

Implementering av CANopen i styrsystem

Joel Bjons

Examensarbete för ingenjörsexamen (YH)

Utbildningsprogrammet för el- och automationsteknik

Vasa 2024

EXAMENSARBETE

Författare: Joel Bjons

Utbildning och ort: El- och automationsteknik

Inriktning: Automationsteknik

Handledare: Hans Lindén

Titel: Implementering av CANopen i styrsystem

Datum: 12.05.2024

Sidantal: 23

Abstrakt

Detta examensarbete har utförts på uppdrag av företaget NoSwing Oy. Arbetet gick ut på att analysera CAN och CANopen-protokollen både teoretiskt och praktiskt för att sedan integrera en CANopen-modul i en färdig traverskranstyrning. Syftet med arbetet var att ge fler komponentvalmöjligheter vid nybyggnation och renovering för framtida projekt vilket gynnar både montörer och kunder.

Arbetet innehåller en genomgång av CAN-protokollet och en djupdykning i CANopen som är en påbyggnation av CAN för att få en enklare meddelande hantering.

Examensarbetets teoretiska del presenterar, med hjälp av akademiska dokument och webbsidor, protokollets funktioner och användningssätt. Dessutom ingår en genomgång av vilken hårdvara och mjukvara som kommer användas för att möjliggöra processen.

Arbetets praktiska del beskriver uppbyggnationen av CAN-nätverket och hur det används för projektets utförande.

Slutresultatet blev ett fungerande funktionsblock i Siemens TIA-portal som lätt går att integrera in i färdiga projekt.

Språk: svenska

Nyckelord: CAN, CANopen, PLC

OPINNÄYTETYÖ

Tekijä: Joel Bjons

Koulutus ja paikkakunta: Sähkö- ja automaatiotekniikka, Vaasa

Suuntautumisvaihtoehto: Automaatiotekniikka

Ohjaaja(t): Hans Lindén

Nimike: CANopenin käyttöönotto ohjausjärjestelmissä

Päivämäärä: 12.06.2024

Sivumäärä: 23

Tiivistelmä

Tämä opinnäytetyö on tehty NoSwing Oy:n toimeksiannosta. Työ käsitti CAN- ja CANopen-protokollien analysoinnin sekä teoreettisesti että käytännöllisesti ja sen jälkeen CANopen-moduulin integroinnin valmiiseen siltanosturin ohjausjärjestelmään. Työn tarkoituksena oli tarjota lisää komponenttivalintoja uudis- ja korjausrakentamisessa tulevaisuuden projekteihin, mistä hyötyvät sekä asentajat että asiakkaat.

Työssä tarkastellaan CAN-protokollaa ja perehdytään CANopeniin, joka on CAN:n rakenne yksinkertaisemman viestinkäsittelyn aikaansaamiseksi.

Työn teoriaosuudessa esitellään akateemisten dokumenttien ja verkkosivujen avulla protokollan toimintoja ja käyttötarkoituksia. Mukana on myös tarkastelu laitteistosta ja ohjelmistosta, joita käytetään prosessin käyttöönottoon.

Työn käytännön osuudessa kuvataan CAN-verkon rakentamista ja sen käyttöä projektin toteuttamisessa

Lopputuloksena oli toimiva toimilohko Siemens TIA-portaalissa, joka on helppo integroida valmiisiin projekteihin.

Kieli: Ruotsi

Avainsanat: CAN, CANopen, PLC

BACHELOR'S THESIS

Author: Joel Bjons

Degree Programme: Electrical and Automation Engineering

Specialisation: Automation Engineering

Supervisor(s): Hans Lindén

Title: Implementation of CANopen in Control Systems

Date: 12.06.2024

Number of pages: 23

Abstract

This thesis has been carried out on behalf of the company NoSwing Oy. The work consisted of analysing CAN and the CANopen protocols both theoretically and practically and then integrating a CANopen module into a finished overhead crane control system. Providing more options for components in new construction and renovation for future projects, which benefits both technicians and customers.

The work contains a presentation of the CAN protocol and a deep dive into CANopen, which is an extension of CAN to get simpler message handling.

The theoretical part of the thesis is presented with the help of academic documents and web pages which contain information on the functions and uses of the protocol. Furthermore, a presentation of the hardware and software that will be used to achieve the process is included.

The practical part of the project describes the construction of the CAN network and how it is used for the execution of the project.

The result was a working function block in Siemens TIA portal that can easily be integrated into finished projects.

Language: Swedish

Key words: CAN, CANopen, PLC

Innehållsförteckning

1	Inledning	1
1.1	Uppdragsgivare	1
1.2	Uppgiftsbeskrivning och syfte	1
2	Teori.....	2
2.1	CAN	2
2.1.1	Fysiska skiktet.....	2
2.1.2	Kommunikation.....	3
2.1.3	Flexible Data Rate	4
2.1.4	CAN in Automation.....	4
2.2	CANopen.....	4
2.2.1	Kommunikation.....	5
2.2.2	CAN-ID /COB-ID	6
2.2.3	Object Dictionary	7
2.2.4	Synchronization Protocol.....	7
2.2.5	Time-stamp Protocol.....	7
2.2.6	Emergency Protocol.....	8
2.2.7	Process Data Object	8
2.2.8	Service Data Object	9
2.2.9	Heartbeat.....	10
2.2.10	Network management.....	11
2.3	CANopen-protokollets övriga enhetsprofiler.....	12
2.3.1	CANopen Battery.....	12
2.3.2	CANopen Medical.....	12
2.3.3	CANopen Safety	12
2.4	Hårdvara.....	13
2.4.1	Siemens ET200 SP	13
2.4.2	Siemens CM CAN	13
2.4.3	HBC-Radiomatic	14
2.4.4	Kvaser U100.....	14
2.5	Mjukvara.....	14
2.5.1	TIA portal	14
3	Praktiskt utförande	15
3.1	CAN-nätverket	15
3.2	Övervakning av trafiken	16
3.3	Utveckling av programvara.....	17
4	Resultat.....	20
5	Diskussion	21

6	Källförteckning.....	22
---	----------------------	----

1 Inledning

I det första kapitlet presenteras både uppdragsgivaren och uppgiften med syfte.

1.1 Uppdragsgivare

NoSwing Oy AB är ett automatiseringsföretag med huvudkontor i Jakobstad, Företaget grundades 2015 och hade då endast försäljning och installering av NoSwing sensorer som huvudsaklig verksamhet. Succesivt har verksamheten växt och i dagsläget hanterar företaget kompletta automatiseringspaket av traverskranar och diverse industrimaskiner.

1.2 Uppgiftsbeskrivning och syfte

Målet med slutarbetet var att utforska och analysera både CAN och i huvudsak CANopen-protokollet både teoretiskt och praktiskt. Den praktiska delen för detta arbete går igenom idriftsättningen av en CANopen-radiomodul av tillverkaren HBC-radiomatic tillsammans med en Siemens kommunicerings modul av version CM CAN.

Det praktiska syftet med arbetet är att ge fler komponentvalmöjligheter och flexibilitet vid nybyggnation och renovering för framtida projekt eftersom en ny marknad av komponenter blir användbara.

Önskat resultat med arbetet var att få fram en passlig lösning i form av ett TIA-portal funktionsblock men även en bra genomgång och förklaring av flera CANopen-termer och funktioner.

