

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Yunifly – autonomní řízení bezpilotních letounů

Bedřich Said

Brno 2012

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

obor SOČ: 18. Informatika

Yunifly – autonomní řízení bezpilotních letounů
Yunifly - Autonomous Control of Unmanned Aircraft

| | |
|-------------|---|
| Autor: | Bedřich Said |
| Škola: | Gymnázium Brno tř. Kpt. Jaroše 14 658 70 Brno 2 |
| Konzultant: | Bc. Martin Vejnár |

Brno 2012

Prohlášení

Prohlašuji, že jsem svou práci vypracoval samostatně, použil jsem pouze podklady citované v práci a postup při zpracování práce je v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) v platném znění.

V dne

podpis:

Bedřich Said

Poděkování

Děkuji svému konzultantovi Mgr. Marku Blahovi za odborné vyjádření k této práci, za jeho podnětné příspěvky týkající se kontroly a kritiky, za rady a návrhy, které mi aktuálně průběžně poskytoval během mé tvůrčí činnosti. Srdečně mu děkuji za všechny chvíle, které mi věnoval svým obětavým a rychlým přístupem v dané problematice. Děkuji DDM Junior Brno, zejména Bc. Martinu Vejnárovi a Ing. Jiřímu Váchovi za poskytnutí zázemí, rad a podkladů nutných pro vývoj tohoto projektu.

Bedřich Said

Abstrakt

Komerční řízení bezpilotních letounů je běžnému uživateli téměř nedostupné. Mnoho mobilních telefonů disponuje internetovým připojením a sensory jako GPS, akcelerometr nebo kamera. Této skutečnosti využijeme k vývoji nízkonákladového bezpilotního letounu, který využívá služeb palubního mobilního telefonu.

Projekt Yunifly vyvíjí řídicí systém, který je primárně určen pro autonomní řízení bezpilotních letounů. Skládá se z řídicí elektroniky, která komunikuje s mobilním telefonem, a softwaru, který umožňuje řízení celého letounu a komunikaci s uživatelem. Pokud nemáme k dispozici data potřebná k řízení, použijeme program pro řízení letounu pouze na základě dat z akcelerometrů a dynamického gyroskopu. Akcelerometr nám vždy poskytne korektní údaje, neboť gravitaci nelze odstínit jako například GPS signál. V neposlední řadě byl vyvinut simulátor, na kterém můžeme předem otestovat programy určené pro roboty a snížit tak riziko nefunkčnosti či poškození celého stroje.

Díky tomuto systému může i běžný modelář přeměnit svůj model v plně autonomní stroj, který plní zadaný úkol. Vývojářům jsou tak sníženy náklady na pořízení všech modulů potřebných k řízení autonomního robota a je jim umožněno otestovat funkčnost nově vyvíjených programů v idealizovaném virtuálním světě.

Klíčová slova: RC model, letadlo, řízení letu, mobilní telefon, software, simulátor, elektronika

Abstract

Commercial management of unmanned aircraft is inaccessible for standard users. Many mobile phones have an Internet connection and sensors like GPS, accelerometer or camera. We will use this fact to develop a low-cost unmanned aircraft that uses mobile phone services.

The Yunifly project develops a control system which is primarily designed for autonomous control of RC models. It consists of control electronics which communicates with a mobile phone and software that allows control and communication with the user. If we do not have basic data for control, we use a program for control the airplane based on data from accelerometers and dynamic gyroscope. An accelerometer always gives us correct information because the gravity cannot be lost, such as the GPS signal. A simulator was developed for testing new programs. The simulator allows to reduce the risk of damage because we can test our programs before we use them in a real plane.

Standard modelers can convert their models to a fully autonomous system that performs a specified task. The developers do not have to buy all sensors to control the plane because they can use their mobile phone. They can test their new programs in an idealized virtual world.

Key words: RC model, aircraft, flight control, mobile phone, software, simulator, electronics

Obsah

| | | |
|----------|---|-----------|
| 1 | Úvod | 1 |
| 2 | Architektura systému Yunify | 3 |
| 2.1 | Řízení robota mobilním telefonem | 3 |
| 2.2 | Software pro řídicí elektroniku | 4 |
| 3 | Vývojová deska | 7 |
| 3.1 | Řízení a příjem signálu z RC přijímače | 7 |
| 3.2 | Architektura řídicí elektroniky | 8 |
| 3.3 | Součásti elektroniky | 9 |
| 3.4 | Funkce hardwaru | 9 |
| 3.4.1 | Komunikace desky Basic a Main | 9 |
| 3.4.2 | Příklady extrémních zařízení | 9 |
| 3.5 | Náklady na pořízení elektroniky Yunify | 12 |
| 4 | Řízení letu pouze pomocí akcelerometrů a dynamického gyroskopu | 14 |
| 4.1 | Měření pohybu letounu ve vzdušném prostoru | 14 |
| 4.1.1 | Vztahy pro výpočet pohybu letounu | 15 |
| 4.1.2 | Data získaná akcelerometrem a gyroskopem | 16 |
| 4.1.3 | Data získaná kalibrací před startem | 16 |
| 4.1.4 | Data získaná během prvního letu | 17 |
| 4.2 | Řízení letounu na základě dat z akcelerometru a dynamického gyroskopu | 18 |
| 5 | Simulátor skupiny robotů | 20 |
| 5.1 | Popis a funkce simulátoru | 20 |
| 5.1.1 | Server | 20 |
| 5.1.2 | Client | 21 |
| 5.1.3 | Viewer | 22 |
| 6 | Shrnutí | 27 |
| A | Podrobná specifikace hardwaru Yunify | 30 |
| A.1 | Hmotnost a rozměry | 30 |
| A.2 | Pinová konfigurace | 31 |
| A.3 | Schéma zapojení elektroniky | 34 |
| B | API, firmware a řízení mobilním telefonem | 34 |
| B.1 | Firmware Yunify | 34 |
| B.1.1 | Firmware desky Basic | 34 |
| B.1.2 | API desky Main | 34 |
| B.1.3 | Knihovny Yunify | 37 |
| B.2 | Řízení letounu na základě dat z akcelerometrů a dynamického gyroskopu | 38 |
| B.3 | Aplikace pro mobilní telefon | 39 |

Seznam obrázků

| | | |
|----|--|----|
| 1 | Hardwarová konfigurace systému Yunifly | 4 |
| 2 | Komunikace uživatele a elektroniky s mobilním telefonem | 5 |
| 3 | Snímání obrazu mobilní kamerou v reálném čase | 5 |
| 4 | Architektura platformě nezávislých programů a knihoven | 6 |
| 5 | Kompilace programu pro danou platformu | 6 |
| 6 | Fotografie řídicí elektroniky | 7 |
| 7 | Schéma stabilizovaného dálkově řízeného letu pomocí RC přijímače | 8 |
| 8 | Architektura řídicí elektroniky | 10 |
| 9 | Schéma připojení externích modulů | 12 |
| 10 | Schéma trojosého akcelerometru a dynamického gyroskopu | 14 |
| 11 | Znázornění silového působení za letu | 15 |
| 12 | Řízení směru letu pomocí naklápění | 17 |
| 13 | Schéma programu pro řízení letounu pomocí akcelerometrů | 19 |
| 14 | Komunikace mezi jednotlivými částmi simulátoru | 20 |
| 15 | Specifikace komunikace se serverem | 21 |
| 16 | Architektura a funkce serveru | 22 |
| 17 | Chování klienta od startu po ukončení | 23 |
| 18 | Architektura a funkce vieweru | 24 |
| 19 | Export simulace jako 3D animace - sledování čáry | 24 |
| 20 | Export simulace jako 3D animace - hledání kapky vody | 25 |
| 21 | Ukázka simulace 2D viewerem v reálném čase | 25 |
| 22 | Ukázka simulace 3D viewerem v reálném čase | 26 |
| 23 | Řídicí elektronika v rozloženém a složeném stavu | 30 |
| 24 | Rozměry celé elektroniky Yunifly | 31 |
| 25 | Schéma konektoru, kterým je spojena deska Main s deskou Basic | 31 |
| 26 | Pinová konfigurace elektroniky | 32 |
| 27 | Elektronické schéma zapojení desky Main | 35 |
| 28 | Elektronické schéma zapojení desky Basic | 36 |
| 29 | Obraz tištěného spoje desky Main bez součástek | 37 |
| 30 | Obraz tištěného spoje desky Basic bez součástek | 37 |

1 Úvod

Komerční řešení autonomního řízení bezpilotních letounů či jiných strojů je v současné době téměř nedostupné běžnému uživateli a komerční software nelze dále upravovat podle vlastních potřeb. Většina novějších mobilních telefonů disponuje vestavěnými sensory jako například GPS, akcelerometry nebo kamera. Díky tomu můžeme pro řízení a navigaci mobilního robota využít sensory v telefonu a možnost připojení k internetu či GSM síti.

Systém Yunifly je open-source soubor hardwaru a softwaru poskytující nástroje pro řízení mobilních autonomních strojů či bezpilotních letounů. Umožňuje snadnější vývoj a ovládání nových mobilních strojů. Skládá se z dvouprocesorové nízkonákladové řídicí elektroniky včetně firmwaru a API. Další součástí je program pro komunikaci letounu s uživatelem pomocí mobilního telefonu a software pro řízení bezpilotních letounů pouze na základě údajů z akcelerometrů a dynamického gyroskopu. Vývojovou pomůckou Yunifly je v neposlední řadě simulátor skupiny uživatelem definovaných robotů, který je navržen pro vývoj a testování synchronizované skupiny robotů plnící zadaný úkol.

Vývojová elektronika Yunifly: Je nízkonákladová dvouprocesorová elektronika určená pro autonomní řízení mobilních strojů. Primárně je navržena k vývoji nízkonákladového bezpilotního letounu. Elektronika umožňuje kompletní uživatelské přeprogramování, avšak procesor Basic obsahuje firmware se základními obsluhami řízení. V případě nefunkčnosti uživatelského programu v procesoru Main zastupuje celé řízení procesor Basic v nouzovém režimu. Je tak výrazně snížena pravděpodobnost havárie způsobená nefunkčností uživatelského programu.

Elektronika je vybavena porty sériové linky pro připojení dalších zařízení jako například WiFi, Bluetooth, GPS, akcelerometru nebo kompasu. Obsahuje slot pro microSD/SDHC kartu o maximální kapacitě 32 GB. Samozřejmostí jsou porty pro řízení serv, regulátorů a čtení analogových i digitálních dat.

Součástí elektroniky je firmware pro procesor Basic a API (application programming interface) pro procesor Main. API umožňuje kompatibilitu elektroniky s knihovnamy Arduino [1] a následnou komaptibilitu se systémy ArduPilot [2] a ArduPilot Mega [3]. Dále umožňuje komunikaci s palubním mobilním telefonem, který lze připojit k internetu nebo získávat data z vestavěné GPS nebo akcelerometrů. Díky tomu odpadá nutnost pořizování dalších sensorů pro kvalitní řízení autonomního stroje.

Software pro autonomní řízení mobilním telefonem: Mobilní telefon je k řídicí elektronice připojen bezdrátově pomocí bluetooth nebo WiFi. Drátové propojení přes USB není podporováno z důvodu nízké kompatibility mezi různými zařízeními. Po spuštění aplikace v mobilním telefonu se telefon připojí k elektronice a případně naváže internetové připojení pro vzdálenou komunikaci s uživatelem. Lze nastavit i pravidelné zasílání SMS zpráv o stavu plnění daného úkolu.

Mobilní telefon poskytuje elektronice data z GPS a vestavěných akcelerometrů s dynamickým gyroskopem. Pokud telefon nedisponuje těmito sensory, je možné nastavit jejich externí připojení k řídicí elektronice. Řídicí elektronika není závislá na připojení mobilního telefonu, jeho odpojením pouze ztrácí možnost komunikace přes internetové připojení. Telefon je možné nahradit GSM modulem.

Pokud je telefon připojen, může řídicí elektronika (nebo uživatel prostřednictvím internetu) spustit nebo zastavit nahrávání kamery mobilního telefonu, získat data z mobilní GPS a akcelerometru. Streamování videa z mobilního telefonu uživateli zařizuje program Skype [20] nebo Fring [21]. Telefon dále umožňuje komunikaci řídicí elektroniky s uživatelem. Díky tomu může uživatel přes internet zasílat elektronice nová data, příkazy nebo dotazy. Aplikaci je možné

upravit tak, aby výpočty probíhaly v mobilním telefonu a řídicí elektronika jen nastavovala konkrétní ovládací prvky a komunikovala s telefonem či dalšími zařízeními (sensors).

Řízení letounu pouze na základě dat z akcelerometrů a dynamického gyroskopu:

Autonomní řízení bezpilotního letounu (helikoptéry, lodě, ...) je podmíněno přesnými údaji o poloze a pohybu. V současné době existují komerční projekty, které umožňují autonomní navigaci, avšak nelze je přeprogramovat nebo upravit dle představ uživatele. Ve sféře open-source je autorovi znám pouze projekt ArduPilot [2] a ArduPilot Mega [3], který se zaměřuje na navigaci letounu pomocí GPS, kompasu a výškoměru. Tyto sensory nám bohužel negarantují stálý příjem korektních dat. Jejich přesnost je závislá na počasí a kvalitě signálu GPS.

Výhodou akcelerometru je měření na základě gravitace, kterou není možné odstínit ani vyrušit. Dynamický gyroskop rozpoznává rychlost rotace vzhledem k inerciální vztažné soustavě. Tento údaj též není možné odstínit.

Pokud letoun ztratí GPS signál nebo data z dalších sensorů, je možné jej omezeně řídit pouze na základě dat z akcelerometrů a dynamického gyroskopu. Toto řízení poskytuje software systému Yunifly, jehož knihovny jsou napsány nezávisle na použité platformě. Program garantuje let s volitelnou rotací letadla vůči zemi. Díky tomu je možné letět zadaným směrem nebo kroužit nad daným místem. Pokud letoun ztratí důležité údaje, může letět zadaným směrem nebo kroužit dokud potřebný signál opět nezachytí. Díky programově nastavitelné rotaci lze jednoduše definovat například akrobatickou sestavu. Řízení lze dokalibrovávat na základě údajů z GPS, kompasu nebo výškoměru, a zajistit tak přesnější polohu letadla vzhledem k definované trajektorii.

Simulátor skupiny robotů: Simulátor je client-server aplikace určená pro vývoj a testování softwaru pro mobilní roboty. Server poskytuje simulaci virtuálního světa s uživatelem definovanými podmínkami, client simuluje robota připojeného do simulace a viewer je uživatelské rozhraní pro sledování a řízení celé simulace. Simulátor je postaven na definovaném API, které podporují i procesory Atmel [4] a řídicí systémy postavené na vyšších platformách. Uživatel si může dodefinovat vlastní funkce vyhovující jeho API.

Simulátor poskytuje real-time simulaci skupiny robotů ve virtuálním světě, kteří mohou komunikovat mezi sebou nebo s reálnými roboty připojenými přes internet, bluetooth, WiFi nebo po sériové lince. Vývojáři je tak umožněno vyvíjet a testovat software pro roboty bez nutnosti jejich fyzické existence. Může idealizovat podmínky virtuálního světa a sledovat tak chování programů v nereálném nebo těžko sestavitelném prostředí. Jedná se o nástroj, který zjednodušuje vývoj nového softwaru pro roboty a odstraňuje problémy nekorektního chování reálných robotů.

2 Architektura systému Yunifly

System Yunifly je navržen tak, aby jej bylo možné použít v několika následujících konfiguracích (graficky znázorněno na obrázku 1):

1. Pouze elektronika Yunifly autonomně řídí daný stroj, případně čte informace z připojených sensorů. (vyznačeno černě)
2. Uživatel řídí svého robota pomocí RC vysílače. Informace z RC vysílače čte elektronika Yunifly, která řídí robota programově nebo na základě dat z RC vysílače. (vyznačeno fialově)
3. K elektronice Yunifly je připojeno komunikační zařízení (Bluetooth, WiFi, ZigBee, ...), které zajišťuje komunikaci mezi uživatelem a robotem, případně mezi několika různými roboty. Uživatel tak může zasílat řídicí elektronice příkazy nebo se dotazovat na informace získané během plnění úkolu. Skupinka robotů může synchronně plnit zadaný úkol a komunikovat po síti vytvořené pomocí připojeného komunikačního modulu. (vyznačeno zeleně)
4. K řídicí elektronice připojíme mobilní telefon (například pomocí technologie bluetooth), který obsahuje některé sensory (kamera, GPS, akcelerometr, ...) a umožňuje komunikaci přes GSM síť nebo přes internet. Díky tomu je uživatel v kontaktu s robotem vždy, když je připojen k internetu a robot disponuje dostatečným signálem. (vyznačeno červeně)
5. Pokud uživatel nemá k dispozici elektroniku Yunifly nebo potřebuje použít vlastní modul, může využít Yunifly API pro svůj přístroj, které zajistí kompatibilitu mezi jeho přístrojem a dalšími zařízeními. Na obrázku lze zaměnit elektroniku Yunifly například za jednotku ArduPilot nebo vlastní řídicí elektroniku.

