PRVI KORAKI V ARDUINO – AVTOMOBILČEK

Milan Gaberšek in Slavko Kocijančič

V desetem zaporednem prispevku na temo Prvi koraki v Arduino bomo krmilnik Arduino uporabili za krmiljenje modela avtomobilčka. Prek štirih tipk bomo lahko avtomobilček premikali levo, desno, naprej in nazaj. Pri tem bomo poleg Arduina uporabili tiskano vezje H-mostič – modul z dvema integriranima vezjema L9110. Modul omogoča, da Arduino lahko krmili dva enosmerna motorčka za poganjanje levega in desnega kolesa avtomobilčka. V naslednjih prispevkih bomo model avtomobilčka nadgradili tako, da tipke za ročno krmiljenje ne bodo potrebne in bo vožnja avtonomna.

Material

- krmilnik Arduino Nano ali podoben,
- USB-kabel (mini USB) za povezavo krmilnika z računalnikom,
- prototipna ploščica (angl. breadboard),
- usmerniška dioda 1N4007 (ali primerljiva),
- stabilizator napetosti LM 7805,
- keramični kondenzator 100 nF (ali primerljiv),
- modul L9110 za dva motorja (ali primerljiv H-mostič)
- štiri mikrotipke (ali podobne),
- dva motorčka z reduktorjem, ki delujeta pri napetosti od 3 V do 6 V,
- vezne žičke (najbolje 4-krat rdeča, 2-krat vijoličasta, 2-krat rjava in 9-krat črna),
- ploščati (angl. flat) kabel s šest ali več barvnimi veznimi žičkami in konektorji F-M dupont breadboard dolžine 20 cm (za priklop modula lahko uporabimo tudi šest posameznih žičk z ustreznimi konektorji),
- internetni UTP-kabel dolžine vsaj en meter za povezavo z avtomobilčkom (lahko tudi 2-krat dvožilni kabel enake dolžine),
- 9-V baterija ali enosmerni napetostni vir 9 V,
- ohišje avtomobilčka (lahko kar posodica s pokrovom za prehrano),
- dve kabelski vezici za pritrditev motorčka,
- ščipalne klešče, modelarski nožek oziroma klešče za snemanje izolacije,
- spajkalnik z žičko za spajkanje (priporočljivo),
- flomaster za označitev luknje v ohišju,
- vrtalnik s svedri premera 5 oziroma 6 mm (odvisno od uporabljenega motorčka in žičk),
- izvijač,
- elastike ali vezice za pritrditev modula L9110 in baterije na prototipno ploščico.

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*.

