

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2019

Michal Čičatka



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

DEPARTMENT OF BIOMEDICAL ENGINEERING

POSTUPY SEŘÍZENÍ DIAGNOSTICKÉHO PŘÍSTROJE ARRAY READER C-SERIES

ADJUSTMENT OF DIAGNOSTIC INSTRUMENT ARRAY READER C-SERIES

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Michal Čičatka

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Martin Mézl, Ph.D.

BRNO 2019

Bakalářská práce

bakalářský studijní obor **Biomedicínská technika a bioinformatika**

Ústav biomedicínského inženýrství

Student: Michal Čičatka

ID: 195723

Ročník: 3

Akademický rok: 2018/19

NÁZEV TÉMATU:

Postupy seřízení diagnostického přístroje Array Reader C-series

POKYNY PRO VYPRACOVÁNÍ:

1) Proveďte rešerši v oblasti microarray readerů a jejich využití v laboratorní technice. 2) Seznamte se s principy kontroly jakosti a finalizace diagnostických přístrojů. 3) Seznamte se se zařízením BioVendor Array Reader C-Series a navrhnete postupy pro povýrobní a servisní seřízení přístroje. 4) Navržené postupy implementujte pomocí automatizovaných rutin založených na algoritmech zpracování obrazu. 5) Proveďte kontrolu správnosti seřízení přístroje pomocí autodiagnostické aplikace. 6) Proveďte diskuzi navrženého řešení s ohledem na časovou náročnost celého seřízení a uživatelský komfort.

Práce je realizována ve spolupráci s firmou BioVendor.

Konzultant práce: Dr. Karásek

DOPORUČENÁ LITERATURA:

[1] BRONZINO, Joseph D. The biomedical engineering handbook. 3rd ed. Boca Raton: CRC/Taylor & Francis, 2006. ISBN 9780849321245.

[2] WANG, X., M. J. HESSNER, Y. WU, N. PATI a S. GHOSH. Quantitative quality control in microarray experiments and the application in data filtering, normalization and false positive rate prediction. Bioinformatics. 2003, 19(11), 1341-1347 . DOI: 10.1093/bioinformatics/btg154. ISSN 1367-4803.

Termín zadání: 4.2.2019

Termín odevzdání: 24.5.2019

Vedoucí práce: Ing. Martin Mézl, Ph.D.

Konzultant: Ing. et Ing. Jan Karásek Ph.D.

prof. Ing. Ivo Provazník, Ph.D.
předseda oborové rady

UPOZORNĚNÍ:

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

ABSTRAKT

Práce se věnuje aktualizaci aplikace pro seřízení biochemického microarray analyzátoru Array Reader C-series společnosti BioVendor. Nejprve obecně definuje a rozdělí chemické analyzátoři; následně popíše jejich fyzikální principy. Další kapitoly shrnují poznatky o microarray testech, které jsou využívány právě analyzátoři microarray a jsou popsány jejich chemické a fyzikální principy. Dále se práce zabývá jakostí a její kontrolou a seřízením přístrojů. Krátce se práce věnuje také .NET Frameworku, softwarové platformě, pomocí níž je implementována praktická část práce. V závěru teoretické části je následně popsán samotný Array Reader C-series. Praktická část dokumentuje stávající aplikaci pro seřízení. Na základě osobních zkušeností se stávající aplikací je vytvořen a podrobně popsán návrh nové aplikace. Hlavní úlohou nové aplikace by mělo být zpřehlednění, zjednodušení a automatizace seřizovacího procesu. Dále práce popisuje implementaci návrhu aplikace včetně všech automatizačních procesů. Závěr práce srovnává původní verzi a novou verzi aplikace z hlediska časové náročnosti a uživatelského komfortu. Aplikace byla vyhodnocena jako lepší z hlediska časové náročnosti i uživatelského komfortu oproti své původní verzi. Aplikace byla dále rozšířena pro fluorescenční analyzátor a pro oba přístroje byla vytvořena i odvozená aplikace pro servisní seřízení. Všechny čtyři aplikace jsou již implementované do softwaru společnosti BioVendor a užívají se při běžném provozu.

KLÍČOVÁ SLOVA

analyzátor, microarray, Array Reader C-series, seřízení, OpenCV

ABSTRACT

Thesis deals with actualization of application for adjustment of the chemical analyser Array Reader C-series developed by company BioVendor. At the beginning it defines and sorts out chemical analysers; then describes their physical principles. Following chapters sum up knowledge about microarray analysis and describes it's chemical and physical principles. Furthermore, the thesis deals with quality and it's management and machine adjustment. In the thesis is also mentioned .NET Framework – software platform that will be used in the practical part of the thesis. At the end of the theoretical part is described Array Reader C-series. Practical part documents current application for adjusting the analyser. Based on personal experiences with the adjusting application was created and thoroughly described concept of a new adjusting application. Main task of this new concept should be simplification and automatization of the adjusting process. After that implementation of the new concept is described including automatation processes. In the end of this thesis both original and new versions are compared with regards to time consumption and user experience. Newer version of the application was evaluated as better than it's previous version in light of time consumption and user experience. Application was expanded for fluorescent analyzer and for both types of analysers were developed derivated applications for service adjustment. Each of these applications was implemented to the software of BioVendor company and is used on daily basis.

KEYWORDS

microarray, analyser, Array Reader C-series, adjustment, OpenCV

ČIČATKA, Michal. *Postupy seřízení diagnostického přístroje Array Reader C-series*. Brno, Rok, 96 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav biomedicínského inženýrství. Vedoucí práce: Ing. Martin Mézl, PhD.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Postupy seřízení diagnostického přístroje Array Reader C-series“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

podpis autora

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Martinu Mězlovi, Ph.D. za odborné vedení a podnětné návrhy k práci. Velké díky patří rovněž softwarovému oddělení společnosti BioVendor Instruments, zvláště potom Ing. et Ing. Janu Karáskovi, PhD. a Ing. Haně Dvořáčkové za věcné připomínky k práci a Ing. Petru Dvořáčkovi za odborné konzultace.

Brno

.....

podpis autora

Obsah

Úvod	14
1 Chemické analyzátory	15
1.1 Analyzátory pracující s kapalnými činidly	15
1.2 Analyzátory pracující se suchými činidly	15
2 Fyzikální principy chemických analyzátorů	16
2.1 Absorbance	16
2.1.1 Kolorimetrie	17
2.2 Fluorescence	17
2.3 Luminiscence	18
2.4 Nefelometrie a turbidimetrie	18
3 Microarray testy	19
3.1 Nosiče vzorků	19
3.1.1 Mikrotitrační destičky	19
3.1.2 Sklíčka	21
3.1.3 Gely	21
3.1.4 Stripy	21
4 Imunoanalytické metody	22
4.1 Enzymové metody na fotometrických principech	22
4.1.1 Enzymoimunoanalýza v homogenním prostředí (EIA)	22
4.1.2 Enzymoimunoanalýza v heterogenním prostředí (ELISA)	23
4.2 Luminiscenční metody	23
4.2.1 Luminiscenční imunoanalýza (LIA)	23
4.3 Fluorescenční metody	23
4.3.1 Fluorescenční imunoanalýza (FIA)	23
5 Analyzátory microarray	25
5.1 Komponenty microarray analyzátorů	25
5.1.1 Optické detektory	25
5.1.2 Posuvný systém	28
5.1.3 Osvit	29
5.1.4 Počítač a doplňující elektronika	29
5.2 Analýza microarray	30

6	Jakost, kontrola jakosti a seřizování přístrojů	31
6.1	Definice jakosti	31
6.2	Parametry jakosti	31
6.3	Jakost biochemických analyzátorů	32
6.4	Kontrola jakosti a seřizování	32
6.5	Instalační, operační a procesní kvalifikace	33
6.6	Interkoefficient a intrakoefficient variance	33
7	.NET Framework	34
7.1	Součásti .NET Framework	34
7.1.1	Jazyk a virtuální stroj	34
7.1.2	Microsoft Visual Studio	35
8	Array Reader C-series	36
8.1	Charakteristika zařízení	36
8.2	Výstupní kontrola Array Readeru C-series	36
8.2.1	Aplikace pro seřizování	37
8.2.2	Příprava analyzátoru na seřizování	42
8.2.3	Seřizování analyzátoru	42
8.2.4	Časová náročnost seřizování	43
8.2.5	Problematické části stávajícího řešení aplikace pro seřizování	43
8.3	Srovnání s konkurenčním microarray analyzátozem Sensovation CLAIR	44
9	Návrh aktualizace aplikace pro seřizování přístroje Array Readeru C-series	45
9.1	Zadání aplikace	45
9.2	Implementace aplikace	45
9.3	Struktura aplikace	46
9.3.1	Společné funkce všech obrazovek	46
9.3.2	První obrazovka - obecné nastavení	47
9.3.3	Druhá obrazovka - nastavení krajních poloh	47
9.3.4	Třetí obrazovka - zaostření horní kamery	49
9.3.5	Čtvrtá obrazovka - nastavení referenčních pozic	51
9.3.6	Pátá obrazovka - osvit	52
9.3.7	Šestá obrazovka - nastavení spodní kamery	53
9.3.8	Sedmá obrazovka - záloha a archivace seřizování	54
10	Implementace návrhu aplikace	56
10.1	Struktura a základní funkce	56
10.1.1	Implementace jádra aplikace	56

10.1.2 Implementace jednotlivých obrazovek	57
10.2 Automatizační algoritmy	64
10.2.1 Implementace algoritmu pro nalezení bezpečných oblastí	64
10.2.2 Algoritmy využívající zpracování obrazu	65
11 Srovnání původní a nové aplikace	70
11.1 Časová náročnost	70
11.1.1 Časová náročnost původní aplikace	70
11.1.2 Časová náročnost nové aplikace	70
11.1.3 Srovnání časových náročností staré a nové aplikace	71
11.2 Uživatelský komfort	71
12 Závěr	73
Literatura	75
Seznam symbolů, veličin a zkratk	79
Seznam příloh	80
A Srovnání analyzátorů microarray Array Reader C-series a Sensation CLAIR	81
B Zdrojové kódy	84

Seznam obrázků

2.1	Blokové schéma spektrofotometrie	16
2.2	Blokové schéma fluorimetrie	17
2.3	Blokové schéma nefelometrie (vlevo) a turbidimetrie (vpravo)	18
3.1	Mikrotitrační destičky [39]	20
3.2	Nákres rozložení antigenů ve vzorku [40]	20
5.1	Blokové schéma analyzátorů microarray	25
5.2	Schématické zobrazení CCD čipu (V_1, V_2 - gradientní napětí v sloupcích, H_1, H_2 - gradientní napětí v posuvném registru, V_{out} - výstupní napětí) [38]	26
5.3	Schéma fotonásobiče [37]	27
5.4	Krokový motor [36]	28
5.5	Servomotor [41]	29
5.6	Výsledek analýzy jedné jamky mikrotitrační destičky	30
8.1	Array Reader C-series [35]	37
8.2	Aplikace pro seřízení	37
8.3	Panel <i>Basic</i>	38
8.4	Fantom mikrotitrační destičky s vyznačenými referenčními pozicemi	39
8.5	Panel <i>Lights/Camera</i> - seřízení dolního osvitu (vlevo a uprostřed), seřízení horního osvitu (vpravo)	40
8.6	Rozvržení os v Array Readeru C-Series	41
8.7	Panel <i>Motors</i> - první část (vlevo), druhá část (vpravo)	41
9.1	Návrh společné části všech obrazovek	47
9.2	Návrh první obrazovky - obecné nastavení	48
9.3	Návrh druhé obrazovky - bezpečné oblasti	49
9.4	Návrh třetí obrazovky - nastavení zaostření horní kamery	50
9.5	Bílý kruh způsobený prosvětlením otvoru fantomu	51
9.6	Návrh čtvrté obrazovky - nastavení referenčních pozic	52
9.7	Návrh páté obrazovky - nastavení osvitu	53
9.8	Návrh šesté obrazovky - nastavení spodní kamery	54
9.9	Návrh sedmé obrazovky - záloha a archivace seřízení	55
10.1	UML třídový diagram jádra aplikace	57
10.2	Společná část obrazovek v prostředí Microsoft Visual Studio	58
10.3	První obrazovka aktualizované aplikace - obecná nastavení	59
10.4	Druhá obrazovka aktualizované aplikace - nastavení bezpečných oblastí	60
10.5	Třetí obrazovka aktualizované aplikace - nastavení výšky kamery	61
10.6	Čtvrtá obrazovka aktualizované aplikace - nalezení referenčních pozic	61
10.7	Pátá obrazovka aktualizované aplikace - nastavení osvitu	62

10.8 Šestá obrazovka aktualizované aplikace - nastavení spodní kamery . .	63
10.9 Sedmá obrazovka aktualizované aplikace - záloha a archivace seřízení	64
10.10 Sekvenční diagram popisující práci s metodami při zpracování obrazu	67

Seznam tabulek

8.1	Vybrané seřizovací úkony	43
11.1	Čas strávený na jednotlivých panelech ve staré aplikaci	70
11.2	Čas strávený na jednotlivých obrazovkách v nové aplikaci	71
11.3	Srovnání staré a nové aplikace v jednotlivých krocích	72
A.1	Srovnání ArrayReaderu C-series s analyzátozem Sensovation CLAIR .	81

Seznam výpisů

7.1	Program HelloWorld v jazyce C#	35
B.1	Metody pro nastavení bezpečné oblasti	84
B.2	Algoritmus pro nalezení bezpečných oblastí	85
B.3	Metoda FindCircle	87
B.4	Třída CalibrationImage	89
B.5	Metoda CenterWell()	91
B.6	Algoritmus pro nalezení výšky kamery	93
B.7	Algoritmus pro nalezení referenčních pozic	95

Úvod

V poslední době se v oboru moderních technologií neustále setkáváme s miniaturizací. Produkty se zmenšují za různými účely, od vyšší ekonomičnosti, přes vyšší efektivitu, až po modernější design, nebo lepší ergonomii. Miniaturizace se nevyhla ani medicíně. Stále častěji slyšíme o nanopřenašečích léků či nanovlákních, které budou vyztužovat cévy nejen kardiologických pacientů. Můžeme ji sledovat i v laboratorní medicínské technice. Proto se tato práce zaměřuje na technologii microarray, jenž umožňuje analýzu vysokého počtu testů, na vysokém počtu vzorků v krátkém čase. Postupně jsou v práci popsány chemické principy testů, fyzikální principy analyzátorů pro testy a krátká část je věnována i .NET Frameworku, jenž hrál úlohu v navazující bakalářské práci. Jelikož o technologii microarray není příliš mnoho ucelených informací, při tvorbě této práce se využívalo především dokumentací konkrétních přístrojů. Navazující bakalářská práce bude zaměřena na seřizování konkrétního analyzátoru microarray, a proto i tato práce se bude tomuto tématu věnovat.

1 Chemické analyzátory

Chemické analyzátory se v České republice masově rozšířily do chemických a biochemických laboratoří v 90. letech minulého století. Jejich úkolem byla automatizace chemických testů, které se do té doby prováděly manuálně. Automatizace vedla zejména ke zvýšení efektivity a často i ke snížení nákladů na dané testy. Chemické analyzátory se využívají v infekční sérologii, imunologii, molekulární diagnostice, veterinární diagnostice a mnoha dalších odvětvích medicíny. Měří se pomocí nich koncentrace širokého spektra látek, a to od nejrůznějších proteinů (inzulin, CPR, imunoglobuliny apod.) přes vitamíny až po stopové prvky.

Chemické analyzátory lze členit podle mnoha parametrů; nejčastěji se rozdělují do dvou základních skupin - analyzátory pracující s kapalnými činidly a analyzátory pracující se suchými činidly. [17]

1.1 Analyzátory pracující s kapalnými činidly

S kapalnými činidly pracují spíše velké chemické analyzátory. Většinou využívají fotometrické metody nebo měření iontově selektivními elektrodami. Obsahují karusel s větším počtem reagensů a umožňují tedy stanovení více analytů. Stejně tak mají i vzorkový karusel, díky němuž mohou stanovovat analyt pro více pacientů najednou. Reagencie a vzorky se pipetují dávkovačem do reakčních nádobek, které mohou být současně i měřicími kyvetami. Způsoby měření se liší přístroj od přístroje. Nalézají uplatnění v imunodiagnostice, kardiologii, hematologii a v mnoha dalších odvětvích vědy a medicíny. [17]

1.2 Analyzátory pracující se suchými činidly

Činidla, která slouží k detekci daných chemických látek nepipetuje přístroj do vzorků v kapalném stavu, ale jsou nanášeny na mikrotitrační destičky, proužky (stripy), sklíčka či gely. Analyzátory využívají pro detekci nejčastěji principy kolorimetrie, potenciometrie a imunochemie. Znáмым zástupcem takového analyzátoru je glukózový analyzátor Reflotron. Do této skupiny se taktéž řadí analyzátory microarray. [17]

2 Fyzikální principy chemických analyzátorů

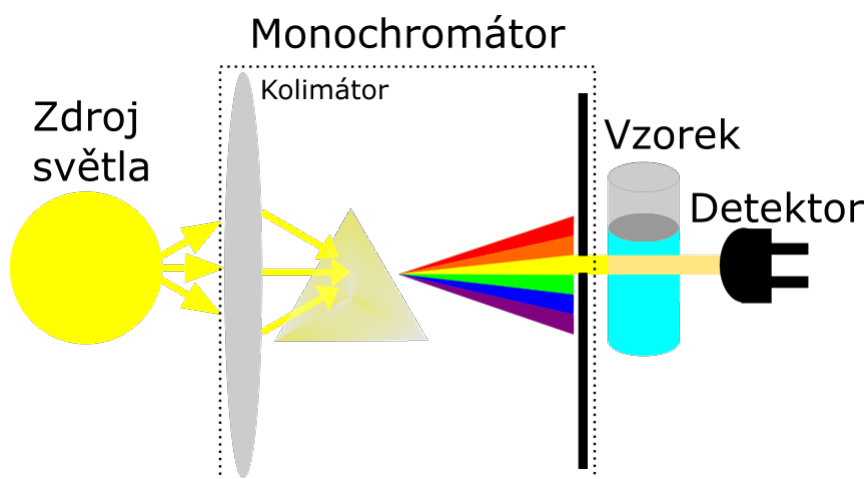
Chemické analyzátoary měří různé veličiny jako např. absorbanci, turbiditu či fluorescenci. Podle měřených veličin se liší konstrukce analyzátorů a možnosti jejich uplatnění. Tato kapitola představí veličiny, které chemické analyzátoary využívají nejčastěji a přiblíží možnosti jejich využití.

2.1 Absorbance

Absorpční analyzátoary fungující na principu optické metody zvané spektrofotometrie. Podle ní je koncentrace zkoumaného zbarveného roztoku přímo úměrná intenzitě prošlého světla. V absorpčním analyzátoaru tedy najdeme zdroj světla a monochromátor (světlo musí mít určitou vlnovou délku). Světlo prochází vzorkem, jeho útlum je měřen detektorem umístěným za měřeným vzorkem. Využívá se vlnových délek nejen viditelného, ale i ultrafialového nebo infračerveného světla. Měřenou veličinou je v případě spektrofotometrie absorbance, jejíž vztah popisuje Lambert-Beerův zákon (rovnice 2.1),

$$A = \epsilon lc \quad (2.1)$$

kde A je absorbance, ϵ je molární absorpční koeficient, l je tloušťka vzorku a c je koncentrace vzorku. Při naměření absorbance daného vzorku pro několik vlnových délek se získá absorpční spektrum. Z absorpčního spektra se určí absorpční maximum – vlnová délka, při které je absorbance nejvyšší. S touto vlnovou délkou se poté měří absorbance vzorků pro známé koncentrace. Směrnice vzniklé křivky ze známých hodnot pro koncentrace popisuje přímou úměru mezi koncentrací a absorbancí. Na obrázku 2.1 je blokové schéma spektrofotometru. [15, 16, 17, 23]



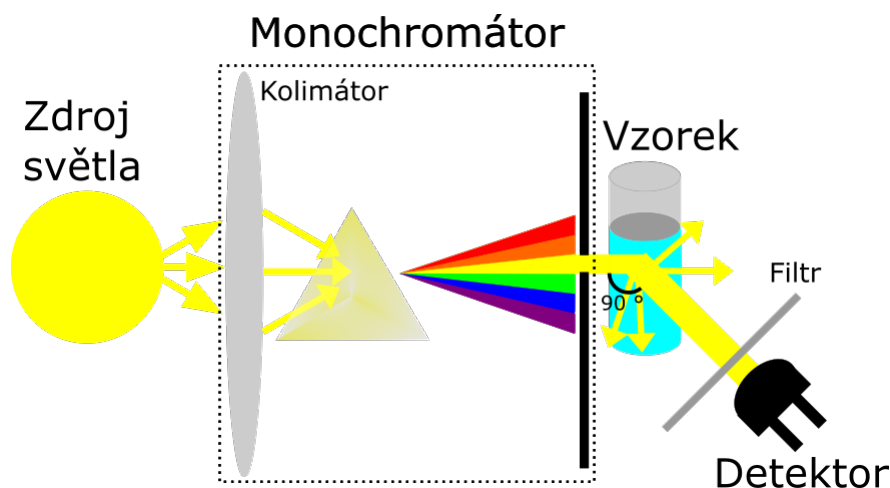
Obr. 2.1: Blokové schéma spektrofotometrie

2.1.1 Kolorimetrie

Kolorimetrie je zvláštní typ spektrofotometrie. Její princip je obdobný, avšak interval vlnových délek je zúžen pouze na viditelnou část spektra (380-760 nm). Namísto fotocitlivých prvků se jako detektorů často využívá kamer. Pokud nemá daný vzorek kalibrační křivku (body o známých koncentracích), výstup analýzy nemá kvantitativní charakter, ale pouze kvalitativní.

2.2 Fluorescence

Fluorescenční analyzátoři pracují s fyzikálním jevem fluorescencí. Elektrony měřeného vzorku jsou pomocí zdroje excitačního záření excitovány (vybuzeny do vyššího energetického stavu). Při jejich návratu do stabilního energetického stavu emitují záření, které snímá druhý optický systém (tzv. emisní) pomocí detektoru. Detektor bývá umístěn kolmo k excitačnímu zdroji, což minimalizuje přenos nežádoucího záření. Budící spektrum z pravidla obsahuje jiné vlnové délky než spektrum vyzařované. Při měření fluorescence pro více vlnových délek vzniká tzv. emisní spektrum. Výhodou proti absorpční spektrofotometrii je vyšší citlivost, a to v některých případech až o čtyři řády. Stejně tak se fluorimetrie vyznačuje vyšší specificitou. Útlum prošlého záření u absorpční spektrofotometrie totiž může zapříčinit ve vzorku jiná látka než zkoumaná. Zásadní nevýhodou je závislost fluorescence na pH a teplotě. Na obrázku 2.2 je blokové schéma fluorimetrie. [17, 23]



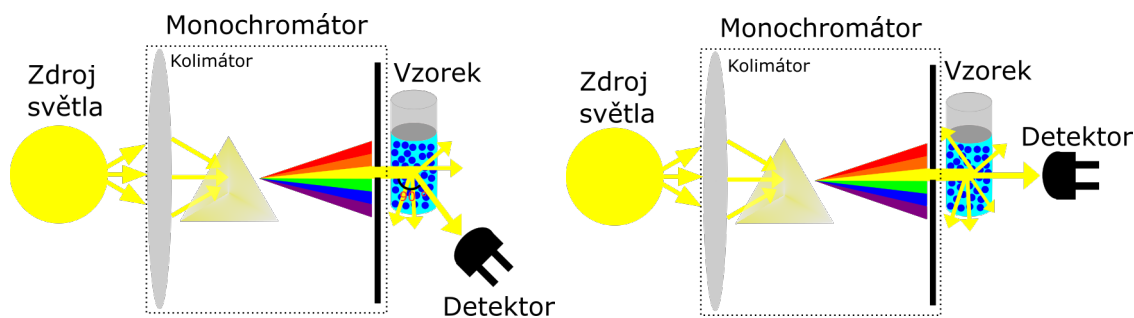
Obr. 2.2: Blokové schéma fluorimetrie

2.3 Luminiscence

Luminiscence je jev, popisující samovolné záření některých chemických látek. Toto záření vzniká jako zbytek tepelného záření látky. Luminiscence vzniká buď v důsledku fluorescence (cizí záření vybudí látku do vyššího energetického stavu) nebo fosforescence (elektron překoná pomocí tunelového jevu potenciálovou bariéru a protuneluje do vrstvy s vyšší energií). Principu fosforescence využívají právě luminiscenční analyzátoři; snímají vzniklé záření pomocí fotonásobiče nebo kamery. Látky, u nichž můžeme pozorovat jev luminiscence, nazýváme luminofory. [25, 17, 23]

2.4 Nefelometrie a turbidimetrie

Nefelometrické a turbidimetrické analyzátoři využívají vzorky s koloidními částicemi. Na těchto koloidních částicích dochází při průchodu světla k tzv. opalescenci (neboli Tyndallovu jevu) - lomu světla. V turbidimetrii se měří pokles intenzity prošlého záření oproti vstupujícímu záření. Pokles intenzity způsobuje odklon od původní trajektorie vlivem lomu na koloidních částicích. Obdobou spektrofotometrické absorbance je turbidance. Zdroj záření, vzorek a detektor jsou při turbidimetrii v jedné ose. U nefelometrie je kyveta se vzorkem umístěna kolmo k detektoru a měří intenzitu rozptýleného záření vystupujícího z kyvety. V praxi se tyto metody využívají nejčastěji ke stanovení proteinů v krvi. Do vzorku krve se přidá protilátka proti měřenému proteinu. Dochází k precipitaci a vzniku komplexů antigen-protilátka. Konkrétním příkladem může být stanovení CRP (C-reaktivní protein), které stoupá při bakteriálních infekcích, nebo imunoglobulinů typů G, A a M, jež napomáhají v diferenciální diagnostice chronických onemocnění jater. Na obrázku 2.3 jsou blokové schémata nefelometrie (vlevo) a turbidimetrie (vpravo). [24, 17, 23]



Obr. 2.3: Blokové schéma nefelometrie (vlevo) a turbidimetrie (vpravo)

3 Microarray testy

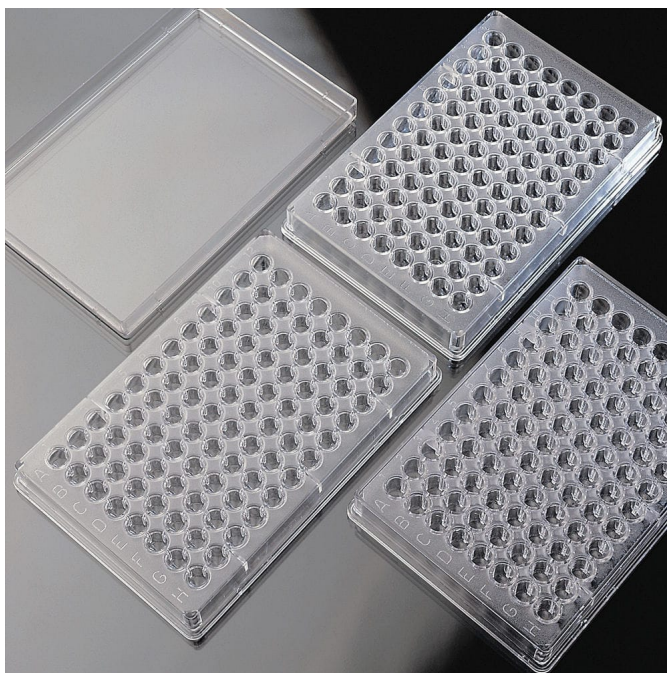
Chemické analyzátory vyhodnocují různé fyzikální veličiny po reakci vzorků s přidávanými reagensy. V dnešní době nabývají stále větší důležitosti tzv. microarray testy. Na nosičích vzorku (sklíčka, gely nebo jamky v mikrotitračních destičkách) jsou nanášeny antigeny jako tzv. spoty. Po přidání vzorku reagují protilátky s antigeny na nosiči. Jedná se o tzv. multiplexní metody; v jednom nosném médiu může proběhnout několik testů zároveň. Principy reakcí spotů a vzorků jsou standardně založeny na imunochemických metodách.

