

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCA

Brno, 2020

Bc. BARBORA VAŠKOVÁ



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

**KAMEROVÉ ZABEZPEČENÍ OBJEKTU S NÍZKÝM
DATOVÝM TOKEM**

CAMERA SECURITY OF THE OBJECT WITH LOW DATA FLOW.

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Barbora Vašková

VEDÚCI PRÁCE

ADVISOR

Ing. Jiří Přínosil, Ph.D.

BRNO 2020



Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**

Ústav telekomunikací

Studentka: Bc. Barbora Vašková

ID: 186227

Ročník: 2

Akademický rok: 2019/20

NÁZEV TÉMATU:

Kamerové zabezpečení objektu s nízkým datovým tokem

POKYNY PRO VYPRACOVÁNÍ:

V rámci práce navrhnete a realizujete zabezpečení vybraného objektu autonomním kamerovým dohledovým systémem s možností vzdálené správy. Dohledový systém bude sloužit zejména k detekci neoprávněného pohybu v okolí objektu (z obrazu kamery, s možností doplňujících pohybových senzorů) a k zasílání zpráv přes mobilní síť. Požadavkem na dohledový systém je velmi nízký datový tok přes mobilní síť a možnost nezávislého krátkodobého napájení. Součástí řešení je rovněž vytvoření mobilní aplikace pro správu navrženého systému.

DOPORUČENÁ LITERATURA:

[1] HUANG, Shih-Chia. An advanced motion detection algorithm with video quality analysis for video surveillance systems. IEEE transactions on circuits and systems for video technology, 2010, 21.1: 1-14.

[2] BOVIK, Alan C. Handbook of image and video processing. Academic press, 2010.

Termín zadání: 3.2.2020

Termín odevzdání: 1.6.2020

Vedoucí práce: Ing. Jiří Přinosil, Ph.D.

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Téma diplomovej práce sa venuje návrhu a realizácii autonómneho kamerového systému vybraného objektu s možnosťou vzdialenej správy. Náplňou teoretickej časti je spracovať návrh systému vrátane zvolenia vhodných komponentov a ako aj priblížiť používaný softvér. V praktickej časti sa pristupuje k prvotnej inštalácii systému a následné overenie funkčnosti jednotlivých komponentov na základe jednoduchých príkazov. Ďalším krokom je naprogramovanie mobilnej aplikácie a uskutočniť jej komunikáciu o nízkom dátovom toku s kamerovým systémom.

KĽÚČOVÉ SLOVÁ

Autonómny kamerový systém, nízky dátový tok

ABSTRACT

The topic of this master's thesis is devoted to design and realization of autonomous camera system of the selected object with possibility of remote access. The content of the theoretical part is description of components of the camera system and close analysis of the used software, including the selection of suitable components. The practical part is approaching to the initial installation of the system and verification of the functionality of individual components based on simple commands. The next step is developing an mobile application comunicatting with the camera system based on low data flow.

KEYWORDS

Autonomous camera system, low data flow

VAŠKOVÁ, Barbora. *Kamerové zabezpečení objektu s nízkým datovým tokem*. Brno, 2020, 70 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedúci práce: Ing. Jiří Přínosil, Ph.D.

VYHLÁSENIE

Vyhlasujem, že svoju diplomovú prácu na tému „Kamerové zabezpečení objektu s nízkým datovým tokem“ som vypracovala samostatne pod vedením vedúceho diplomovej práce, s využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autora uvedenej diplomovej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto diplomovej práce som neporušila autorské práva tretích osôb, najmä som nezasiahla nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomá následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákonníka Českej republiky č. 40/2009 Sb.

Brno

.....

podpis autorky

POĎAKOVANIE

Rada by som poďakovala vedúcemu diplomovej práce pánu Ing. Jiřímu Přínosilovi, Ph.D. za odborné vedenie, konzultácie, trpezlivosť, ochotu a podnetné návrhy k práci. Zároveň veľká vďaka patrí aj pánu Ing. Pavlovi Maškovi, Ph.D. za spoluprácu pri výbere NB-IoT SIM karty a nastavenie prístupu k VUT servera.

Obsah

Úvod	12
1 Kamerový systém	13
1.1 Požiadavky kamerového systému	13
1.2 Komponenty	14
1.3 Mobilná aplikácia	14
2 Zvolené komponenty kamerového systém	15
2.1 Riadiaci procesor	15
2.1.1 Raspberry Pi 4 Model B	15
2.1.2 Zdroj napájania	17
2.1.3 GPIO	17
2.2 Kamera	18
2.3 Externé čidlo pohybu	19
2.4 Waveshare SIM7020E NB-IoT HAT pro Raspberry Pi	20
2.4.1 NB-IoT SIM karta	20
2.5 Huawei E3372H	21
2.5.1 Mobilný internet	22
3 Softvérová časť	23
3.1 Raspbian	23
3.2 NOOBS	23
3.3 Python	24
3.4 Pícamera	24
3.5 MotionEyeOS	24
3.6 MQTT protokol	25
4 Inštalácie	26
4.1 Inštalácia Raspbian na Raspberry Pi	26
4.1.1 Alternatívne zapojenie Raspberry Pi	26
4.2 Inštalácia kamery	27
4.3 Zapojenie PIR senzora	28
4.4 Inštalácia NB-IoT modemu a AT príkazy	28
4.5 Inštalácia Huawei E3372H	29
4.5.1 Omedzenie priepustnosti dát	30
4.6 Záložný zdroj	31
5 Schéma interiéru	33

6	Zapojenie kamerového systému	34
6.1	Pracovný postup zapojenia Raspberry Pi a jeho komponentov	34
6.2	Testy merania	37
6.3	MQTT Protokol	37
6.4	Meranie dátového toku	39
7	Mobilná aplikácia	41
7.1	Android Studio	41
7.2	Zhotovenie MQTT spojenia	41
7.2.1	Inštalácia	41
7.2.2	Pripojenie	42
7.2.3	Publikácia	43
7.2.4	Subskripcia	43
7.2.5	Odpojenie	44
7.3	Prihlásenie do SurveillanceApp	44
7.3.1	Možnosti aplikácie	45
7.3.2	Nastavenia	46
7.4	Detekcia pohybu	47
7.4.1	Harmonogram detekcie	50
7.5	Upozornenie o výpadku prúdu	52
7.6	Snímka objektu	52
7.6.1	Email	54
7.7	História správ	55
8	Realizácia kamerového systému	57
8.1	Testovanie kamerového systému	58
8.1.1	Meranie spotreby energie	58
8.2	Meranie dátového toku	60
8.3	Cena komponentov	61
	Záver	62
	Literatúra	63
	Zoznam symbolov, veličín a skratiek	66
	Zoznam príloh	67
A	Manuál k mobilnej aplikácii	68
B	Inštalovanie mobilnej aplikácie a zoznam použitých balíkov v RPi	69

Zoznam obrázkov

1.1	Návrh kamerového systému	13
2.1	Raspberry PI 4B [3]	16
2.2	Waveshare (H)[17]	18
2.3	PIR detektor HC - SR501[18]	19
2.4	Waveshare SIM7020E NB-IoT HAT pro Raspberry Pi[19]	20
2.5	Huawei E3372H[20]	22
5.1	Interiér	33
6.1	Vývojový diagram	36
6.2	Príkaz na odfotenie objektu	38
6.3	Príkaz na detekovanie pohybu	38
6.4	MQTT Protokol	39
6.5	Dátový prenos	40
7.1	Prihlásenie do aplikácie	45
7.2	Email/Stiahnutie/Odfotografovať/Nastavenia	46
7.3	Nastavenie	46
7.4	Zapnutie detekcie	48
7.5	Bloková schéma detekcie	49
7.6	Harmonogram detekcie	50
7.7	Blokové schéma harmonogramu detekcie	51
7.8	Bloková schéma žiadosti o fotografiu	53
7.9	História správ	55
7.10	Bloková schéma histórie správ	56
8.1	Fotografia používaných komponentov	57
8.2	Snímky objektu cez deň noc	58
8.3	Graf odmeranej spotreby energie	59
8.4	Zachytený prenos fotografie	60
8.5	Dátový prenos v troch scenároch	61

Zoznam tabuliek

2.1	Špecifikácie	17
6.1	Porovnávanie doby zopnutia pri najmenšej citlivosti	37
6.2	Porovnávanie citlivosti senzora	37
8.1	Tabuľka cien zakúpených komponentov	61

Zoznam výpisov

6.1	Skript pre kameru	35
6.2	Skript pre PIR detektor	35
7.1	Gradle konfigurácia	41
7.2	AndroidManifest.xml	42
7.3	MQTT spojenie	42
7.4	Publikovanie	43
7.5	Subscribe	43
7.6	Metóda pre prenos údajov a otvorenie novej triedy	44
7.7	MQTT spojenie v Python jazyku	47
7.8	Vlákno detekujúce pripojený zdroj	52
7.9	Zaslanie emailu	54

Úvod

Najvhodnejší spôsob ako vhodne zabezpečiť svoj majetok je využiť monitorovaný bezpečnostný systém. Bezpečnostné kamery prešli za posledné roky technologickým pokrokom. Dostali sa postupne aj do sféry bežných ľudí. Na trhu sú dostupné rôzne druhy bezpečnostných kamier, každé z nich majú dlhý zoznam parametrov, funkcií a rôznych cien.

Pri vhodnom výbere kamerového systému je nutné zvážiť jeho primárny účel, cenovú hladinu a technológiu prenosu dat. Dôležitými faktormi pri výbere kamerového setu sú parametre ako:

- Technológia prenosu
- Rozlíšenie
- Uhol záberu
- Konštrukcia kamery
- Záznamové zariadenie

Z mnohých možností výberu je kamerový systém značky *Securia Pro*, napríklad model *NVR4CHV2-W NVR* či *EVOLVEO, model Detective D04FHD*. Ponúkajú kamerový set vhodný do domácnosti v cenovej relácii do 10 000 Kč. Zaznamenajú obraz nepretržite a pri detekcii pohybu je užívateľ notifikovaný. Celé dianie môže užívateľ vzdialene sledovať na mobiloch či iných zariadeniach. Avšak tento systém sa spolieha na vysoký dátový tok dostupný užívateľom. Využívajú či už bezdrátový prenos dat cez WiFi alebo prenos cez ethernetový kábel.

Táto diplomová práca sa zameriava na zhotovenie kamerového systému v ešte nižšej cenovej výške s využitím prenosu dát cez nízky dátový tok, a to pomocou 3G/4G sietí. Komunikácia so systémom je umožnená užívateľovi cez mobilnú aplikáciu, kde si môže nastaviť harmonogram sledovania objektu, želané rozlíšenie snímky či sledovať históriu správ.

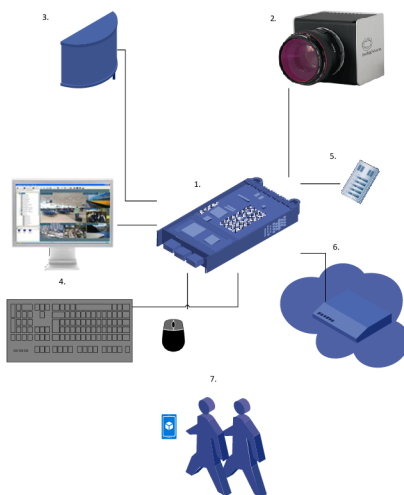
1 Kameraný systém

Vlastnosti zhotoveného kamerového systému spĺňajú základné vlastnosti akéhokoľvek iného kamerového systému s výhodou nenáročnosti na dátový tok a nízkou cenou. Samozrejmosťou je ovládanie systému na diaľku, využitím vybudovanej mobilnej aplikácie. Súčasťou je detekovanie pohybu v snímacej oblasti či už cez deň alebo cez noc, za pomoci IR cut filtra. Tento kamerový systém je navrhnutý na používanie v interiéri alebo v exteriéri. Jeho výhodou je štvormetrová vzdialenosť kamery ako aj externého detektora pohybu od riadiaceho procesora. V prípade výpadku eklektického prúdu v domácnosti bude mať systém v zálohe záložný zdroj na niekoľko hodín. Všetky tieto vlastnosti dopĺňa fakt o malých rozmeroch tohto systému, tým pádom jeho nenápadnosti a nenáročnosti na jeho uskladnenie.

1.1 Požiadavky kamerového systému

Požiadavky na autonómny kamerový systém sú nasledovné:

1. riadiaci procesor
2. kamera + LED osvetlenie
3. externé čidlo pohybu
4. vstupné zariadenia (voliteľné)
5. záložný zdroj (batéria)
6. modem mobilnej siete (3G/4G modem)
7. mobilná aplikácia



Obr. 1.1: Návrh kamerového systému

1.2 Komponenty

Marketingový trh ponúka viaceré možnosti výberu riadiacich procesorov. Je na výber z Arduino, Raspberry Pi, Banana Pi, Orange Pi a iné. Arduino je mikrokontrolér o jednej doske, ktorý dokáže plniť iba jednu úlohu v danú dobu. V porovnaní s inými riadiacimi procesormi je o úroveň nižšie, vhodnejší na jednoduchšie opakovateľné úlohy. Raspberry/Banana/Orange Pi sú viacúčelové dosky obsahujúce vlastný operačný systém, vhodnejší na zložitejšie úkoly. Do diplomovej práce bolo vybrané Raspberry Pi z dôvodu veľkého rozšírenia na trhu, mnohých tutoriálov a cenovej dostupnosti.

Voľba kamery sa odvinula od výberu riadiaceho systému. Raspberry Pi ponúka možnosť pripojenia vlastných kamier cez CSI konektor. Z dôvodu najlepšej kompatibility bola vybraná kamera značky Waveshare. Táto značka ponúka viacero typov kamier. Kamera má slúžiť ako komponent kamerového systému, preto je vhodné, aby mala rozsah záberu čo najväčší. Z tohto hľadiska bol vybraný typ kamery s „Rybíe oko“ šošovkou. Zároveň bolo zvolené, aby kamera ponúkala možnosť nočného videnia. Kamera Waveshare RPi (H) spĺňa požiadavky a práve preto bola vybraná.

Externé čidlo pohybu detekuje pohyb v objekte. Najčastejšie je na výber z pasívnych infračervených čidiel (PIR), mikrovlnných (MW), ultrazvukových detektorov. Neželanou vlastnosťou pri mikrovlnných či ultrazvukových detektorov je ich rozsah aj cez určenú oblasť. Dokážu snímať aj cez steny. To by mohlo vyvolať nežiadúce popluchy v kamerovom systéme. Práve preto bol vybraný PIR detektor, ktorý nedokáže zachytávať pohyb cez steny. Bol zvolený PIR modul HC-SR501, ktorý sa zapojí k Raspberry Pi pomocou troch GPIO pinov.

Záložný zdroj bude vybraný neskôr v diplomovej práci, rovnako ako aj mobilný modem. V semestrálnej práci je Raspberry Pi napájaný adaptérom. Prístup k internetu je možný či už cez ethernet kábel alebo sa využíva WiFi. Neskôr sa vyberie najvhodnejší modem pre nízky dátový tok. Bude vybudovaná mobilná aplikácia umožňujúca ovládanie kamerového systému na diaľku.

1.3 Mobilná aplikácia

Dôležitou súčasťou je mobilná aplikácia. Bude ponúkať niekoľko funkcií nastavenia kamerového systému na diaľku:

- zapnutie/vypnutie kamerového systému
- časový harmonogram snímania objektu
- nastavenie rozlíšenia snímky či videa
- notifikácie o detekovaní pohybu v objekte
- odfotenie objektu a následne jeho zobrazenie

2 Zvolené komponenty kamerového systému

Dohliadací kamerový systém slúži na vzdialenú kontrolu sledovanej oblasti prostredníctvom bezpečnostných kamier a jej komponentov. Kamerové systémy je možné rozdeliť do niekoľko kategórií, a to plno analógový systém, hybridný systém a plne digitálny systém.

- Analógový systém patrí medzi najstaršie kamerové systémy. Pracuje s analógovým video-signalom. Kamera má analógový výstup. Všetky prvky tohto systému sú prepojené koaxiálnym káblom. Obraz je menej kvalitný, medzi jeho štandardy patria PAL a NTSC, avšak už v dnešnej dobe sú pomaly nahradzované štandardnejším typom HD.
- Hybridný systém prináša digitalizáciu analógového signálu z kamier. Tvorí kombináciu analógového a digitálneho systému. Využíva sa najmä pri modernizácii analógových kamerových systémov. Výstup z videorekordéra môže byť analógový alebo digitálny.
- Digitálny systém poskytuje digitálny signál priamo z kamery. Pracuje na princípe číslicového spracovania signálu. Obraz je spracovaný do jedného celku-paketu odoslaný cez bežné počítačové siete, čím sa zvyšuje kvalita obrazu, a zároveň klesá samotná cena systému. Pred odoslaním obrazu z kamery je obraz skomprimovaný.[1]

2.1 Riadiaci procesor

Raspberry Pi vznikol v roku 2012 britskou spoločnosťou Raspberry Pi Foundation s cieľom naučiť deti technického vzdelania, a to ako ovládať rôzne zariadenia - mikrovlnné rúry, automatické pračky, kamery atď.

Tento nízko nákladový jednodoskový počítač porovnateľný so slabším stolným počítačom sa zapája do počítača/ monitoru/ televízie s použitím štandardných vstupov ako klávesnica a myš prepojitelné cez USB. Raspberry Pi (2012) prešlo hardwarovým vývojom od Raspberry Pi po Raspberry Pi 4 (2019), vrátane Raspberry Pi Zero (2017).

Každý z daných druhov má rôzne poddruhy modelov od Model A, Model A+ až Model B+. Distribúcie Linux, Microsoft Windows 10 IoT Core, RISC OS sú použiteľné na tomto počítači. Podporuje programátorské jazyky ako Python či Scratch.

2.1.1 Raspberry Pi 4 Model B

Najnovšia verzia Raspberry Pi, ktorá je prvou plnohodnotnou náhradou počítača je práve Raspberry Pi 4 Model B, ktorý ponúka zvýšenie rýchlosti procesora, multime-

diálneho výkonu, pamäte a konektivity v porovnaní s predchádzajúcimi modelmi.

