

Dokumentace projektu: Řízení modelů pomocí mikrokontroleru do IMP 2017/2018

Jméno a příjmení: Tomáš Aubrecht

Login: xaubre02

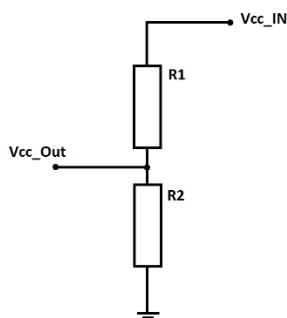
Úvod

Úkolem bylo vytvořit model, který má disponovat alespoň třemi různými funkcemi či možnostmi pohybu. Zvolil jsem možnost vytvoření robota, který je ovládán pomocí Raspberry Pi 3, ke kterému se uživatel připojuje pomocí služby SSH. Samotný robot je složen ze stavebnice LEGO, dvou 6V DC motorů, 9V zdroje napětí pro pohon motorů, ultrazvukového senzoru HY-SRF05, Arduino modulu obsahující h-můstek L298N pro řízení motorů a již zmíněného Raspberry Pi 3 s power bankou pro jeho napájení. Implementace softwaru je realizována v jazyce Python 3.

Hardwarová realizace

Ze stavebnice LEGO byl vytvořen podvozek, který tvoří nosnou konstrukci celého robota. Zde jsou umístěny dva 6V DC motory pro pohon levé a pravé strany robota. Pro převod mezi ozubenými kolečky motorů a hřídelí z LEGO stavebnice byly z dentacrylu odlity dodatečné díly.

Na přední straně podvozku je připevněn ultrazvukový senzor. Ten obsahuje 5 pinů, ze kterých jsou použity 4 a to pro napájení(Vcc), uzemnění(Ground), pro vyslání signálů(Trigger) a pro příjem signálů(Echo). Napětí na Echo je stejné jako vstupní napětí, což je 5V, ale Raspberry GPIO piny podporují maximální napětí 3.3V, proto je potřeba toto napětí snížit. Za tímto účelem jsou do obvodu přidány 2 rezistory, kde R1 má odpor 4.7kΩ a R2 má 10kΩ.



$$V_{out} = V_{in} \times \frac{R2}{R1 + R2}$$

$$\frac{V_{out}}{V_{in}} = \frac{R2}{R1 + R2}$$

Pro změření vzdálenosti je nutné na Trigger vyslat puls, který trvá 0.1ms. Jakmile jej senzor přijme, nastaví Echo do logické jedničky a vyšle 8 ultrazvukových dávek o frekvenci 40kHz do prostoru před sebou. Jakmile se některá z těch dávek vrátí zpět, nastaví se Echo opět do logické nuly. Díky tomu můžeme změřit dobu, jak dlouho trvalo ultrazvuku dorazit k překážce a zpět a následně vypočítat vzdálenost. Ta se vypočítá jako *čas * rychlost*. Rychlost šíření zvuku ve vzduchu za normálních podmínek je 343m/s, což je 34300 cm/s. Rovnici je potřeba vydělit dvěma, protože se jedná o dobu cesty ultrazvuku k překážce a zpět, kde ve výsledku získáme vzorec pro výpočet vzdálenosti: *čas * 17150*.

Na podvozku je dále umístěna power banka, na které jsou umístěny baterie, Raspberry a Arduino modul. Ten obsahuje konektor pro vstupní napětí, které bude poskytnuto DC motorům z výše zmíněných baterií, 5V výstupní napětí, které je použito pro napájení ultrazvukového senzoru, výstupy pro připojení dvou DC motorů a 6 pinů pro jejich řízení. Dva piny(Out) slouží pro nastavování napětí a uzemnění na vstupech DC motoru a tím k nastavování směru otáčení a jeden pin(Enable) pro povolení těchto pinů. To samé platí pro druhý motor. Rychlost otáček je řízena pomocí pulzně šířkové modulace na Enable pinech.

Softwarová realizace

Za účelem zpracování vstupních argumentů programu a jejich následnou syntaktickou kontrolu byla implementována třída `Arguments` a pro uchování zadaných možností třída `Options`. Tyto třídy jsou obsaženy v modulu `args.py`.

Modul `parts.py` obsahuje třídy pro ovládání jednotlivých prvků obvodu, jako jsou motory, h-můstek a senzor pro měření vzdálenosti. Každá třída je inicializována čísly pinů, ke kterým jsou dané části připojeny.

Modul `robot.py` řídí samotný běh programu a obsahuje funkce reprezentující jednotlivé běhové módy robota. Po zpracování argumentů inicializuje ultrazvukový senzor a h-můstek na příslušných pinech a spustí požadovaný mód robota.

Program běží po celou dobu, dokud jej uživatel sám neukončí pomocí klávesové zkratky `Ctrl+D` nebo pomocí klávesy `ESC` v případě manuálního řízení robota. V tu chvíli se uvolní veškerá paměť a vrátí se nastavení GPIO pinů na výchozí hodnoty.

Spouštění

Program se musí spustit jako privilegovaný uživatel pro přístup k ovládání GPIO pinů na Raspberry. Parametry pro spuštění jsou

- `--help` pro výpis nápovědy,
- `-r N` pro udržování robota minimálně N centimetrů od překážky,
- `-k N M` pro udržování robota minimálně N cm a maximálně M cm od překážky a
- `-c` pro manuální ovládání robota.

Příklad spuštění: `sudo python3 robot.py -k 20 50`

Manuální ovládání robota

Klávesa	Činnost
W	pohyb vpřed
S	pohyb vzad
A	zatočení vlevo
D	zatočení vpravo
+	zvýšení rychlosti o 10%
-	snížení rychlosti o 10%
Mezerník	zastavení motorů
ESC	ukončí program