

Jihočeská univerzita v Českých Budějovicích

Přírodovědecká fakulta

Konstrukce fotopasti pomocí počítače Raspberry Pi

Bakalářská práce

Daniel Šváb

Školitel: Mgr. Jiří Pech, PhD.

České Budějovice 2016

Bibliografické údaje

Šváb, D., 2016: Konstrukce fotopasti pomocí počítače RaspBerry Pi. [Construction photo trap using a computer RaspBerry Pi. Bc. Thesis, in Czech.] – 59 p., Faculty of Science, University of South Bohemia, České Budějovice, Czech Republic.

Anotace

Cílem této bakalářské práce je sestrojít funkční Fotopast pomocí minipočítače Raspberry Pi verze 2. Fotopast bude použita jako bezpečnostní kamera, která bude pořizovat fotografie na základě rozpoznání pohybu. Jakmile je pořízena fotografie, Raspberry Pi ji automaticky pošle vlastníkově na email a na mobil pošle notifikaci o pohybu.

Klíčová slova

Raspberry Pi, kamera, pohyb, python, Raspbian, Fedora, Ubuntu

Abstract

The aim of this bachelor thesis is to design a functional photo trap using the platform of oneboard computer Raspberry Pi version 2. Photo trap is going to be used as CCTV or Security camera which takes a photo every time it recognizes a movement. When the photo is taken, Raspberry Pi sends it to owner's e-mail address and a notification about a movement detection is sent via Pushetta app to his mobile.

Keywords

Raspberry Pi, camera, movement, python, Raspbian, Fedora, Ubuntu

Prohlášení

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

České Budějovice

.....

(podpis autora)

Poděkování

Rád bych poděkoval Mgr. Jiřímu Pechovi, Ph.D. za cenné rady, věcné připomínky a vstřícnosti při konzultacích a vypracování bakalářské práce.

Obsah

Úvod	1
1. Teoretická část	2
1.1. Fotopast.....	2
1.2. Seznámení s Raspberry Pi 2.....	3
1.3. Operační systémy	4
2. Hardware	4
2.1.1. Wi-Fi adaptér	5
2.1.2. Pi Camera	6
2.1.3. PIR.....	7
2.2. Software	8
2.2.1. Python	8
2.2.2. Pushetta	8
2.2.3. Weaved.....	8
2.2.4. Motion	9
3. Metodika	9
4. Praktická část.....	11
4.1. Raspbian	13
4.1.1. Testování	20
4.1.2. Shrnutí pro Raspbian	24
4.2. FedBerry	26
4.2.1. Testování	30
4.2.2. Shrnutí pro FedBerry	34
4.3. Ubuntu.....	35
4.3.1 Testování	39
4.3.2 Shrnutí pro MATE	42
5. Celkové srovnání	44
6. Závěr	46
Seznam použité literatury	47
Seznam obrázků	50
Seznam příloh.....	50
Přílohy.....	51

Úvod

V dnešní době je fotopast velice rozšířená věc, hodně využívaná při pozorování zvířat nebo jako bezpečnostní zařízení. Fotopasti používané v lesích vydrží v provozu i týden a fotografie se ukládají přímo v zařízení. Fotografie lze vyzvednout na paměťové kartě a zkontrolovat zařízení. U fotopasti, které se využívají jako bezpečnostní zařízení v budovách, firmách, obchodech a soukromých pozemcích je potřeba, aby se fotografie odeslala majiteli ihned, dřív než si pachatel všimne fotopasti a zneškodní ji. Při detekci pohybu přijde majiteli notifikace s přiloženou fotografií, podle které lze zjistit, jestli nejde o planý poplach, zda je zachycen na fotografii pachatel. Tyto fotopasti musí mít GSM modul pro okamžité odeslání fotografií na mobil, nebo do e-mailové schránky.

Zhotovením fotopasti pomocí počítače Raspberry Pi, bude vytvořeno zařízení za mnohem menší náklady než u profesionálních zařízení. Lze tím dokázat, zdali je to proveditelné pomocí embedded zařízení typu Raspberry Pi, bez využití GSM modulu, pouze přes Wi-Fi síť nebo pomocí síťového kabelu. Dále lze porovnat možnosti použití fotopasti na odlišných operačních systémech.

Praktická část je zaměřena na hardware a software fotopasti a její postupnou realizaci. Dále také bude u každého systému testována použitelnost a stabilita.

Cílem bakalářské práce je navrhnout a sestavit fotopast pomocí jednodeskového počítače Raspberry Pi 2. Následně z několika operačních systémů, volně dostupných, vybrat nejvhodnější systém a zprovoznit fotopast. Po zprovoznění a otestování výsledek zveřejnit na GitHubu.

1. Teoretická část

1.1. Fotopast

V roce 1890 byla vynalezena metoda využití drátku a blesku pro vyfocení divoké zvěře bez fotografa. Ke kameře byl navázán drát, který byl natažen přes část území, kde když přes šňůru přešlo zvíře tak byla pořízena fotografie. Od té doby je princip stejný ale technologie a peníze dovolují fotopastím být stále menší, modernější a dostupnější všem [1] .

Nejčastější využití pro fotopasti je zachycení zvěře ve volné přírodě, ale je možné využít ji pro zabezpečení majetku. Základní princip těchto kamer je, že bývají umístěny na odlehlých místech a pořizují fotografii při detekci pohybu. Fotopasti snímají poté, co pohybový nebo infračervený senzor (PIR) detekuje přítomnost objektu. Některé pasti používají blesk pro přisvětlení v noci, ale to není doporučováno kvůli snadné detekci pasti pachatelem. V případě použití fotopasti při fotografování zvěře by ji blesk plašil. Proto je častěji využíváno InfraRed(IR) kamer, které využívají senzory tepla (PIR). Jestliže senzory zaznamenají teplo, fotopast vyfotí fotografii, ale bez blesku. Nedostatkem infračervených kamer je že záznam je jen černobílý [2] . Záznam je ukládán na paměťovou kartu a majitel musí druhý den zkontrolovat, jaké fotografie byly pořízeny. U dražších modelů je do fotopastí přidáván GSM modul, který umožňuje majiteli zaslání fotografií přes MMS nebo na e-mail.



Obrázek 1 IR Fotopast [3]

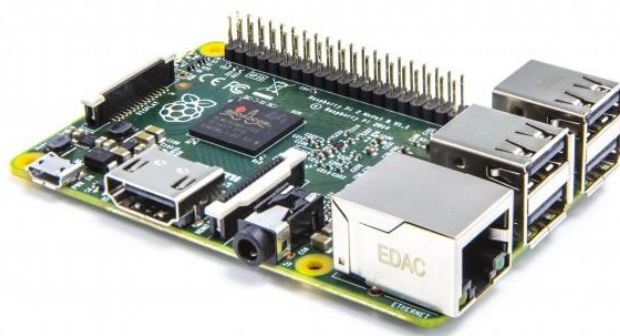
1.2. Seznámení s Raspberry Pi 2

Raspberry Pi je počítač velikosti kreditní karty. Připojením monitoru nebo televize do HDMI portu a použitím klávesnice s myší přes USB porty se dá snadno ovládat. Raspberry je tedy plnohodnotný počítač na rozdíl od Arduino, které je „pouze“ vývojová platforma, a je velmi oblíbený díky své cenové dostupnosti [4] [5] . Raspberry obsahuje i tři základní rozhraní, které z něj dělají nejen hračku, ale hlavně nástroj pro řízení a monitorování.

Tři základní rozhraní:

- **GPIO:** dvě řady pinů na jedné straně. Slouží k připojení externího hardwaru, jako jsou moduly (shieldy), velkou škálu senzorů, expandérů sběrnic a převodníků.
- **CSI camera interface:** slouží pro připojení kamery (Pi Camera) přes rozhraní CSI.
- **DSI display interface:** slouží pro připojení LCD displeje, nejčastěji se připojuje dotykový displej.

Raspberry Pi je často využíván ve školách pro výuku programování v programovacích jazycích Pythonu a Scratch, které jsou na operačním systému Raspbian nainstalovány implicitně. Raspberry je schopno dělat vše co se očekává od stolního počítače. Také umí komunikovat s okolním prostředím, tím je myšleno, že může být použit pro vytvoření různých projektů, například domácí meteorologické stanice, bezpečnostní kamery, NAS serveru, WEB serveru a spousta dalších.



Obrázek 2 Raspberry Pi 2 model B [5]

Hlavní rozdíl mezi Raspberry Pi a Raspberry Pi 2 je v architektuře procesoru. RPi 2 je postavené na ARMv7 architektuře (v dnešní době většina zařízení využívající procesorovou architekturu ARM obsahuje ARMv7 anebo novější) se značně větším výkonem, zatímco Raspberry Pi model B je postaven na ARMv6, která je už dnes výkonově zastaralá. Aktuálně nejnovější je Raspberry Pi 3 model B, který je zpětně kompatibilní s předchozími verzemi.

1.3. Operační systémy

Pro Raspberry Pi existuje mnoho operačních systémů, které jsou z velké části speciálně upravenými Linuxovými distribucemi. Základem těchto systémů je Linuxová distribuce (Debian, Fedora, Ubuntu a další), plus k nim jsou vytvořené speciální softwarové balíčky, které distribuci přizpůsobují pro Raspberry Pi.

Ze seznamu všech podporovaných OS pro Raspberry Pi 2 (dostupné na stránkách raspberrypi.org), byl jako první vybrán Raspbian. Protože je to nejčastěji používaný systém na Raspberry. Je to systém založený na OS Debian wheezy, který byl optimalizován pro Raspberry Pi. Na Raspbianu je defaultně vytvořený uživatelský účet **pi** s heslem **raspberry**. Jelikož je nejčastěji používaný systém, hlavně ve školách, je zde mnoho předinstalovaných výukových programů jako Mathematica, Wolfram, Sonic Pi, Scratch a další. A díky tomu, že je to v podstatě Debian je možné si nainstalovat téměř vše.

2. Hardware

Pro zhotovení fotopasti byl použit tento hardware. Jednodeskový minipočítač Raspberry Pi 2 model 1 GB. Paměťová karta microSDHC značky Kingston o velikosti 8GB s předinstalovaným Raspbianem. Karta slouží jako pevný disk celého počítače a velikost 8GB je minimální, ale v tomto případě zcela dostačující. Pro ukládání fotografií by bylo lepší připojit externí disk a paměťovou kartu nechat pouze pro systém.

Jako napájecí zdroj byl použit 1.2A. Pro komunikaci online byl využit Wi-Fi adaptér od společnosti TP-LINK typ TL-WN725N. Dále bylo k fotopasti přes GPIO piny připojeno, PIR čidlo (infračervený snímač pohybu) typ Modul HC SR501 pro

detekci pohybu. K zachycení fotografií byla použita Raspberry Pi Camera board 5MP REV 1,3, plně kompatibilní s Raspberry Pi.

Náklady na celkovou konstrukci jsou za jednotlivé položky:

Ke dni 15.11.2016	Cena	Zdroje
Raspberry Pi 2	1 029,00 Kč	http://rpishop.cz/raspberry-pi-pocitace/170-raspberry-pi-2-1024-mb-ram.html
Paměťová karta microSDHC 8GB	179,00 Kč	http://rpishop.cz/pametove-karty/465-predinstalovana-karta-kingston-8gb-microsdhc-uhs-i-u1-c10-s-sd-adapterem.html
1,2A napájecí zdroj	185,00 Kč	http://rpishop.cz/raspberry-pi-prislusenstvi/185-12a-napajeci-zdroj-pro-raspberry-pi.html
Wi-Fi adaptér TP-LINK	234,00 Kč	http://rpishop.cz/raspberry-pi-prislusenstvi/30-wifi-adapter-tp-link-tl-wn725n-v2.html
PIR čidlo	159,00 Kč	http://rpishop.cz/raspberry-pi-prislusenstvi/235-pir-pohybovy-senzor.html
Ochranný kryt	71,00 Kč	http://www.ebay.com/itm/Engineering-Case-Enclosure-Computer-Box-for-Raspberry-Pi-3-Model-B-B-Plus-/271664304247?hash=item3f40745077%3A2RcAAOSwyTZUY
Pi Kamera	669,00 Kč	http://rpishop.cz/raspberry-pi-prislusenstvi/22-originalni-kamera-pro-raspberry-pi-camera-board-5mb-rev-13.html
Kryt na kameru	81,00 Kč	http://www.ebay.com/itm/Raspberry-Pi-Camera-Case-Enclosure-Tilting-/122193541144?var=&hash=item1c734d7818:m:mtbeNElnk3myeJISZR-azw
Kryt na PIR čidlo	87,00 Kč	http://www.ebay.com/itm/HC-SR501-Acrylic-Sensor-Enclosure-Box-Mount-Case-Casing-for-Raspberry-Pi-Arduino-/122193541594?var=&hash=item1c734d79da:m:mmK-9IIA8DB01UV--xNRbag
Cena celkem	2 694,00 Kč	

Celková cena je tedy do 2,5 tisíc korun, což je značně méně než běžné fotopasti.

2.1.1. Wi-Fi adaptér

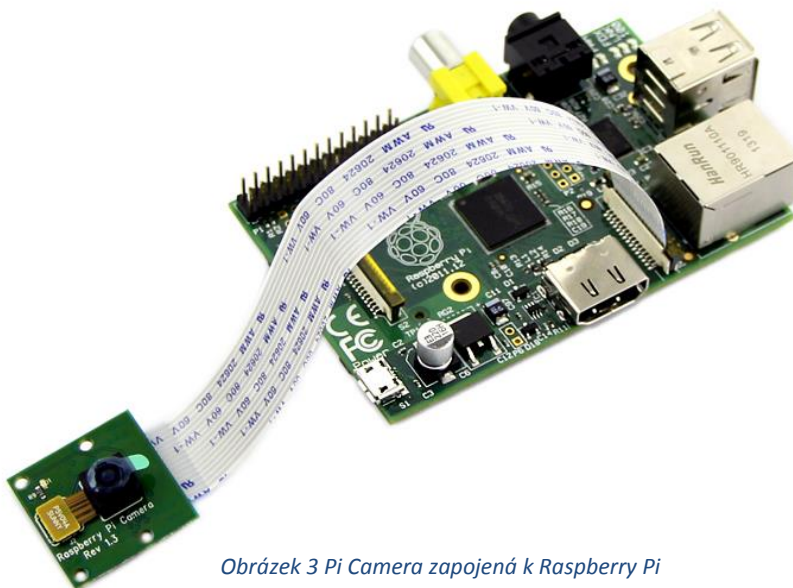
V době přípravy této práce ještě žádné Raspberry nemělo integrovanou síťovou kartu pro bezdrátové připojení, proto Wi-Fi adaptér byl povinnou výbavou pro ty, kteří upřednostňují bezdrátové připojení oproti kabeláži. Nové Raspberry Pi 3 již integrovanou kartu má a nepotřebuje Wi-Fi adaptér. Wi-Fi dongle ušetří prostor i finance, jelikož stojí pár set korun. V této práci byl použit modul od společnosti TP-LINK. Maximální rychlost, které může dosáhnout, je 150Mbit/s. Podporuje zabezpečení WPA2-PSK. Funguje na principu Plug-and-Play, pro Raspberry je zcela dostačující. Výhoda tohoto adaptéru spočívá v jeho velikosti, vůbec nepřekáží, je celkem nenápadný.

2.1.2. Pi Camera

Pi Camera je miniaturní kamera, která byla vytvořena jako rozšiřující prvek pro Raspberry Pi. K RasPi je možné ji připojit pomocí CSI rozhraní do jednoho ze dvou socketů přímo na základní desku. CSI bylo navrženo přímo pro propojení kamer[10]

Pi Camera má rozlišení 5 megapixelů. Pomocí této kamery je možné pořizovat fotografie, videa nebo časové smyčky a používat různé efekty pomocí speciálních knihoven.

Kameru lze ovládat z příkazové řádky nebo přes vytvořený skript v Pythonu, ale existují i další aplikace, programy vytvořené (third-party) vývojáři. Pro Python existuje celá knihovna nazývaná „PiCamera“, pomocí které můžeme vytvořit zajímavé programy využívající Pi Cam. [10] . Z příkazové řádky lze zachytit obraz pomocí příkazu **raspistill** a video pomocí příkazu **raspivid**. Dále lze například vytvořit časovou smyčku.



