



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

DEPARTMENT OF COMPUTER SYSTEMS

**AUTONOMNÍ ŘÍZENÍ AUTÍČKA S ADAPTACÍ NA
PŘEDEM NEZNÁMÝ TVAR AUTODRÁHY**

AUTONOMOUS SLOT-CAR DRIVING WITH ADAPTATION TO UNKNOWN TRACK SHAPE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MICHAL VAŠUT

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JOSEF STRNADEL, Ph.D.

BRNO 2022

Zadání bakalářské práce



Student: **Vašut Michal**
Program: Informační technologie
Název: **Autonomní řízení autíčka s adaptací na předem neznámý tvar autodráhy**
Autonomous Slot-Car Driving with Adaptation to Unknown Track Shape
Kategorie: Vestavěné systémy

Zadání:

1. Seznamte se s vlastnostmi dostupných autodráh. Zvolte autodráhu od konkrétního výrobce, s níž budete dále pracovat.
2. Z ročníků 2009-2012 soutěže Freescale Race Challenge zdokumentujte desky plošných spojů (DPS) navržené pro autonomní řízení autíček pomocí mikrokontrolérů (MCU). Popište principy související s řízením motoru a zpracováním dat z akcelerometru či dalších čidel využitelných pro řízení.
3. Zvolte, upravte či doplňte existující DPS (FRDM, ESP32/8266, Arduino atp.), popř. navrhnete a vyrobte vlastní DPS, vhodnou pro autonomní řízení autíčka. Danou DPS oživte a demonstруйте její schopnost ovládat pomocí MCU rychlost autíčka a rozpoznat rovinku a zatáčku.
4. Navrhnete datovou reprezentaci tvaru dráhy pro uložení v paměti přístupné z MCU.
5. Navrhnete a v MCU realizujete i) algoritmus autonomního mapování tvaru předem neznámé dráhy, ii) algoritmus schopný využít informaci o tvaru dráhy k projetí dráhy v co nejkratším čase.
6. Funkčnost algoritmů vhodně ověřte, demonstруйте a zhodnoťte.
7. Shrňte dosažené výsledky a diskutujte možné směry dalšího vývoje předloženého řešení.

Literatura:

- Dle pokynů vedoucího.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Strnadel Josef, Ing., Ph.D.**
Vedoucí ústavu: Sekanina Lukáš, prof. Ing., Ph.D.
Datum zadání: 1. listopadu 2021
Datum odevzdání: 11. května 2022
Datum schválení: 29. října 2021

Abstrakt

Tato bakalářská práce se zabývá návrhem algoritmů pro mapování neznámého tvaru auto-dráhy, jejich uložení a použití pro projetí dráhy v co nejkratším čase, a jejich implementací. Algoritmus použije uložená data ze senzorů na určení rychlosti, aby autíčko nevytáhlo ze tratí.

Abstract

This bachelor thesis deals with the design of algorithms for mapping the unknown shape of the racing track, their storage and usage for passing the track in the shortest possible time, and their implementation. The algorithm uses stored data from the sensors to determine speed so that the car does not fall off the track.

Klíčová slova

autonomní řízení, adaptace na neznámý tvar, freescale race challenge, arduino, gyroskop, akcelerometr, magnetometr

Keywords

autonomous driving, adaptation to unknown shape, freescale race challenge, arduino, gyroscope, accelerometer, magnetometer

Citace

VAŠUT, Michal. *Autonomní řízení autíčka s adaptací na předem neznámý tvar autodráhy*. Brno, 2022. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Josef Strnadel, Ph.D.

Autonomní řízení autíčka s adaptací na předem neznámý tvar autodráhy

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Josefa Strnadela Ph.D. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Michal Vašut
8. května 2022

Poděkování

Chtěl bych poděkovat panu Ing. Josefu Strnadelovi, Ph.D. za vedení mé práce, cenné rady a odborný dohled.

Obsah

1	Úvod	3
2	Rešerše k řešenému tématu	4
2.1	Vlastnosti dostupných autodráh	4
2.2	Freescale Race Challenge	7
2.3	Principy související s řízením motoru a zpracováním dat ze senzorů	10
2.3.1	Elektromotor	10
2.3.2	Řízení motoru ve vybraném autíčku	12
2.3.3	Senzory	13
2.4	Chyby měření	18
2.5	Fyzikální jevy	19
3	Rozbor řešené problematiky a realizačních možností	21
3.1	Rozbor problémů	21
3.1.1	Rozpoznání ujetého kola	21
3.2	Podmínky pro výběr součástí	22
3.3	Výběr komponent	22
4	Návrh a realizace řešení	25
4.1	Zapojení	25
4.1.1	Zapojení slotu pro microSD kartu	25
4.2	Data ze senzorů a jejich využití	26
4.2.1	Gyroskop	26
4.2.2	Akcelerometr	27
4.3	Hallova sonda	28
4.4	Ovládání motoru	29
4.5	Datová reprezentace tvaru autodráhy	30
4.6	Algoritmy	31
4.6.1	Algoritmus pro mapování autodráhy	31
4.6.2	Algoritmus pro jízdu po zmapované dráze	32
4.6.3	Implementace	32
4.6.4	Další funkce a knihovny	32
4.7	Popis realizace	34
4.7.1	Zapojení	35
5	Zhodnocení řešení	36
5.1	Problémy s řešením	36
5.1.1	Záznam výsledků	36

5.1.2	Měření rychlosti	37
5.1.3	Použití částí programu	38
6	Závěr	41
6.1	Další vývoj	41
6.1.1	Program	41
6.1.2	Platforma	42
	Literatura	43
A	Obsah přiloženého média	46

Kapitola 1

Úvod

Autonomní řízení vozidel je velkým tématem dnešního světa, proto je zkoumání principů a experimentování s modely aut důležitou součástí jejich vývoje.

Zabývá se návrhem zapojení a tvorbě algoritmů pro zmapování neznámého tvaru autodráhy, jeho uložení, a použití těchto dat k dosažení co nejrychlejšího času při projíždění dráhy. Práce rozvíjí zadání soutěže Freescale Race Challenge, kde byli tvořeny podobné algoritmy a desky plošných spojů.

První projetí zmapuje autodráhu pomocí senzorů. Při druhém projetí budou již známy co nejpřesnější data o dráze a podle nich se zrychlí auto tak, aby projelo co nejvyšší rychlostí, přitom nevypadlo, ve všech místech dané dráhy.

Cílem tedy je vybrat vhodnou elektroniku, návrh zapojení, návrh a implementace algoritmů pro snímání tvaru dráhy a použití dat na projetí dráhy.

V práci bude použita vývojová platforma arduino, protože se dá rychle implementovat a její programovací knihovny jí dovolují snadno pracovat s dalším externím vybavením (tzn. připojení k PC, moduly a podobné). Musí se použít malá deska, která má již implementované integrované obvody s některými senzory, kvůli velikosti autíčka.

Kapitola 2

Rešerše k řešenému tématu

V této kapitole jsou popsány známé údaje o autodráze a autíček. Zejména se tedy mluví o autodráze z Freescale Race Challenge a jsou zde popsány jednotlivé její ročníky z roku 2009 – 2012. Poté zde je popsán způsob řízení motoru autíčka a případné senzory, které by se mohli použít při řešení této práce.

2.1 Vlastnosti dostupných autodrah

Autodráha je modulární stavebnice. Je to populární hračka jak mezi dětmi tak i dospělými. Její hlavní vlastností je ovládání rychlosti autíčka, které jezdí po dráze. Dráha je sestavená z různých dílů, které tvoří různé tvary a jejich start i cíl je vždy v jednom místě. Autíčko se drží v určité dráze pomocí drážky, díky které i zatáčí. U drážky jsou i kartáčky, přes které se sbírá elektřina. V nejjednodušším případě se rychlost autíčka ovládá pouze přes změnu napětí v určité dráze. To je dosaženo většinou ovladači připojenými ke startu. Velikosti autodrah i autíček jsou dostupné v měřítku 1:18, 1:24, 1:32, 1:43, 1:55 nebo 1:64.



Obrázek 2.1: Velikosti aut pro srovnání, která nabízí firma Carrera [5]

Ovládání

Analogové autodráhy

Starší autodráhy mění pouze napětí dráhy pomocí ovladačů. Ovladače mění napětí v dráze pomocí potenciometru. Mají pouze jedno tlačítko, kterým se ovládá napětí určité dráhy.

Na takové autodráze je možnost pouze jednoho autíčka na jedné dráze. A její ovládání a možnosti v používání různých dílů je omezeno.

Digitální autodráhy

Při tomto druhu ovládání má každé autíčko dekodér, kterým přijímá signály od ovladače. Signály jsou přenášeny buď po dráze nebo bezdrátově. Tímto způsobem autíčka nejsou závislá na dráze do které byli umístěny a můžou být použity například díly pro křížení drah. V jedné dráze může jet i více autíček.

Napájení

Analogové autodráhy

U starších autodrah ovladač reguluje velikost napětí na dráze. Čím větší napětí tím je autíčko rychlejší. Používá se tedy stejnosměrné napětí. Většinou cca 12 V – 16 V.

Digitální autodráhy

Novější autodráhy používají konstantní napětí, které se nejčastěji pohybuje okolo 14.8 V. Rychlost autíček se neovládá velikostí napětí, proto může v jedné dráze jet více autíček. Autodráha má díl, který slouží k napájení celé autodráhy a zároveň jako její start a cíl. Do napájecího dílu se nejčastěji zapojuje transformátor.

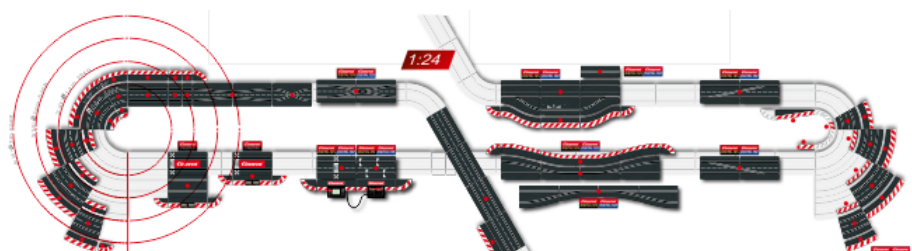
Díly autodráhy

Autodráhy nejčastěji obsahují sadu složenou z rovných dílů různých délek a zatáček různých stupňů záhybu což je základ každé autodráhy, aby se dal vytvořit platný okruh. Další díly jsou například zúžení, výjezd na vyvýšenou plošinu, klopené zatáčky a další podobné díly.

Každý druh dílů existuje v různých velikostech. U většiny autodrah se dá vytvořit nejrůznější tvary autodráhy. Jediné co se musí dodržet je, že vždy bude začínat a končit v jednom místě, to znamená že bude tvořit plnohodnotný okruh.

Díly se navzájem propojují pomocí výstupků, které při zapadnutí do jiného dílu zařizují to, aby se díly nerozpojily a zároveň je zde vodivý výstupek pro propojení napájení dílů. Některé autodráhy obsahují i malé plastové dílky, které je možno vložit do dílů a posilnit tím celkovou konstrukci.

Existují i jiné podpůrné části jako jsou například mantinely, které se dají většinou připevnit na okraj prudkých zatáček.



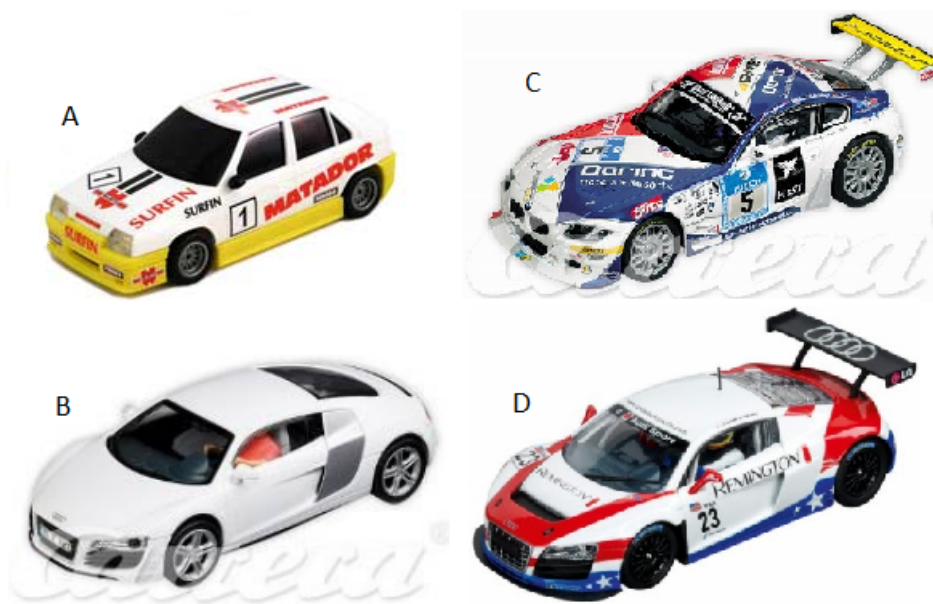
Obrázek 2.2: Díly, které nabízí firma Carrera [5]

Autíčka

Autíčka stejně jako díly autodráhy přichází v různém tvarech a velikostech. Existují modely od nereálných aut až po modely, které jsou postaveny na základě reálných modelů aut akorát v menším měřítku.

Modely

Použitý model je Audi R8 GT LMS United Autosports No.23



Obrázek 2.3: A: Faro FAVORIT pro FRC 2009
B: Carrera Audi R8 pro FRC 2010
C: BMW Z4M Coupe pro FRC 2011
D: Audi R8 GT LMS United Autosports No.23

Autíčka mají kontaktní kartáčky, které jsou potřebná pro sbírání elektřiny z dráhy. Při jízdě tyto kartáčky musí být na vodivé ploše dráhy. Každé autíčko musí mít minimálně dva takové kartáčky, ale některé modernější mají čtyři, aby se co nejvíce omezilo přerušení napájení. Kartáčky jsou většinou připevněny na otočné drážce, která udržuje auto na dráze.

Další důležitý faktor je elektrický motor autíčka.

Analogové

Analogové autíčka mají většinou pouze připojeny kartáčky přímo k motoru.

Digitální

Digitální autíčka ještě obsahují řídicí elektroniku, která ovládá rychlost motoru a přijímá data od určitého ovladače.

2.2 Freescale Race Challenge

Freescale Semiconductor ČR[25], s podporou Faro - česká autodráha[9], pořádali pro studenty elektrotechnických vysokých škol u nás, na Slovensku a v Polsku soutěž. Úkolem zúčastněných studentů je postavit auto na autodráhu, které zajede nejkratší čas na 10+10 okruhů na neznámé trati pomocí algoritmu, který bude ovládat rychlost autíčka.[10]

Pravidla závodění popisují, že závodí každý zvlášť na čas. V prvním kole se jezdí 10 okruhů v pravé dráze, ve druhém kole 10 okruhů v levé dráze. Součet časů obou kol určuje výsledné pořadí. Přítlačné magnety, stejně jako bezdrátová dálková ovládání, jsou zakázány.

Pravidla byla jednoduchá. Tvar dráhy byl předem neznámý, soutěžící pouze věděli, že délka bude v rozsahu 10 až 16 metrů a trať bude poskládána ze specifikovaných dílů. Autíčka nesměla obsahovat magnety ani dálkové ovládání a nemohla být upravena tak, aby cokoliv přesahovalo vnější rozměry karoserie.

Před závodem dostaly týmy novou sadu pneumatik a autíčko mělo jeden seznamovací průjezd, při kterém mělo pomocí dat z akcelerometru zjistit tvar dráhy.

Řízení auta, které jede ve vodící drážce a stačí jej ovládat pouze změnami napětí na motoru. K ovládání autíčka stačí poměrně jednoduchý procesor, který zvládne každý. K úspěchu v soutěži je však potřeba také znát alespoň základy zpracování signálů, a hlavně prokázat dobré analytické myšlení. Způsobů, jak extrahovat informace o neznámé trati, a jak je použít pro řízení auta.

Soutěž v roce 2009

Informace o této soutěži z roku 2009 byli čerpány ze článků o soutěži. [10]

Model Auta

Jako model autíčka bylo vybráno standardní auto na autodráhu Faro FAVORIT 2.3

Napájení

Napájení na této soutěži bylo stabilní napětí 12V.

