

**VŠB - Technická univerzita Ostrava**  
**Fakulta elektrotechniky a informatiky**  
**Katedra kybernetiky a biomedicínského inženýrství**

**Dohledový systém robotického ramena**  
**Surveillance System of Industrial Robot Arm**

**2016**

**Josef Navrátil**

# Zadání diplomové práce

Student: **Bc. Josef Navrátil**

Studijní program: N2649 Elektrotechnika

Studijní obor: 2601T004 Měřicí a řídicí technika

Téma: **Dohledový systém robotického ramena**  
**Surveillance System of Industrial Robot Arm**

Jazyk vypracování: čeština

## Zásady pro vypracování:

Při sériové výrobě průmyslových výrobků se používají roboty, avšak někdy interakce mezi robotem a výrobkem vede k poruchovému stavu, např. nedokonalému provedení výrobní operace nebo zničení výrobku. Účelem dohledového systému je během pohybu robota zaznamenávat obraz z kamer umístěných tak, aby následně bylo možné posoudit dodržení předepsané dráhy ramene. Obraz bude zaznamenáván alespoň 10x za sekundu v rozlišení 1280 x 1024. Zahájení pohybu inicializuje řídicí PLC. Diplomová práce se zabývá výběrem vhodných komponent pro dohledový systém, vývojem SW řešení pro dohledový systém a zpracování nasnímaných dat, nasazením u zákazníka a testováním provozu.

## Body zadání:

1. Rozbor problematiky práce s kamerami a obrazem v oblasti průmyslového testování.
2. Popis vývojových prostředí používaných pro návrh kamerových systémů - přehled a charakteristika.
3. Rozbor funkce a popis SW a HW komponent dohledového systému.
4. Návrh klíčových HW komponent.
5. Návrh a vývoj SW komponent dohledového systému v grafickém programovacím jazyku LabVIEW.
6. Nasazení a odladění záznamníku v průmyslové praxi.
7. Ověření funkce a zhodnocení.

## Seznam doporučené odborné literatury:

- [1] BRESS, Thomas J. *Effective labview programming*. 1st ed. Allendale: NTS Press, 2013, 701 s. ISBN 19-348-9108-8.
- [2] BITTER, Rick, Taqi MOHIUDDIN a Matt NAWROCKI. *LabView advanced programming techniques*. 2nd ed. Boca Raton: CRC Press, c2007, 499 s. ISBN 08-493-3325-3.
- [3] ŠONKA, Milan a Václav HLAVÁČ. *Počítačové vidění*. Praha: Grada, 1992. 252 s. ISBN 80-85424-67-3.
- [4] ŠONKA, Milan, Václav HLAVÁČ a Roger BOYLE. *Image processing, analysis, and machine vision*. 3rd ed. Toronto: Thomson, 2008, xxv, 829 s. ISBN 978-0-495-08252-1. Dostupné z: <http://www.engineering.uiowa.edu/~dip/LECTURE/lecture.html>.
- [5] HLAVÁČ, Václav. *Zpracování signálů a obrazů*. 1. vyd. Praha: Vydavatelství ČVUT, 2001, 220 s. ISBN 80-010-2114-9.
- [6] NATIONAL INSTRUMENTS. *NI Vision Assistant Tutorial*. VA\_Tutorial.PDF, 62 s., součást instalace LabVIEW 2013.
- [7] NATIONAL INSTRUMENTS. *NI Vision Builder for Automated Inspection Tutorial 2012*. VBAI\_Tutorial.pdf, 104 s., součást instalace LabVIEW 2013.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **doc. Ing. Petr Bilík, Ph.D.**

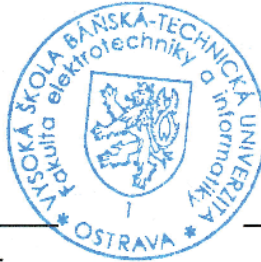
Datum zadání: 01.09.2015

Datum odevzdání: 29.04.2016



---

doc. Ing. Jiří Koziolek, Ph.D.  
*vedoucí katedry*



---

prof. RNDr. Václav Snášel, CSc.  
*děkan fakulty*

## Prohlášení Studenta

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Datum: 29.4.2016

Podpis: J. Nový

## **Abstrakt**

Práce se zabývá návrhem dohledového kamerového systému schopného trvalého nasazení v průmyslových podmínkách. Účelem systému je zaznamenávat pomocí kamer pohyb robotického ramene, u kterého dochází k nežádoucím chybovým situacím a poškození výrobku. Synchronizace záznamu s pracovními cykly robota zajišťuje signál z řídicího PLC. Kromě výběru vhodného hardwaru z oblasti strojového vidění obsahuje práce také návrh softwaru pro PC v programovacím jazyce LabVIEW a popis systémové integrace jednotlivých komponent. Následuje rozbor testování a ladění, především odstranění potíží spojených s trvalým provozem a zatížením velkým objemem obrazových dat.

## **Klíčová slova**

Kamerový systém, strojové vidění, záznam obrazu, robotické rameno, LabVIEW, vývoj softwaru, triggrování, systémová integrace

## **Abstract**

This work deals with design of camera surveillance system capable of permanent operation in industrial environment. The purpose of system is to record movement of robot arm, because its malfunction causes damage of products. Signal from master PLC is used to synchronize records with robot arm operation cycles. Thesis contains choice of suitable machine vision hardware, software implementation for PC platform in LabVIEW programming language and description of system integration of all components. Subsequently testing and debugging is described with special focus at solving problems followed from permanent operation and processing of large amount of visual data.

## **Key Words**

Camera system, machine vision, picture recording, robot arm, LabVIEW, software development, triggering, system integration

## Obsah

|       |  |    |
|-------|--|----|
| 1     | Úvod.....  | 8  |
| 2     | Požadavky zákazníka .....                                | 10 |
| 3     | Možnosti řešení .....                                    | 11 |
| 3.1   | Přehled.....   | 11 |
| 3.2   | IP kamery a SW pro zabezpečení budov .....               | 11 |
| 3.3   | Kamery pro strojové vidění a komerční SW pro záznam..... | 11 |
| 3.4   | Kamery pro strojové vidění a vývoj vlastního SW .....    | 12 |
| 4     | Použití kamer v průmyslu .....                           | 13 |
| 4.1   | Účel .....   | 13 |
| 4.2   | Přehled parametrů .....                                  | 13 |
| 4.3   | Technologie snímače.....                                 | 13 |
| 4.4   | Typ snímače .....  | 14 |
| 4.5   | Rozlišení.....   | 14 |
| 4.6   | Barevnost.....   | 15 |
| 4.7   | Rozhraní .....   | 15 |
| 4.8   | Frame grabber .....                                      | 17 |
| 4.9   | Smart kamery .....                                       | 17 |
| 5     | Vývojová prostředí pro návrh kamerových systémů.....     | 19 |
| 5.1   | LabVIEW firmy National Instruments.....                  | 19 |
| 5.2   | Vision Acquisition Software .....                        | 20 |
| 5.3   | Vision Development Module .....                          | 21 |
| 5.4   | Vision Builder for Automated Inspection .....            | 21 |
| 5.5   | Jazyk C# a Microsoft Visual Studio.....                  | 22 |
| 5.6   | AForge.NET .....   | 22 |
| 5.7   | OpenCV.....  | 23 |
| 6     | Výběr HW .....   | 24 |
| 6.1   | Základní struktura .....                                 | 24 |
| 6.2   | Kamery .....   | 24 |
| 6.2.1 | Výběr rozhraní.....                                      | 24 |
| 6.2.2 | Obrazové parametry .....                                 | 25 |
| 6.2.3 | Basler acA 1300-60gc .....                               | 25 |
| 6.2.4 | Umístění .....   | 26 |
| 6.3   | Triggrovací I/O modul .....                              | 26 |

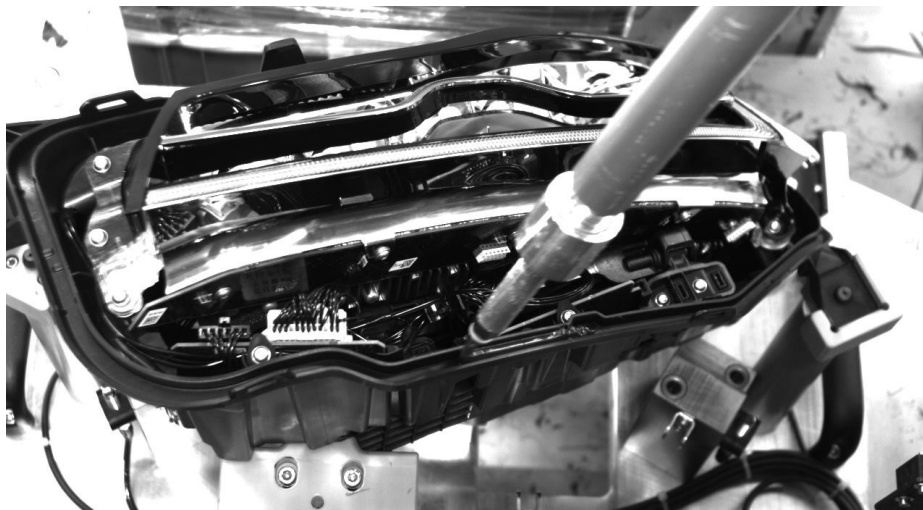
|       |  |    |
|-------|--|----|
| 6.4   | PC.....  | 28 |
| 6.5   | Karta NI PCIe-8236 .....                             | 29 |
| 6.6   | Karta Moxa CP-102E.....                              | 30 |
| 6.7   | Objektivy.....                                       | 30 |
| 6.8   | Schéma zapojení.....                                 | 30 |
| 7     | Software .....                                       | 32 |
| 7.1   | Analýza požadavků .....                              | 32 |
| 7.2   | Modularita.....                                      | 32 |
| 7.3   | Šablona modulu.....                                  | 33 |
| 7.4   | Volba způsobu ukládání obrazu .....                  | 34 |
| 7.4.1 | Struktura a formát.....                              | 34 |
| 7.4.2 | Členění a označení záznamů .....                     | 35 |
| 7.5   | Hlavní úloha aplikace – main.....                    | 36 |
| 7.6   | SW modul trigger.....                                | 38 |
| 7.7   | SW modul kamera.....                                 | 39 |
| 7.8   | SW modul pro ukládání dat.....                       | 40 |
| 7.9   | SW Modul uživatelského rozhraní .....                | 41 |
| 7.9.1 | Wire-frames.....                                     | 41 |
| 7.9.2 | Realizace v LabVIEW.....                             | 43 |
| 8     | Testování a ladění.....                              | 46 |
| 8.1   | Postup a podmínky .....                              | 46 |
| 8.2   | Obsazení operační paměti .....                       | 46 |
| 8.3   | Chyba memory manageru LabVIEW.....                   | 47 |
| 8.3.1 | Výskyt chyby.....                                    | 47 |
| 8.3.2 | Neplatný ukazatel nebo únik paměti .....             | 48 |
| 8.3.3 | Fragmentace operační paměti PC.....                  | 48 |
| 8.3.4 | Nevhodná strategie překreslování obrazů z kamer..... | 48 |
| 8.3.5 | Řešení.....  | 49 |
| 9     | Nasazení .....                                       | 50 |
| 10    | Závěr .....  | 51 |
| 11    | Citovaná literatura .....                            | 52 |
| 12    | Seznam příloh.....                                   | 55 |
| 12.1  | Elektronické přílohy (obsah CD) .....                | 55 |

# 1 Úvod

Podkladem pro tuto diplomovou práci je projekt realizovaný ve firmě ATEsystem s.r.o., určený pro průmyslové nasazení u zákazníka. Vznikl na základě poptávky kamerového systému pro dohled nad pracovištěm robotického ramene, s potenciálem nabídnout vzniklé řešení i dalším zákazníkům.

V průmyslové praxi se využívají roboty pro plnění řady úkolů, při kterých vykonávají předem pevně naprogramovaný pohyb. Tomu odpovídá i pracoviště zákazníka, které mělo být vybaveno kamerovým systémem. Robotické rameno zde dávkuje skrze kovovou jehlu lepidlo na plastový výlisek. Vzhledem k tomu, že lepidlo má být nanášeno do drážky umístěné podél obvodu výrobku, jejíž šířka pouze mírně přesahuje průměr dávkovací jehly, je tato operace náročná na přesnost pohybu robota. Přestože přesnost a opakovatelnost výroby představuje jeden z důvodů použití robotů, nelze chybu úplně vyloučit. Zejména pokud nebyla správně provedena kalibrace a korekce dráhy pohybu robota. Rovněž se negativně může projevit lidský faktor a to zejména při špatném založení výrobku do palety, která slouží k upevnění během celého procesu. Tento úkol totiž provádí ručně obsluha výrobní linky. Dalším problémem je možnost výskytu vadných kusů, které neodpovídají výrobním tolerancím, protože u nich došlo například ke zkroucení. Všechny tři výše uvedené faktory – nepřesnost robota, chyba obsluhy i vada na výrobku mohou vést k poruchovému stavu.

Vzhledem k tomu, že výlisek i dávkovací jehla jsou z poměrně měkkého materiálu, hrozí při kontaktu jehly s drážkou k poškození na obou stranách. Jednak znehodnocení výrobku, protože poškozenou drážku nelze posléze slícovat s lepeným dílem, ale také ohnutí dávkovací jehly. Poškození výrobku zpravidla zjistí obsluha při vyjmutí z palety, ohnutí jehly detekuje řídicí PLC, protože na konci cyklu se kontroluje její průjezd vymezeným prostorem. V takovém případě je nutné zastavit výrobu a jehlu vyměnit. Pochopitelně přerušená výroba, poškozená jehla nebo plastový díl představují nežádoucí finanční ztráty, které je potřeba eliminovat na minimum.



Obr. 1 Dávkování lepidla

Je krajně nevhodné, aby kontrolu chodu robota nějakým způsobem prováděla lidská obsluha například zrakem, rovněž to není možné z bezpečnostních důvodů. Pracoviště robota je při běžném provozu uzavřeno v ochranné kleci a vstup osob je zakázán. Přímo se tedy nabízí použití kamerového systému a to hned ve dvou možných rolích. Za prvé lze pomoci metod zpracování obrazu v rámci strojového vidění



detekovat pohyb ramene a kontrolovat jeho dráhu, respektive porovnat ji s požadovanou v rámci daných tolerancí. V případě detekce odchylky přesahující toleranci by mohl kamerový systém aktivně zasáhnout a výrobní linku zastavit ještě před poškozením dílů. Rovněž může být kamerový systém pouze v roli pasivní, zaznamenávat obraz z jednotlivých výrobních cyklů a tento nasbíraný materiál poslouží obsluze k pozdější analýze chybových stavů.

Již na první pohled je jasné, že první řešení obsahující zpracování obrazu by bylo technicky náročnější a to konkrétně vývoj potřebného softwaru. Hardwarové vybavení by s největší pravděpodobností pro oba systémy mohlo být stejné. Vývoj softwaru ovšem vyžaduje čas, který má podstatný vliv na cenu celkového řešení. Právě z finančních důvodů zákazník zvolil variantu pouze pasivního záznamníku obrazu.

Obsahem této práce je tedy analýza požadavků, jejich konfrontace s technickými možnostmi a návrh řešení. Práce rovněž obsahuje popis HW komponent potřebných pro realizaci, dále principy funkce softwaru navrženého a implementovaného pro kamerový systém v jazyce LabVIEW. Samostatná kapitola je věnována testování a ladění problémů, které se objevily vzhledem k nárokům na trvalý provoz systému a jeho průmyslovému nasazení.

Při návrhu byl kladen důraz na to, aby řešení nejen odpovídalo požadavkům, ale rovněž bylo i o něco univerzálnější, například se širšími možnostmi nastavení, než by pro tuto aplikaci bylo nutné. Týká se to v první řadě softwaru, jehož vývoj je časově nejnáročnější položkou v realizaci takového projektu. Zvolená struktura a použité programátorské přístupy umožní opakované požití, respektive přizpůsobení nějakému jinému, ale principiálně podobnému projektu.

## 2 Požadavky zákazníka

V následujících bodech jsou shrnuty požadavky a podmínky, které byly výsledkem konzultace se zákazníkem. Dva z nich (kurzívou) byly následně v průběhu vývoje upraveny dle dostupného HW a na základě výsledků testů. Jednak byla vybrána kamera s nižším rozlišením, což je podrobněji rozepsáno v kapitole 6.2.2. Snížena byla také provozní snímkovací frekvence. Zákazník dostal k dispozici zkušební záznam pohybu robota pořízený hrubým konceptem aplikace v rámci kamerových testů přímo na pracovišti. Závěr ze zkoumání zkušebního záznamu byl zohledněn ve fázi testování (kapitola 8) a dále se uvažovalo dlouhodobé zatížení záznamem pouze při 15 fps. Vyšší snímkovací frekvence může být použita pro zvláštní účely.

- Záznam barevného obrazu ze dvou kamer.
- Rozlišení každé kamery *1920 x 1080 pixelů*
- Snímkovací frekvence *60 fps*
- Mechanika nutná pro montáž kamer není součástí dodávky – řeší zákazník
- Začátek pracovního cyklu robota, respektive start záznamu bude signalizován napětovým impulzem délky 1 s z řídicího PLC. Kamerový systém musí tento signál detekovat jako triggerovací událost.
- Zatížení v rozmezí 1200 – 1500 pracovních cyklů za 24 hodin.
- Možnost nastavitelného časového zpoždění nahrávání vůči startovacímu impulzu pro každou kameru zvlášť.
- Po každém triggeru se bude ukládat do nového adresáře, jehož název bude odvozen od aktuálního data a času.
- Ruční generování (simulace) triggerovací události bez pulzu z PLC.
- Možnost ručního spuštění záznamu z jedné nebo obou kamer bez triggerovací události.
- Aplikace se bude spouštět automaticky při startu PC, uživatelský účet Windows nebude chráněn heslem
- Aplikace musí být po startu plně funkční bez nutnosti zásahů obsluhy
- Velikost okna uživatelského rozhraní vhodná pro monitor s rozlišením 1280 x 1024
- Monitor není součástí dodávky kamerového systému, zajišťuje jej sám zákazník. Po otestování se předpokládá i možnost nasazení bez monitoru, pouze se vzdálenou správou.
- Otestování systému se vzdálenou správou pomocí softwaru Radmin.
- Živý náhled obrazu z kamer v uživatelském rozhraní.
- Možnost přepínat zobrazení jedné kamery nebo obou.
- Volitelné (lze vypnout) umístění časové značky přímo do obrazu kamery.
- Možnost konfigurování základních parametrů kamery – expoziční čas, snímkový kmitočet, citlivost (zisk neboli gain).
- SW neřeší prohlížení nasnímaných dat – k tomuto účelu bude sloužit prohlížeč souborů a obrázků, který je součástí OS Windows.
- Nasazení aplikace v třísměnném provozu s přestávkami maximálně 30 min a to v časech 10:00 – 10:30, 18:00 – 18:30 a 02:00 – 02:30. O víkendech je odstaven zdroj elektrické energie, obvykle od pátku 22:00 do neděle 20:00.
- Minimalizace kabeláže a zásahů do současné instalace.

## 3 Možnosti řešení

### 3.1 Přehled

Před návrhem vlastního řešení byl proveden průzkum trhu, zda není možné požadavky zákazníka uspokojit vhodným komerčně dostupným produktem, případně jeho přizpůsobením. V podstatě se nabízí celkově 3 možné alternativy:

1. Použití IP kamer a odpovídajícího softwaru, tedy vybavení určeného pro oblast zabezpečení budov a majetku.
2. Použití kamer pro strojové vidění a komerčního SW pro záznam.
3. Použití kamer pro strojové vidění a vývoj vlastního SW na míru potřebám zákazníka.

Podrobněji jsou aspekty jednotlivých variant shrnuty v následujících kapitolách.

### 3.2 IP kamery a SW pro zabezpečení budov

Oblast zabezpečovací techniky představuje na trhu velmi rozsáhlou oblast se širokou nabídkou hardwarových i softwarových produktů. Většina požadavků zákazníka se týká celkové funkce kamerového systému respektive softwaru, klíčové je tedy zaměřit se právě na jeho výběr. Problematické jsou zejména dva body z požadavků – triggrování z PLC a funkčnost aplikace po startu PC, aniž by byl vyžadován zásah obsluhy. Reakce na digitální vstupy nabízí pouze některé softwary, které zpravidla spadají do kategorie komplexních řešení pro náročné aplikace. Princip funkce je takový, že digitálním vstupem je vybavena kamera a software kromě snímání obrazu zjišťuje i jeho stav (vstup slouží např. k připojení detektoru otevření dveří/oken). Příkladem je kombinace SW AXIS Camera Station [1] a kamer například Axis M1125 [2]. Jasnou nevýhodou je složitost SW, který obsahuje i řadu funkcí, které zákazník nepotřebuje, ale musí za ně zaplatit. Z hlediska HW je nevhodné umístění digitálního vstupu na kameře, protože ta bude v pracovním prostoru robota. Zatímco PLC se nachází v rozvaděči, který je součástí technického zázemí pracoviště. Nejvhodnější je tedy umístit PC kamerového systému poblíž rozvaděče a triggrovací signál do něj vést co nejkratší trasou (skrze převodníkovou kartu), nikoliv přidávat kabel směřující do pracovního prostoru.