Obrázek 3 Pi Camera zapojená k Raspberry Pi

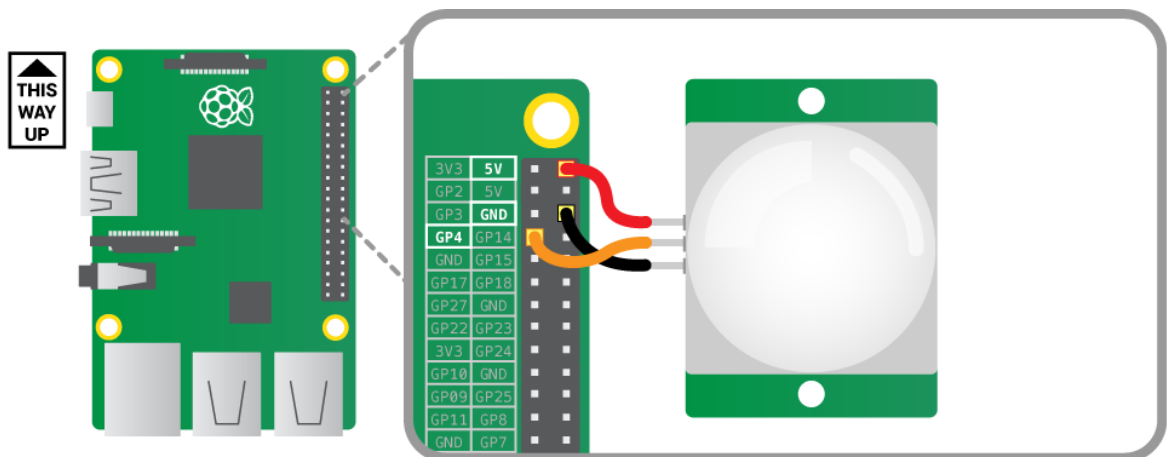
Na trhu lze koupit dva druhy těchto kamer NoIR a IR. Jediný rozdíl mezi nimi je, že NoIR, jak vyplývá z názvu, neobsahuje IR filtr. V praxi to znamená, že fotka pořízená ve dne má horší kvalitu, ale umožňuje pořizovat fotky v noci díky IR přisvětlení [10] .

2.1.3. PIR

Pasivní infračervený senzor funguje na principu pyroelektrického jevu, při kterém se pyroelektrické materiály deformují se změnami teplot. Díky tomuto jevu se na povrchu materiálu indukuje elektrický náboj. Pokud tedy nastane tepelná změna, např. projde člověk, je materiál změnou teploty v části povrchu deformován a je možné detekovat indukovaný náboj na jeho povrchu, tudíž je zaznamenán pohyb. Materiály, které jsou citlivé (měřitelné) v infračervené oblasti se nacházejí v pásmu 4-20 μ m (mikron).

Při použití v praxi se pro projekci okolí na pyroelektrický materiál využívají plastové Fresnelovy čočky, které rozšiřují zorné pole senzoru, ale zároveň chrání senzor před vlivy okolí, jako jsou prach a vlhkost[17].

Zde je znázorněno zapojení PIR senzoru přes GPIO piny k Raspberry Pi 2.



Obrázek 4 Zapojení PIR senzoru

PIR senzor je třeba připojit třemi kabely, na tomto obrázku je červený kabel připojen k napájení 5V. Oranžový kabel je připojen k GP4, což je pin číslo 4 na Raspberry Pi. A poslední černý kabel je připojen k zemi GND, nebo-li je uzemněn.

2.2. Software

2.2.1. Python

V této bakalářské práci byl pro psaní skriptů použit programovací jazyk Python verze 2.7.3 (defaultní verze) a několik jeho knihoven. Jelikož je to starší verze, syntaxe není stejná jako například ve verzi 3.5. Důležitou knihovnou je email.MIME, která umožňuje odesílání emailů s přílohou.

Email = tento balíček není sám o sobě navržen pro posílání emailů přes SMTP, či jiných serverů, tyto funkce jsou obsaženy v balíčku „smtplib“

MIME = **Multipurpose internet mail extensions**. Specifikace pro formát, který dovoluje posílat spolu s emailem přílohy (audio, videa, obrázky a další) přes internet.

2.2.2. Pushetta

Pushetta je systém, který dopraví notifikace v reálném čase na několik různých zařízení (mobilní telefony, prohlížeče, chytré TV atd.). Pushetta se snaží vyřešit problém, jak poskytnout informaci v reálném čase bez využití zastaralých metod jako email [12].

Princip této aplikace spočívá ve vytvoření tématické skupiny např. „Raspberry DoorBell“. Každý uživatel odebírající tuto skupinu, tzv. subscribers, obdrží notifikaci na své zařízení (SmartPhone, Tablet) pokaždé když rozešlete zprávu. K odeslání notifikace při sledované události, musí být provedena autentizace mezi registrovaným uživatelem a kódem aplikace v zařízení.

2.2.3. Weaved

Weaved je software umožňující ovládat síťové zařízení pomocí internetu odkudkoli. Není třeba žádného port forwardingu, či komplexního nastavení sítě ani žádné hardwarové řešení, je to pouze softwarové řešení. Pracuje s populárními protokoly a službami jako jsou SSH, VNC, RDP, Rsync. Také komunikuje s jakýmkoliv zařízením, jež běží přes TCP/IP protokol.

Weaved poskytuje přímé bezpečné tunelové spojení, které běží na TCP/IP. Díky tomu lze přistupovat ke každému zařízení, které je za NAT nebo firewallem a vzdálená administrace se chová jako práce přímo na zařízení.[18]

2.2.4. Motion

Motion je program, který detekuje pohyb. Program monitoruje video signál z jedné nebo více kamer a je schopný detekovat, zda se změnila značná část obrázku [14] .

Program porovnává jednotlivé snímky, a pokud mezi nimi zaznamená rozdíl překračující stanovenou mez, začne zaznamenávat.

Celý program je napsán v jazyku C a je určen pro operační systém Linux (využívá „video4linux“ prostředí). Výstupy tohoto programu jsou buď jpeg obrázky, ppm soubory nebo mpeg video sekvence. Motion nám umožňuje i živé streamování videa. Tento program umožňuje použití i běžné usb webkamery místo Pi Kamery[15] .

3. Metodika

Při tvorbě bakalářské práce „Konstrukce fotopasti pomocí Raspberry Pi“ byl použit stejný hardware pro více operačních systémů. Všechny skripty pro pořízení fotografie, odesílání notifikace, e-mailů a dalších jsou napsané v programovacím jazyce Python verze 2.7.3.

Na začátku práce bylo stanoveno, že celá fotopast bude fungovat na hardwaru Raspberry Pi. V práci bude použita PiCamera, která je vytvořena speciálně pro RPi, z důvodů vyššího rozlišení oproti levným usb webkamerám. Dále bylo třeba stanovit, jak bude samotná detekce fungovat. Zaznamenávání bylo nejdříve zkoušeno pomocí programu **Motion**, který je velmi často využíván, existují k němu i naprogramovaná webová rozhraní. Z důvodů využití GPIO pinů bylo nakonec rozhodnuto, že pohyb bude zaznamenán pomocí infračerveného senzoru (PIR).

Tento způsob byl preferován i proto, že profesionální fotopasti také využívají PIR senzor a program motion výrazně zpomaloval chod RPI a nefungoval vždy správně. Tento fakt by mohl negativně ovlivňovat funkci fotopasti.

Dále přišla na řadu volba operačních systémů. Raspbian, díky tomu že byl předinstalovaný na sd kartě a nejvíce využíván při práci na Raspberry Pi byl při

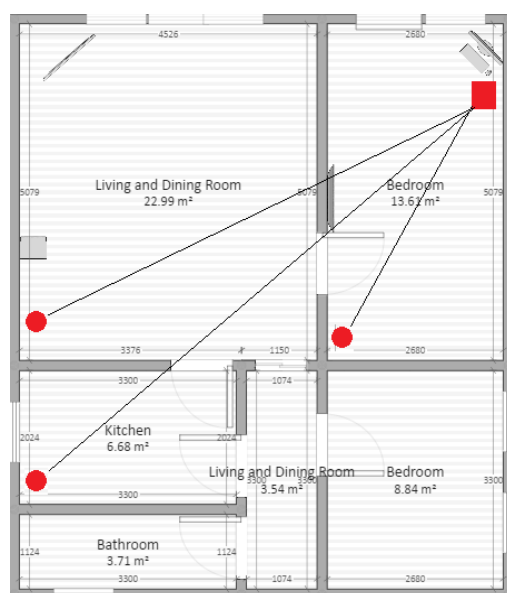
volbě OS ve výhodě. Raspbian je jako jediný z testovaných systémů oficiálně podporovaný systém Raspberry Foundation. Dále byl testován Ubuntu Mate, od verze 16.04 dostupný pro Raspberry Pi a v současnosti dostupný i pro RPi 3.

Nakonec byla testována Fedora konkrétně její remix FedBerry projekt, který je speciálně určen pro Raspberry Pi. Pro Raspberry Pi 1 byla upravená verze nazývaná Pidora, ale i ta již není kompatibilní s RPi 2 a 3.

Verze operačních systémů, které byly použity:

- Raspbian GNU/Linux 7 Wheezy s jádrem Linux 4.1.10-v7+armv7l.
- Ubuntu MATE 16.04 s jádrem Linux 4.8.8
- Fedberry 23 remix s jádrem 4.4.2

Na obrázku níže jsou vyznačeny body kde a jak daleko byla fotopast od AP.



Obrázek 5 Půdorys objektu

Testování systémů probíhalo vždy stejně. Začínalo se testovat od nejmenší vzdálenosti po největší možnou vzdálenost, tím je myšleno kam až dosáhl signál Wi-Fi sítě. Testování probíhalo i venku, bohužel zde je testování ovlivněno počasím. Tudíž bylo prováděno pouze za slunečného počasí. Pro testování venku je třeba řešit problematiku napájení Raspberry Pi a zároveň přístup k internetu. Toto jsem vyřešil použitím outdoorovým FTP kabelem a napájením pomocí powerbanky. Napájet se Raspberry dá i pomocí AA baterií, když je dostupný potřebný hardware. Bohužel v tomto se Raspberry Pi nemůže vyrovnat komerčním fotopastím. Ty dokážou běžet i několik týdnů nebo měsíců.

Konstrukce mnou vyrobené fotopasti má výhodu v tom, že si ji může kdokoliv upravit podle sebe. Jak po hardwarové nebo softwarové stránce. Ale komerční si upravit nejde.

4. Praktická část

V této části bakalářské práce se budu zabývat samostatným řešením zkompletování fotopasti a následným otestováním funkčnosti na různých operačních systémech platformy Linux. Celá fotopast byla postavena z výše uvedeného hardwaru. Na závěr vyberu z mého pohledu nejvhodnější operační systém pro konstrukci a ovládání Fotopasti na Raspberry Pi 2 B.

Pro funkčnost fotopasti byly vytvořeny dva skripty (viz.str.51), které jsou napsány v programovacím jazyce Python. Protože je většinou na všech operačních systémech defaultně po instalaci systému. A také proto, aby mohly být použity stejné skripty na všech testovaných operačních systémech.

První a hlavní skript je pro samotnou detekci a zachycení pohybu přes PIR senzor a Pi Kameru. Druhý slouží pro odeslání emailu s aktuálně vyfocenou fotografií. Oba jsou propojeny, první volá funkci pro odesílání emailu, které je jako atribut přidána cesta k právě vytvořenému souboru.

Abychom mohli používat Pi kameru, je třeba jí nejprve povolit v konfiguračním souboru každého systému. V každém se povoluje jinak, někde to lze povolit v grafické podobě jinde zase pouze v terminálu. Pro pokročilé ovládání Pi kamery, například na základě nějaké události, musíme vždy využívat python knihovnu **picamera**. Tu je potřeba nainstalovat na každém systému, bez ní lze ovládat kameru pouze na příkazové řádce.

Zde jsou základní příkazy v terminálu pro pořízení fotografie, videa.

```
$ raspitill -o image.jpg #vyfocení fotografie s názvem image ve formátu jpeg (-o -filename)
$ raspivid -o vid.h264 #natočí 5s video formátu h264, který lze přehrát ve VLC, délka videa je defaultní
$ raspivid -o vid.h264 -t 10000 #znakem -t se nastavuje delší čas videa v milisekundách
```


Hlavní skript, musíme spouštět s právy „root“, buď za použití příkazu sudo a nebo se přihlásíme jako root uživatel, protože přistupuje k GPIO pinům a k nim běžný uživatel nemá přístup. Skripty by bylo možné sloučit do jednoho, ale pro lepší přehlednost kódu byly ponechány zvlášť.

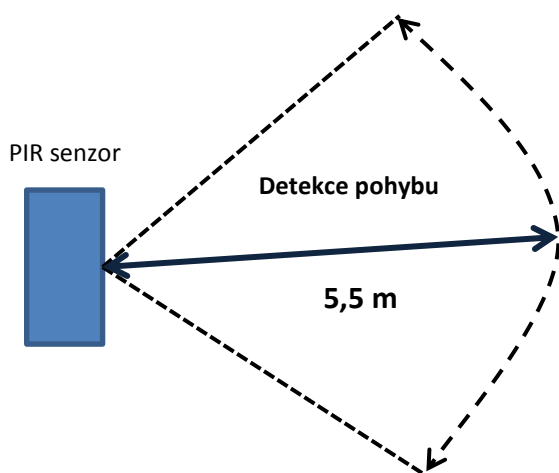
Je důležité, aby bylo Raspberry neustále připojeno k internetu ať pomocí ethernetového kabelu nebo přes bezdrátovou síť, protože bez internetu nebude moci odeslat fotografie na email. I když skript pro odesílání je ochráněn výjimkou **try-except**, proti případnému odpojení od internetu, aby skript nespadol a fotopast přestala fotit. Fotopast si po dobu výpadku internetu fotografie ukládá do vnitřní paměti. Při opětovném připojení k internetu se fotografie začnou opět zasílat na email. Fotografie, které byly pořízeny během doby odpojení od internetu se na email neodešlou, zůstanou na vnitřním úložišti či usb disku.

Detekce pohybu je zjišťována pomocí PIR čidla, které je připojeno na GPIO piny Raspberry Pi.

GPIO = Raspberry Pi 2 má celkem 40 pinů, 26 z nich jsou čistě GPIO a ostatní jsou pro napájení (3.3V a 5V), uzemnění (GROUND), plus dva ID EEPROM. V práci je použit PIR senzor, který jako vstupní data zachytává pohyb. A na výstup pošle signál pro vyfocení fotografie a následné odeslání na email. Nevýhoda těchto pinů je v tom, že při neopatrnosti se může celé Raspberry zničit.

PIR = Jediná nevýhoda těchto senzorů je, že špatně fungují nebo ukazují špatné hodnoty při velkých teplotách, to bohužel ani Fresnelovy čočky neochrání. PIR senzory se využívají velmi často při zabezpečení domácnosti, různých soukromých

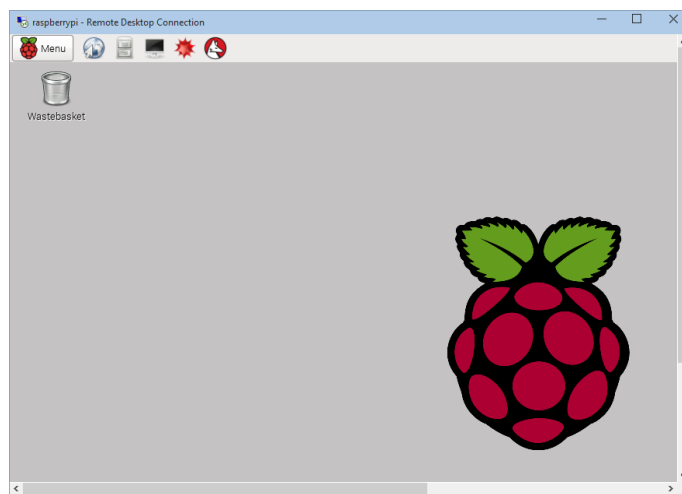
objektů. Nejčastěji se umísťují do horního rohu místnosti, aby pokryly co největší část pokoje. U PIR čidla jsou dva trimery pomocí kterých se nastavuje citlivost senzoru a doba odezvy. Je třeba si vyzkoušet, jak moc citlivý bude senzor a jak velká bude odezva. Senzor rozezná pohyb do vzdálenosti 5,5 metrů a v úhlu téměř 180° . Bez nasazené čočky vzdálenost upadá téměř o dva metry a i pozorovací úhel je menší.



