



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

RUČNÍ DÁLKOVÝ OVLADAČ PRO ROBOT PERSEUS

OPERATOR'S STATION FOR PERSEUS MOBILE ROBOT

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Marek Sabó

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. František Burian, Ph.D.

BRNO 2019

Diplomová práce

magisterský navazující studijní obor **Kybernetika, automatizace a měření**

Ústav automatizace a měřicí techniky

Student: Bc. Marek Sabó

ID: 171152

Ročník: 2

Akademický rok: 2018/19

NÁZEV TÉMATU:

Ruční dálkový ovladač pro robot Perseus

POKyny PRO VYPRACOVÁNÍ:

1. Seznamte se s robotem Perseus, ovládacím protokolem použitých serv Robotis RX-64 a protokoly, které se používají pro přenos videa.
2. Seznamte se s principy tvorby grafického uživatelského rozhraní pro mobilní roboty, a s pákovými ovladači, které jsou vhodné pro ovládání mobilního robotu.
3. Seznamte se s Raspberry Pi a kompatibilním operačním systémem.
4. Navrhněte aplikaci, která je určena pro ovládání mobilního robotu. Aplikace by měla umět zobrazovat videosignál z robotu, vybrané stavové veličiny a měla by ovládat pohybové funkce tohoto mobilního robotu.
5. Aplikaci implementujte do Raspberry Pi a vyzkoušejte na dodaném hardware.

DOPORUČENÁ LITERATURA:

WARNER, Timothy L. Hacking Raspberry Pi. Indianapolis, Indiana: Que, 2013. ISBN 978-0789751560.

LAL, Rajesh. Digital design essentials: 100 ways to design better desktop, web, and mobile interfaces. Beverly, MA: Rockport Publishers, [2013]. ISBN 9781592538034.

JOHNSON, Jeff. Designing with the mind in mind simple: simple guide to understanding user interface design guidelines. Second edition. Boston: Elsevier, Morgan Kaufmann is an imprint of Elsevier, [2014]. ISBN 978-0124079144.

Termín zadání: 4.2.2019

Termín odevzdání: 13.5.2019

Vedoucí práce: Ing. František Burian, Ph.D.

Konzultant:

doc. Ing. Václav Jirsík, CSc.
předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Táto práca sa zaoberá problematikou návrhu a realizácie aplikácie pre ovládanie mobilného robota. V úvodnej časti je analyzovaná použitá platforma GEARS-SMP, princíp funkčnosti protokolu pre riadenie servomotorov Dynamixel, formát M-JPEG a štandard H.264. Ďalej sa práca venuje analýze tvorby užívateľského rozhrania v robotických aplikáciách, dostupných možnostiach ovládacích zariadení a použitému hardvéru v diaľkovom ovládači. Nasledujúca časť je zameraná na návrh robotickej aplikácie, najmä grafického užívateľského rozhrania a virtuálneho priehľadového displeja a následnou implementáciou vytvoreného návrhu do zariadenia Raspberry Pi. Na záver práca popisuje realizované softvérové riešenie a porovnáva výslednú aplikáciu s vytvoreným návrhom.

ABSTRACT

This thesis deals with the design and implementation of application for control of mobile robot. In the introductory section is discussed used platform GEARS-SMP, the principle of functionality of protocol for servo motors control, format M-JPEG and standard H.264. Further work is dedicated to analysis of designing user interface in robotic applications, available options for control devices and hardware used in remote controller. The following part focuses on the design of robotic application, especially on graphic user interface and virtual head-up display and follow-up implementation of created application in Raspberry Pi. In the end, thesis describes implemented software solution and compares resulting application with the created design.

KLÚČOVÉ SLOVÁ

Platforma GEARS-SMP, Mobilný robot Perseus, Robotis RX-64, M-JPEG, H.264, Grafické užívateľské rozhranie, HUD, Diaľkový ovládač, Raspberry Pi, Qt, OpenGL

KEYWORDS

Platform GEARS-SMP, Mobile robot Perseus, Robotis RX-64, M-JPEG, H.264, Graphic user interface, HUD, Remote controller, Raspberry Pi, Qt, OpenGL

BIBLIOGRAFICKÁ CITÁCIA

SABÓ, Marek. *Ruční dálkový ovladač pro robot Perseus* [online]. Brno, 2019 [cit. 2019-05-10]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/119397>. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce František Burian.

ČESTNÉ PREHLÁSENIE

„Prehlasujem, že moju diplomovú prácu na tému Ruční dálkový ovladač pro robot Perseus som vypracoval samostatne pod vedením vedúceho diplomovej práce a s použitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej diplomovej práce ďalej prehlasujem, že v súvislosti s vytvorením tejto diplomovej práce som neporušil autorské práva tretích osôb, hlavne som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a som si plne vedomý následkov porušenia ustanovení § 11 a nasledujúcich autorského zákona č. 121/2000 Zb., vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovení časti druhej, hlavy VI. diel 4 Trestného zákonníka č. 40/2009 Zb.“

V Brne dňa 13.05.2019

.....

Bc. Marek Sabó

POĎAKOVANIE

Týmto by som sa chcel poďakovať Ing. Františkovi Burianovi, Ph.D. za odborné vedenie diplomovej práce, ochotu, cenné rady, trpezlivosť a v neposlednom rade čas, ktorý mi venoval pri tvorbe tejto práce.

V Brne dňa 13.05.2019

.....

Bc. Marek Sabó

OBSAH

1	ÚVOD	12
2	PLATFORMA GEARS-SMP	13
2.1	Mobilný robot Perseus	14
2.1.1	Sieťová kamera	15
2.1.2	Smerovač.....	15
2.1.3	Servomotory pre polohovanie kamery.....	16
2.1.4	Doska plošných spojov pre riadenie pohonu	16
2.1.5	Princíp fungovania mobilného robota	16
2.2	Protokol pre riadenie servomotorov Robotis Dynamixel RX-64 a EX-106	17
2.2.1	Riadiaca tabuľka.....	18
2.2.2	Inštrukčný paket	20
2.2.3	Stavový paket.....	22
2.3	Protokoly pre prenos videa	22
2.3.1	Formát Motion JPEG	23
2.3.2	Štandard H.264/MPEG-4-AVC.....	24
3	GRAFICKÉ UŽIVATEĽSKÉ ROZHRAŇIE	27
3.1	Objekty.....	27
3.2	Text.....	28
3.3	Farby.....	28
3.4	Periférne videnie.....	28
3.5	Priehľadový displej.....	29
3.6	Užívateľské rozhranie robotických aplikácií.....	29
4	OVLÁDACIE ZARIADENIA	32
4.1	Pákový ovládač	32
4.2	Gamepad	33
4.3	Iné druhy ovládacích zariadení.....	33
4.4	Ovládač pre mobilný robot Perseus.....	34
5	RASPBERRY PI	36
5.1	Operačný systém.....	39
5.1.1	Raspbian.....	40
5.2	Rozšírenia Raspberry Pi	40
5.2.1	Raspberry Pi displej.....	40
5.2.2	Wi-Fi modul	41
6	NÁVRH APLIKÁCIE	43
6.1	Základné úkony	43
6.1.1	Zapnutie zariadení.....	43
6.1.2	Vypnutie zariadení	44
6.2	Návrh užívateľského rozhrania	44
6.2.1	Priehľadový displej	45
6.2.2	Menu.....	46
6.2.3	Stavové hlásenia	50
6.3	Popis ovládacích tlačidiel.....	55

7	IMPLEMENTÁCIA APLIKÁCIE	56
7.1	Rozhranie pre programovanie aplikácií OpenGL a EGL.....	56
7.2	Príprava platformy Raspberry Pi.....	57
7.3	Grafická sada programových nástrojov Qt a vývojové prostredie.....	59
7.4	Softvérová realizácia	62
7.4.1	Modul main.c a trieda MainWidget	62
7.4.2	Trieda Hud.....	65
7.4.3	Trieda PerseusControl.....	66
7.4.4	Trieda RasPi.....	68
7.5	Výsledná aplikácia.....	69
8	ZÁVER.....	73
	ZOZNAM POUŽITEJ LITERATÚRY	74
	ZOZNAM PRÍLOH.....	79

ZOZNAM OBRÁZKOV

Obr. 1: Platforma GEARS-SMP [1]	13
Obr. 2: Typy platforiem GEARS-SMP [2]	13
Obr. 3: Mobilný robot Perseus	14
Obr. 4: Sieťová kamera AXIS 1214-E [3]	15
Obr. 5: Smerovač Mikrotik RB411AH (vľavo) a sieťový adaptér Mikrotik RouterBOARD R52Hn miniPCI (vpravo) [4] [5]	15
Obr. 6: Servomotor Robotis Dynamixel EX-106 (vľavo) a RX-64 (vpravo) [6] [7]..	16
Obr. 7: Blokový diagram princípu fungovania robota Perseus	17
Obr. 8: Riadenie komunikácie [8]	18
Obr. 9: GOP [17].....	26
Obr. 10: Usporiadané GOP [17]	26
Obr. 11: HUD systému CASSANDRA pre mobilného robota Orpheus [21].....	30
Obr. 12: HUD systému CASSANDRA pre vzdušnú jednotku Uranus [23].....	31
Obr. 13: Joystick Saitek Cyborg X [24].....	32
Obr. 14: Gamepad Razer Raiju Ultimate [25].....	33
Obr. 15: Ovládač pre RC autá Futaba 4PV [26]	34
Obr. 16: Ovládač pre vzdušné modely Futaba 18MZ WC [27]	34
Obr. 17: Diaľkový ovládač pre robot Perseus	35
Obr. 18: Raspberry Pi 2 model B V1.1 [30]	36
Obr. 19: Rozloženie GPIO pinov [31]	38
Obr. 20: Raspberry Pi Touch Display [35].....	41
Obr. 21: USB adaptér Edimax EW7811Un [37].....	41
Obr. 22: Umiestnenie vypínača robota.....	43
Obr. 23: Umiestnenie vypínača ovládača	44
Obr. 24: Navrhnutý HUD	46
Obr. 25: Zobrazené menu s výberom položky „Network“	47
Obr. 26: Okno pre nastavenie obrazu s výberom položky „HUD transparency“	47
Obr. 27: Okno pre nastavenie siete s výberom položky „Signal strength limit“	48
Obr. 28: Okno nastavení robota s výberom položky „Servo (H) moving speed“	49
Obr. 29: Hlásenie o čakaní na pripojenie	50
Obr. 30: Varovanie o slabom signáli alebo veľkej stratovosti paketov	50
Obr. 31: Hlásenie o strate signálu a čakaní naň	51
Obr. 32: Hlásenie o strate pripojenia	51
Obr. 33: Hlásenie o slabej batérii ovládača	52
Obr. 34: Hlásenie o preťažení pravého zadného motoru	53
Obr. 35: Hlásenie o prehriatí vertikálneho servo motoru	53
Obr. 36: Hlásenie o prepätí batérie robota	54

Obr. 37: Hlásenie o nízkom napätí batérie robota.....	54
Obr. 38: Označenie tlačidiel ovládača	55
Obr. 39: Hardvérová akcelerácia pomocou OpenGL	57
Obr. 40: Vývojové prostredie Qt Creator.....	60
Obr. 41: Inicializačný sled udalostí po spustení aplikácie	63
Obr. 42: Hierarchia výstupov a vstupov tried pri prehrávaní videosignálu	63
Obr. 43: Vnútoraná logika triedy MainWidget.....	64
Obr. 44: Vnútoraná logika triedy MainWidget.....	65
Obr. 45: Trieda Hud a jej najdôležitejšie funkcie	66
Obr. 46: Trieda PerseusControl a jej najdôležitejšie funkcie	67
Obr. 47: Vnútoraná logika triedy PerseusControl.....	68
Obr. 48: Trieda RasPi a jej najdôležitejšie funkcie	69
Obr. 49: Vzhľad priehľadového displeja výslednej aplikácie	70
Obr. 50: Náhľad na ponúkané menu aplikácie	71
Obr. 51: Implementované stavové hlásenia	72

ZOZNAM TABULIEK

Tab. 1: Technické parametre použitej platformy GEARS-SMP [1] [2]	14
Tab. 2: RAM riadiacej tabuľky RX-64/EX-106(+modré) [6] [7]	19
Tab. 3: EEPROM riadiacej tabuľky RX-64/EX-106(+modré) [6] [7]	20
Tab. 4: Štruktúra inštrukčného paketu [8].....	21
Tab. 5: Definície inštrukcií [8]	21
Tab. 6: Štruktúra stavového paketu [8]	22
Tab. 7: Význam chybových bitov [8]	22
Tab. 8: Technická špecifikácia Raspberry Pi 2 model B V1.1 [28] [30]	37
Tab. 9: Rozloženie funkcií na GPIO pinoch [31].....	38
Tab. 10: Technická špecifikácia Wi-Fi adaptéru EDIMAX [37].....	42
Tab. 11: Popis funkcií jednotlivých tlačidiel.....	55

1 ÚVOD

Mobilné roboty patria v súčasnej dobe medzi veľmi významnú časť výskumu a vývoju univerzít zameraných na robotické aplikácie. Dôležitým prvkom týchto aplikácií je aj užívateľské rozhranie, ktoré musí byť jednoduché a účelné.

Táto diplomová práca je pokračovaním semestrálnej práce a zaoberá sa problematikou návrhu a realizácie robotickej aplikácie pre ovládanie mobilného robota Perseus. Cieľom je zoznámenie sa s platformou GEARS-SMP použitou pri tvorbe robota Perseus, samotným robotom, štandardmi pre prenos videa, rozšírením poznatkov z oblasti návrhu grafického užívateľského rozhrania, prístupných ovládacích zariadení, použitým hardvérom a softvérom a na základe týchto znalostí následne navrhnuť a realizovať robotickú aplikáciu pre riadenie mobilného robota Perseus.

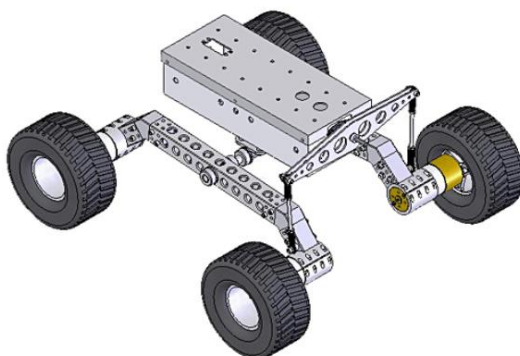
V rešeršnej časti práce je analyzovaná použitá platforma GEARS-SMP, použitý hardvér v robote Perseus a je vytvorený stručný popis princípu fungovania robota. Ďalej je vysvetlený princíp fungovania protokolu pre riadenie servomotorov Robotis Dynamixel RX-64 a EX-106 použitých pre polohovanie kamery. Nasleduje popis formátu M-JPEG a štandardu H.264 pre prenos videa a popis základov pre tvorbu užívateľského rozhrania pre robotické aplikácie v podobe grafického užívateľského rozhrania a priehľadového displeja. Poslednou časťou rešeršnej sekcie práce je prehľad použiteľných ovládacích zariadení, popis použitého diaľkového ovládača, jeho hardvérového a softvérového vybavenia v podobe Raspberry Pi, dotykového displeja a operačného systému Raspbian.

Ďalej sa práca venuje návrhu samotnej robotickej aplikácie, čo zahŕňa popis fungovania celého systému a návrh rozhrania pre ovládanie mobilného robota Perseus pomocou diaľkového ovládača. V danom rozhraní ide najmä o priehľadový displej, grafické užívateľské rozhranie, stavové hlásenia a popis funkcií tlačidiel ovládača.

Na záver je v práci popísaná samotná realizácia navrhnutej robotickej aplikácie. Popisuje použité rozhrania pre programovanie aplikácií, použitú sadu programových nástrojov Qt, prípravu platformy Raspberry Pi pre krížovú kompiláciu a samotnú implementáciu výslednej aplikácie do tejto platformy. Ďalej je popísaný samotný princíp fungovania tejto aplikácie a nakoniec je porovnaná výsledná realizácia aplikácie s vzhľadom k vytvorenému návrhu.

2 PLATFORMA GEARS-SMP

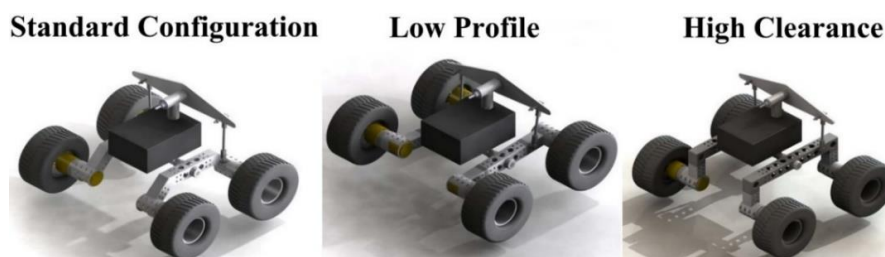
Robot Perseus je založený na mobilnej platforme GEARS-SMP (Obr. 1), ktorá bola navrhnutá pre univerzity zapojené do robotického výskumu v reálnom svete, inštitúciou Gears Educational Systems. Ide o platformu obsahujúcu motory osadené na kolesách a samotnú konštrukciu, ktorá je určená pre implementáciu ďalších technológií ako sú senzory, vzdialená vizualizácia, GPS, autonómne riadenie a iných. [1]



Obr. 1: Platforma GEARS-SMP [1]

Konštrukčné prvky podvozku sú vyrobené z čiernych eloxovaných štvorcových túb zo zliatiny hliníka 6061-T6, ktoré sú pripevnené na hlavnú nápravu z titánovej zliatiny o priemere 1,27 cm. Uvedené prvky sú jednoducho upravovateľné a poskytujú možnosť škálovateľnosti na rôzne veľkosti a rôzne požiadavky nosnosti zostrojeného robota. [1]

Telo podvozku je extrémne stabilné aj pri prechádzaní ťažkým terénom, čo robí z tejto platformy, ktorá je ponúkaná v troch verziách (Obr. 2), vhodnú voľbu pre prieskumné roboty. Jednou z nich je „High Clearance“, ktorá ponúka vysoko postavenú platformu, napríklad pre lepšiu viditeľnosť kamery. Ďalšou je verzia „Low Profile“ pre lepšiu stabilitu vo vyšších rýchlostiach do ľahkých terénov. Posledný variant platformy, zároveň aj použitý v tejto práci, je „Standard Configuration“, ktorý ponúka vyššie posadenie vozidla, dobrú stabilitu a vhodnosť použitia aj do ťažkého terénu. [1] [2]



Obr. 2: Typy platforiem GEARS-SMP [2]

Okrem výberu typu platformy je v ponuke aj voľba použitého pohonu. K dispozícii sú tri hlavné typy motorov, ktoré sa líšia najmä prevodovým pomerom. Dostupné pomery sú 218,4:1, 65,5:1 a 19,7:1. Robot Perseus je vybavený motorom s prevodovým pomerom 65,5:1, ktorý dosahuje ideálny kompromis medzi rýchlosťou a krútiacim momentom. Ostatné parametre použitej platformy sú zobrazené v tabuľke nižšie. [1]

Tab. 1: Technické parametre použitej platformy GEARS-SMP [1] [2]

Váha	8,16 kg	Výrobca motora	Ametek Pittman
Nosnosť	8,16 kg	Typ motora	Jednosmerný kefový
Dĺžka	60,96 cm	Priemer	3,48 cm
Šírka	60,96 cm	Prevodový pomer	65,5:1
Výška	33,02 cm	Maximálny trvalý krútiaci moment	3,39 Nm
Rázvor	41,91 cm	Špičkový krútiaci moment (dočas.)	18,25 Nm
Priemer kolies	17,78 cm	Otáčky (bez záťaže)	71 ot./min
Svetlá výška	11,43 cm	Menovité napätie	12 V
		Enkodér	500 pulzov/ot.

2.1 Mobilný robot Perseus

Robot Perseus (Obr. 3) je postavený na platforme GEARS-SMP ako už bolo zmienené vyššie. Táto platforma je doplnená o ďalšie technológie za účelom použitia daného robota ako prieskumného. Ide najmä o sieťovú kameru, smerovač, servomotory pre polohovanie kamery a dosku plošných spojov (DPS) pre riadenie pohonu robota.



Obr. 3: Mobilný robot Perseus

2.1.1 Sieťová kamera

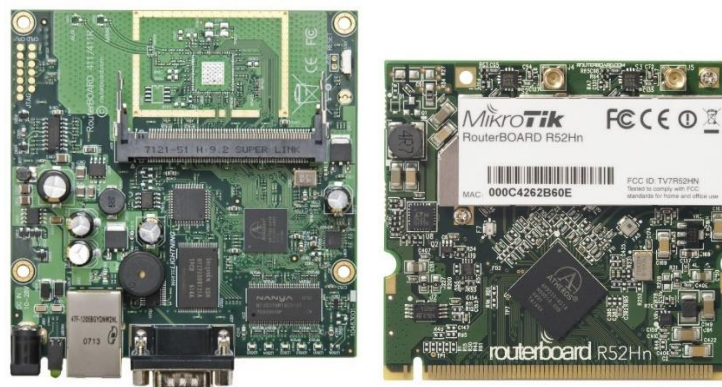
Ide o miniatúrnu verziu HDTV kamery pre exteriérové použitie od firmy AXIS. Skladá sa z kamery so senzorom RGB CMOS, ktorá dosahuje krytie IP66 a jednotky pre šírenie videosignálu z kamery (Obr. 4). Rozlíšenie videa dosahuje od 320x180 pixelov až po 1280x720 pixelov. Podporuje video kompresiu Motion JPEG a štandard H.264, ktoré môžu byť použité v kombinácii s rôznymi sieťovými protokolmi ako napríklad HTTP, RTP, RTSP, RTCP a ďalšími. Ponúka mnoho nastavení obrazu a to napríklad: ostrosť, kontrast, teplotu farieb a mnohé iné. [3]



Obr. 4: Sieťová kamera AXIS 1214-E [3]

2.1.2 Smerovač

Pripojenie k sieti Wi-Fi je zabezpečené pomocou systémového riešenia firmy MikroTik. Ide o kombináciu smerovača a sieťového adaptéru (Obr. 5), ktoré ponúkajú využitie štandardu 802.11a/b/g/n v pásmach 2 a 5 GHz s rýchlosťou 200 Mbps v oboch smeroch komunikácie. Smerovač s procesorom o frekvencii 680 MHz a 64 MB RAM pamäťou pracuje pod operačným systémom RouterOS a ponúka možnosť využitia sériovej linky RS-232 a ethernetového portu. [4] [5]



Obr. 5: Smerovač Mikrotik RB411AH (vľavo) a sieťový adaptér Mikrotik RouterBOARD R52Hn miniPCI (vpravo) [4] [5]

2.1.3 Servomotory pre polohovanie kamery

Tieto inteligentné aktuátory (Obr. 6) od firmy Robotis sú použité pre polohovanie kamery nachádzajúcej sa na mobilnom robote. V oboch prípadoch ide o jednosmerné motory so vstavaným prevodom. Ich ovládanie a komunikácia s nimi je prevádzaná prostredníctvom implementovaného protokolu. Ponúkajú množstvo nastavení a spätnú väzbu, napríklad o svojej aktuálnej polohe, rýchlosti, aktuálneho odoberaného prúdu a iných. Servomotory sú vyrobené z kvalitných materiálov a priemyslového plastu. [6] [7]



Obr. 6: Servomotor Robotis Dynamixel EX-106 (vľavo) a RX-64 (vpravo) [6] [7]

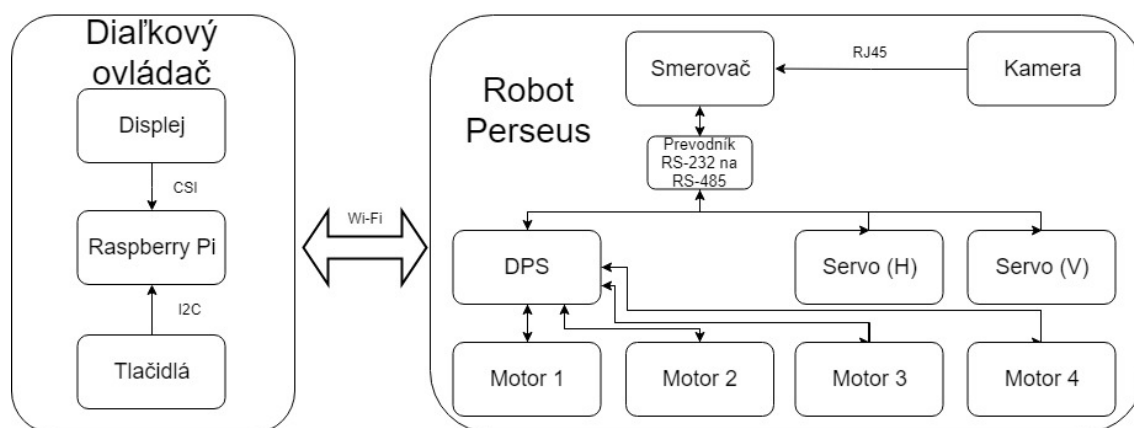
2.1.4 Doska plošných spojov pre riadenie pohonu

V prípade použitej dosky plošných spojov nie je dostupná žiadna schéma ani iná dokumentácia. DPS je prepojená pomocou sériovej linky RS-485 so smerovačom cez prevodník sériových liniek RS-232 a RS-485, od ktorého prijíma dáta k riadeniu pohonu robota. Zdrojový kód procesora DPS je písaný v jazyku C a obsahuje riadenie a komunikáciu s motormi. Pristupuje na registre 0x50, 0x52, 0x54 a 0x56 v pamäti, pomocou ktorých sa riadia motory pohonu robota. Taktiež obsahuje jednoduchý PID regulátor pre riadenie, monitorovanie odberného prúdu a spätnú väzbu od enkodérov motorov.