Mikrokontroler

V soutěži byl použit 8bitový mikrokontroler S08JM32[19], který pracuje na frekvenci 48 MHz. Na čipu obsahuje 60 kB programovatelné flash paměti, 4 kB RAM a 256 B USB RAM. Obsahuje podporu pro periferní zařízení USB, ADC, ACMP, SCI, SPI, IIC, KBI a RTC. Použitý 8bitový procesor S08JM patří do rodiny Flexis a existuje k němu „větší dvojče“ MCF51JM – ve stejném pouzdře, se stejným obsazením pinů a se stejnými periferiemi, ale s 32bitovým jádrem ColdFire a větším paměťovým prostorem.

Akcelerometr

V soutěži byl použit kapacitní akcelerometr MMA7361[22], který měří sílu zrychlení v jednotce g a může měřit ve třech rovinách. Umožňuje volbu dvou úrovní senzitivity $\pm 1.5g$, $\pm 6g$. Napájecí napětí je 2.2 V – 3.6 V.

Akcelerometr byl používán pro snímání zrychlení působících na autíčko a jako hlavní zdroj informací o jízdě autíčka.

H-můstek

V soutěži byl použit H-můstek MCZ33887[21], pro ovládání motoru. Pracuje na napětí 5 V – 28 V. Používá se pro stejnosměrné motory.

Paměť

V soutěži byla použita 24AA512[1], je kompatibilní s IIC. Pracuje na napětí 1.7 V – 5.5 V a hodinovou frekvencí buď 100 kHz nebo 400 kHz. Paměť je 64K x 8 (512 Kbit) sériová elektricky mazatelná PROM (EEPROM).

Soutěž v roce 2010

Informace o této soutěži z roku 2009 byli čerpány z článků o soutěži[11]. Zde jsou uvedeny pouze změny od minulého ročníku.

Tento rok se stejná soutěž pořádala i v Rumunsku. S tímto rozšířením souvisela i změna autodráhy. Česká autodráha FARO, která skvěle posloužila v prvním ročníku soutěže, je dostupná pouze v ČR a SR. Ze značek, které jsou rozšířené po celém světě, vyšla vstříc Carrera, a z jejich produktů byla vybrána dráha Carrera Evolution.

Model Auta

Jako model autíčka bylo vybráno auto Carrera Audi R8 2.3. Napájení se v tomto ročníku změnilo na 14.8V.

Mikrokontroler

Tento ročník byl použit, již minule zmiňovaný, mikrokontroler MCF51JM64[20], který obsahuje na čipu 64 kB programovatelné flash paměti, 6 kB RAM. Jinak má stejné pouzdro, se stejným obsazením pinů a stejné periferie jako předchozí mikrokontroler S08JM32 a má 32 bitové jádro ColdFire. K mikrokontroleru byl navíc přidán externí krystal o frekvenci 8 MHz.

Akcelerometr

Akcelerometr byl použit stejný jako minulý ročník.

H-můstek

Stejně tak H-můstek zůstal stejný jako minulý ročník.

Paměť

Na rozdíl od minulého ročníku se paměť vyměnila za slot pro microSD kartu, což zjednodušuje přenos za jízdy zaznamenaných dat do PC a překonává omezení množství ukládaných dat.

Soutěž v roce 2011

Informace o této soutěži z roku 2009 byli čerpány z článků o soutěži[12]. Tento ročník proběhla menší změna v pravidlech a to, že vyřazovací turnaj se pojede nikoli na neznámém

okruhu, ale na předem známém, poté každý tým obdržel již osazenou desku součástkami. Zde jsou uvedeny pouze změny od minulého ročníku. V roce 2011 se také rozhodlo, že studenti dostanou již osazenou desku, nikoliv plošný spoj a sáček se součástkami. To znamenalo ušetření času na vývoj algoritmu spíše než trávit čas osazováním. Byla zde, ale i možnost vytvořit si vlastní desku nebo dodanou desku jakkoliv rozšířit.

Model Auta

Jako model autíčka byl tento ročník vybráno autíčko BMW Z4M Coupe 2.3.

Napájení

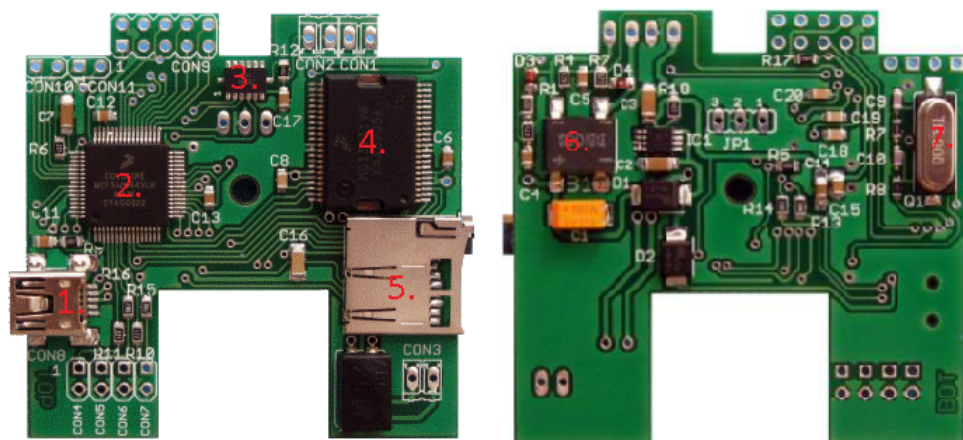
Napájení v tomto ročníku se změnilo na 15V.

Soutěž v roce 2012

Soutěž v tomto ročníku se ve své podstatě nezměnila. Jediná změna v pravidlech byla, že auto využívající kameru mohou mít hmotnost až 145g na rozdíl od minulých 125g.

Závěr Freescale Race Challenge

Po ročnících FRC 2009 - 2012, byli sbírány připomínky, zkušenosti a nápady a doladila se soutěž k dokonalosti. Vývoj probíhal postupně. V ročníku 2010 byl už udělán finální vzhled desky a v ročníku 2011 se ještě doladily součástky a propojení až do výsledného vzhledu.



Obrázek 2.4: Finální vzhled desky tj. vzhled desky z FRC 2011

Popis součástek

Na obrázku 2.4 můžeme vidět microUSB konektor (1.), mikrokontroler MCF51JM64 (2.)[20], akcelerometr MMA7361 (3.)[22], H-můstek MC33931(4.)[21], microSD card slot (5.)[2], usměrňovací můstek DB106S (6.)[6], externí krystal s frekvencí 8 MHz (7.).

2.3 Principy související s řízením motoru a zpracováním dat ze senzorů

2.3.1 Elektromotor

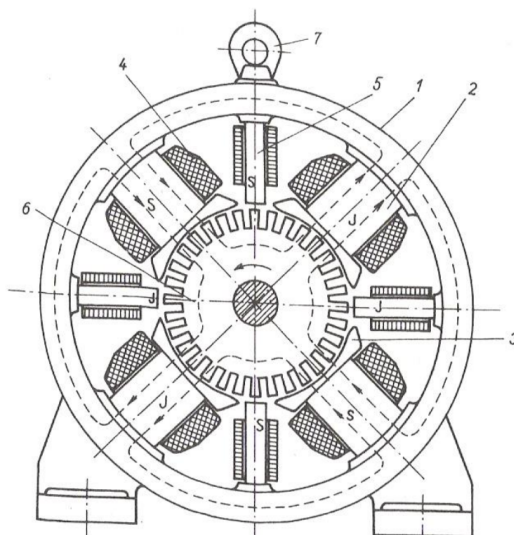
V této části jsou některé informace převzaty z [7] a [8].

Princip elektromotoru

Elektromotory jsou stroje, které přeměňují elektrickou energii na energii mechanickou. Jejich účinnost se pohybuje mezi 75 — 90 %. Každý elektromotor se skládá z nepohyblivé části zvané stator (tj. kotva) a z pohyblivé části zvané rotor, která se otáčí uvnitř statoru.

Stejnoseměrné elektromotory

Činnost stejnosměrných motorů se zakládá na silovém působení, účinku magnetického pole na vodič, kterým protéká proud. Průchodem elektrického proudu vinutím statoru a vinutím rotoru se vytváří dvě magnetická pole, která na sebe vzájemně působí přitažlivými a odpuzivými silami tak, že se rotor otáčí. Vzájemné silové účinky pole uvedou kotvu do pohybu.



Obrázek 2.5: Konstrukce stejnosměrného motoru – příčný řez.

1 — stator, 2 — pólové jádro, 3 — pólový nástavec, 4 – budící cívka, 5 — pomocný pól, 6 — rotor

Kdyby se kotva dostala do neutrální polohy, měl by její počáteční pohyb ustát. Kotva se však svojí setrvačností pootáčí déle, až v určitém okamžiku komutátor přivede do vinutí kotvy znovu proud a kotva se pak nuceně pohybuje dál, ale její pohyb je nerovnoměrný.

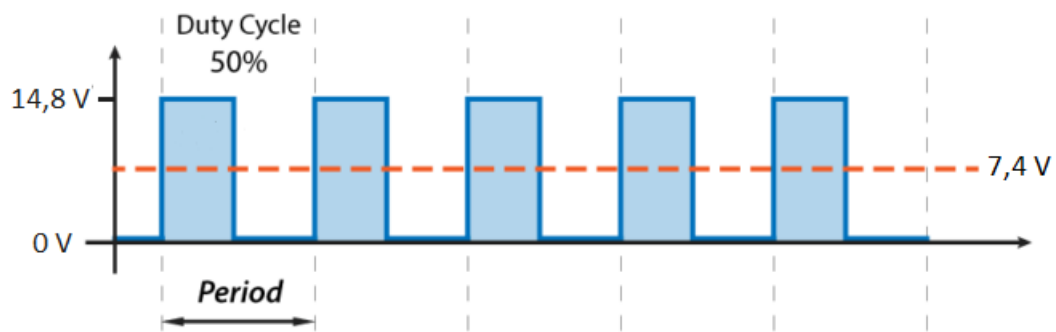
Aby se vždy stejnosměrný elektromotor rozběhl v každé poloze kotvy, dává se na rotor více cívek, nejméně tři. Jestliže má rotor větší počet cívek rovnoměrně rozložených po obvodu kotvy, získává tím větší točivý moment a rovnoměrný otáčivý pohyb.

Podle způsobu zapojení vinutí statoru a rotoru rozeznáváme stejnosměrný sériový motor (vinutí statoru a rotoru je zapojeno do série) a stejnosměrný derivační motor (vinutí statoru

a rotoru je zapojeno paralelně). Důležitou vlastností sériového stejnosměrného elektromotoru je, že mění své otáčky podle zatížení, přičemž čím menší má motor otáčky, tím větší má výkon. Na rozdíl od sériového stejnosměrného elektromotoru, tak derivační stejnosměrný motor své otáčky s rostoucím zatížením téměř nemění.

Kontrola otáček se nejčastěji provádí pomocí pulzně šířkové modulace.

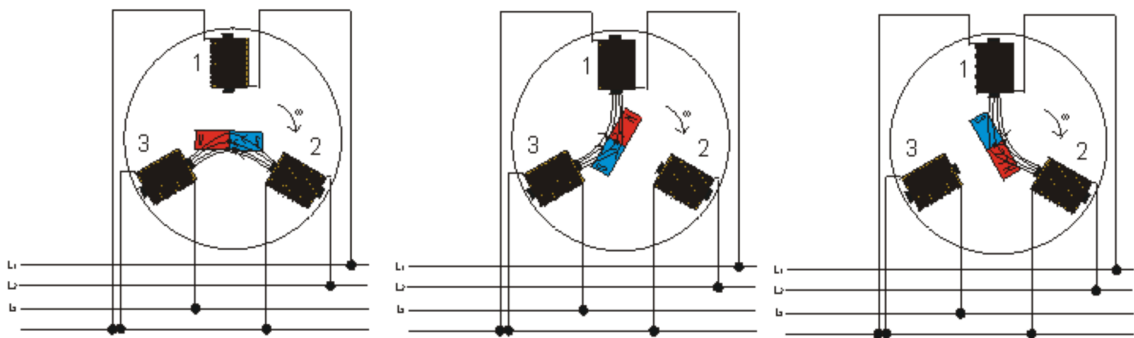
Controlling DC motor with PWM



Obrázek 2.6: Kontrola rychlosti stejnosměrných motorů pomocí pulzně šířkové modulace

Střídavé elektromotory

Stator tvoří tři elektromagnety, které jsou navzájem posunuty o 120° . Každým vinutím elektromagnetu protéká proud vždy jen jedné fáze třífázového proudu. Jestliže do prostoru mezi elektromagnety vložíme magnetku, magnetka se bude otáčet.



Obrázek 2.7: Zapojení a funkce střídavého synchronního elektromotoru

Střídavé elektromotory se rozdělují na synchronní a asynchronní.

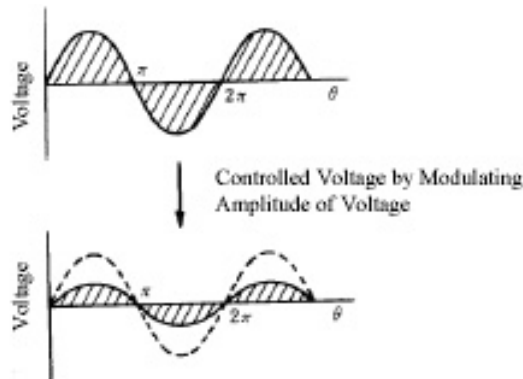
U synchronních elektromotorů se magnetka otáčí se stejnou frekvencí jako je frekvence střídavého proudu. Vinutí statoru je napájeno střídavým proudem, vinutí rotoru je napájeno stejnosměrným proudem, který získáváme ze zvláštního derivačního dynama. Otáčky synchronního motoru jsou stálé a nekolísají při zatížení. Nevýhodou tohoto elektromotoru je, že při větším zatížení se pohyb magnetky zastaví. Výhodou je pak větší účinnost a přesnost otáček.

Asynchronní elektromotory patří mezi nejpoužívanější elektromotory. Velkým rozdílem od synchronního a stejnosměrných elektromotorů je ten, že se do rotoru nepřivádí žádný

proud a střídavý proud prochází jen vinutím statoru. Točivé magnetické pole statoru indukuje ve vinutí kotvy značné proudy, takové že podle Lenzova zákona vznikají magnetické síly působí na vodiče kotvy a ta se roztočí.

Rotor má určitý „skluz“. „Skluz“ říkáme tomu, když se rotor nemůže otáčet stejně rychle jako je frekvence pole. Otáčky rotoru se postupně zvyšují do určitého okamžiku. Rotor neběží shodně s točivým magnetickým polem. Jeho otáčky kolísají podle zatížení z toho tedy asynchronní elektromotor.

Střídavé motory mohou být kontrolovány mnoha způsoby. Jedním z těchto způsobů je změna velikosti napětí.



Obrázek 2.8: Kontrola motoru pomocí změny napětí

Krokové elektromotory

Jejich konstrukce je uzpůsobena tomu, aby se každé vinutí rotoru dalo použít zvlášť. Takto se docílí postupného posouvání rotoru. To znamená, že se otáčí nespojitě po určitých úhlech a že jej můžeme přesně polohovat. U tohoto druhu také můžeme udělat půl krok, což znamená že úhel každého kroku se vydělí dvěma. K tomu je potřeba střídající se napájení dvou fází. Ovladač krokového motoru je speciální elektronický obvod generující impulsy v určitém sledu a délce. Tyto impulsy následně přes výkonovou část budí v přesném pořadí jednotlivá vinutí rotoru. Frekvence, pořadí a délka impulsů z řídicího obvodu řídí počet otáček rotoru a rovněž točivý moment stroje.

