Rozšíření řídicí jednotky multirotorových dronů mikropočítačem se senzory

Matyáš Pokorný

Bakalářská práce
2020

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

Ústav automatizace a řídicí techniky

Akademický rok: 2019/2020

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Matyáš Pokorný**
Osobní číslo:	**A17617**
Studijní program:	**B3902 Inženýrská informatika**
Studijní obor:	**Informační a řídicí technologie**
Forma studia:	**Prezenční**
Téma práce:	**Rozšíření řídicí jednotky multirotorových dronů mikropočítačem se senzory**
Téma práce anglicky:	**Extending a Flight Controller by a Microcontroller with Sensors**

Zásady pro vypracování

1. Prozkoumejte aktuální stav trhu řídicích jednotek pro multirotorové drony.
2. Prostudujte možnosti realizace řízení letového kontroleru dronů z připojeného mikropočítače, např. pomocí signálů PWM, protokolu SBUS nebo FR-SmartPort.
3. Vyberte a otestujte řídicí jednotku a vývojový kit s mikropočítačem, které budou pro tento projekt vhodné.
4. Implementujte a otestujte komunikaci mikropočítače s řídicí jednotkou.
5. Naprogramujte ukázku použití výsledného systému, např. pro udržování zadané výšky dronu nad zemí pomocí senzoru vzdálenosti.

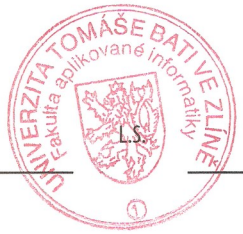
Rozsah bakalářské práce:
Rozsah příloh:
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. KERNIGHAN, Brian W. a Dennis M. RITCHIE. Programovací jazyk C. 2. vydání. Přeložil Zbyněk ŠÁVA. Brno: Computer Press, 2019. ISBN 9788025149652.
2. BARR, Michael. Programming embedded systems in C and C++. Sebastopol, Calif.: O'Reilly, c1999. ISBN 1565923545.
3. PINKER, Jiří. Mikroprocesory a mikropočítače. Praha: BEN – technická literatura, 2004. ISBN 8073001101.
4. PARK, John, Steve MACKAY a Edwin WRIGHT. Practical Data Communications for Instrumentation and Control. Amsterdam ; London: Elsevier, 2003. ISBN 9780750657976.
5. DORF, R. C. a R. H. BISHOP. Modern Control Systems. California, 1998. ISBN 0201326779.

Vedoucí bakalářské práce: **Ing. Tomáš Dulík, Ph.D.**
Ústav informatiky a umělé inteligence

Datum zadání bakalářské práce: **20. prosince 2019**
Termín odevzdání bakalářské práce: **15. května 2020**



doc. Mgr. Milan Adámek, Ph.D.
děkan

prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

Ve Zlíně dne 20. prosince 2019

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor;
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

Matyáš Pokorný, v.r.
podpis diplomanta

ABSTRAKT

Tato bakalářská práce se zabývá řízením multirotorových dronů, řídicími jednotkami, a komunikačními protokoly používanými k jejich ovládání. Cílem práce je využít jednoho z těchto komunikačních protokolů k vytvoření systému pro automatické udržování výšky dronu nad zemí. V práci je navrženo řešení pro naplnění tohoto cíle sestávající z mikropočítače, senzoru vzdálenosti a programu pro mikropočítač, implementující algoritmus diskrétního PID regulátoru.

Klíčová slova: Dron, řídicí jednotka dronu, MCU, STM32, Embedded, PID, SBUS, iBUS, PWM, PPM

ABSTRACT

This Bachelor‘s thesis explores the topic of multirotor drone control, flight controllers and flight controller communication protocols. The aim of the thesis is to use one of these communication protocols to create a system for maintaining constant level hover. The thesis proposes a solution to meet this aim, which involves a microcomputer, a distance sensor and an implementation of discrete PID controller algorithm.

Keywords: Drone, Flight controller, MCU, STM32, Embedded, PID, SBUS, iBUS, PWM, PPM

Děkuji svému vedoucímu, Ing.Tomáši Dulíkovi, Ph.D, za poskytnutou pomoc při realizaci této práce, a své rodině a přátelům za podporu po dobu studia.

OBSAH

ÚVOD

Odvětví dronů zažívá v současné době období velkého rozvoje a popularity. Multirotorové drony pronikají do stále nových oblastí lidské činnosti. Navzdory tomu není jednoduché získat detailní a ucelené informace o technologiích používaných v systémech multirotorových dronů, zvláště pak v českém jazyce. Prvořadým cílem této práce je přispět ke snížení tohoto deficitu, a nabídnout čtenáři pohled do vnitřního fungování dronů. Hlavním předmětem zájmu v této práci jsou řídicí jednotky dronů jejich součásti a fungování. Dále jsou popsány technické detaily některých často používaných způsobů komunikace řídicích jednotek.

Druhým cílem práce je využít získaných poznatků o řídicích jednotkách, komunikačních protokolech a mikropočítačích k sestrojení systému pro automatické udržování zadané výšky dronu nad zemí. K tomuto účelu je využito vývojové platformy STM32 a ultrazvukového senzoru vzdálenosti. Funkce navrženého systému, jeho testování a dosažené výsledky jsou popsány v praktické části práce.

Množství výrobců multirotorových dronů, jejich součástí, rádiové techniky a hotových řešení je nespočetné množství. Tato práce si neklade za cíl obsáhnou všechna existující řešení, ale zaměřuje se na segmenty trhu, které jsou přístupné pro získávání informací, umožňují návrh vlastních řešení a přizpůsobení výrobků potřebám koncového uživatele. Hotová řešení, nevyžadující detailní znalost problematiky uživatelem, nebo uzavřené proprietární systémy, ačkoliv tvoří většinu objemu trhu (například některé populární produkty renomované značky DJI), nebudou v této práci rozebírány.

I. TEORETICKÁ ČÁST

1 ÚVOD DO TECHNOLOGIÍ MULTIROTOROVÝCH DRONŮ

1.1 Komerčně dostupné multirotorové drony

Nejrozšířenějším typem komerčních dronů jsou multirotorové koptéry. Podle počtu poháněných rotorů hovoříme o kvadrokoptérách– 4 rotory, hexakoptérách – 6 rotorů, oktokoptérách – 8 rotorů, atd. Drony malé a střední velikosti (desítky gramů až nízké jednotky kilogramů) jsou typicky kvadrokoptérami, u větších je běžný i větší počet rotorů, zejména pokud je dron určen k nesení vybavení, či nákladu.

1.2 Základní stavba

Řada základních součástí multirotorových dronů je i přes širokou škálu jejich rozměrů a využití stejná. Tyto základní součásti jsou:

- **Rám**
  Drží všechny součásti dronu pohromadě a chrání elektroniku před nárazy a vnějšími vlivy. Často používanými materiály pro rám jsou umělé hmoty a materiály na bázi uhlíkových vláken.

- **Motory**
  Hlavní součást určující výkon dronu. Velmi populární pro toto použití jsou bezkartáčové stejnosměrné motory (BLDC) se statorem tvořeným cívkami a rotorem tvaru zvonu s permanentními magnety. U nejmenších dronů jsou používány i klasické stejnosměrné motory.

- **Elektronické regulátory otáček**
  Řídí otáčky motoru podle signálu z řídicí jednotky. Jsou nezbytné při použití bezkartáčových motorů s elektronickou komutací.

- **Řídicí jednotka**
  Hlavní řídicí počítač, nezbytný pro stabilní let dronu. Stabilizaci provádí regulací tahů jednotlivých motorů v reálném čase na základě údajů ze zabudovaných senzorů přímočarého a úhlového zrychlení.

- **Rádiový přijímač**
  Umožňuje dálkové ovládání dronu z vysílačky nebo pozemní řídicí stanice. Přijímá rádiový signál z vysílače a odesílá hodnoty kanálů do řídicí jednotky. Také umožňuje přenos telemetrických dat zpět řídicímu stanovišti.

- **Baterie**
  Populární jsou vícečlánkové baterie typu Li-pol.

- **Vrtule**
  Vytvářejí tah potřebný k letu. Existuje mnoho typů lišících se počtem listů, jejich sklonem šířkou apod.

Podle účelu ke kterému je dron sestrojen mohou být základní části doplněny dalšími, zajišťujícími rozmanité funkce:

- **GPS přijímač**
  Pro sledování polohy, přesnou navigaci, vytváření tras s přednastavenými body a funkci automatického návratu na výchozí souřadnice (Return to Home).

- Kamery
  Často drony nesou alespoň jednu, někdy i větší počet kamer. Tyto kamery mohou být určeny pro videoprodukci, nebo také pro pilotování (zejména závodních) dronů pouze podle obrazu z kamery (First Person Video).

- **Gimbal**
  Otočné uchycení, kterým je možné nasměrovat osazenou kameru požadovaným směrem.

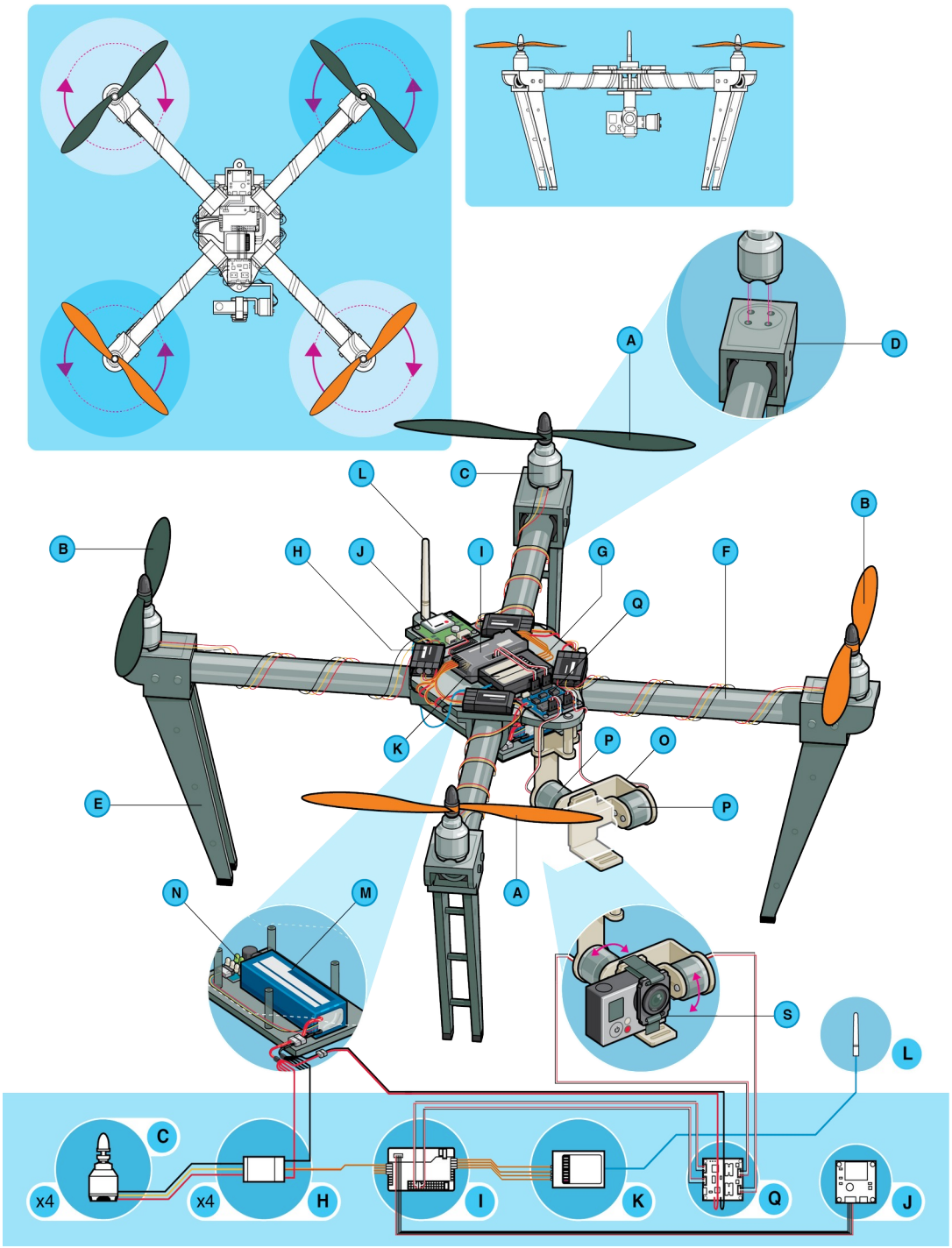
- **Rádiový vysílač videosignálu**
  Pro přenos signálu z kamer zpět k pilotovi.

- zařízení pro záznam letových údajů (černá skříňka)

- bzučák

- LED pásky

- další senzory (barometr, kompas, senzory vzdálenosti apod.)



Obr. 1. Stavba kvadrokoptéry [20, s. 34]

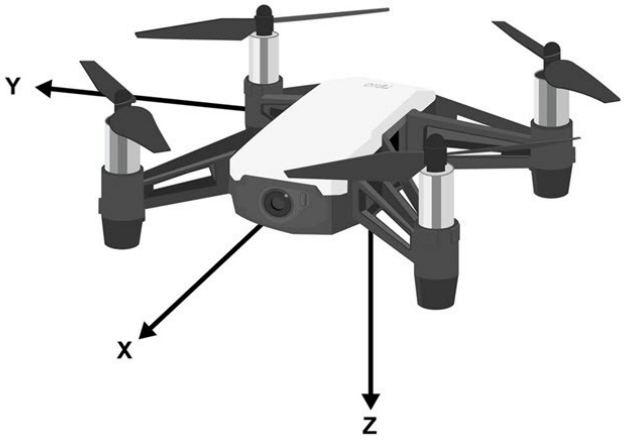
A,B – vrtule (opačné směry otáčení), C – motor, D – uchycení motoru, E – přistávací noha, F - rameno, G - trup, H – elektronický regulátor otáček motoru, I – řídicí jednotka, J – GPS modul, K – radiopřijímač, L – anténa rádiového řízení, M – baterie, N - měřič stavu baterie, O - gimbal, P - aktuátor gimbalu, Q - řadič gimbalu, S - kamera

1.3 Řízení

1.3.1 Popis pohybu

Modelování pohybu dronu je nutné pro návrh stabilizačních a řídicích mechanismů. Toto téma již bylo zpracováno v řadě odborných prací, a nalezené modely jsou poměrně složité. Pro účely této práce bude stačit zjednodušený popis, neboť se nezabývá řízením rotačních pohybů, či horizontální polohy dronu.

Základní popis pohybu využívá kartézského souřadného systému v tří rozměrném prostoru (R^3). Počátek soustavy lokálních souřadnic je umístěn do těžiště dronu, osy x a y jsou položeny v horizontální rovině.



Obr. 2: Souřadný systém dronu[19]

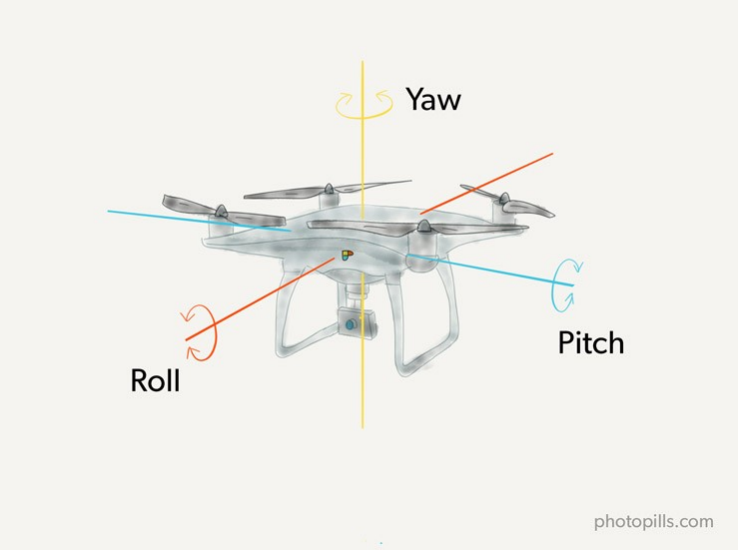
Důležitější částí popisu je popis rotačních pohybů dronu.