2 Teori

Detta kapitel förklarar allmänt vad CAN-protokollet är och hur det fungerar, därefter kommer en djupdykning in i själva CANopen-protokollet och dess funktioner. I kapitlet ingår även övrig hård- och mjukvara genomgång som används i den praktiska delen.

2.1 CAN

Controller Area Network eller CANbus introducerades av Robert Bosch i Detroit år 1986. Följande år var Intel först med att tillverka en CAN-kontroller som då använde protokollversionen CAN 1.0. I dagsläget finns ett CAN-nätverk i största delen av alla nytillverkade bilar samt även allt från fartyg till flygplan. Fastän protokollet var skapat för fordon så har det använts brett i den industriella automationen, detta gör att CAN protokollet under de senaste 35 åren varit det överlägset mest använda nätverksprotokollet.

Utvecklingen av CAN-protokollet grundade sig på idén att försöka minska kabelantalet i bilar och på så sätt skapa en mer simpel och effektiv lösning, för att få detta utfört används flera mikrokontrollers eller noder som kan kommunicera mellan varandra med en simpel dataöverföring. Eftersom systemet blir uppbyggt av flera sektioner underlättar det också vid underhållsbehov och minskar kabel antalet avsevärt. (Pfeiffer, Ayre, & Keydel; Kvaser).

2.1.1 Fysiska skiktet

Det fysiska skiktet på ett CAN-system består av tre ledningar med en 120 Ohms terminering i både början och slutet av dataledningarna, se Figur 1. Ledningarna kallas CAN-H vilket indikerar CAN-Hög signal och CAN-L indikerar CAN-låg signal och CAN-GND indikerar Jord. Spänningen mellan CAN H och CAN L är noll när en logisk nolla skrivs på bussen och en logisk etta representeras med en spännings skillnad på två volt. Eftersom CAN-Kommunikationen är känslig för externa magnetiska fält används alltid twistade par kablar i praktiken för att motverka eventuella störningar. (Pfeiffer, Ayre, & Keydel).

2.1.3 Flexible Data Rate

En vidare utveckling av CAN-protokollet är CAN-FD, Flexible Data Rate, skillnaden mellan dessa är i huvudsak dataöverföringshastigheten och mängden meddelanden de klarar av att hantera. CAN-FD-protokollet klarar av att kommunicera med en högre dataöverföringshastighet på 8Mbps motsvarande standard variantens 1Mbps, Meddelande storleken har också gått upp till 64 byte motsvarande standard variantens 8 byte, vilket avsevärt höjer prestandan men även trafiken på nätverket. (Lennartsson, 2019).

2.1.4 CAN in Automation

CiA är en organisation som skapades av företag och användare för att utveckla standarder för CAN-protokollet över alla sju OSI lager. Det är en global organisation som kontinuerligt växer, i dagsläget medverkar 750 olika företag varav 11 är baserade och verksamma i Finland. (CAN in Automation CiA, 2023).

Deras mest uppmärksammade bedrift är utvecklingen av CAN-Application Layer vilket utgör ett applikationslager påbyggt på standard CAN-protokollet. De vanligaste applikationslager-protokollen är J1939, OBD2 och CANopen. Enda skillnaden mellan dessa protokoll är hur meddelanden uppfattas och används. Skillnaden beror på att deras användningsområden är olika. J1939 används i störst utsträckning till tunga fordon såsom bussar och långtradare. OBD2 står för On-Board-Diagnostics och används för diagnostiken på de flesta personbilar i trafik och CANopen används oftast inom industrin till alla möjliga robotar och övriga automationsmaskiner. (CAN in Automation CiA, 2023).

2.2 CANopen

CANopen-protokollet är en kombination av flera enhetsprofiler där CiA-301 är originalet och grundstandarderna som alla andra byggs ifrån, en enhetsprofil består av flera parametrar som används för olika ändamål, till exempel CiA-401 som beskriver vilka index alla I/O använder eller CiA-402 som beskriver index för drivdon och frekvensomvandlare.

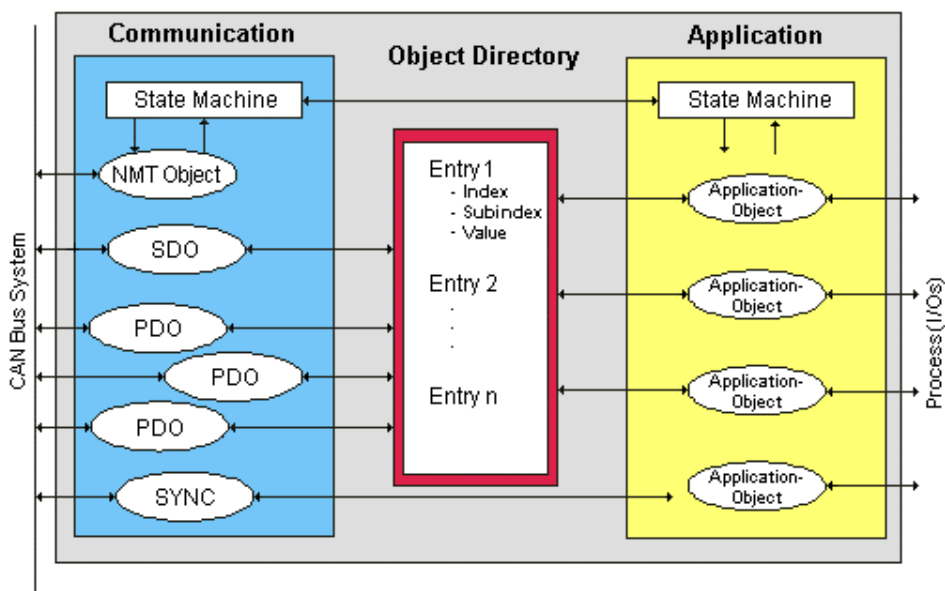
CANopen-protokollet verkar i huvudsakligen i applikationslagret det vill säga det sjunde lagret av OSI-modellen, fastän vissa funktioner ingår i de övriga OSI-lagren till exempel delar av CiA-301 och CiA-302, se Figur 3.

Application level		User program(s)
Data level		CiA 4XX: Device and application profiles
OSI layers	Application layer	CiA 301: NMT, Heartbeat, SDO, PDO, SYNC, EMCY, TIME
	Presentation layer	CiA 301: Data types and encoding rules
	Session layer	Not applicable
	Transport layer	CiA 301: Segmented SDO
	Network layer	(CiA 302-7: SDO and EMCY routing, PDO bridging)*
	Data link layer	ISO 11898-1
	Physical layer	ISO 11898-2, CiA 301 (bit-timing), CiA 303-1 (cable and connectors)

Figur 3: CANopen:s olika enhetsprofiler över OSI modellen.

2.2.1 Kommunikation

CANopen-protokollet består av flera standardiserade meddelanden och kategoriseras utifrån deras användning och typ i Object dictionary. Kommunikationen mellan noderna är standardiserade och därför bör en CANopen-modul fungera enligt ett färdigt bestämt ramverk för att innehålla de rätta protokollen och funktionerna, se Figur 4



Figur 4: Överbild av en standard CANopen-moduls insida och kommunikeringsprocesser (Beckhoff New Automation Technology, 2020).

