

## Fastighetsautomation - Automatisering av ett kontor



---

**Kastriot Gashi**  
**David Ramljak**

Division of Industrial Electrical Engineering and Automation  
Faculty of Engineering, Lund University

# Fastighetsautomation - Automatisering av ett kontor



**LUND**  
**UNIVERSITY**

Kastriot Gashi & David Ramljak

Institutionen för Biomedicinsk Teknik

Avdelningen för Industriell Elektroteknik och Automation(IEA)

LTH Ingenjörshögskolan vid Campus Helsingborg, Lunds Universitet

221 00 Lund, Sverige

© Copyright Kastriot Gashi, David Ramljak

Industrial Electrical Engineering and Automation(IEA)

Faculty of Engineering

Lund University

Box 118, SE-221 00 Lund, Sweden

Industriell Elektroteknik och Automation(IEA)

Lunds Tekniska Högskola

Box 118, 221 00 Lund

# Abstract

Building automation is a growing industry where the property owner wants to automate as much as possible with a centralized control. These possibilities offers a more comfortable and smarter environment for both private users and big companies.

The purpose of this thesis was to automate an office for Beckhoff Automation AB in Malmoe. An analysis of the benefits of using databases, OPC UA, SOA and Beckhoff's embedded PC, CX9020, for the implementation of this assignment was also made. Service-oriented architecture have also been analyzed and discussed. A Human- Machine Interface(HMI) with the SCADA-program Indusoft was also created and it will be used in a touchscreen from Beckhoff. The parameters that we controlled was the light in the room and air quality. Data from the sensors were stored in a database where the users of the room can overview history in graphs and tables.

Keywords: DALI, OPC UA, Embedded PC(CX9020), HMI, Touchpad, Indusoft, TwinCat3, Beckhoff, Database, MySQL, SCADA, Sensor, Service-oriented architecture(SOA).

# Sammanfattning

Fastighetsautomation är en snabbt växande bransch där fastighetsägaren vill automatisera så mycket som möjligt med en centraliserad styrning. Dessa möjligheter ger en bekvämare och smartare miljö för både privat bruk och i stora företag.

Syftet med examensarbetet var att automatisera ett kontor för Beckhoff Automation AB i Malmö. En analys av fördelarna med att använda databaser, OPC UA, SOA och Beckhoffs Embedded PC, CX9020, i implementationen av uppgiften gjordes också. Tjänsteorienterad arkitektur(SOA) har också analyserats och diskuterats. Ett eget Human- Machine Interface(HMI) skapades med hjälp av SCADA- programmet Indusoft för att användas i en pekskärm för styrning av rummet. Faktorerna som skulle styras/användas var ljuset och luftkvaliteten. Data från sensorerna har lagrats i en databas där användare har en överblick över de uppmätta signalernas värden bakåt i tiden.

Nyckelord: DALI, OPC UA, Embedded PC(CX9020), HMI, Peksärm, Indusoft, TwinCat3, Beckhoff, Databas, MySQL, SCADA, Sensorer, SOA(Service Oriented Architecture, sv. Tjänsteorienterad arkitektur).

## Innehållsförteckning

Fastighetsautomation - Automatisering av ett kontor.....	1
Sammanfattning .....	4
Förord.....	7
1. Introduktion .....	8
1.1 Företag .....	8
1.1.1 Beckhoff Automation AB .....	8
1.2 Syfte .....	8
1.3 Målformulering .....	9
1.3.1 Användning av DALI-belysning .....	9
1.3.2 Skapa ett HMI.....	9
1.3.3 Analys av luftkvalitet.....	9
1.3.4 Val av kommunikationsprotokoll .....	9
1.3.5 Service-Oriented architecture(SOA).....	9
1.4 Problemformulering.....	10
1.5 Avgränsningar.....	10
1.5.1 Belysning .....	10
1.5.2 Luftkvalitet.....	11
1.5.3 SOA .....	11
2. Teknisk bakgrund .....	12
2.1. SOA och OPC-UA.....	12
2.1.1 OPC UA.....	14
2.2 DALI.....	15
2.3 SCADA .....	17
2.4 CX9020 .....	20
2.5 tSENSE™ .....	21
2.6 Databaser .....	22
3. Metod .....	23
3.1 Källkritik .....	25
4. Analys .....	26

4.1 Programmering och strukturering.....	26
4.1.1 Funktionsblock.....	27
4.2 DALI.....	28
4.2.1 Installering och konfigurering av DALI-armaturer .....	28
4.3.1 Första HMI-prototyp.....	31
4.3.2 HMI.....	32
4.4 Databasen.....	37
4.4.1 Konfigurering av databas.....	37
4.4.2 Programmeringen till databasen .....	38
4.5 OPC UA och SOA .....	40
4.5.1 OPC UA.....	40
4.5.2 SOA .....	41
5. Resultat .....	42
6. Slutsats .....	44
6.1 Ljusstyrningen .....	44
6.2 Mätning av luftkvalitén.....	44
6.3 HMI.....	45
6.4 Sammanfattning .....	45
7. Framtida utvecklingsmöjligheter .....	46
Akronymlista .....	47
Referenser .....	48

# Förord

Till en början vill vi tacka alla som stöttat oss under denna period av hårt arbete:

**Roger Grönvall**, vår handledare på Beckhoff som har visat stort engagemang, bidragit med nya idéer och erbjudit oss hjälp genom hela arbetet.

**Sina Bagheri**, som trots sina arbetsuppgifter på Beckhoff, alltid var tillgänglig för oss när vi hade svårigheter.

Vi vill också passa på att tacka alla anställda på Beckhoff för deras vänlighet och möjligheten att få göra examensarbetet hos dem.

**Mats Lilja**, vår examinator som hjälpte oss att komma igång med arbetet och fanns där för rådgivning när det behövdes.

**Henriette Weibull**, vår handledare på skolan som varit tillgänglig för oss och har visat stort intresse för vårt arbete.

Examensarbete omfattar 22,5 högskolepoäng och utförs som del av Elektroteknik med Automationsteknikprogrammet på LTH, Campus Helsingborg. Vi, Kastriot Gashi och David Ramljak, har haft ett perfekt samarbete där arbetet delades upp lika och varje delmoment gjordes ihop.

David Ramljak & Kastriot Gashi



# 1. Introduktion

Examensarbetet har utförts under perioden januari – maj 2016 på företaget Beckhoff Automation AB i Malmö.

Efter en del avgränsningar blev examensarbetet att utföra en fastighetsautomatisering i mindre skala. I ett av kontoren skulle den vanliga belysning bytas ut mot DALI-belysning och sensorer skulle sättas upp som mäter och lagrar data i en databas som konfigurerades. Allt styrdes via en Beckhoff CX9020 som är en industriell PC med Windows CE inbyggt. För att kontrollera rummet skapades ett HMI som användes i en pekskärm.

## 1.1 Företag

### 1.1.1 Beckhoff Automation AB

Beckhoff Automation AB grundades 1980 i Verl, Tyskland som en del av Elektro Beckhoff GmbH som var ett elektronikföretag grundat av Arnold Beckhoff år 1953. Beckhoff Automation AB är ett medelstort, internationellt verkande företag. Beckhoff Automation AB tillverkar automationsteknik i olika prestandanivåer, allt från enkla komponenter till avancerade system. Företaget fokuserar främst på PC-baserad styrning.

Beckhoff Automation AB har idag 2900 anställda i 34 länder och omsatte 620 miljoner € år 2015[1][2]

## 1.2 Syfte

Syftet med detta examensarbete var att automatisera ett rum. De mätsignaler som användes var ljus, temperatur och luftkvalitet(CO<sub>2</sub> och luftfuktighet). En undersökning om vilka fördelar OPC UA- protokollet i kombination med att använda databaser och PLC-programmering ger genomfördes också. I examensarbetet ingår även en analys av för- och nackdelar med SOA och hur det används inom fastighetsautomation.

## 1.3 Målformulering

I examensarbetet arbetades det med olika delmoment och fält inom fastighetsautomation. I slutändan var målet att knyta ihop allt till ett fungerande system.

### 1.3.1 Användning av DALI-belysning

DALI är ett system för ljusstyrning som erbjuder nya möjligheter och som ger en mycket bekvämare och smartare lösning än med enkel manuell styrning med strömbrytare. DALI-belysning har installerats och programmerats.

### 1.3.2 Skapa ett HMI

Ett eget SCADA-HMI skapades för att användas för kommunikation via OPC UA till CX9020. Därifrån kommer det vara möjligt att se, ändra och skicka värden via en pekskärm från Beckhoff.

### 1.3.3 Analys av luftkvalitet

Hur luftkvaliteten är i ett rum har stor betydelse för hur effektivt hjärnan kan arbeta under en längre tid. Med hjälp av en sensor som mätte temperatur, CO<sub>2</sub>-halt och luftfuktigheten lagrades all data i en databas. Datan i databasen kan användas för att skapa grafer och tabeller för senare användning.

### 1.3.4 Val av kommunikationsprotokoll

Vilket protokoll som passade bäst till projektet har undersökts, där fördelar och nackdelar har vägts mot varandra för att välja det mest passande. Det stod mellan ADS och OPC UA där man tittade på och jämförde säkerhet, prestanda och enkelhet.

### 1.3.5 Service-Oriented architecture(SOA)

Den betydelse SOA har för arbetet och hur det används ute i branschen har undersökts. Den ökade populariteten hos SOA och dess fördelar har analyserats.

## 1.4 Problemformulering

Examensarbetet som har utförts delades upp i tre problemformuleringar. Karaktären av detta examensarbete var mycket praktiskt arbete vilket krävde mycket planering och det som tänktes ut teoretiskt skulle vara möjligt att implementeras praktiskt. Mycket av det som planerades och tänktes ut i teorin var inte lämpligt för praktiskt bruk och då var man tvungen att ta andra vägar för att komma fram till rätt lösning.

- Vad innebär fastighetsautomation i praktiken och hur kopplar man ihop alla delar till att bli ett system?
- Vilka fördelar ger användandet av OPC UA- protokollet vid kommunikation mellan CX9020 och HMI?
- Vilken roll har SOA(Service Oriented Architecture) i detta projekt?

Problemformuleringarna besvarades under arbetet där framstegen och de olika lösningar man kom fram till gav en klar bild av det hela. När examensarbetet blev färdigt gav det en helt ny bild av fastighetsautomationen och dess byggstenar.

