



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, 12 września 2013

Streszczenie

Głównym celem pracy było zaprojektowanie i zrealizowanie stanowiska laboratoryjnego do badania zależności czasowych w protokole EtherCAT. Wykorzystać należało dostępny na uczelni w laboratorium sprzęt i przedstawić na nim obrazowo, że w protokole tym występują opóźnienia. Dla ich zobrazowania należało wykorzystać dostępne na stanowiskach silniki oraz przygotować przy użyciu dostępnych narzędzi system wizualizacji.

Prawdopodobnie po zakończeniu projektu posłuży on jako podstawa do stworzenia i przeprowadzenia ćwiczeń laboratoryjnych prowadzonych w ramach działalności dydaktycznej prowadzonej przez pracowników Zespołu Przemysłowych Zastosowań Informatyki.

Na końcu dokument zawiera podsumowanie oraz wnioski wyciągnięte z przeprowadzonych badań. Przedstawione zostały również liczne możliwe do zrealizowania i przeprowadzenia w przyszłości eksperymenty. Pozwolą one prawdopodobnie na pogłębienie wiedzy o protokole jak również sprawdzenie opóźnień czasowych występujących w protokole w inny sposób.

Spis treści

1	Wstęp	3
1.1	Geneza	3
1.2	Stanowisko laboratoryjne	9
1.2.1	Sterownik PLC	9
1.2.2	Komputer	13
1.3	Plan pracy	13
2	Analiza tematu	14
2.1	EtherCAT	14
2.1.1	Przetwarzanie „w locie”	15
2.1.2	Telegram EtherCAT	17
2.1.3	Topologia	20
2.1.4	Warstwa fizyczna	21
2.1.5	Synchronizacja	21
2.1.6	Realizacja węzłów EtherCAT	22
2.1.7	EtherCAT Technology Group	24
3	Stanowiska badawcze	26
3.1	Niezależne podstawowe stanowiska	26
4	Badania	29
4.1	Czas stabilizacji sieci po odłączeniu i podłączeniu dowolnego urządzenia	29
4.1.1	Pojedyncze urządzenie	29
4.1.2	Wyspa z modułami I/O	30
4.1.3	Problemy	31
4.2	Badania niewykonalne	33
4.2.1	Zbadanie innych topologii	33
4.2.2	Zbadanie opóźnień na poziomie transmisji pojedynczych ramek	33
5	Uruchamianie i testowanie	34
5.1	Napotkane problemy	34
6	Wnioski	36
6.1	Perspektywy dalszych badań	37
7	Bibliografia	41
8	Spis rysunków, tablic i kodów źródłowych	44
8.1	Spis rysunków	44
8.2	Spis tablic	45
8.3	Spis kodów źródłowych	45
9	Załączniki	46

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 [6, 7], a praca nad tym tematem wydaje się być pomocna i wartościowa w przyszłej pracy zawodowej lub na ewentualnym dalszym etapie kształcenia.

1.1 Geneza

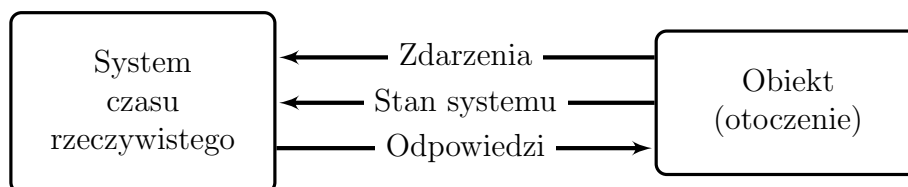
Informatyka przemysłowa to dziedzina wiedzy z pogranicza nauk informatycznych oraz szeroko pojętych nauk o technologiach przemysłowych łącząca te dwa zakresy wiedzy. Informatyka przemysłowa to dziedzina, która jest niezwykle pomocna, zwłaszcza przy analizach danych, ale również podczas tworzenia funkcjonalnych i wydajnych maszyn dla nowoczesnych gałęzi gospodarki. Specjaliści z zakresu informatyki przemysłowej zajmują się takimi zagadnieniami jak monitorowanie produkcji przemysłowej czy też analizowaniem wybranych danych, które później służą do porównań. Niezwykle ważne jest wykorzystanie tych danych, które dostarczą istotnej wiedzy dla każdego przemysłu, a także sterowanie produkcji przemysłowej, która pomaga w dynamicznym rozwoju każdej gałęzi funkcjonującej w gospodarce. Przemysł to szczególna dziedzina gospodarki, która aby się rozwijać potrzebuje nowoczesnych rozwiązań. Jedną z nich jest informatyka, która pozwala na udoskonalanie wyprodukowanych już maszyn przemysłowych, a także tych, które dopiero co są w trakcie rozwoju technologicznego. Informatyka to aspekt, który zwraca uwagę na nowoczesne zastosowanie komputerów w celach przemysłowych. Wielu informatyków już teraz współpracuje z firmami działającymi w przemyśle, gdyż współpraca opłaca się obydwu stronom.

Systemy komputerowe realizują bardzo odpowiedzialne zadania a współczesna technologia stawia coraz poważniejsze wymagania związane przede wszystkim z gwarantowanym i nieprzekraczalnym czasem realizacji pojedynczego cyklu sterowania bądź regulacji, co wymaga wprowadzenia rozwiązań gwarantujących determinizm czasowy. Determinizm czasowy oznacza, że wyniki działania systemu komputerowego pojawiają się w określonym i skończonym czasie.

System czasu rzeczywistego (ang. Real-Time System, w skrócie RTS) wg jednej z definicji jest to system komputerowy, w którym obliczenia są prowadzone równolegle z przebiegiem zewnętrznego procesu w otoczeniu systemu (w sterowanym obiekcie), mają one na celu nadzorowanie, sterowanie oraz terminowe reagowanie na zdarzenia zachodzące w tym procesie. Uproszczony schemat działania przedstawia Rysunek 1.

System czasu rzeczywistego jest takim systemem, który do poprawnego działania musi spełniać następujące warunki:

1. Warunki logiczne – odpowiedź systemu musi zawsze być prawidłowa przy uwzględnieniu stanu systemu, tzn. odpowiedź zależy od stanu otoczenia i występujących zdarzeń oraz jest zgodna z założeniami technologicznymi,



Rysunek 1: Schemat działania systemu czasu rzeczywistego.

2. Warunki czasowe – odpowiedź systemu musi nadejść we właściwym czasie (przed przekroczeniem czasu granicznego dla danego systemu)

Dla określenie podziału systemów czasu rzeczywistego przydatne jest pojęcie funkcji zysku. Jest to funkcja zależna głównie od czasu i dokonuje odwzorowania czasu wykonania zadania na korzyść wynikającą z jego wykonania. Wartość korzyści niekoniecznie jest wielkością wymiarowaną. Źródłem występujących ograniczeń czasowych są zazwyczaj zjawiska fizyczne zachodzące w świecie rzeczywistym kontrolowanym przez system.

Zadanie jest uznawane za zrealizowane jako poprawne, jeśli w chwili jego zakończenia wartość funkcji zysku jest większa od zera.

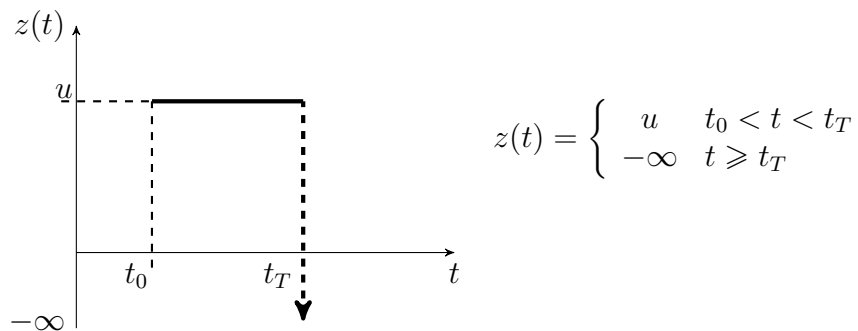
Założenia dla wykresów funkcji zysku:

- t_0 jest to moment w którym zadanie zostało zlecone oraz rozpoczęło się jego wykonywanie,
- t_T jest to graniczny moment w którym przetwarzanie może zostać zakończone,
- w przypadku jak na Rysunku 2c funkcja $z(t)$ jest funkcją ciągłą w punktach t_1 i t_T ,
- Dla wartości $t \leq t_0$ funkcji zysku nie wyznacza się, gdyż jest to sytuacja nie możliwa, tzn. obliczany byłby zysk z zakończenia zadania przed jego rozpoczęciem co nie może się zdarzyć i nie ma najmniejszego sensu.

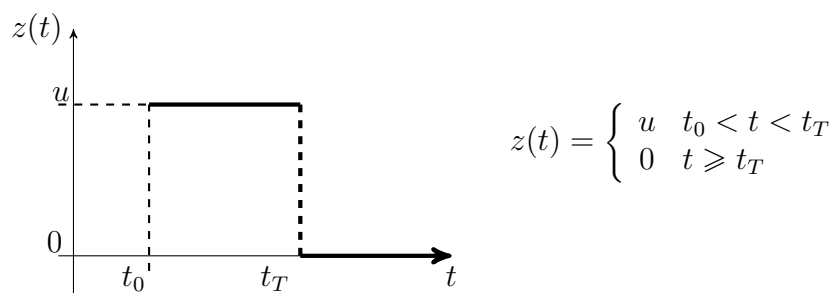
W tego typu systemach przekształcanie danych wymienianych między nim, a środowiskiem zewnętrznym zachodzi w deterministycznie określonym czasie. W opisach stosuje się pojęcie terminu (ang. deadline), oznaczające najdłuższy dopuszczalny czas reakcji systemu na wystąpienie zdarzenia. Szybkość działania systemu czasu rzeczywistego nie jest tak naprawdę bardzo ważna, istotne jest, aby spełnione były bezwarunkowo przyjęte ograniczenia czasowe.

Jak wynika z przedstawionych na Rysunku 2 teoretycznych funkcji zysku rozróżniamy trzy typy systemów czasu rzeczywistego:

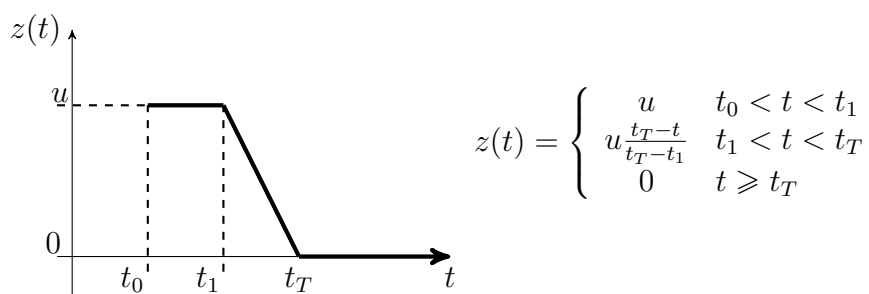
- systemy o ostrych ograniczeniach czasowych (ang. hard real-time) – gdy przekroczenie terminu powoduje poważne, a nawet katastrofalne skutki, jak np. zagrożenie życia lub zdrowia ludzi, uszkodzenie lub zniszczenie urządzeń, przy czym nie jest istotna wielkość przekroczenia terminu a jedynie sam fakt jego przekroczenia. Wynika z tego, że informacja wygenerowana przez system po przekroczeniu czasu granicznego t_T oznacza poważny błąd systemu.
- systemy o mocnych ograniczeniach czasowych (ang. firm real-time) – gdyaa fakt przekroczenia terminu powoduje całkowitą nieprzydatność wypracowanego przez



(a) System typu „hard”



(b) System typu „firm”



(c) System typu „soft”

Rysunek 2: Funkcja zysku dla systemów czasu rzeczywistego

system wyniku, jednakże nie oznacza to zagrożenia dla ludzi lub sprzętu; pojęcie to stosowane jest głównie w opisie teoretycznym baz danych czasu rzeczywistego,

- systemy o miękkich lub łagodnych ograniczeniach czasowych (ang. soft real-time) – gdy przekroczenie pewnego czasu powoduje negatywne skutki tym poważniejsze, im bardziej ten czas został przekroczony; w tym przypadku przez negatywne skutki rozumie się spadek funkcji zysku aż do osiągnięcia wartości zero.

Idealnymi przykładami praktycznymi z życia codziennego wyjaśniającymi podział jak i funkcję zysku są:

1. W przypadku systemów typu „hard real-time” jest to reaktor jądrowy. Załóżmy, że system nadzorujący pracę takiego urządzenia nie zdążyłby wystarczająco szybko zareagować na podnoszenie się temperatury rdzenia i doszłoby do jego przegrzania lub nawet do częściowego stopienia. Jak pokazał przykład awarii reaktora jądrowego w Czarnobylu tego typu zjawiska mają tragiczne i wieloletnie skutki. Jest to idealny przykład, że nie pojawienie się odpowiedzi systemu na czas jest tragiczne w skutkach i nie ma znaczenia o ile czas graniczny został przekroczony.
2. W przypadku systemów typu „soft real-time” jest to bankomat. Jak wiadomo bankomat musi zareagować na żądania zgłaszane przez klienta w skończonym czasie. Jeżeli czas ten będzie zbyt długi, tzn. będzie przekraczał czas graniczny to negatywnym skutkiem będzie spadające zadowolenie klienta wynikające z jakości świadczonej usługi. Wiadomo, że nie wolno ignorować tego faktu, bo niezadowolenie klienta może spowodować, że nie skorzysta on więcej z danej sieci bankomatów. Na tym przykładzie dobrze widać, że przekroczenie czasu granicznego nie jest tragiczne w skutkach, ale powoduje spadek wartości funkcji zysku, którą jest w tym przypadku zadowolenie klienta.