3.1 Nosiče vzorků

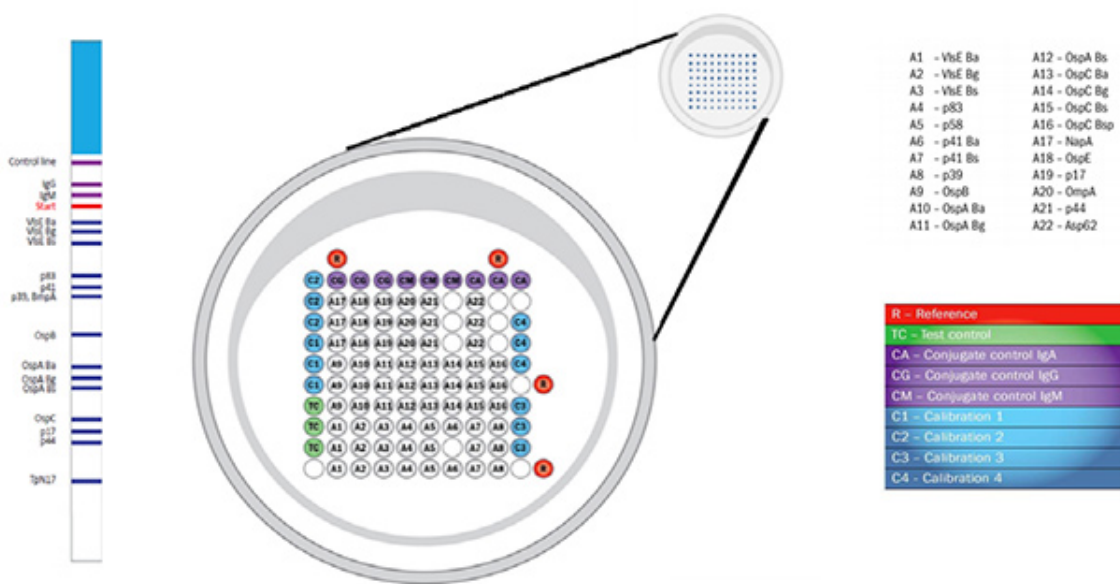
Analyzátory microarray (detailnějšímu popisu analyzátorů microarray se budou věnovat následující kapitoly) pracují s různými typy nosičů (médií, na nichž jsou nanášeny antigeny a po přidání vzorku dochází k chemické reakci). Standardním nosičem vzorků jsou mikrotitrační destičky. Setkat se můžeme také se sklíčkou, gely nebo proužky (tzv. stripy).

3.1.1 Mikrotitrační destičky

Mikrotitrační destičky jsou plastové nebo kovové destičky s jamkami, do kterých se v rámci testu pipetují vzorky. Destičky mají standardně 96 jamek (8 sloupců na 12 řad), 384 jamek (16 sloupců na 24 řad) popř. i 1536 jamek (32 sloupců na 48 řad); mezi sloupci a řadami je dodržen poměr 2:3. V dnešní době jsou naprostým standardem chemických a biochemických analyzačních metod. Pracují s nimi nejen analyzátory microarray, ale i spousta dalších chemických přístrojů. Kromě analyzátorů také např. termocyclery pro *Polymerase Chain Reaction - polymerázová řetězová reakce* (PCR). Stejně tak jsou pro ně vyvíjeny další laboratorní přístroje jako třepačky, centrifugy nebo umývací nástroje. K pipetování vzorků do mikrotitrační destičky je často využíváno multipipetorů. Často se můžeme setkat se speciálně upravenými mikrotitračními destičkami, na jejichž dně je jsou nanášeny reagensy (tzv. spoty) pro biochemické testy (typickým příkladem jsou *Enzyme-Linked ImmunoSorbent Assay - enzymoimunoanalýza v heterogenním prostředí* (ELISA) test). Mikrotitrační destička je na obrázku 3.1. Na obrázku 3.2 je uprostřed vyobrazeno dno mikrotitrační destičky s příklady antigenů. [20, 21, 22]



Obr. 3.1: Mikrotitrační destičky [39]



Obr. 3.2: Nákres rozložení antigenů ve vzorku [40]

3.1.2 Sklíčka

Sklička jsou principiálně podobná mikrotitračním destičkám. Jejich výhodou je, že dokáží pracovat s nižším objemem vzorku (10 μ l). Stejně tak je na skličku naneseno i nižší množství reagensů. [17]

3.1.3 Gely

Zobrazování agarových či akrylamidových gelů je další z odvětví digitálních laboratorních metod. Velice časté je zachycování výsledků z elektroforézy. Gely se nejčastěji zpracovávají pomocí kolorimetrických či fluorescenčních metod. Sklička jsou vyhodnocovány stejně jako mikrotitrační destičky pomocí kolorimetrických nebo fluorescenčních analyzátorů. [17]

3.1.4 Stripy

Stripy jsou úzké proužky s nanesenými antigeny. Vyhodnocovat se mohou pomocí chemických analyzátorů (např. testy moči nebo stanovení množství glukózy v krvi) nebo pouze vizuálně (pH papírky). Stejně jako u mikrotitračních destiček i u stripů se velice často užívá imunochemických metod. Příklad stripu s antigeny je na levé straně obrázku 3.2.

Stripy se často používají v tzv. western blot analýzách, jež slouží ke kvantifikaci určitého proteinu ve směsi jiných proteinů. Proces probíhá následovně. Nejprve jsou proteiny elektroforeticky rozděleny podle své molekulové hmotnosti. Gel se přiloží na speciální membránu a proteiny se přenášejí z gelu na membránu pomocí působení elektrického proudu. Záporně nabitě proteiny migrují na povrch membrány. Povrch membrány se tzv. zablokuje (povrch membrány nespecificky váže všechny proteiny; přidá se levný protein, který se naváže na povrch a protilátky proti hledanému antigenu se poté vážou pouze na dané antigeny a ne na povrch membrány). Na stanovovaný protein se váží primární protilátky a na ty se posléze vážou sekundární značené protilátky (fluorescenčně nebo chemiluminiscenčně). Množství fluorescenčního nebo chemiluminiscenčního záření je přímo úměrně množství stanovovaného proteinu. [18, 19]

4 Imunoanalytické metody

Nosiče vzorků, využívané v microarray testech, velice často pracují na principech imunochemických metod. Jedná se o metody, u nichž dochází k reakci protilátky s antigenem. Reakce je často doprovázena tzv. imunoprecipitací (sražením). Moderní imunoanalytické metody rozdělujeme na:

- radiometrické,
- enzymové na fotometrických principech,
- fluorescenční,
- luminiscenční.

Radiometrické metody nejsou předmětem této práce, jelikož se u nich nevyužívá klasických analyzátorů. Navíc jsou v dnešní době nahrazovány jinými metodami, především kvůli radiální zátěži pro personál, jenž provádí analýzy. Přirozenou náhradou za látky značené radionuklidy přinesly enzymy. [17]

4.1 Enzymové metody na fotometrických principech

Jak již bylo řečeno, enzymové metody víceméně nahradily metody radiometrické. Nejčastěji využívanými enzymy v imunoanalytické chemii jsou peroxidasa (POD) a alkalická fosfatasa (ALP) díky jejich schopnosti katalyzovat reakce, při nichž vznikají barevné produkty. Ty se následně mohou vyhodnocovat chemickými analyzátoři.

Metody můžeme rozdělit na kompetitivní a nekompetitivní. Zástupcem kompetitivních metod je *Enzyme immunoassay - enzymoimunoanalýza v homogenním prostředí* (EIA), zástupcem nekompetitivních metod je ELISA. [17]

4.1.1 Enzymoimunoanalýza v homogenním prostředí (EIA)

EIA využívá antigen (stejný jako stanovovaný analyt), jenž je značený enzymem (nejčastěji glukó-6-fosfátdehydrogenáza) a specifické protilátky. Po smíchání vzorku s analytem, značeným antigenem (konjugátem) a protilátkou (v limitovaném množství) dochází ke „kompetiční soutěži“ mezi analytem a konjugátem o omezená místa protilátky. Množství analytu ve vzorku je poté přímo úměrné enzymové aktivitě nenavázaného konjugátu (konjugát ztrácí enzymovou aktivitu po navázání na protilátku).

Setkat se můžeme i s fluorimetrickou EIA. Po přidání substrátu nevzniká barevný produkt, nýbrž produkt s fluorescenčními vlastnostmi. [17]

4.1.2 Enzymoimunoanalýza v heterogenním prostředí (ELISA)

ELISA je založena na interakci antigenu a protilátky, přičemž na jedno z nich je navázán enzym. Nejčastěji se navazuje protilátka na dno jamek mikrotitračních destiček nebo na plastové kuličky. Po přidání vzorku dochází ke vzniku vazby antigen-protilátka. Nenavázaný antigen se odstraní promytím a přidá se druhá protilátka proti stejnému antigenu, na niž je navázán enzym. Značená protilátka se naváže na antigen a nenavázaná protilátka se odmyje. Přidá se substrát a množství zjišťovaného analytu je přímo úměrné enzymové aktivitě.

4.2 Luminiscenční metody

Metody lze rozdělit na chemiluminiscenční a bioluminiscenční. Bioluminiscenční využívají k detekci enzymy, a proto je možné je řadit i mezi EIA s luminiscenční detekcí. Při chemiluminiscenčních metodách produkt reakce analytu s určitou reagenci vyzařuje světelné záření, nebo má reagentie luminiscenční vlastnosti a po reakci jsou tyto vlastnosti inhibovány. Jako reagentie se používají luminol, isoluminol, deriváty adamantanu apod. [17]

4.2.1 Luminiscenční imunoanalýza (LIA)

LIA lze rozdělit na kompetitivní a nekompetitivní. Metody *Chemiluminescent immunoassay - chemiluminiscenční imunoanalýza (CELIA)* a *Immunoluminometric assay - imunoluminometrická analýza (ILMA)* jsou paralelou metody EIA, místo enzymu na sobě však mají navázané luminofory. Enzymatické a luminiscenční metody mohou být kombinovány (např. pokud reakce substrátu a enzymu u metody EIA nevede ke změně barvy vzorku, ale k luminiscenci). [17]

4.3 Fluorescenční metody

Fluorescenční metody mají oproti spektrofotometrickým metodám vyšší citlivost. Zástupcem fluorescenčních metod může představovat *Fluorescent Immunoassays - fluorescenční imunoanalýza (FIA)*. [17]

4.3.1 Fluorescenční imunoanalýza (FIA)

Jedná se opět o anabázi EIA, místo enzymu je tentokrát navázán ligand s fluorescenčními vlastnostmi. Setkat se můžeme i s pokročilejšími metodami, jako např. *Time-resolved fluorescence immunoassay - fluorescenční imunoanalýza rozložená v čase*

(TR-FIA), u které se měří časové prodlevy mezi excitačním světlem a emitovaným světlem.

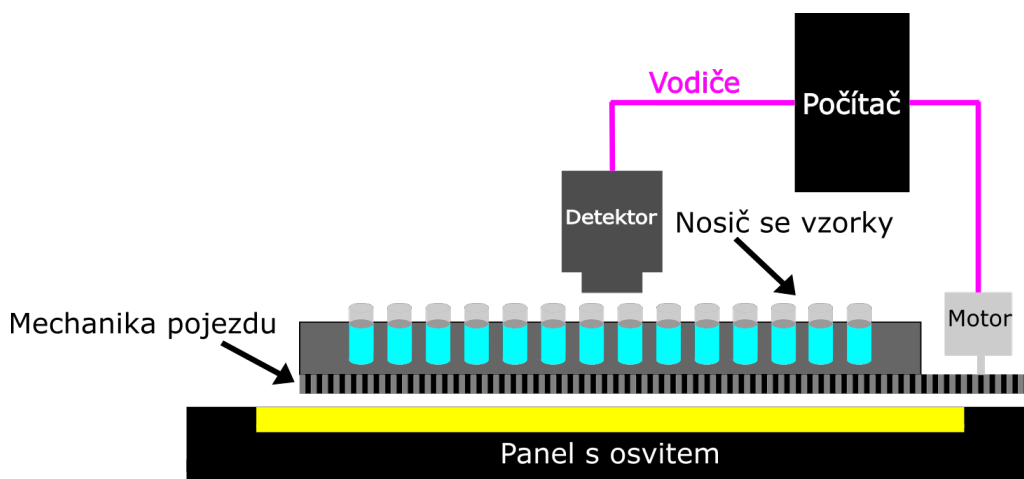
Pokud osvítime fluorofor polarizovaným světlem, bude emitovat opět polarizované světlo. Tohoto principu využívá *Fluorescence Polarization Immunoassay - fluoresenční polarizovaná imunoanalýza* (FPIA), jenž slouží k stanovování koncentrace malých molekul (po navázání na protilátku se výrazně mění směr polarizace). [17]

5 Analyzátory microarray

Analyzátory microarray umožňují čtení tzv. microarray testů. Jsou k dostání v mnoha variantách, a to i v tzv. multimodálních provedeních. Pro detekci biochemických parametrů se využívá zejména kolorimetrie, absorpce světla, luminiscence a fluorescence. Díky většímu počtu čidel v jamkách¹ mikrotitrační destičky a vysokému počtu samotných jamek lze vyhodnocovat širokou škálu testů, a to navíc současně pro více vzorků. Své uplatnění našly zejména v medicíně, farmacii, výzkumu a v mnoha dalších odvětvích. Testy prováděné v analyzátoch microarray mohou mít jak kvalitativní, tak kvantitativní charakter. [1, 2, 3, 4, 5]

5.1 Komponenty microarray analyzátorů

Mezi základní komponenty analyzátorů můžeme zařadit posuvný systém vzorku (a/nebo posuvný systém optického detektoru), detektor, osvětlení, doplňující elektroniku a počítač. Zjednodušené blokové schéma je na obrázku 5.1. [6, 7, 8, 9]



Obr. 5.1: Blokové schéma analyzátorů microarray

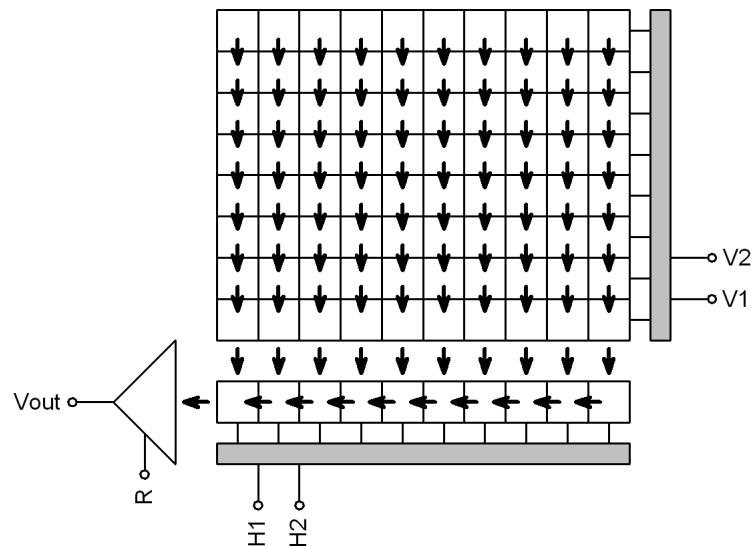
5.1.1 Optické detektory

Jako optické detektory se využívají zejména CCD čipy, CMOS čipy, fotodiody nebo fotonásobiče. Jednotlivé typy detektorů jsou popsány v následujících odstavcích.

¹někdy také označovány anglicismem „well“

CCD čipy

CCD čipy jsou zařízení ke konverzi energie viditelného světla na energii elektrickou. Sestává ze tří vrstev. Na kovové kladně nabitě elektrodě je uložen křemíkový plát, který je izolován vrstvou oxidu křemičitého. Fotony dopadají na atomy křemíku, kde vlivem fotoelektrického jevu dochází k uvolnění elektronů. Elektroda přitahuje elektrony v důsledku Coulombovských sil. Dále se elektrony na elektrodě po řadách přesunují z pixelu na pixel, až na krajní čidlo s vlastní sadou elektrod (posuvný registr). Tento pohyb je zajištěn gradientem napětí v jednotlivých řadách. V posuvném registru se elektrony posunují naopak po sloupcích až ke konci zařízení, kde je načítá terminál jako kontinuální signál. Signál je zesílen a kvantován A/D převodníkem. Celý proces trvá tak dlouho, dokud není získána hodnota každého pixelu. Pokud se tento proces opakuje alespoň pětadvacetkrát za vteřinu, vzniká plynulý obraz. Schéma CCD čipu je na obrázku 5.2. [10]



Obr. 5.2: Schématické zobrazení CCD čipu (V_1, V_2 - gradientní napětí v sloupcích, H_1, H_2 - gradientní napětí v posuvném registru, V_{out} - výstupní napětí) [38]

CMOS čipy

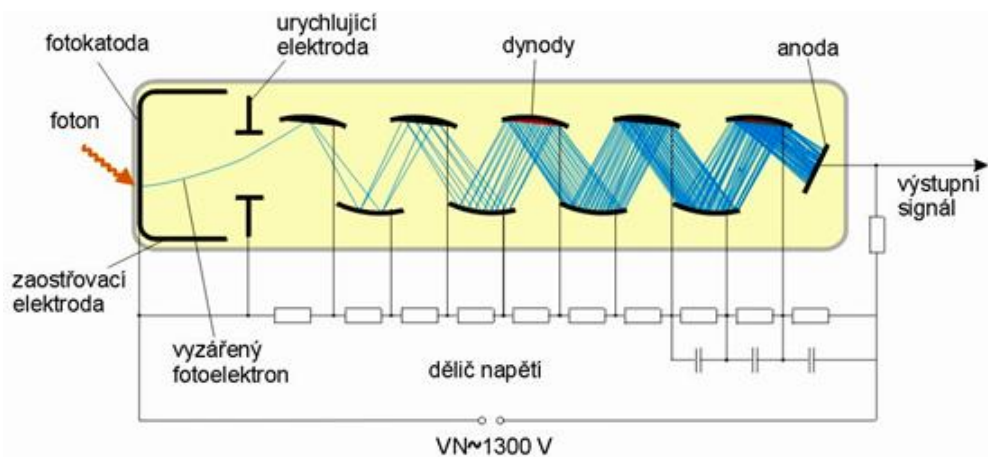
CMOS čip je dvourozměrné pole fotodiód, kde každá má svůj vlastní zesilovač a může být tedy přímo adresována. Výhodou oproti CCD čipům je tedy jednodušší konstrukce, naproti tomu zesilovače zabírají místo, takže plocha CMOS čipu není citlivá celá. [10]

Fotodiody

Fotodioda je elektrotechnická součástka, která funguje obdobně jako běžná dioda, avšak při jejím osvětlení dochází na jejím povrchu k uvolnění elektronů díky vnitřnímu fotoelektrickému jevu. V závěrném směru se tedy stává fotodioda více propustná. [11]

Fotonásobič

Fotonásobič je vakuová elektronka. Obsahuje fotokatodu, vstupní elektronové optiky, systém dynod a anodu. Na fotocitlivou katodu dopadá foton a díky fotoelektrickému jevu uvolní elektron. Elektron dopadá na první dynodu (dynoda má kladný potenciál) a uvolňuje z ní další elektrony (tzv. sekundární emise). Ty jsou přitahovány k dalším dynodám, jelikož potenciály dynod se od fotokatody směrem k anodě zvyšuje a proces se opakuje. Konečně elektrony dopadají na anodu a vzniká výstupní signál. Na obrázku 5.3 je vyobrazeno schéma fotonásobiče. [12, 13]



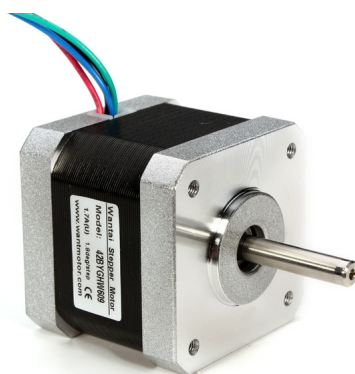
Obr. 5.3: Schéma fotonásobiče [37]

5.1.2 Posuvný systém

Posuv mikrotitrační destičky většinou probíhá ve třech na sebe kolmých osách. Dvě slouží k posuvu vzorku v horizontální rovině. Tento posuv zajišťuje umístění konkrétní jamky pod snímací detektor. Pohyb ve vertikální rovině zajišťuje úpravu hloubky ostrosti detektoru vzhledem k aktuálnímu vzorku. K pohybu je využíváno krokových motorů (obr. 5.4) nebo servomotorů (obr. 5.5).

Krokové motory

Krokové motory jsou napájené stejnosměrným proudem. Stator je tvořen z dvojic protilehle uložených cívek, jejichž póly mají stejné orientace. Pohyb rotoru probíhá vždy jen do stabilní polohy v tzv. krocích. Tento princip umožňuje otáčet motorem s vysokou přesností do zadaných poloh. [14]



Obr. 5.4: Krokový motor [36]

Servomotory

Servomotory jsou motory, jež umožňují natočení osy hřídele do přesně dané polohy. Jedná se v podstatě o klasický elektromotor, jehož poloha je kontrolována snímačem. Součástí je také řídicí elektronika, jež podle vstupních PWM pulsů vypočítá zadanou pozici a otáčí elektromotorem požadovaným směrem. Jednoduchým snímačem polohy může být potenciometr. Převodovkou je přenášen pohyb motoru nejen na hřídel, ale i na jezdcu kontrolního potenciometru. Otáčením potenciometru se zvyšuje jeho odpor, který je měřen řídicí elektronikou. Jelikož se potenciometr může otáčet pouze v určitém rozsahu, je často limitované i maximální natočení osy.



Obr. 5.5: Servomotor [41]

5.1.3 Osvit

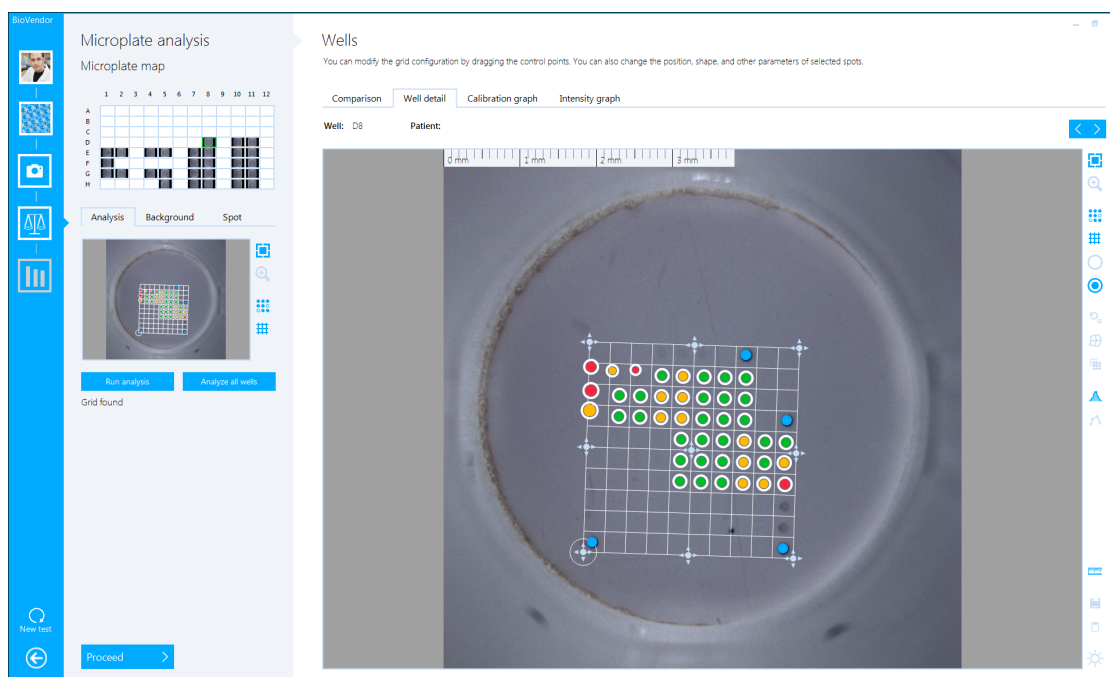
Paleta používaných osvitů v microarray analyzátoch je velice pestrá. Volba osvitu v přístroji se odvíjí zejména od fyzikální metody, již daný přístroj využívá a typem zkoumaného analytu ve vzorku. Běžné jsou xenonové výbojky, LED panely nebo lasery pro různé účely. Velice časté je taktéž použití monochromátorů, pomocí kterých se získává světlo o konkrétní vlnové délce (resp. úzký interval vlnových délek). Princip spočívá v rozdělení světla na optické mřížce či hranolu a poté pomocí optického systému fokusace úzkého intervalu vlnových délek na výstupní štěrbinu. Monochromatické světlo hraje důležitou roli např. ve spektrofotometrii při získávání absorpčních spekter.

5.1.4 Počítač a doplňující elektronika

Analyzátoy se standardně ovládají pomocí počítače nebo rozhraní implementovaného přímo do přístroje. Lze nastavovat konkrétní průběhy testů, konfigurace, pozice se vzorky na mikrotitrační destičce a parametry vyhodnocení. Doplňující elektronika sestává především z řídicí jednotky posuvného systému, ovladače k měničům krokových motorů, přídavné kamery pro čtení čárových kódů, přídavné osvity apod.

