



**ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE**  
Fakulta Elektrotechnická  
Katedra Měření

**Ovládání dronu Ryze Tello přes webové rozhraní**

**Control of the Drone Ryze Tello through the Web Interface**

Bakalářská práce

Studijní program: Otevřená Informatika

Studijní obor: Internet Věcí

Vedoucí práce: RNDr. Petr Štěpán, Ph.D.

**Vladimír Glingar**  
**Praha 2019**



# ZADÁNÍ BAKALÁŘSKÉ PRÁCE

## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Glingar** Jméno: **Vladimír** Osobní číslo: **457873**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávající katedra/ústav: **Katedra měření**  
Studijní program: **Otevřená informatika**  
Studijní obor: **Internet věci**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:  
**Ovládání dronu Ryze Tello přes webové rozhraní**

Název bakalářské práce anglicky:  
**Control of the Drone Ryze Tello through the Web Interface**

Pokyny pro vypracování:  
1. Seznamte se s ovládáním dronu Ryze Tello přes rozhraní Wi-fi pomocí Tello SDK.  
2. Navrhněte webové rozhraní pro komunikaci s dronem a jeho řízení. Webové rozhraní by mělo přenášet obraz z kamery dronu a mělo by umět ovládat dron přes příkazy typu RC – virtuální dálkové ovládání, tak i příkazy letu o zadané souřadnice.  
3. Otestujte možnosti automatizace řízení dronu pomocí zpracování informace z obrazu dronu.

Seznam doporučené literatury:  
[1] Ryze Tello SDK, online: <https://www.ryzerobotics.com/tello/downloads>

Jméno a pracoviště vedoucí(ho) bakalářské práce:  
**RNDr. Petr Štěpán, Ph.D., Multirobotické systémy FEL**

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:  
\_\_\_\_\_

Datum zadání bakalářské práce: **18.02.2019** Termín odevzdání bakalářské práce: \_\_\_\_\_

Platnost zadání bakalářské práce:  
**do konce letního semestru 2019/2020**

\_\_\_\_\_  
RNDr. Petr Štěpán, Ph.D.  
podpis vedoucí(ho) práce

\_\_\_\_\_  
podpis vedoucí(ho) ústavu/katedry

\_\_\_\_\_  
prof. Ing. Pavel Ripka, CSc.  
podpis děkana(ky)

## III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studenta

## **Prohlášení**

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

**Praha, 28. 4. 2019**

**Vladimír Glingar**

## **Poděkování**

**Rád bych tímto poděkoval RNDr. Petru Štěpánovi, Ph.D. za poskytnuté rady a připomínky nejen k bakalářské práci, ale i k samostatnému projektu, který práci předcházel. Dále chci poděkovat rodině za psychickou podporu, díky které jsem se mohl na práci plně soustředit.**

## **Abstrakt**

Práce se zabývá tvorbou webového rozhraní pro ovládání dronu „RYZE Tello“. Vytvořená aplikace vychází z SDK, vydaného společností RYZE Tech. Rozhraní je implementováno v „NodeJS“ a ke správné funkčnosti je potřeba mít komponentu „ffmpeg“. Dále se práce zabývá otázkou možnosti automatizace dronu na základě dat přijatých z kamery dronu.

**Klíčová slova:** dron, webové rozhraní, NodeJS, RYZE Tello, ffmpeg

## **Abstract**

Thesis focuses on creation of web interface to control drone „RYZE Tello“. Created application is based on SDK (software development kit) made by company RYZE Tech. Interface is programmed in „NodeJS“ javascript language and for correct function is mandatory to have „ffmpeg“ installed too. Furthermore, the thesis deals with the possibility of drone automation based on data received from a drone camera.

**Key words:** drone, web interface, NodeJS, RYZE Tello, ffmpeg

# Obsah

Seznam obrázků .....	7
1. Úvod .....	8
2. Drony .....	9
2.1. Historie dronů: drony včera, dnes a zítra .....	9
2.1.1. Drony včera.....	9
2.1.2. Drony dnes .....	12
2.1.3. Drony zítra .....	13
2.2. Dron Ryze Tello .....	13
3. Komunikační protokol a SDK .....	16
4. Aplikace ovládání dronu.....	20
4.1. Návrh aplikace .....	20
4.2. Příkazové rozhraní .....	21
4.3. Stav dronu .....	23
4.4. Kamerové rozhraní.....	24
5. Zpracování obrazu z dronu .....	27
6. Závěr.....	30
Zdroje / použitá literatura .....	31
Seznam příloh .....	33
Příloha č. 1: CD .....	33

# Seznam obrázků

1. Nákres helikoptéry od Leonarda Da Vinciho. Převzato z [1].
2. Aerodrom č. 5 Samuela P. Langleyho. Převzato z [2].
3. Teslův experiment rádiově naváděné loďky, New York 1898. Převzato z [3].
4. Kettering Bug. Převzato z [4].
5. Jeden z prvních modelů „de Havilland Queen Bee K4227“. Převzato z [5].
6. Model radioplánu QQ-2. Převzato z [6].
7. Dron RYZE Tello. Převzato z [7].
8. Zakladatel společnosti Frank Wang se svým dronem. Převzato z [8].
9. Dron DJI Mavic Pro. Převzato z [9].
10. Ukázka kódu, inicializace GUI [vlastní zpracování].
11. GUI Aplikace, zvýrazněné příkazové rozhraní [vlastní zpracování].
12. Ukázka kódu, funkce zajišťující posílání příkazu [vlastní zpracování].
13. GUI Aplikace, zvýrazněné stavové rozhraní [vlastní zpracování].
14. Popis rotace dle jednotlivých os. Převzato z [10].
15. Ukázka kódu, zpracování dat informujících o stavu dronu [vlastní zpracování].
16. Ukázka kódu, zpracování kamerového rozhraní [vlastní zpracování].
17. GUI Aplikace, ukázka kamerového záznamu [vlastní zpracování].
18. Úrovně samostatnosti dronů dle DroneII.com Převzato z [11].
19. Postup zpracování obrazu. Převzato z [12].

# 1. Úvod

Technologie se neustále vyvíjí. V dnešní době technologičtí giganti jako je například Apple, Microsoft nebo Intel určují trendy a mají velký vliv na směr, kterým se vývoj ubírá.

Internetové připojení v domácnosti je dnes běžná věc a málokdo dnes žije nevybaven buď Wi-fi připojením nebo klasickým „ethernet“ připojením. Diskety byly nahrazeny Compact Disky a i ty už dnes končí svou éru. Každý druhý člověk dnes vlastní datové úložiště s enormní kapacitou a ti ostatní využívají oblíbené cloudové služby poskytované nadnárodními korporacemi. Tlačítkové telefony již nejsou populární a až na výjimky u starší generace vlastní téměř každý dotykový telefon (ne-li více), které kdysi bývaly luxusem.

V rámci moderního vývoje se na pomyslný vrchol žebříčku popularity dostala umělá inteligence, která je dnes téměř v každém zařízení. Inteligentní domácnost se postará o minimalizaci provozních nákladů a maximalizaci pohodlí pro uživatele. Dražší automobily obsahují autopilota, který už je dnes schopen zvládnout většinu krizových situací, včetně mikrospánku řidiče.

Toto vše by nemohlo existovat bez obětavé práce mnoha lidí v oboru, kde „zítra“ znamená úplně nový svět.

Má bakalářská práce se zaměřuje na odvětví, které úzce souvisí s umělou inteligencí, a tím jsou UAV („unmanned aerial vehicle“ to jest „bezpilotní vzdušné stroje“). Pro běžnou populaci se pro tento termín více zažilo označení „dron“ nebo „drone“ v anglickém jazyce, jenž ve volném překladu znamená „trubce“ – včelího samce. Toto označení je trefné, protože kooperace dronů je programována na základě chování včel a navíc drony vydávají charakteristický zvuk, který je podobný bzukotu roje včel [13].

