

Prvi koraki v Arduino

Avtomobilček robot

Milan Gaberšek in Slavko Kocijančič

Robot je naprava, ki jo lahko programiramo tako, da opravlja neko nalogo. V enajstem zaporednem prispevku na temo Prvi koraki v Arduino bomo že zgrajeni avtomobilček predelali tako, da ga ne bomo več krmilili s tipkami, temveč bo deloval brez našega posredovanja. Napisali in naložili bomo tak program, da bo avtomobilček zaplesal, s čimer bo postal pravi robot. To bo prvi korak k njegovi samostojnosti. Nato bomo na sprednji del avtomobilčka pritrčili posebno tipalko, prek katere bo robotek zaznaval ovire na desni, levi in na sredini ter se ob primerno napisanem programu ustrezno odzval z umikom in spremembo smeri.

Računalniški pripomočki

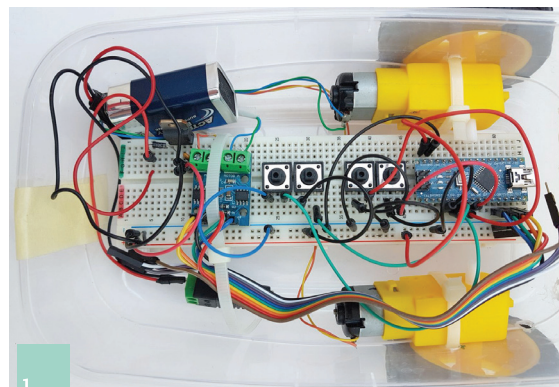
- osebni računalnik z nameščenim operacijskim sistemom Windows, Linux ali Mac OS,
- programsko razvojno okolje Arduino IDE, ki je brezplačno dostopno na spletni strani www.arduino.cc.

Material in orodje

- izdelani avtomobilček (predstavljen v prejšnji številki: TIM 10, junij 2020, str. 24–27),
- trši karton debeline 1 mm in velikosti približno 15 × 5 cm, odvisno od dimenzij avtomobilčka,
- aluminijasta folija velikosti približno 20 × 20 cm,
- gobica za pomivanje (ali podobna),
- tri žičke poljubnih barv dolžine približno 20 cm (v našem primeru bela, modra in oranžno bela),
- ščipalne kleščice, modelarski nožek oziroma kleščice za snemanje izolacije,
- pištola za vroče lepljenje, univerzalno lepilo,
- škarje, pisalo in ravnilo.

Ples avtomobilčka

Preden bomo obstoječi avtomobilček pripravili do plesa, bomo prototipno ploščico vstavili kar v ohišje. Pri tem bomo na motorčka prispajkane žičke obdržali, preostali



1

del kabla pa bomo odstranili. Žičke iz obeh motorčkov nato privijemo s pomočjo vijakov na sponkah modula L9110 (slika 1). Luknjico v ohišju lahko zaradi lepšega videza prelepimo kar z ličarskim lepilnim trakom. S tem je avtomobilček pripravljen in se lahko lotimo pisanja programa za njegov ples. Program lahko pretipkamo ali ga preprosto prenesemo s spletne strani www.drtsi.si/tim.html, kjer je dosegljiv tudi videoposnetek avtomobilčkovega plesanja.

```
//
// Program Arduino -
// ples avtomobilčka

// Pini za oba motorčka (preko modula L9110)
const int pinMotorA1 = 9;
const int pinMotorA2 = 10;
const int pinMotorB1 = 11;
const int pinMotorB2 = 12;

// Nastavitev izhodov krmilnika Arduina
void setup() {
  // Nastavitev izhodov za motorčka
  pinMode(pinMotorA1, OUTPUT);
  pinMode(pinMotorA2, OUTPUT);
  pinMode(pinMotorB1, OUTPUT);
  pinMode(pinMotorB2, OUTPUT);
}

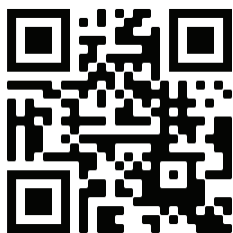
// Prva parametra povesta, katera pina poganjata motor
// tretji parameter pove smer:
// -1 kolo se vrti nazaj, 0 miruje, 1 kolo se vrti naprej
void motor(int pinMotor1, int pinMotor2, int nSmer) {
  if (nSmer == 1) { // Naprej
    digitalWrite(pinMotor1, HIGH);
    digitalWrite(pinMotor2, LOW);
  }
  else if (nSmer == -1) { // Nazaj
    digitalWrite(pinMotor1, LOW);
    digitalWrite(pinMotor2, HIGH);
  }
  else { // Se ne vrti
    digitalWrite(pinMotor1, LOW);
    digitalWrite(pinMotor2, LOW);
  }
}

void peljiNaprej(){
  motor(pinMotorA1, pinMotorA2, 1);
  delay(150);
  motor(pinMotorB1, pinMotorB2, 1);
}

void peljiNazaj(){
  motor(pinMotorA1, pinMotorA2, -1);
  delay(150);
  motor(pinMotorB1, pinMotorB2, -1);
}

void peljiVDesno(){
  motor(pinMotorA1, pinMotorA2, 1);
  motor(pinMotorB1, pinMotorB2, 0);
}

void peljiVLevo(){
  motor(pinMotorA1, pinMotorA2, 0);
  motor(pinMotorB1, pinMotorB2, 1);
}
```



