



Vyšší odborná škola  
a Střední průmyslová škola elektrotechnická,  
Plzeň, Koterovská 85

## ROČNÍKOVÁ PRÁCE S OBHAJOBOU

Téma:            Roverbert

Autor práce:    Matouš Hep

Třída:            3.L

Vedoucí práce:    Jiří Švihla

Dne:              30.4.2024

Hodnocení:



**Vyšší odborná škola a  
Střední průmyslová škola elektrotechnická Plzeň,  
Koterovská 85**

<b>ZADÁNÍ ROČNÍKOVÉ PRÁCE</b>	
Školní rok	2023/2024
Studijní obor	78-42-M/01 Technické lyceum
Jméno a příjmení	Matouš Hep
Třída	3.L
Předmět	Kybernetika
Hodnoceno v předmětu	Kybernetika
Téma	Roverbert
Obsah práce	<ol style="list-style-type: none"><li>1. Návrh softwaru pro prevenci kolize</li><li>2. Implementace antikolizních senzorů do karoserie</li><li>3. Implementace mechaniky pohonu</li><li>4. Návrh a 3D tisk karoserie</li></ol>
Zadávací učitel Příjmení, jméno	Švihla Jiří
Podpis zadávajícího učitele	
Termín odevzdání	30. dubna 2024

# Anotace

Ročníková práce se zabývá řešením problematiky navádění vozítka podle čáry. První část práce nabízí možnosti zpracování vstupu ze senzoru detekujícího překážky. Dále práce obsahuje návrh softwaru řídícího pohonnou jednotku s ohledem na regulaci rychlosti jízdy podle zakřivení vodicí čáry. Poslední část práce je věnována designu karoserie s ohledem na implementaci mechaniky zatáčení, senzorů, baterie, motorů a jednočipového počítače. Finálním výstupem práce je vozidlo schopné následovat vodicí čáru, které je schopno zastavit v případě vyskytnutí se překážky v cestě.

„Já Matouš Hep prohlašuji, že jsem ročníkovou práci na téma Roverbert zpracoval sám se spolužákem Martinem Robbem za konzultace vedoucího práce Jiřího Švihly. Veškeré prameny a zdroje informací, které jsme použili k sepsání této práce, byly citovány a jsou uvedeny v seznamu použitých pramenů a literatury.“

V Plzni dne:

Podpis:

# Obsah

<b>1</b>	<b>Automatická prevence kolize</b>	<b>6</b>
1.1	Princip . . . . .	6
1.2	Zapojení . . . . .	6
1.3	Program . . . . .	8
<b>2</b>	<b>Pohon</b>	<b>10</b>
<b>3</b>	<b>Design karoserie</b>	<b>10</b>

# Úvod

## Motivace

Sledování čáry nachází v současné době značné využití ve skladování, logistice a na autonomních výrobních linkách například jako navigace pro skladovací roboty. Cílem práce je vytvoření vozidla s obdobným navigačním systémem.

## Řešení

Navigaci po čáře zajišťuje řada infračervených (dále jen IR) senzorů. Senzor přijímá odražené IR záření, které se liší na základě barvy odrazové plochy, což umožňuje odlišení kontrastních barev a převedení na logickou jedničku a nulu. Detekci překážky před vozidlem zajišťuje ultrazvukový senzor, z něhož lze zjistit vzdálenost od překážky na základě rozdílu mezi přijetím a vysláním signálu.

Výstupy z ultrazvukového a IR senzorů jsou zpracovávány v jednočipovém počítači Raspberry Pi Pico, který podle příchozího vstupního signálu řídí zatáčení vozidla a rychlost jízdy.

# 1 Automatická prevence kolize

## 1.1 Princip

K automatické prevenci kolize využívám v ročníkové práci ultrazvukového senzoru pro měření vzdálenosti HC-SR04 (viz Obrázek 1).



