

System pro bezdrátové řízení modulu automobilu pomocí mikropočítače

Martin Heryán

Bakalářská práce
2020



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav informatiky a umělé inteligence

Akademický rok: 2019/2020

1. **ZADÁNÍ BAKALÁŘSKÉ PRÁCE**
(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: Martin Heryán
Osobní číslo: A17119
Studijní program: B3902 Inženýrská informatika
Studijní obor: Softwarové inženýrství
Forma studia: Prezenční
Téma práce: Systém pro bezdrátové řízení modelu automobilu pomocí mikropočítače
Téma práce anglicky: A Wireless Control System for a Car Model Based on a Microcontroller

Zásady pro vypracování

1. Popište možnosti bezdrátového řízení modelů a existující konstrukce podobného typu.
2. Zvolte vhodné komponenty a model automobilu pro řízení pomocí rozhraní Bluetooth.
3. Realizujte propojení zvolených komponent a modelu automobilu.
4. Implementujte programové vybavení pro řídicí mikropočítač umožňující ovládání modelu.
5. Vytvořte aplikaci pro ovládání modelu pomocí mobilního telefonu nebo tabletu.

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: tištěná/elektronická

Seznam doporučené literatury:

2. GRIFFITHS, Dawn a David GRIFFITHS. Head first Android development. Sebastopol: O'Reilly, 2015. Head first series. ISBN 1449362184.
3. LADMAN, Josef. Elektronické konstrukce pro začátečníky. Praha: BEN - technická literatura, 2001. ISBN 80-730-0015-6.
4. MARGOLIS, Michael. cookbook. 2nd ed. Sebastopol, Glí: O'Reilly, 2012, xx, 699 p. ISBN 1449313876.
5. NOVÁK, Petr. Mobilní roboty: pohony, senzory, řízení. Praha: BEN - technická literatura, 2005. ISBN 80-7300-141-1.
5. PINKER, Jiří. Mikroprocesory a mikropočítače. 1. vyd. Praha: BEN - technická literatura, 2004, 159 s. ISBN 80-7300-110-1.

Vedoucí bakalářské práce:

Ing. Jan Dolinay, Ph.D.
Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce: 28. listopadu
2019
Termín odevzdání bakalářské práce: 15. května 2020



doc. Mgr. Milan Adámek, Ph.D.

prof. Mgr. Roman Jašek, Ph.D. děkan

ředitel ústavu

Ve Zlíně dne 9. prosince 2019

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne
10.8.2020

Martin Heryán, v. r.
podpis diplomanta

ABSTRAKT

Cílem této práce je návrh a realizace modelu autíčka, ovládaného pomocí mobilního zařízení s operačním systémem Android prostřednictvím rozhraní Bluetooth.

Základem je starší, běžný model autíčka ovládaný pomocí rádiových vln, které je následně přestavěno pro ovládání pomocí aplikace na mobilním zařízení. Zvoleným mikropočítačem, který se nachází uvnitř autíčka je Arduino UNO, mobilní aplikace je vyvíjena ve webovém vývojovém prostředí MIT App Inventor.

Klíčová slova: Android, Arduino, Mikropočítač

ABSTRACT

The aim of this thesis is the design and implementation of a toy car model, controlled by a mobile device with the Android operating system via Bluetooth. The basis is an older, common model of a toy car controlled by radio waves, which is then rebuilt to be controlled using an application on a mobile device. The chosen microcomputer located inside the car is Arduino UNO; the mobile application is developed in the web development environment MIT App Inventor.

Keywords: Android, Arduino, Microcontroller

Chtěl bych poděkovat panu Ing. Janu Dolinayovi, Ph.D. za veškerý věnovaný čas a za mnoho cenných rad. Dále bych chtěl poděkovat rodině, za podporu po celou dobu studií.

Prohlašuji, že odevzdaná verze bakalářské/diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD	8
I TEORETICKÁ ČÁST	9
1 MOŽNOSTI ŘÍZENÍ MODELŮ	10
1.1 RADIOVĚ OVLÁDANÉ MODELY	10
1.2 OVLÁDÁNÍ BLUETOOTH	10
1.3 ROZDÍLY MEZI RÁDIO A BLUETOOTH OVLÁDÁNÍM	10
2 MIKROKONTROLER	12
2.1 ARCHITEKTURY	12
3.2 SHIELDY	14
II PRAKTICKÁ ČÁST	21
7 HARDWARE	22
7.1 MODEL	22
7.2 ELEKTROMOTORY	24
9 KÓD ARDUINA	32
10 APLIKACE PRO ANDROID	37
10.1 MIT APP INVENTOR	37
ZÁVĚR	44
SEZNAM POUŽITÉ LITERATURY	45
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	47
SEZNAM OBRÁZKŮ	48
SEZNAM TABULEK	49

ÚVOD

Autíčka na dálkové ovládání jsou dnes běžné a snadno dostupné, a protože se různé druhy programování vyučují už na základní škole, dají se RC autíčka využít k zábavnému a zajímavému způsobu výuky, a to nejen embedded systémů jako jsou mikropočítače, ale i mobilních aplikací, které lze využít k řízení například RC autíček, které mohou ale nemusejí posílat data z nejrůznějších senzorů zpět do mobilního zařízení.

Existuje několik druhů RC autíček, které lze zapojit do výuky. Také je několik způsobů, jak naprogramovat mobilní aplikaci pro jejich řízení. Tato práce je zaměřena na jeden konkrétní způsob, a to přestavení levné hračky na ovládání pomocí mobilního telefonu přes rozhraní Bluetooth.

Výsledkem praktické části bude model autíčka, který má původní tělo a elektromotor, nově vložený a zapojený mikropočítač pro ovládání a mobilní aplikace pro operační systém Android.

V teoretické části je popsán mikropočítač Arduino, rozhraní Bluetooth, další existující konstrukce dálkových ovládání autíček a možnosti tvorby mobilní aplikace.

Práci jsem prováděl ve stylu amatérského nadšence, který o dálkově ovládaných autíčkách moc neví, a proto jsem se rozhodl pro co nejjednodušší provedení, zanechání co největšího množství původních součástek a vytvoření mobilní aplikace v APP inventoru, který značně ulehčuje její programování, především v oblasti konektivity Bluetooth, kterou jsem osobně našel velmi obtížnou v jiných vývojových studiích jako například MS Visual Studio či Android Studio.

I. TEORETICKÁ ČÁST

1 MOŽNOSTI ŘÍZENÍ MODELŮ

V dnešní době existuje několik možností, jak dálkově ovládat modely autíček, avšak některé jsou považovány za zastaralé a již se nepoužívají jako například kabel mezi ovladačem a autíčkem. V současné době je nejvíce rozšířeným způsobem ovládání pomocí radiových vln, ale do trendu se pomalu dostává i ovládání přes Bluetooth pomocí mobilního telefonu [1].

1.1 Radiově ovládané modely

Aby autíčko mohlo být ovládané rádiovými vlnami (v dnešní době nejčastěji o frekvenci 2,4GHz), je zapotřebí vysílače, který se nachází v ovladači a přijímače nacházejícího se v samotném autíčku. Přijímač poté přemění radiový signál na vhodné elektrické signály pro další komponenty modelu. Většina systémů používá Amplitudovou modulaci pro radiový signál a zakódování pozic ovládání pomocí pulzně šířkové modulace. [2]

Z přijímače se informace přenáší většinou do mikropočítače, který následně vyhodnocuje akci, kterou má provést, a dále předává pokyn elektromotorům, které zajišťují pohyb vpřed či vzad nebo zatáčení kol.

1.2 Ovládání Bluetooth

K ovládání modelu pomocí Bluetooth musíme mít mobilní zařízení, které má v sobě zabudovaný Bluetooth vysílač a aplikaci, která obstarává připojení a vysílání příkazů. Takovou aplikaci si můžeme naprogramovat sami nebo obstarat jiným způsobem na internetu. Aplikace může být na jakémkoliv operačním systému, avšak musíme dávat pozor na kompatibilitu s přijímacím zařízením v modelu.

Na druhém konci, tak jako u rádiového ovládání, musí být přijímač Bluetooth signálu, který následně předává informaci mikrokontroleru pro další zpracování a rozeslání příslušných pokynů pro pohyb či zatáčení motorům.

1.3 Rozdíly mezi rádiem a Bluetooth ovládáním

Výhodou rádiového ovládání je větší vzdálenost, na kterou lze model ovládat, překážky mezi vysílačem a přijímačem neruší signál do takové míry jako u Bluetooth připojení, a dále jednoduchost a robustnost vysílače zajišťují, že signál bude vyslán kdykoliv je požadováno, na rozdíl mobilního zařízení, které běží na komplexním operačním systému, který může selhat a aplikace, která může mít problémy. Nevýhodou rádiového ovládání je možnost vzájemného se rušení v případě že dva či více modelů a ovladačů pracují na stejné rádiové

frekvenci. Toto je časté v běžných, levných modelech, které zpravidla pracují na frekvenci 2,4GHz. Další častou nevýhodou rádiového ovladače bývá nutnost používání tužkových baterií, které se při delším nepoužívání modelu vybijí či dokonce poškodí, Bluetooth ovládání mobilním telefonem tento problém eliminuje, avšak zůstává pro napájení modelu, které může být provedeno tužkovými bateriemi nebo akumulátorem.

