

PRVI KORAKI V ARDUINO – KDO BO HITREJŠI?

▼ Milan Gaberšek in Slavko Kocijančič

Predstavljen bo naprava, s pomočjo katere boste ugotovili, kdo je hitrejši oziroma kdo ima hitrejši refleks. Igra se začne tako, da nekdo od dveh tekmovalcev, še bolje kdo tretji, pridržijo tipko za začetek. Na tri, štiri, zdaj tipko izpusti in vklopi se časovnik, ki čaka naključno dolgo časa. Ko obe lučki zasvetita, se tekma začne. Zmaga tisti, ki prvi pritisne tipko in s tem ugasne lučko. Tudi tokrat bomo nalogo rešili s krmilnikom Arduino, tipkami in svetlečimi diodami.

Material

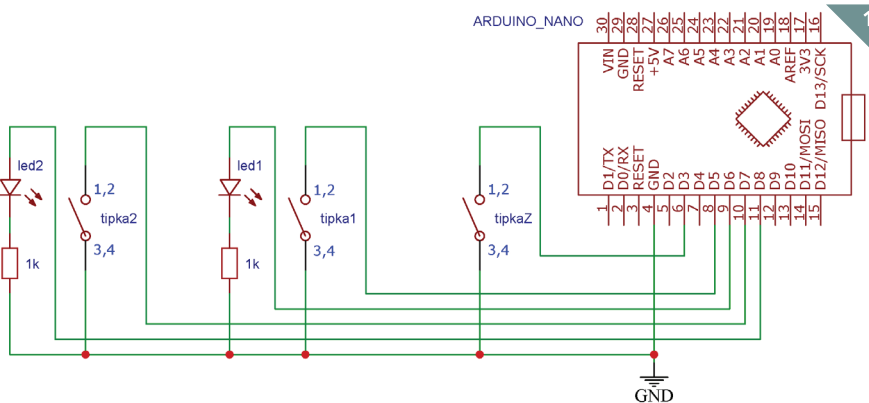
- krmilnik Arduino Nano ali podoben,
- USB-kabel za povezavo mikrokrmilnika (mini USB) z računalnikom,
- prototipna ploščica (angl. breadboard),
- tri mikrotipke (lahko tudi tipke),
- vezne žičke (najbolje rdeča, vijoličasta, zelena, bela, modra in štiri črne),
- dva upora vrednosti 1 kΩ (rjava, črna, rdeča, zlata),
- dve svetleči diodi (rdeče barve).

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

Iz električne sheme (slika 1) najlažje razberemo, kako povežemo posamezne elemente med seboj. Pri povezavi elementov si lahko pomagamo tudi z računalniško sliko iz programa Fritzing (slika 2). Pri sestavljanju ne pozabimo, da je pri svetleči diodi katoda označena z zarezo na kapici ohišja in da jo prek kiloohmskega upora povežemo na maso (GND) spodnje vrstice prototipne ploščice. Pri nekaterih prototipnih ploščicah je ta vrstica v sredini prekinjena, kar rešimo tako, da jo povežemo z dodatno vezno žičko. Pri namestitvi stikal pazimo, da je utor na spodnji strani ohišja vzporeden s stolpci prototipne ploščice. Če so vezne žice, ki so običajno priložene k zbirki Arduino, prekratke, lahko uporabimo tudi kako drugo žičko. Zelo dobro se obnesejo žičke iz debelega večžilnega zunanega telefonskega kabela, saj so te dovolj trde in različnih barv.



Ko je vse ustrezno povezano, se lahko lotimo programiranja krmilnika Arduino. Ker je program nekoliko daljši, je dosegljiv tudi na spletni strani www.drtn.si/tim.html.

```
//
// Program Arduino - Kdo bo hitrejši
//

// Nastavitev spremenljivk na ustrezne
// pine D3, D5, D6, D7 in D8
// za tipke in svetleči diodi
const int tipkaZ = 3;
const int tipka1 = 5;
const int led1 = 6;
const int tipka2 = 7;
const int led2 = 8;

// Spremenljivke boolean
// lahko postavimo na true
boolean bNobenaTipka = true;
// Privzeto so vrednosti false
boolean bIgralec1;
boolean bIgralec2;

// Ob vklopu nastavimo krmilnik Arduino
void setup() {
  pinMode(led1, OUTPUT);
  pinMode(led2, void);
  pinMode(tipkaZ, INPUT_PULLUP);
  pinMode(tipka1, INPUT_PULLUP);
  pinMode(tipka2, INPUT_PULLUP);
}

// Spremljaj krmilnik Arduino
void loop() {
  // Z zanko while čakamo, da nekdo
  // pritisne na tipko tipkaZ,
  // torej da ni več HIGH
  while (digitalRead(tipkaZ) == HIGH){};
  // LEDici izklopimo
  digitalWrite(led1, LOW);
  digitalWrite(led2, LOW);
  // Z zanko while čakamo, da je tipkaZ
  // izpuščena, torej da ni več v stanju LOW
  while (digitalRead(tipkaZ) == LOW) {};
  // Izberi naključno število med 3000 in 6000
  // ter čakaj izbrano milisekund
  delay( random(3000,6000) );
  // Vključimo LEDici
  digitalWrite(led1, HIGH);
  digitalWrite(led2, HIGH);
  // Ponavljanje, dokler je bNobenaTipka true
  while (bNobenaTipka == true) {
    // Preberi stanje tipk obeh igralcev,
    // pri čemer je ta ob pritisku false,
```

