

Prvi koraki v Arduino

Vožnja po črti in izogibanje oviri

Milan Gaberšek in Slavko Kocijančič

V trinajstem zaporednem prispevku na temo *Prvi koraki v Arduino* bomo obstoječi avtomobilček s krmilnikom Arduino nadgradili tako, da se bo med vožnjo po črti izogibal tudi oviram na poti. Njegovo konstrukcijo bi sicer lahko pustili nespremenjeno, a da ne bomo samo programirali, bomo z majhnim dodatkom dosegli lažje gibanje.

Material in orodje

- izdelani in nadgrajeni avtomobilček (TIM 2, oktober 2020, str. 27–31),
- frnikola,
- modelarski nož,
- pištola za vroče lepljenje,
- bela podlaga z dolgo, ravno črto ali primerna proga,
- težja škatlica ali kladica za oviro na progi.

Računalniški 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.

Izboljšava avtomobilčka za vožnjo po črti

V prispevku v letošnji oktobrski številki Tima smo obstoječi avtomobilček predelali tako, da je ob ustreznem programu sledil zaključeni progi v obliki široke črne črte. Za njegovo konstrukcijo smo uporabili plastično posodico, na katero smo pritrdili dve veliki kolesi, zadaj pa je ohišje drselo prosto po tleh, zato je tam prihajalo do razmeroma velikega trenja. Tega bomo zdaj zmanjšali z uporabo frnikole (**slika 1**), ki jo bomo namestili na sredino zadnjega dela avtomobilčkovega ohišja. Na tistem delu ohišja, kjer bo frnikola, z modelarskim nožem naredimo nekaj manjših zarež, da se bo vroče lepilo bolje oprijelo sicer gladke površine plastične posodice. S pištolo za vroče lepljenje naneseemo debelo plast lepila, nekoliko počakamo, nato pa v raztaljeno lepilo potisnemo frnikolo (**slika 2**). Po potrebi jo s strani še dodatno zalijemo z vročim lepilom in pri tem pazimo, da ga ne naneseemo na njen zgornji del, ki bo pozneje v stiku s podlago. Ker je frnikola iz gladkega stekla, obenem pa je stična površina med podlago in zadnjim delom ohišja zdaj veliko manjša, bo avtomobilček z zadnjim delom precej lažje drsel po površini; poleg tega bo tudi gibljivejši, kar nam bo pri obračanju med izogibanjem oviri prišlo zelo prav.

```
//
// Program Arduino -
// Vožnja po črti in izogibanje oviri

// Pini za tipki za zaznavanje ovire
const int pinTipkaLeva = 2;
const int pinTipkaDesna = 4;

//
// POZOR! V tem delu manjkajo vse napovedi spremenljivk, določitve vhodnih in izhodnih pinov krmilnika Arduino ter funkcije za krmiljenje motorčkov. Celoten program je dosegljiv na: www.drta.si/tim.html
//

// Spremenljivka za vklop(true)
// oziroma izklop(false) sledenja po črti
int voznjaPoCrti = true;
void loop() {
  // Z ukazom analogRead preberemo
  // digitalno jakost odbite svetlobe
  jakostSvetlobe = analogRead(senzorSvetlobe);
  // Dobljeni podatek izpišemo na zaslon
  Serial.println(jakostSvetlobe);
  delay(10); // Počakamo na podatke
  // Preverimo, ali je pritisnjena tipka tipalk
  if ( (digitalRead(pinTipkaLeva) == LOW)
    || (digitalRead(pinTipkaDesna) == LOW) ) {
    // Koda za izogibanje oviri
    peljiNazaj(); delay(100);
    vrtiVLevo(); delay(400);
    peljiNaprej(); delay(500);
    vrtiVDesno(); delay(280);
    peljiNaprej(); delay(820);
    vrtiVDesno(); delay(350);
    peljiNaprej(); delay(400);
    vrtiVLevo(); delay(350);
    // Za lažje nastavljanje časov
    // avtomobilček še ustavimo
    ustavi();
  } else {
    // Vklop kode za vožnjo po črti
    if (voznjaPoCrti == true) {
      // Iskanje črte:
      // Če je jakost odbite svetlobe
      // manjša od referenčne, se zavrtimo
      // v levo, drugače v desno
      if (jakostSvetlobe <= refSvetlobe) {
        vrtiVLevo(); delay(20);
      } else {
        vrtiVDesno(); delay(20);
      }
      peljiNaprej(); delay(10);
      ustavi(); // Ustavi
    }
  }
}
```



