

Intelligent fastighetsautomation



Martin Viding

Dept. of Industrial Electrical Engineering and Automation
Lund University

Abstract

This report describes the integration of hardware and software into a building automation system. Commercial automation products from Beckhoff Automation represent the core system and also components from Precise Biometrics, MySQL and Microsoft have been used.

Implemented functions are automatic and manual light control, alarm, access control and logging of users and temperature. The integration of database, PLC system and web has been a good deal of the work.

The entire system has been developed and tested in a lab environment, but has not been installed because of the time limit.

Sammanfattning

Denna rapport beskriver sammansättningen av hårdvara och programvara för ett fastighetsautomationssystem. Kommersiella automationskomponenter från Beckhoff Automation utgör stommen i systemet och dessutom har produkter från Precise Biometrics, MySQL och Microsoft använts.

Funktioner som implementerats är automatisk och manuell ljussättning, larm, accesskontroll, användar- och temperaturloggning. Integration av databas, PLC-system och web har utgjort en stor del av arbetet.

Systemet har i sin helhet utvecklats och testats i labmiljö, men har av tidsskäl ej slutligt installerats.

Förord

I september 2006 kontaktade jag Beckhoffs VD Håkan Brandt och hörde mig för om möjligheterna att utföra mitt examensarbete på automationsföretaget i Malmö. Intresse fanns från företagets sida och inom kort presenterades en idé om utvecklingen av ett styrsystem för den egna fastigheten. Hela upplägget passade mig perfekt, med förkunskaper från bland annat tre automationskurser på LTH, och således behövde jag inte mycket betänketid för att tacka ja.

Jag rådfrågade min tilltänkta handledare på institutionen för Industriell Elektroteknik och Automation, Gunnar Lindstedt, och även han ansåg att det kunde bli ett spännande projekt och var villig att ta rollen som handledare. Saken var klar och i november påbörjades arbetet.

Tillkännagivanden

Projektet har enligt mitt tycke fungerat klockrent från start till slut och jag vill passa på att tacka följande personer:

Mina handledare, Roger Merkel på Beckhoff och Gunnar Lindstedt på IEA, för hjälp i arbetet, både med direkta projektdelar och praktiska frågor kring arbetet.

Daniel Bengtsson på Beckhoff, för mycket engagemang och idé-bollande, samt skjuts till och från Malmö.

Alla andra anställda på Beckhoff – Håkan, Henrik, Patrick, Jon, Roger2, Tessian, Björn och Lotta – för fikapauserna, den ljuva musiken, och sist men inte minst de pretentiösa personalfesterna.

Innehållsförteckning

| | | |
|-----|--------------------------------|----|
| 1 | Introduktion | 4 |
| 1.1 | Bakgrund | 4 |
| 1.2 | Syfte | 4 |
| 1.3 | Resultat | 4 |
| 1.4 | Avgränsningar | 6 |
| 2 | Teori | 7 |
| 2.1 | Hårdvara | 7 |
| 2.2 | TwinCAT | 15 |
| 2.3 | Microsoft Visual Studio .NET | 17 |
| 2.4 | MySQL | 18 |
| 3 | Utförande | 19 |
| 3.1 | PLC-projekt | 20 |
| 3.2 | Databas | 25 |
| 3.3 | Websidor | 26 |
| 3.4 | Integration | 28 |
| 4 | Diskussion | 33 |
| 4.1 | Problem | 33 |
| 4.2 | Personliga lärdomar | 34 |
| 4.2 | Vidareutveckling av systemet | 35 |
| 5 | Referenser | 36 |
| | Appendix | 37 |
| A1 | Databashantering | 37 |
| A2 | Adressering av variabler | 38 |
| A3 | Konfiguration i System Manager | 39 |
| A4 | Förkortningar | 46 |

1 Introduktion

Traditionellt sett vill konsumenter ha den senaste teknologin och de bekvämaste lösningarna till minsta möjliga kostnad. Utvecklingen går mot ett ännu mer datoriserat samhälle samtidigt som miljöfrågor kommer allt mer i fokus.

En marknad som använder ny teknik för att åstadkomma smarta och billiga lösningar och samtidigt tar hänsyn till miljön torde således vara attraktiv.

Fastighetsautomation som integrerar ny teknologi ytterligare i privata boenden är en marknad med ovan nämnda egenskaper. Intelligent styrning av larm och accesskontroll och integration med Internet ger bekvämlighet, och automatisk ljus- och värmereglering kan reducera energiförbrukning.

Detta examensarbete har utförts på Beckhoff Automation AB i Malmö och omfattar utvecklingen av automationen till delar av en fastighet. Ljusstyrning, accessreglering och temperaturkontroll har implementerats i syfte att fungera som demo i företagets satsning att etablera sig på fastighetsmarknaden.

Ambitionen har varit att implementera ett automatiserat PLC-styrssystem integrerat med en databas, och med möjlighet till viss styrning och statuskontroll via ett web-baserat HMI.

1.1 Bakgrund

Beckhoff Automation AB levererar öppna automationsprodukter för alla typer av applikationer, från CNC-maskiner till intelligent fastighetsautomation. Plattformen TwinCAT integrerar realtidsstyrning med PLC, NC och CNC-funktioner i en och samma hårdvara.

Produkterna används i nuläget främst i industrin men till viss del även inom fastighetsbranschen. Som exempel kan ges Örebrobostäder, vars fastigheter automatiserats med hjälp av Beckhoff.

1.2 Syfte

Målet med examensarbetet har varit att automatisera delar av funktionerna i Beckhoffs företagslokal i Malmö, i syfte att kunna demonstrera möjligheterna, flexibiliteten och enkelheten med Beckhoffs produkter för eventuella blivande kunder.

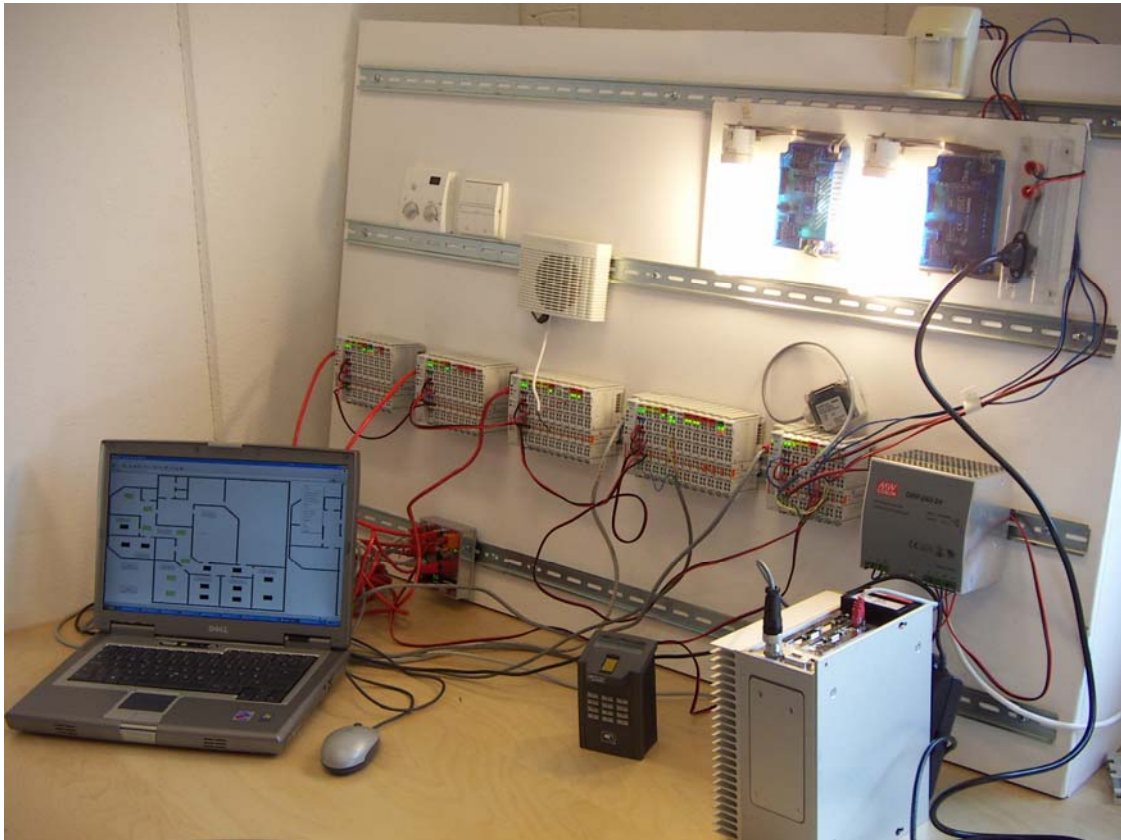
Integrationen av olika system, dels interna Beckhoff- dels utomstående enheter, har utgjort en stor del av arbetet.

1.3 Resultat

Den medvetna intentionen vid arbetets start var att hinna med att implementera så mycket som möjligt av ett förmodligen för omfattande slutmål, för att vid en senare utvärdering se över vad som gjorts och ytterligare rimligen skulle hinna med att göras. Inledningsvis värderades också framtida eventuella prioriteringar.

Som huvudsakligt slutmål stod att finna grunden för det övergripande systemet, det vill säga integrationen av systemen. Tillägg av ytterligare funktioner, som är mer trivialt och kräver ett fungerande underliggande system, fick komma som en bonus.

Genomförandet har resulterat i en slutprodukt där de underliggande funktionerna, det vill säga integration med databas, PLC-system och web, är i drift. Funktioner som automatisk och manuell ljussättning, larm, accesskontroll, användar- och temperaturloggning har också implementerats.



Figur 1.1: Översikt av det färdiga styrsystemet.

1.4 Avgränsningar

Delar som fanns med i den ursprungliga funktionsbeskrivningen men som på grund av tidsbrist ej kunnat beaktas är:

- värmestyrning och framtagning av reglerfunktioner för hur yttertemperatur påverkar innetemperatur,
- energi- och vattenförbrukning samt
- videoövervakning.

De två senare nämnda funktionerna kräver mindre ansträngningar att addera till det nu fungerande systemet, medan den första är mindre trivial.

2 Teori

I detta kapitel beskrivs i projektet ingående komponenter, hård- liksom mjukvara.

2.1 Hårdvara

Komponenter som används är i huvudsak produkter ur Beckhoffs sortiment, men även från andra tillverkare som exempelvis fingeravtrycksläsaren från Precise Biometrics.

I Beckhoffs sortiment finns en mängd olika industri-PCs, inbyggda PCs, busskopplare, bussterminaler och bussterminal-controllere, alla med olika funktion och prestanda, och som därmed är olika lämpade för olika användningsområden.

Bussterminaler, som kopplas på en busskopplare, samlar in signaler och transporterar dem vidare till densamma, varefter informationen bearbetas alternativt skickas vidare till överordnat system.

