

# Projekt 3

---

## Možnosti rozšíření Raspberry Pi o modul kamery

2014

## Úvod

Cílem této práce je prozkoumat možnosti rozšíření minipočítače Raspberry Pi o modul kamery, který by následně bylo možné použít pro snímání obrazu. Práce se zabývá možnostmi USB webkamer i dedikovaných modulů určených konkrétně pro Raspberry Pi. Následně je zmíněna i práce s kamerou v prostředí Linuxu na RPi a v programovacím jazyce Python.

## Kamery pro Raspberry Pi

Existují dvě možnosti rozšíření Raspberry Pi o modul kamery. První univerzálnější řešení je pomocí běžné webkamery, která se připojuje do jednoho z USB slotů. Druhým řešením je využít speciálního modulu určeného výhradně pro minipočítače typu Raspberry Pi. Všechny zmíněné možnosti platí pro všechny, v tuto dobu, dostupné druhy RPi, tedy modely A i B včetně + variant.

### USB webkamery

USB zařízení představují nejjednodušší způsob rozšíření. Nevýhodou může být horší dostupnost ovladačů pro jednotlivé typy kamer. Dále také nižší rychlost než v případě CSI modulů.

Osobně jsem vyzkoušel 2 USB webkamery, které jsem měl k dispozici. Jednalo se o typ Lone Phenix-Q8 a Logitech QuickCam. Stáří kamer je 7 let, nelze tedy očekávat žádné perfektní snímky, přesto šlo o dobrý způsob, jak začít a seznámit se s prací se záznamem v OpenCV.

Kameru Lone se mi, v důsledku nedostupných driverů, vůbec nepodařilo zprovoznit. Druhou kameru od Logitechu ano, avšak pouze v jakémsi základním režimu. Přestože maximální rozlišení je 1280×960, tak kamera pracovala pouze s rozlišením 640 x 480 a nižším. Kvalita byla velice špatná a pro účel rozpoznávání obrazu nedostatečná. Protože toto zařízení bylo určeno jako přenosná webkamera k notebooku s (tehdy) průměrnými parametry, nepřišlo mi opodstatněné investovat čas v nejistém podniku hledání a experimentování s jinými ovladači, které by mohly zprovoznit maximální rozlišení.

Při testování kvality obrazu a rychlosti snímkování jsem využil programu pro Linux Fswebcam. Práce s tímto programem probíhá výhradně v terminálu, kam se pokyny zadávají ve formě příkazů. Tento program jsem použil vlastně pouze k vyzkoušení funkčnosti. V případě potřeby by ale bylo možné využít jazyku Python a pomocí příkazu `os.system()` posílat příkazy do terminálu přes něj. S kamerou jsem poté pracoval pomocí modulu OpenCV – práce s ním bude popsána později.

Vzhledem k tomu, že u USB webkamer není vůbec zaručena dostupnost ovladačů a vzhledem k cenám kvalitních kamer, které by vyhovovaly svými parametry, jsem se rozhodl zaměřit se na kamery do CSI portu, které jsou určeny přímo pro RPi a slibují za nižší cenu lepší kvalitu obrazu.

### Moduly do CSI rozhraní

Minipočítač Raspberry Pi je vybaven speciálním rozhraním MIPI (Mobile Industry Processor Interface) CSI (Camera Serial Interface) typu 2 (CSI-2). Toto rozhraní zajišťuje přímé spojení s procesorem Raspberry. Stejně rozhraní se ve velké míře používá i u fotoaparátů v mobilních telefonech, pro které bylo původně navrženo. [1] Komunikace je pouze jednosměrná (z modulu kamery do mikropočítače). Velikou výhodou tohoto rozhraní je jeho vysoká datová propustnost, která dosahuje 1Gb/s na jednu datovou cestu. Více informací, včetně zapojení jednotlivých pinů konektoru lze nalézt

v [1].Konektor se na desce nachází mezi HDMI a Ethernet portem a modul kamery se k němu připojuje pomocí plochého 15-žilého flex kabelu. Umístění konektoru je zobrazeno na Obr. 1.

