Społeczeństwo informacyjne

Projekt

„Protokoły komunikacyjne w przemyśle”

Bazan Michał – 163881

Spis treści

1. Wstęp

Projekt "Protokoły komunikacyjne w przemyśle" ma na celu przedstawienie i zrozumienie różnych protokołów komunikacyjnych stosowanych w systemach przemysłowych oraz przeprowadzenie testów komunikacyjnych przy użyciu dwóch komputerów. Projekt będzie obejmować aspekty teoretyczne i praktyczne, a także zagadnienia związane z bezpieczeństwem komunikacji.

1.1. Wprowadzenie do tematu komunikacji w systemach przemysłowych

W dzisiejszych czasach systemy przemysłowe są coraz bardziej złożone i wymagają niezawodnej komunikacji między różnymi urządzeniami i systemami. Komunikacja w systemach przemysłowych odnosi się do wymiany informacji między maszynami, czujnikami, sterownikami i innymi komponentami automatyki. Dzięki skutecznej komunikacji możliwe jest monitorowanie, kontrolowanie i optymalizowanie procesów produkcyjnych oraz zarządzanie operacjami w czasie rzeczywistym.

Protokoły komunikacyjne odgrywają kluczową rolę w zapewnieniu skutecznej wymiany danych. Protokoły te definiują zasady, zgodnie z którymi urządzenia mogą komunikować się ze sobą, określając format danych, metody synchronizacji oraz mechanizmy transmisji. W systemach przemysłowych stosowane są różne protokoły, w tym Modbus, Profibus, CAN, Ethernet/IP oraz OPC UA, które różnią się pod względem zastosowania, topologii sieci oraz poziomu złożoności.

1.2. Znaczenie niezawodnej i bezpiecznej komunikacji w przemyśle

Niezawodność komunikacji w systemach przemysłowych jest kluczowa dla zapewnienia ciągłości procesów produkcyjnych i minimalizacji ryzyka awarii. W przemyśle, gdzie każda minuta przestoju może wiązać się z dużymi stratami finansowymi, niezawodna wymiana informacji między urządzeniami pozwala na szybkie reagowanie na zmiany warunków produkcji oraz natychmiastowe podejmowanie działań naprawczych w przypadku wykrycia problemów.

Bezpieczna komunikacja jest równie ważna, zwłaszcza w kontekście rosnącej liczby cyber zagrożeń. Systemy przemysłowe są coraz częściej celem ataków hakerskich, które mogą prowadzić do kradzieży danych, uszkodzenia sprzętu, a nawet do zakłócenia całego procesu produkcyjnego. Dlatego też wprowadzenie odpowiednich środków bezpieczeństwa, takich jak szyfrowanie danych, uwierzytelnianie użytkowników oraz monitorowanie sieci, jest niezbędne dla ochrony systemów przemysłowych przed nieautoryzowanym dostępem i innymi zagrożeniami.

Podsumowując, niezawodna i bezpieczna komunikacja w systemach przemysłowych jest fundamentem efektywnego i bezpiecznego zarządzania procesami produkcyjnymi. W dalszych rozdziałach tego projektu zostaną szczegółowo omówione wybrane protokoły komunikacyjne, ich praktyczne zastosowania oraz metody zapewnienia bezpieczeństwa komunikacji w środowisku przemysłowym.

2. Przegląd teoretyczny wybranych protokołów komunikacyjnych

2.1. Modbus TCP/IP

Modbus [1] jest jednym z najstarszych i najbardziej powszechnie stosowanych protokołów komunikacyjnych w automatyce przemysłowej. Został opracowany w 1979 roku przez firmę Modicon (obecnie część Schneider Electric) dla sterowników PLC. Oryginalnie Modbus wykorzystywał transmisję szeregową (RS-232/RS-485), jednak wraz z rozwojem technologii sieciowych, został przystosowany do pracy w sieciach Ethernet, dając początek Modbus TCP/IP.

