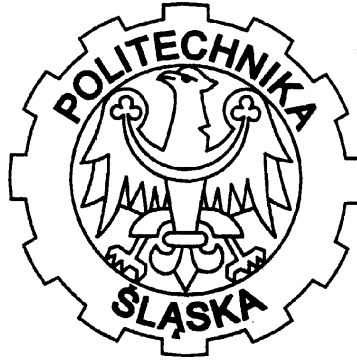


Politechnika Śląska w Gliwicach
Wydział Automatyki, Elektroniki i Informatyki



Projektowanie przemysłowych systemów komputerowych

Projekt – Sprawozdanie

GasAnalyzer

Autorzy: Damian Karbowski, Grzegorz Powąła
Informatyka, SSM3, grupa ISP1
Prowadzący: dr inż. Jacek Stój

2 czerwca 2013

Spis treści

1	Wstęp	2
1.1	Geneza	2
1.2	Temat	2
1.3	Dostępny sprzęt	2
1.3.1	Sterownik PLC	2
1.3.2	Komputer	3
1.4	Analiza tematu	4
1.5	Założenia	4
1.6	Plan pracy	4
2	Instrukcja użytkownika	5
2.0.1	Ekran powitalny	6
2.0.2	Stan robota	6
2.0.3	Stan magazynu i testowanie obsługi	7
2.0.4	Testowanie sterowania ręcznego z poziomu wizualizacji	9
2.0.5	Testowanie sterowania ręcznego z pilota podłączonego do sterownika	10
2.0.6	Testowanie sterowania automatycznego	10
2.1	Specyfikacja wewnętrzna	10
3	Bibliografia	14
4	Spis rysunków, tablic i kodów źródłowych	15
4.1	Spis rysunków	15
4.2	Spis tablic	15
4.3	Spis kodów źródłowych	15
5	Załączniki	16

1 Wstęp

1.1 Geneza

Tematem projektu, którego dotyczy ta praca jest: „Syst”. Pomysł na projekt pojawił się po zrealizowaniu przez autora projektu semestralnego z przedmiotu Sterowniki PLC.

1.2 Temat

Głównymi celami pracy było napisanie oprogramowania

1.3 Dostępny sprzęt

Na potrzeby realizacji projektu wykorzystano istniejące stanowisko laboratoryjne, którego schemat przedstawia Rysunek 1. Składa się ono z:

- Sterownika PLC,
- Komputera,
- Modelu Robota 3D.

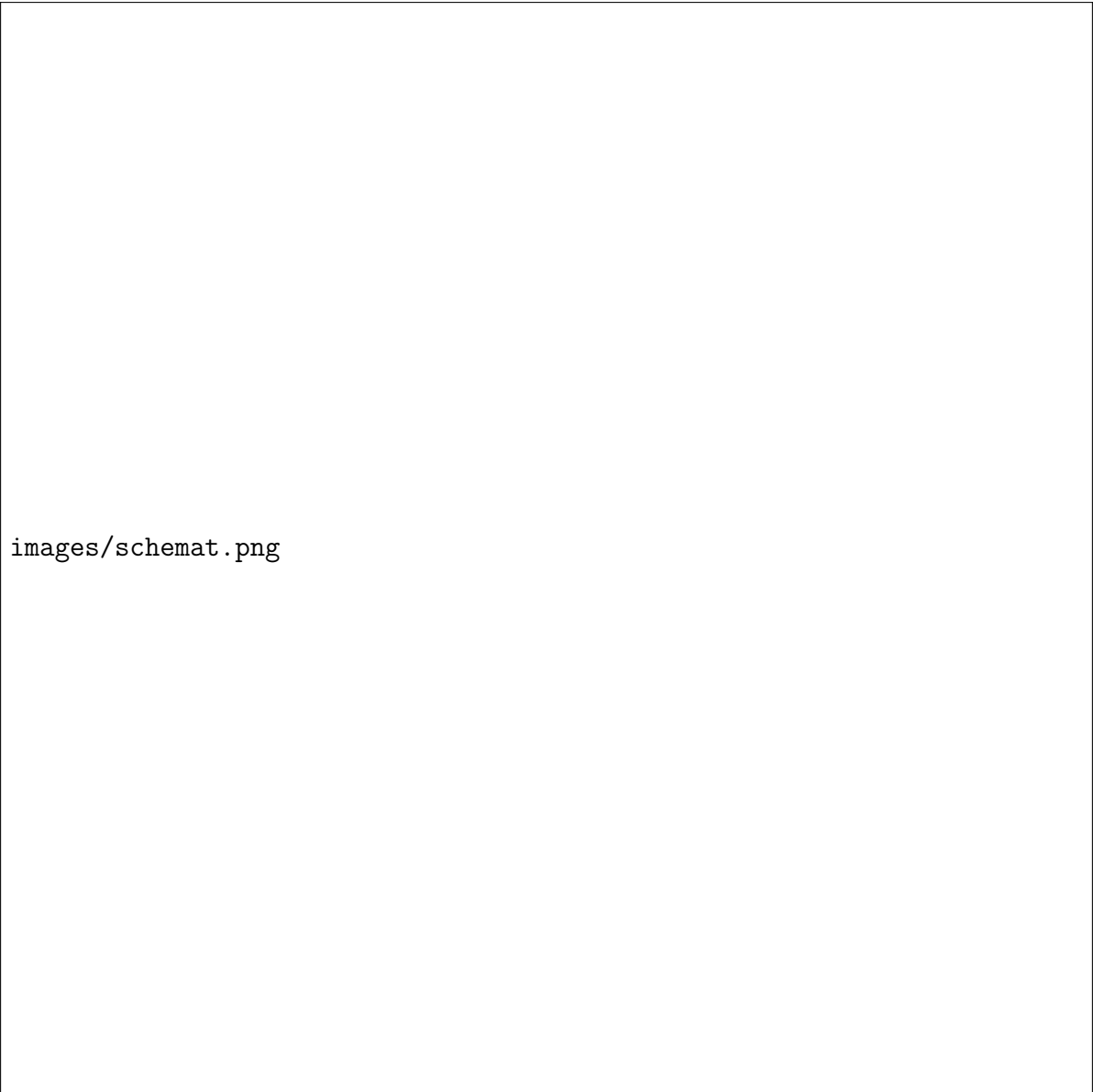
Stanowisko to na potrzeby projektu rozbudowano o model magazynu wysokiego składowania. Poszczególne składowe stanowiska zostały bardziej szczegółowo opisane w kolejnych podrozdziałach.

