičke operacije izvode se samo jednom.

Petlja: Dizajnirana je jednostavna igra simulacije. Za to se koriste random() i randomSeed() ugrađene metode Arduino editora. Kada se svaka funkcija petlje pokrene, podaci senzora 1 i senzora 2 ponovno se proizvode nasumično između zadanog intervala. U svakom procesu ponovno se izvodi 5 matematičkih operacija.

Broj pokusa: 10

10- Kontrolne strukture u Arduino IDE: Kako koristiti kontrolne strukture-1

Cilj eksperimenta

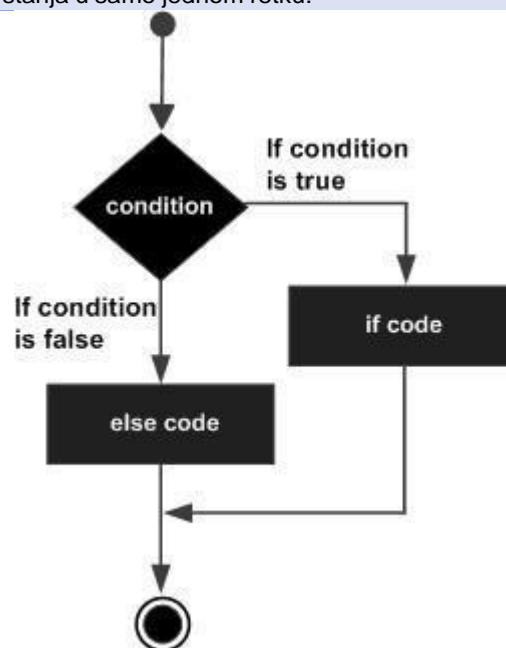
Cilj ovog eksperimenta je pokazati kako se kontrolne strukture koriste u Arduino integriranom razvojnom okruženju (IDE). Učenici će učiti pomoću **operatori usporedbe** **if izjave** ("ako", "ako ... drugo", "ako ... inače ako ... inače").

Teorijska pozadina

Svi programski jezici imaju neke vrste kontrolnih iskaza za donošenje odluke o određenim uvjetima. Na primjer, koristimo ove kontrolne izjave kada odlučujemo o operaciji koju treba izvesti na temelju određene vrijednosti temperature ili kada odlučujemo o operaciji koju treba izvesti na temelju ocjene koju učenik dobije.

Kao i u drugim jezicima, postoje različite kontrolne strukture koje koristimo u Arduino IDE. Ove kontrolne strukture i izjave mogu se navesti na sljedeći način:

Kontrolna izjava	Opis
izjava "ako".	Potreban je izraz u zagradi i izjava ili blok izjava. Ako je izraz istinit, tada se naredba ili blok naredbi izvršava, inače se ove naredbe preskaču.
izjava "if... else".	Anakoizjava može biti praćena izbornim drugoiskaz, koji se izvršava kada je izraz lažan.
izjava "if ... else if ... else".	Theakoizjava može biti praćena izbornim inače akoizjava, koja se može koristiti za testiranje različitih uvjeta.
izjava "zamijeni slučaj".	Slično kao iakoizjave, prekidač...kućištekontrolira tijek programa dopuštajući programerima da specificiraju različite blokove koji se trebaju izvršiti u različitim uvjetima.
Uvjetni operator "?:"	Ovo je ternarni operator koji korisniku omogućuje usporedbu dvaju stanja u samo jednom retku.



Slika 6 Dijagram toka if naredbi

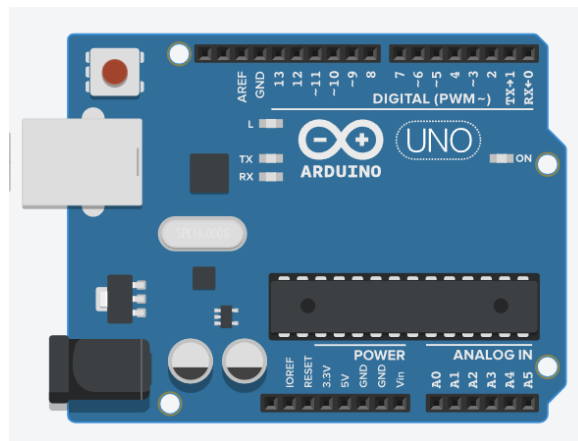
Neki se operatori usporedbe koriste za usporedbu dviju vrijednosti kada se pišu kontrolne izjave. Kao rezultat usporedbe, logičan izlaz **(1)** ili **lažno (0)** dobiva se. Pretpostavimo da varijabla portB ima 63, a portC 127,

Operater Ime	Operater simbol	Opis	Primjer
jednak	==	Provjerava je li vrijednost dvaju operanda jednaka ili nije, ako da, tada uvjet postaje istinit.	(portB == portC) jelažno
nejednak do	!=	Provjerava je li vrijednost dvaju operanda jednaka ili ne, ako vrijednosti nisu jednake tada uvjet postaje istinit.	(portB != portC) jepravi
manje od	<	Provjerava je li vrijednost lijevog operanda manja od vrijednosti desnog operanda, ako da, tada uvjet postaje istinit.	(portB < portC) jepravi
veća od	>	Provjerava je li vrijednost lijevog operanda veća od vrijednosti desnog operanda, ako da, tada uvjet postaje istinit.	(portB > portC) jelažno
manje od ili jednako	<=	Provjerava je li vrijednost lijevog operanda manja ili jednaka vrijednosti desnog operanda, ako da, tada uvjet postaje istinit.	(portB <= portC) jepravi
veća nego ili jednak	>=	Provjerava je li vrijednost lijevog operanda veća ili jednaka vrijednosti desnog operanda, ako da, tada uvjet postaje istinit.	(portB >= portC) jelažno

Potrebni materijali

Arduino UNO R3, Arduino IDE softver ili Tinkercad simulacijski portal.

Eksperimentalni krug



Slika 7 Arduino UNO R3 razvojna ploča

Spojite Arduino ploču na računalo putem USB A/B kabela i otvorite Arduino IDE. Odaberite priključak (Alati→Priključak) na koji je spojen Arduino. Otvorite zaslon serijskog monitora iz izbornika Alati.

