

Dokumentacja do projektu z przedmiotu

*Technologie Informatyczne w Klasycznym i Inteligentnym Sterowaniu Produkcją*

**Temat: Sortowanie elementów**

LP1, EADI-3

Bazan Arkadiusz

Bełch Jakub

Bizior Mateusz

Rzeszów, 24 III 2017

1. Cel i zakres Projektu

Celem projektu było powiązanie robota firmy Kawasaki sortującego elementy na podstawie kodów QR z oprogramowaniem Wonderware MES który generował zlecenia gdzie zadawaliśmy ilość elementów do posortowania.

Dodatkowo przeorganizowaliśmy stanowisko pod kątem przyszłych projektów – organizacja i opisanie kabli, zwiększenie liczby wejść i wyjść, spięcie wszystkiego switchem ethernet.

1. Zastosowane elementy

* Stanowisko laboratoryjne – rama, terminale, moduly I/O, zasilacz itp
* Robot Kawasaki FS03N + kontroler serii D+
* System wizyjny Balluff – kamerka BVS-Universal oraz Monitor BVS
* Sterownik PLC Beckhoff
* Taśmociąg – nie działający sprawnie, proces podawanie elementów przez taśmociąg został zasymulowany ręcznie
* Switch sieciowy - dzięki niemu wszystkie potrzebne nam elementy systemu podłączyliśmy do jednej sieci w której mogliśmy się z każdym urządzeniem.
* Laptop z zainstalowanym oprogramowaniem Wonderware MES oraz jego komponentami.

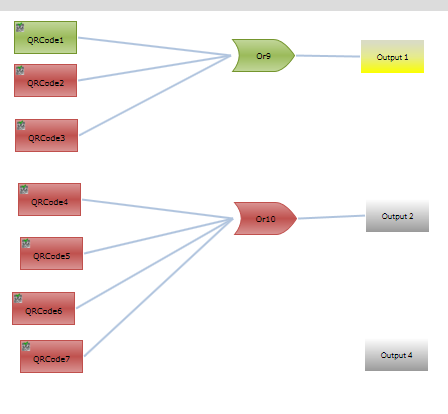
1. Realizacja zadania

3.1 Strona techniczna

I. Rozpoznawanie wadliwych i dobrych elementów

Do realizacji rozpoznawania wadliwych i dobrych elementów została wykorzystana kamerka firmy Balluf. Na początku należało przypisać odpowiednie kody QR do wybranych elementów, ustawić rodzaj wyzwalania, doświetlenie oraz zrealizować algorytm przypisywania. Wszystkie te czynności zrealizowaliśmy za pomocą oprogramowania producenta BVS\_ConVis\_1.5.2. QR kody podzieliliśmy na dwie grupy na dobre i wadliwe.

Kamerka była wyzwalana zewnętrznym sygnałem pochodzącym z bramki laserowej na taśmociągu. Gdy czujnik laserowy wykrył element wtedy wysyłał sygnał do kamerki która w tym czasie robiła zdjęcie i rozpoznawała element. Odpowiednio do rozpoznanego elementu w zależności dobry czy zły czy dobry wysyłała sygnał na wejście robota Kawasaki.



Rys.1 Algorytm grupowania QR kodów

II. Sortowanie elementów

Sortowanie elementów było wykonywane za pomocą robota Kawasaki. Robot uruchamiał swoją pracę gdy otrzymał sygnał pochodzący z oprogramowania MES. Zależnie od sygnału pochodzącego od kamerki BVS, robot zabierał i odkładał element w ustalone miejsce, złe elementy w jedno miejsce, dobre w drugie. Z każdym posortowanym elementem robot wysyłał sygnał na wejście sterownika PLC informując czy był to dobry element czy zły. Program znajdujący się w sterowniku Beckhoff liczył dobre elementy, jeśli została osiągnięta oczekiwana ilość dobrych elementów, sterownik wysyłał sygnał do zatrzymania pracy robota.

Zasada i algorytm działania programu została opisana w komentarzach, ponadto program informuje o swoim stanie i czynnościach na terminalu i kontrolerze. Posiada również tryb serwisowy niezależny od MES polegający na symulacji czyszczenia kamerki.

**Opis działania programu:**

Program czeka w MS i czeka na sygnał z MES ze rozpoczęto zlecenie, po czym po rozpoznaniu przez kamerkę i otrzymaniu z niej sygnał co to za przedmiot robot jedzie SN nad PP, po czym WD w PP, włącza ssawkę i trzymając przedmiot jedzie WD powyżej PP, po czym SN w punkt pośredni i nad B1 (B2), po czym WD w B1(B2) puszcza przedmiot, przesuwa go sobie aby zrobić miejsce na kolejny, następnie SN jedzie do MS i czeka na kolejny sygnał.