„V R^3 prostoru lze rotaci objektu vyjádřit třemi úhly otočení oproti inerciální soustavě (Obr. 1). Historicky tuto vlastnost objevil a popsal již v polovině 18. století Leonard Euler a proto se příslušné úhly nazývají Eulerovy úhly. Tyto úhly rotací kolem inerciálních kartézských souřadnic jsou pojmenovány následovně:

- φ (α) úhel rotace kolem osy x – úhel vlastní rotace (angl. **Roll**)
- ψ (β) úhel rotace kolem osy y – precesní úhel (angl. **Pitch**)
- ϑ (γ) úhel rotace kolem osy z – nutační úhel (angl. **Yaw**) “

[1,s. 11, zvýraznění autorovo]

Na obrázku (Obr. 3) jsou vyznačeny osy rotace podle Eulerových úhlů na kvadrokoptéře.

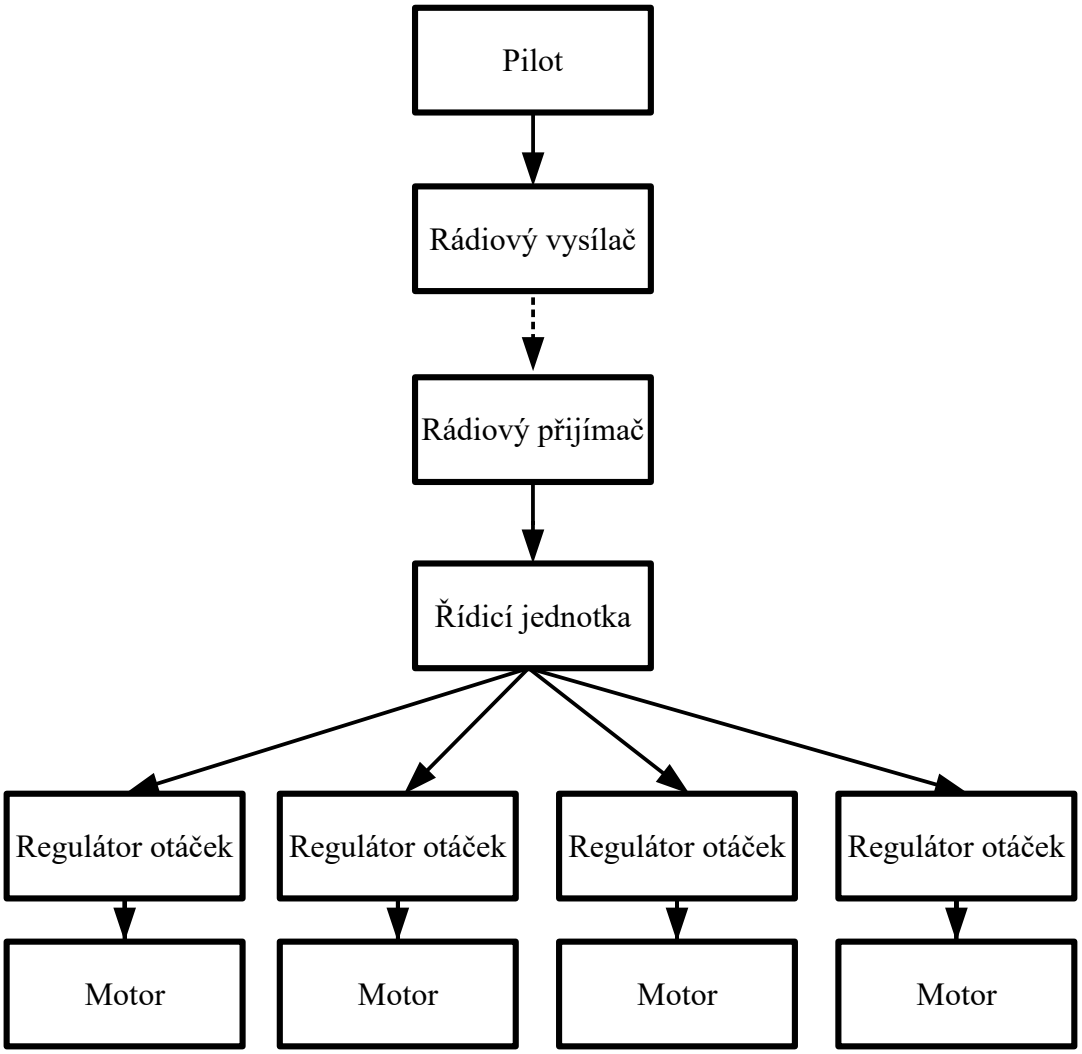


Obr. 3: Eulerovy úhly kvadrokoptéry[25]

1.3.2 Popis ovládání

Systém ovládání multirotorových dronů vychází z technologií řízení dálkově ovládaných modelů. Soustava dálkového ovládání se typicky skládá z vysílačky a radiopřijímače. Radiopřijímač je umístěn v modelu a je připojen k aktuátorům modelu, jako například servomotorům. Manipulací s ovládacími prvky vysílačky pilot nastavuje hodnoty kanálů, které jsou přenášeny do radiopřijímače, a ten jejich hodnoty vysílá k jednotlivým aktuátorům. Kanálem je zde myšlen signál sloužící k řízení konkrétního prvku nebo funkce modelu. Například jeden kanál může být využit pro řízení otáček motoru, další pro nastavení polohy klapek, nebo vysunutí podvozku apod.

U multirotorových dronů je vždy alespoň několik kanálů vedeno do řídicí jednotky. Pro ovládání základních pohybů jsou potřeba alespoň čtyři kanály – pro ovládání rotací Pitch, Roll a Yaw, a tahu motorů. Běžné je ale využití více kanálů pro další funkce, jako jsou zajištění a odjištění motorů, přepínání letových režimů apod. Na Obr. 4 je znázorněna struktura řízení dronu. Typické mapování kanálů na prvky dálkového ovladače je na Obr. 5.



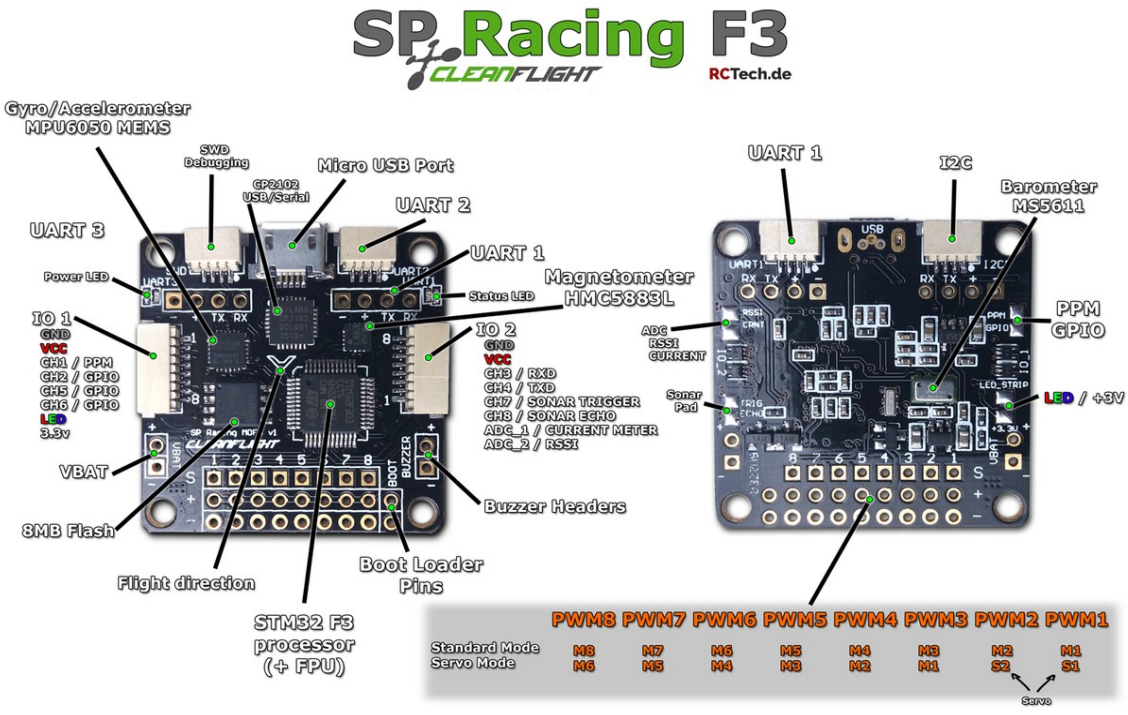
Obr. 4: Struktura řízení dronu

Obr. 5: Nejčastěji používané mapování ovládacích kanálů na prvky ovladače[18]

2 SOUČASNÝ STAV TRHU ŘÍDICÍCH JEDNOTEK

Nabídka řídicích jednotek na trhu je v současné době velmi bohatá a umožňuje realizaci mnoha rozmanitých konfigurací dronů. Základem typické řídicí jednotky je deska plošných spojů (dále DPS), na níž jsou osazeny elektronické komponenty potřebné pro funkci jednotky. DPS je často čtvercová, a je uchycena k rámu dronu pomocí čtveřice šroubů procházejících otvory v jejích rozích. Vzdálenost mezi těmito otvory udává rozměry řídicí jednotky. Často používané jsou rozměry 16x16 mm, 20x20 mm, a 30x30 mm. Tyto rozměry jsou podporovány výrobci rámů a dalších součástí dronu (např. 4v1 regulátorů otáček), a staly se určitým standardem. Existují ale i jiná provedení s jinými rozměry DPS, nebo vlastním plastovým krytem (například populární jednotky Pixhawk). Cena řídicích jednotek se pohybuje v řádu desítek až nižších stovek amerických dolarů za kus. Na Obr. 6 je popsána řídicí jednotka SP Racing F3.

Následující podkapitoly jsou věnovány nejdůležitějším součástem řídicích jednotek, které jsou v současné době dostupné na trhu. Prezentované informace byly získány průzkumem nabídky internetových obchodů nabízejících součásti dronů, a také průzkumem nabídky výrobců na jejich webových stránkách. Takovými obchody jsou například: www.getfpv.-com, www.banggood.com, www.ebay.com, www.rotorriot.com, a mnoho dalších. V České republice je to například www.rotorama.cz.



Obr. 6: Popis řídicí jednotky SP Racing F3 [26]

2.1 Procesor

Jedním z nejdůležitějších parametrů řídicích jednotek je bezpochyby použitý procesor. V tomto směru zaznamenávají v současné době řídicí jednotky růst výpočetního výkonu spolu s růstem výkonu a dostupností mikrokontrolérů. Velké popularitě se těší mikrokontroléry výrobce STMicroelectronics , zejména řady STM32F1, F3, F4, F7. Z důvodů zvyšujících se požadavků na výkon jsou v současné době u nových jednotek nejpoužívanější zejména poslední dvě jmenované řady, založené na jádrech ARM Cortex M4 a M7. Tyto procesory používá ve svých výrobcích celá řada výrobců součástí pro drony, jako například: Holybro, Lumenier, Matek, Flyduino a další.

Dalšími, ač daleko méně rozšířenými mikrokontroléry používanými v řídicích jednotkách jsou MCU řady ATMega (Atmel) použity v jednotce Ardupilot APM 2.8, nebo MCU řady PIC (Microchip) použity v jednotce MatrixPilot. Existují také produkty využívající malého jednodeskového počítače Raspberry Pi – jednotka Erle-Brain, nebo PXFmini. [2]

Nejdůležitějšími parametry mikrokontroléru pro použití v řídicí jednotce jsou zejména výpočetní výkon, velikost paměti, a také vstupně výstupní rozhraní. Výpočetní výkon, počet rozhraní a podpora různých sběrnic jsou určujícím faktorem při připojení senzorů a periferií k jednotce. Běžně používanými sběrnicemi v technologii dronů jsou UART, SPI, I^2C, nebo také CAN. Zejména počet rozhraní UART může být limitujícím faktorem pro současný provoz sériového radiopřijímače, odesílání telemetrie, záznamu letových údajů atd.

2.2 Senzory

Pro účely řízení orientace a zrychlení dronu při letu je nutné tyto veličiny snímat. K tomu slouží gyroskopy a akcelerometry osazené na DPS řídicí jednotky. Tento systém se souhrnně nazývá inerciální měřící jednotka (IMU), a většinou je proveden jako MEMS integrovaný v jediném pouzdře. Hojně jsou používány například produkty MPU6000 a řada ICM20600. Činnost měřící jednotky může být při letu negativně ovlivněna vibracemi pocházejícími z motorů . Z toho důvodu jsou některé řídicí jednotky dodávány s pryžovými podložkami, aby se zabránilo přenosu vibrací na DPS, nebo je měřící jednotka umístěna zvlášť na menší desce, která je k hlavní připevněna pěnovým materiálem, tlumícím přenos vibrací. Některé řídicí jednotky obsahují dvě redundantní IMU, mezi kterými může pilot

přepínat podle preferovaných vlastností, nebo jsou používány současně pro zvýšení spolehlivosti a přesnosti měření orientace (například u jednotky SP Racing F7 DUAL).

Podle zaměření řídicí jednotky mohou být osazeny i další senzory. Například verze určené pro kamerové platformy zpravidla také obsahují přesný barometr pro autonomní udržování letové výšky, a také magnetometr pro orientaci podle magnetického pole Země. Jednotky, nebo jejich verze (často označeny jako „acro“, nebo „6DOF“), určené pro závodní, či akrobatické drony tyto senzory zpravidla neobsahují, neboť jejich funkcí v těchto disciplínách není zapotřebí.

2.3 Vstupní a výstupní rozhraní

Nejdůležitější funkce plněné vstupními a výstupními rozhraními zahrnují příjem řízení z radiopřijímače, výstup řídicích signálů pro elektronické regulátory otáček motorů, a připojení řídicí jednotky k osobnímu počítači za účelem její konfigurace. Standardním způsobem připojení jednotky k PC je konektor USB micro-B, ale existují i jednotky s integrovaným bezdrátovým rozhraním Bluetooth, pomocí kterého je možné provést konfiguraci ze spárovaného smartphone (jednotka SpeedyBee F4). DPS řídicích jednotek obsahují vývody pro připájení vodičů, nebo osazené řady pinů pro jejich jednoduché připojení. Široce používaným standardem jsou také konektory JST řady SH. . Řada jednotek má také osazen slot pro paměťovou Micro SD kartu, pro účely záznamu letových údajů , neboť kapacita flash paměti mikrokontroléru, nebo modulu osazeného na DPS, může být při využití této funkce omezující.