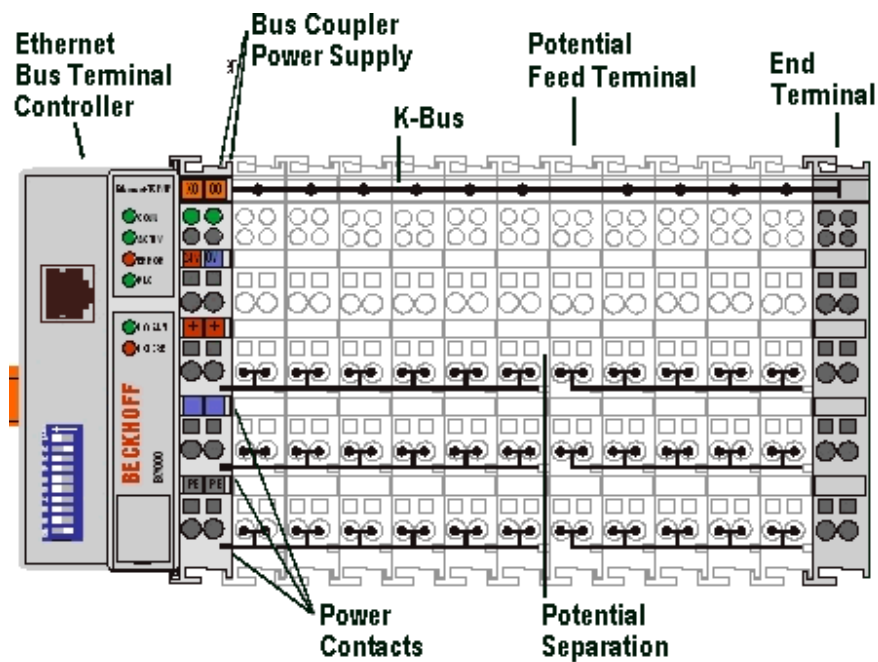
Olika terminaler, beroende på vilken typ av information som ska hämtas samt på hur denna transporteras, finns att tillgå. Även busskopplare finns i många olika utföranden beroende på vilken typ av buss som länkar till det överordnade systemet.

2.1.1 Beckhoff Bus Terminal System

Mer information i *Beckhoff Information System*. Detta är gränssnittet mellan ett fältbussystem och sensor-nivån. En enhet består av en busskopplare som huvudstation och med möjlighet till upp till 64 bussterminaler med olika egenskaper, beroende på vilket protokoll signaler ska samlas in med.

I/O-signaler transporteras mellan bussterminaler och busskopplaren över en så kallad K-buss, varefter signaler på den ena änden av enheten länkas med lämplig buss till sensor-systemet och signaler på den andra änden länkas till högre nivå.

Vanligen utgör dessa enheter noder i ett centraliserat system, dock ej nödvändigtvis eftersom det finns busskopplare med inbyggd CPU, då kallade Bus Terminal Controllers, med vilka intelligens kan decentraliseras och flyttas ut i noderna.



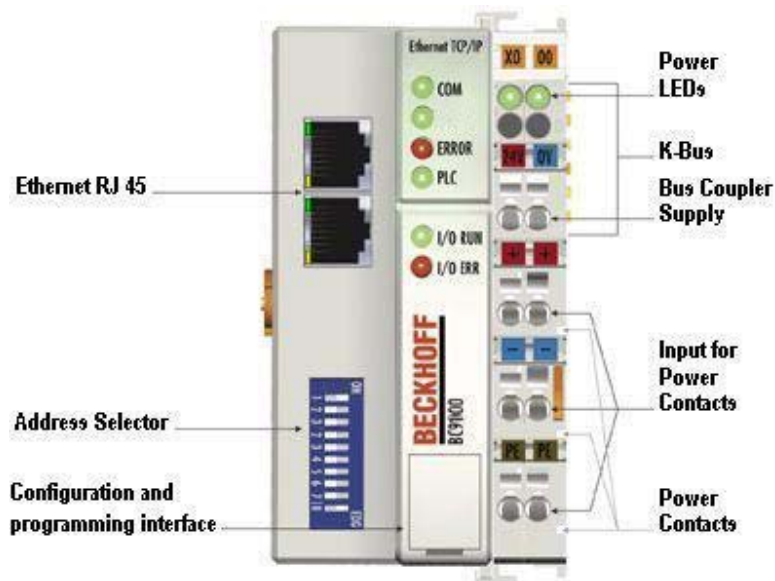
Figur 2.1: Principbild av ett Beckhoff bussterminal-system.

2.1.1.1 Bus Terminal Controller för Ethernet, BC9100

Busskopplaren BC9100 samlar med hjälp av K-bussen in data från sina påkopplade terminaler och kan därefter, beroende på hur intelligensen i systemet är fördelad, antingen agera som en enkel fältbusnod och skicka vidare informationen till överordnat system eller bearbeta den själv. Detta är alltså en busskopplare med intelligens, det vill säga med en inbyggd PLC, – Bus Terminal Controller.

Dataöverföring till överordnat system sker via Ethernet, och med BC9100 som har två Ethernetportar finns dessutom möjlighet att arrangera flera nivåer i systemet.

Strömförsörjningen på 24 Volt matas vidare ut till terminaler, men även andra potentialgrupper kan vid behov åstadkommas med speciella så kallade potential feed terminals. Även om varianter för andra spänningar finns så är 24 Volt vanligast.

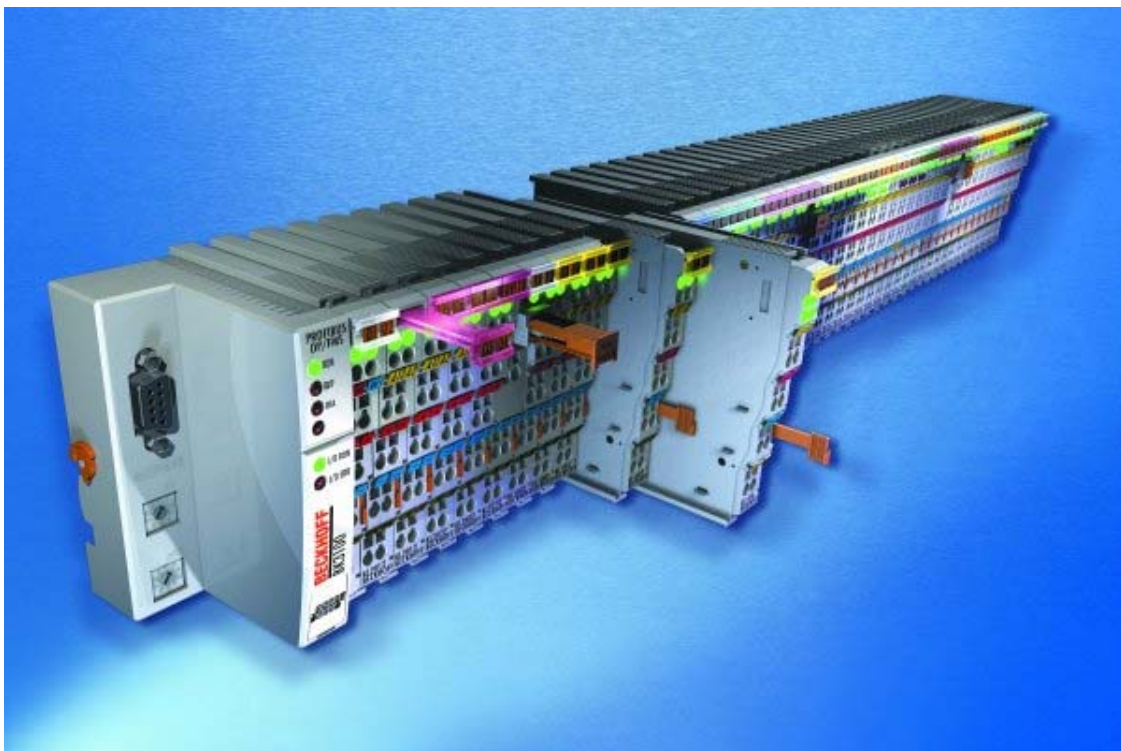


Figur 2.2: BC9100.

2.1.1.2 Bus Terminals

Åtskilliga terminaler för ett stort antal bussystem finns att tillgå i Beckoffs sortiment. Enkla digitala in- och utgångsterminaler (KL1002 resp KL2012) kan användas för till exempel rörelsesensorer, larmslingor och lysrörskontakter, medan mer avancerade behövs för bussystem som DALI och EnOcean vid ljussättning.

Terminaler fästs enkelt med snäppfastsättning på DIN-skena, vilket gör det möjligt att koppla terminaler i valfri ordning eller att efter hand snabbt addera fler, eller ta bort befintliga.



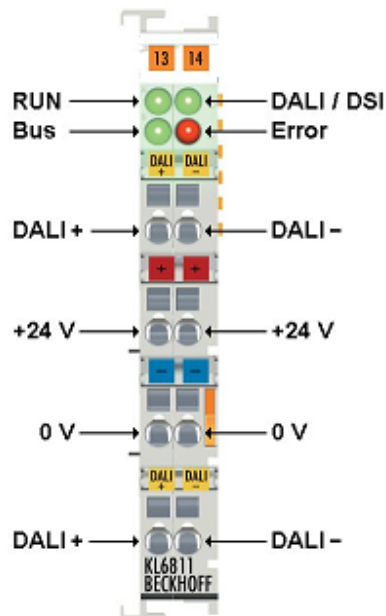
Figur 2.3: Terminaler på en busskopplare.

2.1.1.2.1 KL6811 – Master Terminal för DALI

DALI (Digital Addressing Lighting Interface) är ett standardiserat digitalt interface för ljussättning. KL6811 Bus Terminal (DALI master) fästs som en vanlig terminal på valfri busskopplare och är därför fältbussoberoende.

Mastern skickar sedan DALI-data ut till slavar som därefter sätter korrekt nivå på sina ljusarmaturer. Intelligensen är inte helt centraliserad, utan viss data, som till exempel aktuell ljusnivå, dimningshastighet och internadress, lagras i slavar i form av variabler.