Koraci postupka

- 1- Postavite krug kako je opisano, osiguravajući da su svi spojevi sigurni.
- 2- Napišite i prenesite kod na Arduino.
- 3- Gledajte zaslon serijskog priključka.

4-Promatrajte promjene na ekranu.

Kodovi

```
/* Globalna definicija varijable
*/ int portB=63; int portC=127;

poništi()postaviti() {
    Serijski.poceti(9600);
    Serijski.println(portB==portC);odgoditi(1000);
    Serijski.println(portB!=portC);odgoditi(1000);
    Serijski.println(portB<portC);odgoditi(1000);
    Serijski.println(portB>portC);odgoditi(1000);
    Serijski.println(portB<=portC);odgoditi(1000);
    Serijski.println(portB>=portC);
}

poništi()petlja() {
    /* ako je blok */
    ako(portB<portC)/*ako je uvjet istinit, izvršite sljedeću naredbu*/
    portB++;
    odgoditi(200);
    Serijski.ispisati("PORTB= ");Serijski.println(portB);

    /* if ... else blok */
    ako(portB==portC){
        Serijski.println("PortB je jednak PortC");
        portB++;
    }
    drugo{
        Serijski.println("PortB nije jednak PortC");
    }

    /* if ... else if ... else blok */
    ako(portB>portC){
        Serijski.println("PortB je veći od PortC");
    }
    inače ako(portB<portC){
        Serijski.println("PortB je manji od PortC");
    }
    drugo{
        Serial.println("PortB je jednak PortC");
    }
}
}
```

Objašnjenje Kodeksa

Postaviti:Serijski monitor se aktivira brzinom od 9600 bauda. Za prethodno definirane varijable relevantne operacije usporedbe izvode se samo jednom.

Petlja:Beskonačna petlja koja kontinuirano upravlja blokovima koda unutar.

Broj pokusa: 11

11- Kontrolne strukture u Arduino IDE: Kako koristiti kontrolne strukture-2

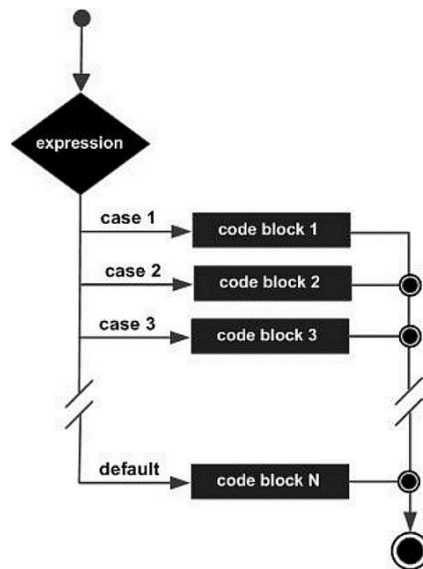
Cilj eksperimenta

Cilj ovog eksperimenta je pokazati kako se kontrolne strukture koriste u Arduino integriranom razvojnom okruženju (IDE). Učenici će učiti pomoću **prebaciti kućište** izjava i **uvjetni operator** "?:".

Teorijska pozadina

Slično if naredbama, **prekidač...kućište** kontrolira tijekom programa dopuštajući programerima da specificiraju različite kodove koji bi se trebali izvršiti u različitim uvjetima. Posebno, **asklopka** iskaz uspoređuje vrijednost varijable s vrijednostima navedenim **uslužaj** izjave.

Apauza ključna riječ je napisana na kraju svake **slučaj** izjava. Inače, svaki **isklopka** izjava se izvodi bez obzira na uvjet **uslužaj** izjava. **Azadan** ključna riječ se koristi za pokretanje ako nema podudaranja velikih i malih slova u prekidaču.

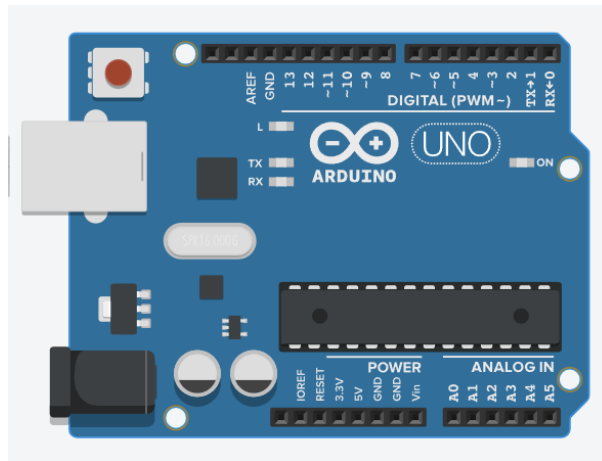


Slika 8. Dijagram toka iskaza slučaja prekidača

Potrebni materijali

Arduino UNO R3, Arduino IDE softver ili Tinkercad simulacijski portal.

Eksperimentalni krug



Slika 9 Arduino UNO R3 razvojna ploča

Spojite Arduino ploču na računalo putem USB A/B kabela i otvorite Arduino IDE. Odaberite priključak (Alati→Priključak) na koji je spojen Arduino. Otvorite zaslon serijskog monitora iz izbornika Alati.

Koraci postupka

- 1-Postavite krug kako je opisano, osiguravajući da su svi spojevi sigurni.
- 2-Napišite i prenesite kod na Arduino.
- 3-Gledajte zaslon serijskog priključka.
- 4-Promatrajte promjene na ekranu.

