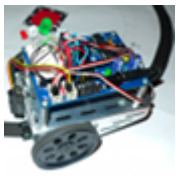
Portál pre odborné publikovanie ISSN 1338-0087

Line Follower – robot nasledujúci čiaru

Tölgyessy Michal · Elektrotechnika, Medicína

24.10.2012

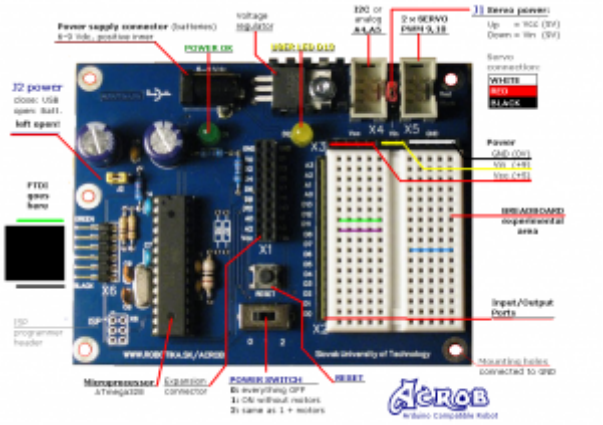


Skúsený robotik sa pri pohľade na nadpis článku iste spýta: „Prečo Line Follower? Tých už predsa existuje obrovské množstvo a každý vie, o čo ide!“ Nuž, dovolím si oponovať. Amatérska robotika je čím ďalej, tým viac dostupná aj mladšej generácii, ktorá sa s danou témou nemusela dosiaľ stretnúť. Článok je skôr odrazovým mostíkom pre tých, ktorí nevedia ako začať a predstavuje pomerne jednoduchú a finančne nenáročnú konfiguráciu. Navyše čitateľa zoznámi s robotickou platformou Acrob, ktorá umožňuje realizovať aj iné aplikácie v oblasti mobilnej robotiky.

Platforma

Platformu Acrob (Arduino Compatible Robot) navrhol a zrealizoval Ing. Richard Balogh na Fakulte elektrotechniky a informatiky, STU a používa sa aj na výučbu mobilnej robotiky. Ako už prezrádza samotný názov, platforma je plne kompatibilná so svetoznámou značkou Arduino [www.arduino.cc]. Programátor teda môže používať voľne stiahnuteľný vývojový softvér, dostupné knižnice podrobne zdokumentované na webovej stránke Arduino a programovať v jednoduchom high-level jazyku založenom na C/C++. Doska plošných spojov je na obr. 1. Je osadená 8-bitovým mikrokontrolérom ATmega328 od výrobcu AVR.

Najdôležitejšími súčiastkami pre popisovanú aplikáciu sú 6-pinový konektor v ľavej časti na pripojenie USB kábla, ktorým sa do robota nahráva program, konektor X5, kam sa pripájajú servo motory a 3-stavový prepínač. Ten môže byť v polohe 0 – robot je vypnutý, 1 – robot je zapnutý, ale motory sú odstavené, 2 – robot je zapnutý vrátane motorov. V pravej dolnej časti je biele kontaktné pole, na ktoré je možné umiestniť súčiastky v požadovanom zapojení bez potreby letovania. Takto sa dá robot použiť na rôzne aplikácie len jednoduchou výmenou súčiastok. Pozdĺž poľa je konektor s digitálnymi a analógovými vstupmi/výstupmi mikropočítača, aby bolo zapájanie praktické a jednoduché. Nad poľom sú konektory napájania Vcc a GND.



Obr. 1

Ako sledovať čiaru?

Aké súčiastky teda potrebujeme? Chceme, aby náš robot bol schopný jazdy, preto ho vybavíme podvozkom s dvomi pogumovanými kolieskami a servo motormi značky Parallax. Ďalej má byť schopný nasledovať čiaru. Človeku stačia oči, robot môže svet pozorovať cez kameru, alebo rôzne typy snímačov. Keďže spracovávať údaje z kamery je pomerne ťažké a náš projekt má byť čo najmenej finančne náročný, použijeme tri infračervené snímače QTI 27401. Pracujú na princípe vyslania infračerveného IR lúča a spracovania jeho odrazu. Ak je odrazený signál silný, senzor sa nenachádza nad čiarou. V opačnom prípade, teda ak je odraz slabý sa senzor nad čiarou nachádza. Je dôležité zdôrazniť, že pre optimálne správanie robota musí byť čiara čo najtmavšia (najlepšie čierna) a okolie čiary čo najjasnejšie (najlepšie biele).