2.1.5 Princíp fungovania mobilného robota

Princíp fungovania mobilného robota je založený na komunikácii medzi diaľkovým ovládačom a mobilným robotom Perseus. Ovládač je vybavený jednodoskovým počítačom Raspberry Pi, v ktorom je naprogramované užívateľské rozhranie zobrazované na displeji ovládača. Toto rozhranie poskytuje videesignál z kamery a zobrazenie dát spätnej väzby od robota. Pomocou tlačidiel prítomných na ovládači je ovládaný pohyb robota a kamery. Celá potrebná komunikácia je sprostredkovaná pomocou siete Wi-Fi.

Na strane robota prebieha príjem a odosielanie dát pomocou smerovača, na ktorý je pripojená kamera a DPS pre ovládanie pohonu robota. Kamera vysielá videosignál pomocou štandardu H.264 cez protokol RTSP (Real Time Streaming Protocol). Smerovač prijíma dáta z ovládača, najmä pokyny k pohybu servomotorov alebo pohonu. Následne ich predáva cez sériovú linku RS-232 na prevodník sériových liniek RS-232 a RS-485, z ktorého sú priamo ovládané servomotory pre polohovanie kamery (vertikálne a horizontálne) a na DPS pre riadenie pohonu. Procesor DPS tieto pokyny spracováva prostredníctvom prítomného riadiaceho programu a sprostredkúva prenos dát na motory pohonu robota. Zjednodušený princíp je zobrazený na obrázku nižšie.



Obr. 7: Blokový diagram princípu fungovania robota Perseus

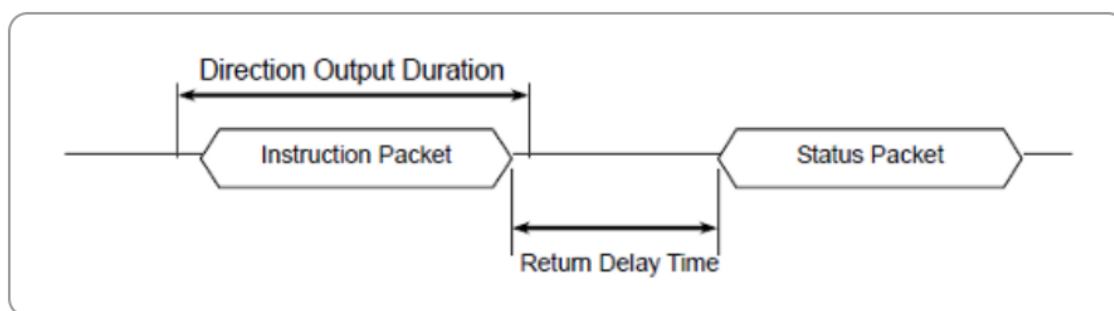
2.2 Protokol pre riadenie servomotorov Robotis Dynamixel RX-64 a EX-106

Pre riadenie servomotorov Robotis Dynamixel je potrebné, aby komunikácia bola nadviazaná podľa protokolu definovaného pre zariadenia Dynamixel. Z hľadiska fyzického pripojenia ide o zbernicu RS-485 s konektorom Molex pre sériové reťazenie pripojených zariadení. Protokol využíva polo duplexnú asynchrónnu sériovú komunikáciu pomocou 8 bitov, jedným stop bitom a bez parity. Šírenie signálu je riešené prostredníctvom paketov obsahujúcich binárne dáta. Rýchlosť komunikácie sa môže pohybovať v rozmedzí 7363 bps až 1 Mbps. [6] [7]

Riadiaca jednotka a zariadenia Dynamixel komunikujú medzi sebou zasielaním a prijímaním paketov. V komunikácii sa využívajú dva druhy. Prvým je inštrukčný paket, ktorý odosiela riadiaca jednotka a používa ho pre riadenie zariadení Dynamixel. Druhým je stavový paket, ktorým servomotor odpovedá riadiacej jednotke. Dôležitým prvkom v tejto komunikácii je ID (identifikačné číslo), ktoré musí byť unikátne pre každé zariadenie, v prípade zapojenia viacerých servomotorov na jednu zbernicu. Prítomnosť ID v oboch paketoch dáva možnosť

riadiacej jednotke komunikovať a riadiť práve jedno zariadenie, aj v prítomnosti viacerých zariadení pripojených na zbernicu. [8]

Ide o polo duplexný UART (Universal asynchronous receiver-transmitter), kde zápis a čítanie nie je možné používať v ten istý čas. Táto metóda je všeobecne používaná práve vtedy, keď veľa zariadení potrebuje byť pripojených na jednej linke. V podstate sa jedná o predchádzanie kolízií na zbernici prevádzané tým, že keď je jedno zo zariadení nastavené ako vysielacie a odosiela pakety, všetky ostatné zariadenia pripojené na zbernicu musia byť v stave prijímania. V tomto prípade nastavuje smer komunikácie riadiaca jednotka. Komunikácia má vždy nastavený vstupný smer a len v prípade odosielania inštrukčného paketu sa nastavuje výstupný smer (Obr. 8). [8]



Obr. 8: Riadenie komunikácie [8]

2.2.1 Riadiaca tabuľka

Ide o štruktúru dát, ktorá slúži riadiacej jednotke a je implementovaná v každom servomotore Dynamixel. V prípade požiadavky čítacím inštrukčným paketom poskytuje informácie o stave daného zariadenia a v prípade zapisovacieho inštrukčného paketu slúži pre modifikáciu práve týchto dát. Každá hodnota reprezentujúca špecifické dáta má v tejto tabuľke unikátnu adresu, ktorá musí byť pre čítanie alebo zápis uvedená v inštrukčnom pakete. [6] [7]

Tabuľka je rozdelená do dvoch pamäťových priestorov. Prvým je pamäť RAM (Tab. 2), kde pri každom zapnutí servomotoru sú dáta obsiahnuté v tejto časti resetované do inicializačných hodnôt. Naopak v pamäťovom priestore EEPROM (Tab. 3) si dáta uchovávajú svoje hodnoty aj po vypnutí. Veľkosť každej hodnoty v oboch tabuľkách sa pohybuje v rozmedzí jedného až dvoch bajtov a v prípade dvoj bajtovej veľkosti je využívaná endianita „little-endian“. Tabuľka má dva druhy prístupu k dátam. Prvým je „RW“ (read and write) prístupové povolenie, ktoré povoľuje zápis aj čítanie dát a používa sa hlavne na riadenie. Druhým typom je „R“ (read only), ktoré naopak povoľuje dáta len čítať a všeobecne sa používa len pre monitorovanie stavu zariadenia. [6] [7]

Tab. 2: RAM riadiacej tabuľky RX-64/EX-106(+modré) [6] [7]

Adresa	Veľkosť (Bajt)	Názov	Popis	Prístup	Inicial. hodnota
24	1	Aktivácia krútiaceho momentu	Zapnutie/vypnutie krútiaceho momentu	RW	0
25	1	LED	Stav LED	RW	0
26	1	CW zhoda pozície	Zhoda pozície (v smere hod. ručičiek)	RW	1
27	1	CCW zhoda pozície	Zhoda pozície (v protismere hod. ručičiek)	RW	1
28	1	CW dodržanie sklonu	Dodržanie sklonu (v smere hod. ručičiek)	RW	32
29	1	CCW dodržanie sklonu	Dodržanie sklonu (v protismere hod. ručičiek)	RW	32
30	2	Cieľová pozícia	Cieľová pozícia	RW	-
32	2	Rýchlosť pohybu	Rýchlosť Pohybu	RW	-
34	2	Limit krútiaceho momentu	Limit krútiaceho momentu	RW	ADD 14&15
36	2	Aktuálna pozícia	Aktuálna pozícia	R	-
38	2	Aktuálna rýchlosť	Aktuálna rýchlosť	R	-
40	2	Aktuálna záťaž	Aktuálna záťaž	R	-
42	1	Aktuálne napätie	Aktuálne napätie	R	-
43	1	Aktuálna teplota	Aktuálna teplota	R	-
44	1	Registrovaný	Ak je inštrukcia evidovaná	R	0
46	1	Pohybujúci sa	Stav pohybu	R	0
47	1	Zámok	Zamknutie EEPROM	RW	0
48	2	Punch	Min. prúdová hranica	RW	32
56	2	Snímaný prúd	Odoberaný prúd	R	-

Tab. 3: EEPROM riadiacej tabuľky RX-64/EX-106(+modré) [6] [7]

Adresa	Veľkosť (Bajt)	Názov	Popis	Prístup	Inicializačná hodnota
0	2	Modelové číslo	Modelové číslo	R	64/107
2	1	Verzia Firmvéru	Verzia Firmvéru	R	-
3	1	ID	Dynamixel ID	RW	1
4	1	Baudová rýchlosť	Komunikačná rýchlosť	RW	34
5	1	Oneskorenie návratu	Oneskorenie odozvy	RW	250
6	2	CW Uhlový limit	Uhlový limit (v smere hod. ručičiek)	RW	0
8	2	CCW Uhlový limit	Uhlový limit (v protismere hod. ručičiek)	RW	1023/4095
10	1	Mód prevádzky	Nastavenie duálneho módu	RW	0
11	1	Teplotný limit	Max. vnútorný teplotný limit	RW	80
12	1	Min. napät'ový limit	Min. limit pre vstupné napätie	RW	60
13	1	Max. napät'ový limit	Max. limit pre vstupné napätie	RW	190/240
14	2	Max. krútiaci moment	Max. krútiaci moment	RW	1023
16	1	Úroveň vráteného stavu	Voľba typov vráteného stavu	RW	2
17	1	Alarmová LED	LED pre alarm	RW	36
18	1	Vypnutie	Chybová informácia o vypnutí	RW	36

2.2.2 Inštrukčný paket

Ako už bolo spomenuté vyššie jedná sa o paket, ktorý je vysielaný riadiacou jednotkou riadenému zariadeniu, v tomto prípade adresovanému servomotoru. Tento paket sa skladá z viacerých častí a má vopred stanovenú štruktúru, ktorá musí byť striktne dodržaná, aby ju adresát pochopil a spracoval. Daná štruktúra je zobrazená v tabuľke nižšie.

Tab. 4: Štruktúra inštrukčného paketu [8]

Header1	Header2	ID	Length	Instruction	Param1	...	ParamN	Checksum
0xFF	0xFF	ID	Length	Instruction	Param1	...	Param N	CHKSUM

Prvé dva bloky sú hlavičky paketu slúžiace pre indikáciu začiatku nového paketu. Nasledujúci blok „ID“, slúži pre identifikáciu zariadenia, ktorému daný paket bude patriť. ID sa používa v rozmedzí hodnôt 0 až 254, pričom hodnota 254 slúži ako „broadcast“, čo znamená, že všetky zariadenia spracujú daný paket akoby bol adresovaný práve im. Blok „Length“ určuje dĺžku paketu a je vyjadrený nasledovne:

$$Length = počet\ parametrov + 2,$$

kde potom nasleduje blok „Instruction“ definujúci inštrukciu, ktorá má byť vykonaná zariadením. Na výber je 9 druhov inštrukcií (Tab. 5), ktoré môžu byť použité. Nasledujú bloky „Param1“ až „ParamN“, slúžiace ako parametre inštrukcie, ak je to potrebné. Posledným blokom je inštrukčný „Checksum“ (kontrolný súčet), ktorý slúži pre kontrolu počas komunikácie, či paket nie je poškodený. Kontrolný súčet je počítaný ako:

$$Checksum = \sim(ID + Length + Instruction + Param\ 1 + \dots + Param\ N),$$

kde „~“ je znamienko vyjadrujúce operáciu binárneho doplnku. V prípade, že výsledok súčtu presiahne hodnotu 255 (0xFF), použije sa táto operácia len na 2 spodné bajty, čo bude odpovedať výslednej hodnote kontrolného súčtu. [8]

Tab. 5: Definície inštrukcií [8]

Hodnota	Inštrukcia	Popis
0x01	Ping	Kontrola či paket dorazil do zariad. so správnym ID
0x02	Read	Čítanie dát zo zariadenia
0x03	Write	Zápis dát do zariadenia
0x04	Reg Write	Zaregistruje inštrukčný paket do záložného režimu, kde paket je neskôr odoslaný pomocou inštrukcie Action
0x05	Action	Spustenie inštrukčného paketu, ktorý bol predtým zaregistrovaný inštrukciou Reg Write
0x06	Factory Reset	Resetuje Riadiacu tabuľku do továrensky predvolených nastavení
0x08	Reboot	Reštartuje zariadenie Dynamixel
0x83	Sync Write	Zápis dát na rovnakú adresu s rovnakou dĺžkou súčasne do viacerých zariadení
0x92	Bulk Read	Zápis dát na rozdielne adresy s rozdielnou dĺžkou súčasne do viacerých zariadení (len MX séria)

2.2.3 Stavový paket

Ide o paket, ktorý je odozvou Dynamixel zariadenia na inštrukčný paket, ako to už bolo uvedené vyššie. Stavový paket, podobne ako inštrukčný, sa skladá z blokov a má svoju štruktúru, ktorá mu je podobná (Tab. 6).

Tab. 6: Štruktúra stavového paketu [8]

Header1	Header2	ID	Length	Error	Param 1	...	ParamN	Checksum
0xFF	0xFF	ID	Length	Error	Param 1	...	Param N	CHKSUM

Jediným rozdielovým blokom je „Instruction“ blok, ktorý je nahradený blokom „Error“. Tento blok slúži pre riadiacu jednotku ako identifikátor chyby. Jedná sa o jedno bajtovú informáciu, kde každý bit má špecifický význam a bližšie popisuje chybu (Tab. 7). V stavovom pakete slúži blok „Param“, ako odpoveď na dotaz riadiacej jednotky, napríklad na aktuálnu pozíciu. Posledným blokom v štruktúre tohto paketu je stavový „Checksum“, ktorý je vyjadrený nasledovne:

$$Checksum = \sim(ID + Length + Error + Param\ 1 + \dots + Param\ N),$$

kde „~“ je opäť znamienko vyjadrujúce operáciu binárneho doplnku. [8]

Tab. 7: Význam chybových bitov [8]

Bit	Chyba	Popis
7	0	-
6	Chyba inštrukcie	Prijatie nedefinovanej inštrukcie alebo prijatie inštrukcie Action bez Reg Write inštrukcie
5	Preťaženie	Aktuálna zaťaž sa nedá prekonať nastaveným krútiacim momentom
4	Chyba Checksumu	Checksum doručeného paketu je nesprávny
3	Chyba rozsahu	Inštrukcia je mimo použiteľného rozsahu
2	Prehriatie	Vnútorná teplota zariadenia Dynamixel je mimo rozsah prevádzkovej teploty nastavenej v riadiacej tabuľke
1	Chyba uhlového limitu	Cieľová pozícia je mimo rozsah limitov CW Angle a CCW Angle
0	Chyba vstupného napätia	Použitie napätie je mimo rozsah prevádzkového napätia nastaveného v riadiacej tabuľke

2.3 Protokoly pre prenos videa

Ide o špecifikáciu pre prenos videa, kde v tomto prípade môžeme skôr hovoriť o video kodeku. Názov video kodek v sebe zahŕňa, nie len softvér, ale aj hardvér,

ktorý slúži na kompresiu a dekompresiu digitálneho videa. Pri kompresii hovoríme o enkodéri a naopak pri dekompresii o dekodéri.

Vo väčšine prípadov ide o stratovú kompresiu, čo znamená, že komprimované video práve počas kompresii stratilo nejakú časť informácií oproti jeho originálu. Po dekompresii táto strata informácií má väčšinou za následok menšiu kvalitu videa, ale menší dátový tok. Dátový formát, ktorý vznikne po kompresii videa zväčša odpovedá nejakej špecifikácii (štandardu).

V dnešnej dobe trh ponúka mnoho štandardov používaných pre prenos videa. Na výber je rodina štandardov MPEG alebo formát M-JPEG od inštitúcie ISO/IEC, rodina štandardov H.2XX, ktoré ponúka organizácia ITU-T, rodina štandardov VC-X od SMPTE, štandardy VPX a mnoho ďalších. Ďalšia časť tejto podkapitoly sa bude venovať formátu MJPEG a štandardu H.264 a to z dôvodu ich podpory zariadením pre vysielanie videesignálu, ktoré bolo použité v mobilnom robote Perseus.

2.3.1 Formát Motion JPEG

Ide o variantu použitia formátu JPEG na kompresiu videa, ktorý je riadený štandardom ISO/IEC JPEG a používaný na kompresiu statických obrazov. Ide o stratovú kompresiu obrazu, založenú na diskretnej kosínusovej transformácii (DCT). JPEG sa vzácné používa priamo vo videu, ale tvorí základ kompresie formátu Motion JPEG (M-JPEG) a DV (Digital Video) a rovnako formuje základ kodeku MPEG. M-JPEG formát používa metódu JPEG, ale vzácné sa riadi štandardom ISO/IEC JPEG. Vlastný štandard pre M-JPEG nikdy nebol vytvorený. DV je špecifický typ M-JPEG formátu s vlastným štandard rovnako ako MPEG. [9] [10]

M-JPEG vo svojej podstate aplikuje formát JPEG na každý snímok videa individuálne. Namiesto kompresie celého videa na jeden bitový tok vracia odpovedajúce JPEG bitové toky za sebou v jednom rámci. Jedná sa o tzv. „intra-frame“ kódovanie, ktoré je nenáročné na výpočtový výkon, ale za cenu väčšieho dátového toku. Z daného faktu, že ide o aplikáciu formátu JPEG vyplýva, že je využívaný Y'CbCr farebný priestor a chroma podvzorkovanie pre väčšiu kompresiu. Kompresný pomer tohto štandardu sa pohybuje v rozmedzí pomerov 2:1 do 20:1. [9] [10]

Vysielanie videesignálu prebieha väčšinou pomocou HTTP (Hypertext Transfer Protocol) protokolu separáciou každého snímku do individuálnej HTTP odpovede so špecifickým popisom. Vytvárajú sa pakety, ktoré obsahujú sekvencie JPEG snímok, ktoré sú následne prijímané klientom, akým je napríklad VLC (VideoLAN client) mediálny prehrávač alebo nejaký webový prehliadač. [11]

Fráza používaná v URL (Uniform Resource Locator) adresách pre požiadavku na vysielanie videesignálu vo formáte M-JPEG so špecifickými argumentami vyzerá nasledovne:

```
mjpg/video.cgi
```

Argumenty môžu byť špecifikované explicitne, ale aj pomocou preddefinovaného profilu. V tomto prípade ide o požiadavku „GET“. Celá požiadavka s argumentom požadovaného rozlíšenia vyzerá nasledovne:

```
http://myserver/axis-  
cgi/mjpg/video.cgi?resolution=320x240
```

V prípade, že požiadavka bola úspešná, odozvou je súvislý tok JPEG snímok. Jedná sa o špeciálny obsahový typ „multipart/x-mixed-replace“, ktorý informuje práve o vyslaní toku snímok. Každý z nich je ukončený hraničnou frázou „<boundary>“. Spojenie je udržiavané dovtedy, kým klient žiada o nové rámce a server ich dokáže poskytnúť. Telo rámca vyzerá nasledovne:

```
--<boundary>  
<image>
```

```
--<boundary>  
<image>
```

Vrátené pole „<image>“ obsahuje nasledovné:

```
Content-Type: image/jpeg  
Content-Lenght: <image size in bytes>
```

```
<JPEG image data>
```

Uvedené časti kódu, sú zdrojovým HTML (Hypertext Markup Language) kódom, pri prijímaní a odosielaní dát. [11]

2.3.2 Štandard H.264/MPEG-4-AVC

H.264/MPEG-4-AVC (Advanced Video Coding), inak označovaný ako aj MPEG-4 Part 10 AVC alebo H.264/AVC je štandard používaný na kompresiu videa. Vznikol v spolupráci organizácií MPEG (Moving Picture Experts Group) a ITU-T VCEG (Video Coding Experts Group), ktoré sa z dôvodu zjednotenia štandardu spojili a vytvorili partnerskú skupinu JVT (Joint Video Team). V roku 2003 bola vydaná jeho prvá verzia pod označením ITU-T Rec. H.264|ISO/IEC 14496-10. Ide o nástupcu štandardov H.263 a MPEG-4 Visual. [12]

Cieľom bolo zníženie dátového toku pri vysokej kvalite videa. Oproti formátu M-JPEG ide o výpočtovo náročnejší štandard, no vzhľadom na výkonnosť dnešného hardvéru je to ľahko implementovateľný formát za cenu lepšej kvality prenášaného obrazu, čo je dôvodom jeho vzniku.

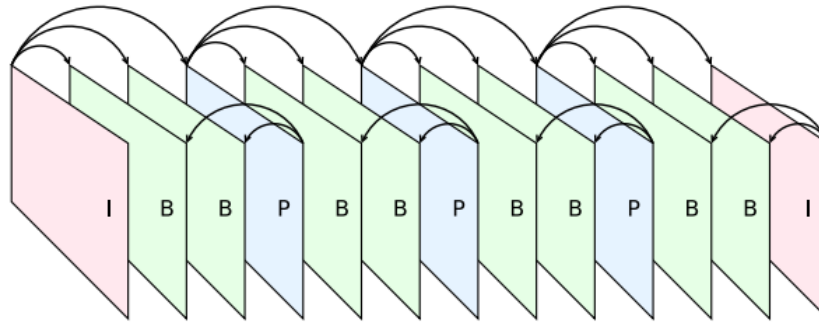
Veľkou výhodou je široký rozsah náročnosti na hardvér, ktorý bol dosiahnutý zavedením profilov (napr. Baseline, Main, High). Profily špecifikujú náročnosť a techniky, ktoré musí enkodér a dekodér zvládať. Zároveň určujú rôzne kategórie využitia (napr. videokonferencie, HDTV atď.) a zaisťujú kompatibilitu medzi zariadeniami. Dekodér alebo enkodér musí zvládať pracovať aspoň s jedným profilom a zároveň s profilmi, ktoré patria do nižšej úrovni. Profil definuje vo svojej podstate implementáciu určitých algoritmov (napr. CABAC entropické kódovanie, prediktívne bezstratové kódovanie, atď.) a nastavenie určitých parametrov (napr. bitová hĺbka, chroma formát, atď.). [13] [14] [15]

Druhým dôležitým pojmom definovaným v štandarde je úroveň. Jedná sa o sadu obmedzujúcich parametrov nepriamo definujúcich požadovaný výkon dekodéra, aby zvládal spĺňať požiadavky na daný profil. Ide napríklad o maximálne rozlíšenie, maximálny počet snímok za sekundu alebo bitovú rýchlosť. Dekodér, ktorý zvláda určitú úroveň podpory, musí zvládať splniť aj požiadavky na nižšie úrovne. [13] [14] [15]

Hlavným rozdielom oproti formátu M-JPEG je využitie „inter-frame“ kódovania. Je to kompresná metóda, ktorá využíva redundanciu v časovej oblasti medzi snímkami. „Inter“ kódovaný snímok je rozdelený do viacerých makro blokov, kde namiesto priameho zakódovania hodnôt pixelov v týchto blokoch, enkodér nájde najpodobnejší blok nachádzajúci sa v referenčnom snímku označovanom ako „I-frame“. Polohu tohto bloku v referenčnom snímku určuje tzv. pohybový vektor. Nie každý blok nájdený v referenčnom snímku je úplne podobný. Preto je vzniknutá diferencia medzi blokmi vyjadrená predikčnou chybou. V rámci tejto metódy kompresie vznikajú tri základné druhy snímok:

- **Snímok typu I** – referenčný snímok pre snímky B a P, ktorý je výhradne „intra“ kódovaný, teda nie sú potrebné žiadne iné snímky k jeho dekodovaniu.
- **Snímok typu P** – je predikovaný len pomocou predchádzajúcich snímok, kde sa väčšinou jedná o snímky I a výnimočne P. Oproti snímku I je potrebný asi polovičný objem dát.
- **Snímok typu B** – je obojsmerne predikovaný a obsahuje 2 referencie a to buď na P alebo I snímok. Oproti snímku I je potrebný asi štvrtinový objem dát k zakódovaniu. [16]

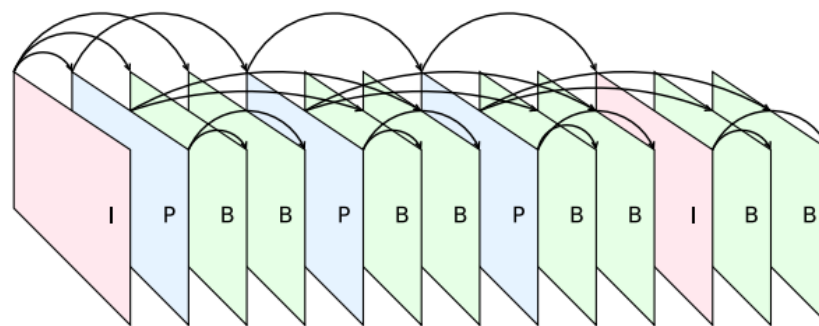
Skupina snímok, v ktorej sú všetky snímky referencované v rámci tejto skupiny sa nazýva GOP (Group Of Pictures). Táto štruktúra je zložená zo sledu snímok typu I, P a B, pričom tento sled začína a končí snímkom I. Štruktúra je definovaná dvoma parametrami. Prvým je N , ktorý vyjadruje vzdialenosť medzi snímkami I, čo vyjadruje celkovú veľkosť GOP. Druhým parametrom je M , vyjadrujúci vzdialenosť medzi snímkami P alebo I. [16]



Obr. 9: GOP [17]

Vzhľadom k tomu, že snímky typu B používajú obojsmernú predikciu, je nutné ich v GOP chronologicky usporiadať. Pri dekódovaní je obnovená pôvodná sekvencia snímok. Efektivita tohto kompresného algoritmu závisí práve na vyššie spomenutých parametroch. [16]

Veľká vzdialenosť N dosahuje veľkého kompresného pomeru, teda zníženie dátového toku, ale za cenu veľkého výpočtového a pamäťového nároku na dekodér a možnosť vzniku artefaktov. Tento stav nie je priaznivý pre vysielanie videosignálu, pretože sa nedosahuje stabilného prenosu. Je to najmä kvôli tomu, že čím väčšia je vzdialenosť N , tým je väčší aj časový úsek medzi I snímkami, čo určuje periodicitu odosielania dát. V praxi to znamená, že odozva obrazu na pohyb robota vykonaný ovládačom je príliš veľká, čo spôsobuje v podstate neovládateľnosť. Čím väčší je tento časový úsek, tým menší je počet dosiahnuteľných snímok za sekundu. Na druhej strane ani malá vzdialenosť N nie je dobrým riešením pre vysielanie videosignálu, pretože sa dátový tok môže razantne zvýšiť. Riešením je nájdenie ideálnej vzdialenosti N , pre stabilný dátový tok.



