



ZADÁNÍ BAKALÁ SKÉ PRÁCE

Název:	Knihovna pro bezdrátové ovládání GoPro kamery z prost edí Wolfram Mathematica
Student:	Michal Sládek
Vedoucí:	Ing. Zden k Buk, Ph.D.
Studijní program:	Informatika
Studijní obor:	Informa ní technologie
Katedra:	Katedra po íta ových systém
Platnost zadání:	Do konce letního semestru 2017/18

Pokyny pro vypracování

Prozkoumejte možnosti dálkového ovládání GoPro kamery (ideální model Hero 3+), navrhn te a implementujte knihovnu v jazyce Wolfram Language, která tuto funkcionalitu zp ístupní v prost edí Wolfram Mathematica. Knihovna by m la nabízet funkce pro nastavení režimu snímání (fotografie, video), nastavení parametr obrazu (rozlišení, kompenzace expozice, FPS, apod.), dálkové ovládání spoušt a zp ístupn ní dat z kamery. Prove te a zdokumentujte testy implementovaných funkcionalit.

Seznam odborné literatury

Dodá vedoucí práce.

prof. Ing. Róbert Lórencz, CSc.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
d kan

V Praze dne 4. února 2017

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
KATEDRA POČÍTAČOVÝCH SYSTÉMŮ



Bakalářská práce

Knihovna pro bezdrátové ovládání GoPro kamery z prostředí Wolfram Mathematica

Michal Sládek

Vedoucí práce: Ing. Zdeněk Buk, Ph.D.

14. května 2017

Poděkování

V první řadě bych chtěl poděkovat svému vedoucímu Zdeňkovi Bukovi, že mi umožnil volbu vlastního tématu a vyvinul kvůli tomu nemalé byrokratické úsilí. Dále bych chtěl poděkovat celé své rodině za nekonečnou podporu během celého mého studia, obzvláště pak Kateřině Sládkové, která strávila velké množství času a energie při čtení výplodů mé fantazie a pokusila se usměrnit mé tvůrčí myšlenky k čitelnějšímu textu. Rád bych též poděkoval svým kamarádům, jmenovitě pak Melounkovi, jelikož mne rozptylovala od problémů týkajících se mého studia a prožívala souběžně se mnou strastiplné chvíle při tvorbě své bakalářské práce.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 14. května 2017

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2017 Michal Sládek. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.

Odkaz na tuto práci

Sládek, Michal. *Knihovna pro bezdrátové ovládání GoPro kamery z prostředí Wolfram Mathematica*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017. Dostupný také z WWW: <https://github.com/slademi2/goProController>.

Abstrakt

Cílem této práce je prozkoumat možnosti vzdáleného ovládání kamery GoPro, navrhnout a implementovat knihovnu v jazyce Wolfram Language, která tuto funkcionalitu zpřístupní v prostředí Wolfram Mathematica. Výsledná knihovna nabízí funkce pro nastavení režimu snímání (pro fotografie i videa), nastavení parametrů obrazu (rozlišení, kompenzace expozice, FPS), dálkové ovládání kamery a zpřístupnění dat.

Klíčová slova GoPro, Wolfram Language, Wolfram Mathematica, vzdálené ovládání, HTTP

Abstract

The aim of this work is to explore the possibilities of remote control for the GoPro cameras, to design and implement a package of functions using the Wolfram Language, which will enable this functionality in the environment of Wolfram Mathematica. The resulting package offers a setting of the camera mode (for photographs and videos), a setting of the image parameters (resolution, exposure compensation, FPS), remote control of the camera and accessibility of the data.

Keywords GoPro, Wolfram Language, Wolfram Mathematica, remote control, HTTP

Obsah

Úvod	1
1 Analýza	5
1.1 Existující řešení	5
2 Návrh vlastního řešení	7
2.1 Porovnání modelů kamer	12
3 Řešení	17
3.1 Rozvržení knihovny	17
3.2 Proměnné a konstanty	20
3.3 Funkce	21
3.4 Práce s pořízenými daty	27
3.5 Dokumentace	28
4 Výsledky	31
4.1 Návod k použití knihovny	31
4.2 Testování	36
Závěr	41
Literatura	43
A Seznam použitých zkratk	45
B Obsah přiloženého CD	47

Seznam obrázků

2.1	Možnosti nastavení kamery pomocí aplikace Capture.	8
2.2	Zachycená komunikace mezi kamerou GoPro a aplikací Capture. . .	8
2.3	Zachycená komunikace mezi kamerou GoPro a aplikací Capture. . .	9
2.4	Stránka na adrese http://10.5.5.9:8080/	10
2.5	Stažení pořízeného obrázku přímo do Wolfram Mathematica. . . .	11
3.1	Diagram načítání vedlejších zdrojových souborů knihovny <i>GoPro-Controller</i>	19
3.2	Struktura souboru s uloženým nastavením kamery.	27
3.3	Našeptávání krátkého popisu funkce ve Wolfram Mathematica. . .	29
4.1	Vypsání uživatelských proměnných obsahujících cestu pro instalaci balíčku.	32
4.2	Ukázky možností načtení balíčku <i>GoProController</i> do Wolfram Mathematica.	32
4.3	Funkce dostupné ihned po načtení balíčku <i>GoProController</i>	33
4.4	Nastavení knihovny <i>GoProController</i> před jejím použitím.	33
4.5	Diagram nastavení knihovny <i>GoProController</i> před jejím použitím. . .	34
4.6	Nastavování parametrů kamery pomocí funkce <i>goProSet</i>	36
4.7	Ukázka testování složení url adresy.	38
4.8	Ukázka testování nastavování za přítomnosti kamery.	40

Seznam tabulek

2.1	Popis parametrů v url adrese pro ovládání kamery.	11
2.2	Jednotlivé parametry pro nastavení kamer generace HERO2, HERO3, HERO3+.	13
2.3	Jednotlivé parametry pro nastavení kamer generace HERO4.	14
2.4	Porovnání kamer generace HERO3+.	15
3.1	Funkce v hlavním souboru GoProController.m.	18
3.2	Vedlejší zdrojové soubory a modely kamer, které ovládají.	18
3.3	Snímací módy jednotlivých generací kamer GoPro.	19
3.4	Prefixy názvů funkcí v knihovně GoProController.	22
4.1	Funkce sloužící k ovládání kamery GoPro	35
4.2	Funkce pro práci se soubory na kameře GoPro.	37

Úvod

V dnešní době mnoho lidí používá takzvané outdoorové kamery vyvinuté k použití v náročných podmínkách. Tyto kamery jsou hojně využívány milovníky sportu či turisty pro dokumentaci zážitků. Jejich hlavní výhodou je nízká hmotnost, která se většinou pohybuje pod hranicí sta gramů. Mezi jejich uživatele ovšem patří i lidé, kteří je používají pro sběr obrazových dat, jež následně zpracovávají pomocí jiných programů. Je velice nepraktické a nekomfortní ovládat kameru a stahovat pořízená data manuálně či pomocí kabelu, obzvláště pokud outdoorové kamery nabízejí snadnější způsob ovládání. Tato zařízení většinou podporují dálkové ovladače (pořízené spolu s kamerami) nebo je lze ovládat pomocí mobilních aplikací s využitím WiFi sítě, kterou vysílají (ať už zabudovaným vysílačem nebo s využitím dalšího příslušenství).

Jednou z největších a nejúspěšnějších firem v oblasti outdoorových kamer je společnost GoPro, Inc (dále jen GoPro). Tato firma začala prodávat první outdoorovou kameru v roce 2004 a od té doby se její produkty řadí k nejoblíbenějším. Bohužel nebyl dosud vydán žádný oficiální software pro ovládání GoPro kamer přímo z počítače. Takové programy samozřejmě existují, ale jsou buď zastaralé nebo není možné rozšiřovat jejich funkčnost. Proto je cílem této práce implementovat pomocí jazyku Wolfram Language takové funkce, které umožní bezdrátově ovládat a nastavovat kamery GoPro a také zpřístupnit zpracovávání dat z kamery v prostředí Wolfram Mathematica. Výsledná knihovna bude volně přístupná a bude možné ji dále rozšiřovat.

Cíle práce

Cílem této práce je prozkoumání možností bezdrátového ovládání kamer Go-Pro, výběr nejvhodnější možnosti a její implementace v knihovně pro prostředí Wolfram Mathematica. V práci se zaměříme hlavně na kameru Go-Pro HERO3+ Black edition, její bezdrátové ovládání a nastavení. Vytvořená knihovna v jazyce Wolfram Language bude nabízet funkce pro nastavení kamery a její následné ovládání, bude tedy možné vzdálené ovládání spouště, nastavení většiny parametrů pro video i fotografii či stažení pořízeného materiálu. Dalším cílem je otestování implementované knihovny a vytvoření dokumentace pro dokumentační centrum Wolfram Mathematica.

Analýza

1.1 Existující řešení

V současné době existuje mnoho nástrojů na ovládání kamer GoPro, nejvíce jich nalezneme v podobě mobilních aplikací, které ovládají kamery pomocí WiFi sítě vysílané samotnou kamerou. Nejpoužívanějším řešením je mobilní aplikace Capture (dříve GoPro App) vyvíjená přímo společností GoPro. Pro operační systém Android nalezneme velké množství dalších aplikací určených k ovládání GoPro kamer, například ProRemote (for GoPro), CamControl for GoPro, LIVE4 GoPro a další. Navíc lze tato zařízení ovládat pomocí bezdrátového ovladače poskytovaného výrobcem kamer.

1.1.1 Oficiální aplikace Capture

Mobilní aplikace Capture [1], dříve známa pod názvem GoPro App, vyvíjená přímo společností GoPro je velice mocným nástrojem pro ovládání kamer od téže společnosti. Aplikace je dostupná pro zařízení s operačním systémem Android či iOS a jedná se o jediné oficiální řešení určené ke vzdálenému ovládání kamer GoPro. Tento produkt umožňuje nastavování všech dostupných parametrů kamery, její ovládání i stahování pořízeného materiálu a podporuje všechny modely kamer GoPro, které umožňují ovládání pomocí WiFi ovladače.

1.1.2 Desktopové aplikace

Dalším logickým krokem k ovládání kamer GoPro je možnost ovládat zařízení z počítače. Společnost GoPro poskytuje programy Quik a GoPro Studio, ty jsou však určeny pouze k importování pořízených materiálů z kamery do počítače, popřípadě k úpravě videí či fotografií. Nicméně neexistuje žádný oficiální program určený k ovládání zařízení GoPro přímo z počítače.

1. ANALÝZA

Několik vývojářů v tomto směru sice pokročilo, ale jejich aplikace jsou buď neaktualizované pro nové kamery nebo nepřenositelné mezi operačními systémy či nerozšiřitelné ve smyslu dalšího zpracování dat z kamery. Navíc většina z těchto produktů je přístupná pouze v placené podobě. Z vyjmenovaných důvodů vyplývá potřeba programu, který by umožnil ovládání GoPro kamer z počítače, zpřístupnil by pořízené materiály pro další zpracování a byl by snadno rozšiřitelný pro další modely kamer.

Návrh vlastního řešení

V akademické i komerční sféře je často používán výpočetní systém Wolfram Mathematica. Množství funkcí, které obsahuje, se každým rokem rozrůstá a posouvá tak hranice své využitelnosti. Jednou z oblastí pokrytých tímto výpočetním systémem je zpracování a analýza obrazu (Image Processing & Analysis). Jedná se o obsáhlou podporu pro zpracování obrazu s využitím zabudovaných matematických a algoritmických schopností Wolfram Language.[2] Wolfram Mathematica podporuje síťovou komunikaci a je proto vhodným kandidátem k implementaci knihovny pro ovládání GoPro kamer. Data pořízená kamerou GoPro mohou být následně využita právě ve Wolfram Mathematica.

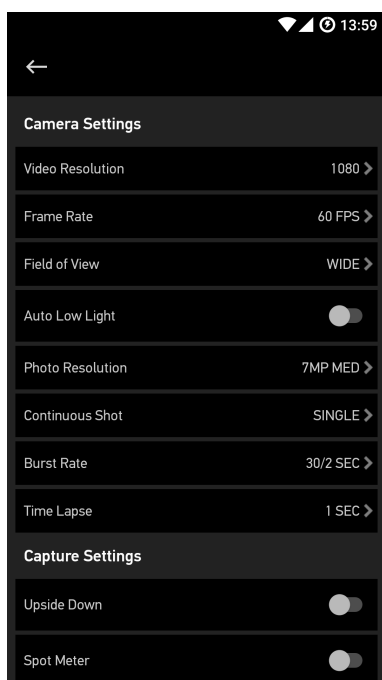
K samotnému vývoji použijeme Wolfram Workbench, nástroj pro editaci, vývoj a testování projektů v jazyce Wolfram Language, který je zároveň rozšířením pro vývojové prostředí Eclipse. Navíc nám umožní efektivně vytvořit a nasadit knihovnu funkcí pro Wolfram Mathematica. Tuto knihovnu pojmenujeme „GoProController“.