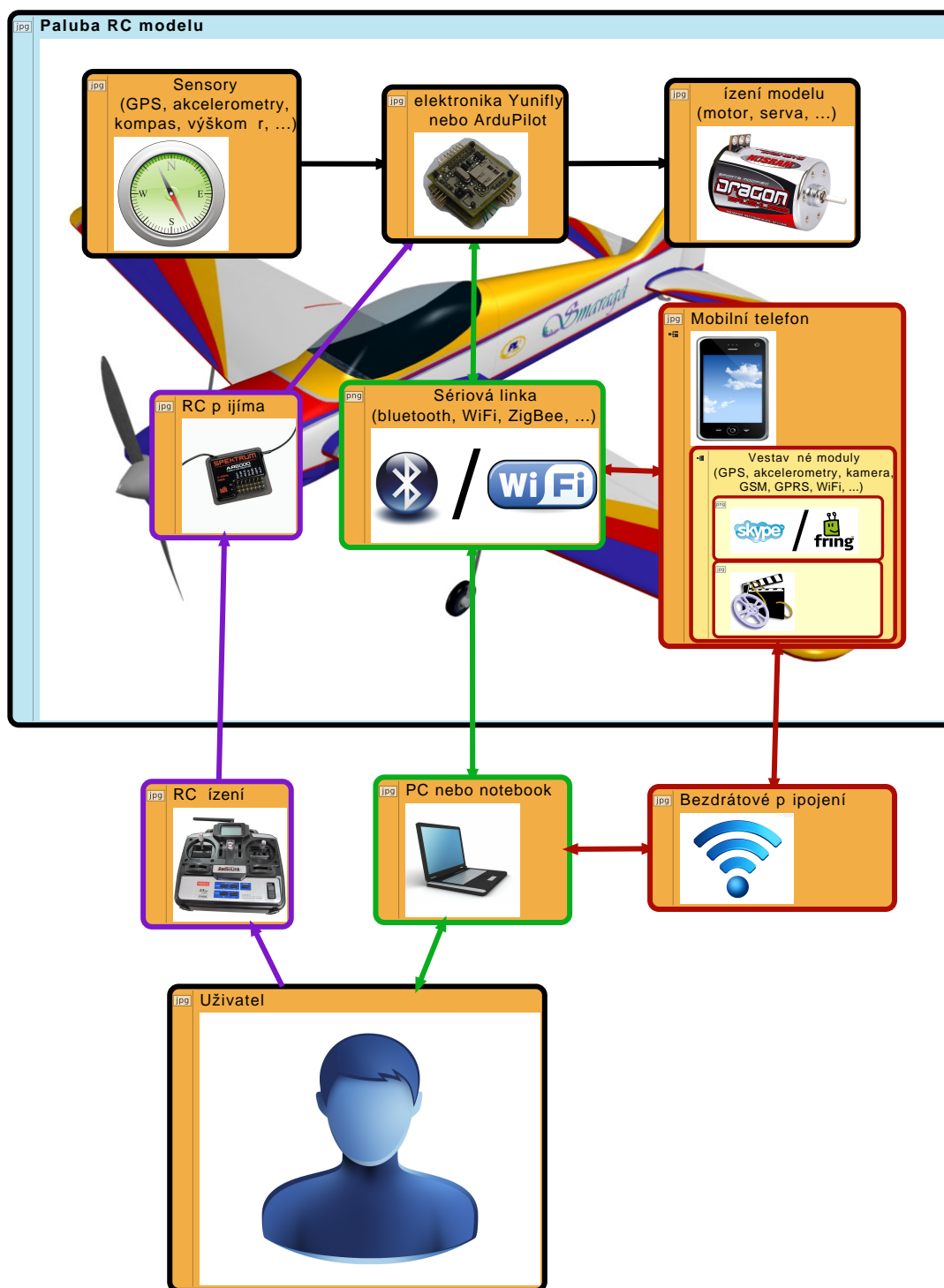
2.1 Řízení robota mobilním telefonem

Pro komunikaci a programové řízení robota lze využít palubního mobilního telefonu, který dále komunikuje s uživatelem nebo serverem přes internet. Nezávisle lze využít následujících funkcí:

1. Mobilní telefon poskytuje data z vestavěných sensorů (kamera, GPS, akcelerometr, ...). Tato data zpracovává nebo odesílá. Příjemcem může být uživatel, server nebo řídicí elektronika.
2. Řídicí program pracuje uvnitř telefonu, využívá dat ze sensorů, případně komunikace přes internet. Uživatel musí dodat řídicí program spustitelný v mobilním telefonu.
3. Telefon poskytuje internetové nebo GSM připojení. Je tak zajištěna komunikace s uživatelem nebo dalšími spolupracujícími roboty.

Uživatel s robotem komunikuje přes sériovou linku pomocí sady příkazů, která je popsána v dokumentaci. Pro komunikaci uživatel využívá počítač nebo mobilní zařízení podporující sériovou komunikaci. Řízení s mobilním telefonem je graficky znázorněno na obrázku 2.

Streamování videa a fotografování mobilním telefonem je zajištěno externí aplikací Skype [20] nebo Fring [21]. Řídicí skript umožňuje pouze pořízení fotografie vestavěnou kamerou. Fotografie lze uložit do paměti nebo s ní dále pracovat podobně jako s dvojrozměrným polem typu integer. Použití vestavěné kamery je graficky znázorněno na obrázku 3.

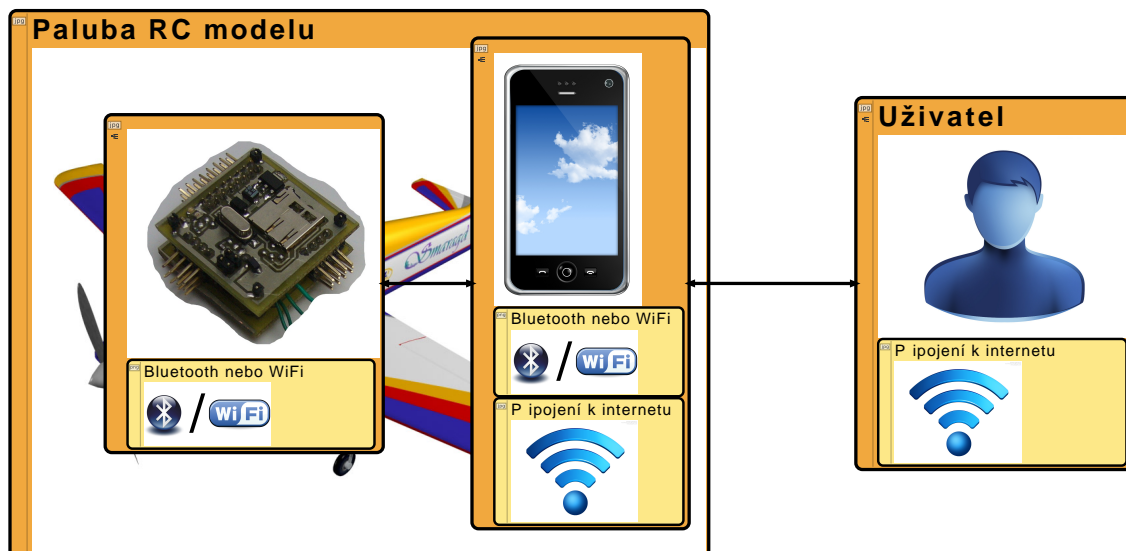


Obrázek 1: Hardwarová konfigurace systému Yunifly

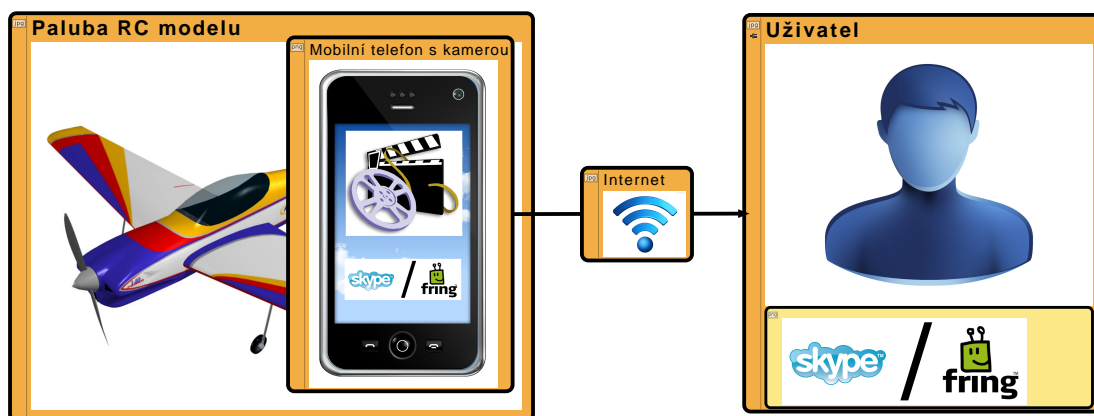
2.2 Software pro řídicí elektroniku

Knihovny řídicího softwaru jsou psány platformě nezávisle. Využívají definovaného API, které lze upravit potřebám jiných systémů nebo platform. Struktura závislosti knihoven je znázorněna na obrázku 4.

Pokud vytváříme vlastní řídicí program, využíváme knihoven Yunifly API. Program můžeme zkompileovat přímo pro řídicí elektroniku nebo pro simulátor Yunimin, na kterém program

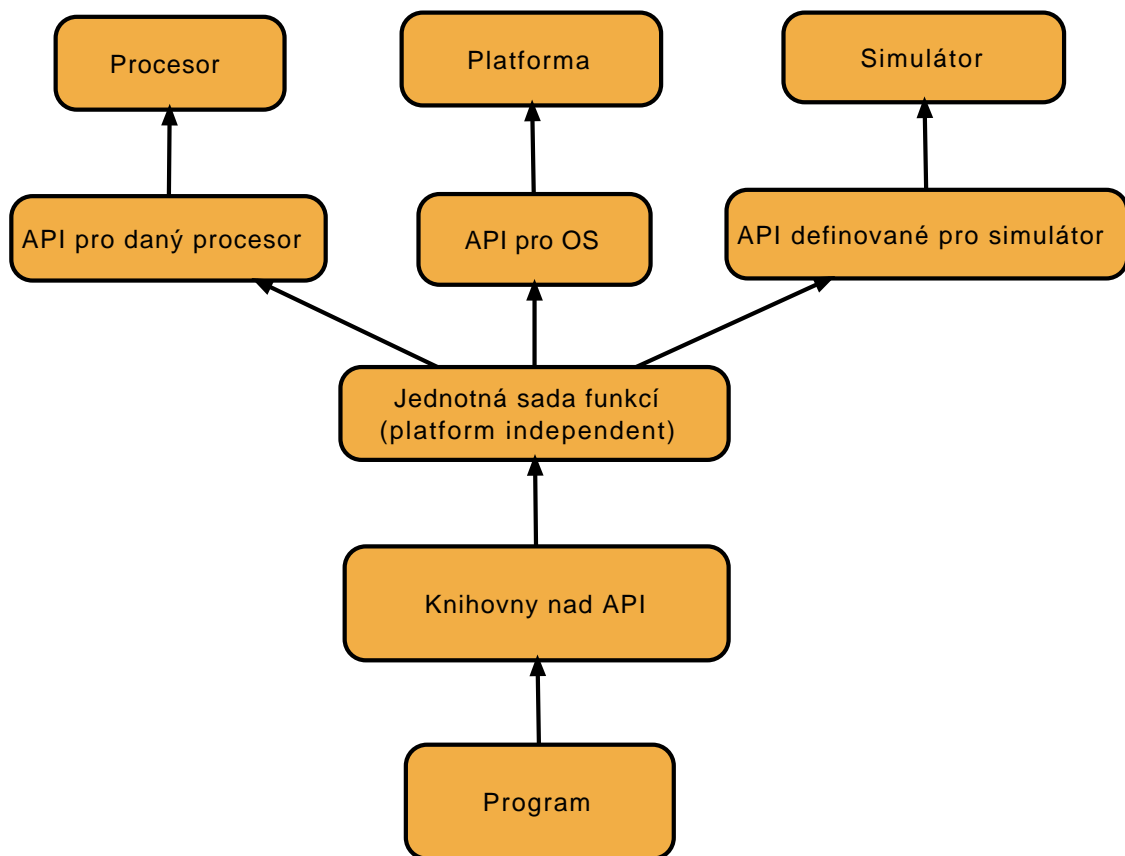


Obrázek 2: Komunikace uživatele a elektroniky s mobilním telefonem

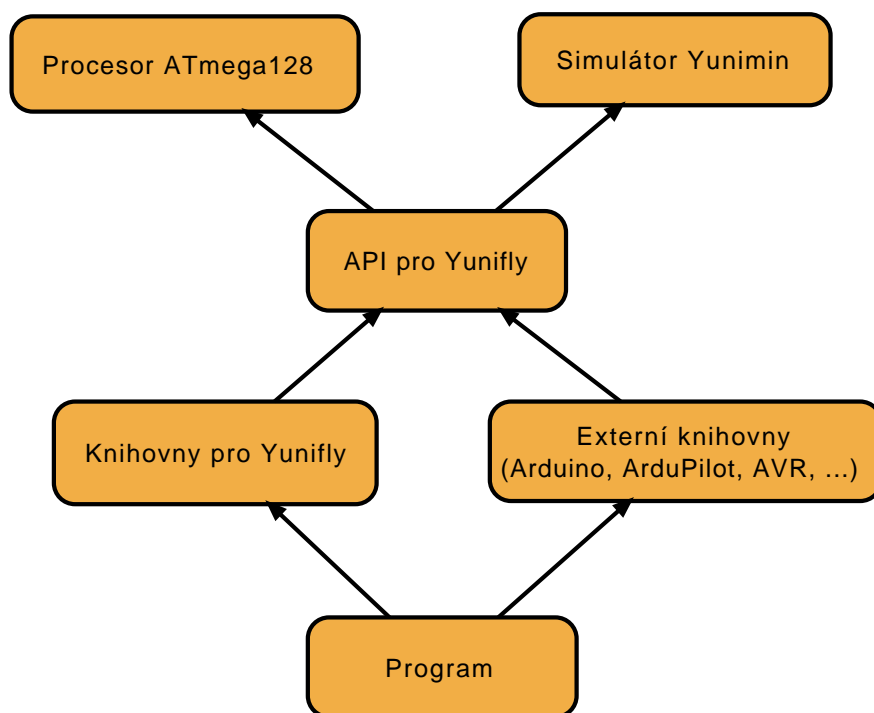


Obrázek 3: Snímání obrazu mobilní kamerou v reálném čase

otestujeme. Pokud potřebujeme kompilovat pro jiný systém, který zde není podporován, musíme definovat vlastní API pro systém, na kterém chceme programy spustit. Možnosti kompilace programů jsou znázorněny na obrázku 5.



Obrázek 4: Architektura platformě nezávislých programů a knihoven



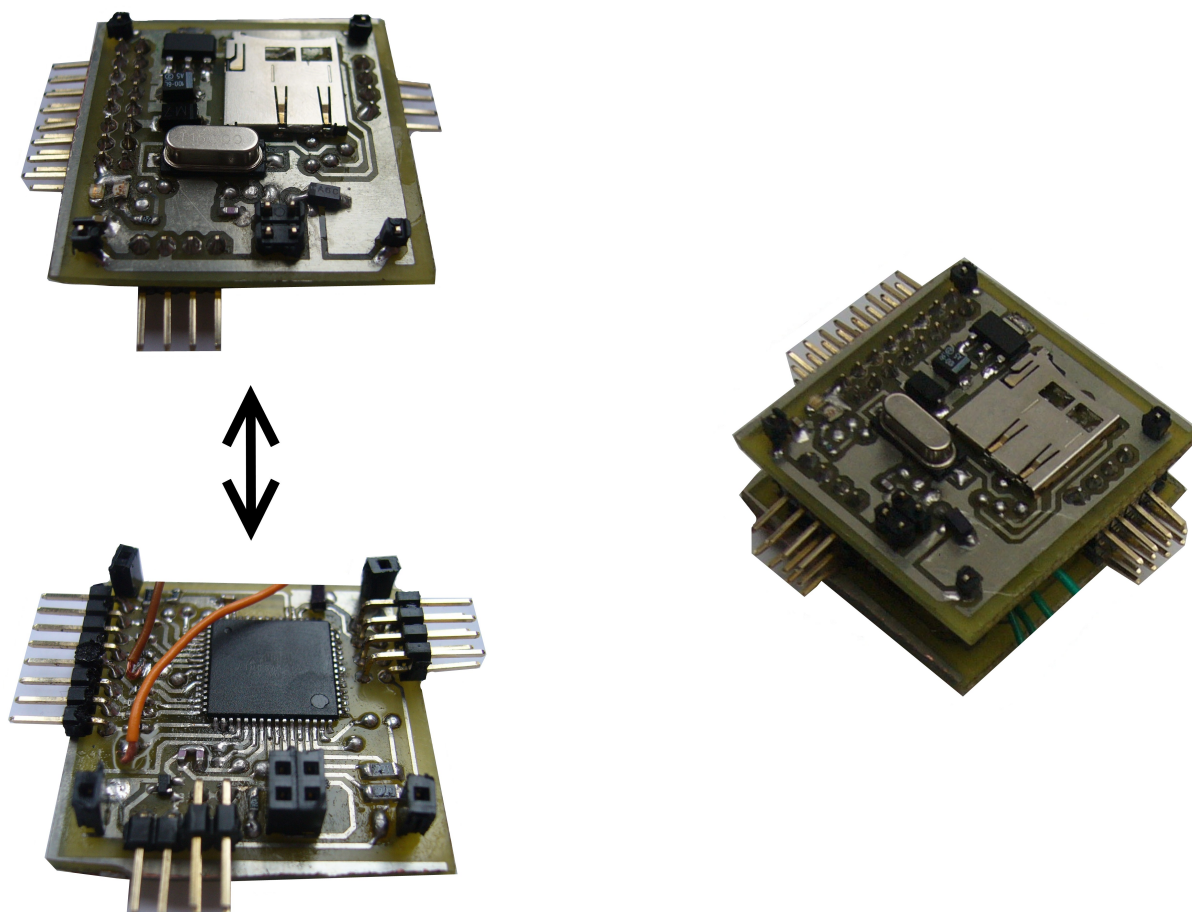
Obrázek 5: Kompilace programu pro danou platformu

3 Vývojová deska

Elektronika Yunifly se skládá ze dvou částí, z desky Basic a desky Main. Obě desky jsou na sobě nezávislé a lze je použít jak samostatně, tak v níže znázorněném propojení. Desky mezi sebou mohou komunikovat pomocí rozhraní I²C [13] nebo USART [14].

Připojení externích zařízení k Yunifly není povinné. Nutné je pouze napájení a komunikace s deskou ovládanými zařízeními.

Podrobná specifikace připojování je uvedena v příloze A.



Obrázek 6: Fotografie řídicí elektroniky

3.1 Řízení a příjem signálu z RC přijímače

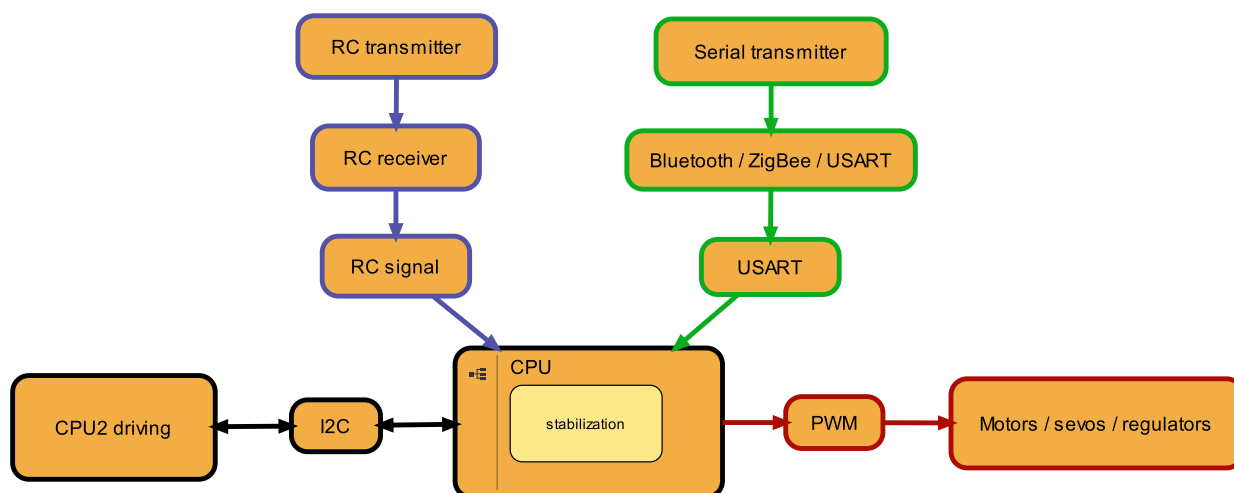
Standardní ze země řízené letouny nezpracovávají přijatý signál a přímo se jím řídí. Aby bylo možné přijatý signál dále zpracovávat nebo létat bez pozemní navigace, je nutné, aby byl letoun řízen vlastním procesorem.