2.4 Integrované součásti

Na DPS řídicí jednotky mohou být integrovány další součásti, které jsou jinak zcela samostatnými produkty. Mohou to být například rozvod napájení a regulátory napětí, systém pro zobrazování letových údajů ve vysílaném videosignálu - OSD , nebo i elektronické regulátory otáček. Produkty, které integrují v řídicí jednotce některé z těchto součástí se označují jako All-in-one (AIO). Zejména pro velmi malé multirotorové drony (whoop) je integrace součástí velmi žádoucí, z důvodu redukce hmotnosti a prostorových nároků. U řídicích

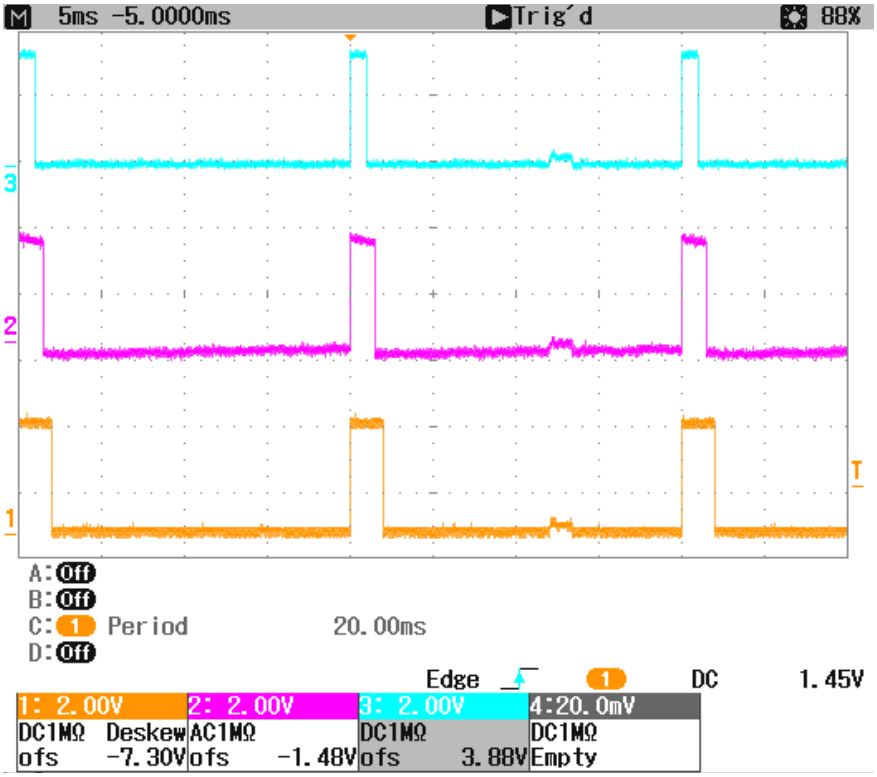
jednotek určených pro tyto drony jsou na jedinou DPS integrovány všechny výše zmiňované komponenty a dokonce i radiopřijímač (například jednotka Happymodel Crazybee F4 Pro).

3 KOMUNIKAČNÍ PROTOKOLY ŘÍDICÍCH JEDNOTEK A RADIOPŘIJÍMAČŮ

Tato kapitola pojednává o vybraných protokolech komunikace řídicích jednotek a radiopřijímačů. Blíže popisované komunikační protokoly jsou široce podporovány výrobci řídicích jednotek, tudíž jsou dobrými kandidáty pro komunikaci řídicí jednotky a připojeného mikropočítače se senzory.

3.1 PWM

Pulzně šířková modulace (Pulse Width Modulation) je známý způsob modulace elektrického signálu, používaný například pro digitální přenos analogového signálu nebo pro řízení výkonu elektrických strojů. Je to také nejjednodušší způsob přenosu řídicích signálů z radiopřijímače do řídicí jednotky dronu. Hodnota přenášeného kanálu je reprezentována dobou trvání kladného pulzu, který se periodicky opakuje. Na Obr. 7 je průběh signálů s modulací PWM pro tři různé současně přenášené kanály. Přenášená hodnota prvního kanálu je 100%, druhého 50%, a třetího 0% (zespoda nahoru). Těmto hodnotám odpovídají šířky pulzů 3 ms, 1,5 ms, a 1 ms. Perioda všech signálů je 20 ms. Nevýhodou tohoto způsobu přenosu je vyšší počet vodičů neboť každý přenášený kanál vyžaduje svůj vlastní signální vodič – jedná se o paralelní přenos.



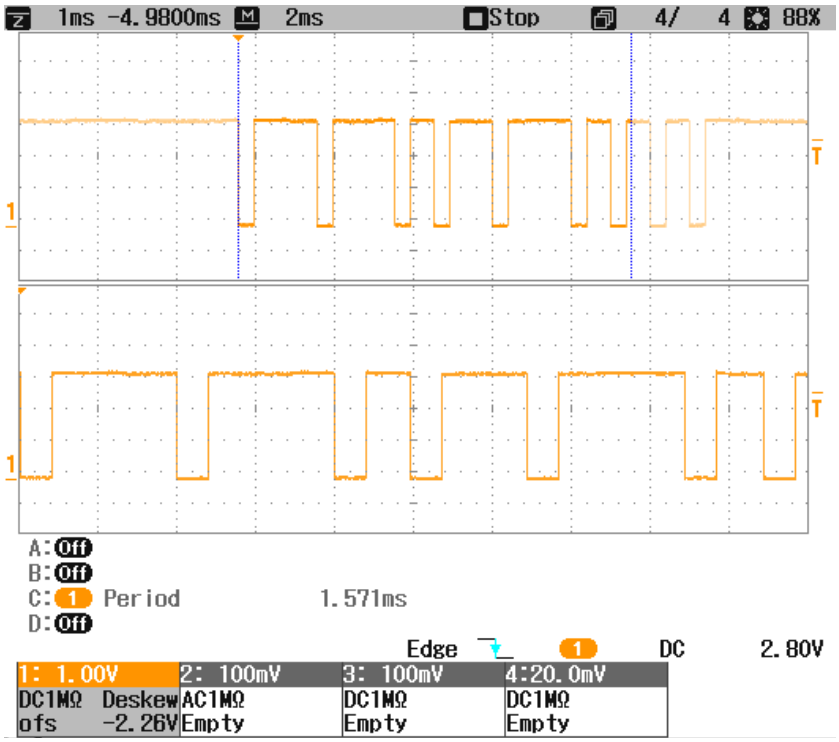
Obr. 7 Přenos hodnot kanálů pomocí pulzně šířkové modulace

3.2 PPM

Pulzně polohová modulace (Pulse Position Modulation) je dalším způsobem modulace digitálního signálu, která umožňuje přenos vzorkovaného analogového signálu, reprezentovaného časovými intervaly.

Signál PPM je tvořen stejně širokými krátkými impulzy s konstantní amplitudou, jejichž poloha v rámci periody závisí na hodnotě přenášeného signálu. Nízkým hodnotám přenášeného signálu odpovídá pozice impulzu blíže k počátku periody, naopak vysokým hodnotám odpovídá pozice impulzu blíže konci periody.[3]

Pro účely přenosu řídicích kanálů z radiopřijímače do řídicí jednotky je tato modulační technika upravena tak, že je možné přenášet hodnoty více kanálů jedním signálním vodičem. Signál je inverzní (klidový stav 2,8 V, impulz 0 V) a impulzy jsou seřazeny za sebe do „rámce“ , přičemž první impulz je synchronizační a další vyjadřují hodnoty přenášených kanálů. Hodnota kanálu je vyjádřena prodlevou mezi daným impulzem a předchozím impulzem, nikoli počátkem periody. Rámce tak mají proměnlivou délku. Průběh signálu PPM přenášejícího osm řídicích kanálů je zachycen na Obr. 8. V horní části obrázku je zachycen celý „rámec“, v dolní části je přiblížen výřez z horní části.



Obr. 8 Přenos hodnot kanálů pomocí pulzně polohové modulace

3.3 SBUS

SBUS je protokol sériové datové komunikace původně vyvinutý společností Futaba pro řízení modelářských servomotorů. Vychází ze standardní sériové komunikace UART s inverzními napěťovými úrovněmi, tedy nízká úroveň (0 V) vyjadřuje log. 1, a vysoká úroveň log. 0. Parametry a struktura sériové komunikace protokolu SBUS jsou shrnuty a názorně vyobrazeny v tabulce 1 a Obr. 9.

Tab. 1 Parametry sériové komunikace SBUS

Parametr	Hodnota
Napěťové úrovně	inverzní
Modulační rychlost	100 000 baudů
Počet datových bitů	8
Parita	sudá
Počet stop bitů	2
Velikost rámce	25 bytů

Data jsou při použití protokolu SBUS posílány v pravidelných intervalech v rámcích tvořenými 25 byty, přičemž první byte má vždy hodnotu 15, poté následuje 22 bytů s hodnotami kanálů, jeden byte s dvěma binárními kanály a dvěma příznaky „frame lost“ a „failsafe“, a nakonec jeden byte s hodnotou 0. Hodnota každého přenášeného řídicího kanálu je vyjádřena 11bitovým číslem. Tyto 11bitové hodnoty následují v 22 datových bytech za sebou, jeden datový byte tak může obsahovat bity náležející k různým kanálům. [4]

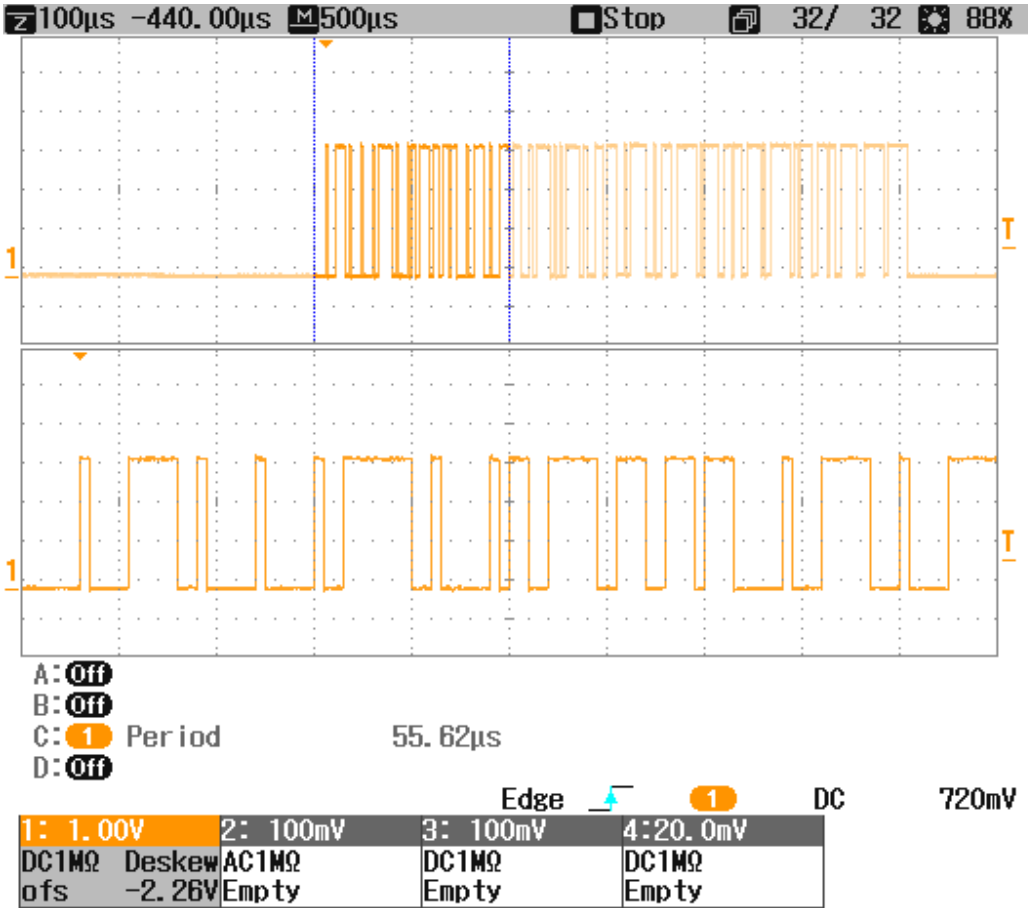
Struktura rámce protokolu SBUS je znázorněna na Obr. 9.

byte		L						M			
1	st	1	1	1	1	0	0	0	0	p	sp	sp
2	st									p	sp	sp
3	st									p	sp	sp
4	st									p	sp	sp
5	st									p	sp	sp
6	st									p	sp	sp
7	st									p	sp	sp
8	st									p	sp	sp
9	st									p	sp	sp
10	st									p	sp	sp
11	st									p	sp	sp
12	st									p	sp	sp
13	st									p	sp	sp
14	st									p	sp	sp
15	st									p	sp	sp
16	st									p	sp	sp
17	st									p	sp	sp
18	st									p	sp	sp
19	st									p	sp	sp
20	st									p	sp	sp
21	st									p	sp	sp
22	st									p	sp	sp
23	st									p	sp	sp
24	st	b1	b2	fl	fs	0	0	0	0	p	sp	sp
25	st	0	0	0	0	0	0	0	0	p	sp	sp

Legenda:
st – start bit
P – paritní bit
sp – stop bit
L – LSB
M – MSB
b1 – 1. birnární kanál
b2 – 2. binární kanál
fl – příznak „frame lost“
fs – příznak „failsafe“

Obr. 9 Struktura rámce SBUS

Barevně jsou rozlišeny bity náležející k jednotlivým řídicím kanálům



Obr. 10 Rámec protokolu SBUS

Výhodami použití protokolu jsou možnost přenášet až 16 klasických kanálů, pouze jeden signální vodič, rychlost přenosu (vyslání rámce trvá 3 ms), a indikace ztráty signálu (fail-safe).

3.4 IBUS

IBUS je sériovým protokolem vyvinutým společností Flysky. Stejně jako SBUS využívá standardní sériové komunikace UART, ovšem s odlišnými parametry.

Tab. 2 Parametry sériové komunikace iBUS

Parametr	Hodnota
Napěťové úrovně	standardní
Modulační rychlost	115 200 baudů
Počet datových bitů	8
Parita	žádná
Počet stop bitů	2
Velikost rámce	32 bytů

Rámce protokolu iBUS jsou tvořeny 32 byty dat. První byte má vždy hodnotu 32 (0x20), a druhý 64 (0x40). Hodnoty kanálů jsou přenášeny v následujících 28 bytech. Celkem je možno přenést hodnoty až 14 kanálů – pro každý kanál jsou vyhrazeny dva celé byty. Pořadí bytů je little-endian, tedy nižší byte je vysílán první. Pro ověření neporušenosti dat slouží dva byty kontrolního součtu na konci rámce, jejichž hodnota je získána odečtením součtu hodnot všech kanálů od konstanty 65535 (0xFFFF) [5].

Struktura rámce protokolu iBUS je znázorněna na Obr. 11. Na Obr. 12 je zobrazena část zachyceného rámce iBUS v detailu.

byte		L						M		
1	st	0	0	0	0	0	1	0	0	sp	sp
2	st	0	0	0	0	0	0	1	0	sp	sp
3	st									sp	sp
4	st									sp	sp
5	st									sp	sp
6	st									sp	sp
7	st									sp	sp
8	st									sp	sp
9	st									sp	sp
10	st									sp	sp
11	st									sp	sp
12	st									sp	sp
13	st									sp	sp
14	st									sp	sp
15	st									sp	sp
16	st									sp	sp
17	st									sp	sp
18	st									sp	sp
19	st									sp	sp
20	st									sp	sp
21	st									sp	sp
22	st									sp	sp
23	st									sp	sp
24	st									sp	sp
25	st									sp	sp
26	st									sp	sp
27	st									sp	sp
28	st									sp	sp
29	st									sp	sp
30	st									sp	sp
31	st	k	k	k	k	k	k	k	k	sp	sp
32	st	k	k	k	k	k	k	k	k	sp	sp

Legenda:
st – start bit
sp – stop bit
L – LSB
M – MSB
k – konrolní součet

Obr. 11: Struktura rámce IBUS

Barevně jsou odlišeny bity náležející k jednotlivým řídicím kanálům

Obr. 12: Rámec protokolu IBUS

Stejně jako u protokolu SBUS je výhodou přenos vysokého počtu kanálů pomocí jediného signálního vodiče. Oproti předchozímu jmenovanému protokolu se však iBUS odlišuje jednodušší implementací, nevyžaduje podporu inverzních napěťových úrovní, a používá standardní modulační rychlost.

