

ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA ELEKTROTECHNICKÁ

Katedra teoretické elektrotechniky

BAKALÁŘSKÁ PRÁCE

Bezpečnostní kamerový systém založený na RaspberryPi

Autor práce: Dominik Pauli
Vedoucí práce: Ing. Petr Kropík, Ph.D.

Plzeň 2015

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta elektrotechnická

Akademický rok: 2014/2015

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Dominik PAULI**
Osobní číslo: **E12B0369P**
Studijní program: **B2612 Elektrotechnika a informatika**
Studijní obor: **Elektronika a telekomunikace**
Název tématu: **Bezpečnostní kamerový systém založený na RaspberryPi**
Zadávající katedra: **Katedra aplikované elektroniky a telekomunikací**

Z á s a d y p r o v y p r a c o v á n í :

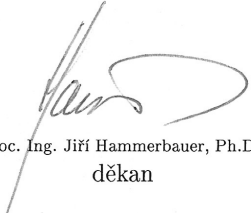
1. Prostudujte problematiku práce s kamerou pro RaspberryPi a detekce pohybu v obraze.
2. Navrhněte montáž, propojení, synchronizaci a software pro kamerový systém tvořený několika Raspberry Pi s kamerou a řídicím Raspberry Pi.
3. Vytvořte systém s využitím dostupných aplikací a vlastního kódu pro vzdálené sledování zabezpečených zón objektů.
4. Zajistěte automatickou detekci pohybu, ukládání snímků (videa) a odeslání informace o narušení bezpečnosti objektu vybraným komunikačním kanálem.
5. Umožněte ukládání nasnímaných dat na vzdálený server a vzdálenou správu vytvořené aplikace.

Rozsah grafických prací: **podle doporučení vedoucího**
Rozsah pracovní zprávy: **20 - 30 stran**
Forma zpracování bakalářské práce: **tištěná/elektronická**
Seznam odborné literatury:


1. **Raspberry Pi as low-cost HD surveillance camera [online]. (c) 2014 [cit. 4.4.2014]. Dostupné z: <http://www.codeproject.com/Articles/665518/Raspberry-Pi-as-low-cost-HD-surveillance-camera>**
2. **Raspberry pi - multiple cameras [online]. (c) 2014 [cit. 4.4.2014]. Dostupné z: http://www.raspberrypi.org/forums/viewtopic.php_t=36280&p=305834**
3. **Raspberry Pi with two webcams [online]. (c) 2014 [cit. 4.4.2014]. Dostupné z: <http://astrobeano.blogspot.cz/2012/08/raspberry-pi-with-two-webcams.html>**
4. **Motion [online]. (c) 2014 [cit. 7.4.2014]. Dostupné z: <http://www.lavrsen.dk/foswiki/bin/view/Motion/WebHome>**
5. **Upton E., Halfacree G. Raspberry Pi Uživatelská příručka, Brno: Computer Press, 2013, ISBN 978-80-2514-116-8.**

Vedoucí bakalářské práce: **Ing. Petr Kropík, Ph.D.**
Katedra teoretické elektrotechniky

Datum zadání bakalářské práce: **15. října 2014**
Termín odevzdání bakalářské práce: **8. června 2015**


Doc. Ing. Jiří Hammerbauer, Ph.D.
děkan




Doc. Dr. Ing. Vjačeslav Georgiev
vedoucí katedry

V Plzni dne 15. října 2014

Abstrakt

Tato bakalářská práce se zabývá návrhem kamerového systému založeného na Raspberry Pi složeného z jednoho řídicího a dvou kamerových Raspberry Pi. Aplikace je naprogramována v jazyce Java s použitím grafického frameworku Java FX a detekce pohybu a záznam obrazu je řešen pomocí software Motion.

Klíčová slova

RaspberryPi, Bezpečnostní kamerový systém, JavaFX, Motion

Abstract

Pauli, Dominik. *The Security Camera System based on RaspberryPi* [*Bezpečnostní kamerový systém založený na RaspberryPi*]. Pilsen, 2015. Bachelor thesis (in Czech). University of West Bohemia. Faculty of Electrical Engineering. Department of Theory of Electrical Engineering. Supervisor: Petr Kropík

This bachelor's thesis deals with application for Raspberry Pi to control camera system based on two Raspberry Pi camera modules and one main Raspberry Pi module for control. This application is written in programming language Java using framework Java FX, motion detection and image saving is solved using software Motion.

Keywords

RaspberryPi, Security camera system, JavaFX, Motion

Prohlášení

Předkládám tímto k posouzení a obhajobě bakalářskou práci, zpracovanou na závěr studia na Fakultě elektrotechnické Západočeské univerzity v Plzni.

Prohlašuji, že jsem svou závěrečnou práci vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 270 trestního zákona č. 40/2009 Sb.

Také prohlašuji, že veškerý software, použitý při řešení této bakalářské práce, je legální.

V Plzni dne 3. června 2015

Dominik Pauli

.....

Podpis

Obsah

Seznam obrázků	vi
Seznam symbolů a zkratek	vii
1 Úvod	1
1.1 Přehled práce	1
1.2 Důvod řešení	1
2 Obecný postup řešení	2
2.1 Obecný způsob řešení	2
3 Rešerše	4
3.1 Existující podobná řešení kamerového systému na RaspberryPi	4
3.2 Existující komerční systémy	4
4 Popis použitého hardwaru	6
4.1 Raspberry Pi	6
4.2 Kamerový modul Raspberry Pi	7
4.3 Řídící modul RaspberryPi	8
5 Popis použitého softwaru	10
5.1 Programovací jazyk Java	10
5.1.1 Důvod použití programovacího jazyku Java	10
5.1.2 Framework JavaFX	11
5.2 Software Motion	11
5.2.1 Motion JPEG MJPEG	12
5.3 Knihovna pro zobrazování MJPEG streamu	12
5.4 Komunikace mezi jednotlivými Raspberry Pi	12
5.4.1 Protokol ICMP a jeho implementace do jazyka JAVA	12
5.4.2 Protokol SSH	13
5.4.3 knihovna Jsch	13

6	Popis aplikace pro řídicí modul kamerového systému	14
6.1	Vnitřní řešení softwaru	14
6.2	Zpracování MJPEG streamu z kamerových modulů v řídicí aplikaci	14
6.3	Nastavování kamerových modulů	15
6.4	Ověření dostupnosti kamerového modulu	15
6.5	Ukládání na vzdálená media	16
6.6	Notifikace uživatele při narušení objektu	16
6.7	Hlavní okno aplikace	17
6.8	Menu pro nastavení aplikace	18
7	Popis instalace kamerového systému a potřebného software	19
7.1	Instalace operačního systému na RaspberryPi	19
7.2	Instalace softwaru na kamerovém modulu	20
7.3	Instalace softwaru na řídicím modulu	20
8	Návrh dalšího rozvoje kamerového systému	22
8.1	Potřebná vylepšení kamerového systému	22
9	Závěr	24
	Reference, použitá literatura	26

Seznam obrázků

2.1	Kamerový systém založený na Raspberry Pi(zde pouze s jedním kamerovým modulem)	3
4.1	Raspberry Pi model B Převzat z [1] 	7
4.2	Raspberry Pi 2 model B Převzat z [2] 	8
4.3	Raspberry Pi model B Převzat z [3] 	9
6.1	Hlavní okno řídicí aplikace	17
6.2	Menu pro nastavení kamer pomocí řídicí aplikace	18

Seznam symbolů a zkratek

A	Amper
V	Volt
GB	Giga byte
MHz	Mega hertz
HDD	Hard disc
RPI	Raspberry Pi
SSH	Secure shell
SFTP	SSH file transfer protocol
ICMP	Internet control message protocol
JNA	Java native access
JVM	Java virtual machine
ftp	File transfer protocol

1

Úvod

1.1 Přehled práce

Tato práce se zabývá návrhem aplikace pro vzdálené ovládání kamerového systému a jeho realizace pomocí řídicího mikropočítače RaspberryPi B2 a a dvou kamerových RaspberryPi B s použitím software Motion pro detekci pohybu a aplikace v jazyce Java pro samotné ovládání.

Teoretická část se zabývá návrhem metody pro řešení kamerového systému a popisem použitých softwarových nástrojů software pro detekci pohybu Motion a frameworku pro výrobu grafického rozhraní aplikace Java FX, popisem již existujících metod a porovnáním s podobnými komerčními kamerovými systémy.

Praktická část se poté zabývá vlastním řešením kamerového systému, realizací aplikace pro vzdálenou správu systému na řídicím modulu Raspberry pi a způsobem komunikace mezi jednotlivými prvky systému a ovládací aplikací. Dále se praktická část zabývá nastavením Raspberry pi potřebným k chodu tohoto systému a samotnou instalací softwarových prvků potřebných k funkčnosti.