Standardní modelářské výrobky využívají k řízení RC signál, který je vyslán pilotem ze země a přijímán RC přijímačem na palubě letounu. Elektronika Yunifly může přijímat signál od libovolného modelářského RC přijímače s doporučeným napětím 3 V nebo 5 V, maximálně však 12 V.

Elektronika Yunifly očekává příjem RC signálu jako posloupnost PWM pulsů pro jednotlivá serva. Lze přijímat každý kanál samostatně nebo souhrnným protokolem, kdy je tento signál procesorem dekódován a lze jej dále zpracovávat. Dekódování probíhá podle standardního

přenosového protokolu. Délka pulsu pro jedno servo je 1-2 ms, kde 1,5 ms označuje střední hodnotu. Tyto pulsy jsou sériově seskládány podle pořadí serv a odeslány.

Ve schématu 7 jsou ukázány možnosti přijímání signálu pro řízení letounu. Jednotlivé barvy označují nezávislé celky, které zajišťují příjem signálu nebo samotné řízení. Červeně je znázorněn výstup, který je reprezentován sadou PWM výstupů pro serva a regulátory. Informace o řízení letu můžeme zasílat nejen pomocí RC přijímače, ale také pomocí libovolného bezdrátového digitálního sériového zařízení. Elektronika nutně nevyžaduje dálkové řízení, neboť autonomitu letounu může poskytnout druhý procesor s řídicím programem.



Obrázek 7: Schéma stabilizovaného dálkově řízeného letu pomocí RC přijímače

3.2 Architektura řídicí elektroniky

Celá řídicí elektronika je primárně navržena pro vyrovnávání stability letounu a testování nových aplikací. Další schopnosti elektroniky jsou chápány jako sekundární účel.

Elektronika je rozdělena na dvě základní části, na desku Basic a desku Main. Tyto desky jsou k sobě rovnoběžně přiloženy a za provozu tvoří kompaktní celek, jehož konektory jsou vyvedeny po bočních stranách. Toto rozložení umožňuje vyvedení tří řad pinů nad sebou, které upotřebíme například při připojování PWM regulátorů a serv. Celá elektronika je zároveň kryta ze všech stran a díky tomu je sníženo riziko poškození řídicí elektroniky.

Dvouprocesorová technologie byla zvolena ke snížení rizika výpadku elektroniky způsobeného chybou v programu a následného poškození nebo zničení celého letounu. Další výhodou je možnost nezávislého řízení stability jedním procesorem a zpracovávání řídicích požadavků druhým procesorem. Druhý procesor může též sloužit k autonomnímu řízení celého letounu. K funkčnímu řízení stačí jeden procesor, díky tomu můžeme druhý procesor za letu například přeprogramovat.

K elektronice je možné připojit mnoho externích zařízení, některá z nich jsou popsána v sekci 3.4.2. K připojení stabilizačních sensorů potřebujeme konektory sériové linky nebo sběrnice I²C. K těmto konektorům můžeme připojit mnohem širší spektrum externích zařízení. Schéma 8 ukazuje možné připojení externích modulů ke konektorům desky Basic a desky Main.

Volba dalších zařízení a architektury je ovlivněna požadavky primárního účelu řízení stability. Méně prioritní konektory, jako například AD převodníky, jsou z kapacitních a hmotnostních důvodů vyvedeny pouze jako pady, které lze vyvést jako samostatné konektory. Pro zesílení mechanické odolnosti je napájecí a propojovací konektor obou desek rozmístěn do jejich rohů. Díky tomu jsou obě desky k sobě uchyceny celkem v pěti bodech. Propojovací konektor umožňuje

odpojení desky Basic od desky Main i přes jejich pevné spojení. Napájení se primárně připojuje k výstupu regulátoru od hlavního motoru, který toto napájení poskytuje na PWM konektoru. Avšak elektroniku můžeme připojit například přímo k bateriím nebo ke stabilizovanému zdroji 3,3 V.

3.3 Součásti elektroniky

Baterie – napájení Každá deska je napájena 3,3 V a odebírá za provozu 100 mA. Napájení můžeme připojit buď přímo, nebo přes vestavěný stabilizátor TS1117CW33 [5], který lze napájet maximálním napětím 12 V. Maximální protékající proud je stanoven na 1 A.

Motory, serva, regulátory Všechna zařízení ovládaná pulsní šířkovou modulací (PWM) jsou připojena k jednomu z šesti 16bitových PWM konektorů nebo k jednomu 8bitovému konektoru. Zařízení je možné ovládat ve dvou režimech, jako serva nebo jako PWM zařízení s libovolnou délkou pulsů.

Analogové vstupy Analogové piny měří na svém vstupu napětí v rozsahu od nuly do napájecího napětí. Lze takto měřit například stupeň vybití baterií a závčas tak varovat před jejich vybitím. Dále tyto konektory můžeme využít pro čtení analogových sensorů a podobných zařízení.

Digitální vstupy/výstupy Každý datový konektor na desce je možné naprogramovat jako standardní digitální vstup nebo výstup. V programu nastavíme, zda budeme daný konektor používat v přednastaveném režimu nebo jako digitální vstup/výstup.

Komunikace Desky podporují komunikaci pomocí dvou nezávislých linek USART, jedné linky SPI a sběrnice I²C. Maximální podporovaná rychlost komunikace je 512 kb/s.

3.4 Funkce hardwaru

Základní elektronika Yunify je navržena tak, aby bylo možné připojit co nejvíce komponent. Níže je uveden seznam testovaných komponent, které úspěšně spolupracovaly s elektronikou Yunify. Softwarová implementace jednotlivých komponent je popsána v příloze B.

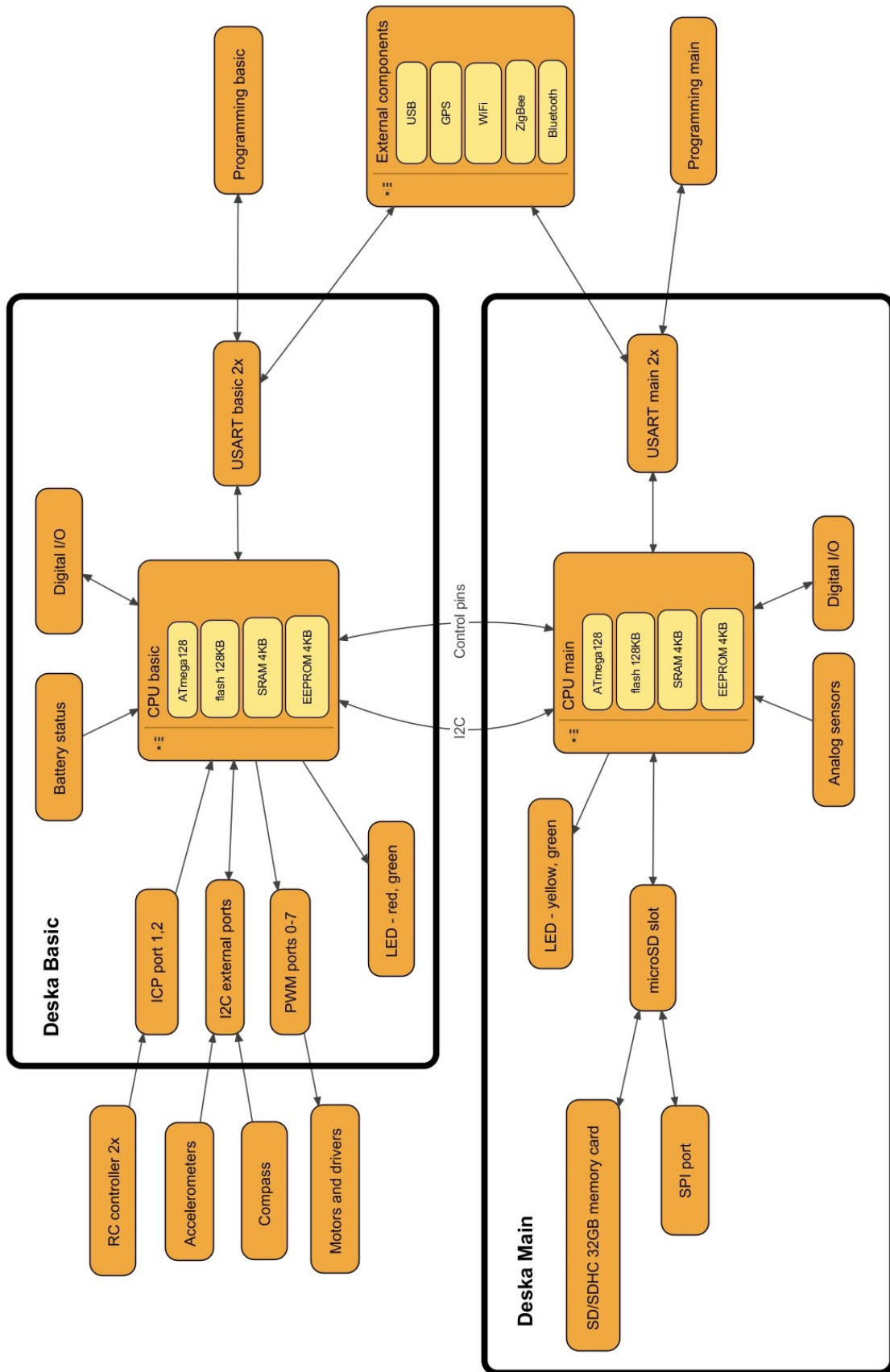
3.4.1 Komunikace desky Basic a Main

Komunikace mezi oběma deskami probíhá po sběrnici I²C. Tato sběrnice je také vyvedena na desce jako konektor. Je tedy možné do komunikace zapojovat i další zařízení podporující tuto sběrnici.

Deska Basic může používat pro svou funkci firmware, který umožňuje podporu nouzového režimu. Jedná se o program Yunify, který je schopen autonomně řídit celý stroj, případně zpracovávat příkazy desky Main. Basic desce zasíláme veškeré požadavky a příkazy z desky Main přes sběrnici I²C.

3.4.2 Příklady extreních zařízení

MicroSD/SDHC karta Slot pro vložení microSD karty se nachází na desce Main. Je připojen k SPI portu, který umožňuje komunikaci nejen s SD kartou, ale s libovolným zařízením, které podporuje tuto sériovou linku. Komunikace byla testována s kartou microSD 2 GB a kartou microSDHC 4 GB [11]. Maximální rychlost čtení i zápisu je omezena rychlostí CPU na 512 kb/s.



Obrázek 8: Architektura řídicí elektroniky

Serva a regulátory řízené pomocí PWM Deska obsahuje 8 PWM výstupů, které umožňují řízení serv a regulátorů motorů či jiných zařízení. Regulátory můžeme ovládat v režimu servo nebo v celém rozsahu hodnot. Nemusíme ručně přepočítávat hodnoty ekvivalentní pro řízení serv. Při testování byla využita modelářská serva „Hitec 8,0g HS-55 Feather Standard Micro RC Servo“ [6].

RC přijímač Obě desky podporují připojení až dvou RC přijímačů. Yunifly postupně čte data, která jsou přijímačem zaslána a může je dále libovolně zpracovávat. Může přímo na základě dat z přijímače ovládat další zařízení nebo tato data dále upravovat.

Díky tomu deska Yunifly nabízí létání v režimu učitel/žák, ve kterém jsou připojeny dva přijímače. Jeden přijímač je veden jako prioritní. Pokud žák udělá při létání nějakou chybu nebo řízení přestane zvládat, učitel může jedním pohybem převzít kontrolu nad celým letounem, aniž by si s žákem vyměňoval ovládací zařízení.

Yunifly může data z RC přijímače využívat za úplně jiným účelem. Můžeme například ovládat aktuální pohled externí kamery při autonomním letu, což by bez procesorem řízené elektroniky bylo zcela nemyslitelné.

GPS Kompatibilita Yunifly byla testována s GPS modulem „GPS smart antenna module, LS20032“ [7]. GPS je k Yunifly připojena přes universal asynchronous receiver/transmitter (USART). V knihovně Yunifly je implementována třída GPS, která umožňuje komunikaci s výše uvedeným GPS modulem.

Akcelerometr Kompatibilita Yunifly byla testována s trojosým akcelerometrem „CHR-6d Digital Inertial Measurement Unit“ [8], který komunikuje po sériové lince (USART). V knihovně Yunifly je implementována třída „accelerometer“, která umožňuje čtení aktuálního zrychlení a přepínání citlivosti akcelerometru.

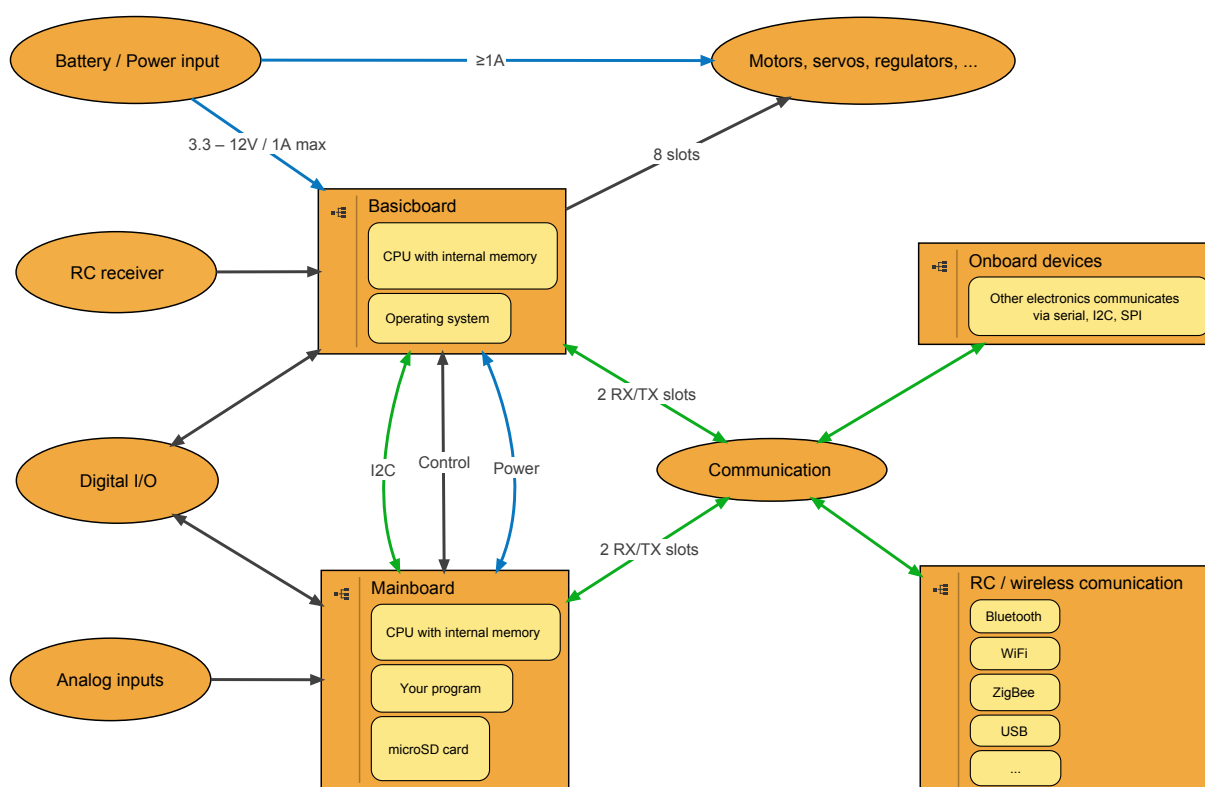
Kompas Při testování byl použit elektronický kompas „CMPS03 - Compass Module“ [9] komunikující po sběrnici I²C. Knihovna Yunifly umožňuje čtení aktuálního směru pomocí třídy „compass“, která uživateli vrací úhel odklonu od severního směru.

Bluetooth S deskou Yunifly je možné komunikovat pomocí bluetooth modulu například s počítačem, který umožňuje nové naprogramování elektroniky nebo přepsání interní paměti. Díky tomu můžeme s Yunifly pracovat bezdrátově včetně programování. Pokud používáme obě desky s podporou nouzového režimu, můžeme desku Main přeprogramovat dokonce za letu.

Dále bluetooth modul umožňuje standardní komunikaci, avšak nevýhodou jeho použití je krátký dosah signálu. Pokud chceme naše zařízení ovládat například mobilním telefonem, můžeme bluetooth modul použít pro vzájemnou komunikaci. Deska Yunifly může být díky tomu řízena přes internet, ke kterému se připojí pomocí mobilního telefonu.

Kompatibilita Yunifly byla testována s bluetooth modulem „SERIAL PORT ADAPTER OEM 310I – CB-OEMSPA310I-04“ [10].

| Typ zařízení | Název zařízení | Komunikační protokol | Datasheet |
|---------------|---|----------------------|-----------|
| GPS | GPS smart antenna module, LS20032 | USART | [7] |
| Akcelerometr | CHR-6d Digital Inertial Measurement Unit | USART | [8] |
| Kompas | CMPS03 - Compass Module | I ² C | [9] |
| Bluetooth | SERIAL PORT ADAPTER OEM 310I – CB-OEMSPA310I-04 | USART | [10] |
| RC přijímač | Spektrum DSM2 AR6100 | PWM | |
| Servo | Hitec 8,0 g HS-55 Feather Standard Micro RC Servo | PWM | [6] |
| MicroSD karta | KINGSTON Micro Secure Digital (Micro SD) 2 GB | SPI | [11] |



Obrázek 9: Schéma připojení externích modulů

3.5 Náklady na pořízení elektroniky Yunifly

Jedním z cílů vývoje Yunifly bylo udržet co nejnižší pořizovací náklady. Výhodou nízkých pořizovacích nákladů je také větší dostupnost programovatelné elektroniky pro modeláře a jiné aktivity zabývající se stavbou malých mobilních strojů.

Pořizovací cena obou desek elektroniky Yunify je srovnatelná s pořizovací cenou běžného RC přijímače. Cena nezahrnuje externí komponenty jako například GPS, akcelerometry, bluetooth modul a další součásti, jejichž cena může být několikanásobně vyšší.

