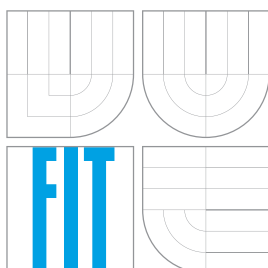


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

UŽIVATELSKÉ ROZHRANÍ AUTOPILOTA KVADROKOPTÉRY

USER INTERFACE FOR AUTOPILOT OF QUADCOPTER

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

ONDŘEJ KLEIN

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. VÍTĚZSLAV BERAN, Ph.D.

BRNO 2014

Abstrakt

Bakalářská práce se zabývá návrhem a implementací uživatelského rozhraní pro autopilota kvadrokoptéry. Ovládání probíhá pomocí tabletu. Aplikace s tímto rozhraním využívá ROS a je určena pro platformu android. Poskytuje manuální a poloautonomní mód ovládání a možnost ovládání virtuálními ovládacími prvky nebo pohybem zařízení. Součástí práce je detailní rozbor schopností použité kvadrokoptéry a popis existujících řešení, sloužících jako podklad pro návrh aplikace. Výsledný program je otestován a výsledky vyhodnoceny.

Abstract

Bachelor's thesis describes the design and implementation of user interface for autopilot of quadcopter. Controlling the quadcopter is done via tablet computer. It provides manual and semi-autonomous mode and the option to control quadcopter by virtual joysticks or device movement. Part of the thesis is detailed analysis of quadcopter's abilities and description of existing solutions, which serves as a basis for the design of the application. The final program is tested and the results are evaluated.

Klíčová slova

uživatelské rozhraní, kvadrokoptéra, ROS, AscTec Pelican, Android.

Keywords

user interface, quadcopter, ROS, AscTec Pelican, Android.

Citace

Ondřej Klein: Uživatelské rozhraní autopilota kvadrokoptéry, bakalářská práce, Brno, FIT VUT v Brně, 2014

Uživatelské rozhraní autopilota kvadrokoptéry

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Vítězslava Berana, Ph.D.

.....

Ondřej Klein
21. května 2014

Poděkování

Děkuji svému vedoucímu bakalářské práce, panu Ing. Vítězslavu Beranovi, PhD, za poskytnutí cenných rad při konzultacích a pomoc při řešení problémů.

© Ondřej Klein, 2014.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	2
2	Teorie	3
2.1	Existující řešení	3
2.2	Platforma AscTec Pelican	6
2.3	Atomboard	12
2.4	Robotický operační systém	13
2.5	Výběr platformy pro návrh a vývoj aplikace	16
3	Návrh řešení	18
3.1	Analýza zadání	18
3.2	Ovládací módy	19
3.3	Ovládací prvky	20
3.4	Aplikace TyraTeleop	24
4	Realizace, experimenty a vyhodnocení	29
4.1	Oživení Tyry	29
4.2	Vývoj aplikace TyraTeleop	30
4.3	Experimenty s aplikací	32
4.4	Výsledky experimentů	35
4.5	Vyhodnocení experimentů	37
5	Závěr	38
A	Obsah CD	41

Kapitola 1

Úvod

Kvadrokopty jsou moderní létající drony se čtyřmi rotory. Na rozdíl od klasických helikopter nevyžadují složitou konstrukci hlavního a ocasního rotoru, případně dvou hlavních protiběžných, o to větší nároky však kladou na senzorové vybavení a přesné řízení každého rotoru. V dnešní době velké miniaturizace a nízké ceně kontrolérů a senzorů je toto řešení výhodnější než v případě mechanicky náročné helikoptéry. Výzkum a vývoj kvadrokopty zažívá v posledních letech velký rozmach, ať už akademické sféře, nebo ve společnostech vyvíjejících dostupné kvadrokopty pro domácí použití.

Tato bakalářská práce se týká studia, návrhu a vývoje uživatelského rozhraní pro ovládání kvadrokopty na mobilní platformě. Program s tímto uživatelským rozhraním bude sloužit k ovládání vestavěného autopilota kvadrokopty Tyra, postavené na platformě Asc-Tec Pelican. Cílem práce je program s dobře použitelným a intuitivním uživatelským rozhraním, umožňující bezpečné ovládání kvadrokopty. Na rozdíl od ovládání RC ovladačem bude možné v aplikaci zobrazovat obraz z kamery na kvadrokopty a sledovat tak okolí z její perspektivy. Na druhou stranu nelze očekávat preciznější ovládání než s RC ovladačem, který má fyzické ovládací prvky, zejména z důvodu absence hmatové zpětné vazby od dotykového displeje mobilního zařízení. Lze také očekávat nižší dosah bezdrátového spojení mezi mobilním zařízením a kvadrokopty. Na samotné kvadrokopty je osazena počítačová deska s x86 procesorem Intel Atom, umožňující instalaci běžných desktopových operačních systémů, v tomto případě linuxové distribuce Ubuntu. Výhodou tohoto řešení je velká výpočetní síla na straně kvadrokopty umožňující vývoj a další zdokonalování její autonomie.

V kapitole 2 jsou popsány existující řešení uživatelských rozhraní pro ovládání kvadrokopty v komerčních aplikacích. Dále je detailně popsána platforma, na které je kvadrokopty postavena a popsány možnosti, které lze při návrhu a vývoji pro ni využívat. V kapitole 3 je rozpracován návrh ovládacích prvků, uživatelského rozhraní a komunikace mezi ovládací aplikací a kvadrokopty. Kapitola 4 pojednává o ověření návrhu aplikace a jejího uživatelského rozhraní při jejím vývoji, ladění a následném testování pohodlnosti a bezpečnosti ovládání.

Kapitola 2

Teorie

Tato kapitola obsahuje popis existujících létajících dronů a jejich ovládacích aplikací, rozbor platformy, na které je postavena kvadrokoptéra, pro kterou bude ovládací aplikace autopilota navržena a popis nástrojů potřebných pro návrh a realizaci vlastního řešení.

2.1 Existující řešení

V současné době je na trhu hned několik komerčních kvadrokoptér, které využívají aplikaci pro mobilní zařízení jako svůj hlavní ovládací prostředek. Staly se tedy dalším rozšířením současných širokých možností chytrých telefonů a tabletů. První, široce rozšířenou, telefonem nebo tabletem ovládanou kvadrokoptérou byl model AR.Drone od francouzské společnosti Parrot. Ten byl uvedený na trh v roce 2010 s aplikací pro operační systém iOS a později přibyla i oficiální verze pro Android. V roce 2012 byla uvedena vylepšená verze označená jako AR.Drone 2.0[10], která přinesla lepší stabilizaci letu a nové možnosti v ovládací aplikaci, jako například mód pro režiséry, umožňující jednoduše natáčet záběry ve stylu filmových sekvencí.

Podobný princip využívá i čínská společnost Walkera u svého modelu W100¹. Ten je však možné ovládat nejen pomocí mobilní aplikace, ale také pomocí modelářského RC ovladače², a zvýšit tím tak nejen dosah, ale i přesnost ovládání.

Tyto kvadrokoptéry jsou určeny zejména k zábavě. Jejich výhodou je jednoduchost použití, bezpečný provoz i nezkušeným uživatelem a cenová nenáročnost. Mobilní aplikace tak nahrazuje nákladné RC ovladače, ovšem za cenu sníženého dosahu a nižší pohodlnosti a preciznosti ovládání.

AR.Drone 2.0

AR.Drone 2.0, která je zobrazena na obrázku 2.1, je evolucí původního modelu AR.Drone. Oproti němu je lépe senzorově vybaven, má novou verzi ovládací aplikace s více možnostmi a kvalitnější kameru s vyšším rozlišením. Využívá data z několika sensorů k zajištění stabilizace kvadrokoptéry. Díky nim dokáže udržovat stálou výšku, automaticky vzlétnout a přistát, kompenzovat vliv větru a po ztrátě wifi signálu se vznášet na místě. Mezi tyto sensory patří:

¹<http://www.walkera.com/en/showgoods.php?id=1776>

²Ovladač s mnoha ovládacími prvky určený ke vzdálenému řízení modelu pomocí rádiových signálů, viz http://en.wikipedia.org/wiki/Radio_control

- ultrazvukový sonar³ pro měření letové výšky,
- kamera s rozlišením QVGA (320x240 obrazových bodů) a snímkovou frekvencí 60fps, která sleduje prostor pod kvadrokoptérou pro korekce bočních posunů (drift) pomocí odometrie,
- 3osý akcelerometr⁴,
- 3osý gyroskop⁵,
- 3osý magnetometr⁶,
- barometr⁷ s přesností ± 10 Pa.

Dalším senzorem je kamera s horizontálním rozlišením 1280x720 obrazových bodů (720p) směřující dopředu, jejíž obraz se přenáší do mobilní ovládací aplikace a slouží k řízení kvadrokoptéry mimo zorné pole uživatele/pilota, nebo záznam fotografií či videa. AR.Drone je otevřená platforma poskytující SDK⁸ pro vývoj uživatelských rozšíření.



Obrázek 2.1: Kvadrokoptéra AR.Drone 2.0[10].

Oficiální ovládací aplikace (obrázek 2.2) je určena pro chytré telefony a tablety s operačními systémy iOS⁹ a Android¹⁰. Základními ovládacími prvky jsou virtuální joysticky,

³<http://en.wikipedia.org/wiki/Sonar>

⁴Elektromechanické zařízení pro měření zrychlení, viz <https://www.dimensionengineering.com/info/accelerometers>

⁵<http://electroiq.com/blog/2010/11/introduction-to-mems-gyroscopes/>

⁶<http://www.sensorplatforms.com/understanding-smart-phone-sensor-performance-magnetometer-2/>

⁷http://en.wikipedia.org/wiki/Pressure_sensor

⁸Zkratka pro Software Development Kit, jedná se o soubor nástrojů pro vývoj programů na dané platformě[6].

⁹<http://www.apple.com/cz/ios/what-is/>

¹⁰<http://www.android.com/>

kdy jeden joystick slouží pro změnu výšky zařízení a otáčení (anglicky *yaw*), a druhý virtuální joystick slouží k pohybu vpřed a vzad (anglicky *pitch*) a do stran (anglicky *roll*). Pro zjednodušení a větší srozumitelnost budu dále uvádět pouze anglické výrazy označující pohyb kvadrokoptéry různými směry[10].



Obrázek 2.2: Ovládací aplikace AR.Drone 2.0[10].

Walkera W100

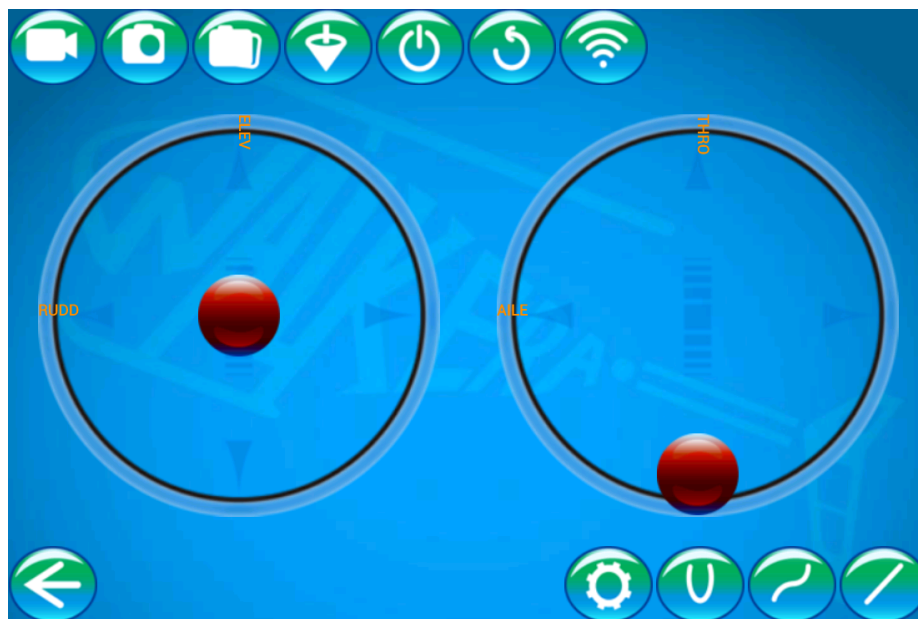
Jedná se o malou a lehkou kvadrokoptéru, která kromě ovládání pomocí tradičního RC ovladače nabízí také ovládání pomocí aplikace pro mobilní zařízení. Oficiální ovládací aplikace WK-Remote¹¹ existuje ve verzích pro mobilní operační systémy Android a iOS. Tato aplikace staví na podobném konceptu, jako aplikace pro ovládání kvadrokoptér AR.Drone. Ovládání je možné buď pomocí virtuálních joysticků, nebo pomocí kombinace virtuálního joysticku pro ovládání výšky a otáčení, a naklápění zařízení pro změnu směru pohybu kvadrokoptéry[16].

PowerUp 3.0

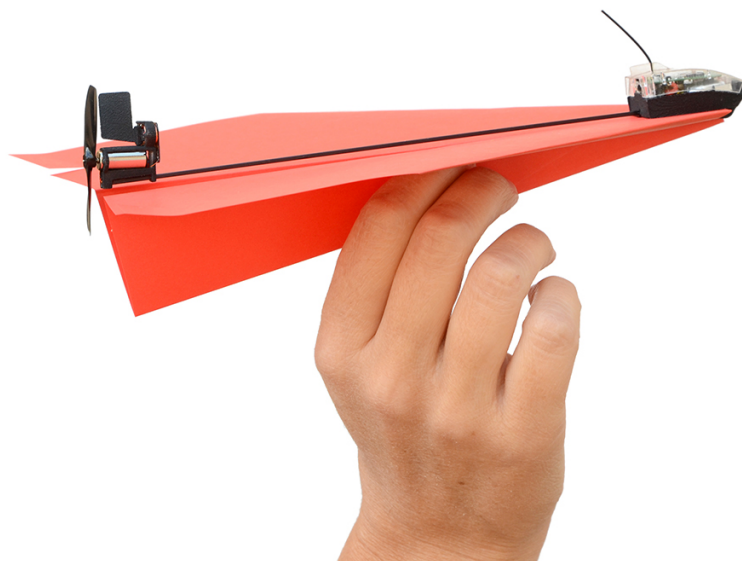
PowerUp 3.0 sice není kvadrokoptéra, ale je to velmi zajímavý návrh nízkonákladového létajícího zařízení pro zábavu. Jedná se o malý modul s Bluetooth¹², baterií a motorem s kormidlem a vrtulkou, který se nasadí na letadélko poskládané z papíru. Ovládá se pomocí aplikace v mobilním zařízení, v současné době je dostupná verze pro iOS a verze pro Android se připravuje. Ovládací aplikace je velmi jednoduchá, jak ukazuje obrázek 2.5. Pro ovládání rychlosti vrtule slouží posuvník a zatáčení se provádí naklápěním mobilního zařízení do stran. Kromě ovládání poskytuje aplikace přehled o stavu baterie, síle signálu a rychlosti vrtule. V horní části obrazovky aplikace je pak letecký kompas kombinovaný s umělým horizontem[11].