Obr. 10: Usporiadané GOP [17]

3 GRAFICKÉ UŽIVATEĽSKÉ ROZHRAINIE

Grafické užívateľské rozhranie (GUI - Graphical User Interface) je rozhranie medzi strojom a človekom. Umožňuje ovládať elektronické zariadenia pomocou vstupných zariadení (napr. myš, klávesnica a rôzne iné ovládacie prvky) a pomocou rôznych interaktívnych obrazových ovládacích prvkov. Tie spúšťajú funkcie systému a vykonávajú rôzne úkony. Klasickými predstaviteľmi týchto ovládacích prvkov sú rozličné druhy tlačidiel, ikon, posuvníkov, formulárov a iných prvkov. Príkladom iného užívateľského rozhrania je napríklad CLI (Command-line interface), teda príkazový riadok, ktorý využíva komunikáciu prostredníctvom textových príkazov. [18]

Hlavné využitie GUI v aplikáciách akými sú rozhrania pre interakciu s robotmi, je najmä v sekciiach pre nastavovanie parametrov. Ide o rôzne typy menu, ktoré ponúkajú nastavenie rôznych hodnôt alebo spustenie určitých funkcií. Primárnym dôvodom je, že hlavná obrazovka takýchto aplikácií tvorí najmä pohľad z kamery robota a zobrazenie len niektorých potrebných informácií, no tie nie sú interaktívne.

Existuje niekoľko základných princípov a pravidiel, ktoré je vhodné pri vytváraní GUI dodržiavať, aby výsledné rozhranie bolo jednoduché a intuitívne použiteľné.

3.1 Objekty

Ľudské videnie je optimalizované najmä pre vnímanie objektov. Pre človeka je rozpoznávanie textu zložitejšie a zaberá dlhší čas pre spracovanie mozgom. Vždy ak je to možné, je vhodné používať objekty ako predstaviteľ a danej entity, prípadne jeho kombináciu s textom. Existuje 7 základných princípov vizuálneho vnímania týkajúcich sa najmä tvarov a rozmiestnenia objektov, ktoré sú podrobnejšie popísané v [19]. Ide o princípy:

- 1) Blízkości
- 2) Podobnosti
- 3) Spojitosti
- 4) Uzavretia
- 5) Symetrie
- 6) Popredia/pozadia
- 7) Spoločného predurčenia [19]

3.2 Text

Ako už bolo spomenuté vyššie, text je pre vnímanie človeka zložitejší ako vnímanie objektov alebo reči. Aby to bolo pre neho jednoduchšie, je vhodné text štrukturalizovať do celkov sebe podobných. Príkladom môžu byť odseky, odstavce, medzery, odrážky alebo rôzne druhy tabuľkového usporiadania. Vizualna hierarchia núti ľudí sústrediť sa na relevantné informácie. [19]

Ďalším faktorom je aj použité písmo, kedy zložitý typ písma, ktorý pre človeka nie je prirodzeným, je zároveň zložitým pre čítanie. Veľkosť písma taktiež zohráva veľkú rolu, kedy príliš malé písmo môže spôsobovať zlievanie textu a problematické čítanie. Pozadie písma by malo byť jednoduché, ideálne jednofarebné bez akéhokoľvek „šumu“ na pozadí spôsobujúceho rozptýlenie. Vhodné je použitie zjednodušeného jazyka, teda krátkych a výstižných slov, čo zaručuje minimálne využitie textu. Dôležité je aj prípadné použitie známej a konzistentnej terminológie. [19]

3.3 Farby

Dôležitým faktorom pri vytváraní GUI je vhodné použitie farieb. Je potrebné si uvedomiť, že náš zrak je optimalizovaný pre detekciu kontrastu a nie jas. Rozpoznávanie farieb podľa jasnosti je človeku v niektorých prípadoch nemožné a veľmi skresľujúce, preto by malo byť zvažované použitie farieb týmto spôsobom. Schopnosť rozlišovať farby závisí hlavne na tom ako sú prezentované. [19]

Základ dobrého GUI je použitie zreteľných farieb, ktoré sú charakteristické danému objektu alebo textu, prípadne situácii (chyba – červená farba, atď.). Dôležité je rovnako separovať výrazne oponentné farby (červený text na modrom podklade). Pri používaní farieb je vhodné rozlišovať ich podľa viacerých faktorov ako napr. nasýtenie, jas a odtieň. [19]

3.4 Periférne videnie

Pri vytváraní GUI si treba uvedomiť, že nie všetko, čo sa nachádza v zornom poli človeka je vnímané rovnako intenzívne. Periférne videnie u ľudí nie je veľmi rozvinuté, a vnímaný je najmä pohyb. Stacionárne objekty v nevýrazných farbách, ktoré sa nachádzajú v periférii zorného poľa užívateľa nebudú zaznamenané, preto je potrebné nejakým spôsobom upriamiť jeho pozornosť. [19]

V prípade vyskakovacích okien, symbolov, textov, či už oznamovacích, varovných alebo iných je vhodné, aby sa objavovali práve tam, kde je predpoklad, kam sa užívateľ práve pozerá. V prípade, že to nie je úplne možné a informácie budú zobrazené v periférii, je odporúčané použitie výrazných farieb, pohybu,

zvukov a iných prostriedkov. Tie si vynútia pozornosť užívateľa a jeho reakciu na dané upozornenie alebo informáciu. [19]

3.5 Priehľadový displej

Priehľadový displej (HUD - Head-Up Display) bol vyvinutý pôvodne pre letecké kokpity bojových lietadiel. Zmyslom tohto „displeja“ je premietat' potrebné informácie do zorného poľa pilota, aby nemusel byť rozptyľovaný získavaním týchto informácií z budíkov alebo iných zariadení, ktoré sú umiestnené mimo výhľad na vzdušný priestor. Táto technológia sa dnes už využíva aj v motocykloch a automobiloch. [20]

HUD našiel využitie aj vo videohrách. V podstate sa nejedná fyzicky o premietač premietajúci informácie, ale len o využitie myšlienky. Ide teda o rozšírenie užívateľského rozhrania (UI) hry o potrebné informácie. Tie sú zobrazené rovnako ako hra, na obrazovku do zorného poľa hráča. Klasickým znakom takéhoto HUD je minimálne využívanie textových informácií, ale práve naopak ich grafické zobrazenie. Dôležité je použitie grafiky vysokej kvality a rýchle renderovanie v kombinácii so svetlými farbami a vhodnými prvkami užívateľského rozhrania. HUD sa nepovažuje za formu GUI, pretože nie je interaktívne. Jeho využitie sa prenieslo aj do robotických aplikácií, kde HUD ponúka vhodné riešenie pre zobrazovanie potrebných stavových informácií o robote. [20]

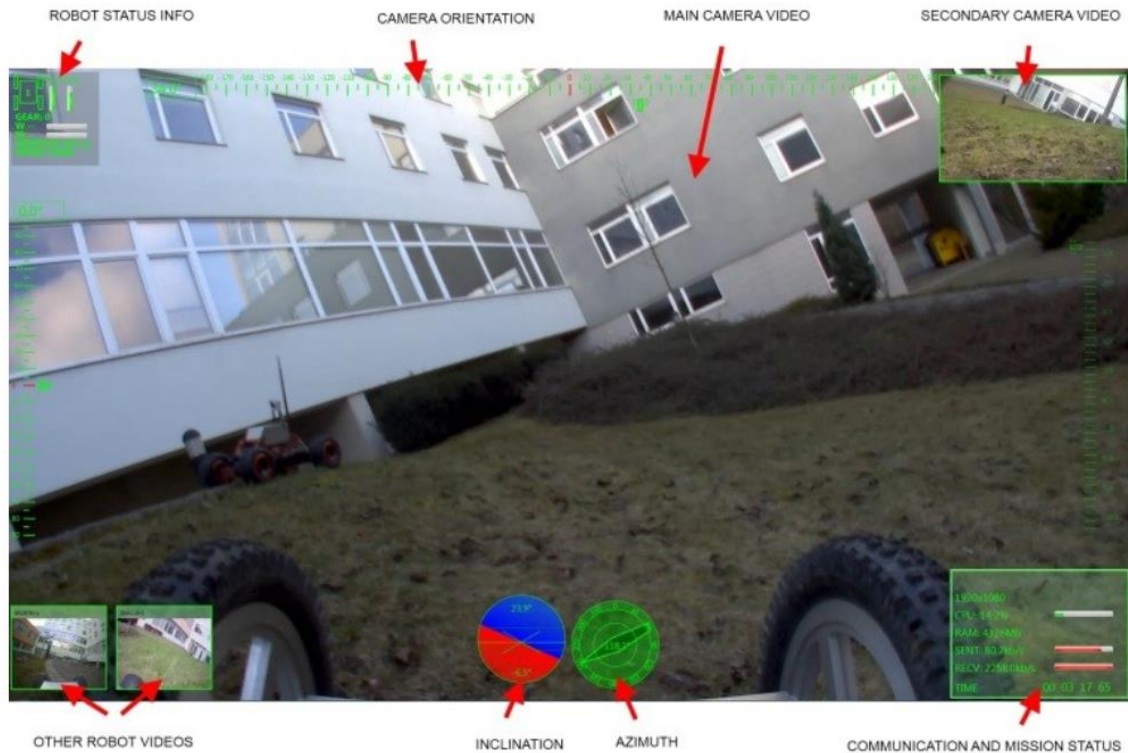
3.6 Užívateľské rozhranie robotických aplikácií

Rozhranie v takýchto typoch aplikácií musí byť maximálne prispôsobené svojmu účelu. Takýmto typom je napríklad robotický systém CASSANDRA vyvinutý firmou LTR s.r.o. a VUT v Brne. Ide o systém ponúkajúci možnosť ovládať skupinu robotov jednou ovládacou stanicou a operátorom. [21] [22]

Daný systém ponúka jednoduché a intuitívne užívateľské rozhranie medzi človekom a robotom naprogramované pomocou Microsoft .NET 4.5. Takéto rozhranie je potrebné najmä preto, aby operátor mohol byť plne sústredený na plnenie úlohy a boli mu poskytnuté všetky potrebné informácie. Typické je poskytovanie pohľadu z hlavnej kamery robota, ktoré pokrýva väčšinu obrazu a séria transparentných prekryvov s žiadanými dátami. [21] [22]

Príkladom je UI pre ovládanie robota Orpheus (Obr. 11), kde je možné pozorovať vyššie spomenuté prvky. Základ tvorí pohľad z hlavnej kamery robota v ľavom spodnom rohu, pohľad kamier iných robotov a v pravom hornom rohu pohľad zo sekundárnej kamery tohto robota. Ostatné časti obrazovky sú pokryté virtuálnym HUD. V strede hornej a ľavej časti obrazovky je zobrazená orientácia kamery, aby bolo možné zistiť kam je ešte možné kameru natočiť. V ľavom hornom

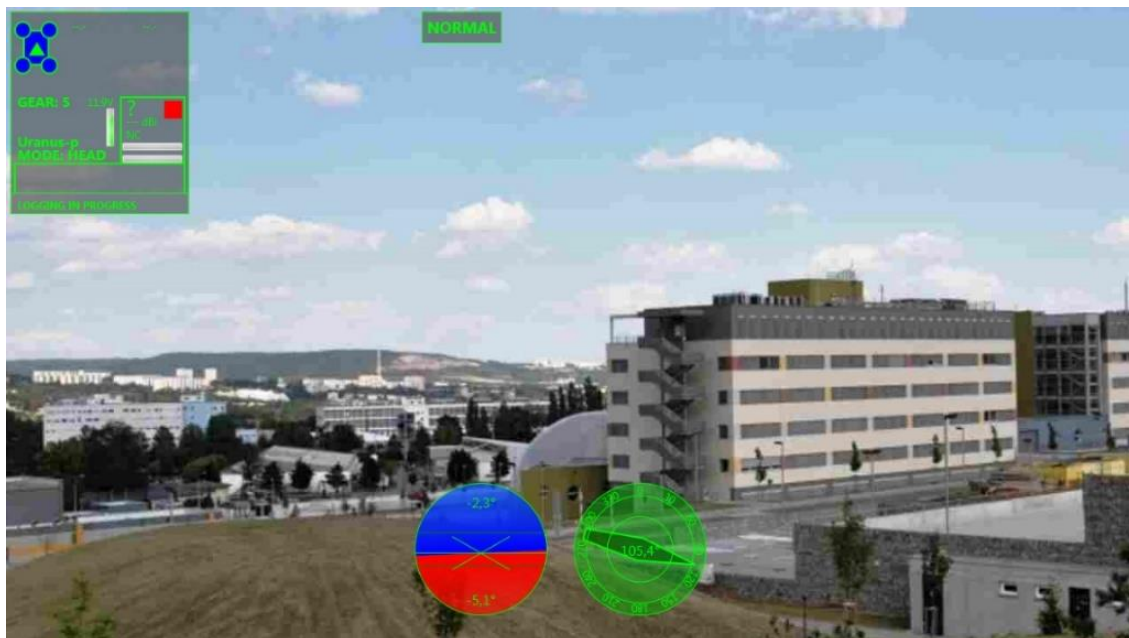
rohu obrazovky sú zobrazené stavové informácie o robote (napríklad stav batérie, rýchlosť chodu, atď.). V opačnom rohu sú informácie o komunikácii medzi robotom a ovládačom (množstvo odoslaných a prijatých dát), využitie procesoru a pamäte RAM. Poslednými poskytnutými informáciami v spodnej časti obrazovky je ukazovateľ azimutu a sklonu.



Obr. 11: HUD systému CASSANDRA pre mobilného robota Orpheus [21]

Podobným, jednoduchým a intuitívnym UI je vybavený aj vzdušný robot Uranus (Obr. 12), ktorého rozhranie je zobrazené na obrázku nižšie. Rovnako ako predchádzajúce UI využíva ako základ obrazu pohľad z kamery robota a virtuálny HUD s transparentnými prvkami. Ponúka ukazovatele sklonu a azimutu, ale aj napríklad ukazovatele stavu napätia batérie, prevodového stupňa a intenzity signálu. Ďalej použitý HUD informuje o stave prihlasovania do zariadenia alebo zobrazuje použitý mód.

Ako je vidieť v oboch prípadoch ide najmä o zobrazovanie relevantných informácií, ktoré sú cielene situované do určitých častí zorného poľa užívateľa. Toto je realizované spôsobom, aby dané informácie nezakrývali pohľad z kamery robota, ale aby najdôležitejšie informácie boli viac v zornom poli užívateľa a tie menej dôležité prevažne v jeho periférii.



Obr. 12: HUD systému CASSANDRA pre vzdušnú jednotku Uranus [23]

4 OVLÁDACIE ZARIADENIA

Dôležitou súčasťou mobilných robotov je ovládacie zariadenie. V dnešnej dobe je k dispozícii mnoho periférnych zariadení, ktoré sú použiteľné na tieto účely. Jedná sa o rôzne druhy pákových ovládačov, tzv. gamepadov, či iných zariadení tohto typu, no každý z nich má svoje špecifické vlastnosti a účel použitia.

V prípade vhodnosti ovládača pre robot Perseus rozhoduje aj priestorová možnosť dodatočného namontovania obrazovky, na ktorú bude prenášaný obraz z kamery. Ďalšou vlastnosťou by mala byť mobilita ovládača, jeho jednoduchá implementácia do platformy Raspberry Pi a možnosť ovládania v stoji a v pohybe.

4.1 Pákový ovládač

Pákový ovládač tzv. joystick (Obr. 13) je vstupné zariadenie v tvare páky pripevnenej k podstavcu. Ovládanie je prevádzané uchopením tejto páky rukou a vychyľovaním do strán, prípadne jej rotáciou. Tento typ ovládača vznikol na základe pákových ovládačov používaných v letectve, na použitie v leteckých simulátoroch (počítačových hier). V dnešnej dobe je jeho využitie všestrannejšie. Okrem hier je používaný aj na ovládanie priemyselných robotov, žeriavov atď..

Ovládač môže byť vybavený mnohými ďalšími funkciami, akými sú rôzne druhy tlačidiel, ktoré sú zväčša konfigurovateľné alebo páčkou, zvyčajne používanou pre ovládanie plynu. Dnešné pákové ovládače používajú ako fyzické rozhranie zbernicu USB a rozhranie pre spracovávanie vstupov DirectInput, ktorý je komponentom DirectX. [24]



Obr. 13: Joystick Saitek Cyborg X [24]

4.2 Gamepad

Gamepad (Obr. 14), nazývaný aj controlpad, prípadne joypad, je periférne vstupné zariadenie, ako vyššie spomínaný pákový ovládač, ktorý je spravidla držaný oboma rukami a ovládaný palcami.

Základnými prvkami sú analógové páčky, smerové tlačidlá (D-pad) a niekoľko funkčných tlačidiel. Analógové páčky majú väčšinou rovnakú funkciu ako smerové tlačidlá, no poskytujú možnosť určiť intenzitu vychýlenia páčky v danom smere. V dnešnej dobe sú pre tieto druhy ovládačov taktiež špecifické analógové tlačidlá na prednej strane (na ramene) ovládača z dôvodov častého použitia pre automobilové hry ako tlačidlá pre plyn a brzdu.



Obr. 14: Gamepad Razer Raiju Ultimate [25]

4.3 Iné druhy ovládacích zariadení

Trh s ovládacími zariadeniami je široký a ponúka mnoho druhov pre rôzne účely, od amatérskych prevedení až po profesionálne. V prípade ovládania mobilného robota je vhodné brať do úvahy zariadenia určené nie len na hranie videohier, ale aj na profesionálne ovládanie rádiom ovládaných (RC – Radio Controlled) modelov áut alebo vzdušných modelov, ako sú lietadlá, helikoptéry a iné.

V prípade ovládača pre RC modely áut (Obr. 15) je jeho uchopenie prevádzkané jednou rukou podobne ako to je u pákového ovládača. V tomto prípade uchopenie slúži len na držanie ovládača a nie na ovládanie, kde je následne používaný väčšinou len jeden prst tejto ruky pre ovládanie plynu. V podstate ide o princíp podobný pri používaní zbrane, kde spúšť slúži ako plyn. Druhou rukou je ovládané otočné koleso slúžiace pre zatáčanie s modelom a ďalšie funkčné tlačidlá, ktoré sú prítomné na ovládači. Na drahších typoch ovládačov sa nachádza displej zobrazujúci rôzne typy dát, ako napríklad časové údaje o zajazdení najrýchlejšieho kola na trati alebo medzičasy.



Obr. 15: Ovládač pre RC autá Futaba 4PV [26]

Iným typom je napríklad ovládač pre vzdušné modely (Obr. 16). Ide o zariadenie, ktoré je držané oboma rukami a prevažne ovládané palcami pomocou analógových páčok, rovnako ako to bolo u gamepadu. Oproti nemu sa, ale na tomto zariadení nenachádza D-pad, je mierne tvarovo odlišný a to hlavne z dôvodu prítomnosti displeja. Ten slúži podobne ako u vyššie zmieneného ovládača pre modely áut na zobrazovanie rôznych dát. Ide napríklad o nadmorskú výšku a iné druhy leteckých dát.



Obr. 16: Ovládač pre vzdušné modely Futaba 18MZ WC [27]

4.4 Ovládač pre mobilný robot Perseus

Na základe inšpirácie vyššie spomínaných druhov ovládacích zariadení, bol zostrojený ovládač pre danú aplikáciu. Ovládač robota Perseus (Obr. 17) bol inšpirovaný gamepadom z hľadiska rozmiestnenia tlačidiel a ovládacích prvkov

a tvarom práve ovládačom pre vzdušné RC modely. Najdôležitejším prvkom bola implementácia displeja, ktorá je hlavným dôvodom výsledného tvaru ovládača.

Ovládač je držaný oboma rukami a ovládaný palcami rovnako ako to je práve u vyššie spomínaného gamepadu a ovládača pre RC lietadlá. Obsahuje analógové páčky pre riadenie mobilného robota, smerové tlačidlá, ďalšie štyri funkčné tlačidlá a 2 signalizačné LED diódy. Jeho rozmery museli byť prispôsobené implementovanému 7" displeju a Raspberry Pi. Šasi ovládača bolo vytlačené na 3D tlačiarňi.

Ovládač ako taký je postačujúci, aby dokázal plniť svoj účel, no z hľadiska ergonómie má nedostatky. V prípade budúcich úprav a vylepšení ovládača je odporúčané odľahčiť toto zariadenie, pretože pri dlhšom používaní sa prejavuje jeho hmotnosť. Ostré hrany ovládača taktiež nie sú veľmi príjemné pri používaní rovnako ako materiál, ktorý je značne klzký a robí pomerne veľké ťažkosti pri držaní.