Huvudsakligen används CANopen-protokollet i kommunikationsnätverket Master-Slave de vill säga endast en modul utses som Masternod, därefter blir modulens uppgift att skicka ut förfrågningar på nätverket och därefter tolka och använda svaren från Slavenoderna. De övriga modulerna i nätverket blir till slavmoduler vars uppgift endast blir att svara på masterns förfrågningar.

Client-Server är ett annat slags kommunikationsnätverk som används i CANopen-protokollet, det utgår från att en klient kommunicerar med en servermodul, klienten initierar med att skicka data från ett specifikt index på servermodulen och därefter svarar servermodulen med innehållet från det specifika indexet.

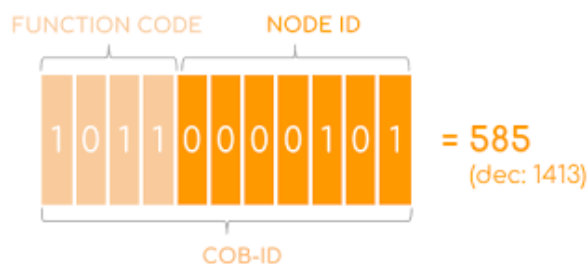
Den tredje kommunikationsnätverket som används är Producer-Consumer det utgår från att en Producer skickar ut data på nätverket som tas emot av en eller flera konsumenter (CAN in Automation CiA, 2011) (Pfeiffer, Ayre, & Keydel).

2.2.2 CAN-ID /COB-ID

I ett CANopen-nätverk består alla meddelanden av standardiserade CAN-ID: n utgående från vilken modul som meddelandet härstammar och vilken typ den har. CAN-ID utgör identifieraren för ett meddelande vilket i normala nätverk utgörs av maximalt 2^{11} (2048) unika ID:n. Denna typ av struktur delar på meddelandet i två viktiga delar:

4-bitars funktionskod: Denna kod definierar själva meddelandet som överförs och vilken typ den har, till exempel temperaturdata överföring, starta motorkörning eller skicka felmeddelanden.

7-bitars Nod ID: Detta unika ID är individuellt för alla enheter i CANopen-nätverket. Det möjliggör att kommunikationen endast utförs mellan avsedda mottagaren och utsändaren. Eftersom Nod id endast utgör 7-bitar begränsar nätverket till maximalt 127 noder i ett nätverk, se Figur 5.



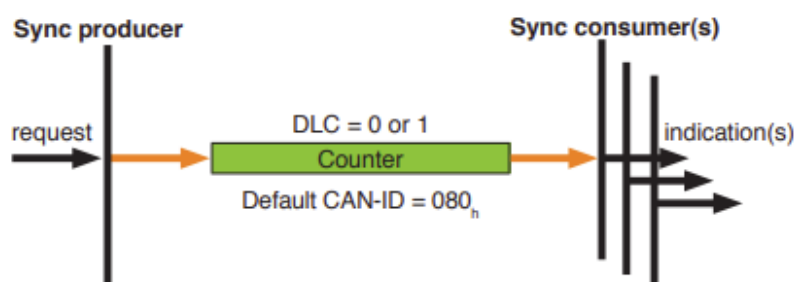
Figur 5: Exempel på CAN / COB-ID som indikerar ett SDO-meddelande från Nod 5.

2.2.3 Object Dictionary

Object Dictionary är en central databas för alla kommunicerings och konfigurerings parametrar för alla individuella noder, informationen är kategoriserad utifrån både typ och användning. Till exempel alla CANopen-inställningsparametrar är under index på 1000h, tillverkarspecifika och komponentspecifika under index på 2000h och profildefinierade med index på 6000h. Det vill säga all datatrafik (meddelanden mellan noderna) som uppstår på nätverket kommer alltid ursprungligen från en plats i OD på någon av CANopen-noderna.

2.2.4 Synchronization Protocol

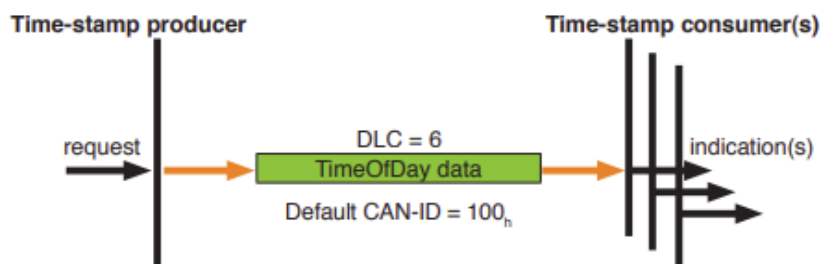
SYNC används för att synkronisera alla nätverkets moduler till samma klockpuls. Detta uppnås genom att en modul producerar ett regelbundet meddelande med COB-ID 080h på nätverket som snappas upp av alla andra moduler som i sin tur utför sina uppgifter, till exempel trigga ett PDO-meddelande, se Figur 6. (CAN in Automation CiA, 2011).



Figur 6: SYNC-protokollets meddelande struktur.

2.2.5 Time-stamp Protocol

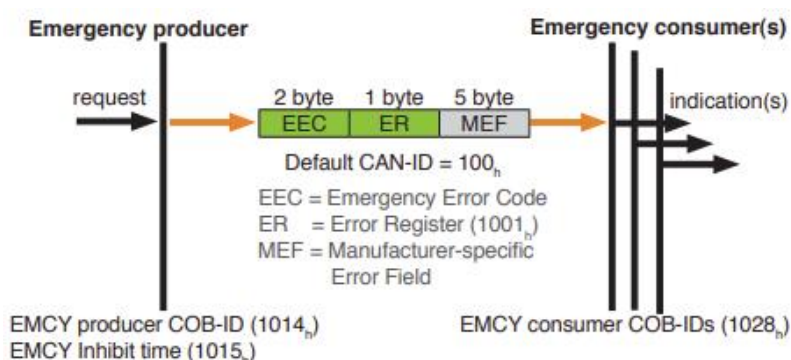
TIME används för att justera tiden på ett CANopen-nätverk, TIME-datan består av en data längd på 6 byte och beskriver tiden i millisekunder som gått sedan midnatt den första januari 1984, se Figur 7. (CAN in Automation CiA, 2011).



Figur 7: TIME-protokollets meddelande struktur.

2.2.6 Emergency Protocol

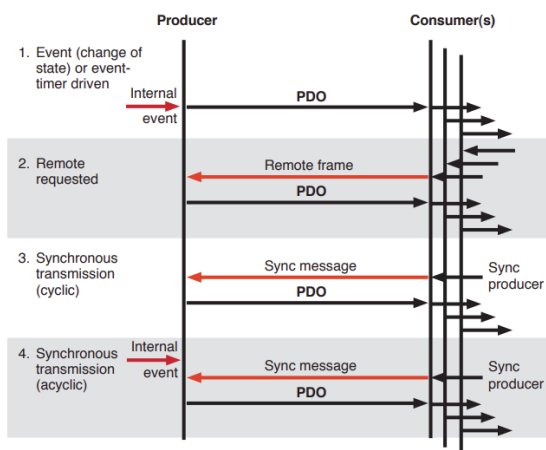
EMCY-meddelanden används för att signalera fel och störningar i nätverket, Det finns en färdig fördefinierad datastruktur för de vanligaste felkoderna, men även möjligheten att lägga till ytterligare projektspecifika felkoder. EMCY detekterar fel på både på noder och själva kommunikationen genom ett Producer-Consumer protokoll, se Figur 8. (CAN in Automation CiA, 2011).



