FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Název:	Dálkově ovládané čtyřkolové vozítko využívající platformu Arduino
Student:	Martin Zemánek
Vedoucí:	Ing. Pavel Kubalík, Ph.D.
Studijní program:	Informatika
Studijní obor:	Počítačové inženýrství
Katedra:	Katedra číslicového návrhu
Platnost zadání:	Do konce letního semestru 2019/20

Pokyny pro vypracování

1) Prozkoumejte existující řešení dálkově ovládaných čtyřkolových vozítek.
2) Navrhněte vlastní řešení řízení vozítka s pomocí platformy Arduino.
3) Aplikace pro Arduino bude umožňovat řízení každého kola tak, aby bylo možné jezdit všemi směry.
4) Veškeré řízení pohybu vozítka bude provedeno pomocí dálkového ovládání.
5) Vozítko bude na sobě obsahovat další senzory potřebné pro pohyb a LED diody pro signalizaci směru.
6) Navržené řešení zrealizujte a řádně otestujte.

Seznam odborné literatury

Dodá vedoucí práce.

doc. Ing. Hana Kubátová, CSc.
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
děkan

V Praze dne 5. února 2019

FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

Bakalářská práce

Dálkově ovládané čtyřkolové vozítko založené na platformě Arduino

Martin Zemánek

Katedra Počítaočového Inženýrství
Vedoucí práce: Ing. Pavel Kubalík, Ph.D.

16. května 2019

Poděkování

Rád bych poděkoval Panu Inženýru Pavlu Kubalíkovi, za jeho trpělivost se mnou.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo"), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 16. května 2019 ……

Odkaz na tuto práci

Zemánek, Martin. *Dálkově ovládané čtyřkolové vozítko založené na platformě Arduino*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2019.

Abstrakt

Práce se zabývá využitím platformy Arduino k sestavení dálkově ovládaného čtyřkolého vozítka. Následně je v této práci navržen, sestaven a otestován prototyp takového vozítka.

Klíčová slova Arduino, Vozítko, Dálkové ovládání, H-můstek, čtyřkolé vozítko

Abstract

In this project are explored possibilities of useage Arduino platform for building remote controlled vehicle. Then in this project is designed, build and tested such vehicle.

Keywords Arduino, Vehicle, Remote control, H-bridge, 4WD vehicle

Obsah

Seznam obrázků

Úvod

Na začátku této práce jsem stál před úkolem rozhýbat čtyřkolý podvozek za použití platformy Arduino.

Arduino je harwarová platforma založena na mikroprocesorech ATmega. Jedná se o velmi snadno použitelnou platformu. K tvorbě aplikací na této platformě je zapotřebí znalost jazyka C, nebo C++. Výhodu této platformy je, že pro ni existuje již mnoho knihoven, což podstatně usnadňuje vývoj. Protože platforma Arduino zahrnuje několik možných vývojových desek, zabývám se v této práci výběrem desky.

V této práci se nezabývám tvorbou a návrhem vlastního ovladače, ani přijímače. Také se v této práci nezabývám tvorbou vlastních senzorů, motorů, či částí potřebných k jejich ovládání. Zabývám se ovšem výběrem vhodných dostupných komponent a jejich připojením k vybrané řídící desce.

Kapitola 1

Cíl práce

Mým hlavním cílem v této práci bylo navrhnout, zkonstruovat a otestovat prototyp čtyřkolého vozítka založeném na platformě Arduino. Abych tento cíl mohl uskutečnit musel jsem prozkoumat již existující řešení, a to jak čtyřkolého vozíka, tak pro zapojení jednotlivých komponent, které jsem se rozhodl použít.

Minimální funkcionalita prototypu je proto komunikace s dálkovým ovládáním a ovládání motorů. Další funkcionalitou tohoto prototypu je světelná signalizace směru a schopnost reagovat na dodatečné senzory.

Kapitola 2

Rešerše

Na počátku práce bylo potřeba najít řešení obdobných problémů. V této kapitole popisuji řešení podobných problémů, které jsem nalezl. Při svém výzkumu jsem se zaměřil na platformu Arduino.

2.1 Arduino Powered Autonomous Vehicle

Prvním řešením, které jsem nalezl je Arduino Powered Autonomous Vehicle. Tento projekt používá jako základ podvozek z RC auta na dálkové ovládání. Toto řešení však dálkové ovládání nepoužívá, místo toho využívá GPS senzor, kompas, senzor pro vzdálenost a nastavený bod v prostoru k navigaci. Nevýhodou tohoto vozítka je nemožnost ovládání vozítka na větší vzdálenost, než je dosah infračerveného senzoru, nicméně jeho výhodu je, to že vzhledem k řešení tohoto vozítka to ani není potřeba. Hlavní výhodou tohoto vozítka je, že při správné funkci nemusí být přímo ovládáno. [1]

2.2 Arduino Bluetooth RC Car (Android Controlled)

Dalším řešením, které jsem nalezl je Arduino Bluetooth RC Car (Android Controlled). Toto řešení je opět vystaveno na podvozku běžného RC auta. Toto řešení k ovládání využívá bluetooth modul a aplikaci na android. Hlavní nevýhodou tohoto řešení je to, že vozítko není schopno nijak reagovat na okolí a překážky. Výhodou tohoto řešení je snadná náhrada ovládání, protože stačí libovolné zařízení s operačním systémem android a funkcí bluetooth. [2]

2.3 Smartphone Controlled Arduino 4WD Robot Car

Následující řešení, které jsem našel je Smartphone Controlled Arduino 4WD Robot Car. Jedná se o čtyřkolové vozítko, vystavené na jiném typu podvozku než ostatní řešení. I když se jedná o vozítko s čtyřkolovým podvozkem, vozítko má náhon na všechna čtyři kola. Každé z těchto kol má pak pevnou osu otáčení. Toto způsobuje, že vozítko se musí ovládat jiným způsobem, než předchozí řešení. Toto vozítko je stejně jako v předchozím případě ovládáno přes bluetooth pomocí aplikace na mobil. [3]

Kapitola 3

Analýza

Na začátku této kapitoly se snažím rozebrat zadání na jednotlivé podproblémy. Následně z toho odvozuji, jaké problémy budu v práci muset řešit. V druhé části této kapitoly pak hledám součástky, které řeší jednotlivé podproblémy. Součástky, které řeší stejný podproblém, pak mezi sebou porovnávám.

3.1 Rozbor problémů

V principu bude vozítko rozděleno do několika komponent. Každá z těchto komponent bude mít své vstupy a výstupy. Je nutné poté zajistit správnou komunikaci mezi komponentami.

První komponentou je řídící jednotka. Tato jednotka musí umět přijímat čtení ze senzorů. Poté musí umět toto čtení zpracovat. Na konec musí tato jednotka vyslat požadované signály svým periferiím, čímž změní chování celku. Tato komponenta je zodpovědná za správné chování ostatních komponent.

Následující jednotkou je ovládání motorů. Ovládání motorů bude přijímat signály od řídící jednotky. Tyto signály zpracuje a jako výstup bude mít vstupní napětí pro motory. Tato jednotka je zodpovědná, za správnou reakci motorů na výstup řídící jednotky.

Mezi řídící jednotkou a uživatelem musí být nějaké rozhraní. V případě tohoto projektu se jedná o kombinaci vysílače a přijímače. Tuto kombinaci je nutné zvolit tak, aby řídící jednotka zvládla zpracovat signál z přijímače. Je samozřejmě důležité, aby zvládl komunikovat přijímač s vysílačem. Z tohoto signálu je nutné vytvořit signály pro řízení motorů a další komponenty.