Cílem práce je navrhnout webové rozhraní, přes které bude možné ovládat dron „Ryze Tello“. Ve druhé části práce se pak zaměřím na možnosti automatizace dronu na základě zpracování dat z obrazu.



## 2. Drony

### 2.1. Historie dronů: drony včera, dnes a zítra

#### 2.1.1. Drony včera

Přestože vývoj dronů jako bezpilotních létajících strojů začal až s technologickým boomem ve 20. století, první náznaky podobných strojů lze nalézt daleko v historii. Osobně si myslím, že první kámen položil významný italský umělec a vynálezce Leonardo Da Vinci, když vytvořil náčrt vrtulníku. Obecný koncept vrtulníku byl sice znám v Číně dlouho před tím, než ho Da Vinci popsal, avšak první evropský návrh můžeme připsat jemu. Bohužel nejsou historická svědectví o aplikaci jeho modelu v praxi.

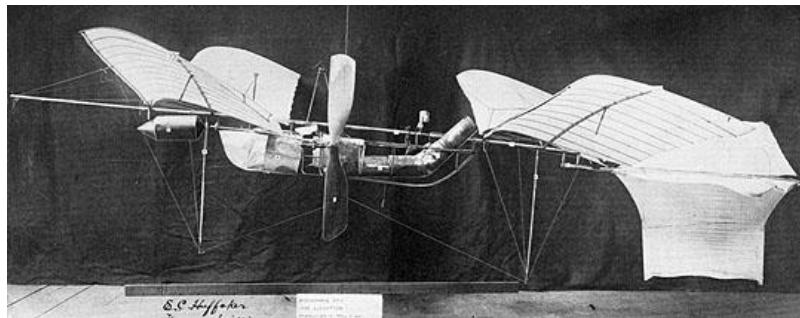


Obrázek č. 1: Náčrt helikoptéry od Leonarda Da Vinciho. Převzato z [1].

První pokus využití létajících strojů v praxi však byl uskutečněn až v 19. století. V červenci roku 1849 rakouská vojska vyslala na Benátky bezpilotní balóny naložené trhavinou se šrapnely. Neví se přesně, kolik bezpilotních balónů reálně bylo, avšak z dopisu konstruktéra těchto zařízení Franze von Uchatia lze usoudit, že ve skutečnosti byly 2 balóny. Vzhledem k tomu, že útoku nebyla ve světě přikládána moc velká pozornost, jsou o této události minimální informace [14, 15].

Vývoj na druhé straně Atlantského oceánu však také nebyl pozadu. V USA byla v té době hojně využívána pára a tak se nelze moc divit, když koncem 19. století

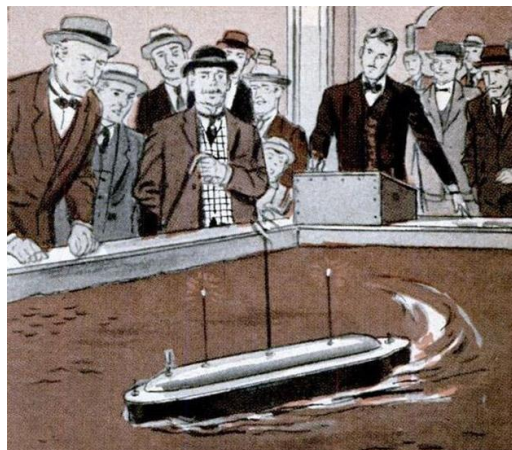
vyvinul Samuel Langley bezpilotní letoun poháněný párou, který dokázal letět déle než 1 minutu.



Obrázek č. 2: Aerodrom č. 5 Samuela P. Langleyho. Převzato z [2].

Dále před přelomem století vznikl nápad na spojení kamery a letounu. Je možné, že tento nápad byl ovlivněn knihou dalšího snílka či vizionáře budoucnosti, francouzského spisovatele Julese Verna *Dva roky prázdnin*.

Jako předchůdce dronů lze také považovat výzkum srbského vědce Nikoly Tesly, který v roce 1898 poukázal na budoucí využití rádiových vln a bezdrátové komunikace obecně, když v New Yorku předvedl rádiem naváděnou loďku.<sup>1</sup>



Obrázek č. 3: Teslův experiment rádiově naváděné loďky, New York 1898. Převzato z [3].

Další vývoj dálkově řízených letadel nastal během První světové války, když USA začala uvažovat o vývoji bezpilotní bomby, shazovatelné ze vzduchu. Za tímto účelem byl osloven vynálezce Charles Kettering, který vyvinul dvouplošník (tzv. „Kettering Bug“), jenž, po nastavení kurzu a délky letu, při dosažení vypočteného cíle

<sup>1</sup> Zajímavé je, že Nikola Tesla se stal v roce 1936 držitelem čestného doktorátu ČVUT [16].

sám odpojil křídla a zřítíl se společně s bombou na cíl. Tento způsob se však ukázal jako neefektivní. Jako jeho další negativum byla neřiditelnost letadla, protože zásah do řízení stoje po startu již nebyl možný.



Obrázek č. 4: Kettering Bug. Převzato z [4].

Ve 30. letech 20. století se konstruktérům Velké Británie podařilo spojit Ketteringovu myšlenku (která byla tehdy sice utajená, ale zřejmě informace nějak unikla) s Teslovým objevem a vytvořit dálkově ovladatelný dvouplošník zvaný „Queen Bee“.



Obrázek č. 5: Jeden z prvních modelů „de Havilland Queen Bee K4227“. Převzato z [5].

Druhá světová válka opět přinesla závody ve zbrojení a tím pádem i masovou výrobu dálkově ovládaných letadel. Ve Velké Británii to byla Queen Bee (tehdy již vylepšená), v USA to byly stoje typu QQ-2. Ty se používaly během Druhé světové války nejčastěji a jen ve válečném období jich bylo vyrobeno více než 400 strojů [5].



Obrázek č. 6: Model radioplánu QQ-2. Převzato z [6].

Vývoj UAV po Druhé světové válce je dobře zdokumentován. Na přelomu 60. a 70. let vznikl mikročip a pomalu začalo období miniaturizace, které trvá do dnešních dnů. Drony byly stále používány pouze pro armádní účely a sloužily k průzkumným misím nejen ve válečných zónách.

### **2.1.2. Drony dnes**

Změna nastala v roce 2000, když 2. května tehdejší americký prezident Bill Clinton oznámil konec vkládání umělé odchylky signálu GPS, která znemožňovala reálné využití systému GPS civilisty. To znamenalo, že civilní obyvatelstvo mohlo využívat systém GPS pro své účely.

Toto otevřelo možnost používání dronů pro komerční účely a zábavu. Postupem času se začaly na trhu objevovat kvadrokoptéry jak je známe dnes a vytvořily produkt, který získal poptávku napříč generacemi.

V dnešní době se drony po celém světě používají pro různé komerční účely: na pořizování leteckých fotografií a videí, pro letecký monitoring, mapování, transport a logistiku, zábavu aj. [13, str. 39-67].

V posledních letech vzniklo mnoho výrobců bezpilotních letounů a dnes se drony vyrábějí nejen ve Spojených státech, ale například v Číně nebo Indii. V roce 2006 vznikla společnost Dà-Jiāng Innovations, dnes známá pod zkratkou DJI. Tato čínská společnost začala podnikat v oboru UAV a záhy se stala špičkou v zaměření na výrobu a vývoj civilních dronů.

### 2.1.3. Drony zítra

Jak se vyvíjí technologie, jde s tím ruku v ruce využití dronů. S rozvojem AI neboli umělé inteligence se vývoj čím dál více zaměřuje na automatizaci a větší užitek dronů pro běžnou populaci.

Dle autorů knihy *Drony „budoucnost dronů se předjímá v globálním hledisku velice pozitivně a nárůst počtu využívání dronů a lidí, které tento trh v budoucnosti zaměstná, se pohybuje v milionech.“* [13, str. 253].