¹¹<https://itunes.apple.com/us/app/wk-remote/id628281032?mt=8>

¹²Bezdrátová technologie pro komunikaci dvou zařízení na krátké vzdálenosti[6]



Obrázek 2.3: Ovládací aplikace pro kvadrokoptéry Walkera W100, verze pro Android[9].



Obrázek 2.4: PowerUp 3.0[11].

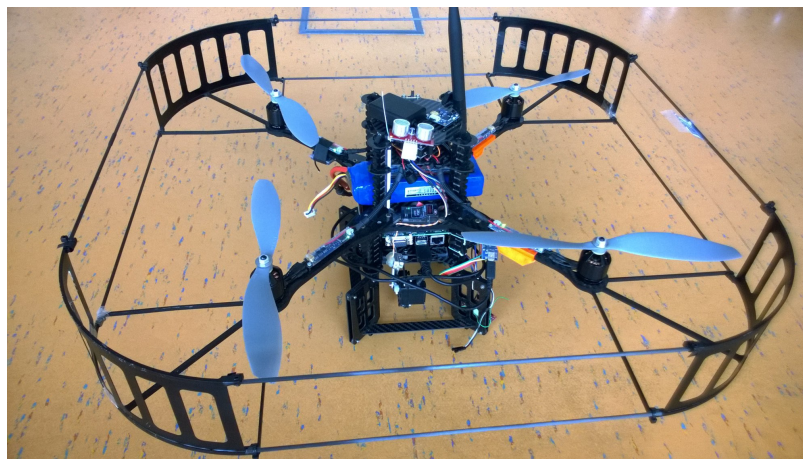
2.2 Platforma AscTec Pelican

Tyra, což je název kvadrokoptéry, pro kterou bude ovládací aplikace určena, je postavena na platformě AscTec Pelican od německé společnosti Ascending Technologies (zobrazena na obrázku 2.6). Pelican umožňuje nést až 650g zátěže a volitelně může být vybaven modulem autopilot board a jedním ze dvou x86 PC z nabídky AscTec. Rám Pelicana se skládá z dílců vyrobených ze sendviče uhlíkových vláken a dřeva balza a laserem vyřezávaných dílů



Obrázek 2.5: Ovládací aplikace pro PowerUp 3.0[1].

z hliníku. To zajišťuje nízkou hmotnost a velkou mechanickou odolnost. Snaha o co nejnižší hmotnost je znát například také z použití umělohmotných šroubů namísto kovových. Pohon zajišťují čtyři motory napájené střídavým proudem. Pelican je modulární konstrukce, která umožňuje měnit rozložení jednotlivých prvků na kvadrokoptěře. Základem jsou čtyři vertikální nosníky, v jejichž středu je přichycen díl se čtyřmi horizontálními rameny pro motory. Na těchto nosnících jsou v pravidelných odstupech zarážky, které umožňují montáž horizontálních pater pro osazení komponentami. Tato patra je možné umístit dle potřeby a požadavků stavitelů kvadrokoptéry[3].



Obrázek 2.6: Kvadrokoptéra Tyra postavená na platformě Asctec Pelican.

Na Tyře je rozložení shora dolů následující:

- horní patro s umístěným GPS¹³ modulem, magnetometrem a sonarem směřujícím

¹³GPS - Global Positioning System, viz <http://www.gps.gov/>

vzhůru,

- patro obsahující autopilot board a plošný spoj řídící napájení,
- díl s rameny nesoucími motory, který ve svém středu zároveň nese akumulátor,
- patro nesoucí Atomboard,
- patro s automaticky vyvažovaným nosičem kamery.

Na spodních částech hlavních nosníků je ještě přichycen DC/DC měnič s výstupním napětím 12V pro napájení Atomboardu (viz 2.3) z akumulátoru během letu a sonar mířící směrem dolů.

Autopilot board

Autopilot board (dále autopilot) je hlavní řídící obvod kvadrokoptéry. Obstarává fúzi dat ze stabilizačních senzorů, zpracování příkazů z RC ovladače a přesné řízení motorů. Autopilot obsahuje dva ARM7¹⁴ procesory a téměř všechny sensory, kromě GPS a magnetometru, které jsou umístěny ve vlastních modulech, mimo tištěný spoj autopilotu. Komunikaci mezi součástmi autopilotu popisuje obrázek 2.8. Přidanou hodnotou autopilota je základní úroveň autonomie kvadrokoptéry, kdy dokáže v závislosti na letovém módu (viz 2.2) udržovat svoji výšku, případně výšku a pozici. Tyto možnosti mohou být dále rozšířeny, například pomocí sonarů, nebo vizuální odometrie¹⁵, která vyžaduje instalaci kamery sledující prostor pod kvadrokoptérou.

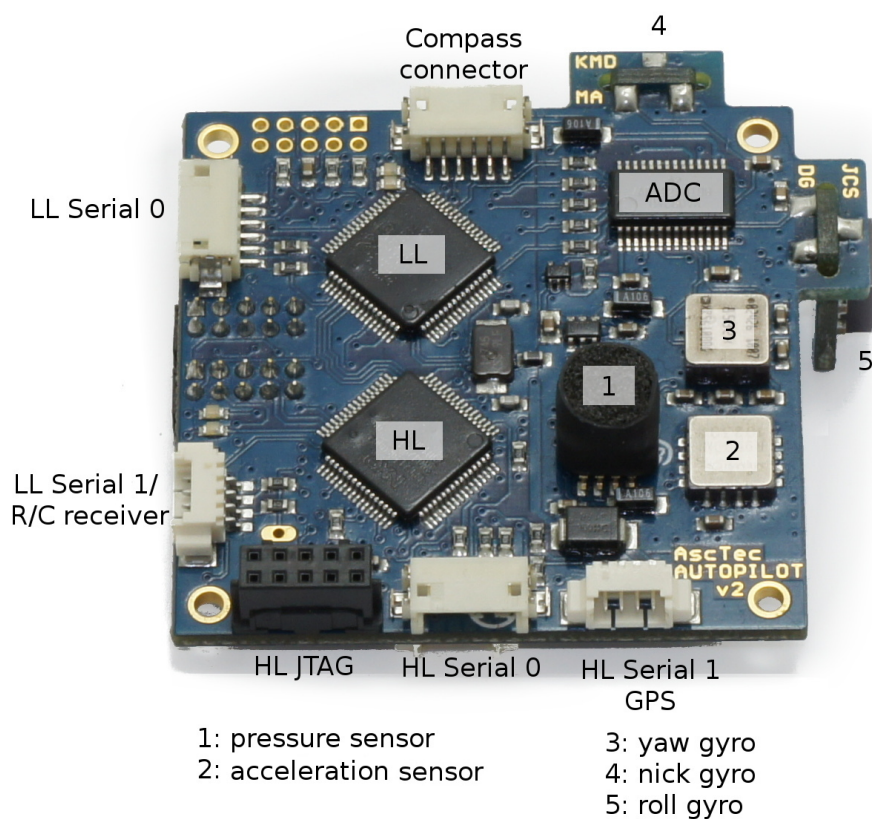
Autopilot umožňuje několik způsobů vzdáleného ovládání kvadrokoptéry:

- **Řízení pomocí RC ovladače:** V tomto režimu reaguje kvadrokoptéra na povely RC ovladače, které jsou interpretovány LL procesorem (popsaný v 2.2).
- **Řízení přes sériové rozhraní LL procesoru:** V tomto režimu jsou z počítače posílány pomocí definovaného protokolu příkazy do LLP kvadrokoptéry.
- **Řízení pomocí HL procesoru:** V tomto režimu řídí kvadrokoptéru uživatelský program, který je prováděn v HL procesoru (popsaný v 2.2).
- **Řízení pomocí počítače přes HL procesor:** V tomto případě zpracovává veškeré řídící příkazy počítač, který je buď nainstalovaný přímo na kvadrokoptéře, nebo externě. Zpracované příkazy jsou posílány do HL procesoru, který je dále předá LL procesoru. Tento způsob bude využit v aplikaci navržené v kapitole 3.

Režimy řízení přes HL procesor fungují pouze pokud je na RC ovladači aktivován přepínač sériové linky mezi HL a LL procesorem na kanálu 5. V případě její aktivace a řízení přes HL procesor jsou ignorovány příkazy posílané přímo do LL procesoru. Deaktivace linky naopak umožňuje zásah do ovládání a bezpečné přistání v případě špatné funkce nebo chybě v řízení přes HL procesor.

¹⁴Rodina 32bitových procesorů využívajících architekturu ARM, viz <http://www.arm.com/products/processors/classic/arm7/>

¹⁵Odometrie je proces, který popisuje transformaci dat poskytnutých enkodéry na změnu pozice a orientace robota, viz <http://robotika.cz/guide/odometry/en>



Obrázek 2.7: Autopilot board s popsanými součástmi[4].

Senzory

Autopilot board obsahuje následující senzory:

- 3osý akcelerometr,
- 3 gyroskopy pro měření rotace na všech třech osách,
- barometr,
- magnetometr (kompas) - připojen externě,
- GPS modul - připojen externě.

Možnosti stabilizace letu byly dále rozšířeny ultrazvukovým sonarem směřujícím k zemi, který měří letovou výšku. Měření výšky tímto způsobem je podstatně přesnější, než při použití dat pouze ze senzoru atmosférického tlaku, ovšem pouze do omezené výšky (barometr tuto limitaci nemá). Tyto senzory však vyžadují další vývoj a v mé bakalářské práci nebudou využívány.

Low Level procesor

Low Level procesor (dále LLP) provádí fúzi dat ze všech senzorů, horizontální stabilizaci s frekvencí výpočtů 1000Hz. Zpracovává také řízení pozice kvadrokoptéry s využitím GPS

a magnetometru. S LLP lze komunikovat přes jeho sériový port, například pomocí bezdrátového modulu využívající technologii XBee. Pro toto rozhraní je definován protokol, který umožňuje přímé zasílání kontrolních příkazů do LLP. Tento interface je také využíván programem Asctec Autopilot Control (dále Autopilot Control)[4].

LLP dále řídí speciální záchranný mód, který se aktivuje, pokud se kvadrokoptéra dostane mimo dosah RC ovladače, což může nastat ve vzdálenosti větší, než 1km. Činnost, která se provede při aktivaci záchranného módu, je možné nastavit.

Záchranný mód

Schopnost kvadrokoptéry převzít kontrolu nad řízením v případě ztráty signálu, nebo vybití RC ovladače snižuje riziko poškození létajících robotů. Způsobů, jakým kvadrokoptéra v krizové situaci bezpečně přistane, je několik. Jejich konfigurace je možná přes aplikaci Autopilot Control¹⁶. Možnosti jsou následující:

- **Přímé přistání:** Kvadrokoptéra ihned zahájí klesání rychlostí 1 m/s a přistane. Pokud byl v případě krizové události aktivní GPS mód, dokáže se při klesání držet své aktuální pozice.
- **Opožděné přistání:** Tato procedura se aktivuje pouze při letu v módu GPS, pokud dojde ke krizové události a zároveň je ztracen signál GPS. Kvadrokoptéra počká na aktuální pozici 5s a pak přistane stejně, jako při přímém přistání.
- **Návrat domů:** V tomto režimu dojde k přepnutí letového módu na GPS, počká 5s, poté vystoupá do výšky 20m nebo udržuje svoji aktuální výšku, pokud je vyšší, než 20m, a vrací se přímou dráhou k místu vzletu, kde přistane rychlostí klesání 1m/s.
- **Návrat domů ve velké výšce:** Režim podobný předchozímu, kvadrokoptéra však před návratem k místu vzletu vystoupá do nejvyšší výšky, které bylo dosaženo od posledního zapnutí motorů a v této výšce se vydá přímou dráhou k místu vzletu. Výšku pro návrat do domovské pozice lze také nastavit.

Jako domovská pozice je automaticky zvolena pozice kvadrokoptéry v okamžiku zapnutí motorů. Pokud není v tu chvíli k dispozici signál GPS, je konfigurace záchranného módu změněna na přímé přistání[4].

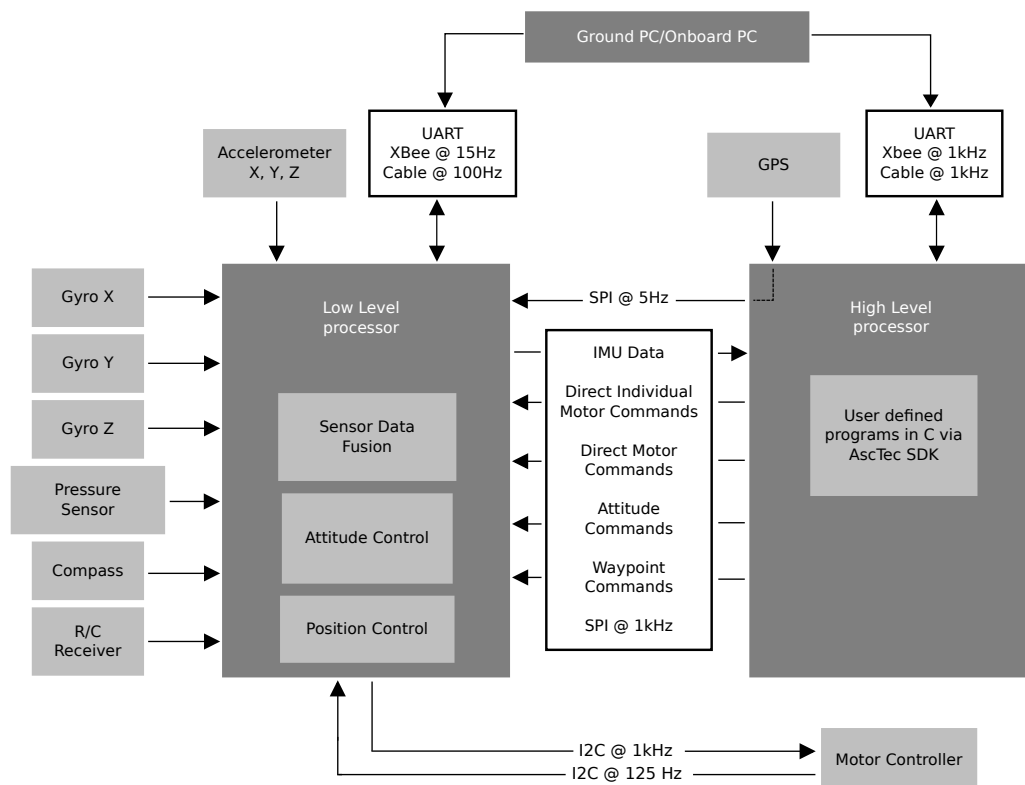
High Level procesor

High Level procesor (dále HLP) je uživatelsky programovatelný procesor. Umožňuje vývoj řídicích algoritmů pro autopilota. HLP má přístup k datům, které zpracovává LL procesor. K vývoji se používá AscTec SDK¹⁷, umožňuje použít jeden hardwarový breakpoint a debugging za běhu. Programy napsané v jazyce C se do HL procesoru nahrávají pomocí USB-JTAG¹⁸ převodníku. Při použití s Robotickým operačním systémem (viz 2.4) a Asctec Mav frameworku (viz 2.4) je do HLP nahrán Asctec Hl Firmware, který zprostředkovává komunikaci mezi autopilotem a Atomboardem.