1.2 Důvod řešení

V současné době se na trhu nachází velké množství kamerových systémů s rozdílnými vlastnostmi a poměrně vysoké ceny. Mikropočítače RaspberryPi disponují poměrně vysokým výkonem a velmi příznivou cenou společně s hojným rozšířením a vysokou dostupností jak samotného hardware, tak také rozšiřujících doplňků a velkého množství mnohdy oficiálně podporovaného software. Je tedy možné vyrobit víceúčelový kamerový systém za zlomek ceny komerčních produktů.

2

Obecný postup řešení

Tato část se zabývá nastíněním obecného řešení problému.

2.1 Obecný způsob řešení

Práce je řešena pomocí tří Raspberry Pi. První Raspberry Pi funguje jako řídicí modul s řídicí aplikací a další dvě Raspberry Pi fungují jako kamerové moduly.

Kamerová RaspberryPi jsou osazena infračervenou kamerou PI NoIR popřípadě standardní kamerou PI Camera Module a společně s řídicím Raspberry Pi jsou všechny osazeny wi-fi moduly nebo připojeny pomocí ethernetového kabelu a osazeny paměťovými kartami s nainstalovaným operačním systémem Raspbian což je odnož linuxového operačního systému Debian.

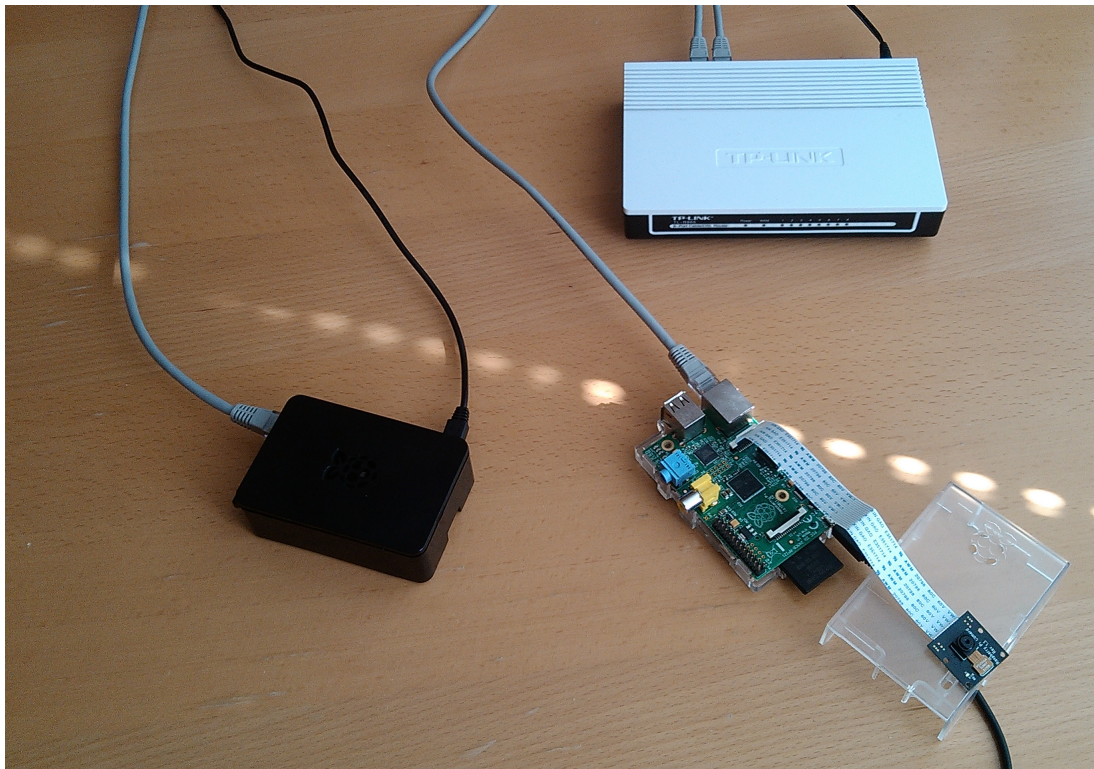
Řídicí jednotka Raspberry Pi je připojena pomocí wi-fi popřípadě ethernetového kabelu k síti a je osazena paměťovou kartou s větší kapacitou s operačním systémem Raspbian a také funguje jako ftp server pro ukládání jednotlivých zachycených snímků a je možností funkce jako web server pro vzdálenou správu řídicí aplikace.

Na kamerové moduly je nainstalován software Motion starající se o řízení kamery a detekci pohybu. Software Motion předává jednotlivé zaznamenané snímky dále ke zpracování podle nastavení uživatele. Na řídicím Raspberry Pi je spuštěna aplikace pro ovládání samotného kamerového systému, která spolupracuje se software Motion. Dále je na řídicím Raspberry Pi nainstalována Java verze 1.8, která je používána pro samotný vývoj aplikace používající Java profil Compact 3, který dovoluje používat Javu i na Raspberry Pi pod profilem Compact 3.

Komunikace mezi řídicím Raspberry Pi a kamerovou Raspberry Pi je řešena pomocí protokolu SSH, přes který je prováděno veškeré nastavování a pro předávání snímků mezi Řídicím a kamerovým modulem běží na řídicím modulu ftp server na který jsou tyto snímky ukládány.

Možností této řídicí aplikace je také automatická notifikace pomocí e-mailu při detekci pohybu.

V současné době je tato řídicí aplikace stále ještě ve vývoji, popisovaná verze je aktuální stabilní verze, která obsahuje pouze základní sadu prvků pro nastavení s ověřenou funkčností a stabilitou a stále je zde velký prostor pro vylepšení a neustále probíhá vývoj této aplikace.



Obr. 2.1: Kamerový systém založený na Raspberry Pi (zde pouze s jedním kamerovým modulem)

3

Rešerše

3.1 Existující podobná řešení kamerového systému na RaspberryPi

Kamerové systémy založené na Raspberry Pi je problematika řešená již množstvím autorů a již existují i některá komerční řešení těchto systémů.

Většina těchto zpracovaných systémů ovšem postrádá některou ze zde podporovaných funkcí. Nejčastěji je chybějící podpora více kamerových modulů. Zaznamenaná data jsou nejčastěji ukládána přímo na kamerovou jednotku která je přímo ovládána popřípadě možnost ovládání úplně chybí, nicméně některá sofistikovanější řešení tyto problémy již řeší.

Mezi nejčastěji zmiňované patří článek Raspberry Pi as low-cost HD surveillance camera [4], který se zabývá podrobnou instalací jednotlivého software a zprovoznění funkčního kamerového systému s jednou Raspberry Pi. Z tohoto článku vychází také tato práce a využívá popisovaný postup pro instalaci kamerových modulů.

Mezi další možnosti vytvoření lze zahrnout článek Turn a Raspberry Pi into a CCTV Security System [18], který se zabývá samotnou instalací hardware a využívá k zobrazování software Motion Pie.

Na internetu je dostupná další celá řada možných řešení popřípadě kombinací řešení předchozích, ať již to je používání software Motion popřípadě používání aplikací v jazyce Python a mnoha dalších.

3.2 Existující komerční systémy

Na trhu existují různá komerční řešení kamerových systémů pohybující se v rozmezí od 5000 Kč až někdy i převyšující 30000 Kč.

V na nižších cenových kategoriích se obvykle systém osazuje dvěma až čtyřmi kamerami a pro ukládání se využívá pevného disku v řídicí jednotce systému.

Ve vyšších cenových kategoriích se systémy osazují osmi popřípadě ještě větším počtem

kamer a ukládání se řeší na pevný disk. Tyto systémy mají často také dálkové ovládání a možnost vzdáleného ovládání stejně jako možnost instalace více pevných disků. Nicméně zde je rozdíl v ceně poměrně značný.

4

Popis použitého hardwaru

Tato kapitola se zabývá popisem specifikací použitého hardware s ohledem na jeho výhody a nevýhody s odůvodněním jeho použití.

4.1 Raspberry Pi

Raspberry Pi je mikropočítač založený na procesoru ARM. V pro tyto účely je použita Raspberry Pi 2 model B pro běh řídicí aplikace a Raspberry Pi model B jako kamerové moduly. Jednotlivé moduly jsou napájeny zdroji ze sítě s výstupním proudem 1A pomocí micro USB rozhraní na napětí 5V.

