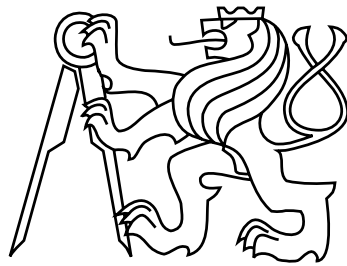


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA ELEKTROTECHNICKÁ
KATEDRA KYBERNETIKY



BAKALÁŘSKÁ PRÁCE

Teleoperace mobilního robotu

Autor: Martin Jarolímek

Vedoucí práce: Ing. Jan Chudoba

Praha, 2015

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: Martin J a r o l í m e k

Studijní program: Kybernetika a robotika (bakalářský)

Obor: Robotika

Název tématu: Teleoperace mobilního robotu

Pokyny pro vypracování:

1. Prostudujte možnosti dálkového ovládání a řízení s přenosem obrazu vhodné pro malé mobilní roboty.
2. Zkonstruuje prototyp mobilního robotu s vhodnou senzorickou výbavou a řídicím počítačem pro účely demonstrace.
3. Vyberte vhodnou platformu pro realizaci uživatelského ovládacího rozhraní s možnostmi vizualizace senzorických dat a okolního prostředí robotu.
4. Navrhněte a implementujte teleoperátorské uživatelské rozhraní pro zvolenou platformu a komunikaci s robotem.
5. Vyhodnoťte funkci implementovaného řešení.

Seznam odborné literatury:

- [1] Nehmzow, U.: Mobile Robotics, A Practical Introduction. Springer, London, 2003.
[2] Siciliano, B., et al.: Robotics, Modelling, Planning and Control. Springer, London 2010.

Vedoucí bakalářské práce: Ing. Jan Chudoba

Platnost zadání: do konce letního semestru 2014/2015

L.S.

doc. Dr. Ing. Jan Kybic
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 10. 1. 2014

Název práce: Teleoperace mobilního robotu

Autor: Martin Jarolímek

Katedra (ústav): Katedra kybernetiky

Vedoucí bakalářské práce: Ing. Jan Chudoba

Abstrakt V bakalářské práci se zabývám možnostmi dálkového ovládní mobilního robotu a sestavením prototypu mobilního robotu. Robot je zkonstruován jako malý průzkumný pásový robot. Základem je počítač Raspberry Pi a deska Arduino. Pro průzkum okolí robot disponuje kamerou a senzory, především dálkoměry. Robot má též klakson a světla. Jako dálkový ovladač je využita spotřební elektronika. Ovládní je realizováno jako aplikace pro operační systém Android. Aplikace obsahuje ovládací i zobrazovací prvky. Robotem zachytávané video je odesíláno do aplikace jako stream. Výseč před robotem je prozkoumávána dálkoměry, aplikace zobrazuje umístění případné překážky. Bezdrátová komunikace mezi ovladačem a robotem je zajišťována technologií Wi-Fi.

Klíčová slova: mobilní robot, teleoperace, dálkové ovládní, Android

Title: Mobile robot teleoperation

Author: Martin Jarolímek

Department: The Department of Cybernetics

Supervisor: Ing. Jan Chudoba

Abstract In the bachelor thesis I deal with the possibilities of mobile robot remote control, as well as constructing the prototype of a mobile robot. The robot is designed as a small tracked robot for exploration. The basis is a computer Raspberry Pi and an Arduino board. To explore the surrounding area robot has a camera and sensors, especially rangefinders. The robot also has a car horn and lights. A consumer electronics is used for remote controller. Controlling is implemented as an application for the Android operating system. The application contains control and display elements. Captured video is sending to the application as a stream. Sector in front of the robot is explored by rangefinders, application shows the location of any obstacles. Wireless communication between the controller and the robot is provided by Wi-Fi technology.

Keywords: mobile robot, teleoperation, remote control, Android

Prohlášení autora práce

Prohlašuji, že jsem předloženou práci vykonal samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne

.....
Podpis autora práce

Poděkování

Rád bych poděkoval Ing. Janu Chudobovi za cenné rady, věcné připomínky a vstřícnost při konzultacích a vypracování bakalářské práce. Dále děkuji svým přátelům za pomoc při testování robotu.

OBSAH

1 Úvod	1
1.1 Koncept	2
1.2 Použité nástroje	2
2 Technologie bezdrátového přenosu dat	4
2.1 Bezdrátové technologie	4
2.1.1 Wi-Fi	4
2.1.2 WiMAX	5
2.1.3 Bluetooth	5
2.1.4 ZigBee	6
2.1.5 Mobilní sítě	6
2.2 Porovnání	6
3 Konstrukce robotu	8
3.1 Motory	8
3.1.1 Řízení motorů	9
3.1.2 Testování motorů	12
3.2 Enkodéry	13
3.3 Napájení	14
3.4 Arduino	16
3.4.1 Řízení motorů	16
3.4.2 Měření napětí	16
3.4.3 Měření otáček	16
3.4.4 Regulace otáček	17
3.4.5 Komunikace	17
3.4.6 Servomotor	17
3.4.7 Struktura programu	17
3.5 Regulátor	18
3.5.1 Identifikace systému	18
3.5.2 Návrh konstant	19
3.5.3 Simulace	22

3.6	Raspberry Pi	22
3.6.1	Stream videa	24
3.6.2	Program <code>host.java</code>	24
3.7	Senzory	25
3.7.1	Dálkoměry	25
3.7.2	Voltmetr	27
3.7.3	Teploměr	28
3.7.4	Klakson	28
3.7.5	Světla	29
4	Platforma uživatelského ovládacího rozhraní	30
4.1	Popis platforem	30
4.2	Porovnání	31
4.3	Android	31
5	Teleoperátorské uživatelské ovládací rozhraní	32
5.1	Komunikační protokol	32
5.2	Činnost aplikace	33
5.2.1	Připojení	33
5.3	Popis komponent	34
5.3.1	Popis tříd	34
5.3.2	Paměť	36
5.3.3	Broadcast Receiver	36
5.3.4	Oprávnění	36
5.4	Ovládání	37
5.4.1	Grafické prvky	37
5.5	Administrátorský mód	38
6	Závěr	40
	Literatura	42
	Přílohy	I
A	Obsah přiloženého CD	II

ÚVOD

Jako téma své bakalářské práce jsem si vybral Teleoperace mobilního robotu. Součástí práce je nejen teoretické zpracování tématu, ale i samotné sestavení prototypu mobilního robotu. Můj zájem o zařízení na dálkové ovládání je dlouhodobý, proto jsem se rozhodl v bakalářské práci prozkoumat možnosti ovládání takových zařízení.

Pod pojmem zařízení na dálkové ovládání si většina lidí vybaví RC model. Princip mého robotu z RC modelu vychází, posouvá ho ale dále. RC model je jednoduché zařízení, které nenabízí žádné doplňkové funkce, nenesení žádné senzory. Můj robot by měl mít senzory pro zkoumání okolí. RC modely se tradičně ovládají mechanickým ovladačem. Ten je osazen množstvím mechanických přepínačů a joysticků. Při tomto způsobu ovládání má uživatel se zařízením pouze vizuální kontakt. Další omezení přichází s požadavkem na přenos videa – takový ovladač většinou nedisponuje monitorem či jiným zobrazovacím zařízením.

V dnešní době existuje množství spotřební elektroniky – mobilní telefony, tablety a počítače všeho druhu. Tato elektronika je ve většině případů dostatečně výkonná a univerzální, zároveň splňuje veškeré hardwarové požadavky, jaké jsou kladeny na dálkové ovladače. Jde hlavně o bezdrátovou komunikaci. Ve své práci bych rád prozkoumal jiné možnosti ovládání dálkových zařízení a z výše uvedených důvodů jsem se zaměřil na využití spotřební elektroniky pro účely ovládání takovýchto zařízení. Používání spotřební elektroniky se vyznačuje jednoduchostí a intuitivností a tento princip bych rád uplatnil i při ovládání dálkových zařízení. Běžná elektronika také ve své podstatě disponuje zobrazovacím zařízením, které rozšiřuje možnosti interakce.

Cílem práce je také ukázat jak sestavit malý průzkumný robot, který bude ovládán bezdrátově dálkovým ovladačem. Cílem není vyvinout autonomní robot, ale pouze zkonstruovat univerzální prototyp robotu, který bude schopen vykonávat teleoperace. Termín teleoperace lze přeložit jako „dálkové ovládání“¹. Znamená to, že ovládané zařízení musí být schopné plnit úkoly (příkazy), které dostane od operátora. Dálkově ovládané zařízení může být schopno i částečné autonomie, např. vyhnout se překážce. Zároveň musí být robot schopen vnímat své okolí – to by měla

¹anglicky „remote control“

zajišťovat především kamera a dálkoměry. Veškerá data, která robot při své činnosti získá, musí být adekvátně vizualizována operátorovi na dálkovém ovladači. Velký důraz by měl být kladen na jednoduchost a univerzálnost použitých komponent.

Robot je určen pro průzkum území, oblastí, nebo objektů, kde je fyzická přítomnost člověka riskantní či nemožná. Robot by měl být malý – měl by se vejít i tam, kam se člověk nedostane.

1.1 Koncept

Robot je zkonstruován jako pásové vozidlo. Každý pás je ovládán jedním motorem. Robot je určen do terénu, musí být schopen překonání malých překážek typu práh dveří. Fyzické velikosti měly být co možná nejmenší, robot má půdorysné rozměry cca 15 cm × 20 cm.

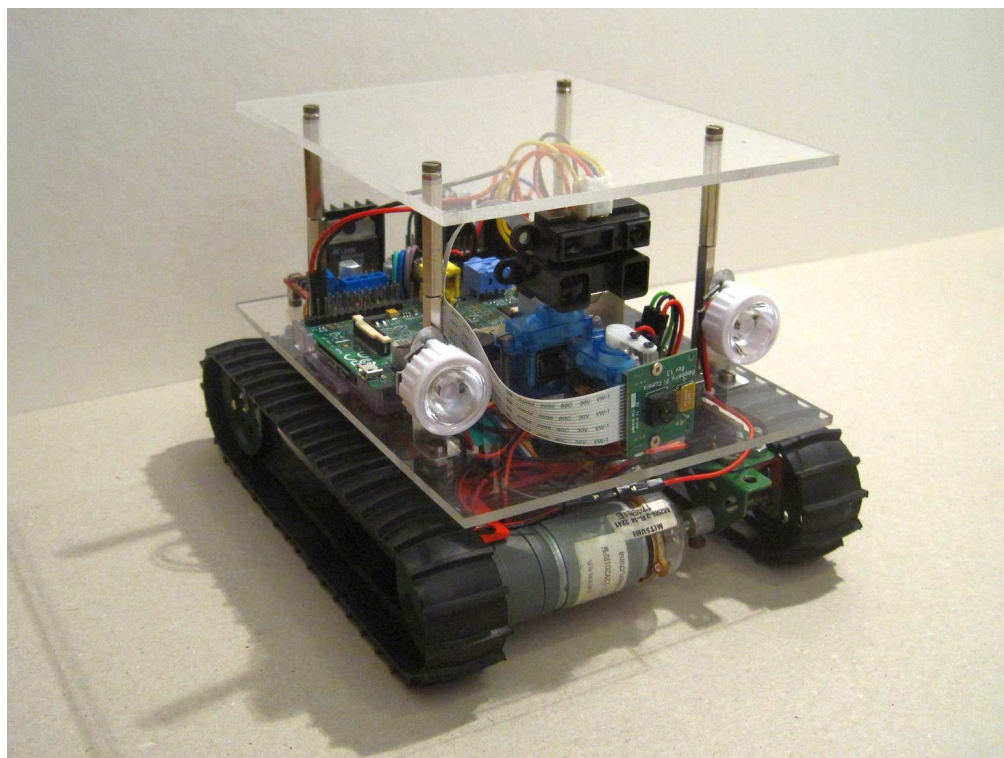
Pro konstrukci robotu je použit velmi populární počítač Raspberry Pi. Jedná se o Linuxové zařízení s velmi nízkou spotřebou, možností připojení kamerového modulu a USB porty. Počítač Raspberry Pi obstarává komunikaci přes Wi-Fi, snímání videa a připojení desky Arduino Micro. Ta je další součástí robotu a stará se o řízení periferií, ovládání motorů, světel, servomotorů a senzorů. Řízení rychlosti a směru motorů obstarává duální H-můstek. Rychlost motorů je řízena regulátorem, který zajišťuje požadovanou rychlost bez ohledu na napájecí napětí nebo zatížení kol. V podvozku jsou zabudovány Li-Ion akumulátory s vysokou kapacitou. Pro napájení elektroniky je potřeba napětí 5 V – to zajišťuje spínaný regulátor napětí.

Ovladačem je tablet či mobilní telefon s operačním systémem Android, který je k robotu připojen přes Wi-Fi. Robot je ovládán aplikací, která zároveň zobrazuje senzorická data a video. Video zachytává kamera připojená k počítači Raspberry Pi. Video je streamované v reálném čase technikou MjpegStream.

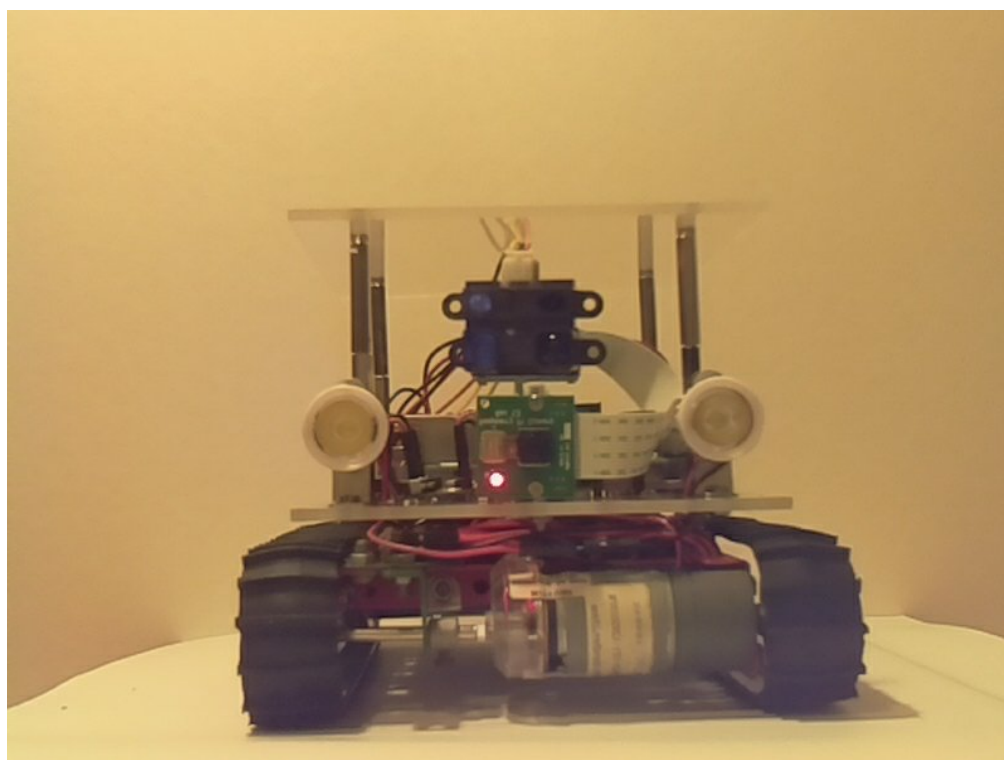
Robot má své uživatelské jméno a heslo. Tyto údaje slouží k identifikaci robotu a ověření oprávněnosti přístupu. Uživatelské jméno je zároveň název Wi-Fi sítě, kterou robot vytváří. Heslo robotu je heslo Wi-Fi sítě. Tyto údaje se uplatní při provozování více podobných robotů.

1.2 Použité nástroje

Při práci jsem používal několik softwarových nástrojů – pro programování desky Arduino oficiální vývojové prostředí Arduino IDE, aplikaci pro systém Android jsem programoval ve vývojovém prostředí Android Studio. Dále jsem používal výpočetní nástroj MATLAB. V něm byly prováděny různé výpočty, návrh regulátoru, odvozování převodních charakteristik a rovněž vytvářeny všechny grafy.



Obrázek 1.1: Obrázek robotu



Obrázek 1.2: Ukázka kvality snímaného videa

TECHNOLOGIE BEZDRÁTOVÉHO PŘENOSU DAT

Dálkové ovládání robotu musí být z důvodu mobility bezdrátové – musí tedy využívat rádiové frekvence, a to pouze ty povolené Českým telekomunikačním úřadem, tzv. bezlicenční pásma ISM (v ČR viz [1]). Je nutné použít jen tyto frekvence, avšak i pro ně platí pravidla, která je třeba dodržet.

V zásadě existují dvě možnosti, jak k dálkovému ovládání přistoupit. Jednou z možností je použití klasického ovládacího panelu s přepínači a joysticky, který se používá jako vysílač pro RC modely v kombinaci se samostatným zobrazovacím zařízením. Takováto kombinace je však obtížná na synchronizaci vzájemné komunikace dvou samostatných zařízení s dalším zařízením. Výhodou by bylo použití hotového zařízení.