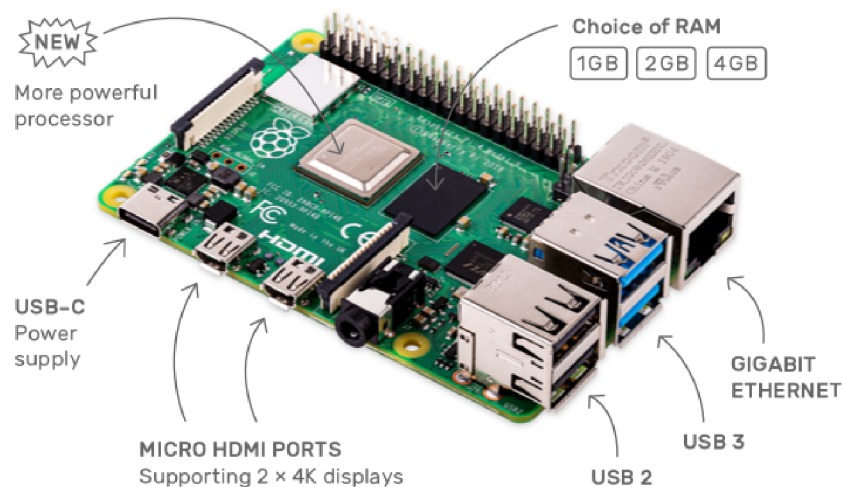
Ponúka možnosť voľby operačnej pamäte v rozmedzí od 1GB RAM cez 2GB RAM až po 4GB RAM. Typ pamäte, ktorý výrazne ovplyvnil jej rýchlosť je LPDDR4.

Základom je 1.5 GHz 4 – jadrový procesor ARM Cortex-A72. Grafický procesor je vybavený VideoCore VI, ktorý prináša rozlíšenie 4K, dekódovanie HEVC/H.265 s režimom dvoch monitorov, avšak na tento dvojitý režim sa nedá nastaviť 4Kp60.

Pôdorys sa zachováva ako v predošlých modeloch, avšak prináša výraznú zmenu v oblasti konektorov a ako napájací konektor využíva USB-C. Štandardný HDMI konektor je nahradený dvoma mikro HDMI konektormi, čo práve umožňuje napojenie dvoch monitorov.

Ethernet je presunutý na opačnú stranu USB konektorov a je zvýšený na gigabajt. Dva USB konektory zo štyroch sú rýchle USB 3.0 porty.[2]

Cena Raspberry Pi 4 v Českej republike sa pohybuje od 998 Kč po 1 536 Kč, závisí od veľkosti RAM.



Obr. 2.1: Raspberry PI 4B [3]

Tab. 2.1: Špecifikácie

Parameter	Raspberry Pi 4 Model B
Procesor	Broadcom BCM2711 (4× Cortex-A72, 1,5 GHz)
RAM	1GB, 2GB alebo 4GB (záleží od modelu)
Bezdrátova konektivita	802.11ac + Bluetooth 5.0
Konektory	2× USB 3.0, 2× USB 2.0, 2× micro HDMI
	Gigabit Ethernet, GPIO
Napájanie	USB-C 5V/3A alebo Power over Ethernet
Hlavné úložisko	microSD karta
Prostredie	Operačná teplota 0-50°C

2.1.2 Zdroj napájania

Raspberry Pi je napájané z USB-C portu s 5V napätím a s minimálnym prúdovým odberom okolo 3A. Za predpokladu, že celkový odber USB periférií neprekročí hranicu 500 mA, môžeme použiť 5V napätie s prúdovým odberom 2.5A. Je možnosť využitia PoE napájanie cez ethernet, ktoré však vyžaduje pridaný modul Raspberry Pi PoE HAT.

Spotreba celkovej elektrickej energie záleží od počtu pripojených periférií. Podľa dokumentácie typicky model B využíva od 700 do 1000 mA. Maximálny odber prúdu, ktorý môže Raspberry Pi čerpať je 1A.

Kamerový modul využíva 250 mA, klávesnica a myš v rozhraní od 100 mA až cez 1000 mA, GPIO piny môžu odoberať 50mA. Jednotlivý GPIO pin môže čerpať okolo 16mA. Spotreba v stave nečinnosti je tým pádom výrazne nižšia, preto pri kúpe modulov treba dbať na ich spotrebu.

2.1.3 GPIO

General-purpose input/output, slovensky Univerzálny vstupný/výstupný pin

Prístupné elektrické kontakty, ktorých využitie je nechané na užívateľovi. Využitie je pestré, od pripojenia LED kontrolky po zariadenia s vysokým odberom. Tieto piny sa využijú pri pripojení PIR detektora.

Raspberry Pi v prvotnej verzii obsahoval 26 pinov, v novších už disponuje so 40 pinmi. Všetky GPIO banky sú napájané 3.3V.

2.2 Kamera

Waveshare RPi kamera (H)

Kamera, ktorá bude využívaná kamerovým systémom je Waveshare RPi (H) kamera pre Raspberry Pi so širokouhlým „rybie oko“ objektívom, nočným osvetlením IR a manuálnym ostrením. Dôvodom výberu tejto kamery bol práve širokouhlý objektív, ktorý je vhodnejší do kamerového systému, ako aj podpora nočného režimu.

Táto kamera podporuje všetky modely Raspberry Pi. Prináša so sebou 5 megapixelový OV5647 senzor. Rozsah, ktorým disponuje je diagonálne 160° a horizontálne 120°. Využíva 1/4" CCD snímač s clonou F2.35. Kamera poskytuje video rozlíšenie 1080p a rozmery modelu sú 25mm × 24mm s 3.3V výstupom.[5]

Možnou nevýhodou je nutnosť špecifického kábla s CSI portom. Tento plochý flexibilný kábel má omedzené dĺžky(5/15/30/50/100/200 cm) čo z hľadiska želanej väčšej vzdialenosti ako dva metre od Raspberry Pi je nevýhodné, preto je nutné použiť prípojku pre CSI konektor, čo umožní navýšiť vzdialenosť aspoň dvojnásobne.

Funkcia denný/ nočný režim je funkciou, ktorá sa pri nízkom osvetlení prepne z farebného režimu kamery do čiernobielyho režimu. Touto zmenou je kamera citlivejšia na infračervené svetlo v porovnaní s farebným režimom.

IR cut filter je umiestnený medzi snímacím čipom kamery a objektívom. Slúži na ovládanie saturácie pri striedaní dňa a noci. Pri dennom svetle nie je potrebné infračervené svetlo, preto sa filtruje cez IR cut filter. Pri prepnutí na čiernobiely režim sa odsunie IR cut filter pred čipu kamery.

IR LED diódy sa zapájajú buď priamo do kamery alebo sú externe pripojené pri kamere. Napomáhajú v nočnom režime osvetliť objekt, čo zlepšujú dosah kamery v noci. CCD snímač je citlivý na infračervené svetlo na rozdiel od ľudského oka, pre ktoré je neviditeľné, takže narušiteľ v nočných hodinách si kameru s IR osvetlením nepovšimne.



Obr. 2.2: Waveshare (H)[17]

2.3 Externé čidlo pohybu

PIR detektor pohybu HC-SR501

Jadro detektora pohybu PIR je pyroelektrický snímač, v ktorom sa nachádzajú dve pyroelektrické dosky. Tieto dosky sú zapojené rozdielne, čo eliminuje rušivé javy, ako zmena teploty vzduchu a mechanické otrasy.

Snímaná oblasť je rozdelená šošovkami na dva laloky a slepú zónu. Na prvú doštičku dopadá žiarenie z prvého laloku, na druhú doštičku dopadá žiarenie z druhého laloku. Intenzita žiarenia na oboch lalokoch je porovnávaná. Pokiaľ je rovnaká, napätie na oboch doštičkách je rovnaké a napätie na rezistoroch je nulové. Pokiaľ je zachytený pohyb na prvom laloku, tak intenzita na tomto laloku je navýšená, čo spôsobí nerovnováhu a je vyvolaný poplach.

Detektory PIR zachytávajú pohybujúce sa objekty o teplote ľudského tela. Najcitlivejšia oblasť detektora je keď sa objekt pohybuje v smere kolmo k detektoru alebo od neho. Pokiaľ sa daný objekt pohybuje pomaly, napätie na doštičkách sa zvyšuje pomaly, čo nevyvolá poplach.

Na našom PIR detektore môžeme nastaviť citlivosť až do 7 metrov a dobu zopnutia v rozmedzí od 5s do 200s, pričom uhol detekcie je do 120°. Citlivosť a doba zopnutia je nastaviteľná potenciometrom.

Pokiaľ je zachytený pohyb, objaví sa na výstupnom pinu logická jednotka - 3.3V. Po vypršaní nastaviteľného oneskorenia, zostane PIR ešte 2.5 sekundy v tvare logickej nuly, v tomto stave nemôže byť zachytený žiaden ďalší pohyb snímaného objektu.

Tento model ponúka dve možnosti režimu detekcie pohybu. Prvým je začatie odpočítavania časového oneskorenia hneď po zachytení pohybu, pričom k reštartovaniu časovača dôjde až po vypršaní nastaveného času. V druhom režime je časovač reštartovaný pri každom zachytenom pohybe.[6]

V kludovom stave je prúdový odber menší ako 50 mikro ampérov a napájacie napätie sa pohybuje od 5 po 20V. Rozmery detektora sú 32×24×24 mm.

Umiestnenie detektora musí byť dôsledne umiestnené. Nie je vhodné, aby detektor bol umiestnený proti oknu a zdroja svetla, z dôvodu vyvolania možného falošného poplachu. Ideálne je umiestniť ho tak, aby mal výhľad na celý strážený objekt.

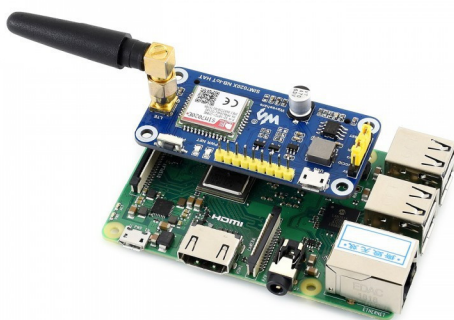


Obr. 2.3: PIR detektor HC - SR501[18]

2.4 Waveshare SIM7020E NB-IoT HAT pro Raspberry Pi

NarrowBand-Internet of Things HAT pre Raspberry Pi je kontrolované AT príkazmi a podporuje protokoly ako LWM2M/COAP/MQTT/TCP/UDP/HTTP/HTTPS a mnohé ďalšie. Tento modem využíva pri aplikáciách internetu vecí, ako sledovanie majetku, ovládanie domácnosti a iné. Je založený na SIM7020E a B1/B3/B5/B8/B20/B28 pásmach. Jeho veľkou výhodou je nízke opozdenie, nízky prúdový odber a nižšia cena. Práve tieto vlastnosti rozhodli pri voľbe modemu pre kamerový systém. V porovnaní s modemami, ktoré podporujú 3G/4G je cena až NB-IoT HAT až trojnásobne nižšia.

Modul je kompatibilný s každou verziou Raspberry Pi. Využíva USB rozhranie ako zdroj napätia aj pre ladenie modulu. Podporuje NB-IoT SIM kartu. Využíva dva LED identifikátory stavu. Rýchlosť odosielania dát sa pohybuje okolo 62.5Kbps a rýchlosť sťahovania dát je 26.15Kbps. Celkový prúdový odber v stave nečinnosti je 18mA.



Obr. 2.4: Waveshare SIM7020E NB-IoT HAT pro Raspberry Pi[19]

2.4.1 NB-IoT SIM karta

Na trhu je viacero možností ako si zabezpečiť NB-IoT kartu. V prvom rade je potrebné zistiť či zariadenie vyžaduje MicroSIM alebo NanoSIM. Následuje odhad spotreby dát. Podľa toho si užívateľ volí taríf. V Českej republike tieto SIM karty ponúka firma *Vodafone*, *T-Mobile* a *Zooco*. Cenu týchto kariet je možné zistiť až po kontakte s firmou.

Taktiež je možnosť získať kartu zo zahraničia. Firma *Things Mobile* ponúka zakúpenie jej SIM karty spolu s ďalšími možnosťami. Ponúka portál pre kontrolu spotreby dát. Spolupracuje s vyše 350 roamingovými partnermi, takže zabezpečuje kvalitné pripojenie. Česká republika sa nachádza v Zóne 1 čo znamená 0.12€/MB. Ponúka rozličné balíky. Napríklad 10MB za 1.10€ či 100MB za 7.50€.

Na podobnom princípe funguje firma *IoT*, ktorá si účtuje 0.05€/MB. Disponuje taktiež portálom a jej výhodou môže byť objednanie testovacej SIM karty.

Firma *INCE* ponúka NB-IoT SIM kartu až na 10 rokov s 500MB. Výhodou je predplatenie tejto karty, jednorázové nastavenie a užívateľ to môže nechať bežať bez ďalšieho nastavovania. Cena za túto kartu je 10€. Jednou nevýhodou je pomerne vysoké poštovné a to okolo 7€.

V prípade tohto kamerového systému, bola NB-IoT MicroSim karta vypožičaná od školy, ktorá spolupracuje s firmou Vodafone.

2.5 Huawei E3372H

USB modem Huawei E3372H je vylepšenou verziou modemu Huawei E3372, podporuje 4G LTE o rýchlosti 150 Mb/s pri sťahovaní a 50 Mb/s pri odosielaní. V prípade potreby disponuje aj 3G (HSPA+ 21,6 Mb/s a HSUPA 5,76 Mb/s) či 2G. Je vybavený internou anténou a konektorom na externú anténu. Modem je vhodný pre operačné systémy ako sú WindowsXP, Vista, Windows 7 / 8, MAC, Win Blue a má zabudované sloty pre MicroSIM kartu. Cena tohto modemu sa pohybuje okolo 1000 Kč.

Po zapojení modemu do USB konektora sa zobrazí jeho obsah s inštalačným súborom. V prípade operačného systému Raspbian, tento súbor nie je možné nainštalovať. Postup ako prepojiť modem s Raspberry Pi bude v ďalšej kapitole rozobrané. Modem signalizuje stav pripojenia cez farebný indikátor, možné stavy sú:

- zelená bliká dvakrát za 2 sekundy - USB modem je napájané
- zelená bliká jedenkrát za 0,2 sekundy - prebieha aktualizácia softvéru
- zelená bliká raz za 2 sekundy - USB je registrované v sieti 2G
- modrá bliká raz za 2 sekundy - USB je registrované v sieti 3G
- azúrová bliká raz za 2 sekundy - USB je registrované v sieti LTE
- zelená svieti raz za 2 sekundy - USB je pripojené v sieti 2G
- modrá svieti raz za 2 sekundy - USB je pripojené v sieti 3G
- azúrová svieti raz za 2 sekundy - USB je pripojené v sieti LTE



Obr. 2.5: Huawei E3372H[20]

Na trhu je dostupný ešte 3G modem značky D-Link (dwm-157) v cene nižšej kategórii, avšak pri výbere sa prihliadalo na recenzie s kompatibilitou operačného systému Raspian. Mobilní operátory ponúkajú dátové balíčky rovno s 3G/LTE modemom. Je možné zakúpiť modem už rovno aj s tarifom podľa aktuálnych ponúk.

2.5.1 Mobilný internet

Mobilný operátor O2 ponúka balík zahrňujúci tri mesiace internetu zdarma spolu s modemom Huawei E3372H, bez záväzku v cene okolo 1105 Kč. Po troch mesiacoch si klient môže nabiť kredit alebo nemusí. Tento balíček bol vybraný na testovanie kamerového systému. Mesačne je možné minúť až 1,5 GB dát.

Množstvo dát, ktoré by užívateľ minul mesačne je podstatne menej. V bežnom používaní by sa pohyboval v rozmedzí od 10 MB do 20 MB. Záležalo by na intenzite odosielať fotografie s vyšším rozlíšením.

Mobilní operátory ako O2 a T-mobile ponúkajú dobíjanie kreditu s nižším objemom dát. Konkrétne mobilný operátor O2 má taríf *Machine*, bez záväzku. Klient si platí mesačne za 10 MB 36,30 Kč, poprípade za 20 MB 42,35 Kč.

Mobilný operátor T-Mobile ponúka objem dát 60 MB za cenu 29 Kč alebo 100 MB za 39 Kč.

3 Softvérová časť

Táto kapitola zahŕňa softvéry, ktoré sú nevyhnutnou časťou kamerového systému. Zahŕňa potrebný operačný systém Raspberry Pi, ako aj programovací jazyk, v ktorom bude písaná mobilná aplikácia. Opisuje komunikačný protokol zabezpečujúci obojstrannú komunikáciu pomocou vzdialeného servera.

3.1 Raspbian

Oficiálny systém používaný všetkými modelmi Raspberry PI je nazývaný Raspbian, ktorý je založený na distribúcií linuxového Debian Buster. Bol vytvorený ako nezávislý projekt od Mike Thompson a Peter Green, avšak od roku 2015 je oficiálne podporovaný Raspberry PI Foundation ako primárny operačný systém jednodoskových počítačov Raspberry. Existuje niekoľko verzií Raspbian ako Raspbian Buster alebo Raspbian Scratch.

Raspbian prichádza s predinštalovanými balíčkami pre vzdelávanie, programovanie a všeobecné využitie. Obsahuje Python, Scratch, Sonic Pi ako aj Javu a mnohé iné. Po stiahnutí získame Raspbian v ZIP formáte. Dosahuje veľkosť cez 4GB. Je voľne dostupný.

Prvá verzia vznikla v roku 2012. Pri písaní tejto semestrálnej práce bola vydaná verzia 2019-09-26, s ktorou pracuje aj naše Raspberry 4B. Užívateľ ho môže stiahnuť cez oficiálnu stránku Raspberry Pi alebo použiť NOOBS, čo značí jednoduchú inštaláciu tohto operačného systému pre začiatočníkov.

Raspbian používa PIXEL (Pi Improved X-Windows Environment Lightweight), ako hlavné desktopové prostredie skladajúce sa z modifikovaného prostredia LXDE a správcu okien Openbox.[7]

3.2 NOOBS

Začiatočníkom sa odporúča začať práve s NOOBS (New Out Of Box Software). Je ponúkaný práve v predinštalovanej forme na SD karte alebo stiahnuteľný na oficiálnych stránkach od Raspberry Pi, kde je následné príručka k manuálu inštalácie. NOOBS je jednoduchý inštalátor operačného systému Raspbian a Libre ELEC (Libre Embedded Linux Entertainment Center), voľno dostupná linuxová distribúcia pre multimediálny softvér Kodi. Ponúka taktiež selekciu ďalších alternatívnych operačných systémov. V písaní semestrálnej práce bola najnovšia verzia 3.2.1 s dátumom 2019-09-30. NOOBS Lite obsahuje rovnaký operačný systém bez predinštalovaného Raspbian. V písaní diplomovej práce bola najnovšia verzia 3.2 s dátumom 2019-07-10.[8]

3.3 Python

Založený v roku 1991 Guidom van Rossumom a vyvinutý Python Software Foundation je interaktívny programovací jazyk Python. Myšlienkou je dôraz na čitateľnosť kódu a jeho syntax, čo umožňuje programátorovi vyjadriť sa v menšom počte riadkov kódu. Python umožňuje rýchly a stručný zápis kódu, čo zefektívňuje celkovú prácu programátora.