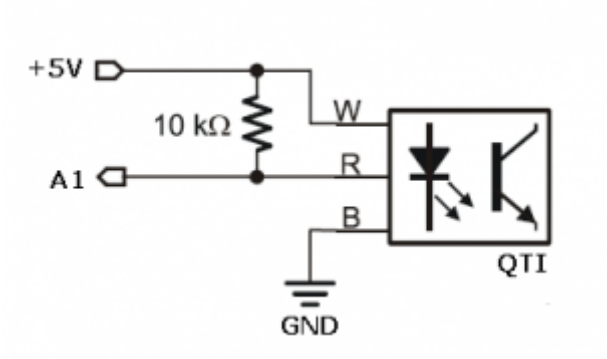
Okrem snímačov na pozorovanie čiary je vhodné zapojiť aj snímač prekážok, ktoré sa môžu vyskytnúť pred robotom. Na tento účel použijeme infračervený snímač SHARP GP2Y0A21. Robot s takýmito súčiastkami je schopný pohybu a sledovania čiary, neposkytuje nám však žiadne informácie o tom, ako sa mu darí. Najjednoduchším spôsobom komunikácie je použitie niekoľkých farebných LED diód. Robota môžeme naprogramovať napríklad tak, aby blikal červenou diódou, keď narazí na prekážku, zelenou diódou, keď má problém so sledovaním čiary a podobne.

Aby bol takýto celok schopný jazdy, potrebuje zdroj energie. Na základné experimenty je možné použiť napájanie z USB portu počítača, takýto prístup má však nevýhody. Robot je schopný jazdiť len na vzdialenosť rovnú dĺžke kábla a tiež mu môže kábel brániť v manévrovaní. Preto je podvozok vybavený štyrmi alkalickými AA batériami s napätím 1,5 V. Keďže sú batérie zapojené do série, poskytujú dostatočné napätie na napájanie riadiacej elektroniky ako aj servo motorov robota. Nezachádzajme však do prílišných fyzikálnych detailov a pozrime sa ako súčiastky zapojiť.

Zapojenie súčiastok

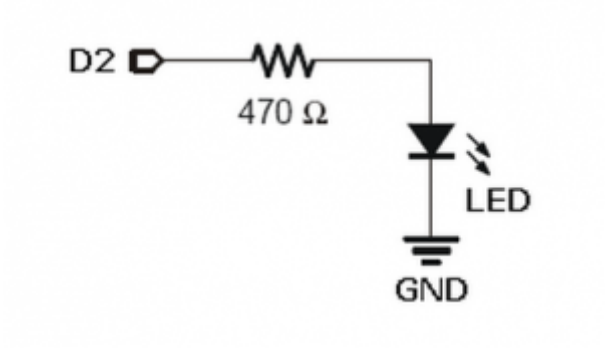
Na zapojenie servo motorov slúži konektor X5. Nachádzajú sa tu dve trojice pinov – zem (GND), napájanie pre motory (Vcc) a výstupný PWM signál na riadenie rýchlosti motora. Zapojenie snímača čiary je na obr. 2. Treba k nemu priviesť napájanie a zem z konektora pri kontaktnom poli. Ako vidno z obrázka, výstupný signál je krátkym káblikom prepojený s analógovým vstupom A1. Na tomto vstupe sa budú v programe čítať hodnoty a zisťovať, či sa daný snímač nad čiarou nachádza alebo nie.

Aby snímač pracoval správne, je potrebné zapojiť aj odpor podľa schémy. Na sledovanie čiary budeme používať 3 snímače. Všetky zapojíme analogicky podľa uvedeného návodu, pre každý výstup však musíme použiť iný analógový vstup mikropočítača. Výsledkom teda budú tri rôzne vstupy, podľa ktorých budeme v každom čase vedieť určiť polohu robota vzhľadom na čiaru. Presné detaily budú popísané neskôr v časti o stavovom automate sledovania čiary.



