TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií

Bezdrátové řízení vozítka pomocí mobilního telefonu

Bakalářský projekt

Studijní program: B2646 – Informační technologie
Studijní obor: 1802R007 – Informační technologie

Autor práce: **Tomáš Moravec**
Vedoucí práce: Ing. Radek Srb

Liberec 2015

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky a mezioborových studií
Akademický rok: **2014/2015**

ZADÁNÍ BAKALÁŘSKÉHO PROJEKTU

Jméno a příjmení:	**Tomáš Moravec**
Název práce:	**Bezdrátové řízení vozítka pomocí mobilního telefonu**
Zadávající katedra:	**Ústav mechatroniky a technické informatiky**
Vedoucí práce:	**Ing. Radek Srb**
Rozsah práce:	**15—20 stran**

Zásady pro vypracování:

1. Přestavte analogový RC model na digitálně řízené vozítko.
2. Prozkoumejte možnosti bezdrátového řízení vozítka pomocí mobilního telefonu.
3. Vybrané řešení realizujte po hardwarové i softwarové stránce.

Seznam odborné literatury:

[1] MARGOLIS, Michael. Make an Arduino-controlled robot. 1st ed. Sebastopol, Calif.: O'Reilly, c2013, xvi, 238 p. Make. ISBN 14-493-4437-2.

[2] SCHILDT, Herbert. Java 7: výukový kurz. 1. vyd. Brno: Computer Press, 2012, 664 s. ISBN 978-80-251-3748-2.

V Liberci dne ....

Ing. Radek Srb

Prohlášení

Byl jsem seznámen s tím, že na můj bakalářský projekt se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářský projekt nebo poskytnu-li licenci k jeho využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářský projekt jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum: 15. 5. 2015

Podpis:

Abstrakt

Tento projekt shrnuje mé vlastní řešení tvorby robotického vozítka, kompletním předěláním elektroniky radiově řízeného modelu, zachováním jeho šasi a vytvořením aplikace pro bezdrátové řízení pomocí technologie Bluetooth na mobilní platformu Android. Původně analogový hardware byl nahrazen digitální technologií integrovaných obvodů včele s jednočipovým počítačem ATmega328P, který byl kompletně naprogramován v jazyce C++ s využitím knihovny Wiring. Software pro bezdrátové řízení byl vytvořen v jazyce Java za pomoci interaktivního vývojového prostředí App Inventor, výsledkem je digitální joystick umožňující pohyb do všech stran.

Klíčová Slova:

Integrované obvody (IO), Bluetooth, Knihovna Wiring, C++, Java, App Inventor

Abstract

This project summarizes my own solution of creating robotic rover, complete remanufacture electronic radio-controlled model, preserving its chassis and developing applications for wireless control via Bluetooth based on Android mobile platform. Originally analog hardware has been replaced by digital technologies integrated circuits headed by microcontroller computer ATmega328P, which has been completely programmed in C ++ using the libraries Wiring. Software for wireless control was created in Java with the help of an interactive development environment App Inventor, the result is a digital joystick allowing movement in all directions.

Keywords:

Integrated circuits (IC), Bluetooth, Wiring library, C++, Java, App Inventor

Obsah

1 Úvod

Naším cílem je předělání analogového RC modelu, na plně digitální vozítko řízené za pomoci jedné z bezdrátových technologií, vyskytujících se v mobilních telefonech. Prvním krokem je výběr vhodného modelu s co nejlepšími parametry, tedy začneme úvodem do problematiky.

1.1 Volba modelu

1.1.1 Velikost

Při výběru vhodného RC modelu je nutné zvážit velikost vnitřních prostor a celkově. Analogová technika je sice povětšinou prostorná, zato případné senzory, měniče napětí a další zabírají taktéž množství prostoru. Proto je třeba brát velikost v potaz, zvolení nevhodných rozměrů, může v budoucnu způsobit problémy s prostorem.

1.1.2 Typ pohonu

Důležitým prvkem je pohon. Každý má své specifické vlastnosti a každý se jinak ovládá. Zde jsou příklady třech nejčastějších.

1.1.2.1 Krokový

Krokový motor převádí digitální impulsy na postupné otáčení hřídele motoru. Počet kroků je úměrný generovaným pulsům a počet otáček, je funkce frekvence vstupních impulsů. Obvykle jeden puls otáčí rotorem o jeden krok. Je možné jej řídit přímo mikrokontrolérem. Při delší neaktivitě hrozí přehřátí. Jeho řízení je složitější, protože je nutné používat driver.

[+] Výhody

- Dostupný
- Velice přesný
- Otáčení dvěma směry
- Při nulové rychlosti stojí zcela na místě

[–] Nevýhody

- Pomalý
- Odebírá proud v klidovém stavu.
- Při přetížení ztratí nenávratně pozici
- Žádná zpětná informace o skutečné poloze

1.1.2.2 Stejnosměrný motor

Jedná se o nejstarší typ motoru. Elektromotor je napájený stejnosměrným proudem. Otáčení je prováděno pomocí rychlých změn polarit magnetů. Kvůli velkým nárazovým proudům je nutné jej řídit přes výkonový H-můstek, díky kterému je vše stejné jako u krokového motoru.