2 MIKROKONTROLER

Jednočipový mikropočítač nebo taky mikrokontroler je procesor s velmi malými rozměry zapojený v integrovaném obvodu společně s pamětí a vstupně výstupním rozhraním. Přítomnost či absence jednotlivých periférií určuje, zda se jedná o mikropočítač univerzální nebo specializovaný, a to tím způsobem, že univerzální mikrokontroler má více periférií, než je nezbytně nutné a zpravidla je takto prodáván, kdežto specializované mají za účelem úspory financí a často i velikosti zabudovány jen nezbytně nutné prvky a často jsou na míru navrženy pro danou činnost. [3]

2.1 Architektury

Existují dva druhy architektur. První architekturou je Von Neumannova specifikovaná společnou pamětí pro instrukce a data. Problémem takového přístupu je, že v jednom okamžiku je možné přistupovat pouze k instrukcím nebo datům, což výrazně zpomaluje funkčnost.

Druhá architektura je Harvardská, která má operační paměť rozdělenou na dvě části. Oproti Von Neumannově je tedy možné najednou přistupovat k instrukcím i k datům.

Často se však využívá kombinace obou. Přičemž je mikroprocesor sestaven podle Harvardské architektury, ale pro přístup externích zařízení se využívá architektura Von Neumannova. [3]

2.2 Instrukční sady

Mikrokontrolery mohou pracovat na jedné ze dvou instrukčních sad, a to buď RISC – Reduced Instruction Set Computer nebo CISC – Complex Instruction Set Computing.

U procesoru RISC je soubor instrukcí zúžen na pečlivě vybrané jednoduché, často používané instrukce. Tím je usnadněna účinná optimalizace obvodů procesoru s cílem maximálního zrychlení činnosti. Program, sestavený z jednoduchých instrukcí RISC, je samozřejmě delší než program, sestavený z instrukcí CISC - celková doba na jeho provedení je však kratší vzhledem k rychlosti provádění jednoduchých instrukcí. Pro účinnost této metody je však nutné pozměnit architekturu procesoru.

Soubory instrukcí moderních CISC procesorů jsou velmi rozsáhlé a mnohé z nich realizují operace tak složité, že by je bylo možné nahradit celým úsekem programu, sestaveného z

jednoduchých instrukcí. Důsledkem je ale velmi složitý řadič procesoru, řešený vždy jako mikroprogramový automat. Celý procesor je pomalejší. [3]

2.3 Základní prvky mikropočítače

2.3.1 Procesor

Úkolem procesoru je provádění instrukcí. Instrukce jsou nejmenší jednotky, ze kterých je složen program. [3]

2.3.2 Paměť

Paměť je součástí počítače, do které procesor zapisuje či z ní čte data a instrukce. Nejčastěji se používají typy FLASH pro instrukce a RAM pro uložení dat. [3]

2.3.3 Sběrnice

Počítač používá tři typy sběrnic pro komunikaci mezi prvky. Datová sběrnice slouží k předávání dat. Jednotka, připojená na sběrnici, může být zdrojem dat (pak se z ní čte), příjemcem dat (pak se do ní zapisuje), nebo střídavě obojím. Adresová sběrnice je nutná pro adresování paměti a pro rozlišování mezi jednotkami, připojenými na datovou sběrnici. Šířka adresové sběrnice určuje maximální počet adres. Čtení, zápis a další aktivity jednotek jsou řízeny signály řídicí sběrnice. Většina řídicích signálů je generována procesorem, ale některé mohou být generovány i ostatními jednotkami, které tak mohou částečně ovlivňovat činnost procesoru. [3]

3 ARDUINO

Arduino je řada malých počítačů, které se často používají pro interakci s fyzickým světem kolem nás. Arduino deska pracuje s údaji od různých snímačů a senzorů (snímač světla, teploty) a na základě naprogramovaných rozhodnutí ovládá různé periferie (LED, elektromotor). [4]

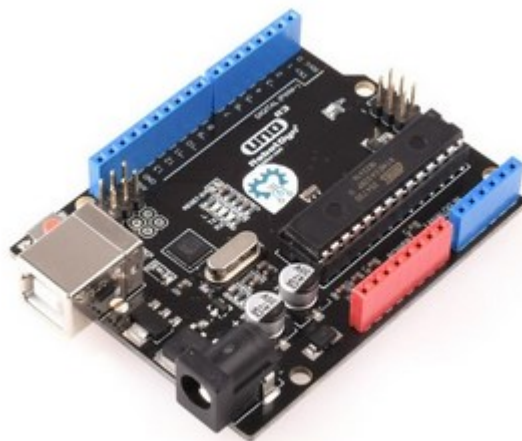
3.1 Arduino Uno

Deska s procesorem ATmega328P od firmy Atmel o frekvenci 16MHz, s pamětí Flash velikou 32 kB, SRAM velikou 2 kB a EEPROM s 1 kB.

Dále se na desce nachází 14 digitálních pinů, ze kterých šest má možnost použít PWM modulaci (piny 3,5,6,7,10 a 11) a dále dva piny označeny Tx a Rx sloužící k sériové komunikaci. Kromě digitálních, se na desce nacházejí i analogové piny, nachází se jich zde celkem šest.

Uno obsahuje port USB – B a USB převodník, díky kterému je možné jednoduše nahrávat programy z PC, slouží navíc jako jedna z variant napájení, dalšími možnostmi na této desce jsou napájecí piny a souosý konektor.

Na desce se nachází diody L, Tx a Rx. Dioda L nemá žádné zabudované využití a nejčastěji se používá pro otestování desky. Diody Tx a Rx blikají při komunikaci na sériové lince. [5]



Obrázek 1 Arduino Uno

3.2 Shieldy

Shieldy jsou přídatné desky pro Arduino, které se dají nasadit na desku a mají za účel přidat určitou funkcionalitu. Můžeme zapojit i několik shieldů najednou, shield na shield.

Mezi nejpoužívanější patří Wifi shield, Bluetooth shield, LCD shield a Motor shield.

3.2.1 Motor shield

Využívá H-můstek k řízení stejnosměrného motoru. Shield může obsahovat jeden nebo dva H-můstky, dle našich požadavků na počet ovládaných motorů. Rychlost otáčení motorů lze kontrolovat napájecím napětím nebo PWM.

4 BLUETOOTH

Bluetooth je jednotná, bezdrátová technologie pro výměnu informací mezi dvěma zařízeními na krátkou vzdálenost. Pro přenos dat se využívá krátkých rádiových vln s ultra vysokou frekvencí. [6]

4.1 Párování

Párováním dvou zařízení docílíme rychlého a jednoduchého spojení kdykoliv to jedno ze zařízení vyžaduje, a to i například v režimu spánku nebo automaticky po zapnutí zařízení.

Párování probíhá ve třech fázích. V první fázi si zařízení vymění informace o tom, čeho je každé z nich schopné, co očekávají a jaký druh párování bude využit. Ve druhé fázi se vygeneruje krátkodobý klíč, pomocí kterého se zařízení mohou spárovat. V případě, že se jedná o mód bezpečného připojení, použije se namísto něj dlouhodobý klíč. Ve třetí fázi se poté generují klíče pro podpis a identifikaci spolu s dlouhodobým klíčem, pokud nebyl vytvořen ve druhé fázi. [6]

4.1.1 Metody párování

Porovnání čísel – na displejích obou zařízení se objeví stejné šestimístné číslo, sloužící pro potvrzení identifikací. Není zabezpečeno proti útoku man-in-the-middle [6]

Přímé párování – obdobně jako v předchozí metodě, i zde se používá šestimístné číslo s tím rozdílem, že jedno ze zařízení nemá displej, tudíž toto číslo bývá natvrdo nastaveno a napsáno v příbalovém letáku obsaženém v balení zařízení. [6]

Přístupový klíč – v případě použití přístupového klíče se šestimístné číslo objeví na displeji pouze jednoho ze zařízení a musí být předáno druhému [6]

Mimo pásmová (OOB) – metoda používaná zařízením Apple Watch. Namířením kamery Iphonu na displej hodinek, který zobrazuje zašifrovaný kód v podobě teček se přenesou informace o párování.

5 TVORBA MOBILNÍ APLIKACE

Před tím, než začneme aplikaci vyvíjet, musíme si určit co je naším cílem. To znamená určit si platformu případně multiplatformnost, vzhled a funkčnost. Různé druhy vývoje mají své výhody a nevýhody. [7]

5.1 Nativní vývoj

Tvořením aplikace nativně se rozumí vyvíjet danou aplikaci na konkrétní platformu, tedy pouze na iOS, Android apod. Výhodami této metody jsou grafický vzhled, optimalizace, efektivní přístup k jednotlivým prvkům bez potřeby použití rozšiřujících knihoven či pluginů a snadná aktualizace na vyšší verzi operačního systému. Obrovskou nevýhodou poté je, když chceme rozšířit cílovou skupinu uživatelů, protože pro podporu více platform musíme v podstatě vyvíjet a následně udržovat dvě „odlišné“ aplikace, čímž se nám zásadně zvedají nároky na čas a výdaje. [7]

5.2 Webový vývoj

Obdobným způsobem, kterým se tvoří běžné webové aplikace je možné tvořit i aplikace na mobilní zařízení. Při vývoji se používají standardní webové technologie jako jsou HTML a Javascript. Tento druh aplikací využívá internetový prohlížeč pro svou funkci a je tedy možné je používat i na stolním počítači. Jelikož se k zobrazení využívá prohlížeč, operační systém, na kterém aplikace běží není zásadní problém. Avšak tyto aplikace nemají přístup k většině periférií zařízení, což nás může limitovat. [7]

5.3 Hybridní vývoj

Hybridní vývoj využívá kombinaci nativního a webového vývoje. Aplikace používá nativní kód jako základ, který je následně rozšířen o webové technologie. Jako u aplikací vyvíjených webovou metodou, jsou hybridní aplikace zobrazovány pomocí internetového prohlížeče a také používají HTML, Javascript a CSS. Na rozdíl od webové vyvíjených, lze tyto aplikace instalovat do zařízení jako u nativních. Hybridní vývoj umožňuje používání pluginů, a tudíž i používání různých hardwarových periférií zařízení.

