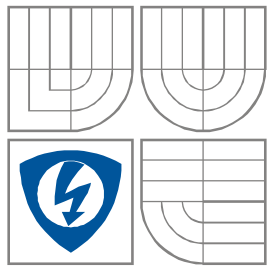


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

BEZPEČNOSTNÍ SYSTÉM S VYUŽITÍM MOBILNÍCH SÍTÍ

SECURITY SYSTEM USING MOBILE NETWORKS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

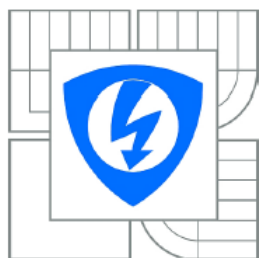
AUTOR PRÁCE
AUTHOR

Adam Svoboda

VEDOUCÍ PRÁCE
SUPERVISOR

prof. Ing. Stanislav Hanus, CSc.

BRNO, 2015



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav radioelektroniky

Diplomová práce

magisterský navazující studijní obor
Elektronika a sdělovací technika

Student: Bc. Adam Svoboda

ID: 134619

Ročník: 2

Akademický rok: 2014/2015

NÁZEV TÉMATU:

Bezpečnostní systém s využitím mobilních sítí

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte možnosti přenosu obrazových signálů v celoplošných mobilních systémech GSM, UMTS a LTE, dále formáty výstupních obrazových signálů z web kamer a typy čidel pro indikaci pohybu. Na základě získaných poznatků navrhnete blokové a podrobné schéma bezpečnostního systému pro monitorování uzavřeného prostoru se zaměřením na přenos videosekvence, případně i doprovodného zvukového signálu. Navržený systém realizujte a jeho funkčnost otestujte. Po informaci z pohybového čidla budou do mobilního zařízení zasílány krátké videosekvence z web kamery v nastavitelných časových intervalech. Současně musí být zaznamenána informace o času a proveden záznam videosekvence. Uživatel systému musí být o narušení prostoru ihned informován.

DOPORUČENÁ LITERATURA:

[1] GLISIC, S. G. Advanced Wireless Communications. 4G Technologies. England: J. Wiley & Sons, 2004.

[2] ANGELIDES, M. C., AGIUS, H. The Handbook of MPEG Applications. Standards in Practice. United Kingdom: J. Wiley & Sons, 2011.

Termín zadání: 9.2.2015

Termín odevzdání: 21.5.2015

Vedoucí práce: prof. Ing. Stanislav Hanus, CSc.

Konzultanti diplomové práce:

doc. Ing. Tomáš Kratochvíl, Ph.D.

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

Tato diplomová práce se zabývá návrhem zabezpečovacího systému, který bude pořízené videozáznamy přenášet pomocí mobilních systémů. Jsou zde vybráni zástupci kamer a pohybových senzorů, které by se při konstrukci mohli použít. Práce se dále zabývá teoretickým návrhem přenosové soustavy, která bude složena z kamery, pohybového čidla, USB modemu a mikroprocesoru, který bude celou činnost řídit. Je zde navrženo blokové schéma a způsob komunikace. Celý bezpečnostní systém je zkonstruován a otestován v různých podmínkách.

KLÍČOVÁ SLOVA

USB modem, PIR, IP kamera, Arduino, FTP

ABSTRACT

This master's thesis deals with design of security system, which transmits captured video using mobile systems. There are chosen representatives of cameras and motion sensors that could be used in the construction. The thesis also deals with the theoretical design of the transmission system, which consists of a camera, a motion sensor, an USB modem and a microprocessor, that manages all operations. There is also a proposed block diagram of a communication method. The security system is designed and tested in various conditions.

KEYWORDS

USB modem, PIR, IP camera, Arduino, FTP

SVOBODA, A. *Bezpečnostní systém s využitím mobilních sítí*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky, 2015. 95 s. Diplomová práce. Vedoucí práce: prof. Ing. Stanislav Hanus, CSc.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma Bezpečnostní systém s využitím mobilních sítí jsem vypracoval samostatně pod vedením vedoucího diplomové 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 diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne

.....

(podpis autora)

Poděkování

Děkuji vedoucímu diplomové práce prof. Ing. Stanislavu Hanusovi, CSc. za pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne 21. května 2015

.....

podpis autora

OBSAH

Seznam obrázků	viii
Seznam tabulek	x
Úvod	1
1 Pohybové senzory	2
1.1 Perimetrická ochrana (0. linie).....	2
1.1.1 Mikrofonní kabely	2
1.1.2 Infračervené závory	2
1.1.3 Mikrovlnné bariéry	2
1.2 Plášťová ochrana (1. linie)	2
1.3 Prostorová ochrana (2. linie).....	3
1.3.1 PIR čidla	3
1.3.2 Mikrovlnná čidla (MW).....	6
1.3.3 Ultrazvuková čidla (US).....	8
1.4 Vyráběné typy pohybových senzorů.....	8
2 Kamery	12
2.1 CMOS (Complementary Metal-Oxide-Semiconductor).....	12
2.2 CCD (Charged Coupled Device) –	13
2.3 Rozlišení a komprese	16
2.4 Vyráběné typy kamer	19
3 Přenos dat	25
3.1 Modem Huawei E398	25
3.2 Systém UMTS.....	25
3.3 Systém LTE.....	27
3.3.1 UPLINK	29
3.3.2 DOWNLINK	30
3.3.3 Systém IMS	30
3.3.4 Protokol IP.....	30
4 Návrh zapojení bezpečnostního systému	32
4.1 Komunikace PIR senzoru s deskou Arduino	34
4.2 Komunikace IP kamery s deskou Arduino	34

4.3	Práce s ATmega32u4	35
4.4	Práce s programem FtpApp.....	40
4.4.1	Nastavení parametrů Arduina.....	41
4.4.2	Nastavení údajů FTP serveru.....	42
4.4.3	Výpis souborů.....	43
4.4.4	Podrobnosti vybraného souboru	43
4.4.5	Uživatelská tlačítka	44
5	Testování bezpečnostního systému	46
5.1	Zapojení bezpečnostního systému pro domácnost.....	46
5.1.1	Instalace zařízení	47
5.1.2	Kvalita videa.....	50
6	Návod na instalaci systému	55
6.1	Návod pro nastavení Arduina YUN.....	55
6.1.1	Rozšíření úložného prostoru.....	55
6.1.2	Instalace ovladačů 3g modemu	56
6.1.3	Konfigurace 3G připojení.....	56
6.1.4	Konfigurace IP kamery.....	57
6.1.5	Nahrání programu do Arduina.....	58
6.1.6	Sestavení systému.....	59
6.2	Návod na ovládání obslužného programu FtpApp	62
6.3	Návod na vytvoření FTP serveru	63
7	Závěr	65
	Literatura	67
	Seznam symbolů, veličin a zkratk	69
	Příloha	71
	Zdrojový kód.....	71
	Program FtpApp	71
	Program v desce Arduino YUN	78
	Program na odesílání SMS zpráv	84
	Obsah příloženého CD.....	85

SEZNAM OBRÁZKŮ

Obrázek 1.1:	Magnetický kontakt [27]	3
Obrázek 1.2:	PIR senzor s usměrňovací čočkou	4
Obrázek 1.3:	Typické zapojení PIR senzoru [4]	5
Obrázek 1.4:	Vyhodnocení průchodu člověka PIR senzorem	5
Obrázek 1.5:	Fresnelova čočka	6
Obrázek 1.6:	Mikrovlnný senzor.....	7
Obrázek 1.7:	a) Senzor LH-941A b) Senzor LS-818-3.....	9
Obrázek 1.8:	a) Senzor LH-930A(II) b) Senzor G1020.....	10
Obrázek 2.1:	Struktura CMOS senzoru [12].....	13
Obrázek 2.2:	Blokové schéma zapojení IP kamery	15
Obrázek 2.3:	Sekvence snímků	18
Obrázek 2.4:	Kodér standardu H.264 [17]	19
Obrázek 2.5:	Webkamera Creative [28].....	20
Obrázek 2.6:	a) AirCam IP-200PHD-24 b) AirCam POE-5010HD [29]	21
Obrázek 2.7:	a) Foscam FI9821W b) Foscam FI9826W [30]	23
Obrázek 3.1:	Modem Huawei E398.....	25
Obrázek 3.2:	Architektura UMTS	26
Obrázek 3.3:	Architektura LTE [20].....	28
Obrázek 3.4:	Princip OFDMA	29
Obrázek 4.1:	Základní blokové schéma bezpečnostního systému	32
Obrázek 4.2:	Blokové schéma Arduina [26].....	33
Obrázek 4.3:	Podrobné blokové schéma bezpečnostního systému	33
Obrázek 4.4:	Vývojové prostředí Visual Studio	41
Obrázek 4.5:	Informace vybraného souboru	43
Obrázek 5.1:	Umístění bezpečnostního systému v místnosti.....	48
Obrázek 5.2:	Zapojený bezpečnostní systém	48
Obrázek 5.3:	Podrobný popis ukrytých periférií.....	49
Obrázek 5.4:	Testování bezpečnostního systému - Foscam.....	51
Obrázek 5.5:	Testování bezpečnostního systému – Canon	51
Obrázek 5.6:	Testování bezpečnostního systému – Foscam	52
Obrázek 5.7:	Testování bezpečnostního systému – Canon	52

Obrázek 5.8:	Testování bezpečnostního systému – Foscam.....	53
Obrázek 5.9:	Testování bezpečnostního systému – Canon.....	53
Obrázek 5.10:	Detail tváře ze zařízení a) Foscam b) Canon.....	54
Obrázek 6.1:	Úvodní stránka operačního systému v Arduino YUN.....	55
Obrázek 6.2:	Konfigurace nového rozhraní.....	57
Obrázek 6.3:	Prostředí kamery Foscam.....	58
Obrázek 6.4:	Nastavení fixní IP adresy.....	58
Obrázek 6.5:	PuTTY – konfigurace SSH.....	59
Obrázek 6.6:	Zapojení pohybového senzoru.....	60
Obrázek 6.7:	Pomocné napájení USB modemu.....	60
Obrázek 6.8:	Kompletní bezpečnostní systém.....	61
Obrázek 6.9:	Aplikace pro přehled videí.....	62
Obrázek 6.10:	Program FileZilla Server.....	64

SEZNAM TABULEK

Tabulka 1.1: Parametry modulu SB00612A-2	8
Tabulka 1.2: Parametry senzoru LH-941A	9
Tabulka 1.3: Parametry senzoru LS-818-3	9
Tabulka 1.4: Parametry senzoru LH-930A(II).....	10
Tabulka 1.5: Parametry modulu G1020	10
Tabulka 2.1: Porovnání CCD a CMOS snímačů	14
Tabulka 2.2: Rozlišení obrazu.....	16
Tabulka 2.3: Porovnání kompresních metod	17
Tabulka 2.4: Parametry kamery Creative.....	20
Tabulka 2.5: Parametry kamery AirCam IP-200PHD-24	21
Tabulka 2.6: Parametry kamery AirCam POE-5010HD.....	22
Tabulka 2.7: Parametry kamery Foscam FI9821W	22
Tabulka 2.8: Parametry kamery Foscam FI9826W	23
Tabulka 3.1: Rozložení kmitočtů 20. pásma	29
Tabulka 4.1: Základní parametry ATmega32u4 [26]	32
Tabulka 5.1: Proudový odběr modemu Huawei	46
Tabulka 5.2: Časy procesů bezpečnostního systému	47
Tabulka 5.3: Přenosové rychlosti různých zařízení	50

ÚVOD

Diplomová práce se zabývá návrhem speciálního bezpečnostního systému, ve kterém bude mimo obvyklých částí, jako jsou kamera a pohybový senzor, použit ještě USB dongle, který bude zaznamenaná videa posílat přes mobilní sítě UMTS nebo LTE na server, kde si je bude možno prohlédnout. V dnešní době, kdy mobilní operátoři nabízejí neomezené tarify a vysokorychlostní internet, je toto řešení velmi výhodné, protože při instalaci kamery nejsme omezeni délkou ethernetového kabelu, který musíme do kamery připojit, ani dosahem Wi-Fi routeru. Jediná podmínka je, aby oblast byla pokryta vysokorychlostním internetem operátora.

Cílem diplomové práce je prostudovat možnosti přenosu obrazových signálů v celoplošných mobilních sítích GSM, UMTS a LTE, dále formáty výstupních obrazových signálů z kamer a typy čidel pro indikaci pohybu. Na základě těchto poznatků je navrženo blokové schéma systému pro monitorování narušení prostoru se zaměřením na přenos videosekvencí.

Celý systém bude ovládán vývojovou deskou Arduino YUN, která má USB konektor pro modem Huawei E398 a ethernetový konektor pro libovolnou IP kameru. Pokud pohybový senzor detekuje pohyb, mikroprocesor pošle do kamery požadavek na nahrávání videa na SD kartu. Čas, po který se bude video zaznamenávat, bude volitelný v programu, který je také součástí práce. Jakmile bude video uloženo v komprimovaném formátu na kartě, je pomocí modemu odesláno na vzdálený FTP server, kde bude dostupné pro uživatele.

Diplomová práce se skládá ze čtyř hlavních částí. První blok je věnován teorii pohybových senzorů, jejich rozdělení a konkrétní typy dostupné v obchodech, dále rozdělení kamer podle druhu snímače, kodeků, které používají, nebo podle rozlišení obrazu. Opět je vybráno několik zástupců dostupných v obchodech a jejich rozbor. Poslední část prvního bloku je věnována mobilní komunikaci. Jsou zde popsány jednotlivé mobilní standardy a blíže je popsán USB modem Huawei E398, který je v diplomové práci používán. Druhý blok je věnován podrobnějšímu popisu komunikace mikroprocesoru s jednotlivými částmi bezpečnostního systému. Jsou zde uvedeny jednotlivé operace potřebné pro komunikaci s kamerou, získání videa z kamery v potřebném formátu nebo synchronizace s internetovým časem. Dále je podrobně rozepsán postup ukládání videa na paměťovou kartu a následné kopírování na vzdálený server. Součástí diplomové práce je také program pro přehlednou práci s videi, které již jsou nahrány na vzdáleném serveru a podrobný popis jeho funkce. Ve třetím bloku je shrnuto testování bezpečnostního systému, rychlost odesílání dat a porovnání kvality videa odeslaného na server s videem pořízeným jiným záznamovým zařízením. V posledním bloku je podrobně rozepsána instalace bezpečnostního systému. Jsou zde uvedeny jednotlivé kroky vedoucí ke správné funkčnosti celého bezpečnostního systému a fotografie kompletního zapojení.

1 POHYBOVÉ SENZORY

Při vybírání střežícího systému záleží na objektu, který má být střežen (auto, dům,...), na prostorách (uvnitř, venku,...) a v neposlední řadě i na peněžích, které chceme do zabezpečovacího systému investovat. Pro střežení jakéhokoliv objektu se v dnešní době využívá jedné ze čtyř úrovní ochran nebo jejich kombinací. Jsou to nejen senzory pohybu, ale i senzory tlaku, optické závory atd. V následujícím textu jsou stručně popsány jednotlivé úrovně zabezpečení.

1.1 Perimetrická ochrana (0. linie)

Tato čidla dokáží signalizovat narušení vnějších prostorů (zahrada, parkoviště,...) a neobejdou se bez prostorové ochrany, jako je oplocení, na kterou se obvykle tato ochrana použije. Při použití této ochrany se musí dbát na krytí před klimatickými vlivy a na správný způsob montáže. Perimetrická čidla mají dosah v řádu stovek metrů a je velmi důležité, aby byla použita po celém obvodu objektu.

1.1.1 Mikrofonní kabely

Vyhodnocují mechanické namáhání kabelu, který je obvykle umístěn na plotě nebo v omítce zdi. Signál z kabelu se převede na akustický a lze z něj analyzovat, o jaký druh deformace se jedná. Jestli je to vítr, který na plot fouká, nebo nežádaná osoba, která se plot snaží překonat. Výhodou je jednoduchá montáž a relativně nízké náklady, nevýhodou jsou však časté falešné poplachu.

1.1.2 Infračervené závory

Infračervené závory jsou nejpoužívanější perimetrická ochrana. Pro použití je třeba vysílač a přijímač infračerveného záření. Mezi těmito komponentami může být jeden nebo více paprsků. Pokud se jeden z těchto paprsků přeruší, může být vyvolán poplach. Obvykle se využívá paprsků více a pro vyvolání poplachu je nutné přerušit alespoň dva paprsky. Předchází se tím falešným poplachům, které způsobí malá zvířata.

1.1.3 Mikrovlonné bariéry

Jde o vytvoření elektromagnetického pole mezi přijímačem a vysílačem, ve tvaru elipsoidu, přičemž vniknutí do tohoto prostoru způsobí změnu elektromagnetického pole a dojde k vyvolání poplachu. Výhodou je velká odolnost proti změnám počasí a povětrnostním vlivům.

1.2 Plášťová ochrana (1. linie)

Plášťová ochrana se stará o zabezpečení pláště objektu – okna, dveře apod. Nejčastěji jsou používány klasické magnetické kontakty (viz obrázek 1.1). Magnetický kontakt se

skládá ze dvou částí – permanentního magnetu a jazýčkového kontaktu. V klidovém stavu jsou obě části blízko u sebe a kontakt je sepnut. V případě oddálení se jazýčkový kontakt rozezne a vyvolá poplach. Tato ochrana nejčastěji hlídá zavřená okna a dveře.

Dalším typem plášťové ochrany jsou senzory pro ochranu skleněných ploch. Ty mohou být kontaktní nebo akustické. Kontaktní senzor, který je na skle přilepený, sleduje vlnění skla. Senzor akustický obsahuje elektretový mikrofon a vyhodnocuje zvuk. Při rozbití skla rozpozná tříštivý zvuk a vyvolá poplach.



Obrázek 1.1: Magnetický kontakt [27]

1.3 Prostorová ochrana (2. linie)

Doplňkem perimetrické a plášťové ochrany může být ochrana prostorová. Lze ji použít zcela samostatně i bez předešlých dvou ochranných systémů. Tato ochrana se také nazývá pohybová, protože detektory jsou citlivé na změny vlnění způsobené pohybem objektů. Obvykle se používají v centrálních budovách, halách nebo na chodbách, kde plní nejen ochrannou funkci, ale také mohou například rozsvěcovat světla.

Nejvíce používané senzory pohybu jsou PIR senzory (Passive Infra Red – pasivní infračervené), US (Ultra Sound – ultrazvukové) a MW (Micro Wave – mikrovlnné). Jednotlivé senzory lze mezi sebou kombinovat, přičemž nejčastějšími kombinacemi jsou PIR-US a PIR-MW. Každé z čidel využívá ke své funkci jinou část frekvenčního spektra elektromagnetického vlnění.

1.3.1 PIR čidla

PIR čidla jsou levná, malá a jednoduchá na používání. Díky tomu jsou nejpoužívanější jak v domácnostech, tak i ve firmách. Lze se také setkat s názvy jako „Pyroelectric sensor“ nebo „IR motion sensor“.

Tyto senzory jsou pasivní. To znamená, že zařízení negeneruje žádné záření ani jinou energii, ale pouze detekuje záření ostatních objektů. Jsou založeny na principu zachycení změny v infračervené části frekvenčního spektra elektromagnetického vlnění. Využívají toho, že každé těleso, které má teplotu větší než absolutní nula, emituje

tepelnou energii ve formě záření. Toto záření je pro člověka neviditelné, ale může být detekováno elektrickými zařízeními, která jsou pro toto frekvenční spektrum navržena. Vlnová délka vyzářená lidským tělem, které má 37 °C, je 9,344 μm [1]. Záření s takovými vlnovými délkami nemůže projít některými materiály, kterými by prošlo záření s vlnovou délkou viditelného světla. Takovým případem je obyčejné sklo nebo plast. Nicméně může procházet jinými látkami, kterými zase neprojde světlo viditelné. To mohou být germanium a křemík. Těchto materiálů se využívá k eliminaci světla a větru, které by při venkovním použití mohly ohrozit správnou funkci (viz obrázek 1.2). Dalším materiálem, kterým projde vlnění s vlnovou délkou okolo 9,4 μm, je polyetylen. Ten se používá jako čočka, která usměrní infračervené záření do senzoru.



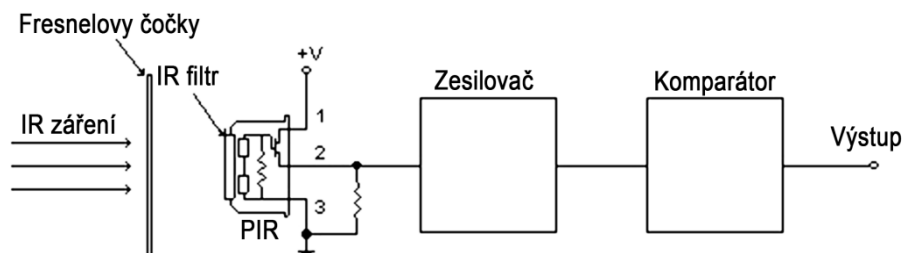
Obrázek 1.2: PIR senzor s usměrňovací čočkou

Čidlo funguje na principu pyroelektrického jevu. Podobně jako mechanickým tlakem lze deformovat krystalickou mřížku dielektrik, můžeme tuto deformaci vyvolat i změnou teploty. V látkách s jednou polární osou symetrie lze změnou teploty vytvářet dipólový moment, jehož velikost je přímo úměrná změně teploty a má směr této polární osy symetrie. U pyroelektrického jevu vzniká a trvá dipólový moment pouze při změně teploty [2]. Obecně lze pyroelektrický jev definovat jako schopnost materiálu generovat dočasný elektrický potenciál při změně jeho teploty.

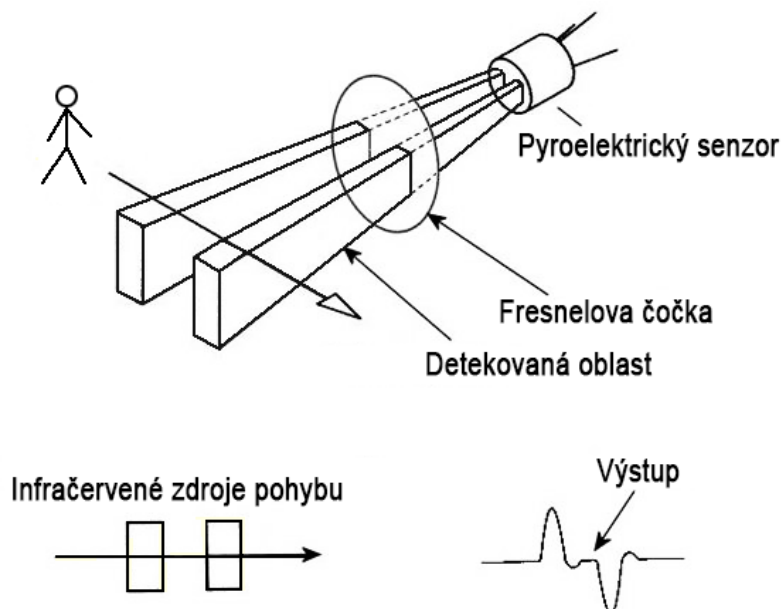
Pyroelektrický senzor je vyroben z krystalického materiálu. Pokud je vystaven teplu v podobě infračerveného záření, vytváří povrchový elektrický náboj. Pokud je velikost infračerveného záření dostatečně velká, je tento náboj měřen a zpracováván vysoce citlivým FET zařízením, které je uvnitř senzoru. Samotný senzor bývá citlivý na velké rozmezí vlnových délek, proto se používá filtr [4], který omezí detekované záření na rozmezí 8-14 μm (viz obrázek 1.3). Potom je senzor nejvíce citlivý na záření lidského těla. Typicky bývá druhý pin (source) této FET součástky připojen na zem přes pulldown rezistor, který má odpor přibližně 100 kΩ. Zesilovač je obvykle omezen pásmovou propustí pod 10 Hz, aby omezil vysokofrekvenční šum. Následuje komparátor, který reaguje na změnu výstupního signálu. Pro správnou funkci je třeba na drain FET tranzistoru připojit napětí 3-15 V.

Senzor má alespoň dva snímací prvky, které jsou zapojeny v „bucking“ režimu (opak „boost“ režimu). Toto uspořádání výrazně eliminuje rušivé signály způsobené vibracemi, změnami teplot a slunečním svitem. Pokud před senzorem projde člověk,

aktivuje první a pak druhý snímací prvek. Zatímco z jiných (rušivých) zdrojů je aktivní první i druhý snímač současně, a proto není vyhodnocen jako pohyb člověka. Zdroj infračerveného záření musí projít kolem čidel tak, aby vyvolal impulsy ve snímačích postupně (obrázek 1.4), obvykle horizontálně. Průběh výstupního signálu z analogového senzoru ukazuje, že při pohybu v jednom směru je impuls nejdříve kladný a potom záporný. Při pohybu v opačném směru by byl první impuls záporný, druhý kladný.



Obrázek 1.3: Typické zapojení PIR senzoru [4]

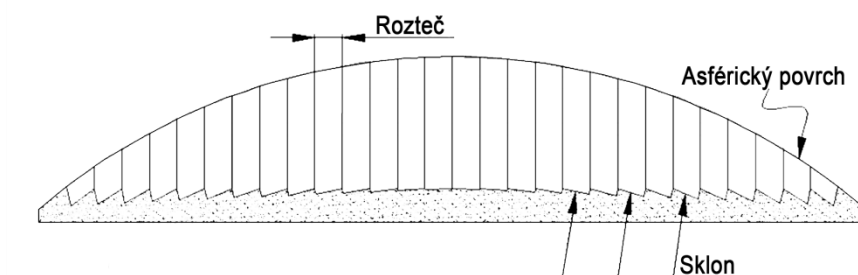


Obrázek 1.4: Vyhodnocení průchodu člověka PIR senzorem

Pokud není použita čočka před pyroelektrickým senzorem, senzor detekuje pouze infračervené záření, které je v blízkosti do cca 1 metru. IR záření působí na jeden pyroelement více než na druhý a na výstupu se objeví napětí odpovídající průchodu člověka. Pokud by byl zdroj záření dále, vyzařovací diagram je stále více rozmazaný a IR záření působí rovnoměrně na oba elementy. Důsledkem toho je, že na výstupu se neobjeví žádné napětí. Při použití Fresnelových čoček se výrazně zvětšuje pozorovatelná vzdálenost.

Fresnelova čočka (viz obrázek 1.5) je podobná jako Plano Convexova čočka (ploskovypuklá spojka) [6], ale je podstatně tenčí a díky tomu má i menší absorpční ztráty. Optické vlastnosti jsou však zachovány. Čočka zachytí infračervené záření a

zaměřuje ho na malé místo. Tento malý bod se projevuje jako zdroj IR záření. Při pohybujícím se zdroji záření se bod pohybuje na čidle a na výstupu vyvolává pohybové impulsy. Fresnelova čočka může rozšířit rozsah snímání asi na 100 m.



Obrázek 1.5: Fresnelova čočka

Při snímání střeženého objektu dopadá na pyroelektrický sensor infračervené záření v podobě spojitého, analogového signálu. U PIR detektoru je velmi důležité zpracování analogového signálu a jeho kvalita na výstupu. Zpracování signálů může být analogové nebo digitální [5]:

- Analogové senzory – signál zpracovávají tak, že vyhodnocují jeho úroveň a při každém překročení dojde k poplachu. Ve většině detektorů je toto vyhodnocování vylepšeno o počítadlo impulsů. Proto k poplachu dojde až po napočítání určitého počtu impulsů. I přesto je tento způsob náchylný na rychlé teplotní změny a dochází k planým poplachům.
- Digitální senzory – analogový signál je navzorkován a kvantován. Digitální zpracování signálu se nazývá multikriteriální. Při něm se vyhodnocuje nejen velikost signálu, ale i jeho strmost, polarita, časování, povaha energie a kmitočtové spektrum společně se statistickými informacemi, které byly zjištěny průzkumem prostředí. Dále je použito mnoho filtrů, které přispějí k lepšímu rozeznání skutečného poplachu od falešného [5].

1.3.2 Mikrovlnná čidla (MW)

Tato čidla (viz obrázek 1.6) spadají do kategorie aktivní ochrany, protože pro správnou funkci musí být využit vysílač i přijímač. Systém vysílá mikrovlnnou energii, kterou zcela pokryje oblast svojí detekce. Stejně tomu je i u ultrazvukových snímačů. Dioda emituje záření v mikrovlnném pásmu (frekvence 3 – 12 GHz) stejně, jako je tomu při využívání Dopplerova efektu. Jakékoliv narušení způsobené pohybem v detekovaném prostoru může aktivovat alarm. Navzdory mnohým obavám nemůže záření z těchto senzorů lidem ublížit ani nemá nepříznivý vliv na kardiostimulátory.