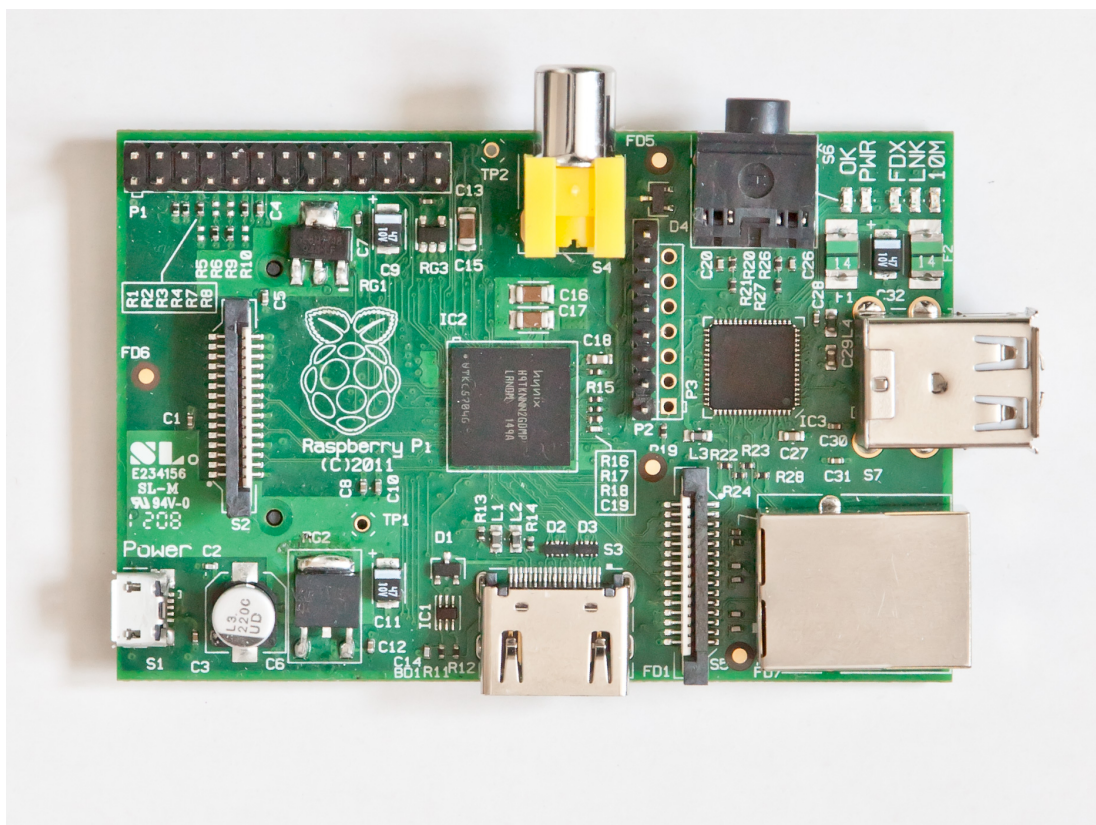
Raspberry Pi model B je postavena na architektuře Broadcom BCM2835 a je osazena procesorem ARM1176JZF-S (ARM11) fungující na frekvenci 700 MHz. Pro tento procesor musí software být kompilován pro ARMv6 [5]. U procesoru je také možné přetaktování pro zvýšení výkonu, ale v tomto případě není využito.

Raspberry Pi 2 model B je postavena na architektuře Broadcom BCM2836, což je architektura totožná s architekturou BCM2835 u předchozích modelů, ale na rozdíl od předchozích modelů je osazen čtyř-jádrovým procesorem Cortex-A7, což přináší poměrně výrazné zvýšení výkonu, stejně jako zvýšení operační paměti na 1GB. Dalším viditelným rozdílem je přidání dalších dvou USB portů což podstatně zlepšuje praktičnost [6] [7].

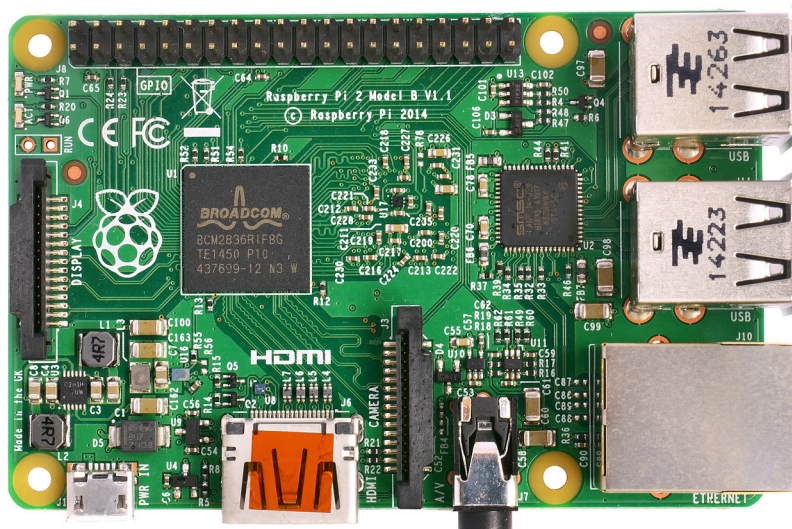
Bezpornou výhodou Raspberry Pi je její cena nepřevyšující 1000kč a hardwarové specifikace společně s tím, že Raspberry Pi je otevřená platforma nám dovolují rozsáhlé rozšíření našeho systému, například je možné pomocí GPIO pinů spouštět vzdálená relé, které by například spustilo alarm popřípadě jiná bezpečnostní opatření, nebo je možné ovládat motory, které by se použily k natáčení vlastní kamery.

Další výhodou Raspberry Pi je, že na rozdíl od většiny mikropočítačů je schopná fungovat s operačním systémem, což v mnoha případech značně zjednodušuje práci. Za zmínku také stojí rozsáhlá komunita kolem Raspberry Pi.

Díky nízké ceně je Raspberry Pi vysoce rozšířený hardware a dohromady s poměrně



Obr. 4.1: Raspberry Pi model B [Převzato z [1]]



Obr. 4.2: Raspberry Pi 2 model B [Převzato z [2]]

kvalitními levnými a dostupnými kamerovými moduly a velkým množstvím dostupných knihoven je Raspberry Pi výhodným hardwarem s velkým množstvím úprav specifických pro uživatele.

Další nespornou výhodou Raspberry Pi je otevřenost hardwarové platformy, kdy jsou dostupné všechny technické údaje a samotný hardware je možné využívat i v komerčních aplikacích bez jakýchkoliv licenčních poplatků.

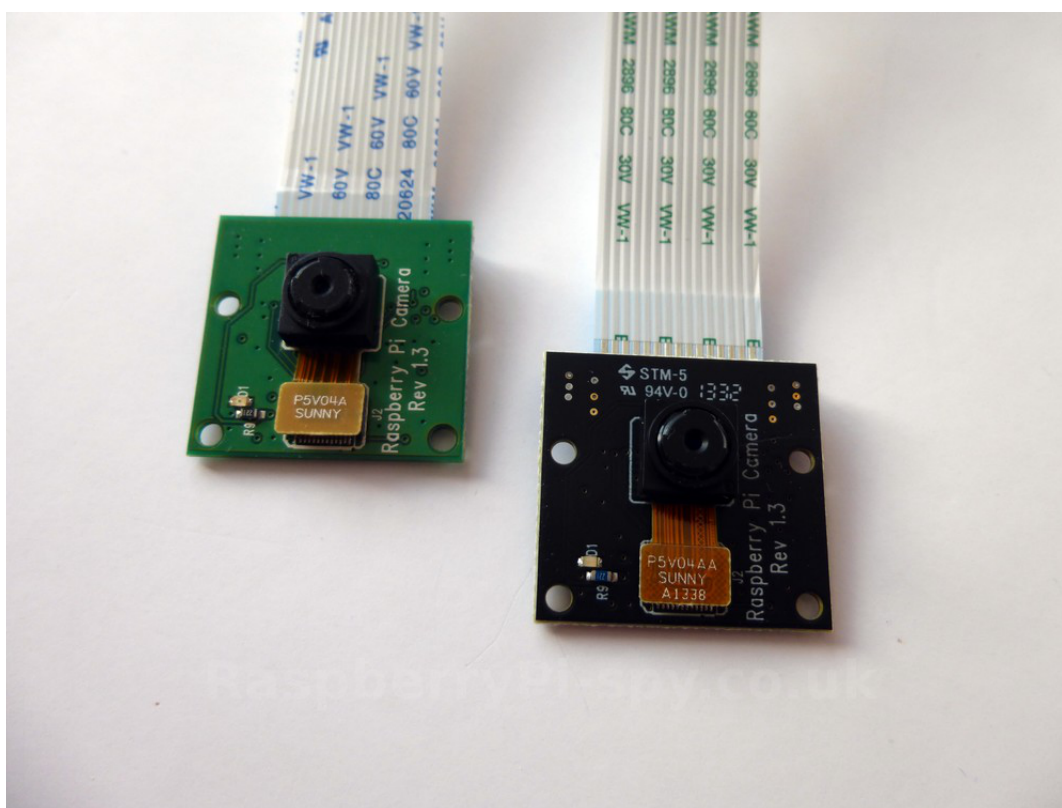
Největší nevýhodou Raspberry Pi je, že v některých případech zdaleka nevyužijeme všechny její funkce a bylo by levnější, popřípadě jednodušší použít levnější více specializovaný hardware.