Kodovi - 1

```

/* Globalna definicija varijable */
byte portC=1; bool lijevo=pravi;
poništitipostaviti(){

  Serijski.početi(9600);
}
poništitipetlja(){
  sklopka(portC) {
    slučaj1:Serijski.println("Prva LED je uključena.");pauza; slučaj2:
    Serijski.println("Drugi LED je uključen.");pauza; slučaj4:Serijski.
    println("Treći LED je uključen.");pauza; slučaj8:Serijski.println(
    "Četvrta LED je uključena.");pauza; slučaj16:Serijski.println(
    "Peta LED je uključena.");pauza; slučaj32:Serijski.println("Šesti
    LED je uključen.");pauza; slučaj64:Serijski.println("Sedma LED je
    uključena.");pauza; slučaj128:Serijski.println("Osma LED je
    uključena.");pauza; zadano:Serijski.println("Nevažne stanje!");
  }
  ako(lijevo)
    portC=portC<<1;//Operator pomaka bita. Svi bitovi se jednom pomaknu lijevo. drugo

    portC=portC>>1;//Operator pomaka bitova. Svi bitovi su jednom pomaknuti
    udesno. ako(portC==0&&lijevo){//Višestruka usporedba s if blokom
    portC=128;
    lijevo=lažno;
    Serijski.println("*****");
  }
  inače ako(portC==0&& !lijevo){//Višestruka usporedba s else if blokom
    portC=1;
  }

```

```

    lijevo=pravi;
    Serijski.println("*****");
}
odgoditi(1000);
}

```

Objašnjenje Kodeksa

Postaviti:Serijski monitor se aktivira brzinom od 9600 bauda.

Petlja:Upravlja se blokom kućišta prekidača. Booleova varijabla se mijenja prema maksimalnoj ili minimalnoj vrijednosti bita, tako da se skeniranje kontinuirano vrši slijeva na desno ili zdesna nalijevo.

Kodovi – 2

```

/* Globalna definicija varijable
*/ intportB=63; intportC=127;
charc='k';

poništipostaviti() {
    Serijski.početi(9600);

    /* Pronađite max(portB, portC): */
    Serijski.println((portB>portC)?portB:portC);/*Vrijednost portB se ispisuje ako je veća od portC. Inače se
    ispisuje vrijednost portC.*/

    /* Pretvori mala slova u velika: */
    c= (c>='a' && c<='z') ? (c-32) :c; Serijski.
    println(c);
}

poništipetlja() {
}

```

Objašnjenje Kodeksa

Postaviti:Serijski monitor se aktivira brzinom od 9600 bauda. Uvjetni operator koristi se za određivanje koji je unos veći od drugog. Pretvaranje zadanog malog slova u veliko u skladu s vrijednošću tablice znakova ASCII također se lako izvodi pomoću uvjetnog operatora.

Petlja:-

Broj pokusa: 12

12- Kontrolne strukture u Arduino IDE: Kako koristiti kontrolne strukture-3

Cilj eksperimenta

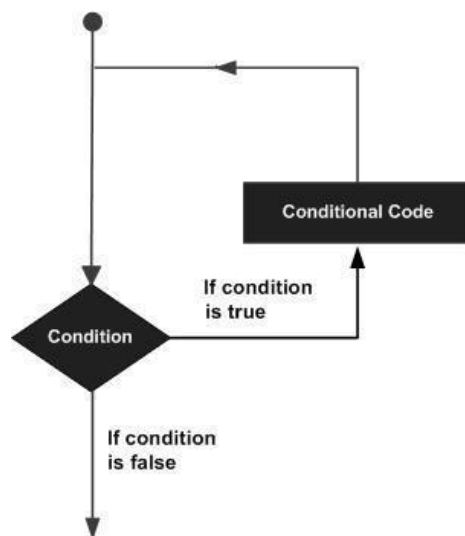
Cilj ovog eksperimenta je pokazati kako se kontrolne strukture koriste u Arduino integriranom razvojnom okruženju (IDE). Učenici će učiti pomoću **za, dokučiniti dok** petlje.

Teorijska pozadina

The **za** izjava se koristi za ponavljanje bloka izjava u vitičastim zagradama. Brojač povećanja obično se koristi za povećanje i prekid petlje. The **za** naredba je korisna za sve operacije koje se ponavljaju. Ova struktura se često preferira, posebno u skeniranju priključaka Arduino ploča. Postoje tri dijela **za** zaglavlje petlje:

```
za (inicijalizacija; stanje; prirast) {  
  //izjava(e);  
}
```

“**dok**” petlje će se petljati kontinuirano i beskonačno, sve dok izraz unutar zagrade, () ne postane lažan. Petlja se izvodi sve dok je uvjet ispunjen. Iz tog razloga treba ga pažljivo koristiti kako bi se izbjegla beskonačna petlja.



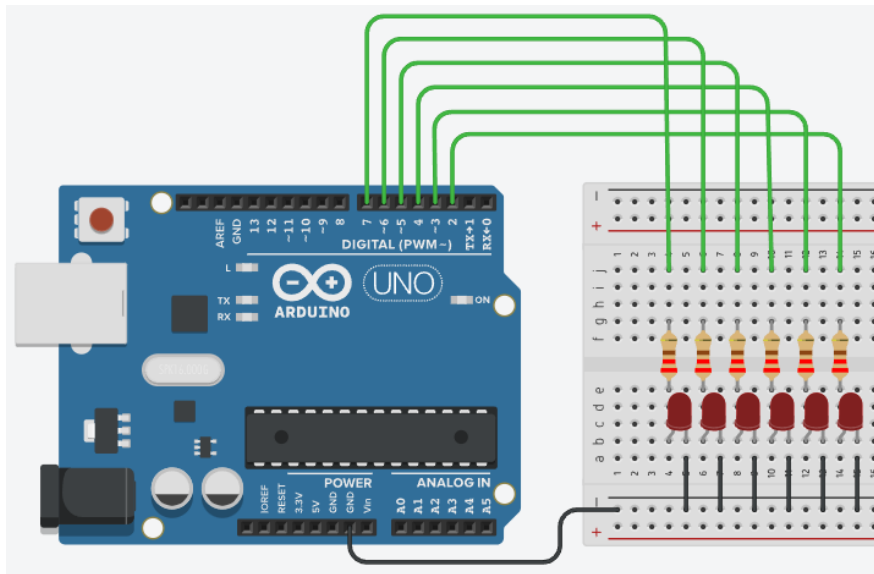
Slika 10 Rad petlji

“**čini...dok**” petlja je slična **dok** petlji, ali se izvodi barem jednom, bez obzira je li uvjet ispunjen ili ne.

Potrebni materijali

Arduino UNO R3, 6 komada otpornika od 220 Ω, 6 LED dioda, Arduino IDE softver ili Tinkercad simulacijski portal.