Vývoj celé elektroniky včetně všech součástí je řádově nesrovnatelný s pořizovací cenou desky Yunify.

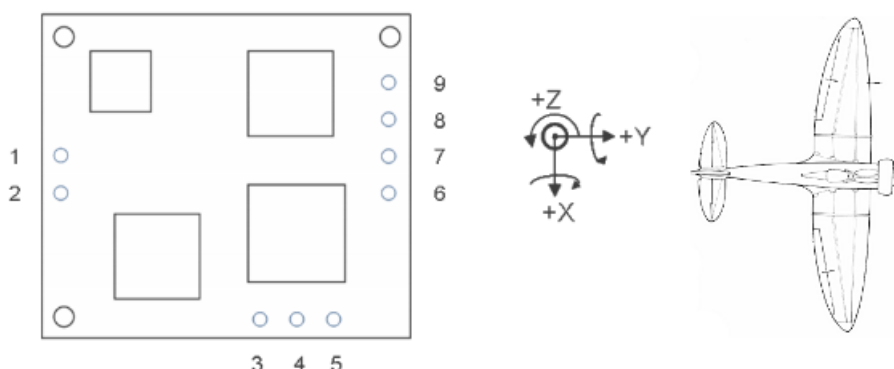
4 Řízení letu pouze pomocí akcelerometrů a dynamického gyroskopu

Pokud chceme, aby se letoun řídil sám bez pomoci pilota, musíme řídicímu systému dodat informace o poloze a rotaci celého letadla. Tyto informace získáváme prostřednictvím sensorů jako například GPS, kompas, akcelerometr, výškoměr nebo rychloměr.

Open-source projekty ArduPilot [2] a ArduPlane [3] poskytují knihovny pro navigaci letounu pomocí GPS, výškoměru, akcelerometrů a gyroskopu. Pro plnohodnotnou navigaci si s těmito sensory sice vystačíme, ale pokud některé ze sensorů nebudou pracovat správně, může dojít ke kolizi.

Projekt Yunifly se snaží získat co nejvíce informací pouze na základě dat z trojosého akcelerometru a dynamického gyroskopu, který je součástí například novějších mobilních telefonů. Trojosý akcelerometr poskytuje systému vektor aktuálního zrychlení letounu včetně gravitačního zrychlení a dynamický gyroskop poskytuje aktuální rychlost rotace kolem všech os. Máme jistotu, že akcelerometr i gyroskop podávají vždy korektní údaje s odchylkou, neboť gravitaci nelze rušit jako například GPS signál. Data nejsou závislá ani na proměnlivosti tlaku vzduchu či rychlosti větru.

Obrázek 10 zobrazuje orientaci akcelerometru včetně dynamického gyroskopu vzhledem k celému letounu. Číselné označení konektorů odpovídá značení použitého akcelerometru v dokumentaci výrobce [8].



Obrázek 10: Schéma trojosého akcelerometru a dynamického gyroskopu

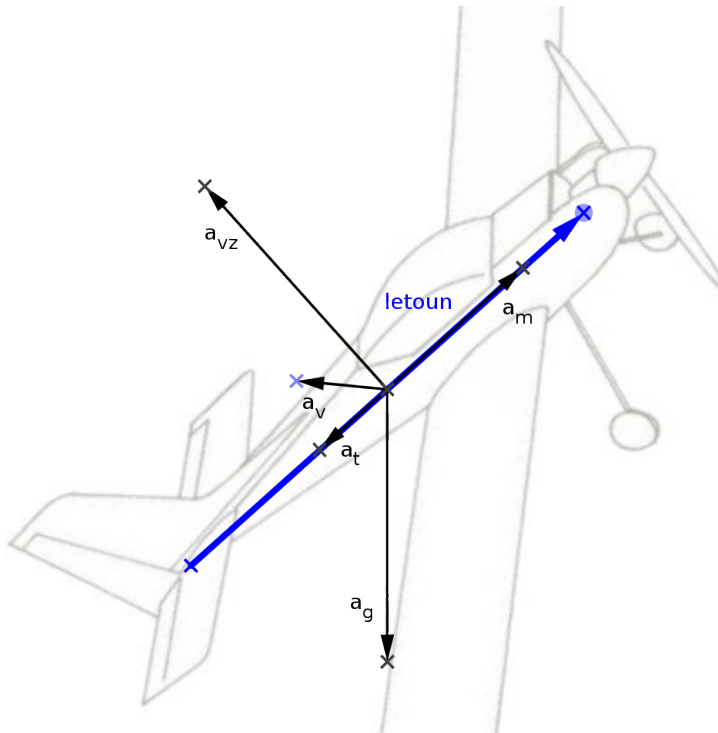
4.1 Měření pohybu letounu ve vzdušném prostoru

Pro výpočet všech hodnot použitých pro řízení letounu na základě akcelerometrů a dynamického gyroskopu zohledňují následující síly působící na letoun. Při výpočtech pracuji s hodnotami zrychlení nikoliv se silami. U jednotlivých veličin záměrně nejsou uvedeny jednotky, neboť jsou závislé na jednotkách, ve kterých měří konkrétní akcelerometr a gyroskop. Veličiny jsou reprezentovány jako 32bitové celé číslo. Další údaje jako rozsah hodnot a pozice střední hodnoty (nuly) se změří během kalibrace.

Kartézský souřadný systém je orientován vzhledem k letounu (k poloze akcelerometru), jak je uvedeno na obrázku 10. Při výpočtech jsou zohledněny následující síly působící na letoun:

1. gravitační síla \vec{a}_g ... známe velikost, kterou změříme pomocí kalibrace před startem, směr a orientace je závislá na rotaci letounu vůči zemi
2. vztlaková síla \vec{a}_{vz} ... známe směr (osa Z), velikost a orientace závisí na rychlosti rotace X

3. tah motoru \vec{a}_m ... známe směr (osa Y), orientace je kladná za předpokladu tahu v kladném směru (vpřed), velikost je přímo úměrná výkonu motoru
4. brždění letounu třením o vzduch \vec{a}_t ... známe směr (osa Y), orientace je záporná za předpokladu letu v kladném směru (vpřed), velikost je přímo úměrná druhé mocnině rychlosti
5. změna proudění vzduchu a turbulence \vec{a}_v ... neznáme žádný údaj, předpokládáme, že změna proudění vzduchu nezpůsobí dlouhodobé zrychlování letadla a součet všech zrychlení způsobených prouděním vzduchu konverguje k nule
6. pozice řídicích klapek S_x, S_y, S_z a výkon motoru S_m ... skalární známé údaje, maximální výchylky a pozice střední hodnoty (nuly) jsou změřeny pomocí kalibrace před startem



Obrázek 11: Znázornění silového působení za letu

4.1.1 Vztahy pro výpočet pohybu letounu

Konstanty G, M, T, V charakterizují konstantní veličiny, které jsou kromě gravitačního zrychlení G závislé na letových vlastnostech daného letounu. V průběhu celého letu používáme následující údaje:

$$\begin{aligned}
 |\vec{a}_g| &= G \\
 \vec{a}_{vz} &= |\vec{a}_{vz}| \cdot (0, 0, 1) \\
 \vec{a}_m &= |\vec{a}_m| \cdot (0, 1, 0) \\
 |\vec{a}_m| &= M \cdot S_m \\
 \vec{a}_t &= -|\vec{a}_t| \cdot (0, 1, 0) \\
 |\vec{a}_t| &= T \cdot v^2 \\
 \lim_{t \rightarrow \infty} \sum_{t_0}^t a_v(t_i) &= 0
 \end{aligned}$$

Neznámou je i rychlost v , kterou můžeme dopočítat na základě změny polohy. Pokud neznáme změnu polohy, lze ji s nízkou přesností určit na základě změny rotace letounu v závislosti na pohybu řídicích klapkek. Čím vyšší rychlost větru, tím silněji letoun reaguje na změnu polohy řídicích klapkek. Vector rot vyjadřuje rychlost rotace celého letounu vzhledem k souřadným osám.

$$\begin{aligned} v &= \int_{t_0}^t a(t) dt \\ v &= \frac{s}{\Delta t} \\ v &\doteq V \cdot \sqrt{\frac{rot_{x,y,z}}{S_{x,y,z}}} \end{aligned}$$

Pokud známe aktuální polohu a rychlost letounu vůči zemi, můžeme na základě dat z akcelerometrů a gyroskopu diskrétně dopočítat polohu v čase $t > t_0$. Odchylna vypočtené hodnoty je přímo úměrná času Δt .

$$\begin{aligned} \vec{v} &= \vec{v}_0 + \vec{a}_a \Delta t \\ \vec{s} &= \vec{s}_0 + \vec{v}_0 \Delta t + \frac{1}{2} \vec{a}_a (\Delta t)^2 \end{aligned}$$

4.1.2 Data získaná akcelerometrem a gyroskopem

Trojosý akcelerometr nám dodává celkem tři diskrétní údaje, zrychlení v ose X, Y a Z. Dynamický gyroskop nám obdobně dodává aktuální diskrétní informaci o rotaci letadla v ose X, Y a Z.

Zrychlení letadla změřené akcelerometrem označme a a celkovou rotaci označme rot . Vektor rot udává aktuální rychlost rotace vzhledem k osám X, Y a Z. Aktuální zrychlení letounu vůči zemi označme a_a . Potom víme, že:

$$\begin{aligned} \vec{a} &= \vec{a}_{vz} + \vec{a}_m + \vec{a}_t + \vec{a}_v \\ \vec{a} - \vec{a}_g &= \vec{a}_a \end{aligned}$$

4.1.3 Data získaná kalibrací před startem

Před startem předpokládáme, že průměrné zrychlení letounu \vec{a}_p je rovno gravitačnímu zrychlení. Proměnné $start$ a end reprezentují čas počátku a konce kalibrace letadla před vzletem.

$$|\vec{a}_p| = \frac{\sum_{i=start}^{end} |\vec{a}_i|}{end - start} = |\vec{a}_g|$$

K dispozici máme dynamický gyroskop, který určuje rychlost rotace celého letounu. Na zemi můžeme změřit aktuální rotaci rot_0 letounu vůči zemi. Kalibrovanou hodnotu si uložíme jako rychlost nutnou k otočení letounu o úhel 2π za jednotkový čas.

Před vzletem je nutné provést kontrolu všech řídicích prvků. Jejich následná nefunkčnost by způsobila havárii celého letadla. Předpokládáme, že před vzletem jsou všechny řídicí klapky (kromě výkonu motoru) vychýleny do maximálních poloh a zpět. Zapamatujeme si všechny maximální výchylky pro pozdější řízení.

$servo_{max}$... maximální výchylky všech řídicích klapek

$servo_{min}$... minimální výchylky všech řídicích klapek

Výkon motoru je řízen vlastním regulátorem, který před startem obdrží signál maximálního a nulového výkonu. (Tuto funkci lze uživatelsky vypnout.) Další hodnoty získáme lineárním proložením maximálního a nulového výkonu.

4.1.4 Data získaná během prvního letu

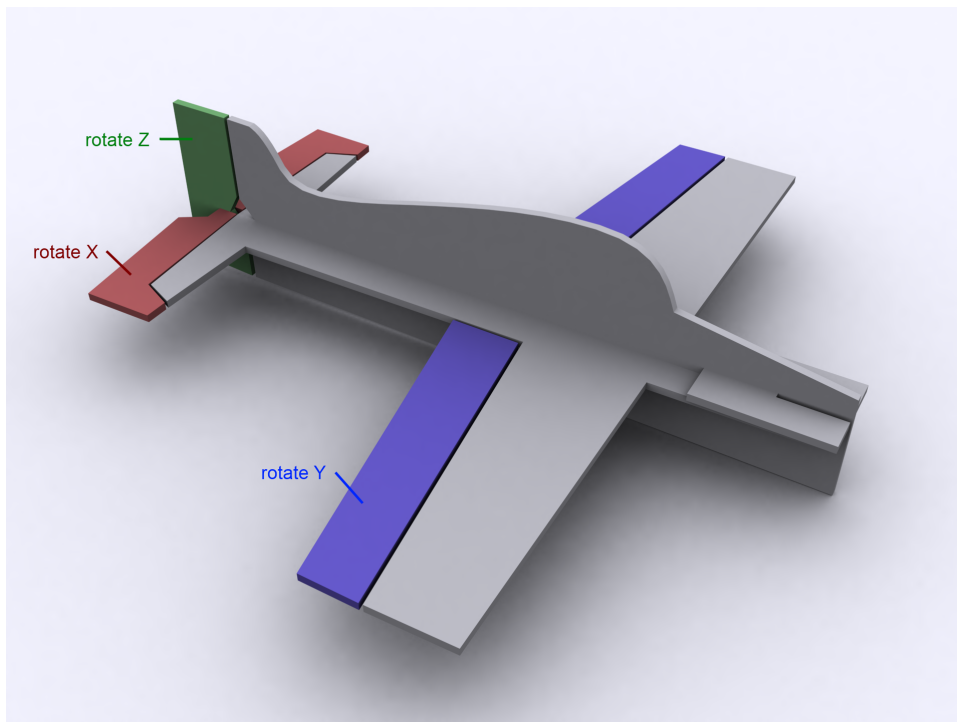
První let je řízen jiným systémem (pilotem, GPS navigací nebo jinými programy), program čte všechny změřené údaje a dopočítá konstanty nutné pro další řízení. Po přepnutí do režimu navigace jsou data neustále dokalibrovávána pro zvýšení přesnosti měřených údajů.

Konstantu M vypočteme jako změnu zrychlení v závislosti na změně výkonu motorů.

$$M = \frac{\Delta a_y}{\Delta S_m}$$

Pokud letoun nemění svou výšku, změříme závislost změny třetího zrychlení na výkonu motoru. Víme, že při neměnné výšce je $a_y = a_m - a_t$. Známe výkon motoru a zrychlení a_y , z toho dopočítáme a_t a dokalibrujeme aktuální rychlost letounu jako $v = X \cdot \sqrt{a_t}$. Konstantu X můžeme zanedbat, neboť můžeme zvolit jednotku rychlosti.

Pokud je aktuální zrychlení letounu $a_a \doteq 0$, můžeme předpokládat, že pohyb letounu je rovnoměrný přímočarý. Díky této vlastnosti víme, že $a = a_g$, a dopočítáme směr k zemi. Nyní známe rotaci letounu vůči zemi a dokalibrujeme údaje o absolutní rotaci letounu a rychlosti. Pokud je $a_a = 2a_g$, letoun zrychluje směrem k zemi. Nestačí tedy porovnávat velikost zrychlení naměřeného akcelerometrem $|\vec{a}| = |\vec{a}_g|$, ale je nutné zohlednit i orientaci \vec{a} .



Obrázek 12: Řízení směru letu pomocí naklápění

4.2 Řízení letounu na základě dat z akcelerometru a dynamického gyroskopu

Díky výše uvedeným speciálním případům můžeme i během samotného letu dokalibrovávat a zpřesňovat údaje o rotaci letounu vůči zemi. Vzhledem k tomu, že z vnějších údajů máme k dispozici pouze gravitační zrychlení, které je v každém místě stejné, nemůžeme během letu dynamicky dokalibrovávat údaje o absolutní poloze letounu v prostoru. V čase $t > t_0$ přímo úměrně roste odchylka vypočtené absolutní rotace letounu v prostoru.

Bez dalších sensorů můžeme dlouhodobě udržovat stálý směr letu nebo zatáčet či kroužit. Řízení akcelerometrem a gyroskopem můžeme použít jako záložní řídicí systém v případě ztráty dalších údajů. Avšak letoun není možné dlouhodobě přesně řídit po zadané trajektorii pouze na základě dat z akcelerometru a dynamického gyroskopu.

Při řízení letounu dodáváme informace o aktuálním a chtěném pohybu letounu. Samotná regulace řídicích klapek je obsluhována PID regulátorem (proportional-integral-derivative controller) [15]. Na vstup je předávána informace o aktuální a chtěné rotaci letounu vůči zemi. Další řízení obstarává PID regulátor.

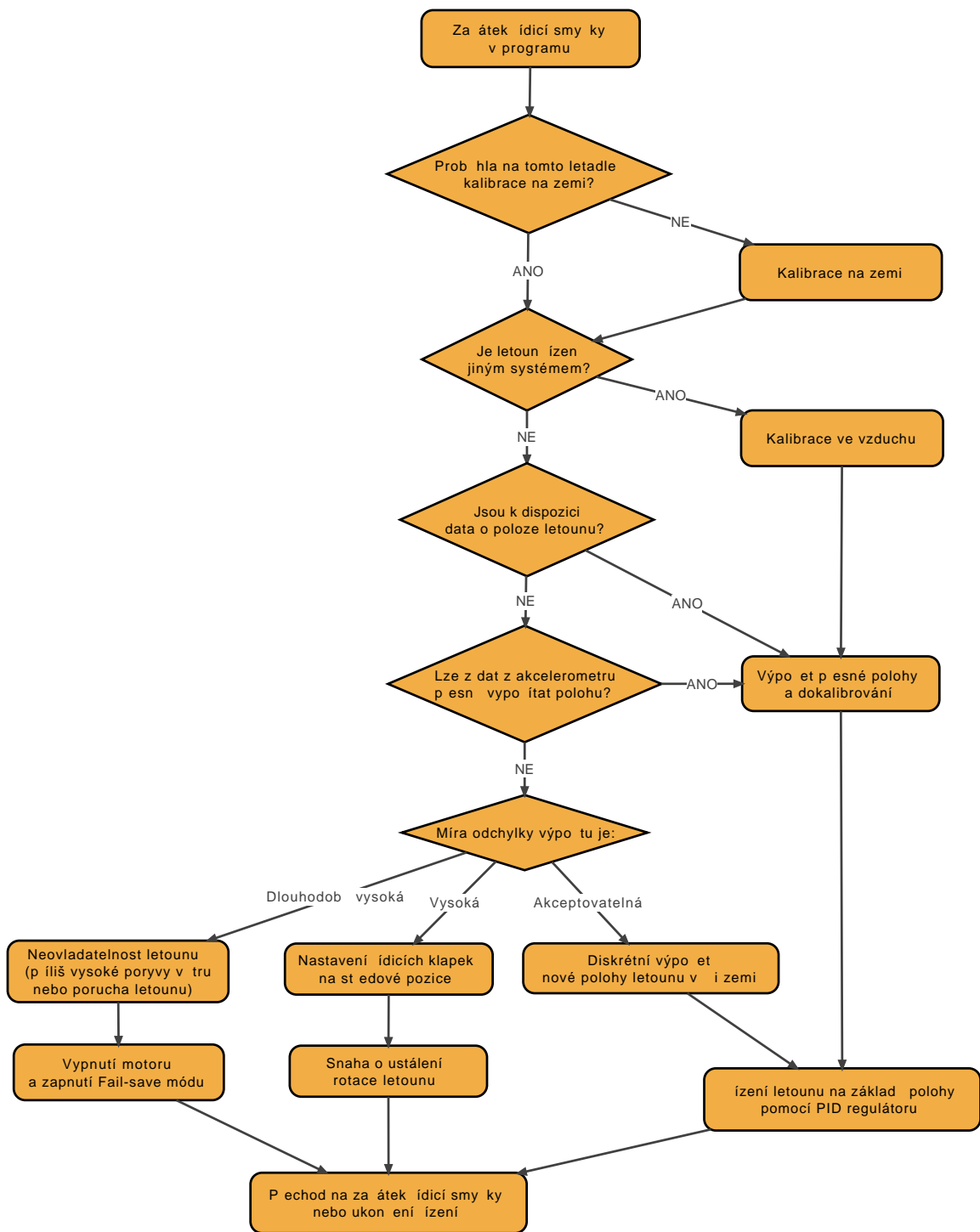
Pokud chceme, aby letoun pokračoval v daném směru v přímém letu, nastavíme nulovou chtěnou horizontální rotaci a jako vertikální rotaci míru stoupání či klesání. Této možnosti řízení využijeme například při ztrátě GPS signálu nebo při průletu nad územím bez možnosti získání dalších dat. Gravitace je veličina měřitelná ve všech místech, díky tomu se nemusíme obávat ztráty i této informace a letoun bude vždy možné řídit.