5.2 Analýza microarray

Jak již bylo řečeno, detektorem v analyzátorech microarray jsou nejčastěji kamery (CCD a CMOS čipy). Základním kamenem při analýze microarray je zpracování obrazu. Po dokončení snímání vzorků musí být každý ze snímků vyhodnocen. Při zpracování obrazu jsou spoty na nosiči rozděleny pomyslnou mřížkou a algoritmus rozhoduje spot po spotu, zda reagovaly. Podle toho, které spoty reagovaly, je vyhodnocen diagnostický výsledek testu. Na obrázku (5.6) je vyobrazen výsledek analýzy jamky mikrotitrační destičky. Body s reagensii, jež změnily své chemické vlastnosti, jsou zvýrazněny různými barvami. Obrázek je pořízen z aplikace společnosti BioVendor.



Obr. 5.6: Výsledek analýzy jedné jamky mikrotitrační destičky

6 Jakost, kontrola jakosti a seřizování přístrojů

Kvalita výrobku se odvíjí od velkého množství parametrů, které jsou přímo závislé na zpracování výrobku. Obecně však všechny tyto parametry (a mnoho dalších) sjednocuje pojem *jakost*. Tato práce je s problematikou jakosti velice úzce spjata. Seřizování přístroje je jednou z mnoha metod pro korigování výrobních nedokonalostí a zajištění popř. zvýšení jakosti. Kapitola blíže popíše jakost a její řízení. Detailněji se poté zaměří na aspekty jakosti, které popisují přesnost a spolehlivost analyzačních přístrojů a jejich korekce.

6.1 Definice jakosti

Jakost (také kvalita) je pojem, jenž podává informaci o výrobcích či službách; částečně také zahrnuje informace o procesech týkajících se těchto výrobků či služeb. Existuje pro ni mnoho definic. Níže jsou uvedené ty, které se dotýkají tématu této práce. Jakost je:

- způsobilost pro užití,
- shoda s požadavky,
- bezpečnost.

K dosažení obecných principů systému řízení jakosti se vztahuje norma ISO 9001. Přístroj používaný v praktické části je hodnocen jako zdravotnický prostředek; aplikuje se rovněž norma ISO 13485. [26, 27]

6.2 Parametry jakosti

Při stanovování jakosti se posuzují zejména požadavky zákazníků využívajících produkt či službu a legislativa týkající se produktu.

Mezi požadavky zákazníků (neboli parametry jakosti) můžeme zařadit nezávadnost, ovladatelnost, opravitelnost, spolehlivost, funkčnost, bezpečnost apod. V kontextu této práce jsou důležité pojmy funkčnost a spolehlivost. Funkčnost je požadavek, aby přístroj konal správně funkci, pro kterou byl navržen; spolehlivost je schopnost výrobku plnit veškeré funkce v jakémkoli okamžiku, aniž by nastala závada.

Aby se přístroj mohl dostat na trh Evropského hospodářského prostoru, musí být označen značkou nezávadnosti CE. Pokud produkt obdrží CE značku, znamená to, že výrobce zajistil bezpečnost uživatele produktu, ochranu životního prostředí a dodržel všechny náležité směrnice. Mezi takové směrnice patří např. směrnice *The Low Voltage Directive - elektrická bezpečnost (LVD)*, *Machinery directive - směrnice pro strojní zařízení (MD)*

či *Restriction of the use of certain Hazardous Substances in electrical and electronic equipment - omezení v použití zakázaných substancí v elektrických a elektronických zařízeních* (RoHS). Produkt taktéž musí projít zkouškou *Electromagnetic compatibility - elektromagnetická kompatibilita* (EMC).

Směrnice LVD pro přístroje nízkého napětí zaručuje vysokou ochranu uživatele při užívání elektrického zařízení, a to i v případě poruchy. Směrnice RoHS zase omezuje užití nebezpečných látek v elektrických zařízeních. Zkouška EMC zaručuje, že přístroj je schopen správně fungovat i v přítomnosti jiných zdrojů elektromagnetických signálů a zároveň samo nevysílá rušivé signály. [42, 43]

6.3 Jakost biochemických analyzátorů

Vzhledem k tématu práce bude dále přiblížena jakost biochemických analyzátorů. Přesnost analyzátorů se dá posuzovat z hlediska variability výsledků měření, a to konkrétně metrologii¹ takového zařízení a reprodukovatelnosti výsledků v rámci jednoho zařízení. K zajištění takovýchto vlastností přístroje slouží různé procesy např. seřízení. Předcházet nepřesnostem měření se dá pomocí pravidelných kontrol.

Biochemické analyzátoři se často klasifikují jako in-vitro diagnostické zdravotnické prostředky a musí splňovat příslušnou směrnici EU (v současnosti platná směrnice 98/79/ES o diagnostických zdravotnických prostředcích in vitro). Zaměřuje se především na zvýšenou ochranu uživatelů výrobku při běžném provozu i poruše. [44]

6.4 Kontrola jakosti a seřizování

Pro zajištění přesných výsledků z analýzy se kromě vhodných komponent a dodržení technologického postupu výroby uplatňuje seřizování přístroje a korekce výrobních nedokonalostí. Cílem je dosáhnout opakovatelnosti shody při výrobě libovolného počtu kusů daných přístrojů. Tato seřizování jsou kritickými a nepostradatelnými procesy v rámci zajišťování jakosti.

V zásadě se procesy mezi sebou mohou lišit rozsahem seřízení, kvalifikací osob provádějících seřízení nebo frekvencemi provádění takovýchto procesů. Po výrobě (popř. instalaci zařízení a každé větší opravě) následuje tzv. předávací zkouška. Jejím úkolem je ověření, popř. seřízení přístroje tak, aby jeho detekční parametry odpovídaly hodnotám garantovaným výrobcem. Ve vhodných časových intervalech poté následují provozní zkoušky. Jejich úkolem je sledovat, zda jsou měřené hodnoty v toleranci stanovené výrobcem a odhadovat krátkodobé změny ve výsledcích měření a dlouhodobé trendy v případných krátkodobých změnách.

¹vědní obor zabývající se vztahem mezi měřenou a skutečnou hodnotou dané veličiny

Na základě výsledků zmíněných zkoušek (resp. k předcházení nepříznivých výsledků dodavatelem) se provádějí korekční procesy zahrnující kalibrace (v případě měřících přístrojů) a seřizování přístrojů, jež provádí kvalifikovaná osoba. K případné korekci chybných výsledků se provádějí méně náročné seřizovací procesy, které mohou provádět i osoby bez kvalifikace (např. laboranti využívající přístroj). Po větších opravách je nutné přístroj znova seřídít servisním technikem nebo kvalifikovaným pracovníkem dodavatele.

6.5 Instalační, operační a procesní kvalifikace

Instalační kvalifikace, operační kvalifikace a procesní kvalifikace jsou další z důležitých pojmů vztahující se k tématu této práce.

Instalační kvalifikace je dokumentované ověření toho, že technologie a zařízení (stroje, měřící zařízení, podpůrné systémy, pracovní prostory) odpovídají schválenému návrhu a doporučením výrobce.

Operační kvalifikace je dokumentované ověření toho, že technologie a zařízení pracují tak, jak je zamýšleno (v kontextu této práce např. bezvadně funguje pojezd, osvit zařízení je vhodně seřizen, kamera bezchybně snímá analyzované vzorky a software vše správně vyhodnocuje).

Obě procedury se provádějí při instalaci nebo modifikaci zařízení. [45]

Ve vztahu ke dvěma předešlým pojmům se uvádí také pojem *procesní kvalifikace*. Provádí se po úspěšném dokončení instalační a operační kvalifikace (v některých případech zároveň s operační kvalifikací). Úkolem procesní kvalifikace je dokumentovaně ověřit, že prostory, systémy a zařízení v podobě, v níž jsou propojeny, mohou účinně a reprodukovatelně pracovat podle schváleného výrobního postupu a specifikací výrobku. [46]

6.6 Interkoeficient a intrakoeficient variance

V problematice imunologických testů se v mnoha publikacích můžeme setkat s pojmy interkoeficient a intrakoeficient variance. Intrakoeficient variance označuje směrodatnou odchylku výsledků měření z jednoho vzorku. Interkoeficient variance označuje směrodatnou odchylku výsledků měření z více vzorků v rámci jednoho testu. Oba tyto pojmy můžeme zařadit mezi důležité parametry jakosti biochemických analyzátorů (resp. testů pro ně navržených). Pro dosažení vysoké jakosti produktu musí být obě tyto veličiny co nejnižší.

7 .NET Framework

Přístroj může být sestaven ze sebelepších komponent, ale bez kvalitního softwaru je bezcenný. Následující kapitola přiblíží platformu .NET. Dále se bude věnovat modernímu programovacímu jazyku C# a část kapitoly je také věnována jazyku C++/CLI. Jazyk C# bude využit při tvorbě praktické části práce. V práci bude také využit pro zpracování obrazu čistý jazyk C++ bez platformy .NET.

7.1 Součásti .NET Framework

.NET Framework je softwarová platforma pro vývoj aplikací pro OS Windows, ale i mobilní zařízení či web. Snad nejdůležitější částí platformy .NET jsou knihovny. Jelikož je dáva k dispozici společnost Microsoft, jsou knihovny moderní a aktuální. Jsou také uzpůsobeny pro správnou funkci na OS Windows a dalších produktech společnosti Microsoft. Pro důkladnější přiblížení .NET Frameworku je nutné vysvětlit pojmy: programovací jazyk, Visual Studio a virtuální stroj (CLR). [31, 32]

7.1.1 Jazyk a virtuální stroj

Aplikace vyvíjené s použitím .NET Frameworku nejsou kompilované ani interpretované. Kódy z různých programovacích jazyků (např. Visual Basic .NET, C# nebo C++/CLI) jsou přeloženy do tzv. CIL společně s datovými soubory aplikace. Výsledkem je soubor s příponou .exe či .dll. Při spuštění jsou za pomoci virtuálního stroje (CLR) kompilovány do strojového kódu jen ty části kódu, které jsou potřebné. Kromě toho se kompilované části cachují (ukládají do mezipaměti), takže je není nutné kompilovat vícekrát a dochází ke zrychlení aplikace oproti klasickým interpretovaným jazykům.

C++/CLI

C++/CLI je nástavba programovacího jazyku C++ od společnosti Microsoft. Jelikož vychází ze standardního C++, je objektově orientovaný. Snaží se o zjednodušení klasické C++ syntaxe a uzpůsobení pro práci s platformou .NET Framework. Z toho plyne, že kód je přeložený do CIL a je spouštěn pomocí virtuálního stroje.

C#

C# (jméno má odkazovat na notu C o půl tón zvýšenou - cis) je vysokoúrovňový objektově orientovaný programovací jazyk vyvíjený společností Microsoft pro platformu .NET Framework. Je založen na jazycích C++ a Java, jeho syntaxe tedy

vychází z tradičního C. Jazyk je navržen moderně, tak aby byly znalosti z C nebo C++ přenositelné, vývoj aplikací jednoduchý a aplikace byly univerzálně použitelné. Níže (7.1) si můžete prohlédnout tradiční „Hello World“ program v jazyce C#. V jazyce C# bylo vytvářeno prostředí aplikace v praktické části práce. [28, 29, 30]

Výpis 7.1: Program HelloWorld v jazyce C#

```
// A Hello World! program in C#.           1
using System;                               2
namespace HelloWorld                         3
{                                             4
    class Hello                              5
    {                                         6
        static void Main()                  7
        {                                     8
            Console.WriteLine("Hello World!"); 9
            // Keep the console window open in debug mode. 10
            Console.WriteLine("Press any key to exit."); 11
            Console.ReadKey();               12
        }                                     13
    }                                         14
}                                             15
```

7.1.2 Microsoft Visual Studio

Visual Studio společnosti Microsoft je profesionální softwarové prostředí využívané pro vývoj aplikací .NET (kromě jiného). Je oblíbeno i mezi laiky. Velkou výhodou skýtají funkce IntelliSense, které programátorovi „napovídají“ při psaní kódu. Ve Visual Studiu byla vytvářena praktická část této práce.

8 Array Reader C-series

Array Reader C-series je kolorimetrický analyzátor společnosti BioVendor Instruments. Praktická část práce byla vyvíjena konkrétně pro tento přístroj, proto budou v následující kapitole přiblíženy jeho parametry a proces povýrobního seřízení. Závěr kapitoly se bude věnovat porovnání s konkurenčním přístrojem CLAIR od společnosti Sensovation.

8.1 Charakteristika zařízení

Array Reader C-series (obr. 8.1) je vytvořený pro analýzu microarray. Umožňuje jak analýzu jednotlivých reagensů na/v nosiči vzorků, tak analýzu nosiče jako celku (lze rovněž vybrat konkrétní místa, jež mají být analyzována; např. u mikrotitračních destiček konkrétní jamky).

Možnosti jeho uplatnění jsou velmi pestré, a to např. v mikrobiologii, imunologii, sérologii, farmacii, planární imunometrii (do které se řadí i zmíněná ELISA) a mnoho dalších. Podporuje mikrotitrační destičky o 96 jamkách a dokáže je analyzovat do dvou minut.

Optické zpracování vzorku zajišťuje CMOS digitální kamera. Pro analýzu testů se uplatňují principy zpracování obrazu. Výsledkem analýzy je posouzení, zda vzorek reagoval s reagensy na dně mikrotitračních destiček na základě změny intenzity barev či velikosti spotů. Stanovit kvantitativní hodnoty z měření přístroj dokáže, pokud nosič vzorků obsahuje kalibrační spoty (spoty o známé koncentraci, ze kterých se získá kalibrační křivka).

Vzorky osvětlují dva statické LED panely. Jeden ze spodní strany, druhý z horní strany. Je dostupné i řešení se spodní kamerou ke čtení čárových kódů nebo QR kódů, jimiž mohou být označeny mikrotitrační destičky, stripy na destičce nebo jednotlivé jamky. Spolu s přístrojem je dodáván i intuitivní software sloužící k jeho ovládání. [9]

8.2 Výstupní kontrola Array Readeru C-series

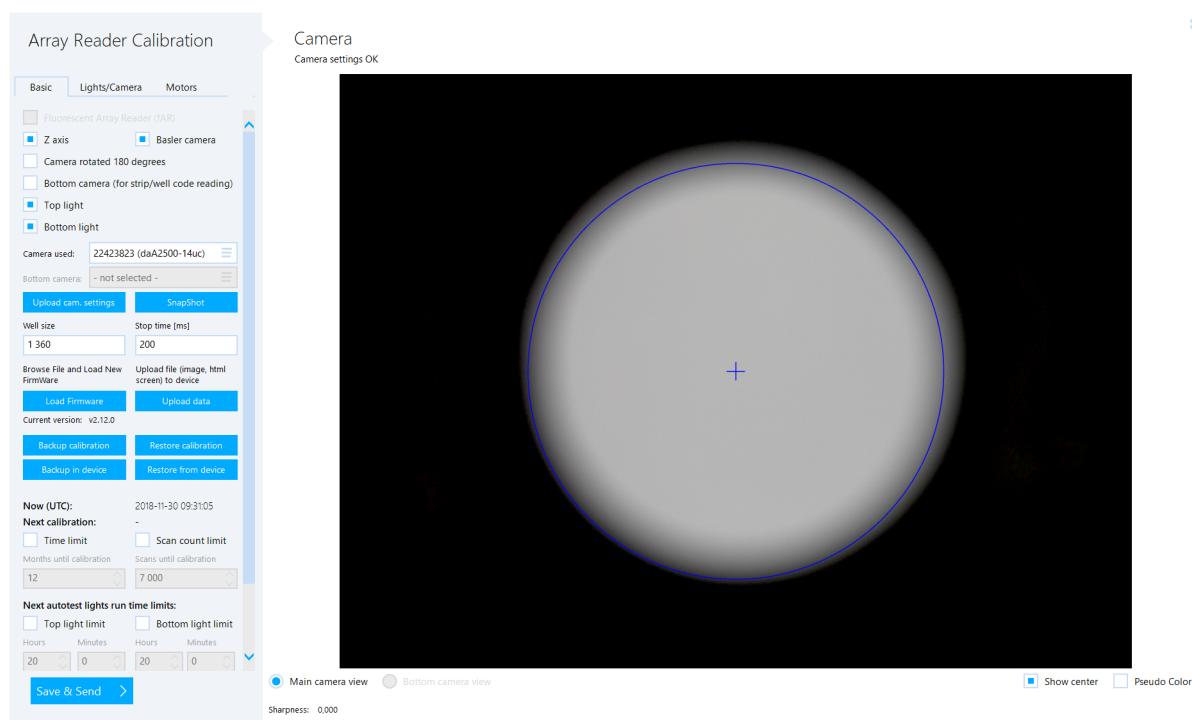
Předtím, než přístroj obdrží zákazník, je nezbytné jej zkontrolovat a provést povýrobní seřízení, aby se korigovaly výrobní nedokonalosti. Tyto procesy provádí zaměstnanec postupující podle připraveného protokolu a využívá při tom aplikaci pro seřízení, která je součástí dodávaného softwaru.



Obr. 8.1: Array Reader C-series [35]

8.2.1 Aplikace pro seřizování

Aplikace je rozdělena na levou a pravou část (obr. 8.2). Na pravé části se nachází především výstup z kamery. Levá část se skládá ze tří panelů: *Basic*, *Lights/Camera* a *Motors*. Pomocí každého z panelů se nastavují různé části přístroje. V průběhu seřizování se pracovník pohybuje mezi všemi panely.



Obr. 8.2: Aplikace pro seřizování

Panel *Basic*

Panel (obr. 8.3) slouží k zadání obecných informací o přístroji, správu konfiguračních souborů a nastavení seřizovacích limitů.

Obecné informace o analyzátoru se zadávají pomocí zaškrtačkových tlačítek v horní části panelu. Většina z nich (např. zda je kamera otočená o 180 ° nebo jestli je přítomný spodní i horní osvit) je již zastaralá a nová aplikace je proto nemusí obsahovat.

Pod tlačítka jsou dvě komponenty pro vybrání použitých kamer, tlačítko pro nahrání konfiguračních souborů do kamer a tlačítko pro pořízení snímku z kamery.

Dále se pomocí textových polí může zadat velikost jamky (taktéž již zastaralá, jelikož všechny vzorky mají standardizovanou velikost) a čas nutný k analýze vzorku.

Níže se nachází série tlačítek, která slouží k nahrání firmwaru do přístroje, poslání jakéhokoli souboru do přístroje, vytvoření továrního nastavení, návrat k továrnímu nastavení nebo jinému seřízení a záloha aktuálního seřízení.

Pod tlačítka se nachází komponenta, ve které se definují limity přístroje před nutností dalšího seřízení.

Fluorescent Array Reader (FAR)

Z axis Basler camera

Camera rotated 180 degrees

Bottom camera (for strip/well code reading)

Top light

Bottom light

Camera used: 22423823 (daA2500-14uc)

Bottom camera: - not selected -

Upload cam. settings SnapShot

Well size Stop time [ms]

1 360 200

Browse File and Load New FirmWare Upload file (image, html screen) to device

Load Firmware Upload data

Current version: v2.12.0

Backup calibration Restore calibration

Backup in device Restore from device

Now (UTC): 2018-11-30 09:31:05

Next calibration: -

Time limit Scan count limit

Months until calibration Scans until calibration

12 7 000

Next autotest lights run time limits:

Top light limit Bottom light limit

Hours Minutes Hours Minutes

20 0 20 0

Save & Send >

Obr. 8.3: Panel *Basic*

Panel *Lights/Camera*

Panel *Lights/Camera* (obr. 8.5) se využívá pro seřízení osvitů a ovládání kamery. Podle zvolené osvitové podmínky v horní části se uzpůsobuje zbytek panelu.

Pomocí textových polí lze měnit expozici a digitální zesílení kamery. Tlačítka pod textovými poli slouží k nastavení vyvážení bílé barvy kamery, pozicování na jamku *A2* a export konfiguračního souboru kamery. Expozice a zisk spodní kamery se nastavuje ve nejspodnější části panelu, která je přístupná pouze pokud je přístroj spodní kamerou vybaven.

Seřízení lze provádět jak s plastovým tak i kovovým fantomem¹. Pracovník musí zaškrtnout, který fantom použije, popř. vybrat i typ fantomu. Na obrázku 8.4 je zmíněný fantom s červeně vyznačenými referenčními pozicemi, o kterých bude řeč dále v této kapitole.

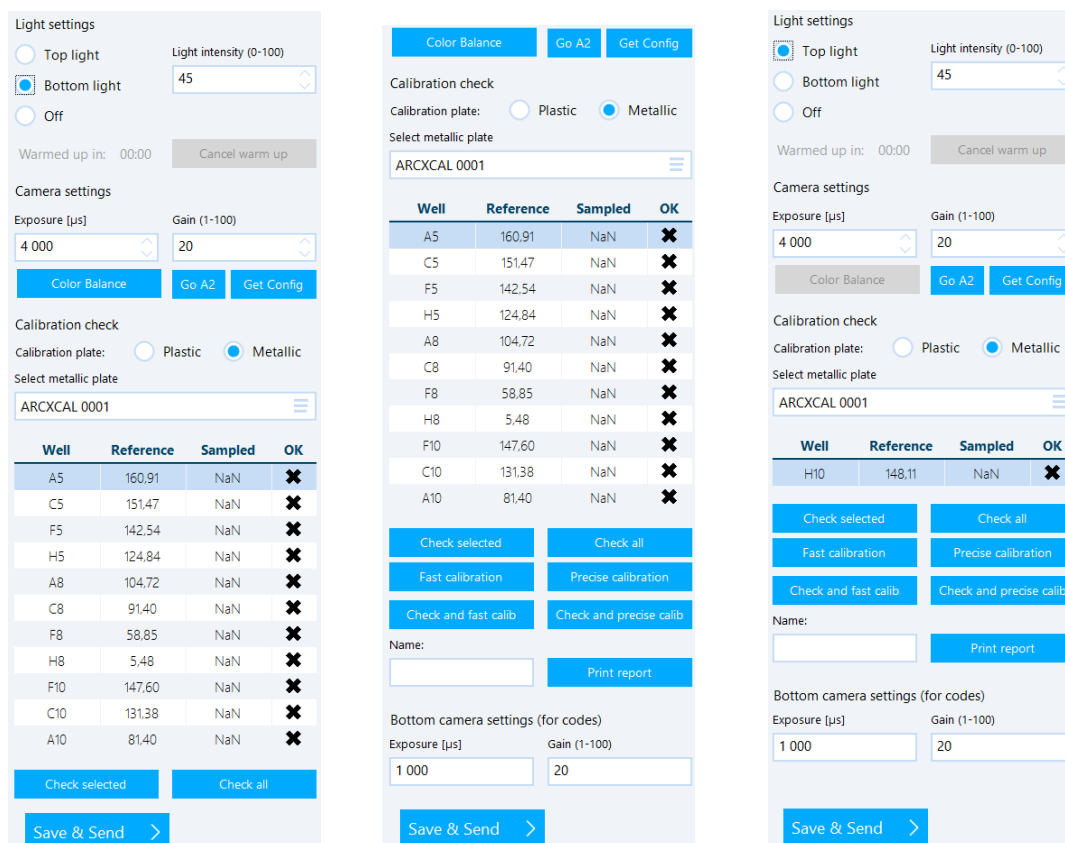
Různé formy seřízení (rychlé seřízení oproti přesnému, změření hodnot osvitů, seřízení osvitů či změření a seřízení osvitů) se spouští pomocí tlačítek ve spodní části panelu.

Na panelu jsou všechny komponenty aktuální a musí být součástí i nové verze aplikace.



Obr. 8.4: Fantom mikrotitrační destičky s vyznačenými referenčními pozicemi

¹neprůhledná deska s otvory o velikosti standardní mikrotitrační destičky; v některých otvorech jsou umístěny optické filtry využívané při seřízení osvitů



Obr. 8.5: Panel *Lights/Camera* - seřizování dolního osvětlení (vlevo a uprostřed), seřizování horního osvětlení (vpravo)

Panel Motors

Polohování vozíku se ovládá na panelu *Motors*. Definují se na něm krajní polohy a další významné pozice.

V horní části panelu (obr. 8.7) je zobrazena aktuální pozice vozíku v ose x a y (rozvržení os je na obr. 8.6). Pomocí tlačítek se nastavuje cílová pozice při vysunutí vozíku nebo polohy referenčních jamek. Tlačítkem *Set* se nastaví aktuální poloha jako příslušná pozice a pomocí tlačítka *Go* zajede vozík na nastavenou polohu.

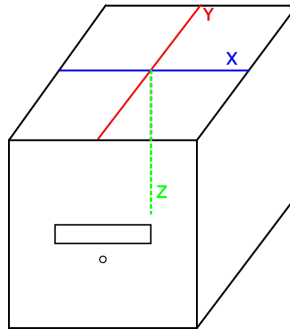
Další část komponenty slouží k nastavování osy z . Pracovník může pozicovat jakoukoli jamku pod kameru, zaostřit na jamku, pohybovat s kamerou v ose z podle počtu zadaných kroků či zajet s kamerou na nulovou pozici.

Další komponenta slouží k seřizování spodní kamery pro čtení QR kódů nebo čárových kódů. Zpřístupňuje se pokud je přístroj spodní kamerou vybaven. Nastavuje se v ní výchozí pozice vozíku pro čtení a velikost a poloha čtené oblasti s kódem. Obrázek načteného kódu a text v něm uložený se zobrazuje v bílém okně vedle tlačítka *Set Up QR*.

V dolní části panelu se nastavují krajní polohy vozíku a vzdálenost vozíku od

dvířek, pokud je vysunutý ven z analyzátoru.

Všechny bloky tohoto panelu jsou nezbytné pro správné seřízení analyzátoru.