¹⁶Aplikace pro konfiguraci a správu AscTec Autopilot Board[4]

¹⁷<http://wiki.asctec.de/display/AR/AscTec+SDK>

¹⁸JTAG - Joint Test Action Group, standard IEEE 1149.1 pro programování a debugging procesorů a paměťových čipů, viz <http://noel.feld.cvut.cz/vyu/ap2/JTAGmoje.htm>



Obrázek 2.8: Schéma komunikace mezi procesory na Autopilot boardu[4].

Letové módy

Autopilot board poskytuje tři letové módy, které se volí pomocí přepínače na RC ovladači. Tyto módy jsou:

- manuální mód,
- mód udržování aktuální výšky - poloautonomní,
- GPS mód - poloautonomní.

Manuální mód

V tomto módu reaguje kvadrokoptéra přímo na pokyny z RC ovladače, stabilizace letu probíhá pouze ve formě udržení vodorovné pozice kvadrokoptéry a regulace motorů. I v tomto módu však dojde k aktivaci záchranného módu pro přistání při ztrátě RC signálu. Let s kvadrokoptérou v tomto módu vyžaduje zkušeného pilota, zejména pro lety v nižších výškách a pro přistání. Při pokusných letech se ukázala nutnost velmi citlivého zacházení s páčkou plynu, což byl pro nezkušené piloty značný problém.

Mód udržování výšky

Mód udržování výšky (dále uveden jako Height mód) využívá dat ze sensorů k vypočítání letové výšky a její udržování. Na příkazy z RC ovladače reaguje jinak než v manuálním módu.

Výchozí pozice levé ovládací páčky je ve středu. Pohyb ze středu nahoru způsobí stoupání, pohyb ze středu dolů klesání. Při umístění páčky do středové polohy udržuje kvadrokoptéra svoji aktuální letovou výšku. Příkazy pro yaw a změny směru jsou interpretovány stejně, jako v manuálním módu.

GPS mód

GPS mód obsahuje stabilizaci letové výšky stejně jako předchozí mód, využívá však navíc ještě GPS modul. Díky GPS je možné stabilizovat kvadrokoptéru tak, aby udržovala nejen svoji výšku, ale i pozici. V softwaru Autopilot Control lze zadávat GPS body, na které má kvadrokoptéra přiletět (tzv. waypoints) a tuto funkcionalitu lze využít i při ovládání kvadrokoptéry pomocí ROSu¹⁹. Příkazy z RC ovladače jsou interpretovány podobně, jako v Height módu.

Srovnání stabilizace letu Pelican a AR.Drone 2.0

Schopnosti stabilizace letu jsou u Pelicana na vysoké úrovni, avšak při porovnání s kvadrokoptérou AR.Drone 2.0 jsou nižší, zejména v udržování konstantní výšky a kompenzace bočních posunů. V základním sestavení využívá Pelican k vypočítání letové výšky data z výše zmíněných senzorů. Tato data jsou zpracována v LL procesoru, kde je zjištěna výška a případně i GPS pozice v GPS módu. AR.Drone 2.0 má podobné senzorové vybavení, jako Pelican (kromě GPS přijímače), navíc však přidává ultrazvukové sonary, které dokáží velmi přesně měřit vzdálenost kvadrokoptéry od země. Dále obsahuje kameru s širokým záběrem mířící dolů k zemi, která se využívá pro měření pohybu kvadrokoptéry pomocí vizuální odometrie. Díky tomu je zaručena velmi přesná stabilizace, kdy se AR.Drone 2.0 dokáže vznášet a udržovat svoji pozici i při nepříznivých povětrnostních podmínkách. Při pokusných letech s Tyrou se ukázalo, že létání v Height módu nebo GPS módu je výrazně jednodušší, než v manuálním módu, avšak mohu odhadovat, že výšková stabilizace bude dobře fungovat až ve větších letových výškách, do kterých jsme si s kolegy ovšem netroufli, jelikož nikdo z nás není zkušený pilot létajících dronů. V malých letových hladinách kolísala naměřená výška až o 2m, jak jsme zjistili po analýze bag souborů z letu (popsány dále v 2.4)[7].

2.3 Atomboard

Aby bylo možné ovládat kvadrokoptéru pomocí mobilního zařízení, je třeba mezičlánek, který s ním bude schopen komunikovat a zároveň bude posílané řídicí příkazy interpretovat do formy srozumitelné pro řídicí obvody kvadrokoptéry. Tímto mezičlánkem bude počítačová základní deska s operačním systémem pro PC, která bude díky přítomnosti procesoru Intel Atom dále označovaná jako Atomboard.

Použitý Atomboard je od společnosti Advantech, a jedná se o model MIO-2261. Tato základní deska je formátu Pico-ITX²⁰, což je vhodné pro létající platformu a zároveň se jedná o maximální rozměry pro montáž na jedno z pater Pelicana. Deska je osazena těmito komponentami:

¹⁹ROS - Robot Operating System

²⁰Formát základních desek s rozměry 100x72mm od společnosti VIA, viz <http://www.via.com.tw/en/initiatives/spearhead/pico-itx/>

- dvoujádrový procesor Intel Atom N2800 s frekvencí 1.86GHz²¹,
- 4GB 1333MHz DDR3 operační paměť Crucial ve slotu SODIMM,
- ADATA 30GB SSD²² ve slotu mSATA,
- Wifi modul Intel 7260AC²³ s podporou standardu 802.11ac²⁴ a Bluetooth 4.0

Atomboard poskytuje na poměry létajících dronů vysoký výkon a možnost snadného vývoje. Napájení je realizováno DC/DC převodníkem z akumulátoru kvadrokoptéry. Atomboard je připojen k autopilotu pomocí sériového rozhraní a komunikuje přímo s HL procesorem. Použitým operačním systémem je Ubuntu 12.04 LTS. Pro komunikaci s Atomboardem je nutné nahrát do autopilota modifikovaný firmware, který je součástí balíku AscTec Mav Framework (popsaný dále) a jmenuje se HL Firmware. Nainstalovaný ROS je ve verzi Hydro a doplňuje jej ještě AscTec Mav Framework, sloužící jako rozhraní ke komunikaci s HLP a zároveň poskytuje všechna data zpracovávaná LLP.

Připojení k autopilotu

Možnosti připojení jsou tři. První možností je použití bezdrátového sériového spojení pomocí XBee²⁵. V tomto případě je možné spojení autopilota s jakýmkoliv počítačem i mimo Atomboard na Tyře, což je výhodné pro testování a experimentování s novými technologiemi. Druhou možností je připojení přes převodník z USB na sériové rozhraní, tedy spojení z USB portu Atomboardu do sériového portu na autopilotu. Díky tomu, že se jedná o kabelové spojení, je přenos spolehlivější než přes XBee a umožňuje také dosažení vyššího baudrate²⁶. Poslední možností je přímé připojení ze sériového rozhraní na Atomboardu do sériového rozhraní na autopilotu, které má velmi podobné vlastnosti jako spojení pomocí převodníku.

2.4 Robotický operační systém

Robotický operační systém (dále ROS) je softwarový mezičlánek, který zajišťuje možnost vysokoúrovňové komunikace mezi mobilním zařízením a Atomboardem. Příkazy jsou poté prostřednictvím ROS balíků určených přímo ke komunikaci s platformou AscTec Pelican interpretovány a posílány do Autopilot Boardu.

ROS je open source meta-operační systém pro vývoj softwaru pracujícího s roboty a jejich částmi. Licence umožňuje použití zdarma i pro komerční projekty. Původně byl vyvinutý na Stanfordově univerzitě v roce 2007 pod názvem Switchyard. V roce 2008 převzal vývoj ROSu Willow Garage, což je technologický inkubátor zabývající se robotickým výzkumem. V současné době řídí vývoj frameworku Open Source Robotics Foundation. ROS

²¹[http://ark.intel.com/products/58917/Intel-Atom-Processor-N2800-\(1M-Cache-1_86-GHz\)](http://ark.intel.com/products/58917/Intel-Atom-Processor-N2800-(1M-Cache-1_86-GHz))

²²SSD - Solid State Drive, pevný disk založený na pamětech typu Flash, viz <http://www.svethardware.cz/vse-co-jste-chteli-vedet-o-ssd/26524>

²³<http://www.intel.com/content/www/us/en/wireless-products/dual-band-wireless-ac-7260-bluetooth.html>

²⁴Standard bezdrátového síťového spojení využívající frekvenci 5GHz s rychlostí až 433Mbit/s při použití jedné antény, viz. <http://www.zive.cz/clanky/novy-standard-wi-fi-gigabit-vzduchem/sc-3-a-165687/default.aspx>

²⁵Nízkopříkonové bezdrátové spojení, viz <http://www.digi.com/xbec/>

²⁶Rychlost přenosu přes sériové rozhraní, viz <http://electronicdesign.com/communications/what-s-difference-between-bit-rate-and-baud-rate>

poskytuje abstrakci nad robotickým hardwarem, ovladače pro zařízení, simulátor, nástroje pro vizualizaci a správu balíků a je pro něj dostupné obrovské množství open source knihoven. Hlavním cílem ROSu je podpora znovupoužitelnosti zdrojového kódu při robotickém výzkumu. Aktuální verze se jmenuje Hydro a byla vydána v září 2013. Předchozí, stále používaná a podporovaná verze se jmenuje Groovy a byla vydána v prosinci roku 2012. Oficiálně odporované operační systémy pro ROS jsou Ubuntu a Mac OS X, komunita však poskytuje podporu také pro Fedoru, Gentoo, Arch Linux a další linuxové distribuce.

Běžící ROS je peer-to-peer²⁷ síť mezi sebou komunikujícími procesy (uzly). ROS nabízí několik různých způsobů komunikace:

- synchronní komunikace přes služby (anglicky services),
- asynchronní streaming přes témata (anglicky topics),
- ukládání dat na parametrovém serveru.

Architektura ROSu

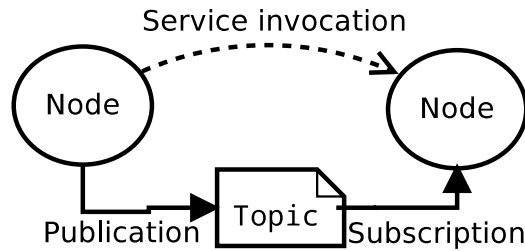
Základním stavebním prvkem peer-to-peer sítě ROSu jsou uzly (anglicky node). Uzly jsou procesy, které provádějí výpočty. Kontrolní systém robota se zpravidla skládá z několika uzlů pro řízení různých věcí. Některé uzly zpracovávají data ze senzorů, jiné zpracovávají řízení motorů nebo serv, další mohou sloužit jako vizualizéry pro znázornění pohybu robota. ROS uzly jsou napsány s použitím klientských knihoven, kterými jsou například `roscpp` v jazyce C++, `rospy` v jazyce Python a experimentální knihovna `rosjava` napsaná v jazyce Java.

ROS Master slouží ke zprostředkování komunikace mezi dvěma uzly. Uzly se u Mastera zaregistrují svým jménem a ten pak poskytuje informace o ostatních zaregistrovaných uzlech a tématech. Master také notifikuje uzly, pokud se změní registrační údaje uzlů, ke kterým jsou připojeny a umožňuje tak dynamicky vytvářet a rušit spojení.

Uzly mezi sebou komunikují pomocí zpráv (anglicky message). Zprávy jsou jednoduché datové struktury obsahující základní datové typy (integer, float, boolean), stejně jako pole těchto základních datových typů. Je také možné skládat zprávy z již existujících zpráv a takto je kombinovat. Zprávy jsou posílány přes tzv. témata (anglicky topic), která zajišťují jednosměrnou komunikaci. Každý uzel může mít témata dvojího typu, odběratelské (anglicky subscribe) a publikační (anglicky publish), sloužící k příjmu, nebo publikaci zpráv. Uzly takto mohou být mezi sebou libovolně propojeny, případně vypínány a znovu spouštěny, aniž by to mělo na samotný kód uzlů nějaký vliv. Poslání zprávy pak proběhne publikováním zprávy v rámci tématu. Pokud chce uzel nějaké zprávy přijímat, přihlásí se k odběru tématu. O posílání a doručování zpráv se stará ROS, uzly jsou na sobě vzájemně nezávislé. Jediným omezením je nutnost dodržování formátu zpráv, které jsou posílány přes dané téma. Témata jsou definována svými jmény, která tvoří podobnou hierarchii jako jména cest v Unixových souborových systémech.

Pro některé účely je vyžadována komunikace typu otázka-odpověď, a zde jednosměrná komunikace, kterou poskytují témata, nestačí. Obousměrná komunikace je zprostředkována přes služby (anglicky services). Služby jsou definovány dvojicí zpráv, jednou pro dotaz a druhou pro odpověď. Služby jsou definovány svým jménem. Princip práce se službami je odeslání dotazu a čekání na odpověď.

²⁷Typ sítí, které jsou založeny na vzájemné komunikaci jednotlivých klientů, viz <http://en.wikipedia.org/wiki/Peer-to-peer>



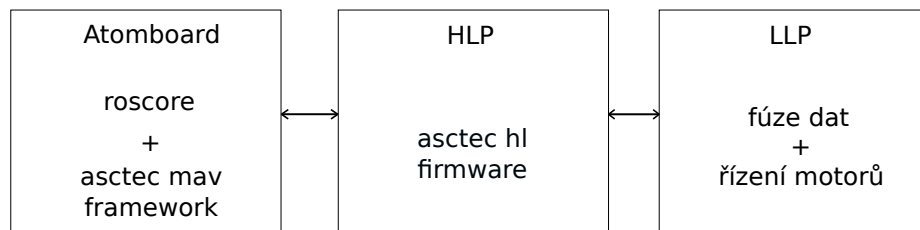
Obrázek 2.9: Komunikace mezi uzly[14].