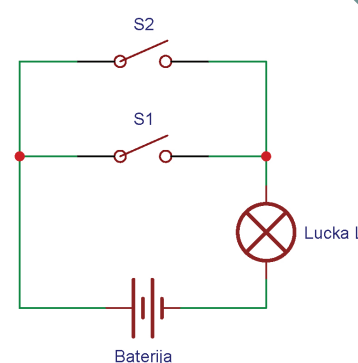
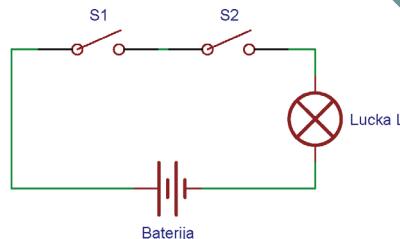
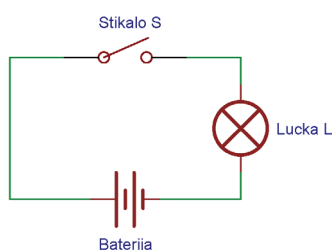
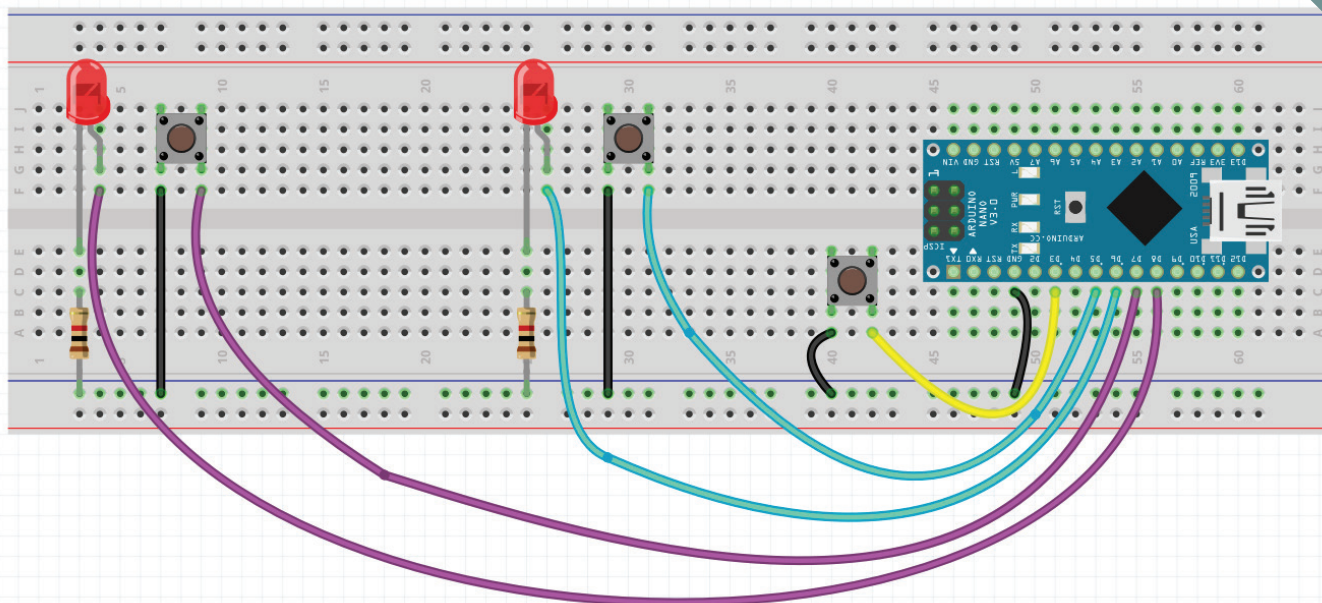
```
// drugače true
bIgralec1 = digitalRead(tipka1);
bIgralec2 = digitalRead(tipka2);
// Ob pritisku tipke kateregakoli igralca
// postane bNobenaTipka false (logični in)
bNobenaTipka = bIgralec1 && bIgralec2;
}
// Glede na stanje spremenljivk bIgralec1
// in bIgralec2 določimo zmagovalca
digitalWrite(led1, bIgralec1);
digitalWrite(led2, bIgralec2);
// Pripravimo se za novo igro
bNobenaTipka = true;
}
```