Vyvíja sa ako voľne dostupný projekt. Existuje niekoľko verzií, a to Python 2 a Python 3. V súčasnosti sa používa verzia 3.7.0. Podporuje objektovo orientované, štruktúrované a funkcionálne programovanie. Využíva niekoľko vysokoúrovňových dátových typov a na správu pamäte používa garbage collection. Je využívaný mnohými spoločnosťami ako Google či BitTorrent. Výhodou tohto programovacieho jazyka je jeho jednoduchosť rozšírenia, využitím rôznych modelov, knižníc, balíčkov, napísaných napríklad v C alebo C++ jazyku. [9]

3.4 Pcamera

Knižnica pre Python umožňujúca ovládanie kamery pre Raspberry Pi je Pcamera, vyvinutá Davom Jamesom. Umožňuje zachytávať statický obraz ako aj video a spracovať ich neskôr. Poskytuje pestré programové API. Táto knižnica je podporovaná verziami Python 2.7(a vyššie) a Python 3.2(a vyššie). [10]

3.5 MotionEyeOS

Voľne dostupný produkt, ktorý pomocou Raspberry Pi alebo iných jednodoskových počítačov vytvorí sofistikovanú sieť s pripojenou kamerou je MotionEyeOS. Samotný kód je dostupný na GitHub. Obsahuje image súbor, ktorý je nutné nahráť na MicroSD kartu a následne vložiť do Raspberry Pi. Tento produkt beží nezávisle. Stačí sa pripojiť na IP adresu Raspberry Pi pre MotionEye, pričom daný softvér ponúka vlastnosti ako:

- Jednoduchú inštaláciu kamery
- Webové rozhranie užívateľsky prístupné
- Ponúka pripojenie niekoľko kamier naraz
- Obsahuje detektor pohybu s nastaviteľnou emailovou notifikáciou
- Možnosť snímania videa, nahrávania na SD kartu, Google Drive či Dropbox
- Možnosť prístupu cez FTP server či SFTP server

S rozmanitou ponukou softvér ponúka vhodnú alternatívu kamerového systému s jednoduchou inštaláciou. [16]

3.6 MQTT protokol

MQTT je skratka pre *MQ Telemetry Transport*. Vznikol v roku 1999 doktorom Andy Stanford z firmy IBM a Arlenom Nipperom z firmy Eurotech. Dnes patrí pod správu združenia OASIS.

Protokol je založený na TCP/IP a navrhnutý pre jednoduché zariadenia s úzkou šírkou pásma a pomerne vysokou latenciou. Základ tohto protokolu je minimálne zaťaženie zariadenia ako aj minimálny požiadavok na zdroj zariadenia. Práve preto je ideálnym protokolom k využitiu *machine to machine* zariadení a *Internet of Things*, ktoré práve tieto podmienky vyžadujú.

Základ komunikácie spočíva na typu zverejniť/odoberať (*publish/subscribe*) smerom do jedného servera (*MQTT Broker*). Jednotlivé správy klienti publikujú do tém (*Topics*), ktoré server rozpošle jednotlivým klientom, ktorí sú prihlásení na ich odber (*subscribers*). Klientom môže byť od webovej služby cez mobilnú aplikáciu po koncové zariadenia ako sú napríklad mikropočítače. Zabezpečenie správ je formou užívateľského mena a hesla.[15]

4 Inštalácie

V tejto kapitole sa nachádzajú postupy potrebné na správnu inštaláciu či už hardwardovej či softvérovej časti kamerového systému.

4.1 Inštalácia Raspbian na Raspberry Pi

Na úspešnú inštaláciu Raspbianu na RPi je nutnosť:

- SD karty
- Druhý počítač
- Stiahnutý image súbor

Na oficiálnej stránke Raspberry Pi je voľne dostupný operačný systém Raspbian. Užívateľ si môže vybrať či chce NOOBS alebo samostatný Raspbian. Stiahnutý image je v zip formáte. Pomocou voľne dostupného softvéru balenaEtcher je možné zazipovaný image nahráť na SD kartu. Stačí si vybrať, ktorý zip súbor má nahráť na správne zvolenú SD kartu a po kliknutí na Flash tlačítka sa po niekoľkých minútach nahraje operačný systém Raspbian na SD kartu.

Ďalším krokom je vložiť SD kartu s nahraným Raspbianom do Raspberry Pi. Ideálne je pripojiť k Raspberry externý monitor cez microHDMI port, externú myš a externú klávesnicu cez USB porty. Alternatívou je tzv. *headless* pripojenie, ktoré sa neskôr rozoberie v ďalšej podkapitole. Dôležitým prvkom je pripojiť zdroj napätia cez microUSB.

Po úspešnom zapojení komponentov sa na obrazovke zobrazia štyri loga Raspberry PI, čo znamená načítavania OS. Po niekoľkých minútach (pri prvom spustení) sa zobrazí Desktop Raspberry. Zobrazí sa okno s nastavením aktuálnej polohy, WiFi pripojenie. Následne je želaný reštart RPi čo ukončí inštaláciu systému.[11]

Zadaním do príkazového riadku

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

sa získa najnovšia verzia Raspbianu.

4.1.1 Alternatívne zapojenie Raspberry Pi

Vzdialený prístup k RPi umožňuje ovládanie tohto jednodoskového počítaču z prístupu druhého počítača na rovnakej sieti pomocou SSH, čo umožní používať príkazový riadok RPi. Pre prístup k vzdialenej obrazovke slúži VNC.

Prvým krokom je nutnosť poznania IP adresy RPi. Zadaním

```
ipconfig
```

do príkazového riadku displej zobrazí používanú IP adresu, alternatívou je príkaz

```
hostname -I
```

na zobrazenie pridelených IP adres.

Druhým krokom je povoliť SSH server v „*Raspberry Preferences- Interfaces- Enable vedľa SSH*“ alebo zadať príkazy do príkazového riadku

```
sudo systemctl enable ssh
```

```
sudo systemctl start ssh
```

V prípade *headless* zapojenia je možné vytvoriť prázdny súbor s názvom *ssh* bez prípony a uložiť ho do *boot partition* SD karty. Takto pri spustení PI sa povolí SSH automaticky.

Tretím krokom, je zvoliť vhodného klienta pre SSH, ako je napríklad voľne dostupný server PuTTY. Po jeho stiahnutí a otvorení sa zadá pridelená adresa RPi a otvorí sa spojenie. Užívateľ je žiadaný zadať používateľské meno a heslo, ktoré zadával pri konfigurácii Raspbianu. Po úspešnom prihlásení sa zobrazí príkazový riadok zo vzdialeného RPi.

Štvrtým krokom je prepojenie vzdialenej plochy pomocou VNC Viewer, ktorý je tiež voľne dostupný softvér. V príkazovom riadku zadáme

```
sudo apt update
```

```
sudo apt install realvnc-vnc-server realvnc-vnc-viewer
```

pre použitie najnovšej verzie softvéru. Následne treba povoliť VNC a to následovne *Menu - Preferences - Raspberry Pi Configuration - Interfaces - VNC - Enabled*. V príkazovom riadku sa spustí

```
vncserver
```

Zobrazenú adresu serverom sa zadá do VNC Viewer okna pre naviazanie spojenia.[14]

4.2 Inštalácia kamery

Kameru pripojíme kamerovým CSI káblom do slotu nachádzajúceho sa medzi microHDMI a ethernetovým portom. Po správnom zapojení je možné spustiť Raspberry Pi.

Následne sa napíše do príkazového riadku

```
sudo raspi-config
```

Otvorí sa okno s možnosťou výberu *Interfacing Options*, kde sa zvolí *Camera* a vyberie sa možnosť *Enable*. Raspberry sa reštartuje. Týmto nastavením sa nastaví správny GPU firmware, ktorý pobeží s kamerovým ovládačom.

Na otestovanie funkčnosti kamery sa do príkazového riadku zadajú príkaz ako

```
raspistill -v -o test.jpg
```

Zobrazí sa päť sekundový náhľad z kamery a odfotí sa snímka. Zadaním príkazu

```
raspivid -o video.h264 -t 10000
```

sa zaznamená video o dĺžke 10000 milisekúnd.[12]

4.3 Zapojenie PIR senzora

Pre správne zapojenie PIR senzora je potrebné mať tri samička - samička (Female-Female) prepojkky, aby bolo možné jednotlivé piny prepojiť s Raspberry Pi. Na module sú 3 piny a to pre napájanie modulu VCC(1) a GND(3). V strede sa nachádza výstupný pin OUT(2), ktorý pri zaznamenaní pohybu dáva logickú jednotku. Toto označenie sa nachádza za Fresnelovou šošovku.

Model obsahuje dva odporové trimre. Prvým sa určuje citlivosť snímača, ktorá podľa výrobcu dosahuje siedmych metrov. Druhým trimrom sa nastavuje doba čakania časovača, kým prejde zo stavu logickej jednotky do logickej nuly. Časové rozpätie je možné nastaviť od 0.3 sekúnd do piatich minút.

Modul VCC sa napája na 5V pin na doske Raspberry Pi, umožní to napájanie PIR senzora. Modul GND sa pripája k pinu GND na RPi doske, čo uzatvorí okruh. Modul OUT môže byť napojený na ľubovoľný pin na RPi, napríklad na GPIO 4. Pri detekovanom pohybe senzor vyvolá napätie, čo následne RPi zachytí a vyvolá sa želaná funkcia.[13]

4.4 Inštalácia NB-IoT modemu a AT príkazy

Pre správne zapojenie tohto modemu do RPi, je potreba použiť sériové rozhranie UART (Universal Asynchronous Receiver Transmitter) a v operačnom systéme vypnúť logovanie kernelu. Na to nám slúžia príkazy:

```
sudo raspi-config
```

V nastavení vybrať -> *Interface Options* -> *Serial* -> *Yes* a následne *No*

```
sudo cd /boot/cmdline.txt
```

V tomto súbore je nutné vymazať *console=ttyS0,115200*

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

Vložiť `enable_uart=1` na konci súboru.

Po týchto zmenách je potrebné reštartovať zariadenie.

Následne napríklad použitím *PuTTY* a SSH sa môžeme pripojiť na modem cez sériové rozhranie */dev/ttyS0* s rýchlosťou 9600.

Modem je ovládateľný AT príkazmi. Po úspešnom pripojení SSH, je potrebné overiť komunikáciu zadaním AT do príkazového okna. Pokiaľ modem spolupracuje vráti spätnú odpoveď AT, OK.

Následne je potrebné nastaviť MicroSIM kartu postupným zadávaním príkazov, ktoré sa nachádzajú v manuále. Jednotlivé hodnoty sa líšia od operátorov. Dôležitými príkazmi sú nastavenie šírky pásma pre NB-IoT a APN operátora.

Pre overenie komunikácie s operátorom slúži príkaz `AT+CGDCONT?`, ktorý vypíše priradenú APN, IP adresu, poprípade defaultnu bránu a DNS server.

Pre spojenie s MQTT brokerom slúžia tieto príkazy:

```
AT+CMQNEW=0, "test.mosquitto.org", "1883", 12000, 100
```

Príkaz pre zhotovenie nového MQTT spojenia, kde nula značí jeho ID. Zadáva sa IP adresa servera, DNS názov, port servera, AT čas vypršania spojenia, buffer veľkosť

```
AT+CMQCON=0, 3, "client", 6480, 0, 0, "nbiot", "nbiot"
```

Zaslanie spojovacieho paketu. Týmto paketom sa zasielajú prihlasovacie údaje, ako aj dĺžka doby života.

```
AT+CMQSUB=0, "TestTopic", 0
```

Subskripcia na tému *TestTopic*.

```
AT+CREVHEX=0
```

Príkaz pre zasielanie *RAW* dát.

```
AT+CMQPUB=0, "TestTopic", 1, 0, 0, 4, "Ahoj"
```

Publikácia správy *Ahoj*, kde je dôležitá posledná hodnota, ktorá určuje dĺžku správy. V tomto prípade sa jedná o číslo 4, keďže ahoj má 4 písmena.

```
+CMQPUB: 0, "TestTopic", 0, 0, 0, 4, "Ahoj"
```

Zobrazená odchytená správa *Ahoj*.

Veľkou výhodou tohto zariadenia je nízka spotreba a cena. Práve preto sa prvotne vybral tento modem. Počas implementácie AT príkazov do Python jazyka sa narazilo na omedzené používanie týchto príkazov. Pri publikovaní správy je limitovaná veľkosť posielania bytov na hodnotu 1000, čo v prípade posielania fotografie o väčšej veľkosti by bolo nutné ju fragmentovať niekoľkonásobne. Z tohto dôvodu bolo opustené z využívania NB-IoT modemu a prešlo sa na 3G/4G USB dongle.

4.5 Inštalácia Huawei E3372H

Nainštalovanie modemu prebieha prvotným nainštalovaním balíčka *ppp* pre nastavenie komunikácie modemu s Raspberry Pi.

```
sudo apt-get install ppp
```

Jednou z možností pre pripojenie modemu k internetu je použiť program *Sakis3G*. Samotný program je inteligentný a sám si dokáže prečítať internetové nastavenia na SIM karte.

```
sudo ./sakis3g connect
```

Jedinou vecou, ktorú si užívateľ musí nastaviť je názov prístupového bodu operátora. Tieto informácie sa dajú zistiť na internetových stránkach mobilných operátorov. V tomto prípade pre mobilného operátora O2 sa volí APN *internet*. Navigačné menu sprevádza užívateľa, vyzve ho k zadaniu PIN a následne pripojí modem k internetu.

Ďalšou alternatívou ako pripojiť modem k internetu je program *wvdial*, avšak tento spôsob je menej intuitívny. Po nainštalovaní balíka je potrebné prispôbiť konfiguračný súbor vybranému modemu. Dôležité je zmeniť hodnoty *APN*, poprípade *Phone* a *Modem* [21].

```
sudo apt-get -y install wvdial
cat /etc/wvdial.conf

[Dialer Defaults]
Init1 = ATZ
Init2 = ATE1
Init3 = AT+CGDCONT=1,"IP","internet"
Stupid Mode = 1
MessageEndPoint = "0x01"
Modem Type = Analog Modem
ISDN = 0
Phone = *99#
Modem = /dev/ttyUSB1
Username = { }
Password = { }
Baud = 460800
Auto Reconnect = on

sudo wvdial
```

4.5.1 Omedzenie priepustnosti dát

Ak si užívateľ bude platiť za objem dát, vhodným riešením je zamedziť všetkým aplikáciám prístup na internet, okrem skriptu kamerového systému. Týmto sa eliminuje neželané používanie dát. Jedným zo spôsobom je využitie firewallu. Môže sa použiť buď už nainštalovaný firewall v operačnom systéme - paketový filter, alebo nainštalovať jednoduchý balíček *ufw*, ktorého použitie je veľmi jednoduché.

```
sudo apt install ufw
```

Pri nastavovaní firewallu je potrebné myslieť vopred, v prípade používania SSH a VNC pre pripojenie k Raspberry Pi, ako to bolo v tomto prípade. Pravidlo *allow* povoľuje a pravidlo *deny* zakazuje portom, IP adresám k odosielaniu/ príjmu dát.

```
sudo ufw allow in from 34.250.16.30 port 13237 to any
sudo ufw allow out from any to 34.250.16.30 port 13237
ufw default deny incoming
ufw default deny outgoing
```

```
sudo ufw enable
```

```
sudo ufw status
```

```
Status: active
```

To	Action	From
--	-----	----
Anywhere	ALLOW	34.250.16.30 13237
22/tcp	ALLOW	Anywhere
5901/tcp	ALLOW	Anywhere
22/tcp (v6)	ALLOW	Anywhere (v6)
5901/tcp (v6)	ALLOW	Anywhere (v6)
34.250.16.30 13237	ALLOW OUT	Anywhere
DNS	ALLOW OUT	Anywhere
465/tcp	ALLOW OUT	Anywhere
DNS (v6)	ALLOW OUT	Anywhere (v6)
465/tcp (v6)	ALLOW OUT	Anywhere (v6)

IP adresa MQTT brokera je *34.250.16.30* a používaný port je *13237*. Vytvorené pravidlo umožňuje všetkým aplikáciám komunikovať s danou IP adresou o danom porte, obojsmerne. Port *465* a DNS umožňujú posielat email. Iná komunikácia je zakázaná. Týmito pravidlami je povolená komunikácia iba s MQTT brokerom.

4.6 Záložný zdroj

Pre výber správneho záložného zdroja je potrebné určiť prúdový odber celého kame-
rového systému. Tieto informácie je možné získať v popisoch jednotlivých zariadení.
U Raspberry Pi 4B je písaný odber prúdu od 700 do 1000mA, typicky má však okolo
600mA Všetko záleží od pripojených komponentov.

PIR senzor ma prúdový odber 65mA, kamera až 250mA, 4G dongle do 100mA.

$$I = 600 + 64 + 250 + 100 \quad (4.1)$$

$$I = 1014mA \quad (4.2)$$

V prípade veľkej záťaže, je vhodné pripočítat aspoň 20% z celkového prúdu.

$$I = 1,217A \quad (4.3)$$

Po určení celkového prúdu si užívateľ volí, koľko hodín bude potrebovať funkčnosť záložného zdroja. Nami volený záložný zdroj bude fungovať ako záloha v prípade výpadku elektrického prúdu. Môžeme rátať s nutnosťou 4 hodín a viac. V tomto prípade:

$$1,217 * 4 = 4,867AH = 4867mAh \quad (4.4)$$

Došlo sa k záveru, že záložný zdroj ma mať kapacitu minimálne 4,9 AH.

Bolo na výber z dvoch možností výberu záložného zdroja k Raspberry Pi. Jednou z nich bola kúpiť nabíjateľnú batériu o kapacite minimálne 4AH. K tejto batérie by bolo potrebné zakúpiť Li-on Battery Hat, ktorá podporuje obojsmerné nabitie. Rovnako zaistuje ochranu pred nadmerným nabitím/vybitím, prepätím a skratu. Celková cena by sa pohybovala od 500 Kč, záležalo by na cene batérie.