**skróty:**

PP – punkt podniesienia na taśmociągu

MS – miejsce spoczynkowe nieco obok osi kamerki aby nie zasłaniać i powyżej PP

MES – system MES

WD – jazda wolna i dokładna

SN – jazda szybka niedokładna, albo aproksymowana

B1, B2 – pudelka nr1 i nr2

Funkcja PutToBox realizuje operacje przenoszenia, a główny program steruje przekazywaniem parametrów do niej.

**Kod programu:**

.**JOINTS** *;jakies tam kluczow epunkty w przestzeni*

#nearKMpickPlace -46.498207 48.842892 -98.645340 179.470032 34.085030 -100.820503*;troche obok i nad*

*miejscem podniesienia zeby nei zaslaniac kamerki*

#station1 -8.444472 43.699108 -105.400833 177.728149 31.977776 -137.378860*;stacja suszenia*

#KMpickPlace -44.783035 98.540512 -18.469378 180.000763 62.996120 -101.335068*;miejsce podnoszenia*

#box1 -15.415906 65.657791 -87.997955 177.694534 27.555635 -130.300781*;pudelko nr 1*

#box2 8.997740 70.355530 -74.744057 177.440689 35.538418 -154.698914*;pudelko nr2*

#cleanpos -44.787865 52.462143 -23.704908 -0.000659 76.153870 -9.961446*;pod kamerka*