[+] Výhody

- Rychlý
- Plynulý
- Jednoduché řízení
- Otáčení dvěma směry

[–] Nevýhody

- Velké nárazové odběry

1.1.2.3 Spalovací

Jedná se o nejrozšířenější způsob pohonu modelů, který poskytuje příznivý poměr cena/výkon. Vzhledem k jeho hlučnosti, rychlosti a síle, je jako pohon pro autonomní vozítko nevhodný.

1.1.3 Napájení

1.1.3.1 Li-pol

Nejnovější typ akumulátoru, který se používá téměř ve všech elektronických zařízeních. Jedná se o vylepšení Li-ion baterií (notebookové), které opravuje jejich nedostatky.

[+] Výhody

- Velká kapacita
- Vysoký proud
- Jednoduché zapojení článků do série

[–] Nevýhody

- Vysoká cena

1.1.3.2 NiMH

Jedná se o starší, ale zároveň jeden z nejpoužívanějších typů baterií.

[+] Výhody

- Eliminuje většinu nedostatků NiCd

[–] Nevýhody

- Nízké napětí a proud

1.1.3.3 NiCd

Starý typ baterie, dnes hojně se používá v levných zařízeních, díky své nízké ceně.

[+] Výhody

- Nízká cena

[–] Nevýhody

- Nízká kapacita
- Nízké napětí a proud
- Krátká životnost

1.1.4 Finální výběr

1.1.4.1 Model

Vybraný RC model [Obrázek 1] má šasi plastové konstrukce, která neohrozí komunikaci. Velký vnitřní prostor poslouží k budoucímu rozšiřování DPS (desky plošných spojů). Obsahuje odpružení pro případný pohyb v terénu, které navíc zvýší životnost spojů. Dále přišroubované náhražky světel, které lze pohodlně nahradit senzory. Volba velkého modelu taktéž umožní budoucí experimenty a usnadní dosavadní vývoj.

Obrázek 1 – Zvolený RC model

1.1.4.2 Motor

Výše zmíněný obsahuje dva stejnosměrné motory, kde každý otáčí jednou stranou kol, jedná se tedy o pohon 4x4. Stejnosměrné motory umožní jak plynulý chod vozítka, ale také plynulé zatáčení, které se běžně řeší pomocí jednoho krokového motoru na předních kolech

1.1.4.3 Napájení

Byli zvoleny baterie typu NiCd díky své velice nízké ceně. Bohužel její technické parametry budou omezovat možnosti a funkce vozítka.

1.2 Seznámení

1.2.1 Metody bezdrátového řízení

Bezdrátovou komunikaci budeme realizovat pomocí mobilního telefonu, jsme tedy omezeni na určité technologie. Pomineme-li infraport a NFC, máme na výběr níže uvedené technologie.

1.2.1.1 Wi-Fi

Standart IEEE 802.11, je velice oblíbeným komunikačním protokolem, který je pro bezdrátové řízení u dražších a sofistikovanějších zařízení již nepsanou normou.

[+] Výhody

- Velký dosah až 100 metrů

[–] Nevýhody

- Vysoká spotřeba
- Cena Wi-Fi modulu
- Není v každém mobilním zařízení
- Nutnost vytvoření přístupového bodu (hotspotu)

1.2.1.2 Bluetooth

Standart pro bezdrátovou komunikaci, který dnes sice upadá, ale i tak je častou volbou u levnějších zařízení, díky své nízké ceně a spotřebě. Vzhledem k rozšířenosti a pomalému nástupu verze eliminující všechny nedostatky (4.0), bereme v úvahu pouze Bluetooth 2.0.

[+] Výhody

- Nízká spotřeba
- Cena Bluetooth modulu
- Nachází se v drtivé většině mobilních zařízení

[–] Nevýhody

- Malý dosah až 10 metrů

1.2.2 Programování obecně

Pro tento projekt budou zapotřebí celkem tři programovací jazyky, každý pro něco jiného.

1.2.2.1 Assembler

Tedy jazyk symbolických adres (JSA), někdy také špatně nazýván Assembler (překladač jazyka JSA), je jazyk nízkoúrovňový, tedy jeho instrukce jsou již posledními obaly pro čistý strojový kód. Využívá se pro programování hardwaru, protože díky němu máme přístup přímo ke stavebním kamenům každého prvku, které můžeme od základu předělávat.

V našem případě využiji jazyk JSA, abych změnil fixně danou frekvenci PWM signálu, který má řídící čip daný od výroby, díky čemuž se mi podaří zvýšit plynulost otáčení motorů.

1.2.2.2 Wiring (C++)

Wiring je velice komplexní knihovnou, která je nádstavbou jazyka C a C++. Někdy se o ní mluví jako o samostatném programovacím jazyku Wiring.

Knihovna bude využita při programování „mozku“ autíčka, tedy mikrokontroléru. Přesněji s její pomocí naprogramujeme veškerou komunikaci a s tím spojené dešifrování dat, která se po sériové sběrnici posílají pomocí znaků ASCII tabulky, dále pro ovládání motorů, distribuci energie, správu indikačních diod a tak dále…

1.2.2.3 Java

Programovací jazyk Java je jeden z nejpoužívanějších jazyků světa, avšak se jedná o jazyk, který nešetří pamětí ani prostorem, není bezpečný ani neumožňuje přímou práci s pamětí, ale jedná se o jazyk jednoduchý, přenositelný a hlavně spustitelném na téměř každém zařízení. To z něj dělá nejlepšího adepta na programovací jazyk grafického rozhraní telefonu.