## 1.5 Avgränsningar

Ett projekt av denna typ kunde växa och bli hur stort som helst. Det enda som satte stopp var fantasin. Idén som Beckhoff hade till en början ansågs vara lite för stor för att utföra på den tiden som gavs. Planen var att analysera SOA i stor skala hos en av Beckhoffs kunder. Idén förändrades till att utföra ett praktiskt projekt som gav en inblick i hur arbetslivet för en automationsingenjör i fastighetsbranschen ser ut samtidigt som man fick utforska nya fält som SOA, DALI och OPC UA.

### 1.5.1 Belysning

Till en början var planen att använda en LUX-mätare och en rörelsedetektor. LUX-mätaren skulle mäta ljusstyrkan i rummet och belysningen skulle anpassas efter det. En rörelsedetektor skulle också installeras och användas för att göra det energisnålt och bekvämt. Med dessa sensorer hade man dessutom kunnat lagra mer data för analys och mätning. Tidsplanen och budgeten satte stopp för planerna och istället valde man att styra det via pekskärmen manuellt.

## 1.5.2 Luftkvalitet

Planen var att använda luftkonditionering av typen Mitsubishi som redan var installerad i rummet. Tanken var att ansluta till den och låta värdena från sensorn styra luftkonditioneringen. På grund av att Mitsubishi saknade anslutningsmöjligheter som hade kunnat användas lämnades idén åt sidan. Det avgränsades till att samla in data med sensorn och lagra dessa i databasen för att sedan kunna användas för bland annat grafitning.

## 1.5.3 SOA

Från början var idén en fullskalig analys och implementation av SOA i stor skala hos en av Beckhoff's kunder. Detta diskuterades och man kom fram till att det inte var möjligt att utföras på den korta tid som funnits tillgänglig. Därför blev det en praktisk installering i ett rum och en analys av SOA som tillägg på det.

## 2. Teknisk bakgrund

Examensarbetet var uppbyggt av många olika byggstenar. Gammal kunskap och erfarenhet fick kombineras med ny information och ny kunskap som hämtades från handledare och internet.

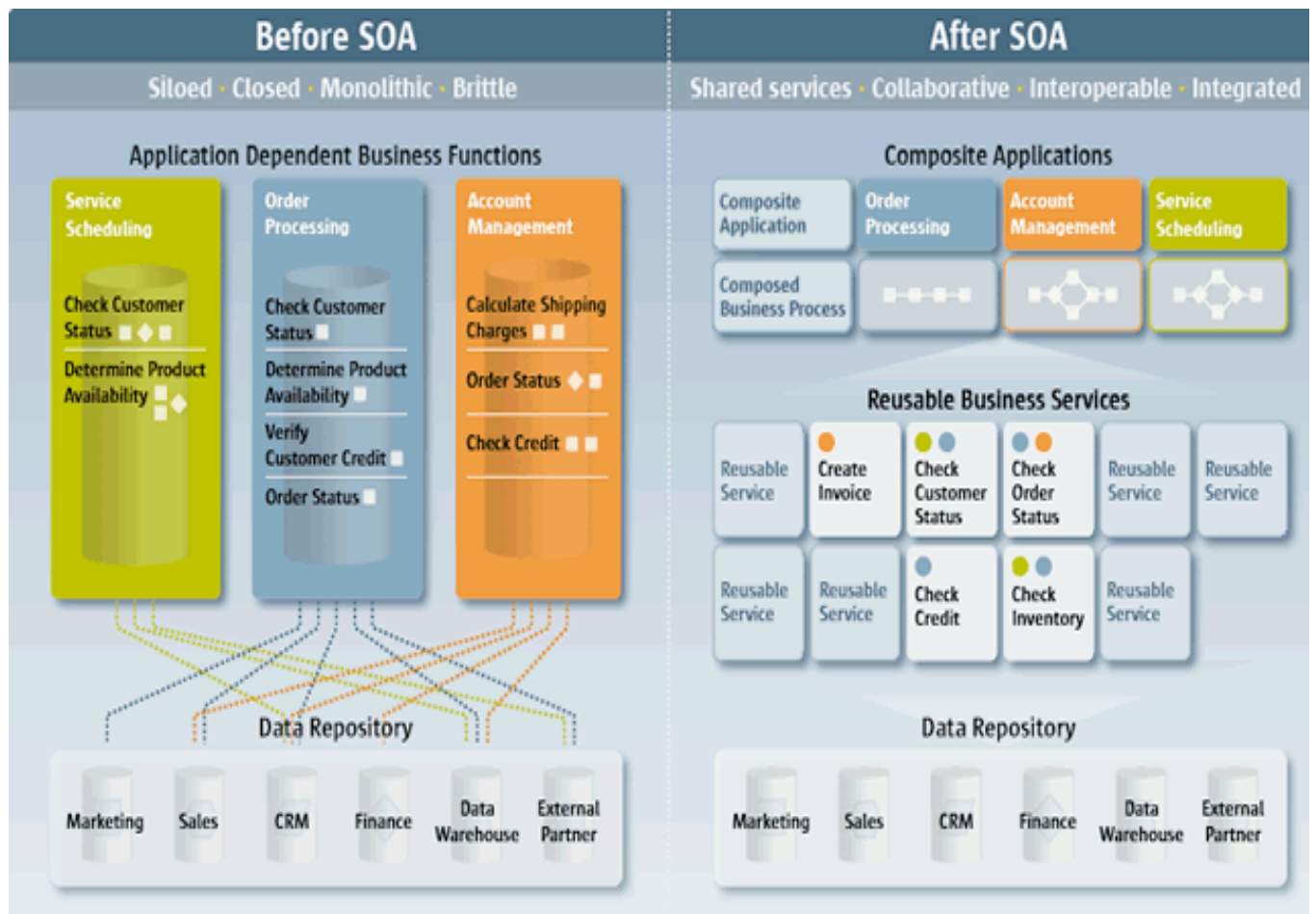
### 2.1. SOA och OPC-UA

SOA(Service- Oriented Architecture) är ett tankesätt som man använder för att skapa en arkitektur som är service-inriktad. Med service-inriktning menar man att det är ett speciellt tankesätt, med tjänster och tjänstebaserad utveckling i fokus. Begreppet tjänst innebär att det är en betjänande funktion som är oberoende av sin omgivning, väldefinierad och självständig.

Ett SOA är i själva verket en samling tjänster. Den förbättrar existerande arkitekturer genom att adressera de större systemen som tjänster och abstraherar sedan dessa tjänster till en enda domän där de formas till lösningar. Tjänsterna kommunicerar med varandra och kommunikationen kan bestå av enkelt datautbyte eller att flera tjänster koordinerar en funktion. I ett system med en tjänsteorienterad arkitektur(SOA) har andra system inom samma nätverk tillgång till systemets resurser.

I SOA försöker man återanvända befintliga tjänster för att spara tid och plats som i sin tur drar ner på kostnader. SOA ger en effektivitet och huvudsyftet är att uppfylla de affärsmässiga kraven på ett IT-system. [3][4]

Figur 1 nedan visar en överskådlig bild över strukturen hos SOA och de förbättringar som det medför.

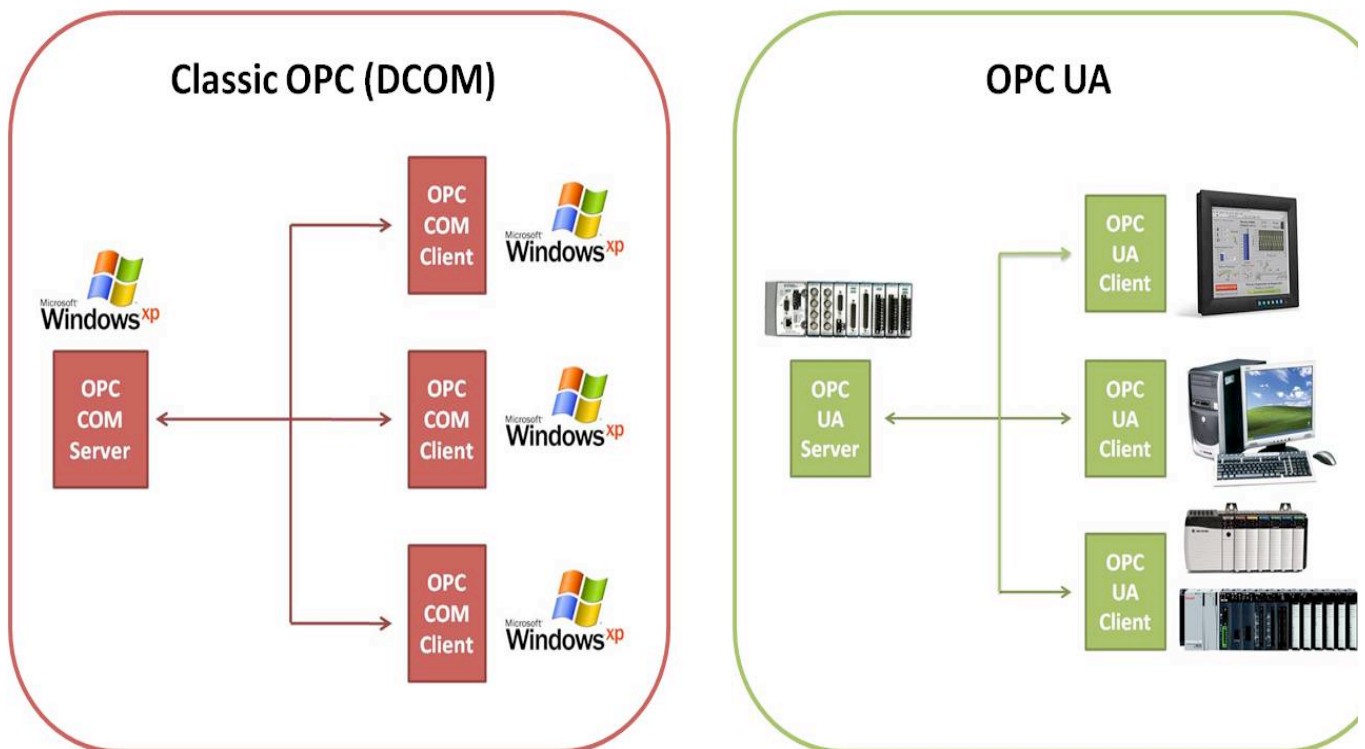


Figur 1: IT-system utan och med SOA arkitektur [7]

## 2.1.1 OPC UA

OPC UA är ett kommunikationsprotokoll utvecklat av ”OPC Foundation”. OPC UA är en plattformsoberoende tjänsteorienterad arkitektur(SOA). Till skillnad från sin föregångare OPC är det möjligt att implementera OPC UA på valfri plattform, t.ex. Apple, Linux eller Windows. Den första versionen av OPC UA släpptes år 2008 och senaste version är 1.02. [8][9]

Figur 2 nedan ger en överblick på hur en OPC UA- arkitektur kan se ut.



