

Programovatelný robot za pár korun

Jonáš Rosecký¹

e-mail: rosecky.jonas@centrum.cz

¹ Student Fakulty Informatiky MU, Brno

Klíčová slova

Arduino, robot, programování, IoT, eBay

1 Úvod

Všem účastníkům letošního ročníku konference Počítač ve škole věnovali pořadatelé jako dárek robota ###. Jedná se o jednorúčkový stroj pro jízdu po čáře. Proč ale nevyužít schopností tohoto zařízení pro vytvoření víceúčelového programového robota?

2 Než začneme s vylepšením

Ještě, než se pustíme do samotného zapojování a programování, je potřeba porozumět tomu, jak vlastně váš robot funguje.

Jádrem celého robota je mikročip LM393. Vůbec nevádí, že vám to nic neříká. Naštěstí žijeme v době internetu a dá se celkem snadno zjistit, že se jedná o 2 porovnávače napětí:

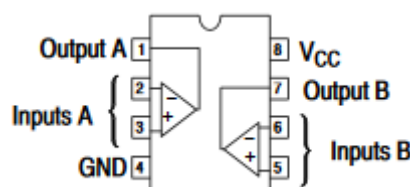


Schéma mikročipu LM393

Díky schématu robota lze odhalit, že vstupy A i B (*Inputs A* a *Inputs B*) jsou tytéž vodiče, pouze prohozené a můžeme z nich číst hodnotu z fotorezistorů umístěných vpředu na robotovi.

S pomocí Výstupů (*Output A* a *Output B*) lze naopak ovládat robotovy motory.

3 Čím mám začít?

První věc, která vás čeká je výběr programovatelného mikrokontroleru, který budete používat. Neděste se! Zní to hrozně, ale zase tak složité to není. Potřebujete takový mikrokontroler, který dokáže ovládat minimálně 2 digitální výstupy (ideálně PWM – umožňuje plynulou regulaci rychlosti) a ideálně alespoň 2 analogové vstupy, pokud chcete používat fotorezistory, které už na robotovi jsou.

Já jsem se rozhodl pro platformu *Arduino*. Jednak kvůli její nízké ceně (při objednávce z neoficiálních obchodů – například eBay – se dá sehnat kolem 1 \$), také pro její velkou komunitní podporu a v neposlední řadě kvůli tomu, že už s ním mám nějaké zkušenosti. S návodem tedy budu pokračovat pro zařízení *Arduino Pro Mini*, které umožňuje napájení z 3.3V zdroje (2 AA baterie v sérii mají jen 3 V, to ale Arduino zvládne). Pro jiné platformy však budou kroky velmi podobné. Další zajímavý mikročip je ESP8266, který sice nemá analogové vstupy, zato umožňuje připojení k WiFi a komunikaci s internetem.

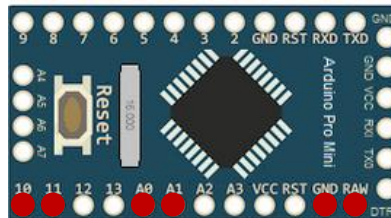
4 Nikdy v životě jsem ale nic nezapojoval

Nebude to nic složitého. Prvním krokem je odpojení původního čipu. Rukou, nebo opatrně šroubovákem zkuste odpojit mikročip z jeho soketu. Odpojený mikročip si dobře schovejte, pokud se rozhodnete vrátit k původním schopnostem, stačí odpojit Arduino a vrátit čip na původní místo.

K propojení Arduina s robotem nám stačí jakékoli vodiče, které se vejdou do slotů po původním čipu (například drátky na nepájené pole).

Robota potřebujeme připojit k Arduino následujícím způsobem (na levé straně je vždy číslo pinu původního čipu, napravo označení pinu na Arduino):

- 4 (GND) → GND (uzemnění)
- 8 (VCC = napájení) → RAW (neregulované napětí 3,3 V až 12 V)
- 2 (Input A1) → A0 (Analogový vstup)
- 3 (Input A2) → A1 (Analogový vstup)
- 1 (Output A) → 10 (Digitální pin s PWM)
- 7 (Output B) → 11 (Digitální pin s PWM)



Nákres mikrokontroleru Arduino Pro Mini s označením použitých portů

5 Kouzlo programování

Nyní už s můžeme robota „naučit“ dělat co chceme. Platforma Arduino má velký potenciál v možnosti rozšíření stávajících součástek o nové bez nutnosti hlubokých znalostí hardwaru (například senzor vzdálenosti, barevná světla, piezzo, Bluetooth, ...). Návody a knihovny pro jednotlivé součástky lze nalézt na internetu.

Níže nabízím jen malou ukázkou „lepší“ jízdy po čáře, která bere v potaz i možnost jízdy rovně. Tento kód můžete použít jako předlohu, ale nebojte se experimentovat. Nebojte se, určitě to zvládnete, držím vám palce. A když ne, tak stačí čip vrátit zpět...

```
// ----- Nastavení -----
//   Piny senzoru
int leftSensorPin = A0;
int rightSensorPin = A1;

//   Piny motoru
int leftMotorPin = 10;
int rightMotorPin = 11;

//   Maximalni rozdíl pro stejnou barvu
int maxDifference = 50;

//   Rychlost pomalejšího motoru v rohu (0-255)
int cornerSpeed = 100;
// ----- Nastavení -----

void motor(int left, int right) {
  // Motor se zapína uzemněním pinu
  digitalWrite(leftMotorPin, 255 - left);
  digitalWrite(rightMotorPin, 255 - right);
}

int leftColor() {
  // Čím tmavší, tím vyšší
  return analogRead(leftSensorPin);
}

int rightColor() {
  return analogRead(rightSensorPin);
}
```

```

void setup() {
  // Nastavit piny s motory, jako vystup
  pinMode(leftMotorPin, OUTPUT);
  pinMode(rightMotorPin, OUTPUT);

  // Na zacatku vypnout motory
  digitalWrite(leftMotorPin, HIGH);
  digitalWrite(rightMotorPin, HIGH);

  // Nastaveni serioveho vystupu (pro kalibraci)
  Serial.begin(9600);
}

void loop() {
  if (leftColor() + maxDifference < rightColor()) {
    // Levy senzor nad bilou, pravy nad cernou
    motor(255, cornerSpeed);

  } else if (rightColor() + maxDifference < leftColor()) {
    // Pravy senzor nad bilou, levy nad cernou
    motor(cornerSpeed, 255);

  } else {
    // Jinak
    motor(255, 255);
  }

  // Hodnoty senzoru muzeme sledovat v Serial Plotteru
  Serial.print(leftColor());
  Serial.print("\t");
  Serial.println(rightColor());
  delay(100);
}

```