



# Arduino Nano

Kurs programiranja za Arduino Nano V3, Nano Every i Nano 33 IoT



Ashwin Pajankar



Agencija Eho  
[www.infoelektronika.net](http://www.infoelektronika.net)





• Sva prava zadržana. Nijedan deo ove knjige ne sme biti reprodukovan u bilo kom materijalnom obliku, uključujući fotokopiranje ili slučajno ili nenamerno smeštanje na bilo koji elektronski medijum sa ili uz pomoć bilo kog elektronskog sredstva, bez pismenog odobrenja nosioca autorskih prava osim u skladu sa odredbama zakona o autorskim pravima, dizajnu i patentima iz 1988. godine ili pod uslovima izdatim od Copyright Licensing Agency Ltd, 90 Tottenham Court Road, London, England W1P 9HE. Prijave za pismene dozvole radi štampanja bilo kog dela ove publikacije upućuje se izdavaču ove knjige.

• Izjava: Autor i izdavač su uložili najveće napore da bi se obezbedila tačnost informacija sadržanih u ovoj knjizi. Autor i izdavač ne mogu da pretpostave neprijatnosti i ovom izjavom isključuju bilo kakvu odgovornost za bilo koju stranku koja bi imala gubitke ili štetu uzrokovanu greškama ili propustima u ovoj knjizi, bez obzira da li su greške ili propusti nastali usled nemara, nezgode ili bilo kog drugog razloga.

ISBN 978-86-80134-43-7

### **Arduino Nano**

Naslov originala: Kickstart to Arduino Nano

Autor: Ashwin Pajankar

Prevod: Biljana Tešić

Izdaje i štampa: Agencija Eho, Niš

e-mail: redakcija@infoelektronika.net

Tiraž: 200

Godina izdanja: 2022

CIP - Каталогизација у публикацији  
Библиотеке Матице српске, Нови Сад

004.42:[621.38:62-55

ПАЈАНКАР, Ашвин

Arduino Nano : kurs programiranja za Arduino Nano V3, Nano Every i Nano 33 IoT / Ashwin Pajankar ; [prevod Biljana Tešić]. - Niš : Agencija Eho, 2022 (Niš : Agencija Eho). - 191 str. : ilustr. ; 24 cm

Tiraž 200.

ISBN 978-86-80134-43-7

а) Програмирање -- Електроника -- Контролери

COBISS.SR-ID 75617033



Priznanja .....	9
Posveta .....	9
Predgovor .....	10
<b>Poglavlje 1 - Uvod u Arduino platformu i Arduino Nano .....</b>	<b>11</b>
Mikrokontroler .....	11
Popularne MCU porodice .....	11
Arduino .....	12
Porodica Arduino „Classic“ .....	12
Porodica Arduino „MKR“ .....	12
Porodica Arduino „Nano“ .....	13
Porodica Arduino „Pro“ .....	13
Arduino je za svakoga .....	13
Arduino ekosistem .....	13
Arduino softver .....	14
Zvanične Arduino ploče .....	14
Arduino klonovi i derivati .....	14
Arduino falsifikati .....	14
Arduino Nano i Arduino Nano Every .....	14
Raspored pinova za Arduino Nano i Arduino Nano Every .....	16
Napajanje Nano i Nano Every MCU ploča .....	19
Arduino IDE .....	20
Instaliranje stabilne verzije Arduino IDE-a .....	26
Instalacija Arduino IDE-a na Raspberry Pi OS-u .....	30
Rad u Board Manager-u .....	32
Rad u Arduino Nano Every ploči .....	35
Rad u Arduino Nano-u .....	39
Rezime .....	41
<b>Poglavlje 2 - Igranje sa elektronikom .....</b>	<b>42</b>
Osnove programiranja pomoću Arduino IDE-a .....	42
Treptanje ugrađenog LED-a .....	43
Rad sa osnovnim elektronskim komponentama .....	45
Prototipske ploče i napajanja .....	46
Kratkospojnici .....	50
Svetleće diode .....	51
Otpornici .....	52
Tasteri .....	53
Poboljšanje skice treptanja LED-a pomoću funkcija .....	54

Izrada vašeg prvog kola na prototipskoj ploči.....	56
Kola koja koriste Nano .....	58
Rad sa više LED-ova.....	59
Dodavanje tastera u kolo.....	63
Rad sa RGB LED-ovima .....	65
Korišćenje Arduino Nano ploča sa modulima (shields) za proširenje .....	67
Rezime .....	70
<b>Poglavlje 3 - Različite magistrale i analogni ulaz .....</b>	<b>71</b>
Paralelni i serijski prenos podataka .....	71
Arduino Serial .....	73
SPI i I2C .....	75
Analogni ulaz .....	75
Iscrtavanje više promenljivih.....	80
Rezime .....	81
<b>Poglavlje 4 – Modulisanje širine impulsa i upravljanje jednapolarnim motorom sa digitalnim I/O .....</b>	<b>82</b>
Koncept modulisanja širine impulsa .....	82
PWM sa Arduino Nano-m.....	84
Rad sa servo motorom .....	89
Rad sa jednapolarnim koračnim motorom 28BYJ-48 i drajverom motora ULN2003A .....	92
Korišćenje prilagođene biblioteke za koračne motore .....	98
Rad sa više koračnih motora.....	101
Rezime .....	102
<b>Poglavlje 5 - Iscrtavanje geometrijske „umetnosti“ na eksternom displeju .....</b>	<b>103</b>
Ilitek 9225 drajver IC i displej .....	103
Programiranje displeja .....	104
Rezime .....	143
<b>Poglavlje 6 - Rad sa zujalicom i senzorom.....</b>	<b>144</b>
Rad sa zujalicom .....	144
Rad sa džojstikom .....	149
Rad sa senzorom temperature DS18B20 .....	151
Rezime .....	153
<b>Poglavlje 7 - Rad sa Arduino Nano 33 IoT-om .....</b>	<b>154</b>
Uvod u Nano 33 IoT ploču.....	154
Početak rada .....	157
Rad sa WiFiNINA bibliotekom .....	159

Server za grupno časovanje zasnovan na Telnet-u .....	172
Pingovanje udaljenog servera .....	174
Jednostavan veb klijent.....	176
Rad sa satom u realnom vremenu .....	177
Korišćenje senzora temperature DS18B20 zajedno sa RTC-om .....	179
Vizuelizacija grafikona temperature pomoću ThingSpeak-a .....	185
Programiranje ugrađenog IMU-a .....	190
Rezime .....	191
Zaključak.....	191



## Priznanja

Želim da se zahvalim svom prijatelju Anuradhai što me je ohrabrio da napišem ovu knjigu. Takođe želim da izrazim svoju iskrenu zahvalnost Ferdinandu TeWalvaartu, Janu Buitingu, Alinai Neacsu i Shenjai Panik iz Elektor tima što su me vodili kroz svaku fazu procesa objavljivanja. Ovo je moja druga knjiga za „Elektor International; Media“ i veliko je iskustvo saradivati sa njihovim izdavačkim timom. Na kraju, želim da iskažem zahvalnost svima koji su direktno i indirektno povezani sa ovim projektom.

## Posveta

Ova knjiga je posvećena sećanju na prof. Govindarajulu Regetija (9. jul 1945. – 18. mart 2021.)

Popularno poznat svima kao RGR, profesor Govindarajulu je stekao diplomu iz elektrotehnike i elektronike na Univerzitetu „JNTU Kakinada“. Takođe je magistrirao i doktorirao na Univerzitetu „IIT-u Kanpur“. Prof. Govindarajulu je bio među prvim članovima fakulteta „IIT Hyderabad“ i odigrao je značajnu ulogu u tome da „IIT Hyderabad“ postane vrhunska institucija kakva je danas. Bio je do sada najomiljeniji i hvaljeni član fakulteta na institutu. Bio je pun energije za podučavanje i staromodnog šarma. Nema sumnje da je bio zabrinut za svakog učenika kao pojedinca, vodeći računa da ga upozna i vodi. Bio je predavač, vodič i mentor mnogim grupama studenata na fakultetu „IIT Hyderabad“, ali i autoru ove knjige.

## Predgovor

Arduino koristim od 2016. godine i verujem da je to odlična platforma ne samo za učenje elektronike, već i za primenu proizvodnih sistema u stvarnom životu. Koristeći Arduino ploče, proizveo sam i primenio mnoge aplikacije iz stvarnog sveta, kao što su električna infrastruktura kod kuće i na radnim mestima sa omogućenim IoT-om, sistemi za praćenje vremenske prognoze na webu i robot za doček gostiju na recepciji mog bivšeg poslodavca.

