

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA STROJNÍ



BAKALÁŘSKÁ PRÁCE

**MODERNIZACE LABORATORNÍ ÚLOHY: MĚŘENÍ ZRYCHLENÍ PŘI
NÁRAZU**

**MODERNIZATION OF LABORATORY TASK: ACCELERATION
MEASUREMENT AT IMPACT**

Autor: Stanislav Linhart

Vedoucí: Ing. Cyril Oswald, Ph.D.

2019

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Linhart** Jméno: **Stanislav** Osobní číslo: **466567**
Fakulta/ústav: **Fakulta strojní**
Zadávající katedra/ústav: **Ústav přístrojové a řídicí techniky**
Studijní program: **Teoretický základ strojího inženýrství**
Studijní obor: **bez oboru**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Modernizace laboratorní úlohy Měření zrychlení při nárazu

Název bakalářské práce anglicky:

Modernization of Laboratory Task: Acceleration Measurement at Impact

Pokyny pro vypracování:

- 1) Prostudujte současnou konstrukci modelu, navrhnete vhodná vylepšení a tyto poté realizujte.
- 2) Proveďte rešerši cenově dostupné instrumentace pro regulaci rychlosti modelu auta.
- 3) Vybrané řešení realizujte včetně naprogramování a ověření funkčnosti.

Seznam doporučené literatury:

- HOFREITER, Milan. Základy automatického řízení, Praha: ČVUT, 2016.
- RIPKA, Pavel. Senzory a převodníky. Praha: ČVUT, 2009.
- Podklady pro laboratorní úlohy měření zrychlení a otáček. Online: <https://moodle-vyuka.cvut.cz/mod/folder/view.php?id=25474> dostupné 28. 3. 2019
- Dokumentace měřicí karty LabJack. Online: <https://labjack.com/products/u3> dostupné 28. 3. 2019

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Cyril Oswald, Ph.D., U12110.3

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **26.04.2019**

Termín odevzdání bakalářské práce: **12.06.2019**

Platnost zadání bakalářské práce:



Ing. Cyril Oswald, Ph.D.
podpis vedoucí(ho) práce



podpis vedoucí(ho) ústavu/katedry



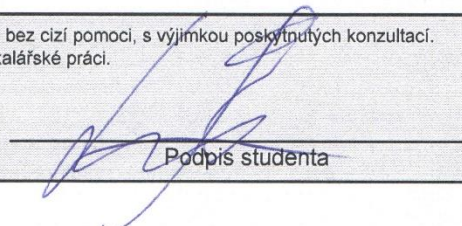
prof. Ing. Michael Valášek, DrSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

26-04-2019

Datum převzetí zadání



Podpis studenta

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a že jsem uvedl v příloženém seznamu veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací, vydaným ČVUT v Praze 1. 7. 2009.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu § 60 Zákona č.121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 12.08.2019

Stanislav Linhart

Poděkování

Chtěl bych poděkovat vedoucímu práce Ing. Cyrilovi Oswaldovi, Ph.D. za skvělé vedení a rady při tvorbě této práce.

Dále děkuji Ing. Daně Baureové za pomoc při hledání zadání práce po předchozí nezdařené volbě.

Děkuji mé rodině, která za mnou stojí a podporuje mě při studiu. Zejména pak Robertu Svitáčovi a Janu Ryplovi.

Anotační list

Autor: Stanislav Linhart
Název práce: Modernizace laboratorní úlohy Měření zrychlení při nárazu
Rok: 2019
Ústav: Ústav přístrojové a řídicí techniky
Vedoucí práce: Ing. Cyril Oswald, Ph.D.
Rozsah práce: 54 stran, 37 obrázků, 4 tabulky, 9 rovnic, 2 přílohy, CD
Klíčová slova: akcelerometr, měření, model, Arduino UNO, Raspberry Pi, LabJack, regulace, PID

Abstrakt

Tato práce se zabývá modernizací laboratorní úlohy, měřené v předmětu Technická měření na bakalářských studijních programech Fakulty strojní ČVUT. Měřicí úloha je zaměřena na práci studentů s osciloskopem a na čtení signálů z akcelometru. V práci se autor zabývá úpravou konstrukce modelu, který je u laboratorní úlohy používán. Dále doplňuje model o nové funkce, a to regulaci rychlosti modelu a ovládáním modelu prostřednictvím bezdrátových technologií. Pro regulaci je použit mikrokontroler Arduino UNO, který je dálkově ovládán pomocí technologie Bluetooth. Model se řídí dotykovou obrazovkou s počítačem Raspberry Pi.

Abstract

This thesis deals with the modernization of the laboratory exercise, concerning the subject "Technical Measurements" at the bachelor's degree programs of the Faculty of Mechanical Engineering, CTU. The measurement task is focused on the work of students employing an oscilloscope and on reading the signals from the accelerometer. The author deals with the modification of the model construction, which is used in the laboratory exercise. It also adds new features to the model, including model speed control and wireless model control. The Arduino UNO microcontroller, which is remotely controlled by Bluetooth technology, is used for regulation. The model follows a touch screen with a Raspberry Pi computer.

Obsah

1	Úvod	8
2	Elektronika pro řízení modelu	10
2.1	Raspberry Pi	10
2.2	LabJack	12
2.3	Arduino	13
3	Senzory neelektrických veličin	15
3.1	Optoelektronický snímač otáček	15
3.2	Induktační snímač otáček.....	16
3.3	Měření rychlosti pomocí tachodynamu.....	16
3.4	Akcelerometry	17
4	Princip a řízení stejnosměrného stroje	19
4.1	Stejnoseměrný stroj	19
4.2	Princip stejnosměrného stroje.....	20
4.3	Stroje s cizím a paralelním buzením.....	21
4.4	Řízení otáček stejnosměrného stroje v motorickém režimu	23
4.5	H-můstek.....	24
5	Algoritmy řízení	25
5.1	Logické řízení	25
5.2	Analogové řízení.....	26
6	Úprava konstrukce modelu	30
6.1	Konstrukce původního modelu	30
6.2	Tvorba nového modelu	32
7	Návrhy variant hardwarové instrumentace.....	37
7.1	Návrh číslo 1	38
7.2	Návrh číslo 2.....	39
7.3	Návrh číslo 3.....	40

8	Volba varianty	42
9	Regulace otáček pohonu pomocí zvolené varianty.....	43
9.1	Zapojení.....	43
9.2	Snímání otáček deskou Arduino.....	44
9.3	Regulace PID	45
9.4	Snímání ujeté vzdálenosti	46
9.5	Průběh měření.....	47
9.6	Ovládání modelu pomocí externího systému.....	47
10	Závěr.....	49
	Seznam použitých zdrojů	51
	Seznam obrázků	53
	Obsah přiloženého CD.....	54
	PŘÍLOHA I	I
	PŘÍLOHA II	II

1 Úvod

Tato bakalářská práce se věnuje modernizaci laboratorního modelu automobilu, který vznikl na základě laboratorní úlohy pro měření zrychlení na akcelerometru. Na modelu byly pomocí osciloskopu měřeny jednotlivé signály průběhů zrychlení na všech osách akcelerometru. Požadavkem ze strany garanta úlohy je zachování měření výstupních signálů z akcelerometru na laboratorním osciloskopu. Dalším požadavkem bylo přidání nepájivého pole na plochu modelu pro případné budoucí úpravy.

- Cílem práce je provést rešerši hardwarové instrumentace, která bude vhodná a cenově dostupná pro řízení a regulaci rychlosti modelu.
- Dále prostudování původní konstrukce modelu, návrh jeho zdokonalení a následná aplikace navržených změn.
- Výsledným cílem je naprogramování a ověření funkčnosti vybraného hardwarového řešení.

V této bakalářské práci se věnuji úpravě modelu, jehož hlavním problémem je pohyb. Model není schopen zachovávat přímočarý pohyb. Mezi osou kol a stěnou vlastního kola dochází při pohybu ke tření, které způsobuje problém s jejich rotací. Další vadou je nemožnost regulovat napětí ozubeného řemenu, který přenáší kroutící moment na zadní kola.

První část práce se zabývá elektronikou pro regulaci pohybu modelu. Dále senzory pro snímání otáček a teorií řízení otáček stejnosměrného stroje.

V další části je samotná úprava konstrukce, kterou chci zajistit hladký pohyb modelu. Pro úpravu konstrukce bych chtěl použít, co nejvíce současných prvků nebo při modelování nových součástí vycházet ze současné podoby. Nekladu si za cíl vytvořit zcela nový model, ale pouze současný upravit ke zlepšení pohyblivosti.

V další kapitole se práce věnuje návrhu možných řešení hardwarové instrumentace, z nichž jsem následně vybral nejvhodnější kombinaci hardwaru. Cílem kapitoly je vybrat k řízení modelu, takovou kombinaci, aby model nebyl

příliš náchylný k otřesům při nárazech, nebyla příliš zvýšena hmotnost modelu a byla zajištěna cenová dostupnost.

V závěru práce je popis fungování regulace otáček stejnosměrného motoru. Dále naprogramování ovládacího panelu a jeho komunikace s řídicí elektronikou modelu.

2 Elektronika pro řízení modelu

2.1 Raspberry Pi

Raspberry Pi je počítač s velmi malými rozměry, který vyvíjí vzdělávací nadace Raspberry Pi Foundation. Zařízení bylo navrženo za účelem přiblížit lidem všech věkových kategorií práci na počítači a naučit je programovat. Díky nízké ceně je velice dostupné a práce s ním jednoduchá. [2]



Obrázek 1: Raspberry Pi 3 B+ [3]

Deska je dostatečně výkonná pro běžnou práci s počítačem jako procházení internetu, práce s dokumenty kancelářských balíčků, a dokonce přehrávání videa s vysokým rozlišením. Kmitočet čtyř jádrového procesoru Cortex-A53 s architekturou ARMv8 přednastavený pro tuto verzi je 1400 MHz. Nejnovější verze disponuje operační pamětí 1 GB.

Deska na obrázku umožňuje připojení k počítačovému monitoru nebo televizoru pomocí standardu HDMI, nebo umožňuje připojení k dotykové obrazovce, která se prodává jako příslušenství, pomocí konektoru MIPI přímo na desce.

Obsahuje zvukovou kartu analogovým stereo výstupem s konektorem jack 3.5, ale pomocí I2S rozhraní umožňuje přenášet i digitální signál. [2]

Připojení k síti umožňuje 300 Mbps ethernet port a bezdrátový modul s podporou standardů 802.11 a/b/g/n/ac (2.4 a 5 GHz). Součástí verze 3 B+ je i modul Bluetooth 4.2 BLE.

Součástí jsou čtyři USB 2.0 konektory z toho jeden je vnitřně propojený s Ethernet portem.

Pro Raspberry Pi 3 B+ je doporučeno napájení 5 V DC 2,5 A. Deska oproti starším verzím výrazně zvýšila výkon a také i spotřebu energie. Jmenovitý výkon při maximálním zatížení se pohybuje kolem 7 W.

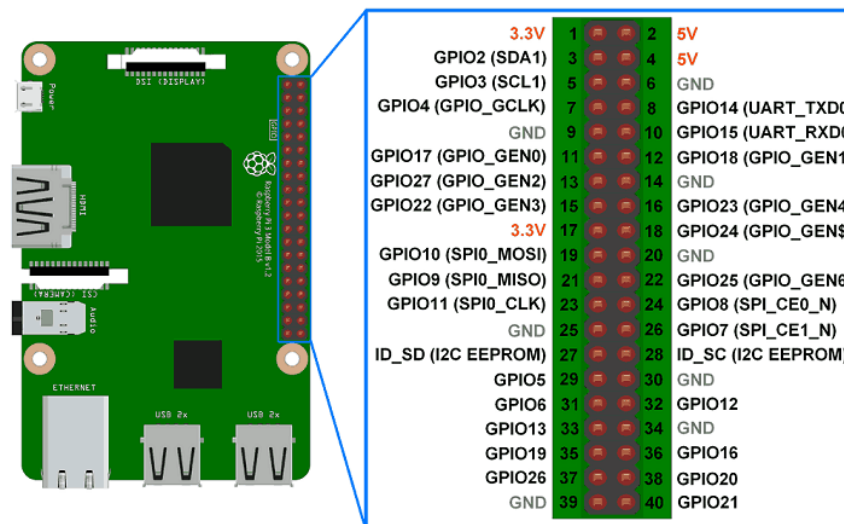


Obrázek 2 PoE HAT modul [4]

Novinkou na této desce je konektor PoE (Power over Ethernet), na který se napojí přídatný HAT modul, který vidíte na obrázku 2 a umožňuje tím napájet desku přes ethernetovou síť. To je výhodné například pro projekty domácí automatizace, když nemáme přístup k jinému zdroji napětí.

Oproti desce Arduino je pro Raspberry Pi potřeba operační systém. K dispozici jsou různé Open Source operační systémy, které jsou založeny na jádru Kernel. Například desktopový operační systém Raspbian, přímo vydávaný nadací, který je úpravou známého a dlouho vyvíjeného Debianu. Jsou k dispozici i různé distribuce, jako například LibreElec a OSMC pro multimédia, nebo Volumio pro hudební server. Dokonce i systém Windows IoT, který nabízí alternativu vůči Kernelu. Operační systém musí být nabořován z SD karty, ale testuje se i možnost bootovat pomocí USB.[5]

To, co rozlišuje Raspberry Pi od běžného počítače je sada GPIO vstupů a výstupů. Umožňují pojit elektrosoučástky a vytvářet tak různé projekty, nebo na ně lze připojit hotové moduly, například zvukové karty, displeje a reléové systémy, které tvoří s Raspberry Pi programovatelný automat pro řízení v průmyslu nebo domácí automatizaci.



Obrázek 3 Schéma GPIO konektorů na Raspberry Pi 3 [6]

Jak je vyobrazeno na obrázku 3 je zde čtyřicet konektorů. Jsou zde dva napájecí konektory s napětím 5 V DC a dva s napětím 3.3 V DC. Dále osm zemních konektorů označených GND a zbytek tvoří konektory GPIO (general-purpose input/output – univerzální vstup/výstup). Ty se při programování nastaví jako vstup, výstup, nebo jako jejich další využití, které je na obrázku napsáno v závorkách. Jsou jimi například sběrnice I²C.

2.2 LabJack

LabJack je zařízení hodící se k laboratorním účelům, které dokáže díky jednoduché konektivě rozšířit osobní počítač o digitální a analogové vstupy a výstupy. Hodí se tak pro použití na měření elektrických veličin v laboratorních experimentech ale i na prototypch.



Obrázek 4 Deska LabJack U3-LV [7]

Disponuje několika druhy pinů, z nichž část je dostupná pomocí šroubovatelných terminálů a část je v konektoru DB15. Ten vidíte v pravé části obrázku 4. [7]

Deska disponuje piny označenými *FIO*. Jsou to piny *Flexible I/O* neboli univerzální piny. Dají se totiž nastavit jako digitální vstup, digitální výstup a analogový vstup. Rozlišení analogových vstupů je 12 bit. Některé z těchto pinů se také dají nastavit jako časovače. Počet a umístění těchto pinů je závislý na konkrétním modelu desky. Dále obsahuje výstupní analogové piny s rozlišením 10 bit. [7]

2.3 Arduino

První deska Arduino byla představena v roce 2005, aby pomohla studentům designu, kteří neměli žádné předchozí zkušenosti s programováním elektroniky nebo mikro kontrolérů, vytvářet pracovní prototypy spojující fyzický svět s digitálním světem. Od té doby se stala nejoblíbenějším elektronickým prototypovým nástrojem, který používají inženýři, a dokonce i velké společnosti. [8]

Oproti výše popsanému Raspberry Pi se Arduino liší. V případě desky Arduino probíhá programování a kompilování skriptů na jiném zařízení. Je tedy potřeba počítač, na kterém se vytvoří skript, následně se přeloží a pomocí USB se nahraje do paměti procesoru. Nespornou výhodou této desky je nízká spotřeba

energie oproti Raspberry Pi, ale hodí se pro projekty, u kterých nevyžadujeme grafické uživatelské prostředí. Je také asi čtyřikrát levnější než Raspberry Pi.