Obrázek 1.6: Mikrovlnný senzor

Tyto senzory mohou být monostatické, kde vysílač i přijímač jsou obsaženy v jednom zařízení, nebo bistatické, kde vysílač a přijímač jsou oddělené v jiných jednotkách. Každý typ má pro uživatele své výhody a nevýhody. Monostatická jednotka dokáže pomocí úzkého mikrovlnného paprsku detekovat oblast do vzdálenosti až 100 m. Může však nastavit tvar vyzařovaného paprsku na oválný, jehož paprsek je kratší, ale širší. Naopak bistatická jednotka nabízí rozsah detekce až 400 m, ale neumožňuje uživateli definovat oblasti detekce tak přesně, jak jednotka monostatická [7]. Bistatické senzory jsou také náchylnější na falešné poplachy.

Vysílač MW čidla vysílá s nepatrným výkonem v řádu jednotek mW elektromagnetické vlny s vlnovou délkou 2,5 - 10 cm. Přijímač přijímá odražené elektromagnetické vlnění a další elektronika vyhodnocuje kmitočet vzniklý složením (interferencí) vlnění vysílaného a přijímaného. Pokud se vyslané vlnění odráží od nepohyblivých objektů, je interference konstantní. Při odrazu od radiálně se pohybujícího objektu dochází ke změně kmitočtu odraženého vlnění a interferenční frekvence se změní (o jednotky Hz) - Dopplerův efekt. Rychlost a úroveň změn je porovnávána s předem nastavenými parametry a při jejich překročení je čidlem vyvolán poplach.

Nevýhodou těchto čidel je, že značná část vyslaného vlnění je absorbována předměty v střeženém prostoru a zpět k přijímači se odrazí menší signál, než byl vyslán. Z toho důvodu může senzor detekovat pohyb tam, kde není a naopak ho nedetekovat tam, kde by to bylo žádoucí. Další nevýhodou mohou být tzv. „mrtvé zóny“. Jsou to oblasti schované za velkými kovovými překážkami, přes které se paprsky nedostanou, a detektor nemůže pohyb detekovat. Tento problém je však typický pro velkou část pohybových a ochranných mechanismů. Na druhou stranu mikrovlny mohou projít zdí a mohou detekovat pohyb za stěnou. Na první pohled se to zdá jako velká výhoda tohoto snímače, ale má i své nevýhody. Jelikož jsou senzory extrémně citlivé, může dojít k falešným poplachům způsobeným venku vlajícím prádlem, člověkem procházejícím kolem domu apod. Dokonce i zářivkové osvětlení, které emituje částice světla, může vyvolat falešný poplach [7].

Vzhledem k tomu, že mikrovlnné senzory jde oklamat velmi pomalým pohybem, při kterém je přijímaná frekvence téměř totožná s vyslanou, nebo využitím „mrtvých zón“, doporučuje se tento typ detektoru kombinovat s detektory PIR, které pomalý pohyb neoklamají. PIR detektory jsou také citlivější na boční pohyb, MW senzory zase na pohyb směrem k čidlu. Tyto dva typy senzorů se vzájemně doplňují. Mnoho lidí však naprogramuje systém tak, že alarm se spustí až ve chvíli, kdy se aktivují obě čidla, což je chyba. Každý ze senzorů je citlivější na něco jiného a takové zapojení je méně spolehlivé.

1.3.3 Ultrazvuková čidla (US)

Společně s čidly mikrovlnnými patří do kategorie aktivních, Dopplerových čidel. Jejich princip je velice podobný MW, jenom pracují na nižší frekvenci (20-45 kHz). Pro člověka je toto vlnění nad slyšitelným pásmem, některá zvířata (psi, netopýři,...) jej ale slyší.

1.4 Vyráběné typy pohybových senzorů

Po zhlédnutí místního trhu s elektronikou a pohybovými senzory byly vybírány výhradně pasivní infračervené senzory, které jsou v dnešní době velice spolehlivé a oproti mikrovlnným typům jsou i levnější. Mezi vybranými jsou pouze dvojité nízkošumové PIR senzory, ve většině případů s velkou citlivostí.

Modul SB00612A-2

Jedná se o PIR modul bez krytu. I přes jednoduché zpracování jsou zde k dispozici regulační prvky pro nastavení citlivosti senzoru a zpoždění sepnutí. Pro různá napájecí napětí stačí na plošném spoji vyměnit rezistor a pak je rozmezí napětí 3,3-15 V. U tohoto modelu není možnost výběru, jestli bude senzor spínat alarm na první nebo druhý impuls. Aktivace na druhý impuls pomáhá s chybnou detekcí pohybu, která je většinou způsobena teplotními faktory a větrem. Všechny ostatní typy senzorů tuto možnost mají. Spotřeba tohoto modulu je velmi nízká, menší než 1mA. Čidlo je vidět na obrázku 1.2 a jeho parametry jsou uvedeny v tabulce 1.1.

Tabulka 1.1: Parametry modulu SB00612A-2

Rozměry	24 x 32 mm
Napájecí napětí	3,3-15 V
Úhel detekce	120°
Vzdálenost detekce	6 m
Cena (GME k 7.11.2014)	75 Kč

Senzor LH-941A

Jedná se o pasivní infračervené pohybové čidlo navržené pro stabilní detekci. Výrobek je již vybaven jednoduchým mikroprocesorem, který dokáže podle neustálého prověřování vzorků infračerveného záření zjistit, jestli se jedná o člověka nebo o falešný signál způsobený faktory prostředí. Oproti předchozímu modelu je zde znatelně větší

spotřeba, která se pohybuje okolo 20 mA. Čidlo je vidět na obrázku 1.8 a) a jeho parametry jsou uvedeny v tabulce 1.2.

Tabulka 1.2: Parametry senzoru LH-941A

Rozměry	106 x 54 x 36 mm
Napájecí napětí	9 – 16 V
Úhel detekce	90°
Vzdálenost detekce	12 m
Cena (GME k 7.11.2014)	249 Kč

Senzor LS-818-3

Tento senzor je popsán jako levnější ekvivalent předchozího typu LH-941A. Obsahuje termistor a algoritmus, který dokáže automaticky kompenzovat aktuální teplotu a předejít tím nevyžádaným falešným poplachům. Čidlo je vidět na obrázku 1.8 b) a jeho parametry jsou uvedeny v tabulce 1.3.

Tabulka 1.3: Parametry senzoru LS-818-3

Rozměry	107 x 59 x 39,5 mm
Napájecí napětí	9 – 16 V
Úhel detekce	120°
Vzdálenost detekce	12 m
Cena (GME k 7.11.2014)	173 Kč



Obrázek 1.7: a) Senzor LH-941A b) Senzor LS-818-3

Senzor LH-930A(II)

Uvedený typ senzoru neobsahuje Fresnelovu čočku, ale optické zrcadlo. Zrcadlo je tvarováno do paraboloidního tvaru, ale funkce je téměř stejná jako u čočky. Výrobce na svých stránkách uvádí, že je s ním zaručena velká šířka záběru paprsku, ale v dokumentaci je opět 90°, jako u většiny ostatních typů. Rozdíl nejspíš bude

v možnosti využití úzkého paprsku, který má detekční úhel pouze 15°, ale dokáže sřežit až do vzdálenosti 20 m. Jako u většiny senzorů, obsahujících MP, lze i u tohoto typu nastavit výstup alarmu na N.O (normally open) nebo N.C (normally closed). Čidlo je vidět na obrázku 1.9 a) a jeho parametry v tabulce 1.4.

Tabulka 1.4: Parametry senzoru LH-930A(II)

Rozměry	98 x 52 x 45,5 mm
Napájecí napětí	9 – 16 V
Úhel detekce	90°
Vzdálenost detekce	12 m (max. 20 m)
Cena (GME k 7.11.2014)	207 Kč

Senzor G1020

Tento typ vyniká především vysokým detekčním úhlem a možností nasměrování senzoru na detekovaný prostor. Přestože obsahuje klasický PIR senzor, je tento typ navržen především pro spínání světel, proto je zde také k dispozici regulační prvek pro nastavení citlivosti na denní světlo – od jaké intenzity osvětlení má začít spínat. Čidlo je vidět na obrázku 1.9 b) a jeho parametry jsou uvedeny v tabulce 1.5.

Tabulka 1.5: Parametry modulu G1020

Rozměry	110 x 80 x 65 mm
Napájecí napětí	220 - 240 V~
Úhel detekce	180°
Vzdálenost detekce	12 m (max. 20 m)
Cena (GME k 7.11.2014)	235 Kč



Obrázek 1.8: a) Senzor LH-930A(II) b) Senzor G1020

Vzhledem k dnešním možnostem a technologickému pokroku není výběr pohybového senzoru těžká záležitost. Většina pohybových čidel obsahuje kvalitní snímače, u kterých se uživatel nemusí bát častých falešných poplachů, jak tomu bylo dříve. Pokud je v přístroji zabudován mikroprocesor, který data ihned zpracovává, na alarmovém výstupu se objeví impuls až v okamžiku, kdy je prostor opravdu narušen. Čidla často obsahují možnost nastavení citlivosti, směrovosti paprsku nebo 1P/2P, která dává na výběr, jestli má být alarm aktivován při prvním detekovaném impulsu nebo až při druhém. Díky chytrým algoritmům jim nevádí ani změna teploty nebo osvětlení během dne.

S přihlédnutím k cenám a parametrům se jako nejlepší zařízení jeví čidlo s označením LS-818-3, které je nově v prodeji. K dispozici je informační LED dioda, která se při detekovaném pohybu rozsvítí, což usnadní další práci.

2 KAMERY

Kamera je snímací zařízení, které pořizuje snímky rychle za sebou s frekvencí vyšší než je schopno rozlišit lidské oko. Tato frekvence obvykle bývá 25 Hz a více, což znamená, že série snímků promítnutých za sebou se jeví jako plynulá scéna. Kamera bývá nejčastěji složena z objektivu, obrazového snímače a podpůrné elektroniky. U levnějších kamer se obvykle používá plastový objektiv s možností ručního ostření. U kamer starších typů vůbec nemusela možnost ostření být. Takovému objektivu se říkalo „statický“. V dnešní době jsou zcela obvyklé kamery s automatickým ostřením a dostatečnou hloubkou ostrosti, což zaručuje optimální kvalitu videa.

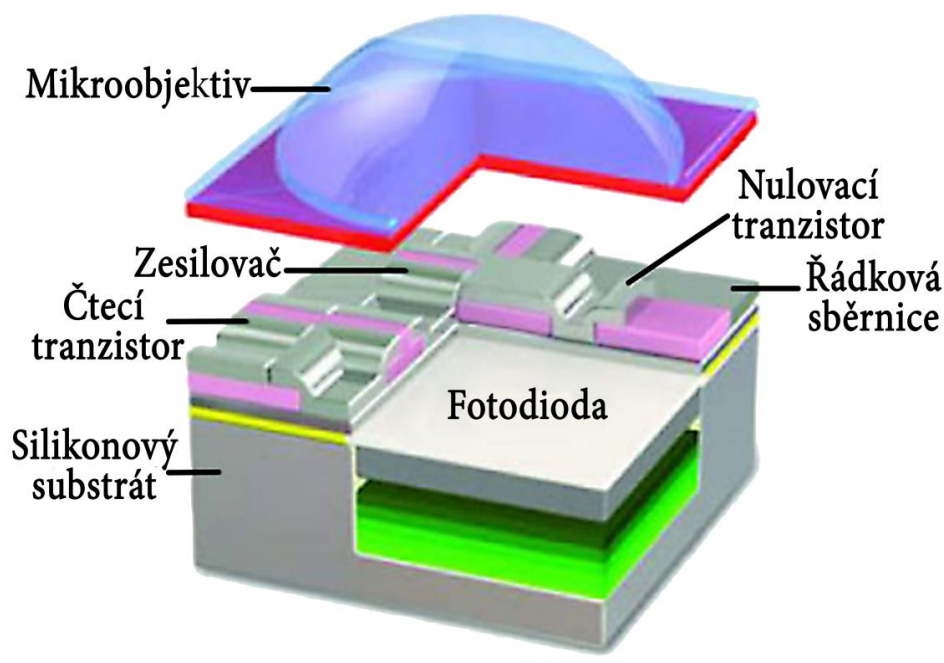
Internetové kamery, které budou v práci rozebírány, mohou být rozděleny od úplně základních vlastností, jako je možnost snímání barevného/černobílého obrazu, jeho velikost a rozlišení, bateriové/kabelové napájení, typ datového konektoru (USB, Firewire, LAN), automatické/manuální ostření, až k pokročilejším funkcím jako je typ snímače, komprese videa, IR-CUT filtr atd.

2.1 CMOS (Complementary Metal-Oxide-Semiconductor)

V dnešní době se kvalitou už přiblížily kvalitě CCD snímačů, ale stále nejsou vhodné pro kamery, od kterých je požadována nejvyšší možná kvalita obrazu. Špatná citlivost pro světlo stále představuje omezení pro využití CMOS snímačů. Tato nevýhoda není problém pro uživatele, kteří potřebují kameru pro dobře osvětlené prostředí. V tmavých prostorách je výsledkem velmi tmavý obraz plný šumu. Jeho výhodou je však rychlost a cena.

Ve standardních CMOS snímačích, tzv. konvenční nebo také Front CMOS, se využívá křemíkové fotocitlivé vrstvy, která je uložena až za kovovými obvody a tranzistory, což způsobuje, že je efektivně využita asi jen $\frac{1}{4}$ možné plochy [10]. Zbytek světla se od kovových součástek odráží zpět. Menší světlocitlivá plocha buňky vyžaduje vyšší zesílení, což zvyšuje obrazový šum. Výrobci CMOS senzorů se proti tomu brání mikroobjektivy (viz obrázek 2.1).

Pokročilejší technologií, která byla vytvořena v roce 2007, je typ CMOS se zpětným osvětlením, tzv. BSI (Back Side Illumination), který částečně řeší neduh konvenčních CMOS snímačů. Princip spočívá v přesunutí vrstvy s kovovými obvody a tranzistory na opačnou stranu křemíkového substrátu, tedy pod světlocitlivou vrstvu. Výhoda, která plyne ze zvýšeného záření na fotocitlivou vrstvu, je vyšší citlivost snímače. Bohužel tato technologie přináší i své nevýhody. Zpětně osvětlené struktury mohou způsobovat vyšší šum, tmavý proud nebo možnost míchání barev oddělených RGB maskou [11].



Obrázek 2.1: Struktura CMOS senzoru [12]

2.2 CCD (Changed Coupled Device) –

Jedním z vynálezců toho snímače je i laureát Nobelovy ceny Goerge E. Smith, který měl přednášku i na VUT v Brně.

Tento typ snímačů využívá unikátní metodu výroby, a to jinou než ostatní integrované obvody. Proto je poněkud problém dosáhnout jeho velkého rozlišení. Ze stejného důvodu je také problematické integrovat do senzoru jinou elektroniku, a proto je většina AD převodníků, zesilovačů a řídicích obvodů mimo sensor [12]. Digitálního obrazu se tak dosáhne až pomocí dalších integrovaných obvodů na desce s plošnými spoji a CCD sensor tak vyžaduje větší rozsah různého napájení, což komplikuje použití v reálném zařízení. Soubor integrovaných obvodů jako celek má potom i výrazně vyšší spotřebu ve srovnání s CMOS technologií.

Mezi výhody jednoznačně patří lepší světelná citlivost, která se projeví lepší kvalitou obrazu při špatném osvětlení. CCD snímače jsou ale dražší, protože se vyrábí nestandardním postupem a je složitější je zabudovat do kamery.

V menších amatérských kamerách a řadě dalších zařízení se používá snímání barevného obrazu jediným CCD, na jehož jednotlivých pixelech jsou naneseny barevné filtry.

Porovnání mezi senzory CCD a CMOS je vidět v tabulce 2.1.

Tabulka 2.1: Porovnání CCD a CMOS snímačů

	CMOS	CCD
Cena	Nízká	Vysoká
Rozměry	Malé	Vyšší
Spotřeba	Nízká	Vysoká
Kvalita obrazu	Nižší	Vysoká
Rozlišení	Střední	Vysoké
Digitální šum	Vysoký	Nízký
Rychlost	Vysoká	Nižší

Dalším možným rozdělením internetových kamer je na základě toho, jestli s počítačem komunikují pomocí sériové komunikace USB, takové kamery se obvykle nazývají webkamery, nebo využívají internetového protokolu, a pak se jmenují IP kamery. Pokud se jedná o síťové video nebo síťové kamery, zahrnuje přenos automaticky video i zvuk, které jsou dostupné napříč celým systémem. S pokrokem technologie se v dnešní době dají tyto kamery koupit velmi levně a i přesto mají velice dobré parametry.

Pokud je řeč o webkamerách, nejčastěji se jedná o malou kamerku připnutou na monitoru nebo vestavěnou v notebooku. Webkamery se používají zejména na videohovory pomocí komunikačních programů, jako je třeba Skype, ICQ atd. Naopak IP kamery se používají hlavně v odvětví vzdáleného monitoringu a zabezpečovacího dohledu.

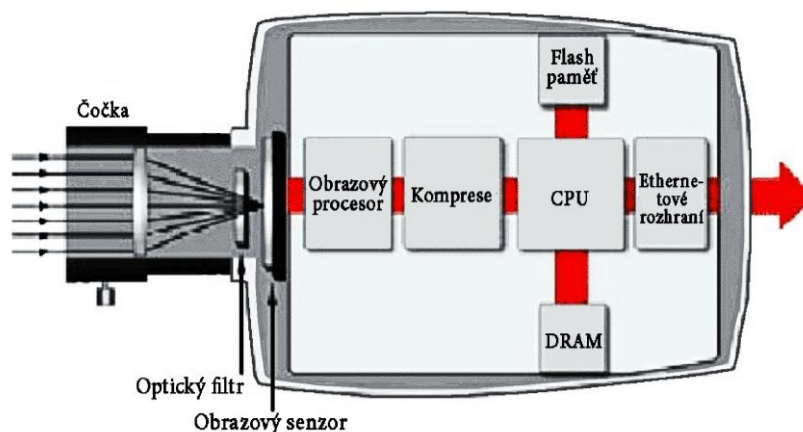
IP kamera může být popsána jako kamera a mikroprocesor v jednom zařízení. Snímá a vysílá záběry přímo přes IP síť a umožňuje tak autorizovaným uživatelům lokálně nebo na dálku sledovat, ukládat a spravovat video záběry pomocí standardní síťové infrastruktury založené na IP. IP kamera nemusí být nutně připojena ke stolnímu počítači, funguje nezávisle a můžeme ji umístit kamkoliv, kde je dostupné připojení k IP síti [13]. Běžná webkamera potřebuje připojení k počítači přes USB nebo Fireware a vyžaduje ke své činnosti zapnutý počítač.

Síťová kamera má svou vlastní IP adresu a vestavěné funkce, které zajišťují síťovou komunikaci. Vše potřebné pro sledování obrazu skrze síť je zabudováno v jednotce kamery. Ta je opatřena vestavěným softwarem pro web server, FTP server, FTP klienta a e-mailového klienta. Dále je opatřena jedním nebo více logickými vstupy (alarm input) a výstupy (relay output). O koordinaci a provádění veškerých činností jako inteligentní analýza obrazu, ovládání programovatelných I/O či komunikace se sítí a webovým serverem se stará řídicí procesor (CPU viz obrázek 2.2) společně s pamětí Flash a DRAM [18]. Hardwarové řešení je individuální pro jednotlivé výrobce.

Jak již název napovídá, tyto kamery pracují s IP, tedy s internetovým protokolem. Je to základní protokol používaný pro internet, email a téměř každou nově nainstalovanou síť. Jedním z důvodů jeho popularity je jeho škálovatelnost. Každé malé i velké zařízení může využívat stejné funkce. Ethernet se za roky své existence postupně vyvíjel a tím se zvyšovaly i přenosové rychlosti datové sítě. Pro připojení jednotlivých IP kamer se využívá zpravidla standardu 100BASE-T, který umožňuje datový přenos o rychlosti 100 Mb/s prostřednictvím kroucené dvoulinky. Pro připojení více kamer se

využívá síťových přepínačů (Switchů), které zároveň zajišťují zpětnou kompatibilitu se starším standardem Ethernetu 10BASE-T, a označují se jako 10/100 přepínače.

Internetový protokol vysílá datagramy na základě síťových IP adres obsažených v jejich záhlaví. Každý datagram je samostatná síťová jednotka, která obsahuje všechny potřebné údaje o adresátovi i odesílateli a pořadovém čísle datagramu ve zprávě. Datagramy putují sítí nezávisle na sobě a pořadí jejich doručení nemusí odpovídat pořadí ve zprávě. Tento protokol se dále stará o segmentaci a znovusestavení datagramů do rámců a z rámců podle protokolu nižší vrstvy (např. ethernet). V současné době je převážně používán protokol IP verze 4. Nahrazuje ho však protokol IPv6, který řeší především nedostatky adres IPv4, ale vylepšuje i další vlastnosti protokolu.



Obrázek 2.2: Blokové schéma zapojení IP kamery

Kromě zaznamenávání obrazu může IP kamera obsahovat i jiné užitečné funkce a dokáže po síti přenášet i jiná než obrazová data. Kamery mohou být vybaveny aktivními nebo pasivními detektory pohybu, přenosem zvuku, digitálními vstupy a výstupy, mechanismy pro otáčení a směřování kamery atd.

Kvalita obrazu je jednou z nejdůležitějších vlastností kamery. Platí to především pro zabezpečovací kamery, kde je důležitý kvalitní obraz pořízený při narušení objektu. Speciální čipy pro zpracování obrazu a sofistikované algoritmy pro síťové video nabízí uživatelům také různé standardní komprese videa (Motion JPEG, H.264, MPEG-4, MPEG-2). Kvalita obrazu však nezáleží jen na typu komprese videa. Naopak, nejkvalitnější záznam je ten, který komprimován není. Komprese je však nezbytná pro snížení přenosové rychlosti obrazových i zvukových signálů.

Kvalita videa se nejčastěji spojuje s rozlišením snímače. Je pravda, že čím větší je rozlišení obrazu, tím jsou patrnější detaily, ale je potřeba brát v úvahu i další parametry, které s kvalitou souvisejí. Je to například typ snímače. Jak je napsáno výše, nelze říci, jestli je lepší snímač CMOS nebo CCD. Každý má jiné přednosti a nedostatky. Dalším parametrem je schopnost kamery pracovat při špatném světle. Levnější kamery jsou vybaveny automatickým prosvětlením obrazu, což jen zesvětlí jas celého obrazu, ale snímek je pořád tmavý a nekvalitní. U všestrannějších kamer může být okolo objektivu několik IR diod, jejichž svit není okem vidět, kamerou je však považován za viditelné světlo a tím pádem je obraz prosvětlený do vzdálenosti, kam diody dosvítí. Další vlastnost kamer, která může zvýšit kvalitu obrazu, je schopnost výměny objektivů.

Kvalitnější objektivy mohou poskytovat lepší záběry. Někteří výrobci objektivy na své kamery nižších cenových tříd dodávají, ale odpovídají kvalitou své ceně. Ale u kamer vyšších tříd bývá zvykem objektivy ke kameře vůbec nedodávat a je na zákazníkovi, aby si vybral ten pravý podle svých nároků. Obvyklé parametry objektivů jsou: velikost objektivu, ohnisková vzdálenost a clona. Dále může být objektiv vybaven o pokročilé funkce, jako je IR korekce nebo asférický objektiv.

Kvalitu může zvýšit i pokročilá funkce zvaná „Kompenzace světla na pozadí“. Ta se snaží ignorovat malé oblasti ostrého světla, jako by tam vůbec nebyly. V kameře bez kompenzace by se automatická kontrola expozice snažila nastavit úroveň světlosti tak, jak by ji vidělo lidské oko. Může být však oklamána třeba právě ostrým světlem v temném pozadí.

Je spousta dalších faktorů, podle kterých se kamery mohou dělit, např. vestavěný mikrofon, který je velmi často vyžadován a i v rámci diplomové práce bude využíván.

2.3 Rozlišení a komprese

Digitální rozlišení je ve světě analogovém i digitálním podobné. Rozdíly jsou v tom, jak je definováno. V případě analogového videa se obraz skládá z řádků, zatímco rozlišení digitální je měřeno počtem efektivních bodů (pixelů) snímáče. Po světě jsou rozšířeny dva standardy pro analogové video. V Evropě se využívá standard PAL (Phase Alternation by Line), který má rozlišení 576 horizontálních řádků, které se zobrazují každých 25 sekund. V Americe a Japonsku je využíván standard NTSC (National Television System Committee), který má rozlišení 480 horizontálních řádků a frekvenci 30 snímků za sekundu.

Diplomová práce se však bude zabývat síťovými kamerami, které jsou v dnešní době již plně digitální. Na trhu je spousta druhů kamer s různým rozlišením. Jsou k dispozici přístroje s nejnižšími rozlišeními (QVGA, VGA), ale za velmi přijatelnou cenu jsou dostupné i přístroje s rozlišením např. 720 HD. Rozměry obrazu spolu s názvem některých typů rozlišení jsou uvedeny v tabulce 2.2.

Tabulka 2.2: Rozlišení obrazu

Rozlišení	Název
320x240	QVGA
640x480	VGA
1024x768	XGA
1280x720	HD 720
1280x1024	SXGA
1600x1200	UXGA
1920x1080	HD 1080
2560x2048	QSXGA

S rozlišením souvisí i poměr stran, který obvykle bývá 16:9 nebo 4:3. U síťových kamer je obvykle možnost tyto parametry volit a v případě potřeby snížit rozlišení na některé z nižších podporovaných. Toto rozhodnutí může způsobit velký datový tok,

který je u vyšších rozlišení obvyklý. Nekomprimované video může mít datový tok až stovky Mb/s, po kompresi klesne tok na jednotky až desítky Mb/s. Velmi důležitým parametrem u kamer a obecně u videa je typ kódování, kam patří i audio a video komprese.

Bez efektivní komprese by většina lokálních sítí LAN přenášejících video data zkolabovala. Digitální video se vždy komprimuje, aby se snížila přenosová rychlost. Proto je výběr správného formátu pro kompresi velice důležitý.

Komprese obrazu a videa může být rozdělena na ztrátovou a bezztrátovou. V případě bezztrátové komprese je každý pixel ponechán nezměněn, takže výsledkem po dekompresi je identický obraz. Nevýhodou tohoto přístupu je, že kompresní poměr, tedy snížení dat, je velmi omezený. Dobře známým bezztrátovým formátem je GIF [13]. Kvůli omezeným kompresním možnostem, se tyto formáty nehodí pro použití v oblasti síťového videa, kde potřebujeme ukládat a posílat velké množství obrázků (záběrů). Proto bylo vyvinuto několik standardů pro ztrátovou kompresi. Základní myšlenkou je redukovat části obrazu neviditelné lidskému oku, a tak výrazně zvýšit kompresní poměr.

Na úvod je zde uveden příklad výpočtu datového toku nekomprimovaného videa s rozlišením 720 HD. Objem dat barevného videa 1280x720 délky 1s s 25 snímků/s se vypočítá $1280 \times 720 \times 25 \times 3 = 69\,120\,000$ bajtů, tedy 69,12 MB ~ 552,93 Mb. Tato hodnota je pro komunikaci po LAN síti nereálná. Pro srovnání jsou v tabulce 2.3 přenosové rychlosti videa s různým rozlišením, pro různé kompresní metody.

Tabulka 2.3: Porovnání kompresních metod

Rozlišení	MJPEG	MPEG-4	H.264	Bez komprese
640x480 (VGA)	7,78 Mb/s	1,58 Mb/s	0,57 Mb/s	184,3 Mb/s
1280x1024 (1,3 MPix)	32,77 Mb/s	6,76 Mb/s	2,46 Mb/s	786,4 Mb/s
1920x1080 (Full HD)	51,81 Mb/s	10,65 Mb/s	3,89 Mb/s	1244 Mb/s

Kamera před odesláním dat provádí určité operace, které redukuje množství dat, jež je potřeba přenést. Tento postup se nazývá kompresní (komprimační, stlačovací). Na přijímací straně je potřeba video dekomprimovat, tedy převést zpět na původní data.

Úkolem zdrojového kódování je redukce redundance a irelevance na nejnižší míru. Redukce redundance (nadbytečnosti) je proces bezztrátový, redukce resp. odstranění irelevance (zbytečnosti) je proces ztrátový. Poměr přenosové rychlosti signálu před a po komprimaci se nazývá komprimační poměr. S vysokým komprimačním poměrem bývá obvykle spojena i horší kvalita obrazu. Tento pokles kvality se projevuje poklesem počtu detailů v obraze, pro určité kompresní metody pak i „rozkostičkováním“ obrazu.

V současnosti používají zabezpečovací kamery dvou typů kompresí obrazu – první typ komprimuje každý jednotlivý snímek, druhý typ si všímá rozdílů mezi jednotlivými snímky. Zastaralé standardy MPEG-1 a MPEG-2 už se v síťových kamerách téměř vůbec nepoužívají. U většiny kompresí se využívá redukce barevného rozlišení ve vertikálním směru – 4:2:0. Znamená to, že chrominanční složky C_B a C_R jsou vzorkovány na poloviční datový dok ve srovnání s luminanční složkou Y .

