

**ROBOCHALLENGE**

# Lokalizacja oraz pobieranie części przy pomocy systemu wizyjnego 3D



**SICK**  
Sensor Intelligence.

**FANUC**

# ROBOCHALLENGE

## Table of Contents

1. Opis sytuacji.....	3
2. Opis zadania .....	3
3. Schemat stanowiska .....	3
4. Opis aplikacji.....	5
4.1. Struktura programu.....	7
4.2. Dodatkowe elementy stanowiska .....	7
5. Opis etapów .....	7
5.1. Etap I.....	8
5.2. Etap II.....	9
5.3. Etap III.....	9
5.4. Etap IV .....	9
5.5. Zakończenie zadania.....	11
6. System oceniania.....	11

# ROBOCHALLENGE

## 1. Opis sytuacji

Zadzwoił do Ciebie klient – Karol – kierownik do spraw rozwoju w firmie specjalizującej się w produkcji rurek aluminiowych, wykorzystywanych w przysie chłodniczym. Karol chciałby zautomatyzować pewien proces w swojej firmie. Chodzi o monotonne zadanie polegające na wkładaniu półproduktu na specjalne tacki, z których trafia on potem do centrum obróbczego, gdzie podawany jest finalnej obróbce. Półprodukt dostarczany jest luzem w skrzyniach, każda po kilkadziesiąt sztuk półproduktu. Pozycja detali w skrzyniach jest losowa – nie są one pozycjonowane, aczkolwiek zawsze oś produktu jest ustawiona wzdłuż dłuższego boku pojemnika. Karol już wcześniej zakupił robota FANUC LR Mate oraz chwytak Schunk, które chciałby wykorzystać do tego zadania. Potrzebny jest jednak również system wizyjny, pozwalający zlokalizować detal w skrzyni. Karol uczestniczył ostatnio w webinarze SICK, gdzie omawiano kamery współpracujące z robotami. Po konsultacji z lokalnym oddziałem SICK okazało się, że najlepszym rozwiązaniem będzie wykorzystanie systemu wizyjnego PLB520-M 5MP. Karol zamówił niezbędne komponenty, zamontował je i połączył z kontrolerem robota. Wgrał również przykładowy program dla robota LR Mate. Do pełnej integracji brakuje jednak kilku kroków...

## 2. Opis zadania

Zadaniem uczestników jest pomoc Karolowi w pełnej integracji robota z systemem wizyjnym.

Wyznaczono poniższe etapy zadania:

- Etap 1 – Zdefiniowanie kosza w programie PLB Engine
- Etap 2 – Dodanie pozycji chwycenia detalu w programie PLB Engine
- Etap 3 – Dodanie dodatkowych, niezbędnych pozycji chwycenia detalu
- Etap 4 – Przeprowadzenie analizy dla wielu scenariuszy ułożenia detali.

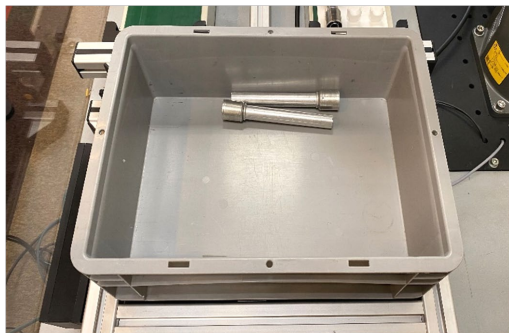
Uruchomienie programu robota, który wykorzystuje dane z systemu wizyjnego

## 3. Schemat stanowiska

Stanowisko zbudowane przez Karola to w zasadzie gotowa cela zrobotyzowana z całym niezbędnym zapleczem w postaci urządzeń bezpieczeństwa. W celi znajduje się robot FANUC LRM200ID14L-30P-M-W wraz z kontrolerem, wyposażony w chwytak SCHUNK MPG40. Po prawej stronie stanowiska są drzwi pozwalające na wymianę skrzynek z detalami.



## ROBOCHALLENGE



Bezpośrednio nad miejscem odkładania skrzynki zamontowana została kamera SICK PLB520-M 5MP.