Eksperimentalni krug



Slika 11 Arduino UNO R3 razvojna ploča

Spojite Arduino ploču na računalo putem USB A/B kabela i otvorite Arduino IDE. Odaberite priključak (Alati→Priključak) na koji je spojen Arduino. Otvorite zaslon serijskog monitora iz izbornika Alati.

Koraci postupka

- 1- Postavite krug kako je opisano, osiguravajući da su svi spojevi sigurni.
- 2- Napišite i prenesite kod na Arduino.
- 3- Promatrajte ponašanje LED dioda.
- 4- Promatrajte promjene na zaslonu serijskog priključka.

Kodovi

```
// Globalne varijable
// definirani su nizovi s dvije dimenzije int
senzori01[3][3]={3,-7,7,8,5,-4,2,3,0}; int
senzori02[3][3]={4,3,0,2,-1,-4,8,9,5}; int
senzor3=50; int senzor 4=100; int brojač=1;
poništi postaviti() {

  Serijski.početi(9600);
}

poništi petlja() {
  PORTD=4;
  odgoditi(250);
  // for počinje petlja za(
  int ja=0; ja<6; ja++){
    PORTD=PORTD<<1;
    odgoditi(250);
  }
  PORTD=128;
  // drugo za početak petlje
  za(int ja=0; ja<6; ja++){
    PORTD=PORTD>>1;
  }
}
```

```

    odgoditi(250);
}

//ugniježdene for petlje
za(int ja=0; ja<=2; ja++){//i varijabla
  za(int j=0; j<=2; j++){// j varijabla
    Serijski.ispisi(senzori01[ja][j] +senzori02[ja][j]); Serijski
    .ispisi("\t");//operator kartice
  }
  Serijski.println();//Povrat prtljage
}

// while petlja
dok(senzor3<senzor 4){
  Serijski.ispisi("Trčim ");
  Serijski.ispisi(brojač);
  Serijski.println(". puta.");
  brojač++;
  senzor3+=10;//unarni operator zbrajanja
  odgoditi(200);
}

//do...while petlja
čini{
  odgoditi(200);
  Serijski.println("Bar jednom trčim...");
}; }dok(senzor3>senzor 4);
}

```

Objašnjenje Kodeksa

Postaviti:Serijski monitor se aktivira brzinom od 9600 bauda.

Petlja:Dvazapetlje se pokreću za pokretanje LED dioda na PORTD Arduino ploče. Ugniježdеноzapokreće se petlja da pokaže kako skenirati dvodimenzionalne nizove. "**dok**" i "**učiniti...dok**" pokreću se petlje za testiranje usporedbe dviju vrijednosti.

Broj pokusa: 13

Naziv eksperimenta: Kontrolne strukture u Arduino IDE: Kako koristiti kontrolne strukture 4

Cilj eksperimenta

Cilj ovog eksperimenta je pokazati kako se korisnički definirane i ugrađene funkcije koriste u Arduino integriranom razvojnom okruženju (IDE). Studenti će naučiti deklarirati korisnički definirane funkcije i koristiti neke važne ugrađene funkcije.

Teorijska pozadina

Kao što ste do sada iskusili, koristili smo neke osnovne ugrađene funkcije u Arduino IDE kao što su `setup()`, `loop()` i `delay()`. Funkcija omogućuje korisniku strukturiranje programa u segmentima koda za obavljanje pojedinačnih zadataka. Arduino razvojno okruženje zahtijeva dvije glavne funkcije `setup()` i `loop()`.

Najčešća sintaksa za definiranje funkcije je sljedeća:

```
return_type naziv_funkcije(argument1, argument2, ...) {  
    izjave;  
    povratak return_type_value;  
}
```

return_type je tip podataka kao što je, `integer`, `float`, `char`, i tako dalje. Ne zahtijeva svaka funkcija vraćanje vrijednosti. U takvom slučaju, *return_type* vrijednost funkcije je definirana kao **poništi**. Argumenti su bilo koja varijabla ili vrijednost s odgovarajućim tipom podataka. Tip podataka argumenta mora odgovarati definiranom tipu.

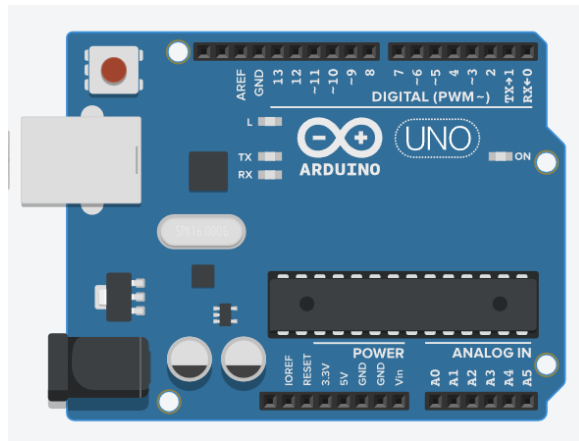
Funkcija je deklarirana izvan bilo koje druge funkcije, iznad ili ispod *petlji* postaviti funkcije. Ako samo napišete samu funkciju, tada je morate deklarirati prije *petlji* postaviti funkcije (gdje se zove).

Ako želite deklarirati funkciju nakon postavljanja ili funkcije petlje, tada morate napisati a **prototip funkcije** iznad ostalih funkcija gdje se poziva. Prototip funkcije mora slijediti točka-zarez (;).

Potrebni materijali

Arduino UNO R3, Arduino IDE softver ili Tinkercad simulacijski portal.

Eksperimentalni krug



Slika 12 Arduino UNO R3 razvojna ploča

Spojite Arduino ploču na računalo putem USB A/B kabela i otvorite Arduino IDE. Odaberite priključak (Alati→Priključak) na koji je spojen Arduino. Otvorite zaslon serijskog monitora iz izbornika Alati.

Koraci postupka

- 1- Postavite krug kako je opisano, osiguravajući da su svi spojevi sigurni.
- 2- Napišite i prenesite kod na Arduino.
- 3- Gledajte zaslon serijskog priključka.
- 4- Promatrajte promjene na ekranu.