Systemy czasu rzeczywistego najczęściej pracują pod kontrolą specjalnych systemów operacyjnych spełniających reżimy czasu rzeczywistego. System operacyjny czasu rzeczywistego (ang. Real-Time Operating System, w skrócie RTOS) jest to system operacyjny, który został opracowany tak, aby spełniać wymagania narzucone na czas wykonania zadanych operacji. Podstawowym wymaganiem dla systemu typu RTOS jest określenie najdłuższego możliwego czasu po którym urządzenie wypracuje odpowiedź na wystąpienie zdarzenia (określenie przypadku pesymistycznego). Ze względu na kryterium ograniczeń czasowych systemy te dzielimy na:

- Twarde (ang. Hard RTOS) – są to systemy dla których znany jest najdłuższy i nie przekraczalny gwarantowany czas odpowiedzi,
- Miękkie (ang. Soft RTOS) – są to systemy, które starają się odpowiedzieć w możliwie najkrótszym czasie, ale niestety nie jest znany najdłuższy możliwy czas odpowiedzi.

Zaawansowanym problemem występujących w systemach operacyjnych tego typu jest dobór algorytmu szeregowania oraz przydziału czasu. W systemie czasu rzeczywistego ze względu na ograniczone zasoby trzeba wybrać, który z procesów będzie miał w danym momencie przydzielony procesor oraz jak długo przydział taki powinien trwać, aby wszystkie wykonywane procesy spełniły określone dla nich ograniczenia czasowe.

Nr	Nazwa	Producent	Platformy
1	Solaris	Sun Microsystems	Sparc
2	Windows CE	Microsoft	ARM, MIPS, PowerPC, SH, x86, Strong ARM, NEC, VR4111
3	QNX Neutrino	QNX Software Systems	MIPS, MPC8xx, PowerPC, SH, ARM, Strong RM
4	RT Linux	Open Source	ARM, PowerPC, x86, SH3, MIPS
5	LynxOS	LynuxWorks	68K, MIPS, MPC8xx, PowerPC, x86, Sparc
6	VxWorks	Wind River Systems	68K, i869, ARM, MIPS, PowerPC, x86, SH, SPARC
7	eCOS	Open Source	ARM, MIPS, MPC8xx, PowerPC, Sparc

Tablica 1: Systemy czasu rzeczywistego

Na rynku dostępnych jest aktualnie wiele systemów operacyjnych czasu rzeczywistego od różnych producentów lub otwartych, których przykłady przedstawia Tabela 1.

Kolejnym niezbędnym elementem oprócz układów wejściowych i wyjściowych oraz kontrolowanych urządzeń (np. robotów) są deterministyczne sieci transmisyjne. W sieciach przemysłowych dane przesyłane są często, ale są one krótkie (mają niewielki rozmiar), w przeciwieństwie do sieci ogólnego przeznaczenia gdzie przesyła się dane rzadko i w dużej ilości. Dane mają charakter np. wartości pomiaru, rozkazów start/stop, alarmu, przekazania wartości zadanej do zrealizowania przez regulator itd. Bardzo często właściwa informacja w tych sieciach to jeden bit albo jedno słowo.

W sieciach, w których zagwarantowany jest determinizm czasowy transmisji, okres od wysłania pakietu danych od nadawcy do jego odbioru po stronie odbiorcy jest z góry znany i pozostaje niezmienny. Wówczas komunikacja między węzłami sieci może być realizowana w czasie rzeczywistym, łatwiejsza jest też ich synchronizacja. Obie te kwestie są kluczowe dla sprawnego działania systemów sterowania. Dodatkowo sieci takie muszą zapewniać wszelkiego rodzaju mechanizmy kontroli poprawności przesyłu danych oraz zabezpieczać przez ewentualną ich utratą. Kolejnym wymaganiem stawianym przed sieciami tego typu jest odporność na zakłócenia zewnętrzne w tym szczególnie zakłócenia elektromagnetyczne. Rola komunikacji jest krytyczna z punktu widzenia technologii, ekonomii i bezpieczeństwa prowadzenia procesu. Podsumowanie najważniejszych wymagań stawianych przemysłowym sieciom informatycznym:

- Niezawodność,
- Przewidywalność procesu komunikacji:
 - Relacja pomiędzy węzłami w sieci,
 - Praca w czasie rzeczywistym,
- Efektywność w przekazywaniu krótkich wiadomości,
- Standaryzacja interfejsów, łatwość podłączania dużej liczby urządzeń,

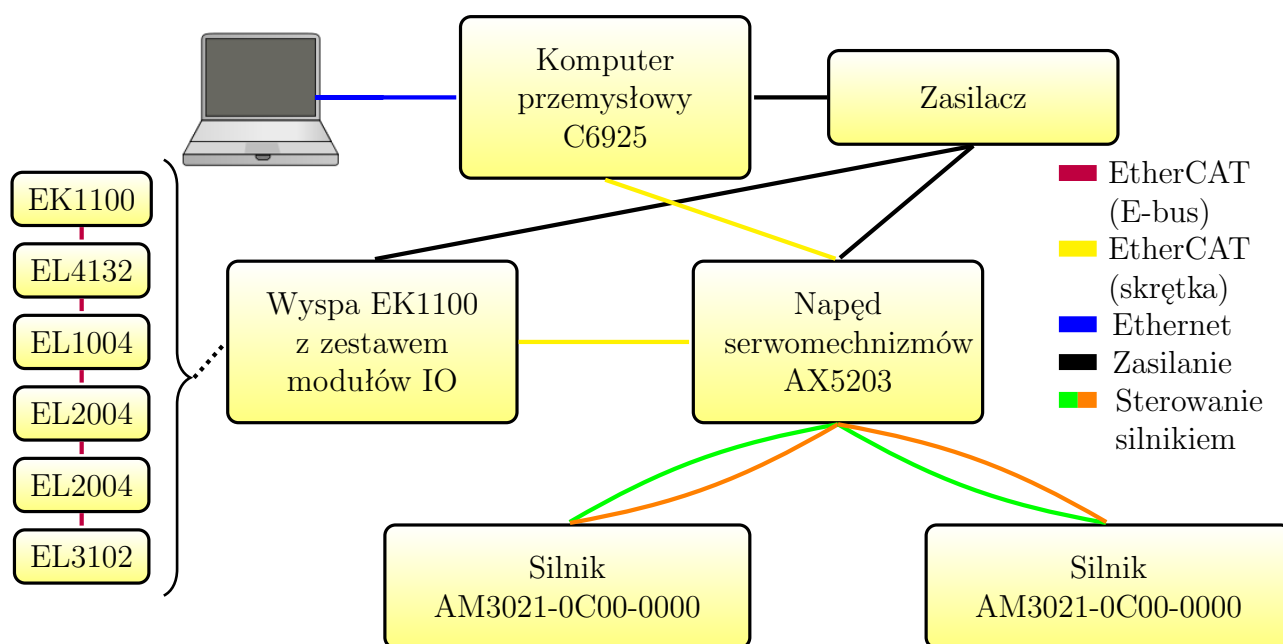
- Możliwość podłączenia do zewnętrznych sieci,
- Łatwość lokalizowania usterek.

Aby system składający się z wielu komponentów był systemem czasu rzeczywistego, konieczne jest spełnianie wymogów przez każdy z elementów składowych. W przypadku systemów informatycznych oznacza to, że zarówno sprzęt, system operacyjny, jak i oprogramowanie aplikacyjne muszą gwarantować dotrzymanie zdefiniowanych ograniczeń czasowych.

Jak wykazuje powyższa geneza systemu czasu rzeczywistego są bardzo ważnym elementem nie tylko w przemyśle, ale również w życiu codziennym. Przemysłowe sieci komputerowe są bardzo istotnym elementem tych systemów, dlatego właśnie istotne jest, aby przebadać i poznać je dokładnie oraz ustalić jakie zależności czasowe w nich występują. Sieć EtherCAT będąca tematem niniejszej pracy bardzo dobrze spełnia wszystkie wymagania stawiane nowoczesnym protokołom pracującym w warunkach przemysłowych dlatego autor uważa, że warto przyrzeć mu się bliżej i przeprowadzić na nim badania oraz znaleźć ewentualne perspektywy na kolejne prace badawcze w przyszłości.

1.2 Stanowisko laboratoryjne

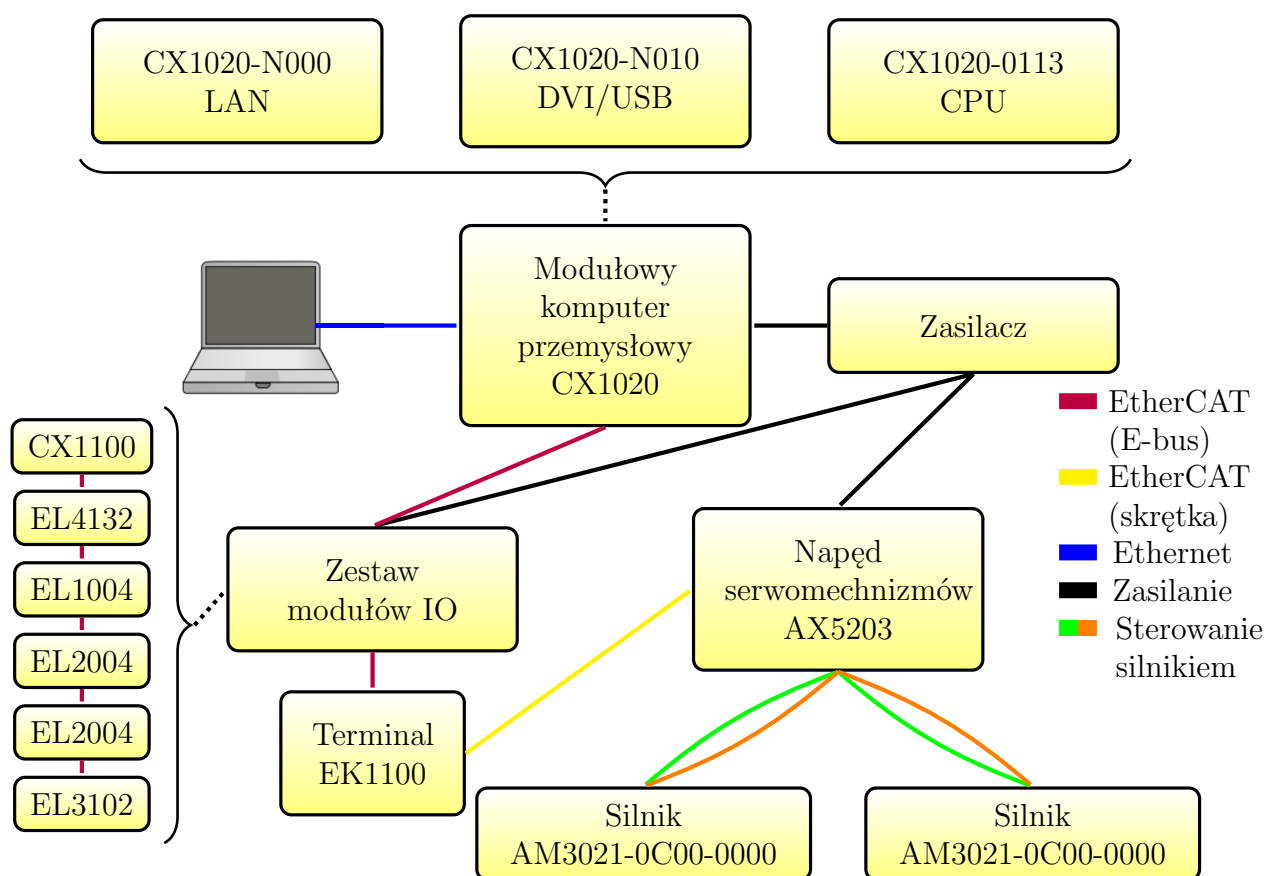
Na potrzeby realizacji projektu wykorzystano dwa różne istniejące stanowiska laboratoryjne, które składały się z elementów opisanych w Tabeli 2.



Rysunek 3: Schemat stanowiska typu CP

1.2.1 Sterownik PLC

W realizacji wykorzystane zostały stanowiska firmy Beckhoff wyposażone w jednostki centralne pracujące pod kontrolą Windowsa CE (Microsoft Windows Compact Edition).



Rysunek 4: Schemat stanowiska typu CX

Stanowisko typu CP (Rysunek 3) Adres IP: 157.158.57.121	Stanowisko typu CX (Rysunek 4) Adres IP: 157.158.57.47
<ol style="list-style-type: none"> 2 silniki AM3021-0C00-0000, Wyspa EK1100 z zestawem modułów IO: <ul style="list-style-type: none"> Terminal sieci EtherCAT EK1100, 2-kanalowy moduł wyjść analogowych EL4132, 4-kanalowy moduł wejść cyfrowych EL1004, 2 4-kanalowe moduły wyjść cyfrowych EL2004, 2-kanalowy moduł wejść analogowych EL3102, Napęd serwo mechanizmów AX5203 (2 osiowy), Komputer przemysłowy C6925, Zasilacz. 	<ol style="list-style-type: none"> 2 silniki AM3021-0C00-0000, Zestaw modułów IO: <ul style="list-style-type: none"> 2-kanalowy moduł wyjść analogowych EL4132, 4-kanalowy moduł wejść cyfrowych EL1004, 2 4-kanalowe moduły wyjść cyfrowych EL2004, 2-kanalowy moduł wejść analogowych EL3102, Terminal sieci EtherCAT EK1100, Napęd serwo mechanizmów AX5203 (2 osiowy), Modułowy komputer przemysłowy CX1020: <ul style="list-style-type: none"> Interfejs USB/DVI CX1020-N010 , Ethernet CX1020-N000, CPU CX1020-0113, Zasilacz CPU i magistrali I/O CX1100, Zasilacz.

