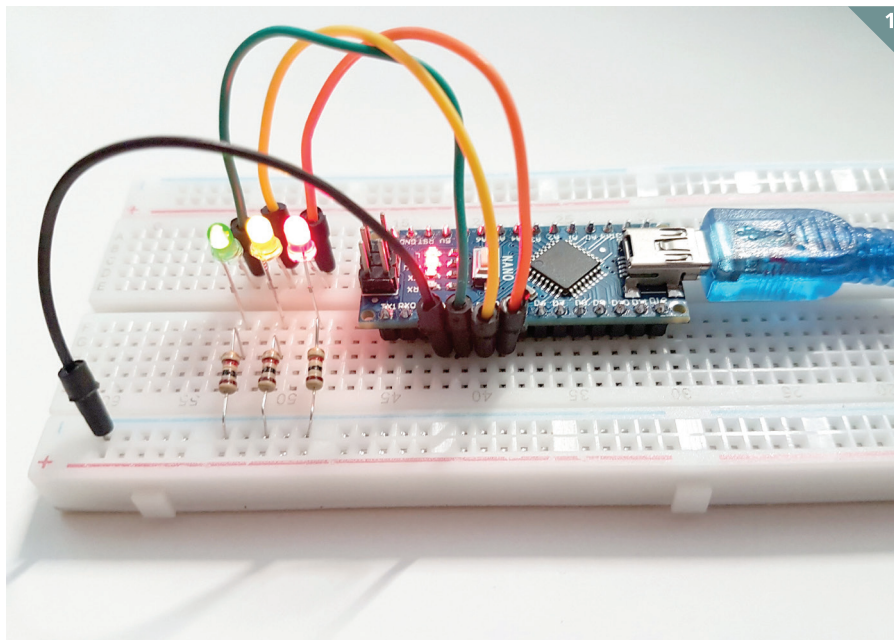


## PRVI KORAKI V ARDUINO – SEMAFOR

▼ Milan Gaberšek in Slavko Kocijančič

**N**a prehodu za pešce običajno naletno na semafor, ki sveti rdeče ali zeleno, medtem ko imajo semaforji za avtomobile, avtobuse in tovornjake še rumeno luč. V prejšnji številki revije smo krmilnik Arduino povezali s svetlečo diodo in jo s pomočjo programa vklopjali in izklopjali. Podobno bomo storili tudi tokrat, le da bomo pri tem uporabili tri svetleče diode različnih barv in jih priključili na tri različne izhode krmilnika.



### Material

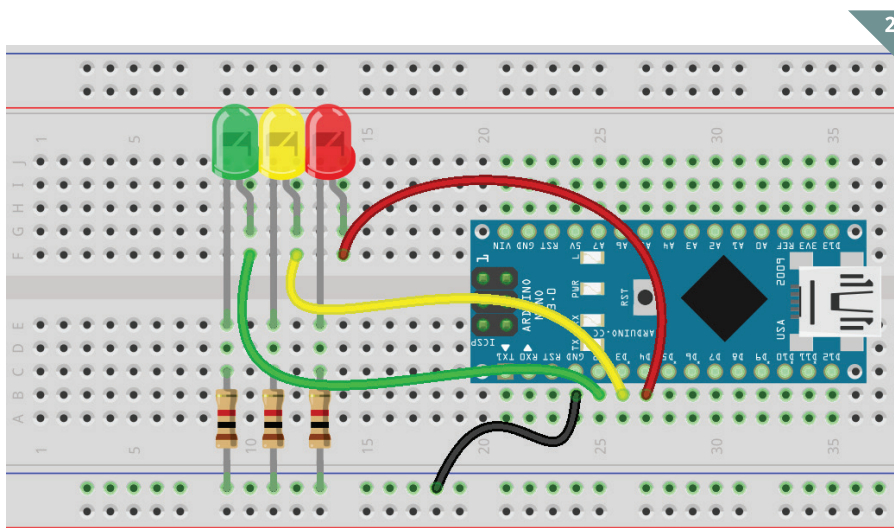
- krmilnik Arduino Nano ali podoben,
- USB-kabel za povezavo mikrokrmilnika (mini USB) z računalnikom,
- prototipna ploščica (angl. breadboard),
- štiri vezne žice različnih barv (zaradi preglednosti najboljše črna, zelena, rumena in oranžna),
- tri upore vrednosti 1 kΩ (rjava, črna, rdeča, zlata),
- tri svetleče diode (zeleno, rumeno, rdeča).