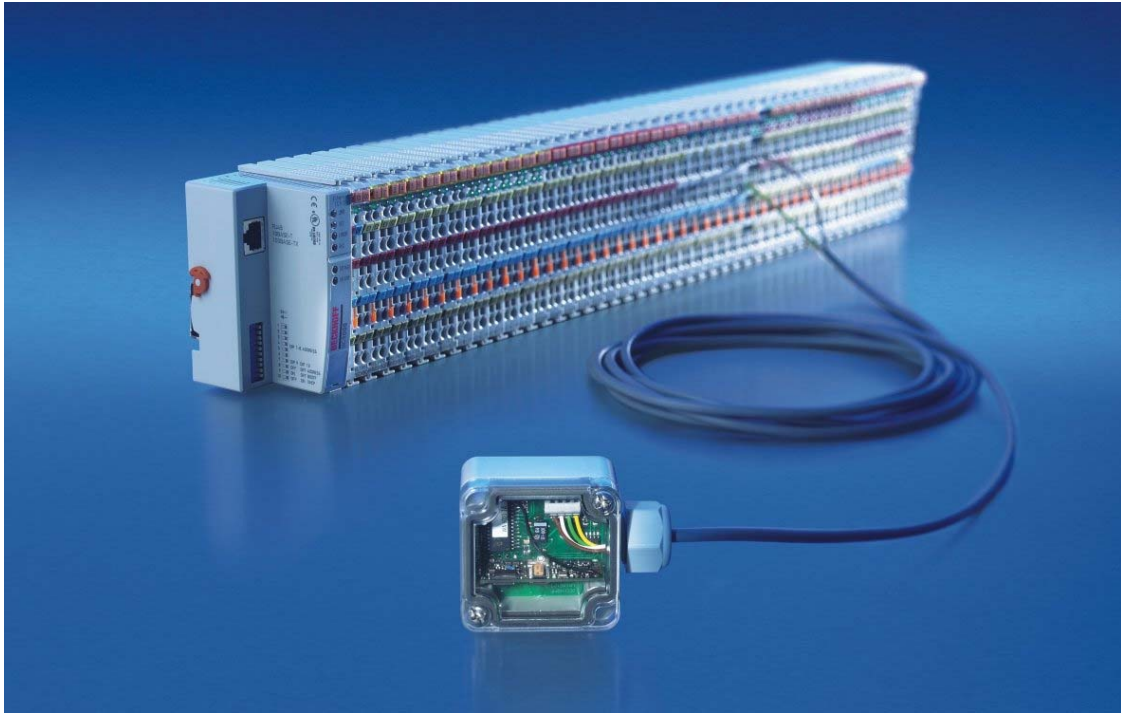
Möjlighet finns att ordna ljusarmaturer i grupper, att dimma olika enheter eller grupper med valfri hastighet, att ställa in olika scenarion etcetera.



Figur 2.4: DALI-terminal

2.1.1.2.2 KL6021-0023, KL6023 EnOcean

EnOcean-givare, i form av temperaturgivare eller knappsats för styrning av valfri funktion, skickar data till en EnOcean-mottagare (KL6023) som länkas till Beckhoffs bussystem via seriekommunikations-terminalen KL6021-0023.



Figur 2.5: EnOcean-mottagare KL023 kopplad till terminal.

Den trådlösa temperaturgivaren drivs av att solljus laddar en kondensator som gör att den därför fungerar under cirka ett dygn i mörker.

Knappsatsen är också fri från yttre elförsörjning och batterier, och drivs istället av energin från en piezoelektrisk givare som aktiveras av knapptryck och kan skicka vidare viss status-information till mottagaren.

Det finns ingen egentlig gräns för antal givare, men för att undvika kollisioner rekommenderas inte fler än hundra per mottagare.



Figur 2.6: Temperaturgivare och knappsats

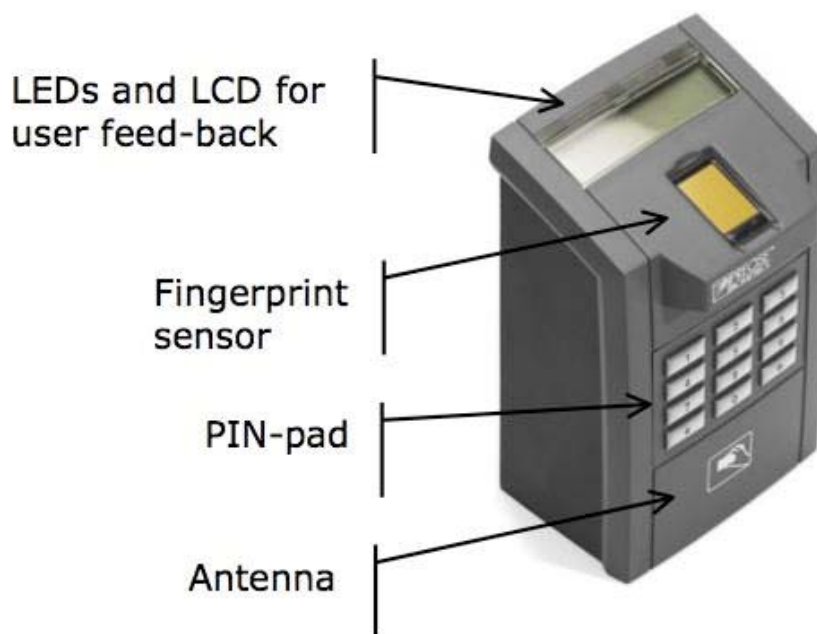
2.1.2 Precise BioAccess 200

Precise BioAccess 200 är en fingeravtrycksläsare utvecklad av Precise Biometrics som kommunicerar med överordnade system på ett av flera valbara sätt. För närvarande stöds Wiegand, DataClock samt RS 232 för att skicka vidare information om användare.

En så kallad Enrolment Station används för att registrera användare och lagra dessa på ”smart cards”. Med hjälp av tillhörande mjukvara lagras fingeravtryck på kortet som blir färdigt att använda.

Kortet läses genom att det hålls framför läsarens antenn, varefter fingeravtryckssensorn scannar det finger som placeras på den. Om fingeravtrycket matchar det som finns lagrat på kortet skickas data till systemet med valfritt protokoll, om inte skickas ingen data.

Två typer av kort kan användas, MIFARE standard och MIFARE DESFire. Standardkortet är ett vanligt minneskort där information kan lagras i olika sektorer och kan programmeras att läsa, skriva etcetera, medan DESFire innehåller en mikroprocessor. För mer information, se *BioAccess 200 User Manual*.



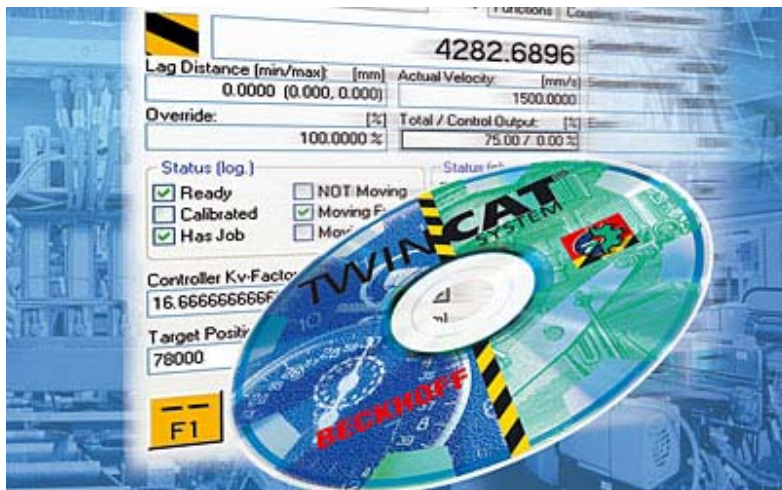
Figur 2.7: Precise Biometrics fingeravtrycksläsare.

2.2 TwinCAT

Styrssystem kräver en bra tidshantering med korta responstider och möjlighet till periodisk exekvering av kod utan tidsförskjutningar och exempelvis Windows NT är inte optimerat för detta.

Beckhoff har av denna anledning gjort en realtidsutvidgning för Windows NT, som gör att ett styrssystem har en exakt tidsbas och exekverar program med maximal prioritet oberoende av andra processoroperationer.

Mjukvarusystemet TwinCAT får en vanlig persondator att fungera som ett realtidsstyrssystem, med bland annat inbyggt IEC 61131-3 PLC-system, stöd för alla vanliga fältbussar, och datakommunikation med standarder som OPC, OCX, DLL etc.



Figur 2.8: TwinCAT

2.2.1 TwinCAT PLC

TwinCAT PLC är en mjukvaru-PLC som körs på en vanlig PC tillsammans med valfritt operativsystem ur Windows NT -familjen. Upp till fyra virtuella PLC CPU:s tillåts på en och samma PC. Utvecklingsmiljö med stöd för alla standarder i IEC 61131-3 inkluderas vid installationen.

2.2.2 TwinCAT System Manager

TwinCAT System Manager är systemets konfigurationscenter. Systemet scannas in via valfri fältbuss (till exempel PROFIBUS, CANOpen, Ethernet och USB) och presenteras grafiskt varpå varje servers processbild åskådliggörs.

Förhållanden mellan olika PLC-system, –program och I/O-kanaler definieras. Variabler och I/O från ett system skall göras tillgängligt i alla system där de behövs. Utbyte av information mellan systemen sker i en periodisk scanningsprocess.

2.2.3 TwinCAT ADS

ADS (Automation Device Specification) är ett gränssnitt utvecklat för att möjliggöra kommunikation mellan olika mjukvarumoduler. De olika modulerna behandlas som oberoende enheter och kategoriseras i servrar och klienter. En server bistår klienter med operationer.

ADS-kontakt kan ske mellan enheter på samma dator, men också mellan olika datorer.

Data skickas enheter emellan via en så kallad "message router" som tar hand om alla meddelanden inom systemet och över tcp/ip-länkar. En "message router" finns på alla TwinCAT PC och alla BCxxxx Bus Controllers. Den identifieras av sitt AdsAmsNetId som kan liknas vid ett TCP/IP-nummer. Som regel brukar också AdsAmsNetId göras till en förlängning av datorns TCP/IP-adress.

ADS-enheter identifieras av ett ADSPortNr, vilket gör att message routern kan hålla reda på dem.

2.2.4 TwinCAT library

Som tillhör till TwinCAT finns ett stort antal bibliotek att tillgå, innehållande funktioner, funktionsblock, variabler med mera. Dessa bibliotek importeras i PLC-projekt vid behov. Exempelvis finns ett bibliotek för DALI med bland annat de olika datastrukturer som används vid dataöverföringen och funktionsblock för att dimma lampor.

Utöver DALI används även biblioteken EnOcean, COMlib (för seriekommunikation RS 232) samt standardbibliotek med vanliga funktioner som timers och trigger.

Utförligare beskrivningar hittas i *Beckhoff Information System*.

2.3 Microsoft Visual Studio .NET 2003

.NET är benämningen på Microsofts strategi att få olika enheter och tekniker att integreras och enkelt kunna samverka. Exempelvis handlar det om att program som är utvecklade med olika tekniker ska kunna köras under samma plattform, eller att applikationer på olika serverplattformar enkelt ska kunna utbyta information i ett gemensamt format.

.NET-verktyget Visual Studio är en programutvecklingsmiljö från Microsoft som möjliggör utvecklande av både internetbaserade applikationer och Windows-applikationer. Fyra olika programspråk stöds, nämligen Visual Basic, C++, C# och JavaScript, vilka genererar ASP .NET som sedan en server, med lämplig programvara installerad, tolkar.

2.3.1 Windows Applications och Web applications

Möjlighet till skapande av flera olika sorters applikationer finns. En webapplikation genererar websidor vars innehåll exekveras när de anropas med sina respektive adresser. En Windows-applikation å andra sidan installeras som ett program som vid anrop utför de operationer som angivits.