Dalšími alternativami je například SW Luxriot VMS [3] nebo SW iSpy, který je dokonce zdarma jako open-source projekt. V dokumentaci ani jednoho z nich není ale deklarována možnost HW triggrování. Nevýhodou open-source projektu může být jeho vlastní podstata – funguje díky nadšení dobrovolníků, kteří ve volném čase vylepšují a udržují jeho kód. Z hlediska průmyslového nasazení to znamená nulovou záruku technické podpory a tedy i obtížné řešení případných problémů.

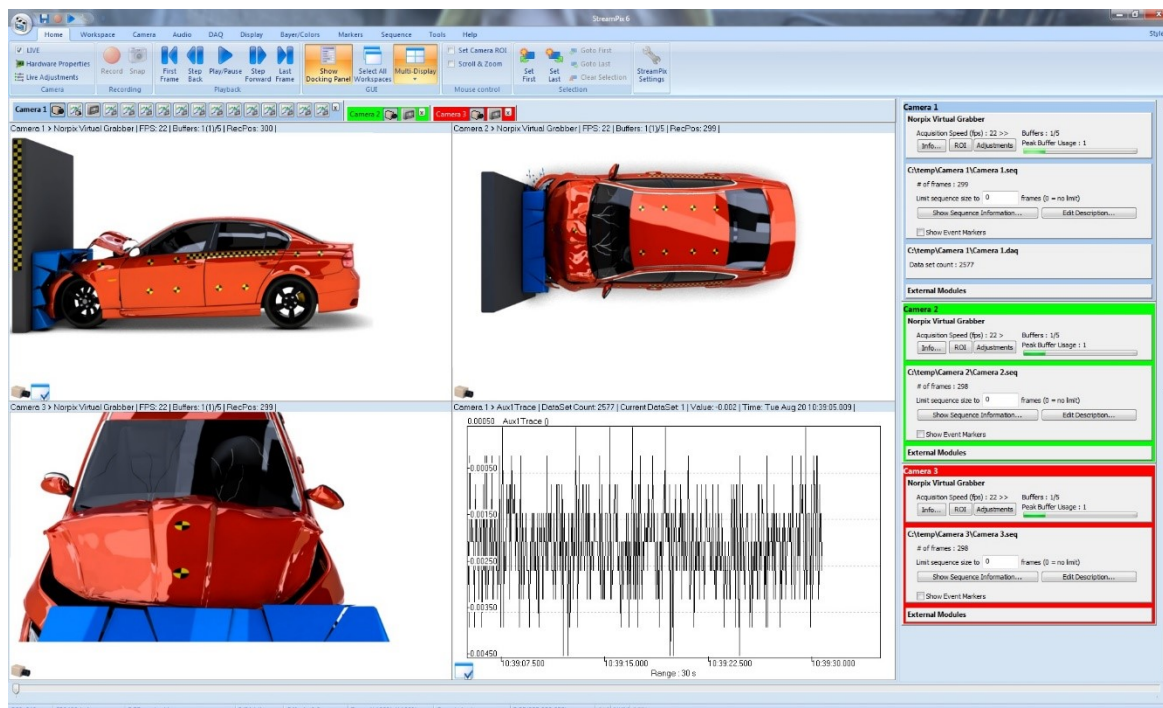
Řešení s IP kamerami a zabezpečovacím softwarem je rovněž nevhodné z hlediska dalšího vývoje. Pokud by se zákazník časem rozhodl pro rozšíření o zpracování a vyhodnocení obrazu, byl by zakoupení software nepoužitelný. Navíc IP kamery jsou zdrojem již komprimovaného obrazu, který není vhodný pro další zpracování. Konkrétně kamery Axis M1125 využívají při kompresi standardu H.264 [2].

### 3.3 Kamery pro strojové vidění a komerční SW pro záznam

Pro oblast strojového vidění, vědy a výzkumu nabízí firma Norpix software pro nahrávání videa pod názvem StreamPix. Možnosti tohoto softwaru jsou poměrně široké, kromě záznamu a přehrávání umožňuje i úpravy barevného podání, výběr z řady kompresních formátů, obsluhu I/O zařízení nebo vzdálenou kontrolu pomocí StreamPix Remote. Kromě podpory digitálních vstupů/výstupů přímo na

kamerách je StreamPix schopen pracovat i s měřicími kartami National Instruments. Dle specifikace je kompatibilní se všemi zařízeními podporovanými balíkem ovladačů NI-DAQmx. [4]

Příkladem použití StreamPix je záznam laboratorních experimentů nebo crashtestů automobilů pomocí vysokorychlostních kamer. Daní za univerzálnost softwaru je nutnost orientovat se v řadě nastavení a složitém uživatelském rozhraní. Tento fakt není na závadu při laboratorním použití, kdy před startem experimentu obsluha otevře a spustí projekt s potřebnými nastaveními. Spouštění s předem danou konfigurací ale nelze automatizovat, takže SW by nebyl schopen po spuštění PC začít ihned pracovat bez zásahu lidské obsluhy.



Obr. 2 Software StreamPix [4]

### 3.4 Kamery pro strojové vidění a vývoj vlastního SW

Vývoj vlastního softwaru vždy vyžaduje nezanedbatelnou časovou investici, nejen přímo do programování, ale následně i do testování a ladění. Na druhou stranu umožňuje vytvořit řešení přímo odpovídající požadavkům, které navíc lze dále rozšiřovat – nejedná se o uzavřený komerční software. Navíc se na trhu nepodařilo nelézt produkt, který by zákazníkovi vyhovoval, právě proto byla zvolena varianta vlastního vývoje a to v grafickém prostředí LabVIEW. Cílem je navrhnout aplikaci maximálně jednoduchou, robustní a spolehlivou. Případně doplňující funkce nebo možnosti nastavení, které nejsou přímo vyžadovány specifikací, ale jsou výhodné z hlediska dalšího použití SW, nesmí zbytečně komplikovat uživatelské rozhraní a je vhodné je začlenit například do konfiguračního souboru. Právě načtením konfigurace z takového souboru je možné elegantně vyřešit start aplikace bez nutnosti zásahu obsluhy. Uživatelské rozhraní zůstane velmi jednoduché, ale přesto bude možné aplikaci přizpůsobit bez zásahů do zdrojového kódu.

## 4 Použití kamer v průmyslu

### 4.1 Účel

Se stále rostoucím objemem vyráběných produktů a souběžně zvyšujícími se nároky na kontrolu kvality přestává být akceptovatelné a mnohdy i prakticky realizovatelné, aby tuto kontrolu prováděli lidé. Například ověření správného tvaru (absence vad nebo deformací) a rozměrů každého výrobku na konci výrobní linky je při dnešní masové produkci mechanická a velmi jednotvárná práce. Po několika hodinách na směně může být větší riziko chyby kontrolora než výskytu skutečně vadného výrobku. Do role kontroly kvality jsou tedy postaveny kamerové systémy složené obvykle z podobných základních komponent, jako je objektiv s kamerou vybavenou vhodným komunikačním rozhraním, následně vyhodnocovací jednotka například PC, případně jiné procesorem vybavené zařízení, na kterém běží speciální software pro zpracování získaného obrazu. Výsledek testu se prezentuje graficky na monitoru a vhodným způsobem – například v podobě digitálního výstupu (prošel/neprošel – dobrý/špatný kus) se předává výrobní lince, která zmetky separuje.

Ne vždy je nezbytně nutné lidský dohled úplně eliminovat, ale stačí jej určitým způsobem usnadnit. Příkladem je umístění kamer do takových míst respektive fází výrobního procesu, kde je přístup člověka buď velmi nepohodlný, nebo úplně nemožný s ohledem na okolní podmínky – vysoké teploty, stísněné prostory, přítomnost nebezpečných látek atd. V takovém případě se obvykle nejedná o kontrolu každého vyráběného kusu, ale o spíše celkový dohled nad technologií výroby. Obsluha pracuje v zázemí výrobní linky s možností zobrazení obrazů z kamer v jednotlivých fázích výrobního procesu. V některých případech je žádoucí i záznam obrazu a jeho uchování po určitou dobu, podobně jako je to vyžadováno u aplikace, která je předmětem této práce.

### 4.2 Přehled parametrů

Existuje řada kritérií a parametrů, podle kterých lze kamery dělit do kategorií. Mezi ty základní patří:

- Technologie snímače: CMOS nebo CCD
- Typ snímače: řádkový nebo plošný (maticový)
- Rozlišení: od VGA po 10 a více megapixelů
- Barevnost: monochromatické nebo barevné
- Maximální snímkovácí kmitočet: od 10 do řádově stovek snímků za sekundu
- Rozhraní: Gigabit Ethernet, USB 2.0 nebo 3.0, FireWire, Camera Link

Při volbě kamery je nutné zohlednit všechna kritéria, jejich vliv na zamýšlenou aplikaci a nalézt přijatelný kompromis, protože z důvodu finanční náročnosti pochopitelně nelze vybírat vždy to nejlepší z dané kategorie. Rovněž může být dosti obtížné rozhodnout, co vůbec označit za nejlepší, protože všechny popsané technologie, rozhraní atd. mají své výhody a nevýhody.

### 4.3 Technologie snímače

Základním principem použitým v obou technologiích je fotoelektrický jev, kdy se v jednotlivých elementech snímače (pixelech) uvolní po dopadu fotonů elektrony. Co se s nimi stane dále, už závisí právě na použité technologii. CCD je zkratka anglických slov charge-coupled device, tedy zařízení s vázanými náboji. V tomto typu snímače se náboj tvořený elektrony odvádí postupně z pixelů pryč a teprve pak se mimo ně převede na napětí. Oproti tomu CMOS (complementary metal oxide semiconductor) obsahuje v každém pixelu několik elektronických součástí potřebných k převedení

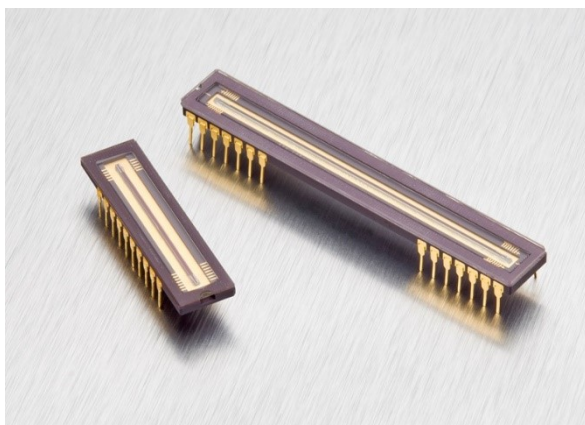
uvolněných elektronů na napětí, které je pak multiplexovaně měřeno. Z této odlišné formy zpracování signálu vyplývají odlišnosti shrnuté v Tab. 1. [5]

**Tab. 1 Srovnání CCD a CMOS čipů**

|                              | <b>CCD</b>        | <b>CMOS</b>       |
|------------------------------|-------------------|-------------------|
| Citlivost na světlo          | Vysoká            | Střední           |
| Šum                          | Nízký             | Střední až vysoký |
| Rychlost snímání             | Střední           | Vysoká            |
| Spotřeba                     | Střední až vysoká | Nízká             |
| Dynamický rozsah             | Vysoký            | Střední           |
| Odolnost proti přeexponování | Nízká             | Vysoká            |
| Cena                         | Vysoká            | Nízká             |

#### 4.4 Typ snímače

Podle aplikace, pro kterou je kamera určena, je potřeba vybrat buď plošný (maticový) nebo řádkový snímač. Plošný čip respektive kamera odpovídá běžné představě o snímání obrazu, tedy ve dvou rozměrech – obraz má konkrétní šířku a výšku. V určitých případech se ale více hodí snímat vždy pouze jeden řádek – jednorozměrný obraz. Takovéto kamery tedy obsahují senzor tvořený jen jednou řadou pixelů. Typickou možností nasazení je snímání objektů, které se pohybují předem známým a definovaným způsobem, například výrobky jedoucí na dopravníku nebo odvíjený pás tkaniny. Druhý rozměr snímku se vytváří postupně snímáním dalších a dalších řádků pohybujícího se objektu. Výhodou řádkových kamer je vysoká rychlost, frekvence snímání řádkového senzoru se pohybuje řádově v desítkách kHz.



**Obr. 3 Snímač řádkové kamery [6]**

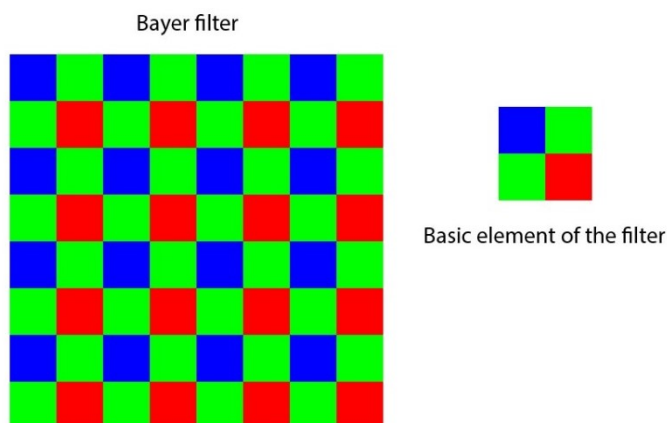
#### 4.5 Rozlišení

Pod pojmem rozlišení se skrývá údaj o počtu pixelů na snímači, u plošných kamer se udává buď celkové číslo, např. 5 Mpx nebo součin obou rozměrů, např. 1600 x 1200 px. K charakterizování řádkové kamery stačí údaj jeden. Větší počet pixelů automaticky nemusí znamenat vyšší kvalitu obrazu, protože ta závisí na dalších faktorech – fyzický rozměr čipu, použitá technologie a samozřejmě i na optice objektivu. S tím souvisí i zásadní rozdíl mezi kamerami pro strojové vidění a kamerami pro spotřební elektroniku. Pro strojové zpracování obrazu je mnohdy výhodnější mít obraz s nižším rozlišením, ale zato ve vysoké

kvalitě – nízký šum, ostrá a detailní kresba snímaného objektu. S rostoucím rozlišením se totiž zvyšují nároky na výpočetní výkon. Výrobci spotřební elektroniky naopak nabízejí stále vyšší a vyšší rozlišení bez ohledu na to, zda při malém čipu a levné optice má informace z tak vysokého množství pixelů reálné využití. Proto portfolia výrobců kamer pro strojové vidění stále obsahují i kamery s rozlišením VGA (640 x 480 px), které v jiném kontextu vyznívá až archaicky.

## 4.6 Barevnost

CCD i CMOS technologie dovedou změřit pouze dopadající množství světla, nikoliv jeho barvu. Všechny jsou tedy v základu černobílé. Přidání barevných filtrů (červený, zelený, modrý - Obr. 4) před jednotlivé pixely a následným výpočtem pomocí jednotlivých barevných složek z několika pixelů je možné informaci o barvě dodatečně získat. Barevný filtr se označuje jako Bayerův. Pro strojové vidění vyplývá z tohoto principu jedna zásadní nevýhoda, výpočet jednoho barevného pixelu z několika černobílých (typicky ze čtyř) snižuje přesnost kamery, například dochází k rozmazání hran, proto pro měření jsou vhodnější kamery černobílé. Druhou možností je použít tři oddělené čipy, jinak řečeno pro každou barvu (R, G, B) jeden. Dopadající světlo je mezi čipy rozděleno na barevné složky pomocí optického hranolu. Toto řešení je pochopitelně podstatně dražší a hodí se pro náročnější aplikace. [5]



Obr. 4 Bayerův filtr [7]

## 4.7 Rozhraní

Gigabit Ethernet je často využívaným rozhraním v oblasti strojového vidění, protože nabízí velmi příznivý poměr ceny a výkonu. Pro stavbu kamerového systému lze využít běžně dostupné komponenty, jako jsou síťové karty pro PC, switche a kabeláž pro LAN. Nad fyzickou vrstvou Ethernetu se uplatňuje standard GigE Vision, který definuje jakým způsobem se obrazové data a povely k nastavení kamery přenášejí. Délka kabelu mezi kamerou a dalším síťovým prvkem může být až 100 m při propustnosti teoreticky až 100 MB/s. Navíc, pokud kamera podporuje technologii PoE (Power over Ethernet), lze ji prostřednictvím datového kabelu i napájet. Z hlediska časování je Ethernet schopen zajistit trigger s přesností okolo jedné milisekundy, při vyšších požadavcích musí být doplněn hardwarovým – digitální vstup na kameře. [8] [9] Nevýhodou může být zvýšené zatížení CPU počítače, pokud není použita vhodná síťová karta a ovladače, výběru obojího je tedy třeba věnovat pozornost.

FireWire jinak také známé pod označením IEEE 1394 bylo řadu let jedinou standardní možností, jak kameru k PC připojit. Nicméně v dnešní době je na ústupu, protože svými technickými možnostmi za ostatními zaostává a nebývá už běžnou součástí PC, jako například Ethernet nebo USB 3.0. Dle standardu IEEE 1394b je propustnost maximálně 64 MB/s a délka kabeláže 4,5 m. Mezi výhody patří

nízké zatížení CPU počítače a nízká latence. FireWire není vhodným kandidátem při návrhu nového systému, je v pozici dosluhujícího rozhraní u starších aplikací.

Camera link má mezi rozhraními určité výjimečné postavení. Je provázeno nevýhodami jako je velmi nákladná kabeláž nebo dodatečný hardware, kterým je potřeba rozšířit PC – cena zásuvné karty s Camera Link obvykle přesahuje cenu celého počítače. Je to dáno faktem, že Camera Link byl navržen výhradně pro oblast zpracování obrazu a cena tedy není snižována masovou výrobou komponent i pro oblast spotřební elektroniky jako například u Ethernetu nebo USB 3.0. Na druhou stranu je toto rozhraní velmi robustní a nabízí propustnost až 850 MB/s, takže v náročných aplikacích nemá vhodnou alternativu. [8]

Zatímco USB ve verzi 2.0 se v oblasti strojového vidění a obecně v průmyslu neprosadilo, u nástupce USB 3.0 je tomu jinak. Disponuje násobně vyšší přenosovou rychlostí až 350 MB/s s maximální délkou kabelu 8 m. Nezatěžuje příliš CPU počítače díky použití DMA - direct memory access, nabízí nízkou latenci a jitter. Klíčová je už běžná dostupnost rozhraní na nových PC, stejně tak možnost napájení kamery přímo z portu. Standard USB3 Vision je zatím poměrně mladý, vznikl v roce 2013. Hlavní nevýhodou oproti Ethernetu je omezená délka kabeláže. [8]

Tab. 2 Srovnání rozhraní pro kamery [8] [9]

|                                      | Gigabit Ethernet | FireWire (IEEE1394b) | Camera Link | USB 3.0   | USB 2.0   |
|--------------------------------------|------------------|----------------------|-------------|-----------|-----------|
| <b>Propustnost (MB/s)</b>            | 100              | 64                   | 850         | 350       | 40        |
| <b>Délka kabelu (m)</b>              | 100              | 4,5                  | 10          | 8         | 5         |
| <b>Náročnost systémové integrace</b> | Nízká            | Střední              | Vysoká      | Nízká     | Nízká     |
| <b>Zatížení CPU</b>                  | Střední          | Nízké                | Střední     | Nízké     | Nízké     |
| <b>Napájení zařízení</b>             | Ano - PoE        | Ano                  | Ano - PoCL  | Ano       | Ano       |
| <b>Připojení několika kamer</b>      | Ano - switch     | Ano                  | Ne          | Ano - hub | Ano - hub |
| <b>Latence a jitter</b>              | Střední          | Nízká                | Nízká       | Nízká     | Vysoká    |

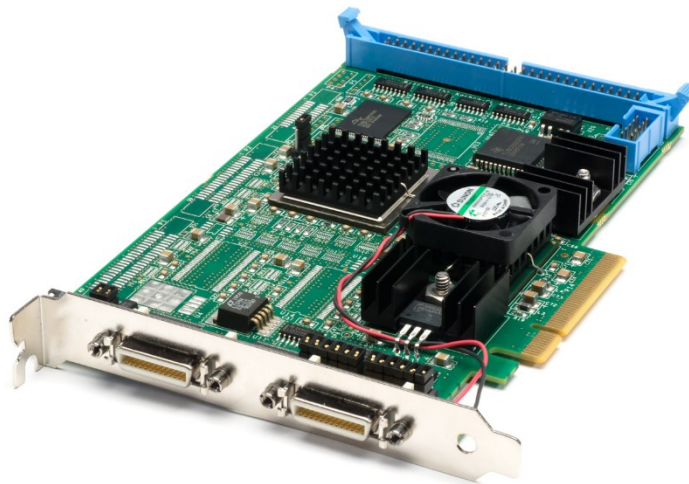


Obr. 5 Srovnání vzhledu konektorů pro tři různá rozhraní na kameře z řady Basler ace [10]



## 4.8 Frame grabber

V kapitole 4.7 bylo zmíněno, že některá rozhraní pro připojení kamer vyžadují rozšíření PC o dodatečný hardware, pro který se používá název frame grabber. V oblasti strojového vidění se typicky jedná o kartu do slotu PCI nebo PCI Express, která obsahuje jeden či více konektorů s příslušným rozhraním – například Camera Link. Existují frame grabbery i pro rozhraní, která se na PC běžně vyskytují, tedy gigabitový Ethernet nebo USB 3.0. Důvodem je, že hardware dostupný na trhu s výpočetní technikou nemusí vždy svými parametry odpovídat nárokům strojového vidění, zejména u aplikací s vysokým rozlišením či snímkovým kmitočtem. Zpracování obrazu je samozřejmě možné provádět i na jiné platformě, například PXI, takže portfolio frame grabberů rovněž není omezeno jen na platformu PC.