Tablica 2: Dostępne stanowiska laboratoryjne

Na jednostce takiej uruchamiane są programy do sterowania z poziomu komputera (ang. Soft PLC). Jest to rozwiązanie alternatywne dla klasycznych sterowników swobodnie programowalnych w postaci dedykowanego urządzenia (ang. Hard PLC), nazywanych przez niektórych prawdziwymi (ang. True PLC). Koncepcja ta powstała i jest rozwijana, ponieważ te klasyczne sterowniki posiadają zbyt małe możliwości obliczeniowe oraz szybkość działania jednostki centralnej. W tradycyjnych rozwiązaniach niestety zwiększanie tych możliwości (ilość dostępnej pamięci oraz szybkości działania) powoduje bardzo szybki wzrost ceny gotowego urządzenia. Niezbędnym elementem konfiguracji zestawu, który przekształcamy w „soft PLC” jest karta komunikacyjna umożliwiająca połączenie sterownika z modułami sygnałowymi i wykonawczymi na obiekcie z wykorzystaniem sieci przemysłowej. Przykładowe zastosowanie programu do sterowania z poziomu komputera zostało szczegółowo opisane w [16]. Tego typu podejście i rozwiązanie ma następujące zalety:

- duże zwiększenie możliwości obliczeniowych przy stosunkowo niewielkim wzroście kosztów,

- możliwość integracji PLC i systemu SCADA na jednym urządzeniu (podobnie jak w panelach operatorskich ze zintegrowanymi sterownikami PLC),
- możliwość zastosowania istniejącej infrastruktury na obiekcie w przypadku przebudowy, należy jedynie podmienić istniejący sterownik typu „hard” na jednostkę wyposażoną w odpowiedni moduł komunikacyjny,
- teoretycznie możliwość zastosowania istniejącego oprogramowania z jednostki „hard PLC”, po modyfikacji ewentualnych różnic między systemami.

Taki „sterownik PLC w komputerze PC” wykorzystuje standardowe języki programowania sterowników PLC (zgodność z normą IEC 61131-3) do tworzenia logiki sterującej takiej jak:

- **IL – Instruction List** to tekstowy język programowania składający się z serii instrukcji, z których każda zaczyna się z nowej linii i zawiera operator z jednym lub więcej argumentem (zależnie od funkcji),
- **LD – Ladder Diagram** jest graficznym językiem programowania, który swoją strukturą przypomina obwód elektryczny. Doskonały do łączenia POUów (Program Organization Units). LD składa się z sieci cewek i styków ograniczonej przez linie prądowe. Linia z lewej strony przekazuje wartość logiczną TRUE, z tej strony zaczyna się też wykonywać linia pozioma.
- **FBD – Function Block Diagram** jest graficznym językiem programowania przypominającym sieć, której elementy to struktury reprezentujące funkcje logiczne bądź wyrażenia arytmetyczne, wywołania bloków funkcyjnych itp.
- **SFC – Sequential Function Chart** to graficzny język programowania, w którym łatwo jest ukazać chronologię wykonywania przez program różnych procesów.
- **ST – Structured Text** jest tekstowym językiem programowania, złożonym z serii instrukcji takich jak If..then lub For...do.
- **CFC – Continuous Function Chart** jest graficznym językiem programowania, który w przeciwieństwie do FBD nie działa w sieci, a w luźno poło onej strukturze, co pozwala na np. stworzenie sprzężenia zwrotnego.

Stanowiska podłączone są do sieci lokalnej Ethernet w laboratorium, więc komunikacja z nimi 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] oraz uczęszczając w toku studiów na zajęcia obowiązkowe oraz specjalizacyjne.

1.2.2 Komputer

Projekt w całości był realizowany na laptopie autora, podłączanym do sieci w laboratorium. Na komputerze uruchomiana była maszyna wirtualna. Na jednej zainstalowane było środowisko TwinCAT do programowania sterownika oraz do tworzenia i uruchamiania wizualizacji. Wizualizacje tworzone w środowisku TwinCAT można uruchomić

bezpośrednio na komputerze wyposażonym w odpowiednie oprogramowanie lub na urządzeniu docelowym po podpięciu do niego monitora (o ile urządzenie docelowe posiada wyjście DVI lub odpowiedni interfejs systemowy w postaci odrębnego modułu).

1.3 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 Analiza tematu

Analiza tematu polegała przede wszystkim na zapoznaniu się z informacjami o standardzie EtherCAT oraz narzędziami programistycznymi do tworzenia oprogramowania sterownika oraz wizualizacji.

Najważniejszym punktem analizy było zapoznanie się z informacjami znajdującymi się na stronie EtherCAT Technology Group, czyli twórców i organizacji zajmującej się popularyzowaniem standardu. Dodatkowo autor przeczytał wiele artykułów polsko oraz angielskich poświęconych właśnie protokołowi EtherCAT. Kilka z nich (starszych) traktowało go jako coś bardzo przyszłościowego i obiecującego, natomiast pozostała część opisywała go już jako coś co aktualnie bardzo szybko popularyzuje się i usprawnia procesy przemysłowe. W czasie tworzenia dokumentu opisującego przeprowadzone badania autor przeszukiwał internet w poszukiwaniu bardziej szczegółowych informacji takich jak budowa nagłówków i znaczenie oraz rozmiar poszczególnych pól. Poszukiwania te pozwoliły natknąć się na sporo interesujących informacji i rozwiązań, które mogą stać się podstawą kolejnych badań w przyszłości.

Autor poznał podstawy obsługi środowiska TwinCAT oraz jego elementów składowych pozwalające na realizację tematu, a w szczególności:

- TwinCAT System Manager - centralne narzędzie konfiguracyjne,
- TwinCAT PLC - narzędzie do tworzenia programów,
- TwinCAT NC/CNC - grupa narzędzi do sterowania osiami w różnych trybach.

2.1 EtherCAT

EtherCAT jest nowoczesnym protokołem sieciowym przeznaczonym do stosowania w aplikacjach przemysłowych, szczególnie takich, które wymagają działania całego systemu w czasie rzeczywistym. Nazwa standardu jest skrótem od hasła: „Ethernet for Control Automation Technology”. W zakresie warstwy fizycznej bazuje na Ethernetie. Dodatkowo zaimplementowano w nim mechanizmy w zakresie organizacji transmisji danych pozwalające na ominięcie głównych ograniczeń sieci Ethernet. Dzięki temu EtherCAT jest obecnie jednym z popularniejszych oraz szybciej rozwijających się protokołów komunikacyjnych w przemyśle.

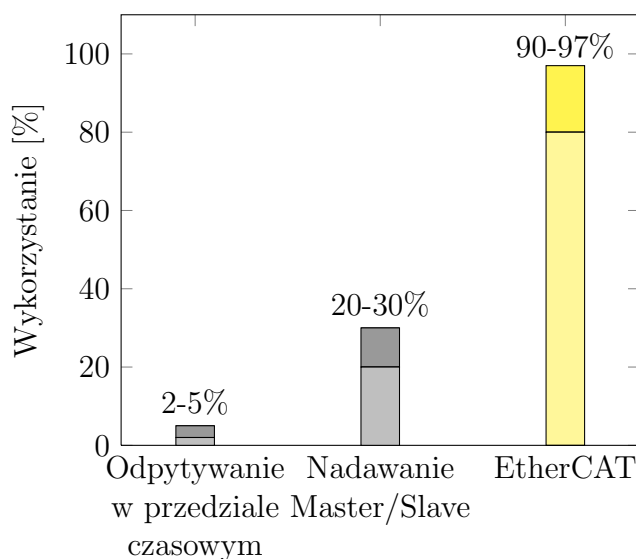
2.1.1 Przetwarzanie „w locie”

W dużej części aplikacji przemysłowych dane mają mały rozmiar rzędu pojedynczych bajtów. Wykorzystując standardowy Ethernet i jego ramki stosunek danych użytecznych do narzutu protokołu jest bardzo niekorzystny. W celu zapewnienia determinizmu oraz zwiększenia przepustowości stosuje się w standardach przemysłowych różne rozwiązania. Jednym z przykładów jest zastępowanie procedury dostępu do medium transmisyjnego z wykorzystaniem wielodostępu z wykrywaniem nośnej oraz detekcją kolizji (ang. Carrier Sense Multiple Access/with Collision Detection w skrócie CSMA/CD) na m.in. mechanizm odpytywania. Nie jest istotnym jaką metodę dokładnie zastosujemy dopóki ramki

są rozsyłane pojedynczo z oraz do urządzeń. Działania te nie wyeliminuje problemu marowania przepustowości kanału transmisyjnego i jego wykorzystania. Dlatego ten właśnie element transmisji danych z wykorzystaniem Ethernetu został potraktowany bardzo szczególnie przy projektowaniu standardu EtherCAT.

Zatem zamiast standardowej dla Ethernetu transmisji pakietowej z koniecznością odtworzenia pofragmentowanych danych zastosowano mechanizm wykorzystujący telegramy zbudowane z datagramów, które są szczegółowo opisane w podrozdziale 2.1.2. Rozwiązanie to polega na tym, że w pojedynczej ramce są zawarte informacje przeznaczone dla wielu różnych węzłów podrzędnych. Transmisja takiej ramki inicjowana jest przez węzeł nadrzędny, a następnie przechodzi ona przez kolejne węzły podrzędne sieci, które przetwarzają ją w locie. W takim podejściu twórcy standardu EtherCAT potraktowali ramkę ethernetową jak pewnego rodzaju pamięć RAM, do której zapisywane i z której odczytywane są dowolne informacje w oparciu o adresy lokalizujące pożądane dane w tej pamięci. Przetwarzanie to polega, że w momencie odebrania ramki sprawdzane jest czy znajduje się w niej informacja przeznaczona dla tego właśnie węzła. Jeżeli zostanie wykryte, że tak jest to węzeł odczytuje odpowiedni fragment danych oraz uzupełnia ewentualnie ją o informacje potwierdzające odbiór lub inne wymagane przez węzeł nadrzędny. Następnie ramka jest przesyłana do kolejnego w topologii węzła lub zwraca jeśli dany węzeł jest ostatnim w sieci.

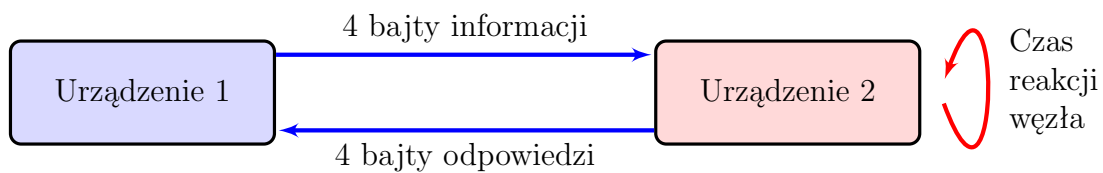
Dzięki takiemu podejściu protokół charakteryzuje się bardzo dużym wykorzystaniem kanału transmisyjnego w porównaniu do odpytywania w przedziale czasowym (ang. Polling Timeslicing) oraz nadawania Master/Slave (ang. Broadcast Master/Slave) znanych ze zwykłego Ethernetu co pokazano na Rysunku 5



Rysunek 5: Współczynnik wykorzystania kanału transmisyjnego w Ethernetie (dwa pierwsze wykresy od lewej strony) i EtherCAT

Minimalny rozmiar ramki ethernetowej wynosi 8 bajtów. Załóżmy przykładowo, że urządzenie okresowo przesyła 4 bajty informacji, na przykład informację o swoich aktualnych ustawieniach, a w odpowiedzi otrzymuje również 4 bajty danych, na przykład zestaw komend i informacji kontrolnych, przy założeniu nieskończonego krótkiego czasu odpowiedzi

węzła, użyteczna przepustowość wyniesie zaledwie $\frac{4}{84} \approx 4,8\%$. Jeżeli średni czas odpowiedzi będzie dłuższy, na przykład wyniesie $10 \mu s$, to użyteczna przepustowość spadnie do zaledwie $1,9\%$.



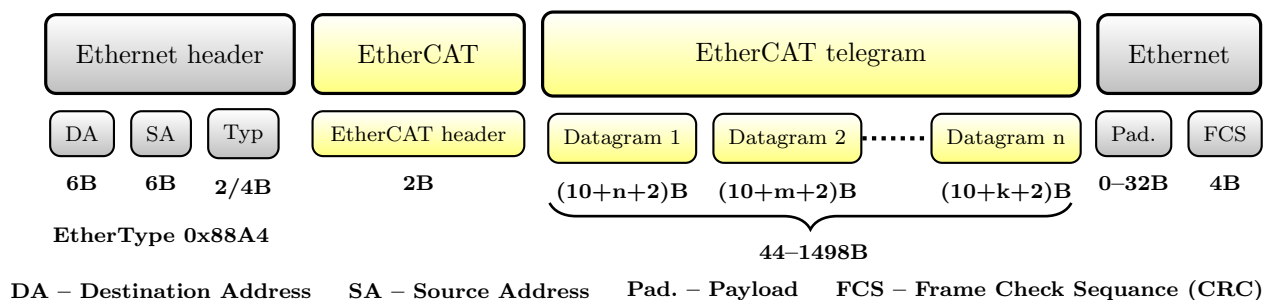
Rysunek 6: Przykład transmisja małej ilości danych (4 bajty) zwykłym Ethernetem.

