PRVI KORAKI V ARDUINO – SEMAFOR ZA PEŠCE S STIKALOM

Milan Gaberšek in Slavko Kocijančič

Ko se kot pešci približamo prehodu čez cesto, običajno na semaforju pritisnemo tipko. S tem sprožimo časovnik, ki čez čas izklopi rdečo in vklopi zeleno luč. Ta sveti samo nekaj časa, nato ponovno zasveti rdeča luč. V tem prispevku bomo s pomočjo krmilnika Arduino sestavili tako vezje in napisali program, da bo napravica oponašala delovanje pravega semaforja. Še več, lahko bomo celo nastavili čas po pritisku tipke, v katerem naj še sveti rdeča, ter koliko časa naj bo prižgana zelena luč.

Material

- krmilnik Arduino Nano ali podoben,
- USB-kabel za povezavo mikrokrmilnika (mini USB) z računalnikom,
- prototipna ploščica (angl. breadboard),
- tipka, po možnosti mikrotipka, ki jo je preprosto namestiti na prototipno ploščico,
- vezne žičke (najbolje rdeča, vijoličasta, zelena in dve črni),
- dva upora vrednosti 1 kΩ (rjava, črna, rdeča, zlata),
- dve svetleči diodi (zelena in rdeča).

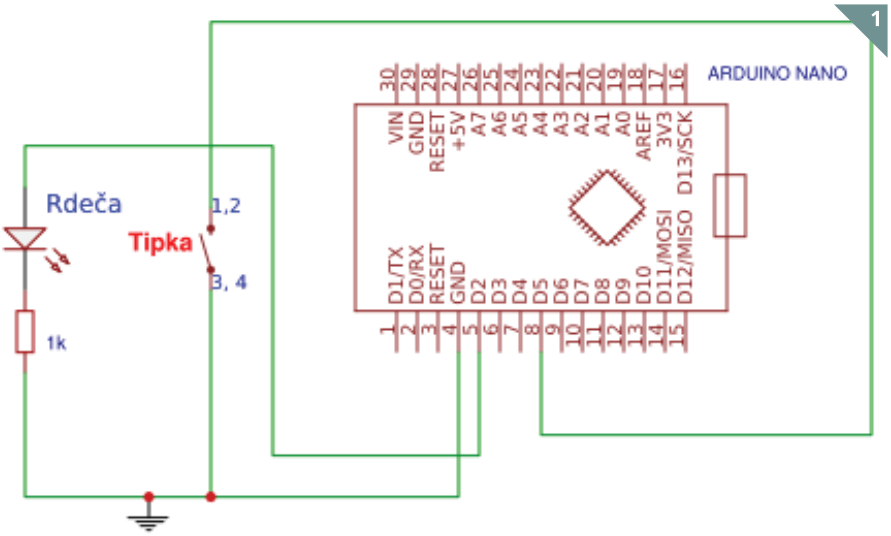
Orodja in pripomočki

- 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*.

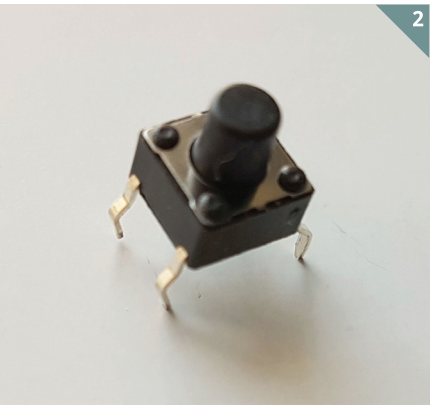
Izvedba

Tokrat se bomo naučili, kako lahko Arduino pripravimo do tega, da se bo odzival na pritisk tipke. Na podoben način bi lahko uporabili tudi stikalo. Razlika med njima je ta, da ostane stikalo sklenjeno tudi po tem, ko ga nehamo pritiskati, medtem ko se pri tipki stik prekine. V primeru semaforja za pešce bomo uporabili tipko, saj bi se v primeru uporabe stikala pešec moral vrniti nazaj in ga izklopiti.

