Industriel Netværksteknologi: Fra Teori til Praksis

Anders S. Østergaard

23. august 2024

Del I

Intro

Kapitel 1

Introduktion

Dette dokument er en samlet ressource til dokumentation af de netværksopgaver, der er udført med henblik på konfiguration og styring af automatiseringssystemer ved hjælp af TIA Portal, Studio 5000, Universal Robots og ABB's robotteknologi. Målet med dokumentet er at skabe en overskuelig guide, som kan assistere teknikere, studerende og undervisere indenfor automatiseringsteknologi med at forstå og udføre netværksrelaterede opgaver på tværs af forskellige automatiseringssystemer.

Rapporten er opdelt i to hoveddele: Ethernet-baserede netværk og busbaserede netværk. I TIA Portal-sektionen vil vi gennemgå opsætningen af kommunikationsnetværk for Siemens automatiseringshardware, hvilket inkluderer konfiguration af PROFINET. I Studio 5000-afsnittet vil fokus være på integrationen af Allen-Bradley udstyr og hvordan man opsætter Ethernet/IP netværk til industriel automatisering. For Universal Robots vil dokumentationen dække opsætningen af netværksforbindelser til UR-robotter, herunder fjernstyring og scriptkommunikation. Endelig vil ABB-robotafsnittet give vejledning i konfiguration af netværksparametre for ABB's robotter, med særlig opmærksomhed på RobotStudio.

Hver sektion vil indeholde skridt-for-skridt vejledning, eksempler på konfigurationsfiler og fejlfindingstips for at sikre, at læseren kan navigere i opgaverne med tillid. Dokumentet er tænkt som en levende ressource, der kan opdateres og udvides, efterhånden som teknologierne udvikler sig og flere erfaringer bliver indarbejdet.

1.1 Historie

For at forstå den nuværende tilstand og fremtidige retning for industrielle netværk, er det nyttigt at se på deres historiske udvikling. Her er nogle nøglepunkter i denne udvikling:

1.1.1 Tidlige Systemer

Før udbredelsen af digitale netværk blev industrielle systemer typisk styret af enkeltstående kontrolenheder og analog kommunikation. Dette resulterede i begrænset funktionalitet og fleksibilitet.

1.1.2 Fremkomsten af PLC'er

Programmable Logic Controllers (PLC'er) blev introduceret i 1960'erne og 1970'erne som en måde at automatisere og styre industrielle processer mere effektivt. PLC'er banede vejen for mere komplekse og fleksible automatiseringsløsninger.

1.1.3 Serielle Protokoller

I 1980'erne begyndte brugen af serielle kommunikationsprotokoller som RS232, RS422 og RS485 at blive almindelig i industrielle netværk, hvilket muliggør mere pålidelig dataoverførsel mellem enheder.

RS232, RS422, RS485: Disse er blandt de tidligste standarder for seriel kommunikation, introduceret i henholdsvis 1960'erne og 1970'erne. RS232 var en af de første bredt anvendte kommunikationsstandarder, der tillod forbindelse mellem computere og modemmer. RS422 og RS485 blev udviklet som forbedringer til RS232, hvor de tilbyder længere kabellængder og understøtter flere enheder på samme netværk. Disse standarder blev grundlaget for mange tidlige automatiseringssystemer.

1.1.4 Fieldbus Teknologier

I 1990'erne blev Fieldbus-protokoller som Profibus og DeviceNet udviklet, hvilket gjorde det muligt at forbinde mange enheder på et enkelt netværk med mere komplekse kommunikationsbehov.

PROFIBUS (Process Field Bus): Introduceret i 1980'erne, er PROFIBUS en af de tidligste standarder for feltbus kommunikation i industriel automatisering. Den blev udviklet for at standardisere kommunikationen mellem kontrolsystemer og feltudstyr, som f.eks. sensorer og aktuatorer. PROFIBUS understøtter både diskret og procesautomatisering og har været en af de mest udbredte industrielle netværksstandarder.

DeviceNet: DeviceNet, udviklet i 1990'erne, er baseret på CAN (Controller Area Network) og tilbyder en netværksløsning, der især fokuserer på lavniveau enhedsforbindelser, såsom sensorer og aktuatorer. Den understreger enkelthed og omkostningseffektivitet i automatiseringssystemer.

AS-I Bus (Actuator Sensor Interface): AS-I Bus, introduceret i begyndelsen af 1990'erne, er designet som en simpel og omkostningseffektiv løsning til forbindelse af binære sensorer og aktuatorer. Det giver en nem måde at implementere en sensor/aktuator-niveau netværksinfrastruktur.

1.2 Ethernet-baserede Netværk

I slutningen af 1990'erne og begyndelsen af 2000'erne blev Ethernet-baserede netværk som Ethernet/IP og Profinet introduceret. Disse teknologier udnyttede de høje hastigheder og standardiserede kommunikationsprotokoller fra IT-verdenen og tilpassede dem til industrielle behov.

PROFINET: Som opfølgeren til PROFIBUS, introduceret i begyndelsen af 2000'erne, er PROFINET en industriel Ethernet-standard designet til at imødekomme behovet for højere datahastigheder og realtid kommunikation. PROFINET udnytter standard Ethernet-teknologi og er optimeret til industrielle applikationer, hvilket giver større fleksibilitet og ydeevne.

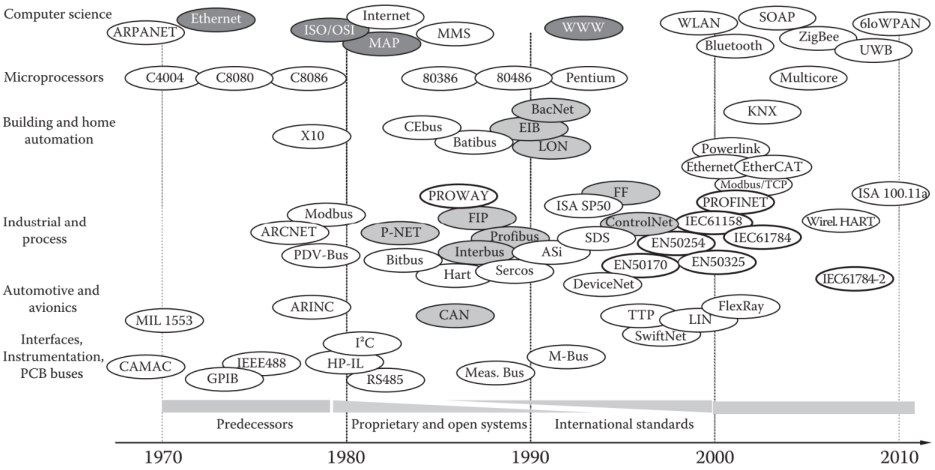
Ethernet/IP: Ethernet/IP, indført i slutningen af 1990'erne, er en industriel netværksstandard, der kombinerer traditionel Ethernet-teknologi med industriel protokol (CIP - Common Industrial Protocol). Den tilbyder en omfattende kommunikationsløsning, der understøtter både diskret og procesautomatisering.

Modbus RTU og TCP: Modbus, udviklet i slutningen af 1970'erne af Modicon (nu Schneider Electric), er en af de ældste og mest udbredte kommunikationsprotokoller i industrien. Modbus RTU (Remote Terminal Unit) er en seriel kommunikationsprotokol, der anvender RS485 standarden, mens Modbus TCP/IP er en version af Modbus, der bruger Ethernet til kommunikation, hvilket gør det muligt at anvende standard netværksinfrastruktur til industriel kommunikation.

1.3 Industri 4.0 og IoT

I det sidste årti har fremkomsten af Industri 4.0 og Internet of Things (IoT) revolutioneret industrielle netværk ved at integrere avancerede sensorer, cloud computing og big data-analyse, hvilket fører til mere intelligente og forbundne produktionssystemer.

IO-Link: IO-Link, som blev standardiseret i 2006, er en punkt-til-punkt kommunikationsteknologi, der anvendes til at forbinde intelligente sensorer og aktuatorer til et automatiseringssystem. IO-Link er designet til at overføre både procesdata og diagnostiske data og giver mulighed for detaljeret



Figur 1.1: Historisk visning af protokoller og standarder

sensor- og aktuatorstyring.

Del II

Fundamentale IT-Netværksteknologier

Kapitel 2

Introduktion til IT-Netværk

2.1 Grundlæggende Netværksbegreber

Netværk er systemer, der forbinder computere og andre enheder for at dele ressourcer og information. Der findes flere typer netværk, hver med specifikke egenskaber og anvendelser.

2.2 Hvad er et netværk?

Et netværk består af flere enheder (f.eks. computere, printere, servere) forbundet sammen for at dele data og ressourcer. Netværk muliggør kommunikation, samarbejde og adgang til information på tværs af geografiske afstande. De anvender forskellige teknologier og protokoller for at sikre effektiv og sikker dataoverførsel.

2.3 Netværkstyper

- **LAN (Local Area Network):** Et LAN er et lokalt netværk, der dækker et lille geografisk område som et kontor, en skole eller en bygning. Det giver højhastighedsforbindelse mellem enheder inden for et begrænset område og bruges til at dele ressourcer som printere og filer.
- **WAN (Wide Area Network):** Et WAN dækker et større geografisk område, som en by, et land eller endda flere lande. Internettet er det mest kendte eksempel på et WAN. WAN'er forbinder flere LAN'er og andre netværk for at muliggøre kommunikation og dataudveksling over lange afstande.
- **MAN (Metropolitan Area Network):** Et MAN dækker et byområde eller en stor campus og forbinder flere LAN'er inden for denne region. Det giver højhastighedsforbindelser og bruges ofte af store organisationer eller kommunale myndigheder.

2.4 Netværkets betydning

Netværk spiller en afgørende rolle i den moderne verden ved at muliggøre:

- **Deling af ressourcer:** Netværk tillader flere enheder at dele hardware (f.eks. printere) og software (f.eks. applikationer), hvilket reducerer omkostninger og øger effektiviteten.
- **Kommunikation:** Netværk muliggør hurtig og pålidelig kommunikation gennem e-mails, chat, videokonferencer og andre kommunikationsværktøjer.
- **Dataadgang:** Brugere kan få adgang til og dele data og filer på tværs af enheder og geografiske placeringer, hvilket fremmer samarbejde og informationsdeling.
- **Sikkerhed og administration:** Netværk gør det muligt at centralisere sikkerhed og administration, hvilket gør det lettere at implementere og håndhæve sikkerhedspolitikker og administrere ressourcer.

2.5 Netværkshardware

For at opbygge og vedligeholde netværk er det nødvendigt at bruge specifikke hardwarekomponenter. Her er en forklaring af de mest essentielle netværkshardwareenheder.

2.5.1 Repeaters og Hubs

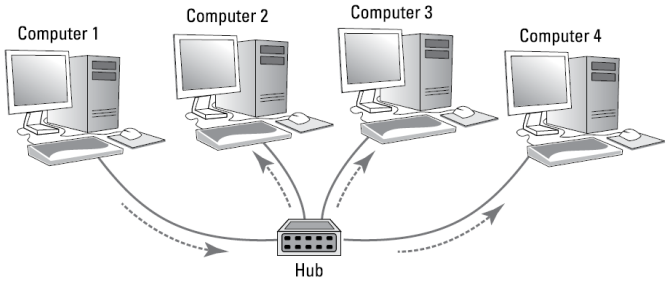
En **repeater** er en enhed på lag 1 i OSI-modellen, designet til at omgå den maksimale længdebegrænsning af twisted-pair netværkskabler. En repeater har to RJ45-porte, som er internt forbundet via en forstærker. Elektriske signaler, der modtages på en af portene, forstærkes og sendes gennem den anden port. Dermed kan kabler på begge sider af repeateren være op til 100 meter lange, hvilket effektivt fordobler rækkevidden af kablet.

En **hub** er en repeater med flere porte. For eksempel kan en hub have fire eller otte porte. Disse porte kan forbinde til andre enheder på netværket såsom en klientcomputer, en server eller en printer. En port på en hub kan også forbindes til en anden hub, hvilket gør det muligt at forbinde flere enheder sammen. For eksempel kan en otte-ports hub forbinde syv computere og en anden otte-ports hub, hvilket kan forbinde til yderligere syv computere. På denne måde kan to otte-ports hubs forbinde 14 computere til hinanden.

Der er to vigtige ting at vide om hubs:

Den første og vigtigste ting at vide om hubs er, at de næsten aldrig bruges længere. Det skyldes, at switches, som opererer på lag 2 i OSI-modellen, er mere effektive og almindeligt anvendt i moderne netværk.

Den anden vigtige ting at vide om hubs er, at et elektrisk signal modtaget på en af hubbens porte forstærkes og gentages på alle de andre porte i hubben. Således vil en enhed tilsluttet en af portene kunne se signalerne fra alle andre enheder tilsluttet de andre porte. For eksempel, i en otte-ports hub, vil et signal modtaget på port 1 blive forstærket og sendt ud til portene 2 til 8. På samme måde vil signaler modtaget på port 4 blive forstærket og sendt ud til portene 1 til 3 samt 5 til 8.



Figur 2.1: Diagram over en repeater og hub forbindelser

En hub er en simpel netværksenhed, der forbinder flere enheder i et LAN (Local Area Network). Den sender data, den modtager, til alle de enheder, der er forbundet til den, uden at tage hensyn til hvilken enhed dataene er beregnet til. Dette medfører ineffektiv dataoverførsel, da unødvendige data sendes til alle tilsluttede enheder. Hubs opererer på det fysiske lag (Layer 1) i OSI-modellen og har begrænset effektivitet sammenlignet med switches, som kun sender data til den specifikke modtager enhed.

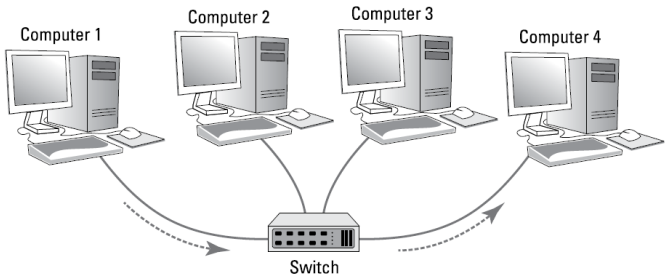
2.5.2 Switches

En **switch** er en mere avanceret netværksenhed, der opererer på lag 2 i OSI-modellen (datalinklaget). I modsætning til hubs, der sender data til alle tilsluttede enheder, sender en switch data kun til den specifikke enhed, som dataene er beregnet til. Dette gøres ved at opretholde en MAC-adressetabel, som kortlægger hver port til en specifik MAC-adresse.

Switches har flere fordele sammenlignet med hubs:

- **Effektivitet:** Fordi switches kun sender data til den specifikke modtager, reduceres unødvendig netværkstrafik, hvilket forbedrer den samlede netværkseffektivitet.

- **Sikkerhed:** Data sendes kun til den tilsigtede modtager, hvilket reducerer risikoen for, at data fanges op af andre enheder på netværket.
- **Ydeevne:** Switches kan håndtere flere samtidige forbindelser uden at forårsage kollisioner, hvilket er et almindeligt problem med hubs.



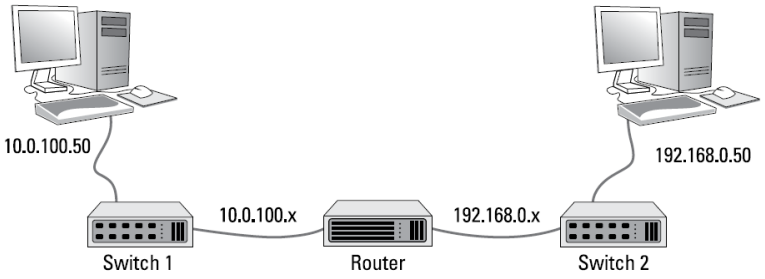
Figur 2.2: Switch

2.5.3 Routere

En **router** er en lag 3-enhed, hvilket betyder, at den arbejder på netværkslaget i OSI-modellen. I praksis betyder det, at routere arbejder med IP-adresser. Routere er afgørende for at forbinde forskellige netværk og styre netværkstrafikken mellem dem. En router adskiller sig fra en switch på flere måder:

- **IP-adresser:** Switches arbejder med MAC-adresser og ved ingenting om IP-adresser. I kontrast arbejder routere med IP-adresser.
- **Kommunikation mellem subnetværk:** Routere kan facilitere kommunikation mellem IP-netværk med forskellige subnet. For eksempel, hvis din organisation har et 10.0.100.x netværk og et 192.168.0.x netværk, kan en router muliggøre, at pakker kan rejse fra 10.0.100.x netværket til 192.168.0.x netværket og omvendt.
- **Internetforbindelse:** Routere muliggør også, at et privat netværk kan kommunikere med internettet. For eksempel, hvis du vil forbinde dit netværk til internettet via en bredbåndsudbyder, skal du bruge en router til at udveksle pakker mellem dit private netværk og internettet via den offentlige IP-adresse.

Den grundlæggende funktion af en router er ganske simpel. Overvej det simple netværk afbilledet i figur 2.3. Her har en organisation to separate IP-netværk, et med 10.0.100.x subnet og et andet med 192.168.0.x subnet. En router bruges til at forbinde disse to netværk. På begge sider af routeren er der en switch, og hver switch har kun en computer forbundet.



Figur 2.3: To IP-netværk forbundet via en router

Routeren bruger IP-adresser til at bestemme den bedste vej for data at rejse fra kilden til destinationen. Når en computer på det ene netværk ønsker at sende data til en computer på det andet netværk, sender den først dataene til switchen. Switch 1 videresender dataene til routeren, som derefter bestemmer den bedste vej og sender dataene videre til Switch 2, som til sidst sender dataene til den rigtige destination.

2.5.4 Modems

Et modem (modulator-demodulator) er en enhed, der konverterer digitale data fra en computer til analoge signaler, der kan overføres over telefonlinjer eller kabelnetværk og omvendt. Modemer bruges til at forbinde computere til internettet, især på steder hvor der ikke er tilgængelige bredbåndsforbindelser.

2.6 Sammenhæng mellem Netværkshardware

For at opbygge et effektivt og pålideligt netværk er det vigtigt at forstå, hvordan disse hardwarekomponenter arbejder sammen:

- **Interaktion mellem routere og switches:** Routere forbinder forskellige netværk, mens switches forbinder enheder inden for samme netværk. I et typisk netværk vil en switch forbinde computere og andre enheder i et LAN, og en router vil forbinde dette LAN til internettet.

- **Brug af hubs i små netværk:** Hubs kan anvendes i små netværk med få enheder, hvor effektiviteten ikke er et stort problem. I større netværk vil switches være mere effektive på grund af deres evne til at videresende data specifikt til den tiltænkte modtager.

- **Modemer til internetadgang:** Modemer bruges til at oprette forbindelse til internettet ved at konvertere digitale signaler til analoge

og omvendt. De kan arbejde sammen med routere for at distribuere internetadgang til flere enheder i et netværk.

2.7 Circuit Switching og Packet Switching

I netværkskommunikation findes der to grundlæggende metoder til dataoverførsel: circuit switching og packet switching.

2.7.1 Circuit Switching

Circuit switching indebærer oprettelsen af en dedikeret kommunikationskanal mellem to enheder for hele varigheden af en kommunikationssession. Dette er ligesom det traditionelle telefonsystem, hvor en linje er reserveret til en samtale fra start til slut. Fordelen ved circuit switching er, at det sikrer en kontinuerlig og stabil forbindelse, hvilket er ideelt til overførsel af data, der kræver konstant båndbredde. Ulempen er, at denne metode kan være ineffektiv og dyr, da ressourcerne forbliver allokeret, selv når der ikke sendes data.

2.7.2 Packet Switching

Packet switching, derimod, bryder dataene op i mindre pakker, som sendes individuelt gennem netværket. Hver pakke kan tage forskellige ruter til sin destination, afhængigt af netværksforholdene. Ved ankomsten samles pakkerne i den korrekte rækkefølge af modtagerens system. Denne metode er mere effektiv, da netværksressourcerne deles mellem flere brugere, hvilket muliggør optimal udnyttelse af båndbredden. Packet switching anvendes i de fleste moderne datanetværk, såsom internettet, hvor det sikrer fleksibel og pålidelig dataoverførsel.

2.7.3 Forbindelsesorienteret og Forbindelsesløs Kommunikation

Packet-switched netværk kan tilbyde både forbindelsesorienterede og forbindelsesløse kommunikationer. Forbindelsesorienteret kommunikation, som f.eks. TCP (Transmission Control Protocol), etablerer en virtuel forbindelse, hvor dataoverførslerne er struktureret og pålidelige. Forbindelsesløs kommunikation, som f.eks. UDP (User Datagram Protocol), sender data uden at etablere en vedvarende forbindelse, hvilket er hurtigere men mindre pålideligt.

2.7.4 Praktisk Anvendelse

I moderne industrielle netværk anvendes packet switching bredt på grund af dets evne til at håndtere store datamængder effektivt. Denne teknologi gør

det muligt at sende data fra mange forskellige kilder samtidigt, hvilket er afgørende for komplekse automationssystemer, hvor forskellige sensorer og enheder konstant kommunikerer.

Sammenfattende er forståelsen af både circuit switching og packet switching vigtig for at kunne designe og implementere effektive netværksløsninger, der opfylder specifikke behov for stabilitet, hastighed og effektivitet.

2.8 Media Access Control (MAC) mechanisms

Medieadgangskontrolmekanismer er essentielle for at regulere, hvordan data sendes og modtages i netværk. Der er tre primære metoder, der anvendes til at styre adgangen til mediet: Master-slave, Token-passing og CSMA/CD.

2.8.1 Master-slave (eller forespørgsel-svar) metode

Master-slave metoden, også kendt som forespørgsel-svar metoden, anvendes ofte i netværk, hvor en central node (master) skal kommunikere med flere underordnede noder (slaver). I denne opsætning styrer masternoden kommunikationen ved at sende forespørgsler til slavenoderne og vente på svar. Processen fungerer således:

- Masternoden sender en besked til den første slave i rækken, der enten anmoder om data eller sender data til slaven.
- Alle slavenoder modtager beskeden, men kun den node, hvis adresse matcher beskedens destination, reagerer. De andre noder ignorerer beskeden.
- Den adresserede slavenode læser beskeden og kontrollerer for eventuelle fejl. Hvis der opdages fejl, såsom uoverensstemmelser i checksummen, afvises beskeden.
- Hvis en slave ikke reagerer, forsøger masternoden at kontakte den op til tre gange, før den går videre til den næste slave i rækken.
- Denne cyklus fortsætter, indtil alle slavenoder er blevet kontaktet, hvilket udgør en fuld forespørgselscyklus.

Fordelen ved denne metode er enkelheden i opsætningen og den fulde kontrol, masternoden har over kommunikationen. Dette gør det nemt at administrere dataflowet, især når hver slave har en forudsigelig mængde data. Ulempen er dog, at systemet kan være ineffektivt, hvis en slave har behov for at sende uforudsete mængder data, eller hvis der opstår behov for hurtig overførsel af kritiske data.

2.8.2 Token-passing

Token-passing er en metode, hvor kontrollen over netværket overføres mellem noder via en særlig besked kaldet en token. Denne metode bruges ofte i netværk, der kræver pålidelig og garanteret dataoverførsel, såsom industrielle kontrolsystemer. Token-passing fungerer på følgende måde:

- En node modtager token-beskeden fra en nærliggende node og får dermed kontrol over netværket.
- Noden beholder token i en bestemt tidsperiode eller indtil den har sendt sine beskeder.
- Noden sender derefter data til andre noder og overfører token til den næste node i rækken.
- Denne proces gentages, hvilket sikrer, at alle noder får en chance for at sende data inden for et givet tidsrum.

Fordelen ved token-passing er, at det sikrer deterministisk adgang til mediet, hvilket betyder, at alle noder får lige mulighed for at sende data. Dette er især vigtigt i systemer, hvor tidssensitive dataoverførsler er afgørende. Eksempler på netværk, der anvender token-passing, inkluderer Arcnet (stjernetopologi), Modbus (bustopologi) og IBM token ring (ringtopologi).

2.8.3 CSMA/CD (Carrier Sense Multiple Access/Collision Detection)

CSMA/CD er en enkel og effektiv metode til at regulere datatrafik i et netværk. Denne metode anvendes ofte i netværk som Ethernet og fungerer ved, at noder lytter efter ledig båndbredde, før de sender data. CSMA/CD processen er som følger:

- En node, der ønsker at sende data, lytter først efter aktivitet på netværket. Hvis netværket er ledigt, begynder noden at sende data.
- Under transmissionen sammenligner noden de sendte data med de data, der er til stede på netværket. Hvis der opdages en kollision (dvs. to noder sender samtidig), stopper transmissionen øjeblikkeligt.
- De kolliderende noder venter en tilfældig periode, før de forsøger at sende igen, hvilket reducerer risikoen for yderligere kollisioner.

CSMA/CD er enkel og effektiv, især i netværk med let trafik. Dog kan metoden blive ineffektiv ved høj trafikbelastning, da kollisioner kan blive hyppige og forsinke dataoverførsler. Det mest almindelige eksempel på CSMA/CD er Ethernet.

2.9 Transmissionsteknikker

Transmissionsteknikker beskriver, hvordan data overføres over et netværk. De to mest anvendte teknikker er baseband og broadband.

2.9.1 Baseband

Baseband transmission er en metode, hvor hele båndbredden af mediet anvendes af en enkelt kommunikationskanal ad gangen. Denne teknik er karakteriseret ved, at signalet sendes direkte uden modulation, hvilket betyder, at den elektriske signalspænding varierer direkte med dataene.

- **Tidssignalering (TDM):** Baseband transmission bruger ofte tidsdeling multiplexing (Time Division Multiplexing, TDM), hvor hver enhed tildeles en bestemt tidslomme til at transmittere data. Dette sikrer, at kun én enhed sender data ad gangen, hvilket eliminerer risikoen for kollisioner.

- **Anvendelse:** Denne metode er almindeligt anvendt i Ethernet-netværk, specielt i det oprindelige 10BASE-T Ethernet, hvor data sendes over twisted pair-kabler uden behov for yderligere modulation.

- **Fordele:**

  - Simpel og nem at implementere.
  - Lav omkostning på grund af enkelheden i udstyr og kabling.

- **Ulemper:**

  - Begrænset rækkevidde og båndbredde sammenlignet med broadband.
  - Ikke velegnet til lange afstande uden brug af repeatere.

2.9.2 Broadband

Broadband transmission er en metode, hvor mediets båndbredde opdeles i flere kanaler, som hver især kan transmittere data samtidigt. Dette gøres ved at modulere dataene på forskellige frekvensbærere, hvilket tillader flere signaler at dele samme fysiske medie uden at interferere med hinanden.

- **Frekvensdeling (FDM):** Broadband bruger frekvensdeling multiplexing (Frequency Division Multiplexing, FDM), hvor hver kanal tildeles en specifik frekvensbånd. Dette muliggør parallel transmission af data fra flere enheder.

- **Anvendelse:** Broadband transmission anvendes typisk i kabelnetværk og fiberoptiske netværk, hvor høj båndbredde og lange afstande er nødvendige. Eksempler inkluderer kabel-tv og bredbåndsinternetforbindelser.

- **Fordele:**
  - Høj båndbredde, hvilket muliggør hurtigere dataoverførsel.
  - Kan transmittere over længere afstande uden behov for repeatere.

- **Ulemper:**
  - Mere komplekst og dyrere at implementere.
  - Kræver specialiseret udstyr til modulation og demodulation.

Data transmitteres ved at modulere en bærebølge med informationen, hvilket gør det muligt at udnytte større båndbredder og tillade højere dataoverførselshastigheder. Koaksialkabler og optiske fibre er typisk foretrukne medier for FDM på grund af deres høje båndbreddekapacitet.

Disse transmissionsteknikker og medieadgangskontrolmekanismer spiller en afgørende rolle i design og implementering af effektive og pålidelige netværk, både i kommercielle og industrielle applikationer.

Eksempel: Profinet bruger både Ethernet-standarder (CSMA/CD) og har specifikationer for det fysiske lag, hvilket gør det til en robust løsning for industrielle netværk.

2.10 OSI-modellen og dens anvendelse i industrielle netværk

Open Systems Interconnection (OSI) modellen er en konceptuel ramme, der bruges til at forstå og implementere standarder for netværkskommunikation. Modellen opdeler netværkskommunikation i syv forskellige lag, hvor hvert lag har specifikke funktioner og tjenester. Disse lag er designet til at interagere med hinanden på en standardiseret måde, hvilket muliggør kommunikation mellem forskellige netværksudstyr og protokoller.

2.10.1 Lag 1: Fysisk Lag

Det fysiske lag er det nederste lag i OSI-modellen. Det håndterer den fysiske forbindelse mellem enheder og den faktiske transmission og modtagelse af data i form af elektriske signaler, lys eller radiobølger. Dette lag omfatter hardwarekomponenter som kabler, switches og netværkskort. I industrielle

netværk inkluderer dette lag også miljømæssige faktorer som elektromagnetisk interferens og temperaturvariationer, hvilket kan påvirke dataoverførslen.

2.10.2 Lag 2: Datalink Lag

Datalink laget er ansvarligt for pålidelig dataoverførsel mellem to enheder på samme netværk. Det opdeler data i frames og håndterer fejlkorrektion fra det fysiske lag. Datalink laget består af to underlag: Media Access Control (MAC) og Logical Link Control (LLC). I industrielle netværk bruges protokoller som EtherCAT og PROFINET på dette lag for at sikre hurtig og pålidelig dataoverførsel mellem kontroller og feltudstyr.

2.10.3 Lag 3: Netværks Lag

Netværks laget styrer routing af data mellem forskellige netværk. Dette lag opdeler data i pakker og bestemmer den bedste vej til destinationen ved hjælp af routingprotokoller som IP (Internet Protocol). Netværks laget håndterer også logiske adressering. I industrielle netværk bruges ofte avancerede routingteknikker til at sikre, at data når deres destination hurtigt og effektivt, selv i komplekse netværksstrukturer.

2.10.4 Lag 4: Transport Lag

Transport laget er ansvarligt for pålidelig dataoverførsel mellem to endepunkter. Det opdeler data i segmenter og sikrer, at dataene når frem uden fejl og i den rigtige rækkefølge. Protokoller som TCP (Transmission Control Protocol) og UDP (User Datagram Protocol) opererer på dette lag. I industrielle netværk er transportlaget afgørende for at sikre, at kontrolsignaler og dataoverførsler er præcise og pålidelige.

2.10.5 Lag 5: Session Lag

Session laget styrer etablering, vedligeholdelse og afslutning af kommunikationssessioner mellem applikationer. Det sikrer, at sessioner forbliver adskilte og organiserer dataudveksling i sessioner. I industrielle applikationer bruges dette lag til at administrere vedvarende forbindelser mellem kontrolsystemer og deres tilhørende enheder, hvilket sikrer kontinuerlig drift og overvågning.

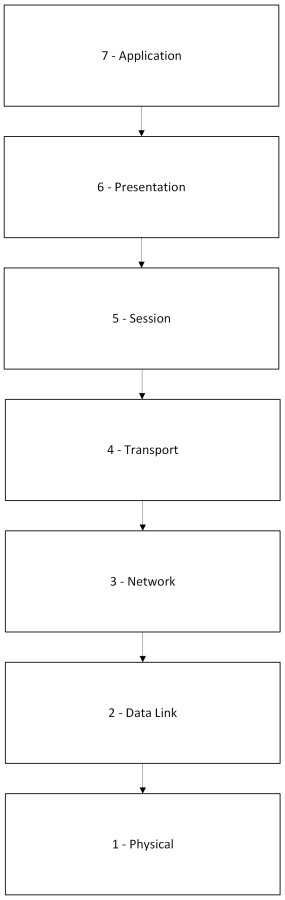
2.10.6 Lag 6: Præsentations Lag

Præsentations laget er ansvarligt for datatranslation, datakomprimering og datakryptering. Det sørger for, at data, der sendes fra applikationslaget, er i et format, der kan forstås af modtagerens applikationslag. I industrielle netværk kan dette lag være involveret i konvertering af dataformater mellem

forskellige systemer og sikring af, at data er korrekt krypteret for sikkerhedsmæssige formål.

2.10.7 Lag 7: Applikations Lag

Applikations laget er det øverste lag i OSI-modellen og fungerer som grænseflade mellem netværkstjenester og applikationssoftware. Det tilbyder tjenester som e-mail, filoverførsel og webbrowseradgang. Protokoller som HTTP, FTP og SMTP opererer på dette lag. I industrielle netværk omfatter dette lag også specialiserede applikationer til processtyring, overvågning og dataindsamling, hvilket gør det muligt for operatører at interagere med og styre industrielle systemer.



Figur 2.4: Illustration af OSI-modellens syv lag

2.10.8 Sammenhæng mellem lagene

Hvert lag i OSI-modellen kommunikerer direkte med de lag, der er umiddelbart over og under det. Data, der sendes fra en applikation, bevæger sig ned gennem lagene, hvor hver lag tilføjer sine egne specifikke oplysninger (f.eks. headers og trailers), inden de transmitteres over netværket. Ved modtagelse bevæger dataene sig op gennem lagene, hvor hver lag fjerner sine tilsvarende oplysninger og videresender de relevante data til det næste lag.

- **Fordele ved OSI-modellen:**
  - Standardisering: Tilbyder en standardiseret ramme for netværkskommunikation.
  - Interoperabilitet: Muliggør interoperabilitet mellem forskellige netværksprodukter og -teknologier.
  - Modularitet: Tillader udvikling og fejlfinding af individuelle lag uden at påvirke de andre lag.

For en visuel forståelse, se Figur 2.4, som illustrerer OSI-modellens syv lag og deres funktioner.

2.10.9 Anvendelse af OSI-modellen i industrielle netværk

I industrielle netværk anvendes OSI-modellen til at designe og implementere netværksinfrastrukturer, der opfylder specifikke krav til pålidelighed, sikkerhed og effektivitet. Hvert lag i OSI-modellen bidrager til den samlede funktionalitet og ydeevne af netværket.

Eksempel på Anvendelse: PROFINET PROFINET er en industriel Ethernet-standard, der bruger OSI-modellen som grundlag. PROFINET opererer primært på datalinklaget (Lag 2) og netværkslaget (Lag 3), hvor det leverer realtidskommunikation og understøtter komplekse industrielle applikationer.

Fordele ved at Bruge OSI-modellen Anvendelse af OSI-modellen i industrielle netværk har flere fordele:

- **Standardisering:** OSI-modellen fremmer brugen af standardprotokoller, hvilket gør det lettere at integrere forskellige enheder og systemer fra forskellige producenter.
- **Fejlfinding:** Ved at opdele netværksfunktioner i adskilte lag bliver det lettere at identificere og isolere problemer.
- **Fleksibilitet:** OSI-modellen tillader udskiftning og opgradering af individuelle lag uden at påvirke hele netværket.

- **Pålidelighed:** Ved at implementere specifikke funktioner på hvert lag kan netværket opnå højere pålidelighed og ydeevne.

Implementeringsstrategier Ved implementering af industrielle netværk skal ingeniører og teknikere:

- Vælge passende protokoller og teknologier for hvert lag baseret på specifikke behov.
- Sikre, at alle lag i OSI-modellen er korrekt konfigureret og integreret.
- Udføre grundig testning og validering for at sikre, at netværket fungerer som forventet.

OSI-modellen er et uvurderligt værktøj i planlægning, design og implementering af industrielle netværk. Ved at forstå og anvende denne model kan netværksingeniører skabe robuste, sikre og effektive netværk, der opfylder moderne industrikrav.

2.11 TCP/IP-modellen

TCP/IP-modellen (Transmission Control Protocol/Internet Protocol) er en konceptuel model og et sæt af kommunikationsprotokoller, der bruges til at forbinde enheder på internettet. Modellen opdeler netværkskommunikation i fire forskellige lag, som tilsammen muliggør dataudveksling mellem netværk.

Lag 1: Netværksadgangslaget Dette lag omfatter de protokoller og teknologier, der anvendes til fysisk at forbinde og overføre data mellem enheder på samme netværk. Det svarer til OSI-modellens fysiske og datalink-lag. Protokoller og teknologier i dette lag inkluderer Ethernet, Wi-Fi, og ARP (Address Resolution Protocol).

Lag 2: Internetlaget Internetlaget er ansvarligt for routing af data mellem forskellige netværk. Dette lag tilsvarer OSI-modellens netværkslag og bruger IP (Internet Protocol) til at bestemme, hvordan pakker skal dirigeres fra afsender til modtager. Andre vigtige protokoller i dette lag inkluderer ICMP (Internet Control Message Protocol) og IGMP (Internet Group Management Protocol).

Lag 3: Transportlaget Transportlaget sørger for pålidelig dataoverførsel mellem to endepunkter. Dette lag svarer til OSI-modellens transportlag og omfatter to primære protokoller: TCP (Transmission Control Protocol) og UDP (User Datagram Protocol). TCP er ansvarlig for at sikre, at data leveres uden fejl og i den rigtige rækkefølge, mens UDP giver en hurtigere, men mindre pålidelig overførsel.

Lag 4: Applikationslaget Applikationslaget er det øverste lag i TCP/IP-modellen og omfatter alle de protokoller, der anvendes af applikationer til at kommunikere over netværket. Dette lag tilsvarer de tre øverste lag i OSI-modellen (session, præsentation, og applikation). Eksempler på protokoller i dette lag inkluderer HTTP (HyperText Transfer Protocol), FTP (File Transfer Protocol), SMTP (Simple Mail Transfer Protocol), og DNS (Domain Name System).

Sammenligning med OSI-modellen TCP/IP-modellen og OSI-modellen er begge designet til at hjælpe med forståelsen af netværkskommunikation. Mens OSI-modellen har syv lag, har TCP/IP-modellen kun fire. OSI-modellen bruges ofte som en teoretisk ramme, mens TCP/IP-modellen er praktisk og implementeret i de fleste netværk i dag.

Fordele ved TCP/IP-modellen Robusthed: TCP/IP er designet til at være robust og pålidelig, selv i tilfælde af fejl og tab af data. Skalerbarhed: Modellen er meget skalerbar og kan håndtere et stort antal enheder og netværk. Interoperabilitet: TCP/IP tillader kommunikation mellem enheder fra forskellige producenter og operativsystemer. Standardisering: TCP/IP er en de facto standard for netværkskommunikation, hvilket gør det universelt accepteret og brugt.

Anvendelser af TCP/IP-modellen TCP/IP-modellen anvendes bredt i moderne netværk, herunder internettet, lokale netværk (LAN), og virksomhedsintranets. Den understøtter et bredt udvalg af applikationer, fra web browsing og e-mail til filoverførsel og streamingtjenester.

2.12 Forskellen mellem OSI- og TCP/IP-modellen

Introduktion

I netværkskommunikation er OSI- og TCP/IP-modellerne to fundamentale rammer, der hjælper med at forstå, hvordan data overføres mellem enheder. OSI-modellen er en konceptuel model, der primært bruges til undervisning og standardisering, mens TCP/IP-modellen er den praktisk anvendte model på internettet i dag.

2.13 OSI-modellen

2.13.1 Struktur

OSI-modellen (Open Systems Interconnection) består af syv lag, der hver især har specifikke funktioner:

1. **Fysisk Lag:** Transmission af rå bitstrømme over et fysisk medium.

2. **Datalink Lag:** Sikrer pålidelig dataoverførsel over et fysisk link.

3. **Netværks Lag:** Routing af data mellem forskellige netværk.

4. **Transport Lag:** Sikrer pålidelig dataoverførsel mellem to endepunkter.

5. **Session Lag:** Styrer etablering, vedligeholdelse og afslutning af kommunikationssessioner.

6. **Præsentations Lag:** Datatranslation, datakomprimering og datakryptering.

7. **Applikations Lag:** Grænseflade mellem netværkstjenester og applikationssoftware.

2.13.2 Formål og Anvendelse

OSI-modellen blev udviklet af ISO (International Organization for Standardization) som en teoretisk ramme for standardisering af netværkskommunikation. Den bruges hovedsageligt i uddannelsesmiljøer til at forstå og designe netværksprotokoller og -systemer.

2.14 TCP/IP-modellen

2.14.1 Struktur

TCP/IP-modellen (Transmission Control Protocol/Internet Protocol) har fire lag, der er designet til praktisk brug i netværkskommunikation:

1. **Netværksadgangslaget:** Håndterer fysisk forbindelse og datalink, herunder hardwarekomponenter som kabler og switches.

2. **Internetlaget:** Ansvarlig for routing af data mellem forskellige netværk ved hjælp af IP (Internet Protocol).

3. **Transportlaget:** Pålidelig dataoverførsel mellem to endepunkter med protokoller som TCP og UDP.

4. **Applikationslaget:** Protokoller, der anvendes af applikationer til netværkskommunikation, som HTTP, FTP, SMTP.

2.14.2 Formål og Anvendelse

TCP/IP-modellen blev udviklet af det amerikanske forsvarsministerium (DoD) for at understøtte netværkskommunikation i ARPANET, som var forløberen til internettet. Denne model er praktisk anvendt og designet til at lette kommunikation over forskellige typer netværk, og den er grundlaget for moderne internetinfrastruktur.

2.15 Sammenligning mellem OSI og TCP/IP

2.15.1 Lagstruktur

- **OSI-modellen:** Består af syv detaljerede lag, der opdeler netværkskommunikation i specifikke funktioner.
- **TCP/IP-modellen:** Har fire lag, hvor flere funktioner fra OSI-modellen kombineres. For eksempel dækker Netværksadgangslaget i TCP/IP både det fysiske lag og datalinklaget i OSI-modellen.

2.15.2 Abstraktionsniveau og Brug

- **OSI-modellen:** Tilbyder en detaljeret og lagdelt tilgang, der primært anvendes som en teoretisk reference for netværkskommunikation.
- **TCP/IP-modellen:** Er mere praktisk og bruges i reel netværkskommunikation og internetprotokoller.

2.15.3 Udvikling og Anvendelse

- **OSI-modellen:** Udviklet af en international standardiseringsorganisation og anvendes mest til pædagogiske formål.
- **TCP/IP-modellen:** Udviklet af det amerikanske forsvar og er den mest anvendte model i moderne netværkskommunikation, især på internettet.

2.16 Opsummering: OSI vs TCP/IP

OSI-modellen giver en teoretisk og detaljeret ramme, der er nyttig for undervisning og standardisering, mens TCP/IP-modellen er praktisk anvendt i moderne netværksinfrastruktur, især på internettet. Sammen supplerer disse modeller hinanden og giver en omfattende forståelse af netværkskommunikation.

Kapitel 3

Ethernet-teknologier og Protokoller

3.1 Internet Protocol version 4 (IPv4)

Internetprotokollen (IP) er kernen i TCP/IP-protokolsuiten. IP er primært ansvarlig for routing af pakker mod deres destination, fra router til router. Denne routing udføres på basis af IP-adresser, som er indlejret i headeren, der er knyttet til hver pakke, der videresendes af IP.

Den mest udbredte version af IP i dag er version 4 (IPv4), som bruger en 32-bit adresse. En IPv4-adresse består af fire oktetter (8-bit segmenter), adskilt af punktummer. For eksempel er `192.168.1.1` en gyldig IPv4-adresse.

IPv4 er ved at nå slutningen af sin levetid, hovedsageligt på grund af det begrænsede antal tilgængelige adresser. For at imødegå dette problem er version 6 (IPv6 eller IPng) blevet udviklet, som bruger en 128-bit adresse. Selvom denne vejledning primært fokuserer på IPv4 for at forklare de grundlæggende processer, vil den også give en introduktion til IPv6.

3.1.1 Kilde til IP-adresser

En IP-adresse er en unik numerisk identifikator, der tildeles hver enhed, der er forbundet til et computernetværk, der bruger internetprotokollen til kommunikation. Disse adresser stammer fra Internet Assigned Numbers Authority (IANA), som er ansvarlig for at fordele dem globalt. IANA har delegeret denne opgave til tre regionale internetregistre (RIRs):

- **APNIC:** Asien og Stillehavsområdet (`http://www.apnic.net`)
- **ARIN:** Nordamerika og dele af Caribien (`http://www.arin.net`)

- **RIPE NCC:** Europa, Mellemøsten og dele af Centralasien (`http://www.ripe.net`)

RIR'erne tildeler blokke af IP-adresser til internetudbydere (ISPs), som derefter tildeler dem til deres kunder. For at kunne oprette forbindelse til internettet skal en enhed have en gyldig IP-adresse. Enheder, der ikke er forbundet til internettet, kan bruge private IP-adresser. Disse adresser er ikke unikke globalt og kan derfor ikke bruges til at kommunikere direkte med enheder på internettet.

3.1.2 IP-adressens rolle i netværkskommunikation

En IP-adresse fungerer som en digital postadresse for enheder, der er forbundet til et netværk. Den gør det muligt at identificere hver enkelt enhed og dirigere data mellem dem pålideligt. Uden IP-adresser ville internettet og andre netværk ikke kunne fungere.

IP-adressens hovedfunktioner

- **Unik identifikation:** Hver enhed på et netværk får tildelt en unik IP-adresse, der adskiller den fra alle andre enheder. Det er lidt ligesom et CPR-nummer for mennesker.

- **Routing af data:** Når du sender data over et netværk, f.eks. en e-mail eller besøger en hjemmeside, bruges IP-adressen til at bestemme den bedste rute for dataene at følge fra din enhed til destinationen.

- **Internetkommunikation:** For at din computer, smartphone eller anden enhed kan kommunikere med andre enheder på internettet, skal den have en gyldig og unik IP-adresse.

3.1.3 IP-adresser vs. MAC-adresser

Du har måske hørt om MAC-adresser (også kaldet hardware-adresser). De er også unikke identifikatorer for netværksenheder, men de er mere som et serienummer, der er indbygget i enheden fra fabrikken. MAC-adresser bruges primært til kommunikation inden for et lokalt netværk (LAN), hvor alle enheder er direkte forbundet.

IP-adresser er derimod nødvendige for kommunikation på tværs af forskellige netværk, som f.eks. internettet. De er mere fleksible end MAC-adresser, da de kan ændres, hvis enhed flyttes til et andet netværk eller skifter internetudbyder.

Hvordan IP-adresser og MAC-adresser arbejder sammen Når du sender data til en anden enhed på internettet, sker der en oversættelse mellem IP-adressen og MAC-adressen. Din enhed bruger IP-adressen til at finde den rigtige destination på internettet. Når dataene når det lokale netværk, hvor destinationen befinder sig, bruges MAC-adressen til at levere dataene til den specifikke enhed.

Denne oversættelse mellem IP- og MAC-adresser sker automatisk ved hjælp af en protokol kaldet ARP (Address Resolution Protocol).