Obrázek 5 Arduino UNO Rev 3 [9]

Nejprodávánější model je Arduino UNO s mikrokontrolerem ATmega328P, který disponuje pamětí 32 KB, SRAM pamětí 2 KB a EEPROM 1 KB. Mikrokontroler pracuje s frekvencí 16 MHz, což je výrazně nižší rychlost než Raspberry Pi. [9]

Počet vstupů a výstupů se liší mezi jednotlivými modely, Arduino ale nabízí oproti Raspberry i Analogové vstupy.

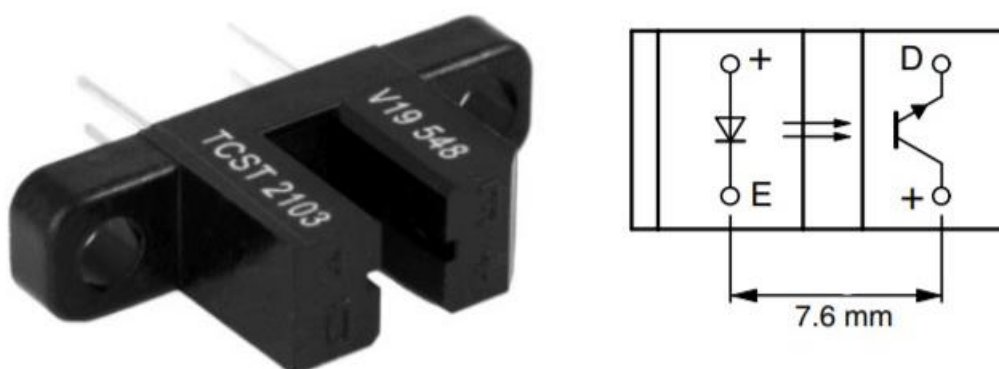
Arduino dokáže také být připojeno pomocí Bluetooth, WiFi, Ethernetu, a dokonce sítě GSM, ale pouze s pomocí modulů, které je možné dokoupit. Tyto moduly, ale je nutné připojit ke konektorům na desce a tím se obsadí velký počet vstupů a výstupů. Tyto sítě ale Arduino nedokáže využít jako běžný počítač k odesílání velkoobjemových dat, ale slouží mu například pro přijímání vzdálených příkazů. Arduino se dá i výborně použít jako vzdálená periferie pro Raspberry Pi. Příkladem může být domácí automatizace. V uživatelském prostředí si uživatel vyžádá zavření garážových vrat. Na motor garážových vrat a bezpečnostní senzory je připojeno Arduino s Wifi modemem. Raspberry vyšle příkaz pro zavření garážových vrat, Arduino vyhodnotí vstupy ze senzorů a spustí se motor. [11]

3 Senzory neelektrických veličin

K snímání neelektrických veličin, jakou je například rychlost se využívá několik fyzikálních principů. Tyto principy mohou být mechanické, elektromagnetické, optoelektronické, magnetické a kapacitní. Pro použití na modelu zvažují použití následujících tří snímačů.

3.1 Optoelektronický snímač otáček

Tento způsob snímání otáček je velice jednoduchý a často používaný. Pracuje na principu zdroje světla a fototranzistoru. Snímače se často se prodávají ve formě již hotového kompletu, který se nazývá vidlicová světelná závora, příklad vidíte na obrázku 6. Zdrojem světla je většinou LED dioda, která vyzařuje světlo v infračerveném spektru. Toto světlo prochází štěrbinou, aby vznikl úzký paprsek světla, který na druhé straně světelné závory prochází do fototranzistoru. Samotné snímání otáček probíhá poté tak, že světlo je v mezeře přerušováno kruhovou deskou, která se otáčí rychlostí, kterou chceme měřit.



Obrázek 6 Vidlicová světelná závora TCST 2103 [15]

Tím že při otáčkách je světlo přerušováno vzniká výstupní čtvercový signál. Z něj lze přechít čas mezi dvěma sousedními pulzy. Tento čas poté vydělíme počtem otvorů v kruhové desce a získáme tím otáčky za čas.

Výhodou těchto snímačů je jejich nízká pořizovací cena, která se pohybuje v řádu desítek korun. Tyto snímače nejsou vhodné do prostředí, kde by mohlo dojít znečištění a tím ucpání štěrbin. Na rozdíl od indukčního snímače.

[16]

3.2 Induktační snímač otáček

Podobným principem snímání otáček pomocí impulzů pracuje i induktační snímač. S tím rozdílem, že se při otáčení nepřerušuje světlo, ale magnetické pole. Na obrázku 7 je induktační čidlo sloužící pro zaznamenání změn magnetického indukčního toku v okolí čidla. Snímač se skládá ze dvou částí, permanentního magnetu, který tvoří magnetické pole nutné pro měření a cívky, na které se při změně magnetického indukčního toku indukuje napětí na vývodech. [16]



Obrázek 7 Indukční čidlo CS18-08P-1 M18 [17]

K narušení magnetického pole se používá kolečko z feromagnetického materiálu. Toto kolečko má na svém obvodu zuby velkých rozměrů. Tím, jak zuby míjí senzor dochází k proměnnému narušení magnetického indukčního toku a na cínce senzoru se indukuje střídavý proud. Z frekvence tohoto signálu získáme otáčky. [16]

Tento senzor je vhodný i do nečistého prostředí a z tohoto hlediska je spolehlivější než optoelektronický senzor.

3.3 Měření rychlosti pomocí tachodynamu

K tomuto způsobu měření se využívá indukovaného napětí na stejnosměrném stroji v generátorovém režimu. Tachodynamo je vlastně pouze stejnosměrný stroj, u kterého známe jeho závislost indukovaného napětí na úhlové rychlosti otáčení. Ze vztahu je patrné, že výstupní napětí je závislé na konstantě stroje, která popisuje konstrukci stroje, magnetickém indukčním toku a úhlové rychlosti.

$$V_o = C_{ss} \cdot \phi \cdot \omega \quad (1)$$

Tachodynamo má od výrobce předepsanou velikost zatěžovacího odporu. Je to proto, že při malém zatěžovacím odporu, by docházelo k úbytku napětí na vinutí a následně se to projeví změnou závislosti výstupního napětí na otáčkách. Ideálním zatěžovacím odpor je nekonečně velký. [16]

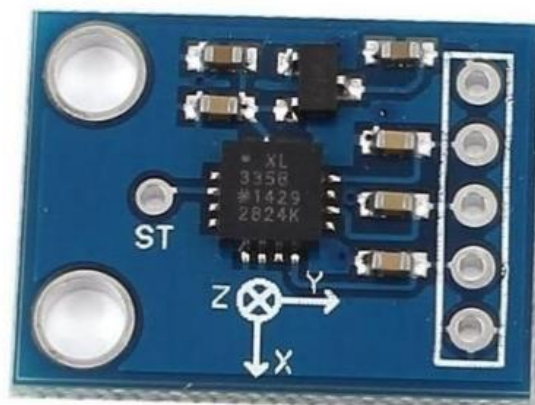


Obrázek 8 Tachodynamo K10A6 [18]

Komutátor uvnitř tachodynamu mechanicky usměrňuje indukované napětí, toto výstupní napětí je mírně zvlňené v důsledku nedokonalého usměrnění.

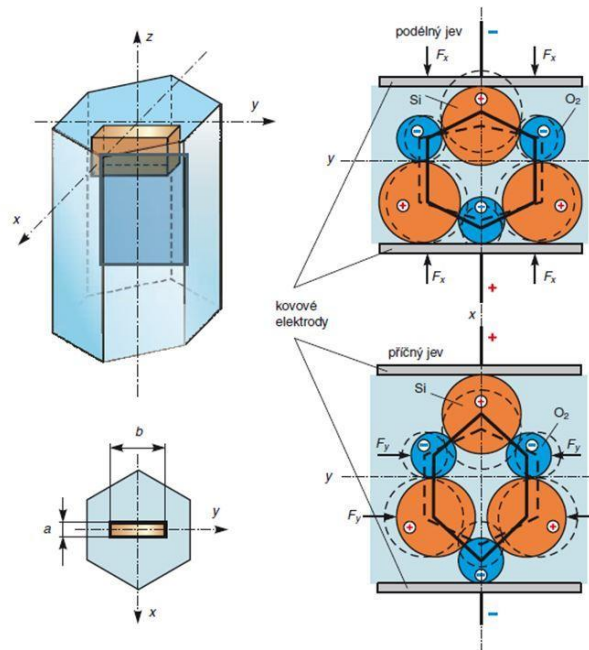
3.4 Akcelerometry

Akcelerometr je elektrická součástka, která slouží k měření statických a dynamických sil. Senzor reaguje na pohyb v měřených směrech a natočení podél jednotlivých os. Výstupem ze senzoru je analogový signál z každé osy.[12]



Obrázek 9 Modul 3-osého akcelerometru ADXL335 [13]

Pro jejich práci se používají různé principy. Některé modely využívají piezoelektrický jev, který způsobuje vznik elektrického napětí při deformaci krystalu bez středu symetrie. Deformací se ionty posunou v krystalové mřížce tak, že elektrická těžiště aniontů a kationtů se od sebe vzdálí. Posunem těžišť mezi ionty vznikne rozdíl potenciálů. [14]



Obrázek 10 Piezoelektrický jev na krystalu křemíku [14]

Dalším možným principem fungování akcelerometru je možnost použití změny kapacity. Pokud máme dvě mikrostruktury, které odděluje dielektrikum, pak změnou jejich vzájemné polohy docílíme změny kapacity.

4 Princip a řízení stejnosměrného stroje

4.1 Stejnosměrný stroj

Zařízení pro převod elektrické energie na mechanickou a naopak, které je poháněno nebo je na jeho svorkách indukován stejnosměrný proud, se nazývá stejnosměrný stroj. Pokud na jeho svorky napětí přivádíme, pracuje stroj v motorickém režimu. Naopak pokud na jeho svorkách naměříme napětí, pak se jedná o napětí indukované a stroj pracuje v generátorovém režimu. U stejnosměrných strojů v tomto režimu je běžnější užívání názvu dynamo. [19]



Obrázek 11 Stejnosměrný motor s převodovkou [20]

Skládá se ze statoru a rotoru a na rozdíl od střídavých strojů je zde budící vinutí na statoru. Tento budící obvod se buď realizuje pomocí cívek napájených stejnosměrným proudem, nebo permanentními magnety. Ať už při použití cívek nebo permanentních magnetů je nutné zajistit, aby se jejich polarita střídala. Minimální počet pólů ve statoru je dva. Můžou být doplněny o pomocné póly, kterým se také říká komutační póly. Vzdálenost mezi sousedními libovolnými póly se nazývá pólová rozteč a měří se na vnitřním obvodu statoru.[19]

V případě použití cívek jako budícího obvodu se při konstrukci statoru nemusí hledět na ztráty Joulovým teplem. Tedy používají se pro ně díly z plného materiálu nebo dokonce odlitky, na rozdíl od konstrukce rotoru, kde je potřeba vířivým proudům zamezit. Proto jsou rotory konstruovány ze vzájemně odizolovaných plechů, které jsou nalisovány na hřídel. Vinutí je

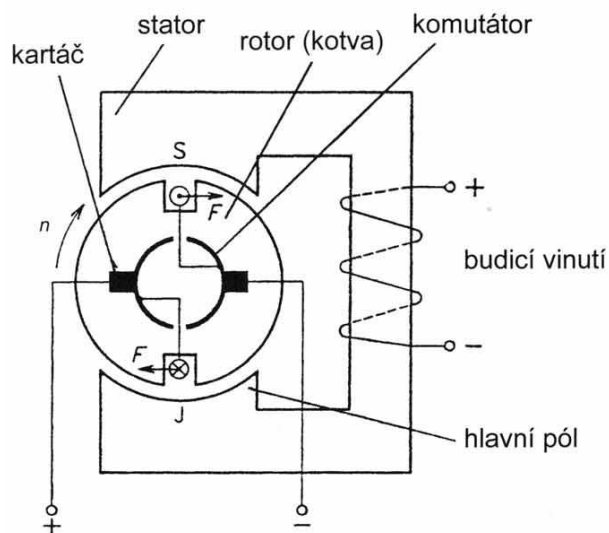
uloženo uvnitř drážek po obvodu rotoru. Podle tvaru drážek a způsobu, jak je v nich vinutí navinuté, rozdělujeme rotory na tři druhy.[19]

Prvním je rotor s pólovým vinutím. Vyznačuje se tím, že se na rotoru nacházejí tři výstupky ve tvaru „T“. Kolem tohoto tvaru jsou navinuté cívky a tvoří tak póly na rotoru. Tento způsob vinutí se používá u malých modelářských motorků a u větších motorů pro strojírenství se nepoužívá.

Nejčastějším způsobem navinutí cívek na rotor je válcové vinutí. To se používá právě pro motory konvenčních velikostí. Zde jsou cívky navinuty do drážek po obvodu rotoru a jejich počet je vyšší než u pólového vinutí. Hlavním rozdílem mezi pólovým a válcovým vinutím je, že při tomto vinutí každá cívka není navinutá právě mezi dvěma sousedními drážkami, ale vždy jich několik vynechává.

