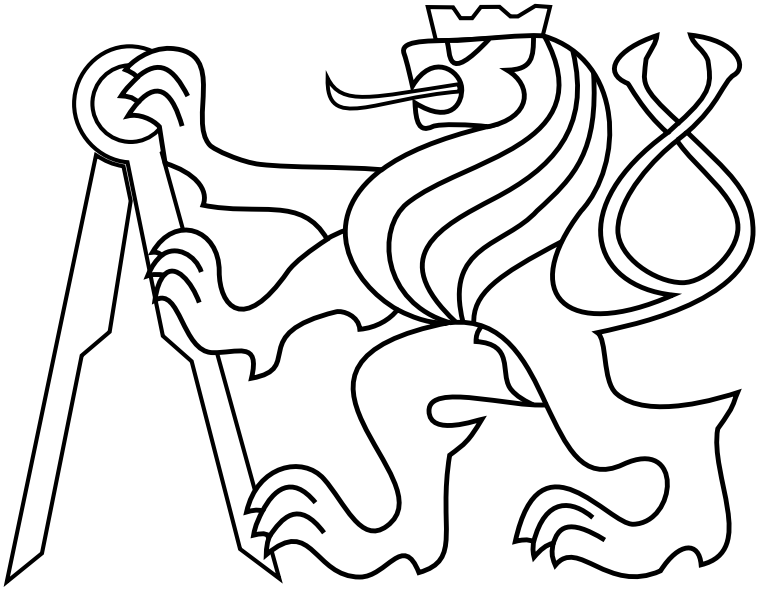
ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická

Diplomová práce



Bc. David Hývl

Aplikace pro detekci požáru pomocí termokamery

Katedra řídicí techniky

Vedoucí práce: **Ing. Jan Kovář**

Prohlášení autora práce

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne..... .....

ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Hývl** Jméno: **David** Osobní číslo: **406378**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra řídicí techniky**

Studijní program: **Kybernetika a robotika**

Studijní obor: **Systémy a řízení**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Aplikace pro detekci požáru pomocí termokamery

Název diplomové práce anglicky:

Application for fire hazard detection using thermal camera

Pokyny pro vypracování:

1. Seznamte se s problematikou bezdotykového měření teploty pomocí mikrobolometrických kamer a jejich připojení k PC .
2. Zpracujte návrh aplikace a její funkce pro detekci požáru pomocí termokamery SMARTIS.
3. Navrhněte a v jazyce C# implementujte aplikaci, která bude řídit potřebná zařízení, využijte při tom návrhový vzor Model-View-ViewModel. Aplikace by měla obsahovat především tyto funkce:
- Funkce pro ovládání parametrů měření přes Ethernet.
- Funkce pro ovládání PTZ motoru přes Ethernet.
- Funkce pro zobrazení infračerveného a viditelného videostreamu z kamery přes Ethernet s minimalni latencí v obraze a podporou až pro 16 kamer současně.
4. Navrhněte a do aplikace a dále implementujte podporu více zařízení a automatický záznam neradiometrického videa a dále radiometrického cyklického videa. Dále využijte digitálních výstupů kamery a průmyslového PC pro ovládání externích zařízení např. pro automatické hašení.
5. Otestujte výpočetní výkonnost a dosahované parametry výsledného obrazu v aplikaci.
6. Diskutujte možnost vylepšení implementovaných nástrojů a použitých algoritmů.

Seznam doporučené literatury:

[1] Raytek (firma): Noncontact temperature measurement, 2004.
[2] Vollmer M.: Infrared Thermal Imaging: Fundamentals, Research and Applications, 2010.
[3] Fluke Corporation: Precision measurement solutions : electrical calibration, temperature calibration, time and frequency standards, calibration software, data acquisition, VXI products, signal sources, value-added services, 2004.
[4] Heitronics Infrarot Messtechnik GmbH: Infrared measuring : instruments and accessories, 2000.

Jméno a pracoviště vedoucí(ho) diplomové práce:

Ing. Jan Kovář, Workswell s.r.o., Praha

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **01.10.2018** Termín odevzdání diplomové práce: **24.05.2019**

Platnost zadání diplomové práce: **30.09.2020**

Ing. Jan Kovář	prof. Ing. Michael Šebek, DrSc.	prof. Ing. Pavel Ripka, CSc.
podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	podpis děkana(ky)

 © ČVUT v Praze, Design: ČVUT v Praze, VIC

Poděkování

Nejdříve bych rád poděkoval Ing. Janu Kovářovi za odborné vedení práce, cenné rady, připomínky a čas strávený konzultacemi. Dále chci poděkovat rodině a přátelům za podporu během studia a při psaní této práce.

Abstrakt

Tato diplomová práce se zabývá implementací a návrhem softwaru pro detekci a signalizaci požáru za pomocí infračervené termografie. Úvodní část práce se zabývá obecným popisem principů infračervené termografie a použitým přístrojům. V další části práce je popsán přístup k návrhu aplikace, návrhový vzor MVVM a software včetně požadavků, implementovaných funkcí a popisem kompletní funkčnosti celého navrhnutého řešení pro detekci požáru. Závěrem této práce je konzultace nad dosaženými požadavky vytvořit komplexní a stabilní řešení pro použití v místech zhoršeného přístupu a vysokého rizika vzniku požáru.

Klíčová slova: termografie, termální kamera, C# aplikace, MVVM, detekce požáru, riziko požáru

Abstract

This diploma thesis deals with the implementation and design of the software for the detection and signalization of the fire hazard using the infrared thermography. The introduction of this thesis deals with general description of the principles of the infrared thermography and the used devices. In the following chapters is described the approach to the application design, the MVVM design pattern and the software itself including the requirements, description of the implemented functions, and the functionality of whole fire detection solution. In the end of this thesis is also included the consultation about fullfilling the requirements to create a complex and stable solution for use in the places with worse accessibility and fire hazard risk.

Keywords: thermography, thermal camera, C# application, MVVM, fire detection, fire hazard

Obsah

Seznam obrázků **III**

Seznam tabulek **IV**

1 Úvod **1**
1.1 Osobní motivace . . . 1
1.2 Stávající řešení detekce požáru . . . 2

2 Infračervená termografie **3**
2.1 Tepelné záření . . . 3
2.1.1 Černé a šedé těleso . . . 4
2.1.2 Emisivita . . . 6
2.1.3 Pohltivost (absorbance) . . . 6
2.1.4 Odrazivost (reflektance) . . . 7
2.1.5 Propustnost (transmitance) . . . 7
2.2 Kirchhoffovy vyzařovací zákon . . . 7
2.3 Planckův vyzařovací zákon . . . 8
2.4 Wienův posunovací zákon . . . 9
2.5 Rovnice termografie . . . 9
2.5.1 Absorpce . . . 10
2.5.2 Rozptyl . . . 10
2.5.3 Turbulence . . . 10
2.5.4 Emise . . . 10
2.6 Detektory infračerveného záření . . . 10
2.6.1 Termální senzory . . . 11
2.6.2 Kvantové senzory . . . 12

3 Návrh systému a použitá zařízení **13**
3.1 Termokamera SAFETIS . . . 13
3.2 Průmyslové PC Vecow IVH9000 . . . 14
3.3 Převodník Edgar . . . 15
3.4 PT motor s řídicí jednotkou . . . 16
3.4.1 Pelco-D . . . 17

4 Popis implementačních nástrojů **18**
4.1 Vývojová platforma .NET Framework a programovací jazyk C# . . . 18
4.2 Popis Windows Presentation Foundation (WPF) . . . 18
4.3 Návrhový vzor MVVM . . . 19

5 Základní popis aplikace pro detekci požárů **21**
5.1 Parametry a popis aplikace . . . 21
5.2 Instalace a šíření aplikace . . . 21
5.2.1 Instalace aplikace . . . 22
5.3 Struktura aplikace a zdrojového kódu . . . 22
5.3.1 Struktura spuštění . . . 22
5.3.2 Struktura grafického rozhraní . . . 23

5.3.3 Hierarchie zdrojového kódu 23

6 Popis grafického rozhraní aplikace a funkcí 25
6.1 Grafické rozdělení hlavního okna a popis komponent 25
6.1.1 Hlavička a hlavní menu aplikace 25
6.1.2 Informační panel 26
6.1.3 Centrální panely hlavního okna 26
6.2 Správa uživatelů aplikace 28
6.2.1 Administrace uživatelů 29
6.3 Konfigurační režim aplikace 30
6.4 Práce s kamerami SAFETIS 31
6.4.1 Přidání a mazání ze seznamu 31
6.4.2 Připojování a komunikace 32
6.4.3 Generování obrazu 33
6.4.4 Konfigurace 34
6.5 Nastavení aplikace 36
6.5.1 Ukládání a implementace dat nastavení 36
6.6 Aktualizace aplikace a kamer 37
6.7 Záznam dat z kamery 38
6.7.1 Formát ukládání radiometrických dat 39
6.7.2 Alarmový stav aplikace 39
6.8 Záznam grafického průběhu teploty 40
6.9 Ovládání a funkce PT motoru 41

7 Závěr 43
7.1 Výsledky testování aplikace 43
7.2 Diskuze možných vylepšení 43
7.3 Zhodnocení dosažených výsledků 44

8 Literatura 46

A Obsah disku 48

B Seznam použitých zkratek 49

C Příkazy pro komunikaci s kamerou SAFETIS 50

D Rozdělení tříd aplikace 50

E Grafické uživatelské rozhraní aplikace 56

F Snímky testovacího zapojení systému 58

Seznam obrázků

1 Termokamera SAFETIS v krytu . . . 1
2 Rozdělení vlnových délek [4] . . . 3
3 Porovnání termogramu a snímku pořízeném ve viditelném spektru . . . 4
4 Chování tělesa při dopadu elektromagnetického záření, dle [11] . . . 5
5 Vliv okolí na bezkontaktní měření teploty povrchu termokamerou . . . 9
6 Mikrobolometrický detektor [11] . . . 11
7 Schéma zapojení systému . . . 13
8 Termokamera SAFETIS . . . 14
9 Průmyslové PC Vecow IVH 9016 [11] . . . 15
10 Převodník Edgar [11] . . . 16
11 PTZ motor s řídicí jednotkou . . . 17
12 Schéma principu návrhového vzoru MVVM [12] . . . 19
13 Náhled instalátoru aplikace . . . 23
14 Členění aplikace do tzv. "namespace" . . . 24
15 Miniatura kamery SAFETIS pro panel "Thermal Streams" . . . 26
16 Panel s detailním pohledem na vybranou kameru SAFETIS . . . 27
17 Okno správy uživatelů . . . 29
18 Okno pro nastavení párování disků se síťovým připojením . . . 30
19 Okno nastavení kamer SAFETIS . . . 32
20 Okno konfigurace kamery SAFETIS při definování měřících oblastí . . . 34
21 Okno konfigurace kamery SAFETIS při nastavení parametrů měření . . . 35
22 Okno pro nastavení aplikace . . . 37
23 Okno aktualizaci kamer a aplikace . . . 37
24 Panel grafického záznamu maximální teploty . . . 40
25 Panel pro ovládání PT motoru . . . 42
26 Obsah složky (Namespace) Models . . . 52
27 Obsah složky (Namespace) View . . . 53
28 Obsah složky (Namespace) ViewModels . . . 54
29 Obsah složky (Namespace) Utils . . . 55
30 Celé grafické uživatelské rozhraní aplikace . . . 56
31 Celé grafické uživatelské rozhraní aplikace při alarmové situaci . . . 57
32 Snímek testovacího zapojení systému . . . 58
33 Snímek kamery SAFETIS v krytu na PT motory . . . 59

Seznam tabulek

1 Přehled možností provázání datového propojení 20
2 Přehled zaznamenaných událostí 28
3 Přehled konfiguračních parametrů 35
4 Popis časových úseků grafu 41
5 Obsah přiloženého disku CD 48
6 Seznam použitých zkratek 49
7 Příkazy pro vyčtení dat z termokamery SAFETIS 50
8 Příkazy pro nastavení dat termokamery SAFETIS 51

1 Úvod

Termokamery jsou zařízení umožnující měřit a interpretovat infračervené záření vydávané každým tělesem a díky nim je nám umožněno měřit bezkontaktně teplotu těchto objektů. Jelikož jsme schopni u jednotlivých materiálů určit teplotu, při které se vznítí, je tedy možné za určitých podmínek detekovat riziko požáru ještě před jeho vznikem, tím předejít kritické situaci a okamžitě reagovat.

Tato diplomová práce se tedy zabývá návrhem řešení pro systém, který bude schopný toto riziko zaznamenat, podle nastavených parametrů varovat uživatele o vzniklém riziku a provést příslušné akce.

V této práci se tedy nejprve podíváme na návrh samotného řešení, použité komponenty a jejich popis a zapojení. Dalším bodem je návrh a implementace obslužné aplikace, která bude podporovat připojení a ovládání popsaného hardwaru. A posledním bodem je shrnutí dosažených výsledků a diskuze nad možnými úpravami a vylepšeními pro další verze této aplikace.

Dle zadání je pro návrh systému uvedena termokamera SMARTIS. Té však především z marketingových důvodů bylo změněno jméno na SAFETIS (viz obrázek č.1). Tedy z pohledu implementace se jedná o stejnou kameru se shodnou platformou, avšak změněným názvem a upraveným firmwarem dle požadavků implementované aplikace. Z tohoto důvodu jsou v aplikaci uvedeny názvy kamery SAFETIS a v názvech tříd aplikace je historicky uveden SMARTIS.

Obrázek 1: Termokamera SAFETIS v krytu

1.1 Osobní motivace

Práce pro mě znamenala novou zkušenost a to jak při implementaci aplikace samotné, tak i v nutnosti propojit hardwarovou a softwarovou část daného řešení. Další motivací pro mě byla vidina možnosti implementovat systém, který se jednak bude využívat po celém světě, a zároveň přispěje k vyšší bezpečnosti zaměstnanců pracujících v prostředí s vysokým rizikem vzniku požáru.

1.2 Stávající řešení detekce požáru

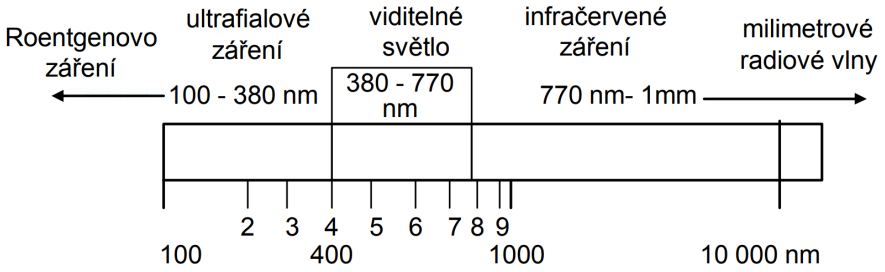
Stávající detekci požáru lze rozdělit do několika skupin, a to na detektory vizuální, kouřové a termální. Vizuální detektory požáru jsou ve většině kamery s viditelným obrazem a požár je detekován ve chvíli propuknutí, případně lze detekovat vizuálně kouř. Toto řešení je velmi často kritické a požáru již nelze plně zabránit. Další možností je použití detektorů kouře či plynu, které jsou schopny reagovat dříve než detektory vizuální. Avšak i v tomto případě je nutnost reagovat okamžitě. Pokud se budeme zabývat termálními detektory požáru, nebo obecně termokamerami, můžeme říci že jsou schopné zachytit současnou teplotu objektu a předejít s dostatečnou časovou rezervou vzniku požáru. Tedy reagují rychleji, než všechna předešlá řešení, a jejich využití je tedy výhodné. V současné době existuje velmi málo variant těchto systémů pracujících na principu termografie. Konkrétně z těch hlavních jsem při výzkumu nalezl systém od společnosti FLIR (Early Fire Detection) [5], dále řešení FIRESCAN od společnosti InfraTec [8] a několik dalších menších systémů využívajících především kamery Flir. Ty mají jako přidanou hodnotu implementovánu vlastní obslužnou aplikaci. Řešení, popsané a navržené v této práci, by se mělo od ostatních odlišovat hlavně v použití termokamery SAFETIS od společnosti Workswell s.r.o a využití digitálních výstupů této kamery, dále možností zapojení Pan-Tilt motoru pro nastavování pozice a možností dalších měřících funkcí. V neposlední řadě také využitím digitálních výstupů průmyslového PC, které je součástí tohoto řešení.

2 Infračervená termografie

Infračervená termografie je vědní obor zabývající se bezkontaktním měřením teploty těles a to jak za pomocí například pyrometrů, tak i termokamer. Jelikož není měření tepelného záření tělesa úplně triviální, je třeba nejprve definovat jeho základní vlastnosti a do jisté míry pochopit jeho fyzikální podstatu, aby nedošlo k vyvození mylných závěrů či nesprávných výsledků. Tímto se bude zabývat následující kapitola, kde bude popsán princip měření teplotního pole na povrchu tělesa, kterou se zabývá právě termografie. Nejprve definujeme základy, včetně důležitých veličin a vzorců, tedy fundamentální základy, ze kterých budeme dále vycházet v další části při popisu fyzikálního modelu využívaného pro měření teplotního pole na povrchu tělesa. Poslední část bude věnována infračerveným kamerám, a to hlavně z pohledu jejich principu fungování.

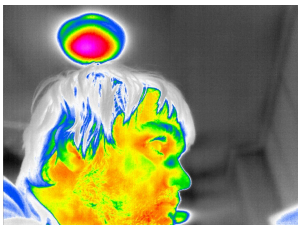
2.1 Tepelné záření

Všechna tělesa, jejichž teplota je vyšší než absolutní nula, tedy vyšší nežli teplota 0 Kelvinů, se na molekulární úrovni pohybují, a to i přesto, že z makroskopického hlediska je předmět v relativním klidu. Tento mikroskopický pohyb či vibrace molekul se nazývá tepelný pohyb a je příčinou tepelného záření všech takovýchto těles. Toto generované záření však není ve viditelném spektru, jelikož jeho vlnová délka se pohybuje především v infračervené oblasti elektromagnetického spektra. Konkrétně můžeme výskyt infračerveného záření pozorovat na vlnových délkách od $770nm$ do $1mm$, přičemž rozdělení vlnových délek můžeme vidět na obrázku č.2 [4], kde lze pozorovat i fakt, že malá část infračerveného záření je pozorovatelná lidským okem, a to jako barvy velmi blízké červené (o vlnové délce blízko $770nm$) [16].



Obrázek 2: Rozdělení vlnových délek [4]

Jelikož ale většina tepelného záření není viditelná pouhým okem, je třeba použít senzory detekující záření o těchto vlnových délkách a změřená data následně zpracovat a vizualizovat. Taková vizualizace se poté nazývá termogram a reprezentuje ve viditelném spektru hodnoty měřené teploty tělesa. Porovnání snímku ve viditelném spektru a termogramu je uveden na obrázku č. 3.



(a) Snímek ve viditelném spektru

(b) Termogram

Obrázek 3: Porovnání termogramu a snímku pořízeném ve viditelném spektru

2.1.1 Černé a šedé těleso

Černé těleso bylo ve fyzice zavedeno v roce 1860 Gustavem Kirchhofem jako konstrukt pro popis dějů, ke kterým dochází při tepelném záření. Na černé těleso lze pohlížet dvěma způsoby, a to jako na ideální těleso, nebo jako na ideální zářič. Ideálním objektem je myšleno, že takovéto těleso pohlcuje všechno dopadající záření ze všech směrů a všech vlnových délek, tedy lze také říci, že takovéto těleso neodráží žádné okolní záření. Důsledkem tohoto faktu je, že všechny jeho vlastnosti jsou závislé pouze na teplotě, nikoliv na dopadajícím záření. Pokud nahlížíme na černé těleso jako na zářič, jedná se o ideální všesměrový zářič, který vyzařuje maximální možné množství energie určené teplotou tohoto tělesa. Navíc pokud budeme uvažovat černé těleso, které dosáhlo stavu teplotní rovnováhy a jeho teplota je konstantní, pak množství pohlcené energie tohoto tělesa se rovná množství energie jím vyzářené [7, 11].

Vztah popisující celkovou intenzitu vyzařování tělesa se nazývá Stefan-Boltzmannův zákon a lze ho vyjádřit jako 2.1

$$M_{BB} = \sigma \cdot T^4 \ [Wm^{-2}] \tag{2.1}$$

kde $\sigma = 5.67 \cdot 10^{-8}\ Wm^{-2}K^4$ je Stefan-Boltzmannova konstanta a T je teplota v Kelvinech.

Jelikož skutečné objekty nedosahují vlastností černého tělesa, a to hlavně v ohledu pohltivosti dopadajícího záření, což znamená že ne všechno dopadnuté záření je dokonale pohlceno, je třeba definovat takzvané šedé těleso. Poté můžeme určit rozklad záření do několika složek, které budou popsány níže, a zároveň lze modifikovat Stefan-Boltzmanův zákon do tvaru pro popis intenzity vyzařování šedého tělesa jako 2.2

$$M_{GB} = \varepsilon \cdot \sigma \cdot T^4 \ [Wm^{-2}] \tag{2.2}$$

kde M_{GB} je celková intenzita vyzařování šedého tělesa ve vakuu a ε je emisivita, která je podrobněji popsána v kapitole 2.1.2.