4.2 Kamerový modul Raspberry Pi

Kamerová jednotka Raspberry Pi je Raspberry Pi verze B opatřená USB wi-fi modulem Wi-Pi pro připojení k síti a kamerou PI NoIR nebo kamerou Raspberry Pi camera module. V tomto případě je použita kamera PI NoIR. Zásadní vlastností této kamery je, že nepoužívá infračervený filtr a proto obrázky zachycené za denního světla mají nerealistickou barevnou paletu.

Výhodou této kamery je, že s přidáním několika infračervených LED je možné zachycovat obrázky i v noci za snížené viditelnosti u míst které osvětlíme neviditelným infračerveným světlem.

Kamerový přídatný modul Raspberry Pi Camera module je schopný zachycovat obraz v rozlišení 5 Mega pixelů popřípadě vide ve formátech 1080p30 nebo 720p60 nicméně pro vysoká rozlišení je tato kamera limitována snímků za sekundu pro více než 47 snímků za sekundu pracuje pouze ve VGA rozlišení [8].



Obr. 4.3: Raspberry Pi model B [Převzato z [3]]

4.3 Řídící modul RaspberryPi

Řídící Raspberry Pi je model 2 což je novější výkonnější verze. Řídící modul je dále připojen pomocí HDMI rozhraní k monitoru pro zobrazení monitorovací aplikace. Další USB porty je možné použít k připojení dalších periférií jako je myš a klávesnice pro ovládání řídicí aplikace. Nebo v případě, že je řídicí modul používám pouze jako server pro vzdálenou správu je pouze připojen wi-fi modul popřípadě ethernetový kabel a napájecí kabel. V případě, že je potřeba ukládat snímky na tento modul, nebo osazená paměťová karta nemá dostatečnou kapacitu je možné připojit flash disk popřípadě externí HDD po provedení příslušných nastavení v aplikaci.

5

Popis použitého softwaru

Tato kapitola se zabývá popisem použitého software a odůvodněním jeho použití s příklady implementace v práci.

5.1 Programovací jazyk Java

Java je velmi rozšířený objektově orientovaný programovací jazyk používající filozofii napiš jednou běž všude, což znamená, že zkompileovaná aplikace je schopná běžet na všech platformách s odpovídajícím runtime prostředím. Java je kompilovaná do byte kódu, který běží na virtuálním stroji JVM. Java je velice rozšířený programovací jazyk, který je v současné době vyvíjen firmou Oracle [9].

Programovací jazyk Java je u Raspberry Pi používán pod profilem Compact 3. Compact profily je koncept představený s verzí Javy 8, kde aplikace je kompilována pouze s částí Platformy Java u nižších profilů Compact se tedy nenachází veškeré knihovny, ale díky tomu omezí velikost a náročnost výsledné aplikace [10].

Pro vývoj tohoto software bylo využito vývojové prostředí Netbeans verze 8.0.2 což je oficiální vývojové prostředí pro jazyk Java od firmy Oracle.

5.1.1 Důvod použití programovacího jazyku Java

Důvod použití programovacího jazyku Java je velká dostupnost mnoha knihoven a samotná filosofie jazyka Java, která je založena na vysoké zpětné kompatibilitě a podpoře co nejvíce platforem.

Nespornou výhodou jazyka Java je, že je možné vyvíjet a kompilovat software na běžném počítači a bez větších úprav ho poté distribuovat na Raspberry Pi. Další výhodou jazyku Java je integrace frameworku Java FX pro vytváření grafických rozhraní a možnosti spouštět aplikace vzdáleně v internetovém prohlížeči.

5.1.2 Framework JavaFX

Java FX je framework programovacího jazyku Java od firmy Oracle, který nahrazuje jejich předchozí zastaralý framework Swing. Tento framework je určený k vývoji grafických rozhraní pro aplikace v jazyce Java.

Výhodou Javy FX je, že využívá takzvaných FXML dokumentů, což je specializovaný druh XML dokumentu upravený pro potřeby Javy FX. Tyto dokumenty jsou určeny pro uchovávání rozložení a vlastností jednotlivých prvků grafického rozhraní v přehledné podobě. Tento způsob značně zkracuje a zpřehledňuje kód a také má zásadní vliv na výkon aplikace a spolu se softwarem pro editaci těchto dokumentů JavaFX Scene Builder 2.0 zásadně urychluje vývoj. Další nespornou výhodou je možnost spouštění Java FX aplikací přímo v prohlížeči.

Nevýhodou JavyFX je, že od verze 8 rev 33 Oracle ukončil podporu tohoto Frameworku na procesorech ARM, který je součástí RaspberryPi. Správa Javy FX na procesorech ARM přešla pod projekt Open JDK pod názvem OpenJFX který má stejnou funkčnost kromě momentálně neimplementovaných knihoven pro media a web pro procesory ARM.

Po ukončení podpory začal vývoj open-source náhrady Javy FX na procesorech ARM, nicméně tato náhrada je ještě stále v rané fázi vývoje a není použitelná jako kompletní náhrada.

5.2 Software Motion

Motion je program, který monitoruje signál jedné nebo více kamer a je schopný detekovat, zda byla významná část obrazu změněna, jinými slovy je schopný detekovat pohyb. Program je napsán v jazyce C a je vytvořen pro operační systém Linux používající video4Linux interface. Motion je nástroj nacházející se v příkazové řádce a jeho výstupy mohou být buď jpeg nebo ppm snímky nebo mpeg video sekvence. Motion používá výhradně příkazovou řádku a je schopný běžet v režimu daemon s relativně malou stopou.

V tomto případě je Motion využíván pro detekci pohybu a následný záznam snímků dvě sekundy před a po době pohybu v podobně snímků v jpeg formátu popřípadě videa ve formátu mpeg podle volby uživatele.

Další vlastností software Motion jsou takzvané trigger eventy neboli události. V podstatě se jedná o reakce na jednotlivé události např.: zachycen snímek, zachycen pohyb. Pomocí této vlastnosti jsme schopni odesílat zachycené snímky na vzdálené úložiště přímo z kamery pomocí skriptu.

Software Motion je provozován pod licencí GNU General Public License version 2.0 (GPLv2) Převzato z [11].

5.2.1 Motion JPEG MJPEG

Mjpeg také nazývaný multipart jpeg je video kompresní formát kde každý jednotlivý snímek popřípadě proložené pole digitální video sekvence je komprimován jako samostatný jpeg obrázek. Původně byl tento způsob komprese určen pro multimediální pc aplikace, ale v současné době se využívá v zařízeních pro záznam obrazu jako jsou digitální kamery, IP kamery, webkamery a softwarem pro nelineární editaci videa (systém, který provádí nedestruktivní úpravu videa). V současné době je podporován všemi nejrozšířenějšími webovými prohlížeči [12].

5.3 Knihovna pro zobrazování MJPEG streamu

Pro zobrazení MJPEG streamu je využita knihovna Webcam Capture distribuovaná pod open-source licencí.

Tato knihovna funguje jako wrapper nad webkamerou a obaluje většinu vlastností webkamer pod jednotlivými ovladači a tím zvyšuje úroveň abstrakce a zjednodušuje práci s webkamerou [19].

V tomto případě je tato knihovna použita v kombinaci s IpCam Driverem tedy ovladačem pro IP kameru který je použit pro zachycení streamu ze software Motion, který je poté potřeba konvertovat pro zobrazení.

Tento proces zachycení a překonvertování způsobuje mírné zpoždění mezi zachycením a zobrazením snímku, nicméně díky způsobu zobrazování a obnovování snímků je toto zpoždění zanedbatelné.

5.4 Komunikace mezi jednotlivými Raspberry Pi

5.4.1 Protokol ICMP a jeho implementace do jazyka JAVA

Protokol ICMP je součástí protokolů internetu používaný pro posílání chybových zpráv a nejčastěji je používán pro nástroj ping pro zjištění dosažitelnosti vzdáleného cíle. V jazyce Java není protokol ICMP implementován a proto metody pro zjištění dostupnosti používají jiné metody, které nejsou velice spolehlivé a jsou velice nestálé.

