



Fagskolen Tinius Olsen
Avdeling for Teknikk og industriell produksjon

PROSJEKTRAPPORT 2016/17:

Robotlab



Oppdragsgiver: Fagskolen Tinius Olsen

Utarbeidet av:

Eirik Hatland

Håvard Vikås

Sindre Hellingsrud

Klasse: 2FME

Antall sider: 56

Vedlegg: 17

Innlevert dato: 30.05.17

Sammendrag:

Denne prosjektrapporten tar for seg prosessen med å utvikle en labøvelse på vegne av fagskolen Tinius Olsen. Prosjektet tar utgangspunktet i et behov fra FTO om en labøvelse som inneholder et samspill mellom vision-system og robot. Denne labøvelsen er tiltenkt en rolle i undervisningen ved den fremtidige smart-automasjonslinjen, og har til hensikt å gi studentene et innblikk i oppbygningen og virkemåten til et vision-system.

Labøvelsen gruppen har produsert baserer seg på en «pick and place» oppgave, hvor plukkobjekter føres inn i robotcellen ved hjelp av et transportbånd. I enden av dette transportbåndet er det montert en vision-sensor, som kontinuerlig søker etter de ulike plukkobjektene. Når et plukkobjekt oppdages av vision-sensoren, vil transportbåndet stoppe, og informasjon om objektets geometriske form og koordinatene til plukkepunktet sendes til roboten via en PLS. Roboten bruker så denne informasjonen til å hente plukkobjektet og plassere det i en tildelt boks.

Prosessen med å lage denne labøvelsen har gått fra ideutvikling til utarbeidelse av et fullstendig produksjonsgrunnlag, med tekniske tegninger og beregninger. I tillegg er det utarbeidet en brukerveiledning for labøvelsen, som beskriver hvordan vision-sensoren, og roboten programmeres.

Labøvelsen har blitt produsert i henhold til kravene og fra oppdragsgiver. De økonomiske rammene har blitt overholdt, og prosjektet ble ferdig stilt innen tidsfristen.

Emneord: Robot, vision-system, labøvelse.

Forord

Dette er et hovedprosjekt utført av fagskolestuderenter i mekatronikk på avdeling for teknikk og industriell produksjon ved Fagskolen Tinius Olsen i Kongsberg. Oppgaven gir 12 fagskolepoeng og omfatter et samspill mellom robot og vision-system som skal bli en labøvelse i undervisningsplanen for robotteknologi.

Prosjektgruppen har erfaring fra mekanisk produksjon, pneumatikk og svak- og sterkstrøms installasjoner og automasjon. Med denne sammensatte bakgrunnen mener vi oppgaven passet oss og vi har hatt tro på gode resultater.

Takk til hovedveileder Tommy Hvidsten for gode tips underveis i prosjektperioden.

Videre retter vi en takk til:

- Stine Hvila Lind
- Per Morten Grøslie
- Kim Sembsmoen

Spesiell takk til

- Erik Lysen fra Omron for god hjelp og support av vision-system.
- Iver Hardeng fra Omron som har hjulpet oss med kommunikasjon mot robot.
- Bent fra Scala RoboTech som har hjulpet oss med programmering av robot.
- Harald Hals fra Advanced Machine Company AS for veiledning i oppstartsfasen.

Tusen takk for god hjelp og veiledning!

Innhold

Del 1.2: Produktet.....	1
1.1: Tema, problemstilling og avgrensning.....	1
1.1.1: Tema.....	1
1.1.2: Problemstilling.....	1
1.1.3: Avgrensning av problemstillingen	2
1.1. Definisjon av begreper i problemstillingen	3
1.2: Teori	4
1.3: Metode/fremgangsmåte	7
1.4: Resultat	12
1.5: Drøfting/diskusjon	28
1.6: Konklusjon/oppsummering:.....	31
Del 2: ProsesSEN.....	33
2.1: Referat fra arbeidet.....	33
2.2: Planlegging:.....	35
2.2.1:Tidsbruk.....	35
2.2.2: Plan for arbeidet.....	37
2.2.3: Gruppeledelse.....	39
2.3: Kommunikasjon.....	40
2.4: Ressurser.....	42
2.5: Læringsutbytte.....	43
2.6: Verktøy og praksis	47
2.7: Konklusjon.....	49
Vedlegg	50

Del 1.2: Produktet

1.1: Tema, problemstilling og avgrensning

1.1.1: Tema

Bakgrunn for prosjektet

Prosjektets opphav kommer fra Fagskolen Tinius Olsen. Tommy Hvidsten ved Fagskolen har lenge ønsket å ta i bruk en eksisterende robot i undervisningen.

Tema

I forbindelse med dette ønsker fagskolen å integrere et vision-system med en robot som tilhører FTO. Samspillet mellom robot og vision-system er tiltenkt for at studenter skal kunne tilegne seg grunnleggende kunnskap og få et innblikk om systemet. Ideen er å få en industrirettet labøvelse slik at studentene kan få et best mulig perspektiv på hvordan robopter samhandler med vision-system i industrien.

1.1.2: Problemstilling

Hvordan kan gruppen på best mulig måte lage en labøvelse som har til hensikt å gi studentene ved fagskolen et innblikk i samspillet mellom robot og vision-system?

1.1.3: Avgrensning av problemstillingen

Produktets funksjon

- Samspill mellom robot og vision-system
- Det valgte vision-systemet må la seg integrere med skolens Motoman Sia-20D robotarm.
- Bilder må kunne tolkes og brukes til kvalitetskontroll av fremførte deler og posisjonsinformasjon til robotstyringen slik at roboten kan gripe deler basert på bildeinformasjon.

Systemet kan anvendes til flere oppgaver

- Ettersom hele robotcellen kan bli flyttet til neste år, må gruppen velge et system som kan lett bygges om ved å bruke de samme komponentene.
- Det skal kunne tilpasses ny funksjonalitet ved en senere anledning.
- Infrastrukturen skal kunne passe inn i robotcellen og den skal kunne flyttes ved evt. endringer.

Økonomi

- Fagskolen Tinius fremskaffer nødvendig midler for en god løsning i en størrelses orden mellom 100-200 000kr.

Labøvelsen

- Labøvelsen skal bygge på tidligere labøvelser fra undervisningsplanen innen robotteknologi.
- Labøvelsen skal inneholde programmering av vision-kamera og programmering av robot.

Dokumentasjon

- Tegninger av elektrisk oppkobling og annet arbeid gruppen har utført.
- Brukermanualer for systemet skal sorteres og legges i egen mappe, slik at det er lett tilgjengelig.

1.1. Definisjon av begreper i problemstillingen

Labøvelse - Det latinske ordet *laboratorium* betyr egentlig arbeidsrom, men blir normalt brukt i betydningen *forsøksrom*. På laboratoriene utføres det målinger, testing, analyse og produktprøving.

Vision-system - Er definert som «*den maskinvaren og programvaren som skal til for å etterligne funksjonen til et øye*».

Robot - En datastyrt enhet som ved hjelp av sensorer kan motta data fra omgivelsene, bearbeide disse og reagere ved å iverksette handlinger i henhold til forhåndsprogrammerte regler.

Samspill - Betyr å virke i fellesskap

1.2: Teori

Vision system

Et vision-system er definert som «*den maskinvaren og programvaren som skal til for å etterligne funksjonen til et øye*». Et synssystem er ikke nødvendigvis begrenset til det synlige spektrum, men kan også omfatte fortolkning av tredimensjonale bilder. Et typisk vision-system består gjerne av et digitalt kamera/vision-sensor, belysning, optikk og programvare for bildebehandling. Ved hjelp av disse fire komponentene er det mulig for en datamaskin å inspisere og analysere innholdet i ett stillestående eller bevegelig bilde. De mest typiske bruksområdene for denne teknologien er i forbindelse med kvaliteskontroll, overvåkning av automatiserte prosesser eller til veiledning av robot.

Vision-systemer brukes ofte til å sortere. Man kan identifisere objektet ved å måle dimensjoner, sjekke overflatestruktur eller finne hvilken form og farge objektet har. Denne informasjonen kan for eksempel brukes for å fortelle en robot hvilket program den skal bruke, eller hvilken bevegelse den skal foreta seg. I bedrifter som driver med kvalitetsteknologi er vision-systemer blitt en nødvendig del av produksjonslinjen ettersom det er praktisk umulig å gjennomføre manuell kontroll av store volum. Sammenligner man manuell kontroll gjort av mennesker og vision-systemer, vil forskjellen være enorm når det kommer til arbeidskapasitet og nøyaktighet.

Et viktig kjennetegn for vellykkete vision-systemer er et godt forarbeid slik at måleoppstillingen sikrer den repeterbarheten, nøyaktigheten og kontrasten som er nødvendig. I systemer der vision-sensorer skal brukes er lyssettingen en svært viktig faktor for vellykkede målinger. Med varierende lysforhold kan det oppstå uønskede skyggefekter og det kan føre til at konstanten rundt måleobjektet blir uklar. Dersom det er en ekstern lyskilde som f.eks. takbelysing som er forsynt med en spenning med 50Hz, vil det også kunne påvirke målingene/kvaliteten på bildet. En annen faktor som er viktig for få vellykkede målinger er kontrasten på flaten der måleobjektet ligger. Denne flaten bør være matt, slik at den ikke reflekterer lys inn i sensoren.

(Electrical and Electronics Engineering Dictionary, 2017)

Robot

Ordet robot kommer fra det tsjekkiske ordet *roboata* som betyr arbeid. Roboter er i dag særlig brukt i industrien og utfører arbeidsoppgaver som enten er for farlige for mennesker eller for monotone. Roboter som utfører denne typen arbeid kalles gjerne for en industrirobot. Man skiller mellom roboter og industriroboter, og ifølge ISO 8372 er en industriell robot definert som «*An automatically controlled, reprogrammable, multipurpose manipulator programmable in three or more axes, which may be either fixed in place or mobile for use in industrial automation applications*».

Typiske arbeidsoppgaver for en industrirobot er maling, sveising eller plukking. Roboter har generelt svært mange bruksområder og blir mer og mer nøyaktig og kan programmeres til å arbeide med stor presisjon. Det finnes utallige varianter av verktøy for industriroboter, eksempelvis vakuumgriper, magnetgriper og parallelgripere. Men som regel er verktøyet designet og tilpasset til den arbeidsoppgaven roboten er satt til å gjøre.

Når det kommer til programmering av robot finnes det mange måter å gjøre det på. Noen roboter har en egen modus, der man kan guide roboten for hånd mens posisjonene lagres fortløpende. Denne typen programmering brukes ofte til å programmere sveise- og malingsroboter, der kjørebanen ofte er kronglete og inneholder bevegelser i mange akser samtidig. Den mest vanlige måten for programmering er å kjøre roboten manuelt mens man lagrer koordinatene i et tredimensjonalt koordinatsystem. Koordinataksene er vanligvis betegnet som XY og Z.

Lover, regler og forskrifter**EMC direktivet**

EMC direktivet sier «*Udstyr skal ikke utstråle for mye elektromagnetisk støy, samtidig som de skal tåle å bli utsatt for støy uten at forventet funksjonalitet reduseres*»

(Westin, 2014)

NEK-håndbok

NEK 144:2004: Grafiske symboler for elektroteknisk dokumentasjon.

(Standard.no, 15.03)

Maskindirektivet, 2017.

EN 606204-1 Maskinen elektrisk utrustning del 1: Generelle krav

(Forskrift om maskiner, 2014)

1.3: Metode/fremgangsmåte

Konseptutvikling

På bakgrunn av krav og produktspesifikasjoner fra oppdragsgiver har gruppen utviklet fire konsepter for infrastruktur og to konsepter for vision-kamera. Konseptene for infrastruktur dreier seg i hovedsak hvordan plukkobjektene blir fremført til kamera for behandling av bildeinformasjon. Konseptet for vision-kamera omhandler hvilken type kamera som ble valgt.

(Se vedlegg 1 for konseptutvikling med tilhørende prisestimat)

Gruppen har i samarbeid med oppdragsgiver valgt konsept 3 for infrastruktur, da dette er det konseptet som gir den mest industrirettet opplæringsplattform. Det gir en bedre innsikt i hvordan en slik oppgave kan løses i industrien sammenlignet med de andre konseptene.

Gruppen mener også at læringsutbyttet for dette konseptet er større enn de andre alternativene. Dette kommer av at labøvelsen vil inneholde en styring av transportbånd, tolking av bildeinformasjon fra vision-kamera samt robotstyring.

HMS-perspektivet vil bli opprettholdt ved at man slipper å gå inn og ut av cellen. Prisestimatet for prosjektet er ikke det billigste, men gruppen har tatt utgangspunkt i styringsparametret fra oppdragsgiver som rangerer kvalitet over kostnad.

Gruppen har valgt konsept 1 for vision-kamera. Dette grunnet flere anbefalinger fra leverandører og eksternt fagpersonell som mener det blir for omfattende å ta i bruk et 3D vision-system i et skoleprosjekt. Et 2D-kamera klarer ikke å måle avstanden mellom kamera og objekt, derfor har gruppen valgt å ha fast høyde på måleobjektene. Gruppen mener også at et 2D-kamera gir et godt nok innblikk i vision-programmering og dekker de grunnleggende prinsippene for forståelse av funksjonaliteten. Det er også en vesentlig prisforskjell mellom disse kameraene. Et 3D kamera har en startpris på 50 000,- mens et 2D kamera har en startpris på 5000,-

Etter det ble tatt en avgjørelse om konseptvalg sammen med oppdragsgiver, innhentet gruppen tilbud fra ulike leverandører. Valg av leverandør for vision-systemet havnet på Omron. Dette var for at de har et samarbeid med Scala Tech (leverandør av robotsystem) og visste hva dette dreide seg om. De kunne også gi oss et bra tilbud på utstyret som skulle brukes ettersom dette skulle benyttes i et skoleprosjekt.

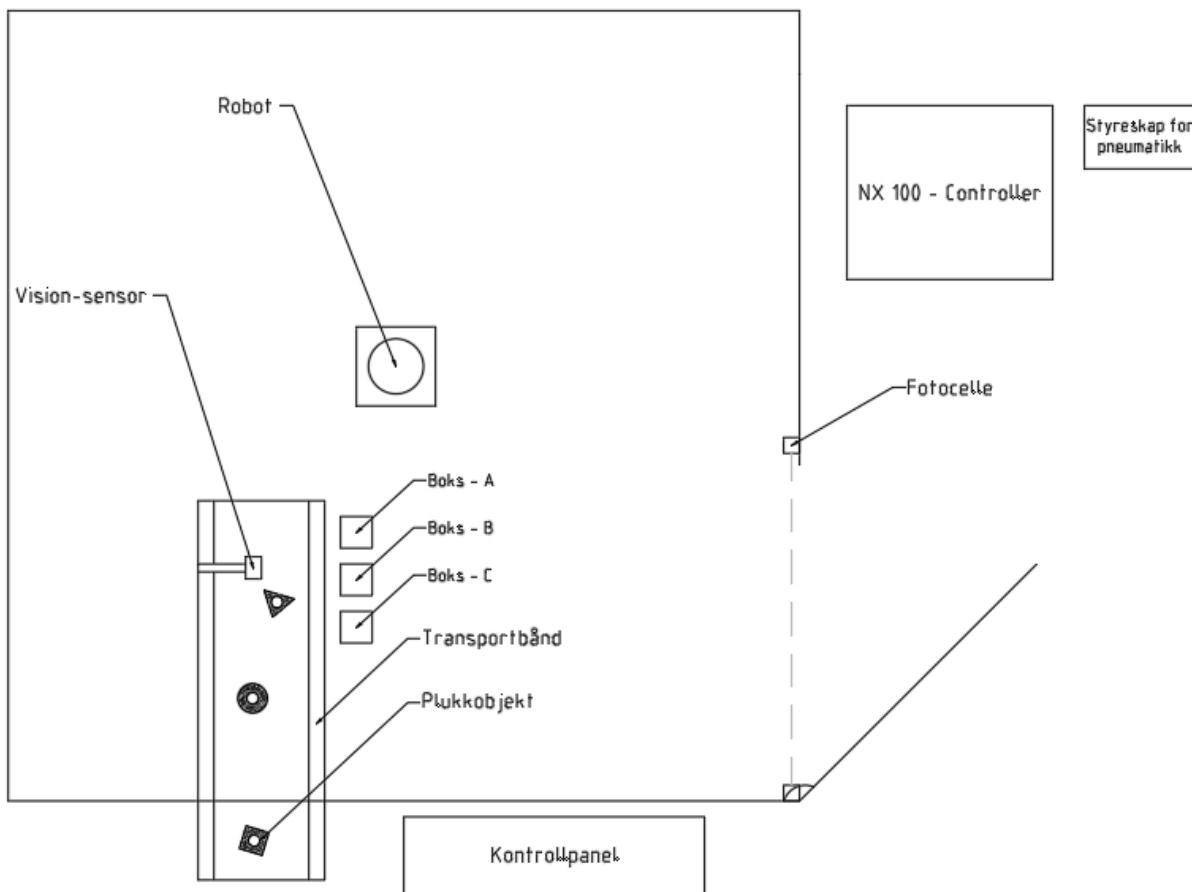
(Se vedlegg 2 for pristilbud fra Omron)

Valg av leverandør til infrastruktur havnet på ITO, de kunne levere et transportbånd etter spesifikt mål og med motor og enkoder.

(Se vedlegg 3 for pristilbud fra ITO)

(Se vedlegg 4 – for Konseptutvikling oversatt til engelsk)

Beskrivelse av valgt konsept



Bilde: 1 Skisse av robotcelle (SHE.engineering, 2017)

Robotcellen

Det valgte konseptet er basert på det som omtales i industrien som en «pick and place» oppgave. Det vil si at roboten i samspill med vision-systemet skal sortere de ulike plukkobjekter etter hvilken geometrisk form de har.

Plukkobjektene blir ført inn i robotcellen ved hjelp av et transportbånd (bilde 1). Samtidig som transportbåndet går, foretar vision-sensoren kontinuerlig søk etter de ulike plukkobjektene. Når sensoren oppdager et objekt som stemmer overens med de programmerte betingelsene, sender den et signal om at transportbåndet skal stoppe. Når båndet har stoppet, sender vision-sensoren informasjon om hvilken geometrisk form objektet har, samt XY koordinatene til plukkepunktet.

Denne informasjonen sendes først til en PLS, før den går videre til roboten. Roboten bruker så denne informasjonen til å plukke opp objektet og plassere det i en tildelt boks; boks A, B og C. Når roboten har gjort dette går den tilbake til venteposisjon, og den sender et signal om at båndet skal starte igjen.

Styring av transportbånd

Transportbåndet er drevet av en asynkron trefas-motor, som er koblet opp til en frekvensomformer. Gruppen har valgt å bruke en frekvensomformer, slik at det er mulig å styre akselerasjon/retardasjon og hastigheten på båndet.

Frekvensomformeren er plassert i et eget skap i kontrollpanelet, og den kommuniserer direkte med PLSen over EtherCat. Båndet startes ved at man aktiverer startbryteren på kontrollpanelet, og går kontinuerlig helt til vision-sensoren sender et signal om at båndet skal stoppe.

Vision-sensor

Vision-sensoren er montert på et justerbart oppheng i enden av transportbåndet. Linsen på sensoren peker ned mot overflaten på båndet, og synsfeltet er justert til å være like bredt som transportbåndet. Så lenge transportbåndet er i bevegelse, vil vision-sensoren søke etter de ulike plukkobjektene lang overflaten på båndet. Når sensoren oppdager plukkobjektet, stoppes båndet og koordinatene sendes til PLSen over Ethernet.

PLS og Robot

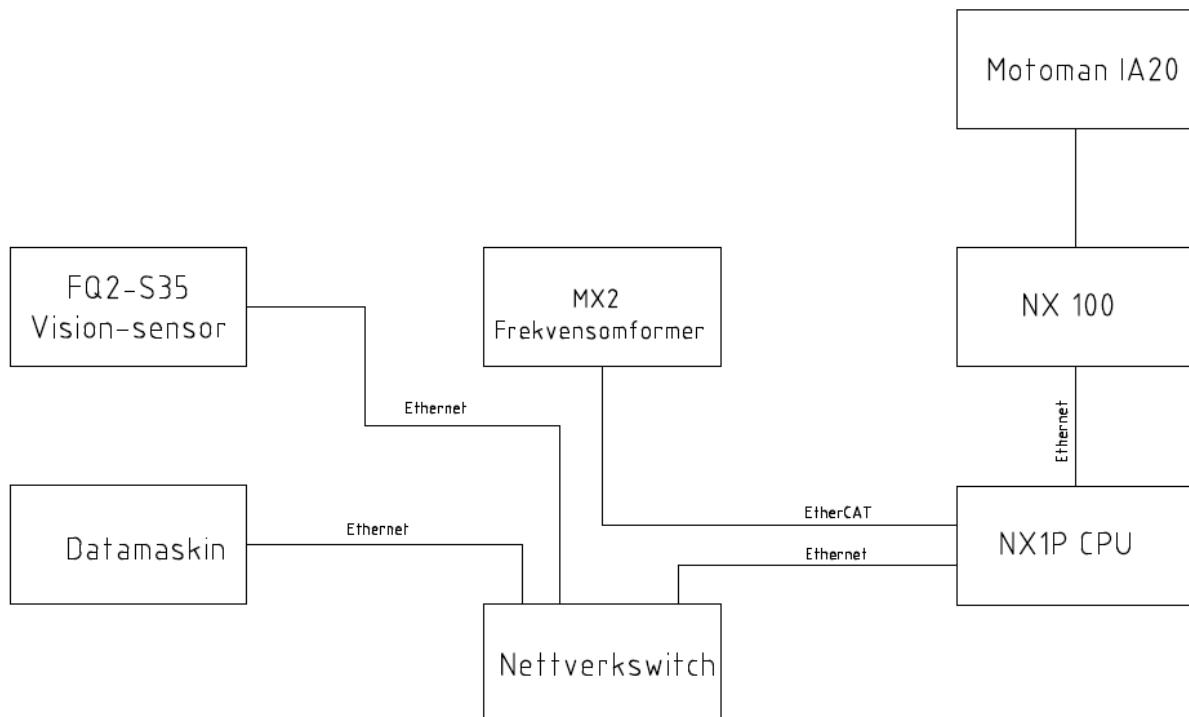
PLSen er på mange måter hovedkomponenten i prosjektet. Det er den som styrer hva de ulike komponentene skal gjøre, og den som mottar informasjon fra de ulike bryterne. PLSen vil bruke den informasjonen den mottar fra vision-sensoren til å fortelle roboten hvilket objekt som ligger på transportbåndet og koordinatene til plukkepunktet. Denne informasjonen bearbeides i PLSen og videresendes til roboten via Ethernet. PLSen er også koblet opp mot en datamaskin, slik at det er mulig å monitorene prosessen underveis.

Sikkerhet i robotcellen

Robotcellen er utstyrt med sikkerhetsfunksjoner og verneinnretninger som gjør at farens risiko for skader på personer og materiell er holdt til det minimale. Ved inngangen til robotcellen er det montert en lysgardin-sensor som detekterer all bevegelse inn til cellen. Denne sensoren fungerer som en nødstopp, og vil stoppe robotens bevegelser dersom noen går inn i cellen mens roboten er i arbeid. For å resette denne alarmen er det montert en kvitteringsbryter ved inngangsdøren. Kontrollpanelet er også utstyrt med en nødstoppbryter. Denne bryteren stopper transportbåndet og roboten dersom den aktiviseres, og startbryteren på kontrollpanelet brukes til å resette denne nødstoppen.

Kommunikasjon

Illustrasjonen viser hvilke komponenter som kommuniserer med hverandre, og hvilken kommunikasjonskabel som er benyttet.



Bilde: 2 Kommunikasjon mellom komponentene (SHE.engineering, 2017)

1.4: Resultat

Dette kapittelet omhandler hvordan gruppen har kommet frem til resultatet. Etter at valg av konsept ble fattet, har gruppen måtte sette seg inn i hver enkelt komponent for å kunne danne et helhetlig system. Videre har gruppen kategorisert de ulike systemene i vision-sensor, infrastruktur, robot, kommunikasjon og kontrollpanel. Kapittelet beskriver teorien bak de ulike komponentene, og hvordan gruppen har brukt denne teorien til å komme frem til resultatet.

Vision-sensor

FQ2-S35 vision-sensor

Vision sensoren som gruppen har valg å bruke i dette prosjektet er et FQ2 -S35 2D sensor fra Omron. Detter er en kompakt sensor som passer for alle typer objekt-gjenkjenning, posisjonering og måle-operasjoner. Sensoren har en innebygd lyskilde, og er i stand til å gjøre 32 målinger samtidig. Sensorene er utstyrt med syv innganger, tre utganger og Ethernet for kommunikasjon. Vision-sensoren inneholder Omrons nyeste utgave av shapesearch, shapesearch III, som benytter seg av bedriftens nye vision-algoritme. Denne nye algoritmen øker toleranse-nivået til shapesearch-verktøyet slik det kan med økt hastighet og høyere presisjon utføre objekt-gjenkjenning selv i vanskelige lys-omstendigheter. Sensoren er utstyrt med en 8mm linse, og har oppløsing på 928x828.

(Omron, 2017)

(Se vedlegg 5 - Elektrotegning, tegningsnr: 1006-FQ2 vision-sensor)

Vision-sensor programmering

Programmering av vision går ut på at man ser på bildet som en matrise. Størrelsen på matrisen avhenger av antallet piksler i x og y retning, og summen av disse piksler defineres også som bildets oppløsning. De pikslene som er i matrisen har en gitt tallverdi som representerer hvilken farge pikselen har. Ut fra denne tallverdien kan man med en programvare som heter touchfinder gjøre en vurdering av matrisen. Denne programvare inneholder en rekke verktøy som kan brukes i til objekt-gjenkjenning. Eksempler på typiske verktøy som brukes er: kantdeteksjon, objekt-gjenkjenning eller mønstergjenkjenning.

I dette prosjektet er vision-sensoren programmert til å søke etter ulike objekter som firkant, trekant og sirkel. Felles for disse objektene er at de har et gjennomgående hull i senter for plukkeverktøyet. For å søke etter de ulike geometriske formene og senter for plukkeverktøy, har gruppen benyttet et søkeverktøy som heter shapeseach III. Shapesearch III -verktøyet er funksjon som detekterer brukerdefinerte figurer for å estimere objektets posisjon, likhet, og orientering. Man kan si at dette verktøyet er det verktøyet som imiterer menneskeøyet på best måte av de verktøyene som er tilgjengelig. Ut fra den vurderingen som gjøres av bildet sendes det et signal som enten er NG eller OK. Dette er et digitalt signal som sendes fra utgangen på sensoren og gir beskjed om at objektet i bildet er godkjent. Informasjons om objektets geometriske form og koordinatene til plukkepunktet sendes til PLSen over Ethernet.

(Se vedlegg 6 – bruksanvisning vision-system)

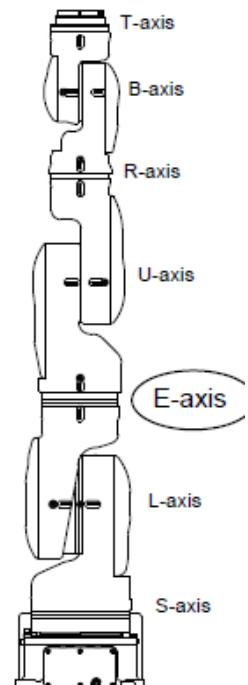
Robot

Motoman IA 20

Robotarmen som er brukt i dette prosjektet er en Motoman SIA 20D. Dette er en syv-akset robotarm med en nyttelast opptil 20kg og rekkevidde på 1,5 meter. Robotarmen har syv akser, noe som gjør at den har stor bevegelsesfrihet og kan manøvrere på meget trange steder. I industrien brukes denne typen robot til oppgaver som montering, plukking og pakking. Det at robotten har en stor bevegelsesfrihet gjør også at den egner seg godt til oppgaver som sveising og lakking. Ved å kombinere bevegelsesaksene kan robotten bevege seg fritt i totalt syv frihetsgrader. T, R, E og S leddene er roterende ledd. (*Bilde 3*)

Robotarmen er utstyrt med en egen kontrollenhett, NX 100. Det er denne kontrolleren som styrer robotarmen, og den har kapasitet styre opptil fire roboter samtidig. Kontrollenhett har også en rekke analoge og digitale I/O'er som kan programmeres til å styre eksternt utstyr som i dette tilfellet er den pneumatiske griperen. Kontrollenheten kommuniserer med PLSen og mottar koordinatene og informasjon om plukkobjektets geometriske form over Ethernet. Etter at robotten har utført oppgaven sender den signal tilbake til PLSen om at jobben er utført og at prosessen kan fortsette.

(Se vedlegg 5 – Elektrotegninger, tegningsnr: 1004 – NX100)



Bilde: 3 Motoman IA20
(Yaskawa, 2017)

Griperverktøy

Robotarmen er utstyrt med en pneumatisk senter-griper. Griperen består av tre like fingere som beveger seg proporsjonalt i forhold til hverandre. Griperen er i utgangspunktet designet til å håndtere hjullager etter den indre sirkelen. Griperens stilling styres av en elektromagnetisk 5/2 ventil som er koblet til en digital utgang på kontrollenheten til roboten. Denne ventilen er plassert i et eksternt skap ved siden av kontrollenheten og er forsynt med et 8bars trykk.

(Se vedlegg 7 - Pneumatikkskjema)

Plukkobjekter

For å produsere plukkobjektene valget gruppen 3D-printing fremfor tradisjonell maskinering. Ettersom modellen ikke skulle utsettes for noe nevneverdig belasting bortsett fra flaten der griperverktøyet skulle plukke objektet, ble det besluttet at plastikk var et sterkt nok materiale. Flaten der griperverktøyet berører plukkobjektet, er forsterket med en stål-ring. Plastmaterialet som ble benyttet var av kvaliteten PA 100.

3D-printing er en form for additiv tilvirkning som baserer seg på å bygge objekter i fast materiale med utgangspunkt i en tredimensjonal digital modell. Prosessen går ut på at materialet ekstruderes gjennom en varm dyse og fester seg lagvis på overflaten. Modellen blir første tegnet i et 3D modelleringsprogram, som deretter genererer ut G-koder. G-kode er et språk som forteller datastyerte verktøymaskiner om hvordan maskinen skal bevege seg, hvor fort den skal gå, og hvilken vei den skal kjøre. Kjørebanen til verktøyet, som i denne sammenheng er dysen til 3D-printern kalles for toolpath. I programvaren som 3D-modellen blir fremstilt i blir også fyllmønsteret bestemt. Fyllmønsteret er med på å bestemme styrken og tyngden til modellen. De vanligste fyllmønsterene er konsentrisk, rektangulært og honeycomb. Sammenlignet med tradisjonell maskinering er 3D-printing en relativt kostnadseffektiv tilvirkningsteknikk. En annen fordel er at det ikke blir noen vrakmateriell sammenlignet med freasing eller dreiling, som er en sponfraskillende prosess.

(Rosvold, 2017)

(Se vedlegg 8 - Tegninger av plukkobjektene)

Programmering av robot

Programmeringen av roboten ble utført online via pedanten, som betyr at roboten er manuelt kjørt til ønsket posisjon og koordinatene har blitt lagret. Koordinatene har blitt lagret som variabler, slik at man kan endre variablene hvis det skulle gjøres endringer i robotcellen ved en senere anledning. Det er opprettet et eget koordinatsystem som er i samsvar med koordinatsystemet til vision-sensoren. Dette koordinatsystemet er et XY og Z koordinatsystem. Nullverdien til Z er omrentlig 30mm høyere enn overflaten på båndet. Når roboten skal plukke et objekt, vil den først gå til nullverdien i Z, før den så går -30mm vertikalt.

Roboten mottar totalt tre forskjellige variabler fra PLSen. To av disse variablene forteller roboten om XY verdien til plukkepunktet. Den siste variabelen forteller roboten om plukkobjektets geometriske form, og hvilken eske det skal plasseres i. Denne variablen kan ha verdien 1,2 eller 3, hvor 1 er firkant, 2 er sirkel og 3 er trekant. Programmet er delt inn i flere jobber, som kalles på av hovedprogrammet. Hvilken av jobbene som skal kalles på, bestemmes utfra hvilken verdi den siste variabelen har.

Det første roboten er programmert til å gjøre er å sjekke om den siste variablen er over null. Hvis innholdet i denne variablen er over null, vil roboten kalle på en jobb og henter plukkobjektet. Denne jobben inneholder flere posisjoner, som hjemposisjon og veipunkter som roboten må gå via for komme trygt frem til plukkeposisjonen. Disse veipunktene gir roboten en fast rute den må gå for å komme frem til plukkeposisjonen, og på denne måten unngår man at roboten bestemmer ruten selv. Når roboten har kommet frem til plukkeposisjonen, kaller den på en annen jobb som lukker griperverktøyet. Deretter vil roboten gå via veipunktene og tilbake til hjemposisjonen. Når roboten har kommet til hjemposisjonen, sjekker den verdien av innholdet i den siste variablen. Ut fra denne verdien kalles det på en jobb som plasserer plukkobjektet i riktig eske. Når dette er gjort, vil roboten gå tilbake til hjemposisjon, variablene resettes og prosessen kan starte på nytt.

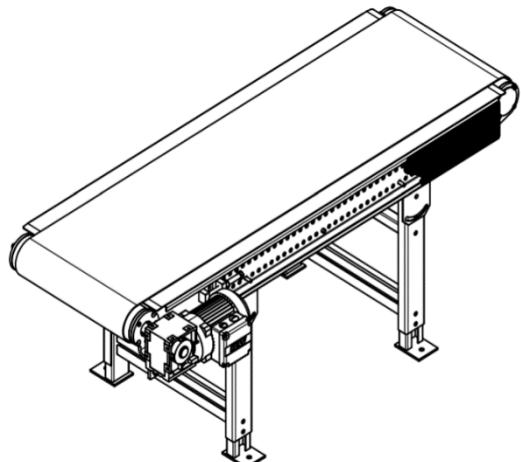
(Se vedlegg 6 – Robotprogram)

(Se vedlegg 7 – Bruksanvisning for labøvelsen)

Infrastruktur

Valget av leverandør til transportbånd havnet på ITO. De kunne levere et transportbånd med følgende krav fra oss:

- En motor med gir for å drive båndet
- En mørkegrønn duk for å ha best kontrast til bildene
- Justerbare ben, (høyde 800+-100mm) for å justere høyden i forhold til robotarmens rekkevidde.
- Enkoder som kan kommunisere med Omrons PLS



Bilde: 4 Transportbånd (ITO, 2017)

Motor og enkoder

Transportbåndet er drevet av en $\frac{1}{4}$ kW asynkron trefas-motor levert av Sew. Motoren er utstyrt med et utvekslingsgir med en utveksling på 1/32,5. I tillegg til dette er motoren utstyrt med en inkrementell enkoder, som gir tilbakemelding om dreiretningen og hastigheten på motoren. Motoren er koblet til frekvensomformeren med skjermet kabel og blir forsynt med 230V. Motoren er koblet i trekant, slik at fasespenning blir 230V.

Funksjon – Asynkron motor

En asynkronmotor er en kortslutningsmotor som drives av et roterende elektrisk felt, hvor rotoren roterer saktere enn det elektriske feltet. Jo mindre belastingen på motoren er, jo mer vil turtallet på rotoren nærme seg statorfeltets hastighet. Samtidig faller dreiemomentet rast mot null, slik at den aldri oppnår synkront turtall. Statoren er lik den i synkronmotoren og mates vanligvis med 3-fase spenning. Rotoren i den vanligste typen kortslutningsmotor har ingen elektriske kontakter og er derfor svært driftssikker. Rotorviklingen er erstattet av staver av kobber og aluminium som er kortsluttet i begge ender.

(Rosvold, 2017)

Funksjon – Inkrementell enkoder

Enkoderen som er montert på drivakselen på motoren, er en Baumer HTL inkrementell enkoder. Jobben til enkoderen er å gjøre om hastigheten på motoren om til elektriske pulser. Dette skjer ved at en gjennomsiktig kodeskive med noen mørke felter blir scannet av en elektrooptisk sensor. Hastigheten og rekkefølgen på de elektriske pulsene gir informasjon om motorens dreieretning og hastighet. For å finne dreiretningen på motoren, leser man A og B pulsen. Disse to pulsene følger hverandre, og man finner dreieretningen ved å se hvilken puls som kommer først og sist. Totalt gir enkoderen 1024 pulser pr omdreining. Det at enkoder er en HTL enkoder betyr at signalspenningen er lik forsyningspenning. Enkoderen er koblet til en terminalblokk på PLSen, som ved hjelp av forskjellige funksjonsblokker regner pulsene om til hastigheten på transportbåndet.

(Incremental encoder interface, 2017)

(Se vedlegg 5 – Elektrotegninger, tegningsnr: 1004 – Encoder og I/O power)

Kontrollpanel

Frekvensomformer

Gruppen har valgt å bruke denne frekvensomformeren fra Omron, 3G3MX2-V. Man styrer turtallet på motoren ved å endre frekvens og spenning, slik at man kan styre hastighet, akselerasjon og retardasjonstid på båndet. Dette var et viktig valg ettersom objektet skal stoppe innenfor et spesifikt søkeområde fra kamera. Motoren som frekvensomformeren skal drive er 0,25kW, så derfor ble det valgt ut en som har en makslast på 0,4kW.

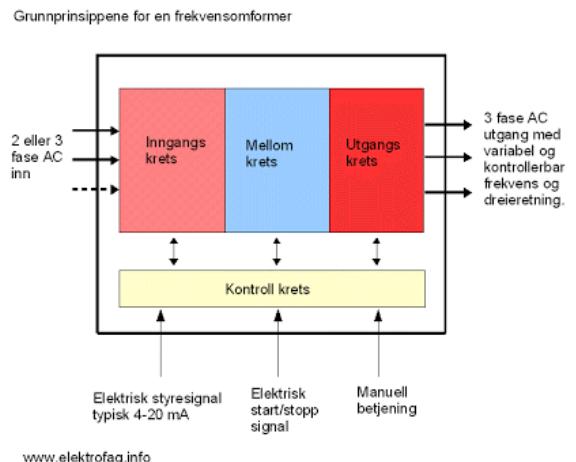


Bilde: 5 MX2 Frekvensomformer
(Omron, 2017)

Utover dette er frekvensomformer mye anvendt i industrien og man har mange muligheter for å gjøre justeringer i fremtiden hvis labøvelsen skal gjøres om. Som sikkerhet har MX2-V1 innebygd, (i henhold til ISO 13849-1, Cat. 3) med to sikkerhetsinnganger og en ekstern enhet Monitoring (EDM) utgang. Det trengs heller ingen eksterne kontaktorer på motorsiden, noe som betyr at det blir en enklere kabling.