5.4 Tvorba aplikací pro Android

Android jakožto nejrozšířenější Open Source platforma má několik způsobů, jak můžeme vyvíjet aplikace pro komunikaci s Arduinem. Mezi nejpoužívanější patří Android studio designované, jak název napovídá, přímo pro práci s Androidy, avšak velmi rozšířené MS Visual Studio také poskytuje tuto možnost. Další méně známou metodou, jak vyvíjet aplikace pro mobilní zařízení je MIT App Inventor.

5.4.1 Android studio

Vývojové prostředí optimalizováno pro vývoj aplikací na operační systém Android s integrovaným emulátorem, který nabízí široké spektrum zařízení, a možností připojení fyzického zařízení, přičemž je zde možnost vytvořit spustitelný soubor a vše potřebné k nainstalování a běh aplikace. Android studio nabízí šablony jednoduchých aplikací, na kterých lze dále stavět, společně s možností začít nad čistým projektem. Součástí Android studia jsou GUI nástroje pro zjednodušení vývoje aplikace. Android studio je přizpůsobeno pro vývoj v jazyce JAVA nebo Kotlin. [8]

5.4.2 MIT App Inventor

Intuitivní, vizuální programovací prostředí, umožňující snadně vyvíjet plně funkční mobilní aplikace pro chytré telefony a tablety. [9]

Vývojové prostředí, dostupné kdekoli na internetovém prohlížeči, funguje na stylu drag-and-drop, kde jednotlivé prvky designu umísťujeme přetažením obdobně jako v Android studiu. Rozdílem však je limitace schopností a možností které App Inventor nabízí. Funkcionalitu aplikací „programujeme“ pomocí bloků kódu. Díky tomu programátor nemusí znát žádné syntaxe jazyka, nýbrž programuje pouze logiku. Tímto způsobem lze aplikaci vytvořit podstatně rychleji a s méně chybami, ale opět jsme velmi limitováni v našich možnostech. Po vytvoření aplikace ji lze otestovat na integrovaném emulátoru, připojeném fyzickém zařízení či stáhnout APK soubor a ručně přenést a nainstalovat do zařízení.

5.5 Tvorba programů pro Arduino

Mikrokontrolery lze programovat pomocí širokého spektra jazyků v různých vývojových prostředích [10], přičemž každý z nich má své výhody a nevýhody, proto bychom měli volit jazyk vhodný pro náš úkol a daný hardware. Pro účely tvoření programů pro desky Arduino existuje vývojové prostředí Arduino.

5.5.1 Arduino IDE

Vývojové prostředí navržené pro práci s deskami Arduino, obsahující knihovny a ovladače pro všechny originální produkty Arduino. Pokud chceme používat klony produktů Arduino, kterým neodpovídají originální knihovny, lze do zařízení nahrát své ovladače a nainportovat knihovny. Jazyky, které můžeme při programování používat jsou C a C++. Pro správný chod je nutné mít alespoň funkci Setup, která proběhne jednou na začátku a inicializuje například piny a jiné potřebné části, a dále funkci Loop, ve které se nachází samotné tělo a hlavní funkcionalita, avšak nemusí to být jediná funkce v našem kódu. Funkce Loop je kritická, jelikož mikropočítač nemá operační systém, který by se staral o chod mikropočítače, tudíž funkce Loop běží neustále dokola a v případě, že se neúmyslně dostaneme mimo, není možnost mikropočítač dále kontrolovat a musí být resetován. [11]

6 MODEL AUTÍČKA

Modely dálkově ovládaných autíček existují na trhu v několika variantách, mezi nejčastější patří rekreační předem sestavené modely, ty bývají často levné, masově produkované, bez možnosti jednoduše je modifikovat a často se je nevyplatí či dokonce vůbec nelze opravit v případě poruchy, protože se jednoduše nepředpokládá, že tyto modely vydrží dlouho či zůstanou delší dobu v trendech. Pokročilejší hobby modely jsou zpravidla dražší, výkonnější, robustnější s větším dosahem. Tyto modely obvykle mají k dostání náhradní součástky, a dají se relativně jednoduše opravit. Předpokládá se, že se při poruše opraví daná součástka namísto nahrazení celého modelu, jako to bývá u levnějších variant. Jsou k dostání v již sestavené podobě, ale i v různých podobách jako stavebnice, kde si uživatel může i kombinovat více modelů dle libosti. Posledním způsobem je tzv. DIY, kdy si uživatel nakoupí či vyrobí jednotlivé součástky dle svých požadavků. Tímto způsobem není uživatel nijak limitován a může si sestavit model jakkoliv. Na druhou stranu to vyžaduje jisté znalosti, jelikož nebude mít žádnou podporu, ať už od výrobce nebo z fóra dalších uživatelů a bude si sám muset dávat pozor na kompatibilitu součástek a jejich přetížení. Dále také poškozené součástky, které si jednou vyrobil a chce je vyměnit, bude muset vyrobit znovu.

II. PRAKTICKÁ ČÁST

7 HARDWARE

Před započítím si musíme zvolit model, se kterým budeme následně pracovat. Protože je naším cílem konvertovat radiově ovládaný model na ovládání pomocí Bluetooth, je v našem zájmu vybrat si model o dostačující velikosti, aby se nám do něj vměstnaly všechny potřebné komponenty a kabely. Dále si musíme určit, jestli a jaké původní komponenty si ponecháme. Nakonec si musíme vybrat a pořídit mikrokontroler se kterým budeme pracovat a k němu potřebný Bluetooth modul, pokud není integrován, a na konec motor shield na ovládání elektromotoru pro pohyb.

7.1 Model

Pro naše účely byl zvolen velmi starý model Ford Explorer o celkové velikosti šířka 32 cm, délka 47 cm a výška 29 cm, jak lze vidět na obrázcích 2 a 3. Avšak užitečný prostor, ve kterém se nachází všechny komponenty pro ovládání je podstatně menší, jak lze vidět na obrázku 4. I přes to, je tento model dostačující.

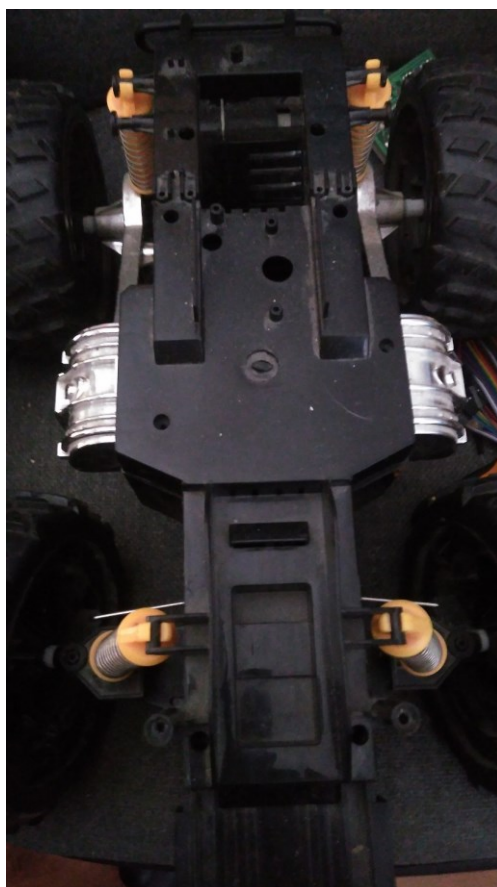
Jedná se o levnou hračku starou cca 15 let. Dnes ji již nelze zakoupit a velmi obtížně se dají dohledat obdobné, ale ne totožné modely. Jak bylo zmíněno v šesté kapitole teoretické části, nelze k tomuto modelu zakoupit náhradní součástky. Původní řídicí jednotka, kterou lze vidět na obrázku 5, byla sestavena výrobcem na míru pro tento model.



Obrázek 2 modele autíčka ze strany



Obrázek 3 model zepředu



Obrázek 4 užitečný prostor



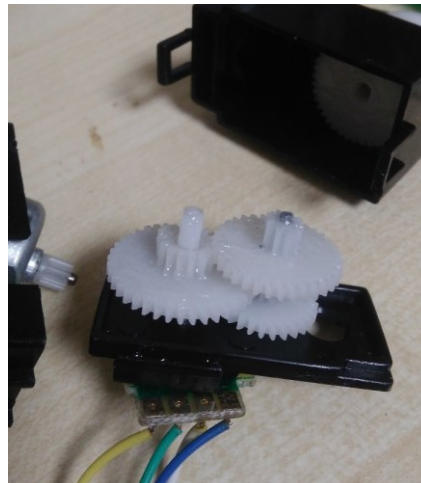
Obrázek 5 původní řídicí jednotka

7.2 Elektromotory

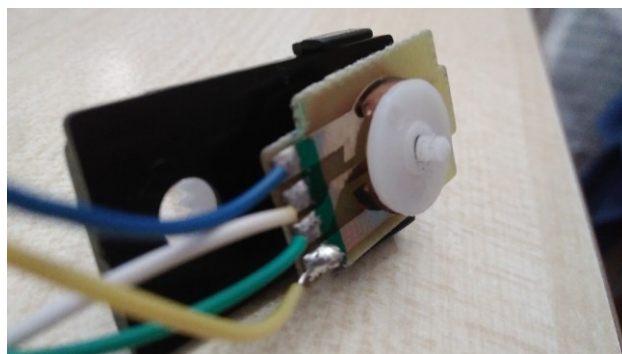
Vzhledem k jednoduchosti modelu, jediné užitečné původní součástky byly elektromotor na pohon zadních kol a specializovaný elektromotor se senzory na ovládání předních kol. Tato součástka, kterou lze vidět na obrázcích 6,7 a 8, se skládá z obyčejného elektromotoru a velmi neobvyklého senzoru, který má čtyři dráty. Princip fungování, který bylo velmi těžké odhalit, jelikož je tento ovládací systém vzácný a dohledat jej je prakticky nemožné. Otáčením elektromotoru se přes ozubená kola pohybuje bílé kolo na obrázku 8 v limitovaném rozmezí a pomocí multimetru lze naměřit odpor mezi určitými dráty. I přes rozluštění funkcionality drátů se však nepovedlo žádným způsobem připojit je k desce Arduino a zajistit jakoukoliv schopnost ovládat tak řízení modelu. Ovládání předních kol je tudíž zajištěno pouze pohybem samotného elektromotoru, které je zapojené do motor shieldu, bez zpětné vazby skrze tyto dráty. Alternativním řešením by bylo vyměnit elektromotor za krokový motor nebo jeho obdobu, avšak v tomto případě nastává problém přenesení pohybu na kola samotná, jelikož je toho zde docíleno pomocí tenkých kovových tyčí, které jsou napojené přímo na původní elektromotor, viz obrázek 9 a 10.