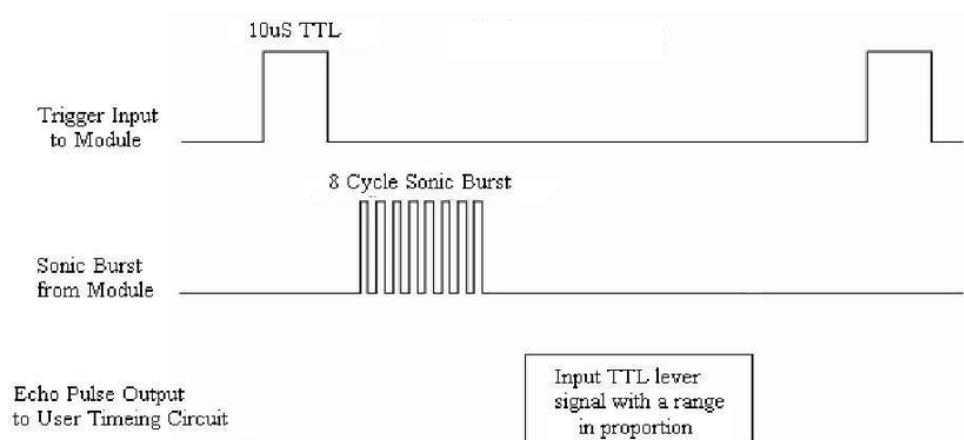
Obrázek 1: Ultrazvukový senzor

Ultrazvukový senzor při přijetí řídicího signálu ve formě napěťového pulzu o minimální délce  $10\text{ }\mu\text{s}$  vyšle 8 cyklů ultrazvuku o frekvenci  $40\text{ kHz}$  (viz Obrázek 2). V případě, že se vyskytuje ve vzdálenosti odpovídající měřicímu rozsahu  $2\text{--}400\text{ cm}$  překážka, vyslaný ultrazvukový signál se od překážky odrazí zpět, kde je zachycen senzorem. Při opětovném přijetí signálu senzor vyšle z výstupního pinu napěťový pulz, jehož délka odpovídá času mezi vysláním a přijetím ultrazvukového signálu. Z délky tohoto pulzu a rychlosti šíření zvuku lze posléze dopočítat vzdálenost překážky. Rychlost šíření zvuku ve vzduchu závisí na teplotě a vlhkosti vzduchu, což jsem při výpočtu vzdálenosti překážky zanedbal a počítám s konstantní rychlostí zvuku  $340\text{ m/s}$ .

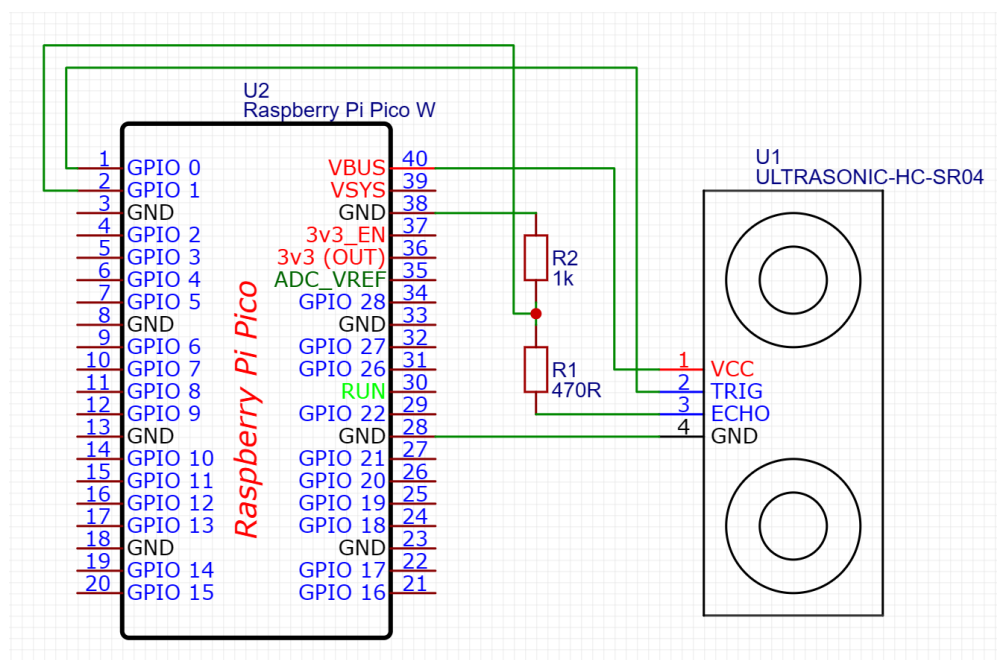
## 1.2 Zapojení

Ultrazvukový senzor HC-SR04 má čtyři piny (viz Obrázek 3): 5V napájení (VCC), vstupní pin řídicího signálu (TRIG), výstupní pin (ECHO) a připojení k zemi (GND).