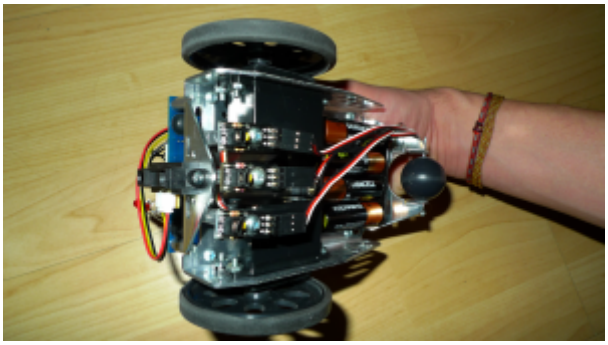
Obr. 2

Snímač vzdialenosti na detegovanie prekážok zapojíme do konektora X4 podobným spôsobom ako servo motory. Poslednými súčiastkami sú LED diódy. Každú diódu pripojíme na samostatný digitálny výstup podľa schémy na obr. 3. Nesmieme zabudnúť na odpor, aby sme diódu používaním nezničili.



Obr. 3

Výsledok zapojenia robota v popisovanej konfigurácii je na obr. 4 a 5. Vidíme podvozok s dvomi kolieskami primontovanými k servo motorom a jedným oporným kolieskom pre udržanie stability. Vpredu sú 3 snímače čiary, nad nimi je umiestnený snímač prekážok. Batérie sú upevnené zdola na podvozku. Na vrchu je primontovaná doska plošných spojov s privedeným napájaním z batérií a kontaktným poľom, ktoré je osadené tromi snímačmi a tromi LED diódami rôznych farieb.



Obr. 4

Obr. 5

Programovanie

Hardvér robota máme zapojený, musíme mu ešte vdýchnuť život – naprogramovať ho. Jeho základné schopnosti je možné zhrnúť do týchto bodov:

- Robot musí byť schopný pohybu (dopredu, doprava, doľava, rotácia, zastavenie)
- Robot musí vedieť spracovávať informácie z okolia pomocou svojich snímačov
- Robot musí vedieť signalizovať pomocou LED diód

Pomocou týchto základných úkonov budeme vedieť zrealizovať aj zložitejšie úlohy ako sledovanie čiary či rozpoznávanie prekážok. Najskôr si však ukážeme ako naprogramovať základy. Všetky konfigurácie sa realizujú vo funkcii setup() a samotný program je umiestnený vo funkcii loop(). Môžeme si vytvárať aj vlastné funkcie, aby obsah loop() nebol príliš zložitý. Pre viac detailov a návodov je dobré navštíviť oficiálnu stránku platformy Arduino. Na pohyb použijeme knižnicu Servo.h, deklarujeme si objekty pre pravý a ľavý servo motor a vytvoríme si pomocné funkcie reprezentujúce danú formu pohybu (rovno, doprava, rotácia...).

```
Servo LeftServo;
Servo RightServo;
void go_straight()
{
  LeftServo.write(140);
  RightServo.write(40);
}
void go_right()
{
  LeftServo.write(96);
  RightServo.write(86);
}

void go_sharp_right()
{
 LeftServo.write(140);
 RightServo.write(91);
}

void rotate()
{
```

```
  LeftServo.write(140);
  RightServo.write(140);
}
```

Každý typ pohybu je definovaný hodnotou, ktorá sa priradí objektu pravého a ľavého servo motora. Vhodné hodnoty je treba zistiť experimentálne, čiže metódou pokus-omyl. Závisia od použitých motorov, kolies a povrchu, na ktorom je robot testovaný. Číslo 90 zodpovedá stojacemu motoru. Zvyšovaním rastie rýchlosť motora jedným smerom, uberaním rastie rýchlosť druhým smerom. Možno sa pýtate, prečo máme dve funkcie na pohyb doprava. Správna otázka. Rozdiel medzi funkciami je v tom, že go_right() zabezpečí jemný pohyb doprava a go_sharp_right() prudký pohyb doprava. Ak čiara, ktorú robot sleduje, zmení náhle prudko smer je vhodné použiť druhú z funkcií. Podobne si definujeme aj funkcie na pohyb doľava a zastavenie robota (vtedy je hodnota na každom motore 90).