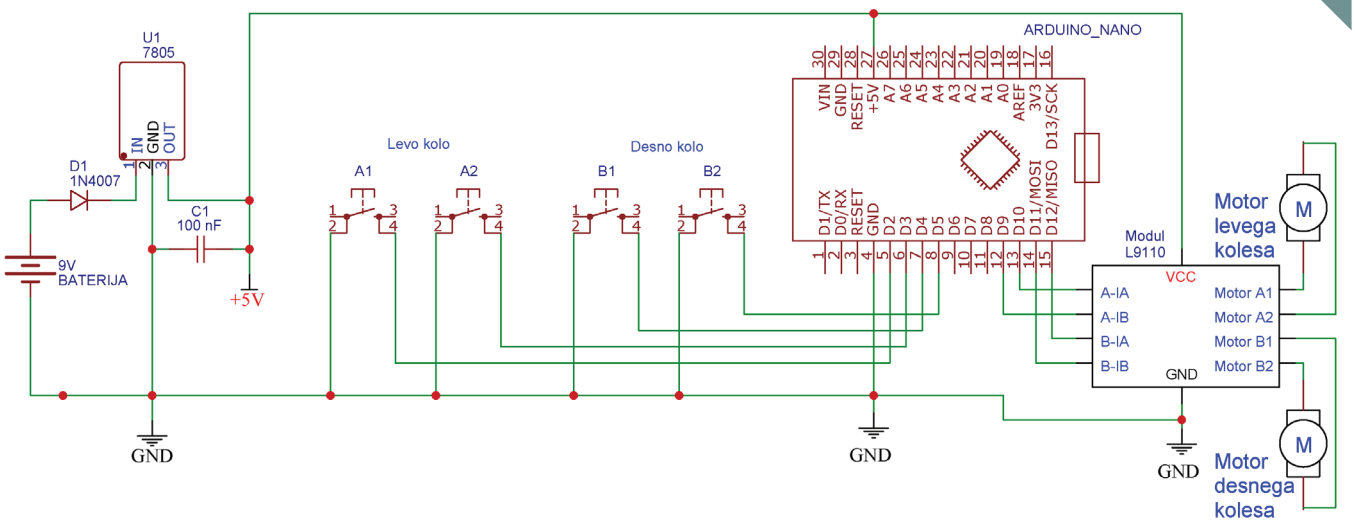
Izdelava avtomobilčka

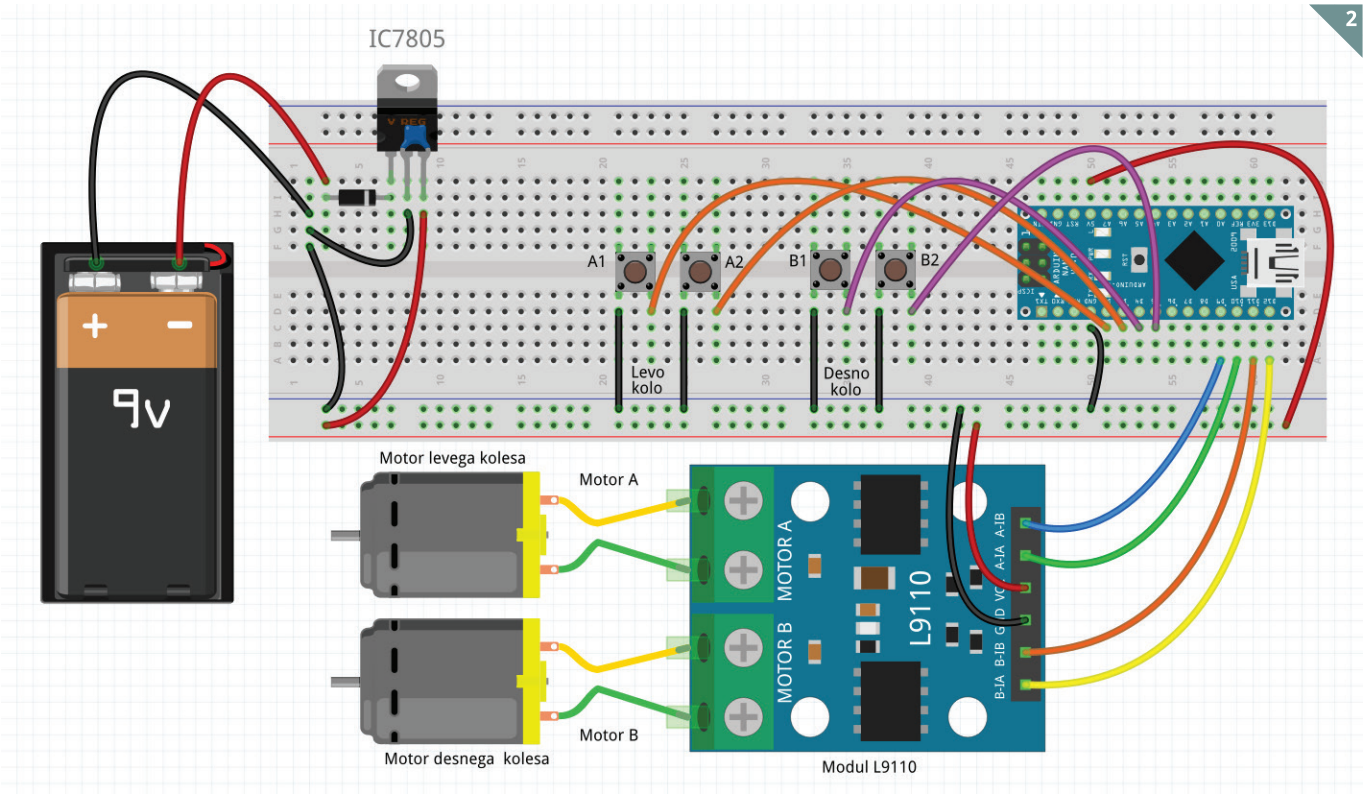
Najprej glede na električno shemo (slika 1) oziroma računalniško sliko, narejeno s pomočjo programa Fritzing (slika 2), sestavimo vezje. Tokrat bomo vezje že od začetka napajali z 9-V baterijo oziroma drugim primerljivim enosmernim virom električne napetosti. Kot smo omenili v prejšnjem prispevku z naslovom Prvi koraki v Arduino – temperaturno krmiljen ventilator (TIM-9, maj 2020), bomo vir priklopili na stabilizator napetosti z oznako LM7805, ki bo napetost znižal na 5 V in jo stabiliziral. Če imamo ta del že sestavljen, ga lahko uporabimo brez sprememb. Seveda je pri sestavljanju vir napetosti izklopljen, kar pomeni, da žička iz vira ni vstavljena v prototipno ploščico. Krmilnik Arduino povežemo z napajanjem (rdeča vezna žička) in GND (črna vezna žička). Sledi vstavljanje tipk, ki jih povežemo z GND in ustreznimi pini krmilnika Arduino, pri čemer smo pozorni na to, da mikrotipke pravilno obrnemo (podrobneje o tem v prispevku Prvi koraki v Arduino – semafor za pešce s tipko, TIM-3, (november 2019).