Funksjon-frekvensomformer

Frekvensomformeren mottar en typisk drivspenning som kan være 2 eller 3 fase, fra nettet. Inngangskretsen oppgave er å likerette den inngående spenningen, til en likespenning og en likestrøm som leveres videre til mellomkretsen. Mellomkretsens oppgave er å kontrollere, stabilisere og glatte ut likespenningen som den mottar fra inngangskretsen. Utgangskretsens oppgave er å omdanne likespenningen fra mellomkretsen til en kontrollerbar tre fase utgangsspenning, der det er mulig å kontrollere både amplitude, frekvens og dreieretning. Kontrollkretsens oppgave er å styre og regulere alt det som skjer inne frekvensomformeren. Kontrollkretsen kan også motta elektriske styringssignaler fra PC, PLS og annet automasjonsutstyr, samt også et eget start/stopp signal og signal for dreieretning.



Bilde: 6 Prinsippet til frekvensomformeren
(Frekvensomformeren, 2017)

Oppkobling

Gruppen har koblet 1-fas fast spenning inn på inngang L1 og N. Ut fra U, V og W er det koplet til motoren. Det har også blitt koblet inn en sikkerhetsfunksjon via sikkerhetsreleet til frekvensomformeren. Når nødstoppbryteren blir slått ut brytes det en kontakt i sikkerhetsreleet som gjør at spenningen ut til motoren blir avslått, men det er fortsatt spenning på frekvensomformeren. Dette er for å ha kommunikasjon med frekvensomformeren selv om motoren er avslått.

(Se vedlegg 5 - Elektrotegning, tegningsnr: 1001 – Hovedstrøm)

(se vedlegg 10 – Utregning av retardasjonstid)

MX2 EtherCAT option card

For å kommunisere og styre frekvensomformeren fra PLSen har det blitt anvendt en MX2 EtherCat option card. Det er en enkel installasjon, man fjerner topplokket på frekvensomformeren og kobler til denne.

Den kommuniserer med det standard industrielle nettverket, EtherCat, CompoNet eller DeviceNet som alternativ. Med den høyhastighets EtherCat kommunikasjon har man muligheter for å koble dette opp mot Sysmac Series. Den har også en innebygd RS-485 Modbus-kommunikasjon.



Bilde: 7 MX2 EtherCAT option card
(Omron, 2017)

Elektromagnetisk støy

EMI, betegnelse på forstyrrelser og støy i elektronisk utstyr som skyldes elektromagnetisk påvirkning fra omgivelsene, enten denne kommer i form av radiobølger eller via ledninger. For å unngå elektromagnetisk forstyrrelser har gruppen gjort installasjonen støy-tett. For at installasjon skal være EMC-tett har frekvensomformeren fått et eget skap, hvor det har blitt montert et støyfilter blant annet. Et støyfilter er en elektronisk krets som kan skille ut og forsterke frekvensområde og undertrykke andre. En frekvensomformer lager mye støy fordi den opererer med bratte flanker og korte stige og falltider på spenningsnivå. Bratte flanker inneholder overharmoniske frekvenser (svært høye frekvenser). Denne støyen blir filtrert bort slik at den ikke kommer ut på forsyningsnettet.

Fra frekvensomformeren har det blitt benyttet en EMC-tett kabel, det vil si at det er en skjermet kabel, den har liten koblingsimpedans og oppfyller kravet i EMC-direktivet. Ved forlegningsmåte av kabel har det blitt tatt hensyn til avstand fra andre kabler, mht. støy og det har heller ikke blitt laget «pig tails» på kabelen. Ved beskyttelse ifb. med EMC så skal kabelen jordes 360 grader, dette er for å skape god kontakt for å unngå kapasitiv kobling. For å tilfredsstille dette kravet har det blitt benyttet en EMC-nippel med en krans av «tenner» som danner god elektrisk forbindelse mellom kabelskjermen og nippelen.

Strømforsyning

Ettersom flere av komponentene i prosjektet krever 24VDC, ble vi nødt til å transformere og likerette 230VAC til 24VDC. Får å få til dette ble det brukt to stk S8VK-S 06024 strømforsyninger fra Omron.

En strømforsyning en kraftforsyningsenhet som leverer elektrisk energi til en elektrisk belastning. Den valgte strømforsyningen består av en transformator som transformere spenningen fra 230V til 24V, og en likeretter som gjør om spenningen fra vekselspenning til like likespenning.

Transformator

En transformator en elektrisk maskin som overfører elektrisk energi fra et spenningsnivå til ett annet spenningsnivå gjennom elektromagnetisk induksjon. Transformatoren består vanligvis av to viklinger på en kjerne av lamellert jernblikk. Viklingene kalles primærvikling og sekundærvikling. Når man tilfører en varierende strøm i primærviklingen dannes det en varierende magnetisk fluks i jernkjernen. Dette magnetfeltet påvirker sekundærviklingen, og det varierende magnetfeltet i sekundærviklingen induserer en varierende spenning i sekundærviklingen. Ved å utnytte Faradays lov endrer transformatoren vekselstrøm fra ett spenningsnivå til ett annet. Forholdet mellom spenningen inn til primærvikling og ut fra sekundærviklingen avhenger av antallet viklinger.

(Høyte, 2016)

Likeretter

Den andre delen av strømforsyningen består av en likeretter som gjør om vekselspenning og vekselstrøm til likespenning og likestrøm. Likeretteren består av en kondensator som brukes til grovglatting av likespenningen. For videre glatting brukes det en transistorstabilisert strømforsyning, en regulert strømforsyning.

(Gunvaldsen, 2009)

(Se vedlegg 5 - Elektrotegning, tegningsnr: 1002 - Styrestørrelse)

PLS

Forkortelsen PLS står for «*Programmerbar logisk styring*» og er en fellesbetegnelse på datamaskiner som er spesielt utviklet for automatiserte prosesstyring. En PLS skiller seg fra vanlige datamaskiner på flere områder, den er beregnet for et bestemt formål(prosesstyring), den er beregnet for kontinuerlig drift og har høye krav til driftsstabilitet, mulighet for tilkobling av eksternt utstyr ved hjelp av inn- og utgangsmoduler og kunne programmeres med bestemte tilpassede programvaren.



Bilde: 8 Omron NX1P2 (Omron, 2017)

Gruppen valgte denne NX1P2 PLSen. Den er utstyrt med 23 innganger og 12 utganger, dette er nok I/O for dette prosjektet og det vil bli mange ledige I/O for senere oppkobling. Den har også mange muligheter for å bygges ut med flere terminalblokker og option boards.

(Se vedlegg 5 - Elektrotegning, tegningsnr: 1003 - PLS)

NX-PF0730

Dette er en skrueløs terminalblokk for power supply connection unit. Denne skal gi ekstra strøm til I/O kretsen i NX-I/O enhetene. Den tillater ekspansjon av I/O konfigurasjon over den maksimale strømkapasiteten på EtherCat-tilkoblingen. Den benyttes når den totale tillatte I/O strøm pr mateklemme er overskredet eller for å splitte I/O strøm inn i grupper. Inn/Utgangene kan brukes som en ekstra spenningsforsyningspunkt for tilkoblede sensorer og aktuatorer. Gruppen har koblet +24 og 0V til klemmene, deretter gir den strøm via pinnene videre i terminalrekken. Se neste punkt for hva som er videre i terminalrekken.



Bilde: 9 Power supply unit
(Omron, 2017)

(Se vedlegg 5 - Elektrotegning, tegningsnr: 1004-Encoder og I/O power)

NX-EC0122

Dette er en terminalblokk for inkrementel enkoder med PNP input. Denne blokken sitter ved siden av power uniten i rekken. Denne tar imot informasjon fra enkoderen og sender det til PLSen.

Gruppen har tilkoblet enkoder og den er klar for bruk, men i denne labøvelsen skal den ikke bli benyttet. Denne er for å kunne bruke pulsene fra enkoderen for tracking av deler senere. Tracking er når delene skal bli plukket i bevegelse, men ettersom delene skal stoppe ved et fast punkt skal gruppen ikke bruke denne.



NX-EC0122

Bilde: 10 Encoder option card
(Omron, 2017)

(Se vedlegg 5 - Elektrotegning, tegningsnr: 1004-Encoder og I/O power)

Programmering av PLS

Programmeringen av PLSen ble utført i en programvare som heter Sysmac studio. Denne programvare støtter den internasjonale standarden IEC 61161-3, som omhandler grunnleggende programvarearkitektur og programmeringsspråk for kontroll program innen PLS. Denne standarden definerer to grafiske programmeringsspråk ladder diagram (LD) og funksjonsblokkdiagram (FBD). I tillegg definerer den to tekstprogrammeringsspråk, strukturert tekst (ST) og instruksjonsliste (IL). I dette prosjektet har gruppen kun brukt LD og ST for programmering av de ulike funksjonene til PLSen. Selve programmet er delt inn i ett hovedprogram og flere underprogrammer for å gjøre det mer ryddig og oversiktlig. Det er blant annet laget et eget program for styring av transportbånd og ett eget sending og mottaking av informasjon fra roboten.

(Se vedlegg 11 - PLS-program)

Kabler

For å skille mellom spenninger på ledere, har gruppen gjort følgende:

230V=Sort RK 1,5mm²

24V=Rød RK 0,75mm²

Det er festet endehylser på alle ledningsender, dette er fordi det er RK og den har en flertrådet leder. Det er da for å gi god kontakt og sikrer at ingen kordeller skiller seg ut.

(Se vedlegg 12 – Utregning av kabler)

Kapsling og skap

Gruppen fikk et kontrollpanel fra Kongsberg Maritime som er blitt benyttet som styring av systemer for båter. Dette kontrollpanelet har fungert som et skap og tilfredsstiller kravene om kapsling, hvor det ikke er noen berøringsfare hvis man ikke åpner skapet. På kontrollpanelet har det blitt montert hjul, som gjør at det er flyttbart ved en endring eller flytting.

DIN-skinner

I kontrollpanelet har gruppen brukt DIN-skinner. DIN-skinner er en standardisert monteringsskinne for elektriske tavler og skap. Brukes for å feste sikringer, rekkeklemmer, kontaktorer, strømforsyninger etc.

Rekkeklemmer og merking

Rekkeklemmer brukes i elektriske tavler for å distribuere ledere, organisere merking, forenkle feilsøking, enkel tilkobling/frakobling av kabler. Alle kabler som går ut til eksterne komponenter er tilkoblet rekkeklemmer slik at gjør det lett hvis kontrollpanelet skal fjernes. Alle interne koblinger går også igjennom rekkeklemmer. Videre har hver eneste ledning blitt merket i kontrollpanelet og det har blitt utarbeidet en fullstendig rekkeklemmetabell.

(Se vedlegg 5 - Elektrotegning, tegningsnr: 1010 rekkeklemmetabell)

Kabelføringer

For å kunne legge kabler mellom de ulike komponentene i robotcellen har gruppen benyttet seg av gitterbaner, dette gir god oversiktighet og ryddighet. For å få et sikkert og ryddig ledningsnett i kontrollmodulen har alle ledninger blitt ført i kabelkanal.

Bryterpanel

Bryterpanelet består av en hovedbryter, nødstopp, start og stoppbryter. Forskriften setter egne krav til nødstoppbrytere, blant annet at de skal være røde. Slike brytere kreves på en del anlegg der det er viktig å få til raske utkoblinger ved farlige situasjoner. Pålegg om slike brytere er blant annet gitt i maskindirektivets forskrifter og anbefalinger, og i FEL og NEK 400 i direktivet fra EU. På nødstoppbryteren har gruppen benyttet tre kontaktsett, den går til ulike komponenter og har derfor ulik spenningspotensiale. Det ene kontaktsettet går til PLsen, dette er for å gi PLsen en tilbakemelding på at nødstoppen har slått ut. Den andre går til nødstopprelelet på robotkontrolleren og med dette stopper robotten også, hvis det skal oppstå en hendelse. Den siste er for sikkerhetsrelelet i kontrollpanelet, den sender videre et signal til frekvensomformeren at den skal stoppe.

Stoppbryteren er normalt i lukket tilstand. De åpner kontakten (bryter strømmen) når de blir betjent. Stoppbrytere med slik kontakt kalles brytere med hvilekontakt, og de er merket med 1 og 2 på tilkoblingene. Den har bare et kontaktsett og skal stoppe båndet. Den er koblet på inngang 01 på PLSen.

Startbryteren er normalt i åpen tilstand. De lukker når de blir betjent, og kontakten kalles arbeidskontrakt. Startbryteren har to kontaktsett og tilkoblingspunktene er merket med 3 og 4. Når man trykker på startbryteren sender den et signal på inngang 00 på PLSen om at prosessen kan starte, samtidig som den resetter sikkerhetsrelelet. Dette er lagt inn som en funksjon at hvis man trykker på nødstopp og slipper, skal ikke prosessen automatisk starte utenom at man har sagt «ok» og resatt sikkerhetsrelelet med startbryteren.

Hovedbryterens oppgave er å skru av/på strømmen til systemet. Dette er en vribryter med to posisjoner. Den skal kun brukes for å stenge av strømmen etter man er ferdig med en øvelse og ikke skal benytte systemet lengre.

(se vedlegg 5 – Elektrotegninger, tegningsnr: 1009 – Arrangementtegning)

Sikkerhetsrele

G9SE Safety relay unit er laget for DIN-skinne montering og har en driftsspenning på 24V. Sikkerhetsrelelet fungerer slik at den har en driftsspenning på A1 og A2 på spolen, i tillegg har nødstoppen og startknappen blitt koblet til. Det vil si at når nødstoppen har blitt slått ut, skal ikke transportbåndet kunne starte før den har blitt resatt av startknappen. Dette har med sikkerheten til operatøren. På det interne kontaktsettet (nr 13,14 og 23, 24) har det blitt koblet til klemmene på frekvensomformeren.



Bilde: 11 Sikkerhetsrele
(Omron, 2017)

(Se vedlegg 5 – Elektrotegninger, tegningngsnr: 1001 - Styrestørrelse)

Stasjonær PC

Får å kunne drive Sysmac Studio programvaren til Omrons PLS måtte gruppen ha en PC som var kompatibel med dette. Etter en prat med IKT ansvarlig på skolen fikk gruppen utdelt en HP med intel core i5 prosessor og et Windows 7 operativsystem. Denne stasjonære PCen skal brukes som en arbeidsplattform hvor man kan programmere vision-kameraet og PLSen.

Ettersom kontrollpanelet inneholder to skjermer og det fantes bare et skjermkort på PCen, har gruppen montert et ekstra skjermkort for å få skjermen splittet. Gruppen ønsket å gi labøvelsen et bedre omfang ved å kunne programmere kamera på den ene skjermen, mens bruksanvisningen er på den andre. Ettersom hva studenten ønsker å gjøre kan man ha full oversikt på labøvelsen gjennom å benytte seg av to skjermer.

Kommunikasjon

NX1W-CIF01

Dette er en RS-232 seriell kommunikasjons option board. Denne er tilkoblet i fronten på PLSen. Etter samtale med Scala Tech anbefalte de at gruppen skulle benytte seg av RS-232 kommunikasjon. Bakgrunnen for dette valget var fordi roboten allerede var klargjort for denne type kommunikasjon, noe som viste seg å være feil. Denne kommunikasjonsformen er «gammeldags» og krever mer programmering enn Ethernet. Gruppen prøvde å få til dette, men det viste seg at gruppen har fått feil informasjon fra leverandør og dette vil ta mer tid enn planlagt. Det ble derfor besluttet Ethernet isteden. Det er lagt opp kabler fra rekkeklemmer og inn på kortet.



Bilde: 12 RS-232 Communication unit (Omron, 2017)

(Se vedlegg 5 - Elektrotegning, tegningsnr: 1006-RS232 kommunikasjon)

Ethernet kommunikasjon

Gruppen fant ut at NX-100 kontrolleren var utstyrt med et ethernetkort som var kompatibel med PLSSens grensesnitt. Gruppen valgte derfor å anvende Ethernet kommunikasjon. Dette medførte mindre programmering og raskere kommunikasjon mellom enhetene. Gruppen la deretter en RJ45 kabel fra switchen til ethernetkortet på roboten. NX-100 kontrolleren er satt opp med en egen IP-adresse hvor data fra PLSSen blir sendt via Ethernet switchen.

Ethernet switch

SDI-550 er en enkel Ethernet switch egnet for bruk i fabrikk og industrielle automasjonsapplikasjoner. Det er en høyhastighets switch som sørger for full fart på alle porter og transparents til enhver Ethernet-protokoll. Enheten er konstruert for DIN-skinne montering og det er etablert Ethernet-forbindelser mellom PLS.

I/O-enheter, PC, vision-sensor og NX-100 kontrolleren. Den har en inngangsspenning på 24V DC som er koblet til trafoen. For å kommunisere med PLS, PC, vision-sensoren og NX-100 kontrolleren, har det blitt brukt patchekabler av typen RJ45. Dette er en standard EtherCat kommunikasjon.



Bilde: 13 Ethernet switch
(Ethernet switch, 2017)

Labøvelsen

Hovedoppgaven i labøvelsen vil være programmering av vision-sensoren. Ideen er da at studentene skal kunne bruke den vedlagte brukermanualen for labøvelsen til å programmere vision-sensoren og på denne måten få et innblikk i oppbygningen og virkemåten til systemet. Ettersom at hensikten med labøvelsen er å gi studentene innblikk i vision-system, vil PLSSen være forhåndsprogrammert.

Ved å tilrettelegge labøvelsen på denne måten vil studentene slippe på å bruke mye tid på PLS-programmering, som i tillegg er svært krevende. Etter at vision-sensoren er programmert vil studentene kunne åpne det forhåndsprogrammerte PLS-programmet for å monitorere prosessen. Selv om PLSSen er forhåndsprogrammert vil studenten allikevel få et innblikk i oppbygningen og logikken bak PLS-programmet. Ettersom kontrollpanelet er utstyrt med to skjermer, vil man kunne monitorere PLS-programmet på den ene, og ha live-feed fra vision-sensoren på den andre.

(Se vedlegg 7 - Bruksanvisning for labøvelsen)

1.5: Drøfting/diskusjon

I 1.4 resultat, valgte gruppen og kategorisere de ulike systemene i vision-sensor, infrastruktur, robot, kommunikasjon og kontrollpanel. Gruppen drøfter derfor temaene hver for seg.

Vision-sensor

Vision-sensoren gruppen valgte å bruke i dette prosjektet har vist seg å fungere bra til formålet. Før valget at sensor ble tatt, ble sett på en rekke ulike sensorløsninger fra flere forskjellige leverandører. De ulike sensorløsningene ble drøftet, og 2D-sensor og 3D-sensor ble veid opp mot hverandre. Det ble etter hvert klart at en 2D-sensor ville inneholde det nødvendige funksjonene får å kunne løse oppgaven. I og med at sensoren skal brukes i skolesammenheng og fungere som en opplæringsplattform, var det også viktig at programvaren til sensoren ikke var for omfattende og krevende å bruke. Dette var naturligvis en viktig faktor som det ble tatt hensyn til ved valg av sensor.

Vision-sensoren brukes til å utføre kvalitetskontroll av plukkobjektene og den sender posisjonsinformasjonen videre til roboten. Dette var også et punkt fra oppdragsgiver, som er beskrevet i avgrensingen av problemstillingen. Vision-sensoren er montert på et justerbart oppheng i enden av transportbåndet. Gruppen ser at det kan oppstå et problem dersom opphenget blir dultet borti eller forskjøvet på noe måte, slik at sensoren flytter seg. Dersom dette skjer, vil ikke koordinatsystemet til vision-sensorer og roboten stemme overens lengere. Da må vision-sensoren kalibreres og stilles inn på nytt. Samtidig har gruppen gjort tiltak for at dette ikke skal skje, ved å montere opphenget på den siden av transportbåndet hvor personer ikke ferdes.

Programvaren til vision-sensoren har et enkelt og oversiktlig brukergrensesnitt, som består av ulike menyer, knapper og nedtrekkslister. Her kan studentene ved hjelp av brukermanualen for labøvelsen gå inn i ulike menyene og gjøre seg kjent med de ulike parameterne og innstillingene for sensoren. I brukermanualen for labøvelsen har gruppen prøvd å legge til rette slik at studentene skal få et godt innblikk oppbygningen til vision-systemet, og hvordan det samhandler med roboten. Programvaren gir også mulighet for monitorering, og på den måten kan studentene overvåke prosessen for å se hva som skjer underveis i prosessen

Infrastruktur

Valget om å benytte seg av et transportbånd som transporterer objekter inn i robotcellen, mener gruppen har vært et godt valg. Dette gir både labøvelsen og prosjektet en større kompleksitet. Ved å bruke transportbånd, får studentene som skal gjøre labøvelsen en bedre forståelse av hvordan hele prosessen foregår i industrien. Transportbåndet er også tilpasset i cellen. Den er tilpasset slik at man kan stå på utsiden av cellen, legger på objekter og objektene transportereres til robotens plukkepunkt. Hvis man sammenlikner dette konseptet med konsept 2, hvor delene blir plassert i en sklie, får man ikke samme læringsutbytte.

Gruppen og oppdragsgiver har også tenkt langsiktig og derfor kjøpt en enkoder som er tilpasset motoren på transportbåndet. Den skal kunne brukes for tracking ved en senere anledning.

Robot

Ettersom at labøvelsen, infrastruktur og vision-system skulle tilpasses FTOs Motoman robot, har ikke gruppen hatt mye spillerom når kommer til å gjøre endringer på roboten. Gruppen hadde allikevel mulighet til å gjøre forandringer på griperverktøyet, dersom det ble funnet nødvendig. Det ble sett på forskjellige griperverktøy og hvordan plukkobjektene skulle utformes. Gruppen fant tidlig i prosjektprosessen ut at det var ønskelig å beholde griperverktøyet, og heller tilpasse plukkobjektene. På denne måten slapp gruppen å bruke tid på å integrere ett nytt griperverktøy, som også ville medført en ekstra kostnad for prosjektet.

Kommunikasjon

Kommunikasjon er noe som ikke har vært helt enkelt i dette prosjektet. Ingen på gruppen har hatt noe erfaring med oppsett av protokoll-løs kommunikasjon tidligere, og overvåkningen av denne kommunikasjonen har ikke vært lett å gjennomføre.

Leverandøren av roboten mente at det var hensiktsmessig å enten kommunisere via RS232, eller kjøpe inn et utvidelseskort til roboten som gjorde at gruppen kunne kommunisere med en enklere ethernet-protokoll. Gruppen valgte her billigste løsning for prosjektet, og gikk til innkjøp av et RS232 utvidelseskort til PLSen. Gruppen brukte en USB til RS232 overgang for å verifisere at tilkoblingen var satt opp rett. Etter litt om og men, ble det verifisert at data ble sendt og mottatt.

Deretter ble det forsøkt å implementere dette mot roboten. På roboten var det ikke mulig å overvåke hva som ble sendt og mottatt, og gruppen greide ikke å få til noe kommunikasjon her. Etter en prat med leverandøren av roboten, fikk gruppen beskjed om at den innebygde serielle kommunikasjonen kanskje opererte på en lukket standard, og kunne derfor være utilgjengelig for denne typen protokoll-løs kommunikasjon. Leverandøren sa også at det var mulig vi kunne benytte oss av den innebygde ethernet-serveren til robot-kontrolleren.

Ethernet-serveren til roboten har mulighet til protokoll-løs kommunikasjon, på lik linje som gruppen ønsket ut av RS232 metoden. Med denne kommunikasjonen var det også enkelt å verifisere at kommunikasjonen virket som den skulle ved å benytte en PC som mottaker fra PLSen. Etter en stund greide gruppen å verifisere at meldingen som ble sendt var etter robotens krav, og fikk skrevet og lest data til og fra robotens variabler.

Meldingene som blir sendt fra PLSen til roboten består av 8-bits karakterer, og følger ASCII-standarden. For at roboten skal kunne motta en melding på riktig måte, må syntaxen på meldingen være konstruert etter kravene som er spesifisert i manualen, samt lengden på meldingen må bestå av riktig mengde tegn. Dette gjør at det meldingene som skal sendes er vanskelige å standardisere, og funksjoner må derfor programmeres hver for seg i PLSen.

Om gruppen hadde gått for den andre løsningen som ble presentert av leverandøren tidligere i prosjektet, og kjøpt inn et ethernet-utvidelseskort til roboten, hadde kommunikasjonen vært mye lettere å gjennomføre. Tidmessig hadde dette også spart gruppen for veldig mye feilsøking på kommunikasjonsstandarder de har jobbet veldig lite med. Her hadde nok tidsbruken veid opp for kosten av utstyret veldig fort.

Kontrollpanel

For å kunne koble sammen alle komponenter, må man ha et el.skap. Den opprinnelige planen til gruppen var å kjøpe inn et el.skap hvor alle komponentene skulle bli montert, og skapet skulle bli montert på en vegg. Etter en prat med Tommy Hvidsten fikk gruppen benytte seg av en kontrollmodul fra Kongsberg Maritime som stod på Fagskolen. Dette var en kontrollmodul med to skjermer og tilstrekkelig skaplass.

Gruppen monterte deretter hjul på den, dette var en stor fordel ettersom den skulle kunne flyttes ved en eventuell endring. Alt av kabling som går inn og ut, er koblet via rekkeklemmer som er merket i henhold til tegninger og skjematikk som har blitt laget. Dette gjør det enkelt å koble ifra, flytte eller gjøre endringer til neste år. Kontrollpanelet tilfredsstiller dessuten kravet om kapsling og berøringsfare. Utstyret som står i skapet har blitt montert på DIN-skinner og gjør det lett for eventuell demontering.

1.6: Konklusjon/oppsummering:

For å oppsummere prosjektet må vi ta et tilbakeblikk på de kravene som har blitt satt

Produktets funksjon

Gruppen har integrert et vision-system med Fagskolens Motoman Sia-20d. Ved hjelp av en vision-sensor blir bilder brukt til gjenkjennelse og posisjonsinformasjon av objekter, deretter griper roboten plukkobjekter basert på denne informasjonen. Kvaliteten på produktet kan konkluderes med at har vært bra. Dette kommer av at gruppen har fulgt retningslinjene fra oppdragsgiver, arbeidet har vært i henhold til produktbeskrivelser fra leverandører og det har blitt tatt hensyn til lover og regler.

Systemet kan anvendes til flere oppgaver

Hele systemet er satt opp for å gjennomføre en oppgave fra PLSen side. Dette gjør at alt som forgår via PLSen kun vil fungere til denne ene oppgaven, altså sortering og plukking av deler. Transportbåndet og vision-sensoren styres via PLSen, og vil derfor være utelukket til å gjennomføre andre øvelser. Roboten kan derimot brukes til flere oppgaver, men den må programmeres til faste koordinater istedenfor tilfeldige som i laboppgaven gruppen har konstruert.

Et forbedringspunkt for dette kunne vært å ha en vribryter som stilte PLSen inn i forskjellige moduser, i henhold til hvilken laboppgave som man ønsker å gjennomføre. For å programmere inn dette må man også ha en plan på hva de forskjellige modusene skal innebære, og det har falt utenfor denne prosjektperiodens omfang.

Økonomi

Det var ikke fastsatt noe spesifikt budsjett for dette prosjektet. Oppdragsgiver kunne skaffe mellom 100-200.000kr hvis konseptet var bra. Gruppen har underveis hatt en åpen dialog med oppdragsgiver hvor innkjøpene har blitt godkjent underveis. Det økonomiske rammene for prosjektet har blitt holdt.

(Vedlegg 13 – Reelt budsjett)

Labøvelse

Gruppen har produsert en labøvelse som er basert på tidligere labøvelser fra robotteknologikompendiet, og tar for seg samspillet mellom vision-system og robot. Som tidligere nevnt baserer labøvelsen seg på en «pick and place» oppgave. Gruppen mener at labøvelsen har klare likhetstrekk med hvordan slike systemer anvendes i industrien, og at det på den måten gjør labøvelsen interessant og lærerik. Labøvelsen inneholder både programmering av robot og vision-sensor. Labøvelsen er tilrettelagt med en brukermanual, som beskriver hvordan programmeringen skal gjøres.

Dokumentasjon

Gjennom prosjektprosessen har det blitt utarbeidet komplett dokumentasjon av labøvelsen. Dette innebærer tegninger og skjematikk av det elektriske anlegget, PLS-programmet, robotprogrammet samt brukermanual.

Gruppen konkluderer dermed med at prosjektet har vært en suksess.

Del 2: Prosessen

2.1: Referat fra arbeidet.

En kronologisk gjennomgang av arbeidet som er gjort fra begynnelse til slutt.

Oppstartsfasen

Punkter som var viktig under oppstartsfasen:

- Her ble det laget en samarbeidskontrakt, og det ble avklart hvilke roller hvert enkelt gruppemedlem skulle ha gjennom prosjektet.
- Opprettet fellesmappe for prosjektet.
- Målet for prosjektet: *Lage en labøvelse som inneholder samspill mellom robot og et vision-system.*
- Viktige datoer som oppstartdato, sluttdato og milepæler.
- Prosjektets bakgrunn, hvor kommer det fra, og hvorfor har vi valgt å jobbe med dette?
- Prosjektets begrensinger i form av økonomi og ressurser ble vurdert.

Forprosjektfase

- Tidsplan: I denne fasen ble det utarbeidet en fremdriftsplan for hele prosjektet, hvor prosjektet delt inn i aktiviteter og gjøremål.
- Produktspesifikasjon: Hvordan skal labøvelsen fungere konkret. Krav fra oppdragsgiver og økonomi knyttet til dette.
- Konseptutvikling: Utvikling av totalt seks forskjellige konsepter med tilhørende prisestimat. Det innebar også innhenting av tilbud fra forskjellige leverandører
- Valg av konsept: Her ble det i samråd med oppdragsgiver foretatt et dokumentert valg av konsept og leverandør.
- Prosjektets omfang: Dokumenterer hvilke fagområder som berøres gjennom prosjektarbeidet.
- Organisasjonsoversikt: Det ble laget en oversikt over hvordan prosjektorganisasjonen ville se ut.
- Ressurser: Kartlegging av ressursbehovet. Hvilke ressurser har vi, og hvilke ressurser trenger vi for å nå målsettingen.
- Risikoanalyse: Kartlegging av spesifikke risikoer som kunne gjøre at prosjektet mislykkes.

Oppstart produksjon

Starten av produksjonsfasen dreide seg om følgene punkter:

- Bestillinger: Bestilling av komponenter, samt innkjøp av forbruksmateriell.
- Dokumentasjon: Utarbeiding av skjemategninger av det elektriske og pneumatiske anlegget.
- Oppstart rapport: Hovedprosjektrapporten ble påbegynt.
- Klargjøring til produksjon: Installere programvare og legge opp kabelgater.
- Kurs: Opplæring i Omrons programvare.
- Utregninger: Utregning av kabler og retardasjonstid.

Produksjonsfase

Produksjonsfasen var den største og mest omfattende fase i dette prosjektet. Denne fasen inneholdt følgende:

- Oppkobling: Montering og oppkobling av komponenter.
- Integrering: Integrere nytt utstyr med eksisterende utstyr.
- Merking: Merking av kabler og komponenter.
- Kommunikasjon: Opprette kommunikasjon mellom de ulike komponentene.
- Programmering: Programmering av PLS, vision-sensor, frekvensomformer og robot.
- Plukkobjekter: Produksjon av plukkobjektene.
- Rapport: Arbeid med hovedprosjekt rapport.

Dokumenteringsfase

- Slutt-test av labøvelsen.
- Kvalitetssikring: Kvalitetssikre kabling, merking, og dokumentasjon.
- Rapport: Ferdigstilling av hovedprosjektrapport.
- Utarbeide dokumentasjonsgrunnlag for labøvelsen.

Overleveringsfase

- Levere hovedprosjektrapport.
- Prosjektfremføring

(se vedlegg 12 – Opprinnelig fremdriftsplan)

(se vedlegg 13 – Reell fremdriftsplan)

2.2: Planlegging:

2.2.1:Tidsbruk

Styringsverktøy

Ved prosjektets oppstart ble det i felleskap utarbeidet en grov oversikt over de aktivitetene som måtte gjennomføres, disse aktivitetene kaltes hovedaktiviteter. Etter at hovedaktivitetene var satt opp, begynte gruppen å se på hver aktivitet i detalj. Disse aktivitetene ble kalt for underaktiviteter og var med på å gi en mer presis antagelse av tidsbruk for prosjektet. Deretter ble alle aktivitetene satt opp i projectLibre, som er et prosjektstyringsverktøy. Her ble alle aktivitetene satt opp med forventet tidsbruk, frist og eventuelle avhengigheter. Fremdriftsplanen ble også brukt til å sette opp de forskjellige milepælene, både de som var pålagt av skolen, og våre egne. Disse milepælene ble også ført inn i ett tidslinjeskjema. I planleggingsprosessen ble det forsøkt å ta hensyn til ferier og travle eksamensperioder.

I planleggingsfasen ble det besluttet at alt av arbeidstimer skulle registreres i ett eget skjema. Dette skjemaet inneholdt ett felt som beskrev hvilken arbeidsoppgave som var jobbet med, og antall timer. På denne måten fikk vi en oversikt over antall nedlagte arbeidstimer, og vi kunne gå tilbake til fremdriftsplanen å se om de estimerte timene stemte overens med det som ble gjort.

Møter

Gjennom prosjektet har det blitt holdt mange møter, både formelle og uformelle. I og med at alle prosjektdeltakerne kommer fra samme klasse, og bruker mye tid sammen i løpet av en arbeidsuke, har prosjektet blitt implementert i våre uformelle samtaler. Det har blitt diskutert og snakket mye om prosjektet, både innad og utad i gruppen i pauser og andre sammenhenger. Dersom det ble tatt noen beslutninger i disse uformelle møtene, ble de skrevet ned og publisert på fellesmappen. Gjennom prosjektet har gruppen så godt det lot seg gjøre prøvd å holde ukentlige møter. Mot slutten av prosjektet gikk disse møtene mer over til å lage gjøremålslister for gjenstående arbeid.

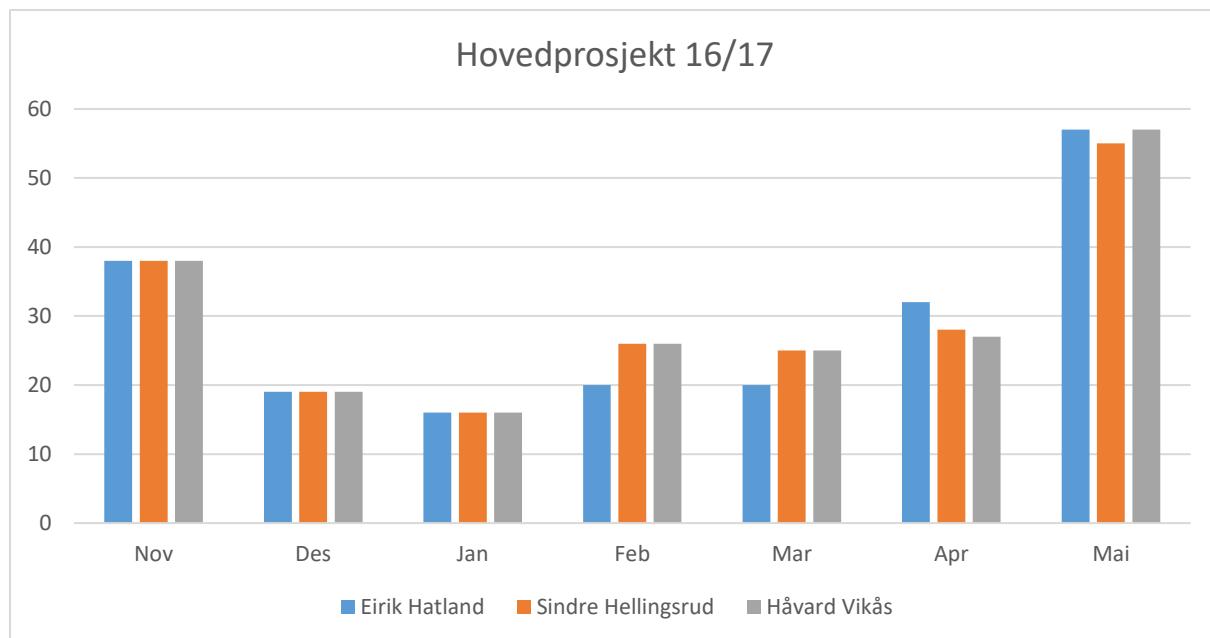
(se vedlegg 16 – Møtereferater)

Endringer

Etter hvert som arbeidet med prosjektet startet, oppdaget gruppen at det ble nødvendig å gjøre endringer i fremdriftsplanen. Bakgrunnen for det var at noen av aktivitetene var estimert med for få timer, og noen hadde for mye. Blant annet oppdaget vi at prosessen med innhenting av tilbud fra forskjellige leverandører var en tidkrevende aktivitet, og at det hadde blitt tildelt for lite timer. Dette oppdaget gruppen ganske tidlig og det ble besluttet å flytte enkelte aktiviteter slik at aktiviteten ble gjennomført innen tidsfristen. Disse endringene ble dokumentert i en kopi av den originale framdriftsplanen, som på mange måter var et levende dokument gjennom hele prosjektperioden.

Konklusjon

Når gruppen startet på prosjektet var det en del nye og ukjente momenter gruppen måtte ta hensyn til i planleggingsprosessen. Det var spesielt vanskelig å estimere timer for de arbeidsoppgavene som var knyttet til vision-systemet, ettersom ingen av gruppemedlemmene hadde noe erfaring fra dette fagområdet. Det gruppen erfarte var at programmeringen av systemet var en svært tidkrevende prosess, og at gruppen hadde planlagt for lite timer til denne arbeidsoppgaven. Heldigvis oppdaget gruppen dette relativt tidlig, slik at det ble justert i fremdriftsplanen, og slik at denne aktiviteten kunne begynne tidligere. Alt i alt kan gruppen konkludere med at den tilgjengelige tiden har blitt brukt på fornuftig måte.