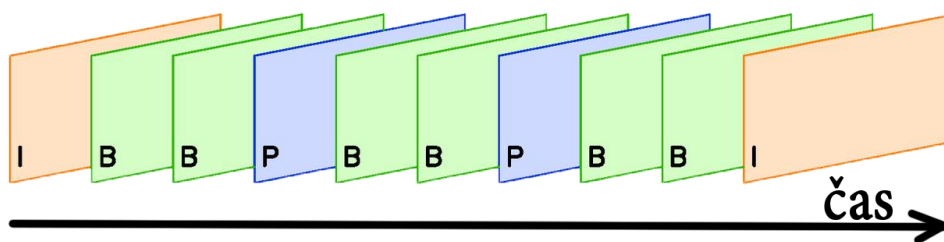
MJPEG komprese – jedná se o typ komprese, která komprimuje každý snímek zvlášť.

Je to tedy video ve formě sekvence JPEG obrázků. Motion JPEG se nejčastěji používá v oblasti síťového videa, kdy podobně jako digitální fotoaparát zachytí jednotlivé snímky a zkomprimuje je do formátu JPEG. Při využití této komprese se obvykle nesnímá větší frekvencí než 30 snímků za vteřinu. Jelikož každý snímek tvoří samostatný JPEG záběr, je tím zaručena kvalita určená úrovní komprese jednotlivých snímků [14]. Nevýhodou je větší velikost výsledného videa než u jiných novodobých formátů.

MPEG-4, H.264 – základním principem tohoto druhu komprese je, že se nejprve přenáší tzv. referenční snímek. Ten může být komprimován podobně jako u předchozího typu komprese. V dalším snímku se již přenáší pouze rozdíl oproti referenčnímu snímku, čímž se výrazně redukuje množství dat k přenosu. Problém při přenosu může nastat, pokud se objeví velké množství po sobě jdoucích chyb [16]. Dekódovaný signál v té chvíli neodpovídá originálu a je třeba jej opravit. MPEG-4 používá k dosažení nejvyšší komprese predikční algoritmy. Pravidelně odesílané videosnímky, tzv. I-snímky, jsou přenášeny kompletní, samostatné obrázky ve formátu JPEG a slouží jako referenční bod pro prediktivní algoritmy. Video snímky P obsahují pouze informace, které byly změněny od předchozího snímku. MPEG-4 komprese využívá tyto typy snímků:

- I-snímky (Intra Frames) – Snímek nezávisle kódovaný. Tyto snímky jsou také označovány jako klíčové snímky a obsahují všechny údaje, které jsou potřebné k zobrazení obrazu v podobě jednoho snímku.
- P-snímky (Predicted Frames) – Tento typ snímku obsahuje pouze obrazová data, která se změnila z předchozího snímku I nebo P.
- B-snímky (bidirectional Frames) – Tomuto typu se také říká obousměrný. Může odkazovat na snímky I nebo P předchozí i následující [15].

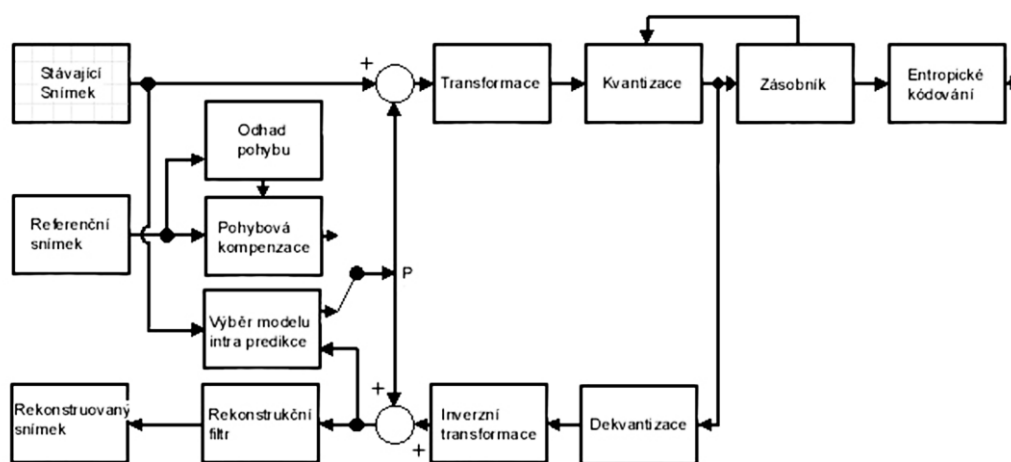
V určitém časovém intervalu je do řady snímků, tzv. GOP (Group Of Pictures – viz obrázek 2.3), vložen referenční snímek I (intracoded) vytvořený bez rozdílů mezi snímky a mezi snímky I jsou vloženy rozdílové snímky P (predicted) a B (bidirectionally predicted). Pokud v obraze nedochází k pohybu, snímky s rozdílovým obrázkem nemusí být vůbec přenášeny.



Obrázek 2.3: Sekvence snímků

Ještě předtím, než je snímek analyzován na statické a dynamické části, je rozdělen na konečné množství bloků 16 x 16 luminančních pixelů a 8 x 8 obou chrominančních pixelů (platí pro starší typy MPEG). Při vzorkování 4:2:0 je 8 x 8 C_B pixelů a 8 x 8 C_R pixelů je vždy překryto další hladinou 16 x 16 Y luminančních pixelů a toto uskupení nazýváme makroblok.

Novější verzí je standard H.264, označován také jako MPEG-4 part 10, ale i pod názvem AVC (Advanced Video Coding). Na obrázku 2.4 je zobrazena vrstva video kódování. Tato vrstva je téměř stejná jako u starších standardů tohoto typu, ovšem jednotlivé funkční bloky kodéru byly vylepšeny, a tím se dosáhlo úspory bitového toku. Mnoho částečných vylepšení dá v součtu výrazně lepší kvalitu při stejném bitovém toku oproti starším standardům, jako je např. MPEG-2. Jednotlivá vylepšení konkrétních funkčních bloků jsou například proměnná velikost bloků pro pohybovou kompenzaci (tzv. stromově strukturovaná pohybová kompenzace), použití celočíselné transformace odvozené z diskretní Kosinovy transformace (DCT) o velikosti 4 x 4 bodů, zavedení rekonstrukčního filtru potlačující blokovou strukturu, vylepšené entropické kódování s proměnnou délkou slova a použití binárního aritmetického kódování.



Obrázek 2.4: Kodér standardu H.264 [17]

Výhodou tohoto typu je vyšší kompresní poměr – pro starší kompresi MPEG-4 je to cca 1:120, novější kompresní metodou H.264 pak cca 1:320. Oproti MJPEG jsou potřeba vyšší nároky na výkonnost hardwaru (výkon procesoru a větší spotřeba paměti) a větší latence v přenosu.

Nástupcem standardu H.264 by měl být kodek H.265 nebo také HEVC (High Efficiency Video Coding). Ten by se měl používat pro videa v rozlišení 4K (4096 x 3112).

2.4 Vyráběné typy kamer

Při vybírání kamer bylo k dispozici daleko větší množství produktů, které bylo možno porovnávat. Produkty jsou vybírány výhradně z českých obchodů a e-shopů, aby se předešlo případným problémům s dopravou zboží do České republiky. V dnešní době již není problém koupit kvalitní kameru určenou přímo na zabezpečení objektu a nahrávání na paměťovou kartu. Mezi vybranými typy jsou většinou IP kamery s ethernetovým rozhraním LAN, které jsou ovládány CGI (Common Gateway Interface) příkazy. Během porovnávání bylo přihlíženo zejména na rozlišení videa a druh kódování, dále na možnost využití integrovaného mikrofónu a v neposlední řadě na cenu kamery.

Creative Live! Cam Sync HD

Nejlevnější z kamer od společnosti Creative je webkamera s možností připnutí na monitor od počítače nebo postavení na stůl. Oproti obyčejným webkamerám má výhodu ve vysokém rozlišení 1280x720 bodů při frekvenci 30 snímků za sekundu a je vybavena integrovaným mikrofonom s funkcí redukce okolního šumu.

Zákazníci, kteří si tuto kameru koupili, však s kvalitou obrazu mají problém s tím, že neodpovídá HD rozlišení. Podle zákazníků má snímač maximální rozlišení 640x480 a do HD rozlišení jsou body dopočítávány softwarově. I přes tyto problémy je to kamera oblíbená, a to především pro internetové volání a aplikace nenáročné na kvalitu obrazu. Kamera je vidět na obrázku 2.5 a její parametry jsou uvedeny v tabulce 2.4.

Tabulka 2.4: Parametry kamery Creative

Rozlišení	1280x720 (při 30 fps)
Cena (alza.cz k 8.11.2014)	399 Kč
Mikrofon	Ano



Obrázek 2.5: Webkamera Creative [28]

AirLive AirCam IP-200PHD-24

IP kamera navržena především pro sledování majetku. Je vybavena ethernetovým rozhraním LAN s možností pasivního PoE (Power over Ethernet) napájení a tím usnadní zapojování i mimo dosah elektrické přípojky. Kamera dokáže snímat obraz v rozlišení až 1600x1200 bodů při snímkové frekvenci 15 snímků/s. Při použití menšího rozlišení 1280x720 bodů pracuje již s 30 snímků za sekundu. Pro přenos video signálu je signál kódován nejmodernějšími kodeky H.264, což zaručuje nízkou přenosovou rychlost i při vysokých rozlišeních. Pomocí kamerového softwaru lze detekovat pohyb a následně spustit nahrávání videa na paměťovou kartu nebo odeslat upozornění mailem. Také zvuk je kódován tak, aby byla přenosová rychlost při vysoké kvalitě zvuku co nejmenší. Kamera je vidět na obrázku 2.6 a) a její parametry jsou uvedeny v tabulce 2.5.

Tabulka 2.5: Parametry kamery AirCam IP-200PHD-24

Rozlišení	1600x1200 (15 fps) 1280x720 (30 fps)
Snímač	1/3" CMOS
Formát obrazu	H.264, MPEG4, MJPEG
Cena (alza.cz k 8.11.2014)	2899 Kč
Mikrofon	Ano
Formát zvuku	RTSP: G.711, G.726 3GPP: AMR
Rozměry	74,5 x 52,6 x 35 mm



Obrázek 2.6: a) AirCam IP-200PHD-24 b) AirCam POE-5010HD [29]

AirLive AirCam POE-5010HD

Tato 5 megapixelová IP kamera je navržena pro snímání videa ve velkém rozlišení přes lokální síť. Díky vysokému rozlišení poskytuje kvalitu až forenzních kvalit a je vhodné ji použít v místech, kde toto rozlišení může být využito. Například pro policejní nebo železniční potřeby, kde jsou nutné detailní informace v obraze. Kamera také obsahuje odnímatelný „IR-CUT“ filtr. Ten při denním světle nepropouští nežádoucí IR záření, a tím zajišťuje věrohodnější podání barev v obraze [8]. Naopak v noci je tento filtr vypnutý a snímač zachytí více světla. Spolu s přídatnou infračervenou diodou je tedy možné spolehlivě využívat obraz i v noci. Tomu nahrává i možnost výměny objektivů. Do kamery je možné vložit paměťovou kartu a v případě potřeby ji využít pro záznam. Kamera je vidět na obrázku 2.6 b) a její parametry jsou uvedeny v tabulce 2.6.

Tabulka 2.6: Parametry kamery AirCam POE-5010HD

Rozlišení	2592x1920 (8 fps) 2048x1536 (15 fps) 1920x1080 (20 fps) 1280x960 (30 fps)
Snímač	1/2.5" CMOS
Formát obrazu	H.264, MPEG-4, MJPEG
Cena (alza.cz k 8.11.2014)	7865 Kč
Mikrofon	Ano
Formát zvuku	RTSP: G.711 3GPP: AMR
Rozměry	134 x 79 x 59 mm

Foscam FI9821W

Kamera od čínské značky FOSCAM na první pohled zaujme svým vzhledem, který naznačuje, že by mohlo jít o kameru ovladatelnou. Kamera nabízí HD rozlišení s H.264 video kompresí, což zaručuje nízkou přenosovou rychlost i přes vysoké rozlišení. Typ FI9821W je vybaven slotem pro paměťovou kartu, mikrofonem, reproduktorem a dokonce i Wi-Fi se standardem IEEE 802.11 b/g/n, což umožňuje další využití. Další výhodou této kamery je možnost vzdáleně ji natáčet a ovládat. To může usnadnit složité lezení po výškách a směřování kamery na určitý prostor, který má být sledován. Možnost natáčení kamery je 300° horizontálně a 120° vertikálně [9]. Přes Wi-Fi nebo LAN síť se lze pomocí IP adresy kamery připojit na její uživatelské rozhraní a kameru ovládat nebo nastavit možnost nahrávání při detekci pohybu ve střeženém místě. Detektor pohybu je opět softwarový a v obraze je možnost výběru zón střežených a nestřežených. Například pokud je v obraze okno, za kterým procházejí lidé, lze oblast okna nestřežit a tím zamezit falešným alarmům. Kamera je také vybavena 11 IR LED diodami, které mají noční dosvit až 8 m, což umožňuje kvalitní videozáběry z nočního prostředí. Kamera je vidět na obrázku 2.7 a) a její parametry jsou uvedeny v tabulce 2.7.

Tabulka 2.7: Parametry kamery Foscam FI9821W

Rozlišení	1280x720 (30/25 fps)
Snímač	CMOS s vysokým rozlišením
Formát obrazu	H.264
Cena (kameryip.cz k 9.11.2014)	2790 Kč
Mikrofon	Ano
Formát zvuku	RTSP: G.711/G.726 3GPP: AMR
Rozměry	117 x 114 x 129 mm



Obrázek 2.7: a) Foscam FI9821W b) Foscam FI9826W [30]

Foscam FI9826W

Tento typ je velmi podobný předchozímu typu FI9821W s tím rozdílem, že má o málo větší rozlišení, k dispozici je také 3x optický zoom, který umožňuje ještě detailnější záběr hlídaného objektu. Oproti předchozímu typu má k dispozici 13 IR LED diod. U některých typů kamer FOSCAM, včetně tohoto, je zdarma udělen přístup na DDNS server, který umožní uživateli s dynamickou IP adresou sledovat obraz odkudkoliv ve světě, kde se připojí k internetu. Kamera je vidět na obrázku 2.7 b) a její parametry jsou uvedeny v tabulce 2.8.

Tabulka 2.8: Parametry kamery Foscam FI9826W

Rozlišení	1280x960 (30/25 fps)
Snímač	CMOS s vysokým rozlišením
Formát obrazu	H.264
Cena (kameryip.cz k 9.11.2014)	4190 Kč
Mikrofon	Ano
Formát zvuku	RTSP: G.711/G.726 3GPP: AMR
Rozměry	117 x 103 x 119 mm

Při výběru kamery sehrála roli spousta parametrů. Nejdůležitějšími faktory byla cena, rozlišení, velikost a v neposlední řadě i dostupnost. Na trhu je mnoho kamer vyhovujících požadavkům diplomové práce od různých výrobců. Na první pohled by se zdála nejlepší kamera AirLive POE-5010, která má rozlišení až 2592 x 1944 dpi a ani její cena není vysoká. Tato kamera je však navržena pro zachycení nejvyšších detailů i na velké vzdálenosti. V případě diplomové práce při využívání PIR senzoru, který má dosah asi 10 metrů, není tak velké rozlišení potřeba a na kvalitní video na tuto vzdálenost stačí rozlišení 1280 x 720 dpi. I na větší vzdálenosti bude vidět obličej osoby, která prostor narušila.

Další kamera od firmy AirLive je typ IP-200PHD-24. Tento typ může pořídít video

s rozlišením 1600 x 1200 dpi, které je snímáno frekvencí 15 snímků/s. Vzhledem k její ceně, která se pohybuje pod hranicí 3000 Kč, je výhodnější kamera FOSCAM FI9821W, která za stejnou cenu nabídne spoustu dalších užitečných funkcí, jako je přisvícení IR LED diodami, připojení k Wi-Fi a možnost kameru vzdáleně otáčet, což se může hodit v momentě, kdy je již celý bezpečnostní systém nainstalovaný a obraz z kamery je potřeba seřadit přesně na střeženou oblast. Novější typ FI9826W nabízí ještě optický zoom, ale ten by nejspíš zůstal nevyužitý. I přes funkce kamery FI9821W, které možná zůstanou nevyužity, se tento typ poměrem cena/výkon jeví jako nejlepší volba. Problémem by však mohla být jeho dostupnost.

Ke kamerám FOSCAM je k dispozici CGI (Common Gateway Interface), přes které se kamera ovládá prostřednictvím ethernetového protokolu. Do kamery tedy stačí poslat například příkaz:

```
/cgi-bin/CGIProxy.fcgi?cmd=getVideoStreamParam&usr=admin&pwd=
```

a odpovědí jsou informace o rozlišení, bitové rychlosti, frekvenci obrázků a počet P snímků mezi snímky I. Na konci příkazu jsou zkratky *usr*, kde se zadá uživatelské jméno a zkratka *pwd* sloužící k zadání hesla pro přístup ke kameře. Podobná je funkce *setVideoStreamParam*, kde se parametry zobrazené výše dají nastavit.

Dále bylo nutné zajistit, aby vlnová délka IR diod neovlivňovala funkci PIR senzoru, který je také citlivý na IR záření. Jelikož vlnová délka IR diod je okolo 940 nm, tak by funkce pohybového senzoru ovlivněna být neměla. Čidlo by mělo reagovat pouze na IR záření lidského těla a na vlnové délky v blízkém okolí. Toto rozmezí je od 8-14 μm , což zaručí funkčnost obou zařízení současně.

Díky novým kodekům MPEG-4 Part 10 neboli H.264 je video komprimováno na velmi malé přenosové rychlosti, což zaručí „neobjemné“ video i při vysokém rozlišení při snímání 30 snímků/s.

3 PŘENOS DAT

3.1 Modem Huawei E398

USB modem od firmy Huawei (viz obrázek 3.1) navazuje na úspěšnou sérii E s typovým označením 398. Tento model již podporuje nově se rozšiřující mobilní komunikaci systémem LTE, který umožňuje dosáhnout přenosové rychlosti v downlinku až 100 Mb/s a v uplinku až 50 Mb/s.

Po připojení k počítači se automaticky spustí nabídka pro instalaci potřebného softwaru a ovladačů potřebných pro komunikaci.

Dual Carrier technologie pracuje podobně jako dvoupásmový router a umožňuje odesílat a přijímat bezdrátově přenášená data přes dva různé kanály současně na síti UMTS HSPA+ [19]. Je vybavena funkcí 2 x 2 MIMO (Multiple Input Multiple Output) a samozřejmě podporuje i starší technologie, jako GPRS apod.

K modemu je možné připojit dvě externí antény, které mohou zlepšit signál, ale modem pracuje velmi dobře i bez nich. K dispozici je také slot na SD kartu. Lze jej využít nejen pro přenášení dat do mobilního telefonu, ale může pracovat i jako USB flash disk.



Obrázek 3.1: Modem Huawei E398

3.2 Systém UMTS

Systém UMTS (Universal Mobile Telecommunications System) je mezinárodní standard využívající frekvenční spektrum od 1885 MHz do 2025 MHz nebo od 2110 MHz do 2200 MHz. Používá technologii WCDM (Wideband CDMA). Jeden rádiový kanál má přesně definovanou šířku 5 MHz.

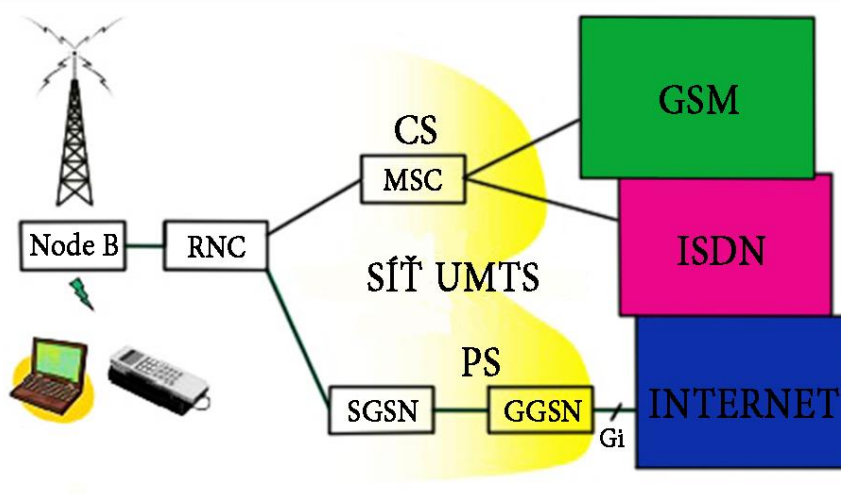
Systém UMTS umožňuje dva druhy duplexního provozu. Prvním, významně rozšířeným typem je FDD (Frequency Division Duplex), kde jsou pro uplink a downlink použity různé rádiové kanály. Druhý je TDD (Time Division Duplex), kde

pro uplink a downlink se používá jeden rádiový kanál, ve kterém se oba směry přenosu střídají v čase.

Hlavní odlišnosti mezi systémy UMTS a GSM vyplývají z rozdílných přístupových technologií použitých na rádiovém rozhraní. Přístup CDMA aplikovaný u systémů UMTS opakovaně využívá stejný kmitočet nosné v každé buňce. Z toho důvodu bylo třeba uvažovat příjem signálů s využitím makrodiverzity, využití přijímačů typu RAKE nebo používání několika různých procedur při handoveru. Systém UMTS používá také novou metodu kódování řečových signálů umožňující adaptivně měnit výstupní přenosovou rychlost signálu [20]. Dále používá kombinovaný přístup FDMA / CDMA. Kmitočtové pásmo přidělené systému je rozděleno na jednotlivé rádiové kanály (FDMA) a v každém rádiovém kanálu jsou jednotlivé uživatelské kanály rozlišeny jedinečnou kódovou sekvencí (rozprostíracím kódem), kterou se kóduje přenášená informace (CDMA). Všichni uživatelé systému tedy používají stejnou nosnou a komunikují ve stejném čase, přičemž pro vzájemné rozlišení používá každý svoji vlastní kódovou sekvenci. V přijímači je signál dekodován na základě znalosti příslušné kódové sekvence, a tím je obnovena přenášená informace.

Starší systém GSM (2G) byl navrhován především pro přepínání okruhů (CS – Circuit Switched), tedy hlavně pro telefonování. Ale jak je vidět na obrázku 3.2, novější systémy, počínaje standardem GPRS (2,5G) a dále UMTS (3G), už jsou přizpůsobeny i datovým přenosům, tedy přepínání paketů (PS - Packet Switched). Přepínání paketů využívá internetový protokol IP, což znamená, že virtuální spojení lze navázat s jakýmkoliv koncovým bodem v síti.

Přepínání paketů je technologie používaná především v počítačových sítích, kdy jsou data posílána postupně po menších částech (pakety, ethernetové rámce). Každá část v sobě nese informaci o cíli své cesty a je počítačovou sítí doručována samostatně. Směrování paketů je zajištěno pomocí přepínačů (switch, router). Původní zprávu sestavuje do původní podoby až příjemce, protože jednotlivé části mohou sítí procházet různými cestami. Převahu paketů zajišťuje na síťové vrstvě IP protokol. Sestavování původní zprávy je zajištěno protokolem TCP na transportní síťové vrstvě.



Obrázek 3.2: Architektura UMTS

3.3 Systém LTE

Zkratka z anglického 3GPP Long Term Evolution. Technologie určená pro vysokorychlostní internet, někdy také nesprávně označovaná 4G. Správné označení by však mělo být 3,9G. Nástupce LTE, systém LTE – Advanced, by už měl splňovat rychlostní požadavky a je označován jako 4G.

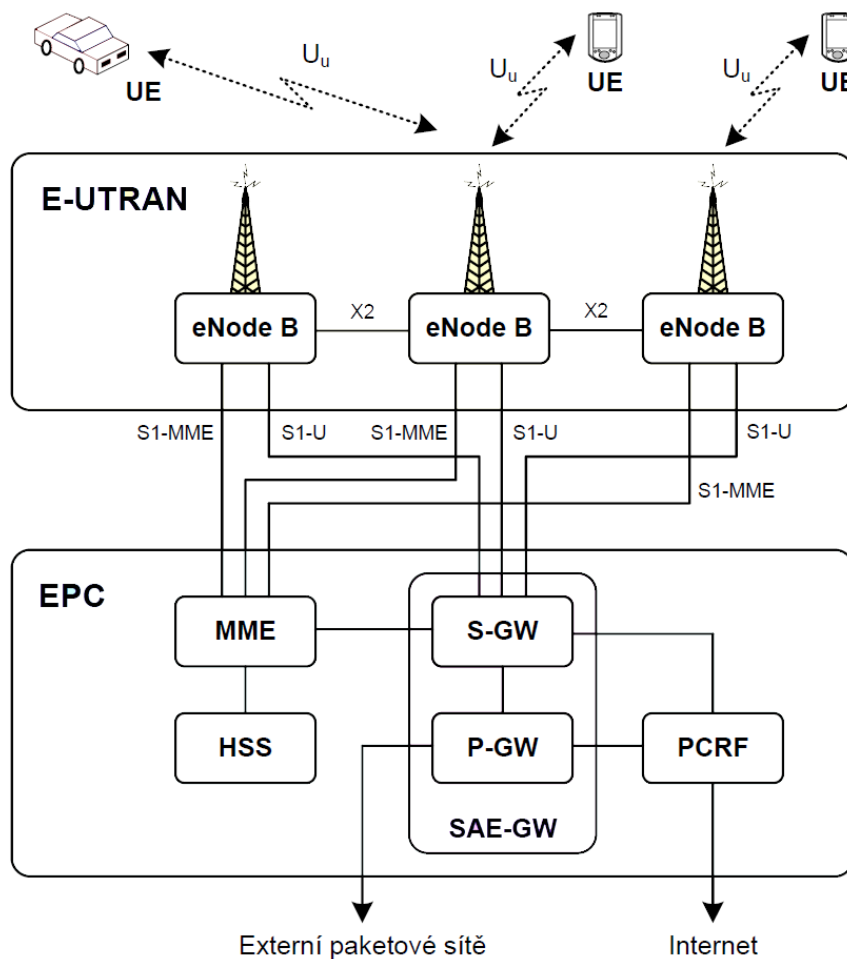
Systém využívá výhradně paketový přenos založený na protokolu IP, přepínání okruhů již zde využito není. Rozšířilo se QoS (Quality of Service), začala se používat modulace OFDM (Orthogonal Frequency Division Multiplexing) a přibyla i možnost použití více antén (technologie MIMO).

Systém LTE dosahuje maximální přenosovou rychlost až 300 Mb/s, dále vyniká velmi malým zpožděním (až 5 ms) a výrazným zvýšením spektrální účinnosti. Podporuje jak FDD, tak i TDD [21].

Jak je vidět na obrázku 3.3, architektura sítě se skládá z dvou základních částí, z páteřní sítě EPC (Evolved Packet Core) a přístupové sítě E-UTRAN (Evolved Universal Terrestrial Radio Access Network).

První blok E-UTRAN se skládá ze základnových stanic eNodeB, které kombinují funkci NodeB a RNC ze systému UMTS, čímž snižují latenci a řídí komunikaci jedné nebo několika buněk. Mobilní stanice (např. telefon) může komunikovat vždy pouze s jedním eNodeB, ale eNodeB může obsluhovat více uživatelských zařízení UE (User Equipment). Základnové stanice mezi sebou komunikují pomocí rozhraní X2, s EPC pomocí rozhraní S1. Druhý blok EPC neobsahuje správu pro komutovaný přenos, jsou zde však nové bloky:

- MME (Mobility Management Entity) - řídí vysokoúrovňové procedury UE. Stará se o zabezpečení a řízení datových toků nesouvisejících se samotnou rádiovou komunikací. Může obsluhovat několik eNodeB, ale samotná data přes něj nevedou
- HSS (Home Subscriber Server) – centrální databáze všech uživatelů v síti. Nese informace o všech účastnících včetně služeb, kam mají přístup.
- S-GW (Serving Gateway) – funguje jako router, směřuje data mezi základnovou stanicí a P-GW [22].
- P-GW (Packet Gateway) – je připojením EPC s jinými paketovými sítěmi. Každá PDN (Packet Data Network) je identifikovaná jménem přístupového bodu. Mimo jiné přiděluje uživatelským terminálům IP adresy.
- PCRF (Policy and Charging Rules Function) – sleduje poskytované služby, kontroluje kvalitu QoS a dohlíží na vyúčtování využívaných služeb.



Obrázek 3.3: Architektura LTE [20]

SIM karta (Subscriber Identity Module) je účastnická identifikační karta, která slouží pro identifikaci účastníka v mobilní síti. Je na ní uloženo číslo IMSI (International Mobile Subscriber Identity), které jednoznačně identifikuje účastníka na celém světě. Každé kartě je také přiřazeno číslo MSISDN (Mobile Station International Subscriber Directory Number), což je z uživatelského pohledu telefonní číslo. Karta může být trojího typu. USIM (UMTS SIM), ISIM (IP Multimedia Subsystem SIM) nebo kombinací předešlých dvou, tedy UICC (Universal Integrated Circuit card).