Druhou možností je vytvoření zcela nového dálkového ovladače, což je v souladu s cílem práce vyvinout alternativní dálkové ovládání robotu. Jak bylo zmíněno v úvodu, jako dálkový ovladač je použita běžná elektronika a k samotnému ovládání slouží aplikace. Ovládání i zobrazování je soustředěné do jednoho zařízení. Bylo však nutné vybrat vhodnou technologii pro přenos dat mezi ovladačem a robotem.

2.1 Bezdrátové technologie

Při výběru technologie bezdrátového přenosu dat bylo třeba mít na paměti požadavek na přenos obrazu nebo videa, což vyžaduje relativně velké přenosové rychlosti. V níže provedeném porovnání není zmíněna možnost ovládání rádiovým signálem, neboť ho de facto používají všechny bezdrátové technologie. Použití obecného rádiového signálu nebylo uvažováno, protože se nejedná o univerzální řešení. Znamenalo by nutnost implementace celé řady protokolů podle ISO/OSI modelu.

2.1.1 Wi-Fi

Technologie Wi-Fi je definována standardem IEEE 802.11 [2] a definuje několik standardů, které využívají různá pásma, rychlosti i modulace. Wi-Fi pracuje na linkové

vrstvě ISO/OSI modelu a nestará se o typ protokolu, který přenáší. Wi-Fi se vyvíjí již řadu let. Za tuto dobu vzniklo již několik standardů. Nejpoužívanější jsou 802.11g, 802.11n a 802.11a. Standardy 802.11g a 802.11n využívají ISM frekvenční pásmo 2,4 GHz, zatímco standard 802.11a využívá ISM pásmo 5 GHz. Maximální přenosová rychlost standardů 802.11g a 802.11a je 54 Mbit/s, 802.11n může dosáhnout rychlosti až 600 Mbit/s. Toho je docíleno použitím více vysílacích a přijímacích antén a využitím širšího pásma. Reálné přenosové rychlosti Wi-Fi jsou zhruba poloviční oproti uváděným rychlostem, záleží však na mnoha parametrech. Nevýhodou je omezený dosah daný maximálním vyzářeným výkonem (100 mW), což odpovídá dosahu v řádu desítek až stovek metrů v závislosti na terénu. Další nevýhodou jsou překrývající se kanály a silné zarušení pásma 2,4 GHz, protože stejné pásmo využívají i jiné technologie (Bluetooth, mikrovlnná trouba). Hlavní výhodou této technologie spočívá v jejím rozšíření – Wi-Fi dnes podporuje většina mobilních telefonů, počítačů a tabletů. Velkou výhodou jsou i dosahované rychlosti. Wi-Fi nabízí dostatečné možnosti zabezpečení. Zařízení pro Wi-Fi síť (vysílače, přijímače, routery) jsou relativně levná.

2.1.2 WiMAX

Technologie WiMAX je definována standardem IEEE 802.16 [3] jako nástupnická technologie Wi-Fi. Standardizace ještě není dokončena. Tato technologie není určena pro koncové zákazníky, ale pro poskytovatele jako infrastrukturní prvek. Existuje však už i standard pro mobilní WiMAX. Využívá jak ISM pásmo 2,4 GHz a 5 GHz, tak různých licenčních pásem od 2 GHz do 60 GHz. Nabízí přenosové rychlosti v řádu desítek Mbit/s a dosah v řádu desítek kilometrů. Mezi její hlavní nevýhody patří malé rozšíření a nedostupnost zařízení.

2.1.3 Bluetooth

Technologie Bluetooth je definována standardem IEEE 802.15.1 [4] jako standard pro ad-hoc síť. Využívá ISM pásmo 2,4 GHz. Bluetooth je tvořeno několika protokoly od fyzické až po aplikační vrstvu. To je velká výhoda i nevýhoda zároveň. Výhodou je jednoduchá implementace, nevýhodou nemožnost využití jakýchkoli jiných protokolů (např. IP protokolů). Další nevýhodou Bluetooth spočívá v omezené rychlosti – max 24 Mbit/s. Při maximálním povoleném vyzářeném výkonu 100 mW je dosah 100 m (v závislosti na terénu). Bluetooth je primárně určen pro ad-hoc síť, jeho zapojení do infrastruktury je náročné. Bluetooth disponuje vysokou úrovní zabezpečení, pro komunikaci dvou zařízení je nutné nejprve jejich tzv. párování. Mezi výhody této technologie patří velké rozšíření, Bluetooth je stejně jako Wi-Fi přítomno ve většině počítačů a mobilních telefonů.

2.1.4 ZigBee

Technologie ZigBee je definována standardem IEEE 802.15.4 [5] jako doplněk technologií Wi-Fi a Bluetooth. Jedná se novou technologií – existuje od roku 2004. Podobně jako Bluetooth je technologie ZigBee tvořena sadou protokolů od fyzické až po aplikační vrstvu a není možné použít jiné protokoly. Pracuje v ISM pásmech 2,4 GHz celosvětově a dalších lokálně. Mezi největší přednosti patří jednoduchost, malé nároky na hardware, velmi nízká spotřeba energie a zabezpečení sítě. Mezi nevýhody patří omezený dosah (desítky metrů) a nízké přenosové rychlosti (< 250 kbit/s), které vycházejí ze zaměření technologie na přenos malého objemu dat na malou vzdálenost. ZigBee zaměřeno hlavně pro nasazení v automatizaci nebo pro monitorovací systémy. Hardware pro ZigBee jsou mikroprocesory s Flash pamětí < 1 MB a frekvencí < 100 MHz. Zigbee umožňuje vytváření topologie sítě, díky níž je možné komunikovat i se zařízením vzdálenějším, než umožňuje dosah technologie. Standard definuje několik režimů přenosu dat – periodické procesy, nepravidelné přenosy a opakující se přenosy s požadavkem na malé zpoždění. ZigBee definuje i tři druhy zařízení: ZigBee Koordinátor je kořen sítě, ZigBee Router je směrovač a ZigBee Koncové zařízení je samotné zařízení. Každý druh zařízení má svou konkrétní funkci a má implementované jen potřebné protokoly.

2.1.5 Mobilní sítě

Mobilní sítě nejsou standardy z rodiny IEEE 802. Standardů mobilních sítí je několik a dělí se na tzv. generace. Jedná se o segment komerčních služeb, který se dramaticky vyvíjí. Následující přehled specifikací je odraz aktuálního stavu (rok 2014, Česká republika). Tyto specifikace jsou zároveň odlišné v každé zemi. V následujících letech lze očekávat zvyšování rychlostí a rozšiřování pokrytí. V současné době nejstarší, avšak stále nejpoužívanější sítí je GSM, síť 2. generace (2G). Nabízí vysokou míru pokrytí signálem a datové přenosy EDGE o rychlosti do 236 kbit/s. Novější síť 3. generace (3G) nabízí datové přenosy HSPA+ o rychlostech maximálně 21,6 Mbit/s. Nejnovější LTE síť 4. generace (4G) nabízí rychlost do 150 Mbit/s. Nevýhodou LTE je dosud malé pokrytí.¹ Uvedené hodnoty rychlostí jsou však pouze teoretická maxima, reálně dosažitelné rychlosti jsou řádově nižší. Největší výhoda mobilních sítí spočívá v širokém pokrytí signálem (kromě LTE). Pro využití mobilních sítí je třeba využít služeb mobilního operátora, který disponuje licenci na tato pásma a zároveň je správcem infrastruktury. Mezi nevýhody patří nízké přenosové rychlosti sítí 2G a 3G a vysoké provozní náklady.

2.2 Porovnání

Při výběru vhodné technologie přenosu dat jsem vybíral pouze z technologií, které jsou v dnešní době dostupné: Wi-Fi, Bluetooth, ZigBee a mobilní sítě.

¹Informace jsem čerpal na webových stránkách českých operátorů. Viz [6–8]

Technologie	Frekvenční pásma	Rychlost	Dosah
Wi-Fi (802.11g)	ISM 2,4 GHz	54 Mbit/s	50 m
Wi-Fi (802.11n)	ISM 2,4 GHz	600 Mbit/s	100 m
Wi-Fi (802.11a)	ISM 5 GHz	54 Mbit/s	50 m
WiMAX	2 – 60 GHz	~100 Mbit/s	~ 10 km
Bluetooth	ISM 2,4 GHz	24 Mbit/s	100 m
ZigBee	ISM 2,4 GHz, 868 MHz	250 kbit/s	50 m
Mobilní síť (EDGE)	900 MHz, 1800 MHz	236 kbit/s	V místě pokrytí
Mobilní síť (HSPA+)	2100 MHz	21,6 Mbit/s	V místě pokrytí
Mobilní síť (LTE)	800 MHz, 2600 MHz	150 Mbit/s	V místě pokrytí

Tabulka 2.1: Porovnání technologií včetně různých generací

Technologie WiMAX je nepoužitelná pro svou nedostupnost. Technologie měla být univerzální, a proto je pro komunikaci ovládacího panelu s robotem vhodné využít již stávajících protokolů (např. protokolů IP). Toto je problém technologií Bluetooth i ZigBee, neboť IP protokoly nepodporují. Přestože jsou technologie takto uzavřené, obsahují protokoly s podobnou funkcí. Dalším argumentem proti technologii Bluetooth je její obtížné zařazení do infrastruktury.

Další požadavek je přenos obrazu. Ten klade důraz na dostatečný datový tok mezi robotem a ovladačem. Pro přenos videa v reálném čase, tzv. videostream, v dostatečné kvalitě a malé latenci je třeba reálné přenosové rychlosti v řádu jednotek Mbit/s. Z tohoto důvodu nevyhovuje mobilní síť EDGE ani technologie ZigBee, jelikož jsou jejich přenosové rychlosti příliš malé. Potřebných přenosových rychlostí pro videostream dosahují Wi-Fi a mobilní sítě 3G a 4G.

Mobilní síť má dobré pokrytí. Wi-Fi má pouze omezený dosah od vysílače, umožňuje však zapojení do infrastruktury a vytvoření lokální sítě, která zvýší dosah. Využití mobilních sítí je však omezující pro závislost na pokrytí a cizí infrastrukturu. Podstatným argumentem proti mobilním sítím je vysoká nákladnost provozu.

Z výše porovnaných technologií přenosu dat jsem z důvodu rychlosti, ceny a dostupnosti (rozšířenosti) vybral technologii Wi-Fi. Z technologie Wi-Fi jsem se snažil využít standardu 802.11n, který je nejnovější a nabízí nejvyšší rychlosti i dosah.

KONSTRUKCE ROBOTU

Pro konstrukci podvozku robotu byla pro svou modularitu použita stavebnice MERKUR. Podvozek je tvořen rámem, na který jsou přimontovány motory a deska z plexiskla, která slouží jako nosná deska pro elektroniku. Rozmístění součástek zachycuje obrázek 3.2. Blokové schéma robotu je na obrázku 3.1.

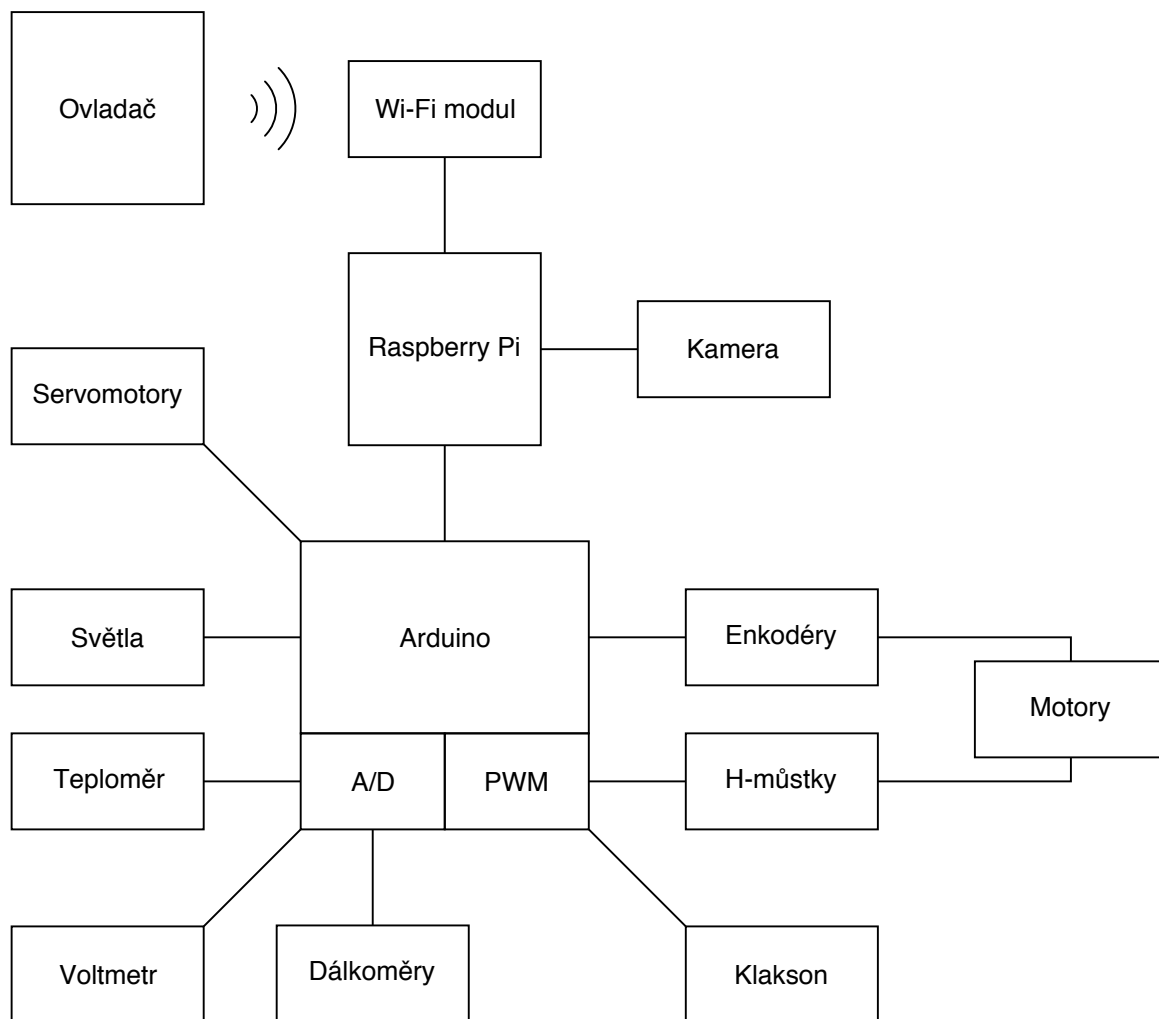
Jelikož by měl být robot schopen se pohybovat v obecném terénu, je postaven jako pásový se dvěma pásy, které pohání dva nezávislé motory. Díky pásové konstrukci bylo dosaženo dobrých manipulačních vlastností – robot je schopen otočit se na místě. Pásy by měly být výhodou i při přejíždění malých překážek. Naopak nevýhodou použití pásové konstrukce je náročnější ovládání, kdy je třeba ovládat každý pás samostatně. Z tohoto důvodu je pro robota obtížnější rovná jízda vpřed. Použité pásy jsou součástí stavebnice MERKUR.

3.1 Motory

O pohyb robotu se starají dva stejnosměrné motory opatřené převodovkou. Jeden motor je označován jako motor „A“, druhý motor „B“. Převodovka převádí rychlost otáčení osy kola a osy motoru v poměru 1: 21, 3. Katalogové parametry motorů jsou uvedeny v tabulce 3.1.

Pracovní napětí	6 V – 24 V
Nominální napětí	12 V
Otáčky (bez zátěže)	201 ot./min
Proud (bez zátěže)	46 mA
Moment síly při 0 ot. (v zátěži)	0,265 Nm
Proud při 0 ot. (v zátěži)	1 A
Převodový poměr převodovky	1 : 21,3
Váha	84 g

Tabulka 3.1: Vybrané katalogové parametry motorů



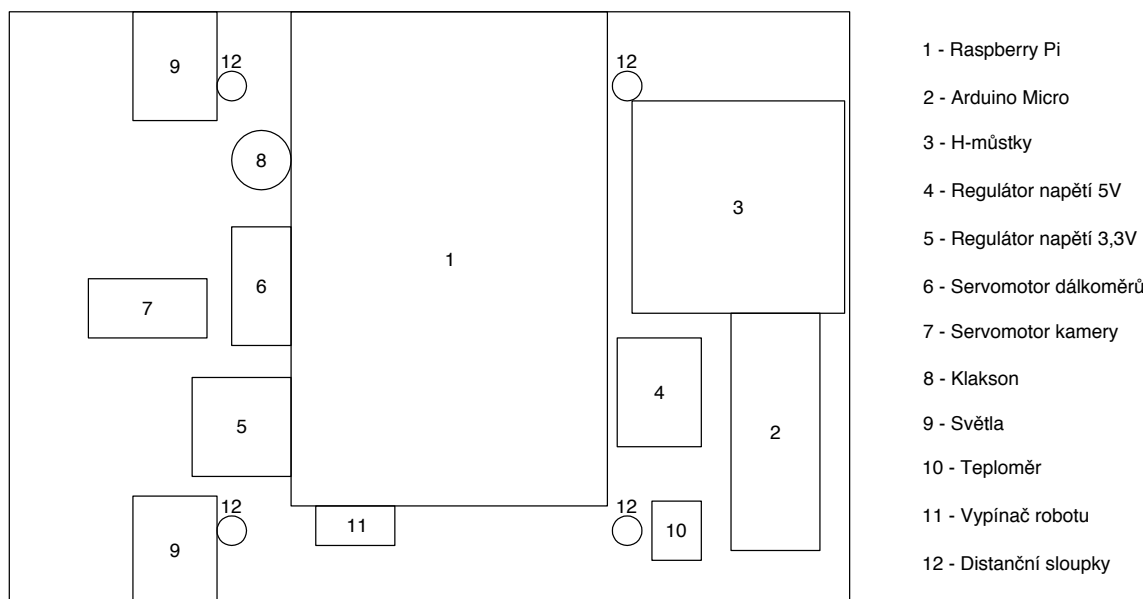
Obrázek 3.1: Blokové schéma robotu

Jako jednotka rychlosti otáčení kol je používán počet otáček za 1 minutu, resp. ot./min. . Tato jednotka není závislá na poloměru kola. Bylo by též možné udávat rychlost v jednotkách m/s , znamenalo by to však nutnost přepočtu na konkrétní velikost kola

3.1.1 Řízení motorů

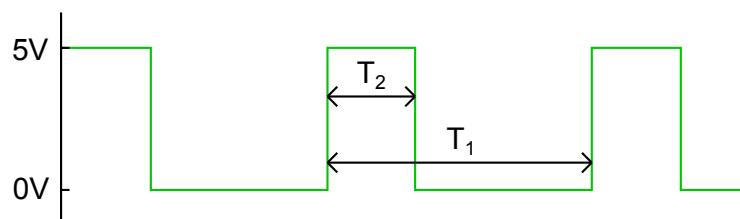
Jednou ze základních vlastností robotu je jízda o určité referenční rychlosti. Toho lze dosáhnout zařazením potenciometru mezi baterii a motor. Řízením velikosti odporu lze řídit napětí na motoru, které je úměrné otáčkám motoru. Tento způsob řízení se nazývá odporový.

