

PRVI KORAKI V ARDUINO – NASTAVLJIVA FREKVENCA ZVOKA IN MELODIJA PESMI

1

▼ Milan Gaberšek in Slavko Kocijančič

V sedmem prispevku iz niza prvih korakov v Arduino bomo krmilnik uporabili za predvajanje različnih tonov glede na njihovo frekvenco in trajanje. Za oddajanje zvoka potrebujemo le mini piezozvočnik. Ker želimo vplivati na frekvenco zvoka, bomo v ta namen vezali še potenciometer. Seveda moramo na krmilnik naložiti tudi ustrezen program.

Material

- krmilnik Arduino Nano ali podoben,
- USB-kabel (mini USB) za povezavo krmilnika z računalnikom,
- prototipna ploščica (angl. breadboard),
- potenciometer ali trimerni potenciometer 5 k Ω ,
- mini piezozvočnik,
- vezne žičke (rdeča, vijoličasta, modra in tri črne).

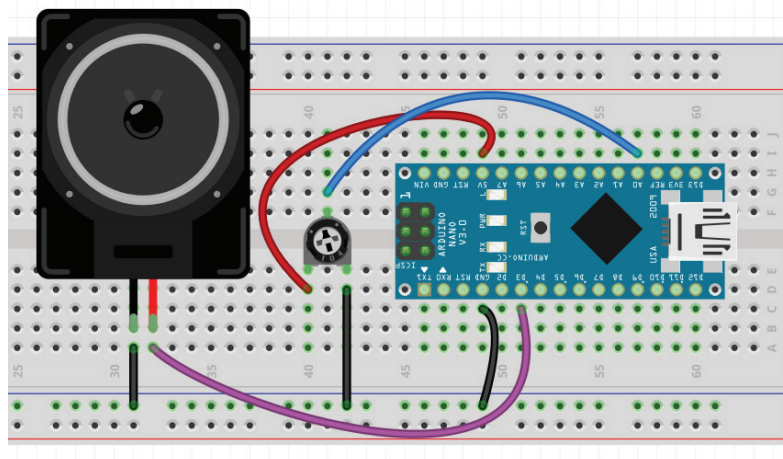
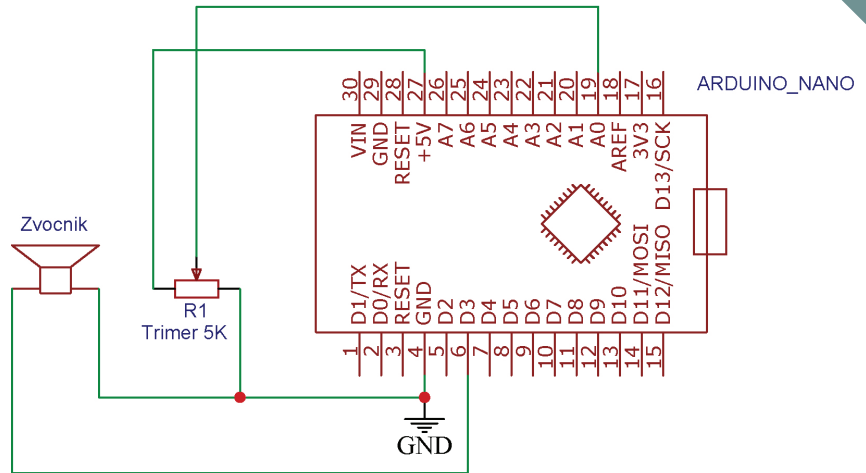
Orodja in pripomočki

- 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

Glede na električno shemo (slika 1) oziroma sliko, narejeno s pomočjo programa Fritzing (slika 2), povežemo posamezne elemente. Pri tem uporabimo mini piezozvočnik (slika 3), ki za delovanje ne potrebuje velike moči in ga zato lahko priklopimo neposredno na krmilnik Arduino. Pri nekaterih izvedbah so žičke zelo tanke in se hitro odtrgajo. Najboljša rešitev je, da prispajkamo nove vezne žičke in vse skupaj okrepimo s termoskrčljivo cevko. Če te možnosti nimamo, si lahko pomagamo tako, da najprej v prototipno ploščico vstavimo tanko žičko piezozvočnika, nato pa v isto luknjico še običajno vezno žičko, ki je debela ravno toliko, da jo lahko vstavimo brez težav (slika 4).

