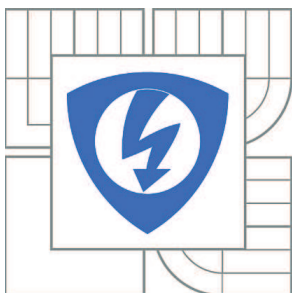


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

PROGRAM PRO INOVACI ŘÍDICÍHO SYSTÉMU SORT

PROGRAM FOR INOVATION OF THE SORT CONTROL SYSTEM

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

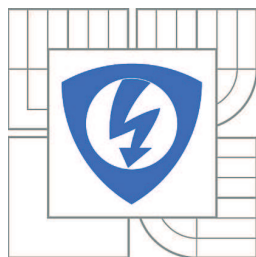
Bc. MARTIN ŠULC

VEDOUCÍ PRÁCE

SUPERVISOR

prof. Ing. FRANTIŠEK ZEZULKA, CSc.

BRNO 2014



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Diplomová práce

magisterský navazující studijní obor
Kybernetika, automatizace a měření

Student: Bc. Martin Šulc

ID: 125665

Ročník: 2

Akademický rok: 2013/2014

NÁZEV TÉMATU:

Program pro inovaci řídicího systému SORT

POKYNY PRO VYPRACOVÁNÍ:

Diplomová práce je součástí prací na vytvoření robotického solárního dalekohledu SORT.

Úkolem studenta je:

1. Implementace komunikačního protokolu pro komunikaci ovládacího počítače solárního optického robotického teleskopu (SORT) s celým systémem SORT.
2. Tvorba programu pro ovládání kamer a fukuserů a pro správu dat.
3. Modifikování programu v programovatelném automatu pro bezprostřední řízení dalekohledu a souvisejících podsystémů (napájení, pohonného systému kupole, osvětlení a dalších).
4. Testování funkce, vyhodnocení testů, tvorba dokumentace.

Vlastní nadřazené řízení SORTu, které bude implementováno na řídicím počítači není předmětem diplomové práce.

DOPORUČENÁ LITERATURA:

[1] BERRY, Richard a James BURNELL. The Handbook of Astronomical Image Processing. Willmann-Bell, 2005. ISBN 0943396824.

[3] IBM CORP. TCP/IP: Tutorial and Technical Overview [online]. 8. vyd. 2006. Dostupné z: <http://www.redbooks.ibm.com/redbooks/pdfs/gg243376.pdf>

Termín zadání: 10.2.2014

Termín odevzdání: 19.5.2014

Vedoucí práce: prof. Ing. František Zezulka, CSc.

Konzultanti diplomové práce:

doc. Ing. Václav Jirsík, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

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

ABSTRAKT

Analýza zadání pro rozšíření řídicího systému a návrhu ovládacího programu pro pozorovací kamery a s tím spojený teoretický rozbor nezbytný pro pochopení problematiky. Návrh komunikačního protokolu pro zpřístupnění všech funkcí systému přes TCP/IP komunikaci. Následná implementace protokolu do ovládacího programu a do řídicího programovatelného automatu.

Naprogramování aplikace s uživatelským rozhraním pro ovládání kamer a zpracování obrazových dat. Program také představuje server pro TCP/IP komunikaci s klienty přes navržený protokol. Navržená aplikace musí splňovat vysoké nároky na datové toky z kamer.

Rozšíření stávajícího systému pro řízení teleskopu, kopule a dalších systémů. Doplnění o měření analogových a řízení digitálních vstupů a výstupů. Dále pak řízení krokových motorů pro ovládání fokusérů. Implementovaný komunikační protokol se rozšíří o nové příkazy.

KLÍČOVÁ SLOVA

speciální kamera, synchronizace snímání, PLC, řízení krokového motoru, komunikační protokol, MFC Desktop aplikace, řízení astronomického systému, Diplomová práce, VUT

ABSTRACT

The analysis for the extension of the control system and proposal of controlling program for surveillance cameras and related theoretical analysis necessary for understanding problems. Proposal communication protocol to access all functions via TCP/IP communication. Subsequent protocol implementation into control program and a programmable controller.

Programming application with a user interface for camera control and basic image processing. The program also represents the server for TCP/IP communication with clients over the proposed protocol. The proposed application has to fulfill high pretensions of data streaming from cameras.

Extension of the existing control system of the telescope, dome and other systems. The addition of analog measurement and control of digital inputs and outputs. Furthermore, control of stepper motors to control focusers. The embedded communication protocol will be expanded of the new instruction.

KEY WORDS

special camera, synchronization scan, PLC, stepper motor control, communication protocol, MFC Desktop application, astronomic system control, Master's thesis, BUT

Bibliografická citace mé práce:

ŠULC, M. *Program pro inovaci řídicího systému SORT*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2014. 68 s. Vedoucí diplomové práce prof. Ing. František Zezulka, CSc..

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. Díl 4 Trestního zákoníku č. 40/2009 Sb.

Zde bych chtěl také poděkovat panu prof. Ing. Františku Zezulkovi, CSc. za odborné vedení, cenné rady a připomínky

Obsah

Seznam obrázků.....	7
Seznam tabulek.....	9
1 Úvod.....	10
1.1 Postup řešení.....	10
2 Návrh zadání a koncepce inovace.....	11
2.1 Stávající řešení.....	11
2.2 Návrh inovace.....	12
2.2.1 Návrh HW úprav.....	13
2.2.2 Změny programového vybavení.....	13
2.2.3 Požadavky na ovládání kamer.....	14
2.2.4 TPServer a TPView.....	14
3 Teoretický rozbor řešení.....	16
3.1 Použitý hardware.....	16
3.1.1 Kamery BM - 141Ge.....	16
3.1.2 Kamera SP-5000-GE.....	20
3.1.3 Harddisky VelociRaptor.....	21
3.1.4 PLC S7 - S300 a řízení krokových motorů.....	22
3.2 Komunikační protokoly.....	23
3.2.1 TCP/IP.....	24
3.2.2 ASCOL.....	25
3.3 FITS soubor.....	25
3.4 MFC Desktop aplikace.....	26
3.4.1 Windowsové zprávy.....	26
3.4.2 Kritická sekce.....	27
4 Návrh komunikačního protokolu.....	28
4.1 Koncepce.....	28
4.2 Přehled příkazů.....	30
5 Synchronizace kamer.....	33
6 Ovládací program cafe3.....	35
6.1 JAI SDK.....	35
6.2 Struktura programu.....	35
6.3 Rozbor jednotlivých funkčních částí.....	37
6.3.1 Vlákno pro obsluhu kamer.....	37
6.3.2 GUI.....	39
6.3.3 TCP/IP komunikace – server/klient.....	40
6.3.4 Stream dat z kamery.....	43
6.3.5 Zápis dat na disk.....	44
6.3.6 Zobrazení snímků.....	45
7 Program v PLC a vizualizace.....	48
7.1 Rozšíření kom. protokolu ASCOL.....	48

7.2	Ovládání krokových motorů fokusérů	50
7.3	Rozšíření programu v PLC o nové funkce.....	51
7.4	Úprava vizualizace	52
8	Dokumentování projektu a funkční zkoušky	53
8.1	Manuál a dokumentace	53
8.1.1	Menu	54
8.1.2	Ovládání kamer	56
8.1.3	LiveView.....	58
8.1.4	Nastavení FITS hlavičky.....	59
8.1.5	Ovládání fokusérů.....	61
8.2	Zkoušení a ladění na procesu	62
9	Závěr	64
	Citovaná literatura.....	65
	Seznam zkratk a symbolů	66
	Seznam příloh	67

SEZNAM OBRÁZKŮ

<i>Obrázek 1 - Schéma současného síťového zapojení hardwaru</i>	11
<i>Obrázek 2 – Koncepce systému</i>	12
<i>Obrázek 3 – 12-pinový multi-konektor kamery BM-141GE [6]</i>	16
<i>Obrázek 4 – Opticky izolovaný vstup[6]</i>	17
<i>Obrázek 5 – Opticky izolovaný výstup [6]</i>	17
<i>Obrázek 6 – GPIO pro kamery BM – 141GE [6]</i>	18
<i>Obrázek 7 – Vertikální binnig</i>	18
<i>Obrázek 8 – Částečné skenování</i>	19
<i>Obrázek 9 – Varianty vyčítání dat z kamer</i>	20
<i>Obrázek 10 – GPIO pro kameru SP-5000-GE</i>	21
<i>Obrázek 11 – Základní sestava PLC</i>	22
<i>Obrázek 12 – Zapojení karty 1STEP-5V pro řízení krokového motoru</i>	23
<i>Obrázek 13 – Komunikační struktura</i>	24
<i>Obrázek 14 – Zapouzdření dat pomocí TCP/IP</i>	25
<i>Obrázek 15 – Komunikace mezi HŘS a Castorem</i>	28
<i>Obrázek 16 – Synchronizace kamer (1. řídicí, 2. řízená kamera)</i>	33
<i>Obrázek 17 – Průběhy signály při synchronizaci kamer</i>	34
<i>Obrázek 18 – Vláknová struktura programu CaFe3</i>	36
<i>Obrázek 19 – Datové a komunikační kanály programu CaFe3</i>	36
<i>Obrázek 20 – Struktura vlákna pro obsluhu kamer</i>	38
<i>Obrázek 21 – Struktura hlavního GUI vlákna</i>	40
<i>Obrázek 22 – Struktura vlákna pro TCP/IP komunikaci</i>	42
<i>Obrázek 23 – Struktura vlákna pro TCP/IP komunikaci – periodická funkce</i>	43
<i>Obrázek 24 – Struktura vlákna pro vyčítání dat z kamer</i>	44
<i>Obrázek 25 – Vývojový diagram pro zobrazování snímků</i>	46
<i>Obrázek 26 – Funkce pro zpětné volání při vykreslování obrázku</i>	47
<i>Obrázek 27 – ASCOL - klasifikace příkazů</i>	48
<i>Obrázek 28 – Nastavení karty 1STEP-5V</i>	51
<i>Obrázek 29 – Vizualizace – ovládání napájení jednotlivých komponent</i>	52
<i>Obrázek 30 – Vizualizace – detekce přimrznutí kopule</i>	52
<i>Obrázek 31 – GUI – Ovládací program CaFe3</i>	53
<i>Obrázek 32 – CaFe3 – Nastavení formátu pixelu</i>	54
<i>Obrázek 33 – CaFe3 – Nastavení velikosti paketů</i>	54
<i>Obrázek 34 – CaFe3 – Nastavení připojení kamer</i>	55
<i>Obrázek 35 – CaFe3 – Nastavení TCP/IP komunikace</i>	55
<i>Obrázek 36 – CaFe3 – Nastavení cesty pro ukládání snímků</i>	55
<i>Obrázek 37 – CaFe3 – Náhled kamery</i>	56
<i>Obrázek 38 – CaFe3 – Nastavení kamery</i>	57
<i>Obrázek 39 – CaFe3 – LiveView</i>	58
<i>Obrázek 40 – CaFe3 – Ovládání LiveView</i>	59

<i>Obrázek 41 – CaFe3 – FITS hlavička</i>	<i>60</i>
<i>Obrázek 42 – CaFe3 – Ovládání fokusérů</i>	<i>61</i>
<i>Obrázek 43 – Snímací kamery</i>	<i>62</i>
<i>Obrázek 44 – Fokusér s krokovým motorem</i>	<i>63</i>

SEZNAM TABULEK

<i>Tabulka 1 – Výsledky testování rychlosti zápisu dat</i>	<i>22</i>
<i>Tabulka 2 – Seznam příkazů pro SOCOL – funkce CaFe.....</i>	<i>30</i>
<i>Tabulka 3 - Seznam příkazů pro PLC – ASCOL s dodatkem.....</i>	<i>31</i>
<i>Tabulka 4 – ISTEP-5V - výstupní byty</i>	<i>50</i>
<i>Tabulka 5 – ISTEP-5V – vstupní byty</i>	<i>50</i>

1 ÚVOD

Slunce představuje nejenom pro lidstvo, ale pro veškerý život na Zemi, nezbytnou součást existence. Už od nepaměti je k němu upínána lidská pozornost. Ať už v aztécké, řecké nebo římské mytologii je slunce zobrazováno jako nadpřirozeno v podobě různých božstev. V dnešní době se již člověk neobrací ke slunci jako k bohu, ale i přesto je jím fascinován. I když o něm máme spoustu informací, vzdálenost 150×10^6 km, teplota 5800 K, stáří 4,6 miliardy let a další, přesto je nutné stále ho pozorovat. Ať už z důvodu monitorování erupcí, které generují elektromagnetické pole s dosahem na povrch země, nebo pro další výzkum této nejbližší hvězdy.

S rozvojem nových technologií přichází i obrovský nárůst zpracovávaných dat. Zejména pak v astronomii, kde se mnohdy jedná o multispektrální obrazová data s vysokým rozlišením snímána vysokou rychlostí. Tento nárůst vyžaduje nové přístupy k navrhování a řešení systémů pracujících s těmito daty.

Další problematikou spojenou se složitými astronomickými systémy je dostupnost náhradních dílů. Zavedením průmyslových komponent do procesu řízení, je opodstatněnou cestou, kterou se v dnešní době vývoj ubírá. Průmyslová řešení představují robustní způsob řízení s vysokou životností a snadnou dostupností náhradních dílů po celém světě.

1.1 Postup řešení

V této kapitole jsem čerpal z literárního zdroje [4].

Úkolem diplomové práce navázání na semestrální projekt, jehož cílem bylo prostudovat systém pro ovládání a komunikaci se solárním dalekohledem. V současné době je již dalekohled řízen průmyslovým automatem, ale nové požadavky na systém vyžadují jak hardwarové, tak softwarové úpravy. Hlavním bodem práce je pak vytvoření nového ovládacího programu pro snímací kamery a krokové motory fokusérů. Neméně důležitý požadavek, je zpřístupnění všech funkcí systému přes TCP/IP komunikaci.

Jelikož se jedná o reálnou zakázku, bude vhodné rozdělit si práci na jednotlivé fáze a postupovat podle nich.

- Fáze 1 – Objasnění projektu / Požadavky zákazníka / Nabídka
- Fáze 2 – Návrh / Řešení
- Fáze 3 – Implementace
- Fáze 4 – Zákaznický test / Factory Acceptance Test (FAT)
- Fáze 5 – Uvedení do provozu
- Fáze 6 – Provozní asistence & Zaškolení
- Fáze 7 – Předání zákazníka do péče oddělení servisu a podpory

Jedná o komplexní zakázku, a tudíž se řešení projektu účastní více týmů. Manažeři, procesní inženýři a programátoři. Tato semestrální práce je psána a vytvářena z pohledu programátora.

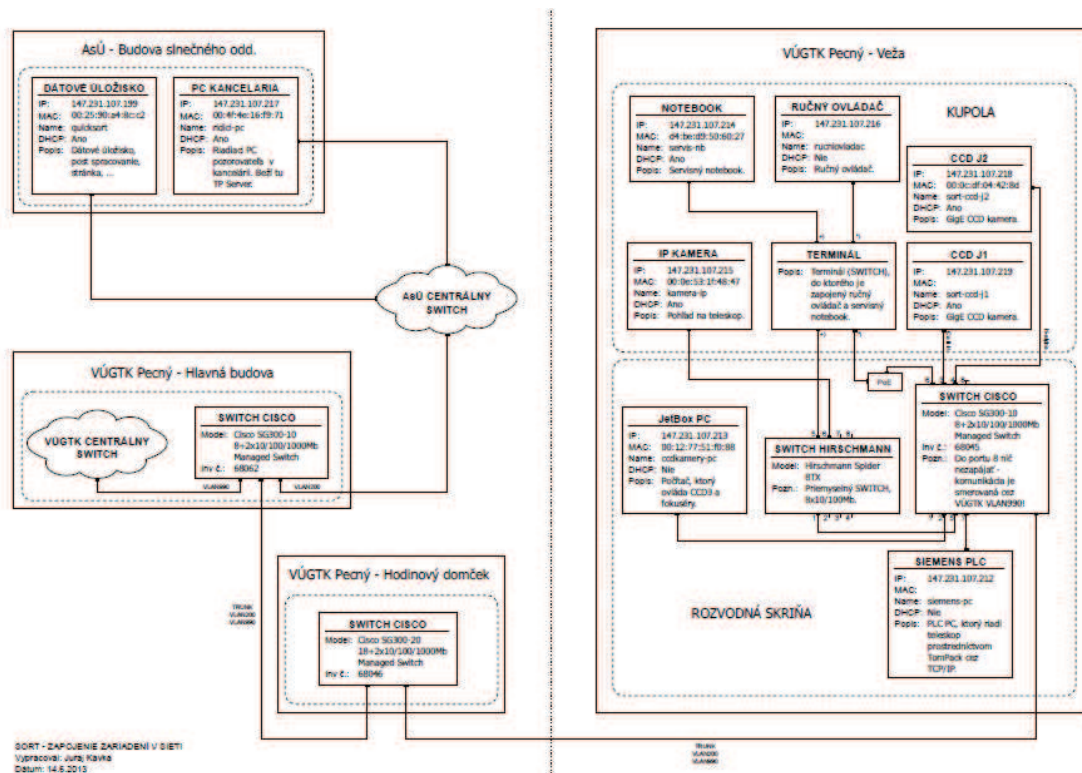
2 NÁVRH ZADÁNÍ A KONCEPCE INOVACE

2.1 Stávající řešení

Jak již bylo zmíněno výše, část práce spočívá v modifikaci stávajícího systému řízení. V této podkapitole probereme současný stav zařízení a jeho funkce.

Řízenými objekty, pro nás relevantními, jsou tři CCD kamery a dva fokuséry. Dvě stávající kamery SBIG ST402 pro detailní pozorování slunce budou nahrazeny novým vysokorychlostním typem BM – 141GE. Místo kamery SBIG ST3200 pro celodiskové snímání bude použita JAI SP – 500M – GE. V budoucnu se očekává i nasazení čtvrté celooblohové kamery.

Za řídicí systém je zde považován jeden stolní počítač založený na windowsové platformě (dále Castor) a jeden programovatelný automat (dále PLC) firmy Siemens. V nové koncepci je zahrnut nový počítač s operačním systémem Linux (dále Pollux), na kterém poběží hlavní řídicí skript. Tento skript bude vyhodnocovat příchozí obrazová data a na jejich základě vysílat příkazy do Castoru. Počítač Castor představuje hlavní komunikační a ovládací jednotku v systému. Běží na něm TomPack, software určený pro vizualizaci a spojení s PLC. Dále na něm bude umístěn program pro navázání komunikace s kamerami a vytvářející uživatelské rozhraní pro ovládání kamer, který bude nazván CaFe3. PLC je určeno pro robustní ovládání kopule, dalekohledu, fokusérů a ostatních členů

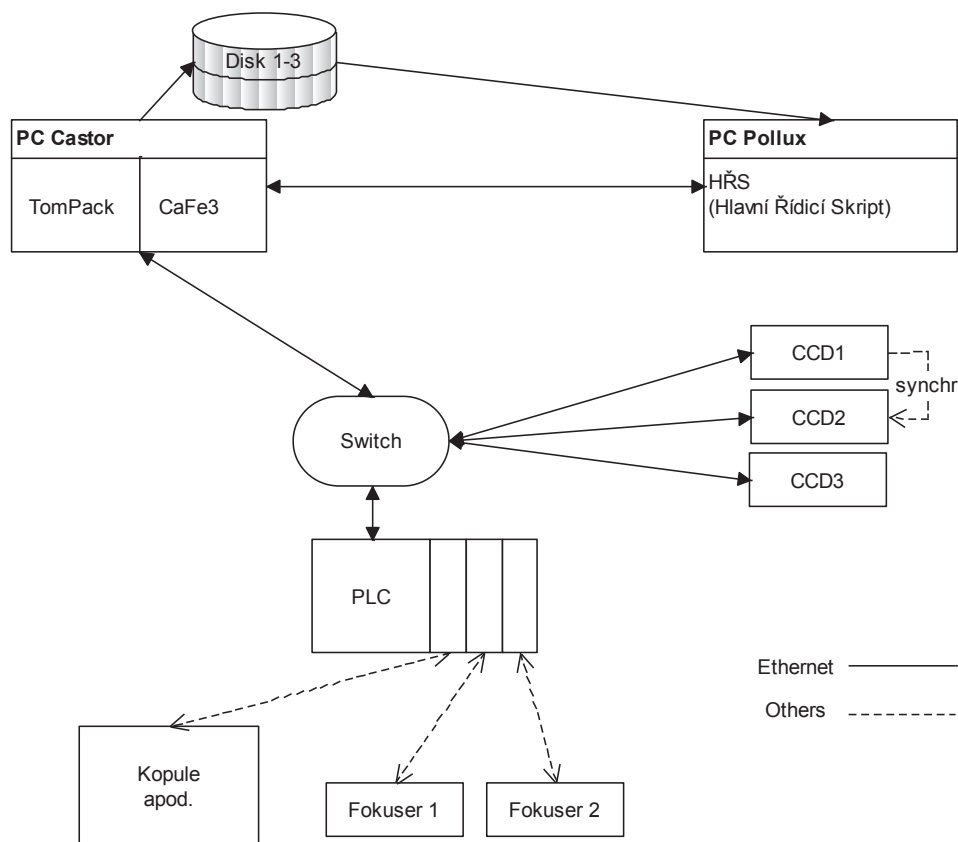


Obrázek 1 - Schéma súčasného síťového zapojení hardwaru

2.2 Návrh inovace

Solární optický robotický teleskop je plně automatizovaný systém určený pro pozorování slunce třemi kamerami. Dvě kamery typu JAI BM – 141Ge s ethernetovým rozhraním rozšířené o filtry H – alfa a Ca II K pro úzkopásmové detailové pozorování. Kamera JAI SP – 5000M – Ge také s ethernetovým rozhraním bude snímat celý disk slunce v bílém světle.

SORT (mimo snímací kamery) je ovládán pomocí PLC Siemens S7-300. Vizualizační program spolu s ovládacím a konfiguračním programem pro kamery a fokuséry běží na PC Castor ve sluneční budově. K tomuto počítači bude přidán druhý počítač Pollux s linuxovou platformou, na které poběží hlavní řídicí skript, který bude mít zpřístupněny všechny funkce SORTU.



Obrázek 2 – Koncepte systému

Systém je rozdělen na čtyři komunikační subsystémy založené na Ethernetu (protokol TCP/IP). Přičemž tři mají společnou část kanálu. Jedná se o komunikaci mezi TomPackem a programovatelným automatem (dále PLC), mezi programem CaFe3 a třemi kamery s CCD čipy, mezi CaFe3 a PLC, a nakonec mezi hlavním řídicím skriptem (dále HŘS) a TomPackem, Cafe3 a harddisky. Jednotkou řídicí tok dat v těchto kanálech je počítač Castor (Windowsová platforma).

Cafe3 ↔ CCD kamery - Výrobce dodává k zařízením soubor knihoven a zdrojových souborů, pomocí kterých se naváže komunikace. Kanál pro parametrizaci a čtení obrazových dat z kamer.