LTE v České republice využívá tři frekvenčních pásem: 800 MHz, 1800 MHz a 2600 MHz. Příklad rozložení kmitočtů 20. pásma (LTE 800), které zahrnuje 30 MHz párového spektra, netradičně s downlinkem umístěným níže než uplink, je vidět v tabulce 3.1 [23].

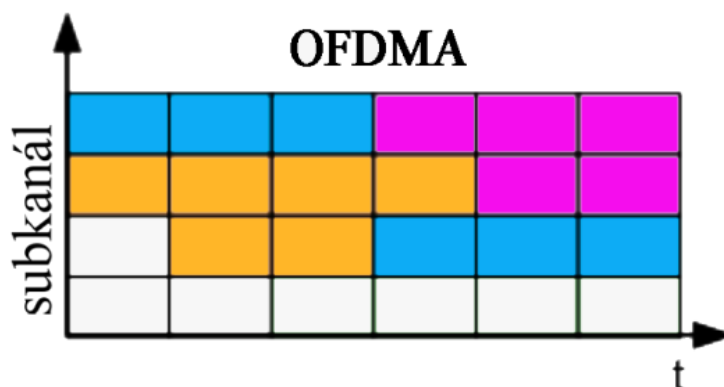
Tabulka 3.1: Rozložení kmitočtů 20. pásma

Operátor	Kmitočet [MHz]
T-Mobile	791 – 796
	796 – 801
O2	801 – 806
	806 – 811
Vodafone	811 – 816
	816 – 821

Pro přenos požadované informace se používají rámce typu 1 - FDD (Frequency Division Duplex) a rámce typu 2 – TDD (Time Division Duplex), které nesou bitový tok, který je do těchto rámců rozdělen. Oba rámce mají dobu trvání 10 ms.

Rámec typu 1 je dále dělen na 10 subrámců a každý subrámec na dva sloty. Sloty obsahují 6 symbolů, pokud je použit rozšířený CP (Cyclic Prefix). V případě, že je použit krátký CP, obsahuje slot 7 OFDM symbolů. Rámec typu 2 je dělen na dva půlrámce a půlrámce jsou děleny na 5 subrámců, přičemž jeden ze subrámců nese uživatelská data, ale informace důležité pro přepínání mezi uplinkem a downlinkem.

Pro přenos dat (uplink a downlink) se v systému LTE používají různé přístupové techniky. Pro downlink se využívá OFDMA (Orthogonal Frequency Division Multiple Access). Jak je vidět na obrázku 3.4, subnosné jsou dělené do skupin a každá tato skupina se nazývá subkanál. Subnosné tvořící subkanál nemusí přiléhat k sobě, ale mohou být rozprostřeny po celém pásmu, což umožňuje vybírat takové kanály, které jsou nejméně rušené, použít vyšší typ modulace (např. 64-QAM), a tím docílit vyšší přenosové rychlosti [24].



Obrázek 3.4: Princip OFDMA

3.3.1 UPLINK

Pro uplink používá systém LTE přístup Single Carrier FDMA. Upload neboli nahrávání dat na vzdálené úložiště je výkonově náročné na mobilní stanici, která data odesílá, proto je třeba myslet i na omezenou kapacitu akumulátorů v mobilních stanicích.

Signál je nejprve podroben kanálovému kódování FEC, je mapován do m-bitových

symbolů podle použité digitální modulace, dále jsou v sériově paralelním převodníku vytvořeny SC-FDMA symboly, které jsou přiváděny na vstup procesoru realizujícího transformaci FFT. Po transformaci je signál rozprostřen tak, že každá subnosná přenáší informace odpovídající všem vstupním datovým symbolům. Dále se symboly mapují na subnosné a signál je přiveden do bloku IFFT, po kterém se do signálu vkládají ochranné intervaly CP (Cyclic Prefix), které jsou tvořeny několika posledními vzorky z OFDM symbolu. Následně je signál převeden do analogové formy a přeložen do příslušného frekvenčního pásma.

3.3.2 DOWNLINK

Pro downlink se používá odlišná metoda multiplexování než u uplinku. Je to OFDMA (Orthogonal Frequency Division Multiple Access) s konstantním rozestupem subnosných 15 kHz. Důvod použití OFDMA v downlinku je vyšší spektrální účinnost, odolnost signálu při průchodu únikovým kanálem, jednodušší implementace MIMO apod. [20].

Datový signál je podrobený FEC (Forward Error Correction) a prokládán, dále je mapován do m-bitových symbolů podle použité modulace. Vytvořené OFDM symboly jsou transformovány IFFT a je vložen CP, jehož délka je obvykle 4,7 μ s, což je pro malé buňky dostačující interval pro omezení interferencí. Ve větších buňkách může být použit CP s trváním až 16,7 μ s. Následně je signál převeden do analogové podoby a konvertován do vysokofrekvenčního pásma.

Pro downlink jsou podporovány vnitřní modulace QPSK, 16-QAM a 64 QAM.

3.3.3 Systém IMS

IMS (IP Multimedia System) je globální, přístupově nezávislá architektura, jejíž standard je založen na protokolu IP. Hlavní myšlenkou IMS je sjednotit přenos hlasu a dat. Standardizován byl v roce 2003 organizací 3GPP (Third Generation Partnership Project) jako součást specifikace Release 5. IMS rozšiřuje architekturu stávajících GSM/UMTS sítí o část pro multimediální přenosy pomocí technologie přepínání paketů. Multimediálními přenosy jsou myšleny jak běžné telefonní hovory, konferenční hovory, tak i libovolné přenosy zvuku a videa včetně streamingu. GSM/UMTS sítě využívají pro přenos telefonních hovorů technologii přepínání okruhů, pro přenos dat technologii přepínání paketů. Síť LTE jsou už plně založené na protokolu IP. Pro veškerou komunikaci používají technologii přepínání paketů a pro signalizaci protokol SIP (Session Initiation Protocol).

Komunikace v IMS využívá převážně dvou protokolů, a to protokolu SIP, který se používá pro sestavení, udržení a ukončení spojení a protokolu Diameter, jenž je využíván pro autentizaci, autorizaci a účtování. Pro vlastní přenos dat je využíván protokol IP.

3.3.4 Protokol IP

Protokol IP definuje základní jednotku dat, která je soustavou TCP/IP sítí přenášena na úrovni síťové vrstvy, tzv. Internet Protocol datagram, zkráceně datagram, a jeho přesný vnitřní formát. Definuje také způsob, jakým mají být jednotlivé IP datagramy směrovány, a toto směrování pak příslušný IP software také zajišťuje. Protokol IP

poskytuje datagramovou službu celé rodině protokolů TCP/IP. Sám o sobě neposkytuje záruky na přenos dat a rozlišuje pomocí IP adresy pouze jednotlivá síťová rozhraní (doplňující služby jsou poskytovány na vyšších vrstvách, viz referenční model ISO/OSI). V současné době je stále ještě používána starší verze protokolu IPv4, nově se přechází na IPv6.

Data se v IP síti posílají po blocích nazývaných datagramy. Jednotlivé datagramy putují sítí zcela nezávisle, takže k cílové stanici mohou dorazit v jiném pořadí, než ve kterém byly odeslány. Protokol IP zajišťuje směrování datagramů (paketů) ze zdrojového počítače do cílového hostitele přes jednu nebo více IP sítí. Paket se skládá z řídicích dat (metadat) a z uživatelských dat (payload). Řídicí data poskytují síti potřebná data k doručení paketu, například adresu zdroje a cíle a kódy pro detekci chyb.

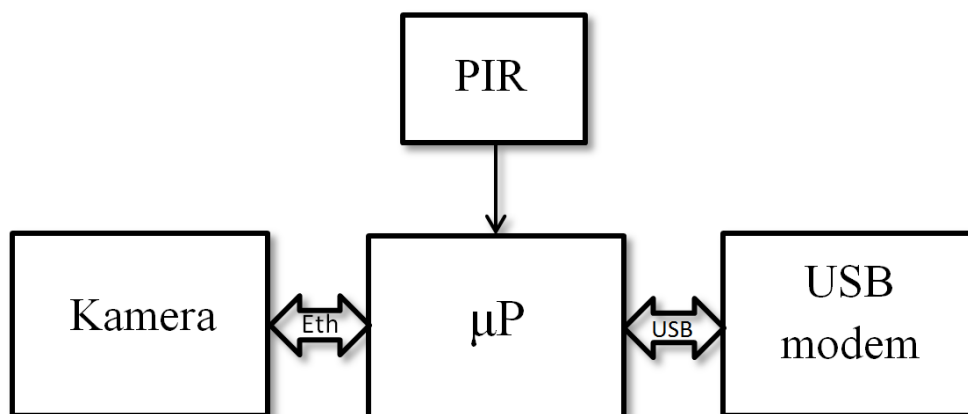
Každé síťové rozhraní komunikující prostřednictvím IP má přiřazeno jednoznačný identifikátor, tzv. IP adresu. V každém datagramu je pak uvedena IP adresa odesílatele i příjemce. Na základě těchto adres pak směrovače (routery) na trase provádí rozhodnutí, jakým směrem paket odeslat, tzv. směrování (routing).

4 NÁVRH ZAPOJENÍ BEZPEČNOSTNÍHO SYSTÉMU

Základní blokové schéma navrženého systému je uvedeno na obrázku 4.1. Celý systém je složen ze čtyř základních částí a je řízen mikroprocesorem. Pro tuto funkci byla vybrána vývojová deska Arduino YUN, na které jsou k dispozici porty pro ethernet i USB. Deska obsahuje mikroprocesor AVR ATmega32u4, jehož parametry jsou uvedeny v tabulce 4.1.

Tabulka 4.1: Základní parametry ATmega32u4 [26]

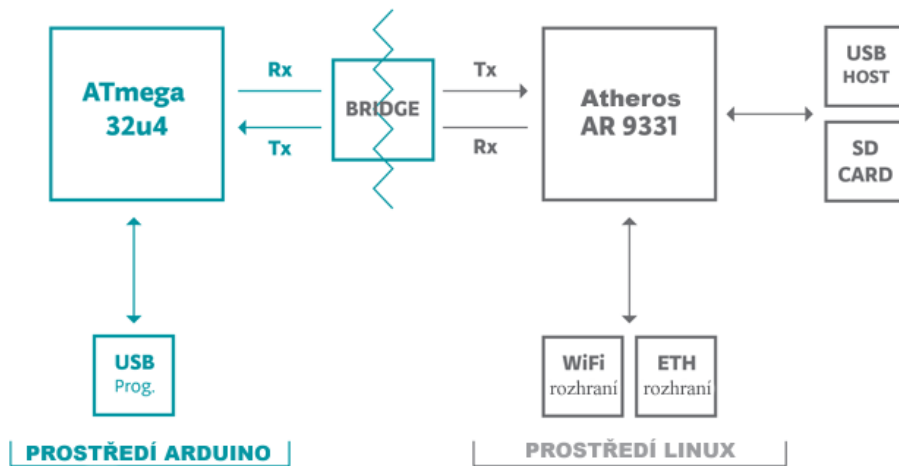
Čip	ATmega32u4
Frekvence	16 MHz
Digitální I/O piny	20
Paměť	32 kB, 4 použity bootloaderem
SRAM	2,5 kB
EEPROM	1 kB



Obrázek 4.1: Základní blokové schéma bezpečnostního systému

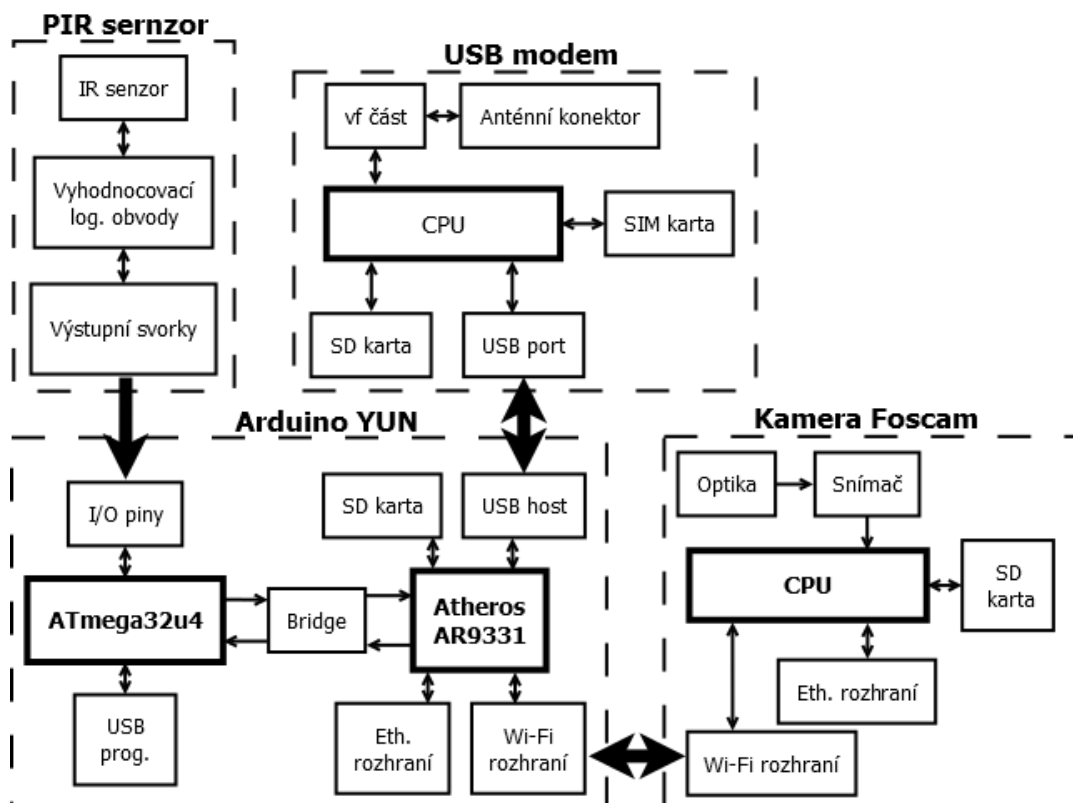
Jak je vidět na obrázku 4.2, deska Arduino obsahuje ještě druhý, linuxový procesor Atheros AR9331, který zajišťuje síťovou komunikaci. Kromě ethernetového rozhraní je možné i připojení k síti Wi-Fi. O komunikaci mezi procesorem ATmega a Atheros se stará „Bridge“ (most), jehož knihovna je volně k dispozici na stránkách arduino.cc.

Ve speciálním programovacím prostředí Arduino IDE, které je vyvinuto zejména pro vývojové desky Arduino, se programuje pomocí jazyku Wiring. Wiring je podobný jazyku C++, někde se dokonce uvádí, že se jedná o jakýsi metajazyk, či vylepšenou sadu maker nad C++.



Obrázek 4.2: Blokové schéma Arduina [26]

Na obrázku 4.3 je uvedeno podrobné blokové schéma zapojení bezpečnostního systému. Čtyři hlavní části systému jsou ohraničeny přerušovanou čarou. Uvnitř těchto bloků je podrobněji nakresleno vnitřní zapojení jednotlivých komponent. Je zde také vidět, že ke komunikaci mezi kamerou a Arduinem se nevyužívá původně plánovaná ethernetová komunikace, ale uživatelsky přívětivější komunikace bezdrátová.



Obrázek 4.3: Podrobné blokové schéma bezpečnostního systému

4.1 Komunikace PIR senzoru s deskou Arduino

Při testování zvoleného PIR senzoru LS-818-3 (viz obrázek 1.8 b) byl naměřen na výstupních alarmových svorkách pulsní průběh signálu, což by mohlo znamenat problém při vyhodnocování aktivního alarmu mikroprocesorem. Výstupní svorky tedy byly naletovány paralelně k červené informativní diodě, která je vždy během alarmu pod napětím. Při aktivaci čidla je tedy na nově vyvedených svorkách napětí vyšší než 2,1 V. V případě, kdy čidlo aktivní není, je na svorkách nulové napětí. Tento výstupní signál ze senzoru je přiveden na digitální vstup č. 3 desky Arduino YUN.

Standardními příkazy je pin č. 3 nastaven jako vstupní.

```
pinMode(pin3, INPUT);
```

Jestli je na svorkách čidla napětí nebo ne je sledováno v nekonečné smyčce hlavní programové části. Pokud ano, program pokračuje vykonáváním příkazů pro ovládání kamery atd.

```
void loop()
{
    if (digitalRead(pin3))
    {
        // spuštění alarmové sekvence
    }
}
```

4.2 Komunikace IP kamery s deskou Arduino

Kamera Foscam FI9821P má k dispozici jak připojení pomocí ethernetového kabelu, tak i prostřednictvím Wi-Fi sítě. Samotnou kameru lze ovládat prostřednictvím počítače velmi snadno. Pokud je kamera ve stejné síti jako počítač, je k dispozici přehledné uživatelské prostředí, ke kterému se lze připojit pomocí programů, které jsou dodávány spolu s kamerou. V případě diplomové práce ale tento software využít není.

V případě diplomové práce není kamera připojena pomocí ethernetu, ale prostřednictvím Wi-Fi sítě. Kromě úspory ethernetového kabelu je zde další velká výhoda tohoto připojení. V budoucnu je možné rozšířit tento bezpečnostní systém pro více kamer, z nichž každá bude připojena pomocí bezdrátové sítě. Nebude tedy docházet k problému s nedostatkem ethernetových přípojek na desce Arduino a současně bude každá kamera identifikovatelná svojí IP adresou.

Samotné nastavování a ovládání kamery je prováděno prostřednictvím CGI. Software Foscam je pouze nadstavbou nad toto rozhraní. V tomto projektu je nutné pracovat přímo s příkazy CGI [31]. Ke komunikaci s kamerou pomocí CGI dochází pouze ve třech případech. Při požadavku na spuštění nahrávání, požadavku na vypnutí nahrávání a při nastavování času.

Jedním z příkazů, který je v práci použit, je příkaz pro nastavení aktuálního času kamery. Volaná funkce *setTime* získá z internetu aktuální čas a rozdělí ho do potřebných proměnných *day*, *mon*, *year*, *hour*, *minute* a *sec*. Takto připravený čas je odeslán do kamery v podobě CGI příkazu, kde se ve hlavičce vždy nachází cílová IP

adresa a typ příkazu. V tomto případě se jedná o nastavování systémového času kamery, tedy *setSystemTime*. Následuje formát času a jednotlivé proměnné, které jsou vkládány na potřebná místa. Celý příkaz je vždy zakončen autorizačními údaji *usr* a *pwd*.

```
"http://192.168.240.112/cgi-bin/CGIProxy.fcgi?cmd=setSystemTime&timeSource=1&dateFormat=1&timeFormat=1&year="+year+"&mon="+mon+"&day="+day+"&hour="+hour+"&minute="+minute+"&sec="+sec+"&usr="+usr+"&pwd="+pwd
```

Při vyhodnocení alarmu je volána funkce s názvem *alarmOn*, která do kamery odešle velmi dlouhý příkaz na spuštění nahrávání. Po hlavičce následuje typ příkazu *setScheduleRecordConfig*, což znamená, že se bude nastavovat kalendář plánovaného nahrávání. Jelikož v tomto případě neexistuje žádný příkaz na přímé zapnutí alarmu a tím pádem i spuštění nahrávání, jednou z možností je nakonfigurovat plánované nahrávání po celý den od pondělí do neděle. Ve zdrojovém kódu je vidět např. parametr *schedule0*, který představuje pondělí. A u něj číslo 281474976710655, které v binární podobě představuje 48 bitové číslo samých jedniček. Každá půlhodina dne je zastoupena jedním bitem, a pokud se v daný čas má nahrávat, je bit nastaven do jedničky, v opačném případě do nuly. Jakmile je odeslán do kamery tento příkaz, kamera začne okamžitě nahrávat až do chvíle, kdy je jiným příkazem nahrávání ukončeno. Parametr *isEnabled* povoluje plánované nahrávání. Pokud je nastaven na 1, lze nastavovat časy, během kterých se bude nahrávat.

```
"http://192.168.240.112/cgi-bin/CGIProxy.fcgi?cmd=setScheduleRecordConfig&isEnabled=1&isEnabledAudio=1&schedule0=281474976710655&schedule1=281474976710655&schedule2=281474976710655&schedule3=281474976710655&schedule4=281474976710655&schedule5=281474976710655&schedule6=281474976710655&usr=adasvobi&pwd=adasvobi"
```

Podobná je i funkce *alarmOff*, která deaktivuje plánované nahrávání. Ve stejném příkazu je nastaven parametr *isEnabled* do nuly, a tím je nahrávání okamžitě ukončeno. Pokud je plánované nahrávání deaktivováno, není nutné vyplňovat ostatní parametry.

```
"http://192.168.240.112/cgi-bin/CGIProxy.fcgi?cmd=setScheduleRecordConfig&isEnabled=1&usr=adasvobi&pwd=adasvobi"
```

4.3 Práce s ATmega32u4

Program pro ovládání vstupů i výstupů bezpečnostního systému je nahrán do mikroprocesoru ATmega32u4. Mimo několik základních instrukcí, které zde již byly uvedeny, mikroprocesor zajišťuje i složitější komunikaci mezi připojenými perifériemi, včetně komunikace mezi mikroprocesorem ATmega a procesorem Atheros. Ta probíhá přes sériovou linku a je zapouzdřena v knihovně *bridge.h*. Díky této knihovně je možné využívat veškeré funkce linuxového prostředí přímo z ATmega32u4.

Vývojový diagram programu je zobrazen na obrázku 4.3. Je zde vidět, že po zapnutí napájení je nutné čekat na inicializaci modemu. K tomu dochází v linuxovém prostředí, kde operační systém pomocí ovladačů komunikuje s USB modemem, zadá PIN kód a APN (Access Point Name – Název přístupového bodu) a pomocí následujících AT příkazů se pokouší navázat internetové spojení.

```

timeout set to 12 seconds
send (AT&F^M)
expect (OK)
AT&F^M^M
OK
send (ATE1^M)
expect (OK)
^M
ATE1^M^M
OK
send (AT+CGDCONT=1,"IP","internet.t-
mobile.cz" ^M)
abort on (NO CARRIER)
timeout set to 15 seconds
expect (OK)
^M
AT+CGDCONT=1,"IP","internet.t-
mobile.cz"^M^M
OK
send (ATD*99***1#^M)
expect (CONNECT)
^M
ATD*99***1#^M^M
CONNECT
send ( ^M)

```

K navázání komunikace s internetem dojde i při opakovaném připojení do jedné minuty.

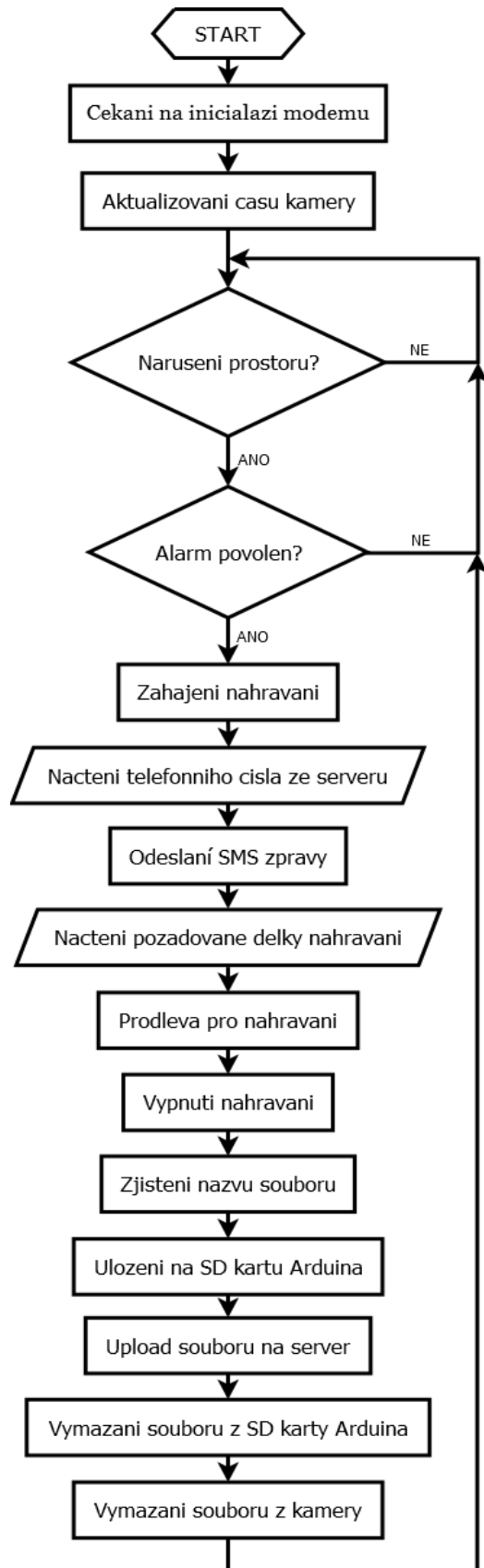
Po inicializaci modemu dojde k automatické synchronizaci času Arduina s časem na internetu. Tento přesný čas je dále nastaven i v kameře pomocí CGI příkazu.

Dále následuje hlavní programová smyčka *loop*, která se opakuje v nekonečné smyčce. Na jejím začátku se program podle podmínky dělí do dvou větví. První větev, pokud prostor není narušen, znamená neustále sledování hodnoty na digitálním vstupu procesoru. Druhá větev představuje alarm pohybového čidla a tím pádem i narušení prostoru. Pokud je vyhodnocen alarmový stav, další *if* vyhodnocuje, jestli si uživatel přeje být o alarmu informovaný nebo objekt v daný moment strážěn není.

```

if (digitalRead(pin3))
{
  if (findAlarmEnable())
  {
    // sekvence nahrávání a uploadování
  }
}

```



Obrázek 4.3: Vývojový diagram programu v desce Arduino

Zda uživatel má být informován či nikoli zjišťuje funkce *findAlarmEnable*. Ve funkci jsou použity příkazy, které jsou obsaženy v knihovně *HttpClient.h*. Tato knihovna umožňuje vytvořit základního HTTP klienta [32], který se připojí k internetu a může stahovat obsah. V tomto případě se stahuje obsah textového souboru, který je uložen na FTP serveru. Dokud je klient aktivní, ukládá se obsah souboru do proměnné *help* a je dále vyhodnocován.

```
boolean findAlarmEnable()
{
    ...
    HttpClient client;
    String findAlarmEnable = F("adresa_serveru/alarmEnable.txt");
    client.get(findAlarmEnable);
    while (client.available())
    {
        char c = client.read();
        help += c;
    }
    Serial.flush();
    if (help[0] == '1') alarmEnabled = true;
    else alarmEnabled = false;
    return (alarmEnabled);
}
```

Příkaz *client.get()* zadá požadavek, a pokud je to požadavek na textový soubor, zpět se vrací textový obsah souboru. Pokud je obsah textového souboru 1, je výstupní parametr nastaven na *true* a pokračuje se dále ve větvi.

Pokud jsou obě podmínky splněny, následuje funkce pro zapnutí nahrávání kamery *alarmOn*, která je již rozepsána výše.

Další funkce se jménem *sendSMS* zajišťuje odeslání informativní SMS zprávy, že je prostor narušen. K odeslání zprávy je třeba znát telefonní číslo, na které se zpráva odešle. Funkce *findNumber*, která se stará o nalezení telefonního čísla, pracuje velmi podobně jako funkce *findAlarmEnable*, tedy pomocí HTTP klienta vyčte telefonní číslo ze souboru *phoneNumber.txt*. Jakmile je telefonní číslo uloženo v proměnné, je volán program uložený na paměťové kartě Arduina.

```
void sendSMS()
{
    Process p;
    p.begin("python");
    p.addParameter("/mnt/sda1/arduino/www/sms.py");
    p.addParameter(phoneNumber);
    p.run();
}
```

Je k tomu využito následujících procesů. Knihovna *Process.h* napomáhá prostřednictvím mikroprocesoru ATmega využívat linuxové příkazy v procesoru Atheros [33]. Nejprve je pomocí *p.begin(...)* inicializován proces *p*. Dále jsou pomocí příkazů *p.addParameter()* přidávány potřebné parametry a nakonec je celý proces spuštěn příkazem *p.run()*. Takto složený příkaz představuje volání Pythonového programu *sms.py*, který je uložen v */mnt/sda1/arduino/www/*, což představuje úložiště paměťové karty. Program je volán s parametrem *phoneNumber*, aby bylo možné nastavit telefonní číslo, na které se SMS zprávy budou odesílat.