2.0.1 Způsob ovládání

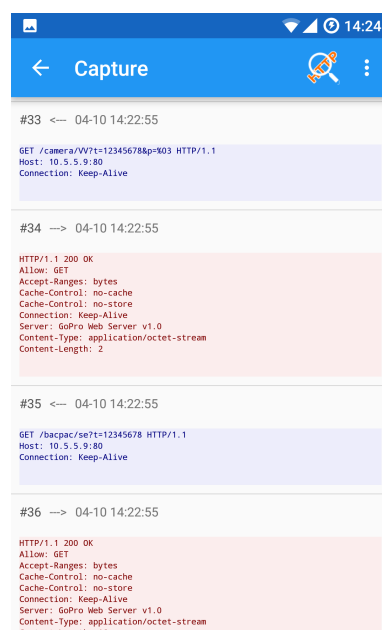
Jelikož je možné kameru ovládat pomocí mobilní aplikace, je velmi pravděpodobné, že to jde i z počítače. Prvním a stěžejním úkolem při návrhu knihovny pro ovládání kamer GoPro je tedy zjistit, jak se kamery ovládají a co je k tomu potřeba. K řešení tohoto úkolu jsme využívali dva různé postupy. Zprv jsme provedli analýzu chování aplikace Capture a zadruhé jsme využili dostupných informací o kamerách GoPro na Internetu.

V dnešní době existuje již poměrně velké množství kamer od společnosti GoPro a jejich ovládání i možnosti se značně liší. Tato práce vzniká za pomoci zařízení GoPro HERO3+ Black edition. Následující postupy a bádání byly prováděny na tomto modelu a později přeneseny na ostatní modely.

2. NÁVRH VLASTNÍHO ŘEŠENÍ



Obrázek 2.1: Možnosti nastavení kamery pomocí aplikace Capture.



Obrázek 2.2: Zachycená komunikace mezi kamerou GoPro a aplikací Capture.

2.0.1.1 Funkčnost aplikace Capture

K získání informace o tom, jak funguje ovládání GoPro kamer jsem využil mobilní aplikaci Capture, která je k jejich ovládání určena. Pro ovládání kamery je třeba se připojit k WiFi síti, kterou kamera vysílá. Po spuštění aplikace Capture a krátkém nastavení (vybrání požadovaného modelu kamery, zadání hesla k WiFi síti atd.) jsme schopni kameru ovládat. Můžeme nastavovat parametry zařízení jako je rozlišení, počet snímků za vteřinu a podobné, viz obrázek 2.1.

Dále pomocí aplikace Packet Capture zachytíme komunikaci mezi kamerou GoPro a aplikací Capture. Tato aplikace umožňuje sledovat provoz na síti a zachytávat odchozí i příchozí pakety, ukázka použití je vidět na obrázku 2.3. Pokud se podíváme dovnitř záznamu zjistíme, že komunikace mezi kamerou a aplikací Capture probíhá s využitím protokolu HTTP 1.1, kde aplikace pomocí metody GET posílá požadavky na hostitele 10.5.5.9 a port 80. Jak je patrné na obrázku 2.2, aplikace posílá požadavky s různými adresami url, například `http://10.5.5.9/camera/FS?t=12345678&p=%07` a `http://10.5.5.9/camera/fs?t=12345678`.

V obou uvedených adresách vidíme stejnou část „12345678“, jedná se o heslo k WiFi síti, kterou vysílá kamera GoPro a tudíž se bude lišit podle nastavení sítě. Pokud například první uvedenou adresu zadáme do interneto-

Icon	Source	Destination	Protocol	Time	Size
Capture	10.5.5.9:80	TCP	TCP	04-10 14:22:51	19 kB
Capture	10.5.5.9:80	TCP	TCP	04-10 14:22:51	346 B
Capture	10.5.5.9:80	TCP	TCP	04-10 14:22:51	669 B
Capture	10.5.5.9:80	TCP	TCP	04-10 14:22:49	346 B
Capture	10.5.5.9:80	TCP	TCP	04-10 14:22:47	No data

Obrázek 2.3: Zachycená komunikace mezi kamerou GoPro a aplikací Capture.

vého prohlížeče, zjistíme změnu v nastavení kamery, konkrétně změnu počtu snímků za vteřinu na hodnotu 60. U druhé adresy url nám již webový prohlížeč mnoho nepomůže.

Využijeme tedy funkcí Wolfram Mathematica, konkrétně funkci *HTTPRequest* k sestavení HTTP požadavku na určenou url adresu a funkci *URLExecute*, která vykoná vytvořený požadavek a importuje jakýkoliv vygenerovaný výsledek. Pokud tedy vykonáme příkaz:

```
URLExecute[HTTPRequest["http://10.5.5.9/camera/fs?t=12345678"]]
```

ve Wolfram Mathematica, vrátí se nám seznam obsahující dvě položky: {0,7}. První položka je pro nás nezajímavá, ale druhá položka obsahuje číslici 7, což je tatáž číslice, kterou jsme nastavili počet snímků za vteřinu na hodnotu 60.

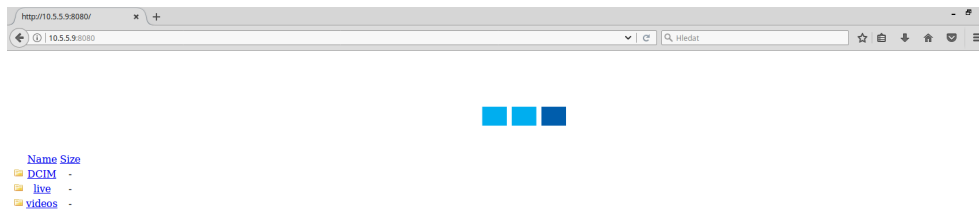
Povedlo se nám tedy zjistit nastavení kamery pomocí HTTP požadavku. To znamená, že kameru GoPro je nejen možné vzdáleně ovládat, ale zároveň jsme schopni zjistit, jak je kamera nastavena. Bádat ovšem po hodnotě každého parametru a jeho argumentech pro všechna nastavení kamery by bylo nesmírně časově náročné.

2.0.1.2 Internet

Kamerám GoPro se na Internetu věnuje veliké množství diskusních serverů a mnoho uživatelů. Nabízí se tedy možnost využít těchto stránek k získání informací o jejich ovládní. Na diskusních fórech jsem našel dotazy, zda lze ovládat kamery GoPro z počítače, zejména z důvodu zpřístupnění pořízených dat pomocí WiFi sítě vysílané kamerou. Celkem častou odpovědí je, že pořízené materiály jsou dostupné na adrese <http://10.5.5.9:8080/> [3].

Po připojení k WiFi síti vysílané kamerou a navštívení adresy <http://10.5.5.9:8080/> se nám zobrazí webová stránka kamery GoPro, která vypadá spíše jako webové rozhraní FTP serveru, viz obrázek 2.4. Tato stránka se

2. NÁVRH VLASTNÍHO ŘEŠENÍ



Obrázek 2.4: Stránka na adrese <http://10.5.5.9:8080/>.

liší pro každý model kamery, ale většinou jen v názvech složek. Konkrétně pro model HERO 3+ Black edition můžeme ve složce <http://10.5.5.9:8080/DCIM/100GOPRO/> nalézt pořizené záznamy.

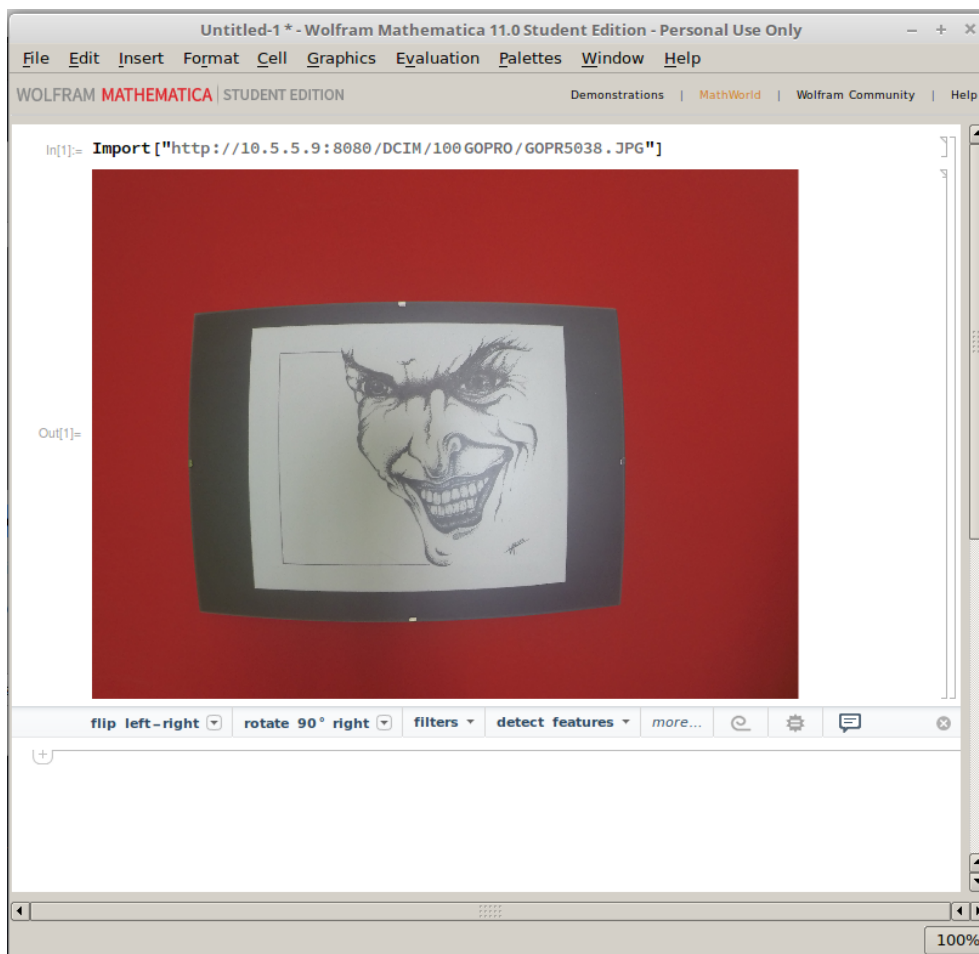
Je tedy možné stáhnout pořizené materiály. Ve Wolfram Mathematica toho dosáhneme využitím funkce *Import*. Tak stáhneme fotografii přímo do notebooku, jak můžeme vidět na obrázku 2.5. K ovládní kamery je potřeba trochu více.

Na serveru <https://github.com/> se nachází projekt věnovaný ovládní GoPro kamer. Nazývá se *goprowifihack* a je dostupný na adrese <https://github.com/KonradIT/goprowifihack> a obsahuje návody na ovládní kamer pro modely HERO2, HERO3, HERO3+ a HERO4 v různém stádiu vývoje. Jedná se o neoficiální WiFi API pro kamery GoPro.[4]

2.0.2 Projekt *goprowifihack* [4]

Z projektu *goprowifihack* zjistíme, že kamery lze ovládat pomocí protokolu HTTP pouze změnami v url adrese. Schéma url adresy pro ovládní kamer je rozdílné pro každý model. Jak jsem již zmínil, tato práce vzniká s využitím modelu HERO3+ Black edition, jejíž konstrukce url adresy se shoduje se všemi modely HERO3 a je následující <http://10.5.5.9/param1/param2?t=password&p=option> [4]. Funkci jednotlivých částí adresy objasní tabulka 2.1.

K ovládní kamery je klíčové znát všechny parametry a jejich argumenty. Jak jsme zjistili v minulé části, parametr *FS* mění nastavení počtu snímků za vteřinu, pro různé nastavení počtu snímků za vteřinu, je třeba předat rozdílný



Obrázek 2.5: Stažení pořízeného obrázku přímo do Wolfram Mathematica.

Tabulka 2.1: Popis parametrů v url adrese pro ovládání kamery.

Část	Funkce
param1	Definuje, zda je akce provedena v kameře či v oddělené WiFi jednotce.
param2	Parametr obsahující dvě písmena, definuje provedenou akci.
password	Heslo pro připojení k WiFi vysílané kamerou.
option	Argument pro parametr param2.