1.3.1 Sterownik PLC

Sterownik PLC wykorzystywany do realizacji projektu był wyposażony w następujące moduły:

1. SIMATIC S7-300, Jednostka centralna S7-300 CPU 315F-2 PN/DP,
2. SIMATIC S7-300, Zasilacz PS 307,
3. SIMATIC S7-300, Wejścia/Wyjścia cyfrowe SM 323,
4. SIMATIC S7-300, Wejścia/Wyjścia analogowe SM 334.

Sterownik podłączony jest do sieci lokalnej Ethernet w laboratorium, więc komunikacja z nim odbywa się tak samo jak z każdym innym urządzeniem sieciowym. Podstawy programowania i korzystania ze sterowników autor poznał zapoznając się z odpowiednią literaturą [1, ?, ?, ?, ?]. Konfigurację sterownika wraz z modułami przedstawia Rysunek ??.



images/schemat.png

Rysunek 1: Schemat stanowiska

1.3.2 Komputer

Projekt w całości był realizowany na laptopie autora, podłączanym do sieci w laboratorium. Na komputerze uruchomiane były dwie maszyny wirtualne. Na jednej zainstalowane było środowisko Step 7 do programowania sterownika, a na drugiej WinCC flexible 2008 do tworzenia i uruchamiania wizualizacji. Wizualizacje tworzone w środowisku WinCC flexible są dedykowane do paneli operatorskich, jednak ta stworzona przez autora na potrzeby projektu była uruchamiana na komputerze za pomocą runtime system.

1.4 Analiza tematu

Analiza tematu polegała przede wszystkim na zapoznaniu się z narzędziami programistycznymi do tworzenia oprogramowania sterownika oraz wizualizacji. W wyniku analizy autor poznał podstawy języków: LAD [?, ?, ?], STL [?, ?, ?], FBD [?, ?, ?], GRAPH [?], SCL [?, ?, ?] i AWL do tworzenia programu sterownika oraz VBScript do tworzenia skryptów w wizualizacji. Poznanie tych podstaw pozwoliło dobrać język odpowiedni do realizacji poszczególnych zadań.

1.5 Założenia

Oprogramowanie dla Robota Fishertechnik powinno zostać stworzone przy użyciu środowiska Step 7 oraz działać na sterownikach firmy Siemens. Funkcjonalności robota wchodzące w skład projektu, to:

- sterowanie ręczne z pilota podłączonego bezpośrednio do sterownika,
- sterowanie ręczne z wizualizacji,
- sterowanie automatyczne,
- wizualizacja stanu magazynu,
- umożliwienie korzystania z magazynu zarówno poprzez sterowanie ręczne, jak i przy użyciu zautomatyzowanych poleceń dostępnych z poziomu wizualizacji.

Powyżej zostały wymienione założenia podstawowe, jednak autor nie wyklucza zrealizowania dodatkowych zadań, które nie zostały zamieszczone w pierwotnej koncepcji realizacji projektu.

1.6 Plan pracy

Realizacja projektu została podzielona na następujące etapy:

- Przygotowanie stanowiska, zebranie odpowiednich materiałów i literatury,
- Analiza wymagań funkcjonalnych aplikacji,
- Projektowanie struktury oprogramowania i interfejsów wymiany danych,
- Implementacja,
- Testowanie i uruchamianie,
- Przedstawienie projektu i ewentualne korekty.

Powyższy plan pracy stanowił dla autora wyznacznik kolejnych działań. Jednak powszechnie wiadomo, że w praktyce poszczególne punkty są wymienne i wpływają na siebie wzajemnie.

