



AGH

AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE

Opracowanie, analiza i testy modelu komunikacji stanowiska inspekcji optycznej z nadrzędnym systemem sterowania

Wykonał zespół badawczy: dr inż. Marcin Nowak
dr hab. inż. Paweł Rotter
Piotr Lizończyk
prof. dr hab. inż. Witold Byrski



Narodowe Centrum
Badań i Rozwoju

Niniejszy raport został przygotowany w ramach projektu „Projektowanie stanowisk zrobotyzowanych wykorzystujących sterowanie ze sprzężeniem wizyjnym” w ramach programu badań stosowanych w ścieżce A, umowa nr PBS1/A9/1/2012. Wykonane badania były częścią zadania 12: „Badania metodologii projektowania nadrzędnych systemów sterowania”



Program Badań Stosowanych

AGH, Kraków 2015

Spis treści

1	Wprowadzenie	3
2	Inspekcja roztworu leku w workach.....	4
3	Propozycja transmisji danych między systemem sterowania robotem i systemem inspekcji.....	5
3.1	NIR1 - Telegramy żądania systemu nadrzędnego do stanowiska inspekcji	6
3.2	NIS1 - Telegramy statusowy systemu nadrzędnego do stanowiska inspekcji	6
3.3	INR1 Telegramy żądania wysyłane ze stanowiska inspekcji do systemu centralnego	7
3.4	INS1 Telegramy statusowy wysyłane ze stanowiska inspekcji do systemu centralnego	7
3.5	Przyporządkowanie portów i adresów	8
4	Framework sieciowy ACE	8
5	Zalecenia przy wdrożeniu	11
6	Współpraca kamery wysokiej rozdzielczości Basler z OpenCV	11
7	Słownik pojęć	13
8	Literatura	14

1 Wprowadzenie

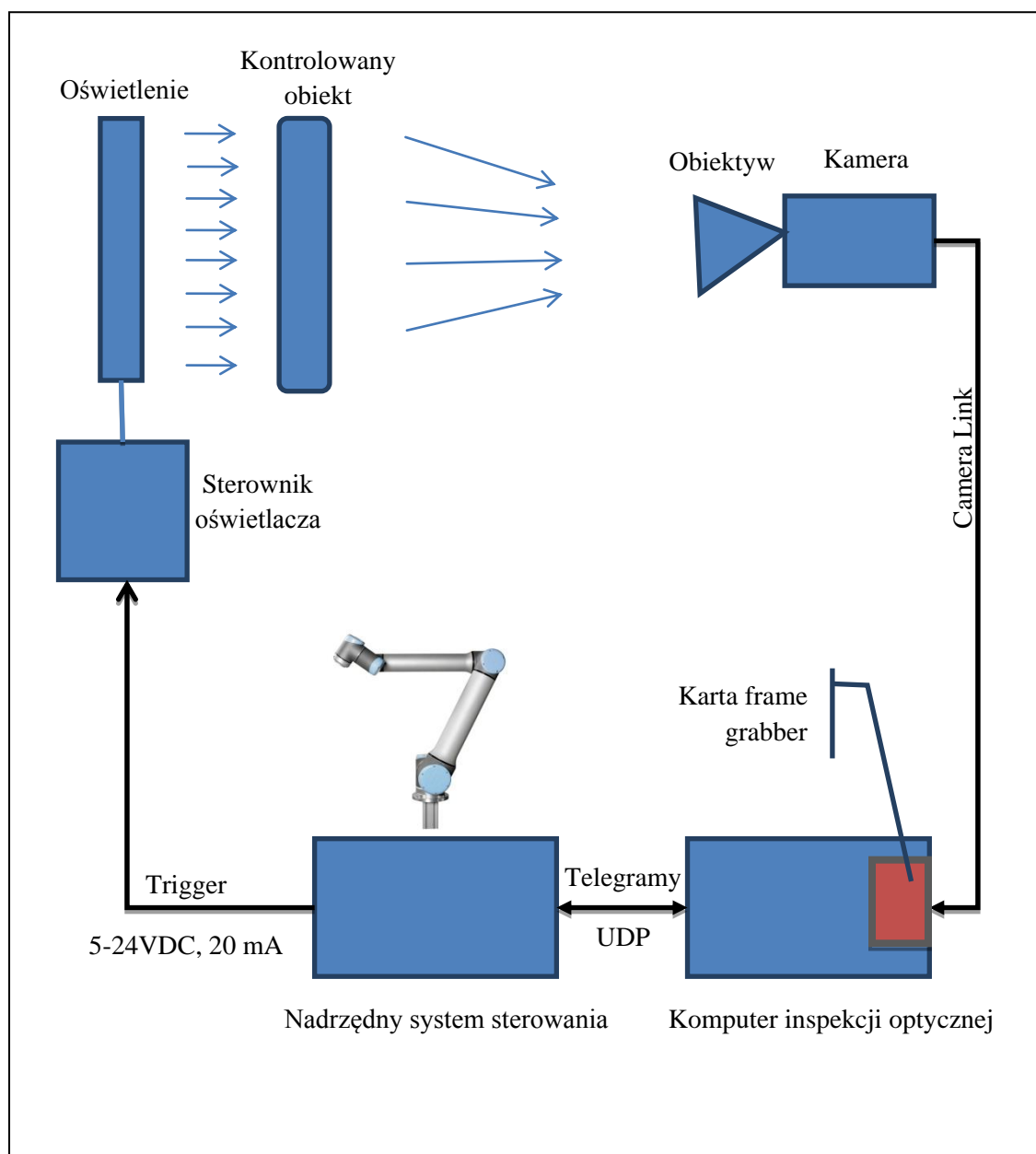
W niniejszym raporcie przedstawiono wyniki badań wykonywanych w ramach projektu „Projektowanie stanowisk zrobotyzowanych wykorzystujących sterowanie ze sprzężeniem wizyjnym”, mających na celu opracowanie wymagań dla komunikacji systemu inspekcji leków cytostatycznych z systemem nadrzędnym.