Amazon se zaměřil na dodávání zásilek pomocí dronů. Facebook testuje drony na solární pohon, které by mohly šířit internetové připojení v rozvojových zemích. Některé pizzerie začali dodávat své produkty s využitím dronů. Tyto malé létající zařízení se staly součástí běžného života a je velmi pravděpodobné, že jejich využití s během času ještě poroste. Kam až se však vývoj dostane, stojí ve hvězdách a možná se v budoucnu některé drony vypraví i do kosmu. Pokud však legislativa nepůjde ruku v ruce s vývojem a nebudou vydány jednotné předpisy platné pro celý svět jako celek, pak je zde velké riziko zneužití této technologie ke špatným účelům, jako je například nezákonná špionáž nebo terorismus.

## 2.2. Dron Ryze Tello

Dron „Ryze Tello“ je produktem společného podniku DJI („Dà-Jiāng Innovations Science and Technology Co.“), RYZE Tech Co., Ltd. a Intel.



Obrázek č. 7: Dron RYZE Tello. Převzato z [7].

Společnost „Dà-Jiāng Innovations Science and Technology Co.“ je čínskou společností sídlící v městě Shenzhen ležícím v provincii Guandong. Tato lokalita z důvodu výskytu mnoha technologických společností bývá označována jako čínský Silicon Valley. Jejím zakladatelem je Frank Wang (Wāng Tāo), který ve svých 26. letech v roce 2006 společnost založil. Díky studiu na Hongkongské vědecko-technologické univerzitě dostal v roce 2005 grant na 2300 USD s cílem vyvinout dron, což se mu nad jeho očekávání vydařilo. Když zjistil, že jeho návrhy mají hodnotu tisíců amerických dolarů, uvědomil si potenciál svých výrobků a zároveň objevil tehdy ne příliš známé odvětví, které nebylo na trhu zastoupeno [8].



Obrázek č. 8: Zakladatel společnosti Frank Wang se svým dronem. Převzato z [8].

Postupným vývojem se v letech 2006 – 2013 startup s méně než 30 zaměstnanci zaměřil na vytvoření co nejlevnějšího modelu, který by byl dostupný pro každého. Společně se snižováním prodejní ceny se začala objevovat poptávka, která byla obrovská.

V roce 2013 se na trhu objevil model „DJI Phantom“. Tento model představoval klíčový krok pro společnost a nákladem 400 000 kusů zvýšil její roční zisk z tisíců amerických dolarů na milióny.

Společnost se záhy stala světoznámou a vytvořila regionální pobočky v Evropě a na americkém kontinentu. Největší ohlas zaregistrovala Evropa a Spojené státy americké, kam společně s Asií putovalo 90% vyrobených produktů. Zároveň výrobky společnosti pokrývali 70% výrobků v odvětví, což v překladech znamenalo dominanci na světovém trhu.

V roce 2016 společnost představila první model ze série „Mavic“ „Mavic Pro“, který se vyjímal svou velikostí, díky níž se vešel i do brašny. Kamera nebyla přímo vestavená uvnitř modelu, ale přidělaná pomocí Kardanového závěsu, který umožňuje s kamerou za letu otáčet. Udávaná maximální doba letu je 24 minut.

Dalším modelem z této série, stojícím za zmínku, je rok stará novinka „Mavic Air“, která se od původního modelu liší menší konstrukcí, lepší kamerou. Užitečnost této série se ukázala při nedávném požáru francouzské katedrály Notre-Dame, kde místní hasiči využili dron „Mavic Pro“ k hledání ohnisek požáru [17].



Obrázek č. 9: Dron DJI Mavic Pro. Převzato z [9].

Model „Ryze Tello“ vznikl v roce 2018 ve spolupráci čínských společností DJI, Ryze Technologies a americké společnosti Intel. Historie společnosti Ryze Technologies není průhledná. Jediné, co o ní s určitostí víme, je to, že se jedná o startup založený v roce 2017. Je však možné, že se jedná o dceřinou společnost DJI. Obě společnosti mají sídlo ve stejné budově a dron „Ryze Tello“ je jediným dronem, který společnost Ryze Technologies vyrábí.

Při bližším průzkumu zjistíme podobnost s modelem DJI Spark, nicméně je zde několik rozdílů. Pokud budeme brát cenu jako ukazatel kvality, pak můžeme Ryze Tello brát jako levnější (a omezenější) verzi dronu DJI Spark. Dron „Ryze Tello“ má ve výbavě slabší kameru, méně kvalitní motory a nemá tolik senzorů. V maximální době letu se sice moc neliší, ale je zde dvacetinásobný rozdíl v maximálním rozsahu [18, 19].

Toto jsou sice nevýhody, ale za cenu přibližně 2300 korun dostaneme praktický malý dron s programovatelným rozhraním, který je ideálním dronem pro případné experimenty a vývoj programů. K tomuto slouží „Tello SDK“ (Software development kit) [20], jehož funkcionality zastřešují rozhraní pro komunikaci s dronem, jenž je základním kamenem mé aplikace a bude popsán v následující kapitole.

### 3. Komunikační protokol a SDK

Jak bylo zmíněno v předchozí kapitole, dron Ryze Tello obsahuje programovatelné rozhraní, přes které lze dron ovládat. Všechny funkce rozhraní jsou popsány v „Tello SDK“ manuálu [20].

K dronu je možné se připojit pomocí Wi-Fi UDP portu a s pomocí textových příkazů vysílaných na tento port dron ovládat. Po zapnutí dronu by se měl objevit Wi-Fi hotspot, který slouží jako komunikační bod s dronem. V defaultním nastavení má hotspot název „Tello“ a za ním následuje sériové číslo dronu. Hotspot není při tovární konfiguraci šifrovaný, nicméně dle potřeby je možnost nastavit heslo a zabránit tomu, aby se nežádoucí návštěvník mohl připojit k dronu a zmocnit se kontroly nad ním.

Při testování dronu jsem zjistil, že některé chytré zařízení mohou vytvářet komplikaci tím, že se stále připojují k dronu a zároveň tím blokují připojení cílového zařízení. Pokud je tedy hotspot viditelný, ale nedaří se k němu připojit, je nutné zkontrolovat, zda se v okolí nenachází nějaké z těchto zařízení a případně toto zařízení vypnout.

Z konfiguračního hlediska má dron pevně nastavenou IP adresu a to 192.168.10.1.

Přednastavené komunikační kanály (porty) jsou:

- a) 8889 – kanál pro posílání příkazů a přijímání reakcí od dronu
- b) 8890 – kanál, na němž dron vysílá svůj aktuální stav
- c) 11111 – kanál pro přenos toku, jenž obsahuje data z kamery dronu

Pro spuštění SDK módu je nutné poslat příkaz „command“ na port 8889. Po úspěšném provedení příkazu lze jednat posílat další pokyny dronu a dále se na portu 8890 aktivuje rozhraní informující o stavu dronu.

Pro spuštění video přenosu musíme být rovněž v SDK módu (tj. poslán úspěšně příkaz „command“) a dále musí být na dron vyslán příkaz „streamon“, jenž aktivuje samotný přenos.



V SDK jsou implementovány 3 základní typy příkazů:

*Ovládací příkazy* – vrací „ok“ v případě úspěchu nebo „error“ v ostatních případech.

*Čtecí příkazy* – několik příkazů, které vrací hodnoty vybraných parametrů, jež jsou až na výjimky vysílány na kanálu 8890.

*Nastavovací příkazy* – příkazy umožňující nastavit některé parametry dronu jako je například rychlost. Stejně jako ovládací příkazy, vrací „ok“ nebo „error“ v závislosti na úspěchu nastavení hodnot.