Chování vozítka budou ovlivňovat senzory. Tyto senzory budou propojeny s řídící jednotkou. Různé senzory mají jiná rozhraní. Je nutné zvolit senzory s použitelným rozhraním. Jejich výstupy a případné vstupy je nutné propojit s řídící jednotkou, tak aby se daly přečíst jejich výstupy. Rozhraní mezi uživatelem a řídící jednotkou je také v principu senzor, nicméně vyžaduje

specifičtější režii, než ostatní senzory. Řídící jednotka se pak musí postarat o úpravu výstupu, v závislosti na čtení ze senzorů.

Poslední jednotkou je světelná signalizace. Jedná se, obdobně jako u řízení motorů, o výstupní jednotku. Účelem této jednotky je zpracovat signál z řídící jednotky a rozsvítit příslušnou signalizaci.

3.2 Možné součástky

3.2.1 Řídící deska

Jak jsem se zmínil dříve jedná se o projekt založený na platformě Arduino. Arduino budu v toto projektu využívat k řízení vozítka. Od volby konkrétního Arduina se odvíjí počet a typ periferií, které bude v projektu dále možno použít. Základní deskou, která se velmi často využívá je Arduino Uno. Současná nejnovější verze je Arduino Uno Rev3. Arduino Uno Rev3 je založeno na mikrokontroleru ATmega328P. [4]

Deska Arduina Uno Rev3 jako celek obsahuje 14 digitálních pinů, z nichž 6 dokáže vytvořit PWM výstup. Dále obsahuje 6 analogových pinů. Tato deska pracuje na frekvenci 16MHz. Deska pracuje s napájením 5V a je napájení a je napájená stejnosměrným napětím mezi 6V a 20V. Doporučené napájení je však mezi 7V a 12V. [5]

Zajímavou variantou pro projekty, vystavěné na platformě Arduina Uno je Arduino Uno Wifi rev2. Jedná se o desku vystavenou na mikrokontroleru ATMEGA4809.[6] Výhoda této desky je v základu vestavěný Wi-fi modul, který umožňuje komunikaci pomocí protokolu TCP/IP s ostatními zařízeními. Tato deska může být napájena pouze stejnosměrným napětím mezí 6V a 12V, s doporučeným napětím mezi 7V a 12V. Tato deska obsahuje také 14 digitálních I/O pinů, ovšem pouze 5 z nich dokáže vytvořit na výstupu PWM signál. Také obsahuje 6 analogových pinů, také pracuje s vnitřním napětím 5V a také pracuje na frekvenci 16MHz. [7]

Další často používanou variantou je Arduino Due. Adruino Due je deska založena na mikrokontroleru Atmel SAM3X8E ARM Cortex-M3.[8] Jednou z odlišujících vlastností této desky je to, že pracuje s napětím 3,3V místo 5V obvyklých u ostatních desek. Tato deska je napájena stejnosměrným napětím mezi 6V a 16V. Doporučené napájení této desky je také mezi 7V a 12V. Tato deska na rozdíl od předchozích desek obsahuje 54 digitálních I/O pinů. 12 z těchto pinů dokáže produkovat na výstupu PWM signál. Tato deska pracuje také s jinou frekvencí hodin. Tato deska pracuje na frekvenci 84 MHz. [9]

Další často používanou deskou je Arduino Mega 2560. Nejnovější verzí Arduina Mega 2560 je třetí revize. Tato deska je založena na mikrokontroleru ATmega2560.[10] Tato deska také obsahuje 54 pinů, stejně, jako Arduino Due, nicméně pracuje s napětím 5V. Tuto desku je možné napájet stejnosměrným napětím mezi 6V a 20V, ale doporučené napětí je mezi 7 a 12V, tedy se stejnými hodnotami, jako Arduino Uno. Na rozdíl od Arduina Due 15 pinů

je schopno na výstupu produkovat PWM signál. Dále tato deska obsahuje 16 analogových vstupních pinů. Tato daska stejně, jako Arduino Uno pracuje s frkvencí hodin 16MHz. [11]

3.2.2 Podvozek

Existuje více variant podvozku. Od podvozku se odvíjí princip a složitost ovládání. Také se od principu podvozku odvádí počet ,motorů, které je potřeba ovládat. Jak je zřejmé z mnou nalezených řešení tohoto projektu existuje více principů, na kterých se projekt může zakládat. Pro čtyřkolé podvozky jsou běžné dva typy řešení. Buď existuje pár kol, který nemá pevnou osu otáčení, nebo všechna kola mají pevnou osu otáčení. U většiny RC aut na dálkové ovládání mají dvě kola s pevnou osou otáčení a dvě kola, které tuto osu mohou měnit. Výhodou takového řešení je možnost náhonu pouze na dvě kola. Tato dvě kola mohou být poháněna pouze jedním motorem. Tento podvozek však neumožňuje otáčení na místě.

Dalšími čtyřkolými podvozky, které jsou běžně k dostání jsou podvozky, kde všechna kola mají pevnou osu otáčení. Výhodou takového podvozku je, že umožňuje otáčení na místě. Nevýhodou takového podvozku je pak složitější ovládání, protože chceme-li s podvozkem mírně zatáčet, znamená to v určitém poměru rozdělit výhon mezi dvojice kol na stejné straně vozidla. Ovládáním se takové vozítko podobá pásovému vozítku.

Pro první variantu podvozku existuje mnoho příkladů implementace. Takový podvozek se používá od jednoduchých RC aut na dálkové ovládání, až po auta skutečně používaná v dopravě. Jak bylo dříve řečeno tento podvozek neumožňuje některé manévry, avšak jeho výhodou je jednoduché ovládání. Projekty „Arduino Powered Autonomous Vehicle“ [1] a „Arduino Bluetooth RC Car (Android Controlled)“ [2] také využívají tento typ podvozku.

Druhý zmíněný podvozek umožňuje velkou škálu jednoduchých manévrů, avšak manévry, které jsou snadné s předchozím podvozkem vyžadují obtížnější řízení. Například pro mírné zatočení doprava u prvního typu podvozku vyžaduje pouze nastavení polohy prvního motoru a poté nastavení rychlosti druhého motoru, kdežto u druhého typu podvozku je míra otáčení vozidla závislá na nastavení rychlosti a směru otáčení dvojice motorů na jedné straně vozidla, oproti rychlosti a směru otáčení druhé dvojice motorů na druhé straně vozidla. Příkladem projektu využívající tento typ podvozku je „Smartphone Controlled Arduino 4WD Robot Car“ [3].

3.2.3 Ovládání motorů

Vzhledem k tomu, že platforma Arduino využívá ke své logice stejnosměrný proud, je vhodné použít i stejnosměrný motor. Takové motory je možné regulovat dvěma způsoby.

Prvním způsobem je změna velikosti napájecího napětí motoru. Tento způsob není vhodné využívat, protože se při něm motor zahřívá a dochází k energetickým ztrátám. Tento způsob také není vhodné využívat ve spojením s platformou Arduino, protože každá deska pracuje s pevným napětím.

Vhodnějším způsobem je ovládání motorů pomocí PWM signálu. Tento způsob je vhodnější, protože pracuje při stejném napětí. Dále je vhodnější pro platformu Arduino, protože většina desek s platformou arduino obsahuje vestavěné piny, které umožňují generovat PWM signál. K ovládání motorů pomocí PWM signálů se poté využívá H můstek, což je součástka, která umožňuje převracet polaritu daného signálu a tím ovládat směr otáčení motoru.[12] Často se H můstky dodávají na deskách a to nejčastěji ve dvojicích.