Obr. 8.6: Rozvržení os v Array Readeru C-Series

Actual X: 0	Y: 0	Go End	
Step X: 54000	Set Step X	Go Step X	
Step Y: 53000	Set Step Y	Go Step Y	
Insert X: 10046 Insert Y: 60530	Set Insert	Insert(A1)	
Eject X: 36095 Eject Y: 147641	Set Eject	Eject	
A12 X: 9587 A12 Y: 7815	Set A12	Go A12	
H1 X: 63784 H1 Y: 60242	Set H1	Go H1	
	Go to plate position		
Z position	Go Focus	Focus	
0	Go Z	Go End Z	
Ref. Z: 750000	Set Ref Z	Go Ref Z	
Positions for bottom camera			
Insert Y: 80000	Set Insert	Insert	
	Prev	Next	
X	Y	Width	Height
165	1 400	2 320	280
Test QR			🔍
Set up QR			
Save & Send >			

Z position	GO FOCUS	Focus	
0	Go Z	Go End Z	
Ref. Z: 750000	Set Ref Z	Go Ref Z	
Positions for bottom camera			
Insert Y: 80000	Set Insert	Insert	
	Prev	Next	
X	Y	Width	Height
165	1 400	2 320	280
Test QR			🔍
Set up QR			
Minimal X: 4016	Set Min X	Go Min X	
Minimal Y: 4056	Set Min Y	Go Min Y	
Maximal X: 65592	Set Max X	Go Max X	
Maximal Y: 147641	Set Max Y	Go Max Y	
Middle Y: 57961	Set Mid Y	Go Mid Y	
Door X0: 31355	Set Door X0	Go Door X0	
Door X1: 40267	Set Door X1	Go Door X1	
Save & Send >			

Obr. 8.7: Panel *Motors* - první část (vlevo), druhá část (vpravo)

8.2.2 Příprava analyzátoru na seřízení

Každý analyzátor je kontrolován z hlediska správnosti montáže, provedení mechanických a elektronických dílů a jejich zapojení již v průběhu výroby. Při seřízení je nutné jej překontrolovat a seřídít z hlediska komunikace se softwarem. Nejprve se kontrolují bezpečnostní značení a fixace elektrických komponent. Poté se přístroj připojuje k počítači.

8.2.3 Seřízení analyzátoru

Nastavení analyzátoru probíhá pomocí zmíněné aplikace podle následujícího postupu:

- Prvním krokem je úprava nastavení aplikace pro konkrétní stroj (nastaví se, zda je vybaven spodní kamerou, uvedou se typy kamer apod.). Dále pracovník do kamery (popř. kamer) nahraje konfigurační soubory.
- Následuje nastavení krajních poloh pro pohyb motorů (vozíček s mikrotitrační destičkou nesmí vyjet z pojezdu, narazit do dvířek atd.). Krajní polohy vozíku se vzorkem jsou hlídány koncovými spínači.
- Pracovník zkontroluje čistotu objektivu a natočení horní kamery (nemusí být usazena dokonale rovně) a případně ji vyčistí a/nebo koriguje do správné pozice.
- Dalším krokem je nastavení pozice, kam najede vozík se vzorkem po vložení mikrotitrační destičky. Vozík by měl pozicovat pod kameru první jamku mikrotitrační destičky (jamka se souřadnicí A1).
- Po nastavení této pozice přichází na řadu zaostření kamery, které kromě pohybu kamery v ose z často vyžaduje také manuální otáčení objektivem. Poté se nastavují další důležité pozice vozíku s destičkou – minimální a maximální pozice před koncovými spínači apod.
- Pracovník zakrytuje zařízení a zapne seřízení osvitů. To je ze značné míry automatizované a po dokončení seřízení tedy pouze kontroluje výsledky. Seřizuje se jak spodní, tak horní osvit.
- Pokud přístroj obsahuje přídatnou kameru pro čtení QR kódu, je potřeba ji zaostřit a nastavit správný posun po jednotlivých řádcích mikrotitrační destičky (tzv. stripech).
- Konečně pracovník nechá provést autodiagnostický test. Pokud proběhne bez problému, zálohuje nastavení a archivuje jej.

8.2.4 Časová náročnost seřízení

Po změření času potřebného k seřízení za pomoci osoby rutinně provádějící povýrobní seřízení bylo zjištěno, že celý proces trvá asi 1 hodinu a 45 minut. Časy některých seřizovacích úkonů jsou uvedeny v tabulce (tab. 8.1) níže.

Úkony uvedené v tabulce jsou závislé na osobě provádějící seřízení a není možné je nijak časově zefektivnit. Prostor pro optimalizaci se tedy nachází především ve struktuře aplikace pro seřízení. Novou aplikaci by bylo vhodné koncipovat tak, aby na sebe úkony navazovaly a tam, kde je to možné, byly zautomatizované.

Tab. 8.1: Vybrané seřizovací úkony

Popis úkonu	Časová náročnost
Uložení nastavení do přístroje a restartování aplikace	5 min
Kontrola čistoty objektivu popř. jeho čištění	5-25 min
Manuální ostření kamery a fixace tavnou pistolí	10-15 min ²
Automatické seřízení spodního osvitu	35 min
Automatické seřízení horního osvitu	5 min
Autodiagnostický test	30 min

8.2.5 Problematické části stávajícího řešení aplikace pro seřízení

Stávající aplikace pro seřízení byla vyvíjena současně s přístrojem a komponenty do ní byly přidávány postupně. Tomu odpovídá i její struktura. Stejně tak je obtěžkána mnoha částmi, které se již nepoužívají. Mezi zásadní nedostatky aplikace patří:

- Posloupnost jednotlivých kroků je zřejmá pouze na základě dostupného manuálu k seřízení.
- Na akce, které musí uživatel provést mimo aplikaci (fixace objektivu kamery, korekce natočení kamery apod.), není uživatel žádným způsobem upozorněn.
- Pokud obsluha aplikace vynechá nebo provede některý z kroků nedostatečně, je nutné celý postup opakovat, což prodlužuje výslednou dobu finálního seřízení. Chybné seřízení se může navíc projevit až při provedení závěrečného autodiagnostického testu.
- Před seřízením určité oblasti přístroje je nutné provést sérii kroků (dojet vozíkem na určitou pozici, zvýšit expozici kamery apod.).
- Pracovník je nucen přecházet z jednoho panelu na druhý.

Návrh nové aplikace se snaží těmito nedostatky vyhnout.

²Při fixaci objektivu se někdy objektivem pootočí, čímž dojde k rozostření a proces je nutné opakovat. Čas se tak významně prodlužuje.

8.3 Srovnání s konkurenčním microarray analyzáto- rem Sensovation CLAIR

Na trhu jsou v současné chvíli dostupné dva kolorimetrické analyzátory pro vyhodnocování microarray testů. Kromě Array Readeru C-series společnosti BioVendor je na trhu ještě microarray analyzátor CLAIR od společnosti Sensovation.

Zásadním hardwarovým rozdílem přístrojů je především rozlišení použitých kamer (kamera Array Readeru C-series má rozlišení 5 MPx; Sensovation CLAIR pouze 3.8 MPx) a rychlejší přenos dat z kamery pomocí portu USB 3.0 v případě Array Readeru C-series. Společnost BioVendor rovněž nabízí možnost vybavení svého přístroje spodní kamerou pro čtení QR kódů a čárových kódů.

Oproti Sensovationu CLAIR vyniká Array Reader C-series také softwarem pro zpracování vzorků. Umí např. stanovit koncentraci pomocí kalibrační křivky nebo zpracovávat barevný obraz. Velkou výhodou je také schopnost Array Readeru C-series zpracovávat transparentní i netransparentní vzorky; CLAIR je dostupný ve dvou variantách, každá z nich je určena pouze pro transparentní nebo pouze pro netransparentní vzorky.

Detailnější srovnání jejich důležitých parametrů je vypracováno v tabulce v přílohách práce (A).

9 Návrh aktualizace aplikace pro seřízení přístroje Array Readeru C-series

Předmětem praktické části práce je aktualizace stávající aplikace pro seřízení analyzátoru Array Reader C-series. V kapitole je vytyčeno zadání nové aplikace a následně rozebráno zpracování tohoto zadání. Nyní používané postupy k seřízení budou upraveny a přispějí k optimalizaci činností při ukončování výroby.

9.1 Zadání aplikace

Zadání bylo vytvořeno na základě osobních zkušeností se seřízením po proškolení a provedení série seřizovacích procesů. Další četné připomínky a upozornění na problematické části pocházely od osob rutinně provádějících seřízení a vývojářů původní aplikace. Aktualizovaná aplikace by měla splňovat následující zadání:

- Aplikace bude po formální a vizuální stránce odpovídat standardům společnosti BioVendor.
- Postup pro provedení seřízení je součástí výstupní kontroly, a formálně bude i nadále zpracován jako dokument, dojde však k celkovému zpřehlednění, což usnadní orientaci pracovníka při provádění činností.
- Aplikace bude reflektovat kroky, jež je nutné provést ze strany uživatele manuálně.
- Seřízení bezpečných oblastí, vyhledání výšky kamery nad vzorkem a detekce referenčních pozic budou automatizovány a pracovník pouze zkontroluje výsledek popř. provede menší korekce. Automatizační procesy budou z velké části využívat zpracování obrazu.
- Aplikace bude z hlediska softwarového provedení využívat nejnovější komponenty společnosti BioVendor.
- Aplikace bude po každém kroku automaticky ukládat současná nastavení, uživatel uložení pouze schvaluje, nemusí se jím sám zaobírat.
- Do každého analyzátoru se bude ukládat historie provedených seřízení. Tato historie může posloužit např. při poruše zařízení ke správné diagnostice chyby.

9.2 Implementace aplikace

Aplikace pro seřízení analyzátoru bude zakomponována do softwaru společnosti BioVendor, který je dodáván společně s analyzátor. Software je vyvíjený v jazyce C# a v tomto jazyce tedy proběhne i implementace aplikace v praktické části práce.

Zpracování obrazu v projektu bude realizováno pomocí externích knihoven. Tyto knihovny budou vytvořené v programovacím jazyce C++. Využije se taktéž knihoven OpenCV, jež jsou určeny pro zpracování obrazu v reálném čase.

Po každém provedeném seřízení je nutné vyexportovat protokol s údaji o jeho průběhu. Protokol je vygenerován automaticky do souboru s příponou *.rdlc*, jež je strukturovaný pomocí jazyka XML. Do zařazení je taktéž nutné nahrát nová nastavení. Ta jsou uložena přímo v souboru XML. Do protokolu i souboru s nastavením bude nutné přidat některé informace, které původně nebyly jejich součástí, takže v průběhu realizace bude jazyk XML použit.

9.3 Struktura aplikace

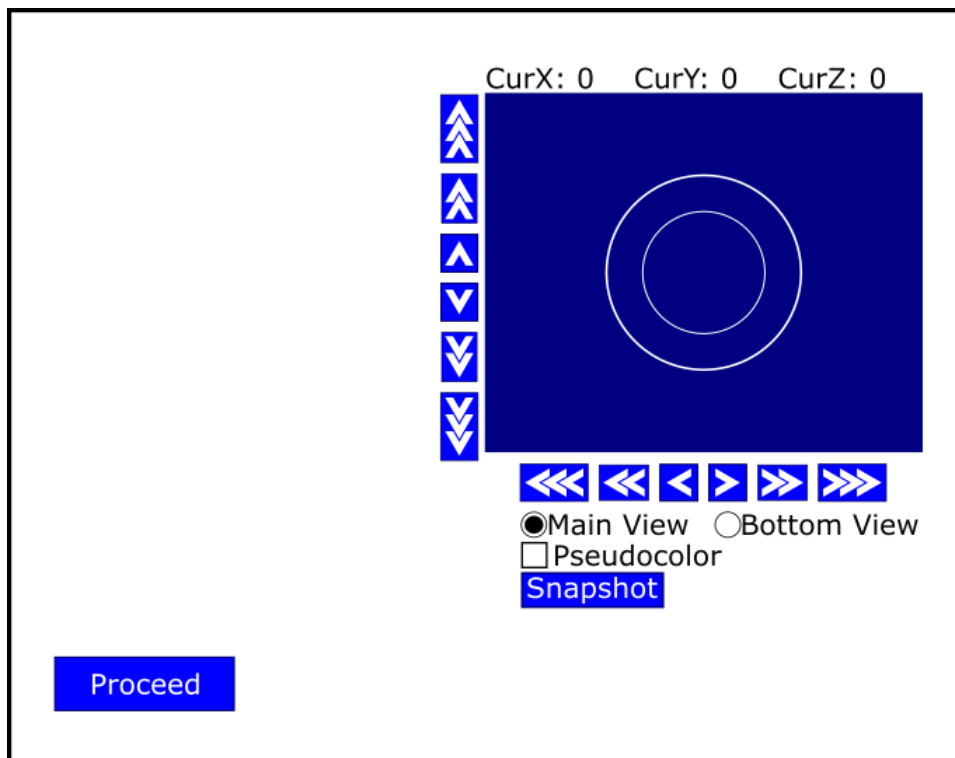
Některé části budou převzaty z původní aplikace, dojde však ke kompletní restrukturalizaci, implementují se nové komponenty a u některých kroků se uplatní procesy automatizace.

Nová aplikace je rozvržena na sedm za sebou jdoucích obrazovek. Na každé z nich se nachází všechny komponenty pro nastavení určité části přístroje (motory posuvníku, motor kamery, osvit apod.). Uživatel tedy nebude nucen mezi jednotlivými obrazovkami přecházet.

V původní aplikaci před seřízením určité části přístroje, musel pracovník vykonat sérii předcházejících kroků (např. najet vozíkem se vzorkem na určitou pozici, zaostřit kameru apod.). Návrh aplikace byl koncipován tak, aby se tyto kroky odehrávaly na pozadí při otevření dané obrazovky.

9.3.1 Společné funkce všech obrazovek

Na společné části všech obrazovek (obr. 9.1) se nachází především obraz z kamery. Na obraz se vykresluje kružnice, díky které se lépe zaměřují jamky fantomu. Vedle a pod obrazem z kamery se nachází komponenta pro ovládání pojezdu vozíku se vzorkem. Pojezdem vzorků je možné pohybovat ve všech směrech horizontální roviny. Pod obrazem z kamery lze nastavit, zda se má zobrazovat výstup horní nebo spodní kamery (pokud je přístroj spodní kamerou vybaven). Zaškrtnutím políčka *Pseudo-color* zapíná pseudobarvení obrazu, jež někdy využívají pracovníci pro viditelnější rozlišení hran a posouzení homogenity osvětlení scény. Tlačítko *Snapshot* pořizuje snímek z kamery. Uložení současného nastavení a přechod na další obrazovku se provádí pomocí tlačítka *Proceed*.



Obr. 9.1: Návrh společné části všech obrazovek

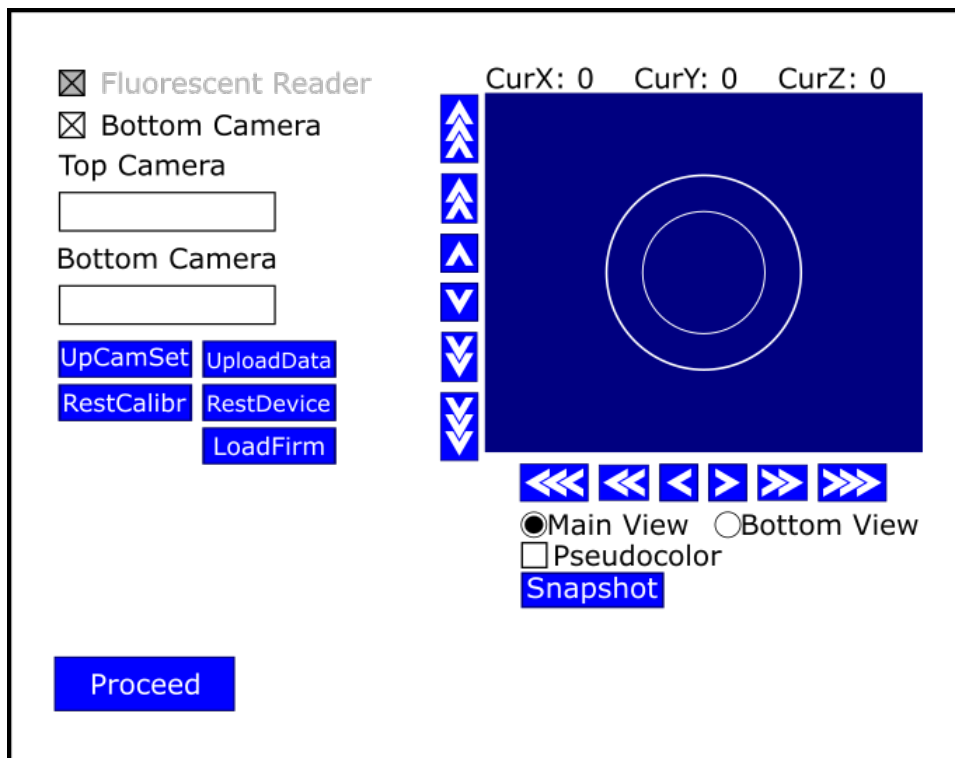
9.3.2 První obrazovka - obecné nastavení

Na první obrazovce (obr. 9.2) pracovník nahraje konfigurační XML soubory do kamery. Konfigurační soubor obsahuje nastavení digitálního zesílení kamery, expozice nebo např. vyvážení bílé. Kamera nemá k dispozici automatické ostření a software si jej řídí sám. Dále si na obrazovce může pracovník ověřit, zda se jedná o fluorescenční či kolorimetrický analyzátor nebo aktualizovat firmware stroje.

Nastavení analyzátoru se ukládá taktéž do souboru XML. Každý analyzátor obsahuje vždy dva takové soubory - v jednom je uloženo aktuální nastavení, ve druhém je nadefinované tovární nastavení. Na obrazovce lze pomocí tlačítek nahrát jiný kalibrační soubor (např. z dřívějšího seřízení), nebo v přístroji obnovit, již zmíněné, tovární nastavení.

9.3.3 Druhá obrazovka - nastavení krajních poloh

Na druhé obrazovce (obr. 9.3) se nastavují krajní polohy pojezdu vozíku se vzorkem. Nastavení probíhá v osách x a y (obr. 8.6). Základní bezpečné oblasti vytyčují koncové spínače. Maximální a minimální pozice vozíku se nastavují tak, aby vozík se vzorkem dojel maximálně 2 cm před koncový spínač. K nastavení těchto pozic slouží tlačítka *Max X*, *Max Y*, *Min X* a *Min Y*. Tlačítko *Eject* určuje pozici v ose y

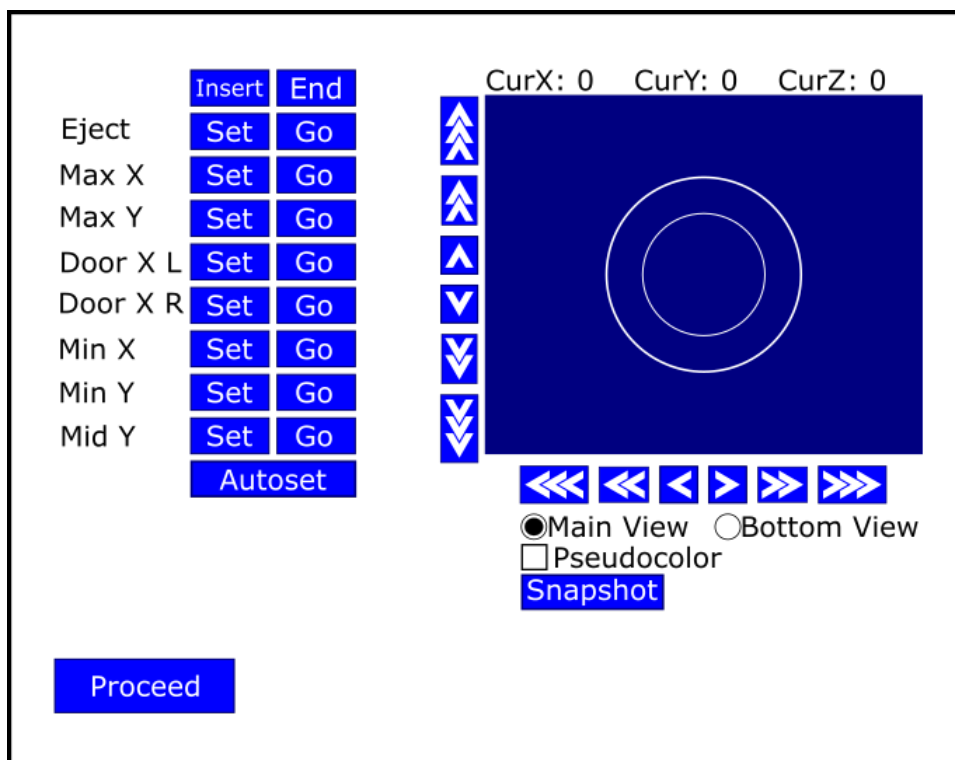


Obr. 9.2: Návrh první obrazovky - obecné nastavení

při vyjetí vozíku ven z analyzátoru pro vložení vzorku. Když je vozík vyjetý ven, je nutné definovat vzdálenost vozíku od krajů dvířek k čemuž slouží tlačítka *Door X L* a *Door X R*. *Mid Y* určuje pozici, kdy se vozík nalézá ve středu analyzátoru v ose *y*. Aplikace by měla obsahovat i automatické nastavení oblastí, jež seřízení stroje značně ulehčí.

Automatické nastavení krajních poloh motoru

Koncové spínače vymezují maximální oblast, kam vozík se vzorkem může dojet. Pokud by vozík měl vyjet za danou oblast, sepnutím jednoho z koncových spínačů se pohyb vozíku přeruší a přístroj upozorní na chybu, kterou dále zpracuje software. K sepnutí koncových spínačů by však vůbec nemělo dojít, jelikož software by neměl dovolit pohyb vozíku za krajní polohy a ty jsou nastaveny s dostatečnou rezervou před koncové spínače. Koncové spínače tedy představují záložní ochranu.



Obr. 9.3: Návrh druhé obrazovky - bezpečné oblasti

Při seřízení krajních poloh se postupně změří vzdálenosti mezi všemi koncovými spínači. Vzdálenosti jsou určeny počtem kroků motoru, které jsou vykonány od sepnutí jednoho koncového spínače do doby, než je sepnut druhý. Z naměřených hodnot se odpočítá konstantní vzdálenost od spínače (např. 5 % z celkové dráhy). Výsledek definuje maximální koncovou polohu před daným koncovým spínačem. Hodnota vzdálenostní konstanty byla stanovena empiricky na 5 % celkové dráhy a odpovídá zhruba 2 cm od koncového spínače.

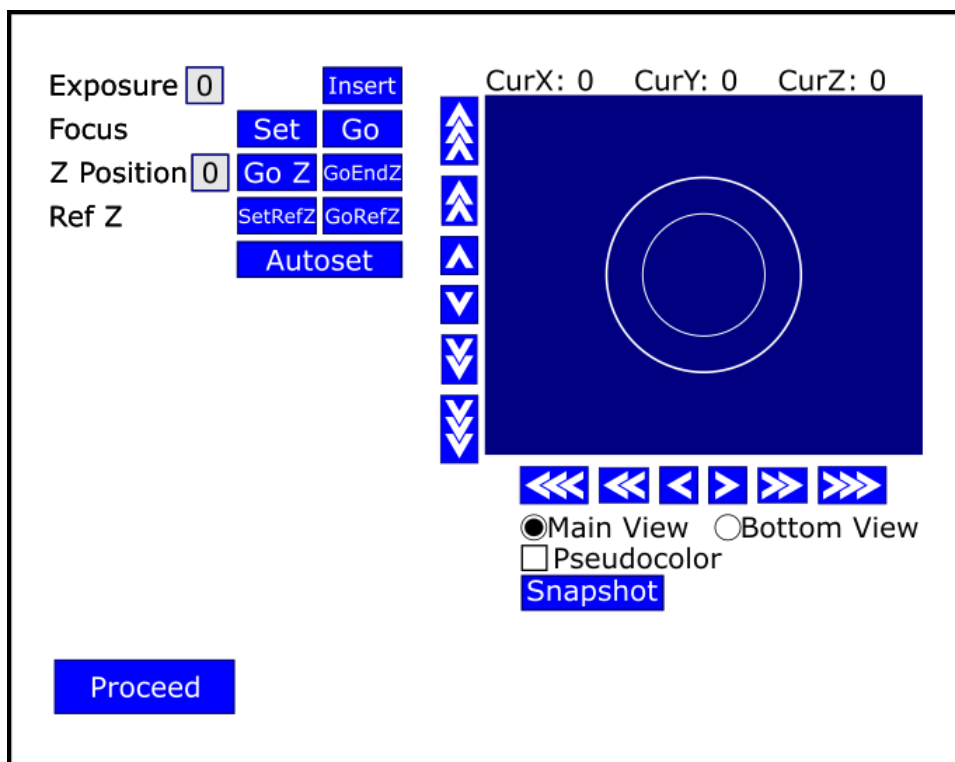
9.3.4 Třetí obrazovka - zaostření horní kamery

Obrazy pořízené kamerou musí být použitelné pro analýzu microarray, proto je nezbytné, aby velikost vzorku zabírala jasně danou plochu v obraze a obraz byl dostatečně ostrý. V praxi to znamená, že kamera musí být správně zaostřena a umístěna v ideální výšce nad vzorkem.

Obrazovka (obr. 9.4) obsahuje všechny komponenty pro zaostření kamery. Kamera se pohybuje pomocí motoru v ose z . Zaostření je částečně zautomatizováno pomocí funkce *Focus*.

V původní aplikaci se výška kamery nastavuje ručně podle kružnice zobrazené na výstupu kamery. Pracovník sjíždí s kamerou dolů, dokud není vykreslená kružnice na obraze shodná s velikostí otvoru ve fantomu pro seřízení. Tento postup je

pro pracovníka mnohdy zdlouhavý a nabízí se možnost uplatnění automatizačního procesu.