Slika 1

V prejšnjem prispevku, objavljenem v junijskem Timu, smo izdelali avtomobilček, ki smo ga krmilili s pomočjo tipk, kar pomeni, da smo morali ves čas bdeti nad njim in ga usmerjati. Tokrat bomo obstoječi avtomobilček najprej predelali tako, da bo samostojno ponavljal izbrane načine vožnje, kar bo videti kot ples. Zaradi lažjega preskušanja delovanja koles bomo obstoječe tipke za krmiljenje kar ohranili, odstranili pa bomo kabel, saj bodo tipke namenjene zgolj testiranju. Pri preskušanja koles bomo tako na krmilnik le prenesli ustrezn program iz prejšnjega prispevka in s pomočjo tipk preverili delovanje. Kdor avtomobilčka še ni izdelal, lahko omenjene tipke tudi izpusti.

```

void vrtiVDesno(){
  motor(pinMotorA1, pinMotorA2, 1);
  delay(150);
  motor(pinMotorB1, pinMotorB2, -1);
}

void vrtiVLevo(){
  motor(pinMotorA1, pinMotorA2, -1);
  delay(150);
  motor(pinMotorB1, pinMotorB2, 1);
}

void ustavi(){
  motor(pinMotorA1, pinMotorA2, 0);
  motor(pinMotorB1, pinMotorB2, 0);
}

void loop() {
  // Petkrat ponovi
  for (int i=0; i<5; i++) {
    // Plesni koraki
    peljiNaprej(); delay(500);
    peljiVDesno(); delay(300);
    peljiNazaj(); delay(500);
    vrtiVLevo(); delay(800);
  }
  ustavi(); delay(100);
}

```

kolesa vrtijo v napačno smer, samo spremenimo številke pinov pri napovedi spremenljivk (na primer `const int pinMotorA1 = 9;` spremenimo v `const int pinMotorA1 = 10;`). Na začetku je tudi koristno, če avtomobilček postavimo na kak podstavek, da kolesa niso v stiku s tlemi, sicer nam bo bezljal po prostoru. S spreminjanjem ukazov lahko sprogramiramo ples po svoji izbiri, na primer dunajski valček ali kak sodobnejši ples. S posegom v zanko `for` lahko tudi povečamo število različnih gibanj na več kot pet.



Avtomobilček robot

Zdaj bomo model avtomobilčka v sprednjem delu nadgradili s posebno tipalko, s katero bo lahko zaznaval ovire. V resnici bomo na ohišju izdelali dve tipki in ju povezali s krmilnikom Arduino. Prek stanja tipalke (sklenjeno/nesklenjeno) bomo zaznali, da je avtomobilček naletel na oviro. Če bo ta na levi strani, se bo sklenila tipka na levem delu tipalke, kar bomo zaznali prek vhoda krmilnika Arduino; avtomobilček bomo zato pomaknili nazaj in ga usmerili v desno. V primeru ovire na desni strani in sklenjenosti tipke v desnem delu tipalke bomo avtomobilček pomakni-