2.3.2 ASP .NET

ASP (Active Server Pages) innebär som namnet vittnar om att kod körs på servern istället för hos användare, och är ett samlingsnamn för olika språk som kan användas i webfiler för att köras på en server.

ASP .NET är den andra generationen av ASP och ingår i Microsofts .NET-koncept. Utöver Visual Basic och Java Script som ingick i den första generationen finns i den andra också C++ och C#.

2.4 MySQL

Användning av ett databssystem är ofrånkomligt i de flesta fall när data behöver lagras för att eventuellt presenteras i ett senare skede. Databssystem gör det möjligt att spara stora mängder data väl organiserat i en databas. Med tabeller som hjälpmedel kan enormt mycket information struktureras och göras lättillgängligt.

Det kostnadsfria databasbehandlingssystemet MySQL innehåller både databserver och utvecklingsmiljö.

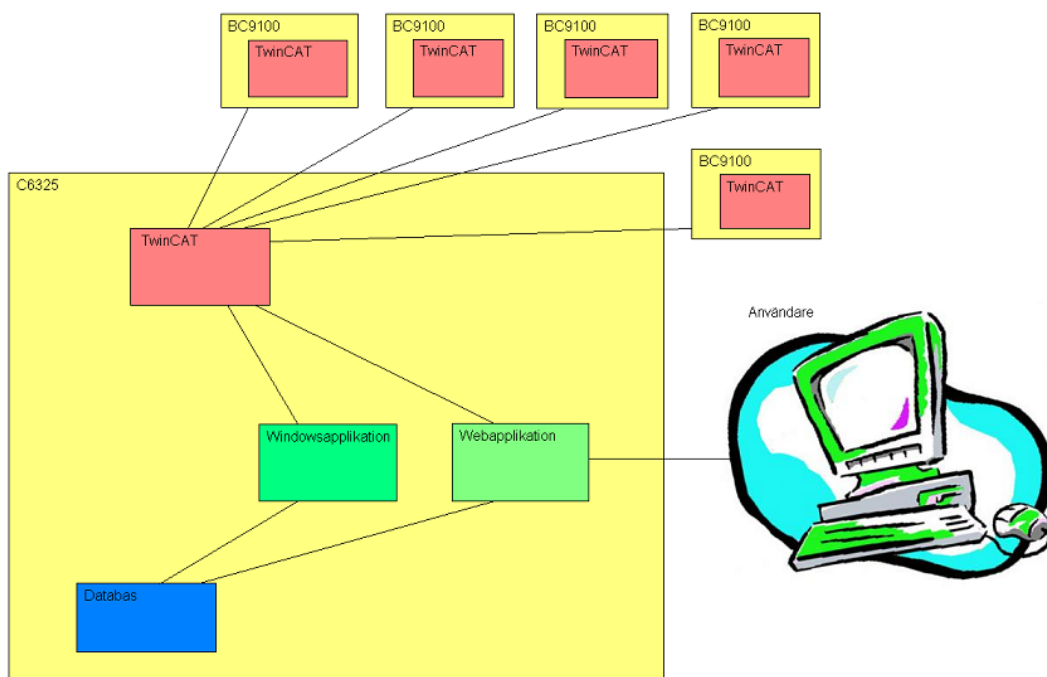
För att manövrera databasen används språket SQL (Structured Query Language) genom vilket man enkelt kan lägga till, ta bort eller ändra information.

För en genomgång av skapandet av en databas i MySQL, med tabell och insättning av värden, se Appendix 1.

3 Utförande

Systemet har till största delen implementerats med centraliserad intelligens. Noder bestående av BC9100 samlar in data och skickar vidare till en central bestående av en kraftfullare dator, C6325, där information utvärderas och beslut om åtgärder tas. Noderna utför som regel inte mycket utan klartecken från centralen.

Web- och databasserver finns installerade på den centrala enheten för lagring av data och visualisering via web.



Figur 3.1: Principiell skiss över systemet.

3.1 PLC-projekt

Den centrala enheten C6325 liksom noderna kör alla sitt eget PLC-projekt, vilket betyder att noderna kan få möjlighet att ta egna beslut i speciella situationer, men även vid normal drift används nodernas PLC-program till en rad uppgifter, bland annat att kontrollera att systemet inte slutat att fungera.

3.1.1 IntHouse

I PLC-projektet IntHouse, som körs på enheten C6325, finns det centrala PLC-projektet i vilket större delen av systemets intelligens koncentreras. Projektet är uppdelat i sex olika program vilka var och ett utför operationer som rör respektive av de fem noder som systemet innehåller, samt ett program (Main) som samordnar de övriga och som utför mer allmänna operationer.

3.1.1.1 BC1

I programmet BC1, i projektet IntHouse, utförs operationer rörande den första noden i systemet, som är placerad i kontorslokalens konferensrum. Här styrs ljuset manuellt med trådlösa EnOcean-givare vars signaler skickas vidare från EnOcean-mastern till DALI-mastern, som i sin tur skickar vidare till sina DALI-ballasts varefter ljusnivån sätts till önskad nivå.

I konferensrummet finns dessutom en rörelsesensor som sätter en digital logisk signal vid rörelse. I programmet finns en timer som dimmar ner ljuset efter en viss tid utan rörelse i rummet, och släcker det helt efter ännu en tid.

3.1.1.2 BC2 till BC5

Eftersom fyra av noderna sitter i kontorsrum med till stor del liknande funktioner skiljer sig inte heller programmen rörande dem särskilt mycket från varandra. Gemensamt är ljussättningen som styrs från tre håll – från web, automatiskt vid inloggning med fingeravtryck, eller manuellt. Dessutom ser programmeringen av larmet likadan ut.

Enheter kopplade till varje nod gör dock att det finns en viss skillnad. Till den andra noden har fingeravtrycksläsaren kopplats. Detta betyder att kommunikationen med den tas om hand i motsvarande program (BC2).

Den tredje noden läser in signalen från en larmslinga som utvärderas i BC3. Programmen som behandlar noderna tre och fyra skiljer sig inte mer än i antal rum och lampor.

3.1.2 BC9100

På varje BC9100 finns ett PLC-program som har till uppgift att behandla en del av den data som kommer in ifrån terminalerna, vilket gör att systemet inte är helt centraliserat. Det går att styra vilken data som BC:n kommer åt och kan köra i PLC-programmet. Mer om det i avsnitt 3.1.2 Konfiguration i System Manager.

Anledningen till att det är önskvärt att behandla viss information direkt i noden är att vissa funktioner, som larm och manuell ljusstyrning, måste vara brukbara även vid kommunikationsfel mellan nod och central.

Således samlas denna datamängd in och görs tillgänglig i BC:ns PLC-program. Sedan behöver nödvändigtvis inte något mer göras vid normal drift än att skicka ut data på fältbussen och vidare upp till respektive program i centralen (IntHouse), men vid en systemnedgång kan BC:n ta hand om all utvärdering.

Det finns ett enkelt och bra knep för att upptäcka ett systemfel. Under drift skickas en bit kontinuerligt mellan nod och central. Vid varje mottagande ändras biten och skickas tillbaka till avsändaren, vilket gör att noden upptäcker om biten inte ändrats på ett visst antal PLC-cykler, och konstaterar då ett systemfel.

3.1.3 Konfiguration i System Manager

Konfigurationen av systemet görs i System Manager, vanligtvis med en extern dator som kopplas in på samma nätverk som systemet. Möjlighet finns i detta fall att göra konfigureringen direkt på den centrala enheten, eftersom den har tillräcklig kapacitet samt en komplett version av TwinCAT installerad. Dock vill man för det mesta hålla systemet så "rent" som möjligt och väljer då att koppla in sig med en bärbar enhet vid installation och underhåll.

Efter uppkopplingen av hårdvara och en genomsökning av nätverket finns möjlighet att välja målenhet för konfigureringen. När den centrala enheten valts scannas PLC-projektet och in- och utvariabler blir synliga och därmed manövrerbara.

Nästa steg är att scanna fältbussen, varefter varje BC9100 med sina respektive terminaler blir synlig. Varje terminals in- och utsignaler samt även BC:ns speciella minnesarea för fältbussvariabler blir åtkomliga.

Nu finns möjlighet att välja varje terminals åtkomlighet från tillhörande BC. För varje fysisk terminal finns nämligen två virtuella terminaler i System Manager, som väljs efter hur man beslutar att organisera sitt system. Valen finns att antingen låta BC:n fungera som en ointelligent fältbussnod för terminalen, eller att låta BC:n komma åt terminal-variablerna.

I det senare fallet görs slutligen en variabelexport till varje BC som då får tillgång till sina exakta minnesadresser. Utvalda terminaler uteblir och blir därför oåtkomliga för respektive BC, som fungerar för dessa som en ointelligent fältbussnod.

Istället nås terminalerna i fråga från den centrala styrenheten efter att variabelänknings gjorts.

För en noggrannare genomgång av ovan nämnda, se Appendix 3.

3.1.4 Lösningar

Följande avsnitt beskriver hur några utvalda delar av implementeringen lösts. Programmeringen utförs uteslutande i Structured Text (ST), ett av de fem möjliga språken i IEC-61131-1.

3.1.4.1 Styrning av lampor

Lampor i alla rum förutom konferensrummet styrs genom reläer, det vill säga terminalerna KL2602. Styrning kan ske på tre olika sätt. Om en person loggar in med fingeravtryck ska utvalda lampor tändas, och samtidigt ska styrning kunna ske från både web och manuellt. Detta är inte helt okomplicerat, och lösningen som används baseras på en statusvariabel som anger vilket av de tre alternativen som för tillfället är aktivt.

Funktionsblocken R_TRIG och F_TRIG, som bevakar en signal och ger utslag vid uppgång respektive nedgång på denna, används för att lyssna efter vilket tänd/släck-alternativ som är aktivt, och statusvariabeln sätts därefter och innehåller således den information som behövs för att göra en korrekt styrning.

Vid till exempel en händelse på en signal från web-ljusstyrning slår statusvariabeln om till motsvarande värde och startar motsvarande del av programmet.

Utöver detta behöver aktuell lampstatus kontrolleras kontinuerligt, och i sin tur styra styr-signalerna från ”andra hållet”, för att signalerna ska ha korrekt status. Om exempelvis en lampa tänds från web, vilket innebär att styrsignalen från web blir TRUE, för att sedan släckas manuellt, måste styrsignalen från web ändras till FALSE för att det ska vara möjligt att tända igen. Samtidigt krävs då att triggern som lyssnar på web-signalen inte ger utslag på sådana operationer, eftersom det skulle göra att statusvariabeln ändras.

Det senare har lösts enligt följande: I den PLC-cykel, då något av de tre styralternativen blir aktivt, inaktiveras de triggers som bevakar de två andra styralternativen. Således behåller statusvariabeln sitt korrekta värde. En prioritetsordning åstadkoms implicit genom ordningen i vilken avläsning sker.