Data v tabulce převzata z [4].

argument (argument, lze najít ve čtvrtém sloupci tabulky 2.2). Přehled hlavních parametrů a jejich argumentů obsahuje tabulka 2.2. Pokud převedeme hodnotu parametru *param2* do malých písmen a usekneme část adresy tak, aby vypadala následovně `http://10.5.5.9/param1/param2?t=password`, získáme možnost zjistit nastavení kamery. Výčet argumentů (*param2*) se bude pro různé modely kamer lišit, neboť každý model má rozdílné možnosti pro dané nastavení.

2.1 Porovnání modelů kamer

Kamery GoPro lze rozdělit do několika skupin a to více způsoby. Nejpříhodnějším z těchto způsobů je rozdělení kamer podle generací (HERO2, HERO3, HERO3+, HERO4 atd.), tedy podle času, kdy byly kamery uvedeny na trh. Kamery v jedné generaci mají společné vlastnosti a rozvržení ovládání. Liší se většinou v rozlišení pořízeného videa, kvalitě fotografií, počtu pořízených snímků za vteřinu pro jednotlivá rozlišení a tak dále. V tabulce 2.4 můžeme vidět hlavní rozdíly mezi jednotlivými modely generace kamer HERO3+.

2.1.1 Rozdíly mezi generacemi kamer

Rozdíly mezi jednotlivými generacemi kamer jsou větší než rozdíly mezi jednotlivými edicemi v rámci jedné generace. Do generace HERO3+ se jednalo hlavně o změny v kvalitě nahrávání videí a počtu snímků za vteřinu. Generace HERO4 naopak přinesla velké změny týkající se zejména nahrávacích módů kamery. Nově byly přidány módy pro použití kamery v noci, jak pro nahrávání videa, tak pro focení. Tyto změny s sebou přinesly i obměnu ručního ovládání kamery a rozvržení nabídky.

Změnou prošlo i vzdálené ovládání kamery, konkrétně složení url adresy, kterou se kamera ovládá. Nová url adresa (pro generaci HERO4) vypadá následovně: `http://10.5.5.9/gp/gpControl/setting/param/argument` [4], kde parametr *param* zastupuje číselné označení nastavení, které měníme, a parametr *argument* je novou hodnotou pro toto nastavení. Nejdůležitější parametry jsou k nahlédnutí v tabulce 2.3, kompletní přehled naleznete na stránkách projektu goprowifihack [4]. Jak je vidět, url adresa nově neobsahuje heslo k WiFi síti, ze které se kamera ovládá.

Změnila se i url adresa pro zjištění nastavení kamery na `http://10.5.5.9/gp/gpControl/status`. HTTP požadavek na tuto adresu z Wolfram Mathematica nám stáhne seznam hodnot obsahující všechna nastavení kamery. Tento seznam lze poté převést na asociativní pole, kde klíčem k získání hodnoty pro dané nastavení je číselný parametr *param*. Tentýž parametr používáme při nastavování kamery, viz předchozí odstavec. Nejdůležitější hodnoty tohoto parametru naleznete v tabulce 2.3 a přehled všech možných hodnot je k nalezení v projektu goprowifihack [4].

Tabulka 2.2: Jednotlivé parametry pro nastavení kamer generace HERO2, HERO3, HERO3+.

Funkce	param1	param2	option
zapnutí/vypnutí kamery	PW	bacpac	01 - zapnutí 00 - vypnutí
zapnutí/vypnutí spouště	SH	bacpac	01 - zapnutí 00 - vypnutí
změna módu kamery	CM	camera	00 - Video 01 - Photo 02 - Burst 03 - Timelapse
rozlišení fotografie	PR	camera	05 - 12mp wide 04 - 7mp wide 06 - 7mp medium 03 - 5mp wide
rozlišení videa	VV	camera	00 - WVGA 01 - 720p 02 - 960p 03 - 1080p 04 - 1440p 05 - 2.7K 06 - 4K 07 - 2.7K 17:9 08 - 4K 17:9 09 - 1080p Super View 0a - 720p Super View
počet snímků za vteřinu	FS	camera	00 - 12 01 - 15 02 - 24 03 - 25 04 - 30 05 - 48 06 - 50 07 - 60 08 - 100 09 - 120 0a - 240 0b - 12.5
ostrost	SP	camera	00 - vysoká 01 - střední 02 - nízká

Tabulka popisuje sestavení url adresy pro požadavek protokolu HTTP, který je potřeba odeslat na kameru, chceme-li změnit určité nastavení kamery generace HERO2, HERO3 nebo HERO3+. V prvním sloupci je popis funkce, kterou chceme provést, ve zbylých třech sloupcích jsou hodnoty potřebných parametrů a argument pro druhý z parametrů. Například pokud chceme změnit nastavení počtu snímků za vteřinu, vyhledáme v tabulce šestý řádek „počet snímků za vteřinu“. V tomto řádku jsou pak jasně dané hodnoty pro oba parametry ($param1=camera$, $param2=FS$) a je nutné zvolit argument pro druhý z parametrů (chceme-li 60 fps argument bude 07). Data v tabulce převzata z [4].

2. NÁVRH VLASTNÍHO ŘEŠENÍ

Tabulka 2.3: Jednotlivé parametry pro nastavení kamer generace HERO4.

Nastavení	param	argument parametru
Vyvážení bílé	11 - Video 22 - Photo 33 - MultiShot	0 - Automaticky 1 - 3000k 2 - 5500k 3 - 6500k 4 - Nativně 5 - 4000K 6 - 4800K 7 - 6000K
ISO Limit	13 - Video 24 - Photo 37 - MultiShot	0 - 6400 (Pouze video) 1 - 1600 (Pouze video) 2 - 400 3 - 3200 (Pouze video) 4 - 800 7 - 200 8 - 100
Ostrost	14 - Video 25 - Photo 38 - MultiShot	0 - Vysoká 1 - Střední 2 - Nízká
Rozlišení videa	2 - Video	13 - WVGA 12 - 720p 10 - 960p 9 - 1080p 7 - 1440p 4 - 2.7K 1 - 4K 6 - 2.7K 4:3 5 - 2.7K Super View 2 - 4K Super View 8 - 1080p Super View 11 - 720p Super View
Rozlišení fotografie	17 - Photo 28 - MultiShot	0 - 12Mpx wide 1 - 7Mpx wide 2 - 7Mpx medium 3 - 5Mpx wide

Tabulka popisuje sestavení url adresy pro požadavek protokolu HTTP, který je potřeba odeslat na kameru, chceme-li změnit určité nastavení kamery generace HERO4. V prvním sloupci je popis funkce, kterou chceme provést, ve zbylých dvou sloupcích jsou hodnoty potřebného parametru *param* a argumentu pro tento parametr. Například pokud chceme změnit nastavení rozlišení videa, vyhledáme v tabulce čtvrtý řádek „Rozlišení videa“. V tomto řádku je pak jasně daná hodnota pro parametr (*param=2*) a je nutné zvolit argument pro tento parametr (chceme-li rozlišení 1080p, bude hodnota argumentu *9*). Data v tabulce převzata z [4].

Tabulka 2.4: Porovnání kamer generace HERO3+.

Parametr	Black edition	Silver edition
Rozlišení videa/FPS	4K/15 4K 17:9/12 2.7K/30,25 2.7K 17:9/24 1440p/48, 30, 24 1080p/60, 48, 30, 24 1080p Super View/60, 48, 30, 24 960p/100, 60, 48 720p/120, 60 720p Super View/100, 60, 48 WVGA/240	1080p/60, 30 960p/60, 30 720p/120, 60, 30 WVGA/120, 60
Rozlišení fotografie	12Mpx 7Mpx 5Mpx	10Mpx 7Mpx 5Mpx
Sekvence fotek (počet fotek/čas)	3/1s 5/1s 10/1s 10/2s 30/1s 30/2s 30/3s	3/1s 5/1s 10/1s

Tabulka obsahuje přehled jednotlivých modelů kamer GoPro, konkrétně generace HERO3+. V levém sloupci nalezneme název parametru kamery, který porovnááme. V prostředním sloupci pak výčet možných hodnot tohoto nastavení pro model Black edition a obdobně v pravém sloupci výčet hodnot pro model Silver edition. Účelem tabulky je ukázat, jak velké rozdíly mohou být mezi dvěma modely jedné generace. Tyto modely se liší v mnoha dalších aspektech, ale jejich výčet je příliš objemný pro zpracování v jedné tabulce. Data v tabulce převzata z [5].

Řešení

K vytvoření knihovny pro vzdálené ovládání kamery GoPro v jazyce Wolfram Language použijeme Wolfram Workbench. Jedná se o plugin do vývojového prostředí Eclipse, který umožňuje vytvářet, testovat a distribuovat knihovny pro Wolfram Mathematica. Knihovny pro Wolfram Mathematica se skládají ze zdrojových souborů s koncovkou „.m“. Jelikož chceme, aby naše knihovna podporovala více kamer, které se ovládají každá trochu jinak, bude potřeba vytvořit více zdrojových souborů. Každý soubor bude obsahovat definice a implementaci funkcí k ovládání kamer, které mají shodnou strukturu ovládací url adresy. Výsledná knihovna ponese název GoProController a bude obsahovat i dokumentaci pro dokumentační centrum Wolfram Mathematica.

3.1 Rozvržení knihovny

Knihovnu rozdělíme do několika souborů: hlavní soubor GoProController.m obsahující základní funkce knihovny a vedlejší soubory zajišťující implementaci funkcí pro konkrétní modely kamer. Rozdělení knihovny do více zdrojových souborů je potřeba, neboť všechny modely kamery mají určité funkce společné, ale jejich implementace je odlišná, protože se každý model ovládá jinak.

3.1.1 Hlavní zdrojový soubor

Jak již bylo řečeno, hlavním souborem je GoProController.m, který obsahuje pouze dvě funkce (*goProSetCamera* a *goProGetPossibleCamera*, jejichž popis je v tabulce 3.1) a proměnnou *camera*. Proměnná *camera* je zástupcem generace kamery, její hodnota se používá v pozdějším užívání knihovny. Pro používání knihovny je nejdříve třeba zadat generaci kamery GoPro. To provedeme pomocí funkce *goProSetCamera*. Tato funkce pak natáhne potřebný zdrojový soubor pro zadanou kameru pomocí funkce *Get* a tím dochází k zpřístupnění funkcí implementovaných v tomto zdrojovém souboru.

3. ŘEŠENÍ

Tabulka 3.1: Funkce v hlavním souboru GoProController.m.

Název funkce	Popis
goProSetCamera	Nastaví proměnnou <i>camera</i> v závislosti na předaném parametru a natáhne zdrojové soubory potřebné k ovládání této kamery.
goProGetPossibleCamera	Vrací seznam podporovaných modelů kamery.

3.1.2 Vedlejší zdrojové soubory

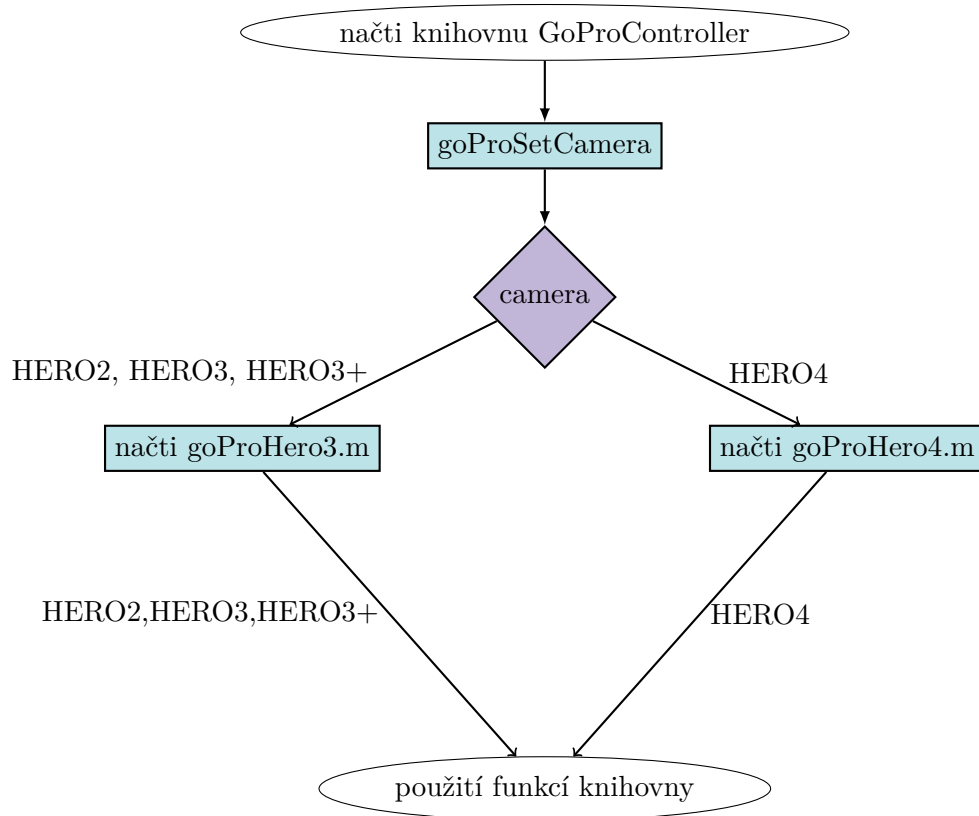
3.1.2.1 Rozdělení kamer do zdrojových souborů

Zdrojové soubory pro konkrétní modely kamer jsou goProHero3.m a goProHero4.m. Každý z těchto souborů obsahuje funkce pro ovládání určitých modelů kamery, viz tabulka 3.2 a diagram 3.1. Rozdělení modelů do souborů se řídí společným způsobem ovládání kamery, tedy podobnou strukturou url adresy, s jejíž pomocí se kamera ovládá.