### Orodja in pripomočki

- osebni računalnik z nameščenim operacijskim sistemom Windows, Linux ali Mac OS,
- Arduino IDE, integrirano programsko razvojno okolje, ki je brezplačno dostopno na spletni strani [www.arduino.cc](http://www.arduino.cc).

### Izvedba

Pri vstavljanju elementov si lahko pomagamo s fotografijo dejanskega vezja (slika 1). Ker je način povezave na fotografiji vezja pogosto težko razbrati, je vezje priporočljivo narisati s pomočjo računalnika (slika 2). Če želimo tovrstna vezja risati sami, si lahko namestimo brezplačno različico programa Fritzing ([www.fritzing.com](http://www.fritzing.com)).



org), ki ponuja skoraj vse, kar lahko priključimo na krmilnik Arduino. Na voljo so tudi nekateri drugi krmilniki. Arduino Nano je skrit v podmeniju, zato ga je najlažje najti s pomočjo lupe. V iskalniku za to uporabimo ključno besedo nano.

Še bolj pregledna, a za začetnike hkrati manj privlačna oblika je narisana elektriška shema vezja (slika 3), pri kateri je narisano le najnujnejše, največkrat le elektronski elementi in povezave med njimi. Na ta način prihranimo pri prostoru, hkrati pa je tudi jasno razvidno, katere elemente vsebuje shema in kako so povezani med seboj. Vezne žice so risane na poseben na-

čin. Običajno se, če je le mogoče, med seboj ne križajo in so risane čim bolj pravokotno in vzporedno glede na ostale elemente v shemi. Mesto, kjer so vezne žice povezane med seboj, označimo s piko. Med brezplačnimi programi za risanje električnih shem sta brezplačna recimo KiCad ([www.kicad-pcb.org](http://www.kicad-pcb.org)) in EasyEda (<https://easyeda.com>).

Oglejmo si izdelavo vezja. Najprej na prototipno ploščico pritrdimo krmilnik Arduino Nano. Sledi namestitev uporov 1 kΩ. Pri tem eno od obeh nožic upora vstavimo v napajalno vrstico prototipne ploščice, ki jo pozneje s črno vezno žičko povežemo z nožico GND Arduino (slika 2).



- TN 1 motorni letalski RV-model basic 4 star
- TN 2 RV-jadrnica lipa I
- TN 3 RV-jadrni model HOT-94
- TN 4 polmaketa letala cessna 180
- TN 5 RV-model katamarana KIM I
- TN 6 Timov HLG-jadrni RV-model za spuščanje iz roke
- TN 7 RV-jadrni model HOT-95
- TN 8 Timov HLG-2-jadrni RV-model za spuščanje iz roke
- TN 9 tomy-E, elektromotorni jadrni RV-model
- TN 10 polmaketa lovškega letala polkarpov I-15 bis
- TN 11 jadrni RV-model gita
- TN 12 ragoon HLG-3
- TN 13 akrobat 40, trenajni motorni RV-model
- TN 14 maketa vodnega letala utva-66H
- TN 15 RV-model trajekta

- TN 16 spiffire, RV polmaketa za zračni boj
- TN 17 trener 40, trenajni motorni RV-model
- TN 18 lupu, elektromotorni RV-model
- TN 19 P-40 warhawk, RV-polmaketa za zračni boj
- TN 20 poteputh, RV-model motorne jahte
- TN 21 bambi, šolski jadrni RV-model
- TN 22 slovenka, RV-jadrnica metrskega razreda
- TN 23 e-trainer, trenajni RV-model z električnim pogonom
- TN 24 P-51 B/D mustang, RV-polmaketa za zračne boje
- TN 25 messerschmitt Bf-109E, RV-polmaketa za zračni boj
- TN 26 RV-polmaketa Aerona L-3
- TN 27 fokker E III, RV-polmaketa park-fly
- TN 28 vektar, RV-model z električnim pogonom v potniški izvedbi