Pro testování a simulaci jsou v ROS připraveny tzv. bagy, které umožňují uložení a přehrávání ROS zpráv. Jejich typické použití je nahrání dat ze senzorů, například při letu kvadrokoptéry, a poté jejich znovupoužití při testování v laboratoři[14].

AscTec Mav Framework

AscTec Mav Framework (dále Mav Framework) je soubor balíčků pro získávání dat z autopilotu a řízení kvadrokoptér od firmy AscTec. Byl vyvinut na Technické Univerzitě v Curychu v Laboratoři autonomních systémů (anglicky Autonomous Systems Lab) ²⁸. Pro jeho funkčnost slouží modifikovaný firmware pro HLP, nazvaný Hl firmware. Aby bylo možné ovládat kvadrokoptéru pomocí Mav Frameworku, je nezbytné mít aktivovaný přepínač sériového rozhraní na RC ovladači. Mav Framework se skládá z následujících částí[13]:

- **AscTec Hl Comm:** Tento balík obsahuje definici ROS zpráv a služeb, které jsou používány pro komunikaci s autopilotem. Mezi tyto zprávy patří `mav_ctrl` pro ovládání pohybu kvadrokoptéry, `mav_ekf` pro posílání výsledků fúze dat ze senzorů, která může probíhat i na Atomboardu v rámci balíku `ethzasl_sensor_fusion`. Dalšími zprávami jsou například `mav_imu` sloužící k posílání údajů o pozici a pohybu kvadrokoptéry, `mav_status` pro posílání stavových informací a další.
- **AscTec Hl Firmware:** Firmware pro HLP umožňující komunikaci s autopilotem. Tento firmware je nutné nahrát do HLP, bez něj není komunikace možná.
- **AscTec Hl Gps:** Balík pro posílání GPS pozic, kterými má kvadrokoptéra proletět.
- **AscTec Hl Interface:** Uzel, který poskytuje rozhraní pro komunikaci s autopilotem.



Obrázek 2.10: Schéma komunikace Atomboardu s autopilotem.

²⁸<http://www.asl.ethz.ch/>

Ovládací režimy

Asctec Mav Framework nabízí tři režimy ovládání kvadrokoptéry, mezi kterými lze přepínat pomocí modifikace parametru `position_control`. Tyto režimy jsou:

- **Režim přímých příkazů:** Parametr `position_control` je nastaven na hodnotu `POSCTRL_OFF`. V tomto režimu jsou příkazy pro pitch, roll, yaw a rychlost motorů posílány přímo do LLP. Zprávy musí přicházet na frekvenci alespoň 10Hz, jinak jsou ignorovány.
- **Režim řízení pomocí GPS:** Parametr `position_control` je nastaven na hodnotu `POSCTRL_GPS`. V tomto režimu je řízení prováděno pomocí zasílání vyžadovaných rychlostí v jednotlivých směrech a pro stoupání a klesání. Pro funkčnost tohoto režimu je nutné mít GPS signál.
- **Režim fúze senzorových dat na HLP:** Tento mód je podle autorů frameworku ze všech nejzajímavější a byl hlavní motivací pro vznik Mav Frameworku. Využívá fúze senzorických dat na HLP a ke standardním sensorům na autopilotu Pelicana přidává podporu využití kamery a zpřesnění řízení polohy pomocí odometrie. Parametr `position_control` je třeba nastavit na hodnotu `POSCTRL_HIGHLEVEL`[13].

Omezení Mav Frameworku

I přes svoji propracovanost má Mav Framework řadu nevýhod. Je nutné mít zapnutý RC ovladač a přepínač aktivující sériové rozhraní mezi HLP a LLP. V případě ztráty RC signálu přejde Tyra do záchranného režimu. Součástí není rozhraní pro ovládání orientace držáku kamery, který je na Tyře přítomen. Také letové módy jsou v Mav Frameworku řešeny jinak, než při ovládání RC ovladačem. Přepínač módů nemá na interpretaci příkazů posílaných do kvadrokoptéry přes Mav Framework žádný vliv, jediná možnost přepínání režimů je změna parametru `position_control`, který ale neumožňuje použití height módu. Height mód by byl pro ovládání mobilním zařízením výhodný, jelikož nejobtížnější část létání s kvadrokoptérou je přesné udržení výšky. Takto je možné stabilizovaně létat pouze ve venkovním prostředí a podmínkou je signál GPS. Další nevýhodou je chování Tyry v případě zpoždění nebo ztráty příjmu řídicích příkazů přes Wifi v manuálním i GPS módu, například z důvodu rušení nebo slabému signálu ve velkých vzdálenostech. V takovém případě by mohlo dojít k havárii kvadrokoptéry, je tedy nutná přítomnost pilota s RC ovladačem, který by bezpečně přistál. To by mohlo být provedeno dvěma způsoby - buď deaktivací sériového rozhraní a manuálním přistáním, nebo vypnutím RC ovladače, po kterém by kvadrokoptéra přešla do záchranného módu a automaticky přistála[13].

2.5 Výběr platformy pro návrh a vývoj aplikace

Pro vývoj programu byla zvolena platforma Android, a to z několika důvodů. Android je nejrozšířenější mobilní operační systém na světě[5], nabízí možnosti na špičkové úrovni a vývoj pro něj je možný na všech běžných desktopových platformách zcela zdarma. Možnost seznámit se s vývojem pro tento operační systém v rámci takového projektu, jako je tento, je velmi lákavá. ROS také poskytuje knihovnu `rosjava`[12], která dokáže běžet na Androidu. Pro vývoj byly zváženy tři vývojová prostředí. První možností bylo použití programovacího jazyka Python a NUI frameworku Kivy²⁹, který nabízí snadný vývoj v ja-

²⁹<http://kivy.org/#home>

zyce Python a definici GUI prvků pomocí jazyka Kivy Language. Dalšími možnostmi bylo využití vývoje v některém z JavaScriptových frameworků pro vývoj mobilních aplikací – například PhoneGap³⁰ nebo Steroids³¹, nebo nativním vývojem na Androidu v Javě. První jmenovaný způsob by přinesl možnost snadného portování na jakoukoli moderní mobilní i desktopovou platformu, nevýhodou by však bylo riziko nižšího výkonu aplikace. Nativní vývoj v Javě přináší možnost dosažení velkého výkonu aplikace a velmi snadné ladění, nevýhodou je však omezení pouze na platformu Android. Výkon bude klíčový, jelikož bude aplikace obsahovat virtuální ovládací prvky, jejichž animace je relativně náročná, a zobrazování videa. Po pečlivém zvážení všech výhod a nevýhod jednotlivých řešení byl vybrán vývoj v Javě pro Android.

³⁰<http://phonegap.com/>

³¹<http://www.appgyver.com/>

Kapitola 3

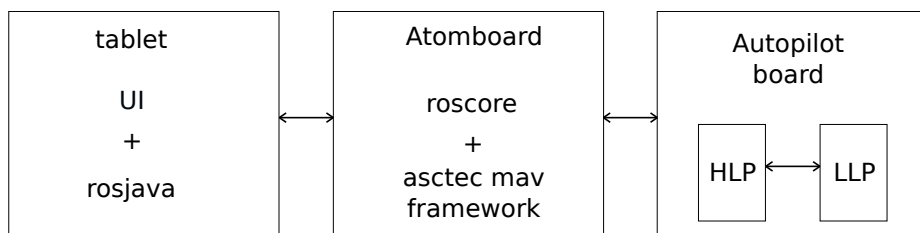
Návrh řešení

Tato kapitola se zabývá analýzou zadání a navržením způsobu jeho řešení. Návrh řešení se týká navržení uživatelského rozhraní tak, aby bylo funkční a dobře použitelné. K tomu budou použity prostředky popsané v předchozí kapitole.

3.1 Analýza zadání

Cílem práce je navrhnout vhodné uživatelské rozhraní pro ovládání kvadrokoptéry pomocí mobilního zařízení. Návrh se bude soustředit především na ovládací prvky, protože ty nejvíce ovlivňují výslednou kvalitu ovládání. Navržené uživatelské rozhraní by mělo zohledňovat přednosti i omezení platformy, na které je Tyra postavena. Tyra bude používat ROS jako prostředníka pro komunikaci mezi tabletem a jejími řídicími obvody. Z důvodu dobré podpory komunikace s ROsem byla pro implementaci aplikace zvolena platforma Android.

Aby bylo možné touto aplikací ovládat kvadrokoptéru, je nezbytné mít knihovnu, kterou lze využít v aplikaci pro Android a která podporuje komunikaci s kvadrokoptérou. Pro tyto účely slouží knihovna `rojava`, podporující běh na systému Android. Na počítači, který je na kvadrokoptěře nainstalován ROS. ROS bude pomocí balíku Asctec Mav Framework interpretovat příkazy zasílané do kvadrokoptéry a posílat korespondující příkazy do autopilotu. V autopilotu převezme příkazy HL firmware v HLP a pošle je do LLP. Komunikaci znázorňuje obrázek 3.1.



Obrázek 3.1: Schéma komunikace mezi tabletem a Tyrou

Způsoby vzdáleného ovládání robotů

Existuje několik přístupů k řízení robotů, které jsou aplikovatelné v závislosti na typu robota a stupni jeho autonomie.

- **Autonomní řízení:** Tento způsob vyžaduje senzorické a řídicí schopnosti robota na velmi vysoké úrovni. Robot je operátorem pouze aktivován a zbytek úkolu, například hledání cesty k cíli a její mapování, už provádí sám. V současné době se tento typ robotů uplatňuje v pokročilých armádních zbraňových systémech, jako jsou například průzkumné a bojové drony, nebo autonomní vozidla. Podobným výzkumem se zabývá ve svém projektu Driverless car¹ společnost Google, který si klade za cíl zvýšit bezpečnost na silnicích a komfort cestujících odstraněním lidského faktoru, který může chybovat. Tímto způsobem řízení se v tomto projektu nebudu zabývat.
- **Poloautonomní řízení:** Tento režim využívá jistou míru inteligence a senzorického vybavení robota, který nevyžaduje nepřerušované řídicí příkazy od operátora. Příkladem poloautonomního řízení může být například ovládání pomocí zasilání GPS pozic, na které kvadrokoptéra doletí. Taková aplikace může mít formu mapy, na kterou uživatel klikne, potvrdí dané místo jako následující GPS waypoint a kvadrokoptéra už sama na dané místo doletí. V potaz však kvadrokoptéra nebere překážky, které se na přímé dráze k cíli mohou vyskytnout. Po doletu na dané místo čeká na další příkazy a udržuje svoji současnou letovou výšku. Pro další akce vyžaduje vždy pokyn od operátora. Dalším příkladem může být také GPS mód, který je podporován Tyrou. V tomto módu je kvadrokoptéra stabilizována na své aktuální pozici a letové výšce a čeká na další příkazy.
- **Teleoperace:** V případě teleoperací je nezbytné, aby měl operátor robota ve svém zorném poli a měl tak vizuální zpětnou vazbu, nebo alespoň přenos obrazu do ovládacího zařízení či aplikace, v případě létajících robotů pak také další nezbytné údaje, jako letovou výšku, stav akumulátoru a sílu signálu. Jedná se o režim, ve kterém robot přímo reaguje na pokyny operátora a při kterém je vyžadována operátorova neustálá pozornost.

V tomto projektu budou využívány dva režimy ovládání, a to poloautonomní a mód teleoperace, neboli manuální mód. Poloautonomní mód bude využívat stabilizace pomocí GPS, kdežto manuální mód přímých příkazů, podobně jako v manuálním módu RC ovladače.

3.2 Ovládací módy

Ovládací aplikace bude mít dva různé módy sloužící k ovládání kvadrokoptéry. První, manuální, nebude využívat poziční ani výškovou stabilizaci kvadrokoptéry a bude s ním možné létat i v budovách. Druhý, GPS mód, bude využívat poziční i výškovou stabilizaci. Ke své funkci bude využívat řízení pomocí GPS a bude ho tedy možné využít pouze venku.

Manuální mód

V tomto módu nemá Tyra žádnou výškovou ani poziční stabilizaci a ovládací příkazy jsou přímo posílány do přes HLP do LLP, který zajišťuje řízení. Příkazy obsahují hodnoty natočení na jednotlivých osách, udávané v radiánech². Tento typ řízení klade velký důraz na pohodlnost ovládání výkonu motorů. V případě horizontálního pohybu kvadrokoptéry (pitch a roll) lze použít běžný virtuální joystick (popsaný v 3.3), který se po vzdálení prstu od

¹<http://technophilicmag.com/2013/10/06/drive-or-be-driven/>

²Radián je jednotka pro měření úhlů, viz <http://www.matematika.cz/radian>

obrazovky vrátí do nulového středového stavu. Kvadrokoptéra se tak přestane hýbat v horizontálním směru a žádné nebezpečí jí nehrozí. To samé platí také o otáčení kvadrokoptéry okolo její horizontální osy (yaw). V případě ovládání výkonu motorů (thrust) pro vertikální pohyb je však situace rozdílná. Původní návrh počítal s použitím virtuálního joysticku i pro ovládání vertikálního pohybu, to se však ukázalo jako nevhodné hned ze dvou důvodů. Prvním důvodem bylo příliš hrubé odstupňování ovládání dané omezenou velikostí virtuálního joysticku a také nevyužití záporného pohybu od středu joysticku směrem dolů, protože záporné hodnoty nemají v tomto módu využití. Druhým důvodem bylo chování v případě vzdálení prstu od obrazovky, kdy se joystick vrátil do své výchozí pozice a rychlost motorů se vrátila do volnoběhu. Takové chování by mohlo být příčinou havárie.

Pro tento mód se hodí jiný ovládací prvek, který se více podobá levé ovládací páčce na RC ovladači. Tato slouží k ovládání plynu a otáčení kolem vertikální osy kvadrokoptéry. Funguje tak, že výchozí pozice je ve spodní středové části základny páčky. Pohybem nahoru se zvyšují otáčky motorů, přičemž páčka není vybavena pružinou, která by ji vracela do výchozí pozice. V kterékoliv pozici lze provést pootočení kvadrokoptéry vychýlením páčky doleva nebo doprava, a tento pohyb je poté vrácen do středové pozice pružinou. Výhodou fyzických páček je jejich hmatová zpětná vazba, není tedy nutné sledovat jejich pozici a spouštět tak oči z ovládané kvadrokoptéry. Výsledkem je návrh posuvníku (anglicky slider), který se lépe hodí pro ovládání vertikálního pohybu. Pro ovládání otáčení kvadrokoptéry budou využita tlačítka, jejichž stisknutí způsobí otáčení kolem vertikální osy kvadrokoptéry konstantní rychlostí.