2 Instrukcja użytkownika

Zaimplementowana przez autora wizualizacja ma na celu zobrazowanie działania modelu oraz umożliwienie operatorowi wpływania na jego działanie. Kolejne podrozdziały zawierają opis specyfikacji zewnętrznej oraz wewnętrznej. Część odnosząca się do specyfikacji zewnętrznej jest skróconą instrukcją obsługi użytkownika. Specyfikacja wewnętrzna jest opisem, jak zostały zrealizowane poszczególne elementy i w jaki sposób wizualizacja współpracuje ze sterownikiem.

Specyfikacja zewnętrzna przedstawiona w dalszej części podrozdziału stanowi skróconą instrukcję obsługi wizualizacji oraz opis możliwości oferowanych przez poszczególne ekrany.

Autor projektu wykorzystał w swojej pracy szereg elementów dostępnych standardowo w środowisku Simatic WinCC flexible. Podstawowymi elementami sterującymi są przyciski w trybie tekstowym oraz przeźroczystym. Głównymi obiektami służącymi do prezentacji informacji są: pola tekstowe, pola wejściowo-wyjściowe oraz pola daty i godziny. Dodatkowo celem uatrakcyjnienia wizualizacji wykorzystane zostały suwaki (ang. *slider*), obrazki oraz zegarek.

Obsługa tej części projektu jest realizowana za pomocą myszy i klawiatury podłączonych do komputera. Za pomocą klawiatury wybieramy interesujący nas ekran lub wprowadzamy żadaną wartość pozycji docelowej na ekranie testowania trybu automatycznego.

Wszystkie ekrany wizualizacji są tworzone na podstawie szablonu. Wszystkie przyciski widoczne w górnej części Rysunku 2 są powiązane z wybranymi klawiszami na klawiaturze z zakresu F1 do F12. Użycie pierwszych 6 przycisków lub związanych z nimi klawiszy F1 - F6 powoduje zmianę aktualnie wyświetlanego ekranu. Przycisk „Koniec” na wizualizacji oraz powiązany z nim klawisz F7 po aktywowaniu powodują zamknięcie wizualizacji poprzez wywołanie funkcji standardowej StopRuntime.

Dodatkowo do przycisków F9 - F12 została przypisana zmiana stanu zmiennej *EmergencyStop*, która pozwala na awaryjne zatrzymanie pracy robota w dowolnym momencie. Modyfikowanie wartości zmiennej odpowiadającej za awaryjne zatrzymanie pracy może się również odbywać poprzez kliknięcie na kontrolkę znajdującą się w prawym dolnym rogu każdego ekranu. Ekrany dostępne w wizualizacji oraz klawisze funkcyjne z nimi związane:

- Ekran powitalny - F1,
- Stan robota - F2,
- Stan magazynu i testowanie obsługi - F3,
- Testowanie sterowania ręcznego z pilota podłączonego do sterownika - F4,
- Testowanie sterowania ręcznego z poziomu wizualizacji - F5,
- Testowanie sterowania automatycznego - F6.

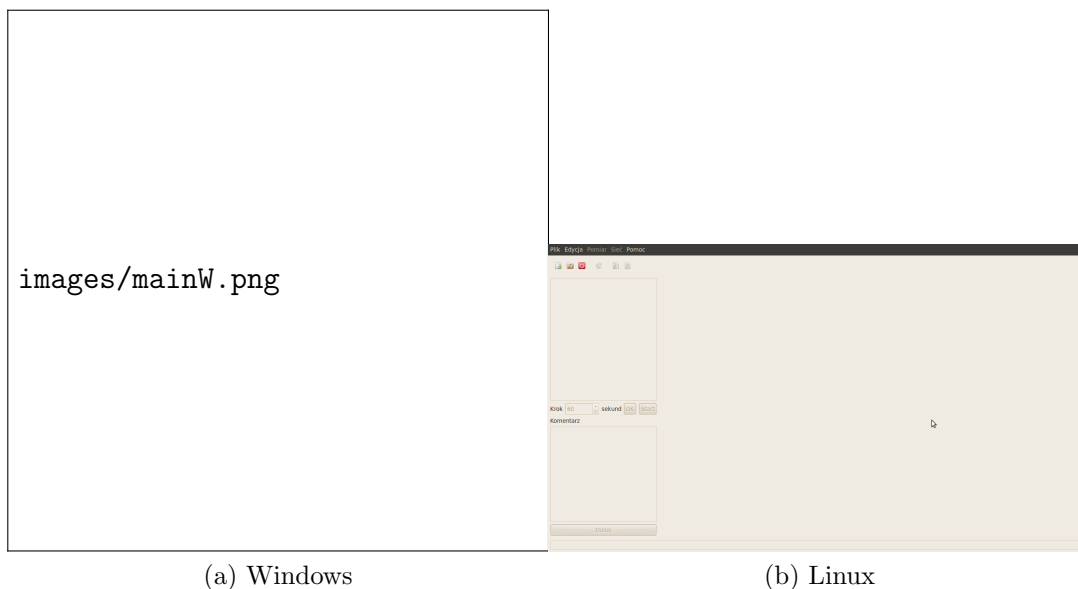
Poszczególne ekrany zostaną szczegółowo opisane w kolejnych podrozdziałach.