Figur 2: Figuren visar hur allt binds ihop med OPC UA[10]

## 2.2 DALI

Sent på 90-talet skapades en grupp, DALI-ag, med uppgiften att designa ett industriellt, digitalt protokoll för ljusstyrning. Arbetsgruppen bestod av representanter från stora företag som Osram, Tridonic som är ledande inom belysning. Kraven var att det skulle vara låga kostnader, enkel kabeldragning, individuell kontroll, återkoppling och möjlighet till att använda sensorer och andra verktyg. DALI, Digital Addressable Lighting Interface, är ett protokoll som är specificerat i IEC 62386 standarden. DALI är ett protokoll för kommunikation och kontroll av ljusutrustning. Det omfattar allt från armaturer till knappar och närvarodetektorer. Ett DALI system har 4 huvudkomponenter, det är:

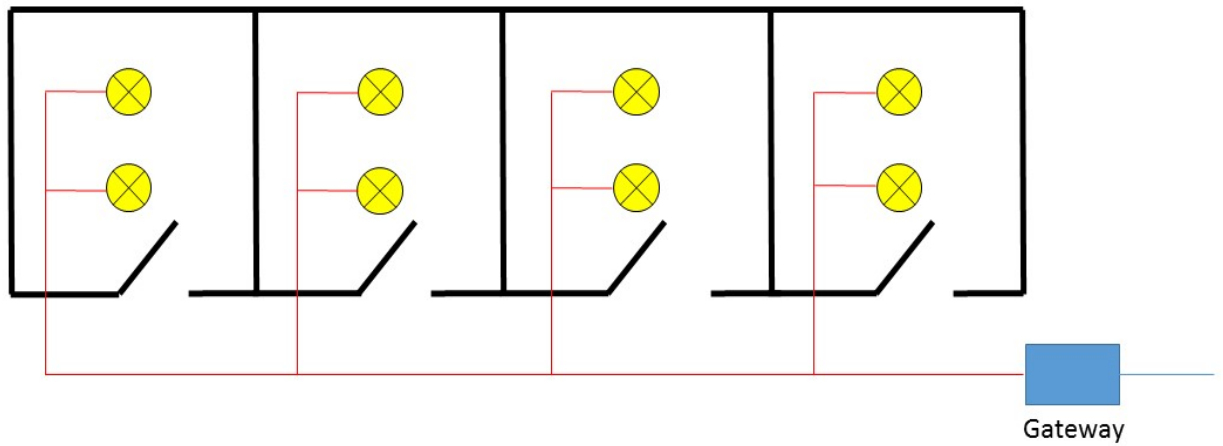
1. En DALI styrning(router, port, hub)
2. En DALI bussmatning
3. DALI komponent
4. Sladdar

DALI har börjat ersätta den analoga styrningen med 0-10V och DSI. För att kunna ha en dimmerfunktion innan DALI krävdes mycket kabeldragning och behövdes en förändring var man tvungen att koppla om kablarna. Det slipper man med DALI-belysning. DALI och DSI är väldigt liknande särskilt sett till det fysiska systemet men DSI kan bara skicka broadcast-värden.

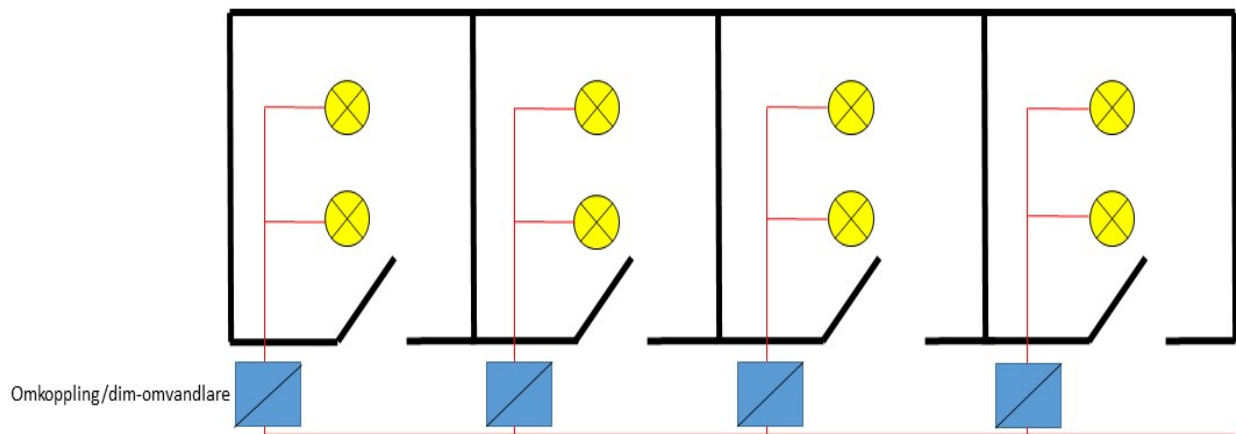
Figur 3 och 4 nedan visar exempel på kopplingar och skillnaden mellan DALI och klassisk 1-10V/DSI. [23]



# DALI



# 1-10V eller DSI



Figur 4: 1-10V eller DSI belysning i ett rum[13]

## 2.3 SCADA

SCADA (Supervisory Control And Data Acquisition) är ett mjukvarusystem som används i industrin för att:

- Kontrollera
- Övervaka
- Analysera

SCADA-system används i de flesta industriella processer i hela världen. Några av många exempel på industriella processer som använder sig av SCADA-system är:

- Elkraftsproduktion, överföring och distribution  
Elbolag använder SCADA-system för att upptäcka strömflöde och nätspänning.
- Vatten- och avloppssystem  
Statlig och kommunal vattenanvändning kontrolleras av SCADA-system för att reglera vattenflöde, trycket i vattenledningarna och andra viktiga faktorer.
- Tillverkning  
Inom tillverkningsindustrin används SCADA-system för att styra industriell automation och robotar. Det används även för kvalitetskontroller.

Några exempel på SCADA-system [14]

Det finns en hel del systemkomponenter i ett SCADA-system.

Man använder en PLC (Programmable Logic Controller) eller en RTU (Remote Terminal Unit) för att samla in signalerna från de olika sensorerna. I dessa görs signaler om till digital data. En PLC har en mer sofistikerad kontrollkapacitet än en RTU. PLC:n används oftast istället för RTU:n då den är mer mångsidig, flexibel och mer ekonomisk.

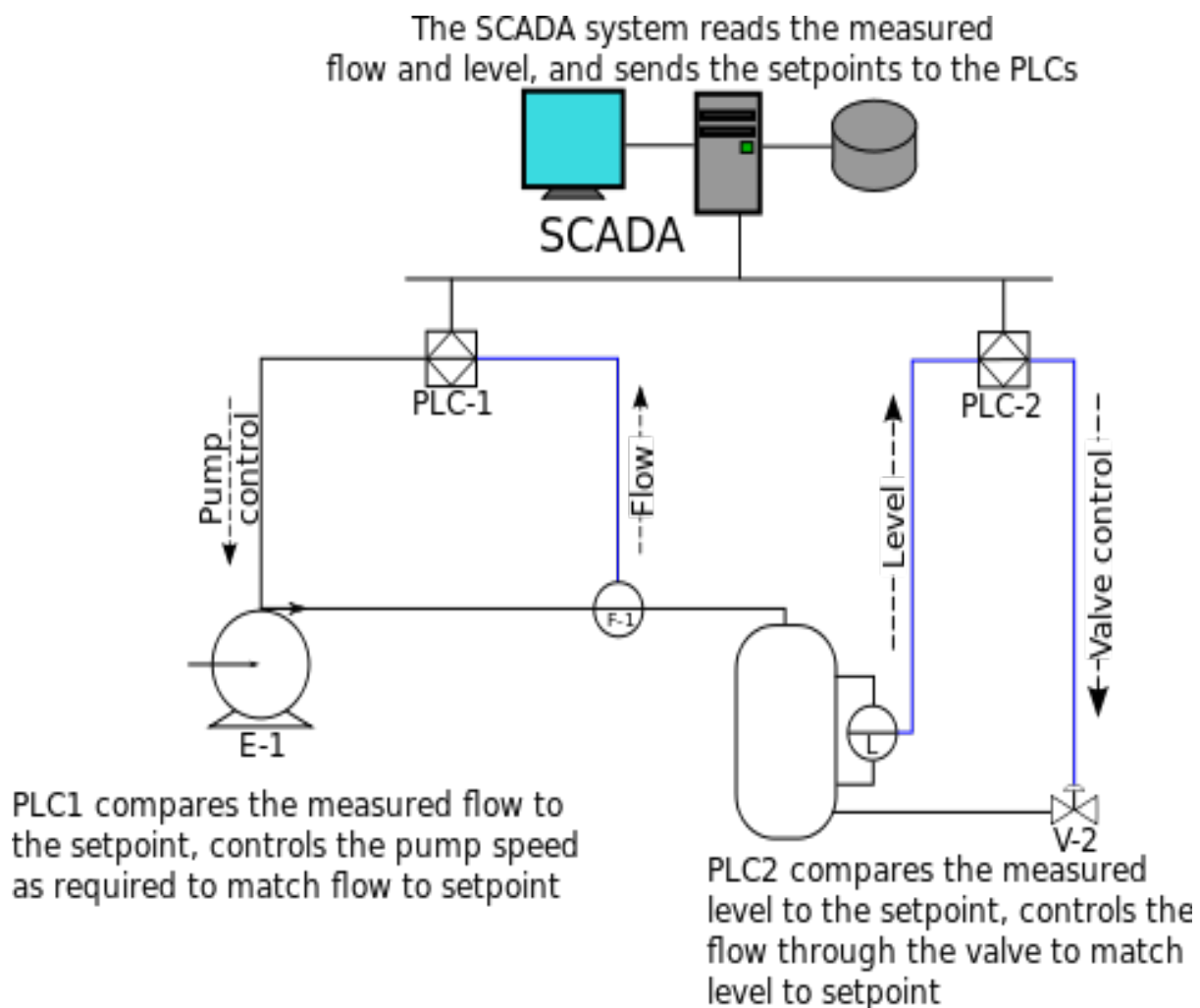
En datainsamlingsserver är en mjukvara som använder industriella protokoll för att kommunicera med fälltenheterna (t.ex. sensorer) via bland annat PLC:n eller RTU:n. Detta gör att klienter får tillgång till fälltenheterna med hjälp av standardprotokoll.

Ett HMI använder den bearbetade datan från datainsamlingsservern för att presentera detta till operatören t.ex. på en skärm. Med hjälp av detta kan operatören sedan ändra och integrera med processen. Ett HMI kan utöva alarm, ge notifikationer eller skapa rapporter.