Dalším faktorem rozdělení je množina společných funkcí, tedy společná množina vlastností kamery. Možnosti kamer se velice liší a to obzvlášť mezi různými generacemi kamer. Kupříkladu kamery patřící do generace HERO4 mají daleko více snímacích módů než kamery z generace HERO3 nebo HERO3+. Přehled snímacích módů u jednotlivých generací kamer naleznete v tabulce 3.3. Generace HERO4 jako první z kamer od společnosti GoPro přidala možnost natáčení a pořizování snímků v nočním režimu. Starší kamery žádnou takovou možnost nepodporovaly.

Tabulka 3.2: Vedlejší zdrojové soubory a modely kamer, které ovládají.

Název souboru	Modely kamery
goProHero3.m	GoPro HERO3 White
	GoPro HERO3 Silver
	GoPro HERO3 Black
	GoPro HERO3+ Silver
	GoPro HERO3+ Black
	GoPro HERO2
	GoPro HERO2
goProHero4.m	GoPro HERO4 Silver
	GoPro HERO4 Black
	GoPro HERO4 Session



Obrázek 3.1: Diagram načítání vedlejších zdrojových souborů knihovny *GoProController*.

Tabulka 3.3: Snímací módy jednotlivých generací kamer GoPro.

	HERO3	HERO3+	HERO4
Snímací módy	Photo Video Time Lapse Burst	Photo Video Time Lapse Burst	Video TimeLapse Video Photo in Video LoopingVideo Photo Continuous Photo Night Photo Burst Time Lapse Photo NightLapse

3.1.2.2 Možné problémy plynoucí z rozdělení kamer do zdrojových souborů

Na diagramu 3.1 můžeme vidět, že v souboru goProHero3.m je implementována funkčnost i pro generaci kamer HERO2. Tato generace nepodporuje některá nastavení parametrů, jež jsou dostupná v tomto zdrojovém souboru. Při pokusu o použití těchto funkcí kamerami z generace HERO2 nám knihovna zobrazí chybovou zprávu. Tomuto jevu by se dalo v mnoha případech předejít, kdybychom rozdělili knihovnu do více souborů. Tím by ale docházelo ke zbytečné duplikaci zdrojového kódu a s tímto nežádoucím jevem bychom se setkávali i nadále v případě generace HERO3. Do té spadají tři modely kamer (White edition, Silver edition, Black edition). Každý z těchto modelů má jiné množství možných nastavení funkcí kamery. Nejvíce jich má model Black edition, který je také nejdražší. Podobně je tomu i u generace HERO3+, která obsahuje dva modely, Black edition a Silver edition. Seznam modelů a jejich rozdělení do zdrojových souborů je vidět v tabulce 3.2.

3.2 Proměnné a konstanty

3.2.1 Konstanty

Pro běh knihovny potřebujeme hned několik proměnných a konstant. Konstantami jsou zde hlavně asociativní pole, která ve Wolfram Mathematica zapisujeme pomocí příkazu:

```
pole=<|key1->value, key->value, ..|>.
```

Tato pole používáme k převodu parametrů předaných funkcím (key) v naší knihovně na číselnou hodnotu, s jakou se požadovaný parametr nastaví na kameře (value). Pokud bychom si vzali například asociativní pole pro převod rozlišení videa na jeho číselnou hodnotu, asociativní pole se zapíše následovně:

```
videoResToCode=<|  
  "WVGA" -> "00",  
  "720p" -> "01",  
  "960p" -> "02" ,  
  "1080p" -> "03",  
  "1440p" -> "04",  
  "2.7K" -> "05",  
  "4K" -> "06",  
  "2.7K-17:9" -> "07",  
  "4K-17:9" -> "08",  
  "1080p-SV" -> "09",  
  "720p-SV" -> "0a"  
|>
```

Tedy mějme funkci nastavující rozlišení videa. Této funkci předáme jako parametr řetězec obsahující jedno z možných rozlišení, například „1080p“, pro které získáme jeho číselnou hodnotu následujícím příkazem:

```
videoResToCode [{"1080p"}]
```

Jak je vidět v ukázce, tato pole jsou pojmenovaná *settingToCode*, kde část *setting* je název nastavení, pro které je dané pole určeno (*videoResToCode*, *fpsToCode* atd.). Jak můžeme vidět, obsah těchto konstant se řídí čtvrtým sloupcem tabulky 2.2.

K těmto asociativním polím vždy existují pole opačná, ve kterých jsou prohozeny klíče a hodnoty a využíváme je při zjišťování aktuálního nastavení kamery. Další konstantou je *goProUrl*, ve které je uložena adresa `http://10.5.5.9/`, neboli adresa kamery.

3.2.2 Proměnné

Většina proměnných v knihovně zastupuje nastavení kamery a tyto proměnné jsou používány k porovnání stávajícího a požadovaného nastavení funkcemi *goProSet* (více o těchto funkcích se dozvíme v následující kapitole 3.3). Uživatel tyto proměnné nemůže nastavovat ani přímo používat jejich hodnoty.

Dalšími proměnnými jsou *goProPassword* (udržuje heslo k WiFi síti kamery a nastavuje se pomocí funkce *goProSetPassword*), *camera* (udržuje název právě používané generace kamer, od čehož se odvíjí, který zdrojový soubor knihovna právě používá a nastavuje se pomocí funkce *goProSetCamera*), *model* (zastupuje konkrétní specifikaci modelu v rámci generace kamery a nastavuje se pomocí *goProSetCameraModel*).

Další důležitou proměnnou je *urlBase*, která obsahuje url adresu složky na kameře, ze které chceme stahovat data. Pokud bychom chtěli stahovat soubory z jiné složky, je potřeba změnit hodnotu této proměnné a to funkcí *goProSetURLBase*, jejímž parametrem je právě cesta do požadované složky. Výchozí adresa je nastavena na `http://10.5.5.9:8080/DCIM/100GOPRO/`.

3.3 Funkce

Výsledné funkce v knihovně můžeme rozdělit do několika částí. Zaprvé funkce nastavující kameru pomocí předaných parametrů. Názvy těchto funkcí začínají vždy řetězcem „goProSet“. Druhou velkou skupinou jsou funkce informativní, jejichž názvy začínají na řetězec „goProGet“, popřípadě „goProGetPossible“. Funkce začínající na řetězec „goProGetPossible“ jsou určeny pro výpis možných parametrů pro funkce, které slouží k nastavení kamery a jejich hlavní náplní je zjednodušení práce s knihovnou. Funkce „goProGet“ nám pak umožňují zjistit, jak je kamera nastavena. Popis funkcí naleznete v tabulce 3.4.

Třetí skupinou jsou vnitřní funkce knihovny, které nejsou pro uživatele viditelné a zajišťují běh knihovny. Poslední skupina funkcí přímo ovládá kameru. Funkce spadající do této skupiny slouží k ovládání spouště, zapínání/vypínání kamery či stahování nebo mazání pořízeného materiálu.

Tabulka 3.4: Prefixy názvů funkcí v knihovně GoProController.

Prefix	Funkce
goProSet	Skupina funkcí nastavujících parametry kamery.
goProGetPossible	Skupina funkcí, které slouží k výčtu parametrů pro funkce <i>goProSet</i> .
goProGet	Funkce pro zjišťování nastavení kamery.

3.3.1 Hierarchie funkcí

Funkce, které uživatel používá k nastavování parametrů kamery (skupina funkcí „goProSet“) a funkce, kterými uživatel ovládá kameru (zapnutí/vypnutí, spoušť), slouží pouze k předání potřebných parametrů vnitřním funkcím knihovny. Mezi důležité vnitřní funkce patří *goProMakeCommand*, která slouží k sestavení url adresy z potřebných proměnných (*goProUrl*, *goProPassword*) a předaných parametrů pro nastavení či ovládání kamery. Parametry, které se předávají funkci *goProMakeCommand*, jsou tři (*param1*, *param2*, *option*). Jejich příklady naleznete v tabulce 2.2 ve druhém až čtvrtém sloupci.

Volání funkce pak vypadá například následovně:

```
goProMakeCommand [" camera " , " VV " , " 03 "]
```

nebo

```
goProMakeCommand [" camera " , " VV " , photoResToCode [ [" 1080p " ] ]]
```

Kde výsledná url adresa (<http://10.5.5.9/camera/VV?t=12345678&p=%03>) nastavuje rozlišení videa na 1080p. Vyhledávání parametru *option* (tedy třetího parametru funkce *goProMakeCommand*) probíhá v příslušném asociativním poli na základě parametru předaného funkci *goProSet*. Na návratovou hodnotu funkce *goProMakeCommand* (url adresa) se poté zavolá funkce *execute*, která provede http požadavek na potřebnou url adresu.

Funkce pro nastavení kamery pak vypadají například následovně:

```
goProSetVideoRes [param_String] :=  
  execute [  
    goProMakeCommand [" camera " , " VV " ,  
      videoResToCode [ [param] ]  
    ]  
  ]
```

3.3.2 Práce s nastavením kamery

Jedním z hlavních cílů této práce je schopnost knihovny ovládat kameru Go-Pro. Potřebujeme tedy nastavovat všechny dostupné parametry kamery a zjišťovat, jak je kamera nastavena. Výsledné funkce by měly být snadno použitelné a co nejvíce uživatelsky přívětivé.

3.3.2.1 Nastavování kamery

Pokud chceme kameru nastavovat, je třeba znát potřebné parametry a jejich atributy. Ty známe díky projektu `goprowifihack` [4] a ty nejdůležitější jsou popsány v tabulce 2.2. Vytvořili jsme tedy funkci pro nastavení každého parametru zvlášť (`goProSetFPS`, `goProSetVideoRes`, `goProSetPhotoRes` atd.). Jak tyto funkce fungují, je popsáno v kapitole 3.3.1, včetně ukázky funkce. Množina nastavovacích funkcí se liší v závislosti na generaci kamery. Generace HERO4 má daleko více možných nastavení než generace HERO3. To je jeden z hlavních důvodů, proč je knihovna `GoProController` rozdělena do více zdrojových souborů.

K funkcím, které slouží k nastavování parametrů kamery jsou velice užitečné informativní funkce, jejichž název začíná na řetězec „`goProGetPossible`“. Tato skupina funkcí nám vrátí seznam dostupných parametrů pro funkci, ke které se pojí. Tento seznam je buď uložen v proměnné, nebo jsme jej získali pomocí funkce `Keys`, která vrátí seznam klíčů z asociativního pole a kterou použijeme na asociativní pole `settingToCode` (účel těchto polí je popsán v kapitole 3.2).

Například funkce `goProGetPossiblePhotoRes` nám vrátí seznam parametrů pro funkci `goProSetPhotoRes`. Usnadňuje nám to nastavování kamery, jelikož je pak možné nastavovat kameru následovně:

```
goProSetPhotoRes [
  goProGetPhotoRes [] [[1]]
]
```

Tento příkaz předá jako parametr funkci `goProSetPhotoRes` první prvek seznamu vráceného funkcí `goProGetPossiblePhotoRes`.

Pokud modely kamery stejné generace mají jiné parametry pro určité nastavení (například jak je tomu v tabulce 2.4), je potřeba znát název tohoto modelu.