Obr. 6 Frame Grabber pro dvě kamery s rozhraním Camera Link [11]

Jako frame grabber se někdy označují například i USB externí karty pro digitalizaci analogového videa, třeba ze starších video kamer nebo zabezpečovacích kamer. Pro oblast machine vision ale nemají význam, proto nebudou dále rozebírány.

## 4.9 Smart kamery

Pod pojmem smart camera (chytrá kamera) se skrývá kompaktní systém, který v jednom pouzdře kombinuje nejen kameru, ale rovněž i hardware pro zpracování obrazu a také výstupy pro signalizaci výsledku inspekce. Výhodou takového řešení je velmi snadná integrace s minimálním využitým prostorem například do řídicích systémů na bázi PLC. Smart kamera plní úlohu senzoru, jehož algoritmus pro vyhodnocení obrazu lze naprogramovat a nadřazenému PLC pomocí digitálního výstupu předává pouze informaci o tom, zda zkoumaný kus je v pořádku či nikoliv. Data z komplexnějších úloh – měření, čtení čárových kódů mohou být do řídicího systému přenášena například pomocí Ethernetu, pokud jím daný model kamery disponuje.

Chytré kamery nabízí výrobci jako Omron, Sick nebo National Instruments. Poslední jmenovaný rovnou celou výkonnostní řadu rozdělenou do dvou hlavních kategorií – low cost a high performance. Přehled možností chytrých kamer National Instruments je v Tab. 3. Software pro zpracování obrazu spuštěný v kameře je možné vytvořit buď ve Vision Builder for Automated Inspection nebo LabVIEW. [12] [13]

Tab. 3 Parametry chytrých kamer National Instruments [12] [14]

|                                       | <b>Low cost</b>            | <b>High performance</b> |
|---------------------------------------|----------------------------|-------------------------|
| <b>Procesor</b>                       | Power PC 400 MHz – 600 MHz | Intel Atom 1,6 GHz      |
| <b>RAM</b>                            | 256 MB                     | 512 MB                  |
| <b>Uložiště</b>                       | 512 MB                     | 2 GB                    |
| <b>Rozlišení</b>                      | 640 x 480 nebo 1280 x 1024 | 640 x 480 až 5 MPx      |
| <b>Barevný snímač</b>                 | Ne                         | Některé modely          |
| <b>Snímkový kmitočet</b>              | 60 nebo 13 fps             | 15 fps až 110 fps       |
| <b>Dig. vstupy/výstupy</b>            | 1/1 až 2/2                 | 4/4                     |
| <b>Komunikace</b>                     | RS232, Ethernet            | RS232, Gigabit Ethernet |
| <b>Triggrování osvětlení</b>          | Některé modely             | Ano                     |
| <b>Vstup pro inkrementální snímač</b> | Některé modely             | NE                      |
| <b>VGA výstup pro monitor</b>         | Ne                         | Ano                     |
| <b>IP67</b>                           | Ne                         | Ano                     |



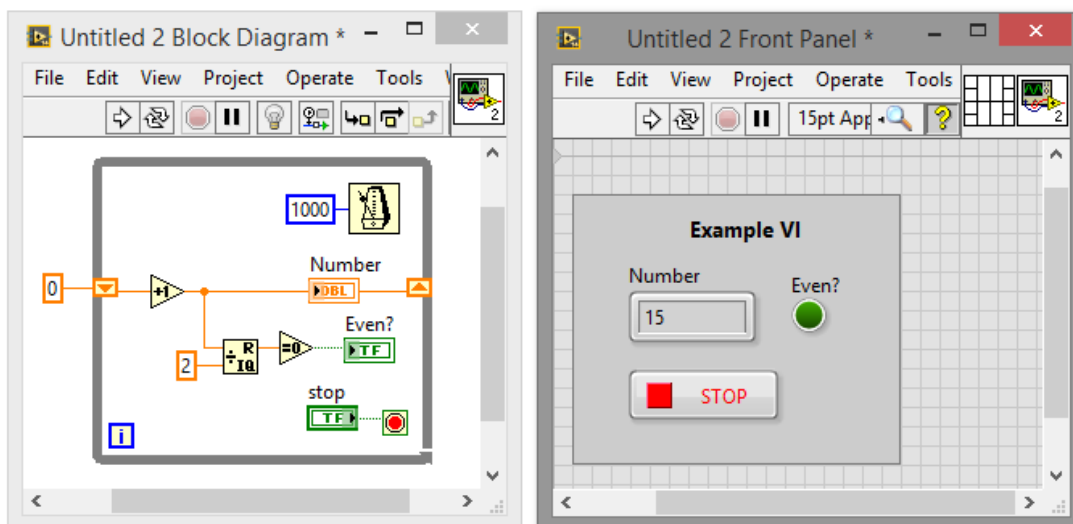
Obr. 7 Smart kamera NI 1774 [15]

## 5 Vývojová prostředí pro návrh kamerových systémů

### 5.1 LabVIEW firmy National Instruments

LabVIEW je grafický programovací jazyk od společnosti National Instruments s již 30 letou tradicí – verze 1.0 určená pro počítače Macintosh byla uvedena na trh v roce 1986. [16] Právě grafické pojetí vytváření kódu jej zásadně odlišuje od jiných ryze textových jazyků jako je C/C++/C#, Visual Basic, Python atd. LabVIEW bylo od počátku zamýšleno a navrženo především pro řešení inženýrských problémů zejména z oblasti měření a testování. Ačkoliv nelze tvrdit, že by v LabVIEW nešel naprogramovat například textový editor nebo hudební přehrávač, jeho síla se projeví až u aplikací z technické praxe. Příkladem je interakce s reálným světem pomocí měřicích přístrojů a karet, generování signálů, zpracování a analýza naměřených dat, jejich následná vizualizace atd. Cílem LabVIEW je poskytnout ucelenou sadu nástrojů pokrývající technické disciplíny tak, aby se minimalizoval čas investovaný do vývoje dané aplikace. To by pochopitelně nebylo možné realizovat pouze SW prostředky a proto National Instruments nabízí i potřebný HW v podobě měřicích karet, real-time systémů, komponent pro zpracování obrazů a dalších. HW a SW je pochopitelně velmi úzce propojen, protože jeden bez druhého nemají smysluplné využití. Pro LabVIEW tedy existují rozsáhlé knihovny a ovladače pro podporu HW od National Instruments, ale i třetích stran. LabVIEW spolu s knihovnami a HW tvoří vývojovou platformu využitelnou pro plnění širokého spektra úkolů, aniž by bylo nutné se jimi zabývat na nejnižší (tzv. low-level) úrovni. [17]

Pokud je například cílem automatizovaně otestovat nějaké zařízení přivedením určitého analogového signálu a měřením jeho odezvy, není žádoucí nejprve zdlouhavě řešit pomocí jakého integrovaného obvodu se tak stane, jak ho připojit k PC, navrhovat komunikační protokol, následně vytvářet ovladač pro zvolený programovací jazyk atd. S ohledem na minimalizaci časové a finanční náročnosti je potřeba sáhnout po ověřených a funkčních komponentách, které dovedou příslušný typ signálu generovat a měřit, a ke kterým je poskytována i SW podpora. Zásadní výhodou takovéto platformy je záruka kompatibility, která nemusí být dosažena, pokud vývojové prostředí dodává nezávisle na ostatních firma *A*, hardware firma *B* a ovladače vznikly za přispění společnosti *C*.



Obr. 8 Vývojové prostředí LabVIEW, vlevo diagram, vpravo návrh čelního panelu

V zásadě stejná situace vzniká ale i u vývoje samotného SW. Například v případě potřeby změřená data převést do frekvenční oblasti pomocí FFT, je žádoucí mít pro takovou operaci hotovou knihovni funkci, nikoliv se zabývat návrhem vlastního algoritmu FFT. Ke zmíněné knihovně ale musí být pochopitelně rovněž zajišťována technická podpora pro řešení případných potíží. Popsaný princip zjednodušení práce pomocí knihoven neplatí jen u natolik specifických záležitostí jako je již zmiňovaná FFT, ale obecně u celého procesu programování. Ideou LabVIEW je maximálně skrýt komplexnější programátorské problémy (např. správa paměti, ukazatele na proměnné) a přiblížit se způsobů, jakým jsou technicky zaměřeni lidé zvyklí přemýšlet. Vývojový diagram je totiž zcela běžným nástrojem pro modelování procesů v reálném světě, proč ho tedy nevyužít přímo k návrhu kódu, který by daný proces měl například kontrolovat. [18]

Pochopitelně portfolio produktů NI s sebou nese i své nevýhody. Zjednodušení programování je vždy dosaženo za cenu snížené kontroly nad kódem, používá-li programátor nějakou komerční knihovni funkci (tedy ne open-source), nemá možnost nahlédnout dovnitř a zjistit nebo ovlivnit, jak pracuje. Pro speciální aplikace – například vysoké nároky na rychlost, množství zpracovaných dat to může být velmi limitující. I samotný princip grafického programování má svá úskalí a to především přehlednost složitějších projektů. Zatímco kód jednoduché aplikace je v grafické podobě velice názorný, s rostoucím rozsahem může přehlednost klesat k nule, pokud není formální úpravě věnována náležitá péče. Zpřehlednit textový kód odsazením a komentáři může být daleko jednodušší, než přerovnávání změní ikon a vodičů v obrovském diagramu, kterému nestačí ani rozlišení běžně dostupných monitorů. Tato nevýhoda je samozřejmě silně individuální a závislá na přístupu programátora. Zásadní nevýhodou pro všechny ale může být u všech produktů NI cena. Například LabVIEW Professional Development System pro Windows spolu s roční technickou podporou a aktualizacemi stojí od 150 tisíc Kč a výše. [19]

Samotné LabVIEW v základní verzi ovšem práci s obrazem nezvládne, je potřeba ho doplnit o Vision Acquisition Software, případně Vision Development Module.

## 5.2 Vision Acquisition Software

NI Vision Acquisition Software (VAS) je softwarový balík, který umožňuje základní operace s obrazem. Konkrétně jeho získání z kamery, zobrazení na monitoru a ukládání. Zároveň obsahuje ovladače pro řádově tisíce typů kamer. Přestože se jedná o produkt NI, lze ho použít nejen s LabVIEW, ale také jazyky C, C++, C#, Visual Basic a Visual Basic .NET. Licence VAS je už obsažena v ceně všeho HW od National Instruments, který je zaměřen na práci s obrazem – například Smart kamery, Compact Vision systémy nebo frame grabbery. Rovněž ho obsahují softwarové produkty pro další zpracování obrazu – Vision Development Module nebo Vision Builder for Automated Inspection. Za cenu přibližně od 13 000 Kč je možné jej zakoupit samostatně – například pro použití s HW od jiného výrobce, než NI. [20]

Software obsahuje dvě základní skupiny ovladačů, které pokrývají celé spektrum podporovaných kamer:

- NI-IMAQ – NI Smart kamery, analogové kamery, dále kamery s digitálním paralelním rozhraním, případně rozhraním Camera Link.
- NI-IMAQdx – GigE Vision a USB3 Vision kamery, IP kamery, rovněž podporuje rozhraní IEEE 1394 neboli FireWire a všechna zařízení kompatibilní se standardem DirectShow –

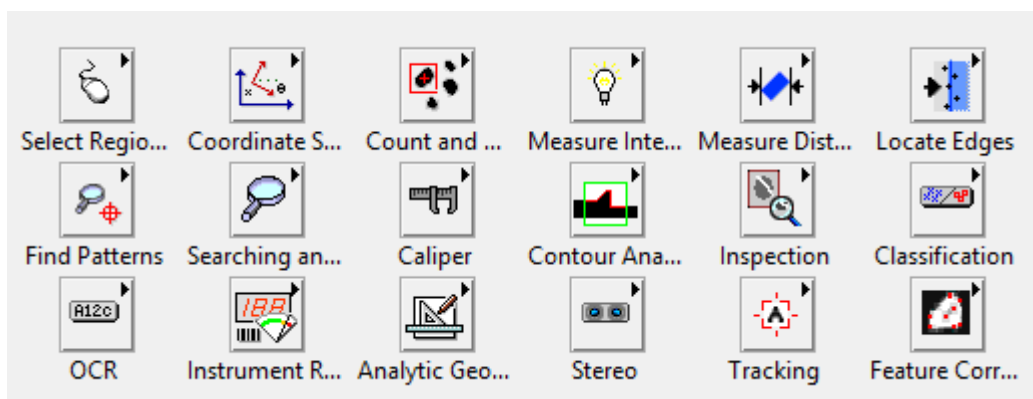
například web kamery, ať už externí s USB rozhraním nebo integrované v přenosných počítačích. [20]

### 5.3 Vision Development Module

Zatímco VAS představuje zjednodušeně řečeno soubor ovladačů ke kamerám umožňující pouze sejmout obrázek, Vision Development Module (dále zkráceně VDM) od NI nabízí řadu možností jeho zpracování. Následuje výčet některých z nich:

- Filtry, FFT, prahování
- Volba ROI (region of interest – oblast zájmu), nastavení koordinačního systému
- Hledání hran, měření vzdáleností
- Rozpoznávání vzorů (pattern matching)
- Čtení znaků, čárových kódů, údajů na ručkových i digitálních přístrojích

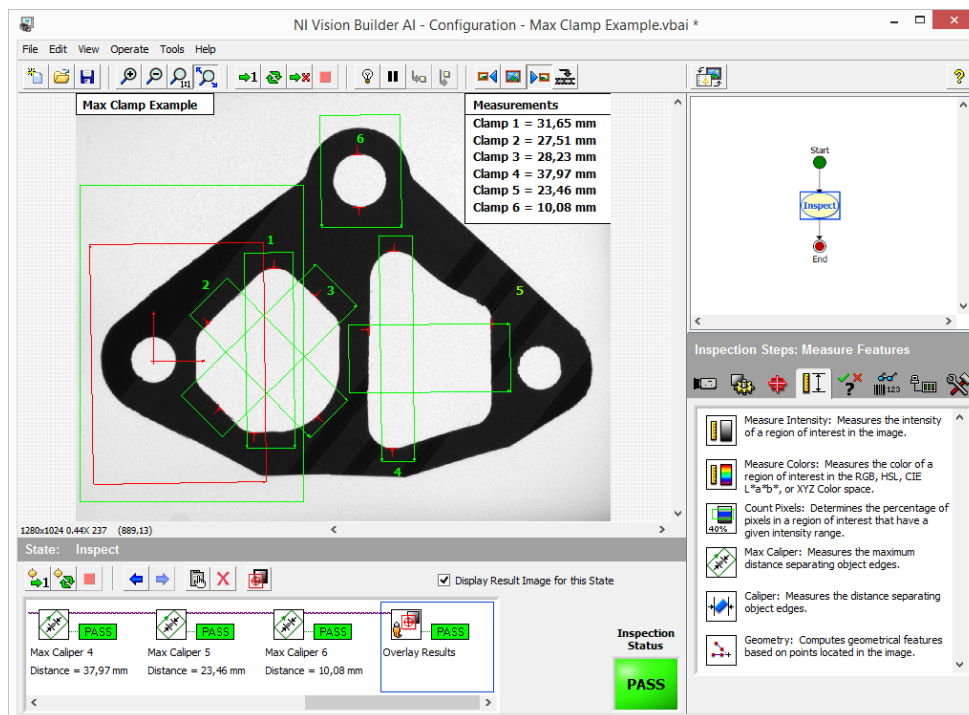
Součástí VDM je rovněž Vision Assistant, což je v podstatě samostatný program, který umožňuje rychlý návrh algoritmů pro zpracování obrazu, ze kterého následně lze vygenerovat kód pro LabVIEW, C, C++, C#, atd. Stejně jako celý VDM a VAS tedy není omezen jen na programovací jazyk od NI.



Obr. 9 Paleta funkcí "Machine Vision" z VDM pro LabVIEW

### 5.4 Vision Builder for Automated Inspection

Vision Builder for Automated Inspection (VBAI) nelze považovat za vývojové prostředí s programovacím jazykem v pravém slova smyslu. Spíše se jedná o software pro vytváření jednoduchých sekvenčních testů, které obsahují jednotlivé kroky pracující nějakým způsobem s obrazem. Prostedí Vision Builder slouží ke konfiguraci těchto kroků a vytváření posloupností, ve kterých se budou vykonávat. Součástí jsou ovšem i nástroje pro větvení či opakování testu, tedy ekvivalenty podmínek a cyklů z programovacích jazyků. Vision Builder obsahuje řadu operací používaných ve strojovém vidění jako je prahování, hledání vzoru nebo měření rozměrů v podobě již zmíněných kroků, které lze přizpůsobit konkrétní aplikaci. Prostedí Vision Builder může běžet ve dvou módech, první je „configuration“, tedy konfigurační a druhým „inspection“ – inspekční. Z názvu vyplývá, že v prvním se provádí konfigurace kroků testů a ve druhém pak test (kamerová inspekce) běží. Na PC zákazníka se instaluje pouze runtime verze obsahující jen inspekční mód bez možnosti testy měnit. Kromě práce s obrazem nabízí Vision Builder i podporu komunikace, například přes sériové rozhraní nebo Ethernet, aby bylo možné integrovat inspekci například do výrobní linky a nadřazenému systému signalizovat výsledky testů. Vision Builder může běžet na PC nebo také NI Smart kamerách, případně NI Compact Vision systémech.



Obr. 10 Prostředí Vision Builder for Automated Inspection

## 5.5 Jazyk C# a Microsoft Visual Studio

C#, respektive jeho varianta Visual C# je programovací jazyk vyvinutý společností Microsoft určený k tvorbě aplikací založených na frameworku .NET pro Windows. Jedná se o textový objektově orientovaný jazyk s podporou moderních programátorských principů jako je zapouzdření, dědičnosti nebo polymorfismus. Syntaxe vychází z jeho předchůdců C a C++, přičemž z druhého jmenovaného a také z jazyka JAVA přebírá základy objektového přístupu, které dále rozvíjí. [21] Oficiálním vývojovým prostředím je Microsoft Visual Studio, nicméně lze využít i jiné. Jazyk C# je široce používaný a .NET je do OS Windows už integrován, spuštění aplikace na cílovém PC nevyžaduje dodatečnou instalaci speciálního runtime engine. Tato výhoda se však projeví u distribuce aplikace mnoha uživatelům, nikoliv však u práce, jako je tato, kdy se předpokládá dodání plně nainstalovaného PC jako součásti kamerového systému. Vývojové prostředí je možné použít zdarma ve variantě Visual Studio Community i pro firemní/komerční účely za určitých podmínek zohledňujících velikosti společnosti a její roční výnos. Pokud nejsou splněny, je nutné zakoupit varianty Professional nebo Enterprise se startovní cenou 500 USD za samostatnou licenci. [22]

Pro práci s obrazem v C# respektive Visual Studio slouží například framework AForge.NET popsáný v následující kapitole nebo s využitím další softwarové vrstvy i OpenCV – kapitola 5.7. Nabízí se i kombinace s komerčními produkty National Instruments – kapitoly 5.2 a 5.3.

## 5.6 AForge.NET

AForge.NET je framework pro jazyk C# zaměřený na zpracování obrazu i další disciplíny. Nejedná se o komerční produkt, ale o svobodný software šířený pod licencí GPLv3 (General Public License). [23] Součástí jsou nejen knihovny s příslušnou dokumentací, ale také řada ukázkových aplikací stažitelných už ve formě spustitelného exe souboru. Lze si tedy udělat přibližnou představu o možnostech frameworku ještě před jeho instalací.