Bilde: 14 Timeforbruk (SHE.engineering, 2017)

Dette stolpediagrammet viser hvordan timeforbruket har vært gjennom prosjektperioden. Den viser kun de timene gruppen har jobbe sammen, og derfor vil det reelle timeforbruket være noe høyere enn det diagrammet viser.

2.2.2: Plan for arbeidet

De forskjellige arbeidsoppgavene under prosjektperioden ble fordelt mellom de ulike gruppemedlemmene. Noen arbeidsoppgaver ble utført i fellesskap, noen på egenhånd og noen to og to. Hvilke arbeidsoppgaver som skulle gjøres ble tatt opp på morgenmøte. Oppgavene ble som regel tildelt den som hadde mest erfaring på fagområdet eller etter eget ønske fra medlemmene. Det var generelt et ønske fra alle medlemmene å være involvert i arbeidsoppgaver som bidro til ny læring og ny kunnskap i prosjektet.

Disposition

Når det kommer til rekkefølgen oppgavene har blitt utørt i, har dette prosjektet i utgangspunktet hatt en naturlig gang. Den ene aktiviteten må være ferdig før den neste kan startes, noen aktiviteter har kunnet forgå parallelt. Gruppen har underveis fulgt den framdriftsplanen som ble utarbeidet på forhånd, men det har som nevnt blitt gjort noen endringer underveis.

Prioriteringer

Hvilke arbeidsoppgaver som ble prioritert først og sist var ofte en diskusjon på møtene gjennom prosjektfasen. Her var det ofte mange hensyn som måtte tas, for eksempel leveringstid på komponenter, veiledning fra eksterne leverandører, tilgang til verktøy. Gruppen prioriterte blant annet å ferdigstille oppkobling av komponenter samt programmere RS-232 kommunikasjonen fra PLSen for å vite om man får kommunisert med en PC før gruppen pratet med Scala Tech. Det ble samtidig jobbet parallelt med rapport, referat og andre administrative oppgaver

Konklusjon

Når man ser tilbake på prosjektperioden, ser man at det er ting som har fungert bra og ting som ikke har vært like vellykkede. Det som har fungert bra er arbeidsfordelingen mellom gruppemedlemmene. Alle har gjort det de har blitt tildelt og alle har stått på for å bli ferdig med sine oppgaver. Prosjektets fremgang har vært avhengig av at gruppemedlemmene har levert det de skal til tide. I tillegg har alle vært flinke til å hjelpe hverandre hvis noen har hatt problemer eller stått fast.

Det at noen arbeidsoppgaver har blitt forskjøvet frem til slutten av prosjektperioden har ikke vært helt heldig. Det har gått utover den planlagte slakken som skulle brukes til rapportskriving, slutt-test av produkt og øve til presentasjonen. Grunnen til at aktiviteten «programmering av robot» har blitt forskjøvet frem var at det tok lengre tid enn antatt å programmere vision-kamera og PLS. Programmering av vision-kamera og PLS henger sammen for å kunne sende informasjon til robot for å programmere den senere.

2.2.3: Gruppeledelse

Ved oppstarten av prosjektet ble det i fellesskap utarbeidet en samarbeidskontrakt.

Denne var bindene for alle medlemmene gjennom hele prosjektfasen og tok for seg kommunikasjon, roller og ansvarsområde, beslutningsprosessen, regler og normer.

Kontrakten beskrev også hvilke sanksjoner som skulle tas i bruk ved eventuelle brudd.

Roller

- Eirik Hatland - Prosjektleader/gruppeleder
- Håvard Vikås - Sekretær
- Sindre Hellingsrud - Logistikkansvarlig

Eirik Hatland ble valgt til gruppeleder/prosjektleder fordi han var den med mest erfaring av gruppemedlemmene både når det kommer til alder og forskjellige yrkesområder. Han hadde ansvar for kommunikasjonen med oppdragsgiver, generell oppfølging av prosjektet, at fremdriftsplanen ble fulgt og at medlemmene gjorde sine arbeidsoppgaver.

Håvard Vikås ble valgt til sekretær og referent. Han hadde ansvar for møtereferater, endringer i fremdriftsplan, sakslogg og dokumenter som var relevant til prosjektet. Sekretæren var også ansvarlig for at alle medlemmene hadde den informasjonen de trengte samt at filer ble opplastet til prosjektmappen på It'sLearning.

Sindre Hellingsrud ble valgt til logistikkansvarlig. Ansvarsområdene hans var innkjøp, materialflyten, komponentliste og budsjett. Logistikkansvarlig var også ansvarlig for at gruppen hadde tilgang på de forskjellige dataprogrammene som ble brukt gjennom prosjektet.

(se vedlegg 17 – Samarbeidskontrakt)

Struktur

Selv om gruppen har hatt en valgt leder, har gruppen opptrådd med flat lederstruktur i gruppa. Samtlige medlemmer har vært selvgående gjennom prosjektet, og derfor har det ikke vært noe behov for en markert leder. Hvert gruppemedlem har hatt ansvar for sitt område, og alle har bidratt på lik linje ved de avgjørelser som har blitt tatt. Det har ikke vært noen store uenigheter innad i gruppa underveis i prosjektet, og heller ikke noen som har utrykt misnøye med hvordan ting har blitt løst underveis i prosjektet. Alle har sagt sin mening under møter og ved diskusjoner av valg som skulle tas. Gruppen har hatt en åpen og ærlig kommunikasjon gjennom prosjektet, noe gruppen tror er nøkkelen til at det har vært lite uenigheter og konflikter i en hektisk periode. Konklusjonen etter at prosjektet er ferdig er at rollene som beskrevet i samarbeidskontrakten har fungert.

2.3: Kommunikasjon

Kommunikasjonen mellom gruppemedlemmene har foregått på en åpen og ærlig måte, der alle har fått kommet med innspill og egne meninger. Gjennom prosjektet har gruppen hatt ukentlige arbeidsmøter som har inneholdt følgende:

- Status for fremdriftsplan.
- Endringer i prosjektet.
- Rapportering til prosjektleder om status på arbeidsoppgaver/aktiviteter og tid til ferdigstilling.
- Fordeling av nye arbeidsoppgaver.
- Øvrig informasjon om prosjektet.

På disse møtene ble det utnevnt en referent og det ble skrevet møtereferat. Gruppen har benyttet flere verktøy for kommunikasjon for å holde hverandre oppdatert på prosjektet. Det ble opprettet en fellesmappe for prosjektet på dropbox, der alt av dokumentasjon ble lagret og oppbevart. Det ble også opprettet en felles mail for prosjektet, som alle hadde tilgang til. Denne mailen ble brukt til å kommunisere med leverandører og andre eksterne som hadde betydning for prosjektet.

Den daglige skriftlige kommunikasjonen mellom gruppemedlemmene foregikk på en lukket gruppesamtale på facebook, noe som er fint verktøy for å kommunisere med flere personer samtidig og for å holde hverandre oppdatert på alle områder.

Her ble beskjeder, gjøremålslister og informasjon om prosjektet delt med alle gruppemedlemmene samtidig. Fordelen med å kommuniserer på denne måten er at alt blir loggført, slik at man kan gå tilbake å se hvilke beskjeder som ble gitt og hva som ikke ble gitt. Gjennom en arbeidsuke foregikk mye av kommunikasjonen muntlig, og dersom det var noe viktig informasjon eller beskjeder ble det skrevet ned og lagt på fellesmappen.

Gjennom prosjektet oppsto det flere situasjoner der gruppen hadde behov for å kommunisere med leverandører for å finne svar på tekniske spørsmål. I starten av prosjektet fikk gruppen telefon nummer og e-post adressen til teknisk personell hos de ulike leverandørene, som kunne bistå oss hvis gruppen sto fast. Denne kommunikasjonen foregikk på en formell måte og hadde et høyt faglig nivå. I enkelte brukte situasjoner ble det brukt teamview, slik at det tekniske personellet kunne fjernstyre datamaskinen. Dette viste å være til stor hjelp, og bydde på mindre problem enn at ting ble forklart over telefon og e-post.

Kommunikasjonen med veileder har foregått på en faglig og formell måte. Dersom det var arbeidsoppgaver og gjøremål som krevde kompetanse som gruppen ikke hadde, kontaktet vi prosjektets veileder.

De metodene gruppen har brukt for å kommunisere intern og eksternt har vist seg å fungere bra gjennom prosjektet, da det ikke har oppstått kommunikasjonsproblemer underveis.

2.4: Ressurser

Faglig ressurser

For å få fullført prosjektet var gruppen avhengig av en hel del faglig relatert kompetanse. Under prosjektperioden innså gruppen at den ikke hadde all den kompetansen gruppen måtte ha for å løse alle problemene, så gruppen har også benyttet mye hjelp og veiledning fra eksterne ressurser.

Økonomiske ressurser

Det har ikke vært fastsatt et budsjett, men oppdragsgiver har kunne skaffe en størrelsesorden mellom 100 og 200 000kr hvis konseptet var bra. Gruppen har holdt kostnadene langt under dette.

Interne ressurser:

Gruppen har internt hatt ressurser i form av relevante fagbrev, der en var industrimekaniker, elektriker og automatiker. Alle har fått bruk for sine kunnskaper, og alle har tilegnet seg nye i løpet av prosjektet. Selv om ingen av gruppemedlemmene har jobbet med verken robot eller vision-system, har gruppen klart og løse problemene som oppstod underveis.

Gruppens interne ressurser:

Eirik Hatland

26år

Elektriker

Håvard Vikås

23 år

Industrimekaniker

Sindre Hellingsrud

23 år

Automatiker

Eksterne ressurser:

Gruppen har vært avhengig å få hjelp av eksterne leverandører under prosjektperioden. Under oppstart av prosjektet fikk gruppen en introduksjon til robotprogrammering av Scala Tech da de skulle kjøre service på roboten, underveis har gruppen hatt kontakt for hjelp til kommunikasjon mellom komponentene. Omron har også hjulpet oss med grunnleggende Sysmac Studio kunnskap, vision-programmering og hjelp til programmering av PLS. Da de har hjulpet oss har gruppen benyttet seg av TeamView og telefonkontakt. Det har ikke bestandig vært lett å få tak i leverandørene, det har vært en del ventetid. Selv om det har vært bestemt tidspunkt for hjelp, ser man i etterkant at man kunne ha vært enda mer tydelig når de skal hjelpe oss. Det har ikke alltid vært slik at de har vært ledig.

2.5: Læringsutbytte

Gruppen har under prosjektperioden hatt et stort læringsutbytte på flere områder. Gruppen har fått en utdypende læring om emner gruppen har vært innom i ulike fag på fagskolen. Gruppen har også vært innom flere områder som har vært nye og ukjente, dette har gjort at gruppen har måtte tilegne seg kunnskap underveis. På enkelte områder har gruppen vært nødt til å bruke andre ressurser for å få tilegnet den kunnskapen og kompetansen som har vært nødvendig for å komme i mål. Dette har vært både interessant og spennende.

Læring med utgangspunkt i problemstilling:

- **Vision system**

Gruppen har måtte tilegne seg kunnskap i hvordan vision-system fungerer.

Det har inneholdt kameraets funksjonalitet, både begrensinger og muligheter, programmering av kamera, oppkobling, optikk og lysforhold.

- **Robot**

Her har gruppen måtte sette seg inn i hvordan en robot fungerer, posisjonering og programmering i tillegg griperverktøyet.

- **Samspill mellom robot og vision-system**

Gruppen har satt seg inn i PLS systemer og kommunikasjon mellom disse systemene.

- **Økonomi**

Gruppen har innhentet tilbud fra flere leverandører og vurdert disse. Gruppen har også satt opp et budsjett og holdt seg til dette.

- **Dokumentasjon**

Gruppen har utarbeidet teknisk tegningsunderlag og laget en brukermanual for prosjektet.

Læring med utgangspunkt i læreplan:

Yrkesrettet kommunikasjon

- **Mål:** Ha kunnskap om språket som verktøy for god kommunikasjon og kjenner til norsk og engelsk fagterminologi innen sine fagområder, relevant dataverktøy som benyttes ved kommunikasjon samt ulike sosiale medier, prosjektdokumentasjon, avtaler og kontrakter.
- Gruppen har fått mye læring i å bruke skriftlig og muntlig kommunikasjon på en riktig og effektiv måte, samt at gruppen har fått bedre forståelse og større kunnskap om bruk av det skriftlige språket som er nødvendig i et prosjekt. For eksempel formell kommunikasjon med leverandører, engelske brukermanualer, rapporter, møtereferater, arbeidsavtaler, kontrakter og fremføring.

Fysikk

- **Mål:** Ha tilstrekkelig grunnlag i realfag til å utføre nødvendige beregninger, dimensjoneringer og annen problemløsning med utgangspunkt i relevante praktiske situasjoner.
- Gruppen har anvendt fysikk for utregning av akselerasjon, retardasjons og hastighet på transportbåndet.

LØM

- **Mål:** Ha kunnskap om organisasjonsteori, organisasjonskultur, ledelsesteori, motivasjonsteori, aktuelle lover, arbeidsbetingelser, økonomibegreper, budsjettering, kjøpsatferd og markedsplanlegging.
- Gruppen har forstått budsjettering og kan bruke denne informasjonen til å iverksette tiltak. Gruppen kan gjøre rede for og vurdere menneskelige, arbeidsmiljømessige, etiske og økonomiske utfordringer etter gjeldende lovkrav og bedriftens behov. Prosjektgruppen kan også kartlegge hvilke arbeidsbetingelser og identifisere faglige problemstillinger gjennom å utarbeide mål og iverksette tiltak. Siden prosjektet ikke skal ut på markedet og heller skal anvendes her på skolen, har det blitt gjort analyse ut ifra hva slags behov skolen trenger.

IKT

- **Mål:** Ha kunnskap om operativsystemer, dataverktøy som benyttes ved kommunikasjon, filstruktur, tekstbehandlingsprogrammer.
- Gruppen har gjennom MS produkter lært oss strukturert oppbygging av tekst, redigering og hatt en god filstruktur på prosjektet, samt kommunisert via epost og meldingstjenester. Gruppen har også utviklet og publisert nettside på internett.

Prosjekt og kvalitetsstyring

- **Mål:** Ha kunnskap om hvordan en utarbeider, dokumenterer og vedlikeholder bedriftens HMS/IKT system og bedriftens kvalitetssikringssystem i samsvar med aktuelle krav, lover, regler og standarder. Ha kunnskap om metodikk for organisering og ledelse av prosjekter.
- Gruppen har gjennom ProjectLibre laget en fremdriftsplan for prosjektet, hvor milepæler og mål har blitt satt opp. Gruppen har også gjennom «ettermannskontroll» sett igjennom hverandres arbeid og kvalitetssikret dette. Ettersom gruppen har jobbet med bevegelig mekaniske deler samt jobbet med strøm har gruppen hatt sikkerhetsrutiner for å ivareta HMS behovet for prosjektet.

Lovkunnskap

- **Mål:** Kan lage aktuelle avtaler, anvende aktuelle lover og løse praktiske problemstillinger ved hjelp av aktuelle rettskilder
- Gruppen har gjennom maskindirektoratet, EMC-direktoratet, NEK 400 tatt hensyn til norske lover og regler for å ivareta sikkerhet, krav og reflektert rundt kunnskap om teknologi og faglig ledelse.

Elektro og automasjon

- **Mål:** Ha kunnskap om vekselstrøm kretser, trefasesystemer, nettsystemer, elektrotekniske anlegg, feilsøking, regler og standarder og planlegging.
- Gruppen har benyttet relevante analysemetoder for normer og forskrifter, verktøy for beregninger og måling av spenning, strøm, effekt, resistans i vekselsstrømkretser. Gruppen har også dimensjonert valg av komponenter som inngår i det elektrotekniske anlegget samt reflektert over resultatene.

Teknisk dokumentasjon

- **Mål:** Kan utarbeide skjema, detalj, sammenstilling og arrangementstegninger, utskrift i ulike formater, lese og forstå tegninger fra beslektede fagområder, toleranser, aktuelle komponentbibliotek og eksportere tegninger til DAP-systemer.
- Gruppen har 3D-printet ut plukkobjekter og har da benyttet seg av Autodesk og fulgt standarder for teknisk dokumentasjon, gruppen har benyttet seg av toleranser i form av tilpassing til robotens griper.

Mekatronikkstyringer

- **Mål:** Ha kunnskap om planlegging og dokumentasjon av automatiserte anlegg, planlegging av sensorer, aktuatorer og andre komponenter som brukes, industrielle standarder i forbindelse med datakommunikasjon, programmering og konstruksjon av mekaniske styresystemer, elektromekanisk, PLS, roboter, elektronikk, pneumatikk, og PC ut ifra industriell sammenheng.

- Gruppen har planlagt og konstruert et elektrisk og mekanisk styresystem med PLS programmering, datakommunikasjon, robotprogrammering og dokumentert prosjektet etter krav og retningslinjer som gjelder innen mekanisk industri.

Mekatronikkdesign og produksjon

- **Mål:** Ha kunnskap om utvikling, vurdering, konstruksjon, dokumentasjon og design av mekatronikk produkter, industridesign, maskinelementer,
- Gruppen har planlagt og konstruert et elektrisk og mekanisk styresystem med PLS programmering, datakommunikasjon, robotprogrammering og dokumentert prosjektet etter krav og retningslinjer som gjelder innen mekanisk industri.

2.6: Verktøy og praksis

Sysmac studio

Sysmac studio er en programvare som brukes til å programmere, simulere og overvåke produkter fra NX, NJ, MX og NA serien til Omron. Programvaren er bygget til å samsvare med den åpne standarden IEC 61131-3, og er basert på ladder-diagrammet, strukturert tekst og funksjonsblokker. En av fordelene med denne programvaren kontra tilsvarende programvarer fra andre leverandører er at man ikke trenger å lære seg det interne minnekartet til kontrolleren fordi Sysmac studio tilordner automatisk minne for de variablene som blir opprettet. Denne programvare ble brukt i å programmere frekvensomformeren og PLSen. I denne programvaren ble også kommunikasjonen med roboten satt opp.

AutoCad

Autocad er et 2D basert tegningsprogram, hvor man kan lage tekniske tegninger, detaljtegninger, skjemategninger eller sammenstillingstegninger. Denne programvaren ble brukt i å lage skjemategninger av det elektriske anlegget. Symbolikken som ble benyttet til dette ble hentet fra NEK-144 standarden. I dette programmet ble det også laget en egen template som inneholdt tittel på tegningen, målestokk, tegningsnummer, firmanavn, dato, revisjons nr.

Dropbox

For lagring av dokumenter til prosjektet benyttet gruppen seg av en skybasert lagringstjeneste som heter dropbox. Her ble data for hele prosjektet systematisk lagret, og det var mulig at flere kunne jobbe på samme dokumentet samtidig. Ved hjelp av dette programmet kunne gruppemedlemmene til enhver tid være oppdatert på hva som foregikk i prosjektet.

Office

Office pakken inneholder flere programmer som gruppen har hatt nytte av:

Word

Word er et tekstbehandlingsprogram som har blitt brukt i forbindelse med rapportskriving, møtereferater og andre tekstdokumenter.

Powerpoint

Powerpoint er et visuelt og grafisk program som hovedsakelig brukes til å lage presentasjoner. Dette programmet ble brukt i forbindelse med forprosjektfremføring og hovedprosjektfremføring. Programmet fungerer slik at man redigerer hvert

Excel

Excel er et regnearkprogram som benyttes til å lage tabeller og beregning og analysering av data. Gjennom prosjektet har dette programmet blitt brukt til timeregistrering, risikoanalyse, komponentlister og rekkeklemmetabeller. Det ble også brukt til å sette opp budsjettet for prosjektet.

ProjectLibre

ProjectLibre er et dataprogram som brukes til prosjektledelse. Gruppen brukte dette programmet til å utarbeide en gjennomføringsplan for prosjektet. Her ble prosjektet brutt ned i aktiviteter og gjøremål slik at gruppen får en oversikt over prosjektets omfang. Aktivitetene ble satt opp i den mest naturlige rekkefølgen, slik at den ene aktiviteten ikke kunne begynne før den foregående var ferdigstilt.

Aktivitetene ble satt opp med en startdato og en sluttdato, og det ble beregnet hvor mye tid det ville ta å gjennomføre hver enkelt aktivitet. Når aktivitetene var satt opp i rekkefølge, ble de overført til et gantt-diagram.

Teamview

Teamview er en programvare som gjøre det mulig å fjernstyre en datamaskin. Denne programvaren ble brukt når gruppen hadde behov for veiledning i forbindelse med programmering av PLS. Ved å oppgi ID og passord til teknisk support hos Omron, kunne de fjernstyre datamaskinen vår. Denne programvaren viste seg å være en effektiv måte for å motta veiledning til programmering, kontra lange telefonsamtaler eller e-post.

2.7: Konklusjon

Prosjektgruppen er meget godt fornøyd med resultatet fra prosjektet. Det er spesielt tre punkter som trekkes frem som suksess:

- Komme i mål og fullføre prosjektet på en tilfredsstillende måte innenfor gitte tids og rammebetinger.
- Samarbeidet i gruppen, prosjektet har gitt et tett samarbeid i gruppen hvor alle er avhengig at alle gjøre sine oppgaver. Kommunikasjonen i prosjektgruppen har vært meget god og arbeidsfordelingen har vært bra.
- Gruppen har lært enormt mye gjennom prosjektperioden og fått meg seg mye nyttig erfaring videre.

Videre er det også noe som kunne ha vært gjort annerledes, trekker derfor frem disse punktene:

- Gruppen skulle ha tidlig fått informasjon fra leverandører på hvilke kommunikasjonsmetoder mellom robot og PLS som var å foretrekke og hva NX-100 kontrolleren hadde fra før og evt. hvor mye arbeid det hadde vært og montert nyt korth. Da kunne gruppen ha slippet mye arbeid.
- Gruppen kunne ha vært bedre på å avtale tidspunkter for hjelp fra eksterne leverandører, ettersom de har som regel mye gjøre og ikke alltid like lett å få tak i.
- Valg av konsept kunne ha vært avgjort tidligere, for å komme fortare i gang med innkjøp av materiell til robotlaben.

Vedlegg

Vedlegg 1 - Konsept utvikling med tilhørende prisestimat

Vedlegg 2 - Pristilbud fra Omron

Vedlegg 3 – Pristilbud ITO

Vedlegg 4 – Kapittel 1.3 oversatt til engelsk

Vedlegg 5 – Elektrotegning

Vedlegg 6 – Robotprogram

Vedlegg 7 – Bruksanvisning for labøvelsen

Vedlegg 8 – Pneumatikkkjema

Vedlegg 9 – Tegninger av plukkobjektene

Vedlegg 10 – Utregning av retardasjonstid

Vedlegg 11 - PLS-program

Vedlegg 12 – Utregning av kabler

Vedlegg 13 – Reelt budsjett

Vedlegg 14 – Opprinnelig fremdriftsplan

Vedlegg 15 – Reell fremdriftsplan

Vedlegg 16 – Møtereferater

Vedlegg 17 – Samarbeidskontrakt

Bildeliste:

Bilde 1. Skisse av robotcelle (SHE.engineering, 2017).....	9
Hentet fra: SHE. Engineering	
Bilde 2. Kommunikasjon mellom komponentene (SHE.engineering).....	11
Hentet fra: SHE.engineering	
Bilde 3. Motoman SIA20 (Yaskawa, 2017).....	13
Hentet fra: https://www.motoman.com/industrial-robots/sia20d	
Bilde 4. Transportbånd (ITO, 2017).....	15
Hentet fra: http://ito.no/produkter/produktkategorier	
Bilde 5. MX2 Frekvensomformer (Omron, 2017).....	17
Hentet fra: http://www.ia.omron.com/products/family/3164/	
Bilde 6. Prinsippet til frekvensomformeren (Frekvensomformeren, 2017).....	18
Hentet fra: http://w3.elektrofag.info/elektroteknikk/frekvensomformeren	
Bilde 7. MX2 EtherCAT option card (Omron, 2017).....	19
Hentet fra: http://www.ia.omron.com/products/family/3164/	
Bilde 8. Omron NX1P2 (Omron, 2017).....	21
Hentet fra: https://components247.com/omron-nx1p2-1040dt-machine-automation-controller.html	
Bilde 9. Power supply unit (Omron, 2017).....	22
Hentet fra: https://www.ia.omron.com/products/family/3190/specification.html	
Bilde 10. Encoder option card (Omron, 2017).....	22
Hentet fra: https://www.ia.omron.com/products/family/3190/specification.html	
Bilde 11. RS-232 Commnication unit (Omron, 2017).....	25
Hentet fra: https://www.myomron.com/index.php?action=kb&print=1694	
Bilde 12. Ethernet switch (Ethernet switch, 2017).....	26
Hentet fra: http://www.westermo.com/web/web_en_idc_us.nsf/AllDocuments/57728AC712BACA4EC1257893003393F0	
Bilde 13. Sikkerhetsrele (Omron, 2017).....	27
Hentet fra: https://www.ia.omron.com/products/family/3419/specification.html	
Bilde 14. Timeregistreringsskjema (SHE.engineering, 2017).....	37
Hentet fra: SHE.engineering	

Kilder:

Electrical and Electronics Engineering Dictionary. (2017, 18.03) Hentet fra:
<http://www.dictionaryofengineering.com/definition/vision-system.html>

Westin. (2014). Nytt EMC direktiv stiller krav – og nye spørsmål. Hentet fra:
http://www.infoklikk.no/images/Elektro/EI-generelt/Nytt_EMС_direktiv.pdf

Standard.no. (2017,15.03) NEK 144:2004. Hentet fra:

https://www.standard.no/fagomrader/elektrofag/elektro/nek-144_2004/

Forskrift om maskiner. (2016) *Forskrift om maskiner*. Hentet fra:

https://lovdata.no/dokument/SF/forskrift/2009-05-20-544#KAPITTEL_6

FQ2 (2017). *Omron.com*. Hentet fra:

<https://www.ia.omron.com/products/family/3131/>

Knut A Rosvold. (2017). 3D-printing. *Store norske leksikon*. Hentet fra:
<https://snl.no/3D-printing>

Knut A Rosvold. (2017). Asynkronmotor. *Store norske leksikon*. Hentet fra:
<https://snl.no/asynkronmotor>

Incremental encoder interface. (2017) *Posital*. Hentet fra:
<https://www.posital.com/en/products/communication-interface/incremental/incremental-encoders.php>

Frode Høyte. (2016). *Transformator*. Hentet fra:
<https://snl.no/transformator>

Ivar Gunvaldsen. (2009). *Likeretter*. Hentet fra:
<https://snl.no/likeretter>

Konseptutvikling av infrastruktur

På bakgrunn av krav og produktspesifikasjoner fra oppdragsgiver har gruppen utviklet fire konsepter for infrastruktur og to konsepter for vision-kamera.

Felles for konseptene:

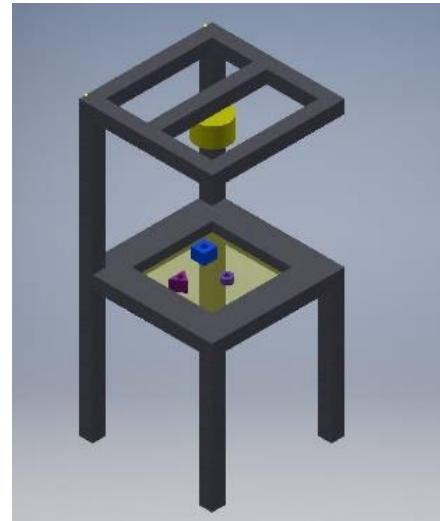
Felles for konseptene for infrastruktur er at roboten i samspill med vision-kameraet skal kunne plukke og sortere ulike objekter. Objektene vil bestå av forskjellige geometriske figurer, med hull i senter for enkel plukking. Griperen på roboten har tre «fingre» som er trapes formet, den er ment for å plukke opp objekter etter den indre diametern, altså en «chuck med tre bakker».

Samtlige konsepter baserer seg på «pick and place» oppgaver som består av gjenkjennelse av geometri og deretter sortering. Vision-kamera vil registrere koordinatene til objektene og sender dette videre til roboten for plukking. Programmering av softwaren til vision-systemet vil foregå på en datamaskin, som også gir mulighet for monitoring.

Konsept 1 - Bord med underbelysning

Konsept 1 går ut på å konstruere et bord med underbelysning.

I dette konseptet blir objektene plassert manuelt på bordplaten, deretter vil vision-systemet ta bilde, registrere plasseringen på objektene og sende koordinatene til roboten. Når roboten har fått koordinatene plukker den og sorterer objektene i esker.



1: Konsept 1. Illustrasjon: SHE-engineering.

Fordeler	Ulemper
<ul style="list-style-type: none">Underbelysning gjør at man ikke trenger kamera med egen lyskilde.Konstruksjonen gjør det mulig å henge opp solskjerming.Enkel konstruksjon å bygge.Minimal infrastruktur.Justerbar kamerahøyde.Bordet er flyttbart.	<ul style="list-style-type: none">Må bryte lysgardinen for å legge på nye objekter.Kan tenkes at studenter står inne i cellen samtidig som roboten jobber.Potensialet for videreutvikling er lavt.Ikke industrirettet.

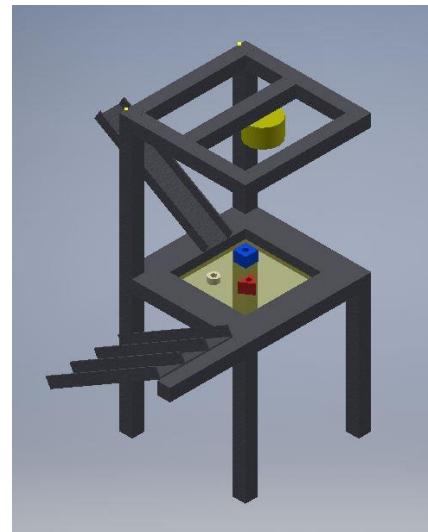
Prisestimat for konsept 1

Konsept 1				
Materiale	Pris	Leverandør	Leveringstid	
Alu.profiler	kr	3 000,00 Koberg, ktech, FTO	1 uke	
Lys	kr	1 000,00 elfa	1 dag	
Plexiplater	kr	200,00 Biltema	1dag	
Kabling	kr	200,00 Elfa	1 dag	
Bryter for lys	kr	150,00 Elfa	1 dag	
Div skruer	kr	500,00 FTO, Wurth	1 dag	
Plukkobjekter	kr	4 000,00 Koberg, ktech, jo	1 dag	
Lysskjerming	kr	200,00 Biltma	1 dag	
<hr/>				
Sum	kr	9 250,00		

Konsept 2- Bord med underbelysning og sklie

Konsept 2 går ut på å konstruere et bord med underbelysning og sklie.

I dette konseptet blir objektene plassert i en sklie utenfor robotcellen som gjør at objektene blir ført ned til bordplaten. Vision-systemet vil så ta bilde og registrere koordinatene til objektene. Når roboten har fått koordinatene plukker den og sorterer objektene i sklier som går ut av cellen.



2: Konsept 2. Illustrasjon: SHE-engineering.

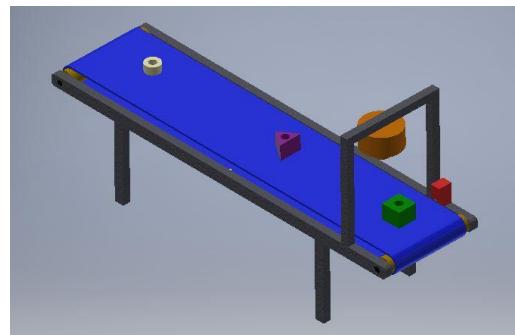
Fordeler	Ulemper
<ul style="list-style-type: none"> Underbelysning gjør at man ikke trenger kamera med egen lyskilde. Konstruksjonen gjør det mulig å henge opp solskjerming. Enkel konstruksjon å produsere. Minimal infrastruktur. Slipper å bryte lasergardinens i cellen for å legge på deler. Ivaretar HMS-regler. 	<ul style="list-style-type: none"> Lage åpninger i nettingveggen, slik at objekter sklir inn og ut Potensialet for videreutvikling er lavt. Ikke industrirettet.

Prisestimat for konsept 2

Konsept 2				
Materiale	Pris	Leverandør	Leveringstid	
Alu.profiler	kr	4 000,00 Koberg, ktech, FTO	1 uke	
Lys	kr	1 000,00 elfa	1 dag	
Plexiplater	kr	200,00 Biltema	1dag	
Kabling	kr	200,00 Elfa	1 dag	
Bryter for lys	kr	150,00 Elfa	1 dag	
Div skruer	kr	500,00 FTO, Wurth	1 dag	
Plukkobjekter	kr	4 000,00 Koberg, ktech, jo	1 dag	
Lysskjerming	kr	200,00 Biltma	1 dag	
Sum		10 250,00		

Konsept 3 -Transportbånd

Konsept 3 går ut på at objektene blir plassert på båndet fra utsiden av cellen. Vision-kamera stopper båndet når objektet blir registrert, deretter forteller den roboten hvilket objekt det er og hvor det skal plukkes opp.



3: Konsept 3. Illustrasjon: SHE-engineering.

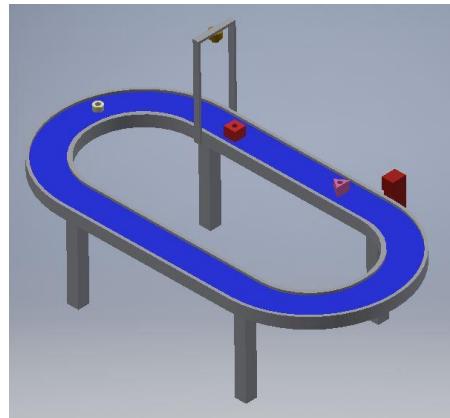
Fordeler	Ulemper
<ul style="list-style-type: none"> • Stort potensiale for videreutvikling. • Slipper å bryte lasergardin. • Justerbar kamerahøyde. • Godt læringsutbytte. • Klare likhetstrekk med industrien. • Flerfaglig labøvelse. 	<ul style="list-style-type: none"> • Trenger større plass i cellen. • Ikke mulighet for underbelysning.

Prisestimat for konsept 3

Konsept 3				
Materiale	Pris	Leverandør	Leveringstid	
Transportbånd 1500x200	kr 14 500,00	Intern transport AS	3-4 uker	
Frekvensomformer	kr 1 500,00	Mitsubishi	2 uker	
Skap	kr 4 500,00	Rittalskap fra Elfa	1 dag	
Din-skinne	kr 100,00	Elfa	1 dag	
PG-nippel	kr 100,00	Elfa	1 dag	
Kabel flerleder	kr 500,00	Elfa	1 dag	
Skjermet kabel for frekvenseomformer	kr 500,00	Elfa	1 dag	
Endehylser	kr 800,00	Elfa	1 dag	
RK	kr 400,00	Elfa	1 dag	
Lasersensor	kr 900,00	Sick, omron,	3 dager	
Kabel til skap	kr 500,00	Elfa	1 dag	
Plukkobjekter	kr 4 000,00	Koberg, ktech, jo	1 uke	
Profil for oppheng av kamera	kr 500,00	Koberg, ktech, jo	1 uke	
Bein til Transportbånd	kr 1 600,00	Intern transport AS	3-4 uker	
Gitterbaner	kr 450,00	Elfa	1 dag	
Strips	kr 100,00	elfa	1 dag	
Rekkeklemmer	kr 700,00	elfa	1 dag	
Start og stopp bryter på bånd	kr 500,00	Elfa	1 dag	
Støyfilter	kr 300,00	Mitsubishi	3-4 uker	
Automatsikring 6A	kr 600,00	Schneider fra Elfa	1 dag	
Kabelkanal	kr 100,00	elfa	1 dag	
Lysskjerming	kr 200,00	Biltema	1 dag	
Sum		kr 33 350,00		

Konsept 4 -Transportbånd som går i ring

Konsept 4 går ut på at objektene blir plassert på båndet fra utsiden av cellen. Vision-kamera stopper båndet når objektet blir registrert, deretter forteller den roboten hvilket objekt det er og hvor det skal plukkes opp. De objektene som ikke oppfyller kravene vil fortsette på båndet.



4: Konsept 4. Illustrasjon: SHE-engineering.

Fordeler	Ulemper
<ul style="list-style-type: none">• Stort potensiale for videreutvikling.• Programmet kan gå i loop, det vil si at roboten kan jobbe kontinuerlig.• Slipper å bryte lasergardinien• Justerbar kamerahøyde.• Godt læringsutbytte.• Klare likhetstrekk med industrien.• Flerfaglig labøvelse.	<ul style="list-style-type: none">• Trenger større plass i cellen.• Ikke mulighet for underbelysning.• Har ikke funnet leverandør for rundbånd.

Har ikke funnet leverandør av rundbånd så derfor er det ikke utarbeidet et prisestimat.

Konseptutvikling av vision-kamera

Det finnes en rekke ulike kameraløsning innen vision-system, gruppen skal se på forskjellen på hovedgruppen vision-kamera som kan benyttes. Behovet for gode lysforhold vil være det samme for begge kameratypene. Her vil 2D og 3D kameraer veies opp mot hverandre.

Konsept 1– 2D kamera

I et standard 2D maskinsyn er bildet flatt, det er kalibrert til målinger av lengde og bredde, men det gir ikke noe høydeinformasjon. Skal man måle høyde må tilleggsutstyr som for eksempel avstandslaser benyttes. Dersom man monterer en avstandsmåler blir det definert som et 2.5D kamera.

Fordeler	Ulemper
<ul style="list-style-type: none">• 2D-kameraet er mye rimeligere enn 3D-kamera.• Softwaren kan være enklere enn 3D-kamera.	<ul style="list-style-type: none">• Objektene som skal plukkes må være av samme høyde.• Mer omfattende å videreutvikle enn et 3D-kamera.

Prisestimat konsept 1

2D-kamera				
Materiale	Pris		Leverandør	Leveringstid
2D-kamera	kr	20 000,00	Omron	3 dager
Objektiv til kamera	kr	3 500,00	Omron	3 dager
Kommunikasjonskabler kamera	kr	2 700,00	Omron	3 dager
Lys på kamera		Ukjent	Omron	3 dager
PLS	kr	17 000,00	Omron	3 dager
<hr/>				
Sum	kr	43 200,00		

Konsept 2- 3D kamera

3D-kamera man måle lengde, bredde og høyde. Dette oppnås ved at kameraet observerer objektet fra to forskjellige synsvinkler. Skjæringspunktet for disse to linjene vil gi XYZ verdier.