Pro úplnost je poslední způsobem vinutí prstencové. To se ale u stejnosměrných motorů v běžné praxi již nepoužívá. [19]

4.2 Princip stejnosměrného stroje



Obrázek 12: Zjednodušené schéma stejnosměrného stroje [21]

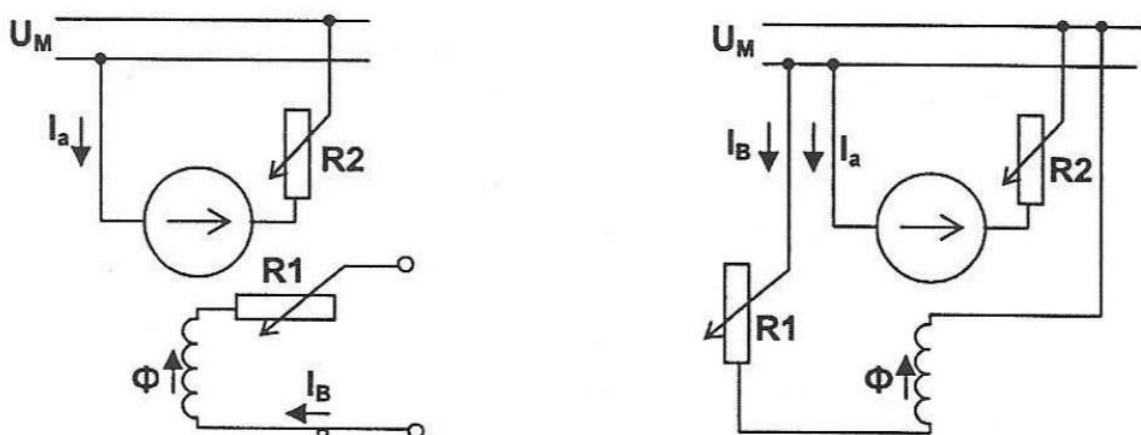
Princip stejnosměrného stroje je založen na jednoduchém jevu. Magnetické siločáry magnetického indukčního toku směřující od severního pólu budícího magnetického obvodu k jeho jižnímu pólu uzavírají tok přes rotor. Na rotoru je cívka, na které přivedený proud vytváří magnetické pole. Magnetické siločáry rotoru směřují opačným směrem oproti těm statoru. A vzniká zde moment, který se snaží o to, aby magnetické siločáry směřovali stejným směrem,

tedy aby severní pól cívky byl nasměrován k jižnímu pólu statoru a naopak. Díky zařízení zvaném komutátor, které se skládá z dvojice segmentů na válci, z nichž každý segment je připojen k jednomu kontaktu cívky, dochází k přepólování cívky na statoru a díky tomu nedojde tak nikdy k vyrovnání magnetických sil a k zastavení rotoru. U reálného motoru je použito cívek více a komutátor je rozdělen na více segmentů. [19]

Existují i motory bez komutátoru. Jsou to takzvané bezkartáčové motory. Nemusí se používat díky tomu, že stator tohoto motoru je tvořen rotorem běžného stejnosměrného stroje s pólovým vinutím a rotor je ze statoru s permanentními magnety. Tedy stator je uvnitř rotoru. Stator je tedy na pevně upevněn a vedou z něj tři vodiče. Obvykle se používá barevné značení černý, červený a žlutý. Jednotlivé cívky statoru jsou totiž zapojeny k zemi. Každý z vodičů je potom zapojen na jednu cívku. Vždy jsou aktivní dvě cívky s navzájem opačnou polaritou. Tento motor potřebuje ke svému provozu řídicí elektroniku, která postupně přepíná mezi jednotlivými cívkami a řídí jejich polaritu.[19]

Zpět ke konvenčním stejnosměrným motorům. Dále se totiž rozdělují podle způsobu napájení budícího vinutí. A to na stroje s cizím buzením, stroje s paralelním buzením a stroje se sériovým buzením.

4.3 Stroje s cizím a paralelním buzením



Obrázek 13: Schéma zapojení buzení při cizím buzení (vlevo) a paralelním buzení (vpravo) [19]

Na obrázku 13 je schéma zapojení prvních dvou typů vinutí, liší se pouze v tom, že v případě paralelního vinutí sdílí stejný zdroj napětí. Do této skupiny motorů se řadí i motory bez budícího vinutí s permanentními magnety. Vnitřní

moment stroje tohoto motoru se vypočítá vztahem, kde C_{ss} je konstanta stejnosměrného motoru, Φ je magnetický indukční tok a I_a je proud kotvě.

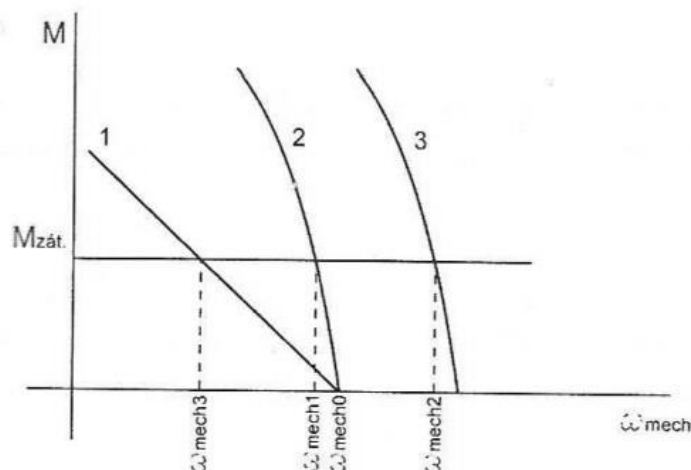
$$M_i = C_{ss} \cdot \Phi \cdot I_a \quad (2)$$

Tento vztah popisuje pouze moment teoretický, skutečný výsledný moment na výstupní hřídeli by byl po odečtení mechanických ztrát.

Po dosazení do vztahů charakterizujících stejnosměrný motor získáme úhlovou rychlost motoru.

$$\omega_{mech} = \frac{U_M - R_a \cdot I_a - \Delta U_k}{C_{ss} \cdot \Phi} \quad (3)$$

Závislost momentu motoru na otáčkách je zobrazena na obrázku 14. Nachází se na něm tři křivky označené čísly. Křivka číslo 1 charakterizuje měkkou charakteristiku, kterou motor má, pokud nesnižujeme budící napětí na statoru, ale snižujeme napětí na vinutí rotoru. Tím dosáhneme snížení úhlové rychlosti motoru. Příklad křivky číslo 2 je tvrdá charakteristika při napájení vinutí rotoru a budícího obvodu jmenovitým napětím. Tato křivka platí po rozběhu motoru, tedy po ustálení otáček. Křivka číslo 3 platí pouze pro motory s cizím a paralelním buzením nikoli pro motory s permanentními magnety. Jde totiž o případ odbuzování motoru. To je také jediný způsob, jakým můžeme zvýšit otáčky nad jmenovitou hodnotu. Stejnosměrné motory nesmí být nikdy odbuzeny.[19]

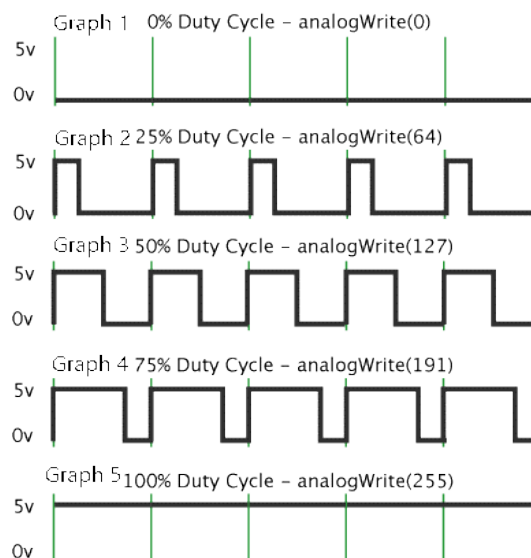


Obrázek 14: Momentová charakteristika stejnosměrného motoru s cizím a paralelním buzením [19]

4.4 Řízení otáček stejnosměrného stroje v motorickém režimu

Tento druh pulzní modulace se využívá pro řízení rychlosti otáček stejnosměrného motoru. Je to způsob, jakým můžeme získat analogový výstup z digitálního signálu.

Na obrázku 15 vidíme, jak vypadá napěťový výstup při pulzně šířkové modulaci. Pokud proložíme přímkou průběhem signálu tak, že obsah plochy ohraničené právě touto přímkou a průběhem nad ní se bude rovnat obsahu ohraničeného průběhem pod přímkou, získáme střední hodnotu napětí. Toto napětí je přivedeno přímo na regulovaný spotřebič, nebo je přivedeno na tranzistor, čímž regulujeme velikost napětí z externího zdroje vyššího napětí pro stejnosměrný motor.



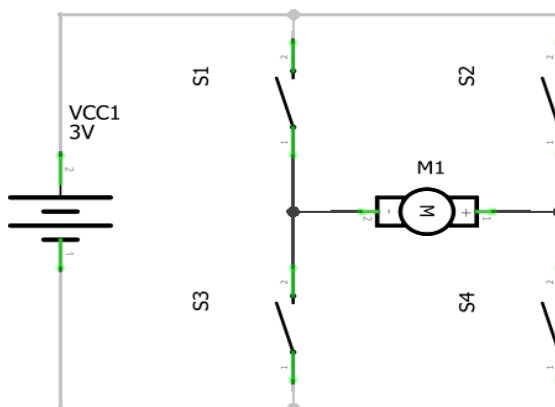
Obrázek 15 Závislost napětí na čase při PWM 0% [22]

V případě programování desky Arduino, se tento výstup vyvolá příkazem `analogWrite()`, kdy se do závorek zadává celé číslo v rozmezí 0 až 255. Pro hodnotu 0 bude výstup jako na prvním grafu. Výstupní napětí bude tedy nulové.

Pro poloviční výstupní napětí bude zadaná hodnota v příkazu 127 a výstupní napětí tedy 3,5 V. Příklad vidíte na třetím průběhu na obrázku 15. Plného výstupního napětí dosáhneme zadáním hodnoty 255. Tento výstup je naprosto stejný, jako bychom zadali příkaz `pinMode(pin, OUTPUT)`. Průběh je na posledním grafu. [22]

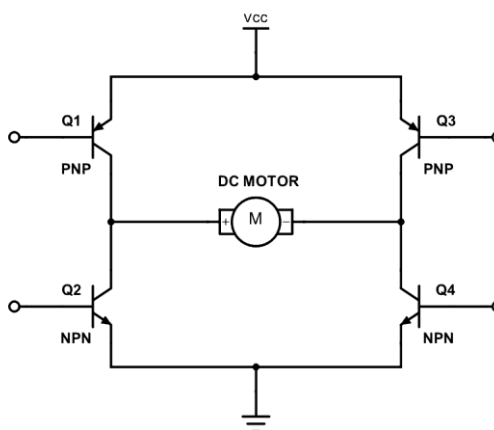
4.5 H-můstek

Obvod, kterým se mění směr otáčení motoru bez nutnosti přepojování kabelů se nazývá H-můstek. Na obrázku 16 vidíme jeho zjednodušené schéma bez možnosti regulace rychlosti. Toto uspořádání za pomoci spínačů umožňuje pouze přepólování motoru, a tedy změnu otáček. Motor se točí jedním směrem při sepnutí spínačů S1 a S4. K přepólování a tedy změně směru otáčení dojde při sepnutí spínačů S2 a S3. [23]



Obrázek 16 Schéma jednoduchého H-můstku [23]

Pokud chceme pomocí H-můstku měnit nejen směr otáčení ale regulovat s ním i rychlost motoru, který má vyšší jmenovité napětí, než je výstupní napětí z řídicí elektroniky, musíme spínače ve schématu na obrázku 17 nahradit tranzistory.



Obrázek 17 Schéma H-můstku s možností regulace výstupního napětí [24]

Vznikne nám tak schéma, které vidíte na obrázku 24. H-můstky jsou dostupné již jako integrované obvody, které jsou připraveny k použití.

5 Algoritmy řízení

5.1 Logické řízení

Nejjednodušší možností, jak něco řídit, je dvoustavové řízení. Při kterém je výstupem pouze vypnuto nebo zapnuto. Neboli binární hodnota reprezentována 0 a 1. Tento způsob řízení se nazývá logické řízení a je to základní stavební kámen tvorby řídicích systémů.

Logické funkce zpracovává logické vstupy a vrací logický výstup. Logická funkce je reprezentována kombinací tří logických operací. Jsou jimi disjunkce neboli logický součet, konjunkce neboli logický součin a negace. [1]

Disjunkce má pravdivý výstup právě tehdy, když alespoň jeden ze vstupů je pravdivý. Tedy se dá nahradit slovem „nebo“. Pravdivý výstup u konjunkce nastává právě tehdy když, všechny vstupy jsou pravdivé. Negace převrací logickou hodnotu z pravdivé na nepravdivou a naopak.

Tabulka 1 Pravdivostní tabulka [1]

<i>a</i>	<i>b</i>	<i>c</i>	<i>y</i>
0	0	0	0
0	0	1	0
0	1	1	1
0	1	0	0
1	1	0	1
1	1	1	1
1	0	1	1
1	0	0	0

Logické funkce se dělí na kombinační a sekvenční. Kombinační funkce je taková, kdy je logický výstup závislý pouze na okamžitém vstupu. Tedy nezávisí na předchozích výstupech. Tuto závislost mají sekvenční funkce. Jinak řečeno, jsou závislé na tom, zda předchozí výstupní hodnotou byla 0 nebo 1.

Tabulka 2 Karnaughova mapa [1]

		y			
		$a b$	0 0	0 1	1 1
$c d$	0 0	1	1	0	1
	0 1	0	1	1	1
	1 1	0	1	1	1
	1 0	0	1	1	1

Závislosti vstupů na výstupech se nejčastěji zapisují do pravdivostní tabulky nebo Karnaughovy mapy. Z nich se dají pomocí prvního a druhého kanonického tvaru získat algebraické tvary logických funkcí. Funkce se dále dají zjednodušovat pomocí booleovy algebry, nebo přímo z Karnaughovy mapy. [1]

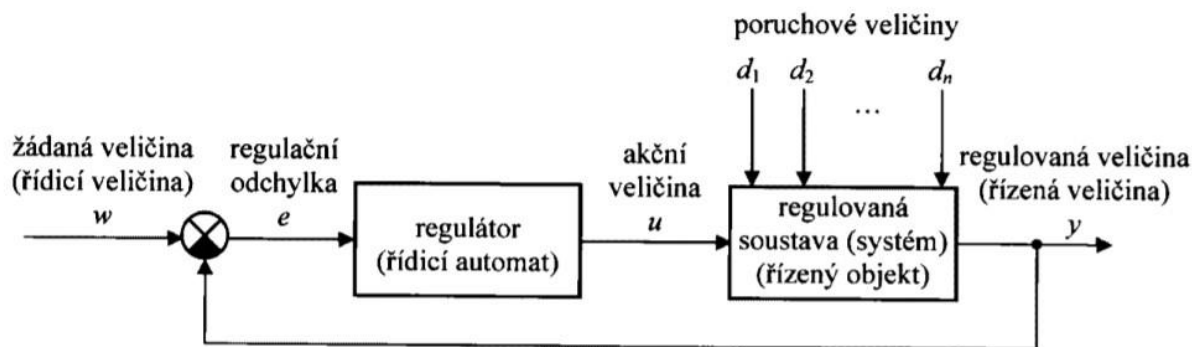
5.2 Analogové řízení

Na rozdíl od logického řízení, kde je řízení dvoustavové, je v některých aplikacích třeba použít řízení se spojitým výstupem neboli analogové řízení. Spojitý výstup zajišťuje vazbu mezi vstupem a výstupem bez skokových změn. Používá se nejčastěji pro soustavy, ve kterých chceme řízenou veličinu udržet na konkrétní hodnotě. Další případy, kdy je nutno použít analogového řízení, jsou pokud řízená soustava je náchylná na skokové změny akční veličiny a pokud je soustava citlivá na rozkmitání.[1]

Analogové spojitě řízení zajišťuje regulační obvod, který je znázorněn na obrázku 18. Do bloku regulátoru vstupuje regulační odchylka. Ta je značena písmenem $e(t)$ a je rovna rozdílu žádané veličiny a regulované veličiny, která do rozdílu přichází ze zpětné vazby.

$$e(t) = w(t) - y(t) \quad (4)$$

Výstupem regulátoru je poté akční veličina označovaná u . Je to veličina, kterou vypočítal regulátor a je jí například velikost vstupního napětí do řízené soustavy. Dalším vstupními veličinami do soustavy jsou veličiny poruchové, označované písmenem d , tyto poruchové veličiny analogové řízení díky zpětné vazbě respektuje.[1]



Obrázek 18 Schéma regulačního obvodu [1]

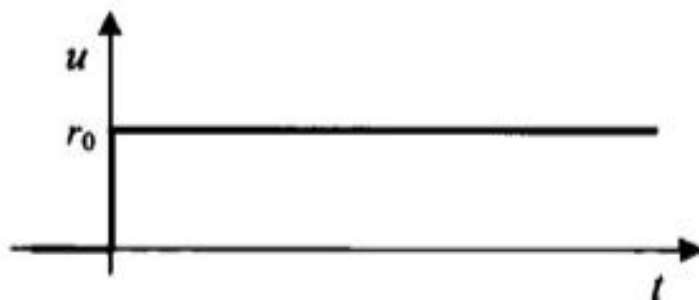
Regulátory, kterými se bude dále tato práce zabývat, jsou se složkou proporcionální, integrační, derivační a jejich vzájemnou kombinací. Označují se zkratkou PID.