Přehled knihoven a možností využití: [24]

- **AForge.Imaging** – obsahuje řadu funkcí pro práci s obrazem, filtry, transformace, úpravu barev, převod mezi barevnými prostory, generátory šumu, textur, hledání hran atd. Tato knihovna je z uvedeného seznamu nejrozsáhlejší a tvoří základ celého frameworku.
- **AForge.Vision** – detekce a zpracování pohybu
- **AForge.Math** – matematický aparát používaný ostatními knihovnami je rovněž k dispozici uživatelům pro vývoj vlastních algoritmů
- **AForge.Video** – přístup k zařízením jako jsou IP kamery (JPEG nebo MJPEG stream) nebo USB či vestavěným webkamerám (standard DirectShow), možnost čtení a zápis videosouborů v kontejneru AVI. Zajímavá je kompatibilita se zařízením Kinect pro XBOX 360.
- **AForge.Robotics** – využití robotických stavebnic jako například LEGO Mindstorms
- **AForge.Neuro** – neuronové sítě a jejich algoritmy učení
- **AForge.Fuzzy** – fuzzy logika

AForge.NET neobsahuje sady ovladačů pro kamery (kromě široce rozšířených IP nebo web kamer), při použití zařízení z oblasti strojového vidění je potřeba zároveň nainstalovat buď SDK – software development kit výrobce kamery nebo některý univerzální balík ovladačů jako VAS.

## 5.7 OpenCV

OpenCV je další alternativa komerčních produktů z kategorie svobodného softwaru, šíření je možné pod licencí BSD - Berkeley Software Distribution [25] pro akademické i komerční účely. Knihovny projektu OpenCV jsou přímo použitelné s jazyky C,C++, Python a Java. Při použití wrapperu<sup>1</sup> Emgu CV se seznam rozšíří o rodinu .NET jazyků, tedy například C#. Z hlediska platforem je kompatibilní s Windows, Linux, Mac OS a Android. Zajímavou vlastností je podpora výpočtů na grafické kartě PC, což nabízí znatelné navýšení výkonu oproti využití pouze procesoru. Podmínkou je zařízení kompatibilní s CUDA – grafické karty firmy nVIDIA. [26] Možnosti ve zpracování obrazu jsou ještě o něco širší než u AForge, například rozpoznávání obličeje nebo sledování pohybu očí. Z hlediska funkčnosti se v podstatě ale jedná o rovnocenné alternativy, které je navíc možné použít společně. OpenCV má ale jasnou výhodu v tom, že je multiplatformní, kdežto AForge je omezen pouze na .NET a tedy OS Windows. OpenCV mimo jiné využívá i Google nebo Sony. [25]

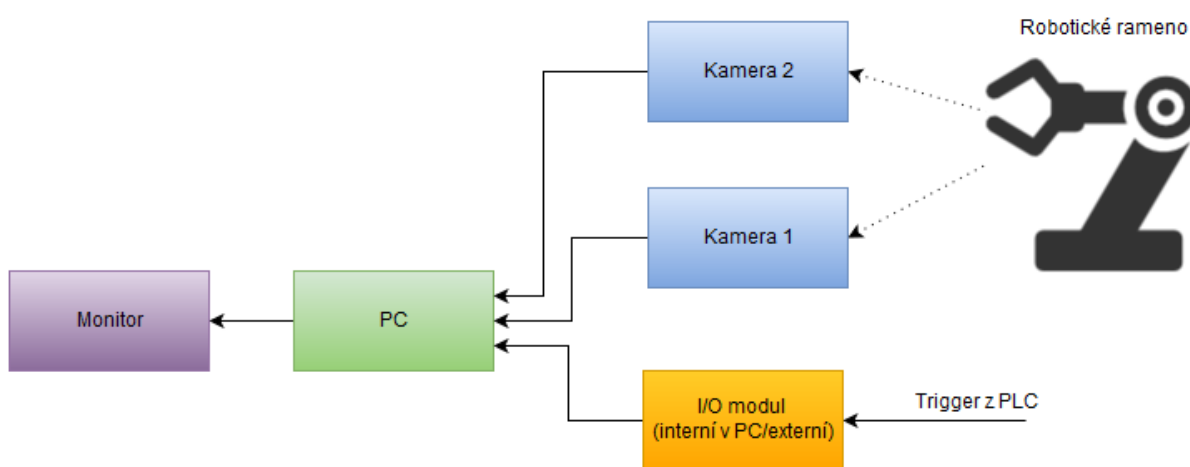
---

<sup>1</sup> V tomto kontextu význam slova koresponduje s jeho anglickým překladem, *wrap* – balit. Z hlediska programování se zjednodušeně řečeno skutečně jedná o „zabalení“, které umožňuje volat funkce knihovny i z jiného jazyka, než pro který je určena.

## 6 Výběr HW

### 6.1 Základní struktura

Vzhledem k požadavkům zákazníka je zřejmé, že systém musí obsahovat minimálně dvě kamery, PC a zařízení schopné detekovat triggorovací impuls z PLC. Takové zařízení může být v prvním případě externí, připojené skrze některé standardní rozhraní, například USB, RS232, Ethernet. Druhou variantou je použití interní zásuvné karty sběru dat do PCI/PCI express slotu. Rovněž u kamer je klíčová volba rozhraní – USB 3.0, Gigabit Ethernet nebo Camera link. Dále je potřeba vyřešit napájení kamer a případně i externího modulu pro detekci triggorovacího impulsu. Schéma zachycuje základní ideu, jak by systém mohl být realizován, aniž by řešilo do detailu například právě napájení jednotlivých komponent nebo typ komunikačních rozhraní.



Obr. 11 Ideové schéma hardwaru kamerového systému

### 6.2 Kamery

#### 6.2.1 Výběr rozhraní

Rozhraní Camera Link je vhodné zejména pro speciální aplikace, kde je prioritou vysoká přenosová rychlost, nevyskytuje se běžně v PC a bylo by nutné pro něj zakoupit rozšiřující kartu, jejíž cena například u firmy National Instruments přesahuje 30 tisíc Kč. [14] Proto byly dále zvažovány pouze rozhraní USB 3.0 a Gigabit Ethernet. Nejjednodušší a nejlevnější řešení představuje právě USB 3.0, protože přítomnost alespoň dvou portů je u dnes prodávaných PC standard. Bez použití dalších zařízení je vyřešeno i napájení kamery, protože samotná sběrnice USB 3.0 poskytuje stejnosměrné napětí 5 V a proud až 1 A. Nevýhodou USB je ale značně omezená délka kabeláže. Výsledkem vyměření možných kabelových tras u zákazníka byl poznatek, že minimálně mezi jednou kamerou a PC bude vzdálenost delší, než 10 m, což už nelze pomocí USB realizovat bez použití speciálních komponent (například prodlužovacích kabelů s opakovačem). Po zvážení možností všech tří dostupných rozhraní byl vybrán gigabitový Ethernet.



### 6.2.2 Obrazové parametry

V oblasti strojového vidění a nasazení kamer v průmyslu se obvykle volí spíše monochromatické kamery, pokud samozřejmě předmětem zkoumání není právě barva výrobku. V případě dohledového systému pro průmyslového robota byl ale ze strany zákazníka požadavek na použití kamer barevných. A to i přes skutečnost, že výlisky jsou vyrobeny z plastu černé barvy, takže přínos informace o barvě je diskutabilní. S ohledem na tento fakt byl proveden kamerový test přímo v předpokládaném místě nasazení kamerového systému a zákazníkovi byly poskytnuty ukázkové černobílé snímky k posouzení. Nicméně k přehodnocení požadavku na barevné snímky nedošlo, proto byl další výběr směřován výhradně na sortiment barevných kamer.

Dle původní specifikace se předpokládalo použití kamer s rozlišením 1920 x 1080 px, které se rovněž označuje jako Full HD. V době návrhu HW pro tuto diplomovou práci nebyly ale na trhu k dispozici kamery pro strojové vidění, které by splňovaly jednak požadavek na rozlišení, nabídly snímkový kmitočet 60 fps a odpovídaly cenovým možnostem. Vhodným kompromisem je kamera firmy Basler acA 1300-60gc, která sice disponuje menším, než požadovaným rozlišením, nicméně stále dostačujícím pro danou aplikaci.

### 6.2.3 Basler acA 1300-60gc

Kompaktní kamera z řady Basler ace o rozměrech pouze 42.0 x 29.0 x 29.0 mm s rozlišením 1280 x 1024 px a snímkovým kmitočtem 60 fps, splňuje mimo jiné standardy GigE Vision a GenICam. Shrnutí nejdůležitějších parametrů v Tab. 4. Téměř celou čelní stranu těla kamery tvoří standardní závit pro objektiv typu C mount. Na zadní straně pak najdeme konektor RJ-45 pro Ethernet a 6 pinový konektor Hirose pro připojení napájení a digitálních vstupů/výstupů. Montážní otvory se nachází na spodní straně kamery, konkrétně tři otvory se závitem M3 a čtyři se závitem M2. Podstatnou výhodou představuje možnost napájení pomocí technologie PoE, takže ke kameře může vést pouze jeden kabel – Ethernet, který kromě komunikace slouží i jako napájecí.

Tab. 4 Parametry kamery Basler acA 1300-60gc [27]

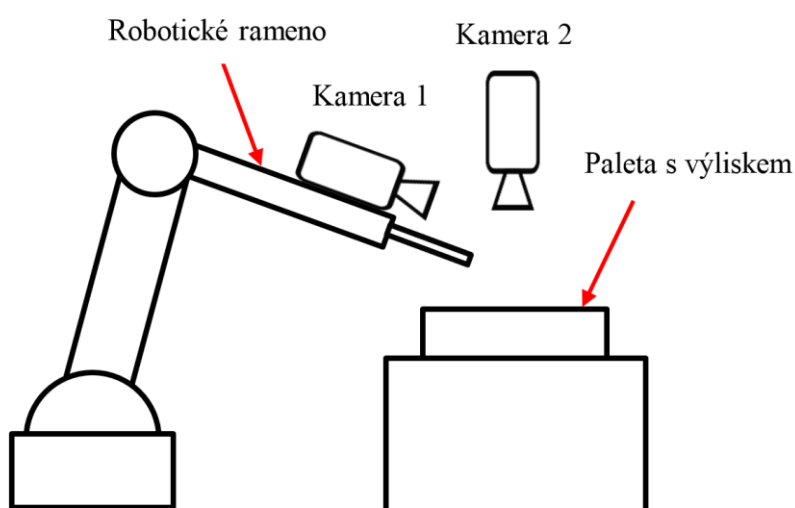
|                                |                        |
|--------------------------------|------------------------|
| <b>Rozlišení</b>               | 1280 x 1024 px barevně |
| <b>Snímkový kmitočet</b>       | 60 fps                 |
| <b>Rozhraní</b>                | Gigabit Ethernet       |
| <b>Závěrka</b>                 | Globální               |
| <b>Bitové rozlišení pixelu</b> | 12 bit                 |
| <b>Typ snímače</b>             | CMOS                   |
| <b>Velikost snímače</b>        | 6,80 mm x 5,40 mm      |
| <b>Napájení</b>                | 12 V DC nebo PoE       |
| <b>Spotřeba</b>                | 2,0 W (2,4 W s PoE)    |



Obr. 12 Kamera Basler acA 1300-60gc

### 6.2.4 Umístění

Kamery jsou umístěny následovně – jedna přímo na robotickém rameni, která zabírá detail dávkovací jehly a drážky s lepidlem. Druhá pak nad plastovým výliskem, takže poskytuje obraz o celé situaci a aktuální pozici ramene vůči výlisku. Schematicky je umístění kamer znázorněno na Obr. 13.



Obr. 13 Umístění kamer

## 6.3 Triggrovací I/O modul

Účelem I/O modulu v kamerovém systému je monitorovat triggrovací výstup z PLC a detekovat napěťový impuls signalizující start pohybu robota. K tomu je potřeba logický vstup s napěťovým rozsahem alespoň 0 V – 24 V DC. Dle specifikace zákazníka trvá vysoká logická úroveň impulsu 1 s a nejsou kladeny vysoké nároky na přesnost časování. Ke zjištění takovéto události dostačuje frekvence měření v řádu jednotek vzorků za sekundu, přesněji řečeno, minimálně dva vzorky za sekundu.

Kamerový systém je od počátku z finančních důvodů zamýšlen jako pasivní, nepředpokládá se tedy, že by poskytoval nějakou zpětnou vazbu řídicímu PLC robota. Do budoucna je ale vhodné, aby použitý hardware byl připraven i na případné rozšíření tímto směrem. Pokud by to neznamenalo podstatné

navýšení nákladů, měl by obsahovat i jeden nebo více logických výstupů rovněž kompatibilních s napěťovými úrovněmi používanými u PLC, tedy 0 V - 24 V.

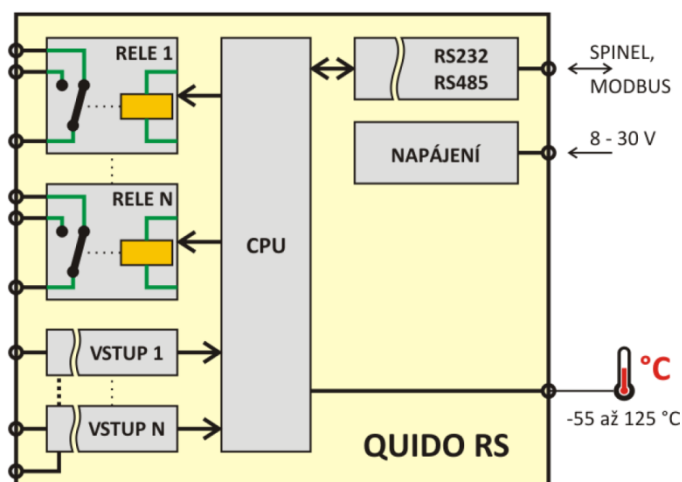
Výše uvedeným požadavkům odpovídají vstupně-výstupní moduly Quido české firmy Papouch s.r.o. Na výběr je z celé řady pokrývající širokou škálu počtu vstupů/výstupů. Většina modulů se vyrábí ve třech variantách komunikačního rozhraní: RS232 + RS485, USB 2.0 nebo Ethernet.

V úvahu přichází moduly v konfiguraci 2/2 nebo 4/4, vzhledem k minimálnímu cenovému rozdílu byla vybrána 4/4. Rozhraní USB 2.0 není příliš vhodné pro nasazení v průmyslových podmínkách, proto jsou vhodnými kandidáty RS232/RS485 nebo Ethernet. Požadavkům kamerového systému vyhovuje i jednodušší a levnější RS232, takže výsledkem výběru je modul Quido RS 4/4. Všechny Quidy se dodávají standardně s napěťovým rozsahem vstupů 0 V až 28 V, což se hodí pro použití spolu s PLC, nicméně lze objednat i provedení s rozsahem 0 V až 10 V.



**Obr. 14 I/O modul Quido RS 4/4**

Modul Quido RS 4/4 tvoří deska plošných spojů o velikosti přibližně 97 x 87 mm, podél dvou stran jsou umístěny pružinové svorkovnice WAGO, které slouží k připojení vstupních i výstupních signálů, napájení i komunikačních rozhraní RS232 a RS485. Modul není opatřen standardním konektorem Canon DB9. Obrázek zachycuje modul umístěný v kovovém krytu bez horního víka. Kryt lze volitelně doplnit o držák na DIN lištu. Funkci modulu objasňuje blokové schéma na Obr. 15, řídicím členem je mikrokontrolér, který na základě příkazů přenášených z PC nebo jiného nadřazeného systému spíná výstupní relé, respektive posílá stavy jednotlivých vstupů.



Obr. 15 Blokové schéma Quido RS [28]

#### Základní vlastnosti Quido RS 4/4 [28]

- 4 galvanicky oddělené logické vstupy
- 4 výstupy s přepínacím relé
- Napájení 8 V až 30 V DC
- Komunikační protokol Spinel a Modbus RTU
- Vstup pro teploměr
- Možnost automatické reakce na změnu stavu vstupu/teploty

## 6.4 PC

Počítač kamerového systému bude umístěn v zázemí pracoviště robotického ramene, kde se nepředpokládají náročnější provozní podmínky – zvýšená teplota, prašnost atd. Rovněž celý systém není kritický pro provoz výrobní linky. Proto vzhledem k požadavku na minimalizaci ceny bylo vybráno obvyklé kancelářské provedení PC, nikoliv průmyslové, které se zpravidla při stejné HW konfiguraci cenově podstatně liší. Konkrétně se jedná o PC výrobce Dell model Optiplex 3020MT.

Tab. 5 Parametry PC Dell Optiplex 3020MT

|                          |  |
|--------------------------|--|
| <b>Procesor</b>          | Intel Core i5-4590 3,30 GHz (čtyřjádrový, Turbo Boost na 3,70 GHz) |
| <b>Operační paměť</b>    | 8 GB DDR3, 1600 MHz  |
| <b>Grafická karta</b>    | Integrovaná Intel HD 4600  |
| <b>Pevný disk</b>        | 1 TB 3,5“, SATAIII, 7200 otáček                                    |
| <b>Síťová karta</b>      | 100/1000 Mb  |
| <b>Optická mechanika</b> | DVD +/- RW   |
| <b>Operační systém</b>   | Windows 7 Professional/Windows 8.1 Pro                             |
| <b>Skříň (provedení)</b> | Mini Tower, 360 x 175 x 417 mm                                     |

Výběr klíčových parametrů PC byl proveden na základě předběžných testů se záznamem a zobrazením snímků ze dvou kamer na notebooku s dvoujádrovým procesorem Intel Core i5-3230M s 8 GB operační paměti. Systém Windows byl sice instalován na SSD disku, nicméně obrazové soubory se ukládaly rovněž na plotnový HDD. Výkon notebooku je pochopitelně podstatně nižší, než u stolní sestavy a nebyl schopen pracovat s požadovaným snímkovým kmitočtem 60 fps. Nicméně poskytl alespoň přibližnou

představu o potřebné HW konfiguraci PC. Procesor i5-4590 podle benchmarků dostupných na internetu poskytuje cca o 85 % větší výpočetní výkon oproti i5-3230M. [29]

## 6.5 Karta NI PCIe-8236

Rozšiřující karta firmy National Instruments do PCI express slotu obsahuje dva porty gigabitového Ethernetu - splňují standard maximálně IEEE 802.3ab (1000BASE-T). Zařízení je primárně určeno pro použití jako takzvaný frame grabber. Navíc nabízí i PoE schopné dodávat až cca 15 W na jeden port. Vybrané kamery Basler ace rovněž PoE podporují. Zatímco elektronika karty je napájena ze sběrnice PCIe, měniče pro PoE vyžadují kvůli vyšší spotřebě napájení zvlášť a to prostřednictvím čtyřpinového konektoru Molex, který se připojí na napájecí rozvod ze zdroje PC. Zde bylo potřeba zvážit celkovou energetickou náročnost, protože vytipované PC Dell obsahuje pouze 300 W zdroj, což je v podstatě jedna z nejnižších běžně používaných hodnot. Zatímco elektronika karty dle dokumentace představuje pouze malou spotřebu nepřesahující 7 W, u PoE je uváděna špičková spotřeba 6,3 A, trvale pak 4,1 A, což při 12 V představuje 76 W/49 W. [30] To už je pro 300 W zdroj znatelné navýšení. Nicméně lze předpokládat, že uvedené hodnoty platí pro maximální odběr zařízení napájených skrze ethernetový kabel, tedy 15 W na jeden port, což u vybraných kamer zdaleka nebude dosaženo. Spotřeba kamer udávaná výrobcem je pouze 2,4 W. (Tab. 4) Za těchto podmínek by použití PoE nemělo ve zvoleném PC představovat problém. Součástí balení NI PCIe-8236 je i licence softwaru Vision Acquisition Software.



Obr. 16 Karta NI PCIe-8236

Tab. 6 Parametry karty NI PCIe-8236 [30]

|                            |  |
|----------------------------|--|
| <b>Porty</b>               | 2 x RJ45   |
| <b>Standard Ethernetu</b>  | IEEE 802.3 10BASE-T, 100BASE-TX, 1000BASE-T  |
| <b>Power over Ethernet</b> | IEEE 802.3af, 15,4 W na port   |
| <b>PC rozhraní</b>         | PCI Express 1.1 x4   |
| <b>Napájení</b>            | + 12 V (Molex, pro PoE) ... 6,3 A špičkově / 4,1 A trvale<br>+12 V (PCIe) ... 0,34 A<br>+3,3 V (PCIe) ... 0,74 A |

## 6.6 Karta Moxa CP-102E

Základní deska PC Dell Optiplex 3020MT bohužel neobsahuje rozhraní RS232 potřebné pro připojení modulu Quido. Řešením je rozšiřující karta do PCI Express slotu, která toto rozhraní nabízí hned dvakrát v podobně obvyklého konektoru Canon DB9.