Delovanje tipke oziroma stikala je zelo preprosto. V resnici bi bil učinek enak, če bi zgolj sklenili oziroma razklenili prekinjeno žičko, le da je tukaj vse varno zaprto v ohišje. Če se zgodi, da tipke oziroma stikala nimamo, ju lahko mirno nadomestimo kar z navadno žičko. Glede na električno shemo (slika 1) bomo



1

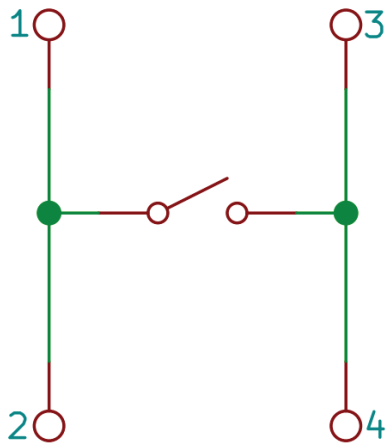

2

najprej preizkusno povezali tipko in rdečo svetlečo diodo. V prispevku bo uporabljena mikrotipka (slika 2), ki je majhna in na preizkusni ploščici zaseda malo prostora. Pri prvi uporabi nas mikrotipka preseneti, saj ima štiri priključne nožice, kar se da lepo razbrati iz diagrama tipke (slika 3). Te ima zato, da jo lahko trdno vstavimo v vezje, hkrati pa po dve v notranjosti povezani nožici zagotavljata zanesljivejše delovanje. Za lažje določanje, kateri nožici sta v notranjosti povezani, si pomagamo z utorom na spodnji strani mikrotipke (rumeni pas na levem delu slike 4). Lahko si predstavljamo, kot da utor prekinja stik, po pritisku na tipko pa poveže levo in desno stran mikrostikala. To moramo upoštevati tudi pri vstavljanju v preizkusno ploščico, pri čemer utor poravnamo v smeri enega od stolpcev preizkusne ploščice (rumeni pas na desnem delu slike 4). Eno od veznih žičk, najbolje črno, povežemo s pinom GND-krmilnika Arduino, drugo, po možnosti vijoličasto, pa s pinom D5, kot je razvidno z računalniške slike iz programa Fritzing (slika 5).

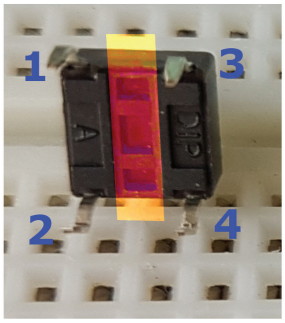
Na pin D2 z rdečo žičko priključimo še rdečo svetlečo diodo, nanjo upor 1 kΩ (barve obročkov so rjava, črna, rdeča in zlata), tega pa s črno žičko na pin GND. Seveda lahko tako za tipko kot za diodo uporabimo tudi kak drug pin, pri čemer pa moramo biti pozorni, da se nanj pravilno sklicujemo v programski kodi.

Sledi programiranje krmilnika Arduino. V integriranem programskem okolju Arduino IDE bomo zapisali naslednji program:

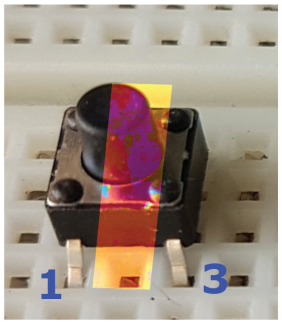
```
//
// Program Arduino semafor s tipko za pešce
- TEST
//

// Rdeča LED bo na nožici D2
const int ledRdeca = 2;
// Tipka bo na nožici D5
const int tipka = 5;

void setup() {
  pinMode(ledRdeca, OUTPUT);
  // Uporabili bomo notranji upornik pull-up,
  // zato ne potrebujemo zunanjega upora
  pinMode(tipka, INPUT_PULLUP);
}

void loop() {
  // Zaradi uporabe pull-up upora
  // je tipka v stanju HIGH,
  // po pritisku pa v stanju LOW
  if (digitalRead(tipka) == HIGH) {
   // LED sveti
  digitalWrite(ledRdeca, LOW);
} else {
   // LED ugasne
  digitalWrite(ledRdeca, HIGH);
  }
}
```