GPS mód

V tomto módu má Tyra jak výškovou, tak poziční stabilizaci. Ovládací příkazy jsou v tomto režimu rozdílné od manuálního režimu a obsahují rychlosti udávané v metrech za sekundu (m/s) v jednotlivých směrech. Při posílání příkazů s nulovými hodnotami udržuje Tyra svoji aktuální pozici a výšku, nehrozí tedy havárie. Toto chování umožňuje použít běžný virtuální joystick. Stoupání pak probíhá posíláním kladných rychlostí na vertikální ose kvadrokoptéry, klesání naopak posíláním negativních rychlostí.

Ovládání pomocí naklápění tabletu

V obou zmíněných módech je možné využít ovládání pomocí detekce pohybu zařízení. Tento režim bude nahrazovat virtuální joystick pro pohyb kvadrokoptéry v horizontálních směrech. Aby nedošlo k nechtěnému ovládání kvadrokoptéry při manipulaci s tabletem, bude tato funkce závislá na aktivaci pomocí aktivčního tlačítka. Po stisknutí aktivčního tlačítka dojde ke čtení hodnot z pohybových senzorů na mobilním zařízení a jejich využití pro řízení kvadrokoptéry. Pozice při stisknutí tlačítka bude zaznamenána jako výchozí, a směrem od ní bude, podle směru naklopení zařízení, počítána intenzita příkazů, které budou posílány - tedy buď velikost úhlu natočení kvadrokoptéry v jednotlivých osách, nebo rychlosti. Při puštění tlačítka dojde k resetu výchozí pozice a vynulování posílaných hodnot.

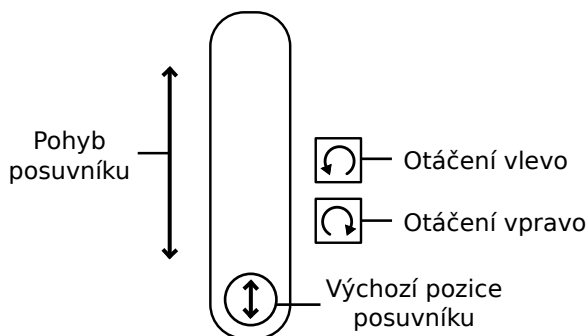
3.3 Ovládací prvky

Ovládací prvky jsou klíčovým předmětem návrhu pro dobrou použitelnost aplikace. Použití ovládacích prvků se bude odvíjet od zvoleného režimu ovládání a ovládaných atributů pohybu kvadrokoptéry. V manuálním režimu je potřebné, aby byla zvolená úroveň rychlosti

motorů zachována i při ukončení kontaktu prstu s obrazovkou, jinak by došlo ke snížení rychlostí motorů na minimum a rychlému pádu kvadrokoptéry. V GPS módu jsou naopak posílány pouze velikosti rychlostí v jednotlivých směrech, a pokud jsou nulové, tak kvadrokoptéra udržuje svoji aktuální pozici. V tomto případě je výhodnější ovládací prvek, který se po ukončení dotyku vrátí do své výchozí pozice a poskytuje nulové hodnoty. Pro první případ navrhuji použití posuvníku, a pro druhý virtuálního joysticku. K ovládání pohybu kvadrokoptéry v horizontálních směrech bude sloužit v obou režimech buď virtuální joystick, nebo tlačítko aktivující ovládání pohybem zařízení.

Posuvník pro ovládání vertikálního pohybu

Tento prvek bude použit v manuálním režimu pro ovládání výkonu motorů. Jeho funkce bude podobná, jako funkce levé ovládací páčky na RC ovladači. Ve výchozí pozici bude posuvník při spodní hraně ovládacího prvku. Pohyb posuvníku bude možný pouze ve vertikálním směru a nebude tak sloužit pro současné ovládání otáčení kvadrokoptéry kolem vertikální osy. Po opuštění prstu zůstane posuvník na svojí poslední pozici. Jelikož je manuální mód náročný na pozornost pilota, a to zejména v malých výškách, nebude nutné, aby se prst dotkl ovládacího prvku v místě přesné pozice posuvníku. Bude tedy možné, aby dotyk a pohyb po obrazovce proběhl kdekoli v rámci ovládacího prvku, a posuvník se bude posouvat ze své předchozí pozice požadovaným směrem. Díky tomu je možné precizně ovládat stoupání či klesání kvadrokoptéry bez sledování ovládacího prvku i přes absenci zpětné vazby od dotykového displeje. Je nutné si však zapamatovat pozici tohoto ovládacího prvku, která je u levého spodního okraje hlavní obrazovky aplikace.



Obrázek 3.2: Detail posuvníku a tlačítek pro otáčení.

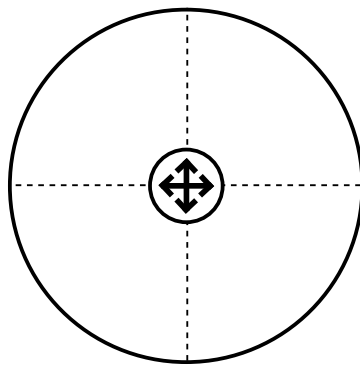
Virtuální joystick

Virtuální joystick (dále VJ) je ovládací prvek, který umožňuje současné ovládání pohybu ve dvou různých směrech. Pohyblivým prvkem, který se pohybuje na základě pohybů prstu po kruhové ploše VJ, je střed VJ (dále střed). Na běžném RC ovladači pro ovládání kvadrokoptér je hlavním ovládacím prvkem pro ovládání pohybu v horizontálních směrech právě fyzický joystick. Joystick je v klidovém neaktivovaném stavu ve středové poloze, jak ukazuje obrázek 3.3. Při dotyku kdekoli na ploše joysticku se střed přesune na místo tohoto dotyku. Po ukončení dotyku, nebo pohybů prstu po obrazovce se střed přesune zpět do centrální pozice. VJ poskytuje dvojici čísel, každé pro jednu osu. Interval hodnot, které VJ vrací, je $[0,1]$, kde 0 je hodnota v centrální pozici a 1 je maximální hodnota při okraji kruhové plochy VJ, pokud se střed nachází na vertikální, nebo horizontální ose. Výsledné hodnoty

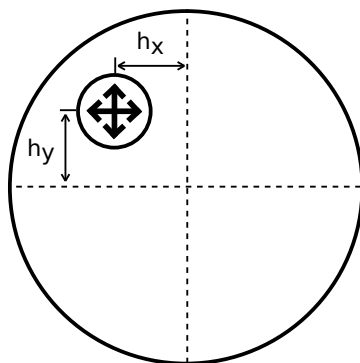
na jednotlivých osách jsou rozdíly souřadnic centrální pozice VJ a středu VJ na dané ose. Uvažujme výpočet hodnoty pro vertikální osu, kterou označme y . Hodnota pro osu y , h_y se vypočítá jako rozdíl příslušné souřadnice pozice středu ss_y a souřadnice centrální pozice VJ sc_y :

$$h_y = ss_y - sc_y$$

. Analogicky se provede i výpočet pro osu x , která představuje horizontální osu. Obrázek 3.4 zobrazuje hodnoty na osách x a y při vychýlení středu z centrální pozice.



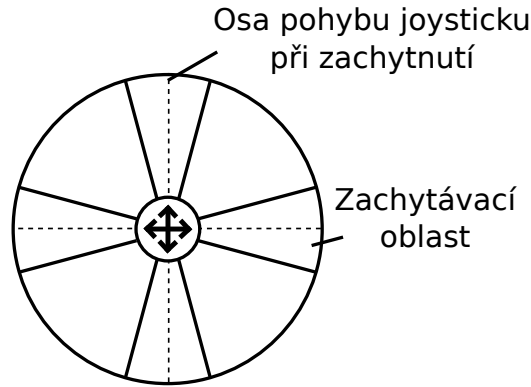
Obrázek 3.3: Detail virtuálního joysticku s naznačenými osami.



Obrázek 3.4: Hodnoty na osách při vychýlení středu.

Zachytávací zóny

Při použití VJ, který má na dotykovém displeji absenci zpětné vazby pro operátorovy ruce, je obtížné udržet pohyb středu joysticku pouze v jedné ose. Aby bylo ovládání přesto pohodlné a dostatečně přesné, má VJ zachytávací zóny, ve kterých jeho střed setrvává na dané ose, i přes mírný pohyb prstu směrem k druhé ose, která je kolmá na požadovaný směr pohybu prstu po joysticku[15]. Zachytávací zóny jsou zobrazeny na obrázku 3.5. Velikost těchto zón je podnětem k experimentům. Příliš malé plochy zón nebudou efektivní a bude docházet k nechtěným pohybům na kolmé ose, naopak při velmi velkých zachytávacích plochách je omezen rozsah hodnot, které dokáže joystick vracet. Pokud bude operátor provádět pohyb po vertikální ose, a zároveň bude chtít ovlivnit pohyb na ose horizontální, tak s vyšší vzdáleností od středu joysticku bude narůstat počáteční hodnota vrácená joystickem na kolmé ose. Rozsah hodnot tak nebude zcela plynulý, ale skokový.



Obrázek 3.5: Zachytávací plochy joysticku[15].

Exponenciální mapování

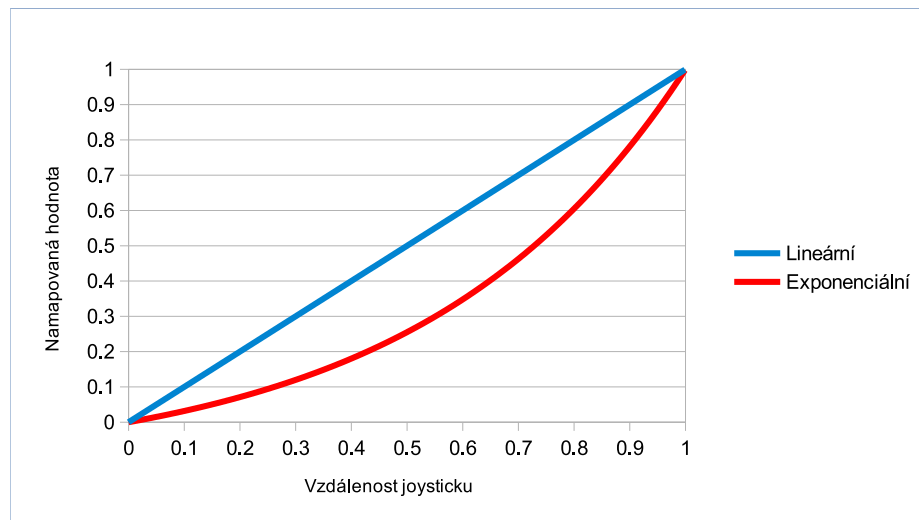
Pro použití virtuálních joysticků je exponenciální mapování výstupu výhodné, protože nabízí možnost velmi jemného ovládání v malých rychlostech a zároveň stejný rozsah vracených hodnot, jako při lineárním mapování. Pro exponenciální mapování je použita exponenciální funkce g :

$$g(d) = e^{((c*d-3)/1.4)} - e^{(-3/1-4)}$$

kde d je vstupní hodnota a c je parametr citlivosti, v mém případě je hodnota tohoto parametru 3. Pomocí této exponenciální funkce sestojíme mapovací funkci f použitou pro úpravu hodnot z virtuálního joysticku:

$$f(d) = g(d)/g(1)$$

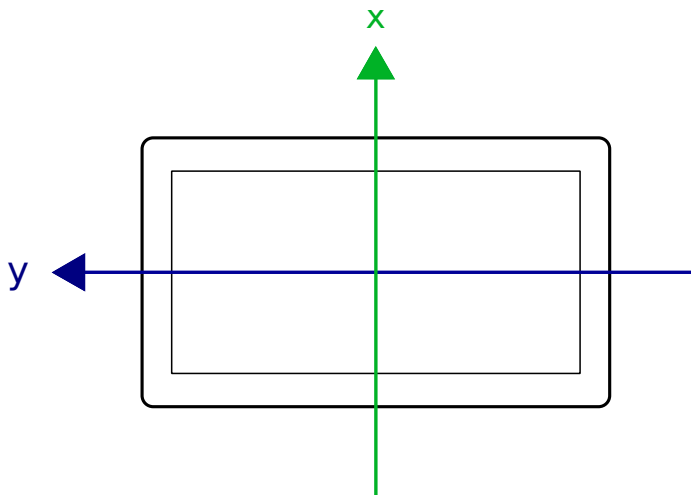
která vrací hodnoty v intervalu $[0, 1]$ [15]. Průběh hodnot je zobrazen na obrázku 3.6



Obrázek 3.6: Graf znázorňující průběh hodnot při lineárním a exponenciálním mapování.

Řízení pohybem

Mód řízení pohybem se aktivuje po stisknutí tlačítka nahrazující pravý virtuální joystick. Využívá data z gyroskopů v mobilním zařízení. Kvadrokoptéra poté reaguje na naklápění mobilního zařízení. Naklopení vpřed způsobí pohyb vpřed, analogicky naklonění vzad, a naklonění do stran. Při nestisknutém aktivačním tlačítku aplikace nereaguje na jakýkoli pohyb zařízení. Jedná se o uživateli dobře známý způsob ovládání z mnoha populárních her na mobilní zařízení, nemusí však každému vyhovovat. Aktivace tohoto režimu se provádí v nastavení po kliknutí na nabídku `Input type` a podnabídky `Use motion`.



Obrázek 3.7: Osy tabletu používané při ovládání pohybem[8].

Obrázek 3.7 zobrazuje osy tabletu, kolem kterých bude detekována rotace a výsledné hodnoty použity pro řízení kvadrokoptéry. Řízení vpřed a vzad (pitch) bude prováděno pomocí detekce rotace kolem osy y , pohyb vlevo a vpravo (roll) bude řízen pomocí rotace okolo osy x .

3.4 Aplikace TyraTeleop

Název aplikace bude TyraTeleop. TyraTeleop bude umožňovat volbu mezi dvěma různými módy ovládání, přehled důležitých telemetrických dat, přenos obrazu z kamery na Tyře a možnost ovládání pohybem tabletu. Tyto módy budou svou funkcí podobné módům voleným přepínačem na RC ovladači, z důvodu limitací Mav frameworku však není možné využít Height mód, který udržuje výšku a nevyžaduje pro svoji funkci signál GPS. Výchozím ovládacím módem bude manuální mód, který se bude svým chováním přibližovat RC ovladači v manuálním módu pomocí ovladače. Druhým módem bude GPS mód, který bude vyžadovat platný GPS signál, provoz v tomto módu bude možný jen ve venkovním prostředí. GPS mód bude využívat výškové a poziční stabilizace pomocí autopilotu, bude tedy více stabilní a bezpečný. V obou módech bude dostupný přenos obrazu z kamery na Tyře, který se zobrazí jako pozadí na ovládací obrazovce aplikace. Kvůli bezpečnostním omezením Mav frameworku budou oba módy vyžadovat zapnutý RC ovladač s aktivovaným přepínačem sériového rozhraní, ale další ovládání RC ovladačem nebude nutné.