V prvem delu programa kot običajno napovemo spremenljivke, katerih vrednosti so povezane z oznakami pinov krmilnika Arduino, in jih inicializiramo v okviru `void setup()`. S funkcijo `motor` nastavimo izhodne pine tako, da se ob vrednosti `nSmer` enako 1 motorček vrti naprej, pri -1 nazaj in pri 0 motorček stoji. Sledijo definicije funkcij `peljiNaprej()`, `peljiNazaj()`, `peljiLevo()`, `peljiDesno()` in `vrtiVLevo()`. Pri vseh funkcijah, ki zahtevajo gibanje obeh koles hkrati, je dodan ukaz za čakanje `delay(150);`. To vrstico kode smo dodali zato, ker motorček ob zagonu za nekaj trenutkov potrebuje zelo velik tok. Ker sta motorčka dva, je skupni začetni tok še večji, česar naše napajanje prek stabilizatorja 7805 ne more zagotoviti. To rešimo tako, da drugi motor zaženemo nekaj trenutkov za prvim, v našem primeru ta zakasnitev traja 150 milisekund.

V okviru `void loop()` v zanki `for`, ki se ponovi petkrat, kličeemo omenjene funkcije in povzročimo, da se bo avtomobilček najprej nekaj časa pomikal naprej, nato v desno, nato nazaj in se nazadnje obrnil v levo. Po petih ponovitvah bo za nekaj trenutkov obstal, nakar se bo zaradi zanke `loop` vse skupaj ponavljalo v neskončnost. Po prenosu programa se nam bo zdelo, da avtomobilček pleše. To bo počel vse do izklopa napajanja. Še enkrat naj spomnimo: če se



Slika 4a

S pištolo za vroče lepljenje na manjšo gobico nanese mo tanek sloj lepila in nanj položimo aluminijasto folijo tako, da je vsaj 3–4 cm gleda čez rob gobice.

Slika 4b

Zgornji del folije previdno zvijemo in pri tem pazimo, da je ne odtrgamo. Odvečni del folije odrežemo s škarjami.

Slika 4c

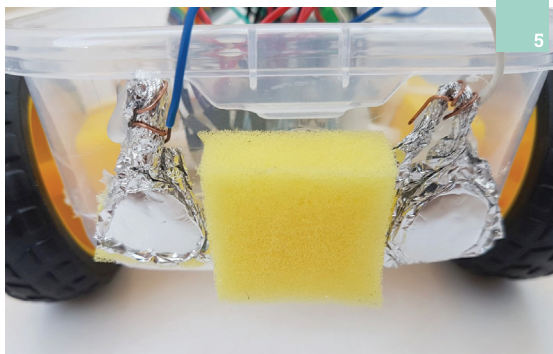
Na štrleči del aluminijaste folije navijemo na koncih olupljeno žičko dolžine približno 20 cm ter ga na koncu zavijemo skupaj. S tem smo zagotovili dober stik med folijo in žičko tudi brez spajkanja.

li nazaj in ga usmerili v levo. Če se bosta hkrati sklenili obe tipki na tipalki, ga bomo pomaknili nazaj in usmerili desno. Podobno žuželke prek tipalk zaznavajo ovire na svoji poti in se primerno odzovejo.

Najprej pripravimo vse potrebno gradivo (slika 2) za izdelavo tipalke. Iz gobice za pomivanje s škarjami izrežemo 1,5 cm debel kvader s stranicama 4 cm in še dva manjša kvadra debeline 1 cm s stranicama 2 cm. Oblika je lahko

Slika 6

Glede na mere avtomobilčka iz tršega kartona izrežemo ustrezno velik pravokotnik; v našem primeru je velik 15 × 5 cm. Folijo na vrhu zvijemo in pritrdimo žičko, podobno kot pri manjših gobicah.



tudi drugačna (slika 3). Z aluminijasto folijo ovite gobice (slike 4a, b in c) z vročim lepilom nalepimo na sprednji del ohišja, kot kaže slika 5.