Kodovi

```
// Globalne varijable
int sum_func(int x, int g); // korisnički definiran prototip funkcije int
senzor1=21; int senzor2=45;

// Deklaracija korisnički definirane funkcije prije njenog poziva
poništi printMessage(){
  Serijski.println("Ova je funkcija deklarirana iznad funkcije setup()");
  Serijski.ispisi("Razlika senzora je "); Serijski.println(senzor1-senzor2);
};
}

poništi postavi() {
  Serijski.početi(9600);
  printMessage();//poziva se korisnički definirana funkcija
}

poništi petlja() {
  ako (sum_func(senzor1, senzor2) >= 50) {
    Serijski.ispisi("Zbroj senzora je, "); Serijski.println(sum_func(senzor1, senzor2));
  }
  drugo {
    Serijski.println("Zbroj senzora je manji od 50");
  }
  senzor2=25;
  odgoditi(1000); //Podaci senzora čitaju se svake sekunde.
```

```
}  
  
// Dekleracija funkcije nakon loop() gdje je pozvana.  
int sum_func(int x, int g){  
    int z=0;  
    z=x+g;  
    povratakz;  
}
```

Objašnjenje Kodeksa

Postaviti: Serijski monitor se aktivira brzinom od 9600 bauda. Poziva se korisnički definirana funkcija.

Petlja: Poziva se korisnički definirana funkcija zbrajanja. Skeniranje se izvodi svake sekunde.

Broj pokusa: 14

14- Stringovi u Arduino IDE: Kako koristiti string literale

Cilj eksperimenta

Cilj ovog eksperimenta je pokazati kako se tip podataka niza koristi u Arduino integriranom razvojnom okruženju (IDE). Studenti će naučiti različite metode deklariranja tipa podataka string.

Teorijska pozadina

Kao i u svim drugim programskim jezicima, također u Arduino IDE postoji važan tip podataka koji se zove "string" na kojem se ne mogu izvoditi matematičke operacije. Nizovi su tekstualni tip podataka. Tekstualni nizovi mogu se prikazati na dva načina. Možete koristiti **Niz** tip podataka ili možete napraviti niz od niza tipa **char** završiti ga nultom ('\0').

- Sintaksa niza

Sve sljedeće valjane su deklaracije za nizove.

```
char Str1[10];
char Str2[8] = {'a', 'r', 'd', 'u', 'i', 'n', 'o'}; char
Str3[8] = {'a', 'r', 'd', 'u', 'i', 'n', 'o', '\0'}; char
Str4[] = "arduino";
char Str5[7] = "arduino";
char Str6[10] = "arduino";
```

- Mogućnosti deklariranja stringova

- Deklarirajte niz znakova bez inicijalizacije kao u Str1
- Deklarirajte niz znakova (s jednim dodatnim znakom) i kompajler će dodati traženi nulti znak, kao u Str2
- Eksplicitno dodajte nulti znak, Str3
- Inicijalizirati konstantom niza u navodnicima; kompajler će odrediti veličinu niza da stane u konstantu niza i završni nulti znak, Str4
- Inicijalizirajte polje eksplicitnom veličinom i konstantom niza, Str5
- Inicijalizirajte niz, ostavljajući dodatni prostor za veći niz, Str6

- Nulti raskid

Općenito, nizovi se završavaju nultim znakom (ASCII kod 0). To omogućuje funkcije (kao što je **Serial.print()**) da kaže gdje je kraj niza. Inače bi nastavili čitati sljedeće bajtove memorije koji zapravo nisu dio niza.

- Jednostruki navodnici ili dvostruki navodnici

Nizovi su uvijek definirani unutar dvostrukih navodnika ("Arduino"), a znakovi su uvijek definirani unutar jednostrukih navodnika ('A').

- Omotavanje dugih žica

Duge žice možete omotati ovako:

```
char myString[] = "Ovo je prvi red"
                  "ovo je drugi red"
                  "ovo je zadnji red";
```

- Niz nizova

String nizovi se koriste kada se radi s velikim količinama tekstova poruka, kao što su projekti s LCD zaslonima. Možete ispisati iste poruke u različitim dijelovima projekta. U takvim slučajevima bilo bi logično pripremiti niz poruka.

```
char*sensorStrings[] = {"Ovo je prvi senzor", "Ovo je  
drugi senzor", "Ovo je treći senzor", "Ovo je četvrti senzor",  
"Ovo je peti senzor", "Ovo je šesti senzor6"  
};
```

U donjem kodu, zvjezdica nakon tipa podataka char "**char ***" označava da je ovo niz "pokazivača".

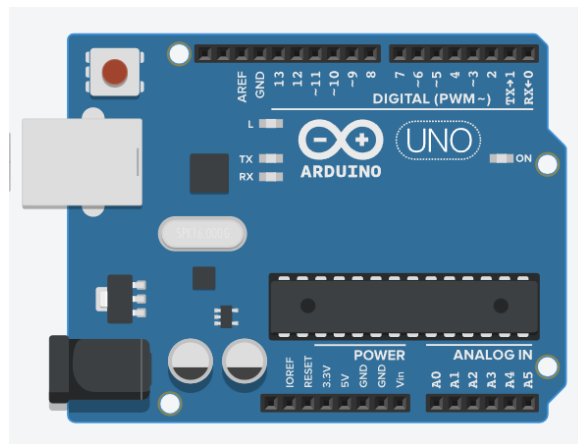
- Ulančavanje nizova

Nizovi stvoreni pomoću **char** tip podataka ne može se spojiti s aritmetičkim operatorom. **Niz** struktura podataka mora se koristiti za ovu operaciju.

Potrebni materijali

Arduino UNO R3, Arduino IDE softver ili Tinkercad simulacijski portal.

Ekperimentalni krug



Slika 13 Arduino UNO R3 razvojna ploča

Spojite Arduino ploču na računalo putem USB A/B kabela i otvorite Arduino IDE. Odaberite priključak (Alati→Priključak) na koji je spojen Arduino. Otvorite zaslon serijskog monitora iz izbornika Alati.

