



POLITECHNIKA ŚLĄSKA
WYDZIAŁ AUTOMATYKI, ELEKTRONIKI I INFORMATYKI
KIERUNEK INFORMATYKA

Praca dyplomowa magisterska

Projekt i realizacja stanowiska laboratoryjnego do badania zależności
czasowych w sieci EtherCAT

Autor: Damian Karbowski

Kierujący pracą: dr inż. Jacek Stój

Gliwice, czerwiec 2013

Spis treści

1	Wstęp	2
1.1	Stanowisko laboratoryjne	2
1.1.1	Sterownik PLC	2
1.1.2	Komputer	3
1.2	•	3
1.3	Analiza tematu	3
1.4	Założenia	3
1.5	Plan pracy	4
2	Oprogramowanie sterownika	5
2.1	Specyfikacja zewnętrzna	5
2.2	Specyfikacja wewnętrzna	5
3	Wizualizacja HMI	6
3.1	Specyfikacja zewnętrzna	6
3.2	Specyfikacja wewnętrzna	6
4	Badania	7
4.1	Badanie 1	7
4.2	Badanie 2	7
5	Uruchamianie i testowanie	8
5.1	Przebieg testowania	8
5.2	Napotkane problemy	9
6	Wnioski	10
7	Bibliografia	11
8	Spis rysunków, tablic i kodów źródłowych	12
8.1	Spis rysunków	12
8.2	Spis tablic	12
8.3	Spis kodów źródłowych	12
9	Załączniki	13

1 Wstęp

Tematem projektu, którego dotyczy ta praca jest: „Projekt i realizacja stanowiska laboratoryjnego do badania zależności czasowych w sieci EtherCAT”. Zagadnienia związane z tworzeniem oprogramowania dla sterowników przemysłowych są dla autora niezwykle interesujące, a zrealizowany projekt miał na celu dalsze pogłębienie jego wiedzy z tego zakresu. Wyboru tego konkretnego tematu autor dokonał, ponieważ protokół EtherCAT jest jeszcze nowością i według wielu źródeł stanowi przyszłość branży informatyki przemysłowej [10, 11], a praca nad tym tematem wydaje się być pomocna i wartościowa w przyszłej pracy zawodowej lub na ewentualnym dalszym etapie kształcenia.

Głównymi celami pracy było napisanie oprogramowania dla sterownika przemysłowego Siemens S7-300 oraz stworzenie graficznej prezentacji pracy modelu Robota 3D takiego jak na Rysunku ???. Dodatkowym elementem pracy inżynierskiej zrealizowanej przez autora jest element pozainformatyczny, a mianowicie zbudowanie magazynu, z którym robot będzie współpracował. Wyżej wspomniany magazyn po zakończeniu projektu posłuży jako pomoc dydaktyczna do ćwiczeń laboratoryjnych.

1.1 Stanowisko laboratoryjne

Na potrzeby realizacji projektu wykorzystano dwa różne istniejące stanowiska laboratoryjne, które składały się odpowiednio z:

Stanowisko typu CX	Stanowisko typu CP
<ol style="list-style-type: none">1. 2 silniki AM3021-0C00-0000,2. Wyspa EK1100 z zestawem modułów IO,3. Napęd serwo mechanizmów AX5203 (2 osiowy napęd),4. Komputer przemysłowy C6925,5. Zasilacz.	<ol style="list-style-type: none">1. 2 silniki AM3021-0C00-0000,2. Wyspa EK1100 z zestawem modułów IO,3. Napęd serwo mechanizmów AX5203 (2 osiowy napęd),4. Modułowy komputer przemysłowy CX1020,5. Zasilacz.

Tablica 1: Dostępne stanowiska laboratoryjne

1.1.1 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, 2, 3, 4, 5]. Konfigurację sterownika wraz z modułami przedstawia Rysunek ??.

1.1.2 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.

!!!!!!!

1.2 ●

1.3 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.

1.4 Założenia

Oprogramowanie dla dostępnego stanowiska laboratoryjnego powinno zostać stworzone przy użyciu środowiska TwinCAT. Funkcjonalności robota wchodzące w skład projektu, to:

- sterowanie ręczne z pilota podłączonego bezpośrednio do sterownika,
- sterowanie ręczne z wizualizacji,
- sterowanie automatyczne,
- wizualizacja stanu stanowiska.

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

1.5 Plan pracy

Realizacja projektu została podzielona na następujące etapy:

- Zapoznanie się ze sterownikami Beckhoff oraz oprogramowaniem TwinCAT,
- Zapoznanie się z dostępnymi serwonapędami Beckhoff,
- Projekt i realizacja stanowiska,
- Przygotowanie stanowiska do współpracy z systemem wizualizacji,
- 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 Oprogramowanie sterownika

W niniejszym rozdziale opisane zostało oprogramowanie sterujące modelem. W kolejnych podrozdziałach zostanie przedstawiona specyfikacja zewnętrzna oraz wewnętrzna.

2.1 Specyfikacja zewnętrzna

Specyfikacja zewnętrzna przedstawiona w dalszej części podrozdziału zawiera opis, jak korzystać z oprogramowania wgranego do sterownika przez jego autora. Opisane zostało, jak ustawiać odpowiednie zmienne, aby uzyskać żądany efekt.