Mezi *ovládací příkazy* patří:

- „command“ – slouží k přepnutí dronu do SDK módu;
- „takeoff“ – pro vzletnutí dronu;
- „land“ – pro přistání dronu;
- „streamon“ a „streamoff“ – pro zapnutí / vypnutí vysílání dat z kamery;
- „emergency“ – pro vypnutí motorů v případě nutnosti;
- „up x“ a „down x“ – pro pohyb dronu nahoru a dolů, kde x je definováno v milimetrech a může nabývat hodnot od 2 cm až po 5 m;
- „left x“ a „right x“ – pro pohyb dronu vlevo nebo vpravo se stejným omezením jako u ostatních příkazů pro pohyb;
- „forward x“ a „back x“ – pro pohyb vpřed a vzad;
- „cw x“ a „ccw x“ – umožňující dronu rotaci kolem své osy – hodnota je dána ve stupních a může nabývat 1 až 3600 stupňů, což znamená, že při maximální hodnotě se dron 10x otočí kolem své osy;
- „flip l“, „flip r“, „flip f“ a „flip b“ – dron vykoná přemet vlevo (l), vpravo (r), vpřed (f) nebo vzad (b);
- „curve x y z x1 y1 z1 s“ - dron provede pohyb na křivce dané třemi body – výchozí pozicí a souřadnice dalších dvou bodů dostane jako argument ([x, y, z], [x1, y1, z1]). Posledním parametrem je rychlost 10 – 60 cm/s.

*Nastavovacích příkazů* je znatelně méně - pouze 3:

- 1) „speed x“ – pro nastavení rychlosti pohybu v rozmezí 10 – 100 cm/s;

- 2) „rc a b c d“ – příkaz pro simulaci rádiového ovládání, kde jsou parametry a, b, c, d v rozmezí -100 až 100 a znamenají pohyb vlevo / vpravo (a), dopředu / vzad (b), nahoru / dolů (c) a rotaci kolem své osy vlevo nebo vpravo (d). Například, pokud zadáme „rc 100 0 0 0“, dron poletí vlevo. Pro zastavení pohybu dronu stačí tedy vyslat příkaz „rc 0 0 0 0“.
- 3) „wifi ssid pass“ – přes tento příkaz můžeme nastavit heslo na hotspot.

Poslední kategorií příkazů zmíněných výše jsou *čtecí příkazy*, které jsme schopni snadno odlišit od ostatních, protože na konci každého příkazu se vyskytuje znak „?“.

Definované příkazy jsou tyto:

- „speed?“ – pro získání aktuálně nastavené rychlosti pohybu v cm/s;
- „battery?“ – informující o procentuálním stavu baterie;
- „time?“ – pro získání aktuální doby letu;
- „height?“ – informující o aktuální výšce;
- „temp?“ – informuje o aktuální teplotě dronu;
- „attitude?“ – vrací hodnoty naklopení dle jednotlivých os;
- „baro?“ – hodnota zaznamenávaná vestavěným barometrem;
- „acceleration?“ – vrací hodnoty zrychlení dle jednotlivých os;
- „tof?“ – pro zjištění vzdálenosti od povrchu (tj. vzdálenosti, kterou má dron pod sebou);
- „wifi?“ – vrací sériové číslo dronu.

Jak jsem zmínil na začátku této kapitoly, na portu 8890 dron v SDK módu vysílá svůj stav. Celkem vysílá 21 hodnot, ve formátu:

```
„mid:%d;x:%d;y:%d;z:%d;mpry:%d,%d,%d;pitch:%d;roll:%d;yaw:%d;vgx:%d;vgy:%d;vgz:%d;templ:%d;temph:%d;tof:%d;h:%d;bat:%d;baro: %.2f;time:%d;agx:%.2f;agy:%.2f;agz:%.2f;\r\n“
```

- „pitch“, „roll“ a „yaw“ – což jsou hodnoty naklopení dle os;
- nejnižší a nejvyšší teplota dronu (*templ*, *temph*);
- výška (*h*) a „TOF vzdálenost“ tj. vzdálenost od povrchu (*tof*);
- stav baterie (*bat*), hodnota barometru (*baro*), doba letu (*time*);

- zrychlení dle jednotlivých os (agx, agy, agz).

Dále jsou na portu 11111, po splnění podmínek výše, vysílány datové rámce se záznamem z kamery dronu. Data jsou zasílána ve formátu H264, který lze dekodovat pomocí knihoven, např. „opencv“ nebo „ffmpeg“.

Je nutné také poznamenat, že pokud nedostane dron žádný příkaz po dobu 15 vteřin, pak automaticky přistane.

## 4. Aplikace ovládání dronu

### 4.1. Návrh aplikace

Před začátkem vývoje samotné aplikace bylo nezbytné rozhodnout, v jakém jazyce bude program napsán. Mé bakalářské práci předcházela samostatný projekt, který byl zaměřen na ovládání dronu skrze příkazovou řádku a byl napsaný v jazyce C/ C++. Pro webové rozhraní však není tento jazyk nejvhodnější. Proto jsem se rozhodl psát aplikaci v javascriptu, který je běžně používán pro běh skriptů webových aplikací.

Komunikace dronu s PC však používá UDP protokol, který je ve webových prohlížečích zakázaný. Byl jsem tedy nucen vyhledat nějakou alternativu. Nakonec jsem se rozhodl pro využití NodeJS, což je běhové prostředí, které umožňuje spouštět aplikace v javascriptu mimo prohlížeč. Pro grafické zobrazení jsem použil open-source knihovnu Electron, která implementuje základní GUI. Použitím těchto knihoven se otevřela možnost použití knihovny dgram, která obsahuje funkce pro tvorbu UDP socketů.

Samotná aplikace se skládá ze tří hlavních částí, které de-facto odpovídají hlavním funkcionalitám SDK pro dron Tello. První částí je rozhraní pro zadávání příkazů, dalším jsou statistiky o stavu dronu a třetím je příjem dat z kamery dronu. Vytvořený program se nachází na přiloženém CD (viz Příloha 1).

Na obrázku níže přikládám ukázkou, jak je celé rozhraní spouštěno. Základním souborem se vzhledem je *index.html* a pomocí funkce *loadURL* jej do aplikace zavedeme.

```
const electron = require('electron');

const {app, BrowserWindow} = electron;
let mainWindow;

app.on('ready', function(){
  mainWindow = new BrowserWindow({width: 1024, height: 850}); // nastavení velikosti okna
  mainWindow.loadURL(`file://${dirname}/index.html`); // nastavení html se stránkou
  mainWindow.setMenuBarVisibility(false); // vypnutí menu - nevhodné pro aplikaci
});
```