Analiza sprzętu wykonano mając na uwadze dwa główne cele systemu wizyjnego:

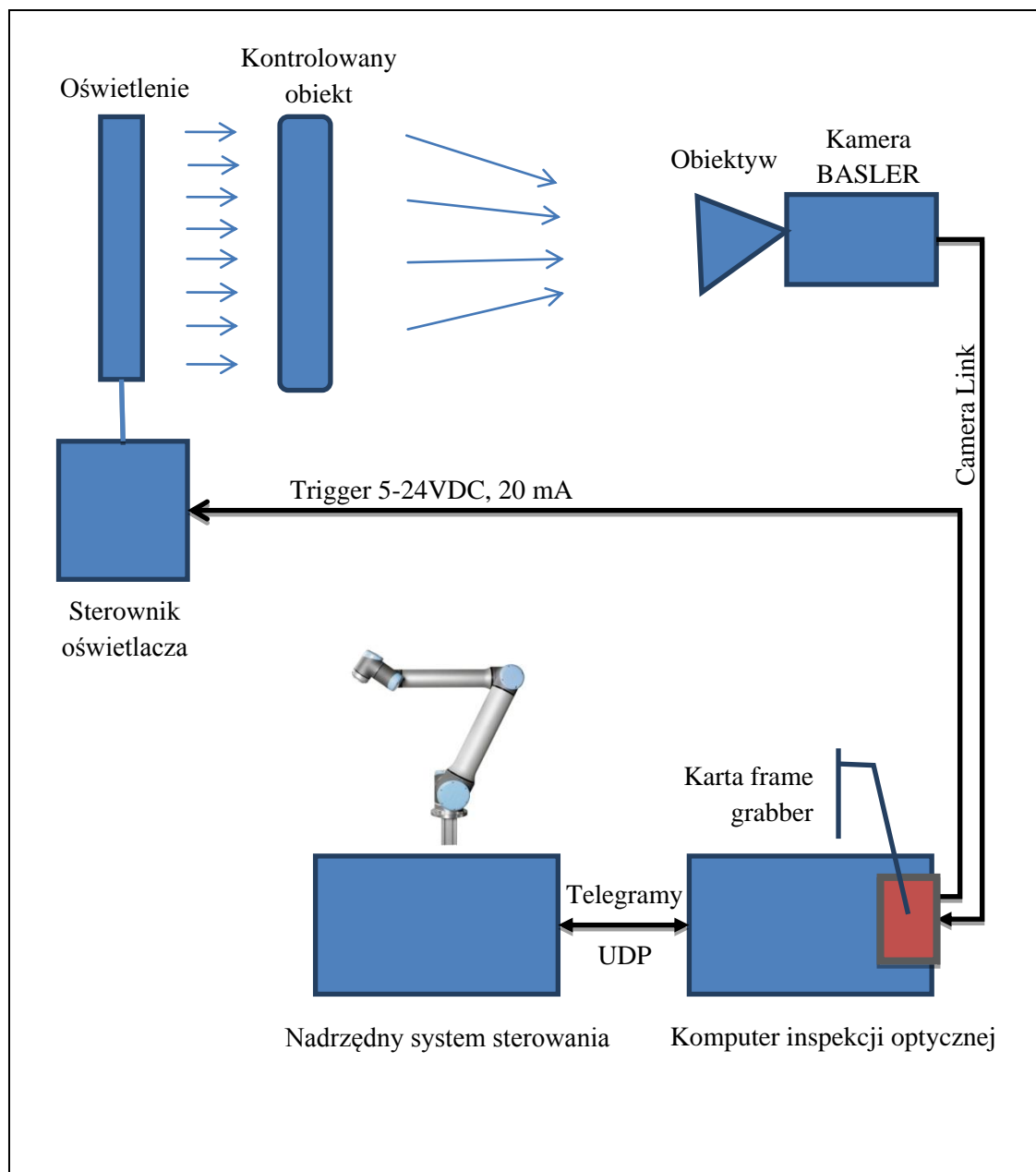
- a) Automatyczne wykrywanie zanieczyszczeń w postaci drobnych ciał stałych zawartych w roztworze. W szczególności zdarza się niekiedy, że podczas wprowadzania igły kawałek korka zostaje ułamany i dostaje się do fiolki z preparatem. W ramach obecnie obowiązujących procedur pracownik apteki sprawdza przygotowany preparat pod tym kątem.
- b) Automatyczne wykrywanie nierozpuszczonego proszku w roztworze. Potrzeba wizyjnej kontroli procesu rozpuszczania wynika z faktu, że proces rozpuszczania niektórych cytostatyków jest powolny i mieszanie roztworu przez określony czas nie gwarantuje rozpuszczenia całości preparatu.