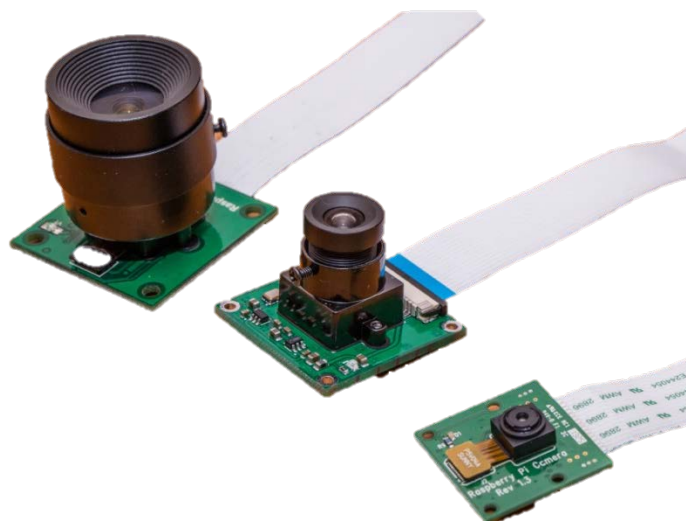


Obr. 1 - CSI rozhraní s připojeným flex kabelem

### Konkrétní kamery

Všechny kamery co jsem objevil určené pro RPi jsou si velmi podobné. Všechny obsahují totožný snímáči senzor Omnivision 5647 s maximálním rozlišením 5 megapixelů (2592x1944). Rozměr čipu je 1/4". Kamery vybavené tímto snímačem jsou schopné zajistit 15 snímků za sekundu při maximálním rozlišení a 30 snímků za sekundu při rozlišení 1080p. Kompletní specifikaci tohoto senzoru lze nalézt v [2].

Raspberry Foundation vydalo vlastní oficiální modul s CSI rozhraním. Aktuálně se prodává revize 1.3. Tento modul je velice malý, rozměry jsou pouze 20 x 25 x 9 mm je tudíž vhodná pokud jsou kladeny nároky na prostor. Nevýhodou oficiálního modulu se zdá být jeho nevyměnitelná optika. Už na první pohled připomíná tu, co můžeme nalézt v mobilních telefonech. Tomu odpovídají i snímky modulem pořízené. Je třeba zmínit, že RPi kamera bude většinou stacionární (jako na stativu) a proto jsou výsledné snímky poměrně dobré. Oficiální modul je úplně napravo na Obr. 2. Tuto kamerku lze u nás pořídit od cca 700 Kč.



Obr. 2 - Druhy kamer do CSI rozhraní

Neoficiální moduly obvykle nabízí výhodu v podobě vyměnitelné optiky. Většina má standardní CS mount, na který lze našroubovat jiná standardní skla. Kamerky se v základu dodávají s rozličnými skly, přičemž modul bývá stejný/podobný. Jak již bylo zmíněno výše, snímací senzory jsou totožné – jde tedy pouze o zvolení dané optiky. Tato možnost byla důvodem, proč jsem se rozhodl pořídit místo oficiálního modulu ten neoficiální. Výrobky pocházejí většinou z Asie a u nás jsem na ně nenašel. Ceny neoriginálních modulů se pohybují v přepočtu od 350 Kč za základní s pevnou čočkou (obdobu oficiální RPi kamery) po asi 1000 Kč, kde se ale převážně projevuje cena výměnných čoček.

Mezi moduly jsou běžné i kamery bez infračerveného filtru. Ty se hodí pro fotografování za velmi špatného světla nebo i v noci, kdy lze použít externí přisvětlení infračervenými diodami. Pro běžné použití se nehodí, protože barvy snímků nejsou věrné, jak je ukázáno na Obr. 3. Tyto kamery jsou označovány jako NoIR a poznávacím znakem je jejich obvyklá černá barva. Ceny jsou téměř totožné jako u varianty s IR filtrem. Originální Raspberry Pi camera se vyrábí rovněž ve variantě bez IR filtru – zde je cena mírně vyšší než u běžného provedení.



Obr. 3 - Porovnání Obrazu z RPi Camera a z RPi Camera NoIR [3]

Po zvážení alternativ jsem se rozhodl pro neoficiální modul, právě díky kvalitnější vyměnitelné optice. Kamera se dodává spolu s čočkou LS-40136 a je možné ji nahradit jinou se závitem M12x0,5 nebo CS Mount. Ostření probíhá šroubováním objektivu. Kamerka je zobrazena na Obr. 2 a její parametry shrnuje Tab. 1. V zahraničí lze pořídit od 450 Kč. Prodejce tuto modul označuje jako Raspberry Pi Camera Board /w M12x0.5 mount Lens. Balení obsahuje 15cm flex kabel, v případě potřeby lze dokoupit delší.