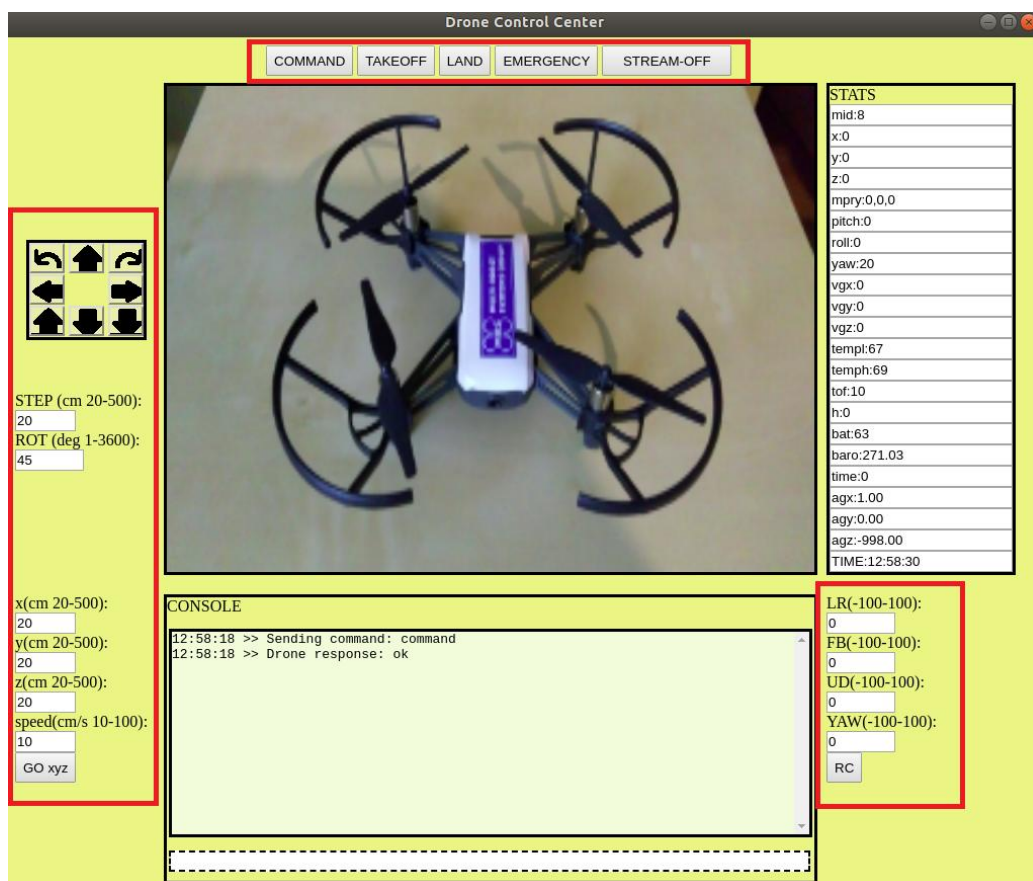
Obrázek č. 10: Ukázkou kódu, inicializace GUI [vlastní zpracování].

## 4.2. Příkazové rozhraní

Nejprve musíme zařízení, na kterém aplikace běží, připojit k wifi hotspotu, který dron vysílá.

Pro aktivaci dronu je nejprve nutné zaslat příkaz „command“, který přepne dron do SDK-módu. Příkazy se posílají na adresu 192.168.10.1, port 8889. Po úspěšném zaslání příkazu „command“ je možné zadávat další příkazy.

Hlavními příkazy jsou dále „takeoff“ pro vzletnutí, „land“ pro přistání. V případě, že by se dron stal neovladatelným z jakéhokoliv důvodu, jsem implementoval příkaz „emergency“, který po vyslání okamžitě vypne motory dronu. Poslední příkaz v této hlavní nabídce je tlačítko pro zapnutí videostreamu, které se dynamicky mění, podle toho, zda je stream zapnutý, či nikoliv.



Obrázek č. 11: GUI Aplikace, zvýrazněné příkazové rozhraní [vlastní zpracování].

Základní poziční příkazy jsem implementoval pomocí tlačítek, uspořádaných do vzhledu joysticku. Tyto příkazy mají jako argument vzdálenost, o kterou se má dron posunout. SDK tuto vzdálenost omezuje do intervalu 20 cm až 5 m. Vzdálenost pro příkazy tohoto typu si uživatel aplikace navolí pod joystickem. Dále se pod joystickem nastavuje úhel, o který se dron pootočí, klikneme-li na tlačítko pro rotaci dronu.

Níže přikládám ukázkou funkce, která posílá příkaz na příslušnou adresu a port. V proměnné *data*, zadané jako vstupní argument, je již zformulovaný příkaz ve formě textového řetězce podle popisu komunikačního protokolu v kapitole 3. Před samotným odesláním je příkaz pomocí funkce *write\_opt* vypsan do konzole v grafickém rozhraní.

```
/**
 * funkce pro poslání jakéhokoliv příkazu
 * @param {*} data = textový příkaz
 */
function send_command(data) {
  write_opt("Sending command: " + data); //vypis zpravu do konzole v GUI
  const message = Buffer.from(data);
  client.send(message, 0, message.length, 8889, '192.168.10.1', function (err, bytes) {
    if (err) throw err;
  });
  sleep(200);
}
```

Obrázek č. 12: Ukázkou kódu, funkce zajišťující poslání příkazu [vlastní zpracování].

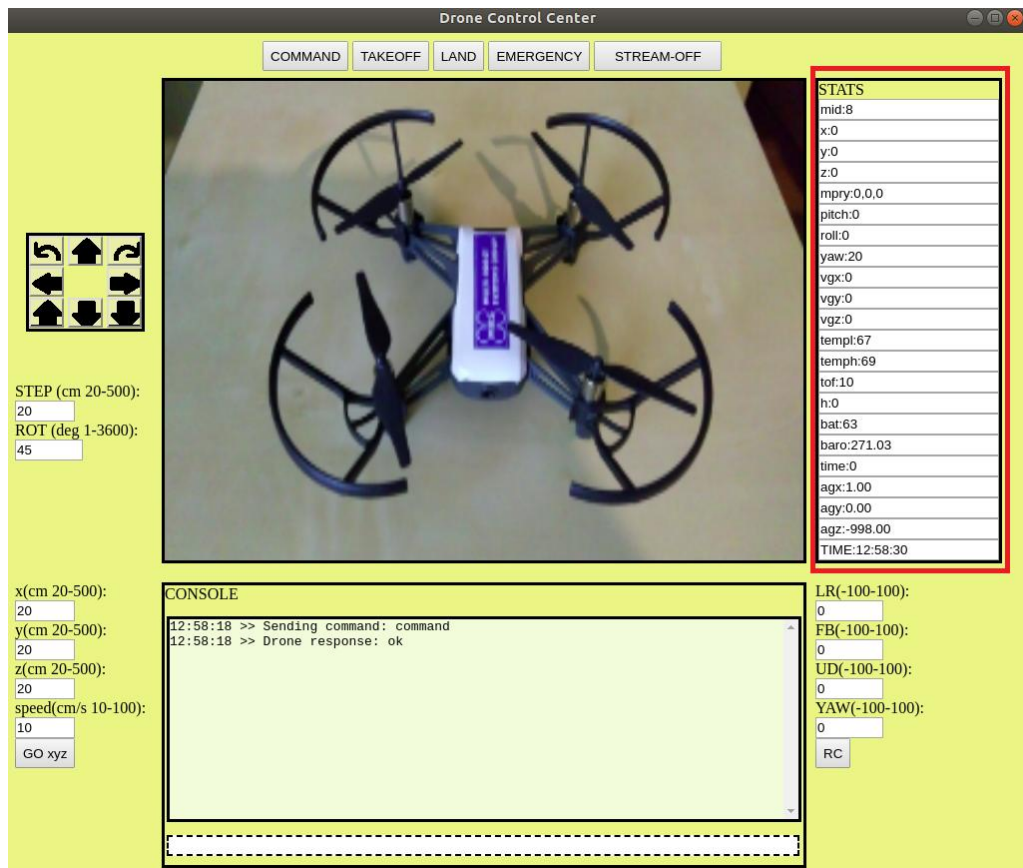
Dalšími příkazy, jež byly do rozhraní doplněny, jsou „rc“ a „go“. První příkaz umožňuje přímo simulovat RC ovládání pomocí 4 hodnot, které mají interval od -100 do 100. První hodnota je „vlevo-vpravo“, druhá „vpřed-vzad“, třetí nastavuje výšku dronu a poslední natočení okolo osy z. Po stisknutí tlačítka RC se vyšle příkaz s příslušnými hodnotami a hodnoty v polích se nastaví zpět na 0. Pokud uživatel nastaví hodnoty špatně a dron se začne pohybovat, kam by neměl, stačí znovu stisknout tlačítko RC a dron se zastaví na místě.

Druhý příkaz „go“ již ve svém názvu naznačuje svůj účel. Pomocí tohoto příkazu lze dron posunout do konkrétního bodu, zadaného třemi souřadnicemi (x, y, z). Souřadnice jsou počítány z aktuální pozice. Čtvrtým argumentem lze nastavit rychlost posunu, která může být až 1m/s. Vzdálenostní limity odpovídají ostatním pozičním příkazům, tj. minimálně 20 cm a maximálně 5 m.