Rysunek 2: Okno ładowania

2.0.1 Ekran powitalny

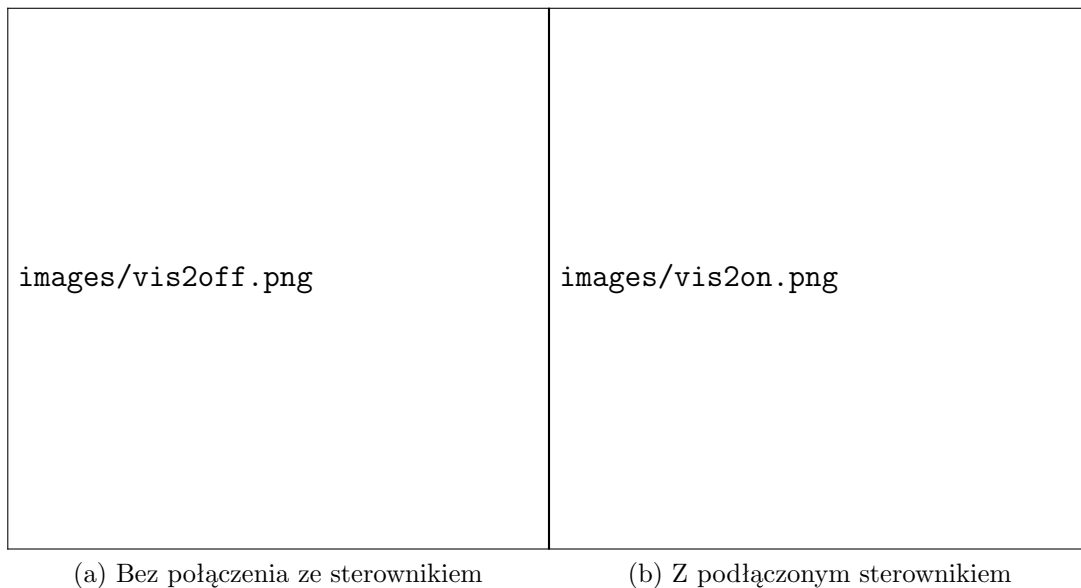
Bezpośrednio po uruchomieniu wizualizacji użytkownik zobaczy ekran powitalny taki jak na Rysunku 3 zawierający informacje o autorze projektu, osobie kierującej projektem (promotorze) oraz informację o przeznaczeniu wizualizacji wraz ze zdjęciem modelu. Dodatkowo na ekranie tym umieszczony został zegar analogowy i cyfrowy oraz aktualna data.



Rysunek 3: Okno główne

2.0.2 Stan robota

Ekran przedstawiający aktualny stan robota widoczny jest na Rysunku 4. Prezentuje on informacje o stanie poszczególnych silników oraz o trybie pracy modelu. Na ekranie zobaczyć można informacje o tym, czy silnik nie osiągnął swojej pozycji minimalnej lub maksymalnej, czy nie występuje sygnał z krańcówki lub z czujnika impulsów oraz aktualne położenie każdego silnika w formie liczbowej i jako wypadkowa na suwaku. Ponadto na ekranie tym znajdują się informacje o ewentualnych błędach. Jeśli dany błąd wystąpi w sterowniku to na ekranie pojawi się pole tekstowe z jego treścią. Klikając na pole z treścią błędu wymuszona jest zmiana flagi odpowiedzialnej za wystąpienie, co skutkuje zniknięciem błędu z ekranu. W sytuacji gdy błąd nie znika oznacza to, że występuje on nadal w sterowniku.



Rysunek 4: Ekran prezentujący stan robota

2.0.3 Stan magazynu i testowanie obsługi

Najważniejszym ekranem w całej wizualizacji jest ekran zawierający stan magazynu, pozostałe pełnią bowiem role informacyjne lub testowe. Ekran ten składa się z kilku bardzo ważnych części, które można zaobserwować na Rysunku 5.

W lewej górnej części widać kolejkę zadań do wykonania wraz z jej aktualną długością. Długość podana jest w nawiasie przy nagłówku „Kolejka”. Poniżej tego nagłówka widocznych jest 11 początkowych elementów kolejki. Każdy element kolejki składa się z indeksu pozycji w magazynie oraz wartości bool decydującej, czy element będzie zabrany czy położony. Pod prezentowaną kolejką widnieje przycisk „Start”, który powoduje rozpoczęcie wykonywania zadań dodanych wcześniej przez użytkownika do kolejki.