Wydażność tego rozwiązania jest dosyć duża, choć zależy od liczby elementów podłączonych do sieci. W praktyce czas opóźnienia nie wzrasta powyżej $1 ms$ i zazwyczaj jest znacznie (kilku- lub kilkunastokrotnie) krótszy. Czas synchronizacji nie przekracza $1 \mu s$. Niniejsza praca miała na celu przebadanie i sprawdzenie czy faktycznie protokół działa tak dobrze jak zapewniają jego twórcy.

Bardzo ważnym elementem węzła sieci jest tak zwana jednostka zarządzania pamięcią FMMU (ang. fieldbus memory management unit). Odpowiada ona między innymi za uniezależnienie szybkości transferu danych od wydajności i mocy obliczeniowej jednostki lokalnej CPU, kontrolę ruchu bez opóźnień oraz za odwzorowanie adresu logicznego na adres fizyczny.

2.1.2 Telegram EtherCAT

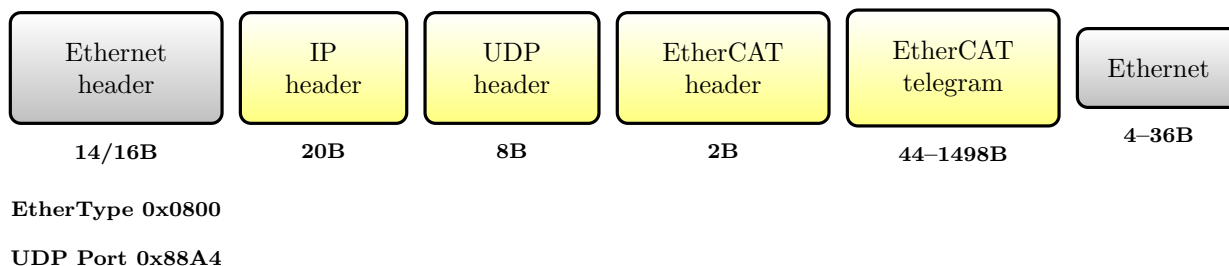
Jak pokazano na Rysunku 7, telegram EtherCAT jest upakowany w ramce Ethernet i zawiera jeden lub więcej datagramów EtherCAT dostarczanych do urządzeń podrzędnych. Dane pomiędzy węzłami są przekazywane jako obiekty danych procesowym (ang. process data objects, w skrócie PDO). Każdy obiekt tego typu zawiera adres konkretnego węzła lub kilku węzłów typu slave.



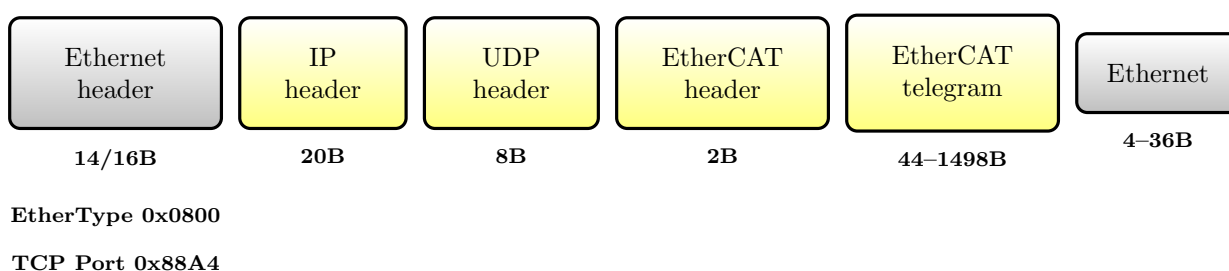
Rysunek 7: Ramka w transmisji EtherCAT i jej podział na datagramy

Dane w systemach opartych na sieciach EtherCAT mogą być przesyłane między różnymi sieciami poprzez protokół UDP, a nie tylko w ramach jednej podsieci. Tam gdzie konieczny jest routing pakietów w oparciu o protokół IP, ramki EtherCAT są przesyłane w ramach pakietów protokołu UDP lub TCP (Rysunki 8 oraz 9). Pakiety tego typu

mogą zostać uformowane i nadane z wykorzystaniem zwykłych urządzeń ethernetowych, dzięki czemu możliwe jest sterowanie aparaturą polową EtherCAT z poziomu klasycznego komputera PC wyposażonego w kartę sieciową.



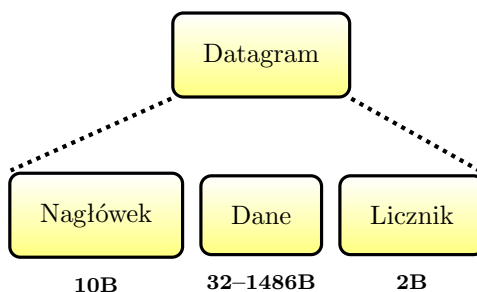
Rysunek 8: Ramka w transmisji EtherCAT z uwzględnieniem UDP i IP



Rysunek 9: Ramka w transmisji EtherCAT z uwzględnieniem TCP i IP

Rozwiązanie to zapewnia elastyczną komunikację pomiędzy urządzeniami pracującymi w standardzie Ethernet oraz EtherCAT, a także umożliwia wymianę danych pomiędzy urządzeniami znajdującymi się w segmentach sieci podłączonych do różnych routerów. W takim przypadku prędkość wymiany danych zależy w dużym stopniu od prędkości działania routerów, co należy uwzględnić przy ewentualnym projektowaniu systemu pracującego w rozdzielonej sieci.

Każdy datagram EtherCAT jest komendą, która zawiera zgodnie z Rysunkiem 10 nagłówek, dane i licznik roboczy. Nagłówek i dane są używane do wyspecyfikowania operacji, które muszą być wykonane przez węzeł podrzędny, natomiast licznik roboczy jest aktualizowany przez niego informując węzeł nadrzędny, że odebrał on i przetwarza komendę.



Rysunek 10: Budowa datagramu

Nagłówek datagramu przedstawiony na Rysunku 11. Jak widać składa się on ze sporej ilości pól wymagających dokładnego objaśnienia.

Cmd Typ rozkazu EtherCAT (ang. EtherCAT Command Type)

Idx Indeks jest numerycznym identyfikatorem używanym przez węzeł nadrzędny do wykrywania zdublowanych lub zagubionych datagramów, pole to nie powinno być nigdy modyfikowane przez węzły podrzędne

Address Adres urządzenia zapisany w sposób zależny od trybu adresacji

Len Długość danych zawartych w analizowanym datagramie (rozmiar pola danych)

R Pole zarezerwowane, powinno mieć zawsze wartość 0

C Pole zabezpieczające ramkę przed zapętleniem

0: oznacza, że przetwarzana ramka nie jest zapętłona

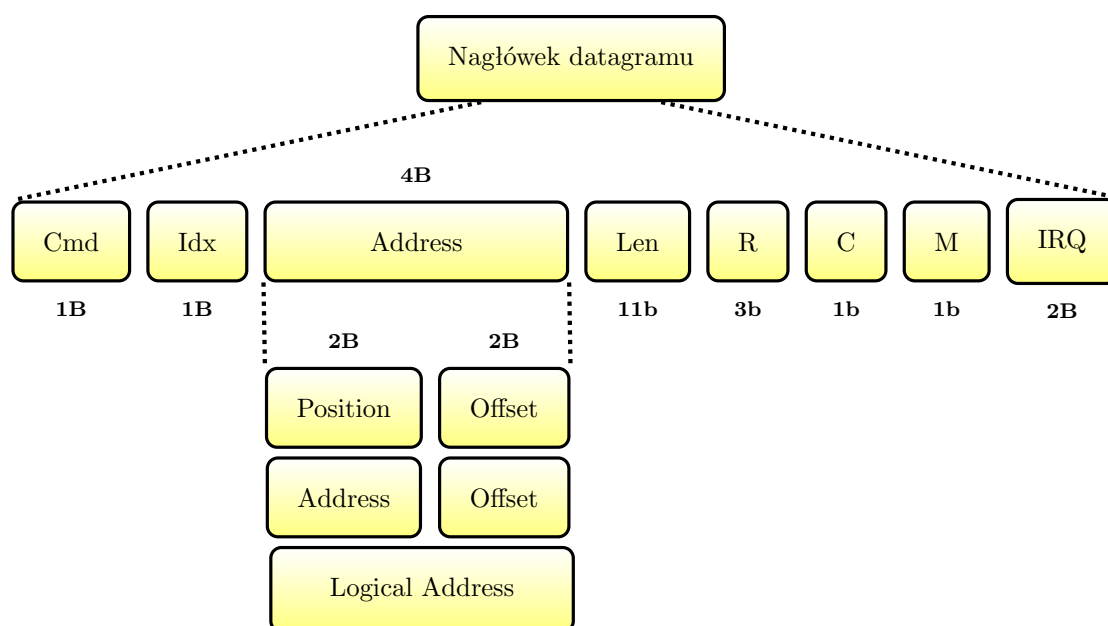
1: oznacza, że za przetwarzana ramka wykonała już jeden obieg pętli

M Pole to określa czy w aktualnie przetwarzanym telegramie są kolejne datagramy

0: oznacza, że przetwarzany datagram jest ostatni

1: oznacza, że za przetwarzanym datagramem są kolejne

IRQ Pole żądania zdarzeń od węzłów podrzędnych, pole jest wynikiem sumy logicznej (ang. logical OR) żądań wszystkich węzłów podrzędnych

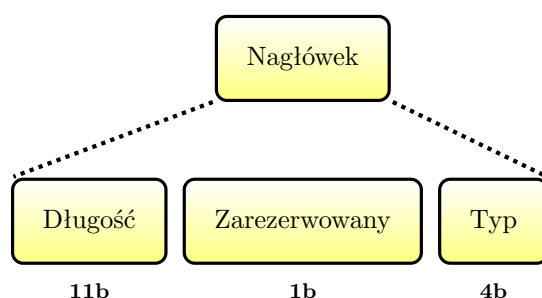


Rysunek 11: Budowa nagłówka datagramu

Oprócz części zawierającej dane pogrupowane w datagramy telegram EtherCAT zawiera również nagłówek, którego budowę przedstawia Rysunek 12. Pierwsze pole określa

długość wszystkich datagramów w danym telegramie, która zależy od ilości węzłów oraz długości wiadomości. Drugie pole jest bitem zarezerwowanym, a jego wartość powinna wynosić 0. Ostatnie pole określa typ protokołu EtherCAT. Typ ten definiuje typ wiadomości, co pozwala na prawidłową interpretację danych. Standard przewiduje 3 dopuszczalne typy:

- Typ 1: Protokół EtherCAT dla urządzeń – Wymiana Datagramów (ang. EtherCAT Device Protocol, EtherCAT Datagram(s)),
- Typ 4: EAP Wymiana danych procesowych – wymiany cykliczne (ang. EtherCAT Automation Protocol, Process data communications),
- Typ 5: EAP Wymiana komunikatów na żądanie (wymiany acykliczne (ang. EtherCAT Automation Protocol, Mailbox communication)).

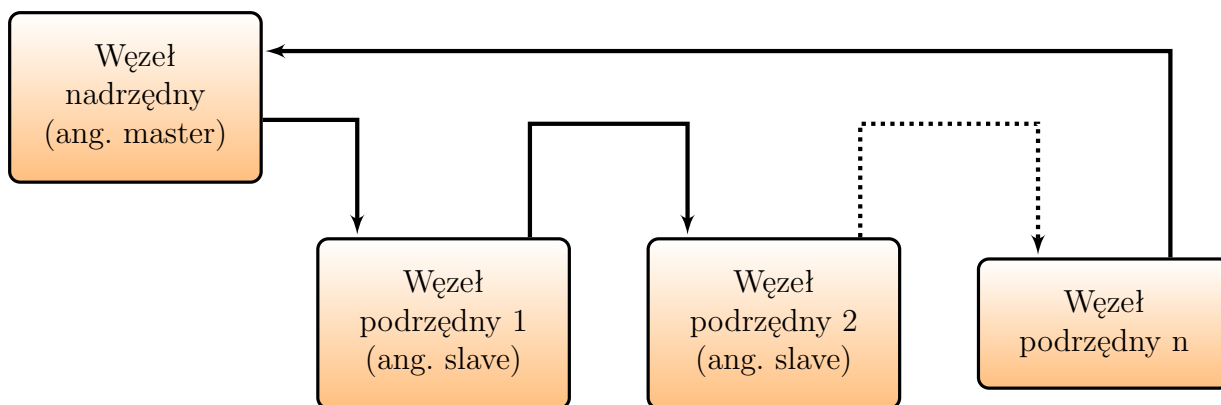


Rysunek 12: Budowa nagłówka EtherCAT

2.1.3 Topologia

Kolejne ramki nadawane są przez urządzenie pełniące rolę kontrolera i przesyłane do najbliższego urządzenia podrzędnego. Urządzenie to ma przydzielony adres, który jednoznacznie identyfikuje datagram, dzięki czemu może odczytać przeznaczone dla siebie dane. Niezależnie od tego, czy urządzenie coś zapisało, czy też tylko odczytywało fragment ramki, przesyła ją dalej do kolejnego z urządzeń zgodnie z tym co pokazano na Rysunku 13. Duża szybkość tej metody transmisji wynika m.in. z tego, że nie ma potrzeby dekodowania całej ramki danych. Pozwala to błyskawicznie przekazywać ramki pomiędzy urządzeniami.

Łatwo zauważyć, że opisana procedura transmisji wymaga zastosowania topologii magistrali, która w przypadku klasycznego Ethernetu jest generalnie nieopłacalna i ma wiele wad. Problem ten został rozwiązany poprzez zwielokrotnienie portów ethernetowych instalowanych w dużej części urządzeń zgodnych z EtherCAT. Dosyć powszechnie spotykane są urządzenia z dwoma lub trzema portami, a te wystarczają już do realizacji topologii gwiazdy, czy nawet zmieszanie topologii gwiazdy i magistrali w ramach jednej sieci i utworzenie struktury mocno redundantnej.