PID regulátor se skládá ze tří složek. Složkou proporcionální, která zajišťuje přímou úměru mezi regulační odchylkou a výstupem regulátoru. Dále složkou integrační, sloužící k úpravě akční veličiny úměrou k integrálu regulační odchylky. Derivační složka zohledňuje poslední změnu regulační odchylky v čase.

$$u(t) = r_0 e(t) + r_i \int_0^t e(\tau) d\tau + r_D e'(t) + u(0) \quad (5)$$

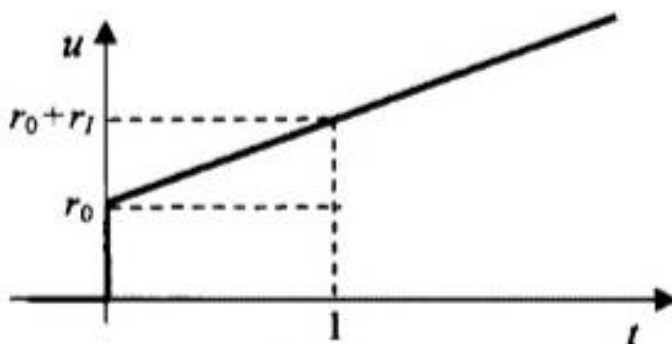
Jak je vidět v rovnici 5, je akční veličina součtem těchto složek, ze kterých se regulátor může skládat. Tyto složky můžeme odstraňovat a kombinovat vyjma případů kdy by rovnice obsahovala derivační složku bez přítomnosti složky proporcionální.

Konstanty u jednotlivých složek, které jsou v rovnici 5, slouží k seřízení chování regulátoru pro konkrétní soustavu. Konstanta r_0 určuje zesílení akční veličiny vůči regulační odchylce e . Proporcionální část regulátoru je částí statickou regulátoru a konstant r_0 je směrnici přímky statické charakteristiky regulátoru. Tedy pouze v ustáleném stavu soustavy. Samotná proporcionální složka v rovnici regulátoru nikdy nezajistí nastavení přesné žádané veličiny neboli vždy bude mít trvalou regulační odchylku. Regulační odchylka je totiž taková odchylka, kterou bude mít regulovaná veličina od žádané veličiny, když se čas bude limitně blížit nekonečnu. [1]



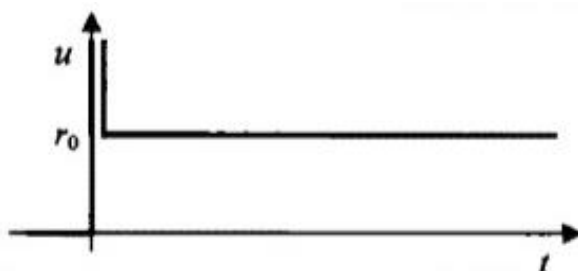
Obrázek 19 Přebodová charakteristika P regulátoru [1]

Integrační složka zajišťuje rychlost změny akční veličiny vůči regulační odchylce. Tato rychlost je nastavitelná pomocí konstanty r_i . Vyjadřuje zesílení rychlosti změny akční veličiny vůči regulační odchylce. Regulátor, který kombinuje integrační a proporcionální složku, odstraňuje trvalou regulační odchylku, protože rychlost změny akční veličiny musí být při ustáleném stavu rovna nule.



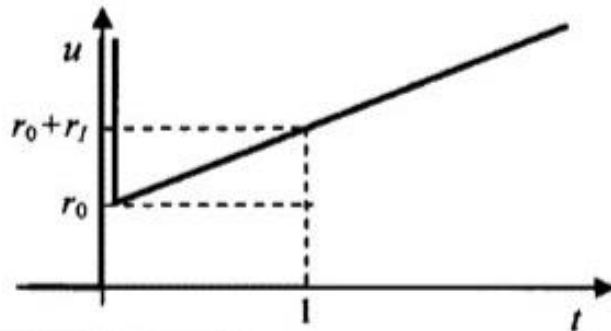
Obrázek 20 Přebodová charakteristika PI regulátoru [1]

Derivační složka PID regulátoru dokáže předpovědět vývoj odchylky. A díky tomu upravuje akční veličinu. Smí se použít pouze v kombinaci s P složkou nebo PI složkou. Opět pro nastavení zesílení akční veličiny vůči změně regulační odchylky slouží koeficient s označením r_D .



Obrázek 21 Přebodová charakteristika PD regulátoru [1]

Výsledná přechodová charakteristika PID regulátoru je na obrázku 22. Jsou zde vyznačeny konstanty pro jednotkový čas a jednotkovou regulační odchylku.



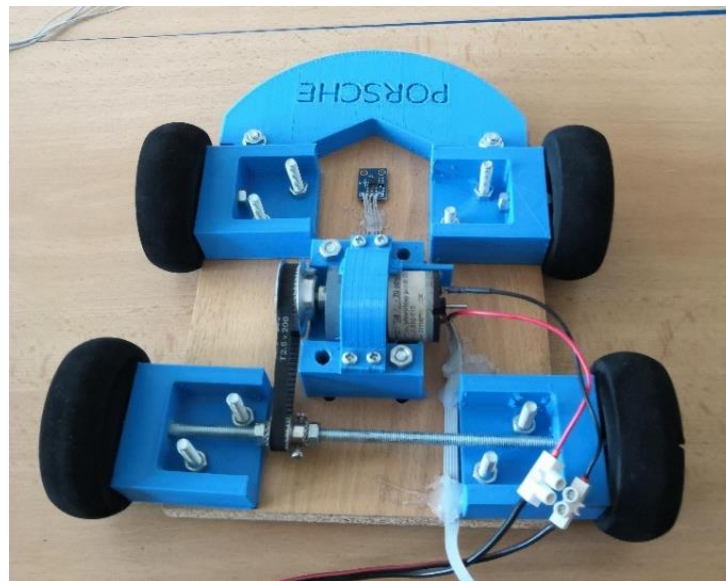
Obrázek 22 Přechodová charakteristika PID regulátoru[1]

6 Úprava konstrukce modelu

V následující podkapitole se zabývám prozkoumáním současného modelu a stanovením jeho konstrukčních problémů při vlastním pohybu. Řešením a realizací regulace a řízení pohybu se zabývám v kapitole 7.

6.1 Konstrukce původního modelu

Původní model pro měření zrychlení se skládal z hnané zadní nápravy se společnou osou, která byla vyrobena z metrické závitové tyče. Ta je uložena volně v otvorech plastových os, které byly vytištěny na 3D tiskárně. Podobně jsou uložena i přední kola, která ale nemají společnou osu a otáčejí se na dlouhých metrických šroubech s hlavou s vnitřním šestihranem. Tento způsob uložení vykazoval problémy, a to zejména ve vřeli v radiálním uložení a tření neoprenových kol o plochu plastových uchycení os. Chování modelu se z tohoto důvodu projevovalo neschopností modelu udržovat přímý pohyb a zasekáváním kol.



Obrázek 23 Fotografie modelu před úpravou

Pro přenos kroutícího momentu mezi převodovkou a zadní nápravou je použit ozubený řemenový převod. Vstupní ozubená řemenice je 40T_{2,5-16} a výstupní řemenice je 19T_{2,5-16}. Mezi řemenicemi je použit řemen T_{2,5-200}. Výsledný převodový poměr je tedy:

$$i_{rem} = \frac{Z_2}{Z_1} = \frac{19}{40} = 0,475 [-] \quad (6)$$

Pro všechny řemenové převody je důležitá napínací síla. V případě převodů řešených pomocí ozubeného řemenu je při nedostatečné přitlačné síle možnost přeskokování zubů na menší řemenici. Důvodem, proč k přeskokování dochází na menší řemenici, je menší úhel opásání, a tedy i méně zubů v záběru. Přitlačná síla se u současného modelu reguluje s využitím excentricity výstupního hřídele převodovky motoru. Natočení motoru poté umožňuje napnutí řemenu. Toto řešení je nedostačující, protože je velmi náročné až nemožné při ruční montáži zajistit dostatečnou napínací sílu.

Motor je uložený v plastovém pouzdře, kde není dostatečně zajištěn proti samovolnému rotačnímu pohybu vůči pouzdru. Svým pootočením v pouzdru způsobí snížení přitlačné síly až pod mez, kdy je řemen schopen kontinuálně přenášet kroutící moment a na řemenici hnané dochází k přeskoku zubů.



Obrázek 24: Fotografie demontovaného DC motoru z modelu

Na obrázku 24 je stejnosměrný motor GM37-20, který byl u modelu použit, má jmenovité napájecí napětí 12 V a jmenovité otáčky bez zatížení 5400 min^{-1} . Tento motor má na výstupní hřídeli převodovku s čelními ozubenými koly, tato převodovka je již spojena s motorem z výroby.

S převodovým poměrem 270:1 je výstupní rychlost velmi malá. Otáčky na výstupní hřídeli bez zatížení jsou totiž pouhých 20 min^{-1} . Tyto malé otáčky vyhovovaly sice pro laboratorní úlohu, ale aby mělo smysl regulovat rychlost a tím splnit zadání této práce, je třeba navýšit rychlost otáček alespoň dvakrát.

Následující seznam shrnuje problémy zmíněné výše, které je nutné vyřešit pro správný pohyb modelu:

- a) Tření v uložení kol
- b) Tření mezi koly a stěnou plastového uchycení hřídele
- c) Nemožnost dostatečného napnutí ozubeného řemenu
- d) Možné protáčení statoru v uchycení motoru na desku
- e) Nedostatečná rychlost otáčení motoru

6.2 Tvorba nového modelu

Dle předchozí kapitoly je třeba na modelu udělat změny, aby byl schopný pohybu a zvýšit jeho rychlost. Tato část práce mapuje změny provedeny na tomto modelu z hlediska konstrukce a pohonu.

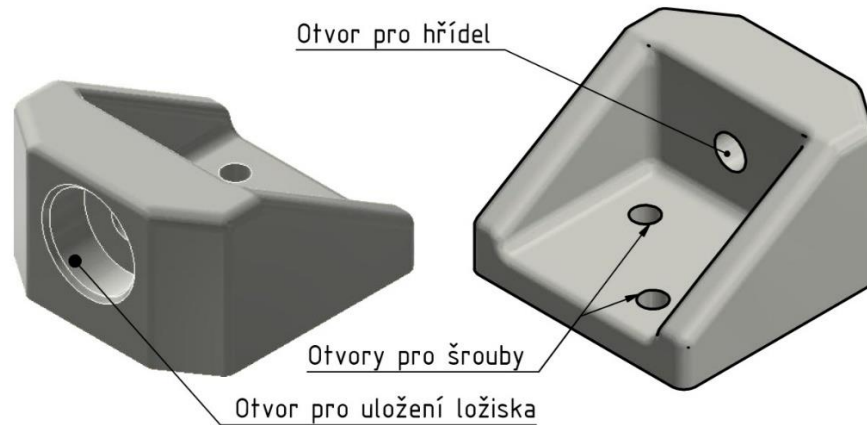
První úpravou tohoto modelu je vytvoření nového uložení kol, které zajistí hladký pohyb modelu. Cílem je tedy vytvořit takové uložení, aby nedocházelo ke tření a zasekávání kol. K řešení tohoto problému jsem zvolil poměrně snadné řešení. A to využití malých kuličkových ložisek, které budou zajišťovat hladký pohyb všech čtyř kol.

Rozměrově vhodná ložiska pro použití v tomto modelu jsou na trhu bez problému k dispozici. Pro tuto práci byla zvolena ložiska standardu ABEC, to je americký standard pro tolerování rozměrů ložisek. Ve rekreačně sportovním odvětví se s touto zkratkou můžeme setkat u ložisek pro *In-line* brusle. Pro jejich dostupnost a vhodné rozměry jsem zvolil tato ložiska pro sportovní účely.

U tohoto modelu je dosahováno otáček mnohonásobně menších než u otáček, kterých je dosahováno při tomto sportu. A proto není jejich stupeň přesnosti, který je použit podstatnou informací pro tuto práci. Účel, který v modelu plní je hladký chod kol a dostatečná souosost mezi osou otvoru pro hřídel a hřídelí.

Pro připevnění ložisek k dřevěné desce byly vymodelovány ložiskové domky, které jsou na desce umístěny v počtu čtyř a jsou identické. Jejich vizuální podobu vidíte na obrázku 25. Tvar a rozměry vychází z původních plastových os, které byly umístěny na modelu. Jsou ale tvarově složitější za účelem snížení

množství materiálu na jejich výrobu. Hlavní změnou je ale vytvoření otvorů pro ložiska, které mají vnější průměr 22 mm. Při volbě rozměru byla brána v potaz nepřesnost, která vzniká při tisku modelu na běžných 3D tiskárnách.



Obrázek 25: 3D model ložiskových domků

Nepřesnost, s kterou autor práce nepočítal, byla ve směru, v kterém tiskárna vrství jednotlivé vrstvy modelu. Došlo k malému zmenšení rozměru u otvoru pro umístění ložiska ve směru vrstev. Otvoru měl tedy díky této skutečnosti mírně elipsovitý průřez. Proto musel být otvor po vytisknutí modelu ručně upraven za pomoci ruční modelářské brusky.

Ložiska v otvorech u předních volných kol drží jen pomocí tření v díře, ale je to dostačující řešení pro udržení kol. Kola jsou připevněna k ložiskům pomocí šroubu, který se hlavou opírá o vnitřní kroužek ložiska. Z druhé strany ložiska se opírá plastový disk, na kterém je nasunuto neoprenové kolo. Plastový disk je k ložisku přitlačován pomocí dvojce matic s podložkou. Druhá matice má funkci pojistné matice.

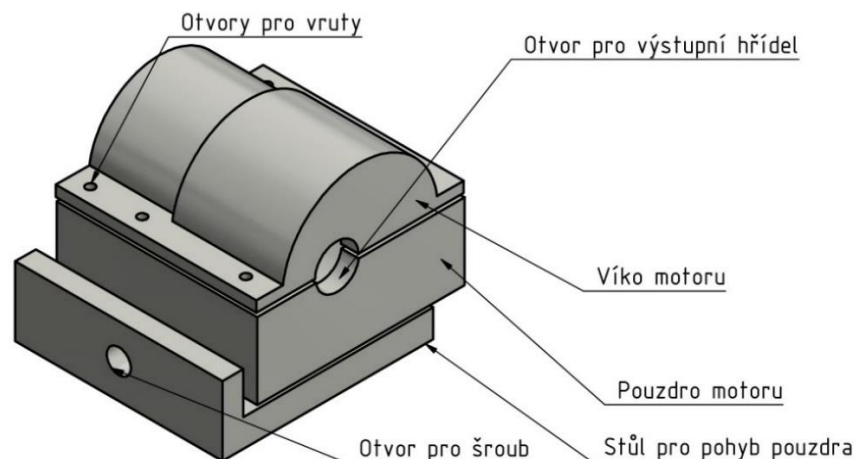
U zadních hnaných kol je důležité, aby se kola na osách neprotáčela. Aby se tomu zabránilo, byly na válcové styčné části, které se nachází mezi disky a neoprenovými koly, vytvořeny natavením plastu výstupky. Ty zabraňují pohybu neoprenových kol na těchto plastových discích.