Uvek sam želeo da napišem kompletnu i detaljnu knjigu o jednoj od najistaknutijih Arduinovih ploča mikrokontrolera „Nano“. U ovoj knjizi su obuhvaćena tri člana ove porodice: Arduino Nano V3, Arduino Nano Every i Arduino Nano 33 IoT, u „srećnom“ skladu sa Arduino IDE 1.x i Arduino IDE 2 RC5 u svrhu demonstracije C i C++ koda. Napisao sam knjigu na „korak po korak“ i detaljan način da objasnim koncepte, da izradim potrebna kola i da na kraju napišem kod. Knjiga sadrži više od 60 primera koji se odnose na različite glavne koncepte iz sveta elektronike i posebno na Arduino ekosisteme.

Ova knjiga je od mene zahtevala da pronađem i pregledam gomilu dokumentacije, da pohađam onlajn tutorijale, da napišem kod i da pišem postove na onlajn forumima kada sam imao neki problem. Ovaj rezultat predstavlja stotine radnih sati koje sam potrošio da završim projekat. Preporučio sam mnoge onlajn izvore za kod i slike. Takođe sam pomenuo te izvore sa svim licencama kad god i gde god sam pozajmio i izmenio izvorne kodove i slike.

Završio sam pisanje nacрта knjige kada su se Indija (zemlja u kojoj trenutno boravim) i svet postepeno oporavljali od pandemije COVID-19. Ovaj projekat i smernice koje je obezbedio Elektor tim pružili su mi sjajnu nepokolebljivost i motivaciju da nastavim da istražujem svet znanja na Arduino. Nadam se da će se svim čitaocima svideti ova knjiga i da će uživati u njoj kao što sam ja uživao da je pišem. Srećno učenje i istraživanje!

Napomena za čitaoce. Arhivska datoteka (.zip), koja sadrži primere softvera i dijagrame kola u Fritzing stilu razmotrene u knjizi, može se besplatno preuzeti sa stranice proizvoda i resursa knjige na sajtu [www.elektor.com](http://www.elektor.com) ili [infoelektronika.net](http://infoelektronika.net)



## Poglavlje 1 - Uvod u Arduino platformu i Arduino Nano

Dobrodošli u prvo poglavlje. U slučaju da niste pročitali „Sadržaj i Predgovor“, toplo preporučujem da to uradite pre nego što počnete da čitate ovo poglavlje. Hajde da započnemo putovanje u neverovatan svet Arduino platforme i njenog ekosistema. Ovo je uvodno poglavlje, tako da ćete naučiti mnogo koncepata potrebnih za izgradnju dovoljnog znanja za početak rada na Arduinou. U ovom poglavlju su razmotrene sledeće teme:

- mikrokontroler
- Arduino
- Arduino je za svakoga
- Arduino ekosistem
- Arduino Nano i Arduino Nano Every
- Napajanje Nano i Nano Every ploča mikrokontrolera
- Arduino IDE
- rad u Board Manager-u
- rad u Arduino Nano Every
- rad u Arduino Nano

Nakon čitanja ovog poglavlja, spremni ste za istraživanje Arduino platforme i ekosistema.

### Mikrokontroler

Mikrokontroler je obično mali, kompletan računarski sistem na jednom integrisanom kolu (skraćeno IC). MCU (MicroController Unit) može imati jedan ili više procesora, memoriju i programabilni ulaz/izlaz (IO) u jednom IC paketu. U većini slučajeva, postoje dve vrste memorije uključene u čip. Prva je programabilna memorija, obično EEPROM (Electrically Erasable Programmable Read-Only Memory), OTP (One Time Programmable) ROM, feroelektrični RAM ili NOR fleš. Takođe, MCU-ovi imaju i malu količinu RAM-a (Random Access Memory) u kojoj se programi učitavaju i izvršavaju. Ukratko, MCU je mikroprocesor sa programskom memorijom, RAM-om i programabilnim IO na jednom čipu. Glavna razlika između mikroprocesora i MCU-a je u njihovoj primeni. Mikroprocesori su procesori opšte namene koji se koriste u desktop računarstvu i računarstvu opšte namene. MCU-ovi se obično koriste u ugrađenim sistemima i imaju zadatak da izvrše neku vrstu fizičke akcije. Na primer, MCU-ovi se mogu koristiti u štampačima, kontrolnoj tabli vozila ili kontrolama motora u automobilima i avionima.

### Popularne MCU porodice

Na osnovu njihove arhitekture, MCU-ovi se mogu podeliti u različite porodice. Evo liste nekih od najčešće korišćenih porodica:

- PIC (Peripheral Interface Controller)
- ARM (Advanced RISC Machines)
- AVR (Alf and Vegard's RISC Processor)
- MSP (Mixed Signal Processor)
- Intel 8051 i derivati

## Arduino

Arduino (<https://www.arduino.cc>) je MCU platforma otvorenog koda, što znači da ima otvoreni kod i besplatni softver i hardver otvorenog koda. Obično kombinuje MCU na jednoj PCB sa drugim funkcijama. Arduino je popularna porodica MCU-a zasnovanih na MCU pločama. Ranije je Arduino grupa takođe proizvela seriju računara zasnovanih na Linuxu. Međutim, do sada se čini da su prekinuli njihovi proizvodnju i da se fokusiraju isključivo na MCU-ove. Arduino porodica ima mnogo članova, a oni se mogu podeliti u tri podporodice na sledeći način:

### Porodica Arduino „Classic“.

Ova porodica ima najpopularnije, najstarije i najuspešnije ploče, uključujući (april 2022.):

- Arduino UNO Rev3
- Arduino Mega2560 Rev3
- Arduino Leonardo
- Arduino UNO Mini Limited Edition
- Arduino Due
- Arduino Micro
- Arduino Zero
- Arduino UNO Wi-Fi Rev2

Možete da ih pronađete na URL-u:

<https://www.arduino.cc/en/hardware#classic-family>

### Porodica Arduino „MKR“.

MKR je porodica Arduino MCU ploča koje kombinuju MCU sa naprednim komunikacijama, kao što su Wi-Fi, LoRa, Bluetooth i Sigfox. Zasnovani su na MCU-ovima koji su članovi Microchipove SAM D21 porodice. Ovo su članovi porodice MKR:

- Arduino Nano 33 IoT
- Arduino Nano RP2040 Connect
- Arduino Nano BLE Sense
- Arduino Nano 33 BLE
- Arduino Nano Every
- Arduino Nano

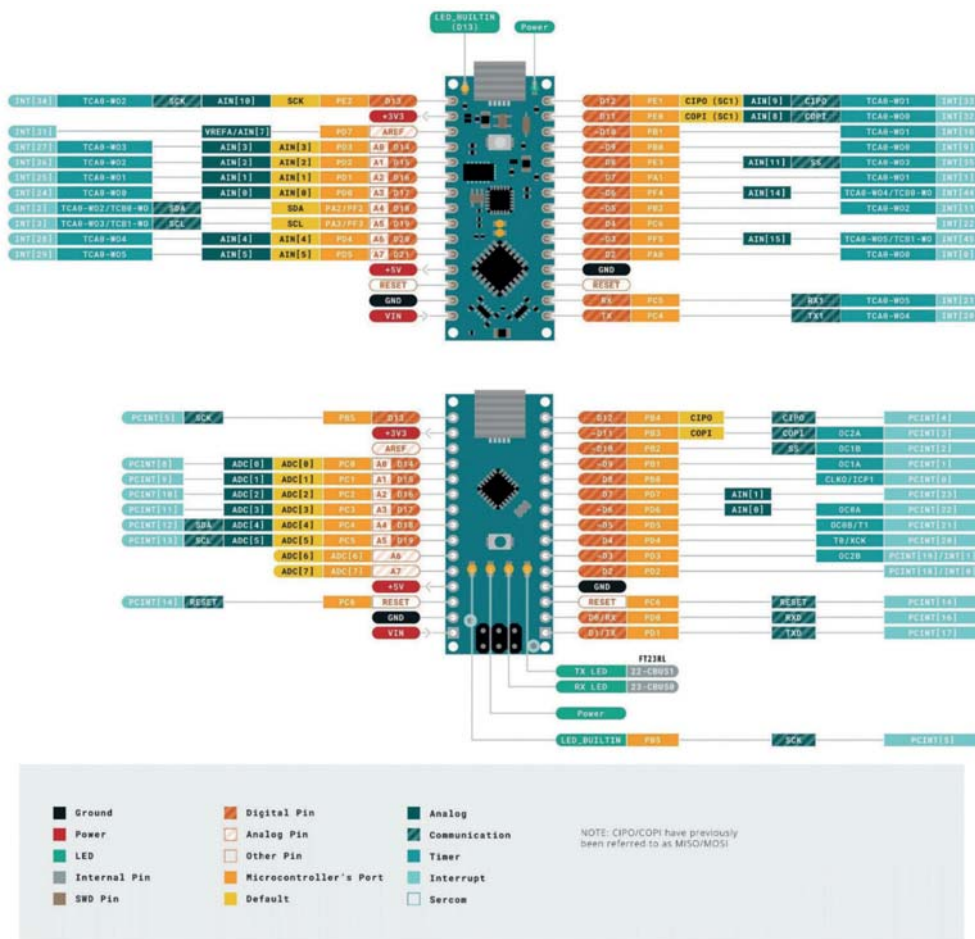
Više o porodici MKR možete pročitati na adresi