Hlavní obrazovka aplikace

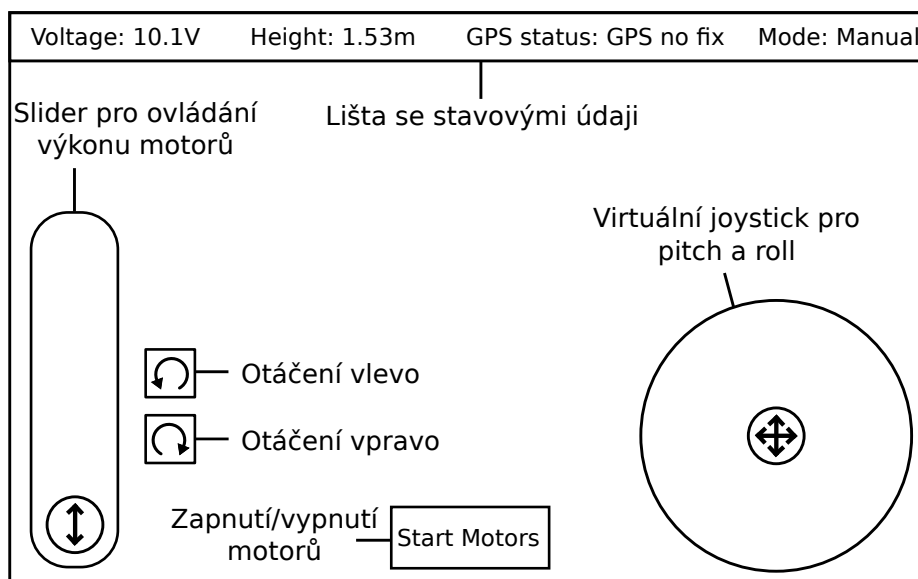
Ve výchozím manuálním módu je v levém dolním rohu posuvník pro ovládání motorů a otáčení kvadrokoptéry, v pravé dolním rohu je joystick pro ovládání pohybu horizontálními směry. Při horním okraji displeje se nachází sekce pro zobrazování stavových informací z Tyry. Jsou zobrazovány tyto informace:

- napětí baterie,
- letová výška,
- stav příjmu GPS signálu,
- použitý letový mód.

Jedná se o nejdůležitější informace o stavu kvadrokoptéry, které vylepšují schopnosti aplikace nad úroveň samostatného použití RC ovladače. V pravém horním rohu je pak tlačítko pro vstup do menu aplikace, kde je možné:

- přepnout letový mód,
- aktivovat ovládání pohybem
- ukončit aplikaci.

U dolního okraje obrazovky, uprostřed mezi ovládacími prvky, je tlačítko pro zapnutí a vypnutí motorů. Návrh rozložení ovládacích prvků je zobrazen na obrázku 3.8. V manuálním módu není vypnutí motorů povoleno, pokud není posuvník pro ovládání motorů ve výchozí poloze.



Obrázek 3.8: Hlavní obrazovka s výchozím rozložením ovládacích prvků

V případě letu v GPS módu je posuvník nahrazen stejným virtuálním joystickem, jako pro ovládání horizontálního pohybu kvadrokoptéry. Při aktivaci ovládání pomocí pohybu tabletu je poté pravý virtuální joystick nahrazen tlačítkem pro aktivaci pohybového ovládání. Režimy řízení (manuální nebo GPS) a ovládání pohybu (joystick nebo pohyb zařízení) lze libovolně kombinovat volbou v nastavení.

Přenos obrazu

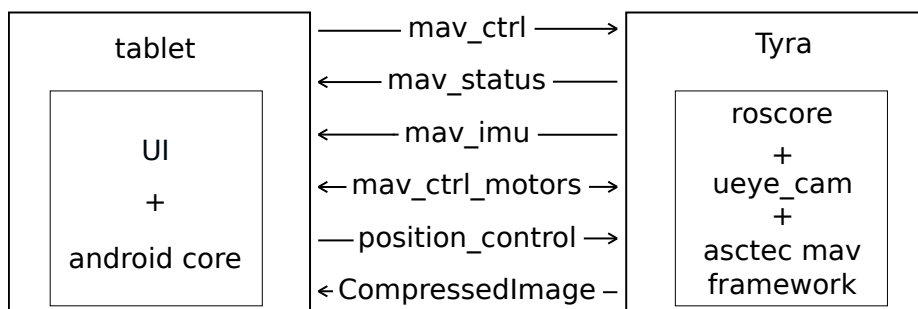
Tyra je vybavena kamerou na držáku, který se sám stabilizuje vůči pohybu kvadrokoptéry při letu, proto je výhodné tuto možnost využít v návrhu aplikace. Díky tomu bude možné přesně sledovat let kvadrokoptéry i mimo zorné pole pilota, případně provádět průzkumné lety.

Kamera namontovaná na Tyře je model uEye UI-1221LE³ od společnosti IDS. Její maximální rozlišení je 752x480 obrazových bodů při snímkové frekvenci 87,2 snímků za sekundu.

Pro využití obrazu kamery v ROSu je využit node `ueye_cam`⁴, který poskytuje jak komprimovaný, tak nekomprimovaný obraz. Obraz z kamery bude zobrazován na pozadí hlavní obrazovky, za ovládacími prvky.

Komunikace mezi tabletem a Tyrou

Aplikace využívá několik komunikačních topiců pro komunikaci s Tyrou, jak ukazuje obrázek 3.9. Zprávy pro řízení kvadrokoptéry jsou definovány v balíku `asctec_hl_comm`, který je součástí frameworku `asctec_mav_framework`. Použité zprávy jsou následující:



Obrázek 3.9: Schéma komunikace mezi tabletem a Tyrou

- **mav_ctrl**: Přes tento typ zpráv probíhá komunikace na topicu `/fcu/control` a slouží k řízení pohybu kvadrokoptéry. V tomto typu zpráv jsou definovány tyto položky:

- typ zprávy
- hodnoty pro osy x, y a z
- hodnoty pro otáčení kolem vertikální osy (yaw)
- maximální rychlost pohybu v horizontálních směrech
- maximální rychlost vertikálního pohybu

Typ zprávy určuje režim, ve kterém bude probíhat řízení. Režimy jsou tři a pro správnou funkčnost musí jejich nastavení korespondovat s nastavením režimů pomocí parametru `position_control`, jak je popsáno v 2.4. V režimu přímých příkazů je typ zprávy nastaven na `acceleration`. Limity pro maximální rychlosti nejsou brány v potaz, protože řídicí obvody kvadrokoptéry neznají rychlost pohybu. Hodnoty pro jednotlivé osy jsou v radiánech a pro yaw v rad/s.

³<http://en.ids-imaging.com/store/produkte/kameras/usb-2-0-kameras/ueye-le/ui-1221le.html>

⁴http://wiki.ros.org/ueye_cam

V režimu řízení pomocí GPS je typ zprávy nastaven na `velocity.body`. Pokud nejsou nastavené žádné limity, nebo jsou nastaveny na negativní hodnoty, tak se používají limity z konfiguračního souboru `hl.interface`. Hodnoty pro jednotlivé osy jsou v m/s a rad/s pro yaw, maximální rychlosti, kterých je možné dosáhnout, jsou 5 m/s pro pohyb v horizontálních směrech a 2 m/s pro vertikální pohyb.

Režim fúze dat na HLP nebude využíván, pro tento režim se typ nastaví na hodnotu `position.body`.

- **mav_imu:** Tento typ je využíván při komunikaci přes topic `/fcu/imu_custom`. Zprávy obsahují:
 - kvaternion⁵ obsahující údaje o orientaci kvadrokoptéry,
 - vektor s kruhovou rychlostí,
 - vektor se zrychlením,
 - výšku kvadrokoptéry,
 - hodnotu změny letové výšky kvadrokoptéry.

Aplikace TyraTeleop bude využívat pouze údaj o letové výšce, který bude zobrazen v panelu se stavovými informacemi.

- **mav_status:** Tento typ zpráv je využíván v topicu `/fcu/status` a obsahuje velký počet informací o stavu kvadrokoptéry. Mezi tyto informace patří:
 - napětí baterie,
 - letový mód zvolený na RC ovladači,
 - způsob výpočtu stavu kvadrokoptéry,
 - režim řízení pozice,
 - stav sériového rozhraní mezi HLP a LLP,
 - čas od zapnutí motorů,
 - vytížení HLP,
 - stav motorů - vypnuté nebo zapnuté,
 - stav příjmu gps signálu,
 - počet gps satelitů,
 - debugovací hodnoty,
 - statistika přijatých a odeslaných paketů mezi HLP a Atomboardem.

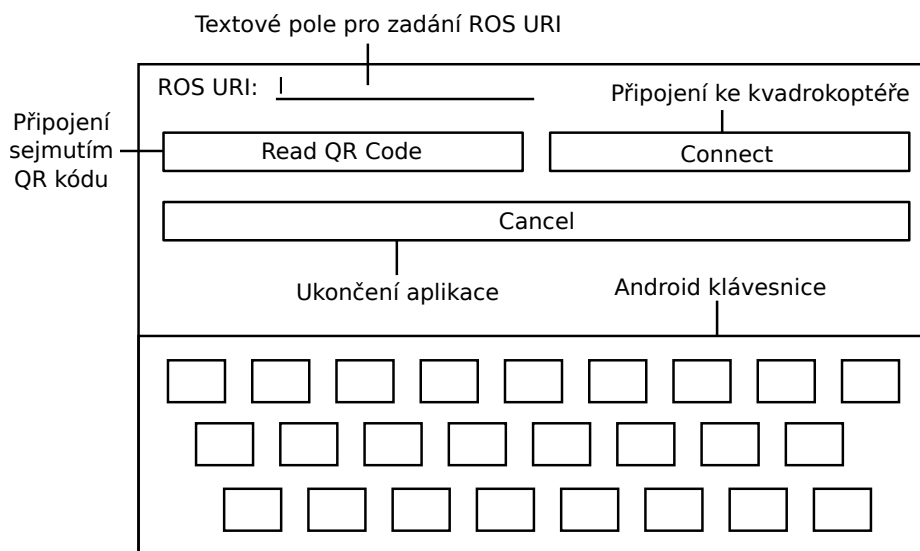
V aplikaci bude zobrazeno napětí baterie, stav příjmu gps signálu a z režimu řízení pozice bude zjištěn aktuální letový mód - buď manuální, nebo GPS.

Pro přenos obrazu jsou použity zprávy typu `CompressedImage`, které vytváří node `uEyeCam` sloužící ke snímání obrazu z kamery na Tyře. Pro nastavení letového módu je použit parametr `position_control`, který je možné dynamicky konfigurovat za běhu nodu.

⁵Nekomutativní rozšíření oboru komplexních čísel, viz <http://cs.wikipedia.org/wiki/Kvaternion>

Připojení ke kvadrokoptěře

Aplikace předpokládá připojení mobilního zařízení do lokální sítě, ve které je přítomna i kvadrokoptéra. Po spuštění aplikace nabídne dialog pro vyplnění informací pro připojení k Tyře, tedy IP adresu a port (ve formátu tzv. `ROS_MASTER_URI`⁶), na kterém je na Tyře spuštěno jádro ROSu (`roscore`). K dispozici je historie zadávaných připojovacích údajů, aby uživatel nemusel při každém připojení zadávat údaje znovu. Po potvrzení se aplikace připojí k `roscore` na Tyře a přejde do základní obrazovky aplikace.



Obrázek 3.10: Návrh dialogu pro připojení.

V případě, že se na zadané adrese nenachází jádro ROS, nebo některý z topiců, tak se aplikace bude 10s snažit připojit. Pokud se jí to nepodaří, uživateli je oznámeno, že se nelze připojit, a aplikace se ukončí.

⁶http://wiki.ros.org/ROS/EnvironmentVariables#ROS_MASTER_URI

Kapitola 4

Realizace, experimenty a vyhodnocení

Tato kapitola popisuje průběh vývoje aplikace TyraTeleop, sloužící k ovládání kvadrokoptéry pomocí tabletu s operačním systémem Android. Tyru bylo nejprve třeba oživit, nainstalovat Atomboard a na něj nainstalovat operační systém, ROS, mav framework a další nezbytnosti. Popsán bude samotný vývoj aplikace, její testování v různých podmínkách a vyhodnocení testů.

4.1 Oživení Tyry

Aby bylo možné ovládat Tyru pomocí mobilního zařízení, bylo nutné ji vybavit počítačem, který by řídící příkazy zpracovával a dále předával autopilotu. Původní Atomboard, který byl na Tyře, měl problémy s komunikací, takže bylo rozhodnuto o koupi nového. Následoval výběr vhodného modelu, instalace operačního systému a dalšího potřebného softwaru a nakonec montáž na Tyru a otestování komunikace.

Výběr vhodné počítačové desky

Prvním krokem k realizaci této bakalářské práce byl výběr vhodné počítačové desky (dále PD), která by měla malé rozměry, nízkou spotřebu, dostatečný výkon a podporu pro běh ROSu. Zváženy byly platformy postavené na procesorových architekturách x86 a ARM ¹. Byl jsem pověřen prozkoumáním trhu, výběrem vhodných kandidátů a navržením nejlepšího řešení ke koupi. Počítačem s nejlepším poměrem cena/výkon byl model Odroid U3² od korejské firmy Hardkernel, který obsahuje čtyřjádrový procesor Samsung Exynos 4412, který je postaven na jádrech ARM Cortex A9³. Pro ARM je připravena jak linuxová distribuce Ubuntu ve verzi 13.10, tak i základní knihovny ROSu. Nevýhodou by byla případná nutnost překladu některých méně obvyklých knihoven ze zdrojových kódů. PD Odroid U3 váží 48g a má velmi malé rozměry, přesto nakonec nebyla vybrána, a to z důvodu absence miniPCI slotu pro výkonnou WiFi síťovou kartu, a drahému pevnému disku, který je nabízen ve formátu eMMC⁴ modulu.

¹<http://www.arm.com/>

²http://hardkernel.com/main/products/prdt_info.php

³<http://www.arm.com/products/processors/cortex-a/cortex-a9.php>

⁴eMMC - embedded Multi-Media Controller, flash paměť a její kontrolér integrovaný na stejném čipu, viz <http://www.datalight.com/solutions/technologies/emmc/what-is-emmc>

Druhým kandidátem byl stejný model jako předchozí nefunkční Atomboard od firmy Commell, ten ovšem jako úložiště používá paměťovou kartu CompactFlash⁵, což je příliš pomalé řešení s omezenou životností z důvodu nízkého počtu možných zápisů a přepisů dat.