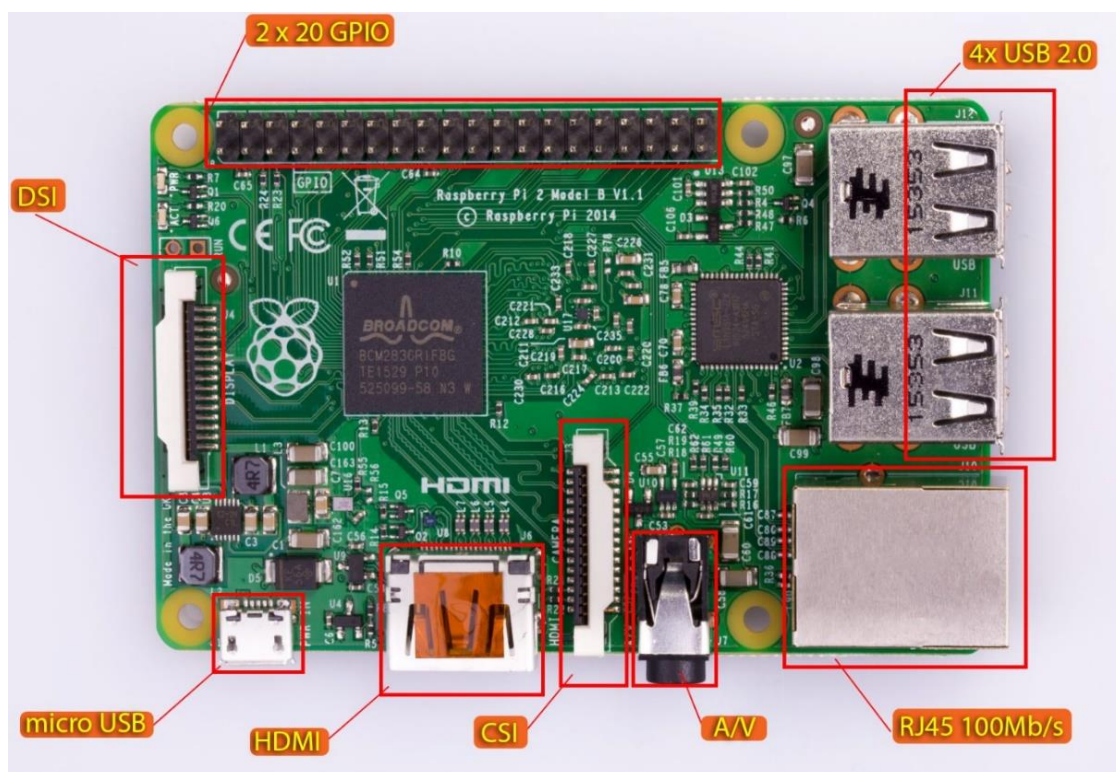


Obr. 17: Diaľkový ovládač pre robot Perseus

5 RASPBERRY PI

Raspberry Pi je nízkonákladový, jednodoskový počítač o veľkosti kreditnej karty. Pochádza od spoločnosti Raspberry Pi Foundation, sídliacej v Spojenom Kráľovstve. Bol vyvinutý pre učebné účely, ale v dnešnej dobe sa často využíva v malých projektoch ako napríklad multimediálne centrum, riadiaca jednotka a iné. Prvý krát prišiel na trh v roku 2011 a od tej doby je k dispozícii viacej variant, ktoré sa líšia najmä svojim hardvérovým vybavením. Ide napríklad o varianty Raspberry Pi 1, 2 a 3 vo verziách A, A+, B, B+ alebo najlacnejšie verzie typu Zero a Zero W. [28] [29]

Táto práca sa zaoberá verziou Raspberry Pi 2 model B V1.1 práve z dôvodu, že tento model je použitý v diaľkovom ovládači robota. Model je vybavený SoC (System on Chip) Broadcom BCM2836 a je založený na architektúre ARM s operačnou pamäťou o veľkosti 1 GB. Ako úložisko dát a miesto inštalácie operačného systému slúži micro SD karta. Toto zariadenie je vybavené štyrmi USB portami, konektorom RJ45 pre pripojenie do siete a portami HDMI, DSI (Display Serial Interface), CSI (Camera Serial Interface) a A/V (Audio/Video). V prípade A/V portu ide o 3.5 mm jack pre pripojenie audio/video periférií (Obr. 18). Pre napájanie slúži micro USB vstup, ktorý nemá vo funkcii dátový prenos. Podrobnejšiu technickú špecifikáciu je možno nájsť v tabuľke nižšie. [28] [29] [30]



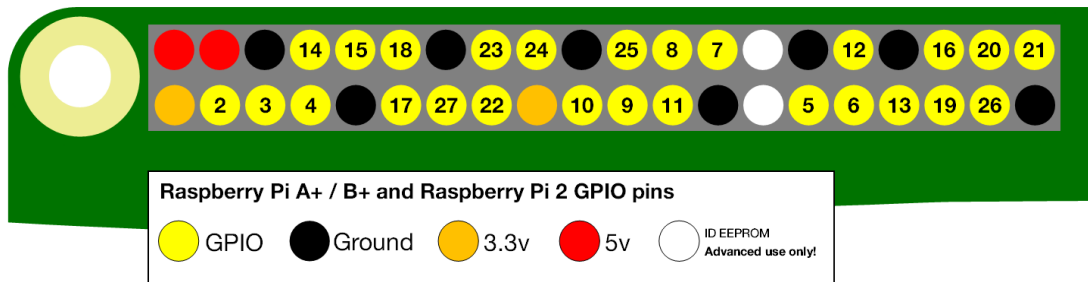
Obr. 18: Raspberry Pi 2 model B V1.1 [30]

Tab. 8: Technická špecifikácia Raspberry Pi 2 model B V1.1 [28] [30]

SoC	Broadcom BCM2836
Jadro	ARM Cortex-A7 MPCore 32-bit
Inštrukčný set	ARMv7
Počet jadier	4
Frekvencia	900 MHz
GPU	VideoCore IV 3D graphics core
Frekvencia	250 MHz
Podpora API	OpenGL ES 2.0, OpenVG 1.1
Hardvérová podpora	H.264/MPEG-4-AVC, H.265, VP8, VP9
Maximálne podporované rozlíšenie	1080p
Pamäť RAM	1 GB LPDDR2
Periférie	
Video	HDMI
	DSI
	CSI
Audio	A/V (3,5 mm jack)
	HDMI
Ostatné	4x USB 2.0
	UART
	I2C
	SPI
	LAN RJ45 10/100 Mb/s
	40 GPIO pinov
Úložisko	micro SD
Napájanie	5 V/1 A, micro USB
Rozmery	8,8 x 5,6 cm

Poslednou vlastnosťou Raspberry Pi, ktorou sa líši od štandardných osobných počítačov okrem svojej veľkosti je prítomnosť GPIO (General Purpose Input/Output) pinov (Obr. 19). Na doske sa nachádzajú dva 5 V piny, dva 3,3 V piny a osem pinov typu zem, ktoré sú nekonfigurovateľné. Všetky ostatné majú všeobecný účel a sú konfigurovateľné. GPIO pin nakonfigurovaný ako výstup dosahuje pri logickej úrovni 1 napätie 3,3 V a pri logickej úrovni 0 napätie 0 V. Vstupy majú rovnakú napäťovú toleranciu, tzn. 0 V a 3,3 V. Konfigurácia pinov prebieha softvérovo a následný prístup k daným pinom závisí od použitého programovacieho jazyka. Napríklad pri použití jazyka C je dostupná knižnica pomenovaná WiringPi, ktorá ponúka jednoduché funkcie na prístup k pinom alebo ich konfigurácii. Dôležitým faktom je, že prúdová záťaž výstupov sa pohybuje

v jednotkách mA, takže napríklad pri riadení motoru je nutné použiť H mostík alebo riadiacu dosku pre motory. [31]



Obr. 19: Rozloženie GPIO pinov [31]

Číslovanie pinov nie je v numerickom poradí a piny 0 a 1 sa nachádzajú na doske (fyzicky sa jedná o piny 27 a 28), ale sú vyhradené pre konkrétne účely uvedené v tabuľke nižšie. Okrem konfigurácie pinov, ako vstupných a výstupných, môžu plniť aj rôzne alternatívne funkcie. Konkrétne ide o:

- PWM (Pulse-width modulation)
- SPI (Serial Peripheral Interface)
- I2C (Inter-Integrated Circuit)
- Sériová komunikácia

V prípade PWM je možné funkciu prevádzkovať na každom pine softvérovým prevedením. K dispozícii je tiež aj hardvérová verzia, ktorá má priradené konkrétne piny na lište GPIO. Štandardne je možné používať I2C a SPI zbernice na pripojenie rôznych periférií ako sú displeje, pamäte, prevodníky, ale aj rôzne snímače. Podrobné priradenie pinov v rámci všetkých funkcií je zobrazené v tabuľke nižšie. [31]

Tab. 9: Rozloženie funkcií na GPIO pinoch [31]

PWM	GPIO12	SPI0-MOSI	GPIO10	SPI1-MOSI	GPIO20
PWM	GPIO13	SPI0-MISO	GPIO09	SPI1-MISO	GPIO19
PWM	GPIO18	SPI0-SCLK	GPIO11	SPI1-SCLK	GPIO21
PWM	GPIO19	SPI0-CE0	GPIO08	SPI1-CE0	GPIO18
Serial-TX	GPIO14	SPI0-CE1	GPIO07	SPI1-CE1	GPIO17
Serial-RX	GPIO15			SPI1-CE2	GPIO16
I2C-Data	GPIO02	I2C-EEPROM Data		GPIO00	
I2C-Clock	GPIO03	I2C-EEPROM Clock		GPIO01	

5.1 Operačný systém

Pre Raspberry Pi existuje mnoho operačných systémov prevádzkovateľných na tomto hardvéri. Len oficiálna stránka výrobcu uvádza 10 operačných systémov, ktoré sú na stránke voľne stiahnuteľné. Jedným z nich je oficiálny operačný systém Raspbian, optimalizovaný pre použitie práve na zariadeniach Raspberry Pi. Ostatné ponúkané operačné systémy tretích strán sú nasledovné:

- Ubuntu Mate
- Snappy Ubuntu core
- Windows 10 IoT Core
- OSMC
- Librelec
- Pinet
- Risc OS
- Wheather Station
- Ichigojam Rpi [32]

V prípade Ubuntu Mate ide o desktopový operačný systém vychádzajúci práve z distribúcie Ubuntu, ktorý je stavaný pre zariadenia ARMhf. Ďalším je Snappy Ubuntu Core a tu hovoríme o jadre vyvinutom pre vývojárov. Operačný systém Librelec od OpenELEC (Open EMbedded Linux Entertainment Center) slúži pre vytvorenie XBMC mediálneho centra z Raspberry Pi. Risc OS je nelinuxový operačný systém spoločnosti Acorn Computers Limited Archimedes, pre ktorý bol vyrobený vlastný RISC procesor pomenovaný ARM (Acorn RISC Machine). Tento operačný systém sa nachádza priamo v ROM pamäti a tak nekladie vysoké hardvérové nároky na prevádzku a ponúka vysokú bezpečnosť. Microsoft Windows 10 IoT je voľne dostupný operačný systém, ktorý je stavaný pre aplikácie zamerané na priemysel 4.0. [32]

Existuje mnoho ďalších operačných systémov, ktoré sú bezproblémovo funkčné na zariadení Raspberry Pi ako napríklad Arch Linux ARM, Xubuntu, FreeBSD, OpenBSD. Niektoré z nich sú optimalizované priamo pre použitie na Raspberry Pi, napríklad operačný systém Pidora vychádzajúci z distribúcie Fedora Remix. Dokonca je už k dispozícii aj operačný systém založený na Androide od spoločnosti Google pod názvom Android Things.

V konečnom dôsledku sa aj tak stal najvhodnejším kandidátom oficiálny operačný systém Raspbian, pretože ide o plnohodnotný linuxový systém optimalizovaný pre zariadenia Raspberry Pi. Taktiež zároveň ponúka mnoho nástrojov a jazykov, v ktorých je možné písať kód, nehovoriac o pravidelných aktualizáciách systému a údržbách.

5.1.1 Raspbian

Raspbian je voľne dostupný operačný systém založený na Linuxovej distribúcii Debian patriacej do rodiny operačných systémov nazývaných „Unix-like“. Ide o oficiálne podporovanú, optimalizovanú a odporúčanú verziu operačného systému spoločnosťou Raspberry Pi Foundation. [32]

Tento operačný systém prichádza aj so sadou nástrojov, programov a utilít, ktoré sú predinštalované v systéme. V ponuke sú dve verzie. Prvou je desktopová verzia „Raspbian Stretch with Desktop“ s grafickým užívateľským rozhraním (GUI), ktorá nebude úplne vhodná pre účely tejto práce. Dôvodom je vyššia záťaž na systém a bezúčelnosť pre danú aplikáciu. Druhou verziou je „Raspbian Stretch Lite“, ktorá nedisponuje GUI, ale len príkazovým riadkom (CLI – Command Line Interface). [33] [34]

Daný operačný systém bol vytvorený priamo pre použitie na zariadeniach, ktoré disponujú procesormi s architektúrou ARMv6 a podporujú VFPv2 (vector floating point version 2), ale aj pre systémy založené na platforme i386. Je vybavený základnými balíčkami APT, dpkg, GNU, ktoré sú využívané v rodine operačných systémov Unix-like. Raspbian používa monolitické jadro poskytujúce väčšiu efektivitu oproti mikrokernelovým jadram, práve tým, že sprostredkuje všetky služby programom a priamo prideluje prístup k hardvéru. [33]

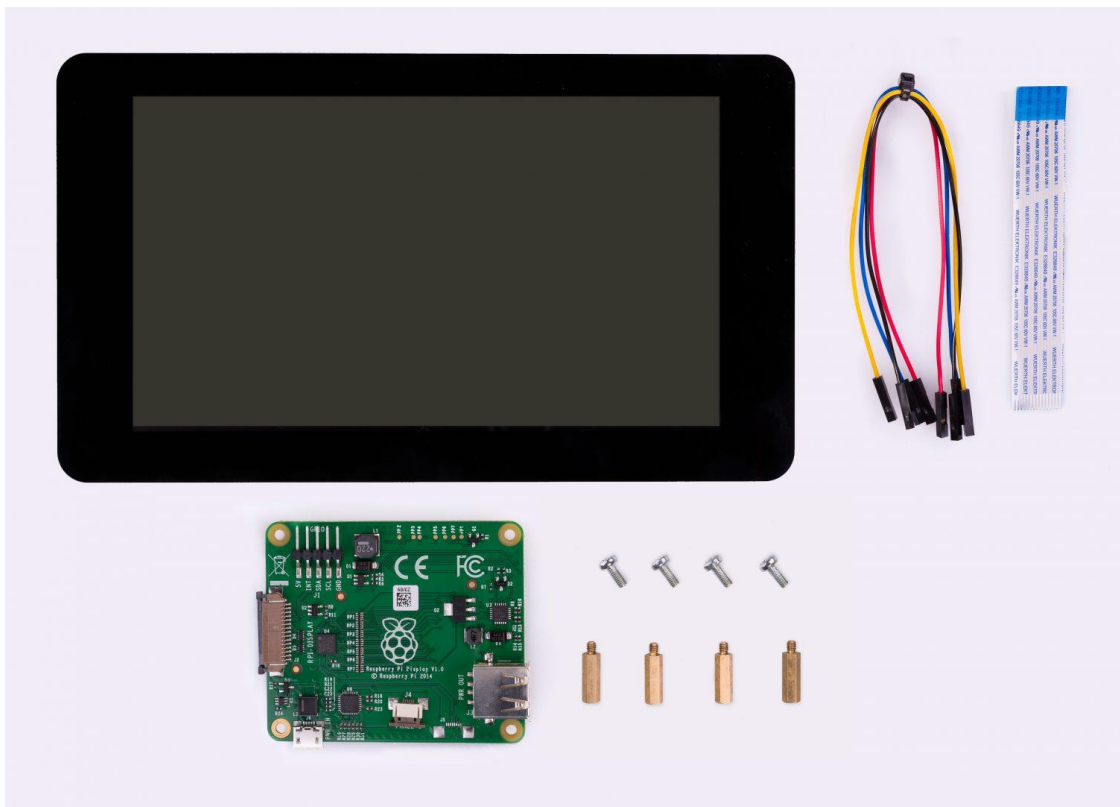
5.2 Rozšírenia Raspberry Pi

Raspberry Pi ponúka veľkú možnosť rozšírení prítomnosťou 4 USB portov, HDMI portu a hlavne GPIO. Je možné zapojiť mnoho snímačov, displej a iných zariadení.

5.2.1 Raspberry Pi displej

Dial'kový ovládač pre robot Perseus je vybavený dotykovým displejom Raspberry Pi Touch Display (Obr. 20) priemyselnej kvality, ktorý je oficiálne vyrábaný pre platformu Raspberry Pi spoločnosťou Inelco Hunter. [35] [36]

Ide o 7 palcový RGB dotykový (10 bodový kapacitný dotykový displej FT5406) displej s rozlíšením 800x480 pixelov s 24 bitovou hĺbkou farieb. Displej ďalej ponúka 140° horizontálny pozorovací uhol a 130° vertikálny pozorovací uhol. Prevádzková teplota je -20° C až 70° C, kontrastný pomer má hodnotu 500 a priemerná hodnota jasu je 250 cd/m². K displeju je dodávaná aj DPS spolu s potrebnou kabelážou na prepojenie s Raspberry Pi pomocou DSI portu pre prenos obrazu. Rovnako ako Raspberry, pracuje pod napätím 5 V a maximálnym odberným prúdom 500 mA. [35] [36]



Obr. 20: Raspberry Pi Touch Display [35]

5.2.2 Wi-Fi modul

Pre možnosť pripojenia Raspberry Pi do siete sa ponúkajú dve možnosti. Prvou je pripojenie prostredníctvom ethernetu a druhou bezdrôtové pripojenie pomocou Wi-Fi. Novšie modely, akým je napríklad Raspberry Pi 3 už majú zabudovaný vlastný Wi-Fi modul, takže nie je potrebné použitie takéhoto rozšírenia. Keďže Raspberry Pi je implementované do diaľkového ovládača a bude vyžadovaná určitá mobilita, no zároveň nie je vybavené Wi-Fi modulom, ponúka sa možnosť využitia bezdrôtového USB adaptéra.

V rámci diplomovej práce bol použitý adaptér od výrobcu Edimax s označením EW-7811Un (Obr. 21). V prípade použitia operačného systému Raspbian, nie je potrebné žiadne dodatočné inštalovanie ovládačov, len následné jednoduché pripojenie k dostupnej Wi-Fi sieti. [37]



Obr. 21: USB adaptér Edimax EW7811Un [37]

Tento bezdrôtový adaptér podporuje štandard IEEE802.11n s prenosovou rýchlosťou až 150 Mbps, je WPS (Wi-Fi Protected Setup) kompatibilný a podporuje šifrovanie WEP, WPA a WPA2. Ostatné parametre tohto adaptéra sú uvedené v tabuľke nižšie. [37]

Tab. 10: Technická špecifikácia Wi-Fi adaptéru EDIMAX [37]

HARDWARE INTERFACE	STANDARD	FREQUENCY BAND
<ul style="list-style-type: none"> 1 USB 1.0/2.0 Type A Internal Antenna 	<ul style="list-style-type: none"> IEEE802.11b, 802.11g, 802.11n 	<ul style="list-style-type: none"> 2.4000~2.4835GHz (Industrial Scientific Medical Band)
DATA RATE	INSTALLATION	LED & DIMENSIONS
<ul style="list-style-type: none"> 11b: 1/2/5.5/11Mbps 11g:6/9/12/24/36/48/54Mbps 11n (20MHz): MCS0-7 (up to 72Mbps) 11n (40MHz): MCS0-7 (up to 150Mbps) 	<ul style="list-style-type: none"> Multi-language EZmax Setup Wizard 	<ul style="list-style-type: none"> Link/Activity 7.1(H) x 14.9 (W) x 18.5 (D) mm
SECURITY	OUTPUT POWER	HUMIDITY & TEMPERATURE
<ul style="list-style-type: none"> WEP 64/128, WPA, WPA2, and IEEE802.1x Software WPS configuration 	<ul style="list-style-type: none"> 11b:17 ± 1.5dbm 11g:b:15 ± 1.5dbm 11n:14 ± 1.5dbm 	<ul style="list-style-type: none"> Operating : 10~90% (Non Condensing) Storage : Max. 95% (Non Condensing) Operating : 32~104°F (0~40°C) Storage : -4~140°F (-20~60°C)
SYSTEM REQUIREMENTS	RECEIVE SENSITIVITY	CERTIFICATIONS
<ul style="list-style-type: none"> Windows XP/Vista/7 / 8 Linux & Mac OS 	<ul style="list-style-type: none"> 11n(20MHz)@MCS7: -68dBm±2dBm 11n(40MHz)@MCS7: -64dBm±2dBm 11g@54Mbps: -71dBm±2dBm 11b@11Mbps: -81dBm±2dBm 	<ul style="list-style-type: none"> CE, FCC , WiFi

6 NÁVRH APLIKÁCIE

Táto kapitola sa venuje návrhu rozhrania pre ovládanie mobilného robota Perseus pomocou diaľkového ovládača, popisu fungovania a správania sa systému. Jednotlivo sú popísané základné úkony ako je zapnutie, či vypnutie robota a ovládača, návrh UI ovládača, popisu chybových stavov, jeho prípadov použitia tzv. „Use case“ a popisu funkcií tlačidiel ovládača.

6.1 Základné úkony

Základné úkony sa budú skladať zo zapnutia a vypnutia zariadení (robot a ovládač) a určitých úkonov potrebných vykonať pred zapnutím či vypnutím týchto zariadení, aby sa predchádzalo nežiadúcim situáciám.

6.1.1 Zapnutie zariadení

Pred zapnutím zariadení je nutné umiestniť robota na rovnú plochu, pretože pri jeho zapnutí je možné predpokladať mierny pohyb, z dôvodu nabudenia vinutí motorov pohonu. To by mohlo mať za následok pád robota v prípade, že by sa nachádzal v nebezpečnej pozícii. Pred zapnutím je rovnako potrebné sa uistiť, či sú antény smerovača natočené smerom nahor a to z dôvodu najlepšieho šírenia signálu (dosah a rozptyl).



Obr. 22: Umiestnenie vypínača robota

Zapnutie robota sa vykonáva preklopením pákového tlačidla umiestneného na prednej časti robota do polohy „ON“ (Obr. 22). Zapnutie má za následok okamžité zablokovanie kolies spôsobené nabudением vinutí motorov. Dosiachnutie stavu plnej funkčnosti robota trvá približne 1 minútu, z dôvodu inicializácie

smerovača. Tento stav je možné pozorovať práve automatickým pripojením ovládača k smerovaču robota.

Zapnutie ovládača sa vykonáva preklopením kolískového vypínača, ktorý sa nachádza na prednej vrchnej časti ovládača, do polohy „|“ (Obr. 23). Po zapnutí nasleduje sled inicializačných úkonov operačného systému sprevádzaných aj vizuálnymi zobrazeniami na displeji. Dosiahnutie stavu plnej funkčnosti trvá približne 1 minútu, kde tento stav je možné pozorovať zobrazením hlavnej obrazovky (viz.6.2.1 Priehľadový displej) alebo oznámením „Waiting for connection...“ na obrazovke ovládača (viz. 6.2.3 Stavové hlásenia).



Obr. 23: Umiestnenie vypínača ovládača

6.1.2 Vypnutie zariadení

Pred vypnutím robota je nutné sa uistiť, že sa robot nenachádza na nerovnom teréne, prípadne v nebezpečnej polohe. Tá by mohla spôsobiť jeho pád, z dôvodu odbudenia cievok motorov, čo má za následok uvoľnenie kolies.

Na poradí vypnutia zariadení nezáleží a vykonáva sa rovnakými tlačidlami, ktoré slúžia aj pre zapnutie zariadení. Vypnutie robota musí byť uskutočnené až po dokončení všetkých pohybov robota alebo polohovania kamery, z dôvodu predchádzania nežiadúceho poškodenia motorov a servomotorov robota.

6.2 Návrh užívateľského rozhrania

Užívateľské rozhranie sa bude skladať z dvoch základných častí. Prvou je hlavná obrazovka, prostredníctvom, ktorej bude robot ovládaný. Tá bude zložená z väčšej

časti z obrazu pohľadu kamery a HUD. Druhou časťou bude GUI v podobe menu, ktoré bude slúžiť pre parametrizáciu systému.

6.2.1 Priehľadový displej

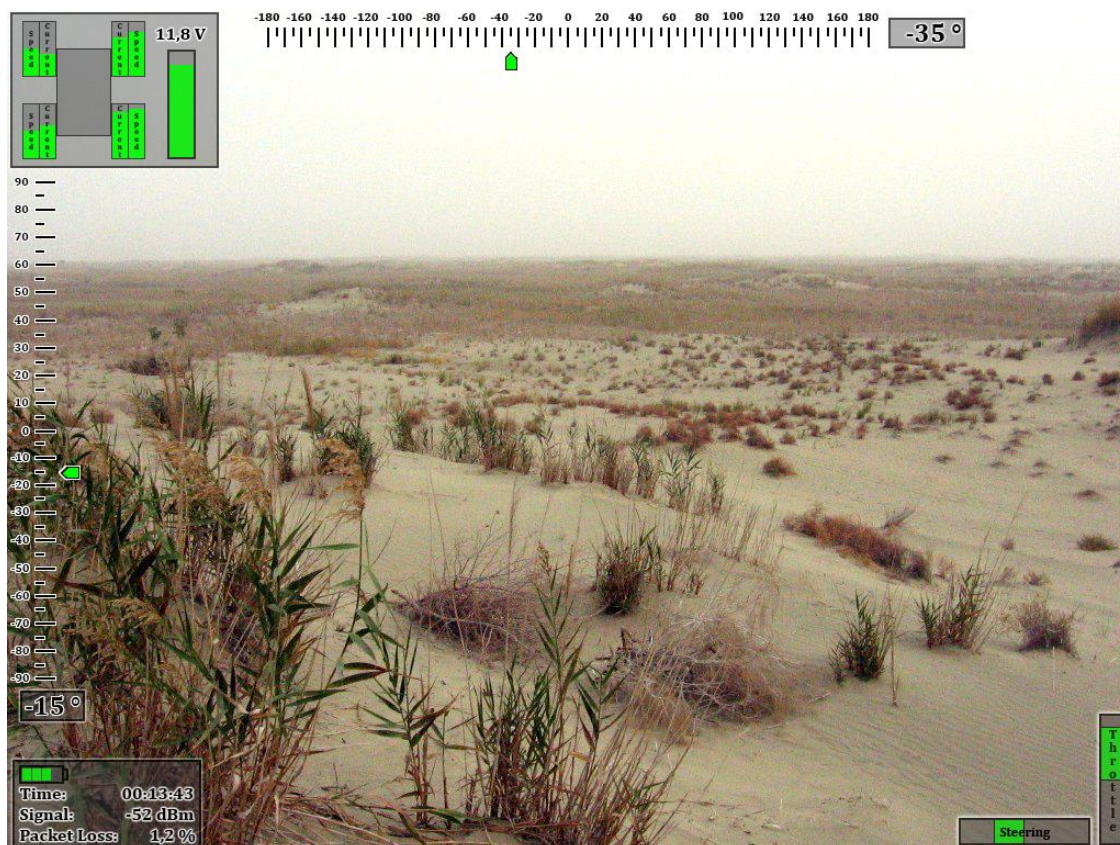
Vytvorený návrh HUD (Obr. 24) sa skladá zo štyroch častí. Prvou je spätná väzba od analógových páčok ovládača, ktorú možno vidieť v pravom dolnom rohu obrazovky. Ide o indikátory intenzity zatáčania a pohybu vpred a vzad. Zobrazované sú práve z dôvodu vytvorenia vizuálnej spätnej väzby od fyzického naklonenia páčky, čo ponúka možnosť vytvorenia si predstavy o koľko je ešte možné zvýšiť výkon v pohybe vpred a vzad a o koľko je ešte možné zatočiť do strany.