Další, pro radiometrii důležitou veličinou, kterou je třeba pro popis definovat, je zářivý tok [7, 11]. Ten lze určit jako 2.3

$$\Phi = \frac{dQ}{dt} = \int_A I dA \, [W] \tag{2.3}$$

kde Q je energie dané elektromagnetické vlny, t je čas, I je intenzita záření a A je plocha, kterou dané záření prochází. Dále můžeme určit vztah pro obecnou intenzitu vyzařování 2.4, který lze aplikovat i na předchozí definice intenzity vyzařování černého a šedého tělesa 2.1 a 2.2

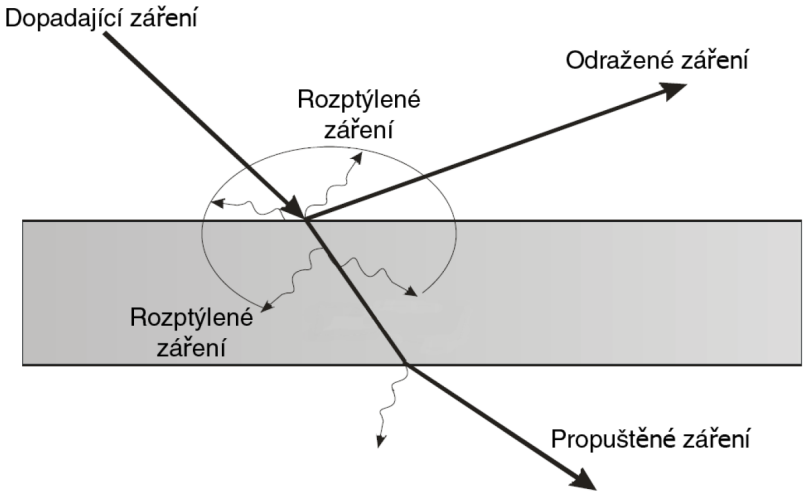
$$M = \frac{\partial \Phi}{\partial A} \, [Wm^{-2}] \tag{2.4}$$

kde Φ je opět zářivý tok a A je plocha vyzařování tělesa. Celkovou intenzitu vyzařování tělesa je možné také určit pomocí vztahu 2.5

$$M = \int_0^\infty M_\lambda d\lambda \, [Wm^{-2}] \tag{2.5}$$

kde λ je označení vlnové délky a M_λ je spektrální hustota intenzity vyzařování, kterou lze určit jako 2.6. Tato spektrální hustota je část energie vyzářená na jednotkovém intervalu vlnové délky.

$$M_\lambda = \frac{\partial M}{\partial \lambda} \, [Wm^{-1}] \tag{2.6}$$



Obrázek 4: Chování tělesa při dopadu elektromagnetického záření, dle [11]

Jak již bylo zmíněno výše, při dopadu elektromagnetického záření na povrch šedého tělesa, nedochází k jeho úplnému pohlcení. Dopad záření lze rozdělit na tři případy a to nejprve odraz, kdy se část dopadajícího záření odrazí od povrchu tělesa. Zbytek záření pronikne do objektu a na základě vlastností materiálu ze kterého se těleso skládá dojde k pohlcení

části energie tohoto záření. Pokud je materiál dostatečně opticky mohutný, dojde k pohlcení zbytkového záření. V opačném případě je možné, že část záření projde skrz těleso. Tento proces je schematicky znázorněn na obrázku č.4.

Pro popis dopadajícího zaření a jeho vztahu k tělesu je ve fyzice zavedeno několik veličin. Ty jsou popsány a definovány v následujících kapitolách.

2.1.2 Emisivita

Emisivita je v termografii pravděpodobně nejvíce využívanou veličinou. Značí se ε a vyjadřuje poměr vyzařování reálného objektu, ku vyzařování černého tělesa. Obecně je emisivita závislá na několika parametrech, a to konkrétně na vlnové délce záření, směru dopadu a na teplotě povrchu. V takovém případě hovoříme o spektrální směrové emisivitě a můžeme ji vyjádřit jako 2.7

$$\varepsilon(\Theta, \Phi, T) = \frac{l_\lambda(\Theta, \Phi, T)}{l_{\lambda BB}(T)} \tag{2.7}$$

kde Θ a Φ jsou úhly pro azimut a elevaci dopadajícího záření, T je Teplota v Kelvinech, $l_\lambda(\Theta, \Phi, T)$ je intenzita záření pro danou vlnovou délku, směr a teplotu a $l_{\lambda BB}(T)$ je intenzita záření černého tělesa pro danou vlnovou délku a teplotu. Při potřebě přesných výpočtů, týkajících se především tepelné výměny radiací, je výhodné využít znalosti o závislosti směru na vlnové délce pro daný materiál, Avšak v praxi se dá považovat průměrná hodnota emisivity pro všechny směry dopadu záření za konstantní. Takto lze definovat jednodušší závislost pro stanovení emisivity jako poměr vyzářených energií zkoumaného tělesa a černého tělesa za shodné teploty a to konkrétně jako 2.8

$$\varepsilon(T) = \frac{M(T)}{M_{BB}(T)} \tag{2.8}$$

kde, $M(T)$ je energie vyzářená zkoumaným tělesem za dané teploty a $M_{BB}(T)$ je energie vyzářená černým tělesem za dané teploty [11, 7, 9].

2.1.3 Pohltivost (absorbance)

Další veličinou je pohltivost, neboli absorbance, která popisuje absorpci záření materiálem. Lze jí definovat jako poměr pohlceného a celkového zářivého toku (viz 2.9)

$$\alpha = \frac{\Phi_a}{\Phi_0} \tag{2.9}$$

kde Φ_a je zářivý tok pohlcený a Φ_0 je celkový zářivý tok. K pohlcení záření zpravidla dochází při několika jevech, a to při elektronové absorpci, absorpci v mřížce a dielektrické disperzi. K elektronové absorpci dochází pouze ve vodivých, či polo-vodivých ma-

teriálech, kde dochází k interakci elektromagnetického záření s volnými elektrony nebo elektronovými dírami v daném materiálu. Absorpce v mřížce se vyskytuje v případě nevodičů a je způsobena společným pohybem dopadajícího záření a tepelné vibrace atomů v krystalové mřížce. Posledním jevem je dielektrická disperze, kde dochází k rozptýlení elektromagnetické vlny z důvodu mikroskopických vad materiálu [11, 7, 9].

2.1.4 Odrazivost (reflektance)

Při dopadu záření nedojde k absorbování celé jeho energie, ale na reálných tělesech dochází k jeho odrážení. Odrazivost popisuje právě tento odraz a lze ji definovat jako poměr odraženého zářivého toku a toku celkového (viz 2.10)

$$\rho = \frac{\Phi_r}{\Phi_0} \tag{2.10}$$

kde Φ_r značí zářivý tok odražený a Φ_0 celkový zářivý tok. Odrazivost nabývá hodnot od 0 do 1 tedy $\rho \in < 0; 1 >$. Pokud budeme uvažovat materiál s velmi lesklým povrchem, bude se odrazivost blížit k hodnotě 1 a bude docházet k takzvanému zrcadlovému odrazu. To znamená, že dopadající záření, respektive paprsky, jsou odraženy v určitém směru. Opakem zrcadlového odrazu je odraz difuzní, který nastává při odrazivosti materiálu blížící se nule. Tyto extrémní případy se v praxi téměř nevyskytují [11, 7, 9].

2.1.5 Propustnost (transmitance)

Propustnost je veličina vyjadřující množství záření, které projde po dopadu daným materiálem. Můžeme ji vyjádřit jako poměr zářivého toku propuštěného materiálem a celkového zářivého toku. Tento vztah lze vyjádřit vzorcem jako 2.11

$$\tau = \frac{\Phi_t}{\Phi_0} \tag{2.11}$$

kde Φ_t je zářivý tok, který prošel materiálem a Φ_0 je zářivý tok celkový. Jakožto všechy předchozí definované veličiny, tedy emisivita (2.1.2), pohltivost (2.1.3) a odrazivost (2.1.4), je i propustnost závislá na vlnové délce dopadajícího záření. U této veličiny lze definovanou závislost demonstrovat nejjednodušeji, a to za použití skla. Sklo je pro záření ve viditelném spektru propustné. Pokud se ale vlnová délka záření dostane do infračerveného spektra, sklo je pro něj téměř nepropustné [11, 7, 9].

2.2 Kirchhoffovy vyzařovací zákon

Kirchhofův vyzařovací zákon určuje závislost mezi pohltivostí (2.1.3), odrazivostí (2.1.4) a propustností (2.1.5) [7, 9]. Tato závislost je vyjádřena jako 2.12

$$\phi_0 = \phi_\alpha + \phi_\rho + \phi_\tau, \tag{2.12}$$

kde ϕ_0 je celkový zářivý tok, ϕ_α je pohlcený zářivý tok, ϕ_ρ je odražený zářivý tok a ϕ_τ vyjadřuje prostupný zářivý tok tělesem. Na základě této závislosti 2.12 lze provést její úpravu, kterou dostaneme Kirchhofův vyzařovací zákon 2.13.

$$1 = \frac{\phi_\alpha}{\phi_0} + \frac{\phi_\rho}{\phi_0} + \frac{\phi_\tau}{\phi_0} = \alpha + \rho + \tau \tag{2.13}$$

Kirchhoffův vyzařovací zákon definuje, že pokud se těleso nachází v termodynamické rovnováze, je jeho emisivita rovna pohltivosti, což znamená, že záření pohlcené tělesem je plně vyzářeno. Z tohoto předpokladu, lze upravit rovnici 2.13 do následujícího tvaru.

$$1 = \varepsilon + \rho + \tau. \tag{2.14}$$

Pokud budeme uvažovat materiál dokonale nepropustný, tedy jeho propustnost bude rovna 0 ($\tau = 0$), můžeme rovnici 2.14 upravit do tvaru, ze kterého vyplývá závislost mezi pohltivostí, případně emisivitou, a odrazivostí. Tato závislost 2.15 udává nepřímou úměru mezi zářením pohlceným a odraženým. Ilustraci jevů vyskytujících se při dopadu záření je možné vidět na obrázku č. 4.

$$\begin{aligned} 1 &= \alpha + \rho, \\ 1 &= \varepsilon + \rho. \end{aligned} \tag{2.15}$$

2.3 Planckův vyzařovací zákon

Pro stanovení intenzity vyzařování černého tělesa, lze využít znalosti Planckova vyzařovacího zákona. Ten definuje intenzitu vyzařování, závislou na vlnové délce a teplotě tělesa z čehož vyplývá, že černé těleso vyzařuje záření o různých intenzitách na různých vlnových délkách při dané teplotě. Planckův vyzařovací zákon je popsán rovnicí 2.16

$$M_{0\lambda} = \frac{2\pi h c^2}{\lambda^5 (e^{\frac{hc}{k\lambda T}} - 1)}, \tag{2.16}$$

kde $M_{0\lambda}$ je intenzita vyzařování černého tělesa, h Planckova konstanta ($h = 6,6256 \cdot 10^{-34}\, J{\cdot}s$), c je rychlost světla ve vakuu ($c = 2,9979{\cdot}10^8\, m{\cdot}s^{-1}$), λ je vlnová délka, T je teplota povrchu tělesa v Kelvinech a k je Stefan - Boltzmannova konstanta ($k = 1,3805{\cdot}10^{-23}\, J{\cdot}K^{-1}$) [7, 9].

2.4 Wienův posunovací zákon

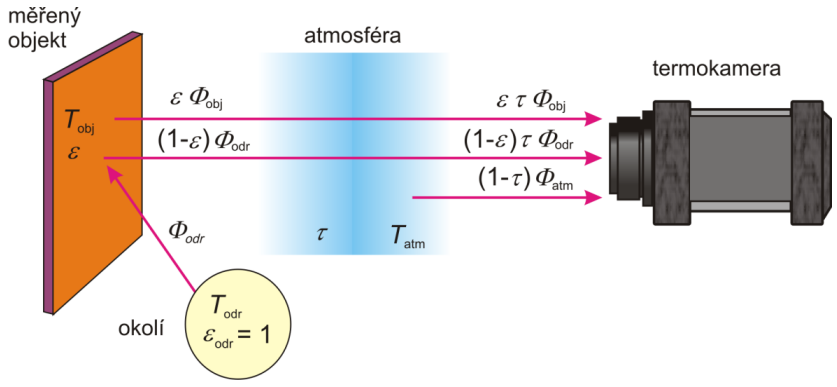
Maximální hodnota intenzity vyzařování tělesa je na různých vlnových délkách definována teplotou tělesa. Tuto závislost popisuje Wienův posunovací zákon 2.17. Zároveň lze pomocí tohoto vztahu určit hodnotu vlnové délky, při které je intenzita vyzařování tělesa maximální [7, 9]. Tento vztah lze vyjádřit jako

$$\lambda_{max} = \frac{b}{T}, \tag{2.17}$$

kde λ_{max} je vlnová délka, při které je intenzita vyzařování tělesa maximální, b je Wienova konstanta ($b = 2,898 \cdot 10^{-3} \cdot K$) a T je teplota povrchu tělesa [7].

2.5 Rovnice termografie

Měření za pomocí termografie se zabývá měřením teplotního pole na povrchu tělesa bezkontaktní metodou. Měření touto metodou má určité výhody, jako například možnost měření pohyblivých těles, či měření nebezpečných objektů z bezpečné vzdálenosti. Další výhodou, která nepřímo s takovým měřením souvisí, je vyhledávání tepelných stop v místech se zhoršenou viditelností. Samozřejmostí je, že i tato metoda má své nevýhody a omezení použití, jakožto i více ovlivňujících parametrů než například kontaktní měření teploty. Některými z ovlivňujících parametrů jsou veličiny popsané v předchozích kapitolách, a to emisivita (2.1.2), odrazivost (2.1.4), propustnost (2.1.5) a pohltivost (2.1.3). Tyto veličiny celkově definují jak zářivý tok vyzářený tělesem, tak tok odražený. Dalším faktorem při termografickém měření je vliv takzvaného vloženého media, tedy média, které se vyskytuje v prostoru mezi termokamerou a měřeným objektem. Tím je ve většině případů atmosféra. Vložené médium ovlivňuje měření několika jevy, jako například absorpcí, rozptylem, turbulencí a emisí záření. Tyto jevy jsou popsány v následujících podkapitolách a schematické zobrazení můžeme vidět na obrázku č. 5.



Obrázek 5: Vliv okolí na bezkontaktní měření teploty povrchu termokamerou

Fyzikální model, který je využíván při měření, bere v potaz tři základní toky záření, a to sledovaného tělesa, atmosféry a odražený tok z okolních zdrojů záření. Jejich součtem lze určit celkový zářivý tok (2.18), který je zároveň měřenou veličinou za pomocí termokamery. Tento součet se nazývá rovnicí termografie a lze ho vyjádřit jako

$$\phi_{celk} = \varepsilon\tau\phi_{obj} + (1-\varepsilon)\tau\phi_{odr} + (1-\tau)\phi_{atm}, \tag{2.18}$$

kde ε je emisivita povrchu sledovaného objektu a τ je propustnost atmosféry. Důležité je podotknout, že tento fyzikální model nezohledňuje vlastnosti optiky měřicího zařízení. Ta je považována za plně transparentní. Přestože optika vliv na měření má, ve většině situací je její vliv na měření zanedbatelný [11, 7, 15].

2.5.1 Absorpce

Absorpce je jev silně závislý na vlnové délce a složení vloženého média (atmosféry). Příkladem může být přítomnost oxidu uhličitého v atmosféře, který absorbuje záření mající vlnovou délku v okolí $15\mu m$. Dalšími ovlivňujícími složkami je například vodní pára, či ozon.

2.5.2 Rozptyl

Rozptyl je ovlivněn jednak molekulami plynu (Rayleighův rozptyl), jednak i aerosoly (Mielův rozptyl) přítomnými v atmosféře. Pokud srovnáme rozptyl záření ve viditelném spektru a ve spektru infračerveném, můžeme říci, že je tento rozptyl nižší. Toho se využívá například k vyhledávání osob v mlze či písečné bouři.

2.5.3 Turbulence

Turbulence je jev silně ovlivněný vzdáleností od měřeného objektu. Při tomto jevu dochází k rozmazání obrazu, jež je zapříčiněno silným větrem či při konvekčním přesunu tepla.

2.5.4 Emise

K emisi tepelného záření dochází obecně u všech těles s absolutní teplotou vyšší než 0 Kelvinů. Jelikož lze atmosféru díky své hmotě také považovat za těleso,tak i atmosféra emituje akumulovanou energii ve formě záření. V mnoha případech je emise atmosféry zanedbatelná, avšak je třeba ji vzít v úvahu v případě měření objektů s podobou teplotou jako je jejich okolí, kde se emise atmosféry signifikantně projeví.

2.6 Detektory infračerveného záření

Za detektor infračerveného záření lze považovat zařízení, které transformuje optické záření na měřitelný a definovaný elektrický signál. Hlavní podmínkou takového zařízení pak je určení závislosti mezi optickou veličinou a výstupním signálem. Pokud je tato závislost stanovena, lze z ní definovat intenzitu dopadajícího záření, a na základě ostatních parametrů a rovnice termografie (2.18) poté určit teplotu měřeného materiálu.

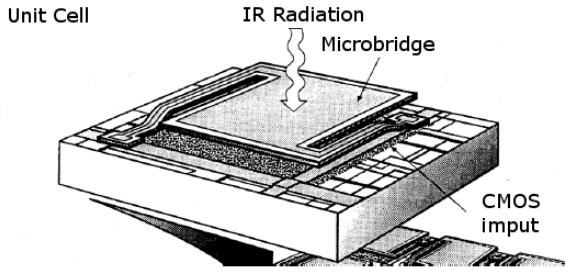
Na současném trhu se vyskytuje nepřeberné množství takových zařízení, ale většinu z nich lze rozdělit do dvou základních skupin. Těmito skupinami jsou zařízení využívající termální nebo kvantové senzory. Termální senzory jsou ve většině případů využity v termokamerách a fungují na principu ohřevu teplotně závislého materiálu, přičemž se mění daná měřitelná vlastnost, typicky rezistivita, a následně je převedena na elektrický signál pro další zpracování. Kvantové, nebo také fotonové, senzory generují volný náboj na základě dopadajících fotonů záření s danou energií. Tím se mění elektrické vlastnosti tohoto senzoru.

2.6.1 Termální senzory

Termální senzory pracují na principu změny svých vlastností na základě dopadajícího záření. Těmito vlastnostmi mohou být například resistivita, kapacitance nebo napětí. Jejich výhodou je, že jsou schopné fungovat za pokojových teplot a nepotřebují tedy pro správnou funkci externí chlazení. Mezi dva hlavní typy využívané v dnešních termokamerách patří bolometry a pyroelektrické senzory, jejichž vlastnosti jsou popsány níže.

Bolometr funguje na principu změny resistivity při změně teploty po dopadu infračerveného záření. Pokud budeme hovořímе o mikrobolometru, jedná se o jeden element pole, většinou umístěném na křemíkovém substrátu. Taková pole, složená z mikobolometrů, můžeme nazvat mikrobolometrickým senzorem, kde jeden element po vyčtení odpovídá jedné teplotní hodnotě obrazu na daném pixelu. Schematicky můžeme mikrobolometr vidět na obrázku č. 6.

6



Obrázek 6: Mikrobolometrický detektor [11]

Dalším typem termálních senzorů je senzor pyroelektrický. Jak již název napovídá, základem je pyroelektrický efekt. Ten spočívá v tom, že především u některých krystalických materiálů dochází vlivem změny teploty k mírné oscilaci atomů v krystalické mřížce, a tedy k dočasné polarizaci materiálu. U pyroelektrických materiálů lze detekovat změnu náboje při změně teploty, tedy určit elektrický proud, který je přímo úměrný této změně. Jelikož je senzor schopen detekovat pouze změnu teploty, a ne její absolutní hodnotu, je nutné pro tento senzor použít například rotující clonu, která bude periodicky zakrývat daný senzor, a tím získat konstantní generovaný proud senzoru, který odpovídá dané teplotě. Zároveň má tato clona pozitivní vliv na filtraci šumu okolí.[11, 10].

2.6.2 Kvantové senzory

Kvantové senzory jsou schopné generovat po dopadu záření náboj, a to díky absorpci fotonů. Ta nemá za následek nárůst vlastní teploty detektoru. Hlavní výhodou těchto senzorů je velmi rychlá odezva, a to řádově v mikrosekundách, a také jejich vysoká citlivost. Nevýhodou, a to hlavně z hlediska cenové dostupnosti, je nutnost chladit senzor na kryogenní teploty. K tomu lze využít například stirlingův nebo peltierův chladič. Díky chlazení se také u těchto senzorů redukuje tepelný šum. Další vlastností těchto detektorů je, že jsou často úzkopásmové, přičemž detekované pásmo vlnových délek takových senzorů je definováno volbou materiálu. [11, 10].