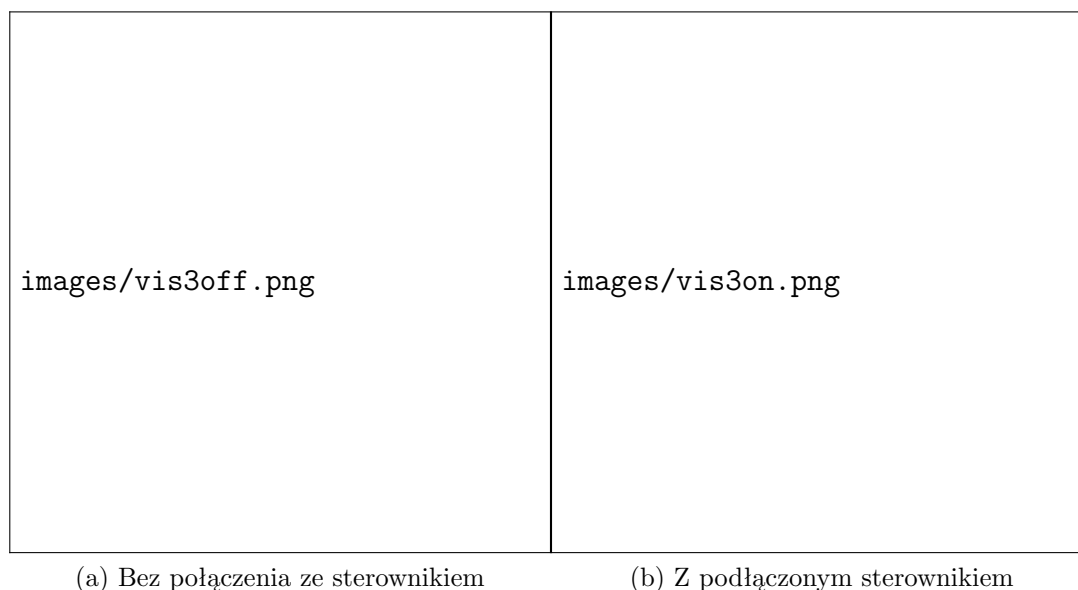
Na środku znajdują się 24 prostokąty obrazujące szufladki w magazynie. Kolor pola informuje o tym, czy jest ono zajęte czy wolne. Kolor zielony oznacza, że dana szufladka jest pusta, natomiast prostokąt w kolorze czerwonym oznacza, że dana szufladka jest

już zajęta. Na każdym polu znajdują się przyciski „Weź” oraz „Połóż”. Podobne do pól magazynu są pola wejściowe oraz wyjściowe. Przy polach tych dodatkowo znajdują się przyciski, które umożliwiają zasymulowanie położenia produktu na platformie wejściowej oraz odbiór produktu z platformy wyjściowej. Po prawej stronie u góry widać grupę przycisków służących do wybierania indeksu elementu w specyficzny sposób. Elementy wyżej służą do wybierania elementów najbliższych lub najdalej położonych w tablicy. Przyciski poniżej służą do wybierania elementu najmłodszego lub najstarszego włożonego wcześniej do magazynu.

Wszystkie pola magazynu oraz platformy wejściowa i wyjściowa posiadają dodatkową informację o dacie i godzinie ostatniego użycia. Po umieszczeniu w danej komórce lub zabraniu z niej produktu, sterownik zapisuje w pamięci informację o dacie i godzinie tego zdarzenia.

Dolna część ekranu pełni tylko funkcję informacyjną. Po lewej stronie znajdują się godzina oraz data odczytane z komputera lub panelu operatorskiego i sterownika przemysłowego. Z prawej natomiast znajdują się informacje o aktualnym położeniu poszczególnych silników w formie wartości liczbowej.

Ostatnim elementem znajdującym się na tym ekranie jest informacja dla użytkownika o ewentualnych błędach. Zasada działania tych informacji jest taka sama jak na ekranie z informacjami o stanie robota, co zostało opisane przez autora wcześniej.