Optický snímač	Omnivision 5647		
Rozlišení	5 Mpx (2592 x 1944) fotografie	1080p video	
Snímkovací frekvence	30 snímků/s @ 1080p		
Použitá optika	LS-40136		
Ohnisková vzdálenost	3,2 mm		
Zorný úhel	D 85°	H 63,7°	V 70°
Držák optiky	M12 x 0,5 nebo CS Mount		
Rozměry	36 x 36 mm		

Tab. 1 - Vlastnosti zakoupené RPi kamery



Obr. 4 - Zakoupená Raspberry Pi Camera

## Práce s USB webkamerou

Práce se zařízeními připojenými přes USB je, v prostředí Raspbianu, poměrně jednoduchá. Po připojení zařízení stačí zkontrolovat, zda jej systém korektně rozpoznal. K tomu slouží následující terminálový příkaz, který vypíše všechna připojená zařízení včetně jejich portu.

```
lsusb
```

Instalace balíčku `fswebcam` [4] obsluhující kameru provedeme příkazem

```
sudo apt-get install fswebcam
```

Nyní lze již program používat. Vše probíhá v prostředí systémového terminálu. Syntaxe je vždy `fswebcam` a název fotografie včetně přípony, volitelně lze připojit další parametry, které ovlivní

```
fswebcam fotka.jpg  
fswebcam -r 1280x720 --no-banner fotka.jpg
```

obrázek. První příkaz ukazuje základní použití, druhý je včetně dalších parametrů.

Parametr `-r` určuje rozlišení snímku a přidáním `--no-banner` odstraníme proužek s časovou informací.

## Spolupráce USB kamery s Pythonem a modulem OpenCV [5]

Modul OpenCV (Open Source Computer Vision), což je knihovna pro realtime práci s obrazovými daty, lze využít i pro práci s kamerou. Následující kus kódu vytvoří fotografii s názvem `fotka.jpg` a uloží ji do paměti. Stejný postup jsem ozkoušel i v notebooku s integrovanou kamerou, kde je rovněž funkční.

```
import cv2  
  
cam_port = 0 #cislo portu kamery  
cap = cv2.VideoCapture(cam_port)  
  
cap.read() #adjustace svetla  
img_name = "fotka.png" #nazev obrazku  
ret, img = cap.read() #vlastni foceni  
cv2.imwrite(img_name, img) #ulozeni do souboru  
  
cap.release() #uvolneni kamery
```

Nejprve se provede import knihovny cv2. Poté definuji port, na kterém se nachází kamera. Číslování běží od 0. Lze rovněž jako číslo portu zvolit -1, kdy je vybrána první dostupná kamera. Následuje vytvoření objektu cap, což je inicializace kamery. Dále je nutné provést alespoň jedno focení „nanečisto“, kdy dojde k adjustaci světla. Tento krok není v oficiální dokumentaci zmíněn, bez něho však program nefunguje. Následuje vlastní focení a uložení snímku do souboru. Nakonec je třeba uvolnit kameru, aby ji šlo opět použít.

Následující kód vytvoří okno, ve kterém se zobrazí kontinuální snímání. K zavření okna dojde po stisku klávesy „q“.

```
import cv2

cap = cv2.VideoCapture(0)      #inicializace
cap.read()                    #adjustace svetla

while(True):
    ret, frame = cap.read()
    cv2.imshow('Fotka',frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()                 #uvolneni kamery
cv2.destroyAllWindows()      #zavreni vseh vytvorených oken
```

Program funguje podobně jako v předchozím příkladu. Změnou je cyklus, ve kterém dochází k pořizování a zobrazování snímků. O vykreslení se stará příkaz *imshow()*. Program poté vyčká 1 ms (což vlastně znamená, že snímkovací frekvence je maximální co kamera dovolí), než začne pořizovat nový snímek. Cyklus se přeruší stiskem klávesy „q“. Následuje uvolnění kamery a uzavření okna.