**Commented [Br1]:** Informacja odnośnie parametrów kamery



Działa ona na zasadzie stereoskopii, czyli w jednej obudowie mamy dwie kamery o rozdzielczości 5MPx każda oraz projektor światła strukturalnego. Taki zestaw gwarantuje bardzo dobrą jakość dostarczonej chmury punktów, niezależnie od objętości, którą skanuje. Możliwe jest skanowanie obiektów małych, jak kuweta w zadaniu oraz dużych – znacznie większych od standardowej palety. Kamera współpracuje z oprogramowaniem PLB Engine, zainstalowanym na dedykowanym komputerze PLB PC.



Do komunikacji kamera – PC wykorzystano dwie karty sieciowe 1Gb/s, dodatkowa karta sieciowa zapewnia komunikację oprogramowania PLB z kontrolerem robota. W sąsiedztwie skrzynki z detalami znajdują się dwa przenośniki, na których do specjalnych tacek odkładane będą przez robota detale. Po kompletowaniu 6 detali, na dwóch tackach, przenośnik transportuje tacki do centrum obróbczego.

# ROBOCHALLENGE

## 4. Opis aplikacji

Celem zadania jest pobranie wszystkich rurek z pojemnika, przez odpowiednie skonfigurowanie aplikacji PLB Engine.

W celu przeprowadzenia integracji systemu wizyjnego PLB520-M 5MP z robotem FANUC LR Mate należy przygotować równolegle dwie rzeczy:

-skonfigurować środowisko PLB Engine pod lokalizowany detal oraz dany model Roboty; częściowo wykonane przez Karola

-wgrać dostarczany przez SICK przykładowy program do kontrolera robota: wykonane przez Karola

Po ustanowieniu komunikacji pomiędzy systemem wizyjnym a kontrolerem robota można przystąpić do procesu łączenia układów współrzędnych obu urządzeń („Alignment”). Jest to bardzo ważny etap ćwiczenia, jednak w tym zadaniu etap ten został już zakończony przez Karola.

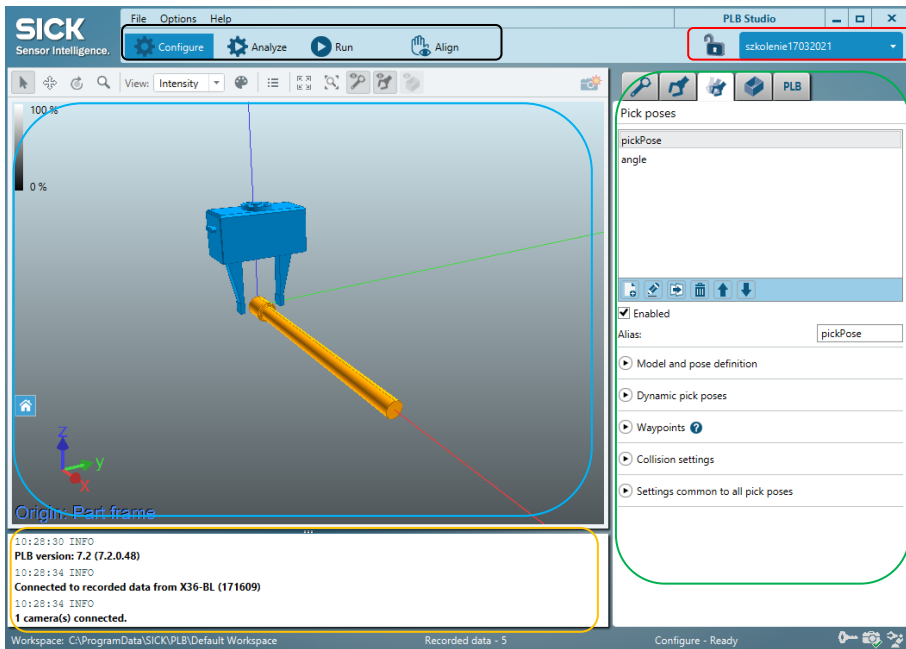
Zadaniem uczestników jest zatem dokończenie konfiguracji w oprogramowaniu PLB Engine oraz późniejsze przetestowanie działania systemu z robotem.

Pobranie przez robota wszystkich detali znajdujących się w skrzynce jest zakończeniem zadania.