Druhou možnosťou je zvoliť vhodnú powerbanku. Taktiež musí vybraná powerbanka disponovať obojsmerným nabíjaním. Nutnosťou je, aby nabíjala zariadenie s prúdom 3A, keďže Raspberry Pi 4B je napájané z USB-C s 5V napätím a minimálnym prúdovým odberom 3A.

Pre náš účel bola zvolená powerbanka Avacom PWRB-100K-PQ o kapacite 10AH. Avšak, k tejto banke bol samostatne dokúpený kábel KB AXAGON BUCM3-AM05G, SUPERSPEED kábel USB-C <-> USB-A 3.2 Gen 1, 0.5m, 3A. Tento kábel zabezpečuje 3A prívod do zariadenia. Celková cena spolu s káblom konala 712kč.

Pokiaľ po prvotnom otestovaní všetko funguje, je vhodné nastaviť, aby spustenie skriptu kamerového systému začalo hneď po reštarte zariadenia. Tento reštart zariadenia nastane pri výpadku elektrického prúdu, prebehne zmena zdroja.

V cmd spustíme *crontab -e*. Pri prvom použití sa volí typ editora, najvhodnejší je *nano* editor. Na konci editora vložíme cestu ku skriptu, ktorý sa má spustiť hneď pri reštarte. Uložíme *CTRL + O* a *CTRL + X* a reštartujeme zariadenie.

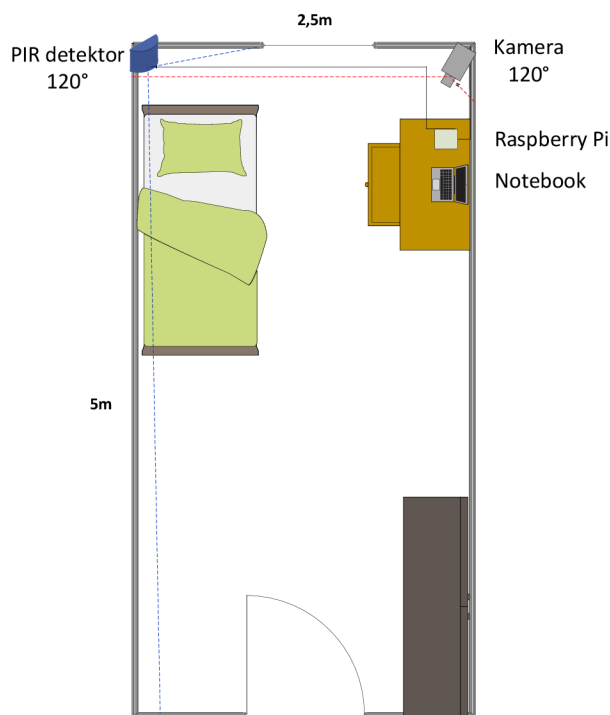
```
pi@raspberrypi ~ $ sudo crontab -e
@reboot python3 home/pi/Desktop/Scripty/MQTTv3.py
pi@raspberrypi ~ $ sudo reboot
```

Otestujeme spustenie skriptu cez mobilnú aplikáciu.

Do záložného zdroja je vedená microUSB rozdvojka. Táto rozdvojka je napájaná klasickým 5V zdrojom pre Raspberry Pi pomocou adaptéra z USB-C na microUSB. Prvý koniec rozdvojky slúži pre napájanie záložného zdroja. Druhý koniec rozdvojky bol upravený tak, aby vodiče 5V a GND boli privedené do GPIO pinov. Nutnosť je použiť delič napätia z 5V na 3,3V. Ak by sa tak nestalo, 5V privedených na pin by mohlo poškodiť samotný pin alebo aj celé zariadenie.

Pri výpadku prúdu sa zmení hodnota pinu z High na Low. V takomto prípade kamerový systém oznámi užívateľovi výpadok elektrického prúdu.

5 Schéma interiéru



Obr. 5.1: Interiér

Kamerový systém je umiestnený v priváte na Kráľovom Poli v Brne. Rozmery izby sú 5×2.5 metra. Nábytok nachádzajúci sa v izbe je posteľ v ľavom rohu, pri okne je stôl a stolička. Napravo od dverí je väčšia skriňa. Kamerový systém je rozvrhnutý nasledovne.

Kamera sa nachádza v pravom hornom rohu na úrovni okna, vo výške 2.5 metra pod uhlom 45°. Umiestnenie kamery musí spĺňať požiadavky pre správne zachytený obraz. Izba sa nachádza na druhom poschodí, čo páchatelovi sťažuje vniknúť do objektu oknom. Preto kamera smeruje svoj väčší záber na vstupné dvere. Vlastnosťou kamery je rozsah 120° horizontálne, čo umožní zachytiť aj možný vstup cez okno.

Umiestnenie PIR senzora je podobné ako u kamery, avšak na opačnej strane. Musí sa dbať na to, aby senzor nečelil slnečnému svetlu, čiže nemôže byť oproti oknu. Práve preto najvhodnejšia možnosť, kde umiestniť PIR je do ľavého rohu vedľa okna.

Samostatné Raspberry Pi sa nachádza na počítačovom stole, spolu s notebookom, ktorý je napojený na mikropočítač vzdialeným prístupom. Užívateľ si tak môže sledovať zachytenú snímku páchatela.

6 Zapojenie kamerového systému

Táto kapitola zahŕňa postup pri inštalovaní Raspbianu do Raspberry Pi, zároveň pripojenie kamery a PIR detektoru. Neskôr sa zameriava na vytvorenie jednoduchého skripta, v jazyku Python, ktorý obnáša odfotenie objektu na základe nežiaduceho pohybu v objekte. Taktiež sa vytvorí obojsmerná komunikácia medzi Raspberry Pi a smartphonom, pomocou MQTT protokolu. Na prípravu do diplomovej práce sa rozoberú možnosti mobilnej aplikácie, a zároveň sa odhadne potrebný záložný zdroj.

6.1 Pracovný postup zapojenia Raspberry Pi a jeho komponentov

V internetovom e-shope boli zakúpené potrebné komponenty na zostavenie kamerového systému. Zakúpili sa Raspberry Pi 4B, PIR detektor, kamera. Bolo potrebné premyslieť želanú vzdialenosť jednotlivých komponentov, vzhľadom na možnosť použitia tohto systému aj v exteriéry. Práve preto CSI kábel kamery, ktorý je na trhu dostupný iba vo veľkosti maximálne 200cm bolo nutné prepojiť s CSI konektorom s ďalším CSI káblom o dĺžke 200cm. Takto sa dosiahla 4 metrová vzdialenosť kamery od Raspberry Pi.

Na pripojenie PIR detektora sú potrebné prepojovacie káble s koncovými pinmi samička - samička. Obdobne ako aj u kamery, na trhu je maximálna vzdialenosť 30cm, čo nespĺňalo očakávania. Práve preto bolo potrebné si dané tri prepojovacie káble vytvoriť. Zakúpili sa šesťkrát piny samičky, šesť jedнопinových konektorov a trikrát 4 metrové jednofázové káble rôznej farby. Postup je jednoduchý. Pomocou klieštikov sa odobrala ochrana na kábloch, spojilo sa s pinmi a zasklieštil sa spoj. Vytvorený prepoj sa ešte vložil do jedнопinového konektora, pre jeho izoláčnú vlastnosť. Tento postup sa zopakoval ešte päťkrát. Červený kábel sa použil pre *VCC* pin, zelený pre *GND* pin a biely pre *OUT* pin 4.3.

Po úspešnom pripojení komponentov bolo potrebné ich otestovať. Prvým krokom obsahoval inštaláciu Raspbian na Raspberry Pi podľa manuálu 4.1, a následné nastavenie prihlasovacích údajov, ako sú meno a heslo.

Druhým krokom bolo testovanie kamery. Pomocou príkazových riadkov viz. 4.2 sa potvrdila funkčnosť kamery či už pri snímaní videa alebo odfotení objektu.

Pomocou programovacieho jazyku Python a knižnice *PIcamera* viz. 3.3 sa napísal skript, ktorý po jeho spustení odfoťí objekt a uloží snímku do vybranej zložky.

Výpis 6.1: Skript pre kameru

```
from picamera import PiCamera
#importovanie knižnice PiCamera

camera= PiCamera()
try :
    camera.capture("test.jpg")
    #odfotenie scény a uloženie fotky na disk s názvom "test"
finally :
    camera.close()
    #ukončenie procesu s kamerou
```

Tretím krokom bolo otestovanie PIR detektora. Do skriptu bol dopísaný kód, ktorý po detekcii pohybu v sledovanom objekte vypíše *Pohyb detekovaný!* a odfoťí objekt, v prípade zastavenia pohybu vypíše *Pohyb zastavený!*

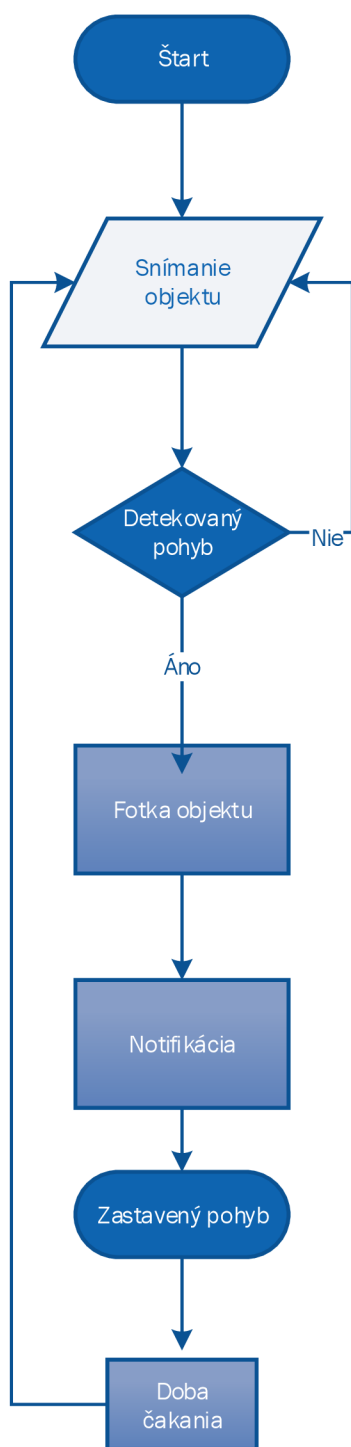
Výpis 6.2: Skript pre PIR detektor

```
from gpiozero import MotionSensor
#importovanie GPIO modulov
import time
#importovanie časového modulu

PIR= MotionSensor(4)
#prepojenie pinu so vstupom PIR detektora

while True:
    PIR.wait_for_motion()
    #čaká na pohyb
    print ("Pohyb□zdetekovaný!")
    camera.capture("test.jpg")
    sleep(10)
    #zastaví proces na 10 sekúnd
    PIR.wait_for_no_motion()
    print ("Pohyb□zastavený!")
    #ohlási zastavený pohyb
```

Pre lepšiu názornosť je zhotovený vývojový diagram, ktorý vysvetľuje prebiehajúci proces skriptu.



Obr. 6.1: Vývojový diagram

6.2 Testy merania

Na otestovanie a správne nastavenie PIR detektora sa uskutočnili dva testy.

Prvým testom bolo postupné otáčanie potenciometra času zopnutia v smere hodinových ručičiek. Týmto posúvaním sa zvyšovala hodnota dĺžky doby zopnutia hodnoty H. Namerané hodnoty boli vynesené do tabuľky.

Druhé meranie testovalo citlivosť detektora. Postupovalo sa podobným postupom ako pri prvom meraní, avšak otáčalo sa potenciometrom pre nastavenie citlivosti senzora v smere hodinových ručičiek. Došlo sa k záveru, že najvhodnejšie nastavenie PIR detektora je nastaviť potenciometer citlivosti na najvyššiu hodnotu, vzhľadom na dĺžku interiéru a dobu zopnutia nastaviť čo na najmenšiu, aby bolo hlásenie správ v čo najpresnejšom čase.

Tab. 6.1: Porovnávanie doby zopnutia pri najmenej citlivosti

Potenciometer	Čas[s]
0	8
0.25	132
0.5	300
0.75	420
1	540

Tab. 6.2: Porovnávanie citlivosti senzora

Potenciometer	Vzdialenosť detekovania[m]
0	2.5
0.25	3
0.5	5
0.75	6.5
1	8

6.3 MQTT Protokol

Na prepojenie smartphonu a Raspberry Pi bol vybraný MQTT protkol. Prvým krokom bolo nutné nainštalovať podporu tohto protokolu príkazovým riadkom do RPi rovnako ako knižnicu do Pythonu.

```
sudo apt-get install -y mosquitto mosquitto-clients
sudo pip install paho-mqtt
```

Následne na smartphone nainštalovať mobilnú aplikáciu *MQTT Dashboard*. Výber tejto aplikácii spočíval na základe kladných recenzií iných užívateľov.

Dôležitým krokom bola voľba severu MQTT Brokera. Z voľne dostupných brokerov je na výber *ThingMQ*, *ThingStudio*, *cloudMQTT*, *IBM Bluemix*....

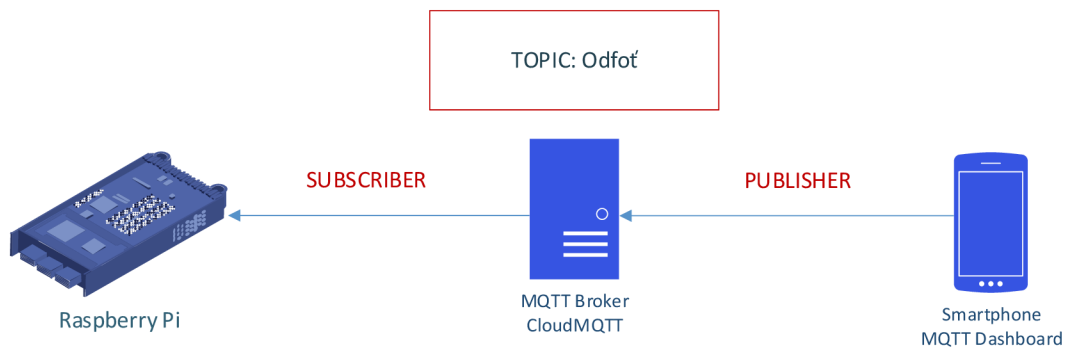
Bol vybraný *cloudMQTT* broker, ktorý ponúka zadarmo desať pripojení (do 10Kbps) a prípadné spoplatnené rozšírenie. Poskytuje zabezpečenie správ, zároveň užívateľské rozhranie je intuitívne. Po registrácii bolo obdržané nasledovné nastavenie brokera:

Server: farmer.cloudmqtt.com

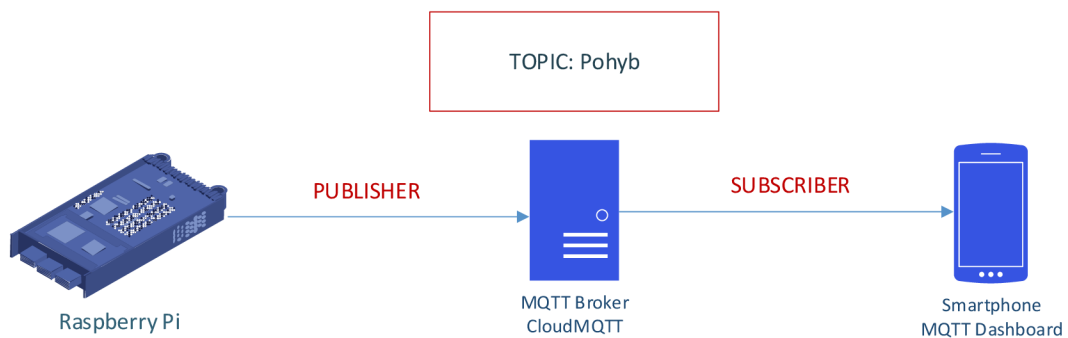
Používateľ: cqeopqjb

Heslo: *****

Port: 13237



Obr. 6.2: Príkaz na odfotenie objektu

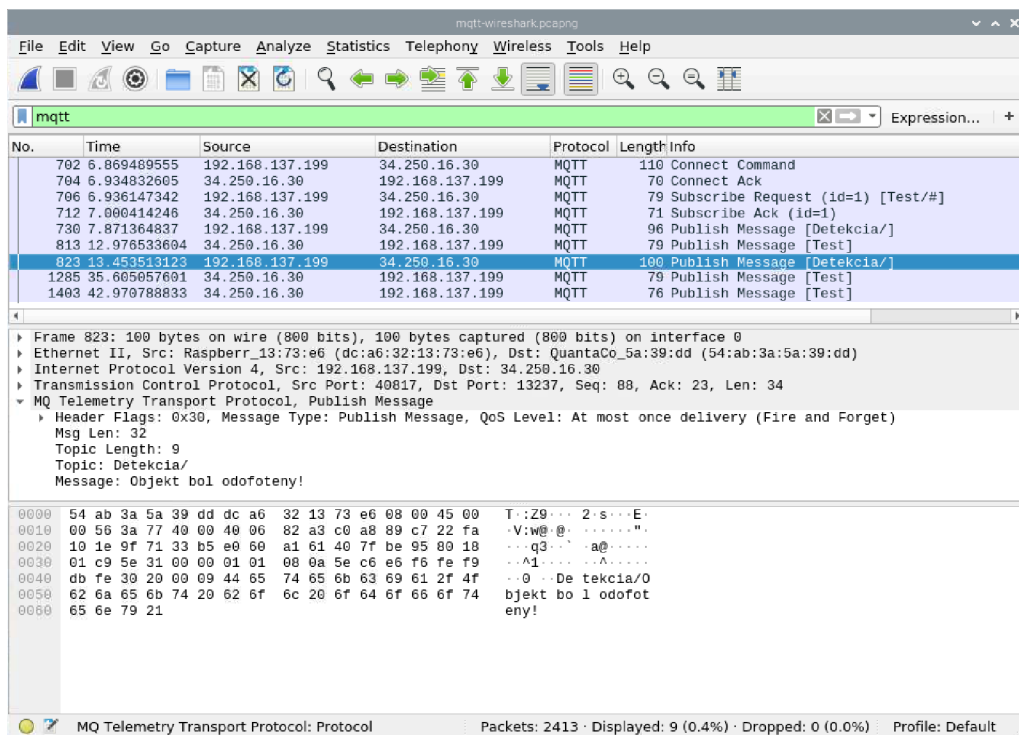


Obr. 6.3: Príkaz na detekovanie pohybu

S myšlienkou znázornených príkazov, bol vytvorený skript, ktorý publikuje a zverejňuje do cloudMQTT príkazy.