Známe-li tedy přesně model kamery, jsme schopni pro každou kameru vytvořit její vlastní asociativní pole s potřebnými hodnotami a klíči. Funkce nastavující kameru mají pak ve své implementaci funkci `Switch`, která právě na základě modelu kamery zvolí potřebné asociativní pole a vyhledá v něm parametr předaný funkci. Ve výsledku se tak nesnažíme nastavovat na kameře parametry, které nepodporuje a vyvarujeme se chyb. Funkce pro nastavení parametru kamery vypadá například následovně:

```
goProSetVideoRes [ param_String ] := Switch [
  camera ,
  "HERO3+" ,
  execute [ goProMakeCommand [ " camera " , "VV" ,
    videoResToCode [ [ param ] ] ]
  ] ,
  "HERO2" ,
  execute [
    goProMakeCommand [ " camera " , "VR" , videoResToCodeHERO2 [ [ param ] ] ]
  ] ,
  "HERO3" , Switch [ model ,
    " Black " , execute [
```

3. ŘEŠENÍ

```
    goProMakeCommand["camera", "VV", videoResToCodeHERO3[[param]]]
  ],
  "Silver", execute [
    goProMakeCommand["camera", "VR", videoResToCodeHERO2[[param]]]
  ],
  "White", execute [
    goProMakeCommand["camera", "VR", videoResToCodeHERO2[[param]]]
  ],
  Message[goProSetCameraModel::model, model]
]
]
```

Jak je v ukázce vidět, v závislosti na proměnných *camera* a *model* se funkce rozhoduje, jaké parametry předá funkci *goProMakeCommand*. Popřípadě vypíše chybovou hlášku, pokud nebyla nastavena proměnná *model*.

3.3.2.2 Nastavení více parametrů najednou

Chceme-li nastavovat více parametrů kamery najednou, je nepraktické používat oddělené funkce pro každý parametr. Wolfram Mathematica nám poskytuje vhodné řešení tohoto problému, v podobě funkce *Options*. Ta nám umožňuje nastavit proměnné množství parametrů požadované funkci. V našem případě se jedná o funkci *goProSet*, která bude mít za úkol nastavovat všechny zvolené parametry najednou. [6]

Funkce *Options* pracuje na principu „pravidel“, kde pro každé pravidlo definujeme počáteční hodnotu a pokud pak při volání funkce není tato hodnota pravidlem změněna, zůstane na počáteční hodnotě. Kvůli tomu ovšem nejsme schopni zjistit, která pravidla byla použita. Při použití této funkce by ovšem bylo třeba nastavovat všechny parametry kamery znovu při každém jejím zavolání, což je poměrně časově náročné (při použití kamery HERO3+ Black edition trvá nastavení všech parametrů přes WiFi síť přibližně 15 vteřin).

Toho se vyvarujeme tak, že počáteční hodnoty pravidel nastavíme každé na jednu proměnnou (pokud se změní hodnota této proměnné, změní se i počáteční hodnota pravidla). Dosáhli jsme tedy proměnné počáteční hodnoty. Nastavení funkce *Options* pro funkci *goProSet* vypadá následovně:

```
Options[goProSet] = {
  videoResolution -> videoResolutionDef,
  fps -> fpsDef,
  fov -> fovDef,
  photoResolution -> photoResolutionDef
}
```

Kde *videoResolution*, *fps*, *fov*, *photoResolution* jsou názvy pravidel pro funkci *goProSet* a *videoResolutionDef*, *fpsDef*, *fovDef*, *photoResolutionDef* jsou proměnné, na které jsou nastaveny počáteční hodnoty pravidel. Výše uvedený příklad je pouze ukázkou, jak pracujeme s nastavením *Options* pro funkci *goProSet*. Samotná implementace se liší v závislosti na generaci či modelu kamery (různé typy kamer mají rozdílné množství nastavovatelných parametrů).

S použitím funkce *Options* se mění definice funkce *goProSet*. Nedefinujeme u ní obvyklé parametry a její definice vypadá následovně:


```

goProSet [OptionsPattern []] := (
  downloadSettings [];
  goProSetVideoRes [
    OptionValue [videoResolution]
  ];
  goProSetFPS [
    OptionValue [fps]
  ];
)

```

Ve skutečné definici se znovu objevuje větvení funkce pomocí *Switch*, neboť takto zamezíme nastavování parametrů, které kamera nepodporuje. V ukázce vidíme, že nejdříve se vykonává příkaz *downloadSettings[]*, který stáhne všechna dostupná nastavení pro daný model a uloží je do proměnných, na které jsou nastaveny počáteční hodnoty pravidel funkce *goProSet* (*videoResolutionDef*, *fpsDef*, *fovDef*).

Dále funkce *goProSet* nastavuje parametry kamery pomocí funkcí k tomu určených (*goProSetVideoRes* pro nastavení rozlišení videa a podobně). Parametry kamery nastavujeme na hodnoty předané pravidly, které získáme funkcí *OptionValue*, jak je vidět na ukázce. V reálné definici funkce *goProSet* (ještě před nastavováním parametrů kamery) zkontroluje, která pravidla změnila hodnotu oproti hodnotě počáteční (tedy té, na kterou je nastavena kamera) a nastavujeme pak jen ty parametry, které jsou rozdílné.

Volání funkce *goProSet* pak vypadá následovně:

```

goProSet [
  videoResolution -> "1080p",
  fps -> "60",
]

```

Tento příkaz by nastavil rozlišení videa na „1080p“ a počet snímků za vteřinu na 60. Výsledkem je tedy funkce, díky které jsme schopni nastavovat velké množství parametrů kamery jediným příkazem (pro všechny parametry by to byl opravdu dlouhý příkaz).

3.3.2.3 Zjištění nastavení kamery

Pro vzdálené ovládání kamery GoPro je velmi důležité vědět, jak je kamera momentálně nastavena. Z tohoto důvodu obsahuje naše knihovna funkci *goProGet*. Tato funkce přijímá jako parametr řetězec, proměnnou nebo seznam (proměnných či řetězců). Jako parametry do funkce *goProGet* můžeme znovu použít proměnné z funkce *goProGetVariables*, čímž bychom získali všechna nastavení kamery.

Funkce *goProGet* nejdříve stáhne všechna nastavení kamery a uloží je do stejných proměnných, které používá funkce *goProSet* pro nastavení počátečních hodnot pravidel (tedy proměnné *videoResolutionDef*, *fpsDef*, *fovDef* atd.). K těm budeme přistupovat zkrze asociativní pole, které má jako klíče řetězce obsahující název pravidla z funkce *goProSet* (tedy řetězce *"videoResolution"*, *"fps"*, *"fov"* atd.). Toto pole pak může vypadat následovně (reálně obsahuje daleko více prvků):

3. ŘEŠENÍ

```
pole=<|
  "videoResolution" -> videoResolutionDef,
  "fps" -> fpsDef,
  "fov" -> fovDef
|>
```

Do tohoto pole lze tedy přistupovat na základě klíče, kterým je řetězec. Pokud je funkci *goProGet* předán jako parametr řetězec či seznam řetězců, stačí nám pouze použít daný parametr jako klíč pro získání hodnoty.

Na problém narazíme, pokud se snažíme funkci *goProGet* předat jako parametr proměnnou nebo seznam proměnných. Při normálním chování by se tato proměnná vyhodnotila a funkci by byla předána pouze její hodnota. Naštěstí pro nás ve Wolfram Mathematica existuje řešení pro tento problém.

Funkci *goProGet* nastavíme atribut „HoldAll“ příkazem:

```
SetAttributes[goProGet, HoldAll]
```

Tento atribut má za následek to, že proměnné předané funkci *goProGet* se nevyhodnocují a lze dále používat jejich názvy. Ty v těle funkce *goProGet* převedeme na řetězec a získáme tedy klíč do asociativního pole obsahujícího nastavení kamery.

Funkci *goProGet* lze pak volat více způsoby:

```
goProGet [{"videoResolution", "fps"}];
goProGet [{videoResolution, fps}];
```

Oba tyto příklady by nám vrátily následující seznam:

```
{"videoResolution" -> "1080p", "fps" -> "60"}
```

3.3.2.4 Uložení nastavení kamery pro pozdější použití

Pro příjemnější používání GoPro kamery se nám hodí možnost ukládat aktuální nastavení kamery do souboru. Toho využijeme zejména, pokud chceme používat GoPro kameru za různých podmínek (světelné podmínky, požadovaná kvalita záznamu či fotografie atd.). Tuto funkčnost nám zajistí funkce *goProSaveSettings*, ta stáhne aktuální nastavení kamery pomocí funkce *goProGet*. Parametrem pro funkci *goProGet* bude seznam vrácený funkcí *goProGetVariables*. Tím získáme seznam všech nastavení kamery pro konkrétní model. Funkce *goProSaveSettings* následně uloží vrácený seznam do souboru (jehož název je předán jako parametr) s příponou „.xls“.

Soubor s příponou „.xls“ jsme využili, protože se skvěle hodí pro ukládání vícerozměrných seznamů. Ukázkou struktury souboru vidíme na obrázku 3.2. Jak je z ukázky patrné, v prvním řádku jsou názvy nastavení kamery (lze je získat funkcí *goProGetVariables*) a v druhém řádku jsou uloženy hodnoty pro tato nastavení.

Pro obnovu nastavení kamery z uloženého souboru je v naší knihovně funkce *goProLoadSettings*. Ta načte data ze souboru, který jí byl předán jako

3.4. Práce s pořízenými daty

	A	B	C	D	E	F	G	H	I	J	K	L
1	videoResolution	fps	fov	videoMode	photoResolution	loopVideo	burstRate	timeLapse	continuousShot	photoInVideo	volume	led
2	720p	60	wide	ntsc	12W	off	3/1	1	1	off	off	4
3												
4												
5												
6												
7												

Obrázek 3.2: Struktura souboru s uloženým nastavením kamery.

parametr a převede je do formátu podporovaného funkcí *goProSet*. Tím se nastaví všechny parametry kamery, které jsou vyjmenovány v daném souboru.

3.4 Práce s pořízenými daty

Jedním z hlavních cílů této práce je zpřístupnění dat pořízených kamerou pro jejich následné zpracování.

3.4.1 Výpis a stažení dat na kameře

Pro práci s pořízenými daty z GoPro kamery je důležité stažení těchto dat do počítače. Tuto část knihovny rozdělíme do dvou bodů. Zaprvé výpis všech dat pořízených kamerou a zadruhé jejich samotné stažení.

3.4.1.1 Výpis dat

Kamerou pořízené záznamy se ukládají do složky `http://10.5.5.9:8080/DCIM/100GOPRO/` (HERO2, HERO3, HERO3+) či `http://10.5.5.9:8080/videos/DCIM/100GOPRO/` (HERO4), knihovna *GoProController* vypisuje či stahuje pouze data z těchto složek (ve výchozím nastavení knihovny). Url adresa do složky, ze které se aktuálně vypisují či stahují data je uložena v proměnné *urlBase* (viz sekci 3.2), ke které může uživatel přistupovat pomocí funkce *goProGetURLBase*. Pokud bychom rádi stahovali data i z jiné složky, lze tuto proměnnou změnit funkcí *goProSetURLBase*. Kamera GoPro občas ukládá záznamy i do jiných složek a je tedy na uživateli, ze které složky chce data stahovat či vypisovat.

Výpis souborů uložených ve složce, jejíž adresa je uložena v proměnné *urlBase*, zajišťuje funkce *goProGetFileList*. Tato funkce nám vrací seznam jmen souborů uložených v této složce. Funkce funguje následovně: nejdříve naimportuje textový obsah na adrese *urlBase* a poté z tohoto textu vytahuje pouze názvy souborů, které přidá do výsledného seznamu.

3.4.1.2 Stažení dat

K samotnému stažení souborů nám poslouží funkce *goProDownloadFile* a *goProDownloadAllFiles*. Funkce *goProDownloadFile* slouží ke stahování jednoho

či více souborů z kamery. Jejím prvním parametrem je buďto řetězec obsahující název souboru, nebo seznam obsahující řetězce s názvy souborů.

Tato funkce ve své implementaci nejdříve zkontroluje, zda je daný soubor na kameře k nalezení. Pokud není, vrátí chybovou hlášku, v opačném případě stáhne požadovaný soubor do složky, jejíž název jsme předali jako druhý parametr. Funkce *goProDownloadAllFiles* provádí v zásadě to samé, akorát nejdříve využije funkce *goProGetFileList* k získání všech názvů souborů uložených na kameře.

Pokud bychom chtěli importovat obrázky rovnou do Wolfram Mathematica, poslouží nám funkce *goProGetFileURL*, která nám vrátí url adresu souboru, jehož název dostane jako parametr a poté stačí použít funkci *Import*.