Środowisko PLB Engine jest oprogramowaniem konfiguracyjnym łączącym surowe dane z kamery stereoskopowej z rozbudowanymi algorytmami lokalizacji detali oraz interfejsem komunikacyjnym dla robota. W aplikacji Karola wykorzystujemy kamerę stereoskopową 5MP do wykonania dokładnej chmury punktów sceny. Algorytm wykorzystany do lokalizacji detali to 'Cad Matching', bazujący na modelu 3D części. Odpowiedni plik CAD detalu został już wskazany w zakładce 'Part'. Karol był tak uprzejmy i ustawił również parametry samej lokalizacji, także ta część również działa. Chwytek jest już dodany w odpowiedniej zakładce. Program podczas analizy sprawdza ewentualne kolizje chwytaka z innymi elementami sceny.

Commented [Br2]: Który częściowo dostarczył SICK

# ROBOCHALLENGE



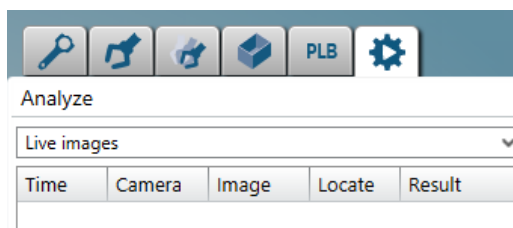
PLB Engine to w zasadzie dwa komponenty: PLB Engine – działający w tle oraz PLB Studio, czyli nakładka graficzna do konfiguracji oraz analizy. Okno programu możemy podzielić na kilka głównych części:

- Pasek wyboru trybu pracy; czarny prostokąt. Mamy tu możliwość dokonania konfiguracji zadania, następnie przeanalizowania tego co zostało skonfigurowane dla różnych scenariuszy rozłożenia elementów. Zakładka 'Run' uruchamiana jest automatycznie przez robota w trakcie normalnej pracy. W zakładce 'Align' wykonywana jest procedura łączenia układów współrzędnych kamery oraz robota. Ten etap został już wykonany przez Karola.
- Wybór zadania oraz kłódka świadcząca o ukończeniu konfiguracji; **czzerwony prostokąt**. Przed zmianą parametrów danego zadania należy otworzyć kłódkę, dopiero ta czynność pozwala edytować wszystkie dostępne parametry. Analogicznie po zakończeniu konfiguracji należy zamknąć kłódkę, aby przejść w tryb analizy.
- Zakładki konfiguracyjne; **zielony prostokąt**. Dostępne są tu najważniejsze zakładki do konfiguracji zadania. Są to:
  - Zakładka części, w której definiujemy detal, wgrujemy jego plik CAD oraz wybieramy metodę lokalizacji oraz jej parametry. Ta zakładka w całości została skonfigurowana przez Karola.
  - Zakładka chwytaka, w której dodajemy model 3D posiadanego chwytaka. Ta zakładka również jest już w pełni skonfigurowana przez Karola.
  - Zakładka pozycji chwycenia detalu, w której definiowane są pozycje w jakich chwytak ma pobrać detal. W zależności od zadania może być konieczne dodanie więcej niż jednej takiej pozycji. Nad tą zakładką będzie trzeba popracować, ponieważ Karol nie zdążył dodać żadnej pozycji chwycenia.
  - Zakładka kosza/obszaru poszukiwania, w której definiujemy obszar lub kosz, w którym kamera poszukuje części. W przypadku definicji kosza, algorytm jest w stanie

## ROBOCHALLENGE

lokalizować najpierw sam kosz, a następnie detale wewnątrz kosza. Ta zakładka wymaga wprowadzenia odpowiednich parametrów.

- Zakładka PLB, czyli definicja kamery. W naszym przypadku kamera już została podłączona przez Karola.
- Zakładka analizy:



Pojawia się w momencie przejścia w tryb analizy. Pozwala na zrobienie nowego zdjęcia:



Zlokalizowanie kosza (jeśli został zdefiniowany):



Oraz przeprowadzenie analizy lokalizacji i pobrania części:



Wyniki wyświetlane są w karcie zakładek w postaci listy:

12:49:29	15		
12:49:36	15		  

Prawidłowa lokalizacja kosza sygnalizowana jest jedną fajką, a lokalizacja i gotowość pobrania części trzema fajkami.