Za pomocí jazyka Java naprogramujeme grafické prostředí pro operační systém Android, které bude pouze odesílat mikroprocesoru informace o tom, co má jak dělat.

2 Hardware

2.1 Úvod do problematiky

Prvním krokem je stabilizace napětí na hodnotu, kterou vyžadují součástky, tedy na optimálních 5 V pro logiku TTL. Dále je nutné vybrat a zapojit mikrokontrolér, který bude řídit všechny operace. Řízení motorů není možné přímo, je tedy nutné vyřešit jejich spínání. Nakonec zapojení komunikačního modulu a součástky pro odrušení motorů.

2.2 Výběr mikrokontroléru

Zvolený mikrokontrolér je ATmega328P. Jedná se o typického zástupce třídy ATmega, který je výhodný díky rozumnému počtu pinu a přívětivé ceně, navíc se jedná o standartní čip pro vývojovou desku Arduino Uno, která je ze všech nejrozšířenější, pro je velice jednoduché získat vhodný programátor.

2.3 Vývoj

Jako stabilizační obvod byl zvolen stejnosměrný usměrňovač LM7805, který za nízkou cenu dokáže pracovat ve vysokých teplotách, tedy s vysokými proudy, a navíc dokáže zpracovávat i vysoké množství napětí. V případě výměny baterie za silnější, zato volba ochrání desku a veškerou elektroniku nebezpečně vysokým napětím. Lze tedy použít i všudypřítomné notebookové baterie.

Zapojení mikrokontroléru sebou přináší i nutnost obstarání periférií, bez kterých nebude čip fungovat. Jednou z nich je odpor na reset, který slouží k ochraně před nežádoucím resetem napájení, to je nutné vždy při programování a podobných operacích. Dále je vyžadovaná krystal, který určí rychlost mikrokontroléru. Tím se stal 16 MHz krystal, který je standardem pro tento typ jednočipového obvodu.

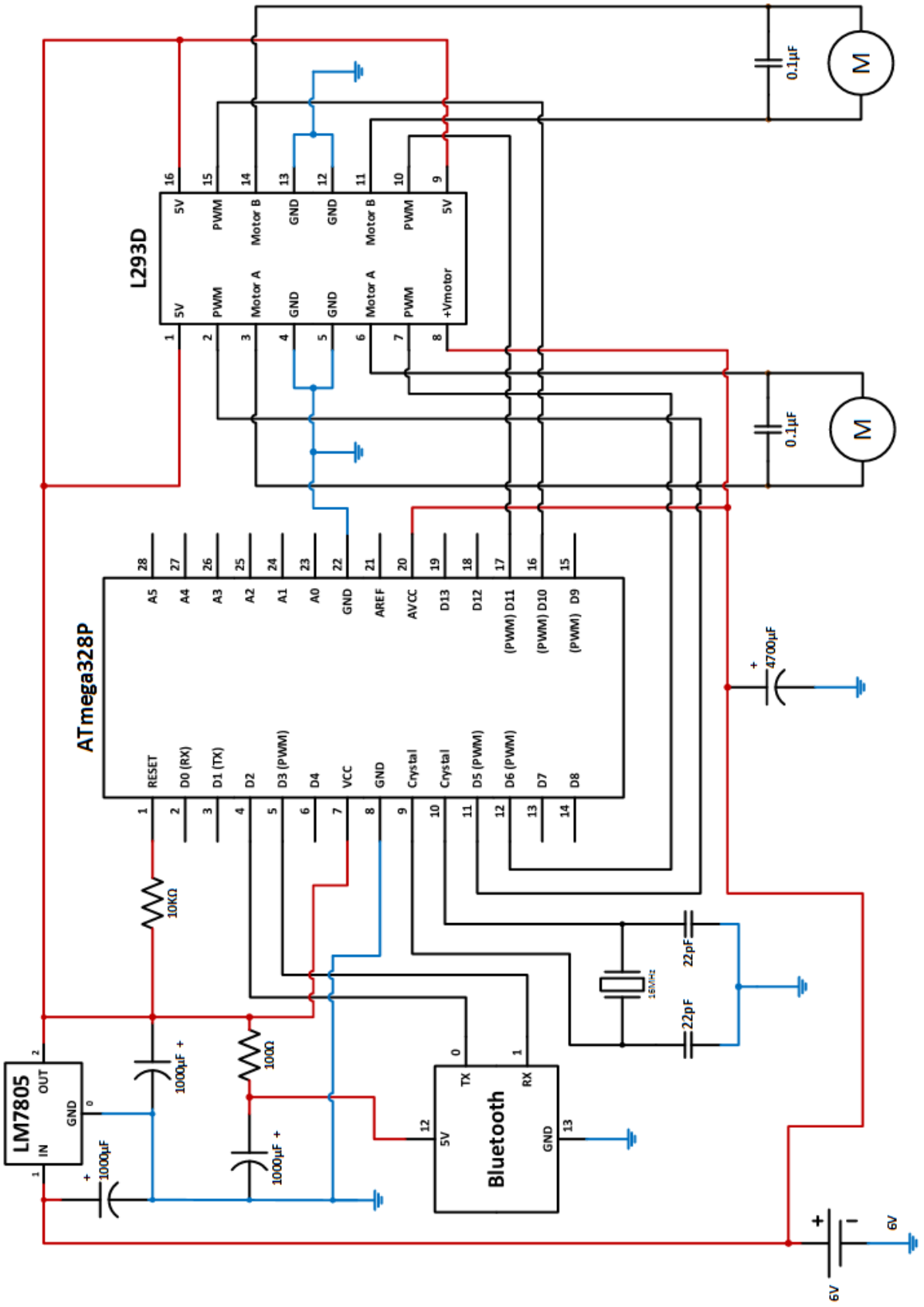
Jak již bylo zmíněno, motory nelze řídit přímo a je tedy nutné využít jiný způsob. Nejzákladnější řešení je použití tranzistorů, ke kterým je ale nutné přidat menší množství součástek. Další metodou je H-mustek, který je zjednodušením předchozí metody. Pro tento projekt byl použit obvod L293D, takzvaný „Driver“, který je zjednodušením všech přechozích metod.