3.4.2 Mazání dat

Pro mazání záznamů na kameře (paměťové kartě) nám poslouží funkce *goProDeleteAll*, která maže všechny záznamy pořízené kamerou a funkce *goProDeleteLast*, která maže poslední záznam na kameře. Konstrukce těchto funkcí je stejná jako u funkcí pro nastavení kamery, liší se pouze v hodnotě parametru. Bohužel se mi nepodařilo naimplementovat mazání pouze vybraných položek z kamery. Touto funkcí nedisponuje ani kamera, která je také schopna mazat buď všechny soubory, nebo jen poslední pořízený záznam.

3.5 Dokumentace

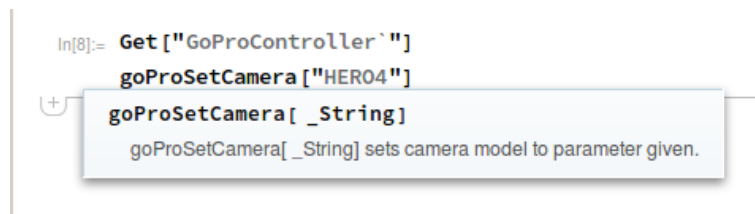
Dokumentaci knihovny *GoProController* provádíme v angličtině a to dvěma způsoby. Zprvce průběžně komentujeme vytvářené části zdrojového kódu přímo ve zdrojových souborech a za druhé vytváříme vlastní dokumentaci pro dokumentační centrum Wolfram Mathematica

Pro dokumentaci zdrojových souborů používáme krátké komentáře k popisu funkcí, proměnných či konstant. Zdrojové soubory knihovny jsou rozděleny na dvě části. V první části je deklarace veřejných funkcí a jejich krátký popis, který vypadá například takto:

```
goProSetCamera::usage=  
"goProSetCamera[_String] sets  
camera model to parameter given."
```

Tyto funkce jsou poté viditelné a použitelné ve Wolfram Mathematica, jejich krátký popis je zobrazen během jejich volání viz obrázek 3.3, což nám umožňuje snadnější používání funkcí. Druhá část zdrojového souboru je privátní a provádí se v ní implementace funkcí veřejných i neveřejných. V této části komentujeme zdrojový kód krátkými komentáři (komentáře se uzavírají do bloku *(*komentář*)*).

Jak jsem již zmínil, druhým způsobem dokumentace je vytvoření vlastní dokumentace pro dokumentační centrum Wolfram Mathematica. Tato dokumentace pak slouží k popisu funkcí a ukázkám jejich využití, ale neposkytuje



Obrázek 3.3: Našeptávání krátkého popisu funkce ve Wolfram Mathematica.

žádné informace o vnitřní implementaci knihovny. Vytváření této dokumentace pro knihovnu `GoProController` provádíme také v prostředí Wolfram Workbench. Toto prostředí nám umožňuje integrovat dokumentaci knihovny přímo do dokumentačního centra Wolfram Mathematica (to se vyvolává stiskem klávesy „F1“ z jakéhokoliv notebooku).

Dokumentační centrum se dělí na tři úrovně, průvodce (Guide), symboly (Symbol) a tutoriál (Tutorial). Každá stránka v dokumentačním centru je samostatným notebookem Wolfram Mathematica. Stránky mají specifické rozvržení podle toho, do jaké úrovně spadají, toto jednotné rozvržení v rámci úrovně nám usnadňuje orientaci v dokumentaci. Ukázky funkcí a jejich využití jsou v dokumentačním centru spustitelné a lze s nimi experimentovat a obměňovat je dle libosti, po uzavření dokumentačního centra se ukázky vrátí do původního stavu.

3.5.1 Vlastní dokumentace

Vytvořili jsme tedy pro dokumentační centrum jednoho průvodce *GoProController package*. Ten obsahuje stručný popis knihovny a odkazy na symboly (funkce) knihovny. Odkazy jsou rozděleny do více částí podle účelu, ke kterému se dané funkce využívají. Máme tedy část, do které spadají funkce pro nastavování knihovny a část, ve které jsou funkce nastavující nebo ovládající kameru. Dalším obsahem průvodce jsou odkazy na návody (tutoriály). Obvykle se do dokumentačního centra vytvářejí stránky pro všechny symboly v knihovně, ovšem to je v našem případě zbytečné. Neboť symbolů v naší knihovně je příliš velké množství a jejich stránka by obsahovala pouze krátký popis toho, co je obsaženo v názvu daného symbolu.

3.5.1.1 Návody

V našem průvodci se nachází odkazy na návod k využití knihovny pro každou generaci či skupinu generací kamery GoPro, kterou knihovna podporuje. I když se využití knihovny pro různé generace shoduje v mnoha ohledech, vytvořili jsme tutoriál pro každou generaci (či skupinu generací) v závislosti na typu ovládání kamery a také v závislosti na potřebném nastavení knihovny pro danou generaci či jejich skupinu. Například generace kamer HERO2, HERO3

3. ŘEŠENÍ

a HERO3+ potřebují ke svému ovládní heslo k WiFi síti, kterou kamery vysílají, a je tedy potřeba nastavit knihovnu podrobněji. Více se dozvíte v kapitole 4.1.

Každý z návodů v této skupině obsahuje také úplný seznam veřejných funkcí pro danou generaci kamer. Pro některé generace a modely kamer je možné, že knihovna umožní volání funkcí, které model kamery nepodporuje. V takovém případě knihovna vypíše chybovou hlášku bez dopadu na běh programu. Tento jev nastává hlavně u kamer generace HERO2 a HERO3, neboť sdílejí zdrojový soubor s generací HERO3+, která má více parametrů k nastavení. Tomu by se dalo předejít rozdělením knihovny do více zdrojových souborů, pak by ale docházelo ke zbytečné duplikaci zdrojového kódu.

Další návody v dokumentaci se zabývají ovládním kamery, jejím nastavováním či stahováním pořízených materiálů. Odkazy na tyto návody rovněž nalezneme v průvodci *GoProController package*. Návod pro ovládní kamery je rozdělen do dvou částí. V první části se dozvíme, jaké funkce je potřeba zavolat před samotným ovládním kamery a proč tomu tak je. Druhá část se věnuje samotnému ovládní kamery (zapnutí, vypnutí, změna nahrávacích režimů a spoušť).

Návod zabývající se nastavováním kamery a zjišťováním jejího stávajícího nastavení popisuje podrobněji funkce *goProGet* a *goProSet*. Je zde také vysvětleno, že knihovna *GoProController* obsahuje nastavovací funkci pro každý parametr kamery a k ní funkci přidruženou, která vrací seznam možných parametrů pro nastavující funkci (název funkce pro zjištění možných parametrů začíná vždy na řetězec „*goProGetPossible*“ viz tabulka 3.4). Součástí tohoto návodu je pak i ukázka využití funkcí *goProSaveSettings* a *goProLoadSettings*, které slouží k ukládání aktuálního nastavení kamery do souboru a k jeho následnému využití pro obnovu tohoto nastavení. Poslední návod obsahuje podrobnější informace a příklady pro stahování a mazání dat pořízených kamerou.

3.5.1.2 Symboly

Poslední skupinou stránek v naší dokumentaci jsou stránky pro funkce (symboly). Obvykle se vytváří stránka v dokumentačním centru pro každý symbol knihovny. To by ovšem v našem případě bylo značně časově náročné a bez účinku, neboť většina funkcí v naší knihovně je pojmenována tak, aby její název vypovídal o jejím účelu. Navíc všechny veřejné funkce mají u své deklarace uveden krátký popis, který vystihuje jejich použití. Z tohoto důvodu jsme vytvořili dokumentaci pouze pro funkce, které budou nejčastěji používány. Na jejich stránce v dokumentačním centru nalezneme popis funkce (včetně všech typů parametrů, které přijímá) a příklad využití dané funkce.

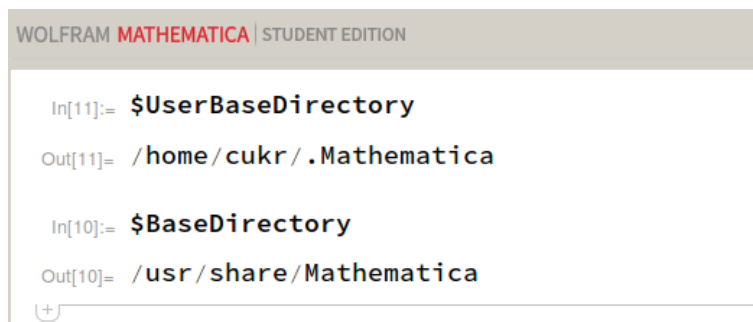
Výsledky

Výstupem práce je knihovna funkcí *GoProController* pro prostředí Wolfram Mathematica. Tato knihovna je určena pro ovládání a nastavování kamer GoPro, konkrétně pak pro modely HERO2, HERO3 White edition, HERO3 Silver edition, HERO3 Black edition, HERO3+ Silver edition, HERO3+ Black edition, HERO4 Silver edition, HERO4 Black edition a HERO4 Session. Tato knihovna dále poskytuje dokumentaci integrovanou přímo do dokumentačního centra Wolfram Mathematica, kde jsou k dispozici stránky s popisy nejdůležitějších funkcí a symbolů. Dokumentace dále obsahuje návody na nastavení knihovny, ovládání různých modelů kamer a nastavování kamery či zjišťování aktuálního nastavení kamery.

4.1 Návod k použití knihovny

4.1.1 Instalace

K samotnému používání knihovny je zapotřebí výpočetní systém Wolfram Mathematica verze 10.0 či novější (ve starších verzích nejsou podporována asociativní pole „Association“, kterých v této knihovně hojně využíváme), kamera GoPro a připojení k WiFi síti, kterou kamera vysílá. Obsah příloženého média, konkrétně složku „GoProController“, je potřeba přkopírovat do počítače, nejlépe do složky určené k instalaci balíčků pro Wolfram Mathematica. Cesta k těmto složkám je uložena v proměnných $\$UserBaseDirectory$ a $\$BaseDirectory$, tyto proměnné stačí vypsát ve Wolfram Mathematica viz obrázek 4.1. Ať už vybereme kteroukoliv z těchto složek, je třeba náš balíček přkopírovat do podsložky *Applications*, tu je potřeba vytvořit, pokud neexistuje.



```
WOLFRAM MATHEMATICA | STUDENT EDITION

In[11]:= $UserBaseDirectory
Out[11]= /home/cukr/.Mathematica

In[10]:= $BaseDirectory
Out[10]= /usr/share/Mathematica
```

Obrázek 4.1: Vypsání uživatelských proměnných obsahujících cestu pro instalaci balíčku.



```
WOLFRAM MATHEMATICA | STUDENT EDITION

In[12]:= Needs["GoProController`"]

In[13]:= Get["GoProController`"]
```

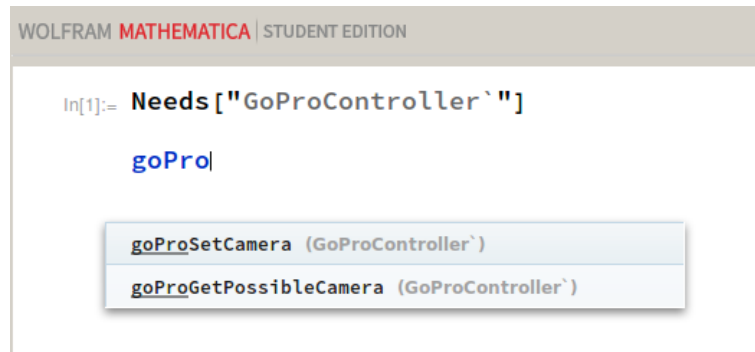
Obrázek 4.2: Ukázky možností načtení balíčku *GoProController* do Wolfram Mathematica.

4.1.2 Načtení a nastavení knihovny *GoProController*

Pro používání naší knihovny pak stačí načíst balíček *GoProController* pomocí funkce *Needs* nebo *Get*, jak je vidět na obrázku 4.2. Ihned po načtení balíčku *GoProController* můžeme začít používat funkce v něm definované.

Všechny veřejné funkce v naší knihovně začínají řetězcem „goPro“. Jelikož Wolfram Mathematica poskytuje našeptávání k symbolům, které byly již použity nebo které jsou definované, stačí nám napsat řetězec „goPro“ a našeptávání nám zobrazí dostupné funkce, jak můžeme vidět na obrázku 4.3. Knihovna *GoProController* potřebuje ke svému běhu informace o kameře, pro kterou chceme knihovnu používat.