SDK dále obsahuje příkazy pro čtení několika údajů o stavu dronu, nicméně tyto informace mi poskytuje již stavové rozhraní (viz níže), a proto tyto příkazy nebyly do aplikace implementovány.

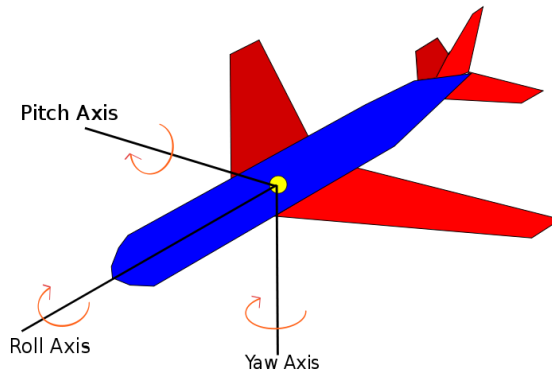
### 4.3. Stav dronu

Další neméně důležitou komponentou je rozhraní informující o aktuálním stavu dronu. Po vyslání příkazu „Command“ jsou data vysílány na portu 8890. SDK manuál definuje formát textového řetězce, který dron vysílá. Avšak při testování aplikace a parsování řetězce jsem zjistil, že je uvedený formát neúplný, protože dron jako prvních 5 údajů poskytuje hodnoty pro využití tzv. „mission pad“, což jsou hrací desky, které umožňují dronu reagovat na znaky na deskách.



Obrázek č. 13: GUI Aplikace, zvýrazněné stavové rozhraní [vlastní zpracování].

Dále rozhraní informuje o stavu rotace dle jednotlivých os, které se označují „pitch“, „roll“ a „yaw“. Jejich význam je znázorněn na obrázku níže.



Obrázek č. 14: Popis rotace dle jednotlivých os. Převzato z [10].

Klíčovými hodnotami, které rozhraní také poskytuje, jsou stav baterie, minimální a maximální teplota stroje, doba provozu a rychlostní informace.

Níže můžeme vidět ukázkou kódu, který zajišťuje odchycení a zobrazení dat, získaných z portu 8890. Jednotlivá data přicházejí ve formě textového řetězce a jsou oddělena znakem „;“, což umožňuje efektivní zpracování jednotlivých hodnot. Jako doplňkovou hodnotu, kterou jsem považoval za užitečnou, ač není přímo získána z rozhraní, jsem doplnil mezi statistiky aktuální čas. Tuto hodnotu můžeme využít, pokud chceme porovnat, kolik uběhlo času od zadání určitého příkazu, protože záznam o každém odeslaném příkazu je viditelný v konzoli grafického rozhraní aplikace.

```

/**
 * zaznam stavu dronu dle informaci ze stavoveho socketu
 */
stat.on('message', (msg, rinfo) => {
  var separated = msg.toString().split(";");
  //console.log(separated);
  var i;
  for (i = 0; i < 21; i++) {
    var output = document.getElementById("s" + i);
    output.value = separated[i];
  }
  var output = document.getElementById("s21");
  output.value = 'TIME:' + get_actual_time(); // posledni hodnota cas - ziskan mimo rozhrani
  sleep(1000)
});

```

Obrázek č. 15: Ukázka kódu, zpracování dat informujících o stavu dronu [vlastní zpracování].

## 4.4. Kamerové rozhraní

Třetí klíčovou částí je kamerové rozhraní. Rozhraní se rovněž aktivuje po zadání příkazu „command“, nicméně je poté nutné poslat dále příkaz „streamon“. Následně



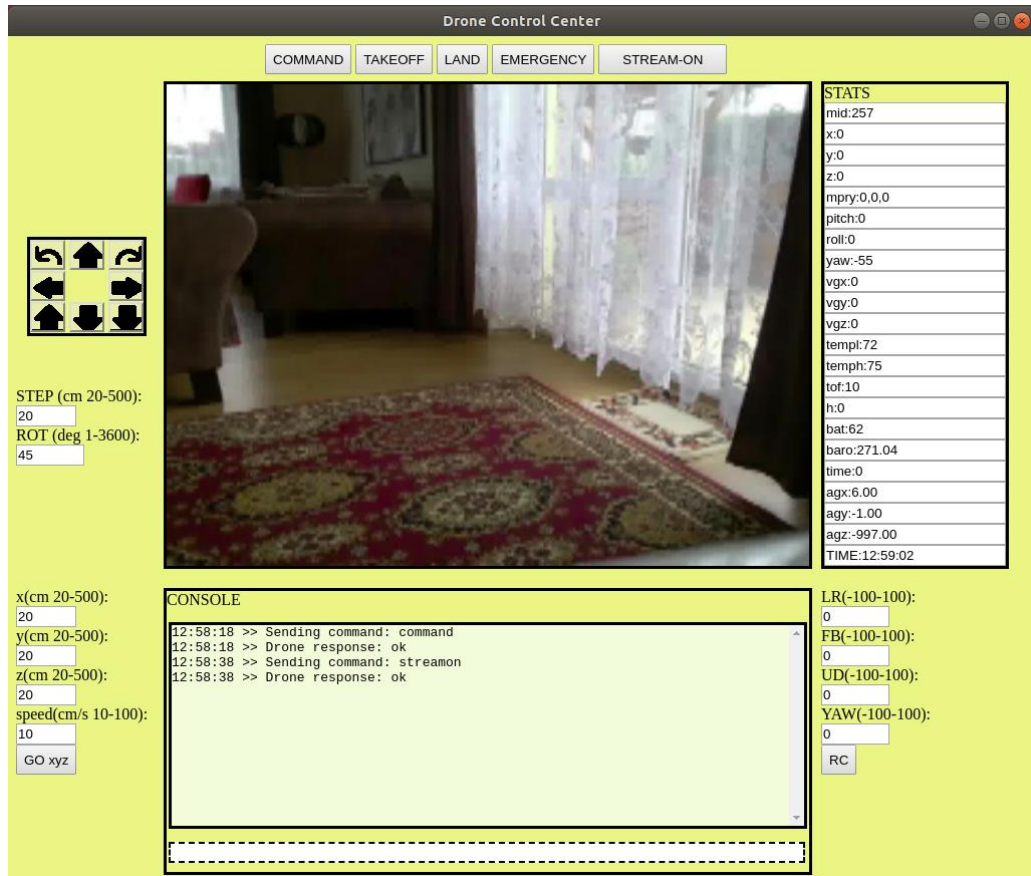
začne dron vysílat obraz na portu 11111 ve formátu H264. V samostatném projektu celou komponentu zastřešovala knihovna „OpenCV“, která slouží k získání a zpracování obrazu mimo jiné i z UDP streamu. NodeJS má k dispozici také obdobu této knihovny s názvem „Opencv4NodeJS“. Při testování jsem však zjistil, že při použití knihovny se vyskytly problémy s navázáním spojení a vytvořením „VideoCapture“. Při pokusu o navázání spojení se program ukončil bez jakékoliv chyby i v případě, že byla kritická část kódu uzavřena do try-catch bloku.