Použitý Bluetooth modul, ke své správné funkci vyžaduje napětí 3,3 V. Nicméně v reálném zapojení je připojen na přímých 5V, protože to nijak výrazně neohrožuje běh zařízení ani jeho správnou funkčnost.

Posledním krokem je správné propojení a zprovoznění. Při testování vozítka bylo zjištěno, že nárazové proudy motorů, způsobené rychlými změnami otáček, mají příliš velký odběr na takto slabou baterii. Při velkém odběru proudu klesá i napětí, které bylo nedostatečné pro komunikaci Bluetooh, komunikace se v tomto případě přerušila. Tento problém byl odstraněn zapojením většího množství velkokapacitních kondenzátorů, které po zapnutí nahromadí dostatečné množství energie, která je při velkých odběrech využita a nedochází tedy k poklesu napětí na baterii.

Na následující stránce [Obrázek 2], je možné vidět kompletní popisované zapojení.

2.4 Zapojení

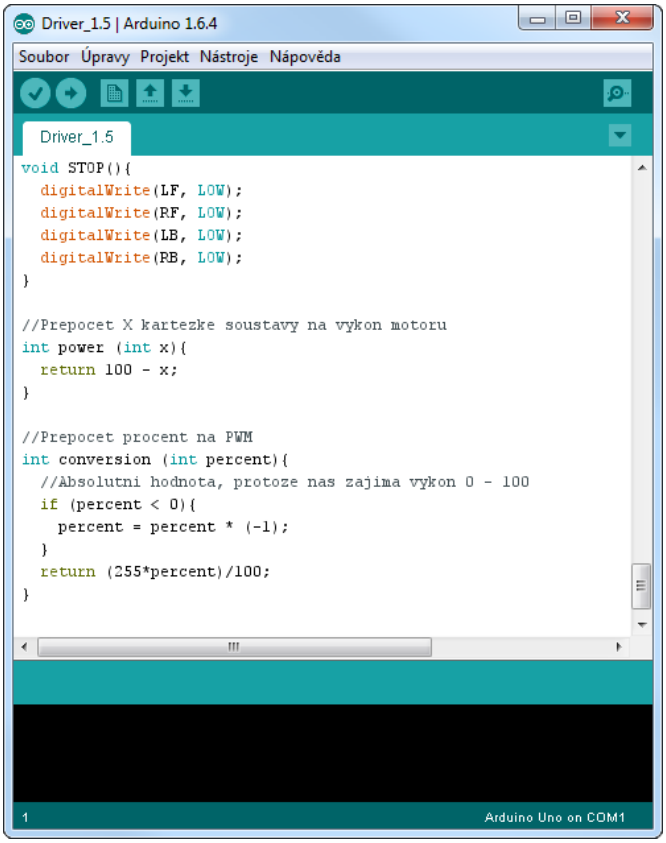


Obrázek 2 – Zapojení desky plošných spojů

3 Firmware

3.1 Úvod do problematiky

Firmware neboli software sloužící pro řízení vestavěného systému (jednoúčelového počítače), častěji definován jako „Kombinace hardwarového zařízení a počítačových instrukcí a dat“, bude mít v tomto projektu úlohu řízení hlavního mikrokontroléru [µC]. Vzhledem k tomu, že se bude nacházet v centru denní, bude to alfa a omega celého projektu.



Obrázek 3 – Vývojové prostředí Arduino

3.1.1 Komunikace

Jedna z hlavních věcí je komunikace s ovladačem. Musíme tedy správně pracovat s daty, která nám budou přicházet z Bluetooth modu. Data budou ve tvaru ASCII znaků, které je tedy nutné je správně dešifrovat a přiřadit jim správné vlastnosti. Zvlášť se bude odesílat, o jaké hodnoty se jedná, zvlášť jestli hodnota nabývá kladných nebo záporných os. Více informací ohledně komunikačního rámce se nachází ve 4. kapitole.

