

Oppsett av buss-kommunikasjon mellom Kuka, Beckhoff og Wago

Wago:

Sett opp programmet med riktig komponenter, dette kan gjøres manuelt eller ved å bruke «scan» funksjonen (koble da til pls-en enten på switchen eller det trådløse nettverket og sett opp ditt nettverkskort til å ha ip i samme subnettet som wago pls-en). Vi legger så til en «Generic_MODBUS_master» som vi kobler opp mot pls-en i «Device structure» (finnes master for TCP, UDP og RTU, vær obs på det når du velger). Dobbeltklikker på pls-en i «Network/devices» og får opp en «MODBUS slave» meny. Høgreklikker du på «PLC variables» kan du enten legge til variable eller legge til fra program. Sist nevnte er den enkleste, men forutsetter at du har variablene klar. Du velger om dette skal være coil eller register, men husk adressen du bruker. For å endre fra output til input og omvendt, dobbeltrykk under «Direction» og den variabelen du ønsker å skifte.

Det er også mulig å bruke wago sitt bibliotek for modbus server, men husk da å deaktivere den innebygde modbus masteren på port 502.

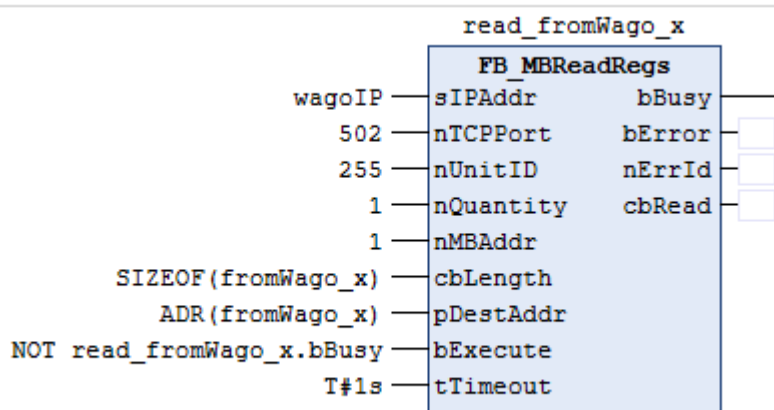
Beckhoff – Wago:

For å sette opp modbus i Beckhoff pls-en bruker vi Twincat, og vi bruker Beckhoff eget bibliotek for modbus, «Tc2_ModbusSrv». Først må alle komponenter legges til i programmet, så kan du ved å dobbeltklikke på «references» legge til biblioteket. Ved å lese dokumentasjonen til biblioteket ser vi at vi har mange blokker vi kan bruke ut i fra hvilket register man ønsker å bruke. I dokumentasjonen går det også godt fram hvilke variabler som trengs.

Eksempel på oppsett av FB_MBReadRegs:

Postadresse	Org.nr. 974 767 880	Besøksadresse	Telefon	Saksbehandler
Postboks 1517	E-post:	Larsgårdsvegen 2	+ 47 70161200	Ottar L. Osen
6025 Ålesund	postmottak@ie.ntnu.no	Blokk B311		ottar.osen@ntnu.no
Norway	http://www.ntnu.no/iir			Tlf: + 47 70161283

Adresser korrespondanse til saksbehandlerenhet. Husk å oppgi referanse.

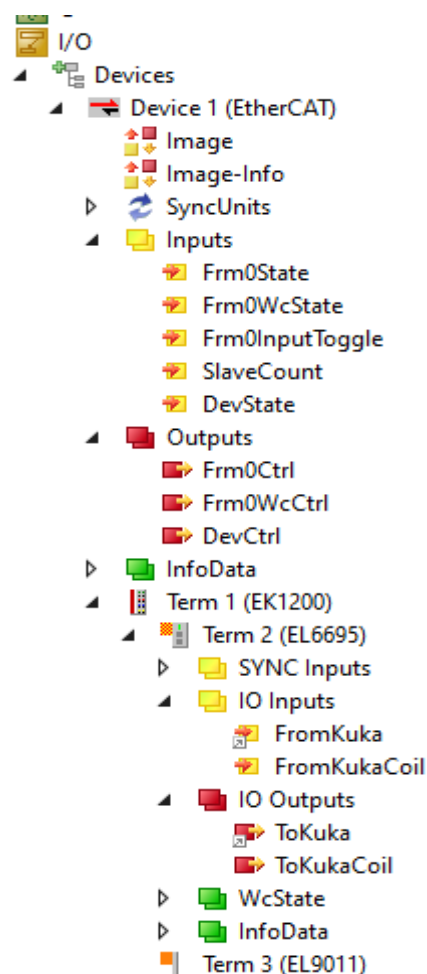


Verdien vi leser av vil da sendes til variabelen «fromWago_x». Sett opp en slik blokk til hver av variablene som skal sendes eller mottas (husk riktig blokk).