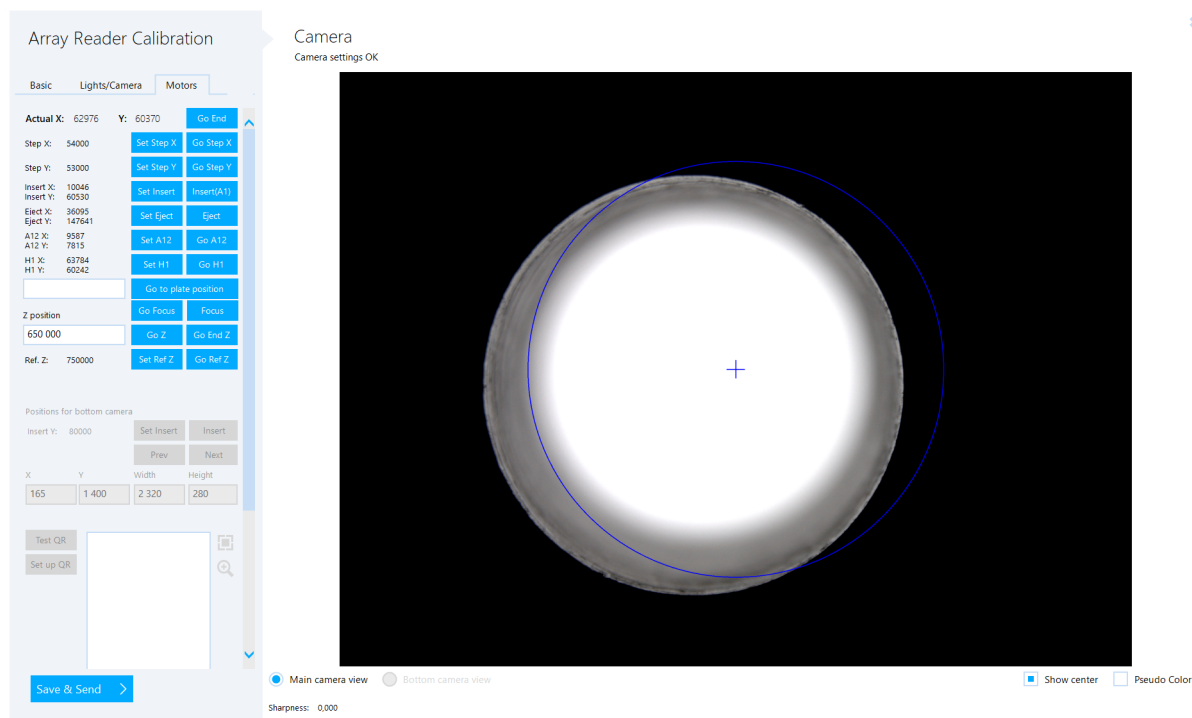


Obr. 9.4: Návrh třetí obrazovky - nastavení zaostření horní kamery

Nastavení pozice osy Z

Princip automatizace pro tento krok spočívá v iterativním oddalování kamery a analýzy aktuálního obrazu. Seřízení se provádí pomocí fantomu. Postup automatizačního procesu je následující:

- Jeden z otvorů fantomu (bez optického filtru) se pozicuje pod kameru. Na obraze z kamery se nyní objeví bílý kruh na černém pozadí, což je způsobeno prosvětlením otvoru jinak neprůhledného fantomu vnitřním osvětlením analyzátoru (obr. 9.5).
- Je pořízen snímek z kamery.
- Ve snímku se detekují kružnice a průměrují se do jedné výsledné kružnice.
- Pokud je průměr výsledné kružnice v mezích hledané velikosti, proces se ukončí. V opačném případě kamera sjede níže ke vzorku, čímž dojde ke zvětšení plochy, jež zabírá kruh v obraze a proces detekce a analýzy kružnic se opakuje.



Obr. 9.5: Bílý kruh způsobený prosvětlením otvoru fantomu

9.3.5 Čtvrtá obrazovka - nastavení referenčních pozic

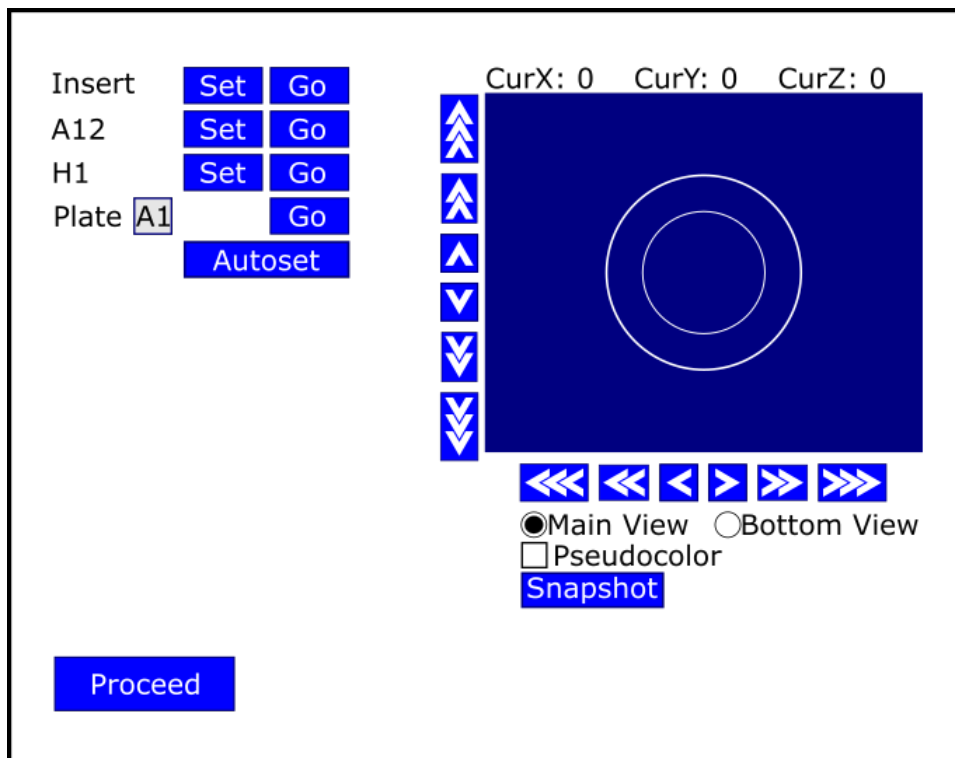
Při analýze vzorků v mikrotitrační destičce pod kameru postupně najíždí vybrané jamky, pořídí se jejich snímek a ten se následně zpracuje a vyhodnotí. Aby jamky mikrotitrační destičky správně najížděly pod kameru, musí být známy pozice tří referenčních bodů (jamky A1, A12 a H1). Tyto tři jamky jsou na krajích mikrotitrační destičky (obr. 8.4) a díky tomu z jejich pozic lze určit polohu kterékoliv jiné jamky.

Obrazovka (obr. 9.6) obsahuje komponenty pro nastavení těchto referenčních pozic a komponentu pomocí níž lze pozicovat jakoukoli jamku mikrotitrační destičky pod kameru pro ověření správného seřízení.

Automatické nalezení referenčních bodů mikrotitrační destičky

V původní aplikaci se pozice nastavovaly ručně, nový návrh však počítá s kompletní automatizací tohoto kroku. Využívá se opět fantomu pro seřizování a proces je následující:

- V zařízení jsou uloženy přibližné polohy referenčních pozic. Postupně je každá z referenčních jamek pozicována pod kameru. Kamera snímá otvor ve fantomu, jenž se v obraze projeví jako bílý kruh na černém pozadí.
- Je pořízen snímek z kamery.
- Ve snímku se detekují a průměrují kružnice; výsledkem je jedna kružnice.



Obr. 9.6: Návrh čtvrté obrazovky - nastavení referenčních pozic

- Je určen střed výsledné kružnice. Pokud je střed kružnice shodný se středem snímku, je proces ukončen. Pokud není dosaženo shody středů, vypočítají se z jejich vzdálenosti počty kroků motoru, o který se posune vozík se vzorkem v osách x a y .

9.3.6 Pátá obrazovka - osvit

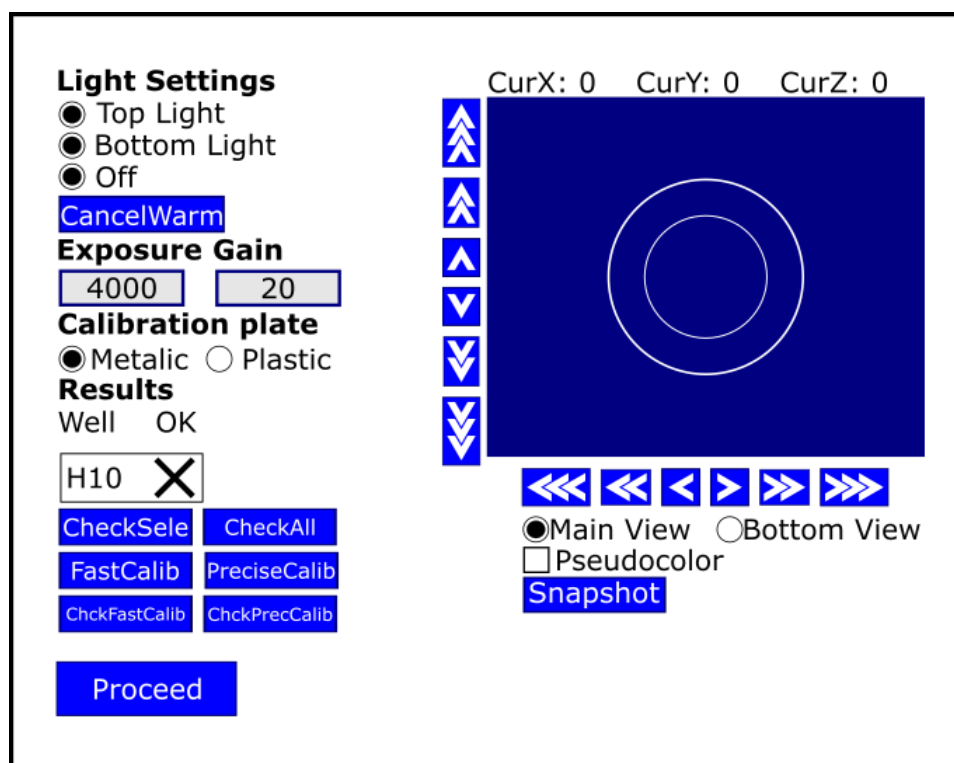
Obrazovka (obr. 9.7) slouží k seřízení spodního a horního osvitu přístroje. Komponenta pro toto seřízení je již navržená a automatizována. Proto byla využita i v novém návrhu aplikace a příliš se neliší od původní verze. K seřízení osvitu se využívají optické filtry zmíněného fantomu.

Intenzita osvitu se mění s jeho teplotou. Proto bude obrazovka přístupná až po uplynutí doby nutné k zahřátí a ustálení teploty osvitu (10 min) na stabilní hodnotě.

Seřízení osvitu probíhá ve dvou krocích. V prvním kroku se změří intenzity osvitu při průchodu různými optickými filtry. V kroku druhém se hledá nejlepší hodnota intenzity pro osvit. Naměřené hodnoty slouží ke srovnání předešlého a aktuálního nastavení. Aby bylo seřízení validní, nesmí se seřízená hodnota osvitu lišit od změřené hodnoty o více než 3 %.

Změny komponent se budou týkat zejména generování záznamu ze seřízení a vyvážení bílé barvy pro kameru. Vyvážení bílé barvy se v původní aplikaci provádělo

tak dlouho, dokud měl pracovník zaškrtnuté příslušné tlačítko. V nové aplikaci se bude provádět po dobu 15 sekund po stisku tlačítka. Tím se pro pracovníka krok ulehčí a nemůže dojít k situaci, kdy pracovník tlačítko zapomene zaškrtnuté a znehodnotí nastavení přístroje. Oproti původní verzi by také mělo být možné vygenerovat záznam ze seřízení i bez změřených srovnávacích hodnot.

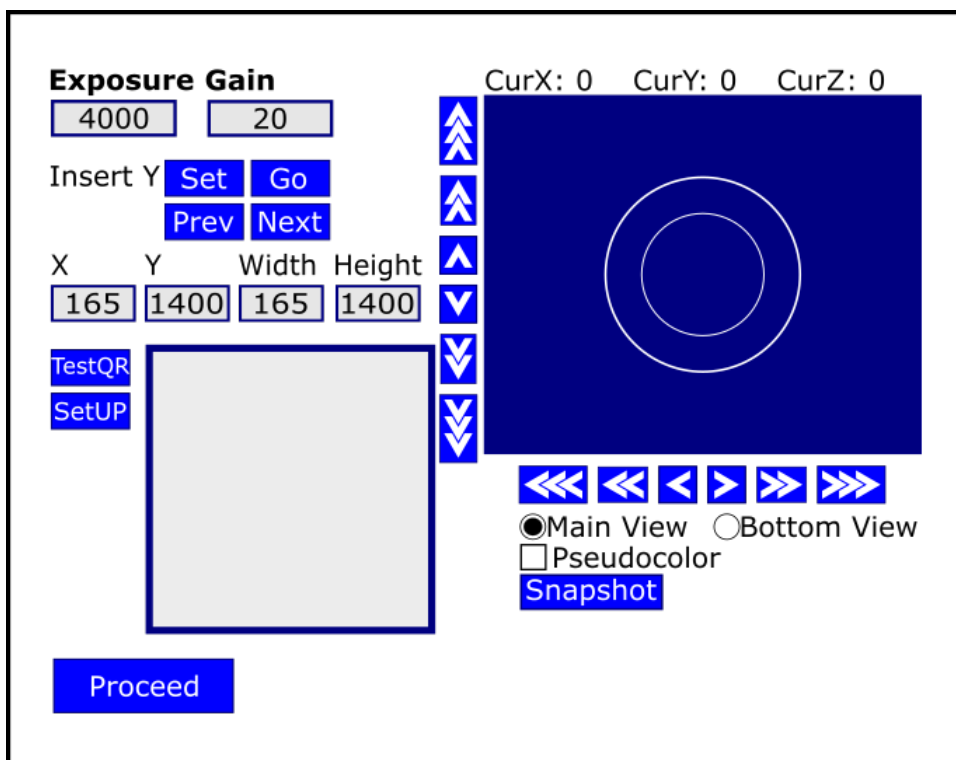


Obr. 9.7: Návrh páté obrazovky - nastavení osvitů

9.3.7 Šestá obrazovka - nastavení spodní kamery

Obrazovka (obr. 9.8) bude přístupná pouze v případě, že zařízení bude vybaveno spodní kamerou. Spodní kamera slouží ke čtení čárových kódů a QR kódů, jimiž se někdy ze spodní strany označují vzorky. Označovat se mohou buď celé mikrotitrační destičky, jednotlivé stripy a nebo každá jamka. V QR kódech nebo čárových kódech jsou uloženy informace o testu na vzorku (např. typ testu, šarže testu, datum expirace apod.).

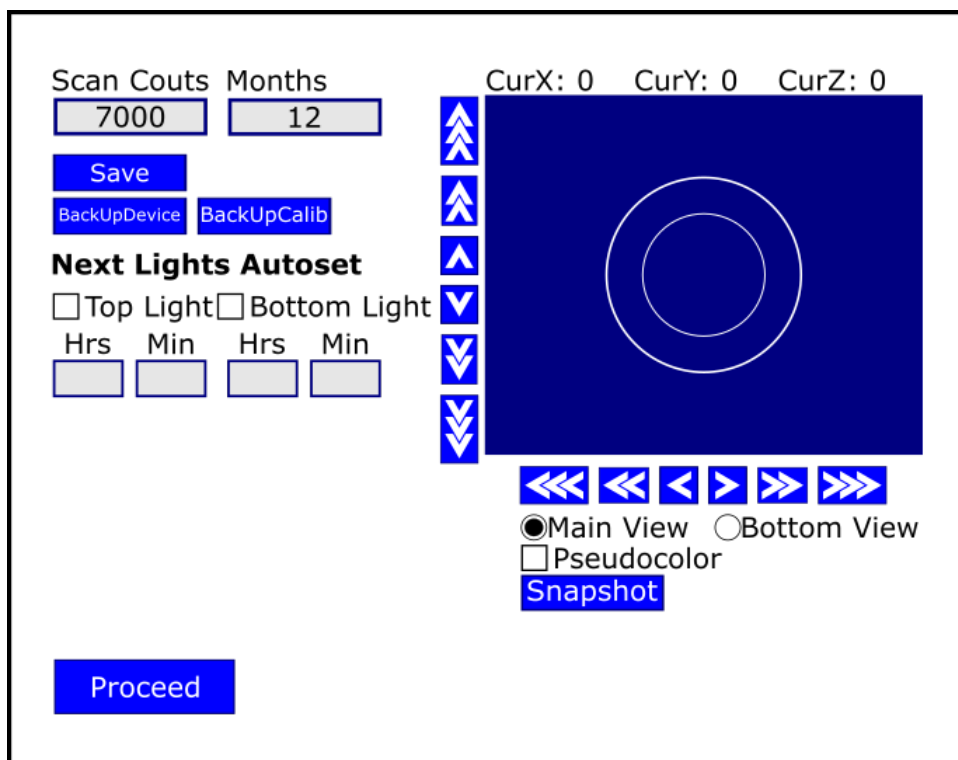
Na obrazovce je nutné nastavit především správnou vzdálenost mezi dvěma stripy, což umožní číst označení ve stripech či jednotlivých jamkách. Označení pro celou mikrotitrační destičku se lepí na pozici prvního stripu.



Obr. 9.8: Návrh šesté obrazovky - nastavení spodní kamery

9.3.8 Sedmá obrazovka - záloha a archivace seřízení

Po seřízení je nutné do přístroje nastavení nahrát a provést zálohu nových nastavení. K tomu slouží poslední obrazovka (obr. 9.9). Nastavuje se na ní rovněž čas nebo počet možných použití před nutností dalšího seřízení.



Obr. 9.9: Návrh sedmé obrazovky - záloha a archivace seřizení

10 Implementace návrhu aplikace

Po vytvoření návrhu nové aplikace proběhla jeho implementace s využitím vývojářského prostředí Visual Studio společnosti Microsoft. Jako první byla vytvořena struktura aplikace. Aplikace byla rozdělena na jednotlivé obrazovky a těm byly přiřazeny určité komponenty a funkce. Po zprovoznění všech komponent proběhla implementace automatizačních funkcí.

10.1 Struktura a základní funkce

Struktura aplikace byla vytvořena pomocí komponent společnosti BioVendor, aby aplikace korespondovala s vizuálním stylem dalších částí softwaru. Stejně tak funkce jednotlivých komponent byly implementované za použití již vytvořených tříd a metod, určených např. pro komunikaci se strojem, ovládání motorů a osvětlení apod.

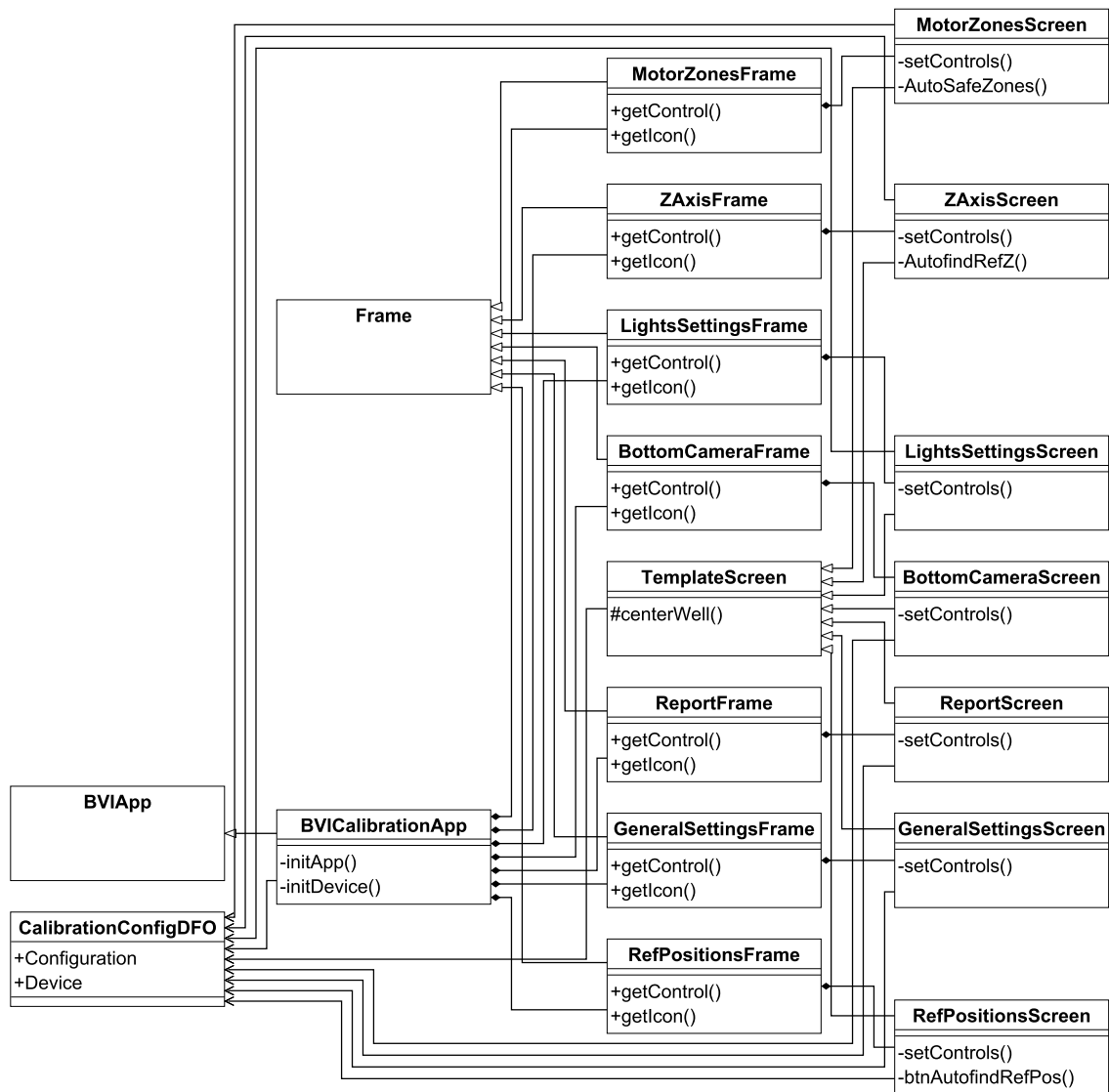
Funkce komponent mohly být často převzaty z původní aplikace a případně upraveny, aby se zachovala jejich funkčnost i v novém prostředí. Zásadním rozdílem mezi původní a novou aplikací je rozdělení do obrazovek. Původní aplikace byla v softwaru definována jako jeden objekt a všechny důležité proměnné byly sdílené mezi všemi komponentami. V nové aplikaci je však každá obrazovka reprezentována objektem samostatně (byť mají společného předka, což alespoň částečně usnadňuje práci s obrazovkami jako celkem); proměnné mezi nimi nejsou sdílené. Tento problém byl vyřešen implementací třídy `MICalibrationConfigDF0`, v jejíž instanci jsou uloženy společné proměnné a instance je přenášena mezi obrazovkami.

10.1.1 Implementace jádra aplikace

Aplikace je vytvořena jako instance třídy `BVICalibrationApp`, která dědí své vlastnosti ze třídy `BVIApp`. V tomto formátu jsou vytvářeny všechny podaplikace softwaru BioVendor Analytics.

Hlavní úlohou objektu `BVIApp` je vytvoření instancí tříd `Frame`, které představují funkční bloky aplikace. Obsahují různé metody a proměnné např. pro přechody mezi jednotlivými bloky, indexy pořadí bloků, ikony bloku apod. Především však při jejich inicializaci vytváří instance třídy `Screen`, které vystupují jako hlavní vizuální kontrolery aplikace. Uživatel pomocí kontroleru zadává pokyny přístroji. Kontrolery jsou odpovědné za funkce a procesy celé aplikace.

V případě této práce bylo vytvořeno sedm tříd, které dědí ze třídy `Frame`. Každé zvláště je poté přidělen individuální kontroler dále nazývaný jako obrazovka. Jedná se o potomka třídy `Screen`. UML třídový diagram jádra aplikace je na obr. 10.1.



Obr. 10.1: UML třídivý diagram jádra aplikace

10.1.2 Implementace jednotlivých obrazovek

Stěžejním úkolem praktické části bylo vytvoření jednotlivých obrazovek (kontrolerů). V následujících podkapitolách bude nejprve představena třída `TemplateScreen`, která představuje rodičovskou třídu pro všechny obrazovky. Poté bude rozebrána každá obrazovka zvlášť.

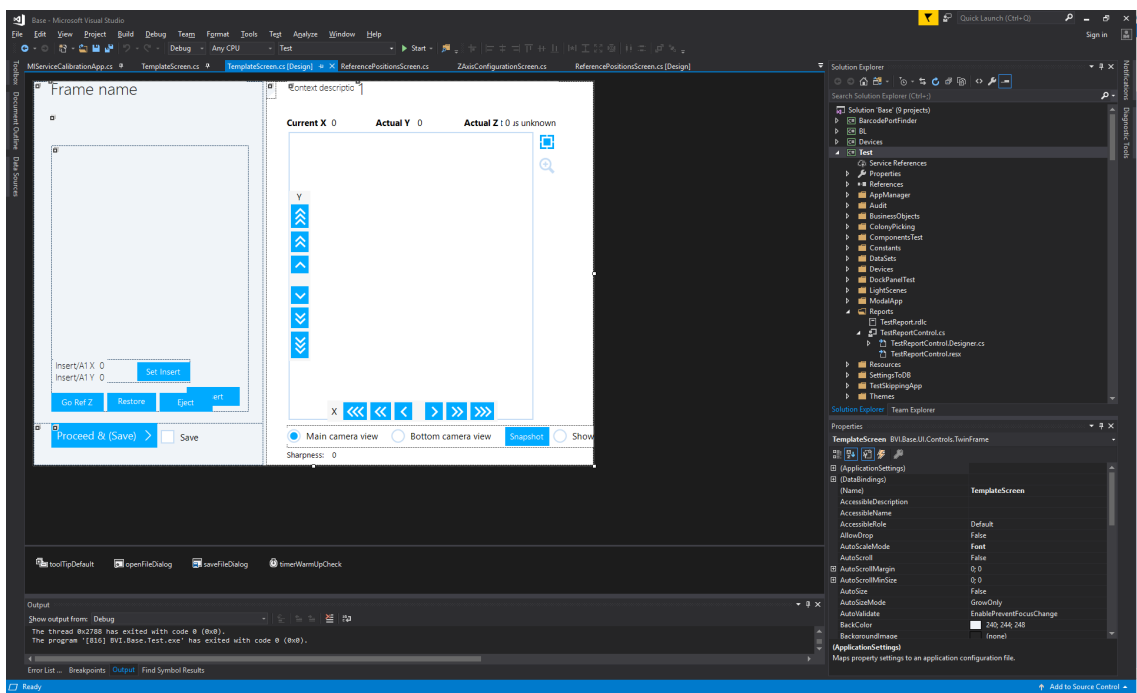
Společná část obrazovek

Společná část všech obrazovek je implementována pomocí třídy `TemplateScreen`; třída je předkem pro všechny odvozené obrazovky. Komponenty třídy `TemplateScreen` jsou dostupné na všech obrazovkách. Dědí se i metody, které pou-

žívá více než jedna obrazovka (např. zobrazení výstupu kamery, překreslování společných indikátorů apod.).

Na levé straně kontroleru se nachází panel, na kterém jsou umístěny komponenty potomků, pravá strana je pro všechny obrazovky shodná. Obsahuje především výstup z kamery, sérii ovládacích tlačítek a textových oken. Pod panelem na levé straně se nachází tlačítko **Proceed & Save**, které vyvolává přechod na další obrazovku. Pokud je zaškrtnuté tlačítko **Save** napravo od něj, rovněž také uloží současné nastavení. V pravé horní části je zobrazena nápověda, ve které jsou popsány kroky prováděné při seřizovacím procesu. V dolní části jsou umístěny přepínací tlačítka, které mění výstup obrazu. Lze zobrazit např. mřížku, oproti které je možné snadno porovnat rovnost hran v obraze, pseudozbarvení obrazu, které zviditelňuje homogenitu osvětlení (uplatňuje se spíše ve fluorimetrickém analyzátoru), záměrný kříž tak, jak je viditelný na obrázku (10.3), nebo pouze nezměněný výstup kamery.