Odporový způsob řízení má několik zásadních nedostatků. Snížením napětí na motoru dochází ke snížení momentu síly motoru a při malých napětích se vlivem hystereze motor vůbec nemusí otáčet. Dále má motor při tomto způsobu řízení velmi nízkou účinnost, protože proud prochází přes odpor, kde se mění na teplo.



Obrázek 3.2: Rozmístění součástek robotu na konstrukční desce

Při teleoperativním řízení robotu je třeba rychlost otáčení kol nastavit digitálně, odporový způsob řízení to však neumožňuje. Nastavení referenční rychlosti se musí provádět pomocí mikroprocesoru.

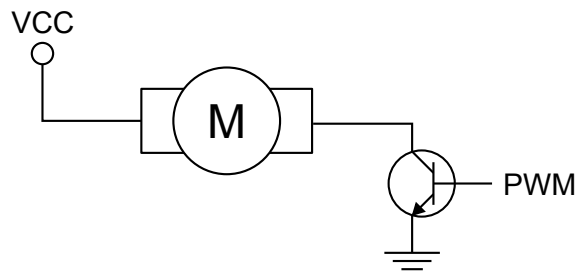


Obrázek 3.3: Ilustrace pulzně šířkové modulace

Všechny výše uvedené nedostatky řeší tzv. bezztrátové řízení motorů – pulzní. Základem pulzního řízení je šířková modulace (PWM¹). Nosný periodický signál má konstantní délku periody T_1 . Modulovanou veličinou je číselná hodnota rychlosti, která je udávána jako bezrozměrné číslo v intervalu $\langle 0, 1 \rangle$, tedy procentuálně vůči maximální rychlosti. Modulací se mění šířka pulzu T_2 tak, aby střída signálu byla rovna rychlosti. Výsledkem PWM je řídicí signál, který dosahuje logické úrovně $\{0, 1\}$. Velikost napětí logické úrovně „1“ je 5 V, logická úroveň „0“ má velikost napětí 0 V². Samotný motor je řídicím signálem spínán externě přes tranzistor, který je zapojen do série s motorem. Signál je přiveden na bázi tranzistoru, motor je připojen na jeho kolektor – viz obrázek 3.4

¹Z anglického „Pulse width modulation“

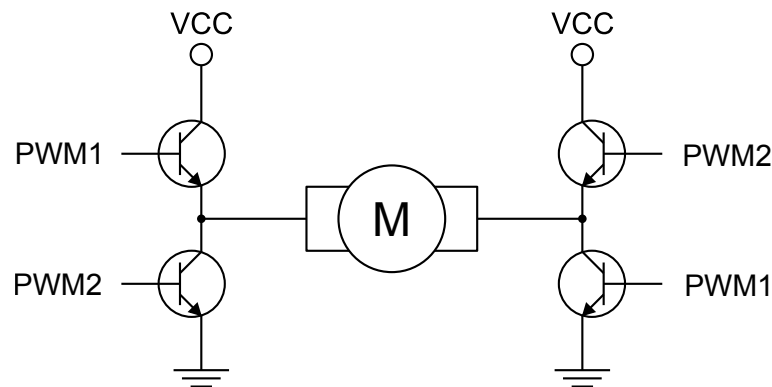
²Velikost napětí logické úrovně „1“ je dána výstupní napětíovou úrovní daného mikroprocesoru, který signál generuje. Obecně by se mělo mluvit o vysoké napětíové úrovni.



Obrázek 3.4: Řízení motoru PWM signálem a tranzistorem

Protože je motor induktivní zátěž, která je modelována jako sériová kombinace induktoru a rezistoru, vzniká při jejím rozepínání napěťová špička. Ta je generována vinutím motoru a má opačnou orientaci než napájecí napětí. Mohla by poškodit spínací tranzistor, proto je třeba zapojit paralelně k motoru ochrannou diodu, která vznikající napěťovou špičku odfiltruje.

Jedním tranzistorem lze řídit otáčky motoru, nelze však řídit směr otáčení. Pro takovou funkci je nutné použít můstkové zapojení – tzv. H-můstek. Obrázek 3.5 ilustruje jeho zapojení.



Obrázek 3.5: Principiální můstkové zapojení motoru

Při použití můstkového zapojení je potřeba použít dva kanály (PWM1 a PWM2). Jeden kanál slouží pro nastavení rychlosti otáčení motorů vpřed a druhý kanál pro otáčení vzad. Pokud je na jeden kanál přiveden signál, otevřou se příslušné tranzistory a skrz motor prochází proud. Přivedením signálu na druhý kanál se otevřou jiné tranzistory a proud prochází motorem opačným směrem. Schéma z obrázku 3.5 je pouze principiální, pro správnou funkci je třeba kanálům předřadit jedno či více hradel, která zabezpečí, aby nebyly použity oba kanály současně – při použití prvního kanálu je třeba druhý kanál zavřít a naopak. Zavřením kanálu se rozumí nastavení nulového potenciálu – nízké napěťové úrovně. Pokud by nedošlo k zavření druhého kanálu, nastal by zkrat baterie.

V tabulce 3.2 je uvedeno, jak nastavit PWM kanály pro příslušnou akci.

Kanál PWM1	Kanál PWM2	Efekt
0	0	Motor odpojen
1	0	Jízda vpřed
0	1	Jízda vzad
1	1	Zkrat baterie

Tabulka 3.2: Nastavení PWM kanálů; „1“ – otevřen, „0“ – zavřen

Po mikroprocesoru je požadováno, aby uměl generovat dva páry PWM signálů³ o dostatečné frekvenci. Pokud má řídicí signál dostatečnou frekvenci⁴, proud do vinutí motoru nestihá sledovat rychlé změny napětí. Otáčky jsou pak úměrné střední hodnotě procházejícího proudu. Při pulzním řízení nedochází ke snižování momentu síly při nízkých rychlostech.

Při konstrukci robotu byl použit integrovaný obvod L298N [9], který v jednom pouzdru obsahuje dva plné H-můstky pro řízení dvou motorů. PWM signály jsou generovány deskou Arduino (viz sekce 3.4). Více o zapojeních se lze dočíst v [10].

3.1.2 Testování motorů

Pro správné řízení motorů je třeba znát jejich chování. Z tohoto důvodu bylo provedeno měření pro různé referenční hodnoty a zjištěny vlastnosti motorů.

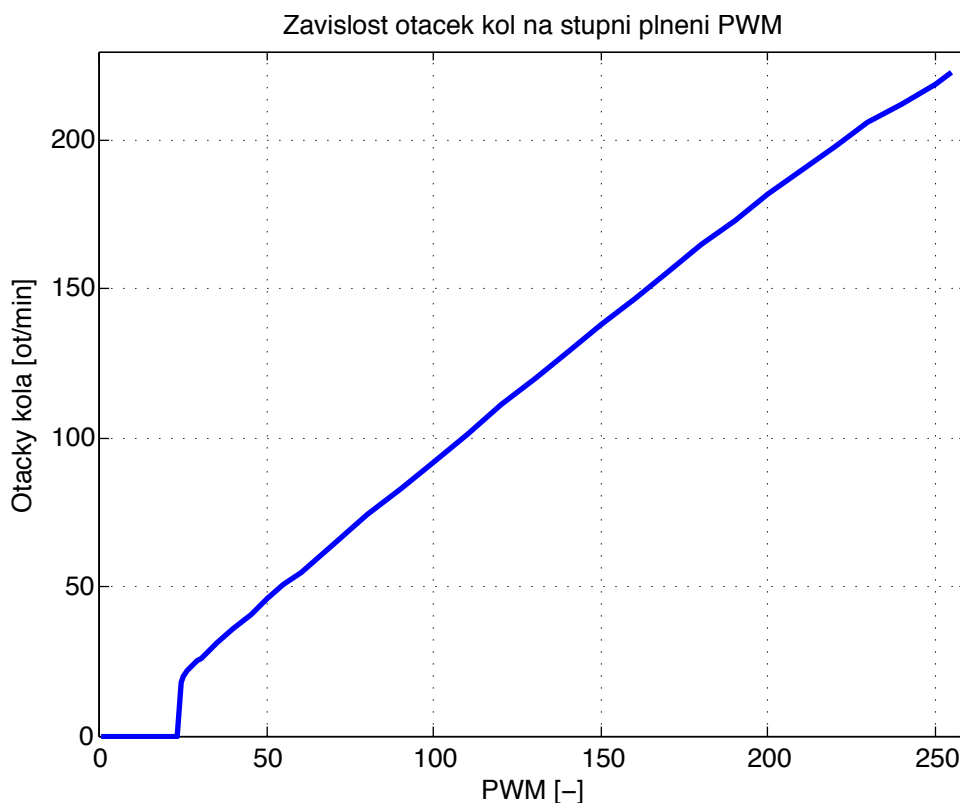
Z grafu 3.6 byla zjištěna hystereze motoru a stanovena minimální rychlost jako 20 ot./min⁵. Je patrné, že rychlost otáčení kol je přímo úměrná napětí. Napětí závisí na akumulátorech – podle jejich stavu nabití kolísá od 12 V do 16,8 V (viz sekce 3.3). Při stejné referenční hodnotě rychlosti je skutečná rychlost motorů s nabitými akumulátory jiná než s vybitými akumulátory⁶. Pro správnou činnost motorů je třeba, aby byly otáčky kol na napětí nezávislé. Toho se dosáhne zařazením regulátoru napětí, nebo měřením napětí na akumulátorech a výpočtem kompenzace. Dále je třeba dosáhnout lineárního průběhu závislosti otáček na stupni PWM. Podle grafu 3.6 je lineární závislosti dosaženo. Této lineární závislosti je však dosaženo pouze bez zátěže, při zatížení motoru (např. po nasazení pásů, při jízdě do kopce, jízdě s kopce, při různé hmotnosti robotu, proměnném tření...) se závislost začne měnit. Problém proměnného napětí i proměnné zátěže vyřeší zavedení regulátoru (viz sekce 3.5).

³Pokud by se využil integrovaný obvod disponující funkcí ENABLE, stačilo by generovat pouze jeden pár.

⁴Frekvence od stovek Hz do jednotek KHz

⁵Z grafu 3.6 si lze všimnout, že při plné rychlosti motor nedosahuje katalogové hodnoty rychlosti 3.1. To je způsobeno zařazením H-můstku a jeho tranzistorů, které nemají nulový odpor.

⁶Při vybitých akumulátorech je rychlost o 30 % nižší.



Obrázek 3.6: Naměřená závislost otáček kol na stupni plnění PWM při nominálním napětí 14,4 V; měřeno bez zátěže

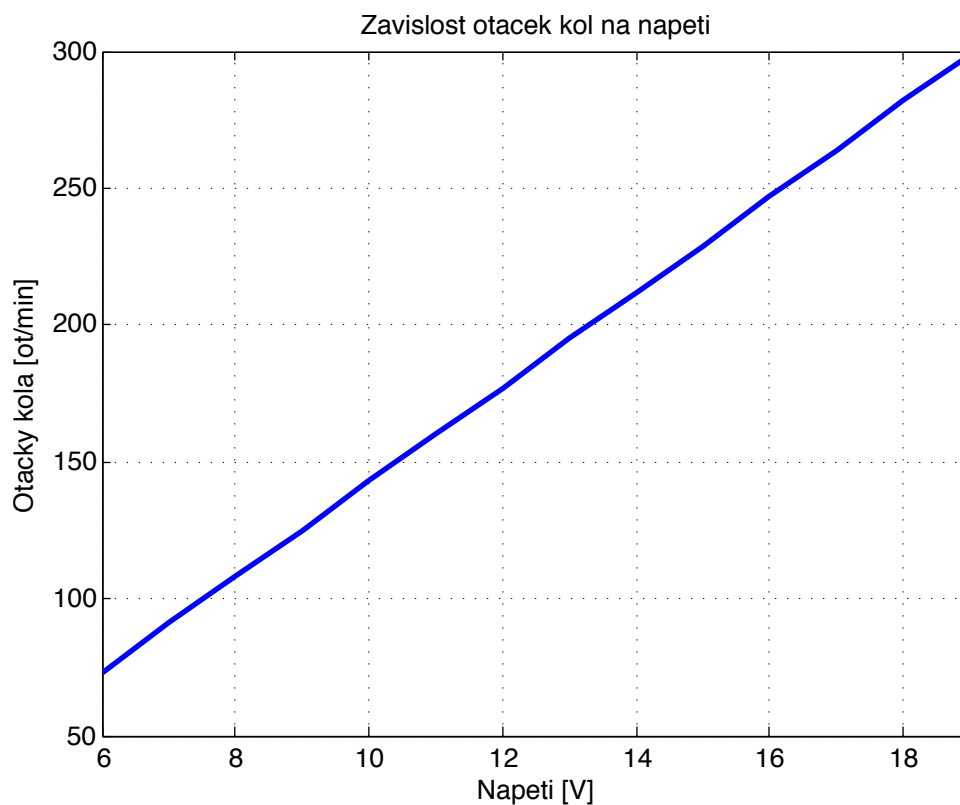
3.2 Enkodéry

Enkodéry měří skutečné otáčky motoru. Aby bylo možné otáčky měřit, jsou enkodéry přimontovány na osu motoru. Enkodéry jsou optické a podle výrobce generují 334 pulzů na jednu otáčku motoru. Frekvence pulzů enkodéru je přímo úměrná otáčkám motoru. Pro výpočet rychlosti otáčení kol OT platí převodní vztah

$$OT = \frac{f \cdot 60}{334 \cdot 21,3} \quad \left[\frac{otK}{min} \right] = \left[\frac{\frac{puls}{s} \cdot \frac{s}{min}}{\frac{puls}{otM} \cdot \frac{otM}{otK}} \right] \quad (3.1)$$

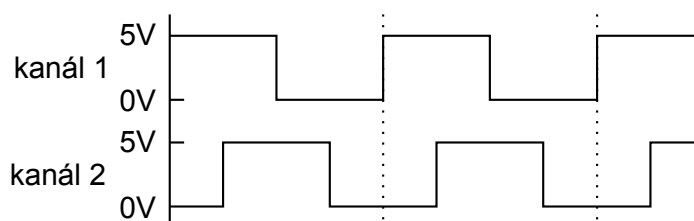
kde f je frekvence pulzů za vteřinu, kterou generují enkodéry. otM a otK jsou otáčky motoru resp. otáčky kola. Konstanta 21,3 je převodový poměr převodovky, konstanta 60 je převod z vteřin na minuty a konstanta 334 je počet pulzů na otáčku motoru.

Enkodéry jsou připojené k desce Arduino (viz sekce 3.4), která zpracovává signály a počítá frekvenci (viz sekce 3.4.3). Z frekvence není možné určit směr otáčení. Aby to možné bylo, má každý enkodér dva kanály, vzájemně posunutě o čtvrt periody. Pokud se motor točí jedním směrem, je druhý kanál posunutý o čtvrt periody doleva. Při otáčení druhým směrem je druhý kanál posunutý o čtvrt periody doprava (jak ilustruje obrázek 3.8).



Obrázek 3.7: Naměřená závislost otáček kol na napětí motoru; měřeno bez zátěže

Informace o aktuálních otáčkách kol jsou zobrazovány operátorovi a slouží jako zpětná vazba pro regulátory (viz sekce 3.5).



Obrázek 3.8: Ilustrace kanálů enkodérů

3.3 Napájení

Akumulátory by měly být nabíjecí a měly by být schopny poskytovat dlouhodobě velký proud. Z aktuálně dostupných typů akumulátorů připadaly ke zvážení následující:

- NiMH článek ⁷

⁷Akumulátory typu Ni-Cd mají výrazně horší parametry, proto ani nebyly zařazeny do srovnání.