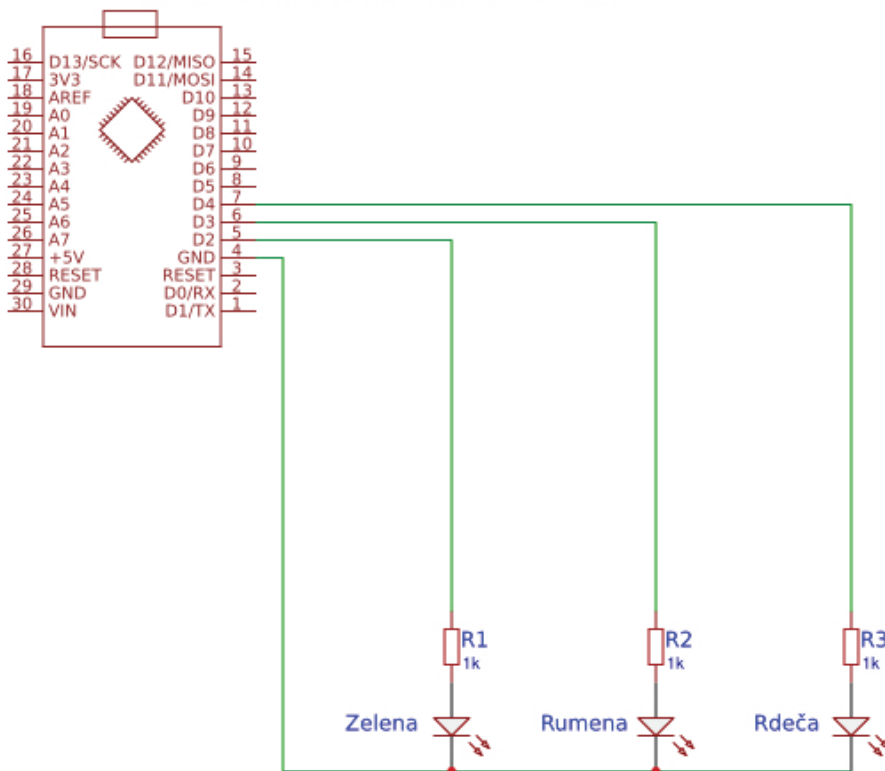
- TN 29 Eifflov stolp, 1 m visoka maketa iz vezane plošče
- TN 30 maketa bagra CAT 262
- TN 31 RV motorni letalski model z električnim pogonom orion
- TN 32 maketa hitre patrolne ladje SV Ankaran

**6,50 €\***

\*Cena posameznega načrta, k čemu pristojejo poštni stroški

Naročila sprejemamo na:  
**ZOTKS, revija TIM,**  
Zaloška 65, 1000 Ljubljana,  
tel.: 01/479-02-20,  
e-pošta: revija.tim@zotks.si.

Arduino nano, USB povezava z računalnikom



Nato namestimo zeleno, rumeno in rdečo svetlečo diodo. Pri tem moramo paziti, da jih pravilno namestimo, saj drugače ne bodo delovale. Katodo (krajša nožica oziroma tista, pri kateri je na ohišju svetleče diode zareza) vstavimo k upor, anodo (daljša nožica) pa s pomočjo vezne žičke ustrezne barve povežemo z ustreznim digitalnim izhodom Arduinoa. Če so upori večji od tistih na sliki, to pomeni, da prenesejo večje obremenitve, kar pa ne bo vplivalo na delovanje našega vezja. Lahko uporabimo tudi večje diode, le da bomo morali pri vstavljanju teh na preizkusno ploščico narediti nekoliko večji razmik. Nenazadnje lahko uporabimo tudi kak drug krmilnik Arduino, na primer Arduino Uno, ki pri preprostih vezjih praviloma deluje enako kot Arduino Nano. Njihovo medsebojno primerjavo si lahko ogledamo na spletni strani <https://www.arduino.cc/en/products/compare>.

Zeleno diodo povežemo z digitalnim izhodom z oznako D2, rumeno z D3 in rdečo z D4. Zagotovo se boste vprašali, zakaj ne uporabimo digitalnih izhodov D0 in D1. Na krmilniku Arduino Nano ju pravzaprav ne moremo najti. Razlog je v tem, da sta izhoda na krmilniku običajno označena s TX1 in RX0. Ta pina sta namenjena predvsem komunikaciji z drugimi napravami (tudi povezavi prek USB), zato ju ob kopici drugih možnosti ni priporočljivo uporabljati kot digitalni izhod.