Avtomobilček in izogibanje oviri

Zdaj lahko pripravimo vse potrebno za tokratni projekt. Namesto kladice smo v našem primeru uporabili kar staro škatlico mobilnika, ki je zaradi kompaktnosti dovolj težka, da se ob naletu avtomobilčka ne premakne (**slika 3**). Najbolje je, da je ovira ozka in dovolj visoka. Da je avtomobilček ob naletu ne bi premaknil, jo po potrebi lahko tudi napolnimo, na primer z rižem ali vijaki. Škatlico oziroma kladico položimo na obstoječo proggo za vožnjo po črti, še bolje pa je, če izdelamo takšno novo proggo, ki jo sestavlja samo dolga ravna črta, saj bo prilagajanje programa s tem precej lažje (**slika 4**). Avtomobilček postavimo na črto na proggi (**slika 5**).

Program za vožnjo po črti in izogibanje oviri pomeni nadgradnjo programa za vožnjo po črti iz prejšnjega nadaljevanja. Tokrat si bomo v prispevku ogledali le del kode znotraj zanke `loop()`, cel program pa je prosto dostopen na spletni strani www.drtni.si/tim.html. Tam najdete še videoposnetek vožnje avtomobilčka po črti z izogibanjem oviri, prav tako pa tudi datoteke za tiskanje proge v obliki ravne črte v formatu A 4 oziroma A 3.

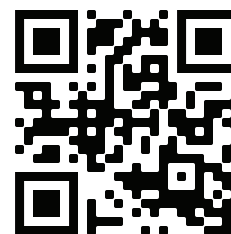
Da bi bil program preglednejši, smo ga močno »oskubili«, saj smo izpustili večino napovedi spremenljivk, določitev pinov krmilnika Arduino v okviru funkcije `setup()` in funkcije za krmiljenje motorjev. Vse našto namreč že dobro poznamo.

V zanki `loop()` – enako kot pri vožnji po črti – najprej preberemo vrednost jakosti svetlobe in jo prek serijske komunikacije pošljemo v računalnik, kar je dobrodošlo na stopnji preskušanja in prilagajanja programa. Ne pozabite nastaviti pravilne vrednosti obeh spremenljivk `jakostCrna` za črno oziroma `jakostBela` za belo podlago, kakor je bilo opisano v prejšnjem prispevku, sicer avtomobilček ne bo sledil črti. Nato pride v programu na vrsto del kode, ki skrbi za avtomobilčkovo izogibanje oviram. Z ukazom `if ((digitalRead(pinTipkaLeva) == LOW) || (digitalRead(pinTipkaDesna) == LOW)) { }` namreč preverimo, ali je katera koli od tipk sprednjih tipalk pritisnjena, kar pomeni, da se je avtomobilček zaletel v oviro. To smo storili tako, da smo med ukazoma `digitalRead` za levo in desno tipko uporabili logični OR, ki ga v programskem jeziku C nadomeščata pokončni črti `||`. Črtico `|` dobimo s hkratnim pritiskom tipk `AltGr` in `W`. Z uporabo te funkcije tako ne preverimo le, ali je pritisnjena posamezna tipka, temveč tudi, ali sta morda pritisnjeni obe hkrati.