- lithiový článek (Li-Ion, Li-Pol)
- olověný akumulátor

	Ni-MH	Li-Ion ⁸	Olověný akumulátor
Jmenovité napětí [V]	1,2	3,6	6 nebo 12
Hustota energie [Wh/kg]	100	200	40
Hustota energie [Wh/l]	300	500	100

Tabulka 3.3: Porovnání akumulátorů

Každý typ akumulátoru se vyrábí v jiném formátu, nelze tedy porovnávat jednotlivé články. Pro potřeby srovnání je pro každý typ vyjádřeno, kolik energie je možno uchovat v jednotkovém objemu a jednotkové hmotnosti.

V tabulce 3.3 jsou uvedeny hodnoty běžných (průměrných) akumulátorů [11]. Ze srovnání nejlépe vychází volba akumulátorů typu Li-Ion (nebo Li-Pol).

Jsou použity průmyslové akumulátory typu Li-Ion označované jako 18650 [12]. Katalogové parametry akumulátorů jsou v tabulce 3.4.

Označení	NCR18650A
Typ	Li-Ion
Výrobce	Panasonic
Jmenovité napětí	3,6 V
Kapacita	3100 mAh
Rozměry	18 mm × 65 mm (válec)

Tabulka 3.4: Vybrané katalogové hodnoty použitých akumulátorů

Akumulátory obsahují ochranný obvod chránící proti přebití, podbití i zkratu akumulátoru. Pro napájení robotu jsou použity celkem 4 články. Jmenovité napětí takového zdroje je 14,4 V. Akumulátory nejsou zdrojem konstantního napětí, proto se napájecí napětí v závislosti na stupni nabití pohybuje v rozsahu 12 V – 16,8 V ⁹. Protože je robot osazen množstvím součástek, jsou používány tři úrovně napájecího napětí. Napětí motorů je přímo napájecí napětí z akumulátorů (připojeno k VCC u H-můstku), napětí pro napájení elektroniky 5 V, které je získáno z akumulátorů regulátorem napětí, a napětí 3,3 V pro napájení světel.

Existují dvě možnosti regulace napětí: spínaný regulátor nebo lineární regulátor. Spínaný regulátor má oproti lineárnímu podstatně vyšší účinnost, a proto je pro akumulátorově napájené zařízení vhodnější. V robotu je použit spínaný regulátor MP1584, který je schopen poskytnout až 3 A a má účinnost 90 %.

⁹Napětí by mohlo být konstantní při použití regulátoru napětí, ten však pracuje s určitou účinností $\eta < 1$ a napájení by tak nebylo efektivní.

3.4 Arduino

Deska Arduino se v robotu stará o řízení motorů a o zpracovávání dat ze senzorů. Konkrétně je použita deska s označením Micro [13]. Arduino Micro je velmi populární vývojová deska, jejíž největší předností je osazení mikroprocesorem ATmega32u4 [14]. Jedná se osmibitový procesor řady AVR od firmy Atmel, který má zabudovaný USB řadič a umožňuje tedy přímé připojení jako USB-client zařízení. Arduino nabízí 20 I/O pinů, pracuje na frekvenci 16 MHz a jeho pracovní napětí je 5 V. 7 pinů může být naprogramováno jako PWM výstup, 12 pinů může sloužit jako vstup 10 bitového A/D převodníku¹⁰ a 4 piny mohou generovat externí přerušení.

Robot z desky Arduino využívá všechny výše popsané vlastnosti: PWM signály pro řízení motorů, externí přerušení pro měření frekvence a A/D převodník pro převod hodnot ze senzorů. Dále deska Arduino ovládá dva servomotory, světla a klakson.

3.4.1 Řízení motorů

Pro řízení motorů se používají PWM signály. Deska Arduino generuje PWM signály o frekvenci 490 Hz s rozlišením 8 bitů. To znamená, že lze nastavit 256 rychlostí v rozsahu 0 – 255, kdy 0 znamená, že se motor netočí a 255 znamená plné otáčky. Protože jeden motor potřebuje dva kanály, je třeba oba kanály nastavit zároveň. PWM signály nebudou nastavovány při přijetí informace o změně referenční rychlosti, ale budou nastavovány automaticky regulátorem, kterému se věnuje zvláštní sekce (viz sekce 3.5).

3.4.2 Měření napětí

Pro měření napětí Arduino disponuje A/D převodníkem s postupnou aproximací integrovaným v mikroprocesoru. A/D převodník má rozlišení 10 bitů a měří napětí v rozsahu 0 V – 5 V. Deska Arduino využívá měření napětí pro zpracování výstupu dálkoměrů (viz sekce 3.7.1) a pro měření napájecího napětí (viz sekce 3.7.2) a teploty (viz sekce 3.7.3).

3.4.3 Měření otáček

Deska Arduino měří frekvenci signálu z enkodéru programově. Na určitý časový interval t se aktivuje externí přerušení na příslušný pin prvního kanálu enkodéru. V momentě vyvolání přerušení, tj. při náběžné hraně signálu, se vykoná přerušovací rutina. Přerušovací rutina zaznamená okamžitou logickou hodnotu druhého kanálu (směr otáčení) a inkrementuje proměnnou c , která se chová jako čítač a uchovává počet přijatých pulzů. Po uplynutí časového intervalu t se deaktivuje externí přerušení. Frekvence f se vypočítá tak, že se počet pulzů vydělí časovým intervalem.

¹⁰Arduino má jeden A/D převodník, který multiplikován na více pinů.

$$f = \frac{c}{t} \quad (3.2)$$

Pro určení otáček kola se frekvence f dosadí do vzorce 3.1. Směr otáčení kol je určen ze zaznamenané okamžité logické úrovně druhého kanálu při přerušovací rutině. Pro měření otáček obou enkodérů se měření cyklicky střídá.

3.4.4 Regulace otáček

Pro výpočet regulátoru je použita knihovna PID Library pro Arduino [15]. Návrhu konstant regulátoru se věnuje sekce 3.5.

3.4.5 Komunikace

Deska Arduino je připojena k Raspberry Pi (viz sekce 3.6) USB kabelem jako zařízení se sériovou komunikací. Parametry sériové linky je třeba nastavit před započítím komunikace. Je použita konfigurace 115200/8-N-1 – tedy rychlost 115200 bit/s, 8 datových bitů, žádný paritní bit a jeden stop bit. Pro sériovou komunikaci je deska Arduino vybavena knihovnou `Serial` [16]. Ta obsahuje funkce pro čtení dat ze sériové linky i pro zápis dat na sériovou linku. V každé programové smyčce se kontroluje zásobník přijatých dat, případně se data okamžitě zpracují. Jednou za určitý časový interval se automaticky odesílají pravidelná informační data.

3.4.6 Servomotor

Servomotor slouží k natočení osy na určitou pozici. Řídící signál je puls o délce 1 ms – 2 ms, který se opakuje každých 20 ms. Výchozí poloha servomotoru je 90°. Této poloze odpovídá řídicí pulz o délce 1,5 ms. Pulz o délce 1 ms nastaví osu serva na úhel 0°, pulz o délce 2 ms nastaví osu serva na úhel 180°¹¹. Pro řízení servomotorů používá deska Arduino knihovnu `Servo` [17]. V robotu jsou použity dva servomotory. Jeden slouží k natáčení kamery a druhý pro otáčení dálkoměrů (viz sekce 3.7.1).

3.4.7 Struktura programu

Hlavní smyčka programu běží relativně rychle a v daných intervalech se provádí příslušné funkce (otáčení servomotorů, výpočet enkodérů, výpočet regulátoru atd). Měření enkodérů probíhá s periodou 5 ms, odesílání informací o otáčkách motorů probíhá s periodou 1 s a odesílání stavových informací (teplota, napětí akumulátoru) probíhá s periodou 10 s.

Program pro desku Arduino je psán v jazyce C s využitím API Arduino a několika knihoven. Zdrojové kódy programu jsou součástí této práce na přiloženém CD. Více o softwaru desky Arduino se lze dočíst v [18].

¹¹Některé servomotory mají rozsah pulzů i rozsah úhlů jiný.

3.5 Regulátor

Jelikož není napájecí napětí ani zátěž motoru konstantní, dochází k tomu, že referenční rychlost motoru neodpovídá rychlosti skutečné. Z tohoto důvodu je použit regulátor otáček ve formátu PID. Jako zpětná vazba z motoru slouží enkodéry (viz sekce 3.2). Celkem jsou regulátory dva – pro každý motor jeden, vzájemně nezávislé. Vstupem regulátoru je referenční (požadovaná) rychlost a výstupem hodnota, která se prostřednictvím PWM signálu a H-můstku nastaví na motor. Měření rychlosti je diskrétní v čase stejně tak jako algoritmus výpočtu hodnot regulátoru.

Protože měření rychlosti i výpočet regulátoru probíhají s vysokou frekvencí, je možné navrhnout regulátor spojitý v čase. Regulovanou veličinou je požadovaná rychlost a požadavky na regulátor jsou následující:

- nulová regulační odchylka v ustáleném stavu
- bez překmitu
- nejkratší možná doba ustálení
- zachování stability

Více informací o návrhu regulátoru je možné načerpat z přednášek prof. Šebka [19].

3.5.1 Identifikace systému

Chování systému popisuje přenosová funkce, cílem identifikace systému je její nalezení. Systém lze identifikovat z přechodové charakteristiky. Motor se vybudí jednotkovým skokem napětí a měří se odezva (rychlost). Výsledkem je odezva na jednotkový skok, tzv. přechodová charakteristika. Pro zvýšení přesnosti byla přechodová charakteristika získána zprůměrováním 20 měření.

Protože práce v časové oblasti je zbytečně náročná, je vhodné použít matematický nástroj, a to Laplaceovu transformaci. Použitím Laplaceovy transformace přejde celý návrh z časové oblasti do oblasti obrazů, kde je návrh konstant výrazně jednodušší.

Motor lze modelovat jako systém prvního řádu. Systém prvního řádu má obecný tvar přenosové funkce

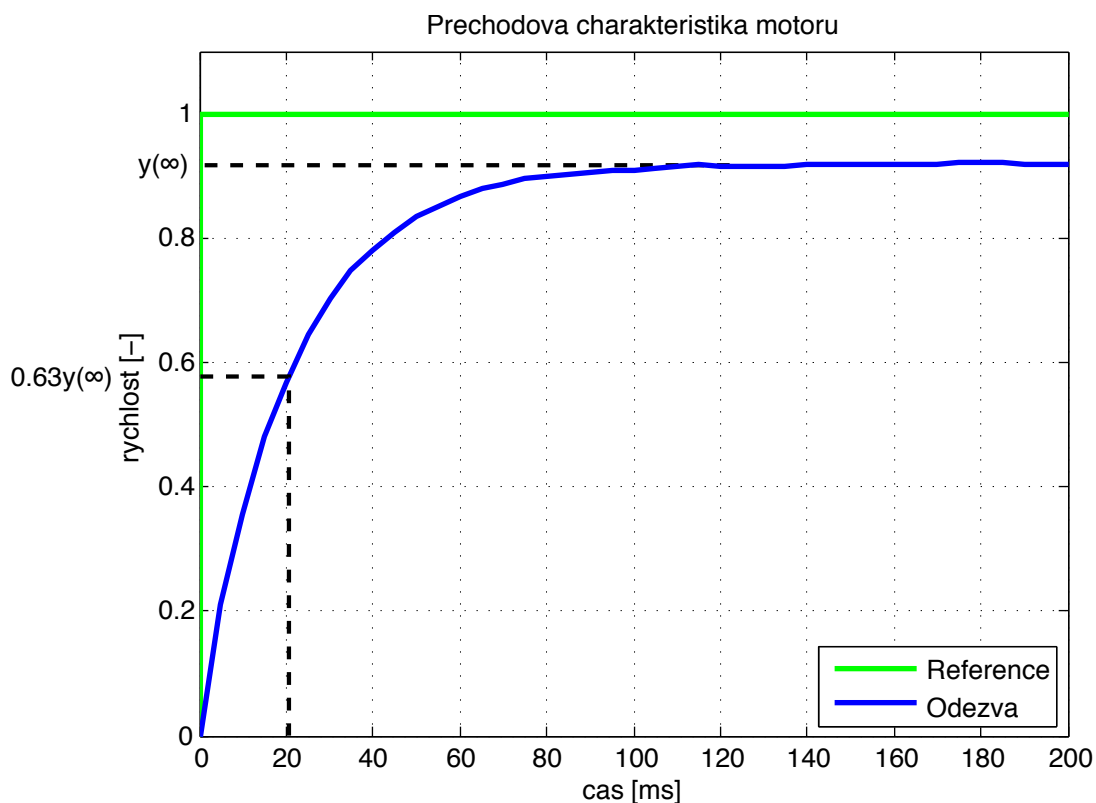
$$H(s) = \frac{K}{1 + T \cdot s} \quad (3.3)$$

Z grafu 3.9 se určí parametry K a T . Pro parametr K platí

$$K = \frac{y(\infty)}{u(\infty)}$$

Protože buzením je jednotkový skok $u(\infty) = 1$, K je ustálená hodnota odezvy a změří se tedy $K = y(\infty)$.

$$K = 0,92$$



Obrázek 3.9: Naměřená přechodová charakteristika motoru

Parametr T se určí jako doba, za kterou odezva dosáhne 63 % své ustálené hodnoty. V grafu 3.9 se odměří $0,63 \cdot y(\infty)$. Čas, který odpovídá této hodnotě, je parametr T .

$$T = 0,021$$

Po dosazení parametrů do rovnice 3.3 je hledaná přenosová funkce

$$H(s) = \frac{0,92}{1 + 0,021s} = \frac{43,8}{47,6 + s} \quad (3.4)$$

3.5.2 Návrh konstant

Nejprve je třeba zvolit formát regulátoru podle zadaných požadavků.

Použitím věty o konečné hodnotě se zjistí ustálená hodnota skokové odezvy.

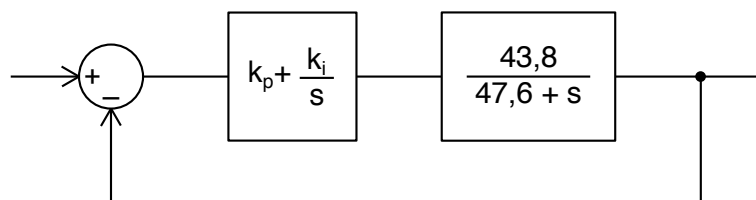
$$h_{ss} = \lim_{s \rightarrow 0} s \cdot H(s) \cdot \frac{1}{s} = 0,92$$

Ustálená hodnota skokové odezvy není rovna jedné, není tedy splněna podmínka nulové regulační odchylky v ustáleném stavu. Nejjednodušší způsob, jak docílit nulové regulační odchylky, je přidání integrační složky. Je tedy použit PI regulátor, což

je regulátor typu PID bez derivační složky, která byla vyřazena pro zjednodušení návrhu – na regulátor nejsou kladeny vysoké požadavky.

PI regulátor $R(s)$ má obecný tvar

$$R(s) = k_p + \frac{k_i}{s}$$



Obrázek 3.10: Schéma zapojení regulátoru

Schéma zapojení regulátoru včetně zpětnovazební větve je na obrázku 3.10.

Hledám konstanty k_p a k_i . Charakteristický polynom uzavřené smyčky systému H a regulátoru R je

$$c(s) = (47,6 + s) \cdot s + 43,8 \cdot (k_p \cdot s + k_i)$$

$$c(s) = s^2 + (47,6 + 43,8k_p) \cdot s + 43,8k_i \quad (3.5)$$

Zavedením regulátoru se systém změnil na systém druhého řádu. Charakteristický polynom vyjádříme i z požadavků na chování regulátoru. Pro systému druhého řádu má charakteristický polynom obecný tvar

$$c(s) = s^2 + 2\zeta\omega s + \omega^2 \quad (3.6)$$

kde ζ je tlumení a ω je frekvence.

Protože není tlumení v požadavcích na regulátor specifikováno, je použito vlastností kritického tlumení, kdy $\zeta = 1$. Kritické tlumení znamená, že systém dosáhne své ustálené hodnoty bez překmitu a zároveň v nejkratším možném čase. Použití kriticky tlumeného systému změnil rovnici charakteristického polynomu 3.6 na tvar

$$c(s) = s^2 + 2\omega s + \omega^2 = (s + \omega)^2$$

Kořeny této rovnice jsou

$$0 = (s + \omega)^2$$

$$\sigma_{1,2} = -\omega$$

Výsledná soustava má jeden reálný dvojnásobný pól, a to v bodě $-\omega$. Tento pól lze s ohledem na určitá pravidla zvolit. S ohledem na robustnost řešení je pól umístěn do stejného bodu jako pól původního systému H . Systém H má jeden pól v bodě

$$\sigma_H = -47,6$$

Do bodu $-47,6$ je umístěn dvojnásobný pól výsledné soustavy.

$$\omega = 47,6$$

Charakteristický polynom soustavy tedy vypadá následovně:

$$c(s) = (s + 47,6)^2 = s^2 + 95,2 \cdot s + 2266 \quad (3.7)$$

Porovnáním mocninných koeficientů rovnice 3.5 a 3.7 lze jednoduše určit konstanty k_p a k_i .

$$47,6 + 43,8k_p = 95,2$$

$$k_p = 1,09$$

$$43,8k_i = 2266$$

$$k_i = 51,7$$

Pro ověření stability stačí zjistit póly soustavy, tedy kořeny charakteristického polynomu. Ty byly přímo umísťovány – soustava má dvojnásobný pól v bodě $-47,6$. Protože se pól nachází v levé polovině s -roviny, jedná se o stabilní systém.