Eksperymenty dotyczące metod przetwarzania i analizy obrazu pod kątem przydatności do celów projektu przedstawione w raporcie (Rotter, Byrski, Muroń 2013) wskazały jednoznacznie, że zanieczyszczenia, zarówno w postaci drobnych ciał stałych jak proszku, mogą być skutecznie wykrywane wyłącznie podczas ruchu, gdyż w innym przypadku zanieczyszczenia znajdujące się w roztworze są praktycznie nie do odróżnienia od obiektów występujących na powierzchni naczynia. Stąd więc opracowano projekt konstrukcji mechanicznej, która odwraca naczynie (worek lub fiolkę) z roztworem. Zanieczyszczenia są analizowane w momencie, gdy naczynie jest już nieruchome a obiekty znajdujące się w roztworze są wciąż w ruchu.

2 Inspekcja roztworu leku w workach



Rys. 1 Schemat wymiany danych komputera inspekcji optycznej z systemem nadrzędnym wersja ze sterowaniem oświetlaczem przez komputer nadrzędny.



Rys. 2 Schemat wymiany danych komputera inspekcji optycznej z systemem nadrzędnym wersja ze sterowaniem oświetlaczem przez komputer inspekcji

Inspekcja butelek z surowcem cytostatycznym.

3 Propozycja transmisji danych między systemem sterowania robotem i systemem inspekcji

Komunikacja między systemami nadrzędnymi i inspekcji optycznej odbywa się za pomocą komunikatów wysyłanych, jako pakiety UDP. Telegramy będą wysyłane cyklicznie między systemami. Częstotliwość będzie uzależniona od czasu przetwarzania algorytmu wizyjnego. Zakłada się, że nie będzie to jednak częściej niż 1 raz na sekundę.

W celu zapewnienia bezpieczeństwa wprowadzono weryfikację adres IP i port nadawcy. Pakiety UDP będą tylko odbierane i przetwarzane z poprawnie zarejestrowanego nadawcy. Weryfikacja na poziomie UDP jest zwykle niewystarczająca. Inny komputer może się podszyć pod zaufanego nadawcę. Z tego powodu dodatkowo unikalny identyfikator każdego nadawcy tzw. sól (ang. *salt*).

Sól posłuży do wyliczania MD5 i nie będzie bezpośrednio jej wartość przekazywana w telegramach. Wartość soli należy podać przy konfiguracji komunikacji. Algorytm MD5 nie zapewnia wysokiego poziomu bezpieczeństwa w porównaniu np. z algorytmem SSH1 natomiast jego wyliczenie jest stosunkowo proste i zapewniany poziom bezpieczeństwa na potrzeby komunikacji między modułami robota jest wystarczający.

Telegramy podzielono na dwie grupy statusowe i żądania. Wszystkie telegramy będą miały stały rozmiar 512 bajtów.

3.1 NIR1 - Telegramy żądania systemu nadrzędnego do stanowiska inspekcji

Tab. 1 Ramka telegramu żądania dla karuzeli wysyłana przez NSS do SIO

Offset (bytes)	Name	Length (bytes)	Function
0	Telegram ID	4	Unikalny identyfikator telegramu. Nie zmienia się w trakcie pracy systemu.
4	Source ID	4	Oznaczenie nadawcy w ASCII „NSST”
8	Destination ID	4	Oznaczenie odbiorcy w ASCII „SIOP”
16	Sequence number	4	Każdorazowo inkrementowany numer zapytania. Wykorzystywany na potrzeby detekcji przez odbiorcę do weryfikacji czy system wysyłający poprawnie funkcjonuje. Dodatkowo poprawia jakość bezpieczeństwa komunikacji wpływając na urozmaicenie pola TOKEN. Przepięnienie wartości nie jest problemem.
20	Inspection request	1	Żądanie przeprowadzenia inspekcji
	Reserved		Rezerwa na przyszłe wykorzystanie
480	Token	32	Weryfikujący przekazane dane. Wyliczany jako MD5 z soli i wszystkich powyższych pól. Wartość HEX reprezentowana w ASCII np. cffcd2919d9c8ef793ce1ac07a440eda

3.2 NIS1 - Telegramy statusowy systemu nadrzędnego do stanowiska inspekcji

Tab. 2 Ramka telegramu statusowego dla karuzeli wysyłana NSS do SIO

Offset (bytes)	Name	Length (bytes)	Function
0	Telegram ID	4	Unikalny identyfikator telegramu. Nie zmienia się w trakcie pracy systemu.
4	Source ID	4	Oznaczenie nadawcy w ASCII „NSST”
8	Destination ID	4	Oznaczenie odbiorcy w ASCII „SIOP”
16	Sequence number	4	Każdorazowo inkrementowany numer zapytania. Wykorzystywany na potrzeby detekcji przez odbiorcę do weryfikacji czy system wysyłający poprawnie funkcjonuje. Dodatkowo poprawia jakość bezpieczeństwa komunikacji wpływając na urozmaicenie pola TOKEN. Przepięnienie wartości nie jest problemem.