V této aplikaci je proto použita implementace protokolu ICMP pomocí knihovny ICMP4J což je open source knihovna pod licencí LGPL implementující protokol ICMP pomocí knihovny JNA. JNA je knihovna pro přístup k nativním knihovnám [14].

Knihovna ICMP4J implementuje nativní funkce ping funkcí pro požadavek a odpověď vzdáleného hosta. Tato knihovna je také schopna formátovat odpovědi, což zjednodušuje jejich vyhodnocení v jazyce Java [13].

5.4.2 Protokol SSH

SSH je kryptografický síťový protokol pro spouštění textové shell spojení na vzdálených přístrojích v bezpečné formě. Dovoluje uživateli zařídit zabezpečené spojení i v nezabezpečené síti a vytvořit klient-server architekturu. SSH bylo designováno jako náhrada Telnetu a jiných ne-bezpečných protokolů pro vzdálený přístup.

Existuje více možností jak navázat spojení. Jedním způsobem je prosté přihlášení k síti pomocí dešifrované sítě a hesla. Druhou možností je vytvoření tajného klíčem který nám dovolí přihlášení bez použité hesla, tyto klíče se musí nacházet na obou přístrojích a nikdy se neposílají, pouze se ověřuje zda druhá strana také vlastní tento specifický tajný klíč. Pro kontaktování SSH serverů je standartně vyhrazen port 22.

SSH-2 je novější verze protokolu SSH, která je podstatně více rozšířená a oproti verzi 1 má lepší zabezpečení a od roku 2006 je považována za standart. SSH-2 je nekompatibilní s předchozí verzí [15].

5.4.3 knihovna Jsch

Komunikace mezi jednotlivými Raspberry Pi je řešena pomocí jazyka Java a přidané knihovny jsch-0.1.51 od organizace Jcraft.

Jsch je čistá implementace SSH-2 do javy. Jsch vám dovoluje připojení na SSH server a použití přesměrování portů, X11 přesměrování, přenos souborů a mnoho dalšího a můžete integrovat její funkčnost do vašich vlastních Java programů [16]. Jsch je licencované pod BSD licencí [17].

Knihovna Jsch je v tomto software využívána ke komunikaci mezi jednotlivými Raspberry Pi.

Spojení vytvoříme pomocí vytvoření nového Jsch spojení tedy zavoláním funkce s parametry host, login a heslo.

Pro vzdálené spuštění příkazu na kamerové jednotce je využívána třída ChannelExec jehož metoda dovoluje se připojit a zasílat potřebné příkazy. Zde je použit příkaz pro restartování kamery sudo reboot.

Knihovna Jsch je použita také pro zaslání konfiguračních souborů pro software Motion kdy místo ChannelExec je použita třída ChannelSftp. ChannelSftp dovoluje otevření sftp kanálu, který se využívá pro přenos souborů pomocí protokolu ssh.

Další aplikací knihovny Jsch je stahování snímků z kamer pomocí sftp protokolu. Zde je postup podobný jako při nahrávání konfiguračních souborů, akorát se použijí metody pro stahování akorát namísto vstupních parametrů jsou udány parametry pro výstup souborů.

6

Popis aplikace pro řídicí modul kamerového systému

6.1 Vnitřní řešení softwaru

Aplikace pro kamerový systém běží na řídicím Raspberry Pi a je spouštěna ihned po naboování operačního systému Raspbian. Aplikace funguje na způsobu editace konfiguračních souborů pro software Motion, který se stará o ovládání samotného kamerového hardwaru, o detekci pohybu a záznam videí a snímků a také poskytuje video stream z jednotlivých kamerových jednotek.

6.2 Zpracování MJPEG streamu z kamerových modulů v řídicí aplikaci

Po zachycení snímku kamerou dá software Motion tento snímek k dispozici na adrese tohoto kamerového modulu pod portem 8081 jako MJPEG stream. Tento stream zachycují funkce knihovny Webcam Capture a produkují jednotlivé snímky ve formátu třídy BufferedImage což je třída z awt části jazyka Java. Tento formát není možné přímo zobrazit pomocí komponent Javy FX a proto tento snímek je následně konvertován na objekt třídy WritableImage což je potomek třídy Image, který je podporován Javou FX a je možné jej zobrazit pomocí komponenty ImageView. Výhodou použití WritableImage je možnost změny dat při zobrazení zapisováním jednotlivých hodnot pixelů během používání.

Operace konverze je prováděna každou sekundu pomocí objektu Timeline, který je využíván zejména pro animace V tomto případě je využíván pro periodické obnovování a konvertování získaných snímků. Nevýhodou této metody, je že dosahujeme pouze jednoho snímku za sekundu nicméně při vhodném umístění kamerového modulu je tato doba dostačující pro zachycení rychle se pohybující osoby a jednotlivé snímky zachycené při pohybu pomocí software Motion tímto nejsou ovlivňovány a stále jsou ukládány při každém pohybu.

Třída pro konverzi snímku z BufferedImage na WritableImage

```
1 private void repaintImage(BufferedImage grabbedImage)
2 {
3     if (image1.getWidth() != grabbedImage.getWidth() || image1.getHeight() != grabbedImage.getHeight())
4     {
5         this.image1 = new WritableImage(grabbedImage.getWidth(), grabbedImage.getHeight());
6         this.imageView1.setImage(this.image1);
7     }
8
9     PixelWriter pixelWriter = image1.getPixelWriter();
10
11     for (int x = 0; x < grabbedImage.getWidth(); x++)
12     {
13         for (int y = 0; y < grabbedImage.getHeight(); y++)
14         {
15             pixelWriter.setArgb(x, y, grabbedImage.getRGB(x, y));
16         }
17     }
18 }
```

6.3 Nastavování kamerových modulů

Mezi jednotlivými Raspberry Pi je komunikováno pomocí protokolu SSH přes rozhraní wi-fi, ale je možné bez jakékoliv úpravy propojit jednotlivé moduly také metalicky bez úpravy software.

Samotná konfigurace jednotlivých kamerových modulů probíhá pomocí úprav konfiguračního souboru pro software Motion, kde vzorový konfigurační soubor je načten do programu a převedeny do array listu a naindexovány podle jednotlivých řádků a poté je nalezen řádek s hledaným parametrem a je přepsán požadovaným parametrem, všechny tyto parametry jsou změněny najednou při vyprodukování nového konfiguračního souboru. Aplikace má také možnost ukládání uživatelské konfigurace do souborů ve formátu XML.

Po vytvoření konfiguračního souboru je tento soubor poslán pomocí protokolu SFTP kanálem SSH do kamerového modulu kde je přepsán původní soubor pro nastavení. Po úspěšném přenosu souboru je kamerový modul restartován a software Motion je znovu spuštěn s novou konfigurací.

Ukládání uživatelské konfigurace je řešeno formátem XML do kterého se ukládají postupně nastavení pro obě kamery a nastavení aplikace, které se nahrává při startu aplikace, v případě, že nejsou uloženy konfigurace se načítá základní konfigurace.

6.4 Ověření dostupnosti kamerového modulu

Ověřování dostupnosti kamerových modulů je využíváno jak při zobrazování streamů pro kontrolu před připojením, tak při přenosu konfiguračních souborů kdy je nejdříve kontrolována dostupnost.

Z důvodu chybějící implementace ICMP v jazyku Java je implementace řešena pomocí knihovny ICMP4J [14]. Samotná implementace ověřování je řešena pomocí funkcí ping,

které vytvoří požadavek pro příkaz ping a zpravují jeho odezvu a vrátí aktuální stav dostupnosti kamerového modulu.

Třída pro vykonání ping příkazu

```
1 public boolean isReachable(String hostAddress, int timeout)
2     {
3         try
4         {
5             final IcmpPingRequest request = IcmpPingUtil.createIcmpPingRequest();
6             request.setHost(hostAddress);
7             request.setTimeout(timeout);
8
9             final IcmpPingResponse response = IcmpPingUtil.executePingRequest(request);
10
11             final String formatResponse = IcmpPingUtil.formatResponse(response);
12
13             return !formatResponse.contains("Error");
14         }
15         catch (Exception e)
16         {
17             System.err.println("Host does not exist ...");
18             return false;
19         }
20     }
```

6.5 Ukládání na vzdálená média

Ukládání na vzdálená média je řešeno pomocí software Motion, kdy se po zachycení snímku vyvolá událost on snapshot save, která po nastavení z ovládací aplikace je schopna nahrát snímek na ftp server na řídicím Raspberry Pi pomocí ftp klientu v příkazové řádce wput. Pro budoucí rozšíření řídicí aplikace je v aplikaci také přítomna implementace pro hromadné stažení snímků z kamerového modulu do řídicího pomocí knihovny Jsch. V současné verzi řídicí aplikace tato implementace není použita z důvodu zachování kompaktnosti a stability aplikace.