Pred uvedením robota do chodu musíme motory ešte nakonfigurovať vo funkcii setup() – čiže priradiť ich zodpovedajúcim pinom mikrokontroléra.

```
void setup()
{
  LeftServo.attach(9);
  RightServo.attach(10);
}
```

Prácu so snímačmi si ukážeme na prípade kalibrácie. Čo to tá kalibrácia je? IR snímače sa správajú v meniacich sa svetelných podmienkach vždy trochu ináč, preto je vhodné zabezpečiť, aby robot vedel s istotou povedať, čo je čierna a čo biela farba. Keď snímače kalibrujeme, postavíme robota tak, aby ľavý a pravý snímač boli nad bielou plochou a stredný snímač nad čiernou čiarou. Následne načítame analógový výstup všetkých troch snímačov cez funkciu analogRead(). Spravíme priemer hodnôt získaných z pravého a ľavého snímača a zapíšeme ho do jednej premennej, obsah druhej premennej bude hodnota zo stredného snímača.

Výsledkom teda budú dve hodnoty načítané v premenných calibWhite a calibBlack. Nakoniec zavedieme prahovú premennú threshold. Bude obsahovať prah – hranicu medzi čiernou a bielou. Akákoľvek načítaná analógová hodnota sa vždy porovná s hodnotou v threshold. Ak bude väčšia, je snímač nad čiarou a považujeme výsledok za logickú 1. Ak bude menšia, je snímač mimo čiary a výsledok je logická 0. Vďaka kalibrácii teda vieme načítané analógové údaje získané počas jazdy robota zodpovedne zjednodušiť na čiernu (1) a bielu (0). Vieme teda robota riadiť. Uvedený zdrojový kód pridáme do existujúcej funkcie setup().

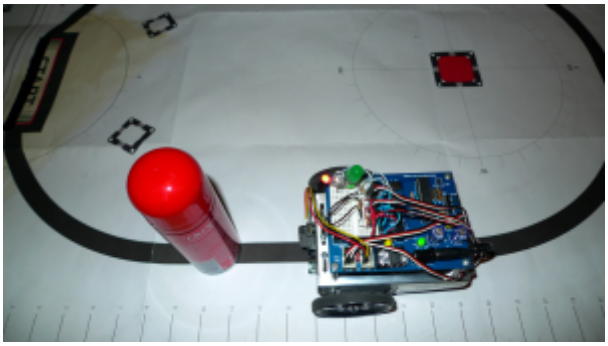
```
#define SENSOR_R 1
#define SENSOR_M 2
#define SENSOR_L 3
void setup()
{
  pinMode(SENSOR_M, INPUT);
  // nastavenie pinov ako vstupnych
```

```
  pinMode(SENSOR_R, INPUT);
  pinMode(SENSOR_L, INPUT);
  calibWhite = analogRead(SENSOR_L);
  calibWhite += analogRead(SENSOR_R);
  calibWhite = calibWhite/2;
  calibBlack = analogRead(SENSOR_M);
  threshold = (calibBlack - calibWhite)/2 + calibWhite;
}
```

Infračervený snímač na meranie vzdialenosti od prekážky naprogramujeme veľmi jednoducho. Z charakteristiky v dokumentácii od výrobcu si pozrieme, aké výstupné napätie zodpovedá vzdialenosti, ktorú chceme dosiahnuť. Napríklad, ak má robot zastať 10 cm pred prekážkou, pozrieme si, aké napätie dáva snímač na svoj výstup pri 10 cm. V prípade nami použitého SHARP GP2Y0A21 je to približne 2,25 V. V programe periodicky načítavame hodnotu cez analogRead() a keď sa načítaná hodnota priblíži napätiu 2,25 V, dáme robotovi príkaz zastaviť. Tak sa vyhneme kolízii s prekážkou.