Ker smo tipko tipkaZ priklopili na pin D3, konstanti tipkaZ v programu določimo vrednost 3. To naredimo v vrstici kode `const int tipkaZ = 3;`, pri čemer int pomeni, da je konstanta tipa celo število (angl. integer). Podobno določimo konstanti tipka1 vrednost 5, led1 vrednost 6, tipka2 vrednost 7 in led2 vrednost 8.

Sledi najava spremenljivk programskega tipa boolean. Ta ima le dve možni vrednosti: true in false. Uporabimo lahko tudi vrednosti 1 in 0, kar pa ni priporočljivo, saj s tem izgubimo preglednost. Z vrednostjo true povemo, da je nekaj resnično, s false pa, da ni. Tako smo v prejšnjih prispevkih z ukazom `if` pogoj preverjali, ali je pogoj izpolnjen ali ni. Če je bil, bi to označili s true, v nasprotnem primeru pa s false. Obe vrednosti lahko uporabljamo tudi pri spremenljivkah. V našem programu tako definiramo spremenljivko `bNobenaTipka` in ji določimo vrednost true. Če vrednosti spremenljivki ne določimo, ima ta po privzetem začetno vrednost false, kar smo uporabili pri `bIgralec1` in `bIgralec2`.

V podprogramu `void setup() {}` glede na vrednosti spremenljivk nastavimo pine krmilnika Arduino kot izhod (angl. output) oziroma vhod (angl. input).

Srce programa je kot običajno v podprogramu `void loop() {}`. Najprej bomo uporabili zanko `while` (pogoj) { blok kode }. Blok kode med zavrtima oklepajema se izvaja, dokler je vrednost pogoj med oklepajema resnična. V našem primeru z `while (digitalRead(tipkaZ) == HIGH){}`; povemo, da naj program ne nadaljuje, dokler je tipkaZ v stanju HIGH, torej ni pritisnjena. Ob pritisku tipke tipkaZ se ta postavi v stanje LOW, kar pomeni, da pogoj ni več izpolnjen. Posledično zapustimo zanko `while` in z `digitalWrite` izklopimo obe LED



svetleči diodi (v nadaljevanju ledici). Ob vklopu Arduina in prvi izvedbi void loop() {} izklop ledic nima posebnega pomena, je pa nujen zaradi poznejšega pravilnega delovanja programa (kot že vemo, se blok kode podprograma void loop() {} ves čas ponavlja). Pri zanki while med zavitima oklepajema { } nismo zapisali nobene kode, saj zgolj čakamo na pritisek tipke.

Z vrstico while (digitalRead(tipkaZ) == LOW); čakamo, da bo uporabnik tipko, ki jo je pred tem pritisnil, spustil. Tedaj bo tipka prek ukaza digitalRead ponovno dobila vrednost HIGH, zato bomo zaradi neizpoljenosti pogoja zanko while zapustili. Z ukazom random(3000, 6000) generiramo naključno število med 3000 in 6000. To v povezavi z ukazom delay pomeni, da v odvisnosti od naključno izbranega števila na naslednji ukaz čakamo med tremi do šestimi sekundami. Šele po preteku časa vklopimo obe svetleči diodi, s čimer se tekma začne. Ta del je potreben zato, da igralca ne moreta vnaprej predvideti, kdaj bosta diodi začeli svetiti, s čimer je igra še bolj privlačna.