Ultrazvukový senzor je určen pro 5V logiku. Jedno-čip Raspberry Pi Pico funguje na 3,3V logice. V případě vstupního pinu TRIG rozdíl v napěťové logice nehraje roli, jelikož senzor vyšle ultrazvukový signál i při přijetí napěťového pulzu s napětovou hladinou 3,3 V.



Obrázek 2: Časový diagram ultrazvukového senzoru



Obrázek 3: Schéma zapojení ultrazvukového senzoru

To už ale neplatí pro výstup z ultrazvukového senzoru. Výstupní signál o hladině 5 V by mohl poškodit Raspberry Pi Pico. Proto je zapotřebí přijímané napětí snížit na požadovanou hodnotu zapojením napěťového děliče (viz Obrázek 3). Podílem napětí zjistíme poměr potřebných rezistorů.

$$\frac{U_1}{U_2} = \frac{3,3}{5} \approx \frac{2}{3} \quad (1)$$

Z toho vyplývá že potřebné rezistory musí být přibližně v poměru 1:2. Dále musí mít dostatečně velkou hodnotu natolik, aby nebylo překročeno maximální zatížení výstupu ultrazvukového senzoru a zároveň musí mít dostatečně malou hodnotu, aby byly zanedbatelné oproti vstupnímu odporu Raspberry Pi Pico. Maximální proud výstupního pinu ultrazvukového senzoru je 20 mA. V rámci bezpečnosti jsem použil pro výpočet hodnotu přibližně o 85 % menší tedy 3 mA.

$$R = \frac{U}{I} = \frac{5}{0,003} \approx 1667 \, \Omega \quad (2)$$

Potřebný odpor tedy musí mít přibližně hodnotu 1667  $\Omega$ , což si díky veliké rezervě můžu dovolit zaokrouhlit na 1500  $\Omega$ . Hodnota odporu 1500  $\Omega$  je také o tři řády nižší než vstupní odpor Raspberry Pi, jenž je v jednotkách M $\Omega$ , takže vstupní odpor Raspberry Pi zásadně neovlivní funkčnost napěťového děliče. Hodnotu 1500  $\Omega$  rozdělím v požadovaném poměru 1:2.

$$R_1 = 1500 \cdot \frac{1}{3} = 500 \quad (3)$$

$$R_2 = 1500 \cdot \frac{2}{3} = 1000 \quad (4)$$

Jelikož hodnota 500  $\Omega$  se nevyskytuje v normalizovaných odporových řadách, použil jsem nejbližší normalizovanou hodnotu 470  $\Omega$ .

## 1.3 Program

Ultrazvukový senzor vrací napěťový pulz o délce odpovídající časovému rozdílu mezi odesláním a přijetím ultrazvukového signálu (viz. Kapitola 1.1). K určení vzdálenosti překážky je potřeba délku tohoto pulzu změřit.

K tomuto ve své ročníkové práci používám funkci `time_pulse_us` knihovny `machine` pro python. Funkce `time_pulse_us` má tři argumenty: `pin`, `pulse_level` a `timeout_us`. Argument `pin` vyžaduje adresu pinu, na kterém má být měřena doba pulzu. Argument `pulse_level` má dvě možné hodnoty logickou nulu nebo jedničku. V Případě, že je v argumentu logická nula, funkce měří dobu po kterou je na pinu logická nula. V opačném případě funkce vrací čas, po který je na pinu lo-



gická jedna. Argument `timeout_us` určuje maximální dobu trvání pulzu a je ze základu nastavený na 1 000 000  $\mu\text{s}$  tedy na jednu sekundu. V případě, že je tato doba překročena v době mezi pulzy, funkce vrátí hodnotu -2. Pokud je samotný měřený signál delší než hodnota `timeout_us`, funkce vrátí hodnotu -1.

Aby senzor vyslal ultrazvukový signál, musí přijmout pulz o minimální délce 10  $\mu\text{s}$ .

2 Pohon

3 Design karoserie