Rysunek 13: Przykładowa topologia sieci

2.1.4 Warstwa fizyczna

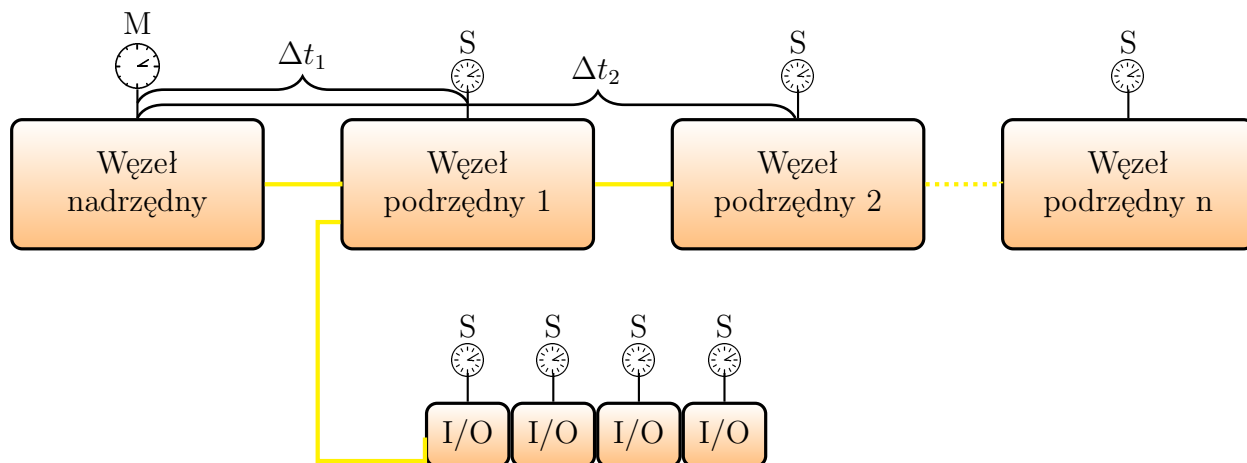
Jako medium komunikacyjne w sieciach EtherCAT można wykorzystać kable miedziane (100Base-TX), światłowody (100Base-FX) lub łącze E-bus w technologii LVDS. Te ostatnie wprowadzono ze względu na to, że transmisja w sieciach EtherCAT często jest realizowana na krótkich dystansach - E-bus sprawdza się w realizacji łączności na odległość do około 10 m. Kable miedziane sprawdzają się na większych odległościach nieprzekraczających 100 metrów, natomiast użyteczna długość światłowodów może dochodzić aż do 20 kilometrów. Jedynym warunkiem związanym z wykorzystaniem światłowodów jest obsługa pełnego duplexu. Wymóg ten jest podyktowany faktem, że transmisja danych jest tak szybka iż z reguły ramka odpowiedzi dociera do urządzenia nadrzędnego zanim całe zapytanie zostanie wysłane. Z tego powodu, aby przesył danych pomiędzy urządzeniami nie był zakłócony musi istnieć możliwość jednoczesnego przekazywania informacji w dwóch kierunkach bez spadku transferu.

W obrębie jednej sieci EtherCAT można dowolnie zmieniać medium transmisyjne zależnie od potrzeb. Na przykład wewnątrz szafy sterowniczej gdzie występują niewielkie odległości między węzłami można zastosować z powodzeniem łącze E-bus, natomiast do połączenia szafy z modułami znajdującymi się przy maszynach wykonawczych możemy zastosować kabel miedziany lub światłowód w przypadku zdecydowanie większych odległości oraz prowadzenia przewodów w otoczeniu występowania zaburzeń elektromagnetycznych. Niestety operacja takiego łączenia wymaga zastosowania dodatkowych modułów. Na przykład, aby połączyć ze sobą swykłą skrętke miedzianą oraz przewód światłowodowy należy zastosować moduł sprzęgający EK1501 oraz terminal przyłączy EK1521 (oba produkcji firmy Beckhoff).

2.1.5 Synchronizacja

Idealna synchronizacja elementów składowych systemu jest bardzo istotna, a szczególnie w przypadku równoległej realizacji zależnych od siebie zadań. Specjalnie opracowana dla protokołu EtherCAT technika zegara rozproszonego pozwala zsynchronizować ze sobą urządzenia z dewiacją mniejszą niż $1 \mu s$. Podejście to polega na wykorzystaniu znaczników czasu zapisywanych przez każdy węzeł podrzędny. Na ich podstawie węzeł master oblicza opóźnienia propagacji sygnałów dla każdego węzła slave. Zegar w każdym węźle

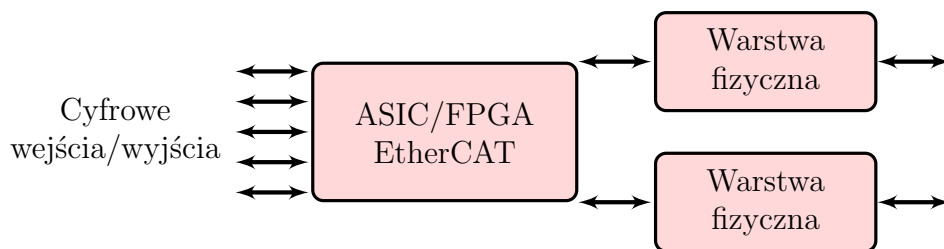
podrzednym jest regulowany z wykorzystaniem wyliczonych opóźnień. Po zainicjowaniu połączenia w celu utrzymania synchronizacji zegarów muszą one przekazywać między sobą regularnie informację, by uniknąć powstania ewentualnych różnic. Rozwiązanie to jest zgodne z protokołem precyzyjnej synchronizacji zegara dla sieciowych systemów kontrolno-pomiarowych lub inaczej protokół czasu precyzyjnego (ang. Precision Time Protocol, w skrócie PTP) IEEE 1588 opisanym w [21].



Rysunek 14: Schemat pracy zegarów rozproszonych

2.1.6 Realizacja węzłów EtherCAT

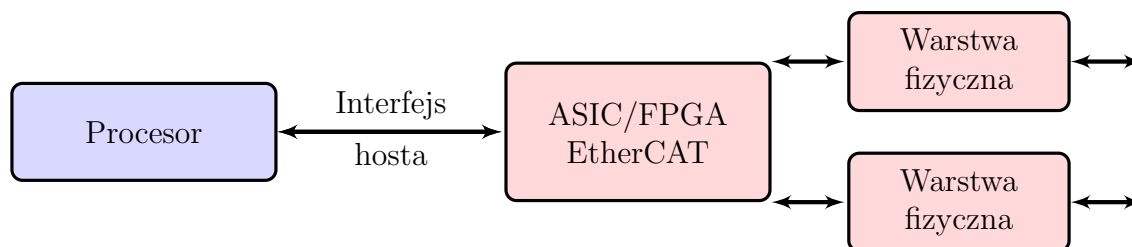
Wiele najprostszych i najtańszych urządzeń z interfejsem EtherCAT można zrealizować z wykorzystaniem pojedynczego układu scalonego FPGA lub ASIC. Przykładami takich prostych urządzeń z zastosowaniem tego rozwiązania są moduły wejść/wyjść cyfrowych. Tego typu węzły nie wymagają tworzenia dodatkowego oprogramowania, ponieważ cała funkcjonalność jest realizowana w pełni sprzętowo. Uogólniona i uproszczona architektura tej metody realizacji została przedstawiona na Rysunku 15.



Rysunek 15: Budowa węzła EtherCAT z wykorzystaniem pojedynczego układu FPGA lub ASIC.

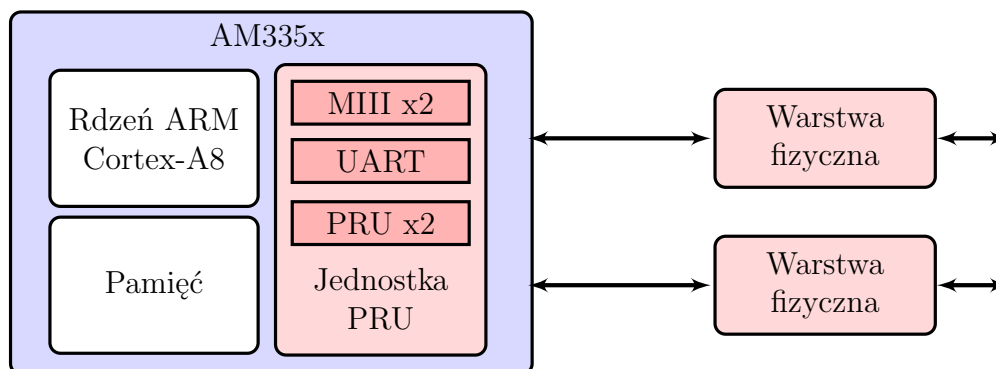
Jeżeli węzeł potrzebuje dodatkowej mocy obliczeniowej lub wymaga realizacji jakiegoś złożonego oprogramowania, którego nie da się zrealizować w prosty sposób sprzętowo (w układzie FPGA) do układu ASIC/FPGA EtherCAT dołączany jest zewnętrzny procesor często wyposażony w pamięć typu Flash tak jak na Rysunku 16. Procesor taki ma

zapewnić obsługę przetwarzania na poziomie aplikacji. Niestety koszt tego typu architektury jest wyższy niż w prostszym przypadku pozbawionym zewnętrznej jednostki, ale konstruktor bazujący na tym rozwiązaniu ma większe pole manewru w doborze procesora odpowiedniego dla wymagań i budżetu realizowanego projektu.



Rysunek 16: Budowa węzła z wykorzystaniem układu ASIC/FPGA EtherCAT z dołączonym zewnętrznym procesorem.

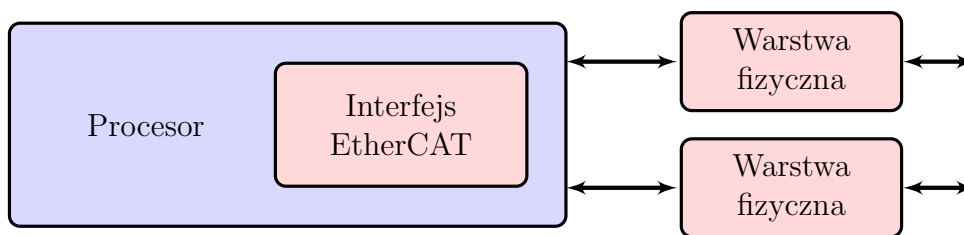
Kolejnym możliwym do wykorzystania rozwiązaniem jest zastosowanie układu FPGA z wbudowanym procesorem jak na Rysunku 17. Wspólną cechą przedstawionych dotychczas możliwych konstrukcji jest fakt, że wymagają z reguły zastosowania dwóch układów. Wynika z tego niestety, że zajmują więcej miejsca oraz zwiększają koszty urządzenia docelowego. Alternatywą pozwalającą wyeliminować oba opisane problemy jest zastosowanie elementów jednoukładowych. Ich wykorzystanie pozwala zredukować całkowity koszt konstrukcji nawet o 30%.



Rysunek 17: Budowa układu FPGA z wbudowanym procesorem.

Jednym z przykładów opisanego jednoukładowego rozwiązania są mikrokontrolery z rdzeniem ARM Cortex-A8 z rodziny Sitara AM335x produkowanymi przez amerykańską firmę Texas Instruments (Rysunek 18). Kluczowym elementem takiego scalaka jest programowalna jednostka czasu rzeczywistego (ang. programmable real-time unit, w skrócie PRU).

W PRU zaimplementowana została warstwa MAC standardu EtherCAT. Dzięki temu odpowiada ona bezpośrednio za przetwarzanie przepływających przez nią telegramów, dekodowanie adresów urządzeń oraz wykonywanie zapisanych w datagramie komend. W wyniku takiego rozwiązania, że wszystkie założenia oraz cała funkcjonalność jest realizowana właśnie przy użyciu opisywanego PRU, zamontowany wewnątrz procesor może być zastosowany do realizacji bardziej zaawansowanych oraz złożonych zadań wynikających ze specyfiki konkretnego sprzętu.



Rysunek 18: Budowa mikrokontrolera Sitara AM335x wyposażonego w programowalną jednostkę czasu rzeczywistego.

2.1.7 EtherCAT Technology Group

Standard EtherCAT został opracowany w 2003 roku przez Beckhoff Automation, niemiecką firmę z branży automatyki przemysłowej. Następnie powołano organizację EtherCAT Technology Group (ETG), która zajęła się standaryzacją tego protokołu. Stowarzyszenie to obecnie zajmuje się też organizowaniem szkoleń oraz popularyzacją tego standardu.

Aktualnie w skład ETG wchodzi ponad 2480 firm (dane na dzień 1 września 2013). Najważniejszym członkiem organizacji jest oczywiście firma BECKHOFF Automation. Pozostałe duże i znane firmy wchodzące w jej skład to między innymi: ABB, Brother Industries, BMW Group, Częstochowa University of Technology, Epson, FANUC, Festo, GE Intelligent Platforms, Hitachi, Hochschule Ingolstadt, Mitsubishi, Microchip Technology, Mentor Graphics, Nikon, National Instruments, OLYMPUS, Panasonic, Rzeszów University of Technology, Red Bull Technology, Samsung Electronics, TRW Automotive, Volvo Group, Volkswagen oraz Xilinx.

Jak widać na powyższej liście w skład organizacji wchodzi firmy z bardzo wielu branż, a nawet ośrodki naukowe. Autor pracy wybrał duże i dobrze znane sobie firmy, aby pokazać jak wiele firm interesuje się rozwojem przemysłowych protokołów komunikacyjnych.