- Widok 3D; **niebieski prostokąt**. Może pokazywać aktualną scenę lub modele 3D części, chwytaka czy pozycji chwycenia.
- Konsola; **pomarańczowy prostokąt**. Podgląd komunikacji z robotem.

### 4.1. Struktura programu

Program robota wykorzystany w zadaniu będzie w całości bazował na przykładzie dostarczonym przez SICK do tego modelu robota. Wszystkie zmienne, interfejsy i podprogramy zostały już odpowiednio skonfigurowane przez Karola.

### 4.2. Dodatkowe elementy stanowiska

Na stanowisku dostępna będzie kamera PLB520-M 5MP oraz komputer - PLB PC. Na komputerze zainstalowane jest oprogramowanie PLB Engine, w którym należy wykonać zadanie.

## 5. Opis etapów



# ROBOCHALLENGE

## 5.1. Etap I

Zdefiniowanie kosza w programie PLB Engine

Na komputerze otwieramy oprogramowanie PLB Engine.

-sprawdzamy, czy nastąpiło automatyczne połączenie komputera z kamerą oraz czy jest wgrany plik licencji. Status poprawnego połączenia oraz aktualnego pliku licencji znajduje się w prawym dolnym rogu aplikacji PLB Engine:



Jeżeli dane zadanie zakończone jest sukcesem przy ikonie pojawia się fajka.

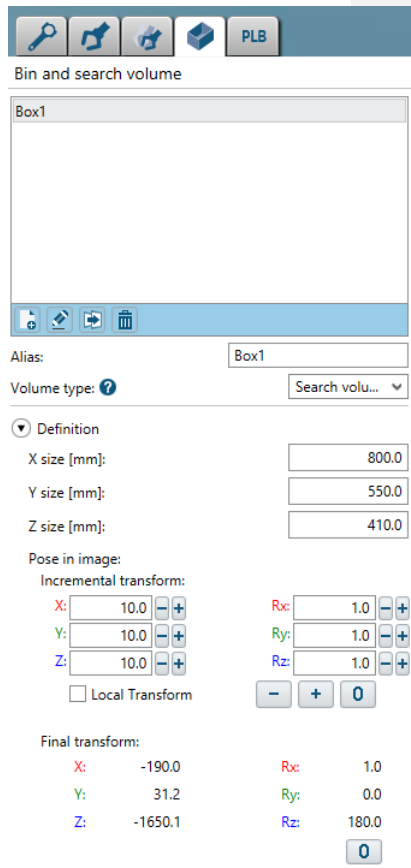


Trzecia dostępna ikona informuje o połączeniu z robotem.

-w zakładce 'Bin and search volume' definiujemy odpowiedni typ obiektu oraz jego wymiary i pozycję. Do zmierzenia pojemnika na stanowisku dostępna będzie miarka. Mierzymy wymiar wewnętrzny kosza. Pozycja pojemnika jest ustalana poprzez wpisanie zadanego przesunięcia lub kąta w 'Incremental transform' i kliknięcie znaku '+' lub '-'. Wynik działania powinien od razu być widoczny na obrazie 3D

-parametr 'Collision plane offset' w sekcji 'Lower plane' ustawiamy na -10.0

-w celu sprawdzenia działania algorytmu przechodzimy do trybu 'Analyze' gdzie można wykonać zdjęcie i sprawdzić lokalizację skrzynki





# ROBOCHALLENGE

## 5.2. Etap II

Dodanie pozycji chwycenia detalu w programie PLB Engine

-W trybie 'Configure' oraz z otwartym zadaniem:  przechodzimy do zakładki 'Pick Poses'



-na bazie dodanego wcześniej przez Karola chwytaka (nazwa chytaka „3”) należy przygotować podstawową pozycję chwycenia detalu. Chwytek znajduje się pionowo nad pobieraną częścią. Detal na środku pojemnika.

-Proponowane nazewnictwo pozycji chwycenia to kolejne cyfry

-W zakładce 'Model and pose definition' ustawiamy pozycję chwytaka względem części dla momentu zamknięcia palców efektora. Chwytek znajduje się pionowo nad pobieraną częścią.