2.3.2 Řízení motoru ve vybraném autíčku

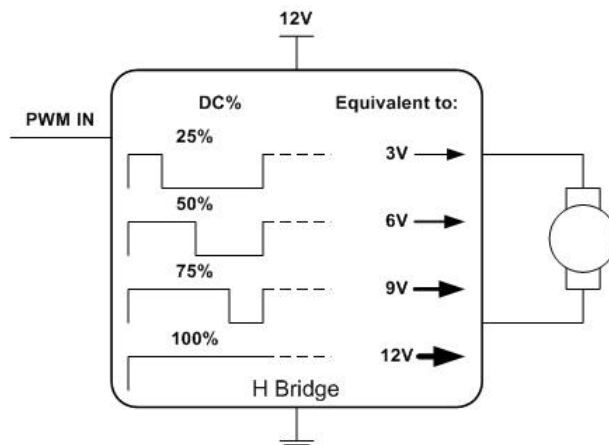
Ve vybraném autíčku 2.3 se používá stejnosměrný motor. Což znamená že auto musí být napájeno stejnosměrným napětím. Směr otáčení motoru závisí na polaritě napětí a otáčky na velikosti napětí. Dráha je tedy napájena transformátorem, který do dráhy pouští 14.8 V.

Pro ovládání motoru je tedy potřeba mít možnost velikost tohoto napětí měnit za jízdy autíčka.

První, nejjednodušší způsob je tedy snižovat a zvyšovat napětí, nicméně tento způsob regulace obvykle není vhodný z důvodů velkých tepelných ztrát.

H-můstek

Další způsob je řízení motoru pomocí regulace rychlosti přes PWM (pulzně šířková modulace), která odstraňuje nevýhodu spojenou s velkou tepelnou ztrátou. Pro řízení tímto způsobem se využívá H-můstek.



Obrázek 2.9: Řízení motoru regulací napětí pomocí H-můstku [26]

Nejčastěji pro řízení dvou stejnosměrných motorů [13]. Obvykle se tedy prodávají integrované obvody či moduly, které obsahují dva H-můstky.

2.3.3 Senzory

Akcelerometry

Informace v této sekci jsou čerpány z [3].

Akcelerometry jsou senzory měřící statické a dynamické zrychlení těles. Statické zrychlení vzniká působením gravitačního pole země. Mezi dynamická zrychlení patří akcelerace (vibrace) tělesa.

Akcelerometr je inerciální senzor, což znamená že vychází z fyzikálního popisu pro vztažnou soustavu. Jinými slovy, pokud senzor nemění svůj pohyb ,tj. setrvává v klidu nebo v rovnoměrném přímočarém pohybu, neměří žádnou hodnotu.

Princip lze vysvětlit na závaží (seismická hmota), které je zavěšeno na pružině. Pohyb závaží je přímo úměrný působícímu zrychlení na akcelerometr. Pro vyhodnocení zrychlení je nutné dráhu závaží vhodně převést například na elektrický signál.

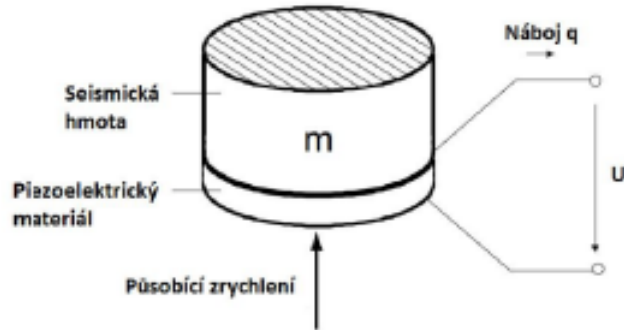
Akcelerometry můžeme rozlišovat například podle způsobu snímání zrychlení. Další dělení může být podle počtu snímaných os nebo typu výstupu (analogový, digitální). U analogových akcelerometrů je výstupní napětí spojitě a úměrné působícímu zrychlení. Digitální akcelerometry mají jako výstupní signál obdélkový průběh. Šířka pulzu je úměrná působícímu zesílení (PWM).

Existují různé druhy akcelerometrů jako piezoelektrické, piezorezistivní, tepelné a kapacitní.

Piezoelektrické akcelerometry

Piezoelektrický jev se projevuje vznikem elektrického náboje na plochách krystalu, který je mechanicky namáhán. Velikost a polarita vzniklého napětí závisí na velikosti a směru

působící deformace na krystal. Piezoelektrický akcelerometr se skládá ze seismické hmoty umístěné na piezoelektrický materiál. Celek je pak umístěn na pevnou podložku. Vyhodnocení zrychlení se provádí změřením vygenerovaného náboje na plochách piezoelektrického materiálu.



Obrázek 2.10: Princip piezoelektrického akcelerometru

Výhodou piezoelektrických akcelerometrů je nízká citlivost snímače na změny magnetického pole, rozsah pracovních teplot a vysoký rezonanční kmitočet. Piezoelektrické akcelerometry nemohou být použity pro měření konstantního zrychlení. Jsou použitelné pro měření kmitů v rozsahu několika Hz až desítek kHz a velmi vysokých hodnot dynamického zrychlení. Nevýhodou naopak může být nízká úroveň výstupního signálu, nelze aplikovat bez zesilovače a tedy bez napájecího zdroje.

Piezorezistivní akcelerometry

Akcelerometry jsou vyrobeny z piezorezistivního materiálu, který mění s mechanickým namáháním svůj elektrický odpor. Provedením piezorezistivního akcelerometru může být například hmota zavěšená na vetknutém nosníku. Působením zrychlení dochází k ohybu nosníku a ke změně velikosti piezorezistivního materiálu umístěného u kořene nosníku.

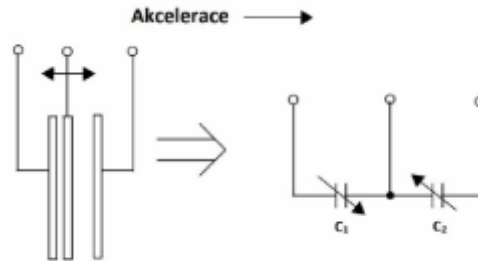


Obrázek 2.11: Princip piezorezistivního akcelerometru

Výhodou piezorezistivních akcelerometrů je možnost měření i statického zrychlení. Piezorezistivní akcelerometry se vyznačují citlivostí pro zrychlení menší než setina gravitačního zrychlení, velkou šířkou pásma, lineárním výstupem a teplotní stabilitou. Výhodou je i vysoká spolehlivost, malá velikost pouzdra a nízká cena.

Kapacitní akcelerometry

Základem kapacitních akcelerometrů je proměnný kondenzátor s pevnou a pružně zavěšenou pohyblivou (hmotnou) deskou. Působením zrychlení na akcelerometr dochází k vychýlení hmoty a pohyblivé desky kondenzátoru z rovnovážného stavu. Pohybem desek se mění vzdálenost desek a tím se mění kapacita kondenzátoru.



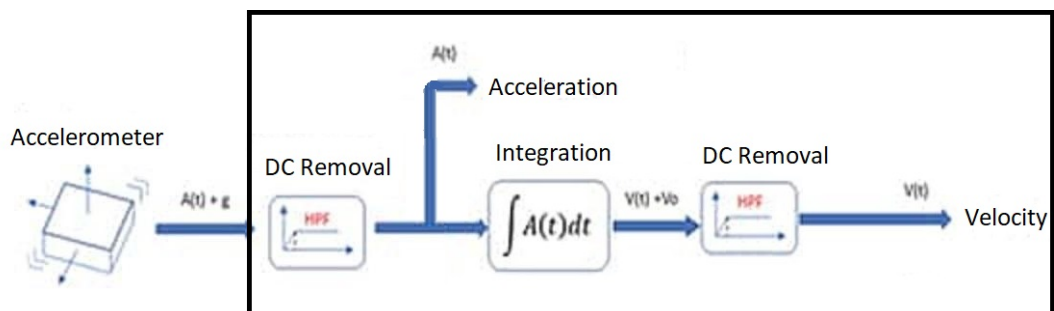
Obrázek 2.12: Princip kapacitního akcelerometru [3]

Setrvačná hmota kapacitního akcelerometru je k jeho pouzdru připevněna ohebnou pružinou vytvořenou z křemíkového substrátu. Na parametrech této pružiny závisí celková citlivost akcelerometru, ale také jeho frekvenční a dynamický rozsah. Vlivem změny teploty působící na snímač, dochází ke změně tuhosti a délky pružiny akcelerometru a to vede ke změně citlivosti. Tato citlivost je však z pravidla mnohem nižší než u piezorezistivních akcelerometrů, proto je kapacitní akcelerometry v současnosti nahrazují.

Příkladem kapacitních akcelerometrů jsou moderní senzory označované jako MEMS. Některé moderní akcelerometry mohou obsahovat korekční obvody, či zpětnovazební regulační smyčky pro potlačení teplotní závislosti. Takové MEMS akcelerometry jsou zpravidla dražší a méně dostupné než klasické

Zpracování dat z akcelerometrů

Výstupem akcelerometru je zrychlení tělesa v gravitačním zrychlení (g). Signály akcelerometru musí být nejprve filtrovány horní propustí, aby se odstranily jakékoli součásti stejnosměrného proudu, jako jsou například účinky gravitace. Tak zvaná nulová odchylka (zobrazuje se za klidového stavu na akcelerometru) se musí vypočítat v klidovém stavu, poté musí být odečtena od vzorků akcelerometru. Tímto dostaneme akceleraci.



Obrázek 2.13: Způsob zpracování dat z akcelerometru

To je pro ideální případ. Ale jelikož akcelerometry zobrazují i šum, který může nastat, tak je dobré použít před tímto (2.13) procesem nějaký datový filtr např. kalmanův filtr.

V závislosti na aplikaci se používají různé analytické techniky pro získání informace o stavu stroje. Jednou z nejpoužívanějších technik je výpočet střední rychlosti (tzv. RMS rychlost) vibrací a určení jejího průběhu v čase. Když se stroje opotřebovávají jejich vibrace se v čase zrychluje. Sledování střední rychlosti vibrace RMS v čase je tedy důležitým indikátorem opotřebení.

Gyroskopy

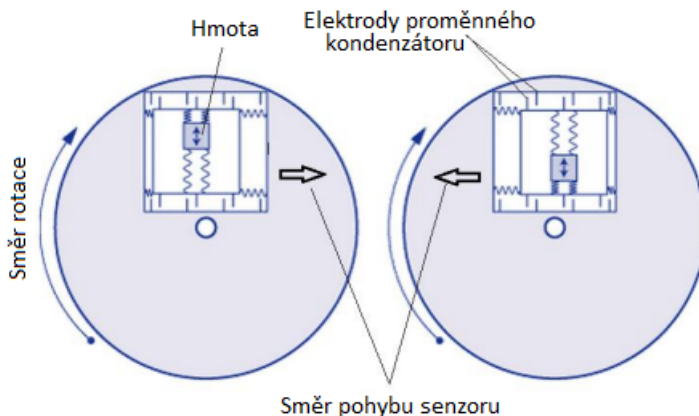
Informace v této sekci jsou čerpány z [3].

Gyroskop je schopen určit orientaci měřeného tělesa v prostoru. Základem gyroskopu je velkou rychlostí rotující setrvačník, který zachovává osu rotace v inerciálním prostoru. Díky příchodu technologie MEMS můžou mít gyroskopy miniaturní rozměry a mohou být integrovány do zařízení. Gyroskop na rozdíl od akcelerometru měří úhlovou rychlost náklonu v měřené ose. Proto je možné určit náklon i u předmětu, který se pohybuje se zrychlením.

Existuje několik druhů gyroskopů jako jsou například rotační, optický, jaderný a elektronický. V našem případě bude využitelný hlavně elektronický gyroskop.

Elektronický gyroskop

Elektronický gyroskop je vyráběný pomocí výrobní technologie MEMS. Gyroskop je integrován v malém pouzdru spolu s vyhodnocovacími obvody. Výstupní signál může být dle provedení snímače analogový nebo digitální. Základem integrovaných gyroskopů je periodicky se pohybující struktura přesně dané hmotnosti pružně spojená s rámem. Gyroskopy MEMS pracují na principu Coriolisovy síly. Směr pohybu musí být kolmý ke směru otáčení.



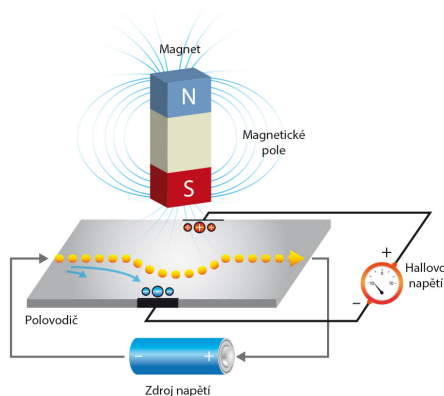
Obrázek 2.14: Princip Coriolisovy síly na mechanický gyroskop [3]

Hallova sonda

Zde jsou využívány zdroje [14, 16] a [15, strana 22].

Hallův jev je vznik elektrického napětí příčně ve vodiči, jímž podélně protéká elektrický proud, který je odchylován magnetickým polem kolmým ke směru proudu.

Vodivou destičku z kovu nebo polovodiče umístíme do homogenního magnetického pole tak, aby vektor magnetické indukce byl na destičku kolmý. Prochází-li destičkou elektrický proud ve směru nejdelší její hrany lze voltmetrem zjistit, že mezi bočními stěnami destičky vzniká napětí tzv. Hallovo napětí.



Obrázek 2.15: Zobrazení vzniku Hallova jevu

Zařízení využívající Hallův jev pro měření magnetické indukce se nazývá Hallova sonda. Dnešní Hallovy sondy obsahují většinou plátek polovodiče s relativně malou koncentrací nosičů nábojů v nichž je díky tomu jev lépe měřitelný. Kvalitní Hallova sonda musí být také teplotně kompenzovaná, protože vlivem teploty se mění koncentrace elektronů a děr, a tím se mění i velikost Hallova napětí. V praxi je teplotní citlivost Hallovy sondy, jeden z nejdůležitějších ukazatelů jakosti Hallovy sondy, protože rozhoduje o přesnosti snímače.

Vlastnosti sondy

O přesnosti také rozhoduje odpor polovodiče, který se pohybuje od $0,1 \Omega$ až po 20Ω , jedná se o kompromis mezi maximem citlivosti a mechanické pevnosti. Pro Hallovu sondu je charakteristický lineární vzrůst velikosti Hallova napětí naprázdno při zvětšující se magnetické indukci při konstantní hodnotě procházejícího proudu I . Nejdůležitější charakteristické hodnoty Hallovy sondy jsou jmenovitý proud I_n - většinou řádově desítky až stovky miliampér, maximální proud I_{max} - výrobci volí tak, aby se jeho průchodem teplota zvýšila maximálně o 10° vzhledem k teplotní závislosti součástky, jmenovité Hallovo napětí naprázdno U_{HO} - jedná se o napětí, naměřené při indukci $B = 1\text{T}$ a jmenovitém proudu I_n s velikostí řádově desetiny voltů, citlivost Hallovy sondy (tj. Hallovo napětí naprázdno při indukci $B = 1\text{T}$ a proudu $I = 1\text{A}$) - může být řádově desetiny až stovky V/A .

Sondy mají bohaté uplatnění v každodenní praxi. Například v automobilových motorech snímají Hallovy sondy (horní) polohu klikové hřídele a umožňují přesné načasování zapálení směsi. Používá se také pro měření a automatickou regulaci magnetických polí, měření velkých stejnosměrných proudů ($0,5$ až 10 kA), ovládání synchronních elektromotorů, BLDC elektromotory (např. v počítačových ventilátorech), bezkontaktní tlačítka, mechanické snímače (poloha, otáčky, zrychlení) apod.

2.4 Chyby měření

Zde využívám informace ze [17, 23, 31, 27, 28, 29, 30]. Základní pojmy a definice jsou vysvětleny na [32].

Výsledek měření je vždy zatížen chybou/nejistotou. Žádné měření není dobrým měřením pokud není určena jeho chyba. Není možné znát pravou hodnotu naměřené veličiny a také není možné znát skutečnou hodnotu chyby výsledku měření, proto se, například hodnota měřené veličiny uvádí ve formě střední hodnoty a intervalu, ve kterém se tato hodnota pravděpodobně pohybuje. Základní klasifikace chyb:

- Náhodné: jsou zjistitelné jen při opakovaných měřeních s konzistentním (bezesporným) nastavením měřicí techniky
- Systematické: Týkají se zařízení požívaných v procesu měření, vč. vnějších vlivů
- Hrubé: způsobeny lidským faktorem

Měření zařízení mohou být ovlivněna různými vlivy. Například zvýšenou teplotou, nižším/-vyšším tlakem a podobně.