Kroužení nad daným územím využijeme například při dočasné ztrátě GPS signálu či dat z jiných sensorů. Po získání dostatku informací předáme řízení opět do rukou GPS navigace či jiných systémů.

Schéma programu pro řízení letounu je znázorněno na obrázku 13. Po prvním spuštění je nutné provést kalibraci na zemi, která je popsána v sekci 4.1.3. Pokud je k dispozici řízení jiným systémem (uživatel, GPS navigací, ...), program pouze naslouchá všem sensorům a dopočítává svou polohu včetně dokalibrovávání konstant. Dokalibrovávání dat ve vzduchu je uvedeno v sekci 4.1.4. Pokud je k dispozici poloha a rotace letounu naměřená jiným systémem (například GPS), program postupuje obdobně jako při dokalibrovávání dat ve vzduchu. Na základě dat o poloze a rotaci je letoun řízen pomocí PID regulace.

Pokud program nemá externě k dispozici údaje o poloze a rotaci, vypočítá rotaci letounu diskrétně na základě dat z dynamického gyroskopu, rychlost a polohu na základě údajů z akcelerometrů. Teoretické vztahy pro výpočet jsou uvedeny v sekci 4.1.1. Odchylka tohoto výpočtu je bohužel přímo úměrná času poslední kalibrace. Tuto nepříjemnost můžeme odstranit hledáním stavů, ve kterých můžeme let považovat za rovnoměrný přímočarý. V takovém případě je rotace letounu vůči zemi dána směrem zrychlení, neboť celkové zrychlení letounu je nulové a akcelerometry měří směr a velikost vztlakové síly. Je možné, že existuje více speciálních případů, ve kterých lze vypočítat přesnou polohu nebo rotaci letounu vůči zemi. Tato práce je však zatím nezohledňuje.

Pokud delší dobu není možné dokalibrovat rotaci letounu vůči zemi, jsou výsledky výpočtu absolutní rotace velmi nepřesné. Program se snaží srovnání letounu do nulové rychlosti rotace nastavením regulačních klapek do středních poloh. S konstantním výkonem motoru přejde letoun do rovnoměrného přímočarého pohybu, který je ovlivněn pouze poryvy větru. Pokud jsou poryvy větru silnější než manévrovací schopnosti letadla, letoun se stává neovladatelným. V takovém případě je řídicí program vypnut a letoun přechází do fail-save módu. Fail-save mód lze předefinovat uživatelským programem. V programu je fail-save mód definován nastavením regulačních klapek do středových poloh a vypnutím motoru pro snížení rozsahu poškození letounu v případě kolize se zemí.



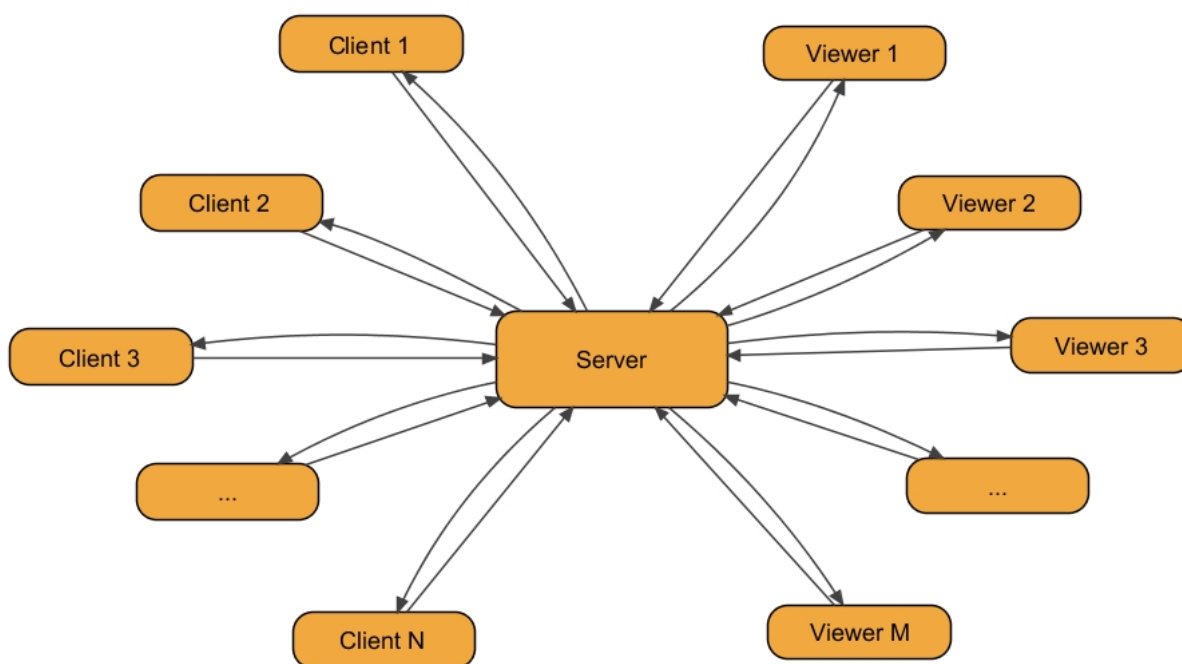
Obrázek 13: Schéma programu pro řízení letounu pomocí akcelerometrů

5 Simulátor skupiny robotů

Simulátor Yunimin [22] je client-server aplikace určená pro simulaci skupiny robotů. Robotem je myšlen mobilní stroj pohybující se po zemi pomocí kol nebo ve vzduchu formou vzducholodě či definovaného letadla. Pro účely vývoje byl použit model pozemního robota Yunimin. Letová verze simulace je v současné době testována.

Celá aplikace se skládá ze serveru, ke kterému se připojují klienti a viewery. Klient je konsolová aplikace, která simuluje program nahraný do robota. Viewer je uživatelská aplikace, která uživateli zobrazuje průběh simulace a dává mu možnost celou simulaci řídit.

Server pracuje jako simulační prostředí, ke kterému se může připojit libovolné množství robotů nebo klientů, kteří simulují svůj chod v reálném čase. Dále se k serveru může připojit libovolný počet viewerů, kteří od serveru získávají veškeré informace o dění ve virtuálním světě a umožňují provádět veškeré uživatelské zásahy do simulace.



Obrázek 14: Komunikace mezi jednotlivými částmi simulátoru

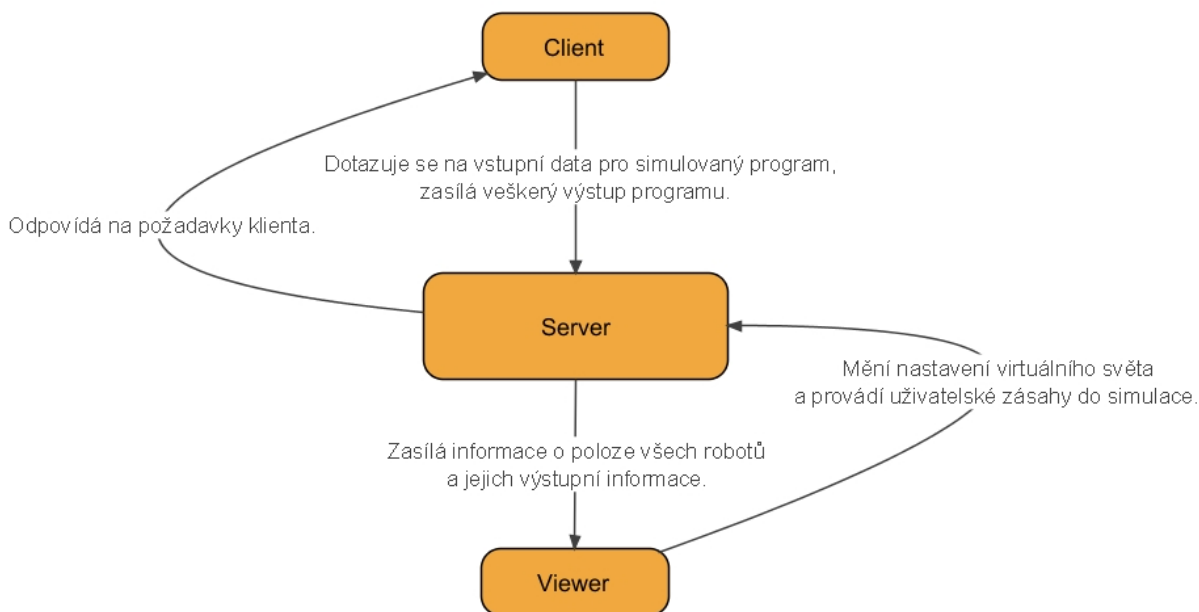
Lze zároveň používat různé verze klientů a viewerů. Aplikace je navržena tak, aby jednotlivé části aplikace musely splňovat pouze kritéria komunikačního protokolu. Je možné vytvořit více různých klientů a viewerů pracujících na různých platformách. Díky tomuto návrhu se celá aplikace stává multiplatformní.

5.1 Popis a funkce simulátoru

5.1.1 Server

Server primárně přijímá spojení od klientů a viewerů a vyřizuje jejich požadavky. Spravuje centrální uložení všech informací o všech robotech zapojených do simulace. Všechny změny zasílá připojeným viewerům. Díky tomuto uložení je server schopen odpovídat na dotazy klientů, kteří potřebují vstupní data k odvedení korektní simulace.

Výpočet polohy a dalších dat ovlivňovaných činností robotů: Výpočetní jednotka serveru počítá údaje dané vnějším prostředím. Ze získaných výkonů motorů počítá aktuální



Obrázek 15: Specifikace komunikace se serverem

rychlost a polohu robota v prostoru. Na základě údajů o poloze a pohybu dále počítá hodnoty snímané jednotlivými sensory v závislosti na dodaném obrazovém podkladu. Výpočetní jednotka dále zajišťuje chod všech přídatných zařízení včetně vzájemné komunikace mezi roboty a komunikace robotů s reálným světem.

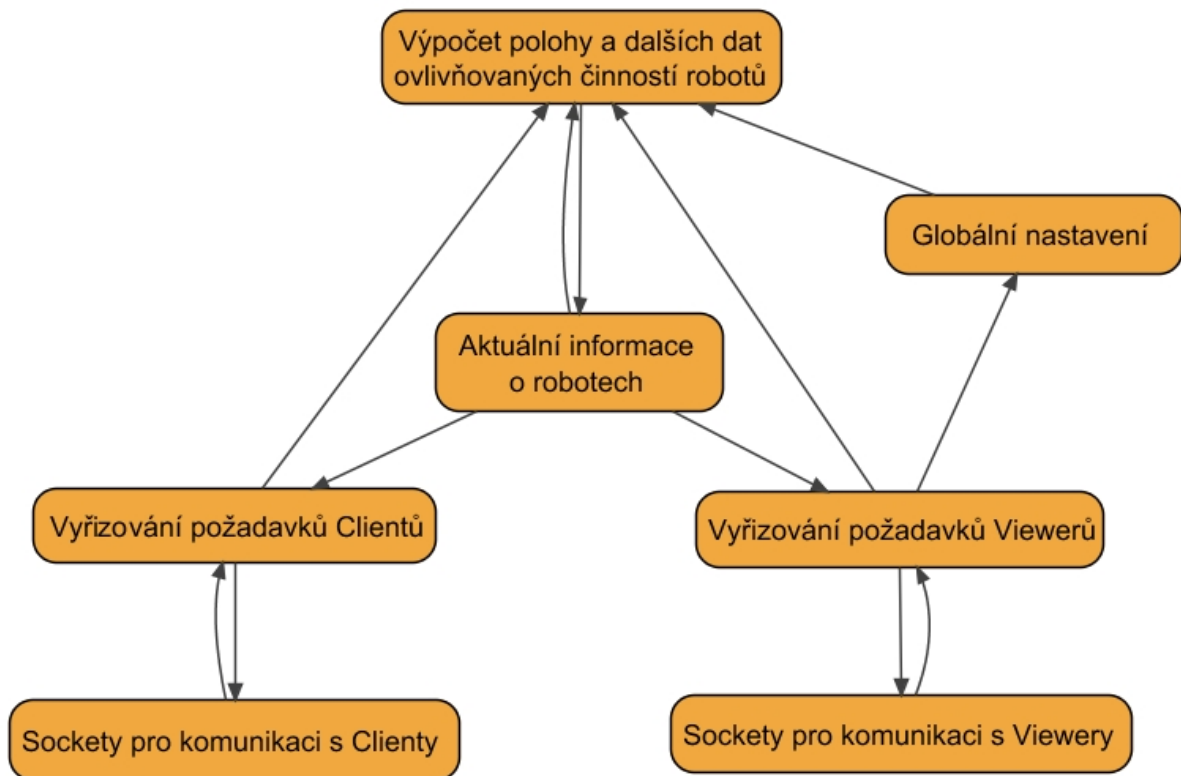
V aplikaci je výpočetní jednotka reprezentována samostatným vláknem.

Globální nastavení: Uchovává informace o virtuálním světě a nastavení provedená uživatelem pomocí vieweru. Obsahuje obrazové podklady pro sensory závislé na poloze robota. Pro každou polohu je vyčleněn bod s nastavenými hodnotami sensorů. Veškeré nastavení, které se netýká konkrétního robota, lze importovat z konfiguračního textového souboru. Obrazové podklady očekává ve formátu PNG nebo BMP. Nastavení pro jednotlivé roboty se dodává opět v konfiguračním textovém souboru, jehož jméno je shodné se jménem robota nebo skupiny robotů.

Vyřizování požadavků clientů a viewerů: Tato vlákna naslouchají na daných portech a vyřizují veškerou komunikaci serveru s ostatními částmi aplikace. Příchozí informace zapisují do centrální paměti, dávají podněty výpočetní jednotce a odpovídají na požadavky clientů. Pro každého připojeného klienta nebo viewera server vytvoří nové vlákno, na kterém probíhá samotná komunikace. Vytvořené vlákno je ukončeno v okamžiku odpojení klienta nebo viewera od serveru.

5.1.2 Client

Každý client běží pouze po dobu účasti konkrétního robota v simulaci. Odstartování a připojení klienta k serveru znamená odstartování simulace programu v konkrétním robotovi. Po připojení k serveru se client snaží identifikovat pod nastavitelným jménem. Pokud není do simulace zapojen stejnojmenný robot, je tomuto clientovi zaslán souhlas ke startu programu. V opačném případě se robot pokouší připojit pod dalšími různými jmény nebo čísly, dokud není některé z nabízených jmen akceptováno serverem. Před startem programu zasílá client informace o počáteční poloze robota, které je možné nastavit i pomocí vieweru. Viewer má dále možnost



Obrázek 16: Architektura a funkce serveru

s robotem provést libovolný ruční pohyb proti jeho vlastní vůli. Viewer má tedy možnost upravit všechny údaje o robotovi kromě výstupních dat programu a jména připojeného robota/clienta. Client nemůže měnit své jméno v průběhu simulace z důvodu jednoznačné identifikace robotů v simulovaném prostředí.

Po přijetí souhlasu ke startu od serveru se rozběhne vlastní simulace programu se zasíláním požadavků o vstupní data. Client vždy čeká na odpověď serveru, teprve po přijetí odpovědi zasílá další dotaz. Takto proběhne celý program až po jeho ukončení. Pokud je simulovaný program zacyklen, provádí se až do ručního odpojení robota pomocí vieweru nebo do ukončení celé simulace.

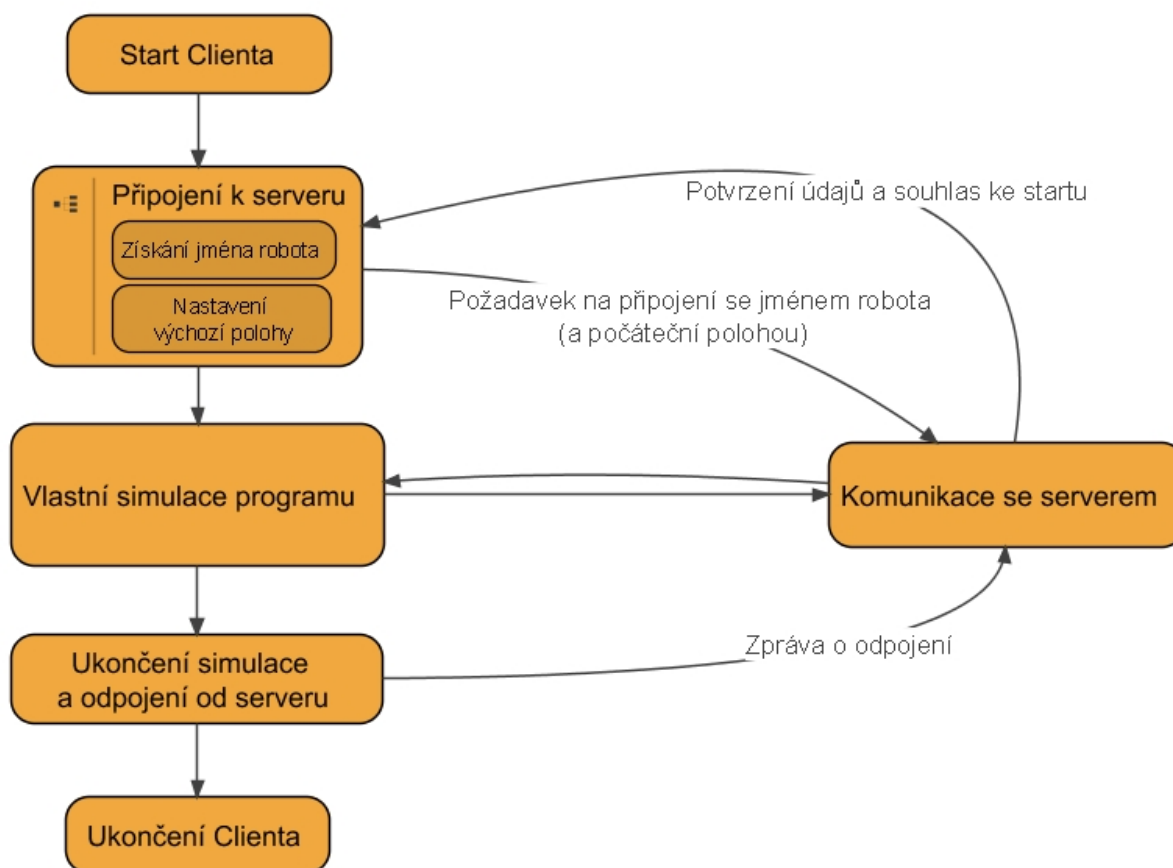
Pokud clientský program skončí, client zašle na závěr zprávu o jeho ukončení, počká na potvrzení přijetí této zprávy serverem, a poté se ukončí.