Z Arduinom moramo povezati tudi modul L9110 (slika 3) in oba motorčka z reduktorjem. H-mostič s krmilnikom povežemo s šestimi barvnimi veznimi žičkami. Slednje imajo na enem koncu ženske konektorje dupont, ki jih pritrdimo na pine na modulu, na drugem pa moške konektorje dupont, ki jih vstavimo v prototipno ploščico. Za lažje delo lahko uporabimo kar ploščati kabel s šest ali več barvnimi veznimi žičkami in konektorji F-M dupont breadboard dolžine 20 cm (slika 4). Kakšna od veznih žičk lahko ostane neuporabljena, prav nam bo prišla pri morebitni nadgradnji modela. Pri sestavljanju je najbolje, da uporabimo žičke enake barve, kot so prikazane na omenjeni računalniški sliki vezja (slika 2). S tem se izognemo napakam pri povezovanju. Pozorni bodimo na oznake na samem modulu, saj se te pri različnih proizvajalcih lahko razlikujejo tako po oznakah kot po razporeditvi pinov. Posebno pozorni moramo biti pri povezavi rdeče napajalne žičke na pin VCC in črne žičke na pin GND, oba na modulu. Modul L9110 na računalniški sliki je narisan povečano. Dejanski modul je običajno zelo majhen, zato je pogosto, še posebno ob slabih svetlobnih pogojih, težko razbrati napise na pinih. V tem primeru si lahko pomagamo z lupo ali kar s fotografijo oziroma namensko aplikacijo na mobilnem telefonu.

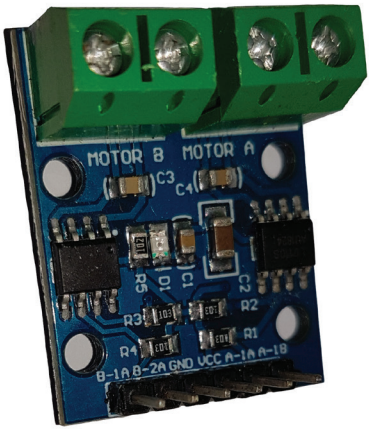
Namesto uporabe modula L9110 bi lahko sestavili vezje H-mostič s pomočjo tranzistorjev, uporov in nekaj zaščitnih elementov, vendar bi za izdelavo porabili veliko časa, velikost projekta bi narasla, povečala bi se tudi možnost napak, razumljivost in preglednost vezja kot celote bi bila slabša. Zato je veliko preprosteje in tudi ceneje kupiti že izdelan modul. Obstajajo podobni moduli različnih proizvajalcev, na tržišču lahko najdemo tudi zmoglji-