Figur 8: EMCY-protokollets meddelande innehåll.

2.2.7 Process Data Object

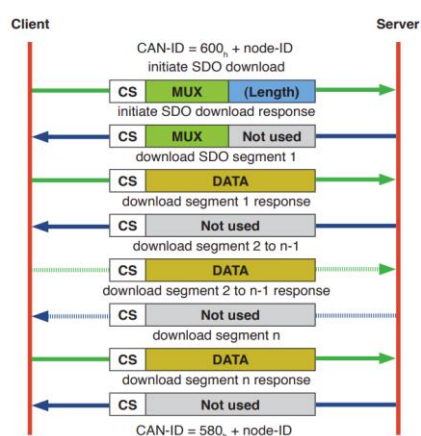
PDO hanterar dataöverföring som används i realtid. Man grupperar ofta in PDO i två olika typer, Receive-PDO, RPDO, och Transmit-PDO, TPDO. RPDO används för att ta emot data från nätverket och TPDO används för att skicka ut data. En dataöverföring med PDO kan utföras regelbundet med hjälp av till exempel SYNC eller händelsestyrt ifall till exempel en ingång ändrat värde eller av en förfrågan från en annan modul. Ett PDO-meddelande kan bestå av allt från en till åtta bitar beroende på mängden data i OD, se Figur 9. (CAN in Automation CiA, 2011).



Figur 9: Process Data Object (PDO) meddelande trafik.

2.2.8 Service Data Object

SDO hanterar dataöverföring som utförs direkt till OD på en nod och används därför främst vid uppstart av noder för att läsa över parametrar och konfigurerings data eller vid error hantering och avläsning av felkoder. Eftersom dataöverföringen ofta sker under kritiska element skickas alltid en mottagningsbekräftelse efter en dataväxling. SDO går att dela in i tre olika typer SDO-download, SDO-upload och SDO-aborttransfer. SDO-download används när data sänds från en klients OD till serverns OD. SDO-upload används när data skickas från serverns OD till en klient. SDO-aborttransfer används ifall man vill avbryta en dataväxling. Abort transfer är inte modul specifik och kan användas av både en klient och en server, se Figur 10 (CAN in Automation CiA, 2011; Pfeiffer, Ayre, & Keydel).

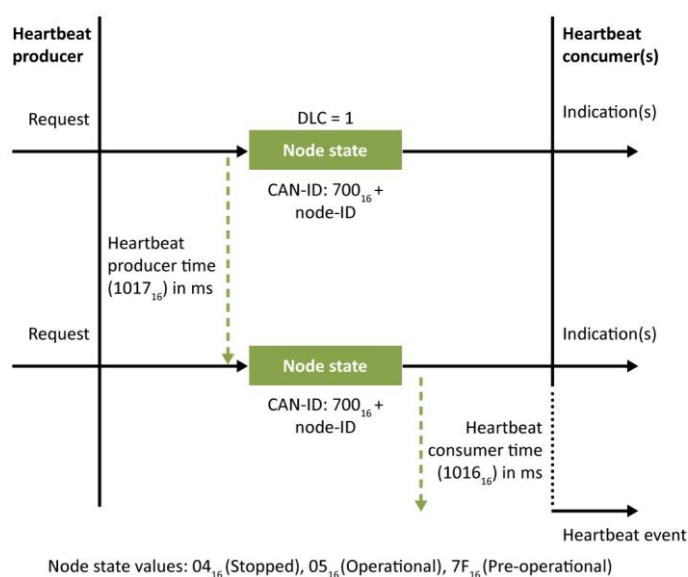


Figur 10: SDO trafik mellan en Klient och en Server enhet (CAN in Automation CiA, 2011).

2.2.9 Heartbeat

Heartbeat används för att övervaka enheternas driftstatus i CANopen-nätverket, meddelandena är endast en byte lång och består endast av den aktuella NMT-statusen. Heartbeat-meddelandena hanterar all realtidsövervakning för hela CAN-nätverket.

Alla enheter i nätverket skickar periodiskt ut Heartbeat-meddelandena helt på eget initiativ och egen periodtid, Detta betyder att alla noder är både Heartbeat-Producers och Heartbeat-Consumers, se Figur 11 (Siemens, 2019; HMS Industrial networks, 2019).



Figur 11: Överblick på Heartbeat-protokollets trafik (HMS Industrial networks, 2019).

Heartbeat-protokollet är efterträdaren för Node-Guarding som fungerar enligt samma princip med vissa undantag. Största skillnaden mellan dem är att i Node-Guarding måste alltid NMT-Masternoden periodiskt fråga av NMT-Slavenoderna om deras funktionsstadie och sedan vänta på ett svar, vilket både ökar trafiken avsevärt jämfört med Heartbeat men nätverket blir också sårbarare eftersom ett fel på NMT-Masternoden inte upptäcks av Slavenoderna som i Heartbeat. (Siemens, 2019).

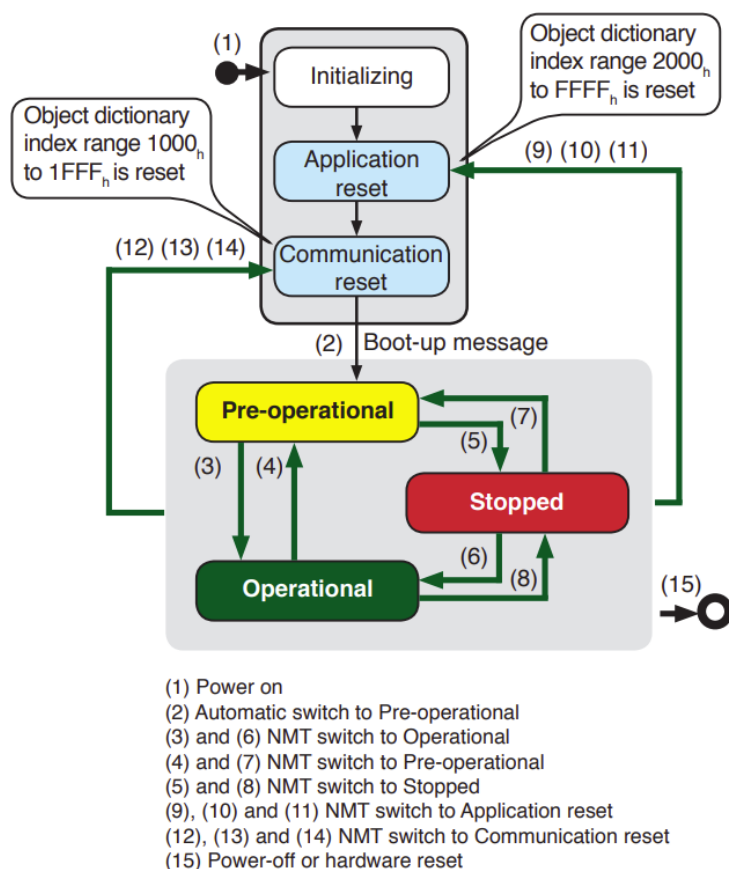
2.2.10 Network management

Network management beskriver en CANopen enhets funktionsstadie i nätverket. NMT består av tre statiska tillstånd Pre-operational, Operational och Stopped. Utöver dessa finns också temporära tillstånd såsom Initialisation, Application-Reset och Communication-Reset.