## Práce s Raspberry Pi Camera

Kamery do CSI rozhraní vyžadují na rozdíl od jejich USB protějšků určitá nastavení, bez nichž nebudou na RPi pracovat. Nejprve je vhodné provést aktualizaci systému. Ta bude trvat, v závislosti na stavu systému, několik minut. Následující příkazy je třeba zapsat do systémového terminálu a RPi musí být připojeno k internetu.

```
sudo apt-get update
sudo apt-get upgrade
```

Následně je třeba povolit kamerový modul v nastavení.

```
sudo raspi-config
```

Objeví se nabídka s rozličnými nastaveními Raspberry. Důležitá je položka „camera“ a v ní zvolit „enable“. Poté je třeba RPi restartovat.

Systém Raspbian v sobě již obsahuje obslužný software pro RPi kameru - RaspiCam. Jde o obdobu fswebcam, pracuje se s ním výhradně v prostředí systémového terminálu. První níže uvedený příkaz vyfotí snímek, u druhého jsou doplněny další volitelné parametry. Při focení je zobrazován náhled na obrazovce.

```
raspistill -o fotka.jpg
raspistill -vf -hf -w 1920 -h 1080 -o fotka2.jpg
```

Parametrem `-o` určujeme výstupní soubor, parametry `-vf` a `-hf` značí vertikální resp. horizontální přetočení obrázku. Není-li definováno rozlišení parametry `-w` a `-h`, vytvoří se snímek v plném 5Mpx rozlišení. Není-li dáno jinak, před vytvořením fotky se zobrazí 5s náhled přes celou obrazovku.

Pro vytvoření videa slouží příkaz `raspivid`. Bez zadaných parametrů vytvoří 5s klip s rozlišením 1080p. Výstupní soubor je formátu `*.h264`. Video je bez zvuku, protože modul nemá mikrofon.

```
raspivid -o video.h264
raspivid -t 10000 -o video2.h264 -b 3500000 -f 15
```

Parametr `-t` udává délku videa v ms, parametr `-b` bitrate v bit/s a parametr `-f` snímkovací frekvenci.

Aplikace nabízí širokou paletu možností od otáčení a zrcadlení obrazu přes práci s barvou (filtry, efekty atd.) a expozicí až po doprovodné nastavení jako je náhled při focení. Přehled všech parametrů dostupných pro `raspistill` a `raspivid` je uveden v dokumentaci k `RaspiCam`. [6]

## Využití kamery v programovacím jazyce Python

Pro pozdější využití kamery potřebuji, aby ji bylo možno provozovat s jazykem Python. Jednou z možností je využít program `RaspiCam` a volat jej pomocí Pythonského příkazu `os.system()`. Příkaz tedy může vypadat následovně – obrázek se bude jmenovat `foto.jpg` a rozlišení bude mít `1920x1080`.

```
import os
os.system("raspistill -o foto.jpg -w 1920 -h 1080")
```

Jako argument uvnitř `os.system()` lze samozřejmě použít cokoliv z [6].

Výše uvedené řešení je funkční, přesto je vhodnější, aby se nemusel volat systémový terminál, kde může docházet ke zdržením – toho lze docílit využitím modulů pro Python `Picamera` nebo `OpenCV`.

## Picamera

`Picamera` je modul pro python, jehož cílem je zastoupit oficiální aplikaci `RaspiCam`. S pomocí tohoto modulu může práce s kamerou probíhat výhradně v python skriptu, bez volání dalších subprocessů. Výhodou `Picamery` oproti později zmíněnému `OpenCV` je, že je speciálně vytvořena pro RPi kamerku. Zatímco `OpenCV` je univerzální a spolupracuje i s RPi kamerou, tak jsou jeho možnosti omezenější a bohužel, o čemž jsem se sám několikrát přesvědčil, méně spolehlivé. K modulu `Picamera` existuje rovněž poměrně pěkná dokumentace [7].

Instalace do Raspbianu probíhá standardně, zde je postup pro Python 2.7.

```
sudo apt-get install python-picamera
```

Zde je příklad použití, jak je vidět je funkce příkazů na první pohled poměrně srozumitelná.

```
import picamera
from time import sleep

camera = picamera.PiCamera()
camera.capture('image.jpg')

camera.start_preview()
camera.vflip = True
camera.hflip = True
camera.brightness = 60

camera.start_recording('video.h264')
sleep(5)
camera.stop_recording()
```

Tento úryvek kódu nejprve importuje potřebné moduly a inicializuje kameru. Následně vyfotí snímek. Poté dojde k zobrazení náhledu na obrazovku. Nastaví se parametry – vertikální a horizontální přetočení a nastavení kontrastu. Nakonec se vytvoří 5s videoklip.