Uporabili bomo ročno nastavljivi delilnik napetosti, za kar uporabimo potenciometer (slika 5) oziroma trimerni potenciometer (slika 6). V načinu delovanja sta enaka, le da pri potenciometru delilnik napetosti nastavljam z neposrednim ročnim vrtenjem osi (vrtljiva palčka), pri trimernem potenciometru pa napetost



2

nastavimo z izvijačem, kar pride prav, če je malo prostora ali ob redkih nastavitvah vrednosti. Oba imata tri priključke in se vedeta kot delilnik napetosti z dvema uporoma, kar lahko prikazemo z nadomestno shemo (slika 7).

V tem prispevku bo nekaj več dela s programom. Najprej bomo napisali program, ki bo ustvaril ton določene višine oziroma določene frekvence. Nato bomo napisali program, ki bo predvajal znano melodijo napetosti s potenciometrom bo potekalo v ločenem programu, pozneje pa bomo vse znanje združili in naredili preprosto napravo z nastavljivo frekvenco zvoka.

Izhajali bomo iz že znanega. Zvočnik bomo krmilili prek digitalnega izhoda D3 (program Zvok 1). Z ukazom `digitalWrite(pinZvocnik, HIGH);` bomo digitalni izhod D3 postavili v stanje logične 1 (+5 V), posledično se bo membrana mini piezozvočnika napela, zrak se bo zgostil (večji zračni tlak) in zgoščina bo začela potovati po prostoru. Prej ali slej bo prišla do našega ušesa, v katerem se bo usločil bobnič. Ob `digitalWrite(pinZvocnik, LOW);` se bo zaradi stanja logične 0 (napetosti okrog 0 V) membrana mini piezozvočnika vrnila v prvotni položaj. Nastala bo razredčina (nižji tlak), ki jo bo zaznal bobnič v našem ušesu. Naši možgani bodo pri ponavljanju zgoščin in razredčin to interpretirali kot zvok. Zanka `loop` in dva ukaza `delay(trajZvocnik);` določata periodo ponavljanja vsaki 2 milisekundi, kar bomo zaznali kot

zvok s frekvenco 500 Hz – po znanem izračunu za frekvenco $f = 1/\text{perioda} = 1/(2 \text{ ms}) = 1/(0,002 \text{ s}) = 500 \text{ 1/s} = 500 \text{ Hz}$.

Tako kot za vse kode programov iz prejšnjih številčk Tima tudi za programe iz tega članka velja, da jih lahko pridobimo na spletni strani www.drtn.si/tim.html.

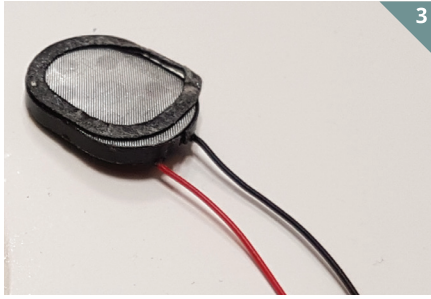
```
//
// Program Arduino - Zvok 1.
//

// Zvočnik bomo priklopili na D3
const int pinZvocnik = 3;
// Začetno trajenje lege membrane
// bo 1 ms (tj. 1 milisekunda)
const int trajZvocnik = 1;

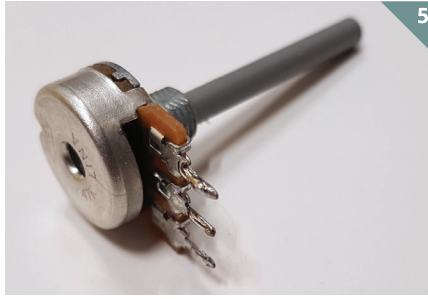
void setup() {
  // Zvočnik bo priklopljen na
  // digitalni izhod
  pinMode(pinZvocnik, OUTPUT);
}

void loop() {
  digitalWrite(pinZvocnik, HIGH);
  delay(trajZvocnik);
  digitalWrite(pinZvocnik, LOW);
  delay(trajZvocnik);
}
```

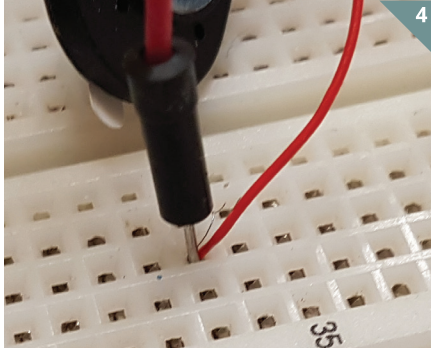
Ko uspešno preizkusimo vezje in program, spremenimo vrednost 1 v vrstici `const int trajZvocnik = 1;` na primer `const`



3



5



4



6

int trajZvocnik = 5; Kolikšna je v tem primeru frekvenca zvoka?