K zabránění protáčení plastových disků kol vůči hřídeli slouží tření mezi maticí tímto diskem. Normálové síly potřebné k tomuto spoji bylo docíleno díky společné hřídeli zadních kol, která je vyrobená ze závitové tyče. Utažením matic

na této tyči dojde k dostatečnému tlaku na disky a kola se při jízdě neprotáčejí. Matice jsou proti povolování opět zajištěny pojistnou maticí.

Tímto jsou vyřešeny problémy a) a b) z předchozí podkapitoly (6.1) a dále je třeba se zabývat dalšími problémy modelu. A to napnutím řemenu mezi ozubenými řemenicemi pohonu a možnému protáčení statoru motoru v uchycení – neboli problémy c) a d).

Na otázku, jak dosáhnout dostatečné napínací síly na ozubeném řemenu, odpovídá jednoduché řešení v podobě posuvného stolu s pohybovým šroubem. Posuvný stůl pro tento model je inspirován velkými posuvnými stoly pro motory, jaké se používají u velkých motorů, které jsou spojeny s převodovkou klínovým řemenem.



Obrázek 26: 3D model sestavy uchycení motoru na dřevěné desce modelu

Na obrázku 26 je znázorněn pohled na model celé sestavy pouzdra pohonu. Ta se skládá ze stolu, pouzdra a víka. Stůl je připevněn k dřevěné desce modelu pomocí 4 vrutů. Ve stolu je vymodelovaná drážka, do které se zasouvá pouzdro motoru. Tato drážka zajišťuje, že pouzdro motoru se může pohybovat pouze ve směru této drážky.

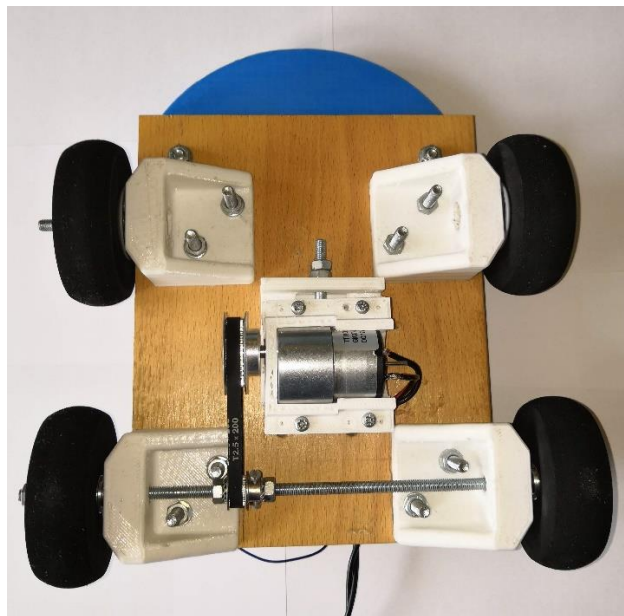
V pouzdru motoru je vymodelovaná drážka pro hlavu šroubu, a ten je při montáži vtlačen do této drážky. Šroub poté prochází otvorem pro šroub, který je označen na obrázku. Utahováním matice na tomto šroubu dochází k pohybu pouzdra motoru ve směru drážky ve stole, a tím k napínání ozubeného řemenu. Matice je zajištěna pojistnou maticí proti povolování.

K zamezení motoru v otáčení v pouzdru slouží otvor pro výstupní hřídel . Díky tomu, že výstupní hřídel převodovky má excentricitu vůči ose rotoru motoru, je při opření pouzdra kolem výstupní hřídele převodovky o otvor v pouzdru motoru zcela zajištěn proti pootočení. Víko poté pouze přidržuje motor ve správné poloze.

Při tisku těchto dílů na 3D tiskárně došlo ke vzniku studených spojů mezi tištěnými vrstvami na víku a na pouzdru motoru. U víka došlo dokonce k odlomení. Došlo k němu ale v dostatečné vzdálenosti od otvoru pro výstupní hřídel, a proto nenarušuje funkci víka. U druhého studeného spoje na pouzdru motoru je zabráněno odlomení díky vrutům, které spojují víko a pouzdro. Ty jsou totiž dostatečně dlouhé pro spojení i těchto dvou částí pouzdra, částečně rozdělených studeným spojem vrstev.

Díky novému uchycení motoru na desku byly vyřešeny problémy c) a d) z předchozí podkapitoly 6.1. Posledním problémem, kterým se tato práce zabývá je problém e), tedy nedostatečná rychlost otáček motoru.

Jako nový motor byl zvolen podobný ze stejné výrobní řady a to **GM37-105**. Z označení motoru tedy vychází, že výstupní otáčky převodovky použitého motoru jsou **105 ot*min⁻¹**. Motor má stejné hlavní rozměry jako původní motor. Tímto byla zvýšena rychlost modelu více než pětinásobně.



Obrázek 27: Fotografie výsledné podoby podvozku modelu po úpravách

Na obrázku 27 je výsledná podoba podvozku modelu po montáži. Model se po odzkoušení funkčnosti pohybuje bez problémů a jeho pohon dostačuje pro jeho pohyb. Výsledná maximální teoretická rychlost otáček na zadní hnané nápravě je:

$$n_{hmax} = \frac{n_{př}}{i_{řem}} = \frac{105}{0,475} = 221,05 \text{ ot} * \text{min}^{-1} \quad (7)$$

Z otáček a průměru kol můžeme vypočítat teoretickou rychlost modelu. Kde průměr kol je $d_{kol}=78 \text{ mm}$, pak výsledná maximální teoretická rychlost modelu je:

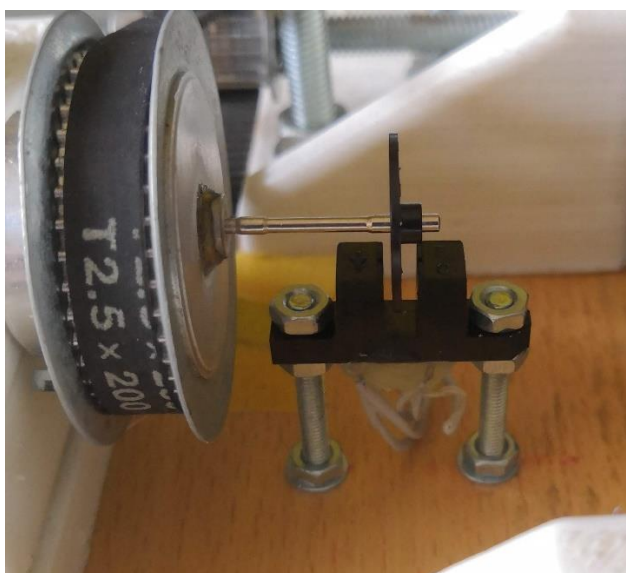
$$v_{mt} = \frac{\pi * d_{kol} * n_{hmax}}{60\,000} = 0,902784 \text{ m} * \text{s}^{-1} \quad (8)$$

Tímto byly vyřešeny všechny problémy stanovené v předchozí kapitole. První cíl práce prozkoumat a upravit konstrukci modelu je tedy považován za splněný.

7 Návrhy variant hardwarové instrumentace

V návrzích se zabývám volbou řídicí elektroniky pro regulaci otáček motoru pomocí PID regulátoru. Pro snímání otáček jsem se rozhodl, že použiji optoelektronický snímač otáček. Důvodem tohoto rozhodnutí, je jednoduchost tohoto snímače a také jeho velmi nízká cena.

Snímač, který jsem u modelu použil, jsem umístil na dvou dlouhých šroubech. Ty procházejí skrz dřevěnou desku modelu a jsou připevněny pomocí podložek a matic. Na šroubech je usazen senzor, který drží pomocí dvou matic na každém ze šroubů. Toto řešení umožňuje nastavení správné polohy senzoru vůči kolečku.

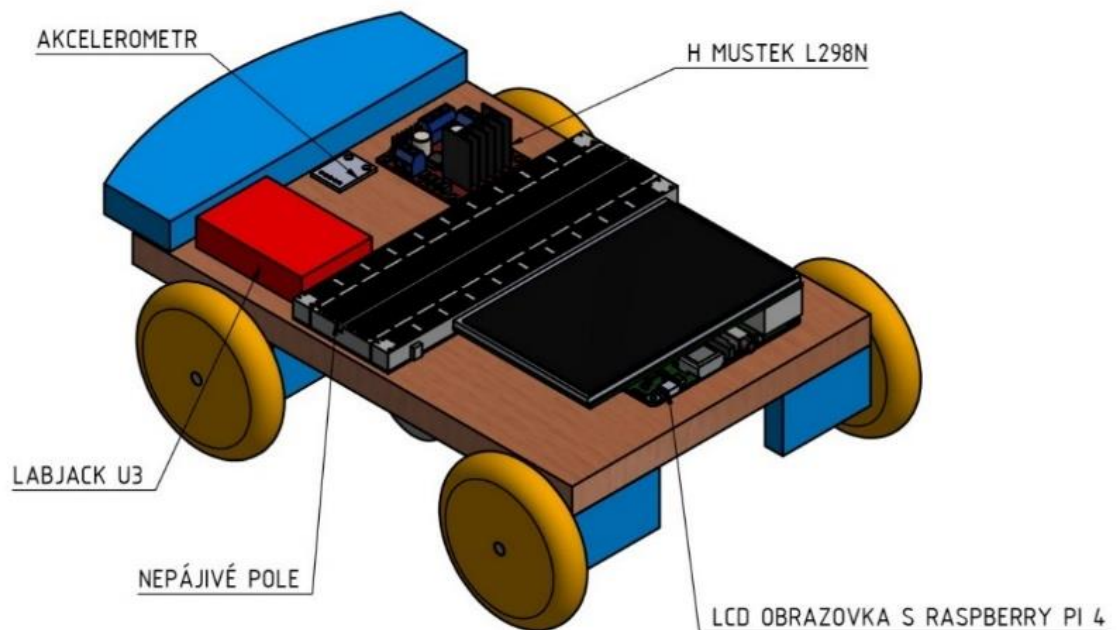


Obrázek 28 Optoelektronický snímač otáček na modelu

Kolečko neboli optický enkodér jsem získal ze staré počítačové myši. Hřídel, na které byl enkodér upevněn, byla využita k upevnění na rotační osu motoru. Byla vložena a následně přilepena za pomoci sekundového lepidla do otvoru pro hřídel ve větší ozubené řemenici. Poté, co jsem odzkoušel toto řešení, jsem zjistil, že nedochází k házení kolečka a upevnění hřídele je velmi silné.

Dále se věnuji návrhům uspořádáním komponent pro regulaci rychlosti a snímání průběhu zrychlení pomocí akcelerometru.

7.1 Návrh číslo 1



Obrázek 29 Zjednodušené schéma umístění hardware na model návrhu číslo 1

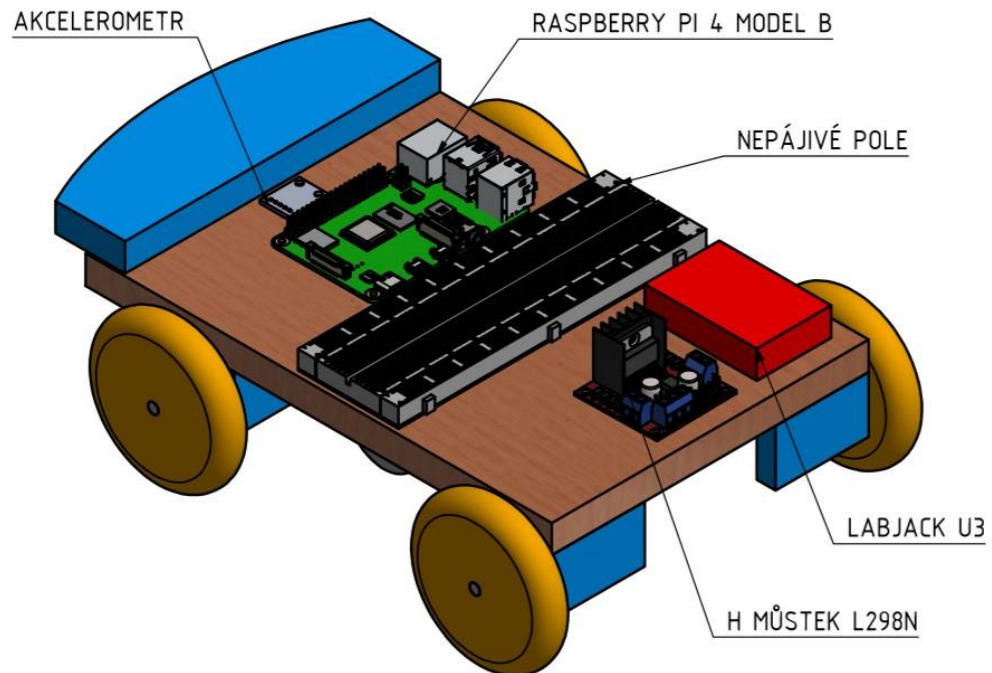
Toto řešení se skládá z následujících komponent:

- H-můstek L298N
- Akcelerometr GY-61
- Dotykový LCD Display - 7"
- Raspberry Pi 4 B
- LabJack U3
- Optická závora TCST2300
- Nepájivé pole

Tento návrh hardwarové instrumentace pro regulaci rychlosti a ovládání modelu využívá pouze jednu řídicí desku. Deska Raspberry Pi 4 B řídí veškeré procesy a reguluje rychlost otáčení. K připojení senzoru otáček a k zajištění PWM výstupu slouží deska LabJack U3, která je připojena k Raspberry pomocí sériového rozhraní USB. K desce je připojena dotyková LCD obrazovka k zadávání vstupních hodnot a k spuštění modelu. Doplněné nepájivé pole je zde pro případ budoucích úprav na modelu.

Výhodou tohoto návrhu je pouze jedna řídicí deska. Nevýhodou je, že se musí doplnit o další zařízení LabJack, a to z důvodu, že Raspberry Pi nedisponuje analogovými vstupními piny, a tedy sama nedokáže vyčíst hodnoty z akcelerometru. Možným problémem tohoto řešení je až příliš velké místo, kterou by zabírala obrazovka a také její vlastní hmotnost.