3.1.4 Netværks-ID og Host-ID

En IPv4-adresse kan opdeles i to dele: Netværks-ID og Host-ID. Netværks-ID identificerer det overordnede netværk, mens Host-ID identificerer en specifik enhed inden for det netværk. Subnetmasken bruges til at bestemme, hvilken del af adressen der er netværks-ID, og hvilken der er Host-ID.

Eksempel:
For IP-adressen `192.168.1.10` med subnetmasken `255.255.255.0`:

- Netværks-ID: `192.168.1.0`
- Host-ID: `10`

3.1.5 Adresseklasser

IPv4-adresser er opdelt i fem klasser (A, B, C, D, og E) baseret på de første bits af adressen:

- **Klasse A:** 0.0.0.0 til 127.255.255.255 (stor mængde hosts pr. netværk)
- **Klasse B:** 128.0.0.0 til 191.255.255.255 (medium mængde hosts pr. netværk)
- **Klasse C:** 192.0.0.0 til 223.255.255.255 (lille mængde hosts pr. netværk)
- **Klasse D:** 224.0.0.0 til 239.255.255.255 (multicast-adresser)
- **Klasse E:** 240.0.0.0 til 255.255.255.255 (eksperimentelle adresser)

3.1.6 Bestemmelse af adresseklasse ved inspektion

Adresseklassen kan bestemmes ved at inspicere de første bits i en IPv4-adresse:

- Klasse A: Første bit er 0 (f.eks. `1.0.0.0` til `127.255.255.255`)

- Klasse B: Første to bits er 10 (f.eks. `128.0.0.0` til `191.255.255.255`)
- Klasse C: Første tre bits er 110 (f.eks. `192.0.0.0` til `223.255.255.255`)
- Klasse D: Første fire bits er 1110 (f.eks. `224.0.0.0` til `239.255.255.255`)
- Klasse E: Første fire bits er 1111 (f.eks. `240.0.0.0` til `255.255.255.255`)

3.1.7 Antal netværk og værter pr. adresseklasse

Antallet af netværk og værter i hver adresseklasse varierer:

- **Klasse A:** 128 netværk, 16.777.214 værter pr. netværk
- **Klasse B:** 16.384 netværk, 65.534 værter pr. netværk
- **Klasse C:** 2.097.152 netværk, 254 værter pr. netværk

3.2 Subnetmasker

En subnetmaske er en 32-bit adresse, der opdeler en IP-adresse i netværks- og hostdele. Eksempler:

- `255.255.255.0` (CIDR notation: /24) - 24 bits til netværksdelen og 8 bits til hostdelen.
- `255.255.0.0` (CIDR notation: /16) - 16 bits til netværksdelen og 16 bits til hostdelen.

Eksempel:
For IP-adressen `192.168.1.10` med subnetmasken `255.255.255.0`:

- Netværks-ID: `192.168.1.0`
- Host-ID: `10`

3.2.1 Subnetting

Subnetting er processen med at opdele et større netværk i mindre subnets for at forbedre effektiviteten og sikkerheden. Subnetting bruger subnetmasker til at opdele IP-adressen i mindre, håndterbare segmenter.

3.2.2 Hvorfor subnetting?

Subnetting bruges til at:

- Reducere netværkstrafik ved at mindske antallet af broadcasts.
- Forbedre sikkerheden ved at isolere segmenter af netværket.
- Effektivisere IP-adressebrug ved at opdele store netværk i mindre, lettere håndterbare subnets.

3.2.3 Matematisk proces for subnetting

For at opdele et netværk i subnets, følg disse trin:

1. Bestem antallet af nødvendige subnets eller hosts pr. subnet.
2. Beregn subnetmasken:
   - Antallet af bits, der skal lånes fra host-delen, bestemmes af antallet af nødvendige subnets.
   - Brug formelen $2^n \geq \textit{antal subnets}$, hvor n er antallet af lånte bits.
   - Subnetmasken kan derefter beregnes ved at tilføje de lånte bits til netværks-delen af den oprindelige subnetmaske.
3. Beregn antallet af hosts pr. subnet:
   - Brug formelen $2^h - 2$, hvor h er antallet af bits i host-delen.
4. Identificer subnet-ID'er:
   - Subnet-ID'er kan identificeres ved at øge subnet-bitsene trinvis.

Eksempel 1: Subnetting et Class C-netværk

Scenarie: En virksomhed har et Class C-netværk `192.168.1.0/24` og ønsker at opdele dette netværk i 4 mindre subnets for at adskille forskellige afdelinger.

Beregninger:

1. Bestem antallet af nødvendige subnets: Virksomheden ønsker 4 subnets.
2. Beregn subnetmasken:
   - Antallet af nødvendige bits for subnetting kan beregnes med $2^n \geq 4$, hvor n er antallet af lånte bits. Så $n = 2$.
   - Den oprindelige subnetmaske er `255.255.255.0` (/24). Ved at låne 2 bits fra host-delen, bliver den nye subnetmaske `255.255.255.192` (/26).
3. Beregn antallet af hosts pr. subnet:
   - Antallet af hosts pr. subnet kan beregnes med $2^h - 2$, hvor h er antallet af bits i host-delen. For /26 subnetmasken har vi 6 bits til hosts, så $2^6 - 2 = 62$ hosts pr. subnet.
4. Identificer subnet-ID'er:
   - Første subnet: `192.168.1.0/26` (hosts: `192.168.1.1` til `192.168.1.62`)

- Andet subnet: `192.168.1.64/26` (hosts: `192.168.1.65` til `192.168.1.126`)
- Tredje subnet: `192.168.1.128/26` (hosts: `192.168.1.129` til `192.168.1.190`)
- Fjerde subnet: `192.168.1.192/26` (hosts: `192.168.1.193` til `192.168.1.254`)

Eksempel 2: Subnetting et Class B-netværk

Scenarie: En skole har et Class B-netværk `172.16.0.0/16` og ønsker at opdele dette netværk i 16 subnets for at adskille forskellige bygninger.

Beregninger:

1. Bestem antallet af nødvendige subnets: Skolen ønsker 16 subnets.
2. Beregn subnetmasken:
   - Antallet af nødvendige bits for subnetting kan beregnes med $2^n \geq 16$, hvor n er antallet af lånte bits. Så $n = 4$.
   - Den oprindelige subnetmaske er `255.255.0.0` (/16). Ved at låne 4 bits fra host-delen, bliver den nye subnetmaske `255.255.240.0` (/20).
3. Beregn antallet af hosts pr. subnet:
   - Antallet af hosts pr. subnet kan beregnes med $2^h - 2$, hvor h er antallet af bits i host-delen. For /20 subnetmasken har vi 12 bits til hosts, så $2^{12} - 2 = 4094$ hosts pr. subnet.
4. Identificer subnet-ID'er:
   - Første subnet: `172.16.0.0/20` (hosts: `172.16.0.1` til `172.16.15.254`)
   - Andet subnet: `172.16.16.0/20` (hosts: `172.16.16.1` til `172.16.31.254`)
   - Tredje subnet: `172.16.32.0/20` (hosts: `172.16.32.1` til `172.16.47.254`)
   - Fortsæt for de resterende subnets: `172.16.48.0/20`, `172.16.64.0/20`, osv.

Eksempel 3: Subnetting et Class A-netværk

Scenarie: En stor virksomhed har et Class A-netværk `10.0.0.0/8` og ønsker at opdele dette netværk i 256 subnets for at adskille forskellige afdelinger og lokationer.

Beregninger:

1. Bestem antallet af nødvendige subnets: Virksomheden ønsker 256 subnets.
2. Beregn subnetmasken:

- Antallet af nødvendige bits for subnetting kan beregnes med $2^n \geq 256$, hvor n er antallet af lånte bits. Så $n = 8$.
- Den oprindelige subnetmaske er `255.0.0.0` (/8). Ved at låne 8 bits fra host-delen, bliver den nye subnetmaske `255.255.0.0` (/16).

3. Beregn antallet af hosts pr. subnet:
   - Antallet af hosts pr. subnet kan beregnes med $2^h - 2$, hvor h er antallet af bits i host-delen. For /16 subnetmasken har vi 16 bits til hosts, så $2^{16} - 2 = 65534$ hosts pr. subnet.

4. Identificer subnet-ID'er:
   - Første subnet: `10.0.0.0/16` (hosts: `10.0.0.1` til `10.0.255.254`)
   - Andet subnet: `10.0.1.0/16` (hosts: `10.0.1.1` til `10.0.1.254`)
   - Tredje subnet: `10.0.2.0/16` (hosts: `10.0.2.1` til `10.0.2.254`)
   - Fortsæt for de resterende subnets: `10.0.3.0/16`, `10.0.4.0/16`, osv.

Eksempel 4: Subnetting et Class B-netværk til små undernet

Scenarie: En IT-afdeling har et Class B-netværk `172.16.0.0/16` og ønsker at opdele dette netværk i mindre subnets, hver med plads til 30 hosts.

Beregninger:

1. Bestem antallet af nødvendige subnets: For at have subnets med 30 hosts skal vi bruge en subnetmaske, der giver mindst 30 hosts pr. subnet.

2. Beregn subnetmasken:
   - Antallet af nødvendige bits for hosts kan beregnes med $2^h - 2 \geq 30$, hvor h er antallet af bits i host-delen. Så $h = 5$.
   - Den oprindelige subnetmaske er `255.255.0.0` (/16). Ved at låne 11 bits fra host-delen (fordi vi har 16 bits til hosts i et Class B-netværk, og vi skal efterlade 5 bits til hosts), bliver den nye subnetmaske `255.255.255.224` (/27).

3. Beregn antallet af subnets:
   - Antallet af subnets kan beregnes med 2^n, hvor n er antallet af lånte bits. For /27 subnetmasken har vi lånt 11 bits, så $2^{11} = 2048$ subnets.

4. Identificer subnet-ID'er:

    - Første subnet: `172.16.0.0/27` (hosts: `172.16.0.1` til `172.16.0.30`)
    - Andet subnet: `172.16.0.32/27` (hosts: `172.16.0.33` til `172.16.0.62`)
    - Tredje subnet: `172.16.0.64/27` (hosts: `172.16.0.65` til `172.16.0.94`)
    - Fortsæt for de resterende subnets: `172.16.0.96/27`, `172.16.0.128/27`, osv.

3.3 IP-adressering: Klassebaseret og Private vs Internet-unikke Adresser

IP-adressering er grundlaget for at identificere enheder i et netværk, både lokalt og globalt. Historisk set blev IP-adresser opdelt i faste klasser (A, B, C) baseret på de første bits i adressen. Dette system blev kaldt klassebaseret adressering.

3.3.1 Klassebaseret Adressering

Klassebaseret adressering inddeler IP-adresser i følgende klasser:

- **Klasse A:** Bruges til netværk med et meget stort antal værter. Adresserne spænder fra 1.0.0.0 til 126.0.0.0.
- **Klasse B:** Bruges til netværk med et mellemstort antal værter. Adresserne spænder fra 128.0.0.0 til 191.255.0.0.
- **Klasse C:** Bruges til netværk med et mindre antal værter. Adresserne spænder fra 192.0.0.0 til 223.255.255.0.

Dette system er nu stort set erstattet af klasseløs adressering (CIDR), som giver en mere fleksibel og effektiv udnyttelse af adressepladsen.

3.3.2 Private vs Internet-unikke IP-adresser

Inden for det klassebaserede system blev bestemte adresser reserveret som private IP-adresser, som kun bruges inden for lokale netværk og ikke er routable på internettet. Disse adresser er defineret i RFC 1918 og falder ind under følgende områder:

- **Klasse A:** 10.0.0.0 til 10.255.255.255
- **Klasse B:** 172.16.0.0 til 172.31.255.255
- **Klasse C:** 192.168.0.0 til 192.168.255.255

Private IP-adresser bruges typisk i lokale netværk, som f.eks. i hjem eller små kontorer, hvor enheder som computere og printere kommunikerer internt. En hjemme-router kan f.eks. bruge en privat IP-adresse som `192.168.1.1` for at give netværksadgang til enheder på det lokale netværk.

Offentlige eller internet-unikke IP-adresser, derimod, er routable på internettet og bruges til at identificere enheder globalt. Disse adresser er nødvendige for at kommunikere med enheder uden for det lokale netværk.

3.4 Classless Inter-Domain Routing (CIDR)

3.4.1 Redegørelse

CIDR blev introduceret i 1993 som en metode til at forbedre effektiviteten og fleksibiliteten af IP-adressetildelingen. I modsætning til den klassebaserede IP-adressetildeling, som begrænsede antallet af mulige netværk og undernet, tillader CIDR en mere nuanceret og effektiv fordeling af IP-adresser. CIDR bruger en teknik kendt som "subnetting", hvor en enkelt IP-adresse kan opdeles i flere mindre netværkssegmenter ved hjælp af en variabel subnetmaske. Dette reducerer spildet af IP-adresser og giver netværksadministratorer mulighed for at tilpasse størrelsen af netværk og undernet til specifikke behov.

3.4.2 Anvendelse

I praksis anvendes CIDR til at skabe mindre og mere håndterbare netværk, hvilket forbedrer netværkseffektiviteten og -sikkerheden. Det gør det muligt for netværksadministratorer at opdele et stort netværk i flere undernet, hvilket kan hjælpe med at organisere netværkstrafik og begrænse omfanget af netværksforstyrrelser. CIDR er også afgørende for routing, da det reducerer størrelsen af routingtabeller i internettets backbone-routere, hvilket gør internettet mere skalerbart og effektivt.

3.4.3 Eksempel

En adresse som `192.168.0.0/23` dækker adresserne
`192.168.0.0` til `192.168.1.255`, hvilket giver 512 adresser. Ved at bruge CIDR kan netværksadministratorer tilpasse subnetmasken efter behovene for specifikke netværkssegmenter.

3.4.4 Analyse

Effektiviteten af CIDR kan ses i dets evne til at forlænge levetiden af IPv4-adresserummet. Uden CIDR ville IPv4-adresserummet være blevet udtømt meget hurtigere. Denne metode har også haft en afgørende betydning for

udviklingen af internettet, da det har gjort det muligt at håndtere en eksponentiel stigning i antallet af netværksenheder. Dog introducerer CIDR kompleksitet i netværksdesign og -forvaltning, hvilket kræver en mere dybtgående forståelse af IP-netværk og subnetting.

3.4.5 Perspektivering

Med overgangen til IPv6, hvor der er et langt større antal tilgængelige adresser, forbliver principperne bag CIDR relevante. IPv6-adressering indebærer en lignende logik for subnetting og effektiv adresseallokering, selvom den implementerer det på en anden måde på grund af IPv6-adressernes størrelse. Derudover understreger CIDR's succes nødvendigheden af kontinuerlig innovation i netværksteknologier for at imødekomme de stadigt skiftende krav til internettet og digital kommunikation.

3.5 IPv4 Header-struktur

IPv4 Header			
Ver	**IHL**	**Type of Service**	**Total Length**
Identification		**Flags**	**Fragment Offset**
Time to Live	**Protocol**	**Header Checksum**	
Source Address			
Destination Address			
Options + Padding (if any)			

Tabel 3.1: IPv4 Header-struktur

IPv4-headeren indeholder vigtige oplysninger for routing og levering af pakker. Den består af flere felter, herunder:

- **Version (Ver):** 4 bits, som angiver versionen af IP-protokollen. I dette tilfælde er det version 4.
- **Header længde (IHL):** 4 bits, som angiver længden af IP-headeren i 32-bit ord. Minimum værdien er 5, hvilket repræsenterer 20 bytes.
- **Type af service (ToS):** 8 bits, som indikerer parametrene for den ønskede servicekvalitet, såsom forsinkelse, gennemstrømning og pålidelighed.
- **Total længde:** 16 bits, som angiver den totale længde af datagrammet, inklusive header og data. Maksimum længden er 65.535 bytes.
- **Identifikation:** 16 bits, som identificerer hvert datagram unikt. Det er nyttigt ved fragmentering for at identificere og genopbygge datagrammer.

- **Flags og fragment offset:** 16 bits, som indeholder 3 flags og 13-bit fragment offset. Flags inkluderer DF (Don't Fragment) og MF (More Fragments).
- **TTL (Time to Live):** 8 bits, som angiver levetiden for datagrammet i sekunder eller hop, før det kasseres.
- **Protokol:** 8 bits, som angiver den næste protokol, som datagrammet skal leveres til, f.eks. TCP eller UDP.
- **Header checksum:** 16 bits, som bruges til at kontrollere headerens integritet.
- **Kilde IP-adresse:** 32 bits, som angiver afsenderens IP-adresse.
- **Destination IP-adresse:** 32 bits, som angiver modtagerens IP-adresse.

Detaljeret beskrivelse af felter

- **Type af service (ToS):** ToS-feltet består af et 3-bit præcedensfelt, som angiver prioriteten af pakken, og 4 bits til specifikke serviceparametre som forsinkelse, gennemstrømning og pålidelighed. Præcedensfeltet bruges til at indikere vigtigheden af datagrammet, mens de resterende 4 bits kan justeres for at minimere forsinkelse, maksimere gennemstrømning, maksimere pålidelighed eller minimere omkostninger.
- **Flags:** Der er to flags i headeren:
  - **DF (Don't Fragment):** Hvis dette flag er sat, må datagrammet ikke fragmenteres. Hvis netværket kræver fragmentering og flaget er sat, vil datagrammet blive droppet.
  - **MF (More Fragments):** Hvis dette flag er sat, indikerer det, at der er flere fragmenter efter dette. Hvis flaget ikke er sat, er dette det sidste fragment.
- **Fragment offset:** Dette felt angiver positionen af fragmentet i det oprindelige datagram. Offset måles i enheder af 8 bytes. Det bruges til at samle fragmenterne korrekt ved destinationen.
- **Time to Live (TTL):** TTL-feltet forhindrer datagrammer i at cirkulere evigt ved at reducere TTL-værdien med én for hver router det passerer. Når værdien når 0, kasseres datagrammet. Dette hjælper med at undgå uendelige loop i netværket.
- **Header checksum:** Dette felt bruges til at kontrollere headerens integritet. Det beregnes ved at opdele headeren i 16-bit ord, summere

dem, og tage en komplement addition. Denne værdi verificeres ved hver router for at sikre, at headeren ikke er blevet korrumperet under transmissionen.

3.5.1 Pakke fragmentering

Pakke fragmentering opdeler store IP-pakker i mindre fragmenter for at tilpasse dem til netværkets MTU (Maximum Transmission Unit). Fragmenterne samles igen ved destinationen for at gendanne den oprindelige pakke.

3.6 DHCP (Dynamic Host Configuration Protocol) i Industrielle Netværk

Dynamic Host Configuration Protocol (DHCP) er en netværksprotokol, der typisk anvendes til automatisk tildeling af IP-adresser til enheder som computere og printere i et netværk. I industrielle miljøer er det dog almindeligt at anvende statisk IP-tildeling for kritisk udstyr som PLC'er, sensorer og aktuatorer. Dette skyldes behovet for at sikre, at disse enheder altid har en fast IP-adresse, hvilket er afgørende for stabiliteten og forudsigeligheden af netværkskommunikationen.

DHCP arbejder ved at bruge en klient-server model, hvor en DHCP-klient (en netværksenhed) anmoder om en IP-adresse fra en DHCP-server. Serveren tildeler en IP-adresse fra en pulje af adresser og sender denne information tilbage til klienten sammen med andre konfigurationsparametre som subnetmaske, gateway-adresse og DNS-servere. Denne proces involverer flere trin:

1. **DHCP Discover:** Klienten sender en broadcast-anmodning for at finde tilgængelige DHCP-servere.

2. **DHCP Offer:** Serverne svarer med et tilbud, der inkluderer en IP-adresse og konfigurationsparametre.

3. **DHCP Request:** Klienten vælger en af de modtagne tilbud og sender en anmodning om at bruge den tilbudte adresse.

4. **DHCP Acknowledgement:** Den valgte server bekræfter tildelingen, og klienten kan nu bruge IP-adressen.

Selvom DHCP ikke traditionelt anvendes til kritiske industrielle enheder, kan det finde anvendelse i mindre kritiske dele af det industrielle netværk, såsom kontorområder, hvor computere og andre ikke-kritiske enheder er placeret. DHCP kan også bruges til hurtig og midlertidig konfiguration af testudstyr eller mobile enheder, der ikke kræver faste IP-adresser.

3.6.1 Praktiske Anvendelser

- **Automatiseret Enhedskonfiguration i Ikke-Kritiske Områder:** I administrative eller supportområder af en industriel facilitet, kan DHCP bruges til automatisk IP-tildeling for computere, printere og mobile enheder.

- **Test og Udviklingsmiljøer:** I testopsætninger, hvor udstyr konstant ændres og omkonfigureres, kan DHCP bruges til hurtigt at tildele IP-adresser til nye enheder uden behov for manuel konfiguration.

- **Mobile og Midlertidige Systemer:** For mobile robotter eller midlertidige opstillinger i produktionen, sikrer DHCP, at disse enheder hurtigt kan integreres i netværket uden omfattende manuel konfiguration.

3.6.2 Konfigurationsinformation leveret af DHCP

Selvom den primære opgave for DHCP er at uddele IP-adresser og subnetmasker, leverer DHCP faktisk mere konfigurationsinformation end blot IP-adressen til sine klienter. Den ekstra konfigurationsinformation består af DHCP-optioner. Følgende er en detaljeret beskrivelse af nogle almindelige DHCP-optioner, der kan konfigureres af serveren:

- **Router-adressen (Default Gateway-adressen):** Denne option specificerer IP-adressen på den router, der fungerer som standardgateway for klienterne. Standardgatewayen er den enhed, der videresender trafik fra klientens lokale netværk til andre netværk eller internettet. Uden denne information ville klienterne ikke kunne kommunikere uden for deres eget subnet.

- **Udløbstid for konfigurationsinformationen:** Denne option, også kendt som lease-tiden, angiver den tid, som en klient må bruge den tildelte IP-adresse, før den skal forny lejen. En kortere lease-tid kan være nyttig i miljøer, hvor enheder ofte skifter, mens en længere lease-tid kan reducere netværksbelastningen ved færre fornyelser.

- **Domænenavn:** Denne option gør det muligt at specificere et domænenavn, som klienterne skal bruge som deres søgedomæne. Dette domænenavn føjes automatisk til alle ikke-fuldstændige domænenavne, som en klient forsøger at løse, hvilket hjælper med at forenkle DNS-opslag inden for det lokale netværk.

- **Domænenavneserver (DNS) serveradresse:** Denne option specificerer IP-adressen på en eller flere DNS-servere, som klienterne skal bruge til at løse domænenavne til IP-adresser. DNS-servere er afgørende

for netværkskommunikation, da de tillader brugere og applikationer at anvende menneskeligt læsbare navne i stedet for at huske IP-adresser.

- **Windows Internet Name Service (WINS) serveradresse:** WINS bruges til at løse NetBIOS-navne til IP-adresser, især i ældre Windows-netværk. Denne option specificerer IP-adressen på en eller flere WINS-servere, som klienterne skal bruge. Selvom WINS er blevet mindre relevant med fremkomsten af DNS, kan det stadig være nyttigt i visse legacy-systemer og applikationer.

Disse DHCP-optioner gør det muligt for netværksadministratorer at levere en bred vifte af netværkskonfigurationsoplysninger automatisk, hvilket reducerer behovet for manuel konfiguration og sikrer, at klienterne har de nødvendige oplysninger til korrekt netværksfunktion. Ved at bruge DHCP-optioner kan administratorer centralisere og forenkle administrationen af netværksindstillinger, hvilket forbedrer netværkets effektivitet og pålidelighed.

3.6.3 DHCP-servere

En DHCP-server kan være en servercomputer placeret på TCP/IP-netværket. Alle moderne serveroperativsystemer har en indbygget DHCP-server. For at opsætte DHCP på en netværksserver skal du blot aktivere serverens DHCP-funktion og konfigurere dens indstillinger. Servere, der kører DHCP, behøver ikke udelukkende at være dedikeret til DHCP, medmindre netværket er meget stort.

Mange multifunktionsroutere har også indbyggede DHCP-servere. Hvis du ikke ønsker at belaste en af dine netværksservere med DHCP-funktionen, kan du aktivere routerens indbyggede DHCP-server.

3.6.4 Sådan konfigureres DHCP på Windows 11

For at konfigurere en DHCP-server på Windows 11:

1. Åbn *Indstillinger* og naviger til *Netværk og internet*.
2. Klik på *Egenskaber* for den netværksforbindelse, du vil konfigurere.
3. Rul ned til *IP-indstillinger*, og klik på *Rediger*.
4. Vælg *Automatisk (DHCP)* under *IPv4* eller *IPv6* afhængig af din netværkskonfiguration.
5. Klik på *Gem* for at anvende ændringerne.
6. For mere avancerede indstillinger kan du bruge *PowerShell* eller *Kommandoprompt* til at konfigurere DHCP-servere ved at bruge kommandoer som `netsh dhcp server`.

3.6.5 Scopes og Lejevarighed

Et scope er simpelthen et område af IP-adresser, som en DHCP-server er konfigureret til at uddele. Du skal oprette et scope, før du kan aktivere en DHCP-server. Når du opretter et scope, kan du specificere flere egenskaber som scope-navn, beskrivelse, start- og slut-IP-adresser, subnetmaske og udløbstid. Lejevarigheden angiver, hvor længe værten har lov til at bruge IP-adressen. Værten forsøger at forny lejen, når halvdelen af lejevarigheden er gået.

3.7 DNS (Domain Name System) i Industrielle Netværk

Domain Name System (DNS) spiller en kritisk rolle i industrielle netværk ved at muliggøre navnebaseret routing af netværkstrafik. DNS oversætter menneskeligt læsbare domænenavne til IP-adresser, hvilket gør det lettere at administrere og tilgå enheder og tjenester i et komplekst netværk. I industrielle miljøer anvendes DNS ofte til at styre adgang til SCADA-systemer, HMI'er, og forskellige servere, hvilket muliggør en mere intuitiv og effektiv netværksnavigation.

DNS fungerer ved hjælp af en hierarkisk og distribueret database, der består af flere niveauer af domæner. Hvert domæneniveau administreres af en autoritativ DNS-server. Når en klient sender en DNS-forespørgsel, følger processen typisk disse trin:

1. **Forespørgsel til Resolver:** Klienten sender en forespørgsel til en lokal DNS-resolver (oftest en del af netværkets infrastruktur).
2. **Kontakt til Root Server:** Resolveren kontakter en root server for at finde den autoritative server for top-level domænet (TLD).
3. **Kontakt til TLD Server:** Root serveren svarer med adressen på TLD serveren, som resolveren derefter kontakter.
4. **Kontakt til Autoritativ Server:** TLD serveren svarer med adressen på den autoritative server for det ønskede domæne, som resolveren kontakter for at få den endelige IP-adresse.
5. **Svar til Klienten:** Resolveren sender den fundne IP-adresse tilbage til klienten, som derefter kan kontakte den ønskede tjeneste.

Ved at anvende interne DNS-servere kan industrielle netværk opretholde sikkerhed og hastighed i navneopslaget. Dette er afgørende i produktionsmiljøer, hvor forsinkelse og nedetid kan føre til betydelige økonomiske tab. Interne DNS-servere kan også tilpasses til at håndtere specifikke industrielle

domæner og underdomæner, hvilket forbedrer netværksorganiseringen og gør det lettere at lokalisere og kommunikere med kritiske enheder og systemer. Kombinationen af statisk IP-tildeling og DNS i industrielle netværk skaber en mere robust og fleksibel infrastruktur, der kan understøtte avancerede automatiseringsprocesser og IoT-enheder.

3.7.1 Praktiske Anvendelser

- **Enhedshåndtering:** DNS gør det lettere at finde og administrere netværkskomponenter ved hjælp af letforståelige navne i stedet for komplekse IP-adresser. For eksempel kan en PLC nås via et navn som `plc1.factory.local` i stedet for en IP-adresse.
- **Fjerndiagnostik og Overvågning:** Teknikere kan bruge DNS-navne til at tilgå og overvåge enheder eksternt, hvilket forenkler fejlfinding og vedligeholdelse.
- **Sikkerhedsforbedringer:** Ved at bruge DNSSEC (DNS Security Extensions) kan industrielle netværk implementere ekstra sikkerhedsforanstaltninger, der beskytter mod DNS-forfalskning og andre typer netværksangreb.

3.7.2 Sådan konfigureres DNS på Windows 11

For at konfigurere en DNS-server på Windows 11:

1. Åbn *Indstillinger* og naviger til *Netværk og internet*.
2. Klik på *Egenskaber* for den netværksforbindelse, du vil konfigurere.
3. Rul ned til *DNS-serverindstillinger*, og klik på *Rediger*.
4. Vælg *Manuel* under *DNS-indstillinger*.
5. Indtast de ønskede primære og sekundære DNS-serveradresser.
6. Klik på *Gem* for at anvende ændringerne.
7. For mere avancerede indstillinger kan du bruge *PowerShell* eller *Kommandoprompt* til at konfigurere DNS-servere ved at bruge kommandoer som `netsh dns add`.

3.8 NAT (Network Address Translation)

Mål: At forstå og implementere NAT for at forbedre netværksfleksibilitet og sikkerhed ved at skjule interne IP-adresser og tillade flere enheder at dele en enkelt offentlig IP-adresse.
Afsnit:

1. Introduktion til NAT: Grundlæggende principper og formål med NAT.

2. Typer af NAT: Beskrivelse af forskellige typer NAT som Static NAT, Dynamic NAT og PAT (Port Address Translation).

3. Konfiguration af NAT på netværksenheder: Trin-for-trin vejledning til opsætning af NAT på en router.

4. Fordele og Ulemper ved NAT: Diskussion af fordelene ved at bruge NAT, såsom forbedret sikkerhed og IP-adresseringsfleksibilitet, samt potentielle ulemper som latency og kompleksitet.

5. Praktiske Anvendelser af NAT: Eksempler på brug af NAT i industrielle netværk.

3.9 VLAN (Virtual Local Area Network)

Et Virtual Local Area Network (VLAN) er en metode til at segmentere et fysisk netværk logisk i mindre, isolerede netværk. VLAN gør det muligt at opdele en fysisk netværksinfrastruktur i flere virtuelle netværk, hvilket forbedrer netværksadministration, sikkerhed og ydeevne.

3.9.1 Teori og Funktioner af VLAN

VLAN tillader netværksadministratorer at oprette separate logiske netværk inden for en enkelt fysisk netværksinfrastruktur. Dette opnås ved at tildele specifikke switch-porte til forskellige VLAN'er. Enheder på samme VLAN kan kommunikere direkte med hinanden, mens kommunikation mellem forskellige VLAN'er kræver routing via en router eller en Layer 3-switch.

Analogier for at forstå VLAN:

- **Kontorbygning:** Tænk på et VLAN som etageplaner i en stor kontorbygning. Selvom alle etagerne er en del af den samme bygning, kan hver etage betragtes som et separat kontorområde (VLAN). Mennesker på samme etage kan nemt kommunikere med hinanden, men for at kommunikere med nogen på en anden etage skal de bruge en elevator (router eller Layer 3-switch) for at nå derhen.

- **Afdelinger i en virksomhed:** En anden analogi er at sammenligne VLAN'er med forskellige afdelinger i en virksomhed. Selvom alle afdelinger er en del af den samme virksomhed, fungerer de som separate enheder. Kommunikation inden for samme afdeling er direkte, men for at kommunikere med en anden afdeling, skal man gå gennem en central reception (router eller Layer 3-switch).

De vigtigste funktioner og fordele ved VLAN inkluderer:

- **Sikkerhed:** VLAN'er kan isolere følsomme data og systemer ved at adskille dem fra resten af netværket, hvilket reducerer risikoen for uautoriseret adgang.
- **Ydeevne:** VLAN'er reducerer broadcast-domæner, hvilket mindsker mængden af broadcast-trafik og forbedrer netværkets samlede ydeevne.
- **Fleksibilitet:** VLAN'er tillader netværksadministratorer at gruppere enheder logisk baseret på funktion eller afdeling, uanset deres fysiske placering i netværket.
- **Forenklet Administration:** VLAN'er gør det lettere at administrere netværkskonfigurationer, da ændringer kan foretages logisk uden at skulle ændre den fysiske kabling.

3.9.2 Hvordan VLAN virker

VLAN'er fungerer ved at tildele switch-porte til specifikke VLAN-ID'er. Når en enhed sender data, tagger switchen pakken med VLAN-ID'et, som angiver, hvilket VLAN den tilhører. Dette tag fjernes, når pakken når sin destination. VLAN-tags gør det muligt at adskille trafik fra forskellige VLAN'er og sikre, at kun enheder inden for samme VLAN kan kommunikere direkte.

Der er to typer af VLAN-konfigurationer:

- **Access VLAN:** En access-port er en switch-port, der er tildelt til et enkelt VLAN. Denne port forbinder typisk til en slutbruger-enhed som en computer eller printer.
- **Trunk VLAN:** En trunk-port er en switch-port, der kan transportere trafik fra flere VLAN'er. Trunk-porte forbinder normalt switches til andre switches eller routere og bruger VLAN-tagging for at adskille trafikken.

3.9.3 Spanning Tree Protocol (STP) og Dets Relevans i VLAN-miljøer

Spanning Tree Protocol (STP) er en netværksprotokol, der bruges til at forhindre loops i Ethernet-netværk med redundante forbindelser. I netværk, der implementerer VLAN'er, spiller STP en afgørende rolle ved at sikre, at der ikke opstår loops, hvilket kan skabe broadcast storms og nedbrud i netværket.

STP virker ved at identificere og deaktivere redundante stier i netværket, så der kun er én aktiv sti mellem enhver to netværksenheder. Dette er særligt

vigtigt i VLAN-miljøer, hvor flere switches er forbundet gennem trunk-links, og det er afgørende at opretholde en loop-fri topologi.
De vigtigste funktioner i STP i VLAN-miljøer inkluderer:

- **Forebyggelse af netværksloops:** STP sikrer, at kun en enkelt sti er aktiv mellem to punkter i netværket, hvilket forhindrer loops, der kan forårsage broadcast storms.

- **Automatisk tilpasning:** Hvis en aktiv sti fejler, kan STP automatisk aktivere en redundant sti for at sikre fortsat netværkskommunikation.

- **Integration med VLAN'er:** STP kan operere over trunk-links, som transporterer flere VLAN'er, og sikrer, at VLAN-trafikken kan flyde sikkert uden risiko for loops.

Aktivering af STP på en Switch

På de fleste Cisco-switches er STP som standard aktiveret, men det kan også aktiveres manuelt. For at aktivere STP på en switch, kan følgende kommandoer anvendes:

```
Switch> enable
Switch# configure terminal
Switch(config)# spanning-tree vlan <VLAN-ID>
```

Forklaring: Denne kommando aktiverer STP for et specifikt VLAN på switchen. Hvis du vil aktivere STP for alle VLAN'er, kan du bruge kommandoen `spanning-tree vlan 1-4094`, hvilket dækker alle mulige VLAN-ID'er.

Deaktivering af STP på en Switch

Selvom det generelt ikke anbefales at deaktivere STP, fordi det beskytter mod netværksloops, kan der være specifikke scenarier, hvor det er nødvendigt. For at deaktivere STP på en switch kan følgende kommandoer bruges:

```
Switch> enable
Switch# configure terminal
Switch(config)# no spanning-tree vlan <VLAN-ID>
```

Forklaring: Denne kommando deaktiverer STP for et specifikt VLAN på switchen. Hvis du vil deaktivere STP for alle VLAN'er, kan du bruge kommandoen `no spanning-tree vlan 1-4094`.

Opsummering af STP's Rolle i VLAN-miljøer

Ved at implementere STP i netværk med VLAN'er kan netværksadministratorer sikre en stabil og pålidelig drift, selv i komplekse miljøer med redundante forbindelser. STP beskytter mod potentielle netværksloops, som kan forstyrre netværkstrafikken, og ved at bruge de ovenstående kommandoer kan STP nemt aktiveres eller deaktiveres alt efter netværkets behov.

3.9.4 Implementering af VLAN for netværksadministration

I Cisco Packet Tracer kan du oprette og konfigurere VLANs på en switch ved hjælp af CLI. Her er trin-for-trin, hvordan du kan gøre dette:

1. **Opret VLAN:**
   - Åbn CLI på switchen i Packet Tracer.
   - Indtast `vlan database` for at gå ind i VLAN-konfigurationsmode.
   - Brug kommandoen `vlan <VLAN-ID>` for at oprette et nyt VLAN. For eksempel `vlan 10` for at oprette VLAN 10.
   - Indtast `name <VLAN-name>` for at tildele et navn til VLAN'et, f.eks. `name Administration`.
2. **Tildel porte til VLAN:**
   - Gå til interface-konfigurationsmode ved at indtaste `interface range <port-range>`, for eksempel `interface range fa0/1-2`.
   - Brug kommandoen `switchport mode access` for at indstille portene til access mode.
   - Tildel portene til et VLAN ved hjælp af `switchport access vlan <VLAN-ID>`, f.eks. `switchport access vlan 10`.
3. **Opsæt trunk-links:**
   - Gå til trunk portens interface-konfigurationsmode, f.eks. `interface fa0/24`.
   - Indstil porten til trunk mode ved at bruge kommandoen `switchport mode trunk`.
   - Specificer hvilke VLANs, der er tilladt på trunk-linket ved at bruge `switchport trunk allowed vlan <VLAN-ID-list>`, f.eks. `switchport trunk allowed vlan 10,20,30`.

3.9.5 Eksempel på implementering af VLAN i Cisco Packet Tracer

I Cisco Packet Tracer kan du oprette og konfigurere VLANs på en switch ved hjælp af CLI. Her er trin-for-trin, hvordan du kan gøre dette, inklusiv en forklaring af vigtige kommandoer.

Oprettelse af VLANs på Switch

Først skal VLAN'erne oprettes og konfigureres på switchen:

```
Switch> enable
Switch# configure terminal
Switch(config)# vlan 10
Switch(config-vlan)# name Kontor1
Switch(config-vlan)# exit
Switch(config)# vlan 20
Switch(config-vlan)# name Kontor2
Switch(config-vlan)# exit
Switch(config)# vlan 30
Switch(config-vlan)# name Kontor3
Switch(config-vlan)# exit
Switch(config)# vlan 40
Switch(config-vlan)# name Gæsteadgang
Switch(config-vlan)# exit
```

Forklaring: Kommandoerne ovenfor bruges til at oprette fire VLAN'er på switchen, der repræsenterer forskellige kontorer og gæsteadgang. VLAN 10, 20, 30, og 40 er oprettet, og hver er blevet tildelt et navn, som repræsenterer deres funktion.

Tildeling af Porte til VLANs på Switch

Derefter tildeles de relevante porte til de forskellige VLAN'er på switchen:

```
Switch(config)# interface range fa0/1-2
Switch(config-if-range)# switchport mode access
Switch(config-if-range)# switchport access vlan 10
Switch(config-if-range)# exit

Switch(config)# interface range fa0/3-4
Switch(config-if-range)# switchport mode access
Switch(config-if-range)# switchport access vlan 20
Switch(config-if-range)# exit

Switch(config)# interface range fa0/5-6
Switch(config-if-range)# switchport mode access
Switch(config-if-range)# switchport access vlan 30
Switch(config-if-range)# exit

Switch(config)# interface range fa0/7-8
Switch(config-if-range)# switchport mode access
```

```
Switch(config-if-range)# switchport access vlan 40
Switch(config-if-range)# exit
```

Forklaring: Disse kommandoer tildeler specifikke switch-porte til de VLAN'er, der blev oprettet tidligere. Hver gruppe af porte sættes i "access mode"og tildeles et VLAN. For eksempel tildeles porte fa0/1-2 til VLAN 10 (Kontor1).

Opsætning af Trunk Links på Switch

Derefter konfigureres trunk-linket på switchen, som forbinder til routeren:

```
Switch(config)# interface fa0/24
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan 10,20,30,40
Switch(config-if)# exit
```

Forklaring: En trunk port tillader trafik fra flere VLAN'er at passere gennem. Ovenstående kommandoer konfigurerer port fa0/24 som en trunk-port, der håndterer trafik fra VLAN 10, 20, 30, og 40.

Konfiguration af Router-on-a-Stick

Nu skal routeren konfigureres til at håndtere trafik mellem VLAN'erne ved hjælp af subinterfaces:

```
Router> enable
Router# configure terminal
Router(config)# interface fa0/0
Router(config-if)# no shutdown

Router(config-if)# interface fa0/0.10
Router(config-subif)# encapsulation dot1Q 10
Router(config-subif)# ip address 192.168.10.1 255.255.255.0
Router(config-subif)# exit

Router(config)# interface fa0/0.20
Router(config-subif)# encapsulation dot1Q 20
Router(config-subif)# ip address 192.168.20.1 255.255.255.0
Router(config-subif)# exit

Router(config)# interface fa0/0.30
Router(config-subif)# encapsulation dot1Q 30
Router(config-subif)# ip address 192.168.30.1 255.255.255.0
Router(config-subif)# exit
```

```
Router(config)# interface fa0/0.40
Router(config-subif)# encapsulation dot1Q 40
Router(config-subif)# ip address 192.168.40.1 255.255.255.0
Router(config-subif)# exit
```

Forklaring: Her bliver routeren konfigureret med subinterfaces for hvert VLAN. Kommandoen `encapsulation dot1Q` bruges til at identificere og tagge trafikken for hvert VLAN. Hver subinterface tildeles også en IP-adresse, der fungerer som gateway for det pågældende VLAN.

Beskrivelse af Kommandoer og Termer

Encapsulation dot1Q:
`Router(config-subif)# encapsulation dot1Q 10`
Encapsulation dot1Q er en kommando, der bruges til at konfigurere IEEE 802.1Q VLAN-tagging på en router-subinterface. Dette er nødvendigt for at identificere og adskille trafikken fra forskellige VLAN'er, som sendes over en trunk-port på en switch. Dot1Q står for 802.1Q, og tallet (f.eks. 10) refererer til VLAN ID'et.

Interface fa0/0:
`Router(config)# interface fa0/0`
Interface fa0/0 refererer til en specifik fysisk netværksinterface på routeren. Fa står for FastEthernet, og 0/0 angiver portnummeret. Denne kommando bruges til at arbejde direkte med konfigurationen af denne netværksport.

Subinterface Konfiguration:
`Router(config)# interface fa0/0.10`
En subinterface er en logisk opdeling af en fysisk interface på routeren. Ved at oprette separate subinterfaces (f.eks. fa0/0.10) kan en enkelt fysisk interface håndtere flere VLAN'er. Hver subinterface tildeles en IP-adresse, som fungerer som gateway for enhederne i det tilknyttede VLAN.

Trunk Link:
`Switch(config-if)# switchport mode trunk`
En trunk port kan håndtere trafik fra flere VLAN'er. Ved at konfigurere trunk-mode på en switch-port, kan du tillade trafik fra specifikke VLAN'er at passere gennem denne port.

Hvordan Det Hele Hænger Sammen

- **VLAN'er på Switchen:** Først opretter du VLAN'erne og tildeler specifikke porte på switchen til hvert VLAN.

- **Trunk Link:** Derefter opsætter du en trunk-port på switchen, som forbinder til routeren. Denne port kan håndtere trafik fra flere VLAN'er.

- **Router-on-a-Stick:** På routeren konfigurerer du subinterfaces på en enkelt fysisk interface (f.eks. fa0/0). Hver subinterface er forbundet med et VLAN via dot1Q-tagging, og tildeles en IP-adresse, som fungerer som gateway for det pågældende VLAN.

3.9.6 Praktiske Anvendelser af VLAN i Industrielle Netværk

VLAN'er er særligt nyttige i industrielle netværk, hvor sikkerhed, ydeevne og administrativ kontrol er afgørende. Her er nogle eksempler på, hvordan VLAN kan anvendes i industrielle omgivelser:

- **Segmentering af produktionslinjer:** Ved at opdele forskellige produktionslinjer i separate VLAN'er kan man sikre, at trafikken fra en produktionslinje ikke forstyrrer andre. Dette er især nyttigt, hvis forskellige produktionslinjer har forskellige netværkskrav eller sikkerhedsniveauer.

- **Adgangskontrol:** VLAN'er kan bruges til at begrænse adgangen til visse dele af netværket. For eksempel kan man oprette separate VLAN'er for gæster, kontorpersonale og produktionsudstyr, hvilket reducerer risikoen for uautoriseret adgang til kritiske systemer.

- **Forbedring af netværksadministration:** VLAN'er gør det lettere at administrere netværkets trafik og enheder, især i komplekse industrielle miljøer. Ved at gruppere lignende enheder i samme VLAN kan man forenkle fejlfinding og vedligeholdelse.

- **Styring af netværksprioritet:** Ved hjælp af VLAN-tagging kan man tildele forskellige prioriteter til forskellige typer af trafik. Dette sikrer, at vigtig trafik, såsom styringssignaler og realtidsdata, får højere prioritet end mindre kritisk trafik.

- **Isolering af IoT-enheder:** I industrielle miljøer, hvor mange IoT-enheder er tilsluttet netværket, kan VLAN'er bruges til at isolere disse enheder for at forbedre sikkerheden og ydeevnen. Dette forhindrer IoT-enheder i at interagere direkte med kritiske systemer og beskytter dem mod potentielle angreb.

Ved at anvende VLAN i industrielle netværk kan man opnå en høj grad af kontrol og sikkerhed, samtidig med at man optimerer netværkets ydeevne og administration. VLAN'er giver fleksibilitet til at tilpasse netværksstrukturen til specifikke behov og krav, hvilket er afgørende i komplekse og dynamiske industrielle miljøer.

3.10 Ethernet-kabling

Ethernet-kabling er en essentiel del af netværksinfrastrukturen, som muliggør dataoverførsel mellem forskellige enheder. Der er forskellige typer Ethernet-kabler, hver med specifikke egenskaber og anvendelser. Dette afsnit dækker de vigtigste aspekter ved Ethernet-kabling, herunder hvordan kabler fungerer som lavpasfiltre, teorien bag fiberoptik, og den generelle driftsteori.

3.10.1 Typer af Ethernet-kabler

De mest almindelige typer Ethernet-kabler inkluderer:

- **Twisted Pair Cables:** Disse kabler består af par af ledere, der er snoet sammen for at reducere elektromagnetisk interferens (EMI). Der er to hovedtyper af twisted pair kabler:
  - **Unshielded Twisted Pair (UTP):** Bruges hovedsageligt i kontormiljøer og mindre udsatte områder.
  - **Shielded Twisted Pair (STP):** Har en ekstra beskyttelse mod interferens, hvilket gør dem velegnede til industrielle miljøer med høj EMI.
- **Coaxial Cables:** Disse kabler har en indre leder omgivet af en isolerende lag og en afskærmning af flettet kobber. Coaxial kabler bruges ofte til ældre netværksteknologier og kabel-tv.
- **Fiber-optic Cables:** Disse kabler bruger lys til at overføre data og er ideelle til lange afstande og høje datahastigheder. Fiber-optic kabler er mindre modtagelige for EMI og har højere båndbredde sammenlignet med kobberkabler.