CaFe3 ↔ PLC – Kanál pro ovládání fokusérů. Komunikace pomocí protokolu ASCOL. CaFe3 v roli klienta průběžně vyčítá data z PLC a také zasílá požadavky klienta spojené s fokuséry (pohyb, zastavení, inicializace).

TomPack ↔ PLC – Přenos dat z automatu je zajištěn protokolem firmy Siemens S7/TCP. Komunikaci směrem k vizualizaci zajišťuje protokol TPLINK (ProjectSoft). Kompletní ovládání kopule, dalekohledu a zařízení s tím spojených.

TomPack, CaFe3, HDD ↔ PC Pollux (linuxová platforma s hlavním řídicím skriptem) – Kanál pro zpřístupnění funkcí a dat hlavnímu řídicímu skriptu. Mimo ovládání funkcí a informací o nich se budou zasílat všechny histogramy a 1 z n snímků (n bude záviset na nastavení a na vytíženosti sítě). Komunikace bude probíhat pomocí TCP/IP.

Z důvodu požadavku na vysoký datový tok (cca 500 Mb/s pro detailové kamery (2 x 10 fps, kde velikost snímku je 3 MB), cca 400 Mb/s pro celodiskovou (5 fps, kde velikost snímku je 10 MB) a zbylých 100 Mb/s pro ostatní provoz na síti) se bude muset jednat o 1Gb síť. Tedy síť propojenou ethernetovými kabely typu Cat5e. Switch bude muset mít na každém portu full-duplex 2Gb/s (1 + 1 Gb/s).

Vzhledem k vysokým rychlostním požadavkům na zápis dat (cca 30 + 30 + 50 MB/s) budou pro ukládání použity tři harddisky. Měřením zapisovací rychlosti byly získány uspokojivé výsledky (79, 72, 72 MB/S pro zápis na všechny disky najednou – rychlost pravděpodobně omezena rychlostí řadiče).

2.2.1 Návrh HW úprav

- Všechny tři kamery (2 x detail, 1x celodisk) budou připojeny a ovládány přes Ethernet oproti rozhraní USB a systému RTS
- Fokuséry budou připojené přes I/O karty PLC, namísto USB a systém RTS
- Vzhledem k předchozím změnám nebude potřeba PC JetBox se systémem RTS
- Synchronizace kamer přes GPIO jednotlivých kamer.*

* Synchronizace zajištěna propojením binárních vstupů a výstupů jednotlivých kamer a snímání bude spouštěno triggrem jedné kamery. Případné odpojení této řídicí kamery bude softwarově ošetřeno, tak aby bylo možné snímat pouze jednou kamerou.

2.2.2 Změny programového vybavení

- Vytvoření nového ovládacího programu CaFe3
- Rozšíření softwaru v PLC o ovládání fokusérů a o nové funkce
- Rozšíření komunikačního protokolu ASCOL
- Zpřístupnění všech funkcí (TomPack, Cafe) přes TCP/IP

2.2.3 Požadavky na ovládání kamer

Pro vytvoření plně robotického systému je nutné propojit komunikačním kanálem dva hlavní počítače Castor a Pollux. Hlavní řídicí skript běžící na Polluxu bude vyžadovat přístup k funkcím TPServeru zajišťujícím globální ovládání dalekohledu. A dále bude nutné zpřístupnit obrazová data přijímaná programem CaFe.

- nezávislé ovládání jednotlivých kamer
- nastavení cest k adresáři pro uložení snímků
- uložení snímků ve formátu fits s hlavičkou, uložení histogramu, nastavení cesty uložení zvlášť pro fits soubor a zvlášť pro histogram (.bmp)
- na vyžádání zobrazit aktuální ukládaný snímek
- spuštění a zastavení snímkování, snímkování rychlostí 10 fps z jedné kamery pro detailové kamery a 5 fps pro celodiskovou kameru
- pixelový formát: 12 bit – unpack (2 byty na pixel)
- expoziční doba v plném rozsahu kamer: 64 μ s – 2 s
- nastavení intervalu mezi snímky v maximálním možném rozsahu
- nastavení počtu sekvencí a počtu snímků v sekvenci, oboje na hodnotu 1 - ∞
- nastavení optical black transfer módu pro referenční účely, partial scan (2/3 scan, 1/2 scan, 1/4 scan, 1/8 scan)
- ošetření vertikálního binningu
- umožnění přenosu dat ve dvou módech
- při spuštění snímkování zobrazit zatížení sítě v Mb/s a fps
- synchronní snímkování obou kamer
- možnost nastavení packet size a jumbo frame
- možnost flat field korekce (pro celodiskovou kameru)
- live view
- zobrazení real-time histogramu
- Dále: kontinuální snímání obrazu s možností nastavit expoziční dobu, jas a kontrast a gain v plném rozsahu. Překlopení snímku ve vertikálním i horizontálním směru, rotace snímku v rozmezí 0° - 90° - 180° - 270°. Možnost zoomu. Funkce zapnout a vypnout dvojitý kříž a možnost nastavit hodnotu šířky kříže v pixelech, kdy při nastavené hodnotě 1 pixel bude průsečík o rozměrech 1 pixel x 1 pixel na středu. Barva kříže modrá nebo červená. Dále se bude v dolním rohu zobrazovat pozice kurzoru

2.2.4 TPServer a TPView

TPServer a TPView spadá to SCADA systému TomPack vyvinutým společností ProjectSoft. Níže je uveden seznam požadavků, které jsou na tento systém kladeny. S podmínkou že všechny funkce budou zpřístupněny pro hlavní řídicí skript, běžící na počítači Pollux.

- možnost čtení změny zapnutí dalekohledu, inicializaci dalekohledu, parkování dalekohledu, stop dalekohledu, vypnutí dalekohledu.
- možnost čtení hodnot meteo dat a indikace „bezvětrí“, „sucho – déšť“
- možnost čtení, změny a najetí na mechanické souřadnice, přepnutí na souřadnice hodinové osy ve stupních – ST, hodinách – HOD
- možnost čtení času UTC, LAST
- možnost čtení souřadnic Slunce, výšky Slunce + indikace nevhodné výšky a umožnění příkazu „NAJEĎ“
- možnost čtení hvězdných souřadnic, epochy, změna hvězdných souřadnic, změna přepnutí východ, západ, výška cíle, možnost najetí na zadané souřadnice, možnost track
- možnost čtení a změny uživatelských rychlostí, vypnutí a zapnutí
- možnost čtení a změny uživatelských korekcí, ovládání nulování
- možnost ručního ovládání, v budoucnu se počítá s ručním ovládáním přes joystick, možnost přepínat mezi rychlostmi T1, T2, T3 a umožnit ovládání guide režimu
- možnost ovládat clonu zrcadla
- možnost ovládat otevírání a zavírání kopule
- možnost ovládat napájení filtrů, kamer, světla v kopuli, linuxového JetBox PC, topení, chlazení, toto se musí přepracovat tak, aby každý z uvedených členů měl vlastní napájení
- vytvoření centrálního STOP, kdy při jeho aktivaci dojde k zaparkování dalekohledu, uzavření clony zrcadla, vypnutí chlazení, uzavření kopule apod., tedy dalekohled se dostane do stavu jako po ukončení pozorování
- upravit zhasínání světla v kopuli – možnost zhasnout světlo ze sluneční budovy při ručně rozsvíceném světle v kopuli – zde je nutný souhlas bezpečnostního technika, kvůli odpovědnosti

3 TEORETICKÝ ROZBOR ŘEŠENÍ

V teoretickém rozboru se budeme zabírat softwarovými i hardwarovými komponentami použitými v novém konceptu řídicího systému pro robotický dalekohled. Jedná se především o vysokorychlostní kamery BM – 141Ge, fokuséry a komunikační protokoly TCP/IP a ASCOL.

3.1 Použitý hardware

3.1.1 Kamery BM - 141Ge

V této kapitole jsem čerpal z literárního zdroje [6].

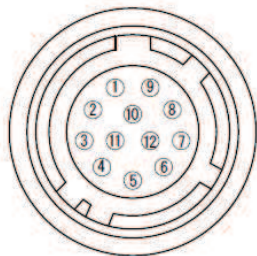
Jedním ze základních požadavků návrhu systému byl vysokorychlostní přenos a zpracování dat. Tomuto kritériu byl přizpůsoben výběr měřicích kamer. Kamery BM-141GE jsou vybaveny rozhraním GigE Vision, které používá Gigabitový Ethernet. Toto rozhraní bylo vyvinuto skupinou AIA (Automated Imaging Association).

Čip těchto kamer má rozměry 1434x1052, přičemž efektivní využití má pouze 1392x1040 pixelů. CCD technologie zajišťuje čipu velmi vysokou citlivost na světlo, malou míru zkreslení obrazové informace a vysokou rozlišovací schopnost. Nevýhodou je vyšší cena této technologie, která ještě značně převyšuje CMOS technologii.

3.1.1.1 GPIO – vstupní a výstupní porty

Všechny vstupní a výstupní signály procházejí přes GPIO (univerzální vstupy a výstupy) modul. Součástí tohoto modulu je i Look-Up Table (LUT – křížový přepínač), dva 19-bitové pulzní generátory a jeden 12-bitový čítač. Pomocí LUT přepínače jsme schopni propojit dané vstupy na výstup.

Type: HR10A-10R-12PB
(Hirose) male.
(Seen from the rear of
camera)



Pin no.	Signal	Remarks
1	GND	
2	+12 V DC input	
3	Opt IN 2 (-) / GND (*1)	GPIO IN / OUT
4	Opt IN 2 (+)/Iris Video out (*1)	
5	Opt IN 1 (-)	
6	Opt IN 1 (+)	
7	Opt Out 1 (-)	
8	Opt Out 1 (+)	
9	Opt Out 2 (-)	
10	Opt Out 2 (+)	
11	+ 12 V DC input	
12	GND	

Fig. 3. 12-pin connector.

*1: Iris Video output function can be set by the internal DIP switch (SW601).

Obrázek 3 – 12-pinový multi-konektor kamery BM-141GE [6]

Přes piny 1 a 2 je kamera napájena 12V, toto napětí je vyvedeno i na piny 11 a 12, které využijeme pro napájení optických vstupů a výstupů (viz *Obrázek 4* a *Obrázek 5*).

Opticky izolované vstupy a výstupy

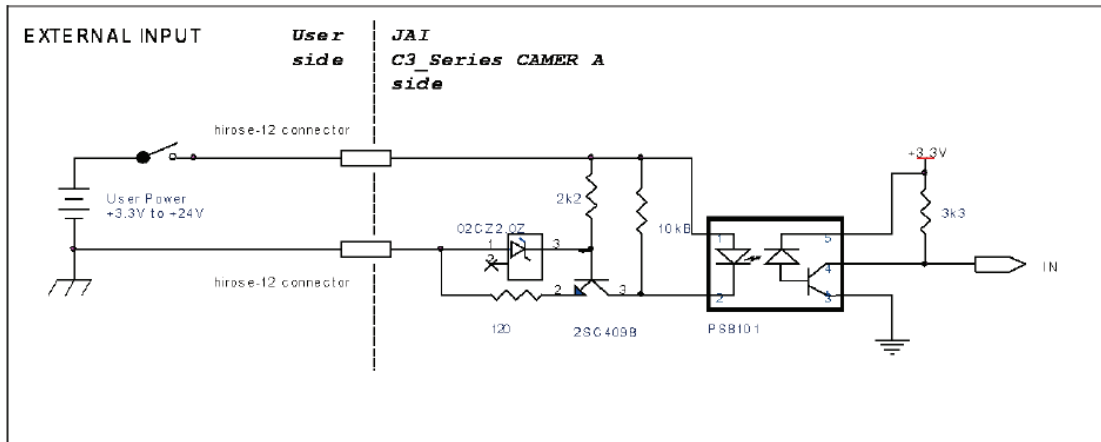


Fig.10. External Input Circuit, OPT IN 1 and 2

Obrázek 4 – Opticky izolovaný vstup[6]

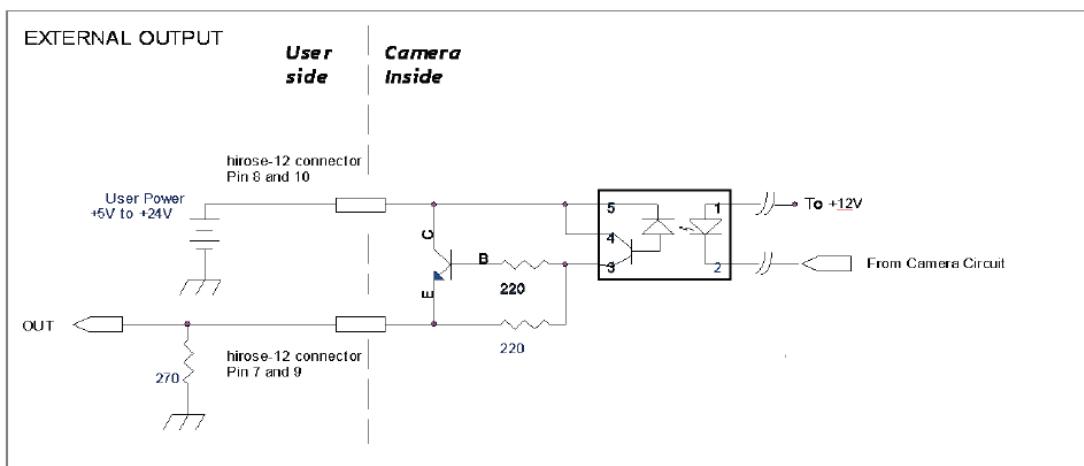
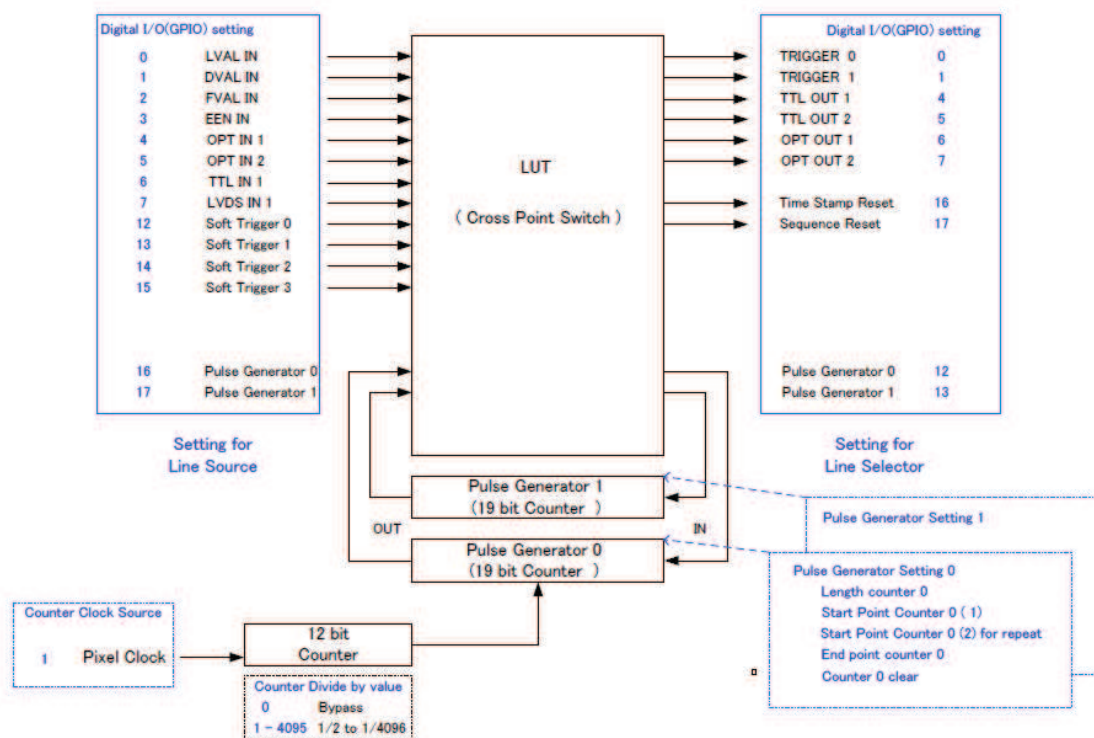


Fig.11. External Output Circuit, OPT OUT 1 and 2

Obrázek 5 – Opticky izolovaný výstup [6]

Na *Obrázek 6* je zobrazeno schéma vstupně – výstupních obvodů kamer BM – 141Ge. Jeho součástí je LUT, což je přepínač, jehož nastavení (vazba mezi vstupy a výstupy) je dáno nastavením vnitřního registru. Signály LVAL_IN, DVAL_IN, FVAL_IN a EEN_IN jsou generovány vnitřními časovacími obvody kamery. Trigger 0 je určen pro spouštění expozice a Trigger 1 pro zpoždění vyčítání dat. Vstupy a výstupy OPT (optical), TTL a LVDS jsou spojeny s fyzickým rozhraním.

Dále je součástí obvodu 12-bitový čítač sloužící pro dělení hodinových pulsů. Při nastavení čítače na nulu dojde k jeho vynechání. Pulzní generátory, jejichž součástí je také čítač, lze nastavit do dvou módů. Ve spouštěcím módu je pulz vygenerován pomocí vstupního signálu a to buď jeho náběžnou nebo sestupnou hranou nebo úrovní signálu. V periodickém režimu jsou kontinuálně generovány pulzy, jejichž šířka, počátek a konec lze nastavit. Frekvence je dána hlavními hodinami a 12-bitovým čítačem.



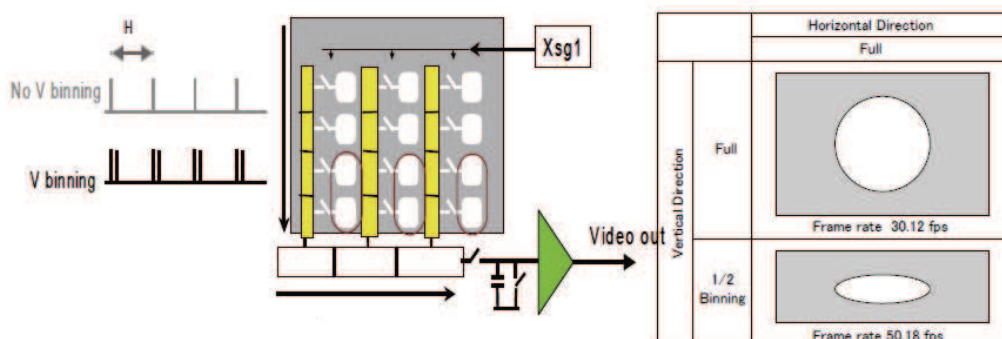
Obrázek 6 – GPIO pro kamery BM – 141GE [6]

3.1.1.2 Vertikální binning a částečné skenování

Dle zadání je nutné do ovládacího programu zakomponovat funkce vertikálního binningu a částečného skenování. V této části si přiblížíme, co tyto názvy znamenají.

Vertikální binning

V překladu lze binningem nazvat sdružování obrazových bodů. Prakticky se jedná vyčítání hodnot více pixelů z CCD jako jeden. Tato operace přináší vyšší citlivost (hodnota pixelů se sčítá) a větší rychlost vyčítání (sníží se množství dat). Nevýhodou je menší rozlišení. Použitá kamera umožňuje sjednocovat pixely pouze ve vertikálním směru.

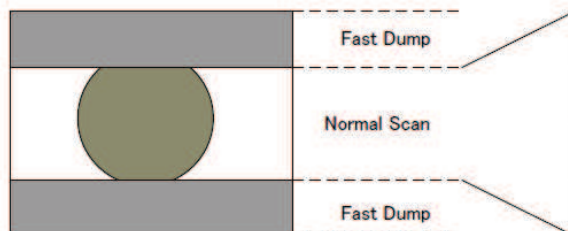


Obrázek 7 – Vertikální binning

Na Obrázek 7 je znázorněn princip vertikálního sdružování pixelů a je zde také uvedeno zvýšení rychlosti snímků za sekundu z 30,12 fps na 50,18 fps při 1/2 slučování.

Částečné skenování

V tomto režimu je využívána pouze středová část obrazu, čímž se dosáhne většího počtu snímků za sekundu. Je to užitečné v případě, že nám stačí informace pouze ze středové části snímku. Kamera BM – 141GE umožňuje pouze snižovat výšku obrazu a ne šířku a má k dispozici čtyři typy částečného skenování (2/3, 1/2, 1/4, 1/8). Mimo to kamery umožňují variabilní nastavení počátku a konce skenování.

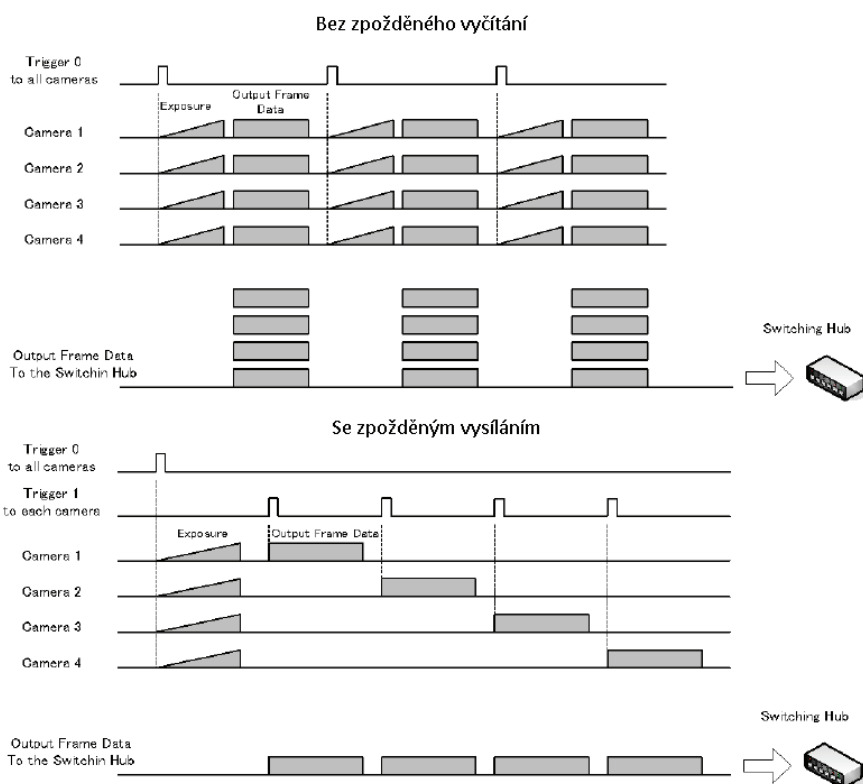


Mode	Start line	End line	Frame Rate
2/3	173	866	41.05 fps
1/2	261	780	50.06 fps
1/4	391	650	74.57 fps
1/8	455	584	98.73 fps

Obrázek 8 – Částečné skenování

3.1.1.3 Vyčítání dat

Data z kamer je možné vyčítat dvěma způsoby a oba je potřeba zpřístupnit z ovládacího programu. Buď mohou kamery posílat data na výstup neřízeně v čase, kdy skončí expozice, nebo lze zaslání dat zpozdít, tak aby jak je znázorněno na *Obrázek 9*. Nevýhodou prvního způsobu je, že v případě velkého množství dat, může dojít k naplnění zásobníku v síťovém prepínači, a tím ke ztrátě dat. Pro správný chod je tedy nutné za běhu kontrolovat stav tohoto zásobníku.