7.2 Návrh číslo 2



Obrázek 30 Zjednodušené schéma rozložení použitých komponent pro návrh 2

Toto řešení se skládá z následujících komponent:

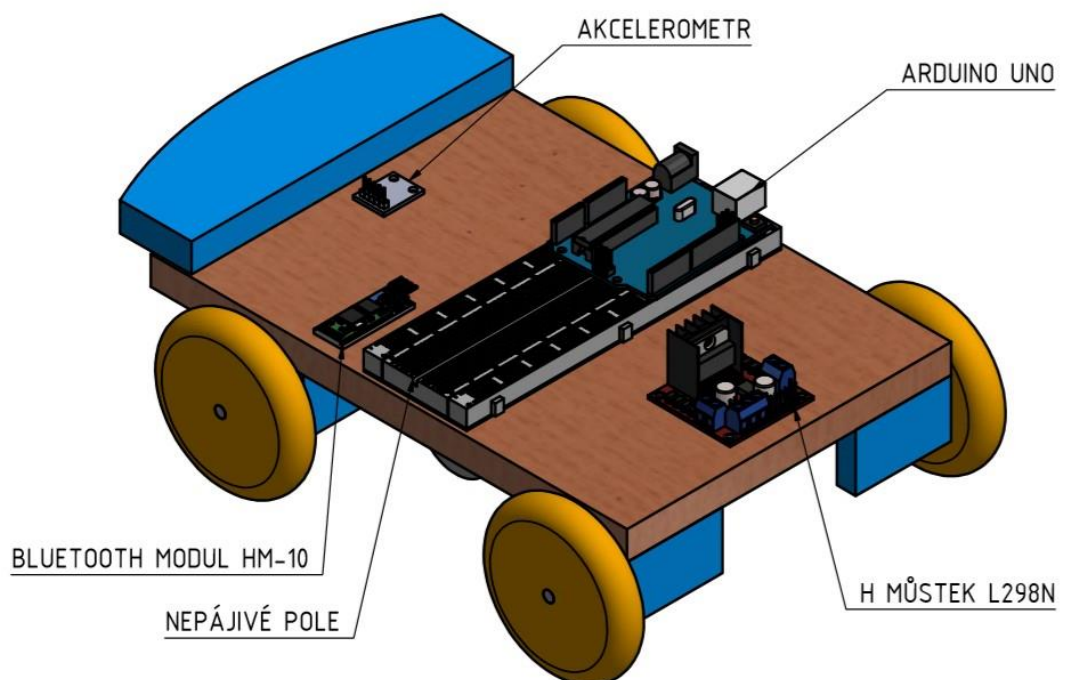
- H-můstek L298N
- Akcelerometr GY-61
- Dotykový LCD Display - 7" – mimo model
- Raspberry PI 4 B – mimo model
- Raspberry PI 4 B
- LabJack U3
- Optická závora TCST2300
- Nepájivé pole

Oproti předchozímu řešení se nabízí možnost použít dvě desky Raspberry. Jedna deska přímo na vlastním modelu k řízení modelu, která opět využívá desku LabJack k čtení analogových vstupních signálů a k řízení otáček motoru.

Druhá deska je součástí kompletu se stejnou LCD obrazovkou jako u předchozího návrhu, ale slouží k předávání příkazů mezi oběma deskami. Desky mezi sebou komunikují prostřednictvím Bluetooth, nebo pomocí bezdrátové sítě Wifi.

Výhodou tohoto uspořádání je odstranění velké zátěže v podobě LCD obrazovky a možnost řízení modelu bezdrátově. Nevýhodou však zůstává používání prostředníka pro měření na akcelerometru, což zvyšuje cenu použité hardwarové instrumentace.

7.3 Návrh číslo 3



Obrázek 31 Zjednodušené schéma rozložení použitých komponent pro návrh 3

Toto řešení se skládá z následujících komponent:

- H-můstek L298N
- Akcelerometr GY-61

- Dotykový LCD Display - 7" – mimo model
- Raspberry Pi 4 B – mimo model
- Arduino UNO
- Bluetooth modul HM-10
- Optická závora TCST2300
- Nepájivé pole

Poslední navrhovaná varianta se zaměřuje na snížení ceny a to tím, že nepoužívá měřící zařízení LabJack ani druhou desku Raspberry Pi. Místo toho, je regulace řízená pomocí levnější a jednodušší řídicí desky Arduino UNO. Ta totiž obsahuje nejen PWM výstup ale oproti Raspberry i analogové vstupní piny.

Arduino samo nedisponuje bezdrátovou technologií, proto je v tomto návrhu doplněno a Bluetooth modul HM-10, s kterým komunikuje po sériové lince. Na tento modul jsou posílány příkazy z externího zdroje. Tímto zdrojem může být libovolný počítač nebo chytrý mobilní telefon se sériovým Bluetooth monitorem.

Výhodou této varianty je tedy snížení ceny o drahé zařízení LabJack. Možnost čtení analogových výstupů přímo deskou Arduino, která má také nižší cenu než deska Raspberry Pi. Další výhodou je možnost napájení desky Arduino 12 V, které již tak musí být přivedený k pro napájení motoru a není tedy potřeba. Nebo možnost napájení desky Arduino pomocí baterie.

8 Volba varianty

Pro rozhodnutí mezi jednotlivými návrhy jsem zvolil jako hlavní kritérium cenu. Dále pak vlastní výhody a nevýhody jednotlivých návrhů. Do ceny bych mohl zahrnout i fakt, že některé komponenty již jsou k dispozici a nemusí se dokoupit, nicméně cenu výsledného modelu jsem uvedl zejména pro úvahu, jestli je dané zařízení zbytečné pro model na měření určený výhradně pro výuku nikoli výzkum. Ceny ze stránek prodejců jsem zapsal do tabulky 3. Jsou v ní uvedeny ceny za jednotlivé díly a počty kusů potřebné pro daný návrh. Výsledná suma je poté pro volbu návrhu důležitým aspektem.

Tabulka 3 Cenová kalkulace hardwarové instrumentace (ceny s DPH ke dni 29. 7. 2019) [7, 13, 25–28]

Hardware	Cena [Kč]	Návrh 1 [ks]	Návrh 2 [ks]	Návrh 3 [ks]
<i>Nepájivé pole</i>	109,-	1	1	1
<i>Akcelerometr GY-61</i>	198,-	1	1	1
<i>H-můstek L298N</i>	69,-	1	1	1
<i>Optická závora TCST2300</i>	24,-	1	1	1
<i>Dotykový LCD display 7"</i>	2 299,-	1	1	1
<i>LabJack U3-HV</i>	2 733,-	1	1	0
<i>Raspberry Pi 4 B 2 GB</i>	1 490,-	1	2	1
<i>Arduino UNO rev. 3 klon</i>	180,-	0	0	1
<i>Bluetooth modul HM-10</i>	129,-	0	0	1
<i>Suma [Kč]</i>		6 922,-	8 412,-	4 498,-

Dle tabulky 3 vidíme, že nejnižší cena nám vychází pro model s deskou Arduino, tedy návrh číslo tři.

Tento návrh se zdá být vhodný i z hlediska dobrého rozložení komponent na desce, a i dostatečného místa pro budoucí rozvoj modelu nebo jeho úpravu. Proto jsem zvolil návrh číslo 3.

9 Regulace otáček pohonu pomocí zvolené varianty

Specifikem mnou zvoleného návrhu je použití vývojové desky Arduino pro regulaci rychlosti otáček motoru. Tu jsem naprogramoval pro regulaci a přijímání příkazů pomocí Bluetooth. Příkazy přijímá Arduino z ovládací aplikace, které je spuštěna na tabletu s Raspberry Pi.

9.1 Zapojení

Nejdříve jsem otestoval zapojení na nepájivém poli, díky čemuž jsem zjistil, že zapojení je funkční.

Pro dostatečnou mechanickou odolnost modelu při nárazech je nutno, aby nebyl žádný vodič pouze zasunut do připojovacích pinů a aby všechny spoje byly řádně připájené.

Při připojení jsem kladl důraz na možnost budoucího rozvoje modelu, a proto jsem plně připojil všechny piny Bluetooth modulu, akcelerometru k mikrokontroleru Arduino. H-můstek L298N je připojen pouze pro použití s jedním motorem, i když je tato součást schopna řídit dva.

Zjednodušené schéma zapojení všech součástí vidíte na obrázku 32.

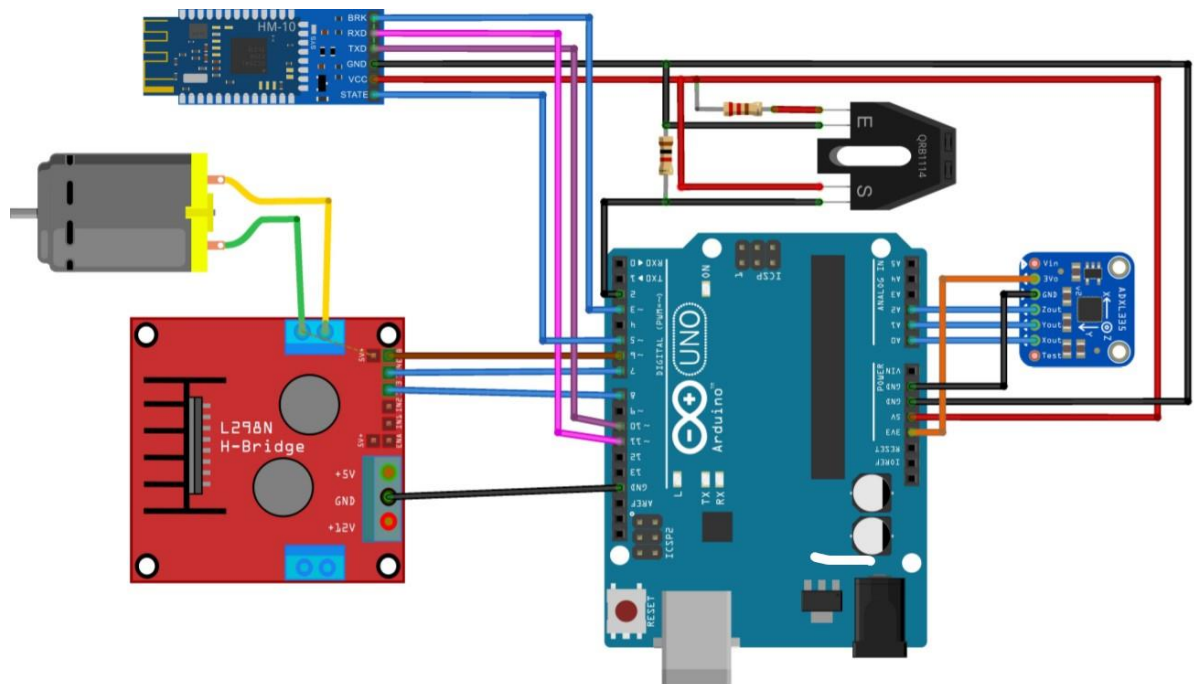
Tabulka 4 Použití pinů desky Arduino UNO

Digitální piny	Použití	10	HM – 10, TX
0	-	11	HM – 10, RX
1	-	12	-
2	Senzor otáček	13	-
3	HM -10, BRK	Analogové piny	
4	-	Použití	
5	HM -10, STATE	A0	GY – 61, X
6	L298N, ENB	A1	GY – 61, Y
7	L298N, IN3	A2	GY – 61, Z
8	L298N, IN4	A3	-
9	Chybný	A4	-
		A5	-

V tabulce 4 je seznam vstupních a výstupních pinů, které jsou využity s popiskem pinu součásti, který je na něj připojen. Pomlčkou jsou označeny volné piny. Napájecí a zemnicí piny nejsou v tabulce označeny.

Pin číslo 9 byl původně určen k řízení otáček motoru, ale při testování model vykazoval špatné chování PWM výstupu na tomto pinu. Pravděpodobně jsem tento pin poškodil při pájení vodiče k desce, proto jsem vodič pro PWM přesunul na pin 6, který rovněž touto funkcí disponuje.

Napájení pro všechna zařízení je odebíráno z napájecích pinů desky Arduino. Na model je přivedeno napětí o velikosti 12 V, které vyžaduje stejnosměrný motor. Protože jsem nepoužil originální desku Arduino nýbrž její kopii, je možnost přivedeného napětí na napájecí konektor až 24 V. Arduino si toto napětí sníží na 5 V.



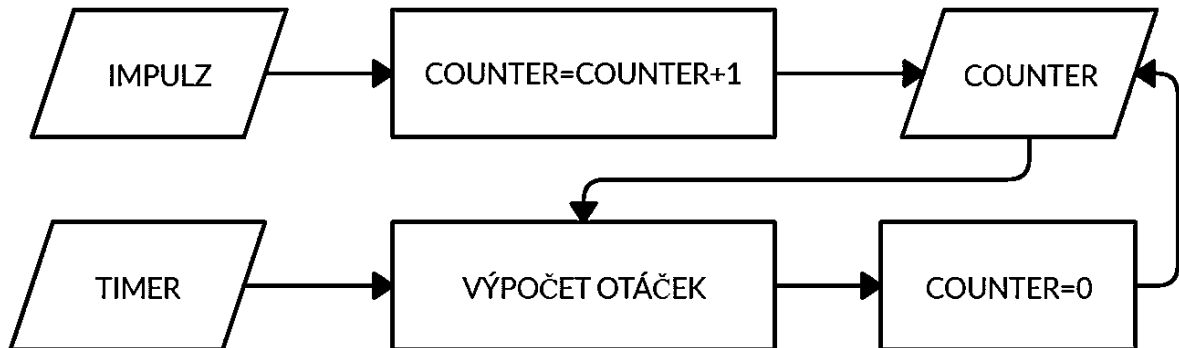
Obrázek 32 Schéma zapojení na modelu vytvořeného programem Fritzing

9.2 Snímání otáček deskou Arduino

Prvním problémem je, jak naprogramovat snímání otáček pomocí impulzů z fototranzistoru optického snímače.

Arduino UNO disponuje dvěma piny, u kterých je možné použít funkci `attachInterrupt()`, jsou jimi digitální piny 2 a 3. Tato funkce pracuje tak, že při stavu `HIGH` na jednom z těchto dvou pinů přeruší program aktuální činnost a vykoná libovolnou definovanou funkci. Funkce `attachInterrupt()` je tedy vhodná pro měření rychlosti pomocí optoelektronického snímače, protože má na výstupu čtvercový signál.

Funkcí, kterou každý impuls signálu vyvolává, je čítač. Tedy probíhá počítání jednotlivých impulsů. To se ukládá do proměnné a ta je v pevných časových intervalech vynulována.



Obrázek 33 Schéma měření rychlosti otáček

Časovač, který v pravidelných intervalech provede přečtení čítače, vypočte aktuální rychlost otáček a čítač vynuluje. Výslednou rychlost vždy zapíše do proměnné, která se tedy v pravidelných intervalech mění.

$$n = \frac{i \cdot 60}{q} [\text{min}^{-1}] \quad (9)$$

Pro výpočet otáček za minutu využívá Arduino rovnici 9, kde i je počet impulsů napočítaný čítačem, q je počet otvorů v enkodéru. Vztah 9 platí při periodě časovače 1 s.

Tímto způsobem získává Arduino informaci o rychlosti. Tuto informaci dále využívá regulátor.

9.3 Regulace PID

Napětí přivedené na motor je řízeno pomocí PWM signálu na pinu 6 mikrokontroleru. Pro získání akční veličiny v podobě proměnné v rozpětí celých čísel 0 až 255 pro PWM výstup, je použita knihovna *PID_v1.h*. Ta je rovněž volně ke stažení ze stránek výrobce mikrokontroleru.

Při spuštění nebo resetování systému se nastaví konstanty PID regulátoru a přepne se režim regulace z manuálního řízení. Dále se nastaví, zda bude hodnota akční veličiny klesat nebo bude vzrůstat. Toto nastavení závisí na tom, co chceme regulovat. V případě, že se jedná o rychlost auta, tak chceme, aby akční veličina rostla. Opačný případ se používá například pro řízení chlazení

klimatizací, při tomto nastavení knihovna převrátí znaménka u konstant regulátoru.

Vlastní výpočet PID pracuje tak, že přijímá vstupní proměnné, které jsou žádaná hodnota a aktuální hodnota. Z rozdílu těchto dvou veličin vypočítá regulační odchylku, dále vypočítá rozdíl poslední uložené hodnoty a aktuální hodnoty. Pro numerické řešení integrační složky využívá tato knihovna pravou obdélníkovou metodu.