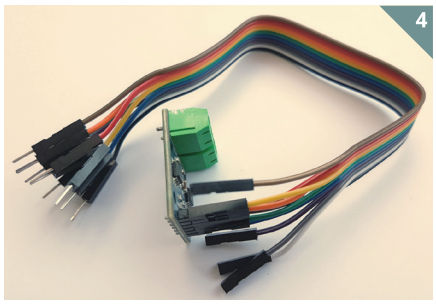
1

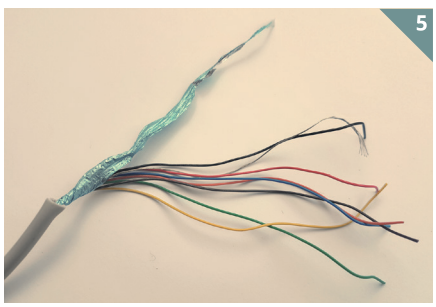


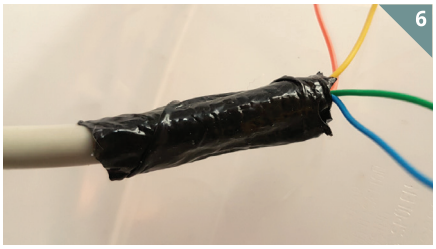


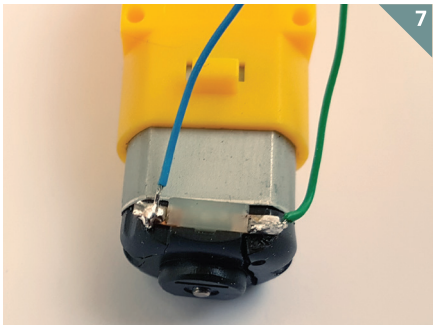
Pozorni bodite na oznake in razpored pinov!

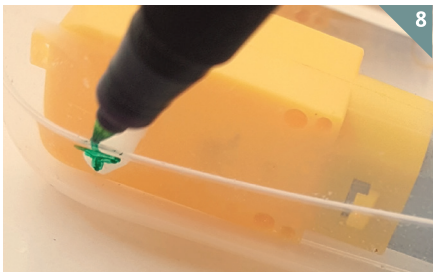


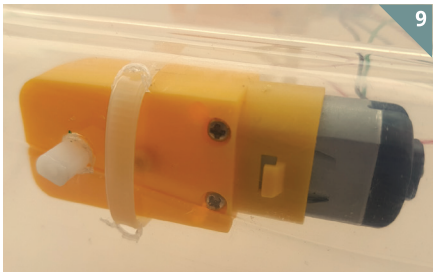












vejše. Uporabljeni modul L9110 prenese tok do 800 mA, napajalna napetost pa se lahko giblje od 2,6 do 12 V.

Motorčka povežemo s pomočjo dveh daljših dvožilnih žičk dolžine vsaj en meter. Priporočljivo je uporabiti kar klasični večžilni internetni UTP-kabel (slika 5), saj je kabel običajno tanek, gibljiv, v notranjosti pa so žičke različnih barv, kar olajša priključitev na motorčka in modul L9110. Neuporabljene žičke ne odstranimo, saj jih bomo lahko uporabili pri nadgradnji modela. Zato jih po potrebi še dodatno povijemo z izolirnim trakom, jih zapognemo nazaj ter povijemo kar k UTP-kablu (slika 6). Motorčka morata imeti vgrajen reduktor, ki zmanjša število vrtljajev. Hkrati se poveča tudi navor, kar pomeni, da se bo avtomobilček sploh lahko premikal. Če imamo možnost uporabe spajkalnika, je najbolje žičkam na koncu odstraniti izolacijo ter jih prispajkati na konektorje motorčka (slika 7). Tako nas ne bo skrbelo, da bi med montažo oziroma delovanjem žički odpadli in bi avtomobilček prenehal delovati.