Odhad údaje x měřené veličiny (při přímém měření veličiny) lze vyjádřit například pomocí aritmetického průměru z opakovaných měření hodnot x_i .

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{\infty} \frac{1}{i}$$

Pojem nejistota (nejistota měření) je označením pro parametr související s výsledkem měření a charakterizující rozsah hodnot, které je možné racionálně přiřadit k měřené veličině. Nejistota se skládá z několika dílčích nejistot (nejistoty typu A, B, C, kombinovaná a rozšířená nejistota - viz dále). Ke stanovení jejich velikosti jsou principiálně k dispozici tyto dvě metody:

Statistické zpracování naměřených údajů

Vyčíslení nejistoty typu A (u_A). Získané z opakovaných měření hodnoty veličiny (statistická analýza série naměřených hodnot). Hodnoty s rostoucím počtem opakovaných měření klesají. Příčina nejistoty není známá.

Postup při vyčíslení nejistoty typu A

1. Výpočet aritmetického průměru naměřených hodnot: $\bar{x} = \frac{1}{i}(x_1 + x_2 \cdots + x_i)$
2. Určení směrodatné odchylky jednoho měření: $s = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}$ zde kontrolujeme tzv. platnost „3s-kritéria“ (pro každé x_i musí platit $(x_i - \bar{x}) \leq 3 \cdot s$, vyloučíme hodnoty pro které toto neplatí a přepočteme od kroku 1.)
3. Pokud je počet měření < 10 , pak je potřeba započítat koeficient k_s ¹
4. Zaokrouhlíme
5. Určíme mezní chybu

¹<https://homel.vsb.cz/khe0007/opory/obrazy/chybtav.jpg>

Jiné než statistické zpracování naměřených údajů

Vyčíslení nejistoty typu B (u_B). Získané jinak než statistickým zpracováním výsledků opakovaných měření (např. od výrobce měřicího přístroje). Jsou vyhodnoceny pro jednotlivé zdroje nejistoty identifikované pro konkrétní měření. Jejich hodnoty nezávisí na počtu opakování měření (obdobně jako systematické chyby měření). Pocházejí z různých zdrojů. Společné působení jednotlivých nejistot typu B vyjadřuje výsledná standardní nejistota typu B. Nejistoty typu B se vážou na známé, identifikované zdroje, na rozdíl od nejistot typu A.

Přímé a nepřímé měření

Často nelze hledanou veličinu zjistit přímo a je nutno ji získat z více přímo měřených veličin. Je-li souvislost mezi hledanou veličinou a přímo měřenými veličinami dána jednoduchou funkcí (součet, rozdíl, součin, podíl nebo mocnina), pak vedou parciální derivace podle Gaussova zákona rozdělení chyb zase na jednoduchou funkci.

Kombinovaná standardní nejistota (u_C)

V praxi si ve většině případů nevystačíme pouze s nejistotou typu A nebo B. Výsledek je často totiž ovlivněn kombinací těchto dvou nejistot. Kombinovaná standardní nejistota udává interval, ve kterém se s poměrně velkou pravděpodobností může vyskytovat skutečná hodnota měřené veličiny.

$$u_{Cx} = \sqrt{u_A^2 + u_B^2}$$

Rozšířená standardní nejistota (U)

Rozšířená standardní nejistota U se zejména ve stavebnictví využívá velmi často. Rozšířená standardní nejistota U umožňuje vyjádřit výsledek měření tak, aby platil pro předem dané procento všech možných měření. Získá se tak, že se kombinovaná standardní nejistota u_C vynásobí součinitelem $k_u = 2$. Velikost koeficientu se volí v rozmezí hodnot 2 - 3.

$$U_x = k_u \cdot u_{Cx}$$

$k_u = 2$ Pro normální rozdělení odpovídá pravděpodobnosti pokrytí asi 95%.

$k_u = 3$ Pro normální rozdělení odpovídá pravděpodobnosti pokrytí asi 99,7%.

2.5 Fyzikální jevy

Mezi jevy, které působí na autíčko patří odpor vzduchu, tření, valivý odpor, odstředivá a dostředivá síla. Tyto síly vztahujeme ke dvěma soustavám, inerciální a neinerciální.

Jako inerciální vztahná soustava se ve fyzice označuje taková vztahná soustava, v níž platí 1. Newtonův pohybový zákon, tj. těleso, na které nepůsobí žádná síla nebo výslednice sil je nulová, je v klidu nebo se pohybuje rovnoměrně přímočaře. Platí zde zákon setrvačnosti a každá vztahná soustava, je-li vzhledem k dané inerciální soustavě v klidu nebo pohybu rovnoměrném přímočarém, je rovněž inerciální.

Soustavy, v nichž neplatí 1. Newtonův pohybový zákon, se nazývají neinerciální vztahné soustavy.

Zde se budeme zabývat hlavně fyzikálními jevy, které jsou potřebné k určení maximální rychlosti v zatáčce a jevy, které zpomalují autíčko, aby bylo schopné včas zpomalit do zatáčky.

Odpor vzduchu

Při výrobě aut se nikdy nesmí zanedbat odpor vzduchu. Právě kvůli tomuto jevu dnešní auta mají tvar takový, aby co nejvíce tento jev odstranili nebo použili ku prospěchu samotného řízení vozidla. Tímto jevem se zabývá aerodynamika. Používá se takzvaný aerodynamický koeficient.

Aerodynamický koeficient (součinitel) je bezrozměrná veličina, která vyjadřuje určitou aerodynamickou čistotu obtékaného tělesa. Čím menší je hodnota c_x , tím menší odpor vzduchu na automobil působí. Obvykle se hodnoty pohybují v rozmezí 0,2 – 0,4. Odpor vzduchu se také často používá pro zlepšení řízení vozidla (například přitlačením vozidla k zemi).

Tření

Pro výpočet maximální rychlosti v zatáčce je potřeba si určit koeficient klidového tření (f). Který pro nás bude něco jako mezi pneumatikou a mokrým asfaltem (Který je 0,2 až 0,5). Náš koeficient bude asi 0,27.

Výpočet smyku autíčka

Při výpočtu maximální rychlosti, které můžeme dosáhnout předtím než dojde ke smyku potřebujeme ještě váhu autíčka (m), poloměr zatáčky (R) a rychlost autíčka (v). Maximální hodnota klidové třecí síly je rovna:

$$F_{tmax} = f \cdot F_n$$

kde F_n je velikost tlakové síly kolmé k silnici a g je tíhové zrychlení.

Jestliže bude rychlost auta taková, že dostředivá síla potřebná k zatočení auta přesáhne hodnotu maximální klidové třecí síly, dojde ke smyku. To znamená, že výsledná rovnice bude taková:

$$F_{tmax} = fmg = m \frac{v_{max}^2}{R}$$

A z toho odvodíme:

$$v_{max} = \sqrt{R \cdot g \cdot f}$$

Z tohoto vzorce vypočteme maximální rychlost pro autíčko před smykem.

Kapitola 3

Rozbor řešené problematiky a realizačních možností

V této kapitole se budu věnovat vybráním součástí, návrhem algoritmů pro realizaci autíčka s autonomním řízením a problémy s tím spojené. Následně jejich zapojení a realizace algoritmu.

3.1 Rozbor problémů

Rozpoznání zatáčky

Jeden z hlavních problémů je rozpoznání zatáčky. V nejlepším případě by mělo autíčko vědět o parametrech zatáčky dostatečně dopředu, aby zvládlo začít zpomalovat dostatečně dopředu. Kdyby tato informace přišla příliš pozdě tak by autíčku hrozilo vypadnutí z dráhy nebo případně by muselo držet nižší rychlost na rovinkách tak, aby stihlo zpomalit.

Jeden z možných způsobů rozpoznání zatáčky je gyroskopem. Pokud hodnota překročí určitou hranici tak v ten moment lze předpokládat, že začala zatáčka. Tato hranice by se měla zpřesnit co nejvíce postupnými experimenty.

3.1.1 Rozpoznání ujetého kola

Rozpoznání ujetého kola je důležité ze dvou důvodů.

Při prvním projetí je autíčko v režimu rozpoznávání tvaru dráhy. Po ujetí prvního kola se musí přepnout do módu jízdy, kdy se snaží pomocí rozpoznání tvaru jet co největší rychlostí. Jedním ze způsobů rozpoznání ujetého kola je pomocí vnějšího prvku upevněného na začátek dráhy (např. magnet) a nějaký vnitřní senzor na jeho detekci (např. Hallova sonda). Takto zařízené rozpoznávání kola může být velice přesné. Záleží hlavně na volbě magnetu (na jeho síle) a také na volbě hranice při které magnetometr zaznamená magnet. Špatná volba těchto parametrů může vést k nerozpoznání kola nebo naopak rozpoznání kola moc brzy.

Další důvod je pro možnost přenastavení hodnot na začátek dráhy. Tímto můžeme odstranit nepřesnosti, které vzniknou při průjezdu dráhou. Program tedy může běžet delší dobu bez generování postupně větších nepřesností.

3.2 Podmínky pro výběr součástí

Zde budu popisovat co jsem považoval za důležité při výběru komponentů.

Hmotnost

Hmotnost určuje sílu potřebnou pro zrychlení auta, změně směru jízdy a potřebnou odstředivou sílu pro vykolejení popřípadě smyku autíčka.

Mikrokontrolér

Při volbě mikrokontroléru jsou důležité faktory hlavně jeho frekvence a flash paměť pro program. Ty určují rychlost vykonávání programu a jak velký program může být. Další faktory jsou například jeho způsob komunikace s připojenými zařízeními. Poté za méně důležité, ale stále důležité, faktory považuji jestli již obsahuje akcelerometr, gyroskop nebo jiné senzory, protože ty se můžou přidat dodatečně. Při volbě s již interními senzory se pouze může zrychlit zapojení a mírně zmenšit velikost.

U mikrokontrolérů je variace možností jak je naprogramovat. Mikrokontrolér jako samotná součástka potřebuje speciální programátor na jeho naprogramování.

Senzory

Při volbě senzorů je důležité uvažovat co potřebujeme měřit, použitelnost na vybraném autíčku, jeho celková velikost a za jakých podmínek lze použít.

Ze senzorů by mělo jít vyčíst co nejvíce dat o jízdě a stavu autíčka. Řešení musí minimálně obsahovat akcelerometr nebo gyroskop na detekování zatačky a měření její prudkosti a senzor ze kterého lze vyčíst projetí cílem popřípadě ujetou vzdálenost.

Další komponenty

Zaznamenaná dráha se musí ukládat do nějakého úložiště. To lze vyřešit více způsoby. Mikrokontroléry mohou obsahovat vnitřní paměť do které je možné tyto data ukládat.

Další způsob je externím úložištěm. Při použití tohoto řešení je důležité si uvědomit, že pro vývoj produktu je nutné mít možnost data vyjmout a prozkoumat popřípadě zkontrolovat.

Při řízení motoru je vhodné mít H-můstek, který zařídí řízení motoru.

3.3 Výběr komponent

Tato část se bude zabývat samotným výběrem komponentů a jejich specifikacemi.

Při výběru jsem uvažoval kromě napsaných faktorů v 3.2 také následující:

- Rychlost implementace
- Celková velikost hardwaru
- Způsob programování mikrokontroléru

Pro tuto práci jsem zvolil vývojovou desku Arduino NANO 33 BLE [4]. Firma Arduino také má vlastní software na programování, který obsahuje různé knihovny pro práci s vývojovými kity od této firmy. Samotné Arduino lze programovat přes počítač pomocí USB.

Deska se může napájet přes USB (5 V) nebo na desce přes pin VIN (od 3.3 V až 21 V). Samotná deska je velice malá (asi 3,5 cm na délku a 1,5 cm na šířku). Obsahuje 30 pinů z toho 22 pinů je určených pro volné používání (zbytek jsou napájecí piny, reset pin, atd.) případně na používání ke komunikaci pomocí různých rozhraní (I2C, SPI, UART).

Maximální výstupní proud z každého pinu je 15mA a maximální vstupní proud je 5mA. Při připojení pinu VIN ke zdroji obsahuje také piny, které pomocí redukce vstupního napětí mohou vysílat 3.3V a 5V, což se hodí pro napájení dalších periférií.

Mikrokontrolér

Deska obsahuje 32 bitový mikrokontrolér nRF52840[24] od nordic semiconductor běžící na 64 MHz, 1 MB paměti pro program a SRAM s velikostí 256 KB. Obsahuje také pět 32-bitových časovačů. Podporuje až čtyřikrát SPI Master a třikrát SPI Slave, další možné rozhraní je dvakrát I2C a dvakrát UART.

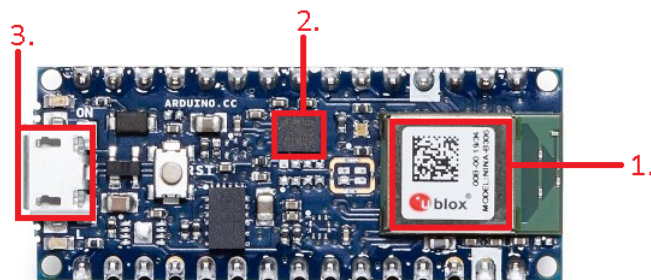
Senzory

Na desce je použit devítiosý inerciální senzor (akcelerometr, gyroskop a magnetometr) LSM9DS1[18], který se použije na detekci a zaznamenání zatáček.

LSM9DS1

Devítiosý inerciální senzor (akcelerometr, gyroskop a magnetometr), který má nastavitelné měřítka.

Pro akcelerometr je to $\pm 2/\pm 4/\pm 8/\pm 16$ g, pro gyroskop to je $\pm 245/\pm 500/\pm 2000$ dps a pro magnetometr $\pm 4/\pm 8/\pm 12/\pm 16$ gauss. Výstup senzoru je 16-ti bitový doplněk. Výstup může být nastaven na FIFO mód, který má 32 slotů pro každý výstup akcelerometru (X, Y a Z) a stejně tak i pro výstupy gyroskopu (yaw, pitch a roll) nebo také do nepřerušovaného módu, kde se registry přepisují při každém dalším měření gyroskopu. Může komunikovat přes SPI nebo IIC sériové rozhraní.



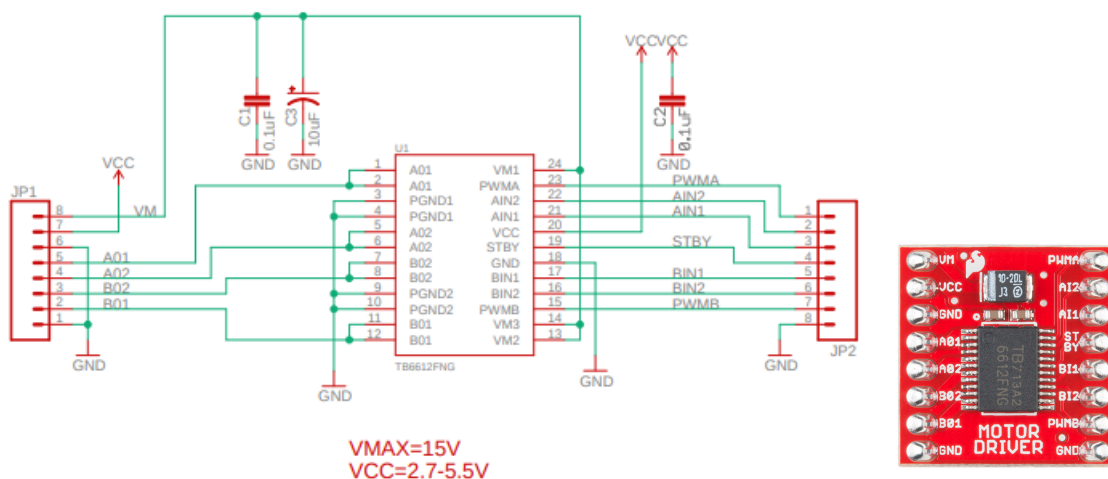
Obrázek 3.1: Deska Arduino NANO 33 BLE, 1. mikrokontrolér nRF52840, 2. devítiosý inerciální senzor LSM9DS1, 3. Micro-USB port

Další komponenty

Pro řízení motoru se využije vývojový nástroj pro řízení energie ROB-14450¹. Dále se použije obvod pro microSD kartu BOB-00544², který používá SPI rozhraní.