Druhou časťou je ukazovateľ natočenia kamery. Ten sa skladá z indikátora vertikálneho natočenia kamery v ľavej časti obrazovky a vo vrchnej časti obrazovky z indikátora horizontálneho natočenia kamery. Vertikálne natočenie je indikované v rozmedzí $\pm 90^\circ$ zelenou šípkou na osi a zároveň je jeho hodnota vyjadrená číselne pod osou v obdĺžnikovom orámovaní. Horizontálne natočenie kamery je rovnako indikované zelenou šípkou na osi v rozmedzí $\pm 180^\circ$ a jeho číselná hodnota je zobrazená napravo od osi v obdĺžnikovom orámovaní. Tento prvok je dôležitý pre vytvorenie si predstavy orientácii robota v priestore, čo je dôležité pri koordinovaní jeho pohybu.

V ľavej spodnej časti obrazovky sa nachádza tretia časť HUD v obdĺžnikovom okne. V ňom je zobrazená informácia o stave batérie ovládača pomocou štyroch zelených obdĺžnikov, kde každý obdĺžnik predstavuje hodnotu 25 %. Ďalej je zobrazené uplynutie času od pripojenia s robotom. Posledné dve zobrazené informácie sa týkajú stavu komunikácie medzi robotom a ovládačom. Prvou je intenzita signálu v decibeloch s referenciou na miliwatt a druhou je percentuálne vyjadrenie straty paketov, teda pomer rámcov, ktoré mali byť prenesené sieťou medzi tými, ktoré prenesené boli. Tieto informácie o stave komunikácie sú zobrazované z dôvodu, aby operátorovi poskytli možnosť rozhodnúť sa o ďalšom vzdialení sa s robotom od miesta, kde sa nachádza operátor bez straty komunikácie medzi ním, prípadne o odhalení rušivých elementov v okolí.

Posledná časť navrhnutého HUD sa nachádza v ľavej hornej časti obrazovky. Nachádza sa tu indikátor stavu batérie robota zobrazujúci informáciu číselne v jednotkách voltov a zároveň pod ňou sa nachádza obdĺžnikové okno. To je naplnené zelenou farbou v závislosti od veľkosti napätia batérie. Nakoniec sa tu nachádzajú ukazovatele aktuálnej rýchlosti a odberného prúdu, individuálne pre každý motor robota. Tie sú vyobrazené ako kolesá robota, kde prislúchajúci ukazovateľ rýchlosti a prúdu sa týka práve toho kolesa a motora, na ktorom je vyobrazený. Tieto indikátory sú interpretované rovnako ako indikátor stavu

batérie, obdĺžnikovými ramami, ktoré su naplnane zelenou farbou v zavislosti od rychlosti a odberaneho prudu. Dane informacie su zobrazovane najma preto, aby bola poskytnuta monost identifikovania zaseknutia kola. V pripade, e sa jedno z kolies robota zasekne a bude vyžadovany pohyb robota, indikator odberneho prudu zaseknuteho kola narastie, no rychlost bude nulova, pripadne minimalna, o poskytne operatorovi monost jednoducho zistit, e nastala prave takato situacia a adekvatne reagovat.



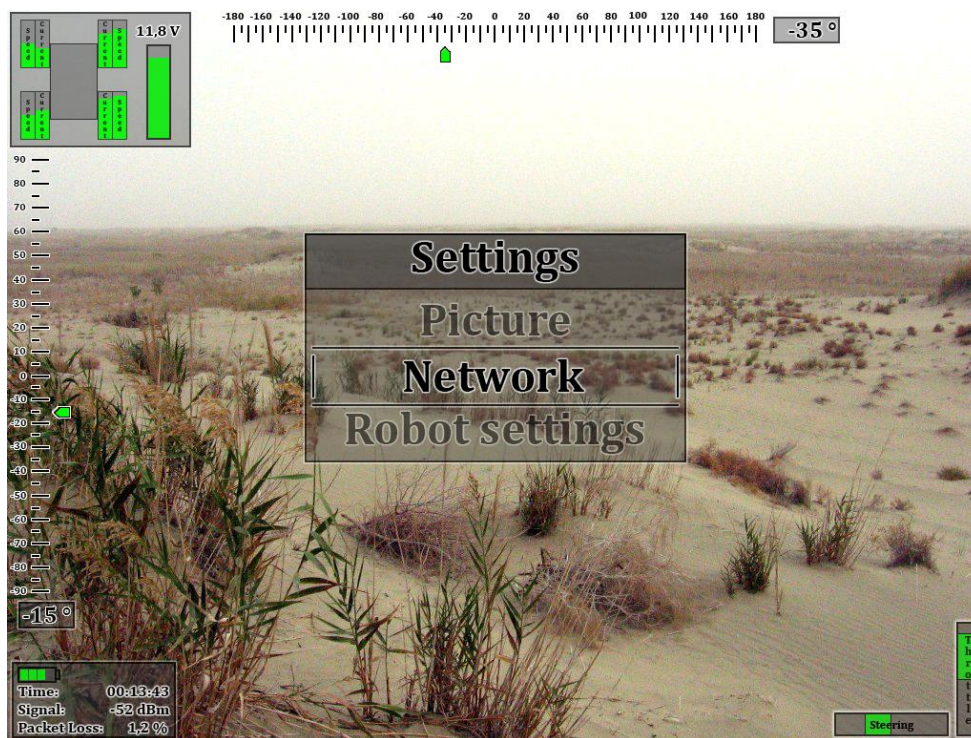
Obr. 24: Navrhnuty HUD

6.2.2 Menu

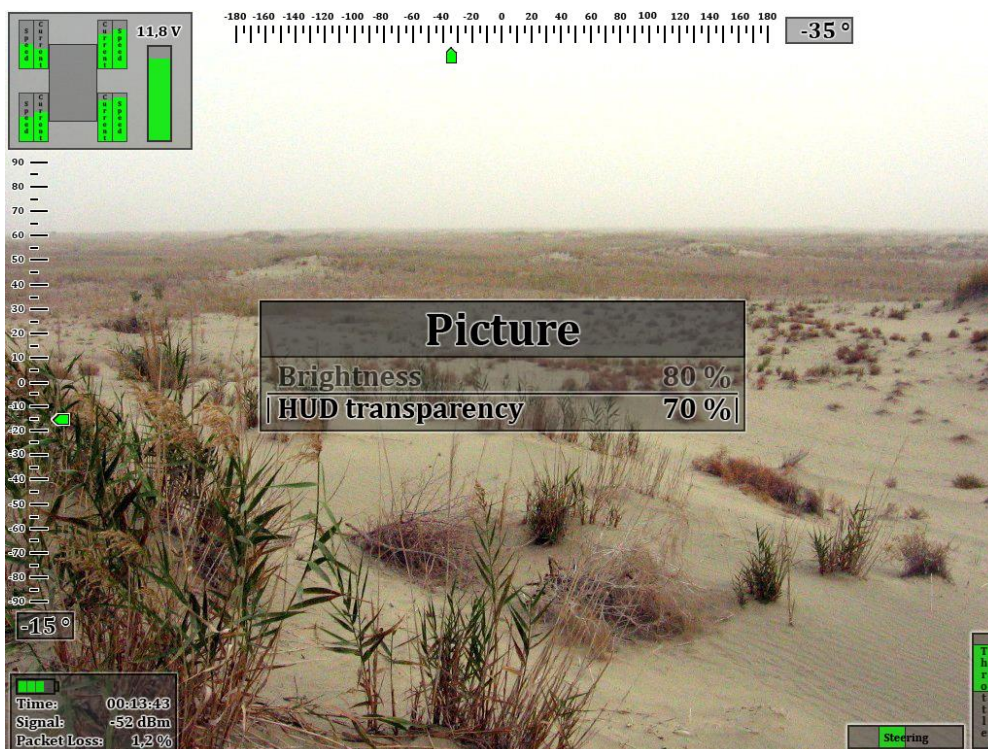
Menu ponuka monost vyberu z troch zakladnych skupin parametrizacie systemu (Obr. 25). Prvou polokou je „Picture“, ktora ponuka nastavenie uritych vlastnost obrazu. Druhou polokou je „Network“, nastavenia tykajuce sa siete. A poslednou polokou je „Robot settings“, ktora ponuka parametrizaciu tykajucu sa robota.

Na obrazku nišie je vyobrazene menu s voľbou poloky „Network“, kde je voľba zobrazovana zvraznenim vybratej poloky, jej ohranienim z bocnych stran a miernym zatmavenim ostatnych poloiek. Tato vizualizacia aktualnego vyberu je rovnaka v celej ponuke menu a v danych polokach. Vstup do menu je vykonany tlacidlom 2R, vystup z menu a navrat z poloky tlacidlom 2L, pohyb v menu

tlačidlami 2U a 2D a nakoniec úprava hodnôt tlačidlami 3L a 3R (viz. 6.3 Popis ovládacích tlačidiel).



Obr. 25: Zobrazené menu s výberom položky „Network“



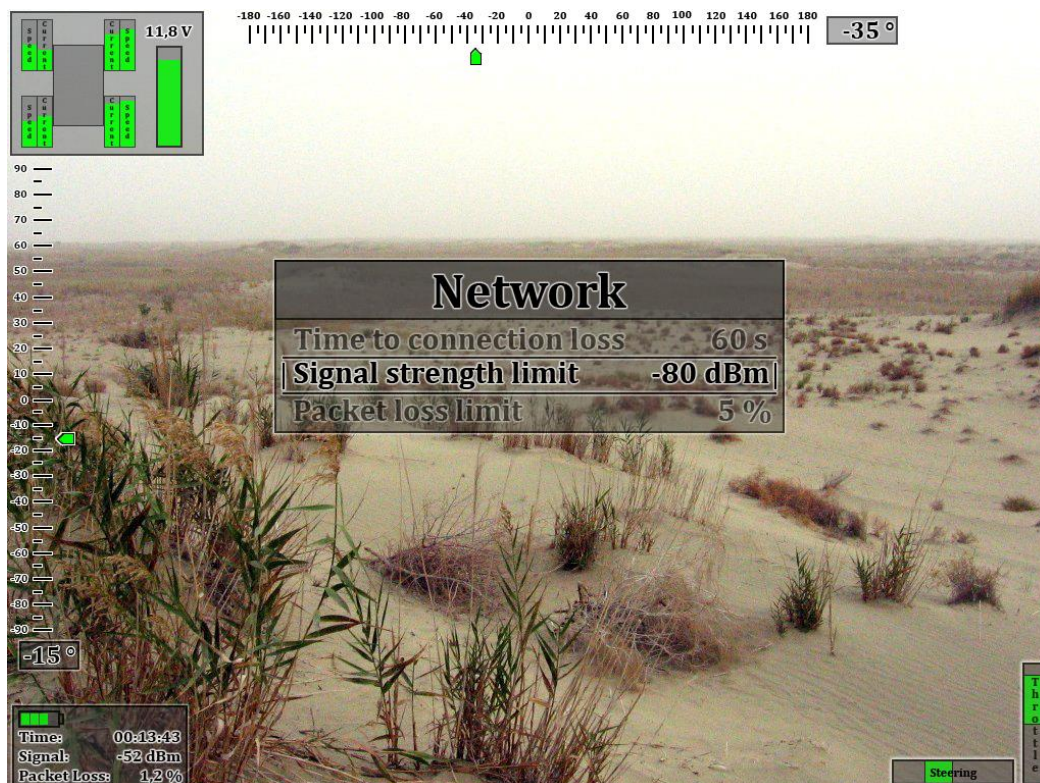
Obr. 26: Okno pre nastavenie obrazu s výberom položky „HUD transparency“

Nastavenie obrazu (Obr. 26) ponúka úpravu jasu „Brightness“ displeja ovládača a nastavenie priehľadnosti HUD „HUD transparency“. Obe hodnoty zmienovaných vlastností sa pohybujú v rozmedzí 0 až 100 %. Okno zobrazujúce túto ponuku je na obrázku vyššie. Zvyšovanie hodnoty je prevádzané tlačidlom 3R a znižovanie tlačidlom 3L, kde minimálna zmena hodnoty je 10 %.

Nastavenie siete (Obr. 27) sa skladá z možnosti nastavenia času „Time to connection loss“, po ktorom systém bude hlásiť stratu pripojenia (viz. 6.2.3 Stavové hlásenia) a nebude sa ďalej snažiť posielat' dáta. Tento čas je nastavovaný v jednotkách sekúnd.

Druhou položkou je nastavenie limitu intenzity signálu „Signal strength limit“ v jednotkách dBm, ktorý určuje, že po podkročení tejto hodnoty bude systém varovne hlásiť slabý signál.

Poslednou položkou je limit straty paketov „Packet loss limit“ vyjadrený v percentách, kde prekročenie tejto hodnoty má za následok varovné hlásenie veľkej stratovosti paketov. Varovné hlásenie slabého signálu a veľkej stratovosti paketov je spoločné (viz. 6.2.3 Stavové hlásenia).



Obr. 27: Okno pre nastavenie siete s výberom položky „Signal strength limit“

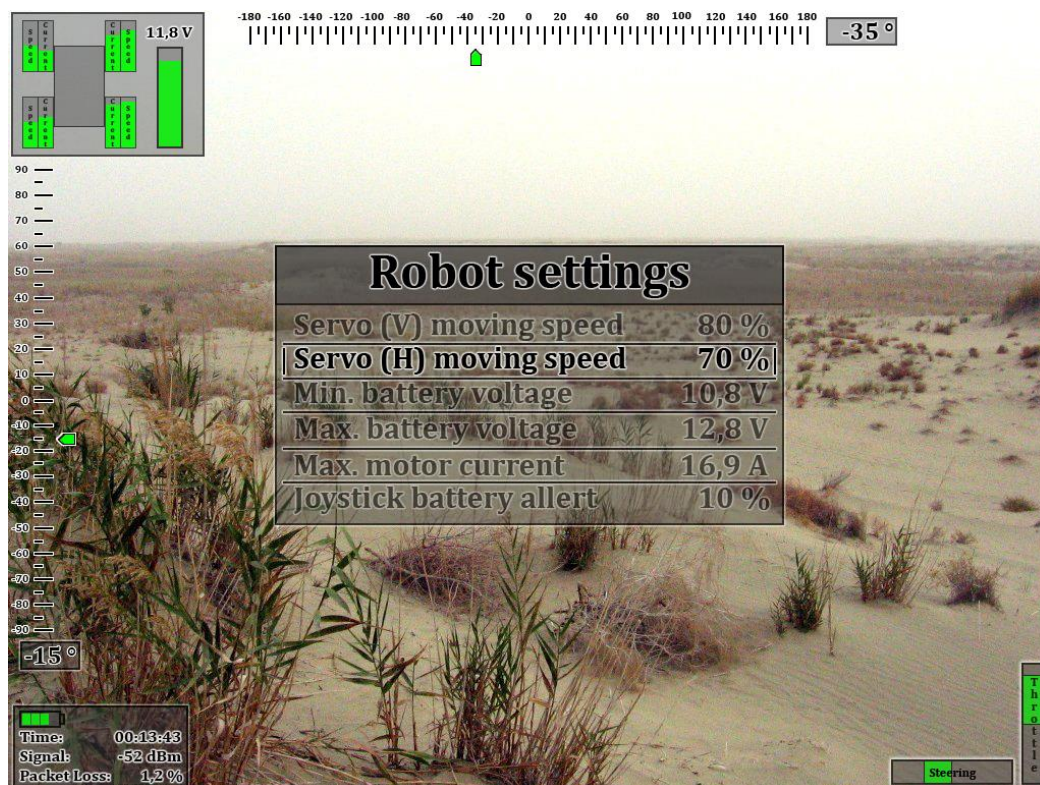
Nastavenie robota (Obr. 28) ponúka možnosť nastavenia rýchlosti pohybu serv v rozmedzí 1 až 100 %, kde zmena hodnoty je o veľkosti 1 %. Prvou položkou je

nastavenie rýchlosti pohybu vertikálneho servomotoru „Servo (V) moving speed“ a druhou položkou je nastavenie rýchlosti pohybu horizontálneho servomotoru „Servo (H) moving speed“.

Ďalšou položkou je „Min. battery voltage“, ktorej hodnota ovplyvňuje výskyt varovného hlásenia o napätí batérie robota (viz. 6.2.3 Stavové hlásenia). Jedná sa o minimálnu prípustnú hodnotu napätia batérie, ktorá je vyjadrená v jednotkách voltov. Zmena tejto hodnoty je v desatinách voltov. Nasledujúca položka „Max. battery voltage“ určuje naopak maximálnu povolenú hodnotu napätia batérie robota a je vyjadrená rovnako ako predchádzajúci parameter.

Predposlednou položkou je „Max. motor current“, ktorý určuje maximálnu prípustnú hodnotu odberaného prúdu motorom, kde po prekročení tejto hodnoty bude systém hlásiť preťaženie (viz. 6.2.3 Stavové hlásenia). Hodnota je vyjadrená v jednotkách ampérov a zmena tejto hodnoty je v desatinách ampéru.

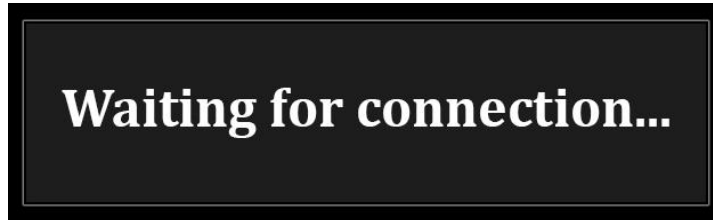
Posledným parametrom je „Joystick battery allert“, ktorého hodnota je vyjadrená v percentách a zmena je vykonávaná po jednotkách percent. Tento parameter určuje hodnotu zostávajúcej kapacity batérie ovládača, kde po podkročení tejto hodnoty systém zobrazí hlásenie o slabej batérii (viz. 6.2.3 Stavové hlásenia).



Obr. 28: Okno nastavení robota s výberom položky „Servo (H) moving speed“

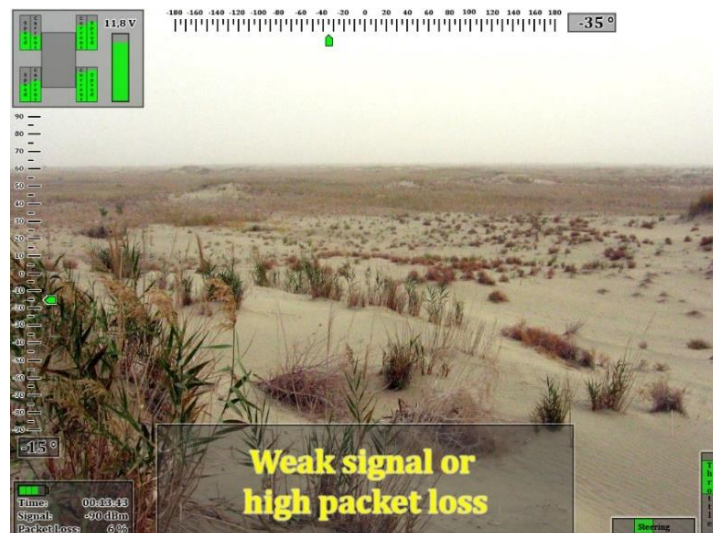
6.2.3 Stavové hlásenia

Stavové hlásenia sú zobrazované systémom v reakcii na určité situácie alebo stavy aby upovedomili operátora a ponúkli mu možnosť reagovať. Prvým stavovým hlásením je „Waiting for connection...“ (Obr. 29), ktoré je hlásené pred prvým pripojením k robotovi po zapnutí ovládača a oznamuje čakanie na pripojenie.



Obr. 29: Hlásenie o čakaní na pripojenie

Ďalším stavom, ktorý môže nastať je slabý signál alebo veľká stratovosť paketov „Weak signal or high packet loss“ (Obr. 30), čo informuje operátora, že je buď veľmi vzdialený od robota alebo je v blízkosti nejaké rušenie, ktoré spôsobuje vysokú stratovosť paketov. Hlásenie je spoločné pre obe príčiny, no pri pohľade na ukazovateľ intenzity signálu alebo stratovosti paketov je možné jednoducho identifikovať, ktorá zo situácií nastala. Nastavenie limitných hodnôt pre zobrazenie tohto hlásenia je popísané v kapitole 6.2.2 Menu.



Obr. 30: Varovanie o slabom signáli alebo veľkej stratovosti paketov

V prípade, že po hlásení spomenutom vyššie nastane strata signálu, systém pomocou hlásenia „Signal lost Waiting for signal...“ (Obr. 31) informuje operátora o tejto strate. Ide o situáciu kedy pripojenie je stále nadviazané, ale neprichádza žiadna odpoveď zo strany robota na odoslané pakety ovládačom. Nastavenie doby, po ktorú sa ovládač snaží odosielať pakety a čakať na odpoveď je popísané

v kapitole 6.2.2 Menu. Reakciou robota je okamžité zastavenie a čakanie na pokračovanie komunikácie.



Obr. 31: Hlásenie o strate signálu a čakaní naň

Posledným hlásením v rade týkajúceho sa komunikácie je hlásenie „Connection lost“ (Obr. 32), ktoré nastane po ubehnutí doby, po ktorú ovládač ešte čaká na odpoveď robota. Toto hlásenie informuje operátora o úplnej strate pripojenia. Reakciou robota na stratu pripojenia je zotrvanie v zastavenej polohe, ktorá nastane po strate signálu. Po strate pripojenia sa ovládač automaticky snaží opätovne nadviazať pripojenie hneď ako bude sieť prístupná.



Obr. 32: Hlásenie o strate pripojenia

Posledným hlásením, ktoré nadväzuje na informačné okno HUD v ľavom spodnom rohu obrazovky je „Battery Low !“ (Obr. 33). Informuje operátora o nízkej ostávajúcej hodnote kapacity batérie ovládača. Nastavenie hodnoty

kapacity batérie kedy sa má dané oznámenie zobrazit' je bližšie popísane v kapitole 6.2.2 Menu. Toto hlásenie je zobrazené po dosiahnutí nastavenej hodnoty po dobu 5 sekúnd sprevádzané jeho blikaním a je opätovne zobrazované po vyčerpaní každých 2 % kapacity batérie.



Obr. 33: Hlásenie o slabej batérii ovládača

Nasledujúca skupina hlásení sa týka informačného okna HUD v ľavom hornom rohu obrazovky a servomotorov pre polohovanie kamery. Prvým hlásením je „RR motor overload!“ zobrazeného na obrázku nižšie, ktoré informuje operátora o preťažení pravého zadného motora sprevádzaného rovnako aj sčervenáním daného kolesa v HUD. Toto upozornenie je sprevádzané blikaním prvku HUD, ale rovnako aj hlásenia na obrazovke po dobu 5 sekúnd.

Toto hlásenie sa môže vyskytovať v štyroch formách, kde zmenou je len identifikátor motora. Prvé písmeno hlásenia vyjadruje, či sa jedná o pravý motor (R - Right) alebo ľavý (L - Left) a druhé písmeno vyjadruje prednú pozíciu motora (F - Front) alebo zadnú (R - Rear).