Tab. 7 Parametry karty Moxa CP-102E

|                             |   |
|-----------------------------|---|
| <b>Podporovaný standard</b> | RS-232                                  |
| <b>Počet portů</b>          | 2                                       |
| <b>Konektory</b>            | Canon DB9                               |
| <b>Datové linky</b>         | TxD, RxD, RTS, CTS, DTR, DSR, DCD, GND  |
| <b>PC rozhraní</b>          | PCI Express x1                          |
| <b>Baudrate</b>             | 50 Bd až 921 600 Bd                     |
| <b>Podpora OS</b>           | Windows 95 až 8.1, Linux, QNX6 a další. |



Obr. 17 Karta Moxa CP-102E

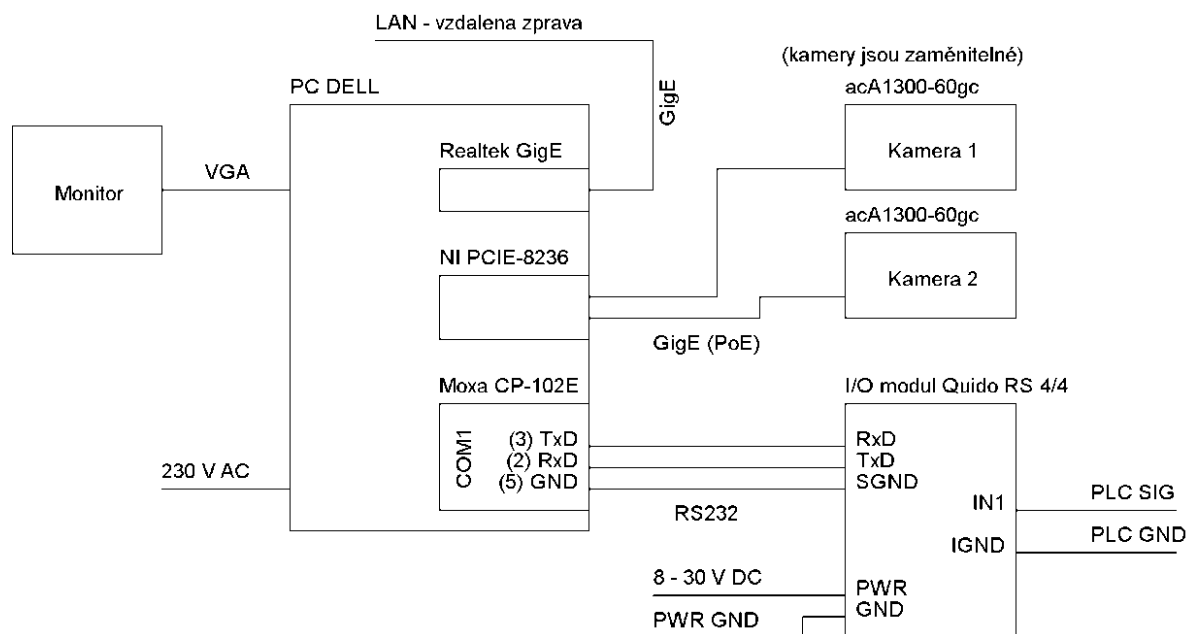
## 6.7 Objektivy

Pro výše popisované nasazení byly použity megapixelové objektivy Computar M0814-MP2 a M1614-MP2, jeden s ohniskovou vzdáleností 8 mm a druhý 16 mm. Oba objektivy mají manuální ovládání clony a zaostření s možností aretace, na kameru se montují pomocí závitů C-mount. Jsou určeny pro kamery s maximální velikostí snímače 2/3".

## 6.8 Schéma zapojení

Obr. 11 představuje zapojení celého kamerového systému z komponent vybraných v předchozích kapitolách. Použití karty NI PCIe-8236 s PoE znamená, že kamery nemusí být zapojeny skrze síťový switch, ani není potřeba vést zvlášť napájecí kabely. Datový kabel k modulu Quido stačí pouze tří vodičový, protože další datové linky rozhraní RS232 nejsou použity. Není zde řešen napájecí zdroj, protože pro modul lze využít stávajících rozvodů u zákazníka, které poskytují 24 V DC. Původní síťová karta Realtek, která byla v PC přítomna ještě před rozšířením o PCIe-8236, bude sloužit k připojení do technologické sítě. Účelem je především využití vzdálené správy. Protože kamery jsou shodného typu,

jejich záměna nezpůsobí závadu, ale pouze za předpokladu vhodné konfigurace IP adres. Vstupy i komunikační rozhraní modulu Quido jsou galvanicky odděleny pomocí optočlenů, nehrozí tedy vznik nežádoucích zemních smyček při připojení k zařízením, jejichž napájecí zdroj by byl na jiném potenciálu.



Obr. 18 Schéma zapojení kamerového systému

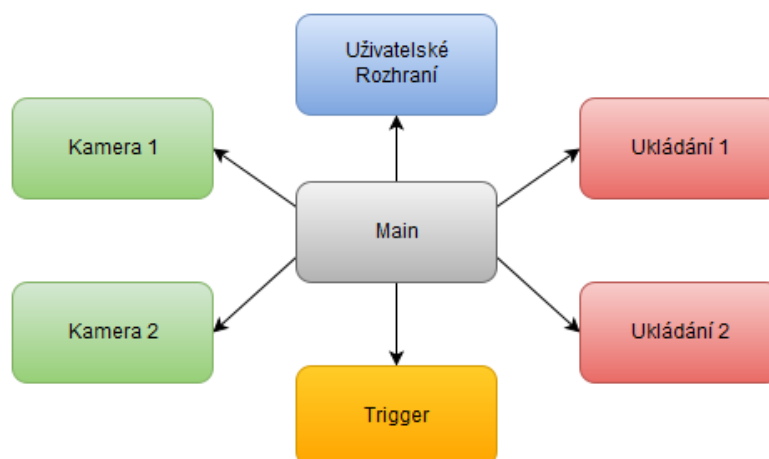
## 7 Software

### 7.1 Analýza požadavků

Z požadavků shrnutých v kapitole 2 na první pohled vyplývá, že software obsluhující kamerový systém by neměl být nikterak složitý. Při návrhu je ale potřeba zohlednit několik faktorů, které budou mít bezpochyby vliv na komplexnost řešení. Především požadavek na trvalý chod systému v třísměnném provozu. Je zřejmé, že například software provádějící nějaký výpočet nebo měření trvajících minuty může trpět více nedostatky, než aplikace, u které požadujeme bezproblémový trvalý chod v režimu 24/7. Zejména, pokud dlouhodobý provoz probíhá bez dohledu lidské obsluhy a není možné spoléhat na řešení problému v podobě restartu aplikace nebo celého systému. S požadavkem na trvalý chod systému souvisí i nakládání s velkým objemem získaných dat, především proto, že se jedná o obraz. Z hlediska možnosti dalšího vývoje a použití aplikace musí návrh i následná implementace proběhnout modulárně, aby bylo možné software snadno modifikovat pro potřeby jiných zákazníků nebo alespoň využít některou součást pro vývoj softwaru jiného.

### 7.2 Modularita

Rozdělení aplikace na moduly/komponenty vychází logicky z toho, jaké funkce má vykonávat. Jednotlivé moduly spolu komunikují a předávají si data pomocí zpráv, které vyvolávají vznik příslušných událostí. Celá aplikace je tedy založena na řízení toku programu událostmi (event-driven).



Obr. 19 Struktura softwaru - rozdělení na jednotlivé funkční moduly

#### Stručná charakteristika modulů:

- **Kamera 1, Kamera 2** – tyto dva moduly jsou identické, liší se pouze identifikátorem kamery, se kterou pracují. Zajišťují inicializaci spojení s kamerou, její nastavení, snímání obrazu a ukončení spojení.
- **Uživatelské rozhraní** – obsahuje indikátory a ovládací prvky pro lidskou obsluhu. Jediná komponenta aplikace, která je za chodu viditelná na monitoru.
- **Ukládání 1, Ukládání 2** – zajišťují uložení nasnímaného obrazu na pevný disk PC. S tím souvisí i kontrola volného místa.
- **Trigger** – spravuje spojení s HW modulem Quido RS 4/4, kontroluje stav triggerovacího vstupu a generuje události pro spuštění nahrávání.



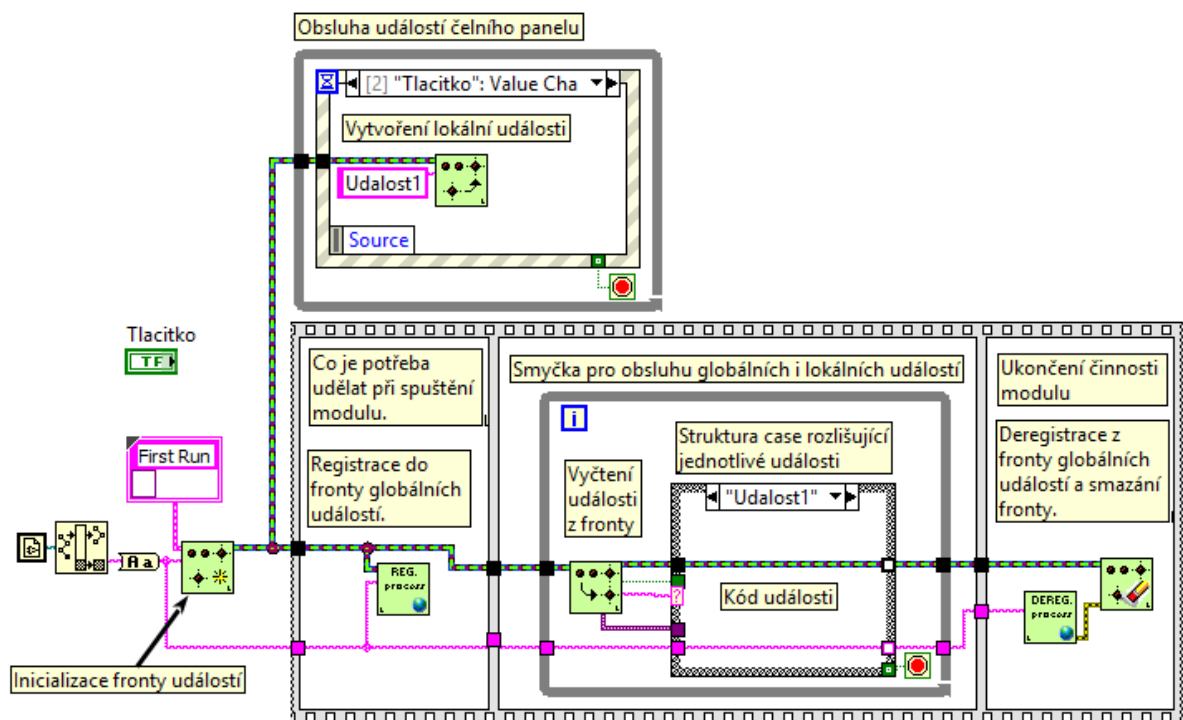
- **Main** – spouští a zastřešuje všechny ostatní komponenty, zajišťuje chod celé aplikace. Není tedy modulem v pravém slova smyslu, ale v hierarchii programu stojí nad nimi.

### 7.3 Šablona modulu

Všechny moduly (vyjma VI main) jsou naprogramovány s použitím stejného návrhového vzoru, který je vždy přizpůsoben konkrétnímu účelu. Každý modul musí být schopen zareagovat na tři typy událostí:

1. **Události vlastního čelního panelu** – v běžném režimu sice tento panel uživatel nevidí, ale při návrh a vývoji umožňuje ovládání modulu a zobrazení potřebných informací.
2. **Lokální události** – ta může být vytvořena na základě události čelního panelu, případně volána jinou lokální nebo globální událostí. Takto se dají vytvářet sekvence kroků.
3. **Globální události** – vyvolány z venku, tedy jiným modulem, případně z VI main.

Zjednodušený kód šablony modulu je na Obr. 20. Horní smyčka *while* se stará o obsluhu událostí čelního panelu pomocí *event structure*. Struktura *flat sequence* zajišťuje časovou posloupnost kroků, které je potřeba vykonat při spuštění modulu, v průběhu chodu – tedy obsluhu lokálních/globálních událostí a následně při ukončování modulu.



Obr. 20 Návrhový vzor modulu

Protože jednotlivé kroky ve skutečnosti obsahují nesrovnatelně více kódu, je pro úsporu velikosti diagramu *flat sequence* nahrazena strukturou *stack sequence*. Navíc pokud například kód vykonávaný při spuštění přesahuje rozumné rozměry a nevešel by se na obrazovku, je možné ho rozdělit do více *frames* (rámečků) a pomocí *stack sequence* přepínat. Zde byla *flat sequence* použita kvůli názornosti, aby všechny kroky byly viditelné. Identifikátorem každého modulu pro posílání globálních událostí je jeho název a to včetně přípony, takže například *trigger.vi*. Návrhový vzor včetně VI zajišťujících práci s událostmi (na Obr. 20 zelené ikony) byl přejat ze softwaru ATEster firmy ATEsystem s.r.o. s laskavým

dovolením Ing. Leoše Maršálka. Účelem je především dodržet jednotnou formu vyvíjeného softwaru a využít už ověřené kusy kódu, na jejichž funkčnost se lze spolehnout.

## 7.4 Volba způsobu ukládání obrazu

### 7.4.1 Struktura a formát

Před implementací softwaru bylo nutné rozhodnout, jakým způsobem se bude nasnímaný obraz kamer ukládat. Tato otázka má dvě podstatné roviny:

1. **Struktura** – kamery poskytují obraz v podobně jednotlivých snímků v nastaveném časovém intervalu. Uložit je ale lze buď samostatně, tedy každý snímek v podobě souboru/fotografie například JPG, nebo vždy určitou skupinu snímků do jednoho souboru ve formě videa. Druhá možnost se zdá vhodná, protože jedno video může obsahovat jeden pracovní cyklus robota, respektive obecně jednu událost po triggeru. Nicméně při posuzování dráhy ramene pravděpodobně obsluha video stejně zastaví a bude se snažit posouvat obraz po co nejmenších krocích, vracet se a podobně. Účel kamerového systému tedy hovoří spíš pro jednotlivé snímky, ale je potřeba zvážit i nevýhodu, což je celkově velký počet souborů. Například místo jednoho souboru 10 s videa s 30 fps vznikne 300 fotografií.
2. **Formát** – ať už pro video nebo fotografie existuje celá řada formátů, přičemž každý se pochopitelně hodí pro jinou informaci a vykazuje jiný poměr velikosti souboru a kvality obrazu.

Prvotní ideou bylo ukládání ve formě videa, proto bylo nutné prozkoumat v tomto ohledu možnosti LabVIEW. Ty bohužel nejsou příliš široké a omezují se pouze na práci s kontejnerem AVI a několika málo kodeky, které lze zjistit pomocí VI „IMAQ AVI2 Get Codec Names“ [31]. Výsledkem při standardní instalaci LabVIEW a Vision Acquisition Software na OS Windows 8 byl následující seznam:

- Kodek Intel IYUV
- Microsoft Video 1
- FF Video Codec 1 (FFV1) (NI Vision)
- Motion JPEG (NI Vision)
- Y800 Uncompressed Grey Scale (NI Vision)
- YUV 4:2:0 Planar (NI Vision)

LabVIEW při vytváření AVI souboru rovněž umožňuje nastavit kvalitu a to v rozsahu 0 až 1000. Pro srovnání kodeků byl vytvořen v LabVIEW program, který sejme z kamery zadaný počet snímků a uloží je pomocí všech dostupných kodeků s pěti různými nastaveními kvality. Pro účel testu byla použita černobílá kamera Basler acA1300-30gm, ukládalo se 300 snímků při kvalitách 200, 400, 600, 800 a 1000. Při dodatečném experimentu s barevnou kamerou se rovněž ověřilo, že ne všechny kodeky podporují barevný obraz – u Y800 Uncompressed Grey Scale je to zřejmé přímo z názvu. Výsledek srovnání se nachází v Tab. 8, ze které vyplývá, že nastavení kvality funguje pouze kodeků Microsoft Video 1 a Motion JPEG. Protože Intel IYUV produkuje příliš velké soubory a poslední dva kodeky není možné použít pro barevný záznam, lze dále zvažovat pouze Microsoft Video 1, FF Video Codec 1 a Motion JPEG.

Tab. 8 Srovnání kodeků

| Kodek                        | Kvalita   | Velikost souboru (MB) | Délka ukládání (s) | Podpora barev |
|------------------------------|-----------|-----------------------|--------------------|---------------|
| Kodek Intel IYUV             | Nemá vliv | 554                   | 23                 | ANO           |
| Microsoft Video 1            | 200       | 46                    | 19                 | ANO           |
|                              | 400       | 47                    | 19                 |               |
|                              | 600       | 51                    | 19                 |               |
|                              | 800       | 74                    | 20                 |               |
|                              | 1000      | 404                   | 37                 |               |
| FF Video Codec 1             | Nemá vliv | 174                   | 26                 | ANO           |
| Motion JPEG                  | 200       | 10                    | 18                 | ANO           |
|                              | 400       | 11                    | 18                 |               |
|                              | 600       | 12                    | 18                 |               |
|                              | 800       | 15                    | 18                 |               |
|                              | 1000      | 115                   | 25                 |               |
| Y800 Uncompressed Grey Scale | Nemá vliv | 370                   | 11                 | NE            |
| YUV 4:2:0 Planar             | Nemá vliv | 555                   | 13                 | NE            |

Protože ani jeden z možných kodeků nenabídl příliš vysokou kvalitu snímků a obrazové záznamy mají posloužit k analýze, pro další vývoj kamerového systému byl zvolen způsob ukládání ve formě jednotlivých snímků ve formátu JPG.

#### 7.4.2 Členění a označení záznamů

S tím ovšem souvisí nutnost navrhnout způsob, jak se vypořádat s velkým množstvím souborů aby uživatel byl schopen v rozumném čase najít přesně ten záznam, který potřebuje. Každý nový záznam (tj. určitý počet snímků), ať už vytvořený na základě SW nebo HW triggeru, případně ručním spuštěním nahrávání, je uložený v adresáři, jehož název je odvozen od aktuálního času a označení kamery. Veškeré záznamy jsou dále tříděny dle dnů, takže s každým dalším dnem nahrávání vzniká nový adresář, jehož název odpovídá datu (rok, měsíc, den).

#### Obecný tvar cesty:

E:\hlavní adresář záznamů\adresář dne\adresář konkrétního záznamu

#### Formát označení adresáře záznamu:

[Rok][měsíc] [den]\_[hodina][minuta][sekunda]\_[kamera]

#### Příklad s konkrétní cestou:

E:\Snimky\20150612\20150612\_140545\_cam1

Tento záznam byl tedy pořízen 12.6.2015 v 14:05:45 kamerou 1.

Veškeré záznamy jsou uloženy do hlavního adresáře, který je možno nastavit v konfiguračním souboru.

## 7.5 Hlavní úloha aplikace – main

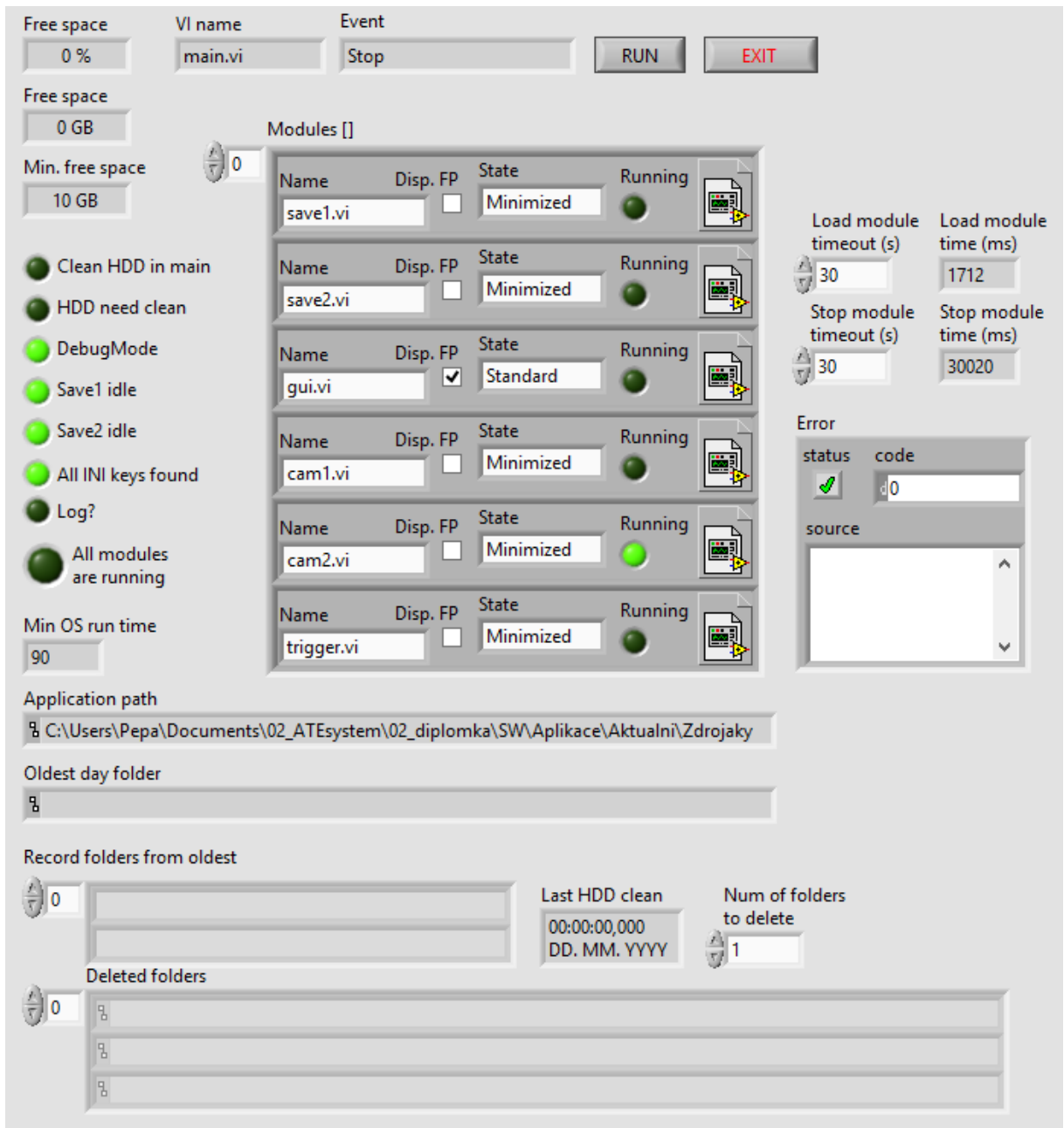
Jak již bylo zmíněno v kapitole 7.2, VI main je zodpovědné za chod celé aplikace, protože spouští a ukončuje jednotlivé moduly. Kromě toho provádí také čištění pevného disku při nedostatku volného místa. Z hlediska logického členění celé aplikace patří tato úloha ale spíše do modulu, který obstarává ukládání. Porušení této logiky má dva zásadní důvody, zaprvé mazání souborů je vhodné provádět přednostně v době, kdy pevný disk není zatěžován zároveň ukládáním. K tomu je potřeba mít k dispozici informaci o tom, že oba moduly pro ukládání jsou nečinné, což je výsadou právě VI main, který je oběma modulům nadřazen. Druhý důvod představuje čas, který operační systém potřebuje ke smazání požadovaných souborů. V závislosti na počtu souborů a stavu OS se může jednat i o desítky sekund. V takovém případě by modul pro ukládání dlouho zaneprázdněný mazáním nemohl zpracovat snímky nového záznamu. Oproti tomu úloha main je po spuštění modulů nevytížená a je tedy výhodné ji zaměstnat právě mazáním.