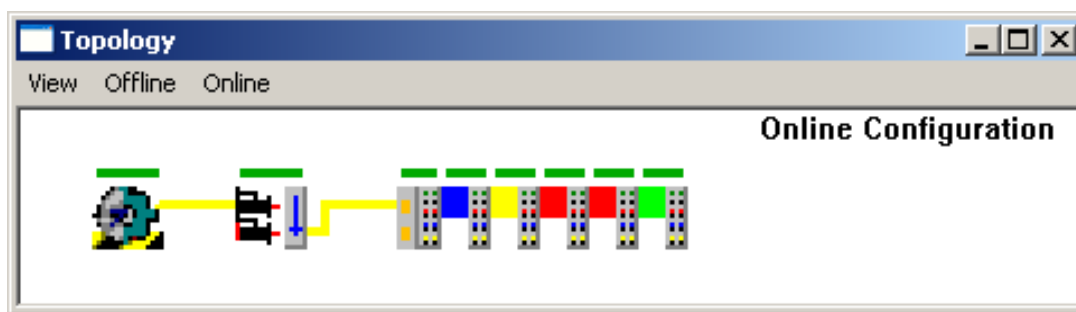
3 Stanowiska badawcze

W niniejszym rozdziale opisane zostaną stanowiska przygotowane do przeprowadzenia kolejnych badań.

3.1 Niezależne podstawowe stanowiska

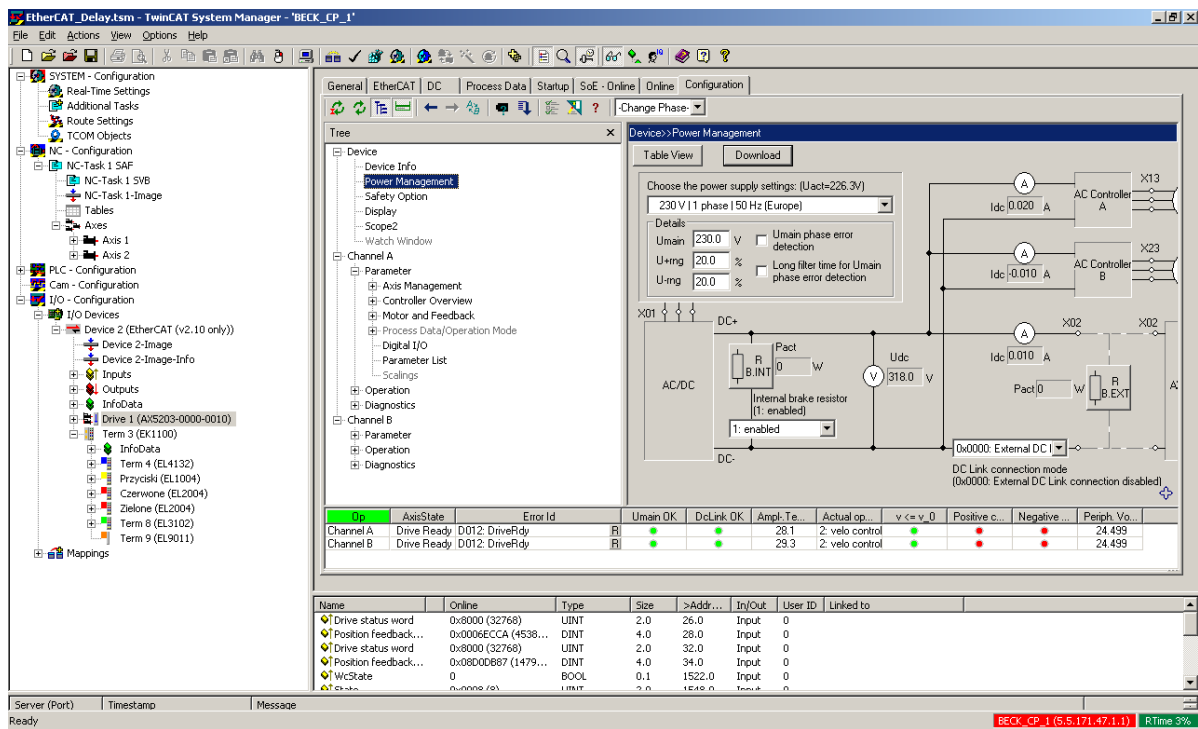
Pierwszym etapem prac mających na celu przygotowanie stanowisk badawczych było skonfigurowanie podstawowej wersji obu stanowisk wraz ze wszystkimi dostępnymi elementami. Stanowiska zostały skonfigurowane i uruchomione w sposób całkowicie nie zależny od siebie.

Konfigurację obu sterowników wraz z modułami przedstawiają Rysunki 20 oraz 21. Tak stworzone konfiguracje pozwoliły uzyskać odpowiednio topologię jak na Rysunku 19 dla stanowiska typu CP oraz jak na Rysunku 22 dla stanowiska typu CX.

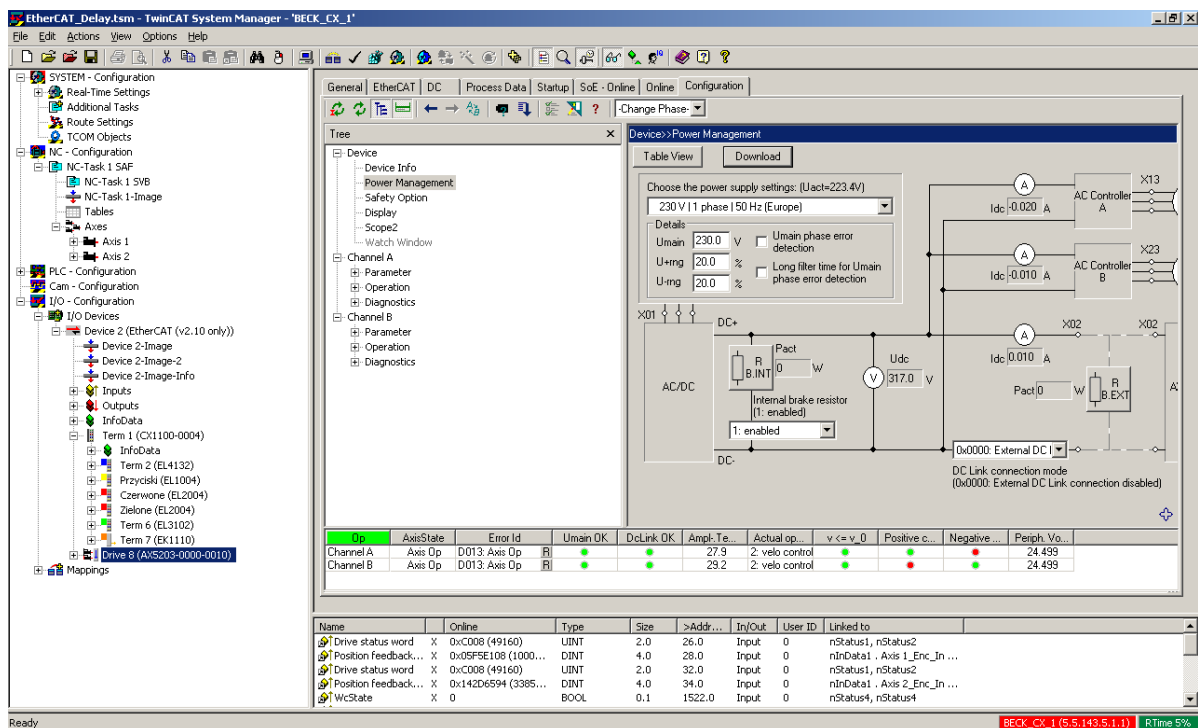


Rysunek 19: Topologia stanowiska typu CP

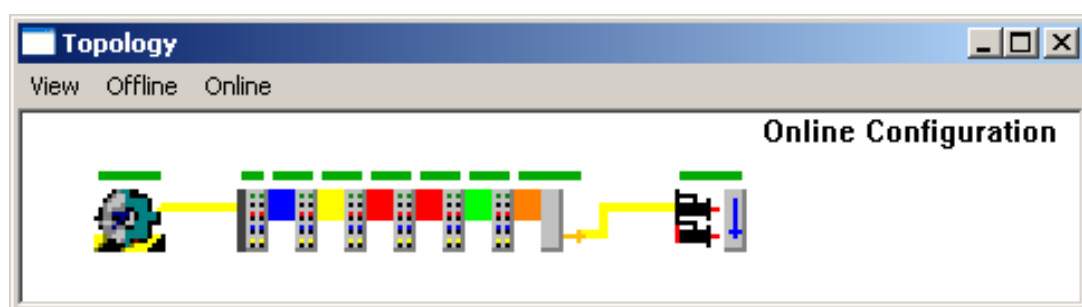
Głównym problemem związanym z uruchomieniem pełnych możliwości stanowisk było prawidłowe skonfigurowanie oraz uruchomienie serwomechanizmów napędzanych przez moduł AX5203.



Rysunek 20: Konfiguracja stanowiska typu CP



Rysunek 21: Konfiguracja stanowiska typu CX



Rysunek 22: Topologia stanowiska typu CX

4 Badania

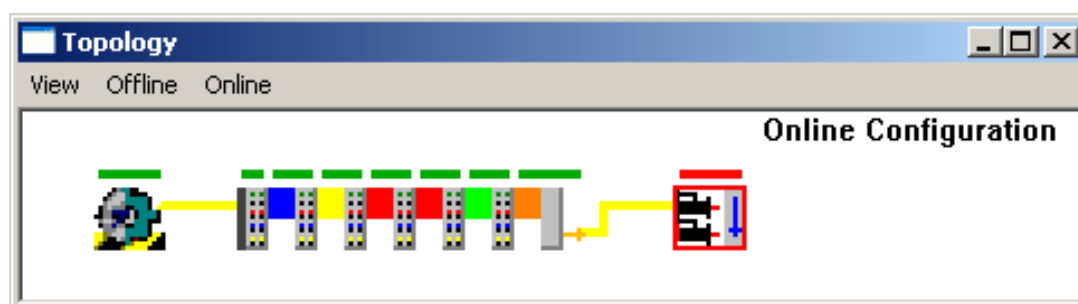
W niniejszym rozdziale opisany został przebieg przeprowadzonych badań oraz analiza ich wyników. W kolejnych podrozdziałach zostaną przedstawione kolejne różne eksperymenty.

4.1 Czas stabilizacji sieci po odłączeniu i podłączeniu dowolnego urządzenia

Badanie miało na celu sprawdzenie czy zgodnie z zapewnieniami producenta faktycznie możliwe jest rozłączanie i podłączanie urządzeń do sieci „w locie” bez większych problemów i bez konieczności żadnej dodatkowej ingerencji. Do przeprowadzenia pomiarów autor odłączał wtyczkę i podłączał ją ponownie do gniazda. Oczywiście metoda ta może mieć minimalny wpływ na dokładność dokonywanych pomiarów, ale zdaniem autora jest on minimalna ze względu na to, że od momentu odłączenia do wykrycia tego faktu przez urządzenie nadzorujące pracę sieci mija pewien czas, który w optymistycznym przypadku może być krótszy niż sam proces ingerencji autora. Aby poprawić dokładność można by zastosować specjalnie stworzone samodzielnie urządzenie co zostanie opisane w perspektywach rozwoju niniejszej pracy.

4.1.1 Pojedyncze urządzenie

Eksperyment miał na celu sprawdzenie jak szybko po odłączeniu i ponownym podłączeniu pojedynczego węzła sieci zaczyna on funkcjonować poprawnie. Zaburzoną pracę sieci przedstawia Rysunek 23, który został wygenerowany przy użyciu środowiska Twin-CAT System Manager i jest to jedna z jego opcji. Jak widać wybrany został węzeł końcowy w topologii.



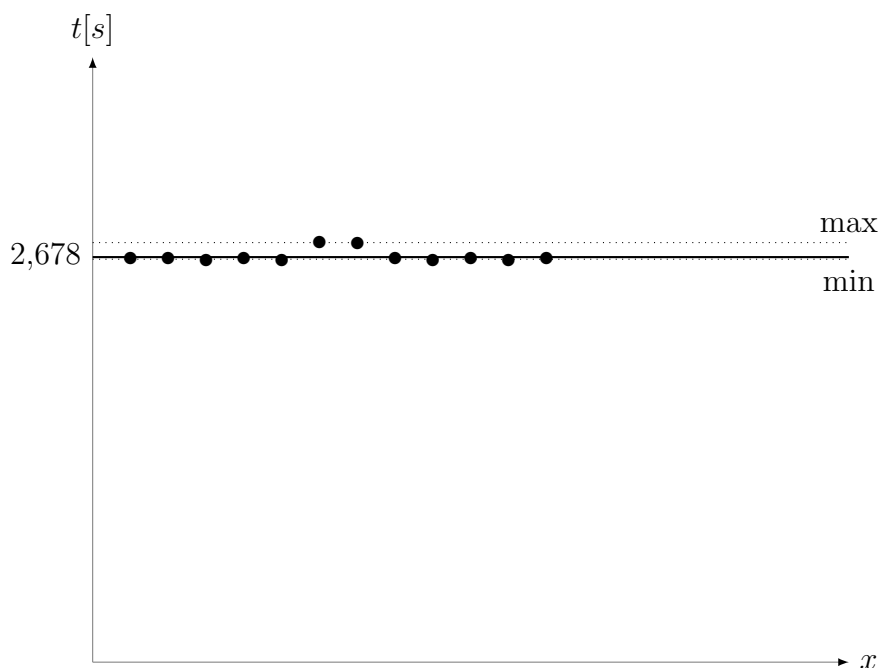
Rysunek 23: Topologia stanowiska z odłączonym jednym węzłem

Wartości zmierzone po ówczesnej obróbce zostały zebrane i przedstawione w Tabeli 3.