3.10.2 Struktureret Kabelføring

Struktureret kabelføring er en standardiseret metode til at designe og installere kablingssystemer i bygninger. Dette inkluderer planlægning af kabellayout, valg af kabler, og installation af kabelkanaler og netværksudstyr. Struktureret kabelføring sikrer en organiseret og skalerbar netværksinfrastruktur.

3.10.3 Ethernet-kabler i industrielle miljøer

I industrielle miljøer er det vigtigt at vælge Ethernet-kabler, der kan modstå hårde forhold såsom støj, vibrationer, ekstreme temperaturer og kemisk eksponering. Shielded twisted pair (STP) og fiber-optic kabler er ofte foretrukne valg i sådanne miljøer på grund af deres modstandsdygtighed over for EMI og deres pålidelighed.

3.10.4 Hvordan kabler fungerer som lavpasfilter

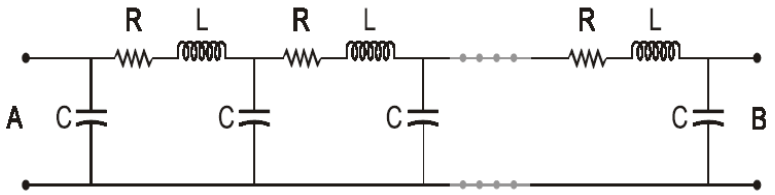
Ethernet-kabler fungerer som lavpasfiltre, der dæmper højfrekvente signaler og tillader lavfrekvente signaler at passere. Dette skyldes de induktive og kapacitive egenskaber ved kablerne. Modstanden (R), induktansen (L), og kapacitansen (C) i kablerne er distribueret langs hele længden af kablet og påvirker signalets kvalitet.

$$V_{\text{drop}} = I \times R$$

Hvor:

- V_{drop} er spændingsfaldet langs kablet.
- I er strømmen gennem kablet.
- R er modstanden i kablet.

Ved høje frekvenser kombineres modstand, induktans og kapacitans og præsenterer effekterne af et lavpasfilter. Dette kan illustreres ved følgende figur:



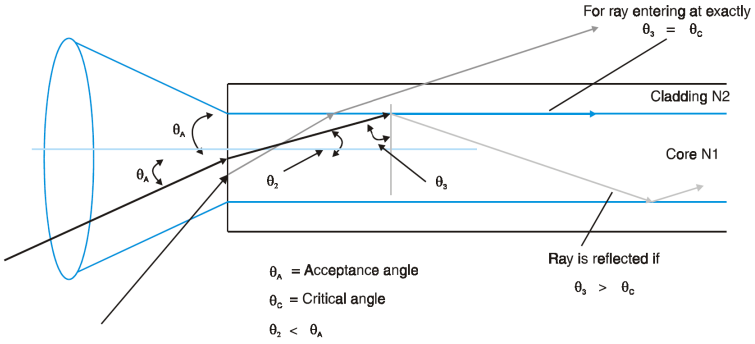
Figur 3.1: De vigtigste parametre for et datakommunikationskabel som lavpasfilter

3.11 Fiberoptik

Fiberoptiske kabler bruges normalt til at overføre digitale signaler over lange afstande med høj båndbredde. De har en kerne af rent optisk glas omgivet af en optisk kappe, der guider lysimpulser gennem kablet. Fiberoptik er mindre modtagelige for EMI og har en større informationsbærende kapacitet end kobberkabler.

3.11.1 Driftsteori for fiberoptik

Kommunikation over fiberoptiske kabler fungerer på princippet om, at lys bevæger sig gennem forskellige medier med forskellige hastigheder (på samme måde som radiobølger). Når lys bevæger sig fra et medium med en bestemt



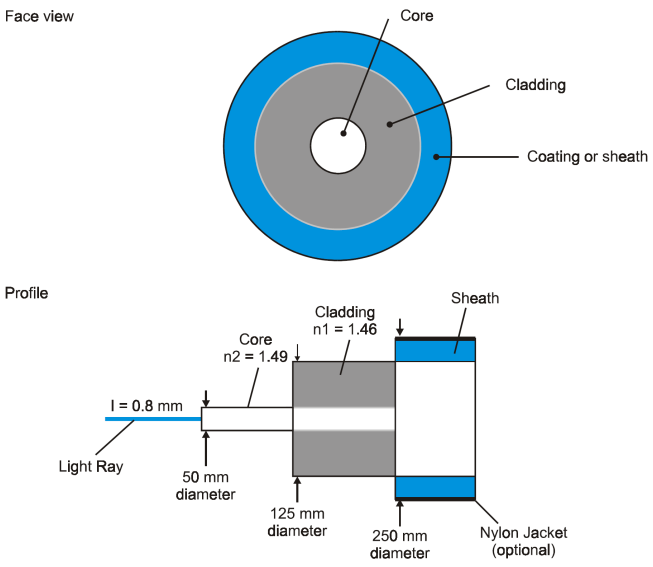
Figur 3.2: Opbygning af et fiberoptisk kabel

densitet til et andet med en anden densitet, ændrer lyset retning. Dette fænomen kaldes brydning.

$$n = \frac{\text{Lysets hastighed i vakuum}}{\text{Lysets hastighed i mediet}}$$

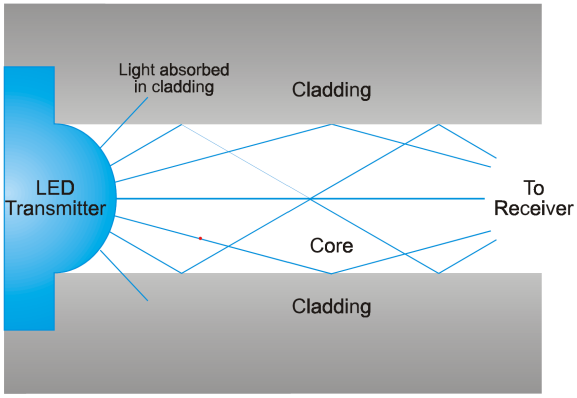
I et typisk fiberoptisk medium bevæger lyset sig ved cirka 2×10^8 m/s. Brydningsindekset er derfor:

$$n_1 = \frac{3 \times 10^8}{2 \times 10^8} = 1.5$$



Figur 3.3: Optisk fiber principper

Den optiske fiber fungerer som en bølgeleder (eller lysleder) for lysimpulser genereret af en lyskilde. Lyskilden er typisk en laserdiode eller lysdiode (LED), der opererer ved bølgelængder på 0.85, 1.3 eller 1.55 mikrometer.



Figur 3.4: LED-lyskilde koblet til en multimode fiber (Step Index)

3.11.2 Installation og vedligeholdelse

Når Ethernet-kabler installeres, er det vigtigt at følge producentens specifikationer og industristandarder for at sikre korrekt ydeevne. Dette inkluderer korrekt bøjning af kabler, undgåelse af skarpe vinkler, og sikring af gode forbindelser i stik og patch paneler.

Vedligeholdelse af Ethernet-kabler omfatter regelmæssig inspektion for skader, sikring af korrekte forbindelser og opdatering af kablingsinfrastrukturen efter behov.

3.11.3 Forbindelsestyper og Stik

En korrekt installation af Ethernet-kabler inkluderer også valg af passende stik og forbindelsestyper. De mest almindelige typer af stik omfatter:

- **RJ45-stik:** Det mest anvendte stik til Ethernet-forbindelser. RJ45-stik har otte ledere og bruges til at forbinde twisted pair kabler til netværksenheder som computere, routere og switches.
- **Andre stiktyper:** I specifikke applikationer kan andre stiktyper som RJ11, DB9, eller specialiserede industrielle stik være nødvendige.

3.11.4 Ydeevne og Hastigheder

Ydeevnen af Ethernet-kabler afhænger af kabeltypen og dens specifikationer:

- **Hastighedskategorier:** Forskellige kategorier af twisted pair kabler (f.eks. Cat5e, Cat6, Cat6a, Cat7) tilbyder varierende niveauer af ydeevne i forhold til båndbredde og hastighed.
- **Maksimal Kabellængde:** For hver kategori af kabel er der en maksimal anbefalet kabellængde, som påvirker netværkets ydeevne. For eksempel er maksimal længde for Cat5e og Cat6 kabler 100 meter.

3.11.5 Standarder og Protokoller

Ethernet-netværk styres af en række standarder og protokoller, der sikrer kompatibilitet og ydeevne:

- **IEEE 802.3:** Denne standard dækker de fleste aspekter af Ethernet-netværk og inkluderer specifikationer for forskellige hastigheder og medier.
- **Power over Ethernet (PoE):** PoE tillader samtidig overførsel af data og strøm via samme kabel, hvilket er nyttigt til enheder som IP-kameraer og trådløse adgangspunkter.

3.11.6 Fejlfinding og Vedligeholdelse

Regelmæssig vedligeholdelse og fejlfinding er afgørende for at opretholde et pålideligt Ethernet-netværk:

- **Typiske Problemer og Løsninger:** Almindelige problemer inkluderer kabelbrud, dårlige forbindelser og interferens. Korrekt terminering og regelmæssig inspektion kan afhjælpe mange af disse problemer.
- **Testudstyr:** Kabeltestere og certificeringsværktøjer er vigtige for at verificere kabelintegritet og ydeevne. Disse værktøjer kan identificere problemer som krydsede par, åben kredsløb, og interferens.

3.12 EMC (Elektromagnetisk Kompatibilitet)

Elektromagnetisk kompatibilitet (EMC) er afgørende for at sikre, at elektroniske enheder og systemer kan fungere korrekt uden at forstyrre eller blive forstyrret af elektromagnetiske felter. I forbindelse med Ethernet-kabling er EMC vigtigt for at forhindre interferens, som kan påvirke signalintegriteten og netværkets ydeevne.

- **Skærmning:** Shielded twisted pair (STP) kabler og korrekt jordforbindelse hjælper med at reducere EMI og forbedre EMC.
- **Kabelruting:** Undgå at placere Ethernet-kabler tæt på strømkabler eller andre kilder til elektromagnetisk støj.

- **Filter og Beskyttelse:** Brug af filtrerings- og beskyttelsesmekanismer kan hjælpe med at beskytte netværksenheder mod EMI.

3.12.1 Sikkerhed og ydeevne

Sikkerhed og ydeevne er afgørende faktorer ved installation af Ethernet-kabler i industrielle miljøer. Korrekt jordforbindelse og afskærmning skal implementeres for at beskytte mod EMI. Derudover skal kablerne være i stand til at opretholde datahastigheder og pålidelighed under alle driftsforhold.

Ved at følge disse retningslinjer kan en pålidelig og effektiv Ethernet-netværksinfrastruktur etableres, som kan modstå de udfordringer, der findes i industrielle miljøer.

Kapitel 4

Hardware Konfigurationer

4.1 Switch

4.1.1 Initial opsætning af en switch

4.1.2 Tildeling af IP-adresse til en switch

I netværksadministration er det ofte nødvendigt at tildele en IP-adresse til en switch for at kunne administrere den via fjernadgang, såsom Telnet eller SSH. En IP-adresse på en switch tildeles typisk til et virtuel interface, også kendt som SVI (Switched Virtual Interface). Dette interface fungerer som et logisk layer 3 interface, der repræsenterer en VLAN på switchen. Ved at tildele en IP-adresse til dette interface, kan du få adgang til switchens CLI (Command Line Interface) via netværket.

Når du tildeler en IP-adresse til en switch, er det vigtigt at sikre, at IP-adressen er i samme subnet som de andre enheder, der vil administrere switchen. Hvis du har flere VLANs konfigureret på switchen, kan du tildele forskellige IP-adresser til forskellige SVI'er for at administrere hver VLAN separat.

Step-by-Step: Tildeling af IP-adresse til en switch

Følg nedenstående trin for at tildele en IP-adresse til en switch:

1. **Log ind på switchens CLI:**
    - Brug en konsolforbindelse eller SSH for at logge ind på switchens CLI.
2. **Gå til global konfigurationsmodus:**
    - Indtast kommandoen `enable` for at få adgang til privilegeret EXEC-tilstand.

- Indtast `configure terminal` for at gå til global konfigurationsmodus.

3. **Vælg det interface, du vil tildele IP-adressen til:**
   - For at tildele en IP-adresse til VLAN 1 (som typisk er standard VLAN), indtast kommandoen `interface vlan 1`.
4. **Tildel IP-adressen og subnetmasken:**
   - Indtast kommandoen `ip address 192.168.1.2 255.255.255.0`, hvor `192.168.1.2` er den IP-adresse, du ønsker at tildele, og `255.255.255.0` er subnetmasken.
5. **Aktiver interfacet:**
   - For at aktivere VLAN interfacet, indtast kommandoen `no shutdown`.
6. **Gem konfigurationen:**
   - Indtast `end` for at vende tilbage til privilegeret EXEC-tilstand.
   - Brug kommandoen `write memory` eller `copy running-config startup-config` for at gemme konfigurationen permanent.
7. **Verificér konfigurationen:**
   - Brug kommandoen `show ip interface brief` for at verificere, at IP-adressen er blevet tildelt korrekt til interfacet.

Med disse trin har du nu tildelt en IP-adresse til din switch, hvilket muliggør administration af switchen via netværket.

4.1.3 VLAN (Virtual LAN)

- Oprettelse og konfiguration af VLANs
- Trunking og inter-VLAN routing
- VLAN Tagging og Untagging

4.1.4 Spanning Tree Protocol (STP)

- Aktivering og konfiguration af STP
- Forståelse af STP's rolle i netværkssikkerhed
- Rapid Spanning Tree Protocol (RSTP)

4.1.5 Port Sikkerhed

- Konfiguration af port security
- Mac-adresse filtrering
- Beskyttelse mod MAC spoofing

4.1.6 EtherChannel

- Oprettelse og konfiguration af EtherChannel
- Forståelse af Load Balancing

4.1.7 Access Control Lists (ACL) på en Switch

- Konfiguration af Layer 2 ACLs
- Filtrering af trafik baseret på MAC-adresser

4.1.8 Quality of Service (QoS)

- Implementering af QoS på en switch
- Prioritering af netværkstrafik

4.1.9 Multicast Routing

- Implementering af IGMP Snooping
- Konfiguration af Multicast VLAN

4.1.10 Switch Sikkerhed

- Forhindre rogue DHCP servers (DHCP Snooping)
- Dynamic ARP Inspection (DAI)
- IP Source Guard

4.1.11 Monitoring og Fejlfinding

- Brug af SNMP til overvågning
- Port Mirroring og SPAN (Switch Port Analyzer)
- Logning og overvågning af netværkstrafik

4.2 Router

4.2.1 VPN (Virtual Private Network)

Introduktion til VPN

- Hvad er en VPN, og hvorfor er det vigtigt i netværkssikkerhed?
- Grundlæggende begreber: VPN tunneling, kryptering, og autentificering.

Site-to-Site VPN

- Konfiguration af Site-to-Site VPN mellem to netværk.
- Anvendelse af IPsec til at sikre forbindelsen.
- Praktisk opsætning af en VPN-tunnel mellem to routere.

Remote Access VPN

- Opsætning af Remote Access VPN til fjernarbejdere.
- Konfiguration af SSL VPN og brug af VPN-klienter.
- Brug af autentificering og autorisation for sikre forbindelser.

VPN Protocols

- Sammenligning af forskellige VPN-protokoller: PPTP, L2TP, IPsec, OpenVPN, SSL VPN.
- Fordele og ulemper ved hver protokol.
- Anvendelsesområder for de forskellige protokoller.

NAT Traversal i VPN

- Håndtering af VPN-trafik gennem en NAT-router.
- Konfiguration af NAT Traversal (NAT-T) for VPN-forbindelser.
- Problemer og løsninger relateret til NAT i VPN-scenarier.

IPsec VPN

- Grundlæggende IPsec-konfiguration.
- Faser af IPsec: ISAKMP/IKE og fase 1 og 2.
- Opsætning af krypterings- og autentificeringspolitikker.

SSL VPN

- Konfiguration af SSL VPN for sikre webbaserede forbindelser.
- Brug af certificater og HTTPS til sikring af VPN-forbindelser.
- Praktisk implementering af SSL VPN på routere.

VPN Failover

- Konfiguration af VPN med failover-mekanisme for høj tilgængelighed.
- Implementering af redundans med flere VPN-tunneler.
- Overvågning og fejlfinding af VPN failover-situationer.

VPN Sikkerhed

- Best practices for at sikre VPN-forbindelser.
- Brug af multifaktor-autentificering (MFA) i VPN.
- Logging og overvågning af VPN-aktivitet for sikkerhed.

Fejlfinding af VPN

- Diagnostik og fejlfinding af VPN-forbindelser.
- Brug af kommandoer som `show crypto isakmp`, `show crypto ipsec`, og `debug` til VPN-fejlfinding.
- Almindelige VPN-fejl og deres løsninger.

Advanced VPN Topics

- Site-to-Site VPN med dynamiske IP-adresser.
- Kombination af VPN med QoS for optimering af trafik.
- Integration af VPN med cloud-tjenester.

Eksempler på VPN-anvendelse i industrien

- Brug af VPN til at forbinde fjerntliggende industrielle anlæg.
- Implementering af VPN i SCADA-systemer for sikker dataoverførsel.
- VPN til sikring af IoT-enheder i industrielle netværk.

Del III

Industrielt Netværk

Kapitel 5

Introduktion til Industrielle Netværk

5.1 Hvad er Industrielt Netværk?

Industrielle netværk refererer til kommunikationssystemer designet specifikt til at forbinde enheder i industrielle miljøer som fabrikker, produktionsanlæg og andre automatiserede systemer. Disse netværk muliggør dataudveksling mellem forskellige maskiner og kontrolsystemer, hvilket optimerer produktionsprocesser, forbedrer effektiviteten og øger pålideligheden.

5.1.1 Historisk Udvikling af Netværk i Industrien

Industrielle netværk har gennemgået en betydelig udvikling over tid. I begyndelsen blev industrielle kontrolsystemer typisk isolerede og brugte proprietære kommunikationsprotokoller. Med fremkomsten af standardiserede protokoller og netværksteknologier som Ethernet og TCP/IP, er det blevet muligt at integrere industrielle netværk med virksomhedens IT-systemer, hvilket har ført til mere sammenhængende og fleksible produktionsmiljøer.

5.1.2 Tidlige Industrielle Netværk

I de tidlige dage af industriel automatisering var kommunikationssystemer ofte punkt-til-punkt forbindelser, hvor hver enhed var direkte forbundet med kontrolsystemet. Dette resulterede i komplekse og dyre installationer, der var svære at vedligeholde og udvide.

5.1.3 Overgang til Standardiserede Protokoller

Med introduktionen af standardiserede protokoller som Modbus og PROFIBUS i 1980'erne, blev det muligt at oprette mere fleksible og skalerbare

netværk. Disse protokoller gjorde det lettere at forbinde enheder fra forskellige producenter og reducere kompleksiteten i installationsprocessen.

5.1.4 Integration med Ethernet og IT-Systemer

I 1990'erne og 2000'erne begyndte industrielle netværk at integrere Ethernet-teknologi, hvilket gjorde det muligt at udnytte de højere hastigheder og den større båndbredde, som Ethernet tilbyder. Dette førte til udviklingen af industri-specifikke Ethernet-protokoller som EtherNet/IP, PROFINET og Modbus TCP. Integration med IT-systemer har desuden gjort det muligt for virksomheder at implementere Industri 4.0-konceptet, hvor produktionsdata kan analyseres i realtid for at optimere driften.

5.1.5 Nutidige og Fremtidige Trends

I dag fortsætter industrielle netværk med at udvikle sig med fokus på trådløse teknologier, cybersikkerhed og Internet of Things (IoT). Fremtidige trends omfatter brugen af 5G til industriel kommunikation, øget brug af kunstig intelligens til netværksovervågning og vedligeholdelse samt integration af avancerede sensorer og aktuatorer for at forbedre produktionsprocesser.

5.1.6 Betydningen af Industrielle Netværk

Industrielle netværk spiller en afgørende rolle i moderne produktionsmiljøer ved at:

- **Forbedre Effektiviteten:** Automatisering og realtidsdataudveksling mellem maskiner og kontrolsystemer optimerer produktionsprocesser og reducerer nedetid.
- **Øge Pålideligheden:** Standardiserede protokoller og robuste netværksdesigns sikrer pålidelig kommunikation selv i krævende industrielle miljøer.
- **Muliggøre Fleksibilitet:** Skalerbare netværk gør det nemmere at tilpasse sig ændringer i produktionskrav og integrere nye teknologier uden omfattende ændringer i infrastrukturen.
- **Styrke Sikkerheden:** Avancerede sikkerhedsprotokoller og netværksovervågning beskytter mod cybertrusler og uautoriseret adgang.

Kapitel 6

Protokoller og Elektriske Standarder

6.1 Introduktion til Protokoller og Elektriske Standarder

I industrielle netværk spiller både netværksprotokoller og elektriske standarder en afgørende rolle for at sikre effektiv og pålidelig kommunikation. Selvom disse to begreber ofte nævnes sammen, tjener de forskellige formål og adresserer forskellige aspekter af datakommunikation. Det er derfor vigtigt at forstå deres forskelle og hvordan de komplementerer hinanden.

6.2 Netværksprotokoller: Reglerne for Kommunikation

En netværksprotokol kan sammenlignes med et sprog eller en samtaleetikette, som alle deltagere i en samtale skal følge for at kunne forstå hinanden. Ligesom mennesker har brug for et fælles sprog for at kommunikere effektivt, kræver netværksenheder en fælles protokol for at udveksle data korrekt. En netværksprotokol definerer de regler og konventioner, som netværksenheder skal følge, herunder hvordan data pakkes, adresseres, transmitteres, modtages og behandles.

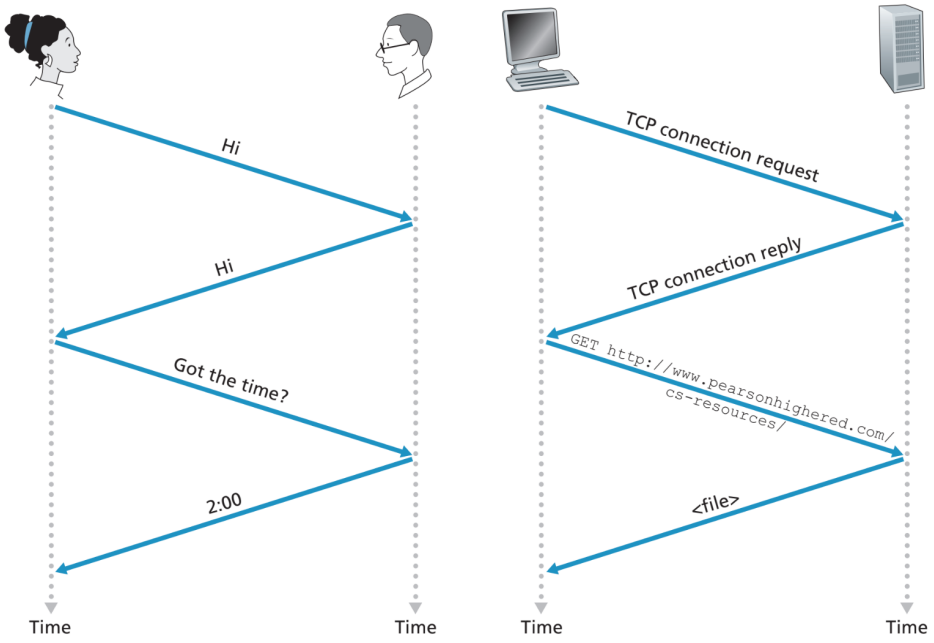
6.2.1 Analogier for at Forstå Netværksprotokoller

For at forstå netværksprotokoller kan vi bruge analogien med en samtale mellem mennesker. Tænk på netværksprotokollen som de regler, der dikterer, hvordan en samtale skal foregå for at sikre, at begge parter forstår hinanden:

- **Start af Kommunikation:** Ligesom en samtale begynder med en hilsen, starter en netværksforbindelse ofte med en `TCP connection`

`request`, hvor en enhed anmoder om at etablere en forbindelse. Modparten svarer med en `TCP connection reply` for at bekræfte.

- **Udveksling af Information:** Under samtalen udveksler parterne information, som spørgsmål og svar, f.eks. "Got the time?"og "2:00". I en netværkskommunikation kunne dette svare til udveksling af data, hvor en enhed sender en kommando, og den anden enhed reagerer med den ønskede information.
- **Afslutning:** Når samtalen er færdig, kan den ene person sige "farvel,"ligesom en netværksforbindelse afsluttes ved, at begge parter signalerer, at de er færdige med kommunikationen.



Figur 6.1: Sammenligning mellem en menneskelig samtale og en TCP-forbindelse

Billedet i figur 6.1 illustrerer, hvordan en sådan kommunikation foregår, både mellem mennesker og i en netværksprotokol som TCP. Denne visuelle analogi hjælper med at forstå, hvordan protokoller sikrer, at begge parter i en netværkskommunikation forstår hinanden korrekt og pålideligt.

Eksempler på almindelige netværksprotokoller i industrielle netværk inkluderer Ethernet/IP, Profinet, og Modbus TCP. Disse protokoller muliggør pålidelig kommunikation mellem forskellige enheder såsom PLC'er, sensorer, og aktuatorer i et automatiseret miljø.

6.3 Elektriske Standarder: Den Fysiske Infrastruktur

Mens netværksprotokoller styrer, hvordan data kommunikeres, beskriver elektriske standarder de fysiske egenskaber og specifikationer for, hvordan data transmitteres over et medie. Dette inkluderer alt fra typen af kabler og stik, til spændingsniveauer og frekvenser, der anvendes under transmissionen.

6.3.1 Analogier for at Forstå Elektriske Standarder

Vi kan forstå elektriske standarder ved at sammenligne dem med jernbanespor:

- **Sporbredde:** Ligesom tog skal køre på skinner med en bestemt bredde, skal data overføres gennem kabler med specifikke fysiske egenskaber. Standarderne specificerer, hvilken type kabler (f.eks. kobber eller fiberoptik) og stik der skal bruges.
- **Sikkerhed og Pålidelighed:** Elektriske standarder sikrer, at dataoverførslen er sikker og pålidelig ved at specificere grænser for elektrisk støj og spænding. Dette kan sammenlignes med sikkerhedsforanstaltninger på jernbanen, der forhindrer ulykker og sikrer, at togene kører uden afbrydelser.
- **Båndbredde:** Ligesom jernbanesystemer har kapacitet til at transportere et bestemt antal tog per tidsenhed, dikterer elektriske standarder også, hvor hurtigt data kan overføres og hvor meget information der kan transporteres samtidigt.

Eksempler på elektriske standarder omfatter RS-232, RS-485, og IEEE 802.3 (Ethernet-standarder). Disse standarder definerer de fysiske parametre, såsom kabeltype og stik, der er nødvendige for at sikre kompatibilitet og pålidelighed i datatransmissionen.

6.4 Eksempel på Samspil mellem Protokoller og Elektriske Standarder

Lad os overveje en situation, hvor en industriel robotarm kommunikerer med en kontrolstation over et netværk. Netværksprotokollen, såsom Profinet, styrer, hvordan data om robotarmens position og status sendes til kontrolstationen og tilbage. Den elektriske standard, såsom Ethernet (IEEE 802.3), bestemmer, hvilken type kabler og stik der bruges, samt de fysiske transmissionsparametre, der sikrer, at dataene overføres korrekt og pålideligt.

I denne situation fungerer netværksprotokollen som "sproget", som enhederne bruger til at "tale"med hinanden, mens den elektriske standard fungerer som "infrastrukturen", der gør det muligt for denne kommunikation at finde sted. Begge er nødvendige for at sikre en problemfri og effektiv drift af det industrielle netværk.

6.5 Opsummering: Forskellen mellem Protokoller og Elektriske Standarder

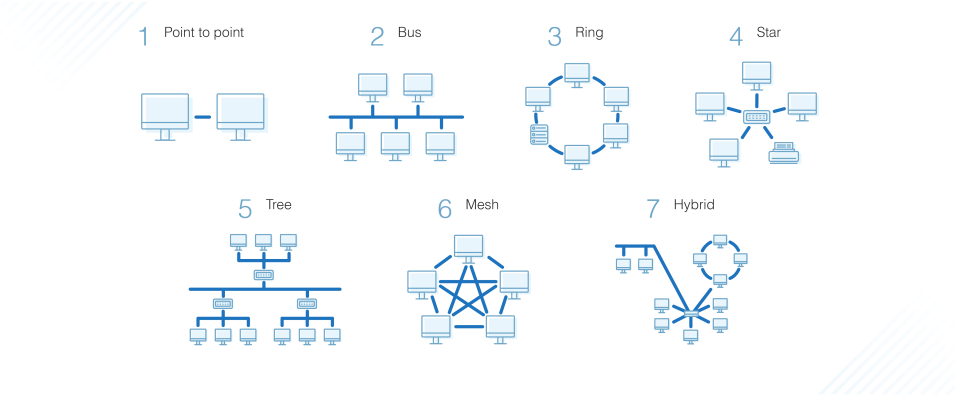
Selvom både netværksprotokoller og elektriske standarder er afgørende for effektiv datakommunikation i industrielle netværk, tjener de forskellige formål:

- **Netværksprotokoller:** Disse styrer, hvordan data pakkes, sendes, modtages, og behandles mellem enheder på et netværk. De skaber et fælles sprog, som alle enheder i netværket forstår, og sikrer, at informationen overføres korrekt.
- **Elektriske Standarder:** Disse definerer de fysiske parametre for datatransmissionen, herunder kabeltyper, spændingsniveauer og forbindelsesmåder. De sikrer, at de tekniske aspekter af dataoverførslen er sikre og pålidelige, hvilket muliggør kompatibilitet mellem forskellige enheder og systemer.

Protokollerne kan ses som de "regler"og "sprog", som enhederne i netværket bruger til at kommunikere, mens de elektriske standarder er den "infrastruktur", der muliggør, at denne kommunikation kan foregå sikkert og effektivt. For at opnå en robust industriel netværksløsning er det nødvendigt at anvende både passende protokoller og elektriske standarder i designet og implementeringen.

Kapitel 7

Netværkstopologi



Figur 7.1: Netværkstopologi

Netværkstopologi refererer til arrangementet og layoutet af noder og forbindelser i et netværk. Det beskriver både den fysiske og logiske struktur af netværket. Der er flere forskellige typer af topologier, hver med sine egne fordele og ulemper. Nedenfor gennemgås de mest almindelige netværkstopologier:

7.1 Punkt-til-punkt Topologi

I en punkt-til-punkt topologi er to noder direkte forbundet til hinanden. Denne type topologi bruges ofte i situationer, hvor en direkte forbindelse mellem to enheder er nødvendig.

- **Fordele:**
  - Meget enkel og hurtig forbindelse.
  - Lav latenstid og høj båndbredde.

- **Ulemper:**
  - Ikke skalerbar; kun egnet til små netværk.
  - Hvis forbindelsen fejler, går kommunikationen tabt.

7.2 Bustopologi

I en bustopologi er alle noder forbundet til en enkelt kommunikationslinje eller bus. Dataene, der sendes af en node, rejser langs bussen og kan modtages af alle noderne.

- **Fordele:**
  - Nem og billig at installere.
  - Kræver mindre kabel end en stjernetopologi.
- **Ulemper:**
  - Svært at fejlfinde og identificere problemer.
  - Begrænset kabel længde og antal noder.
  - Hvis hovedkabel fejler, går hele netværket ned.

7.3 Ringtopologi

I en ringtopologi er noderne forbundet i en cirkulær rækkefølge. Hver node har præcis to naboer, og data bevæger sig i én retning (eller i nogle tilfælde i begge retninger) langs ringen.

- **Fordele:**
  - Dataoverførsel er relativt hurtig, da data bevæger sig i en bestemt retning.
  - Ingen kollisionsdomæner som i bustopologi.
- **Ulemper:**
  - En fejl i en enkelt node eller forbindelse kan påvirke hele netværket.
  - Mere kompleks at installere og konfigurere.

7.4 Stjernetopologi

I en stjernetopologi er alle noder forbundet til en central hub eller switch. Hubben fungerer som en repeater for dataene, hvilket betyder, at data, der sendes fra en node, først sendes til hubben og derefter videre til den destination, der er tiltænkt.

- **Fordele:**
  - Enkel at installere og administrere.
  - Fejl i en enkelt kabel påvirker ikke resten af netværket.
  - Let at tilføje eller fjerne noder uden at forstyrre netværket.
- **Ulemper:**
  - Hvis central hub fejler, går hele netværket ned.
  - Kræver mere kabel end nogle andre topologier.

7.5 Trætopologi

Trætopologi, også kendt som hierarkisk topologi, er en hybrid topologi, der kombinerer egenskaberne af stjernetopologi og bustopologi. Den består af grupper af stjernetopologier, der er forbundet til et bus-lignende backbone-kabel.

- **Fordele:**
  - Udvidelsesvenlig og let at administrere.
  - Fejl i en enkelt node påvirker ikke hele netværket.
- **Ulemper:**
  - Mere kompleks at konfigurere end en simpel stjerne- eller bustopologi.
  - Backbone-kablet er et enkelt fejlpunkt.

7.6 Masketopologi

I en masketopologi er hver node forbundet til flere andre noder, hvilket skaber et netværk af forbindelser. Dette kan være en fuld mesh, hvor alle noder er forbundet til hinanden, eller en delvis mesh, hvor nogle noder kun er forbundet til nogle få andre noder.

- **Fordele:**

  - Meget pålidelig, da flere forbindelser sikrer, at data kan tage alternative ruter.
  - Fejl i en enkelt forbindelse påvirker ikke hele netværket.

- **Ulemper:**
  - Meget dyrt og komplekst at installere og vedligeholde.
  - Kræver mange kabler og porte.

7.7 Hybridtopologi

En hybridtopologi er en kombination af to eller flere forskellige typer af netværkstopologier. Denne type topologi anvendes ofte i store netværk, hvor det er nødvendigt at udnytte fordelene ved flere forskellige topologier.

- **Fordele:**
  - Fleksibel og skalerbar.
  - Kan optimeres for at udnytte styrkerne ved flere topologier.

- **Ulemper:**
  - Kompleks at designe og implementere.
  - Højere omkostninger ved installation og vedligeholdelse.

7.8 Daisy Chain Topologi

I en daisy chain topologi er hver node forbundet til to andre noder, og danner en lineær kæde. Denne type topologi bruges ofte i små netværk eller til at forbinde enheder i en sekventiel rækkefølge.

- **Fordele:**
  - Enkel og billig at installere.
  - Kræver mindre kabel end andre topologier.

- **Ulemper:**
  - Dataoverførsel kan være langsommere, da data skal passere gennem flere noder.
  - Hvis en node i midten af kæden fejler, kan det bryde kommunikationen i netværket.

Valget af netværkstopologi afhænger af flere faktorer, herunder netværkets størrelse, budget, ønsket pålidelighed og krav til datahastighed. Forståelsen af de forskellige topologier og deres egenskaber er afgørende for at kunne designe og implementere effektive netværksløsninger.

Del IV

Seriel/Parallel Kommunikation

Kapitel 8

Grundlæggende Seriel og Parallel Kommunikation

8.1 Introduktion til Seriel Kommunikation

Seriel kommunikation er en metode til at overføre data, hvor bits sendes én efter én over en enkelt kommunikationskanal eller tråd. Dette står i kontrast til parallel kommunikation, hvor flere bits sendes samtidigt over flere tråde. Seriel kommunikation er almindeligt anvendt i forskellige industrielle applikationer på grund af dens enkelhed og effektivitet, især over lange afstande.

I industrielle systemer bruges seriel kommunikation ofte til at forbinde enheder som PLC'er, sensorer, og aktuatorer, der kræver en pålidelig og relativt langsom dataoverførsel. De mest anvendte serielle kommunikationsstandarder inkluderer RS232, RS422, og RS485, som alle har forskellige fordele afhængigt af applikationen og miljøet.

Seriel kommunikation er populær i industrielle miljøer på grund af dens evne til at operere over lange kabellængder med lav støjfølsomhed. Sammenlignet med parallel kommunikation kræver seriel kommunikation færre forbindelser, hvilket reducerer kompleksiteten og omkostningerne ved installation og vedligeholdelse.

I dette afsnit vil vi udforske grundlæggende begreber inden for seriel kommunikation, herunder de mest anvendte standarder og deres specifikationer. Vi vil også se på praktiske anvendelser af seriel kommunikation i industrielle netværk samt de udfordringer og løsninger, der er forbundet med denne kommunikationsform.

8.2 Bits, Bytes og Tegn

En computer anvender det binære talsystem, som kun har to cifre, 0 og 1. Ethvert tal kan repræsenteres ved en streng af disse cifre, kendt som bits (fra binært ciffer). For eksempel svarer det decimale tal 5 til det binære tal 101. Da en bit kun kan have to værdier, kan den repræsenteres af en spænding,

Bit	1 eller 0
Dibit	To bits (10)
Nibble	Fire bits (1001 eller et Hex-tegn)
Byte	Otte bits eller to nibbles (11000001, C1 Hex)
Word	Bredden af computerens bus

Tabel 8.1: Forskellige sæt af bits

der enten er tændt (1) eller slukket (0). Dette kaldes også logisk 1 og logisk 0. Typiske værdier, der bruges i en computer, er 0 V for logisk 0 og +5 V for logisk 1, selvom det også kan være omvendt, dvs. 0 V for 1 og +5 V for 0.

En streng af otte bits kaldes en 'byte' (eller oktet) og kan have værdier, der spænder fra 0 (0000 0000) til 255_{10} ($1111\ 1111_2$). Computere manipulerer generelt data i bytes eller multipla af bytes. Programmører bruger 'hexade-

Decimal	Hexadecimal	Binær
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Tabel 8.2: Den hexadecimale tabel

cimal' notation, fordi det er en mere bekvem måde at definere og håndtere bytes på. I det hexadecimale talsystem er der 16 cifre (0–9 og A–F), som hver

repræsenteres af fire bits. En byte repræsenteres derfor af to hexadecimale cifre.

Et 'tegn' er et symbol, der kan udskrives. Alfabetet, både store og små bogstaver, tal, tegnsætningstegn og symboler som '*' og '&' er alle tegn. En computer skal kunne udtrykke disse tegn på en sådan måde, at de kan forstås af andre computere og enheder. Den mest almindelige kode til at opnå dette er den amerikanske standardkode for informationsudveksling (ASCII) beskrevet i afsnit 2.8.

8.3 Kommunikationsprincipper

Ethvert datakommunikationssystem kræver følgende komponenter:

- En datakilde (en sender eller linjedriver), som konverterer informationen til en form, der er egnet til transmission over en forbindelse.
- En modtager, der modtager signalet og konverterer det tilbage til de oprindelige data.
- En kommunikationsforbindelse, der transporterer signalerne. Dette kan være kobberledninger, optiske fibre, og radio- eller satellitforbindelser.

Derudover skal senderen og modtageren kunne forstå hinanden. Dette kræver enighed om en række faktorer. De vigtigste er:

- Den anvendte signaleringstype.
- Definitionen af en logisk '1' og en logisk '0'.
- De koder, der repræsenterer symbolerne.
- Opretholdelse af synkronisering mellem sender og modtager.
- Hvordan dataflowet kontrolleres, så modtageren ikke bliver overbelastet.
- Hvordan man opdager og korrigerer transmissionsfejl.

De fysiske faktorer kaldes 'grænsefladestandard'; de andre faktorer udgør 'protokoller'.

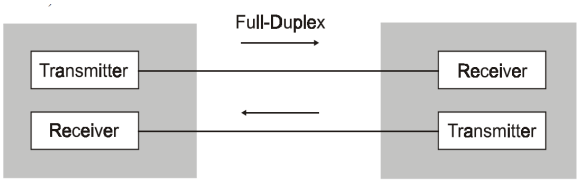
Den fysiske metode til at overføre data over en kommunikationsforbindelse varierer afhængigt af det anvendte medium. For eksempel kan de binære værdier 0 og 1 signaleres ved tilstedeværelsen eller fraværet af en spænding på en kobberledning, ved et par af lydtoner genereret og dekodet af et modem i tilfælde af telefonisystemet, eller ved brug af moduleret lys i tilfælde af optisk fiber.

8.4 Kommunikationsmodi

I ethvert kommunikationslink, der forbinder to enheder, kan data sendes i en af tre kommunikationsmodi. Disse er:

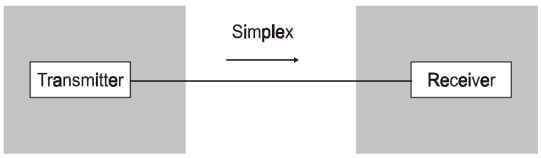
- Simplex
- Half duplex
- Full duplex

En simplex system er designet til at sende beskeder i én retning. Dette er illustreret i Figur 8.1.



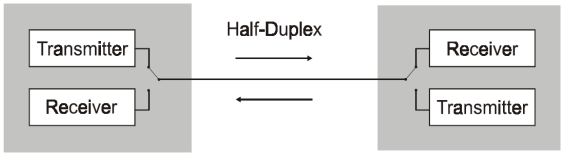
Figur 8.1: Simplex kommunikation

Et duplex system er designet til at sende beskeder i begge retninger. Half duplex opstår, når data kan strømme i begge retninger, men kun i én retning ad gangen (Figur 8.2).



Figur 8.2: Half-Duplex kommunikation

I et full-duplex system kan data strømme i begge retninger samtidig (Figur 8.3).



Figur 8.3: Full-Duplex kommunikation

8.5 Asynkrone systemer

Et asynkront system er et, hvor hvert tegn eller byte sendes inden for en ramme. Modtageren begynder ikke at detektere, før den modtager den første bit, kendt som 'startbiten'.

Startbit: Startbiten er i den modsatte spændingstilstand i forhold til hvilspændingen og tillader modtageren at synkronisere med senderen for de følgende data i rammen.

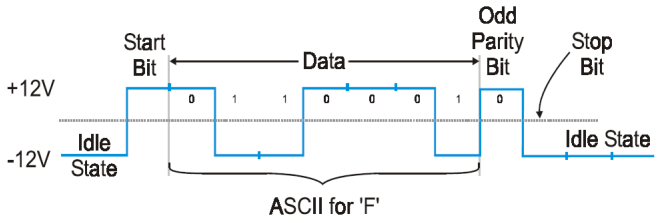
Modtagelse af bits: Modtageren læser de enkelte bits i rammen, efterhånden som de ankommer, og ser enten logisk 0-spændingen eller logisk 1-spændingen på det rette tidspunkt. 'Clock'-hastigheden i hver ende skal være den samme, så modtageren ser efter hver bit på det tidspunkt, hvor senderen sender den.

Synkronisering: Da urene er synkroniseret i starten af hver ramme, kan der tolereres nogen variation ved lavere transmissionshastigheder. Den tilladte variation falder, når dataoverførselshastighederne stiger, og asynkron kommunikation kan have problemer ved høje hastigheder (over 100 kbps).

8.6 Meddelelsesformat

En asynkron ramme kan have følgende format:

- **Startbit:** Signalerer starten af rammen
- **Data:** Normalt 7 eller 8 bits data, men kan være 5 eller 6 bits
- **Paritetsbit:** Valgfri fejldetektionsbit
- **Stopbit(er):** Normalt 1, 1,5 eller 2 bits. En værdi på 1,5 betyder, at niveauet holdes 1,5 gange så længe som en enkelt bit.



Figur 8.4: Asynkront rammeformat

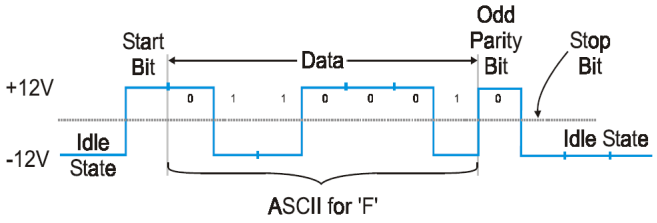
Et asynkront rammeformat er vist i Figur 8.4. Senderen og modtageren skal indstilles til præcis den samme konfiguration, så dataene kan udtrækkes korrekt fra rammen. Da hvert tegn har sin egen ramme, er den faktiske dataoverførselshastighed mindre end bithastigheden. For eksempel, med en startbit, syv databits, en paritetsbit og en stopbit, er der ti bits nødvendige for at sende syv bits data. Derfor er transmissionen af nyttige data 70% af den samlede bithastighed.

8.7 Synkrone systemer

I synkrone systemer synkroniserer modtageren indledningsvis til senderens klokkeimpulser, som er indlejret i de transmitterede datastrømme. Dette gør det muligt for modtageren at opretholde sin synkronisering gennem store beskeder, som typisk kan være op til 4500 bytes (36 000 bits). Dette tillader store rammer at blive transmitteret effektivt ved høje datahastigheder. Det synkrone system pakker mange tegn sammen og sender dem som en kontinuerlig strøm, kaldet en pakke eller en ramme.

8.8 Meddelelsesformat

Et typisk synkront system rammeformat er vist i Figur 8.5.



Figur 8.5: Typisk synkront system rammeformat

- **Preamble:** Dette omfatter en eller flere bytes, der gør det muligt for modtageren at synkronisere med rammen.
- **SFD:** Start af ramme delimitter signalerer starten af rammen.
- **Destination:** Adressen, som rammen sendes til.
- **Source:** Adressen, som rammen stammer fra.
- **Length:** Antallet af bytes i datafeltet.
- **Data:** Den faktiske besked.
- **FCS:** Frame check sequence er til fejldetektion.

Definition og Grundlæggende Koncept Seriel kommunikation er en teknik, hvor data overføres én bit ad gangen over en enkelt kommunikationskanal, som typisk er en ledning eller et kabel. Denne form for kommunikation bruges ofte, når data skal sendes over længere afstande, eller hvor det er vigtigt at minimere antallet af ledninger, der er nødvendige for kommunikationen.

I modsætning til seriel kommunikation involverer parallel kommunikation overførsel af flere bits samtidigt over flere kanaler eller ledninger. Selvom parallel kommunikation kan opnå højere datahastigheder på korte afstande, kræver den flere ledninger og kan være mere udsat for interferens og synkroniseringsproblemer, især over lange afstande.

Den primære fordel ved seriel kommunikation er dens enkelhed og pålidelighed, især i miljøer, hvor der er behov for kommunikation over lange kabler. Det reducerede antal ledninger mindsker risikoen for støj og signalforvrængning, hvilket gør seriel kommunikation ideel til mange industrielle applikationer, hvor robusthed og stabilitet er afgørende.