.**END**

.**TRANS**

cleanpos -395.219879 398.156036 292.094330 137.567856 0.013197 -102.741600

p0 -319.160187 303.462402 -126.257584 -32.802105 178.400070 179.436188

p4 -332.167725 382.277496 214.104309 129.135468 49.154022 -54.130322

ak 18.222315 442.180542 9.993858 -94.911812 179.998077 117.328278

bk 16.331390 443.385406 9.991417 -80.919800 179.997589 131.319717

.**END**

.**PROGRAM** p1

*;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;*

*;;; program do oblugi kawasaki i MES*

*;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;*

*;*

*;##### ustawienia poczatkowe*

lowSpeed**=**10*;*

highSpeed**=**80*;*

lowAccuracy**=**100*; w[mm]*

highAccuracy**=**2*; w[mm] przy acc=1 robi przerwy w sciezce ruchu*

*;*

*;;;sygnaly numery INPUT 1001-1032*

INsigItem1**=**1005*;IN sygnal z kaemrki ze rozpoznno kod nr1-3 czyli typ1*

INsigItem2**=**1006*;IN sygnal z kaemrki ze rozpoznno kod nr4-7 czyli typ2*

INsigRunRobot**=**1007*;sygnal z systemu MES ze robot rozpoczac program*

INsigCleanCam**=**1008*;sygnal poza MES ze robot ma wyczyscic kamerke*

*;*

*;;;sygnaly numery OUTPUT 1-32*

OUTsigItem1Recognized**=**3*; rozpoznano item typu 1*

OUTsigItem2Recognized**=**4*; rozpoznano item typu 2*

OUTsigItem1InBox1**=**5*; item1 w pudelku1*

OUTsigItem2InBox2**=**6*; item2 w pudelku2*

*;*

**SPEED** highSpeed **ALWAYS**

**ACCURACY** lowAccuracy **ALWAYS**

*;*

*;jazda w poblize KMpickPlace*

*;HOME*

**OPENI**

**jappro** #nearKMpickPlace, lowAccuracy*;*

**OPENI**

PRINT *"BOMBOWO START rev3";*

*;;;to potem skasuje tylko do testow*

*;item1=false;*

*;item2=false;*

item1**=SIG (**INsigItem1**)***;*

item2**=SIG (**INsigItem2**)***;*

**while true do**

*;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;*

*;item1=not item1;*

*;item2=not item2;*

item1**=SIG (**INsigItem1**)***;*

item2**=SIG (**INsigItem2**)***;*

RunRobot**=SIG (**INsigRunRobot**)***;*

cleanCam**=sig(**INsigCleanCam**)***;*

*;RunRobot=true;*

*;cleanCam=false;*

*;;cleanCam=true;*

*;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;*

**if not** RunRobot **then**

**jmove** #nearKMpickPlace*;*

**else**

**if** item1 **then**

**PULSE** OUTsigItem1Recognized, 1*;*

$itemName **=** *"Item1";*

$targetBoxName**=***"box1";*

targetSignal**=**OUTsigItem1InBox1*;*

**POINT** #targetPlace**=**#box1*;*

PRINT *"klocek "***+**$itemName **+** *"Arrives";*

**CALL** PutToBox

item1**=false***;*

item2**=false***;*

**end**

**if** item2 **then**

**PULSE** OUTsigItem2Recognized, 1*;*

$itemName **=** *"Item2";*

$targetBoxName**=***"box2";*

targetSignal**=**OUTsigItem2InBox2*;*

**POINT** #targetPlace**=**#box2*;*

PRINT *"klocek "***+**$itemName **+** *"Arrives";*

**CALL** PutToBox

item1**=false***;*

item2**=false***;*

**end**

**end**

**if** cleancam **then**

**accuracy** highAccuracy **always***;*

**speed** highSpeed **always***;*

**jappro** p0,lowAccuracy*;*

**jappro** p4,lowAccuracy*;*

**jmove** cleanpos*;*

**accuracy** lowAccuracy **always***;*

**SPEED** lowSpeed**\***0.75 **always***;*

**draw** 40,40,**-**20*;*

**draw -**40,**-**40,20*;*

**draw** 40,40,**-**20*;*

**draw -**40,**-**40,20*;*

**draw** 40,40,**-**20*;*

**draw -**40,**-**40,20*;*

**speed** highSpeed **always**

**jmove** #nearKMpickPlace*;*

**end**

**end**

.**END**

.**PROGRAM** PutToBox

*;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;*

*;;; JAZDA PO PRZEDMIOT*

*;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;*

**ACCURACY** highAccuracy **ALWAYS***;*

**SPEED** lowSpeed **ALWAYS***;*

*;jazda szybka niedokladna 20mm nad KMpickPlace*

**POINT** KMpickPlace**=**#KMpickPlace

**POINT** aboveItem**=SHIFT (**KMpickPlace **BY** 0,0,20**)**

PRINT *"jazda nad klocek "***+**$itemName*;*

**ACCURACY** lowAccuracy*;*

**SPEED** highSpeed*;*

**jMOVE** aboveItem*; chwytak nad przedmiotem*

*;*