3 Návrh systému a použitá zařízení

V této kapitole je popsán návrh systému a popis zařízení, která byla použita pro jeho vytvoření. Jak již bylo zmíněno v úvodní kapitole, cílem této práce je navrhnout komplexní systém pro detekci požáru, který je schopný dlouhodobého běhu a detekce a záznamu rizika požáru. Dalším požadavkem byla implementace možnosti připojení až 16 kamer, a to v možné konfiguraci s motorem.

Pro tento systém bylo tedy třeba zvolit průmyslové PC s parametry, které zvládnou jak náročnější prostředí, tak vhodně zvolit výkonnostní parametry. Dále bylo nutné zahrnout do systému PTZ motor s jeho řídicí jednotkou a samozřejmostí je termální kamera SAFETIS, která je v případě venkovní montáže umístěna v krytu. V neposlední řadě byl kvůli úspoře kabeláže, a také vzdálenostnímu omezení sériové linky, ještě použit převodník mezi ethernetem a sériovou linkou.

Všechna tato zařízení jsou popsána v následujících kapitolách a jejich schematické zapojení je znázorněno na obrázku č. 7.

Obrázek 7: Schéma zapojení systému

3.1 Termokamera SAFETIS

Termokamera SAFETIS [20] (viz obrázek č. 8) je zařízením společnosti Workswell s.r.o. Jedná se o kameru, která kromě senzoru obsahuje řídicí desku založenou na platformě RiCoBoard, I/O kartu, která disponuje možností ovládání digitálních výstupů a v neposlední

řadě také TCP/IP rozhraní. Taktéž je možné ji využít jako samostatné zařízení bez nutnosti externího ovládání díky zabudovanému web-serveru.

Tato kamera má v sobě mikrobolometrický LWIR senzor, který pracuje ve spektrálním rozsahu $7.5 - 13.6\mu m$, a je možné zvolit mezi rozlišením $336x256px$ nebo $640x512px$ při rychlostech snímaní 9Hz nebo 30Hz. S tímto zařízením je možné měřit na rozsahu od $-25°C$ do $550°C$ případně s útlumovým filtrem a speciální kalibrací až do $1500°C$, a to s přesností $\pm 2\%$, případně $\pm 2°C$, a citlivostí až $50mK$, což odpovídá změně o $0.05°C$.

Obrázek 8: Termokamera SAFETIS

V aplikaci, kterou se zabývá tato práce, je kamera připojena za pomocí TCP/IP komunikačního protokolu, a to konkrétně na daných IP adresách určených sériovým číslem zařízení a dvou portech a to 2251 a 2252. Na základě tohoto připojení je kamera ovládána definovanou sadou příkazů, které jsou uvedeny v tabulkách č. 7 a č. 8 v příloze C. Těmito příkazy je ovládáno samotné měření a nastavení kamery, a to konkrétně sadou příkazů zaslaných na port 2251 a také radiometrický stream kamery, který je řízen implementovanou aplikací a probíhá na druhém portu 2252. Více o komunikaci se zařízením a jeho připojením do grafické aplikace je uvedeno v kapitole 6

3.2 Průmyslové PC Vecow IVH9000

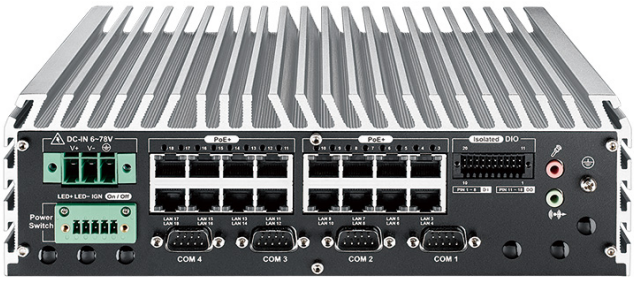
Pro tuto aplikaci bylo nutné zvolit průmyslový počítač, který bude odolný vůči ztíženým okolním podmínkám, především prachu a teplotám. Taktéž bylo třeba zvolit takové zařízení, které bude svými parametry vyhovovat náročnosti aplikace a zpracování dat z několika zařízení zároveň. Proto bylo zvoleno průmyslové PC od společnosti Vecow [22] a to konkrétně dvě možné varianty IVH9016 a IVH9008. Tyto varianty se liší pouze v počtu výstupních ethernetových portů a níže jsou popsány jejich parametry.

Výhodami použití tohoto zařízení, a tedy i parametry díky, kterým byl tento systém vybrán, jsou pasivní chlazení, a tedy eliminace zatížení systému prachem, operační teplota od

$-25°C$ do $70°C$, možnost vzdáleného startu celého systému, vestavěný SSD disk a možnost zapojení čtyř vyjímatelných pevných disků a také 16 ethernetových portů s technologií PoE (power over ethernet), čtyři sériové porty a 16 DIO pinů.

Výkonově byla zvolena varianta zařízení s procesorem Intel Core i7 šesté generace, 16 GB paměti RAM DDR4 s frekvencí 2133MHz, vnitřním 128GB SSD diskem pro operační systém Windows 7 a vlastní aplikaci a čtyřmi 2TB disky pro ukládání zaznamenaných dat.

Dle uživatelského manuálu je z 16 I/O pinů zařízení využito osm výstupních pro možné spínání ostatních periferií. Důvodem je fixní konfigurace pinů výrobcem na 8 vstupních a 8 výstupních. Pro ovládání byla použita interní knihovna dodaná společností Vecow, která povoluje přímé nastavení logické úrovně této periferii.



Obrázek 9: Průmyslové PC Vecow IVH 9016 [11]

3.3 Převodník Edgar

Převodník Edgar je zařízeni od společnosti Papouch s.r.o [19]. Toto zařízení umožňuje převod sériové komunikace RS232 nebo RS485 na komunikace Ethernetovou. V původním konceptu celého měřicího zařízení, navrženého pro tuto práci, bylo počítáno s propojením průmyslového PC 3.2 a řídicí jednotky ovládání motoru 3.4 za pomocí sériového rozhraní, avšak sériová komunikace má dle specifikací omezení na délku propojení 5m. Z tohoto důvodu jsme přistoupili na převod komunikace na ethernetovou. Díky tomuto kroku lze prodloužit délku propojení těchto dvou elementů. Další výhodou tohoto převodníku je možnost napájení pomocí PoE (Power over Ethernet). Přičemž Vecow IVH9000 má tuto technologii přímo podporovanou, a tedy došlo i k redukci počtu propojení mezi počítačem a měřicím systémem.

V původní verzi implementované aplikace bylo využito možnosti vytvoření virtuálního portu COM, který odesílal sériová data na statickou IP adresu daného převodníku. Z důvodu možnosti připojení až 16 měřicích modulů, a možnosti zjednodušení prvotní konfigurace, jsme dle specifikací zařízení Edgar přistoupili k revizi a využili možnosti definovat IP adresu převodníku a reimplemetovat připojení z virtuálního sériového připojení na protokol TCP.

Obrázek 10: Převodník Edgar [11]

3.4 PT motor s řídicí jednotkou

Pro virtuální zvětšení zorného úhlu kamery, či možnosti nahradit více zařízení jedním, byl do systému implementován takzvaný PT (Pan,Tilt) motor PTH300 [18] od společnosti Videotec S.P.A. Na tento motor (viz obrázek č. 11a) je umístěna kamera SAFETIS, a je tedy možné její natáčení ve dvou osách, a to na základě fyzických limitů motoru. Motor PTH300 byl taktéž zvolen z důvodu možnosti rotace o 330° horizontálně a 360° vertikálně, a to na základě nastavených limitů rotace a také díky stupni krytí IP66, což znamená krytí před nebezpečným dotykem jakoukoliv pomůckou, úplnému krytí proti prachu a ochraně proti intenzivně tryskající vodě.

Pro ovládání chodu motoru PTH300 byla použita řídicí jednotka DTMRX2 (na obrázku č. 11b) od stejného výrobce tedy Videotec S.P.A. Toto zařízení umožňuje propojení PT motoru s dalším zařízením, v našem případě průmyslovým PC 3.2 pomocí rozhraní RS232 nebo RS485. V případě této práce je využito rozhraní RS232, které je z důvodu možnosti potřeby větších komunikačních vzdáleností transformováno převodníkem Edgar 3.3 na posílání sériových příkazů přes ethernetové zapojení. Struktura těchto příkazů je definována podporovaným Pelco-D protokolem, který je popsán v následující sekci 3.4.1. Pro využití tohoto protokolu byla jednotka konfigurována pomocí fyzických přepínačů na základní desce, dle uživatelského manuálu. Taktéž byly nastaveny parametry sériové komunikace a zvolena varianta napájení jak samotné jednotky, tak i připojeného motoru. Výhodami této řídicí jednotky je stupeň krytí IP56, tedy oproti motoru je rozdíl pouze v částečném krytí proti prachu a dále možností jak manuálního nastavení pozic, tak nastavením přednastavených pozic. Hlavní nevýhodou je chybějící zpětná komunikace, a to hlavně z pohledu chybějící informace o aktuálním natočení motoru a informaci zda se motor aktuálně pohybuje.

(a) PTZ motor PTH300 [18]

(b) Řídicí jednotka DTMRX2 [17]

Obrázek 11: PTZ motor s řídicí jednotkou

3.4.1 Pelco-D

Protokol Pelco-D je jeden z řady protokolů vyvinutý převážně pro komunikaci s PTZ motory a v CCTV průmyslu. Jedná se o protokol používaný v sériové komunikaci a definuje formát a sadu příkazů včetně parametrů. Pro nastavení tohoto protokolu je třeba v sériové komunikaci nastavit osm datových bitů, jeden stop bit a žádnou paritu. Volba rychlosti sériové komunikace musí být zvolena dle zařízení, se kterým se komunikuje, což je v našem případě 2400Bd pro komunikaci s řídicí jednotkou DTMRX2, dle uživatelského manuálu. Samotná Pelco-D zpráva obsahuje 7 bajtů jejichž význam je:

- První bajt je synchronizační a vždy nabývá hodnoty FF v hexadecimálním kódu.
- Druhý bajt obsahuje adresu řídicí jednotky pro PTZ motor, který je zpravidla uveden v uživatelském manuálu.
- Třetí a čtvrtý bajt obsahuje data o ovládání (směr otáčení motoru, řízení PTZ bezpečnostní kamery a podobně).
- Pátý bajt je datový a nastavuje rychlost otáčení kolem svislé osy PTZ motoru.
- Šestý bajt je datový a udává rychlost otáčení kolem vertikální osy PTZ motoru.
- Sedmý bajt ma podobu kontrolního součtu a je to zbytek po dělení 100 ze sumy zaslaných bajtů vyjma synchronizačního.

Na bajty tři a čtyři lze pohlížet jako na příkazy. Každý bit ovládá jinou funkci kamery, například pokud je třetí bit čtvrtého bajtu nastaven na hodnotu 1 a je nastavena nenulová hodnota dat pro vertikální rychlost, motor se začne pohybovat danou rychlostí vertikálně vzhůru, a to do doby, dokud nezašleme příkaz na zastavení pohybu [1].

4 Popis implementačních nástrojů

V této kapitole jsou popsány softwarové nástroje použité pro vývoj aplikace. Jedná se o přehled informací o programovacím jazyku C#, dále popis grafického subsystému WPF a také analýza návrhového vzoru MVVM, dle kterého je aplikace implementována.

4.1 Vývojová platforma .NET Framework a programovací jazyk C#

Vývojová platforma .NET Framework, vyvinutá firmou Microsoft koncem 90. let, je kolekcí nástrojů, technologií a taktéž sadou programovacích nástrojů pro implementaci aplikací. Klíčovou komponentou .NET Frameworku je "Common Language Runtime"(CLR), což je prostředí starající se hlavně o běh programu a jeho kompilaci. Proces kompilace probíhá ve dvou fázích, nejprve je program přeložen do tzv. "Common Intermediate Language", jehož výstupem je kód společný pro všechny programovací jazyky v sadě .NET Frameworku, avšak není spustitelný. O překlad aplikace do spustitelného souboru se stará takzvaný just-in-time kompilátor, který kód převede na sadu příkazů podléhající operačnímu systému a architektuře počítače. CLR neplní jenom funkci kompilace, zároveň také obsluhuje za běhu aplikace její správu paměti, obsluhu výjimek nebo například správu vláken. Dalším nástrojem je "Common Type System"CTS. Ten obsahuje informace o datových typech jednotlivých jazyků ze sady a stará se o jejich vzájemnou kompatibilitu a přenositelnost informace.

Jak již bylo zmíněno, platforma .NET Framework obsahuje sadu programovacích jazyků, mezi něž patří například VisualBasic, C++, F#, JScript, J#, C# a další. Tím nejpokročilejším a nejrozsáhlejším z této sady je právě C#. Představen byl v roce 2000 a od jeho představení vzrůstala jeho popularita a momentálně je mezi jedním z 10 světových nejpoužívanějších programovacích jazyků. Jazyk C# je objektově orientovaný s přísnou kontrolou datových typů a nabízí jeho uživatelům běžné nástroje objektového programování, jako jsou dědičnost, zapouzdření, polymorfismus, překrývání metod a další [13, 6].

4.2 Popis Windows Presentation Foundation (WPF)

Windows Presentation Foundation neboli WPF je grafický subsystém sloužící pro vykreslování grafického rozhraní aplikací implementovaných pomocí jazyka C#. První verzí tohoto systému, která měla nahradit předchozí Windows Forms, byla uvedena v rámci .NET Frameworku 3.0 v roce 2006. K úplnému nahrazení nedošlo a předchozí Windows Forms je stále v některých oblastech využíván. Podsystém WPF lze rozdělit do 3 hlavních částí, a to na Presentation Framework, Presentation Core a MIL Core.

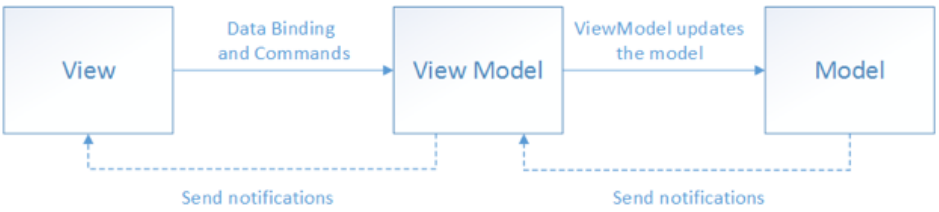
Presentation Framework obsahuje sadu definovaných komponentů pro vytváření GUI, jako například tlačítka, textová pole, zaškrtávací pole a podobně. Druhá část, tedy Presentation Core, obsahuje třídy a rozhraní klíčové pro jednotlivé grafické prvky. Poslední MIL Core (Media Integration Layer Core) zaobaluje DirectX pomocí obslužných tříd a umožňuje tak propojení CLR a DirectX.

Výhodou implementace grafického rozhraní za pomocí WPF je jednoznačné oddělení gra-

fické a funkční části. Grafická část je implementována za pomocí jazyka XAML (Extensible Application Markup Language), přičemž součástí je i obslužná třída obecně nazvaná "Code behind". V této třídě lze specifikovat, či rozšířit, funkcionalitu grafických prvků, případně spravovat události vyvolané grafickými prvky. Použitím této třídy ve smyslu přístupu k datům však narušuje princip oddělení grafické a programové části. Standardně se třída code behind používá příkladně pro obsluhu událostí, při kterých není třeba kontakt s daty aplikace, čímž může být přiblížení obrázku [14, 2].

4.3 Návrhový vzor MVVM

Pro návrh aplikací existuje několik návrhových vzorů, z nichž každý má své výhody a nevýhody. Těmito vzory jsou Model-View-Controler (MVC), Model-View-Presenter (MVP) nebo Model-View-ViewModel (MVVM). Tato kapitola je věnována popisu vzoru MVVM, jelikož je součástí této práce, a byl využit při návrhu a implementaci požadované aplikace.



Obrázek 12: Schéma principu návrhového vzoru MVVM [12]

Strukturu MVVM (viz obrázek č. 12) můžeme, jak již název napovídá, rozdělit na 3 prvky a těmi jsou Model, View a ViewModel. Modelem je objekt představující danou logickou část aplikace a modeluje její vybrané aspekty. Jsou zde tedy uvedena jak požadovaná data, tak obslužné a výpočetní funkce, které modelují chování. Část View je zodpovědná za grafickou podobu aplikace. V našem případě je pro tento případ využito tříd generovaných za pomocí WPF. Zde je pracováno s daty, která jsou zobrazena v grafice aplikace, jakožto jsou zde vytvořeny ovládací prvky samotné aplikace. V případě WPF je cíleno na separaci zobrazení od vlastních dat, tedy je zde snaha o nepoužití code behind třídy v co největším rozsahu. Poslední částí je takzvaný ViewModel, ten slouží k propojení již výše zmíněných částí. Pomocí něj jsou vyměňována data s Modelem a volány jeho obslužné funkce a je z něj obsluhována grafická část, případně grafické prvky volají funkce a předávají data do příslušného ViewModelu.

Pro propojení View s ViewModelem je využito principu datového bindingu. Obecně datový binding slouží k provázání dat a zobrazení bez přímého přístupu k daným datům z code behind. Z toho důvodu byly do základního objektu třídy System.Object přidány "DependencyObject"a "DependencyProperty". Principem poté je, že každý ovládací prvek GUI je potomkem třídy DependencyObject může registrovat jednotlivé DependencyPropetry. Ty jsou typem proměnné, jejichž hodnota je zapsána ve speciálním slovníku, který obsahuje jednotlivé DepencyPropery a jejich klíče a hodnoty. Klíčem těchto proměnných je jejich název a jejich hodnota je dynamicky vyčtena pomocí funkce "GetValue". Výsledkem toho provázání je nižší paměťová náročnost, a také možnost dědění. Obnova hodnot v grafickém rozhraní probíhá automaticky dle potřeby, nebo je možné využít volání specifické události RaiseProperty, která

vynutí okamžitou aktualizaci hodnoty v přiřazených prvcích GUI. Provázání lze také dále ovlivnit specifikací typu, které jsou uvedeny v tabulce č. 1

Název	Popis
Two way	Načítaní i zápis do zdroje.
One way	Pouze načítání ze zdroje.
One time	Načtení ze zdroje pouze jednou při inicializaci.
One way to source	Pouze zápis do zdroje.

Tabulka 1: Přehled možností provázání datového propojení

Pokud tedy shrneme výhody použití návrhového vzoru MVVM můžeme konkrétně hovořit o těchto:

- Separace grafické a funkční části, což vede k modulárnosti aplikace a možnosti spolupráce více programátorů na různých úsecích.
- Díky separaci a vlastnostem tohoto návrhového vzoru aplikace obsahuje kratší úseky kódu. Tedy je přehlednější a méně náchylná na chybu. Zároveň je poté jednodušší chyby odhalit
- Na základě stanovených principů a organizaci je výsledný kód přehlednější a lze se v něm lépe orientovat.

Za nevýhody můžeme považovat složitější syntaxi jazyka XAML, případně náročnější princip funkce datového bindingu [12, 3].

5 Základní popis aplikace pro detekci požárů

V této sekci jsou popsány základní parametry aplikace, implementované na základě zadání této práce. Dále se budeme věnovat obecnému popisu, šíření, instalaci a její struktuře. Funkce a parametry aplikace budou popsány v následující kapitole.

5.1 Parametry a popis aplikace

Tato aplikace byla vyvinuta v jazyce Visual C#, ve vývojovém studiu Microsoft Visual Studio 2017. Výhodou tohoto nástroje při implementaci byla možnost využití měření paměti a vytížení procesoru již v průběhu implementace. Tento software byl vyvinut jako desktopová aplikace pro operační systém Windows Vista a vyšší, a to z důvodu využití grafického subsystému WPF. Ten byl implementován v .NET Frameworku 3.0 a vyšším, a tím právě disponují tyto operační systémy. Konkrétně byl využit .NET Framework 4.5.2, který je tedy i prerekvizitou tohoto programu. Aplikace byla navržena dle návrhového vzoru MVVM, a to díky možnosti využít grafickou akceleraci a také pro umožnění využití vícevláknového přístupu, neboli využití více jader procesoru. Dalším důvodem byl předpoklad růstu aplikace, tedy byl při porovnání s modelem MVC model MVVM vyhodnocen jako přehlednější z hlediska modulárnosti a přehledu implementovaného kódu.

Z pohledu parametrů pro aplikaci, je předpokladem, že aplikace bude použita na 64 bitovém systému Windows, tedy je i pro tento systém sestavena. Důvodem byla paměťová náročnost aplikace z pohledu ukládání snímků do paměti, a tedy nutnost uvolnění co největšího množství paměti systémem. Z pohledu pevného disku je využitý 128GB systém dostačující pro samotnou aplikaci, která po instalaci dosahuje velikosti přibližně 40MB. Z pohledu náročnosti na výkon procesoru byla aplikace v průběhu běhu sledována a její vytěžování bylo určeno průměrně na 30% rovnoměrného vytížení všech jader.