Koraci postupka

- 1- Postavite krug kako je opisano, osiguravajući da su svi spojevi sigurni.
- 2- Napišite i prenesite kod na Arduino.
- 3- Gledajte zaslon serijskog priključka.
- 4- Promatrajte promjene na ekranu.

Kodovi

```
// Globalne varijable  
char Str1[10]; //inicijalizirano, ali bez prve dodjele vrijednosti  
char Str2[8] = {'a','r','d','t','i','j','a','n','o'}; //inicijalizirano s prvom dodjelom vrijednosti, nulti završetak se dodaje automatski */
```

```
charStr3[8] = {'a','r','d','t','j','a','n','o','\0'}; /*inicijalizirano s prvom dodjelom vrijednosti, završetak nule dodaje se ručno */
```

```
charStr4[] = "arduino"; //veličina niza stvara se dodijeljenom vrijednošću.
```

```
charStr5[8] = "arduino"; //veličina polja i dodjela se rade zajedno. charStr6[
```

```
10] = "arduino"; //veličina polja je veća od dodijeljene vrijednosti. char
```

```
mojNiz[] = "Ovo je prvi red" "ovo je drugi red"
```

```
"ovo je zadnji red";
```

```
char*senzorStrings[] = {"Ovo je prvi senzor", "Ovo je drugi senzor", "Ovo je treći senzor", "Ovo je četvrti senzor", "Ovo je peti senzor", "Ovo je šesti senzor6"
```

```
};
```

```
poništi()postaviti() {
```

```
  Serijski.poceti(9600);
```

```
  Serijski.println(Str1);odgoditi(500);
```

```
  Serijski.println(Str2);odgoditi(500);
```

```
  Serijski.println(Str3);odgoditi(500);
```

```
  Serijski.println(Str4);odgoditi(500);
```

```
  Serijski.println(Str5);odgoditi(500);
```

```
  Serijski.println(Str6);odgoditi(500);
```

```
  Serijski.println(mojNiz);
```

```
}
```

```
poništi()petlja() {
```

```
  za(intja=0;ja<6;ja++) {
```

```
    Serijski.println(senzorStrings[ja]);
```

```
    odgoditi(500);
```

```
  }
```

```
}
```

Objašnjenje Kodeksa

Postaviti:Serijski monitor se aktivira brzinom od 9600 bauda. Ispisuju se različiti string literal.

Petlja:Niz nizova ispisuje se tehnikom skeniranja pomoću for petlje.

Broj pokusa: 15

15- Stringovi u Arduino IDE: Kako koristiti strukturu podataka String

Cilj eksperimenta

Cilj ovog eksperimenta je pokazati kako se String() objekt (struktura podataka) koristi u Arduino integriranom razvojnom okruženju (IDE). Učenici će naučiti različite metode deklaracije String() objekta.

Teorijska pozadina

Arduino IDE koristi objektno orijentiranu strukturu programiranja i postoji toliko mnogo tipova objekata koji se koriste u Arduino platformama. Objekt String() jedan je od najpopularnijih među njima. String() je vrsta strukture podataka klase. Postoji više verzija konstruiranja instance klase String.

- stalni niz znakova, u dvostrukim navodnicima (tj. niz znakova)
- jedan stalni znak, u jednostrukim navodnicima
- drugu instancu String objekta
- konstantan cijeli broj ili dugi cijeli broj
- konstantni cijeli broj ili dugi cijeli broj, koristeći specificiranu bazu
- cjelobrojna ili duga cjelobrojna varijabla
- varijabla cijelog ili dugog cijelog broja, koristeći specificiranu bazu
- float ili double, koristeći navedena decimalna mjesta

Konstruiranje niza iz broja rezultira nizom koji sadrži ASCII reprezentaciju tog broja. Zadana je baza deset, dakle

```
String thisString = String(15);
```

daje niz "15". Međutim, možete koristiti druge baze. Na primjer,

```
String thisString = String(15, HEX);
```

daje niz "f", koji je heksadecimalni prikaz decimalne vrijednosti 15. Ili ako više volite binarno,

```
String thisString = String(15, BIN);
```

daje vam niz "1111", koji je binarna reprezentacija 15.

- Sintaksa

```
String(val)  
Niz (val, baza)  
String(vrijednost, decimalna mjesta)
```

- Parametri

val: varijabla za formatiranje kao niz. Dopuštene vrste podataka: *niz, char, bajt, int, dugo, nepotpisani int, unsigned long, plutati, dvostruko*.

baza: (neobavezno) baza u kojoj se oblikuje integralna vrijednost. decimalPlaces: samo ako je val float ili double. Željena decimalna mjesta.

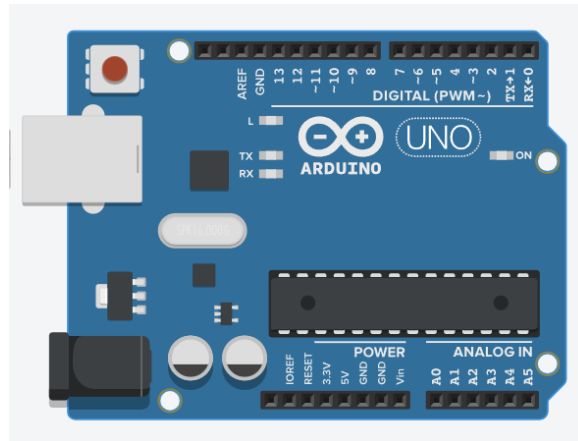
- Ulančavanje nizova

Kako bi se nizovi mogli zbrajati, prvo moraju biti definirani kao String objekt i moraju dobiti početnu vrijednost prije nego što počnete spajati različite tipove podataka.

Potrebni materijali

Arduino UNO R3, Arduino IDE softver ili Tinkercad simulacijski portal.

Eksperimentalni krug



Slika 14 Arduino UNO R3 razvojna ploča

Spojite Arduino ploču na računalo putem USB A/B kabela i otvorite Arduino IDE. Odaberite priključak (Alati→Priključak) na koji je spojen Arduino. Otvorite zaslon serijskog monitora iz izbornika Alati.

Koraci postupka

- 1- Postavite krug kako je opisano, osiguravajući da su svi spojevi sigurni.
- 2- Napišite i prenesite kod na Arduino.
- 3- Gledajte zaslon serijskog priključka.
- 4- Promatrajte promjene na ekranu.