Odesílání SMS zpráv bylo původně řešeno zadáváním linuxových příkazů prostřednictvím procesů. Bylo využito příkazů *cat* a *echo*, které umožnily navázat sériovou komunikaci s USB modemem Huawei. K této komunikaci byly využity AT příkazy, které po nastavení SMS módu a telefonního čísla odeslaly SMS zprávu.

```
p.runShellCommand("echo AT+CMGF=1 > /dev/ttyUSB1")
p.runShellCommand("echo AT+CMGS=\"+420123456789\" > /dev/ttyUSB1")
p.runShellCommand("echo \"text\32\" > /dev/ttyUSB1")
```

Příkaz na otevření sériového portu a sledování odpovědi USB modemu, ve kterém však nešly příkazy zadávat, byl následující:

```
cat /dev/ttyUSB1
```

Tato komunikace byla velmi nespolehlivá, protože AT příkazy byly modemem přijaty jen v polovině případů, druhá polovina příkazů se odpovědí nedočkala vůbec nebo na ně bylo odpovězeno hlášením „ERROR“, které se odesílá v případě, že byl odeslán neznámý AT příkaz. Další nevýhodou tohoto způsobu byla malá rychlost komunikace bez možnosti sledování odezvy. Každý proces musel být napsán a odeslán separátně, což chod mikroprocesoru velmi zpomalovalo.

Další metoda odesílání informativní SMS zprávy byla s využitím linuxového programu *minicom* pro komunikaci se sériovými zařízeními. Do základní verze linuxu byl tedy doinstalován tento program a spolu s ním i doporučené ovladače *kmod-scsi-cdrom* a *kmod-scsi-generic*. K testování tohoto programu byl použit příkazový řádek linuxového prostředí pracující s protokolem SSH (Secure Shell). Zadáním příkazu

```
minicom -s
```

se otevřela přehledná tabulka s nastavením sériových portů a jejich rychlostmi. Po uložení konfiguračního okna již stačilo napsat příkaz

```
minicom -c on
```

a otevřela se sériová komunikace s USB zařízením. Zde stačilo pomocí AT příkazů zadávat potřebná hesla a SMS zpráva byla bez problémů odeslána. Problém však nastal při ovládní Minicomu pomocí procesů mikroprocesoru ATmega. Knihovna procesů nebyla schopna inicializovat tento doinstalovaný program, a tím pádem ani komunikovat po sériové lince.

Konečným řešením je již výše uvedené volání Pythonového programu (viz příloha). Takto zvolený způsob odesílání SMS zprávy je z výše uvedených jediný korektně funkční a během testování asi 200 SMS zprávami nedošlo k jediné chybě. Prodleva mezi zavoláním programu a doručení SMS zprávy do mobilního telefonu je proměnlivá, nejčastěji však mezi 3-4 sekundami. Telefonní číslo je zadáno parametrem a informativní text zprávy je zapsán v souboru *sms.py*. Použit je stručný, avšak výstižný text „Alarm“. Pokud uživateli přijde textová zpráva z telefonního čísla USB modemu s tímto textem, není pochyb o tom, že došlo k narušení střeženého prostoru.

Pokud by tento bezpečnostní systém byl použit pro střežení více prostor, bylo by nutné použít více pohybových čidel i kamer, a v SMS zprávě by bylo nutné upřesnit, který prostor byl narušen. K tomu by opět bylo využito volání Pythonového programu s parametrem. Tentokrát by nestačil parametr jeden (telefonní číslo), ale byl by potřeba i druhý, který by do SMS zprávy zdůraznil, o který narušený prostor se jedná.

Po odeslání zprávy mikroprocesor čeká, až nebude aktivní výstup pohybového

čidla. Dokud je výstup aktivní, znamená to, že se po střeženém objektu stále někdo pohybuje a kamera musí dále nahrávat. Jakmile je PIR senzor neaktivní, pokračuje program na funkci *findDuration()*, jejíž výstup je v číselné podobě uložen do proměnné a představuje čas, po který se bude video nahrávat po skončení aktivního stavu pohybového čidla. Následuje příkaz *delay*, který pozastaví procesor na dobu, která byla vyčtena výše uvedenou funkcí. Během této doby je však kamerou stále pořizován videozáznam, který je průběžně ukládán na paměťovou kartu kamery.

Teprve v této chvíli je funkcí *alarmOff* odeslán do kamery příkaz na ukončení nahrávání. Jak je psáno výše, stačí nastavit parametr CGI příkazu *isEnabled* do nuly. Než je možné videosoubor zpracovávat a kopírovat, je nutné krátký čas počkat, aby se videosoubor uzavřel do kontejneru a uložil celý na paměťovou kartu. Proto je do programu vloženo několikasekundové zpoždění, které tento nepříznivý stav eliminuje.

Následují funkce pro zjištění cesty k nahranému souboru v kameře *lastDate*, *lastTime* a *fileNameNew*. Typická cesta k videosouboru může vypadat takto:

```
ftp://adasvobi:adasvobi@192.168.240.112:50021/IPCamera/FI9821P_00626E517E62/record/20150220/20150220_151313/schedule_20150220_164016.avi
```

Jak je vidět, cesta je velmi dlouhá a potřebný videosoubor může být uložen v různých složkách. Cesta k souborům je z velké části neměnná, až složky *20150220* a *20150220_151313* se mění podle aktuálního data a času záznamu. Ke zjištění cesty k videosouborům je opět využito procesů, tentokrát se vstupním parametrem *curl*. Jelikož úložiště IP kamery je v lokální síti vidět jako pasivní FTP server, je při sestavování připojení nutné poslat příkazem i IP adresu Arduina a port, na kterém naslouchá. V příkazovém řádku by takový příkaz vypadal následovně:

```
curl --ftp-port 192.168.240.1 ftp://...adresa_kamery...
```

Jako sekvence příkazů v mikroprocesoru desky Arduino vypadá takto:

```
p.begin("curl");
p.addParameter("--ftp-port");
p.addParameter("192.168.240.1");
p.addParameter("ftp://adasvobi:adasvobi@192.168.240.112:50021/
                IPCamera/FI9821P_00626E517E62/record/");
p.run();
```

Odpovědí na takovou sekvenci je řádkový výpis složek, které jsou umístěny ve složce *record*. Tuto odezvu Arduino ukládá do proměnné a následně z ní vyčte nejaktuálnější složku. Jakmile je znám název nejnovější složky, funkce s názvem *lastTime* provede téměř identický postup a získá název složky s nejaktuálnějším časem. Díky abecednímu řazení stačí zaznamenat složku na posledním místě. Totožný postup je praktikován i při zjišťování názvu videosouboru funkcí *fileNameNew*. Po proběhnutí těchto tří funkcí na získání cesty k souborům IP kamery jsou v proměnných *newDateSch*, *newTimeSch* a *fileName* kompletní informace o cestě k potřebnému souboru.

Následuje funkce s názvem *downloadToSd*, která je volána právě s parametry, které jsou uvedeny výše, tedy s informací, kde v kameře se nachází právě nahraný soubor. V těle této krátké funkce je definován proces, který do lokální sítě odešle příkaz na stažení souboru na paměťovou kartu v Arduinu. Soubor je stažen do složky *Bridge*, která se sama vytvoří.

```
p.runShellCommand("wget  
ftp://adasvobi:adasvobi@192.168.240.112:50021/IPCamera/FI9821P_00626E5  
17E62/record/"+newDateSch+"/"+newTimeSch+"/"+fileName);
```

Je vidět, že v této funkci se k procesům přistupuje jiným způsobem než dosud, tedy příkazem *runShellCommand*. Funkce tohoto příkazu je stejná jako při použití sekvence *p.begin*, *p.addParameter* a *p.run* a i pracovat by měly totožně. Bohužel tomu tak není. Například při volání programu *wget*, který slouží jako jednoduchý stahovač souborů, lze použít pouze *runShellCommand*. Při použití *p.begin*(„*wget*“) se příkaz odešle, ale nevykoná se. To je způsobeno nesprávnou komunikací mezi knihovnami *Process.h* a *Bridge.h*. Následující příkaz

```
while(p.running());
```

má za úkol sledovat, jestli už je proces dokončený. Pokud dokončený není, stále se opakuje smyčka *while*. Jakmile se proces dokončí, pokračuje se dále ve vykonávání programu. Díky nízké přenosové rychlosti paměťové karty a omezené přenosové rychlosti procesoru Atheros je tato funkce výpočetně a tedy i časově nejnáročnější.

Nyní už stačí soubor na paměťové kartě odeslat na vzdálený FTP server. K tomu slouží funkce *uploadToFtp*. Pomocí procesů a příkazu *curl* s vhodně zvolenými parametry je možno odeslat soubor na konkrétní adresu.

```
p.runShellCommand("curl -T /usr/lib/python2.7/bridge/"+fileName+"  
ftp://ms2.urel.feec.vutbr.cz/Diplomova_prace/Svoboda/Video/ --user  
student:urel");
```

V tomto případě je jako úložný FTP server použit *ms2.urel.feec.vutbr.cz*, kde je pro testování diplomové práce vyhrazena speciální složka. Autorizační údaje k serveru jsou odesílány pomocí parametru *--user*. Dokud soubor není kompletně odeslán, program čeká ve smyčce.

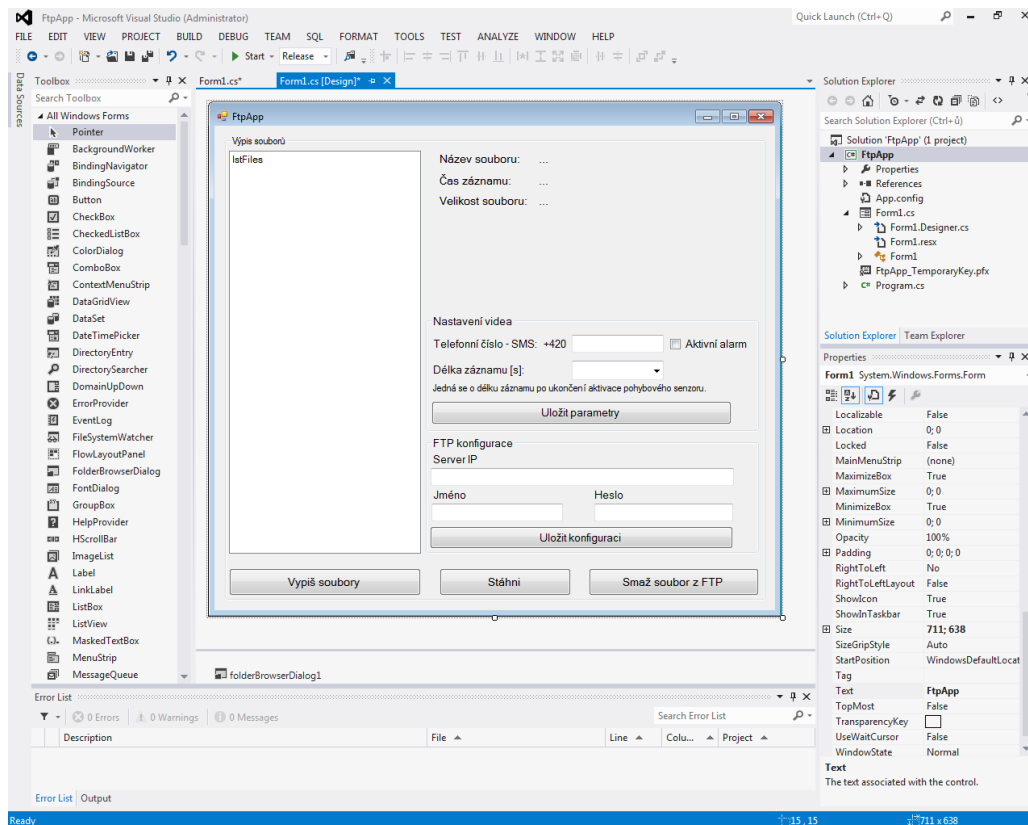
Po odeslání souboru již stačí funkcemi *removeFile* a *removeFileFromCam* odstranit videosoubor z paměťové karty Arduina a kamery, aby bylo úložiště připraveno na nahrání dalšího souboru. V obou případech se využije proces, který odešle požadavek na smazání konkrétního souboru.

Po vymazání souboru je program opět vrácen na začátek smyčky *loop* a čeká, až přijde impuls od pohybového detektoru.

4.4 Práce s programem FtpApp

Pro pohodlný a přehledný výpis souborů na FTP serveru byl v prostředí Visual Studio 2012 napsán program pro správu videí (viz obrázek 4.4).

Okno programu (viz obrázek 6.9), psaného v C#, je rozděleno do několika přehledných částí, z nichž každá část má svoji funkci. Je zde blok pro nastavování autorizačních údajů, potřebných pro připojení k FTP serveru, dále blok s možností nastavení telefonních údajů a délky videa, v levé části je pole pro výpis souborů a v horní části je prostor pro výpis informací z jednotlivých souborů. Na spodní části okna jsou tři ovládací tlačítka.



Obrázek 4.4: Vývojové prostředí Visual Studio

4.4.1 Nastavení parametrů Arduina

V okně programu FtpApp je k dispozici prostor pro nastavení délky záznamu videa po ukončení aktivace pohybového senzoru a telefonního čísla, kam se odešle SMS zpráva oznamující narušení prostoru. Nad rámec diplomové práce bylo doplněno pole *Aktivní alarm*, které dovoluje vzdáleně deaktivovat sítězení prostoru.

Celý tento blok je pojmenován *Nastavení videa* a jednotlivé komponenty jsou umístěny ve skupině oken, neboli *GroupBoxu*. Uvnitř je textové pole pro vypsání telefonního čísla. Toto pole, neboli *TextBox*, je ošetřeno tak, aby bylo možno vepsat pouze devítimístné číslo, a tak předejít problémům s nesprávným formátem telefonního čísla. Zdrojový kód pro takovou podmínku je volán při každém stisknutí klávesy uvnitř textového pole pro telefonní číslo a vypadá následovně:

```
if (!char.IsDigit(e.KeyChar) && e.KeyChar != 8)
{
    e.Handled = true;
}
```

Pro nastavování délky záznamu videa po ukončení aktivace pohybového senzoru je k dispozici seznam hodnot, neboli *ComboBox*, ve kterém je možnost výběru 1, 2, 5, 10, 15, 20, 25 nebo 30 sekund. Posledním prvkem v této sekci je zaškrťací políčko *Aktivní alarm*, neboli *CheckBox*, jehož funkce je popsána výše.

Všechny tři parametry jsou při stisku tlačítka *Uložit parametry* uloženy do proměnných a dále zpracovávány. Pro každou proměnnou je navíc vytvořen textový soubor na disku počítače, do kterého se tato proměnná uloží. V případě telefonního čísla se do souboru *phoneNumber.txt* navíc před samotné číslo uloží i symboly „+420“, což je předvolba telefonního čísla pro Českou republiku. Jakmile jsou textové soubory vytvořeny, jsou pomocí webového klienta odeslány na FTP server, kde jsou k dispozici pro nastavení samotného mikroprocesoru. Při použití webového klienta je nejprve nutné do kódu přidat odkazy na komponenty webového projektu sady Visual Studio [34]. Tento odkaz je napsán na začátku zdrojového kódu ve tvaru:

```
using System.Net;
```

S jeho pomocí je možno využívat základní internetové funkce. Pro odeslání souboru na server je třeba nejprve zadat autorizační údaje a teprve potom odeslat samotný textový soubor. V sekvenci příkazů pro odeslání souboru s telefonním číslem je nejprve deklarován webový klient, k němuž jsou dále připojovány parametry:

```
using (System.Net.WebClient client = new System.Net.WebClient())
{
    client.Credentials = new System.Net.NetworkCredential(ftpUserID,
        ftpPassword);
    client.UploadFile("ftp://" + ftpServerIPdata + "/" + new
        FileInfo(fileDur).Name, "STOR", fileDur);
}
```

Přihlašovací údaje (Credentials) jsou uloženy v proměnných *ftpUserID* a *ftpPassword*. K samotnému odeslání souboru je využito metody *client.UploadFile*, která obsahuje proměnné *ftpServerIPdata* a *fileDur*. V nich je uložena cesta k FTP serveru a název souboru, který má být na server odeslán.

Totožným postupem jsou odeslány i zbývající dva textové soubory, samozřejmě s odpovídajícími proměnnými. Pokud již na serveru je takový soubor nahrán, ale pomocí programu je odesílán nový s novými hodnotami, starý soubor se ze serveru nejprve smaže a teprve potom se vloží soubor nový.

4.4.2 Nastavení údajů FTP serveru

Pro nastavení základních přihlašovacích údajů a adresy serveru je v okně vyčleněna část s názvem *FTP konfigurace*. K dispozici jsou textová pole *txtServerIP*, *txtUsername* a *txtPassword* s předvyplněnými informacemi. V poli pro zadání hesla je z bezpečnostních důvodů nastavena vlastnost pro skrytí napsaného hesla. Tato vlastnost *PasswordChar* umožňuje nahradit symboly hesla za libovolný znak, v tomto případě „*“. Do pole pro adresu serveru je možné psát adresu jak ve formátu *ms2.urel.fec.vutbr.cz*, tak i ve formátu *147.229.144.119*.

Při jakékoliv změně autorizačních údajů je odkryto tlačítko *Uložit konfiguraci*, které je až do této chvíle skryto. Stiskem tohoto tlačítka se text vepsaný v polích uloží do globálních proměnných *ftpServerIP*, *ftpUserID* a *ftpPassword*, a ty jsou dále využívány k přístupu k vzdálenému FTP serveru.

4.4.3 Výpis souborů

Hlavní funkcí programu *FtpApp* je přehledný výpis souborů uložených na FTP serveru. Po stisku tlačítka *Vypiš soubory* je v textovém poli na levé straně okna, neboli *ListBoxu*, zobrazen přehledný seznam všech videí v abecedním pořadí. Videá jsou defaultně ukládána ve formátu „schedule_RRRRMMDD_HHMMSS.avi“, kde RRRR představuje rok záznamu, MM – měsíc, DD – den v měsíci, HH – hodinu, MM – minutu a SS – sekundu záznamu. To znamená, že při abecedním řazení odpovídá i řazení časovému.

Tlačítko pro vypsání souborů je umístěno v levé dolní části okna a využívá třídu *FtpWebRequest*, která implementuje protokol FTP klienta [35]. Po stisku tlačítka je volána funkce *update_lstText()*, která má aktualizovat položky v *ListBoxu*. Nejprve je však nutné stáhnout ze serveru kompletní seznam všech souborů. Proto je volána funkce *GetFileList()*, která do proměnné *filenames* vrací seznam souborů na serveru. V této funkci je implementován FTP klient *reqFTP*, kterému jsou dále vkládány vlastnosti jako přihlašovací údaje *ftpUserID* a *ftpPassword*, IP adresa serveru *ftpServerIP* a požadovaný výstup požadavku. V tomto případě je to požadavek na vypsání obsahu adresáře, neboli *ListDirectory*.

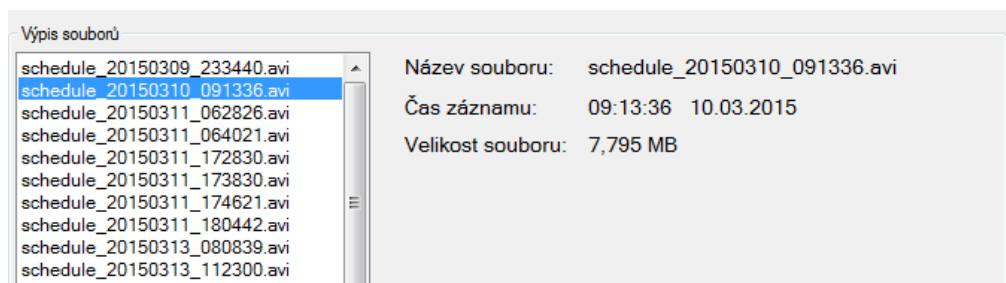
```
reqFTP = (FtpWebRequest)FtpWebRequest.Create(new Uri("ftp://" +  
    ftpServerIP + "/"));  
reqFTP.UseBinary = true;  
reqFTP.Credentials = new NetworkCredential(ftpUserID, ftpPassword);  
reqFTP.Method = WebRequestMethods.Ftp.ListDirectory;
```

Jakmile je požadavek správně deklarován, pomocí třídy *WebResponse* je požadována odezva, která je za pomoci *StreamReaderu* uložena do proměnné. Dále jsou názvy souborů řádek po řádku vkládány do proměnné *result*, která je následně přetypována na datový typ *String*. Nyní už stačí pouze každý řádek proměnné vložit jako nový objekt *ListBoxu*.

Při správném zadání přihlašovacích údajů a IP adresy serveru je po stisku tlačítka vypsán seznam video souborů, které jsou uloženy serveru.

4.4.4 Podrobnosti vybraného souboru

V textovém poli jsou vypsány všechny soubory, které jsou uloženy ve složce pro ukládání videosouborů. Jejich formát je ale trochu nepřehledný, proto je v pravé horní části okna prostor pro podrobnější popis vybraného souboru. Při výběru souboru je daný řádek zvýrazněn a jsou vypsány jeho podrobnější informace (viz obrázek 4.5).



Obrázek 4.5: Informace vybraného souboru

Vybráním souboru je do proměnné uložen index řádku textového pole, který je právě označen. Index představuje číslo řádku a je ukládán do stejnojmenné proměnné *index*. Pomocí čísla řádku je v kompletním seznamu souborů vyhledán odpovídající název souboru a následně vypsán v okně programu. Z názvu je pomocí metody *Substring* postupně vyčteno datum a přesný čas záznamu, který je přehledně vypsán v pravé části okna.

```
String sec = item.Substring(22, 2);
String min = item.Substring(20, 2);
String hrs = item.Substring(18, 2);
String day = item.Substring(15, 2);
String month = item.Substring(13, 2);
String year = item.Substring(9, 4);
label_time.Text = hrs + ":" + min + ":" + sec + "   " + day + "."
+ month + "." + year;
```

Při vypisování velikosti souboru je postup o něco komplikovanější a opět se musí využít požadavku na FTP server. Prvním krokem je nastavení vlastností daného požadavku, tedy zadání jména, hesla, IP adresy serveru a cesty k danému souboru. Dále je zadána metoda požadavku *GetFileSize*:

```
request.Method = WebRequestMethods.Ftp.GetFileSize;
```

Odpovědí je velikost daného souboru v bytech, která je následně dělena tak, aby výsledkem byla velikost v MB. Následně je vypsána na tři desetinná místa do okna programu.

4.4.5 Uživatelská tlačítka

Mimo tlačítka *Vypiš soubory*, jehož funkce byla popsána výše, jsou v programu ještě tlačítka *Stáhni* a *Smaž soubor z FTP*, která uživateli usnadní práci se soubory na vzdáleném serveru.

Jak už napovídá název tlačítka *Stáhni*, slouží toto tlačítko ke stažení video souboru na disk počítače, kde může být soubor přehrán. Po stisku tlačítka jsou deklarovány proměnné, které budou během stahování souboru potřebné. Za zmínku stojí proměnná *buffer*, která má datový typ *byte* a deklarovanou velikost 2048 bytů. Ta bude využita při samotném stahování souboru, kdy se daný soubor bude do prostoru počítače stahovat právě po dvou kilobytových částech.

Jakmile jsou proměnné deklarovány, je volána funkce objektu *folderBrowserDialog*, která velice usnadní výběr složky na disku, kam se soubor uloží. Tato cesta se uloží do proměnné *path*.

Opět bude využito FTP požadavku, tentokrát však bude metoda požadavku *DownloadFile*, která slouží ke stažení libovolného souboru uloženého na internetu. Na požadovaném místě na disku je nyní vytvořen soubor stejného jména, jaké má soubor právě vybraný.

```
FileStream fs = new FileStream(path + "/" + item, FileMode.Create);
```

Díky postupnému čtení a ukládání dvou kilobytových částí se soubor stahuje do prostoru počítače.

```
while (ReadCount > 0)
{
    fs.Write(buffer, 0, ReadCount);
    ReadCount = stream.Read(buffer, 0, buffer.Length);
}
```

Během procesu stahování je v okně programu červeným písmem napsán text upozorňující uživatele na právě probíhající stahování. Po dokončení je opět upozorněn, že je stahování dokončeno. Během procesu stahování jsou veškeré funkce programu nefunkční.

Poslední tlačítko *Smaž soubor z FTP* umožňuje uživateli ze vzdáleného serveru mazat nepotřebná videa, a tím předejít problémům s nedostatkem paměti na serveru nebo velkému a nepřehlednému množství souborů.

Při stisku tohoto tlačítka jsou opět nastaveny vlastnosti webového požadavku a je použita metoda *DeleteFile*, která z daného serveru maže soubor daného jména.

```
request.Method = WebRequestMethods.Ftp.DeleteFile;  
FtpWebResponse response = (FtpWebResponse) request.GetResponse();  
response.Close();  
MessageBox.Show("Soubor " + item + " byl smazán");
```

Po smazání souboru se objeví informační okno, že byl soubor úspěšně smazán. Funkce je také ošetřena proti nekorektnímu zacházení. Pokud není vybrán žádný soubor, tlačítko neumožní komunikaci s FTP serverem.

5 TESTOVÁNÍ BEZPEČNOSTNÍHO SYSTÉMU

Bezpečnostní systém je díky nekrytým elektronickým součástkám navržen především pro vnitřní použití. Nic však nebrání jeho umístění do zastřešených venkovních částí (např. může pod střechou střežit garáž). Testován však byl zejména uvnitř zástavby a jeho chování při použití venku nemusí být stejné.

Při prvních testovacích pokusech docházelo během uploadování videa z neznámého důvodu k neustálému restartování modemu připojeného k Arduino. Po nainstalování nových ovladačů tento problém přetrvával a zabraňoval jakékoliv další práci. Po konzultaci problému s vedoucím diplomové práce byl změřen proudový odběr připojeného zařízení Huawei. Jak je vidět z tabulky 5.1, nejnižší odběr má zařízení, které je fyzicky zapojeno do USB, ale není připojeno do sítě. Nejvyšší odběr je naopak při stahování nebo nahrávání dat. Změřen byl i odběr vývojové desky Arduino, který byl při aktivním Wi-Fi modulu přibližně 350 mA.

Tabulka 5.1: Proudový odběr modemu Huawei

Mód modemu Huawei E398	Proudový odběr
Stand-by	90 mA
Připojování zařízení	270 mA
Zařízení připojeno do sítě	95 mA
Přenos upload	295 mA
Přenos download	300 mA

Je tedy zřejmé, že celkový proud potřebný na napájení bezpečnostního systému musí být přes 650 mA. Takový proud je však pro rozhraní USB 2.0 příliš velký. I při využití stabilních, výkonných 5 V zdrojů se modem při přenosu dat stále restartoval. Jako pomocné napájení byl využit i vstupní pin Vin, který je na desce Arduino. Bohužel ani při zdvojeném napájení nestačil USB port, ve kterém byl zapojen modem, zásobovat zařízení dostatečným proudem. Jako pomocné napájení tedy byl do USB portu Arduina připojen USB hub, do jehož jednoho portu je připojen modem Huawei a do druhého přídavné napájení 5 V (viz obrázek 6.7). Spotřeba modemu tedy nyní není závislá na napětí desky Arduino.

5.1 Zapojení bezpečnostního systému pro domácnost

Zařízení bylo testováno v různých destinacích, aby se otestoval vliv signálu na přenosovou rychlost. Testování proběhlo ve Žďáře nad Sázavou, v Brně - Žabovřeskách a v Laboratoři mobilních komunikací na FEKT VUT v Brně. V době testování ještě nebyl spuštěn systém LTE, takže výsledky ve všech místech byly téměř totožné.

Pro přehlednější sledování délky jednotlivých procesů mikroprocesoru byla informativní LED dioda naprogramována tak, že po každém významném procesu změni svůj stav. Významnými procesy se myslí zejména jednotlivé podprogramy, na které

hlavní program skáče. Například při zapnutí systému se LED dioda rozsvítí, při aktivním signálu z pohybového čidla zhasne, po odeslání SMS zprávy se rozsvítí, po uplynutí definovaného zpoždění opět zhasne. Naposledy se stav mění po dokončeném stažení souboru na SD kartu Arduina a po nahrání videa na FTP server. Lze tedy přehledně a přesně sledovat časové intervaly jednotlivých procesů. Zpočátku byla komunikace sledována přes sériovou linku. Při odpojení zařízení od počítače a napájení externím zdrojem již nebyl tento způsob možný. Časové intervaly naměřené pro 5 MB soubor jsou uvedeny v tabulce 5.2.

Tabulka 5.2: Časy procesů bezpečnostního systému

Akce	Časový interval
Odeslání SMS zprávy	3-4 s
Uložení souboru do Arduina	17 s
Nahrání souboru na server	15 s

Z tabulky lze zjistit, že nejdéle trvá proces stahování na paměťovou kartu. Průměrná přenosová rychlost signálu byla naměřena 300 kB/s. Toto rychlostní omezení je způsobeno omezenou rychlostí kamery, odkud se video soubor stahuje, dále maximální přenosovou rychlostí paměťové karty a v neposlední řadě omezením Arduina, které není navrženo na přenášení souborů. Na oficiálních stránkách Arduina je udáván takt procesoru 400 MHz. Při testování přenosových rychlostí nikdy nebyla překročena přenosová rychlost signálu 500 kB/s. Tato rychlost byla naměřena při přenosu souboru z interního úložiště Arduina na jeho paměťovou kartu. Nebyly tedy zapojeny žádné síťové komponenty.