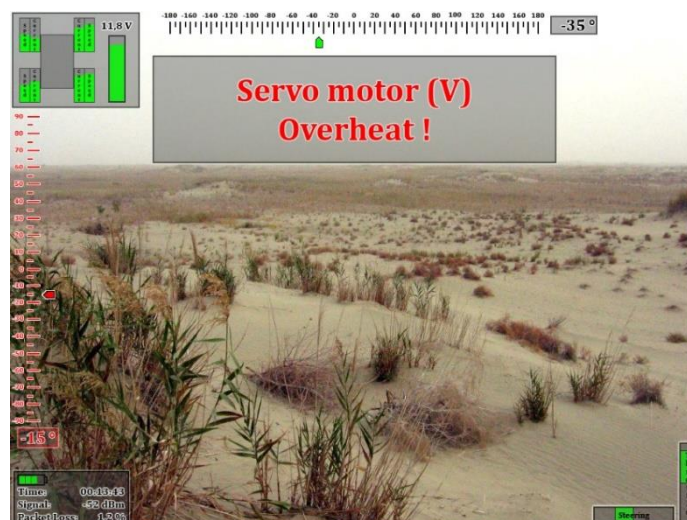
Nastavenie maximálnej hodnoty odberného prúdu, po ktorého prekročení sa zobrazí toto hlásenie je popísané v kapitole 6.2.2 Menu. Reakciou systému na toto hlásenie je okamžité vyradenie daného motora z prevádzky a pokus o jeho opätovné spustenie nastane po 60 sekundách, z dôvodu predpokladu prehriatia motoru a vymedzenia vhodnej doby na jeho schladenie.



Obr. 34: Hlásenie o preťažení pravého zadného motoru

Ďalšie hlásenie „Servo motor (V) Overheat !“ zobrazené na obrázku nižšie informuje operátora o prehriatí vertikálneho servomotoru pre polohovanie kamery. Pre identifikáciu servomotoru sa používajú rovnaké označenia ako vo vyššie spomínanom hlásení.

Upozornenie je zobrazované prostredníctvom hlásenia na obrazovke a sčervenania príslušnej osi natočenia kamery so súčasným blikaním týchto prvkov. Reakcia systému je okamžité vyradenie servomotoru z prevádzky pokiaľ nebude dosiahnutá akceptovateľná teplota.



Obr. 35: Hlásenie o prehriatí vertikálneho servo motoru

Posledné dve hlásenia sa týkajú stavu batérie robota. Prvým z nich je hlásenie „Battery overvoltage !“ (Obr. 36), ktoré ohlasuje prekročenie maximálneho limitu napätia batérie robota, teda prepätie. Bližšie informácie o nastavení hodnoty limitu sú popísané v kapitole 6.2.2 Menu.

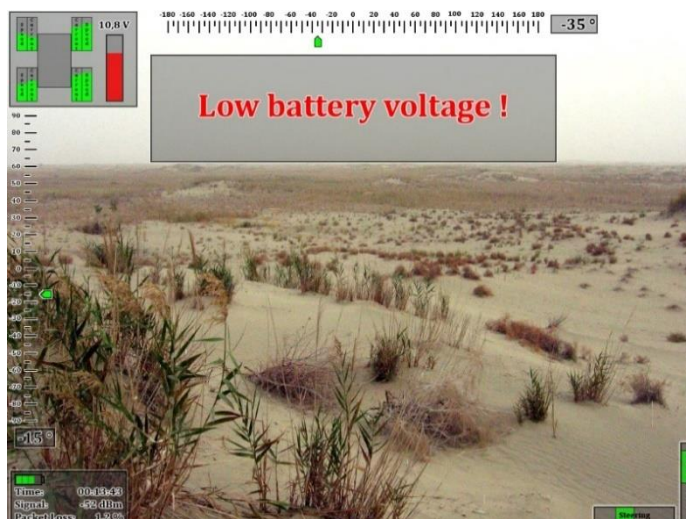
Hlásenie je sprevádzané sčervenaním indikátoru batérie a daným hlásením za súčasného blikania týchto prvkov. Varovanie pretrváva po dobu prekročenia limitu, kde vrátenie sa napätia batérie do stanovených medzí ukončuje toto hlásenie systému. Reakciou robota na situáciu je dočasné vyradenie všetkých pohonov z prevádzky, aby sa predišlo poruche batérii, kde po navrátení sa napätia do daných medzí je funkčnosť pohonov obnovená.



Obr. 36: Hlásenie o prepätí batérie robota

Posledným hlásením je „Low battery voltage !“, ktoré naopak ako vyššie zmienené upozornenie, oznamuje stav nízkeho napätia na batérii robota. Tento stav nastane po podkročení limitu minimálneho napätia, ktorého nastavenie je rovnako popísané v kapitole 6.2.2 Menu.

Hlásenie je sprevádzané sčervenaním indikátoru batérie a daným hlásením za súčasného blikania týchto prvkov pokým je daný limit podkročený.



Obr. 37: Hlásenie o nízkom napätí batérie robota

6.3 Popis ovládacích tlačídel

Dial'kový ovládač je vybavený tromi skupinami tlačídel (Obr. 38). Skupina 1 pozostáva z dvoch analógových páčok, kde ľavá páčka má označenie 1L (skupina 1, Left - ľavá) a páčka na pravej strane 1R (skupina 1, Right - pravá). Skupina 2 sa skladá zo štyroch smerových tlačídel (D-pad) a jednej LED diódy L1. Označenie tlačídel je analogické, ako u skupiny 1, kde navyše „U“ označuje „up“ smer hore a „D“ ako „down“ pre smer dole. Poslednú skupinu tvorí znova D-pad a LED dióda L2, kde značenie je analogické s predchádzajúcimi skupinami. Význam jednotlivých tlačídel je popísaný v tabuľke nižšie.

Tab. 11: Popis funkcií jednotlivých tlačídel

1L	Pohyb robota smerom doprava a doľava	1R	Pohyb robota smerom dopredu a dozadu
2U	Smerové tlačidlo hore	3U	Pohyb kamery hore
2D	Smerové tlačidlo dole	3D	Pohyb kamery dole
2L	Tlačidlo späť	3L	Pohyb kamery doľava/Menu -
2R	Potvrdzovacie tlačidlo/Menu	3R	Pohyb kamery doprava/Menu +
L1	Signalizácia stlačenia tlačidla skupiny 2	L2	Signalizácia stlačenia tlačidla skupiny 3



Obr. 38: Označenie tlačídel ovládača

7 IMPLEMENTÁCIA APLIKÁCIE

Aby bolo možné navrhnutú aplikáciu vytvoriť, je nutné vybrať vhodnú grafickú sadu programových nástrojov (tzv. grafický toolkit), prostredníctvom ktorého bude možné implementovať všetky navrhnuté funkcie aplikácie. Na výber je mnoho grafických toolkitov, no vzhľadom ku komplexnosti danej aplikácie, je vhodné vyberať z tých rozvinutejších. Ďalšími požiadavkami je určite voľne šíriteľná distribúcia a podpora hardvérovej akcelerácie z dôvodu nedostatočného výkonu procesora Raspberry Pi vzhľadom k náročnosti aplikácie. Takými to zastupiteľmi sú grafické toolkity GTK a Qt.

Najdôležitejšou požiadavkou je prenos videa v reálnom čase z kamery robota, ktorý bude potrebné do daného toolkitu implementovať prostredníctvom aplikačného rámca (tzv. frameworku). Z najdostupnejších riešení, ktoré ponúkajú dostatočnú dokumentáciu aj komplexnosť, je multimediálny framework GStreamer, FFmpeg a multimediálny prehrávač VLC, ktorý ponúka knižnice pre implementáciu do daných toolkitov.

Po preskúmaní kombinácií spomenutých toolkitov a frameworkov, bol vzhľadom k dostupnej dokumentácii, podpore hardvérovej akcelerácie vykresľovania videa a vytvorenej grafickej aplikácie, ale aj vzhľadom k použitému programovaciemu jazyku toolkitu, vybraný grafický toolkit Qt. Qt ponúka programovanie v jazyku C++ oproti toolkitu GTK, ktorý ponúka programovací jazyk C. Vzhľadom k subjektívnemu postoju, prívetivejšieho a prehľadnejšieho programovania v jazyku C++, oproti nízkoúrovňového jazyka C, bol aj toto dôvod k výberu toolkitu Qt.

7.1 Rozhranie pre programovanie aplikácií OpenGL a EGL

Hardvérová akcelerácia renderovania je dosiahnutá pomocou použitia rozhrania pre programovanie aplikácií (API - Application programming interface) OpenGL, ktoré je typicky používané pre interakciu s grafickou jednotkou. Ide o jazykovo a platformovo nezávislé API, ktoré je definované ako sada funkcií a konštánt, ktoré môžu byť volané klientskými programami. Vytváranie kontextu je ponechávané na okenný systém.

V tomto konkrétnom prípade je použitá podmnožina OpenGL, ktorá je nazývaná OpenGL for Embedded Systems (OpenGL ES alebo GLES) vo verzii 2.0, ktorá už ako vyplýva z názvu bola vytvorená pre vstavané systémy akým je aj Raspberry Pi. Ako jeho softvérová implementácia v tomto prípade slúži Mesa 3D. V podstate ide o preklad OpenGL kontextu pre konkrétny ovládač grafického hardvéru. Preložený kontext je následne spracovaný v jadre pomocou rozhrania

DRM (Direct Rendering Manager), ktorý priamo komunikuje s GPU a jej framebufferom.

Aby Qt mohlo pracovať s OpenGL, využíva rozhranie EGL. Ide o rozhranie medzi renderovacími API akým je aj OpenGL a natívnym okenným systémom. Vo svojej podstate poskytuje mechanizmus pre vytváranie povrchu, na ktorý následne API ako OpenGL môžu kresliť a vytvárať grafický kontext. Qt používa EGL pre manažment kontextu a povrchu.

Náhrada okenného systému, ktorý nie je v operačnom systéme Raspbian nainštalovaný je zabezpečená pomocou EGLFS. Tento platformový zásuvný modul zabezpečuje beh Qt aplikácie na vrchu EGL a OpenGL ES 2.0 bez potreby použitia skutočného okenného systému (ako napr. X11 alebo Wayland). EGLFS nastaví prvé okno v najvyššej vrstve na celú obrazovku. Okno je zároveň zvolené ako koreňové, kde všetky ostatné následne vytvorené okná sú zlúčené s týmto oknom. Toto je dôležité z dôvodu, že pri použití EGLFS je vždy možná prítomnosť len jedného okna.

Qt poskytuje aj prvky (napr. trieda QOpenGLWidget), ktoré majú implementované vytváranie OpenGL kontextu a preto nie je potrebné ďalej softvérovo riešiť použitie OpenGL API. Na obrázku nižšie je zobrazená hierarchia hardvérovo akcelerovaného vykresľovania na platforme Raspberry Pi prostredníctvom Qt.



Obr. 39: Hardvérová akcelerácia pomocou OpenGL

7.2 Príprava platformy Raspberry Pi

Prvým krokom je inštalácia operačného systému Raspbian, ktorého obraz je stiahnutelný na oficiálnych stránkach spoločnosti Raspberry Pi Foundation vo formáte ZIP. Druhým krokom je rozbalenie a skopírovanie stiahnutého obrazu na SD kartu prostredníctvom daného príkazu v príkazovom riadku:

```
unzip -p 2019-04-08-raspbian-stretch-lite.zip | sudo dd  
of=/dev/mmcblk0 bs=4M conv=fsync
```

Ďalším krokom je príprava Raspberry Pi pre krížovú kompiláciu aplikácie s podporou hardvérovej akcelerácie OpenGL s využitím EGLFS. Pomocou príkazu:

```
sudo raspi-config
```

je spustené konfiguračné okno Raspberry Pi, v ktorom je potrebné nastaviť tzv. bootovanie do konzoly a nastaviť pamäť GPU na hodnotu 256 MB. Ďalej je vykonaná aktualizácia a reštart systému Raspbian:

```
sudo rpi-update
reboot
```

Nasleduje aktualizácia balíčkov systému a inštalácia vývojárskych balíčkov potrebných pre chod aplikácie:

```
sudo apt-get update
sudo apt-get build-dep qt4-x11
sudo apt-get build-dep libqt5gui5
sudo apt-get install libudev-dev libinput-dev libts-dev
libxcb-xinerama0-dev libxcb-xinerama0
```

Následne je vytvorený cieľový priečinok, v ktorom sa bude nachádzať Qt toolkit potrebný pre chod aplikácie a nastavenie vlastníckych práv pre užívateľa „pi“, ktoré budú využité neskôr pri synchronizácii priečinkov:

```
sudo mkdir /usr/local/qt5pi
sudo chown pi:pi /usr/local/qt5pi
```

Po synchronizácii priečinkov, ktorá je popísaná v nasledujúcej podkapitole 7.3 je potrebné aktualizovať zariadenie, aby linker dokázal nájsť Qt knižnice:

```
echo /usr/local/qt5pi/lib | sudo tee
/etc/ld.so.conf.d/qt5pi.conf
```

Poslednou časťou tohto kroku je oprava EGL/GLES knižníc. Raspberry Pi má nainštalované knižnice libEGL a libGLESv2 verzie Mesa. To má za následok, že si Qt vyberá práve tieto knižnice kvôli ich prítomnosti v priečinku „*/usr/lib/arm-linux-gnueabi/hf*“, namiesto tých správnych, ktoré sa nachádzajú v priečinku „*/opt/vc/lib*“. Pôvodné knižnice sú postačujúce pre X11 (okenný systém) desktopové aplikácie, no nie pre výkonné OpenGL aplikácie, ktoré nie sú založené na okenných operačných systémoch, akým je práve nainštalovaný operačný systém Raspbian. To je dosiahnuté nasledujúcimi príkazmi v príkazovom riadku:

```
sudo ln -s /opt/vc/lib/libEGL.so /usr/lib/arm-linux-
gnueabi/hf/libEGL.so.1.0.0
sudo ln -s /opt/vc/lib/libGLESv2.so /usr/lib/arm-linux-
gnueabi/hf/libGLESv2.so.2.0.0
sudo ln -s /opt/vc/lib/libbrcmEGL.so
/opt/vc/lib/libEGL.so
sudo ln -s /opt/vc/lib/libbrcmGLESv2.so.2
/opt/vc/lib/libGLESv2.so
```

```
sudo ln -s /opt/vc/lib/libEGL.so /opt/vc/lib/libEGL.so.1
sudo ln -s /opt/vc/lib/libGLESv2.so
/opt/vc/lib/libGLESv2.so.2
```

Ďalším krokom je inštalácia frameworku GStreamer, ktorá nie je možná pomocou balíčkovacieho systému *apt-get*, z dôvodu poskytovania zastaralej verzie tohto frameworku, preto je potrebné jeho stiahnutie z verzovacieho systému GIT. Potrebné je nainštalovanie základného frameworku verzie 1.14 a tejto verzii odpovedajúcim zásuvných modulov:

- `gst-plugins-base`
- `gst-plugins-bad`
- `gst-plugins-good`
- `gst-omx`

Inštalácia frameworku a modulov prebieha pomocou nasledovných príkazov v príkazovom riadku:

```
./autogen.sh
make
make install
```

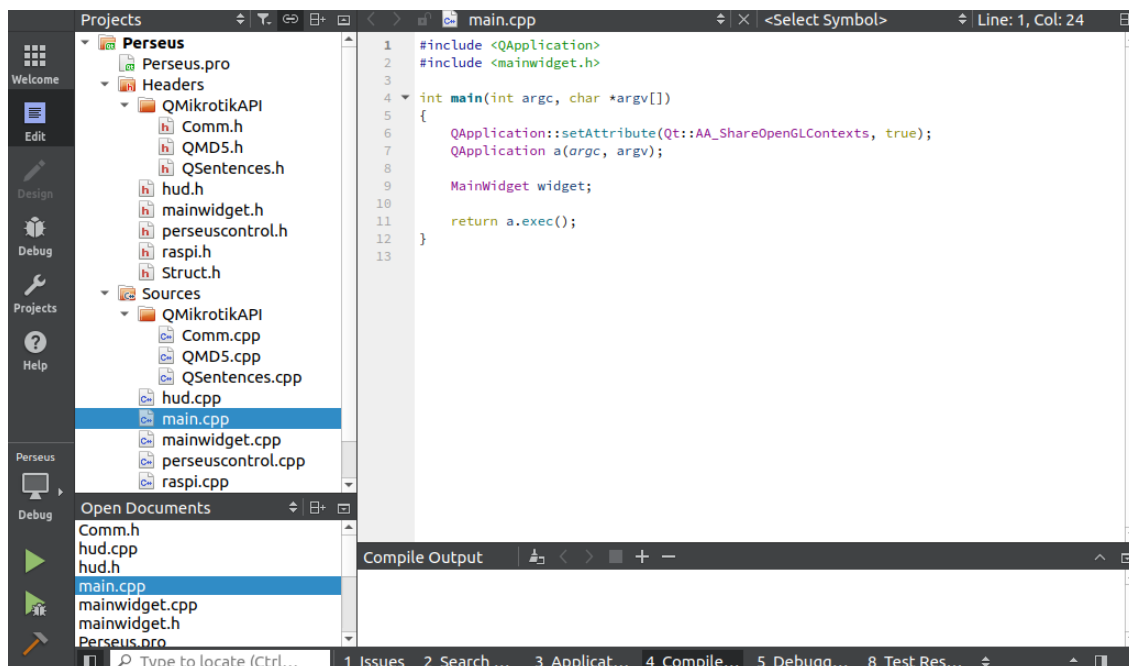
Posledným krokom prípravy platformy Raspberry Pi je nastavenie niektorých premenných prostredia, ktoré sú potrebné pre správne fungovanie frameworku GStreamer a toolkitu Qt. Tieto premenné sú pridané do súboru „*/etc/environment*“, aby boli prístupné každému užívateľovi:

```
QT_QPA_PLATFORM=eglfs
QT_QPA_PLATFORM_PLUGIN_PATH=/usr/local/qt5pi/plugins/plat
tforms
LD_LIBRARY_PATH=/usr/local/qt5pi/lib
QT_QPA_EGLFS_PHYSICAL_WIDTH=155
QT_QPA_EGLFS_PHYSICAL_HEIGHT=86
GST_PLUGIN_PATH=/usr/lib/gstreamer-1.0/:/home/pi/gst-
plugins-base/:/home/pi/gst-plugins-bad/:/home/pi/gst-
plugins-good/:/home/pi/gst-omx/
```

7.3 Grafická sada programových nástrojov Qt a vývojové prostredie

Qt je voľne šíriteľná, multiplatformová sada programových nástrojov pre tvorbu grafického užívateľského rozhrania a multiplatformových aplikácií, ktoré môžu byť rovnako postavené na rôznych hardvérových platformách. Ponúka aj vývojové prostredie pod názvom Qt Creator a Qt Designer, kde je pre vytvorenie tejto

aplikácie použité vývojové prostredie Qt Creator (Obr. 40) pod operačným systémom Ubuntu 18.04 LTS.



Obr. 40: Vývojové prostredie Qt Creator

Ako už bolo spomínané v podkapitole vyššie, pre vytvorenie možnosti krížovej kompilácie je nutná séria krokov aj na zariadení, na ktorom táto kompilácia bude prebiehať. Prvým krokom je vytvorenie pracovného priečinku a stiahnutie programových nástrojov a utilít, ktoré poskytnú možnosť krížovej kompilácie z verzovacieho systému GIT:

```
mkdir ~/raspi
cd ~/raspi
git clone https://github.com/raspberrypi/tools
```

Druhým krokom je vytvorenie tzv. priečinku „sysroot“, ktorý slúži ako koreňový adresár pre lokáciu potrebných hlavičkových súborov a knižníc pre kompiláciu. Následne je daný adresár synchronizovaný s potrebnými priečinkami Raspberry Pi, čo je vykonané pomocou nasledovných príkazov s IP adresou zariadenia, ktorá nahrádza „raspberrypi.local“:

```
mkdir sysroot sysroot/usr sysroot/opt
rsync -avz pi@raspberrypi.local:/lib sysroot
rsync -avz pi@raspberrypi.local:/usr/include sysroot/usr
rsync -avz pi@raspberrypi.local:/usr/lib sysroot/usr
rsync -avz pi@raspberrypi.local:/opt/vc sysroot/opt
```

Nasleduje upravenie synchronizovaných symbolických adries z Raspberry Pi, aby boli relatívne pomocou stiahnutelného skriptu:

```
wget https://raw.githubusercontent.com/Kukkimonsuta/rpi-buildqt/master/scripts/utils/sysroot-relativelinks.py
chmod +x sysroot-relativelinks.py
./sysroot-relativelinks.py sysroot
```

Ďalším krokom stiahnutie, konfigurácia a inštalácia Qt toolkitu na zariadení, z ktorého bude prebiehať kompilácia. Pri konfigurácii je potrebné špecifikovať pomocou argumentov, niektoré parametre. Medzi najdôležitejšie argumenty patrí „*-opengl es2*“, ktorý špecifikuje použitie API OpenGL ES 2.0. Ďalším dôležitým parametrom je „*-device*“, ktorý špecifikuje zariadenie, na ktoré je daný kód kompilovaný. Parameter „*-device-option*“ nastavuje prídavné premenné, v tomto prípade premennú prostredia „*CROSS_COMPILE*“, ktorá obsahuje cestu k programovým nástrojom a utilitám potrebným pre krížovú kompiláciu. Nasleduje parameter „*-sysroot*“ špecifikujúci cestu ku koreňovému adresáru obsahujúcemu hlavičkové súbory a knižnice zariadenia Raspberry Pi. Poslednými dôležitými argumentmi je špecifikácia prefixov. Prvý prefix je špecifikovaný parametrom „*-prefix*“, ktorý odkazuje na cestu do inštalačného adresára zariadenia obsahujúceho binárne súbory a knižnice toolkitu Qt, ktorý slúži pre beh aplikácie. Ďalší prefix špecifikovaný argumentom „*-extprefix*“ obsahuje cestu adresára, do ktorého prebieha inštalácia toolkitu. Tento adresár bude následne nahraný do zariadenia Raspberry na miesto kde odkazuje prefix „*-prefix*“. Posledným argumentom je „*-hostprefix*“, ktorý dovoľuje separáciu hositeľských nástrojov slúžiacich pre kompiláciu od binárnych súborov, ktoré nie sú pre cieľové zariadenie dôležité. Po upresnení daného argumentu sú tieto súbory nainštalované do tohto adresára namiesto do adresára špecifikovaného prefixom „*-extprefix*“.

```
git clone git://code.qt.io/qt/qtbase.git -b 5.12.2
cd qtbase
./configure -release -opengl es2 -device linux-rasp-pi2-
g++ -device-option CROSS_COMPILE=~/.raspi/tools/arm-
bcm2708/gcc-linaro-arm-linux-gnueabihf-raspbian/bin/arm-
linux-gnueabihf- -sysroot ~/.raspi/sysroot -opensource -
confirm-license -make libs -prefix /usr/local/qt5pi -
extprefix ~/.raspi/qt5pi -hostprefix ~/.raspi/qt5 -v -no-
use-gold-linker
make
make install
```

Nasleduje stiahnutie a nainštalovanie potrebného modulu pre Qt toolkit, ktorý slúži pre vývoj multimediálnych aplikácií.

```
git clone git://code.qt.io/qt/qtmultimedia.git -b 5.12.2
cd qtmultimedia
~/raspi/qt5/bin/qmake
make
make install
```

Posledným krokom je stiahnutie nainštalovaného Qt toolkitu na Raspberry Pi pomocou nasledujúceho príkazu:

```
rsync -avz qt5pi pi@raspberrypi.local:/usr/local
```

Nakoniec je potrebné prídanie nového zariadenia do vývojového prostredia Qt Creator, s požadovaným kompilátorom a debuggerom, ktoré sú prítomné v zložke programových utilít a nástrojov spomínaných vyššie. Výsledkom je možnosť krížovej kompilácie, stiahnutie spustiteľného súboru do zariadenia Raspberry Pi a jeho následné spustenie zo zariadenia, kde prebieha kompilácia.

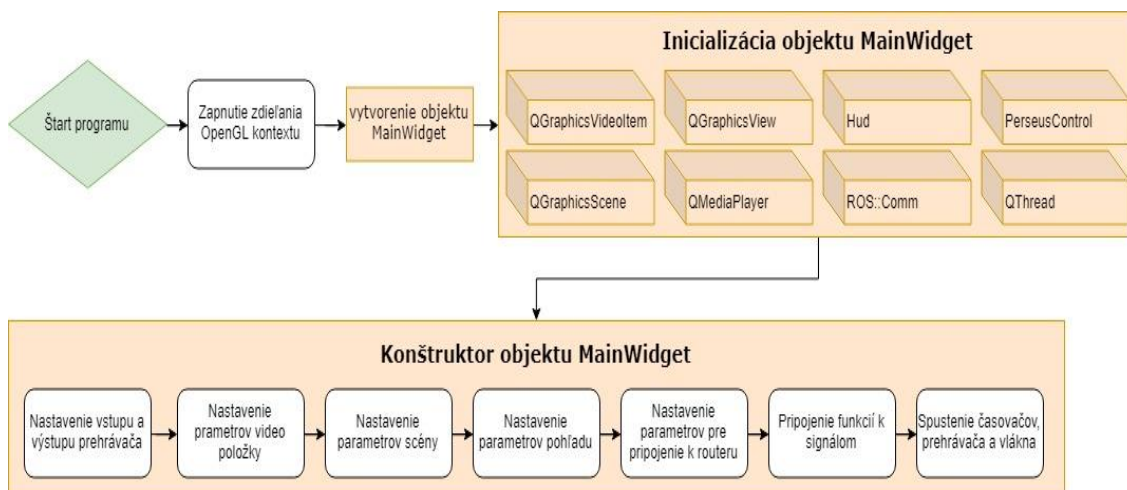
7.4 Softvérová realizácia

Realizácia pozostáva z vytvorenia nových tried, ktoré budú realizovať určité elementárne funkcie celej aplikácie, použitia integrovaných tried a knižníc Qt. Okrem týchto je použitá aj knižnica tretej strany WiringPi napísaná v jazyku C, slúžiaca pre jednoduchšie pracovanie s pinmi GPIO Raspberry Pi vyvinutá pre použitie priamo na operačnom systéme Raspbian. Poslednou využitou súčasťou tretej strany pri vytváraní softvéru je API pre komunikáciu so smerovačom MikroTik. Ide o API napísané priamo pomocou Qt SDK (Software development kit), ktoré je vytvorené pre použitie s triedami ponúkanými samotným Qt. API je využité pre riadenie komunikácie protokolom, ktorý používa ROS (Routerboard OS), čo umožňuje jednoduchú komunikáciu so smerovačom pre zber informácií, konfiguráciu a manažment.