Iz 1 mm debelega kartona izdelamo zadnji del tipalke, ki bo hkrati tudi drugi kontakt za bodoči tipki. Kot mate-



```
//
// Program Arduino -
// avtomobilček robot

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

// Pini za oba motorčka (preko modula L9110)
const int pinMotorA1 = 9;
const int pinMotorA2 = 10;
const int pinMotorB1 = 11;
const int pinMotorB2 = 12;

// Nastavitev vhodov in izhodov krmilnika Arduina
void setup() {
  // Nastavitev tipk
  pinMode(pinTipkaLeva, INPUT_PULLUP);
  pinMode(pinTipkaDesna, INPUT_PULLUP);
  // Nastavitev izhodov za motorčka
  pinMode(pinMotorA1, OUTPUT);
  pinMode(pinMotorA2, OUTPUT);
  pinMode(pinMotorB1, OUTPUT);
  pinMode(pinMotorB2, OUTPUT);
}

// Prva parametra povesta, katera pina poganjata motor
// tretji parameter pove smer:
// -1 kolo se vrti nazaj, 0 miruje, 1 kolo se vrti naprej
void motor(int pinMotor1, int pinMotor2, int nSmer) {
  if (nSmer == 1) { // Naprej
    digitalWrite(pinMotor1, HIGH);
    digitalWrite(pinMotor2, LOW);
  }
  else if (nSmer == -1) { // Nazaj
    digitalWrite(pinMotor1, LOW);
    digitalWrite(pinMotor2, HIGH);
  }
  else { // Se ne vrti
    digitalWrite(pinMotor1, LOW);
    digitalWrite(pinMotor2, LOW);
  }
}

void peljiNaprej(){
  motor(pinMotorA1, pinMotorA2, 1);
  delay(150);
  motor(pinMotorB1, pinMotorB2, 1);
}

void peljiNazaj(){
  motor(pinMotorA1, pinMotorA2, -1);
  delay(150);