Na obrázku 10.2 je vizuální návrh třídy `TemplateScreen` v prostředí Microsoft Visual Studio.

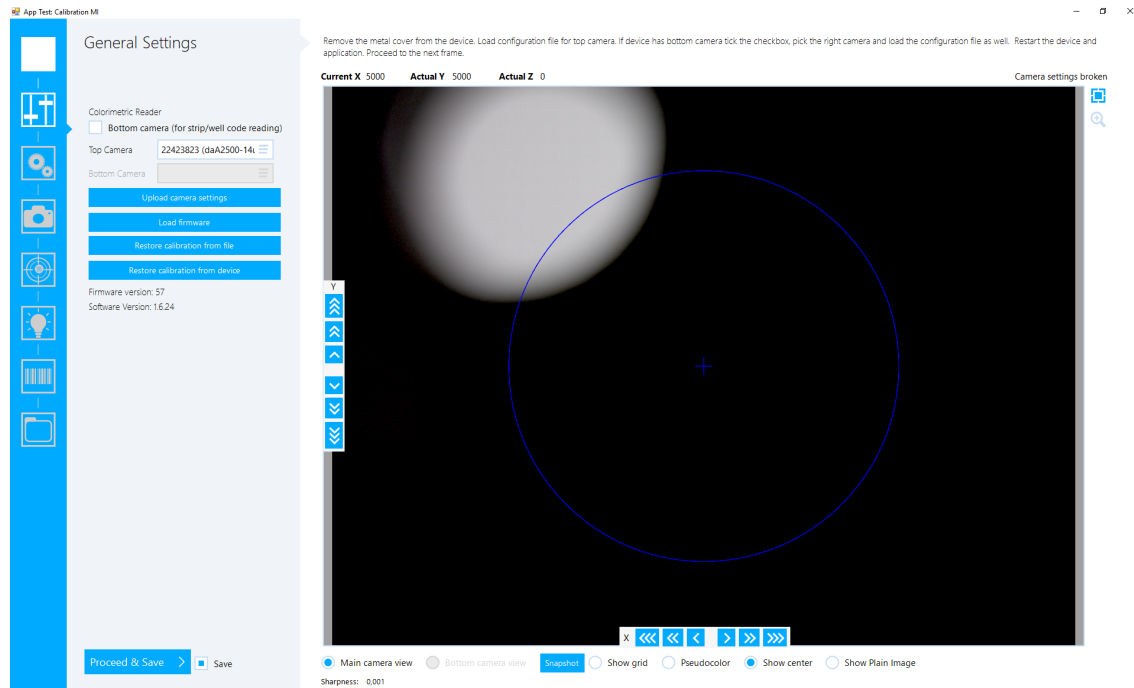


Obr. 10.2: Společná část obrazovek v prostředí Microsoft Visual Studio

První obrazovka - obecná nastavení

První obrazovka je reprezentována třídou `GeneralSettingsScreen`. Uživatel si může ověřit, zda se jedná o kolorimetrický analyzátor, nahrát konfigurační soubory, aktualizovat firmware stroje nebo obnovit nastavení z přístroje popř. externího souboru.

Oproti návrhu přibyly textové komponenty, které informují uživatele o verzích firmwaru a softwaru připojeného stroje. V původní aplikaci se zobrazovala pouze verze firmwaru stroje a navíc se zastaralým verzováním, takže tato informace nebyla pro uživatele užitečná. Oproti návrhu byla také odstraněna komponenta pro nahrání libovolného souboru do stroje, jelikož se již nepoužívá.



Obr. 10.3: První obrazovka aktualizované aplikace - obecná nastavení

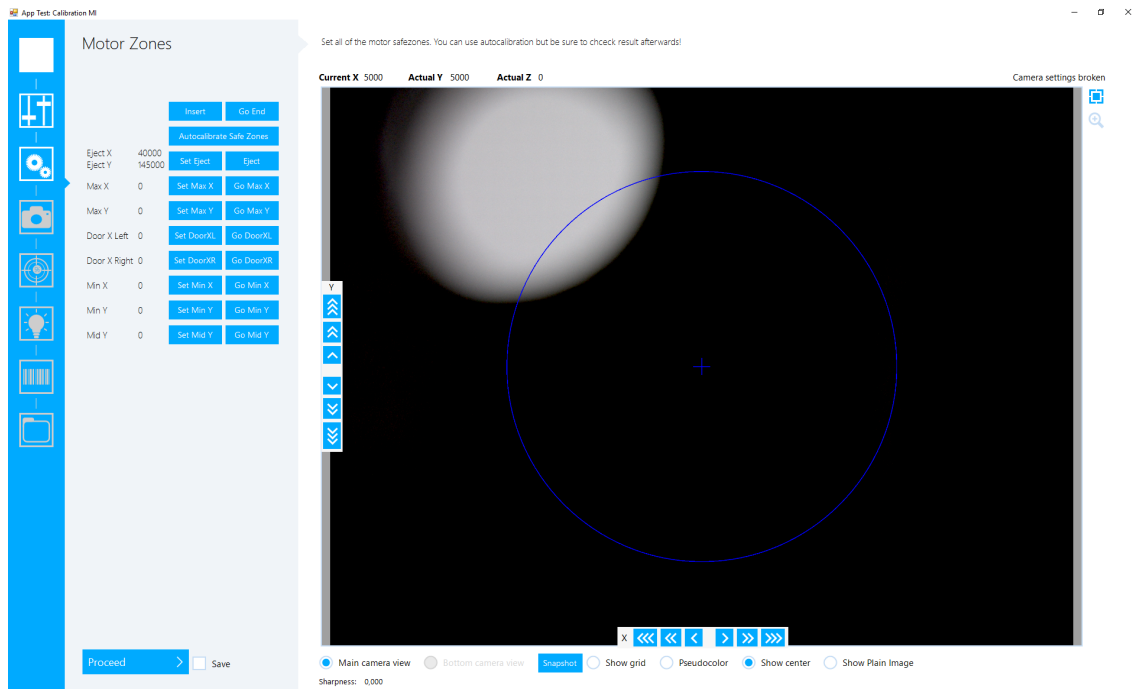
Druhá obrazovka - bezpečné oblasti vozíku

Na obrazovce (obr. 10.4) jsou umístěny komponenty pro nastavení bezpečných oblastí přístroje. Obrazovka je instancí třídy `MotorZonesScreen`. Od návrhu se neliší. V jednotlivých řadách se nastavují určité oblasti; tlačítko `Set` slouží k nastavení aktuální polohy motorů, jako poloha dané oblasti, tlačítko `Go` slouží k posunu vozíku na souřadnice dané oblasti (využívá se např. pro kontrolu). V přílohách lze nalézt reprezentativní zdrojový kód, sloužící pro nastavení jedné oblasti, konkrétně jednu z bezpečných oblastí v ose x (výpis B.1).

Tlačítko `Autocalibrate Safe Zones` slouží ke spuštění automatizačního procesu, o kterém bude řeč dále v kapitole.

Třetí obrazovka - zaostření kamery

Na třetí obrazovce (obr. 10.5) se nastavuje zaostření kamery (resp. její výška nad vzorkem). Tlačítko `Go Edge` pozicuje vozík pod kameru tak, aby byla vidět hrana



Obr. 10.4: Druhá obrazovka aktualizované aplikace - nastavení bezpečných oblastí

vozíku. Díky tomu lze vidět, zda je kamera usazena rovně (pro lepší viditelnost byla implementována již zmíněná mřížka). Tlačítko nebylo původně zahrnuto v návrhu.

Ostatní tlačítka slouží k ovládání motoru, jenž posouvá kamerou ve vertikální ose.

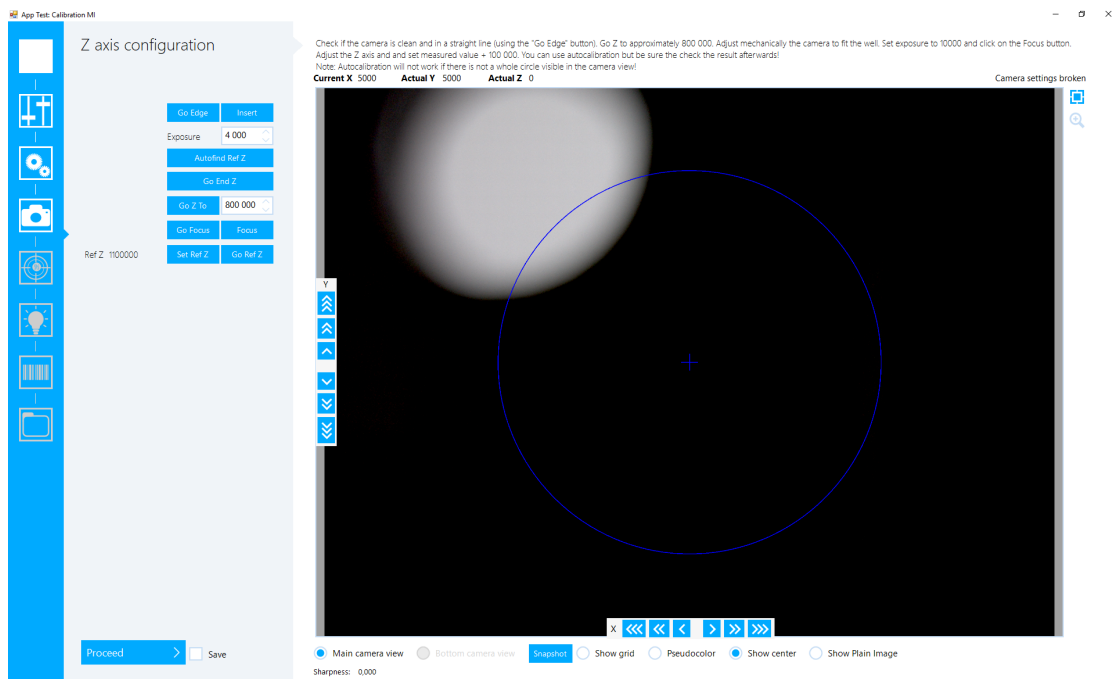
Obrazovka obsahuje tlačítko **Autofind Ref Z** pro spuštění automatizačního procesu, o jehož funkci bude řeč dále v kapitole.

Čtvrtá obrazovka - referenční pozice

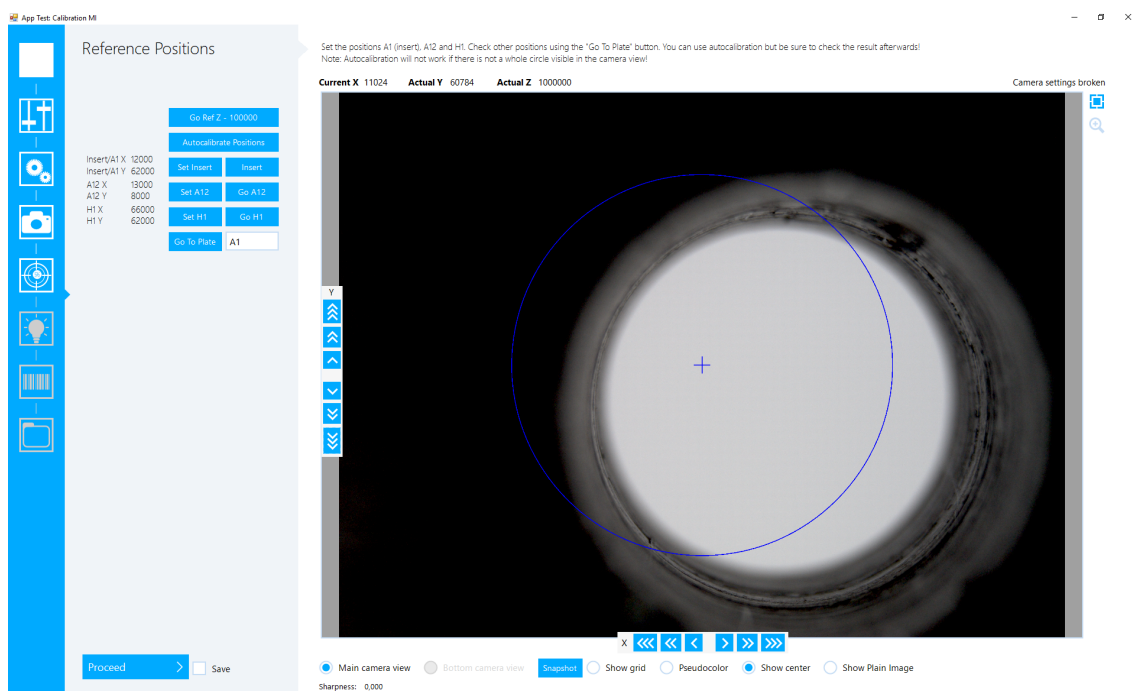
Při spuštění obrazovky (obr. 10.6) se kamera automaticky posune na úroveň o 100 000 kroků níže pod referenční pozici. V této výšce je ostře viditelná metalická destička, se kterou by se měly provádět úkony spojené s touto obrazovkou. Tlačítko **Go Ref Z** slouží k najetí kamery do referenční pozice, ve které je zaostřena na plastovou destičku. Destička se někdy do přístroje vkládá pro kontrolu nastavených pozic.

Tlačítko **Autocalibrate Positions** spouští automatizační proces. Při selhání automatizačního procesu nastaví uživatel pozice manuálně.

Po správném nastavení by měly být překontrolovány i ostatní pozice (dopočítávají se právě z pozic referenčních). K tomu se využívá komponenta umístěna v dolní části panelu (**Go To Plate**).



Obr. 10.5: Třetí obrazovka aktualizované aplikace - nastavení výšky kamery



Obr. 10.6: Čtvrtá obrazovka aktualizované aplikace - nalezení referenčních pozic

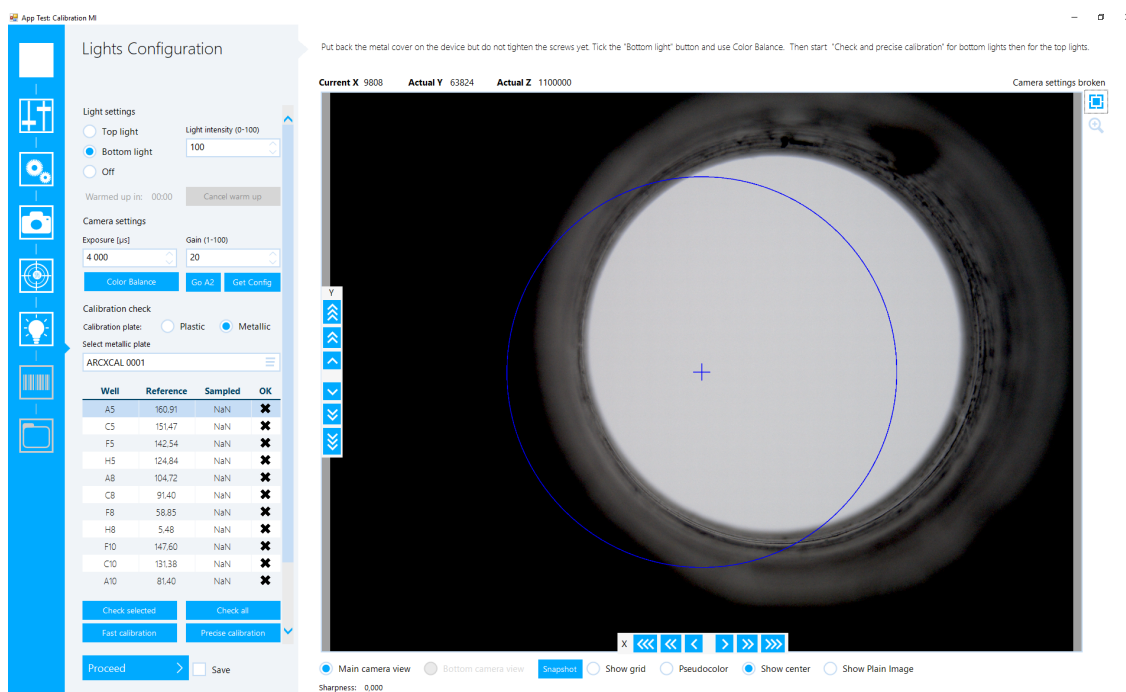
Pátá obrazovka - nastavení osvitů

Veškeré komponenty použité na páté obrazovce (obr. 10.7) jsou součástí jednoho komplexního celku, který je převzatý z původní aplikace. Z důvodu jiné struktury

nové verze aplikace bylo nutné komponentu pouze zprovoznit v novém prostředí.

Ovládat komponentu a tím i osvit přístroje je možné až po deseti minutách od zapnutí aplikace. Tato doba je nutná k zahřátí osvitu a ustálení intenzity světla. Tím se vyloučí špatné nastavení osvitu. Časovač, který zajišťuje odpočet deseti minut byl původně globální pro celou aplikaci. Při implementaci musel být časovač uložen do instance třídy, která je společná pro všechny snímky a funguje tedy nezávisle na tom, na které obrazovce uživatel právě pracuje.

Další změnou bylo zpřístupnění generování seřizovacího protokolu i z nekompletních dat. Tato změna byla učiněna proto, aby při první kalibraci nebylo nutné kontrolovat předchozí nastavení osvitu a měřit zbytečně jeho intenzitu. Pracovník tedy může zapnout automatické seřízení osvitu bez kontroly předchozích hodnot a proces je méně časově náročný (zkrátí se asi o 10 minut).



Obr. 10.7: Pátá obrazovka aktualizované aplikace - nastavení osvitu

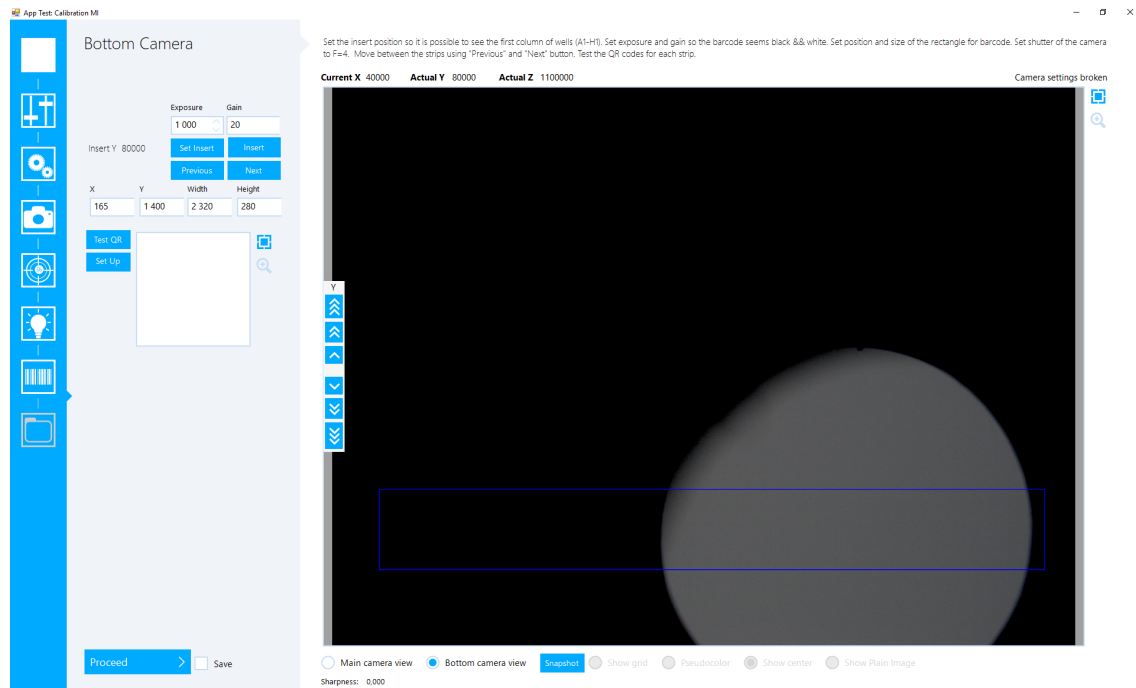
Šestá obrazovka - spodní kamera

Šestá obrazovka (obr. 10.8) slouží k seřízení spodní kamery v případě, že ji seřizovaný analyzátor obsahuje. Obrazovka je tedy přístupná pouze, pokud přístroj obsahuje správně nakonfigurovanou spodní kameru a zároveň uběhla doba nutná k zahřátí osvitu.

Při seřízení spodní kamery se nejprve nastaví výchozí pozice pro čtení QR nebo čárových kódů. Je viditelný první kód; poté se pomocí tlačítek Previous a Next

pozicují pod spodní kameru ostatní čárové kódy, kterými jsou označovány řádky mikrotitračních destiček. Pomocí komponenty ve spodní části panelu (konkrétně tlačítka **Set Up**) se testuje načítání kódů. Nastavení se potvrdí tlačítkem **Set Up**.

Při implementaci se využilo komponent z původní aplikace. Automatizační principy by při těchto úkonech pravděpodobně nepřinesly výrazné zjednodušení nebo zrychlení seřizovacího procesu.

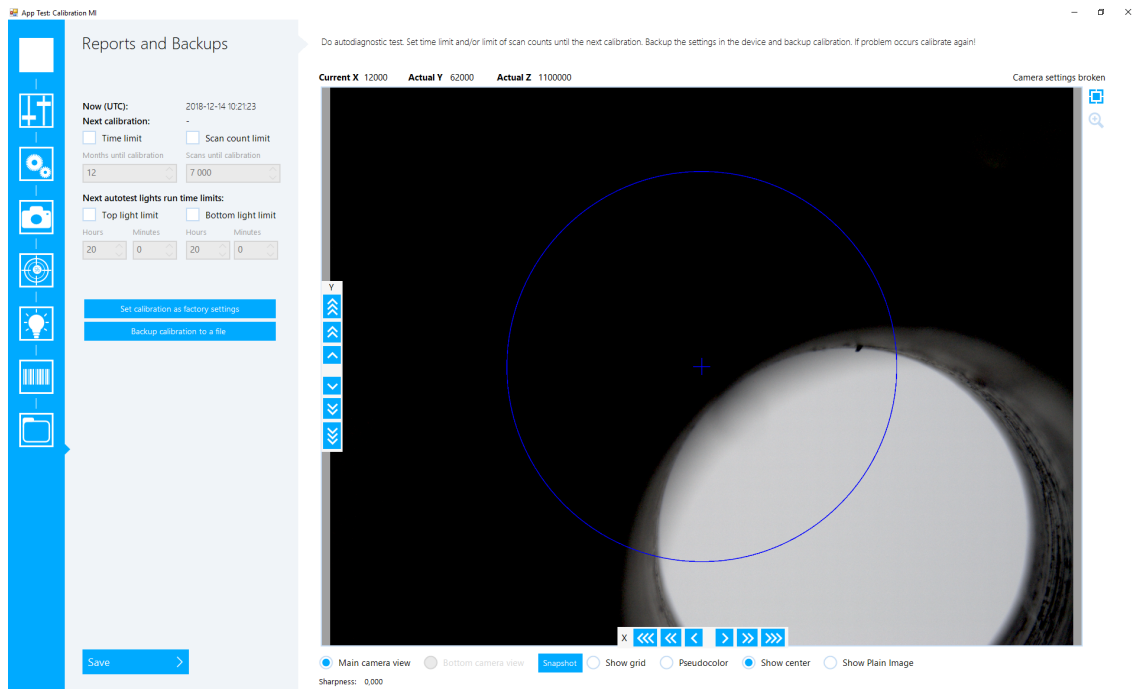


Obr. 10.8: Šestá obrazovka aktualizované aplikace - nastavení spodní kamery

Sedmá obrazovka - kalibrační protokoly a limity přístroje

Poslední obrazovka (obr. 10.9) slouží k uložení údajů o seřízení do přístroje. Oproti původní aplikaci (a návrhu) se hodnoty ukládají současně s hlavním konfiguračním souborem a tím pádem aplikace nemusí obsahovat zvláštní tlačítko, které může být pro pracovníky matoucí.

Jedním z bodů zadání nové aplikace bylo, aby přístroj obsahoval historii všech svých seřízení. Původně bylo v plánu i sloučit dva různé konfigurační soubory ukládané do přístroje, to se však nakonec ukázalo nevhodné z důvodu zpětné kompatibility přístrojů. Historie konfiguračních souborů se však do přístroje ukládá; původní aplikace aktuální konfigurační soubor přepisovala novými. V nové aplikaci se kopie starého konfiguračního souboru uloží do historie a starý konfigurační soubor je dále upravován jako nový.



Obr. 10.9: Sedmá obrazovka aktualizované aplikace - záloha a archivace seřízení

10.2 Automatizační algoritmy

Důležitým bodem zadání aplikace byla automatizace některých kroků seřizovacího procesu. Dle návrhu bylo pro automatizaci využito koncových spínačů a zpracování obrazu. Pro zpracování obrazu byl použitý C++ balíček OpenCV, jelikož pro programovací jazyk C# není obdobný balíček k dispozici. Kód byl kompilován do dynamické knihovny.

10.2.1 Implementace algoritmu pro nalezení bezpečných oblastí

Při automatickém nalezení krajních oblastí vozík postupně najede na všechny koncové spínače (představuje maximální možnou polohu vozíku v daném směru). Při pohybu se počítá počet kroků, které vykonal motor při přemístění vozíku z jedné polohy do druhé. Z těchto údajů se poté počítají ostatní polohy, podle empiricky zjištěných vzorců.

Výpis algoritmu pro nalezení bezpečných oblastí analyzátoru je obsažen v přílohách práce (B.2).

Výpočet pozic v ose x

Dle algoritmu vozík nejprve najede na jeden ze dvou koncových spínačů v ose x . Souřadnice v tomto bodě je rovna nule. Dále vozík najede na druhý z koncových

spínačů. Souřadnice x vozíku, odpovídá počtu kroků motoru, které byly vykonány při přesunu vozíku. Tento počet se uloží do proměnné `endPosition`. Středu osy x odpovídá proměnná `halfX`, která se vypočte podle vzorce $halfX = endPosition/2$. Pohyby vykonávají příkazy `GoToEndSwitchX_Insert` a `GoToEndSwitchX_Eject`.

