

PRVI KORAKI V ARDUINO – PRENOSLJIVI KLAVIR

▼ Milan Gaberšek in Slavko Kocijančič

V tem prispevku bomo krmilnik Arduino uporabili kot preprost elektronski klavir, s katerim bomo lahko zaigrali tri osnovne note. Ker se za klavir spodobi, da deluje tudi brez povezave z računalnikom, ga bomo na koncu priklopili na standardno devetvoltno baterijo in mu s tem zagotovili prenosljivost.

Material

- krmilnik Arduino Nano ali podoben,
- USB-kabel (mini USB) za povezavo krmilnika z računalnikom,
- prototipna ploščica (angl. breadboard),
- mini piezozvočnik,
- tri tipke,
- vezne žičke (vijoličasta, tri modre, rdeča in šest črnih),
- 9-V baterija (ali podobna),
- konektor za priklop na 9-V baterijo.

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

Po električni shemi (slika 1) oziroma glede na sliko, narejeno s pomočjo programa Fritzing (slika 2), povežemo posamezne elemente. Pri tem podobno kot v prispevku Prvi koraki v Arduino – nastavljava frekvenca zvoka in melodija pesmi v prejšnji številki revije TIM uporabimo mini piezozvočnik, ki ga priklopimo neposredno na pin D3 krmilnika Arduino. Tipke priklopimo na pine D5, D6 in D7 (slika 2). Če imamo tipke s štirimi nožicami, bo v primeru napačne vezave tipka v kratkem stiku, kar je podobno, kot če bi bila ta ves čas pritisnjena. V takem primeru tipko iztaknemo, jo zavrtimo za 90° in jo ponovno vstavimo (več o tem v prispevku Prvi koraki v Arduino – semafor).

Program je dosegljiv na spletnem naslovu www.drti.si/tim.html, od koder ga lahko prenesemo na svoj računalnik.

```
// Zvočnik bomo priklopili na D3
const int pinZvocnik = 3;
const int pinC = 5;
const int pinD = 6;
const int pinE = 7;

// Frekvence za posamezne note, več not na
// https://www.arduino.cc/en/Tutorial/
// toneMelody
const int notaC = 262; // tudi C4
const int notaD = 294; // tudi D4
const int notaE = 330; // tudi E4

// Nastavitev vhodov za tipke
void setup() {
  pinMode(pinC, INPUT_PULLUP);
  pinMode(pinD, INPUT_PULLUP);
  pinMode(pinE, INPUT_PULLUP);
}

// Ponavlja
void loop() {
  if (digitalRead(pinC) == LOW) {
    // Če je pritisnjena tipka na pinC,
    // predvajaj ton s frekvenco note C
    tone(pinZvocnik, notaC);
  } else if (digitalRead(pinD) == LOW) {
    // Če je pritisnjena tipka na pinD,
    // predvajaj ton s frekvenco note D
    tone(pinZvocnik, notaD);
  } else if (digitalRead(pinE) == LOW) {
    // Če je pritisnjena tipka na pinE,
    // predvajaj ton s frekvenco note E
    tone(pinZvocnik, notaE);
  } else {
    // Drugače izklopi predvajanje tona
    noTone(pinZvocnik);
  }
}
```

V programu najprej prek konstant napovemo, iz katerih pinov bomo pridobivali informacijo o stanju tipk (nestisnjena, stisnjena) in prek katerega izhoda bomo krmilili mini piezozvočnik. Imena spremenljivk za pine tipk smo poimenovali kar z imeni tonov not, ki jih bodo te predvajale. Tako smo z najavo pinC = 5; najavili, da bomo na pinu D5 spremljali stanje tipke za noto C, ob pritisku katere bomo pozneje z ustreznimi ukazi prek zvočnika predvajali ton te note. Podobno smo najavili spremljanje tipk za noti D in E. Z najavo pinZvocnik = 3; pa smo najavili, da bomo prek pina D3 krmilili zvočnik. Z ukazom const int notaC = 262; smo določili frekvenco tona note C, podobno tudi za noti D in E.