5.1.3 Viewer

Viewer zajišťuje kompletní spojení mezi uživatelem a celým systémem. Díky této výhodě je možné sledovat a řídit běžící simulaci přes síť bez ručního zásahu do počítačů, které tuto simulaci počítají. Je také možné pomocí více viewerů z různých míst připojovat další roboty a clientské programy spouštět na vzdálených počítačích. Je tedy možné provést simulaci libovolného množství robotů bez závislosti na poloze zapojených počítačů.

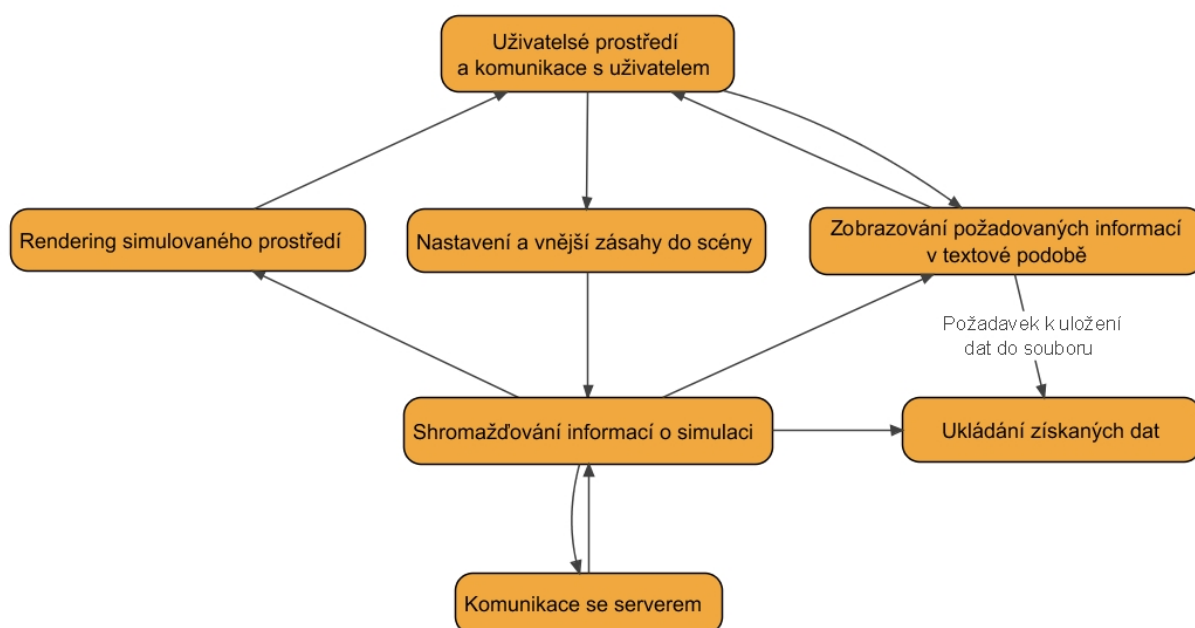
Viewer přijímá od serveru všechna data o probíhající simulaci. Jeho úkolem je tato data zobrazovat uživateli a případně je ukládat na uživatelem zvolená úložiště. Poloha robotů je reprezentována na 2D ploše dvojrozměrnými útvary nebo 3D náhledem renderovaným pomocí Ogre engine [16].

Uživatel má možnost pomocí vieweru přidávat do simulace nové roboty, měnit polohu robotů, zastavovat nebo spouštět programy v zúčastněných robotech, zasílat simulovaným

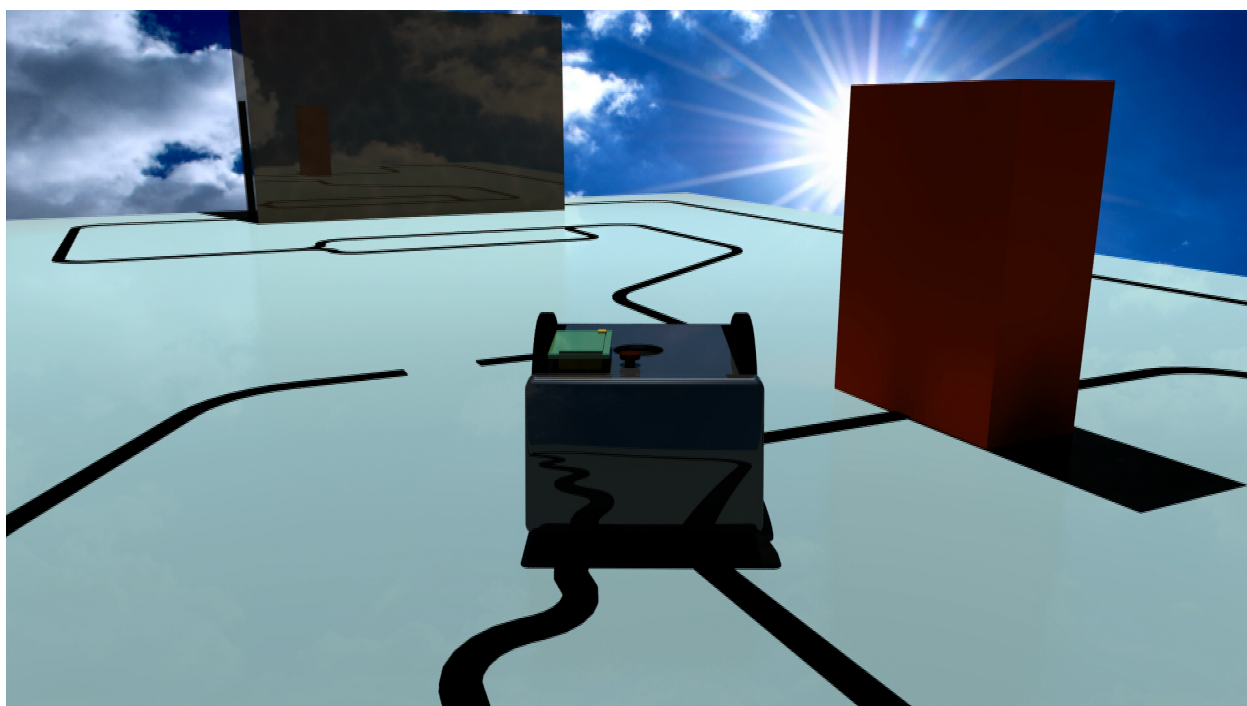


Obrázek 17: Chování klienta od startu po ukončení

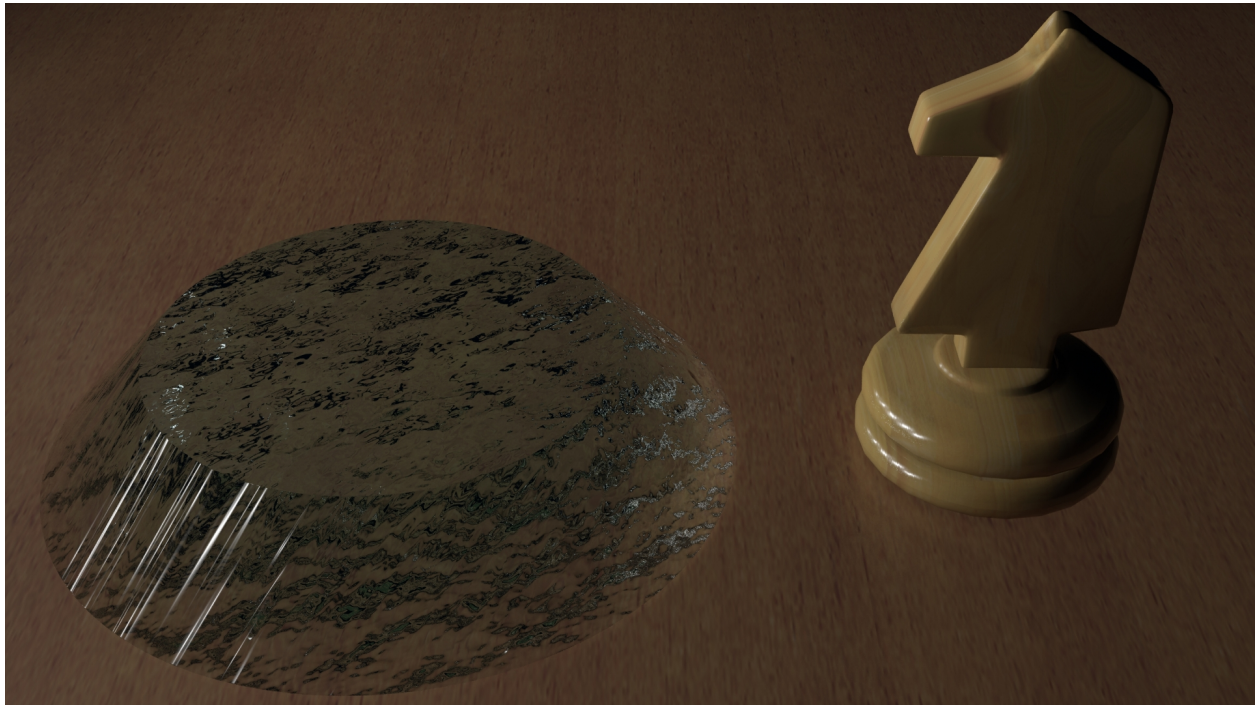
robotům zprávy, připojovat do simulace komunikaci s reálnými roboty a odečítat všechny vypočtené hodnoty v reálném čase. Celou aplikaci lze ovládat pouze za použití vieweru, který vyřídí všechny uživatelské požadavky přes síťovou komunikaci.



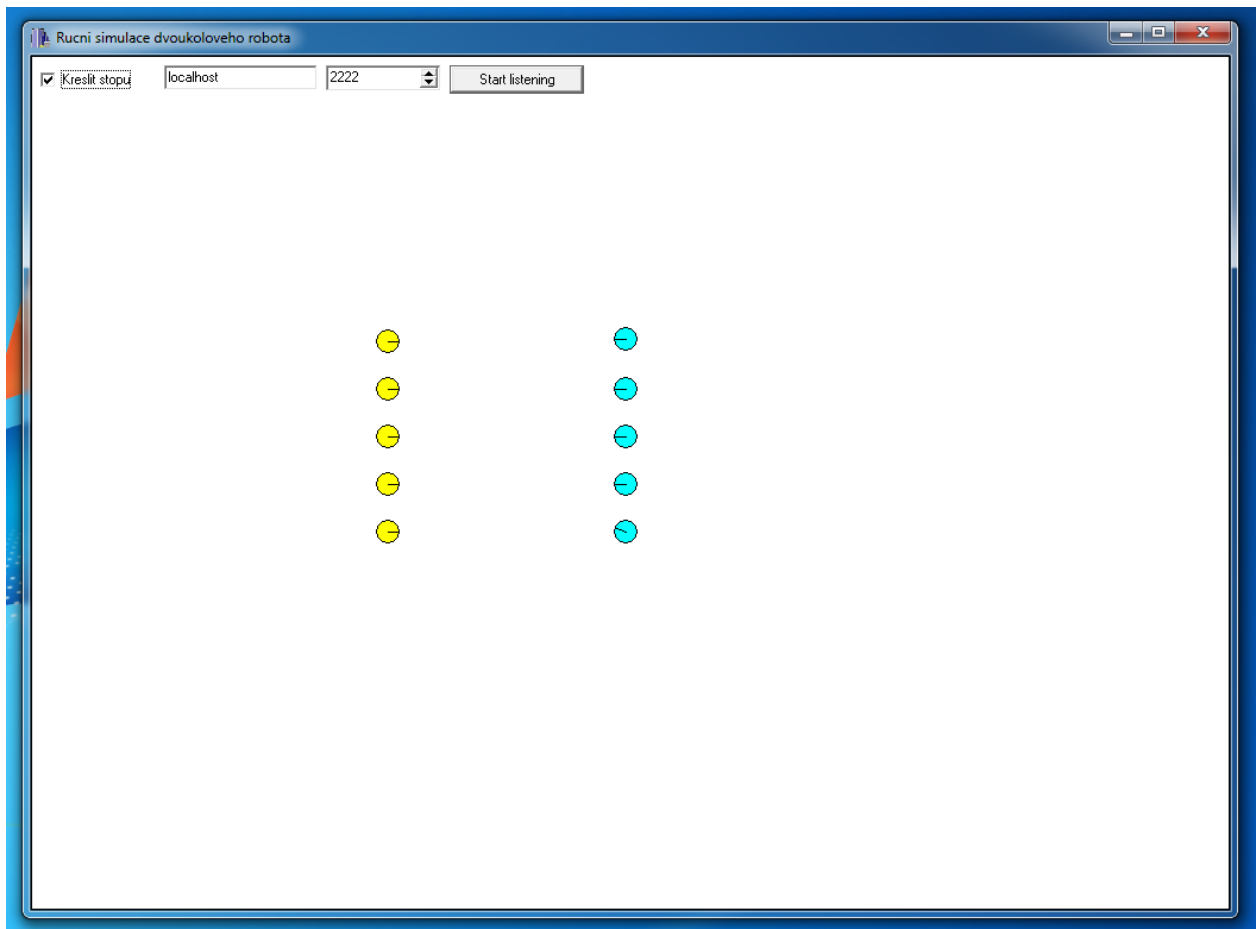
Obrázek 18: Architektura a funkce vieweru



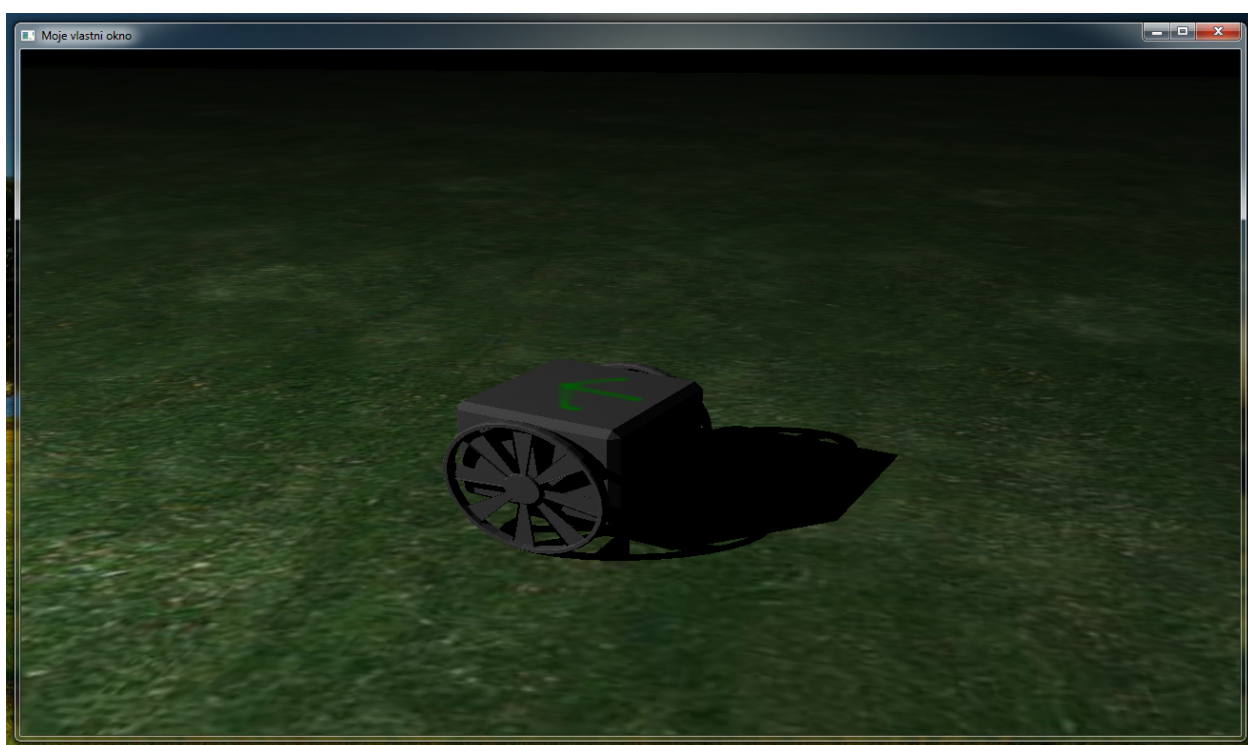
Obrázek 19: Export simulace jako 3D animace - sledování čáry



Obrázek 20: Export simulace jako 3D animace - hledání kapky vody



Obrázek 21: Ukázka simulace 2D viewerem v reálném čase



Obrázek 22: Ukázka simulace 3D viewerem v reálném čase

6 Shrnutí

Vyvinuté programy byly spouštěny na prototypu řídicí elektroniky, která byla postupně testována na RC modelu lodi, akrobatického letadla a samokřídla. Řízení letounu akcelerometry a dynamickým gyroskopem umožňuje udržet přímý let nebo kroužení nad danou oblastí. Při řízení akcelerometry musíme počítat s unášením letounu větrem. Navigaci pomocí systému GPS poskytuje open-source projekt ArduPilot Mega, jehož navigační program je kompatibilní s elektronikou Yunifly. Mobilní telefon spojuje bluetooth komunikaci řídicí elektroniky s internetovým připojením k uživateli. Řídicí elektronika nebo uživatel může mobilnímu telefonu zaslat příkaz nebo dotaz po sériové lince. Simulátor skupiny robotů nyní simuluje pozemní roboty řízené pomocí kol a roboty s řízením podobným vzducholodi. Simulace letounu a helikoptéry je stále ve vývoji z důvodu složitějšího fyzikálního popisu.