3.5 Další možnosti komunikace

Uvedený výčet není vyčerpávající, existuje celá řada komunikačních protokolů, které nebyly jmenovány.

Mezi další používané protokoly řídicích jednotek a radiopřijímačů patří :

- XBUS, vyvíjen společností JR Americas
- MultiWii Serial Protocol (MSP)

- SUMD a SUMH, vyvíjeny společností Graupner
- CRSF, vyvíjen společností TBS
- FPort, vyvíjen společností FrSky[6]

Některé radiopřijímače, zejména ty integrované do řídicích jednotek mohou komunikovat i pomocí rozhraní SPI.[7]

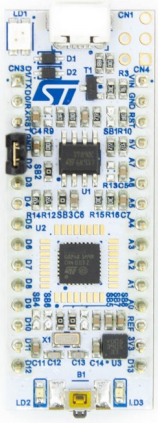
Existují také komunikační protokoly zvlášť pro telemetrická data. Příkladem může být protokol **Smart Port** společnosti FrSky, kterým může být také propojena řídicí jednotka a radiopřijímač, data jsou však posílána opačným směrem, než při přenosu řídicích kanálů.

4 HARDWARE, SOFTWARE A NÁSTROJE PRO REALIZACI PRAKTICKÉ ČÁSTI

4.1 Vývojová deska Nucleo-32 s mikrokontrolérem STM32L432KC

Nucleo-L432KC je vývojová deska z řady Nucleo-32, kterou vyvíjí společnost STMicro-electronics pro účely testování a vývoje prototypů produktů s MCU řady STM32. Vývojové desky Nucleo-32 obsahují následující komponenty:

- vlastní mikrokontrolér
- vestavěný debugger ST-LINK
- napájecí zdroje pro mikrokontrolér a debugger
- spínač pro reset cílového MCU
- krystalový oscilátor pro generování hodinového signálu
- LED indikace napájení, LED indikace stavu debuggeru, a LED ovladatelná z cílového MCU.
- 2 řady 16 pinů pro připojení periferií a napájení k MCU
- USB micro-B port pro připojení k PC



Obr. 13: Nucleo-L432KC[17]

4.1.1 STM32L432KC

MCU STM32L432KC je mikrokontrolér z produktové řady L, zaměřené na aplikace, kde je důležitá nízká spotřeba energie. Jádrem mikrokontroléru je procesor ARM Cortex M4 s 32-bitovou délkou slova, s matematickým koprocesorem (FPU), a maximální taktovací frekvencí 80MHz. Mikrokontrolér dále obsahuje 256 kB flash paměti a 64 kB operační paměti.[8]

Periferie a subsystémy MCU zahrnují:

- **Časovače:**
  Celkem 11 časovačů, 6 16-bitových, z toho 2 pro všeobecné použití, 2 základní, a 2 s nízkou spotřebou; dále jeden 16-bitový pro řízení motorů a jeden 32-bitový pro všeobecné použití; a také 2 watchdog časovače, a SysTick časovač.

- **14-kanálový DMA řadič**

- **Generátor náhodných čísel**

- **Akcelerátor CRC**

- **Analogové periferie:**
  12-bitový ADC s 10 kanály, 2 12-bitové DAC, operační zesilovač, 2 komparátory s velmi nízkou spotřebou.

- **Komunikační rozhraní:**
  USB 2.0, SAI (sériové audio rozhraní), 2 rozhraní I^2C, 3 rozhraní USART, 1 roz-hraní UART s nízkou spotřebou, 2 SPI, rozhraní sběrnice CAN, SWPMI (single wire protocol), IRTIM (infračervené rozhraní)

- **Hodiny reálného času (RTC)**

- **Vstupy/výstupy:**
  až 26 vstupů/výstupů, většina toleruje napětí 5V[8]

4.1.2 Možnosti programování vývojových platforem STM32

Rozšířenost platformy STM32, umožňuje ubírat se při tvorbě software několika různými směry. Obvyklými jazyky pro tvorbu zdrojového kódu aplikace jsou tradičně jazyky C a C++, případně jazyk symbolických adres. Existují však také projekty umožňující programování mikrokontrolérů STM32 v dalších jazycích, jako například MicroEJ (Java), eLua (Lua), MicroPython (Python). Také prostředí MATLAB/Simulink umožňuje automatické generování zdrojového kódu pro tuto platformu. [9, 10]

Pro jazyky C a C++ existuje více úrovní abstrakce od cílového hardware, mezi kterými je možno při vývoji aplikace volit. Nejnižší úrovní je přímý přístup k registrům a periferiím MCU, za pomocí hlavičkových souborů definujících adresy registrů, a referenčního manuálu daného produktu. Tato úroveň umožňuje nejvyšší míru kontroly nad cílovým hardware, ale tvorba programu je náročná a vytvořený zdrojový kód nemusí být snadno přenositelný. Další úrovní je využití vrstvy Low Level API (LL API), která definuje základní funkce pro práci s periferiemi. Tato vrstva překrývá přímý přístup k registrům, stále je však potřeba znalost funkce a struktury periferií. Vyšší míru abstrakce než LL poskytuje vrstva HAL (Hardware Abstraction Layer), která také definuje funkce pro práci s periferiemi a díky vysoké úrovni abstrakce od hardware umožňuje přenositelnost vytvořeného kódu například mezi MCU stejné rodiny.[11, 12]

Nejvyšší míru abstrakce nabízejí operační systémy pro embedded systémy, jako například Mbed OS, které zahrnují knihovny s ovladači pro standardní periferie i externí zařízení, a zcela tak izolují vývojáře od nízkoúrovňové práce s periferiemi.

4.2 Řídicí jednotka Seriously Pro Racing F3 Acro

Jedná se o řídicí jednotku klasické konstrukce s rozměry 30x30 mm. Výpočetní výkon obstarává mikrokontrolér STM32F303. Inerciální měřící jednotka je MPU 6050. Vstupní a výstupní rozhraní zahrnují (mimo jiné):

- 8 vstupů PWM
- 8 výstupů PWM
- 3 hardwarová sériová rozhraní

- I^2C rozhraní
- USB port[13]

Řídicí je jednotka je popsána na obrázku (Obr. 6) v kapitole 2. Na rozdíl od výše zobrazené verze neobsahuje verze Acro barometr a magnetometr.

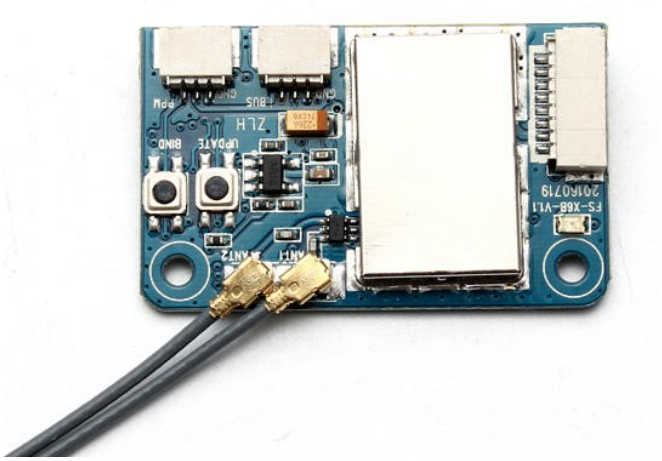
4.3 Betaflight

Betaflight je open-source komunitně vyvíjený firmware pro řídicí jednotky dronů. Podporujecelou řadu řídicích jednotek a umožňuje využití široké škály součástí pro konstrukci dronů.

Při tvorbě této práce byla v řídicí jednotce SP Racing F3 Acro použita verze Betaflight 3.5.0 .

4.4 Radiopřijímač FlySky X6B

FlySky X6B je kompaktní radiopřijímač o rozměrech 36x22 mm. Výstupní signál je možno volit mezi PWM, PPM a protokolů SBUS a iBUS. Výběr protokolu je možný ze spárovaného vysílače.

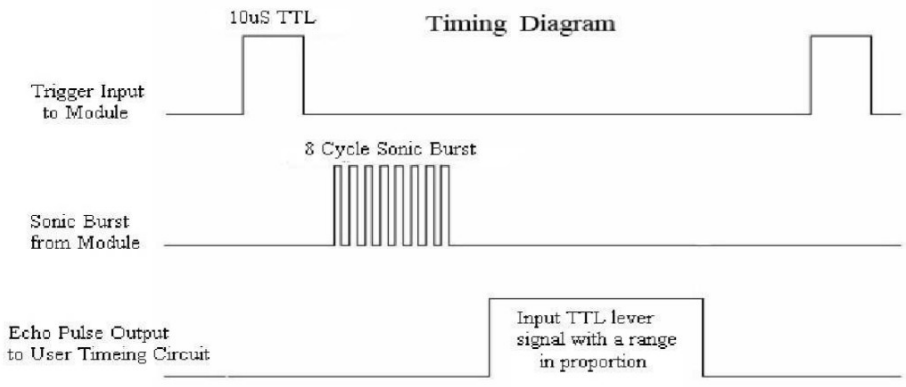


Obr. 14: Radiopřijímač FlySky X6B[23]

4.5 Senzor vzdálenosti HC-SR04

Modul HC-SR04 je levný senzor vzdálenosti pracující na principu ozvěny. Senzor vysílá modulovaný ultrazvukový signál a detekuje jeho návrat po odrazu od překážky. Mezi vysláním a návratem signálu je výstupní pin Echo ve stavu logické 1, jinak ve stavu logické 0. Délka výstupního kladného pulzu je tedy funkcí vzdálenosti senzoru od překážky. Měření je vyvoláno přítomností alespoň 10 µs dlouhého kladného impulzu na vstupním pinu Trig. Na obrázcích níže je modul HC-SR04 a časový průběh signálů při měření.

Obr. 15: Ultrazvukový senzor HC-SR04[24]

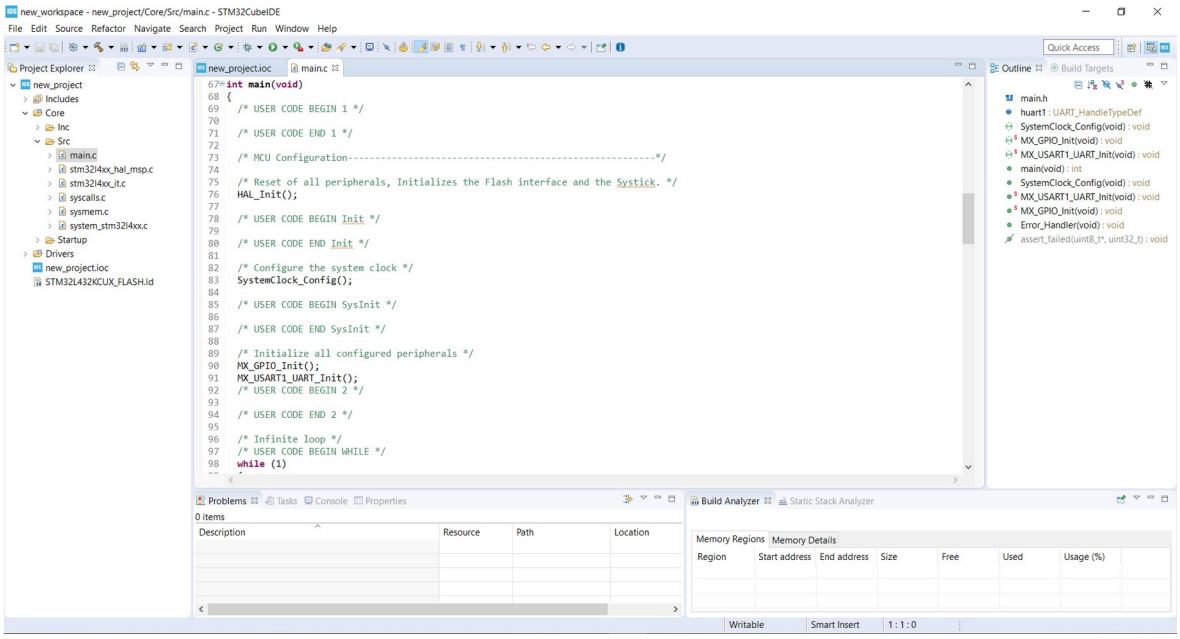


Obr. 16: Časový průběh signálů při měření vzdálenosti[22]

4.6 STM32 CubeIDE

STM32 CubeIDE je vývojové prostředí určené pro vývoj aplikací určených pro produkty řady STM32 společnosti STMicroelectronics. Prostředí je založeno na vývojovém fra-

meworku Eclipse a GCC toolchainu. Integruje také samostatně dostupný nástroj CubeMX, který poskytuje grafické rozhraní pro konfiguraci jádra, periferií, hodinových signálů a spotřeby cílového MCU. CubeMX podle vytvořené konfigurace generuje zdrojový kód obsahující inicializaci všech potřebných subsystémů, který slouží jako základ vyvíjené aplikace. CubeIDE také obsahuje standardní nástroje pro ladění software.



Obr. 17: Grafické rozhraní vývojového prostředí STM32CubeIDE

4.7 Betaflight Configurator

Betaflight Configurátor je softwarový nástroj pro osobní počítače určený ke konfiguraci instalací firmware Betaflight na řídicích jednotkách. Nástroj umožňuje z připojeného PC provádět na cílové řídicí jednotce tyto akce:

- **Nahrání firmware Betaflight**

  Na záložce Firmware Flasher na úvodní straně programu je možné vybrat předkompilovaný binární soubor pro danou řídicí jednotku, který je následně automaticky stažen z internetového repozitáře a nahrán do mikrokontroléru jednotky.

- **Konfiguraci**

  Nástroj poskytuje velmi detailní nastavení všech důležitých parametrů řídicích jednotek, jako jsou konfigurace dronu (počet, umístění a směr rotace motorů), letových módů, komunikační protokol a rozhraní radiopřijímače, telemetrie, nastavení a kalibrace IMU, nastavení OSD, GPS, baterie, PID regulátorů atd.