Vybranou PD se stal Advantech MIO-2261⁶, zejména kvůli srovnatelné ceně s PD od firmy Commell, ale s lepší nabídkou dostupných rozhraní. Důležitým prvkem je také mSATA slot, umožňující osazení plnohodnotným SSD diskem, což zvyšuje nejen výkon (zejména rychlost spouštění operačního systému), ale i životnost celého řešení oproti použití paměťové karty.

Instalace softwaru a montáž na Tyru

Dalším krokem byla instalace operačního systému. Zvolil jsem odnož Ubuntu s prostředím Xfce, Xubuntu 12.04. Xfce bylo zvoleno zejména kvůli nižším nárokům na výkon procesoru a velikost operační paměti, bez subjektivního snížení uživatelského komfortu.

Po instalaci operačního systému bylo nastaveno automatické přihlašování a připojení k bezdrátové síti v konfiguračním souboru `/etc/network/interfaces`. Grafické prostředí bylo deaktivováno, ale pro případné pohodlnější ladění a instalaci dalších balíčků je výhodné, že si jej uživatel může zapnout. Dále byl nainstalován ROS Hydro⁷, nejnovější verze Mav Frameworku a balíku `ethzasl_sensor_fusion`⁸, který je vyžadován pro správnou funkci Mav Frameworku.

Pro správnou funkci kamery bylo nutné nainstalovat příslušné ovladače⁹ pro kamery s připojením přes USB 2.0 společně s balíkem `ueye_cam`, který umožní posílat obraz z kamery přes ROS.

Díky stejným rozměrům, jako měl předchozí Atomboard na Tyře byla instalace nového Atomboardu rychlá a obešla se bez komplikací. Dále bylo nainstalováno spojení s Autopilot boardem, řešené převodníkem z USB na sériové rozhraní.

4.2 Vývoj aplikace TyraTeleop

Praktický vývoj aplikace navržené v kapitole 3 umožní ověřit navržené koncepty v praxi a provést testy bezpečnosti a stability ovládání kvadrokoptéry.

Vývoj probíhal ve vývojovém prostředí Android Studio postaveném na platformě IntelliJ IDEA¹⁰. Android Studio je v současné době stále ve vývoji a je dostupné ve verzi Early access preview. Při vývoji jsem však nezaznamenal žádné chyby nebo nedostatky a s tímto vývojovým prostředím jsem spokojen. Bylo použito Android Studio ve verzi 0.5.2.

Použité moduly

Hlavním modulem aplikace je modul `TyraTeleop`. Ten rozšiřuje třídu `RosActivity` a obsahuje hlavní aktivitu aplikace. V tomto modulu jsou zpracovávány výstupy z ovládacích prvků, konfigurace pomocí menu a jsou zde také spuštěny dva komunikační nody, `Communication` a `RosImageView`, jejichž zdrojové kódy jsou v oddělených modulech.

⁵<http://en.wikipedia.org/wiki/CompactFlash>

⁶http://www.advantech.com.tw/products/80af04dc-200f-44bb-bd7c-e99d43b72bd9/MIO-2261/mod_a06b85aa-8734-42c1-8ff7-a3854bf1eadf.aspx

⁷<http://wiki.ros.org/hydro>

⁸http://wiki.ros.org/ethzasl_sensor_fusion

⁹<http://en.ids-imaging.com/download-ueye.html>

¹⁰<http://developer.android.com/sdk/installing/studio.html>

Modul `Communication` slouží k hlavní komunikaci s kvadrokoptérou. Jsou zde vytvářeny kontrolní zprávy typu `mav_ctrl` a publikovány. Odesílání kontrolních příkazů je řízeno časovačem, který odesílá zprávy každých 80ms, frekvence odesílání zpráv je tedy 12,5Hz.

Dále jsou zde vytvořeny subscribery pro čtení stavových informací z Tyry. Při obdržení zprávy na některém čteném topicu je zavolána metoda, která data zpracuje a aktualizuje popisky na hlavní obrazovce aplikace. Jelikož je frekvence příchozích zpráv velká a aktualizace popisků náročná na výkon, byla provedena optimalizace, která minimalizuje náročnost a zároveň příliš neomezuje frekvenci překreslování. Díky optimalizaci se provede překreslení jen při každém padesátém volání dané metody, a údaje jsou tak aktualizovány přibližně dvakrát za sekundu.

Modul `RosImageView` z balíku `android_gingerbread_mr1`, který je součástí `android_core` z knihovny `rosjava`[12] byl upraven kvůli svému nízkému výkonu. Původní widget `ImageView`, který zajišťoval zobrazení bitmapy, měl velmi nízký výkon. Proto byl po hledání řešení nahrazen widgetem `SurfaceView`¹¹. Tím bylo dosaženo dostatečné rychlosti překreslování obrazu z kamery.

Ovládací prvky

Další moduly obsahují implementaci ovládacích prvků. Balík `android_honeycomb_mr2`, který je také součástí `android_core`, obsahuje implementaci virtuálního joysticku (dále VJ). Tato implementace však obsahuje velké množství animací, které VJ zpomalují na hranici použitelnosti a zároveň obsahuje komunikační metody, které publikují řídicí zprávy typu `TwistStamped` na topic, který se jmenuje `cmd_vel`. Veškerý nadbytečný zdrojový kód byl z VJ vymazán a výsledkem je rychlý ovládací prvek s funkcí, která je popsána v kapitole 3. VJ nyní neobsahuje žádné přebytné animace a byly naprogramovány zachytávací plochy, jak je popsáno v 3.3. Tento modul se jmenuje `VirtualJoystick`.

Modul obsahující posuvník byl vytvořen modifikací VJ. Upravena byla grafická textura do podoby posuvníku a zakázán horizontální pohyb ovládacího pinu. Při dotyku na plochu posuvníku je uložena pozice dotyku. Při pohybu prstu po displeji je pak volána metoda `onContactMove`, která vypočítává rozdíl mezi předchozí a současnou pozicí dotyku a o tuto hodnotu posune pin posuvníku. Výsledkem je ovládací dotykový prvek, který není nutné vizuálně sledovat, a přesto nabízí vysokou přesnost ovládání. Modul obsahující posuvník se jmenuje `Slider`. Oba ovládací prvky a přenos obrazu z kamery zobrazuje obrázek 4.1.

Přepínání mezi ovládacími režimy

Přepínání mezi manuálním a GPS módem je v Mav frameworku realizováno pomocí parametru `position_control`. Jedná se o parametr, který je možné dynamicky konfigurovat za běhu nodu pomocí balíku `dynamic_reconfigure`. Ten má API¹² pro knihovnu `rospy`¹³, ovšem podpora pro knihovnu `rosjava` chybí. Přepínání mezi módy je pro aplikaci TyraTeleop jedna z klíčových věcí, rozhodl jsem se tedy tento problém vyřešit implementací nodu, který bude běžet na Tyře a využívat knihovnu `rospy` ke změně parametru `position_control`. Tento node čeká na příkazy na topicu `/fcu/position_params`. Řídící hodnoty jsou typu `std_msgs/String` a jsou stejné, jako v případě nastavování režimu přes

¹¹Viz <https://code.google.com/p/rosjava/issues/detail?id=141>

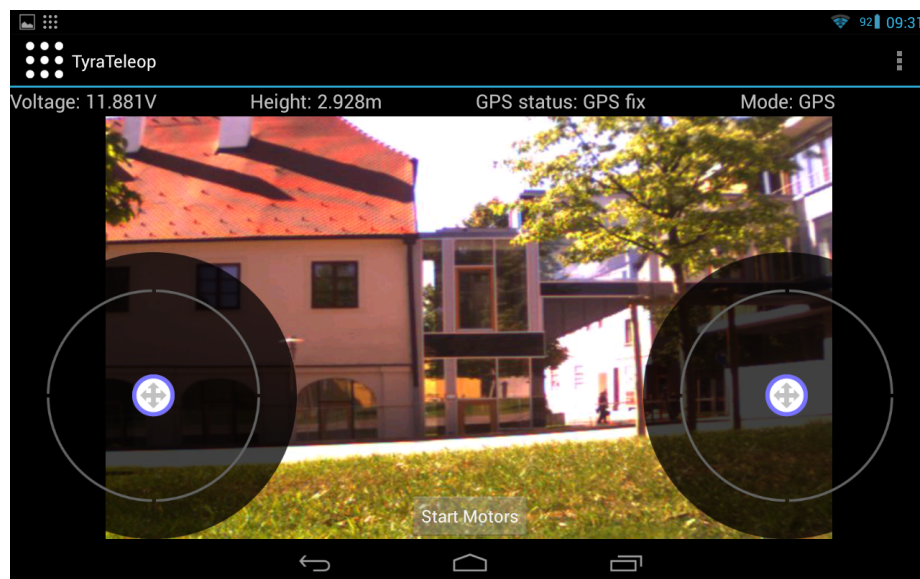
¹²API - Application Program Interface, sada příkazů, funkcí a protokolů pro vývoj programů spolupracujících s určitým systémem[6]

¹³Klientská knihovna ROS pro jazyk Python, viz <http://wiki.ros.org/rospy>



Obrázek 4.1: Screenshot z aplikace v manuálním módu a přenosem obrazu.

`dynamic_reconfigure` z příkazové řádky - `POSCTRL_OFF` pro manuální mód a `POSCTRL_GPS` pro GPS mód. V případě zaslání jiných hodnot na tento topic se neprovede žádná akce.



Obrázek 4.2: Screenshot z aplikace v GPS módu a přenosem obrazu.

4.3 Experimenty s aplikací

Experimenty s aplikací mají za cíl ověřit použitelnost a funkčnost návrhu, popsaného v kapitole 3.

Příprava k testování

Testování létajících robotů je značně obtížné, jelikož hrozí riziko pádu při nezkušeném operátorovi, nebo chybě v ovládací aplikaci. Z toho důvodu jsme s kolegy, kteří vyvíjejí také pro Tyru, vymysleli způsob, jak pádům předejít. Využíváme teleskopickou tyč, která slouží k mytí oken a provaz, který přivážeme k Tyře. Tím se zajistí, že při selhání programu, nebo lidského faktoru, nedojde k pádu a poškození, nebo dokonce zničení kvadrokoptéry. Aby se provaz při stoupání nezamotal do rotorů, je opatřen pružnou gumou, která provázek stahuje a umožňuje tak vystoupat o téměř 80cm, aniž by došlo ke klesání povoleného provázku do rotorů. Přesto je třeba, aby byl člověk, který obsluhuje teleskopickou hůl, nanejvýš opatrný a vyrovnával napnutí provazu tak, aby k podobné věci nedošlo. Celé zabezpečovací zařízení je vyobrazeno na fotografii 4.3.



Obrázek 4.3: Fotografie Tyry s teleskopickou tyčí a jisticím provazem.

Nejlépe se nám testovalo ve třech lidech. Jeden člověk držel teleskopickou hůl, další RC ovladač, kterým deaktivoval sériovou linku v případě nouzové události, a poslední testoval a ladil svoji aplikaci.

Pro spojení mezi mobilním zařízením a kvadrokoptérou jsme v laboratoři používali WiFi router, pro spojení během venkovního testování jsme využili funkce hotspot na mobilním telefonu, který byl po celou dobu v kapse pilota, a tedy blízko kvadrokoptéry. Testování maximálního dosahu se také neobešlo bez komplikací. V případě, že Mav framework přestane dostávat řídicí příkazy, nebo klesne frekvence jejich příchodu pod 10 Hz, je rychlost motorů nastavena na volnoběh[4]. Pokud by se podobná situace stala za letu, musel by člověk držící RC ovladač buď vypnout sériovou linku a přistát, nebo vypnout ovladač a tím aktivovat nouzový režim, který přistane autonomně. Při testování se ukázalo, že vzdálenosti okolo 10m již způsobují nepříjemné latence komplikující ovládání, a při vzdálenostech větších klesá frekvence příkazů natolik, že jsou ignorovány. Vzhledem k těmto skutečnostem probíhalo testování výhradně na krátké vzdálenosti mezi mobilním hotspotem, kvadrokoptérou a mobilním zařízením s ovládací aplikací.

Omezující faktory při testování

Testování probíhalo nejčastěji na trávníku mezi budovami L a Q naší fakulty. Jelikož poloautonomní mód vyžaduje GPS, bylo vyloučeno testování v laboratoři. V laboratoři jsme testovali pouze neletecké atributy prací, například start a vypnutí motorů, přenos obrazu nebo přenos stavových informací.

Z důvodu použití teleskopické tyče a provazu jako zabezpečovacího zařízení nebylo možné testovat ve výškách větších, než přibližně 4m. Žádné testy neprobíhaly bez zabezpečovacího zařízení.

Při testování se také ukázalo, že stabilizace kvadrokoptéry v GPS módu je relativně spolehlivá, občas však začne, při požadavku na udržení aktuální pozice, letět nějakým směrem a test je třeba ukončit. Létání také není možné blízko vysokých budov. Při přiblížení se budově L přibližně na 4m už byly znát zhoršené schopnosti udržování pozice a v několika případech začala kvadrokoptéra rychle letět některým směrem, a to bez zavinění pilota. V takovém případě byl test také ukončen a začal z výchozí pozice.