Knihovnu „Opencv4NodeJS“ jsem tedy nemohl ke svému účelu použít. Alternativu jsem našel v knihovně „ffmpeg“, která je rovněž dostupná i pro příkazovou řádku. Spuštěná jako „child process“ byla knihovna schopna navázat UDP spojení s portem 11111 a rovněž získat datové rámce s obrazem z kamery dronu. Získaný obraz jsem poté přeškáloval a vykreslil na připravené plátno („canvas“) v grafickém rozhraní aplikace. V případě ukončení přenosu obrazu z kamery pomocí příkazu „streamoff“ se na plátno vykreslí defaultní nákres dronu. Detail inicializace knihovny „ffmpeg“ a zpracování přijatého obrazu z kamery dronu si můžeme prohlédnout na obrázku níže.

```
document.getElementById("streaming").value = "STREAM-ON";
send_command("streamon");
drone_stream = 1;
ffmpeg= require('child_process').spawn("ffmpeg", [
  "-re", // nastavení standardni framerate
  "-y", // prepisani vystupu
  "-i", // URL vstup
  "udp://192.168.10.1:11111?fifo_size=1000000", // nastaveni adresy a portu
  "-preset",
  "ultrafast",
  "-f", // format souboru
  "mjpeg",
  "pipe:1"
]);
ffmpeg.stdout.on('data', function (data) {
  var frame = new Buffer(data).toString('base64');
  //console.log(frame);
  try { // vykresli obraz
    var canvas = document.getElementById('strim');
    var context = canvas.getContext('2d');
    var imageObj = new Image();
    imageObj.src = "data:image/jpeg;base64," + frame;
    imageObj.onload = function () {
      context.drawImage(imageObj,0,0 ,canvas.width, canvas.height);
    }
  } catch (error) {
    console.log(error);
  }
});
```

Obrázek č. 16: Ukázka kódu, zpracování kamerového rozhraní [vlastní zpracování].

Je nutné poznamenat, že tento způsob zpracování obrazu vytváří zpoždění mezi reálným okamžikem a momentem, kdy se rámeček objeví v aplikaci. Toto zpoždění odpovídá přibližně 10 sekundám.



Obrázek č. 17: GUI Aplikace, ukázka kamerového záznamu [vlastní zpracování].

## 5. Zpracování obrazu z dronu

Dle M. Radovic z firmy Drone Industry Insights lze rozlišit až 6 stupňů samostatnosti dronů, počínaje výhradně manuálním řízením a plnou autonomií konče [11].

DRONE INDUSTRY INSIGHTS						
THE 5 LEVELS OF DRONE AUTONOMY						
Autonomy Level	Level 0	Level 1	Level 2	Level 3	Level 4	Level 5
Human Involvement						
Machine Involvement						
Degree of Automation	No Automation	Low Automation	Partial Automation	Conditional Automation	High Automation	Full Automation
Description	Drone control is 100% manual.	Pilot remains in control. Drone has control of at least one vital function.	Pilot remains responsible for safe operation. Drone can take over heading, altitude under certain conditions.	Pilot acts as fall-back system. Drone can perform all functions 'given certain conditions'.	Pilot is out of the loop. Drone has backup systems so that if one fails, the platform will still be operational.	Drones will be able to use AI tools to plan their flights as autonomous learning systems.
Obstacle Avoidance	NONE	SENSE & ALERT		SENSE & AVOID	SENSE & NAVIGATE	
Source: DRONEII.com	Date: March 12 <sup>th</sup> 2019					DRONEII.COM DRONE INDUSTRY INSIGHTS

Obrázek č. 18: Úrovně samostatnosti dronů dle DroneII.com Převzato z [11].

0. stupeň – *plně manuální ovládání*, oblasti využití: závodní a rekreační;

1. stupeň – *slabá autonomie*, což znamená, že automatická je minimálně jedna funkce, oblasti využití: průzkum a údržba, lokalizace, monitoring, fotografování a filmování;

2. stupeň – *částečná autonomie*, když za určitých podmínek přebírá řízení autopilot, nicméně konečné rozhodnutí je v gesci pilota, oblasti využití: mapování terénu, měření;

3. stupeň – *podmíněná autonomie*, když jsou drony naprogramovány na práci v určitých podmínkách, jsou schopni pracovat autonomně a pilot slouží pouze jako pojistka při selhání, oblasti využití: mapování terénu, průzkum, dodej zboží;

4. stupeň – *vysoká autonomie*, dron může pracovat samostatně, ale zároveň může být ovládán pilotem, oblasti využití: fotografování a filmování;

5. stupeň – *plná autonomie*, když dron pracuje plně samostatně, bez jakéhokoliv zásahu pilota. Tento model zatím nebyl realizován, je to však otázkou budoucnosti.

Abychom dosáhli určité míry automatizace (stupeň 2 – 5), je nutné, aby byl dron vybaven výpočetní jednotkou (procesorem). Základním problémem je vyhýbání se překážkám tak, aby dron nehavaroval. Problém lze řešit dvěma způsoby: hardwarově pomocí dotykových, ultrazvukových nebo infračervených senzorů, které mohou zaznamenat překážku. Druhé řešení, softwarové, je právě pomocí dat z kamery dronu.

Hardwarové řešení je u dronu „Ryze Tello“ prakticky nerealizovatelné, protože dron neposkytuje mnoho senzorů tak, aby se daly využít pro autonomní řízení. Maximálně můžeme dosáhnout toho, že částečně budeme schopni regulovat minimální a maximální výšku letu, díky infračervenému senzoru ve spodní části dronu.

Co se týče druhé – softwarové možnosti, za určitých okolností lze dosáhnout částečné autonomie, dle mého názoru s ohledem na vybavenost dronu, maximálně třetího stupně dle škály zmíněné výše.

Důležité je nalézt způsob, jak z 2D obrazu poznat, že má dron před sebou překážku, které je třeba se vyhnout. Řešení by mohla poskytnout *metoda hledání významných bodů v obraze*. Pod tímto pojmem se označuje několik algoritmů, které na základě různých parametrů hledají v obraze body a plochy, které se liší od svého okolí.



Obrázek č. 19: Postup zpracování obrazu. Převzato z [12].

Jak je znázorněno na schématu výše, musíme nejprve obraz převést na matici, se kterou budeme dále pracovat. Poté je třeba provést korekce obrazu, protože obraz může být zkreslený. Aplikací různých filtrů zredukujeme šum a zaostříme obraz, aby byly dobře viditelné hrany. Dále pak musíme separovat pozadí od cíleného objektu.

Běžně využívané algoritmy, které bych rád zmínil, jsou algoritmy „SURF“ („Speed Up Robust Features“) a „SIFT“ („Scale Invariant Feature Transform“).

První zmíněný algoritmus „SURF“ vyvinul Herbert Bay v roce 2006 [21]. Jeho práce byla inspirována algoritmem „SIFT“, jenž vznikl o 2 roky dříve. Algoritmus „SIFT“ využívá frekvenční spektrum k určení vzdálenosti objektu na obraze. Problémem u 2D obrazu je, že nejsme na první pohled schopni určit vzdálenost objektu, tj. třetí dimenzi. Pokud však analyzujeme frekvenční spektrum, zjistíme, že menší frekvence znamená menší měřítko a tím větší detail, kdežto vyšší frekvence zaznamenávají menší detail. Dle Baye [21, str. 4] „SURF“ používá k nalezení významných bodů aproximaci determinantu Hessiany matice (matice druhých parciálních derivací) nad tzv. integrálním obrazem, což je tabulka, ve které každá buňka odpovídá součtu buněk nacházejících se směrem doleva a nahoru.

Z hlediska programování je nejlepší využít knihovnu „OpenCV“, která je dostupná jak pro C++ a Javu, tak pro Python nebo Javascript. Knihovna má již implementované funkce pro zpracování obrazu a detekce významných bodů. S jejich pomocí detekujeme překážku a můžeme na ní případně zareagovat. Tuto teoretickou možnost jsem bohužel nemohl otestovat prakticky z toho důvodu, že jak jsem zmínil výše (viz. 3.2.3), se vyskytly problémy s propojením UDP a knihovny OpenCV v NodeJS.

Dalším problémem, který komplikuje autonomní chování dronu je zpoždění kamery. Systém v dronu Ryze Tello je uzavřený a lze programovat pouze externí ovládací program. Pokud by tento program měl pracovat interaktivně, bylo by potřeba výrazně zkrátit zpoždění obrazu, což se mi v mé práci nepovedlo.