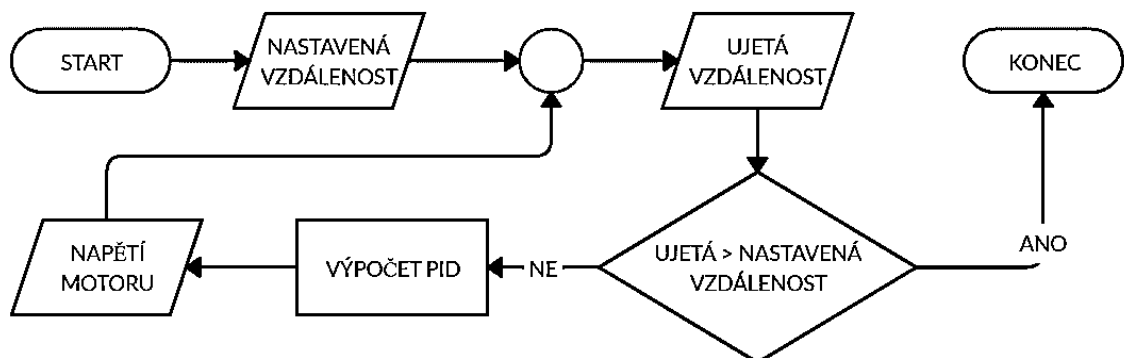
Vynásobí chyby konstantami a upraví hodnotu tak, že výstupním signál je již v rozsahu 140 až 255. Pokud akční veličina by byla vyšší než 255, tak knihovna bude vracet z výpočtu právě toto číslo.

Rozmezí výstupního signálu vychází z minimální hodnoty, kdy je motor ještě schopen rozjet model. Nastavuje se taktéž při spuštění nebo resetování desky Arduino.

Výstupní signál je zapisován příkazem `analogWrite()` na pin 5, který řídí napětí vstupující do H-můstku.

9.4 Snímání ujeté vzdálenosti

Model potřebuje poznat, kdy dojel dostatečně daleko od stěny a kdy ukončil měření. K tomu slouží sčítání počtu pulzů bez nulování v časových intervalech. Stejně jako při měření rychlosti, a dokonce ve stejné funkci vyvolané signálem ze senzoru, dochází ke sčítání počtu impulzů ve dvou čítačích. Jeden je pro rychlost a ten se v pravidelných intervalech a druhý pro celkový počet impulzů.



Obrázek 34 Schéma regulované jízdy na určitou vzdálenost

Celkový počet impulzů řídí, kdy se má model zastavit a rozjet opačným směrem proti zdi. Není podstatné, jak přesně daleko od stěny odjede, ale jen tak daleko, aby se stihla rychlost ustálit na určité hodnotě a touto rychlostí vrazil do stěny.

Při překročení počtu pulzů, které jsou definované v kódu programu, se model zastaví a provádí další úkon. Tímto způsobem je řízen chod programu a postup chování modelu.

9.5 Průběh měření

Model se před započítím měření přiloží ke stěně tak, aby se nárazník dotýkal stěny v místě nárazu, kterého chceme docílit. Zapnutím modelu se spustí funkce pro inicializaci modelu.

Prvním úkolem je vyčkávání příkazů, které jsou, jakou rychlostí má model při nárazu jet, dále vzdálenost, o kterou se má oddálit od stěny, a vzdálenost, kterou má ujet vpřed ke stěně. Tyto hodnoty model očekává přes sériové rozhraní Bluetooth. Jakmile model získá hodnoty z funkce zajišťující vyčítání sériové komunikace, převede tento řetězec znaků na číslo a uloží si ho. Poté dojde ke spuštění sekvence jízdy pro měření.

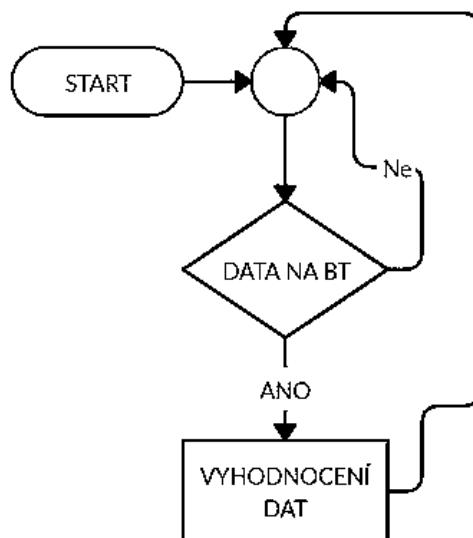
Ta probíhá tak, že model o uloženou vzdálenost odjede od stěny rychlostí, která je nastavená uvnitř programu. Dále vyčká 5 sekund a rozjede se rychlostí získanou ze sériové komunikace. Po nárazu do stěny model ukončí sekvenci. Zjednodušený průběh jízdy je na obrázku 35.



Obrázek 35 Průběh sekvence jízdy při měření

9.6 Ovládání modelu pomocí externího systému

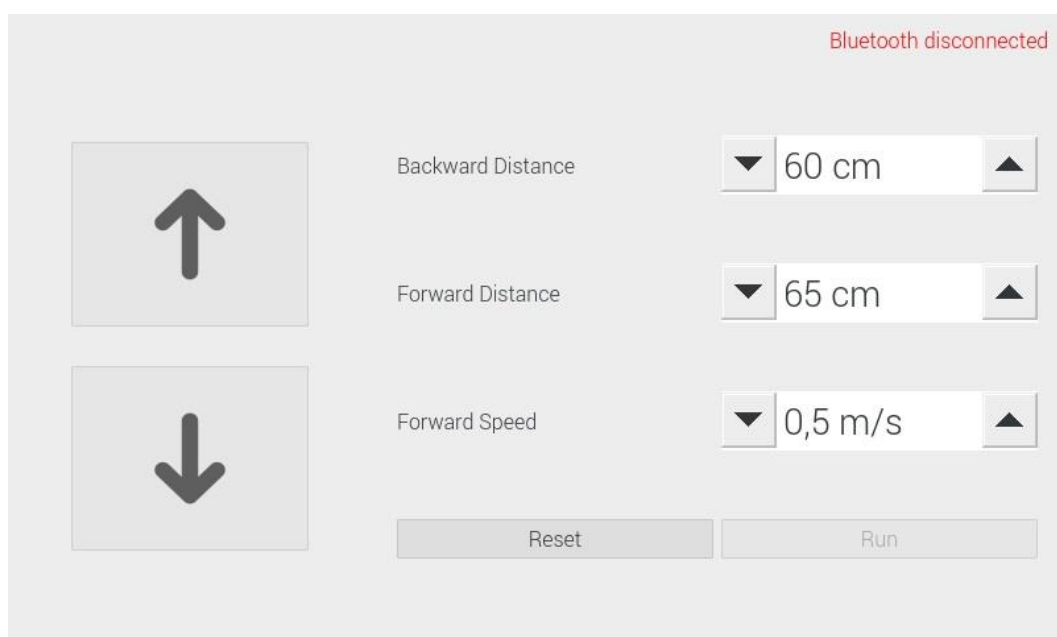
Model komunikuje s uživatelským rozhraním, které je na tabletu s Raspberry Pi. Komunikují přes sériové rozhraní Bluetooth.



Obrázek 36 Hlavní smyčka programu

V hlavní smyčce programu, kterou vidíte na obrázku 37, vyčkává Arduino na příkaz z Bluetooth. Ten posílá ovládací panel v podobě písmena. Jednotlivá písmena označují funkci, kterou model disponuje. Jsou jimi okamžitá jízda vpřed, jízda vzad a stop. To jsou funkce, které slouží k pohybu modelu mimo měřicí sekvenci a jsou aktivovány pomocí tlačítek se šipkami v levé části rozhraní, které vidíte na obrázku 370,70.

Další funkce, kterou aktivují písmena, jsou nastavení žádané rychlosti, nastavení vzdálenosti vpřed a vzad, spuštění sekvence jízdy a odeslání nastavených dat zpět do ovládacího rozhraní.



Obrázek 37 Uživatelské prostředí

10 Závěr

V této práci jsem se zabýval modernizací modelu automobilu k laboratorní úloze, která je měřena v předmětu Technická měření.

V této práci jsem měl za úkol zpracovat rešerši hardwarové instrumentace, kterou bych mohl použít k vytvoření modelu. Dále jsem měl upravit konstrukci modelu a na závěr naprogramovat zvolené řešení a odzkoušet funkčnost modelu. Rešerši hardwarové instrumentace jsem rozdělil na čtyři hlavní kapitoly.

V první z těchto čtyř kapitol se zabýval teorií logického a analogového řízení. V části analogového řízení jsem se věnoval zejména charakteristice a chování PID regulátoru.

V následující kapitole jsem se zabýval elektronikou vhodnou pro výše zmíněný model, která zajistí řízení modelu. Zmapoval jsem zařízení Arduino, Raspberry Pi a LabJack, protože jsem tyto tři zvažoval pro použití na modelu.

Na tuto kapitolu navazuje část, v které se věnuji senzorům k měření otáček. Zaměřil jsem se zde na senzory, které nejsou příliš velké a jejich měřící metody jsou vhodné pro model, který jsem upravoval v této práci. Popsal jsem zde měření optoelektronickou závorou, indukčním snímačem a pomocí tachodynamu. Rešerši jsem pro úplnost doplnil o princip akcelerometru, přestože nebyl předmětem cíle práce.

V poslední části rešerše jsem se zabýval teorií fungování stejnosměrného stroje. Dále jsem se zaměřil na stejnosměrný stroj s paralelním buzením, protože se do této skupiny řadí motor s permanentními magnety, který je použitý u modelu.

Poté jsem se zabýval principem řízení otáček motoru pomocí předřadného odporu a pulzně šířkové modulace. A závěrem jsem popsal princip H-můstku, nutného pro ovládání směru otáček motoru a vstupního napětí. Tímto jsem splnil první cíl práce, kterým bylo provedení rešerše cenově dostupné hardwarové instrumentace pro řízení modelu.

V další kapitole jsem se zabýval již vlastním řešením práce, a to nejprve úpravou konstrukce modelu. Nejdříve jsem zmapoval původní model a vytvořil

seznam problémů, které bylo potřeba vyřešit. Byly to problémy s pohyblivostí kol, napnutím řemenu, špatného uchycení motoru a nedostatečná rychlost modelu.

Následně jsem provedl řadu úprav, kdy jsem využil některé původní části jako nárazník, šrouby, kola a dřevěnou desku, která sloužila jako hlavní nosný prvek všech součástí. Dále jsem v modelu nahradil nevyhovující součástky za nové, které jsem vymodeloval a následně vytisknul na 3D tiskárně. Model jsem sestavil a odzkoušel jeho pohyblivost. Ten se pohyboval hladce a přímočaře. Tím jsem splnil další cíl práce, upravit konstrukci modelu.

Další část práce se věnovala hardwaru pro regulaci. Zde jsem zpracoval tři návrhy. Z těch jsem vybral návrh třetí, který byl nejvýhodnější z hlediska ceny, dále vhodný ze stránky ergonomie práce s modelem a zatížení modelu součástkami.

V poslední části práce jsem se věnoval zapojení obvodu pro regulaci. Dále jsem měl za úkol naprogramovat model. Práce obsahuje popis principu fungování měření otáček modelu. Dále jsem rozebíral princip regulace modelu a jakým způsobem model zaznamenává ujetou vzdálenost.

Na závěr jsem popsal průběh vlastního měření. Tedy, jak probíhá komunikace mezi modelem a ovládacím programem. Popsal jsem uživatelské prostředí ovládacího panelu. Tímto jsem splnil poslední cíl práce, a to naprogramovat a otestovat zvolené řešení.

Seznam použitých zdrojů

- [1] HOFREITER, Milan, ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE a STROJNÍ FAKULTA. *Základy automatického řízení*. V Praze: České vysoké učení technické, 2012. ISBN 978-80-01-05007-1.
- [2] What is a Raspberry Pi? *Raspberry Pi* [online]. [vid. 2019-03-14]. Dostupné z: <https://www.raspberrypi.org/help/what-is-a-raspberry-pi/>
- [3] Raspberry Pi 3 Model B+. *Raspberry Pi* [online]. [vid. 2019-03-14]. Dostupné z: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>
- [4] *Raspberry Pi PoE HAT - RPishop.cz* [online]. [vid. 2019-03-18]. Dostupné z: <http://rpishop.cz/raspberry-pi/1040-raspberry-pi-poe-hat.html>
- [5] Download Raspbian for Raspberry Pi. *Raspberry Pi* [online]. [vid. 2019-03-14]. Dostupné z: <https://www.raspberrypi.org/downloads/raspbian/>
- [6] *Raspberry Pi Raspberry Pi Gpio Access | Raspberry Pi* [online]. [vid. 2019-03-19]. Dostupné z: <https://www.electronicwings.com/raspberry-pi/raspberry-pi-gpio-access>
- [7] *U3 | LabJack* [online]. [vid. 2019-07-26]. Dostupné z: <https://labjack.com/products/u3>
- [8] *Arduino - Home* [online]. [vid. 2019-03-20]. Dostupné z: <https://www.arduino.cc/>
- [9] *Arduino Uno Rev3* [online]. [vid. 2019-03-20]. Dostupné z: <https://store.arduino.cc/arduino-uno-rev3>
- [10] MARCUSJ, Author. Arduino Pinout Diagrams. *marcusjenkins.com* [online]. 13. červen 2014 [vid. 2019-05-04]. Dostupné z: <http://marcusjenkins.com/arduino-pinout-diagrams/>
- [11] UPTON, Eben a Gareth HALFACREE. *Raspberry Pi: uživatelská příručka*. Brno: Computer Press, 2013. ISBN 978-80-251-4116-8.
- [12] *A beginner's guide to accelerometers* [online]. [vid. 2019-05-01]. Dostupné z: <https://www.dimensionengineering.com/info/accelerometers>
- [13] R.O, CzechProject spol s. Akcelerometr ADXL335 3-osy Analogový výstup | *arduino-shop.cz*. *arduino-shop.cz* [online]. [vid. 2019-05-02]. Dostupné z: <https://arduino-shop.cz/arduino/1260-akcelerometr-adxl335-3-osy-analogovy-vystup.html>
- [14] *ELUC* [online]. [vid. 2019-05-02]. Dostupné z: <https://eluc.kr-olomoucky.cz/verejne/lekce/1590>
- [15] *TCST2103 | GM electronic, spol. s.r.o.* [online]. [vid. 2019-07-26]. Dostupné z: <https://www.gme.cz/tcst2103>
- [16] NOVÁK, Martin, ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE a STROJNÍ FAKULTA. *Technická měření*. 2018. ISBN 978-80-01-06388-0.
- [17] *Indukční čidlo CS18-08P-1 M18, 0-8 mm, PNP/NO, konektor | GM electronic, spol. s.r.o.* [online]. [vid. 2019-07-26]. Dostupné z: <https://www.gme.cz/indukcni-cidlo-m18-0-8-mm-pnp-no-konektor>
- [18] *TACHODYNAMA, TACHOGENERÁTORŮ | Ložiska Vokoun* [online]. [vid. 2019-08-04]. Dostupné z: <http://www.loziska-vokoun.cz/elektromotory-atas/tachodynamata-tachogeneratoru/>