Výslednou soustavu $G(s)$ tvoří původní systém $H(s)$, regulátor $R(s)$ a zpětná vazba podle obrázku 3.10.

$$G(s) = \frac{R(s) \cdot H(s)}{1 + R(s) \cdot H(s)}$$

$$G(s) = \frac{(1,09 \cdot s + 51,7) \cdot 43,8}{s \cdot (47,6 + s) + (1,09 \cdot s + 51,7) \cdot 43,8} = \frac{47,6 \cdot s + 2266}{s^2 + 95,2 \cdot s + 2266} \quad (3.8)$$

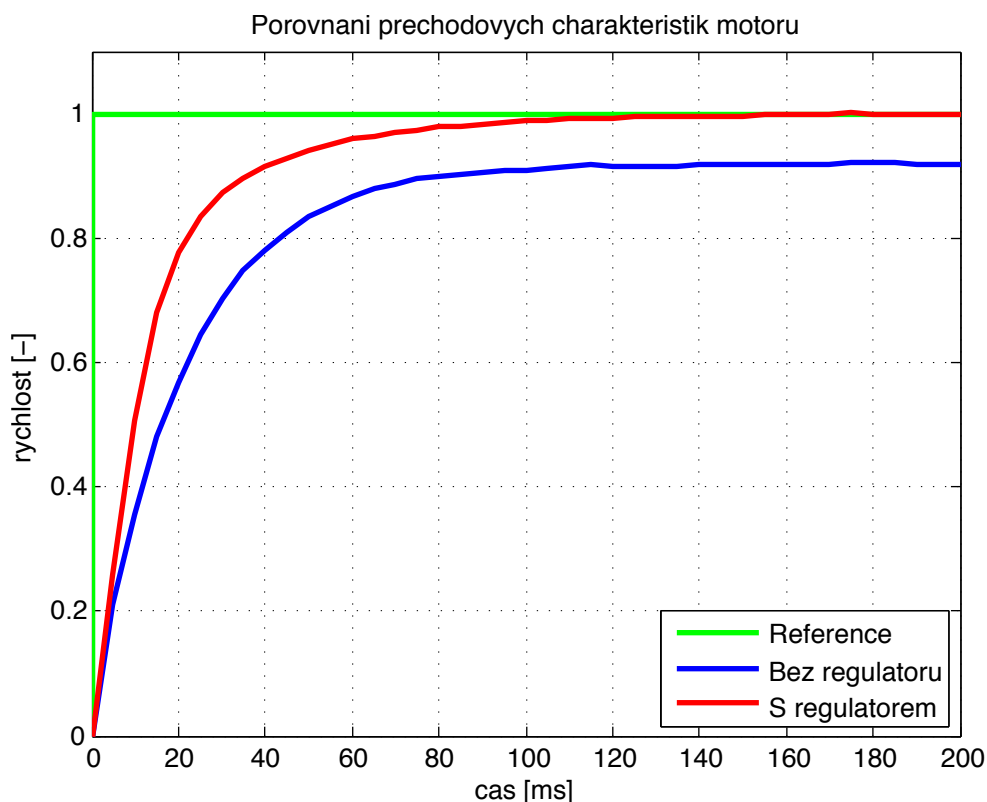
Použitím věty o konečné hodnotě se zjistí ustálená hodnota skokové odezvy soustavy $G(s)$.

$$h_{ss} = \lim s \cdot G(s) \cdot \frac{1}{s} = \frac{47,6 \cdot 0 + 2500}{0^2 + 95,2 \cdot 0 + 2500} = 1$$

Protože je ustálená hodnota skokové odezvy soustavy rovna jedné, je ustálená regulační odchylka nulová a soustava je schopna sledovat skokovou referenci.

3.5.3 Simulace

Na obrázku 3.11 je porovnáno chování motoru bez regulátoru a s regulátorem. Z grafu je patrné, že má motor s regulátorem rychlejší náběh. Dále si lze všimnout nulové regulační odchylky motoru. Volbou kritického tlumení bylo dosaženo nejrychlejšího možného náběhu bez překmitu.

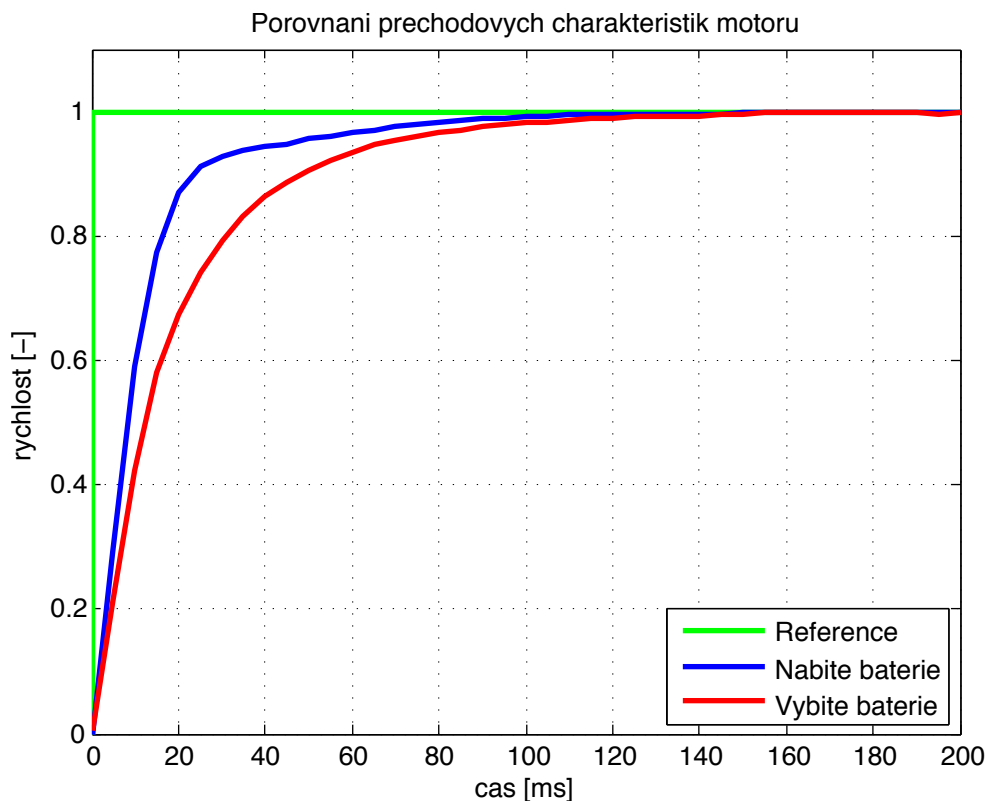


Obrázek 3.11: Porovnání motoru bez regulátoru a motoru s regulátorem

Navržený regulátor pracuje při nominálním napětí akumulátoru 14,4 V. Při změně napájecího napětí dochází ke změně celého systému, protože je motor závislý na napětí podle obrázku 3.7. Při změně napětí by tedy bylo nutné přepočítávat konstanty regulátoru. Porovnání chování regulátoru při vybitém a nabitém akumulátoru zachycuje obrázek 3.12. Z něj je patrné, že se regulátor chová různě a při vybitém akumulátoru má pomalejší náběh. Důležitá je schopnost regulátoru pracovat i v těchto mezních situacích. Není tedy nutné navrhovat jiné konstanty regulátoru.

3.6 Raspberry Pi

Raspberry Pi je hlavní počítač robotu [20]. Zabezpečuje komunikaci a snímání obrazu. Tento počítač vyvíjí britská nadace Raspberry Pi Foundantion. Základem je SoC BCM2835 firmy Broadcomm, který obsahuje procesor ARM1176JZF-S s taktem



Obrázek 3.12: Porovnání činnosti regulátoru při různě nabitém akumulátoru

700 MHz, grafický procesor VideoCore IV a 512 MB RAM. Operačním systémem je Raspbian [21], jedná se o linuxovou distribuci Debian upravenou pro Raspberry Pi. Velikost počítače je 86 cm × 56 mm. Raspberry Pi je napájeno z 5V zdroje a průměrná spotřeba činí 3 W. K dispozici jsou dva USB porty. Jeden USB port slouží pro pro připojení desky Arduino, druhý pro připojení Wi-Fi adaptéru.

Deska Arduino se v systému identifikuje jako sériové zařízení na adrese `/dev/ttyACM0`. Nastavení parametrů sériové komunikace se provede programem `stty`. Se sériovým zařízením lze zacházet jako se souborem (čtení a zápis dat).

Jako Wi-Fi adaptér je použit Edimax EW-7811Un. Aby mohl fungovat v režimu AP, bylo nutné přeinstalovat ovladač [22]. Nastavení sítě probíhá stejně jako na většině linuxových OS. Wi-Fi adaptér má určenou statickou IP adresu `192.168.42.1` a zároveň je povolený DHCP server, takže je připojenému robotu automaticky přidělena IP adresa z rozsahu `192.168.42.4 – 192.168.42.20`.

Při zapnutí robotu spustí Raspberry Pi skript `init.sh`, který provede příslušná inicializační nastavení, aby robot mohl fungovat – zapne Wi-Fi adaptér v režimu AP a tím vytvoří Wi-Fi síť. SSID sítě je jméno robotu, heslo pro přístup do sítě je heslo robotu. Jméno i heslo je pro každý robot unikátní a slouží pro verifikaci, bez jejich znalostí není možné robot ovládat. Další součástí, kterou je nutné zapnout při spuštění, je webový server, jehož prostřednictvím probíhá streamování videa. Rovněž se spustí program `host.java` poslouchající na určitém portu připojení robotu.

3.6.1 Stream videa

Raspberry Pi obsahuje konektor pro připojení kamerového modulu. Tento modul disponuje snímačem s rozlišením 5 megapixel a je schopen natáčet video s rozlišením až 1080p 30 fps nebo VGA 90 fps. Kamerový modul nabízí široké možnosti nastavení efektů (např. time-lapse, slow motion). Kamera je na robotu umístěna v přední části a zachycuje dění před robotem. Je připevněna na servomotor, který umožňuje natáčení celé kamery do stran a výrazně tím zvětšuje její pozorovací úhel. Nasnímaný obraz je zobrazován operátorovi ve formě streamu. Stream je přenos videa v reálném čase, který má vysoké nároky na přenosovou rychlost i na výpočetní výkon kodéru a dekodéru. Hlavním parametrem, kterému byla věnována pozornost, je latence přenosu obrazu. Pro použití při teleoperaci nesmí latence překročit určitou empirickou hranici. Dále je třeba vhodně nastavit rychlost snímkování (fps) a rozlišení zachytávaného videa.

Streamování videa zajišťuje program MjpegStreamer [23]. Do něj je nainstalován plugin `input_raspicam` [24], který je získává snímky přímo z kamery bez použití původní aplikace. Použitím pluginu bylo dosaženo výrazné snížení latence oproti původní aplikaci. Spuštěním programu MjpegStreamer dojde ke spuštění webového serveru, který je dostupný na portu 8080. Na něm je streamované video přístupné přes HTTP protokol. Prohlížení streamu je možné z libovolného zařízení, které je schopné se připojit k počítači Raspberry Pi, a libovolným programem, který pracuje s HTTP protokolem. To znamená, že po připojení k Raspberry Pi je možné video prohlížet v libovolném internetovém prohlížeči nebo přehrávači videa, např. stačí v internetovém prohlížeči zadat adresu `http://192.168.42.1:8080`¹².

Streamované video¹³ má rozlišení 800 px × 600 px a frekvenci snímků 10 fps. Jedná se o kompromis, rozlišení a snímkovací frekvence jsou nepřímo úměrné. Rozlišení je možné zvolit vyšší (až FullHD), pak je nutné počítat se sníženou frekvencí snímků. Naopak pro vyšší snímkovací frekvenci by bylo nutné snížit rozlišení videa.

Přenášené video má vysoký datový tok, cca 8 Mbit/s. To je dáno tím, že program MjpegStreamer nepoužívá žádnou kompresi videa a přenáší surový jpeg obrázek. Kvůli tomu není výsledné video tak kvalitní jako při použití video kodeků. Výhodou je naopak nižší vytížení procesoru při zpracovávání videa (cca 5 %).

3.6.2 Program `host.java`

Program `host.java` je napsán v jazyce java a je spouštěn při startu počítače. Jeho činností je zprostředkování komunikace mezi deskou Arduino a operátorským ovládacím panelem. Zdrojové kódy programu, včetně jeho částí, jsou součástí této práce na příloženém CD.

Program spustí `ServerSocket`, který poslouchá na portu 8081 příchozí žádosti. Při připojení robotu je od ovládacího panelu přijata žádost o `Socket`. Po jejím

¹²192.168.42.1 je IP adresa Wi-Fi adaptéru v počítači Raspberry Pi

¹³Použití slova „video“ není úplně správné, nejedná se totiž o video, nýbrž o sled obrázků. Ten je ale tak rychlý, že vyvolává dojem videa. Proto si dovolím používat slovo „video“.

přijetí je ustaveno spojení a pro **Socket** jsou otevřeny komunikační kanály – vstupní a výstupní. Vstupní kanál přijímá data z ovládacího panelu a zapisuje je do souboru `/dev/ttyACM0` (zápis dat do desky Arduino). Výstupní kanál čte data ze souboru `/dev/ttyACM0` (čtení dat z desky Arduino) a odesílá je do ovládacího panelu.

3.7 Senzory

Vedle kamery, kterou lze také považovat za senzor, je hlavním průzkumným prvkem robotu dvojice infračervených senzorů (dálkoměrů) měřících vzdálenost od překážky. Dalšími senzory jsou voltmetr, který měří napětí na akumulátoru, a teploměr, který měří okolní teplotu robotu. Dále je zde zařazen klakson a světla – nejedná se sice o senzory, ale činnosti senzorů pomáhají.

3.7.1 Dálkoměry

Pro měření vzdálenosti je možné použít několik druhů senzorů – laserové, ultrazvukové, nebo infračervené optické.

Laserové senzory jsou nejpřesnější, mají však velké rozměry a jsou velmi drahé.

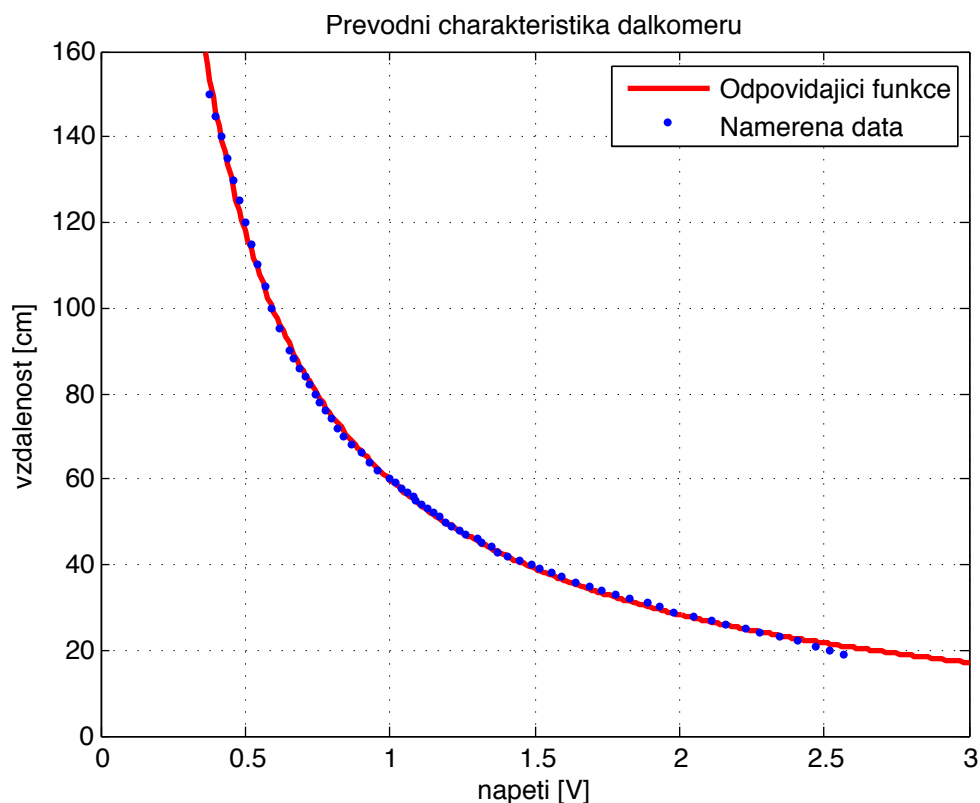
Ultrazvukové dálkoměry jsou levné. Jsou schopné změřit vzdálenost překážky, její rozměry ale změřit nedokážou, protože má vyslaný akustický pulz tvar kuželu. Měření u ultrazvukových dálkoměrů probíhá tak, že senzor měří čas mezi vyslaným a přijatým akustickým signálem.¹⁴ Nevýhodou ultrazvukových dálkoměrů jsou odrazy akustického signálu, které mohou výrazně znehodnotit měření.