7.4.1 Modul main.c a trieda MainWidget

Ide o hlavný modul zabezpečujúci celý beh aplikácie, ktorý je kompilovaný ako spustiteľná aplikácia a obsahuje len nastavenie zdieľania OpenGL kontextu a vytvorenie inštancie triedy *MainWidget*, čo je možné vidieť na obrázku nižšie.

Trieda *MainWidget* je základnou triedou zabezpečujúcou všetky funkcie aplikácie, pričom ostatné použité objekty sú vytvorené vo vnútri tejto triedy. Na obrázku nižšie sa nachádza stavový diagram aplikácie po spustení vyobrazujúci najdôležitejšie použité triedy, ktorých inštancie sú vytvorené hneď po vytvorení inštancie triedy *MainWidget* a sled úkonov vykonaných jej konštruktorom.



Obr. 41: Inicializačný sled udalostí po spustení aplikácie

Na obrázku nižšie je vyobrazená nadväznosť tried použitých pri prehrávaní videosignálu na obrazovke, kde vstup do triedy *QMediaPlayer* je sieťová adresa videosignálu a výstup triedy *QGraphicsView* je zobrazovaný na obrazovke.



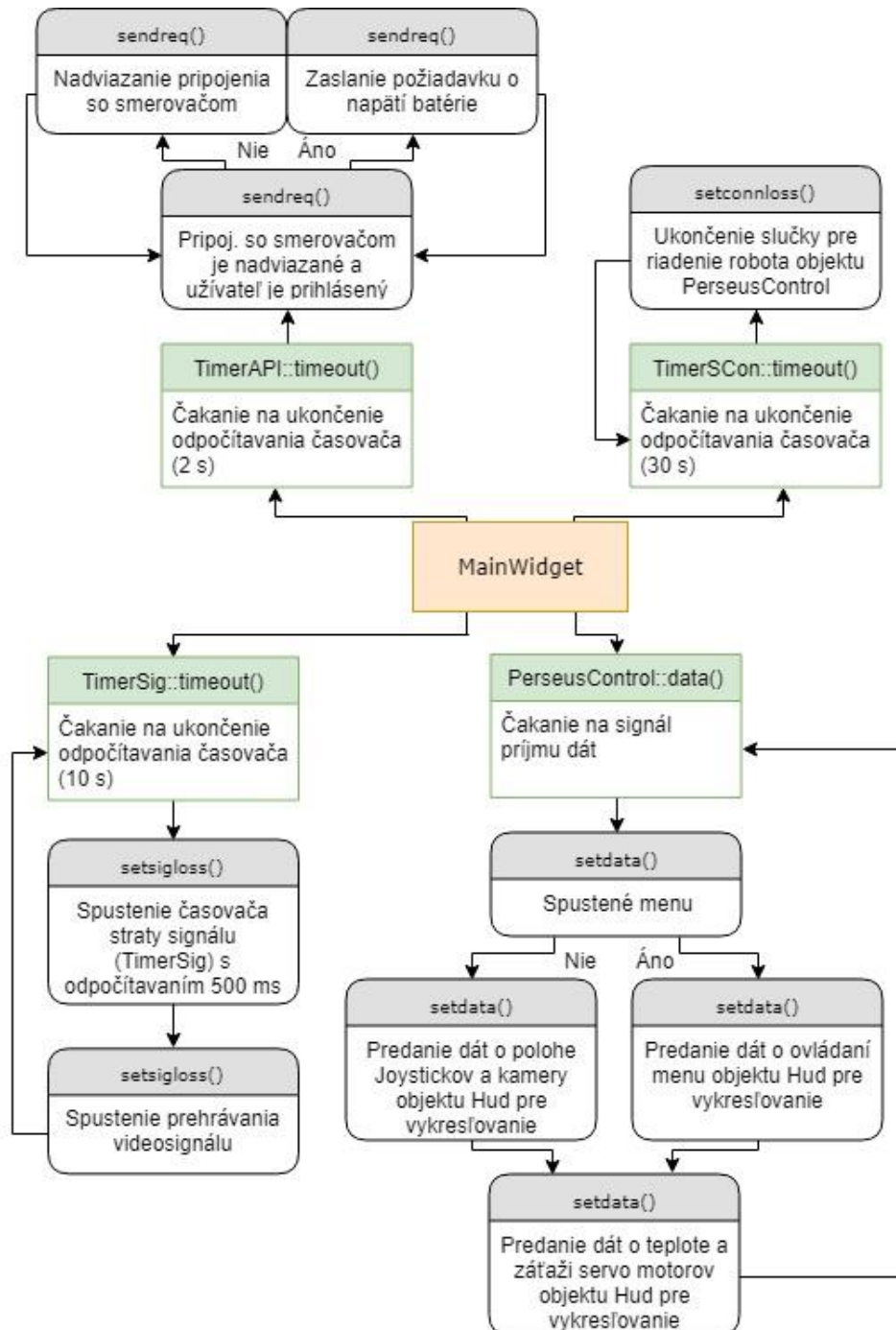
Obr. 42: Hierarchia výstupov a vstupov tried pri prehrávaní videosignálu

Po inicializačnom slede udalostí spomínaných vyššie sa vytvorený objekt *MainWidget* dostáva do stavu čakania na príjem signálov, ktoré inicializujú spustenie špecifických funkcií priradených k daným signálom (Obr. 43 a Obr. 44). V týchto funkciách je obsiahnuté celé fungovanie aplikácie.

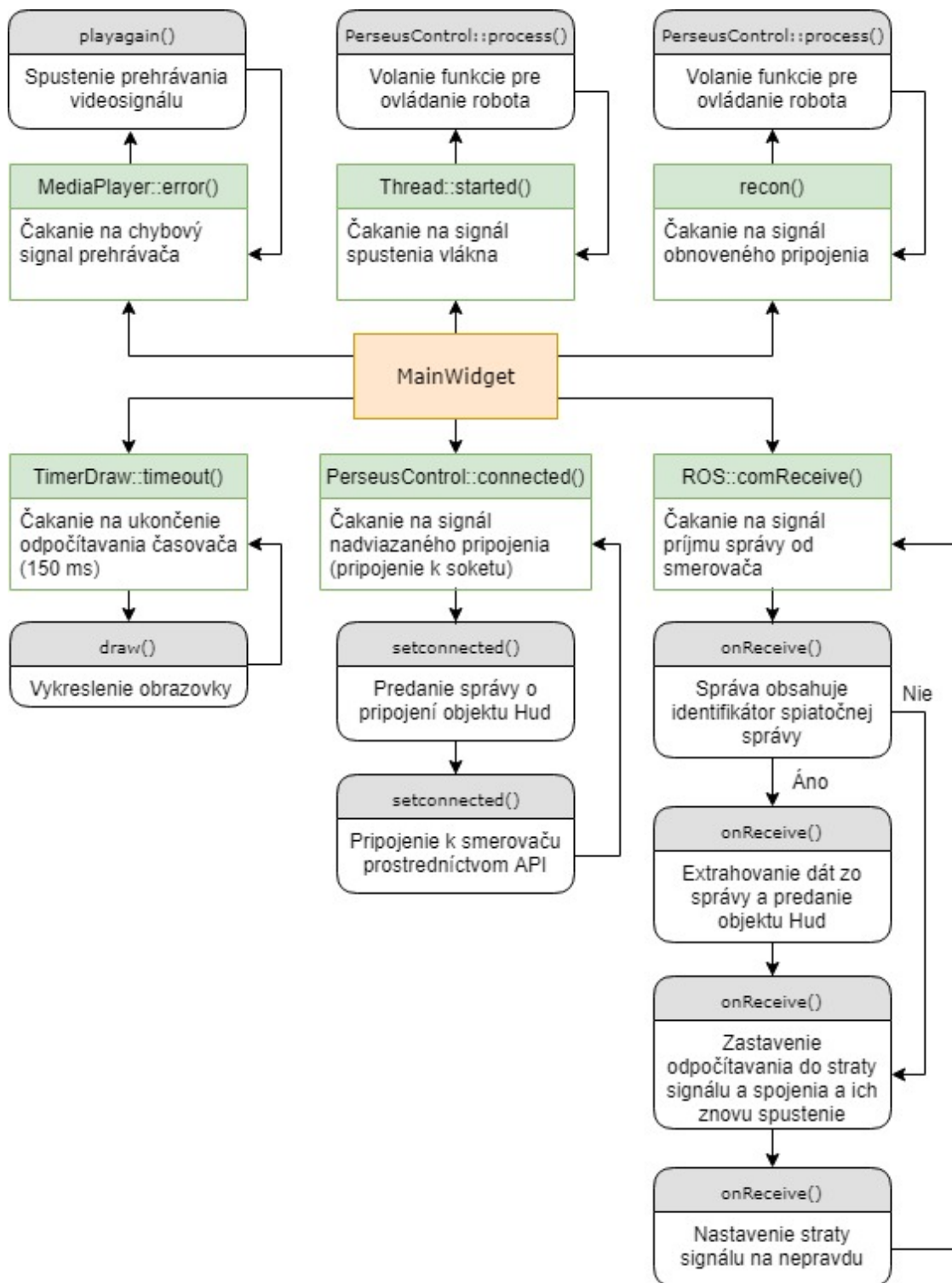
Aplikácia pracuje v dvoch vláknach, kde prvým a zároveň hlavným vláknom je vnútorná logika triedy *MainWidget*. Druhé vlákno je vytvorené práve v tejto triede a po jeho spustení je vygenerovaný signál „started“ triedy *QThread*, ktorému je priradené volanie funkcie „process“ triedy *PerseusControl* (Obr. 44). Táto funkcia má na starosti riadenie robota a spracovanie logiky ovládacích prvkov diaľkového ovládača.

Ďalšími použitými signálmi sú signály generované triedou *PerseusControl* poskytujúce dáta a informácie slúžiace pre vykresľovanie UI aplikácie. Ďalším je chybový signál generovaný triedou *QMediaPlayer*, kde reakciou je pokus o znovu spustenie prehrávania. Použité sú aj signály, ktoré generujú časovače svojím ukončením odpočítavania špecifického času. Jedným z nich je signál časovača podnecujúci odoslanie požiadavky o stave napätia batérie prostredníctvom implementovaného API pre komunikáciu so smerovačom, čo zabezpečuje použitá trieda *ROS*. Po prijatí odozvy, teda odpovedi smerovača o stave napätia je prijatý

signál o prijatí správy, kde je následne táto správa spracovaná a sú z nej extrahované relevantné dáta. Ďalším je signál časovača podnecujúci vykresľovanie grafiky aplikácie. Zostávajúce signály časovačov slúžia pre spravovanie pripojenia, ktoré sú generované ak neprichádza odozva zo strany smerovača. Posledným signálom je signál generovaný triedou *MainWidget* po obnovení pripojenia, ktorý sa pokúsi znova nadviazať spojenie s robotom prostredníctvom soketu a to znova zavolaním vyššie spomenutej funkcie „process“ triedy *PerseusControl*.



Obr. 43: Vnútrná logika triedy *MainWidget*



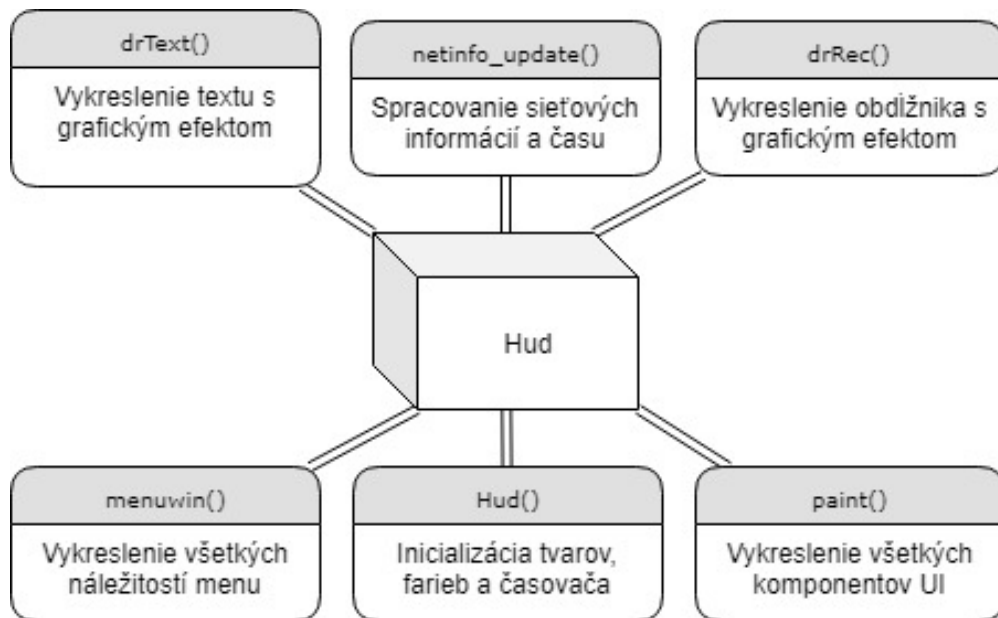
Obr. 44: Vnútoraná logika triedy MainWidget

7.4.2 Trieda Hud

Účelom tejto triedy je vykresľovanie grafiky aplikácie. Jej inštancia je vytvorená v triede *MainWidget*. Najdôležitejšie funkcie triedy sú zobrazené na obrázku nižšie.

Princíp fungovania je vo svojej podstate len cyklické volanie funkcie „paint“, ktorá vykresľuje všetku potrebnú grafiku. Trieda *Hud* je potomkom triedy *QGraphicsItem*, pričom spomínaná funkcia „paint“ je od nej zdedená a je upravená, aby vykonávala požadované funkcie. Funkcia je nepriamo volaná z hlavnej triedy *MainWidget* signálom časovača *TimerDraw* prostredníctvom funkcie „update“ patriacej triede *QGraphicsItem*.

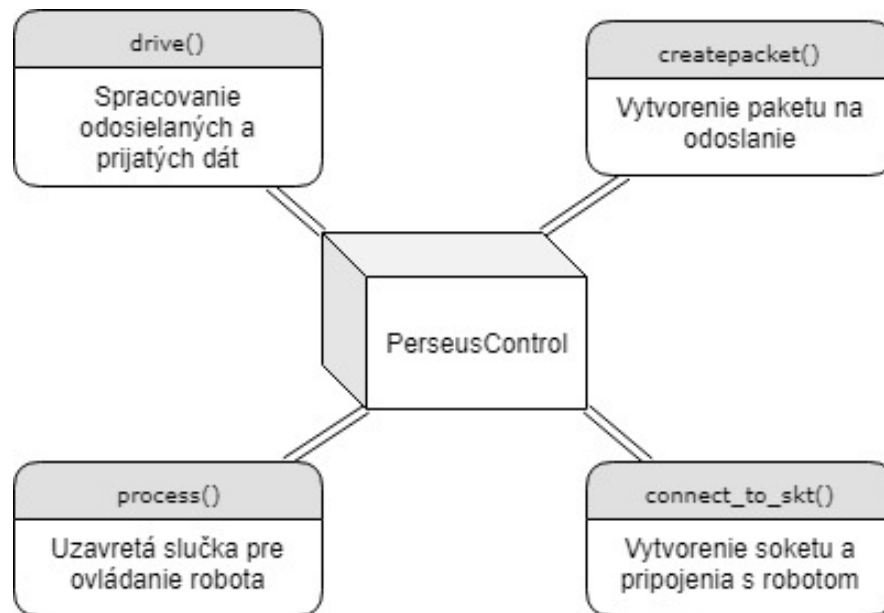
Funkcia „netinfo_update“ je volaná cyklickým odpočítavaním časovača integrovaného do triedy *Hud* v slučke o dĺžke jednej sekundy. Všetky ostatné funkcie slúžia pre väčšiu prehľadnosť kódu a kvôli častému používaniu rovnakej časti kódu, ktorý je v nich obsiahnutý.



Obr. 45: Trieda *Hud* a jej najdôležitejšie funkcie

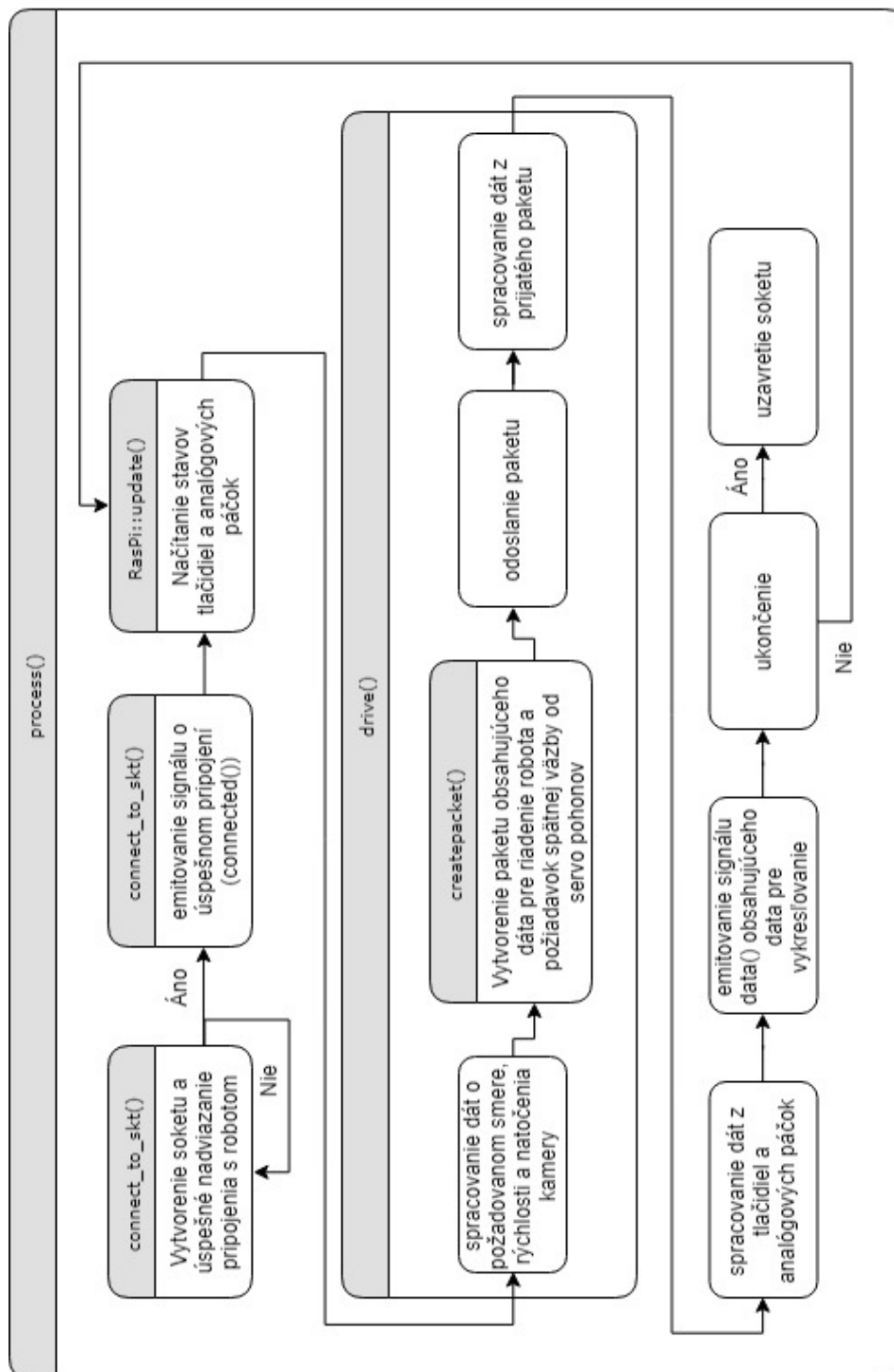
7.4.3 Trieda *PerseusControl*

Daná trieda je vytvorená za účelom riadenia robota. Inštancia triedy je vytvorená v triede *MainWidget* ako už bolo spomenuté vyššie a jej základné fungovanie spočíva vo funkcii „process“, ktorá je volaná pri štarte nového vlákna a v prípade výpadku spojenia pri opätovnom nadviazaní spojenia prostredníctvom príslušného signálu. Tieto volania sú iniciované signálmi triedy *MainWidget* a *Qthread*. Základnou funkciou, ktorú trieda *PerseusControl* zabezpečuje je vytvorenie komunikácie medzi sériovou linkou smerovača MikroTik a Raspberry Pi prostredníctvom soketu a nasledujúca komunikácia prostredníctvom tohto pripojenia za účelom získavania spätnej väzby a riadenia robota. Poslednou dôležitou funkciou je spracovanie spätnej väzby od tlačidiel diaľkového ovládača prostredníctvom triedy *RasPi*, ktorej inštancia je vytvorená vo vnútri tejto triedy. Na obrázku nižšie sú vyobrazené základné funkcie triedy *PerseusControl*.



Obr. 46: Trieda PerseusControl a jej najdôležitejšie funkcie

Stavový diagram funkcie „process“ a v podstate aj vnútorná logika celej triedy *PerseusControl* je vyobrazená na obrázku nižšie. Funkcia v sebe implementuje všetky ostatné funkcie triedy a rovnako využíva aj všetky dostupné funkcie triedy *RasPi*. Prvým krokom funkcie „process“ je pokus o vytvorenie pripojenia prostredníctvom socketu, kde tento stav pretrváva pokiaľ toto spojenie nie nadviazané. Po úspešnom pripojení je emitovaný signál, ktorý ďalej spracúva hlavná trieda *MainWidget*. Po týchto stavoch sa funkcia dostáva do slučky, ktorá je vykonávaná cyklicky. Táto časť pozostáva zo spracovania vstupov od tlačidiel diaľkového ovládača prostredníctvom funkcií triedy *RasPi*, následnej reinterpretácie týchto dát, aby s nimi bolo možné ovládať pohon a servomotory polohujúce kameru a vytvorenie paketu v súlade s protokolom, ktorý je použitý pre komunikáciu. Tento paket obsahuje dáta pre riadenie robota, ale aj požiadavky na odpoveď o stavových veličinách servomotorov. Ďalej cyklická časť funkcie pozostáva s odoslania paketu, následného príjmu paketu, ktorý je jeho odpoveďou a spracovanie dát v ňom obsiahnutých. Nakoniec sú spracované všetky spomenuté dáta na grafickú interpretáciu a prostredníctvom signálu sú predané hlavnej triede *MainWidget*, ktorá ich následne spracuje a predá triede *Hud* pre vykreslenie grafiky. Slučka je ukončená z iniciatívy hlavnej triedy a to po opakovaných a neúspešných pokusoch o komunikáciu so smerovačom. Po jej ukončení je socket uzavretý a sú uvoľnené použité prostriedky.

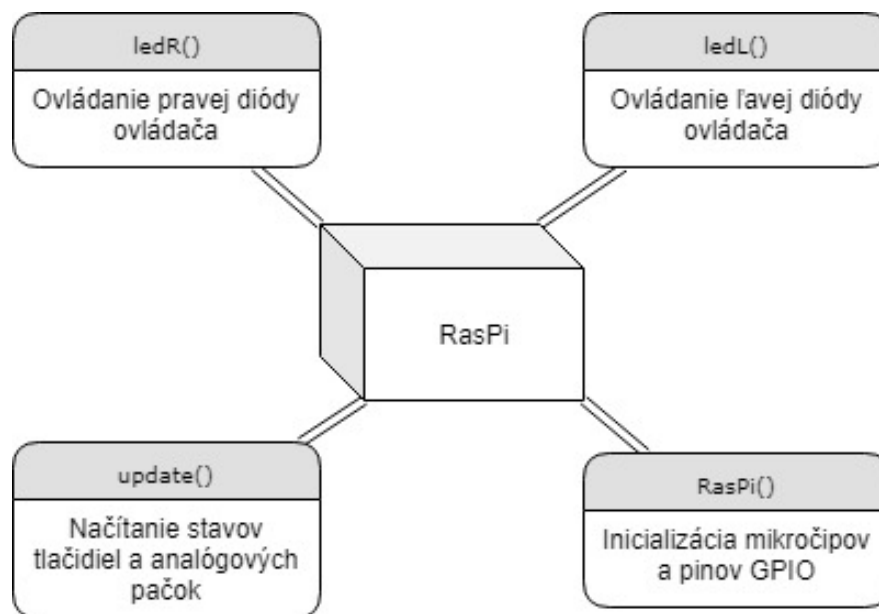


Obr. 47: Vnútorňa logika triedy PerseusControl

7.4.4 Trieda RasPi

Poslednou vytvorenou triedou je trieda *RasPi*, ktorá už ako bolo spomenuté slúži pre spracovanie vstupov od tlačidiel a analógových páčok diaľkového ovládača. Daná trieda využíva knižnicu tretej strany *WiringPi* poskytujúcej rozhranie pre jednoduchšiu prácu s pinmi GPIO. Konštruktor slúži pre nakonfigurovanie

použitých mikročipov (2x A/D prevodník a 2x I/O expandér), teda priradenie pinov k adresám použitej zbernice I2C. Ďalej sú v konštruktore nastavené módy daných pinov, či ide o vstup, výstup, PWM alebo hodinový signál. Nakoniec sú pre vstupné piny nastavené integrované pull up rezistory. Na obrázku nižšie sú vyobrazené základné funkcie triedy *RasPi* a ich popis.