Samotný přenos souboru na FTP server probíhá v režii linuxového procesoru Atheros, který data odešle přes USB modem. Rychlost je o něco vyšší než při kopírování souboru z kamery, průměrně se pohybuje okolo 350 kB/s.

Při síťových komunikacích je nutné pamatovat, že před zahájením kopírování souboru musí být navázáno spojení se zvoleným serverem. Díky pomalým vzájemným operacím mezi procesory ATmega a Atheros dochází vždy k drobné prodlevě.

5.1.1 Instalace zařízení

Při ověřování činnosti zařízení byl bezpečnostní systém nainstalován tak, aby střežil dveře do místnosti (viz obrázek 5.1). Společně s kamerou Foscam byl záznam ze stejného místa pořizován fotoaparátem Canon EOS 1100D, který má rozlišení obrazu stejné, jako použitá kamera.



Obrázek 5.1: Umístění bezpečnostního systému v místnosti

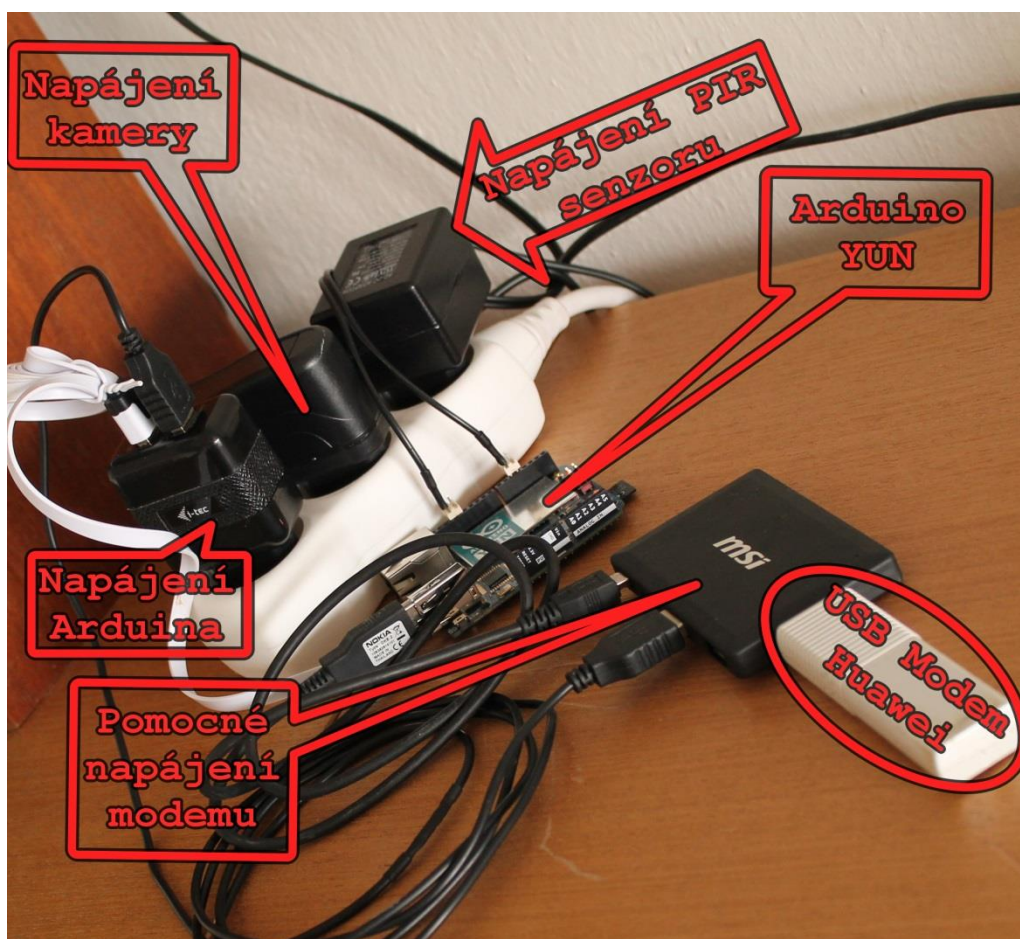
Na obrázku 5.2 je vidět kompletní a zapojený systém. Při použití v praxi by však byla vidět jen kamera a pohybový senzor. Napájecí zásuvka společně s Arduinem a modemem by byly bezpečně schovány v krabici.



Obrázek 5.2: Zapojený bezpečnostní systém

Výhodou bezdrátové komunikace kamery s Arduinem je, že kamera nemusí být ve stejném místě jako pohybový senzor, jak obvykle bývá zvykem. Dle příslušného prostoru, který má být střežen, se kamera může umístit kamkoliv v dosahu Wi-Fi signálu. Vysílací výkon Wi-Fi modulu Arduina není velký, ale při přímé viditelnosti lze dosáhnout dostatečného signálu i při vzdálenosti do 8 metrů. Umístění kamery tedy není omezeno délkou ethernetového kabelu, a proto si uživatel může kameru nainstalovat tam, odkud bude mít nejlepší přehled o dění v místnosti.

Na obrázku 5.3 jsou popsány jednotlivé periferie, které jsou normálně veřejnosti skryté. Součástí jsou tři napájecí zdroje. Stejnoseměrný 12 V zdroj pro napájení pohybového senzoru a dva 5 V zdroje. V případě potřeby by mohly být napájecí zdroje pro kameru a Arduino sloučeny v jeden. Opět by se tím však riskovalo nedostačující proud pro modem, a tím způsobené jeho restartování. Aby se předešlo takovým nechtěným stavům, je napájení pro obě zařízení separované. Pro napájení Arduina a modemu byl zakoupen zdroj se dvěma napájecími USB konektory a maximálním proudovým zatížením do 2,1 A.



Obrázek 5.3: Podrobný popis ukrytých periferií

5.1.2 Kvalita videa

Kvalita výstupního videa je jeden z nejdůležitějších parametrů bezpečnostního systému. Pokud by systém využíval k pořizování záznamu kameru s nízkým rozlišením, mohlo by dojít k situacím, kdy by osoba zachycená na videozáznamu byla rozmazaná a nedala by se identifikovat. Rozlišení obrazu je velmi důležitý parametr, avšak není jediný. Pokud má kamera vysoké rozlišení, ale nízkou snímkovou frekvenci, malou citlivost snímače nebo vysoký kompresní poměr, kde se zanedbávají vysoké obrazové frekvence a tedy detaily v obraze, video je nekvalitní a kamera není vhodná pro bezpečnostní účely.

Kamera Foscam FI9821P zakoupená pro účely diplomové práce je vhodná pro střežení prostor v domácnosti, kde by se případný nezvaný host mohl pohybovat v blízké vzdálenosti od kamery. Tato kamera i přes HD rozlišení není vhodná pro nahrávání ve velkých výrobních halách, kde je nutné získat vysoké detaily i na větší vzdálenosti. Pro takové situace jsou vyráběny kamery s velmi vysokým rozlišením, výbornou citlivostí a možností přiblížení pomocí optických výměnných objektivů. Jedna z takových kamer firmy AirCam je na obrázku 2.6 b). Kamery umožňující zobrazení detailů obrazu jsou používány pro sledování železničních stanic, nemocnic nebo soukromých areálů. Cena takových kamer je vysoká, proto byla pro diplomovou práci zakoupena kamera střední třídy.

Na následujících obrázcích jsou vidět rozdíly mezi obrazem pořízeným kamerou Foscam a obrazem pořízeným fotoaparátem Canon EOS 1100D, přičemž obě zařízení mají stejné rozlišení 1280 x 720 bodů i stejný standard pro kompresi, tedy H.264. Na obrázku 5.4 je první snímek videa odeslaného na server. Po otevření dveří bylo aktivováno pohybové čidlo a zahájeno nahrávání. Na obrázku 5.5 je stejný moment, ale ze záznamového zařízení Canon. Nepřesně nastavená ohnisková vzdálenost objektivu fotoaparátu způsobuje, že zachycený obraz není na obou obrázcích úplně stejný.

Velikost obou souborů se velmi výrazně liší. Video trvající 10 sekund má v případě záznamu z kamery Foscam 1,93 MB, zařízení Canon pořídilo stejně dlouhé video o velikosti 30,055 MB. Z velikosti souboru a délky videa byla vypočítána přenosová rychlost pro tuto videosekvenci.

$$\text{Přenosová rychlost}_{\text{Canon}} = 30,055 * 8 / 10 = 24,044 \text{ Mb/s}$$

Výsledky jsou vidět společně s rychlostí pro vysoký kompresní poměr v tabulce 5.3. Jelikož je obraz téměř statický a jsou přenášeny pouze rozdíly v obraze, je v tomto případě úspora dat velká. Rozdíly přenosových rychlostí zařízení jsou způsobeny rozdílným zpracováním videa v kodéru, kódováním zvuku atd. I přesto je přenosová rychlost kamery Foscam neobvykle nízká. Během testování nebyl kamerou nikdy pořízen dokonale ostrý záběr, je tedy možné, že k úspoře dat dochází na úkor kvality. Zřejmě jsou z videa odstraněny i frekvence, které tvoří detaily a již jsou okem viditelné.

Tabulka 5.3: Přenosové rychlosti různých zařízení

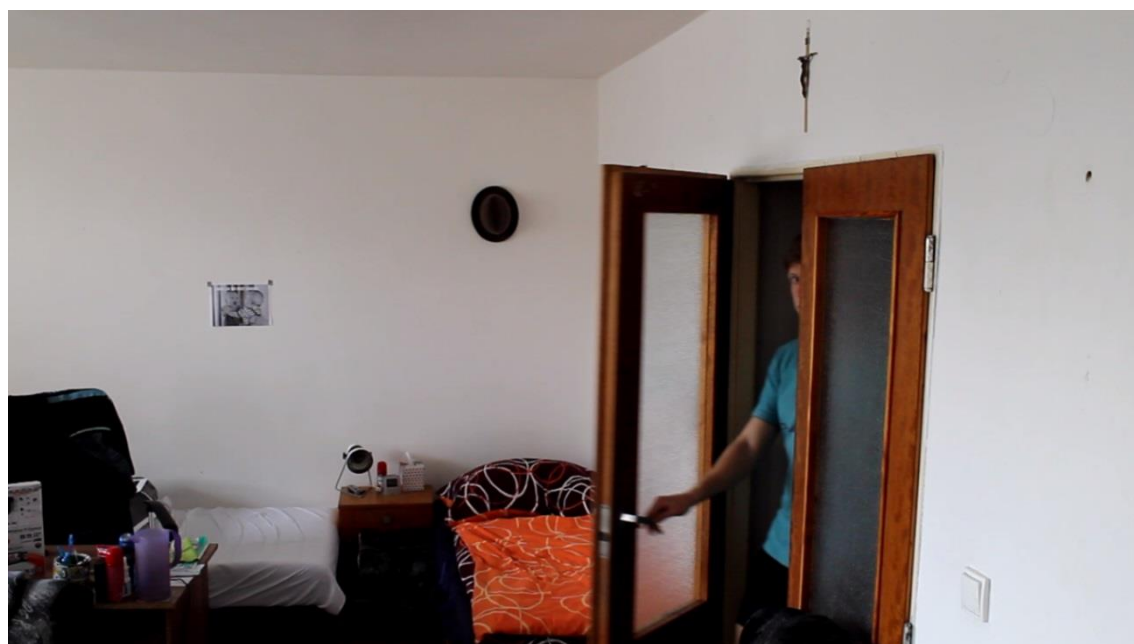
Typ zařízení	Přenosová rychlost
Canon EOS 1100D	24,04 Mb/s
Foscam FI9821P	1,54 Mb/s
Vypočtený standard pro kompresní poměr 320:1	1,72 Mb/s

Již na prvním snímku je vidět rozdíl v kvalitě záznamu. Nejcitelnější je rozdíl v ostrosti. Při použití kamery Foscam jsou neostře hrany objektů, barevné podání a dynamický rozsah jsou také horší. Pro základní funkci diplomové práce je i přesto obraz naprosto dostačující.

Do videa odesílaného na server je do levého horního rohu automaticky vkládána informace o datu a času záznamu.



Obrázek 5.4: Testování bezpečnostního systému - Foscam

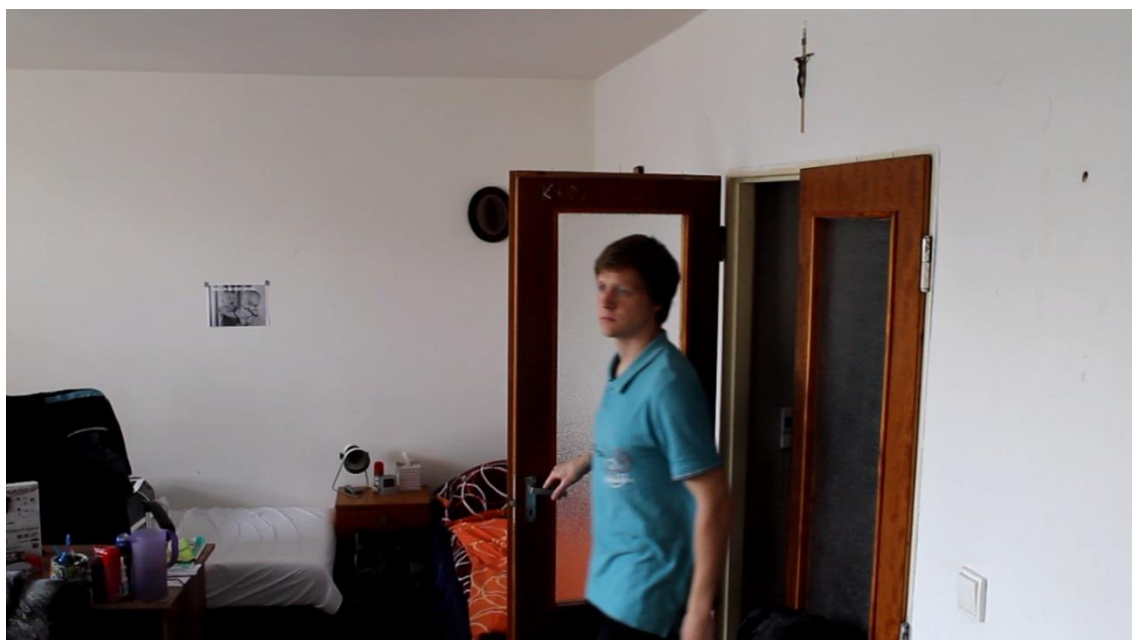


Obrázek 5.5: Testování bezpečnostního systému – Canon

Další porovnávací snímky jsou vidět na obrázcích 5.6 a 5.7. V tento moment je osoba v pohybu a je vidět, že ani video pořízené fotoaparátem Canon nezachytilo pohybující se osobu dokonale ostrou. K identifikaci pachatele by však byl v obou případech záznam dostačující.



Obrázek 5.6: Testování bezpečnostního systému – Foscam



Obrázek 5.7: Testování bezpečnostního systému – Canon

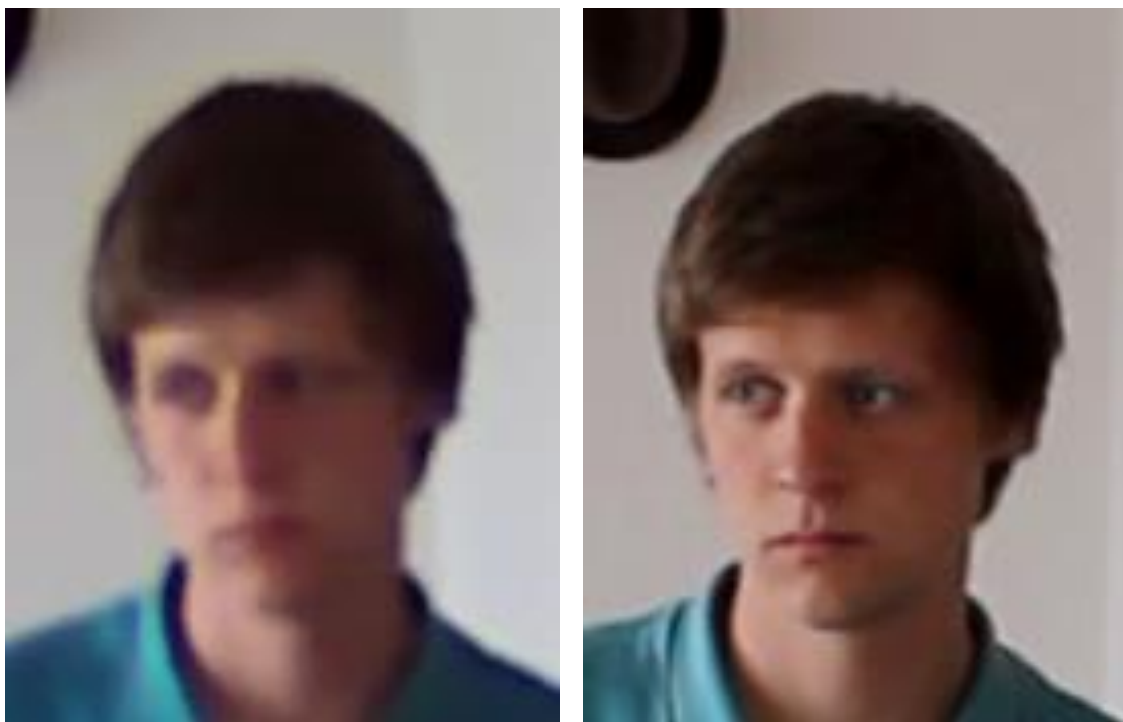
Na obrázcích 5.8 a 5.9 jsou naopak záběry z momentu, kdy se osoba téměř nepohybovala. V tomto případě je vidět znatelný rozdíl mezi kvalitním záznamovým zařízením Canon a relativně levnou kamerou pro použití v domácnosti. V případě kamery Foscam by obrázek sice byl dostačující pro identifikaci, ale detaily v tváři vidět nejsou. Na obrázku 5.10 je výřez tváře z těchto snímků a je vidět, že u snímku pořízeného fotoaparátem Canon by byly vidět i detailnější rysy obličeje, barva očí, náušnice, piercing a další informace potřebné k bezpečnému rozpoznání pachatele.



Obrázek 5.8: Testování bezpečnostního systému – Foscam



Obrázek 5.9: Testování bezpečnostního systému – Canon



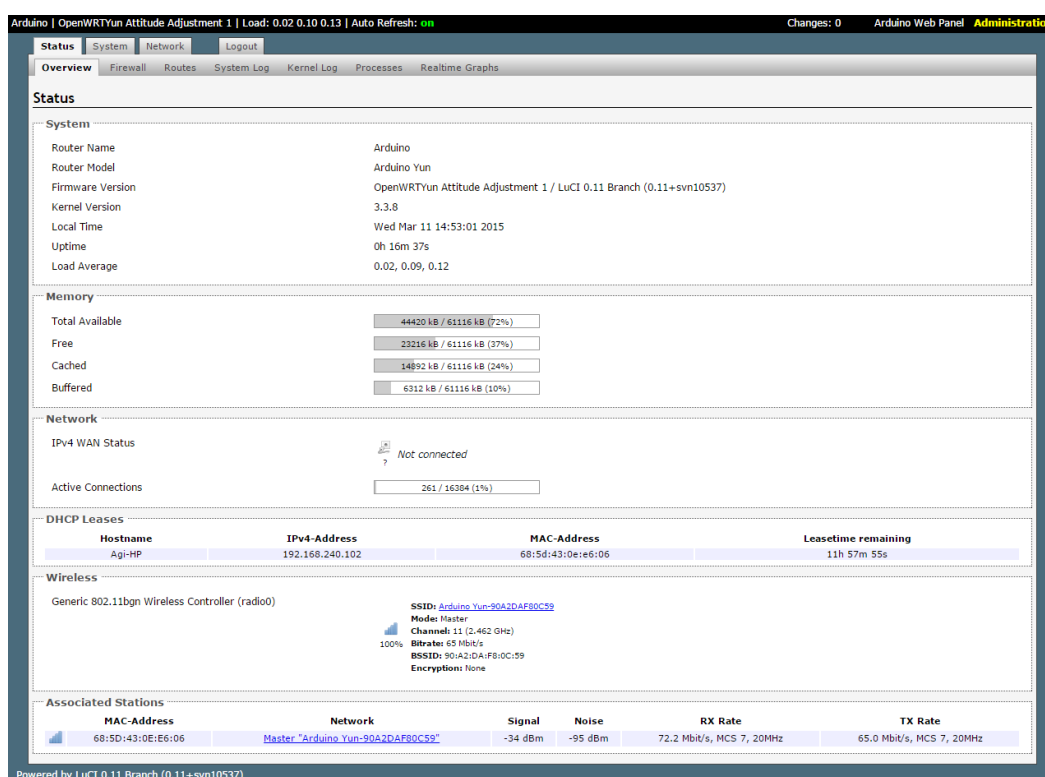
Obrázek 5.10: Detail tváře ze zařízení a) Foscam b) Canon

Během testování bylo zjištěno, že koupí bezpečnostní kamery s vysokým rozlišením není zajištěn obraz, jaký by si uživatel pod tímto pojmem představoval. Pro oživení a otestování bezpečnostního systému s využitím mobilních sítí však byla tato kamera plně dostačující.

6 NÁVOD NA INSTALACI SYSTÉMU

Do desky Arduino YUN zapojíme mikro USB konektor kabelu, jehož druhý konec je připojen k počítači, ve kterém máme k dispozici zdrojové kódy. Rozsvítí se LED dioda informující uživatele o tom, že je deska napájena. Po chvíli by se deska měla objevit v seznamu dostupných Wi-Fi zařízení připojeného počítače. Po připojení se k Wi-Fi *ArduinoYun-90A2DAF80C59* otevřeme ve webovém prohlížeči internetovou adresu 192.168.240.1, kde se objeví jednoduchá linuxová distribuce OpenWRT. Zadáním hesla “arduino” se dostaneme na hlavní obrazovku. Kliknutím na *Configure* a následně na *advanced configuration panel (luci)* se dostaneme do prostředí OpenWRTYun (viz obrázek 6.1).

V záložce *System* nastavíme časové pásmo pro Českou Republiku.



Obrázek 6.1: Úvodní stránka operačního systému v Arduino YUN

6.1 Návod pro nastavení Arduina YUN

6.1.1 Rozšíření úložného prostoru

Pomocí vývojového prostředí Arduino IDE, vybereme příslušný COM port a nahrajeme do procesoru program *ExpandingYunDiskSpace*, který je volně dostupný na stránkách arduino.cc. Pomocí instrukcí ze sériové linky lze rozdělit vloženou microSD kartu na libovolně velká úložiště. V našem případě je použita karta s kapacitou 2GB.

6.1.2 Instalace ovladačů 3g modemu

Jelikož je OpenWRT “holý” operační systém, musí se všechny potřebné ovladače stáhnout a doinstalovat. K tomu je však nutné, aby Arduino bylo připojeno k internetu. Připojíme ho tedy ethernetovým kabelem do domácí internetové sítě.

V záložce *System - Software* je seznam dostupných a již nainstalovaných ovladačů. K tomu, abychom mohli pracovat s vysokorychlostním USB portem a k němu připojeným 3g modemem, je potřeba doinstalovat následující ovladače:

- kmod-usb-core
- kmod-usb-serial
- kmod-usb-serial-option
- kmod-usb2
- usb-modeswitch
- usb-modeswitch-data
- luci-proto-3g
- ppp
- comgt
- chat

Po nainstalování potřebných ovladačů je vyžadován restart desky Arduino, který provedeme dočasným odpojením napájení.

6.1.3 Konfigurace 3G připojení

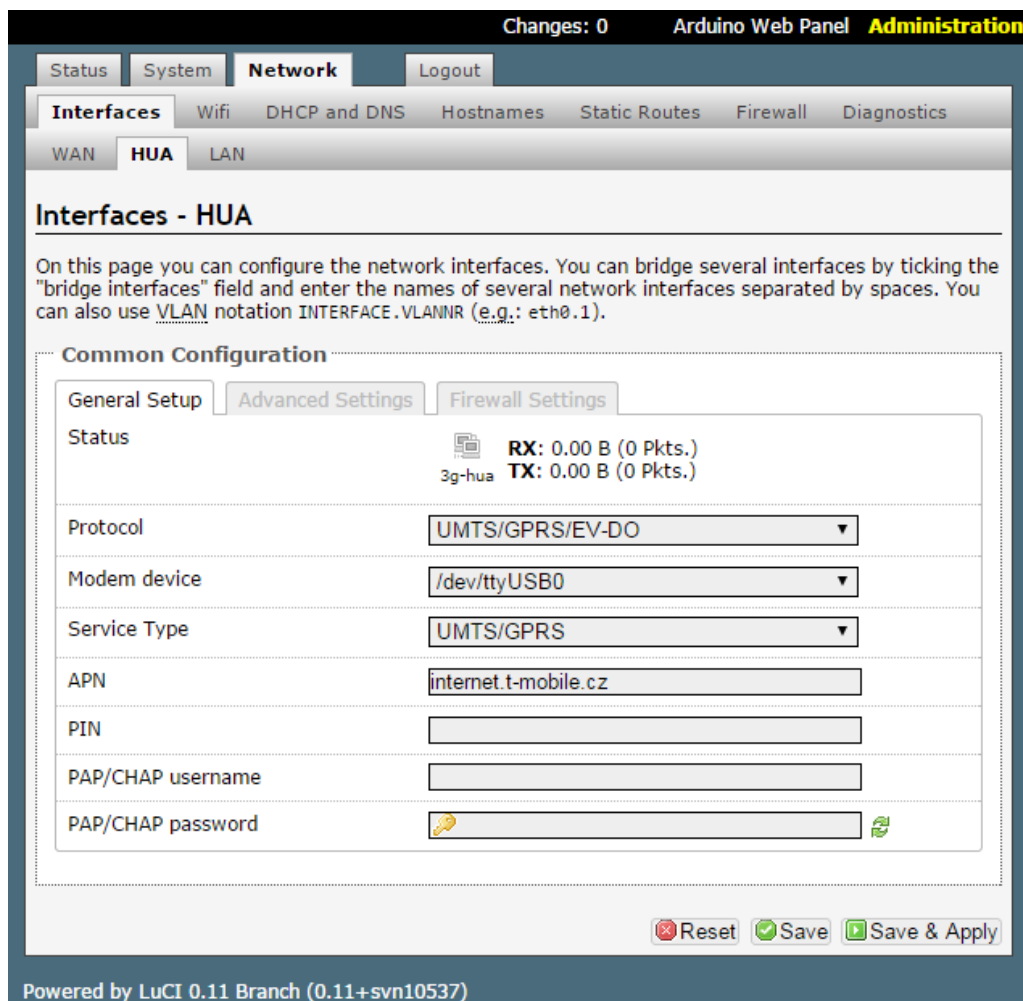
Deska Arduino je defaultně nakonfigurována tak, že připojení k internetu je možné pouze přes ethernetové rozhraní nebo přes bezdrátovou síť. V našem případě však potřebujeme přidat rozhraní, které bude komunikovat přes USB port s modemem.

Po připojení donglu do portu Arduina si v záložce *Status - System Log* ověříme, že bylo zařízení skutečně rozpoznáno jako GSM modem. Vypsát by se mělo toto hlášení:

```
option 1-1.2:1.0: GSM modem (1-port) converter detected
usb 1-1.2: GSM modem (1-port) converter now attached to ttyUSB0
option 1-1.2:1.1: GSM modem (1-port) converter detected
usb 1-1.2: GSM modem (1-port) converter now attached to ttyUSB1
option 1-1.2:1.2: GSM modem (1-port) converter detected
usb 1-1.2: GSM modem (1-port) converter now attached to ttyUSB2
```

Pokud je modem korektně rozpoznán, v záložce *Network - Interfaces* klikneme na tlačítko *Add new interface...* . Po vyplnění názvu rozhraní vybereme protokol *UMTS/GPRS/EV-DO* a klikneme na tlačítko *Submit*. Otevře se stránka s konfigurací, kterou vyplníme podle obrázku 6.2. Pokud je SIM karta v modemu zabezpečena PIN kódem, vyplníme i pole pro tento kód. Dále přepneme do záložky *Firewall Settings*, kde vybereme možnost *WAN*. Po uložení a aktivování změn by se měl modem automaticky připojit k síti pomocí sekvence AT příkazů, které můžeme vidět ve výpise. Jestliže je připojení úspěšné, LED dioda na modemu se rozsvítí modře.

Jakmile je funkční připojení přes mobilní síť, můžeme odpojit ethernetový kabel.