Jako dálkoměry byly zvoleny infračervené optické senzory, které měří vzdálenost bodově. Senzory pracují na principu triangulace – měří úhel mezi vyslaným a přijatým paprskem. Z tohoto způsobu měření vyplývá, že převodní charakteristika nemá mít lineární průběh.

Pro robot jsou použity dva IR dálkoměry od firmy SHARP, každý o jiném rozsahu měření. Krátký dálkoměr má označení GP2Y0A41SK0F [25] a je schopen měřit v rozsahu 4 cm – 30 cm. Dlouhý dálkoměr má označení GP2Y0A02YK0F [26] a je schopen měřit v rozsahu 20 cm – 150 cm. Dálkoměry jsou napájené ze zdroje napětí 5 V, výstupem je napěťová úroveň odpovídající naměřené vzdálenosti. Analogovou hodnotu napětí měří deska Arduino A/D převodníkem. Změřené napětí je potřeba přepočítat na vzdálenost. Tento výpočet provádí na základě znalosti převodní charakteristiky deska Arduino.

Převodní charakteristiky obou dálkoměrů byly určeny experimentálně, protože data z dokumentace nejsou přesná. Změřená data byla proložena křivkou pro zjištění funkčního předpisu převodu. Z převodních charakteristik vyplývá, že je vhodné používat krátký dálkoměr pouze pro měření vzdáleností < 20 cm, proto byla i převodní charakteristika určována pouze do této hodnoty. Pro vzdálenosti > 20 cm je

¹⁴Lepší dálkoměry měří více přijatých odrazů, jsou tedy schopné změřit vzdálenost překážky za překážkou.



Obrázek 3.13: Změřená převodní charakteristika dlouhého dálkoměru proložená křivkou

použit dlouhý dálkoměr. Převodní charakteristiky byly proloženy křivkou mocninné funkce. Pro dlouhý dálkoměr byla odvozena převodní charakteristika jako funkce

$$d_1 = 68,45 \cdot U_1^{-0,8888} - 8,641$$

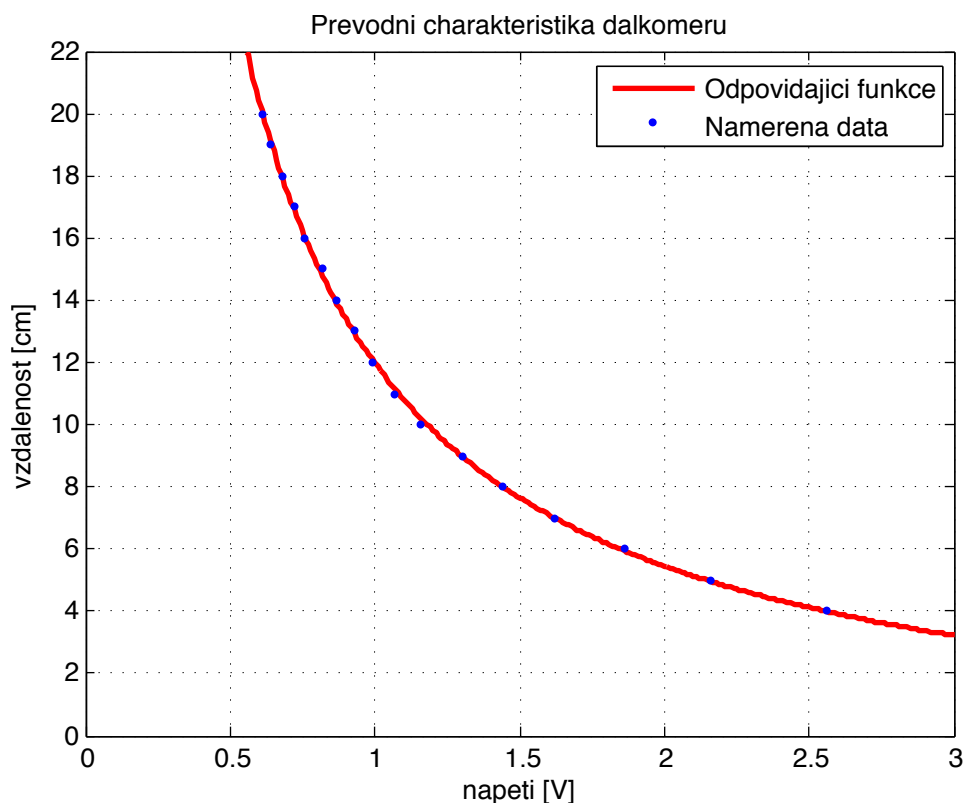
kde U_1 je naměřené napětí dlouhého dálkoměru. Pro krátký dálkoměr byla odvozena převodní charakteristika jako funkce

$$d_1 = 13,49 \cdot U_2^{-0,952} - 1,52$$

kde U_2 je naměřené napětí krátkého dálkoměru.

Dálkoměry jsou umístěny na servomotoru, který jimi umožňuje otáčet. Senzory jsou tedy schopné proměřit určitou výšeč před robotem. Oba dálkoměry se při své činnosti vzájemně neovlivňují, protože každý pracuje na jiné vlnové délce – dlouhý na vlnové délce 850 nm, krátký na vlnové délce 870 nm.

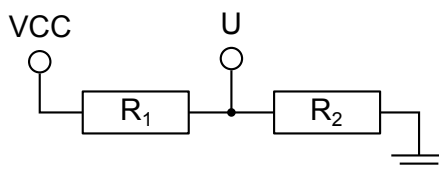
Data z dálkoměrů jsou odesílána do ovladače jako dvojice hodnot – úhel a vzdálenost naměřená v tomto úhlu. Úhel je udáván ve stupních a vzdálenost je udávána v centimetrech. Operátorovi na ovladači je tak zobrazena celá výšeč před robotem a jsou vykresleny překážky.



Obrázek 3.14: Změřená převodní charakteristika krátkého dálkoměru proložená křivkou

3.7.2 Voltmetr

Voltmetr je jednoduchý senzor měřící napětí na akumulátoru. Údaj o napětí akumulátoru slouží k detekci vybitých akumulátorů, případně pro přibližné určení doby provozu robotu. Napětí akumulátoru může být až 17 V. Aby bylo možné měřit takto vysoké napětí, je třeba ho zmenšit do rozsahu 0 V – 5 V. K tomu je využito napěťového děliče, který napětí akumulátoru analogově vydělí ve vhodném poměru. Takto zmenšené napětí se změří A/D převodníkem a získaná číselná hodnota se stejným poměrem vynásobí.



Obrázek 3.15: Schéma zapojení voltmetru a teploměru

Pro napětí U platí

$$U = VCC \cdot \frac{R_2}{R_1 + R_2}$$

kde VCC je připojeno na akumulátor, R_1 a R_2 jsou vhodně zvolené rezistory.

Při konstrukci robotu byly použity odpory s hodnotami $R_1 = 30\text{ k}\Omega$ a $R_2 = 10\text{ k}\Omega$. Dělič napětí s těmito hodnotami dělí čtyřmi, senzor je tedy schopen pracovat až do napětí 20 V. Výkonová ztráta na této sériové kombinaci je $< 10\text{ mW}$.

3.7.3 Teploměr

Pro sestavení teploměru byl použit termistor NTC [27]. Schéma zapojení je obdobné jako na obrázku 3.15. Napěťový dělič je použit tak, že je VCC připojeno k napětí 5 V a R_1 je termistor. Pro nejvyšší možnou přesnost je vhodné zvolit velikost rezistoru R_2 stejnou, jako je nominální hodnota termistoru ¹⁵.

Při konstrukci robotu byl použit termistor s nominální hodnotou $10\text{ k}\Omega$ a odpor o stejné hodnotě. Z dokumentace je určena převodní charakteristika jako funkce

$$T = \left(A + B \cdot \ln \frac{R}{R_{25}} + C \cdot \ln^2 \frac{R}{R_{25}} + D \cdot \ln^3 \frac{R}{R_{25}} \right)^{-1}$$

kde R je naměřená hodnota odporu a konstanty A, B, C, D jsou uvedeny v tabulce 3.5. Teplota T je v Kelvinech.

Hodnota odporu se vypočítá ze změřené hodnoty napětí U napěťového děliče jako

$$R = \frac{R_2}{U} \cdot (5 - U)$$

Napětí U měří deska Arduino, která provádí i výpočet odporu R a teploty T . Naměřená teplota je pravidelně odesílána do ovládacího panelu.

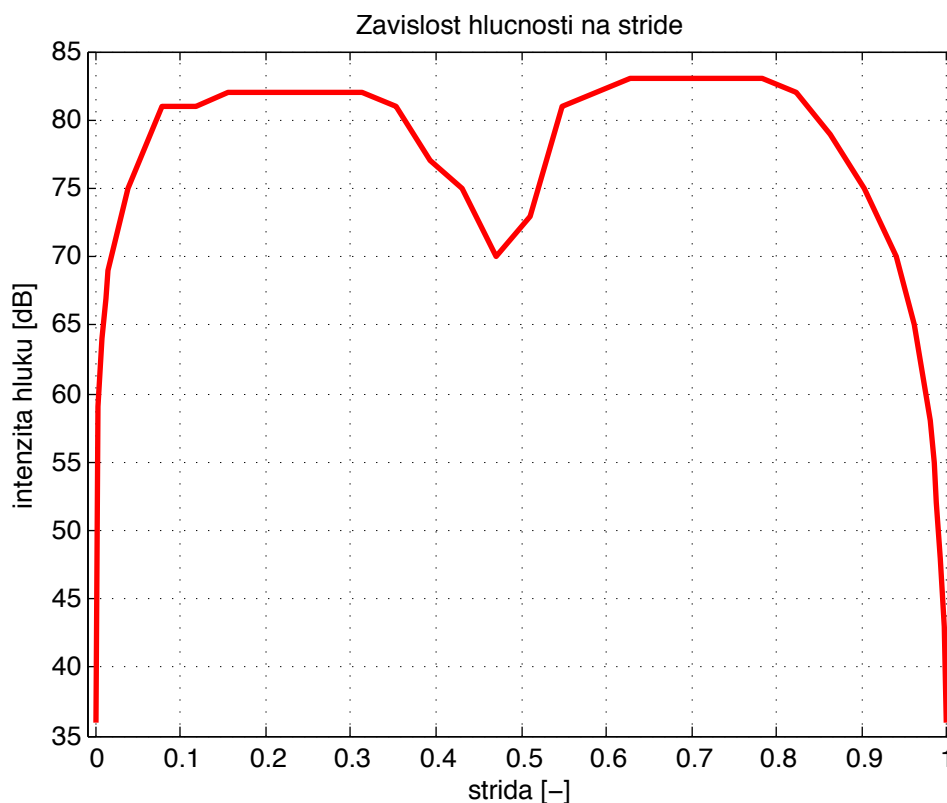
Konstanta	Hodnota
A	$3,354016E - 03$
B	$2,569850E - 04$
C	$2,620131E - 06$
D	$6,383091E - 08$

Tabulka 3.5: Konstanty pro výpočet teploty

3.7.4 Klakson

Klakson je součástí robotu proto, aby byl robot schopný na sebe při své činnosti upozornit. Jako klakson je použit piezoreproduktor, který je buzen PWM signálem z desky Arduino. Ta generuje PWM signál o frekvenci 490 Hz, což je vhodná slyšitelná frekvence. K určení střídy signálu, při které je piezoreproduktor nejhluchnější, byl použit zvukoměr. Naměřená závislost intenzity akustického hluku na střídě budícího signálu je na obrázku 3.16.

¹⁵V dokumentaci je nominální hodnota termistoru udávána jako hodnota R_{25} a znamená odpor při 25°C .



Obrázek 3.16: Závislost hluchnosti piezorekroductoru na střídě PWM signálu

Podle obrázku 3.16 je nejvyšší hluchnosti dosaženo při střídě signálu 0,7, což při 8-bitovém PWM signálu odpovídá stupni plnění PWM 180.

3.7.5 Světla

Světla jsou důležitá pro činnost robotu za zhoršených světelných podmínek, kdy kamera není schopná pracovat. Robot je osazen dvěma 1 W bílými diodami LED, které pracují na nominálním napětí 3,3 V. Jejich vyzařovací úhel je 120°, proto jsou navíc doplněny čočkami, které vyzařovací úhel zmenší na 60°. Kvůli tomu, že se jedná o výkonové diody LED, i kvůli jinému pracovnímu napětí je pro napájení použita separátní větev, kterou tvoří spínaný regulátor napětí s vyvedeným pinem ENABLE pro povolení funkce regulátoru. Tento pin je připojen k desce Arduino a slouží k vypínání a zapínání světla.

PLATFORMA UŽIVATELSKÉHO OVLÁDACÍHO ROZHRANÍ

Robot nebude ovládán klasickým ovladačem na RC modely. Cílem práce je prozkoumání možností ovládání robotu obyčejnou spotřební elektronikou – ta je totiž v současné době velmi rozšířená, dostupná, univerzální a levná. Výběr vhodné platformy zahrnuje výběr pracovního prostředí jak po stránce hardwaru, tak softwaru.

Po hardwarové stránce je požadována podpora Wi-Fi technologie v režimu Wi-Fi client. Tato práce se zaměřuje na zařízení mobilní, použitým zařízením tedy je mobilní telefon či tablet, který disponuje technologií Wi-Fi.

Ovládacím rozhraním je aplikace, bylo tedy nutné vybrat vhodné softwarové prostředí [28]. Na poli mobilních zařízení v současné době existují 3 nejrozšířenější platformy. Jedná se o platformy Android, Windows a iOS.

4.1 Popis platformem

Platforma Android je relativně nová platforma – je vyvíjena od roku 2007 firmou Google. Platforma je schopná fungovat na různém hardwaru, proto se rychle stala velmi oblíbenou a nyní je nejrozšířenější. Aktuálně nejnovější (rok 2014) verze operačního systému je 5.0. Platforma Android je založena na linuxovém jádře a je podobná platformě Java. Aplikace je kompilátorem přeložena do bytecode, který je následně v zařízení spuštěn na virtuálním stroji Dalvik. Od verze 4.4 se používá Android RunTime (ART) [29]. Jedná se o složitější kompilaci, protože při instalaci dochází k optimalizaci na konkrétní zařízení. Výhodou je vyšší efektivita a nižší spotřeba.

Platforma Windows je známá především z počítačů PC. Poslední verze operačního systému Windows přizpůsobená pro mobilní zařízení je označována jako Windows Phone 8. Platforma je vyvíjena firmou Microsoft. Prostedí potřebné pro běh aplikací se nazývá „.NET“.

Platforma iOS je vyvíjena firmou Apple. Jedná se též o novou platformu, nejnovější verze operačního systému je iOS 8. Nevýhodou této platformy je vazba na konkrétní hardware firmy Apple. Další nevýhodou je omezená možnost výběru vývojářských nástrojů, která je zúžena pouze na oficiální sadu.

4.2 Porovnání

Všechny platformy nabízí širokou škálu programovacích jazyků i vývojových prostředí. Tabulka 4.1 uvádí oficiálně podporované vývojářské nástroje i porovnání tržního podílu jednotlivých platforem ¹.

Platforma	Android	Windows	iOS
Programovací jazyk	Java	C#	Swift (Objective-C)
Vývojové prostředí	Android Studio	Visual Studio	Xcode
Tržní podíl [28]	85 %	3 %	12 %

Tabulka 4.1: Porovnání platforem mobilních zařízení

Nevýhodou platformy iOS je její uzavřenost a nutnost placení za vývoj aplikací roční poplatky. Nevýhodou platformy Windows je její slabé rozšíření. Zvolena byla softwarová platforma Android pro svou univerzálnost a velké rozšíření.

Jako vhodný hardware byl vybrán tablet, protože má velký displej pro vizualizaci senzorických dat a snímků z kamery. Nejvhodnější velikost je tablet s úhlopříčkou 10 palců. Aplikace ale musí být funkční i na zařízeních s menším displejem.

4.3 Android

Aplikace je vyvíjena pro verzi Android 5.0, musí být však schopná zpětné kompatibility. Procentuální zastoupení různých starších verzí systému Android ukazuje tabulka 4.2. Tabulka je převzatá z [31] a jsou v ní uvedeny jenom verze se zastoupením > 1 %.

Verze	Název	API	Zastoupení
2	Gingerbread	10	10 %
4.0	Ice Cream Sandwich	15	8 %
4.1	Jelly Bean	16	21 %
4.2		17	20 %
4.3		18	7 %
4.4	KitKat	19	34 %

Tabulka 4.2: Procentuální zastoupení různých verzí systému Android

Protože Android verze 2 je již zastaralý a nepodporuje tablety, je aplikace vyvíjena se zpětnou kompatibilitou do verze 4.0 resp. API 15. Aplikaci tak bude možné spustit na většině zařízeních s operačním systémem Android.

¹Údaj tržního podílu je za období Q1/2014.

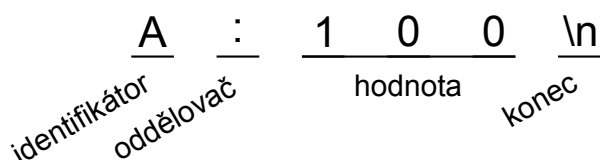
TELEOPERÁTORSKÉ UŽIVATELSKÉ OVLÁDACÍ ROZHRANÍ

Ovládacím rozhraním robotu je aplikace pro operační systém Android. Aplikace je psána v programovacím jazyce Java. Zdrojové kódy aplikace jsou součástí této práce na příloženém CD. Pro vývoj aplikací pro systém Android je třeba znát tzv. Application framework [32]. Ten obsahuje základní stavební prvky systému Android včetně komponent pro komunikaci, správu běhu aplikace a grafické prvky.