Aplikaci lze spustit ve dvou režimech – normální provozní nebo debug. V normálním režimu se zobrazí spouštěcí obrazovka (Obr. 30), což je malá část čelního panelu VI main, přičemž start aplikace nelze ovlivnit. Naopak v debug režimu se zobrazí kompletní čelní panel a ke spuštění modulů dojde až po stisknutí tlačítka *RUN*. Pro účely ladění je tedy možné například vynutit zobrazení i dalších čelních panelů modulů, které jsou jinak běžnému uživateli skryty, případně některý modul nespouštět vůbec.

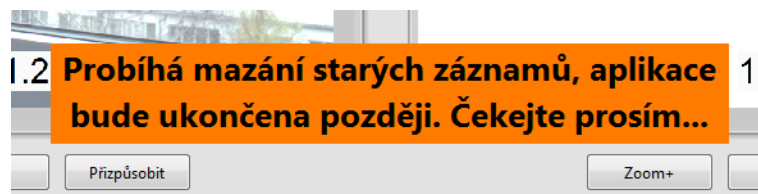
Spouštění i ukončování modulů je vždy časově omezeno, jinými slovy pokud se nepovede všechny moduly spustit například do 30 sekund (výchozí hodnota), vyhlásí aplikace chybu a bude ukončena. Spuštění i ukončení každý modul potvrzuje pomocí události *\_Module is running* po provedení kritických operací, jako je například inicializace nebo ukončení spojení s kamerou. Předjde se tak nepříjemným situacím, kdy pro uživatele aplikace „zamrzne“, protože neustále čeká na nějakou nefunkční komponentu. V případě, že některý modul nelze korektně ukončit, provede to modul main násilně pomocí *VI Quit LabVIEW*, čímž pochopitelně ukončí i sám sebe. Tímto je zaručeno, že aplikace uvolní všechny alokované systémové prostředky.

Pokud volné místo na HDD klesne pod určitou mez nastavenou v konfiguračním souboru, odešle modul pro ukládání událost *\_HDD maintain* - požadavek na údržbu disku, respektive mazání starých záznamů. Vzhledem k tomu, že denní adresáře jsou pojmenovány podle data (kapitola 7.4.2), lze jednoduše vybrat, který je nejstarší. Také názvy adresářů záznamů přímo obsahují datum a čas, vybere se tedy požadovaný počet nejstarších, které se smažou. Počet mazaných záznamů se rovněž volí v konfiguračním souboru aplikace. Tento poměrně jednoduchý systém má ovšem i svá úskalí:

1. **Přejmenování/přesunutí záznamů** - Aplikace má povoleno mazat záznamy pouze v nastaveném adresáři, pokud jsou ručně přesunuty jinam, zůstanou nedotčené, i když už na HDD nebude žádné volné místo a aplikace bude ukončena s chybou. Přejmenování adresářů může mít za následek smazání jiného, než nejstaršího záznamu.
2. **Ukončení aplikace** – uživatel se pokusí ukončit aplikaci v průběhu mazání záznamů, tato situace je ošetřena, zobrazí se hlášení (Obr. 22) a aplikace se ukončí až po smazání záznamů.
3. **Mazaný záznam je právě prohlížen** – v tomto případě aplikace vyhlásí chybu.



Obr. 21 Čelní panel mainu v režimu ladění



Obr. 22 Dokončení mazání starých záznamů před ukončením aplikace

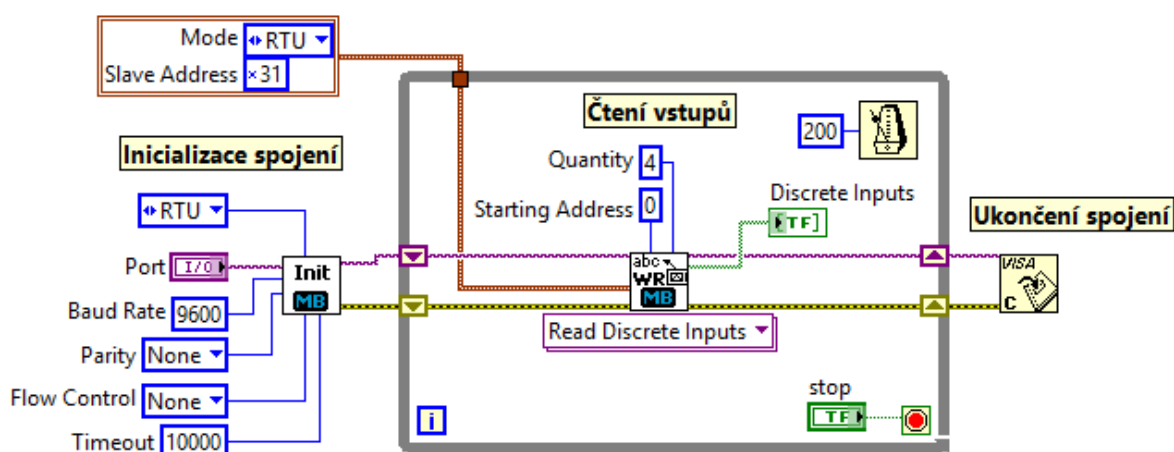
Tab. 9 Seznam globálních událostí odesílaných hlavní úlohou aplikace (main)

| Událost           | Příjemce       | Data         | Popis                                  |
|-------------------|----------------|--------------|--|
| _Default settings | gui.vi         | -            | Provedení výchozího nastavení aplikace |
| Stop              | všechny moduly | Název modulu | Pokyn k ukončení modulu                |
| _Clean HDD        | gui.vi         | Ano/Ne       | Informace o probíhajícím čištění disku |

## 7.6 SW modul trigger

Komunikace s HW modulem Quido RS4/4 probíhá pomocí protokolu Modbus RTU, který je realizován na sériovém rozhraní RS232. PC je v roli master, HW modul Quido funguje jako slave. Využití Modbusu je velmi výhodné, protože není nutné programovat pro Quido přístrojový driver. Knihovna pro LabVIEW obsahující Modbus je dostupná na webu National Instruments [32]. Součástí knihovny jsou VI pro všechny běžné operace podporované protokolem Modbus – čtení diskretních vstupů, zápis stavu cívek, čtení vstupních registrů atd.

Ke zjištění trigrovačního impulsu tedy stačí periodicky číst vstup pomocí polymorfního VI *MB Serial Master Query Read Discrete Inputs*. Značně zjednodušeně je princip komunikace s Quido naznačen v diagramu na Obr. 23. Ve skutečnosti je kód modulu pochopitelně daleko složitější a odpovídá šabloně z kapitoly 7.3. Především bylo nutné ošetřit chyby, které mohou během komunikace nastat.



Obr. 23 Princip komunikace pomocí protokolu Modbus v LabVIEW

Spuštění a ukončení nahrávání jednotlivých kamer zajišťují globální události *\_Start Record* a *\_Stop Record*, které jsou generovány na základě stavu časovače, jehož hodnota se porovnává s požadovaným zpožděním nahrávání a délkou záznamu. Časovač se spouští vždy při detekování trigrovačního impulsu, následně se jeho hodnota inkrementuje o uplynulý počet milisekund. Pokud hodnota přesáhne nastavené zpoždění kamery (například 10 s), generuje se událost *\_Start Record*, jakmile hodnota přesáhne součet zpoždění a délky nahrávky, generuje se *\_Stop Record*. Protože kamery jsou dvě, obecně jich ale může být i více, běží časovač, dokud nahrává alespoň jedna kamera. Po tuto dobu se nekontroluje stav vstupu modulu Quido a nelze tedy znovu trigrovat. Zakázaný je i softwarový trigger z uživatelského rozhraní.

Tab. 10 Seznam globálních událostí odesílaných modulem trigger

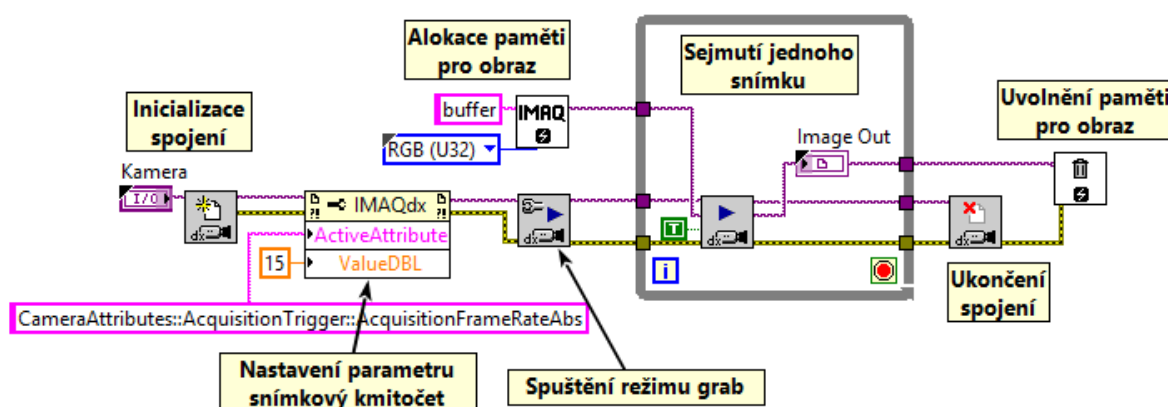
| Událost            | Příjemce        | Data         | Popis                             |
|--------------------|-----------------|--------------|-----------------------------------|
| Trigger            | gui.vi          | -            | Došlo k triggerovací události     |
| _Module is running | main.vi         | Název modulu | Modul byl korektně spuštěn a běží |
| _Start Record      | cam1.vi/cam2.vi | -            | Odstartování nahrávání            |
| _Stop Record       | cam1.vi/cam2.vi | -            | Ukončení nahrávání                |
| _Terminated        | main.vi         | Název modulu | Modul byl ukončen                 |

## 7.7 SW modul kamera

Modul obsluhující kameru rozšiřuje strukturu popsanou v kapitole 7.3, protože použitý systém obsluhy událostí se hodí pro děje opakující se řádově několikrát za sekundu. Požadovaný snímkový kmitočet kamer je ale až 60 fps, proto čtení obrazu z kamery zajišťuje samostatná smyčka, která není ovlivněna příchodem globálních/lokálních událostí.

Každý nový snímek z kamery je vložen do bufferu tvořeného polem konstantní velikosti, které se cyklicky přepisuje. Na prvky tohoto pole jsou pak další moduly odkazovány pomocí referencí (ekvivalent ukazatele/pointeru z jazyka C). Reference jsou spolu dalšími informacemi o snímku, jako například označení kamery a čas, posílány jako data připojená ke globálním událostem *\_Save* a *\_Display*. Příjemcem těchto událostí jsou moduly pro ukládání a GUI. Pokud je zapnuto nahrávání, jsou k uložení předány všechny snímky, v opačném případě se událost *\_Save* vůbec negeneruje. Pokud je aktivován živý náhled, mohou být k zobrazení předány pouze některé snímky, a to v závislosti na kmitočtu kamery. Pokud kamera snímá rychlostí do 25 fps, zobrazují se všechny snímky. Dále už se počet snímků redukuje, například při 26 až 30 fps se zobrazuje pouze 5/6 snímků, při 31 až 35 fps pouze 3/4 snímků atd. Tímto způsobem se snižuje zatížení PC, protože zobrazování více než 25 snímků za sekundu jako náhled pro lidské oko nemá příliš smysl.

Pro práci s kamerou je použit NI Vision Acquisition Software, který LabVIEW rozšiřuje o řadu ovladačů a SW komponent pro snímání, zobrazování a ukládání obrazu. Základní princip obsluhy kamery znázorňuje Obr. 24, který je v aplikaci kamerového systému rozšířen o nastavení dalších parametrů, ošetření chyb a již zmíněný buffer pro řadu snímků.



Obr. 24 Princip práce s kamerou pomocí Vision Acquisition Software

V aplikaci paralelně běží dva moduly kamery, protože dohledový systém obsahuje dvě fyzické kamery – každá je tedy naprosto nezávislá a může mít nastavený i jinou snímkovou frekvenci.

Kromě základních parametrů jako je právě počet snímků za sekundu nebo expoziční doba, umožňují kamery Basler ace několik desítek dalších nastavení, které je možné upravit například pomocí softwaru NI MAX. Změna některého z nich může způsobit nefunkčnost celého dohledového systému. I přes předpoklad, že k PC má přístup pouze proškolená obsluha a neměly by hrozit neobdobné zásahy do konfigurace, provádí se při startu aplikace nastavení všech parametrů kamery pomocí konfiguračního souboru (odlišný od konfiguračního souboru celé aplikace). A teprve pak se provede uživatelské nastavení snímkovací frekvence, expozice a citlivost. Jedná se o velmi robustní řešení, které umožňuje například při poruše snadnou výměnu kamery za stejný model. Není nutné ji předem konfigurovat. V případě, že by bylo potřeba ladit nastavení kamer nebo vyzkoušet dohledový systém s jiným modelem kamery (i od jiného výrobce) je možné automatickou konfiguraci vypnout. Zůstanou tedy platná nastavení provedená v NI MAX nebo softwaru výrobce.

Tab. 11 Globální události modulu kamera

| Událost            | Příjemce           | Data                           | Popis                               |
|--------------------|--------------------|--------------------------------|-------------------------------------|
| _Save              | save1.vi/ save2.vi | Snímek s časovou značkou       | Uložení snímku                      |
| _Display           | gui.vi             |                                | Zobrazení snímku v GUI              |
| _Save init         | save.vi            | Cesta k adresáři všech záznamů | Inicializace ukládání               |
| _Module is running | main.vi            | Název modulu                   | Modul byl korektně spuštěn a běží   |
| _New Folder        | save.vi            | Cesta k adresáři záznamu       | Vytvoření adresáře pro nový záznam  |
| _Start Record      | gui.vi             | Název kamery                   | Indikace, že kamera začala nahrávat |
| _Terminated        | main.vi            | Název modulu                   | Modul byl ukončen                   |

## 7.8 SW modul pro ukládání dat

Kód modulů pro ukládání (save1.vi/save2.vi) je oproti ostatním velmi jednoduchý, zajišťuje následující úkoly:

- **Vytváření nových adresářů** – jednak denní adresáře, kde jsou sdruženy všechny záznamy jednoho dne, a také adresář konkrétního záznamu po triggeru.
- **Ukládání snímků z kamery** – na základě reference v globální události převezme obrázek z bufferu a uloží ho jako jpg soubor na disk.
- **Kontrola volného místa na disku** – v původním konceptu zajišťoval i mazání starých záznamů, které ale bylo přesunuto do hlavní úlohy aplikace z důvodů popsaných v kapitole 7.5.

V aplikaci jsou souběžně spuštěny vždy dva moduly pro ukládání, přičemž obraz z kamery 1 ukládá save1.vi a analogicky obraz z kamery 2 zase save2.vi. Do kterého ukládacího modulu bude obraz z kamery směřovat lze nastavit v konfiguračním souboru aplikace, je možné i ukládat obraz z obou kamer v jednom modulu save.vi. Kód je totiž napsán univerzálně a není vázán na konkrétní kameru, jinak by ani modularita aplikace neměla smysl. Důvodem, proč běží více než jedno ukládání paralelně, je lepší využití HW prostředků PC. Takto konfigurovaná aplikace byla totiž schopna více vytížit procesor i disk počítače a uložit větší množství snímků za sekundu, než když ukládání zajišťuje jedno VI. I když je to programátorovi na první pohled skryto, funkce pro ukládání obrazu dostupné



v LabVIEW musí nejprve zajistit převod obrazu do požadovaného formátu, případně i kompresi, což vyžaduje určitý čas. Stejně tak uložení na disk je spjato s určitou režii a požadavky vzhledem k OS. Protože dnešní procesory v PC i LabVIEW podporují vlákna, je výhodné, aby se úkony v aplikaci paralelizovaly.

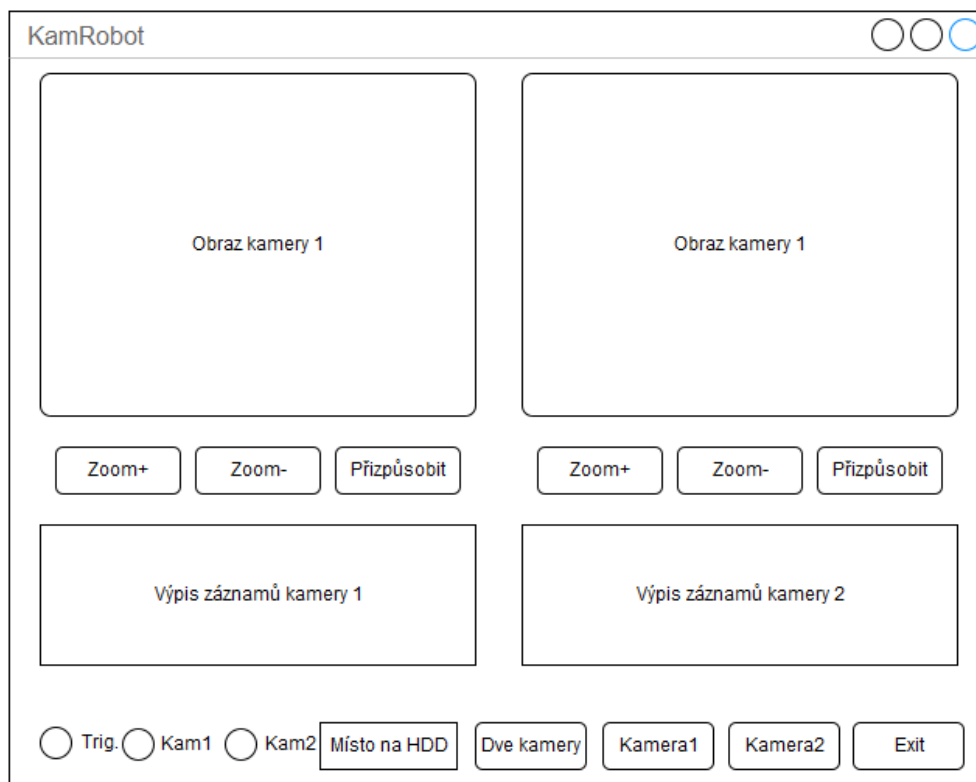
**Tab. 12 Globální události modulu pro ukládání**

| Událost            | Příjemce | Data                           | Popis   |
|--------------------|----------|--------------------------------|---|
| _HDD maintain      | main.vi  | Cesta k adresáři všech záznamů | Požadavek na čištění HDD                      |
| _State             | main.vi  | Ano/Ne                         | Informuje, zda je modul aktuálně nečinný      |
| _Module is running | main.vi  | Název modulu                   | Modul byl korektně spuštěn a běží             |
| _Folder created    | gui.vi   | Informace o adresáři           | Data pro seznam zobrazující poslední záznamy  |
| _HDD free space    | gui.vi   | Volné místo na HDD             | Data pro zobrazení volného místa na HDD v GUI |
| _Terminated        | main.vi  | Název modulu                   | Modul byl ukončen                             |

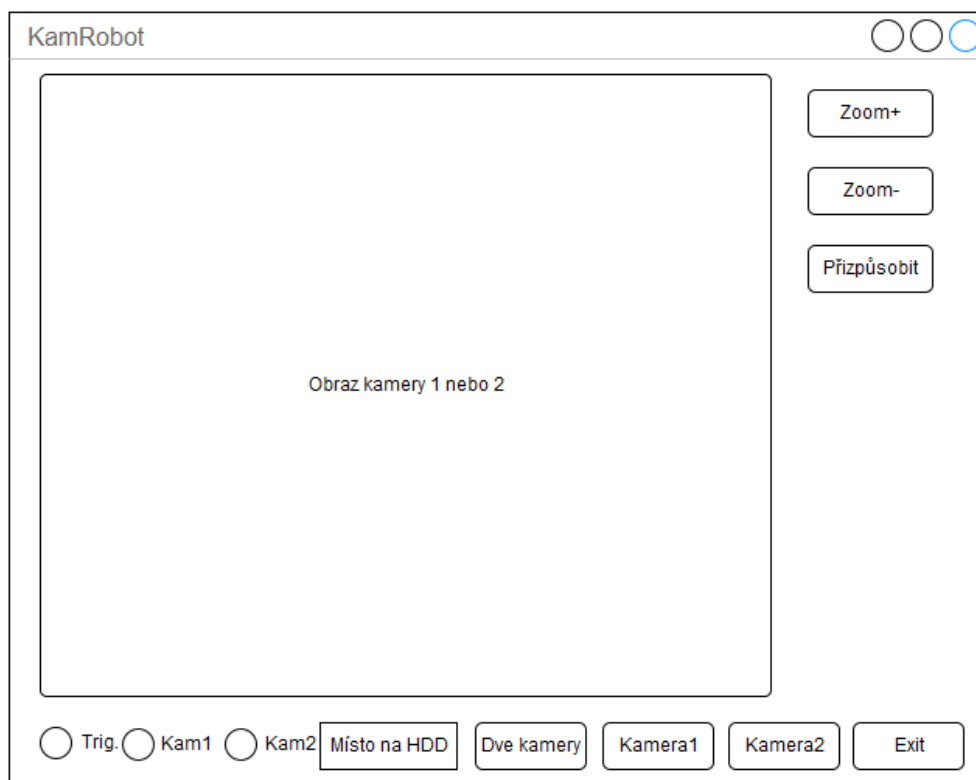
## 7.9 SW Modul uživatelského rozhraní

### 7.9.1 Wire-frames

Ještě před implementací ve vývojovém prostředí vznikl přibližný model (wire-frame) pro ujasnění představy o tom, jak by uživatelské prostředí mělo vypadat. Oproti požadavkům v kapitole 2 přibily boxy obsahující přehled posledních záznamů kamer. Obr. 25 představuje výchozí vzhled programu, ve kterém jsou viditelné obrazy z obou kamer. Tlačítka *Kamera 1* a *Kamera 2* přepínají uživatelské rozhraní na obraz pouze jedné kamery (Obr. 26). Tlačítko *Dvě kamery* obnoví výchozí zobrazení. Rozhraní neobsahuje dialogy pro nastavení aplikace – například čas záznamů kamer a podobně, protože je realizováno pomocí konfiguračního souboru. U aplikace se totiž nepředpokládá, že by se nastavení provádělo často, ale po odladění spíše výjimečně.