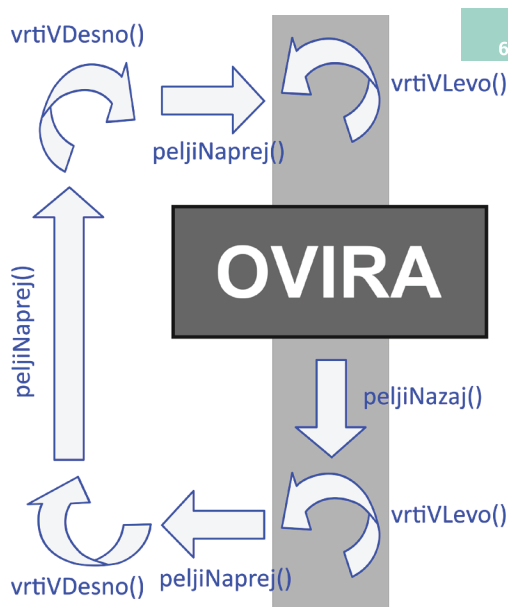
Če je pogoj izpolnjen, se izvede koda znotraj zavitih oklepajev, ki je jedro procesa izogibanja oviri. Da si bomo lažje predstavljali njeno delovanje, si zamislimo, da se po črti bližamo omari. Ko se je dotaknemo, se najprej pomaknemo nekoliko nazaj, nato se obrnemo v levo in naredimo nekaj korakov, da pridemo mimo omare. Nato se obrnemo na desno, tj. v prvotno smer, in spet naredimo nekaj korakov naprej, da pridemo mimo omare. Zdaj se lahko vrnemo na prvotno črto, ki poteka tudi naprej od omare. To storimo tako, da se obrnemo na desno in naredimo nekaj korakov, da pridemo do črte. Popolnoma enako naredi tudi omenjeni del kode, pri čemer se zaporedje ukazov zvrsti takole: `peljiNazaj()`, `vrtiVLevo()`, `peljiNaprej()`, `vrtiVDesno()`, `peljiNaprej()` in `vrtiVDesno()`, `peljiNaprej()`, `vrtiVLevo()`.

Da bi si še lažje predstavljali delovanje posameznih ukazov, si oglejmo diagram na **sliki 6**. Odvisno od avtomobilčka in velikosti ovire z ukazi `delay()` nastavimo čas delovanja posameznega ukaza. Avtomobilček na koncu še ustavimo z ukazom `ustavi()`, ki pride prav pri samem preskušanju in nastavljanju časov, pozneje pa nima posebnega pomena.

Pred nastavljanjem časa del kode `int voznjaPoCrti = true;`, ki je tik pred `void loop()`, nastavimo na `false`, s čimer posredno izklopimo del kode za sledenje po črti, ki bi nas pri nastavitvi časov samo motil. Nastavljanje časa delovanja ukazov najlažje izvedemo tako, da preverjamo in nastavljamo le posamezni ukaz, druge pa sproti spreminjamo v komentar z dodajanjem `//` pred ukaz, s čimer preprečimo njihovo izvedbo. Če želimo oznako `//` za komentar dodati več vrsticam hkrati, jih najprej označimo z miško, nato pa v meniju Arduino IDE izberemo Uredi, Komentiraj/odkomentiraj. Najbolje je, če damo v komentar vse omenjene ukaze, razen prvega, tj. `peljiNazaj()`, in njegove nastavitve časa `delay`.



Avtomobilček po opravljenem prenosu programa postavimo tik pred oviro, pritisnemo na eno od stikal njegovih sprednjih tipalk in opazujemo ustreznost nastavljenega časa pri ukazu `peljiNazaj()`. Ko ga s poskušanjem določimo, je smiselno izvesti nekaj



ponovitev, saj zaradi podlage kakšno od koles včasih ne prime enako, trenje s podlago je lahko različno ipd. Le tako bomo prepričani, da je čas res ustrezen. Nato si zapomnimo lego avtomobilčka, »zakomentiramo« `peljiNazaj()` in se lotimo naslednjega ukaza `vrtiVLevo()`. Ko je tudi ta nastavljen, preverimo delovanje obeh in pri tem seveda odstranimo komentar pri `peljiNazaj()`. Postopek ponavljamo za vsakega izmed navedenih ukazov, dokler se avtomobilček z mesta pred oviro po pritisku na stikalo tipalke z zaporedjem navedenih ukazov samostojno ne odpelje na črno črto za oviro, pri čemer mora biti obrnjen v smeri črte. Ko izogibanje oviri naposled deluje pravilno, lahko spremenljivko `voznjaPoCrti` postavimo nazaj na `true`, s čimer vključimo tudi delovanje že znane kode za sledenje po črti.