Obrázek 6 Schéma zabírajícího úhlu

4.1. Raspbian

Raspbian je open-source operační systém postaven na Linuxové distribuci Debian a optimalizován pro Raspberry Pi. Raspbian byl vytvořen malým specializovaným týmem vývojářů, kteří podporují cíle Raspberry Pi Foundation, tak i hardware Raspberry Pi. Raspbian obsahuje více než 35.000 balíčků a před-kompilovaných software svázaný v pěkném formátu pro snadnou instalaci na vaše Raspberry Pi [6]. Raspbian je nejčastěji doporučováno pro začátečníky a obsahuje grafické prostředí zvané LXDE, ale lze jej ovládat i pouze z příkazové řádky. Nejnovější verze je Raspbian Jessie [6].



Obrázek 7 GNOME Grafické prostředí Raspbian wheezy

Jako první systém byl použit Raspbian verze 7 typ Wheezy. Raspbian je uživatelsky velmi přívětivý a je postavený na distribuci Debian, příkazy jsou stejné jako v Debianu. Instalace systému proběhla přes program NOOBS, který je vytvořený pro instalaci všech podporovaných operačních systémů na oficiálních stránkách. Na instalaci Raspbianu je doporučováno mít minimálně 8GB microSDHC kartu třídy 10 (lze i 6). Pomocí NOOBS si lze na kartu nainstalovat i více operačních systémů, ale některé již nejsou kompatibilní pro Raspberry PI 2 a 3, jako ARCH X a Pidora. Na Raspbianu je vytvořený předem definovaný uživatelský účet, ***uživ_jméno = pi*** a ***heslo = raspberry***.

Pro zhotovení práce bylo nutné nainstalovat knihovny programovacího jazyka Python verzi 2.7.3. Také byla nainstalována pythonovská knihovna pro Pi kameru.

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

Pro přístup k Raspberry pomocí internetu a ne jenom v lokální síti, je nutno si nainstalovat aplikaci **weaved**, která má dostupnou dokumentaci přímo pro použití na Raspberry. Zde bylo nastaveno, že komunikace bude probíhat přes **SSH port 22**, lze však komunikovat i přes HTTP (80), VNC (5901) a vlastní TCP spojení. Tato aplikace nahrazuje **port-forwarding**. Zároveň je to rychlejší a jednodušší, neboť si stačí vytvořit účet a na Raspberry si jen vybrat přes jakou službu budete komunikovat a ta se trvale uloží. Navíc si lze přidat do **crontab** příkaz, který spustí tuto aplikaci při startu RasPi.

```
$ @reboot /usr/bin/weavedstart.sh
```

Dále je použita aplikace Pushetta pro posílání notifikací na mobilní telefon.

Pro používání kamery jí bylo třeba nejdřív povolit v konfiguračním souboru.

```
$ sudo raspi-config
```

Tato konfigurace se otevře i po prvním startu, ale tímto příkazem ji můžeme opětovně vyvolat a povolit kameru.

Oba skripty byly vytvořeny a poprvé vyzkoušeny na Raspbianu a po té použity i na ostatních systémech.

Raspberry pořizuje fotky každých 6 sekund při nepřetržitém provozu fotopasti v rušném prostředí bez problému. Fotografie se ukládají do složky na vnitřní úložiště (micro SDHC), nebo si lze připojit USB flash disk a fotografie ukládat přímo na flash disk.

Během odesílání emailů s aktuálně vyfocenou a přiloženou fotografií nebyl zaznamenán žádný problém, pokud je kvalitní signál Wi-Fi sítě. Samostatný email se odešle, po vyfocení fotografie s 3 sekundovým zpožděním. Pro odesílání byl vytvořen emailový účet na **google.com** a posílání probíhá pomocí **smtp serveru**.

Pro připojení flash disku, si nejdříve musíme najít kde je připojen a jak se jmenuje. Také lze použít druhý příkaz a případně tak zjistit proč se nepřipojil.

```
$ sudo fdisk -l  
$ dmesg | tail
```

Většinou bývá označen jako **sda,sda1,sdb** atd. Také lze použít **dmesg** a případně tak zjistit, proč se disk nepřipojil.

Následně si vytvoříme složku, kam ho budeme chtít připojit, v tomto případě je to složka s fotografiemi. Po vytvoření složky, disk připojíme a můžeme používat.

```
$ sudo mkdir /Detector/photo
$ sudo mount /dev/sda1 /Detector/photo
```

Až budeme chtít disk vyjmout tak ho odpojíme. Všechny příkazy je nutné spouštět jako superuživatel.

```
$ sudo umount /Detector/photo
```

Ukázka samotných skriptů

Aby bylo možné posílat email, je třeba si nainportovat potřebné knihovny, bez kterých by to nešlo.

```
import os
import smtplib
from datetime import datetime
from email.MIMEText import MIMEText
from email.MIMEImage import MIMEImage
from email.MIMEMultipart import MIMEMultipart
```

Další důležitá část je nastavit si proměnné kdo bude odesílatel, kdo příjemce. Je dobré si to dát jako proměnné globální, pro ušetření času a vyvarování se případných chyb. Heslo k emailovému účtu je zde v otevřené podobě, každý ho může vidět. Proto je dobré mít celý skript ochráněn, například změnit práva aby ho mohl spouštět a číst pouze superuživatel

```
frm = 'czechbery@gmail.com'
to = 'danielswab@gmail.com'
psw = '*****'
```

Následuje definování hlavní metody. Tato metoda je volána v hlavním skriptu a jako atribut je použit název a cesta k aktuálně vyfocenému souboru. Zároveň je zde vytvořena hlavička emailu.

```
def SendMail(ImgFileName):
    msg = MIMEMultipart()
    msg['Subject'] = 'Motion'
    msg['From'] = frm
    msg['To'] = to
```

Následuje tělo emailu, které obsahuje text s časovým údajem, kdy byl mail odeslán.

```
body = 'Alert!! Motion Detected!!\nTime: %s' % str(datetime.now())
msg.attach(MIMEText(body, 'plain'))
```

Pro přílohu fotografie, je využita knihovna MIMEImage, ta slouží k připojení fotografie jako přílohy.

```
fp = open(ImgFileName, 'rb')
msgImage = MIMEImage(fp.read())
fp.close()
msg.attach(msgImage)
```

Jako poslední je navázáno spojení se smtp serverem, kterému je definováno přes jaký server a port bude probíhat komunikace. Komunikace probíhá šifrovaně přes TLS. Zde je vidět použití globálních proměnných

```
s = smtplib.SMTP('smtp.gmail.com', 587)
s.ehlo()
s.starttls()
s.ehlo()
s.login(frm, psw)
s.sendmail(frm, to, msg.as_string())
s.quit()
```

Pro spuštění celé fotopasti po startu Raspbianu, poslouží skript napsaný v bashi a program crontab, kde je volán bash skript při zapnutí nebo restartování Raspberry.

V hlavním skriptu jsou naimportovány důležité knihovny například pro ovládání GPIO pinů, Pi Kamery nebo aplikace Pushetta.

```
import RPi.GPIO as GPIO
import time
import sys
import traceback
from pokus import SendMail
import picamera
import datetime
from pushetta import Pushetta
```

Ještě před definováním hlavní metody, si vytvořím metodu, která je použita v **while** cyklu. Udává, jak se bude vyfocená fotografie jmenovat a kam se bude ukládat.

```
def get_file_name():
    return datetime.datetime.now().strftime("/home/pi/Detector/photo/%Y-%m-%d_%H.%M.%S.jpg")
```

V hlavní metodě je použit **try-except** blok, pro přerušení klávesovou zkratkou **Ctrl+C** a pro všechny výjimky.

```
def main():
    try:
    ...
    ...
except KeyboardInterrupt:
    print " System is terminated"
except Exception:
    print "Nastal Error"
```

Jelikož je pro detekci pohybu použit PIR senzor, musí se určit na jakém portu je připojen. Zvolím port číslo 4. Je možné si připojit jakýkoli port GPIO, který je označen jako „**general purpose I/O**“ pin (GPIO). Není snadný si zapamatovat který pin je který, tak je lepší se podívat na nějaký návod. Aby bylo možné se odkazovat na vstupní pin, je třeba ho nějak pojmenovat. Po té nastavíme číslování GPIO pinů. Jsou dva druhy buď **BOARD** nebo **BCM**. Dále definujeme pojmenovaný pin jako vstupní pin.

```
sensor = 4
GPIO.setmode(GPIO.BCM)
GPIO.setup(sensor, GPIO.IN, GPIO.PUD_DOWN)
```

Zde jsou předdefinová dvě proměnné nazvané předchozí stav a současný stav. Obě mají hodnotu nastavenou na **False**, kvůli použití ve **while** cyklu. Také je zde vytvořená proměnná **cam** pro zkrácení zápisu.

```
previous_state = False
current_state = False
cam = picamera.PiCamera()
```

Tato část kódu je pro samotnou Pushettu, tento kód je k dispozici na stránkách této aplikace.

```
API_KEY="yourTOKEN"
CHANNEL_NAME="CzechBery"
p=Pushetta(API_KEY)
print "...Camera trap is ready... "
```

Celý cyklus obsahuje několik podmínek typu **if**. Hlavní použití je pro zachycení změny stavu na senzoru, kdy se porovnávají dva stavy kde se jeden přiřadí k senzoru a jestliže se nerovnají vznikne nový stav. Neboli je detekován pohyb a následně pořízení fotografie.

```

while True:
    time.sleep(0.1)
    previous_state = current_state
    current_state = GPIO.input(sensor)
    if current_state != previous_state:
        new_state = "HIGH" if current_state else "LOW"
        print("GPIO pin %s is %s" % (sensor, new_state))

```

Jakmile jsou splněny podmínky a současný stav dostal novou hodnotu, začne se kamera tzv. „zahřívat“ před vyfocením. Zároveň se odešle notifikace na mobilní telefon. Po té kamera počká 2 s a vyfotí fotografii. Je důležité aby kamera počkala alespoň 2 sekundy než začne fotit, kvůli nastavení úrovně světla. Po 1 se spustí metoda z druhého skriptu, která odešle email s aktuálně vyfocenou fotografií.

```

if current_state:
    fileName = get_file_name()
    cam.start_preview()
    p.pushMessage(CHANNEL_NAME, "Motion Detected... Look at mail!")
    time.sleep(2)
    cam.capture(fileName)
    time.sleep(1)
    SendMail(fileName)
else:
    cam.stop_preview()

```

Toto jsou již výše zmíněné zachycené výjimky pro přerušení klávesovou kombinací **Ctrl+C** a pro ostatní neočekávané události. Toto je syntaxe použitá v Pythonu 2.7.3.

```

except KeyboardInterrupt:
    print " System is terminated"
except Exception:
    print "Nastal Error"

if __name__ == "__main__":
    main()

```

Základní bash skript je o pár řádcích.

```

cd /
cd /home/pi/Detector/
sudo python nazev_skriptu.py
cd /

```

Poté je spuštěn crontab.

```
sudo nano crontab -e
```

A přidám do něj na konec tuto řádku.

```
@reboot /home/pi/bash_skript.sh
```

Pro zrušení spuštění při startu, stačí napsat před tuto řádku „#“.

Je třeba, aby Raspberry bylo neustále připojeno k internetu, jinak by nastala chyba a skript by se ukončil. Pro ukončení skriptu spuštěný po startu, je třeba si vyhledat jeho PID pomocí.

```
ps aux | grep python
```

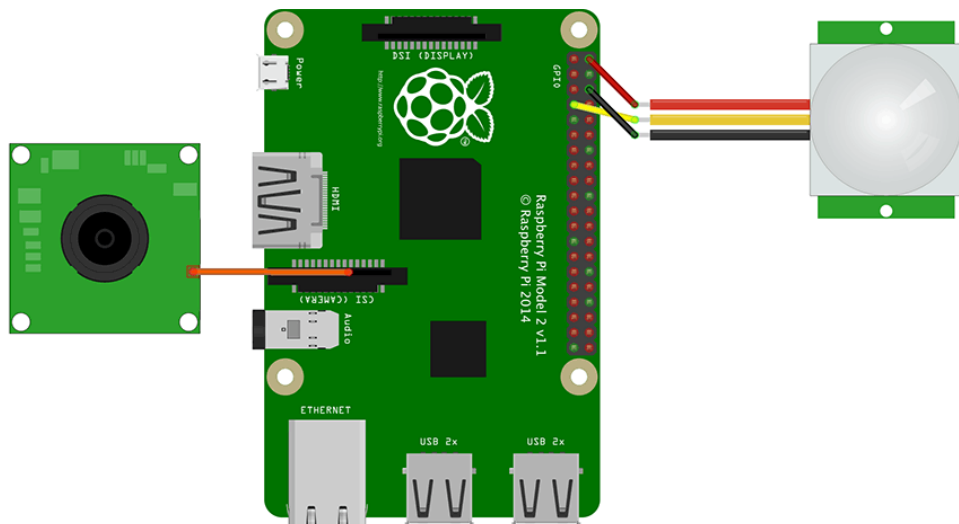
A následně „zabít“ skript.

```
sudo kill pid
```

Pro spuštění celé fotopasti stačí spustit hlavní skript. Po zapnutí už vidíme, zda byl zaznamenán pohyb nebo ne.

```
Sudo python pirCam.py  
...Camera trap is ready...  
GPIO pin 4 is HIGH  
GPIO pin 4 is LOW  
GPIO pin 4 is HIGH  
GPIO pin 4 is LOW
```

Kdykoliv je pohyb zaznamenán (hodnota **HIGH**), je odeslán email s přiloženou fotografií a zároveň je na mobil odeslána notifikace přes aplikaci Pushetta. Notifikace chodí rychle, když je uživatel připojen přímo v lokální síti, ale když je uživatel připojen například na 4G od poskytovatele je ten čas odezvy mnohem větší. Takto vypadá schéma zapojení fotopasti. Toto schéma bylo použito na všech testovaných systémech.



fritzing

Obrázek 8 Schéma zapojení [40]

4.1.1. Testování

Během testování se zaměřím hlavně na použití wi-fi donglu značky TP-LINK typu TL-WN725N, který je schopen dosáhnout max rychlosti 150Mbps a pracuje ve frekvenci 2,4-2,4835 GHz.

Při začátku testování byl problém, že se ne vždy Raspberry připojilo k bezdrátové síti. Pro vyřešení tohoto problému, byl použit wpa_supplicant, který jsem upravil tak, aby Raspberry rozpoznalo ke které síti se má automaticky při startu připojit. Výhoda tohoto řešení je, že si můžu zadat několik Wi-Fi sítí, ke kterým bych se mohl připojit a Raspberry si při startu zjistí v jaké síti je.

```
GNU nano 2.2.6 File: /etc/network/interfaces
auto lo

iface lo inet loopback
iface eth0 inet dhcp

allow-hotplug wlan0
iface wlan0 inet manual
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf

iface Doma inet dhcp
iface MojAP inet dhcp
```

Obrázek 9 Interfaces

WPA_Supplicant je program, který zajišťuje podporu WPA zabezpečení bezdrátových sítí. Má plnou podporu WPA2 tak i starších bezdrátových bezpečnostních protokolů.

```
GNU nano 2.2.6 File: /etc/wpa_supplicant/wpa_supplicant.conf
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
    ssid="Doma"
    psk="mojeheslo"
    proto=WPA
    scan_ssid=1
    id_str="Doma"
    priority=5
    key_mgmt=WPA-PSK
    pairwise=CCMP
    auth_alg=OPEN
}

network={
    ssid="MojAP"
    psk="mojeheslo"
    proto=RSN
    scan_ssid=1
    id_str="MojAP"
    priority=5
    key_mgmt=WPA-PSK
    pairwise=CCMP
    auth_alg=OPEN
}
```

Obrázek 10 Wpa_Supplicant

Bezdrátové sítě, ke kterým se chceme připojit, se nastavují v konfiguračním souboru `/etc/wpa_supplicant/wpa_supplicant.conf`. Je možno mít několik Wi-Fi sítí definovaných v konfiguračním souboru, program se připojí k aktuálně dostupné síti a té která má největší prioritu (viz. obr 9) [25].