Liczba próbek	Wartość średnia	Wartość minimalna	Wartość maksymalna
12	2.678s	2.654s	2.774s

Tablica 3: Wyniki przeprowadzonego badania

Na podstawie zmierzonych wartości oraz po dokonaniu stosownych obliczeń udało się wygenerować wykres przedstawiony na Rysunku 24

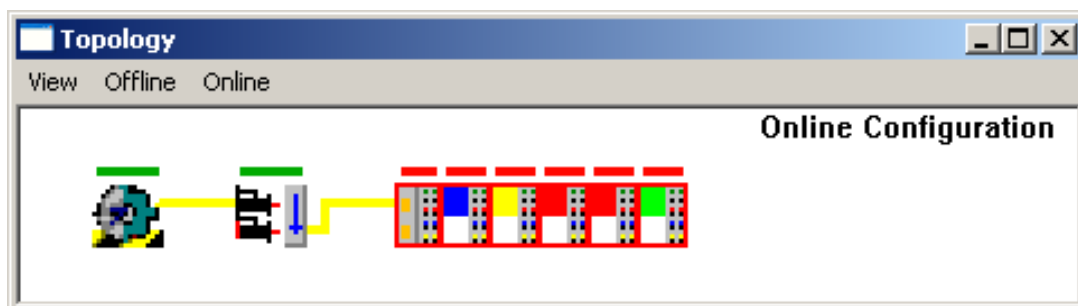


Rysunek 24: Pomiary czasu ponownego podłączenia oraz obliczona wartość średnia

Jak wykazują wyniki przeprowadzonego eksperymentu faktycznie można pozwolić sobie na odłączenie i ponowne podłączenie pojedynczego węzła sieci „w locie”. Czas potrzebny na stabilizację połączenia po jego utracie jest zdaniem autora zadowalający. Analiza uzyskanego wykresu pozwala zauważyć, że różnice pomiędzy kolejnymi iteracjami eksperymentu są bardzo małe i wpływ na nie może mieć zastosowana metoda pomiarowa.

4.1.2 Wyspa z modułami I/O

Eksperyment miał na celu sprawdzenie jak szybko po odłączeniu i ponownym podłączeniu wyspy z dołączonymi kilkoma modułami wejścia/wyjścia zaczynają funkcjonować poprawnie wszystkie węzły. Zaburzoną pracę sieci przedstawia Rysunek 25.



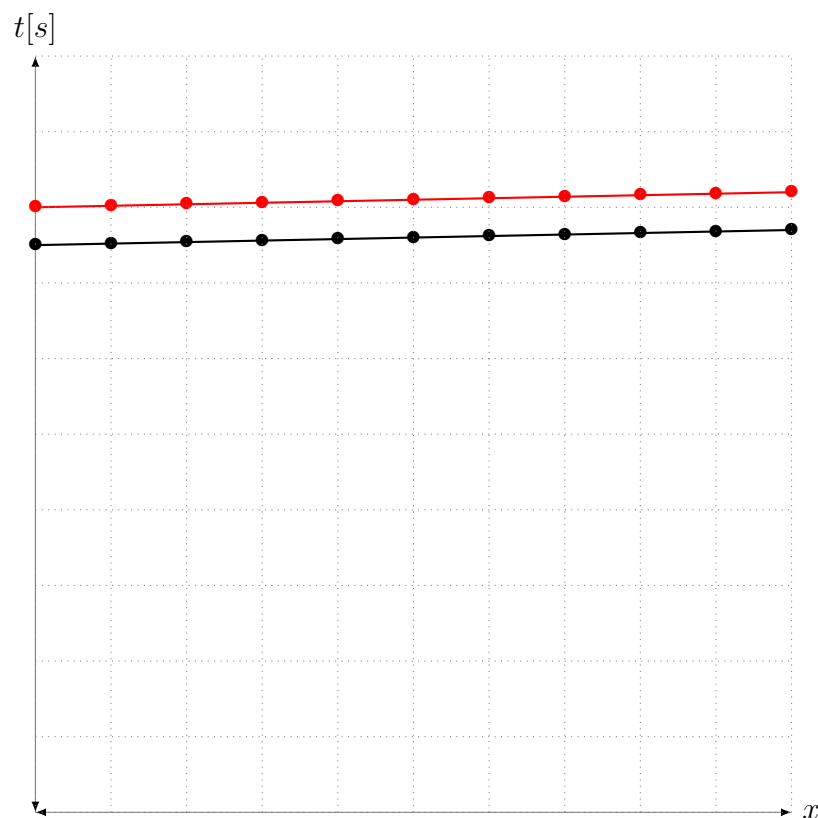
Rysunek 25: Topologia stanowiska z odłączoną wyspą

Wartości zmierzone po ówczesnej obróbce zostały zebrane i przedstawione w Tabeli 4.

Liczba próbek	Wartość średnia	Wartość minimalna	Wartość maksymalna
2	$1.51+5*0.02$	$1.45+5*0.024$	$1.56+5*0.02$

Tablica 4: Wyniki przeprowadzonego badania

Na podstawie zmierzonych wartości oraz po dokonaniu stosownych obliczeń udało się wygenerować wykres przedstawiony na Rysunku 26. Różnymi kolorami są oznaczone kolejne iteracje badania, a kolejne punkty tego samego koloru przedstawiają kolejne węzły.



Rysunek 26: Pomiary czasu ponownego podłączenia oraz obliczona wartość średnia

Jak widać zwiększenie liczby odłączanych i podłączanych ponownie węzłów nie wpłynęło w sposób negatywny na możliwość dokonywania tej operacji w czasie pracy sieci. Ciekawy wydaje się równy czas wynoszący 2ms pomiędzy podłączeniem się pierwszego urządzenia z zestawu, a każdym kolejnym. Tak więc liczba modułów wejścia/wyjścia ma proporcjonalnie mały wpływ na czas potrzebny do ustabilizowania się całego zdalnego zestawu.

4.1.3 Problemy

W czasie XYZ przeprowadzonych testów ku zaskoczeniu aktora wystąpiły sytuacje w których sieć nie powróciła już do prawidłowego działania. Jest to sprzeczne z zapewnieniami twórców standardu i dlatego wszystkie te przypadki zostaną tu zawarte i opisane.

1. Device 2 (EtherCAT (v2.10 only)’: 'INIT to PREOP' failed! Error: 'read slave count'. Communication Error '0x707 (1799)'.

ADS, problem ze stanem urządzenia

4.2 Badania niewykonalne

4.2.1 Zbadanie innych topologii

4.2.2 Zbadanie opóźnień na poziomie transmisji pojedynczych ramek

5 Uruchamianie i testowanie

Rozdział ten zawiera 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 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.

- Problem ze zbyt wysokim napięciem napędu serwomechanizmów:

- Problem z wykryciem jednego z silników:

aa

- Utrata komunikacji z jednym ze sterowników:

Przy pewnej modyfikacji konfiguracji stanowiska typu CX związanej z próbą uruchomienia silników została utracona komunikacja z urządzeniem. Dokładnie przestało ono odpowiadać na poprzednim adresie. Sytuacja była o tyle dziwna, że system na urządzeniu działał całkowicie normalnie. Po podpięciu zewnętrznego monitora okazało się, że w systemie operacyjnym karty sieciowe są skonfigurowane prawidłowo oraz udaje się nawiązać połączenie i uzyskać żądany adres (konfiguracja adresów jest statyczna). Próby wykorzystania programu ping nie dały początkowo

żadnego efektu, ponieważ urządzenie nie odpowiadało. W licznych próbach i pomysłach udało się ustalić, że urządzenie podczas uruchamiania (dokładnie podczas uruchamiania systemu operacyjnego) odpowiada na kilka zapytań (od 4 do 6 w kilku próbach), podobnie w momencie zamykania systemu. To odkrycie zasugerowała autorowi, że coś blokuje, przekonfigurowuje lub wywłaszcza urządzenia. Pierwszy został przeanalizowany autostart systemu Windows lecz okazał się on pusty. Pojawił się pomysł przywrócenia urządzenia do ustawień fabrycznych lecz nie udało się odnaleźć takiej możliwości. Problem udało się rozwiązać uruchamiając w systemie operacyjnym standardowy menadżer plików (w tym przypadku explorer), odnajdując stworzone pliki konfiguracji TwinCAT i usuwając je. (*\HardDisk\TwinCAT\Boot*)

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

6 Wnioski

Protokół EtherCAT jest rozwiązaniem zdecydowanie bardzo nowoczesnym, zaawansowanym oraz pomysłowym. Jest on stosunkowo bardzo młody w porównaniu z innymi znanymi autorowi protokołami komunikacyjnymi stosowanymi powszechnie w przemyśle takimi jak:

- Modbus – wprowadzony na rynek przez firmę Modicon w 1979 roku,
- Profibus (**P**rocess **F**ield **B**us) – wprowadzony przez Niemiecki departament edukacji i badań (niem. Bundesministerium für Bildung und Forschung, w skrócie BMBF) w roku 1989,
- CAN (Controller Area Network) – wprowadzony przez Robert Bosch GmbH w roku 1989
- EGD (Ethernet Global Data) – wprowadzony przez firmy GE Fanuc Automation oraz GE Drive Systems w roku 1998

Fakt, że protokół jest tak młody wpływa bardzo pozytywnie na liczbę materiałów dostępnych w internecie. Niestety jest też drobna wada, a mianowicie informacje są często nie kompletne i trzeba doszukiwać się pożądaných szczegółów w różnych miejscach. Na szczęście po zapoznaniu się z kilkoma stronami oraz artykułami można uzyskać zupełnie przyzwoity zakres wiedzy o tym protokole.

Pozwala bardzo dobrze wykorzystać wzrastające moce obliczeniowe sterowników PLC, przy zachowaniu wszelkich rygorów czasowych. Zdecydowanie ogromny wpływ na tak szybki rozwój ma fakt, że technologia jest bardzo otwarta i postawiona na współpracę wszystkich zainteresowanych rozwojem stron. Olbrzymim plusem jest fakt wykorzystania standardowej struktury Ethernetu co upraszcza proces integracji w obrębie jednego systemu obu standardów.

Badany protokół jest bardzo rozbudowany i skomplikowany. Jego zrozumienie wymaga czasu oraz zapoznania się z dużą ilością stosownych dokumentacji. Mnogość możliwości i konfiguracji będąca niewątpliwą zaletą tego standardu może na początku przerażać, ale niestety takie są koszty tak rozbudowanej i zaawansowanej funkcjonalności. Autor sam był początkowo zaskoczony tak wysokim stopniem zaawansowania protokołu przemysłowego, w porównaniu przykładowo do poznanego dobrze wcześniej protokołu Modbus.

Prawdopodobnie po zakończeniu projektu posłuży on jako podstawa do stworzenia i przeprowadzenia ćwiczeń laboratoryjnych prowadzonych w ramach działalności dydaktycznej prowadzonej przez pracowników Zespołu Przemysłowych Zastosowań Informatyki. Można z wykorzystaniem dołączonych konfiguracji, projektów oraz dokumentacji przygotować proste laboratorium z podstaw programowania sterowników firmy Beckhoff, bardziej zaawansowane programowanie ze sterowaniem silnikami lub najbardziej zaawansowane laboratorium związane z działaniem protokołu EtherCAT.

6.1 Perspektywy dalszych badań

Temat zdecydowanie nadaje się do pogłębiania i prowadzenia dalszych badań. Autor dopuszcza taką możliwość w ramach prac badawczych w toku swoich studiów doktoranckich. Zarówno same sterowniki firmy Beckhoff oparte o koncepcję „soft PLC” jak i sam

badany protokół EtherCAT są na tyle rozbudowane i rozwojowe, że zawsze znajdzie się jakiś aspekt do przeanalizowania od strony teoretycznej i eksperymentalnej. W momencie powstawania niniejszej pracy trzy bardzo interesujące kwestie wydają się autorowi ciekawą postawą do przeprowadzenia badań.

Po pierwsze przetestowanie elementów sieci EtherCAT pochodzących od innych producentów. Jak już zostało to opisane w rozdziale zawierającym analizę tematu węzły EtherCAT mogą być realizowane na wiele sposobów zależnie od wymagań funkcjonalnych. Autor bardzo chętnie podjąłby badania mające na celu zaprojektowanie własnego układu cyfrowych wejść/wyjść i porównanie takiego rozwiązania z dostępnymi na rynku, aby wyznaczyć stosunek jakości do kosztów. W dalszym etapie zdecydowanie warto by przyjrzeć się bardziej elastycznym i rozbudowanym rozwiązaniom pozwalającym na realizację bardziej zaawansowanych węzłów. Interesująca wydaje się możliwość konstruowania zupełnie od podstaw węzłów o dowolnej funkcjonalności. Dzięki temu można by spróbować zbudować węzeł eksperymentalny mający za zadanie tylko monitorowanie przepływających przez niego ramek i ich ewentualne gromadzenie. Być może taka analiza krążących w sieci ramek pozwoliłaby wykryć jakieś interesujące właściwości lub zachowania charakterystyczne dla omawianego protokołu.

Zdaniem autora najbardziej interesujące pod względem badawczym są właśnie jednostki wyposażone w procesor dedykowany do realizacji bardziej złożonych zadań niezależnie od działania protokołu. Ciekawym rozwiązaniem jest oferowany przez firmę Texas Instruments mikrokontroler z rdzeniem ARM Cortex-A8 z rodziny Sitara AM335x. Być może rozwiązania firm wchodzących w skład ETG okażą się lepsze lub gorsze pod względem szybkości w porównaniu do pierwszych twórców protokołu.