**Obr. 25 Model uživatelského rozhraní - zobrazení obou kamer**



**Obr. 26 Model uživatelského rozhraní - zobrazení jedné kamery**

## 7.9.2 Realizace v LabVIEW

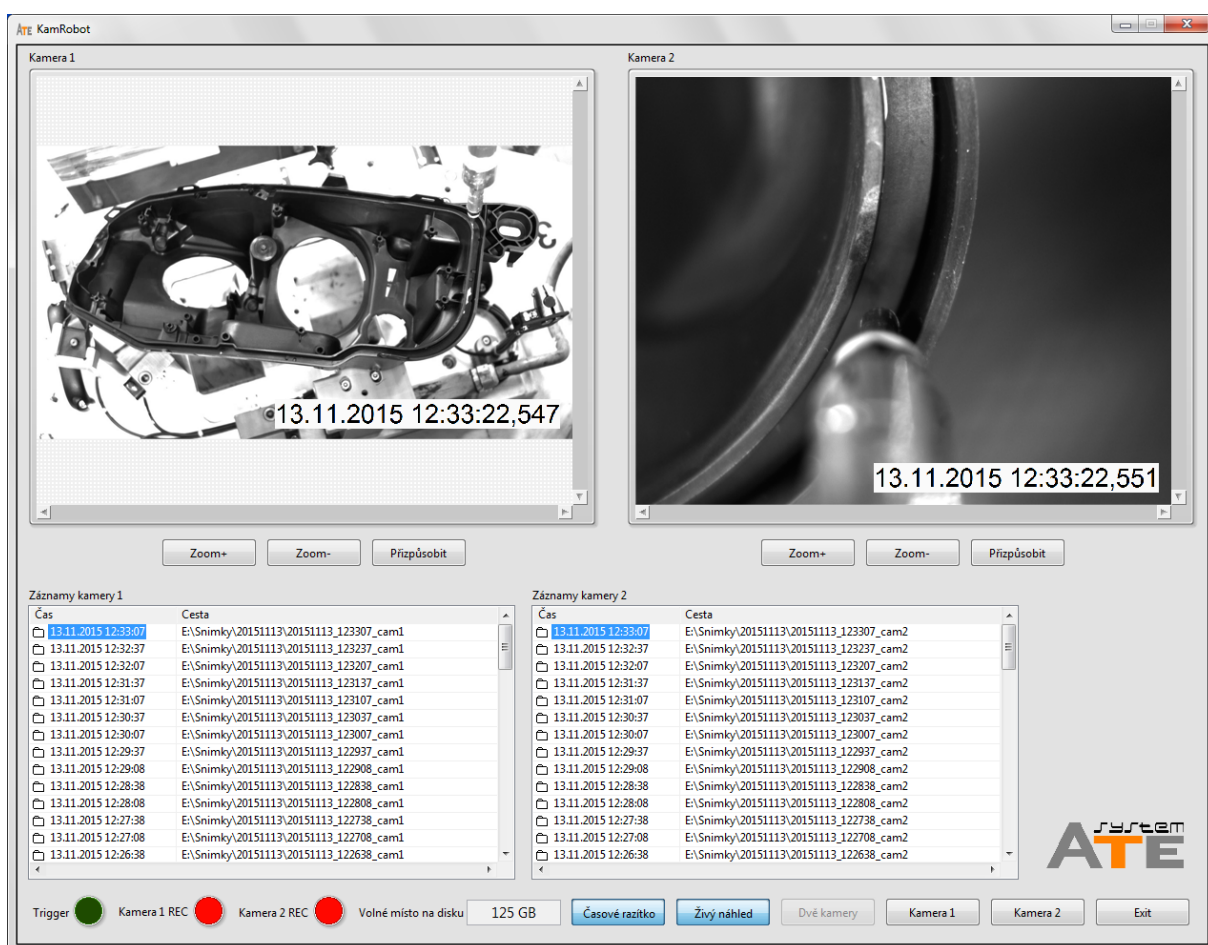
Na Obr. 27 je výsledná podoba uživatelského rozhraní se zobrazením obou kamer. Podrobný popis jednotlivých ovládacích prvků a indikátorů:

- **Zoom+:** přiblížení (zvětšení) obrazu příslušné kamery
- **Zoom-:** oddálení (zmenšení) obrazu příslušné kamery
- **Přizpůsobit:** upravení velikost obrazu tak, aby odpovídal rozměrům indikátoru
- **Záznamy kamery 1/Záznamy kamery 2:** Seznam posledních 200 záznamů kamery s uvedením data a času začátku záznamu a plné cesty k adresáři, kde jsou snímky uloženy. Dvojklikem na požadovaný záznam lze otevřít Průzkumníka Windows v příslušném adresáři.
- **Trigger:** Rozsvícení světla zelenou barvou indikuje, že byl zaregistrován trigger a aplikace čeká na spuštění nahrávání. Při zahájení záznamu alespoň jedné kamery trigger zhasne.
- **Kamera 1 REC/Kamera 2 REC:** Červená barva indikátoru signalizuje probíhající záznam.
- **Volné místo na disku:** Informace o zbývajícím volném prostoru pro záznamy na HDD v gigabajtech.
- **Časové razítko:** Umožňuje zapnutí respektive vypnutí časového razítka v obrazu kamery. Pokud je časové razítko zapnuto, bude přítomno i v uložených snímcích, nikoliv pouze v okně aplikace.
- **Živý náhled:** Aktivace/deaktivace zobrazení aktuálního obrazu v okně aplikace. V případě vypnutí živého náhledu běží snímání obrazu z kamery i záznam dále, pouze nejsou obcerstovány indikátory *Kamera 1* a *Kamera 2*. Zůstanou zobrazeny poslední snímky zachycené před vypnutím funkce. Deaktivace živého náhledu sníží zatížení PC a je vhodná, pokud dlouhodobě nebude běh aplikace sledován lidskou obsluhou. Pomocí konfiguračního souboru je možné nastavit automatické vypínání živého náhledu v případě nečinnosti uživatele. Dále i dobu, po jejímž uplynutí od posledního zásahu obsluhy k vypnutí náhledu dojde. Zásahem je myšleno kliknutí levým tlačítkem myši na některý z ovládacích prvků nebo i libovolně mimo ně v uživatelském rozhraní. Důvody přidání této funkce jsou vysvětleny v 8.3.5
- **Dvě kamery:** Zobrazení obrazu obou kamer.
- **Kamera 1/Kamera 2:** Zobrazení pouze jedné kamery.
- **Exit:** Ukončení aplikace.

Z hlediska členění aplikace je uživatelské rozhraní rovněž modul založený na stejné šabloně jako kamera nebo trigger a komunikuje s ostatními prostřednictvím globálního události. Zásadně se liší tím, že čelní panel je při běžném chodu zobrazen, zatímco u ostatních modulů skryt.

Tab. 13 Globální události odesílané uživatelským rozhraním

| Událost            | Příjemce        | Data         | Popis  |
|--------------------|-----------------|--------------|--|
| _Module is running | main.vi         | Název modulu | Modul byl korektně spuštěn a běží                          |
| Exit               | main.vi         | -            | Ukončení aplikace  |
| _View              | cam1.vi/cam2.vi | Ano/Ne       | Nastavení kamery, zda má odesílat snímky k zobrazení v GUI |
| _SW trigger        | trigger.vi      | -            | Ručně vytvořená triggrovací událost                        |
| Terminated         | main.vi         | Název modulu | Modul byl ukončen  |

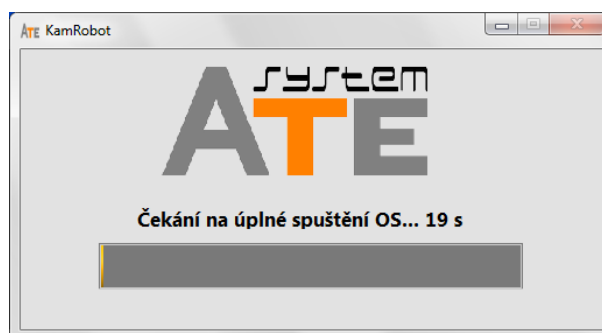


Obr. 27 Uživatelské rozhraní - zobrazení obou kamer

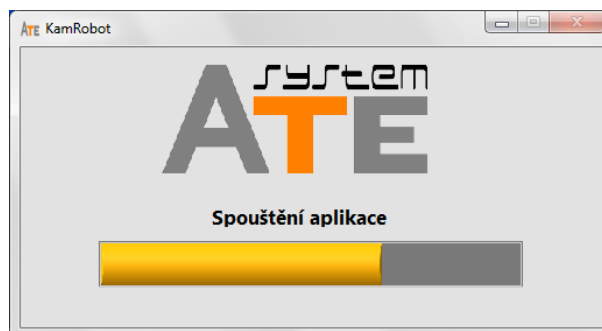
V režimu Kamera 1 zabírá většinu plochy uživatelského rozhraní snímáný obraz, nezobrazují se výpisy posledních záznamů – Obr. 28. Režim Kamera 2 vypadá shodně, takže obrázek není uveden. Protože se aplikace spouští automaticky při startu PC, je nutné počkat na úplné rozběhnutí operačního systému a načtení ovladačů. Po tuto dobu vidí uživatel dialog na Obr. 29, který ho informuje o zbývajícím čase do startu aplikace. Následuje spuštění jednotlivých modulů aplikace - Obr. 30.



Obr. 28 Uživatelské rozhraní - zobrazení jedné kamery



Obr. 29 Čekání na úplné rozběhnutí OS



Obr. 30 Spouštění aplikace

## 8 Testování a ladění

### 8.1 Postup a podmínky

V první fázi byly odstraněny chyby a nedostatky, které se objevily téměř okamžitě po vyzkoušení všech funkcí aplikace a různých nastavení. Bohužel zůstaly dva zásadní problémy, které se projeví až při provozu aplikace v řádu dní. Z toho vyplývá i velká časová náročnost ladění, protože zejména v případě chyby popsané v kapitole 8.3 bylo po úpravě kódu a vygenerování nového exe souboru potřeba počkat i několik dní, než došlo k novému pádu aplikace a celý postup se opakoval.

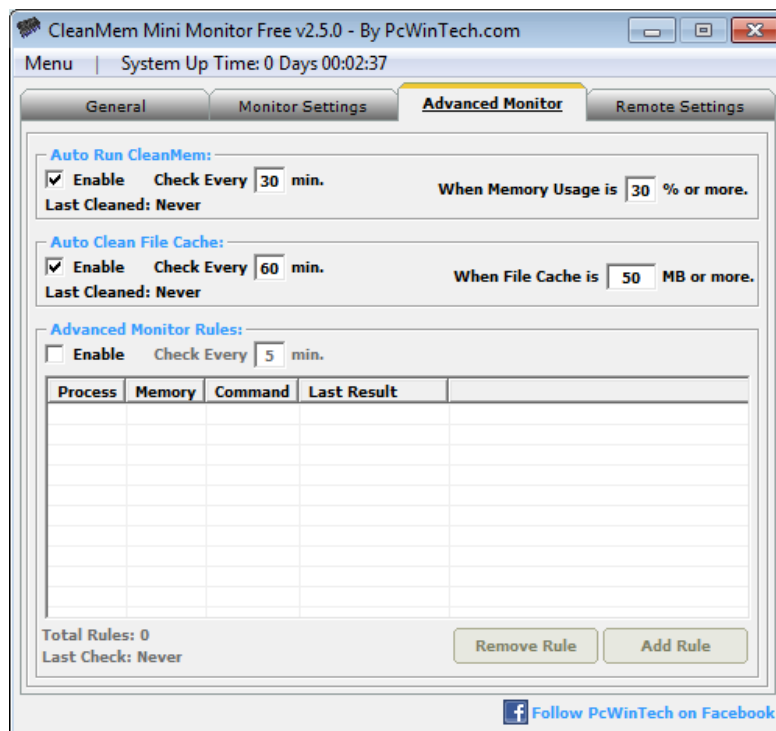
Pro testování bylo potřeba dohledový systém trvale zatěžovat. Kamery byly nastaveny na maximum, tedy 60 snímků za sekundu, záznam se prováděl souběžně z obou naráz po dobu 15 s, následovala 45 s pauza, živý náhled byl zapnut. (podmínky později upraveny – kapitola 8.2) Takto se cyklus triggování opakoval. Pro generování napěťového impulzu délky 1 s opakovaného v periodě jedné minuty sloužila vývojová platforma Arduino založená na mikrokontroléru ATmega. Přizpůsobení napěťových úrovní z 5 V TTL na 24 V zajistil tranzistorový převodník zapojený na kontaktním poli.

### 8.2 Obsazení operační paměti

Prvním problémem bylo pozvolné narůstání hodnoty obsazené operační paměti PC, které bylo znatelné už po 24 hodinách provozu. Po přibližně 3 až 4 dnech byla RAM plná téměř z 80 %, což už znatelně ovlivnilo provoz PC, jeho odezvu při prohlížení snímků atd. Z údajů ve Správci úloh OS Windows nicméně vyplývalo, že proces aplikace kamerového systému si po celou dobu udržuje téměř totožnou hodnotu alokované paměti. Tento fakt vylučoval nevhodný programátorský přístup při práci s proměnnými respektive paměti – například neustálé alokování nových prostředků aniž by byly uvolněny staré a podobně. Zjevně se tedy jednalo o problém na straně OS Windows 7 Professional, který se zřejmě nezvládl vypořádat s množstvím ukládaných souborů nebo byl špatně nastaven. Aplikace CleanMem [33] odhalila, že neustále narůstá tzv. file cache, který OS používá při čtení a zápisu na disk [34], a to až na hodnoty v řádu gigabajtů. Tento problém se projevoval i přes fakt, že systém stihl prokazatelně uložit veškeré soubory na disk, protože záznamy byly kompletní. Vysvětlením tedy nemohlo být, že OS není schopen dané množství souborů za jednotku času zvládnout, proto zůstávají ve filecache, který časem logicky narůstá. Navíc na přání zákazníka byly testovací podmínky změněny na 15 snímků za sekundu, doba záznamu 25 s, pauza pouze 5 s, zapnutý živý náhled. Celkový počet uložených snímků z obou kamer za minutu byl tedy 750. Oproti předchozímu nastavení, kdy systém musel během minuty uložit 1800 snímků, byly nároky méně než poloviční, přesto problém přetrvával.

Protože se nepodařilo zjistit, jaké nastavení OS by mohlo problém vyřešit, byly využity možnosti aplikace CleanMem (Obr. 31), konkrétně nastavení periodické kontroly filecache a následného vyčištění, pokud velikost přesahuje zvolenou mez. Díky tomu zůstalo obsazení RAM na přibližně stejné úrovni i po delším provozu.

Zajímavý byl výsledek testu aplikace na jiném PC, na kterém byl ovšem OS Windows 8.1 Pro, k narůstání filecache nedocházelo vůbec a obsazení operační paměti tedy zůstávalo konstantní i bez použití utility CleanMem.

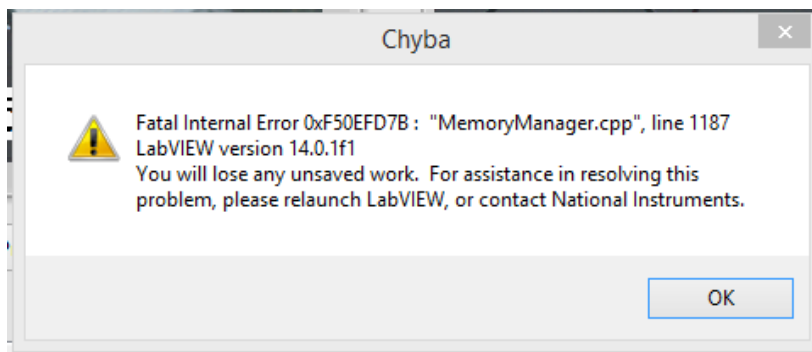


Obr. 31 Nastavení aplikace CleanMem pro periodické čištění filecache

## 8.3 Chyba memory manageru LabVIEW

### 8.3.1 Výskyt chyby

Nebylo možné vysledovat žádnou časovou závislost této chyby, protože se opakovaně projevovala v rozsahu přibližně 1 až 7 dní provozu dohledového systému. Z chybového hlášení na Obr. 32 jasně vyplynulo, že došlo nejen k pádu samotné aplikace, ale i celého LabVIEW Runtime Engine, protože se problém týkal jeho interní komponenty – memory manageru.



Obr. 32 Chybové hlášení - memory manager LabVIEW

V případě takové chyby generuje Runtime tzv. Dump File [35], který obsahuje informace o podmínkách a stavu HW/SW, které ji provázely. Při pádu vývojového prostředí LabVIEW zase vzniká Crash Log, který je vhodné použít při řešení problémů s NI technickou podporou [36]. Z analýzy logu technickou podporou se zjistilo, že chyba vznikla při alokaci paměti pro funkce překreslující čelní panel VI. Logickým krokem pro potvrzení této informace bylo otestování aplikace s vypnutým živým náhledem, při kterém neustále dochází k obnovování obrazu z kamer v obou indikátorech, což je z hlediska

překreslování čelního panelu pravděpodobně nejnáročnější operace. S vypnutým živým náhledem vydržela aplikace bez problému trvale fungovat po dobu 14 dní, následně byl test ukončen.

Další konzultace s technickou podporou poskytly návrhy možných příčin problému, které jsou popsány v následujících kapitolách 8.3.2 až 8.3.4.

### 8.3.2 Neplatný ukazatel nebo únik paměti

Jedná se o situaci, kdy se program pokusí přistoupit k místu v paměti, avšak ukazatel odkazuje například úplně mimo její fyzický rozsah. V aplikaci dohledového systému se ukazatele používají pro práci s bufferem obrázků, který je ovšem alokovan už při startu a uvolněn během ukončování. Po celou dobu běhu se jednotlivé komponenty aplikace odkazují vždy dovnitř bufferu a nezdá se tedy pravděpodobné, že by k takové situaci mohlo dojít. Únik paměti je důsledkem opakované dynamické alokace paměti, kterou následně program neuvolní – způsobeno chybou programátora, který zapomene na příkaz pro uvolnění paměti, pokud už ji dále v programu nehodlá použít. Důsledkem takové chyby by byl ale postupný nárůst alokované paměti pozorovatelný i ve Správci úloh, což by bylo odhaleno už při řešení problému z kapitoly 8.2.