5.2 Instalace a šíření aplikace

Předpokladem aplikace je její předinstalace na průmyslové PC Vecow, tedy její šíření je svázáno s tímto zařízením. Pro případné nově implementované funkce či opravy je možné stáhnout instalační soubor z webových stránek. Instalační soubor je vytvořen ve formátu ".exe", ve kterém jsou obsaženy kromě vlastní aplikace i prerekvizity, a to hlavně pro případ prvotní instalace, nebo potřeby reinstalace celého zařízení. Další výhodou tohoto souboru je, že není pro instalaci či update třeba internetové připojení zařízení, což u průmyslových počítačů instalovaných v provozu nebývá časté, a tedy lze provést update spuštěním souboru například z flash disku. Tento instalační soubor je obecně nazýván Bootstraper a jeho součásti lze rozdělit do dvou skupin. Obecně je jeho součástí samotný program a jeho prerekvizity, tedy součásti, na kterých je správný běh vlastní aplikace závislý. V tomto případě je jedinou prerekvizitou Microsoft .NET Framework 4.5.2 a tedy v tomto případě instalátor před instalací vlastní aplikace ověřuje, zda je nainstalován příslušný framework a zajišťuje jeho případnou doinstalaci. V případě potřeby lze využít i možnosti ověření architektury vlastního zařízení a dle toho nainstalovat buď 32-bitovou nebo 64-bitovou verzi dané aplikace. V našem případě

tato možnost využita nebyla, jelikož předpokladem celé aplikace je 64-bitová architektura a operační systém.

Instalátor vlastní aplikace byl vytvořen jako "msi"instalační soubor a druhý pomocný soubor s příponou ".cab", který slouží pro vykreslení grafiky při instalaci. Pro jeho vytvoření i následného Bootstrapperu byla využita sada nástrojů WiX [23] pro vývojové prostředí Microsoft Visual Studio. WiX je zkratkou pro "Windows Installer XML"a jedná se o volně šiřitelný plugin pro vytváření právě těchto instalačních souborů. Jeho výhodou je možnost upravit instalátor dle potřeb aplikace, například úpravou vizualizace či vložením licenčních podmínek, a zároveň tento instalátor poté využívá službu Windows Installer.

5.2.1 Instalace aplikace

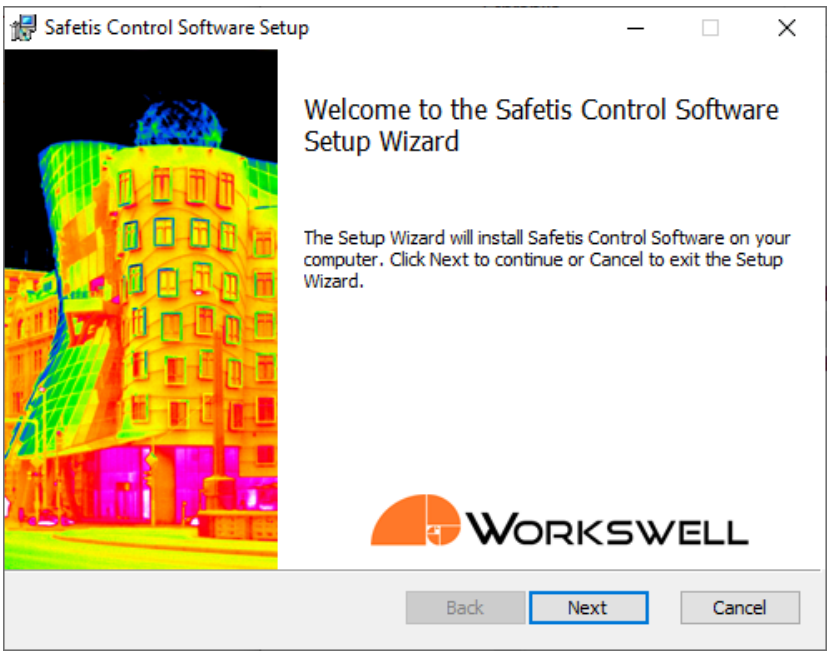
Instalace aplikace za využití Boostrapperu je rozdělena do sady navazujících dialogových oken, které byly vytvořeny úpravou dostupné standardní sady "WixUI_InstallDir". Po spuštění instalátoru je uživatel informován o následujících krocích, kterými jsou ověření a případná instalace prerekvizit a následná instalace samotné aplikace. Po potvrzení dojde k ověření potřebných nástrojů za pomocí ověření registrů. Pokud je třeba instalace nebo update prerekvizit, dojde ke spuštění zabaleného externího instalátoru (viz obrázek č. 13). Po instalaci prerekvizit je otevřeno okno instalátoru vlastní aplikace, kde jsou v prvním dialogu uvedeny obecné informace o instalaci. Následují licenční podmínky vlastnictví a šíření aplikace společnosti Workswell s.r.o., po kterých při potvrzení následuje výběr kořenového adresáře pro instalaci a možnost vytvoření zástupce na ploše a v menu Start. Posledním dialogem před samotnou instalací je dialog s možností návratu a změny předchozích zadání a potvrzením k instalaci. Samotná instalace slouží k vytvoření instalačních a potřebných složek pro spuštění, rozbalení souborů a k zápisu do registrů a vytvoření zástupců. O jejím průběhu je uživatel informován ukazatelem průběhu s informačním textem a je zakončen informací o dokončení instalace a jejím potvrzením.

5.3 Struktura aplikace a zdrojového kódu

V této kapitole je popsána struktura aplikace z pohledu spuštění, rozdělení grafického rozhraní na základě WPF a třídní hierarchie na základě návrhového vzoru MVVM. Jak již bylo uvedeno pro implementaci aplikace, bylo využito vývojové prostředí Microsoft Visual Studio 2017, včetně jeho nástrojů pro debuggování, sledování výkonu a správy projektu.

5.3.1 Struktura spuštění

Klíčovým souborem, a zároveň objektem pro spuštění aplikace, je "App", jenž je potomkem modelu WPF aplikace ("Application"). V jeho přidruženém souboru s kódem v pozadí "App.cs" je definován hlavní aplikační model a zároveň vytvořena instance objektu "ApplicationBase". Při vytvoření této instance objektu jsou ověřeny a načteny jazykové mutace programu, dále ověřeny a vytvořeny struktury složek potřebné pro následný běh programu a také



Obrázek 13: Náhled instalátoru aplikace

inicializována ikona celé aplikace. V dřívějších verzích se v této instanci dále řešilo taktéž licencování aplikace jako celku, které bylo poté reimplementováno. V neposlední řadě je zde vytvořena instance hlavního okna aplikace a taktéž jsou v následujícím sledu vytvořeny klíčové objekty hlavního objektu "ApplicationViewModelu" aplikace, které jsou jeho potomky.

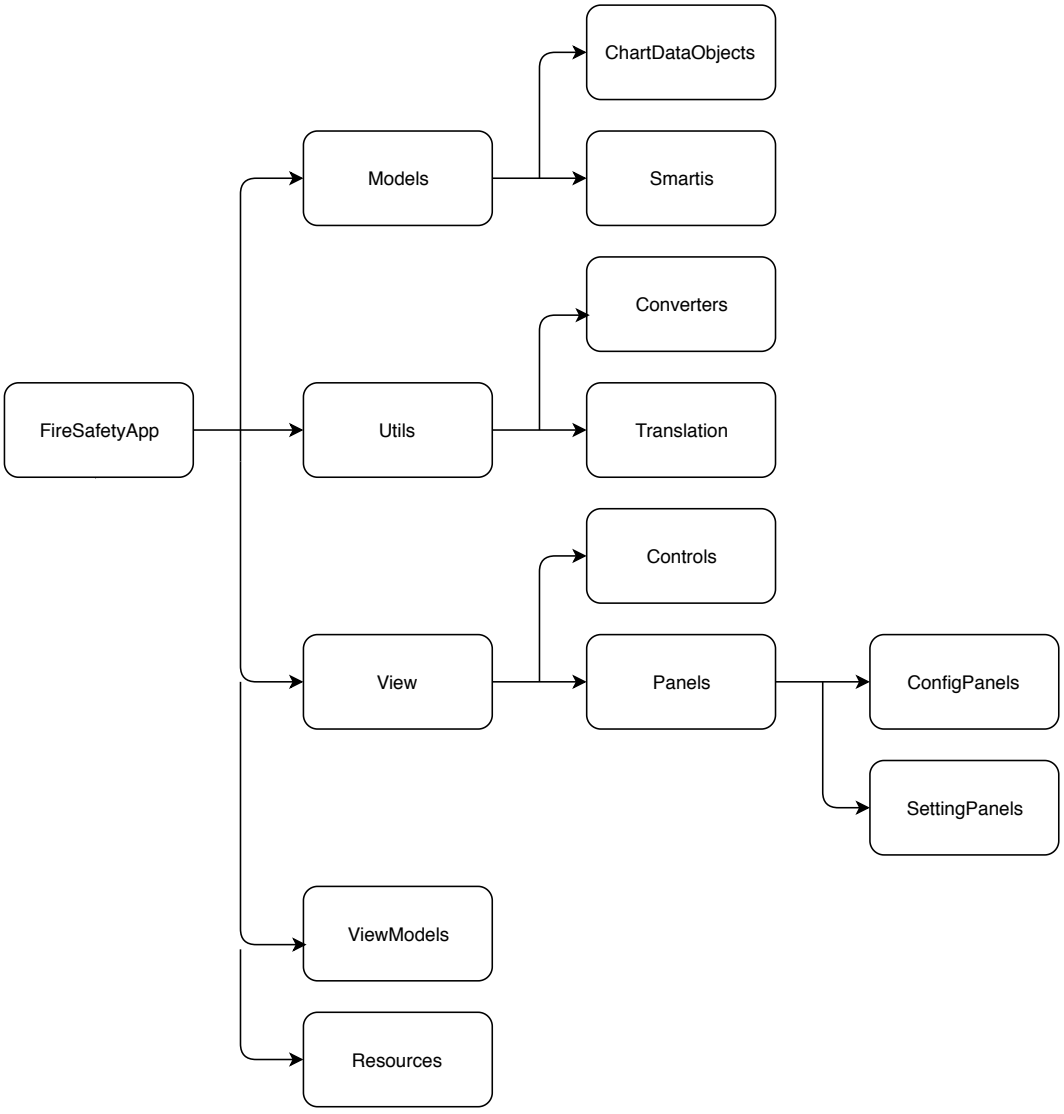
5.3.2 Struktura grafického rozhraní

Jak již bylo uvedeno, grafická část aplikace byla implementována s pomocí WPF. Pro implementaci některých grafických prvků bylo využito sady knihoven Telerik [21], které jsou "nadstavbou" WPF komponent. Respektive upravují jejich funkcionalitu a vizuální podobu. Grafické rozhraní aplikace lze rozložit primárně na hlavní okno, ve kterém jsou využité panely. Tyto panely mají dokovací funkci s možností uložení a resetování jejich rozložení. Díky této funkci si může uživatel přizpůsobit aplikaci dle svých potřeb, a to hlavně z pohledu přehlednosti. Výhodou tohoto rozhraní z pohledu implementace je možnost vytvoření vlastních grafických objektů, se kterými lze poté pracovat stejně jako s objekty knihovními, a které jsou poté vkládány do jednotlivých panelů. Kombinací těchto vlastností bylo docíleno modifikovatelného, přehledného a uživatelsky přívětivého rozhraní pro koncové zákazníky a uživatele aplikace.

5.3.3 Hierarchie zdrojového kódu

Při implementaci aplikace bylo cíleno na udržení přehlednosti a modulárnosti implementace. Z pohledu přehlednosti se jedná hlavně o třídění souvisejících tříd do vhodně pojme-

novaných složek, a to taktéž i kvůli vlastnosti vývojového prostředí, které na základě pojmenování těchto rozdělujících složek a jejich zanoření vytváří tzv. namespace, neboli obal pro určitou část zdrojových kódů. Jelikož úkolem pro implementaci této aplikace bylo využití návrhového vzoru MVVM, bylo logickým krokem vytvoření pojmenování složek na základě konceptu tohoto modelu. Jmenovitě se tedy jedná o složky s názvy Models, View, ViewModels a Utils jejichž členění můžeme vidět na obrázku č. 14. V projektu ještě můžeme nalézt složku Resources, která slouží pro uchování grafických objektů, jako jsou například miniatury loga, statické obrázky, či vlastní grafika ovládacích prvků. Samozřejmé je, že tyto jmenované složky jsou kořenové, a že dále dochází k dalšímu větvení na základě logických souvislostí mezi obsaženými prvky. Podrobnější mapa tříd je uvedena v příloze D. Pro dosažení modulárnosti programu byla jednak snaha o oddělení spolu nesouvisejících prvků kódu, jednak také o co největší možné využití kódu společného.



Obrázek 14: Členění aplikace do tzv. "namespace"

6 Popis grafického rozhraní aplikace a funkcí

Obsahem této kapitoly je konkrétní popis jednotlivých grafických komponent aplikace a popis jejich funkčnosti. V jednotlivých podkapitolách jsou dále popsány implementované nástroje a jejich funkce. Konkrétně se zde budeme věnovat správě uživatelů, přidání, připojení a konfiguraci kamer, ukládání dat aplikace jak pro nastavení, tak záznam, připojení PT motoru a jeho funkcím, grafům a logování událostí. V některých oblastech, kde je to potřebné, se dotkneme minulé implementaci, a to z důvodu jak výsledného porovnání, tak faktu, že v některých implementacích z testovacích důvodu zůstaly částečně implementované i funkce sloužící pro staré způsoby ovládaní aplikace.

6.1 Grafické rozdělení hlavního okna a popis komponent

V této sekci jsou popsány jednotlivé části grafického rozhraní z pohledu jejich ovládání, významu a funkce. Podrobněji budou jednotlivé funkce popsány v dalších sekcích věnujících se jejich implementaci. Celkový přehled grafického rozhraní je přiložen v příloze E. Při implementaci těchto panelů a oken bylo v případě oken s nastavením využito především tabulkového rozložení prvků "Grid Layout". V případě panelů a modulárních oken bylo využito takzvaných "Stack panelů" v kombinaci s tabulkovým rozložením. Hlavním důvodem pro využití byla možnost automatické úpravy velikostí jednotlivých prvků jak při změně rozlišení obrazovky, tak i při změně rozložení dokovatelných panelů

6.1.1 Hlavička a hlavní menu aplikace

První částí grafického rozhraní je hlavička celé aplikace. Zde, kromě loga, názvu aplikace a standardních ovládacích prvků aplikace pro minimalizaci, maximalizaci a zavření, stojí za zmínku implementované zobrazení statusu aplikace. Tento status slouží pro informaci o momentálním stavu aplikace a taktéž pro upozornění, pokud je přihlášený administrátor celé aplikace. Tyto stavy můžeme rozdělit následovně:

- OK - Standardní chování aplikace.
- DISSCONNECTED - Některá z kamer není připojená.
- ALARM - Některá z připojených kamer detekuje měřením nestandardní situaci.
- WARNING - LOW DISK SPACE! - Blíží se vyčerpání místa na záznamovém disku, bude přerušeno ukládání dat.

Za zmínku ještě stojí, že ve stavu aplikace "ALARM" je vizuálně, pomocí přerušovaného červeného pruhu, signalizována tato situace pro zlepšení orientace operátora.

Hlavní menu aplikace obsahuje v současné verzi dvě záložky, a to "Options" a "Application". Záložka "Options" obsahuje funkci pro navrácení rozložení aplikace, dále možnost vyvolání okna pro nastavení aplikace a okna pro správu a nastavení kamer a taktéž možnosti

práce se správou uživatelů, která je podrobněji popsána v kapitole 6.2. Pod položkou "Application" se skrývají funkce pro aktualizace aplikace a kamer, prohledávání sítě, otevření uživatelského manuálu a vyvolání "About" okna pro obecný popis aplikace.

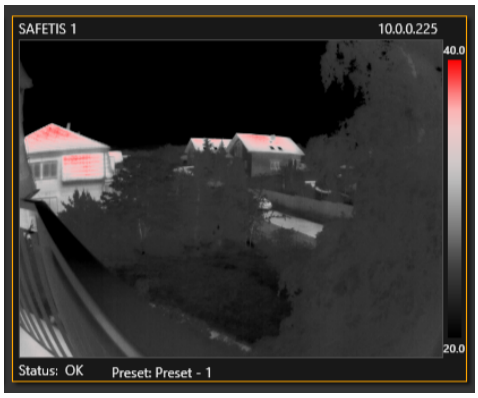
6.1.2 Informační panel

Informační panel je v základním rozložení aplikace ukotven v levé části. Slouží především pro práci s aplikací a kamerami, výběr jednotlivých kamer a zobrazení potřebných informací o dané vybrané kameře. Konkrétně se zde v horní části nachází ovládací prvky pro vyvolání okna přidání a nastavení kamer, pořízení manuálního snímku, nahrávání manuální radiometrické sekvence, konfiguraci měření kamery a pokus o znovupřipojení. Tyto funkce jsou spřaženy s vybranou kamerou.

Další částí je textový přehled kamer se zobrazením jejich stavů a možností jejich výběru a mazání. Na základě výběru, který je umožněn kliknutím na jednotlivé položky seznamu se aktualizují informace o vybrané kameře, které jsou dostupné a zobrazené přímo pod tímto seznamem. Konkrétně jsou zde zobrazeny informace o IP adrese vybrané kamery, sériové a artiklové číslo a verze firmwaru. Dále zapsaná aktuální měření v kameře s jejich aktuálními hodnotami a taktéž stavy digitálních výstupů kamery. Poslední komponentou tohoto panelu je možnost volby mezi automatickou barevnou paletou termálního obrazu a manuální, kde je možné nastavit fixní vizuální reprezentaci teplot v barevné škále.

6.1.3 Centrální panely hlavního okna

Centrální panel obsahuje několik záložek, ve kterých jsou obsaženy panely pro přehled různých funkcí aplikace. Těmto panelům lze díky dokování změnit polohu, případně je lze oddělit od aplikace a použít jako separátní okna.

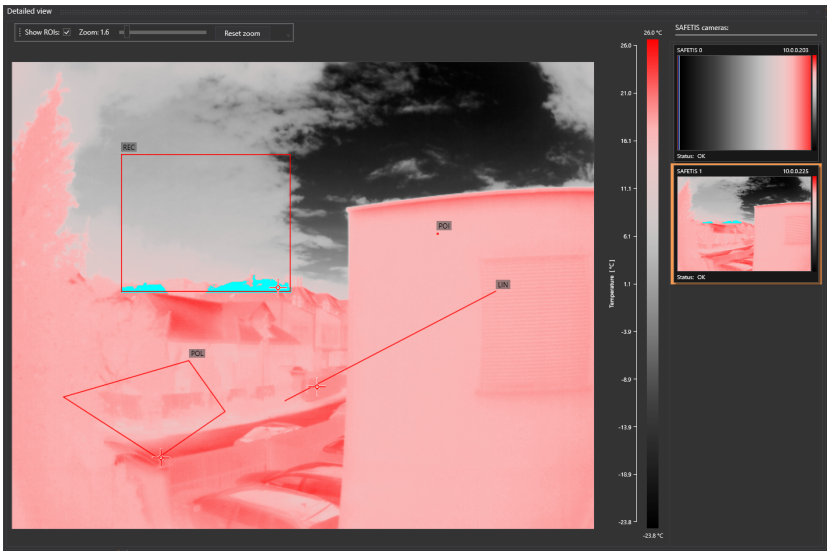


Obrázek 15: Miniatura kamery SAFETIS pro panel "Thermal Streams"

Prvním panelem je "Thermal streams", kde jsou zobrazeny miniatury vložených a připojených kamer (viz obrázek č. 15). Tyto miniatury zobrazují kromě obrazu kamer také jejich název, aktuální teplotu a IP adresu. Pokud se jedná o připojený systém s PT motorem, je zde taktéž zobrazena i aktuální pozice daného systému. Další funkcionalitou je možnost výběru kamery

levým tlačítkem myši pro zobrazení informací v informačním panelu a taktéž vizuální signalizace, opět červeným blikáním, v případě, že kamera zaznamená požadovanou měřenou situaci.

Panel "Detailed view", zobrazený na obrázku č. 16, slouží pro detailnější pohled na vybraný obraz z kamery. Na základě jakéhokoliv výběru je v levé části zobrazen zvětšený obraz streamu, který je možné přiblížit pomocí kolečka myši nebo příslušného posuvníku. Pro tento výběr taktéž slouží seznam miniatur kamer s aktuálním streamem implementovaný v pravé části. Tato část taktéž slouží pro zobrazení hranic měřicích nástrojů a vizualizaci pixelů, překračujících limitní hodnotu.



Obrázek 16: Panel s detailním pohledem na vybranou kameru SAFETIS

Dalším panelem v řadě je "Graphs", kde je implementovaný časový graf průběhu maximální teploty v čase z vybraných kamer. Z důvodu paměťových nároků lze vybrat časové úseky po kterých se graf automaticky začne přepisovat, zároveň je implementována redukce četnosti vzorků, dle vybraného časového úseku. Další možností přizpůsobení je manuální škálování osy grafu, případně změna barvy jednotlivých průběhů. Jelikož se jedná o aplikaci zkoumající prevenci zahoření, byly zvoleny pouze hodnoty maximálních teplot celého obrazu.