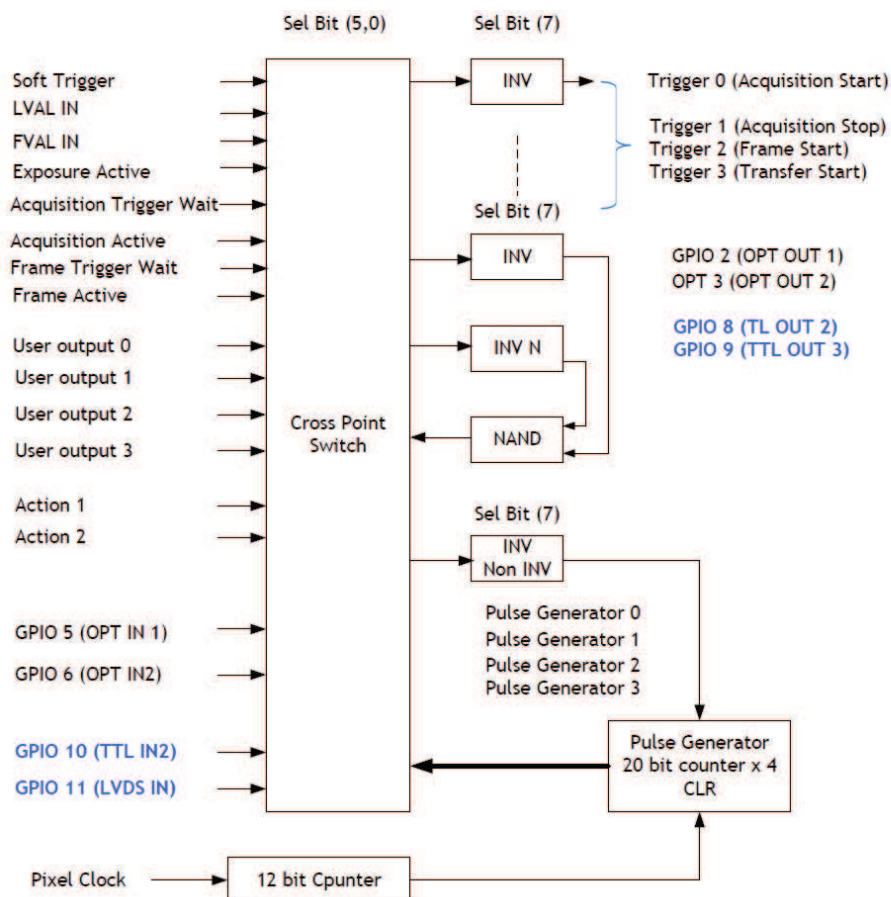
Obrázek 9 – Varianty vyčítání dat z kamer

3.1.2 Kamera SP-5000-GE

Jedná se o kameru od stejného výrobce (JAI) jako dvě kamery předcházející (BM-141GE). V sérii SP-5000 je to nejnovější typ s rozhraním GigE Vision 2.0. Nevýhodou toho, že se jedná o nový typ, je, že manuál byl zveřejněn relativně pozdě a zatím se jedná pouze o předběžnou verzi. V porovnání s kamerami BM-141GE se naštěstí moc neliší, co se týče do softwarového vybavení a rozhraní. Z hlediska hardwaru však umožňuje více funkcí.

Prvním značným rozdílem je použitý čip. Kamera SP-5000-GE je vybavena čipem o velikosti 2560x2048 s technologií CMOS. Tato technologie má sice horší vlastnosti než CCD, ale při dané aplikaci je dostačující. Při využití obou ethernetových portů a při výstupním formátu dat 8-bit na pixel, kamera umožňuje při plném rozlišení až 44 snímků za sekundu. Tato vlastnost je pro naše potřeby značně předimenzovaná. Kamera SP-5000-GE v roli celodiskové kamery bude snímat s maximální rychlostí 5 snímků za vteřinu.

Dále kamera umožňuje mimo vertikální binning i binning horizontální a rozsah expoziční od 10 μ s do 8s.



Obrázek 10 – GPIO pro kameru SP-5000-GE

Na Obrázek 10 je znázorněno všeobecné rozhraní pro vstupy, výstupy a pro kontrolu spouštěcích vstupů, výstupů a pulzních generátorů. Pomocí tohoto rozhraní je možné ovládat externí zdroj světla, vytvořit zpožďovací funkci pro vstupní spínací signál, nebo přesně řídit dobu expozice pomocí PWC spouště.

3.1.3 Harddisky VelociRaptor

Datové uložení určené pro obrazová data z detailových kamer bylo z důvodu nárůstu objemu dat upgradováno. Harddisk pro celodiskovou kameru nevyžaduje takovou rychlost zápisu. Původní dva pomalejší disky byly nahrazeny rychlejšími typy WD VelociRaptor DH – 1TB. Výrobce u těchto disků uvádí rychlost otáček 10 000 za minutu, rozhraní SATA 6Gb/s, 64 MB vyrovnávací paměti a kapacitu 1 TB.

Rychlost zápisu byla otestována programem, který generoval 18 000 souborů po 3 MB (3 MB = jeden snímek, tedy 10 minut snímkování pro 10snímků/s pro 3 kamery).

Výsledky testů:

disky d: e: nove WD VelociRaptor – pro detailové kamery

disk g: starší disk WD - pro celodiskovou kameru

Tabulka 1 – Výsledky testování rychlosti zápisu dat

Samostatně každý disk:			test spuštěn na disku d: e:			test spuštěn na všech discích:		
disk	čas	rychlost	disk	čas	rychlost	disk	čas	rychlost
d:	334s	162 MB/s	d:	364s	148 MB/s	d:	685s	79 MB/s
e:	342s	158 MB/s	e:	377s	143 MB/s	e:	748s	72 MB/s
g:	685s	79 MB/s				g:	752s	71.8 MB/s

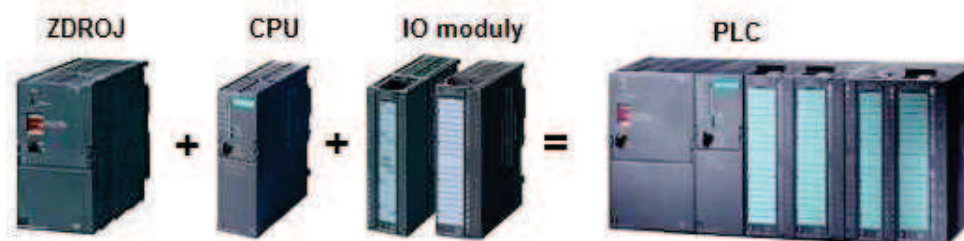
Při uvažování plné velikosti snímku z použité kamery BM – 141GE 1392x1040 pixelů a 16-bitů na pixel, dojdeme podle vzorce (1) k datovému toku 29 MB/s. Při výpočtu zanedbáváme velikost hlavičky FITS souboru. Z toho vyplývá, že při zápisu dat ze všech tří kamer budou disky vytíženy ani ne na 50%.

$$DataRate = (1392 \times 1040 \times 16) \times 10 = 232 \frac{Mbit}{s} = 29 \frac{MB}{s} \quad (1)$$

3.1.4 PLC S7 - S300 a řízení krokových motorů

V této kapitole jsem čerpal z literárních zdrojů [8] a [9].

Programovatelný automat zahrnuje tři základní druhy modulů. Vstupní moduly, na které jsou napojeny měřené procesní veličiny. Výstupní moduly, které provádějí akční zásah do procesu. A procesní jednotka CPU, která provádí naprogramované logické, matematické a časové funkce mezi vstupními a výstupními signály. Součástí zařízení často bývá i napájecí zdroj 24V/DC.



Obrázek 11 – Základní sestava PLC

Průmyslový řídicí systém SIMATIC S7-300 je nejprodávanejším řídicím systémem z široké nabídky firmy Siemens AG. Je určen pro realizaci rozmanitých automatizačních úloh středního rozsahu. Poskytuje univerzální automatizační platformu pro systémová řešení s hlavním důrazem na výrobní technologii. [8]

3.1.4.1 Krokový motor

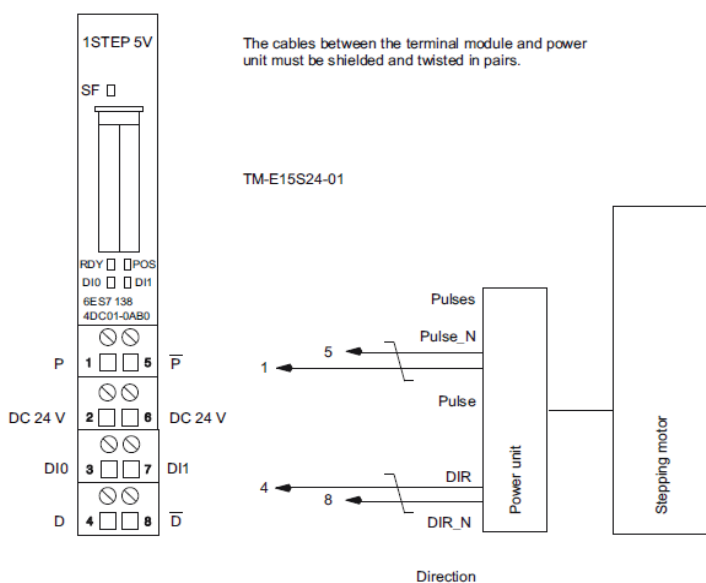
Krokový motor je elektrický stejnosměrný bezkartáčový motor, který rozděluje rotační pohyb na jednotlivé stejně velké kroky. Motor umožňuje řízení pohybu bez zpětné vazby, tak že jsme schopni nastavit jeho polohu, na libovolnou pozici v rozmezí výše

zmíněných kroků. Řízení probíhá přivedením sekvence obdélníkových pulzů na vstup motoru, což má za následek inkrementaci polohy hřídele.

Základní konstrukce krokového motoru spočívá v rovnoměrném rozmístění několika elektromagnetů okolo rotorové části v podobě ozubeného železného kola. Vstupní pulzy motoru musí být napájecí jednotkou převedeny na napětí, pro jednotlivá vynutí elektromagnetů. Rotace hřídele se dosáhne zapínáním jednotlivých vynutí a to se vzájemným fázovým posunem. Při aktivaci prvního magnetu se hřídel nastaví, tak že vrchol jednoho zubu je ve středu daného magnetu. Oproti jinému je však další zub posunut a proto, když dojde k jeho aktivaci a k deaktivaci prvního magnetu, hřídel se stočí o jeden krok.

3.1.4.2 Řízení krokových motorů pomocí 1STEP-5V

Karta 1STEP-5V je modul ze skupiny ET 200S, speciálně určený pro řízení krokových motorů. Karta generuje výstupní pulzy pro napájecí jednotku krokového motoru. Přičemž počet pulzů je přímo úměrný úhlu, o který se motor otočí. Rychlost pohybu je pak dána frekvencí těchto pulzů.



Obrázek 12 – Zapojení karty 1STEP-5V pro řízení krokového motoru

Na Obrázek 12 je znázorněn způsob připojení karty k napájecí jednotce. Mimo nezbytných výstupů v podobě řídicích pulzů a směru, karta vlastní i dva napájecí výstupy a dva digitální vstupy. Funkce těchto dvou vstupů je možné naprogramovat v hardwarové konfiguraci karty.

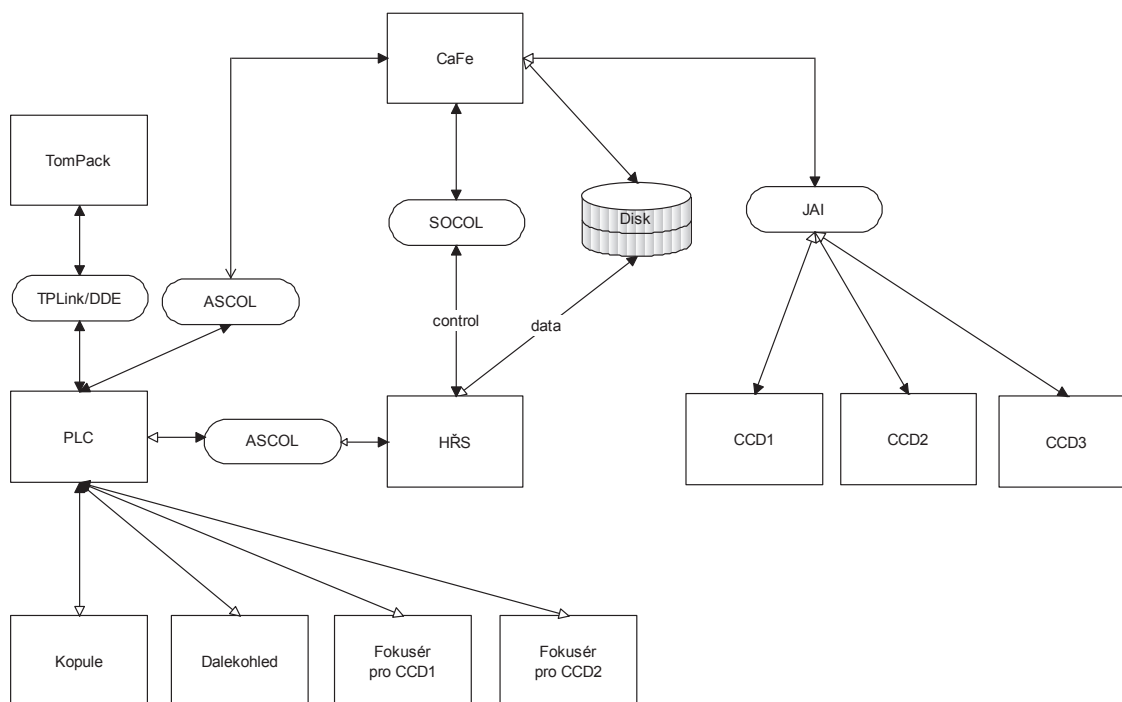
Napájecí jednotka je prostředník mezi řídicí kartou a krokovým motorem, který převádí napěťové pulzy na proud, který způsobuje velmi přesný pohyb motoru.

3.2 Komunikační protokoly

Na Obrázek 13 jsou znázorněny jednotlivé použité komunikační protokoly. Přičemž základem, který využívají všechny komponenty je TCP/IP protokol.

Síťová struktura je rozdělena na vrstvy. Komunikace mezi sítěmi v rámci jedné vrstvy je zajištěna protokoly a používá spojení vytvořené nižší vrstvou.

U všech těchto protokolů je potřeba sjednotit jejich portfolio funkcí, aby byla zajištěna plná funkčnost systému.



Obrázek 13 – Komunikační struktura

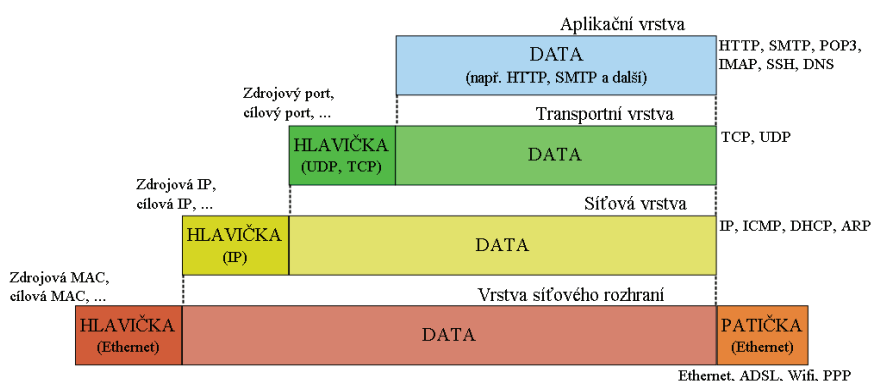
Protokol ASALC je cílem této práce a jeho vytvoření a popis bude rozebráno v další kapitole.

3.2.1 TCP/IP

TCP/IP je celosvětově rozšířená rodina protokolů, používaná a spojovaná zejména s pojmem Internet. Stejně jako jiná síťová spojení, je i TCP/IP rozděleno na vrstvy, které vycházejí z OSI modelu (nejsou však stejné). Jedná se o vrstvu síťového rozhraní, síťovou, transportní a aplikační vrstvu.

Vrstva síťového rozhraní specifikuje přístup k fyzickému přenosovému médiu. Síťová vrstva zajišťuje adresování na úrovni sítě a také rozděljuje data do datagramů neboli paketů. Je používána všemi zařízeními v síti. Transportní vrstva je implementována v koncových zařízeních a slouží pro řízení a kontrolu přenosu dat. Umožňuje také adresování aplikací pomocí portů. Aplikační vrstvu představuje samotný program (proces), který využívá přenosu dat pro své specifické potřeby.

V našem případě je aplikační vrstva realizovaná programem Cafe3.



Obrázek 14 – Zapouzdření dat pomocí TCP/IP

Na obrázku *Obrázek 14* je znázorněno zapouzdření dat pomocí protokolu TCP/IP. Je zde také vidět jaké informace která využívá.

Protokol IP – spadá do síťové vrstvy a zajišťuje tedy adresování v rámci celého Internetu pomocí IP adres, které tato vrstva zapouzdřuje spolu s daty do datagramů. Doručení těchto datagramů IP protokol nezajišťuje.

Protokol TCP – vytváří virtuální spojení mezi koncovými zařízeními. Toto spojení zajišťuje spolehlivost přenosu dat. Protokol spadá do transportní vrstvy.

3.2.2 ASCOL

ASCOL protokol slouží k ovládání dalekohledu. Je postaven na protokolu TCP.

Řídící počítač poslouchá na TCP portech 2000 až 2004, přičemž na jeden port je možné pouze jedno připojení. Každý příkaz se posílá jako ASCII posloupnost znaků, ukončená znakem LF (0x0A) nebo dvojicí CR LF (0x0D 0x0A). Odpovědi jsou zakončeny dvojicí znaků CR LF (0x0D 0x0A).

Teprve po úspěšném přihlášení (příkaz GLLG) je možné používat všechny aktivní a parametrizační příkazy, bez přihlášení fungují pouze dotazy. Jednotlivé parametry musí být odděleny minimálně jednou mezerou.

Pokud klientská stanice nepošle žádný příkaz po dobu 2 minut, řídicí počítač spojení ukončí. Pokud klientská stanice pošle více než 100 znaků bez ukončovacího znaku 0x0A, řídicí počítač také spojení ukončí. V jiných případech řídicí počítač spojení neukončuje.

Po zadání špatného příkazu nebo špatných parametrech odpoví počítač textem ERR<CRLF>.

3.3 FITS soubor

V této kapitole jsem čerpal z literárního zdroje [1].

FITS (Flexible Image Transport System) je standard určený pro ukládání dat v astronomii. Impulsem pro vznik bylo používání CCD kamer a tím přechod na digitální data. Hlavním cílem byla možnost přenášet data z jedné observatoře do druhé. Oproti

jiným formátům (TIFF, JPEG, ...) poskytuje FITS vědecké informace ohledně pořízení snímku. Soubor FITS se skládá ze tří částí, hlavička, obrazová data a doplněk.

Hlavička obsahuje informace, které umožňují popis obrazu uloženého v souboru. Je psána v ASCII znacích, její velikost je násobkem 2880 bytů a je rozdělena do řádků po 80 bytech. Čitelnost hlavičky je pro člověka snadná, avšak jeden chybný znak může zapříčinit nekompatibilitu s počítačovým programem, protože FITS standard je velmi detailně definovaný. Každý řádek obsahuje klíčové slovo, které určuje následující data na řádku. Indikátor hodnoty „=" odděluje klíčové slovo od hodnoty, která je definovaná ve zbytku řádku spolu s komentářem.

Obrazová data jsou reprezentována binárně, a to v přesně definovaných formátech. Tyto formáty musí být striktně dodržovány z toho důvodu, že různé operační systémy zapisují data různými způsoby. Dodržení těchto standardů umožní plnou přenositelnost mezi počítači.

Doplněk slouží pro standardizování počtu bytů ve FITS souboru a skládá se z ASCII znaků „0“.

3.4 MFC Desktop aplikace

Microsoft Foundation Class (MFC) knihovna poskytuje objektově orientovaný obal pokrývající většinu z Win32 a COM API. Může být použita k vytvoření velmi jednoduché desktopové aplikace, je ale více užitečná, když potřebujete vytvořit více komplexní uživatelské rozhraní s mnoha ovládacími prvky. Výhodou MFC je zabaluje složitou architekturu WinAPI do tříd a obtížnou funkcionalitu operací skrývá do maker. Díky makrům je také dosaženo optimalizace výkonu

3.4.1 Windowsové zprávy

Systém automaticky netvoří fronty zpráv pro každé vytvořené vlákno. Místo toho systém vytvoří frontu zpráv pouze pro vlákna, která provádějí operace, které vyžadují fronty zpráv. Pokud vlákno vytváří jedno nebo více oken, musí v něm být smyčka zpráv. Tato smyčka načítá zprávy z fronty zpráv vlákna a odesílá je do příslušných procedur oken.

Vzhledem k tomu, že systém směřuje zprávy do jednotlivých oken v aplikaci, musí vlákno vytvořit alespoň jedno okno před zahájením jeho smyčky zpráv. Většina aplikací obsahuje jediné vlákno, které vytváří okna. Typická aplikace zaregistruje třídu okna pro jeho hlavní okno, vytvoří a zobrazí hlavní okno, a pak začne jeho smyčka zpráv - to vše ve funkci WinMain.

Smyčku zpráv je možné vytvořit pomocí funkcí GetMessage a DispatchMessage. Pokud má aplikace získávat vstupy od uživatele, musí ve smyčce obsahovat funkci TranslateMessage. TranslateMessage překládá virtuální klíč zprávy do znakové zprávy. V následujícím příkladu je ukázáno vytvoření a zaregistrování třídy okna a následné vytvoření samotného okna spolu se smyčkou zpráv.

```

WNDCLASS wc;
// Register the window class for the main window.
wc.style = 0;
wc.lpfnWndProc = (WNDPROC) WndProc;
wc.cbClsExtra = 0;
wc.cbWndExtra = 0;
wc.hInstance = hInstance;
wc.hIcon = LoadIcon((HINSTANCE) NULL, IDI_APPLICATION);
wc.hCursor = LoadCursor((HINSTANCE) NULL, IDC_ARROW);
wc.hbrBackground = GetStockObject(WHITE_BRUSH);
wc.lpszMenuName = "MainMenu";
wc.lpszClassName = "MainWndClass";

if (!RegisterClass(&wc))
    return FALSE;

// Create the main window.
hwndMain = CreateWindow("MainWndClass", "Sample",
    WS_OVERLAPPEDWINDOW, CW_USEDEFAULT, CW_USEDEFAULT,
    CW_USEDEFAULT, CW_USEDEFAULT, (HWND) NULL,
    (HMENU) NULL, hinst, (LPVOID) NULL);

// Start the message loop.
while ( GetMessage( &msg, NULL, 0, 0 ))
{
    TranslateMessage(&msg);
    DispatchMessage(&msg);
}

```

3.4.2 Kritická sekce

Objekt kritické sekce poskytuje synchronizaci podobnou té, kterou poskytuje mutex, s tím rozdílem, že kritická sekce může být použita pouze na vlákna jednoho procesu. Event, mutex a semafor objekty mohou být také použity v jedno-procesové aplikaci, ale kritická sekce poskytuje rychlejší a efektivnější mechanismus pro synchronizaci. Stejně jako u mutexu, mohou být objekty kritické sekce vlastněny pouze jedním vláknem najednou, což je užitečné pro ochranu sdílených prostředků proti souběžnému přístupu. Na rozdíl od mutex objektu, zde není žádný způsob, jak zjistit, zda kritická sekce byla opuštěna.