20	Inspection request ACK	1	Potwierdzenie przyjęcia żądania przeprowadzenia inspekcji
21	Inspection finished	1	Sygnalizacja zakończenia inspekcji. Ciągłe trwać może przetwarzanie zebranych danych pomiarowych.
22	Inspection data valid	1	Sygnalizacja zakończenia przetwarzania danych pomiarowych. Wynik inspekcji jest ważny
23	Pollution contaminant value	4	Wartość zanieczyszczenia
	Reserved		Rezerwa na przyszłe wykorzystanie
480	Token	32	Weryfikujący przekazane dane. Wyliczany, jako MD5 z soli i wszystkich powyższych pól. Wartość HEX reprezentowana w ASCII np. cffcd2919d9c8ef793ce1ac07a440eda

3.3 INR1 Telegramy żądania wysyłane ze stanowiska inspekcji do systemu centralnego

Tab. 3 Ramka telegramu żądania dla karuzeli wysyłana z SIO do NSS

Offset (bytes)	Name	Length (bytes)	Function
0	Telegram ID	4	Unikalny identyfikator telegramu. Nie zmienia się w trakcie pracy systemu.
4	Source ID	4	Oznaczenie nadawcy w ASCII „NSST”
8	Destination ID	4	Oznaczenie odbiorcy w ASCII „SIOP”
16	Sequence number	4	Każdorazowo inkrementowany numer zapytania. Wykorzystywany na potrzeby detekcji przez odbiorcę do weryfikacji czy system wysyłający poprawnie funkcjonuje. Dodatkowo poprawia jakość bezpieczeństwa komunikacji wpływając na urozmaicenie pola TOKEN. Przepelnienie wartości nie jest problemem.
20			
	Reserved		Rezerwa na przyszłe wykorzystanie
480	Token	32	Weryfikujący przekazane dane. Wyliczany jako MD5 z soli i wszystkich powyższych pól. Wartość HEX reprezentowana w ASCII np. cffcd2919d9c8ef793ce1ac07a440eda

3.4 INS1 Telegramy statusowy wysyłane ze stanowiska inspekcji do systemu centralnego

Tab. 4 Ramka telegramu statusowego dla karuzeli wysyłana z SIO do NSS

Offset (bytes)	Name	Length (bytes)	Function
0	Telegram ID	4	Unikalny identyfikator telegramu. Nie zmienia się w trakcie pracy systemu.
4	Source ID	4	Oznaczenie nadawcy w ASCII „NSST”
8	Destination ID	4	Oznaczenie odbiorcy w ASCII „SIOP”

16	Sequence number	4	Każdorazowo inkrementowany numer zapytania. Wykorzystywany na potrzeby detekcji przez odbiorcę do weryfikacji czy system wysyłający poprawnie funkcjonuje. Dodatkowo poprawia jakość bezpieczeństwa komunikacji wpływając na urozmaicenie pola TOKEN. Przepiętnienie wartości nie jest problemem.
	Reserved		Rezerwa na przyszłe wykorzystanie
480	Token	32	Weryfikujący przekazane dane. Wyliczany, jako MD5 z soli i wszystkich powyższych pól. Wartość HEX reprezentowana w ASCII np. cffcd2919d9c8ef793ce1ac07a440eda

3.5 Przyporządkowanie portów i adresów

W Tab. 5 przedstawiono przykładową konfigurację adresów IP i portów. W trakcie docelowego wdrożenia mogą ulec zmianie wartości w zależności od potrzeb.

Nazwa telegramu	Nadawca adres:port	Odbiorca adres:port	Uwagi
NIR1	192.168.67.100:2000	192.168.67.110:2001	
INS1	192.168.67.110:2010	192.168.67.100:2011	
INR1	192.168.67.110:2020	192.168.67.100:2021	
NIS1	192.168.67.100:2030	192.168.67.110:2031	