6.4 Meranie dátového toku

Pomocou voľne dostupného programu *Wireshark* bola zachytená komunikácia s brokerom CloudMQTT. Hlavným krokom bolo zmeniť vo Wiresharku port MQTT z pôvodného 1883 na pridelený 13237. Týmto bola zabezpečená možnosť filtrovať *mqtt* prevoz. Komunikácia bola testovaná cez WLAN.



Obr. 6.4: MQTT Protokol

Na obrázku 6.4 je možné sledovať naviazané spojenie Raspberry Pi s brokerom, potvrdenie tohto spojenia, publikovanie správy o detekcii pohybu a zachytenie správy s príkazom odfoťiť daný objekt.

Pomocou *Statistics->Endpoints* sa odmeralo, koľko bajtov tento prevoz zaberie.

The screenshot shows the 'Endpoints' window in Wireshark. The window title is 'Wireshark: Endpoints - mqtt-wireshark.pcapng'. The interface includes tabs for 'Ethernet · 2', 'IPv4 · 2', 'IPv6', 'TCP · 2', and 'UDP'. The main table displays the following data:

Address	Port	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
34.250.16.30	13237	9	760	5	375	4	385
192.168.137.199	40817	9	760	4	385	5	375

At the bottom of the window, there are checkboxes for 'Name resolution' (unchecked) and 'Limit to display filter' (checked). On the right side, there is a dropdown menu for 'Endpoint Types' and buttons for 'Copy', 'Zavřít' (Close), and 'Nápověda' (Help).

Obr. 6.5: Dátový prenos

Z tohto merania je možné predpokladať možnosť použitia 3G modemu, vzhľadom na jeho teoretickej maximálnej rýchlosti 7.2Mbps. Pri odfotení objektu a následnom poslaní fotky, počet bajtov narástol na hodnotu 1960B.

7 Mobilná aplikácia

Komunikácia cez vzdialenú správu je každodenným pomocníkom modernej domácnosti. Jedným takýmto inštrumentom sú práve mobilné aplikácie, ktoré je možné ovládať na rôznych zariadeniach. V tejto diplomovej práci sa vybuduje mobilná aplikácia ovládajúca základne vlastnosti kamerového systému.

7.1 Android Studio

Za vývojové prostredie pre budovanie mobilnej aplikácie bolo zvolené Android Studio. Firma *Google* ho priviedla na trh v roku 2013 a od vtedy je k dispozícii zadarmo pre každého užívateľa.

Dôležitou súčasťou pri zhotovení projektu bol výber programovacieho jazyka. Android Studio ponúka jazyk Java alebo Kotlin. Na základe predošlých skúseností bol vybraný jazyk Java. Na internete je množstvo tutoriálov pre začiatočníkov, mnoho z nich je na stránke *Coding in Flow*, odkiaľ boli položené základy tejto mobilnej aplikácie.

7.2 Zhotovenie MQTT spojenia

Existuje mnoho návodov ako zhotoviť MQTT spojenie s vybraným brokerom, ale takmer všetky referujú na webový manuál od firmy *HiveMQ*.

MQTT spojenie je zahrnuté v *Android-Service*, ktorý beží na pozadí samotnej android aplikácie. To znamená, že naviazané spojenie s MQTT brokerom potrvá celú dobu bez prerušenia pri prepínaní rôznych Aktivít.

7.2.1 Inštalácia

Prvým potrebným krokom je pridanie *Paho Android Service* do *Gradle* (buildovací systém a riadiaca správa prostredia). V *Android Studio* sa pridáva tento service do *gradle* súboru.

Výpis 7.1: Gradle konfigurácia

```
repositories {
    maven {
        url "https://repo.eclipse.org/content/repositories/paho-releases/"
    }
}
dependencies {
    compile('org.eclipse.paho:org.eclipse.paho.android.service:1.0.2')
    {exclude module: 'support-v4'}
```

V prvej časti tohto zápisu sa v *Gradle* určuje cesta k balíku *Paho Android Service JAR*. Druhá časť zápisu umožňuje rozšírenie android knižnice o staršie moduly.

7.2.2 Pripojenie

Aby bolo možné deklarovať spojenie s *Paho Android Service* je nutné upraviť *AndroidManifest.xml* súbor, pridaním medzi `<application>` tag. Druhá časť zápisu sa pridáva to `<manifest>` tag.

Výpis 7.2: AndroidManifest.xml

```
<service android:name="org.eclipse.paho.android.service.MqttService" >
</service>
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
```

Na vytvorenie MQTT-spojenia sa použila následná časť kódu.

Výpis 7.3: MQTT spojenie

```
String clientId = MqttClient.generateClientId();
client = new MqttAndroidClient(this.getApplicationContext(),
    "tcp://" + textMQTTHost, clientId);
    // názov servera spolu s používaným portom
options = new MqttConnectOptions();
options.setUsername(textUserName); //užívateľské meno
options.setPassword(textPassword.toCharArray()); //užívateľské heslo

try {IMqttToken token = client.connect(options);
    //príkaz na vytvorenie spojenia so zadanými prihlasovacími údajmi
    token.setActionCallback(new IMqttActionListener() {
        @Override
        public void onSuccess(IMqttToken asyncActionToken) {
            setSubscription();
            //volana funkcia pre subscribtion }

        @Override
        public void onFailure(IMqttToken asyncActionToken, Throwable exception) {
            Toast.makeText(ActivityTwo.this, "Connection was unsuccessful",
                Toast.LENGTH_LONG).show();
            //oznamenie o zlyhani spojenia }));}
    catch (MqttException e) {
        e.printStackTrace();}
```

7.2.3 Publikácia

MqttAndroidClient umožňuje publikovať správy cez *publish(topic, MqttMessage)*. V našom prípade bol *client* vytvorený staticky a publikovanie do brokera je volané cez vytvorenú metódu *sendMessage*. Táto zmena dovoľuje publikovať správy z iných tried, čo by inak nebolo umožnené.

Výpis 7.4: Publikovanie

```
public static boolean sendMessage(String topic, String message){
    try {
        client.publish(topic, message.getBytes(), 0, false);
        return true;
    } catch (MqttException e) {
        e.printStackTrace();
    } return false;}

```

7.2.4 Subskripcia

Zapísanie k *Topic* sa dá pomocou metódy *MqttAndroidClient.subscribe*. Jednotlivé topiky sa môžu prioritizovať. Pri programovaní nebola táto možnosť využitá, keďže každá topica má rovnakú váhu dôležitosti. V kóde je možné vidieť, že aplikácia načúva piatim topicám. V prípade novej správy z danej *Topic* je volaná metóda *setCallback*, kde v *messageArrived* sa programuje čo sa s jednotlivými správami ma diať.

Výpis 7.5: Subscribe

```
String topicStr1 = "Motion/";
String topicStr2 = "Photo/";

public void setSubscription(){
    try {
        client.subscribe(topicStr1,0);
        client.subscribe(topicStr2,0);}
    catch (MqttException e){
        e.printStackTrace();}}

client.setCallback(new MqttCallback() {
@Override
public void connectionLost(Throwable cause) { }
@Override
public void messageArrived(String topic, MqttMessage message)
throws Exception {
    if (topic.contains("Motion/")) {

```

```

        subText.setText("Motion/" + new String(message.getPayload()
        ));}
    else if (topic.contains("Photo/")) {
        // prijaté dáta sú vo formáte bitmap, ktorý je nutné
        dekodovať na zobrazenie fotky
        byte [] src = message.getPayload();
        InputStream input = new ByteArrayInputStream(src);
        ext_pic = BitmapFactory.decodeStream(input);
        image1.setImageBitmap(ext_pic);
    }
}

```

7.2.5 Odpojenie

V prípade želaného odpojenia od brokera a samotného kamerového systému slúži metóda *MqttAndroidClient*, ktorá ukladá hodnotu tokenu. Zároveň oznámi správu o úspešnom odpojení. V mobilnej aplikácii je toto odpojenie použité pri stlačení šipky späť v dolnom navigačnom bare naľavo. Ukáže sa okno, ktoré si overí či užívateľ si je istý o odpojení z brokera.

7.3 Prihlásenie do SurveillanceApp

Po spustení aplikácie je užívateľ vyzvaný na zadanie prihlasovacích údajov, ktoré slúžia pre úspešné spojenie s MQTT brokerom.

Do prvého okna sa zadáva *meno servera* spolu s *portom*, ktorý je používaný. Druhé okno reprezentuje *meno užívateľa* a tretie okno *heslo*. Všetky tieto údaje sú užívateľovi dostupné pri vytvorení konta, v našom prípade *CloudMQTT* brokera.

Užívateľ má možnosť zaškrtnúť ikonu *Remember login*. Táto ikona si zapamätá prihlasovacie údaje pre ďalšie prihlásenie.

Po správnom zadaní údajov, tlačítkom *Login* je užívateľ presunutý do novej triedy. Zároveň toto tlačidlo zdieľa prihlasovacie údaje do novej triedy, pre správne naviazanie spojenia s MQTT brokerom. Tento presun údajov môžeme vidieť v názornom kóde.

Výpis 7.6: Metóda pre prenos údajov a otvorenie novej triedy

```

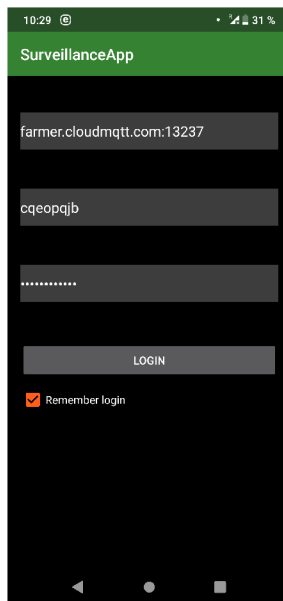
//public static final String:
EXTRA_SERVER = "com.example.surveillanceapp.EXTRA_SERVER";
EXTRA_NAME = "com.example.surveillanceapp.EXTRA_NAME";
EXTRA_PASSWORD = "com.example.surveillanceapp.EXTRA_PASSWORD";

```

```

public void openActivityTwo() {
    Intent intent = new Intent(this, ActivityTwo.class);
    intent.putExtra(EXTRA_SERVER, textServer);
    //do EXTRA_SERVER je vložený text z textServer
    intent.putExtra(EXTRA_NAME, textName);
    intent.putExtra(EXTRA_PASSWORD, textPassword);
    startActivity(intent);
}

```



Obr. 7.1: Prihlásenie do aplikácie

7.3.1 Možnosti aplikácie

Po prihlásení do mobilnej aplikácií má užívateľ možnosť zažiadať o fotografiu objektu, stlačením ikony fotoaparátu. Táto fotka sa zobrazí v strede aktivity. Zároveň je možné si túto fotku poslať na vlastný email, ktorý je si potrebné nastaviť v Nastaveniach (ikona vpravo) alebo si rovno stiahnuť fotografiu do galérie stlačením ikony na stiahnutie (naľavo od fotoaparátu).

Správy, ktoré kamerový systém publikuje sa zobrazujú dole tesne nad ikonami. Pokiaľ je detekcia zapnutá, príde správa o zapnutí tejto funkcie. Ak je detekovaný pohyb, hneď je informovaný užívateľ patričnou správou o pohybe v objekte. Forma správ je v tvare: *Topic/ text*.

V pravom hornom rohu je ikona (*trojbodka*), ktorá po kliknutí zobrazí menu. V menu sa nachádza možnosť spustenia/vypnutia detekcie, nastavenie harmonogramu detekcie, história správ a PIN (v prípade detekcie pohybu, užívateľ zadaním PIN hesla deaktivuje detekciu).



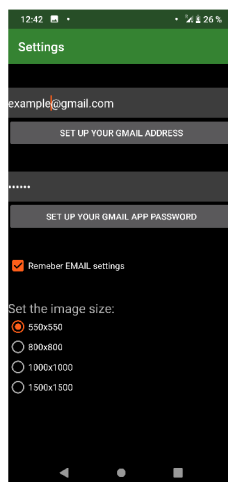
Obr. 7.2: Email/Stiahnutie/Odfotografovať/Nastavenia

7.3.2 Nastavenia

Funkcia zasielania emailu je závislá od zadania užívateľského emailu a hesla pre aplikáciu tretích strán. To je možné vygenerovať v nastaveniach Gmailu. Je to takto bezpečná forma používania údajov, ktoré užívateľ zadáva.

Ďalšie nastavenie je rozlíšenie obrázka. Je na výber zo štyroch veľkostí. Prednastavené rozlíšenie je 300×300 . Ak si užívateľ zadá lepšie rozlíšenie, musí rátať s dlhšou odozvou prichádzajúcej snímky. Čím je zvolené vyššie rozlíšenie, tým je ľahšia identifikácia útočníka.

Tak ako aj v úvode aplikácie, aj tu si môže užívateľ zvoliť zapamätanie emailu a hesla.



Obr. 7.3: Nastavenie

7.4 Detekcia pohybu

Podobne ako v mobilnej aplikácii, aj tu je potreba importovať balík podporujúci MQTT protokol. Dôležitým faktorom je použitie vlákien. Dôvodom je detekcia pohybu. MQTT protokol pracuje s jednotlivými správami sériovo. To znamená, že pokiaľ by bola spustená detekcia pohybu, užívateľovi by nebolo umožnené zároveň poslať snímku objektu na jeho zažiadanie, pretože správa ktorá by bola obdržaná z mobilnej aplikácii, by sa dostala iba do čakajúceho stavu. Tento problém rieši práve využitie vlákien. Konkrétnejšie použitie dvoch vlákien. Prvé vlákno sa venuje subskribovaniu a druhé publikovaniu. Ak je spustená detekcia v objekte a užívateľ si zažiada publikovaním správy o snímku objektu, využije sa vlákno *sub*, ktoré naslúchalo na servery. Takto vlákno *pub* je voľné a prípadný pohyb v objekte môže publikovať do servera. Tým pádom kamerový systém zvláda viacej funkcií zároveň. Vo vlákne *pub* je používaná udalosť *detect*, vďaka ktorej sa ovláda samotné vlákno (spustenie a zastavenie). V znázornenom kóde je možné vidieť nastavenie dvoch vlákien. Jedno pre subskripciu na rôzne témy vďaka metóde *callback*, ktorá umožňuje rozdeliť jednotlivé správy do jednotlivých funkcií. Znak *#* v tomto prípade zahrňuje všetky správy, ktoré patria pod jednotlivú tému.

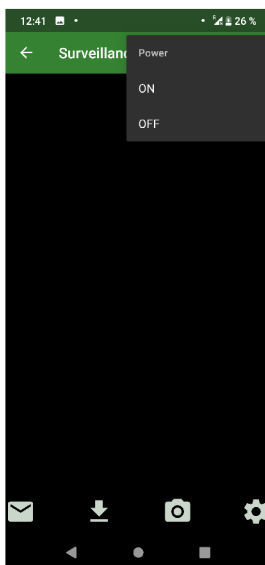
Výpis 7.7: MQTT spojenie v Python jazyku

```
class pubThread(threading.Thread):
    def __init__(self, threadId, name):
        threading.Thread.__init__(self)
        self.threadId = threadId
        self.name = name
        self.detect = threading.Event()
    def run(self):
        while True:
            if self.detect.isSet():
                print(self.detect.isSet())
                detection()
    def pause(self):
        self.detect.clear()
    def resume(self):
        self.detect.set()

class subThread(threading.Thread):
    def __init__(self, threadId, name):
        threading.Thread.__init__(self)
        self.threadId = threadId
        self.name = name
    def run(self):
```

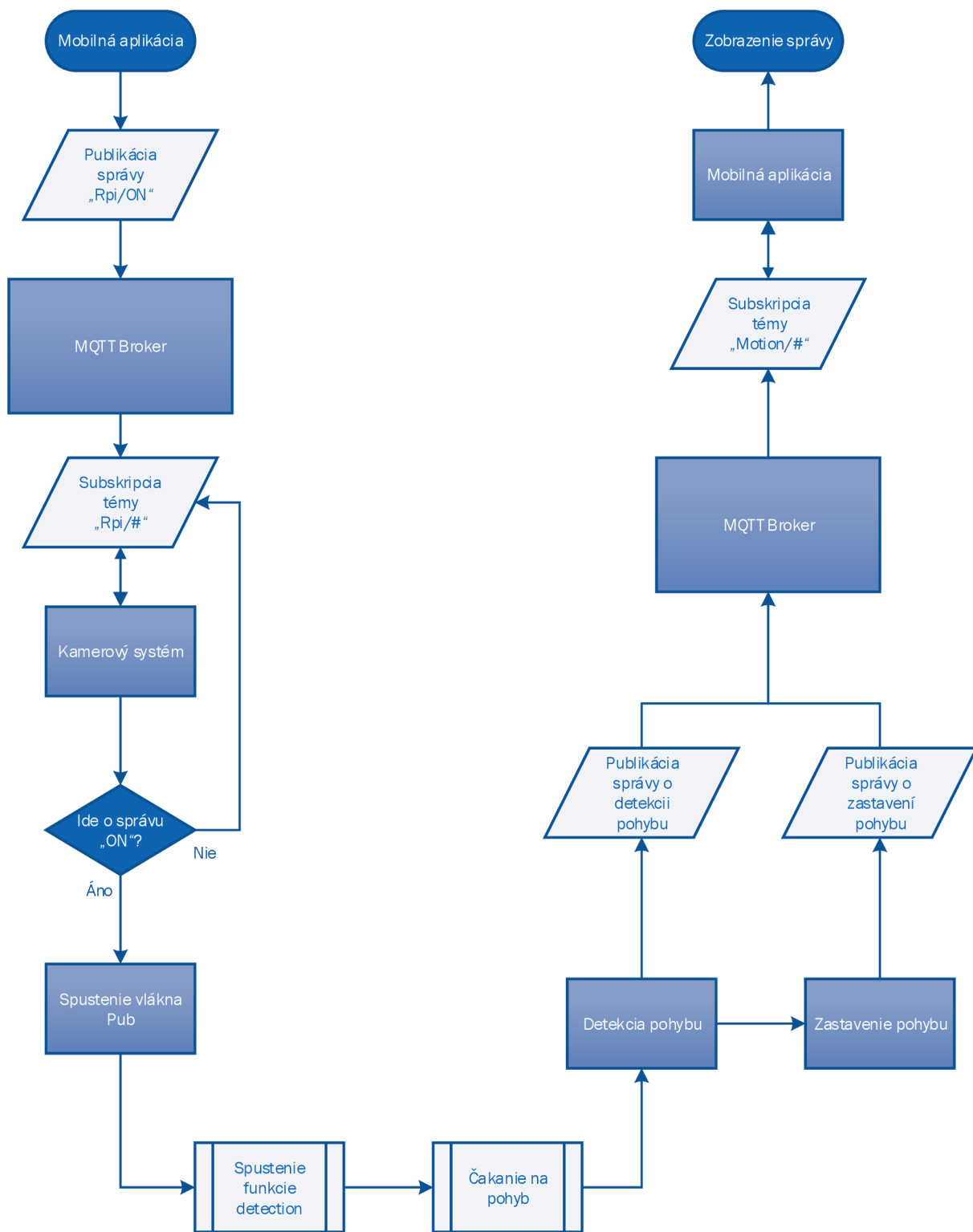


```
client.on_message = on_message
client.message_callback_add('Rpi/EveryDay/#',on_message_everyday)
client.loop_forever()
```



Obr. 7.4: Zapnutie detekcie

Blokové schéma popisuje zapnutie detekcie z mobilnej aplikácie užívateľom, zaslanie správy kamerovému systému, a následné spustenie vlákna *Pub*. Spustením vlákna sa spustí aj udalosť *detect*, ktorá aktivuje funkciu *detection*. Objekt je monitorovaný až kým užívateľ nevypne detekciu v mobilnej aplikácii, stlačením tlačítka *OFF* v menu *Detection*. Medzičasom je užívateľ upozorňovaný o pohyboch v objekte.