  motor(pinMotorB1, pinMotorB2, -1);
}
```

```
}

void peljiVDesno(){
  motor(pinMotorA1, pinMotorA2, 1);
  motor(pinMotorB1, pinMotorB2, 0);
}

void peljiVLevo(){
  motor(pinMotorA1, pinMotorA2, 0);
  motor(pinMotorB1, pinMotorB2, 1);
}

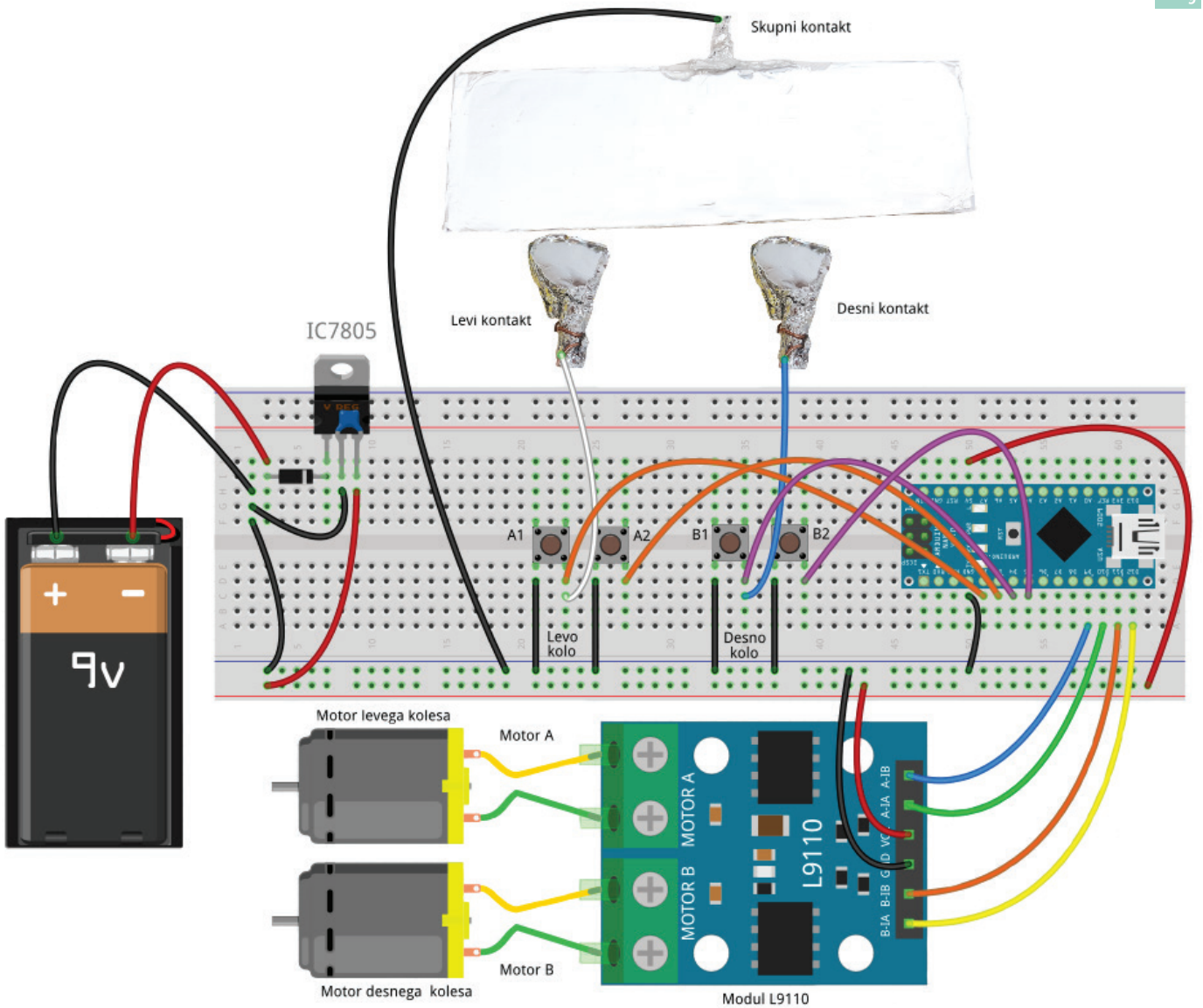
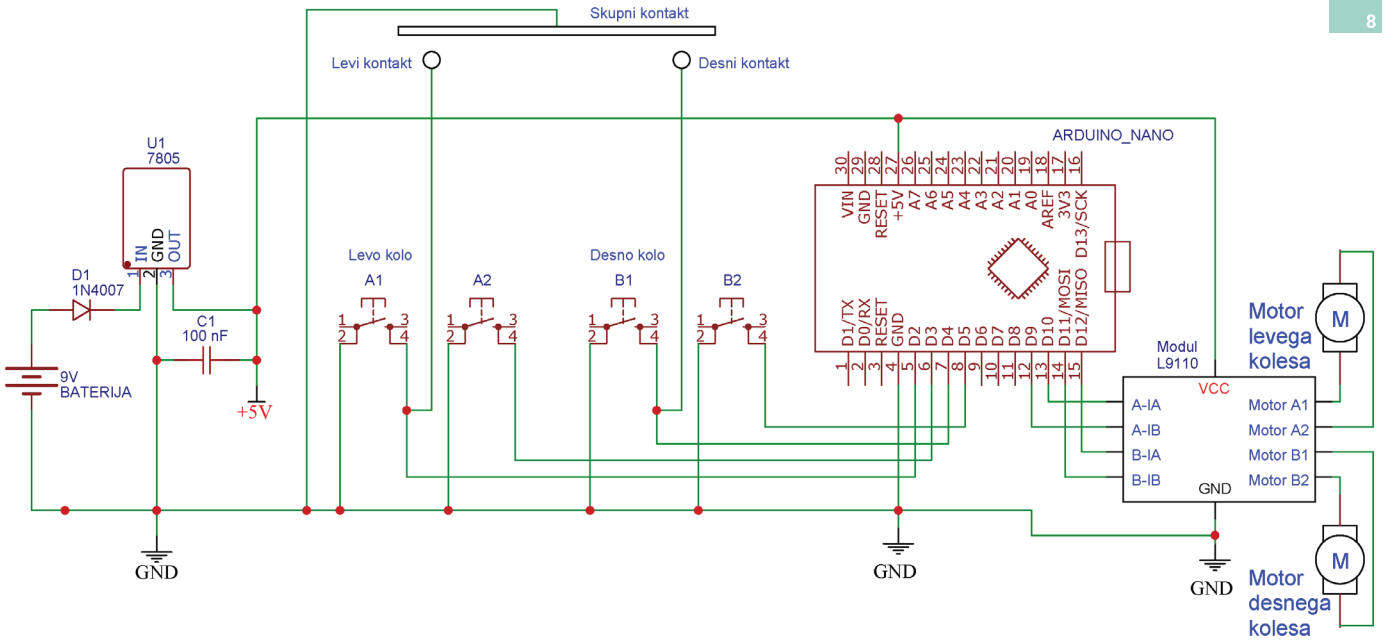
void vrtiVDesno(){
  motor(pinMotorA1, pinMotorA2, 1);
  delay(150);
  motor(pinMotorB1, pinMotorB2, -1);
}

void vrtiVLevo(){
  motor(pinMotorA1, pinMotorA2, -1);
  delay(150);
  motor(pinMotorB1, pinMotorB2, 1);
}

void ustavi(){
  motor(pinMotorA1, pinMotorA2, 0);
  motor(pinMotorA1, pinMotorA2, 0);
}

void loop() {
  if ((digitalRead(pinTipkaLeva) == LOW)
    && (digitalRead(pinTipkaDesna) == LOW)) {
    // Obe tipki sta sklenjeni, ovira pred nami,
    // zato pomik nazaj, nato obrni v desno
    peljiNazaj(); delay(300);
    vrtiVDesno(); delay(400);
  }
  else if (digitalRead(pinTipkaLeva) == LOW) {
    // Pritisnjena leva tipka, ovira na levi,
    // zato pomik nazaj, nato obrni v desno
    peljiNazaj(); delay(300);
    vrtiVDesno(); delay(400);
  }
  else if (digitalRead(pinTipkaDesna) == LOW) {
    // Pritisnjena desna tipka, ovira na desni,
    // zato pomik nazaj, nato obrni v levo
    peljiNazaj(); delay(300);
    vrtiVLevo(); delay(400);
  }
  else {
    // Pritisnjena nobena tipka, vozi naprej
    peljiNaprej();
  }
}
```