Seriel kommunikation anvender typisk standarder som RS232, RS422 og RS485, der hver især har deres egne specifikationer og anvendelsesområder, afhængigt af kravene til datahastighed, rækkevidde og interferensbeskyttelse.

Kapitel 9

Industriel Seriel Kommunikation og Feltbusprotokoller

9.1 RS232

RS232 (Recommended Standard 232) er en standard for seriel kommunikation, der er udviklet af Electronic Industries Alliance (EIA). Den bruges primært til at forbinde dataterminaludstyr (DTE) såsom computere til datakommunikationsudstyr (DCE) såsom modemmer. RS232-standardens primære formål er at definere elektriske egenskaber, signalering og timing for serielle dataoverførsler mellem enheder.

9.1.1 Fysiske Lag

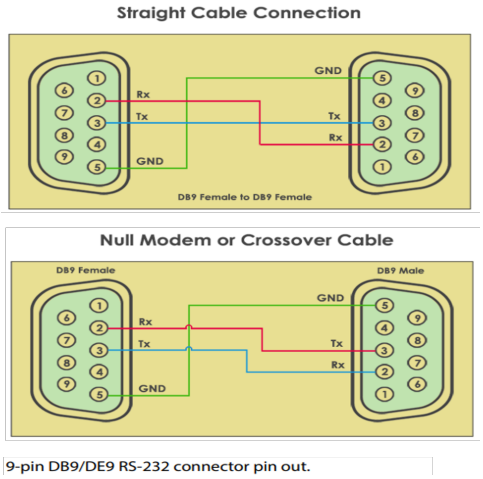
RS232 definerer det fysiske lag af kommunikationsforbindelsen, herunder stiktyper, kabler og elektriske signaler. Den mest almindelige stiktype er DB9 (9-pins) eller DB25 (25-pins), selvom andre konfigurationer også findes.

9.1.2 Elektriske Signaler

RS232 bruger single-ended signalering, hvilket betyder, at signalet sendes som en spændingsforskel mellem en signalleder og en fælles jordleder. Typiske spændingsniveauer er:

- **Logisk 0 (Marking)**: +3V til +15V
- **Logisk 1 (Spacing)**: -3V til -15V

Støjniveauer under ±3V betragtes som udefinerede, hvilket skaber en bufferzone for at minimere signalforstyrrelser.



Figur 9.1: DB9 stik for RS232

9.1.3 Signalering og Pins

De vigtigste signaler og deres respektive pins i et DB9-stik er som følger:
Pin 1: DCD (Data Carrier Detect)
Angiver, at modemet har opdaget en bærer fra fjernmodemet.

Pin 2: RXD (Receive Data)
Bruges til at modtage data fra fjernmodemet.

Pin 3: TXD (Transmit Data)
Bruges til at sende data til fjernmodemet.

Pin 4: DTR (Data Terminal Ready)
Signalerer, at terminalen eller computeren er klar til at kommunikere.

Pin 5: GND (Signal Ground)
Fælles jordforbindelse for alle signaler.

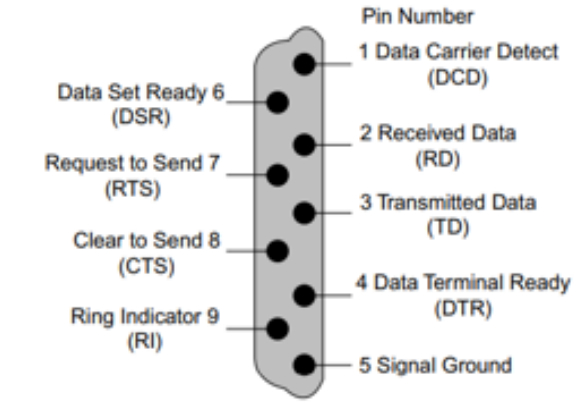
Pin 6: DSR (Data Set Ready)
Indikerer, at modemet er klar til at kommunikere.

Pin 7: RTS (Request To Send)
Bruges til at anmode om tilladelse til at sende data.

Pin 8: CTS (Clear To Send)
Signalerer, at det er klart at modtage data.

Pin 9: RI (Ring Indicator)

Indikerer, at der kommer et indgående opkald.bab



9-pin DB9/DE9 RS-232 connector pin out.

9.1.4 Dataoverførsel

RS232 bruger asynkron dataoverførsel, hvilket betyder, at data sendes i et kontinuerligt strøm af bits uden en fast tidsbase. Dataoverførslen styres af start- og stopbits, som definerer begyndelsen og slutningen af hver dataramme. En typisk dataramme består af:

- **Startbit**: 1 bit (logisk 0)
- **Databits**: 5 til 9 bits (typisk 8 bits)
- **Paritetsbit**: 1 bit (valgfri, bruges til fejlregistrering)
- **Stopbits**: 1, 1.5 eller 2 bits (logisk 1)

9.1.5 Baudrate

Baudrate refererer til antallet af signalændringer pr. sekund og bestemmer dataoverførselshastigheden. Typiske baudrater for RS232 er 9600, 19200, 38400, 57600 og 115200 baud. Det er vigtigt, at begge kommunikerende enheder er indstillet til samme baudrate for at sikre korrekt dataoverførsel.

9.1.6 Fejlhåndtering

RS232 anvender enkle fejlhåndteringsmetoder som paritetskontrol, hvor en ekstra bit føjes til hver dataramme for at gøre antallet af 1'er enten lige (even parity) eller ulige (odd parity). Hvis det modtagne antal 1'er ikke matcher den forventede paritet, registreres en fejl.

9.1.7 Anvendelser

RS232 er blevet brugt i en lang række applikationer, herunder:

- Forbindelse af computere til modemmer
- Industriel automation og kontrolsystemer
- Seriel kommunikation med mikrokontrollere og indlejrede systemer
- Diagnostisk interface til netværksudstyr

9.1.8 Fordele og Ulemper

- **Fordele:**
  - Simpel og udbredt standard
  - Velegnet til korte afstande og lave hastigheder
- **Ulemper:**
  - Begrænset dataoverførselshastighed og afstand
  - Følsom overfor elektrisk støj
  - Kræver flere ledninger til fuld duplex kommunikation

RS232 har været en grundlæggende teknologi i mange år og bruges stadig i dag på trods af fremkomsten af mere avancerede serielle kommunikationsstandarder som USB og RS485.

9.2 RS422

RS422 (Recommended Standard 422) er en standard for seriel dataoverførsel, som er udviklet af Electronic Industries Alliance (EIA). RS422 blev designet til at forbedre de begrænsninger, der findes i RS232, især med hensyn til dataoverførselshastighed og afstand. RS422 anvender differentiel signalering, hvilket gør den mere robust overfor elektrisk støj og muliggør længere kommunikationsafstande.

9.2.1 Fysiske Lag

RS422 definerer det fysiske lag af kommunikationsforbindelsen, inklusive stiktyper, kabler og elektriske signaler. De mest almindelige stiktyper er DB9 (9-pins) og DB25 (25-pins), men andre konfigurationer findes også.

Figur 9.2: DB9 stik for RS422

9.2.2 Elektriske Signaler

RS422 bruger differentiel signalering, hvilket betyder, at signalet sendes som en spændingsforskel mellem to ledninger (A og B). Dette reducerer følsomheden overfor elektromagnetisk interferens (EMI). Typiske spændingsniveauer er:

- **Logisk 0 (Marking)**: -2V til -6V (A-B)
- **Logisk 1 (Spacing)**: +2V til +6V (A-B)

9.2.3 Signalering og Pins

De vigtigste signaler og deres respektive pins i et DB9-stik er som følger:

- **Pin 1**: GND (Signal Ground)
- **Pin 2**: TXD+ (Transmit Data Positive)
- **Pin 3**: TXD- (Transmit Data Negative)
- **Pin 4**: RXD+ (Receive Data Positive)
- **Pin 5**: RXD- (Receive Data Negative)
- **Pin 6-9**: Ikke brugt eller valgfri for kontrolsignaler

9.2.4 Dataoverførsel

RS422 understøtter asynkron og synkron dataoverførsel. Asynkron overførsel bruger start- og stopbits til at definere begyndelsen og slutningen af en dataramme, mens synkron overførsel bruger en klokkesignal til at synkronisere dataoverførslen.

9.2.5 Baudrate og Afstand

RS422 kan understøtte dataoverførselshastigheder op til 10 Mbps over korte afstande (op til 12 meter). Over længere afstande (op til 1200 meter) kan hastigheder op til 100 kbps opnås. Det er vigtigt at matche baudraten på begge kommunikerende enheder for at sikre korrekt dataoverførsel.

9.2.6 Fejlhåndtering

RS422's differentielle signalering reducerer fejl forårsaget af elektrisk støj. Desuden kan paritetskontrol og andre fejldetekteringsmetoder anvendes for at sikre dataintegritet.

9.2.7 Anvendelser

RS422 anvendes ofte i industrielle og kommercielle applikationer, herunder:

- Industriel automation og kontrolsystemer
- Seriel kommunikation mellem computere og periferiudstyr
- Netværksforbindelser over lange afstande
- Kommunikationslinjer i høj EMI-miljøer

9.2.8 Fordele og Ulemper

- **Fordele:**
  - Højere dataoverførselshastigheder og længere rækkevidde sammenlignet med RS232.
  - Robust overfor elektrisk støj på grund af differentiel signalering.
  - Mulighed for at forbinde flere modtagere (op til 10) på samme sender.
- **Ulemper:**
  - Mere kompleks end RS232 med hensyn til kabelføring og stik.
  - Kræver specielle drivere og modtagere til differentiel signalering.

RS422 er en kraftfuld standard for seriel kommunikation, der tilbyder højere hastigheder og længere afstande end RS232, hvilket gør den velegnet til krævende industrielle applikationer.

9.3 RS485

RS485 (Recommended Standard 485) er en standard for seriel dataoverførsel, udviklet af Electronic Industries Alliance (EIA). RS485 er designet til at muliggøre pålidelig kommunikation over lange afstande og i støjfyldte miljøer. Den anvender differentiel signalering, som gør den robust overfor elektromagnetisk interferens (EMI) og tillader multi-drop netværk, hvor flere enheder kan kommunikere over samme bus.

9.3.1 Fysiske Lag

RS485 definerer det fysiske lag af kommunikationsforbindelsen, herunder stiktyper, kabler og elektriske signaler. De mest almindelige stiktyper er terminalblokke og DB9-stik.

Figur 9.3: Terminalblok stik for RS485

9.3.2 Elektriske Signaler

RS485 bruger differentiel signalering, hvilket betyder, at signalet sendes som en spændingsforskel mellem to ledninger (A og B). Dette reducerer følsomheden overfor EMI. Typiske spændingsniveauer er:

- **Logisk 0 (Marking)**: -1.5V til -5V (A-B)
- **Logisk 1 (Spacing)**: +1.5V til +5V (A-B)

9.3.3 Signalering og Pins

De vigtigste signaler og deres respektive pins i et typisk terminalblok-stik er som følger:

- **Pin 1**: A (Data Line Positive)
- **Pin 2**: B (Data Line Negative)
- **Pin 3**: GND (Signal Ground)
- **Pin 4-5**: Valgfri for terminering eller skærmning

9.3.4 Dataoverførsel

RS485 understøtter både asynkron og synkron dataoverførsel. Asynkron overførsel bruger start- og stopbits til at definere begyndelsen og slutningen af en dataramme, mens synkron overførsel bruger et klokkesignal til at synkronisere dataoverførslen. RS485 tillader også multi-drop netværk, hvilket betyder, at op til 32 enheder kan tilsluttes på samme bus.

9.3.5 Baudrate og Afstand

RS485 kan understøtte dataoverførselshastigheder op til 10 Mbps over korte afstande (op til 15 meter). Over længere afstande (op til 1200 meter) kan hastigheder op til 100 kbps opnås. Det er vigtigt at matche baudraten på alle kommunikerende enheder for at sikre korrekt dataoverførsel.

9.3.6 Fejlhåndtering

RS485's differentielle signalering reducerer fejl forårsaget af elektrisk støj. Desuden kan paritetskontrol og andre fejldetekteringsmetoder anvendes for at sikre dataintegritet. RS485-netværk kan også bruge terminering modstande for at minimere refleksioner på kablet, hvilket forbedrer signalintegriteten.

9.3.7 Anvendelser

RS485 anvendes ofte i industrielle og kommercielle applikationer, herunder:

- Industriel automation og kontrolsystemer
- Seriel kommunikation mellem computere og periferiudstyr
- Netværksforbindelser over lange afstande
- Kommunikationslinjer i høj EMI-miljøer
- Bygningsautomation (f.eks. HVAC-systemer, belysningskontrol)

9.3.8 Fordele og Ulemper

- **Fordele:**
  - Højere dataoverførselshastigheder og længere rækkevidde sammenlignet med RS232.
  - Robust overfor elektrisk støj på grund af differentiel signalering.
  - Mulighed for at forbinde op til 32 enheder på samme bus (kan udvides med repeaters).
- **Ulemper:**
  - Mere kompleks end RS232 med hensyn til kabelføring og stik.
  - Kræver specielle drivere og modtagere til differentiel signalering.
  - Kan være komplekst at konfigurere og fejlfinde i store netværk.

RS485 er en kraftfuld standard for seriel kommunikation, der tilbyder højere hastigheder, længere afstande og multi-drop kapaciteter, hvilket gør den velegnet til krævende industrielle applikationer.

9.4 DeviceNet

DeviceNet er en industriel netværksprotokol, der bruges til at forbinde industrielle enheder såsom sensorer, aktuatorer og controllere. Det er en del af CIP (Common Industrial Protocol) og bruges ofte i automatiseringssystemer til at sikre pålidelig og effektiv kommunikation mellem enheder.

9.4.1 Introduktion

DeviceNet er designet til at være en fleksibel og skalerbar løsning til industriel kommunikation. Det giver mulighed for at tilføje og fjerne enheder uden at forstyrre det overordnede system, hvilket gør det ideelt til brug i dynamiske miljøer, hvor kravene til netværket kan ændre sig over tid.

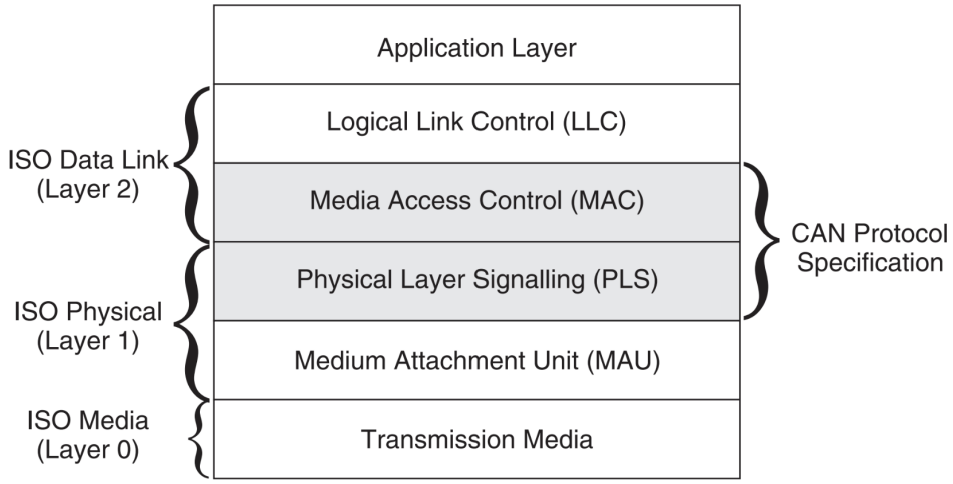


Figure 12.1
DeviceNet vs the OSI model

9.4.2 Fysisk Lag

Topologi

DeviceNet bruger en bus-topologi, hvor alle enheder er forbundet til en fælles kommunikationsbus. Dette layout er både simpelt og økonomisk, da det kræver minimal kabling og giver nem adgang til alle enheder på netværket. Det understøtter også stjerne- og trunk-line topologier, hvilket giver fleksibilitet i netværksdesign.

9.4.3 Stikforbindelser

Pluggbare (utætte) stik

Disse stik er nemme at tilslutte og afbryde, hvilket gør dem ideelle til brug i miljøer, hvor hurtig installation og vedligeholdelse er nødvendig. De er dog ikke tætte og bør derfor ikke bruges i miljøer, hvor de kan udsættes for væsker eller støv.

Fastforbundne (utætte) stik

Ligesom de pluggable stik er disse stik ikke tætte, men de giver en mere permanent forbindelse. De bruges ofte i mere stabile miljøer, hvor der ikke er behov for hyppig til- og frakobling.

Mini (tætte) stik

Disse stik er designet til at være tætte og beskytte mod indtrængen af væsker og støv. De er ideelle til brug i barske industrielle miljøer.

Mikro (tætte) stik

Mikrostik er mindre end mini-stik, men tilbyder samme niveau af tæthed. De bruges ofte i applikationer, hvor pladsen er begrænset.

9.4.4 Kabelbudgetter

Kabelbudgetter i DeviceNet bestemmer den maksimale længde og type af kabel, der kan bruges, uden at det påvirker netværkets ydeevne. Dette inkluderer overvejelser om signalstyrke, spændingsfald og dataoverførselshastigheder.

9.4.5 Enhedstaps

Tætte taps

Tætte taps giver en sikker forbindelse, der beskytter mod indtrængen af væsker og støv. De bruges i miljøer, hvor pålidelighed og beskyttelse er afgørende.

IDC taps

IDC (Insulation Displacement Connector) taps er nemme at installere og kræver ikke, at kablets isolering fjernes. Dette gør installationen hurtigere og reducerer risikoen for skader på kablet.

Åbne taps

Åbne taps er ikke beskyttede mod miljøpåvirkninger og bør kun bruges i kontrollerede indendørs miljøer.

Multiport åbne taps

Disse taps giver mulighed for tilslutning af flere enheder på samme punkt, hvilket kan reducere kablingsomkostningerne og forenkle installationen.

Strømtaps

Strømtaps giver en dedikeret forbindelse til strømforsyning af enheder på netværket, hvilket kan være nødvendigt i applikationer med høje strømkrav.

9.4.6 Kabler

Tykt kabel

Tykt kabel bruges i DeviceNet-applikationer, hvor lange kabellængder eller høje strømkrav er nødvendige. Det giver lavere modstand og bedre signalintegritet over lange afstande.

Tyndt kabel

Tyndt kabel bruges i applikationer, hvor pladsen er begrænset, og kabellængderne er korte. Det er lettere at håndtere og installere i trange rum.

Fladt kabel

Fladt kabel bruges i applikationer, hvor kablet skal lægges under gulve eller tæpper. Det er nemt at skjule og giver en pænere installation.

9.4.7 Netværksstrøm

Generel tilgang

Netværksstrøm i DeviceNet sikrer, at alle enheder på netværket får tilstrækkelig strøm til at fungere korrekt. Dette inkluderer overvejelser om strømforsyningens kapacitet og distribution.

Enkel forsyning – ende tilsluttet

Denne tilgang til netværksstrøm bruger en enkelt strømforsyning, der er tilsluttet i enden af netværket. Dette er en simpel og økonomisk løsning, men kan være mindre pålidelig i store netværk.

Enkel forsyning – midter tilsluttet

Denne tilgang placerer strømforsyningen i midten af netværket, hvilket giver en mere jævn fordeling af strømmen og kan forbedre pålideligheden i større netværk.

Forslag til undgåelse af fejl og strømforsyningsmuligheder

For at undgå fejl i netværksstrømmen skal der tages højde for faktorer som korrekt dimensionering af strømforsyningen, brug af redundante strømforsyninger og regelmæssig vedligeholdelse.

9.4.8 Systemjord

Systemjord sikrer, at alle enheder på netværket har en fælles jordforbindelse, hvilket er afgørende for at undgå jordsløjfer og signalstøj.

9.4.9 Signalering

DeviceNet bruger signaleringsteknikker til at kommunikere data mellem enheder. Dette inkluderer brugen af specifikke spændingsniveauer og tidsrammer for at sikre pålidelig dataoverførsel.

9.4.10 Data Link Lag

Rammeformat

Rammeformatet i DeviceNet definerer strukturen af de data, der overføres mellem enheder. Dette inkluderer felter som start- og stopbits, adresseinformation og fejlkontrol.

Mediumadgang

Mediumadgang kontrollerer, hvordan enheder får adgang til kommunikationsmediet. DeviceNet bruger en deterministisk tilgang, hvor hver enhed tildeles et bestemt tidspunkt til at sende data.

Fragmentering

Fragmentering bruges til at opdele store datarammer i mindre dele, der kan overføres effektivt over netværket. Disse fragmenter samles igen ved modtageren for at gendanne den oprindelige data.

9.4.11 Applikationslaget

Applikationslaget i DeviceNet definerer de protokoller og services, der bruges til at udføre specifikke funktioner, såsom læsning og skrivning af data, alarmhåndtering og diagnostik.

9.4.12 Fejlfinding

Introduktion

Fejlfinding i DeviceNet involverer identifikation og løsning af problemer, der kan påvirke netværkets ydeevne og pålidelighed.

Værktøjer til fejlfinding

Værktøjer som netværksscannere, multimetre og protokolanalysatorer kan bruges til at diagnosticere problemer i DeviceNet-netværk.

Fejlfindingsprocedurer

Fejlfindingsprocedurer inkluderer systematisk kontrol af kabler, stik, strømforsyninger og netværkskonfigurationer for at identificere og løse problemer.

9.4.13 Opsummering

DeviceNet er en robust og fleksibel netværksprotokol, der er designet til at opfylde kravene i moderne industrielle automatiseringssystemer. Ved at forstå de forskellige aspekter af DeviceNet, fra fysisk lag til applikationslag og fejlfinding, kan teknikere og ingeniører sikre, at deres netværk fungerer optimalt og pålideligt.

1.2.3 General Specifications

DeviceNet Communication Specifications

Item	Specification			
Supported Connection	- I/O messaging connection (Polling, Strove, Cyclic, Change of State) - Explicit messaging connection All connections are conformed to DeviceNet communication protocol.			
Baud Rates	125 kbps, 250 kbps, 500 kbps			
Transfer Distance	Baud Rates	Max. Network Length	Drop Length	Total Drop Line Length
	0 kbps	10 m	6m or under	39 m or under
	250 kbps	250 m *	6 m or under	78 m or under
	125 kbps	500 m *	6 m or under	156 m or under
Maximum Nodes	64 (including master unit)			
Data Length / Frame	8 byte (data can be divided and transferred.)			
Bus Access	CSMA/NBA			
Error Detection	CRC error / Duplicate node address check			
Cable	5-wire cable dedicated to DeviceNet (2 wires for signal, 2 wires for power supply, 1 shield wire)			
Communications Power Supply Voltage	24 V DC (supplied from a connector)			

* When thin cable is used for trunk line, the maximum network length is 100 m.

9.5 ProfiBus PA/DP/FMS Overview

ProfiBus (Process Field Bus) er en standardiseret industrielt kommunikationsprotokol, der bruges til at forbinde automatiseringsenheder som sensorer, aktuatorer og controllere. Det er en af de mest udbredte feltbusstandarder i verden og understøtter både procesautomatisering (PA), decentraliseret periferikommunikation (DP) og feltbussystemer (FMS).

9.5.1 Introduktion

ProfiBus er udviklet til at muliggøre hurtig og pålidelig kommunikation mellem industrielle enheder. Det er en alsidig protokol, der kan bruges i en bred vifte af applikationer, herunder fabriksautomatisering, proceskontrol og bygningsautomatisering.

9.5.2 ProfiBus Protocol Stack

ProfiBus-protokollen er opdelt i flere lag, som hver især håndterer forskellige aspekter af kommunikationen. Denne lagdelte arkitektur gør det muligt at isolere og håndtere forskellige funktioner effektivt.

Fysisk Lag (Layer 1)

Det fysiske lag i ProfiBus definerer de elektriske og mekaniske egenskaber ved netværksforbindelserne. Dette inkluderer specifikationer for kabler, stik og elektriske signaler, der bruges til at overføre data mellem enheder.

Datalink Lag (Layer 2)

Datalink-laget håndterer pålidelig overførsel af data mellem enheder på netværket. Dette inkluderer fejlregistrering og korrektion, adressering af enheder og styring af datarammer.

Applikationslag

Applikationslaget definerer de protokoller og tjenester, der bruges til at udføre specifikke funktioner som læsning og skrivning af data, alarmhåndtering og diagnostik. Dette lag sikrer, at applikationer kan kommunikere effektivt over netværket.

Fieldbus Message Specification (FMS)

FMS specificerer de meddelelser, der bruges til at kommunikere mellem enheder på feltbusnetværket. Dette inkluderer standarder for meddelelsesformat og kommunikationsprotokoller.

Lower Layer Interface (LLI)

LLI fungerer som en grænseflade mellem de lavere lag (fysisk og datalink) og de højere lag (applikation og FMS). Det sikrer, at data kan overføres effektivt mellem disse lag.

Fieldbus Management Layer (FMA 7)

FMA 7 håndterer styring og konfiguration af feltbusnetværket. Dette inkluderer opgaver som netværksinitialisering, konfigurationsstyring og diagnosticering af netværksfejl.

9.5.3 ProfiBus Kommunikation Model

ProfiBus kommunikationsmodellen beskriver, hvordan data overføres mellem enheder på netværket. Dette inkluderer beskrivelser af kommunikationscyklusser, dataoverførselshastigheder og synkroniseringsmetoder.

9.5.4 Forhold mellem Applikationsproces og Kommunikation

Dette afsnit beskriver, hvordan applikationsprocesser interagerer med kommunikationsprotokollen for at sikre effektiv dataoverførsel. Det forklarer, hvordan data fra applikationslaget omsættes til meddelelser, der kan sendes over netværket.

9.5.5 Kommunikationsobjekter

Kommunikationsobjekter i ProfiBus refererer til de enheder, data og tjenester, der kan adresseres og styres over netværket. Dette inkluderer beskrivelser af de forskellige typer af kommunikationsobjekter og deres funktioner.

9.5.6 Ydeevne

Dette afsnit diskuterer ydeevnen af ProfiBus-netværket, herunder dataoverførselshastigheder, responstider og netværkskapacitet. Det giver også retningslinjer for optimering af netværksydeevne.

9.5.7 Systemoperation

Konfiguration

ProfiBus-netværket kræver korrekt konfiguration for at fungere optimalt. Dette afsnit beskriver de trin, der er nødvendige for at konfigurere enheder, tildele adresser og indstille netværksparametre.

Dataoverførsel mellem DPM1 og DP-slaver

Dette afsnit beskriver processen for dataoverførsel mellem masterenheden (DPM1) og slaveenhederne (DP-slaver). Det forklarer, hvordan data pakkes, adresseres og sendes over netværket.

Synkronisering og Frysemodi

Synkronisering og frysemodi bruges til at koordinere dataoverførsel mellem enheder og sikre, at dataene overføres på det rigtige tidspunkt. Dette afsnit beskriver, hvordan disse funktioner implementeres og bruges.

Sikkerhed og Beskyttelse af Stationer

Sikkerhed og beskyttelse er afgørende for at sikre, at netværket fungerer pålideligt og uden forstyrrelser. Dette afsnit beskriver de mekanismer, der bruges til at beskytte netværket mod fejl og angreb.

Blandet Drift af FMS- og DP-stationer

Dette afsnit beskriver, hvordan FMS- og DP-stationer kan fungere samtidigt på samme netværk. Det forklarer, hvordan kompatibilitet og interoperabilitet sikres mellem forskellige typer enheder.

9.5.8 Fejlfinding

Introduktion

Fejlfinding er en vigtig del af vedligeholdelsen af ProfiBus-netværket. Dette afsnit introducerer de grundlæggende principper for fejlfinding og diagnosticering af netværksproblemer.

Fejlfinding Værktøjer

Der findes forskellige værktøjer, der kan bruges til at diagnosticere og løse problemer i ProfiBus-netværket. Dette afsnit beskriver nogle af de mest almindelige værktøjer og deres anvendelser.

Tips

Dette afsnit giver praktiske tips til fejlfinding og vedligeholdelse af ProfiBus-netværket. Det inkluderer anbefalinger til forebyggelse af almindelige problemer og optimering af netværksydelsen.

9.5.9 Opsummering

ProfiBus er en alsidig og pålidelig feltbusprotokol, der er designet til at opfylde behovene i moderne industrielle automatiseringssystemer. Ved at forstå de forskellige aspekter af ProfiBus, fra protokolstack til fejlfinding, kan teknikere og ingeniører sikre, at deres netværk fungerer effektivt og pålideligt.

1.3.3 General Specifications

PROFIBUS DP Communication Specifications

Item	Specification	
Communication Method	Hybrid (token passing procedure and master-slave communication)	
Baud Rates	9.6 kbps, 19.2 kbps, 93.75 kbps, 187.5 kbps, 500 kbps, 1500 kbps, 3 Mbps, 6 Mbps, and 12 Mbps.	
Transfer Distance	Baud Rates	Cable Length
	12 Mbps	100 m
	6 Mbps	100 m
	3 Mbps	100 m
	1500 kbps	200 m
	500 kbps	400 m
	187.5 kbps	1000 m
	93.75 kbps	1200 m
	19.2 kbps	1200 m
	9.6 kbps	1200 m
Maximum Stations	126 (including master unit and repeater)	
Data Length / Frame	244 bytes	
Cable	2-wire cable dedicated to PROFIBUS (2 wires for signal)	

Del V

Ethernet-baseret Kommunikation

Kapitel 10

Industriel netværksprotokoller og standarder

10.1 Modbus

Modbus er en netværksprotokol med varianter for både seriel og TCP/IP-baseret kommunikation.

Modbus RTU: Modbus RTU (Remote Terminal Unit) er en seriel kommunikationsprotokol, der ofte bruges over RS-485-standarden. Den definerer, hvordan data pakkes i enheder kaldet frames og overføres mellem master- og slave-enheder.

Protokol: Modbus RTU beskriver, hvordan data formateres og adresseres inden for en ramme. Den definerer også mekanismer til fejldetektion og fejlkorrektion.

Elektrisk Standard: Modbus RTU bruger elektriske standarder som RS-232, RS-422, og mest almindeligt, RS-485 for den fysiske transmission af data. RS-485 muliggør længere kabelafstande og understøtter multi-drop forbindelser.

Modbus TCP: Modbus TCP (Transmission Control Protocol) er en version af Modbus-protokollen, der kører over TCP/IP-netværk.

Protokol: Modbus TCP definerer, hvordan Modbus-data pakkes inden for TCP/IP-rammer. Det muliggør kommunikation over Ethernet og bruger IP-adresser til at identificere enheder på netværket.

Elektrisk Standard: Selvom Modbus TCP primært er en protokol, afhænger den af Ethernet's elektriske standarder for fysiske forbindelser. Dette

inkluderer brug af Ethernet-kabler og standard netværksudstyr som switches og routers.

10.1.1 Sammenfatning

Både Profinet og Modbus (RTU og TCP) er primært protokoller, men de har også elektriske specifikationer og krav til det fysiske lag:

- **Profinet:** En industriel Ethernet-protokol med specifikationer for elektriske standarder på det fysiske lag.
- **Modbus RTU:** En seriel protokol med tilknyttede elektriske standarder (RS-232, RS-422, RS-485).
- **Modbus TCP:** En TCP/IP-baseret protokol, der afhænger af Ethernet's elektriske standarder.

Dette afsnit understreger vigtigheden af at forstå både protokoller og elektriske standarder i industrielle netværk, hvilket er afgørende for at kunne designe og vedligeholde pålidelige og effektive kommunikationssystemer i industrielle miljøer.

10.1.2 Netværksprotokoller

En netværksprotokol er et sæt regler og konventioner, der bestemmer, hvordan data udveksles mellem enheder som computere, smartphones, tablets og routere. Protokoller styrer flowet af bits mellem netværksinterfacekort, kontrollerer overførselshastigheden mellem sender og modtager, og bestemmer pakkernes vej fra kilde til destination.

For eksempel, når du anmoder om en webside ved at skrive en URL i din browser, sender din computer en forbindelsesanmodning til webserveren og venter på et svar. Webserveren svarer med en forbindelsesbekræftelse, hvorefter din computer sender en GET-besked med navnet på den ønskede side. Webserveren sender derefter den anmodede webside (fil) til din computer.

Protokoller definerer formatet og rækkefølgen af de beskeder, der udveksles, samt de handlinger, der udføres ved afsendelse eller modtagelse af en besked. De er essentielle for internet- og computernetværk, hvor nogle er simple og ligetil, mens andre er komplekse og dybdegående.

Menneskelig analogi: Ligesom mennesker kommunikerer for at udveksle information, bruger netværksenheder protokoller til at udveksle data og styresignaler. Dette kan illustreres med et sekvensdiagram, hvor hver besked repræsenterer en del af kommunikationen mellem to parter. For eksempel kan en anmodning om tid sammenlignes med en TCP-anmodning om en fil.

10.1.3 Elektriske Standarder

Elektriske standarder definerer de fysiske egenskaber ved netværksforbindelser, såsom spændingsniveauer, signalstyrker, kabeltyper og stikforbindelser. Disse standarder sikrer, at de fysiske komponenter i netværket kan overføre data pålideligt.

Standarderne fastlægger, hvordan elektriske signaler skal behandles og transmitteres over netværket. Dette inkluderer specifikationer for voltagesvingninger, der styrer dataoverførsler, og hvor meget støj der kan tolereres uden at forstyrre signalet. For eksempel definerer Ethernet standarder for maksimale kabellængder og kabeltyper for at sikre, at signalerne forbliver stærke over lange afstande. USB-standarder specificerer typer af stik og kabler til data- og strømoverførsel.

Uden disse standarder ville der være betydelige kompatibilitetsproblemer mellem udstyr fra forskellige producenter, hvilket ville hæmme effektiv kommunikation og dataudveksling. Standarderne sikrer, at alle enheder på et netværk kan samarbejde effektivt, hvilket er afgørende for pålidelige og hurtige forbindelser i moderne kommunikationssystemer.

Menneskelig analogi: Ligesom bygningsreglementer sikrer, at bygninger er sikre ved at specificere elektriske installationer, strukturel integritet og brandmodstand, sikrer elektriske standarder for netværk, at data kan overføres sikkert og effektivt. Bygningsstandarder sikrer kompatibilitet og sikkerhed, uanset hvilken entreprenør der udfører arbejdet. Tilsvarende garanterer netværksstandarder, at komponenter fra forskellige producenter kan arbejde sammen uden problemer, og at signaler kan transmitteres pålideligt over netværket.

10.2 EtherNet/IP

EtherNet/IP (Ethernet Industrial Protocol) er en avanceret industrielt netværksprotokol, der bygger på standard Ethernet-teknologi. Den bruges til at forbinde automatiseringsenheder som PLC'er, sensorer, aktuatorer og andre industrielle kontrolsystemer i realtid. EtherNet/IP er en af de mest udbredte netværksprotokoller i industriel automation og er kendt for sin fleksibilitet, skalerbarhed og høje ydeevne.

10.2.1 Introduktion

EtherNet/IP er udviklet til at understøtte både realtidsdataudveksling og ikke-realtids kommunikation i industrielle miljøer. Protokollen anvender standard TCP/IP og UDP/IP for at sikre kompatibilitet med eksisterende IT-

netværk og tilbyder samtidig de nødvendige funktioner til at håndtere krævende industrielle applikationer, herunder motion control, procesautomation og sikkerhed.

10.2.2 EtherNet/IP Protocol Stack

EtherNet/IP-protokollen er opdelt i flere lag, hvor hvert lag håndterer specifikke funktioner i netværket. Denne lagdelte arkitektur tillader en høj grad af fleksibilitet og gør det muligt at implementere protokollen på forskellige typer hardware og i forskellige applikationer.

Fysisk Lag (Layer 1)

Det fysiske lag i EtherNet/IP bygger på standard Ethernet-teknologi, hvilket betyder, at det understøtter brugen af standard Ethernet-kabler og stik (f.eks. Cat5e, Cat6). Dette lag definerer de elektriske og mekaniske egenskaber ved netværksforbindelserne og sikrer kompatibilitet med eksisterende Ethernet-infrastrukturer.

Datalink Lag (Layer 2)

Datalink-laget i EtherNet/IP bygger på IEEE 802.3 Ethernet-standarden. Dette lag håndterer de grundlæggende funktioner for datatransmission, herunder adressering, rammeopbygning, fejlregistrering og fejlkontrol. Det sikrer pålidelig kommunikation mellem enheder på netværket.

Transport- og Netværkslag

Transportlaget og netværkslaget i EtherNet/IP bruger standard TCP/IP- og UDP/IP-protokoller. TCP/IP bruges primært til ikke-realtids kommunikation, mens UDP/IP bruges til realtidskommunikation. Disse lag håndterer routingen af data gennem netværket og sikrer, at dataene leveres korrekt og i rette tid.

Applikationslag

Applikationslaget i EtherNet/IP er baseret på Common Industrial Protocol (CIP). CIP definerer de dataobjekter, services og kommunikationsprotokoller, der bruges til at udføre specifikke opgaver som dataudveksling, kontrol og diagnostik. Dette lag muliggør interoperabilitet mellem enheder fra forskellige producenter.

Real-Time Communication

EtherNet/IP understøtter både standard og realtidskommunikation. Real-time Communication (RTC) er en kritisk funktion for applikationer som

motion control, hvor præcis timing og lav latenstid er nødvendigt. RTC opnås ved at anvende UDP/IP til at minimere kommunikationsforsinkelser.

10.2.3 EtherNet/IP Kommunikationsmodel

EtherNet/IP kommunikationsmodellen beskriver, hvordan data udveksles mellem enheder på netværket. Denne model understøtter både producer/consumer-modellen og client/server-modellen, hvilket giver stor fleksibilitet i, hvordan data struktureres og overføres.

10.2.4 Forhold mellem Applikationsproces og Kommunikation

Dette afsnit forklarer, hvordan applikationsprocesser interagerer med EtherNet/IP-protokollen for at sikre effektiv dataudveksling. Det beskriver, hvordan data fra applikationslaget omsættes til meddelelser, der kan sendes over netværket ved hjælp af CIP.

10.2.5 Kommunikationsobjekter

Kommunikationsobjekter i EtherNet/IP refererer til de enheder, data og tjenester, der adresseres og styres over netværket. Dette inkluderer en beskrivelse af forskellige typer af kommunikationsobjekter, såsom input/output-data, diagnostikbeskeder og konfigurationsdata, og deres anvendelse i forskellige scenarier.

10.2.6 Ydeevne

Dette afsnit diskuterer ydeevnen af EtherNet/IP-netværket, herunder dataoverførselshastigheder, latenstider og netværkskapacitet. Det giver også retningslinjer for optimering af netværksydeevne og sikring af, at systemet kan håndtere både realtids- og ikke-realtidskommunikation.

10.2.7 Systemoperation

Konfiguration

Konfigurationen af et EtherNet/IP-netværk er afgørende for dets korrekte funktion. Dette afsnit beskriver trinene til at konfigurere enheder, tildele IP-adresser, og opsætte CIP-parametre, såsom prioriteringer og cyklustider for forskellige datatyper.

Dataoverførsel mellem PLC'er og Enheder

Dette afsnit beskriver, hvordan data overføres mellem PLC'er og de tilsluttede enheder på netværket. Det inkluderer beskrivelser af cyclical data exchange og acyclic messaging, samt metoder til at sikre pålidelig dataudveksling.

Synkronisering og Timing

Synkronisering og timing er afgørende for applikationer, der kræver præcis koordination mellem enheder, såsom i motion control. Dette afsnit forklarer, hvordan timingmekanismer implementeres i EtherNet/IP for at sikre nøjagtig dataoverførsel.

Sikkerhed og Netværksbeskyttelse

Sikkerhed i EtherNet/IP-netværk er kritisk for at beskytte mod fejl og uautoriseret adgang. Dette afsnit beskriver de sikkerhedsforanstaltninger, der implementeres i EtherNet/IP, såsom autentificering, adgangskontrol, og kryptering for at beskytte dataoverførsel og enheder på netværket.

Integration med IT-systemer

EtherNet/IP's basering på standard Ethernet og TCP/IP gør det velegnet til integration med eksisterende IT-systemer. Dette afsnit beskriver, hvordan EtherNet/IP kan anvendes sammen med enterprise-netværk, SCADA-systemer og andre IT-platforme for at skabe en fuldt integreret industriel løsning.

10.2.8 Fejlfinding

Introduktion

Fejlfinding i et EtherNet/IP-netværk er en væsentlig del af vedligeholdelsen. Dette afsnit introducerer de grundlæggende metoder og principper for fejlfinding i EtherNet/IP-miljøer, herunder identifikation af almindelige problemer og deres løsninger.

Fejlfinding Værktøjer

Der findes forskellige værktøjer til fejlfinding i EtherNet/IP-netværk. Dette afsnit beskriver nogle af de mest anvendte værktøjer, såsom netværksanalyzatorer, diagnostisk software og integrerede funktioner i Studio 5000, og hvordan de bruges til at identificere og løse problemer.

Tips

Dette afsnit giver praktiske tips til optimering og vedligeholdelse af et EtherNet/IP-netværk. Tipsene inkluderer forebyggende foranstaltninger, overvågning af netværkets sundhed, og metoder til at undgå almindelige faldgruber i konfiguration og drift.

10.2.9 Opsummering

EtherNet/IP er en kraftfuld og fleksibel netværksprotokol, der er designet til at opfylde de komplekse krav i moderne industrielle automatiseringssystemer. Med sin evne til at håndtere realtidskommunikation, sikkerhed og integration med IT-systemer, er EtherNet/IP blevet en af de mest udbredte protokoller i industriel automation. Ved at forstå de forskellige aspekter af EtherNet/IP, fra protokolstacken til fejlfinding, kan teknikere og ingeniører sikre, at deres netværk fungerer optimalt og opfylder kravene til nutidens automatiseringsudfordringer.

1.4.3 General Specifications

EtherNet/IP Communication Specifications

Item	Specification
Supported Connection	- I/O messaging connection (Cyclic, Change of State) - Explicit messaging connection All connections are conformed to EtherNet/IP communication protocol.
Baud Rates	100 Mbps, 10 Mbps
Maximum Nodes	128 (including master unit)
Data Length / Frame	244 bytes
Access Control Type	CSMA/CD
Cable	Universal Ethernet cable

10.3 PROFINET

PROFINET (Process Field Network) er en moderne, industrielt netværksprotokol, der muliggør hurtig og pålidelig kommunikation mellem automatiseringskomponenter, såsom PLC'er, sensorer, aktuatorer og HMI'er. PROFINET er udviklet som en efterfølger til ProfiBus, med fokus på Ethernet-baseret kommunikation og støtte til realtidsdataudveksling i komplekse industrielle miljøer.

10.3.1 Introduktion

PROFINET er designet til at imødekomme kravene i moderne automatiseringssystemer, hvor der er behov for højhastighedskommunikation, integration med IT-systemer og understøttelse af avancerede funktioner som sikkerhed og trådløs kommunikation. Protokollen gør det muligt at forbinde og styre en bred vifte af industrielle enheder i realtid, hvilket gør den ideel til både fabriksautomatisering og processtyring.

10.3.2 PROFINET Protocol Stack

PROFINET-protokollen er opdelt i flere lag, der hver især håndterer forskellige aspekter af kommunikationen. Denne strukturerede tilgang sikrer, at forskellige funktioner kan integreres og udføres effektivt inden for det samme netværk.

Fysisk Lag (Layer 1)

Det fysiske lag i PROFINET er baseret på standard Ethernet-teknologi, hvilket betyder, at det understøtter standardiserede Ethernet-kabler og stik (f.eks. Cat5e, Cat6) og tilbyder høj båndbredde og fleksibilitet i netværksdesign. Dette lag specificerer også kravene til signalering og elektriske egenskaber.

Datalink Lag (Layer 2)

Datalink-laget i PROFINET bygger på Ethernet MAC-laget, men tilføjer protokoltilpasninger, der understøtter realtidskommunikation (RT) og isokron realtidskommunikation (IRT). Dette lag håndterer pålidelig datatransmission, fejldetektion, adressering af enheder og datarammer.

Applikationslag

Applikationslaget i PROFINET omfatter protokoller og tjenester, der er nødvendige for at implementere specifikke funktioner såsom procesdataudveks-

ling, alarmer og diagnostik. Dette lag er også ansvarligt for konfiguration og parameterisering af enhederne på netværket.

Real-Time Communication (RT)

RT-funktionen i PROFINET muliggør overførsel af procesdata med minimal latenstid, hvilket er afgørende for tidskritiske applikationer. RT kan håndtere periodiske og applikationsbestemte cykliske dataoverførsler.

Isocronous Real-Time Communication (IRT)

IRT er en avanceret funktion i PROFINET, der gør det muligt at synkronisere dataoverførsel med en meget præcis timing, hvilket er nødvendigt i applikationer som motion control, hvor selv små forsinkelser kan føre til fejl.

10.3.3 PROFINET Kommunikationsmodel

PROFINET kommunikationsmodellen beskriver, hvordan data udveksles mellem enheder på netværket, inklusive de forskellige kommunikationscyklusser og dataoverførselshastigheder. Modellen understøtter både realtids- og ikke-realtidskommunikation, hvilket gør det muligt at kombinere proces- og konfigurationsdata på samme netværk.

10.3.4 Forhold mellem Applikationsproces og Kommunikation

Dette afsnit forklarer, hvordan applikationsprocesser interagerer med PROFINET-kommunikationsprotokollen for at sikre effektiv dataudveksling. Det beskriver, hvordan data struktureres og sendes mellem forskellige lag i protokolstacken for at opnå optimal kommunikationsperformance.

10.3.5 Kommunikationsobjekter

Kommunikationsobjekter i PROFINET refererer til de enheder, data og tjenester, der adresseres og styres over netværket. Dette afsnit inkluderer en beskrivelse af forskellige typer af kommunikationsobjekter, såsom input/output-data, alarmer og diagnostikbeskeder, og deres anvendelse i forskellige scenarier.

10.3.6 Ydeevne

Dette afsnit diskuterer ydeevnen af PROFINET-netværket, herunder dataoverførselshastigheder, latenstider og netværkskapacitet. Det indeholder også retningslinjer for optimering af netværkets ydeevne og sikring af, at systemet kan håndtere både realtids- og isokron dataoverførsel.

10.3.7 Systemoperation

Konfiguration

Korrekt konfiguration er afgørende for at sikre, at PROFINET-netværket fungerer optimalt. Dette afsnit beskriver trinene til at konfigurere enheder, tildele IP-adresser og opsætte PROFINET-parametre, såsom cyklustider og prioriteter for forskellige datatyper.