En mjukvara t.ex. en databas används för att hålla koll på historik av datan. Även denna mjukvara får sin information från datainsamlingsservern. Genom att hålla koll på datans historia kan man även beräkna hur det kommer se ut i framtiden. Man brukar även använda den gamla datan för att skapa grafer.

Systemkomponenter i SCADA [15]

Figur 5 visar hur ett typiskt SCADA-system fungerar.



Figur 5: Exempel på ett SCADA-system [16]

## 2.4 CX9020

Beckhoff CX9020 är ett kompakt Ethernet styrsystem som kan monteras på DIN-skenor. CX9020 har en 1 GHz ARM Cortex™ A8 CPU. CX9020 är ingen traditionell PLC, utan en embedded PC. Det är i princip en industriell dator med Windows CE installerat. Windows CE, nu officiellt kallat Windows Embedded Compact är ett operativsystem från Microsoft som används i så kallade inbyggda system.

Anslutningen för Beckhoff I/O-systemen är integrerade i CPU-modulen. Enheten erbjuder automatisk identifiering av bussystem. CPU har två microSD-kortplatser, 4 USB-2.0 gränssnitt och ett DVI-D gränssnitt. Till CX9020 används TwinCAT3 mjukvaran vilket gör CX9020 till en kraftfull PLC. CX9020 klarar av temperaturer mellan -25 och +60°C vilket gör den till en flexibel industri-PC. Till CPU:n är det enkelt att lägga till terminaler som t.ex. digitala utgångar, DALI terminalen. Terminalerna fästs på DIN-skenan och ihop med närliggande terminal eller CPU. [17]

Bild nedan visar CX9020 med flera olika terminaler

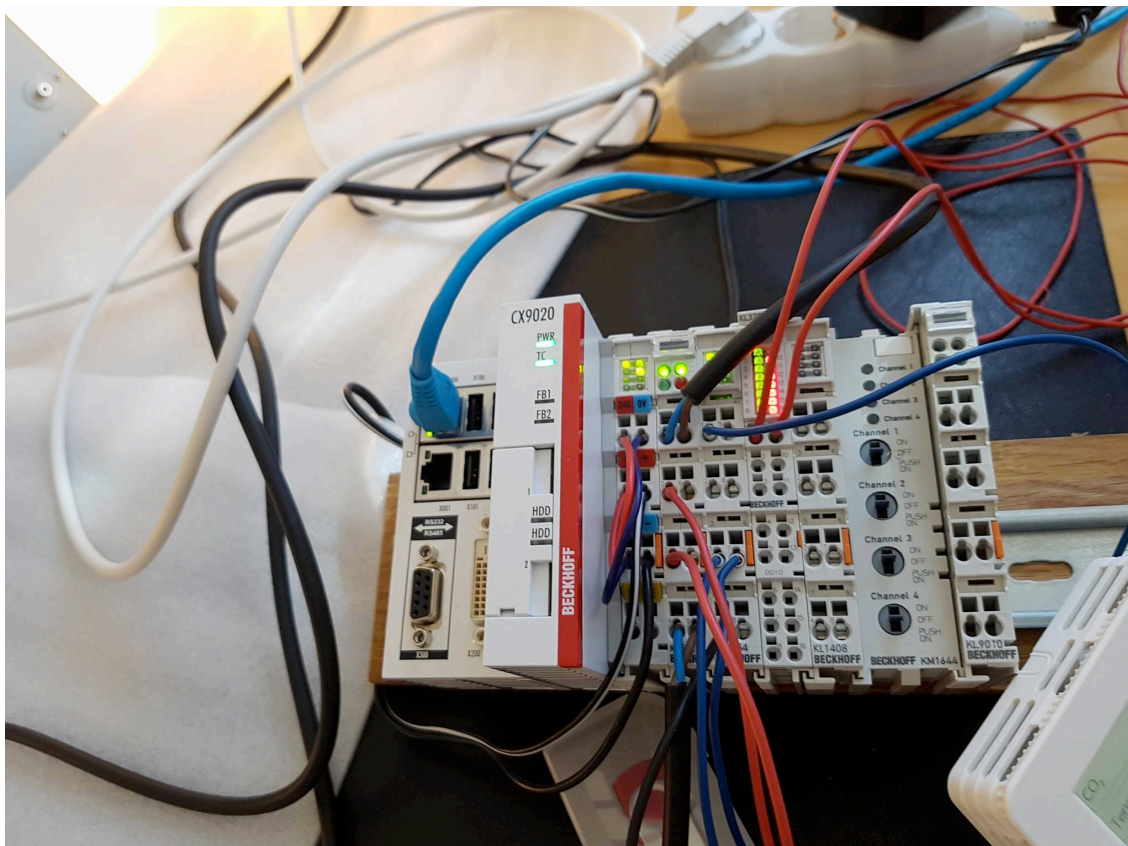


Bild 6: CX9020 med terminaler(foto)

## 2.5 tSENSE™

tSENSE™ används för att samla data från miljön i rummet. Den har tre inbyggda sensorer. En som mäter luftfuktigheten (%RH), en som mäter temperaturen (°C) och en som mäter luftrenligheten alltså koldioxidhalten (ppm).

Fördelar med tSENSE™:

- Underhållsfri
- Tre sensorer i en apparat
- Touchdisplay

Några fördelar med tSENSE™[18]

tSENSE™ sänder data till styrsystem som t.ex. en PLC via utsignaler som följer industrins standard för kommunikation. Utsignalerna görs om från 0-10V till 0-2000 ppm, 0-50 CO<sub>2</sub> och 0-100 %RH.

För att ändra inställningar och trimma sensorerna tillkommer det även en mjukvara som heter UIP5 som hjälper till med detta. [18]

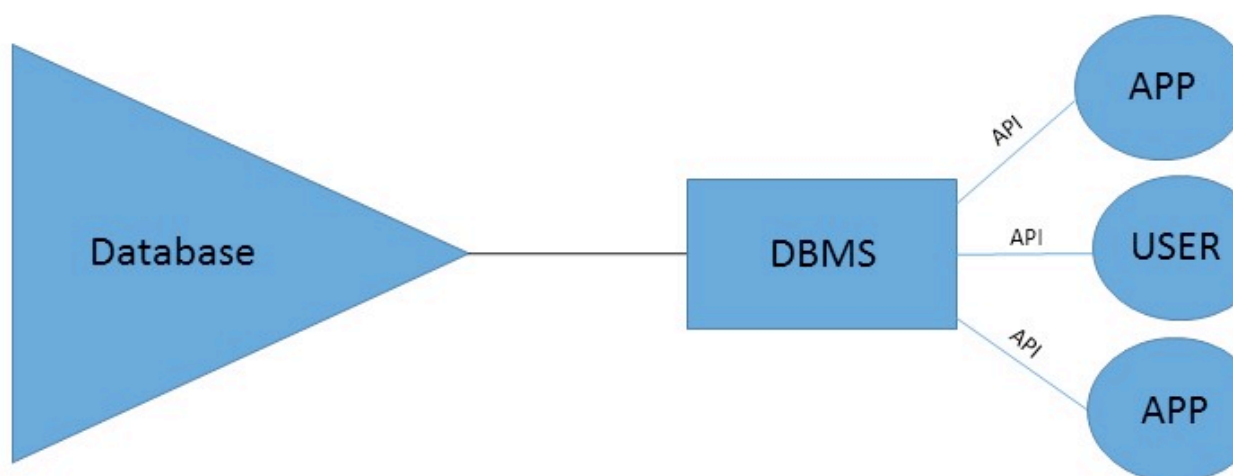
## 2.6 Databaser

Om man vill lagra data är det en bra lösning att använda en databas. En databas är en samling information, där det är lätt att hämta bitar V information och även ändra informationen. Informationen i en databas hanteras av en databashanterare. En databas kan givetvis lagra information från flera olika klienter som t.ex. sensorn och manuellt insatta värden.

Det finns olika typer av databaser:

- Hierarkisk databas  
I en hierarkidatabas lagras informationen i ett träd. Det går snabbt att bearbeta informationen, men långsamt att manipulera den.
- Relationsdatabas:  
En relationsdatabas innehåller tabeller. Tabellerna som även kallas relationer består i sin tur av rader. Det är i dessa rader som information lagras. I vanliga fall relaterar tabellerna med varandra t.ex. en tabell med personer och en tabell med lediga platser i en biosalong.
- Objektorienterad databas  
Objektorienterade databaser även kallade XML-databaser, gör dem java-objekt och lagrar dem i vanliga relationsdatabaser. [19]

Figur 8 visar hur kommunikationen mellan en databasserver och klienter fungerar där DBSM är databashanteraren



Figur 8: Exempel på struktur i databaskommunikation

### 3. Metod

Under arbetets gång har alla uppgifter och delmoment gjorts tillsammans. Varför man valde att göra allt tillsammans berodde på tidigare erfarenhet av det arbetssättet och att man kompletterar varandra väldigt bra. Allt arbete gjordes hos Beckhoff där ett laborationsrum upprättades för arbetet. Dessutom fanns det alltid någon att fråga om tips och hjälp när det väl behövdes.

Planeringen av examensarbetet var det första som gjordes. Olika deadlines för varje delmoment sattes upp och ett ungefärligt datum på när arbetet skulle vara färdigt planerades också in. I planeringen ingick även möten samt skrivningen av examensarbetebeskrivningen. Efter ett par veckor påbörjades en utbildning i TwinCAT där huvudfunktionerna samt programmeringstekniken på Beckhoff förklarades.

När materialet fanns på plats satte arbetet igång. Det första som påbörjades var att ansluta DALI-armaturerna och adressera dessa för att kunna labba med. De kommande veckorna spenderades helt och hållet till att vänja sig vid TwinCAT och programmera DALI-armaturerna.

Efter belysningen var det sensorns tur. Sensorn som beställdes in mätte temperatur, CO<sub>2</sub>-halt och luftfuktighet och gav en bra beskrivning av miljön i rummet samtidigt som man slapp ha olika sensorer för varje mätvärde. Nu behövde man ansluta, konfigurera den och sedan var det programmering i TwinCAT som gällde. De råa värdena från sensorn skalades om till temperatur, CO<sub>2</sub> och %RH. När det var färdigt kunde man gå vidare till nästa steg som var konfigureringen av databas.