Beckhoff – Kuka:

Siden både CX5120 og KUKA sunrise kabinettet er mastere i EtherCAT-nettverket er det koblet en EL6695 EtherCAT Bridge til IPCen. Dette er en fysisk komponent som gjør det mulig å kommunisere med KUKA sunrise kabinettet via EtherCAT. Etter at EtherCAT bridgen ble installert ble modulen lagt til i TwinCAT 3 automatisk ved oppdatering av IPCens IO-moduler. Under EtherCAT fanen ligger IO Inputs og IO Outputs som gir variablene som skal sendes til og mottas fra KUKA sunrise kabinettet. Ved å høyre klikke på IO inputs eller IO outputs kan du legge til variabler. Det vises en meny hvor variabelens navn og datatype må velges.

For å koble disse variablene opp mot variabler i PLS-programmet ble det konstruert globale variabler med samme navn (ikke nødvendig, men ble gjort for enkelhets skyld). Disse variablene ble skrevet på formen ToKuka AT %Q* : USINT; og FromKukaCoil AT %I* : BOOL; hvor %Q* setter variablene til å være outputs og %I* setter variablene til å være inputs. Disse variablene kan bli funnet under CX5120 Instance. Etter å ha laget variablene i PLS-programmet, er det viktig å bygge prosjektet på nytt slik at EtherCAT bridge modulen var oppdatert på pls-en sine inputs og outputs. For å koble variablene på EtherCAT bridge modulen



mot variablene på PLSen dobbeltklikkes det på en av variablene i EtherCAT bridge modulen. I menyen som kommer opp vises PLS-variablene som har tilsvarende datatype. Når PLS variabelen er linket til EtherCAT-variabelen vil det vises en liten hvit boks med ei pil i hjørnet av både PLS- og EtherCAT-variabelen. De globale variablene kunne nå brukes fritt i PLS-programmet og deres verdier ble sendt fra EtherCAT bridgen.

KUKA – Beckhoff:

For oppsett av variabler i Kuka brukes to programmer, «Sunrise Workbench» for skrive programmet til Kuka samt å laste over programmet til Kuka. Det andre programmet er «WorkVisual», som vi bruker til å sette opp komponentene til Kuka og mappe variabler.

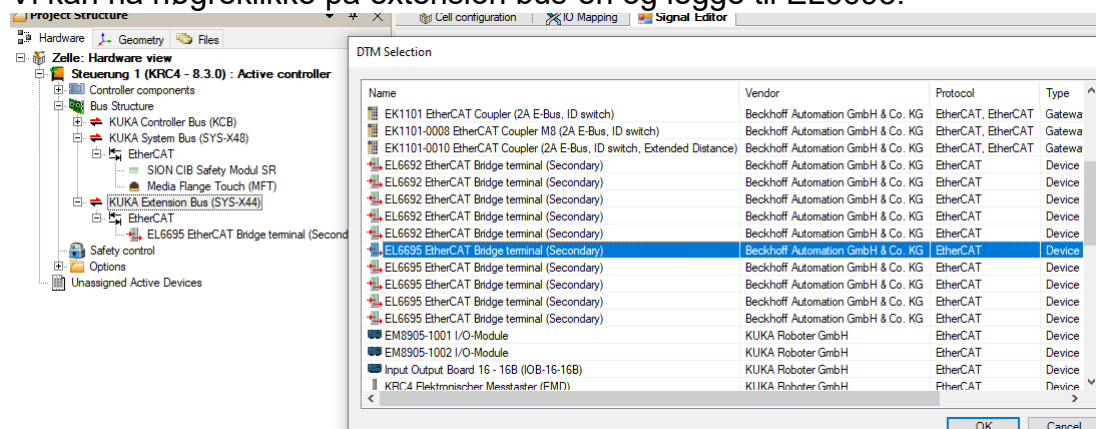
NB! Pga kompatibilitets problemer med windows 10 skal kun Workvisual V4.0.17_build 0070 være den eneste som fungerer med windows 10 samt med Sunrise 1.11.0.7.