Dataoverførsel mellem PLC'er og Enheder

Dette afsnit beskriver, hvordan data overføres mellem PLC'er og de tilsluttede enheder på netværket. Det inkluderer beskrivelser af cyklisk og acyklisk dataoverførsel samt metoder til at sikre pålidelig dataudveksling.

Synkronisering og Timing

Synkronisering og timing er centrale elementer i PROFINET-netværk, især i applikationer, der kræver præcis timing og koordinering mellem enheder. Dette afsnit forklarer, hvordan IRT og andre synkroniseringsmetoder implementeres i PROFINET.

Sikkerhed og Netværksbeskyttelse

Sikkerhed i PROFINET-netværk er kritisk for at beskytte mod fejl og uautoriseret adgang. Dette afsnit beskriver de sikkerhedsforanstaltninger, der implementeres i PROFINET, såsom adgangskontrol, dataautentificering og kryptering.

Integration med IT-systemer

Et af PROFINET's styrker er dets evne til at integrere med eksisterende IT-systemer. Dette afsnit beskriver, hvordan PROFINET kan anvendes sammen med enterprise-netværk, SCADA-systemer og andre IT-platforme for at skabe en fuldt integreret industriel løsning.

10.3.8 Fejlfinding

Introduktion

Fejlfinding i et PROFINET-netværk er en essentiel del af vedligeholdelsen. Dette afsnit introducerer de grundlæggende metoder og principper for fejlfinding i PROFINET-miljøer, herunder de mest almindelige problemer og deres løsninger.

Fejlfinding Værktøjer

Der findes en række værktøjer til fejlfinding i PROFINET-netværk. Dette afsnit beskriver nogle af de mest anvendte værktøjer, som netværkssniffere, diagnostiske software og integrerede funktioner i TIA Portal, og hvordan de anvendes til at identificere og løse problemer.

Tips

Dette afsnit giver praktiske tips til optimering og vedligeholdelse af et PROFINET-netværk. Tipsene inkluderer forebyggende foranstaltninger, overvågning af netværkets sundhed, og metoder til at undgå almindelige faldgruber i konfiguration og drift.

10.3.9 Opsummering

PROFINET er en kraftfuld og fleksibel netværksprotokol, der er designet til at opfylde de komplekse krav i moderne industrielle automatiseringssystemer. Med sin evne til at håndtere realtidskommunikation, sikkerhed og integration med IT-systemer, er PROFINET blevet en af de mest anvendte protokoller i industriens 4.0 æra. Ved at forstå de forskellige aspekter af PROFINET, fra protokolstacken til fejlfinding, kan teknikere og ingeniører sikre, at deres netværk fungerer optimalt og opfylder kravene til nutidens automatiseringsudfordringer.

10.4 AS-Interface (AS-i) Overview

AS-Interface (AS-i) er en simpel og effektiv feltbusløsning designet til at forbinde sensorer og aktuatorer i automatiseringssystemer. AS-i er kendt for sin nemme installation og vedligeholdelse, hvilket gør det til et populært valg i industrielle applikationer.

10.4.1 Introduktion

AS-i blev udviklet som en økonomisk og brugervenlig måde at forbinde enheder som sensorer, aktuatorer og PLC'er (Programmable Logic Controllers). AS-i er kendetegnet ved en to-leder fladkabel teknologi, som både overfører data og leverer strøm til tilsluttede enheder. Denne enkelhed gør det muligt for AS-i at reducere omkostningerne og kompleksiteten ved installation og vedligeholdelse.

10.4.2 Layer 1 – The Physical Layer

Det fysiske lag i AS-i definerer de elektriske og mekaniske egenskaber ved netværksforbindelserne. Dette inkluderer specifikationer for kabler, stik og

elektriske signaler, der bruges til at overføre data og strøm mellem enheder. AS-i bruger et fladkabelsystem, som er nemt at installere og tilslutte, hvilket minimerer fejl og reducerer installationsomkostningerne.

10.4.3 Layer 2 – The Data Link Layer

Datalink-laget håndterer pålidelig overførsel af data mellem enheder på AS-i netværket. Dette lag sikrer korrekt adressering, fejlregistrering og dataintegritet ved hjælp af cyklisk redundanskontrol (CRC). Datalink-laget koordinerer også kommunikationen mellem master- og slaveenheder, hvilket sikrer synkroniseret dataudveksling.

10.4.4 Operating Characteristics

AS-i systemet er designet til at fungere under forskellige industrielle forhold og tilbyder robusthed og pålidelighed. Nogle af de vigtigste driftskarakteristika inkluderer:

- **Fejltolerance:** AS-i netværk er modstandsdygtige over for elektrisk støj og andre forstyrrelser, hvilket sikrer stabil og pålidelig kommunikation.
- **Modularitet:** AS-i enheder kan nemt tilføjes eller fjernes uden at forstyrre det overordnede system, hvilket giver fleksibilitet ved ændringer eller udvidelser.
- **Diagnosticering:** AS-i systemet inkluderer diagnosticeringsfunktioner, der gør det muligt at identificere og rette fejl hurtigt og effektivt.

10.4.5 Troubleshooting

Fejlfinding i AS-i systemer er designet til at være enkel og effektiv, takket være indbyggede diagnostikværktøjer og funktioner.

Introduktion

Dette afsnit giver en introduktion til de grundlæggende principper for fejlfinding i AS-i netværk. Fejlfinding er en afgørende del af vedligeholdelsen og sikrer, at systemet fungerer korrekt og pålideligt.

Tools of the Trade

Der findes flere værktøjer til rådighed for fejlfinding i AS-i systemer. Nogle af de mest anvendte værktøjer inkluderer:

- **AS-i Master Monitor:** Dette værktøj bruges til at overvåge og diagnosticere kommunikationen mellem master- og slaveenheder.

- **Handheld Testers:** Bærbare enheder, der kan tilsluttes direkte til AS-i netværket for at udføre diagnostik og fejlfindingsopgaver.

- **Software Tools:** Specialiseret software, der kan bruges til at analysere netværksdata og identificere problemer.

10.4.6 Opsummering

AS-Interface (AS-i) er en effektiv og brugervenlig løsning til netværk af sensorer og aktuatorer i industrielle applikationer. Ved at forstå de forskellige lag i AS-i protokollen og have de rette værktøjer til fejlfinding, kan teknikere sikre, at deres AS-i netværk fungerer optimalt og pålideligt.

10.5 IO-Link

IO-Link er en standardiseret I/O-teknologi, der giver mulighed for kommunikation mellem sensorer og aktuatorer samt kontrolsystemer. Denne teknologi forbedrer diagnoser og parametrering af enheder, hvilket øger effektiviteten og fleksibiliteten i automatiseringssystemer.

10.5.1 Purpose of Technology

Formålet med IO-Link teknologien er at levere en åben standardiseret grænseflade til kommunikation med intelligente sensorer og aktuatorer. Dette muliggør ikke kun overførsel af procesdata, men også diagnostik og parametreringsinformation, hvilket resulterer i forbedret vedligeholdelse og procesoptimering.

10.5.2 Positioning within the Automation Hierarchy

IO-Link er placeret på det laveste niveau i automatiseringshierarkiet og fungerer som en point-to-point kommunikationsprotokol mellem en IO-Link master og forskellige IO-Link enheder. Det integreres problemfrit med eksisterende feltbusser og industrielle Ethernet-netværk, hvilket sikrer dataoverførsel fra feltniveau til kontrolniveau.

10.5.3 Wiring, Connectors, and Power

IO-Link bruger standard industrielle kabler og stik til forbindelser, hvilket gør installationen enkel og omkostningseffektiv. Et standard 3-leder kabel bruges til både data og strøm, hvilket eliminerer behovet for specialkabler og reducerer ledningsomkostningerne.

10.5.4 Communication Features of IO-Link

IO-Link kommunikationsprotokollen tilbyder flere avancerede funktioner:

- **Automatisk Device Identification:** IO-Link enheder identificeres automatisk af masteren, hvilket forenkler opsætningen.
- **Diagnostik:** IO-Link muliggør løbende overvågning og diagnostik af enheder, hvilket hjælper med at identificere og løse problemer hurtigt.
- **Parametrering:** Enheder kan parametreres via IO-Link, hvilket gør det muligt at justere indstillinger uden fysisk adgang til enhederne.

10.5.5 Role of a Master

IO-Link masteren fungerer som en kommunikationshub mellem kontrolsystemet og IO-Link enhederne. Den samler data fra flere IO-Link enheder og sender dem videre til kontrolsystemet, samtidig med at den sender kontrolkommandoer til enhederne. Masteren håndterer også diagnostik og parametrering af tilsluttede enheder.

10.5.6 IO-Link Configuration

Konfiguration af IO-Link enheder udføres typisk via IO-Link masteren ved hjælp af softwareværktøjer. Disse værktøjer giver et brugervenligt interface til at indstille parametre, overvåge status og diagnosticere fejl. Konfigurationsdata kan gemmes og genanvendes, hvilket forenkler opsætning af nye enheder eller udskiftning af defekte enheder.

10.5.7 Mapping to Fieldbuses and System Integration

IO-Link kan integreres med forskellige feltbusser og industrielle Ethernetnetværk ved hjælp af gateways eller IO-Link mastere med indbyggede feltbusinterfaces. Dette muliggør problemfri dataoverførsel fra IO-Link enheder til overordnede kontrolsystemer, hvilket forbedrer dataudnyttelse og systemintegration.

10.5.8 Implementation and Engineering Support

Implementering af IO-Link teknologien understøttes af omfattende dokumentation og softwareværktøjer fra producenter og standardiseringsorganer. Disse ressourcer hjælper med design, opsætning og vedligeholdelse af IO-Link systemer. Desuden tilbyder mange leverandører teknisk support og træningsprogrammer for at sikre en vellykket implementering.

10.5.9 Test and Certification

IO-Link enheder og systemer gennemgår test og certificering for at sikre kompatibilitet og pålidelighed. Standardiseringsorganer som IO-Link Consortium tilbyder certificeringsprogrammer, der sikrer, at produkter overholder de nødvendige standarder og fungerer korrekt i forskellige anvendelser.

10.5.10 Profiles

IO-Link understøtter forskellige profiler, som definerer specifikke funktioner og parametre for forskellige typer af enheder, såsom sensorer, aktuatorer og komplekse enheder. Disse profiler sikrer standardisering og interoperabilitet mellem forskellige producenters enheder.

10.5.11 Functional Safety

IO-Link teknologien inkluderer også funktioner til funktionel sikkerhed, hvilket gør det muligt at bruge IO-Link i applikationer, hvor sikkerhed er kritisk. Dette inkluderer sikkerhedsprotokoller og redundansmekanismer, der sikrer, at systemerne fungerer sikkert og pålideligt selv under fejlforhold.

10.6 KepServerEX

10.6.1 Introduktion til KepServerEX

En kort introduktion til hvad Kepware KepServerEX er, og hvorfor det er vigtigt i industrielle applikationer.

10.6.2 Hvad er KepServerEX?

10.6.3 Definition af KepServerEX

En grundig forklaring af KepServerEX, herunder dets funktion som en industristandard OPC-server og dataintegrationsplatform.

10.6.4 Historie og Udvikling

En kort gennemgang af KepServerEX's historie og hvordan det har udviklet sig over tid.

10.6.5 Markedets Position

Diskussion om KepServerEX's rolle og position på markedet i forhold til andre lignende produkter.

10.6.6 Funktioner og Kapaciteter i KepServerEX

10.6.7 Grundlæggende Funktioner

En detaljeret gennemgang af de vigtigste funktioner i KepServerEX, såsom OPC-kommunikation, datalogging, protokolkonvertering, og sikkerhedsaspekter.

10.6.8 Udvidede Funktioner

Uddybning af avancerede funktioner som scripting, skalerbarhed, og IoT-integration.

Scripting

Hvordan scripting bruges i KepServerEX til at automatisere processer.

IoT Gateways

Brugen af IoT Gateways i KepServerEX til at integrere med industrielle IoT-enheder.

10.6.9 Hvordan KepServerEX Anvendes i Industrien

10.6.10 Integration med Kontrolsystemer

Eksempler på, hvordan KepServerEX integreres med forskellige kontrolsystemer, såsom PLC'er og DCS'er.

10.6.11 Anvendelse i SCADA og MES

Hvordan KepServerEX bruges sammen med SCADA- og MES-systemer for at forbedre produktionsprocesser.

10.6.12 Eksempler fra Industrien

Reelle eksempler på praktisk brug af KepServerEX i forskellige brancher.

10.6.13 Opsætning og Konfiguration af KepServerEX

10.6.14 Installation af KepServerEX

Trin-for-trin vejledning i installationen af KepServerEX.

10.6.15 Grundlæggende Konfiguration

Hvordan man opsætter og konfigurerer KepServerEX for at kommunikere med forskellige industrielle enheder og systemer.

OPC-konfiguration

Opsætning af OPC-kommunikation i KepServerEX.

Sikkerhedsindstillinger

Konfiguration af sikkerhedsaspekter i KepServerEX for at sikre dataintegritet og beskyttelse.

10.6.16 Integration af KepServerEX med Andre Systemer

10.6.17 Integration med PLC'er

Forklaring af, hvordan KepServerEX kan integreres med PLC-systemer.

10.6.18 Integration med DCS og SCADA

Beskrivelse af integration med DCS- og SCADA-systemer.

10.6.19 Integration med Cloud-platforme

Hvordan KepServerEX kan forbindes til cloud-platforme for dataanalyse og lagring.

10.6.20 Fordele og Udfordringer ved at Bruge KepServerEX

10.6.21 Fordele

Diskuter de fordele, KepServerEX bringer til industrielle applikationer.

10.6.22 Udfordringer

Eventuelle udfordringer eller overvejelser, der skal tages i betragtning ved brug af KepServerEX.

10.6.23 Eksempler på Anvendelse af KepServerEX i Forskellige Brancher

10.6.24 Fremstillingsindustrien

Hvordan KepServerEX bruges i fremstillingsindustrien for at forbedre produktionseffektiviteten.

10.6.25 Energisektoren

Eksempler på KepServerEX's anvendelse i energisektoren til dataovervågning og kontrol.

10.6.26 Vandbehandling

Anvendelse af KepServerEX i vandbehandlingsanlæg for at sikre kontinuerlig overvågning og styring.

10.6.27 Bygningsteknologi

Hvordan KepServerEX kan integreres i bygningsautomatiseringssystemer.

10.6.28 Avancerede Funktioner i KepServerEX

10.6.29 Scripting og Automatisering

Hvordan scripting i KepServerEX kan bruges til at automatisere komplekse processer.

10.6.30 Skalerbarhed i KepServerEX

Diskussion om skalerbarhed og hvordan KepServerEX kan tilpasses voksende netværkskrav.

10.6.31 IoT-integration

Hvordan KepServerEX understøtter integration med IoT-enheder og systemer.

10.7 OPC

Del VI

Avancerede Netværksapplikationer

Kapitel 11

Cisco Packet Tracer Netværksopgaver

Introduktion til Cisco Packet Tracer

Cisco Packet Tracer er et kraftfuldt netværkssimuleringsværktøj udviklet af Cisco Systems. Det er designet til at hjælpe studerende og netværksprofessionelle med at lære om og praktisere netværkskonfigurationer, fejlfinding og netværksdesign uden behov for fysisk netværkshardware. Packet Tracer giver brugerne mulighed for at oprette komplekse netværksscenarier og simulere deres funktioner i et virtuelt miljø.

Hvad er Cisco Packet Tracer?

Cisco Packet Tracer er en dynamisk, visuel simuleringsapplikation, der gør det muligt for brugerne at opbygge, konfigurere og analysere netværkstopologier. Værktøjet understøtter en bred vifte af netværksenheder, herunder routere, switches, computere og forskellige IoT-enheder, der alle kan simuleres i realtid.

Funktioner og Anvendelser

Cisco Packet Tracer er kendt for sine mange funktioner, herunder:

- **Netværkssimulering:** Brugere kan simulere komplekse netværksscenarier med flere enheder og forbindelser. Dette giver mulighed for at forstå, hvordan netværksprotokoller fungerer og interagerer.
- **Interaktiv læring:** Packet Tracer bruges ofte i undervisningsmiljøer til at hjælpe studerende med at forstå netværkskonfigurationer gennem praktiske øvelser. Studerende kan eksperimentere med netværksopsætninger uden risikoen for at beskadige fysisk udstyr.

- **Real-time feedback:** Når brugere opsætter netværkskonfigurationer, giver Packet Tracer real-time feedback, hvilket hjælper med at identificere og rette fejl undervejs.

- **IoT Integration:** Packet Tracer understøtter også Internet of Things (IoT), hvilket giver brugerne mulighed for at simulere og lære om, hvordan IoT-enheder interagerer i netværk.

- **Multi-user funktion:** Værktøjet understøtter multi-user netværksopsætninger, hvor flere brugere kan arbejde på samme projekt samtidig, hvilket fremmer samarbejde og gruppeopgaver.

Hvordan bruges Cisco Packet Tracer?

Cisco Packet Tracer anvendes i vid udstrækning i både uddannelses- og professionelle miljøer. Nogle af de mest almindelige anvendelser inkluderer:

- **Uddannelse og Certificering:** Cisco Packet Tracer er en central del af Cisco Networking Academy-programmet, som forbereder studerende til Cisco-certificeringer som CCNA (Cisco Certified Network Associate). Studerende bruger værktøjet til at simulere netværksscenarier og forberede sig til eksamener.

- **Netværksdesign og Fejlfinding:** Netværksprofessionelle bruger Packet Tracer til at designe og afprøve netværkskonfigurationer før implementering i den virkelige verden. Det giver også mulighed for at fejlsøge netværksproblemer i et kontrolleret miljø.

- **Eksperimenter med nye teknologier:** Packet Tracer giver mulighed for at eksperimentere med nye netværksteknologier og -protokoller uden at kræve fysisk udstyr, hvilket gør det ideelt for at holde sig ajour med den nyeste udvikling inden for netværksteknologi.

Fordele ved Cisco Packet Tracer

Nogle af de vigtigste fordele ved at bruge Cisco Packet Tracer inkluderer:

- **Koste- og tidsbesparelse:** Det eliminerer behovet for fysisk udstyr, hvilket reducerer omkostningerne og den tid, der kræves for at opsætte og vedligeholde netværk.

- **Tilgængelighed:** Packet Tracer er gratis for alle, der deltager i Cisco Networking Academy-kurser, og det er også tilgængeligt for selvstuderende.

- **Fleksibilitet:** Værktøjet understøtter flere platforme, herunder Windows, Linux, og macOS, og kan bruges både offline og online.

Cisco Packet Tracer er et uundværligt værktøj for alle, der ønsker at lære om netværksteknologi, og det giver en omfattende platform til at udvikle og teste netværksfærdigheder i et sikkert og kontrolleret miljø.

11.1 Hub

11.1.1 Simpel Hub Opsætning

Mål: Forstå hvordan en hub fungerer ved at oprette et simpelt netværk og teste forbindelsen mellem flere computere.

Opgavebeskrivelse:

1. **Forbind tre computere til en hub:**
    - Brug Ethernet-kabler til at forbinde PC1, PC2, og PC3 til en hub i Cisco Packet Tracer.
2. **Konfigurer IP-adresser:**
    - PC1: 192.168.1.2/24
    - PC2: 192.168.1.3/24
    - PC3: 192.168.1.4/24
3. **Test forbindelsen:**
    - Brug kommandoen `ping` fra PC1 til PC2 og PC3 for at verificere, at alle enheder kan kommunikere.

11.1.2 Hub Kommunikation med Fire Computere

Mål: Udforsk hvordan en hub håndterer datatrafik, når flere computere er tilsluttet og kommunikerer samtidigt.

Opgavebeskrivelse:

1. **Forbind fire computere til en hub:**
    - Brug Ethernet-kabler til at forbinde PC1, PC2, PC3, og PC4 til en hub i Cisco Packet Tracer.
2. **Konfigurer IP-adresser:**
    - PC1: 192.168.2.2/24

- PC2: 192.168.2.3/24
- PC3: 192.168.2.4/24
- PC4: 192.168.2.5/24

3. **Test forbindelsen:**
   - Pinge fra PC1 til alle de andre computere (PC2, PC3, PC4).
   - Notér, hvordan dataene sendes til alle computere tilsluttet hubben, og hvordan de reagerer på de indkomne pakker.

11.1.3 Hub og Broadcast Trafik

Mål: Lær hvordan en hub håndterer broadcast-trafik i et netværk.

Opgavebeskrivelse:

1. **Forbind tre computere til en hub:**
   - Brug Ethernet-kabler til at forbinde PC1, PC2, og PC3 til en hub i Cisco Packet Tracer.
2. **Konfigurer IP-adresser:**
   - PC1: 192.168.3.2/24
   - PC2: 192.168.3.3/24
   - PC3: 192.168.3.4/24
3. **Send en broadcast-ping:**
   - Fra PC1, send en broadcast-ping til 192.168.3.255.
   - Observer hvordan hubben håndterer denne broadcast, og hvordan alle computere modtager ping-forespørgslen.

11.2 Switch

11.2.1 Grundlæggende Netværksforbindelser

Grundlæggende Switch Forbindelse

Mål: Lær at forbinde to computere til en switch og teste deres forbindelse.

Opgavebeskrivelse:

1. **Forbind to computere til en switch:**
   - Forbind PC1 og PC2 til en switch ved hjælp af Ethernet-kabler.

2. **Konfigurer IP-adresser:**
    - PC1: 192.168.1.2/24
    - PC2: 192.168.1.3/24
3. **Test forbindelsen:**
    - Pinge fra PC1 til PC2 for at verificere forbindelsen.

Tilføjelse af en en ekstra Computer

Mål: Lær at udvide netværket ved at tilføje en tredje computer til switchen.

Opgavebeskrivelse:

1. **Forbind en tredje computer til switchen:**
    - Forbind PC3 til switchen med et Ethernet-kabel.
2. **Konfigurer IP-adressen:**
    - PC3: 192.168.1.4/24
3. **Test forbindelsen:**
    - Pinge fra PC3 til PC1 og PC2 for at sikre, at alle enheder kan kommunikere.

11.2.2 Switch Funktionalitet og Læring

Observer MAC-adresse Tabel

Mål: Lær at observere, hvordan en switch lærer MAC-adresser.

Opgavebeskrivelse:

1. **Forbind to computere til switchen:**
    - Forbind PC1 og PC2 til en switch og konfigurer deres IP-adresser.
2. **Udfør ping:**
    - Pinge fra PC1 til PC2.
3. **Observer MAC-adresse tabellen:**
    - Brug kommandoen `show mac address-table` på switchen for at observere MAC-adresserne.

Fjernelse og Genforbindelse

Mål: Forstå hvordan switchen opdaterer sin MAC-adresse tabel.

Opgavebeskrivelse:

1. **Fjern netværkskablet fra en computer:**
   - Fjern kablet fra en af de tilsluttede computere og tilslut det igen til en anden port på switchen.
2. **Udfør en ny ping:**
   - Pinge igen mellem PC1 og PC2.
3. **Observer MAC-adresse tabellen:**
   - Brug kommandoen `show mac address-table` for at se, hvordan tabellen er opdateret.

11.2.3 Grundlæggende Switch Konfiguration

Skift Switchens Hostname

Mål: Lær at ændre switchens hostname.

Opgavebeskrivelse:

1. **Log ind på switchens CLI:**
   - Log ind på switchen via CLI.
2. **Ændr hostname:**
   - Brug kommandoen `hostname` til at ændre navnet til "Switch1".
3. **Gem konfigurationen:**
   - Brug `write memory` for at gemme ændringerne.

Sæt en Adgangskode på Konsoladgang

Mål: Sikre konsoladgang ved at tilføje en adgangskode.

Opgavebeskrivelse:

1. **Log ind på switchens CLI:**
   - Log ind på switchen via CLI.
2. **Sæt en adgangskode:**

- Brug kommandoerne `line console 0` og `password cisco` for at indstille adgangskoden.

3. **Aktivér adgangskode:**

   - Brug kommandoen `login` for at aktivere adgangskodebeskyttelsen.

11.2.4 Udvidet Switch Konfiguration

Navngivning af Switch

Mål: Forstå grundlæggende navngivning og konfiguration.

Opgavebeskrivelse:

1. **Log ind på switchens CLI:**

   - Log ind på switchen via CLI.

2. **Navngiv switchen:**

   - Brug kommandoen `hostname` for at navngive switchen.

3. **Verificér navnet:**

   - Brug kommandoen `show running-config` for at tjekke det nye navn.

Indstilling af System Beskrivelse

Mål: Tilføj en beskrivelse af systemet for dokumentationsformål.

Opgavebeskrivelse:

1. **Log ind på switchens CLI:**

   - Log ind på switchen via CLI.

2. **Tilføj en besked:**

   - Brug kommandoen `banner motd` for at tilføje en meddelelse, der vises ved login.

3. **Test beskeden:**

   - Log ud og log ind igen for at se den nye besked.

11.2.5 Spanning Tree Protocol (STP)

Aktiver STP på en Switch

Mål: Introduktion til Spanning Tree Protocol.

Opgavebeskrivelse:

1. **Log ind på switchens CLI:**
   - Log ind på switchen via CLI.
2. **Aktiver STP:**
   - Brug kommandoen `spanning-tree` for at aktivere STP.
3. **Verificér STP status:**
   - Brug `show spanning-tree` for at se STP-status.

Forbind To Switches og Observer STP

Mål: Forstå, hvordan STP arbejder i et simpelt netværk.

Opgavebeskrivelse:

1. **Forbind to switches med to kabler:**
   - Forbind to switches med to kabler for at simulere en loop.
2. **Observer STP:**
   - Brug kommandoen `show spanning-tree` på begge switches for at se, hvordan STP blokerer en af forbindelserne.

11.2.6 Redundans og Failover

Redundant Forbindelse til Enkelt Switch

Mål: Forstå enkel redundans med to links.

Opgavebeskrivelse:

1. **Forbind en computer til en switch med to kabler:**
   - Forbind en computer til switchen med to Ethernet-kabler.
2. **Fjern og tilslut et kabel:**
   - Fjern det ene kabel og observer forbindelsens opførsel.
   - Tilslut kablet igen og observer, hvordan forbindelsen genoprettes.

Redundant Forbindelse med To Switche

Mål: Forstå failover mellem to switche.

Opgavebeskrivelse:

1. **Forbind to switche med to kabler:**
    - Forbind to switche med to Ethernet-kabler.
2. **Fjern et kabel:**
    - Fjern et kabel og observer, hvordan netværket håndterer fejlen.

11.2.7 EtherChannel Konfiguration

Opret en EtherChannel Forbindelse

Mål: Lær at oprette en EtherChannel mellem to switches.

Opgavebeskrivelse:

1. **Forbind to switches med to kabler:**
    - Forbind to switches med to Ethernet-kabler.
2. **Konfigurer EtherChannel:**
    - Brug kommandoen `channel-group 1 mode on` på begge switches.
3. **Test forbindelsen:**
    - Test forbindelsen mellem to computere på hver switch for at sikre, at EtherChannel fungerer.

Verificér EtherChannel Konfiguration

Mål: Verificér en fungerende EtherChannel.

Opgavebeskrivelse:

1. **Verificér EtherChannel status:**
    - Brug kommandoen `show etherchannel summary` for at se status på EtherChannel.
2. **Test redundans:**
    - Afbryd et af kablerne og kontrollér, at forbindelsen stadig fungerer.

11.3 Router

11.3.1 Opret en Grundlæggende Router-til-Router Forbindelse

Mål: Lær at forbinde to routere og konfigurer deres grænseflader.

Opgavebeskrivelse:

1. **Forbind to routere med et serielt kabel:**
   - Brug et serielt kabel til at forbinde Router1 og Router2.
2. **Konfigurer IP-adresser på serielle interfaces:**
   - Router1: 192.168.1.1/30
   - Router2: 192.168.1.2/30
3. **Test forbindelsen:**
   - Pinge fra Router1 til Router2 for at sikre, at der er forbindelse.

11.3.2 Grundlæggende Router Konfiguration

Skift Routerens Hostname

Mål: Lær at ændre routerens hostname.

Opgavebeskrivelse:

1. **Log ind på routerens CLI:**
   - Log ind på Router1 via CLI.
2. **Ændr hostname:**
   - Brug kommandoen `hostname` til at ændre navnet til "Router1".
3. **Gem konfigurationen:**
   - Brug `write memory` for at gemme ændringerne.

Sæt en Adgangskode på Konsoladgang

Mål: Sikre konsoladgang ved at tilføje en adgangskode.

Opgavebeskrivelse:

1. **Log ind på routerens CLI:**

- Log ind på Router1 via CLI.

2. **Sæt en adgangskode:**
   - Brug kommandoerne `line console 0` og `password cisco` for at indstille adgangskoden.
3. **Aktivér adgangskode:**
   - Brug kommandoen `login` for at aktivere adgangskodebeskyttelsen.

11.3.3 Enkel Router Routing

Konfigurér en Statisk Route

Mål: Lær at opsætte en simpel statisk route på en router.

Opgavebeskrivelse:

1. **Log ind på routerens CLI:**
   - Log ind på Router1 via CLI.
2. **Konfigurér en statisk route:**
   - Brug kommandoen `ip route 192.168.2.0 255.255.255.0 192.168.1.2` for at opsætte en statisk route til et tilstødende netværk.
3. **Test ruten:**
   - Pinge fra Router1 til et device på 192.168.2.0-netværket for at verificere ruten.

11.3.4 Grundlæggende Interface Konfiguration

Konfigurér et Ethernet Interface

Mål: Lær at konfigurere en grundlæggende IP-adresse på et Ethernet interface.

Opgavebeskrivelse:

1. **Log ind på routerens CLI:**
   - Log ind på Router1 via CLI.
2. **Konfigurér IP-adresse på Ethernet-interface:**
   - Brug kommandoen `interface gigabitEthernet 0/0` og sæt IP-adressen til 192.168.10.1/24.

3. **Aktivér interface:**
    - Brug kommandoen `no shutdown` for at aktivere interfacet.
4. **Test forbindelsen:**
    - Pinge til en tilsluttet enhed for at sikre, at interfacet fungerer korrekt.

11.3.5 Enkel Router Sikkerhed

Sæt en Enable Password

Mål: Sikre routerens privilegerede EXEC-tilstand ved at tilføje en enable-adgangskode.

Opgavebeskrivelse:

1. **Log ind på routerens CLI:**
    - Log ind på Router1 via CLI.
2. **Sæt en enable-adgangskode:**
    - Brug kommandoen `enable secret cisco` for at indstille en adgangskode til privilegeret adgang.
3. **Verificér adgangskoden:**
    - Log ud og log ind igen for at teste den nye adgangskode.

11.3.6 Grundlæggende Netværks-ID og Subnetting

Identificér Netværks-ID fra en IP-adresse

Mål: Lær at finde netværks-ID'et for en given IP-adresse og subnetmaske.

Opgavebeskrivelse:

1. **Givet IP-adressen 192.168.10.14 med subnetmasken 255.255.255.0:**
    - Beregn netværks-ID'et for denne IP-adresse ved hjælp af subnetmasken.
2. **Skriv netværks-ID'et ned:**
    - Notér den del af IP-adressen, der repræsenterer netværket.

Bestem Antallet af Hosts i et Subnet

Mål: Forstå hvordan man beregner antallet af mulige hosts i et subnet.

Opgavebeskrivelse:

1. **Givet subnetmasken 255.255.255.224:**
    - Beregn antallet af brugbare IP-adresser (hosts) i dette subnet.
2. **Skriv antallet af hosts ned:**
    - Notér, hvor mange IP-adresser der kan tildeles til enheder i dette subnet.

Opdel Et Simpelt Netværk i To Subnets

Mål: Lær at opdele et lille netværk i to mindre subnets.

Opgavebeskrivelse:

1. **Givet netværket 192.168.10.0/24:**
    - Del dette netværk op i to lige store subnets.
2. **Skriv subnet-ID'er og nye subnetmasker ned:**
    - Notér de to subnet-ID'er og deres tilhørende subnetmasker.

Tildel IP-adresser inden for et Simpelt Subnet

Mål: Praktisk anvendelse af IP-adressering inden for et givet subnet.

Opgavebeskrivelse:

1. **Givet subnet-ID'et 192.168.10.0/28:**
    - Tildel IP-adresser til fire enheder inden for dette subnet.
2. **Identificér broadcast-adressen:**
    - Bestem broadcast-adressen for dette subnet.

11.4 VLAN

11.4.1 Opret en Grundlæggende VLAN

Mål: Lær at oprette en grundlæggende VLAN på en switch og tildele porte til VLAN'et.

Opgavebeskrivelse:

1. **Opret VLAN 10 for "Production"på en switch:**
    - Log ind på switchens CLI.
    - Brug kommandoen `vlan 10` for at oprette VLAN 10.
2. **Navngiv VLAN 10:**
    - Brug kommandoen `name Production` for at navngive VLAN'et.
3. **Tildel porte til VLAN 10:**
    - Brug kommandoen `interface range gigabitEthernet 0/1 - 2` for at vælge portene.
    - Brug kommandoen `switchport access vlan 10` for at tildele portene til VLAN 10.
4. **Verificér VLAN-konfigurationen:**
    - Brug kommandoen `show vlan brief` for at verificere, at portene er korrekt tildelt til VLAN 10.

11.4.2 Grundlæggende VLAN Routing

Konfigurér Inter-VLAN Routing

Mål: Lær at konfigurere routing mellem VLAN'er ved hjælp af en router-on-a-stick-konfiguration.

Opgavebeskrivelse:

1. **Konfigurér underinterface på routeren for "Production":**
    - Log ind på routerens CLI.
    - Brug kommandoen `interface gigabitEthernet 0/0.10` for at oprette et underinterface for VLAN 10 ("Production").
    - Tildel IP-adressen 192.168.10.1/24 til underinterfacet.
    - Brug kommandoen `encapsulation dot1Q 10` for at angive VLAN ID.
2. **Konfigurér underinterface på routeren for "Finance":**
    - Opret et underinterface `gigabitEthernet 0/0.20` for VLAN 20 ("Finance").
    - Tildel IP-adressen 192.168.20.1/24.
    - Brug `encapsulation dot1Q 20`.
3. **Test Inter-VLAN Routing:**
    - Pinge fra en enhed i VLAN 10 ("Production") til en enhed i VLAN 20 ("Finance") for at sikre, at der er routing mellem VLAN'erne.

11.4.3 Sikkerhed i VLAN

Opret og Konfigurér et Management VLAN

Mål: Lær at oprette et dedikeret management VLAN for at sikre adgang til switchens administrative interface.

Opgavebeskrivelse:

1. **Opret VLAN 99 som "Management":**
   - Log ind på switchens CLI.
   - Brug kommandoen `vlan 99` for at oprette management VLAN'et.
2. **Tildel en IP-adresse til VLAN 99 interface:**
   - Brug kommandoen `interface vlan 99`.
   - Tildel IP-adressen 192.168.99.1/24 til VLAN 99 interface.
3. **Sæt en standard gateway:**
   - Brug kommandoen `ip default-gateway 192.168.99.254`.
4. **Verificér adgang til management VLAN:**
   - Pinge til switchens management IP fra en computer i netværket for at sikre, at VLAN 99 fungerer korrekt.

Konfigurér Port Security på et VLAN

Mål: Lær at anvende port security på et VLAN for at forbedre netværkssikkerheden.

Opgavebeskrivelse:

1. **Aktivér port security på en switch port i "Guest"VLAN:**
   - Log ind på switchens CLI.
   - Vælg en port med `interface gigabitEthernet 0/1`.
   - Brug kommandoen `switchport mode access` for at sætte porten i access mode.
   - Aktivér port security med `switchport port-security`.
2. **Konfigurér en maksimal grænse for MAC-adresser:**
   - Brug kommandoen `switchport port-security maximum 2` for at tillade maksimalt to MAC-adresser.

3. **Indstil en port security action:**

    - Brug kommandoen `switchport port-security violation shutdown` for at lukke porten, hvis sikkerheden krænkes.

4. **Verificér port security:**

    - Brug kommandoen `show port-security interface gigabitEthernet 0/1` for at verificere indstillingerne.

11.5 NAT (Network Address Translation)

11.5.1 Konfigurér Statisk NAT

Mål: Lær at konfigurere statisk NAT for at mappe en enkelt intern IP-adresse til en enkelt ekstern IP-adresse.

Opgavebeskrivelse:

1. **Log ind på routerens CLI:**

    - Log ind på Router1 via CLI.

2. **Konfigurér en indvendig lokal IP-adresse:**

    - Brug kommandoen `ip nat inside source static 192.168.10.10 203.0.113.10` for at mappe den interne IP-adresse (192.168.10.10) til en ekstern IP-adresse (203.0.113.10).

3. **Angiv interfacet som indvendig eller udvendig:**

    - Brug kommandoen `interface gigabitEthernet 0/1` for at vælge det interne interface og angiv det som "inside"ved at bruge `ip nat inside`.
    - Brug kommandoen `interface gigabitEthernet 0/0` for at vælge det eksterne interface og angiv det som "outside"ved at bruge `ip nat outside`.

4. **Test den statiske NAT-konfiguration:**

    - Pinge den eksterne IP-adresse (203.0.113.10) fra en ekstern enhed og verificér, at forbindelsen oversættes korrekt til den interne IP-adresse.

11.5.2 Konfigurér Dynamisk NAT

Mål: Lær at konfigurere dynamisk NAT for at oversætte flere interne IP-adresser til en pool af eksterne IP-adresser.

Opgavebeskrivelse:

1. **Opret en pool af eksterne IP-adresser:**
   - Brug kommandoen `ip nat pool NAT-POOL 203.0.113.10 203.0.113.15 netmask 255.255.255.248` for at definere en pool af eksterne IP-adresser.
2. **Konfigurér en access-list for indvendige IP-adresser:**
   - Brug kommandoen `access-list 1 permit 192.168.10.0 0.0.0.255` for at tillade interne IP-adresser fra 192.168.10.0/24-netværket.
3. **Anvend dynamisk NAT:**
   - Brug kommandoen `ip nat inside source list 1 pool NAT-POOL` for at anvende dynamisk NAT ved at oversætte de tilladte interne IP-adresser til poolen af eksterne IP-adresser.
4. **Angiv interfacet som indvendig eller udvendig:**
   - Brug kommandoen `interface gigabitEthernet 0/1` for at vælge det interne interface og angiv det som "inside"ved at bruge `ip nat inside`.
   - Brug kommandoen `interface gigabitEthernet 0/0` for at vælge det eksterne interface og angiv det som "outside"ved at bruge `ip nat outside`.
5. **Test den dynamiske NAT-konfiguration:**
   - Pinge fra en intern enhed til en ekstern adresse og verificér, at NAT oversættelsen sker fra den interne til den eksterne IP-pool.

11.6 DNS (Domain Name System)

11.6.1 Konfigurér en DNS-klient

Mål: Lær at konfigurere en computer til at bruge en specifik DNS-server for at oversætte domænenavne til IP-adresser.

Opgavebeskrivelse:

1. **Åbn netværksindstillinger på klientcomputeren:**

- Gå til netværksindstillinger på en Windows- eller Linux-maskine.

2. **Konfigurér DNS-serveradresse:**
   - Indtast DNS-serverens IP-adresse (f.eks. 8.8.8.8) i feltet for DNS-server.

3. **Test DNS-konfigurationen:**
   - Brug kommandoen `nslookup www.example.com` for at verificere, at DNS-klienten kan oversætte domænenavne til IP-adresser ved hjælp af den konfigurerede DNS-server.

11.6.2 Konfigurér en Grundlæggende DNS Server

Mål: Lær at opsætte en simpel DNS-server, der kan svare på forespørgsler for et specifikt domæne.

Opgavebeskrivelse:

1. **Installer DNS-server software:**
   - Installer en DNS-server software som BIND på en Linux-maskine eller brug indbygget DNS-server på en Windows Server-maskine.

2. **Konfigurér en zonefil for domænet "example.com":**
   - Opret en zonefil og tilføj en A-record for `www.example.com`, der peger på IP-adressen 192.168.1.10.

3. **Start DNS-serveren:**
   - Start DNS-serveren og sørg for, at den kører korrekt.

4. **Test DNS-serveren:**
   - Brug kommandoen `nslookup www.example.com` på en klientcomputer for at verificere, at DNS-serveren korrekt oversætter domænenavnet til den rigtige IP-adresse.

11.6.3 Konfigurér DNS Forwarding

Mål: Lær at konfigurere en DNS-server til at videresende forespørgsler til en anden DNS-server, hvis den ikke kan svare på forespørgslen selv.

Opgavebeskrivelse:

1. **Åbn DNS-serverens konfigurationsfil:**

- Åbn DNS-serverens konfigurationsfil (f.eks. `/etc/named.conf` for BIND).

2. **Tilføj en forwarding-indstilling:**
   - Tilføj en sektion for forwarding og angiv IP-adressen på den eksterne DNS-server (f.eks. 8.8.8.8).
3. **Genstart DNS-serveren:**
   - Genstart DNS-serveren for at anvende de nye indstillinger.
4. **Test DNS-forwarding:**
   - Forsøg at slå et domænenavn op, som DNS-serveren ikke har i sin zonefil, og verificér, at forespørgslen videresendes korrekt til den eksterne DNS-server.

11.6.4 Tilføj en CNAME Record

Mål: Lær at tilføje en CNAME (alias) record til en DNS-server for at pege et domænenavn til et andet domænenavn.

Opgavebeskrivelse:

1. **Åbn DNS-serverens zonefil for "example.com":**
   - Find og åbn zonefilen for `example.com`.
2. **Tilføj en CNAME record:**
   - Tilføj en CNAME record, så `mail.example.com` peger på `www.example.com`.
3. **Genstart DNS-serveren:**
   - Genstart DNS-serveren for at anvende ændringerne.
4. **Test CNAME record:**
   - Brug kommandoen `nslookup mail.example.com` for at sikre, at `mail.example.com` korrekt peger på IP-adressen til `www.example.com`.

11.7 DHCP (Dynamic Host Configuration Protocol)

11.7.1 Konfigurér en Grundlæggende DHCP Server

Mål: Lær at opsætte en grundlæggende DHCP-server for automatisk at tildele IP-adresser til klienter på netværket.

Opgavebeskrivelse:

1. **Log ind på routerens CLI:**
   - Log ind på Router1 via CLI.
2. **Konfigurér et DHCP-pool:**
   - Brug kommandoen `ip dhcp pool LAN` for at oprette en ny DHCP-pool med navnet "LAN".
   - Indstil netværksadressen ved hjælp af `network 192.168.10.0 255.255.255.0`.
   - Indstil standard gateway med `default-router 192.168.10.1`.
   - Angiv DNS-serveren med `dns-server 8.8.8.8`.
3. **Udsæt visse IP-adresser fra DHCP-poolen:**
   - Brug kommandoen `ip dhcp excluded-address 192.168.10.1 192.168.10.10` for at undgå, at disse adresser tildeles af DHCP-serveren.
4. **Verificér DHCP-konfigurationen:**
   - Brug kommandoen `show ip dhcp binding` for at se de IP-adresser, der er blevet tildelt af DHCP-serveren.
   - Test DHCP ved at tilslutte en klient til netværket og verificér, at klienten modtager en IP-adresse automatisk.

11.7.2 Konfigurér DHCP-udlejningstid

Mål: Lær at konfigurere udlejningstiden (lease time) for IP-adresser, der tildeles af DHCP-serveren.

Opgavebeskrivelse:

1. **Åbn DHCP-pool konfigurationen:**
   - Brug kommandoen `ip dhcp pool LAN` for at åbne DHCP-pool konfigurationen.
2. **Indstil udlejningstiden:**
   - Brug kommandoen `lease 2 12 00` for at indstille udlejningstiden til 2 dage og 12 timer.
3. **Verificér udlejningstiden:**
   - Brug kommandoen `show ip dhcp pool` for at se den konfigurerede udlejningstid.
   - Test ved at tilslutte en klient og verificere, at den modtager en IP-adresse med den angivne udlejningstid.

11.7.3 Konfigurér DHCP Reservation

Mål: Lær at konfigurere en DHCP-reservation for at sikre, at en bestemt enhed altid modtager den samme IP-adresse.

Opgavebeskrivelse:

1. **Find MAC-adressen på enheden:**
    - På klienten, brug kommandoen `ipconfig /all` (Windows) eller `ifconfig` (Linux) for at finde MAC-adressen.
2. **Konfigurér DHCP-reservation:**
    - På routeren, brug kommandoen `ip dhcp pool LAN`.
    - Indstil en reservation med kommandoen `host 192.168.10.50` og tilknyt MAC-adressen med `hardware-address <MAC-adresse>`.
3. **Verificér DHCP-reservationen:**
    - Test ved at genstarte netværksforbindelsen på klienten og verificér, at den modtager den reserverede IP-adresse.

11.7.4 Konfigurér en DHCP Relay Agent

Mål: Lær at konfigurere en DHCP Relay Agent for at videresende DHCP-forespørgsler fra klienter på forskellige netværk.

Opgavebeskrivelse:

1. **Log ind på routeren, der forbinder to netværk:**
    - Log ind på Router2 via CLI.
2. **Aktivér DHCP Relay Agent på det lokale interface:**
    - Brug kommandoen `interface gigabitEthernet 0/1`.
    - Indstil DHCP Relay Agent ved at bruge kommandoen `ip helper-address 192.168.20.1`, hvor `192.168.20.1` er IP-adressen på DHCP-serveren.
3. **Test DHCP-relay:**
    - Tilslut en klient til netværket på det lokale interface og verificér, at den modtager en IP-adresse fra DHCP-serveren på det fjerntliggende netværk.

Kapitel 12

Siemens

12.1 TIA Portal Netværksopgaver

Dette afsnit fokuserer på de centrale opgaver og procedurer, der er nødvendige for at konfigurere og vedligeholde netværksforbindelser inden for TIA Portal. Gennem en række trin-for-trin instruktioner vil læseren blive guidet igennem processerne for oprettelse af PROFINET-netværk, integration af enheder og fejlfinding af netværksproblemer.

Før du går i gang med opgaverne i dette afsnit, skal du læse og forstå de forskellige metoder for IP-adressering og konfiguration af PROFINET enhedsnavne i TIA Portal. Disse guides dækker statisk IP-adresse, DHCP og direkte konfiguration på enheden.

12.2 IP-adresse guide

Indstilling af statisk IP-adresse

Mål: Opsætning af en statisk IP-adresse på en Siemens PLC i TIA Portal.

Trin-for-trin Guide:

1. Åbn TIA Portal og vælg det relevante projekt.
2. Naviger til den PLC-enhed, du ønsker at konfigurere.
3. Klik på fanen *Properties*.
4. Vælg *Ethernet addresses* under *General*.
5. Under *Internet protocol version 4 (IPv4)*, vælg *Set IP address in the project*.
6. Indtast den ønskede IP-adresse og subnetmaske.

