

# Dokumentace robota: Sledovač čáry řízený smartphonem

Damián Kořalka

## 1 Úvod

Cílem tohoto projektu bylo sestavit autonomního robota na platformě stavebnice *bit-Beam*, který využívá výpočetní výkon běžného smartphonu. Na rozdíl od mikrokontrolérů (Arduino) je zde řízení realizováno pomocí jazyka Python běžícího na telefonu. Telefon zpracovává obraz z kamery pro detekci čáry a ovládá motory generováním audio signálu přes 3,5mm Jack konektor.

## 2 Konstrukce a Hardware

Mechanická konstrukce vychází z modelu *Robot pro experty (bB4\_3)*. Šasi je tvořeno plastovými nosníky a tištěnými díly, které nesou držák pro smartphone.

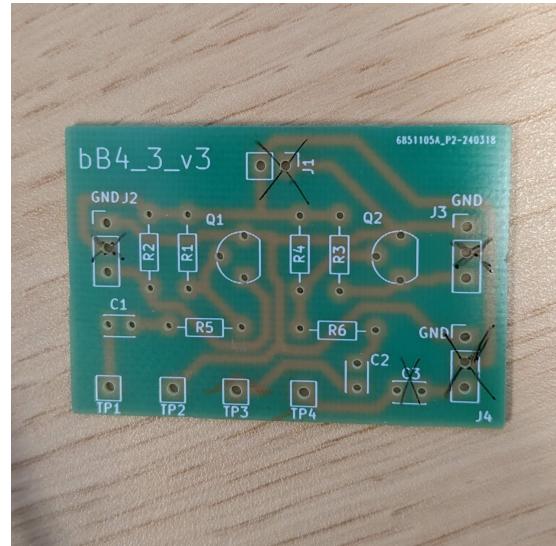
### 2.1 Princip ovládání přes Audio Jack

Robot využívá unikátní způsob komunikace a napájení/řízení motorů:

- **Vstup:** Kamera smartphonu snímá vodicí čáru.
- **Zpracování:** Aplikace v Pythonu (využívající knihovnu OpenCV) vypočítá odchylku od středu čáry.
- **Výstup:** Telefon generuje stereo zvukový signál. Levý kanál ovládá levý motor, pravý kanál pravý motor.
- **Elektronika:** Speciální plošný spoj připojený k audio výstupu telefonu usměrňuje a zesiluje signál pro DC motory.



(a) Konstrukční návrh z manuálu



(b) Detail řídicího plošného spoje

Obrázek 1: Hardware robota

### 3 Software a Algoritmy

Program je napsán v jazyce Python a spouští se v prostředí na Androidu (např. Pydroid). Využívá knihovny cv2 (OpenCV) pro zpracování obrazu, numpy pro matematické operace a pygame pro generování zvuku.

#### 3.1 Zpracování obrazu

Algoritmus snímá obraz, ořezává jej na oblast zájmu (ROI) a hledá nejtmavší bod (černá čára) v porovnání s nejsvětlejším (pozadí). Na základě polohy čáry se počítá regulační odchylka.

#### 3.2 Generování signálu pro motory

Funkce `motory(mL, mR)` převádí požadovanou rychlosť (0–100) na zvukový signál. Využívá se principu, kdy amplituda nebo střída zvukové vlny o frekvenci 500 Hz odpovídá výkonu motoru.

```

1 # Nastaveni konstant
2 frekvence = 500
3 sample_rate = 40000
4
5 # Funkce pro ovladani motoru pres audio vystup
6 def motory(mL, mR):
7     global signalLR, mL0, mR0
8     # Omezeni maximalnich a minimalnich hodnot
9     mL = max(min(mL, 95), 5)
10    mR = max(min(mR, 95), 5)
11
12    # Generovani signalu pouze pri zmene rychlosti
13    if (mL != mL0) or (mR != mR0):
14        dutyL = int(mL)
15        dutyR = int(mR)

```

```

16     # Uprava signaloveho pole (generovani PWM pomocí zvuku)
17     signalLR[0 : int(sample_rate/frekvence) : 2] = 30000
18     # ... (kod pro vyplneni bufferu) ...
19     sound = pygame.mixer.Sound(signalLR.tobytes())
20     sound.play(-1)
21
22 # Hlavni smycka zpracovani obrazu
23 # Vypocet chyby (rozdil mezi stredem a carou)
24 if maximum - nejmensi < 600000:
25     # Ztrata cary - pokracuj v predchozim smeru
26     if m1_last > m2_last:
27         motory(40, 5)
28     else:
29         motory(5, 40)
30 else:
31     # PID regulator
32     m1, m2 = pid(error, p=1, i=0, d=0, default_speed=40, max_speed=55)
33     motory(m1, m2)

```

Listing 1: Ukázka generování signálu a PID regulace

## 4 Seznam použitých komponent

Níže uvedená tabulka shrnuje klíčové součástky použité při stavbě dle přiloženého návodu.

| Kategorie   | Komponenta   |
|-------------|--|
| Řízení      | Smartphone s OS Android (s 3.5mm Jack)   |
| Software    | Pydroid  |
| Elektronika | Plošný spoj (Audio to Motor Driver)<br>Kabely Jack 3.5mm (Male-Male)   |
| Pohon       | 2x DC Motor s převodovkou (žlutý TT)<br>2x Kolo s gumovou pneumatikou  |
| Konstrukce  | Díly bitBeam (nosníky 11x13, L-profily)<br>Spojovací materiál (šrouby M4, matice)<br>Gumička pro uchycení telefonu |

Tabulka 1: Soupis materiálu

## 5 Závěr

Robot úspěšně demonstruje využití moderního smartphonu v robotice. Použití audio výstupu pro řízení motorů je netradiční, ale funkční řešení, které eliminuje potřebu drahých mikrokontrolérů a umožňuje využít vysoký výpočetní výkon telefonu pro pokročilé zpracování obrazu.