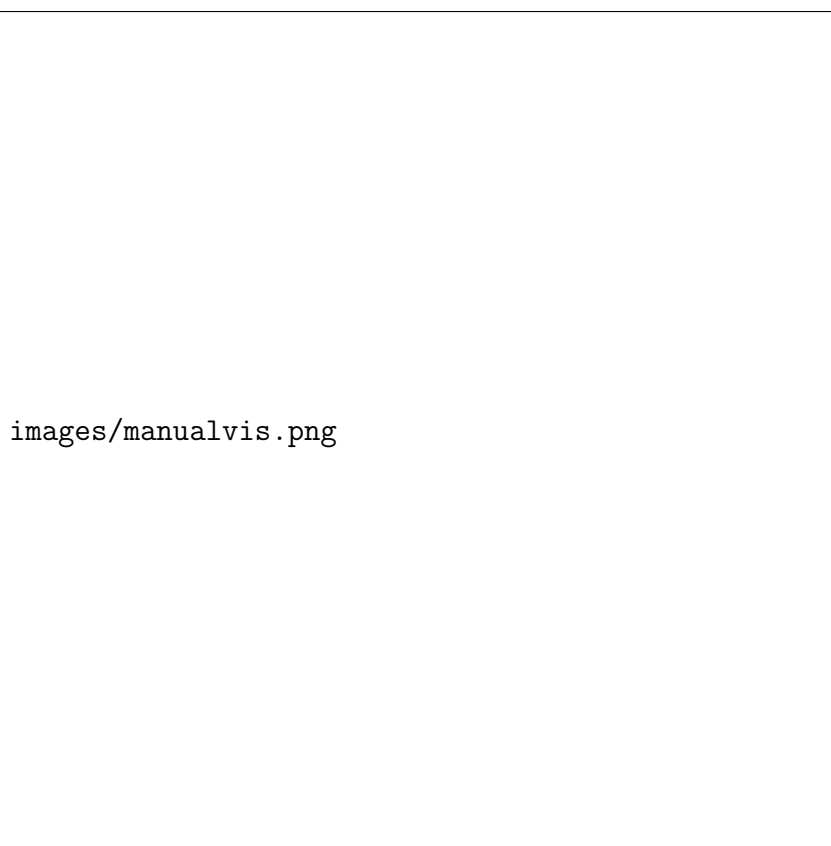


Rysunek 5: Ekran prezentujący stan magazynu i umożliwiający obsługę

2.0.4 Testowanie sterowania ręcznego z poziomą wizualizacją

Testowanie trybu manualnego z poziomą wizualizacją polega na wybieraniu kierunku pracy oraz załączaniu poszczególnych silników wykorzystując do tego celu 8 przycisków znajdujących się na ekranie (4 grupy po 2 przyciski na każdy silnik). Na ekranie oprócz wymienionych już przycisków znajdują się również informacje o tym, jaki jest stan poszczególnych silników, jakie jest ich bieżące położenie, czy któryś z nich nie osiągnął

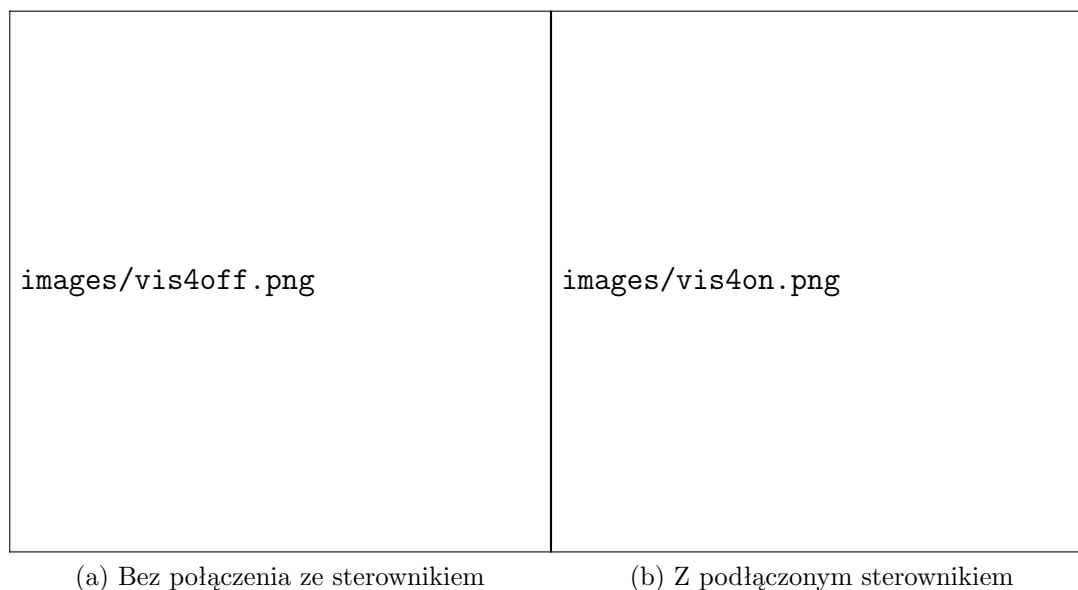
swojego minimum lub maksimum oraz sygnały z zamontowanych na robocie czujników. Ekran ten został zaprezentowany na Rysunku 7. Dodatkowo do ekranu tego zostały przypisane klawisze ułatwiające obsługę. Na Rysunku 2 pokazane zostały przyciski związane ze wszystkimi ekranami, natomiast przyciski przypisane tylko do wybranego ekranu mają żółty kolor trójkąta informacyjnego, co zauważyć można na Rysunku 6. Dla odróżnienia klawiszy globalnych od tych dla konkretnego ekranu autor postanowił wykorzystać kombinacje klawiszy Shift+F1 do Shift+F8. Dla sterowania silnikiem Lift wykorzystano kombinację Shift+F1 (start) i Shift+F2 (kierunek), dla silnika Arm Shift+F3 (start) i Shift+F4 (kierunek), dla silnika Rotate Shift+F5 (start) i Shift+F6 (kierunek) oraz dla silnika Grab Shift+F7 (start) i Shift+F8 (kierunek).



Rysunek 6: Widok dodatkowych klawiszy przypisanych do ekranu testowania sterowania ręcznego z poziomu wizualizacji