- **Use router:** Kryds dette felt af, hvis PLC'en skal kunne kommunikere med enheder uden for det lokale netværk. Indtast routerens IP-adresse.
- **PROFINET device name is set directly at the device:** Kryds dette felt af, hvis du vil sætte enhedsnavnet direkte på PLC'en i stedet for at konfigurere det i projektet.
- **Generate PROFINET device name automatically:** Kryds dette felt af, hvis du ønsker, at TIA Portal automatisk genererer et PROFINET enhedsnavn.

7. Klik på *OK* eller *Apply* for at gemme ændringerne.

Bemærk: Sørg for, at *IP-adressen* og *PROFINET device name* er unik inden for netværket og ikke konflikter med andre enheder.

Indstilling af IP-adresse via DHCP

Mål: Konfigurering af en Siemens PLC til at modtage en IP-adresse fra en DHCP-server.

Trin-for-trin Guide:

1. Åbn TIA Portal og vælg det relevante projekt.
2. Naviger til den PLC-enhed, du ønsker at konfigurere.
3. Klik på fanen *Properties*.
4. Vælg *Ethernet addresses* under *General*.
5. Under *Internet protocol version 4 (IPv4)*, vælg *IP address from DHCP server*.
    - **Mode:** Vælg *Use MAC address as client ID*, hvis PLC'en skal bruge sin MAC-adresse som klient-ID ved anmodning om en IP-adresse fra DHCP-serveren.
    - **Client ID:** Hvis du ønsker at specificere et brugerdefineret klient-ID, kan du indtaste det her. Kryds *Client ID can be changed during runtime* af, hvis dette ID skal kunne ændres under kørsel.
    - **PROFINET device name is set directly at the device:** Kryds dette felt af, hvis du vil sætte enhedsnavnet direkte på PLC'en i stedet for at konfigurere det i projektet.
    - **Generate PROFINET device name automatically:** Kryds dette felt af, hvis du ønsker, at TIA Portal automatisk genererer et PROFINET enhedsnavn.

6. Klik på *OK* eller *Apply* for at gemme ændringerne.

Bemærk: DHCP-serveren skal være korrekt konfigureret til at tildele IP-adresser inden for det ønskede netværk.

Note for Mode: Når du vælger *Use MAC address as client ID*, betyder det, at PLC'en automatisk bruger sin unikke MAC-adresse som klient-ID, når den anmoder om en IP-adresse fra en DHCP-server. Du skal ikke manuelt indtaste MAC-adressen; PLC'en bruger sin egen hardware-adresse, som allerede er indkodet i dens netværkskort.

Note for Client ID: Hvis du ønsker at specificere et brugerdefineret klient-ID i stedet for at bruge MAC-adressen, kan du indtaste det her. Dette kan være nyttigt i situationer, hvor du har brug for en mere meningsfuld identifikation for PLC'en i netværket. Hvis du sætter kryds ved *Client ID can be changed during runtime*, tillader du, at dette ID kan ændres dynamisk, mens systemet er i drift, hvilket giver fleksibilitet i netværksadministrationen.

Indstilling af IP-adresse direkte på enheden

Mål: Konfiguration af en Siemens PLC til at have en IP-adresse sat direkte på enheden.

Trin-for-trin Guide:

1. Åbn TIA Portal og vælg det relevante projekt.
2. Naviger til den PLC-enhed, du ønsker at konfigurere.
3. Klik på fanen *Properties*.
4. Vælg *Ethernet addresses* under *General*.
5. Under *Internet protocol version 4 (IPv4)*, vælg *IP address is set directly at the device*.
    - **PROFINET device name is set directly at the device:** Kryds dette felt af, hvis du vil sætte enhedsnavnet direkte på PLC'en i stedet for at konfigurere det i projektet.
    - **Generate PROFINET device name automatically:** Kryds dette felt af, hvis du ønsker, at TIA Portal automatisk genererer et PROFINET enhedsnavn.
6. Klik på *OK* eller *Apply* for at gemme ændringerne.

Bemærk: Når denne indstilling er valgt, skal IP-adressen konfigureres direkte på PLC-enheden via dens brugergrænseflade.

12.3 Oprettelse af PROFINET Netværk

Mål: Denne sektion guider dig gennem oprettelsen af et PROFINET netværk i Siemens TIA Portal, trin for trin.

Følg nedenstående trin for at sikre korrekt opsætning:

1. **Dokumentation og Planlægning:**
    - Opret en detaljeret plan for netværksopsætningen, herunder en liste over alle enheder, der skal tilføjes, deres tildelte IP-adresser, enhedsnavne og andre relevante konfigurationsdetaljer.
    - Gem planen i et let tilgængeligt format, f.eks. en PDF-fil eller et regneark.
    - Sørg for, at planen er opdateret og inkluderer eventuelle ændringer eller tilføjelser til netværket.
2. **Åbn TIA Portal og vælg det relevante projekt:**
    - Start TIA Portal-softwaren.
    - Opret et nyt projekt.
3. **Indsæt en fysisk S7-1200:**
    - Tilføj en fysisk S7-1200 PLC til projektet som den første enhed.
4. **Naviger til "Netværksoversigt":**
    - Gå til *Netværksoversigt* i projektets hovedmenu.
5. **Angiv unikt IP-adresseområde og "Device name":**
    - Højreklik på S7-1200 PLC'en og vælg *Properties*.
    - Definer et passende IP-adresseområde for netværket for at sikre, at alle enheder får unikke IP-adresser.
    - Giv S7-1200 PLC'en en unik IP-adresse.
    - Indtast et beskrivende *Device name*.
    - Dokumenter Device name, IP-adresse og MAC-adresse.
6. **Brug 'Drag and Drop' for at tilføje en ny simuleret S7-1500 PLC til netværket, og tildel unik IP-adresse:**
    - Træk den simulerede S7-1500 PLC fra enhedslisten ind i netværksoversigten ved hjælp af 'Drag and Drop'.
    - Sørg for, at S7-1500 PLC'en får tildelt en unik IP-adresse inden for det specificerede IP-adresseområde.

7. **Konfigurer netværksindstillinger for hver enhed:**
    - Klik på hver enhed i netværksoversigten for at åbne dens konfigurationsindstillinger.
    - Tildel et unikt enhedsnavn og den relevante IP-adresse til hver enhed, så de passer inden for det definerede IP-adresseområde.
8. **Online & Diagnostic:**
    - Gå online med netværket ved hjælp af TIA Portal.
    - Brug *Online & Diagnostic* til at finde MAC-adresserne for alle tilsluttede enheder.
    - Undersøg om alle enheder fungerer korrekt, og om der er nogen netværksfejl.
    - Opdater dokumentationen med de fundne MAC-adresser og eventuelle bemærkninger om netværksfejl eller status.
9. **Gem og kompiler projektet for at anvende ændringerne:**
    - Efter at have konfigureret alle enheder, skal du gemme projektet.
    - Klik på *Kompiler* for at sikre, at alle ændringer er korrekte og integreret i projektet.
10. **Dokumentation:**
    - Opdater den oprindelige dokumentation med eventuelle ændringer eller tilføjelser foretaget under opsætningsprocessen.
    - Sørg for, at alle detaljer er nøjagtige og let tilgængelige for fremtidig reference.

Note: Hvis det ikke er muligt at bruge en fysisk S7-1200 og en simuleret S7-1500, kan to simulerede PLC'er anvendes i stedet.

Ekstra Tips til PROFINET Netværksopsætning

- **Meningsfulde Enhedsnavne:** Brug meningsfulde navne til enhederne, der afspejler deres funktion eller placering, for lettere identifikation og fejlfinding.
- **Ensartet IP-adressering:** Sørg for, at alle IP-adresser er inden for samme subnet for at sikre korrekt kommunikation mellem enhederne.
- **Dokumentation:** Dokumentér alle netværksindstillinger, enhedsnavne og IP-adresser for fremtidig reference og fejlfinding.
- **Netværkstest:** Efter kompilering og implementering, test netværket for at sikre, at alle enheder kommunikerer korrekt og effektivt.

Ved at følge disse detaljerede trin og tips kan du oprette og konfigurere et robust PROFINET netværk i TIA Portal, der opfylder dine industrielle automatiseringskrav.

12.4 Fejlfinding af Netværk i TIA Portal

Mål: Formålet med opgaven er at du forstår og kan ændre IP-adresser på enheder i et PROFINET netværk som har fået konfigureret en forkert IP-adresse.

1. Arbejd videre fra forrige opgave.
2. Konfigurer en IP-adresse som ligger uden for dit subnet og derefter lav "Hardware download"til enheden.
3. Brug *Diagnostic* funktionerne i TIA Portal til at identificere eventuelle fejl eller advarsler.
4. Prøv at konfigurer en IP-adresse som er i det rigtige subnet og se om du kan komme i kontakt med enheden.
5. Anvend *Accessible Devices*-værktøjet til at scanne netværket og bekræfte, at alle enheder er korrekt forbundet og konfigureret.
6. Naviger til *Online & Diagnostic* for at give enheden en ny IP-adresse.
7. Dokumentér netværkskonfiguration og eventuelle ændringer grundigt for fremtidig reference og fejlfinding.

Lav enten en video eller dokumenter din måde at finde frem til fejlen og udbedre fejlen.

12.5 S7-Communication

Før du begynder at løse netværksopgaverne i TIA Portal, er det vigtigt at du har forståelse for de forskellige metoder til dataoverførsel og netværkskommunikation. Følgende afsnit vil give dig indsigt i de vigtigste kommunikationsblokke og deres anvendelsesområder. Dette vil danne grundlaget for de praktiske opgaver, du skal udføre.

12.6 Kommunikationsblokke og deres Anvendelse

BSEND

- **Datatyper:**

- **Store Datasæt:** Egnet til at sende større datasæt som komplette datasæt, konfigurationsfiler eller batchprocesdata.
- **Komplekse Strukturer:** Kan håndtere komplekse datastrukturer, herunder arrays, records eller multi-dimensionelle data.
- **Kritisk Kontroldata:** Bruges til at transmittere kritisk kontrolinformation, såsom setpunkter, kontrolparametre eller systemtilstande.

- **Brugstilfælde:**
  - **Batchbehandling:** Overførsel af hele batches af data i fremstillingsprocesser.
  - **Konfigurationsoverførsler:** Initiering eller opdatering af systemkonfigurationer.
  - **Kvalitetskontroldata:** Transmitterer detaljerede kvalitetskontroldata til analyse.
  - **Synkronisering af Tilstand:** Holder systemer synkroniseret ved at sende omfattende tilstandsoplysninger.

TSEND_C

- **Datatyper:**
  - **Almindelige Datasæt:** Brugt til almindelig dataoverførsel såsom statusopdateringer og kontrolsignaler.
  - **Strukturerede Data:** Kan håndtere simple strukturer og mindre datasæt.

- **Brugstilfælde:**
  - **Realtidskommunikation:** Bruges til realtidsdataudveksling mellem PLC og andre systemer, fx HMI'er.
  - **Integrering med IT-systemer:** Muliggør kommunikation med IT-systemer eller andre enheder via standard TCP/IP.

TSEND

- **Datatyper:**
  - **Sekventielle Data:** Velegnet til overførsel af data i sekventielle eller kontinuerlige strømme.
  - **Mindre Datasæt:** Ideel til mindre, hyppigt opdaterede datasæt.

- **Brugstilfælde:**
  - **Data Logging:** Sender data kontinuerligt til en datalogger.

- **Overvågning:** Overfører statusopdateringer til overvågningssystemer.

PUT

- **Datatyper:**
  - **Enkelte Variabler:** Overfører specifikke dataelementer, såsom enkelte variabler eller små datasæt.
  - **Kritiske Opdateringer:** Egnet til at sende kritiske opdateringer eller kontrolsignaler.
- **Brugstilfælde:**
  - **Direkte PLC-til-PLC Kommunikation:** Anvendes til simpel og direkte dataoverførsel mellem to PLC'er.
  - **Kontrolkommandoer:** Sender kontrolkommandoer til en anden PLC.

GET

- **Datatyper:**
  - **Enkelte Variabler:** Henter specifikke dataelementer fra en anden PLC.
  - **Mindre Datasæt:** Henter små datasæt til brug i kontrolprogrammer.
- **Brugstilfælde:**
  - **Data Retrieval:** Bruges til at hente aktuelle værdier fra en anden PLC til overvågning eller videre behandling.
  - **Tilstandsopdateringer:** Modtager opdateringer om systemtilstand fra andre enheder.

USEND

- **Datatyper:**
  - **Pakker:** Sender data i form af individuelle pakker.
  - **Best Effort Data:** Velegnet til data, hvor pålidelighed ikke er kritisk.
- **Brugstilfælde:**
  - **Broadcast Kommunikation:** Sender data til flere modtagere i et netværk.
  - **Ikke-kritiske Opdateringer:** Bruges til opdateringer, hvor tab af data kan tolereres, såsom periodiske statusmeddelelser.

PUT-metode

Mål: Denne øvelse har til formål at demonstrere konfiguration og anvendelse af PUT-metoden for dataoverførsel mellem en fysisk SIMATIC S7-1200 PLC og en simuleret SIMATIC S7-1500 PLC. Målet er at udvikle en dybere forståelse for effektiv datakommunikation i automatiserede systemer og at opnå praktisk erfaring med direkte PLC-til-PLC kommunikation. Du skal også designe en simpel produktionslinje i Emulate3D, hvor data udveksles mellem to PLC'er for at koordinere maskinernes drift.

Opgave:

1. **Planlægning og Dokumentation:**
    - Udarbejd en detaljeret plan for placering af PLC-tavleskabe og maskiner. Tavleskabene skal være placeret med en afstand på mindre end 100m, men mere end 50m.
    - **Note:** To rullebånd kan bruges til at symbolisere maskiner.
    - Dokumentér planen med et layoutdiagram, der viser placeringen af tavleskabe og maskiner.
2. **Scenarieopbygning i Emulate3D:**
    - Design en simpel produktionslinje i Emulate3D med to maskiner, der skal koordinere deres aktiviteter.
    - Den første maskine (PLC A) skal starte en operation og sende en besked til den anden maskine (PLC B), når operationen er fuldført.
    - Den anden maskine (PLC B) skal modtage beskeden og begynde sin operation baseret på den modtagne data.
3. **Konfiguration i TIA Portal:**
    - Konfigurer en fysisk S7-1200 PLC og en simuleret S7-1500 PLC i TIA Portal, hvor S7-1200 fungerer som sender (bruger PUT-metoden) og S7-1500 som modtager.
    - Tildel unikke IP-adresser til begge PLC'er og opsæt passende netværksparametre for at muliggøre kommunikation via Ethernet.
4. **Simulering med PLCSIM Advanced:**
    - Brug PLCSIM Advanced til at simulere S7-1500 PLC'en og etablere en virtuel forbindelse mellem den og den fysiske S7-1200 PLC.
    - **Note:** Vælg *TCP/IP Single Adapter* og derefter vælg det fysiske netkort **(ikke wireless)**.

5. **Konfiguration af PUT-funktionsblok:**
    - I sender-PLC'en (S7-1200), konfigurer en PUT-funktionsblok med relevante data og destinationsadressen for modtager-PLC'en (S7-1500).
    - Initialiser PUT-funktionsblokken ved at aktivere 'REQ'-variablen, når både 'DONE' og 'ERROR' er inaktive.
    - I modtager-PLC'en (S7-1500), opsæt en modtagefunktion for at håndtere og lagre de modtagne data så de evt. ville kunne vises på en HMI
6. **Test af dataoverførsel:**
    - Test dataoverførslen ved at sende forskellige datatyper fra sender til modtager og verificer korrekt modtagelse og integritet (nøjagtighed).
7. **Fejlhåndtering:**
    - Implementer fejlhåndteringsmekanismer for at sikre pålidelighed og korrekt respons på eventuelle kommunikationsfejl.
    - Fejl gemmes i et array for yderligere analyse.
8. **Dokumentation:**
    - Fysiske layouttegninger: Viser den fysiske placering af netværksudstyr, kabler og andre komponenter i automatiseringsmiljøet.
    - Logiske diagrammer: Illustrerer dataflowet og kommunikationen mellem forskellige enheder og systemer i netværket.
    - Netværksdiagrammer: Viser den fysiske og logiske struktur af netværket, herunder enheder, forbindelser og protokoller.
    - Device name & IP-adresseplan: En tabel eller liste over alle enheder på netværket med deres tildelte device name IP-adresser, subnetmasker og gateways.
    - Revisionshistorik: En oversigt over ændringer og opdateringer til netværkskonfigurationen over tid.
    - Oversigtsdiagram: Et oversigtsdiagram over produktionslinjen i Emulate3D og plantegning.

Krav:

- Forståelse af PLC-til-PLC kommunikation og grundlæggende principper i netværksopsætning.
- Erfaring med programmering og konfiguration i TIA Portal og brug af PLCSIM Advanced.

- Evne til at konstruere og fejlsøge avancerede PLC-programmer, der involverer direkte datakommunikation.

Note: Hvis det ikke er muligt at bruge en fysisk S7-1200 PLC og en simuleret S7-1500 PLC, kan to simulerede PLC'er anvendes i stedet.

GET-metode

Mål: Formålet med denne øvelse er at demonstrere konfiguration og anvendelse af GET-metoden for datahentning mellem to SIMATIC S7-1500 PLC'er. Målet er at forbedre forståelsen af direkte PLC-til-PLC kommunikation og at opnå hands-on erfaring med effektiv dataudveksling i automatiseringssystemer. Du skal også designe en simpel produktionslinje i Emulate3D, hvor data udveksles mellem to PLC'er for at koordinere maskinernes drift.

Opgave:

1. **Planlægning og Dokumentation:**
    - Udarbejd en detaljeret plan for placering af PLC-tavleskabe og maskiner. Tavleskabene skal være placeret med en afstand på mere end 101m fra hinanden.
    - **Note:** To rullebånd kan bruges til at symbolisere maskiner.
    - Dokumentér planen med et layoutdiagram, der viser placeringen af tavleskabe og maskiner.
2. **Scenarieopbygning i Emulate3D:**
    - Design en simpel produktionslinje i Emulate3D med to maskiner, der skal koordinere deres aktiviteter.
    - Den første maskine (PLC A) skal starte en operation og sende en besked til den anden maskine (PLC B), når operationen er fuldført.
    - Den anden maskine (PLC B) skal modtage beskeden og begynde sin operation baseret på den modtagne data.
3. **Konfiguration i TIA Portal:**
    - Konfigurer to S7-1500 PLC'er i TIA Portal, hvor den ene fungerer som datakilde og den anden som datahentende enhed (bruger GET-metoden).
    - Tildel unikke IP-adresser til begge PLC'er og opsæt netværksparametre for Ethernet-kommunikation.
4. **Simulering med PLCSIM Advanced:**

- Brug PLCSIM Advanced til at simulere begge PLC'er og etablere en virtuel Ethernet-forbindelse mellem dem.
- **Note:** Vælg *TCP/IP Single Adapter* og derefter vælg det fysiske netkort **(ikke wireless)**.

5. **Konfiguration af GET-funktionsblok:**
   - I kildens PLC, opsæt et datalager, der indeholder de data, som skal hentes af den anden PLC.
   - I den datahentende PLC, konfigurer en GET-funktionsblok med kildens IP-adresse og specifikationer for de data, der skal hentes.
   - Initialiser GET-funktionsblokken ved at aktivere 'REQ'-variablen, når både 'DONE' og 'ERROR' er inaktive.
   - I modtager-PLC'en, opsæt en modtagefunktion for at håndtere og lagre de modtagne data, så de evt. kan vises på en HMI.
6. **Test af datahentning:**
   - Test datahentningsprocessen ved at anmode om og modtage data fra kilden og verificere dataenes integritet (nøjagtighed) og korrekthed.
7. **Fejlhåndtering:**
   - Implementer fejlhåndteringslogik for at adressere eventuelle kommunikationsfejl og sikre pålidelighed i dataoverførslen.
   - Fejl gemmes i et array for yderligere analyse.
8. **Dokumentation:**
   - Fysiske layouttegninger: Viser den fysiske placering af netværksudstyr, kabler og andre komponenter i automatiseringsmiljøet.
   - Logiske diagrammer: Illustrerer dataflowet og kommunikationen mellem forskellige enheder og systemer i netværket.
   - Netværksdiagrammer: Viser den fysiske og logiske struktur af netværket, herunder enheder, forbindelser og protokoller.
   - Device name & IP-adresseplan: En tabel eller liste over alle enheder på netværket med deres tildelte device name, IP-adresser, subnetmasker og gateways.
   - Revisionshistorik: En oversigt over ændringer og opdateringer til netværkskonfigurationen over tid.
   - Oversigtsdiagram: Et oversigtsdiagram over produktionslinjen i Emulate3D og plantegning.

Krav:

- Grundlæggende forståelse for PLC-kommunikation og principper for netværkskonfiguration.
- Kompetencer indenfor TIA Portal-programmering og anvendelse af PLCSIM Advanced.
- Evner til at udvikle og debugge komplekse PLC-programmer, som involverer direkte dataudveksling mellem PLC'er.

Note: Hvis det ikke er muligt at bruge en fysisk S7-1200 PLC og en simuleret S7-1500 PLC, kan to simulerede PLC'er anvendes i stedet.

12.7 S7-Communication Others

USEND & URCV

Mål: Formålet med denne øvelse er at demonstrere konfiguration og anvendelse af USEND og URCV metoderne for dataoverførsel mellem en fysisk SIMATIC S7-1200 PLC og en simuleret SIMATIC S7-1500 PLC. Målet er at udvikle en dybere forståelse for effektiv datakommunikation i automatiserede systemer og at opnå praktisk erfaring med direkte PLC-til-PLC kommunikation. Du skal også designe en simpel produktionslinje i Emulate3D, hvor data udveksles mellem tre rullebånd for at koordinere deres drift.

Opgave:

1. **Planlægning og Dokumentation:**
   - Udarbejd en detaljeret plan for placering af PLC-tavleskabe og rullebånd. Tavleskabene skal være placeret med en afstand, der kræver kommunikation via et netværk.
   - **Note:** Tre rullebånd kan bruges til at symbolisere maskiner.
   - Dokumentér planen med et layoutdiagram, der viser placeringen af tavleskabe og rullebånd.
2. **Scenarieopbygning i Emulate3D:**
   - Design en simpel produktionslinje i Emulate3D med tre rullebånd, der skal koordinere deres aktiviteter.
   - Det første rullebånd (PLC A) skal starte en operation og sende en besked til det andet rullebånd (PLC B) og det tredje rullebånd (PLC C), når operationen er fuldført.
   - De to andre rullebånd (PLC B og PLC C) skal modtage beskeden og begynde deres operationer baseret på den modtagne data.

3. **Konfiguration i TIA Portal:**

    - Konfigurer en fysisk S7-1200 PLC og 2 simuleret S7-1500 PLC i TIA Portal, hvor S7-1200 fungerer som sender (bruger USEND-metoden) og S7-1500 PLC'erne som modtager (bruger URCV-metoden).
    - Tildel unikke IP-adresser til begge PLC'er og opsæt passende netværksparametre for at muliggøre kommunikation via Ethernet.

4. **Simulering med PLCSIM Advanced:**

    - Brug PLCSIM Advanced til at simulere S7-1500 PLC'erne og etablere en virtuel forbindelse mellem den og den fysiske S7-1200 PLC.
    - **Note:** Vælg *TCP/IP Single Adapter* og derefter vælg det fysiske netkort **(ikke wireless)**.

5. **Konfiguration af USEND- og URCV-funktionsblokke:**

    - I sender-PLC'en (S7-1200), konfigurer en USEND-funktionsblok med relevante data og destinationsadressen for modtager-PLC'en (S7-1500).
    - Initialiser USEND-funktionsblokken ved at aktivere 'REQ'-variablen, når både 'DONE' og 'ERROR' er inaktive.
    - I modtager-PLC'en (S7-1500), opsæt en URCV-funktionsblok for at håndtere og lagre de modtagne data, så de evt. ville kunne vises på en HMI.

6. **Test af dataoverførsel:**

    - Test dataoverførslen ved at sende forskellige datatyper fra sender til modtager og verificer korrekt modtagelse og integritet (nøjagtighed).

7. **Fejlhåndtering:**

    - Implementer fejlhåndteringsmekanismer for at sikre pålidelighed og korrekt respons på eventuelle kommunikationsfejl.
    - Fejl gemmes i et array for yderligere analyse.

8. **Dokumentation:**

    - Fysiske layouttegninger: Viser den fysiske placering af netværksudstyr, kabler og andre komponenter i automatiseringsmiljøet.
    - Logiske diagrammer: Illustrerer dataflowet og kommunikationen mellem forskellige enheder og systemer i netværket.

- Netværksdiagrammer: Viser den fysiske og logiske struktur af netværket, herunder enheder, forbindelser og protokoller.
- Device name & IP-adresseplan: En tabel eller liste over alle enheder på netværket med deres tildelte device name, IP-adresser, subnetmasker og gateways.
- Revisionshistorik: En oversigt over ændringer og opdateringer til netværkskonfigurationen over tid.
- Oversigtsdiagram: Et oversigtsdiagram over produktionslinjen i Emulate3D og plantegning.
- Tegning af Netværkstopologi

Krav:

- Grundlæggende forståelse for PLC-kommunikation og principper for netværkskonfiguration.
- Kompetencer indenfor TIA Portal-programmering og anvendelse af PLCSIM Advanced.
- Evner til at udvikle og debugge komplekse PLC-programmer, som involverer direkte dataudveksling mellem PLC'er.

Note: Hvis det ikke er muligt at bruge en fysisk S7-1200 PLC og en simuleret S7-1500 PLC, kan to simulerede PLC'er anvendes i stedet.

BSEND & BRCV

Mål: Formålet med denne øvelse er at demonstrere konfiguration og anvendelse af BSEND og BRCV metoderne for dataoverførsel mellem en fysisk SIMATIC S7-1200 PLC og en simuleret SIMATIC S7-1500 PLC. Målet er at udvikle en dybere forståelse for effektiv datakommunikation i automatiserede systemer og at opnå praktisk erfaring med direkte PLC-til-PLC kommunikation. Du skal også designe en simpel produktionslinje i Emulate3D, hvor data udveksles mellem tre rullebånd for at koordinere deres drift.

Opgave:

1. **Planlægning og Dokumentation:**
   - Udarbejd en detaljeret plan for placering af PLC-tavleskabe og rullebånd. Tavleskabene skal være placeret med en afstand, der kræver kommunikation via et netværk.
   - **Note:** Tre rullebånd kan bruges til at symbolisere maskiner.
   - Dokumentér planen med et layoutdiagram, der viser placeringen af tavleskabe og rullebånd.

2. **Scenarieopbygning i Emulate3D:**

   - Design en simpel produktionslinje i Emulate3D med tre rullebånd, der skal koordinere deres aktiviteter.
   - Det første rullebånd (PLC A) skal starte en operation og sende en besked til det andet rullebånd (PLC B) og det tredje rullebånd (PLC C), når operationen er fuldført.
   - De to andre rullebånd (PLC B og PLC C) skal modtage beskeden og begynde deres operationer baseret på den modtagne data.

3. **Konfiguration i TIA Portal:**

   - Konfigurer en fysisk S7-1200 PLC og en simuleret S7-1500 PLC i TIA Portal, hvor S7-1200 fungerer som sender (bruger BSEND-metoden) og S7-1500 som modtager (bruger BRCV-metoden).
   - Tildel unikke IP-adresser til begge PLC'er og opsæt passende netværksparametre for at muliggøre kommunikation via Ethernet.

4. **Simulering med PLCSIM Advanced:**

   - Brug PLCSIM Advanced til at simulere S7-1500 PLC'en og etablere en virtuel forbindelse mellem den og den fysiske S7-1200 PLC.
   - **Note:** Vælg *TCP/IP Single Adapter* og derefter vælg det fysiske netkort **(ikke wireless)**.

5. **Konfiguration af BSEND- og BRCV-funktionsblokke:**

   - I sender-PLC'en (S7-1200), konfigurer en BSEND-funktionsblok med relevante data og destinationsadressen for modtager-PLC'en (S7-1500).
   - Initialiser BSEND-funktionsblokken ved at aktivere 'REQ'-variablen, når både 'DONE' og 'ERROR' er inaktive.
   - I modtager-PLC'en (S7-1500), opsæt en BRCV-funktionsblok for at håndtere og lagre de modtagne data, så de evt. ville kunne vises på en HMI.

6. **Test af dataoverførsel:**

   - Test dataoverførslen ved at sende forskellige datatyper fra sender til modtager og verificer korrekt modtagelse og integritet (nøjagtighed).

7. **Fejlhåndtering:**

   - Implementer fejlhåndteringsmekanismer for at sikre pålidelighed og korrekt respons på eventuelle kommunikationsfejl.

- Fejl gemmes i et array for yderligere analyse.

8. **Dokumentation:**

   - Fysiske layouttegninger: Viser den fysiske placering af netværksudstyr, kabler og andre komponenter i automatiseringsmiljøet.
   - Logiske diagrammer: Illustrerer dataflowet og kommunikationen mellem forskellige enheder og systemer i netværket.
   - Netværksdiagrammer: Viser den fysiske og logiske struktur af netværket, herunder enheder, forbindelser og protokoller.
   - Device name & IP-adresseplan: En tabel eller liste over alle enheder på netværket med deres tildelte device name, IP-adresser, subnetmasker og gateways.
   - Revisionshistorik: En oversigt over ændringer og opdateringer til netværkskonfigurationen over tid.
   - Oversigtsdiagram: Et oversigtsdiagram over produktionslinjen i Emulate3D og plantegning.

Krav:

- Grundlæggende forståelse for PLC-kommunikation og principper for netværkskonfiguration.
- Kompetencer indenfor TIA Portal-programmering og anvendelse af PLCSIM Advanced.
- Evner til at udvikle og debugge komplekse PLC-programmer, som involverer direkte dataudveksling mellem PLC'er.

Note: Hvis det ikke er muligt at bruge en fysisk S7-1200 PLC og en simuleret S7-1500 PLC, kan to simulerede PLC'er anvendes i stedet.

12.8 Open User Communication

TCON & TDISCON

Mål: Formålet med denne opgave er at konfigurere og teste kommunikationen mellem to SIMATIC S7-1500 PLC'er ved brug af TCON (Transport Control Protocol) over en simuleret industrielt Ethernet-netværk. Du vil udvikle forståelse for etablering af forbindelser, dataudveksling og grundlæggende netværksdiagnostik i et virtuelt miljø ved brug af PLCSIM Advanced.

Opgave:

1. **Planlægning og Dokumentation:**

- Udarbejd en detaljeret plan for placering af PLC-tavleskabe og maskiner. Tavleskabene skal være placeret med en afstand, der kræver kommunikation via et netværk.
- **Note:** To rullebånd kan bruges til at symbolisere maskiner.
- Dokumentér planen med et layoutdiagram, der viser placeringen af tavleskabe og maskiner.

2. **Scenarieopbygning i Emulate3D:**

   - Design en simpel produktionslinje i Emulate3D med to maskiner, der skal koordinere deres aktiviteter.
   - Den første maskine (PLC A) skal starte en operation og sende en besked til den anden maskine (PLC B), når operationen er fuldført.
   - Den anden maskine (PLC B) skal modtage beskeden og begynde sin operation baseret på den modtagne data.

3. **Konfiguration i TIA Portal:**

   - Åbn TIA Portal og opret et nyt projekt med en fysisk S7-1200 og en simuleret S7-1500 PLC. Tildel dem unikke IP-adresser inden for samme subnet.
   - Konfigurer de nødvendige netværksindstillinger i begge PLC'er, så de kan etablere en TCP-forbindelse. Sørg for at aktivere TCON-blokken i brugerprogrammet.

4. **Simulering med PLCSIM Advanced:**

   - Initialiser PLCSIM Advanced og opret to separate instanser for hver af dine PLC'er.
   - Etabler en forbindelse mellem de to simulerede PLC'er ved at bruge TSEND og TRECV datablokkene for at sende og modtage data.

5. **Test af dataudveksling:**

   - Implementer en simpel dataudveksling, hvor PLC 1 sender en streng eller et heltal til PLC 2, og PLC 2 sender en bekræftelse tilbage.
   - Overvåg og verificer kommunikationen mellem de to PLC'er ved hjælp af TIA Portal's diagnostiske værktøjer. Dokumenter alle trin og resultater.

6. **Fejlhåndtering:**

- Identificer og fejlfind enhver kommunikationsfejl ved hjælp af PLCSIM Advanced diagnostiske funktioner.
- Implementer fejlhåndteringsmekanismer for at sikre pålidelighed og korrekt respons på eventuelle kommunikationsfejl.

7. **Analyse:**

   - Diskuter mulige anvendelsesområder for PLC-til-PLC-kommunikation i industrielle automatiseringsmiljøer og hvordan simulation kan bistå i design og fejlfinding af disse systemer.

8. **Dokumentation:**

   - Fysiske layouttegninger: Viser den fysiske placering af netværksudstyr, kabler og andre komponenter i automatiseringsmiljøet.
   - Logiske diagrammer: Illustrerer dataflowet og kommunikationen mellem forskellige enheder og systemer i netværket.
   - Netværksdiagrammer: Viser den fysiske og logiske struktur af netværket, herunder enheder, forbindelser og protokoller.
   - Device name & IP-adresseplan: En tabel eller liste over alle enheder på netværket med deres tildelte device name, IP-adresser, subnetmasker og gateways.
   - Revisionshistorik: En oversigt over ændringer og opdateringer til netværkskonfigurationen over tid.
   - Oversigtsdiagram: Et oversigtsdiagram over produktionslinjen i Emulate3D og plantegning.

Krav:

- Fuldstændig forståelse af TIA Portal interface og navigering.
- Grundlæggende viden om industrielle Ethernet-netværk og TCP/IP-protokollen.
- Evne til at implementere og fejlfinde TCON kommunikationsblokke i et SIMATIC S7-1500 miljø.

Aflevering (optional): En rapport, der inkluderer konfigurationsdetaljer, skærmbilleder, der viser den vellykkede dataudveksling, og en fejlanalyse med løsninger på eventuelle problemer, der opstod under opgaven.

Note: Hvis det ikke er muligt at bruge en fysisk S7-1200 PLC og en simuleret S7-1500 PLC, kan to simulerede PLC'er anvendes i stedet.

TSEND_C & TRCV_C

Mål: I denne øvelse vil de studerende oprette en sikker og konsistent dataudveksling mellem to SIMATIC S7-1500 PLC'er ved brug af de konsistente datakommunikationsblokke 'TSEND_C' og 'TRCV_C'. Formålet er at forstå og anvende metoder for konsistent datatransmission i et simuleret industrielt Ethernet-netværk og at få praktisk erfaring med avancerede datakommunikationsmekanismer.

Opgave:

1. **Planlægning og Dokumentation:**
   - Udarbejd en detaljeret plan for placering af PLC-tavleskabe og maskiner. Tavleskabene skal være placeret med en afstand, der kræver kommunikation via et netværk.
   - **Note:** To rullebånd kan bruges til at symbolisere maskiner.
   - Dokumentér planen med et layoutdiagram, der viser placeringen af tavleskabe og maskiner.
2. **Scenarieopbygning i Emulate3D:**
   - Design en simpel produktionslinje i Emulate3D med tre maskiner, der skal koordinere deres aktiviteter.
   - Den første maskine (PLC A) skal starte en operation og sende en besked til den anden maskine (PLC B), når operationen er fuldført.
   - Den anden maskine (PLC B) skal modtage beskeden og begynde sin operation baseret på den modtagne data og sende en besked til den tredje maskine (PLC C), når operationen er fuldført.
   - Den tredje maskine (PLC C) skal modtage beskeden og begynde sin operation baseret på den modtagne data.
3. **Konfiguration i TIA Portal:**
   - Start med at konfigurere to S7-1500 PLC'er i TIA Portal, inklusive tildeling af passende IP-adresser og enhedsnavne.
   - Opret en konsistent datatilkobling ved at anvende 'TSEND_C' og 'TRCV_C' blokkene i henholdsvis sender- og modtager-PLC'ens program.
   - Konfigurer PLC'erne til at køre i et simuleret miljø ved hjælp af PLCSIM Advanced, og sikre, at begge PLC'er er korrekt forbundet til den simulerede netværksinfrastruktur.
4. **Simulering med PLCSIM Advanced:**

- Brug PLCSIM Advanced til at simulere begge PLC'er og etablere en virtuel forbindelse mellem dem.

5. **Konfiguration af TSEND_C og TRCV_C-funktionsblokke:**
   - Definer et datasæt, der skal overføres fra sender-PLC'en til modtager-PLC'en, og konfigurer 'TSEND_C'-blokken til at sende denne data cyklisk.
   - Indstil 'TRCV_C'-blokken i modtager-PLC'en til at modtage den sendte data og implementer en kvitteringsmekanisme for at sikre, at transmissionen er sket.
6. **Test af dataudveksling:**
   - Demonstrer og valider konsistensen af dataoverførslen ved at observere og registrere dataudvekslingen i TIA Portal's overvågningsværktøjer.
7. **Fejlhåndtering:**
   - Analyser systemets adfærd og reaktioner ved eventuelle netværksforstyrrelser eller kommunikationsfejl, og dokumenter procedurerne for fejlfinding.
   - Implementer fejlhåndteringsmekanismer for at sikre pålidelighed og korrekt respons på eventuelle kommunikationsfejl.
   - Fejl gemmes i et array for yderligere analyse.
8. **Analyse:**
   - Diskutér, hvordan konsistent datakommunikation kan være kritisk i visse industrielle applikationer, og hvordan man kan sikre høj systempålidelighed gennem simulation og test.
9. **Dokumentation:**
   - Fysiske layouttegninger: Viser den fysiske placering af netværksudstyr, kabler og andre komponenter i automatiseringsmiljøet.
   - Logiske diagrammer: Illustrerer dataflowet og kommunikationen mellem forskellige enheder og systemer i netværket.
   - Netværksdiagrammer: Viser den fysiske og logiske struktur af netværket, herunder enheder, forbindelser og protokoller.
   - Device name & IP-adresseplan: En tabel eller liste over alle enheder på netværket med deres tildelte device name, IP-adresser, subnetmasker og gateways.
   - Revisionshistorik: En oversigt over ændringer og opdateringer til netværkskonfigurationen over tid.

- Oversigtsdiagram: Et oversigtsdiagram over produktionslinjen i Emulate3D og plantegning.

Krav:

- Forståelse for konfiguration af kommunikationsblokke og netværksparametre i TIA Portal.
- Bekendtskab med funktionaliteten af konsistent dataoverførsel ved hjælp af 'TSEND_C' og 'TRCV_C'.
- Evne til at udføre simulering og fejlfinding af netværkskommunikation med PLCSIM Advanced.

Aflevering (optional): En teknisk rapport, der indeholder de anvendte konfigurationer, skærmbilleder af datatransmissionen, og en diskussion af resultaterne samt eventuelle udfordringer og løsninger.

Denne opgave vil understøtte de studerendes færdigheder i håndtering af konsistent og sikker PLC-til-PLC kommunikation og vil forbedre deres evner til at anvende komplekse kommunikationssystemer i industrielle automatiseringsprojekter.

TMAIL_C

Mål: I denne øvelse skal de studerende konfigurere og anvende TMAIL_C-blokken til at sende e-mails fra en SIMATIC S7-1500 PLC. Øvelsen kombinerer brugen af PUT/GET til dataudveksling mellem to linjer og sender en e-mail ved fejl. Formålet er at forstå opsætning af e-mail-kommunikation, anvende denne teknologi i industrielle applikationer og få praktisk erfaring med avancerede kommunikationsmetoder i TIA Portal.

Opgave:

1. **Planlægning og Dokumentation:**
   - Udarbejd en detaljeret plan for anvendelse af e-mailkommunikation i et industrielt miljø.
   - **Note:** To rullebånd kan bruges til at symbolisere maskiner.
   - Dokumentér planen med et layoutdiagram, der viser anvendelsen af e-mailkommunikation i processen.
2. **Scenarieopbygning i Emulate3D:**
   - Design en simpel produktionslinje i Emulate3D med tre maskiner, hvor e-mails bruges til at sende beskeder om maskinens status og fejlmeddelelser.

- Den første maskine (PLC A) skal sende en e-mail til operatøren, når en operation er fuldført.
- Den anden maskine (PLC B) skal sende en e-mail, hvis der opstår en fejl.
- Den tredje maskine (PLC C) skal sende en e-mail med en daglig statusrapport.

3. **Konfiguration i TIA Portal:**
   - Konfigurer en S7-1500 PLC i TIA Portal med de nødvendige netværksparametre og e-mailindstillinger.
   - Indstil SMTP-serveroplysningerne og e-mailkontoen, der skal bruges til at sende meddelelser.
4. **Simulering med PLCSIM Advanced:**
   - Brug PLCSIM Advanced til at simulere PLC'en og sikre, at e-mailkommunikationen fungerer korrekt i det simulerede miljø.
5. **Konfiguration af PUT/GET-funktionsblokke:**
   - Konfigurer en fysisk S7-1200 PLC og en simuleret S7-1500 PLC i TIA Portal, hvor S7-1200 fungerer som sender (bruger PUT-metoden) og S7-1500 som modtager.
   - I sender-PLC'en (S7-1200), konfigurer en PUT-funktionsblok med relevante data og destinationsadressen for modtager-PLC'en (S7-1500).
   - Initialiser PUT-funktionsblokken ved at aktivere 'REQ'-variablen, når både 'DONE' og 'ERROR' er inaktive.
   - I modtager-PLC'en (S7-1500), opsæt en modtagefunktion for at håndtere og lagre de modtagne data, så de evt. ville kunne vises på en HMI.
6. **Test af dataoverførsel:**
   - Test dataoverførslen ved at sende forskellige datatyper fra sender til modtager og verificer korrekt modtagelse og integritet (nøjagtighed).
7. **Fejlhåndtering:**
   - Implementer fejlhåndteringsmekanismer for at sikre pålidelighed og korrekt respons på eventuelle kommunikationsfejl.
   - Fejl gemmes i et array for yderligere analyse.
   - Konfigurer TMAIL_C-blokken til at sende en e-mail ved fejl.

8. **Analyse:**

    - Diskutér, hvordan e-mailkommunikation kan anvendes i industrielle applikationer, og hvordan det kan forbedre systemets pålidelighed og effektivitet.

9. **Dokumentation:**

    - Fysiske layouttegninger: Viser den fysiske placering af netværksudstyr, kabler og andre komponenter i automatiseringsmiljøet.
    - Logiske diagrammer: Illustrerer dataflowet og kommunikationen mellem forskellige enheder og systemer i netværket.
    - Netværksdiagrammer: Viser den fysiske og logiske struktur af netværket, herunder enheder, forbindelser og protokoller.
    - Device name & IP-adresseplan: En tabel eller liste over alle enheder på netværket med deres tildelte device name, IP-adresser, subnetmasker og gateways.
    - Revisionshistorik: En oversigt over ændringer og opdateringer til netværkskonfigurationen over tid.
    - Oversigtsdiagram: Et oversigtsdiagram over produktionslinjen i Emulate3D og plantegning.

Krav:

- Forståelse for konfiguration af kommunikationsblokke og netværksparametre i TIA Portal.
- Bekendtskab med funktionaliteten af PUT/GET og TMAIL_C-blokken.
- Evne til at udføre simulering og fejlfinding af netværkskommunikation med PLCSIM Advanced.

Aflevering (optional): En teknisk rapport, der indeholder de anvendte konfigurationer, skærmbilleder af e-mailtransmissionen, og en diskussion af resultaterne samt eventuelle udfordringer og løsninger.

Denne opgave vil understøtte de studerendes færdigheder i håndtering af e-mailkommunikation fra PLC'er og vil forbedre deres evner til at anvende avancerede kommunikationssystemer i industrielle automatiseringsprojekter.

12.9 Modbus TCP (Client/Server)

Mål: Denne øvelse fokuserer på at oprette og konfigurere en Modbus TCP client/server-kommunikation mellem to SIMATIC S7-1500 PLC'er, hvor PLC A er forbundet til Emulate3D og læser knapper og analoge værdier og sender

dem til PLC B for videre behandling. Målet er at udvikle en dybere forståelse for Modbus TCP-protokollen og praktisk erfaring med implementering af standardiserede industrielle kommunikationsprotokoller i et simuleret netværk.

Opgave:

1. **Planlægning og Dokumentation:**

    - Udarbejd en detaljeret plan for placering af PLC-tavleskabe og maskiner. Tavleskabene skal være placeret med en afstand, der kræver kommunikation via et netværk.
    - **Note:** To rullebånd kan bruges til at symbolisere maskiner.
    - Dokumentér planen med et layoutdiagram, der viser placeringen af tavleskabe og maskiner.

2. **Scenarieopbygning i Emulate3D:**

    - Design en simpel produktionslinje i Emulate3D med to maskiner, hvor data udveksles mellem to PLC'er for at koordinere maskinernes drift.
    - PLC A skal læse status for knapper (coils), digitale sensorer (diskret input) og analoge værdier (holding registre) og sende disse data til PLC B via Modbus TCP.
    - PLC B skal modtage dataene og udføre yderligere handlinger baseret på de modtagne værdier (f.eks. starte/stoppe maskiner, justere parametre).

3. **Konfiguration i TIA Portal:**

    - Konfigurer S7-1500 PLC i TIA Portal, en som Modbus TCP server (PLC A) og den anden S7-1200 PLC som Modbus TCP client (PLC B).
    - Tildel unikke IP-adresser og opret passende netværksparametre for begge PLC'er for at muliggøre kommunikation over TCP/IP.

4. **Simulering med PLCSIM Advanced:**

    - Brug S7-1200 og PLCSIM Advanced til at netværksforbindelse mellem dem. Hvis dette ikke virker så anvend 2 simulerede PLC'er.

5. **Konfiguration af Modbus TCP-funktionsblokke:**

- På Modbus TCP serveren (PLC A), definer dataområder som coils (til knapper), diskret input (til digitale sensorer) og holding registre (til analoge værdier), der skal tilgængeliggøres for klienten (PLC B).
- På Modbus TCP klienten (PLC B), konfigurer Modbus kommunikationsblokke til at forespørge data fra serveren og håndtere læse/skrive operationer for coils, diskret input og holding registre.