3.1.4.2 Styrning med EnOcean

Ljusstyrningen i konferensrummet sker med en EnOcean-knappsats, i detta fall med 4 knappar. Vid knapptryck skickas en statusbyte, unik för varje knapp, till mottagaren. Eftersom detta sker både när knappen trycks in och när den släpps kommer systemet att veta när ljusdämpning ska ske.

I konferensrummet finns fyra lampor som kan styras var och en för sig, alla tillsammans, eller i olika grupper. I nuläget, när ambitionen endast är att visa vad som kan åstadkommas, är ljussättning aningen osmidigt, men många scenarion kan åstadkommas med dessa fyra knappar.

Programmeringen är utförd så att inställningar av hur ljussättning ska ske görs med knappsatsen. Första knapptrycket anger vilken typ av styrning som ska ske – en lampa, alla lampor, eller en grupp. En statusvariabel sätts i PLC-programmet för att hålla reda på valen. Följande knapptryck anger hur många lampor och vilka lampor, beroende på föregående val.

Slutligen styrs de valda lamporna med två av knapparna. För att byta scenario används den fjärde knappen, som vid tryck nollställer valen och således inte kan användas till något annat.

Vid en installation av systemet hos slutkund finns möjligheten att använda en knappsats med fler knappar samt att ha förinställda scenarion, för bästa användarvänlighet.

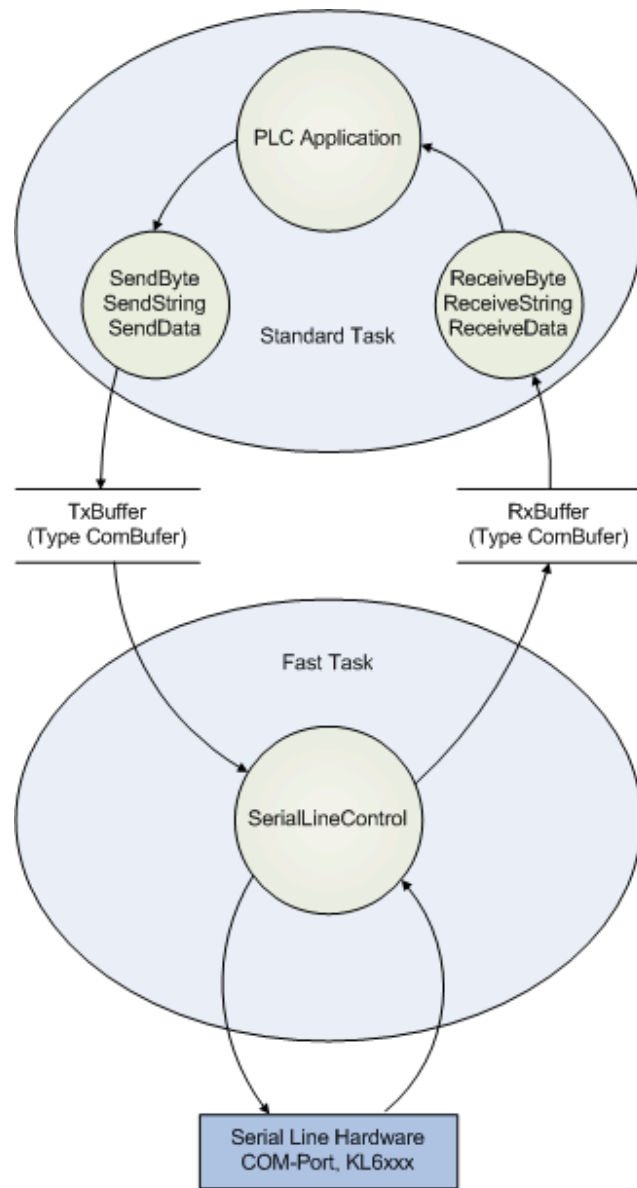
3.1.4.3 Access-kontroll med fingeravtryck

Fingeravtrycksläsaren kontrollerar att information på ett kort överrensstämmer med information från ett finger, och om så är fallet skickas ett förprogrammerat nummer från enheten.

Kommunikation med systemet sker över RS 232 och för detta behövs biblioteket COMlib vilket innehåller funktionsblock och datastrukturer för seriekommunikation.

En konfiguration av seriekommunikationsterminalen KL6001 är nödvändig för att ställa in korrekta parametrar. Detta sker med hjälp av mjukvaruverktyget KS2000.

När slutligen kommunikationen är igång inkommer data till terminalen för att sedan transporteras vidare till PLC:n. Vid tät kommunikation skulle PLC:n behöva arbeta mycket snabbt för att kunna ta hand om all data, och därför används buffertar mellan terminal och PLC. Dataöverföringen mellan terminal och buffert kan vid behov behandlas i en separat ”task” med snabbare cykel, något som inte är nödvändigt vid den låga belastning som finns i nuvarande system.



Figur 3.2: Kommunikationsprincip vid seriekommunikation.

3.2 Databas

För att lagring av data såsom energiförbrukning, inloggningsuppgifter etcetera skall vara möjlig behövs en databas. MySQL är ett kostnadsfritt databassystem vars kapacitet är fullt tillräcklig för de flesta mindre applikationer såsom denna.

Databasservern installeras på den centrala enheten varefter en databas med lämpliga tabeller skapas manuellt. Fyra tabeller är för närvarande aktuella i projektet:

- AnvOchLosen. Denna är huvudsakligen statisk och innehåller användare, det vill säga personal, med lösenord för att kunna logga in på webben. Information om användare matas in manuellt av administratören.
- Room. Statisk tabell som innehåller fastighetens olika rum, dessutom med tilldelade nummer för enklare hantering.
- Temp. Denna tabell innehåller temperatur i olika rum vid olika tidpunkter. Den fylls på varje gång det sker en temperaturändring i lokalen. I nuläget finns temperaturgivare endast i ett rum, varför endast data om detta rum står att finna i tabellen.
- AccessLog. Innehåller information om vilken användare som loggat in på webben, samt vid vilken tidpunkt.

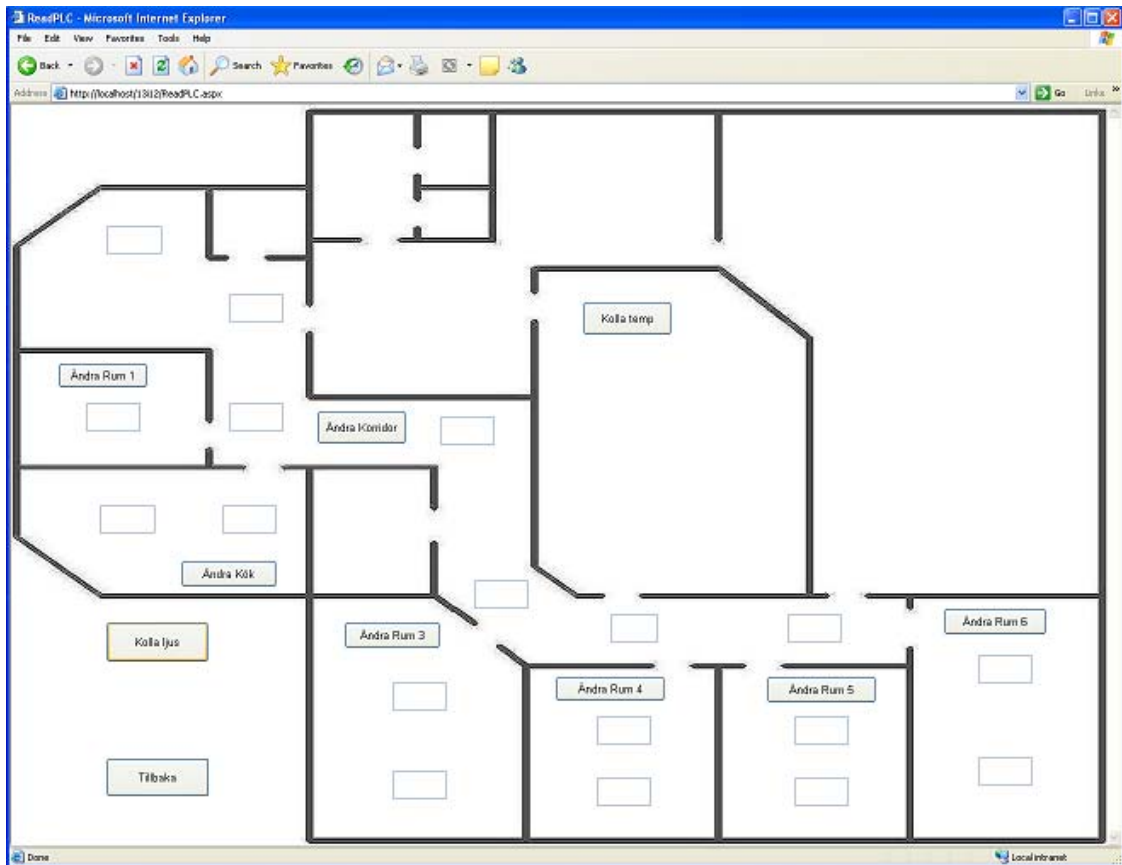
En liten genomgång av databashantering finns i Appendix 1.

3.3 Websidor

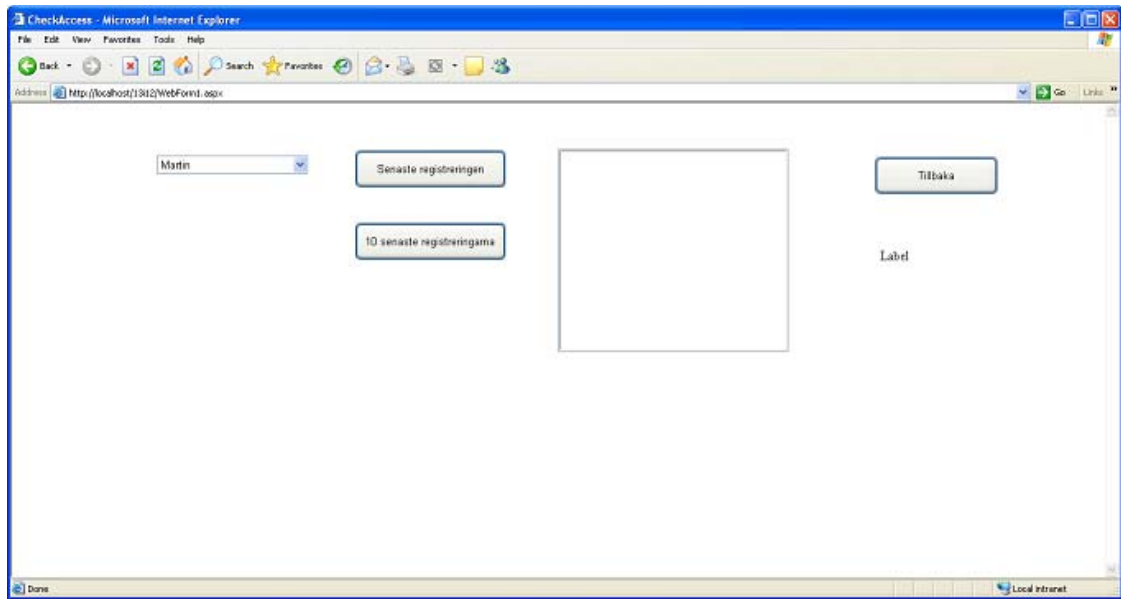
Visualisering och presentation med websidor möjliggör, för den med tillgång till nätverket, att ta del av och påverka statusen på funktioner i fastigheten. Med utgångspunkt i en startsida finns möjlighet för användare att klicka sig vidare för att ta del av diverse uppgifter.