Počínaje systémem Windows Server 2003 s aktualizací Service Pack 1 (SP1), vlákna nezískávají přístup ke kritické sekci podle pravidla „kdo dřív přijde, ten dřív mele“, ale náhodně. Tato změna výrazně zvyšuje výkon pro většinu kódu. Nicméně, některé aplikace jsou závislé na principu FIFO obsluhy přístupu (první -dovnitř, první - ven) a může fungovat špatně na aktuálních verzích systému Windows.

Použití kritické sekce v kódu je zobrazeno níže.

```

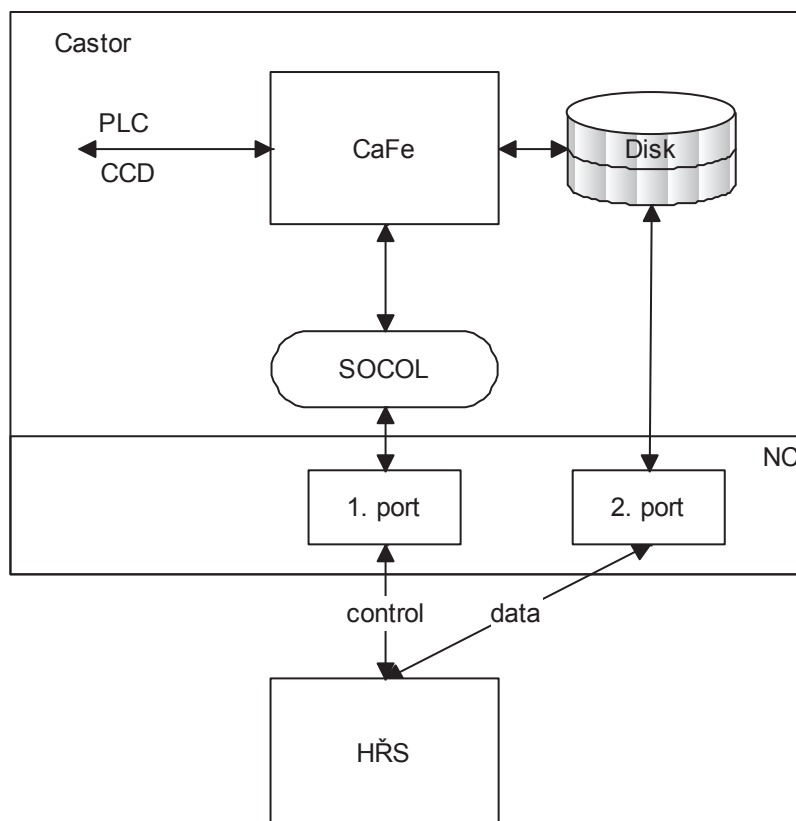
CRITICAL_SECTION m_cs;
InitializeCriticalSection( &m_cs );
//access shared resources
LeaveCriticalSection( &m_cs);
DeleteCriticalSection( &m_cs);

```

4 NÁVRH KOMUNIKAČNÍHO PROTOKOLU

Přidáním nového počítače Pollux do koncepce systému a při existenci požadavku na plně robotický solární teleskop, je nutné vytvořit a implementovat komunikační protokol, který bude zpřístupňovat všechny požadované funkce. Z toho vyplývá, že bude nutné komunikovat jak s programem CaFe3, tak s PLC.

Přehled komunikačních cest je znázorněn na *Obrázek 15*.



Obrázek 15 – Komunikace mezi HŘS a Castorem

Všechna obrazová data přijatá programem CaFe je nutno uložit na harddisk, výjimkou je akorát režim LiveView, při kterém se data neukládají. Přístup k datovému uložišti bude mít HŘS pomocí druhého portu. První port slouží pro řídicí komunikaci. Jeho prostřednictvím jsou hlavnímu řídicímu skriptu zpřístupněny všechny potřebné funkce. Pro správnou funkčnost jsem navrhnul komunikační protokol s vhodnou paletou instrukcí SOCOL (SOort COmmunication Layer).

4.1 Koncepce

Protokol založený na TCP/IP slouží pro komunikaci uživatele s kamerovým systémem. Zahrnuje mimo samotné ovládání kamer i funkce pro nastavování uživatelských parametrů (adresář k ukládání snímků, velikost paketů, login). Uživatel je v roli klienta.

Každý požadavek se posílá jako příkaz složený z ASCII posloupnosti znaků, ukončenou znakem LF (0x0A) nebo dvojicí CR LF (0x0D 0x0A). První čtyři znaky představují klíčové slovo, po kterém následují parametry oddělené mezerou. Odpověď obsahuje informaci o provedení příkazu a v případě dotazovacího příkazu také požadované hodnoty. Odpovědi jsou zakončeny dvojicí znaků CR LF (0x0D 0x0A).

Pokud klientská stanice nepošle žádný příkaz po dobu 2 minut, řídicí počítač spojení ukončí. Pokud klientská stanice pošle více než 100 znaků bez ukončovacího znaku 0x0A, řídicí počítač také spojení ukončí. V jiných případech řídicí počítač spojení neukončuje.

Teprve po úspěšném přihlášení (příkaz GLLO) a po povolení komunikace v programu CaFe3, je možné používat všechny aktivní a parametrizační příkazy, bez přihlášení fungují pouze dotazy. Jednotlivé parametry musí být odděleny minimálně jednou mezerou.

Role serveru v komunikaci je čistě pasivní, tzn., že přijímá příkazy a zasílá pouze příslušnou odpověď. Komunikaci lze podle druhu příkazů rozdělit na asynchronní, pro aktivní a parametrizační požadavky, a na synchronní pro dotazovací příkazy. U asynchronního části server po přijetí příkazu, kontrole syntaxe a parametrů odešle potvrzení o jeho přijetí (viz návratové hodnoty). Příkazy se hromadí ve frontě požadavků a jsou vykonávány v pořadí, v jakém přišly. Dotaz na stav fronty příkazem GLRQ lze zjistit, jestli je již fronta prázdná a jestli byly všechny příkazy vyzvednuty. Zdali se příkaz opravdu provedl, nemůže být z důvodu asynchronní povahy příkazů garantováno, proto je vhodné, aby klient ověřil výsledek dotazem na stav dané veličiny. Dále je nutné brát v úvahu, že nastavení kamer se může nezávisle na probíhající komunikaci měnit (přes GUI, připojení dalšího klienta nebo restart kamery).

Následuje-li po příkazu pro kameru, začínající „CA“, dva a více parametrů, jedná se o aktivní příkaz. Jestliže je příkaz poslán pouze s číslem kamery (jeden parametr), jedná se o dotaz. (Př. CARU 1 1<CRLF> – přijme požadavek na spuštění kamery 1 a vrátí potvrzení přijetí 1<CRLF> . Na příkaz CARU 1<CRLF> vrátí server aktuální stav kamer 1 1<CRLF> (potvrzení přijetí, kamera 1 běží)).

V případě globálních příkazů (začínajících GL) jsou dotazy bez parametrů. Některé globální příkazy jsou pouze dotazovací.

Změny hodnot ovlivňující měřená data (gain, exposure time, ...) lze zadávat i při spuštěné expozici, ale projeví se, až při následujícím snímku.

Návratové hodnoty:

- | | |
|----|---|
| 1 | příkaz i parametry OK, příkaz přijat |
| -1 | příkaz i parametry OK, plná fronta požadavků, příkaz nepřijat |
| -2 | příkaz i parametry OK, kamera odpojena |
| -3 | špatná syntaxe příkazu |
| -4 | špatný počet parametrů |
| -5 | špatný formát nebo rozsah parametrů |
| -6 | uživatel není přihlášen nebo není povolena komunikace v CaFe3 |

4.2 Přehled příkazů

Požadované funkce se odvíjejí od samotných požadavků na systém. Můžeme je rozdělit na příkazy pro komunikaci s PLC, sem spadá ovládání kopule, fokusérů apod. A příkazy vycházející z ovládacího programu CaFe, které souvisejí s řízením kamer, zpracováním dat a síťovým provozem.

Tabulka 2 – Seznam příkazů pro SOCOL – funkce CaFe

Příkaz	Název	Popis
CARU	CAmera RUn	Spuštění expozice kamery
CAET	CAmera Exposition Time	Nastavení doby expozice kamery
CAGM	CAmera – GAin	Nastavení zesílení kamery
CAIS	CAmera–Interval between Snaps	Perioda snímků
CANS	CAmera – Number of Snaps	Počet snímků v sekvenci
CABM	CAmera-optical Black transfer Mode	Nastaví optical black transfer mode pro referenční účely
CAFC	CAmera – Flat field corection	Flat-field korekce
CAPS	CAmera - Partial Scan	Nastavení poměru částečného skenování
CAMD	CAmera – MoDe	Měřicí mód
CAPF	CAmera – Pixel Format	Pixelový formát
CABN	CAmera - BiNning	Nastavení počtu slučovaných pixelů
CASP	CAmera – Snap Path	Cesta k ukládaným snímkům
CAHP	CAmera – Hist Path	Cesta k ukládaným histogramům
CFAP	Camera FITS – Active area Parameters	Parametry pro FITS hlavičku, které se mění s nájezdem na aktivní oblast
CFDP	Camera FITS – Daily Parametrs	Parametry pro FITS hlavičku, které se mění denně
CLLO	Camera Liveview–Liveview On	Zapnutí LiveView
CLCO	Camera Liveview- Cross On	LiveView – zapnutí kříže
CLCS	Camera Liveview- Cross Size	LiveView – velikost kříže
CLRI	Camera Liveview- Rotate Image	LiveView – rotace snímku
CLFI	Camera Liveview- Flip Image	LiveView – překlopení snímku
CLHO	Camera Liveview–histogram On	LiveView – zapnutí histogramu
GLTL	GLobal - Trafic Level	Vrací vytížení sítě
GLPS	GLobal - Packet Size	Nastavení velikosti paketu
GLLO	GLobal LOGin	Funkce pro přihlášení
GLCE	GLobal Control Enable	Info o povolení komunikace přes TCP/IP
GLCS	GLobal - Camers Synchronous	Synchronní snímání kamer
GLRQ	GLobal – Request Queue	Dotaz na stav fronty požadavků

Příkazy pro PLC vycházejí z protokolu ASCOL, pro náš systém je však nutné přidat funkce nové. Přehled používaných příkazů z ASCOLu plus dodatečné funkce u vedené v *Tabulka 3*. Podrobný seznam těchto, i předcházejících příkazů, je uveden v příloze.

Tabulka 3 - Seznam příkazů pro PLC – ASCOL s dodatkem

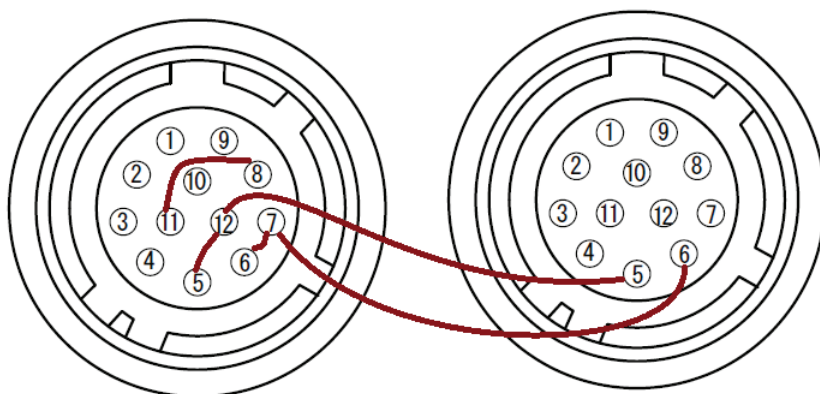
TEON	Telescope ON or off	Zapnutí a vypnutí teleskopu
TETR	Telescope TRack	Povolení pohybu
TEPA	Telescope PArk	Zaparkování teleskopu
TESY	Telescope SYnchronization	Synchronizace teleskopu
TSRA	Telescope Set Right ascension and declination Absolute	Nastavení souřadnic rektascenze a deklinace
TGRA	Telescope Go to Right ascension and declination Absolute	Nájezd na souřadnice rektascenze a deklinace
TSHA	Telescope Set new Hour and declination axis Absolute	Nastavení mechanických souřadnic
TGHA	Telescope Go to new Hour and declination axis Absolute	Nájezd na mechanické souřadnice
TSCR	Telescope Set Correction of the Refraction	Zapínání korekce refrakce
TSCM	Telescope Set Correction of the telescope Model	Zapínání korekce chybového modelu
TSGM	Telescope Set Guiding Mode	Nastavení do guide režimu
TSGV	Telescope Set Guiding Value	Nastaví korekci pro guide
TSGR	Telescope Set Guiding value Relative	Nastaví změnu korekce pro guide
TSS1	Telescope Set Speed 1	Nastavení rychlosti ručního pohybu 1
TRS1	Telescope Read Speed 1	Čtení rychlosti ručního pohybu 1
TSS2	Telescope Set Speed 2	Nastavení rychlost ručního pohybu 2
TRS2	Telespoce Read Speed 2	Čtení rychlosti ručního pohybu 2
TSS3	Telescope Set Speed 3	Nastavení rychlost ručního pohybu 3
TRS3	Telescope Read Speed 3	Čtení rychlosti ručního pohybu 3
TESS	TElescope Select Speed	Volba rychlosti ručního pohybu
TMDM	Telescope Manual Declination Move	Manuální pohyb po deklinační ose
TMRM	Telescope Manual Right ascension Move	Manuální pohyb rektascenze
TGSC	Telescope Go to Sun Coordinates	Nájezd na slunce
TERE	TElescope – Read Epoch	
DOLS	DOme Left Slit open or close	Ovládání levé kopule
DORS	DOme Right Slit open or close	Ovládání pravé kopule

DOSM	DOme Stop Movement	Zastavení pohybu kopule
DOSH	DOme SHade open or close	Ovládání clony
DORF	DOme Read Frozen state	Dotaz na přimrznutí kopule
DOLO	DOme Light On or off	Ovládání světla v kopuli
FOMO	FOcus MOve absolute	Ovládání fokusérů
FOIN	FOcus INitialization	Inicializace fokusérů
FOSM	FOcus Stop Movement	Zastavení pohybu fokusérů
FORP	FOcus Read Position	Čtení pozice fokusérů
USRS	User Speed – Read speed	Čtení uživatelské rychlosti
USSS	User Speed – Set Speed	Nastavení uživatelské rychlosti
USON	User Speed – ON or off	Zapnutí uživatelských rychlostí
UCRC	User Corection– Read Corection	Čtení uživatelských korekcí
UCSC	User Corection – Set Corection	Nastavení uživatelských korekcí
GLVE	GLobal VErSION	Vrací aktivní komponenty systému
GLLG	GLobal LoGin	Přihlášení
GLLL	GLobal read Latitude and Longitude	Vrací zeměpisnou šířku a délku
GLUT	GLobal read UTc	Vrací UTC
GLSD	GLobal read SiDereal time	Vrací aktuální local apparent sidereal time
GLMD	GLobal read Meteo Data	Vrací meteo data
GLPO	GLobal Power On or off	Ovládání napájení jednotlivých prvků
GLRP	GLobal Read Power	Vrací stav napájení
GLSC	GLobal read Sun Coordinate	Vrací souřadnice slunce
GLCS	GLobal Central Stop	Globální STOP
GLST	GLobal STate	Stav systému
GLAS	GLobal Aditonal State	Stav zbytku systému

5 SYNCHRONIZACE KAMER

Dalším z požadavků pro řídicí systém bylo synchronní snímání obou kamer. Důvodem tohoto požadavku je, že kdyby se obě kamery spouštěly ovládacím programem CaFe z počítače mohlo by ve vytížené síti dojít ke zpoždění doručení a vyhodnocení paketů s informací o startu expozice. Pro splnění této podmínky se nabízely dvě možnosti. První bylo použití speciálních časovacích modulů pro PLC Simatic. Tyto karty generují po příchodu řídicího signálu synchronizované impulzy na svých výstupech. Toto řešení by zajišťovalo vysokou robustnost, přesnost a v případě odpojení jedné kamery by druhá dále fungovala. Nevýhodou je časová i finanční náročnost spočívající v rozšíření PLC o tyto moduly a vytvoření kódu pro jejich ovládání.

Druhou možností, tou kterou jsme zvolili, je časování pomocí interních obvodů a GPIO kamer. V tomto případě se jedna kamera stává řídicí a druhá řízená. Nevýhodou tohoto uspořádání je, že v případě odpojení nebo poruchy řídicí kamery nebude možné v synchronizovaném režimu tuto druhou kameru spustit.

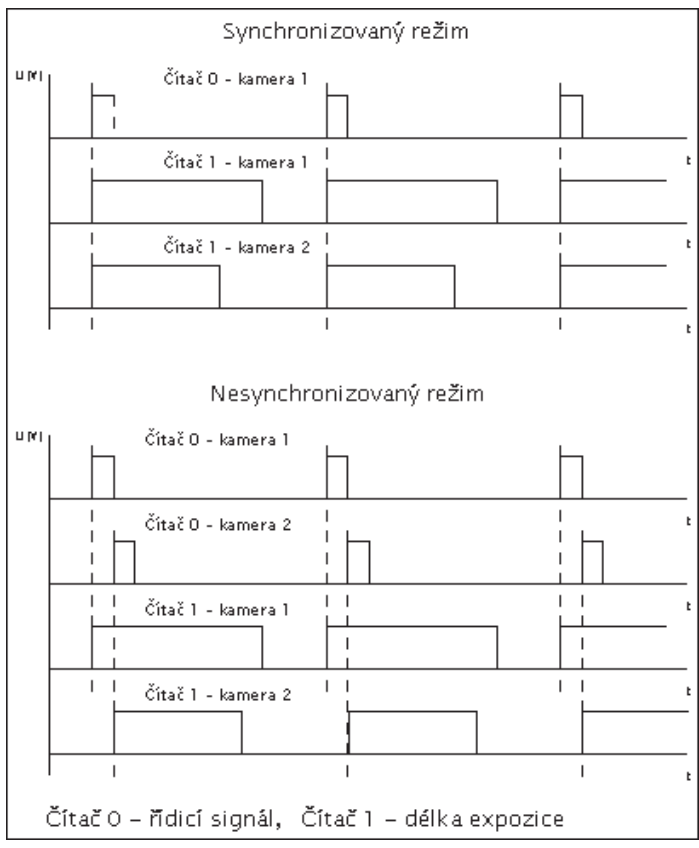


Obrázek 16 – Synchronizace kamer (1. řídicí, 2. řízená kamera)

11: DC+, 12: DC-, 8: Opt OUT+, 7: Opt OUT-, 6: Opt IN+, 5: Opt IN-

Schéma propojení konektorů kamer, za účelem jejich synchronizace, je zobrazeno na *Obrázek 16*. Výstupní port Opt OUT řídicí kamery je napájen z portu 11, na který je vyvedeno napájecí napětí kamery. Tento výstupní port je ovládán (spínán) vnitřním čítačem pulzů řídicí kamery. Perioda těchto pulzů je dána požadovanou periodou snímků. Tento synchronizační signál je přiveden na vstupy Opt IN obou kamer. Náběžnou hranou tohoto signálu je spouštěn druhý čítač, jehož šířka pulzu odpovídá délce expozice snímků. Doba expozice může pro jednotlivé kamery lišit, proto je nutné tyto druhé čítače nastavovat pro každou kameru zvlášť.

Průběh těchto signálů je zobrazen na obrázku *Obrázek 17*. Je zde zřejmé, že v případě nesynchronizovaného režimu, kdy jsou čítače určující dobu expozice nulovány pro každou kameru zvlášť, může dojít k vzájemnému časovému posunu začátků expozic.



Obrázek 17 – Průběhy signály při synchronizaci kamer

6 OVLÁDACÍ PROGRAM CAFE3

V této kapitole se budeme zabývat návrhem a vývojem ovládacího programu CaFe3. Před samotným návrhem je třeba si uvědomit, jaké funkce má software mít a jaké požadavky jsou na něj kladeny. Část těchto požadavků jsme si přiblížili v předcházejících kapitolách. Zde se na ně podíváme podrobněji.

Jelikož bude program pracovat s více datovými toky a více komunikačními úrovněmi, je vhodné řešit jednotlivé části v samostatných vláknech. Rozdělení na více vláken lépe využije výkon vícejádrových procesorů, které jsou dneska již standardem. Dále je pak zajištěno, že v případě blokování vlákna, čekáním na systémové prostředky, mohou ostatní vlákna pokračovat v běhu. Na druhou stranu je však výhodné počet vláken minimalizovat, a to z důvodu problematického synchronizování vláken, komunikaci mezi nimi a sdílení společné datové oblasti.

6.1 JAI SDK

Jedná se o vývojový balíček pro práci s kamerami od výrobce JAI. Je strukturován do série modulů, kde pojmenování jednotlivých API funkcí odpovídá modulu, ke kterému funkce patří. Nejvýznamnějším je *Factory* modul, který představuje hlavní přístupový bod pro zbytek funkcí. Všechny aplikace musí začínat vytvořením a otevřením jedné instance *Factory* před používáním ostatních funkcí. Samozřejmostí je pak zavření této instance před ukončením aplikace. Po úspěšném otevření je možné hledat na síti GigE Vision zařízení a to funkcí *J_Factory_UpdateCameraList()*. Dále se zjistí počet nalezených kamer *J_Factory_GetNumOfCameras()* a pro každou nalezenou kameru dostaneme její ID, které slouží pro její připojení, funkcí *J_Factory_GetCameraIDByIndex()*. Posledním krokem pro přípravu kamery k ovládní, je její otevření pomocí *J_Camera_Open()*. Tato funkce vrací handle pro práci s kamerou. Ten je po ukončení nutné zavřít funkcí *J_Camera_Close()*.

Nyní máme přístup ke kameře umožněn přes sbírku GenICam uzlů. Příklad uveden níže.

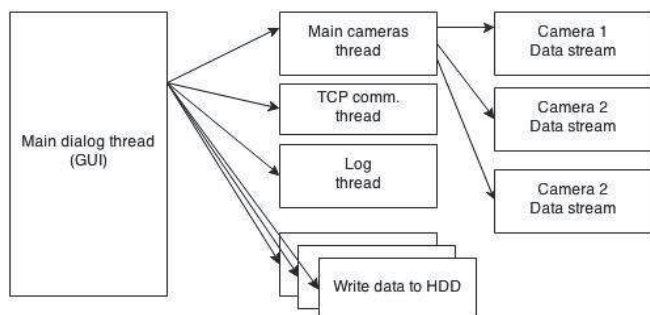
```
J_Camera_SetValueString(m_hCam, (int8_t*)"ExposureMode",  
(int8_t*)"TriggerWidth")
```

Vstupní parametr *m_hCam* představuje handle kamery. Řetězec „ExposureMode“ je název uzlu podle standardu GenICam a „TriggerWidth“ představuje jednu z možných vstupních hodnot tohoto uzlu. Obdobným způsobem pak lze z kamery data vyčítat s tím rozdílem, že název funkce *Set* je místo toho *Get*.