3.1.2 Pohyb

Z dešifrovaných dat musíme zjistit, zda musíme zjistit, co znamenají a jak je správně předal motorům. Musíme tedy řešit také převod z procent [0 - 100] na signál PWM [0 - 255].

3.2 Vývojové prostředí

Jako vývojové prostředí jsem si zvolil open-source software Arduino IDE (Integrated Development Environment) [Obrázek 3]. Jedná se o vývojové prostředí pro vývoj stejnojmenného vývojového kitu Arduino, který stejně jako já používá procesory Atmega, navíc plně podporuje knihovnu Wiring, což je značnou výhodou oproti jiným odborným softwarům.

3.3 Programování

Pro začátek je nutné, použít již před vytvořenou knihovnu SoftwareSerial.h, která nám usnadní komunikaci mezi Mikrokontrolérem a Bluetooh modulem, kterou již nemusíme řešit. Díky tomu se můžeme plně soustředit na zpracovávání přijatých dat z telefonu. Knihovna SoftwareSerial také poslouží pro testování aplikaci přes sériovou sběrnici USB. Komunikační piny pro Bluetooth komunikaci, je nutné nastavit na jiné pozice než je defaultní RX, TX, která jsou již rezervována pro zmiňovanou komunikaci přes USB. Jejich zapojením také pro Bluetooth by bylo znemožněno programování mikrokontroléru při jiné aktivní komunikaci.

Nejdříve musíme nastavit správné piny pro spínání motorů. Všechny proměnné nastavíme jako výstupní, protože neočekáváme žádné příchozí hodnoty. Každý motor bude mít vždy dva piny. První udává, jakou rychlostí se má motor točit [0 (min) – 255 (max)]. Druhý pin bude vždy uzemněný, tedy se uzavře okruh. Jejich prohozením (programově) bude umožněna změna otáčení druhým směrem. Je pouze důležité zvolit všechny piny s funkcí PWM, aby byla možná regulace rychlosti.

Bluetooth modul má speciální paměť FIFO (First In, First Out), kam se ukládají všechna příchozí data, obsah paměti zjistíme příkazem bluetooth.available(), kde návratová hodnota je počet uchovávaných hodnot typu Char. Tedy pokud bluetooth.available() > 0, máme příchozí data, která je nutné zpracovat. Aby bylo možné rozložená data (na char) znovu spojit do jednoho čísla, je nutné udržet vždy stejný rámec dat. Je potřeba pamatovat, že Bluetooh modul data odesílá jako znaky Char, tedy jsou kódovány dle tabulky znaků ASCII. V našem případě se vždy po odeslání znaménka [+ / -], zašlou čtyři znaky ve tvaru [x . x x], tedy desetinné číslo, s přesností na dvě desetinná místa. Pokud data načteme do proměnné typu String a použijeme její metodu .toFloat(), získáme číselnou hodnotu. Hodnota je ve tvaru desetinném, tedy je nutné jí vynásobit číslem 100. Dále následuje ověření, zda znaménko nebylo záporné. Pokud ano, je výslednou, nyní celočíselnou, hodnotu nutné vynásobit číslem -1. Předposledním krokem je převod do hodnot, které budeme předávat motorům, proto je nutné si odvodit vzoreček na přepočet procent do PWM.

$$PWM = \frac{255 * Hodnota}{100}$$

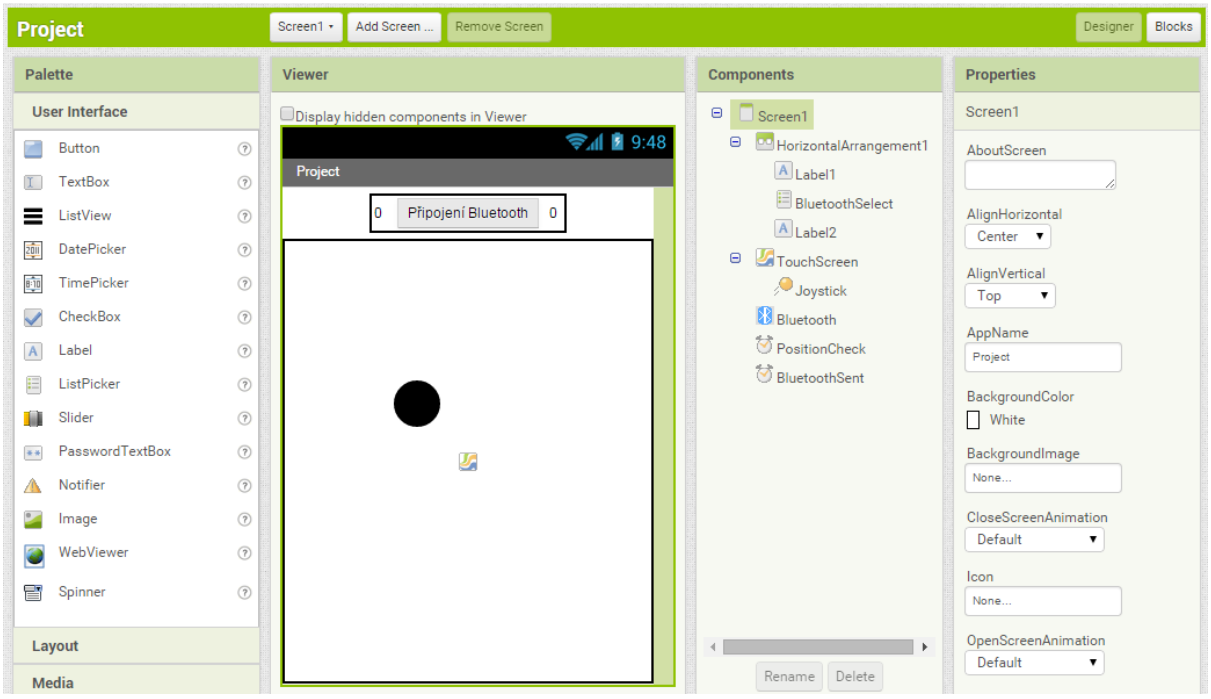
Konečná fáze je přiřadit datům správný pohyb se správnou rychlostí. Tento úkol je velice primitivní a vyžaduje pouze základní logiku pro vyhodnocení v jakém kvadrantu Kartézské soustavy Joysticku [Obrázek 5 – Koncept ovládání vozítka pomocí Joysticku], se data nachází. Toto vyhodnocení se děje zatím ještě s původními hodnotami v procentech. Nakonec je pouze nutné pouze dle přednastaveného pohybu aktivovat příslušné piny se vstupní hodnotou PWM, spočítaného dle odvozeného vzorce.