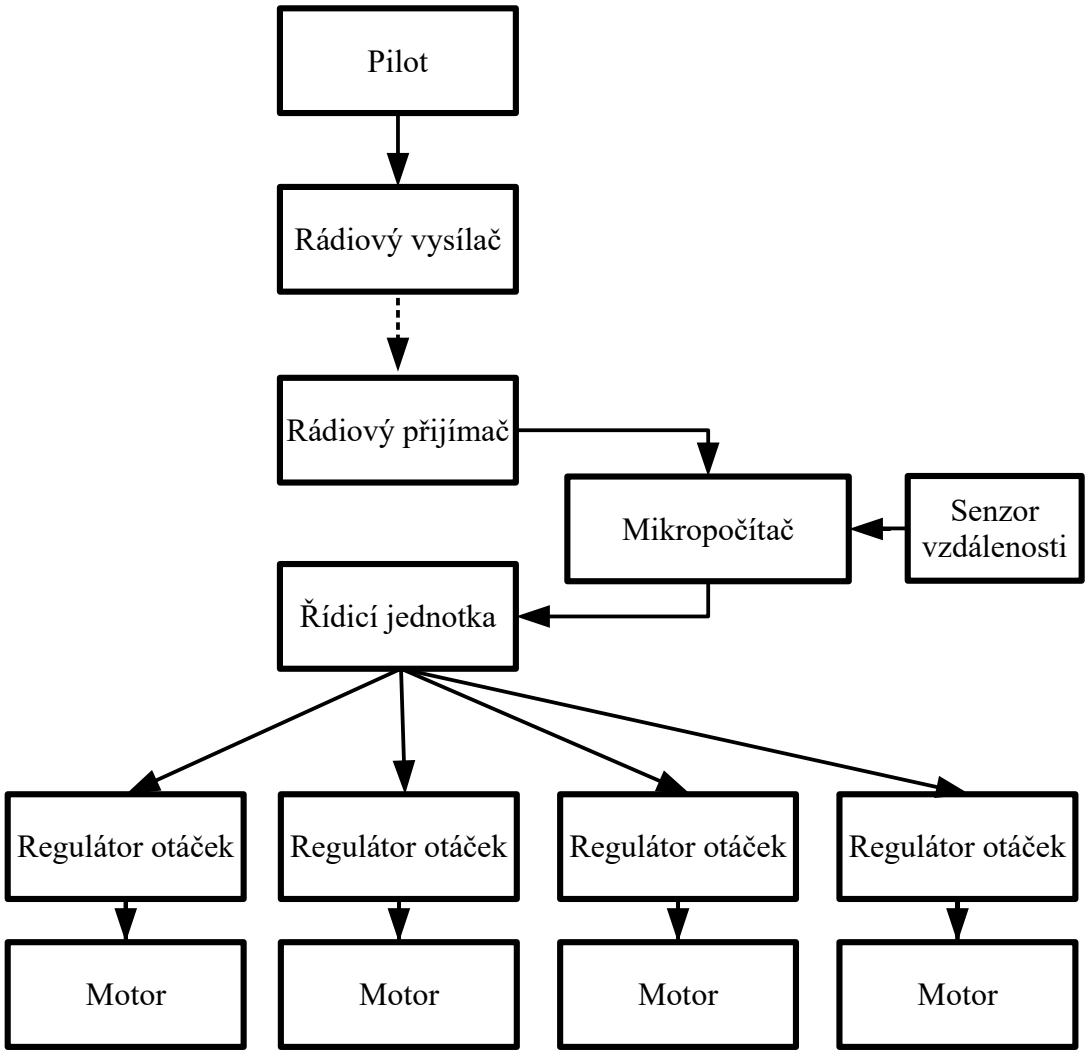
- **Ladění**

  Některé funkce konfigurátoru umožňují efektivně vyhledat chyby v konfiguraci, či v hardware dronu. Jedná se zejména o záložky Receiver, Motors, CLI a Blackbox. Záložka Receiver zobrazuje v reálném čase hodnoty přijímaných kanálů z radiopřijímače. V záložce Motors je možné manuálně nastavovat otáčky jednotlivých motorů a sledovat časový průběh hodnot úhlové rotace dronu. Záložka CLI obsahuje interaktivní příkazový řádek, pomocí kterého je možné provádět nastavení a diagnostiku řídicí jednotky příkazy. V záložce Blackbox je možné nastavit zaznamenávání letových údajů a také již zaznamenané údaje v paměti jednotky uložit do PC.

II. PRAKTICKÁ ČÁST

5 NÁVRH SYSTÉMU

Účelem navrhovaného systému je umožnit automatické udržování výšky kvadrokoptéry nad zemí. K uskutečnění tohoto cíle musí být mikropočítač se senzorem vzdálenosti vložen do řetězce ovládání mezi radiopřijímač a řídicí jednotku dronu – viz obrázek (Obr. 18). Tedy mikropočítač musí přijímat komunikaci z radiopřijímače a sám odesílat zprávy řídicí jednotce. Pomocí jednoho z řídicích kanálů (ovládaného přepínačem), bude indikováno, zda má být režim udržování výšky aktivní. Když bude režim aktivní mikropočítač bude na základě výšky změřené senzorem vzdálenosti regulovat hodnotu kanálu pro výkon motorů. K regulaci bude využit diskrétní PID regulátor. Dalším z řídicích kanálů (ovládaným potenciometrem) bude možné měnit aktuální požadovanou výšku ve vhodném rozmezí měřícího rozsahu senzoru.



Obr. 18: Začlenění navrhovaného systému do řetězce řízení dronu

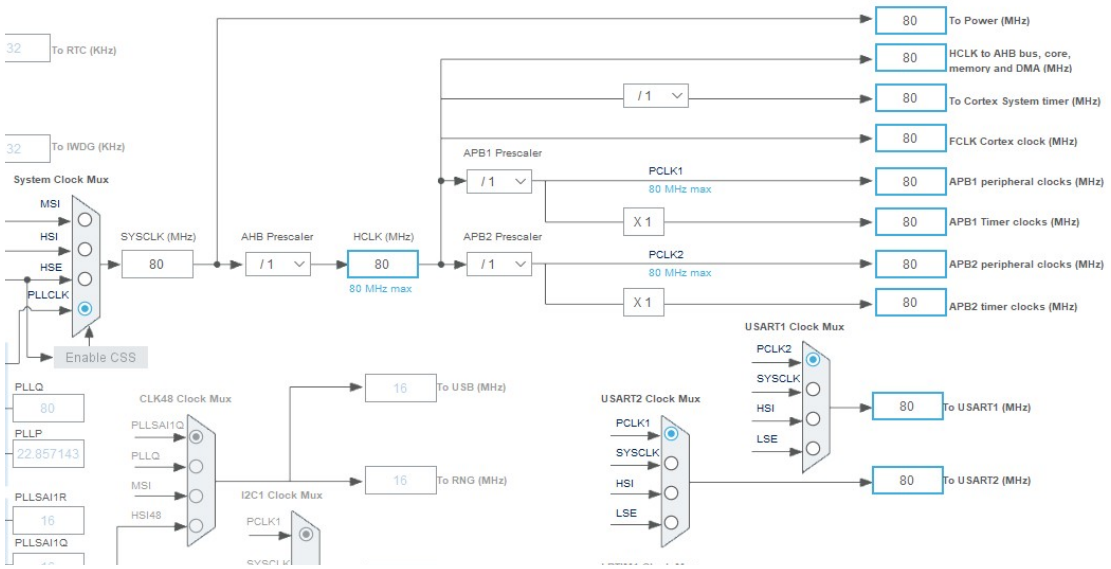
6 REALIZACE PROGRAMU MIKROPOČÍTAČE

6.1 Vytvoření projektu a konfigurace MCU

Pomocí CubeIDE byl vytvořen nový projekt pro platformu STM32. Jako cílová platforma byla v průvodci, v záložce Board Selector, zvolena vývojová deska **NUCLEO-L432KC**. Jazykem projektu byl zvolen **C++** za účelem využití principů objektově orientovaného programování při tvorbě softwarových komponent aplikace. Z důvodu osobního zájmu o porozumění detailů vnitřního fungování MCU řady STM32 jsem se rozhodl využít při tvorbě aplikace výrobcem poskytovaných knihoven hardwarové abstrakční vrstvy (HAL), jako kompromisu mezi rychlostí vývoje a kontrolou nad detaily implementace. Ostatní nastavení projektu byla ponechána na výchozích hodnotách. Následně byly vestavěným nástrojem CubeMX nejdříve inicializovány všechny periferie MCU na výchozí hodnoty, a poté postupně konfigurovány pro potřeby aplikace. Pro obsluhu senzoru vzdálenosti HC-SR04 je využit časovač **TIM1**. Pro komunikaci s radiopřijímačem a řídicí jednotkou dronu je použito rozhraní **USART1**. Konfigurace jednotlivých součástí MCU je rozepsána dále.

6.1.1 Konfigurace hodinových signálů

Frekvence hodinového signálu jádra a periferií byla nastaven na maximální možnou hodnotu, tj. 80 MHz, pro maximalizaci výkonu MCU. Úspora energie dosažitelná snížením této frekvence je pro navrhovanou aplikaci nepodstatná. Výřez z diagramu nastavení frekvencí hodinových signálů zahrnující nastavení hodinových signálů jádra a periferií MCU je na Obr. 19.



Obr. 19: Nastavení hodinových frekvencí jádra a periferií STM32L432KC

6.1.2 Konfigurace časovače TIM1

Prvním upraveným parametrem časovače je hodnota děličky frekvence vstupního hodinového signálů. Pro účely generování spouštěcího pulzu pro senzor HC-SR04 a měření délky vráceného impulzu je postačující, když bude časovač pracovat s přesností 1 µs, tedy frekvence inkrementace jeho čítače musí být 1 MHz. Z toho vyplývá, že vstupní hodinový signál 80 MHz je nutné dělit 80.

Dalším důležitým parametrem je perioda čítače. Perioda je 16-bitová hodnota po jejímž dosažení při čítání nahoru se čítač vynuluje. Při dosažení maximální měřitelné vzdálenosti senzoru HC-SR04, tj. 4 m, odpovídá doba letu vyslaného zvukového signálu (za předpokladu přibližné rychlosti zvuku ve vzduchu 340m/s):

$$t = \frac{s}{v} = \frac{4 \cdot 2}{340} \approx 23{,}5\,ms \tag{1}$$

kde **t** je doba letu zvukového signálu, **s** je dráha letu signálu (dvojnásobek vzdálenosti) a **v** je rychlost zvuku ve vzduchu. Skutečná doba potřebná pro změření výšky 4 m je vyšší, neboť určitý čas zabere vyslání zvukového signálu senzorem po přijetí spouštěcího impulzu. Délka této časové prodlevy však není výrobcem stanovena a při zběžném otestování několika senzorů bylo zjištěno, že se pro každý kus liší. Výrobcem doporučená minimální délka periody meření 60 ms je příliš dlouhá a mohla by negativně ovlivnit zamýšlený regulační proces. Po uvážení těchto skutečností byla zvolena pevná perioda měření **25 ms** za cenu mírného snížení maximální měřitelné vzdálenosti. Případné překročení této perio-

dy impulzem vráceným ze senzoru je ošetřeno v software. Převedením zvolené periody na mikrosekundy získáme hodnotu pro nastavení příslušného registru pro periodu časovače – **25 000**. Ostatní obecná a pokročilá nastavení časovače jsou ponechána na výchozích hodnotách.

Pro první kanál časovače TIM1 je nastavena funkce PWM. Režim PWM je nastaven na PWM mode 1, při kterém je výstup kanálu aktivní, pokud je hodnota časovače nižší než hodnota v porovnávacím registru. Hodnota porovnávacího registru je nastavitelná parametrem „Pulse“. Pro generování spouštěcího impulzu měření o délce 10 µs, je tedy tento parametr nastaven na hodnotu **10**. Ostatní parametry pro první kanál jsou ponechány na výchozích hodnotách.

Pro čtvrtý kanál časovače TIM1 je nastavena funkce zachycení vstupu. Zde je pouze nutné, aby byla polarita zachycení nastavena na vzestupnou hranu signálu, ostatní parametry zůstávají na výchozích hodnotách. Shrnutí důležitých parametrů nastavení časovače TIM1 je v tabulce 3.

Tab. 3: Shrnutí konfigurace časovače TIM1

Parametr	Označení v CubeIDE	Hodnota	Označení v CubeIDE
Nastavení časovače			
dělička	Prescaler	80	
směr čítání	Counter Mode	nahoru	Up
perioda	Counter Period	25000	
Nastavení kanálu 1			
funkce		PWM	PWM Generation CH1
režim PWM	PWM Mode	PWM mode 1	
délka impulzu	Pulse	10	
Nastavení kanálu 4			
funkce		zachycení vstupu	Input Capture direct mode
polarita	Polarity Selection	vzestupná hrana	Rising Edge

6.1.3 Konfigurace sériového rozhraní USART1

Rozhraní USART1 je využito aplikací k příjmu sériové komunikace protokolu SBUS z radiopřijímače a také posílání zpráv řídicí jednotce. Rozhraní je tedy nastaveno tak, aby odpovídalo parametrům protokolu SBUS. V první řadě je režim nastaven na asynchronní. Modulační rychlost je nastavena na **100000 baudů**, délka slova včetně parity na **9 bitů**, parita je **sudá**, a slovo je ukončeno **2 stopbity**. V pokoročilých funkcích rozhraní je dále

nastavena inverze napěťových úrovní sběrnice, a také je zakázáno přetečení. Přetečení je situace, kdy je připraveno uložení nově přijatých dat do registru přijatých dat, který ještě nebyl přečten. Tato funkce je zakázána neboť vyvolává chybu při synchronizaci rozhraní. Zbylé parametry jsou ponechány na výchozích hodnotách.

V záložce nastavení DMA jsou dále nastaveny dva kanály DMA pro výměnu dat s rozhraním USART, přičemž režim kanálu pro přijatá data je nastaven pro opakovaný přenos.

Tab. 4: Shrnutí konfigurace sériového rozhraní USART1

Parametr	Označení v CubeIDE	Hodnota	Označení v CubeIDE
režim činnosti	Mode	asynchronní	Asynchronous
modulační rychlost	Baud rate	100000	
délka slova	Word Length	9 bitů	9 bits
parita	Parity	sudá	Even
počet stopbitů	Stop Bits	2	
inverze napěťových úrovní vysílače	TX Pin Active Level Inversion	povoleno	Enable
inverze napěťových úrovní přijímače	RX Pin Active Level Inversion	povoleno	Enable
detekce přetečení	Overrun	zakázáno	Disable
Nastavení DMA kanálu přijímače			
režim	Mode	opakovaný přenos	Circular
Nastavení DMA kanálu vysílače			
režim	Mode	jednorázový přenos	Normal

6.1.4 Konfigurace sériového rozhraní USART2

Rozhraní USART2 je aktivováno s výchozími parametry pouze s funkcí vysílání pro potřeby ladění aplikace. Toto rozhraní je skrze ladící nástroj ST-LINK připojeno k virtuálnímu konzolovému portu. Pro vysílač USART2 je také nastaven jeden DMA kanál v režimu jednorázového přenosu.

6.1.5 Konfigurace řadiče přerušení NVIC

V záložce nastavení řadiče přerušení jsou nastaveny priority pro jednotlivé zdroje přerušení tak, že systémová přerušení mají nejvyšší prioritu, poté následují přerušení od nastavených

kanálů DMA, následně přerušení od časovače TIM1, a nakonec globální přerušení rozhraní USART. Celá tabulka priorit přerušení je na Obr. 20.