Kodovi

```
// Globalne varijable
int intSensor=15; plutati
float floatSensor=13.495;
String stringString="Hello String";//pomoću konstantnog niza
String stringCharacter=Niz('a');//pretvaranje konstante char u String
String stringObject=Niz("Ovo je niz");//pretvaranje konstantnog niza u String objekt String
stringConcatenation=Niz(stringString+"s više");//ulančavanje dva niza String stringInteger=
Niz(15);//pomoću konstantnog cijelog broja Niz stringDEC=Niz(intSensor,PROS);//koristeći
int i bazu
Niz stringHEX=Niz(intSensor,HEX);//koristeći int i bazu (heksadecimalni) Niz
stringBIN=Niz(intSensor,BIN);//koristeći int i bazu (binarni) Niz
stringLongDEC=Niz(millis(),PROS);// korištenje duge i baze
String stringFloat=Niz(floatSensor,2);// koristeći float s preciznošću od dvije decimale

String stringOne,nizDva,stringTri; poništiti
postaviti() {
  Serijski.početi(9600);
}

poništi petlja() {
  Serijski.println(stringString);odgoditi(500);
```

```

Serijski.println(stringCharacter);odgoditi(500);
Serijski.println(stringObject);odgoditi(500); Serijski
.println(spajanje niza);odgoditi(500); Serijski
.println(stringInteger);odgoditi(500); Serijski.println
(stringDEC);odgoditi(500); Serijski.println(niz HEX);
odgoditi(500); Serijski.println(stringBIN);odgoditi(
500); Serijski.println(stringLongDEC);odgoditi(500);
Serijski.println(stringFloat);odgoditi(500);
stringLongDEC=Niz(millis(),PROS); Serijski.println(
stringLongDEC);

```

```

stringOne=Niz("Dodali ste");
nizDva=Niz("ovaj niz"); stringTri
=Niz();
// dodavanje konstantnog cijelog broja nizu: stringTri
=stringOne+123; Serijski.println(stringTri);odgoditi(
500); // dodavanje konstantnog dugog cijelog broja
nizu: stringTri=stringOne+123456789; Serijski.println(
stringTri);odgoditi(500); // dodavanje konstantnog
znaka nizu: stringTri=stringOne+'A'; Serijski.println(
stringTri);odgoditi(500); // dodavanje konstantnog
niza u niz: stringTri=stringOne+"abc"; Serijski.println(
stringTri);odgoditi(500); stringTri=stringOne+nizDva;
Serijski.println(stringTri);odgoditi(500); // dodavanje
varijable cijelog broja nizu: intsenzorVrijednost=
analognoCitaj(A0); stringOne="Vrijednost senzora: ";
stringTri=stringOne+senzorVrijednost; Serijski.println
(stringTri);odgoditi(500); // dodavanje promjenjivog
dugog cijelog broja nizu: stringOne="millis()
vrijednost: "; stringTri=stringOne+millis(); Serijski.
println(stringTri);odgoditi(500); dok(pravi);//
beskonačna petlja, ništa nije učinjeno.

```

```

}
```

Objašnjenje Kodeksa

Postaviti:Serijski monitor se aktivira brzinom od 9600 bauda.

Petlja:Poziva se korisnički definirana funkcija zbrajanja. Skeniranje se izvodi svake sekunde.

Broj pokusa: 16

16- Digitalni I/O operacije

Cilj eksperimenta

Primarni cilj ovog eksperimenta je upoznati učenike s osnovnim digitalnim ulazno/izlaznim (I/O) operacijama pomoću Arduina. Sudionici će naučiti kako konfigurirati i koristiti digitalne pinove na Arduino ploči za čitanje ulaza (poput gumba) i kontrolu izlaza (poput LED dioda).

Teorijska pozadina

Arduino platforma opremljena je digitalnim pinovima koji se mogu konfigurirati kao ulaz ili izlaz. Ovi digitalni pinovi koriste se za čitanje digitalnih signala i kontrolu fizičkih uređaja. Kada su konfigurirani kao ulazi, mogu otkriti prisutnost ili odsutnost signala (VISOKO ili NISKO). Kada su konfigurirani kao izlazi, mogu postaviti pin na HIGH (obično 5V) ili LOW (0V), što im omogućuje kontrolu uređaja poput LED dioda, motora itd.

Digitalni ulaz: Očitavanje stanja digitalnog signala. Obično se koristi s gumbima, prekidačima i senzorima.

Digitalni izlaz: Postavljanje digitalnog pina na HIGH ili LOW za kontrolu uređaja poput LED dioda ili releja.

Arduino funkcije `pinMode()`, `digitalWrite()` i `digitalRead()` temeljne su za digitalne I/O operacije:

`pinMode(pin, mod):` postavlja pin kao INPUT, OUTPUT ili INPUT_PULLUP.

`digitalWrite(pin, value):` Zapisuje VISOKU ili NIZKU vrijednost na digitalni pin.

`digitalRead(pin):` Čita vrijednost s digitalnog pina.

Potrebni materijali

Arduino UNO R3 ploča

Breadboard

LED

Otpornik od 220 ohma

Prekidač s tipkama

Premosne žice

Arduino IDE softver

Eksperimentalni krug

1- Spojite LED na jedan od digitalnih pinova kroz otpornik od 220 ohma da ograničite struju.

2- Spojite jednu stranu tipkala na drugi digitalni pin, a drugu stranu na uzemljenje. Upotrijebite pull-up otpornik ili interni INPUT_PULLUP način kako biste osigurali stabilan HIGH signal kada tipka nije pritisnuta.

Koraci postupka

- 1-Sastavite krug kako je opisano, osiguravajući da su svi spojevi sigurni.
- 2-Napišite Arduino kod za kontrolu LED-a pomoću tipke.
- 3-Učitajte kod na Arduino ploču.
- 4-Pritisnite i pustite tipku za testiranje LED kontrole.
- 5-Promatrajte reakciju LED-a na tipku.