Slika 7

V enajstem nadaljevanju serije Prvi koraki v Arduino smo izdelali pravega robota. S spremembo programa smo dosegli odzive avtomobilčka, ki je najprej plesal, nato pa se je odzival še na ovire. Ob uporabi tipk lahko s primernim programom preverimo tudi delovanje posameznih koles. Prav to je velika prednost robotov – s spremembo programa namreč lahko spremenimo njihovo delovanje, ne da bi morali spreminjati konstrukcijo in električno vezje. Tako bi lahko napisali povsem drugačen program, naredili bi lahko na primer prenašalca listkov s sporočili ali kaj podobnega.

rial lahko uporabimo kar odpadno embalažo. Karton prelepimo z aluminijasto folijo, vendar tokrat raje uporabimo univerzalno lepilo, da brez potrebe ne povečujemo skupne teže avtomobilčka. Tipalo s pomočjo vročega lepila nalepimo na večjo gobico, ki smo jo pred tem že nalepili na ohišje. Pri tem pazimo, da je veliki aluminijasti kontakt obrnjen proti manjšima kontaktoma, razdalja med njimi pa je 3–5 mm (slika 7).

Tako izdelana tipalka vsebuje dve tipki. Večji kontakt, ki je skupen obema tipkama, povežemo s sponko GND na prototipni ploščici. To je razvidno s sheme (slika 8), še jasneje pa s slike 9, pripravljene s programom Fritzing, kjer je žička GND obarvana črno. Belo in modro žičko z manjših gobic na prototipni ploščici priključimo k žičkama pri tipkah za krmiljenje levega in desnega kolesa naprej, ki sta že povezani s krmilnikom Arduino (pina D2 in D4). V resnici smo s tem vsaki od obstoječih tipk na prototipni ploščici vzporedno vezali levo in desno tipko tipalke. S tem smo dosegli, da lahko uporabljamo tipki tipalke ali tipke na prototipni ploščici. Če avtomobilček gradimo na novo in tipk na prototipni ploščici nimamo, lahko levo tipko tipalke povežemo neposredno na pin D2, desno pa na pin D4.

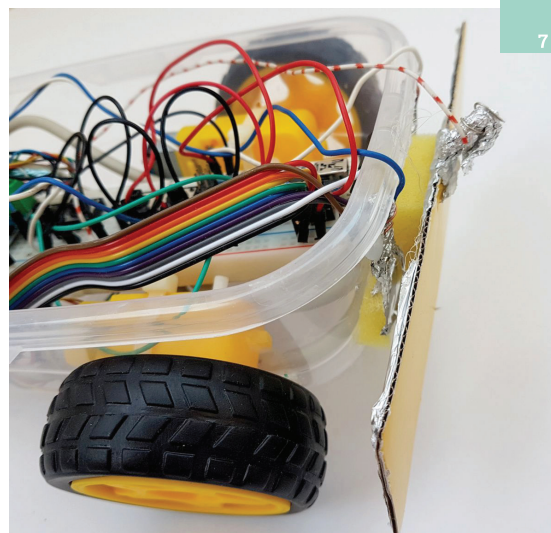
Mehanski del nadgradnje je s tem končan (slika 10). Manjka le še ustrezn program, ki ga na krmilnik Arduino prenesemo kar s spletne strani www.drta.si/tim.html, kjer si je mogoče ogledati tudi videoposnetek delovanja avtomobilčka robota.