Obrázek 6 ovládání předních kol



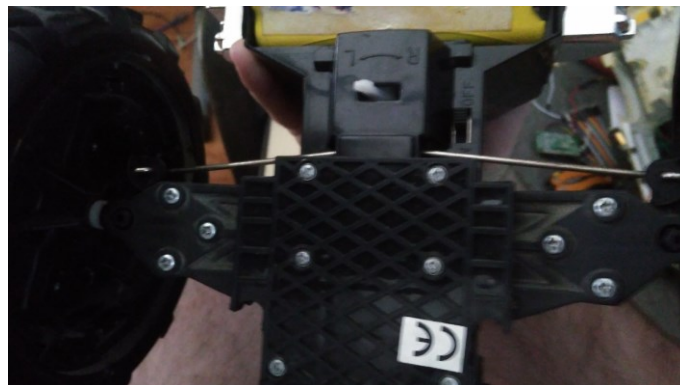
Obrázek 7 ozubená kola ovládání



Obrázek 8 senzory



Obrázek 9 vodící tyč



Obrázek 10 vodící tyč napojená na kola

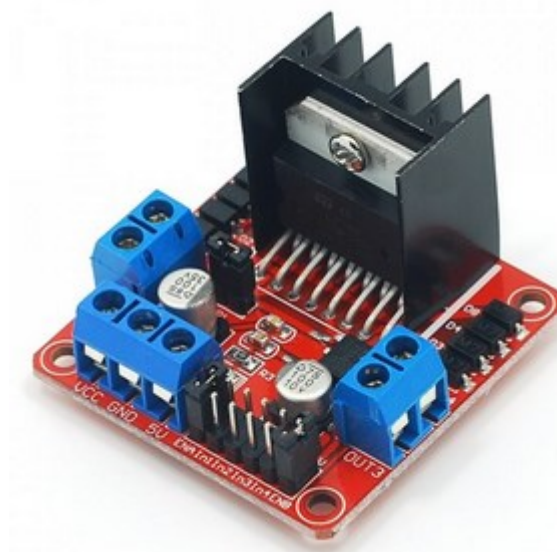
Co se týče elektromotoru pohánějícího zadní kola, situace je mnohem jednodušší. Elektromotor je napojen na zadní kola a drát je příhodně zaveden do hlavní části modelu, kde se nacházel původní mikropočítač. Není tudíž potřeba s ním jakkoliv manipulovat, stačí pouze napájecí kabely zapojit do motor shieldu, který se už postará o zvolenou rychlost a směr otáčení kol.



Obrázek 11 pohon zadních kol

7.3 Motor shield

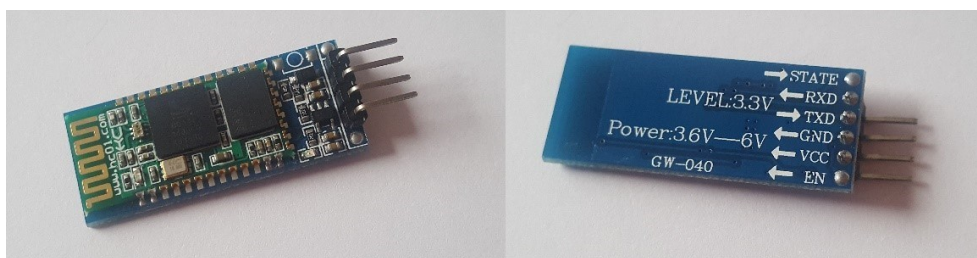
Za účelem snazšího zapojení a ovládání elektromotorů byl zvolen Arduino rozšiřující modul motor shield L298N. Tento modul je určen k ovládání jednoho krokového motoru nebo až dvou stejnosměrných elektromotorů. Napájení shieldu může být až do hodnoty 45 V, napětí pro motory je poté v rozsahu 5-35 V.



Obrázek 12 motor shield L298N

7.4 Bluetooth modul

Pro umožnění komunikace mezi deskou Arduino a mobilním zařízením pomocí Bluetooth byl zvolen modul HC-06. Jedná se o jeden z nejpobulárnějších a nejrozšířenějších modulů na trhu, je dobře dostupný se spoustou návodů na použití. Rychlost přenosu dat sériovou linkou je 9600 baudů. Baud je jednotka modulační rychlosti, která udává počet změn stavu přenosového média za jednu sekundu. Modul má integrovanou anténu s dosahem 10 metrů. Optimální vstupní napětí je v rozmezí 3,3 až 5 V a nemělo by být vyšší než 7 V, protože se tím modul zničí. Bluetooth modul pracuje na režimu Slave, což znamená, že k připojení potřebuje Mastera (chytré telefony pracují v obou režimech) a nemůže se připojit k dalšímu Slave modulu. Režim Master žádá další zařízení o navázání spojení, kdežto režim Slave umožňuje pouze přijímat žádosti.



Obrázek 13 bluetooth modul HC-06

7.5 Deska Arduino

Řídící jednotkou celého modelu bylo zvoleno Arduino Uno R3 od společnosti RobotDyn. Jedná se o kopii originálního Una se stejnými parametry. Arduino Uno bylo zvoleno, protože má optimální velikost, dostatečný počet pinů, USB pro jednoduché nahrávání programů a zároveň je velmi rozšířenou deskou se spoustou návodů, jak s ní pracovat. Díky rozšířenosti je tato deska snadno dostupná na trhu od různých výrobců. Detailnější informace, spolu s obrázkem, se nacházejí v teoretické části, kapitola 3.1, strana 15.

7.6 Napájení

Jako zdroj napájení modelu byl zvolen původní osmi článkový Ni-Cd akumulátor s kapacitou 700 mAh o napětí 9,6 V. Konektor akumulátoru je typu „Tamiya“ 2 - pin.

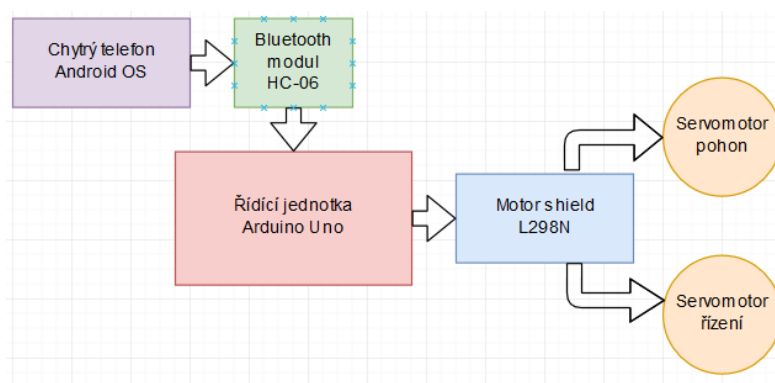


Obrázek 14 Ni-Cd 9.6 V akumulátor

8 ZAPOJENÍ

8.1 Blokové schéma

Obrázek č.15 představující blokové schéma zapojení se skládá ze šesti prvků. Blok Chytrý telefon představuje počátek komunikace v zařízení, přichází z něj příkazy pro pohyb. Tyto příkazy zachytí blok Bluetooth modul, který je předá dál. Blok Řídící jednotka přijímá tyto dekodované příkazy z Bluetooth modulu a předává je skrze blok Motor shield do bloku Elektromotor pohon či blok Elektromotor řízení.



Obrázek 15 blokové schéma

Shrnutí:

Chytrý telefon vyše příkaz odpovídající tlačítku zobrazenému na displeji. Tento příkaz putuje Bluetooth modulem do řídicí jednotky, která daný příkaz rozluští a předá motor shieldu, který vykoná odpovídající funkci. Příklad: tlačítko rychlost 1 roztočí elektromotor na nejnižší rychlost.

8.2 Zapojení

Jakmile máme všechno připraveno a ujasněné blokové zapojení, můžeme se pustit do samotného zapojení. Zapojovat budeme do desky Arduina jakožto řídicího prvku. Deska potřebuje napájení, k tomu použijeme přívod od baterie z původního modelu. Stejný přívod použijeme i pro napájení motor shieldu. Desku napájíme pomocí pinů VIN a GND, motor shield poté přes piny +12 V a GND. Piny Bluetooth modulu, které umožňují komunikaci s deskou zapojíme následovně: pin Rx na Arduino pin 5 a pin Tx na Arduino pin 4. Z pinů ENA a ENB nacházejících se na motor shieldu odstraníme svorky a připojíme na Arduino piny 11 a 10. Piny INT 1 a 2 připojíme na Arduino piny 12 a 13, INT 3 a 4 poté na Arduino

pinů 9 a 8. Pro lepší orientaci se odkážeme na tabulku 2. Nakonec zapojíme dráty elektromotorů do pinů OUT 1 a 2 pro motor pohonu a pinů OUT 3 a 4 pro motor řízení.