## 6. Závěr

V bakalářské práci, věnované dronu „Ryze Tello“, jsem se nejprve zaměřil na historii bezpilotních letounů (dronů). První bezpilotní letouny se začaly vyvíjet výlučně pro vojenské využití, nicméně díky zpřístupňování armádních technologií veřejnosti na počátku 21. století byl objeven obrovský potenciál ve využití dronů v komerční sféře (od využití filmovými producenty po přepravu zboží a využití pro širokou veřejnost). Jedním z novějších modelů, určeným pro širokou veřejnost s cenou okolo 100 USD, je právě dron „Ryze Tello“ od společností Ryze Tech, DJI a Intel.

Cílem práce bylo navrhnout webové rozhraní pro komunikaci s dronem „Ryze Tello“ a jeho řízení. Ve třetí kapitole jsem popsal klíčové údaje týkající se SDK dronu, ze kterého moje aplikace vychází. Samotný program je přiložen na CD v příloze. Pro jeho správnou funkčnost je nutné nejprve nainstalovat „NodeJS“, „ffmpeg“ a „npm“. Potřebné knihovny získáme pomocí příkazu „npm install“. Poté by program měl fungovat bez potíží a v rámci specifikací uvedených ve třetí kapitole.

Dalším úkolem mé práce bylo otestování možnosti automatizace řízení na základě informace z obrazu dronu. Ve čtvrté kapitole jsem se zaměřil na teoretické možnosti, které by tento úkol mohly řešit. Na základě prostudované literatury, týkající se tématu zpracování obrazu, jsem zjistil, že by existovala teoretická možnost využití za tímto účelem populární knihovny „OpenCV“. Prakticky se mi však tuto možnost otestovat nepodařilo, protože se vyskytly komplikace s propojením UDP a knihovny OpenCV v NodeJS, které knihovnu „OpenCV“ má. Nepodařilo se mi pomocí této knihovny navázat spojení s dronem a získat s její pomocí záznam z kamery. Příčinou jsou s nejvyšší pravděpodobností omezení UDP v javascriptu. Další překážkou automatizace je zpoždění obrazu z kamery, které se mi nepodařilo odstranit.

## Zdroje / použitá literatura

- [1] LeonardodaVinci.net [Online] <http://www.leonardo-da-vinci.net/helicopter/>
- [2] flyingmachines.org [Online] <http://www.flyingmachines.org/no5.html>
- [3] TURI, Jon, Tesla's toy boat: A drone before its time, *Engadget*, 19. 1. 2014 [Online] <https://www.engadget.com/2014/01/19/nikola-teslas-remote-control-boat/>
- [4] Foto Kettering Bug [Online] [https://en.wikipedia.org/wiki/Kettering\\_Bug](https://en.wikipedia.org/wiki/Kettering_Bug)
- [5] The Mother of all Drones. How the pilotless dde havilland queen bee spawned the nemesis of Al Qaeda. *Vintage wings of Canada* [Online] <http://www.vintagewings.ca/VintageNews/Stories/tabid/116/articleType/ArticleView/articleId/484/The-Mother-of-All-Drones.aspx>
- [6] ROTHMAN, Wilson, Unmanned Warbots of WWI and WWII, *Gizmodo*, 25. 3.2009 [Online] [https://www.gizmodo.com.au/2009/03/unmanned\\_warbots\\_of\\_wwi\\_and\\_wwii-2/](https://www.gizmodo.com.au/2009/03/unmanned_warbots_of_wwi_and_wwii-2/)
- [7] Foto DJI Ryze Tello Mini Drone [Online] <https://www.verizonwireless.com/products/dji-ryze-tello-mini-drone/>
- [8] NOVÁK, Jan A., Příběh zakladatele DJI: ze studentské koleje miliardářem, *Droneweb*, 24. 5. 2018 [Online] <http://droneweb.cz/aktuality/item/244-dji-frank-wang>
- [9] Foto DJI Mavic Pro [Online] <https://hely-shop.co.uk/wp-content/uploads/2018/07/mavickpro.jpg>
- [10] Foto Yaw Axis [Online] [https://upload.wikimedia.org/wikipedia/commons/1/16/Yaw\\_Axis.svg](https://upload.wikimedia.org/wikipedia/commons/1/16/Yaw_Axis.svg)

- [11] RADOVIC, Millie, Tech Talk: Untangling The 5 Levels of Drone Autonomy, *DroneII*, 7. 3. 2019 [Online] <https://www.droneii.com/drone-autonomy>
- [12] RICHTER, Miloslav, *Detekce geometrických tvarů, Houghova transformace* [Online]  
[http://www.uamt.feec.vutbr.cz/~richter/vyuka/0910\\_pov/POV\\_HT\\_objekty\\_d.pdf](http://www.uamt.feec.vutbr.cz/~richter/vyuka/0910_pov/POV_HT_objekty_d.pdf)
- [13] KARAS, Jakub; TICHÝ, Tomáš: *Drony*, Computer Press, 2016, ISBN 978-80-251-4680-4.
- [14] HOLMAN, Brett, The first air bomb: Venice, 15 July 1849, *Airminded*, 22.8.2009 [Online] <https://airminded.org/2009/08/22/the-first-air-bomb-venice-15-july-1849/>
- [15] SHAW, Ian G. R., The Rise of the Predator Empire: Tracing the History of U.S. Drones, *Understanding Empire*, 2014 [Online]  
<https://understandingempire.wordpress.com/2-0-a-brief-history-of-u-s-drones/>
- [16] Honoris causa ČVUT [Online]  
<https://web.archive.org/web/20111104055806/http://www.cvut.cz/cs/historie/honoris-causa>
- [17] LAJČÁK, Martin, Za záchranu Notre-Dame vdáčíme aj dronom DJI. Zohrali dôležitú úlohu, *Fontech*, 19. 4. 2019 [Online] <https://fontech.startitup.sk/za-zachranu-katedraly-notre-dame-mozno-vdacit-aj-dronom-dji-mavic-pro/>
- [18] Tello Vs. Spark: Which DJI Starter Drone is Right For You?, *Half Chrome*, 14. 1. 2018 [Online] <https://www.halfchrome.com/tello-vs-spark/>
- [19] I. COLEMAN, DJI Tello Quadcopter Drone, *Dronethusiast*, 2018 [Online]  
<https://www.dronethusiast.com/dji-tello-quadcopter-drone/>
- [20] Tello SDK 1.3.0.0 [Online] [https://dl-cdn.ryzerobotics.com/downloads/tello/20180910/Tello%20SDK%20Documentation%20EN\\_1.3.pdf](https://dl-cdn.ryzerobotics.com/downloads/tello/20180910/Tello%20SDK%20Documentation%20EN_1.3.pdf)
- [21] BAY, Herbert; TUYTELAARS, Tinne; VAN GOOL, Luc, *SURF: Speeded Up Robust Features*, 2006, [Online] <https://www.vision.ee.ethz.ch/~surf/eccv06.pdf>



# Seznam příloh

## Příloha č. 1: CD

K bakalářské práci je přiloženo CD s aplikací, které je práce věnovaná.

Struktura souborů:

*./BP – kořenový adresář*

*./BP/images – složka s obrázky, které aplikace potřebuje k běhu*

*./BP/images/arr\_ccw.png*

*./BP/images/arr\_cw.png*

*./BP/images/arr\_down.png*

*./BP/images/arr\_down2.png*

*./BP/images/arr\_left.png*

*./BP/images/arr\_right.png*

*./BP/images/arr\_up.png*

*./BP/images/arr\_up2.png*

*./BP/images/drone\_def.jpg – defaultní obraz dronu*

*./BP/drone.js – spouštěč aplikace*

*./BP/drone\_main.js – kód aplikace - funkce*

*./BP/index.css – css soubor upravující vzhled index.html*

*./BP/index.html – kód aplikace – základní zobrazení*

*./BP/package.json*

*./BP/package-lock.json*