Lista zmiennych wejściowych i wyjściowych wymieniana między sterownikiem a modelem została już opisana w pierwszym rozdziale, w Tablicach 1 oraz ???. Pozostałe zmienne znajdują się w wewnętrznej pamięci sterownika.

Oprogramowanie może sterować modelem w sposób automatyczny lub ręczny. Tryb automatyczny w trybie obsługi magazynu zostanie opisany w podrozdziale 2.1.2. Tryb ręczny może być realizowany przy pomocy zadajnika podpiętego do sterownika lub przy pomocy przycisków umieszczonych na odpowiednim ekranie wizualizacji. W trybie tym o pracy robota decydujący jest stan przycisków. Dopuszczalne są wszystkie możliwe ruchy w przedziale od wyłącznika krańcowego do wartości maksymalnej.

2.2 Specyfikacja wewnętrzna

Podrozdział specyfikacja wewnętrzna opisuje sposób rozwiązania przez autora kwestii sterowania modelem przy użyciu dostępnego na stanowisku sterownika oraz poszczególnych trybów sterowania. W tworzeniu oprogramowania zostały wykorzystane następujące języki programowania:

- język drabinkowy (Ladder), wykorzystany do stworzenia głównych elementów programu,
- S7-SCL, który został zastosowany do korzystania z tablic. Niestety do korzystania z nich nie można zastosować języka LAD, ponieważ nie da się w nim odwoływać do elementów tablicy przez indeksy będące zmiennymi, a jedynie przez stałe. Po zapoznaniu się z dokumentacją okazało się, że taka możliwość istnieje w języku STL, ale jest to metoda skomplikowana w implementacji. Właśnie dlatego najlepszym i najprostszym rozwiązaniem okazują się S7-SCL, który jest kompilowany do kodu w języku STL.

3 Wizualizacja HMI

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.

3.1 Specyfikacja zewnętrzna

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.

3.2 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.

4 Badania

W niniejszym rozdziale opisany został przebieg przeprowadzonych badań oraz analiza ich wyników. W kolejnych podrozdziałach zostaną przedstawione kolejne różne eksperymenty.

4.1 Badanie 1

4.2 Badanie 2

5 Uruchamianie i testowanie

Rozdział ten zawiera generalne podsumowanie przebiegu prac nad projektem. Opisane zostaną tu wszelkie poważne problemy, które wystąpiły w czasie realizacji projektu. Ponadto zawarto tu opis przebiegu procesu testowania.

5.1 Przebieg testowania

W procesie weryfikacji poprawności działania projektu zastosowano testowanie wstępujące.

Głównym testerem był autor projektu więc większość testów przebiegała na zasadzie białej skrzynki (ang. *white box*), bardzo często z użyciem podglądu stanu w środowisku TwinCAT System Manager. Takie testowanie pozwala stosunkowo łatwo wyszukać źródło błędu i je wyeliminować.

Autor kilka razy przeprowadzał testy stosując metodę czarnej skrzynki (ang. *black box*), nie biorąc pod uwagę zależności wykonywanych czynności, od realizowanego przez sterownik kodu. Kilukrotnie w czasie realizacji projektu do testów zgłaszały się osoby trzecie, które były nim zaintrygowane. Testy wykonane przez takie osoby są niezwykle cenne ze względu na dużą nieprzewidywalność oraz całkowitą niezależność działań od rozwiązań ze względu na brak ich znajomości.

W czasie realizacji autor stosował testowanie oparte na dwóch metodach analizy. Testowanie oprogramowania można wykonywać pod kątem analizy statycznej i dynamicznej. Analiza statyczna polega na sprawdzaniu kodu źródłowego i znajdowaniu w nim błędów bez uruchamiania sprawdzanego kodu. Ta metoda była stosowana poza laboratorium, gdzie brak był dostępu do sterownika i modelu. Podczas analizy dynamicznej oprogramowanie jest uruchamiane i badane pod kątem ścieżki przebiegu i czasu wykonywania. Ta metoda z kolei była najważniejsza i często wyniki tych testów były zaskakujące w stosunku do przeprowadzonych wcześniej z zastosowaniem analizy statycznej.

Ostatnim etapem testów były te przeprowadzone w obecności promotora oraz te wykonane przez niego. Ostatecznie oprogramowanie zostało zatwierdzone i uznane za spełniające wszystkie wstępne założenia przedstawione w podrozdziale 1.2.