Fordeler	Ulemper
<ul style="list-style-type: none">• Større mulighet for videreutvikling av laboppgaver.• Objektene som skal plukkes må ikke ha samme høyde.	<ul style="list-style-type: none">• Mer kompleks software.• Et 3D-kamera har et større plassbehov.• Kostbart.

Prisestimat konsept 2

3D-kamera				
Materiale	Pris	Leverandør	Leveringstid	
3D-kamera	kr 50 000,00	Sick	2 uker	
Objektiv til kamera	kr 3 500,00	Omron	3 dager	
Kommunikasjonskabler kai	kr 2 700,00	Omron	3 dager	
Lys på kamera	<i>Ukjent</i>	Omron	3 dager	
PLS	kr 17 000,00	Omron	3 dager	
<hr/>				
Sum	kr 73 200,00			

Produktnr	Typebetegnelse	Produktbeskrivelse	EL nummer	Nettopris	Rabatt	Antall	Totalt
669508	S8VK-S06024	Book type power supply, 60 W, 24VDC, 2,5A, DIN rail mounting, Push-in terminal, Coated		275,00		2	550
353025	WES SDI-550	i-Line 5-port Unmanaged Switch - 5 x 10/100BaseT		575		1	575
374581	XSGW-6LSZH8SS100CM-Y	Ethernet Cat.6 patch cable, LSZH double shielding (Yellow), 1 m		30		1	30
374603	XSGW-6LSZH8SS100CM-B	Ethernet Cat.6 patch cable, LSZH double shielding (Blue), 1 m		30		1	30

Produktnr	Typebetegnelse	Produktbeskrivelse	EL nummer	Listepris	Rabatt	Antall	Totalt
379091	3G3MX2-AB004-E CHN	0.4/0.55kW(HD/ND),	41 208 26	2 528,00	50 %	1	1264
379093	3G3MX2-AB007-E CHN	MX2 Frekvensomformer, 0.75/1.1kW(HD/ND), 5.0/6.0A(HD/ND), 230VAC, 1-fas forsyning	41 208 27	2 985,00	50 %	0	0
386171	AX-FIM1010-SE-LL	MX2 RFI filter, 10A, 230VAC, 1-fas, for 0.12 til 0.55kW modeller		220,00	50 %	1	110
386172	AX-FIM1014-SE-LL	MX2 RFI filter, 14A, 230VAC, 1-fas, for 0.75kW modeller, lav lekkasjestrom		320,00	50 %	0	0
416241	AX-GPM01-ICE	MX2 jordingsbeslag: For ..AB001-AB004 og ..A2001-A2007		125,00	50 %	1	62,5
416242	AX-GPM02-ICE	MX2 jordingsbeslag: For ..AB007-AB022 og ..A2015-A2037 og ..A4004-A4040		125,00	50 %	0	0
343139	3G3AX-MX2-ECT	MX2 EtherCAT option card		2 140,00	50 %	1	1070

Produktnr	Typebetegnelse	Produktbeskrivelse	EL nummer	Listepris	Rabatt	Antall	Totalt
672504	NX1P2-1040DT1	Sysmac NX1P CPU with 40 Digital Transistor I/O (PNP), 1,5 MB memory, EtherCAT (2 servo axes, 4 PTP axes, 16 EtherCAT nodes), EtherNet/IP and 2 serial option ports		10 340,00	45 %	1	5687
380140	NX-PF0730	NX I/O power feed unit, 5-24 V DC input, 8 terminals, 10A, screwless push-in connector, 12mm wide		279,00	45 %	1	153,45
375645	NX-EC0122	1 x Incremental Encoder PNP Input, Phases A and B: Single-phase 500 kHz (phase difference pulse Input x4: 125 kHz), Phase Z: 125 kHz, screwless push-in connector, 12 mm wide		1 807,00	45 %	1	993,85
672505	NX1W-CIF01	Sysmac NX1P RS-232C screwless (15 m max.) serial communication option board		195,00	45 %	1	107,25

Produktnr	Typebetegnelse	Produktbeskrivelse	EL nummer	Listepris	Rabatt	Antall	Totalt
372122	FQ2-S35100F-08	FQ2-S35 vision sensor, avansert funksjon, 32 inspeksjon, 32 scener, lang avstand 970 mm, synsfelt 240mm, PNP, Ethernet og I/O innebygd		22 650,00	45 %	1	12457,5
351984	FQ-WD003-E	FQ I/O kabel, 3m		1 070,00	50 %	1	535
360745	FQ-WN003-E	FQ Ethernet kabel, rett M12-RJ45, 3m		596,00	50 %	1	298

Produktnr	Typebetegnelse	Produktbeskrivelse	EL nummer	Listepris	Rabatt	Antall	Totalt
672883	SYSMAC-LE201L	Sysmac Studio Lite license only, single user (requires SYSMAC-SE200D download or installation DVD, available separately)		10 340,00	50 %	1	5170
355294	SYSMAC-SE200D	Sysmac Studio installation DVD only, for Windows XP or Windows 7/8, 32 or 64-bit versions (requires SYSMAC-SE2**L license)		309,10	50 %	1	154,55

HTL-Enkoder er å foretrekke på transportbåndet for å passe med PNP interfacet på PLS'en.

29 248,10

TILBUD NR: P-748701

SHE.Engineering

E-post: SHE.engineering@outlook.com

Deres/Your ref.:

Vår/Our ref.:

Dato:

KK

22.11.2016

TRANSPORT AV HJULLAGRE VED ROBOTCELLE - SKOLEPROSJEKT

Viser til hyggelig forespørsel i forbindelse med ovennevnte.

Vi tilbyr vår mest standardiserte transportør, beregnet for max last ca. 50 kg/m. Selve båndet glir på et stålplateunderlag og det styres kun ved at driv- og vendetromlene er bomberete, dvs. tykkere på midten. Det betyr at båndet vandrer noen millimeter sideveis og tvangsstyres altså ikke. Transportøren leveres med girmotor for fast hastighet og eventuelle fotoceller og el. styring er ikke medregnet. Se separat pris for frekvensomformer for hastighetsregulering og start stopp.

Vii har fornøyelsen av å kunne tilby følgende utstyr:

Pos. 1.: 1 stk.

ITO BÅNDTRANSPORTØR

Type: ITOLETT Type 2-C120-100T/230-1,5-0,25-12

Lengde c-c tromler: 1 500 mm

Banebredde innv: 230 mm

Banebredde tot utv: 310 mm

Rammeprofil: C-120x40x15x4 mm

Trommeldia.: Ø100 mm

Båndtype: HABASIT NAB 8 EEDV11 - PVC

Båndbredde: 200 mm

Båndunderlag: 3,0 mm stålplate.

Banehast.: ca. 12 m/min?? – Fast nom.

Motor: SEW spiroplangirmotor 0,25

Motorplassering: Direktemontert i avlastningsende

Uten banestøtter (ben) og sidestyringer

Pris:

kr. 14.500,-

Pos. 2.: 2 stk.

ITO BANESTØTTER (BEN)

for høyde 800?? ± 40 mm

Pris: à kr. 800,-

kr. 1.600,-

Intern Transport AS

Konowsgt. 5, 0192 Oslo - P. O. Box 9176 Grønland, 0134 Oslo, Norway.

Tel: +47 23 30 26 00 - Fax: +47 23 30 26 01 - e-mail: ito@ito.no - www.ito.no

Bank: 7047.05.05514. - Org no.: NO. 812983482 MVA

Pos. 3.: 1 stk.

SEW FREKVENSOMFORMER

Type: MOVITRAC LTE-B - 0,75 kW
For trinnløs hastighetsregulering
samtidig med start/stopp
Kapsling: IP66
For el. tilførsel: 1-fas 230 V

Pris ukoblet og umontert:

kr. 4.250,-

Pris og leveringsbetingelser:

Prisene er regnet ekskl. mva., eksp. og emballasje, fritt levert verksted Tyrstrand.

Leveringstid:

ca. 3-4 arbeidsuker fra bestilling.

Betalingsbetingelser:

Netto pr. 30 dager.

Finish:

Vanger og banestøtter leveres primet og lakkert i blå RAL 5010 industrifarge.
Tromler i stålblank utførelse. Topplaten er varmgalvanisert.

Garanti:

Alt utstyr er garantert i 1-ett år mot eventuelle fabrikasjonsfeil, basert på normal drift og vedlikehold.

Vi håper å ha tjent Dem så langt og står ellers til tjeneste med alle ytterligere opplysninger som måtte være ønskelig.

Det ville glede oss å motta Deres ordre.

Med vennlig hilsen

ITO-

INGENIØRFIRMAET INTERN TRANSPORT A/S

Kjell Knutsen

Der ikke annet skriftlig er avtalt gjelder følgende:

Eventuell emballasje debiteres til selvkostpris og tas ikke tilbake. Ved Force Majeure forbeholder vi oss retten til å heve avtalen, forlenge leveringstiden eller heve prisen i forhold til de omstendigheter som vi da ikke har kontroll over. Ved Deres eventuelle bestilling er vår ordrebekrefte utslagsgivende. Det tas salgspant i levert vare iht. pantel. §3-14 og tvfl. §215a inntil fullt betalt. Etter forfall beregnes 1,2% morarenter pr. mnd. For leveransen gjelder Norsk Industris alminnelige leveringsbetingelser NL 01.

Intern Transport AS

Konowsgt. 5, 0192 Oslo - P.O. Box 9176 Grønland, 0134 Oslo, Norway.
Tel: +47 23 30 26 00 - Fax: +47 23 30 26 01 - e-mail: ito@ito.no - www.ito.no
Bank: 7047.05.05514. - Org no.: NO. 812983482 MVA

Concept development

Based on the requirements and product specifications from the client, the group has developed four concepts for infrastructure and two concepts for vision-camera. The concepts for infrastructure are based on how the objects for picking are transferred to the camera for processing image information. The vision-camera concepts is about what type of camera that was chosen.

The group and the client in cooperation has chosen concept 3 for infrastructure, as this concept provides the most industrial training platform. This concepts gives a better insight into how such a task can be solved in the industry compared with the other concepts. The group also believes that the outcome of learning are greater than the other options. This is because the lab exercise will include control of a conveyor, vision-camera programming and robotprogramming.

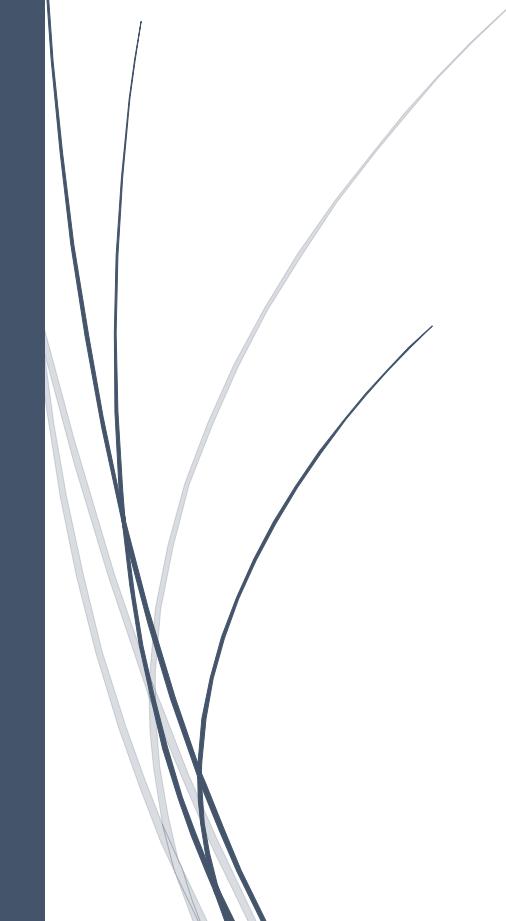
The health safety and environment (HSE) perspective will be maintained by avoiding entering the robotcell. The price estimat of the project is not the cheapest option, but the group has based the management parameter from the client, who ranks quality over cost.

The group has chosen concept 1 for vision-camera. This choice is based on recommendations from suppliers and external professionals. They consider it to be too extensive to use a 3D vision-camera in a project like this. A 2D camera can not measure the distance between the camera and the object, so the group decided to have a fixed height on the objects. The group also believes that a 2D camera provides a good insight into the vision-programming and it covers the basic knowledge of understanding the functionality. There is also a significant price difference between these cameras. A 3D camera has a starting price of 50 000 NOK,-, while a 2D camera has a starting price of 5000NOK,-

After making the decision of the concept with the client, the group talked to different suppliers and compared prices and products. The group choosed Omron as an supplier for the vision-system. This was because they had a connection with Skala RoboTech (supplier for the robot) and they knew the system. They could also offer us a good price for the equipment. The group choosed ITO as a supplier for infrastructure, they could deliver a conveyor according to our specification about measurement and they could deliver with an engine with encoder.



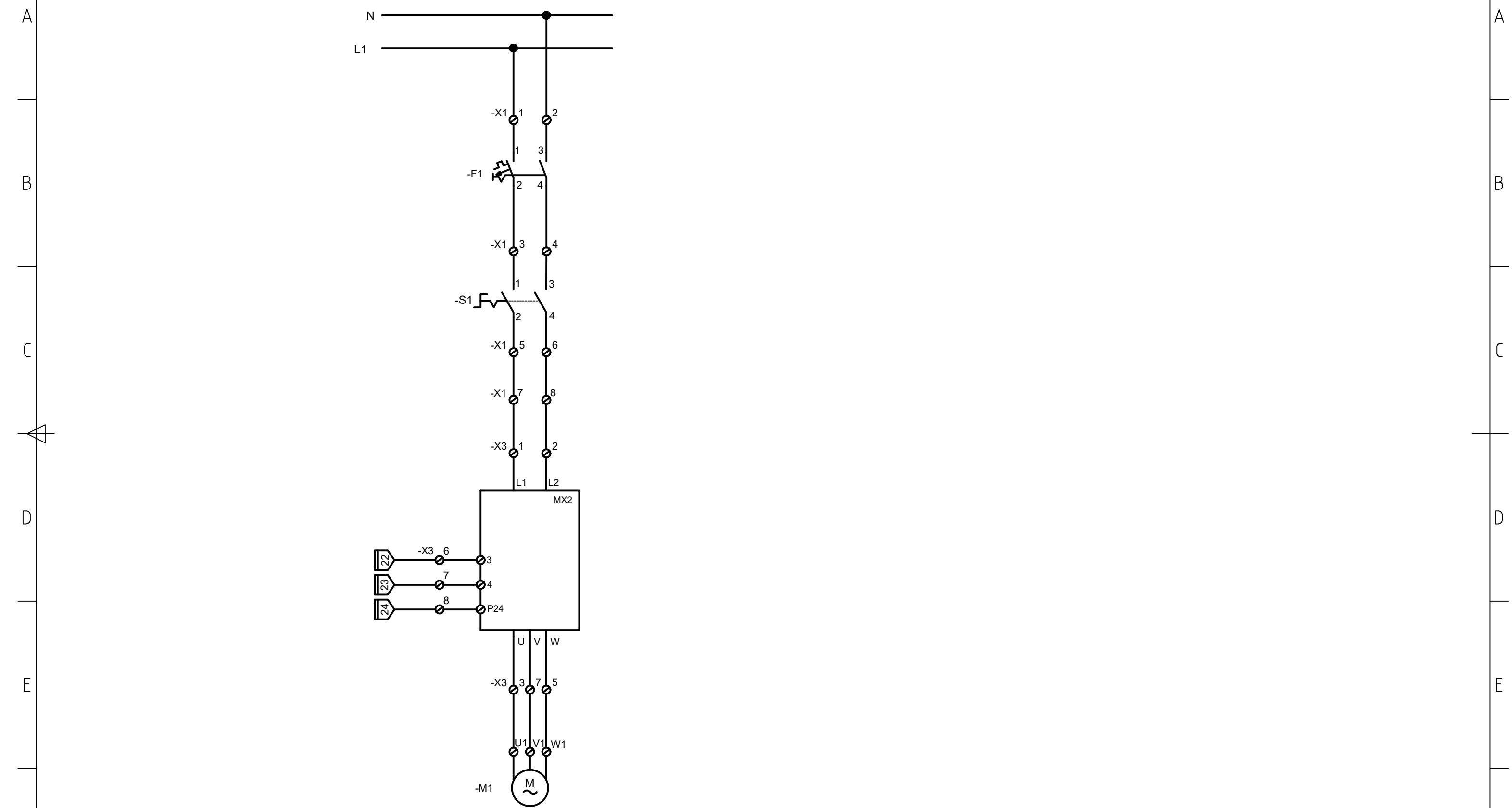
Elektrotegninger



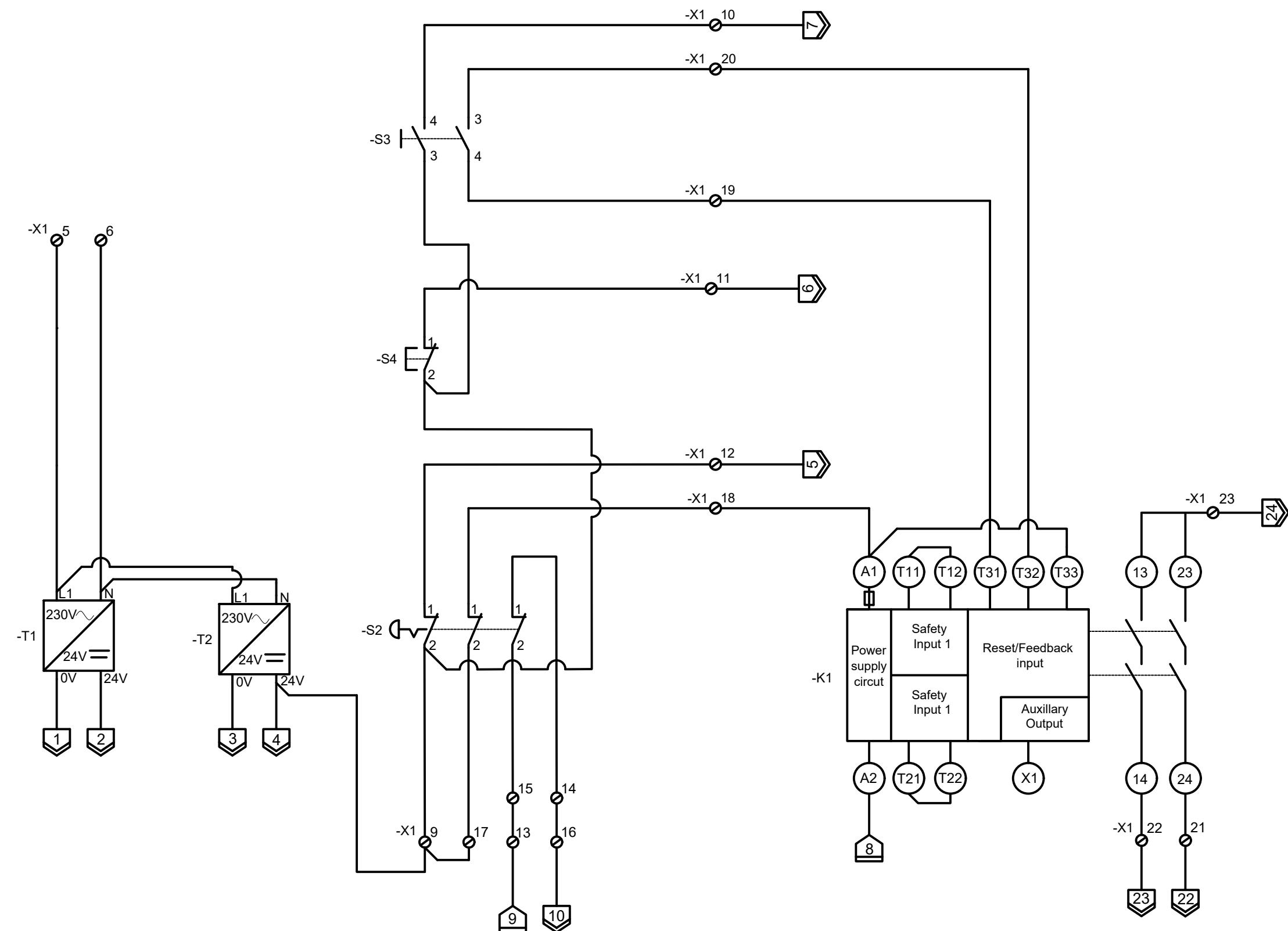
S.H.E-ENGINEERING
SINDRE HELLINSRUD, HÅVARD VIKÅS OG EIRIK HATLAND

Innholdsfortegnelse

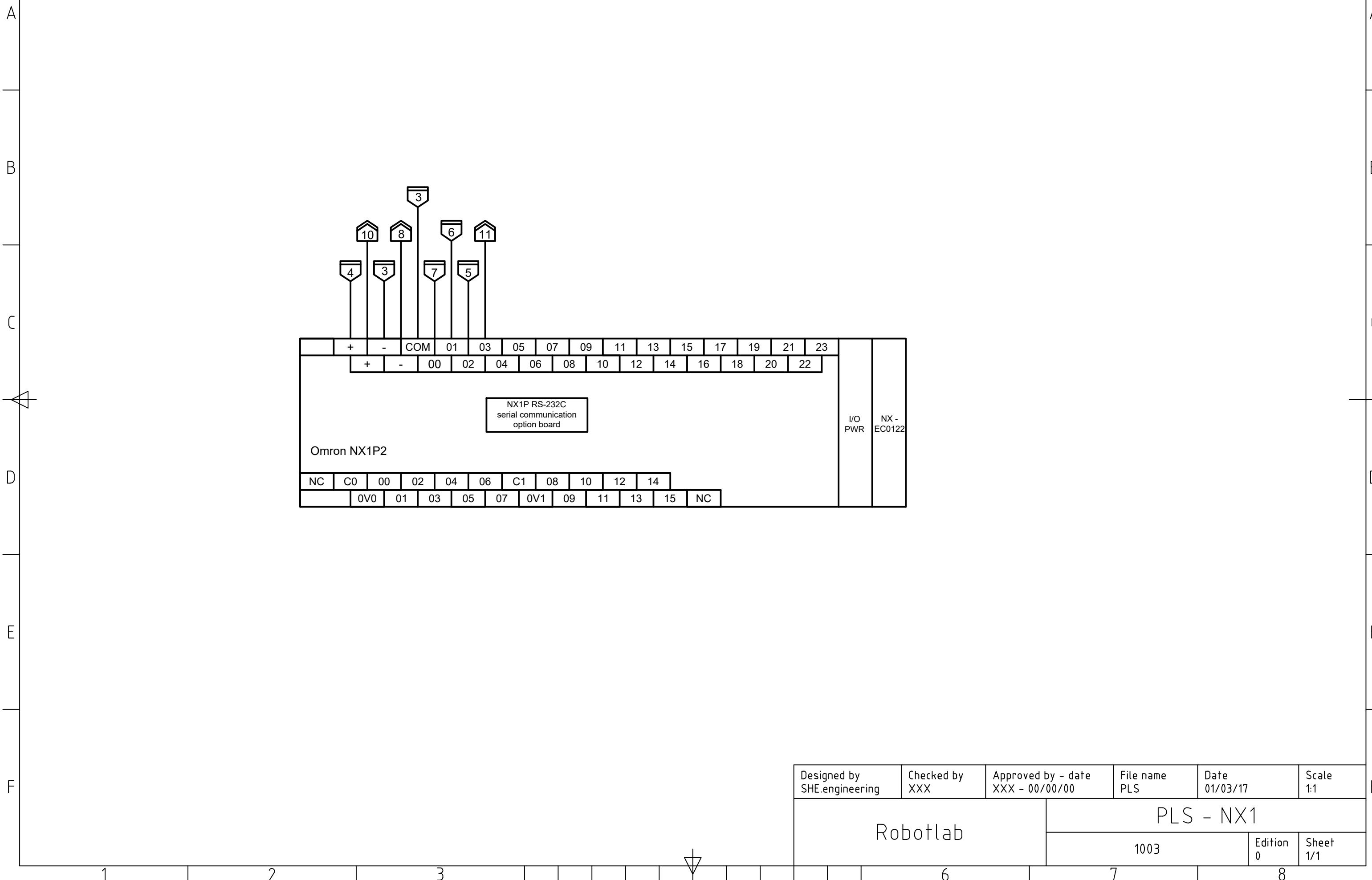
1001 - Hovedstrøm.....	1
1002 - Styrestrøm.....	2
1003 - PLS	3
1004 - Encoder og IO power.....	4
1005 - NX100.....	5
1006 - RS232 - Kommunikasjon.....	6
1007 - FQ2 - Vision sensor	7
1008 - Kommunikasjon	8
1009 -Arrangementtegning	9
1010 -Rekkeklemmetabell	10

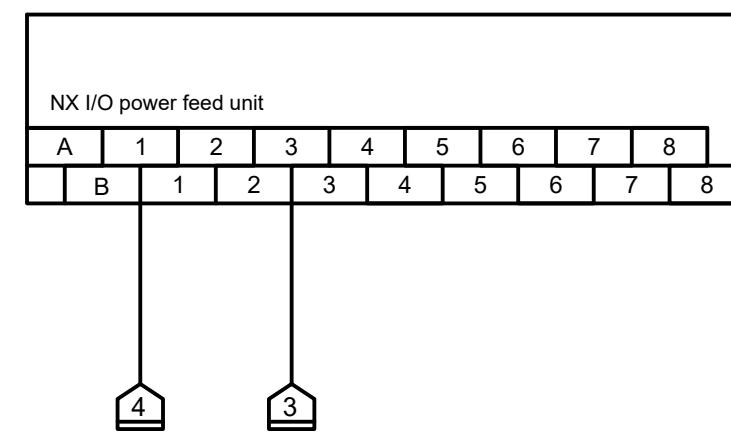
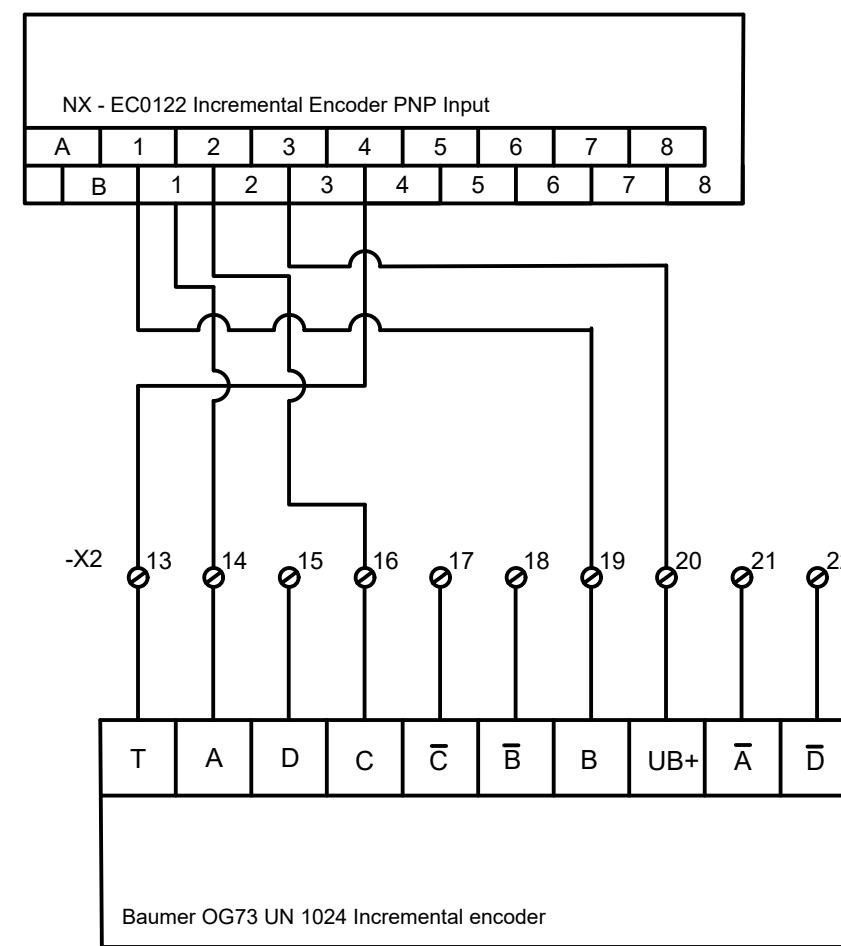


Designed by SHE.engineering	Checked by XXX	Approved by - date XXX - 00/00/00	File name Hovedstrøm	Date 01.03.17	Scale 1:1
Robotlab				Hovedstrøm	
1001		Edition 01		Sheet 1/1	



Designed by SHE.engineering	Checked by XXX	Approved by - date XXX - 00/00/00	File name Styrestrom	Date 01.03.17	Scale 1:1
Robotlab				Styrestrom	
1002		Edition 01		Sheet 1/1	





A

A

B

B

C

C

D

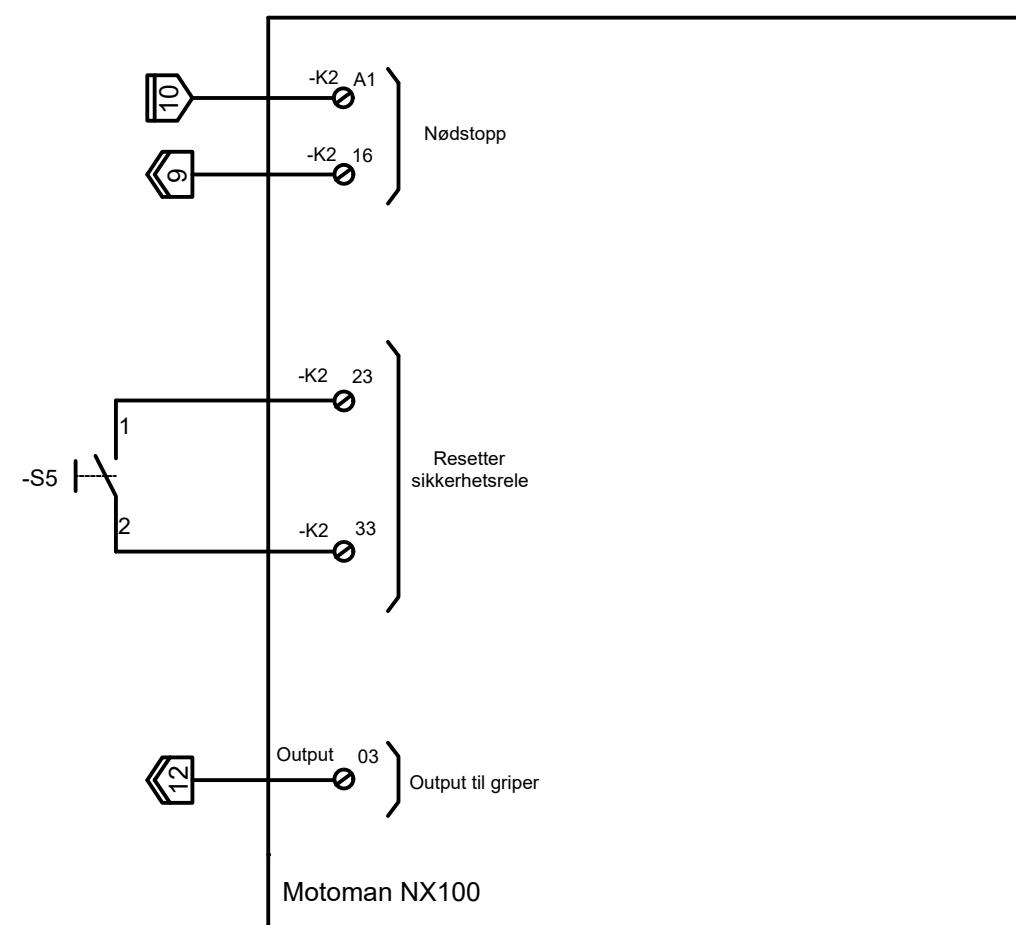
D

E

E

F

F



Designed by SHE.engineering	Checked by XXX	Approved by - date XXX - 00/00/00	File name NX100	Date 01/03/17	Scale 1:1
Robotlab			Motoman - NX100		
1005			Edition 01	Sheet 1/1	