Modbus TCP/IP jest wersją Modbus opartą na protokole TCP/IP, co pozwala na wykorzystanie standardowej infrastruktury sieciowej Ethernet do komunikacji między urządzeniami. Główne elementy architektury Modbus TCP/IP to:

- **Master/Slave**: Modbus TCP/IP działa w modelu klient-serwer, gdzie jeden z urządzeń pełni rolę klienta (master), wysyłającego zapytania, a inne urządzenia są serwerami (slave), które odpowiadają na te zapytania.
- **Adresacja**: Każde urządzenie w sieci Modbus TCP/IP ma unikalny adres IP, a każdy serwer może mieć do 247 urządzeń podrzędnych (adresowanych od 1 do 247).
- **Ramki danych**: Dane są przesyłane w formie ramek, które składają się z adresu urządzenia, kodu funkcji, danych oraz sumy kontrolnej.

Modbus TCP/IP jest szeroko stosowany w różnych sektorach przemysłu, w tym w automatyce budynkowej, systemach HVAC, systemach monitorowania i kontroli procesów przemysłowych, oraz w energetyce. Jego prostota i niezawodność sprawiają, że jest preferowany w wielu aplikacjach, gdzie wymagana jest stabilna i łatwa do implementacji komunikacja.

Modbus TCP/IP obsługuje różne funkcje, w tym:

- **Czytanie i zapisywanie rejestrów**: Umożliwia odczyt i zapis danych w rejestrach urządzeń.
- **Diagnostyka**: Umożliwia wykonywanie operacji diagnostycznych na urządzeniach.
- **Zarządzanie urządzeniami**: Umożliwia zarządzanie i konfigurację urządzeń w sieci.

Zalety*:*

- Prosty i łatwy do implementacji.
- Wysoka kompatybilność z różnymi urządzeniami.
- Wykorzystuje istniejącą infrastrukturę sieciową Ethernet.

Wady*:*

- Ograniczone funkcje w porównaniu z bardziej zaawansowanymi protokołami.
- Brak wbudowanych mechanizmów bezpieczeństwa.

Ramka modbus w trybie ASCII

:	Adres		Kod funkcji		Dane			Suma kontrolna		CR	LF
						...					

Bajty wysyłane są szesnastkowo (po dwa znaki ASCII) a odstępy pomiędzy kolejnymi znakami ramki są mniejsze niż 1 sekunda.

Ramka komunikacyjna w trybie RTU

Adres		Kod funkcji		Dane			Suma kontrolna	
					...			

Bajty wysyłane są jako znaki ośmiobitowe a każda ramka jest poprzedzona odstępem większym niż 3.5T (gdzie T oznacza czas transmisji jednego znaku).

Znaczenie bajtów

- **adres**:

0 – adres rozgłoszeniowy

1 – 247 – adres jednostki server

- kod funkcji:

1 $01 odczyt wyjść bitowych

2 $02 odczyt wejść bitowych

3 $03 odczyt n rejestrów

4 $04 odczyt n rejestrów wejściowych

5 $05 zapis 1 bitu

6 $06 zapis 1 rejestru

7 $07 odczyt statusu

8 $08 test diagnostyczny

15 $0F zapis n bitów

16 $10 zapis n rejestrów

17 $11 identyfikacja urządzenia server

128 – 255 $80–$FF zarezerwowane na odpowiedzi błędne

Obsługa błędów komunikacji:

- odesłanie przez server ramki z kodem błędu:

01 – niedozwolona funkcja
02 – niedozwolony numer rejestru
03 – niedozwolona wartość danej
04 – uszkodzenie w przyłączonym urządzeniu
05 – potwierdzenie pozytywne
06 – brak gotowości, komunikat usunięty
07 – potwierdzenie negatywne
08 – błąd parzystości pamięci

- przy przekroczeniu czasu oczekiwania na odpowiedź serwer nie odsyła odpowiedzi.