Návrh experimentů a jejich průběh

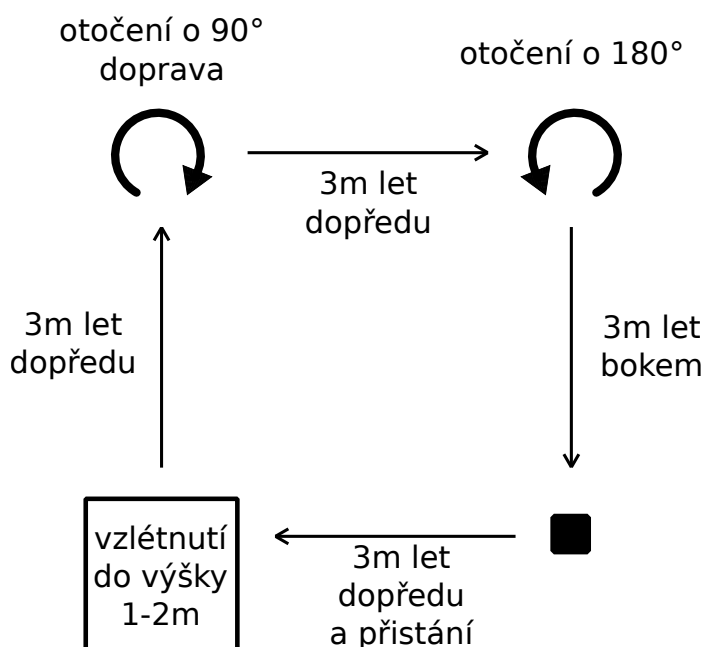
Při testování je třeba zohlednit všechny kroky od startu aplikace, přes ovládání kvadrokoptéry, až po její ukončení a zařadit je do testů. Testy obsahují následující kroky:

- **Připojení k Tyře:** Tento krok vyžaduje znalost IP adresy, která byla Tyře přidělena na lokální síti. Jelikož používáme jak Wi-Fi router v robotické laboratoři, tak mobilní hotspot, ponechali jsme přidělení adres na DHCP serveru. Přidělená adresa je zjišťována přes aplikaci Fing pro Android. Po startu aplikace se otevře dialog, ve kterém je uživatel vyzván k zadání Robot URI. Po doplnění adresy uživatel stiskne tlačítko Connect a aplikace se připojí ke kvadrokoptěře.
- **Změna ovládacích režimů:** K přepínání mezi letovými a ovládacími režimy slouží menu, jejíž ikona je v pravém horním rohu. Uživatel bude vyzván, aby:
 - změnil ovládání na pohybové,
 - nastavil použití GPS módu,
 - vrátil se do výchozího nastavení s manuálním módem a deaktivovaným pohybovým ovládáním.
- **Zapnutí a vypnutí motorů:** Uživatel bude vyzván, aby vypnul a zapnul motory pomocí aplikace. K ovládání motorů slouží tlačítko ve středu spodního okraje aplikace.
- **Čtení stavových informací:** Uživatel bude vyzván, aby sdělil informace o:
 - napětí baterie,
 - letové výšce,
 - stavu příjmu gps signálu
 - použitým letovým módem
- **Ovládání v manuálním módu:** Uživatel bude mít za úkol nastavení manuálního módu, start motorů, vzletnutí a let v manuálním módu bez použití ovládání pohybem. Úkolem bude krátký let ve tvaru čtverce, kdy se po obletění vrátí zpět do výchozí pozice a přistane.

- **Ovládání v GPS módu:** Uživatel zjistí, zda je dostupný GPS signál a přepne aplikaci do GPS módu. Následuje aktivace motorů, vzletnutí a obletění stejné dráhy, jako v případě předchozího testu.
- **Ovládání pohybem:** Úkol bude stejný, jak v předchozích bodech, ale s tím rozdílem, že se použije pohybové ovládání naklápěním tabletu.

Porovnání s jinými způsoby ovládání

Hlavním cílem testů bude porovnání pohodlnosti, intuitivnosti a přesnosti ovládání v porovnání s RC ovladačem. RC ovladač je výchozím způsobem ovládání kvadrokoptéry a díky fyzickým ovládacím prvkům by měl nabízet nejlepší komfort ovládání. Oběma způsoby ovládání bude proletěna přibližně stejná trasa, sestávající z obletění dráhy ve tvaru čtverce s hranou přibližně 3m, několika otočení a následného přistání. Trasa je vyobrazena na obrázku 4.4 a je shodná s testovací trasou kolegy Petra Běla, jelikož testování létání probíhalo společně. Po celou dobu letu trasou se testovací piloti budou snažit o co nejpřesnější udržení letové výšky 1 až 2 metry.



Obrázek 4.4: Testovací trasa.

Tyto testy pomohou ověřit výsledky návrhu uživatelského rozhraní, zejména použitelnost a uživatelský komfort.

4.4 Výsledky experimentů

Experimenty ukázaly silné a slabé stránky navrženého ovládání v porovnání s ovládáním RC ovladačem a ovládáním pomocí gest. Dále byla testována intuitivnost a jednoduchost připojení ke kvadrokoptěře, změny ovládacích režimů, ovládání motorů a čtení stavových informací. Letové testy byly prováděny mnou a kolegou Petrem Bělem, ostatní testy, zaměřující se na neletové atributy aplikace, byly testovány 5 osobami.

Připojení k Tyře

Tento krok byl všechny testovací subjekty snadný, pokud byla v adresním řádku zadaná již dříve použitá adresa. Pokud byla stejná, subjekty se kliknutím na tlačítko Connect připojili k Tyře. Pokud byla předchozí adresa odlišná od zadávané, bylo třeba adresu editovat, což trvalo déle. Test při prázdném adresním řádku ukázal komplikovanost zadávání adresy ve formátu ROS_MASTER_URI (například: `http://192.168.104.100:11311`). Subjektům bylo nutné vysvětlit, jak má adresa vypadat, a zadávání tak trvalo delší dobu.

Testovací subjekt	Předvyplněná adresa	Změna adresy	Zadání celé adresy
subjekt č.1	0.5s	12s	28s
subjekt č.2	1s	8s	40s
subjekt č.3	1s	13s	30s
subjekt č.4	2s	12s	32s
subjekt č.5	1s	20s	35s

Tabulka 4.1: Tabulka s dobou provádění jednotlivých úkolů testovacími subjekty.

Ve většině případů praktického použití aplikace se používá poslední zadaná adresa, nebo se modifikují poslední číslice z adresy. Celá adresa se zadává jen zřídka, ale dlouhé časy zadávání a složitost použitého formátu jsou podnětem ke zlepšení.

Změna ovládacích režimů

Díky nízkému počtu nabídek v menu bylo přepínání módů testovacími subjekty velmi rychlé. Stačilo vysvětlit, kde se nachází tlačítko pro menu a k čemu slouží jednotlivé nabídky.

Testovací subjekt	Pohybové ovládání	GPS mód	Výchozí nastavení
subjekt č.1	5s	4s	7s
subjekt č.2	7s	7s	10s
subjekt č.3	4s	3s	4s
subjekt č.4	5s	5s	9s
subjekt č.5	6s	3s	7s

Tabulka 4.2: Doby potřebné pro změnu režimů.

Zapnutí a vypnutí motorů

U tohoto testu nebudou uváděny časy, jelikož zapnutí i vypnutí bylo provedeno u všech subjektů velmi rychle a nevyskytl se žádný problém. Tlačítko s nápisem Start motors nebo Stop motors je dostatečně intuitivní a přehledné.

Čtení stavových informací

Řádek se stavovými informacemi je dobře viditelný. Subjektům nečinilo žádné potíže sdělit stavové informace o kvadrokoptéře. Z reakcí vyplývá, že místo napětí baterie by bylo vhodnější zobrazovat stav nabití baterie v procentech, nebo zbývající čas letu. Ostatní informace byly jasné a pochopitelné.

Letové testy

Letové testy byly nejdůležitější částí testování. Ukázaly, zda je možné využít navržené uživatelské rozhraní k praktickému ovládání kvadrokoptéry. Testy byly prováděny dvěma osobami a porovnávalo se mezi RC ovladačem, ovládání pomocí joysticků a pohybovým ovládáním. Všechny testy byly provedeny jak v manuálním, tak v GPS módu. Test spočíval v proletění trasy podle obrázku 4.4. Tabulka 4.3 obsahuje časy potřebné pro proletění dané trasy. Hodnota v závorce udává počet pokusů, který byl třeba k úspěšnému proletění trasy. Maximální počet pokusů byl 5, poté byl test označen jako neúspěšný.

Testovací pilot	RC ovladač		Virtuální prvky		Pohybové ovládání	
	Manuální	GPS	Manuální	GPS	Manuální	GPS
pilot č.1	1:35 (2)	1:08 (1)	neúspěšné (5)	1:27 (1)	2:30 (4)	1:50 (2)
pilot č.2	neúspěšné (5)	1:20 (1)	1:58 (3)	1:35 (1)	neúspěšné (5)	1:43 (4)

Tabulka 4.3: Doby potřebné pro zproletění testovací trasy.

4.5 Vyhodnocení experimentů

Z experimentů vyplývá, že navržené uživatelské rozhraní je použitelné pro ovládání kvadrokoptéry. Nejvíce se osvědčilo ovládání pomocí virtuálních joysticků v GPS módu, které nabízí podobný komfort, jako RC ovladač. Jako druhé se umístilo pohybové ovládání v GPS módu. To nenabízí takovou přesnost, jako ovládání joysticky, ale je stále použitelné. Třetí nejlepší výsledek podalo ovládání pomocí virtuálních joysticků v manuálním módu. V tom je velmi složité letět v malé letové výšce dostatečně stabilizovaně. Jako poslední se umístilo ovládání pohybem v manuálním módu, které bylo pro netrénované piloty velmi obtížné, jelikož vyžaduje vzájemnou koordinaci ovládání motorů a jemného pohybu se zařízením.

Nejlépe se pro ovládání osvědčily virtuální ovládací prvky na displeji zařízení, které se v GPS módu vyrovnaly pohodlnosti řízení pomocí RC ovladače. Létání v GPS módu je pro málo trénovaného pilota výrazně jednodušší, než v manuálním módu.

Kapitola 5

Závěr

Cílem bakalářské práce byl návrh uživatelského rozhraní autopilota kvadrokoptéry pro mobilní zařízení typu tablet. Návrh byl implementován a byla otestována jeho použitelnost a uživatelská přívětivost. Aplikace TyraTeleop s tímto uživatelským rozhraním nabízí manuální a poloautonomní mód pro řízení kvadrokoptéry a možnost volby mezi virtuálními ovládacími prvky, nebo ovládáním pomocí detekce pohybu zařízení. Do aplikace je také přenášén obraz z kamery na kvadrokoptéry, což umožňuje ovládání i bez přímého sledování kvadrokoptéry.

V kapitole 2 byly prozkoumány existující řešení uživatelských rozhraní, vybavení platformy, na které je kvadrokoptéra postavena a použitého softwaru. V kapitole 3 byly navrženy ovládací prvky a letové módy aplikace. V kapitole 4 pak byla popsána implementace návrhu a experimenty s implementovanou aplikací.

Experimenty ukázaly, že v poloautonomním módu, který využívá stabilizace pomocí GPS, je kvadrokoptéra dostatečně stabilní na to, aby byla pohodlně ovládána pomocí mobilní aplikace. V práci byl kladen důraz na vyladění ovládacích prvků i pro použití v manuálním módu bez poziční a výškové stabilizace. Ukázalo se však, že ovládání tímto způsobem je podobně náročné, jako s RC ovladačem a vyžaduje velmi zkušeného pilota. Testování probíhalo výhradně v malých výškách s kvadrokoptérou zavěšenou provazem na teleskopické tyči. Tím byla limitována maximální letová výška a letové testy tak byly prováděny ve výškách 1-3m.

Přínosem práce je ověření, že uživatelské rozhraní pro mobilní zařízení může nabízet dostatečný komfort ovládání tak složitého zařízení, jako je kvadrokoptéra. Byl proveden rozbor schopností platformy AscTec Pelican, na které je kvadrokoptéra postavena, umožňující jejich další zlepšení. I přes své dobré stabilizační schopnosti trpí kvadrokoptéra problémy s příjmem GPS signálu v blízkosti budov a občasným náhodným zahájením letu některým směrem.

Další vývoj aplikace by se zaměřil na optimalizaci přenosu obrazu pro rychlejší překreslování, lepší notifikace v případě chyb připojení a zvýšení dosahu ovládání, který je nyní na hranici 10m. Podnětem pro zlepšení je také zpřesnění ovládání pomocí pohybu a stabilizačních schopností kvadrokoptéry, například použitím sonarů.

Literatura

- [1] Apple: iTunes app store [online]. Dostupné z: <https://itunes.apple.com/us/genre/ios/id36?mt=8>, [cit. 2014-05-12].
- [2] Ascending: AscTec Hummingbird with AutoPilot User's Manual [online]. Dostupné z: <https://itunes.apple.com/us/genre/ios/id36?mt=8>, [cit. 2014-05-12].
- [3] Ascending: AscTec Research Line Catalogue 2014 [online]. Dostupné z: http://www.asctec.de/downloads/flyer/AscTec-Research-Line_Catalogue_2014.pdf, [cit. 2014-05-12].
- [4] Ascending: Wiki [online]. Dostupné z: <http://wiki.asctec.de/>, [cit. 2014-05-12].
- [5] Chandrashekar, K.: Android vs. iOS Market Share: Google's OS Snags Q1 2014 [online]. Dostupné z: <http://www.latinpost.com/articles/12202/20140510/android-vs-ios-market-share-googles-os-snags-q1-2014.htm>, 2014-05-10 [cit. 2014-05-11].
- [6] Christensson, P.: The Tech Terms Computer Dictionary [online]. Dostupné z: <http://www.techterms.com/>, [cit. 2014-05-12].
- [7] Dillow, C.: Parrot AR.Drone 2.0 Review: Fly Higher, Farther, and More Intuitively [online]. Dostupné z: <http://www.popsci.com/technology/article/2012-07/parrot-ardrone-20-review-enhanced-drone-piloting-experience-seeks-long-lasting-battery>, 2012-03-07 [cit. 2014-05-04].
- [8] Google: Sensor Overview [online]. Dostupné z: http://developer.android.com/guide/topics/sensors/sensors_overview.html, [cit. 2014-05-01].
- [9] Google: Play Store [online]. Dostupné z: <https://play.google.com/store>, [cit. 2014-05-12].
- [10] Parrot: AR.Drone 2.0 [online]. Dostupné z: <http://ardrone2.parrot.com/>, [cit. 2014-05-12].
- [11] PowerUp: PowerUp 3.0 Smartphone Controlled Paper Airplane [online]. Dostupné z: <http://www.poweruptoys.com/pages/app-toys>, [cit. 2014-05-12].
- [12] Rosjava: Rosjava Core Documentation [online]. Dostupné z: http://docs.rosjava.googlecode.com/hg/rosjava_core/html/javadoc/index.html, [cit. 2014-05-09].

- [13] ROS.org: AscTec Mav Framework [online]. Dostupné z: http://wiki.ros.org/asctec_mav_framework, [cit. 2014-05-09].
- [14] ROS.org: Documentation [online]. Dostupné z: <http://wiki.ros.org/>, [cit. 2014-05-12].
- [15] SCHEIBEL, J.-B.; PIERSON, C.; MARTIN, B.; aj.: Virtual Stick in Caret Positioning on Touch Screens [online]. In *Proceedings of the 25ième conférence francophone on l'Interaction Homme-Machine*, 2013, iSBN 978-1-4503-2407-6.
- [16] Walkera: QR W100 [online]. Dostupné z: <http://www.walkera.com/en/showgoods.php?id=1776>, [cit. 2014-05-03].

Příloha A

Obsah CD

K bakalářské práci je přiloženo CD s touto adresářovou strukturou:

- `application`
 - `src` - zdrojové soubory projektu pro Android Studio
 - `apk` - instalační apk soubor
- `poster` - plakát
- `video` - prezentační video
- `thesis`
 - `src` - zdrojové soubory pro $\text{\LaTeX}$
 - `pdf` - přeložený pdf soubor s textem bakalářské práce