Posledná hardvérová súčasť robota, ktorú potrebujeme naprogramovať sú LED diódy. Lepšou formou signalizácie ako svietiaca dióda je blikajúca dióda. Ukážeme si preto, ako naprogramovať robota v situácii, keď sa mu do cesty dostane prekážka. Situáciu nám pomôžu pochopiť nasledovné kroky:

1. Robot ide po čiare bez prekážok
2. Predný IR snímač zaznamená prekážku
3. Robot zastaví a začne blikať červenou diódou
4. Ak je prekážka odstránená robot prestane blikať diódou a pokračuje v nasledovaní čiary



Obr. 6

Ako prvé musíme naprogramovať funkciu, ktorá zabezpečí samotné blikanie. Bude to funkcia s argumentom udávajúcim celkové trvanie blikania v milisekundách. Tento čas rozdelíme na menšie úseky, v ktorých sa striedajú dva stavy – zapni LED a vypni LED. Aby bolo ľudské oko schopné pozorovať blikanie, musí byť čas zapnutia a rovnako aj vypnutia dostatočne dlhý, napríklad 75 milisekúnd.

```
#define LED_Red 3
void led_delay(int del)
{
  int i;
     for(i  =  0;  i  <  del;  i  =  i+150)      {
      digitalWrite(LED_Red, HIGH);       delay(75);
```

```
    digitalWrite(LED_Red, LOW);     delay(75);   } }
```

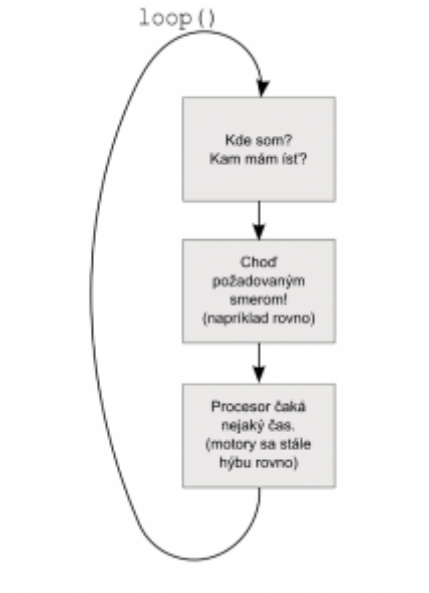
Nesmieme zabudnúť na nastavenie všetkých pinov, na ktoré sú zapojené LED diódy, ako výstupných vo funkcii setup(). Časť zdrojového kódu starajúca sa o detegovanie prekážky a blikanie diódy v rámci slučky loop() môže vyzerať nasledovane (DIS_SENSOR je pin, na ktorom je pripojený IR snímač prekážok).

```
void loop()
{
  distance = analogRead(DIS_SENSOR);
  if(distance > 350) //350 predstavuje cca. 1,7 V (asi 15
cm)
  {
    stop_vehicle();
    led_delay(300);
  }
// … dalsi kod riadenia robota
}
```

Nasledovanie čiary

V tomto okamihu je náš robot osadený hardvérom, schopný pohybu, detekcie prekážky, detekcie čiary a vie nám signalizovať blikaním jednej či viacerých LED diód. Posledné, čo ho musíme naučiť je skĺbiť všetky tieto vlastnosti do jedného celku tak, aby bol schopný kontinuálne nasledovať čiaru.

Ako prvé musíme pochopiť základný princíp pohybu v nekonečnej slučke loop(). Keďže robot je digitálne riadený systém, musí svoj pohyb rozdeliť na pravidelné časové úseky a v každom z nich sa rozhodnúť ako sa bude ďalej pohybovať. Mikropočítač však stále vykonáva nejaké inštrukcie, čiže na začiatku každého úseku zadáme motorom príkaz ako sa majú pohybovať a potom prikážeme procesoru aby nejaký čas čakal (čiže vykonával tzv. NOP inštrukcie, ktoré nemajú na robota reálny vplyv). Počas tohto čakania sa motory stále hýbu. Keď čakanie skončí začína ďalší úsek. Poskladaním týchto tzv. diskrétnych úsekov vznikne plynulý pohyb. Tento princíp je ilustrovaný na obr. 7.