Jednou z těchto desek je Adafruit motor / stepper / servo shield - V2.3. Tato deska je založena na H můstku TB6612. Tato deska obsahuje celkem čtyři H můstky tohoto typu, díky čemuž je k ní možné připojit až čtyři stejnosměrné motory. Tato deska pracuje se stejnosměrným napájením mezi 4,5V a 13,5V.[13]

Další často užívanou deskou je L298N. Tato deska je založena na integrovaném spoji L298.[14] Jedná se o integrovaný obsahující dva H můstky. Deska jako celek tedy umožňuje ovládat až dva stejnosměrné motory. Tato deska může pracovat s napětím mezi 5V a 35V. Dále krom H můstku tato deska obsahuje 5V regulátor, který je možno využít, pokud deska pracuje s napětím nižším než 12V. Tento regulátor pak umožňuje použít tuto desku, jako stejnosměrný zdroj 5V napětí pro zbytek obvodu.[15]

3.2.4 Dálkové ovládání

Kritickou částí mého projektu je dálkové ovládání. S vedoucím jsme se dohodly, že budu využívat dálkové ovládání spektrum DX5e.[16] K tomuto ovládání je tedy třeba najít přijímač. Standartní přijímač k tomuto ovládání je AR500. Tento přijímač stejně jako vysílač vyrábí firma spektrum. Tento přijímač je třeba napájet stejnosměrným napětím 3,5V až 9,6V. Jedná se o pětikanálový přijímač.[17] Dalším přijímačem pro tento vysílač je MK610 od společnosti Mkron. Tento přijímač je ve většině parametrů podobný předchozímu přijímači. Jeho podstatnou výhodou je pořizovací cena. [18]

3.2.5 Světelná signalizace

Ke světelné signalizaci je možné využít několik možností. K Arduinu se nejčastěji připojují svítivé LED diody, které mají nízkou spotřebu energie a mají nízký odpor, díky čemuž se nezahřívají tolik, jako ostatní možná řešení. Tuto diodu je však nutné zapojit do série s ochranným rezistorem. Tento způsob je využit například v projektu „Smartphone Controlled Arduino 4WD Robot Car“. [3]

3.2.6 Vnímání okolí

Vnímat okolí lze několika způsoby. Častým způsobem vnímání okolí je vzdálenostní senzor. Často využívanou variantou je senzor HC-SR04. Jedná se o desku obsahující ultrazvukový vysílač a přijímač. Tato deska pracuje s napětím 5V. Tato deska je schopna rozpoznat překážku na vzdálenost až 4m. [19]

Dalšími způsoby vnímání okolí je například náraz, kde se na tlačítko přidělá nárazník, a nárazem se tlačítko sepne. Tento způsob je použitelný pouze pro nízké rychlosti.

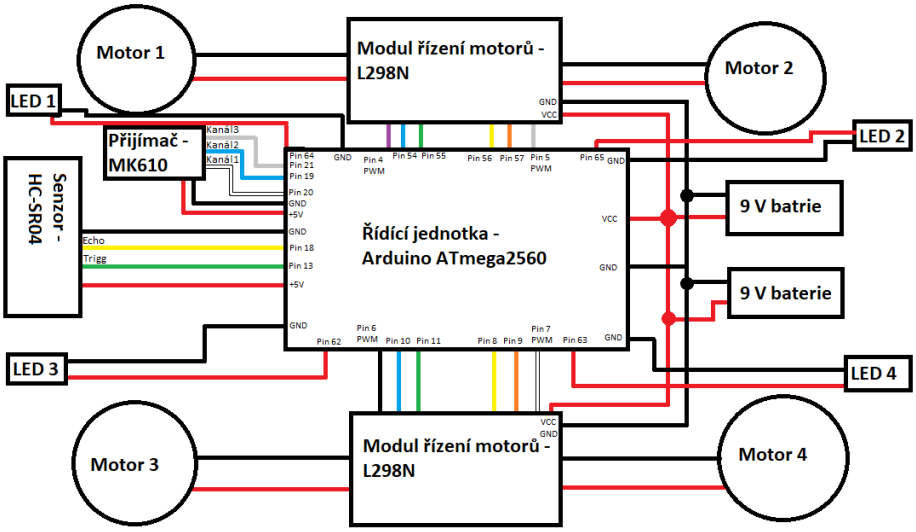
Zajímavými možnostmi pro vnímání okolí jsou použité v projektu „Arduino Powered Autonomous Vehicle“[1], který jsem dříve zmínil. V tomto projektu používá autor kombinaci kompasu, GPS modulu a již dříve zmíněného vzdálenostního senzoru. Existují i další způsoby vnímání okolí, nicméně tyto způsoby mi přišli, jako nejvíce relevantní k projektu.

Kapitola 4

Návrh

Prototyp bude obsahovat několik modulů. Některé budou programovatelné. Je třeba prvně navrhnout, jaké moduly budou potřeba a jak budou propojeny Tímto se zabývá tato kapitola.

4.1 Moduly



Obrázek 4.1: HW - blokové schéma

Ústředním modulem celého prototypu bude řídící modul. K tomuto modulu bude připojen zbytek modulů. Účelem tohoto modulu bude zpracovat informace ze všech vstupů a nastavit odpovídajícím způsobem všechny výstupy.

Na prototypu budou dva typy modulů, které vytvářejí výstup. prvním typem takového modulu jsou motory. Mezi motory a řídící jednotkou bude modul pro zpracování řídících signálů motorů, Tento modul bude přijímat výstup řídícího modulu a poté předělá tento signál na výstup, který bude ovládat motory.

Druhým typem výstup vytvářejícího modulu je světelná signalizace. Tento modul bude připojený přímo na řídící modul. Tento modul bude brát výstup z řídícího modulu a ten bude přetvářet přímo na výstup.

Na prototypu budou také dva moduly, které přijímají vstup a předají ho řídící jednotce ke zpracování. Prvním typem takového modulu je senzor. Tento modul bude měřit okolí a naměřené hodnoty posílat řídící jednotce.

Druhý takovýto modul je přijímač. Přijímač přečte hodnoty vysílané vysílačkou. Tyto hodnoty pak předá řídícímu modulu.

4.2 Program

Hlavní programovatelným modulem celého návrhu je řídící modul. Ten zpracovává všechny vstupy a vytváří všechny výstupy. Proto navrhuji program výhradně pro ni.

Jelikož máme vstupy, které nečekají na požádání, je třeba mít program složený z více částí. První vlákno bude obstarávat výstupy. Prvně se tedy do paměti nahrají vstupy, spočítají se hodnoty výstupů a poté se tyto hodnoty pošlou na výstup. Senzory také mohou vyžadovat aktivaci. To také provedeme v této části programu.

Další části programu pak budou pasivně čekat na vstupy. V momentě, kdy program dostane vstup se tento kus kódu provede a zastaví vše ostatní na délku svého trvání. V těchto kusech programu je nepraktické provádět výpočty, proto tyto části programu pouze zaznamenají čtené hodnoty. Přepočítání se provede v první části programu.

Takto přibližně bude vypadat program.

Kapitola 5

Implementace

V této kapitole se zabývám prvně moduly, použitými pro realizaci prototypu. Následně způsobem, jakým moduly propojit, aby prototyp fungoval. Na konec se v této kapitole zabývám programem, který požívám pro provoz prototypu.

5.1 Zvolené Díly

5.1.1 Arduino

Svůj projekt jsem se rozhodl využít Arduino Mega 2560 rev. 3.[11] Učinil jsem tak, protože pracuje s napětím 5V a přesto má velké množství pinů, které pro tento projekt budu potřebovat Více informací k této desce jsem již dříve zmínil.