Motorček vstavimo v plastično ohišje in označimo mesto, kjer bo gred reduktorja pogledala iz ohišja (slika 8). Pred pritrditvijo motorčka z reduktorjem obvezno preverimo, ali se kolo dotika tal in ali ne drsa po ohišju, sicer moramo mesto izvrtine prilagoditi. Ko je mesto ustrezno izbrano, izvrtamo luknjo za gred in dodatni luknji za vezico, s katero bomo motorček pričvrstili ob ohišje (slika 9). Ko je motorček nameščen, pritrdimo še kolo (slika 10). Na nasprotni strani ohišja enako storimo še z drugim motorčkom, pri čemer pazimo, da sta gredi reduktorja levega in desnega motorčka poravnani (na skupni premici), saj drugače avtomobilček ne bo vozil naravnost (slika 11). Iz slike je razvidno tudi, da smo v ohišje navrtali dodatno

luknjo za internetni UTP-kabel, ki smo ga pred izstopom iz ohišja zavili v zanko, s čimer preprečimo, da bi se ob premikanju avtomobilčka zaradi napetih žičk te odtrgale od motorčkov.

Manjka nam le še povezava žičk od avtomobilčka do modula L9110 (slika 12). Z izvijačem najprej odvijemo vijak na vijačni sponki, pri čemer pazimo, da nam ta ne pade s sponke. Nato žički, ki smo ji odstranili izolacijo, vstavimo v sponko in privijemo vijak, s čimer zagotovimo trden spoj. Če je vse v redu, žička ne sme izpasti iz sponke. Podobno naredimo za vse uporabljene žičke. Ko končamo s pritrjevanjem žičk, lahko ohišje pokrijemo s pokrovom. Modul L9110 in baterijo imamo lahko na mizi ob prototipni ploščici, lahko pa ju z vezico pritrdimo na prototipno ploščico (slika 13), s čimer bomo lažje krmilili avtomobilček. Izdelava modela avtomobilčka je tako končana (slika 14).

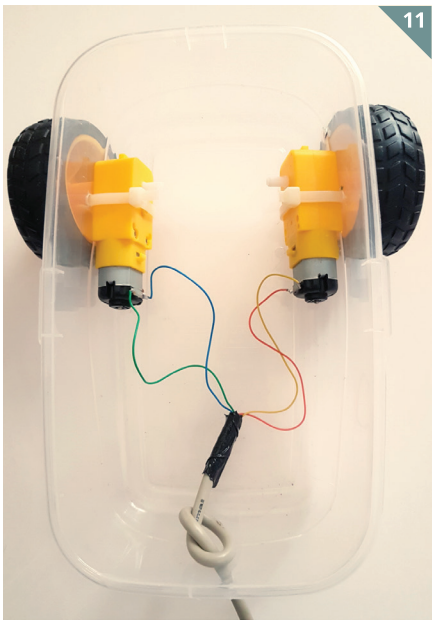
Izdelava programa z uporabo funkcij

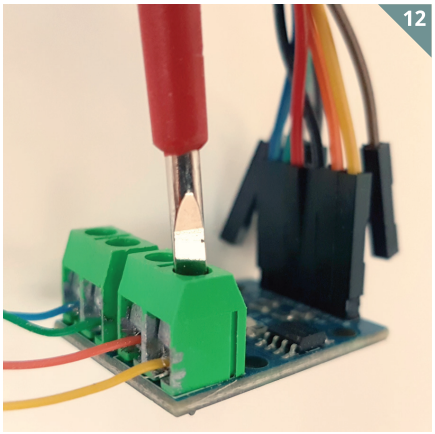
Sledi izdelava programa in prenos na krmilnik Arduino. Program lahko pretipkamo ali ga prenesemo s spletne strani *www.drti.si/tim.html.*

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

// Pina tipk za krmiljenje motorja levega
kolesa
const int pinTipkaA1 = 2;
const int pinTipkaA2 = 3;
// Pina tipk za krmiljenje motorja desnega ko-
lesa
const int pinTipkaB1 = 4;
const int pinTipkaB2 = 5;
//Pina za krmiljenje