Tab. 5 Konfiguracja adresów IP i portów na potrzeby komunikacji

4 Diagram interakcji

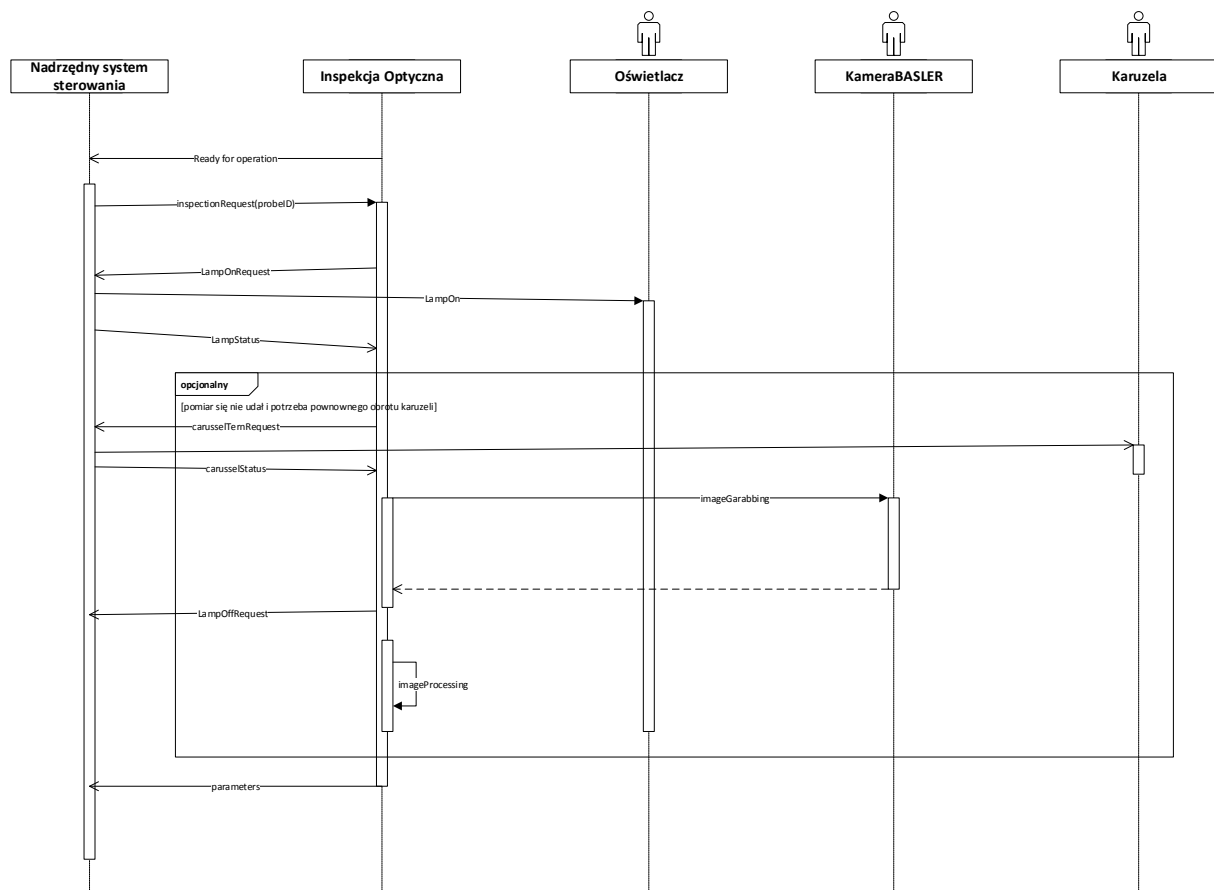
Na Rys. 3 przedstawiono diagram interakcji pomiędzy realizowanym stanowiskiem Inspekcji Optycznej a Nadrzędnym Systemem Sterowania, oświetlaczem, kamerą wysokiej rozdzielczości Basler i karuzelą obracającą przygotowany lek. Razem będą one tworzyły podsystem inspekcji gotowych leków umieszczonych w workach.

Nadrzędny System Sterowania (NSS) komunikuje się z podsystemem sterowania bezpośredniego, do którego podłączone są serwomechanizmy i czujniki pomiarowe. Podsystem inspekcyjny komunikuje się z podsystemem sterowanie jedynie za pośrednictwem Nadrzędnego Systemu Sterowania. Z tego powodu na rysunku nie uwzględniono podsystemu sterowania. NSS komunikuje się z innymi systemami aptecznymi, magazynowymi oraz systemami szpitalnymi (ang. *Hospital Information System*, HIS) w celu zrealizowania recepty.

Wszelkie dane potrzebne do pracy Stanowiska Inspekcji Optycznej będą przekazywane z NSS z użyciem protokołu UDP (Tab. 1 - żądania, Tab. 2 - statusy). SIO dane statusowe i żądania będzie przekazywało protokołem UDP (Tab. 3 i Tab. 4), a zdjęcia kontrolne z przeprowadzonych inspekcji będą zapisywane na dysk sieciowy. Pliki będą miały nazwę np. karuzela_01_20150101_12345_02.png, gdzie karuzela to oznaczenie stanowiska