Inloggningssidan är central – utan att först logga in finns ingen möjlighet till annat än att se aktuell status på ljussättning i fastigheten. För att hålla reda på inloggade användare används så kallade Sessions, vilka gör att information kan skickas mellan olika sidor.

För att skifta mellan websidor används Server.Transfer(”ny sida”).



Figur 3.3: Websida för ljusstyrning och statuskoll på kontoret.



Figur 3.4: Websida för kontroll av inloggade användare.

3.3.1 IIS – Internet Information System

För att distribuera websidorna används Microsofts webserver IIS, som medföljer Windows XP Professional. Denna installeras på den centrala enheten och sidor som placeras i enhetens rotkatalog blir då tillgängliga om korrekt sökväg i förslagsvis Internet Explorer anges.

3.3.2 Tillverkning av websidor i Visual Studio

Visual Studio erbjuder möjligheten till visuellt skapande av websidor. Vid val av nytt projekt finns möjlighet att välja programmeringsspråk, i detta fall Visual C#, samt applikation, ASP .NET Web Application. Automatiskt skapas då programskelett och html-kod. Önskvärda objekt på den aktuella websidan såsom textrutor och knappar adderas grafiskt.

Skapade websidor läggs automatiskt i rotkatalogen på utvecklingsdatorn, sedan är det trivialt att flytta dessa till måldatorm, även om vissa modifikationer, som till exempel att sökvägar är desamma, är nödvändiga.

3.4 Integration

Centralt i projektet har varit att få kopplingarna mellan de olika delarna av systemet att fungera. Data som inkommer till PLC-programmet ska kontinuerligt lagras i databasen, och data ska transporteras mellan databas och web. Dessutom ska data skickas direkt mellan web och PLC-program.

3.4.1 TwinCAT.Ads DLL

TwinCAT.Ads DLL¹ är ett klassbibliotek i .NET som låter klasser kommunicera med en Ads-enhet. Efter installation hittar C#-programmet biblioteket genom att en referens till dll-filen läggs till.

I biblioteket finns ett antal klasser som används i projektet:

- TcAdsClient – ger tillgång till en Ads-enhet, med metoden Connect. Innehåller bland annat metoden AddDeviceNotification som kan användas för att bevaka en PLC-variabel.
- AdsBinaryWriter/Reader – skriver till/läser PLC-datatypeer binärt.
- AdsNotificationEventArgs – tillhandahåller data till händelser i TcAdsClient
- AdsStream – Klass som sparar läst data.

¹ DLL – Dynamically Linked Library. En eller flera programfunktioner som kan delas mellan olika program för att tjäna ett visst syfte, exempelvis för att spara diskutrymme om flera program behöver samma resurser.

3.4.2 Connector/NET

För möjliggöra kopplingen mellan web- och Windowsapplikationer och databas installeras Connector/NET. Efter att referensen MySql.Data lagts till görs bland annat följande användbara klasser tillgängliga:

- MySqlConnection – etablerar en kontakt med MySql.
- MySqlCommand – representerar ett SQL-kommando till databasen.
- MySqlDataReader – läser hämtad information från databasen.

3.4.3 TwinCAT/MySQL

Data ska kontinuerligt skrivas till en databas varför denna koppling är nödvändig. Lösningen bygger på en windows-applikation som programmeras i Visual Studio och installeras på den centrala enheten.

Efter installation startas Windowsapplikationen och exekveras sedan konstant.

Kopplingen mellan MySQL och ASP .NET görs möjlig med Connector/NET som är en ADO.NET Driver för MySQL.

Efter installation av Connector .NET och tillagd referens till MySql.Data.dll fås tillgång till en rad klasser som används till kopplingen.

Figur 3.5 visar ett programutdrag som låter information läsas händelsestyrt från ett PLC-program till en Windowsapplikation med hjälp av TcAdsClient. I detta fall innehåller PLC-variabeln information om användare som loggar in med fingeravtryck. Figur 3.6 visar hur information om inloggade användare sedan skickas vidare till databasen.

```

private TcAdsClient myTcAdsClient;
myTcAdsClient = new TcAdsClient();

private AdsStream myAdsStream;
myAdsStream = new AdsStream(2);

private BinaryReader myBinaryReader;
myBinaryReader = new BinaryReader(myAdsStream, System.Text.Encoding.ASCII);

// Skapar en kontakt med TwinCAT på lokal enhet
myTcAdsClient.Connect(801);

private int hConnect;

/* PLC-variabeln som läses är den globala LastReceived.
Datan (offset 0, längd 2) hamnar i myAdsStream.
AdsTransmode.OnChange innebär att reaktion ska ske vid ändring av LastReceived.
Kontroll av ändring sker med 10 ms intervall, ingen försening tillåts
*/
hConnect = myTcAdsClient.AddDeviceNotification(".LastReceived",myAdsStream,0,2,
AdsTransMode.OnChange,10,0);

myTcAdsClient.AdsNotification += new AdsNotificationEventHandler(OnNotification);

private void OnNotification(object sender, AdsNotificationEventArgs e)
{
    DateTime time = DateTime.FromFileTime(e.TimeStamp);
    e.DataStream.Position = 0;
    int personId;
    personId = binRead.ReadByte();
}

```

Figur 3.5: Händelsestyrd läsning av PLC-variabel från Windows- eller webapplikation.

```

private MySql.Data.MySqlClient.MySqlConnection myConnection;
myConnection = new MySql.Data.MySqlClient.MySqlConnection();

private MySql.Data.MySqlClient.MySqlCommand myCommand;
myCommand = new MySql.Data.MySqlClient.MySqlCommand();

myConnection.ConnectionString = "server=127.0.0.1;uid=root;" + "pwd=54321;database=house;";

string person="";
if (personId==10)
    person="Kenny";

myConnection.Open();
myCommand.Connection = myConnection;
/* Sätter in person och tid i tabellen accesslog. Frågetecknen gör det
möjligt att använda variabler. */
myCommand.CommandText = "INSERT INTO accesslog VALUES (?loggedPerson,?time)";
myCommand.Prepare();
myCommand.Parameters.Add("?loggedPerson",person);
myCommand.Parameters.Add("?dt", time);
myCommand.ExecuteNonQuery();
myConnection.Close();

```

Figur 3.6: Skrivning från Windowsapplikation alternativt Webapplikation till databas.

3.4.4 TwinCAT/web

Vid direktiv från en användare på webben om ljussättning, eller begäran om statusinformation, behöver information transporteras i realtid mellan webb och PLC-program.

Nedan visas i ett utdrag från koden hur detta kan ske med hjälp av TcAdsClient.

```
private int intReceive;
private TcAdsClient myTcAdsClient;

myTcAdsClient = new TcAdsClient();
myTcAdsClient.Connect(801);

intReceive = myTcAdsClient.CreateVariableHandle("MAIN.plcVarLamp1");

AdsStream myAdsStream = new AdsStream(2);

// Läser data och lägger i myAdsStream
tcAdsClient.Read(intReceive,myAdsStream);

BinaryReader binRead = new BinaryReader(myAdsStream);

/* Färgsätter boxar (som symboliserar lampor) efter vilken
   status respektive lampa har. */
if (binRead.ReadBoolean()==false)
    TextBox1.BackColor=System.Drawing.Color.Black;
else
    TextBox1.BackColor=System.Drawing.Color.GreenYellow;
```

Figur 3.7: Läsning av status på PLC-variabeln plcVarLamp1.

```
private int intSend;
private TcAdsClient myTcAdsClient;

myTcAdsClient = new TcAdsClient();
myTcAdsClient.Connect(801);

intSend = myTcAdsClient.CreateVariableHandle(".boolLightLamp1FromWeb");

AdsStream myAdsStream = new AdsStream(1);
BinaryWriter binWrite = new BinaryWriter(myAdsStream);

binWrite.Write(bool.Parse("true"));
myTcAdsClient.Write(intSend,myAdsStream);
```

Figur 3.8: Skrivning till PLC-variabeln boolLightLamp1FromWeb.

3.4.5 MySQL/web

Denna koppling görs exempelvis när personer försöker logga in via web. Principen är densamma som koden i Figur 3.6.

4 Diskussion

4.1 Problem

Även om projektet har kunnat slutföras tillfredsställande så har det självklart funnits motgångar på vägen mot slutmålet. Bekymmer som kraftigt försenat arbetet är ett datorhaveri med datorbyte som följd, en felaktig uppgift om tillgängliga gränssnitt för hårdvara, samt ytterligare ett datorbyte.

Det andra fallet gäller fingeravtrycksläsaren från Precise Biometrics. Enligt manual ska kommunikationen från enheten kunna skickas med seriell kommunikation RS 485, men detta visade sig efter många turer vara en felaktig uppgift. Gränssnittet RS 232 fungerar dock, med kortare räckvidd som skillnad, och är därför det som för närvarande används.

Det andra datorbytet gällde överflyttning från utvecklingsdator till den centrala datorn i systemet och var nödvändigt eftersom de inledande testerna av integrering gjordes internt på utvecklingsdatorn. Bytet gick ändå relativt smidigt jämfört med vad som förväntades.

Ett problem som uppstått två gånger under arbetets gång är att klockan i TwinCAT går för långsamt. Detta orsakade dels felaktiga tidsangivelser vid olika loggningar till databasen, dels att ljusdämpning av lampor med DALI gick mycket sakta. Orsaken till problemet är oklar, men misstankar finns om att det har att göra med DELL Latitude, och problemet löstes automatiskt vid byte av central dator.

Ett fortfarande olöst problem är att temperaturgivaren (EnOcean) ger fel temperatur. Empiriska studier visar att den presenterade temperaturen är omvänt proportionell mot den verkliga, det vill säga att den sjunker i samma takt som den verkliga stiger och tvärtom. Möjligheten finns att problemet löses med en ny temperaturgivare, något som inte kunnat testas under tiden för arbetet.

4.2 Personliga lärdomar

I många uppkomna situationer vid implementeringen har dator- och nätverkskommunikation varit centrala delar. En del har varit nytt för mig, och även om viss förkunskap har funnits är det kanske inom detta område jag lärt mig mest under projektperioden.