Poslední částí je takzvaný "Event logger". V této části je zobrazeno několik druhů vnějších událostí, které mohou nastat při běhu programu. Zároveň jsou tato data zaznamenávána fyzicky na interní disk PC a je tedy možné je exportovat. Data se ukládají textově do příslušné složky v dokumentech. Přestože se kvůli přehlednosti do aplikace načítá pouze posledních 200 záznamů, jsou data uložena v souboru všechna od prvního spuštění, až do fyzického přemazání tohoto souboru. Aplikace rozlišuje tři typy událostí, a to informační, varovné a situace alarmu. Přehled těchto událostí je uveden v tabulce 2.

Součástí tohoto záznamu je i časová značka události. Kromě těchto viditelných událostí je vytvořen vnitřní logovací systém zaznamenávající sled událostí a případných chybových hlášek, který je taktéž ukládán do textového souboru. V případě nestandardní situace programu je využíván pro rychlejší lokalizaci případné chyby.

Typ	Událost
Info	Spuštění aplikace
Info	Ukončení aplikace
Info	Přidání kamery
Info	Načtení dat z kamery
Info	Připojení kamery
Info	Znovupřipojení kamery
Info	Smazání kamery
Info	Změna na automatický rozsah
Info	Změna na manuální rozsah
Info	Změna konfigurace kamery
Info	Přihlášení daného uživatele
Info	Přihlášení daného administrátora
Info	Změna hesla administrátorem
Info	Odhlášení uživatele
Warning	Výpadek připojení na control portu
Warning	Výpadek připojení na stream portu
Warning	Neúspěšný pokus o znovupřipojení
Warning	Obecné upozornění při změně stavu kamery
Alarm	Detekování alarmové situace

Tabulka 2: Přehled zaznamenaných událostí

6.2 Správa uživatelů aplikace

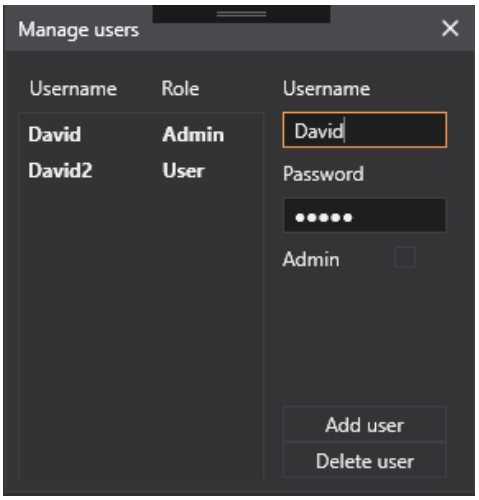
Jak již bylo zmíněno, hlavní funkcí aplikace je upozornit na vznikající riziko požáru, tedy předpokladem byla vizuální kontrola aplikace operátorem. Pro případné zjištění pochybení byl do aplikace implementován uživatelský režim, který slouží jak k blokování některých funkcí uživatelům a umožnění konfigurace aplikace pouze administrátorům, tak k záznamu osoby, která v danou dobu je zodpovědná za kontrolu aplikace.

Po spuštění aplikace je tedy současně se spuštěním hlavního okna vyvoláno přihlašovací okno uživatelů. Program je implementován tak, že neumožní přístup k žádným funkcím pokud není nikdo přihlášený a po odhlášení uživatele se automaticky aplikace vrátí do stavu zablokování a výzvy na zadání uživatelského jména a hesla. Pro prvotní přihlášení a testovací funkce je základní přístup nastaven jako admin-admin. Ten odblokuje přístup ke všem funkcím aplikace v takzvaném administrátorském režimu. V tomto režimu je možné konfigurovat připojená zařízení, jakožto i měnit samotné nastavení aplikace, nebo konfigurovat pozice a presety PT motoru.

Druhým režimem je režim operátorský. Zde je zablokován přístup ke konfiguracím a nastavením, a tedy lze v aplikaci pouze kontrolovat aktuální měření. Tento přístup byl zaveden i jako preventivní opatření před vědomou či nechtěnou změnou nastavení, která by mohla mít za následek selhání systému.

6.2.1 Administrace uživatelů

Správu seznamu uživatelů může provádět pouze administrátorský přihlášený účet, a to jednak v okně "Manage users" (viz obrázek č. 17), které je možné vyvolat z hlavního menu aplikace, v záložce "Options" pod položkou "Admin". Dalším nástrojem je možnost změna hesla libovolného uživatele v okně "Change password", které má stejné umístění jako předchozí zmíněné. Poslední položkou v podzáložce "Admin" v hlavním menu, je funkce na odhlášení aktuálního uživatele, která je společně s možností změny hesla přístupná i operátorům.



Obrázek 17: Okno správy uživatelů

V okně "Manage users" je uveden aktuální seznam vytvořených uživatelů s jejich jménem a rolí a možností přidáváním a mazáním uživatelů. Délka uživatelského jména musí mít minimální počet 5 znaků. Pro smazání jakéhokoliv uživatele je potřeba znalost jeho hesla, aby nedošlo k náhodnému smazání uživatele. V okně "Change password" má uživatel možnost při znalosti starého hesla změnit po jeho zadání své aktuální heslo. Tato změna je opět kontrolována na minimální počet znaků, jakož i na správnost starého hesla a shodě duplicitního zadání nového hesla.

V současné verzi jsou data o uživatelích uložena v dokumentech ve skrytém textovém souboru se změněnou příponou ".dat". Jelikož je při instalaci celého systému vytvořen a automaticky přihlašován po startu uživatelský účet Windows s omezenou možností konfigurace, a tedy nemožností změnit viditelnost složek a souborů, nebylo dosud přistoupeno k šifrování dat. Přesto je počítáno, a jsou připraveny funkce, na šifrování dat v souboru za pomocí statického klíče a systémových funkcí třídy "Cryptography" z namespace "System.Security". Tyto funkce pracují na principu PBKDF2 (Password-Based Key Derivation Function 2) a jsou schopné na základě klíče a šifrovaného streamu uložit, a poté otevřít a dekódovat data.

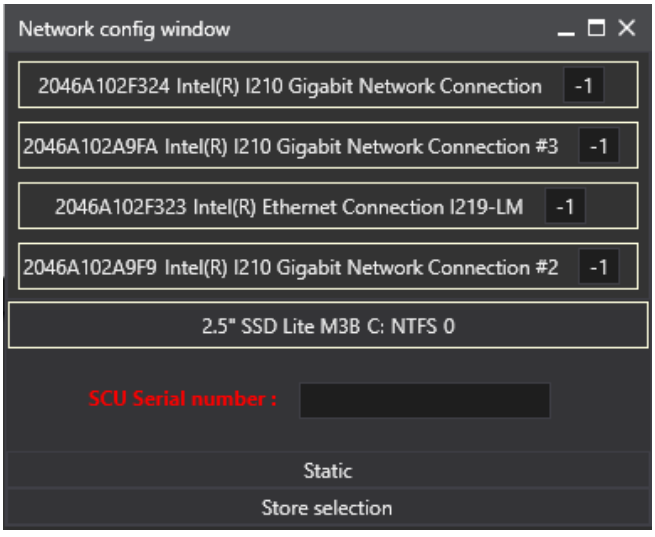
Z implementačního hlediska jsou data o uživatelích načtena v moment startu aplikace z daného souboru do seznamu objektů uživatelů. Během případných úprav, jako je změna hesla, či přidání nebo smazání uživatele, dojde po ověření správnosti všech náležitostí k aktualizaci objektů v seznamu a přepsání kompletního souboru. Data v souboru jsou přepsána okamžitě z důvodu možnosti okamžitého přihlášení nových uživatelů, ale hlavně k zachování

dat i při případném nestandardním chování aplikace. Pro zápis dat nebyla zvolena serializace, ale řádkový zápis s definicí separačních znaků pro jednotlivá potřebná data a následné inverzní vyčítání a separace.

6.3 Konfigurační režim aplikace

Konfigurační režim aplikace slouží pro nastavení párování ethernetových připojení k zásuvným modulům pro pevné disky. Zároveň slouží jako funkce pro detekci fyzické manipulace s aplikací a ošetření proti použití aplikace mimo stanovené průmyslové PC. V aplikaci je implementované konfigurační okno, zobrazené na obrázku č. 18, které lze vyvolat při přihlášení administrátora s definovaným jménem, a to "NetAdminWSW". Důvodem implementace konfigurátoru bylo zjednodušení úvodního nastavení aplikace. V okně konfigurátoru nalezneme seznam nalezených síťových zařízení s jejich ID hardwaru, názvem a polem pro zadání párovaného diskového slotu, a dále seznam nalezených disků s informací o ATA kanálu, který slouží právě pro toto párování. Další funkcí tohoto okna je možnost nastavení statických IP adres všem síťovým kartám.

Konfigurace propojení ATA kanálů a čtveřic ethernetových připojení byla implementována hlavně z důvodu cenové náročnosti SSD disků s velikostí 2TB. Proto byl zvolen pevný plotnový disk, a to s ohledem na rychlost zápisu. U tohoto konkrétního disku je maximální povolená rychlost zápisu pouze 140 MB/s. Pokud tedy budeme uvažovat kameru SAFETIS s rozlišením 640x512 pixelů a frekvencí 30 Hz, je teoretická rychlost pro zápis radiometrického videa 18,75 MB/s. Tento údaj je orientační, jelikož při záznamu se ukládají ještě další hodnoty, a ne jenom obrazové. Tedy pokud budeme počítat se 16 připojenými kamerami, které by nahrávaly současně, bylo by třeba využít disk s rychlostí zápisu minimálně 300 MB/s. Díky této vyšší potřebné rychlosti zápisu bylo přistoupeno k rozdělení ethernetových portů na čtveřice, přičemž ke každé z nich je přiřazen jeden ze čtyř dostupných zásuvných modulů.



Obrázek 18: Okno pro nastavení párování disků se síťovým připojením

Z pohledu implementace jsou před konfigurací prohledány všechny dostupné objekty typu

"NetworkInterface", přičemž pokud je daný objekt typu "Ethernet", je uložena jeho fyzická adresa a název do interního vytvořeného objektu. Pro vyhledání disků je využit "ManagementObjectSearcher", který na základě vstupního řetězce prohledává nástroje správy a vrací objekty typu "ManagementObject" s informacemi o daných zařízeních. Po nalezení aktuálně připojených disků jsou všechny uloženy interně do seznamu jako objekty typu "DriveItemViewModel" s parametry přiřazené písmeno disku, název disku, souborový systém a číslo ATA kanálu. Po nastavení čísla kanálu jednotlivým připojením a potvrzení je uložen fyzický soubor obsahující právě toto párování. Ten je při každém dalším startu otevřen a jsou načteny a ověřeny jeho data. Validace těchto dat probíhá stejným prohledáním, které bylo zmíněno, a porovnáním s uloženými daty. Pokud data nejsou validní, je programově zabráněno aplikaci ukládat data z kamer na disky, a uživatel je o tom upozorněn varovným oknem a statusem aplikace. Toto zabránění ukládání dat, jak již bylo zmíněno, slouží také jako ochrana před šířením na vlastní platformy.

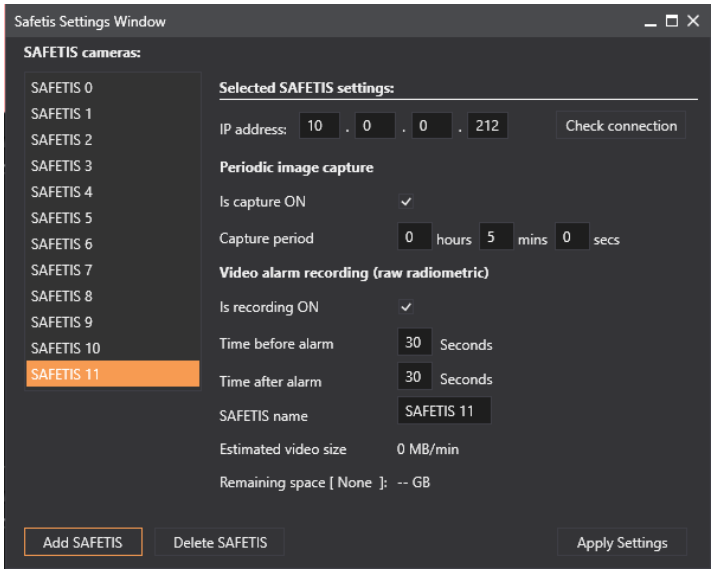
6.4 Práce s kamerami SAFETIS

Aplikace je implementována pro práci až se 16 kamerami, respektive kamerovými systémy, s PT motorem. Je k tomu přizpůsobeno grafické zpracování aplikace i limit počtu přidaných kamer. V této sekci se jednak budeme věnovat práci se seznamem kamer, jejich připojení a konfiguraci. Pro každou kameru je vytvořen objekt typu "SmartisViewModel" obsahující instanci modelu kamery SAFETIS, dle návrhového vzoru MVVM.

6.4.1 Přidání a mazání ze seznamu

Pro aplikaci je implementován seznam objektů typu "SmartisViewModel", nad kterým jsou prováděny operace spojené jak s přidáváním jednotlivých zařízení, tak prací s nimi. Pro práci s jednotlivými kamerami si aplikace v každém čase drží instanci jedné kamery jako vybrané. Tu lze zvolit již zmíněnými několika způsoby, například v levém informačním panelu. Původní koncepcí aplikace bylo plně manuální přidávání kamer v okně správy kamer. Tato varianta zde stále zůstala validní, a to hlavně z testovacích důvodů. Tedy kliknutím na tlačítko "Add SAFETIS" v okně "SAFETIS Settings" (viz obrázek č. 19) je automaticky vytvořen objekt s parametry uvedenými v tomto okně. Zde je možné taktéž instance kamer mazat ze seznamu. V pozdější fázi implementace bylo přistoupeno k automatickému přidávání kamer. To funguje na principu prohledávání sítě, a to na předem definovaných rozsazích adres. Původní zvyklostí byla přítomnost kamer na adresách 10.0.0.2 až 10.0.0.254, po konzultacích byla do webbrowseru aplikace implementována funkce na změnu IP adresy dle artiklového čísla. To je složeno z textového označení a číslem vyrobeného kusu počítaného od čísla 1000. Tedy toto číslo je fyzicky rozděleno na 2 poloviny a kamery v této konfiguraci se nyní nacházejí v rozsazích 10.0.10.00 až 10.0.19.99. Oba tyto rozsahy aplikace prohledává automaticky, přičemž v případě druhého rozsahu aplikace prohledává na posledním bajtu adresy pouze čísla v rozmezí 0 až 99. Hledání kamery je provedeno ve vlastním vlákně po spuštění aplikace nejprve pingem na danou adresu. Pokud je odezva pingu kladná, je vytvořeno dočasné TCP připojení za pomocí TCP klientu a je odeslán příkaz HIS na dané připojení. Pokud se na dané adrese nachází kamera SAFETIS, odpoví na tento příkaz OK a je automaticky přidána do seznamu

kamer. V negativním případě je vyhodnoceno jiné zařízení připojené na dané adrese a je tedy ignorována pozitivní odpověď na předchozí ping.



Obrázek 19: Okno nastavení kamer SAFETIS

6.4.2 Připojování a komunikace

Model objektu kamery SAFETIS v sobě z pohledu komunikace a připojení obsahuje instance obslužných tříd a to "SmartisTcpControl" a "SmartisTcpStream". Každá instance kamery tedy z pohledu připojení obsahuje dva TCP klienty, a to na dané IP adrese a masce podsítě a na dvou portech a to 2251 a 2252. První zmiňovaný port slouží pro konfiguraci a čtení dat měření a druhý slouží pro přenos obrazového streamu. Při úspěšném připojení TCP klienta si aplikace uloží instanci daného klienta zároveň s vyčtením a uložením instance objektu "TcpStream". Tomuto streamu jsou potom nastaveny časy pro vypršení komunikace a jsou na něj zapisována data daná formátem definované komunikace na základě seznamu příkazu viz tabulky č. 7 a č.8 a následným převedením textové zprávy na pole bajtů. Po odeslání dat je daný stream vyčištěn pomocí funkce "Flush" a je očekávána definovaná odpověď kamery. Na základě odpovědi je poté postupováno dále. Standardní odpovědí na příkazy typu set je potvrzení kamery OK v případě validního zpracování kamerou a v případě chyby je to zpráva ERR. V případě příkazů typu get jsou data převedena z pole bajtů na textovou zprávu to stringu a poté dle typu zprávy zpracována příslušnými implementovanými textovými parsery.

Možností připojení kamery je několik. Prvním z nich je v případě automatického nalezení kamery při spuštění a nebo nalezeném výskytu v konfiguračním souboru aplikace kamera připojena automaticky. Další možností je využití manuálního připojení kamery z okna "SAFETIS settings", případně pomocí tlačítka reconnect v horní části informačního panelu aplikace. V případě opětovného připojení je za podmínky nefunkčního alespoň jednoho ze dvou připojení ukončen a vyčištěn stream a ukončen klient TCP. Poté jsou zavolány znovu funkce na jejich připojení. Při implementaci a testování jsme narazili na problém se stabilitou daných připojení, proto je v aplikaci taktéž implementován systémový časovač s minutovou periodou.

Jeho funkcí je kontrola jednotlivých instancí kamer a ověření zda je zařízení stále připojeno. Pokud detekuje odpojení, automaticky se snaží o nové navázání spojení. Současně s tím bylo pro tuto variantu implementováno resetování připojení ze strany kamery v případě zastavení komunikace z druhého směru, a to s periodou 30 s. Hlavním důvodem pro implementaci resetování připojení z pohledu kamery byl předpoklad neustále komunikace.

Implementačně z pohledu komunikace jsou v aplikaci zavedeny dva časovače. První z nich obsluhuje vyčítání obrazu, přičemž jeho perioda je nastavena podle vyčtené hodnoty frekvence kamery. Ten poté s danou periodou obsluhuje nové snímky a jejich vyčítání. Zároveň má implementovanou toleranci na nevalidní snímky a případné odpojení streamu. Pokud funkce na generování obrazu ze zaslaných dat detekuje více snímků jak pětinásobek frekvence kamery za sebou, dojde automaticky k odpojení obrazového připojení. Druhý časovač obsluhuje jak vyčítání aktuálních měření kamery, tak získávání informací. Jeho perioda je fixně nastavena na 250 ms jelikož není vyžadována rychlejší odezva. Ve vnitřní implementaci volání se v každém cyklu data měřících nástrojů a digitálních výstupů kamery. Zároveň je zde implementován čítač volání a při každém 70. průběhu jsou vyčítány informace kamery především kvůli teplotě jádra, na základě které je zpřesňováno měření kamery výběrem vyčtených kalibračních hodnot. Taktéž je v tomto časovači implementován čítač problémů, tedy pokud během vyčítání dojde k neočekávané situaci, je ukončeno připojení stejným způsobem jako v předchozím případě. Pro aktualizaci dat ViewModelu je ve většině případů využito odchycení implementovaných událostí. Tato implementace funguje na principu pozorovatele, ten na základě definované generované události v našem případě reaguje na nově příchozí data a propaguje je z instance Modelu do ViewModelu, aby bylo možné provést grafickou změnu.

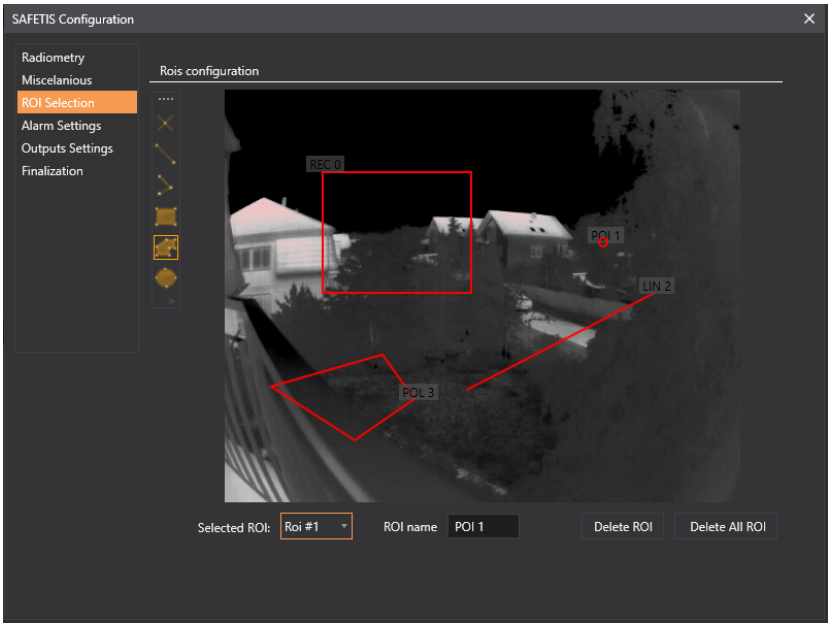
6.4.3 Generování obrazu

Obraz z kamery je generovaný po obdržení dat snímku v iteraci streamovacího časovače. Získaná data jsou ve formátu pole bajtů a prvním krokem je jejich převedení na pole hodnot typu ushort, jelikož jsou data ve formátu surových hodnot kamery tak zvaných RAW hodnot. Tato data jsou poté předána funkci obstarávající generování obrazu. Tato funkce na základě minimální a maximální hodnoty v obraze v případě automatického rozsahu nebo na základě limitních hodnot manuálního rozsahu nejprve vytvoří přiřazení barev teplotní palety k jednotlivým teplotním skokům. Tedy je nejprve vypočítán teplotní rozsah příslušný jedné z 255 barev teplotní palety. Na základě těchto dat je poté vytvořen obraz ve formátu "BitmapSource", který je poté graficky zobrazován jak v miniaturách, tak v panelu detailního zobrazení. Součástí generování obrazu je taktéž vizualizace pixelů, které splňují podmínku měření. Pro optimalizaci výkonu, bylo definováno 12 možných barev těchto pixelů. Jelikož aplikace slouží primárně pro detekci vysokých teplot, tedy blízkých hornímu rozsahu, jsou tyto barvy zafixovány programově v barvách palety a je jim přiřazený kód 1 až 12. Pokud je při vyčtení obrazu detekována teplota pixelu překračující mezní hodnotu, je tomuto pixelu v duplicitním poli změněna RAW hodnota na kod vybrané barvy a při přiřazení je tato hodnota kontrolována a automaticky je těmto pixelům přiřazována barva. Při zobrazení není tedy třeba využití překrývání obrazu a barvy pixelů jsou generovány přímo ve zdrojovém "BitmapSource", který je zobrazován. Pro ukládání dat je poté použito pole hodnot nedegradované o tyto data. Dalším prvkem, který je obrazově zobrazen, jsou měřicí oblasti ROI, které jsou z důvodu přehlednosti zobrazeny pouze v panelu detailního zobrazení. Tyto oblasti jsou

vykresleny za pomocí plátna ("Canvas") jako grafické objekty systémové třídy "UIElement". Na základě typu a jejich souřadnic jsou přidány do seznamu těchto objektů a vykresleny.