Tabulka 1 zapojení pinů

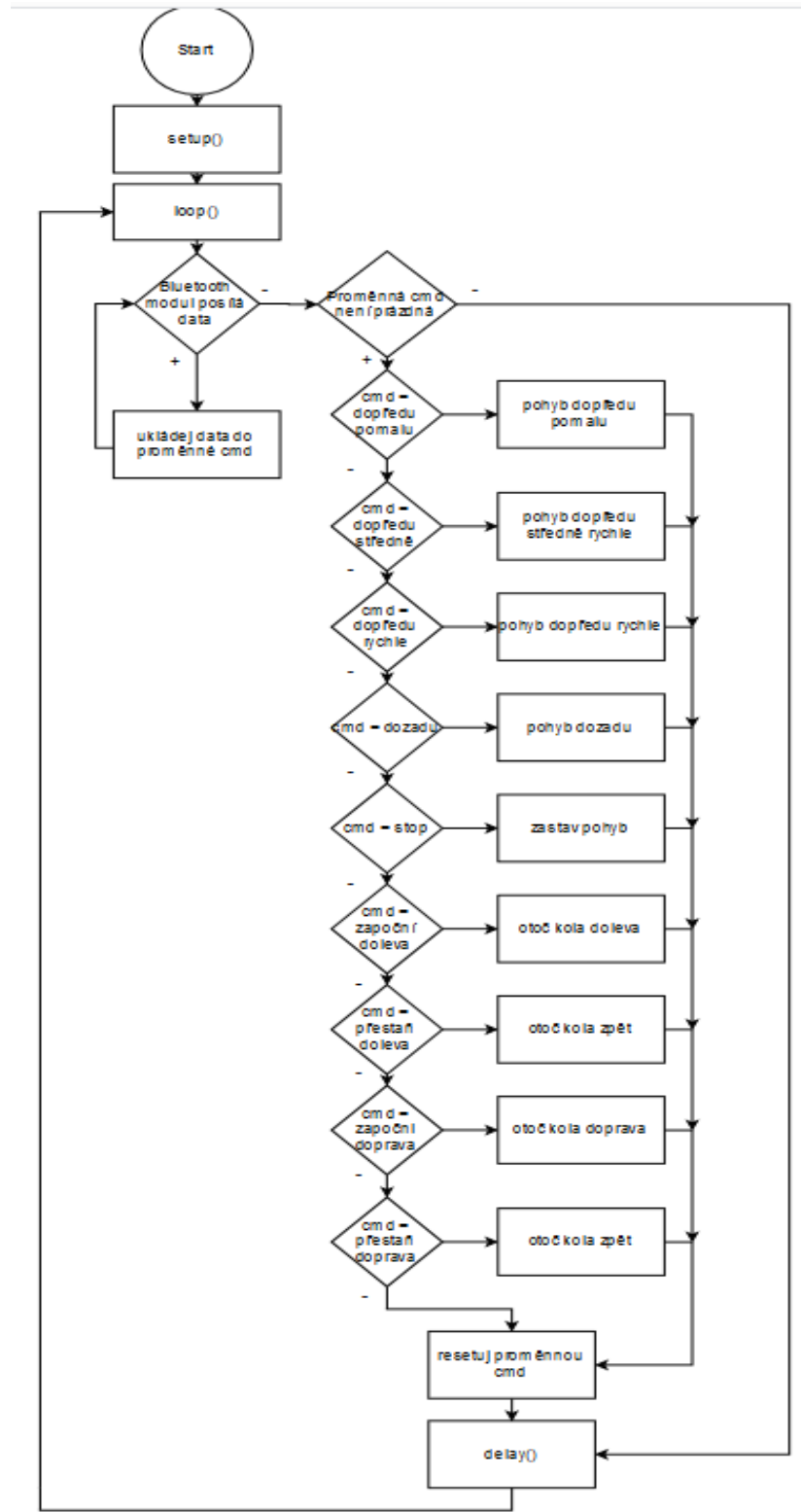
Modul	Piny modulu	Piny Arduina
Motor Shield	ENA	11
	ENB	10
	INT1	12
	INT2	13
	INT3	9
	INT4	8
HC-06	Rx	5
	Tx	4
	GND	GND
	VCC	+5 V

9 KÓD ARDUINA

V této kapitole se budeme zabývat vytvářením programu, který bude řídit naše Arduino.

9.1 Vývojový diagram

Jako první se ve funkci `setup()` inicializuje Bluetooth sériový port a piny Arduina na výstupy. Po skončení této funkce se automaticky spustí funkce `loop()`. Tato funkce běží po celou dobu, kdy je zařízení zapnuto a vykonávají se v ní všechny hlavní funkce programu. Účelem funkce `loop()` je načíst data přicházející ze sériového portu Bluetooth modulu a uložit si je do proměnné `cmd`. Následně se pracuje s touto proměnnou, porovnáváme její obsah s předdefinovanými konstantami v sérii podmínek a podle shody posíláme na elektromotory chtěné příkazy. Po vykonání příkazu se proměnná `cmd` vyresetuje a na konec se počká krátkou dobu než se začne funkce `loop()` znovu opakovat. Vývojový diagram se nachází na obrázku číslo 16.



Obrázek 16 vývojový diagram

9.2 Popis kódu

Jako první si definujeme knihovny, se kterými budeme pracovat. V tomto projektu je použita jedna knihovna, a to SoftwareSerial. Tu použijeme kódem „#include <SoftwareSerial.h>“.

Následně nastavíme, na jakých pinech Arduina se sériová linka nachází a to takto: „SoftwareSerial hc06(4,5);“, kde *hc06* je námi zvolené libovolné jméno a čísla *4* a *5* jsou piny. Poslední věc, kterou musíme udělat, před funkcí `setup()`, je definování globálních proměnných. Celkem se jich zde nachází sedm. String `cmd` se používá k ukládání příkazů, které přicházejí ze sériové linky a následně je použit k určení co se má vykonat. Zbylých šest proměnných typu `integer` s názvy `enA`, `enB` a `in 1` až `4` definují na jakých pinech se nachází motor A a motor B.

```
#include <SoftwareSerial.h>
SoftwareSerial hc06(4,5);
String cmd="";

//motor A
int enA = 11;
int in1 = 12;
int in2 = 13;

//motor B
int enB = 10;
int in3 = 9;
int in4 = 8;
```

Obrázek 17 definice knihoven a proměnných

Následně přecházíme do funkce `setup()`. Zde dochází k prvotnímu nastavení pinů, inicializace sériového portu a případných dalších prvků, kdybychom nějaké měli. K inicializaci dochází pomocí příkazu „`hc06.begin(9600)`“, `begin()` je funkce knihovny `SoftwareSerial`, kterou jsme předtím nadefinovali a hodnota *9600* je rychlost přenosu dat.

K nastavení pinů použijeme funkci „`pinMode`“, která je nativně implementovaná ve vývojovém prostředí Arduino. V závorce této funkce, se nachází číslo pinu, který se má nastavit a to, zda se jedná o výstupní, vstupní či výstupně-vstupní pin. Protože jsme si definovali globální proměnné typu `integer` s hodnotami, které odpovídají našim pinům, do první části příkazu „`pinMode`“ nepíšeme číslo, protože by to bylo redundantní, ale vepíšeme zde název proměnné. Všechny naše piny jsou výstupní.

Po proběhnutí funkcí `setup()` se program dostane do hlavní funkce `loop()`, která obsahuje hlavní tělo programu. Ta bude probíhat až do vypnutí mikrokontroleru. V případě potřeby, se dá dočasně vyskočit z funkce `loop()` do libovolné jiné funkce, kterou si můžeme buďto sami vytvořit mimo blok kódu `loop()` a `setup()`, nebo to může být funkce obsažená

v některé z knihoven, které jsme si naimportovali. V každém případě se z této funkce musíme vrátit zpět. V našem kódu se kromě funkce `loop()` a `setup()` nenacházejí žádné další námi vytvořené funkce.

V našem případě program začíná cyklem *while*, který čte znak po znaku, jak přicházejí ze sériové linky a ukládá je do proměnné `cmd`. Dělá tak do té doby, dokud existují data v bufferu sériové linky. V kódu je toto zapsáno jako příkaz „`hc06.available() > 0`“. Dalším krokem je zjištění, zda se do proměnné `cmd` něco uložilo. Toho je docíleno jednoduchou podmínkou „`if(cmd != "")`“. V případě, že se v proměnné `cmd` nic nenachází, nejedná se nutně o chybu, tudíž se přeskočí na konec, kde procesor chvíli počká a celá funkce se spustí znovu. Pokud však podmínka proběhla úspěšně, přichází na řadu rozpoznání příkazu. K tomu dochází pomocí série příkazů `if-else if`. Procesor postupně prochází řadou podmínek a porovnává, zda se obsah proměnné `cmd` rovná některému ze stringů, které jsme si přiřadili k námi dané činnosti, viz obrázek níže.

```
if(cmd=="F_L")
{
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    analogWrite(enA, 100);
}
else if(cmd=="F_M")
{
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    analogWrite(enA, 170);
}
else if(cmd=="F_H")
{
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    analogWrite(enA, 250);
}
```

Obrázek 18 část hlavního kódu uvnitř `loop()`

V případě, že se obsah proměnné `cmd` neshoduje s prvním stringem „`F_L`“, který reprezentuje pohyb dopředu v malé rychlosti, pokračuje se na dalším příkazu `else if`, dokud se nenalezne shoda. Pokud se obsah proměnné `cmd` neshoduje se žádnou z podmínek, nastaví se na prázdný string, procesor chvíli počká a opakuje se funkce `loop()`. Když procesor při porovnání narazí na shodu, pokračuje se do těla podmínky.