V raspbianu je `wpa_supplicant.conf` navázaný na `/etc/network/interfaces`. V *interfaces* jsou obsaženy doposud známé informace o síťových rozhraních. Přidáním řádku `/etc/wpa_supplicant/wpa_supplicant.conf` říkáme Raspberry, že má povolit WLAN jako metodu síťového připojení a je použita konfigurace z `wpa_supplicant.conf`. Tato konfigurace bude přidělovat Raspberry dynamickou IP adresu [26].

Pro nastavení statické adresy je třeba v `/etc/network/interfaces` přidat tyto atributy: **adresa, adresa sítě, síťovou masku, bránu, broadcast.**

```
$ iface doma_static inet static
    address 192.168.1.15
    netmask 255.255.255.0
    network 192.168.1.0

    gateway 192.168.1.1
    broadcast 192.168.1.255
```

V konfiguraci `/etc/network/interfaces` je použito **wpa-roam** místo **wpa-conf** [27].

Wpa-roaming je metoda s kterou můžeme prohlížet a připojovat se k bezdrátovým sítím. Výsledek tohoto použití je že pokud není připojen ethernetový kabel, tak wlan0(Wi-Fi dongl) dostane přednost a připojí se k námi definované Wi-Fi síti, nebo k otevřené bezdrátové síti. Když připojíme síťový kabel tak se Wi-Fi přístup ihned ukončí a přednost dostane eth0 (ethernet) a opět při vypojení eth0 se Wi-Fi přístup nastartuje [26].

Wpa-conf je statická konfigurace, která nepotřebuje prohledávat síťové rozhraní. Wpa-conf je řešení problému, kdy při připojení ethernetového kabelu nejde Wi-Fi síť.

Pro měření síly signálu a kvality spojení byl používán příkaz `iwconfig wlan0` a program `wavenom`.

Při vzdálenosti 5m od aktivního prvku byla síla signálu:

```
pi@raspberrypi ~ $ iwconfig wlan0 | grep -i --color quality
    Link Quality=96/100   Signal level=66/100   Noise level=0/100
pi@raspberrypi ~ $
```

Hodnota 66/100 je víc než dostačující, při takovémto signálu by neměl nastat žádný problém s odesláním. Všechny byly doručeny na emailový účet. Nedošlo k žádnému výpadku, popřípadě komplikacím, tím pádem výsledek je 100%.

Na vzdálenost cca 10m byla síla signálu okolo 40% (16-18dBm) a kvalita okolo 60/100 (+/-).

```
pi@raspberrypi ~ $ iwconfig wlan0 | grep -i --color signal
Link Quality=73/100 Signal level=38/100 Noise level=0/100
pi@raspberrypi ~ $
```

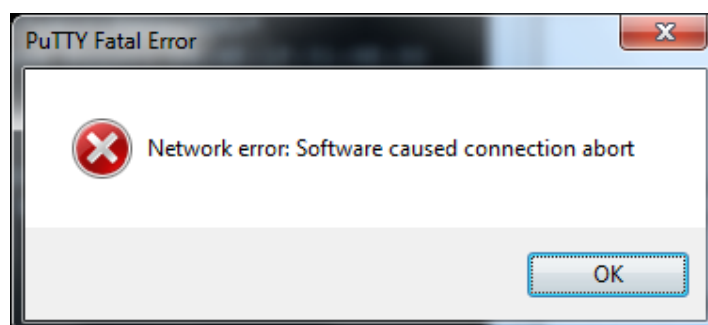
Raspberry bylo odděleno jednou zdí od aktivního prvku, ale stále ani jednou nevypadlo spojení, vše probíhalo bez problémů. Při této vzdálenosti byla fotopast stále spolehlivá.

Síla signálu 50%, při vzdálenosti cca 15m.

```
pi@raspberrypi ~ $ iwconfig wlan0 | grep -i --color signal
Link Quality=89/100 Signal level=52/100 Noise level=0/100
pi@raspberrypi ~ $
```

Při úrovni 50% signálu probíhalo vše tak jak má. Raspberry reagovalo svižně na příkazy a fotopast ze začátku běžela a fotila pravidelně. Problém nastal až po 2 h a 15m běhu. Celé 2 hodiny byla fotopast vystavena normálnímu prostředí (nebyl zde po celou dobu testování nepřetržitý pohyb, mezi fotkami jsou různé časové mezery, někdy byl pohyb s většími pauzy někdy byl nepřetržitý). Během poslední hodiny se pohyb ustálil téměř na nulový. Poslední zaznamenaný pohyb byl dle fotky ve 2:14 a zbylých 45 minut již fotopast byla mimo provoz. Výpadek byl zjištěn až po celých 3 hodinách. Vina byla na straně síly signálu a nestabilitě připojení. Při opětovném spuštění Raspberry na stejném místě, spadl signál z 50% na 20%, při kterých už je Raspberry nespolehlivé. Všechny fotky, které byly pořízeny do doby výpadku se v pořádku odeslaly na email.

Při vzdálenosti cca 15m byla síla signálu podstatně nižší, okolo 20% a začínaly se vyskytovat problémy. Fotopast byla oddělena dvěma zdmi a špatně přijímala signál Wi-Fi sítě.



Obrázek 11 Putty Error

Hlavní problém byl, že se s Raspberry nedalo při této úrovni signálu téměř vůbec pracovat. Protože při vzdáleném připojení přes SSH pomocí programu **PuTTY**, se ukončilo spojení, jakmile jsem se přihlásil a chtěl spustit fotopast. Dvě nejčastější chyby, které se stávaly byly **Network error: Software caused connecion abort** a **Network error: Connection timed out**.

Oboje byl následek slabé síly signálu. Tudíž 20% se dá považovat za nedostatečnou úroveň signálu a fotopast byla při této úrovni nespolehlivá.

Když je Raspberry připojeno pomocí ethernetového kabelu, není žádný problém s výpadky sítě. Pro vyzkoušení zda je síťový kabel opravdu spolehlivý byla fotopast připojena k internetu přes 15m FTP kabel. Jelikož během doby co byla připojena a spuštěna nenastal ani jediný výpadek, více testování neproběhlo i z důvodů že delší než 15m kabel nebyl k dispozici.

Venkovní testování.

I když je tato Fotopast určena spíše do vnitřních prostorů, otestoval jsem její funkčnosti i venku. Pro použití jako venkovní fotopast, by bylo třeba zajistit voděodolný kryt, aby Raspberry nenamoklo. Dále také vyřešit problém s připojením, já jsem jí testoval na rohu baráku, kam mi dosáhl outdoorový FTP kabel. Připojení přes bezdrátovou síť by zde nefungovalo tak dobře, protože venku mám slabý signál Wi-Fi i na mobilu a bezdrátový adaptér, který byl používán po celou dobu této práce není moc kvalitní. Pro testování venku je třeba mít nějaké přenosné napájení, často se používá napájení pomocí například šesti AA baterií. Já sem zkusil napájení pomocí powerbanky. V dnešní době je powerbanka celkem dost rozšířená, takže není problém jí sehnat. Jediná nevýhoda je, že levnější powerbanky mají malou kapacitu **mAh**. Moje má 2600mAh, to vystačí zhruba na 2x nabití telefonu. Jako výstupní napětí je schopná dodávat 5V-800mA. Raspberry model B využívá okolo 700-1000mA, záleží, jaké další periférie jsou připojeny. Kamerový modul vyžaduje 250mA a GPIO piny mohou čerpat 50mA což je rozděleno do všech pinů. Nejdříve jsem vyzkoušel, jak dlouho vydrží Raspberry běžet na této powerbance. Při zapnuté fotopasti, kdy Rpi neustále fotilo a odesílalo fotografie, vydrželo Raspberry běžet 3,5h. Během té doby nedošlo ani jednou k pádu či vypnutí systému z nedostatku dodávaného napájení. Bohužel 3,5 hodiny je krátká doba, to je jediná nevýhoda slabých powerbank. Na provoz například alespoň 12 h by to chtělo

mnohem silnější powerbanku. Ale napájení pomocí šesti AA baterií prý dokáže běžet v plném provozu přes 13 h. "

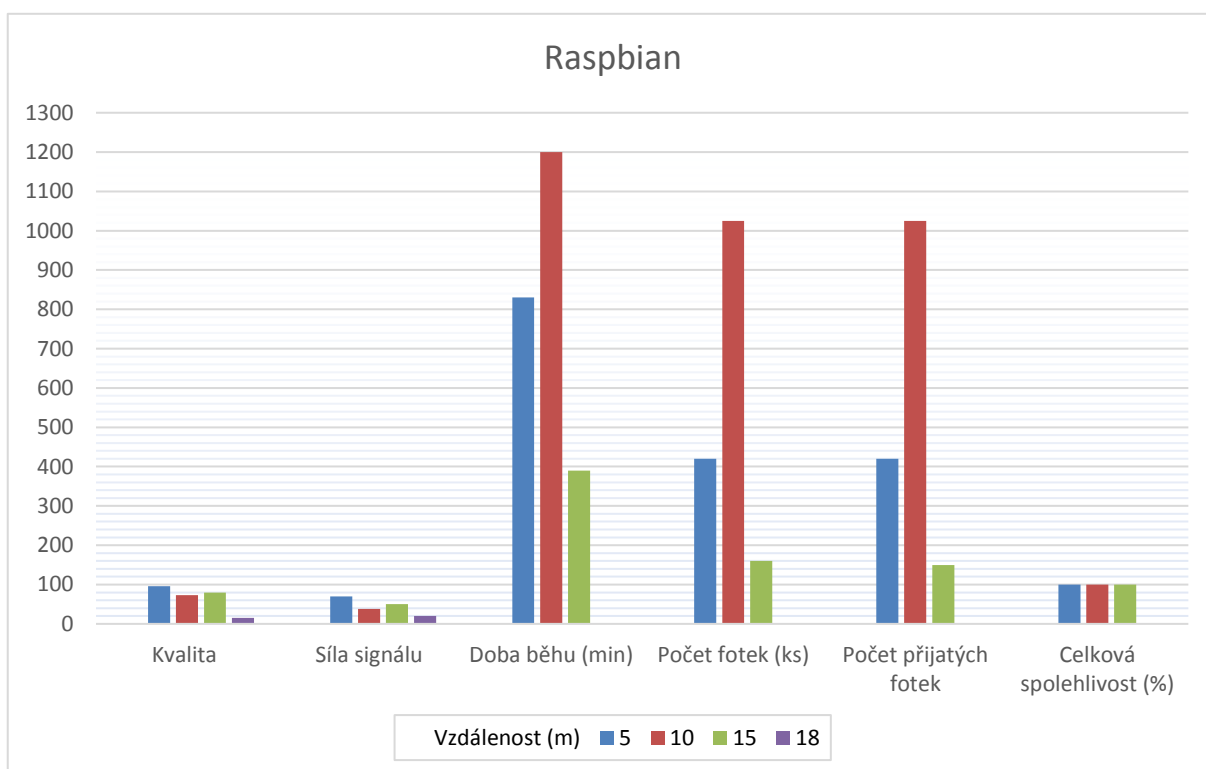
Během pár hodin testování venku bylo zjištěno, že PIR senzor má větší dosah než v uzavřené místnosti cca 10m. Samotné fotky byly s postupem času horší kvality, čím méně slunce bylo tím byly horší. PIR senzor byl také ovlivňován větrem, když foukal hodně čidlo se hýbalo a bralo to jako pohyb. Navíc kamera nestíhá vyfotit jedoucí auto a fotky jsou rozmazané. Aby byly fotky lepší kvality při stmívání a během noci, bylo by třeba vyměnit kameru za NoIR s infračerveným přisvětlením.

4.1.2. Shrnutí pro Raspbian

Fotopast byla testována v rodinném domě s několika místnostmi a zdmi, kde síla signálu klesala za každou další zdi o několik procent. Když byla fotopast oddělena dvěma zdmi, síla i kvalita byla nejhorší, signál byl střídavě okolo 20-30%. A samotné Raspberry se nedokázalo stabilně udržet připojené k domácí Wi-Fi síti po delší dobu. Při této úrovni nebylo možné fotopast spustit. Výsledné naměřené hodnoty by mohli být rozdílné, kdyby testování probíhalo ve velké místnosti nebo hale, kde nejsou žádné zdi či jiné překážky. Zde by mohla například vzdálenost pro 50% sílu signálu být dvakrát tolik delší.

Ve výsledku byl Raspbian velmi uživatelský přívětivý systém pro tvorbu fotopasti. Největší výhoda je v jeho popularitě, protože na tento systém vychází spousta nových aktualizací, balíčků a podporuje téměř vše co k Raspberry připojíte. Jediný problém byl v kvalitě bezdrátového spojení. S tímto Wi-Fi donglem jsem nebyl úplně spokojený, protože připojení přes něj bylo nestabilní. Síla signálu často kolísala třeba i o 20% a to i ve vzdálenosti 3m od aktivního prvku. Když se dostala na hranici 20% tak připojení k Raspberry přes SSH spadlo a bylo nutné obnovit relaci nebo přenést Raspberry blíž k Wi-Fi routeru. Toto se dělo i když se s Raspberry nepracovalo. Následně pomohlo až upgrade firmwaru, to nestabilitu trochu snížilo, a tak časté výpadky se již neopakovali při krátké vzdálenosti, pouze při cca 18 m.

Při venkovním testování nenastaly žádné nečekané výpadky či závažné problémy. Raspberry bylo připojeno pomocí síťového kabelu, protože zde nebyl žádný Wi-Fi signál. Celkově bych toto testování považoval za kladné a fotopast by se dala používat i venku, ale pouze když nebude pršet nebo příliš velké vlhko. To by totiž mohlo poškodit celé Raspberry.



Tabulka 2 Graf Raspbian

4.2. FedBerry

Další z testovaných systému byla Fedora, konkrétně její remix nazvaný FedBerry. Je to projekt určen speciálně pro Raspberry Pi 2 model B. FedBerry staví na Fedoře 23. [31]

Před instalací samotného, systému je nutné si vybrat jaký typ si stáhneme. Na oficiálních stránkách FedBerry si stáhneme obraz, který budeme chtít používat. Testovaný systém byl verze minimal, to znamená, že je bez grafického prostředí XFCE, obsahuje pouze základní balíčky vše ostatní je třeba si doinstalovat ručně a není zde žádný předdefinovaný uživatel. Hlavní výhodou ve FedBerry je, že stačí stáhnout obraz systému, zapsat jej na kartu a může se spustit. Není třeba žádné úpravy, zvětšování oddílů či souborů, jako tomu bylo u jiných remixů Fedory.[28]

Fedora má odlišný balíčkovací systém RPM (RedHat Package Manager) oproti DEB, který je v Ubuntu a Raspbianu. Také tedy má jiný instalátor, starší verze Fedory měli pouze YUM. Ale od roku 2012 vznikl nový management softwaru DNF, který postupně nahradil YUM. Pro začátečníky práce s Fedorou může být ze začátku trochu těžší kvůli odlišným příkazům, ale čím déle s Fedorou pracujete, tím rychleji si je zapamatujete.[33]

Po stažení obrazu Fedberry, je potřeba ho zapsat na SD kartu. Opět je doporučováno mít kartu o velikosti alespoň 8GB. Uživatelé, kteří používají nějaký Unixový/Linux systém zapíší obraz pomocí těchto pár řádků.