Ko je `voznjaPoCrti` enaka `true`, s čimer bo del kode za sledenje črti vključen, se bo ta izvedel v primeru neskljenjenih tipalk. Koda je znotraj oklepajev ukaza `else`. Vidimo lahko, da je tu ukaz `if (voznjaPoCrti == true) {}`, ki dejansko izvaja oziroma ne izvaja dela kode za vožnjo po črti. Ta je nespremenjena in – kot že vemo – deluje tako, da s stavkom `if` preverimo, ali je referenčna vrednost jakosti svetlobe presežena

ali ne. Če je manjša, kar pomeni, da smo na črni črti, vozimo v levo; če je večja, kar pomeni, da smo na beli podlagi, pa v desno. V kombinaciji s pomikom naprej in več izvedbami zanke `loop` se tako avtomobilček nekako »odbija« od roba črte. Dodali smo še funkcijo `ustavi()`, da ima krmilnik Arduino pred novo zanko `loop()` in morebitnim pomikom čas obdelati podatke. Ko se bo avtomobilček pri pomikanju naprej zaletel v oviro, se bo izvedel del kode za umik oviri.

Sklep

S tem končujemo prispevke na temo avtomobilček robot. Z njim zdaj znamo početi že marsikaj: lahko ga krmilimo s pomočjo tipk ali ga pripravimo k plesu, lahko se vozi in odbija od ovir, lahko išče črto in vozi po njej, v zadnjem nadaljevanju pa smo ga naučili še izogibati se oviram. Kodo za ta manever bi seveda lahko izvedli tudi drugače. Z ustreznim dodatnim ukazom in natančno nastavitvijo bi bilo mogoče še izboljšati obhod okrog ovire.

Na dosedanjih tekmovanjih iz robotike so bile predstavljene različne rešitve, med katerimi je bila najizvirnejša tista, pri kateri se je avtomobilček z zaporednim zaletavanjem in preverjanjem stanja desne tipalke s postopnim pomikanjem proti levi gibal okrog ovire, dokler ni spet naletel na črto. Tako je deloval kot ribica, ki ob pomolu nabira hrano. Prednost takšne rešitve je v tem, da oblika in velikost ovire nista pomembni. Nekaj podobnega lahko poskusimo sprogramirati tudi sami.

Prav tako je mogoče precej izboljšati programsko kodo za vožnjo po črti, ki je v zdajšnji obliki še precej okorna. Uporaba več senzorjev za zaznavanje črte korenito izboljša delovanje avtomobilčka, predvsem pa ta vozi hitreje. Izdelek lahko izboljšamo z možnostjo, da se notranje kolo med zavijanjem vrtil počasneje kot zunanje, za kar pri pravih avtomobilskih poskrbi diferencial. Tudi namestitve različnih lučk, stikal in okrasitev lahko poveča atraktivnost našega avtomobilčka.

Vabimo vas, da nam pošljete svoje izboljšave in seveda fotografije izdelkov. Najbolj domiselne bomo objavili v reviji in nagradili.



- TN 1 motorni letalski RV-model basic 4 star
- TN 2 RV-jadrnica lipa
- 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 lovskega letala polikarpov I-15 bis
- TN 11 jadrni RV-model gita
- TN 12 racoon HLG-3
- TN 13 akrobat 40, trenažni motorni RV-model
- TN 14 maketa vodnega letala utva-66H
- TN 15 RV-model trajekta
- TN 16 spitfire, RV polmaketa za zračni boj
- TN 17 trener 40, trenažni motorni RV-model

- TN 18 lupo, elektromotorni RV-model
- TN 19 P-40 warhawk, RV-polmaketa za zračni boj
- TN 20 potepuh, RV-model motorne jahte
- TN 21 bambi, šolski jadrni RV-model
- TN 22 slovenka, RV-jadrnica metrskega razreda
- TN 23 e-trainer, trenažni RV-model z električnim pogonom
- TN 24 P-51 B/D mustang, RV-polmaketa zračne boje
- TN 25 messerschmitt Bf-109E, RV polmaketa za zračni boj
- TN 26 RV-polmaketa Aeronca L-3
- TN 27 fokker E III, RV-polmaketa park-fly
- TN 28 vektra, RV-model z električnim pogonom v potisni 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

Naročila sprejemamo na:

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

6,50 €*

*Cena posameznega načrta, k čemur prištetemo poštno stroške