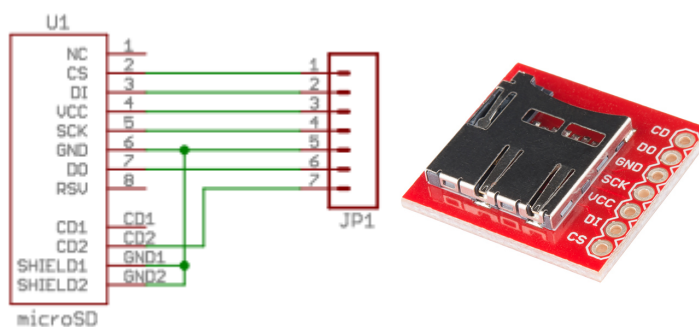
Řízení energie (ROB-14450)

Tento obvod obsahuje H-můstek TB6612FNG pro řízení dvou stejnosměrných motorů. Má dva vstupní signály, které se mohou použít na řízení motoru se čtyřmi funkčními módy (CW, CCW, short-brake a stop). Je napájen 2.7 V až 5.5 V a vstup, kterým je napájen motor může být až do 15 V.



Obrázek 3.2: Schéma zapojení H-můstku TB6612FNG

Slot pro microSD kartu (BOB-00544)



Obrázek 3.3: Schéma zapojení slotu pro microSD kartu

¹<https://www.sparkfun.com/products/14450>

²<https://www.sparkfun.com/products/544>

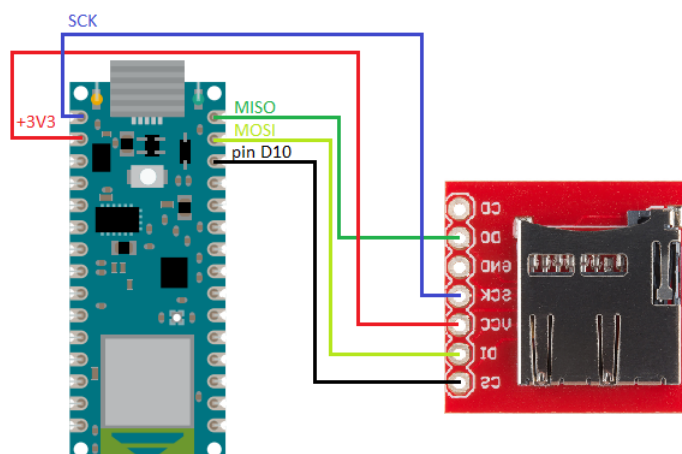
Kapitola 4

Návrh a realizace řešení

4.1 Zapojení

U zapojení je nutno vědět rozhraní jednotlivých součástek. Pro většinu součástek platí, že potřebují být pouze volné piny. Výstupy těchto pinů poté lze nastavit v programu. Výjimka je třeba SD karta, která komunikuje přes SPI rozhraní.

4.1.1 Zapojení slotu pro microSD kartu



Obrázek 4.1: Zapojení slotu pro microSD kartu k platformě

- +3V3 - Slouží pro napájení SD karty. Karta by měla být napájena 3,3 V, použitím vyššího napětí by mohlo dojít ke zničení karty.
- SCK - Generuje hodinový signál, který řídí komunikaci.
- MISO - Master in, slave out. Posílá data z karty.
- MOSI - Master out, slave in. Posílá data na kartu.
- pin D10, CS(Chip Select) - Pin nastavený na vysílání logické 1/0, také slouží jako výběr chipu, který se má aktivovat.

4.2 Data ze senzorů a jejich využití

Data ze senzorů jsou důležitá pro správné chování autíčka. Výstupy z prvních testování budou použity při optimalizaci programu. Chyba měření otáček na dráze by měla být v řádech jednotek procent (jak při experimentech s autíčkem bez dráhy tak na dráze).

4.2.1 Gyroskop

Čtení vzorků

Při čtení vzorků z gyroskopu musíme uvažovat podobný proces jako u akcelerometru, ale nemusíme odstraňovat nulovou odchylku. Poté musíme znát vzdálenost ve které jsme vzorek načetli, abychom mohli případně zaznamenat zatáčku v té vzdálenosti. Není potřeba znát čas ve který gyroskop vzorek udělal.

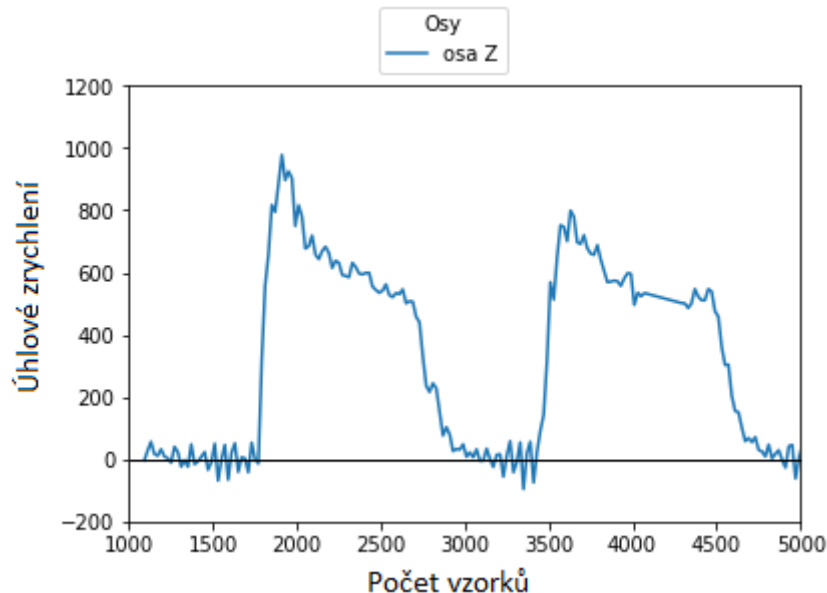
Důležité je vědět jak dlouho potrvá udělat jeden vzorek. Na čtení vzorků z akcelerometru se program nejvíce zdrží. Je to způsobeno hlavně tím, že se musí vzorky dále filtrovat. Používaný gyroskop má frekvenci 952 Hz. Jednoduchý filtr, který používá 20 vzorků bude trvat minimálně:

$$T = \frac{1}{952} = 1,05ms$$

$$t = 20 * 1,05 = \underline{\underline{21ms}}$$

Zkouška funkce

Gyroskop měří úhlovou rychlost ve stupních za sekundu. Při zkoušení funkce gyroskopu na dráze o dvou stejných zatáčkách při konstantní rychlosti, vyšel následující graf.



Obrázek 4.2: Ukázka vzorků gyroskopu z dráhy

Graf ukazuje úhlové zrychlení, pokud toto zrychlení překročí určitou hranici (autíčko bude v zatáčce příliš rychlé) tak může dojít ke smyku a popřípadě vypadnutí autíčka z dráhy.

4.2.2 Akcelerometr

Čtení rychlosti

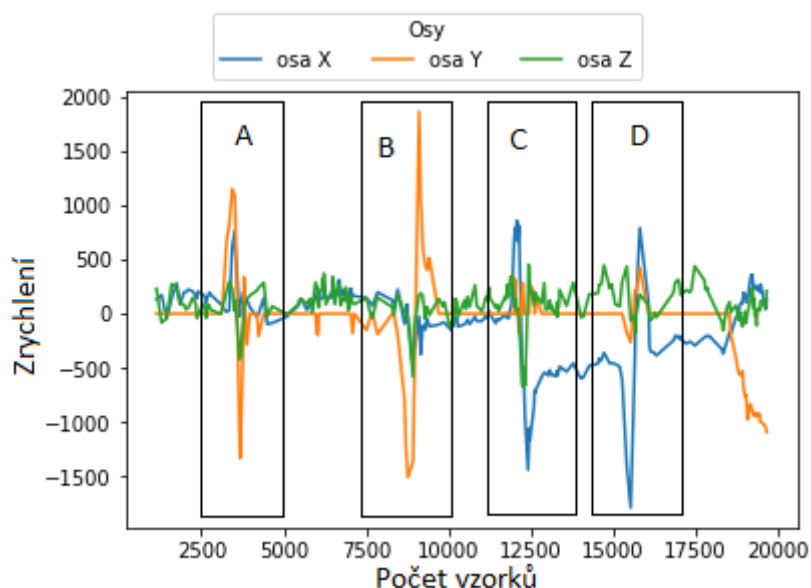
Aby bylo možné akcelerometr použít je potřeba vědět časové rozmezí mezi změřenými vzorky. Existují dva způsoby jak toto vyřešit.

První způsob je započítávání času do výpočtu ujeté vzdálenosti tím způsobem, že se změří čas od posledního měření až k nynějšímu měření. Tento způsob je snadnější, ale nese s sebou rizika zejména nepřesnost a zpomalení vykonávání programu.

Dalším způsobem je používat přerušení pro zaznamenávání vzorků ve stejný čas. Takto nemusíme zaznamenávat čas a usnadní to výpočet vzdálenosti.

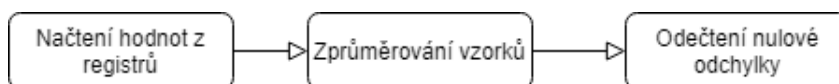
Zkouška funkce akcelerometru

Hlavní část, bylo rozpoznat jednotlivé osy. Prvotní experiment se tedy skládal z rychlých pohybů autíčka postupně a v tomto pořadí dozadu, dopředu, doprava a doleva.



Obrázek 4.3: Ukázka vzorků akcelerometru po výpočtu nulové odchylky a jejím odstranění při pohybu A: dozadu, B: dopředu, C: doprava a D: doleva

Z tohoto grafu je patrné že osa Y zobrazuje pohyb dopředu a dozadu a osa X zobrazuje pohyb doleva a doprava. Akcelerometr by šlo použít při výpočtu zrychlení působící na auto, tím pádem i jeho rychlost a přidalo by to možnost sledovat přesnou délku dráhy a vzdálenost, kterou autíčko ujelo.



Obrázek 4.4: Průběh programu na zpracování vzorků z akcelerometru

Čtení vzorků

Pro čtení vzorků platí to samé jako u gyroskopu 4.2.1. To znamená, že používaný akcelerometr má frekvenci 952 Hz. Jednoduchý filtr, který používá 20 vzorků bude trvat:

$$T = \frac{1}{952} = 1,05ms$$

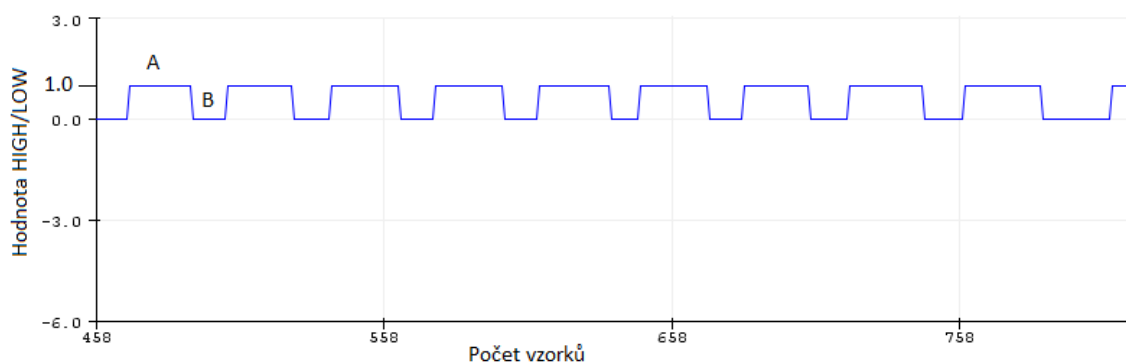
$$t = 20 * 1,05 = \underline{\underline{21ms}}$$

Další možnosti

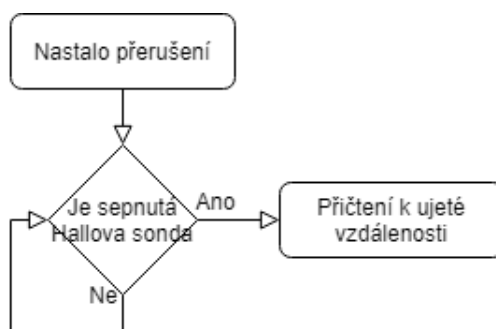
Další možnost čtení rychlosti je z Hallovy sondy. Výhoda je zejména rychlejší a snadnější zpracování vzorků. Do nevýhod patří například přidání dalšího prvku (složitější manipulace) a udržení rychlejší snímání hodnoty ze sondy než rychlosti otáčení kol.

4.3 Hallova sonda

Pro detekci otáček je zapotřebí obstarat magnet, který bude dostatečně silný, ale nesmí být moc silný, jinak by sonda nerozpoznala otáčku kol. Další problém, který je nutno vyřešit je rychlost otáček proti rychlosti kontroly stavu sondy.



Obrázek 4.5: Ukázka vzorků z Hallovy sondy, A: senzor nedetekuje magnet, B: senzor detekuje magnet



Obrázek 4.6: Zaznamenávání ujeté dráhy pomocí akcelerometru při použití přerušeni na pin s Hallovou sondou

Počítání otáček

Aby rozpoznávání magnetu na kolech probíhalo správně a každé kolo se zaznamenávalo, musí být rychlost snímání větší než rychlost otáček. Experimentem při střídě 21,8%(PWM posílané na motor driver/Maximální hodnota PWM(8 bit): $\frac{55}{255}$) jsem zjistil, že autíčko projede dráhu velkou 2,80 metrů za asi 4,5 vteřiny.

Velikost kol autíčka: $r = 1\text{cm}$

Obvod kol: $o = 2\pi * 1 = 6,28\text{cm}$

Velikost dráhy: $l = 2,80\text{m}$

Počet otáček potřebných k projetí dráhy: $c = \frac{280}{6,28} = 44,56$ otáček

Rychlost otáčení kol při střídě 21,8%: $rps = \frac{44,56}{4,5} \approx \underline{\underline{9,9}}$

Což znamená, že platforma musí zvládnout zaznamenat a zpracovat aspoň 10 otáček za sekundu. V nejhorším případě by vzorkování mělo probíhat minimálně 5-krát rychleji než je otáčení kol (50-krát za sekundu), aby nevznikala příliš velká chyba, takže každých 20 ms. V nejlepším případě by to mělo kontrolovat pozici magnetu asi 10-krát rychleji než je otáčení kol(100-krát za sekundu). Takže každých 10 ms. Nebo to lze provést nastavením pinu se sondou jako signál přerušení.

4.4 Ovládání motoru

Ovládání motoru je pomocí komponenty s H-můstkem TB6612FNG. Tato komponenta je popsána v 3.3. Tento řadič má několik módů. Mezi těmito módy se přepíná pomocí pinů na desce. Zejména jde o piny AIN1 a AIN2. Těmito piny se nastavuje mód pro motor „A“, neboli výstupy z pinů AO1 a AO2. Zvolené nastavení je popsáno v tabulce 4.7, V tabulce jsou popsány piny, které představují piny na desce (např. pin IN1 může být buď AIN1 nebo BIN1, OUT1 je AO1 nebo BO1, atd.).

Input				Output		
IN1	IN2	PWM	STBY	OUT1	OUT2	Mode
H	H	H/L	H	L	L	Short brake
L	H	H	H	L	H	CCW
		L	H	L	L	Short brake
H	L	H	H	H	L	CW
		L	H	L	L	Short brake
L	L	H	H	OFF (High impedance)		Stop
H/L	H/L	H/L	L	OFF (High impedance)		Standby

Obrázek 4.7: Ukázka tabulky zobrazující možné stavy řadiče

Dalším vstupem je VM. Tento pin přijímá celkové napětí a proud, které se redukuje podle nastavení PWM posílaného na pin PWM. Jelikož používám vstupy a výstupy pro motor A, tak výstup pinů AO1 a AO2 určují piny AIN1, AIN2 a PWMA.