Pro výpočty krajních poloh vozíku od dvířek je nutné zavést proměnnou `doorGap`, která odpovídá vychýlení (počtu kroků) vozíku od středové polohy `halfX`. Tato poloha byla empiricky určena jako 5 % z celkové dráhy vozíku, tedy $doorGap = endPosition * 0.05$. Poté souřadnice krajních poloh dvířek `SafeDoorX1` a `SafeDoorX2` lze vypočítat jako $SafeDoorX1 = halfX - doorGap$ a $SafeDoorX2 = halfX + doorGap$.

Minimální a maximální poloha vozíku (proměnné `SafeMinX` a `SafeMaxX`) v ose x je opět empiricky určena jako 5 % z celkové dráhy od krajních spínačů. Souřadnice krajních poloh lze tedy vypočítat dle vztahů $SafeMinX = endPosition * 0.05$ a $SafeMaxX = endPosition * 0.95$.

Výpočet pozic v ose y

Další pozice jsou vypočteny z pohybu v ose y , to znamená z dráhy od zadní stěny analyzátoru až do nejzazšího bodu, ve kterém je vysunut vozíček ven z analyzátoru.

Nejprve je vozíček pozicován na střed analyzátoru v ose x , tak aby při vyjetí ven z přístroje nedošlo k nárazu do dvířek a poškození přístroje. Poté vozíček najede na koncový spínač umístěný u zadní stěny analyzátoru; tato pozice odpovídá nulové pozici. Pohyb je vykonán pomocí příkazu `GoToEndSwitchY_Insert()`. Následně vozíček najíždí na druhý koncový spínač (příkaz `GoToEndSwitchY_Eject()`) a odečítá se počet kroků (`endPosition`). Vzdálenostem od zmíněných koncových spínačů odpovídají proměnné `SafeMinY` a `SafeMaxY`. Empiricky bylo zjištěno, že se jedná přibližně o 3 % z celkové dráhy, což odpovídá vztahům

$SafeMinY = endPosition * 0.03$ a $SafeMaxY = endPosition * 0.97$. Pozice `EjectY` by měla mít větší rezervu od koncového spínače a tato rezerva byla stanovena na 5 % celkové dráhy (tedy vztah $SafeEjectY = endPosition * 0.95$). Poslední z nastavovaných poloh je myšlená pozice ve středu analyzátoru, tak aby vozíček byl v dostatečné vzdálenosti od dvířek. Souřadnice byla určena jako 42 % celkové dráhy ($SafeMiddleY = endPosition * 0.42$).

10.2.2 Algoritmy využívající zpracování obrazu

Automatizační procesy implementované v praktické části využívají zpracování obrazu; konkrétně již zmíněnou knihovnu OpenCV. Jádro zpracování obrazu obstarává metoda `FindCircle()`, jež detekuje kružnice pomocí Houghovy transformace.

Metoda je definovaná ve třídě `CalibrationImage`, která slouží k přehlednému použití metod definovaných v dynamické knihovně. `FindCircle` dále volají funkce `centerWell` a `AutofindRefZ`.

Struktura implementace

V průběhu automatizačních procesů se prostupně využívají tři základní třídy. Samotné zpracování obrazu zajišťuje třída `Calibration`, jež je implementována v C++ a kompilována do dynamické knihovny s příponou `.dll`. Díky její implementaci v C++ je možné v jejích metodách využít knihovny OpenCV pro zpracování obrazu. Metody této třídy jsou volány metodami třídy `CalibrationImage`. Je implementována v jazyce C# a tvoří v jakýsi „obal“ pro metody definované v dynamické knihovně. Převádí metody definované v C++ do přehlednější a použitelnější formy v jazyce C#. Metody ze třídy `CalibrationImage` je poté možné volat ze kterýchkoli jiných tříd; v případě této práce jsou metody této třídy volány z obrazovek. Na obr. 10.10 je vyobrazen sekvenční UML diagram, který popisuje práci s metodami pro zpracování obrazu.

Houghova transformace

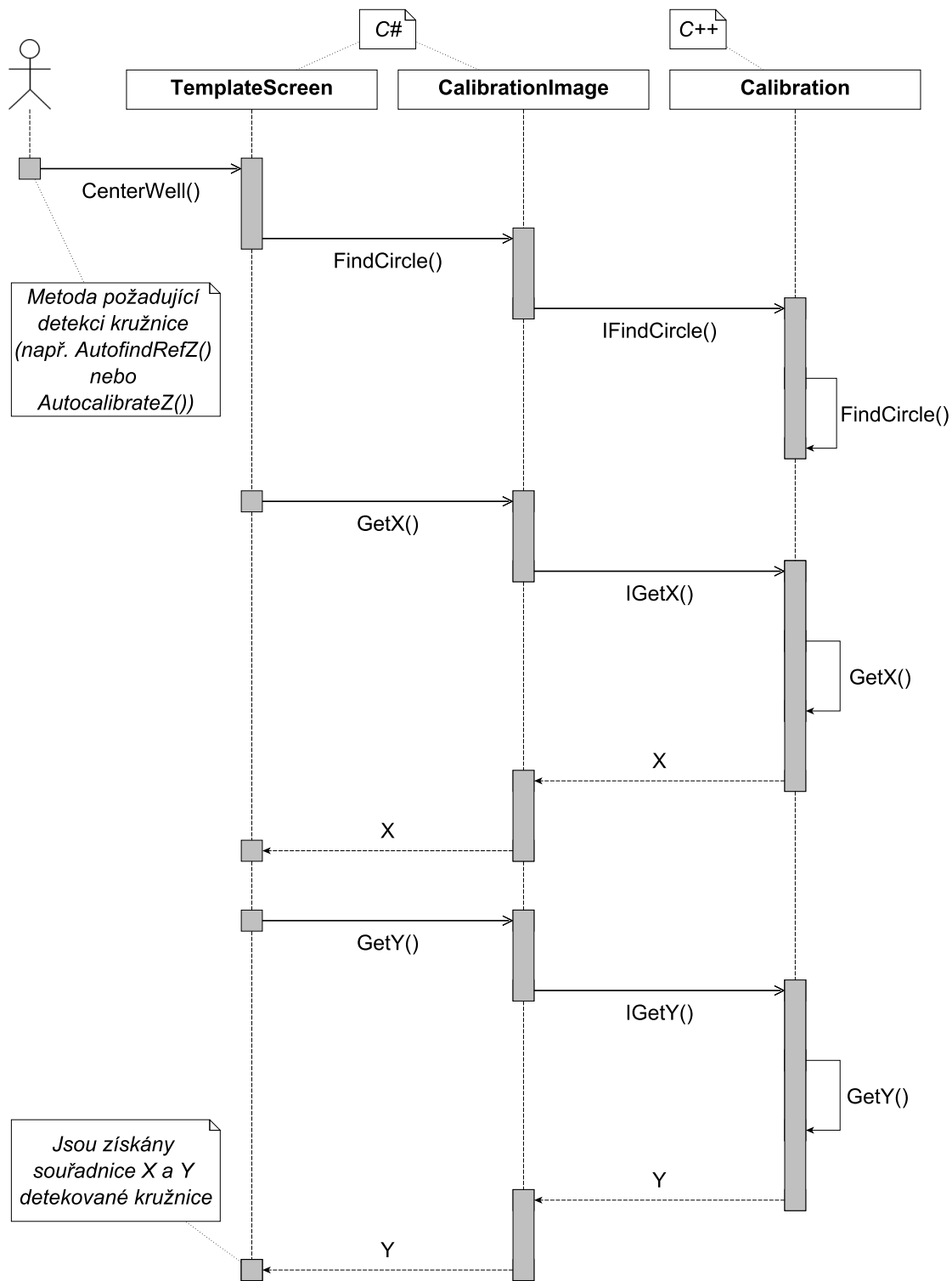
Houghova transformace je metoda, původně vytvořená pro detekci přímk v obraze, později upravena pro detekci i jiných objektů např. úseček, kružnic nebo elips.

Princip klasické Houghovy transformace spočívá v popsání jakéhokoli bodu v obraze dvěma parametry (r a σ). Parametr r představuje vzdálenost bodu od středu souřadného systému; σ potom úhel, který svírá s osou x přímka proložená středem souřadného systému a daným bodem. Rovnici přímky lze definovat jako $x \cos \sigma + y \sin \sigma = r$. Dosazením souřadnice každého pixelu v obraze do rovnice přímky dojde k vytvoření množiny všech možných řešení parametrů σ a r . Všechny body ležící na jedné přímce v obraze se protnou v Houghově prostoru v jednom bodě, které odpovídají hledaným parametrům.

Metoda `FindCircle()`

Metoda `FindCircle()` slouží k detekci kružnic v obraze na vstupu a jejich následné zprůměrování do jedné referenční kružnice.

Na vstupu metody je obraz (`Mat aImg`), průměr hledané kružnice v pixelech (`int radius`) a tolerance velikosti průměru v pixelech (`int area`). Obrázek se nejprve převede na šedotónový obraz, který je dále na vstupu metody `HoughCircles`, která pomocí Houghovy transformace detekuje kružnice o průměru v intervalu ($radius - area, radius + area$). Pokud se detekují kružnice, vyberou se tři největší a ty se průměrují (pokud je kružnic méně než tři průměruje se daný počet kružnic).



Obr. 10.10: Sekvenční diagram popisující práci s metodami při zpracování obrazu

Tento proces byl empiricky vyzkoušen a přinášel nejlepší výsledky. U kružnic se průměrují především souřadnice středů, průměry kružnic jsou nadbytečná informace. Výsledný zprůměrovaný střed je uložen do proměnné `center` typu `Point`.

Definice metody `FindCircle()` je v přílohách (výpis B.3).

Třída `CalibrationImage`

Jak již bylo řečeno třída `CalibrationImage` slouží především ke zvýšení přehlednosti kódu při práci s dynamickou knihovnou. Mezi její důležité funkce patří definice metod z dynamické knihovny. Další důležitou funkcí je uschování instance třídy `Calibration`. Ukazatel na tuto instanci se předává jako vstupní parametr při volání všech funkcí a díky této třídě není nutné ukazatel nutně ukládat do tříd, ve kterých se volají tyto metody. Definici třídy `CalibrationImage` lze nalézt ve výpisu B.4 v přílohách.

Metoda `CenterWell()`

Metoda `CenterWell()` je součástí třídy `TemplateScreen`. Je definovaná pomocí klíčového slova `protected`, což znamená že bude zpřístupněna pro všechny třídy, které dědí ze třídy `TemplateScreen`. V případě práce se jedná o všechny obrazovky.

Metoda slouží k vycentrování středu jamky fantomové destičky pod střed kamery. V kódu je důležitý objekt `CalibrationImage` (a jeho instance `calibration`).

Jádrum metody je iterativní opakování následujícího procesu. Je pořízen snímek z kamery metodou `GetSnapshotBVIBitmap()`. Ve snímku je detekována referenční kružnice již známou metodou `FindCircle()`. Dále jsou zjištěny středy detekované kružnice z objektu `calibration` metodami `GetX` a `GetY`. Vzdálenosti středu kružnice od středu obrazu je přepočtena na počty kroků motorů v osách x a y . Metoda se uplatňuje především u nalezení referenčních pozic, kde tvoří základní kámen celého procesu. Doplňkově se také používá při nalezení ideální výšky kamery, kdy po skončení procesu vycentruje jamka na střed, aby byl lépe viditelný výsledek procesu.

Výpis z metody `CenterWell()` je umístěn v přílohách (B.5).

Implementace algoritmu pro nalezení ideální výšky kamery

Princip nalezení výšky kamery spočívá v iterativním detekování kružnic a snižování výšky kamery, dokud není nalezena ideální poloha nebo není překročen počet pokusů o hledání polohy. K nalezené výšce je nutné přičíst 100 000 kroků, jelikož skutečné vzorky jsou výše, než je povrch fantomové seřizovací destičky.

Prvním krokem je najetí vozíku, tak aby byla viditelná jamka A1 (`MoveToPlatePosition(0,0)`). Na ní je doporučeno provádět seřízení. Dále je spuštěn iterativní proces. Je pořízen snímek `img` pomocí metody

(`GetSnapshotBVIBitmap()`). Ten je následně analyzován metodou `FindCircle()`. Metoda bude detekovat pouze úzký interval kružnic, jejichž průměr se od průměru jamky neliší více než o 5 pixelů. Pokud se podaří kružnici detekovat, jamka se vycentruje na střed (`centerWell()`), uloží se nalezená pozice s přičtenými 100 000 kroky (`m_miCalibrationConfigDF0.m_configuration.ReferenceZ = refZ;`) a vyplní se patřičné indikátory a upozornění o úspěšném dokončení procesu. V případě, že žádná kružnice detekována není, kamera sjede o 4000 kroků níže a proces se opakuje. Snížením výšky kamery dojde k přiblížení jamky a kružnice tak bude zabírat větší plochu obrazu a tudíž mít i větší průměr.

Zdrojový kód algoritmu pro nalezení výšky kamery je umístěn v přílohách (B.6).

Implementace algoritmu pro nalezení referenčních pozic

Základním principem algoritmu pro nalezení referenčních pozic je již zmíněná metoda `CenterWell()`. Proces pro každou z pozic je stejný; je proto umístěn ve smyčce `for`, která se provádí celkem třikrát (jednou pro každou z pozic).

V každém cyklu smyčky `for` je pozicována jedna z referenčních pozic pod kameru (příkazy `MoveToPlatePosition()` a `GoPlatePosition2D`). Přibližné polohy referenčních pozic jsou uloženy do každého nově vyrobeného zařízení. Následně je spuštěn proces centrování kamery na jamku pomocí metody (`CenterWell()`). Pokud proces úspěšně skončí vozíček pozicuje další jamku v pořadí a proces se opakuje. V případě poslední jamky se proces ukončí a uživatel je informován o úspěšném seřízení. Pokud je proces ukončen s chybou (při průběhu metody `CenterWell` nebyla jamka nalezena), skončí seřizování neúspěšně, o čemž je pracovník informován a provede seřízení manuálně.

Zdrojový kód metody pro nalezení referenčních pozic si lze prohlédnout v přílohách (výpis B.7).

11 Srovnání původní a nové aplikace

V kapitole bude srovnána nová verze aktualizované aplikace se svou předešlou verzí. Srovnání bude nejprve vycházet z časové náročnosti provedení jednoho seřízení. Časy byly experimentálně změřeny a také rozděleny do různých kategorií tak, aby byly viditelné časové náročnosti série kroků.

V druhé podkapitole poté budou poté shrnuty zásadní výhody a nevýhody obou provedení s ohledem na uživatelský komfort.

11.1 Časová náročnost

V podkapitole budou postupně představeny časové náročnosti seřízení při použití nové a původní verze aplikace. Následně bude představeno srovnání obou verzí pomocí pomyslného rozdělení obrazovek nové aplikace, tak jak korespondují seřizovací kroky s panely původní verze aplikace.

11.1.1 Časová náročnost původní aplikace

Tabulka 11.1 popisuje časové náročnosti dvou seřízení s použitím původní aplikace. Výsledky jsou zprůměrovány a poté použity do dalších tabulek. Je patrné, že druhé seřízení bylo méně časově náročnější. Tento fakt je způsoben tím, že při druhém seřízení měl přístroj oblasti, které je nutné seřizovat manuálně, již seřízené.

Tab. 11.1: Čas strávený na jednotlivých panelech ve staré aplikaci

	Časové náročnosti [s]		
	Seřízení 1	Seřízení 2	Průměr
Panel Basic	187	95	141
Panel Motors	834	564	699
Panel Lights & Camera	2546	2288	2417
Celkem	3567	2947	3257

11.1.2 Časová náročnost nové aplikace

V tabulce 11.2 jsou uvedeny vynaložené časy na seřízení jednotlivých částí přístroje dle implementovaných obrazovek. Výsledky jsou opět zprůměrovány. Na předposlední obrazovce (Bottom Camera) a poslední obrazovce (QR Configuration) nebylo nutné provádět žádné kroky, proto jsou časy velice nízké nebo nulové.

Tab. 11.2: Čas strávený na jednotlivých obrazovkách v nové aplikaci

	Časové náročnosti [s]		
	Seřízení 1	Seřízení 2	Průměr [s]
První obrazovka General Settings	41	42	41,5
Druhá obrazovka Motor Zones	143	99	121
Třetí obrazovka Z-axis Configuration	145	108	126,5
Čtvrtá obrazovka (Reference Positions) Lights	55	95	75
Pátá obrazovka Lights	2264	2317	2290,5
Šestá obrazovka Bottom Camera	2	0	1
Sedmá obrazovka Reports and Backups	9	0	4,5
Celkem	2659	2661	2660

11.1.3 Srovnání časových náročností staré a nové aplikace

Tabulka 11.3 porovnává jednotlivé kroky seřizovacího procesu (dle implementovaných obrazovek), tak aby odpovídaly panelům původní aplikace. Z těchto údajů lze názorně ukázat, při jakých krocích došlo k optimalizaci procesu.

Z hlediska prováděných seřizovacích úkonů panelu **Basic** odpovídají obrazovky **General Settings** a **Reports and Backup**. Časový rozdíl mezi starou a novou aplikací je v tomto případě nesměrodatný, jelikož nové řešení neobsahuje žádný automatizační proces. Určité zrychlení mohlo přinést pouze restrukturování komponent; uživatel mohl např. na panelu **Basic** původní aplikace strávit určitý čas hledáním některé z komponent.

Panel **Lights & Camera** je v podstatě stejný jako obrazovka **Lights**. Jakákoli časová změna je čistě náhodná.

Největší časový rozdíl je patrný mezi panelem **Motors** a obrazovkami **Motor Zones**, **Z axis configuration** a **Reference Positions**. Všechny tyto obrazovky obsahují nové automatizační procesy; čas strávený seřizováním se zkrátil o 376 sekund, což je více než šest minut. Z toho vyplývá, že zahrnutí automatizačních procesů v nové aplikaci přineslo významné zrychlení seřizovacího procesu.

11.2 Uživatelský komfort

Návrh i implementace nové aplikace probíhala za pravidelné konzultace s pracovníkem, který bude aplikaci aktivně využívat. U všech komponent na jednotlivých

Tab. 11.3: Srovnání staré a nové aplikace v jednotlivých krocích

	Časové náročnosti [s]	
	Nová aplikace	Původní aplikace
Panel Basic /aproximace obrazovek 1 a 7	46	141
Panel Lights & Camera /aproximace obrazovky 5	2290,5	2417
Panel Motors /aproximace obrazovek 2, 3 a 4	323,5	699
Celkem	2660	3257

obrazovkách je tedy ověřené, že při seřizování budou použité právě v daném kroku.

Struktura aplikace byla rovněž odsouhlasena jako nejlepší řešení různých rozvržení, i přes špatnou koordinaci aplikace při jejím restartu, který je v průběhu seřízení nutný. Tato nevýhoda však byla vyhodnocena jako přijatelná vzhledem k celkovému výsledku.

Návrh byl koncipován tak, aby nebylo nutné při seřizování přístroje nahlížet do manuálu. Vzhledem ke komplexnosti seřízení jistá navigace pracovníka při seřizování vždy bude nutná. Díky kontextovému popisu každé obrazovky však byla nutnost nahlížet do manuálu snížena na naprosté minimum.

12 Závěr

Teoretická část práce je zaměřena na klíčové informace týkající se problematiky microarray. Jsou stručně popsány chemické analyzátory (včetně analyzátorů microarray), jejich funkce a fyzikální principy. Dále se práce věnuje imunoanalytickým metodám. Na jejich základech jsou často postaveny biochemické testy, vyhodnocované právě chemickými analyzátory. V kontextu praktické části práce je vysvětlen pojem jakost, který shrnuje důležité poznatky o seřizování přístrojů, povýrobních kontrolách, normách dané problematiky apod. Závěr teoretické části se věnuje platformě .NET Framework, pomocí které bude probíhat implementace praktické části práce.

V praktické části je popsán analyzátor Array Reader C-series společnosti BioVendor a jeho stávající povýrobní kontrola. Informace byly získané po proškolení odpovědnou osobou a provedení série povýrobních kontrol.

Další kapitola praktické části představuje návrh aktualizace stávající aplikace pro seřízení. Návrh byl vytvořen na základě osobních zkušeností se seřizovacím procesem a diskuzí s osobami rutinně provádějícími povýrobní kontroly a vývojáři stávající aplikace. Kromě designu a struktury aplikace se práce zabývá metodami pro automatizaci. Automatizaci je možné využít především v nastavení kamery přístroje pomocí zpracování obrazu. Částečná automatizace je možná při nastavení koncových poloh pojezdu.

Hlavní náplní praktické části byla implementace aplikace, které splňuje všechny požadavky vytyčené návrhem. Aplikace naviguje uživatele seřizovacím procesem. Kroky nalezení optimální výšky kamery, nalezení referenčních pozic a nalezení bezpečných oblastí jsou automatizované. Do analyzátorů je ukládána historie všech seřízení a celá aplikace je vytvořena pomocí komponent společnosti BioVendor a formálně tedy odpovídá jejím standardům. K seřizovací aplikaci je rovněž k dispozici návod. V praktické části práce je implementace aplikace podrobně popsána; konkrétně struktura celé aplikace, principy její funkce a dále také algoritmy pro jednotlivé metody automatizace. Pro popsání struktury aplikace a volání metod pro zpracování obrazu z dynamických knihoven byly vytvořeny UML diagramy.

V závěru práce je srovnána původní aplikace s novou aktualizovanou aplikací z hlediska uživatelského komfortu a časové náročnosti. Aproximace časových průběhů z více měření ukázala, že se proces zrychlil téměř o deset minut. Zvýšení komfortu při seřízení bylo rovněž dosaženo; jednotlivé kroky seřízení jsou poskládány do logických celků, které na sebe navazují. Všechny kroky jsou na každé obrazovce vysvětleny. I přesto je uživatel při seřízení odkázán na návod, především kvůli komplexním mechanickým úkonům jako je např. manuální zaostření kamery. Návod byl v rámci dokončení celé práce vypracován pomocí aktualizace původního návodu pro

seřízení.

V současnou chvíli je již aplikace stálou součástí softwaru společnosti BioVendor a je aktivně využívána. Mimo bakalářskou práci byla poté ještě několika úpravami vytvořena seřizovací aplikace pro fluorescenční analyzátor a pro oba stroje byly vytvořeny také servisní verze seřizovacích aplikací. Na praktickou část bakalářské práce přímo navazuje aktualizace autodiagnostické aplikace, která je nyní v konečné fázi vývoje.

Literatura

- [1] *BMG LABTECH - Microplate reader* [cit. 5.7.2018] Dostupné z URL: <<https://www.bmglabtech.com/microplate-reader/>>.
- [2] *Wikipedia - Plate reader* [cit. 5.7.2018] Dostupné z URL: <https://en.wikipedia.org/wiki/Plate_reader>.
- [3] *jove - Introduction to microplate reader* [cit. 5.7.2018] Dostupné z URL: <<https://www.jove.com/science-education/5024/introduction-to-the-microplate-reader>>.
- [4] *The Scientist - Microarray Readers: Pushing the Envelope* [cit. 5.7.2018] Dostupné z URL: <<https://www.the-scientist.com/technology-profile/microarray-readers-pushing-the-envelope-53918>>.
- [5] *Suchá chemie I* [cit. 5.7.2018] Dostupné z URL: <https://is.muni.cz/el/1411/podzim2013/KBOMII/um/Sucha_chemie_I_2013.pdf?lang=en>.
- [6] *BMG Labtech - microplate reader catalog* [cit. 5.7.2018] Dostupné z URL: <https://www.bmglabtech.com/fileadmin/06_Support/Download_Documents/Brochures/BMG-microplate-reader-catalog.PDF>.
- [7] *moleculardevices.com - SpectraMax ABS and ABS Plus Microplate Readers* [cit. 5.7.2018] Dostupné z URL: <<https://www.moleculardevices.com/sites/default/files/en/assets/data-sheets/br/spectramax-abs-and-abs-plus-microplate-readers.pdf>>.
- [8] *biotek.com - Microplate Readers: Multi-Mode and Absorbance Readers* [cit. 5.7.2018] Dostupné z URL: <<https://www.biotek.com/products/detection/>>.
- [9] *BioVendor Instruments - Array Reader C-series* [cit. 5.7.2018] Dostupné z URL: <http://www.biovendor-instruments.com/data/flyer_arc.pdf>.
- [10] *Obrazové problémy digitální fotografie I.- Senzor* [cit. 5.7.2018] Dostupné z URL: <<http://www.fotografovani.cz/vybirame/technologie1/obrazove-problemy-digitalni-fotografie-i-senzor-152899cz?send>>.
- [11] *Fotodioda* [cit. 5.7.2018] Dostupné z URL: <http://elektross.gjn.cz/soucastky/jeden_prechod/fotodioda.html>.

- [12] *Wikipedia - Fotonásobič* [cit. 5.7.2018] Dostupné z URL:
<<https://cs.wikipedia.org/wiki/Fotonásobič>>.
- [13] *Jaroslav Reichl - Fotonásobič* [cit. 5.7.2018] Dostupné z URL:
<<http://fyzika.jreichl.com/main.article/view/747-fotonasobic>>.
- [14] *Krokový motor* [cit. 5.7.2018] Dostupné z
URL:<<https://www.mylms.cz/text-krokovy-motor-princip/>>.
- [15] HRAZDIRA, Ivo a Vojtěch MORNSTEIN. *Lékařská biofyzika a přístrojová technika*. Brno: Neptun, 2001. ISBN 80-902896-1-4.
- [16] *Spektrofotometrie (2. LF UK)* [cit. 5.7.2018] Dostupné z URL:
<[https://www.wikiskripta.eu/w/Spektrofotometrie_\(2._LF_UK\)](https://www.wikiskripta.eu/w/Spektrofotometrie_(2._LF_UK))>.
- [17] CHROMÝ, Vratislav. *Analytické metody v klinické chemii*. Brno: Masarykova univerzita, 2000. ISBN 80-210-2363-5.
- [18] *MIC Stripy - BioVendor* [cit. 5.7.2018] Dostupné z URL:
<<https://www.biovendor.cz/12-jamkove-kultivacni-desky-tp-1bal-126ks/p91.92012/>>.
- [19] *MIC Stripy - ThermoFisher Scientific* [cit. 5.7.2018] Dostupné z URL:
<<https://www.thermofisher.com/cz/en/home/clinical/clinical-microbiology/antimicrobial-susceptibility-testing/manual-antimicrobial-susceptibility-testing/mic-strips.html>>.
- [20] *Mikrotitrační destičky - PerkinElmer* [cit. 5.7.2018] Dostupné z URL:
<<http://www.perkinelmer.com/category/microplates>>.
- [21] *Mikrotitrační destičky - BMG Labtech* [cit. 5.7.2018] Dostupné z URL:
<<https://www.bmglabtech.com/which-is-the-best-microplate-for-my-assay/>>.
- [22] *Mikrotitrační destičky - ThermoFisher Scientific* [cit. 5.7.2018] Dostupné z URL: <<https://www.fishersci.com/us/en/products/I9C8KCZU/microplates.html>>.
- [23] CHMELAR, Milan. *Lékařská laboratorní technika*. Brno: Vysoké učení technické, 2000. Učební texty vysokých škol. ISBN 80-214-1770-6.
- [24] *Nefelometrie a turbidimetrie* [cit. 5.7.2018] Dostupné z URL: <https://is.mendelu.cz/eknihovna/opory/zobraz_cast.pl?cast=52957>.