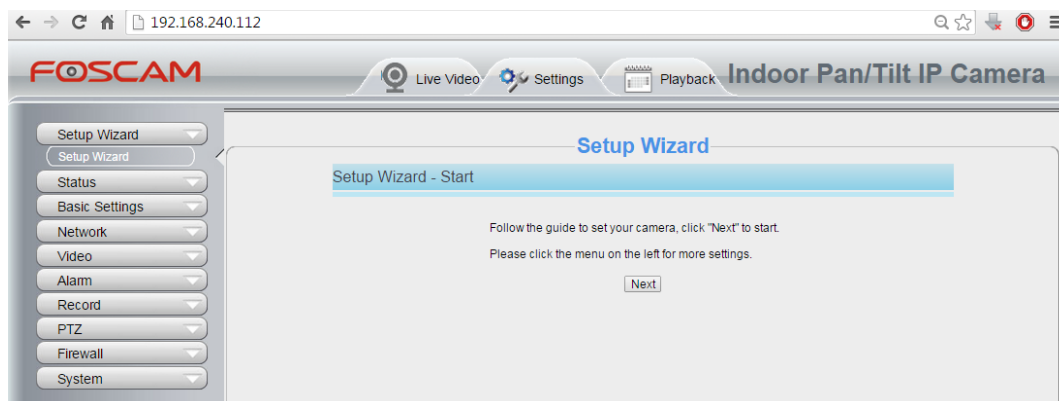
Obrázek 6.2: Konfigurace nového rozhraní

6.1.4 Konfigurace IP kamery

Ke kameře připojíme externí anténu, vložíme SD kartu, připojíme napájecí kabel a kameru připojíme k internetu pomocí ethernetového kabelu. Pomocí programu *IP Camera Search Tool* přiloženého na CD u kamery zjistíme její IP adresu a pomocí webového prohlížeče se k ní připojíme. Otevře se přihlašovací stránka, kde vyplníme jméno a heslo pro přístup ke kameře. Jméno i heslo jsou defaultně nastaveny „adasvobi“. Prostředí kamer Foscam, které je vidět na obrázku 6.3, obsahuje tři záložky, z nichž využijeme pouze záložku *Settings*.

V nabídce *Record -> Storage Location* nastavíme defaultní úložiště na SD kartu, dále v *Network - Wireless Settings* vyhledáme dostupné Wi-Fi zařízení a připojíme se k síti s názvem *Arduino Yun-90A2DAF80C59* a nakonec v nabídce *PTZ – Start-Up Options* vybereme *Disable Start-Up*, což po restartu kamery zabrání tomu, aby se nasměrovala do výchozí pozice.

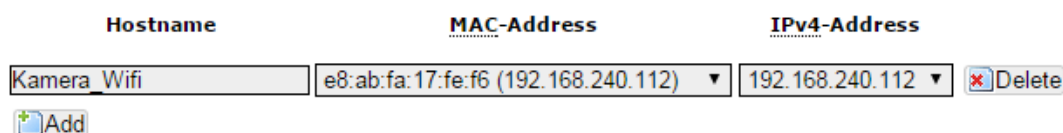
Po uložení nastavení odpojíme od kamery ethernetový kabel a restartujeme ji odpojením od napájecího napětí a opětovným připojením. Po restartování by již měla být kamera připojena k bezdrátové síti desky Arduino, což můžeme ověřit v systémovém výpisu desky Arduino YUN.



Obrázek 6.3: Prostředí kamery Foscam

Jelikož je kamera přiřazována IP adresa pomocí DHCP, je možné, že se tato IP adresa může měnit, což je v tomto případě nežádoucí. Fixní adresu můžeme přiřadit v záložce *Network – DHCP and DNS*, kde ve spodní části stránky vyplníme název, MAC adresu a IP adresu (viz obrázek 6.4).

Tím je dokončeno síťové nastavení Arduina a kamery.



Obrázek 6.4: Nastavení fixní IP adresy

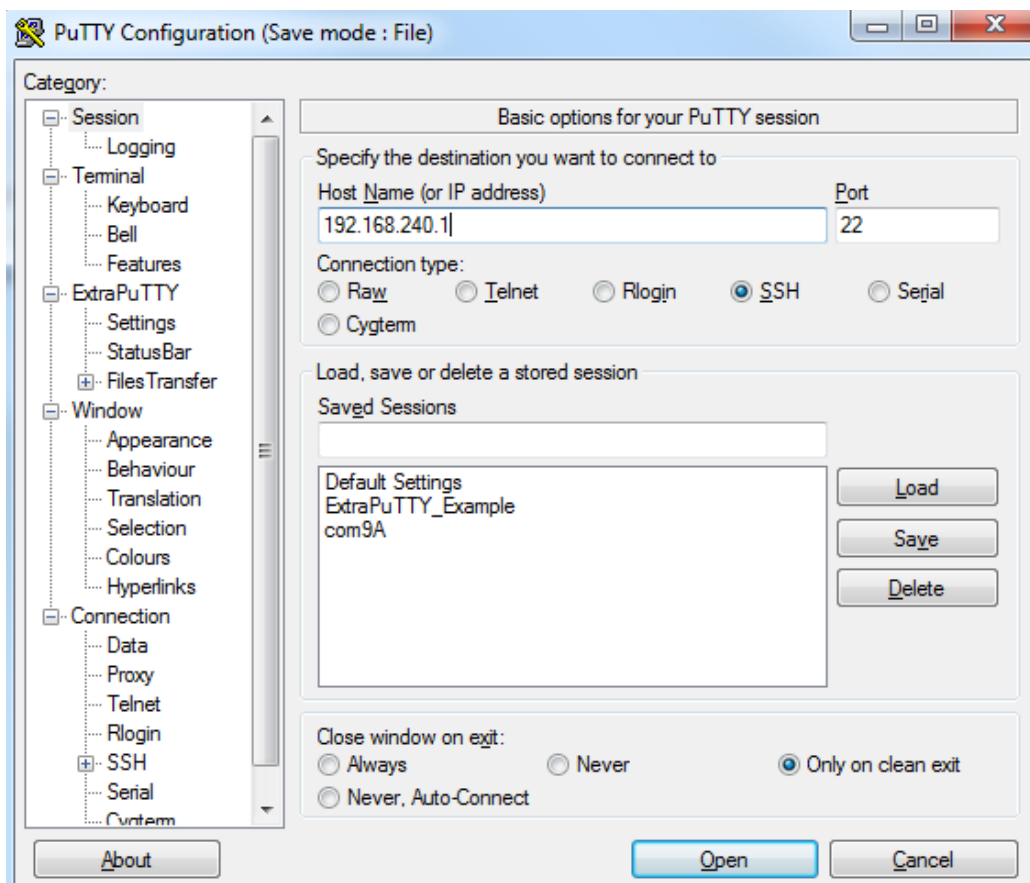
6.1.5 Nahrání programu do Arduina

Aby mohl bezpečnostní systém pracovat, je potřeba do mikroprocesoru nahrát program, který bude celý systém řídit. V prostředí Arduino IDE vybereme příslušný COM port, ke kterému je Arduino připojeno a nahrajeme přiložený program *SecuritySystem.ino*.

Jelikož je v tomto programu funkce, ve které se volá soubor uložený na SD kartě, musíme nejprve tento soubor na SD kartu stáhnout. Jedním ze způsobů, jak nahrát soubor na paměťovou kartu Arduina, aniž bychom kartu vyjmuli, je pomocí příkazu *wget*. Přiložený soubor *sms.py* je uložený také na FTP serveru, odkud ho stáhneme.

Použijeme zabezpečený komunikační protokol, který se používá v TCP/IP – tzv. SSH (Secure Shell). Jedním z programů, kde je možnost pracovat s tímto protokolem, je PuTTY, kde vybereme možnost SSH a zadáme IP adresu Arduina – defaultně 192.168.240.1 (viz obrázek 6.5). Odsouhlasením tohoto okna se otevře příkazový řádek, tzv. Bash, kam se přihlásíme uživatelským jménem „root“ a heslem „arduino“. Dále už je možné pracovat s klasickými linuxovými příkazy. Následující sekvencí příkazů uložíme soubor do požadovaného adresáře na paměťové kartě:

```
cd /mnt/sd1/arduino/www
wget ftp://student:urel@ms2.urel.feec.vutbr.cz/Diplomova_prace/Svoboda/sms.py
```



Obrázek 6.5: PuTTY – konfigurace SSH

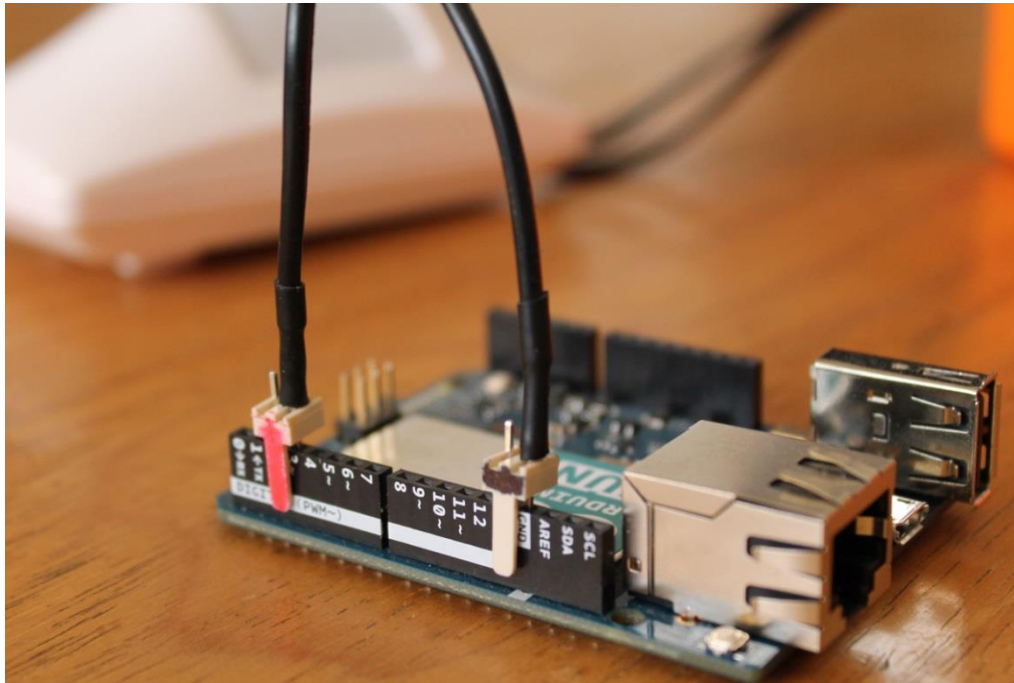
6.1.6 Sestavení systému

Nyní jsou mikroprocesor i periférie připraveny a je možné celý systém sestavit dohromady.

V desce Arduino je nakonfigurován pin č. 3 pro připojení pohybového čidla. Podle obrázku 6.6 připojíme zemnicí konektor na GND a plusový (červený) konektor k pinu číslo 3.

USB port na Arduino je dimenzován spíše na napájení flash disků a podobných zařízení s nízkým odběrem proudu, je potřeba připojit pomocné napájení (viz obrázek 6.7) pro USB modem, který má v aktivním režimu odběr podstatně vyšší. Použijeme k tomu USB hub.

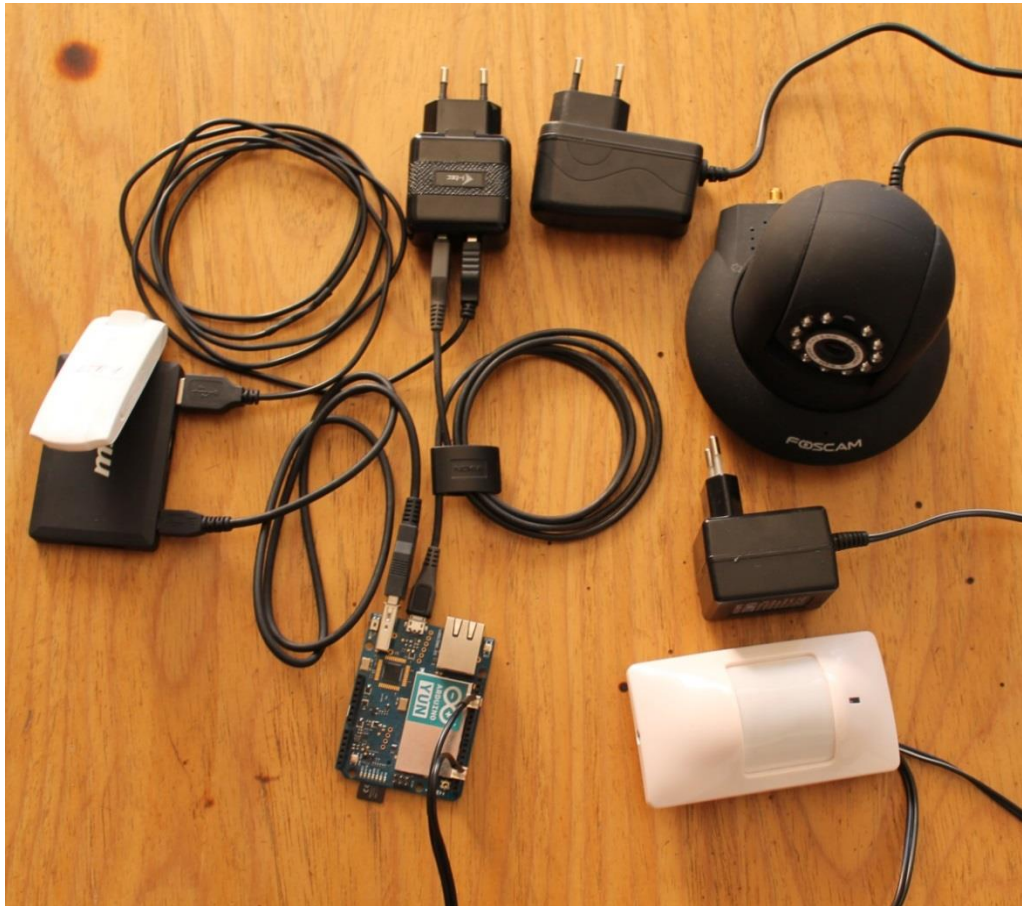
Jakmile je celý systém zapojen podle obrázku 6.8, můžeme připojit napájení pohybového senzoru. Pokud je v obslužném programu *FtpApp* povoleno aktivovat alarm, tak po minutové kalibraci čidla je alarm aktivován a je zahájeno nahrávání videosekvence, odeslána SMS zpráva a video je uploadováno na FTP server. Po kalibraci čidla již systém funguje korektně.



Obrázek 6.6: Zapojení pohybového senzoru



Obrázek 6.7: Pomocné napájení USB modemu



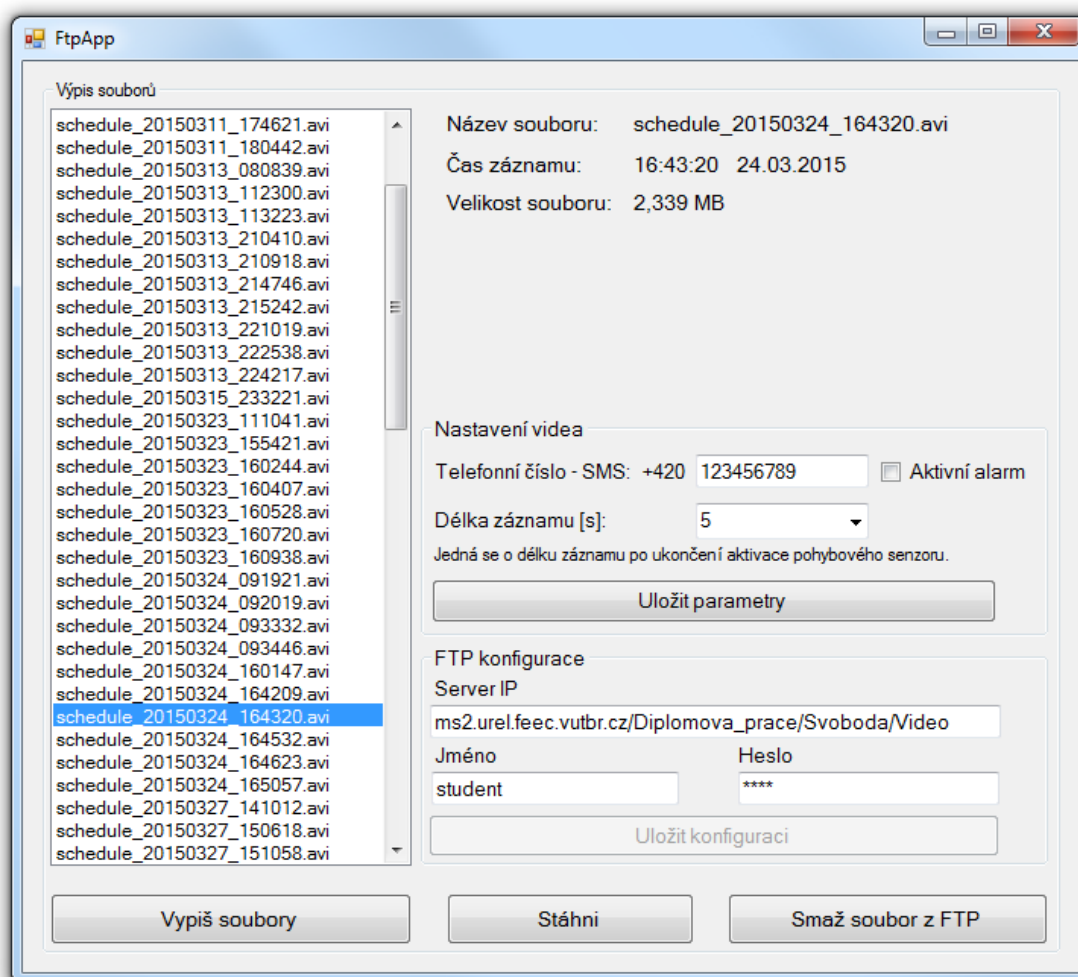
Obrázek 6.8: Kompletní bezpečnostní systém

6.2 Návod na ovládání obslužného programu FtpApp

Pro jednoduchou komunikaci uživatele bezpečnostního zařízení s FTP serverem byl vytvořen program *FtpApp*. Pomocí grafického rozhraní může uživatel nastavit některé parametry systému, případně provádět základní operace s nahranými soubory. Na obrázku 6.9 je vidět základní okno rozdělené do několika částí.

V sekci *FTP konfigurace* je prostor pro vyplnění informací o FTP serveru, na který se soubory videa odesílají. Po vypsání IP adresy serveru, uživatelského jména a hesla se aktivuje tlačítko *Uložit konfiguraci*, které uloží potřebné informace pro připojení k serveru. Pokud nedojde ke změnám v přihlašovacích údajích, zůstává tlačítko neaktivní.

Stiskem tlačítka *Vypiš soubory* jsou do levé části okna vypsány názvy všech souborů, které byly během alarmu vytvořeny. Názvy souborů jsou ve formátu *schedule_RRRRMMDD_HHMMSS.avi*. Po kliknutí na jeden ze souborů se v pravé horní části vypíšou základní informace o souboru daného videa. Zobrazí se název videa, přehledný čas a datum záznamu a velikost souboru (video o velikosti 4 MB odpovídá zhruba 30 s videa bez větších změn v obraze).



Obrázek 6.9: Aplikace pro přehled videí

Po stisku tlačítka *Stáhni* se zobrazí okno pro výběr cílové složky, kam se video uloží. Během stahování videa je v okně zpráva o stavu stahování, a pokud není stahování dokončeno, jsou pozastaveny všechny další procesy programu.

Tlačítkem *Smaž soubor z FTP* se z FTP serveru nenávratně smaže označený video soubor.

Důležitou sekci okna je část *Nastavení videa*. Zde může uživatel nastavit telefonní číslo, kam budou odesílány SMS zprávy informující o narušení prostoru, dále délku záznamu, kterou bude kamera nahrávat po ukončení pohybu v monitorovaném prostoru a pole pro volbu, jestli má být objekt střežen nebo pohyb v daném místě nevádí a není třeba prostor sledovat. V takovém případě zůstane pole *Aktivní alarm* nezaškrtnuté (viz obrázek 6.9).

6.3 Návod na vytvoření FTP serveru

Ke správné funkci bezpečnostního systému je bezpodmínečně nutný přístup k FTP serveru, na který se budou nahraná videa odesílat a odkud pak bude možnost videa stahovat a prohlížet. Pokud není k dispozici nějaký komerční server nebo nemáme v počítači takové úložiště připravené, je nutné si jej vytvořit pomocí programů, které jsou k tomu navrženy.

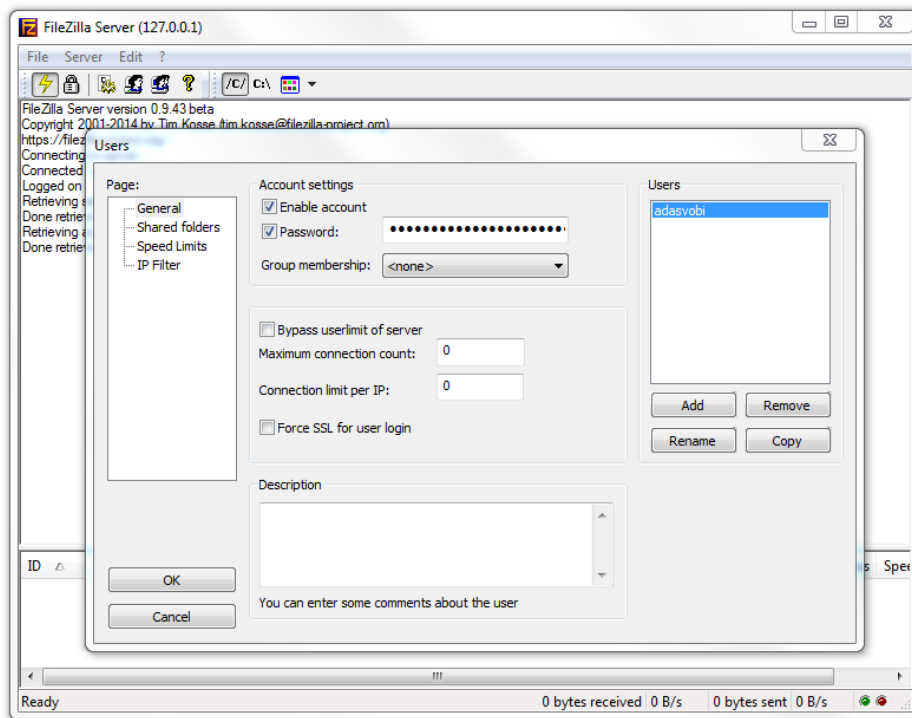
Jedním z nejlepších a zároveň zdarma dostupných programů je *FileZilla Server*. Za opensourcovou licenci se skrývá malý, jednoduchý, ale přitom vyspělý nástroj, který je k dispozici přímo na stránkách www.filezilla-project.org. Umožňuje na počítači s veřejnou IP adresou vytvořit úložiště, které je následně dostupné odkudkoliv na světě, kde je k dispozici připojení k internetu.

K dispozici jsou i spousty jiných programů se stejnou funkcí, avšak ve většině případů již nenabízí své služby zdarma.

Jak je vidět na obrázku 6.10, jedná se o okno s výpisy událostí serveru. Aby bylo možno server na počítači využívat, je nutné nastavit pro konkrétní IP adresu počítače seznam možných uživatelů, kteří budou mít ke sdílené složce přístup. Dále je třeba vybrat složku nacházející se v počítači, která bude v síti sdílená, a právě do ní se budou ukládat videa pořízená bezpečnostním systémem. Pro jednotlivé uživatele je možnost nastavení uživatelských práv. Na výběr je možnost, kde uživatel může pouze prohlížet soubory dostupné na serveru, a druhá možnost, kdy uživatel může provádět i zápis do sdílené složky. Jelikož je videa potřeba na server ukládat, je nutné vybrat možnost *read/write*. Dále jsou zde k dispozici možnosti nastavení rychlosti datových přenosů a seznam IP adres, kterým je připojení k FTP serveru zamítnuto.

Jakmile jsou práva nastavená, stačí v levém horním rohu stisknout tlačítko blesku, a tím přepnout server do online režimu. Nyní by měl být počítač dostupný na internetu jako zabezpečený FTP server.

Detailní návod k používání tohoto programu je taktéž k dispozici na výše uvedených stránkách.



Obrázek 6.10: Program FileZilla Server

7 ZÁVĚR

Cílem diplomové práce bylo prostudovat možnosti přenosu obrazových signálů v celoplošných mobilních systémech GSM, UMTS a LTE, dále formáty výstupních obrazových signálů z web kamer a typy čidel pro indikaci pohybu. Na základě získaných poznatků navrhnout blokové schéma systému pro monitorování narušení prostoru se zaměřením na přenos videosekvence, případně i doprovod zvukového signálu. Navržený systém realizovat a jeho funkčnost otestovat. Po informaci z pohybového čidla mají být do mobilního zařízení zasílány krátké videosekvence z kamery v nastavitelných časových intervalech.

V první kapitole jsou rozebrány jednotlivé typy pohybových senzorů, přičemž nejdůležitější jsou rozebírány senzory typu PIR, které jsou díky spolehlivosti a relativně nízké ceně vhodné pro využití v diplomové práci. Z dostupných typů byl jako nejlepší vybrán typ LS-818-3, který se jevil poměrem cena/výkon jako nejlepší. Druhá kapitola se věnuje vlastnostem kamer. Pro bezpečnostní účely jsou už několik let používány zejména kamery s ethernetovým připojením neboli IP kamery. Opět bylo vybráno pět zástupců, od nejlevnějších až po profesionální s velkým rozlišením. Jako vhodná kamera pro diplomovou práci byla vybrána megapixelová kamera firmy FOSCAM FI9821W, která vyčnívá svými možnostmi a nízkou cenou. V další kapitole jsou rozebírány možnosti přenosu dat přes mobilní síť a podrobněji jsou popsány systémy UMTS a LTE.

V kapitole *Návrh zapojení bezpečnostního systému* je podrobně rozepsán obsah vývojové desky Arduino YUN, která je řídicím členem celého systému. Po jednotlivých částech je popisován způsob komunikace s připojenými zařízeními. Místo připojení kamery přes ethernetový kabel bylo zvoleno pohodlnější připojení přes bezdrátovou síť Wi-Fi. To umožňuje připojení dalších bezpečnostních kamer, a tím zabezpečení několika místností současně.

Další kapitola, věnovaná testování bezpečnostního systému, popisuje jednotlivé procesy, které právě probíhají. Během testování bylo objeveno několik problémů. Nejkritičtější problém nastal ve chvíli, kdy mikroprocesor spustil proces odesílání dat na vzdálený server, neboť v tu chvíli docházelo k restartování připojeného modemu a patřičné video zůstalo neodesláno. Tento problém byl způsoben nedostatečným proudem na USB portu desky Arduino, proto bylo zapojení doplněno o pomocný napájecí prvek, který se stará o napájení výkonného USB modemu Huawei E398. Další z problémů nastal při odesílání informační SMS zprávy. Způsob odesílání zprávy z USB modemu byl zpracován různými způsoby, ale díky nevyhovující verzi Linuxu v desce Arduino bylo nutné vytvořit samostatný program uložený na paměťové kartě Arduina. Program psaný v jazyce Python je volán pouze ve chvíli, kdy má být SMS zpráva odeslána. V další části kapitoly určené testování je porovnáváno video z kamery Focsam a video pořízené digitálním fotoaparátem Canon EOS 1100D. Rozdíly ve videu jsou značné. Kamera Focsam použitá v diplomové práci má na první pohled horší ostrost a podání barev. Na jednotlivých snímcích pořízených ze stejného místa je přehledné srovnání těchto dvou záznamových zařízení.

Součástí diplomové práce bylo i vytvořit program, ve kterém budou přehledně zobrazená videa pořízená bezpečnostním systémem. Program *FtpApp* byl napsán jazykem C# a umožňuje přehlednou práci se soubory na FTP serveru. Program byl

taktéž testován na více počítačích. Při prvním spuštění dochází k několikasekundové prodlevě, během které dochází ke skenování programu antivirem. Po pár vteřinách již program pracuje bez problémů.

V poslední části je podrobně popsán návod na instalaci bezpečnostního systému, na programování jednotlivých procesorů, konfiguraci IP kamery a USB modemu. Na fotografiích je přehledně znázorněno zapojení celého systému. Dále je popsána práce s pomocným programem *FtpApp* a návod na vytvoření vlastního FTP serveru, na který lze následně videa odesílat.

V této práci je možno pokračovat především připojením více IP kamer a pohybových senzorů, a tím vytvořit komplexní bezpečnostní systém, který bude umožňovat střežení několika prostor současně.