4 Software

4.1 Úvod do problematiky

Velkou výzvou je v této části vytvoření softwaru pro bezdrátové ovládání vozítka, na platformu Android. Hlavním problémem je plně digitální Joystick, který byl zvolen jako hlavní řídící mechanismus pro udávání rychlosti pohybu vpřed, vzad, stání na místě, jízdě do všech směrů a otáčení na místě a to za pomocí pouze dvou motorů řídících 4 kola. Je tedy nutné vymyslet graficky přijatelné prostředí, pro pohodlné odesílání dat. Také je nutné vyřešit, v jaké formě budeme data odesílat, vzhledem k tomu, že Bluetooh komunikuje pomocí znaků (1 bit = 1 znak), je nutné také vytvořit jistý komunikační rámec, který bude standardem komunikace mezi mobilním telefonem a řídícím mikroprocesorem.

4.2 Vývojové prostředí



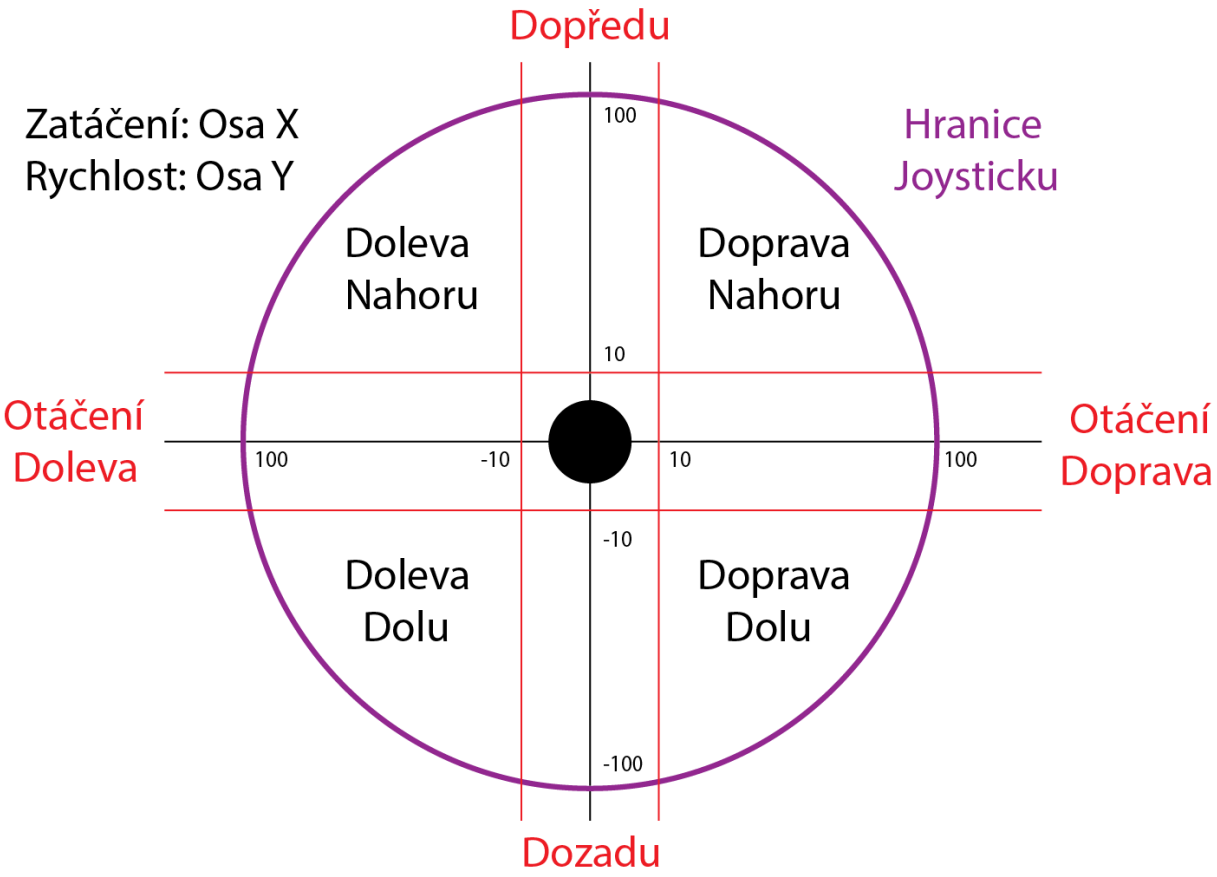
Obrázek 4 – Vývojové prostředí APP Inventor