*;jazda w KMpickPlace -> zalaczenie chywtaka -> 20mm nad KMpickPlace*

**jMOVE** KMpickPlace*; w item*

**STABLE** 0.1*; czekanie 0.1s po znieruchomieniu chwytaka*

PRINT *"biere klocek "***+**$itemName*;*

**CLOSEI** *;ZAMYKANIE CHWYTAKA*

**jMOVE** aboveItem*; chwytak nad przedmiotem*

*;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;*

*;;; PRZEDMIOT W CHWYTAKU JAZDA DO PUDELKA*

*;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;*

**ACCURACY** lowAccuracy **always** *;*

**SPEED** highSpeed **always***;*

*;jazda szybka niedokladna do puntu posrdeniego (suszenie?)*

PRINT *"jazda nad pudelko "***+**$targetBoxName*;*

**jAPPRO** #station1, lowAccuracy

*;twait 0.5; takei jkaby suszenie*

*;pulse sigInStation1, 0.2 ;impuls na wyjscie*

PRINT $itemName**+***" wysuszony";*

*;*

*;jazda szybka niedokladna 50mm nad pudelko*

**POINT** targetPlace**=**#targetPlace*;*

**POINT** aboveItem**=SHIFT (**targetPlace **BY** 0,0,50**)**

*;*

*;jazda szybka niedokladna 50mm nad pudelko*

**jMOVE** aboveItem*;*

*;*

*;jazda w pudelko -> wylaczenie chywtaka -> przesuniecie przedmiotu*

**ACCURACY** highAccuracy **always** *;*

**SPEED** lowSpeed **always***;*

**jMOVE** targetPlace*;*

**STABLE** 0.1*;*

PRINT *"wdupcam "***+** $itemName **+** *" do "* **+** $targetBoxName*;*

**OPENI***;otwieranie chwytaka*

**PULSE** targetSignal, 1*;*

*;TWAIT 0.2;*

*;*

*;przesuwanie klocka*

**ACCURACY** lowAccuracy **ALWAYS***;*

**SPEED** lowSpeed**\***3 **ALWAYS***;*

**POINT** pt**=SHIFT (**targetPlace **BY** 0,0,10**)***;*

**jMOVE** pt*;*

**POINT** pt**=SHIFT (**pt **BY** 0,**-**40,0**)***;*

**jMOVE** pt*;*

**POINT** pt**=SHIFT (**pt **BY** 0,0,**-**30**)***;*

**jMOVE** pt*;*

**POINT** pt**=SHIFT (**pt **BY** 0,75,0**)***;*

**jMOVE** pt*;*

**POINT** pt**=SHIFT (**pt **BY** 0,0,10**)***;*

**jappro** pt,lowAccuracy*;*

*;*

*;jazda szybka niedokladka w poblize KMpickPlace*

**ACCURACY** lowAccuracy **ALWAYS***;*

**SPEED** highSpeed **ALWAYS***;*

PRINT *"jazda w poblize KMpickPlace ";*

**jAPPRO** #nearKMpickPlace, lowAccuracy*;*

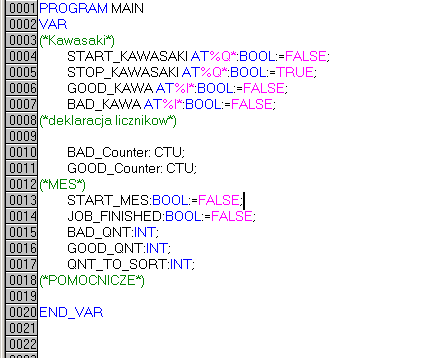
**RETURN***;*

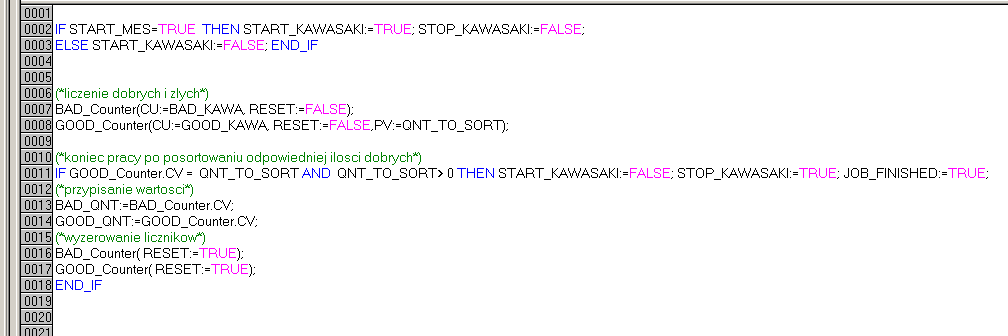
.END

III. Sterownik PLC

Do pośredniczenia wymianą danych pomiędzy robotem Kawasaki a oprogramowanie Wonderware MES wykorzystaliśmy sterownik PLC firmy Beckhoff z zainstalowanym na nim oprogramowaniem TwinCat 2. Komunikacja oraz sterowanie pomiędzy robotem polegała na ustawianiu odpowiednich wartości na moduł wyjść oraz odczytywaniu z modułu wejść sterownika. Wymiana danych z systemem MES została zrealizowana za pomocą serwera OPC.

Kod programu sterującego:





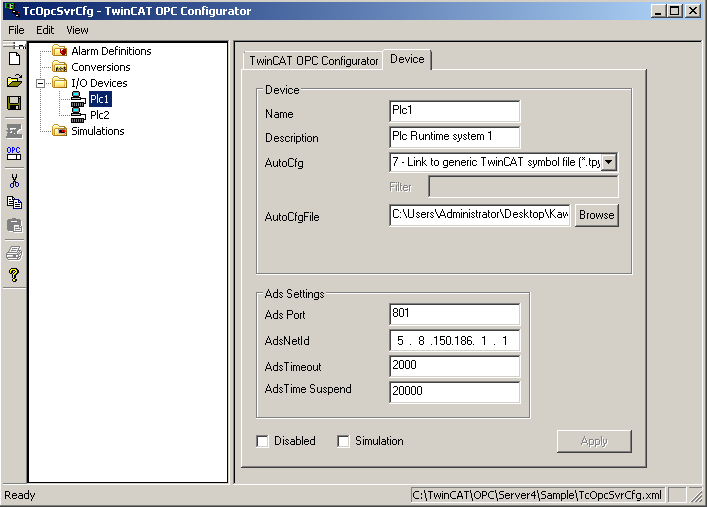
Opis działania programu

Gdy z systemu MES wyślemy sygnał do rozpoczęcia realizacji zlecenia program uruchamia pracę robota Kawasaki. Zaimplementowane zostały liczniki dobrych i złych elementów, które odpowiednio zwiększają się w odpowiedzi na sygnały wystawiane przez robota. Podczas gdy zostanie osiągnięta pożądana ilość dobrych elementów program wysyła sygnał sterujący w celu zatrzymania czynności sortowania. Zainicjowane są tu także zmienne informujące o przebiegu procesu wykorzystywane przez system MES.

3.2 Strona programowa

I. Komunikacja InTouch ze sterownikiem Beckhoff

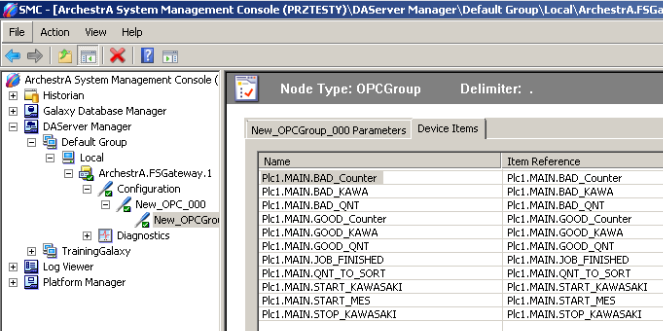
Do komunikacji oporogramowania InTouch ze sterownikiem Beckhoff został wykorzystany protokół OPC. W tym celu musieliśmy zainstalować oraz skonfigurować Serwer OPC firmy Beckhoff. Konfiguracja po stronie TwinCat’a wyglądała następująco:



Rys. 2 konfiguracja serwera OPC

Do konfiguracji serwera musieliśmy wskazać plik o rozszerzeniu .tpy z naszym programem sterującym robotem a także port oraz adres AdsNetId naszego sterownika na którym pracowaliśmy.