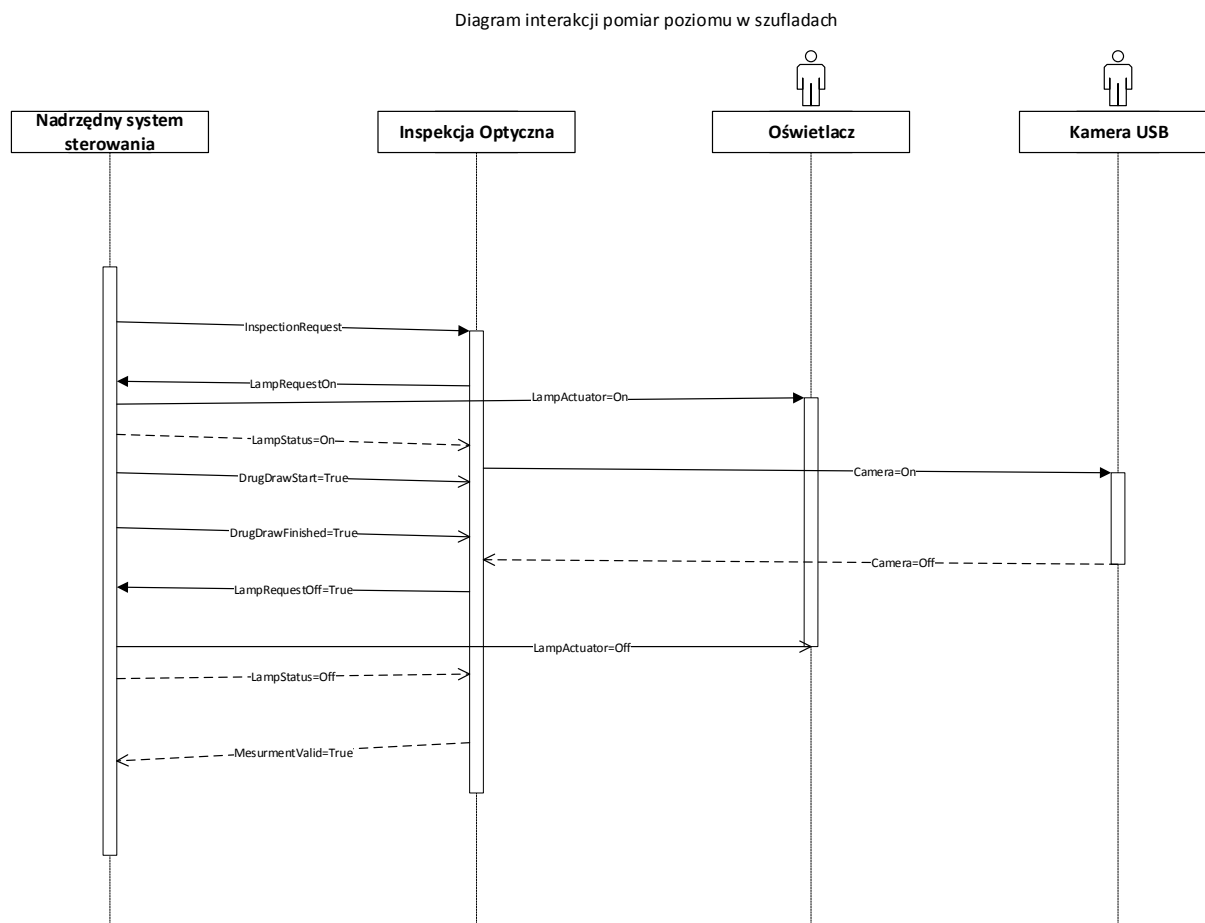
inspekcji, a 01 to numer karuzeli (u nas tylko 01), 20150101 to data inspekcji, a 12345 to identyfikator zlecenia i 02 to numer badania danej próbki. Badanie może być przeprowadzone wielokrotnie, jeżeli SIO uzna to za stosowne. Stanowisk z szufladami będzie wiele. Przykładowy zapis pliku szulada_05_20150101_12345_02.png, gdzie 05 to numer szuflady. Pozostałe elementy oznaczenia jak dla karuzeli. Maksymalna długość nazwy pliku to 255 znaków. Pliki będą zapisywane w formacie bezstratnej kompresji PNG.

Diagram interakcji pomiar zanieczyszczeń



Rys. 3 Diagram interakcji dla kamery wysokiej rozdzielczości i karuzeli

Diagram interakcji Stanowiska Inspekcji Optycznej z Nadrzędnym Systemem Sterowania, oświetlaczami i kamerami USB przedstawiono na Rys. 4. Stanowisk z surowcami umieszczonymi w szufladach jest wiele. Każda szuflada jest wyposażona w kamerę, oświetlacz, serwomechanizm napędu igły oraz inne czujniki położenia.



Rys. 4 Diagram interakcji dla kamer USB umieszczonych w szufladach

5 Framework sieciowy ACE

W budowie serwisów stanowiska inspekcji optycznej zaleca się wykorzystanie framework siechowych ADAPTIVE Communication Environment, ACE (<http://www.cs.wustl.edu/~schmidt/ACE.html>), POCO (<http://pocoproject.org/>) albo Qt (<http://www.qt.io/>).

ACE jest wysoko zoptymalizowany framework sieciowym napisany w języku C++ i dostępnym na wiele platform np. MS Windows, Linux i QNX. Biblioteka OpenCV wykorzystana do pisania algorytmów przetwarzania obrazów jest również napisana w C++ co ułatwi ich integrację w jeden program. ACE jest oprogramowaniem typu Open-Source i w ramach licencji można je bez przeszkód kopiować, modyfikować i dystrybuować zarówno w aplikacjach komercyjnych jak i nie komercyjnych bez opłat.

Podobnie POCO Project podobnie jak ACE jest nowoczesnym framework sieciowy napisany w C++ na platformy MS Windows, Linux itp. POCO wspiera takie technologie jak TCP/IP, HTTP, JSON, SSL, XML, ODBC itp. niezależnie od platformy docelowej.

Qt to środowisko rozwijane od ponad 20lat umożliwiające napisanie wymagających aplikacji zarówno na platformy typu Desktop (MS Windows, Linux i Mac) jak i urządzenia wbudowane.