V okviru void setup() {} z ukazom pinMode(pinC, INPUT_PULLUP); krmilniku Arduino najavimo, da bomo pinC uporabljali kot digitalni vhod, pri čemer z INPUT_PULLUP najavimo uporabo notranjega upora, zaradi katerega bomo lahko spremljali stanje tipke. Brez uporabe notranjega upora bi morali uporabiti zunanji upor, na katerem bi spremljali spremembo napetosti ob pritisku na tipko. Z neposredno povezavo tipke na vhod krmilnika brez uporabe notranjega ali zunanjega pullup upora vezje ne bi delovalo.

Srce programa se skriva v podprogramu void loop(), kjer z if (digitalRead(pinC) == LOW) {} najprej preverimo, ali je stanje na pinu C enako LOW. Ob uporabi notranjega pullup upora to pomeni, da je tipka pritisnjena. V tem primeru z ukazom tone(pinZvocnik, notaC); prek pina pinZvocnik na zvočniku predvajamo ton s frekvenco note C. Če tipka za noto C ni bila pritisnjena, potem z ukazom else if (digitalRead(pinD) == LOW) {} preverimo, ali je bila morda pritisnjena tipka za noto D. Če tudi ta ni bila pritisnjena, preverimo še noto E. Če ni bila pritisnjena nobena od tipk, se zaradi ukaza else {} izvede koda noTone(pinZvocnik);, ki izklopi predvajanje tonov. Ukazi se nato ciklično ponavljajo v okviru void loop() {}.

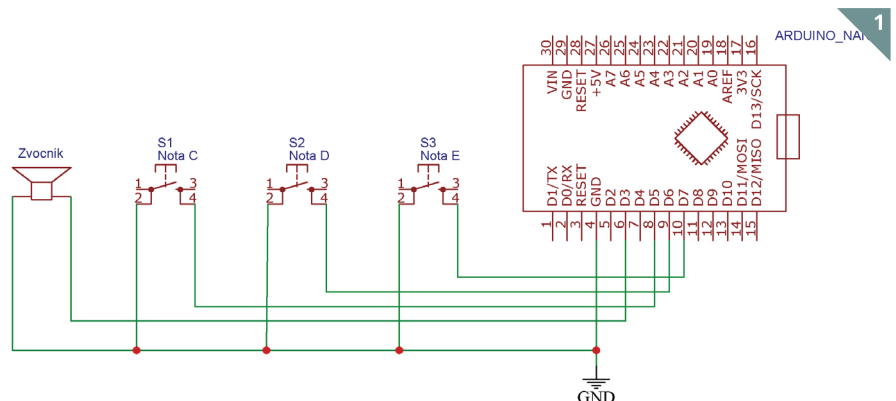
Po prenosu programa na krmilnik in preizkusu tipk lahko zaigramo znano skladnico Kuža pazi tako, da pritisnemo ustrezne tipke v naslednjem zaporedju: C, C, C, C, D, D, D, D, E, E, D, D, C, C, C.

Prenosljivost

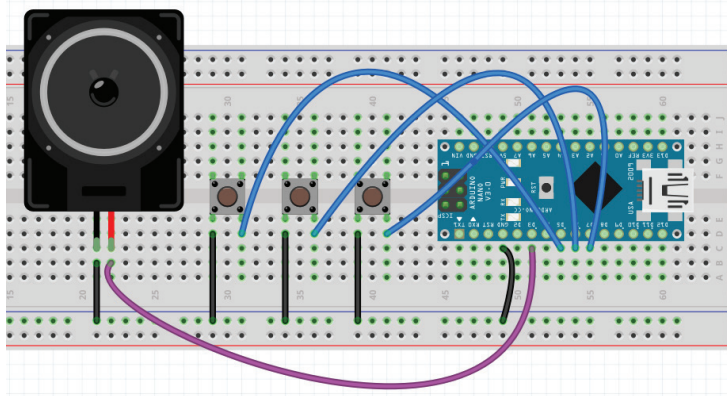
Uporaba naprave je precej nepriložna, saj jo moramo imeti prek USB-kabla ves čas priključeno na računalnik. Naj izdamo skrivnost: po prenosu programa na krmilnik Arduino računalnika v resnici ne potrebujemo več, zagotoviti moramo le ustrezno napajanje.