Knjižnica ukazov programskega okolja Arduino IDE ponuja preprostejšo rešitev za predvajanje tona določene frekvence z ukazom tone. Pri tem kot parameter navedemo pin, na katerega bomo priključili mini piezozvočnik, frekvenco v hercih (Hz) in čas trajanja v milisekundah. Najnižja frekvenca, ki jo lahko predvajamo, je 31 Hz. Z ukazom tone lahko naenkrat uporabimo le en izhod vmesnika Arduino in posledično en sam zvočnik. Kot primer bomo napisali program, ki bo predvajal preprosto melodijo pesmi Kuža pazi. Tokrat smo vse ukaze strnili v podprogram setup(), saj želimo, da se program izvede le enkrat. Zanka loop() ostane prazna, vendar jo moramo vseeno navesti, saj drugače program ne deluje.

```
tone(pinZvocnik, notaD, trajanje); delay(pavza);
tone(pinZvocnik, notaE, trajanje); delay(pavza);
tone(pinZvocnik, notaE, trajanje); delay(pavza);
tone(pinZvocnik, notaD, trajanje); delay(pavza);
tone(pinZvocnik, notaD, trajanje); delay(pavza);
tone(pinZvocnik, notaC, trajanje); delay(pavza);
tone(pinZvocnik, notaC, trajanje); delay(pavza);
tone(pinZvocnik, notaC, trajanje); delay(pavza);
```

```
void loop() {
}
```

krajše s hkratnim pritiskom tipk Ctrl, Shift in M. Z vrtenjem osi potenciometra oziroma sukanjem trimerja z izvijačem vidimo, da se vrednosti spreminjajo nekje med 0 in 1023. To vrednost pozneje uporabimo za spreminjanje frekvenca zvoka.

```
//
// Program Arduino - potenciometer
//

const int pinPotenciometer = A0;
// Vrednost iz potenciometra
// kot delilnika napetosti bomo
// shranjevali v spremenljivko
int potVrednost = 0;

void setup() {
// Branje podatkov preko
// serijskega vmesnika USB
Serial.begin(9600);
}

void loop() {
// Z ukazom analogRead preberemo vrednost
// in jo shranimo v spremenljivko
potVrednost = analogRead(pinPotenciometer);
// Dobljeni podatek izpišemo na zaslon
Serial.println(potVrednost);
}
```

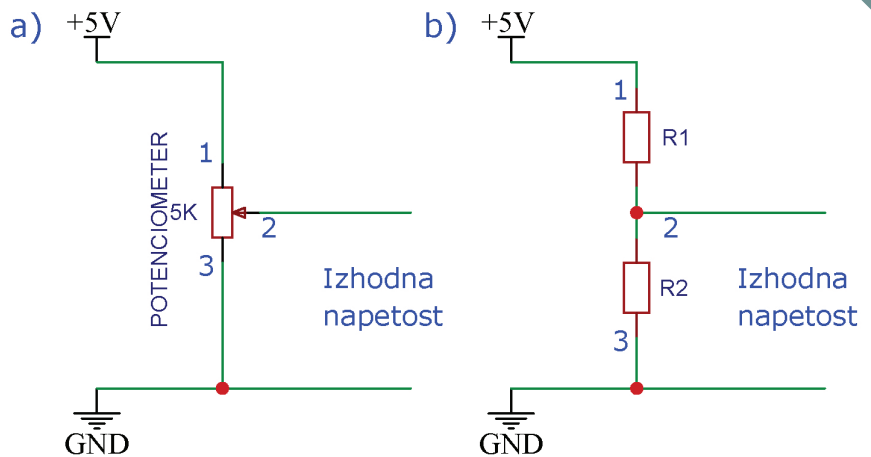
Usvojeno znanje združimo v nov program, s katerim bomo frekvenco tona spreminjali s potenciometrom kot delilnikom napetosti. Vrednosti v programu ne bomo izpisovali na zaslon, zato bo programska koda preprostejša. V programu najprej navedemo spremenljivki, ki se sklicujeta na posamezne pine krmilnika Arduino, to sta pinPotenciometer in pinZvocnik. Napovemo tudi spremenljivko vrednostHz, v katero bomo prebrali vrednost iz potenciometra v vlogi ročno nastavljivega delilnika napetosti. S stavkom vrednostHz = analogRead(pinPotenciometer) * 10; v okviru podprograma loop() vrednost preberemo iz analognega vhoda in jo pomnožimo z 10. Tako dobimo vrednosti frekvenc med 0 in 10230, pri čemer frekvenc do okoli 16 Hz ne slišimo, saj gre za infrazvok. Z ukazom tone(pinZvocnik, vrednostHz); ton dejansko predvajamo,

```
//
// Program Arduino - Kuža pazi
//

// Zvočnik bomo priključili na D3
const int pinZvocnik = 3;
// Frekvence za posamezne note, več not na
// na https://www.arduino.cc/en/Tutorial/tone-Melody
const int notaC = 262; // tudi C4
const int notaD = 294; // tudi D4
const int notaE = 330; // tudi E4
const int trajanje = 300;
int pavza = trajanje * 1.30;

// Ker bi radi le eno izvedbo,
// je že v setup() celotni program
void setup() {
// Uporabimo ukaz tone za predvajanje
// tona frekvence določene note ter počakamo
tone(pinZvocnik, notaC, trajanje); delay(pavza);
tone(pinZvocnik, notaC, trajanje); delay(pavza);
tone(pinZvocnik, notaC, trajanje); delay(pavza);
tone(pinZvocnik, notaD, trajanje); delay(pavza);
tone(pinZvocnik, notaD, trajanje); delay(pavza);
tone(pinZvocnik, notaD, trajanje); delay(pavza);
}
```