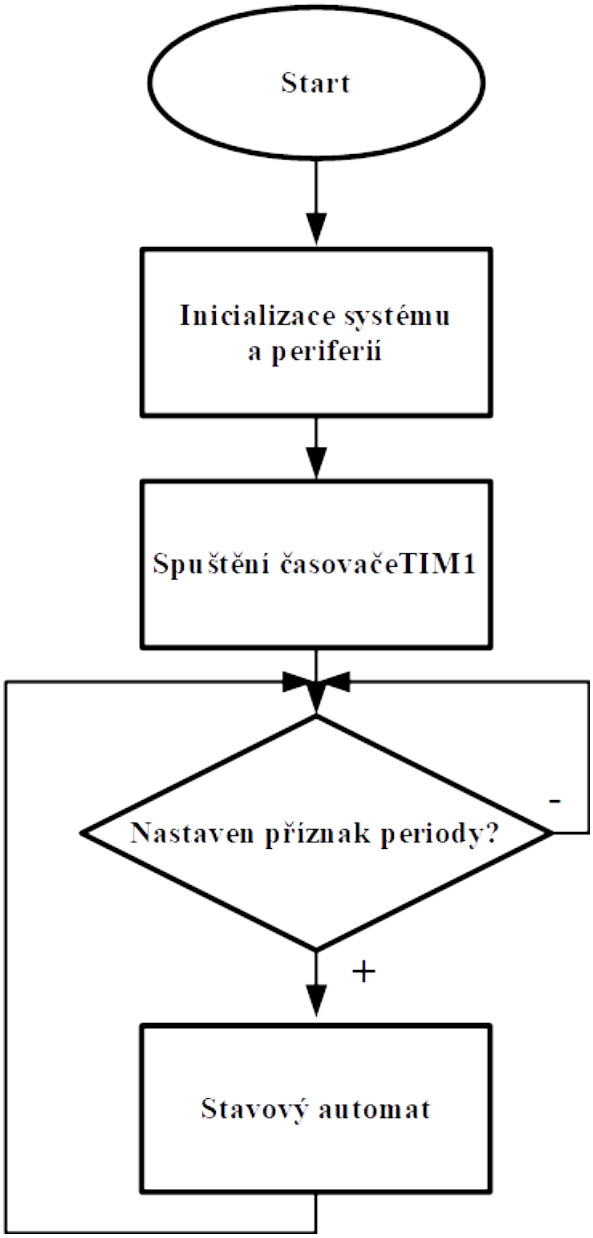
NVIC Interrupt Table	Enabled	Preemption Priority	Sub Priority
Non maskable interrupt	☑	0	0
Hard fault interrupt	☑	0	0
Memory management fault	☑	0	0
Prefetch fault, memory access fault	☑	0	0
Undefined instruction or illegal state	☑	0	0
System service call via SWI instruction	☑	0	0
Debug monitor	☑	0	0
Pendable request for system service	☑	0	0
Time base: System tick timer	☑	0	0
PVD/PVM1/PVM2/PVM3/PVM4 interrupts through EXTI lines 16/35/36/37/38	☐	0	0
Flash global interrupt	☐	0	0
RCC global interrupt	☐	0	0
DMA1 channel4 global interrupt	☑	2	0
DMA1 channel5 global interrupt	☑	2	0
DMA1 channel7 global interrupt	☑	2	0
TIM1 break interrupt and TIM15 global interrupt	☐	0	0
TIM1 update interrupt and TIM16 global interrupt	☑	4	0
TIM1 trigger and commutation interrupts	☐	0	0
TIM1 capture compare interrupt	☑	4	0
USART1 global interrupt	☑	6	0
USART2 global interrupt	☑	6	0
FPU global interrupt	☐	0	0

Obr. 20: Tabulka priorit přerušení

Poznámka: Jádro ARM Cortex-M používá systém číslování priorit, při kterém mají přerušení s nižší číslicí vyšší prioritu než přerušení s vyšší číslicí

6.2 Hlavní část programu

Hlavní část programu je umístěna v souboru main.cpp. V tomto souboru je definována hlavní funkce main(), ve které je nejprve provedena inicializace systému a periferií, podle zdrojového kódu vygenerovaného nástrojem CubeMX, a dále je prováděna nekonečná smyčka s kódem aplikace. Obecný průběh programu je znázorněn v diagramu na Obr. 21. Dalšími funkcemi definovanými v tomto souboru jsou funkce pro obsluhu přerušení časovače TIM1 a sériového rozhraní USART1.

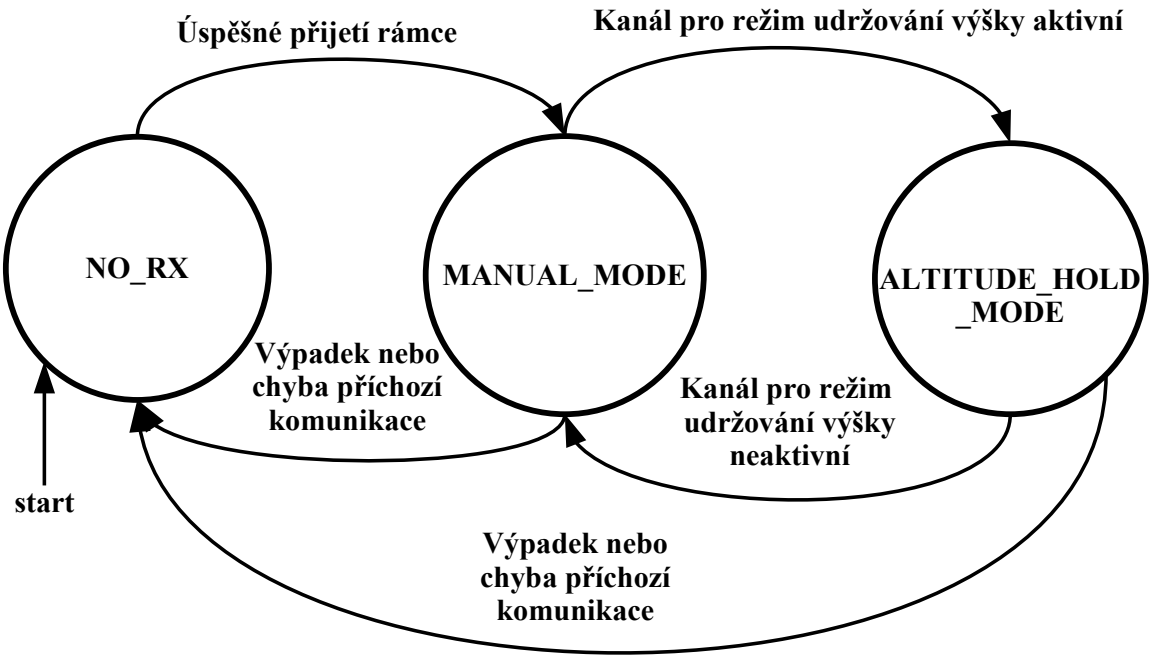


Obr. 21: Vývojový diagram hlavní části programu

6.2.1 Funkce main()

Po inicializaci systému a periferií jsou aktivovány oba nastavené kanály časovače TIM1. Pro kanál č. 4 jsou povolena přerušení. Následně je časovač spuštěn s povoleným přerušením při uplynutí periody. Po tomto kroku vstupuje program do nekonečné smyčky while, ve které čeká na příznak uplynutí periody časovače. Když je příznak nastaven, provede se aplikační kód, příznak je vyčištěn, a program čeká než uplyne další perioda. Činnost aplikačního kódu lze vyjádřit stavovým automatem, jehož stav je uložen v proměnné **state**. Tato proměnná je definována jako vlastní datový typ ProgramMode. ProgramMode v sobě

definuje jedinou položku datového typu enum, která může nabývat tří hodnot – NO_RX, MANUAL_MODE a ALTIUTDE_HOLD_MODE. Diagram stavového automatu je na Obr. 22.



Obr. 22: Stavový automat hlavní funkce main()

Při prvním provedení kódu v hlavní smyčce je program ve stavu **NO_RX**. V tomto stavu se pokusí o příjem komunikace z radiopřijímače. K tomu slouží pomocná funkce Synchronize_UART (popsána dále), a následné volání funkce HAL_UART_Receive_DMA, která zahájí automatický příjem zpráv a jejich přenos na určenou adresu v paměti pomocí DMA. Je-li příjem zprávy úspěšný, je vyvoláno příslušné přerušení, které je obslouženo funkcí HAL_UART_RxCpltCallback. Tím přechází program do režimu ručního řízení. Protože DMA kanál pro příjímač rozhraní USART1 pracuje v režimu opakovaného přenosu, není již potřeba opětovné volání funkce HAL_UART_Receive_DMA, každá další příchozí zpráva je automaticky přenesena na určenou adresu v paměti.

Při ručním řízení (**MANUAL_MODE**) je nejprve testováno, zda byl od posledního konce periody přijat nový rámec protokolu SBUS. Tato podmínka je za normálních okolností splněna, neboť radiopřijímač vysílá nový rámec přibližně každých **7 ms**. Nedošlo-li k přijetí nového rámce, přechází program do stavu NO_RX. V opačném případě je přijatý rámec zkopírován do pomocné proměnné, aby nemohlo dojít při případném přijetí dalšího rámce k přepsání dat uprostřed následných operací. Rámec je poté dekódován, aby byly získány

hodnoty kanálů. Pokud je hodnota kanálu pro aktivaci režimu udržování zadané výšky vyšší než střední hodnota kanálu SBUS, přechází program do stavu ALTIUTDE_HOLD_MODE. Kopie přijatého rámce je nezměněna odvysílána vysílačem rozhraní USART1 řídicí jednotce.

Ve stavu udržování výšky (**ALTITUDE_HOLD_MODE**) je přijat a zkopírován příchozí rámec stejně jako při ručním řízení. Pokud je kanál pro spuštění režimu udržování výšky neaktivní, přechází program do režimu ručního řízení. Dále je proveden výpočet PID regulátoru a výsledná akční veličina je převedena na hodnotu kanálu výkonu motorů kvadrokoptéry. Následně je upravený kanál spolu s ostatními zakódován zpět do rámce protokolu SBUS a tento je odvysílán do řídicí jednotky dronu.

6.2.2 Funkce HAL_TIM_IC_CaptureCallback()

Tato funkce obsluhuje přerušení zachycení vstupu časovače. Jediným úkonem vykonaným touto funkcí je zavolání metody edgeDetected() objektu ultrazvukového senzoru.

6.2.3 Funkce HAL_TIM_PeriodElapsedCallback()

Obslužná rutina přerušení při uplynutí periody časovače má v programu důležitou funkci. Nastavuje příznak periody, podle kterého je vykonáván kód v hlavní smyčce a zpracovává změřenou hodnotu vzdálenosti. Zpracování probíhá ve dvou krocích: Prvním krokem je omezení odlehlých hodnot a druhým průchod přes jednorozměrný digitální filtr. K těmto operacím jsou využívány metody třídy Filter. Na konci rutiny je zavolána metoda periodElapsed() objektu ultrazvukového senzoru.

6.2.4 Funkce HAL_UART_RxCpltCallback()

Tato funkce obsluhuje přerušení při úspěšném přijetí rámce sériovým rozhraním. Při prvním úspěšně přijatém rámci změní stav programu v hlavní smyčce na režim ručního řízení. Také nastavuje příznak přijetí nového rámce při každém úspěšném přenosu.

6.2.5 Funkce HAL_UART_Error_Callback()

Tato funkce obsluhuje přerušení při chybě v příchozí sériové komunikaci. Pokud je tato funkce volána když je program ve stavu ručního řízení, nebo režimu udržování výšky, přechází program do stavu bez rádiového spojení (NO_RX).

6.2.6 Funkce Synchronize_UART()

Slouží k synchronizaci s proudem přijímaných rámců na sériovém rozhraní. Funkce čeká určitý čas na nastavení příznaku volné linky rozhraní USART. Při nastavení příznaku vrací HAL_OK, jinak po uplynutí času HAL_TIMEOUT. Čekání na nastavení příznaku je indikováno vestavěnou LED desky Nucleo.

6.3 Třída UsoundDistSensor

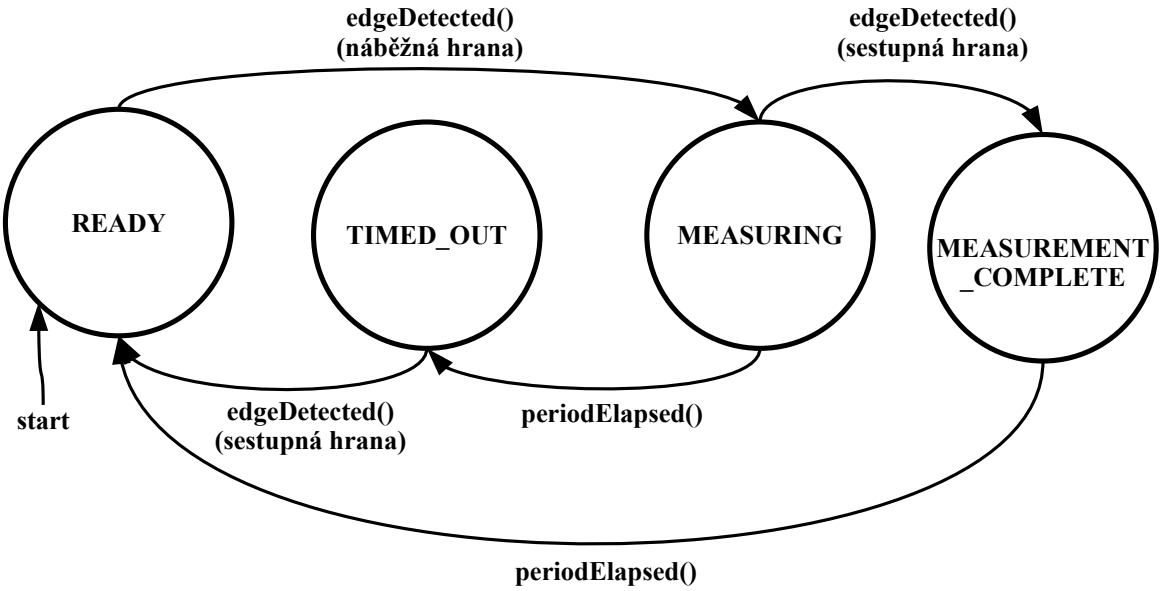
Tato třída slouží k reprezentaci ultrazvukového senzoru vzdálenosti HC-SR04 v programu.

Tab. 5: Atributy a metody třídy USoundDistSensor

atribut	popis
TIM_HandleTypeDef *timer	ukazatel na instanci časovače
unsigned int channel	kanál časovače
StateEnum state	Stav senzoru
uint16_t rising_edge	čas vzestupné hrany
uint16_t falling_edge	čas sestupné hrany
float distance	změřená vzdálenost
metoda	popis
TIM_HandleTypeDef * get_timer()	getter ukazatele časovače
StateEnum get_state()	getter stavu
float get_distance()	getter naměřené vzdálenosti
void edgeDetected()	obsluha náběžné hrany
void periodElapsed()	obsluha uplynutí periody časovače

Vytvořené metody mají za cíl zachytit čas náběžné a sestupné hrany impulzu ze senzoru (viz kapitola 4.5), a z těchto časů vypočítat vzdálenost překážky od senzoru. Stav senzoru je uložen v atributu state. Ten může nabývat čtyř hodnot: READY, MEASURING, MEASUREMENT_COMPLETE, a TIMED_OUT. Přechody mezi těmito stavy jsou popsány

stavovým diagramem na Obr. 23. Měření délky impulzu probíhá ve stavu MEASURING. Stav MEASUREMENT_COMPLETE značí úspěšné dokončení měření před koncem periody časovače, kdežto TIMED_OUT značí, že sestupná hrana nebyla před koncem periody detekována.



Obr. 23: Stavový automat objektu třídy USoundDistSensor

6.4 Třída SBUSProtocol

Tato třída poskytuje statické metody pro kódování a dekódování rámců protokolu SBUS, a související konstanty.

Tab. 6: Atributy a metody třídy SBUSProtocol

atribut	popis
static const int frame_length	délka rámce SBUS v bytech
static const int channels	počet kanálů v rámci SBUS
metoda	popis
static void decodeFrame(uint8_t * frame, uint16_t * channels)	dekódování rámce na kanály
static void encodeFrame(uint16_t * channels, uint8_t * frame)	kódování kanálů do rámce

6.4.1 Metoda decode

Pro dekódování hodnot kanálů je nutné vždy vypočítat ve kterém bytu rámce začínají bity daného kanálu a na jaké bitové pozici. Počáteční byte a tři za ním následující (protože

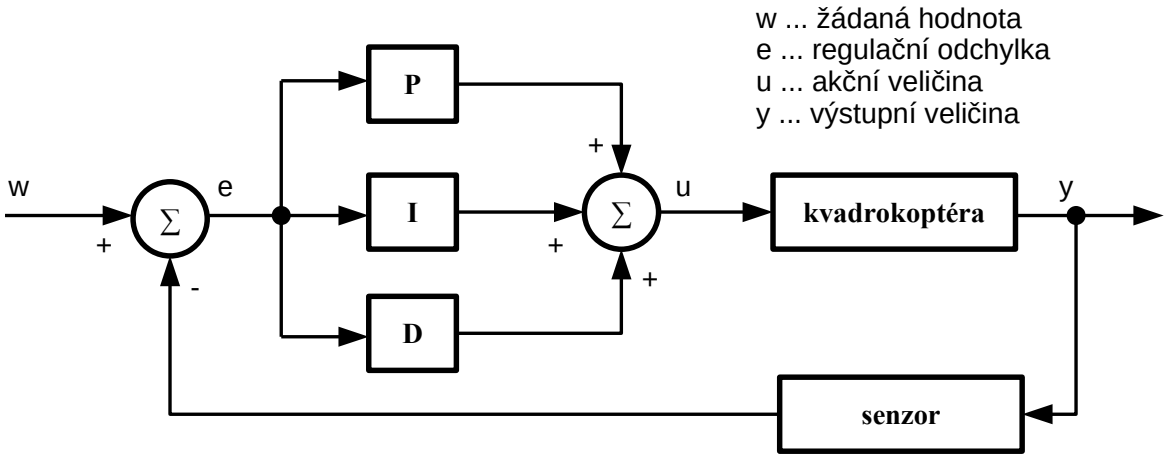
některé kanály mají bity rozprostřeny až do tří po sobě jdoucích bytů) jsou poté přetypovány na jediné 32-bitové celé číslo (uint32_t) a pomocí masky jedenácti bitů s hodnotou 1,posunutou na správnou pozici, jsou logickým součinem získány bity kanálu. Tyto bity jsou následně posunuty zpět na nultou pozici, výsledek je přetypován na 16-bitové celé číslo a uložen do pole hodnot kanálů.