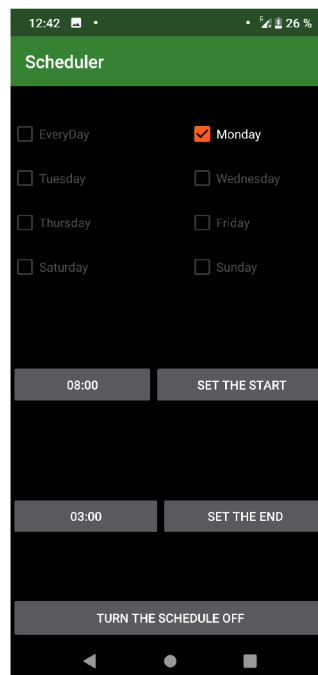


Obr. 7.5: Bloková schéma detekcie

7.4.1 Harmonogram detekcie

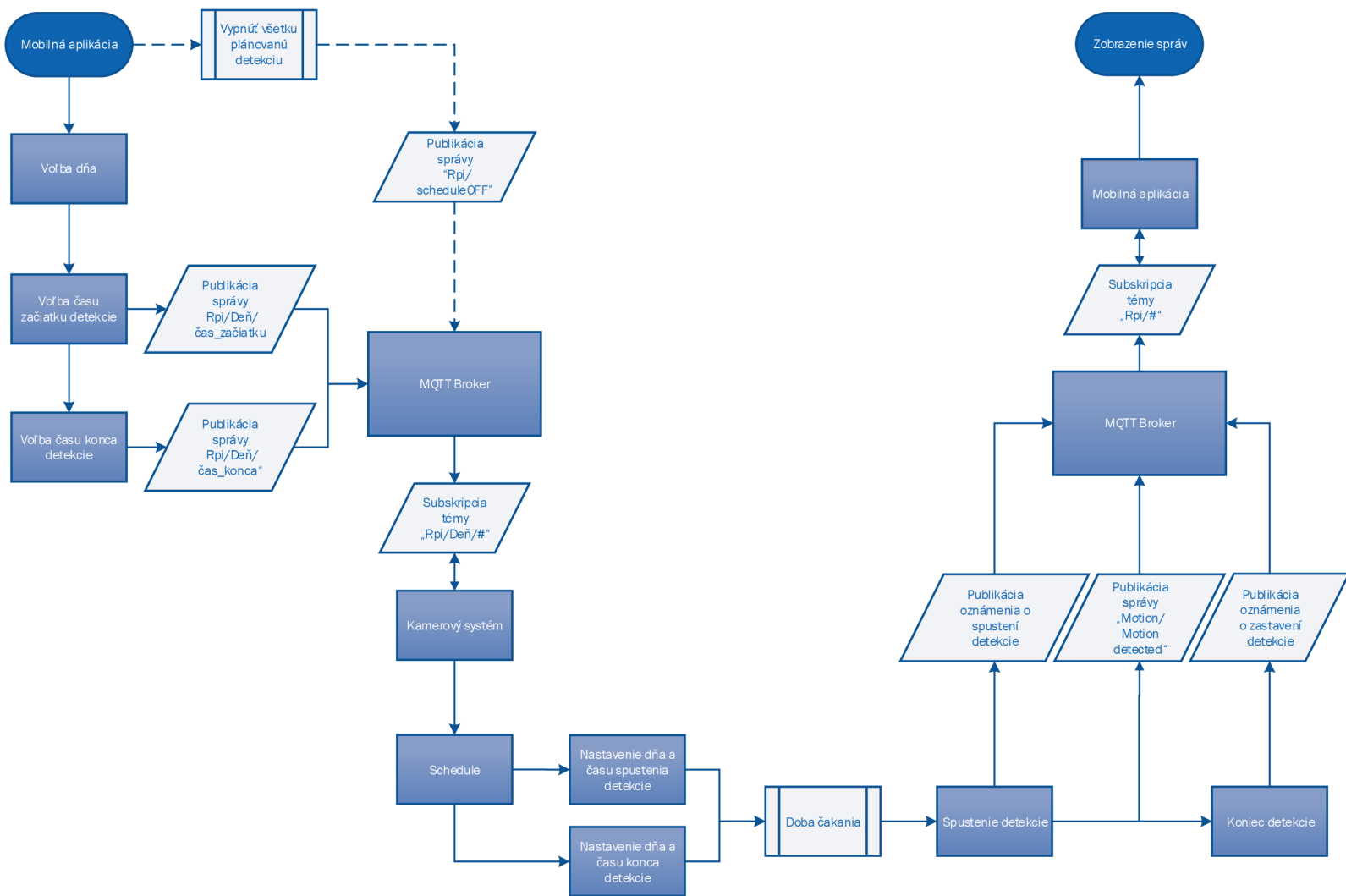
Užívateľ si môže nastaviť pre jednotlivé dni rôzny čas detekcie. Stačí si zaznačiť deň a vybrať začiatok, a koniec detekcie. Platí, jeden deň, jeden čas. Poprípade pokiaľ má pravidelnú časovú dobu kedy je mimo objektu, zvolí si každý deň *Every Day* a nastaví si želaný čas.

V blokovej schéme je znázornený postup, ktorý bol zvolený pri programovaní. Začína sa užívateľskou voľbou dňa, času a končí pri upozornení o detekcii pohybu. Mobilná aplikácia publikuje správy o želanom nastavení harmonogramu detekcie. Broker tieto správy rozpošle zariadeniam, ktoré načúvajú na danú tému. Kamerový systém prijme správy a pomocou funkcie *schedule* nastaví začiatok a koniec detekcie v daný deň. Po spustení detekcie je užívateľ informovaný o jej začatí, rovnako ako o jej zastavení. Tlačidlo *Turn The Detection OFF* umožňuje vypnúť harmonogram detekcie, zaslaním správy do Brokera, ktorý oznámi kamerovému systému o vypnutí funkcie *schedule*.



Obr. 7.6: Harmonogram detekcie

Obr. 7.7: Blokové schéma harmonogramu detekcie



7.5 Upozornenie o výpadku prúdu

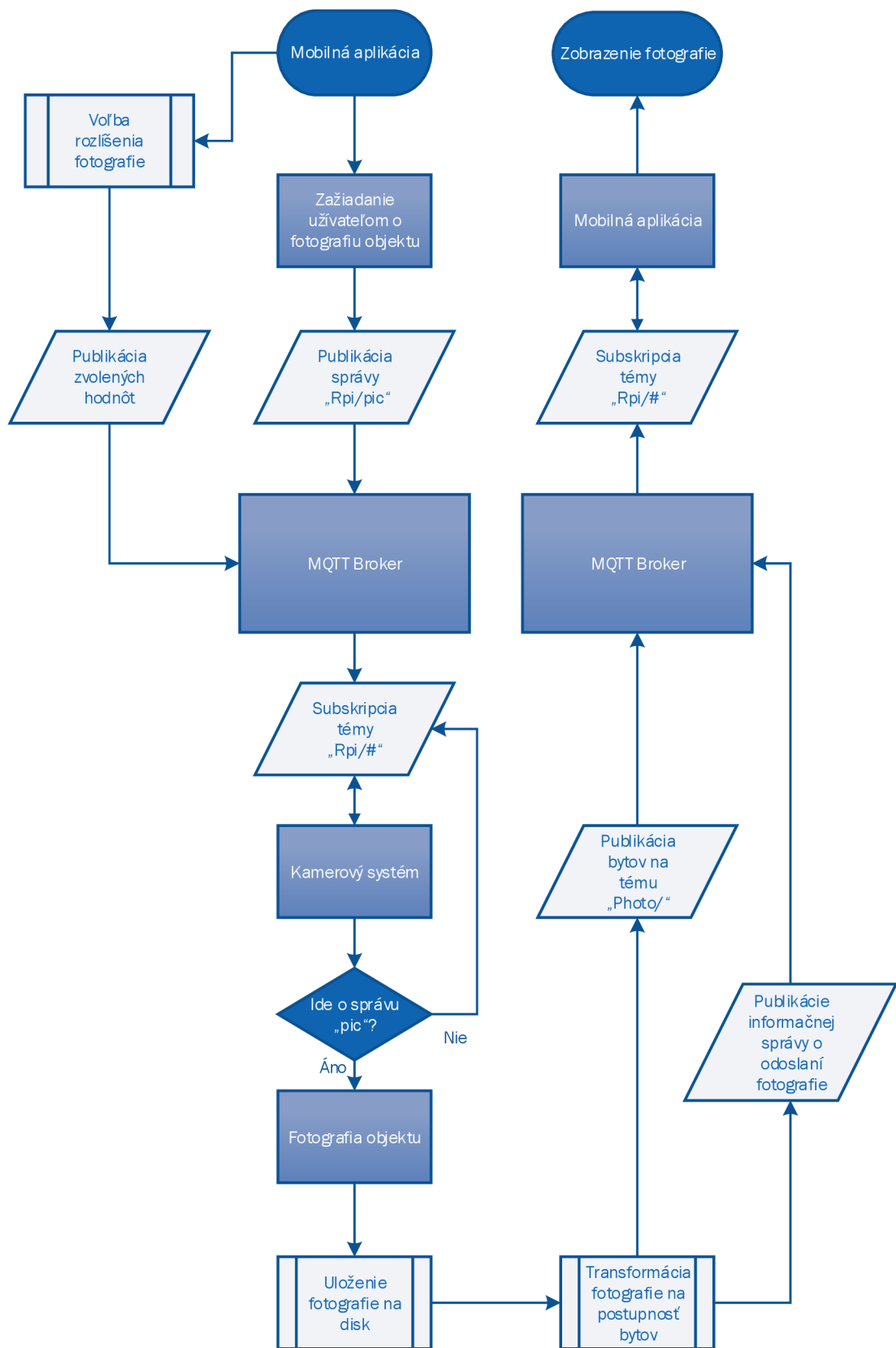
V prípade výpadku elektrickej energie sa zmení zdroj napájania na záložný zdroj. Užívateľ bude informovaný o tomto výpadku informačnou správou, opakujúcou sa každých 30 minút. Takto môže užívateľ odhadnúť ako dlho mu ešte záložný zdroj bude napájať kamerový systém. Kód je riešený veľmi jednoducho. Vyvedení kábel z napájania o hodnote 5V, prevedení pomocou dvoch rezistorov (1K1 a 2K) na 3.3V, je pripojený na GPIO pin. Pokiaľ RPi detekuje hodnotu H na vybranom pine, zdroj napájania je v poriadku. V prípade zmeny stavu pinu z H na L, kamerový systém pošle notifikáciu užívateľovi o výpadku prúdu.

Výpis 7.8: Vlákno detekujúce pripojený zdroj

```
class powerThread(threading.Thread):
    def __init__(self, threadId, name):
        threading.Thread.__init__(self)
        self.threadId = threadId
        self.name = name
    def run(self):
        print("Starting: " + self.name, "\n")
        while True:
            if GPIO.input(power) == 0:
                print("Power state is OFF")
                client.publish("Info/", "Power Outage!")
                sleep(1800)
            elif GPIO.input(power) == 1:
                print("Power state is OK")
                sleep(1800)
        print("Exiting: " + self.name)
```

7.6 Snímka objektu

Ako už bolo spomínané, užívateľ si má možnosť zažiadať o snímku objektu. Tento proces začína publikovaním správy *pic* na tému *Rpi/*. Raspberry Pi naslúcha na túto tému a po jej zachytení začne proces odfotenia objektu. V prvom rade sa kamera aktivuje, neskôr je vytvorený súbor *filename*, ktorý v sebe nesie názov *preview* plus aktuálny deň a hodina zhotovenia obrázku. Následne je zachytený objekt do tohto súboru. Kamera zastaví snímanie a vypne sa. Zachytený obrázok je transformovaný do bytov a publikovaný na tému *Photo/*. Zároveň je publikovaná správa aj pre užívateľa s informáciou, že objekt bol odfotený a má vyčkať na snímku. Bloková schéma zachytáva tento postup.



Obr. 7.8: Bloková schéma žiadosti o fotografiu

7.6.1 Email

Doplňujúcou funkciou je zasielanie emailu na zadanú emailovú adresu v nastaveniach. Raspberry Pi naslúcha na témy *Rpi/Address*, *Rpi/Password* a *Rpi/Mail*. V momente keď si užívateľ nastaví emailovú adresu s heslo, prídu jednotlivé údaje kamerovému systému. Tieto údaje sa uložia do premenných *mailAdd* a *mailPass*. Akonáhle užívateľ stlačí zaslanie fotky na mail zavolá sa importovaný balík Email-Message a zašle sa email na zvolenú adresu. Predmet správy je *ImagePreview*, text správy iba informuje o želanom emaile. Ako príloha je snímka objektu. Zasielanie emailu je umožnené pomocou SMTPS serveru konkrétne *smtp.gmail.com*.

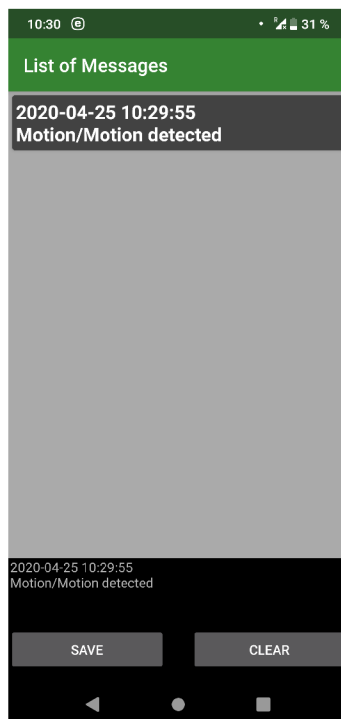
Výpis 7.9: Zaslanie emailu

```
import smtplib
import imghdr
from email.message import EmailMessage

def on_message_mail( client , userdata , msg ):
    message = msg.payload.decode(encoding='UTF-8')
    print("mail: ", message)
    if message=='mail':
        try:
            msg = EmailMessage()
            msg['Subject'] = 'ImagePreview'
            msg['From']= mailAdd
            msg['To'] = mailAdd
            msg.set_content('Here is your requested image preview')
            with open(filename, 'rb') as f:
                file_data = f.read()
                file_type = imghdr.what(f.name)
                file_name = foto
            msg.add_attachment(file_data, maintype='image',
                subtype=file_type, filename=file_name)
            with smtplib.SMTP_SSL('smtp.gmail.com', 465) as smtp:
                smtp.login(mailAdd, mailPass)
                smtp.send_message(msg)
            client.publish("Info/", "Email was sent")
        except:
            print("Something went wrong...")
            client.publish("Info/", "Unable to send email, check settings...")
```

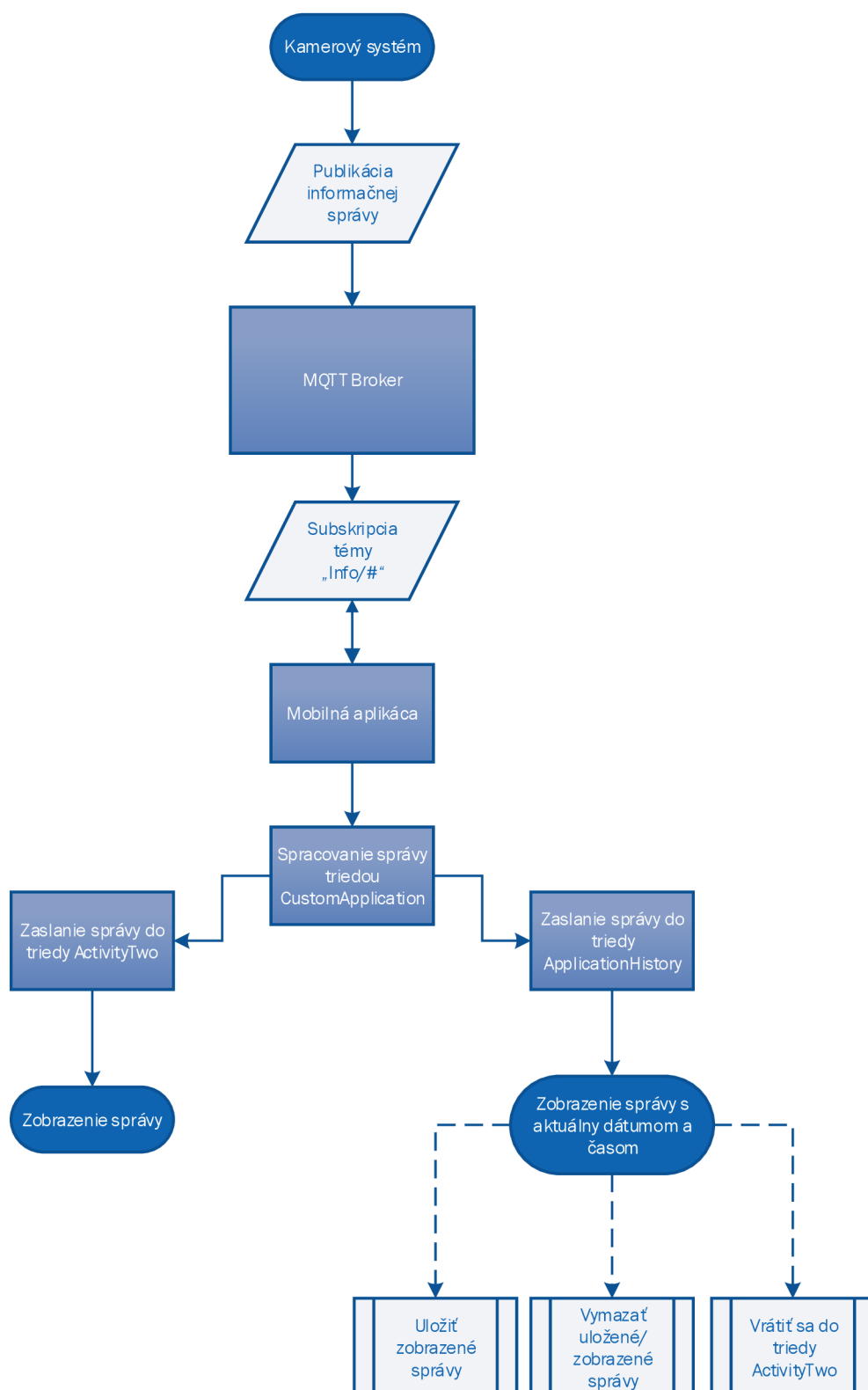
7.7 História správ

Jednou z možností kde sa zobrazujú správy je *List of Messages*. Prakticky sa jedná o históriu zobrazených správ. Užívateľ si môže zvoliť uloženie práve zobrazených správ ako aj ich vymazanie. Pri budovaní tejto funkcie bolo nutné zabezpečiť, aby sa prichádzajúce správy z hlavnej aktivity odoslali aj do aktivity *List Of Messages*. Využila sa nová trieda, rozšírená o *Appliction*, ktorá zachováva globálne spojenie s brokerom.