Kód pro nastavení ftp uploadu v konfiguračním souboru software Motion

```
1 on_picture_save wput ftp://LOGIN:HESLO@FTPADRESA.COM %f
```

6.6 Notifikace uživatele při narušení objektu

Notifikace uživatele při narušení hlídaného prostoru je řešena pomocí e-mailové zprávy obsahující datum a čas narušení. Zásílání e-mailu je v současné verzi ovládacího software řešeno pomocí nastavení chování software Motion při události on snapshot taken. Tato událost se vyvolá automaticky při detekci pohybu a uložení snímku. Součástí software Motion je miniaturní mail server, který dovoluje odesílání e-mailových zpráv bez potřeby jakýchkoliv externích nástrojů. Ovládací aplikace má také zabudovanou funkčnost pro zásílání e-mailových zpráv pomocí java mail api nicméně tyto funkce nejsou v současné době podporovány a implementace je zde z důvodu možnosti budoucího rozšíření ovládacího software.

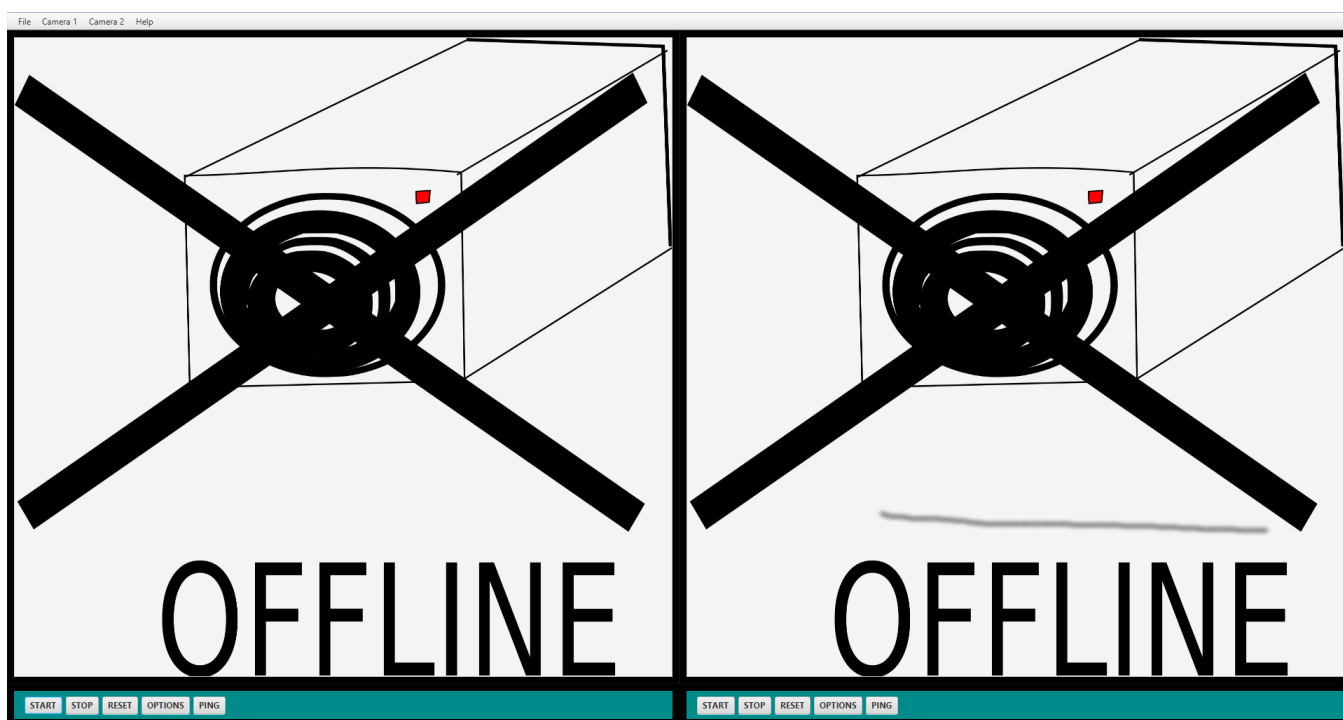
Příkaz pro zasílání e-mailu při detekci pohybu v konfiguračním souboru software Motion |Převzato z [20]|

```
1 on_event_start sendEmail -f GMAILLOGIN@gmail.com -t GMAILLOGIN@gmail.com -u "PREDMET" -m
2 "Byl zjisten pohyb v: \%d \%m \%Y. Cas pohybu: \%H:\%M (Hour:Minute). "
3 -s smtp.gmail.com:25 -xu GMAILLOGIN@gmail.com -xp GMAILHESLO
```

6.7 Hlavní okno aplikace

Na hlavním okně aplikace jsou zobrazeny video streamy u jednotlivých kamer. Tyto streamy jsou získány ze software Motion, který tento stream vysílá na ip adrese kamery a portu specifikovaném uživatelem v nastavení při instalaci software Motion v tomto případě se jedná o port 8081. Tyto streamy jsou zobrazeny pomocí komponenty Image view do které jsou snímky dodávány pomocí knihovny Webcam Capture.

V horní části aplikace se nachází rozbalovací menu ve kterém se nachází ovládací tlačítka pro jednotlivé kamery a také tlačítko pro ukončení aplikace a roztažení aplikace na celou obrazovku. Pod položkou Help se nachází tlačítko pro zobrazení nápovědy pro ovládání programu. Pod video streamy se nachází tlačítka pro ovládání jednotlivých kamer a informační textové pole ve kterém se objevují chybová hlášení a také informují o současné činnosti aplikace.

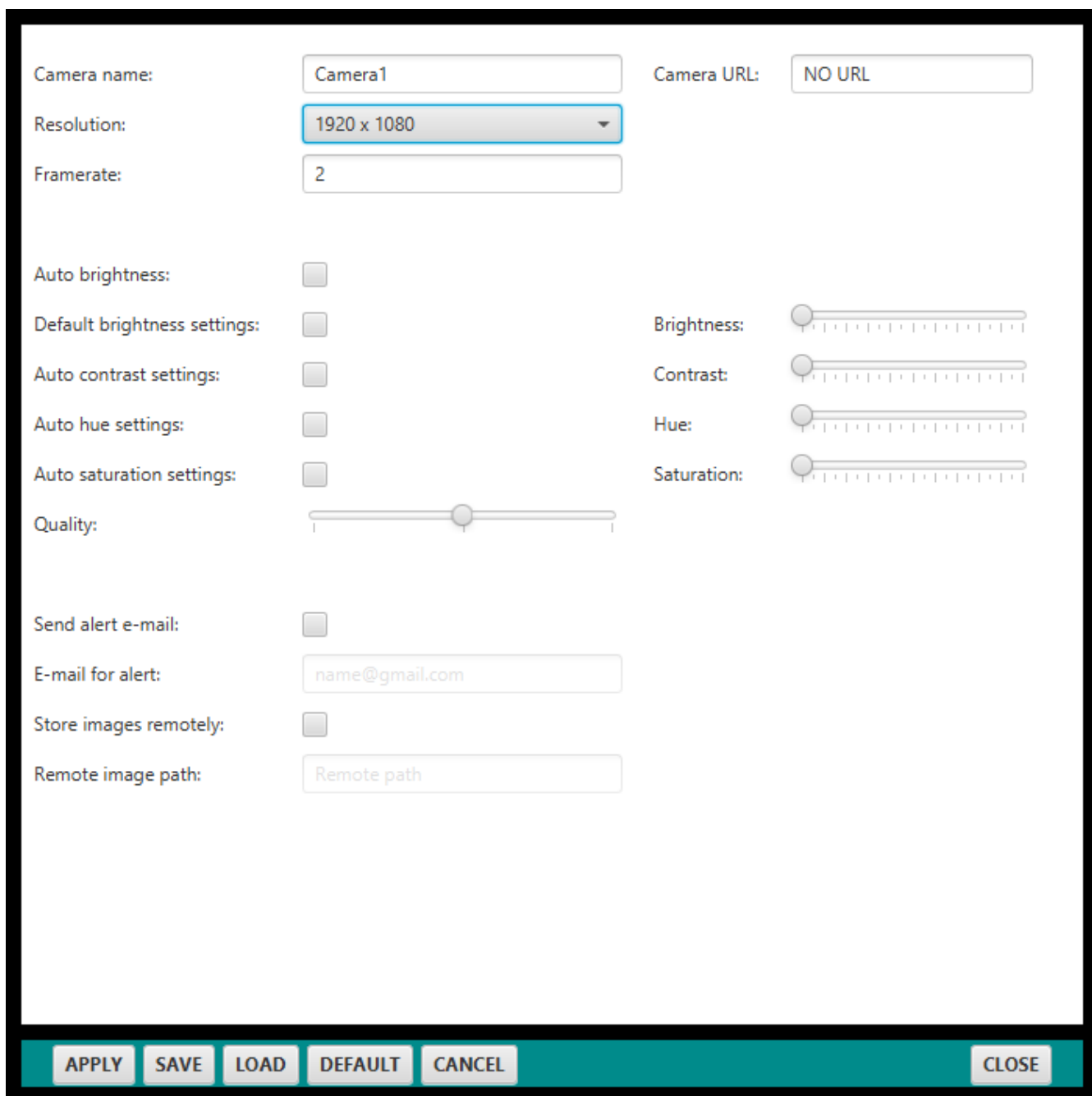


Obr. 6.1: Hlavní okno řídicí aplikace

6.8 Menu pro nastavení aplikace

V menu pro nastavení kamer se nachází jednotlivé slidery pro nastavení parametrů obrazu včetně místa pro vzdálené ukládání snímků a nastavení emailové adresy pro notifikaci při narušení objektu. Veškeré slidery je možné vyřadit pomocí zaškrtačkových polí pro automatické nastavení jednotlivých vlastností.