4.5 Datová reprezentace tvaru autodráhy

Při mapování dráhy je pro algoritmus důležité hlavně zapamatování si délky zatáčky (začátek, konec), prudkost a umístění na dráze. Při prvním průjezdu je nutné si tato data uložit. Při dalším průjezdu se budou tato data využívat na výpočet rychlosti na kterou auto bude muset zpomalit a kdy bude moci auto začít zrychlovat až na maximální hodnotu.

O zatáčce se bude zaznamenávat prudkost, začátek, konec. A takto bude vypadat i datová struktura dráhy (zatáčka1->zatáčka2->...). Rovinka se na dráze zaznamenávat nemusí (bude se předpokládat, že když je auto mezi dvěma zatáčkami tak jede po rovince).

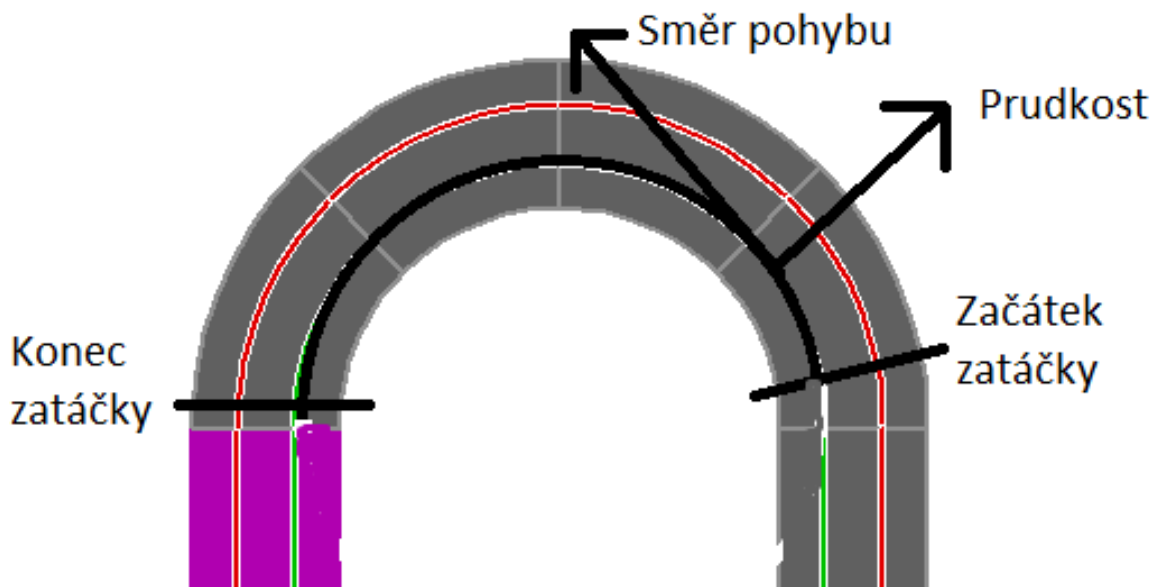
Začátek a konec

Začátek zatáčky bude reprezentovat čas, který uběhl od poslední zatáčky a ve kterém zatáčka byla rozpoznána programem. Začátek v nejlepším případě zná program dostatečně dopředu, aby stihl zpomalit na požadovanou rychlost.

Konec bude reprezentovat čas, který uběhl než algoritmus rozeznal konec zatáčky.

Prudkost

Prudkost určuje maximální rychlost, kterou autíčko může dosáhnout při projíždění zatáčky aniž by došlo ke smyku.



Obrázek 4.8: Ukázka ukládaných hodnot na dráze

4.6 Algoritmy

4.6.1 Algoritmus pro mapování autodráhy

Tento algoritmus mapuje dráhu při prvním projetí kola. První akcí autíčka je přijet k magnetu, který určuje začátek a konec dráhy (řádek 3). Pomocí Hallovy sondy rozpozná posun autíčka na autodráze. Ujetou vzdálenost ukládá do proměnné (řádek 7), jako začátek zatáčky, a každé projetí tuto proměnnou přepíše. Začátek zatáčky ukládá do pole, pokud narazí na zatáčku. Zatáčku rozpozná pomocí průměrné hodnoty z gyroskopu. Hodnota, která určuje hranici pro vyhodnocení zatáčky na dráze je možná získat experimentací (v tomto případě 100, řádek 8). Hodnota z experimentů byla získána na základě hranice, kdy už jistě nebude docházet k šumu. Hodnota z gyroskopu také určuje rychlost, které je možno docílit, aniž by autíčko vypadlo z autodráhy. Nastavení počátečních hodnot (řádek 1) znamená, nastavení prvotní rychlosti (taková hodnota ve které nenastane smyk), nastavení módu dvumotorového řadiče, sledování přerušení, senzorů a komunikace s SD kartou. Algoritmus končí, když rozpozná magnet umístěný na začátku dráhy.

K nastavení nulové odchylky dochází na začátku programu. Pro změření této odchylky je potřeba, aby se autíčko nehýbalo, proto k tomu dochází ještě před rozjetím.

K průměrování hodnot se používá 20 hodnot pro gyroskop a 12 hodnot pro akcelerometr. Tyto hodnoty (20 a 12) byli získány experimentací s těmito hodnotami a byli vybrány tak, aby bylo potřeba co nejméně vzorků (algoritmus se zpomalí až o 1 milisekundu za každý vzorek) a zároveň, aby algoritmus odstranil co nejvíce šumu, který by mohl ovlivnit výsledek.

```
1 Prvotní inicializace a nastavení počátečních hodnot;
2 Rozjed autíčko;
3 while Nejsi u magnetu do
4   | Zkontroluj magnetometr;
5 end
6 while Projíždíš první kolo do
7   | Nastav začátek zatáčky;
8   | Načti vzorky z gyroskopu a magnetometru;
9   | Zprůměruj vzorky a odstraň nulovou odchylku;
10  while Vzorek z gyroskopu je větší než 100 do
11    | Přidej zatáčku do pole se zaznamenanou dráhou;
12    | Načti vzorky z gyroskopu a magnetometru;
13    | Zprůměruj vzorky a odstraň nulovou odchylku;
14    | Ulož vzorek z gyroskopu;
15  end
16  if Je potřeba uložit zatáčku then
17    | Zprůměruj vzorky z gyroskopu a ulož jako vzorek;
18    | Ulož konec zatáčky;
19  end
20 end
21 Nastav začátek dráhy;
22 Zrychli auto;
```

4.6.2 Algoritmus pro jízdu po zmapované dráze

Algoritmus využívá dříve zpracované data o dráze. Pokud vjíždí do zatáčky zpomalí a při výjezdu zrychlí. Jeho polohu zjistí z Hallovy sondy, která přičítá k ujeté dráze při otočení předních kol(řádek 2). Z pole kontroluje začátek zatáčky do které má vjet(řádek 3). Poté co vyjede ze zatáčky načte se následující zatáčka(řádek 8). Jakmile dojde na konec dráhy načte se první zatáčka z pole(řádek 12) a nastaví se ujetá vzdálenost na nulu(řádek 13) a program pokračuje od začátku.

```
1 Načti zatáčku;
2 Zrychlení autíčka na nastavenou hodnotu;
3 Zkontroluj umístění autíčka na dráze;
4 while Projíždíš zatáčku do
5   | Nastav rychlost autíčka;
6   | Zkontroluj umístění autíčka na dráze;
7   | if Autíčko došlo na konec zatáčky then
8     |   | Nastav rychlost autíčka maximální rychlost;
9     |   | Načti další zatáčku v seznamu;
10  | end
11 end
12 if Rozpoznáš magnet na dráze then
13   | Načti první zatáčku;
14   | Nastav ujetou vzdálenost na 0;
15 end
```

4.6.3 Implementace

Program byl implementován v jazyce C++. Zdrojový kód je zpracován preprocesorem a poté přeložen klasickým překladačem pro jazyk C++. Skrytá funkce `main()`, která spustí funkce `init()` (inicializace), poté `setup()` a nakonec `loop()`. Funkci `main()` obsahuje soubor `main.cpp`, který se stáhne s programem ArduinoIDE ve kterém byla i tato práce programována. Funkce `setup()` je provedena pouze jednou na začátku programu a poté funkce `loop()` je prováděna do nekonečna. Vývojové prostředí Arduino IDE je velice jednoduché a neposkytuje různé funkce pro zlepšení, zrychlení programování a hlavně program se kompiluje vždy celý a ne jen jeho změněné části(nahrání programu trvá mnohem déle a se zvyšující se velikostí programu se zvyšuje i doba čekání na kompilaci). I přesto jsem používal Arduino IDE, protože je rychle aplikovatelný, ale delší program by bylo vhodné udělat například ve VS studiu, které podporuje vývoj desek od Arduina.

4.6.4 Další funkce a knihovny

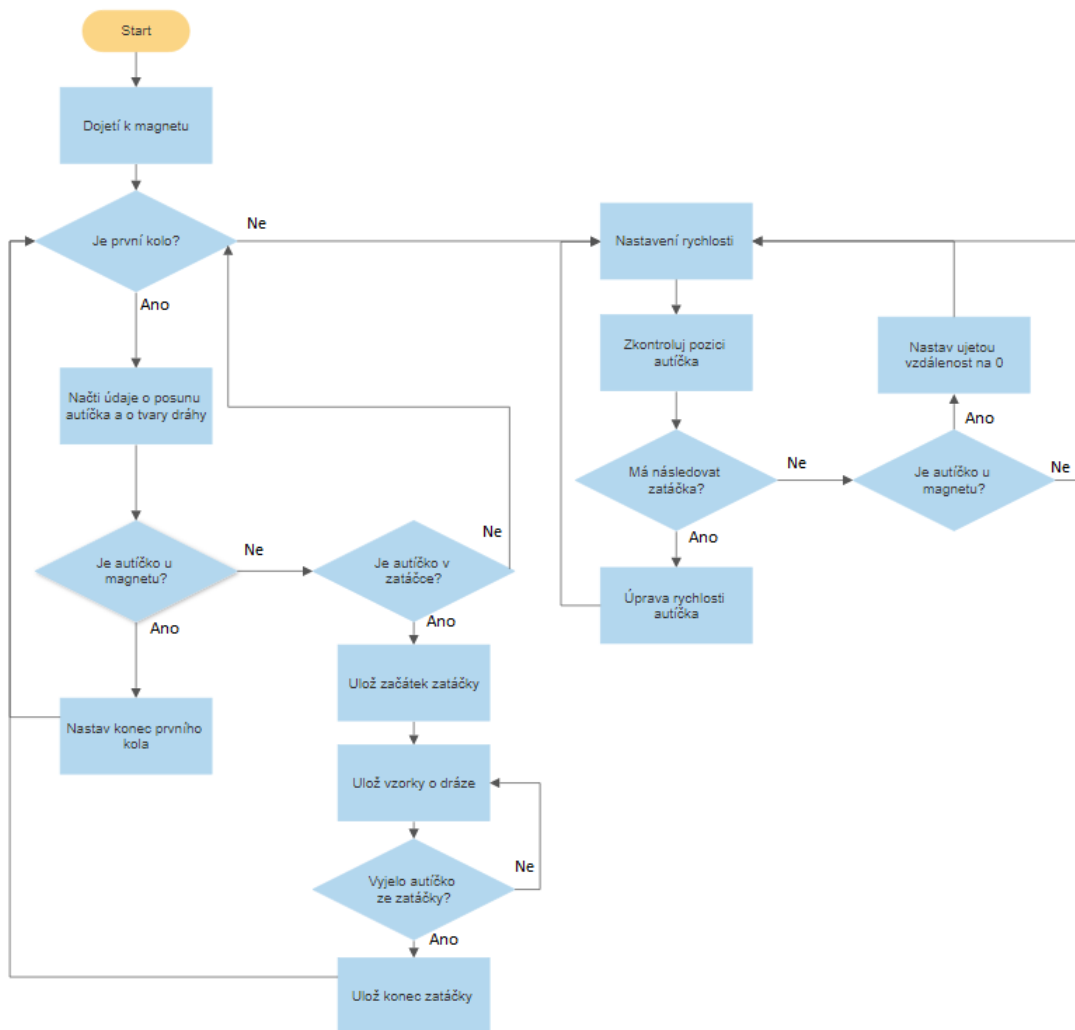
Arduino také přináší vlastní knihovny, které podporují abstrakci kódu, jednodušší práci a ušetření práce s testováním, jelikož knihovny obsahují i ukázkový případ použití.

Dále jsou použity knihovny, které usnadňují komunikaci s SD kartou a čtení ze senzorů. Jsou to knihovny `Arduino_LSM9DS1.h`, `SPI.h` a `SD.h`.

Knihovna `Arduino_LSM9DS1.h` byla upravena tak, že rychlost vzorkování gyroskopu je nastavena na maximum (952 Hz) a další úpravy se týkali zpracování hodnot vzorků. Vzorky byli posílány jako `float` hodnoty, ale lépe se pracuje s celočíselnými hodnotami, proto se převedli do `int`. Čtení z registrů probíhá pomocí funkce `readAcceleration()` pro

akcelerometr, `readGyroscope()` pro gyroskop a `readMagneticField()` pro magnetometr. Knihovna při zavolání funkce `IMU.begin()` zapíše do registrů senzoru hodnoty takové, že aktivuje získávání vzorků pro všechny osy (3 osy od akcelerometru, gyroskopu a magnetometru, neboli pro 9 os). Získávání vzorků se ukončí zavoláním funkce `IMU.end()`.

Ve stavu kde dochází k dojetí k magnetu značící začátek/konec dráhy se autíčko pohybuje pomalu kupředu. Tento stav je tam z důvodu synchronizace dráhy s algoritmem (první projetí kolem magnetu). Konec prvního kola určuje také magnet na dráze (druhé projetí kolem magnetu). Kontrola pozice autíčka probíhá kontrolou hodnoty proměnné do které přičítá přerušování z Hallovy sondy. Tato hodnota se poté porovnává se začátkem následující zatáčky a případně sníží rychlost autíčka dostatečně předem. Nastavení ujeté vzdálenosti na nulu slouží pro synchronizaci pozici autíčka s dráhou, aby v důsledku delší jízdy nevznikali příliš velké chyby algoritmu.



Obrázek 4.9: Zjednodušený vývojový diagram algoritmů pro mapování a průjezd po dráze.

Ukázka kódu pro ukládání na SD kartu

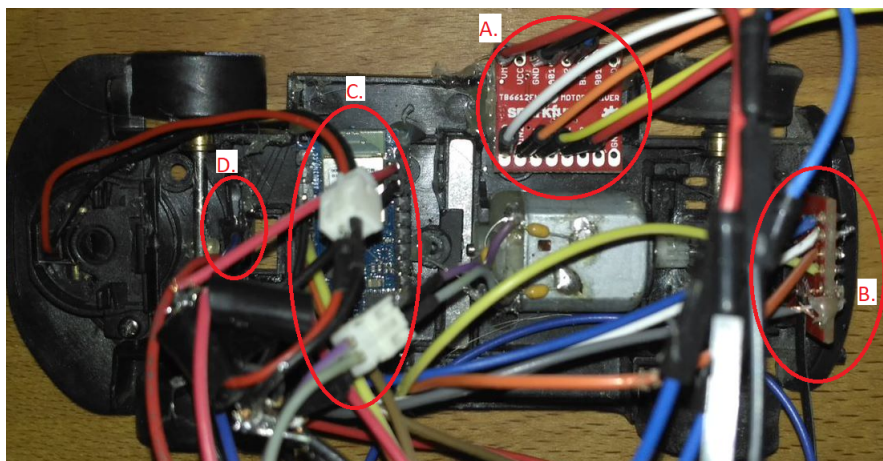
Kód využívá knihovny vyvinuté firmou Arduino, které usnadňují komunikaci přes SPI rozhraní. Konkrétně knihovny `SPI.h` a `SD.h`.