Drugim ciekawym rozwiązaniem i pomysłem na które autor natrafił w sieci w czasie analizy tematu, rozwiązywania problemów oraz pisania niniejszej pracy jest EtherLab. Jest to technologia łącząca sprzęt i oprogramowanie w celach testowych oraz do sterowania procesów przemysłowych. Jest to niejako technika zbudowana z dobrze znanych i niezawodnych elementów. EtherLab pracuje jako działający w czasie rzeczywistym moduł jądra otwartego systemu Linux, który komunikuje się z urządzeniami peryferyjnymi poprzez protokół EtherCAT. Rozwiązanie jest całkowicie darmowe i otwarte co na pewno jest jego olbrzymią zaletą. Można zdecydować się na pobranie sobie wszystkich komponentów i ich samodzielne uruchomienie lub zakup gotowego preinstalowanego zestawu startowego bezpośrednio od twórców. Oprogramowanie całego zestawu może zostać wygenerowane przy użyciu Simulinka/RTW lub napisane ręcznie w C. Następnie tak przygotowane jest uruchamiane w środowisku kontrolującym proces (jądro Linuksa oraz moduł czasu rzeczywistego) komunikującym się z „obiektem przemysłowym” poprzez EtherCAT. Dodatkowo można rozszerzyć możliwości całego zestawu poprzez Ethernet TCP/IP dołączając interfejs użytkownika (ang. Frontend) w wersji dla Linuksa lub Windowsa albo jeden z innych dodatkowych serwisów. Przykładowe serwisy to:

- Raportowanie poprzez SMS,
- Zdalne usługi: Internet, ISDN, DSL,
- Usługi sieciowe: Web, DHCP, Drukowanie,
- Logowanie danych (ang. data logging).

Jeżeli autor będzie miał taką możliwość to na pewno chętnie przyjrzy się tej koncepcji ze względu na swoją sympatię do systemu Linux oraz wszystkich rozwiązań go wykorzystujących. Ciekawe wydają się badania wydajności takiego rozwiązania oraz porównanie ich z drogimi rozwiązaniami komercyjnymi.

Trzecim w kolejności pomysłem, który powstał na etapie realizacji pracy dyplomowej oraz zainspirowany po części realizacją przez kolegów z roku projektu z przedmiotu Projektowanie Przemysłowych Systemów Komputerowych mającego na celu umożliwienie podłączenia do sterownika Beckhoff bazy danych. Studenci zaproponowali i zrealizowali rozwiązanie bazujące na RS-232. Zdaniem autora protokół EtherCAT nadawałby się do tego celu bardzo dobrze głównie ze względu na prędkość działania, ale również ze względu na możliwość bezpośredniej modyfikacji danych zapisywanych do bazy przez dowolny węzeł sieci. Pomysł nie jest w swoich podstawach badawczy, a bardziej praktyczny. Autor dostrzega jednak możliwość przygotowania stanowiska eksperymentalnego na którym można by przeprowadzić badania wydajności takiego rozwiązania.

Kolejnym pomysłem jest przebadanie protokołu z wykorzystaniem stworzonego na naszym wydziale i opisanego w literaturze analizatora sieci czasu rzeczywistego opartych na Ethernetie [17]. Można by to urządzenie zastosować do zewnętrznej analizy protokołu EtherCAT, a nie wewnątrz działającego systemu jak to miało miejsce w przypadku badań opisanych niniejszym dokumentem. Możliwe byłoby porównanie opóźnień wprowadzanych przez węzły typu EtherCAT z tymi wprowadzonymi przez omawiany analizator.

Następnym pomysłem do ewentualnego przebadania w przyszłości znalezionym w sieci jest możliwość uruchomienia kontrolera urządzenia podrzędnego EtherCAT na układzie FPGA firmy Xilinx lub Altera [18, 19]. Rozwiązanie tego typu można swobodnie przetestować i zaimplementować dzięki dostarczonym przez firmę Beckhoff zestawom ewaluacyjnym wyposażonym w wymienione układy. Przykładowe dwa zestawy to EL9830 wyposażony w układ Altery (EP3C25) oraz EL9840 z Xilinxem (XC3S1200).

Teoretycznie dzięki wykorzystaniu w protokole EtherCAT standardu Ethernet możliwe jest sterowanie urządzeniami z poziomu zwykłego komputera klasy PC z wykorzystaniem modelu TCP/IP. Ciekawym wydaje się na ile skomplikowane okaże się zbudowanie ramki do celów typowo testowych. Autor ma świadomość, że rozwiązanie takie nie nadaje się raczej do profesjonalnego stosowania w przemyśle, ale wydaje się być interesującą alternatywą do testowania działania urządzeń wykonawczych bez konieczności zaprzęgnięcia sterownika PLC. Dodatkowym interesującym aspektem wykorzystania takiego rozwiązania wydaje się lepsze poznanie i zrozumienie zasad działania protokołu. Autor w czasie realizacji jednego z projektów semestralnych prowadzonego w ramach przedmiotu Projektowanie Przemysłowych Systemów Komputerowych i swojej działalności w Kole Naukowym Przemysłowych Zastosowań Informatyki „Industrum” implementował samodzielnie oprogramowanie do komunikacji w sieci ELAN. Działanie to pozwoliło zdecydowanie lepiej zrozumieć zasady działania tego dość prostego, ale rozbudowanego interfejsu.

Tak jak już zostało napisane w rozdziale opisującym badania do zwiększenia precyzji badań mówiących o czasie ponownego podłączenia odłączonego i podłączonego węzła sieci przydałoby się zaprojektować i wykonać urządzenie pozwalające w precyzyjny sposób wpływać na ciągłość linii transmisyjnej. Urządzenie takie można by oprzeć o układ mikroprocesorowy do kontroli oraz tranzystory do przerywania ciągłości magistrali. Pozwoliło by to w bardzo precyzyjny sposób określić czas ponownego podłączenia z pominięciem błędu pomiarowego wprowadzanego przez człowieka. Takie urządzenie pozwoliłoby

również przebadac inne możliwe scenariusze. Przykładowo można by spróbować wygenerować przerwę trwającą równowartość czasu potrzebnego na transmisję pojedynczego znaku (ewentualnie kilku znaków). Pozwoliłoby to zaobserwować zachowanie protokołu w momencie wystąpienia błędów transmisji. Ewentualnie wywołując odpowiednio dużo błędów można sprawdzić stabilność pracy protokołu w sytuacji problemów transmisyjnych. Dzięki stworzeniu omówionego urządzenia dałoby się również przebadac inne protokoły komunikacyjne.

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] Andrzej Gawryluk: *„EtherCAT – to nie takie trudne. Ethernet jako sieć real-time”*, Elektronika Praktyczna, 2/2010.
- [7] Maneesh Soni: *„EtherCAT w praktyce. Zastosowanie mikrokontrolera Sitara do implementacji EtherCAT.”*, Elektronika Praktyczna, 7/2012.
- [8] Michał Gosk: *„Szybkość, niezawodność, doskonała synchronizacja. EtherCAT - system przyszłości.”*, Magazyn Sensor, 1/2013, kwiecień.
- [9] Monika Jaworowska: *„Determinizm czasowy transmisji w Ethernetie przemysłowym.”*, Portal branżowy dla Automatyków – AutomatykaB2B, [online], 8 listopada 2012, [dostęp 28 sierpnia 2013]. Dostępny w Internecie:
<http://automatykab2b.pl/tematmiesiaca/5061-determinizm-czasowy-transmisji-w-ethernecie-przemyslowym>
- [10] Monika Jaworowska: *„Sieci przemysłowe w standardzie EtherCat.”*, Portal branżowy dla Automatyków – AutomatykaB2B, [online], 20 czerwca 2012, [dostęp 26 sierpnia 2013]. Dostępny w Internecie:
<http://elektronikab2b.pl/technika/16954-sieci-przemyslowe-w-standardzie-ethercat>
- [11] Beckhoff Automation: *„EtherCAT – przyszłość sieci przemysłowych.”*, Portal branżowy dla Automatyków – AutomatykaB2B, [online], 28 czerwca 2011, [dostęp 22 sierpnia 2013]. Dostępny w Internecie:
<http://automatykab2b.pl/prezentacja-artykul/3898-ethercat—przyszlosc-sieci-przemyslowych>
- [12] Beckhoff Automation: *„EtherCAT i światłowody – szybko i skutecznie.”*, Portal branżowy dla Automatyków – AutomatykaB2B, [online], 2 listopada 2011, [dostęp 22 sierpnia 2013]. Dostępny w Internecie:
<http://automatykab2b.pl/prezentacja-artykul/4159-ethercat-i-swiatlowody—szybko-i-skutecznie>

- [13] Beckhoff Automation: „*Nowa, szybka sieć EtherCAT.*”, Portal branżowy dla Automatyków – AutomatykaB2B, [online], 2 listopada 2011, [dostęp 22 sierpnia 2013]. Dostępny w Internecie:
<http://automatykab2b.pl/prezentacja-artykul/4159-ethercat-i-swiatlowody—szybko-i-skutecznie>
- [14] Real Time Automation: „*EtherCAT Open Real-Time Ethernet Network.*”, Portal Real Time Automation, [online], 2009, [dostęp 2 września 2013]. Dostępny w Internecie:
<http://www.rtaautomation.com/ethercat/>
- [15] EtherCAT Technology Group: „*EtherCAT – the Ethernet Fieldbus.*”, Oficjalna strona internetowa EtherCAT Technology Group, [online], [dostęp 30 sierpnia 2013]. Dostępny w Internecie:
<http://www.ethercat.org/en/technology.html>
- [16] Krzysztof Oprzędkiewicz: „*Realizacja predyktora Smitha na platformie sprzętowo-programowej „soft PLC” bazującej na komputerze klasy PC*”, Automatyka / Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie, 2006, t. 10, z. 2, s. 177–186, ISSN 1429-3447.
- [17] Rafał Cupek, Piotr Piękoś, Marcin Poczobut, Adam Ziebinski: „*FPGA based „Intelligent Tap” device for real-time Ethernet network monitoring*”, Computer Networks, Communications in Computer and Information Science, Springer-Verlag Berlin Heidelberg, 2010, t. 79, s. 58-66, ISBN 978-3-642-13860-7.
- [18] Dokumentacja producenta: „*ET1815 / ET1817 EtherCAT Slave Controller IP Core for Xilinx FPGAs IP Core Release 2.02a*”, Wersja 2.2.1, 1 wrzesień 2008.
- [19] Dokumentacja producenta: „*ET1810 / ET1812 EtherCAT Slave Controller IP Core for Altera FPGAs IP Core Release 2.4.0*”, Wersja 1.0 15 marca 2011.
- [20] Dokumentacja producenta: „*AX5000 – Motion control for high dynamic positioning*”, 29 października 2007
- [21] IEEE 1588-2002: IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems

8 Spis rysunków, tablic i kodów źródłowych

8.1 Spis rysunków

Rysunek 1: Schemat działania systemu czasu rzeczywistego.	4
Rysunek 2: Funkcja zysku dla systemów czasu rzeczywistego	5
Rysunek 3: Schemat stanowiska typu CP	9
Rysunek 4: Schemat stanowiska typu CX	10
Rysunek 5: Współczynnik wykorzystania kanału transmisyjnego w Ethernecie (dwa pierwsze wykresy od lewej strony) i EtherCAT	16
Rysunek 6: Przykład transmisja małej ilości danych (4 bajty) zwykłym Ethernetem.	16
Rysunek 7: Ramka w transmisji EtherCAT i jej podział na datagramy	17
Rysunek 8: Ramka w transmisji EtherCAT z uwzględnieniem UDP i IP . . .	17
Rysunek 9: Ramka w transmisji EtherCAT z uwzględnieniem TCP i IP . . .	18
Rysunek 10: Budowa datagramu	18
Rysunek 11: Budowa nagłówka datagramu	19
Rysunek 12: Budowa nagłówka EtherCAT	20
Rysunek 13: Przykładowa topologia sieci	20
Rysunek 14: Schemat pracy zegarów rozproszonych	22
Rysunek 15: Budowa węzła EtherCAT z wykorzystaniem pojedynczego układu FPGA lub ASIC.	23
Rysunek 16: Budowa węzła z wykorzystaniem układu ASIC/FPGA EtherCAT z dołączonym zewnętrznym procesorem.	23
Rysunek 17: Budowa układu FPGA z wbudowanym procesorem.	23
Rysunek 18: Budowa mikrokontrolera Sitara AM335x wyposażonego w programowalną jednostkę czasu rzeczywistego.	24
Rysunek 19: Topologia stanowiska typu CP	26
Rysunek 20: Konfiguracja stanowiska typu CP	27
Rysunek 21: Konfiguracja stanowiska typu CX	27
Rysunek 22: Topologia stanowiska typu CX	28
Rysunek 23: Topologia stanowiska z odłączonym jednym węzłem	29
Rysunek 24: Pomiar czasu ponownego podłączenia oraz obliczona wartość średnia	30
Rysunek 25: Topologia stanowiska z odłączoną wyspą	31
Rysunek 26: Pomiar czasu ponownego podłączenia oraz obliczona wartość średnia	32

8.2 Spis tablic

Tablica 1: Systemy czasu rzeczywistego	8
Tablica 2: Dostępne stanowiska laboratoryjne	11
Tablica 3: Wyniki przeprowadzonego badania	30
Tablica 4: Wyniki przeprowadzonego badania	31

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 !?
 - Pliki zawierające wyniki przeprowadzonych badań,
 - L^AT_EX-owe, TikZ-owe oraz PGF-owe pliki źródłowe pracy magisterskiej.