Sledi program za testiranje delilnika napetosti s potenciometrom oziroma trimernim potenciometrom. Program je zelo podoben tistemu iz prispevka Prvi koraki v Arduino – stik s svetom in računalnikom, objavljenega v januarški številki TIM leta 2020. Tudi tu vrednost napetosti merimo prek analognega vhoda A0 in jih nato prek USB-vodila pošiljamo v računalnik. Vrednosti si lahko ogledamo v razvojnem okolju Arduino IDE, kjer prikaz vklopimo v meniju Orodja, Serijski vmesnik, oziroma



7

pri čemer velja opozoriti, da mini piezo-zvočnik večino tonov predvaja popačno, občasno celo nekoliko zahrešči.

```
//
// Program Arduino - potenciometer in zvok
//

const int pinPotenciometer = A0;
int vrednostHz = 0;
const int pinZvocnik = 3;

void setup() {
  // tokrat ni potrebno nič vnaprej
  // nastavljeni
}

void loop() {
  // Z ukazom analogRead preberemo vrednost
  // iz potenciometra kot delilnika napetosti,
  // jo shranimo v spremenljivko in nato
  // z njo nastavimo frekvenco tona na zvočniku
  vrednostHz = analogRead(pinPotenciometer)
  * 10;
  tone(pinZvocnik, vrednostHz);
}
```

Zaključek

Potenciometer kot delilnik napetosti in zvočnik odpirata možnosti za kopico različnih projektov. V povezavi s temami iz prejšnjih člankov lahko naredimo tudi projekt klaviature s tipkami. Vsaka tipka naj ima svojo tonsko frekvenco na glasbeni lestvici, čas trajanja tona pa naj bo odvisen od trajanja pritiska na posamezno tipko. Začnemo lahko, recimo, s štirimi tipkami.

ZVEZA ZA TEHNIČNO KULTURO SLOVENIJE

100 in 1 MAKETA
PETER LOGORELEC
50 LET MODELARSTVA

29,80 EUR

Knjiga **Sto in ena maketa**, katere sozaložnik je ZOTKS, bo dragocen pripomoček za vse tiste, ki se podajajo na pota tehničnega ustvarjanja in natančnega upodabljanja objektov v pomanjšanem merilu, mladim pa izziv za udejstvovanje na področjih, ki spodbujajo razvijanje ročnih spretnosti. Ob tem ne smemo spregledati dejstva, da gre tudi za dokument posebnega pomena za ohranjanje slovenske tehnične kulturne dediščine.

Naročila sprejemamo na:
info@zotks.si
(01) 25 13 743

Zveza za tehnično kulturo Slovenije
Zaloška 65, p. p. 2803
1000 Ljubljana

REGISTRACIJA PREČKANJA CILJNE ČRTE (2. del)



Jernej Böhm

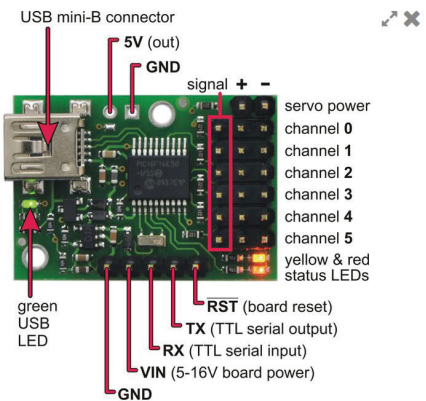
Za začetek naj na kratko komentiram obstoječi pravilnik ZOTKS za tekmovanje na področju D1 (Tekmovanje modelov električnih avtomobilov in cilj – prosta vožnja v cilj).