### RPI camera a OpenCV

Práce s RPI kamerou v OpenCV probíhá víceméně stejně jako v případě USB webkamery. Ve výchozím stavu však kamera s OpenCV spolupracovat nebude. Nejprve je třeba načíst driver terminálovým příkazem.

```
sudo modprobe bcm2835-v4l2
```

Driver zůstane v paměti až do dalšího restartu systému v Raspberry. Aby bylo možné vše volat pomocí Pythonu, je opět výhodné využít příkazu `os.system()`.

Následující část je obdoba již dříve použitého kódu pro zobrazování scény na obrazovku.

```
import cv2
import os

os.system('sudo modprobe bcm2835-v4l2')

cap = cv2.VideoCapture(0)
cap.set(3, 480)          #sirka videa
cap.set(4, 320)         #vyska videa
cap.read()              #adjustace svetla

while(True):
    ret, frame = cap.read()          #foceni
    cv2.imshow('frame',frame) #zobrazeni
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()              #uvolneni kamery
cv2.destroyAllWindows() #zavreni oken
```

Navíc je zde načtení video driveru a definice rozlišení. Testováním jsem zjistil, že RPI je schopné plynule zobrazovat do rozlišení 480 x 320. U vyšších dochází již k znatelnému zpomalení a trhání. Nejvyšší rozlišení, které se mi podařilo zprovoznit v tomto režimu, bylo 800 x 600. Při vyšších z neznámého důvodu byla výplň okna černá.

### Rychlost kamery na RPi

V rámci práce jsem se rozhodl otestovat, jakých snímkovacích frekvencí lze s dostupnou kamerkou na RPi dosáhnout. Rychlost snímkování jsem měřil pro různá rozlišení od plných 5 MPx až po 320 x 240. Fotky jsem ukládal v jednom pokusu do paměti RAM, tedy velmi rychlé paměti, kde odpadá zdržení vlivem pomalé SD karty a v druhém pokusu právě na kartu SD (v tomto případě karta s rychlostní třídou 10).



Měření jsem prováděl pomocí následujícího kódu založeného na modulu Picamera, zde ukládání do RAM.

```
import time
import picamera

frames = [15,25,30,40,50,60,60,60,60]
xres = [2592,1920,1600,1280,1024,800,640,480,320]
yres = [1944,1080,900,720,600,600,480,320,240]
j = 0

while (j < len(xres)):
    with picamera.PiCamera() as camera:
        camera.resolution = (xres[j], yres[j])
        camera.start_preview()
        time.sleep(2) #autoexpozice
        start = time.time()
        cesta = '/run/shm/image%03d.jpg' #/run/shm je RAMdisk
        camera.capture_sequence((cesta % i for i in range(frames[j])), \
            use_video_port=True)
        print('%d %dx%d: Vyfoceno %d snímku pri %.2ffps' \
            %(j+1,xres[j],yres[j],frames[j],(frames[j]/(time.time()-start))))
        camera.stop_preview()
        j = j + 1
```

Nejprve dojde k importu potřebných modulů. Následuje list s počtem snímků, které se budou fotit a listy s rozlišením. V těle hlavního cyklu dojde k nastavení parametrů snímku (rozlišení z listu). Náhled je zde kvůli adjustaci světla pro auto expozici. Následuje fotící sekvence a výpis výsledků.

Výsledky měření shrnuje tabulka Tab. 2.

Rozlišení	Snímků/s (do RAM)	Snímků/s (na SD)
2592 x 1944	6,4	2 – 5
1920 x 1080	15,8	4,5 – 12,5
1600 x 900	22,2	10,3 – 16,7
1280 x 720	27,3	21,5 – 27,2
1024 x 600	27,8	24 – 27,7
800 x 600	28	27,5
640 x 480	28,3	28
480 x 320	28,1	28,1
320 x 240	28,3	28

Tab. 2 - Snímkovací frekvence RPi kamery v závislosti na rozlišení

Z tabulky je vidět zpomalení při zápisu na kartu SD, zvláště při rozlišení 1920 x 1080. Rychlosti zápisu na SD kartu rovněž vykazují velký rozptyl, který se u zapisování do RAM téměř nevyskytuje. Rychlosti jsou poměrně dost závislé na fotografované scéně (která se projevuje na velikosti souboru). Měření jsem zkoušel i po přetaktování RPi, kdy jsem očekával vyšší snímkovací frekvence, ale oproti očekávání nemělo taktování téměř žádný vliv. U nižších rozlišení se snímkovací frekvence drží na přibližně 28 fps a už dále nestoupá.