5.1.2 Podvozek

Pro svůj projekt jsem se rozhodl využít podvozek „Smart Car Kit 4WD Smart Robot Car Chassis Kits with Speed Encoder and Battery Box for arduino Diy Kit“.[20] Jedná se o podobný podvozek, jako je využit v projektu „Smartphone Controlled Arduino 4WD Robot Car“ [3].

Jedná se o podvozek se čtyřmi koly. Každé z těchto kol má svůj stejnosměrný motor. Tento podvozek obsahuje také enkodery, pro měření rychlosti a směru otáčení. Podvozek je také velmi dobře uzpůsoben pro připevňování dalších částí. [20]

Tento podvozek jsem se rozhodl zvolit, protože s tímto typem podvozku je možné provézt manévry, jako je otočení na místě. Také je jednoduché na něj přidělat potřebné části.

5.1.3 Motory a jejich ovládání

Mnou zvolený podvozek již obsahuje čtyři stejnosměrné motory, které je třeba ovládat. Pro svůj projekt jsem se rozhodl využít princip manipulace stejnosměrných motorů pomocí PWM signálu, za využití H můstku. Jako tento H můstek jsem se rozhodl využít desku L298N.[15] Protože potřebuji hýbat čtyřmi motory rozhodl jsem se použít dvojici těchto obvodů. Pro své řešení jsem se rozhodl nevyužít tyto desky, jako zdroj napětí pro zbytek obvodu.

5.1.4 Dálkové ovládání

Jak bylo dříve řečeno rozhodli jsme se s vedoucím použít jako vysílač, vysílačku Spektrum DX5e.[16] Tuto vysílačku jsme zvolili, protože pro nás byla snadno dostupná. K této vysílačce jsem pak já zvolil přijímač MK610.[18] Oproti ostatním přijímačům použitelným pro zmíněné ovládání měl hlavní výhodu v ceně, protože byl levnější, než ostatní možnosti. Na výstupu tohoto zařízení se generuje PWM signál, pro každý kanál. Tyto signály je nutné pak zpracovat Arduinem.

5.1.5 Světelná signalizace

Pro tuto funkci jsem se rozhodl využít čtveřici svítivých LED diod, ke kterým, jak bylo již zmíněno, je nutné připojit vyrovnávací odpory.

5.1.6 Vnímání Okolí

Rozhodl jsem se, že pro vozítko bude nejpraktičtější vnímat Pro tento účel jsem se rozhodl pro již zmíněnou desku HC-SR04.[19] Tato deska, jak je již dříve zmíněno obsahuje ultrazvukový přijímač a vysílač. Pomocí těchto součástek pak deska generuje signál, který je třeba zpracovat Arduinem.

5.1.7 Napájení

Jelikož je vozítko pohyblivé, bylo jasné, že bude muset být napájeno z baterie. Toto vozítko jsem se rozhodl napájet dvojicí paralelně zapojených 9V baterek Philips 9VB1A17/10. Jedná se o dobíjecí 9V baterii. Hlavní důvod, proč jsem zvolil tuto baterii, je to, že tato baterie se dá snadno a bez změny konektoru nahradit dvojicí běžně dostupných 9V baterií. Dalším důvodem, proč jsem zvolil tuto baterii, je možnost dobít tuto baterii.[21]

5.2 Zapojení

Na začátku jsem připojil Arduino k bateriím. Jak jsem na začátku zmínil jedná se o dvojici 9V baterií. [21] K tomuto jsem využil VIN pin přímo na desce Arduina, který umožňuje napájení 7V až 12V. [11]

K Arduinu jsem musel připojit všechny periferie. Každý z modulů připojený k Arduinu používá jiné rozhraní.

Na začátku jsem se rozhodl připojit dvojici modulů L298N. K těmto modulům jsem pak připojil motory z podvozku. [20] Každý z těchto modulů vyžaduje napájení. Připojil jsem je také k výše zmíněné dvojici baterií. Následně jsem připojil k těmto modulům ovládací signály z Arduina. Každý z modulů vyžaduje šestici řídících signálů, tedy tři řídící signály na jeden motor. Jeden z těchto tří signálů je PWM signál. PWM signál jsou schopné generovat na Arduinu maga2560 rev3 pouze piny 2 až 13.[11] Pro své řízení motorů jsem se rozhodl využít piny 4 až 7. Dále je potřeba pro každý motor vybrat dva řídící signály, které je možné nastavit do stavu LOW a HIGH. Tato dvojice pinů slouží k ovládání směru otáčení. Vždy, když se má motor otáčet, musí být jeden ze dvojice signálů nastavený na HIGH a druhý na LOW. Pro tyto signály jsem se rozhodl zvolit dvojice pinů (54, 55), (56, 57), (10, 11) a (8, 9). Toto rozhodnutí jsem udělal, protože jsem chtěl, aby bylo možné každý z motorů ovládat zvlášť, i když toho v projektu později nevyužívám. Hlavním důvodem, proč jsem vybral právě tyto dvojice pinů, je dobré rozložení na desce.

Poté jsem zapojil přijímač vysílačky. Přijímač vysílačky MK610, který jsem se rozhodl využít potřebuje napájet 3,5V až 9,5V. Arduino Mega 2560 obsahuje pin, který konvertuje vstupní napětí Arduina na 5V. Pro napájení tohoto modulu jsem se rozhodl tento pin využít. Dále tento modul produkuje 5 PWM signálů. Informace v tomto signálu je zakódována, jako časová prodleva mezi vzestupnou a sestupnou hranou signálu. Ke čtení tohoto signálu jsem se rozhodl využít přerušení. Na Arduinu Mega 2560 je k tomuto účelu tímto způsobem možno využít pouze pouze piny 2, 3, 18, 19, 20 a 21. [22] I když se jedná o pěti-kanálový přijímač, pro ovládání takového vozítka byli potřeba pouze tři kanály, proto jsem so rozhodl připojit pouze tři signály. Dva kanály používám k ovládání směru jízdy a jeden používám k ovládání světelné signalizace. Ke čtení těchto signálu jsem se rozhodl využít pinů 19, 20 a 21.

Dále je na řadě světelná signalizace. Jak jsem se zmínil dříve, rozhodl jsem se ke světelné signalizace využít čtveřici LED diod. LED diody k ovládání potřebují pouze jeden signál, druhá nožička je pak připojena na zem. K těmto diodám je nutno připojit vyrovnávací odpor. Pro tento účel jsem zvolil běžné odpory s odporem 220 Ω. Pro tento účel jsem využil piny 62 až 65.

Na konec jsem připojil moduly HC-SR04. Tento modul vyžaduje zapojení k napájení 5V a k zemi. Pro tento účel jsem opět využil pinu na Arduinu produkující 5V. Dále vyžaduje řídící signál trigger. V momentě, kdy je třeba vyčíst hodnotu z tohoto modulu, je třeba po tomto signálu vyslat 10μs dlouhý signál. Následně modul provede měření a výsledek pošle po signálu echo. Výstup signálu echo je v principu stejný, jako signál pro čtení z vysílačky. Z tohoto důvodu jsem se pro tento signál rozhodl využít i stejného principu čtení. V této fázi používám pro signál echo pin 18 a pro signl trigg pin 13. [19]

5.3 Aplikace

Aplikaci jsem se rozhodl psát v Arduino 1.8.9.[23] Jediný modul, který v mé aplikaci programuji je Arduino Mega 2560 Rev3. [11] Tento modul slouží, jako řídící jednotka.