<https://www.arduino.cc/en/hardware#mkr-family>

Takođe možete pročitati više o porodici SAM D21 na adresi:

[https://content.arduino.cc/assets/mkr-microchip\\_samd21\\_family\\_full\\_datasheets40001882d.pdf](https://content.arduino.cc/assets/mkr-microchip_samd21_family_full_datasheets40001882d.pdf)

Na slici 1-2, fokusirajte se na pinove Serial Communication dodeljene za I<sup>2</sup>C (SDA i SCL), SPI (SS, MOSI/COPI, MISO/CIPO, SCK) i UART (RX i TX),



Slika 1-2 Serijska komunikacija i opis pina za prekid

Na slici 1-2, u gornjem delu prikazane su funkcije pina Nano Every, a u srednjem delu funkcije pina Arduino Nano.

Značenje dodatnih pinova možete proučiti sa slike 1-3.

## Poglavlje 2 - Igranje sa elektronikom

U prethodnom poglavlju ste naučili osnove **Arduino ekosistema**, nekoliko ploča iz **porodice Nano** i osnove trenutne stabilne i sledeće verzije Release Candidate **Arduino IDE-a**. U ovom poglavlju ćete naučiti osnove pisanja programi/skice pomoću Arduino IDE-a i rad sa elektronikom. Evo liste tema koje ćete naučiti u ovom poglavlju:

- osnove programiranja pomoću Arduino IDE-a
- rad sa osnovnim elektronskim komponentama
- poboljšanje skice „LED treptanja“ pomoću funkcija
- izrada vašeg prvog kola na prototipskoj ploči
- rad sa RGB LED-ovima
- korišćenje Arduino Nano ploča sa modulima (shields) za proširenje

Nakon što pročitate ovo poglavlje, trebalo bi da znate osnove Arduino programiranja i nekoliko elektronskih komponenti.

### Osnove programiranja pomoću Arduino IDE-a

U prethodnom poglavlju, otpremili ste podrazumevani osnovni program na Arduino ploču. U ovom odeljku ćete naučiti značenje koda. Ako se sećate, sačuvali ste program pre otpremanja. Čak i ako niste sačuvali program (ili skicu, možete koristiti ove reči bez razlike, što ću ja raditi tokom cele knjige), uvek možete da kreirate novu skicu i ona će podrazumevano imati osnovni kod. Osnovni kod izgleda ovako:

```
prog00.ino
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

Kopirao sam i nalepio kod kakav jeste, bez ikakvih modifikacija. To je minimalni kod potreban za kompajliranje (ili proveru, ponavljam, ove dve reči mogu da se koriste nizmjenično) skice. Kao što vidite, postoje dva bloka koda, *setup()* i *loop()*. Da biste definisali vaš osnovni program, oba bloka su neophodna za skicu. Sada pokušajte da uklonite blok *setup()* i da kompajlirate skicu i naći ćete sledeću poruku o grešci u delu **Output**:

```
undefined reference to 'setup'
```

Slično tome, ako uklonite deo `loop()` i pokušate da izvršite kompajliranje, ono će ponovo biti neuspešno i prikazaće se sledeća poruka:

```
undefined reference to 'loop'
```

Ovo je minimum koda koji je potreban za bilo koju kompilaciju – uvek će biti tu u svakoj skici koju neko napiše za Arduino.

Kada se bilo koja skica učita na ploču, ona se čuva u fleš memoriji ploče. Čak i nakon isključivanja ploče sa računara (ili bilo kog drugog izvora napajanja), fleš memorija će zadržati kod i neće biti obrisan dok ne otpremite novu skicu. Svaki put kada se ploča uključi, deo `setup()` se izvršava jednom. Nakon toga, deo `loop()` nastavlja da se izvršava sve dok se ploča napaja. Ako se ploča više ne napaja, prestaje da pokreće kod (očigledno!). Kada ponovo povežete ploču sa izvorom napajanja, ceo proces se ponavlja.

### Treptanje ugrađenog LED-a

Sve Arduino ploče se isporučuju sa ugrađenim LED-om. Na ploči se LED može identifikovati kao komponenta označena slovom “L”. Hajde da napišemo neki kod za treptanje LED-a. Napisaćemo program **Hello World!** iz „sveta“ programiranja elektronike. Takođe, korišćiću Arduino Nano Every ploču sa IDE-om 2.0 R5. Međutim, možete koristiti Arduino Nano i bilo koju verziju IDE-a za pisanje skice. Hajde da napišemo jednostavnu skicu (pogledajte: `prog00.ino`) da bismo uključili LED na ploči.

```
prog00.ino
/*
This Program is written by Ashwin Pajankar
for Elektor on 20-APR-2022
*/

// the setup function runs once
void setup()
{
  pinMode(LED_BUILTIN, OUTPUT);
}

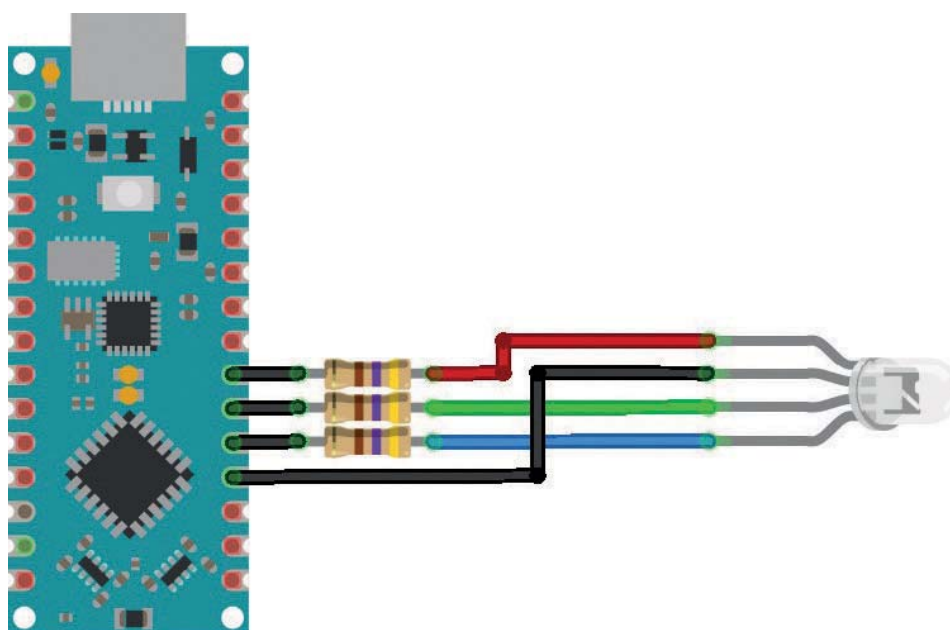
// the loop function runs forever
void loop()
{
  digitalWrite(LED_BUILTIN, HIGH);
}
```

Hajde da ispitamo ovaj kod liniju po liniju. Kod koji počinje znakovima `/*` i završava se sa znakovima `*/` predstavlja višelinijski komentar. Programeri obično koriste komentare da bi obezbedili dodatne informacije o kodu. Slično tome, linije koje počinju znakovima `//` su jednolinijski komentari. Spomenuo sam u prethodnom poglavlju da većina Arduino ploča ima ugrađen LED.

Svaki digitalni pin Arduina ima ugrađeni pull-up otpornik. Kada inicijalizujete digitalni pin pomoću režima **INPUT\_PULLUP**, on proizvodi LOW signal kada se pritisne. Kada je dugme u normalnom stanju, ono je HIGH. Funkcija **digitalRead()** čita vrednost digitalnog pina koji je prosleđen kao argument. Ova skica omogućava da LED svetli kada se pritisne dugme.

### Rad sa RGB LED-ovima

Do sada ste radili sa LED-ovima u jednoj boji. Sada ćete naučiti osnove RGB LED-ova. To su u suštini tri različite LED-a u jednom paketu. RGB LED ima četiri pina. Jedan pin je zajednička anoda (koja se povezuje sa pozitivnim nivoom napajanja) ili zajednička katoda (koja se povezuje sa uzemljenjem). Ostali pinovi su za crvene, zelene i plave LED-ove i mogu se povezati sa digitalnim izlaznim pinom. U RGB LED-a sa **zajedničkom katodom**, potrebno je da pošaljete HIGH signal na LED pin da bi se LED uključio. U RGB LED-u sa **zajedničkom anodom**, potrebno je da pošaljete LOW signal na LED pin da bi se LED uključio. Na slici 2-21 prikazano je kolo za LED sa zajedničkom katodom. Svi otpornici imaju vrednost 470 oma.



Slika 2-21 RGB LED kontrolno kolo sa zajedničkom katodom

Skica (**prog10.ino**) koristi binarni brojač da prikaže sve kombinacije boja koje su moguće pomoću digitalnog izlaza. Prikazuje 7 boja i stanje Off (tj. 8 kombinacija, koristeći tri digitalna izlazna pina).

**prog10.ino**