En enhet går först in i tillstånd Initialisation vid uppstart av en modul, Där initieras både enheten och applikationslagret. När initieringen är färdig går den in i nästa tillstånd som är Pre-operational, där kan moduler kommunicera mellan varandra med allt förutom PDO-meddelanden. En modul väntar i Pre-Operational tillståndet ända tills Master-noden i nätverket skickar ut ett NMT-meddelande med vidare instruktioner.

Operational tillståndet är det nästa logiska steg i ett normalt nätverk, Där körs alla kommunikationsprotokoll och allt är i normal drift.

Stopped tillståndet används ifall man vill stoppa sändningen av PDO- och SDO-meddelanden men fortsatt få NMT- och Heartbeat-meddelanden, se Figur 12. (CAN in Automation CiA, 2011; Pfeiffer, Ayre, & Keydel).



Figur 12: Network managements olika funktionsstadier (CAN in Automation CiA, 2011).

2.3 CANopen-protokollets övriga enhetsprofiler

CANopen-protokollet består av ett stort antal enhetsprofiler som implementerats i ett brett urval av komponenter, i detta kapitel redogörs ett fåtal.

2.3.1 CANopen Battery

Det finns två enhetsprofiler som specificerar CANopen-protokollet i batterisammanhang, profilen för Batterimoduler och en profil för Batteriladdare. I dem ingår flera färdiga index för till exempel upp och urladdning, temperatur och övriga batteriparametrar. (CAN in Automation CiA, 2012; CAN in Automation CiA, 2012)

2.3.2 CANopen Medical

Profilen som definierar medicinska komponenter beskrivs av, CiA 412, CANopen används brett bland flera medicinska apparater såsom mätinstrument, dosberäknings maskiner och röntgen. Profilen lägger till index för bland annat doserings skalor, strålnings tid och tryckmätning. (CAN in Automation CiA, 2008)

2.3.3 CANopen Safety

CANopen Safety, CIA 304, är en enhetsprofil som specificerar kommunikation på CAN-nätverket som kan bli säkrat upp till Safety Integrity Level 3. Detta uppnås genom att addera till en meddelande specifikation i normala trafiken som kallas Safety-Related Data Object, SRDO. Meddelandet består av samma meddelande två gånger med ena halvan inverterad, båda meddelande måste även skickas under en utsatt tid för att vara validerat. Själva data meddelandet skickas med en så kallad Black-Channel vilket beskriver en kommunikationsnivå som endast säkerhetskomponenter kommunicerar på. (Luger, 2017)

2.4 Hårdvara

I denna del redogörs vilka olika komponenter som blivit valda att användas för att testköra och övervaka den praktiska delen.

2.4.1 Siemens ET200 SP

Siemens ET200SP är ett Programmerbart logiskt styrsystem som i grundutförande saknar integrerade in eller utgångar, utan den byggs upp av moduler som läggs till. Detta innebär att man enkelt kan välja antalet in och utgångar beroende på behov, det gör att PLC:n är otroligt flexibel och kompakt utan att inverka på prestandan.

Eftersom man enkelt kan lägga till och ta bort moduler är formatet utmärkt ifall man behöver använda flera kommunikationsprotokoll på samma gång som i detta projekt. (Siemens, 2023).



Figur 13: Siemens Simatic ET200SP PLC utan tilläggsmoduler.

2.4.2 Siemens CM CAN

För att PLC:n skall förstå CAN och CANopen-kommunikationen behövs även en CAN-kommunikationsmodul som är kompatibel med ET200SP formatet. Eftersom Siemens enbart tillverkar en officiell CAN/CANopen-modul till ET200SP PLC-modellen blir det naturligt att använda den.

Ett annat val av kommunikationsmodul hade varit en variant tillverkad av Ixxat, den modulen hade även använt samma bottenplatta som den officiella versionen. Eftersom all

dokumentation var föråldrad och flera funktioner inte var uppdaterade valdes den snabbt bort (Siemens, 2023; Ixxat by HMS Networks, 2012).

2.4.3 HBC-Radiomatic

HBC-Radiomatic 510 är en robust radiomottagare som kommunicerar med CANopen-protokoll, mottagaren är en väldigt liten passform vilket gör den perfekt för användning vid automatiseringsprojekt där plats alltid är en kritisk fråga.

Radiosändaren blir en enkel HBC-Radiomatic micron 5 vilket är en klassisk kontrollerversion som endast består av tio tvåstegs tryckknappar, en trestegs brytare och en nödstoppsknapp som strömbrytare.

En ytterligare sak att hålla i åtanke vid kranautomation är ifall man vill använda en radiomottagare måste den vara säkerhetsklassad med bland annat SFS-EN 13557 + A2 vilket är Europastandarden för lyftkranar och traverser för att få användas (HBC-radiomatic).

2.4.4 Kvaser U100

För att underlätta avläsning av CANopen-trafiken används en Kvaser U100 parallellt med resten av nätverket.

Kvaser U100 visar grafiskt när fel och störnings meddelanden uppstår och också mängden trafik på nätverket vilket underlättar vid felsökning och testkörning av projekt eftersom indikationen alltid är lätt tillgänglig.

2.5 Mjukvara

För att möjliggöra att hårdvaran skall fungera enligt önskat ändamål i projektet behövs mjukvara som både klarar av att hantera funktionerna men även tolka och övervaka driften.

2.5.1 TIA portal

TIA, Totally Integrated Automation, Portal är en programvara som är utvecklad av Siemens för att hantera automationssystem. Mjukvaran fungerar som en central plattform för att utföra flera olika uppgifter inom industriellautomation. TIA Portal ingår även funktioner som grafiska programmeringsgränssnitt och diagnostiska verktyg för att undersöka felsökning och på så sätt öka produktiviteten (Siemens AG, 2018).

3 Praktiskt utförande

I detta kapitel beskrivs hur hela projektet byggs upp och hur det utförs praktiskt, dessutom redogörs hur avläsandet av CAN-kommunikationen görs med vald hård- och mjukvara.

3.1 CAN-nätverket

CAN-nätverket är uppbyggt enligt följande, Siemens ET200SP med modulen CM CAN blir konfigurerad som Masternod och SYNC-Producer, modulen får CAN-ID: 2, se Figur 14.



Figur 14: Siemens PLC med CM CAN-modul i TIA-portal.

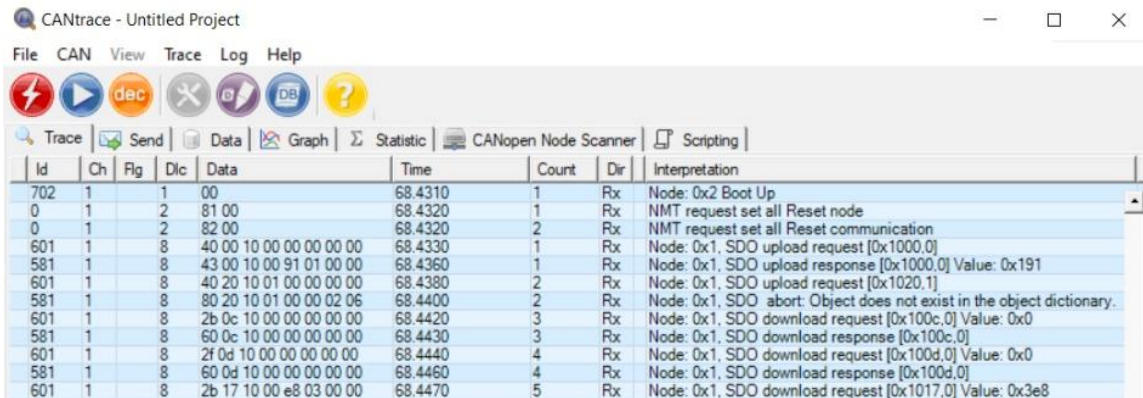
Radiomottagarens Electronic Data Sheet eller EDS-fil blir uppladdad i TIA portal och modulen blir automatiskt konfigurerad som Slavenod och får CAN-ID: 1. För att underlätta datatrafikavläsningen används en Kvaser U100 mellan modulerna, se Figur 15.