5.3.1 Zpracování vstupů

Jako vstup mému Arduino slouží modul MK610 [18] a modul HC-SR04 [19]. Oba tyto moduly vytvářejí PWM signál, jehož délku měřím. Na začátku programu nastavím přerušení na piny, ze kterých chci číst hodnoty. Je-li z příslušného pinu vyvoláno přerušení a je-li příslušný signál ve stavu HIGH, pak program zaznamená čas v μs. Je-li tento signál ve stavu LOW a zaznamenal-li jsem předtím čas, vypočítám délku signálu z rozdílu současného a předtím zaznamenaného času. Tento získaný čas je poté zpracováván periodicky v proceduře loop(). Procedura pro zpracování vstupu ReadInput() odstraní šum ze vstupu. Poté se zpracování těchto vstupu rozdělí na dvě možnosti.

Hodnota pro z modulu MK610[18] pro ovládání LED diod se dále už nemění a je rovnou přetvořen na signál pro LED diody. K tomuto dochází v proceduře setLED().

Hodnoty pro ovládání motorů vozítka se přepočítávají na rychlosti jednotlivých motorů. Toto vyplývá z mého rozhodnutí, ovládat tento typ podvozku na jedné páčce ovladače. Vypočítal jsem poměr mezi rychlostmi motorů na každé straně vozítka. V principu se vypočítá pomocí Pythagorovy věty absolutní vzdálenost páčky od klidové polohy ovladače. Následně se vypočítá poměr o který každou z motorů zpomalit. Toto je provedeno v proceduře genMotorSpeedsFromReading(). Na konec se v proceduře setMotorValues() nastaví příslušným motorům příslušné hodnoty.

Hodnota z modulu HC-SR04[19] také není dále zpracována až do vykonání procedury setMotorValues(). V této proceduře se motory, v případě že se má vozítko pohybovat směrem, kterým senzor měří, zpomalí nebo zastaví, dle naměřené hodnoty.

5.3.2 Výstupy

Pro motory je výstup tvořen v proceduře setMotorValues(). V této metodě se pouze ze znaménka dříve vypočítané hodnoty pro rychlosti určí směr pohybu každé strany vozítka. V případě, že je to nutné, změní se rychlost dané strany vozítka, dle čtení ze senzorů. Následně se dané dvojici motorů nastaví signál směru a nastaví se příslušná rychlost.

Nastavení světelné signalizace se provádí v proceduře setLED(). Zde se zjistí, které diody mají svítit, a nastaví hodnoty na jejich pinech, tak aby se rozsvítili a zhasly příslušné diody.

5.4 Princip ovládání

Během vývoje prototypu jsem vyzkoušel dva možné principy ovládání. Každý měl své výhody a nevýhody, jak jsem očekával. Myslím, že ani jeden ze způsobů nebyl ideální, nicméně je uvádím oba.

Prvním principem, který jsem vyzkoušel, bylo přiřadit každé dvojici kol rychlost, podle polohy příslušné páčky na ovladači. Tento princip je snadný na implementaci. Každá páčka na dálkovém ovládání má přiřazené dva kanály na přijímači. Každá poloha páčky na ovladači vrací hodnoty mezi 1000 a 2000 (tato hodnota vychází z měření, které jsem prováděl s výše zmíněným přijímačem a ovladačem). Pro toto ovládání motorů je nutné vymezit tedy jeden kanál na každé páčce, který bude reprezentovat rychlost jedné sady motorů. Od této hodnoty je třeba odečíst 1500. Tímto způsobem se získá vzdálenost páčky od střední polohy. Poté je třeba vypočítat absolutní hodnotu výše zmíněné hodnoty. Následně je třeba přepočítat tento výsledek z hodnoty mezi 0 a 500 na 0 až 255. Tento výsledek se předá motorům pomocí funkce analogWrite() [24] na příslušných pinech. Poté je třeba podle znaménka načtené hodnoty nastavit dvojici motorů, tak aby se točila požadovaným směrem. Na výše zmíněném ovládání má tento způsob tři nevýhody.

První z nevýhod je, že na ovládání se obě páčky chovají na ose zdola nahoru jinak. Levá páčka se vrací na střed, kdežto pravá páčka setrvá v poloze, kde ji ovládající člověk zanechá.

Druhá nevýhoda je spojena s ovládáním světelné signalizace. Vzhledem k tomu, že jsem se rozhodl nechat ovládání světelné signalizace oddělené od zvoleného směru jízdy, bylo nutné pro něj vymezit jeden kanál. Vzhledem k tomu, jaké kanály jsou k dispozici, bylo vhodné využít pouze osu zleva doprava na jedné z páček. To by znamenalo, že jedna páčka by ovládala pouze rychlost jedné strany vozidla a druhá by ovládala rychlost jedné strany vozidla spolu se světelnou signalizací.

Poslední nevýhodou je zvýšená obtížnost jízdy rovně, obzvlášť v případě, že by chtěl ovládající člověk měnit rychlosti. Toto je navíc ztíženo ovládání osvětlením a rozličným chováním páček v příslušných osách.

Druhý způsob ovládání, který jsem vyzkoušel je přepočítání obou os jedné páčky na rychlosti jednotlivých dvojic motorů vozítka. Tento způsob je náročnější na výpočty. Prvně se pomocí Pythágorovy věty vypočítá vzdálenost páčky od polohy (0, 0). Toto je nejvyšší rychlost jedné ze stran vozítka.

Dále se podle sektoru ve kterém se páčka nachází určí, který a jak moc změní rychlost druhý z pásu. Pohybuje-li se páčka po přímé ose, je rychlost obou stran stejná. Výsledkem tohoto manévru je jízda vpřed, nebo vzad. Pohybuje-li se páčka na ose zleva doprava je rychlost motorů přesně opačná. Výsledkem tohoto manévru je otáčení na místě různou rychlostí. Je-li pak páčka v rohu, pak zpomalená strana vozidla stojí. Výsledkem je zatáčení vozidla příslušným směrem. Poměr mezi těmito hodnotami se stupňuje, podle blízkosti k bodu s jasným poměrem.

Jak jsem zmínil, tak nevýhodou této metody je větší režie pro výpočet rychlosti jednotlivých stran vozítka.

Druhou nevýhodou je nutnost změny směru otáčení na ose 0, ve směru zezadu dopředu. Po troše testování jsem to shledal docela intuitivní, takže tato nevýhoda není tak velká.

Velkou výhodou této metody je snadnější ovládání na straně uživatele. Pro ovládání pohybu vozítka potřebuje uživatel pouze jednu páčku. Druhá páčka je využita k ovládání světelné signalizace.

Pro svoji implementaci jsem se rozhodl použít k ovládání vozítka levou páčku. K tomuto jsem se rozhodl, protože tato páčka se vrací do polohy (0, 0).

5.5 Úprava rychlosti dle čtení senzoru

Nyní mám načtenou vzdálenost ze senzorů. Nyní je třeba upravit rychlost motorů, dle rychlosti ze senzorů. Původně jsem zkoušel pouze zastavit, přiblíží-li se vozítko moc blízko k objektu. Tedy v momentě. kdy motory točí určitým směrem a na senzoru je moc nízká hodnota, rychlost motorů je nastavena na nulu. Tento způsob byl špatný, protože vozítko buď ve velké vzdálenosti od překážek, nebo nezastavilo v čas, což vedlo k nepřesným čtením za senzoru.

Na konec jsem zvolil, že vozítko má nastaveny vzdálenosti, ve kterých sníží rychlost vozítka. Pro změnu rychlosti jsem zvolil dělení rychlosti načtené z ovladače. Načte-li vozítko překážku blíže, než 50 placů, vozítko zpomalí na polovinu přijímané hodnoty. Je-li překážka blíže než 25 palců, řídící jednotka vydá pokyn pro jízdu čtvrtiny zadané rychlosti. Na konec ve vzdálenosti 6 palců od překážky, vydá řídící jednotka signál k zastavení. Jede-li vozítko plnou rychlostí, tak zastaví, vzhledem k setrvačnosti asi 2.5 palce od překážky, což mi na konec přišlo, jako přijatelné.