6.4.2 Metoda encode

Při kódování kanálů do rámce SBUS je postupováno podobným způsobem, jako při dekódování, jen v opačném směru. Bity kanálu jsou posunuty na správnou pozici a logickým součtem jsou přidány k bitům dalších kanálů. Poté je přidán byte s příznaky a binárními kanály. Na začátek rámce je vložen byte s hodnotou 15 a na konec byte s hodnotou 0.

6.5 Třída PIDController

Třída PIDController implementuje navržený regulátor výšky. Regulátor má strukturu paralelního diskrétního PID.



Obr. 24: Struktura navrženého PID

Matematicky může být výstup regulátoru v čase **t = kT** vyjádřen rovnicí (2). Jedná se o rovnici přírůstkového PID regulátoru převzatou z výukové webové stránky Řízení technologických procesů. [14] Rovnice je upravena do formy s koeficienty zesílení jednotlivých složek regulátoru, namísto časových konstant.

$$u(kT)=u[(k-1)T]+K_p\{e(kT)-e[(k-1)T]\}+K_i e(kT)T+\frac{K_d}{T}\{e(kT)-2e[(k-1)T]+e[(k-2)T]\} \quad (2)$$

(podle [14])

Kde **u** je akční veličina, **k** je pořadí periody, **T** je doba periody, **e** je regulační odchylka, $\mathbf{K_p}$ je zesílení proporcionální složky, $\mathbf{K_i}$ je zesílení integrační složky a $\mathbf{K_d}$ je zesílení derivační složky.

Přírůstková forma regulátoru je výhodná zejména proto, že umožňuje plynulý přechod z režimu ručního ovládání do režimu udržování výšky. Při první iteraci regulátoru po spuštění je jako hodnota akční veličiny z předchozí iterace u[(k-1)T] nastavena hodnota vypočtená z posledního stavu kanálu výkonu motorů při ručním řízení. Nedochází tak k přechodnému výpadku tahu motorů.

Atributy a metody třídy PIDController jsou shrnuty v tabulce (Tab. 7). Kromě metody pro výpočet akční veličiny implementuje třída také metody pro nastavování zesílení složek regulátoru za chodu, které mohou být využity při ladění regulátoru.

Tab. 7: Atributy a metody třídy PIDController

atribut	popis
Kp	zesílení proporcionální složky
Ki	zesílení integrační složky
Kd	zesílení derivační složky
last_error	regulační odchylka z předchozí periody
last_last_error	regulační odchylka z druhé předchozí periody
metoda	**popis**
void set_Kp(float Kp)	setter Kp
void set_Ki(float Ki)	setter Ki
void set_Kd(float Kd)	setter Kd
float calculate(float error, float period)	kalkulace akční veličiny
void initialize()	inicializace (vynulování předchozích regulačních odchylek)

6.6 Třída Filter

Třída Filter definuje statické metody používané v některých funkcích k úpravě hodnot.

Tab. 8: Metody třídy Filter

metoda	popis
static float filter_1D(float previous_value, float new_value, float coef)	diskrétní dolní propust
template <typename t> Static t constrain(t value, t min, t max)	omezení vstupní hodnoty do zvoleného rozmezí

Metoda **filter_1D** implementuje jednoduchý diskrétní filtr typu dolní propust. Výstup filtru pro hodnotu z posloupnosti s pořadím **k** lze vyjádřit následovně (podle [15]):

$$y(k)=\beta\cdot x(k)+(1-\beta)\cdot y(k-1) \tag{3}$$

Kde **y** je výstup filtru, **x** je vstupní hodnota , a **β** je koeficient z intervalu <0 , 1> určující intenzitu filtrace. Čím nižší je hodnota **β** tím méně (a tím pomaleji) je výstup ovlivněn aktuálním vstupem. Metoda je využívána při filtraci naměřených hodnot ze senzoru vzdálenosti.

Druhá metoda třídy implementuje funkci omezení hodnot. Pokud je vstupní hodnota (první parametr) v intervalu mezi hodnotou druhého a třetího parametru, je vrácena beze změny. Je-li vyšší než horní mez, nebo nižší než dolní mez intervalu, je vrácena tato mez. Metoda je implementována jako šablona a lze ji proto použít pro různé číselné datové typy.

7 TESTOVÁNÍ A LADĚNÍ SYSTÉMU

Pro otestování funkce vytvořené aplikace byla vývojová deska Nucleo a senzor vzdálenosti připevněny na testovací kvadrokoptéru. Senzor vzdálenosti HC-SR04 byl umístěn na spodní stranu kvadrokoptéry. Vývojová deska Nucleo byla vložena do nepájivého pole na horní části trupu vzadu (viz přílohu 2).

7.1 Zapojení součástí systému

Vývojová deska Nucleo-L432KC je napájena z regulátoru na napájecí desce dronu, který poskytuje napětí 10V. Napájení je přivedeno na pin VIN, kterým jsou napájeny regulátory pro mikrokontrolér STM32L432KC a ladící nástroj ST-LINK. Pokud není ST-LINK napájen program na cílovém mikrokontroléru se nespustí, neboť neaktivní ladící nástroj jej udržuje ve stavu resetování. Vzájemné zapojení součástí systému je shrnuto v následujících tabulkách. Pro rozložení pinů na desce Nucleo-L432KC viz přílohu 1.

Tab. 9: Zapojení řídicí jednotky SP Racing F3 a Nucleo-L432KC

SP Racing F3	Pin NUCLEO-L432KC
UART2 Gnd	GND
UART2 RX	D1/TX

Tab. 10: Zapojení radiopřijímače Flysky X6B a Nucleo-L432KC

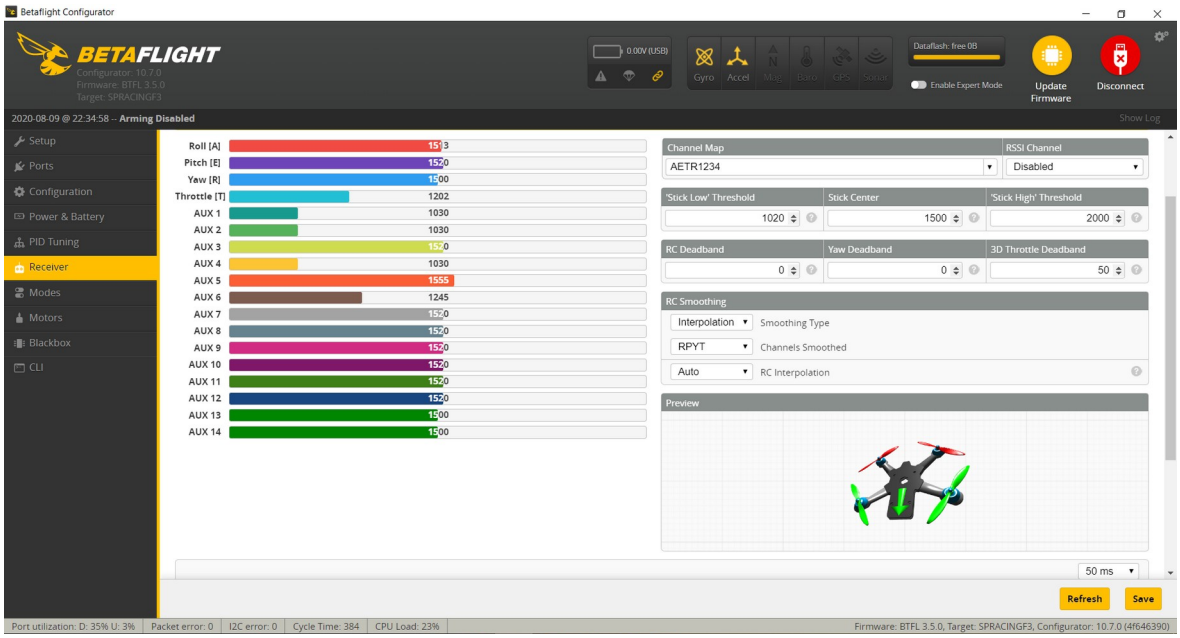
Flysky X6B	Pin NUCLEO-L432KC
5V	5V
GND	GND
iBUS	D0/RX

Tab. 11: Zapojení senzoru HC-SR04 a Nucleo-L432KC

Pin HC-SR04	Pin NUCLEO-L432KC
Vcc	5V
Gnd	GND
Trig	D9 (TIM1 kanál 1)
Echo	D10 (TIM1 kanál 4)

7.2 Testování komunikace

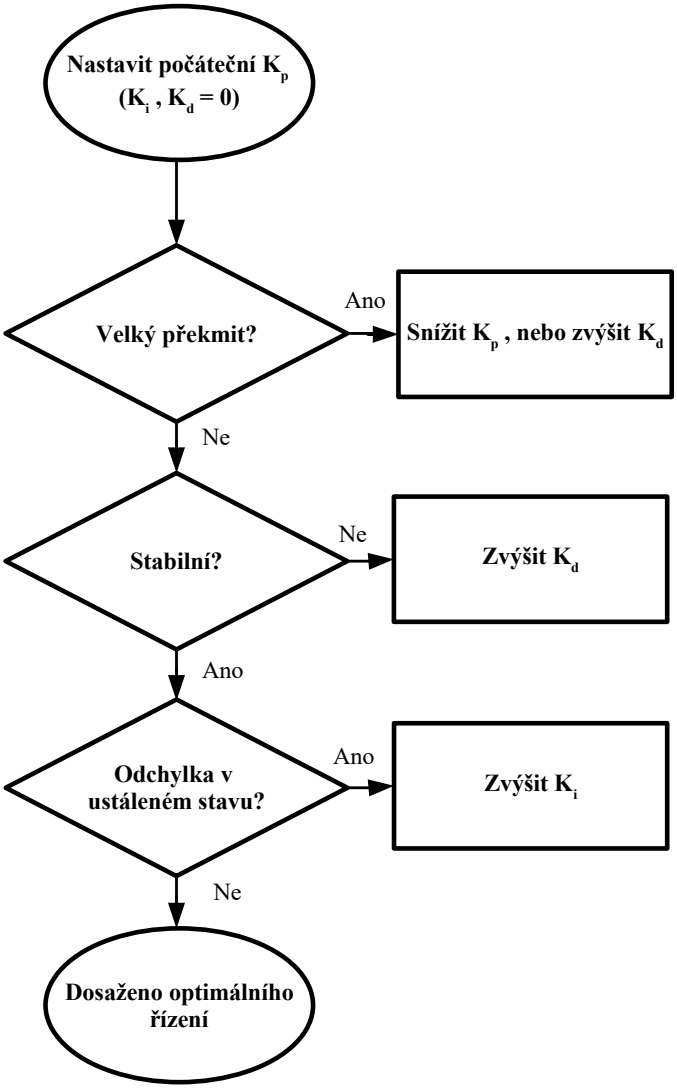
Použitelnost vytvořených funkcí a metod pro sériovou komunikaci mikropočítače STM32L432KC s řídicí jednotkou dronu a radiopřijímačem pomocí protokolu SBUS, byla otestována pomocí záložky **Receiver** v programu Betaflight Configurator. Prvním krokem bylo ověření funkce příjmu sériové komunikace z radiopřijímače a její odeslání do řídicí jednotky. Řídicí jednotka i vývojová deska byly připojeny k počítači pro zajištění napájení. Následně byl zapnut rádiový vysílač a bylo ověřeno zda jsou změny kanálů přijímány řídicí jednotkou. Poté byl otestován režim udržování výšky, při kterém byla podle očekávání hodnota kanálu tahu motorů měněna vytvořenou aplikací nezávisle na hodnotě vysílané rádiovým ovladačem. Na Obr. 25 je zachyceno zobrazení přijímaných hodnot kanálů řídicí jednotkou v průběhu testování.



Obr. 25: Zobrazení přijímaných kanálů v záložce Receiver programu Betaflight Configurator

7.3 Ladění a testování regulátoru

Při hledání parametrů regulátoru bylo využito poznatků z publikace [16] která se zabývá stejnou problematikou jako tato práce – tj. návrh a implementace systému s mikropočítačem a ultrazvukovým senzorem vzdálenosti ke stabilizaci výšky letu kvadrokoptéry. Autoři publikace navrhli empirickou metodu ladění PID regulátoru, využitelnou v případě, kdy není k dispozici přesný matematický model soustavy, nebo parametry získané simulací na matematickém modelu jsou při testu na reálné soustavě shledány nevyhovujícími. Diagram navržené metody je na Obr. 26.



Obr. 26: Empirická metoda ladění PID regulátoru

[16, s. 462, překlad: autor]

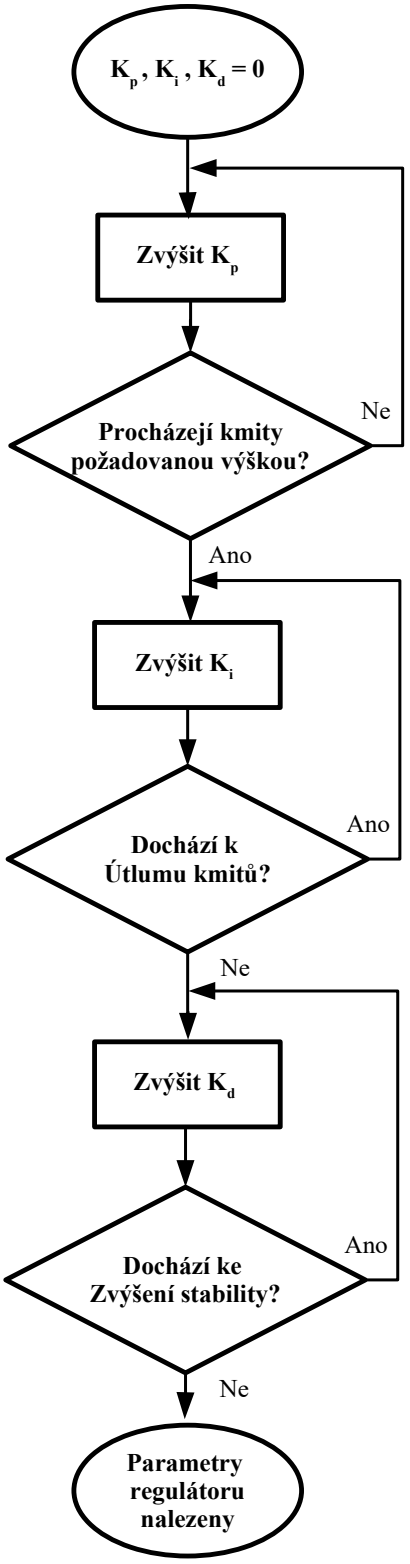
Tato metoda se však při praktickém otestování ukázala být nevyhovující, protože nebylo možné stabilizovat kvadrokoptéru pouze nastavením zesílení proporcionální a derivační složky. Proporcionální složka při použití přírůstkové formy regulátoru způsobuje kmitání

výšky kolem požadované hodnoty, neboť při dosažení požadované hodnoty má akční zásah regulátoru nulovou velikost, a tah motorů je příliš nízký (či vysoký) pro udržení této výšky. Zvyšování zesílení derivační složky není samo o sobě schopné tyto kmity potlačit, a od určité hodnoty dochází spíše k jejich zesílení. Je tedy nutné nastavit v druhém kroku zesílení integrační složky, namísto derivační. Postupným zvyšováním zesílení integrační složky došlo při praktickém testování ke značnému útlumu kmitů. Příliš vysoká hodnota však způsobí opětovnou destabilizaci. Posledním krokem je zvýšení zesílení derivační složky, která eliminuje zbývající tendenci systému oscilovat. Upravený empirický postup pro nalezení parametrů regulátoru je na Obr. 27. Při použití tohoto postupu u přírůstkové formy regulátoru musí být také vhodně zvolená počáteční hodnota akční veličiny. Příliš nízká hodnota může vést k nastavení vysokého zesílení proporcionální složky, a tím k těžko odstranitelným kmitům.