Detta arbetsmoment gjordes med Beckhoffs Database Configurator, se bild 20. Databasen placerades i den lokala datorn för att där fanns gott om diskutrymme vilket kan behövas i framtiden. Som verktyg till databasen valdes MySQL Workbench eftersom det har använts tidigare och är ett bra program som användes för att kontrollera insättningar som gjordes från TwinCAT. Ett program skrev vars uppgift var att skicka in de skalade mätvärdena till databasen var femtonde minut, se bild 21. Delmoment fem var avklarat och man gick vidare till valet av protokoll.

Valet av protokoll diskuterades med handledaren och OPC UA föreslogs. Redan vid första mötet förklarades OPC UA och dess fördelar, se 4.5.1. En bedömning

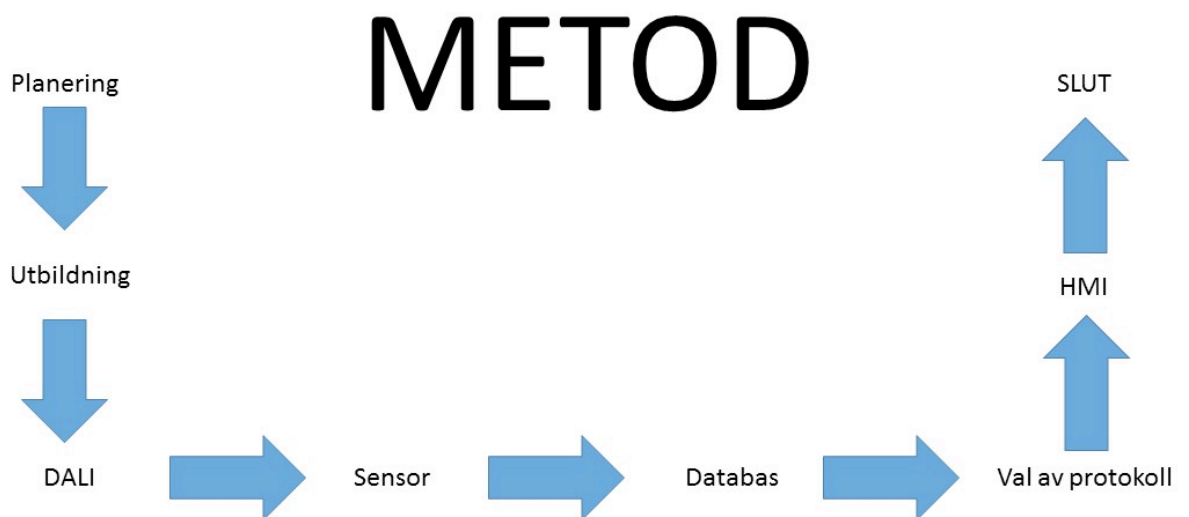


gjordes efter påläsning och OPC UA valdes som kommunikationsprotokoll. En OPC UA server konfigurerades och det som återstod var ett HMI som knyter ihop allt.

Till en början skapades prototyper i TwinCAT för att användas vid laborationerna eftersom hårdvaran inte hade tillräckligt med ”switchar”. Prototyperna gav dessutom en bild på hur det slutliga HMI:et skulle se ut och man upptäckte fel och missar innan man hade påbörjat det slutliga HMI:et.

För att skapa HMI:et som skulle användas rekommenderades programmet Indusoft av Beckhoff. Ett gratis SCADA-program med många nyttiga funktioner. En utbildning om Indusoft arrangerades och efter den satte arbetet med HMI:et igång. HMI krävde mycket arbete och det skulle vara snyggt och lättanvänt för att användas i praktiken efter att allt hade installerats.

Bilden nedan visar varje delmoment och i vilken ordning de utfördes.



## 3.1 Källkritik

De flesta källor som har valts för att kunna slutföra examensarbetet har varit källor som är från företag som producerar produkten/produkter som använts i arbetet.

Encyklopedier som Wikipedia har man försökt undvika då de är inte fullständigt trovärdiga. Vid val av källor har man dessutom försökt se dem ur ett objektiva perspektiv så att man inte förlitade sig blint på vad tillverkaren av en viss produkt hävdade.Handledaren och anställda på Beckhoff fick också säga sitt, vad de tyckte och tänkte om besluten som togs. För vanlig fakta som t.ex. fakta om Beckhoff användes Wikipedia. För information och referenser om Beckhoffs produkter användes deras hemsida[18].

För valet av protokoll användes flera olika källor som [8][9][10] för att jämföra och se vilka fördelar protokollet gav.

För DALI-belysning användes både grundarens hemsida [11] samt tillverkare av DALI-belysning [13].

Analysen av SOA innebar mycket teori och flera källor[3][5][6] användes för diskussionen av SOA. Beckhoff introducerade och förklarade SOA och dess fördelar vid första mötet och man förstod att det var något som de ansåg vara väldigt bra och nyttigt.

Sammanfattningsvis var valet av referenser baserat på trovärdighet, upplevda kvaliteten samt innehållet. Referenserna lästes igenom noggrant och sedan togs ett gemensamt beslut om den var användbar eller ej.

## 4. Analys

### 4.1 Programmering och strukturering

På Beckhoff tillämpades programmeringsprincipen att själva programstrukturen och organisationen av filer och kataloger var väl så viktiga som programmeringen i sig. Namnen på program och variabler skulle vara enkla och lättlästa. Man använde många mappar så att allt var på rätt plats och lätt att finna vid sökning. Detta ansågs väldigt viktigt för att underlätta för den som programmerar men främst för andra användare. En bra strukturering gav en överblick på hur programmet var uppbyggt och hur det fungerade. Bilden nedan visar hur projektet i TwinCAT3 såg ut.

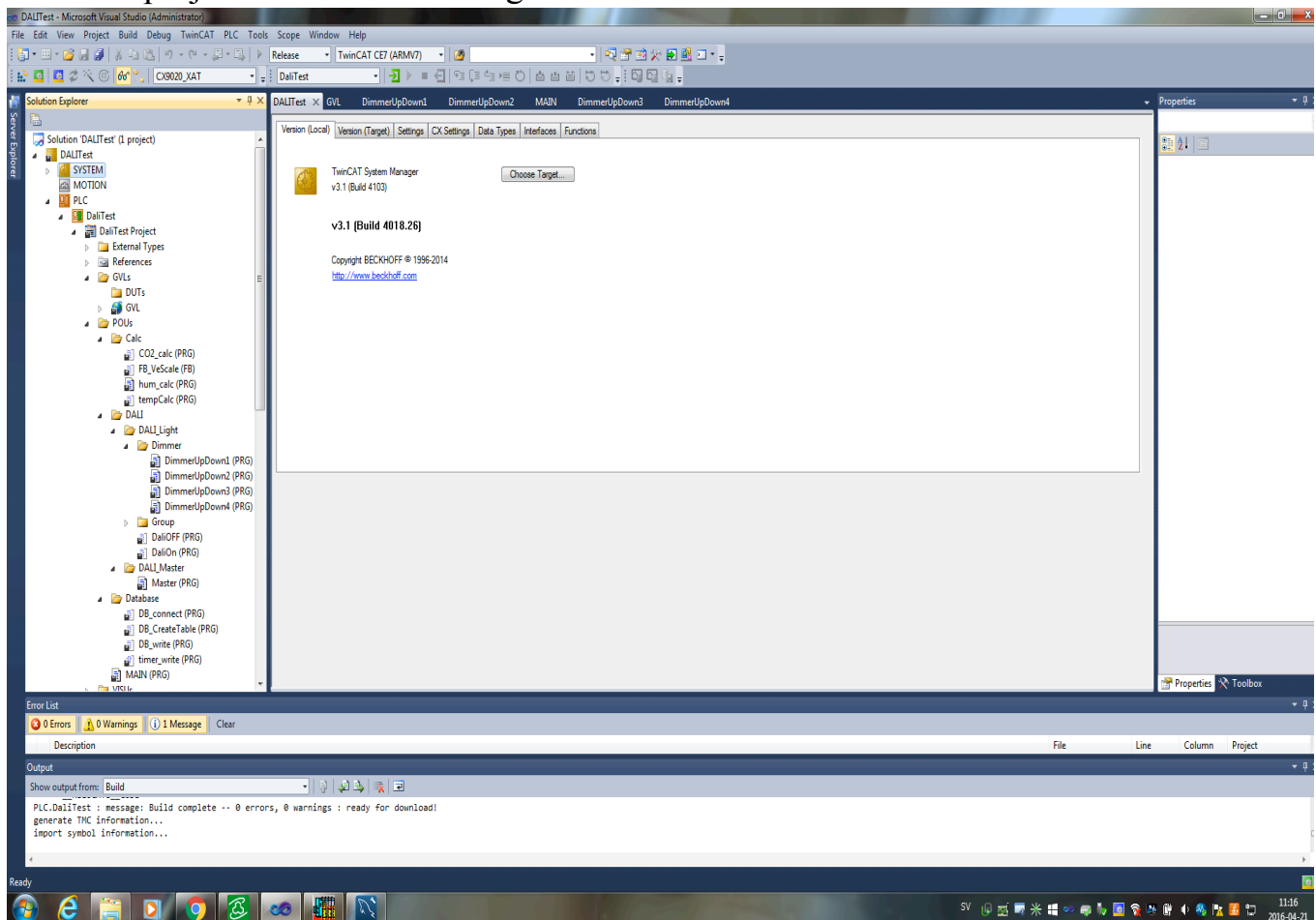


Bild 9:Projekt i TwinCAT3

## 4.1.1 Funktionsblock

Hos Beckhoff användes i stor utsträckning ST(strukturerad text) som programmeringsspråk för PLC-programmering. Man använde färdiga block som hämtades från Beckhoffs Information System. Till en början var det svårt att förstå vad som skulle skrivas men när man blev bekant med hemsidan kunde man snabbt hitta information och hjälp.

Bilden nedan visar hur ett funktionsblock kan se ut. Här är det ett funktionsblock som användes för att slå på/av och dimma belysningen.

```
1 PROGRAM DimmerUpDown1
2 VAR
3     fb_Dim1:FB_DALIV2Dimmer2Switch;
4
5 END_VAR
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
fb_Dim1(
    bSwitchDimmUp:= ,
    bSwitchDimmDown:= ,
    bOn:=butt_ON1,
    bOff:=butt_OFF1,
    bSetDimmValue:= ,
    nDimmValue:= dimmer_1,
    tSwitchOverTime:=T#20MS ,
    tCycleDelay:= ,
    bMemoryModeOn:= true ,
    nOnValueWithoutMemoryMode:= ,
    nAddr:=1 ,
    eAddrType:= 0 ,
    nMasterDevAddr:= ,
    nMinLevelMasterDev:= 125,
    nMaxLevelMasterDev:=254 ,
    tCycleActualLevelMasterDev:= ,
    nActualLevelMasterDev=> ,
    bBusy=> ,
    bError=> ,
    nErrorId=> ,
    stCommandBuffer:=master.stDaliData );
```

Bild 10: Funktionsblock för belysning

Här har alltså en instans av FB\_DALIV2Dimmer2Switch som är ett färdigt block på Beckhoffs sida skapats. bOn, bOff och nDimmValue var kopplade till variabler som kunde styras via HMI. Dessa tre variabler skötte dimfunktionen, på och av.