Kapitola 6

Testování

Jednotlivé komponenty bylo potřeba otestovat. Na konec jsem testovat i celý prototyp. Při testování jsem si určil vstupy a očekávané výstupy. V případě testování vstupů a výstupů jednotlivých komponent jsem využíval sériovou linku na desce Arduino ATmega2560 rev. 3[10] a sériový monitor dostupný v Arduino IDE[23].

V případě senzorů jsem z Arduina posílal načtené hodnoty na sériovou linku a tam si přečetl, zda hodnoty odpovídají očekávaným hodnotám. Během tohoto jsem měnil vstup do senzoru.

V případě motorů a světelné signalizace jsem pak nastavil opět pomocí sériové linky výstupní hodnoty. Poté jsem pozoroval, jak se periferie chová.

Dále jsem testoval propojení vstupu a výstupu. Zde jsem již Arduino IDE[23] nepoužíval. Místo něj jsem použil již otestované periferie.

6.1 Jednotlivé testy

6.1.1 Ovládání motorů

Ovládání jednoho motoru, jak bylo zmíněno dříve, vyžaduje tři piny, z nichž jeden musí být PWM signál. PWM signál na Arduinu odpovídá hodnotě mezi 0-255.[24] Změní-li se hodnota na vstupu jednotky, očekávám, že rychlost otáčení motoru se změní. Čím větší hodnota na výstupu bude do jednotky posíláno, tím rychleji se bude motor točit. Dále jsou zde ještě dva vstupy. V tomto případě je nutné nastavit jeden na jedničku a druhou na nulu. V testu jsem se to rozhodl měnit tuto hodnotu se znamínkem přijaté hodnoty.

Poté je třeba připojit ovládání motoru a řídící jednotku k baterii. Nejsou-li obě jednotky připojené ke stejné baterii, nepůjde komunikace mezi těmito jednotkami.

Pošlu-li pak do motoru hodnotu 0, očekávám, že motor bude stát, potažmo se zastaví. Pošlu-li na vstup hodnotu mezi 1 a 255, očekávám že motor se bude otáčet jedním směrem různými rychlostmi. Pošlu-li na vstup hodnoty mezi -1 a -255, očekávám že motor se bude otáčet druhým směrem různými rychlostmi.

V tomto testu jsem zjistil, že pošlu-li do motoru moc malé hodnoty (absolutní hodnota PWM bude blízko nule), motor se neroztočí. Myslím, že toto je spojeno s nedostatečným příkonem motoru. Při ostatních hodnotách se chová motor, dle očekávání.

6.1.2 Ovládání a přijímač

K tomuto testu jsem použil pouze ovladač, přijímač a řídící jednotku. Nejprve bylo třeba sesynchronizovat ovladač s ovládáním. V tomto testu není třeba baterie, protože přijímač nemá takový odběr, jako motory. V tomto případě napájím řídící jednotku z počítače. Přijímač pak napájím přímo z řídící jednotky.

Po zapojení načítám hodnoty z přijímače a posílám výsledek čtení na sériovou linku. Očekávané hodnoty čtení jsem popsal výše.

V tomto testu používám dříve popsaný způsob, kde pomocí přerušení měřím délku impulzu na vstupu.

V tomto testu jsem načetl hodnoty od 1000 do 2000, pro každý kanál. Na vysílačce je celkem pět kanálů. Čtyři z těchto kanálů jsou určeny pro čtení ze dvou páček. Každý z těchto kanálů je přiřazen jednomu směru jedné páčky. Pátý čtený kanál je určený pro přijímání stavu přepínače nahoře na pravé straně ovladače. Tento přepínač má pouze dva stavy, proto na tomto kanálu lze načíst pouze hodnoty 1000 a 2000.

6.1.3 Senzor

Jako senzor používám modul HC-SR04[19]. Ke správnému použití tohoto modulu, dle dokumentace byly potřeba dva datové piny. Tento modul, je stejně jako přijímač vysílače, možné napájet přímo z řídící jednotky. Tento modul potřebuje přijmout na pinu trigg 10μs. Po přijmutí tohoto signálu provede jednotka měření a následně na pinu echo pošle naměřenou hodnotu.

Program pro tento test posílá na pin trigg signál pro měření a následně čeká na naměřenou hodnotu. Tuto hodnotu pak pošle přes sériovou linku do počítače.

Ke zbytku tohoto testu je ještě potřeba metr a pevná překážka. Při tomto testu opět vystačí napájení z počítače přes kabel.

Test probíhá tak, že roztáhnu metr tak, že senzor je na nule a následně měním polohu překážky. Očekávané chování je, že čím dál bude překážka, tím vyšší hodnotu senzor načte.

V průběhu tohoto testu jsem zjistil, že senzor vrací hodnotu v setinách palce. Jinak se senzor choval dle očekávání. Zjistil jsem také, že přiblíží-li se překážka blíž než dva a půl palce, senzor začne provádět neplatná měření. Během mého testu se mi bohužel nepodařilo nalézt limitní vzdálenost, do které senzor měří.

6.1.4 Propojení motorů a přijímače s vysílačkou

Jak jsem dříve zmínil požíval jsem během vývoje prototypu dva typy ovládání. Každý z těchto způsobů potřeboval otestovat. Oba tyto způsoby používají stejné zapojení. Nejprve je nutné připojit k řídící jednotce řízení motorů. K ovládání motorů je potřeba, aby byl modul napájený z baterie. V těchto testech jsem již využíval celý podvozek. K řídící jednotce jsem tedy připojil všechny čtyři motory a jejich ovládací moduly.

Poté bylo potřeba k řídící desce přijímač vysílačky. Pro každý typ ovládání používám jinou kombinaci kanálů z přijímače. Vzhledem k omezenému množství počtu pinů, které jsou schopny asynchronně zpracovat tyto kanály zapojoval jsem pouze kanály nutné pro první, či druhý způsob.

Pro první způsob ovládání posílám do řídící jednotky čtení z kanálů z páček, pro pohyb zepředu dozadu. Řídící jednotku nastavím tak, aby přepočítanou hodnotu čtení z přijímače přímo předala motorům. Následně musí řídící jednotka podle znaménka této hodnoty nastavit směr otáčení motorů.

Pro druhý způsob posílám do řídící jednotky dva kanály ze stejné páčky. Řídící jednotka přepočítá tato čtení na rychlosti motorů na každé straně vozítka. Tyto rychlosti předá řízení motorů. Následně podle těchto rychlostí nastaví směr otáčení motorů.

U těchto testů očekávám, že vozítko bude jezdit, dle pokynů z ovladače. V prvním případě tedy pokud změním polohu páčky v přímém směru, změní se rychlost příslušné strany vozítka příslušným způsobem. V druhém případě očekávám od vozítka výše zmíněné chování.

V průběhu testování jsem zjistil, že v prvním případě je obtížnější jet přímo rovně. Toto ztěžuje různé chování páček na vysílačce. V druhém případě se vozítko chová přesně dle očekávání.

6.1.5 Celý prototyp

Pro testování celého prototypu bylo nutné sestavit celý prototyp, dle výše zmíněného návrhu. Následně je třeba zvolit do řídícího modulu program pro celé vozítko.

Očekávaným chováním je, že vozítko bude jezdit dle pokynů z vysílačky. Světelná signalizace se také bude chovat dle pokynů z vysílačky. Tedy že vozítko se bude pohybovat směrem, kterým ukáže levá páčka. Světelná signalizace se rozsvítí na straně, kam ukazuje pravá páčka. Na zbytek příkazů z vysílačky vozítko nebude reagovat.