Ihned po načtení balíčku jsou dostupné jen dvě funkce *goProGetPossibleCamera* pro vypsání podporovaných generací kamer GoPro a *goProSetCamera* pro nastavení požadované generace. Na výběr máme generaci HERO2, HERO3, HERO3+, HERO4. Pokud zvolíme jakoukoliv generaci kromě HERO2, je ještě potřeba nastavit konkrétní model kamery pomocí funkce *goProSetCameraModel*. Dostupné modely kamery pro každou generaci dostaneme zavoláním funkce *goProGetPossibleCameraModel*. Poté je potřeba (v případě kamer generace HERO2, HERO3 a HERO3+) předat knihovně heslo pro WiFi síť, kterou kamera vysílá. Obrázek 4.4 a diagram 4.5 ukazují, které funkce je potřeba zavolat pro konkrétní generace kamer, než je knihovna plně nastavena a připravena k použití.



```
WOLFRAM MATHEMATICA | STUDENT EDITION

In[1]:= Needs["GoProController`"]

goPro|

goProSetCamera (GoProController`)
goProGetPossibleCamera (GoProController`)
```

Obrázek 4.3: Funkce dostupné ihned po načtení balíčku *GoProController*.

```
WOLFRAM MATHEMATICA | STUDENT EDITION

In[1]:= Needs["GoProController`"]
        goProGetPossibleCamera[]

Out[2]= {HERO3+, HERO4, HERO2, HERO3}

Generace HERO2

In[3]:= goProSetCamera["HERO2"]
        goProSetPassword["12345678"];

Generace HERO3

In[5]:= goProSetCamera["HERO3"]
        goProSetCameraModel["Black"];
        goProSetPassword["12345678"];

Generace HERO3+

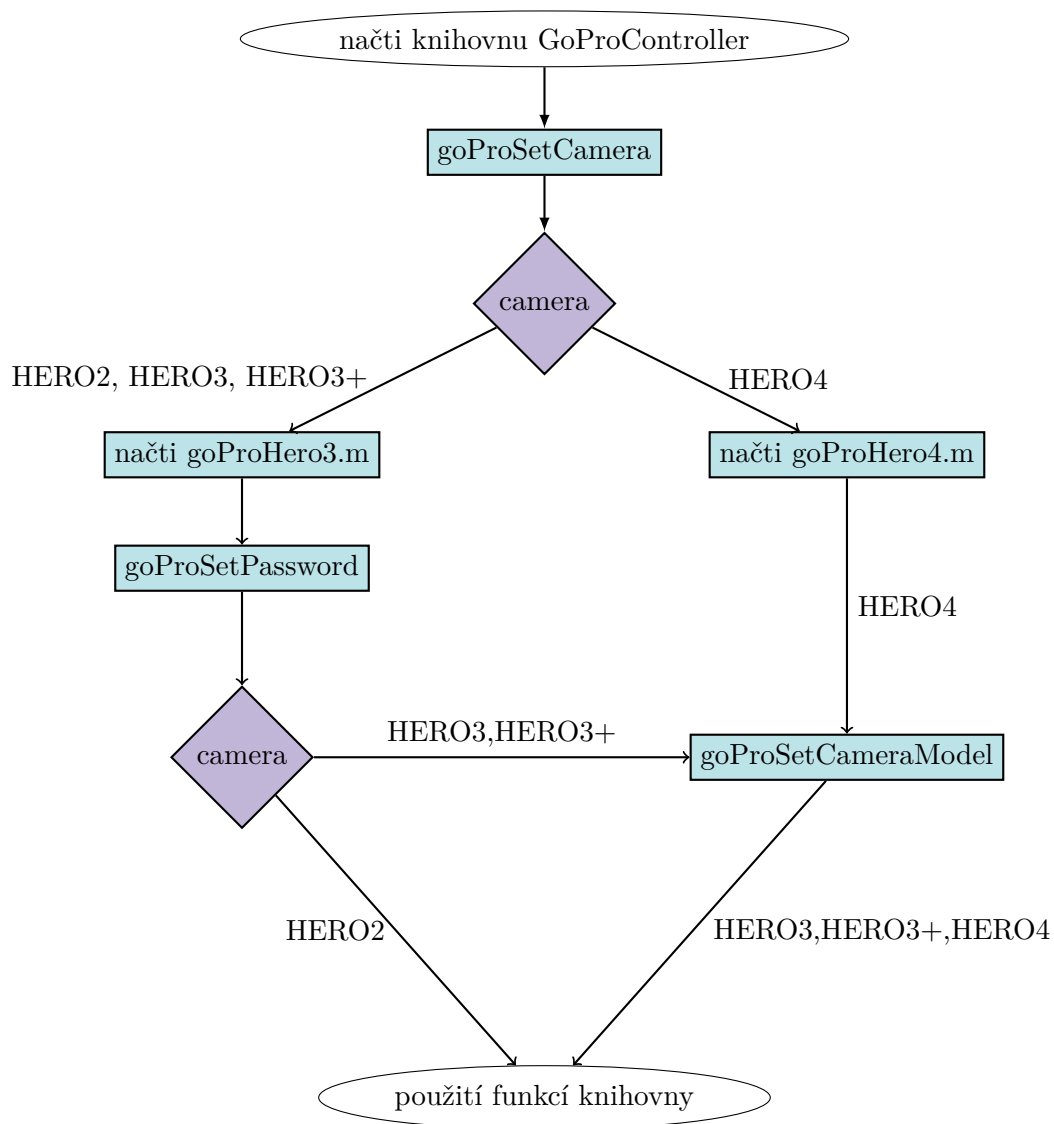
In[8]:= goProSetCamera["HERO3"]
        goProSetCameraModel["Black"];
        goProSetPassword["12345678"]

Generace HERO4

In[11]:= goProSetCamera["HERO4"]
         goProSetCameraModel["Black"];

(+)
```

Obrázek 4.4: Nastavení knihovny *GoProController* před jejím použitím.



Obrázek 4.5: Diagram nastavení knihovny *GoProController* před jejím použitím.

Tabulka 4.1: Funkce sloužící k ovládání kamery GoPro.

Funkce	Popis
<code>goProTurnOn</code>	Zapíná kameru.
<code>goProTurnOff</code>	Vypíná kameru.
<code>goProShutter</code>	Spouští kameru.
<code>goProStop</code>	Zastavuje nahrávání kamery.
<code>goProMode</code>	Nastavuje video mód kamery v závislosti na parametru.
<code>goProGetPossibleModes</code>	Vrací seznam dostupných módů.
<code>goProVideoMode</code>	přepne kameru do video módu.
<code>goProPhotoMode</code>	přepne kameru do foto módu.
<code>goProBurstMode</code>	přepne kameru do módu sekvence.
<code>goProTimeLapseMode</code>	přepne kameru do módu časosběrné fotografie.

4.1.3 Ovládání kamery

Před ovládáním kamery je nutné kameru zapnout funkcí `goProTurnOn`. Tato funkce se vyskytuje jen ve zdrojovém souboru `goProHero3.m`, lze ji tedy použít pouze pro kamery generace HERO2, HERO3 a HERO3+. Kameru generace HERO4 naše knihovna zapnout neumí, neboť vyžaduje ke svému zapnutí WoL (Wake on LAN), čehož Wolfram Mathematica není schopna bez použití jiných programů.

Dalšími funkcemi pro ovládání kamery jsou funkce pro změnu nahrávacích módů. Mezi ty patří funkce `goProMode`, které jako parametr předáváme název módu, který chceme používat a k ní patří nápovědná funkce `goProGetPossibleModes`, která nám vrátí možné parametry pro funkci `goProMode`. Nahrávání či focení spustíme funkcí `goProShutter` a zastavíme jej užitím funkcí `goProStop`. Přehled všech funkcí pro ovládání kamery GoPro naleznete v tabulce 4.1.

4.1.4 Nastavování kamery

Pro nastavování parametrů kamery jsme vytvořili funkce začínající na řetězec „`goProSet`“, které nastaví daný parametr kamery na hodnotu předanou funkci. K těmto funkcím slouží jako nápověda funkce začínající na řetězec „`goProGetPossible`“, které vracejí seznam možných parametrů pro dané nastavení. Seznam těchto funkcí zde nebudeme vypisovat, neboť by byl příliš dlouhý (pro každý nastavitelný parametr kamery jedna funkce).

Všechny funkce, které nastavují kameru, jsme spojily do jedné funkce s názvem `goProSet`. Tato funkce nám umožňuje nastavovat více parametrů kamery

4. VÝSLEDKY

```
WOLFRAM MATHEMATICA | STUDENT EDITION

Setting each parameter separately.

In[20]:= goProGetPossiblePhotoRes []
         goProGetPossibleVideoRes []
Out[20]= { 5M, 7W, 12W, 7M }

Out[21]= { WVGA, 720p, 960p, 1080p, 1440p, 2.7K,
         4K, 2.7K-17:9, 4K-17:9, 1080p-SV, 720p-SV }

In[25]:= goProSetPhotoRes ["7W"];
         goProSetVideoRes ["720p"];
         goProGet[{photoResolution, videoResolution}]
Out[27]= { photoResolution -> 7W, videoResolution -> 720p }

Setting more parameters at once.

In[28]:= goProSet[photoResolution -> "12W", videoResolution -> "1080p"];
         goProGet[{photoResolution, videoResolution}]
Out[29]= { photoResolution -> 12W, videoResolution -> 1080p }
```

Obrázek 4.6: Nastavování parametrů kamery pomocí funkce *goProSet*.

najednou. Více je o této funkci napsáno v kapitole 3.3.2.1 a ukázkou jejího použití naleznete na obrázku 4.6.

K funkci *goProSet* se pojí funkce *goProGet*, která vrací seznam stávajícího nastavení kamery v závislosti na předaném parametru. Ukázkou jejího použití naleznete také na obrázku 4.6.

4.1.5 Práce s daty na kameře

Knihovna disponuje také funkcemi pro stahování pořízeného materiálu. Funkce *goProGetFileList* nám vrací seznam jmen položek, které se nacházejí na kameře. Tento seznam můžeme využít pro jejich stažení a to pomocí funkce *goProDownloadFile*. Této funkci předáváme dva parametry. Prvním je název souboru nebo seznam názvů souborů, které chceme stáhnout a druhým je cesta do složky, kam chceme soubory stáhnout. Další možností je využít funkci *goProDownloadAllFiles*, která stáhne všechny soubory do složky, jejíž cesta je předána jako parametr.

Mazat soubory na kameře je možné dvěma způsoby. Buď smažeme poslední pořízený soubor (v případě sekvence smažeme celou sekvenci) funkcí *goProDeleteLast* nebo smažeme všechny soubory funkcí *goProDeleteAll*. Přehled všech funkcí pro práci se soubory na kameře naleznete v tabulce 4.2.

4.2 Testování

Knihovna funkcí *GoProController* byla vyvíjena a testována na operačním systému Linux Mint 18.1 Serena. A samotné testování probíhalo ve Wolfram Mathematica verze 11.0.1 a to dvěma způsoby. Zaprvé jsme testovali samotné

Tabulka 4.2: Funkce pro práci se soubory na kameře GoPro.

Funkce	Popis
<code>goProSetURLBase</code>	Nastavuje cestu do složky na kameře, ve které se hledají soubory.
<code>goProGetURLBase</code>	Vypíše cestu do složky na kameře, ve které se hledají soubory.
<code>goProDownloadFile</code>	Stáhne soubor, jehož název je předaný jako první parametr, do složky předané v druhém parametru.
<code>goProDownloadAllFiles</code>	Stáhne všechny soubory z kamery do složky předané funkci parametrem.
<code>goProGetFileList</code>	Vrátí seznam souborů přítomných na kameře.
<code>goProGetFileURL</code>	Vrátí URL adresu k souboru na kameře, jehož název byl předán jako parametr.
<code>goProDeleteLast</code>	Smaže poslední záznam z kamery.
<code>goProDeleteAll</code>	Smaže všechny soubory z kamery.