Obr. 7.9: História správ

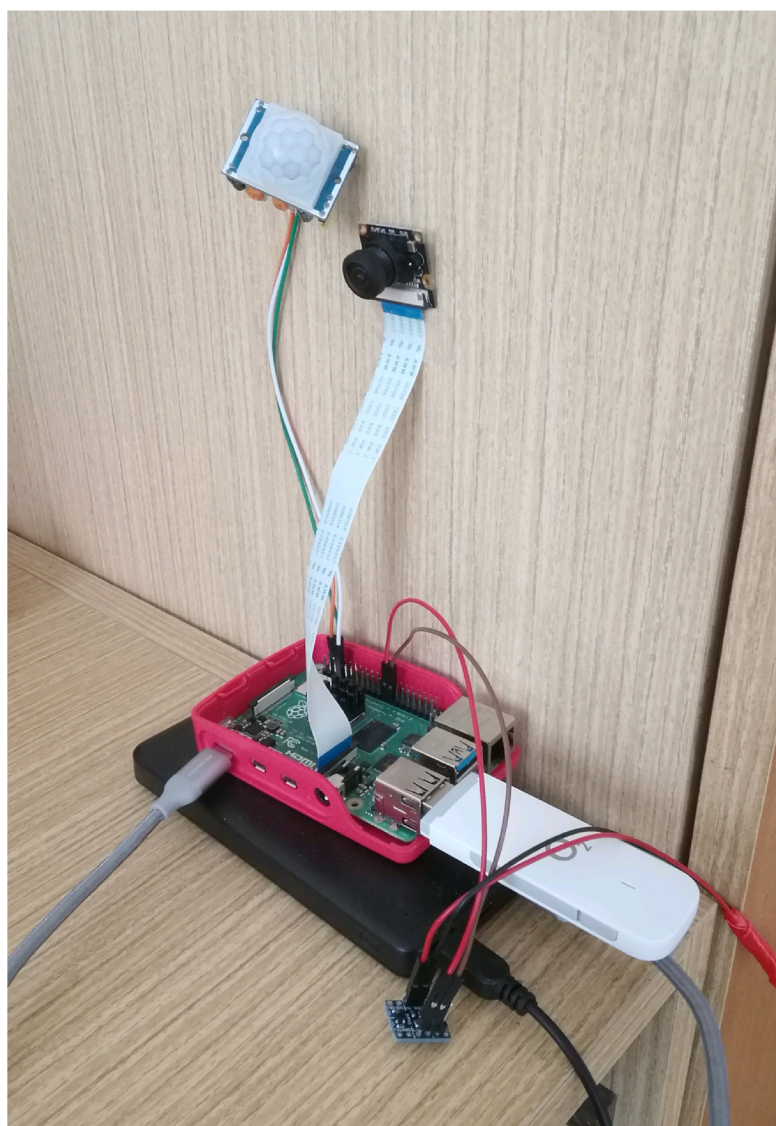
Blokové schéma začína u kamerového systému, odkiaľ sa publikuje informačná správa, napríklad o odfotohrovaní objektu. MQTT Broker posiela túto správu subskriptorom, v tomto prípade aj mobilnej aplikácii. Správa je spracovaná v triede *CustomApplication*, ktorá pošle text do hlavnej aktivity a pomocou *Broadcast Managera* pošle správu do aktivity *List of Messages*



Obr. 7.10: Blokova schéma histórie správ

8 Realizácia kamerového systému

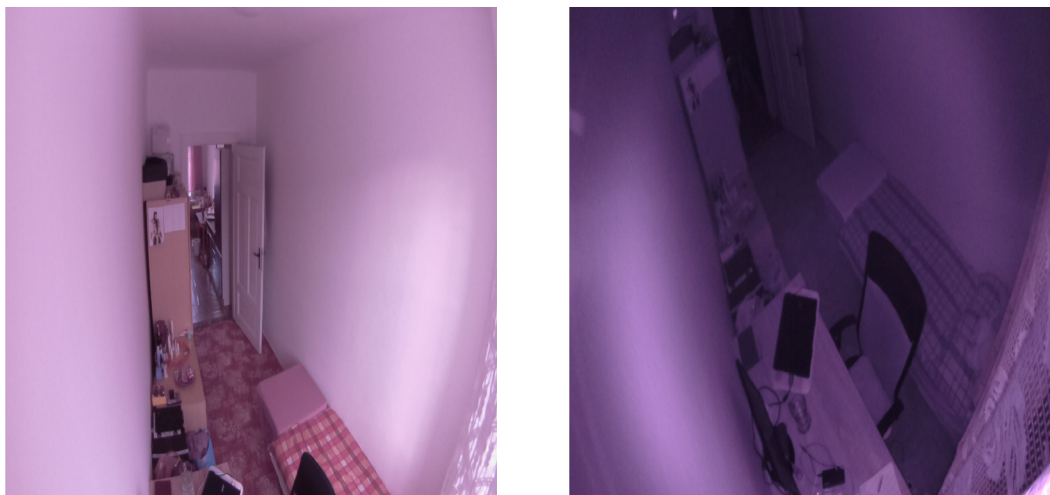
Praktickou časťou bolo zrealizovanie kamerového systému. Na obrázku 8.1 sú odfotené používané komponenty. Na úplnom spodku sa nachádza externá batéria, ktorá je zapojená do Raspberry Pi ako aj do záložného zdroja. Na batérii je položené samotné Raspberry Pi, do ktorého sú cez GPIO piny pripojené komponenty ako PIR detektor - naľavo hore a kamera - napravo hore. Na fotografii je kamera pripojená dvadsať centimetrovým CSI káblom, alternatívou je možnosť využiť dvojmetrový kábel a manuálne priskrutkovať IR osvetlenie. Taktiež to platí aj pre PIR detektor, pre ktorý bol vytvorený štvormetrový prepojovací drátový kábel. V neposlednom rade je cez USB pripojený 4G USB dongle.



Obr. 8.1: Fotografia používaných komponentov

8.1 Testovanie kamerového systému

V nasledujúcich snímkach je možné vidieť zachytený objekt počas dňa a noci.



(a) Deň

(b) Noc

Obr. 8.2: Snímky objektu cez deň noc

8.1.1 Meranie spotreby energie

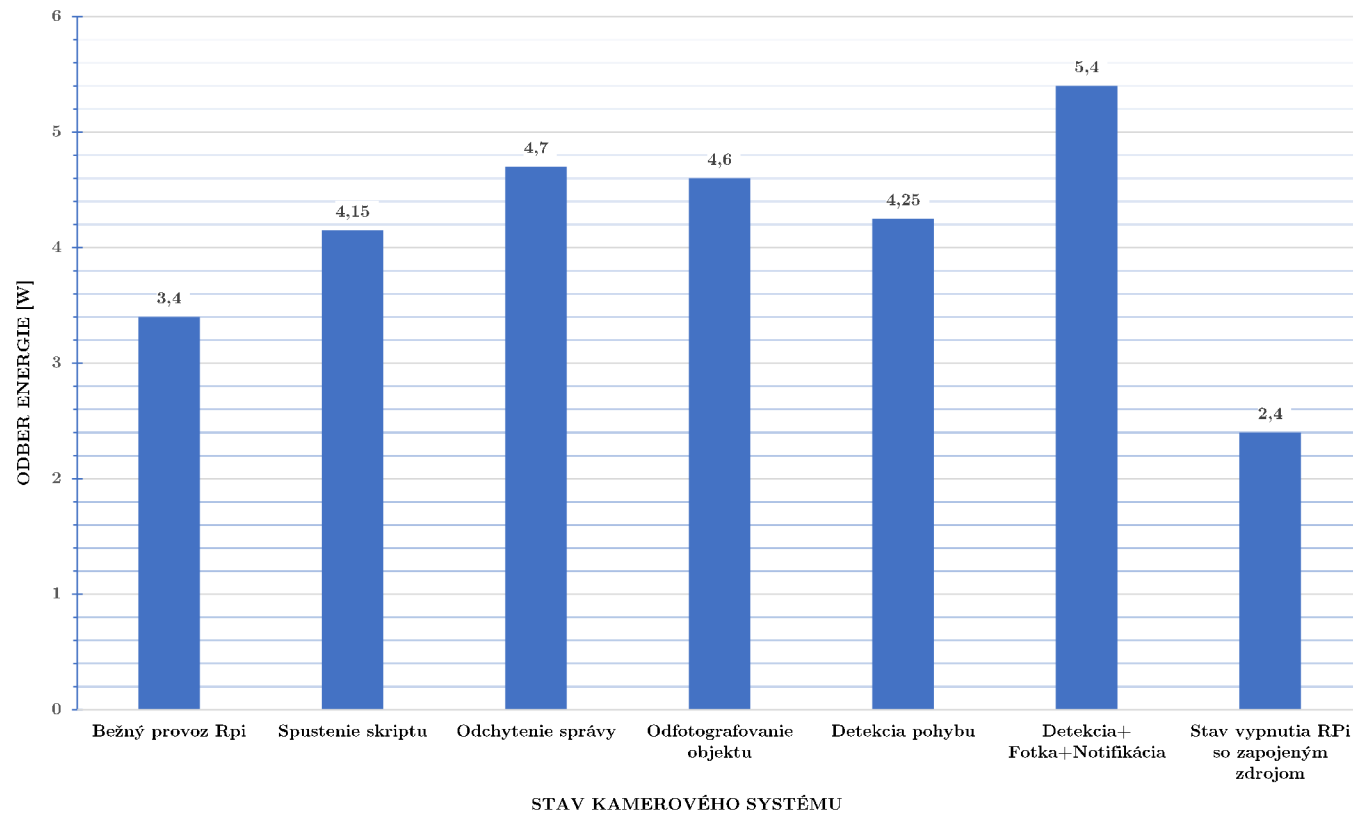
Kamerový systém bol otestovaný aj z pohľadu spotreby elektrickej energie. Toto meranie bolo uskutočnené vďaka meraču spotreby energie *Geti GPM02*. Prakticky sa jedná o zástrčku, ktorú treba zapojiť do elektrickej zásuvky, následne zapojiť zdroj kamerového systému do tohto merača. Na displeji sa zobrazí nameraná hodnota vo Wattoch. Takto boli namerané okamžité hodnoty počas rôznych stavoch kamerového systému. Z týchto nameraných hodnôt je možné porovnať teoreticky odhadnutú spotrebu z kapitoly 4.6.

$$I = 5.4[W]/5[V] \quad (8.1)$$

$$I = 1,08[A] \quad (8.2)$$

Je možné vidieť, že maximálna spotreba sa približne zhoduje s predpokladanou teoretickou spotrebou energie. Najbežnejší očakávaný stav kamerového systému je stav detekcie. Spotreba tým pádom bude menšia než nameraná maximálna hodnota, z čoho sa môže predpokladať vyššia výdrž batérie.

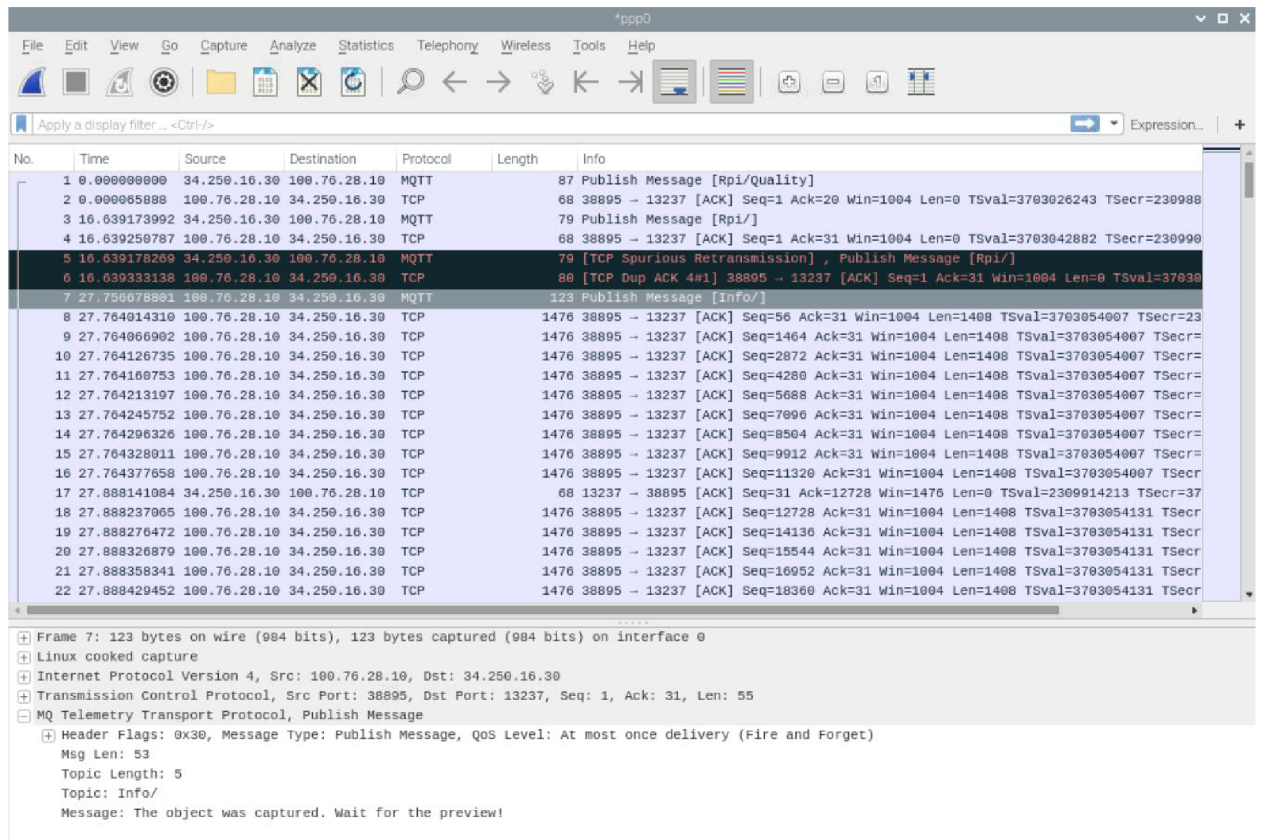
Nameraná spotreba energie



Obr. 8.3: Graf odmeranej spotreby energie

8.2 Meranie dátového toku

Postup merania bol rovnaký ako v predošlej kapitole 6.4, avšak už cez 4G LTE sieť.



The screenshot shows a Wireshark capture of network traffic. The main pane displays a list of packets with columns for No., Time, Source, Destination, Protocol, Length, and Info. The traffic is between source IP 100.76.28.10 and destination IP 34.250.16.30. The packets include MQTT Publish Messages and TCP ACKs. Packet 7 is highlighted, showing a 123-byte MQTT Publish Message with Topic: Info/. The packet details pane below shows the structure of this message: Header Flags: 0x30, Message Type: Publish Message, QoS Level: At most once delivery (Fire and Forget), Msg Len: 53, Topic Length: 5, Topic: Info/, and Message: The object was captured. Wait for the preview!