Pierwotnie była to niezależna od platformy biblioteka graficznych kontrolek. Aktualnie wspiera wiele protokołów sieciowych (TCP/IP), bazy danych, grafikę 2D i 3D oraz audio.

Aktualnie na korzyść POCO przemawia to, że jest aktywnie rozwijany, znacznie mniejszy od ACE i Qt, ale oferujący wystarczającą w projekcie funkcjonalność. Koszt nauczania i opanowania tej biblioteki jest najmniejszy.

6 Zalecenia przy wdrożeniu

- Intensywność świecenia oświetlacza w sterowniku oświetlacza należy ustawić na minimum, co skutkować będzie wydłużeniem czasu życia oświetlacza. Opis przeprowadzenia strojenia intensywności można znaleźć w podręczniku użytkownika sterownika (Pollytec, 2009).
- Ustawienie unikalnych soli w systemie nadrzędnym i inspekcji wizyjnej.
- Adresy IP poszczególnych komputerów powinny być przypisane statycznie w komputerach albo stosowna rezerwacja tych adresów powinna być skonfigurowana w serwerze DHCP. Adresy IP nie powinny się zmieniać w trakcie normalnej pracy.

7 Współpraca kamery wysokiej rozdzielczości Basler z OpenCV

Na podstawie przykładu udostępnionego przez producenta karty przechwytyjącej obraz, powstał program pozwalający na odczyt obrazu z kamery w wysokiej rozdzielczości oraz dużą ilością klatek. Kod przeprowadza konwersję obrazu do postaci obiektu klasy `cv::Mat`, będącej podstawową reprezentacją obrazu w bibliotece OpenCV. Dane w takiej postaci są łatwe w dalszym przetwarzaniu.

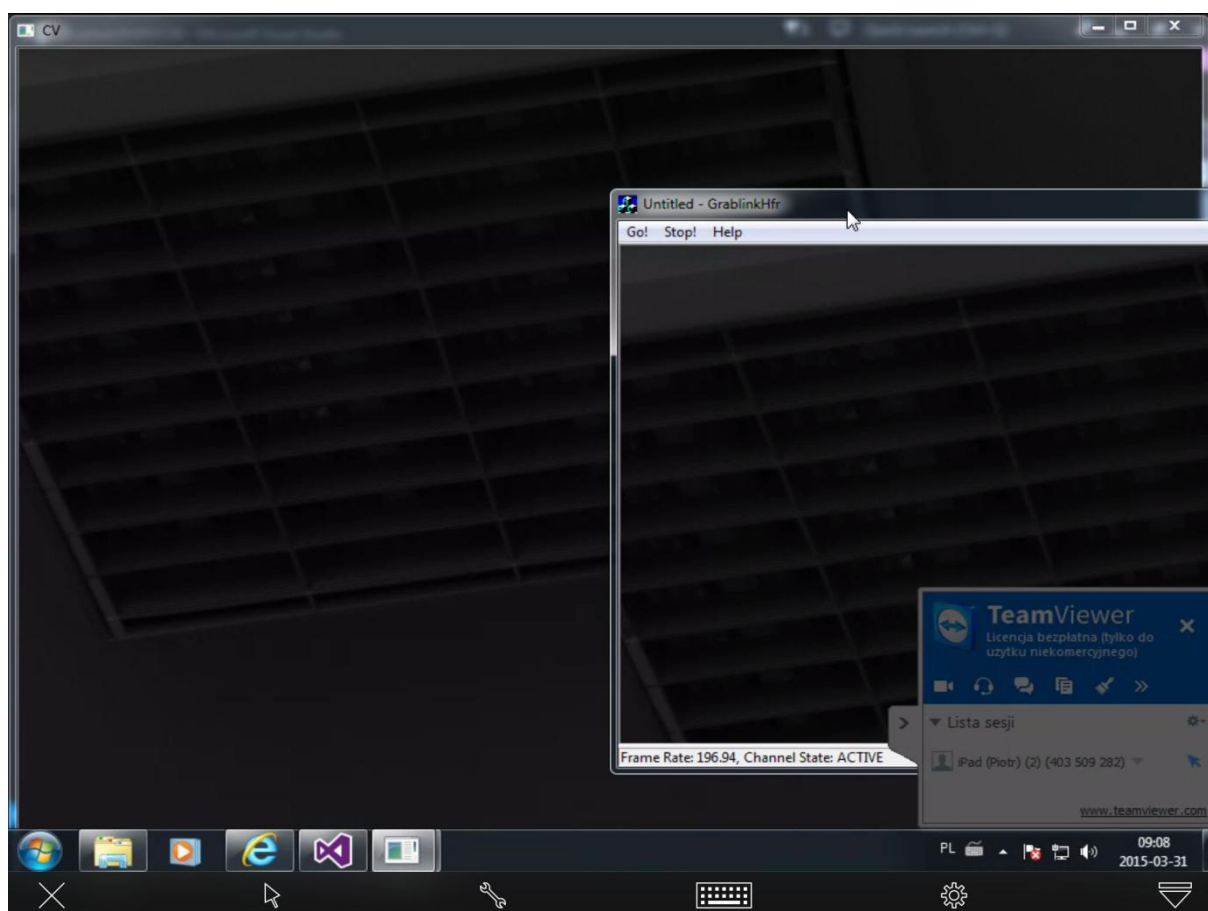
W związku ze skomplikowaną budową użytego przykładowego kodu, przeprowadzono próby zmniejszenia jego objętości (usunięcie zbędnych funkcji, próba przebudowy programu z pominięciem oryginalnej struktury), jednak udało się tego dokonać tylko częściowo. Przeszkodą okazały się bardzo silne zależności w kodzie oraz duża generyczność stosowanych klas.

```
// Retrieve image size in the document
int SizeX = pDoc->m_SizeX; // #### image width ####
int SizeY = pDoc->m_SizeY; // #### image height ####
int BufferPitch = pDoc->m_BufferPitch;

// Configure the bitmap info according to the image size
m_pBitmapInfo->bmiHeader.biWidth = BufferPitch / ( m_pBitmapInfo->
bmiHeader.biBitCount / 8); // Width = Pitch(bytes) divided by the number of bytes
per pixel
m_pBitmapInfo->bmiHeader.biHeight = -SizeY ;

// #####
// DIB - https://msdn.microsoft.com/en-us/library/windows/desktop/dd183562%28v=vs.85%29.aspx
// pDoc->m_pCurrent - ptr to data stored in array of bytes, those are uchar type
```

```
// m_pBitmapInfo - bmiHeader contains info about image (  
https://msdn.microsoft.com/pl-pl/library/aa930622.aspx ), bmiColors array of  
https://msdn.microsoft.com/en-us/library/windows/desktop/dd162938\(v=vs.85\).aspx  
  
uchar *data = (uchar*)pDoc->m_pCurrent;  
  
// construct cv Mat  
// Mat(nrows, ncols, type), we use CV_8UC1 - 8 bit x 1 channel  
cv::Mat frame(SizeY, SizeX, CV_8UC1);  
//biBitCount = 8 => The bitmap has a maximum of 256 colors, and the bmiColors  
member contains up to 256 entries. In this case, each byte in the array represents a  
single pixel.  
  
// rewrite data.  
int arrayCnt = 0;  
for (int y = 0; y < SizeY; y++){  
    for (int x = 0; x < SizeX; x++){  
        frame.at<uchar>(y, x) = data[arrayCnt++];  
    }  
}  
  
cv::imshow("CV", frame);
```