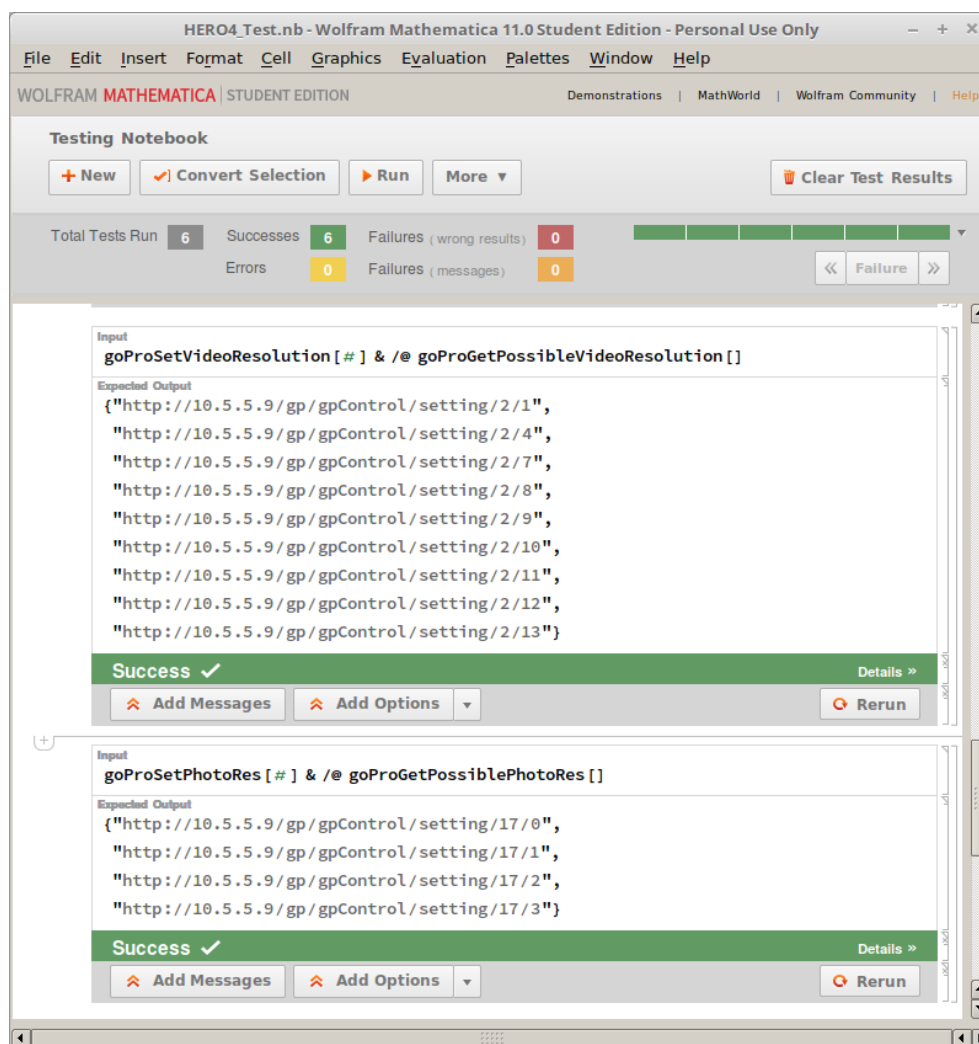
nastavování kamer pomocí knihovny *GoProController* a zadruhé jsme testovali správnost url adres složených funkcemi knihovny.

4.2.1 Testování složení url adres

Složení url adres testujeme abychom předešli možným problémům při nastavování parametrů kamery. Ke skládání url adres dochází následovně. Při zavolání funkce *goProSet* či jakékoliv jiné funkce, která má za úkol změnit nastavení kamery GoPro, se parametry předané této funkci zpracují a převedou na jiné hodnoty pomocí asociativních polí. Tyto hodnoty následně předáme funkci *goProMakeCommand* spolu s dalšími parametry. A právě v této funkci dochází ke složení výsledné url adresy z předaných parametrů a vnitřních konstant knihovny (více o funkci *goProMakeCommand* naleznete v kapitole 3.3.1).

Pro účely testování jsme provedli malou změnu ve zdrojových souborech knihovny. Změna spočívá v tom, že funkce, které mají normálně za úkol posílat HTTP požadavky na určitou url adresu tuto adresu pouze vypíšou a požadavek se neodešle. Díky tomu není potřeba k testování složení url adres

4. VÝSLEDKY



Obrázek 4.7: Ukázka testování složení url adresy.

kamera GoPro. K samotnému testování jsme vytvořili notebooky s názvy *HERO3_Test.nb* a *HERO4_Test.nb*, které jsou k dispozici na přiloženém médiu.

Tyto notebooky mají speciální rozvržení a také obsahují funkce, které nám umožňují testování. Těmito funkcím předáme příkaz, který chceme vykonat a očekávaný výsledek (v našem případě řetězec obsahující konkrétní url adresu). Po spuštění testování se příkazy vyhodnotí a následně porovnájí s očekávaným výstupem. Po dokočení testování se nám zobrazí výsledky proběhlých testů. Jak testování vypadá můžete vidět na obrázku 4.7. Požadované url adresy pro očekávané výsledky získáme z projektu *goprowifihack* [4].

4.2.2 Testování nastavení kamery

Nastavování kamery testujeme na kamerách GoPro HERO3+ Black edition a HERO4 Silver edition. Pro každou z kamer jsme vytvořili jeden notebook ve Wolfram Mathematica, ve kterém provádíme testování.

Testování probíhá následovně. Na začátku importujeme knihovnu *GoProController* a nastavíme ji pro příslušný model kamery. Následně pomocí funkce *goProSet* nastavujeme parametry kamery. Zda se nastavení provedlo kontrolujeme jednak vizuálně na displeji kamery, jednak pomocí mobilní aplikace Capture, ta obsahuje přehlednější výčet aktuálního nastavení kamery než samotný displej kamery, viz obrázek 2.1. Správnost právě provedeného nastavení kamery kontrolujeme také funkcí *goProGet*, s jejíž pomocí si necháme vypsat požadované nastavení. Tím kontrolujeme správnou funkčnost nastavování kamery i zjišťování aktuálního nastavení.

Testovat správnou funkčnost každého nastavení kamery pro všechny parametry daného nastavení by bylo časově velmi náročné. Z tohoto důvodu neprovádíme kontrolu pro všechny kombinace nastavení a jejich parametrů. Testovací notebooky jsou uloženy na příloženém médiu a jejich názvy jsou *HERO3_Test_withCamera.nb* a *HERO4_Test_withCamera.nb*. Ukázkou toho, jak probíhalo testování, je možné vidět na obrázku 4.8.

Pro kameru GoPro HERO4 probíhalo testování ještě před úplným dokončením knihovny *GoProController* z důvodu zapůjčení této kamery pouze na omezenou dobu. Funkčnost a implementace funkce *goProSet* se od té doby nezměnila a tato funkce je tedy řádně otestována. Měnila se pouze struktura funkce *goProGet*, která v té době nebyla ještě kompletně implementována. Proto k zjištění aktuálního nastavení kamery HERO4 v notebooku *HERO4_Test_withCamera.nb* používáme i jiné funkce.

4.2.3 Problémy při testování

Jelikož testování složení url adres slouží spíše ke kontrole nastavených konstant v knihovně (asociativní pole), požadovali jsme v tomto testu stoprocentní úspěšnost. To se nám povedlo splnit. V druhém způsobu testování, tedy testování nastavení kamery, jsme neobjevili žádné implementační chyby. Ovšem to neznamená, že jsme se nesetkali s problémy při nastavování parametrů kamery.

U kamery GoPro HERO3+ Black edition jsme se čas od času setkali s vypínáním kamery či WiFi sítě vysílané kamerou po odeslání HTTP požadavku na kameru. Tento jev byl pozorován jen zřídka, ale nepodařilo se odhalit jeho příčinu, neboť k němu nedocházelo pravidelně při konkrétní činnosti. To znamená, že se tak stalo ojediněle u nastavování parametrů, mazání souborů či výpisu nastavení kamery. Pokud jsme znovu spustili příkaz, který vypl kameru či WiFi síť, ihned po zapnutí kamery či WiFi sítě, nedošlo znovu ke stejnému problému. Tudíž nelze s jistotou říci, co takového chování způsobuje a taková chyba pravděpodobně není chybou implementační.

PhotoResolution

```
goProGetPossiblePhotoRes []
{5M, 7W, 12W, 7M}

goProSet[photoResolution → "5M"]
goProGet["photoResolution"]
goProSet[photoResolution → "7W"]
goProGet["photoResolution"]
goProSet[photoResolution → "12W"]
goProGet["photoResolution"]
goProSet[photoResolution → "7M"]
goProGet["photoResolution"]
goProSet[photoResolution → "12W"]
goProGet["photoResolution"]

{photoResolution → 5M}

{photoResolution → 7W}

{photoResolution → 12W}

{photoResolution → 7M}

{photoResolution → 12W}
```

Obrázek 4.8: Ukázka testování nastavování za přítomnosti kamery.

S dalším nežádoucím jevem jsme se setkali při mazání všech souborů na kameře najednou. V jednom jediném případě se nám stalo, že po smazání souborů se na displeji kamery objevil nápis „No SD“. Tedy kamera nerozpoznala vloženou SD kartu. Po vložení SD karty do počítače a jejím naformátování byl tento problém odstraněn. S tímto jevem jsme se ovšem setkali i u manuálního mazání dat z GoPro kamery, tedy při mazání dat pomocí tlačítek přímo na kameře. Tento jev je tedy pravděpodobně způsoben firmwarem kamery.

4.2.4 Shrnutí

Testování probíhalo pouze na modelech HERO3+ Black edition a HERO4 Silver edition a můžeme tedy říci, že pro tyto dva modely knihovna funguje. Pro ostatní modely, jejichž přehled naleznete v tabulce 3.2, je ovládání víceméně totožné jako u dvou modelů použitých při testování. Rozdíl nalezneme pouze v parametrech určitých nastavení (tyto parametry jsou uloženy v asociativních polích).

Do jisté míry jsme byli schopni otestovat knihovnu i pro další modely kamer. A to na modelech, které jsme měli k dispozici, pokud jsme testovali parametry, jenž jsou totožné pro více kamer. Domníváme se tedy, že knihovna bude fungovat správně i pro modely, na kterých nebyla testována.

Závěr

Cílem práce bylo vytvořit knihovnu funkcí pro prostředí Wolfram Mathematica, která by umožňovala vzdálené ovládání kamer GoPro a nastavování parametrů těchto zařízení. Tyto cíle jsme splnili a otestovali funkčnost námi implementované knihovny jménem *GoProController* na modelech GoPro HERO3+ Black Edition a HERO4 Silver Edition.

Knihovna obsahuje funkce pro ovládání kamery (zapnutí, vypnutí, změna nahrávacích módů), změnu nastavení parametrů (rozlišení nahrávání, rozlišení fotografie, počet snímků za vteřinu atd.), stahování pořízených záznamů a jejich mazání, ukládání stávajícího nastavení kamery do souborů s příponou „.xls“ a načtení uloženého nastavení zpět do kamery.

V rámci práce jsme také vytvořili dokumentaci knihovny pro dokumentační centrum Wolfram Mathematica, která navíc obsahuje návody k používání knihovny pro různé modely kamer. Knihovna je volně dostupná a lze ji stáhnout na adrese <https://github.com/slademi2/goProController>.

V budoucnu je možné přidávat do naší knihovny podporu pro další kamery. K tomu stačí přidat další zdrojový soubor a zajistit, aby struktura tohoto souboru byla shodná se strukturou použitou u ostatních zdrojových souborů.

Literatura

- [1] GoPro Apps. [online], [cit: 2017-05-02]. Dostupné z: <https://shop.gopro.com/EMEA/softwareandapp/>
- [2] Image Processing & Analysis. [online], 2017, [cit. 2017-04-09]. Dostupné z: <https://reference.wolfram.com/language/guide/ImageProcessing.html>
- [3] Stack Exchange Inc. [online], červen 2015, [cit. 2017-04-09]. Dostupné z: <https://superuser.com/questions/877280/how-do-i-wirelessly-transfer-data-from-a-gopro-hero4-to-a-mac>
- [4] The unofficial GoPro Wifi API. [online], [cit: 2017-05-02]. Dostupné z: <https://github.com/KonradIT>
- [5] GoPro product manuals. [online], [cit: 2017-05-02]. Dostupné z: <https://gopro.com/help/HERO3Plus-Black-Edition>
- [6] Options. [online], 2017, [cit. 2017-04-09]. Dostupné z: <https://reference.wolfram.com/language/ref/Options.html>

Seznam použitých zkratk

fps Frames per second

Mpx Megapixel

URL Uniform Resource Locator

HTTP Hypertext Transfer Protocol

WoL Wake on LAN

Obsah přiloženého CD

	readme.txt.....	stručný popis obsahu CD
	GoProController.....	adresář obsahující knihovnu GoProController
	src	
	thesis	zdrojová forma práce ve formátu $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$
	BP_Michal_Sládek_2017.pdf	text práce ve formátu PDF
	ZZP_Michal_Sládek_2017.pdf ..	zadání závěrečné práce ve formátu PDF