```
int duration = 200;

void setup()
{
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
}

void loop()
{
  digitalWrite(2, HIGH); digitalWrite(3, HIGH); digitalWrite(4, HIGH);
  delay(duration);
  digitalWrite(2, HIGH); digitalWrite(3, HIGH); digitalWrite(4, LOW);
  delay(duration);
  digitalWrite(2, HIGH); digitalWrite(3, LOW); digitalWrite(4, HIGH);
  delay(duration);
  digitalWrite(2, HIGH); digitalWrite(3, LOW); digitalWrite(4, LOW);
  delay(duration);
  digitalWrite(2, LOW); digitalWrite(3, HIGH); digitalWrite(4, HIGH);
  delay(duration);
  digitalWrite(2, LOW); digitalWrite(3, HIGH); digitalWrite(4, LOW);
  delay(duration);
  digitalWrite(2, LOW); digitalWrite(3, LOW); digitalWrite(4, HIGH);
  delay(duration);
  digitalWrite(2, LOW); digitalWrite(3, LOW); digitalWrite(4, LOW);
  delay(duration);
}
```

Ako je potrebno, možete zameniti RGB LED sa zajedničkom katodom RGB LED-om sa zajedničkom anodom. Za to morate da prilagodite kolo, kao što je prikazano na slici 2-22.

## Poglavlje 3 - Različite magistrale i analogni ulaz

U prethodnom poglavlju ste naučili osnove elektronskih kola i programiranja pomoću Arduina. Sada već znate da kreirate osnovna kola i da programirate na početničkom nivou.

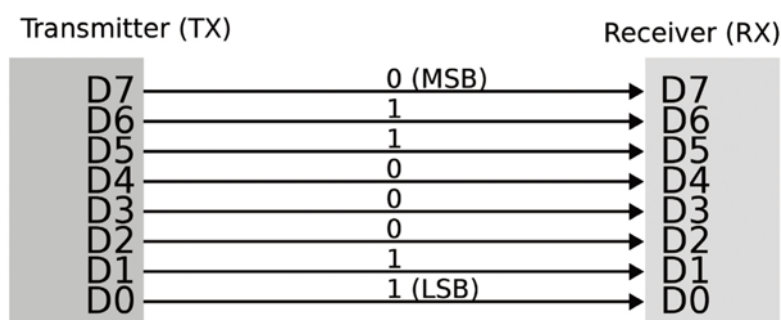
Nastavićete uzbudljivo putovanje Arduino programiranja. U ovom poglavlju ćete naučiti sledeće teme:

- paralelni i serijski prenos podataka
- Arduino serial
- SPI i I<sup>2</sup>C
- analogni ulaz
- iscertavanje više promenljivih

Nakon što pročitate celo poglavlje, biće vam lako da koristite različite tipove magistrala i koncept analognog ulaza.

### Paralelni i serijski prenos podataka

U svetu računarstva postoje dva načina za prenos elektronskih signala (HIGH, LOW i analogni), paralelni i serijski. Magistrala (bus) je komunikacioni sistem za prenos podataka. U paralelnom prenosu podataka (takođe poznatom kao **paralelna magistrala**), signali se prenose pomoću više magistrala, kao što je prikazano na slici 3-1.

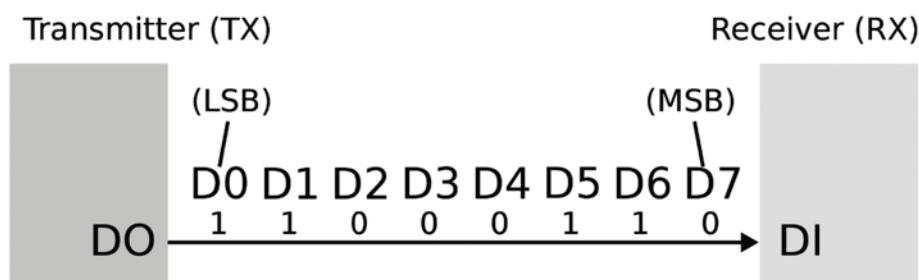


Slika 3-1 Paralelni prenos podataka

Kao što vidite, postoji posebna linija magistrale za svaki deo. Ovaj metod je veoma brz, jer se više podataka može preneti za kraće vremena. Međutim, ovaj metod nije naročito efikasan, jer zahteva mnogo hardvera, a samim tim i skup je za implementaciju. Većina internih magistrala u mikroprocesoru, kao što su magistrala podataka, adresna magistrala i kontrolna magistrala, su paralelne.

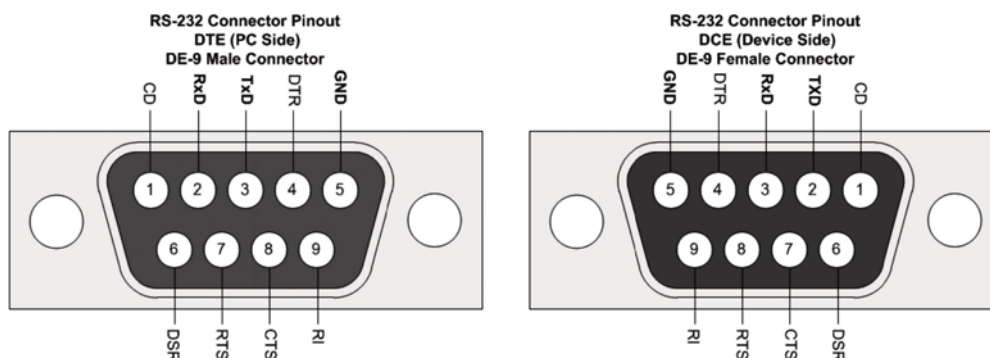
Efikasnija tehnika prenosa podataka je serijska komunikacija (ili **serijska magistrala**). Na slici 3-2 predstavljena je serijska magistrala.





Slika 3-2 Serijski prenos podataka

U serijskim komunikacionim sistemima, bitovi podataka se prenose serijski, odnosno sekvencijalno, pomoću jednog kanala. Obično, serijski komunikacioni sistem ima par predajnika (TX) i prijemnika (RX) za dvosmernu komunikaciju. Serijski prenos podataka je sporiji, ali efikasniji kada je reč o hardveru i troškovima. Postoje dve vrste serijske komunikacije: asinhrona i sinhrona. U asinhronoj serijskoj komunikaciji, TX i RX nisu sinhronizovani radnim taktom, već bitovima podataka. Najbolji i najčešće korišćeni primer asinhrona serijske komunikacije je RS-232. Na slici 3-2 prikazane su „šeme“ „muških“ i „ženskih“ konektora za par RS-232 komunikacionih uređaja.



Slika 3-3 RS-232 „muški“ i „ženski“ konektori

Drugi primer asinhrona serije je UART (univerzalni asinhroni prijemnik-predajnik). To ćete detaljno naučiti u ovom poglavlju.

Drugi tip serijskog prenosa podataka je sinhroni prenos podataka. Ovde su krajnje tačke i prenos podataka sinhronizovani impulsima radnog takta. Postoji mnogo načina na koje se ovo može primeniti. Istaknuti primeri su USART (univerzalni sinhroni i asinhroni prijemnik-predajnik), Serial Peripheral Interface (SPI) i Inter-Integrated Circuit (I<sup>2</sup>C ili jednostavno I2C). U ovom poglavlju ćemo ih detaljno istražiti.