Příklad: v první podmínce se pin *in1* motoru, který ovládá pohon, nastaví na logickou 1 a pin *in2* na logickou 0. Tím se určí, jakým směrem se mají točit kola autíčka (v našem případě dopředu). Zapsáním hodnoty 100 na pin *enA* nastavíme požadovanou rychlost. Rozmezí rychlosti je 0, kdy se kola vůbec netočí a 255, kdy se točí plným výkonem.

Maximální hodnota je rovna 255 jelikož se jedná o osmibitové číslo, tedy $2^8 = 256 - 1$, -1 se zde nachází kvůli hodnotě 0, 0 až 255 kombinací je rovno 256.

Po vykonání všech příkazů umístěných v těle podmínky se přeskočí ke konci kódu, kde se proměnná *cmd* opět nastaví na prázdný string, aby nedocházelo k chybám, procesor chvíli počká, v našem případě se konkrétně jedná o funkci „*delay(100)*“, kde *delay()* je integrovaná funkce Arduina a hodnota 100 je v milisekundách. Následně se opět opakuje funkce *loop()*.

10 APLIKACE PRO ANDROID

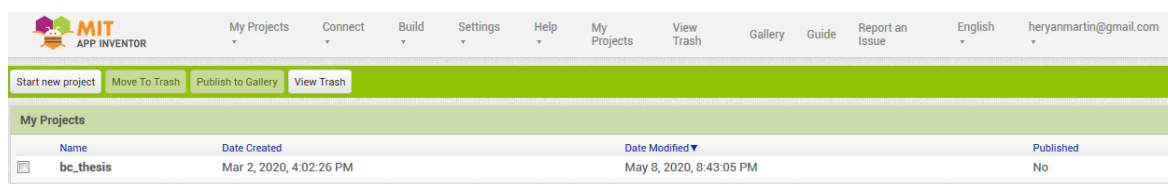
V této části se budeme podrobněji zabývat vývojovým prostředím, ve kterém je aplikace vytvořena a dále samotným procesem vývoje naší aplikace.

10.1 MIT App Inventor

MIT App Inventor je internetové vývojové prostředí zaměřené na jednoduchý vývoj aplikací pro chytrá zařízení jako jsou telefony nebo tablety. V této kapitole si ukážeme, jak s ním pracovat

10.1.1 Vytvoření nového projektu

Abychom mohli pracovat v Inventoru, musíme se nejdříve zaregistrovat na webové adrese appinventor.mit.edu. Lze toho dosáhnout sloučením s účtem Google, přes který se následně budeme přihlašovat, takže není nutné tvořit si nové přihlašovací jméno a heslo.



Obrázek 19 Inventor

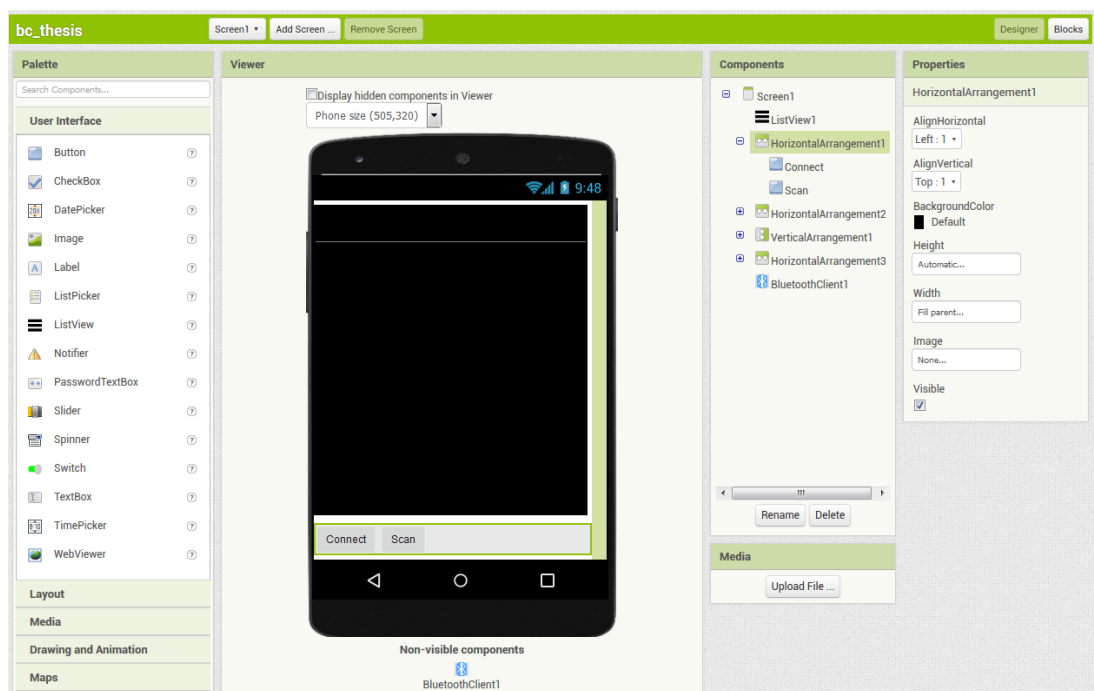
Po přihlášení se dostaneme na úvodní obrazovku obdobnou jako je na obrázku výše. Po kliknutí na tlačítko *Start new project* nám vyskočí okénko pro pojmenování projektu. Jakmile potvrdíme zadané jméno, budeme přesměrováni na další stránku, kde už budeme moci začít pracovat.

Okno vývojového prostředí se skládá z několika částí. Na vrchu okna najdeme lištu s názvem projektu a pěti tlačítky. Tyto tlačítka slouží na přepínání mezi několika obrazovkami, které můžeme v projektu mít, přidání a odebrání obrazovky a na přepínání mezi vizuální stránkou projektu a její funkční částí. Mezi částmi designu a pseudokódu můžeme libovolně přepínat a každá obrazovka má své vlastní.

Část vývojového prostředí zaměřená na design aplikace je rozdělena na pět podoken. *Palette*, je část obsahující nástroje, které budou v naší aplikaci obsaženy. Mezi tyto nástroje patří tlačítka, labely, textBoxy, ale také například BluetoothClient, nástroj pro

konektivitu užitečný v našem projektu. Tyto nástroje použijeme přetažením ze sekce *Palette* do sekce *Viewer*. V této sekci se nám zobrazuje náhled, jak bude naše aplikace přibližně vypadat. Vpravo od této sekce se nachází *Components*. Zde můžeme vidět všechny nástroje, které aktuálně používáme. Kliknutím na komponent jej v seznamu *Components* zvýrazní a zároveň nám jej ohraničí ve *Viewer* pro lepší orientaci.

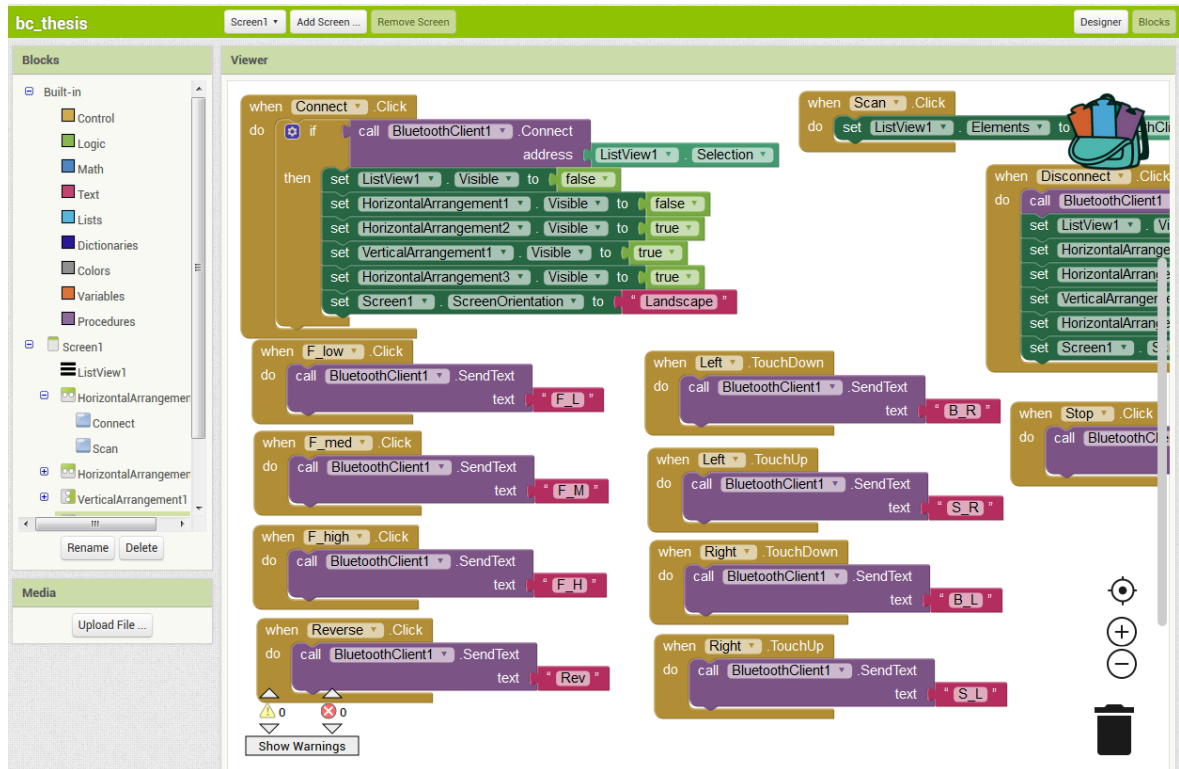
V předposlední sekci *Properties* se nám po kliknutí na položku ze seznamu zobrazí příslušné nastavení, které můžeme upravovat. Poslední částí je *Media*, kde máme možnost nahrát vlastní obrázek pro pozadí apod.