5.1 Komunikační protokol

Ke sjednocení komunikace mezi jednotlivými moduly slouží komunikační protokol. Jedná se o předpis, v jakém formátu budou přenášeny informace, aby jim rozuměly jednotlivé části robotu a byly čitelné pro člověka. Čitelnost je požadována z důvodu usnadnění ladění aplikace. Při volbě protokolu je třeba volit kompromis mezi objemem přenášených dat a čitelností.

Byl zvolen příkazový protokol skládající se z identifikátoru a hodnoty. Identifikátorem je jedno libovolné písmeno. Následuje oddělovač, kterým může být libovolný znak – např. čárka, středník, dvojtečka, pomlčka či mezera. Poté následuje argument (číselná hodnota), nebo více argumentů oddělených libovolným oddělovačem. Posledním znakem je znak nový řádek – LF (ASCII znak 10), který znamená konec příkazu.



Obrázek 5.1: Skladba komunikačního protokolu

Identifikátor s velkým písmenem znamená požadavek na nastavení hodnoty. Iden-

tifikátor s malým písmenem slouží k informování o aktuální hodnotě. Některé příkazy slouží pouze k informování o stavu robotu a není třeba příslušné veličiny nastavovat. Takové příkazy jsou implementovány pouze jednostranně. Seznam všech implementovaných příkazů uvádí tabulka 5.1.

Identifikátor	Počet argumentů	Význam
A	1	Nastavení rychlosti motoru „A“
a	1	Aktuální rychlost motoru „A“
B	1	Nastavení rychlosti motoru „B“
b	1	Aktuální rychlost motoru „B“
C	1	Nastavení úhlu natočení kamery
c	1	Aktuální úhel natočení kamery
d	2	Data dálkoměrů
K	1	Zapnutí/vypnutí klaksonu
k	1	Stav klaksonu
L	1	Zapnutí/vypnutí světel
l	1	Stav světel
S	0	Dotaz na stavové informace
t	1	Okolní teplota robotu
v	1	Napětí akumulátoru
X	0	Ping
x	0	Odpověď na ping

Tabulka 5.1: Seznam implementovaných příkazů komunikačního protokolu

5.2 Činnost aplikace

Samotné spuštění aplikace nevyvolá žádnou činnost. Aplikace zobrazí všechny ovládací prvky, které jsou neaktivní. Nejprve je nutné nastavit přístupové údaje k robotu – jeho jméno a heslo. Toto nastavení se provede stiskem tlačítka 1 z obrázku 5.2. Uživateli je zobrazeno dialogové okno pro vyplnění jména a hesla robotu. Po zadání přístupových údajů je možné zahájit samotné připojení. Přístupové údaje jsou uloženy do paměti aplikace (viz sekce 5.3.2).

Při procesu připojování je zapnuta služba Wi-Fi. Při přechodu do režimu pauzy je připojený robot odpojen, Wi-Fi však zůstává zapnuta. V případě ukončení aplikace je Wi-Fi síť vypnuta a záznam o ní ze seznamu Wi-Fi sítí je vymazán. Aplikace v zařízení vrátí všechny změny, které provedla.

5.2.1 Připojení

Připojení se provede stiskem tlačítka 2 z obrázku 5.2. Při připojování je nejprve zkontrolováno zapnutí služby Wi-Fi. Poté jsou z paměti vyčteny přístupové údaje



Obrázek 5.2: Horní stavová a ovládací lišta aplikace

k robotu. V případě, že přístupové údaje nebyly nastaveny, není možné se na robot připojit. Tato skutečnost je uživateli oznámena a připojování je ukončeno. V případě úspěšného vyčtení přístupových údajů z paměti je spuštěna nová úloha, které jsou přístupové údaje předány. Protože síťová komunikace nemůže probíhat v UI vlákne aplikace¹, probíhá připojování k Wi-Fi síti v novém vlákne. V tomto vlákne probíhá nastavení Wi-Fi konfigurace s přístupovými údaji jako hlavními parametry. Následuje pokus o připojení s danou konfigurací. Protože zapínání Wi-Fi, připojování a autentizace trvá několik vteřin, vlákno má nastaveno limit, po který čeká, jestli se připojení podaří. V případě úspěchu je zkontrolována správnost připojení. Při neúspěšném připojení je tato skutečnost oznámena uživateli.

V případě úspěchu připojení je vytvořen `Socket`, který odešle žádost do počítače Raspberry Pi, kde proběhne přijetí žádosti `Socketu` a otevření komunikačních kanálů (viz. 3.6). Dále je otevřeno HTTP spojení, které je využito pro přijímání videa `MjpegStream`. Pokud je `Socket` a HTTP spojení úspěšné, vlákno sloužící pro síťovou komunikaci se ukončí s úspěchem a činnost přejde opět na UI vlákno aplikace. Ta je uvědoměna o úspěchu připojení a oznámí ho uživateli změnou barvy horní lišty na zelenou. Zároveň se aktivují ovládací prvky aplikace, tlačítka i posuvníky.

5.3 Popis komponent

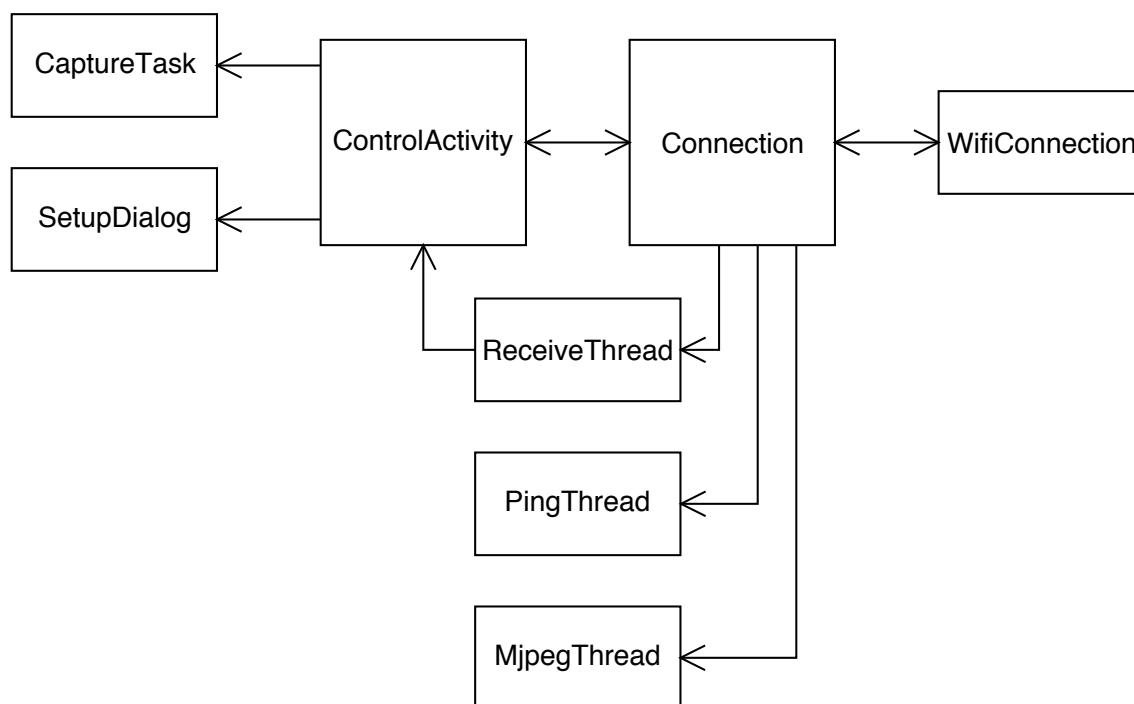
Blokové schéma tříd obsažených v aplikaci je na obrázku 5.3. Obrázek ilustruje souvislosti mezi jednotlivými třídami a neobsahuje pomocné třídy.

5.3.1 Popis tříd

`ControlActivity` je potomek třídy `Activity`. Jedná se o jedinou `Activity` v aplikaci. Grafická a komunikační část aplikace jsou přísně odděleny. Třída `ControlActivity` má na starosti vzhled aplikace. Obsahuje veškeré ovládací prvky a metody pro jejich manipulaci. `Connection` je třída, která zajišťuje komunikační část aplikace. Uchovává stav aplikace a zároveň obsahuje veškeré funkce pro síťovou komunikaci. Vytváří i ukončuje vlákna `WifiConnection`, `ReceiveThread`, `PingThread` a `MjpegThread`. Instance třídy `Connection` je vytvořena při spuštění aplikace.

`WifiConnection` je potomek třídy `AsyncTask`. Zjednodušeně řečeno se jedná o úlohu spouštěnou v novém vlákne. Tato třída obstarává připojení robotu, definuje celý proces připojování i ověřování. Úloha při neúspěchu připojení informuje operátora

¹UI vlákno je hlavní vlákno aplikace.



Obrázek 5.3: Blokové schéma tříd aplikace

zprávou `Toast`, která obsahuje i informaci, ve které fázi připojování došlo k chybě. Při úspěšném připojení jsou vytvořeny instance tříd `Socket` a `HttpURLConnection`. Ty jsou předány třídě `Connection`, která spustí tři separátní vlákna zajišťující běh aplikace – `ReceiveThread`, `PingThread` a `MjpegThread`.

`ReceiveThread` je vlákno, které zajišťuje přijímání zpráv od robotu. Při vytváření instance je vláknu předán vstupní kanál `Socketu`, který se vytváří při připojování. Přijatá zpráva je rovnou vyhodnocena a obsah zprávy vykonán v `ControlActivity`. K přístupu ke grafickým prvkům je oprávněno pouze UI vlákno aplikace. Aby vlákno `ReceiveThread` získalo přístup ke grafickým prvkům, je nutné vytvořit instanci třídy, která implementuje rozhraní `Runnable`, a tu zařadit do UI vlákna.

`PingThread` je vlákno, které v intervalu jedné vteřiny odesílá do robotu ping ve formě velkého písmene „X“ bez argumentů.

`MjpegThread` je vlákno, které zpracovává přijímané video. Parametry při vytváření instance jsou vstupní kanál HTTP spojení, které je ustaveno při připojování, a odkaz na grafický prvek, kam se má video vykreslovat.

`CaptureTask` je potomek třídy `AsyncTask`. Jedná se o úlohu, která ze streamovacího serveru robotu stáhne právě zobrazovaný obrázek a uloží ho do paměti zařízení (viz sekce 5.3.2). Obrázek je uložen do složky, která se jmenuje stejně jako robot, s názvem „ddMMyyyyhhmss.jpg“, kde „ddMMyyyyhhmss“ je časová a datová značka.

`SetupDialog` je dialogové okno sloužící k zadání přístupových údajů k robotu. Je zobrazeno stiskem tlačítka 1 z obrázku 5.2.

5.3.2 Paměť

Aplikace používá dva druhy paměti pro uložení dat – `Shared Preferences` a `External Storage`. Více o možnostech uložení dat se lze dočíst v [33].

`Shared Preferences` je jednoduchá datová struktura ve formě klíč–hodnota, která je určena pro uložení primitivních veličin. Do této struktury jsou uloženy přístupové údaje robotu jako datový typ `String`. Takto uložené hodnoty nejsou smazány při zavření aplikace. Při opětovném spuštění aplikace jsou načteny a uživatel je tedy nemusí zadávat znovu.

`External Storage` je paměť zařízení, nejedná se o vlastní paměť aplikace, ale o sdílenou paměť přístupnou libovolnou aplikací. Většinou se jedná o paměť `Flash`. Do této paměti jsou ukládány obrázky vyfocené robotem, konkrétně do systémové složky obrázků, která je na většině zařízení `/storage/emulated/0/Pictures`. Snímky jsou pak dostupné v souborové struktuře zařízení. Získání adresy do systémové složky obrázků je možné kódem

```
Environment.getExternalStoragePublicDirectory(Environment
    .DIRECTORY_PICTURES);
```

5.3.3 Broadcast Receiver

Při připojení je zaregistrován `Broadcast Receiver` jako

```
context.registerReceiver(receiver, new IntentFilter("
    android.net.wifi.STATE_CHANGE"));
```

Takto zaregistrovaný receiver hlídá stav Wi-Fi připojení. Jakmile dojde k odpojení Wi-Fi z libovolné příčiny, vyvolá odpojení robotu a bezpečné ukončení činnosti a vláken aplikace. Pak dojde k jeho odregistrování

```
context.unregisterReceiver(receiver);
```

5.3.4 Oprávnění

Operační systém Android chrání přístup ke zdrojům a aplikaci povolí jen to, k čemu má oprávnění. Tato oprávnění jsou zapsaná v souboru `AndroidManifest.xml`. Aplikace má následující oprávnění:

- `INTERNET`
- `CHANGE_NETWORK_STATE`
- `ACCESS_WIFI_STATE`
- `CHANGE_WIFI_STATE`
- `WRITE_EXTERNAL_STORAGE`

5.4 Ovládání

Veškeré ovládání aplikace je prováděno přes grafické prvky. Na obrázku 5.4 je zachycen vzhled aplikace při připojeném robotu. Aplikace je uzpůsobena pro ovládání dvěma rukama.

V horní části displeje se nachází stavová a ovládací lišta. Ta svou barvou informuje o stavu připojení/nepřipojení robotu k ovladači. Po stranách se zobrazují rychlosti motorů. Dále je zde informační ikona stavu akumulátoru v robotu. Stav nabití akumulátoru je přibližně určován z velikosti napětí, kterou měří voltmetr, a znalosti vybíjecí charakteristiky článků akumulátoru. Jedná se o přibližný údaj. Vedle ikony akumulátoru je údaj o okolní teplotě robotu a tlačítka pro nastavení přístupových údajů a připojení robotu. V případě, že je robot aktuálně připojen, slouží tlačítka připojení k odpojení robotu.

Pro ovládání motorů disponuje aplikace dvojicí posuvníků. Ty jsou umístěny po stranách aplikace, aby při držení zařízení v obou rukách byly tyto posuvníky pohodlně ovladatelné dvěma palci. Posuvníky se nacházejí ve středové, nulové poloze. Při vychýlení směrem nahoru se příslušný motor rozjede vpřed, při vychýlení dolů se motor rozjede vzad. Při uvolnění posuvníku dojde k jeho automatickému nastavení do nulové polohy a příslušný motor se zastaví. Aktuální rychlost motorů je zobrazována nad posuvníky v horní liště. Maximální rychlost motorů je 200, minimální rychlost je 20. Jednotka rychlosti je počet otáček za minutu.

Mezi dvojicí posuvníků se nachází velké okno. V jeho horní části je zobrazováno video z kamery. Pod videem se nachází trojice tlačítek, která slouží pro otáčení servomotoru držícího kameru. Tímto způsobem lze otáčet snímané video do stran, stiskem prostředního tlačítka se kamera nastaví do středové polohy. Pod ovládacím kamery se nachází výšeč, která je prozkoumávána dálkoměry. Naměřené překážky se do ní zobrazují jako bílé body. Naměřené hodnoty jsou uváděny v centimetrech, k lepší orientaci jsou vykresleny i vodící čáry. Čísla po stranách udávají jednotky centimetrů a slouží k determinaci vzdálenosti zobrazené překážky. V dolní části displeje je umístěna trojice tlačítek. První umožňuje zapnutí/vypnutí světel, druhé zachycení a uložení právě viděného snímku na video a třetí ke spuštění klaksonu.

5.4.1 Grafické prvky

Grafické rozvržení (layout) třídy `Activity` je uloženo jako `.xml` dokument, podle zvyklostí systému Android. Jsou v něm definovány všechny grafické prvky včetně jejich umístění a dalších vlastností. Aplikace obsahuje dvě grafická rozvržení – jedno pro zařízení s velkým displejem (tablety), druhé pro zařízení s malým displejem (mobilní telefony).

Aplikace obsahuje jedno rozvržení pro dialogové okno, které slouží pro zadávání přístupových údajů.

Většina grafických ovládacích prvků jsou obrázky (`ImageView`) opatřené funkcí kliknutí. Posuvníky pro řízení motorů jsou prvky označované jako `SeekBar`, které jsou otočené o 90° do svislé polohy. Systém Android takový prvek neobsahuje, proto

byla použita upravená verze označená jako `VerticalSeekBar` [34]. Video je zobrazováno v grafickém prvku `SurfaceView`, což je specializovaná kreslicí plocha. Pro zobrazování videa byla použita již napsaná třída dostupná na internetu [35].

Pro vykreslování překážek existuje třída `DrawingView`, která je přímým dědicem základního prvku `View`. Pro vykreslování bodů slouží funkce `addPoint()`, která přepočítá polární souřadnice přijatého bodu na souřadnice kartézské a poté tento bod zobrazí. Body se vykreslují do třídy `Canvas`.