## Měření rychlosti FFT

Jako jeden z možných přístupů jak porovnávat obraz je využití rychlé Fourierovy transformace, neboli FFT (Fast Fourier Transform ). Díky FFT lze porovnat vyfocený obraz s jiným obrazem uloženým v databázi a hledat shodu. Tato část má za úkol zjistit, rychlosti výpočtu FFT na RPi, vliv rozlišení obrazu a jeho datové velikosti.

Pro měření jsem použil funkci z modulu Numpy – `fft2()`. Funkce vyžaduje jediný argument s obrázkem - ten načítám pomocí modulu OpenCV příkazem `imread()`. Pro měření času posloužil zabudovaný modul `time` a příkaz `time()`.

Rychlost jsem zjišťoval na 3 sadách obrázků o 9 různých rozlišení. První sada obsahuje obrázky ve formátu jpg s nízkou kompresí, které jsou datově největší. Druhá sada obsahuje tytéž fotky, ovšem silně komprimované. Přestože datová úspora je značná, viditelně jsou obrázky téměř stejné. Poslední sada obsahuje prázdné snímky, pouze s bílou barvou. Tyto sady měly ukázat zda, případně jaký vliv, má snímaná scéna (a tedy datová velikost) na dobu načítání obrázku a na FFT. Raspberry jsem testoval s dvěma stupni přetaktování – pro původních 700 MHz a potom 800 a 900 MHz. Pro porovnání jsem měření provedl i na notebooku s Windows. Výsledky se všemi důležitými parametry zobrazuje Tab. 3. Zkratka (VK) u hodnoty rozlišení značí snímky s vysokou mírou komprese, (NK) snímky s nízkou kompresí (největší), zkratka (B) označuje prázdné, čistě bílé obrázky. Výsledky uvedené v tabulce jsou průměrem z 15 měření.

Jak se ukázalo, výpočet FFT je na RPi poměrně dost zdlouhavý. Pro snímek s nejvyšším rozlišením (2592 x 1944) RPi vůbec nedokáže FFT provést, skončí chybou *Memory Error* (Jak se ukázalo i na PC, výpočet je poměrně paměťově náročný). S 2Mpx snímkem již transformaci zvládne, ovšem s časem výpočtu přesahujícím 10 s. Datová velikost snímku ovlivňuje dobu jeho načtení, na rychlost FFT ale vliv nemá. Vliv přetaktování má naopak pozorovatelný pozitivní efekt.

Z výsledků vyplývá, že se porovnávání snímků s databází a hledání shody pomocí FFT se tedy pro rozpoznávání SPZ příliš nehodí, tedy alespoň ne testované řešení z Numpy. Pro databázi větší než několik obrázků by trvalo porovnávání příliš dlouho a to i na mnohem silnějším hardwaru než kterým disponuje RPi.

Rozlišení	Velikost [kiB]	RPi @ 700 MHz (základ)		RPi @ 800 MHz		RPi @ 900 MHz		PC @ 2,53 GHz	
		Načtení [ms]	FFT [ms]	Načtení [ms]	FFT [ms]	Načtení [ms]	FFT [ms]	Načtení [ms]	FFT [ms]
2592 x 1944 (B)	95	-	-	-	-	-	-	118	1407
2592 x 1944 (VK)	312	-	-	-	-	-	-	138	1406
2592 x 1944 (NK)	2540	-	-	-	-	-	-	240	1393
1920 x 1080 (B)	46,2	687	11335	710	11268	554	9616	50	564
1920 x 1080 (VK)	171	770	10515	744	9228	608	8682	59	560
1920 x 1080 (NK)	1440	1080	10473	990	9078	846	8690	115	562
1600 x 900 (B)	35,9	460	7059	422	6498	366	5774	34	404
1600 x 900 (VK)	135	530	7109	485	6559	423	5779	42	415
1600 x 900 (NK)	1080	754	7054	685	6513	601	5781	83	413
1280 x 720 (B)	27	290	4319	268	4019	234	3558	23	268
1280 x 720 (VK)	135	348	4342	320	4015	280	3574	33	250
1280 x 720 (NK)	817	514	4345	443	3741	402	3560	54	249
1024 x 600 (B)	22,5	199	2951	179	2767	156	2456	19	179
1024 x 600 (VK)	98,2	232	2934	212	2743	189	2445	20	168
1024 x 600 (NK)	589	347	2932	315	2732	273	2443	37	165
800 x 600 (B)	20,2	149	2166	131	1892	122	1784	11	137
800 x 600 (VK)	79,2	182	2154	164	2004	143	1783	20	135
800 x 600 (NK)	434	263	2165	233	1855	207	1782	33	140
640 x 480 (B)	17,2	96	1333	88	1233	80	1095	8	76
640 x 480 (VK)	60	117	1327	106	1230	94	1090	9	78
640 x 480 (NK)	302	173	1327	153	1236	136	1097	19	78
480 x 360 (B)	15,1	50	729	45	677	40	599	6	47
480 x 360 (VK)	44,8	64	724	55	675	47	597	8	48
480 x 360 (NK)	194	98	731	86	673	77	595	21	47
320 x 240 (B)	13,4	22	274	20	251	18	225	2	17
320 x 240 (VK)	33,1	28	277	26	257	23	228	2	17
320 x 240 (NK)	99	42	280	46	240	34	227	5	18