Obrázek 20 designer Inventoru

Sekce *Blocks* vývojového prostředí, zaměřená na logiku aplikace má celkem tři části: *Blocks*, *Viewer* a *Media*. Okénko *Media* je totožné a sdílené s *Designer*. V části *Blocks* najdeme list zabudovaných funkcí společně se všemi komponenty, které jsme si v okně *Designer* vybrali. Po kliknutí na jeden z prvků se nám zobrazí bloky pseudokódu týkající se daného konkrétního prvku, které následně přetáhnutím umístíme libovolně do *Viewer*. Funkcionalita aplikace se poté programuje napojováním jednotlivých bloků do sebe. Umístění na sebe nenavazujících částí kódu je nepodstatné a nemá na chod aplikace žádný vliv. Příklad funkčnosti: Blok „when F_low.Click do“ nalezneme kliknutím na tlačítko *F_low* v sekci *Blocks* které je na obrázku 22 pod srolovaným *VerticalArrangement1*, to přetáhneme do *Viewer*, klikneme na *BluetoothClient1* a vybereme „Call *BluetoothClient1.SendText* text“. Tento blok přesuneme dovnitř prvního bloku, oba bloky

se automaticky spojí. Posledním blokem je „F_L“, který nalezneme v sekci *Built-in text* a stejným způsobem přiložíme k předchozím dvěma. Když pracujeme s *Built-in* prvky, musíme ručně dodat potřebné parametry, při výběru textu nalezneme pouze prázdný blok, do kterého vepíšeme text, který chceme, aby obsahoval.



Obrázek 21 sekce Blocks

10.1.2 Finalizace aplikace

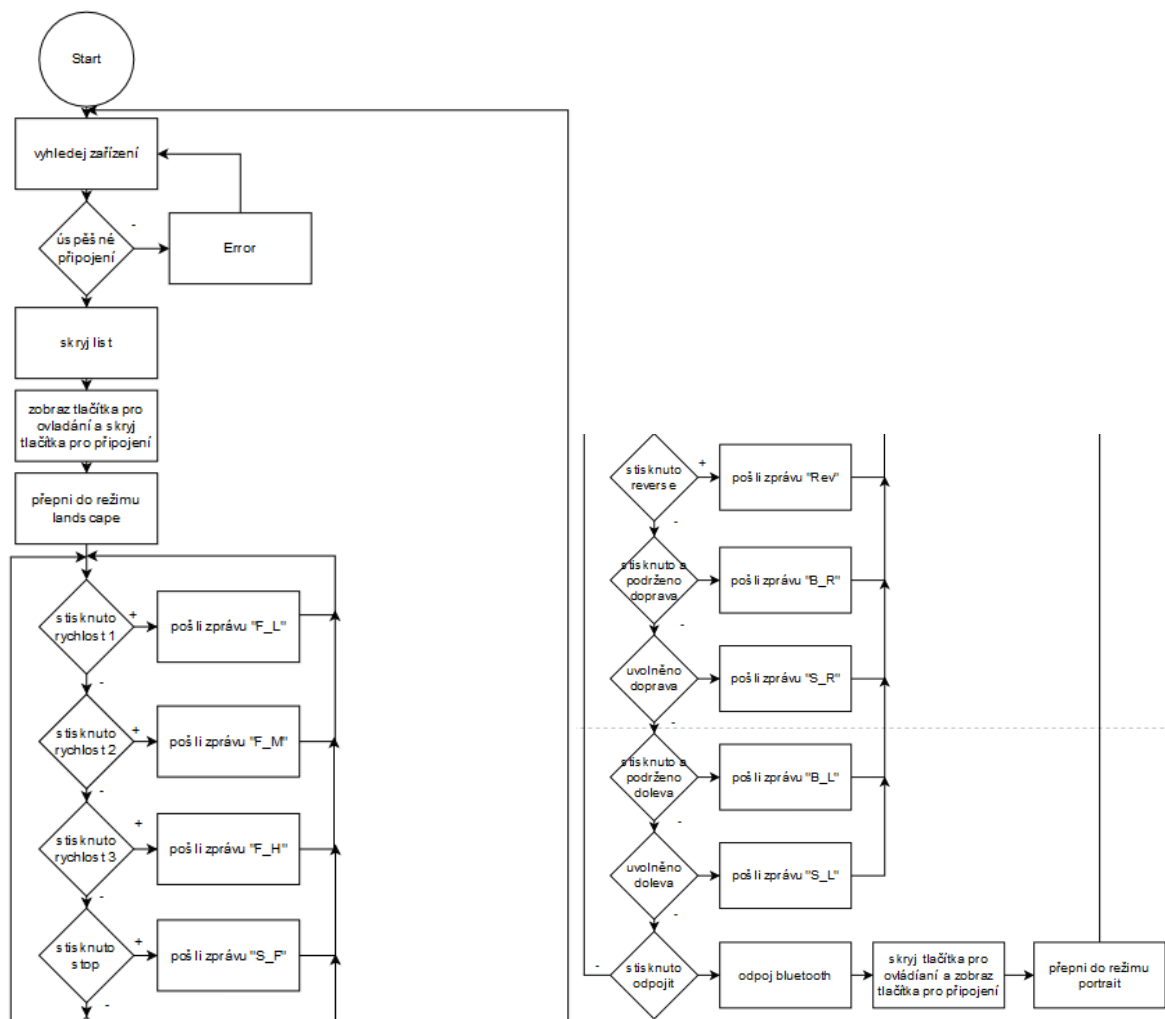
Po dokončení aplikace máme několik možností, jak aplikaci otestovat a finalizovat. V horním panelu na stránce, v záložce *Connect* máme možnosti *AI Companion*, *Emulator* a *USB*. První a poslední možnost se týkají fyzického zařízení, které máme po ruce, kde aplikaci můžeme vyzkoušet buďto pomocí MIT App Inventor aplikace dostupné na Google Play nebo přenesením přes USB kabel. Možnost *Emulator* vyžaduje dodatečný software od MIT, ve které budeme následně pracovat. V případě, že jsme s aplikací spokojeni a chceme ji finalizovat případně distribuovat jako .apk soubor, je zde záložka *Build* ve které máme možnost si soubor uložit do počítače nebo transferovat pomocí QR kódu do mobilního zařízení.

10.2 Tvorba aplikace

Jelikož je aplikace tvořena v MIT App Inventoru pomocí abstraktního blokového programování, samotný kód aplikace je nedostupný. Z toho důvodu si zde představíme logiku, na které je aplikace založena.

10.2.1 Vývojový diagram aplikace

Vývojový diagram byl vzhledem k rozměrům rozdělen na dvě části. Jedná se o abstraktní pojetí principu funkčnosti z důvodu způsobu programování.

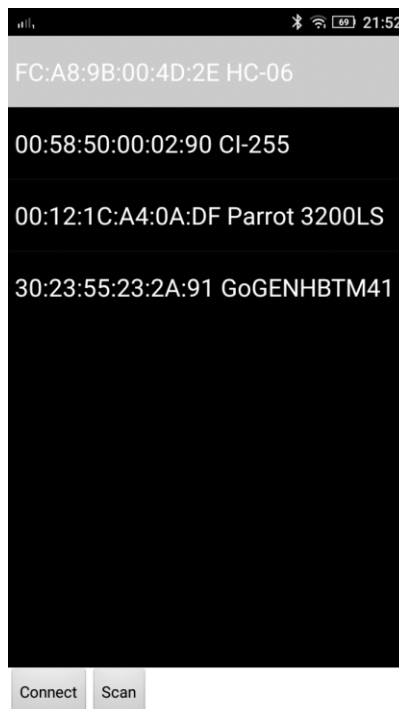


Obrázek 22 vývojový diagram

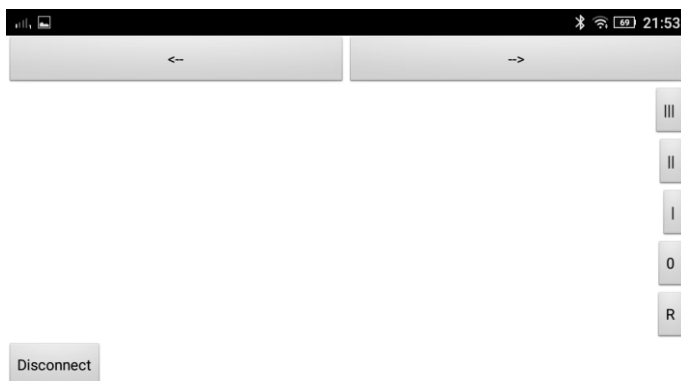
10.2.2 Design aplikace

Jednoduchost a omezenost vývojového prostředí se odráží na kvalitě designu aplikace. Aplikace obsahuje jednu obrazovku, na které se ihned po spuštění nacházejí všechny prvky, avšak ty, jež jsou aktuálně nepoužívané jsou skryty. Po spuštění tedy lze vidět prázdný seznam, tlačítka *Connect* a *Scan*. Po stisknutí tlačítka *Scan* se seznam naplní dostupnými

zařízeními disponujícími aktivovaným Bluetooth. Kliknutím na jméno zařízení ze seznamu se zvýrazní a po kliknutí na *Connect* a úspěšném připojení se aplikace přepne do režimu na šířku, skryje tyto tlačítka i list a zobrazí tlačítka pro pohyb, zatáčení a odpojení.