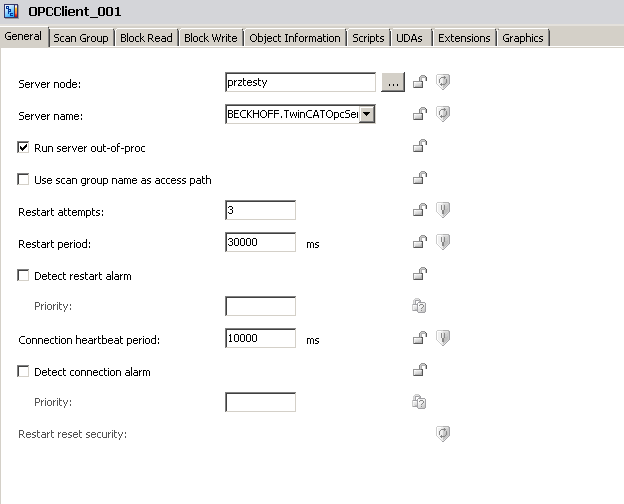
Konfiguracja po stronie Wonderware odbywała się w oprogramowaniu SMC ( System Management Console) gdzie musieliśmy dodać obiekt OPC, zaimportować zmienne a następnie aktywować serwer.



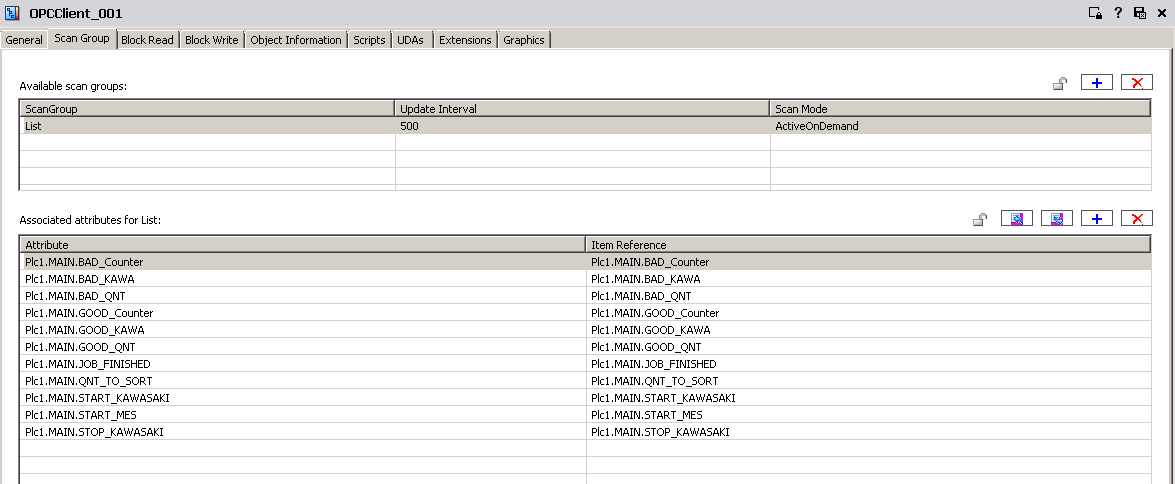
Rys. 3 Zaimportowane zmienne

II. Odczyt danych po stronie Archestra IDE

W celu odczytu danych z serwera OPC w Archestre IDE musieliśmy do galaktyki dodać nową instancję ‘OPC Client’ a następnie odpowiednio ją skonfigurować.