Jako vývojové prostředí byl vybrán APP Inventor od MIT [Obrázek 4], který je ideálním řešením pro projekty s nižší úrovní algoritmizace. Řeší za programátora komunikaci se samotným telefonem. Programátor tedy nemusí řešit, jak funguje na platformě Android dotyková vrstva, Bluetooth a podobné. Díky tomu se můžeme věnovat pouze samotnému programování joysticku a posílání dat.

4.3 Programování

Při vývoji se vyskytl problém, a to ten, že se o obdobné řešení zajímalo větší množství programátorů, nicméně z důvodu neúspěchu se většina omezila pouze na ovládání pomocí směrových šipek. Problém po několika týdnech vývoje vyřešen pomocí matematických funkcí a vnořených funkcí androidu pro vykreslování kružnic. Pomocí metody DrawCircle se postupně podařilo vytvořit kruhovou plochu pro výsledný Joystick a následně dokončit také mapování pohybu prstu po displeji a navíc k této pozici přichytávat samotný Joystick, který je omezen kruhovou výsečí, tvořící kýžený efekt ovladače.

Na obrázku níže [Obrázek 5] si ukážeme, jak je koncipován sběr dat z Joysticku. Představme si osu X (zatáčení) a Y (rychlost), tato kartézská soustava souřadná o dvou souřadnicích, je ohraničená kružnicí o poloměru 100. Rychlosti ležící na ose Y byli přirazeny hodnoty -100 (zpět) do 100 (dopředu), stejně tak zatáčení na osa X je -100 (doleva) až 100 (doprava). Pokud chceme vozítko otáčet na místě, je nutné se Joystickem nacházet na rychlosti nula, kde hodnota zatáčení, znamená rychlost otáčení.



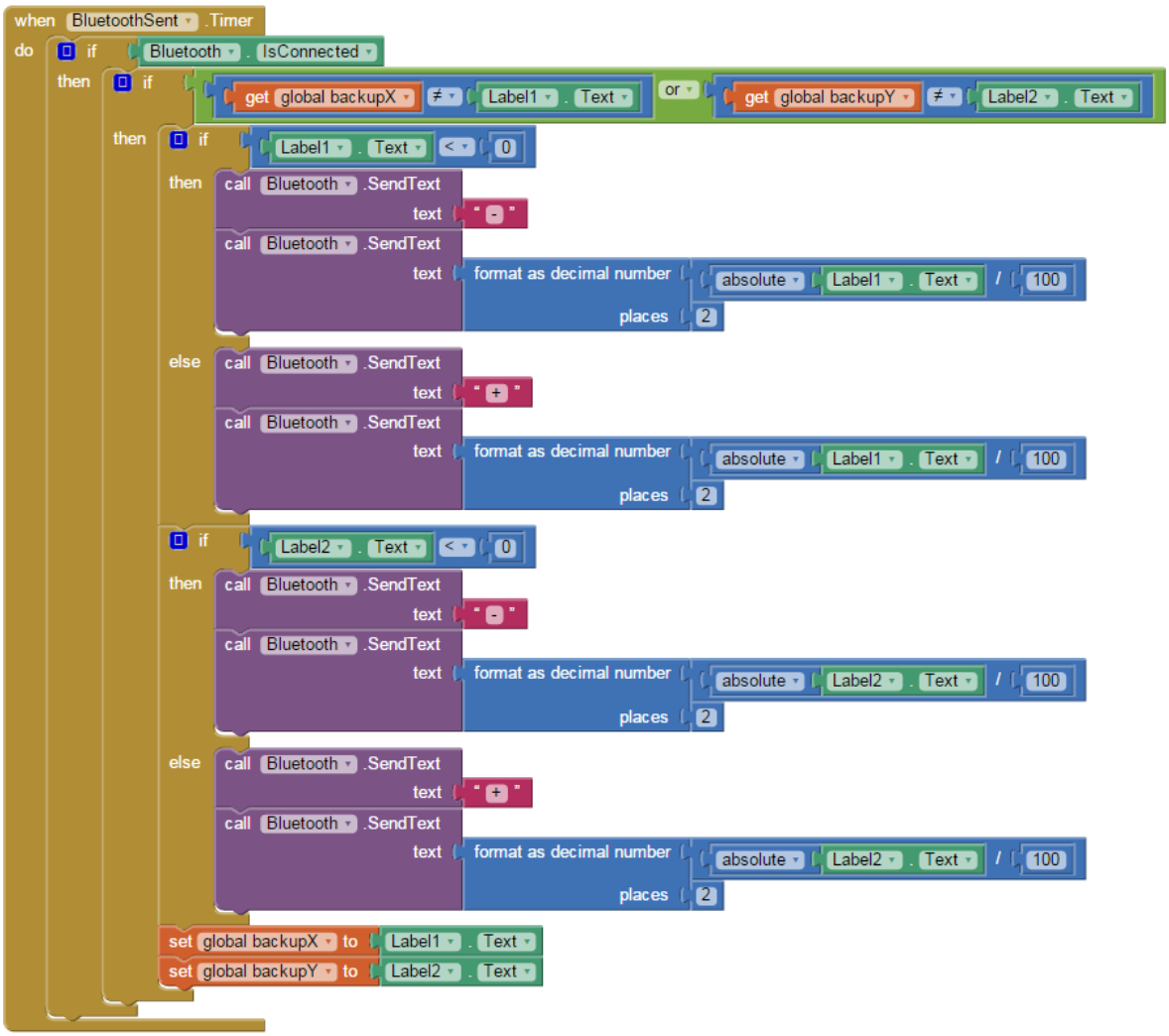
Obrázek 5 – Koncept ovládání vozítka pomocí Joysticku