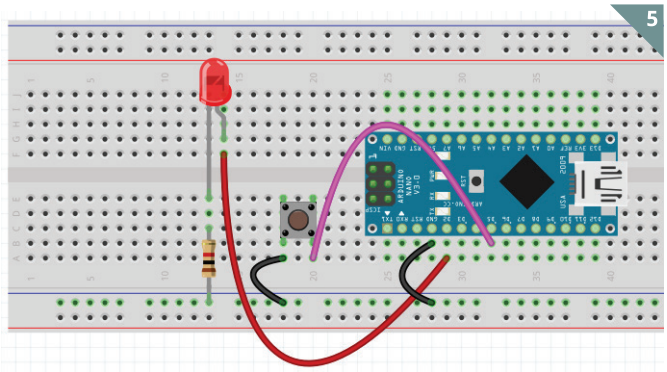


3



4





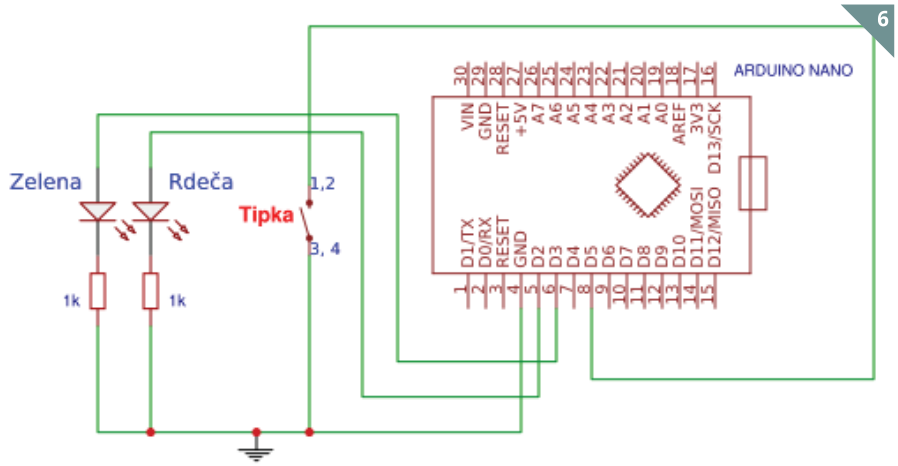
5

Najprej smo s spremenljivkama led Rdeca in tipka določili, na katera pina Arduina smo priklopili tipko in diodo. V podprogramu void setup() {} z ukazom pinMode(ledRdeca, OUTPUT); določimo, da bomo na pin s svetlečo diodo pošiljali signal (ukaz OUTPUT). Z ukazom pin Mode(tipka, INPUT_PULLUP); določimo, da bomo na ta pin prejemali neko stanje vhoda (ukaz INPUT). Priklop tipke bi morali izvesti s pomočjo dodatnega upora za dviganje nivoja napetosti (angl. pullup resistor), kar pa elegantneje in predvsem varneje storimo tako, da z ukazom INPUT_PULLUP uporabimo vgrajen notranji upor mikrokrmilnika. Tako dobimo s pomočjo ukaza digitalRead(tipka) vrednost nepritisnjene tipke logični HIGH, po pritisku na tipko pa logični LOW.