Vozítko dále nepojede směrem, kde zaregistruje moc blízkou překážku. Také, je-li dán příkaz z vysílačky, aby vozítko jelo do příliš blízké překážky, směrem, který zabírá senzor, bude vozítko tento příkaz ignorovat.

U tohoto testu jsem zjistil, že aby vozítko do překážky nevjelo je třeba začít zpomalovat ve větší vzdálenosti od překážky. Dále jsem zjistil, že vrazí-li vozítko senzorem do překážky, tak se stává, že senzor přestane načítat vzdálenost. Stane-li se toto, je třeba vozítko restartovat. Také jsem zjistil, že kdykoliv je vozítko zapnuté, je dobré mít zapnutou vysílačku. Je-li vysílačka zapnutá, tak vozítku posílá hodnoty neustále. Je-li vysílačka vypnutá, tak přijímač chytá náhodné hodnoty. Vypne-li se vysílačka během provozu vozítka, přijímač chytí z vysílačky vypínací sekvenci, tato sekvence pak způsobí, že vozítko vyrazí náhodným směrem.

Kapitola 7

Nalezené problémy

Během stavby a testování prototypu jsem narazil na několik problémů, jejichž řešení nespadá do předchozích kapitol. V této kapitole bych vám rád popsal některé z těchto řešení.

7.1 Upevnění řídících desek k podvozku

Při upevnění Arduina[10] a modulů L298N[15] jsem narazil na to, že podvozek[20] na sobě nemá upevnění pro tyto moduly. Nemá ani kombinaci děr, do kterých by šlo přímo tyto moduly upevnit. Vzhledem k tomu, že je prototyp pohyblivý je nutné tyto moduly upevnit.

K řešení tohoto problému jsem navrhl své vlastní destičky s upevněním. K tomuto návrhu jsem použil program OpenSCAD. Jedná se o program který pomocí jednoduchého jazyka vytváří 3D objekt.[25] Z tohoto programu pak potřebuji dostat tento objekt ve formátu STL, aby šel dále zpracovat.

K dalšímu zpracování tohoto modelu používám program Slic3r. Tento program přijme model ve formátu STL a následně vygeneruje, dle přednastavených parametrů soubor s instrukcemi pro 3D tiskárnu. Výstupem tohoto programu je G-code, což je kód s jednotlivými pohyby, které má 3D tiskárna provézt. [26]

Následně jsem k tisku použil 3D tiskárnu od firmy Průša, konkrétně Průša MK3[27]. Tato tiskárna s prokázala, jako velmi dobrá volba. Tuto tiskárnu jsem zvolil, protože pro mě byla snadno dostupná.

Tisknuté modely jsou tisknuty z materiálu ABS. Tento materiál se taví při teplotě 235 stupňů Celsia. Tiskárna provede nezbytnou práci, dle vygenerovaných příkazů.

Pro přidělání desek jsem navrhl dva typy desek. První typ desek jsou L298N[15]. Tyto desky jsem se rozhodl přidělat na strany podvozku. Toto rozhodnutí jsem učinil kvůli délce a dostupnosti kabelů. Na podvozku na každé straně je trojice děr uskupených ve tvaru trojúhelníku. Na desce jsou čtyři díry.

Navržená deska je tlustá 0,5 centimetru. Obsahuje dva typy děr. Každý typ děr z každé strany kopíruje odpovídající tvar. Poté je třeba deskou protáhnout šrouby a přišroubovat ji k podvozku. Následně se na čtveřici šroubů, trčící z podvozku nasadí modul L298N. Následně se na tyto šrouby našroubují matice a tento modul je upevněn.

Pro připevnění Arduina jsem použil stejný princip, akorát deska je větší a tvar na podvozku je jiný. Oba tyto designy jsou na přiloženém DVD.

7.2 Upevnění světelné signalizace, přijímacího modulu a baterie

K upevnění těchto součástek bylo nejprve nutné určit, kde na podvozku je chci mít upevněné Pro všechny tyto součástky jsem použil podobnou metodu. Jak se při testování ukázalo, takto přidělané díly drží dostatečně pevně.

Pro světelnou signalizaci jsem urči, že chci mít v každém rohu podvozku jedno světlo. K tomuto účelů jsem použil dvojice otvorů v každém rohu vozítka. Prvně bylo třeba vytvořit drobný modul z diody a kabel, který umožňuje připojení diody k řídící jednotce. Poté jsem tento modul připojil k podvozku pomocí tenké plastové pásky připevnil k podvozku.

K připevnění přijímače vysílačky jsem našel místo na podvozku blízko řídící jednotky. Zde jsem protáhl dírou v podvozku plastovou pásku. Poté jsem přijímač připevnil pomocí této pásky k podvozku. Tento modul jsem původně zkoušel připojit pomocí suchého zipu. Suchý zip se nicméně ukázal, jako nedostatečné řešení, protože modul při vyšších rychlostech padal.

Baterie jsem chtěl přidělat tak, aby šli snadno vyměnit. Rozhodl jsem se je proto připevnit blízko k okraji vozítka. V této části je velké množství malých děr. Těmito děrami jsem protáhl plastovou pásky. Chtěl jsem připevnit obě baterie stejnou páskou, protože pak bude snazší je vyměnit. Na konec jsem použil tři pásky, které jsem propojil a použil, jako jednu. Toto řešení se prokázalo, jako dostatečné.

7.3 Připevnění senzoru

Senzor bylo potřeba připevnit k vozítku tak, aby snímalo překážky před vozítkem. Proto jsem si určil jednu stranu, jako předek vozítka. K připevnění senzoru jsem použil plastový držák prodávaný se senzorem. Tento držák jsem se rozhodl připojit na spodek horní části vozítka. Takto umístěný senzor snímá vše dostatečným způsobem. Držák jsem instaloval pomocí dvou šroubů a dvou matek do děr na přední straně podvozku.

7.4 Poničení podvozku

V průběhu testování došlo k několika potížím s podvozkem.

Jednou při utahování šroubu při přidělávání modulů jsem zabral moc silně a poničil horní díl podvozku. Tento díl praskl a část z něho upadla. Tento díl jsem pak slepil lepidlem.

Druhým takovýmto problémem byl pád podvozku na zem z výšky jednoho metru. Při této nehodě došlo ke zničení úchytů motorů. V první řadě bylo nutné prohodit vrchní a spodní část podvozku, protože motory jsou uchyceny ke spodní části a část podvozku, na kterou jsou přichyceny motory, byla zničena.

Dále bylo nutné vyměnit plasty držící motory na podvozku. Bohužel se mi nepodařilo najít rychle náhradu, takže jsem navrhl vlastní. Prvně jsem si změřil původní úchyty. Poté jsem vytvořil nový 3D model opět v programu OpenSCAD[25]. Následně jsem z tohoto modelu pomocí programu Slic3r[26] vygeneroval gcode pro 3D tiskárnu. Dále jsem vytiskl tyto modely opět z materiálu ABS na 3D tiskárně Průša MK3. Na konec jsem tyto úchyty vyměnil. Výsledné designy jsou dostupné na přiloženém DVD.

Závěr

Hlavním cílem této práce bylo navrhnout a sestavit prototyp vozítka na dálkové ovládání založeném na platformě Arduino. V průběhu návrhu tohoto prototypu jsem prozkoumal dostupná řešení podobných projektů. Z těchto poznatků jsem rozložil problém vozítka na jednotlivé podproblémy. Navrhl jsem řešení. Dále jsem vybral kombinaci součástek, která mi umožní požadovanou funkcionalitu. Toto řešení jsem pak sestavil a otestoval.

Při testech jsem si ověřil vlastnosti podvozku. Práce se zabývala i návrhem testů a zhodnotila, jak se prototyp choval, vzhledem k předpokládanému chování.