Nejprve zkontrolujeme, zda soubor není poškozen pomocí kontrolního součtu.

```
$ sha256sum fedora-23-remix-rpi2-minimal-1.raw.xz
```

Následně připojíme SD kartu, tu je třeba správně rozpoznat kde je připojená, protože když se špatně určí, kde je připojená může se stát, že si smažete všechny existující oddíly běžícího systému. Umístění SD karty bývá většinou podobné těmto /sda/sda1, /sdb/sdb1, /dev/mmcblk0.[30]

```
$ dmesg | tail
$ lsusb
$ watch lsblk
```

Když známe umístění karty, je nutné odpojit oddíl, pokud se připojil.

```
$ umount /dev/sda/sda1
```

Další krok byl, zkopírování obrazu na SD kartu, ke spuštění příkazu je nutné být přihlášen jako root.

```
$ xzcat fedora-23-remix-rpi2-minimal-1.raw.xz | dd of=/dev/sda  
$ sync
```

Příkaz **sync** zajistí, že zápis cache byl proveden a je bezpečné odpojit kartu. Tento poslední příkaz bude probíhat velmi dlouho cca 15min.[31]

Pro uživatele Windows existuje program Fedora LiveUSB Creator, bohužel jsou zde pouze staré verze Fedory a nové remixy 23 a nyní 24 zde již nejsou. Lepší nástroj je opět Win32DiskImager. Postup téměř stejný jako u ostatních systémů. Po stažení obrazu FedBerry je ve formátu **raw.xz**, ten můžeme extrahovat pomocí programu **7 - zip**. Po extrahování lze obraz zapsat na microSD kartu, jediné na co je třeba dát si pozor tak abychom zvolili správný disk a nepřepsali jiný. Po zapsání stačí kartu vložit do Raspberry Pi a spustit.

Další možnost je mít například VirtualBox kde mít nějaký Linuxový systém a přes něj pak zapsat obraz na disk. Já jsem použil přímo Raspberry s operačním systémem Raspbian.

Fedora obsahuje inicializační skript pro příkazovou řádku (CLI), který při prvním spuštění umožní nastavení základních vlastností systému jako je lokalizace, heslo pro uživatele root a vytvoření nového uživatele. Tento skript je pouze u verze minimal a s grafickým prostředím. U verze mini je předem vytvořený **,root‘** uživatel s heslem **,fedberry‘** a také normální user **,raspberry‘** s heslem **,raspberry‘**, ale po prvním přihlášení je vyžadována změna hesel. [29]

Může se stát, že se heslo jak pro uživatele root tak i nově vytvořeného nenastaví. Nikdy by neměl být root uživatel bez hesla, hlavně kvůli bezpečnosti by k němu běžní uživatelé neměli mít přístup. Toto lze nastavit i při prvním nabofování systému v terminálu.


```
$ sudo user root
$ passwd new_pass #heslo se musí zadat dvakrát pro ověření
```

Heslo by mělo splňovat bezpečnostní pravidla, ale lze zadat i jednoduché čtyřmístné heslo například jen samé 0. Při zadání takového hesla jste varováni, že není bezpečné či, že nesplňuje délku a složitost, ale i tak je možné ho použít.

Po úspěšném přihlášení do systému je třeba spustit aktualizace a nainstalovat několik balíčků.

```
$ dnf install binutils
$ dnf install update
```

V této práci byla zvolena verze minimal bez grafického prostředí hlavně kvůli tomu, že se na Raspberry pracuje nejčastěji vzdáleně pomocí SSH a přes připojení k bezdrátové síti. SSH je na Fedoře povoleno při prvním startu automaticky, takže stačí jen zjistit, jakou IP adresu přidělil DHCP server a může se pracovat vzdáleně. Ale aby nebylo nutné při každém startu Fedory zjišťovat jakou novou IP adresu RPi dostalo, tak můžeme nastavit jednu pevnou neboli statickou.

Po provedení předchozích příkazů si stačí doinstalovat pouze balíček **picamera**, opět pro používání kamery v pythonu. Python samotný a všechny jeho verze a doplňky by ve Fedoře měli být. Stačí již pomocí **pip** nainstalovat potřebný balíček.

```
$ pip install picamera
$ pip install pushetta
```

Já jsem si zde i rovnou nainstaloval **Pushettu**, kterou jsem používal i v Raspbianu, pro posílání notifikací na smartphone.

Nyní ještě nelze používat kameru, protože není povolena v konfiguračním souboru, který se nachází v **/boot/config.txt**. Otevřeme si ho a upravíme velikost **gpu_mem** na minimálně **128MB**. Kamera nebude fungovat bez této hodnoty. Samotnou kameru povolíme okomentováním tohoto řádku.

```
$ start_x=1
```

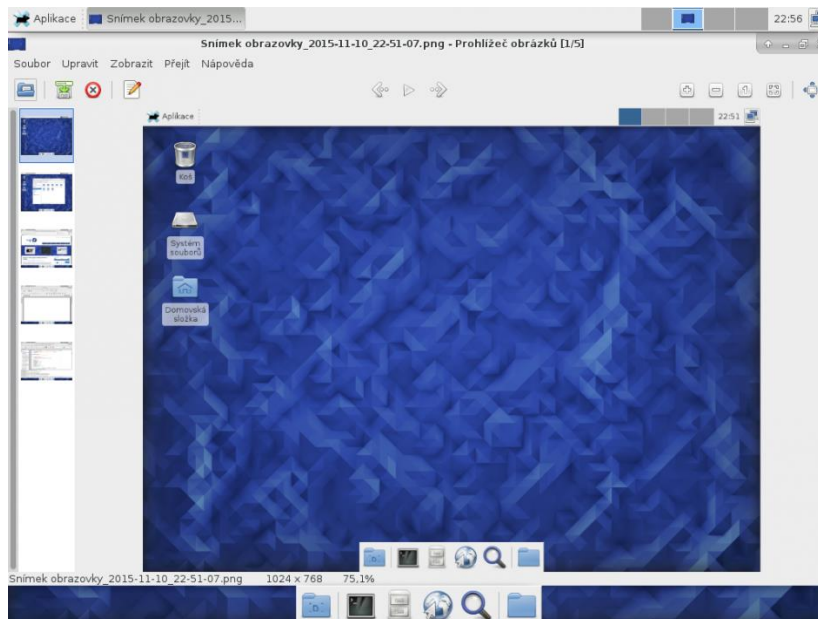
Pro nastavení statické IP adresy Wi-Fi připojení, je třeba upravit pár řádků v tomto souboru. Nutno spouštět s root právy.

```
$ sudo nano /etc/sysconfig/network-scripts/ifcfg-wlan0
```

Toto je umístění požadovaného souboru, samotné **ifcfg** znamená **Interface Configuration File**. Jsou to konfigurační rozhraní soubory konkrétního síťového zařízení. Tyto soubory ovládají softwarové rozhraní individuálních síťových zařízení. Při startu systému, jsou tyto soubory využity k rozhodnutí jaké rozhraní se má zapnout a jak ho nakonfigurovat. Obvykle jsou pojmenovány jako **ifcfg-<název zařízení>**. [32]

Po otevření konfiguračního souboru stačí přidat tyto řádky

```
$ PREFIX=24  
$ IPADDR=192.168.0.108  
$ GATEWAY=192.168.0.1  
$ BOOTPROTO= none
```



Obrázek 12 Grafické prostředí XFCE Fedora 23 [35]

Aby se konfigurace načetla, je třeba restartovat systém. V dokumentaci fedory je psáno, že není třeba zadávat **gateway**, protože je vše automaticky vypočítáno pomocí **ipcalc**. Tím se ale není dobré řídit, když **gateway** nezadáte tak se při restartu systému vymaže a Fedora ji sama získat neumí. Tento problém vás může na několik hodin zdržet, protože bez zadané **gateway**, neboli brány vaší sítě, v mém případě routeru, se nepřipojíte do internetu. [37]

4.2.1. Testování

Testovací systém byl Fedberry remix Fedory 23 s verzí jádra 4.4.15. Stejně podmínky jako u ostatních testovaných systémů.

Jelikož jsem měl staženou verzi minimal, nebylo na Fedoře nic nainstalováno. Je to v podstatě téměř holý systém a vy si na něj přidáváte svoje nástroje, které potřebujete. Tím myslím například pomocné síťové nástroje jako **net-tools**, **wireless tools**, pro provádění důležitých příkazů, které jsem v práci nejčastěji využíval.

```
$ iwconfig wlan0
$ iwlist
$ ifconfig
```

A další užitečné nástroje **usbutils**, **unzip**, **mc** pro vyhledání připojených usb zařízení, rozbalení souborů a například i Midnight Commander vizuální souborový manažer.

Jelikož jsem chtěl testovat funkčnosti fotopasti přes Wi-Fi adaptér, musel jsem ho zprovoznit. Wi-Fi připojení šlo navázat ihned, ale nedostával jsem korektní údaje z příkazu **iwconfig**. Nevěděl jsem jaká je úroveň signálu a jaká je kvalita. Řešením bylo stažení a nainstalování ovladače pro **RTL8188eu**. To je typ mého Wi-Fi adaptéru.

```
$ lsusb
Bus 001 Device 004: ID 0bda:8179 Realtek Semiconductor Corp. RTL
8188EUS 802.11n Wireless Network Adapters
```

Takto si zobrazím verzi adaptéru. Dále jsem si našel na githubu repositář se samostatnou verzí tohoto typu ovladače od uživatele **iwfinger**, který na githubu má

několik verzí ovladačů **RTL** [36]. Ovladač stáhneme pomocí příkazu **wget** a následně rozbalíme pomocí **unzip**. Může se stát, že u příkazu **wget** se objeví hláška: **unable to resolve host address** může za to DNS server. Nemusí být žádný zadaný, a proto je třeba zadat alespoň jeden například 8.8.8.8 (google.com). Pro případ že by google nefungoval lze použít některý z DNSSEC validujících resolverů od CZ.NIC.

```
$ sudo nano /etc/resolv.conf
nameserver 8.8.8.8
nameserver 217.31.204.130
nameserver 192.168.0.1
```

Přidal jsem dva veřejné DNS servery od googlu a adresu mého routeru (GW)

```
$ wget https://github.com/lwfinger/rtl8188eu/archive/master.zip
$ unzip master.zip
```

Po rozbalení vznikne složka **rtl8188eu-master**.

Otevřeme si tuto složku a spustíme kompilaci a instalaci.

```
$ make all
$ sudo make install
```

Po dokončení kompilace a instalace, stačí restartovat systém nebo jen samotné síťové zařízení. Nyní již příkaz **iwconfig** ukazuje správné hodnoty a sílu signálu i v dBm.

```
[root@localhost wifi]# iwconfig wlan0
wlan0 IEEE 802.11bgn ESSID:"Doma" Nickname:"<WIFI@REALTEK>"
Mode:Managed Frequency:2.457 GHz Access Point: C4:6E:1F:31:8E:30
Bit Rate:150 Mb/s Sensitivity:0/0
Retry:off RTS thr:off Fragment thr:off
Encryption key:****-****-****-****-****-****-****-**** Security mode
:open
Power Management:off
Link Quality=63/100 Signal level=-52 dBm Noise level=0 dBm
Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
Tx excessive retries:0 Invalid misc:0 Missed beacon:0
```

Po zprovoznění Wi-Fi adaptéru jsem již mohl začít testovat Fotopast.

Pro případ že se Fedberry nepřipojí automaticky k bezdrátové síti při startu, je dobré si upravit nastavení Wi-Fi připojení v **NetworkManageru**.

```
$ nmcli connection edit Wi-FiHome
nmcli> set connection.autoconnect TRUE
nmcli> save
nmcli> quit
```

Byly stanoveny stejné podmínky při běhu fotopasti jako u předešlých systémů. [34] Při pokusu instalovat tento ovladač se objevila chyba, že příkaz `make` nefungoval. Bylo to způsobeno tím, že jsem kompiloval jinou verzi jádra, konkrétně starší verzi, než byla verze jádra systému. Při instalaci `kernel-header` a `kernel-devel` se může stát, že tyto dva balíky budou mít jinou verzi, než je verze jádra operačního systému.

Dá se to opravit instalací správné verze. Stačí si pomocí příkazu `uname -r` vypsat verzi jádra a tu zkopírovat za **kernel-devel** a za **kernel-headers**.

```
$ uname -r
$ yum install kernel-devel-3.9.5-301.fc23.x86_64
$ yum install kernel-headers-3.9.5-301.fc23.x86_64
$ yum install gcc
```

Po těchto příkazech by mělo vše fungovat.

I přesto není zaručeno, že bude všechno v pořádku. Protože poprvé co jsem nainstaloval takto ovladač, jsem následně spustil upgrade systému. Což se ukázalo jako fatální chyba. Po upgradu se systém restartoval a bohužel tím to skončilo. Systém již znovu nenaběhl, chyba byla v samotném jádru. Rozhodil jsem jádro systému, opravit by to šlo, pouze sestavením vlastního jádra. Celá oprava by trvala mnohem déle než samotná reinstalace celého systému. Po druhé, co jsem instaloval celý systém od začátku jsem si vybral verzi s GNU. Také jsem spustil upgrade systému před samotnou instalací **kernel-devel** a **kernel-headers**.

Začal jsem na 5 metrové vzdálenosti. Při této vzdálenosti je Fotopast v jedné místnosti společně s AP, nedělí je žádná zeď, a tak je síla signálu výborná. Nestane se, že by Raspberry spadlo, pouze se občas zasekne systém, ale po chvíli se opět rozeběhne.

```
root@localhost dan]# iwconfig wlan0 | grep -i --color signal
Link Quality=69/100 Signal level=-50 dBm Noise level=0 dBm
root@localhost dan]# █
```

U vzdálenosti cca 10m od AP již Raspberry dělí zeď, už se zde objevují i občasné výpadky. Když jsem fotopast pouštěl bez nainstalovaného ovladače pro Wi-Fi adaptér, tak Raspberry spadlo během jedné hodiny pětkrát. A **iwconfig** ukazoval nulové hodnoty. Proto je lepší si ovladače nainstalovat. Měli by zajistit lepší podporu Wi-Fi adaptéru.

```
[dan@localhost ~]$ iwconfig wlan0 | grep -i --color signal
Link Quality=0/100 Signal level=10/100 Noise level=0/100
[dan@localhost ~]$
```

Při této vzdálenosti se dá stále plynule pracovat na Raspberry. Sice občas nastanou výpadky, ale je to mnohem snesitelnější než při vzdálenosti okolo 15m. Zde se často ani spojení nenaváže napoprvé. Každý půlmetr je zde znát, protože když jsem přesunul Fotopast na kraj zdi ke dveřím, tak mohla být spuštěná i 3h bez výpadku. Ale jakmile jsem zkusil posunout Raspberry i jen o půl metru ode dveří podél zdi, došlo k výpadku po 15 minutách.

```
[root@localhost etc]# iwconfig wlan0 | grep -i --color link
Link Quality=0/100 Signal level=-74 dBm Noise level=0 dBm
```

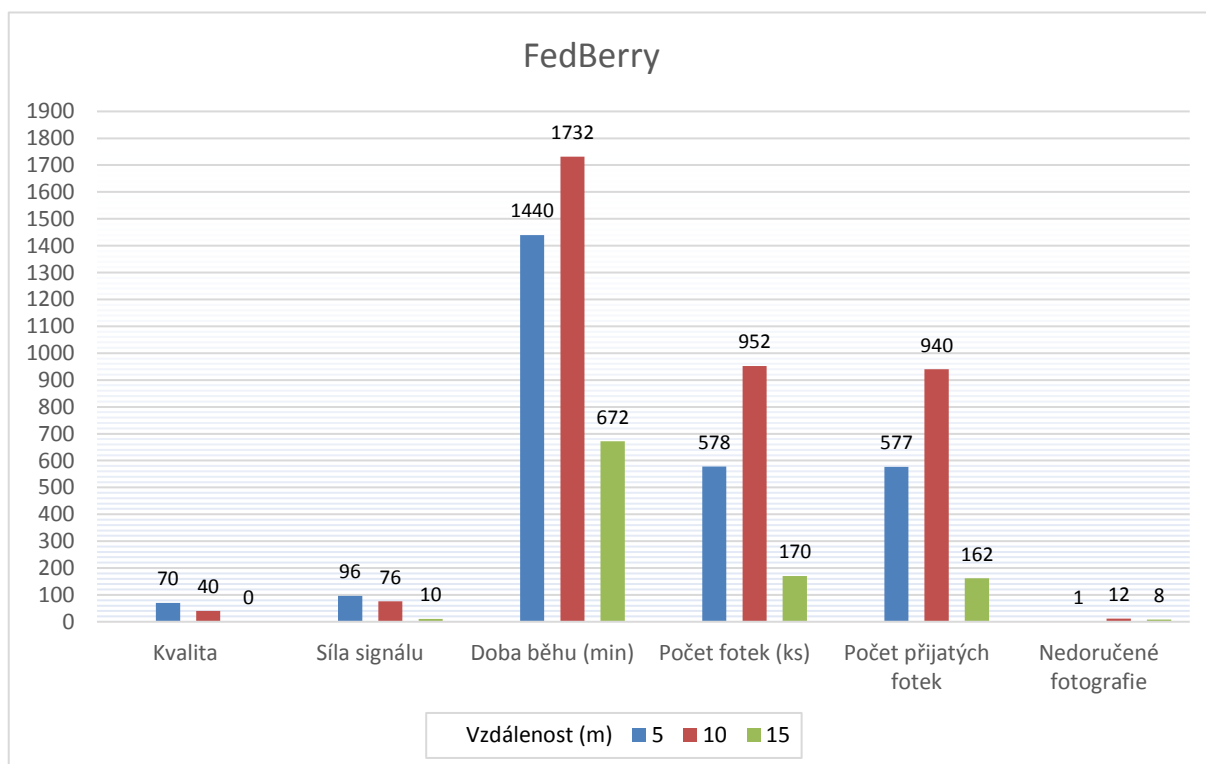
Opět jako poslední a hraniční vzdálenost byla cca 15 m, třetí místnost od AP oddělená dvěma zdmi. Připojení bylo opět velice nestabilní, fotopast zde nebyla testována ani polovinu času jako u předchozích vzdáleností, protože výsledek nebyl jiný. Stále dokola se opakoval jen jeden problém, kvůli slabé síle signálu se spojení s Raspberry odpojovalo a často bylo nutné Raspberry vypnout a zapnout. Proto je tato vzdálenost opravdu hraniční a na práci není vhodná.