Figur 15: CAN-nätverkets uppbyggnad.

3.2 Övervakning av trafiken

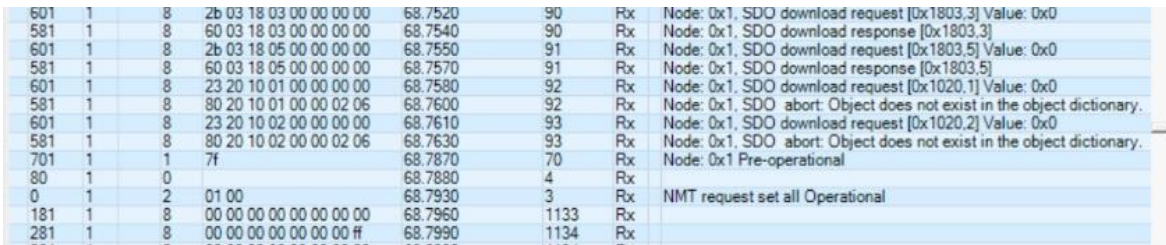
Vid första uppstart av systemet sätts noderna i NMT läget Boot-up, där återställs först all information på noderna men också all kommunikation för att rensa trafiken. Därefter börjar noderna sända ut CANopen-inställningsparametrar under index 1000h genom SDO-protokollet, se Figur 16.



Id	Ch	Flg	Dlc	Data	Time	Count	Dir	Interpretation
702	1	1	00		68.4310	1	Rx	Node: 0x2 Boot Up
0	1	2	81 00		68.4320	1	Rx	NMT request set all Reset node
0	1	2	82 00		68.4320	2	Rx	NMT request set all Reset communication
601	1	8	40 00 10 00 00 00 00 00		68.4330	1	Rx	Node: 0x1, SDO upload request [0x1000,0]
581	1	8	43 00 10 00 91 01 00 00		68.4360	1	Rx	Node: 0x1, SDO upload response [0x1000,0] Value: 0x191
601	1	8	40 20 10 01 00 00 00 00		68.4380	2	Rx	Node: 0x1, SDO upload request [0x1020,1]
581	1	8	80 20 10 01 00 00 02 06		68.4400	2	Rx	Node: 0x1, SDO abort: Object does not exist in the object dictionary.
601	1	8	2b 0c 10 00 00 00 00 00		68.4420	3	Rx	Node: 0x1, SDO download request [0x100c,0] Value: 0x0
581	1	8	60 0c 10 00 00 00 00 00		68.4430	3	Rx	Node: 0x1, SDO download response [0x100c,0]
601	1	8	2f 0d 10 00 00 00 00 00		68.4440	4	Rx	Node: 0x1, SDO download request [0x100d,0] Value: 0x0
581	1	8	60 0d 10 00 00 00 00 00		68.4460	4	Rx	Node: 0x1, SDO download response [0x100d,0]
601	1	8	2b 17 10 00 e8 03 00 00		68.4470	5	Rx	Node: 0x1, SDO download request [0x1017,0] Value: 0x3e8

Figur 16: Cantrace bild över all datatrafik vid första uppstart /del1.

När alla inställningsparametrarna är uppdaterade övergår modulen automatiskt in i NMT-stadiet Operational vilket inleder normaloperationen för nätverket och noderna, se Figur 17.



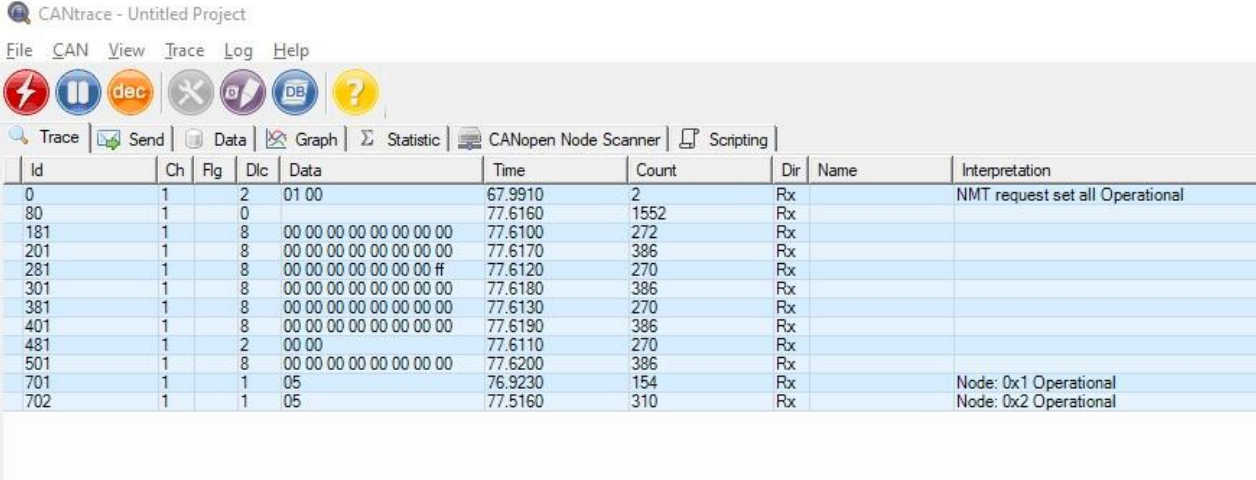
601	1	8	2b 03 18 03 00 00 00 00		68.7520	90	Rx	Node: 0x1, SDO download request [0x1803,3] Value: 0x0
581	1	8	60 03 18 03 00 00 00 00		68.7540	90	Rx	Node: 0x1, SDO download response [0x1803,3]
601	1	8	2b 03 18 05 00 00 00 00		68.7550	91	Rx	Node: 0x1, SDO download request [0x1803,5] Value: 0x0
581	1	8	60 03 18 05 00 00 00 00		68.7570	91	Rx	Node: 0x1, SDO download response [0x1803,5]
601	1	8	23 20 10 01 00 00 00 00		68.7580	92	Rx	Node: 0x1, SDO download request [0x1020,1] Value: 0x0
581	1	8	80 20 10 01 00 00 02 06		68.7600	92	Rx	Node: 0x1, SDO abort: Object does not exist in the object dictionary.
601	1	8	23 20 10 02 00 00 00 00		68.7610	93	Rx	Node: 0x1, SDO download request [0x1020,2] Value: 0x0
581	1	8	80 20 10 02 00 00 02 06		68.7630	93	Rx	Node: 0x1, SDO abort: Object does not exist in the object dictionary.
701	1	1	7f		68.7870	70	Rx	Node: 0x1 Pre-operational
80	1	0			68.7880	4	Rx	
0	1	2	01 00		68.7930	3	Rx	NMT request set all Operational
181	1	8	00 00 00 00 00 00 00 00		68.7960	1133	Rx	
281	1	8	00 00 00 00 00 00 00 ff		68.7990	1134	Rx	

Figur 17: Cantrace bild över all datatrafik vid första uppstart /del2.