### 8.3.3 Fragmentace operační paměti PC

V případě nevhodného zacházení s pamětí může při dlouhodobém provozu aplikace dojít ke stavu, že celkově sice obsahuje dostatek volného prostoru, ten ale je fragmentovaný do mnoha menších částí. Konkrétně u kamerového systému mohlo být příčinou použití referencí na obrázek a to souběžně ve více procesech – kamera, modul pro ukládání i grafického uživatelského rozhraní. Dle emailu z technické podpory NI ze dne 9.7.2015 dochází v takové případě k alokaci paměti, pravděpodobně za účelem vytvoření kopie ukazatele (reference). Ke zjištění vlivu tohoto jevu lze použít například utilitu WMMMap [37], která oproti Správci úloh podrobněji monitoruje využití paměti jednotlivými procesy a zobrazuje i strukturu neobsazené paměti. Analýza pomocí WMMMap ukázala, že největší celistvý blok volného místa v paměti přesahuje i po několika dnech provozu dohledového systému hodnotu 2 GB, následován bloky o velikosti v řádu stovek MB. Fragmentaci bylo tedy možné vyloučit.

### 8.3.4 Nevhodná strategie překreslování obrazů z kamer

Lze velmi snadno vyčíslit, kolik snímků se na monitoru teoreticky objeví například za týden provozu i při nízké snímkovací frekvenci:

$$N = K \cdot FPS \cdot D \cdot 24 \cdot 60 \cdot 60 = 2 \cdot 15 \cdot 7 \cdot 24 \cdot 3600 = 18\,144\,000 \text{ snímků}$$

Kde K je počet kamer, FPS snímkový kmitočet a D počet dní. Při hodnotě přes 18 milionů se jistě stojí za to zamyslet nad tím, jakým způsobem s nimi pracovat. Aby počet zbytečně nenarůstal, posílají se při vysokém FPS k zobrazení pouze některé snímky – rozebráno v 7.7. Dále se nabízí dvě základní možnosti, jakým občerstvovat indikátory čelního panelu. Za prvé je možné v GUI zapsat nový obrázek do indikátoru pokaždé, když od kamery přijde. Za druhé uložit si ukazatel v proměnné a občerstvovat oba indikátory kamer naráz v samotné smyčce s časováním, například 15 krát za sekundu. Ačkoliv se první způsob může zdát nesmyslný a druhý daleko hospodárnější, na výskyt chyby to při testování nemělo vliv a proto byl zvolen způsob první, protože je jednodušší. Je totiž třeba zohlednit, jakým způsobem vůbec pracuje samotné LabVIEW. Příchod nových dat do terminálu indikátoru totiž neznamená, že se okamžitě aktualizují i na čelním panelu. Ve skutečnosti totiž kód aplikace běží v jiném vlákne než uživatelské rozhraní – čelní panel. (Toto je vlastnost všech VI a nesouvisí s modulární strukturou aplikace kamerového systému!) Mezi těmito dvěma vlákny existuje buffer, ve kterém

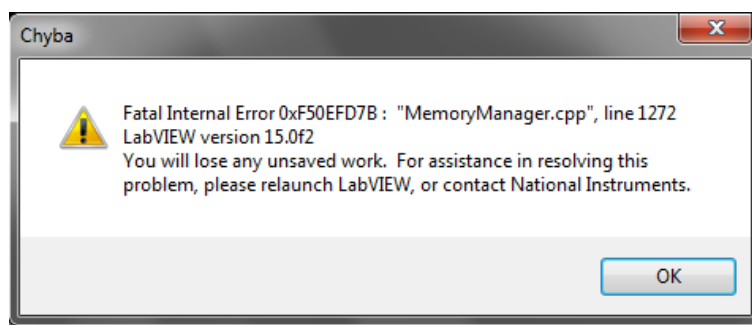


například běžící kód aktualizoval hodnotu proměnné 100 krát za sekundu, ale k aktualizaci čelního panelu dojde ve skutečnosti například jen 10 krát. Strategii překreslování čelního panelu tedy volí LabVIEW samo a programátor má pouze možnost vynutit synchronní zobrazení, tedy překreslení s každým příchodem nových dat do terminálu indikátoru. [38] To by samozřejmě v aplikaci kamerového systému nemělo žádný smysl a na funkci by to mělo naopak naprosto katastrofální následky. Nefungoval by vůbec nebo jen velmi pomalu.

### 8.3.5 Řešení

Žádná z navrhovaných příčin problémů se bohužel nepotvrdila, nikam nevedlo ani obnovení PC respektive OS do stavu, ve kterém byl dodán výrobcem, pomocí záložního obrazu disku, a následně instalace nejnovější verze potřebného SW od National Instruments. Stejná chyba se objevila i při použití aplikace na jiném PC s odlišnou HW konfigurací a OS Windows 8.1 Pro. Zbývalo tedy pouze nějak využít faktu, že chyba se projevuje jen se zapnutým živým náhledem. Vzhledem k určení aplikace se nepředpokládá, že by obraz z kamer trvale někdo sledoval a kontrola bude prováděna spíše namátkově. Živý náhled by tedy mohl být trvale vypnutý a v případě potřeby by jej obsluha zapnula tlačítkem. Nelze ovšem spoléhat na to, že následně náhled bude zase ručně vypnut, proto byla aplikace vybavena nastavitelným časovačem, který po uplynutí dané doby od posledního zásahu obsluhy náhled vypne automaticky. Zákazník s touto úpravou souhlasil. Takto upravená aplikace s občasným zapnutím náhledu bez problému fungovala 14 dní v kuse, aniž by se chyba projevila, proto byl test ukončen.

Řešení sice umožňovalo uvedení aplikace do provozu, ale pochopitelně nebylo ideální a v podstatě problém jen obcházel. Především omezovalo použití aplikace v jiných projektech, kde by trvalý náhled mohl být vyžadován. Zásadní obrat měla znamenat informace od technické podpory NI ze dne 31.12.2015, že byla vydána aktualizace pro vývojové prostředí LabVIEW 2015 f3, která by reportovaný problém měla řešit. Byl tedy vytvořen nový exe soubor aplikace zkompileovaný v LabVIEW s nainstalovanou výše uvedenou aktualizací. Pro test nové verze posloužilo PC HP ProDesk s Windows 7 professional, na kterém byl LabVIEW Runtime verze 15.0f2. Bohužel výsledek byl prakticky totožný s minulými testy, chyba se pouze posunula o několik řádků kódu memory manageru níže Obr. 33.



Obr. 33 Chybové hlášení - memory manager LabVIEW (verze runtime 15.0f2)

Protože další testování by opět znamenalo velkou časovou investici s nejistým výsledkem, bylo rozhodnuto o nasazení aplikace s automatickým vypínáním živého náhledu.

## 9 Nasazení

Všechny komponenty kamerového systému byly na základě předávacího protokolu dodány zákazníkovi, včetně CD se zálohou softwaru a uživatelským manuálem. Před předáním byla provedena navíc záloha celého PC v podobě obrazu disku pomocí softwaru Acronis True Image, kterou lze použít například při poruše HDD nebo problému s OS. Mechanickou stránku instalace zajistili technici zákazníka, tj. upevnění kamer a propojení komponent dle schématu na Obr. 18, který je součástí uživatelského manuálu.

Po oživení proběhlo školení osoby zodpovědné za provoz kamerového systému, především o možnostech konfigurace softwaru, které jsou podrobně rozebrány v uživatelském manuálu. Při následném provozu nebyly hlášeny problémy, kamerový systém běží s automatickým vypínáním živého náhledu a zatížení odpovídá specifikacím popsáním v kapitole 2.



Obr. 34 Záběr z kamery nad plastovým vyliskem



Obr. 35 Detail drážky a dávkovací jehly z kamery na robotu

## 10 Závěr

V rámci diplomové práce se podařilo realizovat kamerový systém pro monitoring průmyslového robotického ramena, který umožňuje triggovat záznam napět'ovým signálem z PLC. Parametry systému jsou přesně na míru přizpůsobeny požadavkům zákazníka. Řešení bylo navrženo z takových hardwarových komponent, aby celkově bylo co nejjednodušší a vyžadovalo minimální množství kabeláže, které by zasahovalo do již současné instalace pracoviště robota. Dvě kamery Basler ace jsou připojeny k PC pomocí Ethernetu a napájeny technologií PoE, k detekci triggovacího signálu slouží externí I/O modul Quido komunikující přes RS232. Kamerový systém je možné ovládat buď přímo z technického zázemí pracoviště, nebo přes vzdálenou správu – PC je připojeno do LAN. Při výběru parametrů hardwaru bylo nutné konfrontovat představy zákazníka s produkty reálně dostupnými na trhu a rovněž zohlednit finanční stránku, což vedlo k určitému kompromisu a tedy i úpravě požadavků.

Před realizací byl proveden průzkum trhu, zda není možné zadání naplnit pomocí některého softwaru pro záznam z kamer nebo zabezpečovací systémy. Protože žádný z dostupných softwarů nesplňoval požadavky, byla pro kamerový systém vyvinuta aplikace v programovacím jazyce LabVIEW, která umožňuje zobrazit a zaznamenávat obraz ze dvou kamer. Mezi klíčové vlastnosti aplikace patří nastavení časování nahrávání kamer vůči triggovacímu signálu, dále možnost triggovací události ručně simulovat a rovněž plně automatický management záznamů. Je zajištěno jejich přehledné třídění do adresářů podle dní a data, stejně jako mazání starých položek při nedostatku volného místa na disku. PC ihned po startu Windows nabíhá do aplikace kamerového systému a automaticky nastaví potřebné parametry z konfiguračního souboru, při zapnutí napájení pracoviště robota tedy není nutný zásah obsluhy.

Při implementaci softwaru byl kladen důraz na jeho modularitu a dostatečnou univerzálnost jednotlivých komponent, aby bylo možné jej použít v dalších podobných aplikacích buď ve stejné podobě nebo s mírnou modifikací. Zároveň ale uživatelské prostředí zůstalo maximálně jednoduché, protože rozšířené možnosti zůstávají skryty v konfiguračním souboru. Tímto se navržený software odlišuje od jiných programů pro záznam z kamer dostupných na trhu, které sice umožňují plnit podstatně širší škálu úkolů, než vyžaduje nasazení v tomto konkrétním projektu, ale za cenu složitějšího uživatelského rozhraní, které při spuštění vyžaduje zásah obsluhy.

Před nasazením kamerového systému bylo potřeba ho důkladně otestovat a to zejména při trvalém provozu. Kromě PC, na kterém měl software běžet, byly použity i dvě další s odlišnou konfigurací a OS. Objevily se dvě zásadní chyby, přičemž u jedné bylo možné vysledovat časovou závislost a projevila se pouze na operačním systému Windows 7 Professional. Časově náročnější bylo hledání příčiny u chyby, která nevykazovala časovou závislost a projevila se na různých operačních systémech. Každá i sebemenší úprava v kódu znamenala test v podobě několikadenního chodu aplikace a vyčkávání, zda znovu dojde k pádu. Při hledání řešení byla intenzivně využívána možnost konzultace s technickou podporou National Instruments. Nalezené řešení sice znamenalo pouze kompromis, který mírně omezil parametry aplikace, ale nebyl v rozporu s požadavky zákazníka.

Z hlediska dalšího vývoje má kamerový systém potenciál pro rozšíření o vyhodnocení pohybů robota, následně by výsledek detekce a měření mohl být signalizován řídicímu PLC pomocí digitálního výstupu. Hardware by nebylo nutné nijak modifikovat, protože se s možnostmi rozšíření při návrhu počítalo. Úpravy by se týkaly pouze softwaru, ovšem bylo by nutné dokoupit licenci pro Vision Development Module, který obsahuje potřebné knihovny.

## 11 Citovaná literatura

1. Axis Communications. Axis Camera Station Datasheet. *Axis Communications*. [Online] [Citace: 20. leden 2016.] [http://www.axis.com/files/datasheet/ds\\_camera\\_station\\_63819\\_en\\_1509.pdf](http://www.axis.com/files/datasheet/ds_camera_station_63819_en_1509.pdf).
2. —. AXIS M1125. *Axis Communications*. [Online] [Citace: 20. leden 2016.] <http://www.axis.com/cz/cs/products/product-selector#/product/20581?id=20711&id=20581&id=17052>.
3. Luxriot. Luxriot® VMS Software. *Luxriot*. [Online] [Citace: 20. leden 2016.] <http://www.luxriot.com/eng/products/vms/>.
4. NorPix. StreamPix: Digital Video Recording Software. *Digital Video Recording Software for High Speed - NorPix*. [Online] [Citace: 21. leden 2016.] [https://www.norpix.com/products/streampix/streampix\\_more.php](https://www.norpix.com/products/streampix/streampix_more.php).
5. Edmund Optics. Imaging Electronics 101: Understanding Camera Sensors for Machine Vision Applications. *Edmund Optics*. [Online] [Citace: 4. únor 2016.] <http://www.edmundoptics.com/technical-resources-center/imaging/understanding-camera-sensors-for-machine-vision-applications/>.
6. Pacer. Cameras and Imaging. *Cameras and Imaging Solutions from Pacer*. [Online] [Citace: 4. únor 2016.] <http://www.pacer.co.uk/products/lasers-and-detectors/camerasandimaging>.
7. X Instruments. Everything you need but don't want to know about digital imaging. *X Instruments*. [Online] [Citace: 4. únor 2016.] <http://www.xinstruments.com/knowledge-base/digital-imaging-glossary.html>.
8. Basler. Comparison of the Most Common Digital Interface . *Baslerweb*. [Online] březem 2013. [Citace: 10. únor 2016.] [http://s.baslerweb.com/media/documents/BAS1302\\_White\\_Paper\\_Interface\\_Comparision\\_EN.pdf](http://s.baslerweb.com/media/documents/BAS1302_White_Paper_Interface_Comparision_EN.pdf).
9. Point Grey. Machine Vision Interface Comparison and Evolution. *Point Grey*. [Online] [Citace: 10. únor 2016.] <https://www.ptgrey.com/white-paper/id/10696>.
10. Basler. Images. *Baslerweb*. [Online] [Citace: 10. únor 2016.] <http://www.baslerweb.com/en/news-press/press-images>.
11. VisionX. Karbon Camera Link. *VisionX*. [Online] [Citace: 20. únor 2016.] <http://www.visionx.cz/produkty/podle-vyrobce/bitflow/karbon-camera-link/>.
12. National Instruments. High-Performance NI Smart Cameras. *National Instruments*. [Online] [Citace: 21. únor 2016.] <http://www.ni.com/datasheet/pdf/en/ds-370>.
13. —. Value Smart Cameras for Embedded Vision. *National Instruments*. [Online] [Citace: 21. únor 2016.] <http://www.ni.com/datasheet/pdf/en/ds-481>.
14. —. NI PCIe-1427 - Low-Cost Camera Link. *National Instruments*. [Online] [Citace: 18. 11 2015.] <http://sine.ni.com/nips/cds/view/p/lang/cs/nid/14844>.

15. VisionX. Smart Camera High Performance. *VisionX*. [Online] [Citace: 21. únor 2016.] <http://www.visionx.cz/site/assets/files/1193/smartcamera.900x0.jpg>.
16. Wikipedia. LabVIEW. *Wikipedia, the free encyclopedia*. [Online] [Citace: 31. leden 2016.] <https://en.wikipedia.org/wiki/LabVIEW>.
17. National Instruments. Hardware Integration with NI LabVIEW. *National Instruments*. [Online] 14. červen 2013. [Citace: 31. leden 2016.] <http://www.ni.com/white-paper/14555/en/>.
18. —. Benefits of Programming Graphically in NI LabVIEW. *National Instruments*. [Online] 14. červen 2013. [Citace: 31. leden 2016.] <http://www.ni.com/white-paper/14556/en/>.
19. —. Buy LabVIEW. *National Instruments*. [Online] [Citace: 31. leden 2016.] <http://www.ni.com/labview/buy/>.
20. —. NI Vision Acquisition Software. *National Instruments*. [Online] [Citace: 31. leden 2016.] <http://sine.ni.com/nips/cds/view/p/lang/cs/nid/12892>.
21. Microsoft. Introduction to the C# Language and the .NET Framework. *Microsoft - Developer Network*. [Online] [Citace: 3. únor 2016.] <https://msdn.microsoft.com/cs-CZ/library/z1zx9t92.aspx>.
22. —. Možnosti nákupu nástroje Visual Studio. *Visual Studio*. [Online] [Citace: 3. leden 2016.] <https://www.visualstudio.com/products/how-to-buy-vs>.
23. AForge.NET. AForge.NET - license. [Online] [Citace: 3. únor 2016.] <http://www.aforgenet.com/framework/license.html>.
24. —. AForge.NET - Features. [Online] [Citace: 3. únor 2016.] <http://www.aforgenet.com/framework/features/>.
25. Itseez. OpenCV - About. *OpenCV - Open Source Computer Vision*. [Online] [Citace: 3. únor 2016.] <http://opencv.org/about.html>.
26. —. OpenCV - Platforms. *OpenCV - Open Source Computer Vision*. [Online] [Citace: 3. únor 2016.] <http://opencv.org/platforms.html>.
27. Baslerweb. *aca1300-60gc*. [Online] [Citace: 21. 11 2015.] <http://www.baslerweb.com/en/products/area-scan-cameras/ace/aca1300-60gc>.
28. Papouch. *Quido RS 4/4*. [Online] [Citace: 17. 11 2015.] <http://www.papouch.com/cz/shop/product/quido-rs-4-4-vstupy-vystupy-a-teplomer-na-232-485/>.
29. CPU Benchmarks. *CPU Performance Comparison*. [Online] [Citace: 10. 11 2015.] [http://www.cpubenchmark.net/compare.php?cmp\[\]=1849&cmp\[\]=2234](http://www.cpubenchmark.net/compare.php?cmp[]=1849&cmp[]=2234).
30. National Instruments. National Instruments website. *User Guide and Specification NI PCIe-8236*. [Online] 2 2013. [Citace: 2. 11 2015.] <http://www.ni.com/pdf/manuals/374010a.pdf>.
31. National Instruments. *AVI Codec Support with AVI2 in NI Vision Products*. [Online] 26. 2 2013. [Citace: 2. 12 2015.] <http://www.ni.com/tutorial/14466/en/>.

32. National Instruments. *MODBUS Library for LabVIEW*. [Online] 2. 9 2015. [Citace: 28. 12 2015.] <http://www.ni.com/example/29756/en/>.
33. PcWinTech. CleanMem Free/Pro. *PcWinTech.com*. [Online] [Citace: 6. Leden 2016.] <http://www.pcwintech.com/cleanmem>.
34. Microsoft. File Caching. *Windows Dev Center*. [Online] [Citace: 6. leden 2016.] <http://msdn.microsoft.com/en-us/library/windows/desktop/aa364218%28v=vs.85%29.aspx>.
35. National Instruments. Obtaining a WinDbg Dump File to Troubleshoot Crashes in National Instruments Software. *NI KnowledgeBase*. [Online] 10. březen 2011. [Citace: 7. leden 2016.] <http://digital.ni.com/public.nsf/allkb/D12F9263065435DA8625784F006ECFD4?OpenDocument>.
36. —. NI Error Reporting FAQ. *NI KnowledgeBase*. [Online] 21. červenec 2011. [Citace: 7. leden 2016.] <http://digital.ni.com/public.nsf/allkb/2FF01AD2F6A02ABD862578D40053FCC7?OpenDocument>.
37. Russinovich, Mark. VMMap v3.21. *Windows Sysinternals*. [Online] 20. červenec 2015. [Citace: 7. leden 2016.] <https://technet.microsoft.com/en-us/sysinternals/vmmap.aspx>.
38. Bress, Thomas. *Effective Labview Programming*. Allendale : National Technology & Science, 2013. 19-348-9108-8.
39. National Instruments. Using Synchronous Display for LabVIEW Controls and Indicators. *NI KnowledgeBase*. [Online] 4. listopad 2013. [Citace: 7. leden 2016.] <http://digital.ni.com/public.nsf/allkb/7B3E63FBA3714447862565D20069CE5F>.
40. Papouch. *Řada Quido*. [Online] [Citace: 18. 11 2015.] <http://www.papouch.com/cz/shop/products/io-moduly/quido/>.
41. National instruments website. *NI Vision Acquisition Software*. [Online] [Citace: 6. 11 2015.] <http://sine.ni.com/nips/cds/view/p/lang/cs/nid/12892>.

## 12 Seznam příloh

### 12.1 Elektronické přílohy (obsah CD)

- **KamRobot.zip** – kompletní software kamerového systému včetně konfiguračních souborů pro kamery
- **Manual.pdf** – uživatelský manuál