Rys. 5 Wyświetlenie przechwyconego obrazu za pomocą oryginalnego (przykładowego) kodu oraz bezpośrednio za pomocą biblioteki OpenCV

Przeprowadzenie konwersji umożliwiło uruchomienie uprzednio używanych algorytmów w czasie rzeczywistym na kamerze, która będzie docelowo użyta w robocie.

Przeprowadzenie konwersji nie zmniejsza rozdzielczości oraz ilości klatek na sekundę w przechwyconym obrazie.

8 Słownik pojęć

Sugerowany wygląd słownika pojęć:

DHCP	Protokół dynamicznej konfiguracji hostów w sieci IP (WikiDHCP)
UDP	Protokół pakietów użytkownika w sieci IP (WikiUDP)
Telegramy	Pakiety UDP wysyłane pomiędzy systemami o zdefiniowanej strukturze
System nadrzędny	Komputerowy system komunikujący się z poszczególnymi składowymi zrobotyzowanego stanowiska ze sprzężeniem wizyjnym np. sterownik robota, inspekcja wizyjna, wagi ipt. Dodatkowo odpowiedzialny za rejestrację zdarzeń i komunikację z innymi systemami biznesowymi np. zarządzaniem wytwarzaniem MES i ERP.
MD5	128-bitowa suma kontrolna stosowana w kryptografii (WikiMD5)

9 Literatura

Rotter, P., Byrski W., Muroń W. (2013) *Analiza możliwości optycznej identyfikacji zanieczyszczeń w roztworach leków cytostatycznych pod kątem określenia wymagań dla stanowiska zrobotyzowanego*. Raport techniczny, AGH, Kraków, 2013.

FujiFilm *Calculation of object size*, <http://www.fujifilm.eu/eu/products/optical-devices/cctv-and-machine-vision/lens-calculator/>

FujiFilm CCTV and Machine Vision HF25SA1 i HF35SA1 Brochure
http://www.fujifilm.eu/uploads/tx_ffproducts/files/files/HF25_35SA-1_09.pdf

FujiFilm CCTV and Machine Vision CF25HA-1 i CF35HA-1 Brochure
http://www.fujifilm.eu/uploads/tx_ffproducts/files/files/CF25_35HA-1_14.pdf

Pentax <http://security-systems.pentax.de/en/product/C32500KP>

Pollytec http://www.latab.net/fileadmin/user_uploads/Products/Lichtquellen/LED-Beleuchtung/Documents/manuals/M_PAD21135_3.pdf 2009

WikiMD5 <http://pl.wikipedia.org/wiki/MD5>

WikiDHCP http://pl.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol

WikiUDP http://pl.wikipedia.org/wiki/User_Datagram_Protocol

ACEweb <http://www.cs.wustl.edu/~schmidt/ACE.html>

ACElic <http://www.cs.wustl.edu/~schmidt/ACE-copying.html>