Vidare har jag fått en introduktion i webbkonstruktion och databashantering, båda för mig helt nya områden.

PLC-programmering fanns sedan tidigare bland mina kunskaper och har givetvis utvecklats ytterligare. Structured Text har jag inte tidigare använt mig av, även om skillnaden inte är särskilt stor jämfört med högnivåspråk som Java och C.

Slutligen har jag fått en insyn i automationsbranschen, med lärdomar om branschspecifika termer, aktuell marknad och framtid, samt skaffat mig lite kontakter som alltsammans kan vara nyttigt i en framtida karriär.

4.3 Vidareutveckling av systemet

Grunderna är nu lagda och ett fungerande system finns. Tillägg av funktioner som videoövervakning, energimätning, SMS-utskick vid larm, etcetera bör kunna ske utan större komplikationer.

Värmestyrning, som är en önskvärd funktion, kräver förmodligen en del mer arbete att få att fungera tillfredsställande, eftersom reglerfunktioner för hur innetemperatur påverkas dels av yttertemperatur, dels av vindförhållanden bör tas fram.

Enligt projektplanen ska resultatet vara ett automationssystem att demonstrera för eventuella kunder, med främsta syfte att visa vad som kan åstadkommas med Beckhoffs produkter. Nästa steg i företagets arbete att närma sig kunder bör vara att ta fram verktyg för enkel installation och programmering av systemet, det vill säga att skapa ett överliggande system där programmeringen sker i en användarvänlig miljö och utan krav på speciell utbildning. Detta är vad som efterfrågas på marknaden och vad som i viss mån också erbjuds.

Således torde underlag för ytterligare projektarbeten inom området finnas på företaget.

5 Referenser

Beckhoff Information System

Innehåller bland annat teknisk information och manualer för både mjuk- och hårdvara.

MySQL 5.0 Reference Manual, <http://dev.mysql.com/doc/#refman>

Innehåller manualer och exempel av databashantering med MySQL.

Visual Studio .NET 2003 Documentation

Innehåller manualer och dokumentation av utvecklingsverktyget.

BioAccess 200 Installation Manual, BioAccess 200 User Manual

Innehåller manualer för biometrics fingeravtrycksläsare med tillbehör.

Nationalencyklopedin

<http://www.ne.se>

Har främst använts för att reda ut begrepp inom webbredigeringen.

Appendix

A1 – Databashantering

Operationer i MySQL utförs i språket SQL.

Nedan följer en steg-för-steg-guide över hur en databas innehållande en tabell med personer och deras födelsedagar skapas. (*Referens: MySQL 5.0 Reference Manual*)

1. CREATE DATABASE temp; // skapar en databas med namn temp
2. USE temp // gör databasen temp aktiv
3. CREATE TABLE fodelsedag (namn VARCHAR(20), datum date);
// skapar en tabell fodelsedag med två kolumner, den första med plats för namn och den andra med plats för datum
4. INSERT INTO fodelsedag VALUES ('Conny Carlsson', 1899-04-01);
// fyller på tabellen med data

Att sedan hämta information från databasen är lika enkelt, till exempel:

```
SELECT datum FROM fodelsedag WHERE namn='Conny Carlsson';  
// returnerar Conny Carlssons födelsedag
```

```
SELECT namn FROM fodelsedag WHERE datum='1987-02-02';  
// returnerar de vars födelsdag är det angivna datumet
```

Ofta vill man sortera den data man hämtar:

```
SELECT namn FROM fodelsedag ORDER BY datum DESC;  
// returnerar alla personer i tabellen och sorterar dem efter födelsedag, yngst först.
```

```
SELECT namn FROM fodelsedag ORDER BY datum ASC LIMIT 0,5;  
// returnerar de 5 äldsta personerna
```

A2 – Adressering av variabler

Variabler som ska ha knytningar utanför ett PLC-projekt behöver på något vis deklarerats på en minnesarea. Beroende på situationen finns fördelar både med att deklarerera en variabel i PLC-projektet på en exakt adress och att inte göra det.

Har till exempel redan mappningar gjorts i System Manager och variabelinformation importerats till PLC-projektet, så är det enkelt att knyta variabler till exakta platser på minnesarean. Detta görs enligt följande:

```
PlcInBool AT %IX128.0 : BOOL;
```

```
// Läger variabeln PlcInBool på byte 128, första biten. I står för invariabel och X för bit.
```

Variationer kan göras beroende på variabelns storlek och karaktär:

```
PlcOutByte AT %QB128 : BYTE;
```

```
// Läger variabeln PlcOutByte, som är en byte stor, på byte 128. Q betyder utvariabel och B byte.
```

Finns inte en önskan att lägga en variabel på en exakt minnesplats, till exempel om en knytning sedan ska göras i System Manager, fungerar följande:

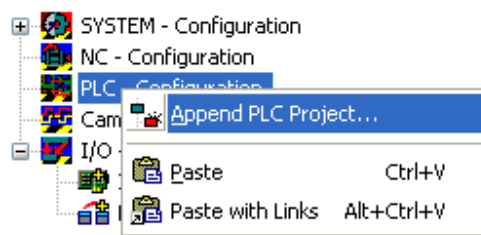
```
PlcInBool AT %I* : BOOL;
```

```
// Användaren behöver inte veta den exakta adressen utan gör en knytning efter att variabeln dykt upp under det hämtade PLC-projektet i System Manager.
```

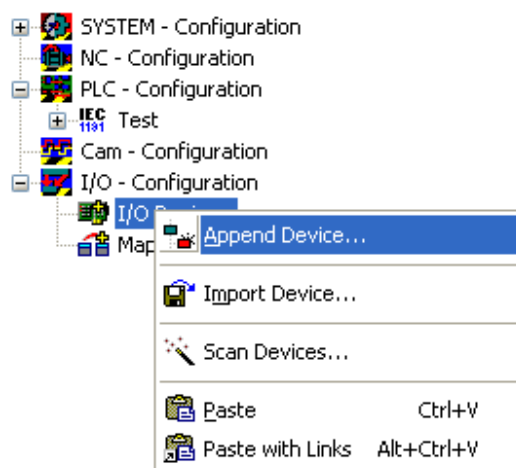

A3 – Konfiguration i System Manager

Efter att programmering är gjord och ett system kopplats upp kvarstår konfigurationen. I Appendix 3 beskrivs denna för ett enkelt system innehållande en bärbar dator (som både används för konfiguration och som central i systemet) och en nod i form av en BC9100 med ett antal påkopplade terminaler.

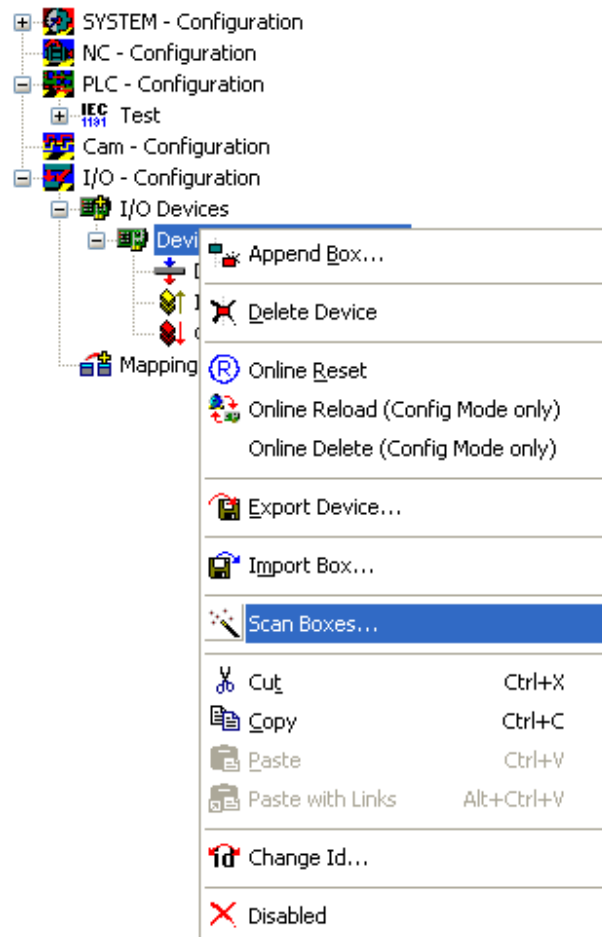
1. Öppna TwinCAT System Manager.
2. Välj Actions/Choose Target System. Den lokala enheten är förvald, men om utveckling sker på en temporär enhet finns här möjlighet att välja den tilltänkta målenheten.
3. Högerklicka på PLC – Configuration och välj Append PLC project. Välj sedan det PLC-projekt som är tänkt att ligga på den enhet som valdes till ”Target System”. (Det måste finnas ett kompilerat, felfritt projekt.)



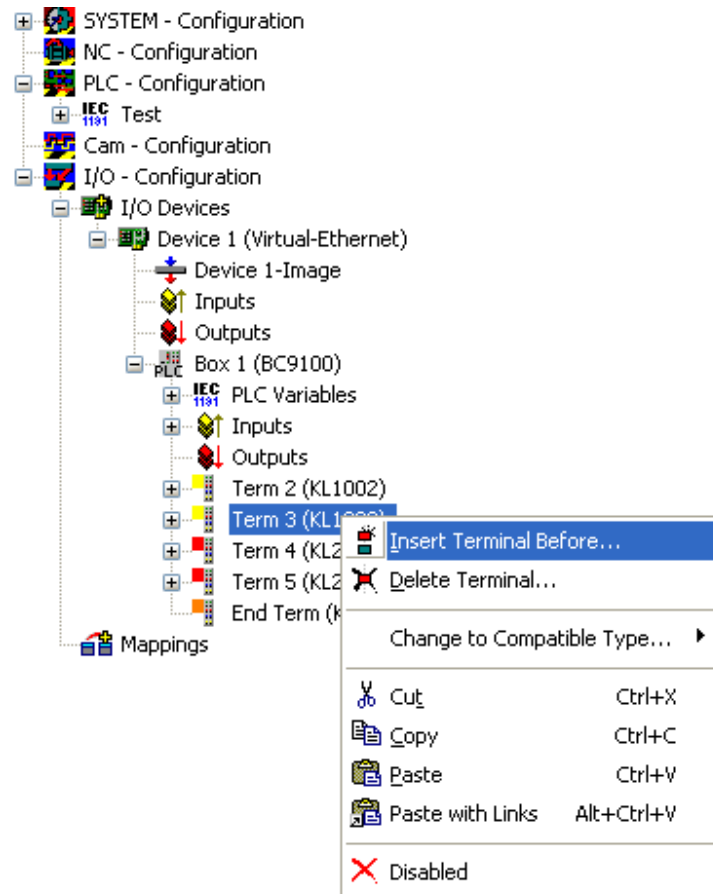
4. Högerklicka på I/O Devices och välj Append Device. Välj sedan Virtual Ethernet Interface.