6.4.4 Konfigurace

V této podkapitole se budeme věnovat konfiguraci kamery z pohledu parametrů a měřicích nástrojů. V první řadě se jedná o grafickou strukturu a význam jednotlivých parametrů a poté vlastní implementaci. Pro konfiguraci kamery je implementováno okno "SAFETIS Configuration", které je rozděleno na několik panelů. Zde je postupným procházením a nastavením možné změnit aktuální konfiguraci. Prvním panelem je "Radiometry", zde je možné konfiguro-



Obrázek 20: Okno konfigurace kamery SAFETIS při definování měřících oblastí

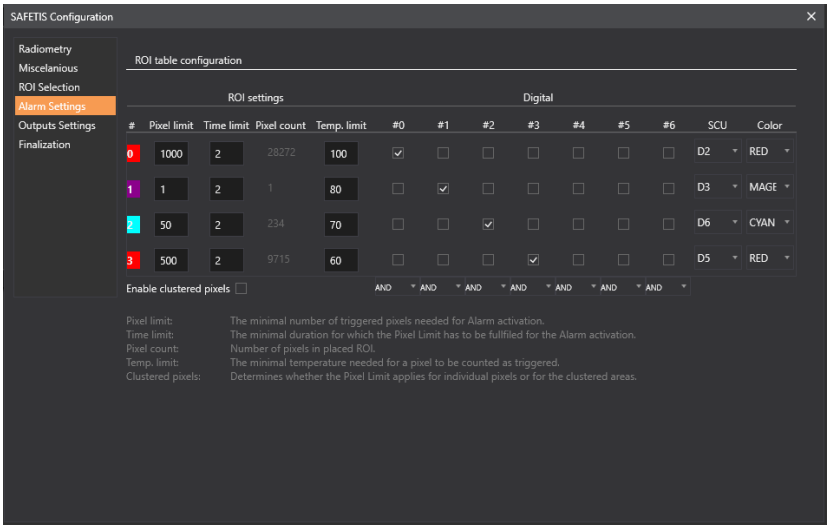
vat parametry kamery, jako je emisivita, odražená teplota a teplota atmosféry, teplotní rozsah a periodu interní kalibrace NUC. Panel "Miscelanious" slouží pro změnu IP adresy a masky podsítě kamery. Zároveň je zde možné definovat aktuální datum a čas kamery. V další sekci "ROI Selection" (viz obrázek č. 20) je možné graficky definovat měřicí oblasti takzvané ROI do obrazu kamery. V levé části lze vybrat požadovaný nástroj a za pomocí kreslení jsou uloženy příslušné parametry vybraného nástroje, dále je zde možné těmto oblastem definovat název a případně je mazat. S těmito nástroji se dále pracuje v následujícím panelu "Alarm Settings". V této sekci probíhá podrobnější nastavení parametrů měření přičemž význam jednotlivých položek je uveden v tabulce č. 3 a je graficky zobrazen na obrázku č. 21.

Poslední možností nastavení je zapnutí clusterovací funkce kamery, tedy nestandardní situace je vyhodnocena pouze pokud daný počet pixelů překračuje limitní teplotu po danou dobu, a zároveň jsou tyto pixely sousedící. V panelu "Outputs Settings" je možné definovat pro jednotlivé párované digitální výstupy kamery délku přesahu pulsu po skončení alarmu, a taktéž zda se jedná o puls s náběžnou nebo sestupnou hranou. Poslední sekce "Finalization" slouží

Název	Popis
#	Číslo měření
Pixel limit	Minimální počet pixelů potřebný pro spuštění alarmu
Time limit	Minimální doba po kterou musí být pixely aktivovány
Pixel count	Informace o počtu pixelů v měřícím nástroji
Temp. limit	Limitní teplota pro spuštění alarmu
#0 až #6	Jednotlivé digitální výstupy kamery pro přiřazení k měření
SCU	Jednotlivé digitální výstupy PC Vecow a jejich párování k měření
Color	Barva zobrazení pixelů, které překročily teplotní limit

Tabulka 3: Přehled konfiguračních parametrů

pouze k potvrzení a nahrání této konfigurace buď do kamery, nebo v případě zařízení s PT motorem, do vybrané konfigurace.



Obrázek 21: Okno konfigurace kamery SAFETIS při nastavení parametrů měření

Z pohledu implementace konfiguraci zařízení spravuje instance "SmartisConfigViewModelu" obsažená s hlavním aplikačním ViewModelu. Ja je inicializována jako prázdná při spuštění programu a vytváření jednotlivých ViewModelů a její data jsou naplněna v moment otevření konfiguračního okna vybrané kamery. Data, která jsou této instanci předána jsou porovnáním uložených a aktuálních dat z kamery. Po otevření a při změně parametrů jsou kontrolovány jednotlivé aspekty daných hodnot. Příkladem může bý test hodnoty emisivity jak na validní znaky, tak i na rozsah hodnot 0 až 1. V případě výběru a kreslení měřicích nástrojů, jsou vytvořeny a uloženy objekty typu "SmartisROI" a "SmartisRoiViewModel", v nichž jsou uložena data o neškálovaných bodech, typu nástroje a ostatních parametrů měření. Taktéž je vytvořen jejich seznam. Dále je vytvořen seznam digitálních výstupů kamery s jejich nastavením. Po nastavení všech parametrů a při aplikování do kamery je na základě volání přiřazeného příkazu datovým bindingem spuštěna sekvence příkazů. Nejprve je pozastavena komunikace na obou portech, poté je odeslán příkaz pro pozastavení měření kamery a následně příkaz definující začátek konfigurace kamery. Po provedení těchto kroků jsou sekvenčně odeslána všechna data nastavená během konfigurace, pomocí instance objektu

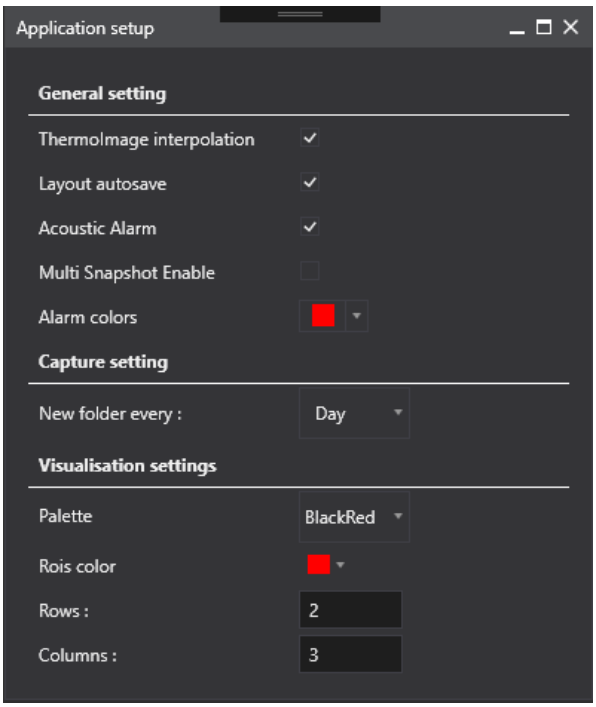
"TcpStream" připojené a vybrané kamery. Pokud odeslání proběhlo validně, je ukončen konfigurační mód kamery a jsou opět spuštěny časovače pro vyčítání obrazu a čtení dat. Pokud došlo ke změně IP adresy kamery, je automaticky přenastavena i aplikačně a je spuštěn příkaz na změnu připojení. Po dokončení jsou aktualizována data v objektu "SmartisSettings", která slouží pro ukládání konfigurace do souboru s nastavením ale i pro vyčítání konfiguračních dat při selekci a otevření tohoto okna. Kamera po změně konfigurace provede vnitřně dané změny a po provedení se automaticky spustí komunikace s programem.

6.5 Nastavení aplikace

Pro nastavení parametrů ovlivňujících samotnou aplikaci slouží okno "Application setup" na obrázku č. 22, které lze vyvolat z hlavního menu aplikace. Zde je možné najít následující parametry rozdělené do několika skupin. V sekci "General settings" lze nastavit interpolaci obrazu, tedy vykreslení hran pixelů nebo jejich rozmazání, dále "Layout autosave", který pokud je zapnut, slouží pro uložení fyzického souboru s rozložením panelů aplikace, který je následně vyvolán při následujícím startu aplikace. Další možností je zapnutí a vypnutí akustické signalizace alarmové situace při měření. Možnost "Multi Snapshot Enable" slouží k zobrazení výběru jednotlivých kamer, přičemž při výběru více než jedné je umožněno vytvářet snímky z těchto kamer jedním kliknutím. Poslední v této sekci je položka "Alarm colors", která slouží pro nastavení barvy vizuální signalizace alarmu. V části "Capture settings" je umožněna uživateli volba tvorby nových složek automatického záznamu v rozmezí dne, týdne a měsíce. Dále v sekci "Visualisation settings" nalezneme možnost nastavení teplotní palety obrazu, barvy hranic měřících nástrojů ROI a možnost nastavení počtu řádků a sloupců mřížky miniatur kamer.

6.5.1 Ukládání a implementace dat nastavení

Pro ukládání dat nastavení, a to nejenom aplikačních, ale i kamer a presetů PT motoru, jsou implementovány v aplikaci třídy "Settings" a "SettingsViewModel". V těchto třídách jsou držena data sloužící právě pro uložení do fyzického souboru a následné načtení. V třídě "Settings" nalezneme data aplikačního nastavení, ale také seznam objektů typu "SmartisSettings". Tyto objekty v sobě drží informace o přidaných kamerách, jakožto i v případě systému s PT motorem seznam těchto přednastavených pozic s konfiguracemi. Všechna data obsažená v třídě "Settings" a jejich obsažených instancích a proměnných jsou za pomocí instance třídy "StreamWriter" a pomocí systémové třídy "XmlSerializer" zapsána do fyzického XML souboru při standardním ukončení aplikace. V některých případech lze ignorovat zápis dat, respektive jejich serializaci, za pomocí direktivy "[XmlIgnore]", přičemž takto označená data nejsou uložena. Tento konfigurační soubor je při spuštění aplikace vyhledán, a pokud je přečten a je vyhodnocen jako validní, je díky jeho uložení pomocí serializace z něj automaticky vytvořena instance třídy "Settings" a naplněna daty obsaženými v tomto souboru. Není tedy třeba využití implementace parseru, jako například při ukládání konfiguračního souboru.

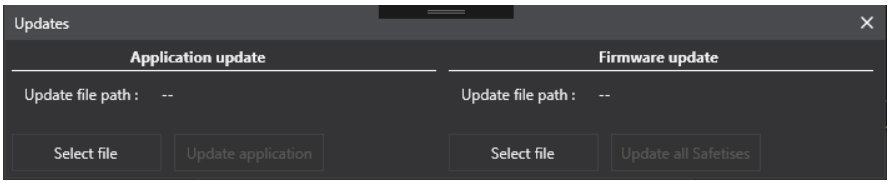


Obrázek 22: Okno pro nastavení aplikace

6.6 Aktualizace aplikace a kamer

Z důvodu předpokladu nemožnosti minimalizace aplikace a přístupu do systému Windows byla implementována možnost aktualizace jak aplikace tak firmwaru kamer. Pro tyto účely slouží okno "Update", které je možné vyvolat z hlavního menu aplikace a je zobrazeno na obrázku č. 23. Toto okno v sobě obsahuje obě varianty aktualizací.

Pro aktualizaci aplikace je možné stáhnout aktuální verzi instalátoru z webových stránek firmy Workswell s.r.o. Pro ověření nutnosti aplikace je soubor z webových stránek stažen s názvem podporující parsování textu aplikace. Při ověřování je kontrolována přítomnost klíčových slov "SCS" a "update", po kterých následují čísla udávající verzi instalátoru oddělených tečkami. Tato čísla se poté porovnávají s aktuálně nainstalovanou verzí aplikace, a pokud je instalátor vyhodnocen jako novější, je po kliknutí na tlačítko "Update application" aktuální aplikace ukončena a je spuštěn externím procesem tento instalátor. Po reinstalaci je vynuceno restartování celého počítače a díky nastavení automatického spouštění aplikace je spuštěna automaticky aktualizovaná, nainstalovaná verze.



Obrázek 23: Okno aktualizaci kamer a aplikace

Aktualizace firmwaru kamer probíhá podobným způsobem z pohledu ověření verze. Shodné je parsování názvu aktualizačního souboru s ověřením klíčového slova SAFETIS, přičemž následuje číslo verze firmwaru. Po vybrání a ověření validity souboru je možné provést aktualizaci firmwaru všech připojených kamer. Jelikož jsou zařízení držena v programu jako instance jejich objektů, je po potvrzení aktualizace procházen v aktualizačním vlákně jejich seznam a kontrolována aktuální programová verze kamer z nich vyčtená. Pokud načtený soubor obsahuje novější verzi, pozastaví se programově komunikace s kamerou a začne se definovaným způsobem odesílat aktualizační soubor. Nejprve je odeslán příkaz, který kameru upozorní na aktualizační proces, v němž je taktéž zaslána délka dat vyčtených z fyzického aktualizačního souboru. Následovně jsou odeslána datovým proudem "TcpStream" pomocí připojeného klienta všechna vyčtená data. Kamera po zaslání provede vnitřní aktualizaci a po dokončení se celé zařízení restartuje.

6.7 Záznam dat z kamery

Obecně aplikace podporuje několik typů ukládání dat. Můžeme je rozdělit na manuální a automatické ukládání, přičemž toto rozdělení je z pohledu implementace totožné, respektive využívá shodné funkce pro dané ukládání. Druhým dělením je ukládání radiometrických snímků, radiometrických sekvencí a případně i ukládání videa ve formátu ".avi". V této sekci bude popsáno obecně ukládání, v případě manuálního i jeho ovládací prvky a formát ukládaných radiometrických dat. Pro oba typy ukládání je automaticky aplikací generovaná stromová struktura adresářů. Jak již bylo zmíněno, kořenovým adresářem pro ukládání dat je párovaný disk s daným ethernetovým portem. Informaci o disku a odhadovanou velikost záznamu videa za minutu lze nalézt v okně "SAFETIS Settings". Dále jsou automaticky vytvářeny pro jednotlivé kamery příslušné složky s jejich názvem a interně jsou rozděleny na snímky a sekvence. V těchto složkách jsou umístěna data z manuálního ukládání, dále jsou zde taktéž vytvářeny složky pro automatický záznam, a to dle nastavení aplikace. Pro kontrolu složek a jejich vytváření je implementován časovač s hodinovou periodou, který kontroluje vytváření nových složek a zároveň je existence cesty kontrolována při každém spuštění aplikace a začátku ukládání dat.

Manuální ukládání snímků je umožněno za pomocí tlačítka v informačním panelu, taktéž pokud je v nastavení zapnuta možnost vytváření více snímků v jeden moment lze po vybrání požadovaných kamer pro tento typ snímání možné využít tuto funkci. Pro automatické periodické ukládání radiometrických snímků slouží nastavení periody taktéž v okně "SAFETIS Settings". Pro automatický záznam snímků je v aplikaci vytvořený objekt typu "CaptureViewModel", který kontroluje existenci složek, ale jeho hlavní funkcí je systémový časovač, který každou sekundu kontroluje zbývající čas do pořízení snímku a taktéž spouští funkce ukládání těchto jednotlivých snímků.

Pro manuální nahrávání sekvencí slouží taktéž obslužné tlačítko v informačním panelu aplikace, po jeho stisknutí se spustí nahrávání radiometrické sekvence z kamery a zobrazí se údaj o aktuální délce nahrávání. Automatické nahrávání sekvence je spuštěno pokud aplikace zaznamenaná alarmovou situaci, tedy se splní podmínky měření. Pokud je zjištěno probíhající manuální nahrávání, je před spuštěním automatického záznamu zastaveno. Více o implementaci a nastavení automatického nahrávání je uvedeno v sekci 6.7.2. V aplikaci je taktéž

implementováno ukládání ve formátu ".avi" za pomocí knihovny Aforge. Tyto funkce pro obsluhu záznamu jsou však v momentální verzi zakomentovány z důvodu testování ostatních záznamových funkcí a testování výkonnosti.

6.7.1 Formát ukládání radiometrických dat

Pro ukládání radiometrických dat je využita interní knihovna společnosti Workswell s.r.o, ThermoLib. Tato knihovna slouží jak pro načítání a ukládání radiometrických snímků, tak i sekvencí. Z pohledu formátu ukládání snímků se jedná o ".jpeg" s uloženými daty v APP1 hlavičce, které se nazývají fff file. V těchto datech jsou uloženy nejprve univerzální hlavičky jednotlivých tagů obsahující odkazy na příslušná data v APP1 hlavičce, po rozkódování těchto tagů lze poté přistupovat k jednotlivým datům, jako jsou teplotní parametry, RAW hodnoty obrazu, či další podstatné parametry, jako GPS nebo měřící nástroje ROI. Opačným způsobem poté dochází i k ukládání těchto dat. Pro ukládání radiometrických sekvencí ve formátu ".seq", respektive ".wseq" slouží implementovaná třída "ThermoVideo" v této knihovně. Z pohledu dat se jedná o soubor snímků řazených sekvenčně za sebe v datovém proudu s časovou značkou, tedy při práci se sekvencí jsou stejným způsobem data kódována jako v případě radiometrických snímků.

6.7.2 Alarmový stav aplikace

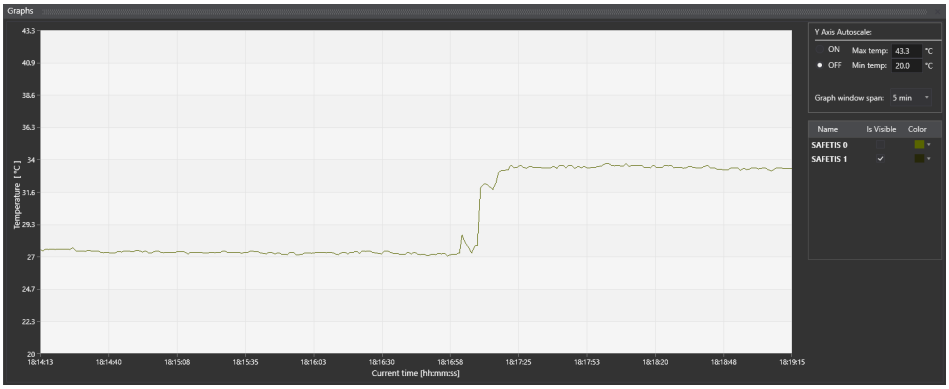
Jednou z hlavních funkcí aplikace je alarmový stav. Ten je vyvolán, pokud alespoň jedna z připojených kamer detekuje svým měřením nestandardní situaci, tedy splní nastavenou podmínku měření. Pokud taková situace nastane, dojde v aplikaci k několika událostem. Je spuštěna vizuální signalizace za pomocí blikání hlavičky aplikace a zvýraznění miniatury kamery, která je v alarmové situaci. Dále, pokud je povolena tato v funkce v nastavení aplikace, je spuštěna i akustická signalizace. Vizuální signalizace alarmu je znázorněna v příloze E na obrázku č. 31. Další událostí obslouženou při detekci alarmové situace je vyhodnocení měření, a na základě nastavených párování s digitálními výstupy průmyslového PC jsou za použití interní knihovny společnosti Vecow nastaveny logické hodnoty příslušným digitálním výstupům. Dále je aplikaci nastaven stav Alarm a je proveden záznam do aplikačního logu s časem a názvem kamery. Z pohledu blokace funkcí je znemožněno jakékoliv nahrávání z dané kamery. Pokud je spuštěno manuální nahrávání je automaticky ukončeno. Dále je zablokován přístup ke konfiguraci zařízení, aby nemohlo dojít k jeho změně. Taktéž se spustí automatické nahrávání radiometrického záznamu. Ten je možný nastavit v okně "SAFETIS Settings", a to konkrétně hodnoty přesahu nahrávání jak před danou událostí, tak po ní.