Obr. 48: Trieda *RasPi* a jej najdôležitejšie funkcie

7.5 Výsledná aplikácia

Softvérová realizácia je vykonaná s úmyslom dodržať a naprogramovať všetku navrhnutú funkcionality aplikácie popísanú v kapitole 6. Táto podkapitola popisuje všetky implementované funkcie výslednej aplikácie s ohľadom na vytvorený návrh v kapitole 6 a zdôvodňuje nemožnosť vytvorenia niektorých funkcií.

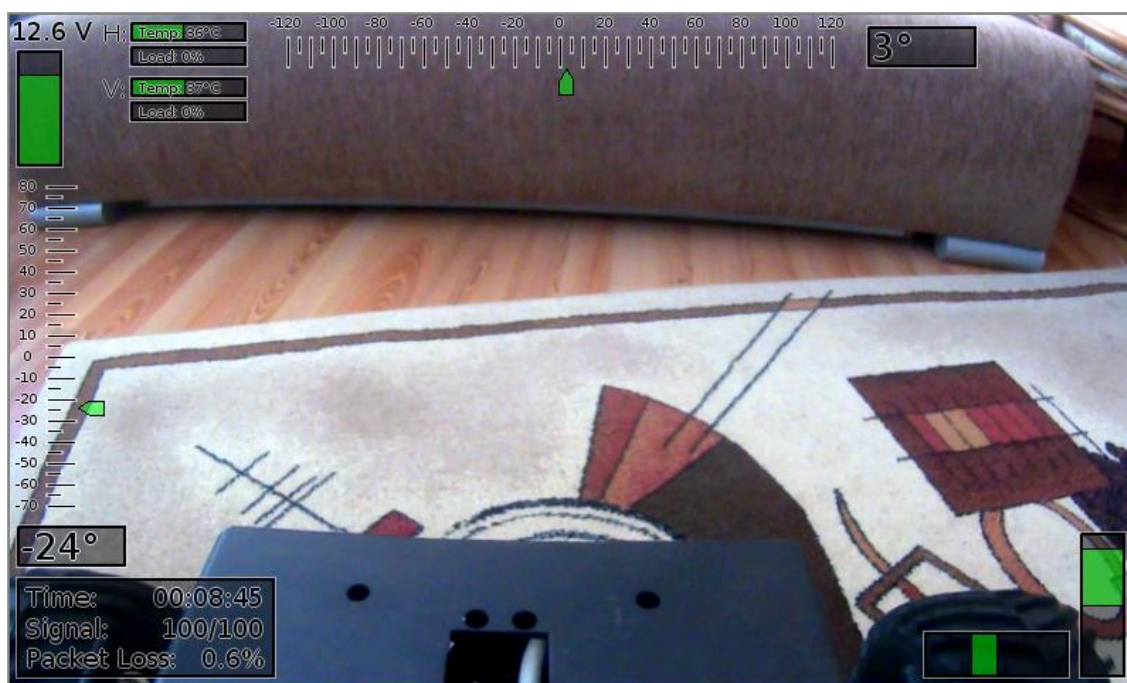
Základnou časťou aplikácie je projekcia HUD na obrazovku (Obr. 49), na ktorej je prehrávaný videesignál. Podľa vytvoreného návrhu v kapitole 6.2.1 sa skladá zo 4 častí. Funkcionalita prvej časti, ktorá implementuje spätnú väzbu od analógových páčok ovládača je totožná s návrhom s rozmerovým rozdielom a absenciou textového popisu z dôvodu lepšej prehľadnosti daných prvkov.

Druhou implementovanou časťou je ukazovateľ natočenia kamery, kde je rozdielom len rozsah vertikálneho a horizontálneho natočenia kamery a to z dôvodu iných rozsahov natočenia servomotorov. Toto je spôsobené najmä fyzickou stavbou daného pohybového prvku.

Ďalšou časťou je informačný panel v ľavom spodnom rohu obrazovky, do ktorého nebolo možné implementovať detekciu stavu batérie diaľkového ovládača z dôvodu nutnosti implementácie ďalšieho potrebného hardvéru, aby bolo možné

túto funkcionálnosť vytvoriť. Ďalším rozdielom je vyjadrenie intenzity signálu, ktoré nebolo možné vyjadriť v jednotkách decibelov s referenciou na miliwatt z dôvodu, že použitý Wi-Fi modul v Raspberry Pi neposkytuje danú informáciu v tejto jednotke, ale vyjadruje intenzitu signálu číselne v rozmedzí od 0 do maximálnej intenzity o hodnote 100.

Poslednou navrhnutou časťou HUD, sú informácie zobrazované v ľavej hornej časti obrazovky. Ukazovateľ stavu napätia batérie je totožný s návrhom. Navrhnuté ukazovatele odberaného prúdu a aktuálnej rýchlosti jednotlivých motorov nebolo možné implementovať. Dôvodom je, že aj napriek prítomnosti týchto nameraných stavových veličnín v zdrojovom kóde DPS pre riadenie robota, neboli dáta obsiahnuté v paketoch odosielaných danou jednotkou a to ani v prípade vytvorenia špecifickej požiadavky o odoslanie týchto dát a rovnako nebolo možné ani upravovať zdrojový kód obsiahnutý v DPS pre riadenie robota. Na miesto spomínaných ukazovateľov boli vytvorené ukazovatele aktuálnej teploty a záťaže servomotorov, ktoré je možno vidieť vedľa ukazovateľa stavu batérie.



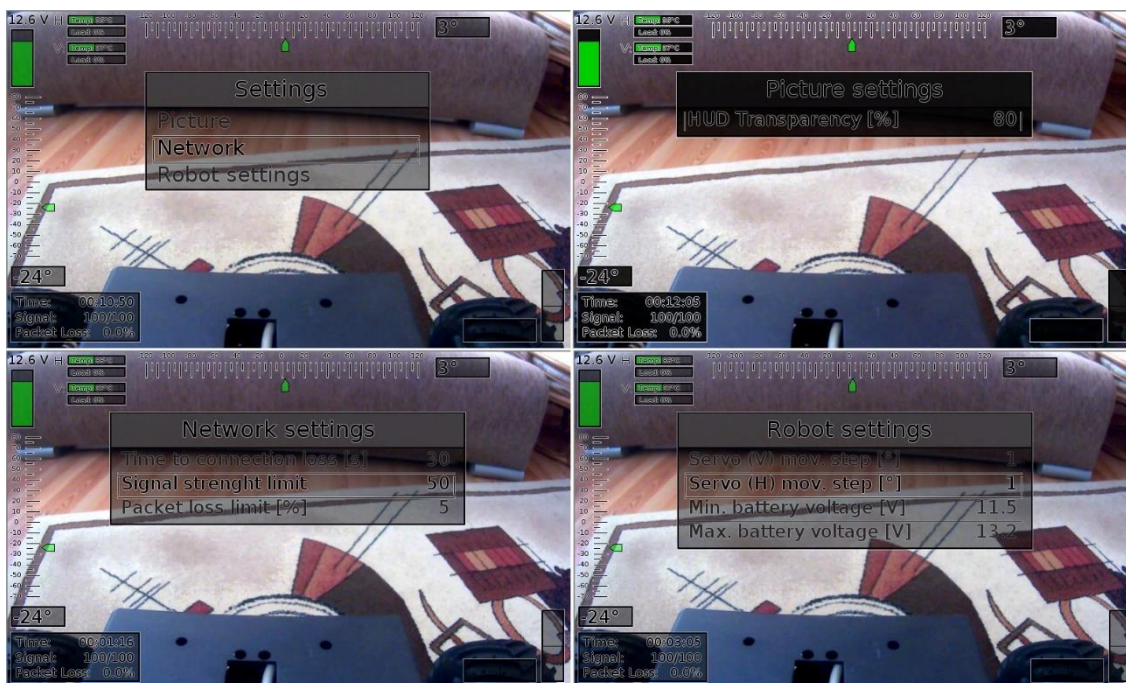
Obr. 49: Vzhľad prieťahového displeja výslednej aplikácie

Ďalšou funkcionálnosťou aplikácie je poskytnutie GUI v podobe menu (Obr. 50), ktoré poskytuje parametrizáciu aplikácie. Vykonaná implementácia návrhu z kapitoly 6.2.2 obsahuje niekoľko rozdielov. V prvej časti menu obsahujúcej nastavenie obrazu, ktorá je vyobrazená v pravej hornej časti obrázku nižšie absentuje položka nastavenia jasnosti obrazu z dôvodu, že použitý displej pochádza z nižšej modelovej rady, ktorá túto funkcionálnosť poskytuje len v 2 stavoch. Prvým

stavom je plný jas displeja a druhým je vypnutie osvetlenia displeja. Implementovanie funkcie by bolo v tomto prípade bezvýznamné.

Druhá časť menu vyobrazená v ľavej spodnej časti obrázku je totožná s návrhom, s rozdielom jednotiek pre nastavenie limitu intenzity, ktorý vyplýva z faktu spomenutého vyššie, že daný Wi-Fi modul poskytuje túto informáciu v iných jednotkách.

Poslednou implementovanou časťou menu zobrazenou v pravej spodnej časti obrázku je nastavenie robota. V tomto prípade je rozdiel oproti návrhu absencia nastavenia percentuálneho stavu batérie, kde pokles pod túto hodnotu mal vyvolať stavové hlásenie o slabej batérii ovládača a to z dôvodu nemožnosti implementovania monitorovania stavu batérie spomenutého vyššie. Ďalej sa tu nenachádza nastavenie hodnoty maximálneho odberného prúdu pre pohon robota, ktoré opäť vyplýva z nemožnosti implementácie funkcie pre monitorovanie odberných prúdov spomenutého vyššie. Posledným rozdielom je, že namiesto nastavovania rýchlosti pohybu servomotorov, bola implementovaná funkcia nastavovania veľkosti pohybového kroku v jednotkách stupňov a to z dôvodu, že toto riešenie poskytuje väčšiu variabilitu pri polohovaní kamery pri využití maximálnej rýchlosti pohybu servomotorov, čo je v konečnom dôsledku žiaduce.



Obr. 50: Náhľad na ponúkané menu aplikácie

Poslednou implementovanou časťou aplikácie sú stavové hlásenia navrhnuté v kapitole 6.2.3. Všetky implementované stavové hlásenia sú totožné s vytvoreným návrhom s malými geometrickými zmenami a preformulovanými frázami s pochopiteľne absentujúcimi hláseniami týkajúcimi sa batérie diaľkového

ovládača a pohonov robota. Rovnako chýba implementácia reakcií robota na rôzne stavy, ktoré nebolo možné implementovať z dôvodu nemožnosti zasiahnúť do zdrojového kódu DPS pre riadenie robota. Na obrázku nižšie je možné vidieť implementované stavové hlásenia.



Obr. 51: Implementované stavové hlásenia

8 ZÁVER

V práci sú spracované poznatky o použitej platforme GEARS-SMP a hardvérovom vybavení robota Perseus, ktorým je sieťová kamera, smerovač, servomotory a doska plošných spojov. Z dôvodu chýbajúcej dokumentácie fungovania celého systému je vytvorený jeho popis. Ďalej je popísaný protokol pre riadenie servomotorov Robotis Dynamixel RX-64 a EX-106 polohujúcich kameru, formát M-JPEG a štandard H.264 použitý pre prenos videa.

Práca sa okrajovo venuje teoretickým poznatkom ohľadom návrhu užívateľského rozhrania v robotických aplikáciách. Tie sú využité práve pri samotnom návrhu aplikácie. Rovnako je spracovaná problematika vhodnosti ovládacích zariadení, ktoré dnešný trh ponúka. Následne je popísaný použitý diaľkový ovládač a jeho nedostatky. Rešeršná časť je zakončená popisom hardvéru a softvéru v použítom diaľkovom ovládači, kde ide o platformu Raspberry Pi, operačný systém Raspbian a použité periférie.

Ďalej je popísaný vytvorený návrh robotickej aplikácie, ktorý sa týka užívateľského rozhrania vo forme virtuálneho priehľadového displeja, menu pre parametrizáciu aplikácie v podobe grafického užívateľského rozhrania, popisu stavových hlásení a popisu funkcií tlačidiel ovládača.

Na záver sa práca venuje samotnej implementácii navrhnutého riešenia do zariadenia Raspberry Pi. Popisuje použité rozhrania pre programovanie aplikácií a použitú sadu programových nástrojov Qt ako aj vývojové prostredie. Ďalej je popísaný postup krížovej kompilácie do zariadenia Raspberry Pi a samotné softvérové riešenie, ale aj princíp jeho fungovania. Poslednou časťou je popis realizovanej aplikácie a jej porovnanie vzhľadom k vytvorenému návrhu.

Výsledkom práce je návrh a realizácia robotickej aplikácie s následnou implementáciou do zariadenia Raspberry Pi obsiahnutého v diaľkovom ovládači, ktorá dokáže ovládať pohybové funkcie robota, zobrazovať videesignál, vybrané stavové veličiny a ponúka GUI pre parametrizáciu tohto systému.

V súvislosti s ďalším vývojom projektu je navrhovaná implementácia reakcií robota popísaná v kapitole venujúcej sa návrhu aplikácie, doplnenie hardvérového vybavenia diaľkového ovládača umožňujúce meranie stavu batérie obsiahnutého v tomto ovládači a úprava zdrojového kódu riadiacej jednotky robota, ktorá bude umožňovať odosielanie dát získaných z enkodérov motorov pohonu robota a odberaných prúdov týmito motormi.

ZOZNAM POUŽITEJ LITERATÚRY

- [1] The Surface Mobility Platform. *Gears Educational Systems - Robust Robotic Programs Teach Science, Technology, Engineering and Math* [online]. Pittsburg: DEPCO, 2018 [cit. 2018-12-23]. Dostupné z: http://www.gearseds.com/surface_mobility_platform.html
- [2] Worksheet for GEARS Surface Mobility Platform Parts. In: *The Surface Mobility Platform* [online]. Hanover (Massachusetts): Gears Educational Systems, 2018 [cit. 2018-12-23]. Dostupné z: http://www.gearseds.com/files/smp-bom_nocost.pdf
- [3] AXIS P1214-E Network Camera. In: *Axis Communications - Leader in network cameras and other IP networking solutions | Axis Communications* [online]. Lund: Axis Communications AB, 2018 [cit. 2018-12-23]. Dostupné z: https://www.axis.com/files/datasheet/ds_p1214e_t10058916_en_1709.pdf
- [4] MikroTik Routers and Wireless - Products: R52Hn. *MikroTik Routers and Wireless* [online]. Riga: SIA Mikrotīkls, 2018 [cit. 2018-12-23]. Dostupné z: <https://mikrotik.com/product/R52Hn%fndtn-specifications>
- [5] MikroTik Routers and Wireless - Products: RB411AH. *MikroTik Routers and Wireless* [online]. Riga: SIA Mikrotīkls, 2018 [cit. 2018-12-23]. Dostupné z: <https://mikrotik.com/product/RB411AH>
- [6] RX-64. *ROBOTIS* [online]. Soul: ROBOTIS Co., 2018 [cit. 2018-12-23]. Dostupné z: <http://emanual.robotis.com/docs/en/dxl/rx/rx-64/>
- [7] EX-106+. *ROBOTIS* [online]. Soul: ROBOTIS Co., 2018 [cit. 2018-12-23]. Dostupné z: <http://emanual.robotis.com/docs/en/dxl/ex/ex-106+/>
- [8] Protocol 1.0. *ROBOTIS* [online]. Soul: ROBOTIS Co., 2018 [cit. 2018-12-23]. Dostupné z: <http://emanual.robotis.com/docs/en/dxl/protocol1/>
- [9] QuickTime File Format. In: *Apple Developer* [online]. Cupertino: Apple Computer, 2001 [cit. 2018-12-23]. Dostupné z: <https://developer.apple.com/standards/qtff-2001.pdf>
- [10] POYNTON, Charles. *Digital video and HDTV: algorithms and interfaces*. San Francisco: Morgan Kaufmann Publishers, 2003, xlii, 692 s. : il. ISBN 1-55860-792-7.

- [11] Video Streaming API. In: *Axis Communications - Leader in network cameras and other IP networking solutions / Axis Communications* [online]. Lund: Axis Communications AB, 2013 [cit. 2018-12-23]. Dostupné z: https://www.axis.com/files/manuals/vapix_video_streaming_52937_en_1307.pdf
- [12] Joint Video Team. *ITU: Committed to connecting the world* [online]. Geneva: ITU, 2018 [cit. 2018-12-24]. Dostupné z: <https://www.itu.int/en/ITU-T/studygroups/com16/video/Pages/jvt.aspx>
- [13] RICHARDSON, Ian E. *THE H.264 ADVANCED VIDEO COMPRESSION STANDARD*. Second Edition. Chichester: John Wiley & Sons Ltd, 2010. ISBN 978-0-470-51692-8.
- [14] ISO/IEC 14496-10:2014(E). *Information technology — Coding of audio-visual objects — Part 10: Advanced Video Coding*. Eighth edition. Switzerland: ISO/IEC, 2014.
- [15] H.264. *SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS: Advanced video coding for generic audiovisual services*. Edition 12. Geneva: ITU-T, 2017.
- [16] Encoding options for H.264 video. *Adobe: Creative, marketing and document management solutions* [online]. California: Adobe, 2009 [cit. 2018-12-24]. Dostupné z: https://www.adobe.com/devnet/adobe-media-server/articles/h264_encoding.html
- [17] URBÁNEK, Pavel. *Srovnání videokodeků* [online]. Brno, 2011, 59 s. [cit. 2018-12-25]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=117717. Bakalářská práce. Vysoké učení technické v Brně. Vedoucí práce Ing. David Bařina.
- [18] LAL, Rajesh. *Digital design essentials: 100 ways to design better desktop, web, and mobile interfaces*. Beverly, MA: Rockport Publishers, 2013. ISBN 978-1-59253-803-4.
- [19] JOHNSON, Jeff. *Designing with the mind in mind simple: simple guide to understanding user interface design guidelines*. Second edition. Boston: Elsevier, Morgan Kaufmann is an imprint of Elsevier, 2014. ISBN 978-0-12-407914-4.

- [20] Head-up display. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-2018 [cit. 2018-12-25]. Dostupné z: https://en.wikipedia.org/wiki/Head-up_display
- [21] ŽALUD, Luděk, Lukáš KOPEČNÝ, František BURIAN a Tomáš FLORIÁN. CASSANDRA - heterogeneous reconnaissance robotic system for dangerous environments. In: *System Integration (SII), 2011 IEEE/SICE International Symposium on* [online]. Kyoto: IEEE Publishing, 2011, s. 1275-1280 [cit. 2018-12-24]. DOI: 10.1109/SII.2011.6147632. ISBN 978-1-4577-1523-5.
- [22] BURIAN, František, Luděk ŽALUD, Petra KOCMANOVÁ, Tomáš JÍLEK a Lukáš KOPEČNÝ. Multi-robot system for disaster area exploration. In: *Flood Recovery, Innovation and Reponse IV* [online]. 184. Poznan: WIT Press, 2014, s. 263-274 [cit. 2018-12-24]. DOI: 10.2495/FRIAR140221. ISBN 9781845647841. ISSN 1743-3541.
- [23] Cassandra | Laboratory of Telepresence and Robotics. *Laboratory of Telepresence and Robotics* [online]. Brno: LTR, 2015 [cit. 2018-12-24]. Dostupné z: <http://www.orpheus-project.cz/cassandra/>
- [24] CYBORG X PC Flight Stick: USER MANUAL. In: *Flight Simulator and Licensed Cessna Pro Flight Sim Products and the latest X-56 | Saitek.com* [online]. China: Saitek, 2008 [cit. 2018-12-24]. Dostupné z: http://www.saitek.com/manuals/Cyborg%20X_manual.pdf
- [25] Wireless PS4 Controller - Razer Raiju Ultimate. *Razer Europe / For Gamers. By Gamers.* [online]. San Francisco: Razer, 2018 [cit. 2018-12-24]. Dostupné z: <https://www.razer.com/eu-en/gaming-controllers/razer-raiju-ultimate>
- [26] Futaba 4PV 4-Channel T-FHSS, S-FHSS Radio System. *Futaba® Radio Control Systems and Accessories* [online]. Champaign: Horizon Hobby, 2018 [cit. 2018-12-24]. Dostupné z: <https://www.futabarc.com/systems/futk4700-4pv/index.html>
- [27] Futaba 18MZ WC 18-Channel Computer System With R7008SB Receiver. *Futaba® Radio Control Systems and Accessories* [online]. Champaign: Horizon Hobby, 2018 [cit. 2018-12-24]. Dostupné z: <https://www.futabarc.com/systems/futk9542-18mz-wc/index.html>
- [28] RPi Hardware. *Embedded Linux Wiki* [online]. Europe: Embedded Linux Wiki, 2016 [cit. 2018-12-25]. Dostupné z: https://elinux.org/RPi_Hardware

- [29] FAQs - Raspberry Pi Documentation. *Raspberry Pi — Teach, Learn, and Make with Raspberry Pi* [online]. UK: Raspberry Pi Foundation, 2018 [cit. 2018-12-25]. Dostupné z: <https://www.raspberrypi.org/documentation/faqs/>
- [30] Raspberry Pi 2 Model B - Raspberry Pi. *Raspberry Pi — Teach, Learn, and Make with Raspberry Pi* [online]. UK: Raspberry Pi Foundation, 2018 [cit. 2018-12-25]. Dostupné z: <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>
- [31] GPIO - Raspberry Pi Documentation. *Raspberry Pi — Teach, Learn, and Make with Raspberry Pi* [online]. UK: Raspberry Pi Foundation, 2018 [cit. 2018-12-25]. Dostupné z: <https://www.raspberrypi.org/documentation/usage/gpio/README.md>
- [32] Raspberry Pi Downloads - Software for the Raspberry Pi. *Raspberry Pi — Teach, Learn, and Make with Raspberry Pi* [online]. UK: Raspberry Pi Foundation, 2018 [cit. 2018-12-25]. Dostupné z: <https://www.raspberrypi.org/downloads/>
- [33] Raspbian - eLinux.org. *Embedded Linux Wiki* [online]. Europe: Embedded Linux Wiki, 2014 [cit. 2018-12-25]. Dostupné z: <https://elinux.org/Raspbian>
- [34] *Raspbian* [online]. EU: Raspbian Wiki, 2018 [cit. 2018-12-25]. Dostupné z: <https://www.raspbian.org/>
- [35] Raspberry Pi Touch Display - Raspberry Pi. *Raspberry Pi — Teach, Learn, and Make with Raspberry Pi* [online]. UK: Raspberry Pi Foundation, 2017 [cit. 2018-12-25]. Dostupné z: <https://www.raspberrypi.org/products/raspberry-pi-touch-display/>
- [36] Raspberry Pi Touch Display - Raspberry Pi Documentation. *Raspberry Pi — Teach, Learn, and Make with Raspberry Pi* [online]. UK: Raspberry Pi Foundation, 2015 [cit. 2018-12-25]. Dostupné z: <https://www.raspberrypi.org/documentation/hardware/display/README.md>
- [37] EDIMAX - Wireless Adapters - N150 - 150Mbps Wireless IEEE802.11b/g/n nano USB Adapter. *EDIMAX - Wireless Series,Wired Series,Router Series,Switches,USB Hubs,KVM Switches,Print Server Series,IP Cam* [online]. Taiwan: EDIMAX, 2014 [cit. 2018-12-25]. Dostupné z: https://www.edimax.com/edimax/merchandise/merchandise_detail/data/edimax/in/wireless_adapters_n150/ew-7811un/

- [38] SABÓ, Marek. *Ruční dálkový ovladač pro robot Perseus*. Brno, 2019. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/115799>. Semestrální práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce František Burian.

ZOZNAM PRÍLOH

Príloha 1 – Zdrojový kód programu je uložený na priloženom CD