Jak je vidět na obrázcích, položka **Link Quality** se rovná vždy nule, tento problém se mi nepodařilo vyřešit žádnou aktualizací nebo instalací ovladačů. Tyto hodnoty se ukazují pouze když měřím ve vzdálenosti 10 a 15 metrů. Když jsem hledal, jak tento problém vyřešit často lidé psali, že jim nejde internet při této hodnotě, ale mě připojení fungovalo.

4.2.2. Shrnutí pro FedBerry

FedBerry byl zcela jiný systém než ostatní testované systémy. Začátky byly složité, protože jsem neznal správné příkazy, a tak jsem musel neustále hledat příkazy na internetu. Je pravda že čím častěji jsem na Fedberry pracoval, tím jsem si víc příkazů pamatoval. Vyzkoušel jsem verzi jak minimal (bez grafického prostředí), tak verzi s grafickým prostředím XFCE. Ale většinu času jsem pracoval pouze v terminálu, připojený přes SSH. Běh systému byl plynulý, nezasekával se. Plus je, že na systém často vychází update a je spousta programů dostupných k nainstalování.

Jediné, co bych vytkl je to, že díky neznalosti příkazů se dá systém lehce poškodit, mně se to povedlo několikrát. Ale to není chyba systému, spíše lidská chyba. Největší problém zde byl opět instalace ovladače pro Wi-Fi adaptér.



Tabulka 3 Tabulka srovnání Fedberry

Systém bez instalace ovladače ukazoval velice nízké hodnoty, i když Raspberry bylo blízko AP. Po instalaci se to trochu zlepšilo, ale bohužel jen o málo.

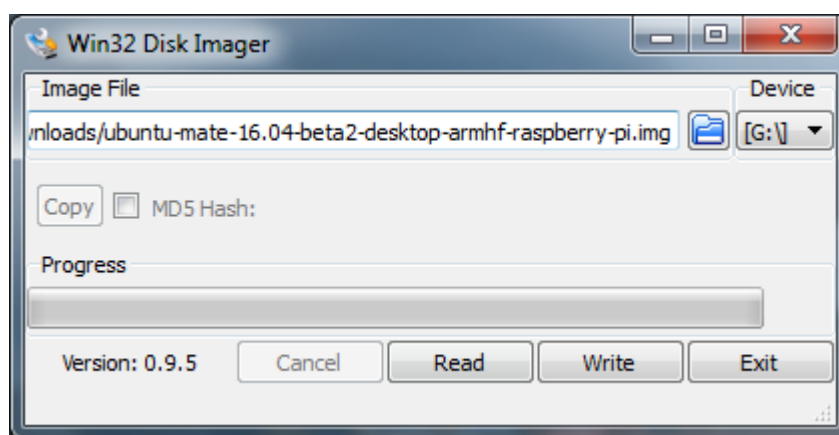
Podle mého názoru, je to tímto typem adaptéru, tudíž bych doporučoval použít jiný. Celkově se mi tento systém zalíbil a podle mě je to ideální systém pro tvorbu Fotopasti. Tento systém bych nedoporučoval začátečníkům, je spíše pro ty kdo již s Fedorou pracovali a mají větší zkušenosti.

4.3. Ubuntu

Jako třetí systém byl použit operační systém Ubuntu typu MATE verze 16.04 určen a optimalizován pro Raspberry Pi 2 a nově i Raspberry Pi 3. Ubuntu MATE v 16.04 má již plně funkční bluetooth a Wi-Fi pro Raspberry Pi 3 [23].

The Raspberry PI Ubuntu MATE je založen na Ubuntu armhf procesoru a ne na novém Snappy Ubuntu Core. Z toho vyplývá, že funkce jsou stejné jako na desktopové verzi a používá i stejný balíčkovací systém, takže spravovat balíčky lze přes **apt-get** [23].

Celková instalace je velmi lehká, nejdříve je nutné si stáhnout obraz z oficiálních stránek Raspberry. Pro zapsání obrazu na kartu (nejlépe třída 10 a minimální velikost 8GB) na operačním systému Windows, stačí použít program Win32DiskImager a extrahovaný obraz zapsat na kartu.



Obrázek 13 Win32 Disk Manager

Uživatel operačního systému Linux, obraz zapíše na kartu pomocí třech řádků v terminálu. Ještě před provedením posledního řádku je nutné si zjistit id zařízení SD karty (**/dev/sdx**), stačí použít příkaz **lsblk**.


```
$ lsblk
$ sudo apt-get install gddrescue xz-utils
$ unxz ubuntu-mate-16.04-desktop-armhf-raspberry-pi.img.xz
$ sudo ddrescue -D --force ubuntu-mate-16.04-desktop-armhf-raspberry-pi.img /
dev/sdx
```

K samotné instalaci Ubuntu bylo potřeba mít k dispozici obrazovku s HDMI, klávesnici, myš a připojení k internetu pomocí ethernetového kabelu. Během instalace je spuštěn průvodce celkovým nastavením počítače, jako nastavení časového pásma, rozložení klávesnice a nejdůležitější vytvoření uživatelského účtu a nastavení práv. Po instalaci není možné využívat celou velikost karty. Je třeba manuálně přenastavit souborový systém. [23]

Ve verzi 16.04 to lze dvěma způsoby, buď pomocí **Ubuntu MATE Welcome** který automaticky přenastaví oddíly na plné využití kapacity microSDHC karty. Na úvodní obrazovce stačí kliknout na tlačítko **Raspberry Pi Information**, pak na tlačítko **Resize** a restartovat Raspberry Pi. Druhá možnost je přenastavit to manuálně v terminálu. Po prvním nabootování zadat do terminálu:

```
$ sudo fdisk /dev/mmcblk0
```

Následně smazat druhý oddíl (**d, 2**), poté jí opětovně vytvořit s defaultním nastavením (**n, p, 2, enter, enter**), zapsat a ukončit (**w**). Nakonec restartovat systém a na závěr napsat:

```
$ sudo resize2fs /dev/mmcblk0p2
```

Po dokončení si lze ověřit souborový systém příkazem **df -h**. [24]

Kdo by chtěl spíše využívat grafickou podobu Ubuntu, jsou dostupné i hardwarové akcelerace, především pro video přehrávače OmxPlayer a VLC.

Ubuntu MATE se liší od Raspbianu pouze v absenci předdefinovaného uživatele. Pro lepší bezpečí je dobré mít vytvořeného superuživatele (root) a běžného uživatele (Raspi). Protože když se v souboru „**sudoers**“ mění práva, například aby mohl běžný uživatel používat příkaz **sudo** bez zadávání hesla, či přistupovat do chráněných složek, může se stát, že to bude zadáno chybně. V důsledku toho bude ztracen přístup k root účtu či příkazu **sudo**, což je fatální chyba. [23]

Ve verzi 16.04 se mohou vyskytovat problémy s wi-fi připojením. Jeden je že během první konfigurace systému balíček Ubiquity nevyzve k připojení k dostupné bezdrátové síti. Další je, že po kompletní instalaci nastavení Wi-Fi připojení nefunguje, ale po restartování již je dostupné a funguje.[23]

U wi-fi adaptéru TP Link WN725N se může stát, že problém s nastavením bezdrátového připojení přetrvává i po restartování. Potom se musí stáhnout a nainstalovat ovladače ručně.

Nejprve si musíte vyhledat verzi jádra

```
$ uname -a
```

To nám ukáže například pro Pi 2 B a 3 Linux raspberrypi 4.4.8-v7+ #881 SMP Sat Apr 30 12:16:50 BST 2016 armv7l GNU/Linux. Takže soubor, který musíme stáhnout je **8188eu-4.4.8 - v7-881.tar.gz**

Na této stránce si najdeme číslo jádra, které se zobrazilo při zadání **uname -a**. V tomto případě je to níže uvedený odkaz, kdyby bylo číslo jádra jiné, tak by se v odkazu změnilo pouze číslo za **8188eu-X.X.X**.tar.gz

```
$ wget https://dl.dropboxusercontent.com/u/80256631/8188eu-4.4.8-v7-881.tar.gz
$ tar xzf 8188eu-4.4.8-v7-881.tar.gz #(-x -extract -z -gzip -f -file)
$ ./install.sh
```

Takto stáhneme driver z úložiště na dropboxu, následně extrahujeme a spustíme instalaci. Pak by bezdrátové připojení mělo fungovat.

Abychom mohli pracovat s Ubuntu vzdáleně, je třeba zajistit automatické připojení k bezdrátové síti při bootování a ne až po přihlášení uživatele. Toto se zařídí, když upravíme řádek v souboru, kde máme informace o nastavení naší bezdrátové sítě.

```
$ cd /etc/NetworkManager/system-connections
$ sudo nano Wi-Fi-Home
```

Zde najít řádek **permission** a smazat vše za znamínkem ‚=‘, uložit a restartovat Raspberry. V Ubuntu MATE neexistuje příkaz pro zobrazení konfiguračního souboru jako v Raspbianu.

```
$ raspi-config
```

Když je třeba povolit například Pi Cameru, tak si ho musíme upravit pomocí tohoto příkazu v terminálu. Tento postup je stejný jako u FedBerry.

```
$ sudo nano /boot/config.txt
```

Potom najít řádek s nápisem „Camera Settings“, nebo jednoduše vyhledat pomocí klávesových zkratk „control + W“ a odkomentovat řádek.

```
##start_x=1  
start_x=1
```

Tímto se povoluje kamera, defaultně je zde nastavená hodnota „0“ neboli zakázáno. Dále je doporučeno mít hodnotu gpu_mem na minimálně 128MB.

```
gpu_mem=128
```

Následně uložit „control + X“ a restartovat systém.

```
$ sudo reboot
```

Po restartování by již kamera měla fungovat, to si lze ověřit příkazem:

```
$ sudo raspistill -o image.jpg
```

Když je kamera funkční, je nutné stáhnout ještě python knihovnu „picamera“, aby bylo možné kameru používat ve skriptech.

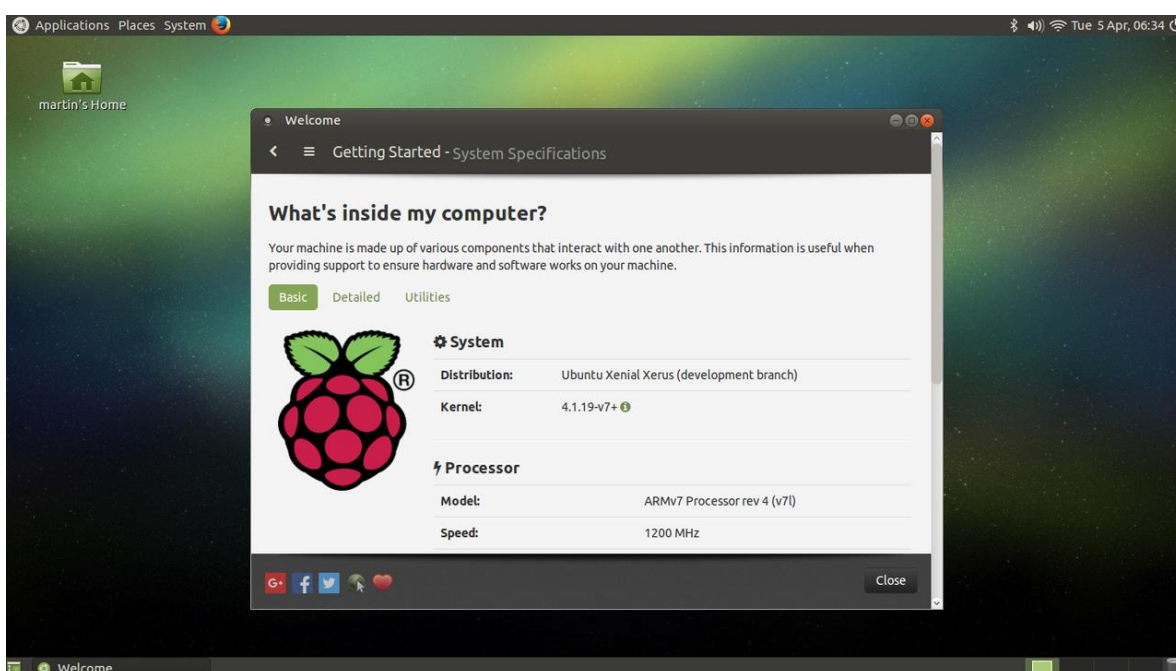
```
$ sudo apt-get install python-picamera
```

4.3.1 Testování

Testování funkčnosti fotopasti na operačním systému Ubuntu MATE verze 16.04, probíhalo se stejným wi-fi donglem jako u Raspbianu. Vše bylo prováděno ve stejných vzdálenostech i s téměř stejnou délkou běhu.

Pro funkčnost Wi-Fi adaptéru je zapotřebí si stáhnout ovladače, pro TP Link WN725e z oficiálních stránek Raspberry.

Grafické prostředí Ubuntu MATE je typu gnome, jednoduché na ovládání, podobné jako na Raspbianu.



Obrázek 14 Grafické prostředí Ubuntu MATE [23]

Fotopast byla opět testována od nejmenší vzdálenosti po co největší možnou vzdálenost.

Ze začátku se vyskytovala chyba s připojením k Wi-Fi síti při startu systému. Dokud se nepřipojil ethernetový kabel k Raspberry a nepřihlásil uživatel, tak se Ubuntu nepřipojilo k bezdrátové síti. Je to způsobeno defaultní konfigurací síťového připojení, kde je nastaveno, že se systém připojí až po přihlášení prvního uživatele.

U tohoto systému se opakuje jedna chyba, při zapnutí Raspberry se nepřipojí k bezdrátové síti, dokud není připojen síťový kabel. Po té již bezdrátové připojení naběhne. Tato chyba se vyskytuje častěji u vzdálenosti 15 metrů od Wi-Fi Routru. Opět je to způsobeno nedostatečnou a kolísavou silou signálu. Pro zlepšení

přijímaného signálu nepomohlo ani upgrade ovladačů Wi-Fi donglu. Bylo by zapotřebí použít výkonnější Wi-Fi router, nebo lépe rozmístit po budově AP.