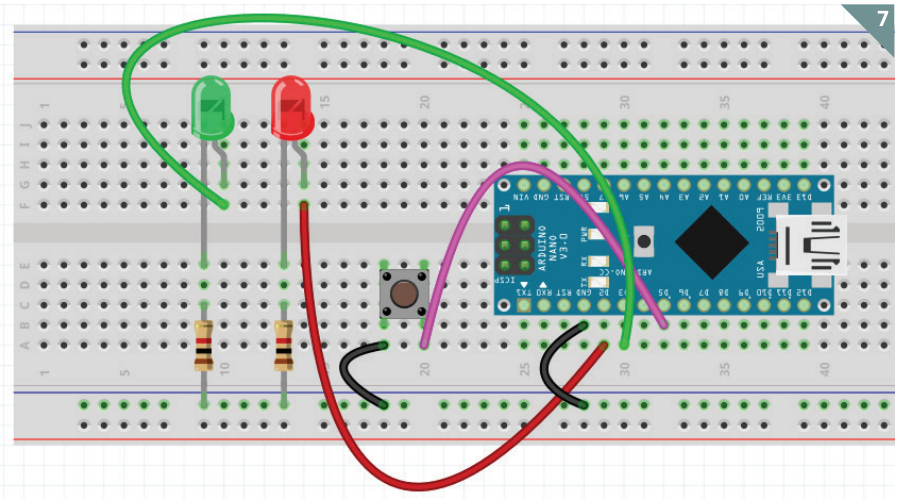
Za preverjanje stanja smo uporabili pogojni stavek if, ki odloča, kaj naj se zgodi, če je pogoj izpolnjen, ukaz else pa, če ni. V našem primeru z ukazom digitalRead(tipka) in dvojnim enačajem == preverimo, ali je tipka v stanju HIGH, torej, ali je ta pritisnjena. Tedaj se izvede vsa koda med zavitima oklepajema if pogoj { }. V nasprotnem primeru, če tipka ni pritisnjena, se izvede vsa koda med zavitima oklepajema ukaza else { }.

Program na običajen način prenesemo na krmilnik Arduino. Če je vse v redu, je rdeča svetleča dioda ugasnjena, po pritisku na tipko pa zasveti.

Zdaj se lotimo končne izdelave semaforja za pešce s stikalom. Na pin D3 povežemo zeleno svetlečo diodo, na drugi konec povežemo upor 1 kΩ in tega nazaj na GND-krmilnik Arduino (glejte električno shemo na sliki 6 oziroma računalniško sliko vezja 7). Program je nadgradnja obstoječega programa, zato lahko ustvarimo kar njegovo kopijo in jo prilagodimo.



6



7

```
//
// Program Arduino semafor s tipko za pešce
//

// Rdeča LED bo na nožici D2
const int ledRdeca = 2;
// Zelena LED bo na nožici D3
const int ledZelena = 3;
// Tipka bo na nožici D5
const int tipka = 5;

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

void loop() {
  if ( digitalRead(tipka) == HIGH ) {
    // Ko je tipka izklopljena,
    // sveti le rdeča luč
    digitalWrite(ledRdeca, HIGH);
    digitalWrite(ledZelena, LOW);
  } else {
    // Po pritisku tipke počakaj 4 s
    delay(4000);
    // Zelena luč sveti, rdeča ugasne
    digitalWrite(ledRdeca, LOW);
    digitalWrite(ledZelena, HIGH);
    // Počakaj 8 sekund
    delay(8000);
  }
}
```

Ko bomo program zagnali, bo najprej zasvetila rdeča svetleča dioda. Po pritisku na tipko se bo ta čez nekaj sekund izklopila, vklopila pa se bo zelena luč in svetila določeno število sekund. Z nastavitvijo časa čakanja v programu lahko naredimo semafor, kakršnega smo si od nekdaj želeli.

Zaključek

Možnost uporabe tipke nam zelo razširi nabor projektov, ki jih lahko samostojno izvedemo s pomočjo krmilnika Arduino. Malo razmislimo in zagotovo se bomo kakšnega domislili. Eden takih je na primer izdelava preproste alarmne naprave s svetlečo diodo. Seveda lahko namesto tipke uporabimo tudi stikalo, s čimer je možnosti še več.