Razumevanje tretje zanke while (bNobenaTipka == true) {} je nekoliko zahtevnejše. Vrednost bNobenaTipka smo že pri najavi spremenljivke nastavili na true, zato je pogoj izpolnjen in izvede se pripadajoča koda znotraj zavinitih oklepajev. Najprej s pomočjo ukaza digitalRead v spremenljivki bIgralec1 in bIgralec2 shranimo vrednost stanja tipk obeh igralcev. V vrstici bNobenaTipka = bIgralec1 && bIgralec2; s pomočjo logičnega IN (oznaka &&) primerjamo obe vrednosti in re-

zultat shranimo v spremenljivko bNobenaTipka. Če je pritisnjena katera koli od tipk, je rezultat preverjanja logični false, drugače pa logični true. Ob ponovnem preverjanju resničnosti pogoja ukaza while zanko zapustimo (bNobenaTipka ima vrednost false) oziroma ponovimo (vrednost true).

Ko zapustimo zanko, glede na stanji spremenljivk bIgralec1 in bIgralec2 z ukazom digitalWrite nastavimo obe ledici. Ugasne ledica tistega igralca, katerega spremenljivka ima vrednost false, kar pomeni, da je bil hitrejši in postane zmagovalec. Spremenljivki bNobenaTipka nato priredimo vrednost true, da se ob naslednji ponovitvi void loop() {} podprogram ponovi na enak način, kot smo opisali, s čimer se začne nova igra.

Slabost programa je, da ne preverja, če je kateri od igralcev pritisnil tipko še pred vklopom obeh svetlečih diod. To je treba pojasniti tudi tekmovalcema, saj bo drugače na nepošten način zmagovalec znan še pred začetkom tekme. Slabost bi lahko odpravili z nekaj dodatnimi ukazi, vendar to presega namen prispevka in naj ostane izzivi.

Dolžni smo še nekaj dodatnih pojasnil glede logičnih stanj. Logični stanji sta dve: 0 ali 1. Za lažje razumevanje si bomo pomagali s preprosto električno shemo z lučko L in stikalom S (slika 3). Vsa možna stanja stikala in lučke lahko prikažemo z logično tabelo (tabela 1).

Pri tem logična 1 za stikalo pomeni, da je sklenjeno, za lučko pa, da sveti. Podobno logična 0 pomeni, da stikalo ni skle-

TABELA 1 TABELA STANJ	
Vhod S	Izhod L
0	0
1	1

njeno oziroma da lučka ne sveti. S tem so opisana vsa možna stanja stikala in lučke.

Na podoben način lahko prikažemo tudi logični IN, pri čemer imamo dve stikali S1 in S2 ter lučko L, kot je razvidno iz električne sheme (slika 4). Zapišimo tabelo vseh možnih stanj (tabela 2).

TABELA 2 LOGIČNI IN		
Vhod S2	Vhod S1	Izhod L
0	0	0
0	1	0
1	0	0
1	1	1

Kot lahko razberemo iz tabele, še lažje pa z razmislekom, lučka sveti le v primeru, če sta sklenjeni obe stikali.

Podobno lahko vpeljemo tudi logični ALI (slika 5). Hitro ugotovimo, da lučka ne sveti le, ko sta obe stikali izklopljeni. To razberemo tudi iz tabele stanj (tabela 3).

TABELA 3		
LOGIČNI ALI		
Vhod S2	Vhod S1	Izhod L
0	0	0
0	1	1
1	0	1
1	1	1

Predstavimo še tabelo stanj za logični NE oziroma za negacijo. Pri logičnem NE se logično stanje 0 na vhodu spremeni v logično 1 oziroma obratno iz logične 1 v logični 0. Tabela stanj je razvidna iz tabele 4.

TABELA 4	
LOGIČNI NE	
Vhod S	Izhod L
0	1
1	0

Na podoben način lahko zapišemo tudi tabelo stanj za naše vezje (tabela 5).

Iz tabele lahko hitro razberemo, da so vsa stanja tipk v našem vezju negirana. To pomeni, da ko tipka Tipka1 ni pritisnjena (logično stanje 0), dobimo na vho-

TABELA 5				
STANJE TIPK		LOGIČNA OPERACIJA IN		
Tipka 1	Tipka 2	Vhod bIgralec1	Vhod bIgralec2	Izhod bNobenaTipka
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0
1	1	0	0	0

du krmilnika Arduino zaradi uporabe notranjega pullup upora logično stanje 1. To vrednost nato shranimo v spremenljivko bIgralec1. Podobno storimo tudi za drugega igralca. Nato z logičnim IN primerjamo vrednosti obeh spremenljivk bIgralec1 in bIgralec2 in rezultat shranimo v spremenljivko bNobenaTipka (bNobenaTipka = bIgralec1 && bIgralec2;). bNobenaTipka postane true (oziroma 1) le, če nobeden od igralcev ni uspel pritisniti tipke (Tipka 1 in Tipka 2 sta obe v stanju 0, na vhidih pa dobimo zaradi notranjih pullup uporov logično 1). Kot smo že zapisali, pri programiranju običajno namesto števil 0 in 1 uporabljamo izraza false (neresnično) in true (resnično), saj laže razumemo njun pomen. Podoben

pomen imata tudi izraza LOW in HIGH, pri čemer ju uporabljamo predvsem v povezavi z dejanskimi elementi vezja priklopljenimi na krmilnik Arduino.