Jako první byla vzdálena 5m od aktivního prvku. Stabilita připojení byla vynikající, při této vzdálenosti nebyl očekáván žádný problém a také žádný nenastal, vše bylo úspěšné.

```
dan@dan-desktop:~/Detector$ iwconfig wlan14cc2015cad9 | grep -i --color signal
Link Quality=100/100 Signal level=96/100 Noise level=0/100
dan@dan-desktop:~/Detector$
```

Síla signálu byla stejně jako u Raspbianu kolísavá a v této vzdálenosti se pohybovala od 96/100 do 60/100. A to, i když se s Raspberry vůbec nehýbalo a bylo stále na stejném místě.

Samotná fotopast byla vystavena středně rušnému prostředí, pohyb nebyl nepřetržitý. Při této vzdálenosti se ani na Ubuntu MATE, nevyskytla žádná chyba.

Vzdálenost 10m

```
dan@dan-desktop:~/Detector$ iwconfig wlan14cc2015cad9 | grep -i --color signal
Link Quality=100/100 Signal level=46/100 Noise level=0/100
dan@dan-desktop:~/Detector$
```

Raspberry mělo problém se při této vzdálenosti připojit k bezdrátové síti, bylo zapotřebí nejdříve připojit Raspberry ethernetovým kabelem. Bylo nutné přihlášení uživatele, a až pak se připojilo k Wi-Fi síti. Tento drobný nedostatek byl vyřešen automatickým připojením k Wi-Fi síti při startu systému (viz. výše). Fotopast byla vystavena nepřetržitému provozu. V průměru pořídila a odeslala každou minutu alespoň 7 fotek.

Během první hodiny měření došlo k dvěma výpadkům Raspberry, při prvním nedošlo k žádné ztrátě dat. Výpadek byl chvilkový a Raspberry se znovu připojilo k Wi-Fi síti. Při druhém výpadku se nedešly dvě fotografie. Tyto dvě fotografie sice byly uloženy na vnitřním úložišti, ale obě byly poškozeny a nešly otevřít. Výpadek trval 4 minuty a pro opětovné připojení bylo třeba připojit ethernetový kabel. Skript byl stále spuštěn v pozadí, ale nic už nefotil, takže byl ukončen pomocí příkazu **kill pid**

```
$ ps aux |grep python #vyhledání všech spuštěných procesů python
$ kill pid #pid = id běžícího procesu
```

Vzdálenost 15m

```
dan@dan-desktop:~$ iwconfig wlan14cc2015cad9 | grep -i --color signal
      Link Quality=72/100   Signal level=32/100   Noise level=0/100
dan@dan-desktop:~$ █
```

Tato vzdálenost je už hraniční s normálním používáním Raspberry. Dochází k častému ukončení spojení v programu Putty kvůli vypršení časového limitu.

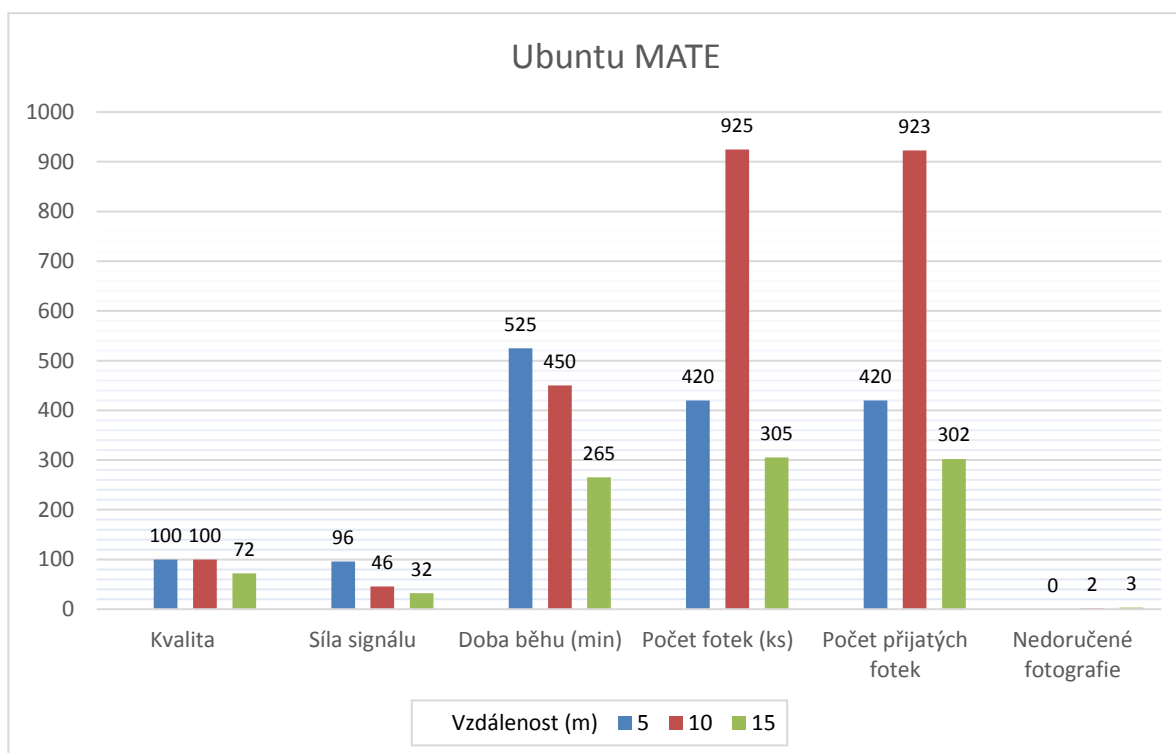
To je způsobeno nedostatečným signálem bezdrátové sítě. V podstatě zde nastává to samé, co u Raspbianu a Fedory. Při této vzdálenosti již nerozhoduje, jaký operační systém používáte, ale pouze jak silný je signál, nebo jak výkonný máte Wi-Fi adaptér.

Když dojde k výpadku sítě a Raspberry se snaží opětovně navázat spojení, dochází k spadnutí celého systému. Protože napájecí adaptér nestačí napájet Raspberry, které má najednou větší výkon a dojde k pádu. V lepším případě se pouze odpojí od Wi-Fi sítě a musíte pak dojít k Raspberry, manuálně se přihlásit a zkusit se znovu připojit nebo vypnout celé RPi.

4.3.2 Shrnutí pro MATE

Při testování bylo vyzkoušeno, že když je Raspberry připojeno pomocí síťového kabelu a Wi-Fi donglu k síti, tak odpojení kabelu nebo adaptéru nemá vliv na běh skriptu. Ten dále běží na pozadí, dokud není násilně ukončen. Byla přidána výjimka, že v případě výpadku sítě, skript nespadne, ale pokračuje dál bez odesílání na email, pouze ukládá na disk a při opětovném připojení sítě se začnou fotografie opět posílat na email.

Celkově je systém Ubuntu Mate 16.04 podobný Raspbianu. Při vzdáleném ovládní systému přes protokol SSH (22), nelze poznat rozdíl mezi těmito systémy. Ale při používání grafického prostředí to znát již bylo.



Tabulka 4 Graf Ubuntu

Ubuntu bylo pomalejší a pro mě hůře ovladatelné. Také bylo zapotřebí provést více příkazů, abych mohl systém plně využívat, nejdřív znovu zvětšení SD karty, aby bylo možné využívat celou velikost. Dalším problémem bylo zprovoznit Wi-Fi adaptér. Přimo na stránkách Ubuntu MATE, jsou popsány dva známé problémy s Wi-Fi připojením, které nejsou zatím vyřešené.

Pro zprovoznění mého Wi-Fi adaptéru, jsem musel stáhnout ovladače jako pro Raspbian, které naštěstí fungovaly. Ubuntu má pomocí NetworkManageru

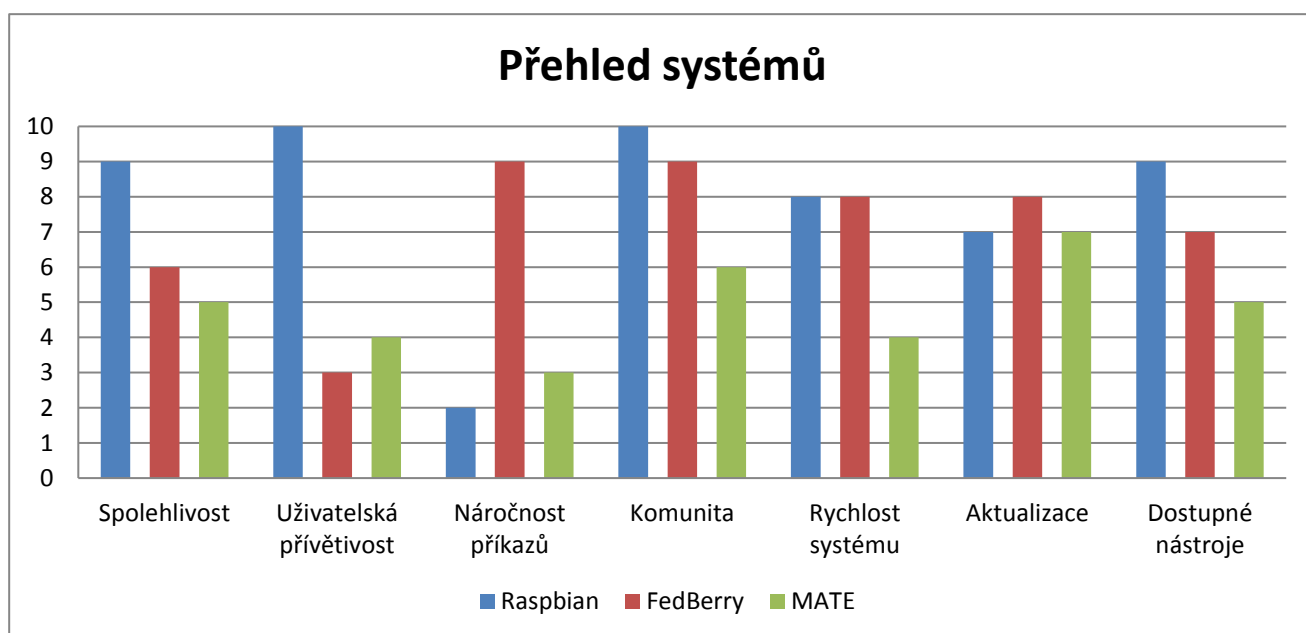
vytvořený soubor s uskutečněným Wi-Fi připojením, kde jsou uloženy informace o této Wi-Fi síti. Je zde defaultně nastavené, že se připojí k síti až po přihlášení uživatele. Toto nastavení se musí zrušit, aby to mohlo fungovat při bootování systému.

Ale na práci pouze v grafickém režimu na Raspberry Pi 2 tento systém není stavěný, je stále příliš pomalý a běh systému není plynulý. Vhodný není ani na surfování po internetu, načítání a zobrazování stránek je pomalé. Ani v terminálu nebyl běh systému plynulý při náročnější operaci se občas zasekával, zde to ale mohlo být způsobeno i Wi-Fi adaptérem, který nebyl bezproblémový. Raspbian byl mnohem více optimalizovaný na Pi 2 a běh systému byl rychlejší a plynulejší. Toto platí pouze pro Raspberry Pi 2 B, na Pi 3 by již mohl být systém svižnější a rychlejší.

5. Celkové srovnání

Na všech systémech bylo použito stejné schéma zapojení hardwaru. Jako programovací jazyk byl zvolen Python verze 2.7.3.

Raspbian byl testován jako první ještě ve verzi Wheezy 7, ale i po upgradu na novou verzi Jessie 8 vše stále funguje, jak má. Ze všech systémů je opravdu podle mého názoru nejlepší možný systém pro tvorbu fotopasti, i jiných projektů. Je jednoduchý na ovládání, spousta programů je zde předinstalovaných, navíc má velkou komunitu lidí, kteří jsou ochotni pomoci. Největší plus bylo v instalaci Wi-Fi adaptéru, jediné u Raspbianu byla bezproblémová. Jelikož jsem během tvorby mé práce používal Raspberry Pi 2 bylo GUI Raspbianu stále trochu pomalejší. S novým Raspberry Pi 3 si myslím že bude GUI stabilnější a plynulejší. Na pomyslné druhé místo bych zařadil Fedoru tedy její remix FedBerry 23. Před pár měsíci ještě nešel její image zapsat na kartu, když jste pracovali v operačním systému Windows. Po vypuštění verze 24 již lze normálně zapsat pomocí nástrojů jako u ostatních systémů. Jen škoda že stále není obsažená v programu NOOBS. Tento systém je podle mě taky velice dobrý, ale když porovnáme, jak se mi pracovalo v tomto systému a v Raspbianu, vítězí u mě Raspbian, protože ta práce byla pro mě lehčí. Navíc prý po prvním spuštění Fedberry neobsahuje přehrávače medií s vysokým rozlišením, například omxplayer. To pro mě nic neznamenalo, protože jsem pracoval převážně vzdáleně přes SSH. A jako poslední systém byl Ubuntu MATE, který mě velmi zklamal.



Tabulka 5 Srovnání systémů

Celková práce na tomto systému byla velice pomalá, často došlo k zamrznutí celého systému, a to při vzdáleném připojení přes terminál i při práci přímo v grafickém prostředí. Z těchto důvodů bych tento systém nedoporučoval.

6. Závěr

Na závěr bych chtěl říci, že na všech testovaných systémech se mi povedlo fotopast zprovoznit. Schéma zapojení hardwaru bylo na všech systémech stejné. Za nejvhodnější systém pro uživatele, kteří jsou začátečníci s Linuxovými distribucemi bych vybral Raspbian. Naopak pro ty, kteří mají zkušenost s Fedorou, bych doporučil Fedberry.

Všechny cíle mé bakalářské práce, tedy návrh, samotná realizace sestavení fotopasti za pomoci jednodeskového počítače Raspberry Pi, a následné otestování funkčnosti na několika operačních systémech, byly splněny. Všechny postupy jsou zveřejněny jako open-source na webu GitHub[38]. Povedlo se mi tedy vytvořit funkční fotopast na mini počítači Raspberry Pi, které bylo vybráno kvůli jeho malým rozměrům a nízké ceně. Tím bylo prokázáno, že lze fotopast pořídit za mnohem menší náklady a možností mít lepší kvalitu kamery. Výhodou této konstrukce je, že si ji lze upravit dle vlastních potřeb na rozdíl od komerčních kamer, které jsou uzavřené a nelze si je upravovat (software a hardware).

Jelikož zařízení není voděodolné je lepší ho používat pouze ve vnitřních objektech. Zároveň bych doporučil umístit ji do většího prostoru, například sklad, hala a další objekty kde se nenachází moc zdí, aby se zvedla síla signálu a tím i vzdálenost kam až můžeme Raspberry Pi umístit. Samotná fotopast je určena pro focení ve dne. Pro focení v noci by se musela použít jiná kamera, konkrétně Pi Camera NoIR, která je zbavená infračerveného filtru a pomocí infračerveného přisvětlení je schopná fotit v noci.

Na internetu existuje mnoho zajímavých projektů či jen nápadů pro práci s Raspberry Pi například vytvoření levné 360° kamery. Na mojí práci by dále šlo navázat, například nahráváním fotek na lokální NAS server, či FTP server, nebo si vytvořit aplikaci pro android kde by s notifikací přišla i fotografie.