V průběhu vývoje se vyskytl problém s komunikací, kdy bylo nutné data odesílat ve stejné délce, přičemž hodnoty +0, -55, +100, +3, … nesplňovali. Pro tento problém byl vytvořen vlastní komunikační rámec [Obrázek 6], ve kterém je první komunikační bit znaménko, tím se potvrdí odesílání dat, zbytek je hodnoty na ose, vydělaná číslem 100. Tedy získáváme vždy desetinné číslo s přesností na 2 desetinná místa. Postup se opakuje také pro druhou souřadnici.

+	0	.	5	4

Obrázek 6 – Ukázka přenášených dat

Aby se zamezilo nepřetržité komunikaci, která by zbytečně zahlcovala Bluetooh modul daty. Byl vytvořen algoritmus, který vždy při změně polohy Joysticku, zašle nová data. Pokud je Joystick puštěn, přesune se do bodu [0;0], je tedy vyslán signál o zastavení vozítka. Na obrázku níže [Obrázek 7], vidíme ukázku části algoritmu, který vyhodnocuje a odesílá data.



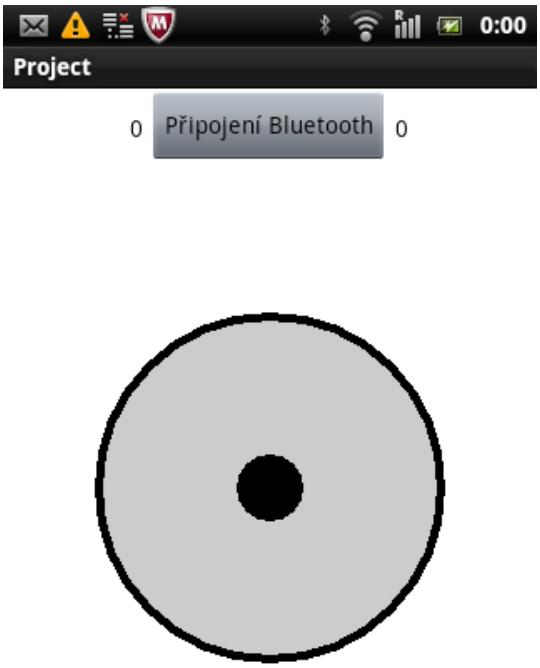
Obrázek 7 – Ukázka algoritmu vyhodnocení a zasílání dat

5 Závěr

5.1 Výsledky

Hlavním úkolem této práce bylo předělání analogového RC modelu na plně digitální vozítko.

Velkým úspěchem je, že se podařilo dokončit všechny body semestrálního projektu. Elektronika bylo kompletně dokončena, včetně regulace napájení, řídících integrovaných obvodů a také driverů pro motory. Podařilo se nejenom zprovoznit software, zprostředkovávající Bluetooth komunikaci mezi samotným modulem na DPS a modulem Bluetooth s vlastním grafickým prostředím [Obrázek 8] na platformě Android, ale také vytvořit vlastní komunikační rámec, pomocí kterého jsou data přenášena. Řídící firmware mikroprocesoru, byl taktéž kompletně dokončen.



Obrázek 8 – Ukázka grafického rozhraní

5.2 Plány do budoucna

S ohledem do budoucna je v plánu vytvořit autonomní režim, přidáním většího množství senzorů, zahrnující ultrazvukové senzory, GPS modul a gyroskopy. Při testování bylo zjištěno, že pokud je mobilní aplikace nakloněna na šířku, její rozložení se změní a grafika nesouhlasí s opravdovými křivkami, které ohraničují plochu Joysticku, tato chyba bude v budoucnu opravena a stejně tak je v plánu modernizace grafické rozhraní ovladače. Nakonec je v plánu, vytvoření plnohodnotné desky plošných spojů (DPS), namísto nynějších nepájivých DPS.

6 Literatura

[1] MARGOLIS, Michael. Make an Arduino-controlled robot. 1st ed. Sebastopol, Calif.: O'Reilly, c2013, xvi, 238 p. Make. ISBN 14-493-4437-2.

[2] SCHILDT, Herbert. Java 7: výukový kurz. 1. vyd. Brno: Computer Press, 2012, 664 s. ISBN 978-80-251-3748-2.