Možnosti je več. Najlažja je kar uporaba ustreznih napajalnikov USB, kakršne imajo na primer mobilni telefoni, uporabimo lahko namenske USB-napajalnike in celo USB-konektorje sodobnih naprav (nekateri televizije, zasloni ...) za napajanje zunanjih enot. V vseh primerih žal ostaja težava povezanosti prek USB-kabla oziroma povezave na omrežno napetost 230 V.

Glede na dokumentacijo za Arduino Nano je ustrezna napajalna napetost Vin med 7 in 12 V (več o tem na spletni strani proizvajalca <https://store.arduino.cc/>



```
//
// Program Arduino - prenosljivi klavir
//
```



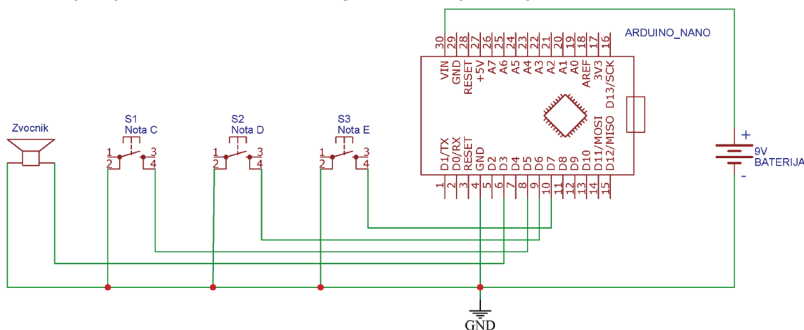
2

ELEKTRONIKA
REGISTRACIJA
PREČKANJA CILJNE ČRTE
(3. del)

▼ Jernej Böhm

Načrt B

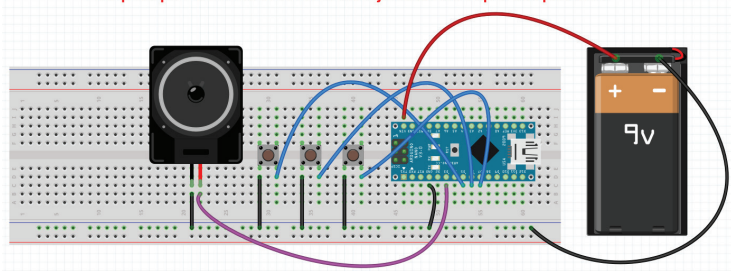
Pred priklopom krmilnika Arduino na baterije obvezno izklopite USB povezo!



3

V nadaljevanju projekta o registraciji prevoza cilja na tekmovanjih modelov električnih avtomobilov opisujem programsko opremo, ki povezuje strojno opremo (prenosni računalnik, modelarski semafor, ciljna mikrostikala ali laserski merilnik razdalje, zdaj kot možna varianta) v zaključeno celoto. Hkrati obdeluje pridobljene merilne podatke, prikaže doseženi rezultat na modelarskem semaforju (TIM LVII-8, 9) ter v zaključku tudi uredi rezultate in sestavi vrstni red tekmovalcev. Celoten podatkovni prikaz skupaj z rezultati je viden tudi na računalniškem zaslonu.

Pred priklopom krmilnika Arduino na baterije obvezno izklopite USB povezo!



4

Program je napisan v Microsoftovem Visual Basicu (VB6). Marsikdo bo pripomnil, da je nekoliko zastarel, saj ga ameriško podjetje že nekaj časa ne podpira več. Naj kar takoj pojasnim, da gre morda za najuspešnejši odprtokodni izdelek, izjema je le OS Linux. Nekje sem prebral, da VB6 še vedno uporablja vsaj 16 % računalniških programerjev (leto star podatek), kar je v primerjavi z danes najbolj priljubljeno Javo (58 %) kar precejšen odstotek uporabnikov. Obstaja celo mednarodna skupnost računalniških programerjev, ki VB6 vzdržuje še naprej (<http://www.vbforums.com/showthread.php?869641-Vb6-Community>).