// Pina motorčka levega kolesa za modul
L9110
const int pinMotorA1 = 9;
const int pinMotorA2 = 10;

// Pina motorčka desnega kolesa za modul
L9110
const int pinMotorB1 = 11;
const int pinMotorB2 = 12;

// Nastavitev vhodov in izhodov krmilnika
Arduina
void setup() {
  // Nastavitev tipk
  pinMode(pinTipkaA1, INPUT_PULLUP);
  pinMode(pinTipkaA2, INPUT_PULLUP);
  pinMode(pinTipkaB1, INPUT_PULLUP);
  pinMode(pinTipkaB2, INPUT_PULLUP);
  // Nastavitev izhodov za motorčka
  pinMode(pinMotorA1, OUTPUT);
  pinMode(pinMotorA2, OUTPUT);
  pinMode(pinMotorB1, OUTPUT);
  pinMode(pinMotorB2, OUTPUT);
}

// Prva parametra povesta, katera pina poga-
njata 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);
  }
}

// Preveri stanje tipk in vrni vrednost
// true, če je pogoj izpolnjen, oziroma false,
če ni
bool preveriTipki(int pinTipka1, bool stanje1,
int pinTipka2, bool stanje2) {
  if ( (digitalRead(pinTipka1) == stanje1)
    && (digitalRead(pinTipka2) == stanje2) ) {
      return true;
  }
  else {
    return false;
  }
}

void loop() {
  // Motor levega kolesa
  int stanjeMotorja = 0; // Privzeto stanje -
  kolo se ne vrti
    if (preveriTipki(pinTipkaA1, HIGH, pinTip
    kaA2, LOW)) {
    stanjeMotorja = 1;
  }
  else if (preveriTipki(pinTipkaA1, LOW, pin
  TipkaA2, HIGH)) {
    stanjeMotorja = -1;
  }
  motor(pinMotorA1, pinMotorA2,
  stanjeMotorja);