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	34.250.16.30	100.76.28.10	MQTT	87	Publish Message [Rpi/Quality]
2	0.000065888	100.76.28.10	34.250.16.30	TCP	68	38895 → 13237 [ACK] Seq=1 Ack=20 Win=1004 Len=0 TSval=3703026243 TSecr=230998
3	16.639173992	34.250.16.30	100.76.28.10	MQTT	79	Publish Message [Rpi/]
4	16.639250787	100.76.28.10	34.250.16.30	TCP	68	38895 → 13237 [ACK] Seq=1 Ack=31 Win=1004 Len=0 TSval=3703042882 TSecr=230998
5	16.639178209	34.250.16.30	100.76.28.10	MQTT	79	[TCP Spurious Retransmission], Publish Message [Rpi/]
6	16.639333138	100.76.28.10	34.250.16.30	TCP	88	[TCP Dup ACK 4#1] 38895 → 13237 [ACK] Seq=1 Ack=31 Win=1004 Len=0 TSval=37030
7	27.750078891	100.76.28.10	34.250.16.30	MQTT	123	Publish Message [Info/]
8	27.764014310	100.76.28.10	34.250.16.30	TCP	1476	38895 → 13237 [ACK] Seq=56 Ack=31 Win=1004 Len=1408 TSval=3703054007 TSecr=23
9	27.764066902	100.76.28.10	34.250.16.30	TCP	1476	38895 → 13237 [ACK] Seq=1464 Ack=31 Win=1004 Len=1408 TSval=3703054007 TSecr=
10	27.764126735	100.76.28.10	34.250.16.30	TCP	1476	38895 → 13237 [ACK] Seq=2872 Ack=31 Win=1004 Len=1408 TSval=3703054007 TSecr=
11	27.764160753	100.76.28.10	34.250.16.30	TCP	1476	38895 → 13237 [ACK] Seq=4280 Ack=31 Win=1004 Len=1408 TSval=3703054007 TSecr=
12	27.764213197	100.76.28.10	34.250.16.30	TCP	1476	38895 → 13237 [ACK] Seq=5688 Ack=31 Win=1004 Len=1408 TSval=3703054007 TSecr=
13	27.764245752	100.76.28.10	34.250.16.30	TCP	1476	38895 → 13237 [ACK] Seq=7096 Ack=31 Win=1004 Len=1408 TSval=3703054007 TSecr=
14	27.764296326	100.76.28.10	34.250.16.30	TCP	1476	38895 → 13237 [ACK] Seq=8504 Ack=31 Win=1004 Len=1408 TSval=3703054007 TSecr=
15	27.764328011	100.76.28.10	34.250.16.30	TCP	1476	38895 → 13237 [ACK] Seq=9912 Ack=31 Win=1004 Len=1408 TSval=3703054007 TSecr=
16	27.764377658	100.76.28.10	34.250.16.30	TCP	1476	38895 → 13237 [ACK] Seq=11320 Ack=31 Win=1004 Len=1408 TSval=3703054007 TSecr=
17	27.888141084	34.250.16.30	100.76.28.10	TCP	60	13237 → 38895 [ACK] Seq=31 Ack=12728 Win=1476 Len=0 TSval=2309914213 TSecr=37
18	27.888237065	100.76.28.10	34.250.16.30	TCP	1476	38895 → 13237 [ACK] Seq=12728 Ack=31 Win=1004 Len=1408 TSval=3703054131 TSecr=
19	27.888276472	100.76.28.10	34.250.16.30	TCP	1476	38895 → 13237 [ACK] Seq=14136 Ack=31 Win=1004 Len=1408 TSval=3703054131 TSecr=
20	27.888326879	100.76.28.10	34.250.16.30	TCP	1476	38895 → 13237 [ACK] Seq=15544 Ack=31 Win=1004 Len=1408 TSval=3703054131 TSecr=
21	27.888358341	100.76.28.10	34.250.16.30	TCP	1476	38895 → 13237 [ACK] Seq=16952 Ack=31 Win=1004 Len=1408 TSval=3703054131 TSecr=
22	27.888429452	100.76.28.10	34.250.16.30	TCP	1476	38895 → 13237 [ACK] Seq=18360 Ack=31 Win=1004 Len=1408 TSval=3703054131 TSecr=

Obr. 8.4: Zachytený prenos fotografie

Prvé meranie bolo zamerané na pripojenie do servera, zapnutie detekcie objektu cez mobilnú aplikáciu, a následne oznámenie o detekcii. Dátový provoz si vyžiadala 2073 bajtov, čo značilo komunikáciu o piatich MQTT správach.

Druhé meranie otestovalo spotrebu dát pri odoslaní fotografií o veľkosti 550×550 . Výsledok bolo prenos 162 kilobajtov.

Tretím meraním sa zvýšilo rozlíšenie fotografie na 1500×1500 , čo zvýšilo spotrebu dát na 889 kilobajtov.

Z nameraných hodnôt sa môže odhadnúť maximálna spotreba dát za jeden deň, a to je jeden megabajt. V úvah sa berie rozlíšenie fotografie o 550×550 a celodenné detekovanie oblasti.

Address	Port	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
34.250.16.30	13237	27	2 073	12	865	15	1 208
100.76.28.10	38643	27	2 073	15	1 208	12	865
Fotografia 550x550							
34.250.16.30	13237	151	162 k	39	2 683	112	159 k
100.76.28.10	38895	151	162 k	112	159 k	39	2 683
Fotografia 1500x1500							
34.250.16.30	13237	685	899 k	82	5 617	603	883 k
100.76.28.10	38895	685	899 k	603	883 k	82	5 617

Obr. 8.5: Dátový prenos v troch scenároch

8.3 Cena komponentov

Komponenty boli kupované priebežne preto platnosť cien spadá do obdobia od november 2019 do konca apríla 2020.

Tab. 8.1: Tabuľka cien zakúpených komponentov

Názov komponentu	Cena vrátane DPH	Obchod
Oficiálna Raspberry Pi 4B/2GB sada, biela	2099,71 Kč	Rpishop
SanDisk Ultra 16GB microSDHC+ SD adaptér	167,8 Kč	Rpishop
Waveshare RPi kamera (H)	803,97 Kč	Rpishop
Kábel pre Raspberry Pi kameru (2 m)	232,5 Kč	Rpishop
Prepojka kamerových kabelov pre Rpi	89,91 Kč	Rpishop
Waveshare SIM7020E NB-IoT HAT pre Rpi	549 Kč	Rpishop
PIR pohybový senzor	149,89 Kč	Rpishop
Dratové prípojky (20cm, 40ks)	99 Kč	GM electronic
Dratové prípojky (4m, 3ks)	69 Kč	GM electronic
AVACOM externá batéria PRISMA PQ-10	547 Kč	Xevos
AXAGON USB-C <> USB-A 3.2 Gen 1 kábel 0.5m	165 Kč	T.S.Bohemia
Mobilný internet O2 4G/LTE sada + Huawei E3372H	1105 Kč	Digitor
MicroUSB rozdvojka	89 Kč	Sunnysoft
Adaptér z USB-C na microUSB	99 Kč	Sunnysoft

Záver

Cielom tejto diplomovej práce bolo navrhnuť a realizovať zabezpečenie vybraného objektu autonómnym kamerovým dohľadovým systémom, ktorý je založený na nízkom dátovom toku. Na základe získaných informácií o dostupných možnostiach výberu vhodných komponentov pre zhotovenie tohto systému, bol zhotovený autonómny kamerový dohľadový systém.

V práci sa kladie dôraz na vhodný výber komponentov kamerového systému a následne ich inštalácií. Návrh systému sa realizuje a nainštaluje v interiéri. Jeho funkčnosť je overená jednoduchými testami.

Neskôr sa práca zameriava na budovanie mobilnej aplikácie, ktorá v sebe zahŕňa zapnutie/vypnutie detekcie objektu, fotografovanie objektu, nastavenie harmonogramu detekcie či ukladanie histórie správ.

Kamerový systém je doplnený o nabíjateľný záložný zdroj, ktorý v prípade výpadku elektrickej energie bude slúžiť minimálne šesť hodín, a zároveň užívateľ bude informovaný o tejto zmene zdroja cez mobilnú aplikáciu.

Po úspešnom naprogramovaní mobilnej aplikácie sa otestuje funkčnosť kamerového systému meraním dátového toku využitím LTE 4G modemu, spotreby elektrickej energie a zhotovením finálneho cenníka.

Literatúra

- [1] BURDA, Karel. Základy elektronických zabezpečovacích systémů. Brno: Akademické nakladatelství CERM, 2018. ISBN 978-80-7204-967-7.
- [2] RASPBERRY PI TRADING LTD. Raspberry Pi 4 Computer Model B [online]. UK: Raspberry Pi Trading, 2019 [cit. 2019-11-29]. Dostupné z: <<https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-4-Product-Brief.pdf>>.
- [3] Raspberry Pi 4 Computer Model B. In: Raspberry Pi [online]. UK: RASPBERRY PI FOUNDATION, 2019 [cit. 2019-11-29]. Dostupné z: <<https://www.raspberrypi.org/homepage-9df4b/static/pi4-labelled-99c2e8935bb3cfdb27d23f634ea98a7e.png>>.
- [4] Raspberry Pi GPIO. In: Raspberry Pi Spy [online]. UK: Matt Hawkins, 2012 [cit. 2019-11-29]. Dostupné z: <https://www.raspberrypi-spy.co.uk/wp-content/uploads/2012/06/raspberry_pi_3_model_b_plus_gpio-1024x1024.jpg>.
- [5] Waveshare RPi kamera (H). RPishop [online]. České Budějovice: rpishop.cz, 2019 [cit. 2019-11-29]. Dostupné z: <<https://rpishop.cz/kamery/1045-waveshare-rpi-kamera-h.html>>.
- [6] AL HAFOUDH, Silvia. Ako funguje pohybový senzor PIR a ako ho použiť. In: ROBO KING [online]. Bratislava: Magnepelia, 2019, 06.07 2018 [cit. 2019-11-29]. Dostupné z: <<https://www.roboking.sk/blogs/novinky/ako-funguje-pohybovy-senzor-pir-a-ako-ho-pouzit>>.
- [7] Raspbian. Raspberry Pi [online]. UK: RASPBERRY PI FOUNDATION, 2019 [cit. 2019-11-29]. Dostupné z: <<https://www.raspberrypi.org/downloads/raspbian/>>.
- [8] NOOBS. Raspberry Pi [online]. UK: RASPBERRY PI FOUNDATION, 2019 [cit. 2019-11-29]. Dostupné z: <<https://www.raspberrypi.org/downloads/noobs/>>.
- [9] Python Language Introduction. GeeksforGeeks [online]. Uttar Pradesh: geeksforgeeks, 2019 [cit. 2019-11-29]. Dostupné z: <<https://www.geeksforgeeks.org/python-language-introduction/>>.
- [10] NELLI, Fabio. PiCamera & Python – programming a webcam on Raspberry Pi. In: Meccanismo Complesso [online]. Italy:

- Meccanismo Complesso, 2017, 14 September 2016 [cit. 2019-11-29]. Dostupné z: <<https://www.meccanismocomplesso.org/en/picamera-python-programming-a-webcam-on-raspberry-pi/>>.
- [11] Raspberry Pi Documentation. Raspberry Pi [online]. UK: RASPBERRY PI FOUNDATION, 2019 [cit. 2019-11-29]. Dostupné z: <<https://www.raspberrypi.org/documentation/>>.
- [12] Camera configuration. Raspberry Pi [online]. UK: RASPBERRY PI FOUNDATION, 2019 [cit. 2019-11-29]. Dostupné z: <<https://www.raspberrypi.org/documentation/configuration/camera.md>>.
- [13] Connect the PIR motion sensor. Raspberry Pi [online]. UK: RASPBERRY PI FOUNDATION, 2019 [cit. 2019-11-29]. Dostupné z: <<https://projects.raspberrypi.org/en/projects/parent-detector/3>>.
- [14] Remote Access. Raspberry Pi [online]. UK: RASPBERRY PI FOUNDATION, 2019 [cit. 2019-11-29]. Dostupné z: <<https://www.raspberrypi.org/documentation/remote-access/>>.
- [15] BANKS, Andrew, Ed BRIGGS, Ken BORGENDALE a Rahul GUPTA. MQTT Version 5.0. MQTT [online]. IBM: OASIS Standard, 2019, 07 M8j 2019 [cit. 2019-12-03]. Dostupné z: <<https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html>>.
- [16] Build a remote Surveillance Camera using motionEyeOS & Raspberry Pi. In: DroneBot Workshop [online]. UK: DroneBot Workshop, 2019, 2019 [cit. 2019-12-03]. Dostupné z: <<https://dronebotworkshop.com/motioneyeos-raspberry-pi/>>.
- [17] RPi Camera (H), Fisheye Lens, Supports Night Vision. In: Waveshare [online]. China: Waveshare Electronics, 2019 [cit. 2019-12-03]. Dostupné z: <<https://www.waveshare.com/rpi-camera-h.htm>>.
- [18] PIR Motion Sensor Interfacing with Raspberry Pi using Python. In: Raspberry Pi Robotics [online]. USA: LEGO, 2018 [cit. 2019-12-03]. Dostupné z: <<http://www.raspberrypirobotics.com/pir-motion-sensor-interfacing-with-raspberry-pi-using-python/>>.
- [19] Waveshare SIM7020E NB-IoT HAT pro Raspberry Pi. In: RPishop [online]. České Budějovice: rpishop.cz, 2020 [cit. 2020-05-22]. Dostupné z: <https://rpishop.cz/5437-large_default/waveshare-sim7020e-nb-iot-hat-pro-raspberry-pi.jpg/>

- [20] Huawei E3372h USB modem 4G LTE, bílý. In: Czc.cz [online]. Praha: czc.cz, 2020 [cit. 2020-05-22]. Dostupné z: <https://iczc.cz/41dtv11a26j5mb1rcbjpeqfb9f_1/obrazek/>.
- [21] TOKAR, Jacek. Installing 3G modem. In: Raspberry @ home [online]. USA: Raspberry @ home, 2013 [cit. 2020-05-22]. Dostupné z:<<http://raspberry-at-home.com/installing-3g-modem/>>.

Zoznam symbolov, veličín a skratiek

Skratka	Celý názov	Význam
CCD	Charge-Coupled Device	súčiastka/obvod s nábojovou väzbou
AT	-	sekvencie príkazov ovládajúce modem
NB-IoT	Narrowband Internet of Things	úzkopásmová sieť internetu vecí
HAT	Hardware Attached on Top	pridaný modul pre Raspberry Pi
SIM	Subscriber Identification Module	účastnícka identifikačná karta
Wifi	-	pripojenie na bezdrôtovú lokálnu sieť
IR	Infrared-cut filter	filter blokujúci dlhšie vlnové dĺžky
LED	Light-Emitting Diode	svetlo emitujúca dióda
RPi	Raspberry Pi	jednodoskový počítač
CSI	Camera Serial Interface	sériové rozhranie pre kameru
PAL	Phase Alternating Line	štandard kódovania farebného signálu
HD	High Definiton	vysoké rozlíšenie
HDMI	HD Multimedia Interface	digitálne rozhranie vo vysokom rozlíšení
PoE	Power over Ethernet	nabíjanie cez drátový sieťový kábel
RAM	Rrandom-Access Memory	druh počítačové pamete
H.265	High Efficiency Video Coding	štandard formátu kódovania videa
API	Application Programming Interface	rozhranie pre programovanie aplikácií
IP	Internet Protocol	základný protokol v počítačových sieťach
(S)FTP	(Secure) File Transfer Protocol	zabezpečený protokol prenosu súborov
OS	Operating Systemy	operačný systém
SSH	Secure Shell	protokol pre vzdialenú správu
VNC	Virtual Network Computing	vzdialená správa s grafickým prostredím
DNS	Domain Name System	systém doménových mien
SMTP	Simple Mail Transfer Protocol	protokol umožňujúci prenos emailov

Zoznam príloh

A	Manuál k mobilnej aplikácii	68
B	Inštalovanie mobilnej aplikácie a zoznam použitých balíkov v RPi	69
C	Obsah priloženej elektronickej prílohy	70

A Manuál k mobilnej aplikácii

1. Prihlásenie užívateľa
 - zadať mqtt server:port
 - prihlasovacie meno vygenerované serverom
 - prihlasovacie heslo vygenerované serverom
 - nutnosť zaškrtnúť „Remember Me“ pre zapamätanie si údajov
2. Odfotografovanie objektu a zaslanie fotografie na email, prípadné stiahnutie fotografie
 - kliknúť na ikonu nastavenia
 - zadať gmail e-mailovú adresu
 - zadať e-mailové heslo (heslo pre aplikáciu tretej strany)
 - zaškrtnúť „Remember“
 - zvoliť želané rozlíšenie snímky (pôvodná hodnota je 300x300)
 - kliknúť na ikonu fotoaparátu
 - pri úspešnom odfotografovaní kamerový systém pošle správu
 - kliknúť na ikonu e-mailu
 - pri úspešnom odoslaní e-mailu kamerový systém pošle správu
 - pri neúspešnom odoslaní e-mailu kamerový systém pošle správu
 - kliknúť na ikonu stiahnutia
 - povoliť práva k prístupu k médiám
 - fotografia je uložená v Galérii mobilného zariadenia
3. Detekcia a nastavenie harmonogramu detekcie - menu - ikona trojbodky
 - Detekcia „Power“
 - „ON“ spustí detekciu objektu
 - „OFF“ vypne detekciu objektu
 - Harmonogram „Scheduler“
 - výber dňa v týždni
 - nastaviť začiatok detekcie kliknutím na tlačidlo „START TIME“
 - poslať čas začatia detekcie kliknutím na tlačidlo „SET THE START“
 - nastaviť koniec detekcie kliknutím na tlačidlo „END TIME“
 - poslať čas začatia detekcie kliknutím na tlačidlo „SET THE END“
 - vypnutie nastavených detekcií „TURN THE SCHEDULE OFF“
4. Zobrazovanie správ - v menu - ikona trojbodky
 - vybrať „List Of Messages“
 - notifikácie sa zobrazujú priebežne v okne s dátumom a časom doručenia
 - možnosť ukladania správ kliknutím na tlačidlo „SAVE“
 - možnosť vymazania správ kliknutím na tlačidlo „CLEAR“

B Inštalovanie mobilnej aplikácie a zoznam použitých balíkov v RPi

Balíčky inštalované v Rasperry Pi potrebné pre funkčnosť kamerového systému:

- python3
- sakis3G / wvdial
- ssh (voliteľné)
- ufw (voliteľné)
- vnc (voliteľné)

Pre budovanie mobilnej aplikácie bolo použité Android Studio verzia 3.6.3. Aplikácia nie je dostupná v Google Play. Pre užívateľa, ktorý chce aplikáciu využívať, je potrebné si Android Studio nainštalovať do svojho počítača. Po nainštalovaní Android Studio si užívateľ musí povoliť vo svojom mobilnom zariadení *Režim ladenia USB*. Po pripojení mobilu cez USB kábel k počítaču je vhodné zmeniť nastavenie z USB napájania na *PTP*. Android Studio automaticky rozpozná mobilné zariadenie. Je potrebné otvoriť priečinok *SurveillanceApp* z elektronickej prílohy v Android Studio následovne: *File -> Open... -> umiestnenie prílohy -> SurveillanceApp*. Týmto spôsobom sa importuje mobilná aplikácia do Android Studio a môže sa už nainštalovať do mobilného zariadenia kliknutím na ikonu *Play* v hornom paneli napravo. Aplikácia sa nainštaluje a spustí sama v mobilnom zariadení. Po úspešnom spustení už užívateľ nemusí pripájať mobilné zariadenie znova k počítaču, pretože mobilná aplikácia už bola nainštalovaná a funguje samostatne.

C Obsah priloženej elektronickej prílohy

Priložená elektronická príloha obsahuje skript kamerového systému písaný v jazyku Python, export mobilnej aplikácie napísanej v jazyku Java a k nej priložený manuál.

```
/.....koreňový adresár prílohy
├── SurveillanceApp ..... export mobilnej aplikácie
├── Manual_mobile_app.txt ..... manuál k mobilnej aplikácií
└── KamSys.py ..... skript kamerového systému
```