Man kan med Cantrace lätt avläsa vilken typ av CANopen-trafik som körs på nätverket, Först i listan ser man ett meddelande med CAN-ID 0 och en data längd på 2 byte. Det är ett meddelande från NMT-Masternoden till noden med nod ID: 1 som sätter noden i Operational-stadiet.

Meddelandet med ID 80 är alltid SYNC eftersom innehållet är tomt, vidare är alla PDO-meddelanden listade genom att CAN-ID: 181, 281, 381, 481 är Transmit-PDO och CAN-ID: 201, 301, 401 ,501 är Recieve-PDO.

De två sista meddelandena med ID 701 och ID 702 är Heartbeat-meddelandena och beskriver NMT-statusen för Noderna med ID :1 och 2 som Operational, se Figur 18.



Id	Ch	Flg	Dlc	Data	Time	Count	Dir	Name	Interpretation
0	1		2	01 00	67.9910	2	Rx		NMT request set all Operational
80	1		0		77.6160	1552	Rx		
181	1		8	00 00 00 00 00 00 00 00	77.6100	272	Rx		
201	1		8	00 00 00 00 00 00 00 00	77.6170	386	Rx		
281	1		8	00 00 00 00 00 00 00 ff	77.6120	270	Rx		
301	1		8	00 00 00 00 00 00 00 00	77.6180	386	Rx		
381	1		8	00 00 00 00 00 00 00 00	77.6130	270	Rx		
401	1		8	00 00 00 00 00 00 00 00	77.6190	386	Rx		
481	1		2	00 00	77.6110	270	Rx		
501	1		8	00 00 00 00 00 00 00 00	77.6200	386	Rx		
701	1		1	05	76.9230	154	Rx		Node: 0x1 Operational
702	1		1	05	77.5160	310	Rx		Node: 0x2 Operational

Figur 18: Cantrace bild över all datatrafik i normalutförande.

3.3 Utveckling av programvara

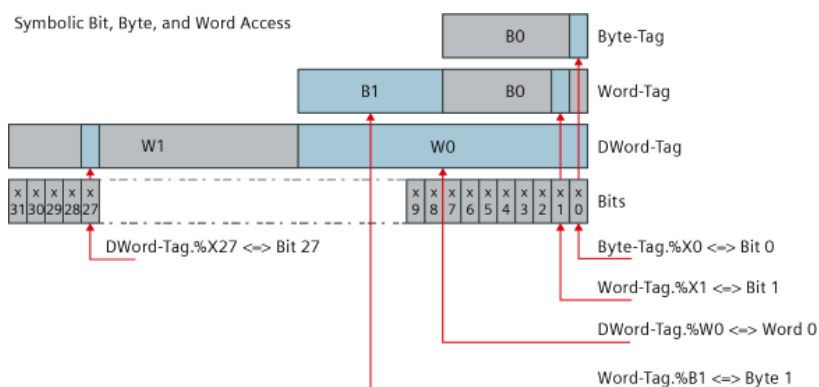
För att Siemens TIA-portalprogrammet skall uppfatta CANopen-meddelandena behövs ett funktionsblock för att konvertera hexadecimala tal till Boolean utgångar.

Eftersom kontrollen ser ut på följande sätt se Figur 19, blir det alltså sammanlagt 24 olika utgångar eftersom alla knappar har ett högre och ett lägre knappläge, vredet längst ner utgör tre knapplägen och röda nödstoppen högst upp endast utgör ett läge.



Figur 19: HBC-Radiomatic micron 5 radiomottagare (HBC-Radiomatic, 2023).

Utgående datan från CM CAN-modulen kommer i formen av 26 olika byte-värden, dessa sätts in i en Array med samma längd och används på ingångssidan av funktionsblocket för att föra in radiomottagarens signaler. Därefter måste ingångsvärdet tolkas, detta utförs enkelt med hjälp av Siemens instruktioner för individuell bit-avläsning, se Figur 20.



Figur 20: Hur man avläser individuella värden från Byte och Word och DWord (Siemens Industry Support, 2023).

För att hitta rätt byte kan man antingen gå in i Object Dictionary på CM CAN modulen och se vilken input de är riktade till, eller så kan man rent praktiskt ta hjälp av Cantrace se Figur 18, för att grafiskt utläsa både byte och vilken bit som korrelerar till knapp tryckningarna.

När man känner till rätt värde på både byte och bit klarar man av att skapa funktionsblocket, som kodspråk var det lättast att använda sig av Structured-Text (ST) eftersom det är enda språket som bit-avläsningen är möjlig att utföra, se Figur 21.

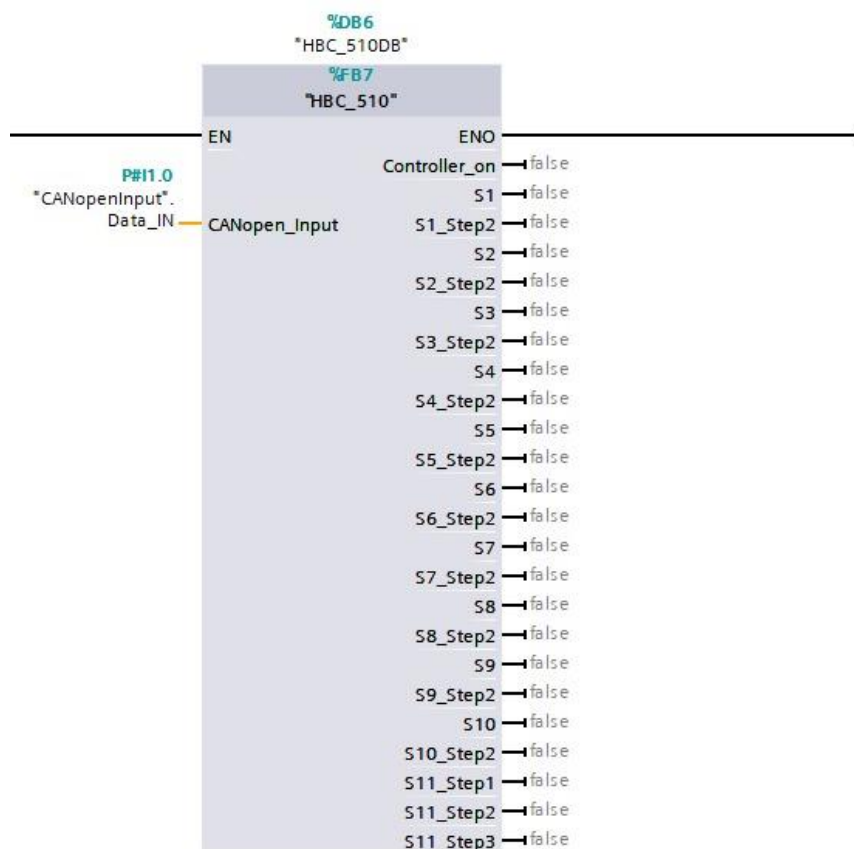
```

1
2 #Controller_on := #CANopen_Input[7].%X0;
3
4 #S1 := #CANopen_Input[8].%X1 ;
5
6 #S1_Step2 := #CANopen_Input[8].%X3;
7
8 #S2 := #CANopen_Input[8].%X0;
9
10 #S2_Step2 := #CANopen_Input[8].%X2;
11
12 #S3 := #CANopen_Input[8].%X5;
13
14 #S3_Step2 := #CANopen_Input[8].%X7;
15
16 #S4 := #CANopen_Input[8].%X4;
17
18 #S4_Step2 := #CANopen_Input[8].%X6;
19
20 #S5 := #CANopen_Input[9].%X1;
21
22 #S5_Step2 := #CANopen_Input[9].%X3;
23
24 #S6 := #CANopen_Input[9].%X0;
25
26 #S6_Step2 := #CANopen_Input[9].%X2;
27
28 #S7 := #CANopen_Input[2].%X0;
29
30 #S7_Step2 := #CANopen_Input[2].%X2;
31
32 #S8 := #CANopen_Input[1].%X7;
33
34 #S8_Step2 := #CANopen_Input[2].%X1;
35
36 #S9 := #CANopen_Input[1].%X0;
37
38 #S9_Step2 := #CANopen_Input[1].%X2;
39
40 #S10 := #CANopen_Input[1].%X1;
41
42 #S10_Step2 := #CANopen_Input[1].%X3;
43
44 #S11_Step1 := #CANopen_Input[1].%X4;
45
46 #S11_Step2 := #CANopen_Input[1].%X5;
47
48 #S11_Step3 := #CANopen_Input[1].%X6;
49
50