## 4.2 DALI

### 4.2.1 Installering och konfigurering av DALI-armaturer

DALI-armaturerna som skulle användas saknade stickpropp till vägguttag så man var tvungen att koppla dit nya sladdar med vanlig stickpropp för att kunna labba med dem. DALI-armaturen har en sladd med två kontakter, DALI+ och DALI-, som går direkt till KL6811 terminalen.

För att kunna gå vidare behövde man sätta adress på varje armatur. Det gjordes med Beckhoffs mjukvaruprogram KS2000. Programmet sökte efter enheter som var inkopplade och sedan kunde man styra dem och sätta adress. Adressen var väldigt viktig och alla armaturer skulle markeras.

Nedan kan man se en skärmbild under arbetet med KS200.

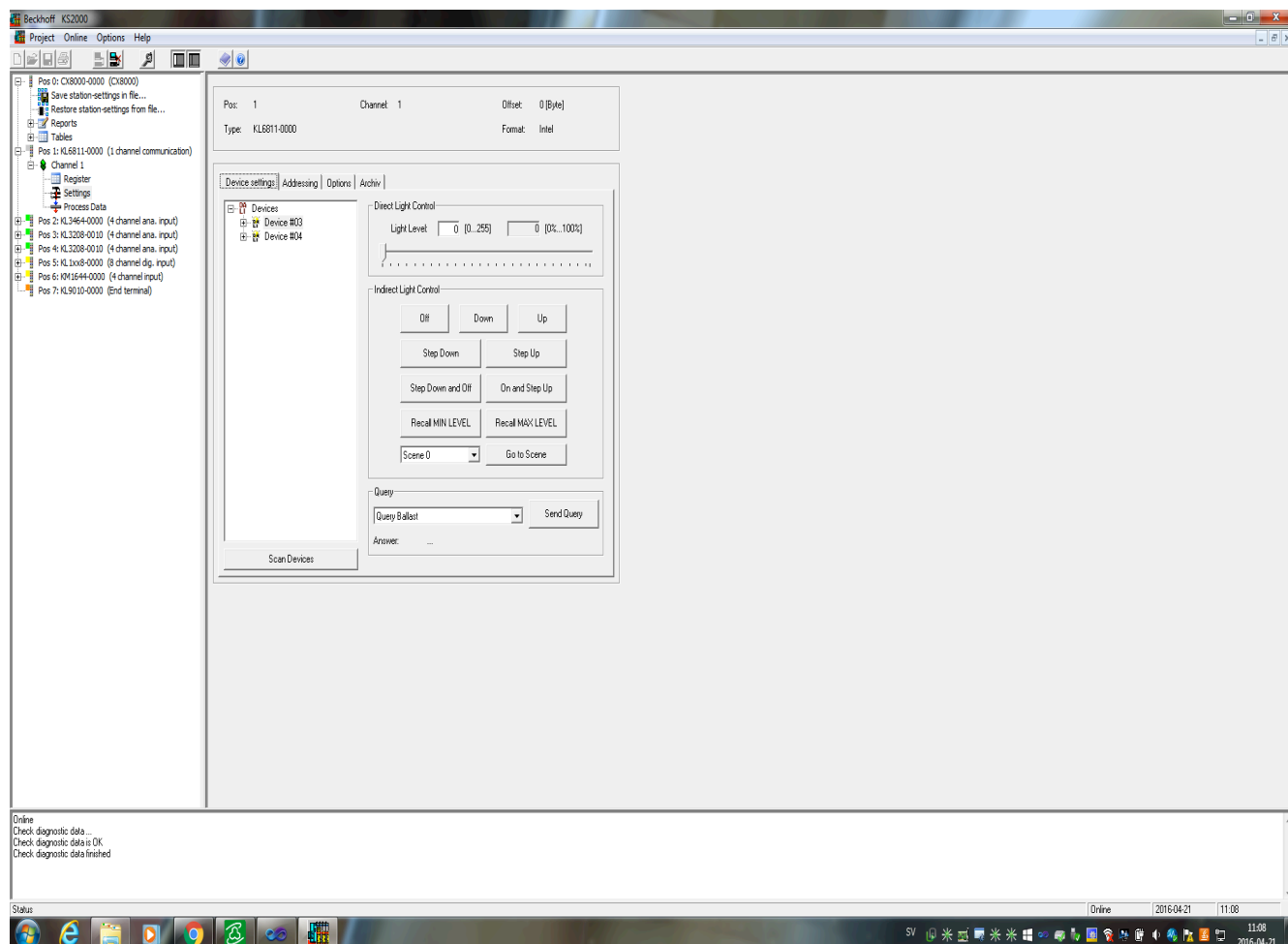


Bild 12: Skärmbild på KS2000 vid konfigurering av armaturer

## 4.2.2 Programmering av DALI

Efter att ha satt adresser på alla armaturer så kunde programmeringen av dem påbörjas. Till en början var man tvungen att ha ett så kallat ”master-program” som skötte all kommunikation med KL6811.

Detta viktiga block som användes är FB\_DALIV2Communication(bilden nedan). Detta block användes eftersom resterande DALI-block inte har direkt tillgång till DALI-terminalen (KL6811). Blocken placerar istället sina kommandon i en buffert(vektor) som skickas till FB\_DALIV2Communication-blocket. Blocket läser in dessa kommandon i tur och ordning och skickar dem till CX9020 via KL6811-terminalen. Detta görs för att förhindra att flera block försöker få tillgång till KL6811 samtidigt. I TwinCAT ställdes det samtidigt in att FB\_DALIV2Communication ska ha en cykeltid på 4ms för att anropas oftare medan alla andra har 10ms. stCommandBuffer är där en vektor initieras och i alla andra DALI-block finns det en ingång där man refererade till just denna vektor(se bild 8). Skärmbilden nedan visar hur blocket FB\_DALIV2Communication såg ut. [21]

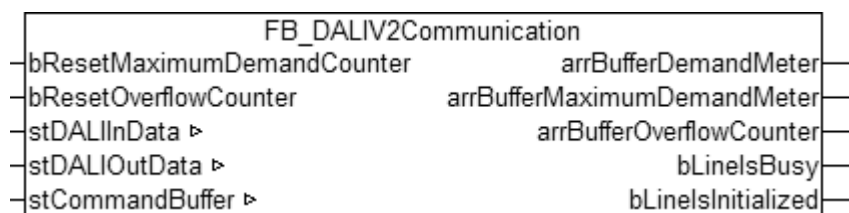


Bild 13: FB\_DALIV2Communication[21]

Nu kunde man skapa sina funktionsblock för att styra DALI-armaturerna. I varje block anropade man master-vektorn till en början och sedan kunde man fylla i ingångarna som behövdes.

Bilden nedan visar blocket som användes för att styra armaturerna.

Ingångarna bOn och bOff var vanliga boolean-ingångar som skiftade mellan true och false. Dessa variabler var kopplade till HMI:et via OPC UA och styrdes via pekskärmen. nDimmValue var en INT som ändrades med ”slidern” som finns i HMI:et och där man har ställt in ett max och min-värde på 254 respektive 125 vilket kan ses längre ner i blocket. bMemoryModeOn var en annan boolean-ingång som innebar att man hade den ljusstyrkan som var aktuell när man släckte lampan sist. Vi aktiverade denna funktion vilket var väldigt behagligt vid användning. Ingångarna nAddr och eAddrType är två ingångar som gjorde att man behövde konfigurera armaturerna i KS2000. I nAddr skrevs en av adresserna in, 1-4, och eAddrType kunde ändras till 0, 1 eller 2. Adresstypen 0

innebar att man kommunicerade med en enda armatur, 1 var kommunikation till två eller flera armaturer och 3 var broadcast. Längst ner finns stCommandBuffer, där skrevs master-vektorn från master-blocket in så att blocken kunde kommunicera mellan varandra.



Bild 14: Funktionsblock för styrning av DALI-armaturer[22]

## 4.3 HMI

### 4.3.1 Första HMI-prototyp

När det första HMI:et påbörjades var det ganska komplicerat att knyta ihop det med TwinCAT

Ett enkelt HMI i TwinCATs inbyggda HMI-funktion skapades först. Denna funktion var tillräcklig för att göra ett acceptabelt HMI men designmässigt var det inte tillräckligt.

Till en början implementerades funktioner så att lamporna kunde tändas och släckas. ”Switcharna” på CX9020 byttes ut mot ON/OFF-knappar i HMI:et. En dimfunktion lades till som gjorde att man kunde dimma lamporna.

Bild 15 visar en bild på första HMI.