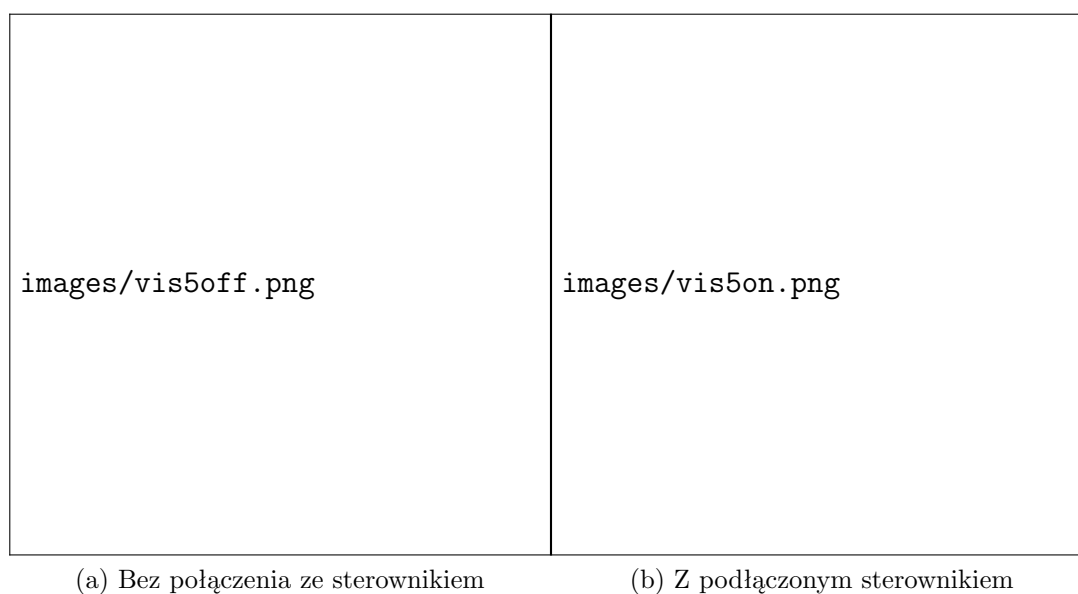
2.0.5 Testowanie sterowania ręcznego z pilota podłączonego do sterownika

Podobnie jak w przypadku testowania trybu manualnego z poziomu wizualizacji w trybie manualnym z pilota sterujemy pracą robota za pomocą przycisków. Tym razem przyciski znajdujące się na ekranie pełnią tylko rolę informacyjną na temat stanu tych znajdujących się na pilocie podłączonym bezpośrednio do sterownika. Sterowanie faktyczne odbywa się za pomocą przycisków znajdujących się na pilocie. Oprócz tego na ekranie



Rysunek 7: Ekran umożliwiający testowanie sterowania ręcznego z poziomu wizualizacji

widocznym na Rysunku 8 znajdują się takie same informacje jak w przypadku ekranu ze sterowania z poziomu wizualizacji.

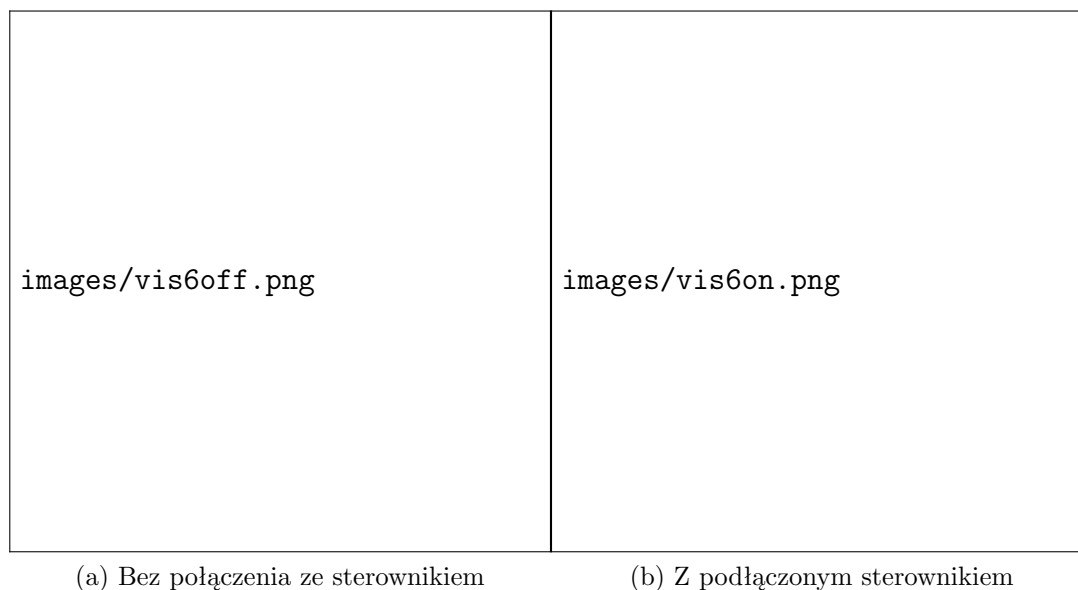


Rysunek 8: Ekran umożliwiający testowanie sterowania ręcznego z pilota podłączonego do sterownika

2.0.6 Testowanie sterowania automatycznego

Testowanie trybu automatycznego polega na wprowadzaniu przez użytkownika wartości docelowej dla poszczególnych silników. Oprócz pól do wprowadzania tych wartości

widocznych na Rysunku 9 użytkownik widzi takie same informacje jak dla testowania trybów sterowania ręcznego.



Rysunek 9: Ekran umożliwiający testowanie sterowania automatycznego

2.1 Specyfikacja wewnętrzna

Wizualizacja komunikuje się z komputera klasy PC ze sterownikiem za pośrednictwem protokołu Ethernet w sieci lokalnej. Odniesienia do odpowiednich adresów w pamięci sterownika dokonywane są za pomocą nazw symbolicznych zdefiniowanych w tablicy Tags. Do działania wizualizacja używa tylko jednej zmiennej wewnętrznej i jest to zmienna tablicowa *MagazynDTEnable* z elementami typu bool. Elementy te odpowiadają za wyświetlanie dat oraz godzin na ekranie ze stanem magazynu po kliknięciu na wybraną komórkę. Obsługa wyświetlania dat polega na tym, że po kliknięciu w wybrane pole ustawiana jest odpowiednia zmienna w tej tablicy na wartość *true*, a po zwolnieniu klawisza myszki na wartość *false*. Za zmiany te odpowiadają niewidzialne przyciski umieszczone na tych polach.