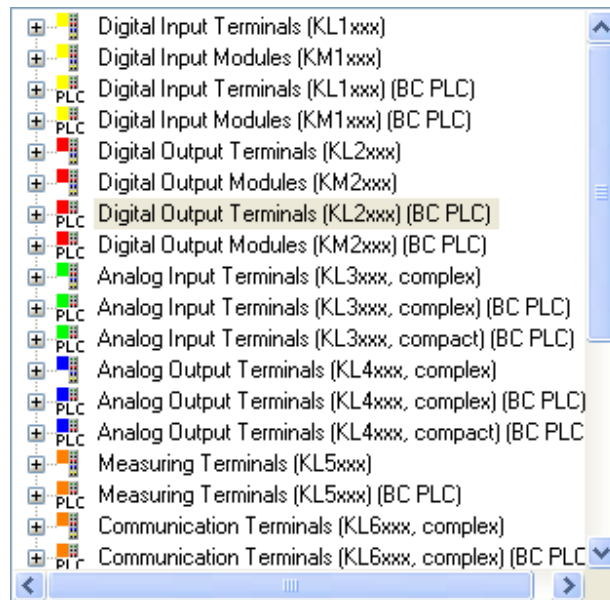
5. Högerklicka på den tillagda Ethernet-enheten och välj Scan Boxes. Nu hittas de noder som är fysiskt nåbara över Ethernet-nätverket, med tillhörande terminaler. I exemplet används en BC9100 med fyra terminaler, två digitala ingångsterminaler och två digitala utgångsterminaler.



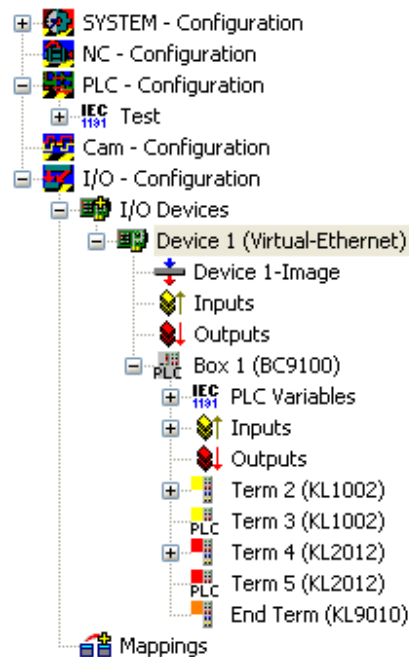
6. Förvalt är att terminalerna är näbara från System Manager, det vill säga att det går att komma åt variabler i dem genom System Manager. Vill man istället komma åt variablerna i BC:ns PLC-program får man göra följande ändring: Byt ut terminalen i fråga mot en terminal markerad PLC, och ta därefter bort den gamla.



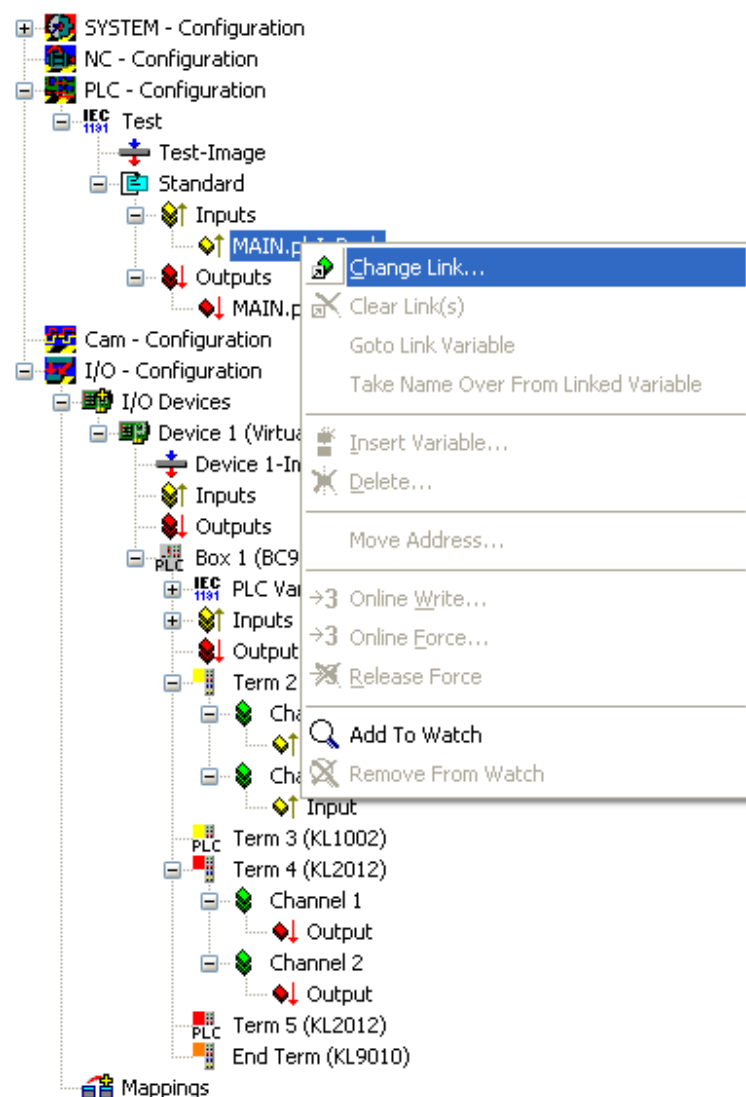
I rutan som dyker upp finns att välja mellan alla terminaler i Beckhoffs sortiment, och varje terminal finns i två versioner, som PLC-terminal för BC, eller som en terminal nåbar från System Manager.

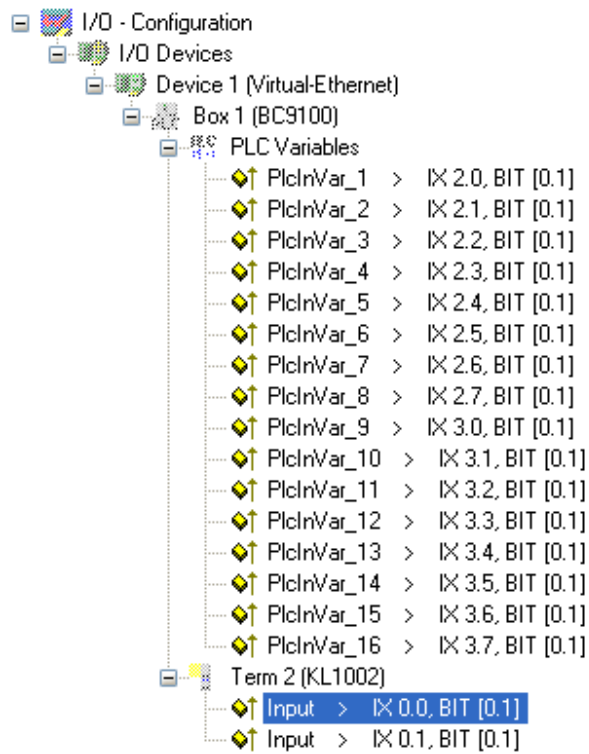


Resultatet:



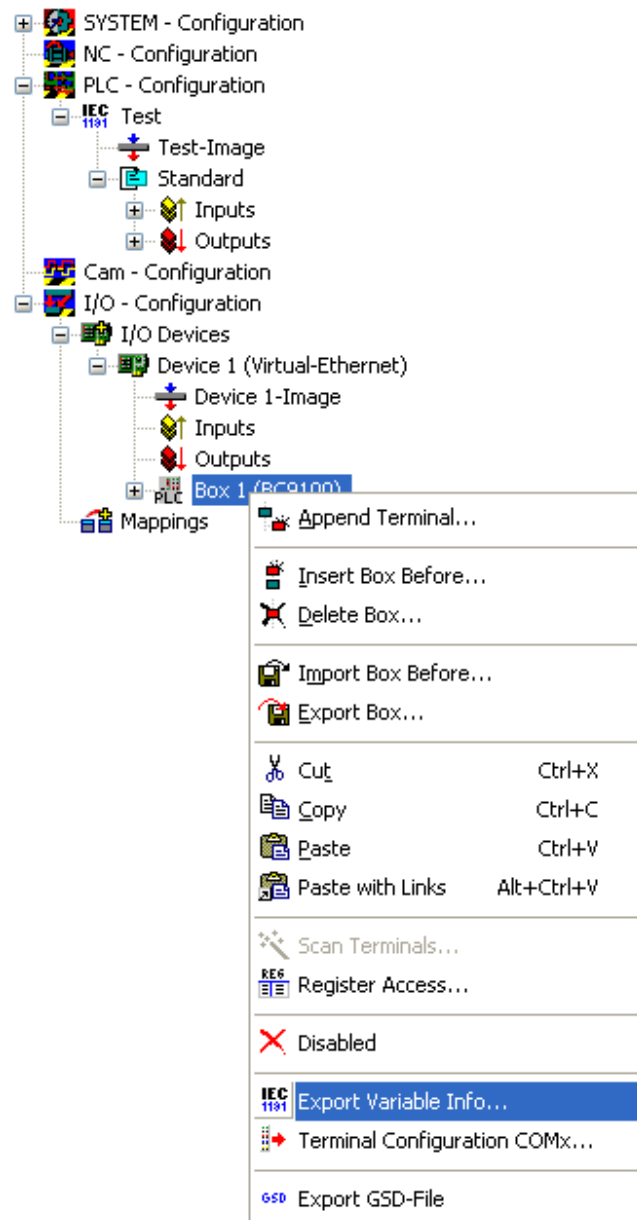
7. I detta skede görs variabelknytningar mellan PLC-projektet och variabler på terminalerna. Högerklicka på en in- eller utvariabel från PLC-projektet och välj Change Link. En ruta med matchande variabler på nätverket dyker upp. Välj för att lägga till.





Vill man istället knyta PLC-variabeln till en fältbussvariabel på BC:n väljer man som i första fallet Change Link, men tar sedan någon av de variabler som mappas mot fältbussen (PlcInVar_1...).

För att sedan få tillgång till variablerna i BC:ns PLC behöver en variabelexport göras. Högerklicka på den enhet, BC:n i detta fall, vars variabelinformation ska exporteras, och välj Export Variable Info. På BC:ns PLC-projekt importeras sedan denna information.



A4 – Förkortningar

ASP - Active Server Pages

CNC – Computer Numerical Control

CPU – Central Processing Unit

DLL – Dynamically Linked Library

HMI – Human-Machine Interface

I/O – Input/Output

IEC – International Electrotechnical Commission

IIS – Internet Information Services

NC – Numerical Control

OCX – OLE Control Extension

OLE – Object Linking and Embedding

OPC – OLE for Process Control

PLC – Programmable Logical Controller

RS – Recommended Standard

SQL - Structured Query Language

TCP/IP – Transmission Control Protocol / Internet Protocol

USB – Universal Serial Bus