System Yunifly můžeme použít například pro měření sensorických údajů ve vzdušném prostoru nebo jako nástroj pro fotografování zemského povrchu. Schopností letounu plánují otestovat fotografováním Hádeckého kamenolomu v Brně s následným vygenerováním 3D modelu z obrazových dat. Kvalitu řízení letounu pomocí akcelerometrů a dynamického gyroskopu můžeme otestovat například v nepříznivých povětrnostních podmínkách nebo při akrobatickém letu. Řízení akrobatického letu vyžaduje znalost směru gravitačního zrychlení, které souvisí s údaji naměřenými akcelerometrem.

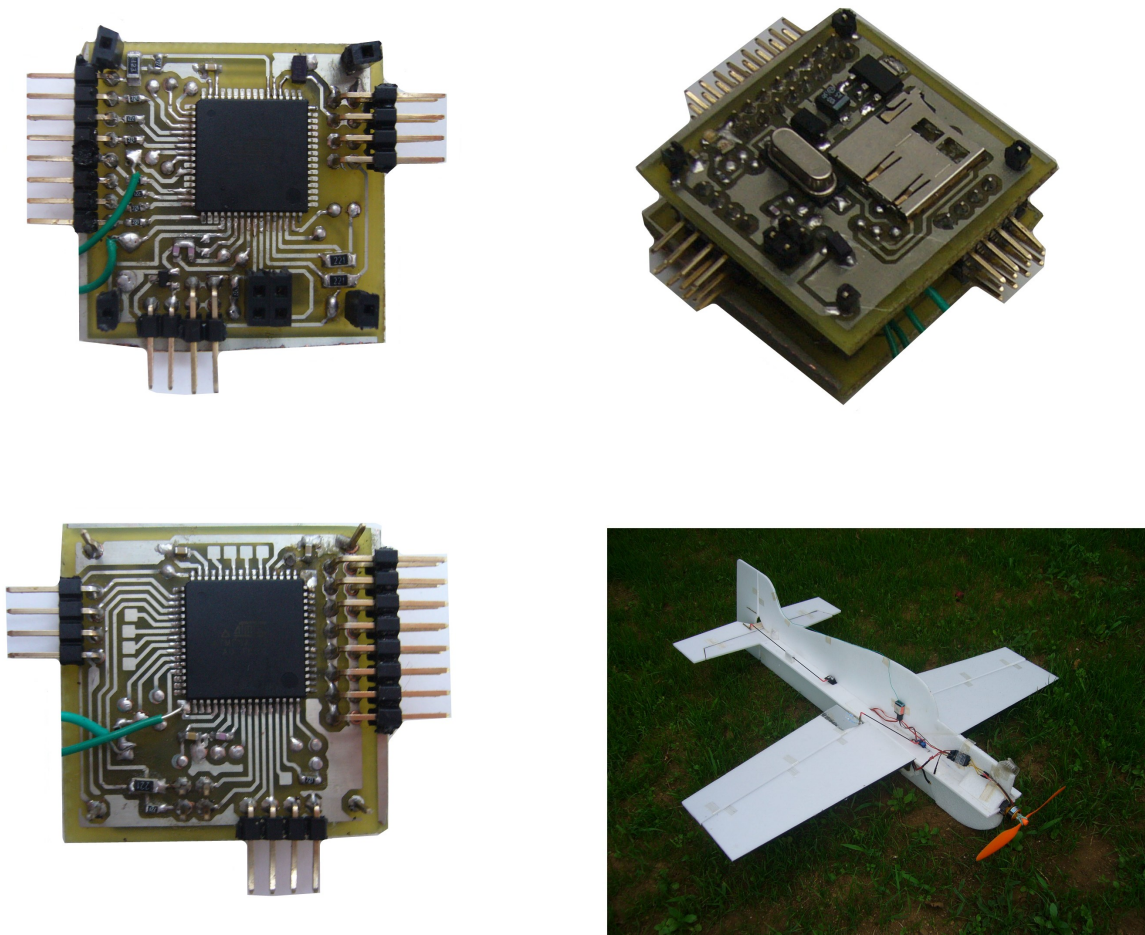
Reference

- [1] Arduino [online]. 2010 [cit. 2011-03-01]. Home page. Dostupné z WWW: <http://www.arduino.cc/>
- [2] ANDERSON, Chris, et al. ArduPilot : Arduino-based autopilot with GPS [online]. 2010 [cit. 2011-12-11]. Dostupné z WWW: <http://code.google.com/p/ardupilot/>
- [3] ArduPlane : ArduPilot-mega - official ArduPlane repository [online]. 2010, 2011-12-11 [cit. 2011-12-11]. Dostupné z WWW: <http://code.google.com/p/ardupilot-mega/>
- [4] Atmel.com [online]. 02.2011 [cit. 2011-03-01]. 8-bit Microcontroller with 128KBytes In-System Programmable Flash ATmega128, ATmega128L. Dostupné z WWW: <http://www.atmel.com/dyn/resources/prod-documents/doc2467.pdf>
- [5] GM electronic [online]. B07. 2010 [cit. 2011-03-01]. TS1117 datasheet. Dostupné z WWW: <http://www.gme.cz/dokumentace/dokumenty/934/934-140/dsh.934-140.1.pdf>
- [6] Rctoys.com [online]. 2010 [cit. 2011-03-01]. Hitec 8.0g HS-55 Feather Standard Micro RC Servo. Dostupné z WWW: <http://www.rctoys.com/rc-toys-and-parts/HIT-HS55/RC-PARTS-HITEC-SERVOS.html>
- [7] Embeddedartists.com [online]. 1.0. 2010 [cit. 2011-03-01]. GPS smart antenna module, LS20032. Dostupné z WWW: <http://www.embeddedartists.com/projects/lpcpresso-ls20031/LS20030-3-datasheet-v1.0.pdf>
- [8] Pololu.com [online]. 1.1. 2010 [cit. 2011-03-01]. CHR-6d Digital Inertial Measurement Unit. Dostupné z WWW: <http://www.pololu.com/file/0J276/chr6d-datasheet.pdf>
- [9] Krause-robotik.de [online]. 15.10.2007 [cit. 2011-03-01]. CMPS03 - Robot Compass Module. Dostupné z WWW: <http://www.krause-robotik.de/service/download/doku/cmeps3docRel14.pdf>
- [10] Spezial.com [online]. 1.4. 2008 [cit. 2011-03-01]. OEM Serial Port Adapter™. Dostupné z WWW: <http://www.spezial.com/doc/conblu/em-ds-oemspa-310.pdf>
- [11] Alldatasheet.com [online]. 2010 [cit. 2011-03-01]. MicroSDHC Card series datasheet. Dostupné z WWW: <http://www.alldatasheet.com/datasheet-pdf/pdf/330195/TRANSCEND/MICROSDHC.html>
- [12] DharmaniTech.com [online]. 2009 [cit. 2011-03-01]. SD/SDHC Card Interfacing with ATmega8/32 (FAT32 implementation). Dostupné z WWW: <http://www.dharmanitech.com/2009/01/sd-card-interfacing-with-atmega8-fat32.html>
- [13] Wikipedia.org [online]. 23.1.2011 [cit. 2011-03-01]. I2C. Dostupné z WWW: <http://cs.wikipedia.org/wiki/I2C>
- [14] Wikipedia.org [online]. 25.8.2010 [cit. 2011-03-01]. USART. Dostupné z WWW: <http://cs.wikipedia.org/wiki/USART>
- [15] Wikipedia.org [online]. 2011 [cit. 2011-03-01]. PID regulátor. Dostupné z WWW: http://cs.wikipedia.org/wiki/PID_regulátor
- [16] OGRE : Open Source 3D Graphic Engine [online]. 2011 [cit. 2011-11-1]. Dostupné z WWW: <http://www.ogre3d.org/>

- [17] Python Programming Language : Official Website [online]. 2011 [cit. 2011-11-1]. Dostupné z WWW: <http://www.python.org/>
- [18] Symbian Portal : Python for S60 [online]. 2011 [cit. 2011-11-1]. Dostupné z WWW: <http://www.symbianportal.cz/aplikace/83-python-for-s60/>
- [19] Python : Python for mobile devices [online]. 2011 [cit. 2011-11-1]. Dostupné z WWW: <http://python.objectis.net/python-links-for-mobile-devices>
- [20] Skype : Aplikace Skype [online]. 2011 [cit. 2011-11-1]. Dostupné z WWW: <http://www.skype.com/>
- [21] Fring : Fring Video Chat [online]. 2011 [cit. 2011-11-1]. Dostupné z WWW: <http://www.fring.com/>
- [22] SAID, Bedřich. Simulátor Yunimin - virtuální svět pro simulaci robotů. Brno, 2010. 30 s. SOČ. DDM Junior Brno. Dostupné z WWW: <http://bitbucket.org/bsaid>
- [23] DDM Junior : Repozitáře technika.junior.cz [online]. 2011 [cit. 2011-11-1]. Dostupné z WWW: <https://technika.junior.cz/trac>

A Podrobná specifikace hardwaru Yunifly

Zde je uvedena specifikace celé řídicí elektroniky Yunifly. Na obrázku 23 jsou uvedeny fotografie elektroniky v rozloženém stavu (vlevo) a ve složeném stavu s příkladem letounu, který tuto desku používá (vpravo).



Obrázek 23: Řídicí elektronika v rozloženém a složeném stavu

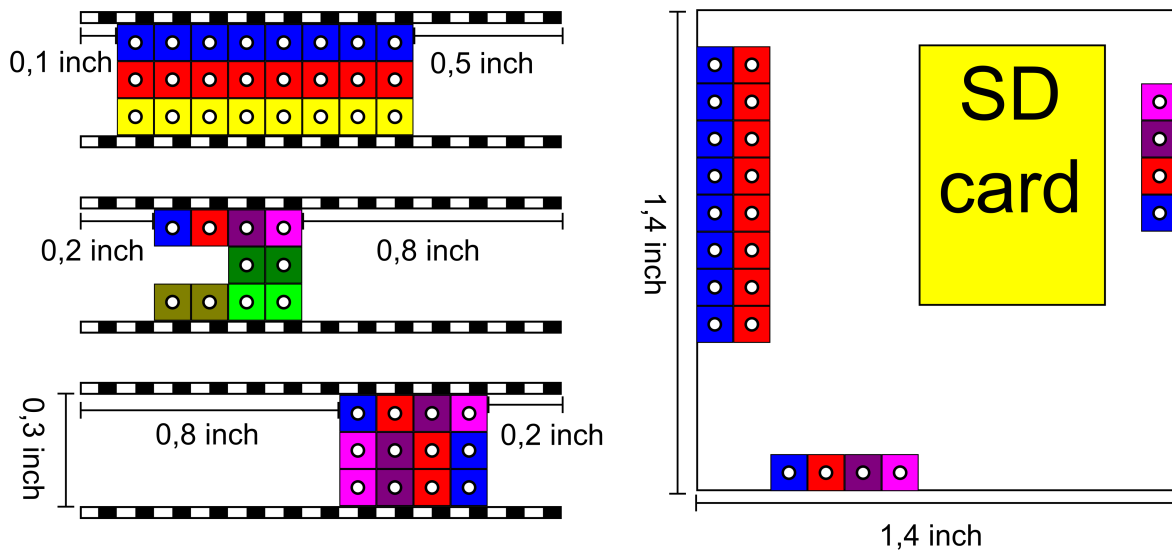
A.1 Hmotnost a rozměry

Elektronika Yunifly je složena ze dvou samostatných částí, které můžeme použít jako jeden dvouprocesorový celek. V tabulce jsou uvedeny hmotnosti jednotlivých částí elektroniky.

| Hmotnost elektroniky Yunifly | |
|------------------------------|--------------|
| Název součásti | Hmotnost [g] |
| Deska Basic | 9 g |
| Deska Main | 9 g |
| Yunifly celá | 18 g |

Obě desky jsou navrženy tak, aby jejich vzájemné propojení bylo co nejvíce kompaktní. Desku Main můžeme nasadit na konektor desky Basic a získáme tak dvě propojené desky stejných rozměrů, které se dohromady uzavírají do kvádrů (viz obrázek 25). Jsou tak odolnější vůči vnějšímu mechanickému poškození.

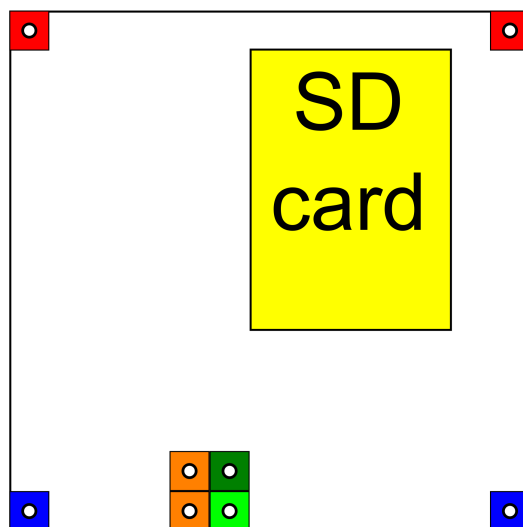
Rozměry celé elektroniky nalezneme na obrázku 24.



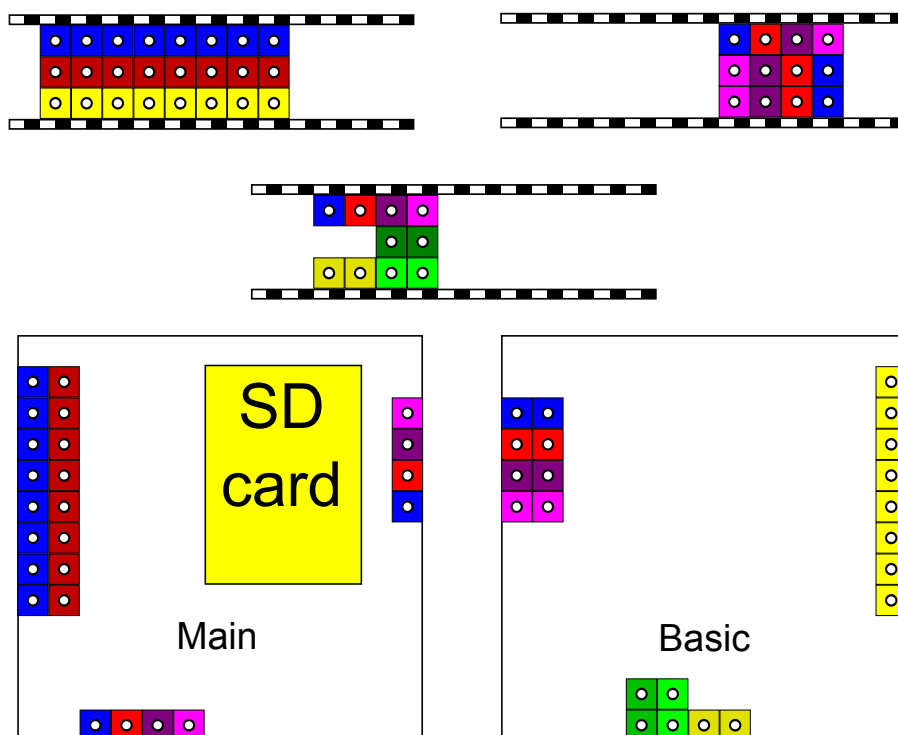
Obrázek 24: Rozměry celé elektroniky Yunify

A.2 Pinová konfigurace

Na obrázku 25 a 26 je znázorněna poloha jednotlivých konektorů na obou deskách. Desky jsou znázorněny v perspektivě a dále jsou zobrazeny tři boční strany s jednotlivými konektory.



Obrázek 25: Schéma konektoru, kterým je spojena deska Main s deskou Basic



- VCC
- BATT
- GND
- INPWM
- SDA
- PWM
- SCL
- RXD
- CONTROL
- TXD

Obrázek 26: Pinová konfigurace elektroniky

| Propojovací konektor | |
|-----------------------------|---|
| Pin | Popis |
| VCC | Napájení 3,3 V, které je rozvedeno po celé desce od stabilizátoru. Konektor je možné využít pro napájení dalších zařízení do 500 mA. |
| GND | 0 V |
| SDA, SCL | I ² Clinka, která je využita ke komunikaci mezi deskou Basic a deskou Main. |
| CONTROL_AID | Pin, kterým je propojena deska Main s deskou Basic. Pokud nastane selhání desky Main, nabyde tento pin hodnoty LOW a deska Basic je přepnuta do nouzového režimu. |
| CONTROL_IO | Pin, kterým je propojena deska Main s deskou Basic. Tímto pinem předáváme desce Basic informaci o jednom ze dvou stavů, jejichž výběr nastavíme programově. |
| RESET | Reset procesoru dané desky. |

| Deska Basic | |
|---------------------------|--|
| Pin | Popis |
| BATT | Napájení desky 5-12 V a detekce stavu vybití baterií. |
| VCC | Napájení 3,3 V, které je rozvedeno po celé desce od stabilizátoru. Konektor je možné využít pro napájení dalších zařízení do 500 mA. |
| GND | 0 V |
| PWM0..7 | Konektory pulsní šířkové modulace (PWM), ke kterým je možné připojit serva a jiné regulátory. |
| ACC_RXD, ACC_TXD | Konektor sériové linky, primárně určený k připojení akcelerometrů. (Lze připojit libovolné zařízení s podporou USART.) |
| FREERX_RXD, FREERX_TXD | Konektor sériové linky. Lze připojit libovolné USART zařízení, například bluetooth modul. |
| KOMPAS_SDA, KOMPAS_SCL | Konektor I ² Csběrnice. Primárně určený pro připojení kompasu. |
| OTHER_SDA, OTHER_SCL | Druhý paralelní konektor I ² Csběrnice. |
| INPWM1..2 | Dva nezávislé konektory pro zpracování multi PWM signálu od RC přijímače nebo jiného zařízení totožného protokolu. |
| START | Pad, ke kterému lze připojit programově detekovatelné tlačítko. |

| Deska Main | |
|---------------------|--|
| Pin | Popis |
| BATT | Napájení desky 5-12 V a detekce stavu vybití baterií. |
| VCC | Napájení 3,3 V, které je rozvedeno po celé desce od stabilizátoru. Konektor je možné využít pro napájení dalších zařízení do 500 mA. |
| GND | 0 V |
| GPS_RXD, GPS_TXD | Konektor sériové linky, primárně určený k připojení GPS. (Lze připojit libovolné zařízení s podporou USART.) |
| ZEM_RXD, ZEM_TXD | Konektor sériové linky. Lze připojit libovolné zařízení s podporou USART. |
| ADC0..3 | 10bitové analogové vstupy. Naměřené napětí je programu předáno jako 10bitové číslo v rozsahu od GND po hodnotu VCC. |
| SD | Slot microSD karty. Lze použít microSD/SDHC karty. |
| OC0..3 | Pady, které lze použít pro připojení dalších zařízení ovládaných pomocí PWM. |
| INPWM3 | Pad, který lze použít jako přijímač PWM signálu. |
| START2 | Pad, ke kterému lze připojit programově detekovatelné tlačítko. |

Každý z výše uvedených pinů lze použít jako digitální vstup nebo výstup. Použití každého pinu je dáno programem. Pokud chceme použít jednotlivé piny jako digitální vstupy nebo výstupy, použijeme označení PORT A..G (PA0..PA7..PF0..PF7..PG0..PG4). Viz datasheet Atmel ATmega128 [4].