Tab. 3 - Rychlosti načítání obrázků a FFT

## Závěr

V rámci této práce jsem provedl průzkum dostupných kamer pro RPi a vyzkoušel práci s nimi. Vyzkoušel jsem jejich spolupráci s několika obslužnými programy a moduly do jazyka Python. Jako nejvhodnější kameru jsem vybral neoficiální Raspberry Pi Camera s výměnnou optikou. S touto kamerou jsem testoval spolupráci s RPi. Jako nejspolehlivější se mi jeví obslužný software RaspiCam a modul Picamera pro Python. OpenCV je rovněž funkční, ale nabízí méně možností (při obsluhování kamery) s nižší spolehlivostí. Rychlost FFT s modulem Numpy je s RPi nízká a hodí se tak pouze jako doplňkové řešení při rozpoznávání obrazu.

K práci jsem měl k dispozici Raspberry Pi model B s operačním systémem Raspbian. Python jsem používal v nejnovější verzi 2.7.8, modul OpenCV ve verzi 2.4.9 a modul Picamera ve verzi 1.8.

## Seznam zkratk

RPi – Raspberry Pi

IR – Infračervený (Infrared)

FFT – Rychlá Fourierova transformace (Fast Fourier Transform)

## Bibliografie

- [1] P. Vis, „Raspberry Pi CSI Camera Interface,“ [Online]. Available: [http://www.petervis.com/Raspberry\\_Pi/Raspberry\\_Pi\\_CSI/Raspberry\\_Pi\\_CSI\\_Camera\\_Interface.html](http://www.petervis.com/Raspberry_Pi/Raspberry_Pi_CSI/Raspberry_Pi_CSI_Camera_Interface.html). [Přístup získán 2014].
- [2] „Omnivision OV5647 datasheet,“ [Online]. Available: [http://www.seeedstudio.com/wiki/images/3/3c/Ov5647\\_full.pdf](http://www.seeedstudio.com/wiki/images/3/3c/Ov5647_full.pdf). [Přístup získán 2014].
- [3] „PiNoIR – what’s it for? Comparison of RasPiCam and Pi NoIR output in daylight,“ [Online]. Available: <http://raspi.tv/2013/pinoir-whats-it-for-comparison-of-raspicam-and-pi-noir-output-in-daylight>. [Přístup získán 2014].
- [4] „Using a standard USB webcam,“ [Online]. Available: <http://www.raspberrypi.org/documentation/usage/webcams/>. [Přístup získán 2014].
- [5] „Getting Started with Videos,“ [Online]. Available: [http://docs.opencv.org/trunk/doc/py\\_tutorials/py\\_gui/py\\_video\\_display/py\\_video\\_display.html](http://docs.opencv.org/trunk/doc/py_tutorials/py_gui/py_video_display/py_video_display.html). [Přístup získán 2014].
- [6] „RaspiCam Documentation,“ [Online]. Available: <http://www.raspberrypi.org/wp-content/uploads/2013/07/RaspiCam-Documentation.pdf>. [Přístup získán 2014].
- [7] „Picamera,“ [Online]. Available: <http://picamera.readthedocs.org/en/release-1.8/index.html#>. [Přístup získán 2014].