Z implementačního pohledu je automatické ukládání radiometrického záznamu omezeno úvodním během kamery. Po připojení kamery a spuštění streamu jsou postupně ukládány jednotlivé snímky do bufferu až do velikosti dané nastaveným časovým přesahem a frekvencí kamery. Po naplnění tohoto bufferu je povolen automatický záznam, a také dochází k postupnému současnému mazání prvního snímku a přidávání aktuálního snímku na konec. Tedy je implementovaná FIFO fronta pro snímky, pro každou kameru. Pokud dojde k alarmové situaci je spuštěno vlákno pro nahrávání právě těchto snímků, toto vlákno není nijak časově omezeno tedy zároveň dochází k vyprazdňování fronty pro nahrávání a přidávání nových

snímků ze streamu kamery. Jelikož ukládací vlákno není frekvenčně omezeno, dojde k uložení předchozích snímků rychleji a fronta se vyprázdní. Po vyprázdnění bufferu je předána kontrola nad nahráváním přímo časovači vyčítaní snímků z kamery a každý nový snímek je ihned nahráván do dané sekvence. Po skončení alarmové situace je spuštěn časovač nahrávající danou sekvenci ještě po dobu nastavenou taktéž v okně "SAFETIS settings", taktéž pokud je v této periodě opět dosaženo alarmu, automaticky je navázáno na předchozí nahrávání a nedojde k ukončení záznamu.

6.8 Záznam grafického průběhu teploty

Jak již bylo uvedeno, do grafického záznamu se ukládají aktuální maximální měřené teploty kamery. Pro grafické zobrazení je využitý kartézský graf "RadCartesianChart" knihovny Telerik, který je zobrazen na obrázku č. 24. Ten je propojen datovým bindingem se seznamem podporovaných objektů typu "LineSeries" ve třídě "GraphsPaneViewModel". Pro každou kameru je vytvořený objekt "LineSeries" a přidaný do tohoto seznamu. Pro ukládání dat je použit seznam objektů typu "ChartDateTemperatureObject", který v sobě uchovává informace o časové značce a zadané teplotě. Tato data se obnovují po uplynutí interně nastavené periody systémového časovače a jsou přidána do daného seznamu.



Obrázek 24: Panel grafického záznamu maximální teploty

Pro přehlednost a omezení vytížení systému byly definovány jak časové úseky po jejichž uplynutí dochází k odebírání nejstaršího prvku, tak počet prvků v dané kolekci. Pokud dojde uživatelem ke změně daného okna, jsou všechna data z kolekcí odstraněna, a to z důvodu možného nejednotného zobrazení na časové ose. V tabulce č. 4 jsou uvedena jednotlivá časová okna s jejich periodou a maximálním počtem záznamů. Pro nastavení osy y grafu je při každém záznamu testována globální maximální a minimální hodnota teploty. V případě, že nová hodnota přesáhne aktuální limity, je automaticky přepsána osa y. Pro zpřehlednění jsou limity posunuty o 5 % aktuálního rozsahu grafu, případně je možné nastavil limity manuálně fixní.

Časové okno	Počet vzorků	Perioda
10 s	100	10 ms
30 s	100	30 ms
1 m	250	240 ms
5 m	250	1.2 s
1 0m	250	2.4 s
30 m	250	7.2 s
1 h	500	7.2 s
6 h	500	43.2 s
12 h	1000	43.2 s
24 h	1000	86.4 s

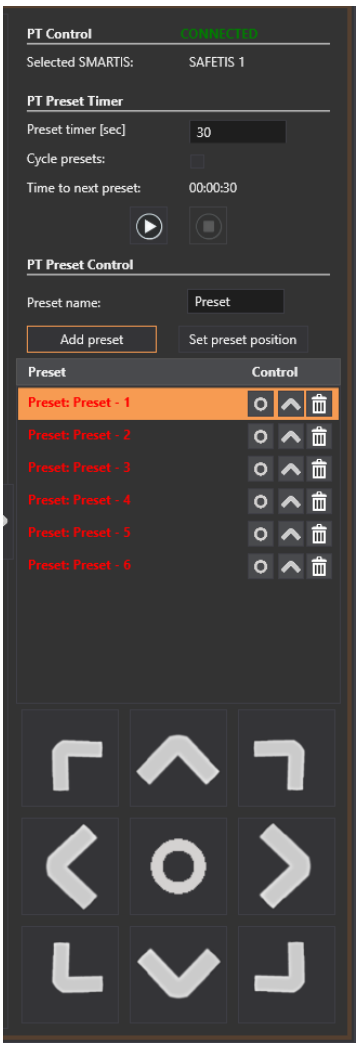
Tabulka 4: Popis časových úseků grafu

6.9 Ovládání a funkce PT motoru

Jak již bylo uvedeno, PT motor, respektive jeho řídicí jednotka, je do průmyslového PC Vecow připojena pomocí převodníku Edgar RS232 na Ethernet. IP adresa pro připojení k tomuto převodníku je nastavena při konfiguraci zařízení, a to na základě artiklového čísla kamery SAFETIS. Toto nastavení je obdobné, na základě čtyřmístného čísla, které je rozděleno fyzicky na dvě poloviny, přičemž k první polovině je přičteno 100. Tedy pro příklad, je-li artiklové číslo kamery WSO-PT-1019, bude IP adresa kamery 10.0.10.19 a převodníku 10.0.110.19. Detekce zda zařízení obsahuje řídicí jednotku s PT motorem je taktéž provedena na základě artiklového čísla kamery, a to konkrétně ověřením klíčového slova PT, respektive WSO-PT. Pokud je přítomnost těchto klíčů vyhodnocena jako pozitivní jsou pro danou kamery odemčeny a zobrazeny po jejím vybrání obslužné grafické prvky, a taktéž dojde k automatickému pokusu o připojení TCP klienta na adresu danou artiklovým číslem kamery.

Grafickými ovládacími prvky jsou především možnost aplikování konfigurace pomocí daného tlačítka v konfiguračním okně vybrané kamery a ovládací panel PT motoru v pravé části hlavního okna celé aplikace (viz obrázek č. 25). V ovládacím panelu je zobrazena informace, zda je komunikace s řídicí jednotkou připojena, a taktéž název aktuálně vybraného zařízení, které je ovládáno. V další části se nachází ovládací prvky cyklovače presetů, a to nastavení času setrvání v pozici, zapnutí cyklování, zobrazení aktuálního času zbývajícího do přejezdu a ovládací prvky pro spuštění a zastavení cyklovače. V momentální verzi jsou ještě zakomentované prvky pro manuální vynucení přejezdu na předchozí nebo další pozici. V poslední části se nachází ovládací prvky pro přidání přednastavené pozice a nastavení jeho pozice a možnost změny názvu, dále seznam těchto pozic a manuální ovládání motoru.

Z implementačního pohledu je manuální ovládání motoru obsluhováno za pomocí odchycení dvou událostí, a to zmáčknutí a uvolnění daného tlačítka pro pohyb. Při stlačení je odeslán přes dané připojení příkaz ve formátu Pelco-D pro spuštění pohybu v daném směru. Po dobu stisku daného tlačítka se motor otáčí buď do nastaveného fyzického limitu, nebo do uvolnění, kdy je odeslán příkaz k zastavení. Prostřední tlačítko slouží pro vyvolání takzvané základní pozice, kterou lze po podržení tohoto prvku po dobu alespoň 2 sekund změnit na aktuální nastavenou pozici. Jednotlivé presety jsou vnitřně uloženy jako objekty typu "Pre-



Obrázek 25: Panel pro ovládání PT motoru

setViewModel", kde hlavním údajem, který se používá při komunikaci s řídicí jednotkou, je identifikační číslo daného presetu, které udává zároveň i pozici. Dále zde můžeme nalézt informace o názvu, signalizační barvě či instanci nastavené konfigurace dané pozici. Tyto objekty jsou poté přidávány do listu, který je dále obsluhován především pro cyklování a je zobrazen v již zmíněném seznamu. Nastavení konfigurace kamery pro pozici je provedeno v konfiguračním okně a při jeho provedení se přiřadí instance aktuální konfigurační třídy vybranému presetu. Ta je uchována a vyvolána při výběru, a také je poté uložena do souboru s nastavením pro možné načtení při restartu programu. Pro cyklování mezi pozicemi je vytvořen v aplikaci systémový časovač, který po vypršení nastavené doby cyklování odešle příkaz řídicí jednotce o změně pozice a pokud je nastavena konfigurace kamery, spustí nahrání tohoto nastavení. Jelikož není možnost z této řídicí jednotky získat informaci o natočení motoru či signalizaci, zda je motor stále ještě v pohybu, je po odeslání všech příkazů spuštěn další časovač, který má momentálně nastavenou fixní periodu 15 s. Ten slouží pro zamezení spuštění měření kamery během přejezdu, a tedy i zamezení falešných alarmů. Tato perioda byla zjištěna jako maximální doba přejezdu mezi krajními pozicemi motoru.

7 Závěr

V této finální kapitole jsou diskutována témata týkající se testování aplikace jak na uživatelské tak implementační straně, dále uvedeny a diskutovány možná vylepšení stávajícího řešení a taktéž jsou zde zhodnoceny dosažené výsledky oproti zadání této diplomové práce.

7.1 Výsledky testování aplikace

Testování aplikace probíhalo jak paralelně při vývoji, tak u vybraných koncových uživatelů buď simulovaně, nebo v provozu. Na základě těchto testů došlo jak k opravám chyb, které se při používání vyskytly, jakožto k reimplementaci či vylepšení některých nástrojů používaných v této aplikaci. Obecně se jednalo o chyby plynoucí z nastavení parametrů, či chybějících ošetření nestandardních situací.

Paralelně s testováním samotné aplikace probíhalo i testování aplikace z pohledu stability připojení s kamerou, testy unikátních situací, jako například odpojení kabelů, a podobně. Tyto testy byly prováděny ve spolupráci s kolegy z týmu implementace firmwaru a byly diskutovány jednotlivé problémy a jejich řešení spojené s vnitřní implementací kamer. Konzultace se týkaly i stanovení sady příkazů a jejich doplňování pro potřeby aplikace. Příkladem je uvedený časovač testu připojení a resetování připojení v interní implementaci kamery.

Jelikož je pro tento systém klíčová stabilita programu, byly taktéž současně prováděny na několika průběžných verzích i dlouhodobé interní testy s menším počtem reálných zařízení, které měly odhalit nestability programu při záznamu dat nebo při prostém datovém streamu, a to především z pohledu náročnosti na paměť RAM. Pro tyto testy bylo hlavním důvodem využití bufferu pro záznam streamu před nestandardní situací, a tedy ukládání této sady přímo do paměti.

Poslední testy dlouhodobého běhu byly provedeny rámcově po dobu 14 dní a neprokázaly paměťový únik aplikace. Taktéž byla aplikace sledována na paměťové úniky pomocí vývojového studia interním nástrojem Memory Profiler. Ten však má omezenou dobu běhu na 450 minut, přičemž během tohoto testu se v poslední verzi taktéž neprokázaly paměťové úniky.

Testy aplikace s maximálním možným počtem kamer nebylo možné provést, a to z důvodu nedostatečného počtu daných zařízení. Maximální počet reálných kamer připojených stabilně v poslední verzi aplikace bylo 5 a dále byl běh aplikace stabilně simulován na osmi základních deskách kamery SAFETIS s nahraným firmwarem a implementovanou simulací obrazu.

7.2 Diskuze možných vylepšení

Navržený systém sestavený a implementovaný v rámci této práce je pilotní verzí. Přestože jsou nástroje a funkce v momentální verzi funkční a prošly sérií testů, včetně prvotního nasazení do provozu, je zde stále možnost vylepšení a implementace nových funkcí, jakožto i případných oprav. Z důvodu projeveného zájmu se předpokládá budoucí reimplementace některých nástrojů, jakož i jejich rozšíření.

Konkrétně prvním navrhovaným vylepšením je volba jiné řídicí jednotky s PT motorem. Hlavním důvodem je nemožnost vyčtení aktuální pozice motoru, případně informace o pohybu. To znemožňuje časovou optimalizaci změny přednastavené polohy. Dalším vylepšením je doplnění viditelné kamery z pohledu HW, ať již do stávající kamery SAFETIS, případně volba jiné platformy.

Nespornou výhodou této aplikace je použití kamery vyráběné stejnou společností jako celý systém, aplikace tedy není závislá na externím SDK a je tedy možná případná optimalizace či vylepšení na straně firmwaru kamery. Nad rámec této práce byla provedena implementace SDK obsluhující kameru SAFETIS, a proto jedním z navrhovaných vylepšení je i integrace tohoto SDK do implementace stávajícího softwaru. Z pohledu ukládání radiometrického i neradiometrického avi záznamu, by z důvodu paměťové optimalizace, případně i možnosti prodloužení doby záznamu před událostí, mohla být implementována redukce frameratu výstupní sekvence. Hlavní důvod pro tento krok je fakt, že děje při případném zahoření nejsou rychlé a proto byla navrhována rychlost 2Hz. Z grafického pohledu je třeba v aplikaci provést revizi některých výpočtů. Ty se v průběhu implementace zesložitily a momentálně zatěžují grafické vlákno aplikace a znemožňují obnovu například některých progress barů. Z pohledu rozložení je ještě doporučeno umožnit ovládání PT motoru taktéž v panelu detailního zobrazení a revidovat správu uživatelů.

7.3 Zhodnocení dosažených výsledků

V úvodní části této práce, zaměřené na návrh systému pro detekci požáru za pomocí termokamery SAFETIS, je zpracováno seznámení a teoretický pohled na principy měření teploty bezkontaktním způsobem za pomocí mikrobolometrických kamer a obecný pohled na principy termografie.

V další části této práce jsem se věnoval návrhu systému a jednotlivým komponentům, včetně jejich zapojení, implementaci jejich funkcí a jejich vzájemným propojením. Tyto komponenty jsou uvedeny v seznamu přístrojů jakožto i finální návrh zapojení celého systému. Na základě uživatelských manuálů a datasheetů bylo toto schéma realizováno a testováno na jeho funkčnost a odolnost, jak je shrnuto v předchozích kapitolách.

Pro implementaci aplikace v jazyce C# byla provedena obecná analýza jazyka a jeho nástrojů, jakožto i seznámení s návrhovým vzorem MVVM. Na základě těchto poznatků byl udělán prvotní návrh aplikace a byla vytvořena kostra projektu. V první fázi implementace byla definována sada příkazů obsluhujících připojení a komunikaci s kamerou SAFETIS. Na základě této sady byl implementován přenos obrazu v reálném čase a testována výkonnost. Postupně byly implementovány požadované obslužné nástroje. Těmito nástroji jsou zobrazení měřících funkcí, konfigurace měření, nahrávání dat a další.

V další fázi implementace byl do systému zařazen PT motor s řídicí jednotkou. Nejprve byla provedena analýza připojení a komunikace za pomocí Pelco-D protokolu, přičemž bylo v další fázi rozhodnuto o použití převodníku a připojení komunikace pomocí ethernetového připojení. V další fázi byly implementovány obslužné funkce pro manuální ovládání motoru, a poté konfigurace presetů a přepínání mezi nimi včetně implementace cyklovače.

Z pohledu záznamu dat byl implementován záznam radiometrického videa, a to jak automatický, tak manuální, periodický a taktéž manuální záznam radiometrických snímků. Přičemž obojí využívá pro ukládání interní implementovanou knihovnu ThermoLib. Dalším implementovaným záznamem je ukládání neradiometrického záznamu ve formátu avi. Tento záznam je momentálně v aplikaci zakomentován z důvodu testování a pro jeho implementaci bylo využito knihovny AForge.

V poslední části této práce jsou shrnuty výsledky testování poslední verze aplikace, jakož i navrhovaná možná vylepšení. Oproti zadání aplikace není implementováno zobrazení viditelného streamu z kamery, a to z důvodu změny HW platformy kamery, která nepodporuje tento záznam. Aplikace je však na tuto možnost připravena.

Uvedený systém a jeho výše uvedené funkce, zmíněné v zadání, a i další obsažené v implementované aplikaci tvoří v současné verzi ucelený systém, který je vhodný v aplikacích pro detekci rizika požáru. Díky možnosti využití digitálních výstupů je možné uplatnit ho v aplikacích automatického hašení a v momentální verzi je již testován v reálném provozu.

8 Literatura

[1] CommFront Communications. Pelco-d protocol tutorial [online], 2017. [cit. 2017-03-27]. Dostupné z: https://www.commfront.com/pages/pelco-d-protocol-tutorial.

[2] C# Corner. Wpf: An introduction [online], 2019. [cit. 2019-04-17]. Dostupné z: https://www.c-sharpcorner.com/UploadFile/nipuntomar/wpf-an-introduction-part-1//.

[3] DOTNETPORTAL.CZ. mvvm: model-view-viewmodel [online], 2019. [cit. 2019-04-17]. Dostupné z: https://www.dotnetportal.cz/clanek/4994/MVVM-Model-View-ViewModel.

[4] J. Fischer. Radiometrické a fotometrické veličiny. České vysoké učení technické - Videometrie a bezdotykové měření, 2015.

[5] Inc. FLIR Systems. Flir early fire detection [online], 2019. [cit. 2019-03-27]. Dostupné z: https://www.flir.eu/instruments/early-fire-detection/.

[6] GeeksforGeeks. Introduction to .net framework [online], 2019. [cit. 2019-04-17]. Dostupné z: https://www.geeksforgeeks.org/introduction-to-net-framework/.

[7] Claes Öhman. *Measurement in thermography.* Flir systems and the infrared training center, 2014. ISBN 978-91-637-6802-6.

[8] InfraTec infrared LLC. Fire prevention - fire-scan [online], 2019. [cit. 2019-03-27]. Dostupné z: https://www.infratec-infrared.com/thermography/observation-investigation/fire-scan/.

[9] K. Kadlec. Teoretické základy bezdotykového měření teploty (část 1) [online], 2014. [cit. 2017-03-13]. Dostupné z: http://automa.cz/Aton/FileRepository/pdf_articles/52890.pdf.

[10] K. Kadlec. Teoretické základy bezdotykového měření teploty (část 2) [online], 2014. [cit. 2017-03-13]. Dostupné z: http://automa.cz/Aton/FileRepository/pdf_articles/52626.pdf.

[11] Carosena Meola. *Infrared Thermography Recent Advances And Future Trends.* University of Naples Federico II, 2012.

[12] Microsoft. The model-view-viewmodel pattern [online], 2019. [cit. 2019-04-17]. Dostupné z: https://docs.microsoft.com/cs-cz/xamarin/xamarin-forms/enterprise-application-patterns/mvvm/.

[13] Microsoft. Overview of the .net framework [online], 2019. [cit. 2019-04-17]. Dostupné z: https://docs.microsoft.com/cs-cz/dotnet/framework/get-started/overview.

[14] Microsoft. Wpf overview [online], 2019. [cit. 2019-04-17]. Dostupné z: https://docs.microsoft.com/cs-cz/visualstudio/designers/introduction-to-wpf?view=vs-2019/.

[15] Jonathan M. Nichols Ronald G. Driggers, Melvin H. Friedman. *Introduction to Infrared and Electro-Optical Systems.* Boston: Artech House, 2012. ISBN 978-1-60807-101-2.

[16] J. Sova. Bezdotykové měření teplotních polí ii [online], 2014. [cit. 2017-03-13]. Dostupné z: https://www.aldebaran.cz/bulletin/2017_19_erm.php.

[17] VIDEOTEC S.P.A. Dtmrx2 telemetry data receiver [online], 2017. [cit. 2017-03-27]. Dostupné z: http://www.videotec.com/cat/en/products/pt-motors-and-telemetry/telemetry/dtmrx2/.

[18] VIDEOTEC S.P.A. Pth300 outdoor p&t motor [online], 2017. [cit. 2017-03-27]. Dostupné z: http://www.videotec.com/cat/en/products//pt-motors-and-telemetry/pt-motors/pth300.

[19] Papouch s.r.o. Tmu - usb teploměr [online], 2019. [cit. 2019-04-18]. Dostupné z: https://www.papouch.com/cz/shop/product/tmu-usb-teplomer/.

[20] Workswell s.r.o. Produkty [online], 2017. [cit. 2017-03-26]. Dostupné z: https://workswell.cz/produkty.

[21] Telerik. Getting started with xaml [online], 2019. [cit. 2019-04-17]. Dostupné z: https://docs.telerik.com/devtools/wpf/getting-started/common-faq.

[22] Vecow. Vecow ivh-9000 series [online], 2019. [cit. 2019-04-17]. Dostupné z: http://www.vecow.com/dispPageBox/vecow/VecowCT.aspx?ddsPageID=PRODUCTDTL_EN&dbid=4470318881/.

[23] WiX. Documentation for the wix toolset [online], 2019. [cit. 2019-04-17]. Dostupné z: https://wixtoolset.org/documentation/.

A Obsah disku

V tabulce 5 jsou zaznamenány názvy složek na přiloženém CD i s popisem jejich obsahu.