A.3 Schéma zapojení elektroniky

Na schématech 27 a 28 je uvedeno vnitřní zapojení elektroniky Yunify. Schémata 29 a 30 znázorňují tištěný spoj desek Main a Basic bez součástek. První schéma znázorňuje celou desku, druhé vrchní stranu desky a třetí schéma znázorňuje spodní stranu desky.

B API, firmware a řízení mobilním telefonem

B.1 Firmware Yunify

Elektronika Yunify je řízena dvěma procesory ATmega128 [4]. Firmware je nahrán v procesoru Basic a procesor Main je určen k uživatelskému naprogramování. Do procesoru Main je možné například nahrát program pro řízení letounu na základě dat z akcelerometrů a dynamického gyroskopu, který je popsán v kapitole B.2. Uživatelské programy používají soubor knihoven, které tvoří API (Application Programming Interface) elektroniky Yunify.

B.1.1 Firmware desky Basic

Firmware desky Basic řídí celý stroj v případě nefunkčnosti programu Main nebo na základě pokynu od uživatele. Díky tomu je možné například za letu bezdrátově přeprogramovat procesor Main.

Deska Basic reguluje všechny ovládací prvky pomocí PWM výstupů, čte přijímaný RC signál na input capture pinu (ICP1 a ICP3), komunikuje po dvou sériových linkách (USART0 a USART1) a disponuje sběrnici I²C, po které komunikuje s deskou Main a dalšími zařízeními připojenými k této sběrnici. Z desky Basic je možné vyvést i další porty, které jsou z hmotnostních důvodů vyvedeny pouze jako pady.

Program desky Basic přijímá po sběrnici I²C pokyny k řízení od desky Main. V případě neobdržení žádných dat pro řízení je stroj řízen na základě RC signálu nebo dat přijatých po sériové lince. Pokud jsou k dispozici data z akcelerometrů, letoun se pokusí o přímý let nebo kroužení. Pokud nejsou dostupná žádná data, program vypne motor a všechna regulační serva nastaví do střední polohy (FailSave mode).

Konkrétní chování procesoru Basic nastavíme příkazem pomocí sériové linky připojené k PC nebo mobilnímu telefonu s podporou sériového terminálu. Ihned po spuštění procesoru můžeme chování nakonfigurovat pomocí RC vysílače. Režim může dále změnit program desky Main příkazem po sběrnici I²C. Podrobný popis příkazů a konfigurace pomocí RC vysílače nalezneme v dokumentaci programu Yunify.

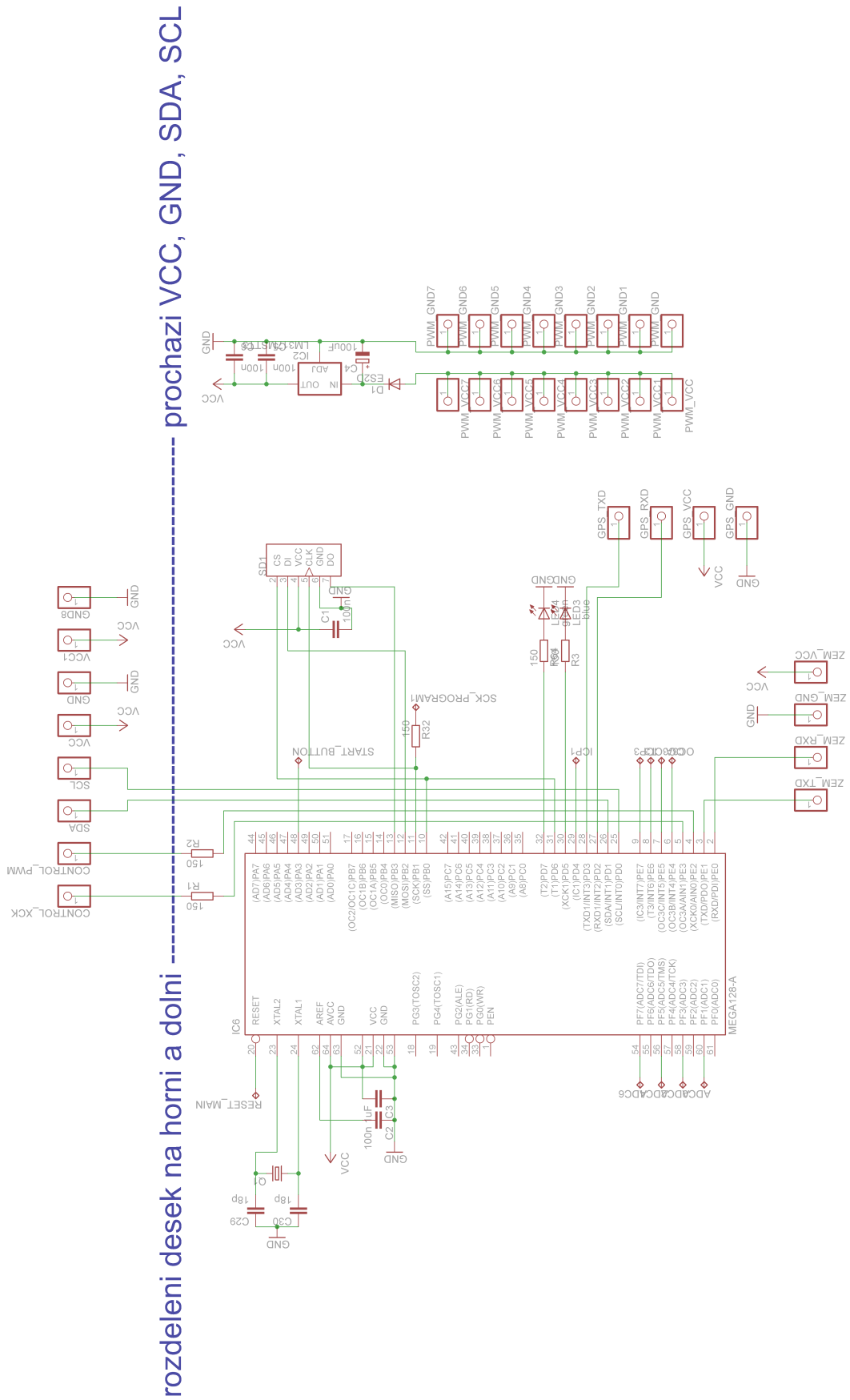
B.1.2 API desky Main

Deska Main je určena k naprogramování uživatelským programem. K vytvoření tohoto programu slouží API včetně několika referenčních programů.

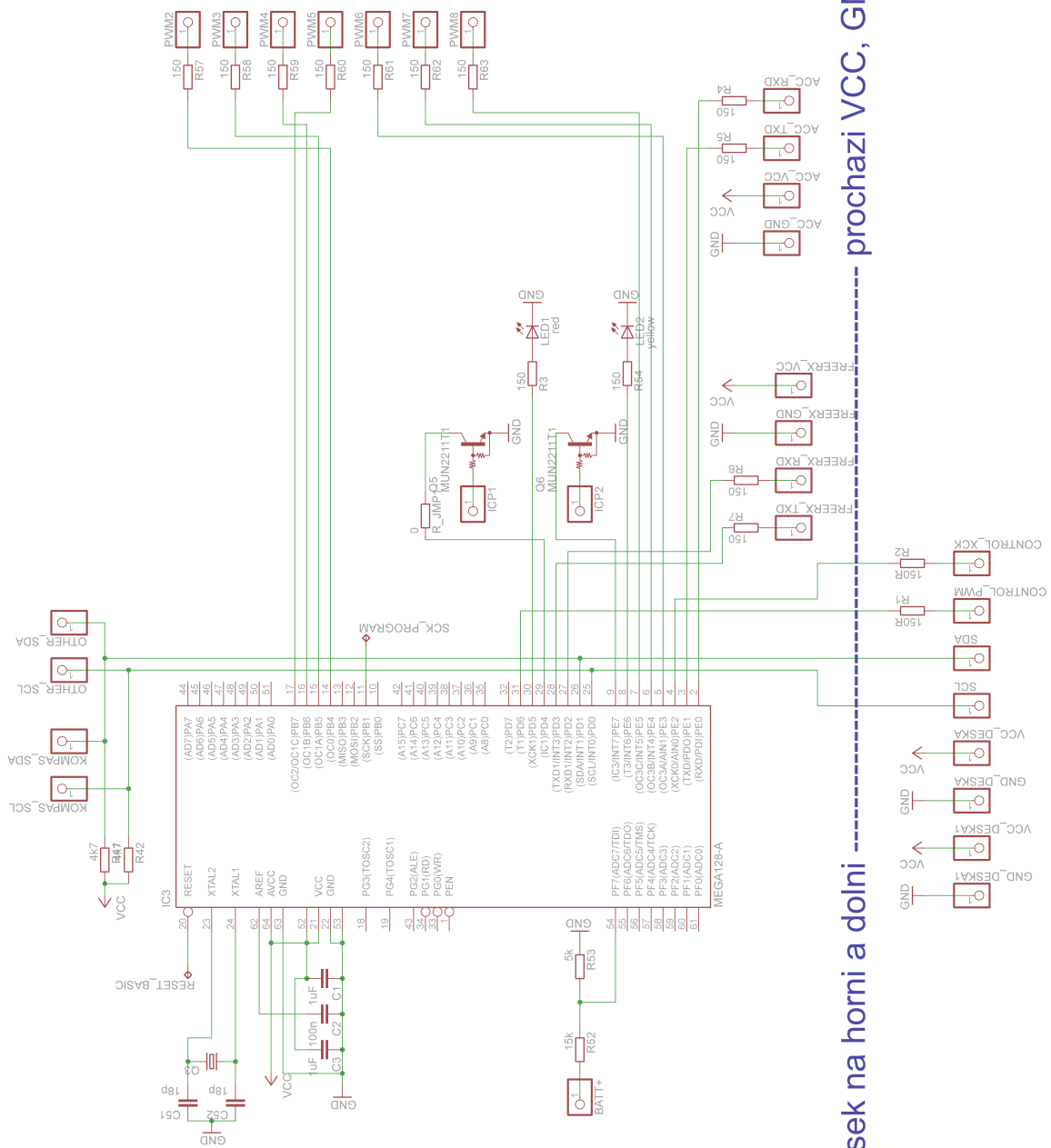
API se skládá z následujících částí:

1. knihovna AVRlib (avrilib a kuba_avrilib) [23], která poskytuje funkce pro základní obsluhu procesoru a dat
2. knihovny pro systém Arduino [1] poskytující rozšířenou sadu ovládání
3. knihovna pro komunikaci s microSD/SDHC kartou [12]
4. knihovny Yunify poskytující další funkce

Externí knihovny zde nejsou dále rozebírány, neboť veškeré informace můžeme najít v jejich dokumentaci.

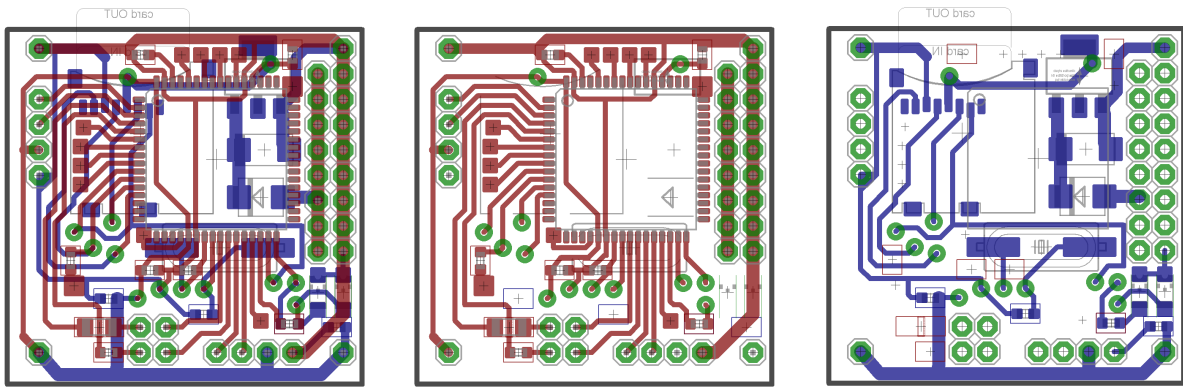


Obrázek 27: Elektronické schéma zapojení desky Main

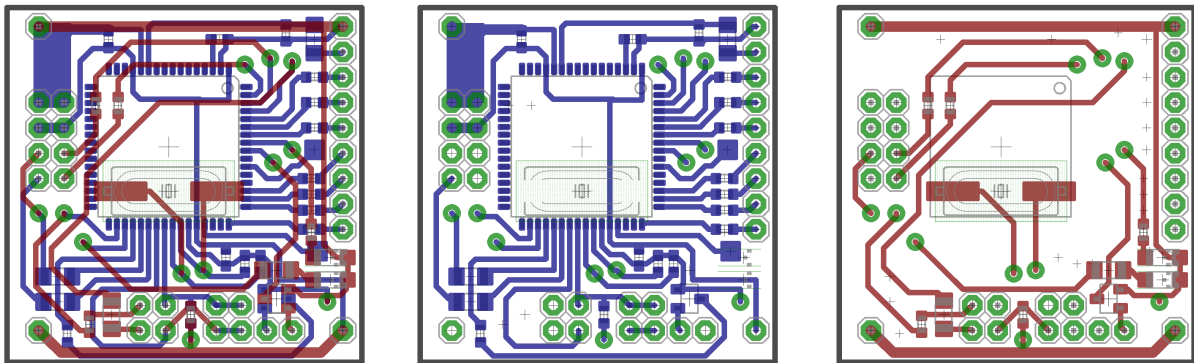


rozdeleni desek na horni a dolni ----- prochazi VCC, GND, SDA, SCL

Obrázek 28: Elektronické schéma zapojení desky Basic



Obrázek 29: Obrázek tištěného spoje desky Main bez součástek



Obrázek 30: Obrázek tištěného spoje desky Basic bez součástek

B.1.3 Knihovny Yunifly

Knihovny Yunifly obsahují základní strukturu každého programu pro Yunifly. Programátor vytvoří některé z následujících funkcí, které jsou dále volány ve funkci main().

1. **void run()** ... funkce zavolaná ihned po inicializaci programu, pokud obsahuje nějaký kód, je proveden ihned po spuštění, může pracovat do nekonečna
2. **void loop()** ... tato funkce je periodicky volána po skončení funkce run(), dokud není elektronika vypnuta

Soubor basic.h obsahuje funkci main():

```
int main()
{
    init ();
    run ();
    for (;;)
    {
        loop ();
    }
    return 0;
}
```

Funkce `init()` povolí přerušení, aktivuje sériovou linku a umožní tak softwarový reset a vstup do bootloaderu. Počká 10 ms pro ustálení napětí a proudu na desce a na sériovou linku odešle informaci o úspěšném spuštění. Dále pokračuje uživatelský program, neboť další obsluha je prováděna uvnitř přerušení.

Soubory `accelerometer.h`, `GPS.h`, `kompas.h`, `paralTransmitter.h` a `RCcapture.h` obsahují funkce pro komunikaci s externími zařízeními uvedenými v kapitole 3.4.2. Definují funkce pro inicializaci zařízení, komunikaci, získávání dat a odpojení. Podrobná specifikace je uvedena v dokumentaci k daným souborům.

Soubory `fastPWM.h`, `softPWM.h` a `led.h` definují specifické porty pro ovládání LED a PWM. Jejich připojení je nutné výhradně v programech pro elektroniku Yunifly. V případě absence těchto knihoven nebudou fungovat funkce:

```
void initLEDs();
void setLed(char LEDconfig = 0xFF);
void clearLed(char LEDconfig = 0xFF);
void toggleLed(char LEDconfig = 0xFF);
void initFastPWM(
    uint32_t overflowValue = 20000,
    int initialValue = 0
);
void initFastPWMServo(
    uint32_t overflowValue = 20000,
    int initialValue = 0
);
void setFastPWM(int id, int value);
void setFastPWMServo(int id, int value);
void setAllServos(int v1, int v2, int v3, int v4, int v5, int v6);
void testRangePWM();
```

B.2 Řízení letounu na základě dat z akcelerometrů a dynamického gyroskopu

Soubor `stabilization.h` knihovny Yunifly a k němu připojené soubory obashují funkce nutné pro řízení letounu pouze na základě dat z akcelerometrů a dynamického gyroskopu. Stabilizace a řízení letounu je definováno ve třídě `Tstabilization`, která je popsána v dokumentaci.

```
class Tstabilization
{
public:
    bool narrow = true;           //fly narrow or on the circle
    int radius = 0;               //radius of the circle

    void update(int * accData);
        //get new data from accelerometer and gyro
    void calibration();
        //calibration on the ground
    void listen();
```

```
        //calibration on air
void setDriving(int * RCcontrol);
        //setting RCcontrol for driving
}
```

B.3 Aplikace pro mobilní telefon

Aplikace pro mobilní telefon je skript napsaný v jazyce Python [17]. Podporují jej všechny telefony s aplikací Python interpret, který lze bezplatně stáhnout na domovských stránkách [19].

V konfiguračním souboru config.ini je nutné nastavit adresu serveru, se kterým bude telefon komunikovat. Aplikace představuje most sériové linky mezi elektronikou Yunifly a serverem. Díky tomu je možné ze serveru přímo komunikovat s celým systémem.

Soubor config.ini :

```
address = 127.0.0.1
port = 2121
user = Yunifly
password = Yunifly
```

Aplikace dále čte data z mobilního sensoru GPS, akcelerometru a případně kompasu. Jednotlivé hodnoty jsou zasílány po sériové lince elektronice Yunifly a na server. Je možné nastavit odesílání těchto dat formou SMS s určitým intervalem opakování nebo v případě ztráty internetového spojení se serverem.

Nahrávání, fotografování a streamování videa na server je zajištěno externě programem Skype [20] nebo Fring [21]. Python skript umožňuje pouze vyfotografování snímku prostřednictvím příkazu po sériové lince a následným uložením snímku do paměti telefonu nebo na paměťovou kartu. Skript má k vyfotografovanému obrázku přístup jako k dvojrozměrnému poli typu integer. Není proto problém k programu přidat i zpracování vyfotografovaného obrazu.

Skript je testován na telefonu Nokia 5230 s interpretem Python for S60v5 [18].