Stanovených cílů se mi podařilo dosáhnout. Prototyp byl sestaven, naprogramován a otestován.

Literatura

[1] CPARKTX: Arduino Powered Autonomous Vehicle. [online], 2018 [cit. 2019-02-11]. Dostupné z: https://www.instructables.com/id/Arduino-Powered-Autonomous-Vehicle/

[2] Ardumotive_com: Arduino Bluetooth RC Car (Android Controlled). [online], 2018 [cit. 2019-02-11]. Dostupné z: https://www.instructables.com/id/Arduino-Bluetooth-RC-Car-Android-Controlled/

[3] Baranov, A.: Smartphone Controlled Arduino 4WD Robot Car. [online], 2017 [cit. 2019-02-11]. Dostupné z: https://www.hackster.io/andriy-baranov/smartphone-controlled-arduino-4wd-robot-car-14d239

[4] Atmel: ATmega328P Datasheet. [online], 2015 [cit. 2019-02-11]. Dostupné z: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf

[5] Arduino: Arduino Uno Rev3. [online], 2019 [cit. 2019-02-11]. Dostupné z: https://store.arduino.cc/arduino-uno-rev3

[6] Atmel: ATmega3209/4809 – 48-pin Datasheet. [online], 2018 [cit. 2019-02-13]. Dostupné z: http://ww1.microchip.com/downloads/en/DeviceDoc/40002016A.pdf

[7] Arduino: Arduino Uno WiFi REV2. [online], 2019 [cit. 2019-02-11]. Dostupné z: https://store.arduino.cc/arduino-uno-wifi-rev2

[8] Atmel: SAM3X / SAM3A Series Datasheet. [online], 2015 [cit. 2019-02-13]. Dostupné z: http://ww1.microchip.com/downloads/en/DeviceDoc/40002016A.pdf

[9] Arduino: Arduino Due. *www.arduino.cc*, [online], 2019 [cit. 2019-02-11]. Dostupné z: https://store.arduino.cc/arduino-due

[10] Atmel: Atmel ATmega640/V-1280/V-1281/V-2560/V-2561/V Datasheet. [online], 2014 [cit. 2019-02-13]. Dostupné z: https://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf

[11] Arduino: Arduino Mega 2560 Rev3. [online], 2019 [cit. 2019-02-11]. Dostupné z: https://store.arduino.cc/arduino-mega-2560-rev3

[12] Paštyka, J.: DC motory řízené pomocí H můstku. *České vysoké učení technické v Praze, Fakulta elektrotechnická*, [online], 2017 [cit. 2019-02-11]. Dostupné z: https://embedded.fel.cvut.cz/sites/default/files/kurzy/lpe/h-bridge/H_Bridge.pdf

[13] Arduino: Adafruit Motor / Steper / Servo Shield - V2.3. [online], 2019 [cit. 2019-02-11]. Dostupné z: https://store.arduino.cc/adafruit-motor-stepper-servo-shield-v2-3

[14] STMicroelectronics: L298. [online], 2000 [cit. 2019-02-11]. Dostupné z: https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf

[15] Tech, H.: L298N Dual H Bridge DC Motor Driver Module. [online], 2018 [cit. 2019-04-23]. Dostupné z: https://handsontec.com/index.php/product/l298n-dual-h-bridge-dc-motor-driver-module/

[16] SPEKTRUM: DX5e. [online], 2011 [cit. 2019-02-11]. Dostupné z: https://www.spektrumrc.com/Products/Default.aspx?ProdId=SPM5520#prod_manuals

[17] SPEKTRUM: AR500 DSM2 5-Channel Sport Receiver. [online], 2008 [cit. 2019-02-11]. Dostupné z: https://www.spektrumrc.com/Products/Default.aspx?ProdId=SPMAR500

[18] Aliexpress: MK610 Receiver. [online], 2019 [cit. 2019-02-11]. Dostupné z: https://www.aliexpress.com/item/1pcs-MK610-Receiver-2-4GHz-Replace-AR6100-Receiver-for-Dx5e-Dx6i-Dx7-Dx8/32970328829.html?+spm=a2g0s.9042311.0.0.13554c4dYeLlm

[19] ELECFreaks: Ultrasonic Ranging Module HC - SR04. [online], [cit. 2019-02-11]. Dostupné z: https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf

[20] Aliexpress: Smart Car Kit 4WD Smart Robot Car Chassis Kits with Speed Encoder and Battery Box for arduino Diy Kit. [online], [cit. 2019-02-11]. Dostupné z: https://www.aliexpress.com/item/Smart-Car-Kit-4WD-Smart-Robot-Car-Chassis-Kits-with-Speed-Encoder-and-Battery-Box-for/32952005054.html

[21] Philips: Rechargeables Baterie 9VB1A17/10. [online], [cit. 2019-02-11]. Dostupné z: https://www.philips.cz/c-p/9VB1A17_10/rechargeables-baterie/specifications

[22] Arduino: attachInterrupt(). [online], 2019 [cit. 2019-04-19]. Dostupné z: https://www.arduino.cc/reference/en/language/functions/external-interrupts/attachinterrupt/

[23] Arduino: Download the Arduino IDE. [online], 2019 [cit. 2019-04-19]. Dostupné z: https://www.arduino.cc/reference/en/Main/Software

[24] Arduino: analogWrite. [online], 2019 [cit. 2019-05-10]. Dostupné z: https://www.arduino.cc/reference/en/language/functions/analog-io/analogwrite/

[25] OpenSCAD: OpenSCAD. [online], 2018 [cit. 2019-05-13]. Dostupné z: https://www.openscad.org

[26] Ranellucci, A.: Slic3r. [online], 2019 [cit. 2019-05-13]. Dostupné z: https://slic3r.org/

[27] Průša, J.: 3D TISKÁRNA ORIGINAL PRUSA I3 MK3S. [online], 2017 [cit. 2019-05-13]. Dostupné z: https://www.prusa3d.cz/original-prusa-i3-mk3/

Příloha **A**

Seznam použitých zkratek

HW Hardware

PWM Pulse Width modulation

ABS Akrylonitrilbutadienstyren

IDE Integrated Development Enviroment

DVD Digital Versatile Disk

Příloha B

Obsah přiloženého DVD

- `readme.txt` ..... stručný popis obsahu CD
- `src`
  - `arduino`
    - `final.ino` ..... Celkový kód splňující funkcionalitu
    - `tests`
      - `motorTest.ino` ..... Program pro testování motoru
      - `recieverTest.ino` Program pro testování Ovládání a přijímače
      - `sensorTest.ino` ..... Program pro testování senzoru
      - `oldRemoteTest.ino` .. Program pro otestování staršího způsobu ovládání
      - `newRemoteTest.ino` ... Program pro otestování nového způsobu ovládání
  - `3D modely`
    - `STL`
      - `ArduinoMegaHolder.stl` ..... 3D model držáku na Arduino ATmega2560 Rev3
      - `MotorControlerHolder.stl` ..... 3D model držáku na L298N
      - `MotorHolder.stl` ..... 3D model držáků na motory
    - `gcode`
      - `ArduinoMegaHolder.gcode` ... gcode pro Prusa MK3 držáku na Arduino ATmega2560 Rev3
      - `MotorControlerHolder.gcode` gcode pro Prusa MK3 držáku na L298N
      - `MotorHolder.gcode` ... gcode pro Prusa MK3 držáků na motory
  - `thesis` ..... zdrojová forma práce ve formátu LaTeX
- `Zemanm30BAP.tex` ..... text práce
  - `Zemanm30BAP.pdf` ..... text práce ve formátu PDF