For å sette opp IO høgreklikker vi på prosjektet vi har opprettet i Sunrise, så «new» og «I/O Configuration». Workvisual vil nå automatisk åpnes. Ved å høgreklikke på kontrolleren (KRC4 – 8.3.0) og trykke «set as active controller», kan vi konfigurere kontrolleren. Under «Bus Structure» legger vi til en «KUKA Extension Bus» ved å høgreklikke, trykke «add» og velge «KUKA Extension Bus» fra menyen.

Så må ethercat bridgen(xml-fil) legges til i programmet da den ikke ligger der som standard. Filen finnes i twincat 3 under C:\TwinCAT\3.1\Config\io\EtherCAT\Beckhoff EL66xx.xml eller så finnes den på Beckhoff sine sider.

Denne filen trengs for å kunne sette opp en buss mellom KUKA KRC-kontrolleren og EtherCAT-bridgen. Etter enhetsbeskrivelsen er lastet til PCen, kan den importeres ved å trykke følgende: File -> Import/Export -> Import device description file og så velge xml-fila. Når dette er gjort, ligger enhetsbeskrivelsene i DTM-Catalog, som kan finnes ved å trykke på følgende: Extras -> DTM-Catalog.

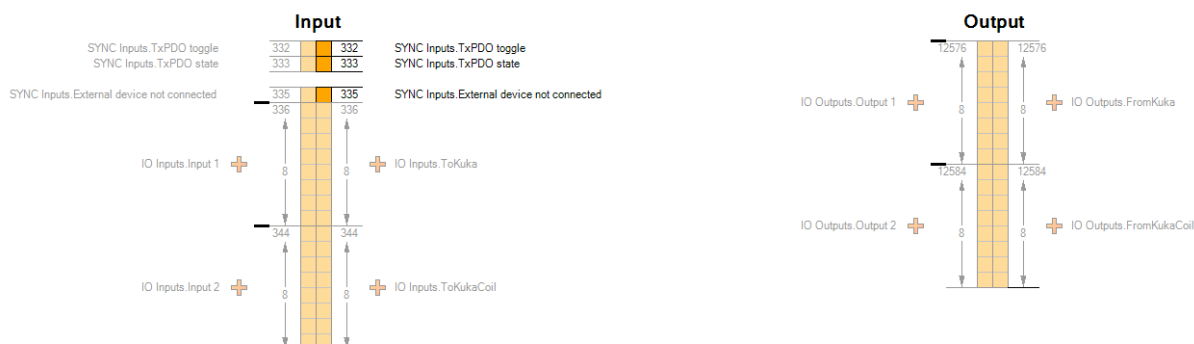
Vi kan nå høgreklikke på extension bus-en og legge til EL6695.



Dobbeltklikk på EtherCAT-bridgen gjør at det kommer opp ett nytt vindu i WorkVisual. Dette er et vindu som skal representere EtherCAT-bridgen. EtherCAT-bridgen trenger at antallet variabler som skal konfigureres fra Beckhoff IPCen, stemmer overens med antall variabler som roboten skal kunne ta inn. Dette kan gjøres under Slave Settings i menyen som vist øverst. Der kommer det opp både inputs og outputs. EtherCAT-bridgen må også vite hvilken datatype som skal være til hver enkelt variabel. Når alle variabler er lagt til, klikken knappen «OK» og I/O-mappingen kan starte.

I/O Mapping skjer i vinduet «I/O Mapping». Bildet over viser at I/O Mapping vinduet ligger over EtherCAT-vinduet. Bildet under viser I/O Mapping-vinduet. For å finne den feltbussen som skal brukes, trykkes knappen Fieldbusses. Da kommer EtherCAT-bridgen opp og variablene kommer opp på skjermen under.