2.2. Ethernet/IP

Ethernet/IP [2] (Ethernet Industrial Protocol) został opracowany przez Rockwell Automation i jest zarządzany przez organizację ODVA (Open DeviceNet Vendors Association). Ethernet/IP wykorzystuje standardowy protokół Ethernet i warstwę transportową TCP/IP, integrując je z przemysłowym protokołem aplikacyjnym CIP (Common Industrial Protocol), co pozwala na jego szerokie zastosowanie w przemysłowych systemach automatyki.

Ethernet/IP działa w modelu producent-konsument, co oznacza, że dane mogą być przesyłane od jednego nadawcy (producenta) do wielu odbiorców (konsumentów). Kluczowe elementy architektury Ethernet/IP to:

- **CIP (Common Industrial Protocol)**: CIP zapewnia jednolity model danych i mechanizmy komunikacji, które mogą być używane w różnych sieciach przemysłowych.
- **TCP/IP i UDP**: Ethernet/IP wykorzystuje zarówno TCP/IP do niezawodnych transmisji oraz UDP do szybkich i mniej wymagających czasowo transmisji danych.
- **Enkapsulacja**: Dane CIP są enkapsulowane w standardowe ramki Ethernet [3], co pozwala na wykorzystanie standardowych urządzeń sieciowych, takich jak switche i routery.

Ethernet/IP jest szeroko stosowany w automatyce przemysłowej, w tym w systemach sterowania procesami, systemach monitorowania, systemach transportowych, a także w robotyce i automatyzacji produkcji. Jego zdolność do obsługi dużych ilości danych w czasie rzeczywistym sprawia, że jest idealnym rozwiązaniem dla aplikacji wymagających wysokiej wydajności.

Ethernet/IP oferuje szereg funkcji, które obejmują:

- **Kontrola cykliczna (I/O)**: Umożliwia szybką i niezawodną wymianę danych wejścia/wyjścia między urządzeniami.
- **Messaging (Explicit Messaging)**: Umożliwia wysyłanie specjalnych komunikatów konfiguracyjnych, diagnostycznych i sterujących.
- **Bezpieczeństwo**: Ethernet/IP obsługuje funkcje bezpieczeństwa, takie jak szyfrowanie danych i uwierzytelnianie, co jest istotne w nowoczesnych systemach przemysłowych.

Zalety*:*

- Wysoka przepustowość i niskie opóźnienia.
- Możliwość obsługi dużej liczby urządzeń w sieci.
- Zintegrowane funkcje bezpieczeństwa.

Wady*:*

- Złożoność implementacji w porównaniu z prostszymi protokołami.
- Wyższe wymagania sprzętowe i kosztowe.

3. Praktyczne zastosowanie protokołów

W tej części projektu zostaną omówione kroki konieczne do skonfigurowania komunikacji Modbus TCP/IP oraz Ethernet/IP pomiędzy dwoma komputerami: komputerem fizycznym z systemem Linux oraz maszyną wirtualną również z systemem Linux.

3.1. Konfiguracja komunikacji Modbus TCP/IP pomiędzy dwoma komputerami

```
sudo apt update -y
sudo apt install python3 python3-pip -y
pip3 install pymodbus
```

Listing 3.1.1 Instalacja python3 i biblioteki pymodbus

```
import asyncio
import logging
import sys
import pymodbus.client as modbus_async_client
from pymodbus.exceptions import ModbusException

async def run_modbus_client():
    client = modbus_async_client.ModbusTcpClient('127.0.0.1', port=5020)

    try:
        client.connect()

        rr_coils = client.read_coils(0, 10)
        rr_registers = client.read_holding_registers(0, 10)

        print("Odczytane I/O: ", rr_coils.bits)
        print("Odczytane rejestry przed ustawieniem: ", rr_registers.registers)

        write_registers = [i for i in range(10)]
        client.write_registers(0, write_registers)
        rr_registers_after = client.read_holding_registers(0, 10)
        print("Odczytane rejestry po ustawieniu: ", rr_registers_after.registers)

    except ModbusException as e:
        print("Błąd Modbus:", e)

    finally:
        client.close()

async def main():
    await run_modbus_client()

if __name__ == "__main__":
    asyncio.run(main())
```