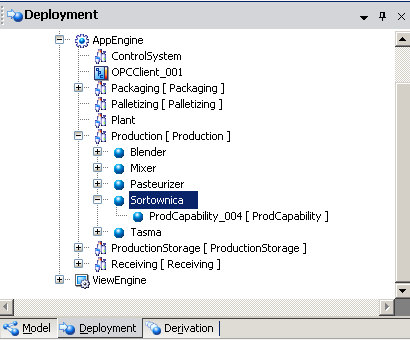
Rys. 4 konfiguracja instacji OPC Client



Rys. 5 Konfiguracja zmiennych do odczytu

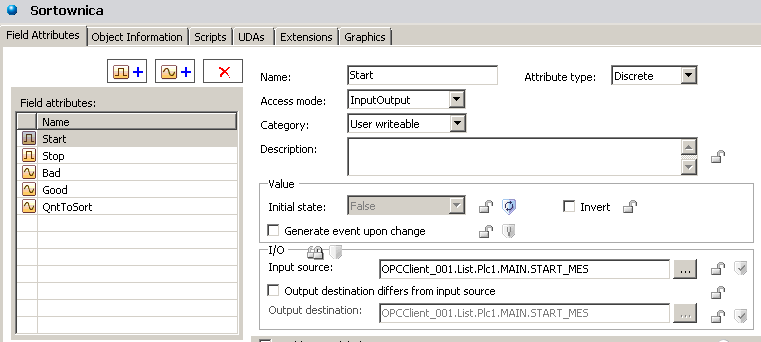
III. Dodanie nowej instancji

Do już istniejących dodaliśmy do galaktyki własną instancję Sortownica która reprezentowała nasz robot Kawasaki.



Rys. 6 Dodanie nowej instancji

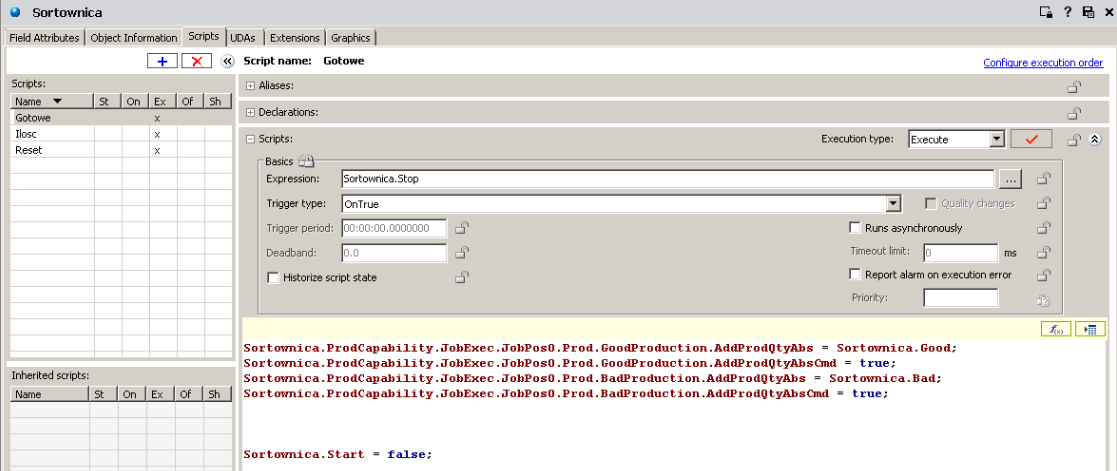
Do dodanej encji dodaliśmy atrybuty potrzebne do sterowania procesem produkcji a następnie powiązaliśmy je ze zmiennymi z klienta OPC.



Rys. 7 Powiązanie atrybutów encji ze zmiennymi z klienta OPC

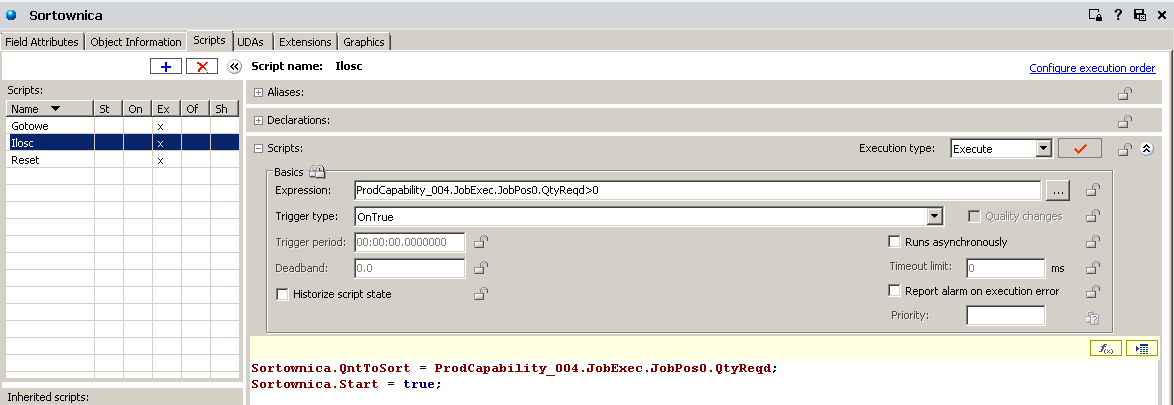
IV. Dodanie skryptów

Aby dodane przez nas atrybuty realizowały określone akcje musieliśmy napisać do nich skrypty przedstawione poniżej.