Primjer koda

```
// Definirajte pin brojeve

    const int buttonPin = 2; // broj ige tipkala
    const int ledPin = 13; // broj LED pina

// Varijable će se promijeniti:

    int buttonState = 0; // varijabla za očitavanje statusa tipkala
    void setup() {
// inicijalizirati LED pin kao izlaz:
        pinMode(ledPin, IZLAZ);
// inicijaliziraj pin tipke kao ulaz:
        pinMode(gumbPin, INPUT);
    }
void loop() {
// čitanje stanja vrijednosti tipke:
    buttonState = digitalRead(buttonPin);
// provjeriti je li tipka pritisnuta.
// ako jest, buttonState je VISOKO:
    if (buttonState == HIGH) {
// uključi LED:
        digitalWrite(ledPin, HIGH);
    } inače {
// isključi LED:
        digitalWrite(ledPin, LOW);
    }
}
```


Objašnjenje Kodeksa

- Postavljanje: funkcija `pinMode()` konfigurira LED pin kao izlaz, a pin gumba kao ulaz.
- Petlja: program kontinuirano čita stanje gumba pomoću `digitalRead()`. Ako se pritisne gumb (očitava HIGH zbog konfiguracije pull-up), LED se uključuje. U suprotnom, LED je isključen. Stanje LED diode se mijenja prema pritisku tipke.

Ovaj eksperiment pokazuje osnovne digitalne I/O operacije, koje su temelj za složenije Arduino projekte koji uključuju senzore, motore i druge komponente.

Broj pokusa: 17

17-Analogne I/O operacije

Cilj eksperimenta

Cilj ovog eksperimenta je upoznati učenike s konceptom i primjenom analognih ulazno-izlaznih operacija pomoću Arduino ploče. Sudionici će naučiti kako čitati analogne signale sa senzora i upravljati analognim uređajima poput prigušivih LED dioda ili motora s modulacijom širine impulsa (PWM).

Teorijska pozadina

Za razliku od digitalnih signala koji su VISOKI ili NISKI, analogni signali mogu predstavljati niz vrijednosti. Arduino ploče mogu čitati analogne signale pomoću analogno-digitalnih pretvarača (ADC) i proizvoditi izlaz sličan analognom pomoću modulacije širine impulsa (PWM).

Analogni ulaz: Arduino ploče, kao i Uno, imaju set analognih ulaznih pinova označenih od A0 do A5. Ovi pinovi mogu čitati analogne signale sa senzora, kao što su temperatura ili svjetlo, i pretvarati ih u digitalnu vrijednost koju može obraditi mikrokontroler.

Analogni izlaz (PWM): Iako Arduino ne može generirati pravi analogni izlaz, može ga simulirati putem PWM-a. PWM kontrolira relativni radni ciklus digitalnog signala kako bi se mijenjala snaga koja se isporučuje uređajima poput LED dioda ili motora.

Ključne funkcije za analogne operacije:

- `analogRead(pin)`: Čita vrijednost s analognog pina i vraća vrijednost između 0 (0V) i 1023 (5V).
- `analogWrite(pin, value)`: Zapisuje analognu vrijednost (PWM val) na pin za simulaciju analognog izlaza. Vrijednost može biti između 0 (uvijek isključeno) i 255 (uvijek uključeno).

Potrebni materijali

Arduino UNO R3 ploča

Breadboard

Potenciometar (npr. 10kΩ)

LED

Otpornik od 220 ohma

Premosne žice

Arduino IDE softver

Ekperimentalni krug

1-Spojite potenciometar na jedan od analognih ulaznih pinova (npr. A0). Spojite jednu stranu na 5V, srednji pin na A0, a drugu stranu na GND.

2-Spojite LED na digitalni pin koji ima PWM (npr. 9, 10, 11 na Uno) preko otpornika od 220 ohma da ograničite struju.

Koraci postupka

1-Postavite krug kako je opisano, osiguravajući da su svi spojevi sigurni.

2-Napišite i prenesite Arduino kod koji čita analognu vrijednost s potenciometra i u skladu s tim kontrolira svjetlinu LED-a.

3-Okrenite potenciometar i promatrajte promjenu svjetline LED-a.

4-Razumite odnos između položaja potenciometra i svjetline LED-a.

Primjer koda

```
// Definirajte pin brojeve
const int potPin = A0; // broj pina potenciometra
const int ledPin = 9; // broj LED pina

// Varijabla za pohranu vrijednosti potenciometra
int PotVrijednost = 0;

void setup() {
  // inicijalizirati LED pin kao izlaz:
  pinMode(ledPin, IZLAZ);

  // inicijalizirati serijsku komunikaciju na 9600 bita u sekundi:
  Serial.begin(9600);
}

void loop() {
  // očitajte vrijednost s potenciometra:
```

```

potValue = analogRead(potPin);

// mapiranje vrijednosti potencimetra od 0-1023 do 0-255:
int ledBrightness = map(potValue, 0, 1023, 0, 255);

// postavite svjetlinu LED-a:
analogWrite(ledPin, ledBrightness);

// ispis rezultata na serijski monitor:
Serial.print("Potentiometar: ");
Serial.print(potValue);

Serial.print("\t LED svjetlina: ");

Serial.println(ledBrightness);

kašnjenje(10); // mala odgoda za stabilnost
}

```

Objašnjenje Kodeksa

Postaviti: Inicijalizira LED pin kao izlaz i započinje serijsku komunikaciju za nadzor.

Petlja: Funkcija `analogRead()` čita analognu vrijednost s potencimetra, koja se zatim preslikava u odgovarajuću PWM vrijednost (0-255). Ova se vrijednost koristi u `analogWrite()` za postavljanje svjetline LED-a. Vrijednosti potencimetra i LED-a ispisuju se na serijski monitor radi promatranja.

Ovaj eksperiment pokazuje kako očitati analogne ulaze i proizvesti PWM izlaze, neophodne za povezivanje sa širokim rasponom senzora i upravljanje različitim aktuatorima u složenijim Arduino projektima.



INA-CODE

roby<code

web & mobile



UNIVERSITY OF ZAGREB
Faculty of Electrical
Engineering and
Computing



I.I.S. ABRAMO LINCOLN ENNA