Zaključek

Program in vezje lahko precej preprosto prilagodimo tudi za tri ali več igralcev. Iz tega sledi, da lahko z razmeroma malo elementi, kot so tipka, svetleča dioda, krmilnikom Arduino ter nekaj programske kode, izdelamo že kar zanimive in uporabne projekte. Z uporabo novih elementov, kot so stikala, motorčki in različni senzorji, se možnosti še razširijo.



PRODAJNA MESTA REVİJE TIM

- AJDOVŠČINA**
 - 3DVA, d. o. o., Gregorčičeva 3
- CELJE**
 - Interspar IM 102 Celje, Mariborska 100
 - 3DVA, d. o. o., Prešernova 9
- ČRNOMELJ**
 - Delo prodaja, d. d., Ulica 21. oktobra 13
- DOMŽALE**
 - Trafika, Kolodvorska c. 11
 - Acron, PE Domžale, Mestni trg 1
- GROSUPLJE**
 - Delo prodaja, d. d., Adamičeva c. 11
- KOPER**
 - Interspar IM 105 Koper, Ankaranska c. 3 A
- KRANJ**
 - Delo prodaja, d. d., Bleiweisova
 - Interspar IM 108 Kranj, Qlandia, Cesta 1. maja 77
 - Delo prodaja, d. d., Glavni trg
- LAŠKO**
 - 3DVA, d. o. o., Mestna ul. 4
- LJUBLJANA**
 - Interspar IM 103 Lj. Vič, Jamova cesta 105
 - Rudnids, trgovina, Jurčkova cesta 225
 - 3DVA, d. o. o., Slovenska 29
 - Interspar IM 101 Lj, Citypark, trafika, Smartinska 152 G, BTC
 - 3DVA, d. o. o., Smartinska 156, BTC, hala A
 - Mercator, d. d. – Maxmarket, Trg republike 1
 - Delo prodaja, d. d., Žel. postaja - peron
- LOGATEC**
 - Mibo Modeli, d. o. o., Tržaška cesta 87b
- MARIBOR**
 - Interspar IM 111 Maribor, Qlandia, Cesta proletarskih brigad 100
 - Interspar IM 104 Maribor, Europark, Pobreška 18, Europark
- MURSKA SOBOTA**
 - Trgovina Salamon, Kocljeva ulica 1
 - Interspar IM 107 Murska Sobota, Nemčavci 1 D
- NOVA GORICA**
 - 3DVA, d. o. o., Kidričeva 20
- NOVO MESTO**
 - Interspar IM 113 Novo Mesto, Otoška cesta 5
- PTUJ**
 - Delo prodaja, d. d., Miklošičeva 3
 - Interspar IM 110 Ptuj, Ormoška cesta 15
- RADOVLJICA**
 - 3DVA, d. o. o., Avtobusna postaja
- SEVNICA**
 - Trafika, Trg svobode 1
- SEŽANA**
 - Acron, PE Sežana, Partizanska 48
- VIPAVA**
 - Delo prodaja, d. d., C. 18. aprila, Na trgu

NAROČILNICA

Nepreklicno (do pisne odpovedi) naročam revijo TIM. Cena letne naročnine je 33,75 EUR in že vključuje 9,5 % DDV. Naročnino bom poravnal po položnici.



Ime in priimek: _____

Naslov: _____

Kraj: _____

Poštna št.: _____

Telefon: _____

e-pošta: _____

Datum: _____

Podpis: _____

* Naročilo mora podpisati polnoletna oseba. Če je naročnik mladoletna oseba, mora naročilnico podpisati eden od staršev ali njegov zakoniti zastopnik.

Naročilnico, prosimo, pošljite na naslov: Revija TIM, Zveza za tehniško kulturo Slovenije, Zaloška 65, 1000 Ljubljana. Lahko jo pošljete po faksu na številko: 01/25 22 487 ali pa nam napišete elektronsko pismo na e-naslov: revija.tim@zotks.si. Za morebitne dodatne informacije nas pokličite na telefon: 01/4790 220. Več na www.tim.zotks.si.