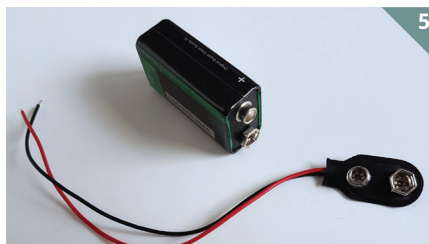
arduino-nano). Tako lahko problem prenosljivosti rešimo kar z napajanjem prek baterije. Vse, kar je treba narediti, je, da povežemo pozitivni pol baterije na napajalni vhod Vin, negativni pol pa na GND-krmilnika Arduino, kot je prikazano na sliki vezja (slika 3) oziroma na različici vezja v programu Fritzing (slika 4). Uporabili smo kar standardno 9-V baterijo, priporočljivo pa si je zagotoviti tudi namenski konektor (slika 5), ki ga lahko kupimo v trgovinah z elektronskimi komponentami.

Za napajanje lahko uporabimo tudi zaporedno vezane standardne 1,5-V baterije, na primer tipa AA ali AAA, pri čemer mora skupna napetost vezanih baterij presegati 7 V. Pogoju zadostimo, če zaporedno zvežemo pet baterij, saj je $5 \times 1,5 \text{ V} = 7,5 \text{ V}$. Uporabimo lahko tudi baterije, ki jih lahko polnimo. Ker pa ima posamezen člen teh baterij napetost 1,2 V, potrebujemo vsaj šest takih baterij, saj je $6 \times 1,2 \text{ V} = 7,2 \text{ V}$. Avtomobilski akumulator ima 12 V, kar je hkrati tudi najvišja napetost, ki jo smemo uporabiti. Seveda je za našo napravo tako napajanje zaradi velikosti neprimerno, je pa res, da bi jo lahko zelo dolgo uporabljali. Pred priklopom napajanja krmilnika Arduino prek baterij moramo zaradi zaščite računalnika obvezno izklopiti USB-kabel.

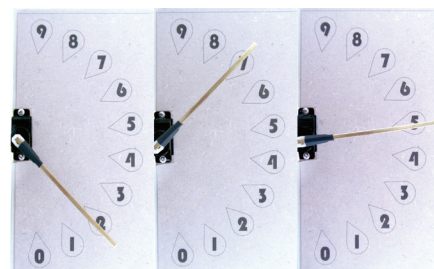
Zaključek

Uporaba baterije ali drugega vira napajanja prinaša krmilniku Arduino prenosljivost in neodvisnost od računalnika, kar zelo razširi nabor možnih uporab in projektov. Ob tem moramo v primeru klaviirja razmisliti tudi o ustreznem ohišju in razporeditvi zunanjih tipk, vendar to za zdaj presega okvir naše serije prispevkov. Lahko pa za zabavo na napravi povečamo število tipk in s prilagoditvijo programa dodamo še kakšno novo noto. Za tiste najbolj navdušene naj povemo, da lahko tudi del analognih vhodov uporabimo kot digitalne vhode, s čimer lahko naredimo že kar spodobno klaviaturo. Pri tem pa ne pozabimo uporabiti notranjega pullup ali zunanje upor (več o tem na spletnih straneh proizvajalca: <https://www.arduino.cc/en/reference/board> in <https://www.arduino.cc/en/Tutorial/AnalogInputPins>).

5



Laserski merilnik razdalje je še vedno na poti iz Kitajske, bolj verjetno pa čaka na obnovev proizvodnje v tamkajšnji tovarni, saj so mi trgovci že večkrat vljudno podaljšali rok dobave. Zaradi tega sem bil prisiljen spremeniti načrt izvedbe registracije prevoza ciljne črte. Ideja je preprosta. Vsaka točkovna vrata, ki jih je na ciljni črti 22, omejimo s parom mikrostikal s podaljšanim vzvodom. Ta so pomnožena izvedba maksimuma, mikrostikala za 100 točk, opisanega v prejšnjem delu projekta, in so nekakšna 23. vrata. Tista naj-



Modelarski semafor krmilijo trije običajni RV-servomehanizmi.