- [25] *Luminscence* [cit. 5.7.2018] Dostupné z URL:
<<https://www.webchemie.cz/luminscence.html>>.
- [26] VEBER, Jaromír. *Řízení jakosti a ochrana spotřebitele*. 2., aktualiz. vyd. Praha: Grada, 2007. Manažer. ISBN 978-80-247-1782-1.
- [27] *Kvalita (jakost)* [cit. 5.7.2018] Dostupné z URL:
<<https://managementmania.com/cs/kvalita-jakost>>.
- [28] *Základní elementy VB.NET a C#* [cit. 5.7.2018] Dostupné z URL: <<https://www.dotnetportal.cz/clanek/126/Zakladni-elementy-VB-NET-a-C->>.
- [29] SHARP, John. *Microsoft Visual C# 2010: krok za krokem*. Brno: Computer Press, 2010. Krok za krokem (Computer Press). ISBN 978-80-251-3147-3.
- [30] NAGEL, Christian. *C# 2008: programujeme profesionálně*. Brno: Computer press, 2009. Programujeme profesionálně. ISBN 978-80-251-2401-7.
- [31] *Úvod do .NET Frameworku* [cit. 5.7.2018] Dostupné z URL:
<<https://www.dotnetportal.cz/clanek/125/Uvod-do-NET-Frameworku>>.
- [32] *Lekce 1 - Úvod do C# a .NET frameworku* [cit. 5.7.2018] Dostupné z URL:
<<https://www.itnetwork.cz/csharp/zaklady/c-sharp-tutorial-uvod-do-jazyka-a-dot-net-framework>>.
- [33] *Reflotron* [cit. 21.12.2018] Dostupné z URL:
<https://www.woodleyequipment.com/product/254/The-Reflotron-Plus->
- [34] *Analyzátor Roche* [cit. 21.12.2018] Dostupné z URL: http://tekmedical.co.uk/index.php?main_page=product_info&products_id=71
- [35] *Array Reader C-series* [cit. 21.12.2018] Dostupné z URL:
<http://www.labvolution.de/exhibitor/biovendor-instruments/L240369>
- [36] *Krokový motor* [cit. 21.12.2018] Dostupné z URL: <http://robotstore.cz/obchod/motory/nema17-krokovy-motor-cnc-1-8-1-68a-42x42/>
- [37] *Fotonásobič* [cit. 21.12.2018] Dostupné z URL:
<http://fyzika.jreichl.com/main.article/view/747-fotonasobic>
- [38] *CCD senzor* [cit. 21.12.2018] Dostupné z URL:
<https://www.gxccd.com/art?id=303&lang=405>

- [39] *Mikrotitrační destičky* [cit. 21.12.2018] Dostupné z URL:
http://www.medicalexpo.com/prod/deltalab/product-68179-838574.html?utm_source=ProductDetail&utm_medium=Web&utm_content=SimilarProduct&utm_campaign=CA
- [40] *Biovendor - microarray* [cit. 21.12.2018] Dostupné z URL:
<https://www.biovendor.com/file/6877/array7.jpg>
- [41] *Servomotor* [cit. 21.12.2018] Dostupné z URL:
<https://potentiallabs.com/cart/servo-motor-5kg-india>
- [42] *Elektromagnetická kompatibilita* [cit. 21.12.2018] Dostupné z URL:
<http://www.elektrorevue.cz/clanky/00025/index.html>
- [43] *Směrnice LVD* [cit. 21.12.2018] Dostupné z URL: http://ec.europa.eu/growth/sectors/electrical-engineering/lvd-directive_en
- [44] *Směrnice MD* [cit. 21.12.2018] Dostupné z URL: <http://www.unmz.cz/test/diagnosticke-zdravotnicke-prostredky-in-vitro>
- [45] *Instalační a operační kvalifikace* [cit. 21.12.2018] Dostupné z URL:
<https://slideplayer.cz/slide/2018215/>
- [46] *Procesní kvalifikace* [cit. 18.5.2018] Dostupné z URL: <https://old.vscht.cz/kot/resources/studijni-materialy/jk-p-005/prezentace.ppt>

Seznam symbolů, veličin a zkratek

CELIA	Chemiluminescent immunoassay - chemiluminiscenční imunoanalýza
CLI	Common Language Infrastructure
CIL	Common Intermediate Language
CLR	Common Language Runtime - virtuální stroj
EIA	Enzyme immunoassay - enzymoimunoanalýza v homogenním prostředí
ELISA	Enzyme-Linked ImmunoSorbent Assay - enzymoimunoanalýza v heterogenním prostředí
EMC	Electromagnetic compatibility - elektromagnetická kompatibilita
FIA	Fluorescent Immunoassays - fluorescenční imunoanalýza
FPIA	Fluorescence Polarization Immunoassay - fluorescenční polarizovaná imunoanalýza
ILMA	Immunoluminometric assay - imunoluminometrická analýza
LIA	Luminescence immunoassay - luminiscenční imunoanalýza
LVD	The Low Voltage Directive - elektrická bezpečnost
MD	Machinery directive - směrnice pro strojní zařízení
PCR	Polymerase Chain Reaction - polymerázová řetězová reakce
RoHS	Restriction of the use of certain Hazardous Substances in electrical and electronic equipment - omezení v použití zakázaných substancí v elektrických a elektronických zařízeních
TR-FIA	Time-resolved fluorescence immunoassay - fluorescenční imunoanalýza rozložená v čase

Seznam příloh

A Srovnání analyzátorů microarray Array Reader C-series a Sensovation CLAIR	81
B Zdrojové kódy	84

A Srovnání analyzátorů microarray Array Reader C-series a Sensovation CLAIR

Tab. A.1: Srovnání ArrayReaderu C-series s analyzáto-
rem Sensovation CLAIR

Název zařízení	Array Reader C-Series & BioVendor Analytics	SensoSpot Colorimetry & Array Reader Software
Rozměry zařízení	250mm x 400 mm x 230 mm	440 mm x 340 mm x 210 mm
Hmotnost zařízení	16 kg	15 kg
Využití	Mikrobiologie, multiplexní imunochemické testy, DNA-/Oligo/Proteinová array	imunochemické testy, genové exprese, proteinové array
Nosiče vzorků	mikrotitrační destičky, stripy	mikrotitrační destičky, stripy
Detektor	CMOS kamera	CCD kamera
Rozlišení kamery	5 MPx	3.8 MPx
Ostřící rovina	1-20 mm	1-10 mm
Hodnoty pixelu	černobílé, barevné (RGB)	černobílé
Rozlišení vzorku	6 μ l/pixel	6 μ l/pixel
Minimální velikost spotu	20 μ m	30 μ m
Maximální velikost spotu	6500 μ m	500 μ m
Snímaná oblast	Kruhová s průměrem 6.5 mm + stitching algoritmus	Kruhová s průměrem 6.5 mm + stitching algoritmus
Formát obrazu	PNG 24bit, JPEG, TIFF 24bit barevný, TIFF 12bit/8bit černobílý	TIFF 16bit, TIFF 8bit
Technologie osvětlení	LED s dlouhou životností	Vysokovýkonná LED
Pozicování	osy - x,y,z	osy - x,y,z
Skenovací čas	cca 2 min	2.5 min
Napájení	110-230 V, 50/60 Hz	100-240 V, 47-63 Hz
Výkon zařízení	15-25 W	Max. 150 W

Připojení k PC	USB 2.0/3.0	USB 2.0
PC vybavení	Vyžadován PC - může být dodáno PC s dotykovou obrazovkou	Integrované PC s dotykovou obrazovkou v přístroji, nebo možnost ovládat externím počítačem
Integrovaný QR/BAR scanner	Ano, na objednání	Ne
Vzdálené ovládání	Ano - TCP/IP (pouze pro diagnostiku)	Ano - TCP/IP (Plné ovládání)
Analýza array	Ano	Ano
Analýza jamky	Ano	Ne
Výběr skenovací oblasti	Ano, zadáváno v aplikaci	Ano, zadáváno graficky/textově
Přerušování skenování	Ano, lze přerušit	Ano, lze přerušit, nebo pozastavit
Korekce obrazu	Úprava světlosti a kontrastu, převod do šedotónového obrazu, inverze barev	Úprava kontrastu
Mechanismus analyzační mřížky	Nevyžaduje referenční spoty (ale podporuje je)	Vyžaduje referenční spoty
Lokalizace spotů	Automatická s manuálním ovládáním	Automatická s manuálním ovládáním
Vyřazení spotu z analýzy	Ano	Ne
Vyřazení kontury spotů	Ano	Ne
Analýzovaný tvar spotu	Kruhový, skutečný tvar	Kruhový, obrys, obdélníkový
Interpretace výsledků	Vyhodnocení výsledků je součástí analýzy	Interpretace výsledků pomocí MO Excel
Statistické parametry spotů	průměr pozadí/spot, median pozadí/spot, minimum pozadí/spot, maximum pozadí/spot, směrodatná odchylka pozadí/spot	průměr pozadí/spot, median pozadí/spot, suma pozadí/spot, směrodatná odchylka pozadí/spot

Export dat	souhrnné informace o skenu a detailní informace o každé jamce exportované do .xlsx	.xml pro každé skenování, csv soubory, TIFF pro každou jamku
Import dat	konfigurační data (XML a GAL), data o pacientech (XML), skenované obrazy (PNG,BMP,TIFF,JPEG)	konfigurační (GAL)
Možnosti mřížky	Počet sloupců a řádků, horizontální a vertikální mezery mezi jamkami	Počet sloupců a řádků, vzdálenosti spotů X a Y
Možnosti spotů	Očekávaný průměr, interval průměru (minimální a maximální)	minimální průměr, tvar spotu, vlastnosti spotu, minimální shoda s kruhem
Filtrace prachu	Ano	Ano
Vyhodnocení intenzity spotu a analýza pozadí	Ano	Ne
Analýza koncentrace spotu	Ano, pokud je k dispozici referenční koncentrace	Ne
Vyhodnocovací meze a pravidla	Ano, ano	Ne, pouze v Excelu a ne, pouze v excelu
Autodiagnostický test	Ano	Ano
Kalibrační destičky	Transparentní a matná	Ne, integrované testy

B Zdrojové kódy

Výpis B.1: Metody pro nastavení bezpečné oblasti

```
private void btnGoMinX_Click(object s, EventArgs e)      1
{                                                         2
    //Pohyb vozíku                                       3
    m_miCalibrationConfigDF0.m_miDevice.GoToPositionX(  4
m_miCalibrationConfigDF0.m_configuration.SafeMinX);    5
    //Přepsání aktuální polohy vozíku                   6
    m_miCalibrationConfigDF0.m_actX =                   7
m_miCalibrationConfigDF0.m_configuration.SafeMinX;    8
    lblCurrentXValue.Text =                             9
m_miCalibrationConfigDF0.m_actX.ToString();           10
}                                                         11
                                                         12
private void btnSetMinX_Click(object s, EventArgs e)    13
{                                                         14
    //Uložení konfigurace                                15
    m_miCalibrationConfigDF0.m_configuration.SafeMinX = 16
m_miCalibrationConfigDF0.m_actX;
    //Přepsání textových oken a změna jejich barvy     17
    lblMinXValue.Text =                                  18
m_miCalibrationConfigDF0.m_configuration.             19
SafeMinX.ToString();                                   20
    lblMinX.ForeColor = Color.Green;                    21
    lblMinXValue.ForeColor = Color.Green;               22
}                                                         23
```

Výpis B.2: Algoritmus pro nalezení bezpečných oblastí

```

//Definice asynchronní metody 1
private async void btnAutoSafeZones(object s, EventArgs e) 2
{ 3
    //Metoda, jež vytvoří instanci indikátoru průběhu. 4
    Progressor.Instance.BeginProgressMarquee( 5
    "Autoset safe zones", "Autosetting safe zones", this); 6
    //Najetí vozíkem do neutrální pozice osy y. 7
    m_miCalibrationConfigDF0.m_miDevice.GoToPositionY( 8
    m_miCalibrationConfigDF0.m_configuration.StartY); 9
    int halfX; 10
    //Najetí vozíku na koncový spínač v ose x. 11
    await Task.Run(() => 12
    m_miCalibrationConfigDF0.m_miDevice.GoToEndSwitchX_Insert()) 13
    //Najetí vozíku na druhý koncový spínač v ose x. 14
    await Task.Run(() => 15
    m_miCalibrationConfigDF0.m_miDevice.GoToEndSwitchX_Eject()) 16
    //Počet kroků mezi spínači. 17
    int endPosition = m_miCalibrationConfigDF0. 18
    m_miDevice.GetPositionMotorX(); 19
    //Polovina kroků odpovídá středu osy x. 20
    halfX = (int)Math.Round(endPosition*.5); 21
    int doorGap = (int)Math.Round(endPosition * .05); 22
    m_miCalibrationConfigDF0.m_configuration.SafeDoorX1 = 23
    halfX - doorGap; 24
    m_miCalibrationConfigDF0.m_configuration.SafeDoorX2 = 25
    halfX + doorGap; 26
    m_miCalibrationConfigDF0.m_configuration.SafeMinX = 27
    (int)Math.Round(endPosition * .05); 28
    m_miCalibrationConfigDF0.m_configuration.SafeMaxX = 29
    (int)Math.Round(endPosition * .95); 30
    m_miCalibrationConfigDF0.m_configuration.EjectX = 31
    m_miCalibrationConfigDF0.m_miDevice.GetPositionMotorX(); 32
    setLabelsX(); 33
    //Kalibrace v ose Y 34
    await Task.Run(() => 35
    m_miCalibrationConfigDF0.m_miDevice.GoToPositionX(halfX)); 36

```

```

    await Task.Run(() =>                                     37
m_miCalibrationConfigDF0.m_miDevice.                       38
GoToEndSwitchY_Insert());                                  39
    await Task.Run(() =>                                     40
m_miCalibrationConfigDF0.m_miDevice.                       41
GoToEndSwitchY_Eject());                                  42
    endPosition =                                           43
m_miCalibrationConfigDF0.m_miDevice.GetPositionMotorY();  44
    m_miCalibrationConfigDF0.m_configuration.SafeMinY =    45
(int)Math.Round(endPosition * .03);                        46
    m_miCalibrationConfigDF0.m_configuration.SafeMaxY =   47
(int)Math.Round(endPosition * .97);                        48
    m_miCalibrationConfigDF0.m_configuration.EjectY =     49
(int)Math.Round(endPosition * .95);                        50
    m_miCalibrationConfigDF0.m_configuration.SafeMiddleY = 51
(int)Math.Round(endPosition * .42);                        52
    setLabelsY();                                          53
    enableXMovement(true);                                  54
    await Task.Run(() => m_miCalibrationConfigDF0.         55
m_miDevice.GoToStartPosition());                           56
    Progressor.Instance.EndProgress(true);                57
    autosetDone(true);                                     58
    updateMotorPosition();                                  59
    return;                                               60
}                                                           61

```

Výpis B.3: Metoda FindCircle

```

void Calibration::FindCircle(Mat aImg, int radius, int area)
{
    //Dvěření vstupu obrázku
    if (aImg.empty())
    {
        m_xCoodr = -1;
        m_yCoodr = -1;
        return;
    }
    //Předzpracování obrázku
    Mat cImg;
    medianBlur(aImg, aImg, 5);
    IABase::ConvertToGrayScale(aImg, cImg);
    vector<Vec3f> circles;
    //Detekce kružnic
    HoughCircles(cImg, circles, HOUGH_GRADIENT, 1, 10, 100, 300,
radius - area, radius + area);
    if (circles.size() == 0)
    {
        aImg.release();
        cImg.release();
        return;
    }
    //Průměrování detekovaných kružnic
    Point center = { 0,0 };
    int max_circle = 0, iterations = 3;
    if (circles.size() < 1) return;
    if (iterations > circles.size())
        iterations = circles.size();
    for (int j = 0; j < iterations; j++)
    {
        for (int i = 0; i < circles.size(); i++)
        {
            if (circles[max_circle][2] <= circles[i][2])
                max_circle = i;
        }
    }
}

```



```

    if (circles[max_circle][2] > 0)                                37
    {                                                            38
        center.x += circles[max_circle][0];                    39
        center.y += circles[max_circle][1];                    40
        circles[max_circle][2] = 0;                            41
    }                                                            42
    else                                                        43
    {                                                            44
        aImg.release();                                        45
        cImg.release();                                        46
        return;                                                47
    }                                                            48
}                                                                49
center.x /= iterations;                                        50
center.y /= iterations;                                        51
m_xCoordr = center.x;                                        52
m_yCoordr = center.y;                                        53
cImg.release();                                            54
aImg.release();                                            55
return;                                                        56
}                                                            57

```

Výpis B.4: Třída CalibrationImage

```

public class CalibrationImage      1
{
    //Definice metod dynamických knihoven      2
    [DllImport("IAArrayReader.dll", CallingConvention =      3
    CallingConvention.Cdecl)]      4
    private static extern IntPtr ICalibration();      5
    [DllImport("IAArrayReader.dll", CallingConvention =      6
    CallingConvention.Cdecl)]      7
    private static extern void IFindCircle(IntPtr obj,      8
    IntPtr bmp, int radius, int radius_area);      9
    [DllImport("IAArrayReader.dll", CallingConvention =      10
    CallingConvention.Cdecl)]      11
    private static extern int IGetX(IntPtr obj);      12
    [DllImport("IAArrayReader.dll", CallingConvention =      13
    CallingConvention.Cdecl)]      14
    private static extern int IGetY(IntPtr obj);      15
    [DllImport("IAArrayReader.dll", CallingConvention =      16
    CallingConvention.Cdecl)]      17
    //Instance třídy      18
    private IntPtr m_instance = IntPtr.Zero;      19
    //Konstruktor      20
    public CalibrationImage()      21
    {      22
        m_instance = ICalibration();      23
    }      24
    //Metoda pro detekce kružnic      25
    public void FindCircle(BVIBitmap bmp,      26
    int radius, int radius_area)      27
    {      28
        IFindCircle(m_instance, bmp.Ptr, radius, radius_area);      29
    }      30
    //Získání souřadnice X detekované kružnice      31
    public int GetX()      32
    {      33
        return IGetX(m_instance);      34
    }      35
}      36

```

```
//Získání souřadnice Y detekované kružnice 37
public int GetY() 38
{ 39
    return IGetY(m_instance); 40
} 41
} 42
```

Výpis B.5: Metoda CenterWell()

```

protected bool centerWell()      1
{                                  2
    //Definice a deklarace proměnných  3
    CalibrationImage calibration = new CalibrationImage();  4
    int center_x, center_y;      5
    int prev_center_x = 0, prev_center_y = 0, steps_x = 100, 6
steps_y = 100, count = 0, errors = 0;  7
    bool first_iter = true      8
    BVIBitmap img;              9
    const double px_st = 3.7;   10
    //Cyklus ecntrování jamky    11
    while (true)                12
    {                             13
        count++;                 14
        //Získání obrázku        15
        img = m_miCalibrationConfigDF0.m_miDevice.  16
Camera.GetSnapshotBVIBitmap();  17
        //Detekce kružnic        18
        calibration.FindCircle(img,  19
m_miCalibrationConfigDF0.m_configuration.WellSize, 75);  20
        //Provede se v případě první iterace  21
        if (first_iter)         22
        {                         23
            prev_center_x = (int)Math.Round(img.Width / 2.0);  24
            prev_center_y = (int)Math.Round(img.Height / 2.0);  25
            first_iter = false;  26
        }                         27
        img?.Dispose();         28
        //získání souřadnic      29
        center_x = calibration.GetX();  30
        center_y = calibration.GetY();  31
        //Rozhodnutí o výsledku centrování  32
        if ((center_x == -1) || (center_y == -1))  33
        {                         34
            if (errors > 2) return false;  35
            else                  36

```

```

        {
            errors++;
            continue;
        }
    }
    if ((Math.Abs(steps_x) < 15) && (Math.Abs(steps_y) < 15))
        //Úspěšné ukončení
        return true;
    else if (count > 19)
        //Neúspěšné ukončení - příliš mnoho pokusů
        return false;
        else
    {
        //Upravení pozice vozíku
        steps_x =
        (int)Math.Round((prev_center_x - center_x) * px_st);
        steps_y =
        (int)Math.Round((prev_center_y - center_y) * px_st);
        if ((Math.Abs(steps_x) > 2500) ||
        (Math.Abs(steps_y) > 2500))
            return false;
        m_miCalibrationConfigDF0.m_miDevice.Move2D(
        steps_x, steps_y);
    }
}
}

```

Výpis B.6: Algoritmus pro nalezení výšky kamery

```

private async void btnAutofindRefZ(object s, EventArgs e) 1
{ 2
    //Indikátor průběhu 3
    Progressor.Instance.BeginProgressMarquee( 4
    "Autofind Reference Z", "Autofinding Reference Z", this); 5
    int refZ = 775000, count = 0; 6
    //Posun na pozici A1 7
    await Task.Run(() => 8
    m_miCalibrationConfigDF0.m_miDevice.MoveToPlatePosition(0, 0)); 9
    BVIBitmap img; 10
    CalibrationImage calibrationImage = new CalibrationImage(); 11
    //Hlavní cyklus metody 12
    while (count <= 50) 13
    { 14
        count++; 15
        //Posun referenční pozice 16
        bool success = await Task.Run(() => 17
        m_miCalibrationConfigDF0.m_miDevice.GoToPositionZ(refZ)); 18
        if (!success) 19
        { 20
            Progressor.Instance.EndProgress(true); 21
            DialogForm.WarningMessage(MessageBoxButtons.OK, 22
            Properties.Errors.BAS20012, this); 23
            return; 24
        } 25
        //Získání obrázku 26
        img = await Task.Run(() => 27
        m_miCalibrationConfigDF0.m_miDevice. 28
        Camera.GetSnapshotBVIBitmap()); 29
        //Analýza obrázku 30
        await Task.Run(() => 31
        calibrationImage.FindCircle(img, 32
        m_miCalibrationConfigDF0.m_configuration.WellSize, 5)); 33
        //Rozhodnutí o analýze 34
        if (calibrationImage.GetX() != -1 && 35
        calibrationImage.GetY() != -1) 36
    }

```

```

    {
        //Úspěšné dokončení - vycentrování jamky
        if (await Task.Run(() => centerWell()))
        {
            m_miCalibrationConfigDF0.m_configuration.StartX =
m_miCalibrationConfigDF0.m_miDevice.GetPositionMotorX();
            m_miCalibrationConfigDF0.m_configuration.StartY =
m_miCalibrationConfigDF0.m_miDevice.GetPositionMotorY();
        }
        lblCurrentZValue.Text = refZ.ToString();
        refZ = refZ + 100000;
        m_miCalibrationConfigDF0.m_configuration.ReferenceZ =48
refZ;
        lblRefZValue.Text = refZ.ToString();
        lblRefZ.ForeColor = Color.Green;
        lblRefZValue.ForeColor = Color.Green;
        updateMotorPosition();
        Progressor.Instance.EndProgress(true);
        DialogForm.MessageBox("Autocalibration result",
"Autocalibration successfully done. Reference Z was found
and added with 100 000.", this);
        return;
    }
    //Snížení referenční pozice o 4000 kroků
    else refZ -= 4000;
}
Progressor.Instance.EndProgress(true);
autosetDone(false);
return;
}

```

Výpis B.7: Algoritmus pro nalezení referenčních pozic

```

private async void btnAutofindRefPos(object s, EventArgs e)1
{
    //Indikátor průběhu
    Progressor.Instance.BeginProgressMarquee(
    "Autofind reference positions",
    "Autofinding reference positions", this);
    int i;
    for (i = 0; i < 3; i++)
    {
        //Najetí na první pozici - A1
        if (i == 0)
            await Task.Run(() =>
m_miCalibrationConfigDF0.m_miDevice.MoveToPlatePosition(
0, 0, m_miCalibrationConfigDF0.m_configuration));
        //Najetí na druhou pozici - A12
        else if (i == 1)
            await Task.Run(() =>
m_miCalibrationConfigDF0.m_miDevice.
GoToPosition2D(m_miCalibrationConfigDF0.m_configuration.
EndRowX, m_miCalibrationConfigDF0.m_configuration.EndRowY));
        //Najetí na třetí pozice - H1
        else
            await Task.Run(() =>
m_miCalibrationConfigDF0.m_miDevice.GoToPosition2D(
m_miCalibrationConfigDF0.m_configuration.EndColumnX,
m_miCalibrationConfigDF0.m_configuration.EndColumnY));
        await Task.Delay(1000);
        //vycentrování jamky
        if (false == await Task.Run(() => centerWell()))
        {
            Progressor.Instance.EndProgress(true);
            autosetDone(false);
            updateMotorPosition();
            return;
        }
        //Uložení konfigurace a přepsání textových polí

```



```

    if (i == 0)                                     37
    {                                               38
        m_miCalibrationConfigDF0.m_configuration.StartX = 39
m_miCalibrationConfigDF0.m_miDevice.GetPositionMotorX(); 40
        m_miCalibrationConfigDF0.m_configuration.StartY = 41
m_miCalibrationConfigDF0.m_miDevice.GetPositionMotorY(); 42
        setLabelInsert();                          43
    }                                               44
    else if (i == 1)                               45
    {                                               46
        m_miCalibrationConfigDF0.m_configuration.EndRowX = 47
m_miCalibrationConfigDF0.m_miDevice.GetPositionMotorX(); 48
        m_miCalibrationConfigDF0.m_configuration.EndRowY = 49
m_miCalibrationConfigDF0.m_miDevice.GetPositionMotorY(); 50
        setLabelA12();                             51
    }                                               52
    else                                           53
    {                                               54
        m_miCalibrationConfigDF0.m_configuration.EndColumnX =55
m_miCalibrationConfigDF0.m_miDevice.GetPositionMotorX(); 56
        m_miCalibrationConfigDF0.m_configuration.EndColumnY =57
m_miCalibrationConfigDF0.m_miDevice.GetPositionMotorY(); 58
        setLabelH1();                              59
    }                                               60
    }                                               61
    Progressor.Instance.EndProgress(true);        62
    updateMotorPosition();                          63
    autosetDone(true);                             64
}                                                  65

```