Seznam použité literatury

- [1] A History of Camera Trapping. *Camera Traps in Animal Ecology*. 1. USA: Springer Japan, 2011, s. 9-26. ISBN 978-4-431-99494-7.
- [2] Deer Hunting Trail Cameras - Types of Trail Cameras. In: *Ezine articles* [online]. 2010 [cit. 2016-01-27]. Dostupné z: <http://ezinearticles.com/?Deer-Hunting-Trail-Cameras---Types-of-Trail-Cameras&id=4682170>
- [3] *Fotopast* [online]. 2014 [cit.2016-01-28]. Dostupné z: <http://www.fotopast.cz/fotopasti-na-zlodeje/23-fotopast-um-565.html>
- [4] *Raspberry Pi* [online]. [cit. 2016-01-29]. Dostupné z: <https://www.raspberrypi.org/help/what-is-a-raspberry-pi/>
- [5] *Raspberry Pi 2 model B* [online]. [cit:2016-01-28]. Dostupné z: <http://rpishop.cz/raspberry-pi-pocitace/170-raspberry-pi-2-1024-mb-ram.html>
- [6] *Raspbian* [online]. [cit. 2016-01-29]. Dostupné z: <http://www.raspbian.org/>
- [7] McManus, S. & Cook, M. 2013., 20., *Raspberry Pi For Dummies*. Hoboken: John Wiley & Sons, Inc,
- [8] *Pidora* [online]. 2014 [cit.2016-01-29]. Dostupné z: <http://zenit.senecac.on.ca/wiki/index.php/Pidora>
- [9] *Arch Linux* [online]. 2015 [cit. 2016-01-29]. Dostupné z: https://wiki.archlinux.org/index.php/Arch_Linux
- [10] *CAMERA MODULE* [online]. [cit. 2016-01-29]. Dostupné z: <https://www.raspberrypi.org/products/camera-module/>
- [11] *Raspberry Pi Camera Board* [online]. [cit. 2016-01-29]. Dostupné z: <https://www.adafruit.com/products/1367>
- [12] *Pushetta* [online]. 2014 [cit. 2016-02-04]. Dostupné z: <http://www.pushetta.com/pushetta-docs/>
- [13] *PIR senzor* [online]. 2012 [cit. 2016-02-04]. Dostupné z: <http://senzory.aspone.cz/pir.htm>
- [14] *Motion* [online]. 2016 [cit. 2016-02-08]. Dostupné z: <http://www.lavrsen.dk/foswiki/bin/view/Motion>
- [15] ŠÍN, Martin. Motion - zachyťte pohyb. In: *Linuxexpres* [online]. 2006 [cit. 2016-02-08]. Dostupné z: <http://www.linuxexpres.cz/praxe/motion-zachytte-pohyb>
- [16] *Raspberry Pi 3*. *Raspberrypi.org* [online]. 2016 [cit. 2016-03-01]. Dostupné z: <https://www.raspberrypi.org/blog/raspberry-pi-3-on-sale/>
- [17] PIR jako detektor pohybu. *Senzory.aspone* [online]. 2012 [cit. 2016-03-01]. Dostupné z: <http://senzory.aspone.cz/pir.htm>
- [18] *Weaved* [online]. 2016 [cit. 2016-03-17]. Dostupné z: <https://www.weaved.com/>
- [19] *Weaved Raspberry Pi*[online]. 2016 [cit. 2016-03-17]. Dostupné z: <https://www.weaved.com/raspberry-pi-remote-connection/>
- [20] MIME: What is MIME? *Computer.howstuffworks* [online]. 2016 [cit. 2016-03-17]. Dostupné z: <http://computer.howstuffworks.com/question334.htm>
- [21] Email. *Python docs* [online]. 2016 [cit. 2016-03-17]. Dostupné z: <https://docs.python.org/3/library/email.html#module-email>

- [22] Python. *Python* [online]. 2016 [cit. 2016-03-17]. Dostupné z: <https://www.python.org/about/>
- [23] Ubuntu Mate. *Ubuntu Mate* [online]. Canocinal, 2016 [cit. 2016-10-04]. Dostupné z: <https://ubuntu-mate.org/raspberry-pi/>
- [24] , Anand. Run Ubuntu on Raspberry Pi 2 with Ubuntu MATE. *HtpcBeginner* [online]. 2015 [cit. 2016-10-04]. Dostupné z: <http://www.htpcbeginner.com/run-ubuntu-on-raspberry-pi-2-ubuntu-mate/>
- [25] Eduroam. *Eduroam wpasupplicant* [online]. CESNET, 2007 [cit. 2016-10-04]. Dostupné z: https://www.eduroam.cz/cs/uzivatel/sw/nix/wpa_supplicant
- [26] Aptosid. *Aptosid manual* [online]. Los Angels, 2012 [cit. 2016-10-04]. Dostupné z: <http://manual.aptosid.com/en/inet-setup-en.htm>
- [27] BEULQUE, Pieter. Automatically connect a Raspberry Pi to a Wi-Fi network. *We Work We Play* [online]. 2011 [cit. 2016-10-04]. Dostupné z: <http://weworkweplay.com/play/automatically-connect-a-raspberry-pi-to-a-Wi-Fi-network>
- [28] AGREZ, Vaughan. Fedberry. *GitHub Fedberry* [online]. 2016 [cit. 2016-10-04]. Dostupné z: <https://github.com/fedberry/fedberry>
- [29] KOTEK, Lukáš. Fedora 23 na Raspberry Pi 2. *MojeFedora* [online]. 2015 [cit. 2016-10-04]. Dostupné z: <https://mojefedora.cz/fedora-23-na-raspberry-pi-2/>
- [30] Fedora 22 or 23 on Raspberry Pi 2. *CHISIGHT* [online]. 2015 [cit. 2016-10-04]. Dostupné z: <https://chisight.wordpress.com/2015/10/19/fedora-22-or-23-on-raspberry-pi-2/>
- [31] *FedBerry* [online]. Red Hat [cit. 2016-10-04]. Dostupné z: <http://fedberry.org/>
- [32] Fedora. *Fedora: Documentation* [online]. [cit. 2016-10-04]. Dostupné z: https://docs.fedoraproject.org/en-US/Fedora/13/html/Deployment_Guide/s1-networkscripts-interfaces.html
- [33] KOZUMPLÍK, Aleš. Nový správce balíčků dnf a fedora 20. *MojeFedora* [online]. 2013 [cit. 2016-10-04]. Dostupné z: <https://mojefedora.cz/novy-spravce-balicku-dnf-a-fedora-20/>
- [34] Příručka uživatele Fedora. *Wiki mojefedora* [online]. 2016 [cit. 2016-10-04]. Dostupné z: <https://wiki.mojefedora.cz/doku.php?id=navody:prirucka:obsah>
- [35] *Fedora xfce* [online]. 2015 [cit. 2016-10-04]. Dostupné z: <https://mojefedora.cz/jak-vypada/fedora-23-xfce/xfce6/>
- [36] iwfinger. RTL8188eu. *GitHub RTL8188eu* [online]. 2016 [cit. 2016-10-04]. Dostupné z: <https://github.com/lwfinger/rtl8188eu>
- [37] IP Networking. *RedHat Documentation* [online]. 2016 [cit. 2016-10-25]. Dostupné z: https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Networking_Guide/part-IP_Networking.html
- [38] ŠVÁB, Daniel. *Thesis* [online]. 2016 [cit. 2016-11-15]. Dostupné z: <https://github.com/DanielSvab/Thesis>
- [39] Fedberry. *FedBerry* [online]. 2016 [cit. 2016-12-03]. Dostupné z: <http://download.fedberry.org/releases/>

- [40] *Schéma zapojení* [online]. 2016 [cit. 2016-12-03]. Dostupné z:
<http://www.girliemac.com/blog/2015/12/25/kittycam-raspberrypi-camera-cat-face-recog-nodejs/>

Seznam obrázků

Obrázek 1 IR Fotopast [3]	2
Obrázek 2 Raspberry Pi 2 model B [5]	3
Obrázek 3 Pi Camera zapojená k Raspberry Pi	6
Obrázek 4 Zapojení PIR senzoru	7
Obrázek 5 Schéma zabírajícího úhlu	12
Obrázek 6 GNOME Grafické prostředí Raspbian wheezy	13
Obrázek 7 Schéma zapojení [40]	19
Obrázek 8 Wpa_Supplicant.....	20
Obrázek 9 Interfaces	20
Obrázek 10 Putty Error	22
Obrázek 11 Grafické prostředí XFCE Fedora 23 [35]	29
Obrázek 12 Win32 Disk Manager	35
Obrázek 13 Grafické prostředí Ubuntu MATE [23].....	39

Seznam příloh

Příloha 1 Skripty	51
Příloha 3 Fotografie "lupiče", Fotopast	54
Příloha 2 Ukázka notifikací a emailu v mobilu	53
Příloha 4 Testování venku	55

Přílohy

Příloha 1 Skripty

```
import RPi.GPIO as GPIO
import time
import sys
import traceback
from pokus import SendMail #importuju ze skriptu pokus.py metodu SendMail
import picamera
import datetime
from pushetta import Pushetta

def get_file_name():
    return datetime.datetime.now().strftime("/home/pi/Detector/photo/%Y-%m-%d_%H.%M.%S.jpg")

def main():
    try:
        sensor = 4

        GPIO.setmode(GPIO.BCM)
        GPIO.setup(sensor, GPIO.IN, GPIO.PUD_DOWN)

        previous_state = False
        current_state = False
        cam = picamera.PiCamera()

        API_KEY="e224478433d166114af5e762433790fb5f5921a5"
        CHANNEL_NAME="CzechBery"
        p=Pushetta(API_KEY)
        print "...Camera trap is ready... "

        while True:
            time.sleep(0.1)
            previous_state = current_state
            current_state = GPIO.input(sensor)
            if current_state != previous_state:
                new_state = "HIGH" if current_state else "LOW"
                print("GPIO pin %s is %s" % (sensor, new_state))
                if current_state:
                    fileName = get_file_name()
                    cam.start_preview()
                    p.pushMessage(CHANNEL_NAME, "Motion Detected... Look at mail!")
                    time.sleep(2)
                    cam.capture(fileName)
                    time.sleep(1)
                    SendMail(fileName)#zde volam na metodu SendMail a pridavam ji
                    #atribut filename coz je nazev souboru který se má poslat mailem
                else:
                    cam.stop_preview()
            except KeyboardInterrupt:
                print " System is terminated"
            except Exception:
                print "Nastal Error"
```



```
if __name__ == "__main__":
    main()
```

Hlavní kód pro fotopast, která byla postavena na Raspbianu.

```
import os
import smtplib
from datetime import datetime
from email.MIMEText import MIMEText
from email.MIMEImage import MIMEImage
from email.MIMEMultipart import MIMEMultipart

frm = 'czechbery@gmail.com'
to = 'danielswab@gmail.com'
psw = '*****'
#ImgFileName je promena pro cestu k obrazkum ktere chci poslat
def SendMail(ImgFileName):
    msg = MIMEMultipart()
    msg['Subject'] = 'Motion'
    msg['From'] = frm
    msg['To'] = to

    body = 'Alert!! Motion Detected!!\nTime: %s' % str(datetime.now())
    msg.attach(MIMEText(body, 'plain'))

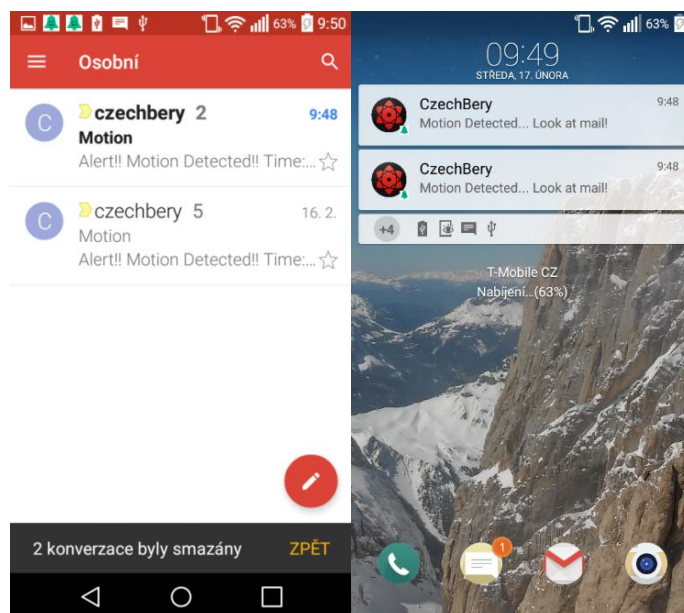
    # msgText = MIMEText('<b>Some <i>HTML</i> text</b> and an image.<br><br>Nifty!', 'html')
    # msgAlternative.attach(msgText)

    fp = open(ImgFileName, 'rb')
    msgImage = MIMEImage(fp.read())
    fp.close()
    msgImage.add_header('Content-Disposition', 'attachment;
filename="image.jpg"$
    msg.attach(msgImage)

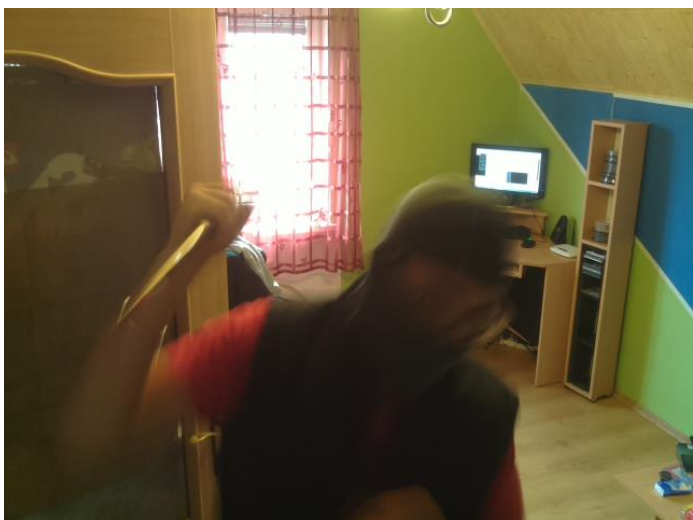
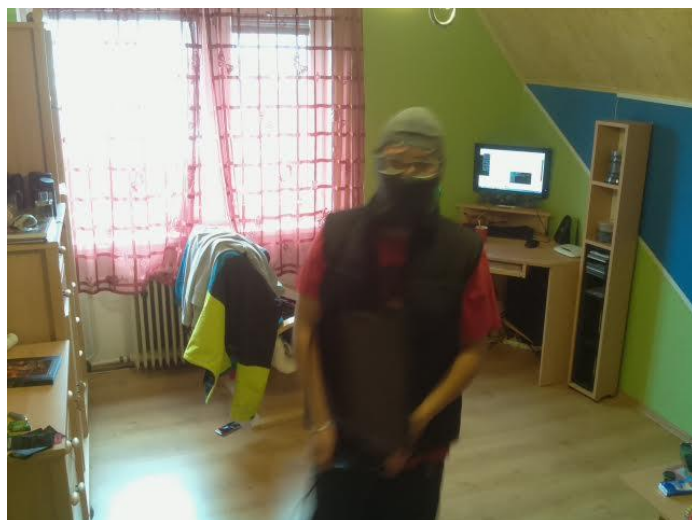
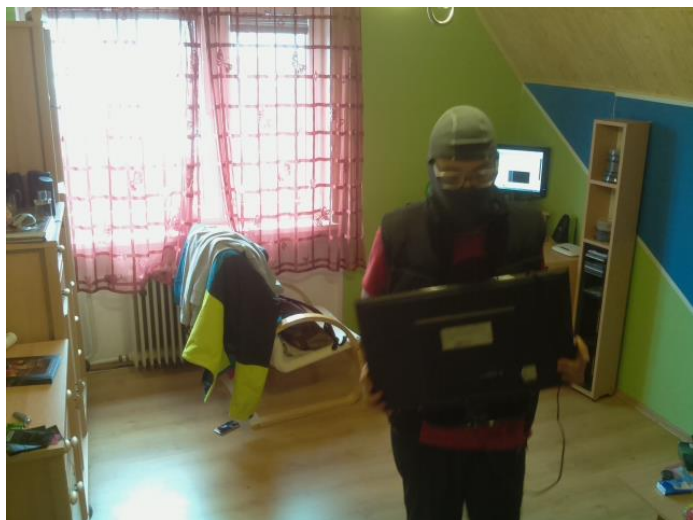
    s = smtplib.SMTP('smtp.gmail.com', 587)
    s.ehlo()
    s.starttls()
    s.ehlo()
    s.login(frm, psw)
    s.sendmail(frm, to, msg.as_string())
    s.quit()
```

Druhý kód pro odesílání mailu z Raspberry Pi, napsáno pro Raspbian

Příloha 2 Ukázka notifikací a emailu v mobilu



Příloha 3 Fotografie "lupiče"



Příloha 2 Testování venku