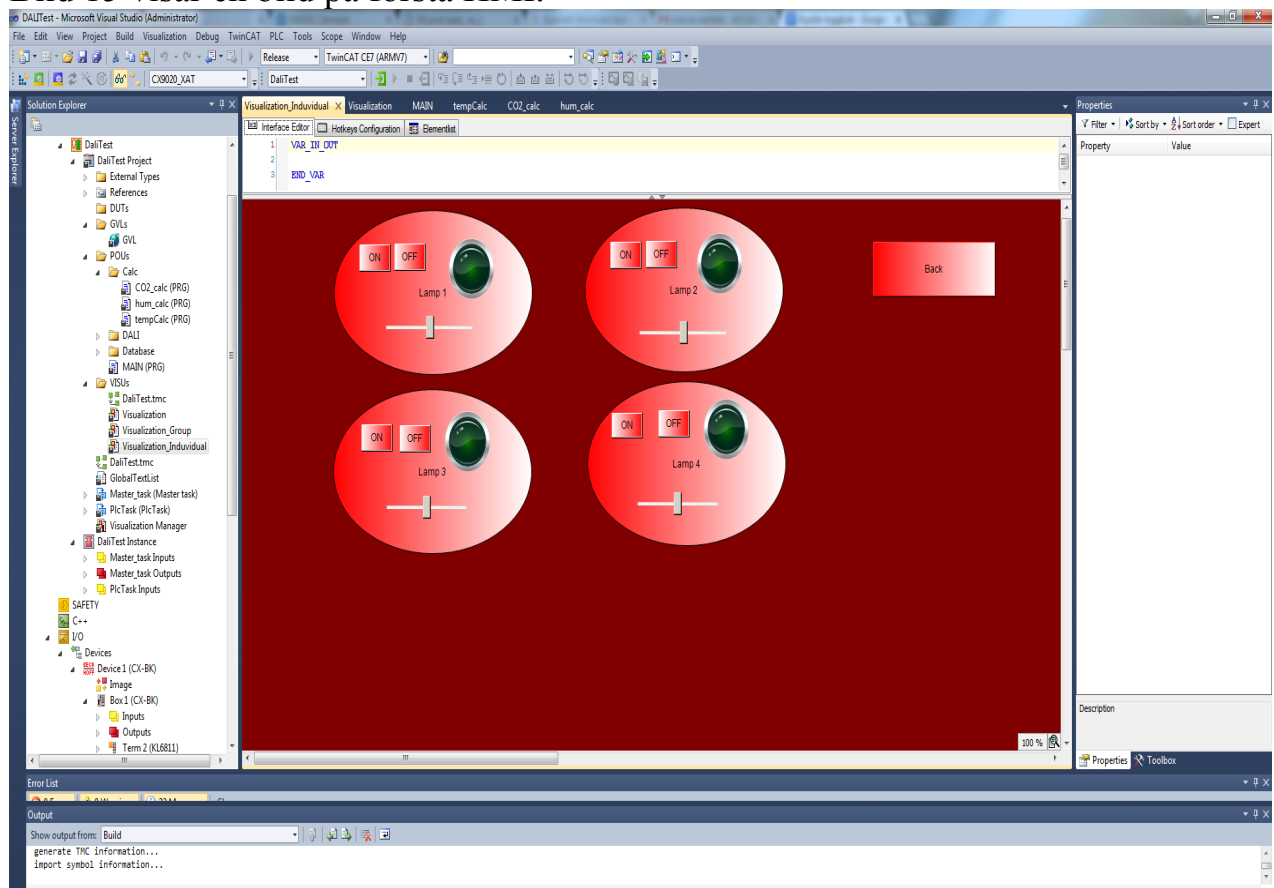


Bild 15 : Första HMI



## 4.3.2 HMI

För att skapa det slutliga HMI:et stod det mellan två program, Indusoft och C#. Programmet Indusoft valdes eftersom det hade fler inbyggda funktioner som kunde utnyttjas. TwinCATs inbyggda HMI valdes bort eftersom det inte uppfyllde de designmässiga kraven. Det erbjöd dessutom inte lika mycket möjligheter som Indusoft gjorde.

Till en början var tanken att gå på samma spår som i TwinCATs inbyggda HMI, med fyra separata styrningar, en för varje armatur och dess knappar. Detta ändrades efter att ha granskat tidigare HMI. Nya idén var att försöka efterlikna kontoret där armaturerna skulle placeras i och placera varje lampa på HMI:et där de skulle vara i verkligheten. Första uppgiften blev att efterlikna rummet så bra som möjligt. Tre nyckelobjekt i rummet var ett bord med stolar, en projektorduk och armaturerna som skulle styras.

Förutom att implementera en ON/OFF och dimmfunktion till lamporna, lades också ett par andra funktioner till. En alarmfunktion för att upplysa användarna av rummet när det blev för varmt, för hög luftfuktighet eller när koldioxidhalten blev för hög. En funktion som kunde visa upp lite historia om rummets luftkvalité och användning implementerades också. Detta gjordes genom att spara värden i en databas och sedan skriva ut dem i grafer.

Bild 16 visar startsidan på HMI:et. I menyn till vänster finns det olika alternativ man kan välja. Om man klickar på "Lightning" kommer man till lampstyrningen. Vill man stänga av ett alarm som gått igång gör man det under "Alarm". Historik av all samlad data de senaste 24 timmarna finns samlad under "Graph".



Bild 16: Startmeny

Bild 17 nedan visar hur lampstyrningen fungerar. När man klickar på "Lightning" dyker det upp två extra knappar i menyn. Dessa två knappar används för att styra alla lamporna samtidigt. "Turn on all" tändar alla lamporna, medan "Turn off all" släcker alla.

Om man vill styra lamporna individuellt kan man också göra det. Man kan klicka på varje lampa för att tända eller släcka den. För att dimma de tända lamporna kan man dra "slidern" upp och ned.

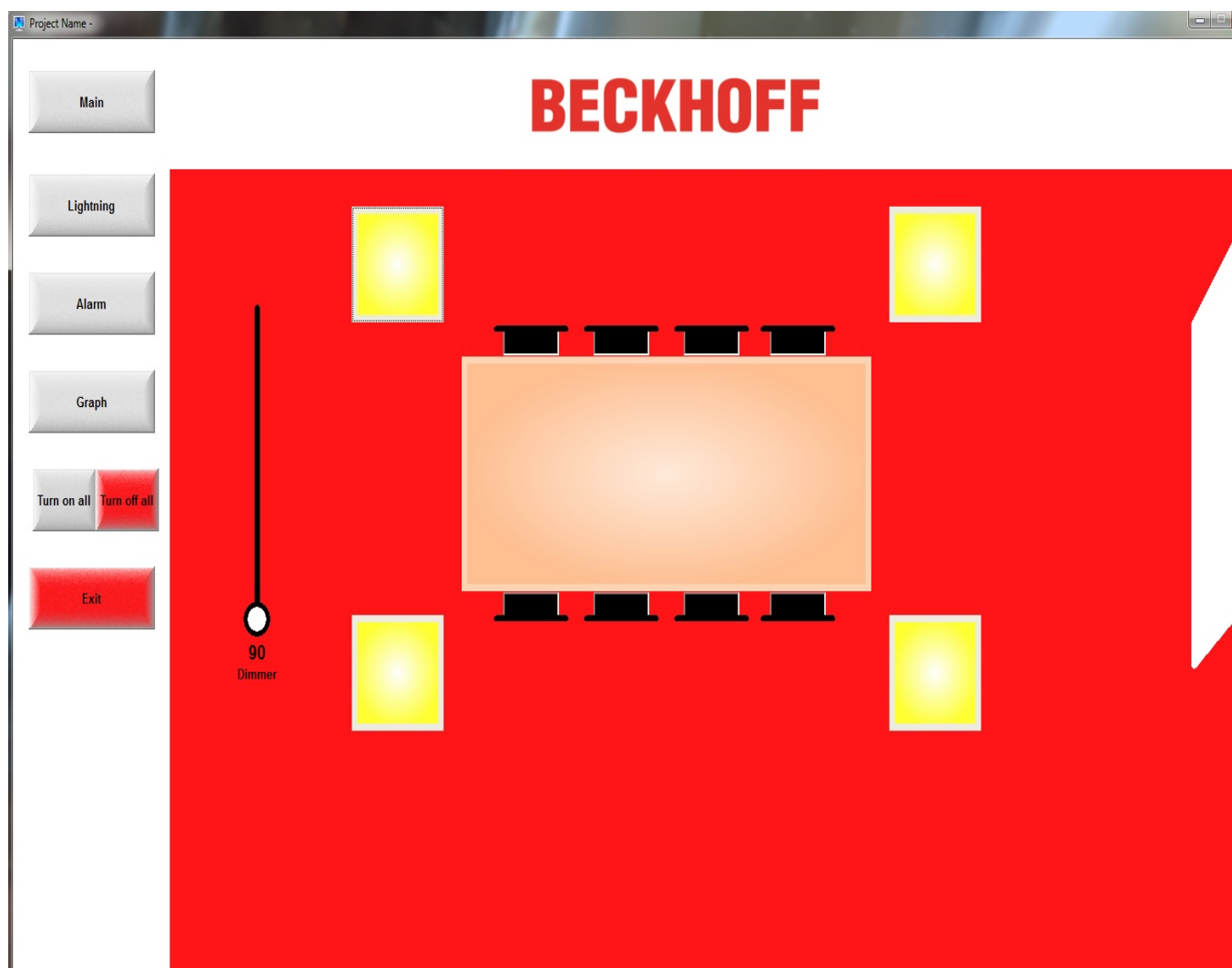


Bild 17: Lampstyrning

Bild 18 visar hur alarmet fungerar. Om temperaturen, luftfuktigheten eller koldioxidhalten är för hög går ett alarm igång med meddelade till användaren.

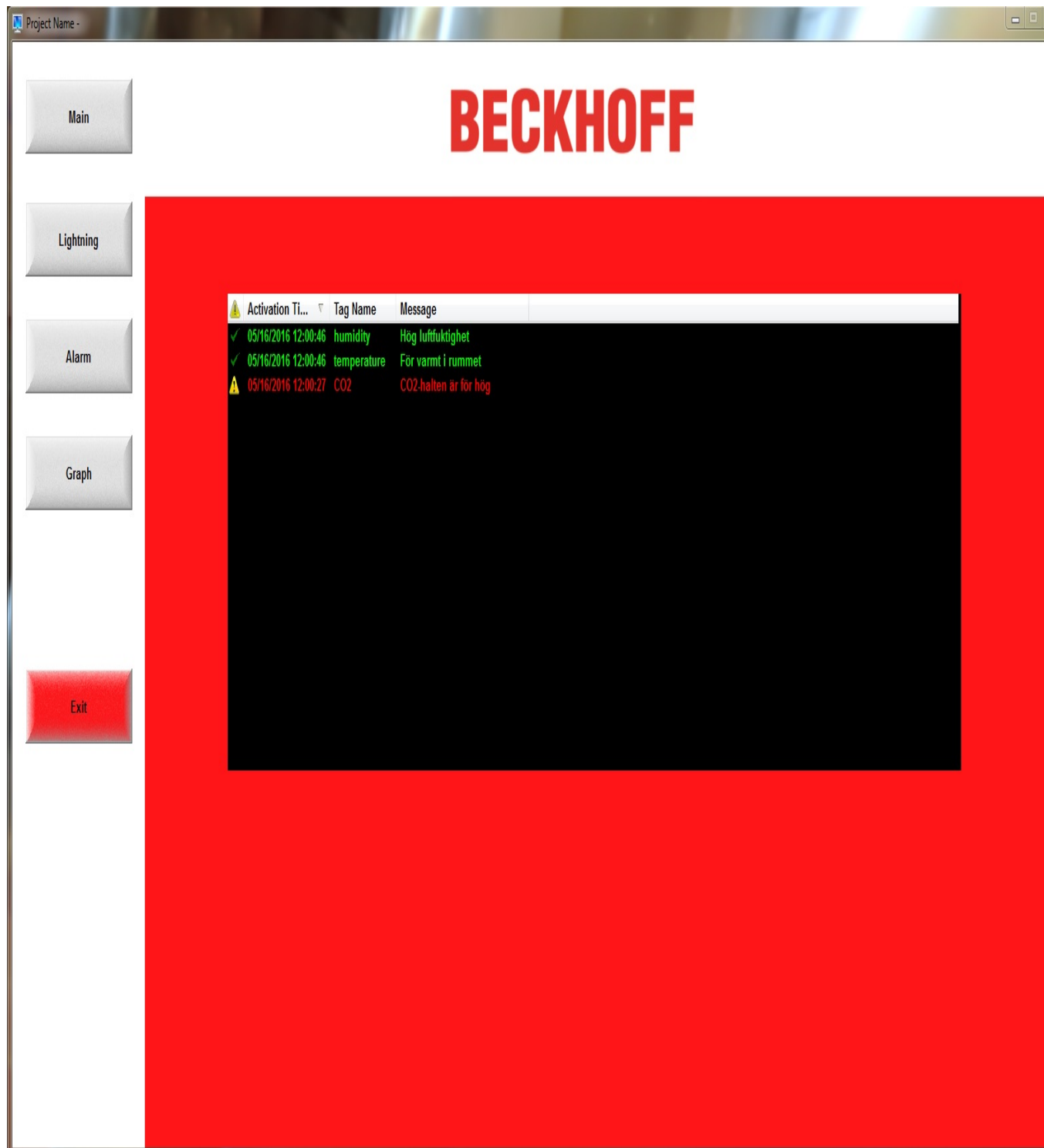


Bild 18: Alarm som triggas när värdena når inställda gränser

Bild 19 visar hur grafen för CO2 ser ut. Till vänster finns alternativen tillgängliga för användaren.