A

A

B

B

C

C

D

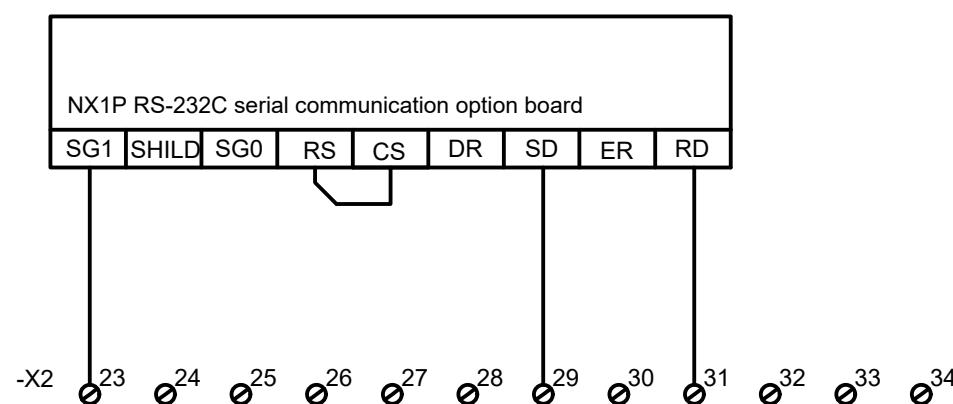
D

E

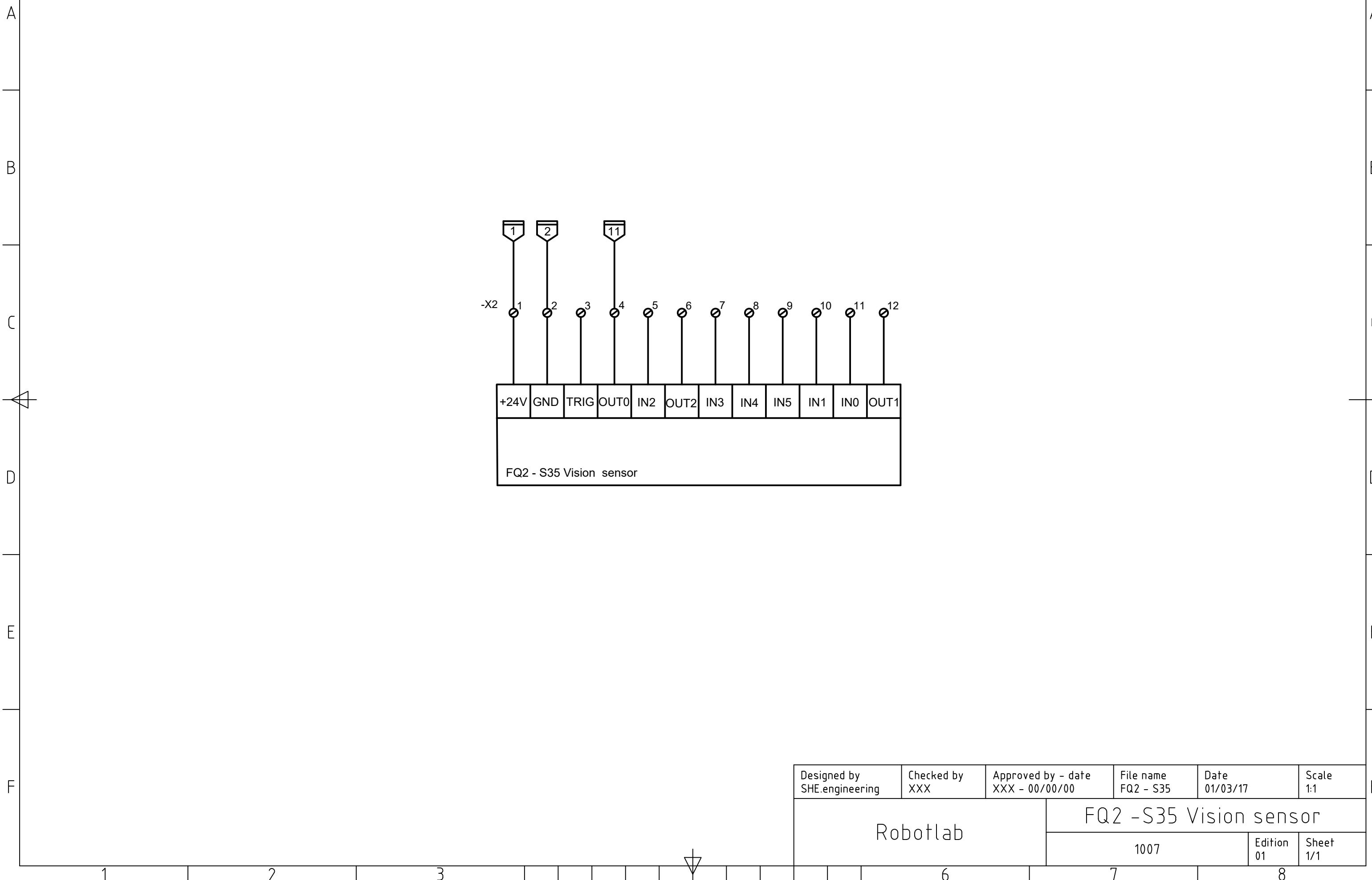
E

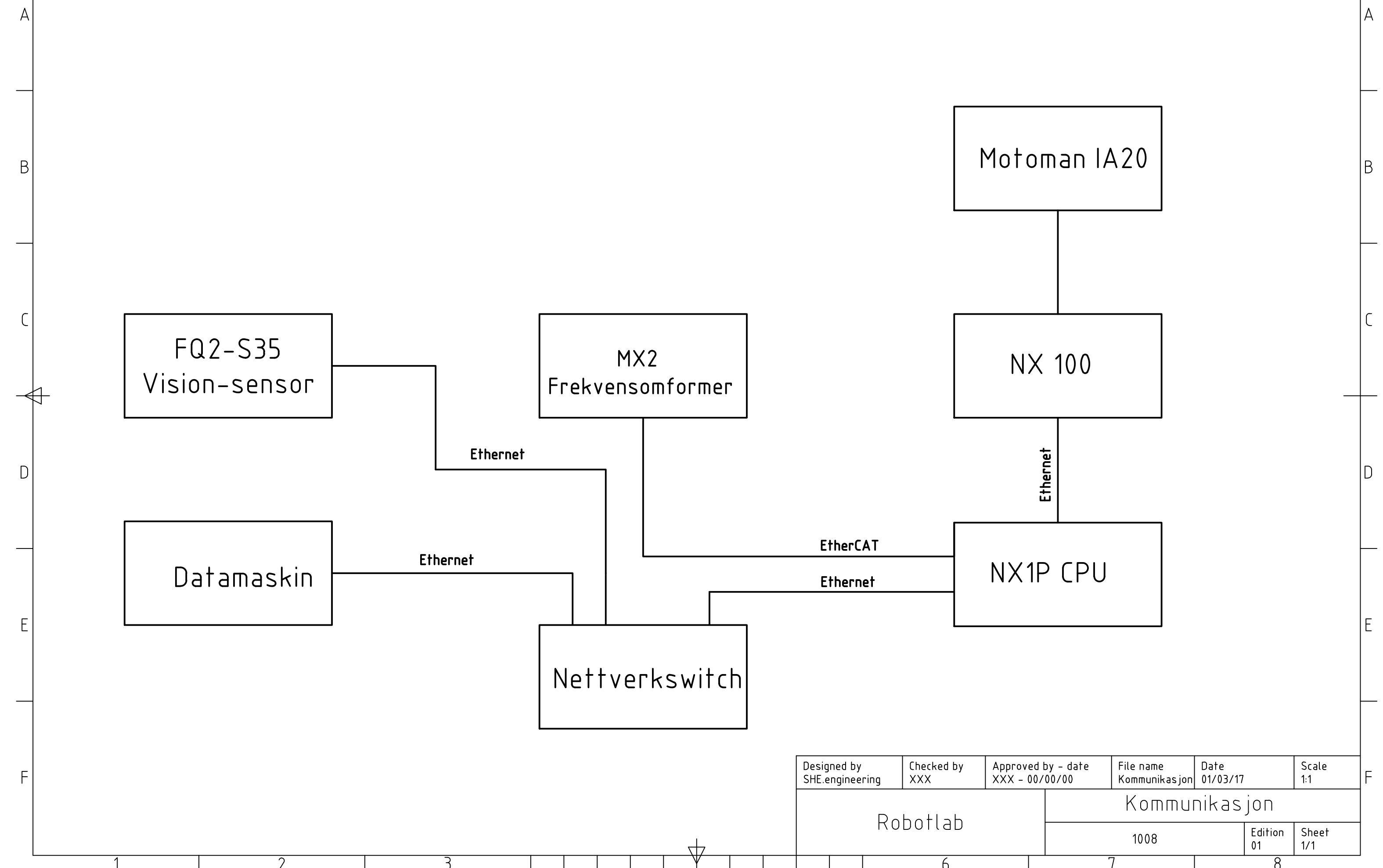
F

F

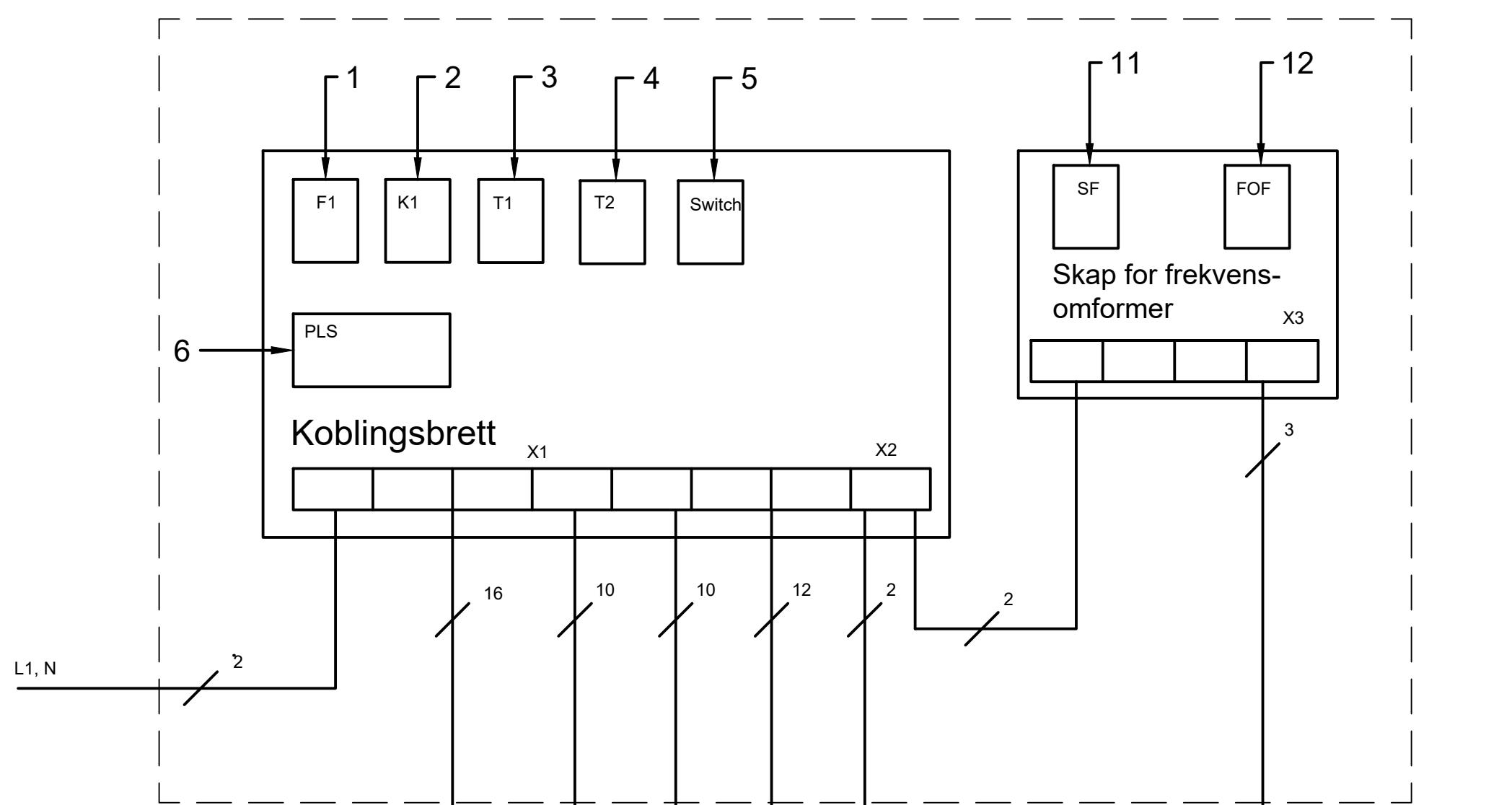


Designed by SHE.engineering	Checked by XXX	Approved by - date XXX - 00/00/00	File name RS232	Date 01/03/16	Scale 1:1
Robotlab				RS232 - Kommunikasjon	
1006		Edition 01		Sheet 1/1	



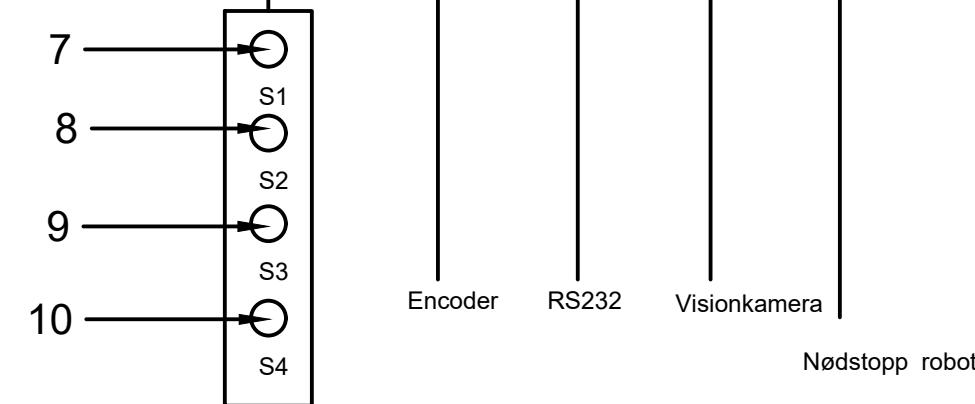


Koblingsskap



Symbol forklaring	
1.	Sikring
2.	Sikkerhetsrele
3.	Trafo 1
4.	Trafo 2
5.	Nettverkswitch
6.	PLS
7.	Hovedbryter
8.	Nødstopp
9.	Start
10.	Stopp
11.	Støyfilter
12.	Frekvensomformer
13.	Motor

Bryterpanel



Itemref	Quantity	Title/Name, designation, material, dimension etc		Article No./Reference
Designed by Eirik Hatland	Checked by Håvard Vikås	Approved by - date XXX - 00/00/00	File name	Date 01/03/2017
Arrangementtegning				
Robotlab		1009	Edition 0	Sheet 1/1

Rekkeklemmetabell								
Kabelside	X3	Apparatside	Kabelside	X1	Apparatside	Kabelside	X2	Apparatside
X1:7	1	SF L	N	1	F1:2	Brown Power supply	1	T1:+24V
X1:8	2	SF N	L	2	F1:4	Blue GND	2	T1:-24V
M:U	3	FOF: U	S1:2	3	F1:1	Pink TRIG	3	Ikke i bruk
M:V	4	FOF: V	S1:4	4	F1:3	Black OUT0	4	PLS - IN(03)
M:W	5	FOF: W	S1:1	5	X1:7	Red IN2	5	Ikke i bruk
K1:24	6	FOF:3	S1:3	6	X1:8	Light blue OUT2	6	Ikke i bruk
K1:14	7	FOF:4	X1:7	7	X3:1	White IN3	7	Ikke i bruk
K1:23,13	8	FOF:P24	X1:6	8	X3:2	Purple IN4	8	Ikke i bruk
		S2:1,S3:2,S4:1		9	T2:+24V	Yellow IN5	9	Ikke i bruk
		S3:4		10	PLS: I-00	Green IN1	10	Ikke i bruk
		S4:1		11	PLS: I-01	Gray IN0	11	Ikke i bruk
		S2:2		12	PLS: I-02	Orange OUT1	12	Ikke i bruk
		S2: robot		13	Lasket til X1:15	Encoder		
		S2: robot		14	Lasket til X1:16	T-Lilla	13	NX-ECO122 Pin A4
		Nødstopp robot		15	Lasket fra X1:13	A-Hvit	14	NX-ECO122 Pin B1
		Nødstopp robot		16	Lasket fra X1:14	D-Gul	15	Ikke i bruk
		S2		17	Lasket fra X1:09	C-Blå	16	NX-ECO122 Pin A2
		S2		18	K1:A1	C-Rosa	17	Ikke i bruk
		S3		19	K1:T31	B-Rød	18	Ikke i bruk
		S3:4		20	K1:T32	B-Svart	19	NX-ECO122 Pin A1
		K1:24		21	X3:6	UB+ - Brun	20	NX-ECO122 Pin A3
		K1:14		22	X3:7	A-Grå	21	Ikke i bruk
		K1:23,13		23	X3:8	D-Grønn	22	Ikke i bruk
				RS-232				
				Sort	23	Ikke i bruk		
				Lilla	24	Ikke i bruk		
				Rosa	25	Ikke i bruk		
				Blå	26	Ikke i bruk		
				Grønn	27	Ikke i bruk		
				Hvit	28	Ikke i bruk		
				Brun	29	Ikke i bruk		
				Grå	30	Ikke i bruk		
				Turkis	31	Ikke i bruk		
				Rød	32	Ikke i bruk		
				Orange	33	Ikke i bruk		
				Gul	34	Ikke i bruk		

Tabell for henvisningspiler		
Fra	Nr	Til
T1 - 0V	1	FQ2 - GND
T1 - 24V	2	FQ2 - 24V
T2 - 0V	3	PLS - (-) & I/O PWR A3
T2 - 24V	4	PLS - (+) & I/O PWR A1
X1:12	5	PLS - IN(05)
X1:11	6	PLS - IN(01)
X1:10	7	PLS - IN(00)
K1:A2	8	PLS - (-)
K2:16	9	X1:13
X1:16	10	K2:A1
PLS - IN(03)	11	X1:4
NX 100 - OUT(03)	12	Y1 - 5/2 ventil
Ikke i bruk	13	Ikke i bruk
Ikke i bruk	14	Ikke i bruk
Ikke i bruk	15	Ikke i bruk
Ikke i bruk	16	Ikke i bruk
Ikke i bruk	17	Ikke i bruk
Ikke i bruk	18	Ikke i bruk
Ikke i bruk	19	Ikke i bruk
Ikke i bruk	20	Ikke i bruk
Ikke i bruk	21	Ikke i bruk
X1:21	22	X3:6
X1:22	23	X3:7
X1:23	24	X3:8



Robotprogram



S.H.E-ENGINEERING
SINDRE HELLINSRUD, HÅVARD VIKÅS OG EIRIK HATLAND

Innhold

VMAIN.....	2
INIT.....	3
LEVER.....	4
LOKKGRIP.....	5
OPENGRIP.....	5
PLUKK.....	6

VMAIN

```
/JOB
//NAME VMAIN
//POS
///NPOS 0,0,0,0,0
//INST
///DATE 2017/05/16 11:04
///ATTR SC,RW
///GROUP1 RB1
NOP
CALL JOB:INIT
CALL JOB:PLUKK IF B001>0
CALL JOB:LEVER IF B001>0
SET B001 0
END
```

INIT

```
/JOB
//NAME INIT
//POS
///NPOS 2,0,0,1,0,0
///TOOL 0
///POSTYPE PULSE
///PULSE
C00000=1,0,0,1,0,0,3
C00001=2,40953,-82779,1,-21618,0,4
///POSTYPE ROBOT
///RECTAN
///RCONF 0,0,0,0,0,0,0
P0002=930.554,0.472,217.055,156.23,-52.09,29.94,0.00
//INST
///DATE 2017/05/16 11:12
///ATTR SC,RW
///GROUP1 RB1
NOP
MOVJ C00000 VJ=10.00
MOVJ C00001 VJ=10.00
MOVJ P002 VJ=10.00
CALL JOB:LUKKGRIP
END
```

LEVER

//NAME LEVER

//POS

///NPOS 0,0,0,4,0,0

///TOOL 0

///POSTYPE ROBOT

///RECTAN

///RCONF 0,0,0,0,0,0,0

P0002=930.554,0.472,217.055,156.23,-52.09,29.94,0.00

P0003=799.568,-227.700,-243.621,-178.43,-1.14,-0.20,0.00

P0004=650.862,-235.536,-246.819,-178.43,-1.14,-0.19,0.00

P0005=504.798,-240.310,-249.865,-178.43,-1.14,-0.19,0.00

//INST

///DATE 2017/05/16 11:03

///ATTR SC,RW

///GROUP1 RB1

NOP

JUMP *DEL1 IF B000=1

JUMP *DEL2 IF B000=2

JUMP *DEL3 IF B000=3

RET IF B001=0

*DEL1

MOVJ P003 VJ=50.00

JUMP *LEVERT

*DEL2

MOVJ P004 VJ=50.00

JUMP *LEVERT

*DEL3

MOVJ P005 VJ=50.00

*LEVERT

CALL JOB:LUKKGRIP

MOVJ P002 VJ=50.00

END

LOKKGRIP

```
/JOB  
//NAME LUKKGRIP  
//POS  
///NPOS 0,0,0,0,0  
//INST  
///DATE 2017/04/28 10:26  
///ATTR SC,RW  
///GROUP1 RB1  
NOP  
DOUT OT#(3) ON  
TIMER T=1.00  
SET B000 1  
END
```

OPENGRIP

```
/JOB  
//NAME OPENGrip  
//POS  
///NPOS 0,0,0,0,0  
//INST  
///DATE 2017/04/28 10:27  
///ATTR SC,RW  
///GROUP1 RB1  
NOP  
DOUT OT#(3) OFF  
TIMER T=1.00  
SET B000 0  
END
```

PLUKK

/JOB

//NAME PLUKK

//POS

///NPOS 0,0,0,3,0,0

///USER 1

///TOOL 0

///POSTYPE USER

///RECTAN

///RCONF 0,0,0,0,0,0,0

P0000=0.000,0.000,0.000,0.00,0.00,0.00,0.00

///POSTYPE ROBOT

P0001=0.000,0.000,-35.000,0.00,0.00,0.00,0.00

P0002=930.554,0.472,217.055,156.23,-52.09,29.94,0.00

//INST

///DATE 2017/05/08 12:57

///ATTR SC,RW

///GROUP1 RB1

NOP

'RETURER OM DEL IKKE TILSTEDE

RET IF B001=0

'GA TIL TRYGG POS

MOVJ P002 VJ=50.00

CALL JOB:LUKKGRIP

'GA TIL DELPOS

MOVJ P000 VJ=50.00

IMOV P001 V=23.0

CALL JOB:OPENGRIP

'GA TIL TRYGG POS

MOVJ P002 VJ=50.00

END



Fagskolen Tinius Olsen
Avdeling for Teknikk og industriell produksjon

Bruksanvisning for robot med visionsystem

Robotlab



Utarbeidet av:

Eirik Hatland

Håvard Vikås

Sindre Hellingsrud

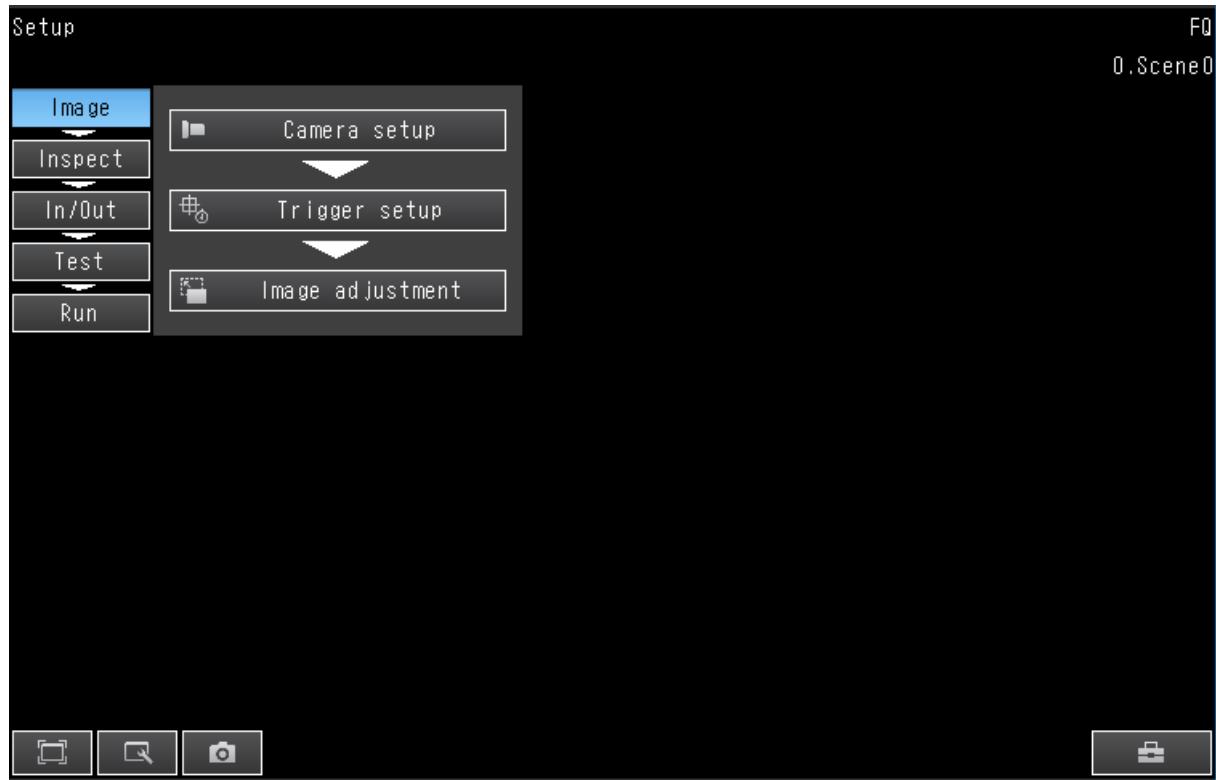
Innhold

Kamera	1
Robot	5

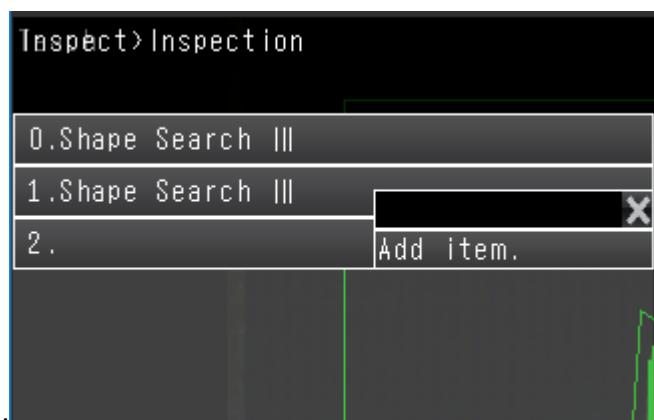
Kamera

For å gjenkjenne deler benytter vi i denne laboppgaven et visionkamera fra Omron. Dette programmeres ved hjelp av et program som heter Touchfinder for PC. Dette programmet ligger på skrivebordet til PCen som hører til laboppgaven. Når programmet startes opp får du valget mellom å koble til et kamera, eller å simulere en oppgave. Her ønsker vi å koble til et kamera (online).

Når kameraet har koblet opp, møter du en skjerm som ser sånn ut:

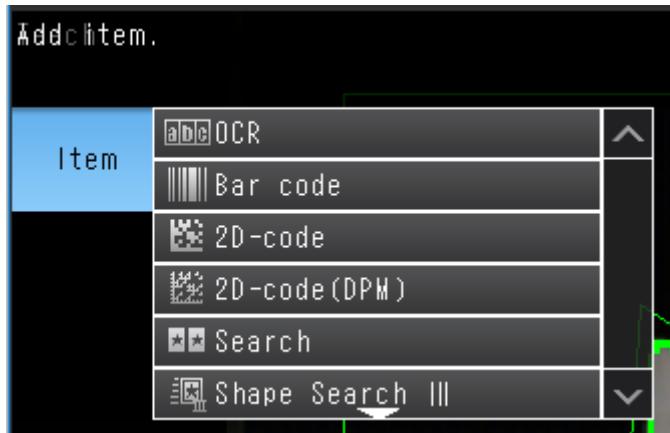


For å endre hvilke former og figurer som detekteres av kameraet må vi gå inn på inspect -> inspection. Her vises hvilke inspeksjoner kameraet er programmert til å benytte seg av. PLSen er satt opp til å hente ut X og Y koordinater fra inspeksjon 0, 1 og 2. Derfor er det hensiktsmessig å slette inspeksjonene som ligger der fra før, og legge til nye. For å endre og legge til oppgaver trykker man



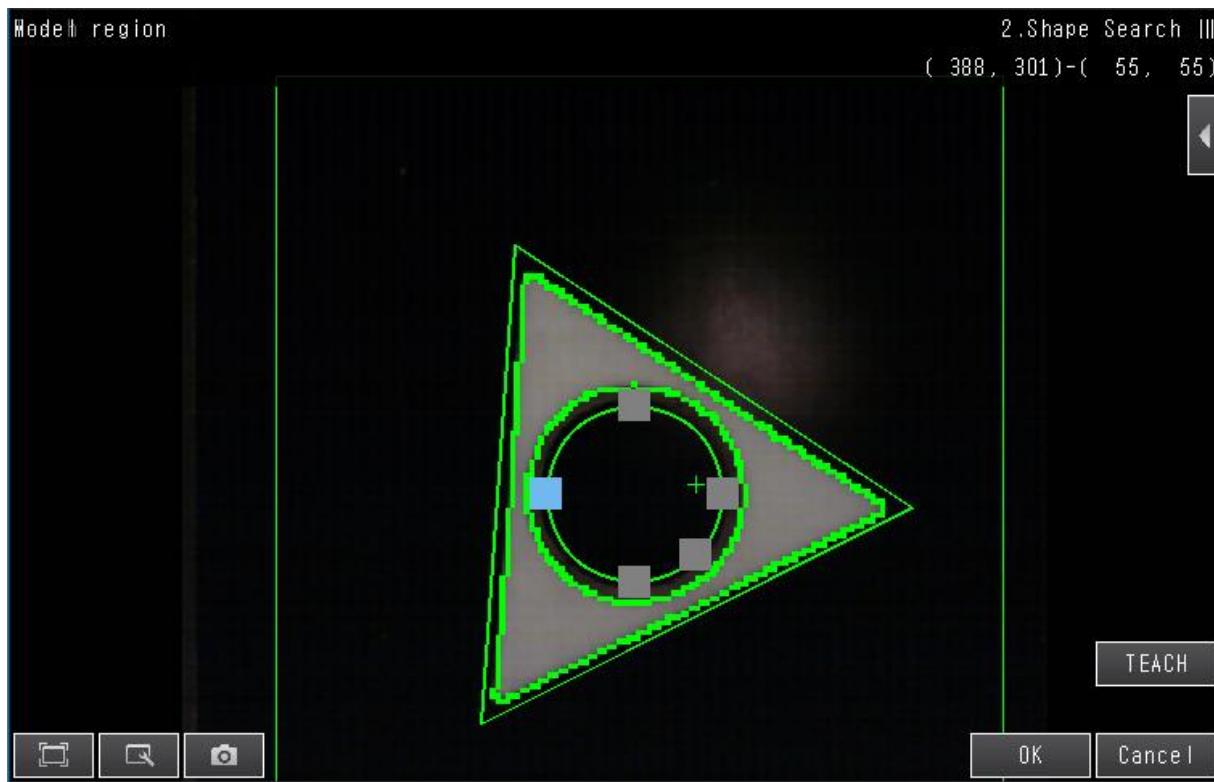
bare på inspeksjonen man vil gjøre noe med:

Når du begynner å lage en ny inspeksjon, må du velge hva du er ute etter å detektere. Vi skal i denne oppgaven søke etter former, og shape search 3 er omrørs raskeste og mest avanserte funksjon for å søke etter former. Vi velger derfor denne:



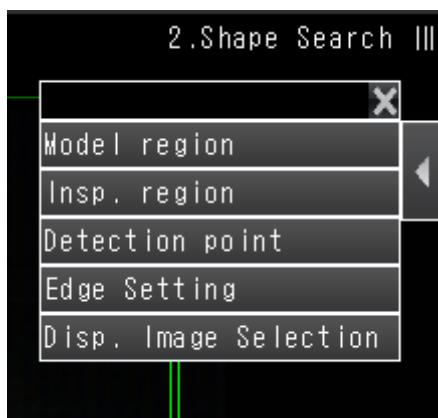
Når vi er inne i den valgte inspeksjonen må vi legge inn noen figurer så vi kan lære kameraet hva det skal se etter. For å legge inn former går man inn på menyen øverst i høyre hjørne, og velger add.



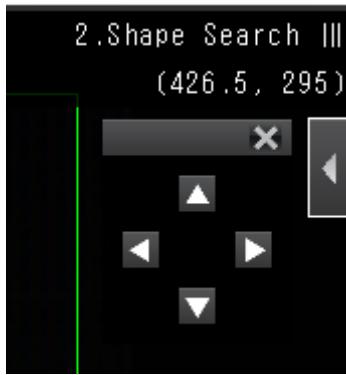


Her er et eksempel på en trekantet figur med et hullet i senter. Hullet skal i denne laboppgaven brukes til å plukke opp, og håndtere delen. Her er det lagt inn en polygon, som omringer delen, og en ellipse som er sentrert i hullet. Når du har lagt inn figurene dine, kan du trykke teach.

+ symbolet som er litt til høyre for senter i trekanten er noe som kalles detection point. Dette er punktet hvor kameraet henter X og Y koordinater fra, så i dette tilfellet må punktet flyttes til senter av hullet manuelt. Vi velger da detection point i menyen som er øverst til høyre, og velger et nytt punkt.

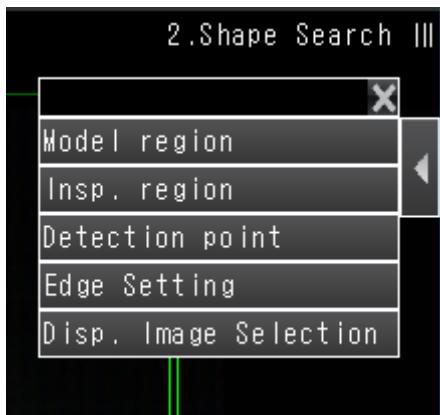


For å velge et punkt med større nøyaktighet, kan du finne console i menyen øverst til høyre. Det vil da dukke opp noen piler du kan trykke på for å flytte punktet en og en pixel.



Når du har flyttet punktet til senter av hullet, kan du trykke på ok.

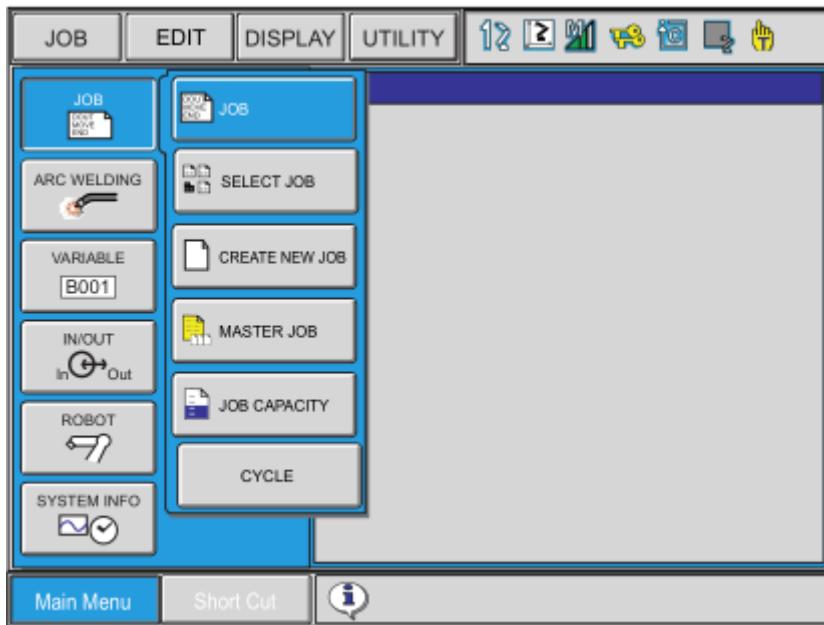
Firkanten som omringer modellområdet, blir kalt inspection region, og denne bør settes så den er like bred som trasportbåndet, og høyden kan være så lang som mulig. For å gjøre det velger du insp. Region i menyen øverst til høyre.



Når du er fornøyd med størrelsen på inspeksjonsområdet trykker du ok. Repeter disse stegene for alle formene du ønsker å inspisere. Du kan nå gå videre til å programmere roboten.

Robot

Roboten som benyttes i denne laboppgaven er en motoman robot, som blir produsert av yaskawa i Japan. Denne styres av en robotcontroller, som heter NX-100, og pendanten vi skal bruke for å programmere roboten finnes på denne styreenheten. For å programmere roboten må vi lage «jobber» som den skal utføre. For å lage en ny jobb, velger du helt enkelt job -> new job.



Jobben må navngis, men er begrenset til 8 karakterer. Det er hensiktsmessig å gi jobbene meningsfulle navn, og å lage jobber som inneholder enkelte istedenfor å ha en lang jobb. PLSen kaller på en jobb som heter VMAIN, så det er hensiktsmessig at hovedjobben heter det. For å programmere roboten, kan vi bruke en del forskjellige instruksjoner. Vi skal i denne oppgaven fokusere på noen av disse. Alle instruksjonene finnes frem ved å trykke på «inform list» knappen på pendanten når du er i en jobb i teach-mode. For å sette inn den valgte funksjonen trykkes man på select -> enter.

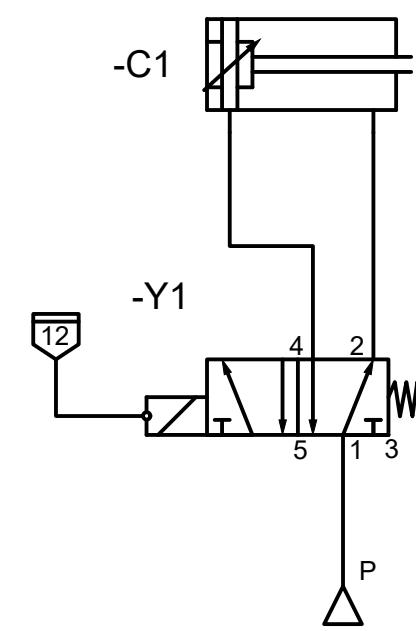
For å sette inn punktposisjoner, kan man kjøre roboten til et punkt, og trykk insert -> enter. For å få kjørt roboten må servo'en være på, og man må holde inne knappen på baksiden av pendanten. Akseknappene kan da benyttes for å bevege roboten i ønsket posisjon.

Instruksjon	Plassering	Beskrivelse
MOVJ	Motion	Beveger roboten til en gitt posisjon på raskest mulig måte. Posisjoner som kan læres bort er enten punktvis eller via variabler i roboten.
IMOV	Motion	Inkrementell bevegelse. Brukes i kombinasjon med P001 for å nærme seg transportbånd.
LABEL	Control	Kan brukes til å sette et punkt i jobb man kan hoppe til. Dette kan være fornuftig om du må lage en loop.
JUMP	Control	Hoppe til en label.
CALL	Control	Kalle på en annen jobb. Når jobben man har kalt på er ferdig, fortsetter programmet der den slapp.
COMMENT	Control	Brukes for å lage en linje med tekst, som kan forklare program.
DOUT	In/out	Man kan skru av eller på digitale utganger fra roboten. I denne øvelsen brukes denne for å åpne og lukke gripperen.
SET	Arith	Brukes for å sette minner til gitte verdier.

Flere av disse instruksjonene kan brukes som hvis-setninger. Dette gjøres ved å gå inn på instruksjonens navn med «select» knappen, og å endre condition til IF. Target settes da til f.eks B001, som blir endret fra PLSen til 1, 2 eller 3 ettersom hvilken del som kommer inn. Det kan da lages en setning som «CALL JOB:PLUKK IF B001>0».

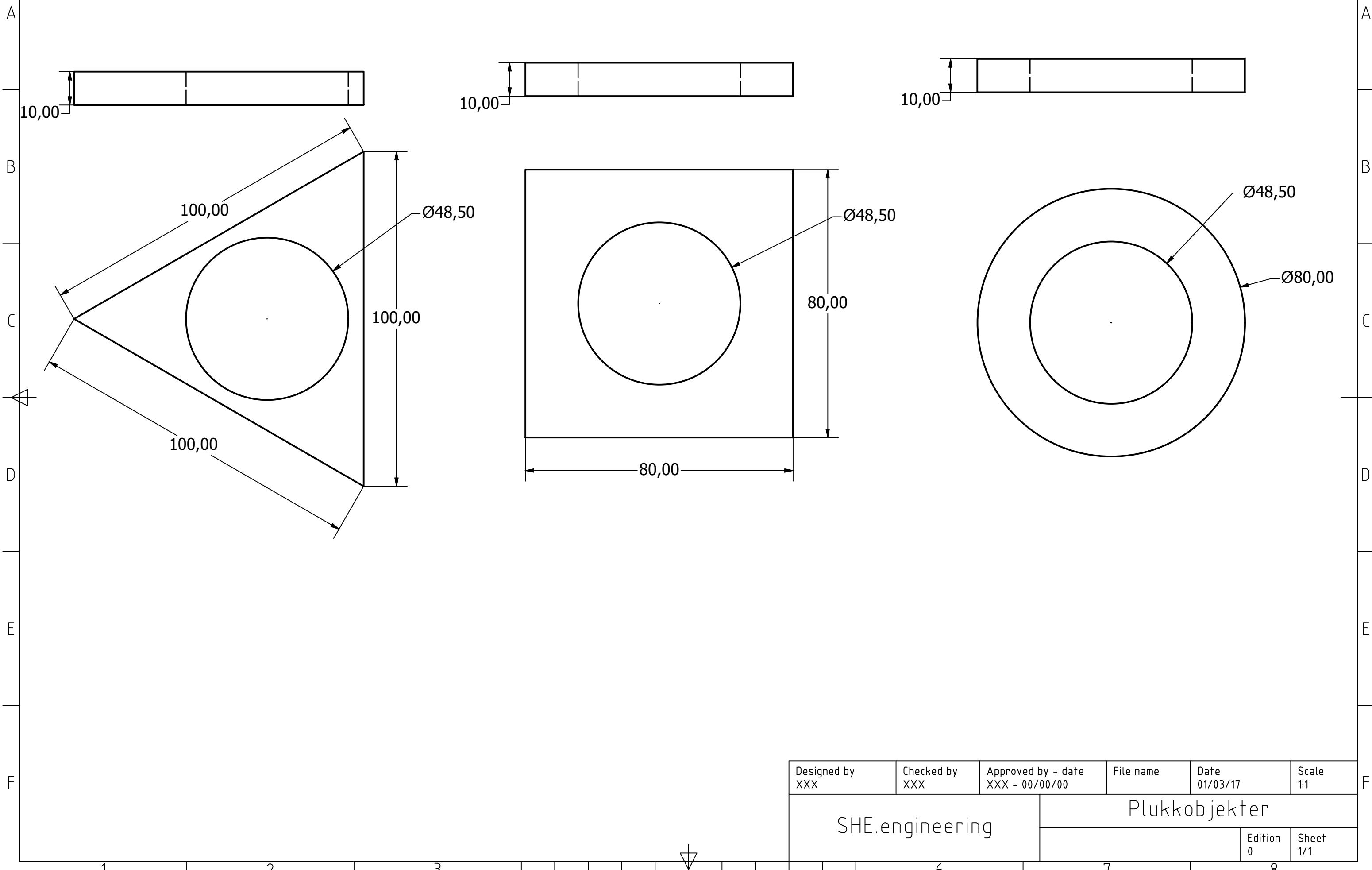
Variablene som blir benyttet av roboten i denne oppgaven er:

P000	Posisjon med bruker-koordinater som blir sendt via kamera til roboten.
P001	Posisjon med en -Z bevegelse for å nærme seg transportbånd inkrementelt.
P002	En trygg posisjon som roboten kan gå til for å unngå kollisjon.
P003	Posisjon for leveranse av del 1
P004	Posisjon for leveranse av del 2
P005	Posisjon for leveranse av del 3
B000	Variabel som kan brukes til å lese av griperens tilstand.
B001	Variabel som forteller hvilken del som er under kamera



-P	Trykkluft tilførsel			
-C1	Pneumatisk griper			
-Y1	5/2 ventil			
Designed by XXX	Checked by XXX	Approved by - date XXX - 00/00/00	File name Pneumatikkjern	Date 01/03/17
XXX		SHE.engineering	Scale N/A	
X			Edition 0	Sheet 1/1

1		2		3		4		5	RevNo	Revision note	6		7		8	Date	Signature	Checked
---	--	---	--	---	--	---	--	---	-------	---------------	---	--	---	--	---	------	-----------	---------

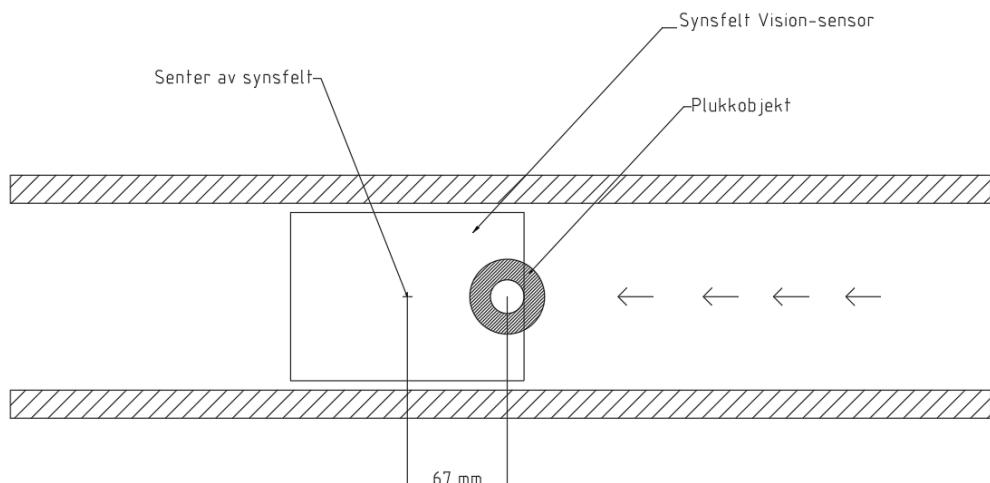


Designed by XXX	Checked by XXX	Approved by - date XXX - 00/00/00	File name	Date 01/03/17	Scale 1:1
SHE.engineering			Plukkobjekter		

Utrengning av nødvendig retardasjonstid

Beskrivelse av tenkt situasjon

Når vision-sensoren har identifisert den innerste sirkelen til objektet vil den sende et signal om at transportbåndet skal stoppe. Det er da ønskelig at objektet befinner seg i senter av synsfeltet når transportbåndet har stoppet. Avstanden fra senter på objektet til senter på synsfeltet er 67mm når stoppsignalet sendes. Retardasjonen er programmert til å være lineær. På bakgrunn av dette er det foretatt en teoretisk utregning av tiden det vil ta fra stoppsignalet sendes til objektet befinner seg i senter av synsfeltet.