Pomocí upravené empirické metody bylo při praktickém testování dosaženo nejvyšší míry stability s parametry regulátoru:

$K_p = 0.19$
$K_i = 0.20$
$K_d = 0.12$

Z důvodu bezpečnosti při testování (testovací kvadrokoptéra je poměrně výkonná) je akční zásah regulátoru omezen na rozmezí 10 – 60 % maximálního výkonu motorů.



Obr. 27: Vlastní empirická metoda pro ladění PID regulátoru

8 OMEZENÍ SYSTÉMU A PROSTOR PRO VYLEPŠENÍ

Navržená stabilizace letové výšky dronu plní svou funkci spolehlivě pouze do výšky přibližně 120 cm nad zemí. Od této vzdálenosti selhává spolehlivost měření výšky senzorem HC-SR04. Možnou příčinou jsou vibrace způsobené chodem motorů nebo elektromagnetické rušení. Pomocí osciloskopu bylo zjištěno, že při chodu motorů spolehlivost měření nad hranicí 1 m významně klesá. Řešením by mohlo být použití jiné techniky měření vzdálenosti, nebo úprava uchycení senzoru, pro potlačení vibrací.

Dalším vylepšením by mohlo být nalezení parametrů regulátoru pomocí matematického modelování.

ZÁVĚR

V Teoretické části byly popsány základní principy konstrukce, řízení a ovládání dronů, a následně byla věnována pozornost řídicím jednotkám. Z provedeného průzkumu trhu byly prezentovány nejužitečnější poznatky o v současnosti používaných jednotkách. Jak bylo zjištěno, drtivá většina jednotek je postavena na mikrokontrolérech společnosti STMicroelectronics s jádry architektury ARM Cortex. Další důležitou částí práce je popis komunikačních protokolů používaných pro ovládání řídicích jednotek. Byly vybrány čtyři významné způsoby komunikace jednotek, které byly detailně popsány. Nejvhodnějšími způsoby komunikace řídicíh jednotek a radiopřijímačů se jeví být sériové protokoly, jako SBUS nebo iBUS, díky redukci potřebných vodičů a vysoké rychlosti přenosu.

Pro realizaci praktické části práce byla využita vývojová platforma Nucleo-L432KC a ultrazvukový měřič vzdálenosti HC-SR04. Všechny použité součásti systému a pomocné nástroje byly popsány v teoretické části práce. Pro realizovanou aplikaci byly stanoveny cíle a požadavky. Konfigurace součástí cílového mikrokontroléru a způsob implementace objektů a funkcí pro dosažení automatického udržování výšky letu byly detailně popsány a vysvětleny pomocí vývojových diagramů a stavových automatů. Jazykem implementace byl zvolen C++, s využitím abstrakční vrstvy HAL.

Na závěr byly součásti vytvořeného systému umístěny na testovací kvadrokoptéru. Funkčnost komunikace byla otestována za použití software Betaflight Configurator. Parametry regulátoru byly nalezeny vlastní empirickou metodou, která byla odvozena od metody publikované v článku, zabývajícím se problematikou blízkou problematice této práce. Systém vykazuje při použití nalezených parametrů stabilní let až do výšky přibližně 1,2 m nad zemí.

Vytvořený systém by mohl být využit například začínajícími piloty dronů. Automatické udržování výšky letu umožňuje pilotovi plně věnovat svou pozornost horizontálnímu pohybu dronu.

SEZNAM POUŽITÉ LITERATURY

[1] JUSTA, Josef. *Snímání polohy a pohybu pro robotickou ruku* [online]. Plzeň, 2015 [cit. 2020-08-09]. Diplomová práce. Západočeská univerzita v Plzni, Fakulta elektrotechnická, Katedra aplikované elektroniky a telekomunikací. Dostupné z: https://otik.zcu.cz/bitstream/11025/18941/1/diplomka%20v1.3.pdf

[2] EBEID, Emad Samuel Malki, Martin SKRIVER, Kristian TERKILDSEN, Kjeld JENSEN a Ulrik SCHULTZ. A Survey of Open-Source UAV Flight Controllers and Flight Simulators. *Microprocessors and Microsystems* [online]. 2018, **61** [cit. 2020-08-01]. Dostupné z: doi:10.1016/j.micpro.2018.05.002

[3] MACHÁČEK, Zdeněk a Pavel NEVŘIVA. *Modulované signály*. 1. vyd. Ostrava: Vysoká škola báňská – Technická univerzita Ostrava, 2012. ISBN 978-80-248-2600-4.

[4] FAESSLER, Matthias. SBUS Protocol. *GitHub* [online]. [cit. 2020-07-16]. Dostupné z: https://github.com/uzh-rpg/rpg_quadrotor_control/wiki/SBUS-Protocol

[5] *The FlySky iBus protocol* [online]. 22. říjen 2017 [cit. 2020-08-09]. Dostupné z: http://blog.dsp.id.au/posts/2017/10/22/flysky-ibus-protocol/

[6] LIANG, Oscar. RC TX RX PROTOCOLS EXPLAINED: PWM, PPM, SBUS, DSM2, DSMX, SUMD. *Oscar Liang* [online]. 24. červenec 2017 [cit. 2020-08-08]. Dostupné z: https://oscarliang.com/pwm-ppm-sbus-dsm2-dsmx-sumd-difference/

[7] LIANG, Oscar. SPI_RX – A NEW RECEIVER PROTOCOL | FC WITH BUILT-IN RX. *Oscar Liang* [online]. 5. březen 2018 [cit. 2020-08-08]. Dostupné z: https://oscarliang.com/spi-rx-receiver-protocol/

[8] *STM32L432KB STM32L432KC Datasheet* [online]. B.m.: STMicroelectronics. 2018 [cit. 2020-08-02]. Dostupné z: https://www.st.com/resource/en/datasheet/stm32l432kc.pdf

[9] micropython. *Github* [online]. [cit. 2020-08-04]. Dostupné z: https://github.com/micropython/micropython

[10] What should I use to develop on STM32 ? *EMCU* [online]. [cit. 2020-08-04]. Dostupné z: http://www.emcu.eu/what-should-i-use-to-develop-on-stm32/

[11] DUDKA, Michal. Přístupy k programování STM32. *root.cz* [online]. 3. říjen 2017 [cit. 2020-08-04]. Dostupné z: https://www.root.cz/clanky/pristupy-k-programovani-stm32/

[12] *Description of STM32L4/L4+ HAL and low-layer drivers* [online]. B.m.: STMicroelectronics. [cit. 2020-08-04]. Dostupné z: https://www.st.com/resource/en/user_manual/dm00173145-description-of-stm32l4l4-hal-and-lowlayer-drivers-stmicroelectronics.pdf

[13] SP RACING™ F3 FLIGHT CONTROLLER. *Seriously Pro* [online]. [cit. 2020-08-08]. Dostupné z: http://seriouslypro.com/spracingf3

[14] Číslicové řízení procesů. *Řízení technologických procesů* [online]. [cit. 2020-08-09]. Dostupné z: http://rtp.webzdarma.cz/rizeni16.php#cislicove

[15] CHATTERJEE, Kirit. A simple digital low-pass filter in C. *Kirit Chatterjee* [online]. [cit. 2020-08-08]. Dostupné z: https://kiritchatterjee.wordpress.com/2014/11/10/a-simple-digital-low-pass-filter-in-c/

[16] LEONG, Bernard Tat Meng, Sew Ming LOW a Melanie Po-Leen OOI. Low-Cost Microcontroller-based Hover Control Design of a Quadcopter. *Procedia Engineering* [online]. 2012, **41**, 458–464. ISSN 1877-7058. Dostupné z: doi:https://doi.org/10.1016/j.proeng.2012.07.198

[17] Nucleo-L432KC. *msalsmon.pl* [online]. [cit. 2020-08-02]. Dostupné z: https://sklep.msalamon.pl/produkt/nucleo-l432kc/

[18] All About Multirotor Drone Radio Transmitters and Receivers. *GetFPV* [online]. [cit. 2020-08-09]. Dostupné z: https://www.getfpv.com/learn/new-to-fpv/all-about-multirotor-fpv-drone-radio-transmitter-and-receiver/

[19] Move Ryze drone in all three axes - MATLAB move. *MathWorks* [online]. [cit. 2020-08-09]. Dostupné z: https://www.mathworks.com/help/supportpkg/ryzeio/ref/move.html

[20] NANCE, Rob. Anatomy of a drone. *Make: Technology on Your Time Volume 37: Homegrown Drones!* 2014, (37), 160. ISSN 1556-2336.

[21] NUCLEO-L432KC. *Mbed* [online]. [cit. 2020-08-09]. Dostupné z: https://os.mbed.com/platforms/ST-Nucleo-L432KC/

[22] *Ultrasonic Ranging Module HC - SR04* [online]. [cit. 2020-08-08]. Dostupné z: https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf

[23] Flysky X6B. *FlyTime* [online]. [cit. 2020-08-08]. Dostupné z: https://www.flytime.cz/rc/flysky-x6b/

[24] HC-SR04 Ultrasonic Range Finder. *RobotShop* [online]. [cit. 2020-08-09]. Dostupné z: https://www.robotshop.com/eu/en/hc-sr04-ultrasonic-range-finder-tys.html

[25] CLADERA, Antoni. Drone Photography: The Definitive Guide (2020). *PhotoPills* [online]. [cit. 2020-08-09]. Dostupné z: https://www.photopills.com/articles/drone-photography-guide#step3

[26] *SP Racing F3* [online]. [cit. 2020-08-01]. Dostupné z: http://wikirotors.com/index.php?title=File:Sp_racing_pro_f3_vorne.png#file

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

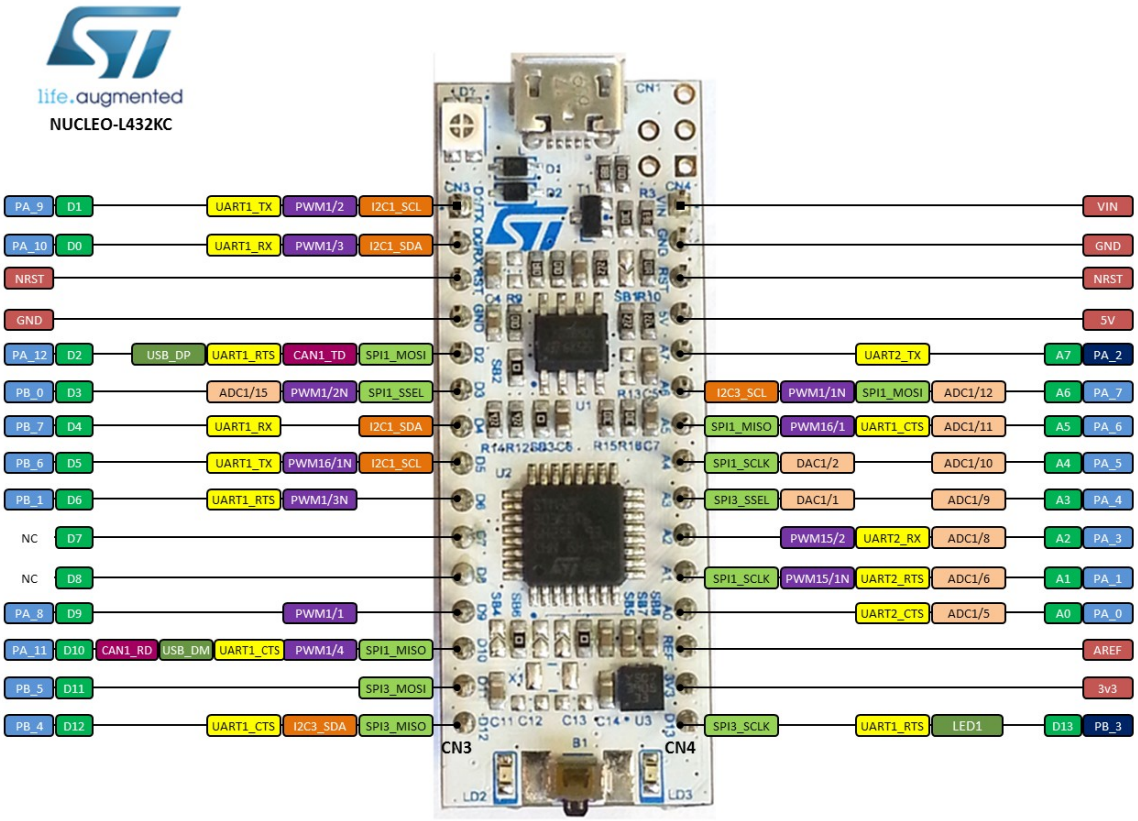
Zkratka	Význam v češtině	Význam v angličtině
API		application programming interface
BLDC	bezkartáčový stejnosměrný (motor)	brush-less direct current (motor)
CAN		controller area network
CLI	příkazový řádek	command line interface
CRC	cyklický redundantí součet	cyclic redudndancy check
DMA	kanál přímého přístupu do paměti	direct memory access
DOF	počet stupňů volnosti	degrees of freedom
DPS	deska plošných spojů	
ESC	elektronický regulátor otáček	electronic speed controller
FC	letový kontroler	flight controller
GCC		GNU compiler collection
GPS	globální polohový systém	global positioning systém
HAL	hardwarová abstrakční vrstva	hardware abstraction layer
I^2C		Inter-integrated circuit
IMU	inerciální měřící jednotka	inertial measurement unit
LL API		low level application programming interface
MCU	mikrokontrolér	microcontroller unit
MEMS	Mikroelektro-mechanické systémy	microelectromechanical systems
NVIC		nested vector interrupt controller
OSD		on-screen display
PDB		power delivery board
PID	proporcionální-integrační-derivační	proportional-integral-derivative
PPM	pulzně poziční modulace	pulse position modulation
PWM	pulzně šířková modulace	pulse width modulation
RTC	hodiny reálného času	real time clock
SPI	sériové periferní rozhraní	seriál peripheral interface
UART	univerzální asynchronní přijímač-vysílač	universal asynchronous receiver-transmitter
USART	univerzální synchronní/asynchronní přijímač-vysílač	universal sunchronous/asynchronous receiver-transmitter

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

SEZNAM PŘÍLOH

PŘÍLOHA P 1: ROZLOŽENÍ A FUNKCE PINŮ NUCLEO-L432KC



Zdroj: [21]

PŘÍLOHA P 2: UMÍSTĚNÍ SOUČÁSTÍ SYSTÉMU NA TESTOVACÍ KVADROKOPTÉŘE