## Poglavlje 4 – Modulisanje širine impulsa i upravljanje jednapolarnim motorom sa digitalnim I/O

U prethodnom poglavlju ste proučavali i vežbali Arduino Serial, serijsko iscrtavanje i analogne ulaze. Takođe ste naučili osnove različitih magistrala i pinova povezanih sa njima.

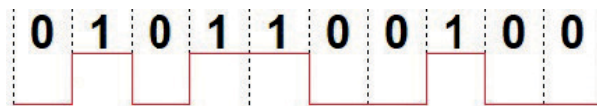
U ovom poglavlju ćete istražiti neverovatan svet **modulisanja širine impulsa** (skraćeno **PWM**). Evo tema koje ćemo proučavati i demonstrirati u ovom poglavlju:

- koncept modulisanja širine impulsa
- PWM sa Arduino Nano-m
- rad sa servo motorom
- rad sa jednapolarnim koračnim motorom 28BIJ-48 i drajverom motora ULN2003A
- korišćenje prilagođene biblioteke za koračne motore

Kada pročitate celo poglavlje, razumećete koncept PWM-a i njegove primene.

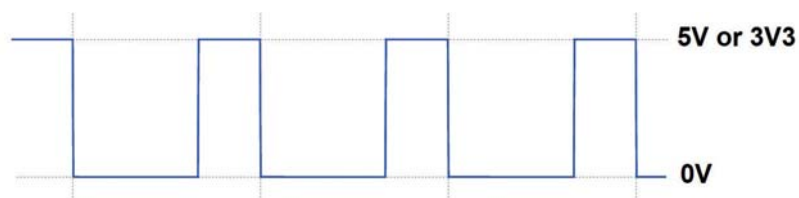
### Koncept modulisanja širine impulsa

Znate da uobičajen digitalni signal ima jedno od dva stanja za svaki period: HIGH (5 V/3V3 ili digitalni 1) ili LOW (0 V ili digitalni 0). Dužina vremena koje je potrebno talasnom obliku da se ponovi poznata je kao period. Slika 4-1 je primer digitalnog signala:



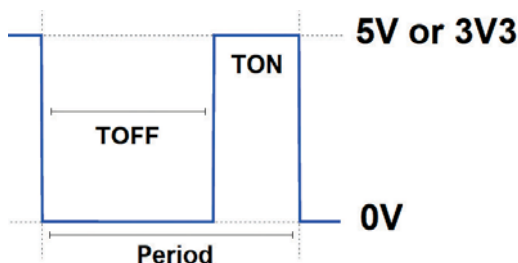
Slika 4-1 Digitalni signal koji prenosi informacije (prilagođeno sa slike koju je obezbedio El Pak pod licencom <https://creativecommons.org/licenses/by-sa/3.0/deed.en>)

Dakle, tokom celog perioda, signal je ili HIGH ili LOW. Moguće je modifikovati digitalni signal tako da deo signala bude HIGH, a preostali deo LOW u jednom periodu (ili jednom impulsu). To je poznato kao modulisanje širine impulsa (PWM). Na slici 4-2 prikazan je modulisan digitalni signal.



Slika 4-2 Modulisanje digitalnog signala

Primićete da u impulsu jednog perioda možete videti oba nivoa signala, HIGH i LOW. U suštini, manipulišete širinom HIGH signala. Na slici 4-3 prikazana je anatomija modulisanja širine digitalnog impulsa.



Slika 4-3 Anatomija modulisanja širine digitalnog impulsa

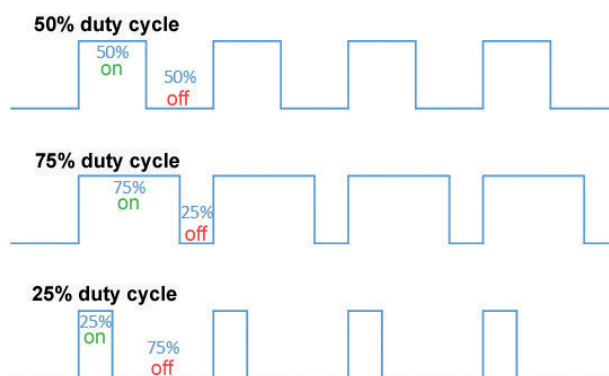
U jednom impulsu, vreme tokom kojeg je impuls LOW je poznat kao TOFF ili  $T_{OFF}$  a vreme za koje je HIGH poznato je kao TON ili  $T_{ON}$ . Sledi formula koja pokazuje odnos između perioda  $T_{ON}$  i  $T_{OFF}$

$$\text{period signala} = T_{ON} + T_{OFF}$$

Radni ciklus (takođe poznat kao ciklus napajanja) je deo perioda tokom kojeg je impuls aktivan. Utvrđuje se u procentima sledećom formulom,

$$\text{radni ciklus} = (T_{ON} / \text{period}) \times 100$$

Na slici 4-4 su prikazani različiti radni ciklusi.



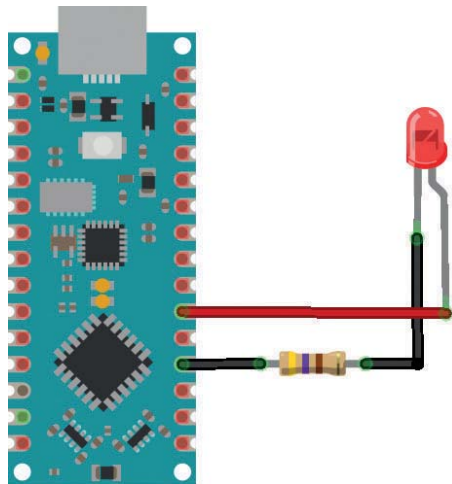
Slika 4-4: Radni ciklus

(prilagođeno sa slike Thewrightstuffa pod licencom <https://creativecommons.org/licenses/by-sa/4.0/deed.en>)

Dakle, možete pametno koristiti Arduino PWM za isporuku željene količine energije na LED ili na bilo koji drugi uređaj po vašem izboru.

## PWM sa Arduino Nano-m

Možete koristiti digitalne pinove D3, D5, D6, D9 i D10 za PWM i na Nano i Nano Every ploči. Digitalni pin D11 na Nano ploči se može koristiti za PWM. Isto ne važi za Nano Every ploču. Pošto želim da skica radi na obe ploče, neću koristiti digitalni pin D11 za PWM ni u jednoj skici. Povežimo anodu LED-a sa digitalnim pinom 6, a katodu sa uzemljenjem pomoću otpornika od 470 oma, kao što je prikazano na slici 4-5.



fritzing

Slika 4-5 LED povezan sa digitalnim pinom D3

Moraćete da koristite funkciju **analogWrite()** koja prihvata PWM broj pina i intenzitet PWM primenjenog kao argumente. Intenzitet je od 0 do 255. Ako je intenzitet 0, 63, 127, 255 to znači 0%, 25%, 75%, 100% radnog ciklusa, tim redom. Osim toga, možete imati srednji procenat vrednosti pomoću sledeće formule:

$$\text{procenat radnog ciklusa} = (\text{intenzitet} / 255) \times 100$$

Hajde da napišemo i postavimo skicu za demonstraciju funkcije **analogWrite()**:

```
prog00.ino
int pwm_pin_d3 = 3;
int signal_duration = 2;
void setup()
{
  pinMode(pwm_pin_d3, OUTPUT);
}
void loop()
{
  for(int i=0; i<=255; i++)
```

## Poglavlje 5 - Iscrtavanje geometrijske „umetnosti“ na eksternom displeju

U prethodnom poglavlju ste naučili da radite sa PWM-om (Pulse Width Modulation). Koristili ste PWM da biste izbleđeli normalnu i RGB LED. Štaviše, koristili ste ga za kontrolu SG90 servo motora. U ovom trenutku, trebalo bi da razumete koncept i primenu PWM-a.

U ovom poglavlju ćete se upoznati sa modulom eksternim displejom zasnovanim na Ilitek 9225 drajveru IC. Takođe ćete naučiti da kreirate „umetnost“ (ili „umetničko delo“) pomoću eksternog displeja.

### Ilitek 9225 drajver IC i displej

Uređaj označen kao “ILI9225” je drajver IC za displej koji je kreirao Ilitek Taiwan. Možete ga povezati na bilo koju TFT LCD ploču sa maksimalnom rezolucijom od  $176 \times 220$  (piksela). Podržava 16 bita po pikselu RGB boja, tj. 262144 različite boje. Čip ima 87120 bajtova RAM-a za grafičke podatke i spreman je za komunikaciju pomoću Nano-ovog SPI interfejsa.

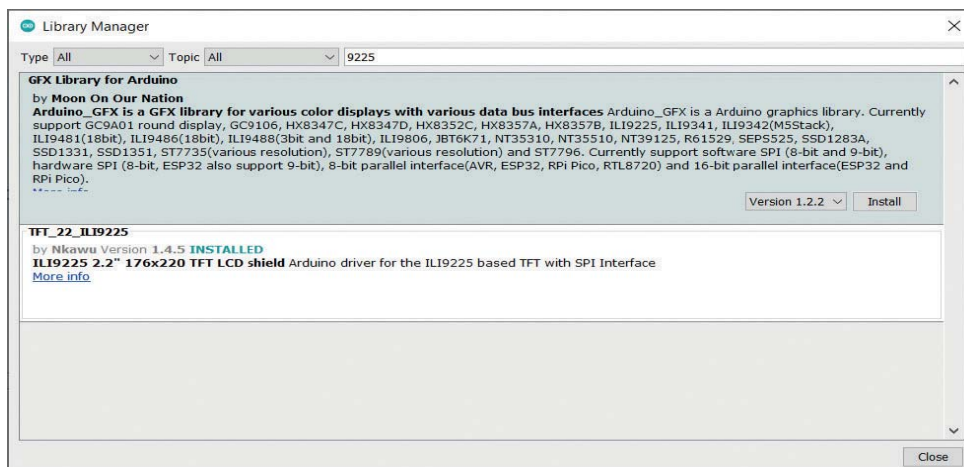
Možete kupiti **2,2-inčni 166x220 TFT LCD sa ugrađenim 9225 drajverom** kao jedan paket. Mnogo puta, uređaj se isporučuje sa ugrađenim čitačem microSD kartica koji ima zasebne pinove. Ovaj displej možete potražiti u onlajn prodavnicama ili prodavnicama elektronike za hobiste. Tabela sa podacima o IC drajveru se nalazi na adresi:

<https://www.displayfuture.com/Display/datasheet/controller/ILI9225.pdf>

Hajde da vidimo kako da povežemo ovu zanimljivu jedinicu sa Nano pločom. U sledećoj tabeli su objašnjeni pinovi modula ekrana koji su mapirani sa pinovima Nano ploče.

Pinovi modula displeja	Pinovi Arduino ploče
VCC	+5 V
GND	GND
GND	nema veze
NC	nema veze
NC	nema veze
NC	nema veze
SCK	A0
SDA	A1
RS	A2
RST	A3
CS	A4

Moraćete da instalirate biblioteku za ovaj modul da bi funkcionisao sa vašim Arduino. Otvorite **Library Manager** iz menija **Sketch** i potražite **9225**. Na slici 5-1 je snimak ekrana otvorenog menija.



Slika 5-1 Instalacija biblioteke TFT\_22\_ILI9225

Instalirajte biblioteku **TFT\_22\_ILI9225** autora **Nkawu**. Dokumentaciju za biblioteku možete pročitati na adresi: [https://github.com/Nkawu/TFT\\_22\\_ILI9225](https://github.com/Nkawu/TFT_22_ILI9225)

ili na Wikipage na adresi: [https://github.com/Nkawu/TFT\\_22\\_ILI9225/wiki](https://github.com/Nkawu/TFT_22_ILI9225/wiki)

### Programiranje displeja

2,2-inčni TFT LCD 9225 displej je jedan od najsvestranijih displeja dostupnih za prikaz statične grafike i jednostavnih animacija. Hajde da saznamo kako da ga programiramo pomoću Nano-a. Naučićete da pišete kod blok po blok, a ja ću objasniti svaku liniju. Kreirajte novi direktorijum (fasciklu) za skice u ovom poglavlju i počnite da pišete sledeći kod u novoj skici koja se zove **prog00.ino**.

Koristićete **SPI** (Serial Peripheral Interface) protokol za rad sa displejom. Takođe ćete morati da koristite nedavno instaliranu biblioteku ILI9225. Dakle, sada ćemo da uvezemo obe biblioteke:

```
#include "SPI.h"
#include "TFT_22_ILI9225.h"
```

Hajde da koristimo direktive pretprocesora da bismo definisali nekoliko makroa za dodeljivanje pinova, na sledeći način:

```
#define TFT_CS  A4
#define TFT_RST A3
#define TFT_RS  A2
#define TFT_SDI A1
#define TFT_CLK A0
#define TFT_BRIGHTNESS 200
```

Imajte na umu da kod proizvođača pinovi mogu imati različite nazive. Hajde da kreiramo objekat koji odgovara TFT displeju:

```
TFT_22_ILI9225 tft = TFT_22_ILI9225(TFT_RST, TFT_RS, TFT_CS, TFT_SDI, TFT_CLK,
TFT_BRIGHTNESS);
```

Zatim, definišite deo setup:

```
void setup()
{
```

Sada inicijalizujte TFT displej:

```
tft.begin();
```

i uključite pozadinsko svetlo i displej:

```
tft.setBacklight(true);
tft.setDisplay(true);
```

Zašto ne biste postavili boju pozadine:

```
tft.setBackgroundColor(COLOR_BLACK);
```

...i orijentaciju...

```
tft.setOrientation(0);
```

...a zatim inicijalizujte serijsku komunikaciju:

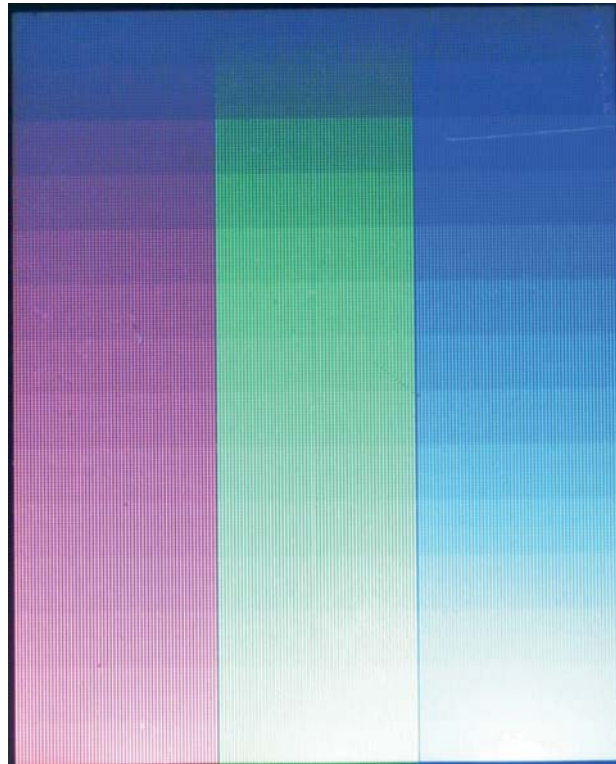
```
Serial.begin(9600);
```

Sledeći blok koda se koristi za detektovanje orijentacije (već smo je postavili):