Illustasjon av tenkt situasjon

Inndata:

Diameter på trommel	104 mm
Utveksling gir	1/32,5
Synsfelt vision-sensor	Lengde: 170 mm Bredde: 200 mm
Hastighet motor ved 50Hz	1390 RPM

Finner mm forflytning på bånd pr rotasjon på trommel

$$\text{flytning pr. rotasjon} = \text{diameter på trommel} \times \pi = 104 \text{ mm} \times \pi = 326,7 \text{ mm/r}$$

Finner av hastighet på transportbånd ved 50Hz

$$\text{rpm trommel} = \frac{\text{rpm motor}}{\text{utveksling}} = \frac{1390}{32,5} = 42,7 \text{ rpm}$$

$$\text{mm/rotasjon} = \text{flytning pr. rotasjon} \times \text{rpm trommel}$$

$$43 \text{ rpm} \times 326 \text{ mm/r} = 14048 \text{ mm/min} = 234 \text{ mm/s}$$

Finner retardasjon

$$a = \frac{v^2 - v_0^2}{2 \times s} = \frac{234^2 \text{ mm/s}}{2 \times 67 \text{ mm}} = 408,6 \text{ mm/s}^2$$

Finner retardasjonstiden

$$t = \frac{v - v_0}{a} = \frac{234 \text{ mm/s}}{408 \text{ mm/s}^2} = 0,58 \text{ s}$$

Robotlab_FTO

Author: FTO

Created: 25.05.2017 14.03.06

Last Modified: 25.05.2017 14.03.06

Comment:

Sysmac Studio Module Version: 1.1.7.21

Table of Contents

Robotlab_FTO	3
1.PLS	3
1-2.CPU/Expansion Racks	3
1-2-1.NX Bus Configuration and Unit Operation Settings	3
1-3.I/O Map	5
1-4.Controller Setup	11
1-4-2.Built-in EtherNet/IP Port Settings	11
1-5.Motion Control Setup	13
1-5-1.Axis Settings	13
1-5-1-1.MC_Encoder (0,MC1)	13
1-8.POUs	15
1-8-1.Programs	15
1-8-1-1.Program0	15
1-8-1-1-1.Variables	15
1-8-1-1-2.SVON_1	17
1-8-1-1-3.SVON_0	20
1-8-1-1-4.CHCK_STATUS	23
1-8-1-1-5.CHNG_B1	26
1-8-1-1-6.START_JOB	30
1-8-1-1-7.CHNG_P0	34
1-8-1-1-8.TICKS	38
1-8-1-1-9.Bånd	39

1.PLS

1-2.CPU/Expansion Racks

1-2-1.NX Bus Configuration and Unit Operation Settings

Width:178[mm]

NX Unit Number	1	2
NXIP2-1040DT1	NX-PF0730	NX-EC0122
NXBus	N2	N1

Rack Number	NX Unit Number	Model name	Product name	Device name
0	0	NXIP2-1040DT1	NX series controller	NXBus
	1	NX-PF0730	Additional I/O Power Supply	N2
	2	NX-EC0122	Incremental Encoder Input Unit	N1

NXBus

Model name	NXIP2-1040DT1
Product name	NX series controller
Unit version	1.13
NX Unit Number	0
Serial Number	---
Supply Power/Available Power	2.15 / 10.00W
Width	148mm
Number of mounted Units	2
NX Unit Connection Time	3sec
Serial Number Check Method	No check
Fail-soft operation setting	Fail-soft operation

N2

Model name	NX-PF0730
Product name	Additional I/O Power Supply
Unit version	1.0
NX Unit Number	1
NX Unit Mounting Setting	Enabled
Serial Number	0x00000000
Power consumption	0,85W
Width	12mm

N1

Model name	NX-EC0122
Product name	Incremental Encoder Input Unit
Unit version	1.2
NX Unit Number	2
NX Unit Mounting Setting	Enabled

Serial Number	0x00000000																												
Power consumption	1,30W																												
Width	12mm																												
I/O allocation settings	0x7008:01 16[bit] Output Data Set 1/Ch1 Pulse Period Measurement Function 0x7004:01 16[bit] Output Data Set 1/Ch1 Latch Function 0x6000:01 8[bit] Input Data Set 1/Ch1 Encoder Counter Status 0x6001:01 8[bit] Input Data Set 1/Ch1 Reset/External Input Status 0x6002:01 32[bit] Input Data Set 1/Ch1 Encoder Present Position 0x6008:01 8[bit] Input Data Set 1/Ch1 Pulse Period Measurement Status --- 8[bit] Input Data Set 1/Padding 0x6004:01 16[bit] Input Data Set 1/Ch1 Latch Status 0x6005:01 32[bit] Input Data Set 1/Ch1 Latch Input 1 Data 0x6006:01 32[bit] Input Data Set 1/Ch1 Latch Input 2 Data																												
Unit operation settings	<table border="1"> <tr> <td>Counter Type/Ch1 Counter Type</td> <td>Ring Counter</td> </tr> <tr> <td>Maximum Counter Value/Ch1 Maximum Counter Value</td> <td>2147483647</td> </tr> <tr> <td>Minimum Counter Value/Ch1 Minimum Counter Value</td> <td>-2147483648</td> </tr> <tr> <td>Pulse Input Method/Ch1 Pulse Input Method</td> <td>Phase Differential Pulse x4</td> </tr> <tr> <td>Time Window/Ch1 Time Window</td> <td>0</td> </tr> <tr> <td>Average Processing Times/Ch1 Average Processing Times</td> <td>0</td> </tr> <tr> <td>Edge Detection Method/Ch1 Edge Detection Method</td> <td>Disable the Function</td> </tr> <tr> <td>Encoder Count Direction/Ch1 Encoder Count Direction</td> <td>Positive Direction of Phase A</td> </tr> <tr> <td>External Input 0 Function Selection/Ch1 External Input 0 Function Selection</td> <td>General Input</td> </tr> <tr> <td>External Input 0 Logic Selection/Ch1 External Input 0 Logic Selection</td> <td>N.O. (Normaly Open)</td> </tr> <tr> <td>External Input 1 Function Selection/Ch1 External Input 1 Function Selection</td> <td>General Input</td> </tr> <tr> <td>External Input 1 Logic Selection/Ch1 External Input 1 Logic Selection</td> <td>N.O. (Normaly Open)</td> </tr> <tr> <td>External Input 2 Function Selection/Ch1 External Input 2 Function Selection</td> <td>General Input</td> </tr> <tr> <td>External Input 2 Logic Selection/Ch1 External Input 2 Logic Selection</td> <td>N.O. (Normaly Open)</td> </tr> </table>	Counter Type/Ch1 Counter Type	Ring Counter	Maximum Counter Value/Ch1 Maximum Counter Value	2147483647	Minimum Counter Value/Ch1 Minimum Counter Value	-2147483648	Pulse Input Method/Ch1 Pulse Input Method	Phase Differential Pulse x4	Time Window/Ch1 Time Window	0	Average Processing Times/Ch1 Average Processing Times	0	Edge Detection Method/Ch1 Edge Detection Method	Disable the Function	Encoder Count Direction/Ch1 Encoder Count Direction	Positive Direction of Phase A	External Input 0 Function Selection/Ch1 External Input 0 Function Selection	General Input	External Input 0 Logic Selection/Ch1 External Input 0 Logic Selection	N.O. (Normaly Open)	External Input 1 Function Selection/Ch1 External Input 1 Function Selection	General Input	External Input 1 Logic Selection/Ch1 External Input 1 Logic Selection	N.O. (Normaly Open)	External Input 2 Function Selection/Ch1 External Input 2 Function Selection	General Input	External Input 2 Logic Selection/Ch1 External Input 2 Logic Selection	N.O. (Normaly Open)
Counter Type/Ch1 Counter Type	Ring Counter																												
Maximum Counter Value/Ch1 Maximum Counter Value	2147483647																												
Minimum Counter Value/Ch1 Minimum Counter Value	-2147483648																												
Pulse Input Method/Ch1 Pulse Input Method	Phase Differential Pulse x4																												
Time Window/Ch1 Time Window	0																												
Average Processing Times/Ch1 Average Processing Times	0																												
Edge Detection Method/Ch1 Edge Detection Method	Disable the Function																												
Encoder Count Direction/Ch1 Encoder Count Direction	Positive Direction of Phase A																												
External Input 0 Function Selection/Ch1 External Input 0 Function Selection	General Input																												
External Input 0 Logic Selection/Ch1 External Input 0 Logic Selection	N.O. (Normaly Open)																												
External Input 1 Function Selection/Ch1 External Input 1 Function Selection	General Input																												
External Input 1 Logic Selection/Ch1 External Input 1 Logic Selection	N.O. (Normaly Open)																												
External Input 2 Function Selection/Ch1 External Input 2 Function Selection	General Input																												
External Input 2 Logic Selection/Ch1 External Input 2 Logic Selection	N.O. (Normaly Open)																												

1-3.I/O Map

Port	Description	R/W	Data Type	Variable	Variable Comment	Variable Type
EtherCAT Network Configuration						
Node1 3G3AX-MX2-ECT						
Command	This object gives an operation command to the inverter.	W	WORD	Bånd_Command		Global Variables
Frequency reference	This object gives an output frequency command to the inverter.	W	UINT	Bånd_Frequency_reference		Global Variables
Status	This object gives the present state of the unit.	R	WORD	Eoo1_Status		Global Variables
Output frequency monitor	This object gives the output frequency of the inverter.	R	UINT	Eoo1_Output_frequency_monitor		Global Variables
Sysmac Error Status	Indicate Sysmac error status.	R	BYTE	Eoo1_Sysmac_Error_Status		Global Variables
Observation	Observation levels of information is read.	R	BOOL	Eoo1_Observation		Global Variables
Minor Fault	Minor Fault levels of information is read.	R	BOOL	Eoo1_Minor_Fault		Global Variables
CPU/Expansion Racks						
Built-in I/O Built-in I/O Settings						
Input Bit 00	Input Bit 00	R	BOOL	Start_bryter	Starter applikasjonen	Global Variables
Input Bit 01	Input Bit 01	R	BOOL	Stopp_bryter	Stopper applikasjonen	Global Variables
Input Bit 02	Input Bit 02	R	BOOL	Nodstopp_bryter		Global Variables
Input Bit 03	Input Bit 03	R	BOOL			
Input Bit 04	Input Bit 04	R	BOOL			
Input Bit 05	Input Bit 05	R	BOOL			
Input Bit 06	Input Bit 06	R	BOOL			
Input Bit 07	Input Bit 07	R	BOOL			
Input Bit 08	Input Bit 08	R	BOOL			
Input Bit 09	Input Bit 09	R	BOOL			
Input Bit 10	Input Bit 10	R	BOOL			
Input Bit 11	Input Bit 11	R	BOOL			
Input Bit 12	Input Bit 12	R	BOOL			
Input Bit 13	Input Bit 13	R	BOOL			
Input Bit 14	Input Bit 14	R	BOOL			
Input Bit 15	Input Bit 15	R	BOOL			
Input Bit 16	Input Bit 16	R	BOOL			
Input Bit 17	Input Bit 17	R	BOOL			

Input Bit 18	Input Bit 18	R	BOOL			
Input Bit 19	Input Bit 19	R	BOOL			
Input Bit 20	Input Bit 20	R	BOOL			
Input Bit 21	Input Bit 21	R	BOOL			
Input Bit 22	Input Bit 22	R	BOOL			
Input Bit 23	Input Bit 23	R	BOOL			
Output Bit 00	Output Bit 00	RW	BOOL			
Output Bit 01	Output Bit 01	RW	BOOL			
Output Bit 02	Output Bit 02	RW	BOOL			
Output Bit 03	Output Bit 03	RW	BOOL			
Output Bit 04	Output Bit 04	RW	BOOL			
Output Bit 05	Output Bit 05	RW	BOOL			
Output Bit 06	Output Bit 06	RW	BOOL			
Output Bit 07	Output Bit 07	RW	BOOL			
Output Bit 08	Output Bit 08	RW	BOOL			
Output Bit 09	Output Bit 09	RW	BOOL			
Output Bit 10	Output Bit 10	RW	BOOL			
Output Bit 11	Output Bit 11	RW	BOOL			
Output Bit 12	Output Bit 12	RW	BOOL			
Output Bit 13	Output Bit 13	RW	BOOL			
Output Bit 14	Output Bit 14	RW	BOOL			
Output Bit 15	Output Bit 15	RW	BOOL			

OptionBoard Option Board Settings**OptionBoard 1 NX1W-CIFO1**

Node location information	Node location information	R	_sOPTBOARD_ID	OP1_Node_location_information		Global Variables
---------------------------	---------------------------	---	---------------	-------------------------------	--	------------------

NXBusMaster NX Bus Master

N2 NX Unit Registration Status	Status whether the NX Unit is registered to Unit configuration information or not.	R	BOOL	NXBus_N2_NX_Unit_Registration_Status		Global Variables
N2 NX Unit Message Enabled Status	Status whether communications with the NX Unit is possible or not.	R	BOOL	NXBus_N2_NX_Unit_Message_Eabled_Status		Global Variables
N2 NX Unit I/O Data Active Status	Status whether the NX Unit I/O data is controlled or not.	R	BOOL	NXBus_N2_NX_Unit_I_O_Data_Active_Status		Global Variables
N2 NX Unit Error Status	Status whether an error occurs on the NX Unit.	R	BOOL	NXBus_N2_NX_Unit_Error_Status		Global Variables
N2 Time Stamp of Synchronous Input	Time information when the NX Unit has synchronous input refresh (in ns)	R	ULINT	NXBus_N2_Time_Stamp_of_Synchronous_Input		Global Variables
N2 TimeStamp of Synchronous Output	Time information when the NX Unit has synchronous output refresh (in ns)	R	ULINT	NXBus_N2_Time_Stamp_of_Synchronous_Output		Global Variables

N1 NX Unit Registration Status	Status whether the NX Unit is registered to Unit configuration information or not.	R	BOOL	NXBus_N1_NX_Unit_Registration_Status		Global Variables
N1 NX Unit Message Enabled Status	Status whether communications with the NX Unit is possible or not.	R	BOOL	NXBus_N1_NX_Unit_Message_Eabled_Status		Global Variables
N1 NX Unit I/O Data Active Status	Status whether the NX Unit I/O data is controlled or not.	R	BOOL	NXBus_N1_NX_Unit_I_O_Data_Active_Status		Global Variables
N1 NX Unit Error Status	Status whether an error occurs on the NX Unit.	R	BOOL	NXBus_N1_NX_Unit_Error_Status		Global Variables
N1 Time Stamp of Synchronous Input	Time information when the NX Unit has synchronous input refresh (in ns)	R	ULINT	NXBus_N1_Time_Stamp_of_Synchronous_Input		Global Variables
N1 TimeStamp of Synchronous Output	Time information when the NX Unit has synchronous output refresh (in ns)	R	ULINT	NXBus_N1_Time_Stamp_of_Synchronous_Output		Global Variables

Unit1 NX-PF0730**Unit2 NX-EC0122**

Ch1 Encoder Counter Status	Aggregated status of Encoder Counter 1	R	BYTE	N1_Ch1_Encoder_Counter_Status		Global Variables
Ch1 Counter Enabled	Encoder Counter 1 is operating.	R	BOOL	N1_Ch1_Counter_Enabled		Global Variables
Ch1 Internal Reset Completed	Internal reset is completed for Encoder Counter 1.	R	BOOL	N1_Ch1_Internal_Reset_Completed		Global Variables
Ch1 Internal Latch Completed	Internal latch is completed for Encoder Counter 1.	R	BOOL	N1_Ch1_Internal_Latch_Completed		Global Variables
Ch1 Preset Completed	Presetting is completed for Encoder Counter 1.	R	BOOL	N1_Ch1_Preset_Completed		Global Variables
Ch1 Preset Command Value Invalid Flag	Preset value error flag for Encoder Counter 1.	R	BOOL	N1_Ch1_Preset_Command_Value_Invalid_Flag		Global Variables
Ch1 Counter Underflow Flag	Encoder Counter 1 underflow flag	R	BOOL	N1_Ch1_Counter_Underflow_Flag		Global Variables
Ch1 Counter Overflow Flag	Encoder Counter 1 overflow flag	R	BOOL	N1_Ch1_Counter_Overflow_Flag		Global Variables
Ch1 Count Direction Flag	Encoder Counter 1 direction flag	R	BOOL	N1_Ch1_Count_Direction_Flag		Global Variables
Ch1 Reset/External Input	Aggregated status of reset/external	R	BYTE	N1_Ch1_Reset_Exernal_Input_Status		Global Variables

Status	inputs for encoder input 1		tus		
Ch1 External Input 0 Status	External input 0 status for encoder input 1	R	BOOL	N1_Ch1_External_Input_0_Status	Global Variables
Ch1 External Input 1 Status	External input 1 status for encoder input 1	R	BOOL	N1_Ch1_External_Input_1_Status	Global Variables
Ch1 External Input 2 Status	External input 2 status for encoder input 1	R	BOOL	N1_Ch1_External_Input_2_Status	Global Variables
Ch1 External Input Enabled	External input enabled status for encoder input 1	R	BOOL	N1_Ch1_External_Input_Enabled	Global Variables
Ch1 External Reset Enabled	External reset enabled status for encoder input 1	R	BOOL	N1_Ch1_External_Reset_Enabled	Global Variables
Ch1 Phase Z Reset Enabled	Phase Z reset enabled status for encoder input 1	R	BOOL	N1_Ch1_Phase_Z_Reset_Enabled	Global Variables
Ch1 External Reset Completed Flag	External reset completed flag for encoder input 1	R	BOOL	N1_Ch1_External_Reset_Complete_d_Flag	Global Variables
Ch1 Phase Z Reset Completed Flag	Phase Z reset completed flag for encoder input 1	R	BOOL	N1_Ch1_Phase_Z_Reset_Complete_d_Flag	Global Variables
Ch1 Encoder Present Position	Encoder 1 current position	R	DINT	N1_Ch1_Encoder_Present_Position	Global Variables
Ch1 Latch Status	Aggregated latch status of encoder input 1	R	WORD	N1_Ch1_Latch_Status	Global Variables
Ch1 Latch Input 1 Enabled	Latch input 1 enabled or disabled status of encoder input 1	R	BOOL	N1_Ch1_Latch_Input_1_Enabled	Global Variables
Ch1 Latch Input 1 Completed Flag	Latch input 1 completed flag of encoder input 1	R	BOOL	N1_Ch1_Latch_Input_1_Completed_Flag	Global Variables
Ch1 Latch Input 2 Enabled	Latch input 2 enabled or disabled status of encoder input 1	R	BOOL	N1_Ch1_Latch_Input_2_Enabled	Global Variables
Ch1 Latch Input 2 Completed Flag	Latch input 2 completed flag of encoder input 1	R	BOOL	N1_Ch1_Latch_Input_2_Completed_Flag	Global Variables
Ch1 Latch Input 1 Data	The value latched by latch input 1 of external input/phase Z input of encoder input 1	R	DINT	N1_Ch1_Latch_Input_1_Data	Global Variables
Ch1 Latch Input 2 Data	The value latched by latch input 2 of external input/phase Z input of encoder input 1	R	DINT	N1_Ch1_Latch_Input_2_Data	Global Variables
Ch1 Pulse Period Measurement	Aggregated status of pulse period	R	BYTE	N1_Ch1_Pulse_Period_Measurement	Global Variables

Status	measurements of encoder input 1		t_Status			
Ch1 Pulse Period Measurement Enabled	Pulse period measurement for encoder input 1 is enabled.	R	BOOL	N1_Ch1_Pulse_Period_Measurement_Enabled		Global Variables
Ch1 Pulse Period Measurement Value Clear Completed	Pulse period measurement value for encoder input 1 is cleared.	R	BOOL	N1_Ch1_Pulse_Period_Measurement_Value_Clear_Completed		Global Variables
Ch1 Pulse Period Measurement Value Overflow Flag	Pulse period measurement value for encoder input 1 is overflowed.	R	BOOL	N1_Ch1_Pulse_Period_Measurement_Value_Overflow_Flag		Global Variables
Ch1 Latch Function	Aggregated data of latch function for encoder input 1	W	WORD			
Ch1 Latch Input 1 Enable	Enable or disable the latch input 1 for encoder input 1	W	BOOL			
Ch1 Latch Input 1 Trigger Condition	Latch input 1 trigger condition for encoder input 1	W	BOOL			
Ch1 Latch Input 1 Trigger Selection	Toggle the external input and the phase Z input for the latch input 1 of encoder input 1	W	BOOL			
Ch1 Latch Input 2 Enable	Enable or disable the latch input 2 for encoder input 1	W	BOOL			
Ch1 Latch Input 2 Trigger Condition	Latch input 2 trigger condition for encoder input 1	W	BOOL			
Ch1 Latch Input 2 Trigger Selection	Toggle the external input and the phase Z input for the latch input 2 of encoder input 1	W	BOOL			
Ch1 Pulse Period Measurement Function	Aggregated byte data of the Pulse Period Measurement Function for encoder input 1	W	WORD	N1_Ch1_Pulse_Period_Measurement_Function		Global Variables
Ch1 Pulse Period Measurement Enable	Enable the Pulse Period Measurement Function for encoder input 1	W	BOOL	N1_Ch1_Pulse_Period_Measurement_Enable		Global Variables
Ch1 Pulse Period Measurement Value Clear	Clear the pulse period measurement value for encoder input 1	W	BOOL	N1_Ch1_Pulse_Period_Measurement_Value_Clear		Global Variables

Ch1 Pulse Period Measurement Value Overflow Flag Clear	Clear the pulse period measurement overflow flag for encoder input 1	W	BOOL	N1_Ch1_Pulse_Period_Measurement_Value_Overflow_Flag_Clear		Global Variables
--	--	---	------	---	--	------------------

1-4.Controller Setup**1-4-2.Built-in EtherNet/IP Port Settings**

Built-in EtherNet/IP Port Settings

TCP/IP Settings	
IP address settings	Fixed setting
IP address	192.168.250.1
Subnet mask	255.255.255.0
DNS	Do not use
Priority DNS server	
Secondary DNS server	
Domain name	
Host Name - IP Address	No settings
Keep Alive	Use
Keep Alive monitoring time	10 sec
Linger option	Do not specify
Default gateway	
IP Router Table	No settings
LINK Settings	
LINK settings	Auto
FTP Settings	
FTP server	Do not use
Port No.	21
Login name	
Password	
NTP Settings	
NTP server clock information	Do not get
Port No.	123
Server specifying method	IP address
IP address	
Host name	
NTP operation timing	Specify a time.
Time	00:00:00
Interval	60 min
Timeout time	10 sec
SNMP Settings	
SNMP service	Do not use
Port No.	161
Address	
Location	
Send a recognition trap.	FALSE
Recognition method	IP address
IP address	0.0.0.0
Host name	
Community name	public
Recognition 2	Do not use
Recognition method	IP address
IP address	0.0.0.0
Host name	
Community name	public
SNMP Trap Settings	
SNMP trap	Do not use

Port No.	162
Specifying method	IP address
IP address	
Host name	
Community name	public
Version	SNMPv2C
Trap 2	Do not use
Specifying method	IP address
IP address	
Host name	
Community name	public
Version	SNMPv2C
FINS Settings	
Node address of built-in EtherNet/IP port	1
FINS/UDP port number	9600
IP address-FINS address conversion method	Automatic generation
IP Address Table	No settings
Dynamically change remote IP	Convert

1-5.Motion Control Setup

1-5-1.Axis Settings

1-5-1-1.MC Encoder (0,MC1)

Axis Parameters

Axis Name	MC_Encoder (0,MC1)
Axis Basic Settings	
Motion control	MC1: Primary periodic task
Axis use	Used axis
Axis type	Encoder axis
Control function	All
Feedback control	No control loop
Input device 1	CPU Rack, Unit : 2 NX-EC0122(N1)
Channel	1
Input device 2	<Not assigned>
Channel	
Input device 3	<Not assigned>
Channel	
Output device 1	<Not assigned>
Channel	
Output device 2	<Not assigned>
Channel	
Output device 3	<Not assigned>
Channel	
Unit Conversion Settings	
Unit of display	pulse
Command pulse count per motor rotation	1024 pulse/rev
Gearbox usage	Do not use gearbox
Work travel distance per motor rotation	10.06 pulse/rev
Work travel distance per work rotation	10000 pulse/rev
Work gear ratio (Numerator of the reduction ratio)	1
Motor gear ratio (Denominator of the reduction ratio)	1
Operation Settings	
Maximum velocity	400000000 pulse/s
Velocity warning value	0 %
Start velocity	0 pulse/s
Maximum jog velocity	1000000 pulse/s
Maximum acceleration	0 pulse/s^2
Acceleration warning value	0 %
Maximum deceleration	0 pulse/s^2
Deceleration warning value	0 %
Acceleration/deceleration over	Use rapid acceleration/deceleration (Blending is changed to Buffered)
Operation selection at Reversing	Deceleration stop
Positive torque warning value	0 %
Negative torque warning value	0 %
In-position range	10 pulse
In-position check time	0 ms
Actual velocity filter time constant	0 ms
Zero position range	10 pulse
Other Operation Settings	
Immediate stop input stop method	Immediate stop
Limit input stop method	Immediate stop

Drive error reset monitoring time	200 ms
Maximum positive torque limit	300.0 %
Maximum negative torque limit	300.0 %
Immediate stop input logic inversion	Do not invert
Positive limit input logic inversion	Do not invert
Negative limit input logic inversion	Do not invert
Home proximity input logic inversion	Do not invert
Limit Settings	
Software limits	Disabled
Positive software limit	2147483647 pulse
Negative software limit	-2147483648 pulse
Following error over value	0 pulse
Following error warning value	0 pulse
Homing Settings	
Homing method	Zero position preset
Home input signal	Use Z-phase input as home
Homing start direction	Positive direction
Operation selection at positive limit input	Reverse turn/immediate stop
Home input detection direction	Positive direction
Operation selection at negative limit input	Reverse turn/immediate stop
Homing velocity	10000 pulse/s
Homing approach velocity	1000 pulse/s
Homing acceleration	0 pulse/s^2
Homing deceleration	0 pulse/s^2
Homing jerk	0 pulse/s^3
Home input mask distance	10000 pulse
Home offset	0 pulse
Homing holding time	100 ms
Homing compensation value	0 pulse
Homing compensation velocity	1000 pulse/s
Position Count Settings	
Count mode	Linear mode
Modulo maximum position setting value	2147483647 pulse
Modulo minimum position setting value	-2147483648 pulse
Encoder type	Incremental encoder
Servo Drive Settings	
PDS state control method	Switched on by Servo OFF
Main circuit power supply OFF detection	Detect

1-8.POUS**1-8-1.Programs****1-8-1-1.Program0****1-8-1-1-1.Variables**

Name	Data Type	Initial Value	AT	Retain	Constant	Comment
VAR						
stage	INT			False	False	
dotcp	BOOL			False	False	
SktTCPConnect_instance	SktTCPConnect			False	False	
SktClearBuf_instance	SktClearBuf			False	False	
SktGetTCPStatus_instance	SktGetTCPStatus			False	False	
SktTCPSend_instance	SktTCPSend			False	False	
SktTCPRecv_instance	SktTCPRecv			False	False	
SktClose_instance	SktClose			False	False	
SendSocketDat	ARRAY[0..255] OF byte			False	False	
RevSocketDat	ARRAY[0..255] OF byte			False	False	
WkSocket	sSOCKET	(Handle:=0, SrcAdr:= (PortNo:=0, IpAdr:="), DstAdr:= (PortNo:=0, IpAdr:="))		False	False	
SendAry	INT			False	False	
SVON_0_RUN	BOOL			False	False	
SVON_1_RUN	BOOL			False	False	
test_size	uINT			False	False	
CHECK_STAT_US_RUN	BOOL			False	False	
CHECK_READ_Y	BOOL			False	False	
CHNG_B1	BOOL			False	False	
B1	INT			False	False	BYTE001 fra robot
SVON_1	BOOL			False	False	
SVON_0	BOOL			False	False	
CHNG_B1_RU_N	BOOL			False	False	
timer_båndstopp	TON			False	False	
str_del	STRING[10]			False	False	
CHNG_P0	BOOL			False	False	
CHNG_P0_RU_N	BOOL			False	False	
START_JOB_RUN	BOOL			False	False	
START_JOB	BOOL			False	False	
Robot_ready	BOOL			False	False	
posisjon_x	dINT			False	False	
posisjon_y	dINT			False	False	
drift	BOOL			False	False	
timer_bånd_fortsett	TON			False	False	
r_trig1	R_TRIG			False	False	
r_trig2	R_TRIG			False	False	
CHECK_STAT	BOOL			False	False	

US						
VAR_EXTERNAL						
trigger	BOOL				False	
EIP1_EtnOnlin eSta	BOOL				True	
stopp	BOOL				False	
Start_bryter	BOOL				False	
Stopp_bryter	BOOL				False	
Bånd_Command	WORD				False	
Bånd_Frequenc y_reference	UINT				False	
EIPInput	S_EIPInput				False	
EIPOutput	S_EIPOutput				False	
Tick	BOOL				False	
Nodstopp_bryter	BOOL				False	

1-8-1-1-2.SVON_1

```
0      1 // Start sequence when Trigger changes to TRUE.
1 IF ( (SVON_1=TRUE) AND (SVON_1_RUN=FALSE) AND
2     (EIP1_EtnOnlineSta=TRUE) ) THEN
3   SVON_1_RUN :=TRUE;
4   Stage :=INT#1;
5   SktTCPConnect_instance(Execute:=FALSE); // Initialize instance.
6   SktClearBuf_instance(Execute:=FALSE); // Initialize instance.
7   SktGetTCPStatus_instance(Execute:=FALSE); // Initialize instance.
8   SktTCPSend_instance(
9     Execute:=FALSE,
10    SendDat:=SendSocketDat[0]); // Dummy
11  SktTCPRecv_instance(           // Initialize instance.
12    Execute:=FALSE,
13    RcvDat :=RcvSocketDat[0]); // Dummy
14  SktClose_instance(Execute:=FALSE); // Initialize instance.
15 END_IF;
16
17 IF Tick then
18   IF (SVON_1_RUN=TRUE) THEN
19     CASE Stage OF
20       1: // Request a connection.
21         SktTCPConnect_instance(
22           Execute :=TRUE,
23           SrcTcpPort:=UINT#0, // Local TCP port number: Automatically
24           assigned.
25           DstAddr :='192.168.250.20', // Remote IP address
26           DstTcpPort:=UINT#80, // Destination TCP port number
27           Socket =>WkSocket); // Socket
28
29   IF (SktTCPConnect_instance.Done=TRUE) THEN
30     Stage:=INT#2; // Normal end
31   ELSIF (SktTCPConnect_instance.Error=TRUE) THEN
32     Stage:=INT#10; // Error end
33   END_IF;
34
35   2: // Clear receive buffer.
36   SktClearBuf_instance(
37     Execute:=TRUE,
38     Socket :=WkSocket); // Socket
39
40   IF (SktClearBuf_instance.Done=TRUE) THEN
41     Stage:=INT#3; // Normal end
42   ELSIF (SktClearBuf_instance.Error=TRUE) THEN
43     Stage:=INT#20; // Error end
44   END_IF;
45
46   3: // Request reading status.
47   SktGetTCPStatus_instance(
48     Execute:=TRUE,
49     Socket :=WkSocket); // Socket
50
51   IF (SktGetTCPStatus_instance.Done=TRUE) THEN
52     Stage:=INT#4; // Normal end
53   ELSIF (SktGetTCPStatus_instance.Error=TRUE) THEN
54     Stage:=INT#30; // Error end
55   END_IF;
56
57   4: // Request sending data
58   StringToAry('CONNECT Robot_access',SendSocketDat[0]);
59   SendSocketDat[20]:=16#0D;
60   SendSocketDat[21]:=16#0A;
```

```
68 SktTCPSend_instance(
69   Execute:=TRUE,
70   Socket :=WkSocket; // Socket
71   SendDat:=SendSocketDat[0], // Send data
72   Size :=22); // Send data size
73
74 IF (SktTCPSend_instance.Done=TRUE) THEN
75   Stage:=INT#5; // Normal end
76   SktTCPSend_instance(
77     Execute:=FALSE,
78     SendDat:=SendSocketDat[0]); // Dummy
79 ELSIF (SktTCPSend_instance.Error=TRUE) THEN
80   Stage:=INT#40; // Error end
81 END_IF;
82
83 5: // Request receiving data
84   SktTCPRecv_instance(
85     Execute:=TRUE,
86     Socket :=WkSocket; // Socket
87     TimeOut:=UINT#1000, // Timeout time
88     Size :=UINT#250, // Receive data size
89     RcvDat :=RcvSocketDat[0]); // Receive data
90
91 IF (SktTCPRecv_instance.Done=TRUE) THEN
92   Stage:=INT#6; // Normal end
93   SktTCPRecv_instance( // Initialize instance.
94     Execute:=FALSE,
95     RcvDat :=RcvSocketDat[0]); // Dummy
96 ELSIF (SktTCPRecv_instance.Error=TRUE) THEN
97   Stage:=INT#50; // Error end
98 END_IF;
99
100 6: // Request sending data
101   StringToAry('HOSTCTRL_REQUEST SVON 2',SendSocketDat[0]);
102   SendSocketDat[23]:=16#0D://CR
103   SendSocketDat[24]:=16#0A://LF
104   SktTCPSend_instance(
105     Execute:=TRUE,
106     Socket :=WkSocket; // Socket
107     SendDat:=SendSocketDat[0], // Send data
108     Size :=25); // Send data size
109   test_size:=SizeOfAry(SendSocketDat);
110
111 IF (SktTCPSend_instance.Done=TRUE) THEN
112   Stage:=INT#7; // Normal end
113   SktTCPSend_instance(
114     Execute:=FALSE,
115     SendDat:=SendSocketDat[0]); // Dummy
116 ELSIF (SktTCPSend_instance.Error=TRUE) THEN
117   Stage:=INT#60; // Error end
118 END_IF;
119
120 7: // Request receiving data
121   SktTCPRecv_instance(
122     Execute:=TRUE,
123     Socket :=WkSocket; // Socket
124     TimeOut:=UINT#1000, // Timeout time
125     Size :=UINT#250, // Receive data size
126     RcvDat :=RcvSocketDat[0]); // Receive data
127
128 IF (SktTCPRecv_instance.Done=TRUE) THEN
129   Stage:=INT#8; // Normal end
130   SktTCPRecv_instance( // Initialize instance
```

```
123      SktTCPRecv_instance(          // Normal instance
124      Execute:=FALSE;
125      RcvDat :=RcvSocketDat[0];      // Dummy
126      ELSIF (SktTCPRecv_instance.Error=TRUE) THEN
127          Stage:=INT#70;           // Error end
128          END_IF;
129
130      8: // Request sending data
131      StringToAry('1',SendSocketDat[0]);
132      SendSocketDat[1]:=16#0D; //CR
133      //SendSocketDat[2]:=16#0A; //LF
134      SktTCPSend_instance(
135          Execute:=TRUE,
136          Socket :=WkSocket,        // Socket
137          SendDat:=SendSocketDat[0], // Send data
138          Size :=3);             // Send data size
139      IF (SktTCPSend_instance.Done=TRUE) THEN
140          Stage:=INT#15;           // Normal end
141          SktTCPSend_instance(
142              Execute:=FALSE,
143              SendDat:=SendSocketDat[0]); // Dummy
144          ELSIF (SktTCPSend_instance.Error=TRUE) THEN
145              Stage:=INT#80;           // Error end
146          END_IF;
147
148      15: // Request closing.
149      SktClose_instance(
150          Execute:=TRUE,
151          Socket :=WkSocket);     // Socket
152
153      IF (SktClose_instance.Done=TRUE) THEN
154          Stage:=INT#0;            // Normal end
155      ELSIF (SktClose_instance.Error=TRUE) THEN
156          Stage:=INT#150;          // Error end
157          END_IF;
158
159      0: // Normal end
160      SVON_1_RUN :=FALSE;
161      trigger:=FALSE;
162
163      ELSE // Interrupted by error.
164      SVON_1_RUN :=FALSE;
165      trigger:=FALSE;
166      END_CASE;
167
168      END_IF;
169
170  IF not SktTCPSend_instance.Execute THEN
171    FOR SendAry:=int#0 TO int#250 BY INT#1 do
172      SendSocketDat[SendAry]:=16#00;
173    END_FOR;
174  END_IF;
175
```