  // Motor desnega kolesa
    int stanjeMotorja2 = 0; // Privzeto stanje -
    kolo se ne vrti
    if (preveriTipki(pinTipkaB1, HIGH,
      pinTipkaB2, LOW)) {
        stanjeMotorja2 = 1;
  }
  else if (preveriTipki(pinTipkaB1, LOW, pin
    TipkaB2, HIGH)) {
      stanjeMotorja2 = -1;
  }
  motor(pinMotorB1, pinMotorB2,
  stanjeMotorja2);
}
```

Za lažje testiranje delovanja je model priporočljivo obrniti na glavo, torej tako, da so kolesa dvignjena od tal. S tem preprečimo, da bi nam avtomobilček ves čas uhajal. Če bomo po prenosu programa pritisnili tipko A1, se bo levo kolo začelo vrteti naprej, če pa bomo pritisnili tipko A2, se bo kolo vrtelo nazaj. Če ne bo tako, moramo v programu zamenjati številko obeh pinov za levi motor. To storimo tako,


10


11


12

da pri najavi spremenljivke const int pinMotorA1 = 9; vrednost 9 spremenimo v 10, pri pinMotorA2 pa iz 10 v 9. Podobno s tipkama B1 in B2 preverimo delovanje desnega motorčka. Če se pri pritisku B1 desno kolo ne vrti naprej, izvedemo postopek za zamenjavo pinov, le da tokrat zamenjamo vrednosti pinMotorB1 in pinMotorB2. Ko se kolesi vrtita v pravo smer, lahko avtomobilček postavimo na tla. S hkratnim pritiskom tipk A1 in B1 se avtomobilček pomika naprej. S hkratnim pritiskom obeh tipk A2 in B2 se avtomobilček pomika nazaj, medtem ko s tipkama A1 oziroma B1 lahko zavijamo. Obstajajo še štiri kombinacije tipk, med katerimi je tudi taka, da se avtomobilček zavrti na mestu.

Delovanje programa

V začetnem delu programa kot običajno s spremenljivkami določimo, na katere pine krmilnika Arduino smo priklopili posamezne elemente. Tipkam smo priredili spremenljivke z imeni oblike pinTipka (npr. pinTipkaA1 za tipko A1). Pri motorčku smo zaradi vrtenja naprej in nazaj uporabili dva pina. Spremenljivke motorčka smo poimenovali s pinMotor (na primer pinMotorA1 in pinMotorA2 za levi motor). V okviru void setup() {} krmilniku določimo, kako bomo uporabljali posamezne pine. S parametrom INPUT_PULLUP pri tipkah vklopimo notranji upor, kar poenostavi izdelavo vezja. S parametrom OUTPUT določimo, da pin uporabljamo kot digitalni izhod.

Novost v programu je najava in uporaba lastnih funkcij. Napisali smo dve funkciji, motor in preveriTipki. Funkcije uporabljamo takrat, ko želimo del kode napisati enkrat in jo uporabljati večkrat, ne da bi pri tem podvajali kodo. S tem so programi krajši, hkrati pa hitreje odpravimo morebitne napake v delovanju. Če želimo znotraj funkcije le izvesti določeno kodo, pred imenom funkcije napišemo void. Taka je na primer funkcija motor, v kateri so ukazi za krmiljenje motorja. Pri funkciji preveriTipki pa želimo preveriti, ali je določena tipka za krmiljenje avtomobilčka pritisnjena, zato mora funkcija vrniti neko vrednost. V tem primeru je rezultat lahko resničen ali neresničen, zato pred imenom funkcije napišemo bool (okrajšava za boolean), s čimer bo funkcija vračala vrednosti true oziroma false.

Najprej si na hitro poglejmo, kaj se dogaja v glavni zanki void loop(){}. Z int stanjeMotorja = 0; najprej nastavimo spremenljivko stanjeMotorja na vrednost 0. Če se pri preverjanju obeh tipk A1 in A2 v obeh if-stavkih vrednost spremenljivke stanjeMotorja ne bo spremenilo, bo po klicu funkcije motor levo kolo stalo. Če bo tipka A1 na HIGH in A2 na LOW, bo pogoj izpolnjen, zato bo stanjeMotorja dobilo vrednost 1. To pomeni, da se bo po klicu funkcije motor levo kolo vrtelo naprej. Podobno se bo ob A1 na LOW, A2 na HIGH in vrednosti stanjeMotorja = –1 levo kolo vrtelo nazaj. Koda za desni motorček je skorajda identična.