Ve spodní části menu se nachází tlačítka pro uložení, aplikování a resetování nastavené konfigurace a tlačítka pro ukončení aplikace a také tlačítka pro vytvoření souboru pro uložení současného nastavení a načtení tohoto uloženého nastavení.



The screenshot displays a settings menu for a camera application. It features several input fields and sliders. At the top, 'Camera name' is set to 'Camera1' and 'Camera URL' is 'NO URL'. 'Resolution' is set to '1920 x 1080' and 'Framerate' is '2'. Below these are five auto-settings (brightness, contrast, hue, saturation) each with an unchecked checkbox. To the right of these are four sliders for 'Brightness', 'Contrast', 'Hue', and 'Saturation'. A 'Quality' slider is also present. At the bottom, there are checkboxes for 'Send alert e-mail' and 'Store images remotely', with an 'E-mail for alert' field containing 'name@gmail.com' and a 'Remote image path' field containing 'Remote path'. A teal bar at the very bottom contains buttons for 'APPLY', 'SAVE', 'LOAD', 'DEFAULT', 'CANCEL', and 'CLOSE'.

Obr. 6.2: Menu pro nastavení kamer pomocí řídicí aplikace

7

Popis instalace kamerového systému a potřebného software

7.1 Instalace operačního systému na RaspberryPi

Pro instalaci systému je potřeba paměťové karty s nainstalovaným operačním systémem Raspbian popřípadě s pomocným instalátorem NOOBS a připojeným wi-fi adaptérem. Po instalaci operačního systému je třeba připojit Raspberry Pi k monitoru a nastavit wi-fi k naší používané síti.

Pro nastavení wi-fi je potřeba otevřít soubor pro síťové rozhraní.

```
1 sudo nano /etc/network/interfaces
```

V tomto souboru je potřeba upravit jeho obsah pro automatické připojení k síti.

```
1 auto lo
2
3 auto eth0
4 iface lo inet loopback
5 iface eth0 inet dhcp
6
7 auto wlan0
8 allow-hotplug wlan0
9 iface wlan0 inet dhcp
```

Pro zadání samotných přihlašovacích údajů je potřeba upravit soubor wpa_ supplicant.

```
1 sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

Do tohoto souboru je potřeba doplnit přihlašovací údaje k wi-fi.

```
1 network={
2     ssid="nazev site"
3     psk="wi-fi heslo"
4 }
```

Popřípadě je možné se připojit pomocí ethernetového kabelu, zde odpadá nastavení informací sítě a k připojení dojde automaticky.

Po nastavení wi-fi, nebo po připojení pomocí ethernetu se může monitor odpojit a může se k Raspberry Pi připojit pomocí terminálu. V tomto případě použijeme pro připojení

software Putty popřípadě jiného SSH klienta se připojí pomocí přidělené ip adresy pomocí portu pro SSH 22. Při vytvoření spojení se přihlásí do RaspberryPi přihlašovacím jménem a heslem (defaultně je přihlašovací jméno pi a heslo raspberry). Po úspěšném přihlášení je třeba aktualizovat Raspberry Pi.

Software Putty je open source SSH klient pro windows a unixové platformy, který je zde využíván ke vzdálenému připojení k jednotlivým Raspberry Pi.

7.2 Instalace softwaru na kamerovém modulu

Instalace software Motion probíhá pomocí balíčků z repozitáře.

```
1 sudo apt-get install motion
```

Po nainstalování tohoto software je nutné doinstalovat ručně potřebné knihovny a rozbalit je do dočasného adresáře.

```
1 /tmp
2 sudo apt-get install -y libjpeg62 libjpeg62-dev libavformat53 libavformat-dev libavcodec53 libavcodec-dev
3 libavutil51 libavutil-dev libc6-dev zlib1g-dev libmysqlclient18 libmysqlclient-dev libpq5 libpq-dev
4
5 wget https://www.dropbox.com/s/xdfcxm5hu71s97d/motion-mmал.tar.gz
6
7 tar zxvf motion-mmал.tar.gz
```

A poté je přesunout do předinstalovaného adresáře.

```
1 sudo mv motion /usr/bin/motion
2 sudo mv motion-mmалcam.conf /etc/motion.conf
```

Po přesunutí je potřeba nastavit přístupová práva pro adresáře.

```
1 sudo chmod 664 /etc/motion.conf
2 sudo chmod 755 /usr/bin/motion
3 sudo touch /tmp/motion.log
4 sudo chmod 775 /tmp/motion.log
```

Tato metoda instalace byla převzata z [4].

Po instalaci software Motion se nainstaluje server Samba pro snazší identifikaci Raspberry pi v síti. Po instalaci tohoto software se změní jméno a heslo v raspi-config na bezpečnější.

```
1 sudo apt-get install samba
```

7.3 Instalace softwaru na řídicím modulu

Instalace na řídicím modulu probíhá stejně jako u kamerového modulu. Na řídicím modulu není nainstalován software Motion, ale je zde nainstalováno runtime prostředí jazyku Java jre a ftp server pro ukládání snímků ze vzdálených kamerových modulů server vsftpd (Very Secure FTP Daemon).

Pro instalaci runtime prostředí jazyka Java je možné zvolit oficiální prostředí jdk od firmy oracle nebo jeho open source ekvivalent OpenJDK. Instalaci je možné provést

pomocí standardních linuxových balíčků nacházejících se v repozitáři pro jdk popřípadě použít odkazu na stránkách openJDK pro stažení potřebné verze. V tomto případě je provedena instalace pomocí balíčků.