1-8-1-1-3.SVON_0

```
0      1 // Start sequence when Trigger changes to TRUE.
1 IF ( (SVON_0=TRUE) AND (SVON_0_RUN=FALSE) AND
2     (EIP1_EtnOnlineSta=TRUE) ) THEN
3   SVON_0_RUN :=TRUE;
4   Stage :=INT#1;
5   SktTCPConnect_instance(Execute:=FALSE); // Initialize instance.
6   SktClearBuf_instance(Execute:=FALSE); // Initialize instance.
7   SktGetTCPStatus_instance(Execute:=FALSE); // Initialize instance.
8   SktTCPSend_instance(
9     Execute:=FALSE,
10    SendDat:=SendSocketDat[0]); // Dummy
11  SktTCPRecv_instance(           // Initialize instance.
12    Execute:=FALSE,
13    RcvDat :=RcvSocketDat[0]); // Dummy
14  SktClose_instance(Execute:=FALSE); // Initialize instance.
15 END_IF;
16
17 IF Tick then
18   IF (SVON_0_RUN=TRUE) THEN
19     CASE Stage OF
20       1: // Request a connection.
21         SktTCPConnect_instance(
22           Execute :=TRUE,
23           SrcTcpPort:=UINT#0, // Local TCP port number: Automatically
24           assigned.
25           DstAddr :='192.168.250.20', // Remote IP address
26           DstTcpPort:=UINT#80, // Destination TCP port number
27           Socket =>WkSocket); // Socket
28
29   IF (SktTCPConnect_instance.Done=TRUE) THEN
30     Stage:=INT#2; // Normal end
31   ELSIF (SktTCPConnect_instance.Error=TRUE) THEN
32     Stage:=INT#10; // Error end
33   END_IF;
34
35   2: // Clear receive buffer.
36   SktClearBuf_instance(
37     Execute:=TRUE,
38     Socket :=WkSocket); // Socket
39
40   IF (SktClearBuf_instance.Done=TRUE) THEN
41     Stage:=INT#3; // Normal end
42   ELSIF (SktClearBuf_instance.Error=TRUE) THEN
43     Stage:=INT#20; // Error end
44   END_IF;
45
46   3: // Request reading status.
47   SktGetTCPStatus_instance(
48     Execute:=TRUE,
49     Socket :=WkSocket); // Socket
50
51   IF (SktGetTCPStatus_instance.Done=TRUE) THEN
52     Stage:=INT#4; // Normal end
53   ELSIF (SktGetTCPStatus_instance.Error=TRUE) THEN
54     Stage:=INT#30; // Error end
55   END_IF;
56
57   4: // Request sending data
58   StringToAry('CONNECT Robot_access',SendSocketDat[0]);
59   SendSocketDat[20]:=16#0D;
60   SendSocketDat[21]:=16#0A;
```

```
68 SktTCPSend_instance(
69   Execute:=TRUE,
70   Socket :=WkSocket      // Socket
71   SendDat:=SendSocketDat[0], // Send data
72   Size  :=22);    // Send data size
73
74 IF (SktTCPSend_instance.Done=TRUE) THEN
75   Stage:=INT#5;          // Normal end
76   SktTCPSend_instance(
77     Execute:=FALSE,
78     SendDat:=SendSocketDat[0]); // Dummy
79 ELSIF (SktTCPSend_instance.Error=TRUE) THEN
80   Stage:=INT#40;         // Error end
81 END_IF;
82
83 5: // Request receiving data
84   SktTCPRecv_instance(
85     Execute:=TRUE,
86     Socket :=WkSocket      // Socket
87     TimeOut:=UINT#1000,    // Timeout time
88     Size  :=UINT#250,     // Receive data size
89     RcvDat :=RcvSocketDat[0]); // Receive data
90
91 IF (SktTCPRecv_instance.Done=TRUE) THEN
92   Stage:=INT#6;          // Normal end
93   SktTCPRecv_instance(           // Initialize instance.
94     Execute:=FALSE,
95     RcvDat :=RcvSocketDat[0]); // Dummy
96 ELSIF (SktTCPRecv_instance.Error=TRUE) THEN
97   Stage:=INT#50;         // Error end
98 END_IF;
99
100 6: // Request sending data
101   StringToAry('HOSTCTRL_REQUEST SVON 2',SendSocketDat[0]);
102   SendSocketDat[23]:=16#0D://CR
103   SendSocketDat[24]:=16#0A://LF
104   SktTCPSend_instance(
105     Execute:=TRUE,
106     Socket :=WkSocket      // Socket
107     SendDat:=SendSocketDat[0], // Send data
108     Size  :=25);    // Send data size
109     test_size:=SizeOfAry(SendSocketDat);
110
111 IF (SktTCPSend_instance.Done=TRUE) THEN
112   Stage:=INT#7;          // Normal end
113   SktTCPSend_instance(
114     Execute:=FALSE,
115     SendDat:=SendSocketDat[0]); // Dummy
116 ELSIF (SktTCPSend_instance.Error=TRUE) THEN
117   Stage:=INT#60;         // Error end
118 END_IF;
119
120
121 7: // Request receiving data
122   SktTCPRecv_instance(
123     Execute:=TRUE,
124     Socket :=WkSocket      // Socket
125     TimeOut:=UINT#1000,    // Timeout time
126     Size  :=UINT#250,     // Receive data size
127     RcvDat :=RcvSocketDat[0]); // Receive data
128
129 IF (SktTCPRecv_instance.Done=TRUE) THEN
130   Stage:=INT#8;          // Normal end
131   SktTCPRecv_instance(           // Initialize instance
```

```
123:          SktTCPRecv_instance;           // Common instance
124:          Execute:=FALSE;
125:          RcvDat :=RcvSocketDat[0];        // Dummy
126:          ELSIF (SktTCPRecv_instance.Error=TRUE) THEN
127:              Stage:=INT#70;               // Error end
128:          END_IF;
129:
130:          8: // Request sending data
131:          StringToAry('0',SendSocketDat[0]);
132:          SendSocketDat[1]:=16#0D; //CR
133:          //SendSocketDat[2]:=16#0A; //LF
134:          SktTCPSend_instance(
135:              Execute:=TRUE,
136:              Socket :=WkSocket,      // Socket
137:              SendDat:=SendSocketDat[0], // Send data
138:              Size :=3);             // Send data size
139:          IF (SktTCPSend_instance.Done=TRUE) THEN
140:              Stage:=INT#15;           // Normal end
141:              SktTCPSend_instance(
142:                  Execute:=FALSE,
143:                  SendDat:=SendSocketDat[0]); // Dummy
144:              ELSIF (SktTCPSend_instance.Error=TRUE) THEN
145:                  Stage:=INT#80;           // Error end
146:              END_IF;
147:
148:          15: // Request closing.
149:          SktClose_instance(
150:              Execute:=TRUE,
151:              Socket :=WkSocket);     // Socket
152:
153:          IF (SktClose_instance.Done=TRUE) THEN
154:              Stage:=INT#0;            // Normal end
155:          ELSIF (SktClose_instance.Error=TRUE) THEN
156:              Stage:=INT#150;          // Error end
157:          END_IF;
158:
159:          0: // Normal end
160:          SVON_0_RUN :=FALSE;
161:          trigger:=FALSE;
162:
163:          ELSE // Interrupted by error.
164:              SVON_0_RUN :=FALSE;
165:              trigger:=FALSE;
166:          END_CASE;
167:
168:      END_IF;
169:  END_IF;
170:  IF not SktTCPSend_instance.Execute THEN
171:      FOR SendAry:=int#0 TO int#250 BY INT#1 do
172:          SendSocketDat[SendAry]:=16#00;
173:      END_FOR;
174:  END_IF;
```

1-8-1-1-4.CHCK_STATUS

```

0
1 // Start sequence when Trigger changes to TRUE.
2 IF ( (CHECK_STATUS=TRUE) AND (CHECK_STATUS_RUN=FALSE) AND
3     (EIP1_EtnOnlineSta=TRUE) ) THEN
4     CHECK_STATUS_RUN :=TRUE;
5     Stage :=INT#1;
6     SktTCPConnect_instance(Execute:=FALSE); // Initialize instance.
7     SktClearBuf_instance(Execute:=FALSE); // Initialize instance.
8     SktGetTCPStatus_instance(Execute:=FALSE); // Initialize instance.
9     SktTCPSend_instance(
10        Execute:=FALSE,
11        SendDat:=SendSocketDat[0]); // Dummy
12     SktTCPRecv_instance(           // Initialize instance.
13        Execute:=FALSE,
14        RcvDat :=RcvSocketDat[0]); // Dummy
15     SktClose_instance(Execute:=FALSE); // Initialize instance.
16 END_IF;
17
18 IF Tick then
19   IF (CHECK_STATUS_RUN=TRUE) THEN
20     CASE Stage OF
21       1: // Request a connection.
22         SktTCPConnect_instance(
23           Execute :=TRUE,
24           SrcTcpPort:=UINT#0, // Local TCP port number: Automatically
25           assigned.
26           DstAddr :='192.168.250.20', // Remote IP address
27           DstTcpPort:=UINT#80, // Destination TCP port number
28           Socket =>WkSocket); // Socket
29
30       IF (SktTCPConnect_instance.Done=TRUE) THEN
31         Stage:=INT#2; // Normal end
32       ELSIF (SktTCPConnect_instance.Error=TRUE) THEN
33         Stage:=INT#10; // Error end
34       END_IF;
35
36       2: // Clear receive buffer.
37         SktClearBuf_instance(
38           Execute:=TRUE,
39           Socket :=WkSocket); // Socket
40
41       IF (SktClearBuf_instance.Done=TRUE) THEN
42         Stage:=INT#3; // Normal end
43       ELSIF (SktClearBuf_instance.Error=TRUE) THEN
44         Stage:=INT#20; // Error end
45       END_IF;
46
47       3: // Request reading status.
48         SktGetTCPStatus_instance(
49           Execute:=TRUE,
50           Socket :=WkSocket); // Socket
51
52       IF (SktGetTCPStatus_instance.Done=TRUE) THEN
53         Stage:=INT#4; // Normal end
54       ELSIF (SktGetTCPStatus_instance.Error=TRUE) THEN
55         Stage:=INT#30; // Error end
56       END_IF;
57
58       4: // Request sending data
59         StringToAry('CONNECT Robot_access',SendSocketDat[0]);
60         SendSocketDat[20]:=16#0D;
61         SendSocketDat[21]:=16#0A;

```

```
68 SktTCPSend_instance(
69   Execute:=TRUE,
70   Socket :=WkSocket; // Socket
71   SendDat:=SendSocketDat[0], // Send data
72   Size :=22); // Send data size
73
74 IF (SktTCPSend_instance.Done=TRUE) THEN
75   Stage:=INT#5; // Normal end
76   SktTCPSend_instance(
77     Execute:=FALSE,
78     SendDat:=SendSocketDat[0]); // Dummy
79 ELSIF (SktTCPSend_instance.Error=TRUE) THEN
80   Stage:=INT#40; // Error end
81 END_IF;
82
83 5: // Request receiving data
84   SktTCPRecv_instance(
85     Execute:=TRUE,
86     Socket :=WkSocket; // Socket
87     TimeOut:=UINT#1000, // Timeout time
88     Size :=UINT#250, // Receive data size
89     RcvDat :=RcvSocketDat[0]); // Receive data
90
91 IF (SktTCPRecv_instance.Done=TRUE) THEN
92   Stage:=INT#6; // Normal end
93   SktTCPRecv_instance( // Initialize instance.
94     Execute:=FALSE,
95     RcvDat :=RcvSocketDat[0]); // Dummy
96 ELSIF (SktTCPRecv_instance.Error=TRUE) THEN
97   Stage:=INT#50; // Error end
98 END_IF;
99
100 6: // Request sending data
101   StringToAry('HOSTCTRL_REQUEST RSTATS 0',SendSocketDat[0]);
102   SendSocketDat[25]:=16#0D://CR
103   SendSocketDat[26]:=16#0A://LF
104   SktTCPSend_instance(
105     Execute:=TRUE,
106     Socket :=WkSocket; // Socket
107     SendDat:=SendSocketDat[0], // Send data
108     Size :=27); // Send data size
109   test_size:=SizeOfAry(SendSocketDat);
110
111 IF (SktTCPSend_instance.Done=TRUE) THEN
112   Stage:=INT#7; // Normal end
113   SktTCPSend_instance(
114     Execute:=FALSE,
115     SendDat:=SendSocketDat[0]); // Dummy
116 ELSIF (SktTCPSend_instance.Error=TRUE) THEN
117   Stage:=INT#60; // Error end
118 END_IF;
119
120 7: // Request receiving data
121   SktTCPRecv_instance(
122     Execute:=TRUE,
123     Socket :=WkSocket; // Socket
124     TimeOut:=UINT#1000, // Timeout time
125     Size :=UINT#250, // Receive data size
126     RcvDat :=RcvSocketDat[0]); // Receive data
127
128 IF (SktTCPRecv_instance.Done=TRUE) THEN
129   Stage:=INT#8; // Normal end
```

```
123: SktTCPRecv_instance(          // Initialize instance.  
124:   Execute:=FALSE;  
125:   RcvDat :=RcvSocketDat[0]);    // Dummy  
126: ELSIF (SktTCPRecv_instance.Error=TRUE) THEN  
127:   Stage:=INT#70;               // Error end  
128: END_IF;  
129:  
130: 8 : // Request receiving data  
131: SktTCPRecv_instance(  
132:   Execute:=TRUE,  
133:   Socket :=WkSocket,        // Socket  
134:   TimeOut:=UINT#1000,       // Timeout time  
135:   Size  :=UINT#250,         // Receive data size  
136:   RcvDat :=RcvSocketDat[0]); // Receive data  
137:  
138: IF (SktTCPRecv_instance.Done=TRUE) THEN  
139:   Stage:=INT#15;           // Normal end  
140:   SktTCPRecv_instance(      // Initialize instance.  
141:     Execute:=FALSE;  
142:     RcvDat :=RcvSocketDat[0]); // Dummy  
143: ELSIF (SktTCPRecv_instance.Error=TRUE) THEN  
144:   Stage:=INT#70;           // Error end  
145: END_IF;  
146:  
147:  
148: 15 : // Request closing.  
149: SktClose_instance(  
150:   Execute:=TRUE,  
151:   Socket :=WkSocket);     // Socket  
152: IF (ColmToLine_BYTE(RcvSocketDat[0],1,3)=0) THEN  
153:   Robot_ready:=1;  
154: ELSE  
155:   Robot_ready:=0;  
156: END_IF;  
157: IF (SktClose_instance.Done=TRUE) THEN  
158:   Stage:=INT#0;            // Normal end  
159: ELSIF (SktClose_instance.Error=TRUE) THEN  
160:   Stage:=INT#150;          // Error end  
161: END_IF;  
162:  
163: 0 : // Normal end  
164: CHECK_STATUS_RUN :=FALSE;  
165: trigger :=FALSE;  
166: IF (AryToString(RcvSocketDat[0],2)='OK') THEN  
167:   Robot_ready:=1;  
168: END_IF;  
169:  
170: ELSE // Interrupted by error.  
171:   CHECK_STATUS_RUN :=FALSE;  
172:   trigger :=FALSE;  
173: END_CASE;  
174: END_IF;  
175: END_IF;  
176:  
177: IF not SktTCPSend_instance.Execute THEN  
178:   FOR SendAry:=int#0 TO int#250 BY INT#1 do  
179:     SendSocketDat[SendAry]:=16#00;  
180:   END_FOR;  
181: END_IF;  
182:
```

1-8-1-1-5.CHNG_B1

```
0      1 // Start sequence when Trigger changes to TRUE.
1 IF ( (CHNG_B1=TRUE) AND (CHNG_B1_RUN=FALSE) AND
2     (EIP1_EtnOnlineSta=TRUE) ) THEN
3     CHNG_B1_RUN :=TRUE;
4     Stage :=INT#1;
5     SktTCPConnect_instance(Execute:=FALSE); // Initialize instance.
6     SktClearBuf_instance(Execute:=FALSE); // Initialize instance.
7     SktGetTCPStatus_instance(Execute:=FALSE); // Initialize instance.
8     SktTCPSend_instance(
9       Execute:=FALSE,
10      SendDat:=SendSocketDat[0]); // Dummy
11     SktTCPRecv_instance(           // Initialize instance.
12       Execute:=FALSE,
13       RcvDat :=RcvSocketDat[0]); // Dummy
14     SktClose_instance(Execute:=FALSE); // Initialize instance.
15 END_IF;
16
17 IF Tick then
18   IF (CHNG_B1_RUN=TRUE) THEN
19     CASE Stage OF
20       1: // Request a connection.
21         SktTCPConnect_instance(
22           Execute :=TRUE,
23           SrcTcpPort:=UINT#0, // Local TCP port number: Automatically
24           assigned.
25           DstAddr :='192.168.250.20', // Remote IP address
26           DstTcpPort:=UINT#80, // Destination TCP port number
27           Socket =>WkSocket); // Socket
28
29   IF (SktTCPConnect_instance.Done=TRUE) THEN
30     Stage:=INT#2; // Normal end
31   ELSIF (SktTCPConnect_instance.Error=TRUE) THEN
32     Stage:=INT#10; // Error end
33   END_IF;
34
35   2: // Clear receive buffer.
36   SktClearBuf_instance(
37     Execute:=TRUE,
38     Socket :=WkSocket); // Socket
39
40   IF (SktClearBuf_instance.Done=TRUE) THEN
41     Stage:=INT#3; // Normal end
42   ELSIF (SktClearBuf_instance.Error=TRUE) THEN
43     Stage:=INT#20; // Error end
44   END_IF;
45
46   3: // Request reading status.
47   SktGetTCPStatus_instance(
48     Execute:=TRUE,
49     Socket :=WkSocket); // Socket
50
51   IF (SktGetTCPStatus_instance.Done=TRUE) THEN
52     Stage:=INT#4; // Normal end
53   ELSIF (SktGetTCPStatus_instance.Error=TRUE) THEN
54     Stage:=INT#30; // Error end
55   END_IF;
56
57   4: // Request sending data
58   StringToAry('CONNECT Robot_access',SendSocketDat[0]);
59   SendSocketDat[20]:=16#0D;
60   SendSocketDat[21]:=16#0A;
```

```
68 SktTCPSend_instance(
69   Execute:=TRUE,
70   Socket :=WkSocket; // Socket
71   SendDat:=SendSocketDat[0], // Send data
72   Size :=22); // Send data size
73
74 IF (SktTCPSend_instance.Done=TRUE) THEN
75   Stage:=INT#5; // Normal end
76   SktTCPSend_instance(
77     Execute:=FALSE,
78     SendDat:=SendSocketDat[0]); // Dummy
79 ELSIF (SktTCPSend_instance.Error=TRUE) THEN
80   Stage:=INT#40; // Error end
81 END_IF;
82
83 5: // Request receiving data
84   SktTCPRecv_instance(
85     Execute:=TRUE,
86     Socket :=WkSocket; // Socket
87     TimeOut:=UINT#1000, // Timeout time
88     Size :=UINT#250, // Receive data size
89     RcvDat :=RcvSocketDat[0]); // Receive data
90
91 IF (SktTCPRecv_instance.Done=TRUE) THEN
92   Stage:=INT#6; // Normal end
93   SktTCPRecv_instance( // Initialize instance.
94     Execute:=FALSE,
95     RcvDat :=RcvSocketDat[0]); // Dummy
96 ELSIF (SktTCPRecv_instance.Error=TRUE) THEN
97   Stage:=INT#50; // Error end
98 END_IF;
99
100 6: // Request sending data
101   StringToAry('HOSTCTRL_REQUEST LOADV 6',SendSocketDat[0]);
102   SendSocketDat[24]:=16#0D://CR
103   SendSocketDat[25]:=16#0A://LF
104   SktTCPSend_instance(
105     Execute:=TRUE,
106     Socket :=WkSocket; // Socket
107     SendDat:=SendSocketDat[0], // Send data
108     Size :=26); // Send data size
109   test_size:=SizeOfAry(SendSocketDat);
110
111 IF (SktTCPSend_instance.Done=TRUE) THEN
112   Stage:=INT#7; // Normal end
113   SktTCPSend_instance(
114     Execute:=FALSE,
115     SendDat:=SendSocketDat[0]); // Dummy
116 ELSIF (SktTCPSend_instance.Error=TRUE) THEN
117   Stage:=INT#60; // Error end
118 END_IF;
119
120 7: // Request receiving data
121   SktTCPRecv_instance(
122     Execute:=TRUE,
123     Socket :=WkSocket; // Socket
124     TimeOut:=UINT#1000, // Timeout time
125     Size :=UINT#250, // Receive data size
126     RcvDat :=RcvSocketDat[0]); // Receive data
127
128 IF (SktTCPRecv_instance.Done=TRUE) THEN
129   Stage:=INT#8; // Normal end
```

```
123: // Normal end
124: SktTCPRecv_instance(          // Initialize instance.
125:   Execute:=FALSE,
126:   RcvDat :=RcvSocketDat[0]);    // Dummy
127: ELSIF (SktTCPRecv_instance.Error=TRUE) THEN
128:   Stage:=INT#70;              // Error end
129: END_IF;

130: 8: // Request sending data
131:   StringToAry(str_del,SendSocketDat[4]);
132:   StringToAry('0,1',SendSocketDat[0]);
133:   SendSocketDat[5]:=16#0D;//CR
134:   //SendSocketDat[2]:=16#0A;//LF
135:   SktTCPSend_instance(
136:     Execute:=TRUE,
137:     Socket :=WkSocket,        // Socket
138:     SendDat:=SendSocketDat[0], // Send data
139:     Size :=6);             // Send data size
140: IF (SktTCPSend_instance.Done=TRUE) THEN
141:   Stage:=INT#9;              // Normal end
142:   SktTCPSend_instance(
143:     Execute:=FALSE,
144:     SendDat:=SendSocketDat[0]); // Dummy
145: ELSIF (SktTCPSend_instance.Error=TRUE) THEN
146:   Stage:=INT#80;              // Error end
147: END_IF;

148: 9: // Request receiving data
149: SktTCPRecv_instance(
150:   Execute:=TRUE,
151:   Socket :=WkSocket,        // Socket
152:   TimeOut:=UINT#1000,        // Timeout time
153:   Size :=UINT#250,           // Receive data size
154:   RcvDat :=RcvSocketDat[0]); // Receive data
155:
156: IF (SktTCPRecv_instance.Done=TRUE) THEN
157:   Stage:=INT#15;              // Normal end
158:   SktTCPRecv_instance(          // Initialize instance.
159:     Execute:=FALSE,
160:     RcvDat :=RcvSocketDat[0]); // Dummy
161:     B1:=BYTE_BCD_TO_INT(RcvSocketDat[0])-30;
162:   ELSIF (SktTCPRecv_instance.Error=TRUE) THEN
163:     Stage:=INT#90;              // Error end
164:   END_IF;

165: 15: // Request closing.
166:   SktClose_instance(
167:     Execute:=TRUE,
168:     Socket :=WkSocket);      // Socket
169:
170: IF (SktClose_instance.Done=TRUE) THEN
171:   Stage:=INT#0;                // Normal end
172: ELSIF (SktClose_instance.Error=TRUE) THEN
173:   Stage:=INT#150;              // Error end
174: END_IF;

175: 0: // Normal end
176:   CHNG_B1_RUN :=FALSE;
177:   trigger :=FALSE;
178:
179: ELSE // Interrupted by error,
180:   CHNG_B1_RUN :=FALSE;
181:   trigger :=FALSE;
182:
183: END_IF;
```

```
185      END_LASE;  
186  
187      END_IF;  
188      END_IF;  
189  
190  IF not SktTCPSend_instance.Execute THEN  
191    FOR SendAry:=int#0 TO int#250 BY INT#1 do  
192      SendSocketDat[SendAry]:=16#00;  
193    END_FOR;  
194  END_IF;  
195
```

1-8-1-1-6.START_JOB

```

0
1 // Start sequence when Trigger changes to TRUE.
2 IF ( (START_JOB=TRUE) AND (START_JOB_RUN=FALSE) AND
3     (EIP1_EtnOnlineSta=TRUE) ) THEN
4     START_JOB_RUN :=TRUE;
5     Stage :=INT#1;
6     SktTCPConnect_instance(Execute:=FALSE); // Initialize instance.
7     SktClearBuf_instance(Execute:=FALSE); // Initialize instance.
8     SktGetTCPStatus_instance(Execute:=FALSE); // Initialize instance.
9     SktTCPSend_instance(
10        Execute:=FALSE,
11        SendDat:=SendSocketDat[0]); // Dummy
12     SktTCPRecv_instance(           // Initialize instance.
13        Execute:=FALSE,
14        RcvDat :=RcvSocketDat[0]); // Dummy
15     SktClose_instance(Execute:=FALSE); // Initialize instance.
16 END_IF;
17
18 IF Tick then
19   IF (START_JOB_RUN=TRUE) THEN
20     CASE Stage OF
21       1: // Request a connection.
22         SktTCPConnect_instance(
23           Execute :=TRUE,
24           SrcTcpPort:=UINT#0, // Local TCP port number: Automatically
25           assigned.
26           DstAddr :='192.168.250.20', // Remote IP address
27           DstTcpPort:=UINT#80, // Destination TCP port number
28           Socket =>WkSocket); // Socket
29
30       IF (SktTCPConnect_instance.Done=TRUE) THEN
31         Stage:=INT#2; // Normal end
32       ELSIF (SktTCPConnect_instance.Error=TRUE) THEN
33         Stage:=INT#10; // Error end
34       END_IF;
35
36       2: // Clear receive buffer.
37         SktClearBuf_instance(
38           Execute:=TRUE,
39           Socket :=WkSocket); // Socket
40
41       IF (SktClearBuf_instance.Done=TRUE) THEN
42         Stage:=INT#3; // Normal end
43       ELSIF (SktClearBuf_instance.Error=TRUE) THEN
44         Stage:=INT#20; // Error end
45       END_IF;
46
47       3: // Request reading status.
48         SktGetTCPStatus_instance(
49           Execute:=TRUE,
50           Socket :=WkSocket); // Socket
51
52       IF (SktGetTCPStatus_instance.Done=TRUE) THEN
53         Stage:=INT#4; // Normal end
54       ELSIF (SktGetTCPStatus_instance.Error=TRUE) THEN
55         Stage:=INT#30; // Error end
56       END_IF;
57
58       4: // Request sending data
59         StringToAry('CONNECT Robot_access',SendSocketDat[0]);
60         SendSocketDat[20]:=16#0D;
61         SendSocketDat[21]:=16#0A;

```

```
68 SktTCPSend_instance(
69   Execute:=TRUE,
70   Socket :=WkSocket      // Socket
71   SendDat:=SendSocketDat[0], // Send data
72   Size  :=22);    // Send data size
73
74 IF (SktTCPSend_instance.Done=TRUE) THEN
75   Stage:=INT#5;          // Normal end
76   SktTCPSend_instance(
77     Execute:=FALSE,
78     SendDat:=SendSocketDat[0]); // Dummy
79 ELSIF (SktTCPSend_instance.Error=TRUE) THEN
80   Stage:=INT#40;         // Error end
81 END_IF;
82
83 5: // Request receiving data
84   SktTCPRecv_instance(
85     Execute:=TRUE,
86     Socket :=WkSocket      // Socket
87     TimeOut:=UINT#1000,    // Timeout time
88     Size  :=UINT#250,     // Receive data size
89     RcvDat :=RcvSocketDat[0]); // Receive data
90
91 IF (SktTCPRecv_instance.Done=TRUE) THEN
92   Stage:=INT#6;          // Normal end
93   SktTCPRecv_instance(           // Initialize instance.
94     Execute:=FALSE,
95     RcvDat :=RcvSocketDat[0]); // Dummy
96 ELSIF (SktTCPRecv_instance.Error=TRUE) THEN
97   Stage:=INT#50;         // Error end
98 END_IF;
99
100 6: // Request sending data
101   StringToAry('HOSTCTRL_REQUEST_START 6',SendSocketDat[0]);
102   SendSocketDat[24]:=16#0D://CR
103   SendSocketDat[25]:=16#0A://LF
104   SktTCPSend_instance(
105     Execute:=TRUE,
106     Socket :=WkSocket      // Socket
107     SendDat:=SendSocketDat[0], // Send data
108     Size  :=26);    // Send data size
109   test_size:=SizeOfAry(SendSocketDat);
110
111 IF (SktTCPSend_instance.Done=TRUE) THEN
112   Stage:=INT#7;          // Normal end
113   SktTCPSend_instance(
114     Execute:=FALSE,
115     SendDat:=SendSocketDat[0]); // Dummy
116 ELSIF (SktTCPSend_instance.Error=TRUE) THEN
117   Stage:=INT#60;         // Error end
118 END_IF;
119
120 7: // Request receiving data
121   SktTCPRecv_instance(
122     Execute:=TRUE,
123     Socket :=WkSocket      // Socket
124     TimeOut:=UINT#1000,    // Timeout time
125     Size  :=UINT#250,     // Receive data size
126     RcvDat :=RcvSocketDat[0]); // Receive data
127
128 IF (SktTCPRecv_instance.Done=TRUE) THEN
129   Stage:=INT#8.          // Normal end
```

```
123: SktTCPRecv_instance(          // Initialize instance.  
124: Execute:=FALSE;  
125: RcvDat :=RcvSocketDat[0];    // Dummy  
126: ELSIF (SktTCPRecv_instance.Error=TRUE) THEN  
127: Stage:=INT#70;              // Error end  
128: END_IF;  
129:  
130: 8: // Request sending data  
131: StringToAry('VMAIN',SendSocketDat[0]);  
132: SendSocketDat[5]:=16#0D; //CR  
133: //SendSocketDat[2]:=16#0A; //LF  
134: SktTCPSend_instance(  
135: Execute:=TRUE,  
136: Socket :=WkSocket;        // Socket  
137: SendDat:=SendSocketDat[0], // Send data  
138: Size :=6);             // Send data size  
139: IF (SktTCPSend_instance.Done=TRUE) THEN  
140: Stage:=INT#9;              // Normal end  
141: SktTCPSend_instance(  
142: Execute:=FALSE,  
143: SendDat:=SendSocketDat[0]); // Dummy  
144: ELSIF (SktTCPSend_instance.Error=TRUE) THEN  
145: Stage:=INT#80;              // Error end  
146: END_IF;  
147:  
148: 9: // Request receiving data  
149: SktTCPRecv_instance(  
150: Execute:=TRUE,  
151: Socket :=WkSocket;        // Socket  
152: TimeOut:=UINT#1000,       // Timeout time  
153: Size :=UINT#250,          // Receive data size  
154: RcvDat :=RcvSocketDat[0]; // Receive data  
155:  
156: IF (SktTCPRecv_instance.Done=TRUE) THEN  
157: Stage:=INT#15;              // Normal end  
158: SktTCPRecv_instance(          // Initialize instance.  
159: Execute:=FALSE,  
160: RcvDat :=RcvSocketDat[0]); // Dummy  
161: B1:=BYTE_BCD_TO_INT(RcvSocketDat[0])-30;  
162: ELSIF (SktTCPRecv_instance.Error=TRUE) THEN  
163: Stage:=INT#90;              // Error end  
164: END_IF;  
165:  
166: 15: // Request closing.  
167: SktClose_instance(  
168: Execute:=TRUE,  
169: Socket :=WkSocket);        // Socket  
170:  
171: IF (SktClose_instance.Done=TRUE) THEN  
172: Stage:=INT#0;              // Normal end  
173: ELSIF (SktClose_instance.Error=TRUE) THEN  
174: Stage:=INT#150;             // Error end  
175: END_IF;  
176:  
177: 0: // Normal end  
178: START_JOB_RUN :=FALSE;  
179: trigger:=FALSE;  
180: ELSE // Interrupted by error.  
181: START_JOB_RUN :=FALSE;  
182: trigger:=FALSE;  
183: END_CASE;  
184:
```

```
185 |     END_IF;
186 | END_IF;
187 |
188 | IF not SktTCPSend_instance.Execute THEN
189 |   FOR SendAry:=int#0 TO int#250 BY INT#1 do
190 |     SendSocketDat[SendAry]:=16#00;
191 |   END_FOR;
192 | END_IF;
193 |
```

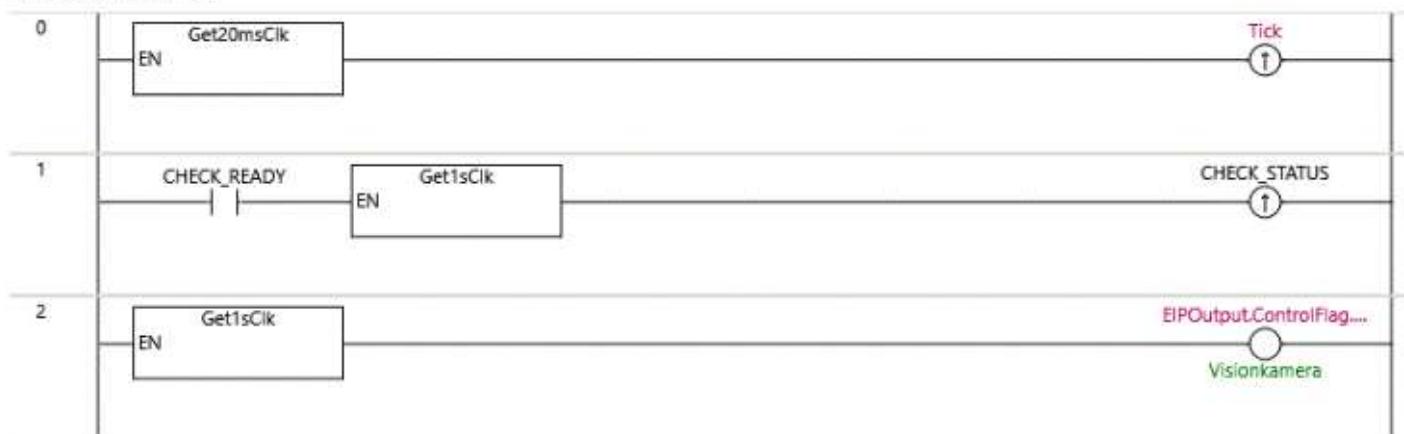
1-8-1-1-7.CHNG_P0

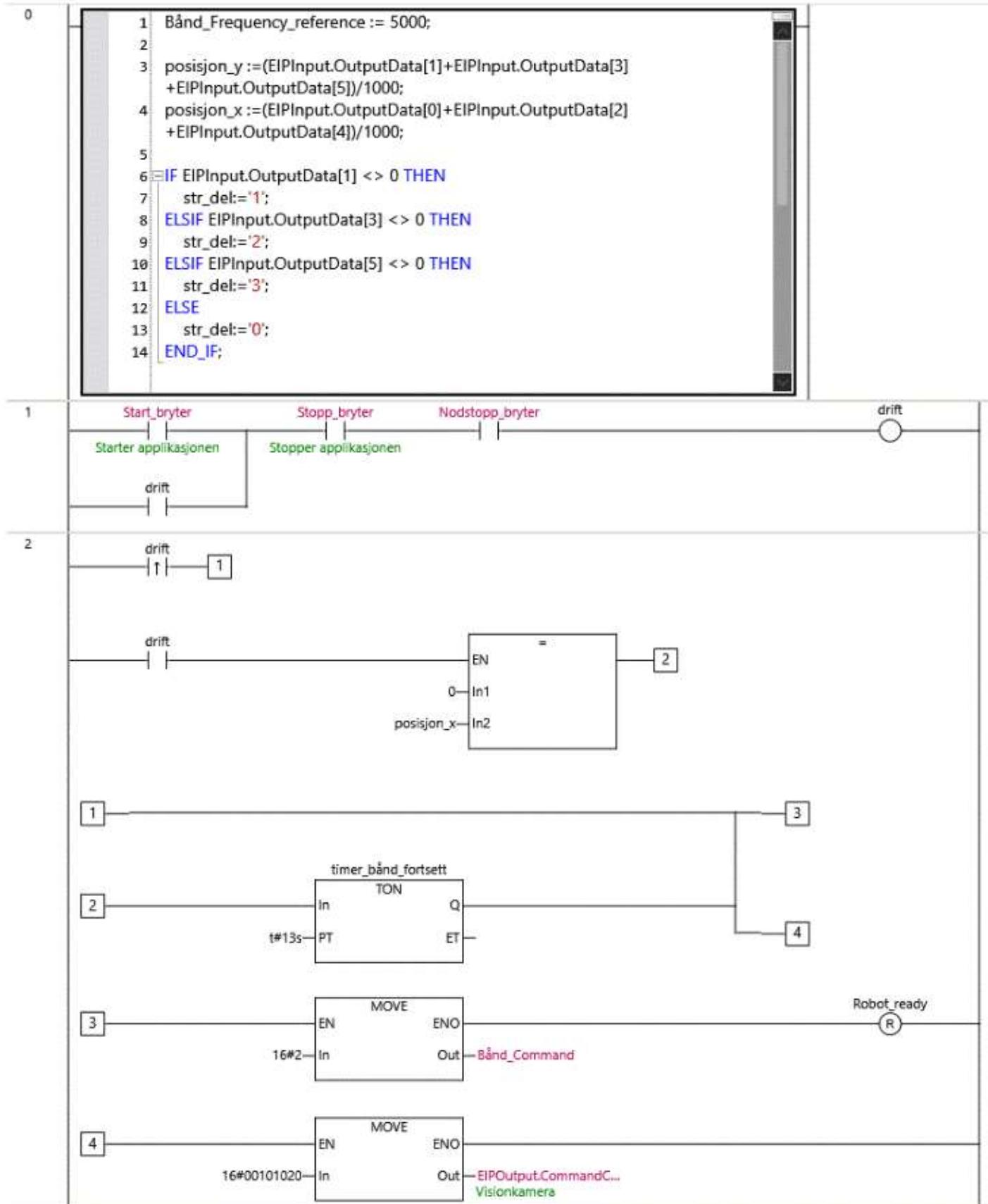
```
0      1 // Start sequence when Trigger changes to TRUE.
1 IF ( (CHNG_P0=TRUE) AND (CHNG_P0_RUN=FALSE) AND
2     (EIP1_EtnOnlineSta=TRUE) ) THEN
3     CHNG_P0_RUN :=TRUE;
4     Stage :=INT#1;
5     SktTCPConnect_instance(Execute:=FALSE); // Initialize instance.
6     SktClearBuf_instance(Execute:=FALSE); // Initialize instance.
7     SktGetTCPStatus_instance(Execute:=FALSE); // Initialize instance.
8     SktTCPSend_instance(
9       Execute:=FALSE,
10      SendDat:=SendSocketDat[0]); // Dummy
11     SktTCPRecv_instance(           // Initialize instance.
12       Execute:=FALSE,
13       RcvDat :=RcvSocketDat[0]); // Dummy
14     SktClose_instance(Execute:=FALSE); // Initialize instance.
15 END_IF;
16
17 IF Tick then
18   IF (CHNG_P0_RUN=TRUE) THEN
19     CASE Stage OF
20       1: // Request a connection.
21         SktTCPConnect_instance(
22           Execute :=TRUE,
23           SrcTcpPort:=UINT#0, // Local TCP port number: Automatically
24           assigned.
25           DstAddr :='192.168.250.20', // Remote IP address
26           DstTcpPort:=UINT#80, // Destination TCP port number
27           Socket =>WkSocket); // Socket
28
29   IF (SktTCPConnect_instance.Done=TRUE) THEN
30     Stage:=INT#2; // Normal end
31   ELSIF (SktTCPConnect_instance.Error=TRUE) THEN
32     Stage:=INT#10; // Error end
33   END_IF;
34
35   2: // Clear receive buffer.
36   SktClearBuf_instance(
37     Execute:=TRUE,
38     Socket :=WkSocket); // Socket
39
40   IF (SktClearBuf_instance.Done=TRUE) THEN
41     Stage:=INT#3; // Normal end
42   ELSIF (SktClearBuf_instance.Error=TRUE) THEN
43     Stage:=INT#20; // Error end
44   END_IF;
45
46   3: // Request reading status.
47   SktGetTCPStatus_instance(
48     Execute:=TRUE,
49     Socket :=WkSocket); // Socket
50
51   IF (SktGetTCPStatus_instance.Done=TRUE) THEN
52     Stage:=INT#4; // Normal end
53   ELSIF (SktGetTCPStatus_instance.Error=TRUE) THEN
54     Stage:=INT#30; // Error end
55   END_IF;
56
57   4: // Request sending data
58   StringToAry('CONNECT Robot_access',SendSocketDat[0]);
59   SendSocketDat[20]:=16#0D;
60   SendSocketDat[21]:=16#0A;
```