Program je zelo podoben tistemu za ples. Razlika je le v dodani napovedi uporabe obeh tipk tipalke in v spremenjeni funkciji loop(). Znotraj programske zanke loop() ves čas preverjamo stanje pritisnjenosti tipk tipalke z ukazom digitalRead(tipka) == LOW in odzivom prek funkcij peljiNaprej(), vrtiVDesno() in vrtiVLevo(). S spremembo vrednosti funkcije delay(milisekund); lahko spreminjamo čas trajanja in s tem način odziva avtomobilčka. Po prenosu programa lahko preskusimo različne možnosti odziva na oviro. Najbolje je, da sta kabel USB in napajanje 9 V med preskušanjem ves čas priključena na avtomobilček, avtomobilček pa postavimo na manjši podstavek, da kolesa niso v stiku s tlemi, sicer ga bomo morali loviti po prostoru. Šele takrat, ko vse deluje tako, kot je treba, lahko odstranimo kabel USB in avtomobilček priključimo na baterijo, da je ne bomo po nepotrebnem praznili.

O napajanju

Zaradi uporabe dveh motorčkov se 9-voltna baterija zelo hitro izprazni. Podatek 300 mAh pomeni, da nam eno uro lahko zagotavlja tok 300 mA, kar je za naš avtomobilček razmeroma malo. Ena od rešitev je uporaba šestih zaporedno vezanih 1,5 V baterij AA, s čimer dobimo napajalno napetost $6 \times 1,5 \text{ V} = 9 \text{ V}$ (slika 11). Še boljša in dolgoročno cenejša rešitev je uporaba polnilnih baterij. Žal imajo te običajno nazivno vrednost 1,2 V, kar pomeni $6 \times 1,2 \text{ V} = 7,2 \text{ V}$. Napetost je torej nekoliko manjša, vendar na srečo vseeno zadostuje. Razveseljivo je, da obstajajo tudi take z zmogljivostjo 2000 mAh in več, kar pomeni, da eno uro zagotavljajo tok več kot 2 A. To pa povsem zadostuje za večino samostojnih projektov.

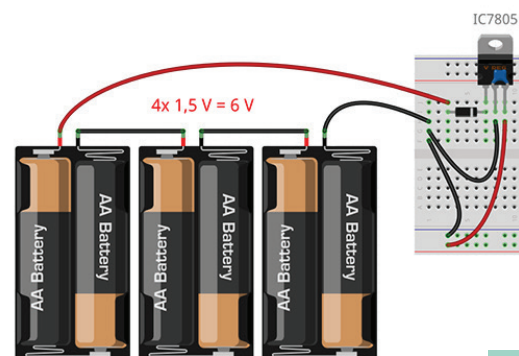
Če želimo hitreje vrtenje motorčkov, lahko napajanje vezja L9110 namesto prek stabilizatorja 7805, ki zagotavlja 5 V, vezemo neposredno za diodo. S tem zvišamo napajalno napetost vezja L9110 in povečamo hitrost vrtenja motorčkov.



7



10



11

Sklep

Kot vedno si v okviru obstoječega projekta lahko zastavimo dodatne naloge. Prva zares priročna stvar je dodano stikalo za vklop in izklop napajanja. Avtomobilčku lahko z neposrednim priklopom na vir napetosti dodamo tudi svetleče diode namesto luči, pri čemer ne pozabimo na ustrezne predupore. Dodamo lahko tudi vzvratne luči, ki bodo delovale le pri vzvratni vožnji. V tem primeru jih seveda povežemo na katerega od prostih digitalnih izhodov krmilnika Arduino. Najspretnejši lahko z uporabo svetlobnega senzorja, ki ga z dodatnim uporom ustrezno priključijo na katerega od analognih vhodov, poskrbijo, da bodo luči vključene samo ponoči. V zadnjih dveh primerih je seveda treba ustrezno prilagoditi tudi programsko kodo.