Knihovny podporují formáty FAT16 a FAT32 na standartních SD nebo SDHC kartách. Podporují i otevírání více souborů na jednu. Komunikace mezi deskou a kartou je zařízena knihovnou `SPI.h`.

```
1 #include <SPI.h>
2 #include <SD.h>
3
4 void sdCard(char printString[], const char fileName[]) {
5     file = SD.open(fileName, FILE_WRITE);
6     // if the file opened okay, write to it:
7     if (file) {
8         file.println(printString);
9         // close the file:
10        file.close();
11    }
12 }
13
14 void setup() {
15     if (!SD.begin(microSD)) { //Beginning communication with card
16         while (1); //Error occurred
17     }
18     char line[] = "Hello World!";
19     sdCard(line, "Hello.txt");
20 }
21
22 void loop() {
23 }
24 }
```

4.7 Popis realizace

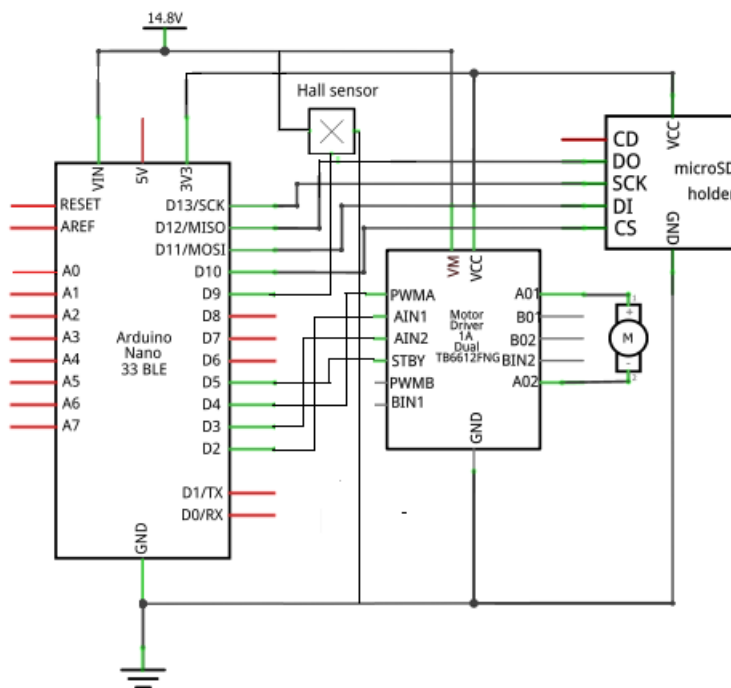
Nejdříve bylo potřeba všechny komponenty po jedné odzkoušet. Po odzkoušení zvlášť všech částí řešení jsem začal zapojovat všechny komponenty k platformě a ověřoval jsem projekt jako celek.



Obrázek 4.10: Zapojení všech komponent, A: řadič motoru, B: slot pro microSD kartu, C: Arduino NANO 33 BLE, D: Hallova sonda

4.7.1 Zapojení

Hallova sonda je připojena k zemi, napájení a pinu D9 u kterého je nastavený pullup rezistor, protože jinak by mohla sonda měnit logickou hodnotu i bez podnětu. „Motor Driver“ je připojený na piny D2 – D5, kde pin D5 je vždy na logické 1, pin D4 určuje PWM motoru a piny D3 a D2 určují mód řadiče. Slot pro microSD kartu využívá piny D13 – D10, kde pin D13 je hodinový výstup, který řídí komunikaci mezi kartou a platformou, piny D12 a D11 je pro samotnou komunikaci z a do platformy a pin D10 pouze vybírá (nebo také zapíná) vybraný čip.

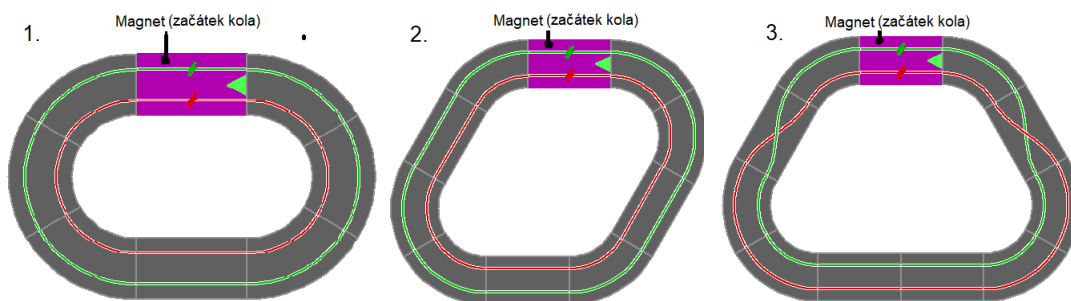


Obrázek 4.11: Zapojení komponentu k platformě

Kapitola 5

Zhodnocení řešení

Zkoušení řešení probíhalo na třech tvarech kol.



Obrázek 5.1: Dráhy, které byly použity při testování autíčka. Dráha č. 1 je na vnější(zelené) dráze dlouhá 2,87 metrů, dráha č. 2 je na vnější(zelené) dráze dlouhá 3,56 metrů a dráha č. 3 je na zeleně vyznačené dráze dlouhá 3,61 metrů

Na první díl, který napájí celou dráhu, byl přidělán magnet, který určoval konec/začátek dráhy. Pro počáteční testování (testování funkcí a možností desky, microSD karty apod.) byla vybrána první dráha z důvodu jednoduchosti.

5.1 Problémy s řešením

5.1.1 Záznam výsledků

Pro záznam výsledků byla použita microSD karta. Testování SD karty probíhalo tak, že se vzal testovací řetězec "Hello world" a ten se poslal přes SPI rozhraní pomocí knihovny "SPI.h" do souboru se jménem "HelloWord.txt". Při tomto testování se platforma zasekla a nadále odmítala komunikovat nejen s kartou, ale i s počítačem. Počítač nedokázal desku rozeznat. Deska má v paměti EEPROM uložený program pro nahrávání programu z počítače a vývojáři této desky navrhli tlačítko na desce, které při jednom stisknutí restartuje program běžící na desce a při dvou stisknutích začne nekonečně opakovat program v EEPROM pro nahrávání programu a tímto způsobem lze deska vrátit do původního stavu.

Dalším problémem byl čas potřebný pro zapsání na SD kartu. Tento experiment probíhal zapsáním řetězců různých délek a sledování rozmezí mezi začátkem a koncem zapisování.

	Pokus 1.	Pokus 2.	Pokus 3.
Počet znaků	2800	5605	5645
Čas (ms)	226	251	264

Tabulka 5.1: Ukázka časového rozdílu při zapisování

Závěr

Velikost řetězce obsahujícího jméno souboru nesmí přesahovat 12 znaků, kde 8 znaků je určeno pro jméno souboru, 1 znak pro tečku a 3 znaky pro typ souboru. Možné řetězce by tedy mohli být "accel.txt", "result.txt" a podobně. Dále zapisování na kartu je velice pomalé (až několik stovek milisekund), proto není vhodné pro ukládání hodnot v průběhu programu a pouze je možné pro ladící výstup. Zapisování větších částí dat naráz je rychlejší než postupné psaní malých částí. Z výsledků nemůžeme říct, že hlavním důvodem pro pomalé zapisování na kartu je navázání korektního spojení mezi kartou a deskou.

5.1.2 Měření rychlosti

Použití akcelerometru

První a největší problém byl, že platforma arduino nedokázala vykonat funkci přerušení ve které by byla funkce na čtení hodnot z registrů. Je to pravděpodobně způsobeno nedostatečně velkým zásobníkem. To znamená, že nebylo možné načítat vzorky z akcelerometru v jasný a vždy stejný čas. Platforma nepodporuje změnu nastavení velikosti zásobníku, proto to nebylo možné implementovat. Započítávání času do měření vzorků při čtení z registrů bez přerušení se také ukázalo být neúčinné, protože testy ukázaly chybu větší jak 20%.

Použití akcelerometru by zjednodušilo a urychlilo vývoj. Bez použití akcelerometru se musí použít jiná komponenta, která by minimálně určovala rychlost autíčka nebo jeho ujetou dráhu. Dále zatáčky se v tomto případě musí zaznamenávat jinou komponentou. Jestli se pro rozpoznání zatáček použije gyroskop, tak samotný výpočet prudkosti zatáčky bude složitější a pomalejší.

Použití Hallovy sondy

Použití Hallovy sondy bez přerušení se ukázalo býti problematické. Hlavní důvod byl, že autíčko se nesmí pohybovat rychleji než je možné zaznamenávat. Jak již bylo zmíněno v sekci 4.3. Tento experiment probíhal na dráze č. 1 5.1. Po experimentech se ukázalo, že chyba se opravdu zvětšuje se zvyšující se rychlostí. Při střídě cca 27,5% ($\frac{70}{255}$) se ukázala největší chyba 20%.

Dráha je dlouhá $d = 2,80m$. Takže kolo autíčka o průměru $r = 1cm$ by se mělo otočit asi: $\frac{d}{2\pi*r} = \frac{280}{2\pi*1} = 44,56 \approx 45$ otáček.

Pokus	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	Očekávaná hodnota
Otáčky	42	50	36	52	40	42	38	50	40	53	45

Tabulka 5.2: Přehled výsledků u pokusů nad autodráhou

Nejhorší případ je pokus č. 3, kde chyba je $45 - 36 = 9$, $(\frac{9}{45}) * 100 = 20\%$

Hallova sonda s přerušením ze začátku nevypadala přívětivě. Testování sondy probíhalo jako nastavení přerušení na pin s výstupem z Hallovy sondy a dále byla deska připojena

k počítači pomocí USB. Ve funkci vyvolané přerušením byla změna logické hodnoty na diodě. Po tomto experimentu bylo zjištěno, že nějaké podněty mohou vést k nesprávnému zaznamenávání hodnoty na pinu se sondou. V tomto případě to bylo zapnutím funkce posílání dat do počítače přes USB. Jakmile začalo posílání dat začala dioda závislá na výstupu Hallovy sondy blikat aniž by byl magnet v blízkosti Hallovy sondy.

Zhodnocení výsledků

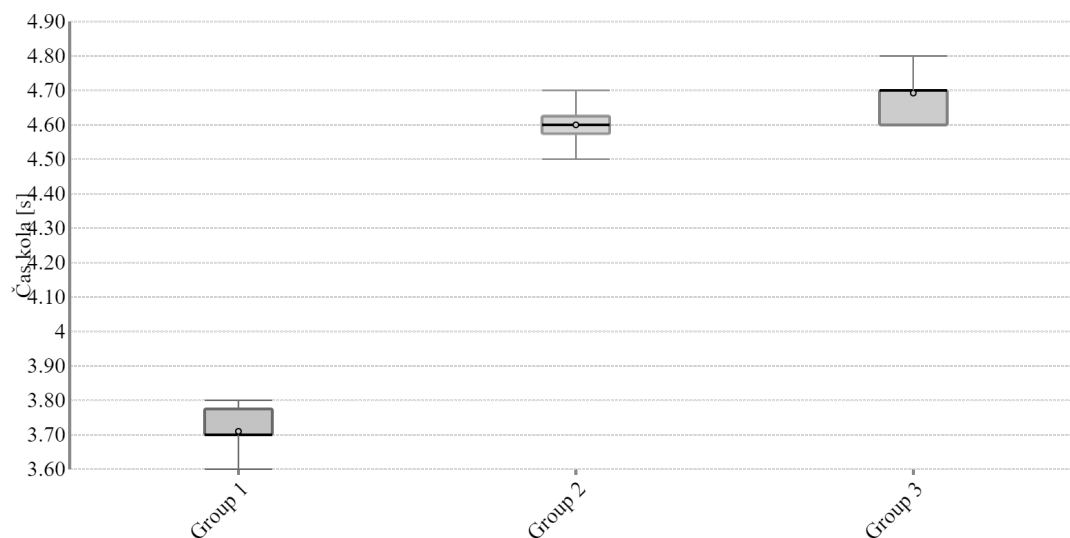
Při testování první kolo proběhlo bez problémů. Autíčko jelo (jak bylo nastaveno) na rychlost takovou, aby spolehlivě projelo kolo bez vykolejení a jiných problémů. Při detekci magnetu zrychlilo, ale ujetá dráha která měla být zaznamenávána Hallovou sondou nebyla správně zaznamenána. Proto na první zatáčce autíčko zrychlí na tolik že nastane smyk.

5.1.3 Použití částí programu

Měření probíhaly na výše zmíněných dráhách 5.1. Kde byli zkoušeny různé části programu.

Konstantní rychlost

Při prvním měření bylo PWM nastaveno na 27,45% ($\frac{70}{255}$) tedy na konstantní rychlost, která může být nastavena po celou dobu kola, aniž by vypadlo autíčko z dráhy nebo dostalo smyk. Rychlost kola byla měřena manuálně stopkami, proto se může vyskytovat rozdíl od skutečných hodnot o asi $\pm 0,2$ s (cca reakční doba).



Obrázek 5.2: Krabicový graf pro vyjádření časů změřených na dráhách

Zde lze vidět chyba, která je nejspíše vytvořena využitím lidského prvku u měření času.

Pomalou zrychlující

První kolo bylo projeto s PWM 27,45% ($\frac{70}{255}$) a autíčko pomalu zrychlovalo.

PWM [%]	27,45	27,84	28,24	28,63	29,02	29,41	29,80	30,20
Dráha 1 [s]	3,8	3,8	3,7	3,5	3,6	3,4	3,3	3,3
Dráha 2 [s]	4,4	4,3	4,3	4,1	4,0	4,0	3,9	3,8
Dráha 3 [s]	4,5	4,4	4,4	4,2	4,1	4,0	4,0	3,9

Tabulka 5.3: Tabulka s ukázkou proměnné rychlosti autíčka

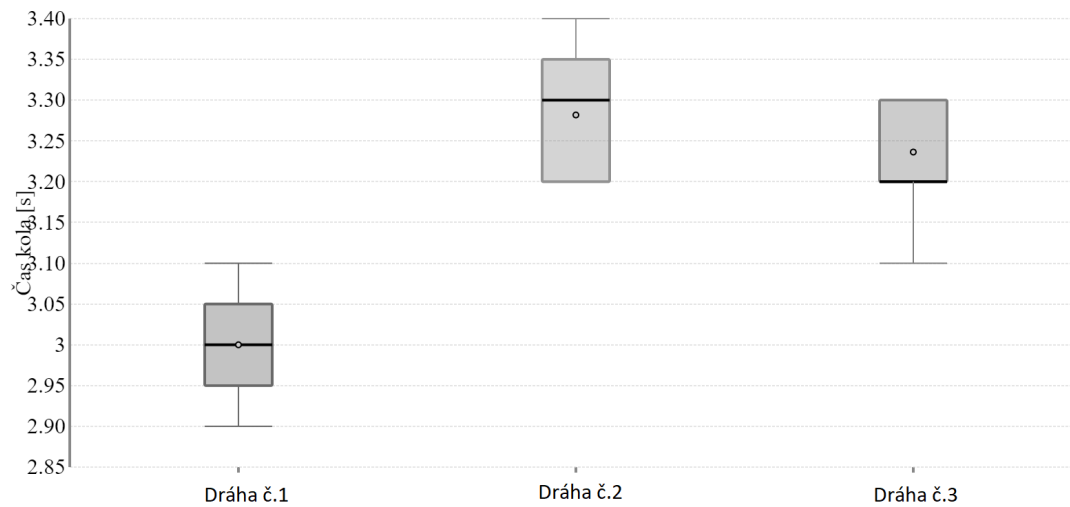
Po střídě 30,20% začalo autíčko dostávat smyk v zatáčkách u první dráhy, proto jsem dále s měřením nepokračoval. U druhé a třetí dráhy autíčko bylo možné dostat autíčko až na střidu ($\frac{80}{255}$) 31,37%.

Výpočet statických chyb [2.4]

Měření s proměnlivou rychlostí v zatáčkách

U tohoto pokusu bylo nejdříve nutné experimentovat s rychlostí v zatáčkách a rychlostí, kterou je možné dosáhnout, aby autíčko zvládlo zpomalit natolik, aby nevypadlo z dráhy v zatáčce.

Z experimentů se ukázalo, že autíčko může jet se střídou asi 33,73%, předtím než gyro-skop rozpozná zatáčku a stihne zpomalit na 27,45%.



Obrázek 5.3: Krabicový graf pro vyjádření časů změřených na dráhách

Zde lze vidět, že se dosahovalo poněkud lepších časů, ale při jednom pokusu nastal smyk. Nejspíše kvůli příliš pozdě rozpoznání zatáčky, což naznačuje špatně nastavené hranice pro rozpoznání zatáčky.