Jméno složky	**Popis obsahu**
Diplomova_prace	Diplomová práce ve formátu pdf.
Diplomova_prace_kód	Latexový zdrojový kód textu práce.
Zdrojový_kód	Zdrojový kód implementované aplikace.
Foto	Fotografie systému.

Tabulka 5: Obsah přiloženého disku CD

B Seznam použitých zkratek

V tabulce č. 6 jsou uvedeny použité zkratky v rámci této diplomové práce.

Zkratka	**Význam**
SW	Software
FW	Firmware
HW	Hardware
PC	Počítač
SMARTIS	Smart Thermal Imaging System
I/O	Input / Output
LWIR	Long Wavelength Infrared
SSD	Solid state disk
RAM	Random access memory
TCP/IP	Transmission Control Protocol/Internet Protocol
PT	Pan Tilt
PTZ	Pan Tilt Zoom
COM port	Communication port
CCTV	Closed circuit television
WPF	Windows Presentation Foundation
XAML	eXtensible Application Markup Language
XML	eXtensible Markup Language
MVVM	Model View ViewModel
CLR	Common Language Runtime
CTS	Common Type System
GUI	Grafické uživatelské rozhraní
MVC	Model View Controller
MVP	Model View Presenter
Wix	Windows Installer XML
ROI	Region of interest
SCU	SAFETIS Control Unit
GPS	Global Positioning System
FIFO	First in First Out
SDK	Software development kit

Tabulka 6: Seznam použitých zkratek

C Příkazy pro komunikaci s kamerou SAFETIS

Příkaz	Popis
GSS	Vyčtení sériového čísla kamery
GPN	Vyčtení sériového čísla jádra
GFV	Vyčtení verze firmwaru
GDT	Vyčtení aktuálního času
GSP	Vrátí rychlost (framerate) kamery
GRE	Vyčtení rozlišení kamery
HIS	Ověření zda je jedná o kameru SAFETIS, pokud ano vrací OK
RNR	Vyčtení aktuálního počtu měřících prvků
RVA	Vrací aktuální hodnoty měření
RNA	Vrací názvy měření
RRE	Vyčtení stavů digitálních výstupů kamery
RCO	Vyčtení nastavených barev měření
RDP	Vyčtení aktuálního limitního počtu pixelů
RDT	Vyčtení časového zpoždění měření
REV	Vrátí limitní hodnoty nastavené pro měření
RGM	Vrátí typ měřícího nástroje
RFP	Ověření splnění podmínky měření
RRP	Vyčtení aktuálního počtu pixelů nad limit měření
RDO	Vyčtení párování měření s digitálními výstupy
GDO	Ověření stavu digitálních výstupů
GSD	Vyčtení nastavení logiky digitálních výstupů
GCA	Vyčtení kalibrační matice kamery
GTT	Vrátí teplotu jádra
GTG	Vrátí teplotní rozsah kamery
GTE	Vrátí nastavenou emisivitu kamery
GTA	Vrátí nastavenou teplotu atmosféry kamery
GTR	Vrátí nastavenou odraženou teplotu kamery
CFS	Ověření zapnutí tzv. FireSafety modu

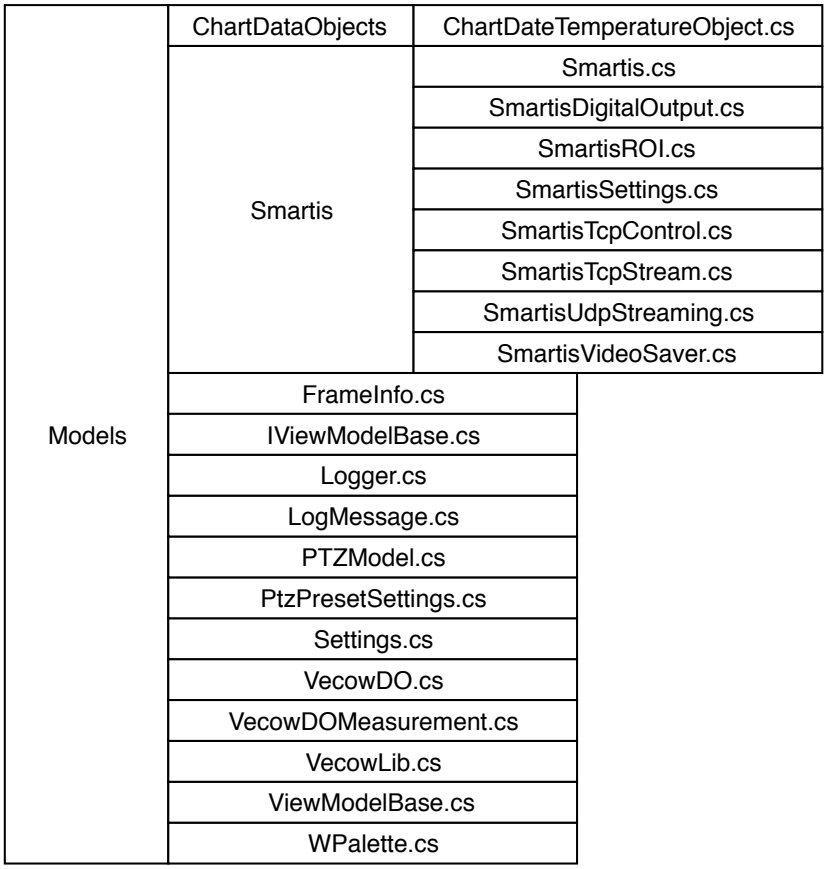
Tabulka 7: Příkazy pro vyčtení dat z termokamery SAFETIS

D Rozdělení tříd aplikace

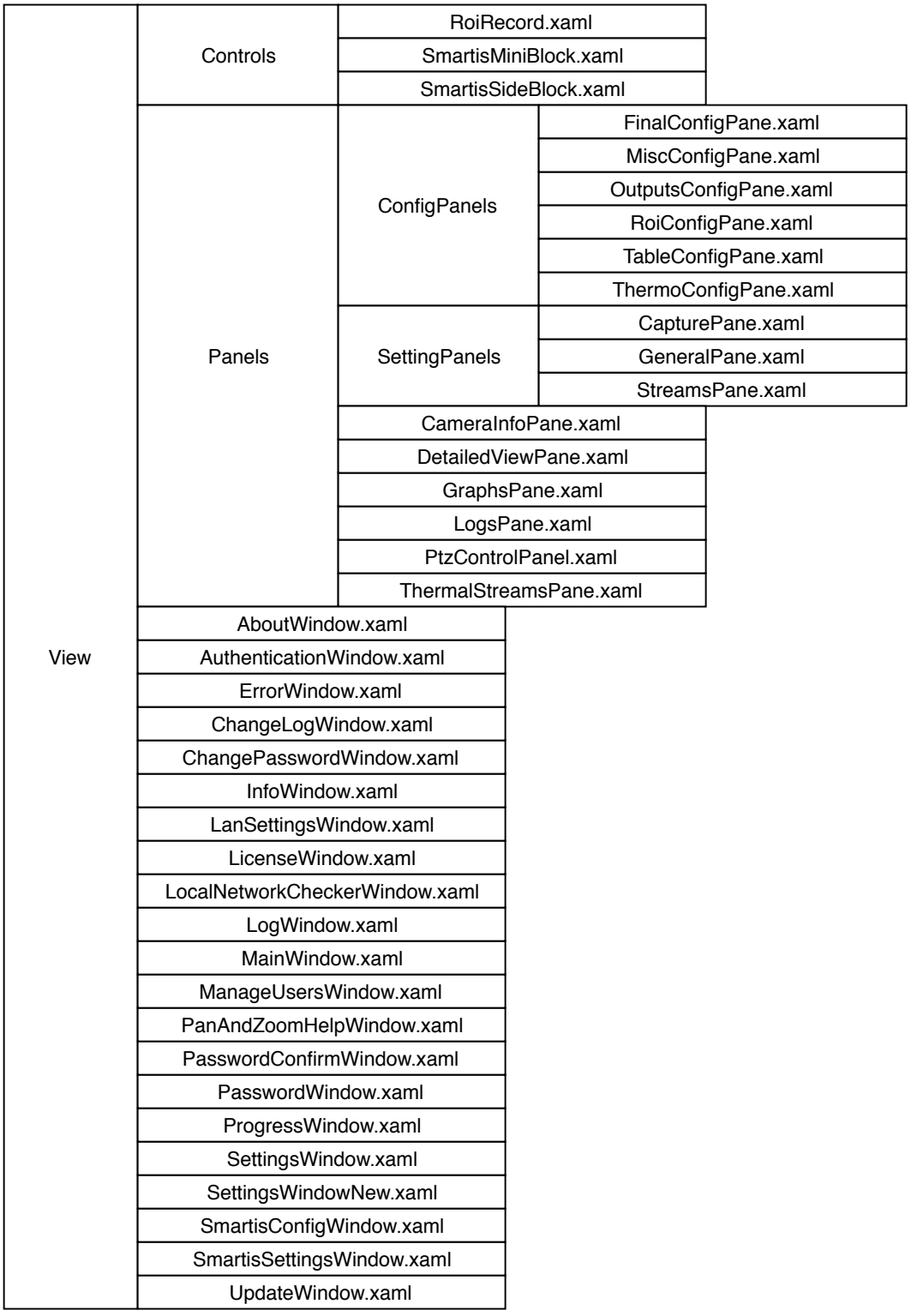
Rozložení aplikace do částí zvaných "Namespace"je uvedeno v kapitole 5.3 na obrázku č. 14. Součástí této přílohy jsou tabulky s obsahem tříd v jednotlivých částech, přičemž v tabulce uvádějící xaml třídy z namespace View je předpokladem existence tříd v code behind tedy stejnojmenných souborů s příponou xaml.cs.

Příkaz	Popis
PFM	Pozastavení měření
CFM	Spuštění měření
SFS	Nastavení tzv. FireSafety modu
SST	Spuštění sekvence pro konfiguraci kamery
FST	Ukončení sekvence pro konfiguraci kamery
SNB	Nastavení počtu měření
SGM	Nastavení typu měření
CRP	Ověření validity pixelů v měření
SCO	Nastavení barvy měření
SDP	Nastavení limitu pixelů
SDT	Nastavení času zpoždění měření
SEV	Nastavení limitní hodnoty měření
SNA	Nastavení jména měření
SDB	Nastavení výstupní logiky digitálních výstupů
SDA	Nastavení párování měření a digitálních výstupů
SRC	Zapnutí a vypnutí clusteringu pixelů
SNT	Nastavení IP adresy a masky podsítě
STD	Nastavení času a data kamery
STE	Nastavení emisivity kamery
STA	Nastavení teploty atmosféry kamery
STR	Nastavení odražené teploty kamery
STG	Nastavení teplotního rozsahu kamery
SCC	Nastavení doby provedení periodické FFC kalibrace

Tabulka 8: Příkazy pro nastavení dat termokamery SAFETIS



Obrázek 26: Obsah složky (Namespace) Models



Obrázek 27: Obsah složky (Namespace) View

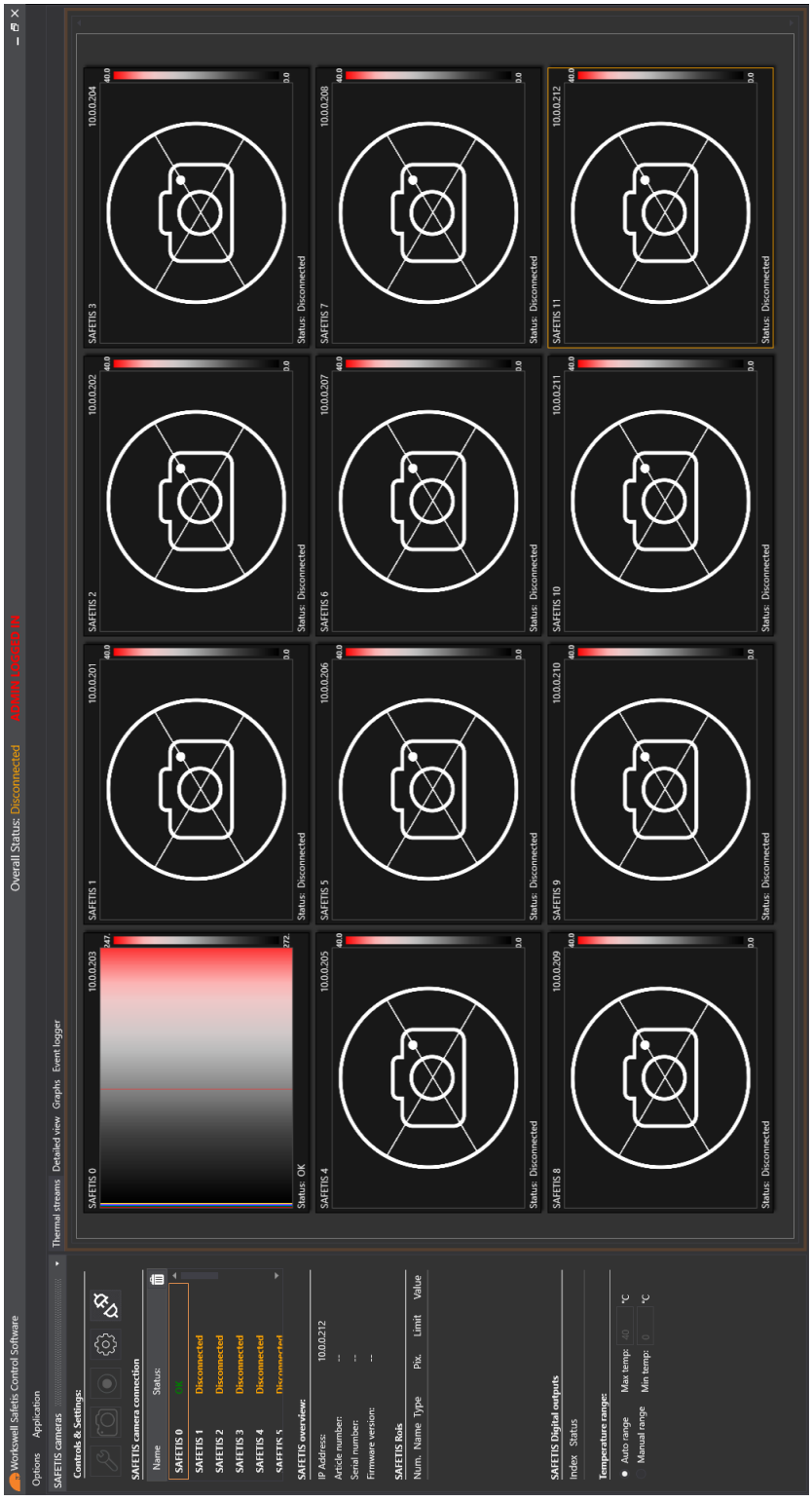
ViewModels	ApplicationViewModel.cs
	CaptureViewModel.cs
	DigitalOutputViewModel.cs
	DiskNetItemViewModel.cs
	DiskNetworkValidationViewModel.cs
	DriveItemViewModel.cs
	DriveManagementViewModel.cs
	GraphsPaneViewModel.cs
	LoginUsersViewModel.cs
	LogWindowViewModel.cs
	MainWindowViewModel.cs
	MaximumGraphViewModel.cs
	NetworkCheckerViewModel.cs
	PaletteViewModel.cs
	PTZPresetViewModel.cs
	PTZViewModel.cs
	SettingsPaneViewModel.cs
	SettingsViewModel.cs
	SmartisCenterViewModel.cs
	SmartisConfigPaneViewModel.cs
	SmartisConfigViewModel.cs
	SmartisRoiViewModel.cs
	SmartisViewModel.cs
	UpdateViewModel.cs
	UserViewModel.cs
	VecowLibViewModel.cs
	WarningViewModel.cs

Obrázek 28: Obsah složky (Namespace) ViewModels

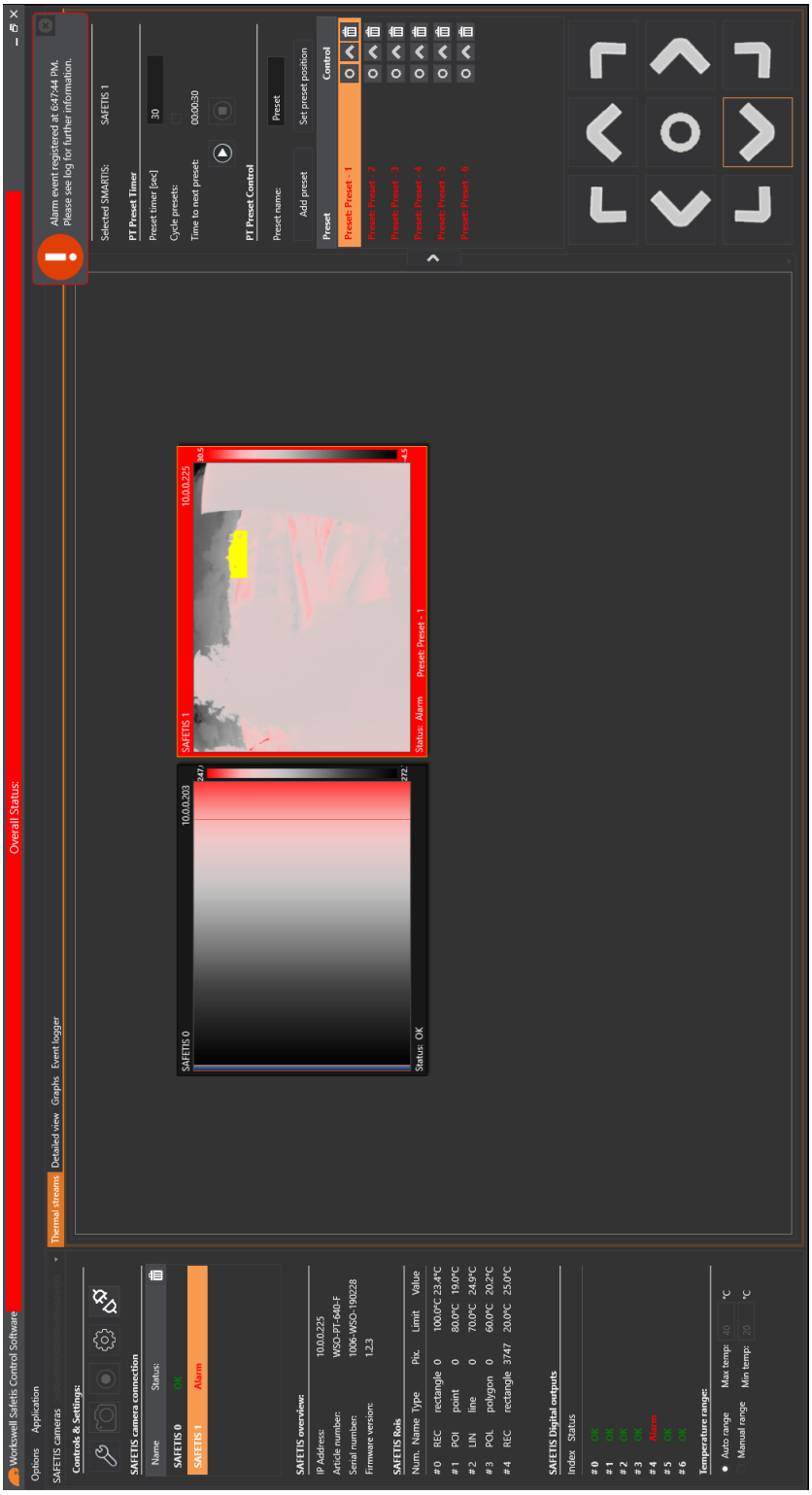
Utils	Converters	BitmapConversion.cs
		BooleanConverter.cs
		BooleanToBrushConverter.cs
		BooleanToSourceConverter.cs
		BooleanToStringConverter.cs
		BooleanToVisibilityConverter.cs
		BooleanToVisibilityHiddenConverter.cs
		BoolToOppositeBoolConverter.cs
		CelsiusToFahrenheit.cs
		DoubleConverter.cs
		EnumDescriptionConverter.cs
		RoundingConverter.cs
		ScaleToPercentConverter.cs
		TicksStringToTimeSpan.cs
	Translation	ITranslationProvider.cs
		LanguageCommands.cs
		LanguageChangedEventManager.cs
		ResxTranslationProvider.cs
		TranslateExtension.cs
		TranslationData.cs
		TranslationManager.cs
		TranslationViewModel.cs
		XmlTranslationProvider.cs
	Authenticator.cs	
	BugReport.cs	
	CaptureDirPeriods.cs	
	ConfigEnums.cs	
	FingerPrint.cs	
	InputBindingsManager.cs	
	NullCommand.cs	
	RelayCommand.cs	
	TempRangesEnum.cs	
	Utilities.cs	

Obrázek 29: Obsah složky (Namespace) Utils

E Grafické uživatelské rozhraní aplikace



Obrázek 30: Celé grafické uživatelské rozhraní aplikace



Obrázek 31: Celé grafické uživatelské rozhraní aplikace při alarmové situaci

F Snímky testovacího zapojení systému

Obrázek 32: Snímek testovacího zapojení systému

Obrázek 33: Snímek kamery SAFETIS v krytu na PT motory