Obrázek 23 aplikace po kliknutí na *Scan*

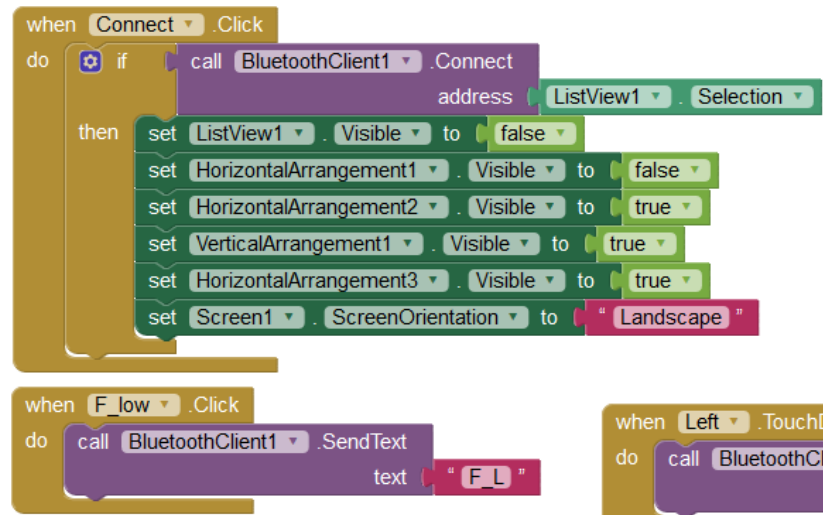


Obrázek 24 aplikace po připojení k zařízení

10.2.3 Funkcionalita aplikace

Logika programu je naprogramována abstraktním blokovým kódem. Jedná se o 12 samostatných bloků obsluhující jednotlivá tlačítka, která se nacházejí v designu aplikace. Tlačítko *Scan* vyhledá všechny dostupné zařízení v okolí a vloží je do seznamu. Po stisknutí tlačítka *Connect* se telefon zkusí připojit k zařízení které je ze seznamu vybráno a při neúspěchu vypíše chybové hlášení. Při úspěšném připojení skryje tlačítka *Connect* a *Scan*, seznam zařízení a přepne telefon do horizontálního režimu. Dále pak zobrazí tlačítka pro

pohon, zatačení a odpojení. Tlačítko *Disconnect* funguje jako opak *Connect*, zobrazí skryté prvky, skryje ty zobrazené, odpojí Bluetooth od zařízení a přepne obrazovku do vertikálního režimu. Tlačítka pro zatačení vysílají zprávu při stisknutí a při uvolnění, tím je zajištěno, že kola budou otočena, dokud je tlačítko stisknuto a vrátí se do původní polohy po jeho uvolnění. Pět tlačítek *R*, *0*, *I*, *II* a *III* udávají rychlost autíčka. Tyto tlačítka vysílají zprávu přes Bluetooth při pouze při stisknutí.



Obrázek 25 ukázka blokového kódování

11 ZPROVOZNĚNÍ MODELU A APLIKACE

Abychom celý systém uvedli do provozu musíme zařízení nejprve spárovat s mobilním telefonem. Připojíme nabitý akumulátor o odpovídajícím napájení (9,6 V) k *Tamiya* konektoru ve spodní části autíčka a přepneme přepínač do polohy ON. V nabídce dostupných zařízení na telefonu klikneme na položku *HC-06* a zadáme kód, defaultně 1234. Následně na mobilním zařízení spustíme aplikaci, klikneme na tlačítko *Scan*, vybereme Bluetooth modul a klikneme na *Connect*, displej se nám otočí do horizontální polohy a zobrací se nám tlačítka pro ovládání.

Model autíčka je již poměrně starý a ani ve svých dobrých dnech zatáčení nebylo jedno z nejlepších. Způsob, jakým bylo zatáčení řešeno není ideální, a i když svou práci odvede, modernější krokový motor s jemnějším zatáčením by byl lepší. Baterie po letech používání také již nemá maximální kapacitu a v budoucnosti by se konektor „Tamiya“ mohl stát nesehnatelným, použití například Li-Pol, Li-Ion nebo dokonce USB je mnohem lepším řešením.

ZÁVĚR

Cílem práce bylo prozkoumat existující způsoby ovládání, zvolit komponenty vhodné pro řízení skrze rozhraní Bluetooth, přetvořit existující model autíčka ovládaného rádiovými vlnami na řízení pomocí mobilního telefonu skrze rozhraní Bluetooth a vytvořit aplikaci pro ovládání modelu pomocí mobilního telefonu.

Základním modelem byla hračka pro děti Ford Explorer Monster Truck, ze které byly ponechány elektromotory pro řízení a pohon, a baterie i s přívodem. Starou řídicí jednotku nahradila deska Arduino Uno, do které je zapojen externí Bluetooth modul HC-06, zprostředkávající komunikaci s mobilním zařízením, a dále rozšiřující deska motor shield L298N, díky které lze snadno ovládat oba dva elektromotory. Kód pro desku Arduino byl napsán v jazyce C/C++ ve vývojovém prostředí Arduino IDE. Mobilní aplikace byla vytvořena v abstraktním blokovém kódu webového vývojového prostředí MIT App Inventor.

Všech cílů bylo dosaženo, aplikace i model jsou funkční. V budoucnosti by bylo dobré nahradit stávající elektromotor ovládající řízení například za krokový motor a následně upravit aplikaci, a umožnit tak lepší řízení. Dále by se práce dala rozšířit například o senzory proti nárazu či LED osvětlení.

SEZNAM POUŽITÉ LITERATURY

[1] Phone-Controlled Cars. *Technology* [online]. Technology, 2019 [cit. 2020-07-27].

Dostupné z: <https://www.technology.org/2019/06/11/what-are-the-latest-technological-trends-of-remote-control-toys/>

[2] Radio-controlled car. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA):

Wikimedia Foundation, 2001- [cit. 2020-07-27]. Dostupné z:

https://en.wikipedia.org/wiki/Radio-controlled_car

[3] PINKER, Jiří. *Mikroprocesory a mikropočítače*. Praha: BEN, 2004. ISBN 80-7300-110-1.

[4] Co je to Arduino? *Arduino* [online]. 2014- [cit. 2020-07-27]. Dostupné z:

<https://arduino.cz/co-je-to-arduino/>

[5] Robotdyn Arduino UNO R3. *Laskarduino* [online]. 2020 [cit. 2020-07-27]. Dostupné z:

<https://www.laskarduino.cz/robotdyn-classic-uno-r3-atmega328p--arduino-kompatibilni/>

[6] Understanding bluetooth security. *Decipher* [online]. Loveless, 2018 [cit. 2020-07-28].

Dostupné z: <https://duo.com/decipher/understanding-bluetooth-security>

[7] SCHILDT, Herbert. *Mistrovství - Java*. Brno: Computer Press, 2014. Mistrovství. ISBN 978-80-251-4145-8.

[8] GRIFFITHS, Dawn a David GRIFFITHS. *Head first android*. 2. nd ed. O'Reilly Media, 2017. ISBN 9781491974056.

[9] About Us. *MIT App Inventor* [online]. Massachusetts, 2012- [cit. 2020-07-28]. Dostupné

z: <http://appinventor.mit.edu/about-us>

[10] NOVÁK, Petr. *Mobilní roboty: pohony, senzory, řízení*. Praha: BEN, 2005. ISBN 80-7300-141-1.

[11] MARGOLIS, Michael. *Arduino Cookbook*. 2nd ed. Sebastopol: O'Reilly, 2012. ISBN 1449313876.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

HTML Hypertext Markup Language

CSS Cascading Style Sheet

MIT Massachusetts Institute of Technology

IDE Integrated Development Environment

RX Receiver

TX Transmitter

USB Universal Serial Bus

RISC Reduced Instruction Set Computer

CISC Complex Instruction Set Computer

PWM Pulse Width Modulation

MITM Man in the middle

DIY Do it yourself

SEZNAM OBRÁZKŮ

Obrázek 1 Arduino Uno.....	14
Obrázek 2 modele autíčka ze strany	22
Obrázek 3 model zepředu	23
Obrázek 4 užitečný prostor	23
Obrázek 5 původní řídicí jednotka.....	24
Obrázek 6 ovládání předních kol	25
Obrázek 7 ozubená kola ovládání	25
Obrázek 8 senzory	25
Obrázek 9 vodící tyč	26
Obrázek 10 vodící tyč napojená na kola	26
Obrázek 11 pohon zadních kol	27
Obrázek 12 motor shield L298N	28
Obrázek 13 bluetooth modul HC-06.....	28
Obrázek 14 Ni-Cd 9.6 V akumulátor.....	29
Obrázek 15 blokové schéma	30
Obrázek 16 vývojový diagram.....	33
Obrázek 17 definice knihoven a proměnných	34
Obrázek 18 část hlavního kódu uvnitř loop()	35
Obrázek 19 Inventor	37
Obrázek 20 designer Inventoru.....	38
Obrázek 21 sekce Blocks.....	39
Obrázek 22 vývojový diagram.....	40
Obrázek 23 aplikace po kliknutí na <i>Scan</i>	41
Obrázek 24 aplikace po připojení k zařízení	41
Obrázek 25 ukázka blokového kódování.....	42

SEZNAM TABULEK

Tabulka 1 zapojení pinů.....	31
------------------------------	----