Rys. 8 skrypt *Gotowe*

Skrypt *Gotowe* po zakończonej pracy robota Kawasaki pobiera wartość posortowanych elementów dobrych i złych.

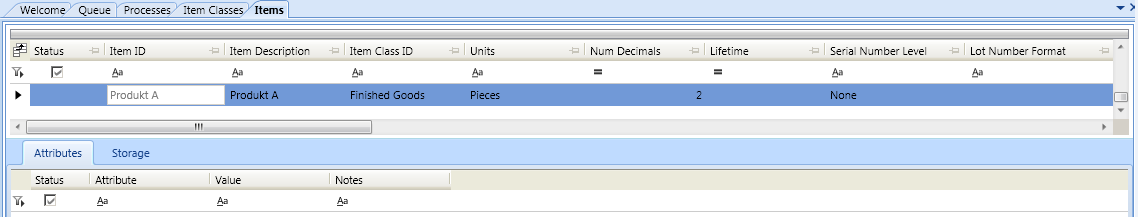


Rys. 8 skrypt *Ilość*

Skrypt *Ilość* pobiera wartość ilości dobrych elementów ze zlecenia które chcemy uzyskać i przypisuje ją do zmiennej w programie TwinCat.

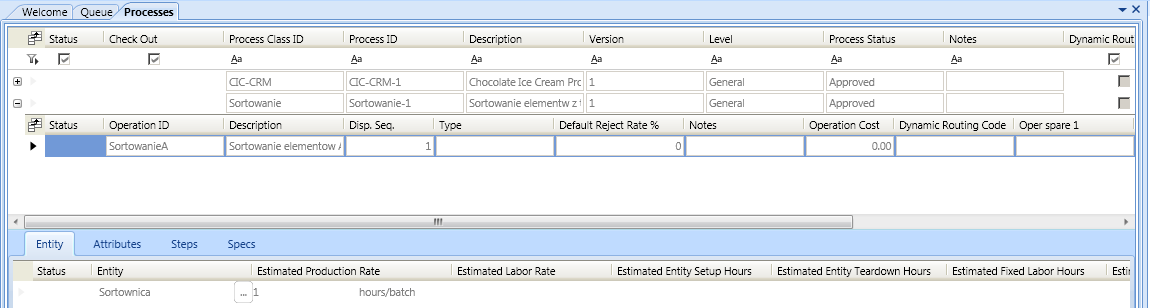
V. Wonderware MES

W oprogramowaniu Wonderware MES zdefiniowaliśmy produkt końcowy A, który jest elementem dobrym przy sortowaniu, jego jednostką miary są sztuki.



Rys. 9 zdefiniowanie produktu

Zdefiniowaliśmy także proces *Sortowanie* i operację *SortowanieA* do której przypisaliśmy encję *Sortownica*.

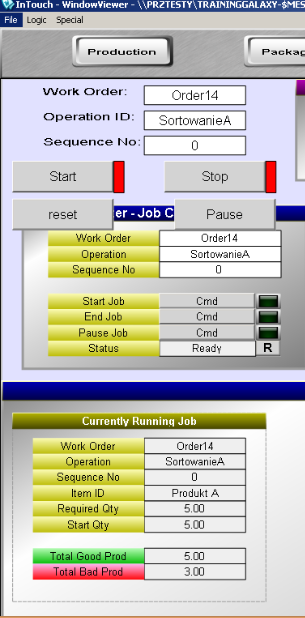


Rys. 10 Zdefiniowanie procesu

Gdy wszystko dobrze zdefiniowaliśmy, mogliśmy złożyć nowe zamówienie klikając prawym przyciskiem myszy na nasz proces.

VI. InTouch

W oprogramowaniu InTouch służącym do wizualizacji procesu, gotową już wizualizację przerobiliśmy na nasze potrzeby dodając 4 przyciski: start, stop, reset oraz pause oraz nowe okna do uruchamiania nowych zleceń.