-W zakładce 'Dynamic pick poses' zaznaczamy opcję 'Cylindrical around x-axis at pick pose', ponieważ obiekt posiada oś obrotu pokrywającą się z osią X

-W zakładce 'Collision settings' ustawiamy 'Point threshold' na 50

## 5.3. Etap III

Dodanie dodatkowych, niezbędnych pozycji chwycenia detalu

-Na bazie doświadczenia z poprzedniego etapu oraz pamiętając, że aplikacja PLB Engine nie pozwala na kolizje chwytaka z koszem, należy przemyśleć jakie inne pozycje chwycenia są niezbędne do opróżnienia kosza. Szczególna uwagę należy zwrócić na detale znajdujące się w narożnikach.


-Wprowadzamy dodatkowe pozycje chwycenia zgodnie z instrukcją z Etapu II.

-Dla dodatkowych pozycji może zająć konieczność zwiększenia 'Point threshold' nawet do 250

## 5.4. Etap IV

Przeprowadzenie analizy dla wielu scenariuszy ułożenia detali. Uruchomienie programu robota, który wykorzystuje dane z systemu wizyjnego

-Przechodzimy w tryb 'Analyze', wykonujemy zdjęcie, lokalizujemy skrzynkę.

-Korzystamy z opcji 'Locate part with breakpoints' , by sprawdzić czy dodane pozycje działają poprawie.

# ROBOCHALLENGE

### Locate part result

(Breakpoint: All steps)

Job name: szkolenie17032021

Image ID: 1

Status: ERROR (11) "The volume of interest is empty."

Zone: Full bin volume

Part pose:

X:  Rx:

Y:  Ry:

Z:  Rz:

Displayed pick pose:

**Localization details**

Average distance to model:  (1.5)

CAD/Located object match [%]:  0.0 (70.0)

Portion of points inside:  (2.5)

Measured point distance:  0.89

Surface visible to camera:

Length:  (50.0)

Surface:  (n/a)

**Overlap details**

Width:  (10.0)

Length:  (50.0)

Area:  (1000.0)

**Pick pose collision details**

Name	Points ...	Length	Width	Area
No content in table				

### Locate part analysis

Zone: Full bin volume

Locate using breakpoint:

- Localization
- Overlap
- Collision
- All steps
- Localization, next part
- Overlap, next part
- Collision, next part
- All steps, next part

from first image.

Recorded data - 5

W dwóch nowo otwartych oknach korzystamy z przycisku 'All steps'. W oknie 'Locate part result' oraz na liście części po prawej stronie aplikacji PLB widzimy wyniki działania programu. Jeśli w wyniku otrzymujemy błąd, tak jak w powyższym przykładzie, cofamy się do poprzedniego etapu i dokonujemy poprawek.

-Wykonujemy analizę dla trzech ustawień detali proponowanych przez prowadzącego.

## ROBOCHALLENGE

-Zamykamy zadanie po stronie oprogramowania PLB Engine - Kłódka

-W kontrolerze robota uruchamiamy program PLB\_LOOP

### 5.5. Zakończenie zadania

Jeśli wszystkie etapy zostały zakończone sukcesem, robot powinien w pętli lokalizować, pobierać i odkładać detale, aż do ich wyczerpania, co jest równoznaczne z zakończeniem zadania.

## 6. System oceniania

### 1. Osiągnięcie celu zadania – 1 pkt

Celem zadania jest opróżnienie skrzynki z detali z wykorzystaniem systemu wizyjnego.

### 2. Zaliczenia etapów – łącznie 4 pkt

- Etap I (1 pkt)
- Etap II (1 pkt)
- Etap III (1 pkt)
- Etap IV (1 pkt)

### 3. Czas wykonania po zapoznaniu się z instrukcją – maksymalnie 3 pkt

- Ukończenie zadania poniżej 30 min (3 pkt)
- Ukończenie zadania poniżej 35 min (2 pkt)
- Ukończenie zadania poniżej 40 min (1 pkt)

Komentarz: w trakcie wykonywania zadania, drużyna może poprosić o pomoc w wykonaniu danego punktu opiekuna stanowiska. Nie przysługuje jej wówczas punkt za wykonanie danego etapu. Warunkiem udzielenia pomocy, jest stosowna ilość czasu na jej udzielenie.