Listing 3.1.2. Klient modbus

Klient Modbus został zaprojektowany do komunikacji z serwerem Modbus za pomocą protokołu TCP/IP. Skrypt klienta napisany jest w języku Python i korzysta z biblioteki `pymodbus`. Po nawiązaniu połączenia z serwerem, klient odczytuje dane wejścia/wyjścia (I/O) oraz rejestry, a

następnie wykonuje operację zapisu na wybranych rejestrach. Każda operacja jest obsługiwana asynchronicznie, co pozwala na efektywne zarządzanie komunikacją Modbus. Po zakończeniu operacji, klient zamyka połączenie, co zapewnia prawidłowe zakończenie sesji komunikacyjnej.

```
import asyncio
import logging
import sys

from pymodbus import __version__ as pymodbus_version
from pymodbus.datastore import (
    ModbusSequentialDataBlock,
    ModbusServerContext,
    ModbusSlaveContext,
    ModbusSparseDataBlock,
)
from pymodbus.device import ModbusDeviceIdentification
from pymodbus.server import (
    StartAsyncSerialServer,
    StartAsyncTcpServer,
    StartAsyncTlsServer,
    StartAsyncUdpServer,
)

block = ModbusSequentialDataBlock(0x00, [0] * 100)
store = ModbusSlaveContext(di=block, co=block, hr=block, ir=block)
context = ModbusServerContext(slaves=store, single=True)

async def run_server():
    server = await StartAsyncTcpServer(context, address=("0.0.0.0", 5020))
    await server.serve_forever()

asyncio.run(run_server())
```

Listing 3.1.3. Serwer modbus

Serwer Modbus został stworzony w celu obsługi żądań klientów Modbus TCP/IP. Skrypt serwera został również napisany w języku Python, wykorzystując bibliotekę `pymodbus`. Serwer został skonfigurowany do obsługi komunikacji na porcie 5020. Po uruchomieniu, serwer nasłuchuje na wszystkich interfejsach sieciowych (`0.0.0.0`), co umożliwia klientom nawiązanie połączenia z dowolnego adresu IP w sieci lokalnej. Serwer Modbus obsługuje komunikację asynchroniczną, co pozwala na efektywne przetwarzanie wielu żądań klientów jednocześnie.

```
michael@debian:~/spoleczenstwo_informacyjne_projekt/scripts/modbus$ python3
modbus_client.py

Odczytane I/O:  [False, False, False, False, False, False, False, False, False, False,
False, False, False, False, False, False]
Odczytane rejestry przed ustawieniem:  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
Odczytane rejestry po ustawieniu:  [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Listing 3.1.4. Logi klienta

Porównanie danych przed i po operacji zapisu pozwala na weryfikację poprawności działania klienta oraz potwierdzenie wykonania operacji zapisu na serwerze Modbus.

3.2. Konfiguracja komunikacji Ethernet/IP między dwoma komputerami

4. Bezpieczeństwo komunikacji

4.1. Przegląd zagrożeń w komunikacji przemysłowej

4.2. Proste metody zabezpieczenia transmisji danych

5. Podsumowanie

Modbus TCP/IP i Ethernet/IP to dwa powszechnie stosowane protokoły komunikacyjne w przemyśle, każdy z nich mający swoje unikalne cechy i zastosowania. Modbus TCP/IP jest prosty i łatwy do implementacji, co czyni go idealnym do prostych aplikacji monitorujących i sterujących. Z kolei Ethernet/IP oferuje zaawansowane funkcje, wysoką wydajność i zintegrowane mechanizmy bezpieczeństwa, co czyni go bardziej odpowiednim dla złożonych i wymagających aplikacji przemysłowych. Wybór odpowiedniego protokołu zależy od specyficznych potrzeb i wymagań systemu przemysłowego.

Źródła

[1]. https://pl.wikipedia.org/wiki/Modbus

[2]. https://en.wikipedia.org/wiki/EtherNet/IP

[3]. https://pl.wikipedia.org/wiki/Ethernet