Výpočet statických chyb [2.4]

Dráha č.1:

1. $\bar{x} = \frac{1}{11}(3.1 + 3.0 + 3.1 + 2.9 + 2.9 + 3.0 + 3.1 + 3.0 + 3.0 + 2.9 + 3.0) = \underline{3}$
2. $s = \sqrt{\frac{1}{10} \sum_{i=1}^{11} (x_i - 3)^2} \approx \underline{0,07746}$ tedy musí platit $(x_i - 3) \leq 0,2324$
3. počet měření není < 10
4. Autíčko s tímto algoritmem zajede jedno kolo za $3 \pm 0,2324$ vteřiny.

Dráha č.2:

1. $\bar{x} = \frac{1}{11}(3.3 + 3.2 + 3.2 + 3.2 + 3.3 + 3.4 + 3.2 + 3.3 + 3.4 + 3.4 + 3.2) \approx \underline{3,282}$
2. $s = \sqrt{\frac{1}{10} \sum_{i=1}^{11} (x_i - 3,282)^2} \approx \underline{0,08738}$ tedy musí platit $(x_i - 3,282) \leq 0,3082$
3. počet měření není < 10
4. Autíčko s tímto algoritmem zajede jedno kolo za $3,282 \pm 0,3082$ vteřiny.

Dráha č.3:

1. $\bar{x} = \frac{1}{11}(3.2 + 3.2 + 3.3 + 3.2 + 3.2 + 3.3 + 3.3 + 3.2 + 3.3 + 3.1 + 3.3) \approx \underline{3,236}$
2. $s = \sqrt{\frac{1}{10} \sum_{i=1}^{11} (x_i - 3,236)^2} \approx \underline{0,08257}$ tedy musí platit $(x_i - 3,236) \leq 0,2569$
3. počet měření není < 10
4. Autíčko s tímto algoritmem zajede jedno kolo za $3,236 \pm 0,2569$ vteřiny.

Zhodnocení výsledků

I přesto, že jsou další dvě dráhy asi o čtvrtinu ($\frac{3,56}{2,87} = 1,24$) delší tak autíčko bylo pomalejší jen o desetinu vteřiny. Zde lze vidět, že oproti první dráze jsou zatáčky méně intenzivní tudíž se nimi autíčko dokázalo pohybovat vyšší rychlostí.

Kapitola 6

Závěr

Cílem práce bylo navrhnout destičku a algoritmus pro zmapování neznámého tvaru autodráhy a dále tyto informace použít pro co nejrychlejší projetí autodráhy.

Pro vývoj bylo vhodné prozkoumat jiná řešení a různé implementační metody. Hlavním zdrojem informací byla soutěž Freescale race challenge, protože se tato soutěž zabývala stejnou problematikou.

V práci byli popsány vlastnosti autodráh, vlastnosti autíček, použitá řešení v soutěži Freescale Race Challenge 2009 - 2012, principy související s řízením elektromotorů, funkce čidel a zpracování dat ze senzorů.

Bylo zvoleno Arduino NANO 33 BLE jako vývojová platforma kvůli tomu, že disponuje 9-ti osým senzorem, kvalitním vývojovým prostředím, malými rozměry a velkým výkonem. Až později se ukázalo, že není úplně vhodná pro tuto aplikaci. Také byli vybrány další komponenty, které byly nezbytné k vytvoření algoritmu. Byli vyzkoušeny různé typy filtrů. Dále byla navržena datová reprezentace autodráhy. Při práci byli vyzkoušeny funkce akcelerometru a gyroskopu. Vyzkoušeny byly také senzory detekující magnety a magnetické pole (Hallova sonda a magnetometr). Oproti Freescale race challenge jsem přidal magnet detekující začátek/konec dráhy, díky čemuž můžu pomoci při zpřesnění začátku algoritmu a zároveň díky tomuto vylepšení je možné projíždět dráhu vícekrát i s drobnými chybami při každém průjezdu.

6.1 Další vývoj

6.1.1 Program

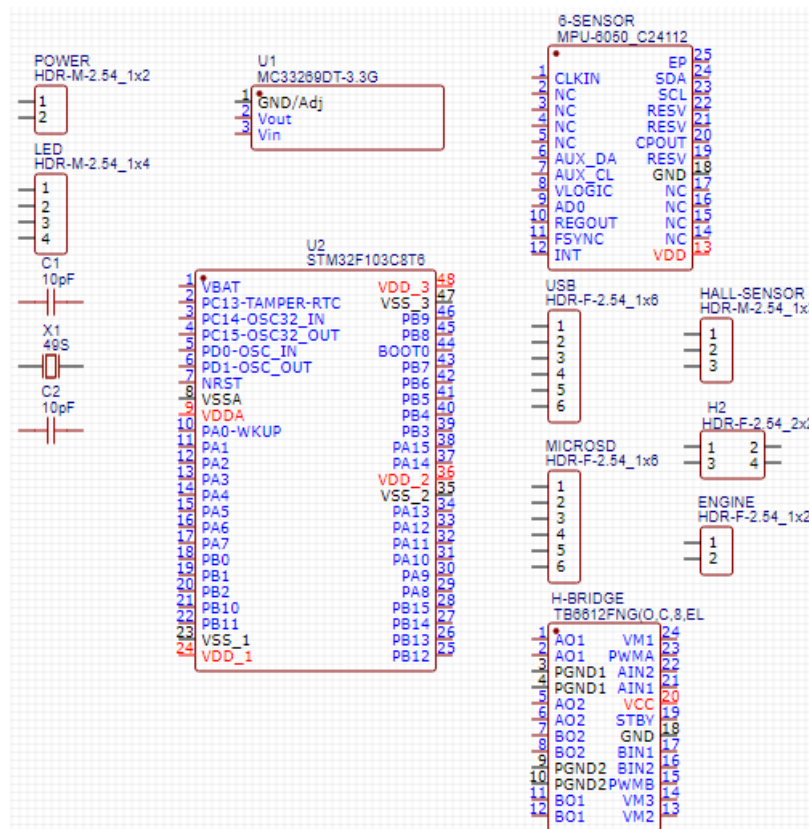
Další vývoj by se věnoval upravením hodnot v programu, což se týče především výpočtu zrychlení v zatáčkách. Další je předělání struktury zaznamenávání zatáček, aby program mohl zaznamenávat výjezdy ze zatáček a vjezdy do zatáček. Poté upravit algoritmus na projíždění dráhou, aby upravoval maximální rychlost jízdy v zatáčce na základě dat ze senzorů a takto by se autíčko mohlo zrychlovat v průběhu jízdy.

Detekování začátku dráhy by se dalo upravit, aby to šlo i bez magnetu nebo jiného externího prvku. To by bylo potřeba vyvinout algoritmus, který by při projíždění dráhy zaznamenával nadále (i po projetí prvního kola) zatáčky, mapoval dráhu a tyto hodnoty použil pro úpravu začátků a konců zatáček.

6.1.2 Platforma

Vývoj programu dále závisí na volbě použitých komponent. U dalšího vývoje bych počítal s výměnou platformy a komponent za vlastní navrhnout desku, přičemž by její vývoj mohl trvat déle, ale výsledek by mohl být mnohem rychlejší, přesnější a více upraven pro tento účel (v tomto případě by se musel upravit i program nebo navrhnout vlastní knihovny pro komunikaci s SD kartou a čtením ze senzorů).

Další návrh by zahrnoval také použité komponenty v této práci jako například slot pro microSD kartu a Hallův senzor. Celkový návrh by mohl být menší díky přidání H-můstku na desku (v práci je použit externí). Mikrokontrolér byl vybrán tak, aby byl dostatečně rychlý (podporuje externí krystal) a je možné jej programovat bez potřeby zakoupení velkého programátoru. Návrh součástek by mohl například vypadat takto:



Obrázek 6.1: Ukázka dalšího vývoje práce (tento vzor byl původně zamýšlen jako řešení, ale kvůli trvání dodání a velké šanci neúspěchu byl zvolen bezpečnější způsob)

Literatura

- [1] MICROCHIP. *21754M* [online]. 2007 [cit. 2020-12-23]. Dostupné z: <https://ww1.microchip.com/downloads/en/DeviceDoc/21754M.pdf>.
- [2] INTERCONNECTS. *Card Connector microSD™* [online]. 2006 [cit. 2020-12-23]. Dostupné z: <https://cdn.sos.sk/productdata/4a/60/75fe3aaf/2908-05wb-mg.pdf>.
- [3] KRÍŽ, J. *Dálkové ovládání s inerciálními senzory* [online]. 2013. [cit. 2020-12-25]. Diplomová práce. Univerzita Pardubice fakulta elektrotechniky a informatiky. Dostupné z: https://dk.upce.cz/bitstream/handle/10195/53714/KrizJ_DalkoveOvladani_PR_2013.pdf?sequence=4&isAllowed=y.
- [4] ARDUINO. *Arduino NANO 33 BLE* [online]. [cit. 2021-01-15]. Dostupné z: <https://store.arduino.cc/arduino-nano-33-ble>.
- [5] CARRERA. *Stránky firmy Carrera* [online]. 2020 [cit. 2020-12-22]. Dostupné z: https://www.carrera.cz/m/katalog/87_carrera-evolution-dily-drahy.
- [6] DC.COMPONENTS. *Bridge rectifier* [online]. 2006 [cit. 2020-12-23]. Dostupné z: http://www.ges.cz/sheets/d/db101_107s_dc.pdf.
- [7] KRÁLOVÁ, M. *Elektromotory* [online]. 2006 [cit. 2020-12-23]. Dostupné z: <https://edu.techmania.cz/cs/encyklopedie/fyzika/elektromagneticka-indukce/elektromotory>.
- [8] VALA, Z. *Stejnoseměrné elektromotory* [online]. 2006 [cit. 2020-12-23]. Dostupné z: http://www.sossoukyjov.cz/data/File/VY_32_INOVACE_6a13.pdf.
- [9] FARO. *Faro - česká autodráha* [online]. 2020 [cit. 2020-12-22]. Dostupné z: <https://www.autodraha-faro.cz/>.
- [10] BREJL, M. *Freescale Race Challenge 2009* [online]. 2008 [cit. 2020-12-22]. Dostupné z: <https://vyvoj.hw.cz/teorie-a-praxe/freescale-race-challenge-2009-soutez-samoridicich-auticek-na-autodrahu.html>.
- [11] BREJL, M. *Freescale Race Challenge 2010* [online]. 2009 [cit. 2020-12-22]. Dostupné z: <https://vyvoj.hw.cz/procesor/freescale-race-challenge-2010-novy-rocnik-souteze-samoridicich-auticek-na-autodrahu.html>.
- [12] BREJL, M. *Freescale Race Challenge 2011* [online]. 2010 [cit. 2020-12-23]. Dostupné z: <https://vyvoj.hw.cz/teorie-a-praxe/freescale-race-challenge-2011-dalsi-rocnik-souteze-samoridicich-auticek-na-autodrahu>.

- [13] PAŠTYKA, J. *DC motory řízené pomocí H-můstku* [online]. ČVUT Fakulta elektrotechnická, 2017. Dostupné z: https://embedded.fel.cvut.cz/sites/default/files/kurzy/lpe/h-bridge/H_Bridge.pdf.
- [14] REICHL, J. Hallův jev. In: *Encyklopedie Fyziky* [online]. [cit. 2020-12-27]. Dostupné z: <http://fyzika.jreichl.com/main.article/view/294-halluv-jev>.
- [15] VACULÍK, V. *Snímače proudu* [online]. 2011. [cit. 2020-12-27]. Diplomová práce. Vysoké učení technické v Brně fakulta elektrotechniky a komunikačních technologií. Dostupné z: <https://core.ac.uk/download/pdf/30298342.pdf>.
- [16] *Hallův jev* [online]. 2013 [cit. 2020-12-27]. Dostupné z: https://www.wikiskripta.eu/w/Hall%C5%AFv_jev.
- [17] JARMILA MILLEROVÁ, K. V. a, ed. *Pokyn pro vyjadřování nejistoty měření (GUM)*. Gorazdova 24, 128 01 Praha 2, Praha 2012: Úřad pro technickou normalizaci, metrologii a státní zkušebnictví, 2012. Sborníky technické harmonizace. Dostupné z: <https://www.unmz.cz/files/Sborn%C3%ADky%20TH/GUM%20-%20celek%20-DEF.pdf>.
- [18] STMICROELECTRONICS. *LSM9DS1* [online]. 2013 [cit. 2021-01-15]. Dostupné z: https://content.arduino.cc/assets/Nano_BLE_Sense_lsm9ds1.pdf.
- [19] FREESCALE. *MC9S08JM32* [online]. 2007 [cit. 2020-12-23]. Dostupné z: <https://pdf1.alldatasheet.com/datasheet-pdf/view/352444/FREESCALE/MC9S08JM32.html>.
- [20] FREESCALE. *MCF51JM128RM* [online]. 2008 [cit. 2020-12-23]. Dostupné z: <https://www.nxp.com/docs/en/reference-manual/MCF51JM128RM.pdf>.
- [21] FREESCALE. *MCZ33887* [online]. 2007 [cit. 2020-12-23]. Dostupné z: <https://datasheetspdf.com/pdf-file/593526/FreescaleSemiconductor/MCZ33887/1>.
- [22] NXP. *MMA7361L* [online]. 2008 [cit. 2020-12-23]. Dostupné z: <https://www.nxp.com/docs/en/data-sheet/MMA7361L.pdf>.
- [23] KHESTL, F. *Nejistota měření - postup stanovení* [online]. [cit. 2022-02-25]. Dostupné z: https://home1.vsb.cz/~khe0007/opory/opory.php?stranka=nejistota_postup.
- [24] NORDICSEMICONDUCTOR. *NRF52840* [online]. 2018 [cit. 2021-01-15]. Dostupné z: https://content.arduino.cc/assets/Nano_BLE_MCU-nRF52840_PS_v1.1.pdf.
- [25] NXP. *Freescale Semiconductor ČR* [online]. 2020 [cit. 2020-12-22]. Dostupné z: <https://www.nxp.com/>.
- [26] *Understanding PWM* [online]. 2009 [cit. 2020-12-25]. Dostupné z: <https://ebldc.com/?p=48>.
- [27] RUDOLF PALENČÁR, F. V. a. M. H. Nejistoty v měření I: vyjadřování nejistot. *Automa*. 2001, 7-8, s. 2–6. Dostupné z: https://automa.cz/Aton/FileRepository/pdf_articles/10960.pdf.
- [28] RUDOLF PALENČÁR, F. V. a. M. H. Nejistoty v měření II: nejistoty přímých měření. *Automa*. 2001, sv. 10, s. 52–56. Dostupné z: https://automa.cz/Aton/FileRepository/pdf_articles/10961.pdf.

- [29] RUDOLF PALENČÁR, F. V. a. M. H. Nejistoty v měření III: nejistoty nepřímých měření. *Automa*. 2001, sv. 12, s. 1–6. Dostupné z:
https://automa.cz/Aton/FileRepository/pdf_articles/10962.pdf.
- [30] RUDOLF PALENČÁR, F. V. a. M. H. Nejistoty v měření IV: nejistoty při kalibraci a ověřování. *Automa*. 2001, sv. 4, s. 41–47. Dostupné z:
https://automa.cz/Aton/FileRepository/pdf_articles/10963.pdf.
- [31] TOMÁŠ PANC, O. T. a. Možnosti vyjádření přesnosti měření I: teoretický základ. *Automa*. 2013, sv. 7, s. 33–35. Dostupné z:
https://automa.cz/Aton/FileRepository/pdf_articles/10548.pdf.
- [32] TOMÁŠ PANC, O. T. a. Možnosti vyjádření přesnosti měření II: použití v praxi. *Automa*. 2013, sv. 7, s. 48–52. Dostupné z:
https://www.automa.cz/Aton/FileRepository/pdf_articles/10914.pdf.

Příloha A

Obsah přiloženého média

1. „BP_text.pdf“
2. složka „Technická zpráva“ – obsahuje soubory potřebné k vytvoření pdf verze práce
3. složka „Zdrojový kód“ – obsahuje zdrojový kód (src.ino) a popis (README.txt) ke zprovoznění
4. složka „Videa“ obsahující videa autíčka