LITERATURA

- [1] Astronomický ústav AV ČR. [online]. [cit. 2014-11-23]. Dostupné z: http://www.asu.cas.cz/~bezdek/fyzika/vlnDelka_frekvence_teplota.php
- [2] Fyzikální východiska pro konstrukci PIR detektorů. Zlín, 2011. Bakalářská práce. Univerzita Tomáše Bati ve Zlíně. Vedoucí práce Kyas Kamil.
- [3] KADLEC. Měřicí technika: Měření teploty [online]. c2008 [cit. 2010-03-20]. Dostupné z WWW: <http://web.vscht.cz/kadleck/aktual/MT_INaRP/MT-INFOaRP09_Teplota.pdf>.
- [4] GLOLAB: Electronic Kits and Molules. [online]. [cit. 2014-11-23]. Dostupné z: <http://www.globlab.com/pirparts/infrared.html>
- [5] Hw.cz: PIR detektor. [online]. [cit. 2014-11-23]. Dostupné z: <http://www.hw.cz/automatizace/pir-cidlo-skvely-sluha-ale-zly-pan.html>
- [6] Davis, A., Bush, R. C., Harvey, J. C., & Foley, M. F. (2001, June). P-95: Fresnel Lenses in Rear Projection Displays. In SID Symposium Digest of Technical Papers (Vol. 32, No. 1, pp. 934-937). Blackwell Publishing Ltd.
- [7] Home Security Guru: Motion Sensor Design. [online]. [cit. 2014-11-23]. Dostupné z: <http://www.homesecurityguru.com/microwave-motion-sensors>
- [8] Kamery AirLive. [online]. [cit. 2014-11-09]. Dostupné z: http://www.airlivecam.eu/ip_kamera_Airlive_POE-5010HD
- [9] FOSCAM IP Camera. [online]. [cit. 2014-11-09]. Dostupné z: <http://foscam.us/products/foscam-fi9821w-megapixel-wireless-ip-camera.html>
- [10] DIGIMANIE. [online]. [cit. 2014-11-11]. Dostupné z: <http://www.digimanie.cz/bsi-cmos-snimac-popis-technologie/3650-2>
- [11] RHODES, H., et al. The mass production of BSI CMOS image sensors. In: International Image Sensor Workshop. 2009. p. 27-32.
- [12] FOTOGRAFOVANI. [online]. [cit. 2014-11-12]. Dostupné z: <http://www.fotografovani.cz/vybirame/technologie1/obrazove-problemy-digitalni-fotografie-i-senzor-152899cz>
- [13] Netcam.cz. [online]. [cit. 2014-11-12]. Dostupné z: <http://www.netcam.cz/encyklopedie-ip-zabezpeceni/co-je-sitova-kamera.php>
- [14] IP Video Surveillance Design Guide. Cisco Systems, Inc [online]. [cit. 2014-11-19]. Dostupné z: http://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Video/IPVS/IPVS_DG/IPVS-DesignGuide/IPVSSchap4.html#wp1054066
- [15] TAYLOR, Jim, Mark R JOHNSON a Charles G CRAWFORD. Velký průvodce DVD: jedinečný zdroj všech dostupných informací o DVD na profesionální úrovni. 1. vyd. Praha: Grada, 2007, 551 s. Profesionál. ISBN 978-80-247-1721-0.
- [16] Wiegand, Thomas, et al. "Overview of the H. 264/AVC video coding standard." Circuits and Systems for Video Technology, IEEE Transactions on 13.7 (2003): 560-576.
- [17] Golias.cz. [online]. [cit. 2014-11-23]. Dostupné z: <http://www.golias.cz/index.php?modul=audio&sub=clanky&page=clanek&id=1697>

- [18] Tomáš Loveček, Peter Nagy. Bezpečnostné systémy: Kamerové bezpečnostné systémy. 2008. 283 s. ISBN 978-80-8070-893-1.
- [19] 4G LTE: 4G Mobile Broadband. [online]. [cit. 2014-11-27]. Dostupné z: <http://www.4gltemall.com/huawei-e398-4g-lte-tdd-fdd-100mbps-usb-surfstick.html>
- [20] HANUS, Stanislav. Nové technologie mobilních komunikací pro integrovanou výuku VUT a VŠB-TUO. Brno, 2013. ISBN 978-80-214-4824-7. Elektronická skripta. VUT v Brně.
- [21] SCHEME, BASIC TRANSMISSION. LTE: the evolution of mobile broadband. IEEE Communications Magazine, 2009, 45.
- [22] SLANINA, Martin. Next Generation Mobile Network Long Term Evolution. Brno, 2014. Přednášky ke skriptům. VUT Brno.
- [23] GSMweb: Frekvenční přiděl na pásmech GSM, DCS, UMTS a LTE v České republice. [online]. [cit. 2014-11-28]. Dostupné z: <http://www.gsmweb.cz/clanky/freq2.htm>
- [24] Access server: Modulační techniky v moderních bezdrátových sítích. [online]. [cit. 2014-11-30]. Dostupné z: <http://access.fel.cvut.cz/view.php?cisloclanku=2010020004>
- [25] EKSTROM, Hannes, et al. Technical solutions for the 3G long-term evolution. Communications Magazine, IEEE, 2006, 44.3: 38-45.
- [26] ARDUINO: Arduino YUN. [online]. [cit. 2014-12-02]. Dostupné z: <http://arduino.cc/en/Main/ArduinoBoardYun?from=Products.ArduinoYUN>
- [27] ADI Global Distribution. [online]. [cit. 2014-12-01]. Dostupné z: www.adiglobal.cz
- [28] Alza.cz. [online]. [cit. 2015-12-01]. Dostupné z: www.alza.cz
- [29] AirLive. [online]. [cit. 2015-12-01]. Dostupné z: www.fs.airlive.com
- [30] FOSCAM. [online]. [cit. 2015-12-01]. Dostupné z: www.foscam.us
- [31] Foscam IPCamera CGI User Guide. [online]. [cit. 2015-04-20]. Dostupné z: <http://www.ipcamcontrol.net/files/Foscam%20IPCamera%20CGI%20User%20Guide-V1.0.4.pdf>
- [32] Arduino: Examples > HttpClient. [online]. [cit. 2015-04-20]. Dostupné z: <http://www.arduino.cc/en/Tutorial/HttpClient>
- [33] Arduino: Examples > Process. [online]. [cit. 2015-04-20]. Dostupné z: <http://www.arduino.cc/en/Tutorial/Process>
- [34] Microsoft Developer Network. [online]. [cit. 2015-04-25]. Dostupné z: [https://msdn.microsoft.com/cs-cz/library/hsy826az\(v=vs.100\).aspx](https://msdn.microsoft.com/cs-cz/library/hsy826az(v=vs.100).aspx)
- [35] Microsoft Developer Network: FtpWebRequest. [online]. [cit. 2015-04-26]. Dostupné z: <https://msdn.microsoft.com/cs-cz/library/system.net.ftpwebrequest%28v=vs.110%29.aspx>

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

APN	Access Point Name (Název přístupového bodu)
AVC	Advanced Video Coding
CCD	Changed Coupled Device
CDMA	Code division multiple access
CGI	Common Gateway Interface
CMOS	Complementary Metal-Oxide-Semiconductor
CP	Cyclic Prefix (Cyklický prefix)
CS	Circuit Switched (Přepínání okruhů)
DCT	Discrete Cosine Transform (Diskrétní kosinová transformace)
DDNS	Dynamic Domain Name System
DRAM	Dynamic Random Access Memory
EPC	Evolved Packet Core
E-UTRAN	Evolved Universal Terrestrial Radio Access Network
FDD	Frequency Division Duplex
FDMA	Frequency Division Multiple Access
FEC	Forward Error Correction (Dopředná chybová korekce)
FET	Field-Effect Transistors
FTP	File Transfer Protocol
GIF	Graphics Interchange Format
GOP	Group Of Pictures (Skupina obrázků)
GPRS	General Packet Radio Service
GSM	Global System for Mobile
HEVC	High Efficiency Video Coding
HSPA	High Speed Packet Access
HSS	Home Subscriber Server
HTTP	Hypertext Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
IFFT	Inverse Fast Fourier Transform
IMSI	International Mobile Subscriber Identity
IP	Internet Protocol
IR	Infra Red
JPEG	Joint Photographic Experts Group

LAN	Local Area Network (Lokální síť)
LED	Light-Emitting Diode (Dioda emitující světlo)
LTE	Long Term Evolution
MIMO	Multiple Input Multiple Output
MME	Mobility Management Entity
MPEG	Moving Picture Experts Group
MSISDN	Mobile Station International Subscriber Directory Number
MW	Micro Wave
NC	Normally Open (Normálně otevřený)
NO	Normally Closed (Normálně zavřený)
NTSC	National Television System Committee
OFDMA	Orthogonal Frequency Division Multiple Access
PAL	Phase Alternation by Line
PCRF	Policy and Charging Rules Function
PDN	Packet Data Network
PIN	Personal Identification Number
PIR	Passive Infra Red (Pasivní infračervený)
PoE	Power over Ethernet (Napájení přes ethernet)
PS	Packet Switched (Přepínání paketů)
QAM	Quadrature amplitude modulation
QoS	Quality of Service
SD	Secure Digital
S-GW	Serving Gateway
SIM	Subscriber Identity Module
SMS	Short Message Service
SSH	Secure Shell
TCP	Transmission Control Protocol
TDD	Time Division Duplex
UE	User Equipment
UMTS	Universal Mobile Telecommunications System
US	Ultra Sound (Ultrazvuk)
USB	Universal Serial Bus (Univerzální sériová sběrnice)
WCDM	Wideband Code Division Multiple Access
Wi-Fi	Wireless Fidelity

PŘÍLOHA

Zdrojový kód

Program FtpApp

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;
using System.Net;

namespace FtpApp
{
    public partial class Form1 : Form
    {
        string ftpServerIP;
        string ftpServerIPdata;
        string ftpUserID;
        string ftpPassword;
        String item = "";

        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            // adresa defaultního ftp serveru
            ftpServerIP =
                "ms2.urel.feec.vutbr.cz/Diplomova_prace/Svoboda/Video";
            // adresa serveru pro ukládání dat
            ftpServerIPdata =
                "ms2.urel.feec.vutbr.cz/Diplomova_prace/Svoboda";
            // přihlašovací jméno k ftp serveru
            ftpUserID = "student";
            // přihlašovací heslo k ftp serveru
            ftpPassword = "urel";
            // zapsání informací do TextBoxů
            txtServerIP.Text = ftpServerIP;
            txtUsername.Text = ftpUserID;
            txtPassword.Text = ftpPassword;

            // naplnění ComboBoxu časovými polážkami
            comboBox1.Items.Add("1");
            comboBox1.Items.Add("2");
            comboBox1.Items.Add("5");
            comboBox1.Items.Add("10");
            comboBox1.Items.Add("15");
        }
    }
}
```

```

comboBox1.Items.Add("20");
comboBox1.Items.Add("25");
comboBox1.Items.Add("30");
comboBox1.SelectedItem = "5";

// zablokování tlačítka pro ukládání informací
btnFtpConfigSave.Enabled = false;
}

public string[] GetFileList()
{
    string[] downloadFiles;
    StringBuilder result = new StringBuilder();
    FtpWebRequest reqFTP; // implementace ftp klienta
    try
    {
        // nastavení potřebných informací pro komunikaci se serverem
        reqFTP = (FtpWebRequest)FtpWebRequest.Create(new Uri("ftp://" +
            ftpServerIP + "/"));
        reqFTP.UseBinary = true;
        reqFTP.Credentials = new NetworkCredential(ftpUserID,
            ftpPassword);
        reqFTP.Method = WebRequestMethods.Ftp.ListDirectory;

        // odeslání požadavku na výpis souborů
        WebResponse response = reqFTP.GetResponse();
        StreamReader reader = new
            StreamReader(response.GetResponseStream());
        string line = reader.ReadLine();
        while (line != null)
        {
            result.Append(line);
            result.Append("\n");
            line = reader.ReadLine();
        }
        result.Remove(result.ToString().LastIndexOf('\n'), 1);
        reader.Close();
        response.Close();
        return result.ToString().Split('\n');
    }
    catch (Exception ex)
    {
        System.Windows.Forms.MessageBox.Show(ex.Message);
        downloadFiles = null;
        return downloadFiles;
    }
}

private void btnLstFiles_Click(object sender, EventArgs e)
{
    update_lstText();
}

private void txtPassword_TextChanged(object sender, EventArgs e)
{
    // pokud byla změněna data, povolí se tlačítko na uložení
    btnFtpConfigSave.Enabled = true;
}

```

```

private void txtServerIP_TextChanged(object sender, EventArgs e)
{
    // pokud byla změněna data, povolí se tlačítko na uložení
    btnFtpConfigSave.Enabled = true;
}

private void txtUsername_TextChanged(object sender, EventArgs e)
{
    // pokud byla změněna data, povolí se tlačítko na uložení
    btnFtpConfigSave.Enabled = true;
}

private void btnFtpConfigSave_Click(object sender, EventArgs e)
{
    // uložení přihlašovacích údajů do proměnných, odstranění
    // prázdných znaků
    ftpServerIP = txtServerIP.Text.Trim();
    ftpUserID = txtUsername.Text.Trim();
    ftpPassword = txtPassword.Text.Trim();
    btnFtpConfigSave.Enabled = false;
}

private void lstFiles_SelectedIndexChanged(object sender,
    EventArgs e)
{
    try
    {
        int index = lstFiles.SelectedIndex;
        // získání názvu souboru z výpisu
        item = lstFiles.Items[index].ToString();
        // vypsání názvu do labelu
        label_name.Text = item;
        label8.Text = "";

        // čas záznamu získán z názvu souboru
        if (item.Substring(0, 3) == "sch")
        {
            String sec = item.Substring(22, 2);
            String min = item.Substring(20, 2);
            String hrs = item.Substring(18, 2);
            String day = item.Substring(15, 2);
            String month = item.Substring(13, 2);
            String year = item.Substring(9, 4);
            label time.Text = hrs + ":" + min + ":" + sec + "  " + day +
                "." + month + "." + year;
        }
        else MessageBox.Show("Vyberte prosím video soubor.");

        // nastavení potřebných informací pro komunikaci se serverem
        String filePath = ("ftp://" + ftpServerIP + "/" + item);
        FtpWebRequest request =
            (FtpWebRequest)WebRequest.Create(filePath);
        request.Credentials = new NetworkCredential(ftpUserID,
            ftpPassword);

        // požadavek na zjištění velikosti souboru
        request.Method = WebRequestMethods.Ftp.GetFileSize;
        FtpWebResponse response =
            (FtpWebResponse)request.GetResponse();
    }
}

```

```

        float size = response.ContentLength;
        size = size / 1048576;
        response.Close();
        // výpis velikosti v MB;
        label_size.Text = size.ToString("0.000") + " MB";
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void buttonDelFile Click(object sender, EventArgs e)
{
    if (item != "")
    {
        try
        {
            // nastavení potřebných informací pro komunikaci se serverem
            String filePath = ("ftp://" + ftpServerIP + "/" + item);
            FtpWebRequest request =
                (FtpWebRequest)WebRequest.Create(filePath);
            request.Credentials = new NetworkCredential(ftpUserID,
                ftpPassword);

            // požadavek na smazání souboru
            request.Method = WebRequestMethods.Ftp.DeleteFile;
            FtpWebResponse response =
                (FtpWebResponse)request.GetResponse();
            response.Close();
            MessageBox.Show("Soubor " + item + " byl smazán");
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
        update_lstText();
    }
    else MessageBox.Show("Nejprve musíte vybrat video soubor");
}

void update_lstText()
{
    // obnovení výpisu souborů v listBoxu
    string[] filenames = GetFileList();
    lstFiles.Items.Clear();
    try
    {
        foreach (string filename in filenames)
        {
            lstFiles.Items.Add(filename);
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
}

```

```

private void buttonSaveFile_Click(object sender, EventArgs e)
{
    if (item != "")
    {
        label8.Text = "Stahování";
        label8.ForeColor = System.Drawing.Color.Red;
        byte[] buffer = new byte[2048];
        // výběr místa pro uložení souboru
        folderBrowserDialog1.ShowDialog();
        String path = folderBrowserDialog1.SelectedPath;
        if (path != "")
        {
            try
            {
                // nastavení potřebných informací pro komunikaci se serverem
                String filePath = ("ftp://" + ftpServerIP + "/" + item);
                FtpWebRequest request =
                    (FtpWebRequest)WebRequest.Create(filePath);
                request.Credentials = new NetworkCredential(ftpUserID,
                    ftpPassword);

                // požadavek na stažení souboru
                request.Method = WebRequestMethods.Ftp.DownloadFile;
                FtpWebResponse response =
                    (FtpWebResponse)request.GetResponse();
                Stream stream = response.GetResponseStream();
                FileStream fs = new FileStream(path + "/" + item,
                    FileMode.Create);
                int ReadCount = stream.Read(buffer, 0, buffer.Length);

                // ukládání souboru po 2kb částech
                while (ReadCount > 0)
                {
                    fs.Write(buffer, 0, ReadCount);
                    ReadCount = stream.Read(buffer, 0, buffer.Length);
                }

                fs.Close();
                stream.Close();
                label8.Text = "Staženo";
                label8.ForeColor = System.Drawing.Color.Green;
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
                label8.Text = "";
            }
        }
        else
        {
            label8.Text = "";
        }
    }
    else
    {
        MessageBox.Show("Nejprve musíte vybrat video soubor");
    }
}

```

```

private void buttonFTPSaveParam_Click(object sender, EventArgs e)
{
    if (txtPhone.Text.Length < 9) //špatně vyplněné pole pro telefonní
        číslo
    {
        MessageBox.Show("Špatný formát telefonního čísla");
    }
    else
    {
        String phoneNumber = "+420" + txtPhone.Text;
        String duration = comboBox1.Text;
        String alarmEnable;
        String fileDur = "duration.txt";
        String filePhoneNum = "phoneNumber.txt";
        String fileAlarmEnable = "alarmEnable.txt";
        if (checkBox1.Checked) alarmEnable = "1";
        else alarmEnable = "0";

        File.WriteAllText(filePhoneNum, phoneNumber);
        File.WriteAllText(fileDur, duration);
        File.WriteAllText(fileAlarmEnable, alarmEnable);

        try
        {
            // nahrání textových souborů s informacemi o čase a
            // telefonním čísle na ftp server
            using (System.Net.WebClient client =
                new System.Net.WebClient())
            {
                client.Credentials = new
                    System.Net.NetworkCredential(ftpUserID, ftpPassword);
                client.UploadFile("ftp://" + ftpServerIPdata + "/" + new
                    FileInfo(fileDur).Name, "STOR", fileDur);
            }

            using (System.Net.WebClient client =
                new System.Net.WebClient())
            {
                client.Credentials = new
                    System.Net.NetworkCredential(ftpUserID, ftpPassword);
                client.UploadFile("ftp://" + ftpServerIPdata + "/" + new
                    FileInfo(filePhoneNum).Name, "STOR", filePhoneNum);
            }

            using (System.Net.WebClient client =
                new System.Net.WebClient())
            {
                client.Credentials = new
                    System.Net.NetworkCredential(ftpUserID, ftpPassword);
                client.UploadFile("ftp://" + ftpServerIPdata + "/" + new
                    FileInfo(fileAlarmEnable).Name, "STOR",
                    fileAlarmEnable);
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }
}

```

```
}  
  
private void txtPhone_KeyPress(object sender, KeyPressEventArgs e)  
{  
    // do pole pro telefonní číslo je povoleno psát pouze čísla  
    if (!char.IsDigit(e.KeyChar) && e.KeyChar != 8)  
    {  
        e.Handled = true;  
    }  
}  
}
```

Program v desce Arduino YUN

```
#include <Process.h>
#include <Bridge.h>
#include <HttpClient.h>

int pin3 = 3;

String usr = "adasvobi";
String pwd = "adasvobi";
String IPCam = "192.168.240.112";
String IPArduino = "192.168.240.1";
String CGIHead = "/cgi-bin/CGIProxy.fcgi?";
String login = "&usr="+usr+"&pwd="+pwd ;
String cameraPath =
    "ftp://adasvobi:adasvobi@192.168.240.112:50021/IPCamera/
    FI9821P_00626E517E62/record/";

String phoneNumber;

HttpClient client;

void setup() {
    pinMode(pin3, INPUT);
    pinMode(13, OUTPUT);
    digitalWrite(13, LOW);
    Bridge.begin();
    digitalWrite(13, HIGH);

    Serial.begin(9600);
    delay(30000); // cekani na nacteni USB modemu

    setTime();
}

void loop() {

    if (digitalRead(pin3))
    {
        String newDateSch;
        String newTimeSch;
        String fileName;
        Process p;

        if (findAlarmEnable())
        {
            alarmOn();
            findNumber();
            digitalWrite(13, LOW);
            sendSMS();
            digitalWrite(13, HIGH);
            int duration = findDuration();
            while (digitalRead(pin3));
            delay(duration * 1000);
            alarmOff();
            digitalWrite(13, LOW);
            delay(5000);
            newDateSch = lastDate();
            newTimeSch = lastTime(newDateSch);
        }
    }
}
```



```

        fileName = fileNameNew(newDateSch, newTimeSch);
        downloadToSd(newDateSch, newTimeSch, fileName);
        digitalWrite(13, HIGH);
        //Serial.print(F("\n ---Po ulozeni do Arduina--- \n"));
        uploadToFtp(fileName);
        digitalWrite(13, LOW);
        //Serial.print(F("\n ---Po uploadu--- \n"));
        removeFile(fileName);
        //Serial.print(F("\n ---Smazano--- \n"));
        removeFileFromCam(newDateSch, newTimeSch, fileName);
    }
}

void alarmOn()
{
    String startAlarm = F("http://192.168.240.112/cgi-
        bin/CGIProxy.fcgi?cmd=setScheduleRecordConfig&isEnabled=1&
        isEnabledAudio=1&schedule0=281474976710655&schedule1=281474976710655&
        schedule2=281474976710655&schedule3=281474976710655&
        schedule4=281474976710655&schedule5=281474976710655&
        schedule6=281474976710655&usr=adasvobi&pwd=adasvobi");
    client.get(startAlarm); // požadavek na zapnutí alarmu
    /*Serial.print(F("Zapni alarm\n"));
    while (client.available() // přijímá odpověď od kamery
    {
        char c = client.read();
        Serial.print(c); // vypíše odpověď
    }*/
    Serial.flush();
}

void alarmOff()
{
    String offAlarm = F("http://192.168.240.112/cgi-
        bin/CGIProxy.fcgi?cmd=setScheduleRecordConfig&isEnabled=0&
        usr=adasvobi&pwd=adasvobi");
    client.get(offAlarm); // požadavek na vypnutí alarmu
    //Serial.print(F("\nVypni alarm\n"));
}

void setTime()
{
    String time = getTime();
    String day = time.substring(0,2);
    String mon = time.substring(3,5);
    String year = time.substring(6,10);
    String hour = time.substring(11,13);
    String minute = time.substring(14,16);
    String sec = time.substring(17,19);

    String setTimeCGI =

    "http://" + IPCam + CGIHead + "cmd=setSystemTime&timeSource=1&dateFormat=1
        &timeFormat=1&year="+year+"&mon="+mon+"&day="+day+"&hour="+hour+
        &minute="+minute+"&sec="+sec+login;
    client.get(setTimeCGI);
}

```

```

}

String getTime()
{
    Process p;
    String time;
    p.begin("date");
    p.addParameter("+%d-%m-%Y_%T");
    p.run();
    while (p.available()>0)
    {
        char c = p.read();
        //Serial.print(c);
        time += c;
    }
    return time;
}

String lastDate()
{
    Process p;
    String newDateSch;
    String newDateSchHelp;

    p.begin("curl");
    p.addParameter("--ftp-port");
    p.addParameter(IPArduino);
    p.addParameter(cameraPath); //Vypíše soubory
    p.run();

    while (p.available()>0)
    {
        char c = p.read();
        //Serial.print(c);
        newDateSchHelp += c;
    }
    newDateSch = newDateSchHelp.substring(newDateSchHelp.length()
        -9,newDateSchHelp.length()-1);
    Serial.flush();
    return newDateSch;
}

String lastTime(String newDateSch){
    Process p;
    String newTimeSch;
    String newTimeSchHelp;

    p.begin("curl");
    p.addParameter("--ftp-port");
    p.addParameter(IPArduino);
    p.addParameter(cameraPath+newDateSch+"/"); //Vypíše soubory
    p.run();

    while (p.available()>0)
    {
        char c = p.read();
        //Serial.print(c);
        newTimeSchHelp += c;
    }
}

```

```

    }
    newTimeSch = newTimeSchHelp.substring(newTimeSchHelp.length()-
        16,newTimeSchHelp.length()-1);
    Serial.flush();
    return newTimeSch;
}

String fileNameNew(String newDateSch, String newTimeSch)
{
    Process p;
    String newFile;
    String newFileHelp;
    int i=0;

    p.begin("curl");
    p.addParameter("--ftp-port");
    p.addParameter(IPArduino);

    //Vypíše soubory
    p.addParameter(cameraPath+newDateSch+"/"+newTimeSch+"/");
    p.run();

    while (p.available()>0)
    {
        char c = p.read();
        //Serial.print(c);
        newFileHelp += c;
    }
    int strLenght = newFileHelp.length()-1;

    while(newFileHelp[strLenght - i] != ' ')
    {
        i++;
    }

    newFile = newFileHelp.substring(newFileHelp.length()-
        i,newFileHelp.length()-1);
    //Serial.print(F("Nazev souboru\n"));
    //Serial.print(newFile + "\n");
    Serial.flush();
    return newFile;
}

void downloadToSd(String newDateSch, String newTimeSch, String
fileName)
{
    Process p;
    p.runShellCommand("wget
        "+cameraPath+newDateSch+"/"+newTimeSch+"/"+fileName);
    while(p.running());
}

void uploadToFtp(String fileName)
{
    Process p;
    p.runShellCommand("curl -T /usr/lib/python2.7/bridge/"+fileName+"
        ftp://ms2.urel.feec.vutbr.cz/Diplomova_prace/Svoboda/Video/ --user
        student:urel");
    while(p.running());
}

```

```

}

void removeFile(String fileName)
{
    Process p;
    p.runShellCommand("rm /usr/lib/python2.7/bridge/"+fileName);
    while(p.running());
}

void removeFileFromCam(String newDateSch, String newTimeSch,
String fileName)
{
    Process p;
    String removeFile = ("curl -X \"DELETE "+fileName+"\" --ftp-port
192.168.240.1 ftp://adasvobi:adasvobi@192.168.240.112:50021/
IPCamera/FI9821P_00626E517E62/record/"+newDateSch+"/"
+newTimeSch+"/");
    p.runShellCommand(removeFile);
    while(p.running());
}

void sendSMS()
{
    //String phoneNumber = "+420732825385";
    Process p;
    p.begin("python");
    p.addParameter("/mnt/sda1/arduino/www/sms.py");
    p.addParameter(phoneNumber);
    p.run();
}

void findNumber()
{
    phoneNumber = "";
    HttpClient client;
    String findNum =

    F("ftp://student:urel@ms2.urel.feec.vutbr.cz/Diplomova_prace/Svoboda
/phoneNumber.txt");
    client.get(findNum);
    while (client.available())
    {
        char c = client.read();
        //Serial.print(c);
        phoneNumber += c;
    }
    Serial.flush();
}

int findDuration()
{
    int duration = 0;
    String help = "";
    HttpClient client;
    String findDur =

    F("ftp://student:urel@ms2.urel.feec.vutbr.cz/Diplomova_prace/Svoboda
/duration.txt");
    client.get(findDur);
}

```

```

while (client.available())
{
    char c = client.read();
    //Serial.print(c);
    help += c;
}
Serial.flush();
return (help.toInt());
}

boolean findAlarmEnable()
{
    boolean alarmEnabled = 0;
    String help = "";
    HttpClient client;
    String findAlarmEnable =

    F("ftp://student:urel@ms2.urel.feec.vutbr.cz/Diplomova_prace/Svoboda
    /alarmEnable.txt");
    client.get(findAlarmEnable);
    while (client.available())
    {
        char c = client.read();
        help += c;
    }
    Serial.flush();

    if (help[0] == '1') alarmEnabled = true;
    else alarmEnabled = false;

    return (alarmEnabled);
}

```

Program na odesílání SMS zpráv

```
import serial
import time
import sys

phoneNumber = sys.argv[1]

class TextMessage:
    def __init__(self, recipient="+420732825385", message="Alarm"):
        self.recipient = recipient
        self.content = message

    def setRecipient(self, number):
        self.recipient = number

    def setContent(self, message):
        self.content = message

    def connectPhone(self):
        self.ser = serial.Serial('/dev/ttyUSB1', 115200, timeout=5)
        time.sleep(1)

    def sendMessage(self):
        self.ser.write('ATZ\r')
        time.sleep(1)
        self.ser.write('AT+CMGF=1\r')
        time.sleep(1)
        self.ser.write(''AT+CMGS="" + self.recipient + ""\r''')
        time.sleep(1)
        self.ser.write(self.content + "\r")
        time.sleep(1)
        self.ser.write(chr(26))
        time.sleep(1)

    def disconnectPhone(self):
        self.ser.close()

sms = TextMessage(phoneNumber, "Alarm.")
sms.connectPhone()
sms.sendMessage()
sms.disconnectPhone()
```

Obsah přiloženého CD

- *FtpApp* – Složka ve které je uložen kompletní program pro správu videí na FTP serveru se zdrojovými kódy
- *SecuritySystem* – Složka se zdrojovým kódem pro mikroprocesor ATmega
- *sms.py* – Zdrojový kód programu na odesílání SMS zpráv