Pravilo simetrale: »Če model s simetralo prepelje črto (široka je 20 mm, op. p.), ki določa sredino ciljne črte, dobi tekmovalec 100 točk.«

To da slutiti možnost subjektivne sodniške odločitve pri razumevanju tega, kaj je simetrala, na katero pa pritožba ni mogoča. Tudi v številnih drugih športnih panogah odloča sodnik, marsikatero sporno odločitev zato spremlja množica različnih mnenj. So pa tudi panoge, ki so imune prot spornim dosežkom, na primer skok v višino. Priča smo primerom, ko tekmovalec vidno zaniha višinsko letvico, a ta obstane v svojem ležišču in skok je veljaven. Še lepši primer se ponuja pri kegljanju: velja podrt kegelj ne glede na to, ali je vanj kroglja trčila frontalno (simetralno) ali ga je le oplazila. Pustimo ob strani, da se keglji v glavnem podirajo med seboj.

Zgornje pravilo simetrale omenjam zato, ker ga je težko udejanjiti s preprostimi senzori, medtem ko je zaznavanje podrttega keglja preprosto, recimo z mikrostikalom. V našem primeru ga namestimo nad sredinsko črto (stotico), na njegovo ročico pa pritrdimo primeren vzvod, ki ga vozilce premakne, ko nanj naleti.

Zakaj izpostavljam problem maksimuma pri prosti vožnji v cilj, ko pa položaj modela tako ali tako izmerimo s pomočjo laserskega žarka in to celo z milimetrsko natančnostjo? Žarek je neviden, zato moramo zaupati v točnost in poštenost meritve, kar pa lahko povzroči določene pomisleke predvsem pri tekmovalcih, ki se v končnici merijo za zmagovalne stopničke. Podrt kegelj pa vidijo vsi. Rezultat, določen z vidnim mikrostikalom oziroma njegovo podrto ročico, predjudicira laserski rezultat. Predpostavljam, da v boju za najboljši rezultat sodeluje več izenačenih tekmovalcev in tedaj je pomembno, da je dosežek nekako »viden«. Ko se bo elektronska laserska meritev uveljavila, podobno kot pri smučarskih disciplinah, se bo senzorično nad stotico zagotovo opustilo, kar bo hkrati poenostavilo merilno službo. Za to pa so potrebne izkušnje, ki jih prinese čas. Poglejmo, kako izdelati ta senzorski dodatek nad stotico.



Mikrokrmilnik micro maestro



Priključki mikrokrmilnika micro maestro

Šestkanalni mikrokrmilnik servomehanizmov micro maestro

V lanski 8. in 9. številki revije TIM sem opisal izdelavo večjega trimestnega semaforja za prikaz rezultatov oziroma podatkov, ki se tičejo modelarskega tekmovanja. Pri tem sem uporabil tri običajne modelarske servomehanizme. Semafor upravljamo prek prenosnega računalnika, in to prav tistega, s katerim je povezano tudi laserski merilnik razdalje. Za povezavo je potrebna elektronika, ki sem jo letos prilagodil izvedbi laserskega merilnika razdalje (v prejšnji številki). Ista elektronika naj bi nadzirala tudi omenjeno mikrostikalno. Za krmiljenje semaforja sem zato izbral cenovno ugodnejši šestkanalni mikrokrmilnik servomehanizmov micro maestro podjetja Pololu (www.pololu.com). V Sloveniji ga zastopa, kar sem že omenil, kamniško podjetje Tech, d. o. o. (www.prohobi.net). Pololujevo krmiljenje servomodulčka je preprosto, morda celo bolj kot krmiljenje z Arduinovim modulom, ki ga bomo prav tako uporabili.

Tudi omenjenemu mikrokrmilniku (Pololu) lahko poljubno preprogramiramo priključke, da so prirejeni za krmiljenje RV-servomodulčka je preprosto, morda celo bolj kot krmiljenje z Arduinovim modulom, ki ga bomo prav tako uporabili.