```
Serial.print("Orientation: ");
switch (tft.getOrientation())
{
case 0:
Serial.print("Portrait");
```

Koristio sam Lumix GH5 u režimu ručnog fotografisanja da snimim rezultat. Pošto nisam veliki fotograf, fotografije možda nisu najboljeg kvaliteta. Štaviše, digitalni fotoaparati potrošačkog kvaliteta ne mogu verno da snime rezultat koji se proizvodi na digitalnom displeju. Na sreću, njegov rezultat izgleda mnogo bolje u stvarnosti.

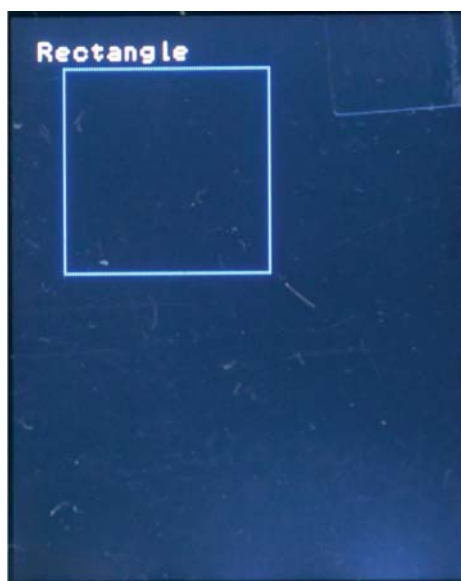
Nakon teksta, sistem prikazuje sliku gradijenta kreiranu pomoću pojedinačnih linija, kao na slici 5-3.



Slika 5-3 Gradijent boje

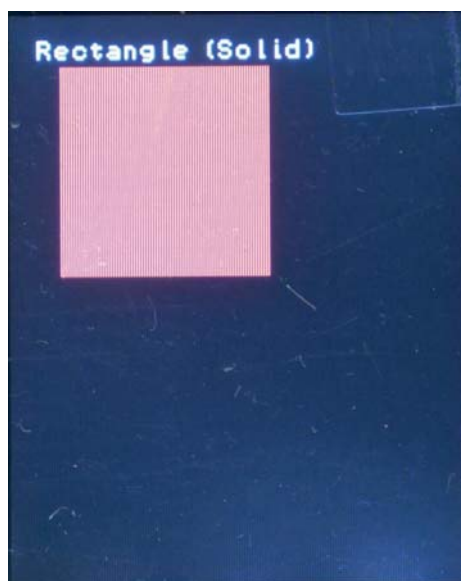
Prikazuje šuplji pravougaonik, kao na slici 5-3.





Slika 5-4 Šuplji pravougaonik

Zatim prikazuje čvrst/popunjen pravougaonik (slika 5-5).



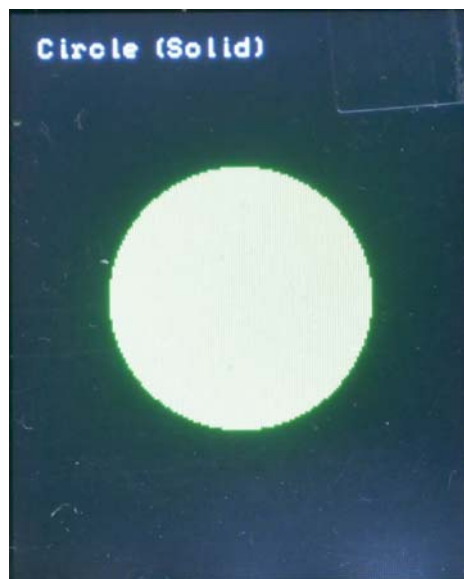
Slika 5-5 Puni (ispunjeni) pravougaonik

Ovo je šuplji krug; slika 5-6.



Slika 5-6 Šuplji krug

A konačna slika u nizu je puni (ispunjen) krug ili lopta; pogledajte sliku 5-7.



Slika 5-7 Puni (ispunjeni) krug

Obratite posebnu pažnju na to kako je krug ispunjen dodeljenom bojom. Pošto ste ovu sekvencu napisali unutar dela **loop()**, ona se ponavlja sve dok je Arduino ploča uključena. Sledimo sličan obrazac za većinu primera u ostatku poglavlja.

## Poglavlje 6 - Rad sa zujalicom i senzorom

U prethodnom poglavlju ste naučili kako da radite sa različitim ekranima sa pločama porodice Arduino Nano. Sada znate da koristite različite tipove ekrana sa Arduinoom.

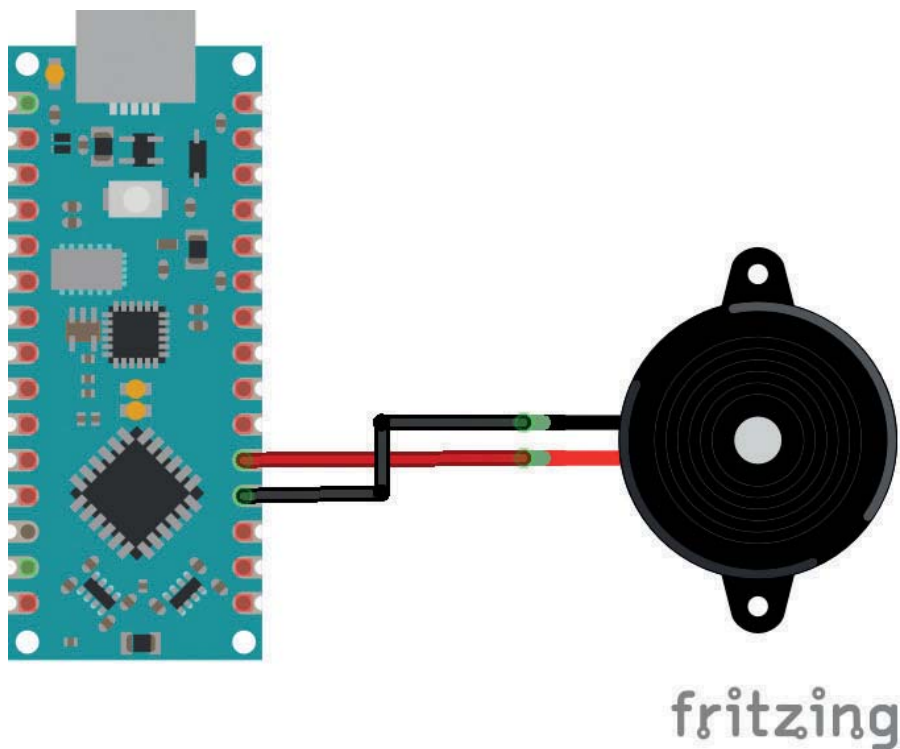
U ovom kratkom poglavlju ćete naučiti kako da radite sa nekoliko senzora i zujalicom. Ovo su teme koje ćete naučiti u ovom poglavlju:

- Rad sa zujalicom
- Rad sa džojstikom
- Rad sa senzorom temperature

Nakon što pročitate ovo poglavlje u celosti, znaćete da povežete vaš Arduino Nano sa perifernim uređajima kao što su zujalica, senzor temperature i džojstik.

### Rad sa zujalicom

Možete koristiti piezo zujalicu da proizvedete jedan po jedan ton određene frekvencije. Zujalica ima dva pina, jedan pozitivan i jedan negativan. Povežite pozitivni pin sa digitalnim izlaznim pinom nano ploče, a negativni pin sa GND-om. Hajde da kreiramo kolo kao na slici 6-1.



Slika 6-1 Piezo zujalica povezana sa Arduino Nano-m

Pozitivni pin zujalice (crvena žica) povezan je sa digitalnim I/O pinom na Nano-u, a negativni pin (crna žica) je povezan sa GND-om. Koristićete ovo kolo za sledećih nekoliko skica. Za ovo ćemo koristiti ugrađenu rutinu **tone()**. Generiše kvadratni talas određene frekvencije sa 50% radnog ciklusa. Možete generisati samo jedan ton po jedan ton na pinu i ploči. Drugim rečima, ako koristite određeni pin da emitujete ton, kada ponovo pozovete ovu funkciju sa nekim drugim pinom, ona neće funkcionisati. Ako već emitujete ton na pinu i ponovo pozovete ovu rutinu sa izmenjenom frekvencijom, ona će promeniti frekvenciju tona. Rutina ima tri argumenta. Prvi i drugi su obavezni, a definisani su kao izlazni pin i frekvencija. Treći argument je opcioni: to je trajanje u milisekundama tokom kojeg se ton reprodukuje na zujalici. Ako treći argument nije naveden, zujalica nastavlja da emituje navedeni ton osim ako se ne pozove druga rutina **noTone()**. U sledećoj skici je prikazana upotreba rutine **tone()**.

```
prog00.ino
const int piezzo_pin = 2;
void setup()
{
  Serial.begin(9600);
  pinMode(piezzo_pin, OUTPUT);
}
void loop()
{
  for (long i = 0; i <= 65535; i=i+500)
  {
    tone(piezzo_pin, i, 2000);
    Serial.println(i);
  }
}
```

Na Arduino Nano porodici ploča, možete generisati kvadratni talas frekvencije u opsegu od 31 Hz do 65535 Hz. Pošto tip podataka **int** ne može da obradi ovaj opseg, morate da koristite **long** tip za čuvanje frekvencija. Možemo napisati sličan kod, koristeći rutinu **noTone()** na sledeći način:

```
prog01.ino
const int piezzo_pin = 2;

void setup()
{
  Serial.begin(9600);
  pinMode(piezzo_pin, OUTPUT);
}

void loop()
{
  for (long i = 31; i <= 20000; i=i+500)
```

## Poglavlje 7 - Rad sa Arduino Nano 33 IoT-om

U posljednjem poglavlju ste proučavali kako da povežete senzor temperature, piezo zujalicu i džojstik sa Arduino Nano-m. U ovom poglavlju ćete naučiti da koristite još jednog člana Arduino Nano porodice, **Arduino Nano 33 IoT**. Ovo je lista tema koje ćete učiti i demonstrirati u ovom poglavlju:

- Uvod u Nano 33 IoT ploču
- Početak rada
- Rad sa bibliotekom WiFiNINA
- Server grupnog ćaskanja zasnovanog na Telnetu
- Pingovanje udaljenog servera
- Jednostavan veb klijent
- Rad sa satom u realnom vremenu
- Korišćenje senzora temperature DS18B20 zajedno sa RTC-om
- Vizuelizacija grafikona temperature pomoću ThingSpeak-a
- Programiranje ugrađenog IMU-a

Nakon što pročitate ovo poglavlje u celosti, znaćete kako da koristite Arduino Nano 33 IoT ploču.

### Uvod u Nano 33 IoT ploču

Sada ćete upoznati **Arduino Nano 33 IoT**. Ova ploča je pin-to-pin kompatibilna sa svim ostalim pločama mikrokontrolera u Arduino Nano porodici. IoT je skraćenica za **Internet Stvari**. Možete koristiti Nano 33 IoT za pokretanje IoT projekata koji imaju male otiške hardvera i funkcionalnosti. Ovim poglavljem obuhvaćeno je nekoliko malih IoT projekata zasnovanih na vebu, kao što su upravljanje LED-ovima sa veba i prikazivanje temperature pomoću veba. Takođe, možete jednostavno zameniti bilo koju drugu ploču iz Arduino Nano porodice Nano 33 IoT-om.

Hajde da razmotrimo hardverske specifikacije Nano 33 IoT-a. Nano 33 IoT se isporučuje u dve verzije: sa zaglavljem i bez zaglavlja pinova. Zbog nedostatka poluprovodnika i elektronskih komponenti, bilo mi je teško da ih nabavim u svom regionu dok sam pisao ovu knjigu (maj 2022.). Nadam se da ćete moći da ga nabavite bez problema kada se reši nedostatak poluprovodnika. Koristim verziju ove ploče sa zaglavljima pinova. Međutim, ako možete da nabavite samo pinove bez zaglavlja, lemljenje zaglavlja je lak posao.

Arduino Nano 33 IoT se isporučuje sa **Microchip SAMD21 Cortex®-M0+ 32-bitnim ARM MCU-om male snage** kao glavnim čipom, a njegova tabela sa podacima se nalazi na adresi:

[https://content.arduino.cc/assets/mkr-microchip\\_samd21\\_family\\_full\\_datasheetds40001882d.pdf](https://content.arduino.cc/assets/mkr-microchip_samd21_family_full_datasheetds40001882d.pdf).

Wi-Fi i Bluetooth povezivanje obezbeđuje **u-blox NINA-W102** modul, čija se tabela sa podacima nalazi na adresi:

[https://content.arduino.cc/assets/Arduino\\_NINA-W10\\_DataSheet\\_%28UBX-17065507%29.pdf](https://content.arduino.cc/assets/Arduino_NINA-W10_DataSheet_%28UBX-17065507%29.pdf).

Nano 33 IoT takođe ima čip **Microchip ATECC608A** za bezbednu komunikaciju. Možete proveriti tabelu sa podacima na adresi:

[https://content.arduino.cc/assets/microchip\\_atecc608a\\_cryptoauthentication\\_device\\_summary\\_datasheet-DS40001977B.pdf](https://content.arduino.cc/assets/microchip_atecc608a_cryptoauthentication_device_summary_datasheet-DS40001977B.pdf).

Štaviše, postoji i inercijalna merna jedinica čipa tipa **IMU LSM6DS3** i možete proveriti njegovu tabelu sa podacima na adresi:

[https://content.arduino.cc/assets/st\\_imu\\_lsm6ds3\\_datasheet.pdf](https://content.arduino.cc/assets/st_imu_lsm6ds3_datasheet.pdf).

U sledećoj tabeli su rezimirane druge istaknute karakteristike ploče.

<b>radni napon</b>	3.3 V
<b>ulazni napon (granična vrednost)</b>	21 V
<b>DC struja po I/O pinu</b>	7 mA
<b>brzina radnog takta</b>	48 MHz
<b>CPU fleš memorija</b>	256 KB
<b>SRAM</b>	32 KB
<b>EEPROM</b>	nema
<b>digitalni ulazni / izlazni pinovi</b>	14
<b>PWM pinovi</b>	11 (2, 3, 5, 6, 9, 10, 11, 12, 16, 17, 19)
<b>UART</b>	1
<b>SPI</b>	1
<b>I2C</b>	1
<b>analogni ulazni pinovi</b>	8 (ADC 8/10/12-bitni)
<b>analogni izlazni pinovi</b>	1 (DAC 10-bitni)
<b>LED-ovi, ugrađeni</b>	13
<b>dužina</b>	45 mm
<b>širina</b>	18 mm
<b>težina</b>	5 gr (za zaglavljima)

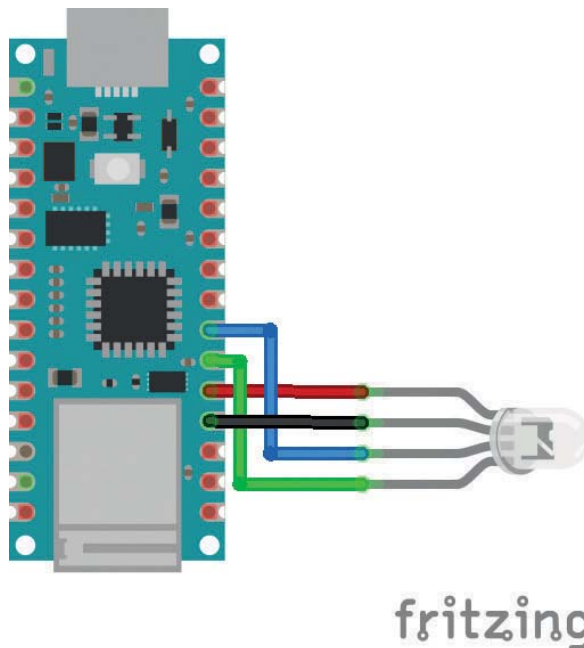
Možete da preuzmete Fritzing deo sa adrese:

<https://content.arduino.cc/assets/Arduino%20Nano%2033%20IOT.fzpz>.

Ako kliknete na hiperlinkove, možete poslati odgovarajuće podatke na veb server i promeniti stanje LED-ova na ploči. Potrebno je nekoliko sekundi da promene budu aktivne. Ako pažljivo posmatrate, možete uočiti izmenjeni URL u traci za adresu pregledača.

Možete da povežete relej na digitalni I/O pin 13 da biste kontrolisali fluorescentnu cev ili ventilator. Budite oprezni dok radite sa naponima naizmenične struje – strujni udar je u većini slučajeva fatalan.

Ovu aplikaciju možete proširiti RGB LED-om sa zajedničkom katodom. Pogledajte kolo prikazano na slici 7-14.



Slika 7-14 Pokrenuti RGB LED sa zajedničkom katodom

Možete omogućiti njegov rad na lokalnoj mreži, pomoću sledećeg programa:

```
prog02.ino  
#include <SPI.h>  
#include <WiFiNINA.h>  
char ssid[] = "TP-Link_710E";  
char pass[] = "internet1";  
int status = WL_IDLE_STATUS;  
WiFiServer server(80);  
WiFiClient client;  
  
const int RED = 2;  
const int GREEN = 3;
```

```

const int BLUE = 4;
void setup()
{
  Serial.begin(9600);
  pinMode(RED, OUTPUT);
  pinMode(GREEN, OUTPUT);
  pinMode(BLUE, OUTPUT);
  if (WiFi.status() == WL_NO_MODULE)
  {
    Serial.println("Communication with WiFi module failed!");
    while (true);
  }
  while (status != WL_CONNECTED)
  {
    Serial.print("Attempting to connect to Network named: ");
    Serial.println(ssid);
    status = WiFi.begin(ssid, pass);
    delay(10000);
  }
  server.begin();
  Serial.print("SSID: ");
  Serial.println(WiFi.SSID());
  IPAddress ip = WiFi.localIP();
  Serial.println("Started the web server on port 80.");
  Serial.print("IP Address: ");
  Serial.println(ip);
  long rssi = WiFi.RSSI();
  Serial.print("Signal strength (RSSI):");
  Serial.print(rssi);
  Serial.println(" dBm");
  Serial.print("Open this address in a browser http://");
  Serial.println(ip);
}
void loop()
{
  client = server.available();
  delay(1000);
  if (client) {
    Serial.println("A new client has connected to the server...");
    String currentLine = "";
    while (client.connected())
    {
      if (client.available())
      {
        char c = client.read();
        Serial.write(c);

```