6.2 Struktura programu

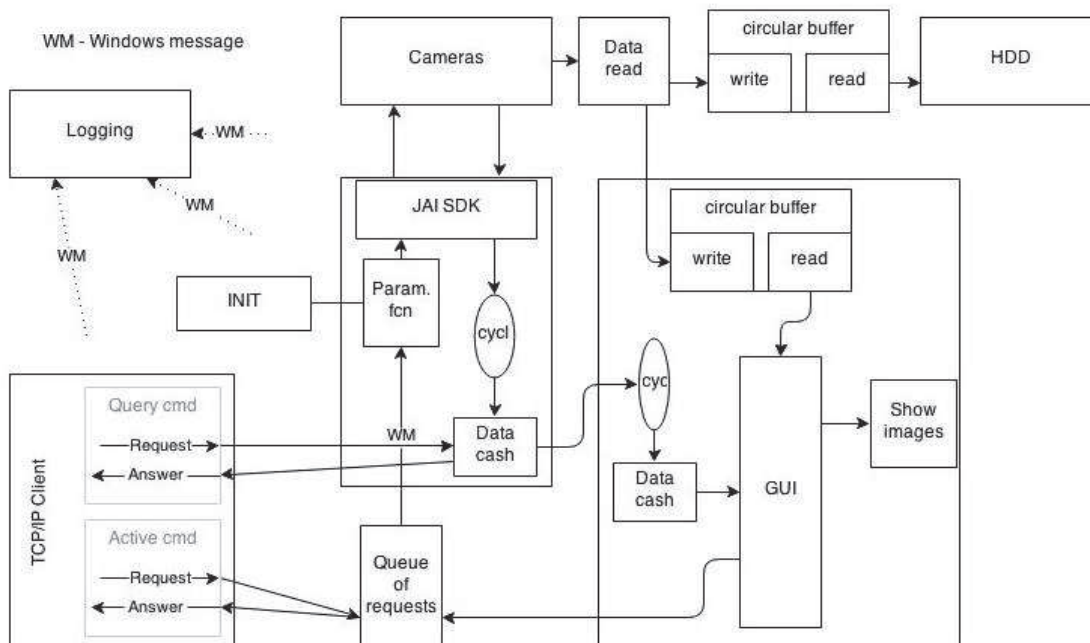
Po konzultacích a po testech rychlostí jednotlivých operací a postupů, jsem došel k finální koncepci programu. Po startu hlavního dialogového okna, které spravuje uživatelské rozhraní, se vytvoří vlákna pro správu dílčích činností. Nejdůležitější vlákno

obstarává kamery, jejich připojení, nastavování a vyčítání parametrů. Při startu expozice, je pro každou kameru vytvořeno vlastní vlákno, které zajišťuje vyčítání obrazových dat z kamer, a data následně ukládá do kruhového bufferu. Z těchto bufferů jsou pak data pomocí dalších vláken vyčítána a ukládána na pevný disk. Neméně důležité je vlákno zajišťující TCP/IP komunikaci s klienty i s automatem v roli serveru. Poslední vlákno zajišťuje vyčítání logovacích řetězců z dalšího kruhového bufferu a ukládá je do logovacího souboru na disku. Jednotlivá vlákna jsou zobrazena na *Obrázek 18*. V GUI, kamerovém a komunikačním vláknech jsou vytvořeny windowsové třídy se smyčkami zpráv. Tyto vlákna tedy běží po celou dobu chodu programu a čekají na událost označenou příslušnou zprávou. Zprávy jsou podrobněji popsány v teoretickém rozboru.



Obrázek 18 – Vlákenná struktura programu CaFe3

Nejpodstatnější částí programu je ovládání tří kamer. Jejich parametrizaci zajišťuje vlákno zmíněné již výše. Pro přímý přístup ke kamerám používám softwarový vývojářský balíček JAI SDK, který výrobce ke kamerám dodává. Jedná se o soubor funkcí a datových typů určených pro práci s kamerami a s obrazem. Vlákno po přijetí zprávy vyjme požadavek z fronty a vykoná jej.



Obrázek 19 – Datové a komunikační kanály programu CaFe3

Aktuální stav kamer je kontinuálně vyčítán a ukládán do datové struktury, ke které pak následně přistupují ostatní vlákna. Případná kolize je ošetřena kritickou sekcí. Při startu programu se do kamer nahrají data z inicializační databáze, do které se změny provedené za chodu ukládají.

Komunikace uživatele s vláknem zajišťujícím obsluhu kamer, ať už prostřednictvím grafického rozhraní nebo přes TCP/IP, je založena na protokolu popsaném v předcházející kapitole. Aktivní požadavky (tj. s cílem zasahovat do nastavení nebo stavu kamer) se řadí do fronty. Vždy při zařazení nového požadavku je poslána windowsová zpráva danému oknu, které zprávu vyzvedne, přeloží a vykoná. Ten kdo požadavek do fronty umístil, však nemá informaci o tom, jestli byla akce vykonána. To si je nutné si ověřit načtením stavu datové struktury reprezentující stav kamer. Načítání dat z kamer a jejich ukládání probíhá v podstatě nezávisle na zbytku programu v samostatných vláknech.

6.3 Rozbor jednotlivých funkčních částí

Pro každé vlákno je vytvořena jedna instance dané třídy, v jejímž konstrukturu je vlákno vytvořeno pomocí funkce `AfxBeginThread`. Před jejím voláním je však nejprve potřeba inicializovat třídu okna (datový typ `WNDCLASS` - Window class) a následnou registraci pomocí funkce `RegisterClass(*WNDCLASS)` ji zpřístupnit. Posledním krokem je pak vytvořit samotné okno ve vláknové proceduře a to funkcí `CreateWindow(...)`. Tento postup nám umožňuje v této proceduře vytvořit pumpu zpráv.

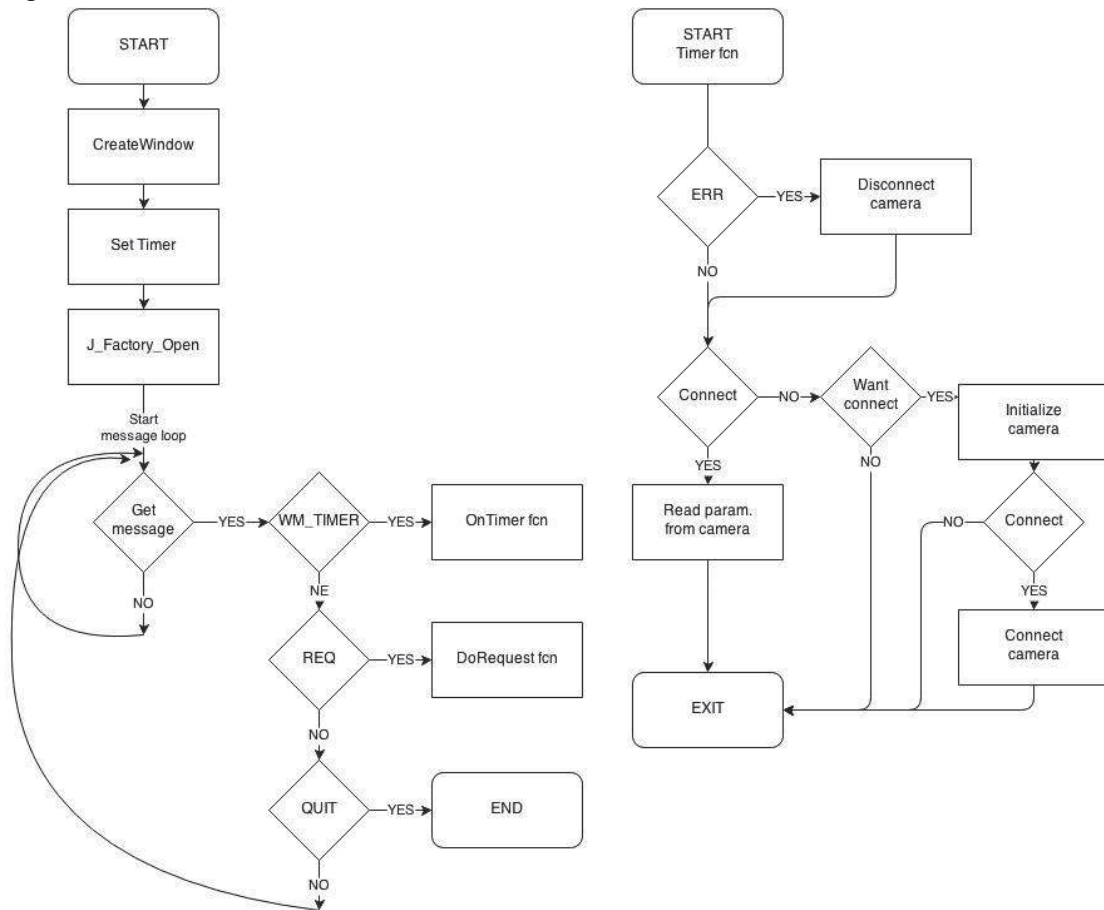
6.3.1 Vlákno pro obsluhu kamer

Vlákno pro obsluhu kamer je vytvořeno hned po startu aplikace v jejím hlavním vlákně. Jelikož chceme, aby toto vlákno zpracovával zprávy, je nutné v něm podle již výše zmíněných postupů vytvořit smyčku zpráv. Tato smyčka běží, dokud není přerušena windowsovou zprávou `WM_QUIT`. Vybírání zpráv z fronty je prováděno funkcí `GetMessage()`, která výkonově nezatěžuje procesor.

Pro dané okno je nastaven časovač s periodou 10 ms, který generuje zprávy typu `WM_TIMER`. V obslužné rutině časovače je prováděno vyčítání parametrů z připojených kamer a také je volána funkce pro připojení kamer. Nejprve si popíše tuto funkci. Před samotnou snahou o připojení je testován alarmový bit každé kamery, v případě, že je nastaven na „1“ dojde k odpojení kamery. Požadavek na její připojení však zůstává, a tak v případě, že je to kamera umožňuje, je znovu připojena. Nastavení alarmového bitu nastává v okamžiku, kdy se nepovede zápis nebo čtení dat z kamery. Poté následuje funkce pro připojení kamer. Jelikož je prohledávání sítě a aktualizování *Factory* objektu pomalé a narušuje již vytvořená spojení, snažíme se mu co nejvíce vyhybat. Nejprve otestujeme, jestli není připojena některá kamera, kterou mít připojenou nechceme. V takovém případě je daná kamera odpojena. Pak se provede kontrola, zdali některá kamera, kterou chceme připojit, není připojena. V takovém

případě je nutné prohledání sítě a znovu připojení všech nalezených kamer. Prohledávání a připojování kamer, je časově náročná operace, která se může protáhnout na dobu delší, než je čas vyhrazený pro timer (v našem případě 10 ms). Po přetáhnutí tohoto času dochází k vygenerování další zprávy WM_TIMER a funkce pro připojení kamer by se volala znovu. Tomu je nutné zabránit pomocným bitem, který signalizuje, že je již jedna tato funkce zavolána.

Mimo připojovací volání funkce *ConnectCameras()* slouží rutina časovače pro vyčítání dat z kamer. Jelikož některé funkce pro vyčítání mohou opět být časově náročné, je situace řešena přes sekvencer. Při načtení jednoho parametru se sekvencer posune do dalšího kroku, a tudíž je při dalším volání načten jiný a pouze jediný parametr. Počet parametrů, které se z kamery vyčítají, je devět. Bereme-li v úvahu, že jsou připojeny všechny tři kamery, pak doba pro načtení všech parametrů všech kamer odpovídá 270 ms.



Obrázek 20 – Struktura vlákna pro obsluhu kamer

Do fronty zpráv jsou mimo zprávy od časovače umísťovány také požadavky od ostatních vláken programu. A to zejména od hlavního vlákna spravujícího uživatelské rozhraní a od vlákna pro TCP/IP komunikaci. Tyto požadavky se vztahují na nastavení kamer nebo parametrů s nimi spojených. Přenos informací mezi vlákny je založen na komunikačním protokolu. Stejně jako při TCP/IP komunikaci je požadavek rozdělen na klíčové slovo a na parametry. Funkce nejprve vyhodnotí klíčové slovo a následně

zkontroluje správný počet parametrů. V případě, že klíčové slovo neexistuje nebo je nesprávný počet parametrů, funkce vrací hodnotu „-1“. Jestliže je syntaxe správná, provede se příslušná nastavovací funkce.

Jednou z nedůležitějších a zároveň nejsložitějších je nastavování doby expozice a periody snímků. Požadovaných hodnot je dosaženo správným nastavením dvou čítačů hodinových pulzů. Kromě kontroly horní a dolní meze je nutné přiřadit nastavovanou dobu expozice do jednoho ze tří intervalů. Pro každý interval je daná jiná hodnota předděliče hodinového kmitočtu. Větší hodnota umožňuje nastavení vyšší periody snímků, ale nižší rozlišení. Pro maximální hodnotu předděliče 4096 je přesnost nastavení 0,1 ms a maximální doba periody 37 025 ms. Doba periody pak odpovídá periodě čítače 0 a doba expozice šířce pulzu čítače 1 (*Obrázek 17*). Bližší popis nastavovaných parametrů je uveden v manuálu k programu.

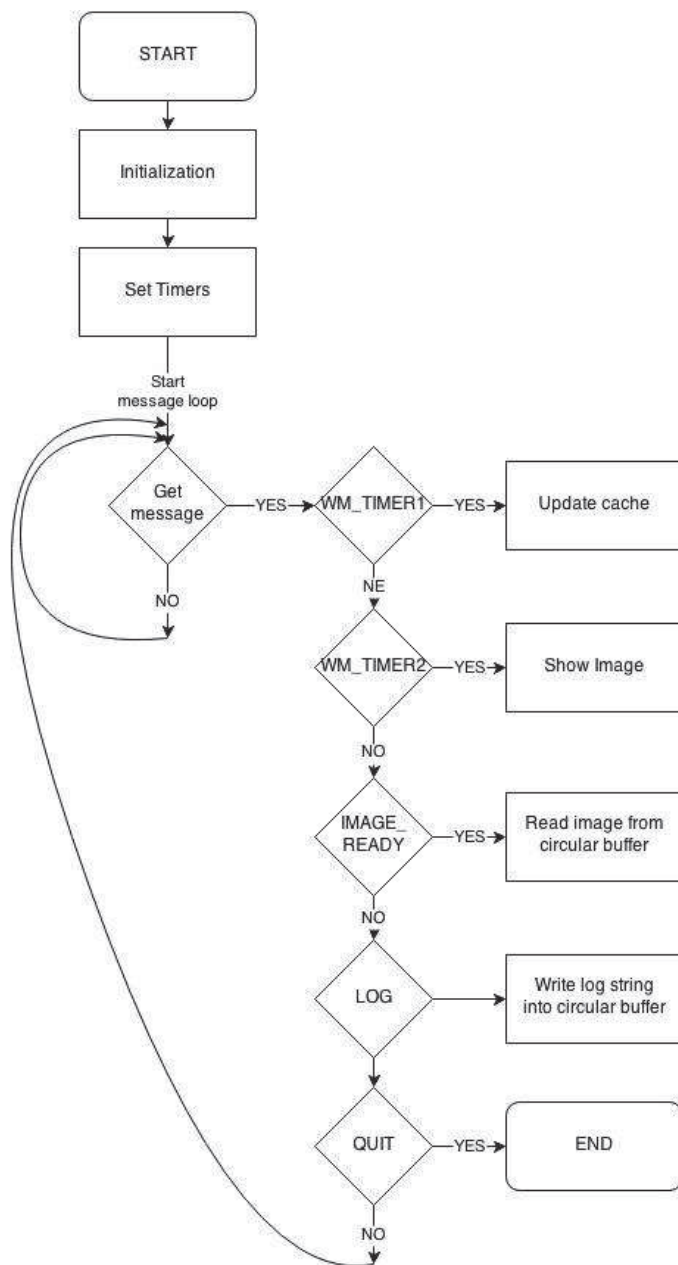
6.3.2 GUI

Vlákno obsluhující grafické rozhraní je zároveň hlavním vláknem aplikace. Z toho důvodu zde není nutné vytvářet smyčku zpráv, protože tu nám i spolu s objektem okna automaticky vytváří MFC knihovna.

Inicializace vlákna v tomto případě zahrnuje vykreslení dialogu, vytvoření objektů pro ostatní vlákna a nastavení dvou časovačů. První slouží pro aktualizaci lokální cache paměti. Druhý časovač pak pro vykreslování snímků v případě zapnutého LiveView. Do fronty zpráv jsou mimo událostí generovaných uživatelským rozhraním a časovači vkládány uživatelské zprávy MESSAGE_LOG a MESSAGE_IMAGE_READY. První zpráva slouží pro vytváření záznamů o chodu aplikace a může být generována všemi ostatními vlákny. Více o zápisu logovacích záznamů v kapitole Zápis dat na disk.

Druhá zpráva slouží pro signalizaci, že vlákno obstarávající vyčítání obrazových dat z kamery uložilo snímek do příslušného kruhového bufferu. Jestliže vlákno přijme tuto zprávu, vyzvedne data z bufferu a uloží si je do vlastní paměti. Z toho vyplývá, že toto vlákno má k vždy k dispozici poslední naměřený snímek. Po startu aplikace má snímek nulové hodnoty. S načítáním snímků souvisí i jejich vykreslování. V případě, že je zobrazování snímků aktivní, dochází s periodou druhého časovače (100 ms = 10 snímků za vteřinu) k jejich vykreslování. Perioda vykreslování je pevná, i když se liší perioda měřených snímků. Je to z důvodu plynulého vykreslování kříže a pozice kurzoru do zobrazovaného snímku.

První časovač s periodou 200 ms, slouží pro aktualizaci lokální paměti vlákna. V jednom cyklu jsou porovnávány všechny proměnné s globálními hodnotami, ať už se jedná o kamerová data nebo informace o TCP/IP komunikaci. V případě, že se lokální a globální hodnota liší, dojde k aktualizaci lokální proměnné a k upravení zobrazení příslušné položky dialogového okna.



Obrázek 21 – Struktura hlavního GUI vlákna

6.3.3 TCP/IP komunikace – server/klient

Vlákno pro obsluhu TCP/IP komunikace zastupuje dvě role. Za prvé slouží jako klient v komunikaci s automatem, který řídí chod teleskopu a kopule. V druhém případě je v roli serveru a zajišťuje komunikaci s připojenými klienty za účelem ovládní kamer. Počet připojených klientů je teoreticky neomezený.

Jelikož se nejedná o hlavní vlákno, je nutné vytvořit smyčku zpráv. Před jejím vykonáváním se provede inicializace Windows Socket API (WSA) a vytvoří potřebné sockety. Jeden pro Listen Socket pro následné připojení klientů a jeden Connect Socket pro připojení se k automatu. V obou případech se jedná o asynchronní sockety, jejichž

vlastností je, že neblokuje běh vlákna. Při inicializaci se danému socketu nastaví, na které události má reagovat a generovat windowsové zprávy. Nastavení do neblokovačím režimu je ukázáno níže na socketu pro naslouchání.

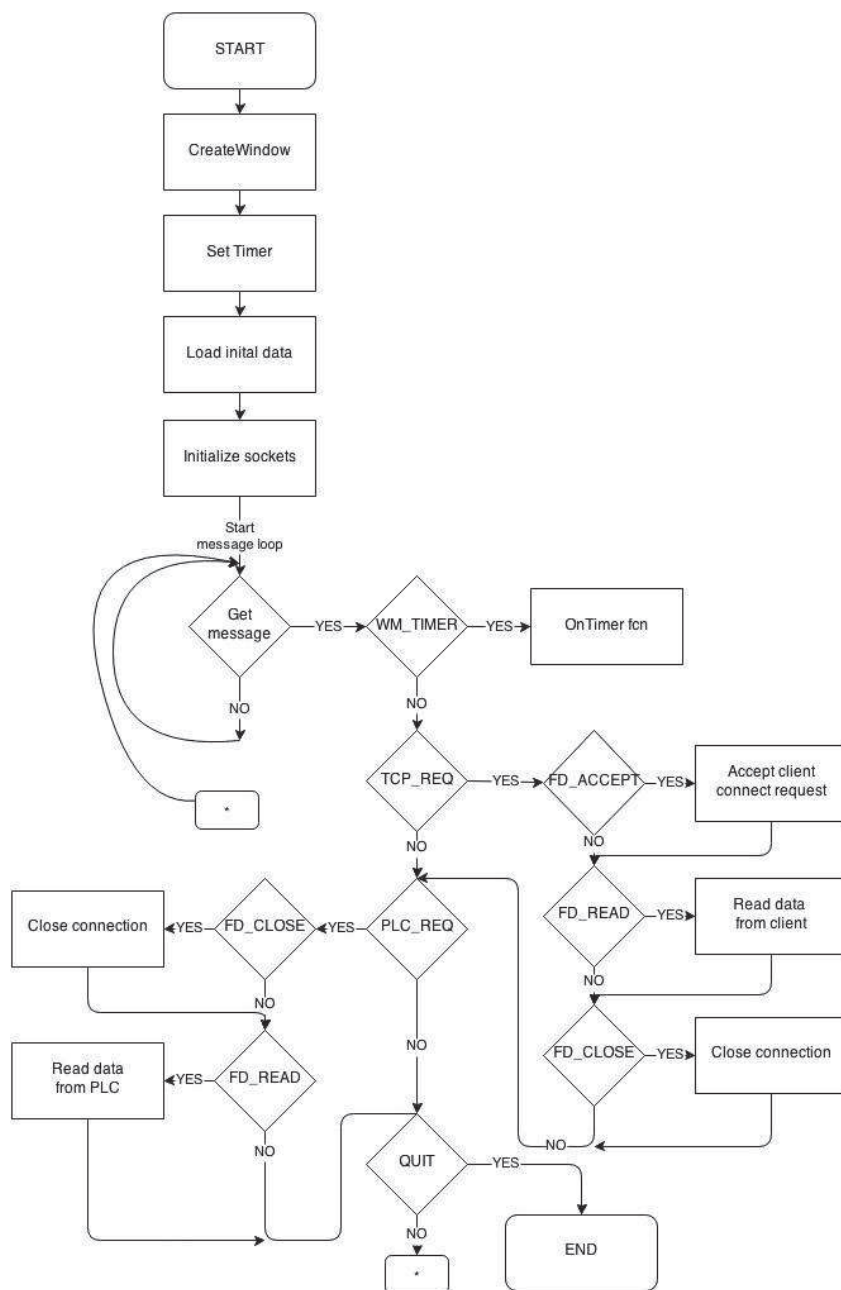
```
WSAAsyncSelect(ListenSocket, m_hwnd, MESSAGE_TCP_REQ,  
(FD_ACCEPT | FD_CONNECT | FD_READ | FD_CLOSE));
```

Parametr *m_hwnd* je ukazatel na okno, kterému se má zasílat zpráva `MESSAGE_TCP_REQ`. Poslední parametr představuje události, na které má socket reagovat.