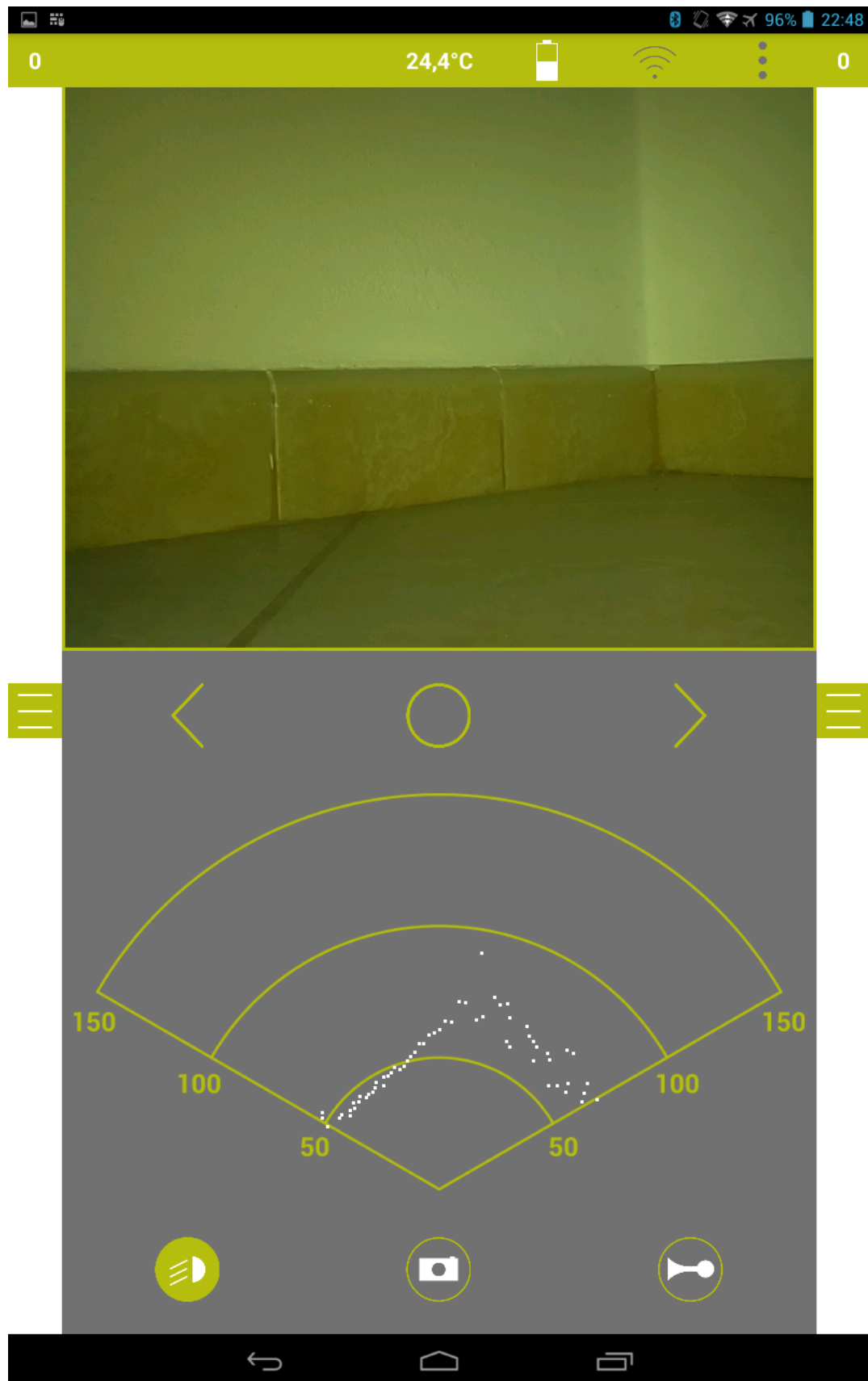
5.5 Administrátorský mód

Přístup k robotu je možný i bez ovládací aplikace. Toho je docíleno použitím univerzálních komponent. Připojit se k robotu je možné i s pouhou znalostí přístupových údajů. Protože robot vytváří Wi-Fi síť, stačí se připojit na síť, která má SSID shodné se jménem robotu. Po zadání hesla je vytvořeno spojení. Protože je Raspberry Pi linuxový systém, poskytuje SSH server. Pro připojení lze použít libovolný SSH klient. Po zadání přístupových údajů počítače Raspberry Pi získá operátor plnou kontrolu nad robotem. Může např. ovládat robot přímo pomocí příkazů z komunikačního protokolu (viz sekce 5.1), aktualizovat software desky Arduino, změnit nastavení Wi-Fi sítě, spravovat spuštěné aplikace, přímo přistupovat ke kameře, atd. Přímé posílání příkazů do robotu v SSH klientu vypadá takto:

```
echo „L,1“ > /dev/ttyACM0
```

Tímto příkazem dojde k rozsvícení světel. Místo `L,1` je možné použít libovolný příkaz podle komunikačního protokolu. Dále je možné použít libovolných Shell příkazů.

Administrátorský mód byl při konstrukci robotu použit pro účely ladění a vývoje. V tomto režimu lze celý robot poškodit, proto by měl být tento administrátorský mód využíván jen zkušeným operátorem.



Obrázek 5.4: Snímek aplikace

ZÁVĚR

Implementovaná ovládací aplikace je uživatelsky příjemná – má srozumitelné ovládání, ovládací tlačítka jsou umístěna přehledně a pohodlně. Aplikace se dá bez problémů použít jak na zařízeních s větším displejem (tablety), tak na zařízeních s displejem menších rozměrů (chytré telefony). Aplikace umožňuje připojení robotu stiskem jednoho tlačítka, celý proces připojování je automatický.

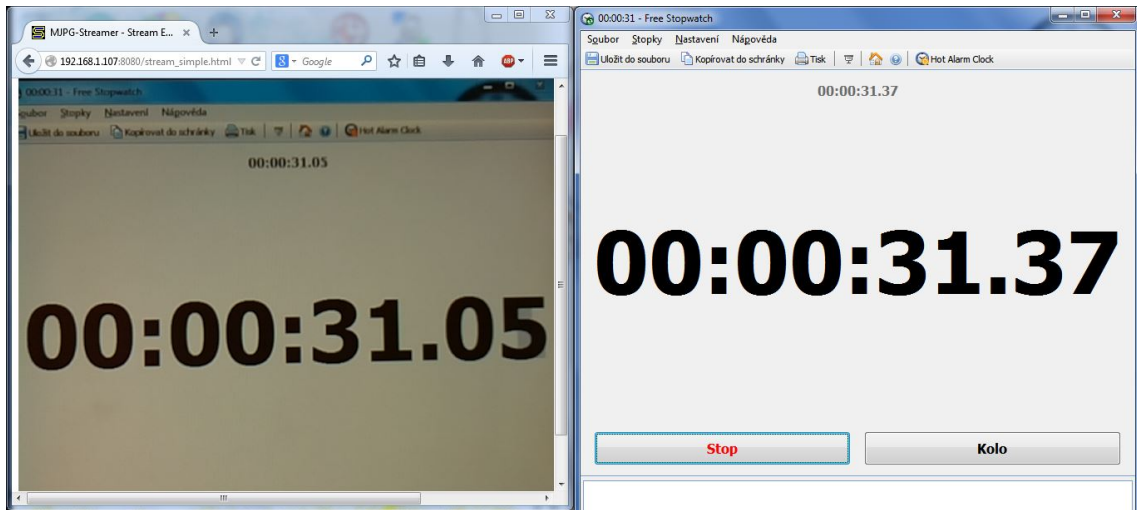
Pro hodnocení kvality implementovaného řešení byl proveden test latence přenášeného videa. Na monitoru byly zobrazeny stopky, které robot snímal a video bylo zobrazeno na tomtéž monitoru v druhém okně. Při sejmutí obrazu v jeden moment bylo možné zachytit aktuální čas stopek a videa. Prostým odečtením zobrazených časů byla určena latence. Na obrázku 6.1 je zachycen snímek monitoru, v pravém okně jsou zobrazené stopky a v levém okně je přenášené video. Naměřená latence byla 300 ms – 500 ms.

Dalším testem bylo měření doby přenosu textových dat. Tento test byl realizován porovnáním časových značek odeslaných a přijatých příkazů. Doba přenosu textových dat z aplikace do robotu byla určena na cca 3 ms.

Při konstrukci robotu se volba linuxového systému ukázala jako vhodná. Dálkoměry měří spolehlivě pouze do vzdálenosti jednoho metru, měření vzdálenějších překážek je už nepřesné. Takovéto chování vyplývá z převodní charakteristiky dálkoměru. Použité akumulátory jsou dostatečně výkonné pro provoz robotu po dobu cca 10 hodin.

Robot disponuje systémem pro zastavování před překážkou, k čemuž využívá data z dálkoměrů. K zastavení robotu dojde, pokud je naměřená vzdálenost menší než 5 cm. Dále je robot vybaven ochranou před ztrátou signálu. K tomu slouží pravidelné odesílání příkazu „ping“ ve vteřinových intervalech. Pokud tento není přijat po dobu dvou vteřin, robot se zastaví na místě.

Wi-Fi adaptér umožňuje připojení více zařízení k robotu, ovládat robot však může pouze jedno zařízení. Při připojeném ovladači neumožňuje obslužný program robotu `host.java` přijetí dalšího socketu. To je možné až při odpojení momentálně připojeného ovladače. Připojení dalšího zařízení zároveň neodpojí zařízení právě připojené. Připojení více zařízení je povolené kvůli možnosti ladění robotu při současném



Obrázek 6.1: Měření latence přenosu videa

zachování plné ovladatelnosti. Jinou možností využití připojení dalších zařízení je možnost sledování streamu, který je na ovládání nezávislý.

Využití výpočetního výkonu počítače Raspberry Pi dosahuje jednotek procent, zbývá tedy velké množství potenciálního výkonu, který je využitelný např. pro analýzu obrazu.

Jedna z možností vylepšení aplikace spočívá v rozšíření možností nastavení různých parametrů, se kterými robot pracuje. Další možností je změna ovládání pohybu robotu na polohování mobilního zařízení (princip volantu). Pro tento typ řízení by se využil akcelerometr.

LITERATURA

- [1] Plán využití rádiového spektra. *Český telekomunikační úřad* [online]. 2014 [cit. 2014-05-15]. Dostupné z: <http://www.ctu.cz/predpisy-a-opatreni/plan-vyuziti-radioveho-spektra.html>
- [2] IEEE 802.11 In: *Wikipedia: the free encyclopedia* [online]. c2013 [cit. 2014-05-15]. Dostupné z: http://en.wikipedia.org/wiki/IEEE_802.11
- [3] WiMAX In: *Wikipedia: the free encyclopedia* [online]. c2014 [cit. 2014-05-28]. Dostupné z: <http://en.wikipedia.org/wiki/WiMAX>
- [4] Bluetooth In: *Wikipedia: the free encyclopedia* [online]. c2014 [cit. 2014-12-29]. Dostupné z: <http://en.wikipedia.org/wiki/Bluetooth>
- [5] ZigBee In: *Wikipedia: the free encyclopedia* [online]. c2013 [cit. 2014-10-26]. Dostupné z: <http://en.wikipedia.org/wiki/ZigBee>
- [6] Mapa pokrytí. *Vodafone Czech Republic* [online]. 2014 [cit. 2014-05-30]. Dostupné z: <http://www.vodafone.cz/mapa-pokryti/>
- [7] Mapa pokrytí. *O2 Czech Republic* [online]. 2014 [cit. 2014-05-30]. Dostupné z: http://www.o2.cz/osobni/199436-mapa_pokryti_a_prodejen/
- [8] Mapa pokrytí. *T-Mobile Czech Republic* [online]. 2014 [cit. 2014-05-30]. Dostupné z: <http://www.t-mobile.cz/podpora/mapa-pokryti>
- [9] STMICROELECTRONICS. *L298: Dual Full Bridge Driver* [datasheet]. c2000. Dostupné z: <http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/CD00000240.pdf>
- [10] NOVÁK, Petr. *Mobilní roboty: pohony, senzory, řízení*. Vyd. 1. Praha: BEN - technická literatura. 2004. 247 s. ISBN 80-730-0141-1.
- [11] CONRAD. *Velkoobchod* [katalog]. 2014.
- [12] PANASONIC CORPORATION. *NCR18650A* [datasheet]. c2010. Dostupné z: <https://industrial.panasonic.com/www-data/pdf2/ACA4000/ACA4000CE254.pdf>

- [13] *Arduino Micro* [online]. 2014. Dostupné z: <http://arduino.cc/en/Main/arduinoBoardMicro>
- [14] ATMEL CORPORATION. *ATmega32u4* [datasheet]. c2014. Dostupné z: http://www.atmel.com/Images/Atmel-7766-8-bit-AVR-ATmega16U4-32U4_Datasheet.pdf
- [15] BEAUREGARD, Brett. *Arduino PID Library* [online]. Verze 1.1.0. 2012 [cit. 2014-03-22]. Dostupné z: <https://github.com/br3ttb/Arduino-PID-Library/>
- [16] Serial. *Arduino* [online]. 2014 [cit. 2014-03-21]. Dostupné z: <http://arduino.cc/en/Reference/Serial>
- [17] Servo library. *Arduino* [online]. 2014 [cit. 2014-04-10]. Dostupné z: <http://arduino.cc/en/Reference/Servo>
- [18] MARGOLIS, Michael. *Arduino Cookbook*. Vyd. 2. O'Reilly Media. 2012. 724 s. ISBN 1-4493-1387-6.
- [19] Automatické řízení 2014. *Moodle Katedry řídicí techniky* [online]. 2014 [cit. 2014-10-05]. Dostupné z: <https://moodle.dce.fel.cvut.cz/course/view.php?id=68>
- [20] *Raspberry Pi* [online]. 2014. Dostupné z: <http://www.raspberrypi.org/>
- [21] *Raspbian* [online]. 2014. Dostupné z: <http://raspbian.org/>
- [22] Turn Your Raspberry Pi Into a WiFi Hotspot with Edimax Nano USB EW-7811Un (RTL8188CUS chipset). *Dave Conroy* [online]. 2013 [cit. 2014-05-30]. Dostupné z: <http://www.daveconroy.com/turn-your-raspberry-pi-into-a-wifi-hotspot-with-edimax-nano-usb-ew-7811un-rtl8188cus-chipset/>
- [23] *MJPEG-streamer* [online]. Verze r63. 2008 [cit. 2014-10-12] Dostupné z: <http://sourceforge.net/projects/mjpg-streamer/>
- [24] Streaming mjpg from Raspberry Cam. *Raspberry Pi Forum* [online]. 2013 [cit. 2014-10-14]. Dostupné z: <http://www.raspberrypi.org/forums/viewtopic.php?f=43&t=45178&start=75>
- [25] SHARP CORPORATION. *GP2Y0A41SK0F* [datasheet]. c2005. Dostupné z: http://www.sharp.co.jp/products/device/doc/opto/gp2y0a41sk_e.pdf
- [26] SHARP CORPORATION. *GP2Y0A02YK0F* [datasheet]. c2006. Dostupné z: http://www.sharpsma.com/webfm_send/1487
- [27] VISHAY. *NTCLE100E3* [datasheet]. c2012. Dostupné z: <http://www.vishay.com/docs/29049/ntcle100.pdf>

-
- [28] Comparison of mobile operating systems In: *Wikipedia: the free encyclopedia* [online]. c2014 [cit. 2014-10-02]. Dostupné z: http://en.wikipedia.org/wiki/Comparison_of_mobile_operating_systems
- [29] FRUMUSANU, Andrei. A Closer Look at Android RunTime (ART) in Android L. In: *ANANDTECH* [online]. 2014 [cit. 2014-10-02]. Dostupné z: <http://anandtech.com/show/8231/a-closer-look-at-android-runtime-art-in-android-l/>
- [30] Google Android lords over 85 pct of smartphone OS market share, Apple's iOS distant second: IDC. In: *The Financial Express* [online]. 2014 [cit. 2014-10-01]. Dostupné z: <http://archive.financialexpress.com/news/google-android-lords-over-85-pct-of-smartphone-os-market-share-apple-s-ios-distant-second-idc/1284435>
- [31] Dashboards. *Android Developers* [online]. 2014 [cit. 2014-10-01]. Dostupné z: <https://developer.android.com/about/dashboards/index.html>
- [32] Application Fundamentals. *Android Developers* [online]. 2014 [cit. 2014-10-05]. Dostupné z: <https://developer.android.com/guide/components/fundamentals.html>
- [33] Storage Options. *Android Developers* [online]. 2014 [cit. 2014-11-16]. Dostupné z: <https://developer.android.com/guide/topics/data/data-storage.html>
- [34] Working Vertical SeekBar for Android. *Hackskrieg* [online]. 2012 [cit. 2014-04-16]. Dostupné z: <https://hackskrieg.wordpress.com/2012/04/20/working-vertical-seekbar-for-android/>
- [35] SimpleMjpegView. *Bitbucket* [online]. 2013 [cit. 2014-10-21]. Dostupné z: <https://bitbucket.org/neuralassembly/simplemjpegview/>

Přílohy

OBSAH PŘILOŽENÉHO CD

- arduino.ino – Zdrojový kód programu pro Arduino
- init.sh – Inicializační skript pro Raspberry Pi
- host.java – Zdrojový kód komunikačního programu pro Raspberry Pi
- slidil – Zdrojové kódy ovládací aplikace; projekt pro Android Studio
- bp.pdf – Tato Bakalářská práce ve formátu pdf
- obsah.pdf – Obsah tohoto CD