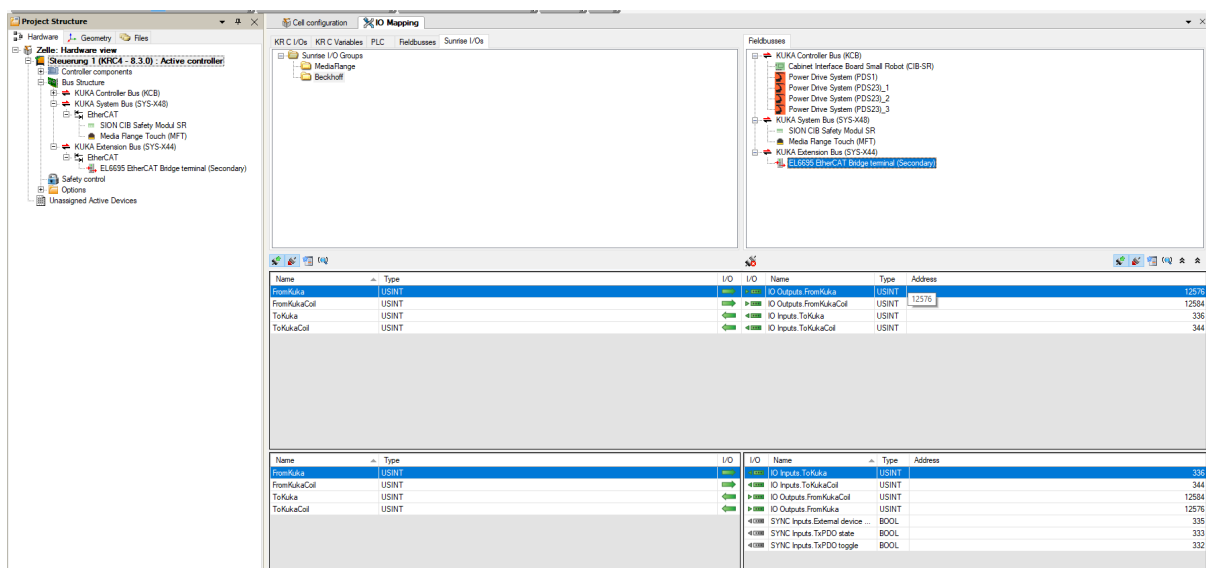
Der ligger allerede variabler der fra før, som ble opprettet i Slave Settings. Før mappingen av variablene starter, kan det være greit å endre navn på variablene som allerede er skapt. Navnene endres på ved å trykke på den lille pennen i midten nede høyre hjørne. Da kommer «Edit signals»-menyen opp, se bildet under.



I den menyen kommer alle variablene som skal være skapt til EtherCAT-bridgen opp, og størrelsen på hver variabel blir også vist, i bytes. For å legge variablene som skal inn til Sunrise WorkBench, trykkes knappen med Sunrise I/Os, så skal knappen Create signals at the provider, helt til venstre nede i hjørnet i I/O mapping-vinduet, trykkes. Da kommer det opp et nytt vindu der det er mulig legge til I/O-grupper og variabler. Det første som skal gjøres er å lage en ny I/O-gruppe.

Da trykkes den store knappen med «Create» på, så skrive inn det navnet som passer til gruppen som skal lages. Det er også mulig legge inn beskrivelse om hva I/O-gruppen skal gjøre. Deretter lages variablene som trengs. Disse variablene MÅ være identiske i forhold til de variablene som ble skapt tidligere. De må ha samme navn, samme retning (input/output), samme datatype og de må være like store (bytes).

Når alle nødvendige variabler er laget, skal de mappe til de andre som ble laget tidligere. Det er da viktig at alle variabel med samme navn blir mappet ilag. Hvert enkelt «par» av variabler skal mappes hver for seg. For eksempel dersom «ToKuka» skal mappes, velges den variabelen på begge sider og trykker på Connect. Etter at alle variablene er mapped og pilene som viser retningen på variabelen er blitt grønn er WorkVisual ferdig.



Det siste som skjer i WorkVisual er å eksportere variablene til Sunrise WorkBench. Dette kan gjøres ved å gå på File -> Import / Export -> Export I/O-configuration to Sunrise Workbench project. Workvisual kan lukkes og Sunrise programmet kan synkroniseres med Kuka.