Před zahájením komunikace si vlákno načte z databáze informace v podobě IP adresy a hesla k automatu, číslo portu, na kterém má naslouchat a heslo pro serverovou komunikaci.

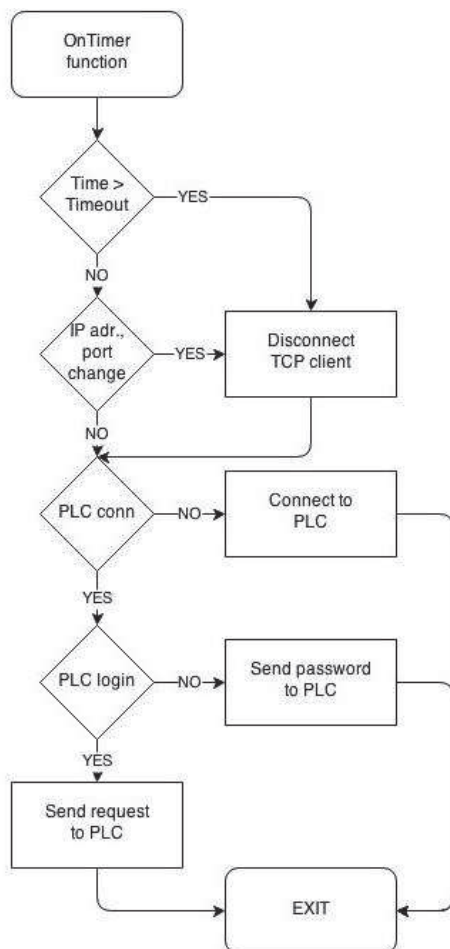
Po přijetí zprávy od socketu, je nutné na základě jejího parametru určit, o jaký typ události se jedná. V případě Listen socketu mohou nastat tyto události. Snaha klienta o připojení neboli požadavek na potvrzení spojení `FD_ACCEPT`. Pro každého nově připojeného klienta je vytvořen nový objekt, který v sobě nese informace o klientovi. V případě, že klienta zašle data, je přijata zpráva `FD_READ`. Vlákno data uloží do bufferu náležícímu k objektu daného klienta. Po přijetí dat, je tento buffer vyhodnocen. V případě, že jsou nalezeny znaky „`\r\n`“ dojde k vyhodnocení příkazu a případně zaslání požadavku do kamerového vlákna. V případě, že délka znaků v bufferu přesáhne 100 nebo je doba neaktivity klienta delší než 2 minuty, dojde k jeho odpojení. Po přijetí dat server okamžitě odpoví podle pravidel komunikačního protokolu `SOCOL`.

Funkce klienta se liší převážně v tom, že jeho role v komunikaci je aktivní. Program po úspěšném připojení a přihlášení k automatu neustále zasílá požadavky. V tomto případě přináší zavedení asynchronní komunikace komplikace. Po zaslání požadavku na automat program nečeká na jeho odpověď, ale pokračuje dál v činnosti. Aby nedocházelo k zasílání dalších požadavků do přijetí odpovědi, musí být nastaven informační bit na jedničku. Po přijetí zprávy `FD_READ` je tento bit shozen a přijatá data jsou vyhodnocena na základě zaslání požadavku. V jednom cyklu časovače se tedy může vykonávat pouze jeden dotaz nebo příkaz pro automat. Proto je nutné správně nastavit prioritu, se kterou se budou dané činnosti vykonávat. Přednost mají požadavky od hlavního dialogového okna, tedy od uživatele. Jedná se o příkazy pro ovládání fokusérů.



Obrázek 22 – Struktura vlákna pro TCP/IP komunikaci

Ve funkci volané při přijetí zprávy od časovače, se testuje doba neaktivity připojených klientů. V případě změny portu, na kterém má server naslouchat, dojde k okamžitému odpojení všech klientů. Dále je pak testováno připojení a přihlášení k automatu. V případě, že není automat připojen, program se neustále snaží navázat spojení a prochází postupně porty určené pro protokol ASCOL (2000 až 2004). Jestliže je automat připojen a program přihlášen, zasílají se mu cyklicky požadavky na stav a polohu krokových motorů fokusérů.

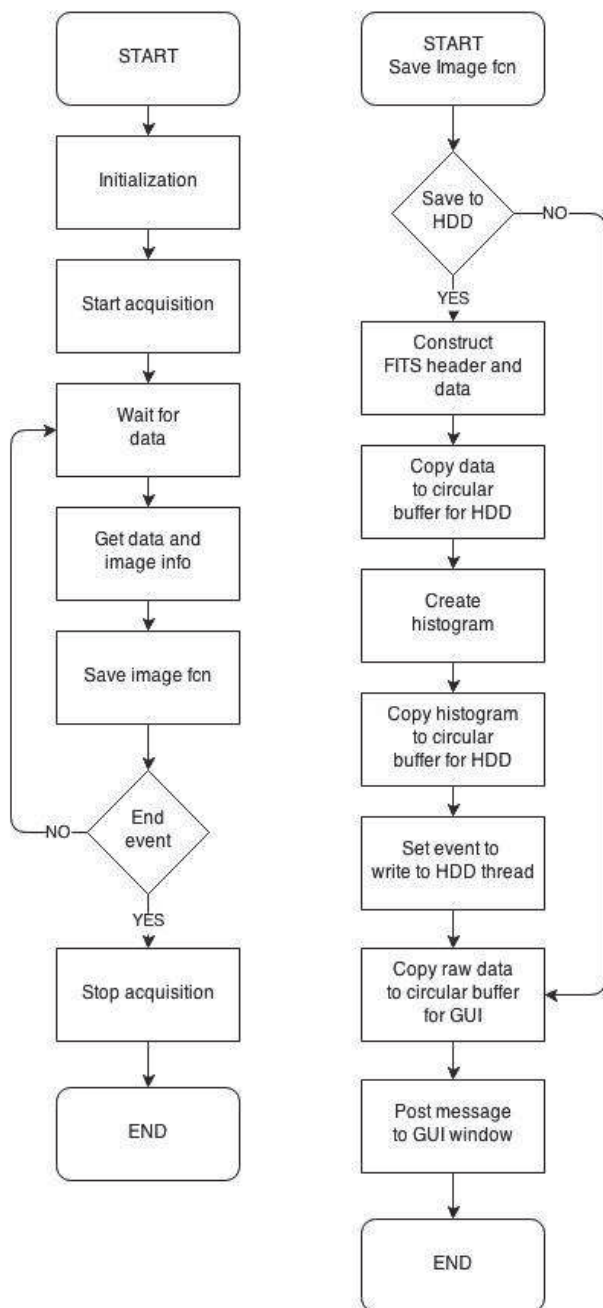


Obrázek 23 – Struktura vlákna pro TCP/IP komunikaci – periodická funkce

6.3.4 Stream dat z kamery

Při zahájení měření kamery je spuštěno nové vlákno, které má na starosti vyčítání dat a jejich ukládání do kruhového bufferu. Po startu vlákna čeká nadřazené vlákno na jeho inicializaci, a až poté pošle příkaz do kamery pro zahájení snímkování. Inicializace spočívá v nastavení podmínek a událostí, které budou generovány při připravení dat na kameře. Po zahájení měření pracuje vyčítací vlákno ve smyčce, kde na začátku vždy čeká na nově přijatá data. Jsou-li data k dispozici, signalizuje daný event. Kromě načtených dat, jsou získávány informace o daném snímku, jako je pixelový formát, velikost a offset snímku, počet ztracených paketů a další.

Toto vlákno snímky neukládá na disk, ale pouze do kruhového bufferu. Jelikož formát těchto dat již odpovídá formátu, ve kterém se budou ukládat, je potřeba vygenerovat FITS hlavičku a uložit ji do bufferu spolu s daty. Z obrazových dat je pak na základě informace o pixelovém formátu kamery vypočten histogram. K histogramu je také vytvořena FITS hlavička, která se však kromě názvu souboru nijak neliší. Po naplnění kruhových bufferů pro snímky a pro histogramy je nastavena událost pro zapisovací vlákno dané kamery. Posledním krokem je uložit snímku v původním formátu do bufferu pro dialogové okno a zaslání zprávy s informací o novém snímku.



Obrázek 24 – Struktura vlákna pro vyčítání dat z kamer

6.3.5 Zápís dat na disk

Pro každou kameru je vytvořené jedno vlákno pro zapisování dat na disk. Na rozdíl od ostatních se jedná o synchronní vlákno čekající na eventy nastavované po uložení dat do kruhového bufferu streamovacím vláknem. Po přijetí eventu nebo po 3 vteřinové prodlevě dojde k vyčtení dat z bufferu do pomocného datového pole. Toto pole je rozděleno na jednotlivé snímky, které jsou následně ukládány zvlášť. Před uložením se provede načtení jména obrázku z jeho FITS hlavičky. S tímto jménem jsou pak data uložena do souboru .fits. Spolu se snímky dochází k ukládání histogramů z jejich vlastního bufferu.

Na obdobném principu funguje i ukládání logovacích záznamů. Události ze všech vláken jsou ukládána společného zásobníku a vlákno určené pro jejich vyzvedávání je ukládá do souboru `CaFe2_Conf.log`. Ukázka kódu hlavní smyčky čekající na událost v podobě přijetí nových dat je zobrazena níže. Důležitým bodem je ochrana zápisu a čtení do bufferu pomocí kritické sekce.

```
bool end = false;
//zapisovat max. po 120 s
do
{
    if(::WaitForSingleObject(((CLog*)x)->m_hStopEvent, TIME_BTW_WRITEBUFF) ==
        WAIT_OBJECT_0)
        end = true;

    //Flush bufferu
    EnterCriticalSection(&((CLog*)x)->m_cs);
    ((CLog*)x)->WriteBuffer();
    LeaveCriticalSection(&((CLog*)x)->m_cs);

} while(!end);
```

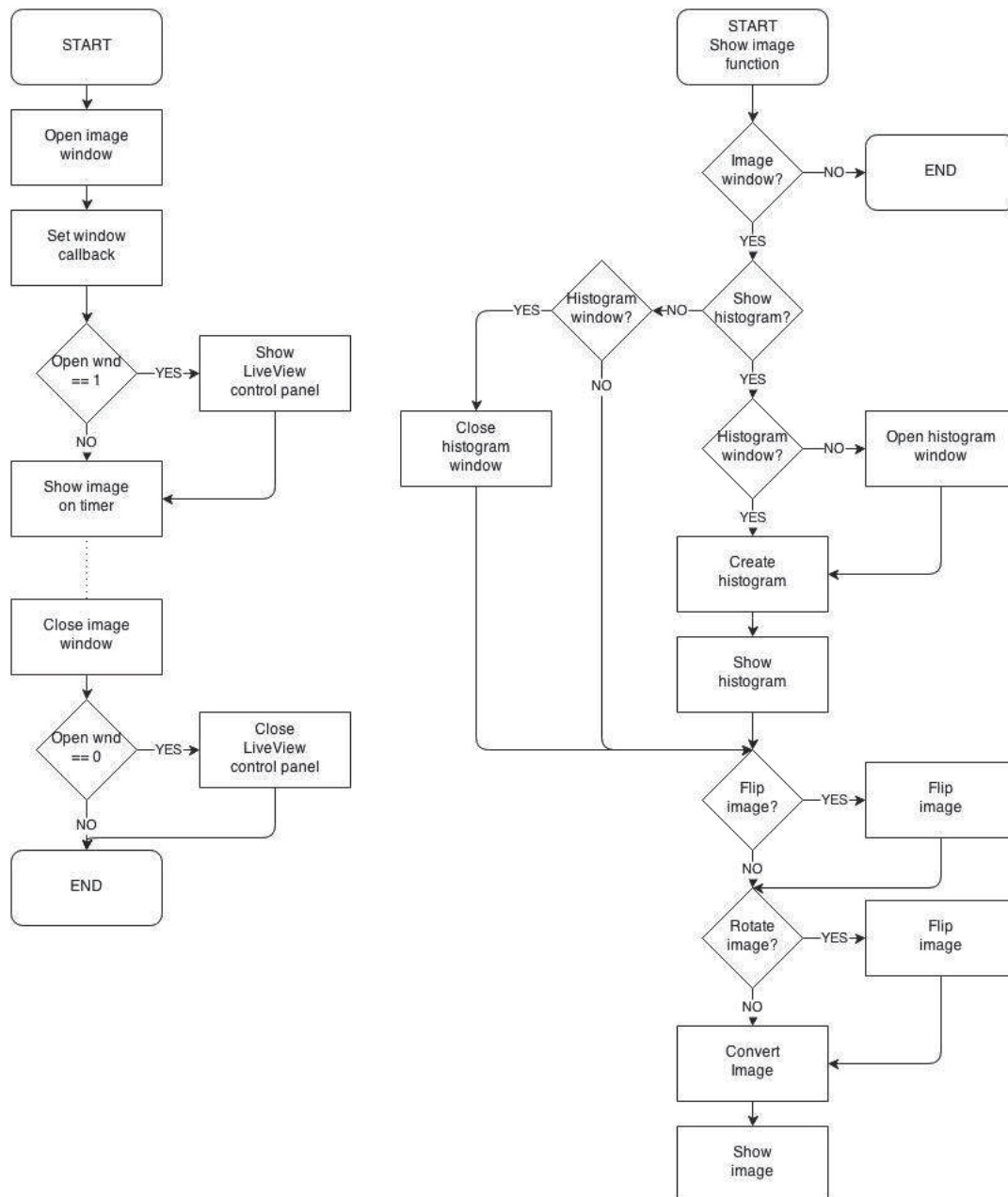
6.3.6 Zobrazení snímků

Zobrazování snímků není oproti předchozím záležitostem vykonáváno ve vlastním vlákně, ale je součástí hlavního dialogového okna, jelikož z logického hlediska dělení zapadá do grafického rozhraní s uživatelem. Ačkoli si toto vlákno obrazová data aktualizuje vždy po pořízení nového snímku, jejich vykreslování je závislé na spuštěném LiveView.

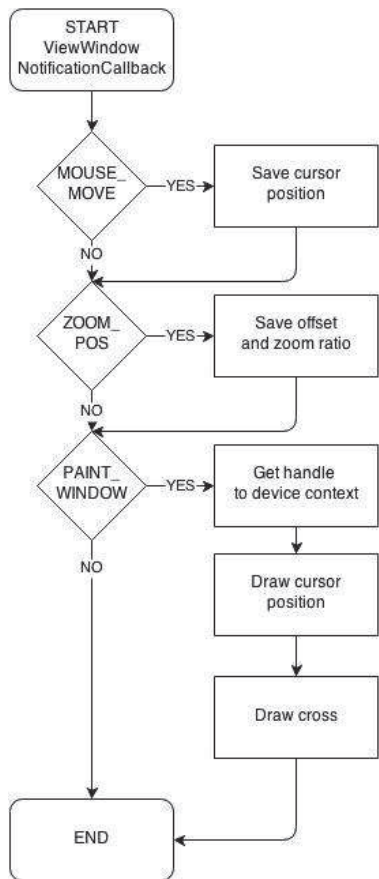
Prvním krokem před zahájením zobrazování je otevření okna, ve kterém se budou snímky zobrazovat. Jeho velikost je vlákně zaslána před zahájením samotného měření. Během měření se pak již jeho velikost nemění. Každé otevřené okno představuje jednu vytvořenou instanci třídy *cWindow* pro snímky a jednu instanci třídy *cHist* pro histogramy. Tyto objekty obalují proměnné nezbytné pro činnost okna. Všechny tyto objekty slučuje jedna instance třídy *cLiveView*, která zahrnuje i dialog pro nastavování zobrazení snímků. V případě, že je zaslán požadavek na otevření prvního okna, je zobrazen i ovládací panel. Ten zaniká spolu se zavřením posledního okna.

Po úspěšném vytvoření a otevření okna pro zobrazování již není ignorována zpráva od časovače hlavního dialogového okna na vykreslování snímků. Před jejich zobrazením dochází k úpravě v závislosti na požadavcích uživatele. Je umožněna rotace snímků, překlopení, úprava jasu a kontrastu a vypočtení a zobrazení histogramu. Funkce rotace a překlopení jsou součástí balíčku JAI SDK. Jas a kontrast je přepočítáván pomocí takzvané LUT (LookUp table), která představuje závislost výstupních hodnot obrazu na vstupních.

Před vykreslením snímku je volána návratová funkce (callback function), kde jsou do snímku kromě obrazových dat dokreslovány objekty pro zobrazení kříže a pozice kurzoru.



Obrázek 25 – Vývojový diagram pro zobrazování snímků



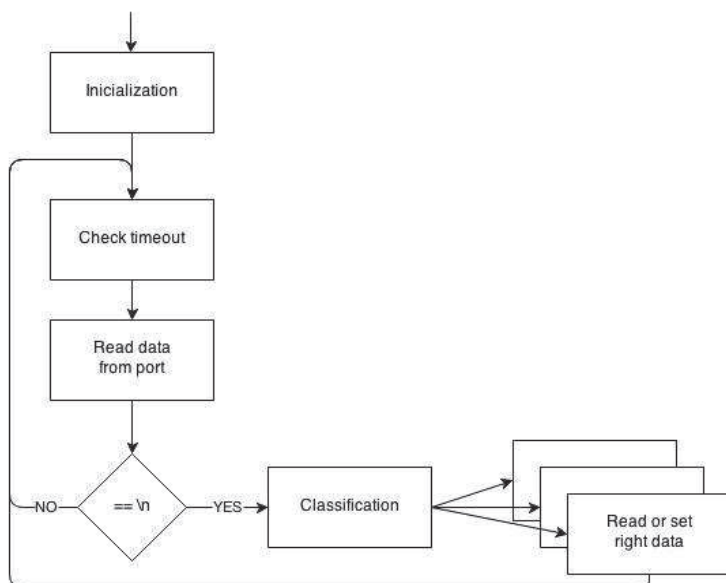
Obrázek 26 – Funkce pro zpětné volání při vykreslování obrázku

7 PROGRAM V PLC A VIZUALIZACE

Program umístěný v PLC pro řízení dalekohledu, teleskopu a kopule, byl vytvořen již v minulosti. Této diplomové práce se týká pouze jeho rozšíření o ovládání krokových motorů fokusérů, ovládání napájení jednotlivých prvků systému a o detekci přimrznutí kopule. Dalším úkolem je zpřístupnění všech funkcí systému přes komunikační protokol ASCOL. Ten již některé potřebné příkazy obsahuje, je však nutné ho rozšířit a nové příkazy implementovat.

7.1 Rozšíření kom. protokolu ASCOL

Každý port obsluhuje jedna hlavní funkce, která mimo přijímání a odesílání dat zajišťuje i inicializaci, kontrolu doby připojení a blokování přijímání do té doby dokud není odeslána odpověď.



Obrázek 27 – ASCOL - klasifikace příkazů

Přijímaná i odesílaná data jsou uchovávána každá ve vlastním datovém bloku. Jelikož funkce pracuje s ukazateli na datový blok a s ukazateli na pozici v bloku AR1 a AR2, je nutné při obsluze požadavků z ASCOLu tyto ukazatele ukládat a následně obnovovat. Při přístupu do jiného datového bloku se přepíše hodnota proměnné DBNO (Data Block Number Opened) a tím by došlo ke ztrátě přístupového bodu k přijímaným nebo odesílaným datům. Uložení a následné načtení potřebných ukazatelů se provede následujícím kódem.

```
//uchovani dat
TAR1 #ar1save
TAR2 #ar2save
L DBNO
T #dbsave
```

```
//obnovení dat
  OPN DB [#dbsave]
  LAR1 #ar1save
  LAR2 #ar2save
```

Mezi uchováním a obnovením dat je možné přistupovat k datovým blokům. Jiném místě funkce obsluhující ASCOL, by došlo ke kolizi a ztrátě dat.

Rozpoznávání a případné následné vykonávání příkazu je zahájeno při přijetí znaku LF nebo dvojicí CR LF. Nejprve je zavolána funkce ASCOL_COMMANDS, která na základě prvního znaku (v některých případech dva znaky) volá další funkce. Pro písmeni T to je funkce ASCOL_TELSCOPE, pro FO pak ASCOL_FOCUS a další. Struktura těchto je funkcí je principiálně stejná. V první fázi se nalezne konkrétní příkaz. Jestliže není nalezen, funkce vrací ERR. V dalším kroku se načtou parametry a zkontroluje se jejich rozsah, v případě že není splněn počet nebo rozsah, vrací ERR. Následně je při dotazu načtena požadovaná hodnota z datového bloku (musí být ošetřeno uložení a obnovením ukazatelů) a uložena do odesílacího bloku. V případě aktivního požadavku je nastaven daný merker, který pak dále v programu funguje jako nastavené tlačítko. Níže je uveden příklad v podobě příkazu USRS (čtení uživatelských rychlostí).

```
usrs: CALL "ASCOL_CHECK_END"           //kontrola konce příkazu
      OK:=#okend
      A #okend
      JCN err

//uchování ukazatelů na otevřené DB a na pozici v něm
  TAR1 #ar1save
  TAR2 #ar2save
  L DBNO
  T #dbsave

//načtení dat z parametrů a následné přepočítání na inženýrské jednotky
//dočasné proměnné jsou používány z toho důvodu, že následující funkce ASCOL_FORMAT_I a
//ASCOL_STORE_SPACE již potřebující pracovat se správnými ukazatel na DB
  L "DB_PARAM".RA_USER_SPEED
  L 6 //EU unit
  *D
  T #dtemp1

  L "DB_PARAM".DEC_USER_SPEED
  L 6 //EÜ unit
  *D
  T #dtemp2

//obnovení dat
  OPN DB [#dbsave]
  LAR1 #ar1save
  LAR2 #ar2save

//uložení požadovaných dat odesílacího bloku
  CALL "ASCOL_FORMAT_I"
  DECIMALS:=3
  DIGITS :=1
  val :=#dtemp1

  CALL "ASCOL_STORE_SPACE"
```

```
CALL "ASCOL_FORMAT_I"
DECIMALS:=3
DIGITS :=1
val :=#dtemp2
```

```
CALL "ASCOL_STORE_END"
BEU
```

7.2 Ovládání krokových motorů fokusérů

Ovládání karty 1STEP-5V se provádí prostřednictvím čtyř výstupních bytů. Naopak informace o stavu karty se získávají ze čtyř vstupních bytů. Význam nejdůležitějších bitů a bytů je uveden v *Tabulka 4* a v *Tabulka 5*. Kompletní tabulka je uvedena v literatuře

Tabulka 4 – 1STEP-5V - výstupní byty

byte 0 - 3	byte 0	Multiplier G: $F_a = F_b \times R \times G$
	byte 1 - 3	Distance or position
byte 4	Bit 6	Hold traversing job (STOP)
	Bit 5	Backward start (DIR_M)
	Bit 4	Forward start (DIR_P)
	Bit 2 – 0	Mode (for our case Mode = 0 – Relative incremental move)
byte 5	Bit 2	Pulse enable (DRV_EN)

Tabulka 5 – 1STEP-5V – vstupní byty

byte 0 - 3	byte 0	Multiplier G: $F_a = F_b \times R \times G$
	byte 1 - 3	Distance or position
byte 4	Bit 6	Hold traversing job (STOP)
	Bit 5	Backward start (DIR_M)
	Bit 4	Forward start (DIR_P)
	Bit 2 – 0	Mode (for our case Mode = 0 – Relative incremental move)
byte 5	Bit 2	Pulse enable (DRV_EN)

Potřebné nastavení karty lze provést dvěma způsoby. Složitějším je nastavení v kódu pomocí výše zmíněných bitů. Druhou variantou je zadání parametrů přímo v hardwarové konfiguraci ve vlastnostech karty. Dialogové okno pro nastavení je zobrazeno na *Obrázek 28*. Lze zde nastavit hodnotu základní frekvence f_b , násobič n pro stanovení pracovní frekvence a čas i definující dobu náběhu a doběhu na pracovní frekvenci. Dále je možné nastavit vlastnosti dvou digitálních vstupů. Typ a jejich funkci.