```

Figur 21: Urklipp från den interna koden i funktionsblocket.

När funktionsblockets programmeringskod är färdig kan man lägga till det med övrig programkod, som för exempel i programmeringsspråket Ladder, se Figur 22.



Figur 22: Färdigt funktionsblock i Siemens TIA-portal.

4 Resultat

Ett funktionsblock konstruerades för radiomottagaren HBC-radiomatic 510 som effektivt översätter alla knapptryckningssekvenser från radiosändaren till individuella utgångar. Blocket är lätt att implementera i både nya och gamla projekt eftersom det endast behöver matas med CANopen-data och ger ut ett sant eller falskt beroende på om knappen är tryckt.

Själva examensarbetet tolkas som lyckat eftersom modulen relativt enkelt går att implementeras och konfigureras, Detta har gjort att radiomodulen kommer att användas som exempel av CANopen-moduler för kommande offertförfrågningar på projekt för företaget.

Vidare utveckling av projektet hade varit att bygga vidare på nätverket genom att antingen implementera flera CANopen-moduler såsom lägesgivare, vågar och växelriktare för att få ett mer fullständigt CANopen-nätverk eller att fundera på eventuell implementering av flera specialfunktioner som till exempel CAN-FD och CAN-Safety.

5 Diskussion

Examensarbetet har varit både lärorikt och givande, svårigheter har varit bland annat att hitta bra dokumentation för komponenterna och att hitta korrekta inställningar för modulerna. Trots de svårigheterna har arbetet varit givande eftersom jag har fått nya kunskaper om både CAN- och CANopen-protokollen som jag tidigare inte alls varit i kontakt med.

Ett av de önskade målen för uppgiften var att undersöka och implementera CANopen-Safety protokollet, men detta visade sig snabbt vara omöjligt vid detta tillfälle eftersom Siemens inte tillverkar eller tillverkat någon modul som klarar av att hantera protokollet. En lösning kunde varit att integrera ett skyddsrelä som enbart hanterar Safety-meddelandena, men det var inte prioriterat i detta skede och lämnade kravlistan.

6 Källförteckning

- Beckhoff New Automation Technology. (2020). *Beckhoff Information System*. Hämtat från CANopen: <https://infosys.beckhoff.com/english.php?content=../content/1033/fbb-x510/9169765771.html&id=>
- CAN in Automation CiA. (2008). *The CANopen device profile for medical devices CiA 412*.
- CAN in Automation CiA. (2011). *CANopen application layer and communication profile CiA 301*.
- CAN in Automation CiA. (2012). *The CANopen device profile for battery chargers CiA DSP 419*.
- CAN in Automation CiA. (2012). *The CANopen device profile for battery modules CiA DSP 418*.
- CAN in Automation CiA. (2023). *CANopen – The standardized embedded network*. Hämtat från CiA: <https://www.can-cia.org/>
- HBC-Radiomatic. (2023). *HBC-Radiomatic micron 5*. Hämtat från Product description: <https://www.hbc-radiomatic.com/en/products/transmitters/micron-5.html>
- HBC-radiomatic. (u.d.). *HBC-radiomatic 510 manual*. HBC-radiomatic.
- HMS Industrial networks. (2019). *CANopen Solutions*. Hämtat från CANopen Solutions: <https://www.canopensolutions.com/index.html>
- Ixxat by HMS Networks. (2012). *User Manual, Anybus Communicator for CANopen*. Hämtat från CANopen Module for ET200S.
- Kvaser. (u.d.). *The CAN Bus Protocol Tutorial*. Hämtat från Kvaser: <https://www.kvaser.com/can-protocol-tutorial/>
- Lennartsson, K. (2019). *Comparing CAN FD with Classical CAN*. Kvaser.
- Luger, P. (2017). *Safety CAN: Why and how?* Hämtat från CAN Newsletter: <https://can-newsletter.org/uploads/media/raw/75f5867acda955cd36b66955c87955a0.pdf>
- Pfeiffer, O., Ayre, A., & Keydel, C. (u.d.). *Embedded Networking with CAN and CANopen*. Copperhill Technologies Corporation. Hämtat från Experiences of Test Automation.
- Siemens. (2019). *CANopen Tutorial*. Hämtat från CANopen Tutorial: https://cache.industry.siemens.com/dl/files/771/109479771/att_993267/v1/109479771_CANopen_Tutorial_V20_en.pdf
- Siemens. (2023). *Communication Module CAN (6ES7137-6EA00-0BA0)*. Hämtat från Equipment Manual: https://cache.industry.siemens.com/dl/files/981/109778981/att_1017685/v2/et200sp_cm_can_manual-en_US_en-US.pdf

Siemens. (2023). *Distributed I/O system*. Hämtat från ET 200SP Distributed I/O System.

Siemens AG. (2018). *Your gateway to automation in the Digital Enterprise*. Hämtat från Totally Integrated Automation Portal:
<https://assets.new.siemens.com/siemens/assets/api/uuid:0c66eeeb-b67d-4c98-aa0d-290f82d5c0d2/7801-09-tia-p-ipdf-en-181029-1.pdf>

Siemens Industry Support. (2023). *Industry Online Support*. Hämtat från How can you access unstructured data types bit-by-bit, byte-by-byte or word-by-word and symbolically in STEP 7 (TIA Portal)?:
[https://support.industry.siemens.com/cs/document/57374718/how-can-you-access-unstructured-data-types-bit-by-bit-byte-by-byte-or-word-by-word-and-symbolically-in-step-7-\(tia-portal\)?dti=0&lc=en-WW](https://support.industry.siemens.com/cs/document/57374718/how-can-you-access-unstructured-data-types-bit-by-bit-byte-by-byte-or-word-by-word-and-symbolically-in-step-7-(tia-portal)?dti=0&lc=en-WW)