- [19] UHLÍŘ, Ivan, ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE a STROJNÍ FAKULTA. *Elektrické stroje a pohony*. Praha: Nakladatelství ČVUT, 2007. ISBN 978-80-01-03730-0.
- [20] R.O, CzechProject spol s. Stejnoseměrný motor s převodovkou DC 6V 15 RPM N20 | arduino-shop.cz. *arduino-shop.cz* [online]. [vid. 2019-08-04]. Dostupné z: <https://arduino-shop.cz/arduino/3196-stejnosmerny-motor-s-prevodovkou-dc-6v-15-rpm.html>
- [21] PAVLINAC. *Princip stejnosměrných motorů* [online]. [vid. 2019-07-10]. Dostupné z: <https://elektrika.cz/data/clanky/princip-stejnosmernych-motoru>
- [22] *Arduino - PWM* [online]. [vid. 2019-05-04]. Dostupné z: <https://www.arduino.cc/en/Tutorial/PWM>
- [23] NAVODY.ARDUINO-SHOP.CZ. *Změna směru otáčení DC motoru | Arduino návody* [online]. [vid. 2019-05-04]. Dostupné z: <https://navody.arduino-shop.cz/technikuv-blog/zmena-smeru-otaceni-dc-motoru.html>
- [24] *What is an H-Bridge? - Build Electronic Circuits* [online]. [vid. 2019-05-03]. Dostupné z: <https://www.build-electronic-circuits.com/h-bridge/>
- [25] *Modul bluetooth 4.0, BLE | GM electronic, spol. s.r.o.* [online]. [vid. 2019-07-29]. Dostupné z: <https://www.gme.cz/modul-bluetooth-4-0-ble>
- [26] *Nepájivé kontaktní pole ZY-103 | GM electronic, spol. s.r.o.* [online]. [vid. 2019-07-29]. Dostupné z: <https://www.gme.cz/nepajive-kontaktni-pole-zy-103>
- [27] R.O, CzechProject spol s. Raspberry Pi 4 Model B - 2GB RAM | arduino-shop.cz. *arduino-shop.cz* [online]. [vid. 2019-07-29]. Dostupné z: <https://arduino-shop.cz/arduino/7591-raspberry-pi-4-model-b-2gb-ram.html>
- [28] *Raspberry Pi Touch display 7"* [online]. [vid. 2019-07-29]. Dostupné z: <https://www.czc.cz/raspberry-pi-touch-display-7/229225/produkt>

Seznam obrázků

Obrázek 1: Raspberry Pi 3 B+ [3]	10
Obrázek 2 PoE HAT modul [4]	11
Obrázek 3 Schéma GPIO konektorů na Raspberry Pi 3 [6]	12
Obrázek 4 Deska LabJack U3-LV [7]	13
Obrázek 5 Arduino UNO Rev 3 [9]	14
Obrázek 6 Vidlicová světelná závora TCST 2103 [15]	15
Obrázek 7 Indukční čidlo CS18-08P-1 M18 [17]	16
Obrázek 8 Tachodynamo K10A6 [18]	17
Obrázek 9 Modul 3-osého akcelerometru ADXL335 [13]	17
Obrázek 10 Piezoelektrický jev na krystalu křemíku [14]	18
Obrázek 11 Stejnoseměrný motor s převodovkou [20]	19
Obrázek 12: Zjednodušené schéma stejnosměrného stroje [21]	20
Obrázek 13: Schéma zapojení buzení při cizím buzení (vlevo) a paralelním buzení (vpravo) [19]	21
Obrázek 14: Momentová charakteristika stejnosměrného motoru s cizím a paralelním buzením [19]	22
Obrázek 15 Závislost napětí na čase při PWM 0% [22]	23
Obrázek 16 Schéma jednoduchého H-můstku [23]	24
Obrázek 17 Schéma H-můstku s možností regulace výstupního napětí [24]	24
Obrázek 18 Schéma regulačního obvodu [1]	27
Obrázek 19 Přechodová charakteristika P regulátoru [1]	28
Obrázek 20 Přechodová charakteristika PI regulátoru [1]	28
Obrázek 21 Přechodová charakteristika PD regulátoru [1]	28
Obrázek 22 Přechodová charakteristika PID regulátoru [1]	29
Obrázek 23 Fotografie modelu před úpravou [autor]	30
Obrázek 24: Fotografie demontovaného DC motoru z modelu	31
Obrázek 25: 3D model ložiskových domků	33
Obrázek 26: 3D model sestavy uchycení motoru na dřevěné desce modelu	34
Obrázek 27: Fotografie výsledné podoby podvozku modelu po úpravách	35
Obrázek 28 Optoelektronický snímač otáček na modelu	37
Obrázek 29 Zjednodušené schéma umístění hardware na model návrhu číslo 1	38
Obrázek 30 Zjednodušené schéma rozložení použitých komponent pro návrh 2	39
Obrázek 31 Zjednodušené schéma rozložení použitých komponent pro návrh 3	40
Obrázek 32 Schéma zapojení na modelu vytvořeného programem Fritzing	44
Obrázek 33 Schéma měření rychlosti otáček	45
Obrázek 34 Schéma regulované jízdy na určitou vzdálenost	46
Obrázek 35 Průběh sekvence jízdy při měření	47
Obrázek 36 Hlavní smyčka programu	48
Obrázek 37 Uživatelské prostředí	48

Obsah příloženého CD

1. Textová část bakalářské práce
2. Kód pro Arduino
3. Kód pro Raspberry PI

PŘÍLOHA I.

```
//vlozeni knihoven
#include "TimerOne.h"
#include <SoftwareSerial.h>
#include <PID_v1.h>

//definice nazvu pinu
#define In1 8
#define In2 7
#define EnA 6
#define RX 11
#define TX 10

//definice bluetooth
SoftwareSerial bluetooth(TX, RX);

//definice promennych
double rotation = 0, Setpoint, Input, Output;
unsigned int counter=0, distance=0;
int i, cislo, distForward, distReverse, SetSpeed;
byte BluetoothData;
String vstup, strSpeed, strDistF, strDistB;
char strVstup, input;

//specifikace PID a nastaveni konstant bluetooth
PID myPID(&Input, &Output, &Setpoint,4,0.01,0, DIRECT);

//funkce k pocitani impulzu z otackomeru
void docount()
{
    counter++;
}
```

```
distance++;
```

```
}
```

```
//funkce k vypoctu otacek za minutu
```

```
void timerIsr()
```

```
{
```

```
rotation = (counter*60*2.0/ 36.0); // vypočítá otáčky
```

```
counter=0;
```

```
}
```

```
//nastaví piny na h-mustku pro jizdu vpřed
```

```
void carForward()
```

```
{
```

```
digitalWrite(In1, HIGH);
```

```
digitalWrite(In2, LOW);
```

```
}
```

```
//nastaví piny na h-mústku pro jízdu zpět
```

```
void carReverse()
```

```
{
```

```
digitalWrite(In1, LOW);
```

```
digitalWrite(In2, HIGH);
```

```
}
```

```
//zastaví motor
```

```
void carStop()
```

```
{
```

```
digitalWrite(In1, LOW);
```

```
digitalWrite(In2, LOW);
```

```
}
```



```

//funkce pro jizdu na urcitou vydalenost
void carGo(int s, int d)
{
    distance=0;
    while (distance < d)
    {
        Setpoint = SetSpeed;
        Input = rotation;
        myPID.Compute();
        delay(200);
        byte Reg = Output;
        //Serial.print("Výstup Reg: ");
        //Serial.println(Reg, DEC);
        analogWrite(EnA,Reg);
    }
    carStop();
    distance = 0;
}

//sekvence pro laboratorni mereni
void start()
{counter=0;
carReverse();
carGo(40, distReverse);
delay(5000);
carForward();
carGo(SetSpeed, distForward);
bluetooth.println("Measurement completed!");
}

//prijimani retezce s cisle

```

```

void getBTnumber()
{ delay(300);
  vstup="";
  strSpeed = "";
  strDistF = "";
  strDistB = "";
  while (bluetooth.available()){
    //strVstup = input;
    input = bluetooth.read();
    //bluetooth.println(input);
    vstup += String(input);
    delay(500);
  }

  strSpeed = String(vstup[0]) + String(vstup[1]);
  SetSpeed = strSpeed.toInt();
  delay(1000);
  //bluetooth.println(SetSpeed);

  strDistF = String(vstup[2]) + String(vstup[3]);
  distForward = strDistF.toInt();
  delay(1000);
  //bluetooth.println(distForward);

  strDistB = String(vstup[4]) + String(vstup[5]);
  distReverse = strDistB.toInt();
  delay(1000);
  //bluetooth.println(distReverse);

  if (SetSpeed != 0 and distForward != 0)
  {

```

```
start();  
}  
else{  
  bluetooth.println("Error! Model didn't get values!");  
}  
  
}
```

```
//funkce pro ovladani pomoci bluetooth  
void getBT(){  
  if (bluetooth.available() > 0) {  
    // načtení prvního znaku ve frontě do proměnné  
    BluetoothData=bluetooth.read();  
    // dekódování přijatého znaku  
    switch (BluetoothData) {  
      // každý case obsahuje dekódování jednoho znaku  
  
      case 'p':  
        // zastavení auta  
        carStop();  
        analogWrite(EnA, 0);  
        break;  
  
      case 'f':  
        // auto vpřed neregulovanou rychlostí  
        carForward();  
        analogWrite(EnA, 200);  
        break;  
  
      case 'r':  
        // v případě přejetí znaku 'a' vypíšeme
```

```
        carReverse();
        analogWrite(EnA, 200);
    break;

    case 's':
        // načtení rychlost
        getBTnumber();
        //bluetooth.println(cislo);
        //bluetooth.println();
    break;

    case 'k':
        // odeslani nastavenych hodnot pro kontrolu v raspberry
        delay(300);
        bluetooth.println(SetSpeed);
        bluetooth.println(distForward);
        bluetooth.println(distReverse);
        //bluetooth.println();
    break;

//default:
    // v případě přijetí ostatních znaků
    // vytiskneme informaci o neznámé zprávě
    //bluetooth.println("e");
}
}
}

void setup()
{
    Setpoint =0;
```

```
myPID.SetMode(AUTOMATIC);
myPID.SetOutputLimits(140, 255);

pinMode(In1, OUTPUT);
pinMode(In2, OUTPUT);
pinMode(EnA, OUTPUT);

//Serial.begin(9600);

Timer1.initialize(500000); //
attachInterrupt(0, docount, RISING); // increase counter when speed
sensor pin goes High
Timer1.attachInterrupt( timerIsr ); // enable the timer

bluetooth.begin(9600);
bluetooth.println("Model ready!");

}

void loop()
{
  getBT();
}
```

PŘÍLOHA II.

```
#!/usr/bin/python3

import os

from PySide2 import QtCore, QtWidgets, QtGuiTools

import pygatt

#import time

class MainWindow(QtWidgets.QMainWindow):

    def __init__(self, parent=None):

        QtWidgets.QMainWindow.__init__(self, parent)

        loader = QtGuiTools.QUiLoader()

        file = QtCore.QFile(os.path.abspath("main.ui"))

        file.open(QtCore.QFile.ReadOnly)

        self.ui = loader.load(file, parent)

        file.close()

        self.setCentralWidget(self.ui)

        self.message = ""

        self.def__speed = 0.5

        self.def__for = 65

        self.def__back = 60

        self.run__button = self.findChild(QtWidgets.QPushButton, 'Run')

        self.reset__button = self.findChild(QtWidgets.QPushButton, 'Reset')

        self.go__forward = self.findChild(QtWidgets.QPushButton, 'Forward')
```

```

        self.go_backward=self.findChild(QtWidgets.QPushButton,
'Backward')

        self.backward_dist=self.findChild(QtWidgets.QSpinBox,
'BackDistance')

        self.forward_dist=                self.findChild(QtWidgets.QSpinBox,
'ForDistance')

        self.speed = self.findChild(QtWidgets.QDoubleSpinBox, 'Speed')

        self.status = self.findChild(QtWidgets.QLabel, 'connection_label')

        self.run_button.clicked.connect(self.run)
        self.reset_button.clicked.connect(self.reset)
        self.go_forward.pressed.connect(self.Forward)
        self.go_forward.released.connect(self.Stop)
        self.go_backward.pressed.connect(self.Backward)
        self.go_backward.released.connect(self.Stop)

        self.adapter = pygatt.GATTToolBackend()
        self.adapter.start()
        self.bluetooth = None

        self.setStyleSheet("""
            QSpinBox::down-button {
                subcontrol-origin: margin;
                subcontrol-position: center left;
                height: 42px;

```

```
        width: 42px;
    }

    QSpinBox::up-button {
        subcontrol-origin: margin;
        subcontrol-position: center right;
        height: 42px;
        width: 42px;
    }

    QDoubleSpinBox::down-button {
        subcontrol-origin: margin;
        subcontrol-position: center left;
        height: 42px;
        width: 42px;
    }

    QDoubleSpinBox::up-button {
        subcontrol-origin: margin;
        subcontrol-position: center right;
        height: 42px;
        width: 42px;
    }
    "" )
```

```
self.showFullScreen()
```

```
self.show()
```

```
self.bt_status()
```

```
self.show_message()
```



```

self.reset()

def run(self):
    print(self.backward_dist.value())
    print(self.forward_dist.value())
    print(self.speed.value())
    self.setEnabled(False)
    self.bluetooth.char_write_handle(0x012, bytes('0', 'ascii'))
    self.bluetooth.char_write_handle(0x012,
bytes("s{}{}".format(int(self.speed.value()*115), int(self.forward_dist.value()),
int(self.backward_dist.value())), 'ascii'))

def reset(self):
    self.backward_dist.setValue(self.def_back)
    self.forward_dist.setValue(self.def_for)
    self.speed.setValue(self.def_speed)

def Forward(self):
    self.bluetooth.char_write_handle(0x012, bytes('0', 'ascii'))
    self.bluetooth.char_write_handle(0x012, bytes('f', 'ascii'))

def Backward(self):
    self.bluetooth.char_write_handle(0x012, bytes('0', 'ascii'))
    self.bluetooth.char_write_handle(0x012, bytes('r', 'ascii'))

def Stop(self):
    self.bluetooth.char_write_handle(0x012, bytes('0', 'ascii'))
    self.bluetooth.char_write_handle(0x012, bytes('p', 'ascii'))

```

```
def bt__status(self):
    if not self.bluetooth:
        try:
            self.bluetooth = self.adapter.connect('C8:FD:19:37:18:37')
            self.bluetooth.subscribe('0000ffe1-0000-1000-8000-
00805f9b34fb', self.get__message)
        except:
            self.bluetooth = None
    if self.bluetooth:
        self.status.setText('Bluetooth connected')
        self.status.setStyleSheet('color: green')
        self.run__button.setEnabled(True)
        self.go__forward.setEnabled(True)
        self.go__backward.setEnabled(True)
    else:
        self.status.setText('Bluetooth disconnected')
        self.status.setStyleSheet('color: red')
        self.run__button.setEnabled(False)
        self.go__forward.setEnabled(False)
        self.go__backward.setEnabled(False)
    QtCore.QTimer.singleShot(500, self.bt__status)
def get__message(self, handle, msg):
    self.message = self.message + msg.decode('latin1')
```

```
def show__message(self):  
    if self.message:  
        box = QtWidgets.QMessageBox()  
        box.setText(self.message)  
        self.message = ""  
        box.exec()  
        self.setEnabled(True)  
        QtCore.QTimer.singleShot(500, self.show__message)
```

```
if __name__ == '__main__':  
    import sys  
    app = QtWidgets.QApplication(sys.argv)  
    mainwindow = MainWindow()  
    sys.exit(app.exec_())
```