Parameter	Value
Parameters	
Group diagnostics	<input type="checkbox"/>
Base frequency	80 Hz
Multiplier n: $Fp-p = Fb * n$	45
Time i: $a = Fb / (i * 0.128 \text{ ms})$	5
Feedback value	Absolute position
Function DI0	External STOP
Function DI1	Reference switch
Input DI0	Normally open contact
Input DI1	Normally open contact
Limit switch	Normally open contact
Modulo axis	<input type="checkbox"/>
Traversing range	16777216

Obrázek 28 – Nastavení karty ISTEP-5V

7.3 Rozšíření programu v PLC o nové funkce

Hardwarová konfigurace automatu byla rozšířena o dva digitální výstupní, dva analogové vstupní moduly a dvě karty pro ovládání krokových motorů. Digitální karty slouží pro zapínání a vypínání napájení podsystému. Funkcionalita tohoto řízení není složitá a proto si ji popíšeme níže.

```
A "DB_TL".CCDI_ON_TL
O "CCDI_ON_ACL"
R "DB_TL".CCDI_ON_TL
R "CCDI_ON_ACL"
S "DB_TL".CCDI_ON_SEL
```

```
A "DB_DIO".CCDI_SEL
= Q 24.0
```

Tato část kódu je umístěna ve funkci pro tlačítka. Bity nastavené událostí, stisk tlačítka ve vizualizaci _TL nebo zaslání požadavku přes ASCOL protokol _ACL, nastavují příslušný bit _SEL a poté jsou shozeny na log. „0“. Bit _SEL je pak již přímo spojen s výstupem karty. Zápis do karty se provádí až na konci celého cyklu automatu.

Pro načítání analogových vstupů z karet byla použita již vytvořená funkce NACTI_AI. Její funkcionalitu zde nebudu z důvodů firemního know-how popisovat. Parametry funkce představují adresu konkrétního vstupu na kartě a jeho rozsah. Při podtečení nebo přetečení je nastaven chybový bit _ER. Hodnota výstupu je zapsána do analogové proměnné _AI. Standardní proudový rozsah, ve kterém karta pracuje, je 4 – 20 mA. Lze však také nastavit rozsah 0 – 20 mA.

```
CALL "NACTI_AI"
INPUT :=PIW540
MAX :=27648
MIN :=0
MOD_0_20:=FALSE
ANALOG :="DB_AIO".MI_AI
ALARM :="DB_AIO".MI_ER
```

```

L "DB_AIO".MI_AI
ITD
DTR
L 1.808449e-001 //0 - 5000mA
*R
RND
T "DB_AIO".MI_AId

```

Po načtení hodnoty karty proveden převod na inženýrské jednotky, v našem případě na ampéry s rozsahem 0 – 5A.

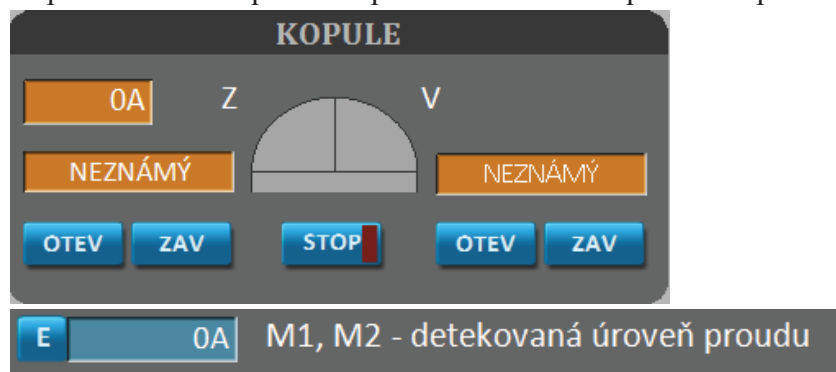
7.4 Úprava vizualizace

Vizualizace k řízení teleskopu, kopule a ostatních podsystému již byla hotova. Úkolem této práce bylo pouze ji rozšířit o ovládání napájení jednotlivých komponent a o detekci přimrznutí kopule. Rozhraní pro ovládání napájení je zobrazeno na obrázku *Obrázek 29*.



Obrázek 29 – Vizualizace – ovládání napájení jednotlivých komponent

Přimrznutí kopule rozpoznávané pomocí měření úrovně proudu na motorech otevírajících levou a pravou část, je signalizováno zobrazením oranžového rámečku s aktuální hodnotou proudu v ampérech vedle příslušné polokopule. Rozhodující úroveň proudu je třeba nastavit na základě testování v provozu. Toto testování bylo prováděno až po odevzdání diplomové práce. Nastavení se provádí v parametrické obrazovce.

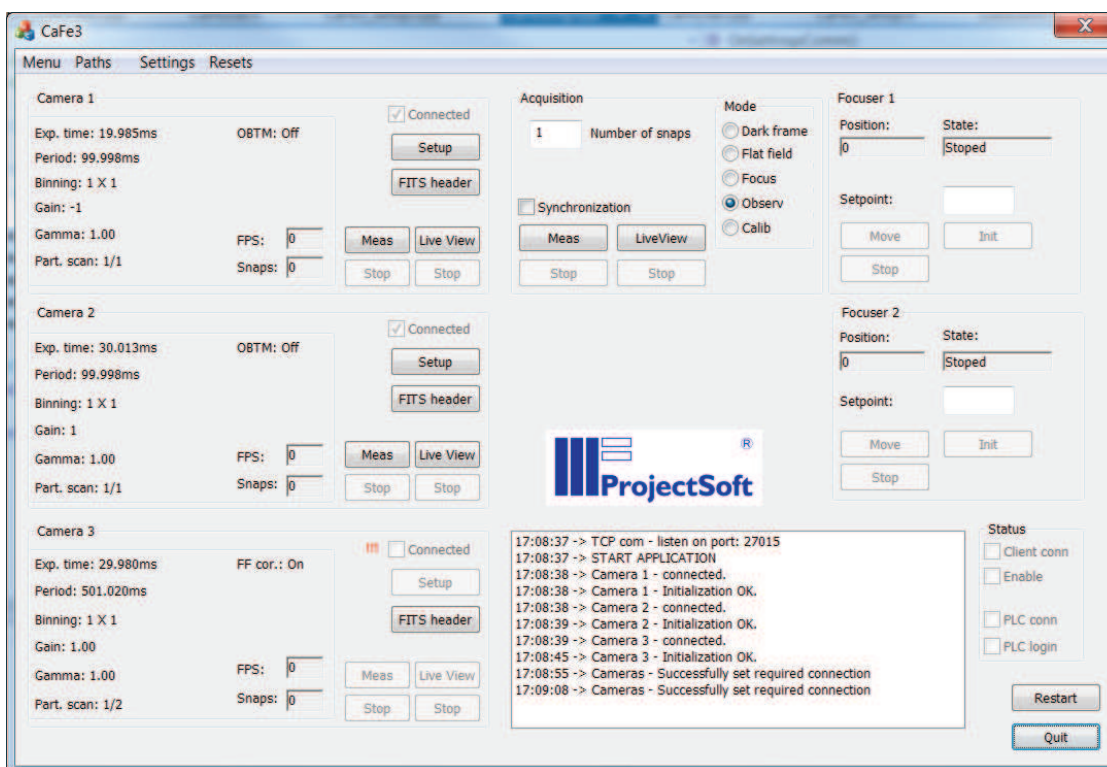


Obrázek 30 – Vizualizace – detekce přimrznutí kopule

8 DOKUMENTOVÁNÍ PROJEKTU A FUNKČNÍ ZKOUŠKY

V této kapitole si přiblížíme jednu z posledních fází při práci na zakázce, a tou je předání zakázky spolu s kompletní uživatelskou dokumentací a zaškolením obsluhy softwaru. V manuálu by měly být popsány všechny funkce programu, postupy jak jednotlivé funkce používat a názorné ukázky. Stav programu při předání by měl splňovat všechny body uvedené ve smlouvě a v jejím dodatku, který byl sepisován během vytváření práce.

8.1 Manuál a dokumentace



Obrázek 31 – GUI – Ovládací program CaFe3

Z hlavního dialogového okna, které se zobrazí po startu aplikace, lze přistupovat ke všem funkcím programu. Po levé straně je přehled jednotlivých kamer. Jsou zde zobrazeny parametry těchto kamer, které jsou kontinuálně kontrolovány s hlavní datovou strukturou kamer. V případě změny těchto hodnot se parametry přepíší a provedou se i potřebné změny v nastavení přístupu. Například při spuštěném snímání již nelze snímání spustit znovu.

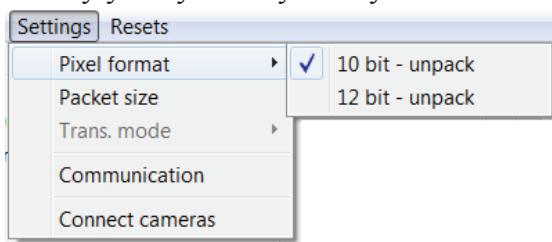
Ve střední části je umístěno společné ovládání kamer. Hromadné spuštění snímání, nastavení počtu snímků v měřící sekvenci (tento parametr je pro všechny kamery společný), zapnutí a vypnutí synchronizace prvních dvou kamer a nastavení režimu snímání. Režim snímání nelze nastavit pro jednotlivé kamery zvlášť. Dále je zde

vyobrazeno logo firmy ProjectSoft a.s. a logovací okno. V logovacím okně se zobrazují události jak chybové, tak stavové, přičemž se události i zároveň zapisují do textového souboru CaFe3_log.log. V případě, že uživatel požaduje podrobnější výpis nebo naopak řidší, je možné nastavit prioritu, s jakou se budou události vypisovat.

V pravé části je umístěno ovládání fokusérů a stav TCP/IP komunikace.

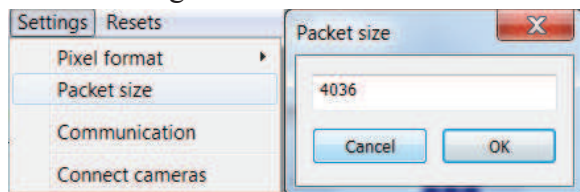
8.1.1 Menu

Menu programu umožňuje nastavení, která se vztahují ke všem kamerám dohromady, anebo se týkají samotného programu. První položka v záložce *Settings* slouží k volbě formátu pixelu. Na výběr je 10-bitový nebo 12-bitový formát. V obou případech se jedná o nezabalený tvar, kdy je počet bitů reprezentujících hodnotu pixelu doplněn na celé byty nevýznamnými bity.



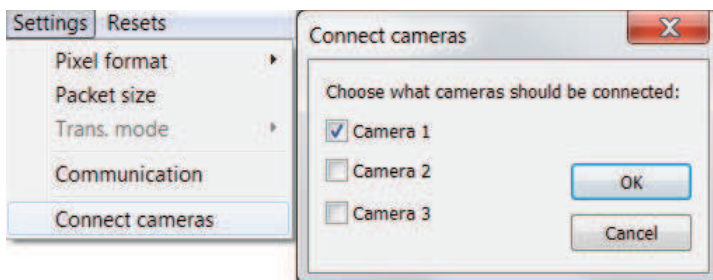
Obrázek 32 – CaFe3 – Nastavení formátu pixelu

Program umožňuje spravovat síťový provoz datového toku z kamer a to formou nastavení velikosti zasílaných paketů. Zadávací dialog definuje rozsah velikosti na 1428 až 16 020 s dílčím krokem 8. Horní hranice tohoto rozsahu však nemusí být vždy dosažitelná, z důvodu omezených vlastností síťové karty. Na té je také potřeba povolit funkci Jumbo frame, která umožňuje přenos větších paketů. V případě, že velikost paketu nastavená na kameře přesahuje maximální možnou hodnotu na kartě, nebude komunikace fungovat.



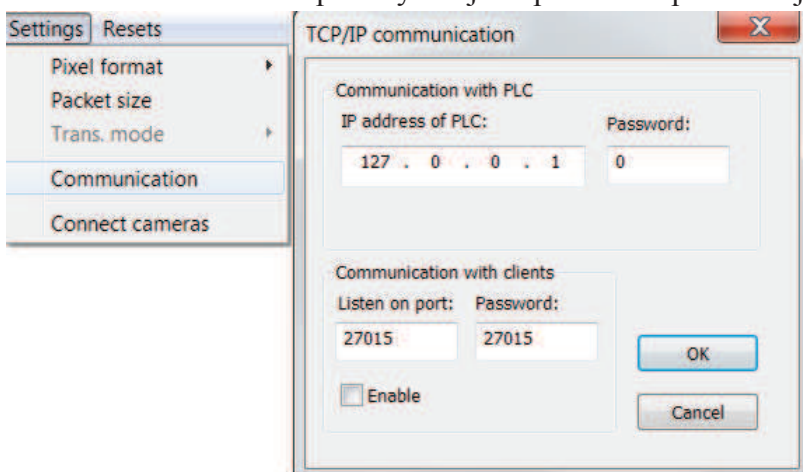
Obrázek 33 – CaFe3 – Nastavení velikosti paketů

Jelikož se program neustále snaží připojit kamery, může tato činnost v případě nepřítomnosti některé kamery, způsobit zpomalení nebo nesprávné fungování programu. Z toho důvodu je umožněna volba kamer, které chceme připojovat. A silně doporučeno toto nastavení používat.

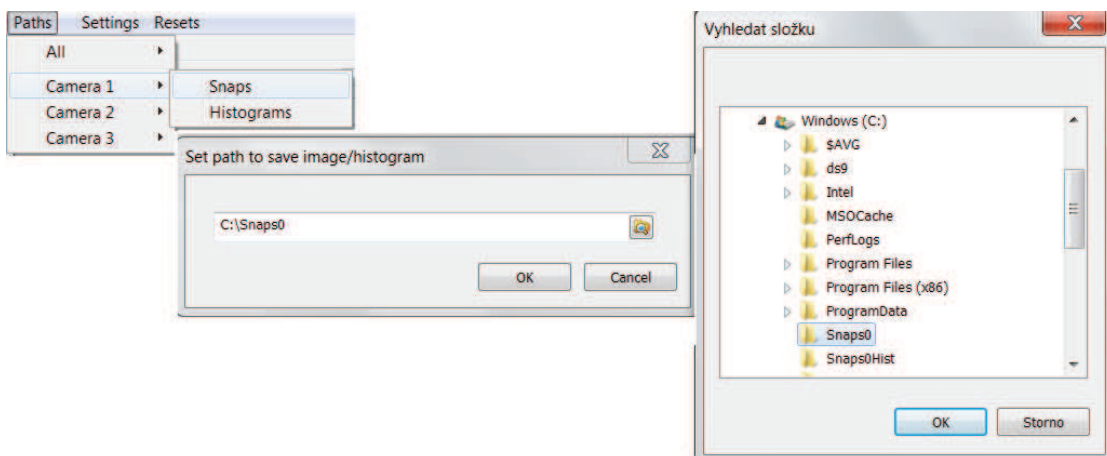


Obrázek 34 – CaFe3 – Nastavení připojení kamer

Položka menu *Communication* otevírá dialog pro nastavení klientského i serverového připojení. Program CaFe3 se jakožto klient připojuje k PLC z důvodu ovládání fokusérů. Lze tedy zde nastavit IP adresa automatu a heslo pro přihlášení. Volný port se hledá automaticky s rozsahu definovaného pro ASCOL a to 2000 až 20004. Program je zároveň serverem pro řízení kamer přes komunikační protokol SOCOL. Je možné port, na kterém bude program naslouchat, heslo pro přihlášení a také povolení této komunikace. Jestliže není povolena, může klient pouze číst hodnoty, ale nemůže zadávat aktivní příkazy. Stejnou podmínku představuje i přihlášení.



Obrázek 35 – CaFe3 – Nastavení TCP/IP komunikace

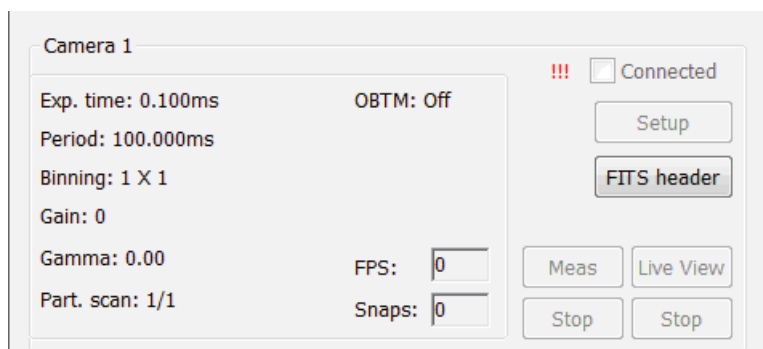


Obrázek 36 – CaFe3 – Nastavení cesty pro ukládání snímků

Pro každou kameru, i pro všechny naráz, lze nastavit adresář, do kterého se budou ukládat snímky a adresář pro histrogramy. V případě, že uživatel zvolí název složky, která neexistuje (cesta musí být správná), je složka vytvořena.

Poslední funkcí menu je restart kamer. V případě, že má uživatel pocit, že s kamerou není něco v pořádku, prvním krokem k vyřešení může právě restart. Restart odpovídá vypnutí a zapnutí napájení, přičemž při opětovném připojení kamery, se do ní znovu nahrají data načtená z databáze.

8.1.2 Ovládání kamer



Obrázek 37 – CaFe3 – Náhled kamery

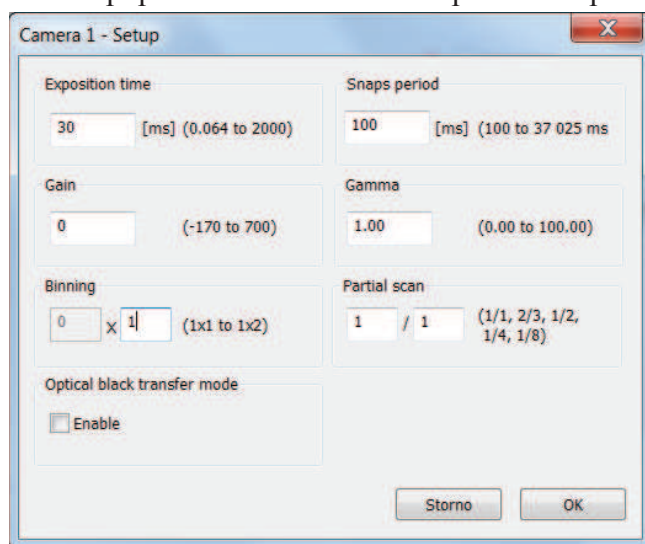
Za předpokladu, že je zvoleno, že chceme danou kameru připojit (nastaveno v menu *Connect Cameras*), probíhá připojení automaticky a program se snaží o navázání spojení do té doby, dokud se mu to nepodaří. V případě, že kameru chceme připojit, ale není připojena, jedná se o nepožadovaný stav. Tento stav zapříčiňuje pomalý chod programu a přerušované spojení s ostatními již připojenými kamerami. Tento stav je symbolizován červenými vykřičníky vedle ukazatele stavu připojení. V tomto případě, je doporučeno nastavit pro kameru, že ji nechceme připojit, anebo ji uvést do takového stavu, aby bylo připojení k ní možné. V okamžiku kdy je kamera v pořádku připojena, načtou se nejprve data z databáze a provede se inicializace kamery. V případě, že není informace v souboru přítomná, nastaví se daná hodnota podle výchozího nastavení programu. Po úspěšném připojení, nastavení a následném vyčtení všech parametrů kamery se zaškrtně *Connected* a jsou zpřístupněna tlačítka *Setup*, *Meas* a *Live View*. Tlačítko *FITS header* je zpřístupněno, i když není kamera připojena.

Informace o nastavení kamery, které jsou zobrazeny v levé části, nesouvisí přímo s tím, co uživatel pošle jako požadavek na nastavení přes dialog *Setup*, ale jsou vyčítány přímo z kamery. Z toho vyplývá, že v případě, že se parametry nepodaří nastavit, nebo že bude kamera odpojena, přenastavena a znovu připojena, budou hodnoty správné a aktuální. Při připojení je do kamery nahráno nastavení načtené z databázového souboru *CaFe3_Conf.db*. Další informací je počet snímků naměřených danou kamerou (společná hodnota pro *LiveView* i pro samotný měřicí režim) a aktuální počet snímků za vteřinu. Při vypnutém snímání je tato hodnota nula.

Sběr dat z kamery je možný dvěma způsoby. Prvním je vlastní měření (*Meas*), kdy je dán počet snímků, který chceme snímat, a všechna přijatá data se spolu s vypočteným histrogramem ukládají na disk do nastaveného adresáře. Pro tyto data je generována

FITS hlavička s potřebnými informacemi. Druhou možností je LiveView, kdy snímání probíhá nezávisle na počtu nastavených snímků a skončí až po zmáčknutí tlačítka *Stop*. Při tomto měření nedochází k ukládání dat, ale obrázky jsou spolu s histogramem zobrazovány ve vlastním okně. LiveView bude podrobněji popsáno v další podkapitole. Obě jdou nezávisle na sobě zapínat a vypínat. Když běží LiveView můžeme spustit měření a uložit potřebný počet snímků, bez nutnosti přerušit LiveView.

Dále si popíšeme nastavení těchto parametrů přes dialog *Setup*.



Obrázek 38 – *CaFe3* – Nastavení kamery

Nastavování doby expozice je velmi úzce spjato s parametrem periody snímků. Prvním logickým omezením je, že doba periody nemůže být kratší než doba expozice. Zadávání hodnot je automaticky kontrolováno s tou podmínkou, že perioda musí být delší o 5ms, aby se stihlo jak vyčítání dat z kamer, tak jejich zpracování a uložení do bufferu. Dále je hlídána minimální doba periody, tak aby počet snímků za vteřinu nepřesáhl stanovené meze. Pro kamery BM-141GE to je deset snímků, tedy minimální hodnota 100ms, a pro kameru SP-5000-GE pět snímků za vteřinu a 200ms.