6. **Test af kommunikationsforbindelse:**
   - Test kommunikationsforbindelsen ved at forespørge og ændre værdier i serverens dataområder (coils, diskret input og holding registre) fra klienten og overvåge de resulterende ændringer.
7. **Fejlhåndtering:**
   - Implementer fejlhåndtering i begge PLC-programmer for at sikre robusthed og systempålidelighed.
8. **Analyse:**
   - Evaluer kommunikationslatens og throughput for Modbus TCP forbindelsen og diskuter, hvordan disse kan optimeres.
9. **Dokumentation:**
   - Fysiske layouttegninger: Viser den fysiske placering af netværksudstyr, kabler og andre komponenter i automatiseringsmiljøet.
   - Logiske diagrammer: Illustrerer dataflowet og kommunikationen mellem forskellige enheder og systemer i netværket.
   - Netværksdiagrammer: Viser den fysiske og logiske struktur af netværket, herunder enheder, forbindelser og protokoller.
   - Device name & IP-adresseplan: En tabel eller liste over alle enheder på netværket med deres tildelte device name, IP-adresser, subnetmasker og gateways.
   - Revisionshistorik: En oversigt over ændringer og opdateringer til netværkskonfigurationen over tid.
   - Oversigtsdiagram: Et oversigtsdiagram over produktionslinjen i Emulate3D og plantegning.

Krav:

- Grundlæggende forståelse af Modbus TCP-protokollen og dets anvendelse i industriel kommunikation.

- Erfaring med netværksopsætning i TIA Portal og anvendelse af PLCSIM Advanced.
- Kompetence i at konstruere og debugge PLC-programmer, der involverer komplekse kommunikationsprotokoller.

Aflevering: En detaljeret teknisk rapport med beskrivelse af implementeringsprocessen, udfordringer, løsninger og en analyse af systemets ydeevne.

12.10 CIP mellem Siemens og Rockwell

12.11 MQTT

Mål: Målet med denne opgave er at konfigurere og demonstrere MQTT-kommunikation mellem en fysisk SIMATIC S7-1200 PLC og en simuleret SIMATIC S7-1500 PLC. Denne opgave vil give de studerende en praktisk forståelse af MQTT-protokollen, som er bredt anvendt i industriel IoT.

Opgavebeskrivelse:

1. **Planlægning og Dokumentation:**
   - Udarbejd en detaljeret plan for placering af PLC-tavleskabe og maskiner. Tavleskabene skal være placeret med en afstand, der kræver kommunikation via et netværk.
   - **Note:** To rullebånd kan bruges til at symbolisere maskiner.
   - Dokumentér planen med et layoutdiagram, der viser placeringen af tavleskabe og maskiner.
2. **Scenarieopbygning i Emulate3D:**
   - Design en simpel produktionslinje i Emulate3D, hvor data udveksles mellem to PLC'er for at koordinere maskinernes drift.
   - Den fysiske S7-1200 PLC skal læse status for knapper og sensorer og sende disse data via MQTT til den simulerede S7-1500 PLC.
   - Den simulerede S7-1500 PLC skal modtage dataene og udføre yderligere handlinger baseret på de modtagne værdier (f.eks. starte/stoppe maskiner, justere parametre).
3. **Opsætning af MQTT:**
   - Opsæt en MQTT-broker, som PLC'erne kan forbinde til. Brug gerne en åben kildekode broker som Mosquitto for simpel opsætning.

Note: Mosquitto broker skal først installeres på egen computer og manualen skal følges for videre konfiguration.

4. **Konfiguration i TIA Portal:**
   - Konfigurer en fysisk S7-1200 PLC og en simuleret S7-1500 PLC i TIA Portal, hvor S7-1200 fungerer som MQTT Publisher og S7-1500 som MQTT Subscriber.
   - Tildel unikke IP-adresser til begge PLC'er og opsæt passende netværksparametre for at muliggøre kommunikation via Ethernet.
5. **Simulering med PLCSIM Advanced:**
   - Brug PLCSIM Advanced til at simulere S7-1500 PLC'en og etablere en virtuel forbindelse mellem den og den fysiske S7-1200 PLC.
6. **Implementering af MQTT-kommunikation:**
   - Konfigurer den fysiske S7-1200 PLC som MQTT Publisher, der sender beskeder til en bestemt topic.
   - Konfigurer den simulerede S7-1500 PLC som MQTT Subscriber, der abonnerer på denne topic.
   - Implementer logikken på begge PLC'er for håndtering af MQTT-kommunikationen. Dette inkluderer at oprette forbindelse til brokeren, abonnere/publish til topics og håndtere indkommende beskeder.
7. **Test af dataoverførsel:**
   - Test dataoverførslen ved at sende forskellige typer af beskeder, herunder tekststrenge, numeriske værdier, og binære kommandoer fra Publisher til Subscriber og verificer korrekt modtagelse.
8. **Dokumentation:**
   - Fysiske layouttegninger: Viser den fysiske placering af netværksudstyr, kabler og andre komponenter i automatiseringsmiljøet.
   - Logiske diagrammer: Illustrerer dataflowet og kommunikationen mellem forskellige enheder og systemer i netværket.
   - Netværksdiagrammer: Viser den fysiske og logiske struktur af netværket, herunder enheder, forbindelser og protokoller.
   - Device name & IP-adresseplan: En tabel eller liste over alle enheder på netværket med deres tildelte device name, IP-adresser, subnetmasker og gateways.
   - Revisionshistorik: En oversigt over ændringer og opdateringer til netværkskonfigurationen over tid.

- Oversigtsdiagram: Et oversigtsdiagram over produktionslinjen i Emulate3D og plantegning.

9. **Analyse:**

   - Diskuter potentielle anvendelsesområder for MQTT i industriel automation, især med hensyn til dets letvægt og evnen til at fungere på tværs af usikre netværk.

Krav:

- Grundlæggende forståelse af MQTT-protokollen og dets anvendelse i industriel kommunikation.
- Erfaring med netværksopsætning i TIA Portal og anvendelse af PLCSIM Advanced.
- Kompetence i at konstruere og debugge PLC-programmer, der involverer komplekse kommunikationsprotokoller.

Aflevering (optional): En detaljeret teknisk rapport med beskrivelse af opsætningen af netværket, MQTT-brokeren, og begge PLC'er. Der skal ligeledes medfølge skærmbilleder, kodeeksempler, og netværksdiagrammer, samt en dybdegående analyse af de udførte tests og resultater.

Note: Hvis det ikke er muligt at bruge en fysisk S7-1200 PLC og en simuleret S7-1500 PLC, kan to simulerede PLC'er anvendes i stedet.

12.12 OPC UA

Mål: Målet med denne opgave er at etablere og demonstrere OPC UA-kommunikation mellem to simulerede PLC'er ved hjælp af Siemens PLCSIM Advanced. Opgaven skal give studerende en praktisk forståelse af OPC UA-protokollens implementering og anvendelse i et simuleret industrielt miljø.

Opgavebeskrivelse:

1. **Planlægning og Dokumentation:**

   - Udarbejd en detaljeret plan for placering af PLC-tavleskabe og maskiner i Emulate3D. Tavleskabene skal være placeret med en afstand, der kræver kommunikation via et netværk.
   - **Note:** To rullebånd kan bruges til at symbolisere maskiner.
   - Dokumentér planen med et layoutdiagram, der viser placeringen af tavleskabe og maskiner.

2. **Scenarieopbygning i Emulate3D:**

- Design en simpel produktionslinje i Emulate3D, hvor en PLC læser status for knapper og sensorer og sender disse data via OPC UA til den anden PLC.
- Simuler sensorer og aktuatorer i Emulate3D og opret relevante tags til dem.

3. **Konfiguration i TIA Portal:**

   - Opstil to separate simulerede S7-1500 PLC'er i TIA Portal ved hjælp af PLCSIM Advanced, og konfigurer dem til henholdsvis at fungere som OPC UA-server og -klient.
   - På server-PLC'en, aktiver OPC UA-server funktionalitet og opret et sæt af tags eller variabler, som klient-PLC'en skal tilgå.
   - På klient-PLC'en, konfigurer OPC UA-klient funktionalitet til at forbinde til serveren og tilgå de nødvendige tags/variabler.
   - Tildel unikke IP-adresser til begge PLC'er og opsæt passende netværksparametre for at muliggøre kommunikation via Ethernet.

4. **Simulering med PLCSIM Advanced:**

   - Brug PLCSIM Advanced til at simulere begge PLC'er og etablere en virtuel forbindelse mellem dem.

5. **Implementering af OPC UA-kommunikation:**

   - Implementér et enkelt automatiseringsscenarie, hvor klienten kontinuerligt læser fra og/eller skriver til server-PLC'ens tags, for eksempel for at styre en proces eller overvåge sensorværdier.
   - Sikr at forbindelsen mellem OPC UA-serveren og -klienten er sikret ved hjælp af passende sikkerhedsforanstaltninger, såsom applikationsautentificering og kryptering.

6. **Test af dataoverførsel:**

   - Test dataoverførslen ved at sende forskellige typer af data fra serveren til klienten og verificer korrekt modtagelse og integritet (nøjagtighed).

7. **Fejlhåndtering:**

   - Implementer fejlhåndteringsmekanismer for at sikre pålidelighed og korrekt respons på eventuelle kommunikationsfejl.
   - Fejl gemmes i et array for yderligere analyse.

8. **Dokumentation:**

- Fysiske layouttegninger: Viser den fysiske placering af netværksudstyr, kabler og andre komponenter i automatiseringsmiljøet.
- Logiske diagrammer: Illustrerer dataflowet og kommunikationen mellem forskellige enheder og systemer i netværket.
- Netværksdiagrammer: Viser den fysiske og logiske struktur af netværket, herunder enheder, forbindelser og protokoller.
- Device name & IP-adresseplan: En tabel eller liste over alle enheder på netværket med deres tildelte device name, IP-adresser, subnetmasker og gateways.
- Revisionshistorik: En oversigt over ændringer og opdateringer til netværkskonfigurationen over tid.
- Oversigtsdiagram: Et oversigtsdiagram over produktionslinjen i Emulate3D og plantegning.

9. **Analyse:**
    - Diskuter potentielle anvendelsesområder for OPC UA i industriel automation, især med hensyn til dets standardisering og interoperabilitet på tværs af forskellige systemer.

Krav:

- Grundlæggende forståelse af OPC UA-protokollen og dens anvendelse i industriel kommunikation.
- Erfaring med netværksopsætning i TIA Portal og anvendelse af PLCSIM Advanced.
- Kompetence i at konstruere og debugge PLC-programmer, der involverer komplekse kommunikationsprotokoller.

Aflevering(optional): En detaljeret teknisk rapport med beskrivelse af opsætningen af netværket, OPC UA-konfigurationerne, og begge PLC'er. Der skal ligeledes medfølge skærmbilleder, kodeeksempler, og netværksdiagrammer, samt en dybdegående analyse af de udførte tests og resultater.

12.13 Others WEB Server

12.14 Opsætning af Webserver på en Fysisk og Simuleret PLC

Mål: Målet med denne opgave er at opsætte og konfigurere en webserver på en fysisk SIMATIC S7-1200 PLC og en simuleret S7-1500 PLC ved hjælp af Siemens PLCSIM Advanced. Opgaven skal give de studerende en praktisk

forståelse af webserverfunktionen på en PLC og dens anvendelse til overvågning og kontrol af industrielle processer.
Opgavebeskrivelse:

1. **Planlægning og Dokumentation:**
    - Udarbejd en detaljeret plan for placering af PLC-tavleskabe og maskiner i Emulate3D.
    - **Note:** To rullebånd kan bruges til at symbolisere maskiner.
    - Dokumentér planen med et layoutdiagram, der viser placeringen af tavleskabe og maskiner.
2. **Konfiguration i TIA Portal:**
    - Opret et nyt projekt i TIA Portal og tilføj en fysisk S7-1200 PLC og en simuleret S7-1500 PLC.
    - Tildel unikke IP-adresser til begge PLC'er og konfigurer netværksparametre for at muliggøre kommunikation via Ethernet.
    - Aktivér webserver-funktionaliteten i begge PLC'er ved at gå til PLC-egenskaberne og vælge webserverindstillingerne.
    - Opret brugerdefinerede websideindstillinger, der viser vigtige procesdata og kontroller.
3. **Simulering med PLCSIM Advanced:**
    - Brug PLCSIM Advanced til at simulere S7-1500 PLC'en og sikre, at den er korrekt forbundet til den virtuelle netværksinfrastruktur.
    - Start simuleringen og verificer, at webserveren på den simulerede PLC er tilgængelig via en webbrowser ved at indtaste PLC'ens IP-adresse.
4. **Implementering af Webserverindhold:**
    - Opret og implementer webserverindhold på begge PLC'er, der viser live data, såsom sensorværdier og statusindikatorer.
    - Implementér kontrolfunktioner på websiden, der gør det muligt at styre procesparametre direkte fra webbrowseren.
5. **Test og Verifikation:**
    - Test webserverens funktionalitet ved at navigere til forskellige sider og verificere, at data vises korrekt og opdateres i realtid.
    - Test kontrolfunktionerne ved at ændre procesparametre via webbrowseren og observere de tilsvarende ændringer i PLC-simuleringen.
6. **Fejlhåndtering:**

- Implementer fejlhåndteringsmekanismer for at sikre pålidelighed og robusthed af webserveren.
- Dokumentér eventuelle fejl og de trin, der blev taget for at løse dem.

7. **Dokumentation:**
   - Fysiske layouttegninger: Viser den fysiske placering af netværksudstyr, kabler og andre komponenter i automatiseringsmiljøet.
   - Logiske diagrammer: Illustrerer dataflowet og kommunikationen mellem forskellige enheder og systemer i netværket.
   - Netværksdiagrammer: Viser den fysiske og logiske struktur af netværket, herunder enheder, forbindelser og protokoller.
   - Device name & IP-adresseplan: En tabel eller liste over alle enheder på netværket med deres tildelte device name, IP-adresser, subnetmasker og gateways.
   - Revisionshistorik: En oversigt over ændringer og opdateringer til netværkskonfigurationen over tid.
   - Oversigtsdiagram: Et oversigtsdiagram over produktionslinjen i Emulate3D og plantegning.
8. **Analyse:**
   - Diskuter potentielle anvendelsesområder for webservere i industriel automation, især med hensyn til fjernovervågning og kontrol af processer.

Krav:

- Grundlæggende forståelse af webservere og deres anvendelse i industriel kommunikation.
- Erfaring med netværksopsætning i TIA Portal og anvendelse af PLCSIM Advanced.
- Kompetence i at konstruere og debugge PLC-programmer, der involverer webserverfunktioner.

Aflevering (optional): En detaljeret teknisk rapport med beskrivelse af opsætningen af netværket, webserverkonfigurationerne, og begge PLC'er. Der skal ligeledes medfølge skærmbilleder, kodeeksempler, og netværksdiagrammer, samt en dybdegående analyse af de udførte tests og resultater.

Mål: Formålet med denne opgave er at implementere og konfigurere en webserver på en SIMATIC S7-1500 PLC og tilgå den ved hjælp af en standard webbrowser. Dette vil demonstrere PLC'ens indbyggede kommunikationsfunktioner og studerendes evne til at anvende disse til overvågning og

kontrolformål.

Opgave:

1. Vælg en SIMATIC S7-1500 PLC i TIA Portal og aktiver webserverfunktionen i PLC'ens enhedsegenskaber.

2. Tildel PLC'en en passende IP-adresse og konfigurer de nødvendige netværksindstillinger for at sikre tilgængelighed på det lokale netværk.

3. Design en simpel webside ved hjælp af TIA Portals indbyggede webeditor, hvilket inkluderer grundlæggende kontrol- og overvågningselementer (f.eks. knapper, indikatorlamper og datavisning).

4. Anvend PLCSIM Advanced til at simulere PLC'en og test webserverens funktionalitet.

5. Brug en standard webbrowser til at tilgå PLC'ens webserver ved hjælp af dets IP-adresse og interager med de designede websideelementer for at udføre kontrol- og overvågningsopgaver.

6. Demonstrer evnen til at læse og skrive til/fra PLC datablokke via websiden.

7. Sikr webserverens kommunikation ved at konfigurere og anvende passende sikkerhedsindstillinger.

8. Evaluer webserverens ydeevne og brugeroplevelse ved at tilgå den fra forskellige enheder og browsere.

9. Forbered en detaljeret teknisk rapport, der beskriver opsætningsprocessen, brugerinterfacet, sikkerhedsaspekter og de opnåede testresultater.

Krav:

- Forståelse for webserverkoncepter og HTTP-protokollen.

- Færdigheder i brugen af TIA Portal til konfiguration af PLC'er og oprettelse af brugerinterfaces.

- Evnen til at implementere og teste industrielle kommunikationsnetværk.

Aflevering: En rapport, der inkluderer den komplette vejledning til opsætningen af webserveren, brugergrænsefladedesignet, sikkerhedskonfigurationer, og en diskussion af den praktiske anvendelse af webservere i industriel automation.

PLC-til-PLC kommunikation via Webserver og TCP/IP

Mål: Målet med denne opgave er at oprette en kommunikationsforbindelse mellem to SIMATIC S7-1500 PLC'er ved hjælp af en webserver og TCP/IP-protokollen. Studerende skal demonstrere evnen til at udveksle data mellem PLC'er over et netværk ved at anvende standard webteknologier.

Opgave:

1. Konfigurer to SIMATIC S7-1500 PLC'er i TIA Portal, og aktiver webserverfunktionerne på begge PLC'er.
2. Tildel hver PLC en unik IP-adresse, og sikr at de kan nå hinanden over det lokale netværk.
3. Opret et simpelt HTTP-baseret API på den første PLC's webserver, som tillader læsning og skrivning af specifikke datablokke.
4. Design en klient-side applikation på den anden PLC's webserver, der kan sende og modtage data ved hjælp af HTTP-anmodninger til den første PLC's API.
5. Simuler begge PLC'er med PLCSIM Advanced, og test kommunikationen mellem de to PLC'er ved at udveksle kontrolsignaler og procesdata via webserverne.
6. Analyser og håndter de potentielle sikkerhedsmæssige udfordringer forbundet med at tillade inter-PLC kommunikation over TCP/IP.
7. Dokumenter processen for at oprette og sikre kommunikationen, samt de trin der er taget for at verificere og validere dataudvekslingen.
8. Udforsk mulighederne for at overvåge og logge kommunikationen mellem PLC'erne ved hjælp af netværksanalyseværktøjer.
9. Afslutningsvis, diskutér i en teknisk rapport anvendelsen af webserverbaseret kommunikation i industrielle automatiseringssystemer og dens fordele og ulemper i forhold til traditionelle metoder.

Krav:

- Dybdegående kendskab til TCP/IP-protokollen og webserver teknologi.
- Kompetencer i at programmere og konfigurere SIMATIC S7-1500 PLC'er og tilhørende webservere.
- Evne til at skabe sikre og pålidelige netværkskommunikationsløsninger.

Aflevering: En detaljeret rapport, der omfatter opsætningsvejledninger, kommunikationsprotokolbeskrivelse, sikkerhedsforanstaltninger, testprocedurer og en evaluering af teknologiens anvendelighed i industriel automation.

PLC-til-PLC Kommunikation via Webserver og UDP

Mål: Formålet med denne opgave er at udforske en alternativ metode til at facilitere kommunikation mellem to SIMATIC S7-1500 PLC'er ved hjælp af UDP-protokollen koordineret gennem en webserver. Opgaven vil introducere de studerende for udfordringer og løsninger ved brug af mindre almindelige kommunikationsmetoder i automationsmiljøer.

Opgave:

1. Opstil to SIMATIC S7-1500 PLC'er i TIA Portal og implementer en webserver på hver PLC.
2. Konfigurer hver PLC med en unik IP-adresse inden for det samme subnet for at muliggøre netværkskommunikation.
3. Udvikl et script på webserveren, der initialiserer og konfigurerer en UDP-kommunikationskanal. Dette script vil agere som en 'startpakke' for at igangsætte UDP-kommunikationen mellem PLC'erne.
4. Skriv et PLC-program, der sender og modtager data via UDP. Brug webserveren til at aktivere og overvåge denne kommunikation.
5. Anvend PLCSIM Advanced til at simulere begge PLC'er og etabler en virtuel testmiljø, hvor UDP-pakker sendes og modtages mellem de to enheder.
6. Vurder effektiviteten og responsiviteten af UDP-kommunikationen i forhold til TCP i en webserverkontekst og dokumenter observationerne.
7. Fremhæv potentielle problemer såsom pakketab, og diskuter hvordan disse kan overvindes eller minimeres i et industrielt netværksmiljø.
8. Konkluder opgaven med en teknisk rapport, der indeholder dine erfaringer og refleksioner over brugen af UDP i sammenhæng med en webserver til PLC-kommunikation, herunder fordele, ulemper og potentielle brugsscenarier.

Krav:

- Grundlæggende forståelse for UDP og TCP/IP-protokollerne.
- Evner i konfiguration og anvendelse af webservere på SIMATIC S7-1500 PLC'er.
- Færdigheder i anvendelse af TIA Portal og PLCSIM Advanced for simulering af netværkskommunikation.

Aflevering: En detaljeret rapport med dine testresultater, kodeeksempler, netværkskonfigurationer og en evaluering af UDP's anvendelighed i PLC-kommunikation.

HTTP-kommunikation mellem to PLC'er via Webserver

Mål: Denne opgave har til formål at etablere en grundlæggende HTTP-kommunikation mellem to SIMATIC S7-1500 PLC'er ved at udnytte deres indbyggede webserver-kapacitet. Studerende vil udvikle forståelse for anvendelsen af HTTP-protokollen inden for industriel automation og hvordan man kan bruge denne protokol til at udveksle data mellem PLC'er.

Opgavebeskrivelse:

1. Konfigurér webserveren på begge SIMATIC S7-1500 PLC'er, og sikr at de er tilgængelige på netværket.
2. Udvikl et simpelt HTML-brugergrænseflade eller RESTful API, som gør det muligt for en PLC at sende HTTP-requests til en anden PLC's webserver.
3. Skriv et script eller et PLC-program, som kan sende HTTP-GET og POST-requests for at hente og sende data til/fra den anden PLC.
4. Implementér logik på modtager-PLC'en til at behandle indkommende HTTP-requests og udføre handlinger baseret på disse anmodninger, som for eksempel at ændre værdier i datablokke eller aktivere outputs.
5. Brug PLCSIM Advanced til at simulere begge PLC'er og deres webserveres funktionalitet.
6. Test kommunikationsprocessen grundigt for at sikre data bliver overført korrekt og pålideligt mellem de to PLC'er.
7. Diskutér potentielle industrielle anvendelser for HTTP-kommunikation mellem PLC'er og overvej sikkerhedsmæssige aspekter ved denne tilgang.
8. Dokumentér hele opsætningen, kode, testprocedurer og konklusioner i en detaljeret teknisk rapport.

Krav til dokumentation: Den tekniske rapport skal inkludere netværksdiagrammer, konfigurationsdetaljer, kodeudsnit, testresultater og en kritisk vurdering af tilgangens anvendelighed i en industriel kontekst.

CoAP-kommunikation mellem to PLC'er

Mål: Formålet med denne opgave er at etablere CoAP-kommunikation mellem to SIMATIC S7-1500 PLC'er. Dette skal introducere studerende for CoAP-protokollen som et letvægts alternativ til HTTP i ressourcebegrænsede systemer, der er typiske for industrielle Internet of Things (IoT)-applikationer.

Opgavebeskrivelse:

1. Opstil to SIMATIC S7-1500 PLC'er, og konfigurer dem til at kunne forbinde over et lokalt netværk.
2. Implementér CoAP-server funktionalitet på den ene PLC, således at den kan modtage og reagere på CoAP-requests.
3. Implementér CoAP-klient funktionalitet på den anden PLC, så den kan sende CoAP-requests.
4. Design et simpelt dataudvekslingsscenarie, hvor den ene PLC regelmæssigt anmoder om data fra den anden PLC, f.eks. sensorlæsninger eller statusopdateringer.
5. Sikr, at CoAP-klienten kan sende både GET og POST-requests til serveren, og at serveren kan respondere korrekt på disse requests.
6. Gennemfør simuleringen af begge PLC'er ved hjælp af PLCSIM Advanced, inklusiv deres CoAP-kommunikation.
7. Udfør grundige test for at verificere, at CoAP-kommunikationen fungerer som forventet og inden for de krævede tidsspecifikationer.
8. Analyser og diskuter hvordan CoAP kan integreres i industrielle automationsløsninger, og sammenlign det med andre protokoller som HTTP og MQTT i konteksten af IoT.
9. Afslut opgaven med en detaljeret rapport, der dokumenterer hele processen fra opsætning til test, inklusive netværksdiagrammer, konfigurationsdetaljer og en diskussion af resultaterne.

Krav til dokumentation: Rapporten skal inkludere skærmbilleder af konfigurationsindstillinger, kodeudsnit med kommentarer, testscenarier, samt en refleksion over anvendeligheden af CoAP i et industriel miljø og eventuelle sikkerhedsmæssige overvejelser.

MQTT-kommunikation mellem to PLC'er via Webserver og CDN

Mål: At konfigurere to PLC'er til at kommunikere med hinanden ved hjælp af MQTT over en webserver og benytte 'cdnjs' til at levere nødvendige JavaScript-biblioteker.

Opgavebeskrivelse:

1. Konfigurér webserveren på begge PLC'er til at levere en webapplikation med HTML og JavaScript-filer.
2. Inkludér `<script>` tags i HTML-dokumentet, der henviser til MQTT.js biblioteket hostet på 'cdnjs'.

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/
paho-mqtt/1.0.2/mqttws31.min.js"
type="text/javascript">
</script>
```

3. Brug Paho MQTT biblioteket i din webapplikations JavaScript til at oprette forbindelse til en MQTT Broker.
4. Abonner på relevante topics for begge PLC'er og implementér logik til at sende og modtage MQTT-beskeder.
5. Simulér PLC'ernes adfærd ved at sende og modtage kontrolbeskeder.
6. Overvåg og test kommunikationen mellem PLC'erne ved hjælp af MQTT Brokerens overvågningsværktøjer.
7. Fejlfind eventuelle kommunikationsproblemer og optimer forbindelsen for pålidelighed.
8. Dokumentér processen, konfigurationen af MQTT, og sikkerhedsovervejelser i en detaljeret rapport.

Krav til dokumentation: Den tekniske rapport skal indeholde konfigurationsdetaljer, kodeudsnit, en gennemgang af testproceduren, og en diskussion af sikkerhedsmæssige aspekter ved at bruge MQTT i en industriel kontekst. Den skal også vurdere anvendeligheden og pålideligheden af MQTT-kommunikation gennem en webserver understøttet af en CDN.

CoAP-kommunikation mellem to PLC'er via Webserver og CDN

Mål: At facilitere CoAP-kommunikation mellem to SIMATIC S7-1500 PLC'er ved at bruge en webserver og CDN til at levere nødvendige CoAP JavaScript-biblioteker.

Opgavebeskrivelse:

1. Konfigurér webserverne på begge SIMATIC S7-1500 PLC'er til at hoste en webside, som indeholder CoAP-klient logik.
2. Inkludér et `<script>` tag i din HTML, som refererer til et CoAP JavaScript-bibliotek gennem en CDN.

```
<script src="https://cdn.jsdelivr.net/npm/
coap-client-browserify@1/coap-client.js"
type="text/javascript"></script>
```

3. Benyt JavaScript og det inkluderede CoAP-bibliotek til at sende og modtage CoAP-beskeder mellem de to PLC'er.

4. Design og implementér et simpelt dataudvekslingsscenarie, hvor PLC'erne kan anmode om og sende forskellige ressourcer til hinanden.

5. Brug simulering til at efterligne PLC'ernes netværk og verificere den korrekte funktion af CoAP-meddelelserne.

6. Analyser dataflowet og CoAP-meddelelseslogikken for at sikre, at begge PLC'er korrekt kan håndtere anmodninger og ressourcer.

7. Udarbejd en grundig testprocedure og dokumenter interaktionerne mellem PLC'erne og valideringen af kommunikationsprotokollen.

8. Skriv en rapport, som inkluderer tekniske aspekter af CoAP-implementeringen, netværksopsætning, og overvejelser omkring brugen af CDN'er til industrielle applikationer.

Krav til dokumentation: Den tekniske rapport skal omfatte konfigurationsdetaljer, kodningseksempler, en diskussion af testmetoder, og en evaluering af CoAP's effektivitet og pålidelighed i en industriel kontekst. Rapporten bør også adressere sikkerhedsmæssige udfordringer og fordele ved at benytte CDN'er for at forbedre ydeevnen og tilgængeligheden af CoAP-tjenester.

AMQP-kommunikation mellem to PLC'er via Webserver og CDN

Mål: Målet med denne opgave er at etablere AMQP-kommunikation mellem to SIMATIC S7-1500 PLC'er ved brug af en webserver og CDN for at facilitere realtidsbeskedudveksling.

Opgavebeskrivelse:

1. Konfigurer webserveren på hver af de SIMATIC S7-1500 PLC'er til at tjene en webside, der vil fungere som en AMQP klient.

2. Indsæt et `<script>` tag i websidens HTML for at referere til et AMQP JavaScript-bibliotek hosted via en CDN.

```
<script src="https://cdn.jsdelivr.net/npm/
amqp-websocket-client/amqp-client.js"
type="text/javascript"></script>
```

3. Brug JavaScript til at benytte det inkluderede AMQP-bibliotek til at etablere en forbindelse mellem de to PLC'er og at muliggøre beskedudveksling.

4. Udvikl et scenarie for udveksling af beskeder, hvor PLC'erne regelmæssigt udveksler data såsom sensorværdier eller kontrolkommandoer.

5. Anvend PLC-simuleringsværktøjer til at teste og validere opsætningen og den succesfulde udveksling af AMQP-beskeder.

6. Sikr, at den etablerede kommunikationsforbindelse overholder de relevante tidskrav og sikkerhedsstandarder.

7. Dokumentér hvert skridt i opsætningsprocessen, fra konfigurering af webserver og PLC til udviklingen af AMQP-klientlogikken.

8. Afslut med at udarbejde en detaljeret teknisk rapport, der dokumenterer opsætningen, kommunikationsprotokollen, og de udførte tests.

Krav til dokumentation: Rapporten skal inkludere en gennemgang af de tekniske implementeringer, skærmbilleder af konfigurationer, eksempler på kode, beskrivelse af testscenarier, og en vurdering af AMQP-kommunikation i en industriel automationskontekst. Overvejelser vedrørende brug af CDN'er for at forbedre ydelsen og pålideligheden af kommunikationen bør også inkluderes.

WebSocket-kommunikation mellem to PLC'er via Webserver

Mål: Denne opgave fokuserer på at opsætte en WebSocket-kommunikation mellem to SIMATIC S7-1500 PLC'er ved hjælp af en webserver. Målet er at forstå og anvende WebSockets til at opnå tovejskommunikation i realtid mellem PLC'er.

Opgavebeskrivelse:

1. Konfigurer en webserver på hver PLC til at hoste en webside, der fungerer som en WebSocket-klient.

2. På websiden, indsæt et `<script>` tag der inkluderer JavaScript-kode for at etablere og håndtere en WebSocket-forbindelse.

```
<script type="text/javascript">
var ws = new WebSocket('ws://PLC_SERVER_ADRESSE');
ws.onopen = function() {
 // Kode for når forbindelsen åbnes
};
ws.onmessage = function(evt) {
 // Kode for at håndtere indkommende beskeder
};
ws.onclose = function() {
 // Kode for når forbindelsen lukkes
```

```
};
</script>
```

3. Skriv logikken for at sende og modtage data over WebSocket-forbindelsen. Dette skal omfatte både at håndtere forbindelsen og at formatere dataene korrekt.

4. Udvikl et enkelt scenarie, hvor PLC'erne kommunikerer, for eksempel udveksling af statusopdateringer eller kommandoer.

5. Test kommunikationsforbindelsen ved hjælp af en simuleret opsætning for at sikre, at dataudvekslingen fungerer som forventet.

6. Undersøg mulighederne og udfordringerne ved at bruge WebSocket i en industriel automationssammenhæng.

7. Dokumentér opsætningsprocessen, udvikling af klientlogikken og testresultaterne i en detaljeret rapport.

Krav til dokumentation: Rapporten skal indeholde beskrivelser af de tekniske opsætninger, eksempler på den anvendte kode, testscenarier og resultater, samt en diskussion af fordele og ulemper ved at bruge WebSockets i industrielle automationssystemer. Det bør også inkludere overvejelser omkring sikkerhedsaspekterne ved brug af WebSockets.

OBS! husk at udskifte PLC_SERVER_ADRESSE med den faktiske adresse til PLC-webserveren, der håndterer WebSocket-forbindelser. Koden antager, at PLC-webserveren er konfigureret til at understøtte WebSockets og er i stand til at acceptere indkommende WebSocket-forbindelser.

Hente Data fra Firebase Realtime Database ved Hjælp af Axios

Mål: Denne opgave har til formål at udvikle studerendes evner til at integrere en Firebase Realtime Database i en webapplikation for at hente data, der kan bruges i automatiseringsscenarier og overvågning.

Opgavebeskrivelse:

1. Opret en Firebase-konto og konfigurér en ny Firebase Realtime Database.

2. Definer en datastruktur i Firebase-databasen, der kunne simulere relevante data for automatisering, såsom sensorlæsninger eller systemstatus.

3. Indhent dine Firebase-databaseindstillinger og adgangsnøgler, der er nødvendige for at autentificere anmodninger.

4. Implementér Axios i dit webservermiljø for at kunne sende HTTP-anmodninger til Firebase REST API'et:

```
<script src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script>
\script type="text/javascript">
const firebaseConfig = {
 // Din Firebase-konfiguration
 apiKey: "DIN_API_NØGLE",
 authDomain: "DIN_PROJECT_ID.firebaseapp.com",
 databaseURL: "https://DIN_PROJECT_ID.firebaseio.com",
 // ... andre nødvendige konfigurationsdetaljer
};

function fetchFirebaseData() {
 const path = 'din/data/sti';
 const url = \`${firebaseConfig.databaseURL}/${path}.json\`;