Wizualizacja wpływa na pracę sterownika poprzez zmianę pojedynczych bitów za pomocą umieszczonych na ekranie przycisków. Wpływa ona również poprzez modyfikowanie wybranych zmiennych odpowiadających pozycjom docelowym lub poprzez dodawanie odpowiednich zadań do kolejki. Bardziej zaawansowane operacje zostały zrealizowane za pomocą skryptów napisanych w języku VBScript, które są bardzo prostą i szybką opcją wykonywania bardziej zaawansowanych czynności.

Skrypt BDScrip wywołuje, poprzez ustawienie odpowiednich bitów w sterowniku, funkcję wybierającą najbliższą lub najdalszą pustą lub zajętą komórkę w magazynie.

```
1 If i=1 Then
2   SmartTags("BD") = True
3   SmartTags("BDen") = True
4   SmartTags("AddBoolWarehouse") = True
5 ElseIf i=2 Then
6   SmartTags("BD") = True
7   SmartTags("BDen") = True
8   SmartTags("AddBoolWarehouse") = False
9 ElseIf i=3 Then
10  SmartTags("BD") = False
11  SmartTags("BDen") = True
12  SmartTags("AddBoolWarehouse") = True
13 Else
14  SmartTags("BD") = False
15  SmartTags("BDen") = True
16  SmartTags("AddBoolWarehouse") = False
17 End If
```

Kod źródłowy 1: BDScrip

Skrypt NNScrip wywołuje w sterowniku wybranie najstarszej lub najmłodszej zajętej komórki z magazynu i dodanie jej do kolejki zadań.

```
1 If i=1 Then
2   SmartTags("NN") = True
3   SmartTags("NNE") = True
4 Else
5   SmartTags("NN") = False
6   SmartTags("NNE") = True
7 End If
```

Kod źródłowy 2: NNScrip

Po wybraniu opcji „Połóż” lub „Weź” są wywoływane odpowiednio skrypty MagazynPush lub MagazynTake. Skrypty te powodują dodanie odpowiedniego indeksu oraz ruchu chwytaka do kolejki zadań.

```
1 SmartTags("AddedIndex") = i
2 SmartTags("AddBoolWarehouse") = False
3 SmartTags("AddQueue") = True
```

Kod źródłowy 3: MagazynPush

```
1 SmartTags("AddedIndex") = i
2 SmartTags("AddBoolWarehouse") = True
3 SmartTags("AddQueue") = True
```

Kod źródłowy 4: MagazynTake

3 Bibliografia

Literatura, która została wykorzystana przez autorów w czasie powstawania projektu, którą opisuje niniejsza dokumentacja.

- [1] Jerzy Kasprzyk: *"Programowanie sterowników przemysłowych"*, Wydawnictwa Naukowo-Techniczne WNT, Warszawa, 2007
- [2] Dokumentacja producenta: *„ELAN Interface Description”*, sierpień 2006
- [3] Materiały szkoleniowe: „SIMATIC S7 - Kurs podstawowy”

4 Spis rysunków, tablic i kodów źródłowych

4.1 Spis rysunków

Rysunek 1:	Schemat stanowiska	3
Rysunek 2:	Okno ładowania	6
Rysunek 3:	Okno główne	7
Rysunek 4:	Ekran prezentujący stan robota	7
Rysunek 5:	Ekran prezentujący stan magazynu i umożliwiający obsługę . . .	8
Rysunek 6:	Widok dodatkowych klawiszy przypisanych do ekranu testowania sterowania ręcznego z poziomu wizualizacji	9
Rysunek 7:	Ekran umożliwiający testowanie sterowania ręcznego z poziomu wizualizacji	10
Rysunek 8:	Ekran umożliwiający testowanie sterowania ręcznego z pilota podłączonego do sterownika	11
Rysunek 9:	Ekran umożliwiający testowanie sterowania automatycznego . .	11

4.2 Spis tablic

4.3 Spis kodów źródłowych

Kod źródłowy 1:	BDScript	13
Kod źródłowy 2:	NNScript	13
Kod źródłowy 3:	MagazynPush	13
Kod źródłowy 4:	MagazynTake	13

5 Załączniki

- Oświadczenie o autorstwie,
- Płyta CD, na której znajdują się:
 - Kod oprogramowania wewnętrznego oraz pliki projektu Step7,
 - Kod wizualizacji oraz pliki projektu WinCC flexible,
 - Plik wykonywalny wizualizacji typu WinCC flexible RT document,
 - Projekt magazynu wykonany w programie Blender,
 - LaTeXowe pliki pracy inżynierskiej,
 - Zdjęcia magazynu oraz robota,
 - Filmy prezentujące działanie projektu.