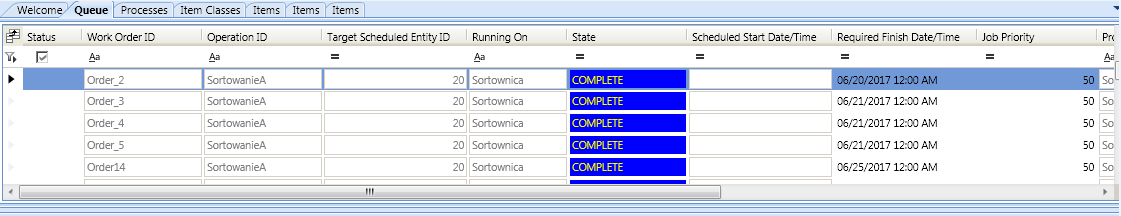


Rys. 11 Zmodyfikowana wizualizacja InTouch

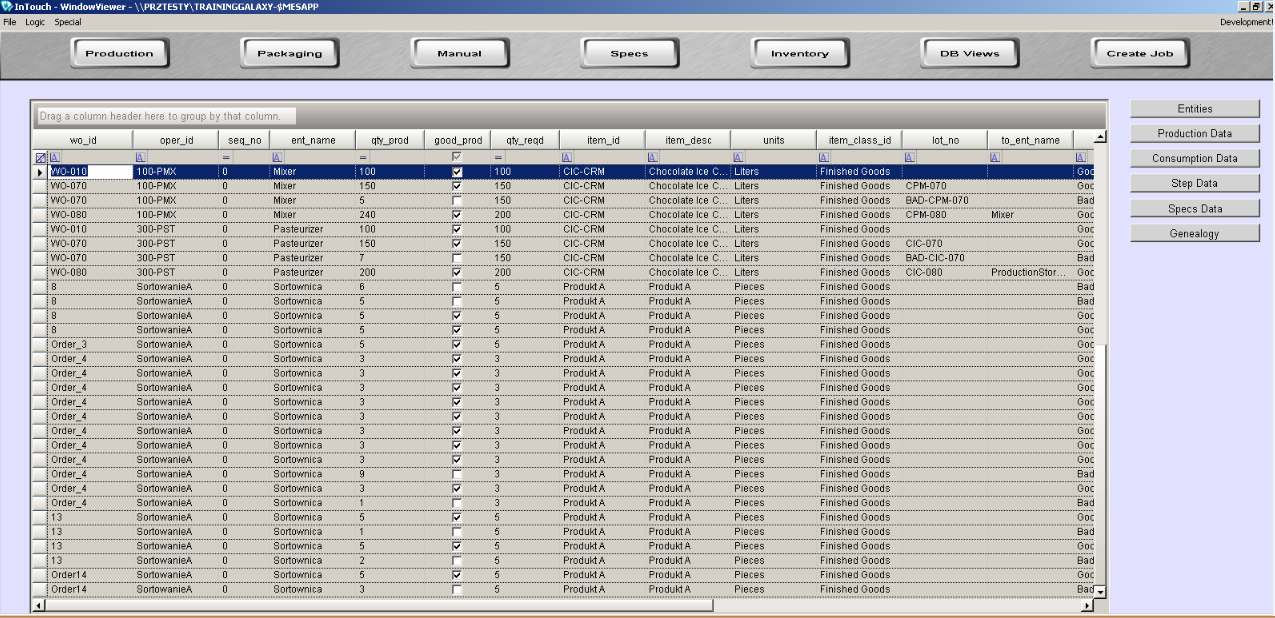
Z poziomu InTouch uruchamialiśmy wybrane przez nas zlecenie do realizacji, naciskając przycisk start uruchamialiśmy robota Kawasaki, gdy ten posortował odpowiednią ilość dobrych elementów kończył swoją pracę wysyłając ilość dobrych elementów oraz ilość złych które pojawiły się podczas sortowania.

VII. Końcowy efekt

Gdy uruchomiony przez nas proces sortowania skończy się można sprawdzić w Wonderware MES że złożone przez nas zamówienia otrzymały starus *Complete,* także w oprogramowaniu InTouch można sprawdzić podsumowania zamówień w zakładce DB Views



Rys. 12 Status zamówień w Wonderware MES



Rys. 12 Podsumowanie zamówień w programie InTouch

1. Podsumowanie działania stanowiska

W skonfigurowanym systemie MES generowaliśmy zlecenie w którym definiowaliśmy ile chcemy uzyskać dobrych elementów. W przygotowanej przez nas prostej symulacji w programie Wonderware Intouch mogliśmy uruchamiać zlecenie ( wysyłanie sygnału w celu rozpoczęcia pracy robota Kawasaki) oraz mogliśmy po skończonej pracy zaobserwować ile podczas sortowania odrzucono wadliwych produktów.

Zadanie polegało na podawaniu różnych rodzajów elementów, i odstawianiu dobrych na jedna paletę a złych na druga.

System MES do wygenerowaniu zlecenia wysyła sygnał uruchamiający robota, który od tej chwili oczekuje na rozpoznanie elementu. Po dojechaniu elementu do czujnika, kamerka robi zdjęcie rozpoznaje kod i wysyła sygnał do robota, który z kolei wysyła sygnał do systemu MES o rodzaju rozpoznanego przedmiotu i rozpoczyna przekładanie elementu na odpowiednia paletę, po czym wraca na miejsce oczekiwania i czeka na kolejny sygnal z kamerki. Po ułożeniu zleconej liczby prawidłowych przedmiotów (typu1 czyli kod 1-3), system MES wyłącza robota który wraca w miejsce oczekiwania.