Jelikož jsou doby periody a expozice řízeny pulzními čítači s předděličem, záleží na nastavení tohoto předděliče maximální čas, který je čítač schopen měřit a také jeho přesnost. Z toho důvodu, je expozice rozdělena do tří intervalů. V prvním intervalu je možné nastavovat čas s přesností 1 μ s a maximální doba periody je 524 ms. V druhém intervalu je přesnost 10 μ s a maximální perioda 5 242 ms, ve třetím pak 0,1 ms přesnost a perioda 37 025 ms. Aktuální rozsah periody v závislosti na době expozice se zobrazuje v dialogovém okně vedle zadávacího pole.

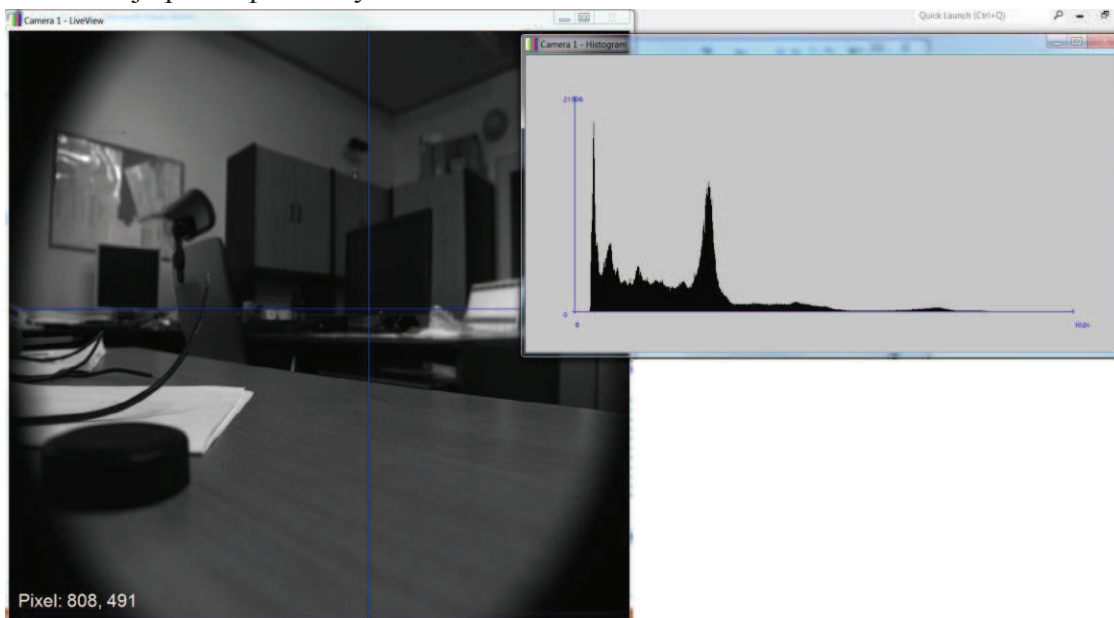
Dále nastavovací dialog umožňuje zadávat zesílení v rozsahu -170 až 700 pro kamery BM-141GE a 0.00 – 16.00 pro kameru SP-5000-GE. Kamery umožňují i nastavení gamma korekce výstupního obrazu ve standardním rozsahu 0.00 až 100.00, kde hodnota 1.00 představuje originální snímek. Posledním parametrem nastavitelným při spuštěném snímání je *Optical Black Transfer Mode*, funkce kdy je obrazovým datům přidáno několik sloupců černé, pro referenční účely. Tuto vlastnost mají pouze kamery BM-141GE. Kamera SP-5000-GE je oproti nim schopná vlastní flat-filed korekce. Tedy

korekce určená k odstranění artefaktů v obraze, způsobených rozdílnou citlivostí dílčích částí snímače a zkrácením informace optickou cestou.

Parametry, které nejde měnit během spuštěného snímání, ovlivňují velikost výstupních obrazových dat. Jedná se o binning a částečné skenování. V případě binningu, jsou sousední pixely slučovány a tím je dosaženo větší citlivosti na světlo, za cenu snížení rozlišení. Kamery BM-141GE umožňují pouze vertikální binning, tedy maximální sloučení 1x2. U kamery SP-5000-GE lze nastavit sloučování až 2x2, což čtyřikrát zmenší množství dat. Dalším způsobem, jak snížit datový tok, je částečné skenování. Tato varianta nesnižuje rozlišení, ale umožňuje nám vybrat si pouze tu část obrazu, která nás zajímá, a tu vyčítat ze CCD senzoru. Omezením je že může být v jednu chvíli zapnuta pouze jedna varianta.

8.1.3 LiveView

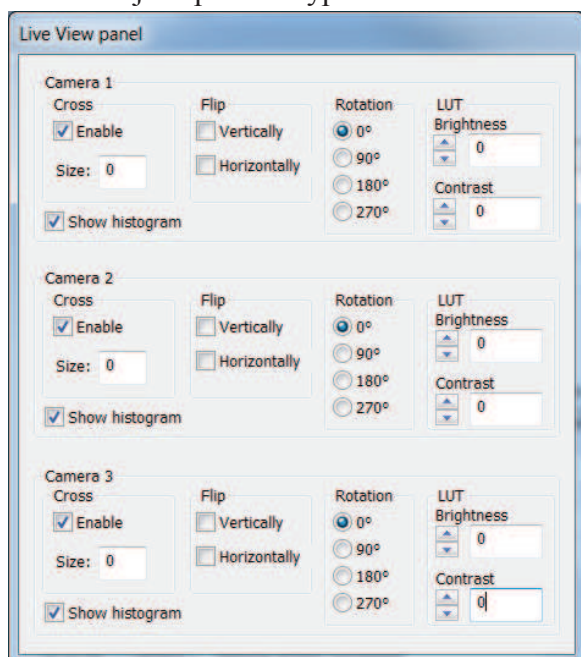
LiveView představuje pro uživatele jednu z nejdůležitějších věcí. Díky němu má vizuální kontakt s pozorovaným objektem a na jeho základě může měnit nastavení kamery nebo samotného zobrazování. Na *Obrázek 39* je vidět okno pro zobrazování snímků, které je doplněno o nastavitelný kříž, který sleduje pohyb kurzoru. Jeho pozice v souřadnicovém systému obrázku je zobrazena v levé dolní části. Spolu s obrazovými daty je vykreslován i histogram. Hodnota histogramu je vypočítávána z dat na výstupu z kamery. Tedy z dat, které jsou ovlivněny zesílením, gamma korekcí nebo flat-field korekcí. Naopak nastavení kontrastu a jasu, nemá na histogram žádný vliv. Toto nastavení je pouze pro účely vizualizace.



Obrázek 39 – CaFe3 – LiveView

Při spuštění LiveView jedné nebo více kamer se zobrazí také ovládací dialog, který umožňuje manipulovat s obrazem. Kromě rotace s krokem devadesáti stupňů a

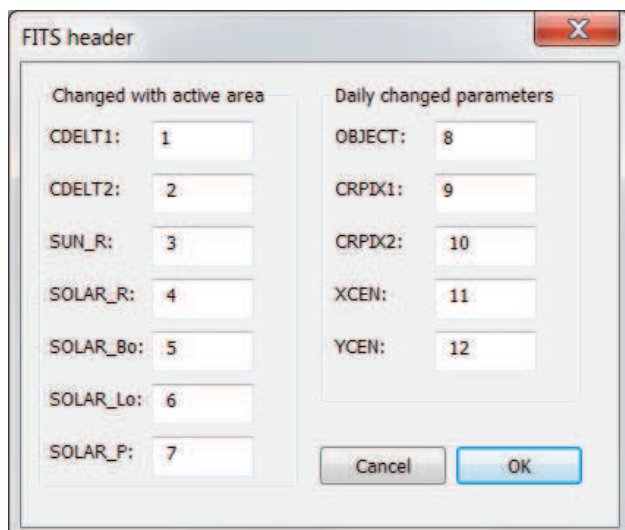
horizontálního nebo vertikálního překlopení, umožňuje také měnit kontrast a jas snímků. Toto nastavení je pouze pro účely zobrazení. Snímky, které se ukládají, jsou v takovém formátu, v jakém se vyčítají z kamer. Další funkcí je zobrazení kříže a nastavení jeho velikosti, kdy při hodnotě nula je průsečík na pozici kurzoru. Se zvyšováním hodnoty dochází k rozdvojení čar a k jejich oddalování od kurzoru. Dialog také umožňuje zapnutí a vypnutí zobrazování histogramu.



Obrázek 40 – CaFe3 – Ovládání LiveView

8.1.4 Nastavení FITS hlavičky

Pro každý ukládaný snímek je generována FITS hlavička obsahující informace o dané kameře, snímky i pozorovaném objektu. Část parametrů je statických, část je nastavována automaticky programem a část je nutná zadat uživatelem. Uživatelskou část je možné měnit dvěma způsoby. Zaprvé přes uživatelské rozhraní, kdy se po kliknutí na tlačítko *FITS header* u dané kamery otevře dialog zobrazený na Obrázek 41. Po potvrzení tlačítkem *OK* jsou parametry uloženy jak do datové struktury programu, pomocí které je pak generována samotná hlavička, tak do databázového souboru *CaFe3_Conf.db* do tabulky *CamX_FITS*, kde X představuje číslo kamery. Z databáze se informace nahrávají pouze při startu programu.



Obrázek 41 – CaFe3 – FITS hlavička

Níže je uveden kompletní přehled FITS hlavičky, tak jak ukládána spolu s vytvářeným snímkem. Zeleně jsou označeny údaje, které doplňuje automaticky program, černě statická, modře a červeně doplňuje uživatel. Přičemž červeně označovaná data budou vyplňována denně a modře vždy při nájezdu na aktivní oblast.

```

SIMPLE =          T / file does conform to FITS standard
BITPIX =          16 / number of bits per data pixel
NAXIS =           2 / number of data axes
NAXIS1 =          1392 / length of data axis 1
NAXIS2 =          1040 / length of data axis 2
DATE =            ''/
TIME_OBS=         ''/
DATE_OBS=         ''/ UTC date and time of data aquisition
FILENAME=         '15_14_07_SnapNum0.fits'/nazev souboru
ORIGIN =          'Ground'/
OBJECT =          '8'/nazev pozorovane oblasti
EXPTIME =         30.0000000000 /exposure duration (milliseconds)
WAVELNTH=        "/Ha,CaH,WL
IMAGETYP=        'obs'/ image type TEST/DARK/FLAT/NORMAL
TELESCOP=        'SORT'/ telescope used to acquire data
INSTRUME=        "/CCD1,CCD2,CCD3
BZERO =          32768.0000000 /linearni skalovani dat
BSCALE =          1.00000000000 /linearni skalovani dat
CTYPE1 =         'Solar-x'/typ souradnic
CTYPE2 =         'Solar-y'/typ souradnic
CRVAL1 =         0.00000000000 /Referencni hodnota sour. systemu
CRVAL2 =         0.00000000000 /Referencni hodnota sour. systemu
CRPIX1 =         '9'/Referencni pixel

```

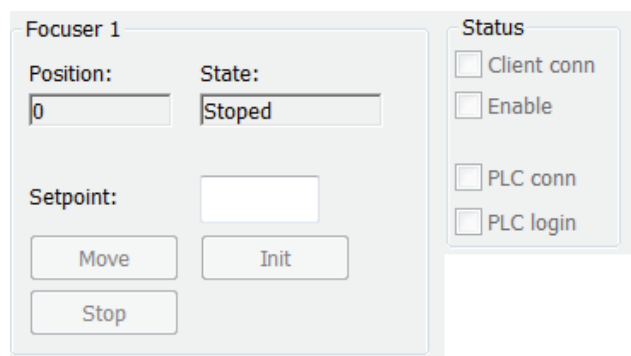
```

CRPIX2 =      ' 10'/Referencni pixel
XCEN  =      ' 11'/Pozice stredu zorneho pole FOC v arcsec
YCEN  =      ' 12'/Pozice stredu zorneho pole FOC v arcsec
CDELTA1 =    ' 1 '/Velikost pixelu v obloukovych sekundach
CDELTA2 =    ' 2 '/Velikost pixelu v obloukovych sekundach
SUN_R  =      ' 3'/Polomer Slunce v pixelech
SOLAR_R =    ' 4'/Polomer Slunce v arcsec
SOLAR_Bo=    ' 5'/Heliograficka souradnice - efemerida
SOLAR_Lo=    ' 6'/Heliograficka souradnice - efemerida
SOLAR_P =    ' 7'/Pozicni uhel - efemerida
COMMENT Cameras controls CaFe3 by ProjectSoft HK,Czech Rep.
END

```

8.1.5 Ovládání fokusérů

Ovládání krokových motorů fokusérů se liší oproti ovládání kamer. Hlavní kód pro jejich obsluhu je totiž napsán v automatu a program CaFe3 akorát pomocí komunikačního protokolu ASCOL čte jejich stav a zasílá automatu potřebné příkazy na jeho řízení. Předpokladem pro možnost ovládání je tedy funkční spojení s PLC. Stav tohoto spojení je zobrazen na status panelu. Samotné připojení však nestačí. Aby bylo možné zadávat aktivní příkazy, nájezd, stop a inicializace, je nutné se přihlásit.



Obrázek 42 – CaFe3 – Ovládání fokusérů

Při startu automatu není známa aktuální poloha krokového motoru a ten se tudíž může nacházet v libovolné pozici. Z tohoto důvodu je nutná nejprve inicializace. Ta spočívá v najetí krokového motoru na nultou pozici označenou koncovým spínačem. V této pozici se inicializace zastaví a poloha je nastavena na hodnotu nula. Od tohoto momentu je možné správně nastavovat pozici v rozmezí 0 až 7000 kroků. Pohyb je možný kdykoli zastavit tlačítkem *Stop*.

8.2 Zkoušení a ladění na procesu

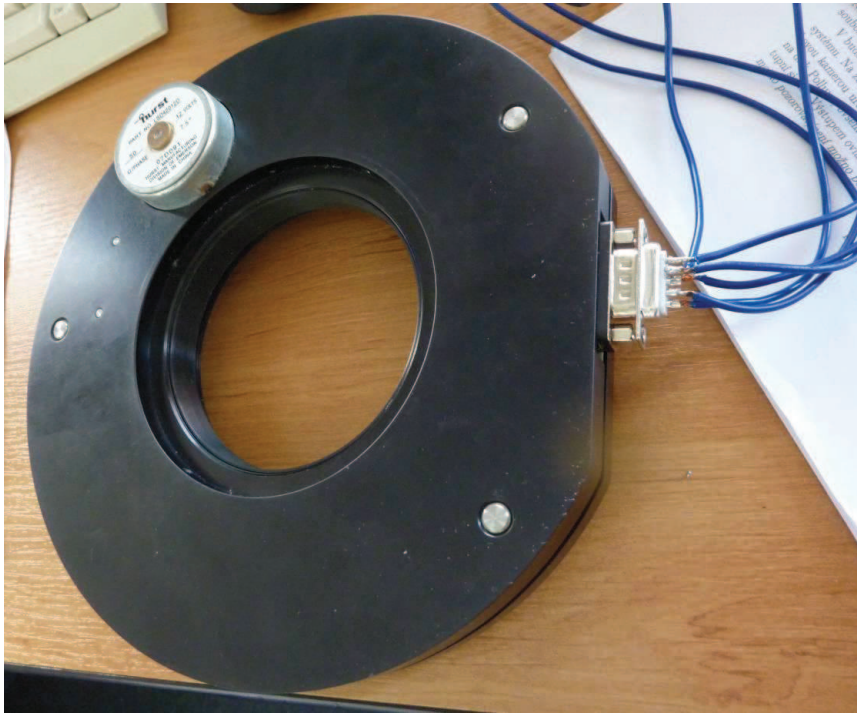
Po vytvoření projektu přichází na řadu zákaznický test, uvedení do provozu a zaškolení obsluhy. V našem případě splývají tyto tři činnosti do jedné, protože u testování na provozu bude přítomný zákazník, který bude zároveň zkoušet funkčnost aplikací a také splnění jednotlivých bodů zadání. Termín pro úspěšné předání zakázky je stanoven 31.5. a do té doby by měl být ukončen vývoj a zákazník bude předán do péče servisu a podpory. Nevýhodou tohoto termínu je, že předání a testování proběhne až po dokončení psaní této práce. Z toho důvodu se v této kapitole zmíním pouze o alfa testování.

Testování správné funkčnosti aplikace CaFe3 probíhalo na kamerách, zapůjčených přímo od zákazníka. Jelikož konkrétně tyto kamery budou umístěny na pozorovací věži, neměly by při uvedení do provozu vznikat komplikace vlivem rozdílného hardwaru. Při testování byl použit i upravený napájecí kabel pro vzájemnou synchronizaci kamer. Připojené kamery jsou zobrazeny na *Obrázek 43*.



Obrázek 43 – Snímací kamery

Spolu s kamerami byly zapůjčeny i krokové motory se zařízením pro ovládání fokusérů. Jak je zřejmé z *Obrázek 44* motor a koncový spínač je připojen přes 9pinový konektor. Čtyři piny slouží pro ovládání dvou vinutí motoru a na dva piny je vyveden koncový snímač pro určení nulté polohy a pro zastavení pohybu v dané mezi. Krokový motor pohání přes ozubená kola tři závitové osy, na kterých celý horní kryt zařízení. Na něm je pak přichycen objektiv kamery.



Obrázek 44 – Fokusér s krokovým motorem

9 ZÁVĚR

Prvním krokem v řešení projektu a semestrální práce bylo jednání se zákazníkem, kterého jsem se účastnil. Výsledkem bylo sestavení zadání projektu a specifikování požadovaných funkcí řídicího systému pro robotický solární dalekohled. Na základě těchto informací jsem vytvořil koncepci systému.

V rámci práce jsem také provedl teoretický rozbor potřebný k uvedení a pochopení dané problematiky. Zaměřil jsem se na použité kamery BM – 141GE a SP-5000-GE, určené pro snímání slunce, problematiku jejich synchronizace a vyčítání dat. Vysvětlil jsem pojmy spojené s parametrizováním těchto kamer. Dále jsem se zabýval programovatelným automatem S7- S300 a způsobu řízení krokových motorů pomocí jeho modulů 1STEP-5V.

Prvním praktickým bodem práce byla implementace navržených komunikačních protokolů. Protokol SOCOL do ovládacího programu CaFe3 pro ovládání kamer a dodatek k protokolu ASCOL do řídicího PLC pro zpřístupnění již vytvořených funkcí systému.

Dalším krokem pro požadovanou robotizaci systému bylo vytvoření ovládacího programu kamer a fokusérů. Z důvodu požadavku na vysoký datový tok a výpočetní náročnost jsem zvolil pro vývoj knihovnu MFC, která splňuje tyto požadavky a zároveň představuje relativně snadný způsob programování. Vytvořený program umožňuje parametrizování kamer, snímání, vyčítání a základní zpracování obrazových dat. Data jsou ukládána na disk ve FITS formátu s hlavičkou obsahující důležité informace. Dále pak slouží pro ovládání krokových motorů prostřednictvím kom. protokolu ASCOL a automatu. Aplikace zajišťuje TCP/IP komunikaci v roli serveru, pro klienty připojené za účelem ovládání kamer, a v roli klienta pro výměnu, dat týkajících se fokusérů, s automatem.

Předposledním bodem zadání byla modifikace programu v programovatelném automatu pro bezprostřední řízení dalekohledu a souvisejících podsystémů. Jednalo se o rozšíření hardwarové konfigurace o dvě výstupní digitální, dvě vstupní analogové karty a dva moduly pro řízení krokového motoru. Digitální výstupy slouží pro ovládání napájení jednotlivých podsystémů, analogové vstupy měří úroveň proudu v motorech ovládajících otevírání a zavírání kopule. Překročení definované hladiny je signalizováno poruchovou hláškou a motory jsou zastaveny. Dále bylo v systému vytvořeno centrální stop tlačítko, pro ukončení pozorování, zaparkování dalekohledu a zavření kopule. Všechny tyto nové funkce jsou zobrazeny ve vizualizaci. K automatu byly také připojeny dvě karty 1STEP-5V, které spolu s vytvořeným kódem slouží pro ovládání krokových motorů fokusérů. Uživatelské ovládání těchto motorů je vyvedeno do ovládacího programu pro kamery přes nově implementovaný ASCOL protokol.

Na závěr jsem vytvořil dokumentaci pro ovládací program a pro změny v automatu a vizualizaci. Testování funkčnosti systému bylo prováděno pouze v laboratorních podmínkách. Uvedení do provozu u zákazníka se uskuteční až po odevzdání této práce.

CITOVANÁ LITERATURA

- [1] **Microsoft.** Using Messages and Message Queues. *http://msdn.microsoft.com/*. [Online] 2014. [http://msdn.microsoft.com/en-us/library/windows/desktop/ms644928\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ms644928(v=vs.85).aspx).
- [2] **JAI.** User manual. *BM- 141GE, BB - 141GE*. 2011. 2.1.
- [3] **JAI.** User manual. *SP-5000M-GE2, SP-5000C-GE2*.
- [4] **ProjectSoft.** *TomPack: Uživatelská příručka*. 2006.
- [5] **IBM.** *TCP/IP: Tutorial and Technical Overview*. 2006. 0738494682.
- [6] **SIEMENS.** SIMATIC. *ET 200S Positioning*. 2007. A5E00124871-04.
- [7] **ŠULC, M.** *Řídicí systém technologického úseku varna*. Brno : Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2012. Vedoucí bakalářské práce prof. Ing František Zezulka CSc..
- [8] PLC: SIMATIC S7-300. SIEMENS AG. [Online] Siemens. [Citace: 5. Leden 2014.] <http://www1.siemens.cz/ad/current/index.php?ctxnh=ee5ad951ae>.
- [9] **Microsoft.** *MSDN: Microsoft Developer Network*. [Online] 2014. <http://msdn.microsoft.com/>.
- [10] Motor knowledge. *SUNWIND*. [Online] Sunwind Electronics Company, Květen 2013-2014. <http://www.stepmotordriver.com/>.
- [11] **Doxygen.** JAI SDK. 2013. 1.8.2.
- [12] **Berry, Richard a Burnell, James.** *Image processing*. 2. Richmond : Willman-Bell, 2005. ISBN 0-943396-82-4.
- [13] **Zezulka, František.** *Automatizace procesů*. Brno : VUT, 2012. str. 155. skripta.

SEZNAM ZKRATEK A SYMBOLŮ

CCD	charge – coupled device
Castor	Počítač s OS Windows – program CaFe
FITS	Flexible Image Transport System
GPIO	General Purpose Input and Output
HŘS	Hlavní řídicí skript
MFC	Microsoft Foundation Class library
PLC	Programmable Logic Controller
Pollux	Počítač s OS Linux – HŘS

SEZNAM PŘÍLOH

Příloha 1. SOCOL – komunikační protokol

Příloha 2. Dodatek ke komunikačnímu protokolu ASCOL

Příloha 3. CaFe3 - zdrojové kódy