Program za krmiljenje semaforja je precej preprost. Kljub temu pa je priporočljivo najprej preveriti, ali smo vse elemente pravilno povezali. Program za preizkus diod je naslednji:

```
//
// Program Arduino testiranje diod
//

// Konstante za delo s pini digitalnih izhodov
const int ledZelena = 2;
const int ledRumena = 3;
const int ledRdeca = 4;

void setup() {
  // Arduino povemo, da delamo z
  // digitalnimi izhodi (OUTPUT)
  pinMode(ledZelena, OUTPUT);
  pinMode(ledRumena, OUTPUT);
  pinMode(ledRdeca, OUTPUT);
}

void loop() {
  // Nastavitev digitalnih izhodov, pri
  // čemer HIGH vklopi, LOW izklopi
  digitalWrite(ledZelena, HIGH);
  digitalWrite(ledRumena, HIGH);
  digitalWrite(ledRdeca, HIGH);
  // Pred nadaljevanjem ukazov počakaj
  // za navedeno število milisekund
  delay(400);
  digitalWrite(ledZelena, LOW);
  digitalWrite(ledRumena, LOW);
  digitalWrite(ledRdeca, LOW);
  delay(400);
}
```

Najprej s spremenljivkami določimo, na katerih pinih Arduinoa smo priključili posamezne diode. V podprogramu void setup() {} določimo, da bomo posamezne pine uporabili kot digitalne izhode. V podprogramu

void loop() {} nato z ukazom digitalWrite(ledZelena, HIGH); krmilniku povemo, naj stanje na izhodu postavi na logično visoko stanje HIGH, s čimer zelena svetleča dioda zasveti. Z ukazom delay() za dano število milisekund obdržimo nastavljeno stanje. Nato z ukazi svetleče diode izklopimo, nekoliko počakamo, se vrnemo na začetek podprograma in ga ponovno izvedemo. Ker se vse ponavlja v nekakšni zanki, diode utripajo. Če katera od diod ne utripa, preverimo povezave in tudi to, ali smo diodo pravilno vstavili (katoda pri upor).

Ko vse normalno deluje, lahko program predelamo tako, da se diode prižigajo kot semafor. Program je pred tem priporočljivo shraniti pod drugim imenom.

```
//
// Program Arduino semafor
//

const int ledZelena = 2;
const int ledRumena = 3;
const int ledRdeca = 4;

void setup() {
  pinMode(ledZelena, OUTPUT);
  pinMode(ledRumena, OUTPUT);
  pinMode(ledRdeca, OUTPUT);
}

void loop() {
  digitalWrite(ledZelena, HIGH);
  digitalWrite(ledRumena, LOW);
  digitalWrite(ledRdeca, LOW);
  delay(3000);
  digitalWrite(ledZelena, LOW);
  digitalWrite(ledRumena, HIGH);
  digitalWrite(ledRdeca, LOW);
  delay(1000);
  digitalWrite(ledZelena, LOW);
  digitalWrite(ledRumena, LOW);
  digitalWrite(ledRdeca, HIGH);
  delay(3000);
  digitalWrite(ledZelena, LOW);
  digitalWrite(ledRumena, HIGH);
  digitalWrite(ledRdeca, HIGH);
  delay(1000);
}
```

## Zaključek

Izdelava semaforja je ena od zanimivih nalog, ki jih lahko rešimo s krmilnikom Arduino. Če želimo, lahko na katerega od preostalih digitalnih izhodov D5 do D12 namestimo še kako dodatno svetlečo diodo, vendar pri tem ne pozabimo uporabiti enakega predupora kot pri tistih svetlečih diodah, ki smo jih uporabili za semafor. Poljubno lahko spreminjamo tudi zaporedje in način delovanja ukazov znotraj podprograma void loop() {}, s čimer lahko ustvarimo različne svetlobne učinke. Če želimo, lahko dodamo še več svetlečih diod, za kar lahko uporabimo tudi digitalni izhod D13 in celo pine z oznakami od A0 do A7. O uporabi teh pinov bomo podrobneje spregovorili v enem od prihodnjih nadaljevanj. Za izziv pa lahko tokrat dodamo še semafor za pešce ali semafor na križišču dveh cest.