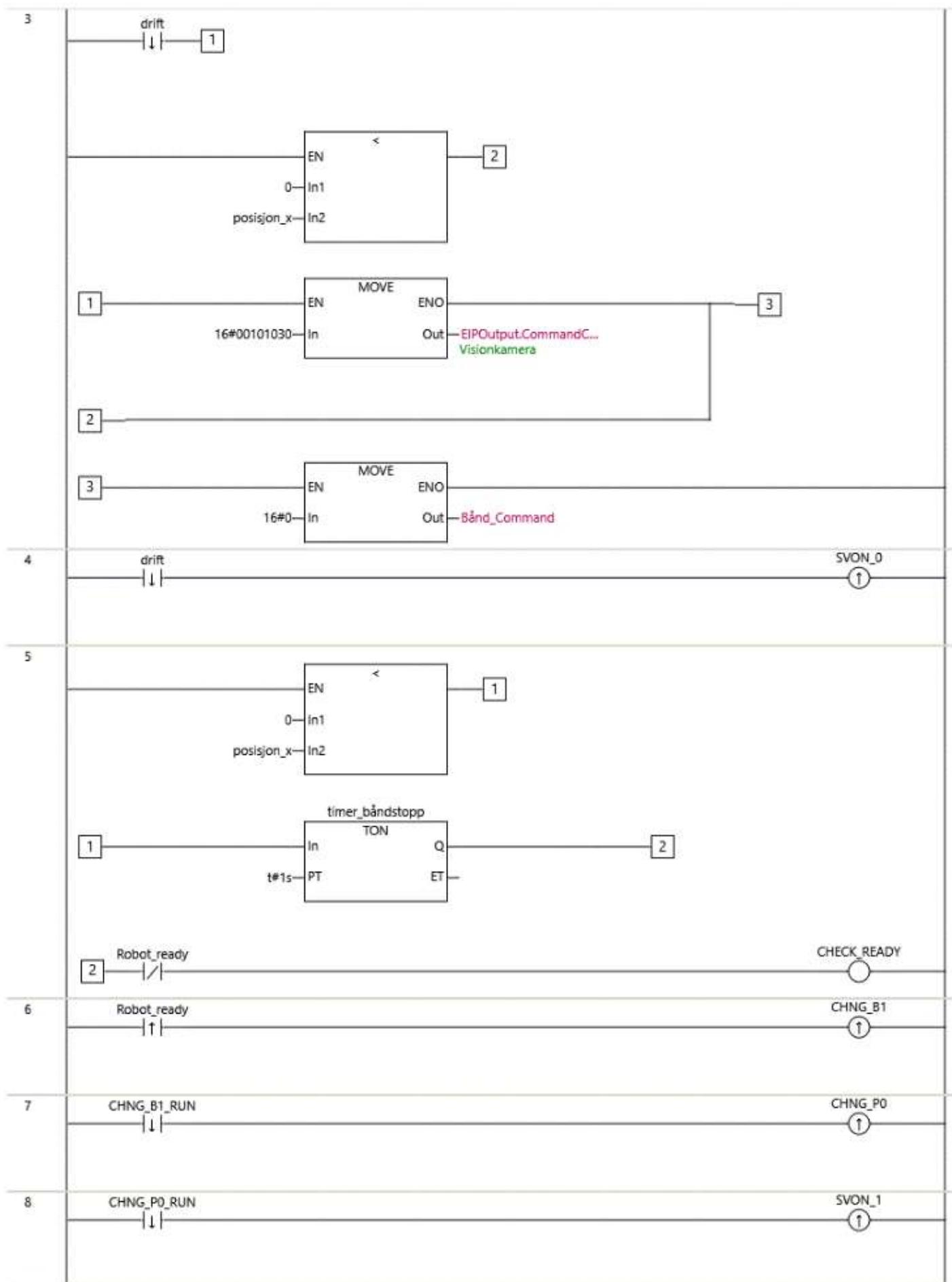
```
68 SktTCPSend_instance(
69   Execute:=TRUE,
70   Socket :=WkSocket      // Socket
71   SendDat:=SendSocketDat[0], // Send data
72   Size  :=22);    // Send data size
73
74 IF (SktTCPSend_instance.Done=TRUE) THEN
75   Stage:=INT#5;          // Normal end
76   SktTCPSend_instance(
77     Execute:=FALSE,
78     SendDat:=SendSocketDat[0]); // Dummy
79 ELSIF (SktTCPSend_instance.Error=TRUE) THEN
80   Stage:=INT#40;         // Error end
81 END_IF;
82
83 5: // Request receiving data
84   SktTCPRecv_instance(
85     Execute:=TRUE,
86     Socket :=WkSocket      // Socket
87     TimeOut:=UINT#1000,    // Timeout time
88     Size  :=UINT#250,     // Receive data size
89     RcvDat :=RcvSocketDat[0]); // Receive data
90
91 IF (SktTCPRecv_instance.Done=TRUE) THEN
92   Stage:=INT#6;          // Normal end
93   SktTCPRecv_instance(           // Initialize instance.
94     Execute:=FALSE,
95     RcvDat :=RcvSocketDat[0]); // Dummy
96 ELSIF (SktTCPRecv_instance.Error=TRUE) THEN
97   Stage:=INT#50;         // Error end
98 END_IF;
99
100 6: // Request sending data
101   StringToAry('HOSTCTRL_REQUEST LOADV 31',SendSocketDat[0]);
102   SendSocketDat[25]:=16#0D://CR
103   SendSocketDat[26]:=16#0A://LF
104   SktTCPSend_instance(
105     Execute:=TRUE,
106     Socket :=WkSocket      // Socket
107     SendDat:=SendSocketDat[0], // Send data
108     Size  :=27);    // Send data size
109   test_size:=SizeOfAry(SendSocketDat);
110
111 IF (SktTCPSend_instance.Done=TRUE) THEN
112   Stage:=INT#7;          // Normal end
113   SktTCPSend_instance(
114     Execute:=FALSE,
115     SendDat:=SendSocketDat[0]); // Dummy
116 ELSIF (SktTCPSend_instance.Error=TRUE) THEN
117   Stage:=INT#60;         // Error end
118 END_IF;
119
120 7: // Request receiving data
121   SktTCPRecv_instance(
122     Execute:=TRUE,
123     Socket :=WkSocket      // Socket
124     TimeOut:=UINT#1000,    // Timeout time
125     Size  :=UINT#250,     // Receive data size
126     RcvDat :=RcvSocketDat[0]); // Receive data
127
128 IF (SktTCPRecv_instance.Done=TRUE) THEN
129   Stage:=INT#8.          // Normal end
```

```
123 SktTCPRecv_instance( // Initialize instance.  
124 Execute:=FALSE.  
125 RcvDat :=RcvSocketDat[0]; // Dummy  
126 ELSIF (SktTCPRecv_instance.Error=TRUE) THEN  
127 Stage:=INT#70; // Error end  
128 END_IF;  
129  
130 8: // Request sending data  
131  
132 StringToAry('4,0,1,2',SendSocketDat[0]);  
133 StringToAry(numtodecstring(posisjon_x,3,_ZERO),SendSocketDat[8]);  
134 StringToAry('.',SendSocketDat[11]);  
135 StringToAry(numtodecstring(posisjon_y,3,_ZERO),SendSocketDat[12]);  
136 StringToAry('.0,180,0,90,0,0',SendSocketDat[15]);  
137  
138 SendSocketDat[30]:=16#0D; //CR  
139 //SendSocketDat[2]:=16#0A; //LF  
140 SktTCPSend_instance(  
141 Execute:=TRUE,  
142 Socket :=WkSocket, // Socket  
143 SendDat:=SendSocketDat[0], // Send data  
144 Size :=31); // Send data size  
145 IF (SktTCPSend_instance.Done=TRUE) THEN  
146 Stage:=INT#9; // Normal end  
147 SktTCPSend_instance(  
148 Execute:=FALSE,  
149 SendDat:=SendSocketDat[0]); // Dummy  
150 ELSIF (SktTCPSend_instance.Error=TRUE) THEN  
151 Stage:=INT#80; // Error end  
152 END_IF;  
153  
154 9: // Request receiving data  
155 SktTCPRecv_instance(  
156 Execute:=TRUE,  
157 Socket :=WkSocket, // Socket  
158 TimeOut:=UINT#1000, // Timeout time  
159 Size :=UINT#250, // Receive data size  
160 RcvDat :=RcvSocketDat[0]); // Receive data  
161  
162 IF (SktTCPRecv_instance.Done=TRUE) THEN  
163 Stage:=INT#15; // Normal end  
164 SktTCPRecv_instance( // Initialize instance.  
165 Execute:=FALSE,  
166 RcvDat :=RcvSocketDat[0]); // Dummy  
167 B1:=BYTE_BCD_TO_INT(RcvSocketDat[0])-30;  
168 ELSIF (SktTCPRecv_instance.Error=TRUE) THEN  
169 Stage:=INT#90; // Error end  
170 END_IF;  
171  
172 15: // Request closing.  
173 SktClose_instance(  
174 Execute:=TRUE,  
175 Socket :=WkSocket); // Socket  
176  
177 IF (SktClose_instance.Done=TRUE) THEN  
178 Stage:=INT#0; // Normal end  
179 ELSIF (SktClose_instance.Error=TRUE) THEN  
180 Stage:=INT#150; // Error end  
181 END_IF;  
182  
183 0: // Normal end  
184 CHNG_P0_RUN :=FALSE;
```

```
185 trigger :=FALSE;
186
187 ELSE // Interrupted by error.
188 CHNG_P0_RUN :=FALSE;
189 trigger :=FALSE;
190 END_CASE;
191
192 END_IF;
193 END_IF;
194
195 IF not SktTCPSend_instance.Execute THEN
196 FOR SendAry:=int#0 TO int#250 BY INT#1 do
197   SendSocketDat[SendAry]:=16#00;
198 END_FOR;
199 END_IF;
200
```

1-8-1-1-8.TICKS

1-8-1-1-9.Bånd





Beregninger

Motor

Tilført effekt

$$P_1 = \frac{P_2}{\eta}$$

$$P_1 = \frac{0,25kW}{0,72} = 0,34kW$$

Effekttap

$$\Delta P = P_1 - P_2$$

$$\Delta P = 0,34kW - 0,25kW = 0,09kW$$

Strømtrekk i motor

$$I = \frac{P_2}{U * \sqrt{3} * \cos \varphi * \eta}$$

$$I = \frac{250W}{230V * \sqrt{3} * 0,75 * 0,72} = 1,16A$$

Resistans i kabel til motor

$$R = \frac{\rho * l}{A}$$

$$R = \frac{0,0175 * 10m}{1,5} = 0,11\Omega$$

Spenningsfall i motor

$$\Delta U = \frac{\rho * \sqrt{3} * l * I * \cos \varphi}{A}$$

$$\Delta U = \frac{0,0175 * \sqrt{3} * 10m * 1,16A * \cos^2 75}{1,5} = 0,17 V$$

Motorfabrikanter anbefaler maksimum 5% spenningsfall for kabler til motorer.

$$230V - 0,17V = 229,83V$$

$$229,83V / 230V = 0,99 = 0,01\%$$

Sluttkontroll

Alle elektriske anlegg skal kontrolleres før de settes i drift, dette er påbudt i forskrifter og normer.

Gruppen skal måle motorstrøm, motorens isolasjonstilstand og jordlederens kontinuitet.

Kontinuitetsmåling

Å måle jordlederens kontinuitet betyr at man måler om det er god forbindelse hele veien fra fordelingen og ut til motoren.

Måling fra jordingspunkt i skap til motor=0,1 Ω

Motorens isolasjonstilstand

Motorens isolasjonstilstand forteller oss hvor god isolasjon hver enkelt fase har mot jord. Dårlig isolasjon fra fase til jord kan føre til farlige situasjoner og gi skader på både mennesker og utstyr. Krav til målinger er slik:

Anlegg inntil 500V skal prøves med 500V DC, og måleresultat skal minst være 1000 Ω =1M Ω

Måling over vikling:

W2=uendelig(målingen er tett)

U2= uendelig(målingen er tett)

V2= uendelig(målingen er tett)

Omron

Power og switch++							
Produktnr	Typebetegnelse	Produktbeskrivelse	EL nummer	Nettopris	Rabatt	Antall	Totalt
669508	S8VK-S06024	Book type power supply, 60 W, 24VDC, 2.5A, DIN rail mounting, Push-in terminal, Coated		275,00		2 kr	550,00
353025	WES SDI-550	i-Line 5-port Unmanaged Switch - 5 x 10/100BaseT		575		1 kr	575,00
374581	XSW6-6LSZH8SS100CM-Y	Ethernet Cat.6 patch cable, LSZH double shielding (Yellow), 1 m		30		1 kr	30,00
374603	XSW6-6LSZH8SS100CM-B	Ethernet Cat.6 patch cable, LSZH double shielding (Blue), 1 m		30		1 kr	30,00

MX2 Frekvensomformer

Produktnr	Typebetegnelse	Produktbeskrivelse	EL nummer	Listepris	Rabatt	Antall	Totalt
379091	3G3MX2-AB004-E CHN	MX2 Frekvensomformer, 0,4/0,55kW(HD/ND), 3,0/3,5A(HD/ND), 230VAC, 1-fas forsyning	41 208 26	2 528,00	50 %	1 kr	1 264,00
386171	AX-FIM1010-SE-LL	MX2 RF1 filter, 10A, 230VAC, 1-fas, for 0,12 til 0,55kW modeller		220,00	50 %	1 kr	110,00
416241	AX-GPM01-ICE	MX2 jordingsbeslag: For ..AB001-AB004 og ..A2001-A2007		125,00	50 %	1 kr	62,50
343139	3G3AX-MX2-ECT	MX2 EtherCAT option card		2 140,00	50 %	1 kr	1 070,00

Sikkerhetsrelé

Produktnr	Typebetegnelse	Produktbeskrivelse	EL nummer	Listepris	Rabatt	Antall	Totalt
409004	G9SE-201 DC24	Safety relay unit, 24VDC, 2 safety outputs 5A max, aux. output		1 013,00	50 %	1 kr	506,50

NX1P kompakt PLS

Produktnr	Typebetegnelse	Produktbeskrivelse	EL nummer	Listepris	Rabatt	Antall	Totalt
672504	NX1P2-1040DT1	Sysmac NX1P CPU with 40 Digital Transistor I/O (PNP), 1,5 MB memory, EtherCAT (2 servo axes, 4 PTB axes, 16 EtherCAT nodes), EtherNet/IP and 2 serial option ports		10 340,00	45 %	1 kr	5 687,00
380140	NX-PF0730	NX I/O power feed unit, 5-24 V DC input, 8 terminals, 10A, screwless push-in connector, 12mm wide		279,00	45 %	1 kr	153,45
375645	NX-EC0122	1 x Incremental Encoder PNP Input, Phases A and B: Single-phase 500 kHz (phase difference pulse Input x4: 125 kHz), Phase Z: 125 kHz, screwless push-in connector, 12 mm wide		1 807,00	45 %	1 kr	993,85
672505	NX1W-CIF01	Sysmac NX1P RS-232C screwless (15 m max.) serial communication option board		195,00	45 %	1 kr	107,25

FQ2-S3 vision sensor, avansert modell med høyere opplesning, 0,8Mpxler/1,3Mpxler

Produktnr	Typebetegnelse	Produktbeskrivelse	EL nummer	Listepris	Rabatt	Antall	Totalt
372122	FQ2-S35100F-08	FQ2-S35 vision sensor,avansert funksjon, 32 inspeksjon, 32 scener, lang avstand 970 mm, synsfelt 240mm, PNP, Ethernet og I/O innebygd		22 650,00	45 %	1 kr	12 457,50
351984	FQ-WD003-E	FQ I/O kabel, 3m		1 070,00	50 %	1 kr	535,00
360745	FQ-WN003-E	FQ Ethernet kabel, rett M12-RJ45, 3m		596,00	50 %	1 kr	298,00

Sysmac Studio Lite

Produktnr	Typebetegnelse	Produktbeskrivelse	EL nummer	Listepris	Rabatt	Antall	Totalt
672883	SYSMAC-LE201L	Sysmac Studio Lite license only, single user (requires SYSMAC-SE200D download or installation DVD, available separately)		10 340,00	50 %	1 kr	5 170,00
355294	SYSMAC-SE200D	Sysmac Studio installation DVD only, for Windows XP or Windows 7/8, 32 or 64-bit versions (requires SYSMAC-SE2**L license)		309,10	50 %	1 kr	154,55
						Sum	kr 29 754,60

ITO

Transportbånd

Produktnr	Typebetegnelse	Produktbeskrivelse	EL nummer	Nettopris	Rabatt	Antall	Totalt
	ITO båndtransportør			kr 20 000,00		1 kr	20 000,00
	Eksport og emballasje			kr 400,00		1 kr	400,00
	25% MVA					kr 5 100,00	
						Sum	kr 25 500,00

ELFA

Utstyr fra ELFA

Produktnr	Typebetegnelse	Produktbeskrivelse	EL nummer	Nettopris	Rabatt	Antall	Totalt
	Metallinnkapsling 380 x 300 x 210 mm Stålplate IP 66, AE1031.500, Rittal		150-03-058	kr 909,00		1 kr	909,00
	Kontrollkabel 3 x 1,50 mm ² skjermet grå, OLFLEX CLASSIC 115 CY 3G1,5, Lapp		155-81-686	kr 25,00		10 kr	250,00
	Kabelmuffe Messing, formiklet M16 x 1,5 - 53112620, SKINTOP MS-SC-M 16X1,5, Lapp		155-19-171	kr 72,70		1 kr	72,70
	Låsemutter M16 x 1,5 - 52103310, SKINDICHT SM-PE-M 16X1,5, COUNT. NUT, Lapp		155-19-192	kr 12,00		1 kr	12,00
	Automatskring 6 A 2-pole C, A9F07206, Schneider Electric		110-82-968	kr 589,00		1 kr	589,00
	Kambryster 3 kW 16 A Bryterposisjoner 2 Poles 1, PR 12-1101 A4 E Q48PN, Baco		135-11-954	kr 146,00		1 kr	146,00
	Kabelkanal 2000 x 40 x 40 mm, T1-E 40X40 G, Iboco		136-49-241	kr 110,00		1 kr	110,00
	Rekkelempre Beige 0,5...4 mm ² , 1023700000, WDU 2,5N, Weidmüller		148-28-307	kr 10,70		40 kr	428,00
	Jordlekemme gul/grønn 0,5...4 mm ² , 1016200000, WPE 2,5N, Weidmüller		148-28-315	kr 36,50		10 kr	365,00
	Endebremmet 56 x 8 x 47 mm Beige, 1061200000, WEW 35/2, Weidmüller		148-28-737	kr 16,70		8 kr	133,60
	Markeringspenn Symbol +, 5 strimler à 10 stk. 5 x 5 mm hvit, DEK 5 GW +, Weidmüller		300-77-027	kr 24,10		1 kr	24,10
	Markeringspenn Symbol -, 5 strimler à 10 stk. 5 x 5 mm hvit, DEK 5 GW -, Weidmüller		300-77-028	kr 24,10		1 kr	24,10
	Markeringspenn Symbol GW L1, 5 strimler à 10 stk. 5 x 5 mm hvit, DEK 5 GW L1, Weidmüller		300-77-031	kr 24,10		1 kr	24,10
	Markeringspenn Symbol GW L2, 5 strimler à 10 stk. 5 x 5 mm hvit, DEK 5 GW L2, Weidmüller		300-77-032	kr 24,10		1 kr	24,10
	Markeringspenn Symbol Jord, 5 strimler à 10 stk. 5 x 5 mm hvit, DEK 5 GW ERDE, Weidmüller		300-77-029	kr 24,10		1 kr	24,10
	Markeringspenn Tall 1-50 Strips 5 x 6 mm hvit, DEK 6 FW 1-50, Weidmüller		300-77-035	kr 24,10		2 kr	48,20
	Krysskontakt WQV 18 x 7,55 x 26,8 mm gult h, 1054960000, WQV 10/3, Weidmüller		300-11-375	kr 19,90		5 kr	99,50
	Krysskontakt WQV 18 x 7,55 x 16,9 mm gult h, 1052560000, WQV 10/2, Weidmüller		300-11-374	kr 11,40		5 kr	57,00
	D-Sub connector set 9P, SUBCON 9/F-SH, Phoenix Contact		144-05-058	kr 310,00		1 kr	310,00
	Data cable unshielded 12 x0,22 mm ² Copper strand tin-plated grey, PFK 12X0,22 MM ² , CablesSwiss		155-69-355	kr 18,70		15 kr	280,50
	Endehylse Rød 1,5 mm ² /8 mm, 460408.00050, Vogt		148-23-507	kr 13,20		2 kr	26,40
	Endehylse Blå 0,75 mm ² /8 mm, 470208.00050, Vogt		148-23-503	kr 13,20		2 kr	26,40
	Flex, 0,75 mm ² , Rød Fortinnet kobber PVC, LYIV 0,75 MM ² RED, Kabeltronik		155-79-887	kr 274,00		1 kr	274,00
	Flex, 0,75 mm ² , Blå Fortinnet kobber PVC, LYIV 0,75 MM ² BLUE, Kabeltronik		155-79-890	kr 274,00		1 kr	274,00
	MVA 25%						kr 1 132,95
						Sum	kr 5 664,75

Biltema

Utstyr fra Biltema

Produktnr	Typebetegnelse	Produktbeskrivelse	EL nummer	Nettopris	Rabatt	Antall	Totalt
	Transporthjul 100-M12		41178	kr 69,90		4 kr	279,60
	Monteringskruer med borespiss		87691	kr 49,90		1 kr	49,90
	Monteringsplate for bunt bånd		35495	kr 19,90		1 kr	19,90
	Rustfrie skiver ø13x24 25stk		19544	kr 69,90		1 kr	69,90
	Låsmuttere 10stk		19915	kr 34,90		1 kr	34,90
	Øyekabelsko 25 stk		44027	kr 29,90		1 kr	29,90
	Spennbånd 2m x25mm		34272	kr 19,90		1 kr	19,90
	Åpningsbare buntbånd 200 20mm		61651	kr 49,90		1 kr	49,90
						Sum	kr 553,90
						Sum total	kr 61 473,25

		Name	Duration	Start
1		Prosjektoppstart	3 days	23.08.16 08:00
2		Prosjektoppstart og planlegging	3 days	30.08.16 12:00
3		Tegne el-skjema	3 days?	06.09.16 08:00
4		Tegne pneumatikkskjema	3 days?	06.09.16 08:00
5		Forslag til ny PLS og sensor	3 days?	06.09.16 08:00
6		Bestille deler	3 days	06.09.16 12:00
7		Motatt deler	3 days?	13.09.16 12:00
8		Montasje av nye deler	3 days?	13.09.16 08:00
9		Lage PLS program	3 days?	13.09.16 08:00
10		Testing av nytt program og igangkjøring av maskin	3 days?	20.09.16 12:00
11		Komponentliste	1 day?	22.08.16 08:00
12		Brukerveiledning	24 days?	23.08.16 08:00
13		Robotlab bør ha tre forslag	3 days?	27.09.16 08:00
14		Avtale med Nortura ang. transport og igangkjøring	3 days?	11.10.16 08:00
15		Ferdigstilling av skinkemaskin	3 days?	11.10.16 08:00
16		Dokumentasjon av skinkemaskin skal være ferdig	3 days?	18.10.16 08:00
17		Evt. endring og forbedring av dokumentasjon	3 days?	25.10.16 12:00
18		Igangkjøring og transport av skinkemaskin på Tynset	6 days?	01.11.16 09:00
19		Forslag til robotlab bør være ferdig til høring	6 days?	15.11.16 09:00
20		Plan for robotlab og finne komponenter	12 days?	29.11.16 09:00
21		Bestille deler til robotlab	6 days?	03.01.17 09:00
22		Tegne el-skjema	1 day?	03.01.17 08:00
23		Bygge el-skap	2 days?	17.01.17 09:00
24		Montere rullebånd	3 days?	23.01.17 09:00
25		Integretere kamera i robot	3 days?	30.01.17 09:00
26		Test av funksjon av robot/kamera	1 day?	03.02.17 09:00
27		Bruke tid på å bli kjent med programmeringen av robot/kamera	3 days?	07.02.17 09:00
28		Sette opp frekvensomformer	1 day?	10.02.17 09:00
29		Programmere PLS	2 days?	13.02.17 09:00
30		Maskinere deler til bruk i lab	3 days?	27.02.17 09:00
31		Dokumentere labøvelse	10 days?	03.03.17 09:00
32		Rapport skal være ferdig og hovedprosjekt skal være ferdig	3 days?	09.05.17 08:00
33		Forbedringer og endringer i rapport og hovedprosjekt	9 days?	16.05.17 08:00
34		Fremføring av hovedprosjekt	1 day?	06.06.17 08:00

		Name	Duration	Start	Finish
1		Hovedprosjekt	212,667 days?	03.02.16 11:00	27.06.17 15:00
2		Ham-o-matic	32 days?	22.08.16 08:00	09.11.16 13:00
21		Robot	212,667 days?	03.02.16 11:00	27.06.17 15:00
22		Lage fremdriftsplansplan for Robotlab	3 days?	11.10.16 12:00	13.10.16 12:00
23		Utvikling av konsepter	23 days?	25.10.16 10:00	14.12.16 13:00
24		Innhenting av tilbud fra forskjellige leverandører	23 days?	25.10.16 10:00	14.12.16 13:00
25		Valg av konsept	1 day?	20.12.16 11:00	20.12.16 15:00
26		Forprosjektfremføring	1 day	21.12.16 11:00	22.12.16 10:00
27		Forprosjektrapport	30 days	19.10.16 12:00	04.01.17 12:00
28		Bestilling av deler	6 days?	09.01.17 11:00	19.01.17 12:00
29		Hovedrapport	60 days	11.01.17 11:00	31.05.17 11:00
30		Instalere software for pls, kamera og frekvensomformer	2 days	08.02.17 11:00	14.02.17 13:00
31		Utarbeidning av tegningsgrunnlag/skjematikk	10 days	16.01.17 11:00	07.02.17 15:00
32		Oppkobling av El.skap	30 days	25.01.17 11:00	05.04.17 11:00
33		Montere kabelbaner	10 days	25.01.17 11:00	16.02.17 10:00
34		3D-printe plukkobjekter	1 day?	03.02.16 11:00	04.02.16 10:00
35		Montere transportbånd	1 day	22.02.17 11:00	23.02.17 10:00
36		Merking av kabler og komponenter	10 days	01.03.17 11:00	23.03.17 10:00
37		Opprette kommunikasjon mellom alle komponentene	7 days?	10.03.17 11:00	28.03.17 15:00
38		Programmering av Pl.S	22 days	15.03.17 11:00	04.05.17 10:00
39		Programmering av Vision-sensor	5 days	30.03.17 10:00	12.04.17 11:00
40		Programmering av robot	10 days	20.04.17 10:00	16.05.17 13:00
41		Lage dokumentet for labøvelsen	14 days?	24.04.17 12:00	24.05.17 13:00
42		Slutt-test av labøvelsen	1 day?	11.10.16 12:00	11.10.16 15:00
43		Ferdigstilling av rapport	13 days?	30.05.17 10:00	27.06.17 15:00
44		Dokumentert labøvelsen	13 days?	01.05.17 10:00	30.05.17 15:00
45		Fremføring av hovedprosjekt	1 day	29.05.17 10:00	30.05.17 15:00

MØTEREFERAT

Møtenr:	9	Referent: Håvard Vikås	
Møtedato:	25.01.17		
Neste møte:			Møtested: Kongsberg
Møtedeltagere :	Eirik Hatland Sindre Hellingsrud Håvard Vikås	Firma:	Møtt/ikke møtt
Distribusjon:	Prosjektmappe på Dropbox		
SAK	TEKST	ANSVAR	FRIST
1	Sak: Gjennomgang av rapport og fordeling av arbeidsoppgaver. Vedtak: se vedlegg		
	Sak: Tommy har tilbud gruppen en styrekonsoll fra Kongsberg Maritime, som kan brukes som koblingsskap. Vedtak: Gruppen har takket ja. Innhenting pris på koblingsskap er derfor avsluttet.		

	<i>Eirik</i>	<i>Sindre</i>	<i>Håvard</i>
<i>Hamomatic</i> <i>Frist</i> 22.02.17	1.5 Drøfting/diskusjon	1.2 Funksjon	1.2 HMS
	1.6 Konklusjon/oppsummering	2.2 Planlegging	2.1 Referat fra arbeidet
	2.5 Læringsutbytte	2.4 Ressurser	2.3 Kommunikasjon
	1.1.4 Begreper i problemstillingen	2.7 Konklusjon	2.6 Verktøy og praksis
<i>Robotlab</i>	1.2 Frekvensomformer	1.2 Robot og programering	1.2 Vision-system
	1.2 PLS	1.2 Kommunikasjon	1.2 Optikk og lysforhold
		1.2 Integrering	1.2 Software

MØTEREFERAT

Møtenr:	9	Referent: Håvard Vikås	
Møtedato:	29.03.17		
Neste møte:		Møtested:	Kongsberg
Møtedeltagere :	Eirik Hatland Sindre Hellingsrud Håvard Vikås	Firma:	Møtt/ikke møtt
Distribusjon:	Prosjektmappe på Dropbox		
SAK	TEKST	ANSVAR	FRIST
1	Sak: Fordeling av nye arbeidsoppgaver i forbindelse med rapport. Vedtak: Se vedlegg 1		
2	Sak: Behov for veiledning i forbindelse med programmering av vision-sensor. Vedtak: Sindre skal kontakte Eirik Lysen fra Omron, og avtale passende tidspunkt for veiledning gjennom teamview.		
3			

MØTEREFERAT

Møtenr:	9	Referent: Håvard Vikås	
Møtedato:	29.03.17		
Neste møte:			Møtested: Kongsberg
Møtedeltagere :	Eirik Hatland Sindre Hellingsrud Håvard Vikås	Firma:	Møtt/ikke møtt
Distribusjon:	Prosjektmappe på Dropbox		

SAK	TEKST	ANSVAR	FRIST
1	Sak: Ny arbeidsfordeling Vedtak: se vedlegg 1		
2	Sak: Hatt Omron på besøk. Gjennomgang av sysmac studio og oppkobling av enkoder. Vedtak: Iver fra omron skal sette opp et forslag til hvordan rs232 kommunikasjonen skal være.		

Vedlegg 1	Eirik	Sindre	Håvard
Robotlab	1.2 Frekvensomformer -Filter -EMC	Programmering av robot og PLS	1.2 Powersupply 24 V
	1.2 Motor og transportbånd -utveksling -motordata osv	Lage brukerveiledning for bruk av kamera og robot	1.2 Vision-system kordinatsystem
	1.2 PLS -NX-PF0730 -NX-EC0122 -MX2 ethercat card	Encoder -Hvordan funger den	1.2 Optikk og lysforhold
	Kabler og utregninger -skap fra Kongsberg Maritime -kabelføringer -PC og skjermkort, kriterier for PC	1.1 Robot Motoman sia20	1.2 Software -SYSMAC-LE201L -Touchfinder
	Brytere -innganger og utganger -vribrytere og funksjonalitet		Pneumatikk, griperverktøy
	1.2 Sikkerhetsrele -hensikt		Elektriske tegninger
	1.2 Kommunikasjon -Switch og ethernet kobling -RS232		3D-printing, Objekter, tegninger og g-koder Tegning av festebrakett

Samarbeidskontrakt

Denne kontrakten gjelder for Håvard Vikås, Eirik Hatland og Sindre Hellingsrud fra august 2016 til juni 2017.

Gruppen har i fellesskap kommet frem til følgende spilleregler for samarbeidet i hovedprosjektet.

Kommunikasjon

- Kommunikasjonen skal foregå formell og saklig.
- Ved hjemnearbeid skal mail eller Facebook messenger brukes til kommunikasjon og dokumentflyt.
- Alle gruppemedlemmene skal ha tilgang til den kommunikasjonen som er gjort med oppdragsgiver.
- Kontakt med veileder gjøres etter behov.

Dokumentasjon

- Dokumenter og annen relevant dokumentasjon knyttet til prosjektet skal lastes opp på en felles dropbox.
- Alle deltakerne har et felles ansvar for at dokumenter og mapper holdes ryddig.
- Før dokumenter blir sendt videre, skal alle ha lest igjennom både for korrektur og for godkjenning av innhold.

Beslutningsprosesser

- Avgjørelser gjøres i fellesskap og alle skal være enige så langt det lar seg gjøre, hvis ikke holdes det en avstemming hvor flertallet bestemmer. Siden gruppen består av tre medlemmer så vil flertallets stemmer vinne.
- Beslutninger som tas på møter skal dokumenteres i møtereferatet.
- Konflikter skal tas opp i gruppen på en saklig måte, slik at alle kan være med å finne en løsning på konflikten.

Forpliktelser

- Gruppemedlemmene forplikter seg til å bruke minimum ni timer i uka.
- Deltakerne forplikter seg til å stille på planlagte møter. Ved eventuelle hindringer skal det gis beskjed minst 24 timer før møtet.

- Deltakerne forplikter seg til å gjennomføre de tildelte arbeidsoppgaver innenfor tidsfristene.
- Deltakerne forplikter seg til å legge ned de timer som kreves for at oppdragsgiver skal bli fornøyd med prosjektet.

Sanksjoner

- Ved brudd på avtalen vil den det gjelder få inntil to advarsler, den tredje gangen må personen forlate gruppen.

Ledelse / roller

- Gruppen har valgt prosjektleder, Eirik Hatland
 - Lederens oppgaver vil i hovedsak bestå av å holde en oversikt over fremgang og status på prosjektet, og følge opp medlemmene og passe på at de gjør det de skal.
 - Prosjektleder er ansvarlig kommunikasjon med oppdragsgiver.
 - Ansvarlig for at frister blir holdt og fremdriftsplanen blir fulgt.
-
- Gruppe har valgt en fast sekretær, Håvard Vikås
 - Sekretær har ansvar for møtereferat, dokumenter og vedlegg til rapporten. Sekretæren har ansvar for å samle alle dokumenter og laste de opp til prosjektmappen på dropbox og sørge for at alle medlemmene har den informasjonene de trenger. Notere eventuelle endringer i fremdriftsplanen.
 - Sekretæren har også ansvar for at HMS reglementet til oppdragsgiver og skolen blir fulgt ved gjennomføring av prosjektet.
-
- Gruppen har valgt en logistikkansvarlig, Sindre Hellingsrud
 - Logistikkansvarlig har ansvar for innkjøp, komponentliste, budsjett, materialflyten og skal sørge for at gruppen har tilgang på dataprogrammene og verktøyene som trengs ved gjennomføring av prosjektet.

Selv om hver enkelt deltaker har fått tildelt egne ansvarsområder forventes det at samtlige deltakere bidrar med arbeidsoppgaver utenfor sine egne ansvarsområder.

Normer

- Alle som har innspill skal bli hørt
- Alle ideer som kommer opp skal vurderes

- Det skal strebes etter å holde en god tone i gruppen for et godt samarbeid, slik at det er lettere å komme med tilbakemeldinger både på det som er bra, og på det som er mindre bra.
- Alle gruppemedlemmene er like viktige!
- Det er lov å si fra vis tiden ikke blir brukt effektivt.

Mål for prosjektet

- Målet for prosjektet er at oppdragsgiver skal være fornøyd med det utførte arbeidet.
- Målet for prosjektet er at deltakerne skal sitte igjen med kunnskap om hvordan man planlegger og gjennomfører et prosjekt med oppdragsgiver.

Kontrakten er godkjent:

Dato: 25.08.2016

Sted: Kongsvinger

Håvard Vikås

Håvard Vikås

Eirik Hatland

Eirik Hatland

Sindre Hellingsrud

Sindre Hellingsrud