```
1 sudo apt-get install oracle-java8-jdk
```

Pro vzdálený přístup pro správu aplikace je potřeba nainstalovat web server v tomto případě web server nginx, který se nachází v repozitáři a je možné jej nainstalovat pomocí standardních balíčků. Konfigurace probíhá nastavením konfiguračního souboru v adresáři programu. Zde je potřeba umístit distribuované soubory řídicí aplikace a generovanou webovou stránku.

Instalace ftp serveru vsftpd probíhá obdobným způsobem jako u web serveru ngingx, poté je potřeba nastavit zdrojový adresář pro ftp server a přihlašovací údaje pro uživatele v konfiguračních souborech.

8

Návrh dalšího rozvoje kamerového systému

8.1 Potřebná vylepšení kamerového systému

Nejdůležitějším vylepšením software by bylo vytvoření nové komponenty pro zobrazení MJPEG streamu tak, aby bylo dosaženo plynulého obrazu a zamrznání softwaru při změně a načítání nového snímku. Odstranění zamrznání by bylo možné odstranit upravením této aplikace na vícevláknovou. A vytvoření nové komponenty na zobrazování MJPEG snímků pro Javu FX.

Další možností jak vyřešit tento problém by bylo vypuštění software motion a získávání streamu přímo z webkamery popřípadě vytvoření vlastní streamovací aplikace. V tomto případě by bylo potřeba vyřešit detekci pohybu, která by byla možná řešit pomocí knihovny Webcam capture popřípadě vlastního řešení detekce pohybu.

Řešení končící podpory Javy FX na Raspberry Pi by bylo možné buď přechodem na projekt Open JFX při ponechání programovacího jazyku Java. V případě nevhodnosti jazyku Java a nekompletnosti projektu Open JFX by byl nejlepší přechod na jiný programovací jazyk, ať už C# a projekt Mono či C++ a odpovídající grafický framework. Popřípadě by bylo možné použít Javu jako back-end webové aplikace kdy by se na řídicím Raspberry Pi vytvořil web server a jako front-end by se využila webová stránka Html 5 v kombinaci s JavaScriptem. Toto řešení by bylo pravděpodobně nejvhodnější a vyžadovalo by nejmenší úsilí, neboť je možné zobrazit MJPEG stream ve webovém prohlížeči. V tomto případě se, ale zbavíme výhodné vlastnosti Javy FX což je možnost spuštění samotné aplikace vzdáleně v prohlížeči.

Další možností vylepšení software by bylo upravení aplikace pro mobilní zařízení jako jsou tablety a mobilní telefony. Toto by bylo možné zejména upravením jednotlivých menu pro dotykové ovládání. Vhodné by také bylo vytvoření multiplatformního instalátoru pro zjednodušení pro uživatele a jednodušší kontrolu vyžadovaného předinstalovaného software a vytvoření kompletního balíčku pro zjednodušení distribuce.

Upravením menu pro nastavení by bylo možné také kontrolovat parametry detekce pohybu software Motion, nicméně v současné aplikaci tyto funkce nejsou otestovány.

Výhodná úprava by také byla možnost přidávání většího množství kamerových modulů a následná úprava grafického rozhraní podle počtu, nicméně tato možnost by vyžadovala důsledné testování z důvodu omezení výkonem Raspberry Pi.

Pro zlepšení výkonu Raspberry Pi by bylo vhodné provést optimalizace a odstranit nepotřebný předinstalovaný software a popřípadě provést přetaktování.

9

Závěr

Tato práce se zabývá návrhem kamerového systému založeném na Raspberry Pi složeném ze dvou kamerových modulů Raspberry Pi a jednoho řídicího modulu Raspberry Pi s řídicí aplikací. Práce se zabývá zejména vývojem řídicí aplikace a popisem zvolených metod a její funkcí.

V obecné části je řešena rešerše podobných řešení a srovnání s komerčními systémy. Dále se zabývá návrhem řídicí aplikace pro tento kamerový systém a popisuje použité hardwarové prvky jako je Raspberry Pi a přídatné moduly. U softwarové části je odůvodněno použití programovacího jazyku Java a ovládacího software Motion pro kamery.

V praktické části se práce zabývá samotnou realizací řídicí aplikace a řešení problémů komunikace mezi jednotlivými moduly, zobrazování obrazu a nastavování samotných kamer a jsou zde popsány jednotlivé použité knihovny spolu s popisem jejich činnosti. Aplikace je vytvořena v jazyce Java s použitím specializovaných knihoven pro potřeby aplikace a software Motion pro záznam snímků a detekci pohybu. V této práci nebyla řešena samotná detekce pohybu, získávání a zaznamenávání snímků, tyto problémy jsou řešeny pomocí software Motion.

Samotná aplikace v současné době trpí řadou problémů, způsobenými ukončením oficiální podpory použitého grafického frameworku pro vytvoření aplikace Java FX. Použité náhradní řešení podstatně zpomaluje software zejména v oblasti zobrazování obrazu, opravám těchto problémů byla věnována samostatná kapitola, stejně jako možným vylepšením software a budoucím úpravám.

Současná verze software je schopna automaticky detekovat pohyb a vytvářet jeho záznam, který je možné ukládat na vzdálený server a je schopná informovat uživatele při jeho detekci pomocí e-mailu. Momentálně je možná vzdálená správa jednotlivých kamerových modulů pomocí hlavního řídicího modulu s připojenými ovládacími periferiemi. Vzdálená správa samotného řídicího software pomocí web serveru se momentálně potýká s problémy s Java certifikáty, nicméně samotná aplikace je pro vzdálenou správu připravena.

V současné době je aplikace stále ještě ve vývoji a obsahuje pouze základní stabilní funkce, které bylo z časových důvodů možné otestovat a implementovat.

V porovnání s komerčními řešeními tento systém nedosahuje ještě jejich kvalit, nicméně

se jim velmi blíží a díky vysoké rozšiřitelnosti a upravitelnosti by po nutných úpravách mohl konkurovat obdobným systémům v podobné cenové kategorii.

Literatura

- [1] MAIAKAAT(nickname). *Front of Raspberry Pi* In: Wikimedia commons[online]. [Cit. 23.5.2015]. Dostupné z WWW: http://upload.wikimedia.org/wikipedia/commons/9/90/Front_of_Raspberry_Pi.jpg
- [2] MULTICHERRY(nickname). *Raspberry Pi 2 Model B v1.1 top new* In: Wikimedia commons[online]. [Cit. 23.5.2015]. Dostupné z WWW: http://commons.wikimedia.org/wiki/File:Raspberry_Pi_2_Model_B_v1.1_top_new.jpg
- [3] MATT(nickname). *pi_noir_camera_module_07* In: raspberrypi-spy.co.uk[online]. [Cit. 23.5.2015]. Dostupné z WWW: http://www.raspberrypi-spy.co.uk/wp-content/uploads/2013/10/pi_noir_camera_module_07.jpg
- [4] BUENGER Christoph. *Raspberry Pi as low-cost HD surveillance camera*[online]Release Date: 12 May 2014 [Cit. 23.5.2015]. Dostupné z WWW: <http://www.codeproject.com/Articles/665518/Raspberry-Pi-as-low-cost-HD-surveillance-camera>
- [5] RASPBERRY PI FOUNDATION. *RASPBERRY PI HARDWARE*[online][Cit. 23.5.2015]. Dostupné z WWW: <https://www.raspberrypi.org/documentation/hardware/raspberrypi/README.md>
- [6] EMBEDDED LINUX WIKI. *BCM2835 datasheet errata*[online]Release Date: 8 December 2014 [Cit. 23.5.2015]. Dostupné z WWW: http://elinux.org/BCM2835_datasheet_errata
- [7] BROADCOM CORPORATION. *BCM2835 ARM Peripherals* [online]Release Date: 06 February 2012 [Cit. 23.5.2015]. Dostupné z WWW: <https://www.raspberrypi.org/wp-content/uploads/2012/02/BCM2835-ARM-Peripherals.pdf>
- [8] RASPBERRY PI FOUNDATION. *CAMERA*[online][Cit. 23.5.2015]. Dostupné z WWW: <https://www.raspberrypi.org/documentation/hardware/camera.md>
- [9] WIKIPEDIA. *Java (programming language)*[online][Cit. 23.5.2015]. [http://en.wikipedia.org/wiki/Java_\(programming_language\)](http://en.wikipedia.org/wiki/Java_(programming_language))

- [10] ORACLE CORPORATION. *Java SE Embedded 8 Compact Profiles Overview*[online][Cit. 23.5.2015]. Dostupné z WWW: <http://www.oracle.com/technetwork/java/embedded/resources/tech/compact-profiles-overview-2157132.html>
- [11] TosiaraT. *Motion*[online]Release Date: 12 Apr 2015 [Cit. 23.5.2015]. Dostupné z WWW: <http://www.lavrsen.dk/foswiki/bin/view/Motion/WebHome>
- [12] WIKIPEDIA. *Motion JPEG*[online]Release Date: 27 January 2015 [Cit. 23.5.2015]. http://en.wikipedia.org/wiki/Motion_JPEG
- [13] WIKIPEDIA. *Internet Control Message Protocol*[online]Release Date: 17 May 2015 [Cit. 23.5.2015]. http://en.wikipedia.org/wiki/Internet_Control_Message_Protocol
- [14] ICMP 4 J. *Internet Control Message Protocol for Java*[online][Cit. 23.5.2015]. <http://www.icmp4j.org/>
- [15] WIKIPEDIA. *Secure Shell*[online]Release Date: 17 May 2015 [Cit. 23.5.2015]. http://en.wikipedia.org/wiki/Secure_Shell
- [16] JCRAFT INC. *JSch - Java Secure Channel*[online][Cit. 23.5.2015]. <http://www.jcraft.com/jsch/>
- [17] JCRAFT INC. *Jsch licence*[online][Cit. 23.5.2015]. <http://www.jcraft.com/jsch/LICENSE.txt>
- [18] SAVILLE Richard. *Turn a Raspberry Pi into a CCTV Security System*[online][Cit. 23.5.2015]. Dostupné z <http://www.averagemanvsraspberrypi.com/2014/09/turn-raspberry-pi-into-cctv-security.html>
- [19] BARTOSZ Firyn. *Webcam Capture*[online] [Cit. 23.5.2015]. Dostupné z WWW: <http://webcam-capture.sarxos.pl/>
- [20] VAN ELST Remy. *Setting up Motion with FTP and Email! support*[online]Release Date: 9 April 2008 [Cit. 23.5.2015]. Dostupné z WWW: <https://raymii.org/s/tutorials/Setting-up-motion-with-ftp-and-email-support.html>