5.2 Napotkane problemy

Podczas tworzenia projektu napotkane i przeanalizowane zostały następujące problemy:

- Problem z automatycznym uruchamianiem stworzonego projektu PLC:

Po utworzeniu oprogramowania sterownika PLC oraz po odpowiednim skonfigurowaniu go w oprogramowaniu TwinCAT System Manager tj. linkowaniu zmiennych programu do odpowiednich fizycznych wejść oraz wyjść modelu oraz aktywowaniu tak przygotowanej konfiguracji, które z kolei wymusza zresetowanie systemu nie następuje uruchomienie projektu sterownika PLC. W początkowej fazie autor przełączał się na oprogramowanie TwinCAT PLC Control gdzie logował się do sterownika i ręcznie uruchamiał stworzony przez siebie kod. Niestety to rozwiązanie na dłuższą metę okazało się męczące i czasochłonne. Okazało się, że w sterownikach firmy Beckhoff trzeba w specjalny sposób przygotować oprogramowanie, które ma być uruchamiane w sposób automatyczny, tzn. trzeba utworzyć projekt, który jest bootowalny. Początkowo takie podejście wydało się autorowi bardzo dziwne, ale po dłuższym zastanowieniu oraz kilku rozmowach z bardziej doświadczonymi w branży osobami okazało się, że ma ono swoje plusy. Przykładowo w przypadku tworzenia oprogramowania w fazie rozwojowej reset urządzenia pozwala przerwać całkowicie wykonywanie oprogramowania zawierającego błędy mające destrukcyjny wpływ na model lub w praktyce na obiekt przemysłowy. Po zastosowaniu nowej metody uruchamianie i testowanie tworzonego oprogramowania stało się zdecydowanie prostsze.

Wszystkie problemy zostały rozwiązane i w ostatecznej wersji oprogramowania nie wpływają one w negatywny sposób na pracę modelu.

6 Wnioski

Tworzenie rozbudowanego oprogramowania do obsługi robota 3D pracującego w magazynie wysokiego składowania wymaga rozwiązania wielu problemów programistycznych. Autor w pierwszej kolejności zapoznał się z kursami programowania w środowisku Step 7 oraz WinCC flexible [7, 8, 9] dostępnymi w sieci.

W procesie tworzenia oprogramowania autor zapoznał się z wieloma zagadnieniami typowymi dla sterowników firmy Beckhoff. Przykładowo,

Wykonanie pojedynczego cyklu sterownika stworzonego przez autora oprogramowania w czasie testowania wynosiło od 1 do 3 ms. Czas ten jest zadowalający biorąc pod uwagę, że sterownik bez funkcji użytkownika oraz z pustym blokiem OB1 ma czas wykonania cyklu równy 1 ms oraz to, że oprogramowanie przynajmniej zdaniem autora jest rozbudowane.

Wizualizacja stanu robota stanowi bardzo atrakcyjną formę korzystania z urządzenia, która jest jednocześnie wyjątkowo przystępna dla osób nie znających zagadnień związanych z programowaniem sterowników przemysłowych. Wizualizacja będzie pełniła ważną rolę w prowadzeniu zajęć laboratoryjnych związanych ze sterownikami firmy Beckhoff

7 Bibliografia

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

- [1] Jerzy Kasprzyk: *"Programowanie sterowników przemysłowych"*, Wydawnictwa Naukowo-Techniczne WNT, Warszawa, 2007.
- [2] *"Programowalne sterowniki PLC w systemach sterowania przemysłowego"*, Politechnika Radomska, Radom, 2001.
- [3] Andrzej Maczyński: *"Sterowniki programowalne PLC. Budowa systemu i podstawy programowania"*, Astor, Kraków, 2001.
- [4] Zbigniew Seta: *"Wprowadzenie do zagadnień sterowania. Wykorzystanie programowalnych sterowników logicznych PLC."*, MIKOM Wydawnictwo, Warszawa, 2002.
- [5] Janusz Kwaśniewski: *"Programowalne sterowniki przemysłowe w systemach sterowania"*, Wyd. AGH, Kraków, 1999.
- [6] Dokumentacja producenta: *"Servo Drive AX5000"*, 28 wrzesień 2012.
- [7] Materiały szkoleniowe: Rudolf W. Meier: *"AX5000 – Motion control for high dynamic positioning"*, luty 2007.
- [8] Materiały szkoleniowe: *"Pierwsze kroki w TwinCAT System Manager i TwinCAT PLC Control"*, 29 październik 2007.
- [9] Materiały szkoleniowe: *"Podstawy obsługi programów: TwinCAT System Manager i TwinCAT PLC Control"*, 15 grudzień 2006.
- [10] Andrzej Gawryluk: *"EtherCAT – to nie takie trudne. Ethernet jako sieć real-time"*, Elektronika Praktyczna, 2/2010.
- [11] Michał Gosk: *"Szybkość, niezawodność, doskonała synchronizacja. EtherCAT - system przyszłości."*, Magazyn Sensor, 1/2013, kwiecień.

8 Spis rysunków, tablic i kodów źródłowych

8.1 Spis rysunków

8.2 Spis tablic

Tablica 1: Dostępne stanowiska laboratoryjne	2
--	---

8.3 Spis kodów źródłowych

9 Załączniki

- Oświadczenie o autorstwie,
- Płyta CD, na której znajdują się:
 - Kod oprogramowania wewnętrznego TwinCAT PLC Control,
 - Pliki projektu TwinCAT System Manager,
 - Kod wizualizacji oraz pliki projektu !?,
 - Plik wykonywalny wizualizacji !?
 - LaTeXowe pliki pracy magisterskiej,
 - Filmy prezentujące działanie projektu.