 axios.get(url)
 .then(function (response) {
  // Håndter succesfuld respons
  console.log(response.data);
 })
 .catch(function (error) {
  // Håndter fejl
  console.error(error);
 });
}
</script>
```

5. Tilføj en knap til din webside, der når den klikkes på, kalder funktionen 'fetchFirebaseData()' for at hente og vise data fra Firebase.

6. Test din webapplikations evne til at hente data fra Firebase og sikre, at den håndterer data korrekt.

7. Diskutér, hvordan integrationen af Firebase kan udvide funktionaliteten i automatiseringssystemer, især med hensyn til realtidsdataovervågning og -styring.

8. Afslut med at dokumentere hele processen, inklusiv konfigurationsindstillinger og testresultater, i en teknisk rapport.

Krav til dokumentation: Den tekniske rapport skal indeholde en fuld beskrivelse af Firebase-konfigurationen, JavaScript-kodeeksempler, et UI-layout til at demonstrere datahentning, og en diskussion af de opnåede resultater samt potentielle sikkerhedsaspekter ved at bruge cloud-baserede databaser i industriel automatisering.

12.15 Profinet

12.15.1 Profinet til UR og Dimensionering af PROFINET-netværk

Mål: Denne opgave består af to dele. I første del er målet at konfigurere og demonstrere PROFINET-kommunikation mellem en Siemens S7-1200 PLC og en Universal Robots (UR) robot, eller mellem en simulerede S7-1500 PLC og simuleret UR. Anden del fokuserer på at designe og dimensionere et PROFINET-netværk, der skal understøtte en produktionslinje med seks robotceller og et SCADA-system. Samlet set vil opgaven give de studerende praktisk erfaring med både opsætning af PROFINET-kommunikation og netværksdesign i et komplekst industrielt miljø.

12.15.2 Del 1: Opsætning af PROFINET-kommunikation

Opgavebeskrivelse:

1. **Forberedelse af GSDML-fil:**
    - Download den relevante GSDML-fil til Universal Robots (UR) fra producentens hjemmeside.
    - Sørg for, at filen er kompatibel med den version af Siemens TIA Portal, der anvendes.
    - Importer GSDML-filen i TIA Portal for at gøre UR robotten tilgængelig som en PROFINET-enhed.
2. **Valg af opsætning:**
    - **Option 1:** Brug en fysisk S7-1200 PLC til at kommunikere med en fysisk UR robot.
    - **Option 2:** Brug en fysisk S7-1200 PLC til at kommunikere med en simuleret UR robot.
    - **Option 3:** Brug en simulerede S7-1500 PLC og en simuleret UR til at simulere kommunikationen.
3. **Opsætning af PROFINET i TIA Portal:**
    - Start et nyt projekt i TIA Portal og opret en ny konfiguration med den valgte PLC.

- Tilføj UR robotten eller den anden PLC som en PROFINET-enhed i netværkskonfigurationen ved hjælp af den importerede GSDML-fil.
- Tildel en unik IP-adresse til UR robotten eller den anden PLC, der er inden for det samme subnet som din valgte PLC.
- Konfigurer de nødvendige PROFINET-parametre, herunder cyklustider, I/O-adressering, og forbindelsesparametre.

4. **Kommunikationstest og I/O Mapping:**
   - Udfør en hardware download til din PLC og verificer, at den oprettede PROFINET-forbindelse er korrekt og aktiv.
   - Opsæt I/O-mapping i TIA Portal for at definere, hvilke input og output data der skal udveksles mellem PLC'en og UR robotten eller den anden PLC.
   - Test kommunikationen ved at sende en simpel kommando fra PLC'en til UR robotten eller den anden PLC (f.eks. at starte en bevægelse eller ændre en hastighed).
   - Verificer, at enheden reagerer korrekt på kommandoen, og at feedback-data fra enheden vises korrekt i PLC'en.
5. **Simuleringsscenarie i UR:**
   - Hvis du bruger en UR robot, opret et simpelt program i UR's kontrolsoftware, hvor robotten udfører en bestemt handling (f.eks. flytter en genstand fra et punkt til et andet) baseret på modtagne PROFINET-kommandoer.
   - Hvis du bruger en anden PLC, opsæt en kommunikationslogik, der simulerer en maskinhandling eller proces.

12.15.3 Del 2: Dimensionering af PROFINET-netværk for Produktionslinje med Robotceller

Opgavebeskrivelse:

1. **Scenarie og Krav:**
   - Produktionslinjen består af seks robotceller, hvor hver celle indeholder en Universal Robots (UR) robot, en Siemens S7-1200 PLC og et SCADA-anlæg til dataopsamling.
   - Robotcellerne er fordelt mellem to haller, med fire celler i Hal A og to celler i Hal B. De to haller er fysisk adskilt.
   - Produktionslinjen skal kommunikere via PROFINET, og der skal opsamles data i realtid fra hver robotcelle til SCADA-systemet.

2. **Netværkstopologi:**

   - Vælg en passende netværkstopologi, der kan sikre effektiv og pålidelig kommunikation mellem alle robotceller og SCADA-systemet. Overvej topologier som stjerne, linje eller ring.
   - Overvej hvordan kommunikationen mellem de to haller skal håndteres, og om der er behov for en bro, router eller fiberforbindelse mellem hallerne.

3. **Kabellængder og Netværkskomponenter:**

   - Beregn de nødvendige kabellængder for hver forbindelse mellem robotceller, SCADA-anlæg og netværksswitche. Tag hensyn til de maksimale længder for Ethernet-kabler (f.eks. Cat5e, Cat6) og eventuelt behov for fiberkabler mellem hallerne.
   - Dimensionér netværket ved at vælge antal og placering af netværksswitche, routere og eventuelle repeatere for at sikre tilstrækkelig dækning og kapacitet i hele produktionslinjen.

4. **Design af Netværksinfrastruktur:**

   - Tegn et detaljeret netværksdiagram, der viser alle robotceller, SCADA-anlæg, switche, routere og kabelløb mellem enhederne. Diagrammet skal tydeligt illustrere forbindelsen mellem Hal A og Hal B.
   - Angiv i diagrammet de anvendte kabeltyper og deres specifikke længder, samt de nødvendige komponenter som switche og routere.

5. **Overvejelser om Sikkerhed og Pålidelighed:**

   - Vurder behovet for redundans i netværket, for eksempel ved at implementere ringtopologi eller anvende flere switche for at undgå enkeltpunktsfejl.
   - Diskuter hvilke sikkerhedsforanstaltninger, der bør implementeres for at beskytte mod netværksfejl eller uautoriseret adgang, såsom VLAN-segmentering, firewall, eller adgangskontrol.

Aflevering: En detaljeret teknisk rapport, der beskriver opsætningen, konfigurationen, og testresultaterne fra Del 1, samt netværksdesign og dimensionering fra Del 2. Rapporten skal inkludere netværksdiagrammer, skærmbilleder fra TIA Portal, beregninger af kabellængder og komponentplaceringer, samt en vurdering af netværkets pålidelighed og sikkerhed.

12.16 Profibus

Del 1: Installation og konfiguration af PROFIBUS-netværk med én ET200 modul

Mål: I denne del af opgaven skal de studerende lære at forbinde og konfigurere en S7-1200 PLC med et ET200-modul ved hjælp af PROFIBUS. De studerende skal selv fremstille et PROFIBUS-kabel og sikre, at kommunikationen mellem PLC'en og ET200-modulet fungerer korrekt.

Opgavebeskrivelse:

1. **Fremstilling af PROFIBUS-kabel:**
   - Fremstil et PROFIBUS-kabel, der skal forbinde S7-1200 PLC'en med ET200-modulet. Følg de nødvendige specifikationer for kabellængde og terminering.
2. **Konfiguration af netværket i TIA Portal:**
   - Opret et nyt projekt i TIA Portal, og tilføj S7-1200 PLC'en og ET200-modulet.
   - Konfigurer PROFIBUS-netværket og tildel unikke adresser til hver enhed.
3. **Test af kommunikation:**
   - Gå online med S7-1200 PLC'en i TIA Portal, og verificer, at der er stabil kommunikation mellem PLC'en og ET200-modulet.
   - Brug diagnoseværktøjer i TIA Portal til at sikre, at der ikke er fejl i netværkskommunikationen.

Del 2: Tilslutning af et ekstra ET200 modul til PROFIBUS-netværket

Mål: I denne del skal de studerende udvide det eksisterende PROFIBUS-netværk ved at tilføje endnu et ET200-modul. De skal igen fremstille et PROFIBUS-kabel og sikre, at begge moduler fungerer korrekt i netværket.

Opgavebeskrivelse:

1. **Fremstilling af ekstra PROFIBUS-kabel:**
   - Fremstil et nyt PROFIBUS-kabel, der skal forbinde det første ET200-modul med det andet ET200-modul.
2. **Opdatering af netværkskonfiguration:**

- Opdater netværkskonfigurationen i TIA Portal ved at tilføje det nye ET200-modul.
- Tildel en unik adresse til det nye ET200-modul, og konfigurer netværksparametrene.

3. **Test af udvidet netværk:**
    - Gå online med S7-1200 PLC'en i TIA Portal, og verificer, at begge ET200-moduler kommunikerer korrekt med PLC'en.
    - Brug diagnoseværktøjer til at teste netværkets stabilitet og pålidelighed.

Del 3: Design og implementering af et stort PROFIBUS-netværk

Mål: Denne delopgave udfordrer de studerende til at designe og implementere et komplekst PROFIBUS-netværk, der inkluderer flere ET200-moduler og Danfoss frekvensomformere. De skal tage højde for fysiske begrænsninger som kabellængder og anvende repeatere, hvor det er nødvendigt.

Opgavebeskrivelse:

1. **Design af netværket:**
    - Tegn et netværksdiagram, der viser forbindelserne mellem S7-1200 PLC'en, 7 ET200-moduler og 5 Danfoss frekvensomformere.
    - Marker kabellængderne mellem hver enhed, og angiv, hvor der skal bruges repeatere.
    - Bestem og dokumentér adressering for hver enhed.
2. **Fremstilling og installation:**
    - Fremstil nødvendige PROFIBUS-kabler til at forbinde alle enhederne og installer repeatere, hvor det er nødvendigt.
    - Tilslut alle enheder til netværket i overensstemmelse med det designede netværksdiagram.
3. **Konfiguration i TIA Portal:**
    - Tilføj og konfigurer alle 7 ET200-moduler og 5 Danfoss frekvensomformere i TIA Portal.
    - Sørg for korrekt opsætning af repeatere og verificér netværkets stabilitet.
4. **Test af netværkets ydeevne:**
    - Gå online med S7-1200 PLC'en i TIA Portal, og brug diagnoseværktøjer til at teste netværkets ydeevne.

- Verificer, at alle enheder kommunikerer korrekt, og at netværket er stabilt og pålideligt.

Krav til dokumentation: Studerende skal aflevere en teknisk rapport, der inkluderer netværksdiagrammer, adresseringsplaner, testresultater. Opgaven skal dokumenteres med skærmbilleder og tekst. I opgaven skal fremgå et teori/metode afsnit.

Kapitel 13

Rockwell

13.1 Studio 5000 Netværksopgaver

13.2 Opsætning af EtherNet/IP Netværk i Studio 5000

Mål: Målet med denne opgave er at lære, hvordan man opretter og konfigurerer et EtherNet/IP-netværk i Studio 5000. Dette er afgørende for effektiv kommunikation mellem Allen-Bradley PLC'er og andre enheder i et industrielt netværk.
Opgavebeskrivelse:

1. **Planlægning og Dokumentation:**
    - Udarbejd en detaljeret plan for netværksopsætningen, herunder IP-adressering og netværkstopologi.
    - Dokumentér planlægningen med et netværksdiagram, der viser alle enheder og deres tilhørende IP-adresser.
2. **Konfiguration i Studio 5000:**
    - Start Studio 5000 og opret et nyt projekt baseret på din specifikke PLC-model.
    - Tilføj EtherNet/IP-moduler til projektet ved hjælp af Hardware Tree.
    - Konfigurer IP-adresser og netværksindstillinger for hvert modul for at sikre korrekt kommunikation på netværket.
3. **Etablering af Forbindelser:**
    - Brug 'Who Active' og 'RSWho' værktøjerne i RSLinx Classic til at etablere en rute til en ekstern enhed.

- Opret og konfigurer en Produced and Consumed datakonfiguration for at sende og modtage data mellem PLC'er.
- Tilføj Remote I/O-enheder i I/O Configuration, og konfigurer deres forbindelse til controlleren.

4. **Test og Verifikation:**
    - Gennemfør en Ping-test for at validere netværkskommunikationen til alle enhederne.
    - Test og verificér dataudveksling mellem PLC'er og I/O-enheder ved hjælp af tagbaseret programmering i Studio 5000.
5. **Dokumentation:**
    - Udarbejd netværksdiagrammer, IP-adressetabeller og skærmbilleder af din konfiguration.
    - Dokumentér processen og resultaterne i en teknisk rapport.

Krav:

- Grundlæggende forståelse af EtherNet/IP-protokollen og dets anvendelse i industriel kommunikation.
- Erfaring med konfiguration og programmering i Studio 5000.
- Evne til at fejlsøge netværksproblemer ved hjælp af diagnostiske værktøjer i RSLinx Classic.

Aflevering: En detaljeret teknisk rapport, der beskriver opsætningen, konfigurationen, og testresultaterne, inklusive netværksdiagrammer og skærmbilleder.

13.3 Ændring af IP-adresse med BOOTP

Mål: Målet med denne opgave er at lære, hvordan man bruger BOOTP (Bootstrap Protocol) til at tildele eller ændre IP-adressen på en SIMATIC S7-1200 PLC. Dette er en vigtig færdighed, når du arbejder med enheder, der bruger dynamiske IP-adresser eller skal konfigureres til et bestemt netværk.
Opgavebeskrivelse:

1. **Opsætning:**
    - Sørg for, at din PLC er korrekt tilsluttet netværket, og at BOOTP/DHCP er aktiveret på enheden.
    - Forbind din computer til det samme netværk som PLC'en.
2. **Installation og Konfiguration af BOOTP Server:**

- Download og installer Siemens BOOTP Server eller et andet kompatibelt BOOTP-værktøj på din computer.
- Start BOOTP Server og konfigurer den til at lytte på det netværksinterface, som din PLC er tilsluttet.

3. **Registrering af PLC:**
   - Sæt din PLC til at anmode om en IP-adresse ved at nulstille netværkskonfigurationen (eventuelt ved at bruge en speciel knap på enheden).
   - BOOTP Serveren bør registrere en forespørgsel fra PLC'en, identificeret ved dens MAC-adresse.
   - Tilføj PLC'en til BOOTP Serverens liste og tildel den en statisk IP-adresse.
4. **Tildeling af IP-adresse:**
   - Efter at have tildelt IP-adressen, skal du bruge BOOTP Serveren til at sende IP-adressen til PLC'en.
   - Verificér, at PLC'en har modtaget og accepteret den nye IP-adresse ved at pinge enheden fra din computer.
5. **Test og Dokumentation:**
   - Test forbindelsen ved at oprette forbindelse til PLC'en via TIA Portal og verificér, at den fungerer korrekt med den nye IP-adresse.
   - Dokumentér alle trin, herunder skærmbilleder af BOOTP-konfigurationen og testresultaterne.

Krav:

- Grundlæggende forståelse af netværkskonfiguration og IP-adressering.
- Erfaring med brug af Siemens TIA Portal og netværksværktøjer.
- Kendskab til BOOTP og DHCP-protokoller.

Aflevering: En teknisk rapport, der beskriver konfigurationsprocessen, IP-adresseringen, samt testresultaterne, inklusive skærmbilleder og ping-tests.

13.4 Producer/Consumer Tags

Mål: Målet med denne opgave er at konfigurere to ECHO Simulatorer til at udveksle data ved hjælp af Producer/Consumer tags i en simulering af en EtherNet/IP-netværksinfrastruktur. Opgaven skal give de studerende en praktisk forståelse af Producer/Consumer kommunikationsmodellen og dens anvendelse i industrielle miljøer.

Opgavebeskrivelse:

1. **Oprettelse af Projekter i Studio 5000:**
    - Brug Studio 5000 til at oprette to separate projekter, hver repræsenterende en virtuel PLC konfigureret til at arbejde med ECHO Simulator.
2. **Konfiguration af ECHO Simulatorer:**
    - Konfigurér hver ECHO Simulator til at fungere som henholdsvis en Producer og en Consumer af data.
    - Definér et sæt af tags på Producer-simulatoren, der skal deles med Consumer-simulatoren, og omvendt.
3. **Opsætning af CIP Connection:**
    - Opsæt den korrekte CIP-connection (Common Industrial Protocol) i begge simulatorer for at tillade korrekt dataudveksling.
4. **Dataoverførsel og Verifikation:**
    - Anvend Producer/Consumer-modellen til at overføre data fra den ene ECHO Simulator til den anden og bekræft overførslen ved at monitorere tag-værdierne på begge enheder.
5. **Simulering af Driftstilstande:**
    - Simulér forskellige driftstilstande, herunder normal drift, pakketab og genforbindelse efter netværksafbrydelse.
6. **Analyse og Dokumentation:**
    - Analyser overførselshastigheder og latenstider under forskellige belastninger og netværksforhold.
    - Dokumentér setuppet, konfigurationsprocessen, og resultaterne af din dataudvekslingstest, herunder skærmbilleder og beskrivelser af de observationer, der er gjort under simulationen.

Krav til dokumentation: Rapporten skal indeholde følgende elementer:

- Detaljerede beskrivelser af hver ECHO Simulators konfiguration.
- Netværksdiagrammer, der viser datastrømme og forbindelserne mellem de to simulatorer.
- Diskussion af de anvendte Producer/Consumer tag konfigurationer og deres formål.
- Gennemgang af de testscenarier, der er kørt, og de fundne resultater.

- Analyse af simulatorens præcision og effektivitet i forhold til en reel PLC-setup.
- Refleksioner over betydningen af nøjagtig simulation og mulige forbedringer i metoder til netværksfejlsøgning.

13.5 Modbus TCP

Modbus - Rockwell til Rockwell

Mål: Målet med denne opgave er at konfigurere og teste Modbus-kommunikation mellem to ECHO Simulatorer for at simulere en industriel netværksforbindelse og dataudveksling mellem enheder. Denne øvelse skal give de studerende praktisk erfaring med Modbus-protokollen og dens anvendelse i industrielle systemer.

Opgavebeskrivelse:

1. **Oprettelse af Projekter i Studio 5000:**
   - Brug Studio 5000 til at oprette to separate projekter, hvor hvert projekt repræsenterer en virtuel PLC konfigureret til at arbejde med ECHO Simulator. Den ene simulator skal fungere som Modbus Master, og den anden som Modbus Slave.
2. **Konfiguration af Kommunikationsparametre:**
   - Indstil den passende Modbus-adresse, baudrate, parity, stopbits, og andre nødvendige kommunikationsparametre på begge simulatorer.
3. **Opsætning af Modbus-registre:**
   - Definér og konfigurér Modbus-registerne i Slave-simulatoren, som skal indeholde de data, der skal tilgås af Master-simulatoren.
4. **Dataforespørgsel og -skrivning:**
   - Skriv et script i Master-simulatoren, der anmoder om data fra Slave-simulatorens registre ved hjælp af Modbus-funktionskoder, såsom læsning af holde-registre (03) eller input-registre (04).
   - Konfigurér Master-simulatoren til at skrive værdier til Slave-simulatorens registre ved hjælp af relevante Modbus-funktionskoder, såsom skriv til enkelt register (06) eller flere registre (16).
5. **Simulering og Test:**
   - Simulér og test kommunikationen for at sikre, at data udveksles præcist og pålideligt mellem Master og Slave.

- Observér og dokumentér hvordan de to simulatorer håndterer forbindelsesafbrydelser og genetableringer.

6. **Analyse af Ydeevne:**

   - Analyser ydeevnen af Modbus-kommunikationen under forskellige belastningsforhold og diskuter dens anvendelighed i en industriel sammenhæng.

Krav til dokumentation: Den tekniske rapport skal omfatte:

- Udførlige beskrivelser af konfigurationsindstillingerne for Modbus-kommunikationen i begge ECHO Simulatorer.
- Diagrammer, der viser datastrømmene og de logiske forbindelser mellem Master- og Slave-simulatorerne.
- Forklaringer på de valgte Modbus-registre og deres anvendelse i testscenariet.
- Detaljerede beskrivelser af testscenarierne, herunder skærmbilleder og forklaringer på observerede dataudvekslinger.

Modbus - Rockwell til Siemens

Mål: Målet med denne øvelse er at konfigurere og teste Modbus TCP kommunikation mellem to simulerede PLC'er ved hjælp af PLCSIM Advanced, hvilket vil give en forståelse for netværkskommunikation mellem automationsenheder.

Opgavebeskrivelse:

1. Start med at oprette to separate projekter i TIA Portal, og indlæs dem i PLCSIM Advanced, hvor de skal fungere som Modbus TCP Master og Modbus TCP Slave.
2. Konfigurér netværksindstillinger for begge simulerede PLC'er, så de kan kommunikere på samme virtuelle netværk. Indstil relevante IP-adresser og subnet masker.
3. På Slave-PLC'en, konfigurér de relevante datablokke (DBs) til at fungere som Modbus-registerområder, som vil være tilgængelige for Master-PLC'en.
4. På Master-PLC'en, skriv et program, som bruger Modbus TCP-biblioteksfunktioner til at anmode om data fra Slave-PLC'ens registre og skriv data til disse registre.

5. Simuler begge PLC'er og etabler en kommunikationsforbindelse mellem dem. Test at Master-PLC kan læse fra og skrive til Slave-PLC'ens Modbus-registre.

6. Analysér og optimer kommunikationen for effektivitet og stabilitet. Overvej eksempelvis anvendelsen af asynkron kommunikation og genoprettelsesmekanismer ved forbindelsestab.

7. Dokumentér og diskutér anvendelsen af Modbus-kommunikation i et automationsmiljø og hvordan simuleret testning kan overføres til den virkelige verden.

Krav til dokumentation: Den tekniske rapport skal indeholde:

- En gennemgang af de anvendte netværkskonfigurationer og -indstillinger for begge PLC'er.
- En forklaring på strukturen og brugen af datablokke som Modbus-registre i Slave-PLC'en.
- En detaljeret beskrivelse af Master-PLC's Modbus-klientprogram med kodeeksempler og forklaringer af de anvendte funktioner.
- Skærmbilleder fra PLCSIM Advanced, der viser den vellykkede Modbus-kommunikation og de data, der udveksles mellem Master og Slave.
- En kritisk analyse af de opnåede testresultater, herunder forsinkelsestider, kommunikationssikkerhed og potentielle fejlkilder.
- Anbefalinger til optimering af kommunikationsprotokollen for at forbedre ydeevnen og pålideligheden i et reelt automationsanlæg.

13.6 OPC UA

Mål: Denne opgave er designet til at instruere studerende i at konfigurere og facilitere OPC UA-kommunikation mellem to Echo simulatorer ved hjælp af Rockwell Automation software. Det vil fremme en forståelse af OPC UA-protokollen og dens anvendelse i simulering af kontrolsystemer.

Opgavebeskrivelse:

1. Brug Rockwell Automation's Studio 5000 til at oprette to kontrolprogrammer, der kan køre på Echo simulatorer, og konfigurer dem til at simulere OPC UA-server og -klient funktionalitet.

2. Konfigurer en Echo Simulator til at fungere som en OPC UA-server, og eksponer et sæt af tags eller data punkter, som klienten vil abonnere på.

3. Konfigurer den anden Echo Simulator som en OPC UA-klient, der opretter forbindelse til serveren og abonnerer på de eksponerede tags/-data.

4. Etabler sikkerhedsmekanismer såsom certifikater og kryptering for at sikre, at kommunikationen mellem server og klient er sikker.

5. Demonstrér, at klienten kan læse og skrive til de tags, som serveren eksponerer, og implementér en kontrollogik, der afhænger af denne dataudveksling.

6. Udfør en række tests for at bekræfte, at kommunikationen fungerer som forventet, herunder fejltolerance og genforbindelsesmekanismer.

7. Analyser og rapportér kommunikationens performance og stabilitet, og identificér eventuelle begrænsninger eller problemer.

8. Skriv en teknisk rapport, der dokumenterer hele processen fra start til slut, herunder de anvendte konfigurationer, simuleringsresultater og eventuelle udfordringer eller læringspunkter.

Krav til dokumentation: Rapporten skal indeholde skærmbilleder af simulatorens og Studio 5000-programmets indstillinger, et flowdiagram der beskriver kommunikationsprocessen, kodeeksempler, og en diskussion om anvendeligheden af OPC UA i simulerede industrielle miljøer.

Kapitel 14

KepServerEX

14.1 OPC UA

Konfiguration af OPC UA Kommunikation for Siemens PLC

Mål: Opgavens formål er at konfigurere og afprøve OPC UA-kommunikation mellem en Siemens PLC og KEPServerEX. Studerende vil lære at etablere netværksforbindelser, der muliggør udveksling af data mellem Siemens PLC og KEPServerEX, der understøtter OPC UA-protokollen.

Opgavebeskrivelse:

1. Konfigurer Siemens PLC Netværksinterface:
    - Sørg for, at Siemens PLC (f.eks. S7-1500) er korrekt forbundet til det lokale netværk med en statisk IP-adresse, der kan kommunikere med KEPServerEX serveren.
2. Aktiver OPC UA Server på Siemens PLC:
    - Gennem TIA Portal, aktiver OPC UA serveren på Siemens PLC'en og konfigurer nødvendige sikkerhedsindstillinger (certifikater, brugerautentificering).
3. Konfigurer KEPServerEX til at Forbinde med Siemens PLC:
    - Åbn KEPServerEX og tilføj en ny OPC UA Client Channel.
    - Indtast Siemens PLC'ens IP-adresse og portnummer.
    - Importer eller tilføj de nødvendige OPC UA tags fra Siemens PLC til KEPServerEX.
4. Definer Tags i KEPServerEX:
    - Definer nødvendige tags i KEPServerEX, der skal læses fra eller skrives til Siemens PLC.

- Brug TIA Portal til at kortlægge disse tags til de respektive PLC-adresser.

5. Opret et PLC Program til Dataudveksling:
   - I TIA Portal, opret et PLC-program, der læser og skriver værdier til de definerede OPC UA tags.
   - Implementér logik til periodisk at opdatere PLC'ens interne variable baseret på OPC UA tagværdier.
6. Test OPC UA Kommunikation:
   - Udfør en serie tests for at bekræfte, at OPC UA-kommunikationen fungerer korrekt.
   - Sikr, at KEPServerEX kan modtage og sende data pålideligt til og fra Siemens PLC.
7. Dokumentér Opsætningen:
   - Dokumentér hele opsætningen og testprocessen, herunder detaljerede netværkskonfigurationer, programlistings og beskrivelser af de registrerede testresultater.
8. Udarbejd en Teknisk Rapport:
   - Afslut opgaven med en teknisk rapport, der indeholder en diskussion om anvendeligheden af OPC UA i sammenhæng med industrielt automatisering, udfordringer ved implementeringen og eventuelle løsninger.

Krav til dokumentation: Rapporten skal indeholde skærmbilleder og beskrivelser af de foretagne indstillinger, PLC-programmer og logiske fløde, tests og analyser af resultater, og en evaluering af OPC UA som en kommunikationsprotokol i industrielle anvendelser.

Konfiguration af OPC UA Kommunikation for Rockwell Automation PLC

Mål: Opgavens formål er at konfigurere og afprøve OPC UA-kommunikation mellem en Rockwell Automation PLC (Allen-Bradley) og KEPServerEX. Studerende vil lære at etablere netværksforbindelser, der muliggør udveksling af data mellem Rockwell PLC og KEPServerEX, der understøtter OPC UA-protokollen.

Opgavebeskrivelse:

1. Konfigurer Rockwell PLC Netværksinterface:

- Sørg for, at Rockwell Automation PLC (f.eks. ControlLogix) er korrekt forbundet til det lokale netværk med en statisk IP-adresse, der kan kommunikere med KEPServerEX serveren.

2. Aktiver OPC UA Server på Rockwell PLC:

   - Gennem Rockwell Studio 5000, aktiver OPC UA serveren på Rockwell PLC'en og konfigurer nødvendige sikkerhedsindstillinger (certifikater, brugerautentificering).

3. Konfigurer KEPServerEX til at Forbinde med Rockwell PLC:

   - Åbn KEPServerEX og tilføj en ny OPC UA Client Channel.
   - Indtast Rockwell PLC'ens IP-adresse og portnummer.
   - Importer eller tilføj de nødvendige OPC UA tags fra Rockwell PLC til KEPServerEX.

4. Definer Tags i KEPServerEX:

   - Definer nødvendige tags i KEPServerEX, der skal læses fra eller skrives til Rockwell PLC.
   - Brug Rockwell Studio 5000 til at kortlægge disse tags til de respektive PLC-adresser.

5. Opret et PLC Program til Dataudveksling:

   - I Rockwell Studio 5000, opret et PLC-program, der læser og skriver værdier til de definerede OPC UA tags.
   - Implementér logik til periodisk at opdatere PLC'ens interne variable baseret på OPC UA tagværdier.

6. Test OPC UA Kommunikation:

   - Udfør en serie tests for at bekræfte, at OPC UA-kommunikationen fungerer korrekt.
   - Sikr, at KEPServerEX kan modtage og sende data pålideligt til og fra Rockwell PLC.

7. Dokumentér Opsætningen:

   - Dokumentér hele opsætningen og testprocessen, herunder detaljerede netværkskonfigurationer, programlistings og beskrivelser af de registrerede testresultater.

8. Udarbejd en Teknisk Rapport:

- Afslut opgaven med en teknisk rapport, der indeholder en diskussion om anvendeligheden af OPC UA i sammenhæng med industrielt automatisering, udfordringer ved implementeringen og eventuelle løsninger.

Krav til dokumentation: Rapporten skal indeholde skærmbilleder og beskrivelser af de foretagne indstillinger, PLC-programmer og logiske fløde, tests og analyser af resultater, og en evaluering af OPC UA som en kommunikationsprotokol i industrielle anvendelser.

14.2 Modbus

Konfiguration af Modbus TCP Kommunikation for Siemens PLC

Mål: Opgavens formål er at konfigurere og afprøve Modbus TCP-kommunikation mellem en Siemens PLC og KEPServerEX. Studerende vil lære at etablere netværksforbindelser, der muliggør udveksling af data mellem Siemens PLC og KEPServerEX, der understøtter Modbus TCP-protokollen.

Opgavebeskrivelse:

1. Konfigurer Siemens PLC Netværksinterface:
   - Sørg for, at Siemens PLC (f.eks. S7-1500) er korrekt forbundet til det lokale netværk med en statisk IP-adresse, der kan kommunikere med KEPServerEX serveren.
2. Aktiver Modbus TCP Server på Siemens PLC:
   - Gennem TIA Portal, aktiver Modbus TCP serveren på Siemens PLC'en og konfigurer nødvendige indstillinger (IP-adresse, portnummer).
3. Konfigurer KEPServerEX til at Forbinde med Siemens PLC:
   - Åbn KEPServerEX og tilføj en ny Modbus TCP Client Channel.
   - Indtast Siemens PLC'ens IP-adresse og portnummer.
   - Importer eller tilføj de nødvendige Modbus registre fra Siemens PLC til KEPServerEX.
4. Definer Tags i KEPServerEX:
   - Definer nødvendige tags i KEPServerEX, der skal læses fra eller skrives til Siemens PLC.

- Brug TIA Portal til at kortlægge disse tags til de respektive PLC-adresser.

5. Opret et PLC Program til Dataudveksling:
   - I TIA Portal, opret et PLC-program, der læser og skriver værdier til de definerede Modbus registre.
   - Implementér logik til periodisk at opdatere PLC'ens interne variable baseret på Modbus registerværdier.
6. Test Modbus TCP Kommunikation:
   - Udfør en serie tests for at bekræfte, at Modbus TCP-kommunikationen fungerer korrekt.
   - Sikr, at KEPServerEX kan modtage og sende data pålideligt til og fra Siemens PLC.
7. Dokumentér Opsætningen:
   - Dokumentér hele opsætningen og testprocessen, herunder detaljerede netværkskonfigurationer, programlistings og beskrivelser af de registrerede testresultater.
8. Udarbejd en Teknisk Rapport:
   - Afslut opgaven med en teknisk rapport, der indeholder en diskussion om anvendeligheden af Modbus TCP i sammenhæng med industrielt automatisering, udfordringer ved implementeringen og eventuelle løsninger.

Krav til dokumentation: Rapporten skal indeholde skærmbilleder og beskrivelser af de foretagne indstillinger, PLC-programmer og logiske fløde, tests og analyser af resultater, og en evaluering af Modbus TCP som en kommunikationsprotokol i industrielle anvendelser.

Konfiguration af Modbus TCP Kommunikation for Rockwell Automation PLC

Mål: Opgavens formål er at konfigurere og afprøve Modbus TCP-kommunikation mellem en Rockwell Automation PLC (Allen-Bradley) og KEPServerEX. Studerende vil lære at etablere netværksforbindelser, der muliggør udveksling af data mellem Rockwell PLC og KEPServerEX, der understøtter Modbus TCP-protokollen.

Opgavebeskrivelse:

1. Konfigurer Rockwell PLC Netværksinterface:

- Sørg for, at Rockwell Automation PLC (f.eks. ControlLogix) er korrekt forbundet til det lokale netværk med en statisk IP-adresse, der kan kommunikere med KEPServerEX serveren.

2. Aktiver Modbus TCP Server på Rockwell PLC:

   - Gennem Rockwell Studio 5000, aktiver Modbus TCP serveren på Rockwell PLC'en og konfigurer nødvendige indstillinger (IP-adresse, portnummer).

3. Konfigurer KEPServerEX til at Forbinde med Rockwell PLC:

   - Åbn KEPServerEX og tilføj en ny Modbus TCP Client Channel.
   - Indtast Rockwell PLC'ens IP-adresse og portnummer.
   - Importer eller tilføj de nødvendige Modbus registre fra Rockwell PLC til KEPServerEX.

4. Definer Tags i KEPServerEX:

   - Definer nødvendige tags i KEPServerEX, der skal læses fra eller skrives til Rockwell PLC.
   - Brug Rockwell Studio 5000 til at kortlægge disse tags til de respektive PLC-adresser.

5. Opret et PLC Program til Dataudveksling:

   - I Rockwell Studio 5000, opret et PLC-program, der læser og skriver værdier til de definerede Modbus registre.
   - Implementér logik til periodisk at opdatere PLC'ens interne variable baseret på Modbus registerværdier.

6. Test Modbus TCP Kommunikation:

   - Udfør en serie tests for at bekræfte, at Modbus TCP-kommunikationen fungerer korrekt.
   - Sikr, at KEPServerEX kan modtage og sende data pålideligt til og fra Rockwell PLC.

7. Dokumentér Opsætningen:

   - Dokumentér hele opsætningen og testprocessen, herunder detaljerede netværkskonfigurationer, programlistings og beskrivelser af de registrerede testresultater.

8. Udarbejd en Teknisk Rapport:

- Afslut opgaven med en teknisk rapport, der indeholder en diskussion om anvendeligheden af Modbus TCP i sammenhæng med industrielt automatisering, udfordringer ved implementeringen og eventuelle løsninger.

Krav til dokumentation: Rapporten skal indeholde skærmbilleder og beskrivelser af de foretagne indstillinger, PLC-programmer og logiske fløde, tests og analyser af resultater, og en evaluering af Modbus TCP som en kommunikationsprotokol i industrielle anvendelser.

14.3 S7-communication

Konfiguration af S7 Kommunikation for Siemens PLC

Mål: Opgavens formål er at konfigurere og afprøve S7-kommunikation mellem en Siemens PLC og KEPServerEX. Studerende vil lære at etablere netværksforbindelser, der muliggør udveksling af data mellem Siemens PLC og KEPServerEX, der understøtter S7-protokollen.

Opgavebeskrivelse:

1. Konfigurer Siemens PLC Netværksinterface:
   - Sørg for, at Siemens PLC (f.eks. S7-1200 eller S7-1500) er korrekt forbundet til det lokale netværk med en statisk IP-adresse, der kan kommunikere med KEPServerEX serveren.
2. Konfigurer S7-Forbindelse på Siemens PLC:
   - Gennem TIA Portal, opret og konfigurer en ny S7-forbindelse på Siemens PLC'en og angiv nødvendige indstillinger (IP-adresse, rack og slot).
3. Konfigurer KEPServerEX til at Forbinde med Siemens PLC:
   - Åbn KEPServerEX og tilføj en ny Siemens S7-200/300/400/1200/1500 Ethernet driver.
   - Indtast Siemens PLC'ens IP-adresse, rack og slotnummer.
   - Importer eller tilføj de nødvendige dataområder (DB, M, I, Q) fra Siemens PLC til KEPServerEX.
4. Definer Tags i KEPServerEX:
   - Definer nødvendige tags i KEPServerEX, der skal læses fra eller skrives til Siemens PLC.

- Brug TIA Portal til at kortlægge disse tags til de respektive PLC-adresser.

5. Opret et PLC Program til Dataudveksling:
   - I TIA Portal, opret et PLC-program, der læser og skriver værdier til de definerede dataområder.
   - Implementér logik til periodisk at opdatere PLC'ens interne variable baseret på S7 registerværdier.
6. Test S7 Kommunikation:
   - Udfør en serie tests for at bekræfte, at S7-kommunikationen fungerer korrekt.
   - Sikr, at KEPServerEX kan modtage og sende data pålideligt til og fra Siemens PLC.
7. Dokumentér Opsætningen:
   - Dokumentér hele opsætningen og testprocessen, herunder detaljerede netværkskonfigurationer, programlistings og beskrivelser af de registrerede testresultater.
8. Udarbejd en Teknisk Rapport:
   - Afslut opgaven med en teknisk rapport, der indeholder en diskussion om anvendeligheden af S7-kommunikation i sammenhæng med industrielt automatisering, udfordringer ved implementeringen og eventuelle løsninger.

Krav til dokumentation: Rapporten skal indeholde skærmbilleder og beskrivelser af de foretagne indstillinger, PLC-programmer og logiske fløde, tests og analyser af resultater, og en evaluering af S7 som en kommunikationsprotokol i industrielle anvendelser.

14.4 Ethernet/IP

Konfiguration af Ethernet/IP Kommunikation for Rockwell Automation PLC

Mål: Opgavens formål er at konfigurere og afprøve Ethernet/IP-kommunikation mellem en Rockwell Automation PLC (f.eks. Allen-Bradley) og KEPServerEX. Studerende vil lære at etablere netværksforbindelser, der muliggør udveksling af data mellem Rockwell Automation PLC og KEPServerEX, der understøtter Ethernet/IP-protokollen.

Opgavebeskrivelse:

1. Konfigurer Rockwell Automation PLC Netværksinterface:
    - Sørg for, at Rockwell Automation PLC (f.eks. Allen-Bradley ControlLogix eller CompactLogix) er korrekt forbundet til det lokale netværk med en statisk IP-adresse, der kan kommunikere med KEPServerEX serveren.
2. Konfigurer Ethernet/IP Forbindelse på Rockwell Automation PLC:
    - Gennem RSLogix 5000/Studio 5000, opret og konfigurer en ny Ethernet/IP-forbindelse på Rockwell Automation PLC'en og angiv nødvendige indstillinger (IP-adresse).
3. Konfigurer KEPServerEX til at Forbinde med Rockwell Automation PLC:
    - Åbn KEPServerEX og tilføj en ny Ethernet/IP driver.
    - Indtast Rockwell Automation PLC'ens IP-adresse.
    - Importer eller tilføj de nødvendige dataområder (tags) fra Rockwell Automation PLC til KEPServerEX.
4. Definer Tags i KEPServerEX:
    - Definer nødvendige tags i KEPServerEX, der skal læses fra eller skrives til Rockwell Automation PLC.
    - Brug RSLogix 5000/Studio 5000 til at kortlægge disse tags til de respektive PLC-adresser.
5. Opret et PLC Program til Dataudveksling:
    - I RSLogix 5000/Studio 5000, opret et PLC-program, der læser og skriver værdier til de definerede dataområder.
    - Implementér logik til periodisk at opdatere PLC'ens interne variable baseret på Ethernet/IP registerværdier.
6. Test Ethernet/IP Kommunikation:
    - Udfør en serie tests for at bekræfte, at Ethernet/IP-kommunikationen fungerer korrekt.
    - Sikr, at KEPServerEX kan modtage og sende data pålideligt til og fra Rockwell Automation PLC.
7. Dokumentér Opsætningen:
    - Dokumentér hele opsætningen og testprocessen, herunder detaljerede netværkskonfigurationer, programlistings og beskrivelser af de registrerede testresultater.

8. Udarbejd en Teknisk Rapport:

    - Afslut opgaven med en teknisk rapport, der indeholder en diskussion om anvendeligheden af Ethernet/IP-kommunikation i sammenhæng med industrielt automatisering, udfordringer ved implementeringen og eventuelle løsninger.

Krav til dokumentation: Rapporten skal indeholde skærmbilleder og beskrivelser af de foretagne indstillinger, PLC-programmer og logiske fløde, tests og analyser af resultater, og en evaluering af Ethernet/IP som en kommunikationsprotokol i industrielle anvendelser.

14.5 MQTT

Konfiguration af MQTT Kommunikation for Siemens PLC

Mål: Opgavens formål er at konfigurere og afprøve MQTT-kommunikation mellem en Siemens PLC (f.eks. S7-1200) og KEPServerEX. Studerende vil lære at etablere netværksforbindelser, der muliggør udveksling af data mellem Siemens PLC og KEPServerEX, der understøtter MQTT-protokollen.

Opgavebeskrivelse:

1. Konfigurer Siemens PLC Netværksinterface:

    - Sørg for, at Siemens PLC er korrekt forbundet til det lokale netværk med en statisk IP-adresse, der kan kommunikere med KEPServerEX serveren.

2. Konfigurer MQTT Forbindelse på Siemens PLC:

    - Gennem TIA Portal, opret og konfigurer en ny MQTT-klient på Siemens PLC'en og angiv nødvendige indstillinger (MQTT broker IP-adresse, portnummer, etc.).

3. Konfigurer KEPServerEX til at Forbinde med Siemens PLC:

    - Åbn KEPServerEX og tilføj en ny MQTT driver.
    - Indtast Siemens PLC'ens MQTT broker IP-adresse.
    - Importer eller tilføj de nødvendige dataområder (tags) fra Siemens PLC til KEPServerEX.

4. Definer Tags i KEPServerEX:

    - Definer nødvendige tags i KEPServerEX, der skal læses fra eller skrives til Siemens PLC.

- Brug TIA Portal til at kortlægge disse tags til de respektive PLC-adresser.

5. Opret et PLC Program til Dataudveksling:
   - I TIA Portal, opret et PLC-program, der læser og skriver værdier til de definerede dataområder.
   - Implementér logik til periodisk at opdatere PLC'ens interne variable baseret på MQTT registerværdier.
6. Test MQTT Kommunikation:
   - Udfør en serie tests for at bekræfte, at MQTT-kommunikationen fungerer korrekt.
   - Sikr, at KEPServerEX kan modtage og sende data pålideligt til og fra Siemens PLC.
7. Dokumentér Opsætningen:
   - Dokumentér hele opsætningen og testprocessen, herunder detaljerede netværkskonfigurationer, programlistings og beskrivelser af de registrerede testresultater.
8. Udarbejd en Teknisk Rapport:
   - Afslut opgaven med en teknisk rapport, der indeholder en diskussion om anvendeligheden af MQTT-kommunikation i sammenhæng med industrielt automatisering, udfordringer ved implementeringen og eventuelle løsninger.

Krav til dokumentation: Rapporten skal indeholde skærmbilleder og beskrivelser af de foretagne indstillinger, PLC-programmer og logiske fløde, tests og analyser af resultater, og en evaluering af MQTT som en kommunikationsprotokol i industrielle anvendelser.

Konfiguration af MQTT Kommunikation for Rockwell Automation PLC

Mål: Opgavens formål er at konfigurere og afprøve MQTT-kommunikation mellem en Rockwell Automation PLC (f.eks. Allen-Bradley) og KEPServerEX. Studerende vil lære at etablere netværksforbindelser, der muliggør udveksling af data mellem Rockwell Automation PLC og KEPServerEX, der understøtter MQTT-protokollen.

Opgavebeskrivelse:

1. Konfigurer Rockwell Automation PLC Netværksinterface:

- Sørg for, at Rockwell Automation PLC (f.eks. Allen-Bradley ControlLogix eller CompactLogix) er korrekt forbundet til det lokale netværk med en statisk IP-adresse, der kan kommunikere med KEPServerEX serveren.

2. Konfigurer MQTT Forbindelse på Rockwell Automation PLC:
   - Gennem RSLogix 5000/Studio 5000, opret og konfigurer en ny MQTT-klient på Rockwell Automation PLC'en og angiv nødvendige indstillinger (MQTT broker IP-adresse, portnummer, etc.).
3. Konfigurer KEPServerEX til at Forbinde med Rockwell Automation PLC:
   - Åbn KEPServerEX og tilføj en ny MQTT driver.
   - Indtast Rockwell Automation PLC'ens MQTT broker IP-adresse.
   - Importer eller tilføj de nødvendige dataområder (tags) fra Rockwell Automation PLC til KEPServerEX.
4. Definer Tags i KEPServerEX:
   - Definer nødvendige tags i KEPServerEX, der skal læses fra eller skrives til Rockwell Automation PLC.
   - Brug RSLogix 5000/Studio 5000 til at kortlægge disse tags til de respektive PLC-adresser.
5. Opret et PLC Program til Dataudveksling:
   - I RSLogix 5000/Studio 5000, opret et PLC-program, der læser og skriver værdier til de definerede dataområder.
   - Implementér logik til periodisk at opdatere PLC'ens interne variable baseret på MQTT registerværdier.
6. Test MQTT Kommunikation:
   - Udfør en serie tests for at bekræfte, at MQTT-kommunikationen fungerer korrekt.
   - Sikr, at KEPServerEX kan modtage og sende data pålideligt til og fra Rockwell Automation PLC.
7. Dokumentér Opsætningen:
   - Dokumentér hele opsætningen og testprocessen, herunder detaljerede netværkskonfigurationer, programlistings og beskrivelser af de registrerede testresultater.
8. Udarbejd en Teknisk Rapport:

- Afslut opgaven med en teknisk rapport, der indeholder en diskussion om anvendeligheden af MQTT-kommunikation i sammenhæng med industrielt automatisering, udfordringer ved implementeringen og eventuelle løsninger.

Krav til dokumentation: Rapporten skal indeholde skærmbilleder og beskrivelser af de foretagne indstillinger, PLC-programmer og logiske fløde, tests og analyser af resultater, og en evaluering af MQTT som en kommunikationsprotokol i industrielle anvendelser.

Konfiguration af MQTT Kommunikation for ESP32

Mål: Opgavens formål er at konfigurere og afprøve MQTT-kommunikation mellem en ESP32 mikrokontroller og KEPServerEX. Studerende vil lære at etablere netværksforbindelser, der muliggør udveksling af data mellem ESP32 og KEPServerEX, der understøtter MQTT-protokollen.

Opgavebeskrivelse:

1. Konfigurer ESP32 Netværksinterface:
   - Sørg for, at ESP32 er korrekt forbundet til det lokale netværk med en statisk IP-adresse, der kan kommunikere med KEPServerEX serveren.
2. Konfigurer MQTT Forbindelse på ESP32:
   - Gennem Arduino IDE eller ESP-IDF, opret og konfigurer en ny MQTT-klient på ESP32 og angiv nødvendige indstillinger (MQTT broker IP-adresse, portnummer, etc.).
3. Konfigurer KEPServerEX til at Forbinde med ESP32:
   - Åbn KEPServerEX og tilføj en ny MQTT driver.
   - Indtast ESP32'ens MQTT broker IP-adresse.
   - Importer eller tilføj de nødvendige dataområder (tags) fra ESP32 til KEPServerEX.
4. Definer Tags i KEPServerEX:
   - Definer nødvendige tags i KEPServerEX, der skal læses fra eller skrives til ESP32.
   - Brug Arduino IDE eller ESP-IDF til at kortlægge disse tags til de respektive MQTT-emner.
5. Opret et Program til Dataudveksling på ESP32:

- I Arduino IDE eller ESP-IDF, opret et program, der læser og skriver værdier til de definerede dataområder.
- Implementér logik til periodisk at opdatere ESP32'ens interne variable baseret på MQTT registerværdier.

6. Test MQTT Kommunikation:
   - Udfør en serie tests for at bekræfte, at MQTT-kommunikationen fungerer korrekt.
   - Sikr, at KEPServerEX kan modtage og sende data pålideligt til og fra ESP32.
7. Dokumentér Opsætningen:
   - Dokumentér hele opsætningen og testprocessen, herunder detaljerede netværkskonfigurationer, programlistings og beskrivelser af de registrerede testresultater.
8. Udarbejd en Teknisk Rapport:
   - Afslut opgaven med en teknisk rapport, der indeholder en diskussion om anvendeligheden af MQTT-kommunikation i sammenhæng med IoT-applikationer, udfordringer ved implementeringen og eventuelle løsninger.

Krav til dokumentation: Rapporten skal indeholde skærmbilleder og beskrivelser af de foretagne indstillinger, programmer og logiske fløde, tests og analyser af resultater, og en evaluering af MQTT som en kommunikationsprotokol i IoT anvendelser.

Kapitel 15

Universal Robots

15.1 Modbus Universal Robots til Siemens

Mål: Formålet med denne opgave er at konfigurere og implementere en Modbus TCP-kommunikation mellem en Universal Robot og en simuleret SIMATIC S7-1500 PLC ved hjælp af Siemens PLCSIM Advanced. Denne opgave vil give de studerende erfaring med at integrere forskellige automatiseringssystemer og teknologier.

Opgavebeskrivelse:

1. Konfigurer netværksindstillingerne for den simulerede S7-1500 PLC i TIA Portal, og sørg for, at den kan nås på det virtuelle netværk.
2. Etabler en Modbus TCP-server på den simulerede PLC ved at benytte de integrerede funktioner til Modbus-kommunikation i TIA Portal. Definér relevante dataregistre, som skal være tilgængelige for UR-robotten.
3. På UR-robottens kontrolpanel, opsæt en Modbus TCP-klient, og konfigurér den til at forbinde til den simulerede S7-1500 PLC's IP-adresse og port.
4. Skab et URScript eller brug den grafiske brugergrænseflade på robottens kontrolpanel til at programmere robotten til at sende Modbus-forespørgsler for at læse fra og skrive til dataregistrene på den simulerede PLC.
5. Simuler begge systemer, og initier dataudveksling for at sikre, at UR-robotten og den simulerede S7-1500 PLC kan kommunikere korrekt over Modbus TCP.
6. Test kommunikationen grundigt, og bekræft, at UR-robotten kan hente og ændre værdierne i den simulerede PLC's datablokke effektivt.

7. Analyser datastrømmen og sikkerheden i kommunikationen, og diskutér, hvordan denne integration kan optimeres i en reel industriel applikation.

8. Dokumentér processen og resultaterne i en teknisk rapport, der indeholder netværkskonfigurationer, programmeringsdetaljer, og testscenarier.

Krav til dokumentation: Rapporten skal indeholde skærmbilleder, kodeudsnit, konfigurationsindstillinger, testresultater, og en kritisk analyse af Modbus-kommunikationens præstation og pålidelighed i integrationen mellem Universal Robots og Siemens PLC-systemer.

15.2 Modbus Universal Robots til Rockwell

Mål: Formålet med denne opgave er at konfigurere og demonstrere Modbus TCP-kommunikation mellem en Universal Robots (UR) robotarm og en Allen-Bradley PLC. Studerende vil få praktisk erfaring med at integrere en UR-robot med en Rockwell PLC ved hjælp af Modbus TCP-protokollen. Denne opgave kan udføres enten med en fysisk robot eller en simuleret robot.

Opgavebeskrivelse:

1. **Opsætning af Modbus TCP på Universal Robots:**

   - Konfigurer Modbus TCP på UR-robotten og opsæt nødvendige parametre som IP-adresse, portnummer, og Modbus-registere.
   - Definér Modbus-coils og registre, der vil blive brugt til at kommunikere med Allen-Bradley PLC'en.

2. **Konfiguration af Allen-Bradley PLC:**

   - I Studio 5000, opsæt et nyt projekt og konfigurer PLC'en til at fungere som en Modbus TCP Master, som kan læse fra og skrive til UR-robotten.
   - Opret en Modbus-tabel i Studio 5000, der matcher de coils og registre, der er konfigureret på UR-robotten.

3. **Simulering og Test:**

   - Opret et simpelt program i Studio 5000, hvor PLC'en kontrollerer UR-robotten via Modbus TCP, for eksempel ved at sende en kommando om at starte robotten, når en specifik coil aktiveres.
   - Implementér en funktion, hvor PLC'en kan overvåge robotarmens position og status via Modbus-registre og reagere på ændringer.

- Test kommunikationen ved at simulere forskellige scenarier, såsom start og stop af robotten eller læsning af sensorværdier fra robotten.

4. **UR Robot Handling:**
   - Programmér UR-robotten til at flytte en kasse fra en position til en anden, når den modtager et signal fra Allen-Bradley PLC'en via Modbus TCP.
   - Robotten skal reagere på Modbus-signaler for at bevæge sig til kassen, samle den op, transportere den til en palle, og placere den præcist.
   - Når robotten har fuldført opgaven, skal den sende en færdigmelding tilbage til PLC'en.
5. **Dokumentation:**
   - Dokumentér hele konfigurationsprocessen for både UR-robotten og Allen-Bradley PLC'en, inklusive Modbus TCP indstillingerne.
   - Inkludér diagrammer, der illustrerer dataflowet mellem PLC'en og UR-robotten, samt screenshots af relevante skærmbilleder fra Studio 5000.
   - Beskriv testscenarierne og de observerede resultater, herunder en analyse af systemets ydeevne og eventuelle kommunikationsfejl.

Krav til dokumentation: Den tekniske rapport skal indeholde:

- En detaljeret beskrivelse af konfigurationen af Modbus TCP både på UR-robotten og i Studio 5000.
- Et netværksdiagram, der viser kommunikationsforbindelsen mellem UR-robotten og Allen-Bradley PLC'en.
- En gennemgang af de observerede resultater fra simuleringen, inklusive eventuelle udfordringer og løsninger.

15.3 Modbus Universal Robots til KepServerEX

Mål: Formålet med denne opgave er at konfigurere en UR-robot til at bevæge sig fra en standby-position, hente en kasse, der er registreret af en sensor, og placere den på en palle. Når robotten har afsluttet sin opgave, skal den sende et færdigsignal tilbage til KEPServerEX via Modbus. Denne øvelse giver studerende praktisk erfaring med anvendelsen af Modbus til styring af robotbevægelser i en industriel kontekst. Opgaven kan udføres enten med en fysisk UR-robot eller ved hjælp af en simuleret robot.

Opgavebeskrivelse:

1. **Opsætning af KEPServerEX:**

    - Installér og konfigurer KEPServerEX med en Modbus TCP/IP driver.
    - Opret coils i KEPServerEX til at sende signaler om, hvornår robotten skal bevæge sig fra standby-position til kassen, når kassen rammer sensoren.
    - Opret holding registers til at modtage statusopdateringer fra robotten, såsom bekræftelse af, at kassen er placeret på pallen.

2. **UR-robot Konfiguration:**

    - Opret et URScript-program, der forbinder til KEPServerEX via Modbus.
    - Programmér robotten til at bevæge sig til en standby-position, klar til at modtage signal fra KEPServerEX.
    - Programmér robotten til at bevæge sig mod kassen, når sensoren aktiveres, og afhent kassen.
    - Robotten skal derefter transportere kassen til pallen og placere den på en foruddefineret position.
    - Tilføj logik, der sender et færdigsignal til KEPServerEX, når kassen er korrekt placeret på pallen.

3. **Test og Simulering:**

    - Simulér en situation, hvor KEPServerEX sender et signal til robotten om at hente en kasse, når sensoren er aktiveret.
    - Observér robotens bevægelser fra standby-position til kassen, og bekræft, at kassen korrekt transporteres og placeres på pallen.
    - Verificér, at robotten sender et færdigsignal tilbage til KEPServerEX, når opgaven er udført.

4. **Fejlhåndtering:**

    - Implementér fejlhåndtering i URScript, der kan håndtere tilfælde, hvor kassen ikke er korrekt placeret, eller forbindelsen til KEPServerEX mistes.
    - Test fejlhåndteringsprocedurerne og evaluer robotens respons på fejltilstande.

Krav til dokumentation: Rapporten skal indeholde:

- En beskrivelse af KEPServerEX og UR-robotopsætningen.

- Diagrammer, der viser robotens bevægelsesbane fra standby-position til kasse og derefter til pallen.
- Skærmbilleder af konfigurationer, dataflow og testresultater.
- En analyse af effektiviteten af robotens opgaver og fejlhåndteringsstrategier.

Kapitel 16

ABB Robot

16.1 Opgave: Konfiguration af Netværkskommunikation for ABB Roboter

16.2 Konklusion

Mål: Målet med denne opgave er at konfigurere netværkskommunikation for ABB robotter, som kan bruges til at interagere med andre enheder i et industrielt netværk, såsom PLC'er, HMI'er og fjernstyrede servere. Studerende skal lære at etablere og diagnosticere netværksforbindelser, konfigurere IP-indstillinger, og oprette dataudveksling over industrielle protokoller.

Opgavebeskrivelse:

1. **Netværkskonfiguration:** Konfigurer netværksindstillingerne på ABB robotcontrolleren, herunder IP-adresse, subnetmaske og gateway for at forbinde robotten til et lokalt netværk.

2. **Protokolimplementering:** Vælg en industriprotokol såsom Modbus TCP, Ethernet/IP eller PROFINET. Opsæt de nødvendige parametre for den valgte protokol på robotcontrolleren.

3. **PLC-Integration:** Etablere en forbindelse mellem ABB robotcontrolleren og en PLC. Dette kan inkludere opsætning af en passende PLC-modul for kommunikation og konfiguration af korrekt dataudveksling.

4. **Dataudveksling:** Implementer et simpelt kontrolsystem, hvor ABB robotten modtager kommandoer fra og sender statusopdateringer til PLC'en. Dette kan omfatte, men er ikke begrænset til, start/stop af robotprogrammer, nødstopsignaler og produktionstællinger.

5. **Fejlfinding og Diagnostik:** Udfør netværksdiagnostik ved hjælp af ABB's værktøjer og/eller tredjepartsværktøjer for at sikre en pålidelig

dataudveksling. Identificer og ret eventuelle konfigurationsproblemer.

6. **Sikkerhedsvurdering:** Analyser netværkssikkerheden for din robotforbindelse og anbefal forbedringer eller best practices for at sikre industrielle netværk.

7. **Rapportering og Dokumentation:** Dokumentér hele processen fra start til slut, inklusive netværkskonfigurationer, programmeringskode, fejlfindingstrin og sikkerhedsanalyse. Indsend en rapport med detaljerede forklaringer og underbygget med relevante skærmbilleder og diagrammer.

Krav til dokumentation: Rapporten skal indeholde omfattende teknisk dokumentation af alle skridt taget i konfigurationsprocessen, problemløsning, sikkerhedsvurdering og eventuelle anbefalinger til yderligere arbejde eller undersøgelser.