Kot vidimo, nam za uporabo funkcij sploh ni treba vedeti, kako te delujejo. To je velika prednost, saj nam lahko funkcije kdo prej pripravi, mi pa jih le še uporabimo. Tako obstajajo raznovrstne knjižnice funkcij za delo z različnimi senzorji, ki jih lahko priklopimo na krmilnik Arduino. Še več, tudi ukazi, ki smo jih uporabljali, na primer digitalRead ali digitalWrite, so v resnici funkcije, le da so vgrajene kot del razvojnega okolja. Celo void setup() {} je le posebna funkcija, ki jo krmilnik Arduino izvede kot prvo po vklopu, za njo pa v neskončni zanki kliče funkcijo void loop() {}.

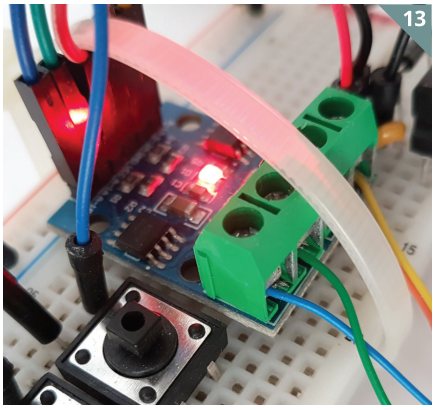
Zdaj si bomo podrobneje ogledali, kako sta zgrajeni funkciji motor in preveriTipki. Pri obeh funkcijah sta za imenom oklepaja, znotraj katerih so navedeni parametri funkcije. Ti parametri so dejansko imena spremenljivk, katerih vrednost določimo ob klicu funkcije in jih nato uporabljamo znotraj njene kode.

Pri funkciji void motor(int pinMotor1, int pinMotor2, int nSmer) tako podamo pine posameznega motorja in smer vrtenja. V okviru zanke void loop() {} jo kličemo z ukazoma motor(pinMotorA1, pinMotorA2, stanjeMotorja); in motor(pinMotorB1, pinMotorB2, stanjeMotorja2);. Znotraj funkcije najprej preverimo vrednost parametra nSmer. Če je ta enak 1, potem izvedemo ukaza digitalWrite(pinMotor1, HIGH); in digitalWrite(pinMotor2, LOW);. To pomeni, da se prvi pin, katerega številko dobimo prek parametra pinMotor1, nastavi na vrednost HIGH, drugi pa na LOW. Ker se posledično med obema pinoma pojavi napetostna razlika, začne teči električni tok, ki prek krmilnega modula L9110 zagotovi vrtenje kolesa v smeri naprej. Če je vrednost parametra nSmer enaka –1, nastavimo ista pina na vrednosti LOW in HIGH, kar pomeni, da teče električni tok v drugi smeri in se posledično tudi kolo vrti v drugo smer. Če je vrednost parametra nSmer enaka 0, postavimo oba pina na vrednost LOW in motorček se ne vrti.

Funkcijo preveriTipki najavimo z ukazom bool preveriTipki(int pinTipka1, bool stanje1, int pinTipka2, bool stanje2). Funkcija dobi prek parametrov pinTipka1 in pinTipka2 številki pinov obeh tipk ter stanji stanje1 in stanje2, ki ju želimo preveriti. Z ukazom digitalRead(pinTipka1) znotraj funkcije dobimo dejansko stanje na tej tipki, ki jo z ukazom digitalRead (pinTipka1) == stanje1 primerjamo z vrednostjo spremenljivke stanje1. Podobno naredimo še s tipko na pinu pinTipka2. Z ukazom && izvedemo nad dobljenima vrednostma logični IN, kar pomeni, da je pogoj resničen le takrat, kadar sta glede na vhodne parametre obe tipki v želenem stanju. Če je pogoj resničen, se zaradi ukaza if () {} izvede ukaz return true;, kar pomeni, da funkcija vrne vrednost true. Če pogoj ni izpolnjen, vrne funkcija vrednost false. Poglejmo si primer, ko je pritisnjena le tipka A1 in je bila funkcija preveriTipki klicana iz vrstice if (preveriTipki(pinTipkaA1, HIGH, pinTipkaA2, LOW)) { stanjeMotorja = 1; }. Ker je pogoj izpolnjen, saj je A1 HIGH in A2 LOW, dobi po klicu funkcije stanjeMotorja vrednost 1. Posledično se po izvedbi ukaza motor (pinMotorA1, pinMotorA2, stanjeMotorja); zavrti levo kolo.

Zaključek

Izdelava avtomobilčka je zanimiva tudi zato, ker moramo poleg elektronskega dela nekaj časa posvetiti tudi izdelavi modela. Uporabljeni modul L9110 zelo poenostavi krmiljenje obeh motorčkov z reduktorjema, ki ju lahko preprosto uporabimo tudi pri drugih projektih. V okviru programskega dela smo spoznali delo s funkcijami, s čimer smo nadgradili tudi znanje programiranja. Na začetku ima marsikdo težave z razumevanjem


13


14

delovanja in uporabe funkcij, vendar se s prakso težave počasi razblinijo. Obstoječi model avtomobilčka bomo v prihodnjih prispevkih nadgradili tako, da ne bo več potreboval tipk in bo deloval samostojno kot pravi robot. Model se bo pomikal nekaj časa naprej, nekaj časa nazaj, levo in desno, morda bo celo zaplesal. Pozneje bomo dodali še tipala, prek katerih bo model zaznaval okolico in se ustrezno odzval z umikom ali obvozom ovire.