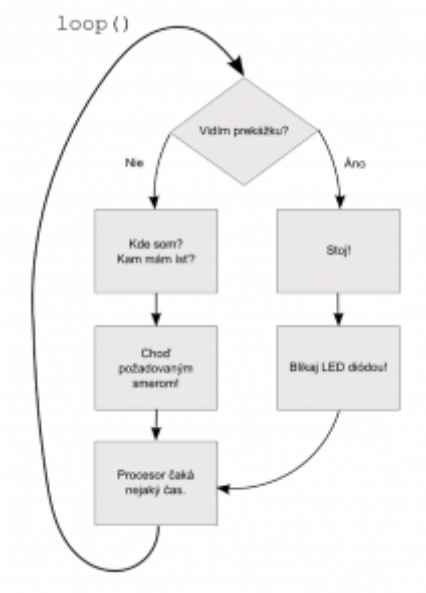
Obr. 7

Pochopili sme, ako funguje pohyb robota. Teraz je potrebné naprogramovať rozhodovací algoritmus. Robot sa rozhoduje na základe informácií z troch snímačov čiary. Každý z nich dáva výstup rovný 1 alebo 0. Vždy, keď načítame dáta máme trojicu ľavý = 1/0, stredný = 1/0, pravý = 1/0, zjednodušene [1/0, 1/0, 1/0]. Čiže, keď je stredný snímač nad čiarou a dva bočné mimo čiary, dostaneme trojicu [0, 1, 0]. Ak je čiara naľavo od robota, dostaneme trojicu [1, 0, 0]. Pre prvý prípad prikážeme robotovi, aby šiel rovno, pretože sa nachádza priamo na čiare. Pre druhý prípad musí ísť doľava, aby sa vrátil na čiaru. Takýmto spôsobom vytvoríme tabuľku (tzv. stavový automat) všetkých prípadov, ktoré môžu nastať. Nakoniec priradíme každému stavu konkrétnu funkciu pohybu robota, ktorú sme už naprogramovali. Výsledok je v nasledujúcej tabuľke.

Tab. 1

Údaj zo snímačov [ľavý, stredný, pravý]	**Funkcia ďalšieho pohybu**
010	go_straight()
011	go_right()
001	go_sharp_right()
110	go_left()
100	go_sharp_left()

Okrem stavov definovaných v tabuľke, môže dôjsť aj k nedefinovaným stavom, napr. [1,0,1]. Spôsobené môžu byť zlými svetelnými podmienkami, alebo povrchom nespĺňajúcim predpokladané kritériá. Každý programátor ich môže ošetriť podľa vlastného uváženia. Výsledný graf ilustrujúci funkcionalitu robota je na obr. 8. Takto naprogramovaný robot dokáže nasledovať čiaru, zastaviť pred prekážkou a signalizovať prítomnosť prekážky blikaním LED diódy.



Obr. 8

Záver

Naprogramovali sme robota, ktorý spĺňa základné požiadavky pre nasledovanie čiary. Rozhodne je však čo zlepšovať a inovovať, ak máme prostriedky a kreativitu. Pridaním viacerých snímačov čiary by bolo možné dosiahnuť vyššiu presnosť a schopnosť rozpoznávať križovatky. Dalo by sa experimentovať s algoritmom obchádzania prekážok. V našej konfigurácii je to trochu problém, pretože robot nemá snímače na bokoch. Museli by sme naprogramovať pevnú predpokladanú veľkosť prekážky a tiež by algoritmus nemusel fungovať stopercentne, pretože na rôznych povrchoch sa kolesá správajú rôzne a môžu prešmykovať.

Pridaním snímačov na boky by však bol robot schopný obísť prekážku ľubovoľnej veľkosti. Aby bol konkurencieschopný na robotickej súťaži, kde je dôležitá rýchlosť, bolo by vhodné vymeniť motory za výkonnejšie. Predstavená robotická platforma Acrob je vhodná najmä na výukové a experimentálne účely. Kontaktné pole umožňuje zapojenie rôznych súčiastok, napríklad kontaktných kovových fúzov na dotykovú detekciu prekážok so zvukovou signalizáciou. O tom však už možno niekedy nabudúce.