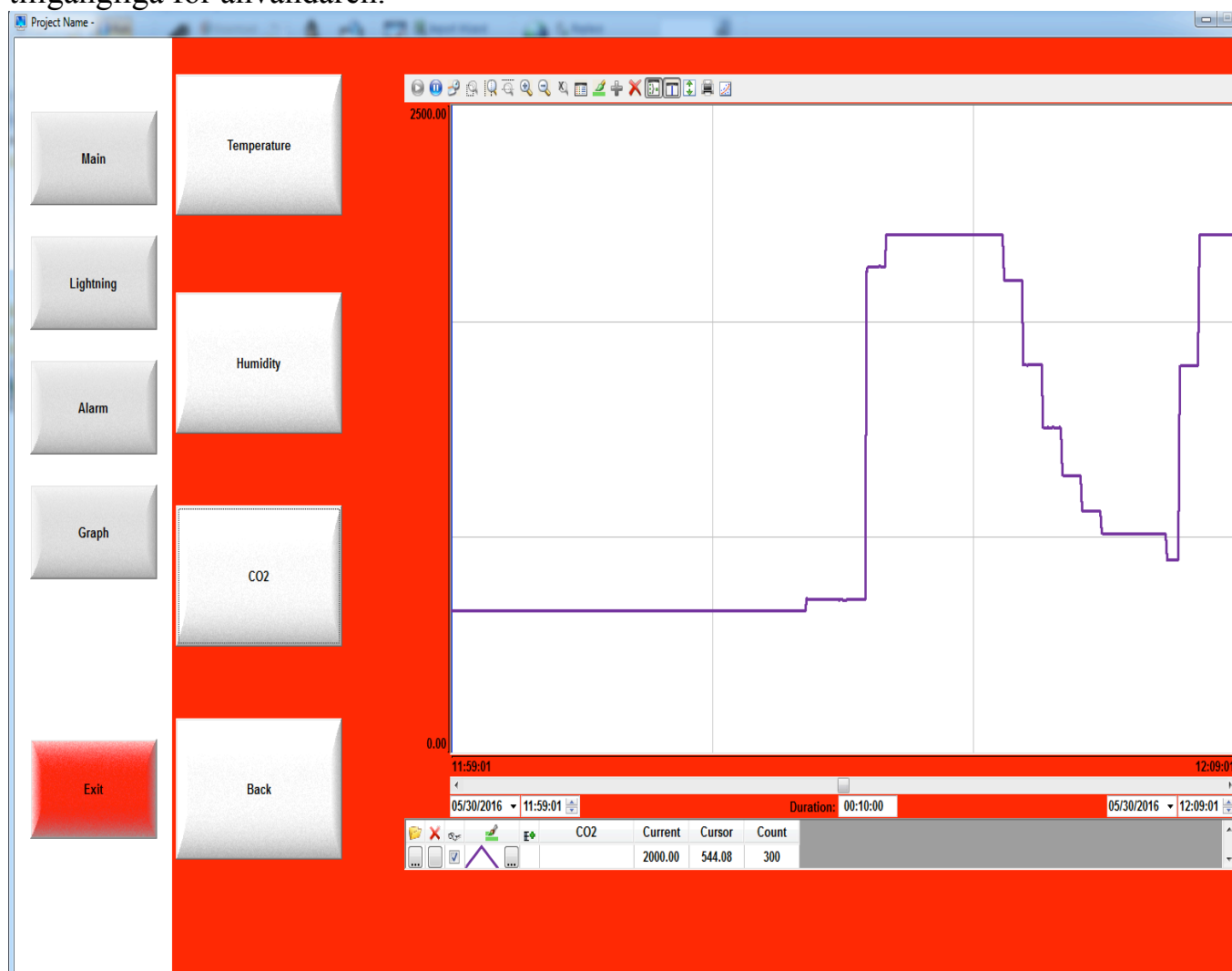


Bild 19: Graf som visar CO2-halten i rummet

## 4.4 Databasen

### 4.4.1 Konfigurering av databas

Till konfigurationen av en databas användes programmet TwinCAT Database Server Configurator. Det är ett program som används för att konfigurera en databasserver antingen lokalt i PC:n eller i CX9020. Konfigurationen gick fort och MySQL valdes som verktyg för att kommunicera med databasservern.

Bilden ovan visar hur konfigurationsprogrammet ser ut. I menyn till vänster finns ADSDevices. Där valde man vilken enhet, i detta fall CX9020, man vill lägga databasen i. Efter att ha gjort konfigurationen laddade man upp det direkt till CX9020.

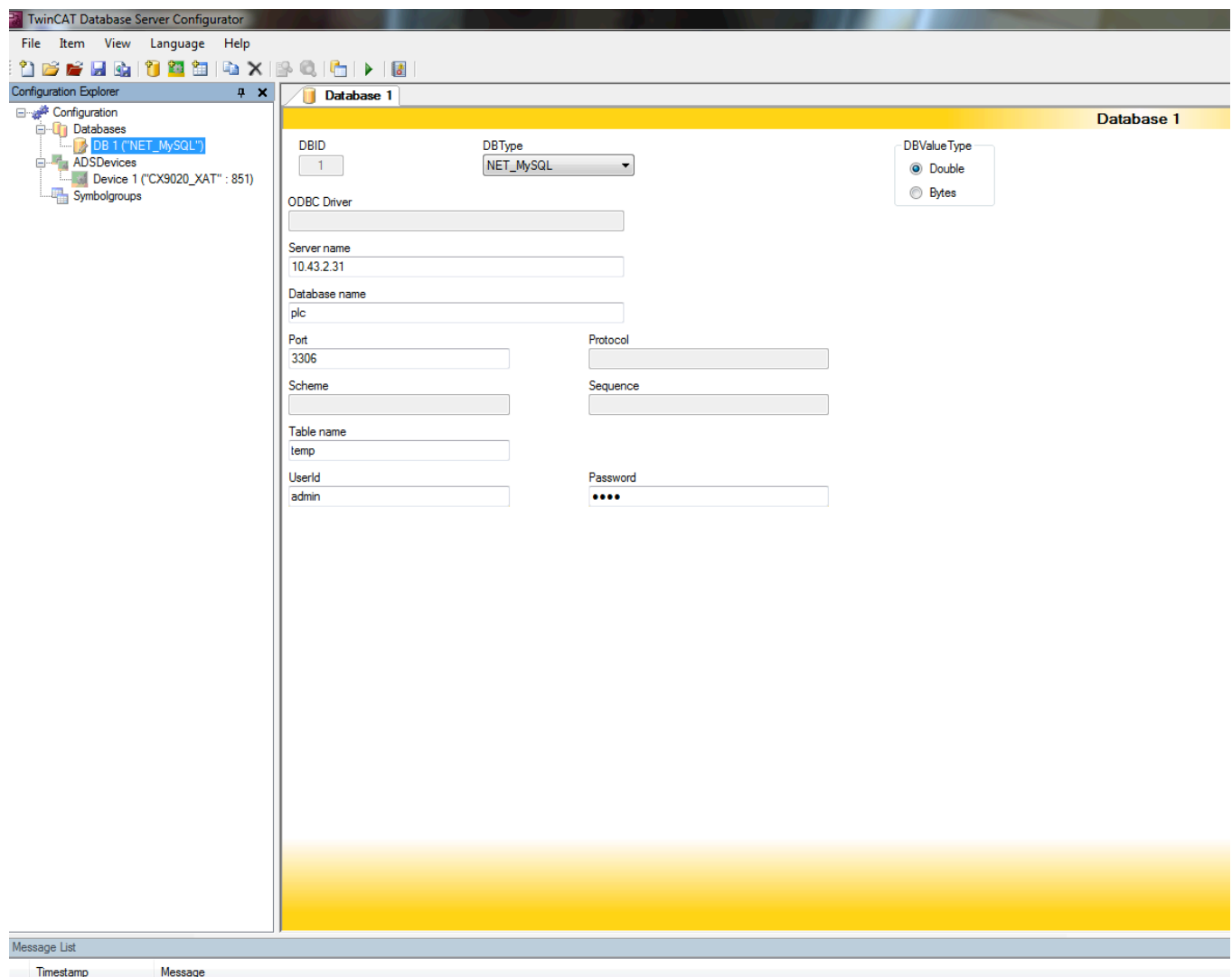


Bild 20: Konfiguration av databas

## 4.4.2 Programmeringen till databasen

Programmeringen för att kunna skicka värden till databasen var lite mer krävande jämfört med tidigare programmering. Här behövde man använda sig av tre olika funktionsblock för att skicka värdena. TwinCAT krävde att man hade en tidsstämpel till varje rad som man skickade till databasen. Här användes funktionsblock som hämtar datorns aktuella tid och skriver ut den som en sträng. Steg två var att skapa en sträng som innehöll all data från sensorn och tidsstämpeln. Till sist användes funktionsblocket som sköter själva insättningen till databasen. Databasens ID skrevs in och strängen som skulle skickas in till databasens tabell. Skärmbilden nedan visar hur vi har gjort för att skicka in sensorvärdena till databasen.

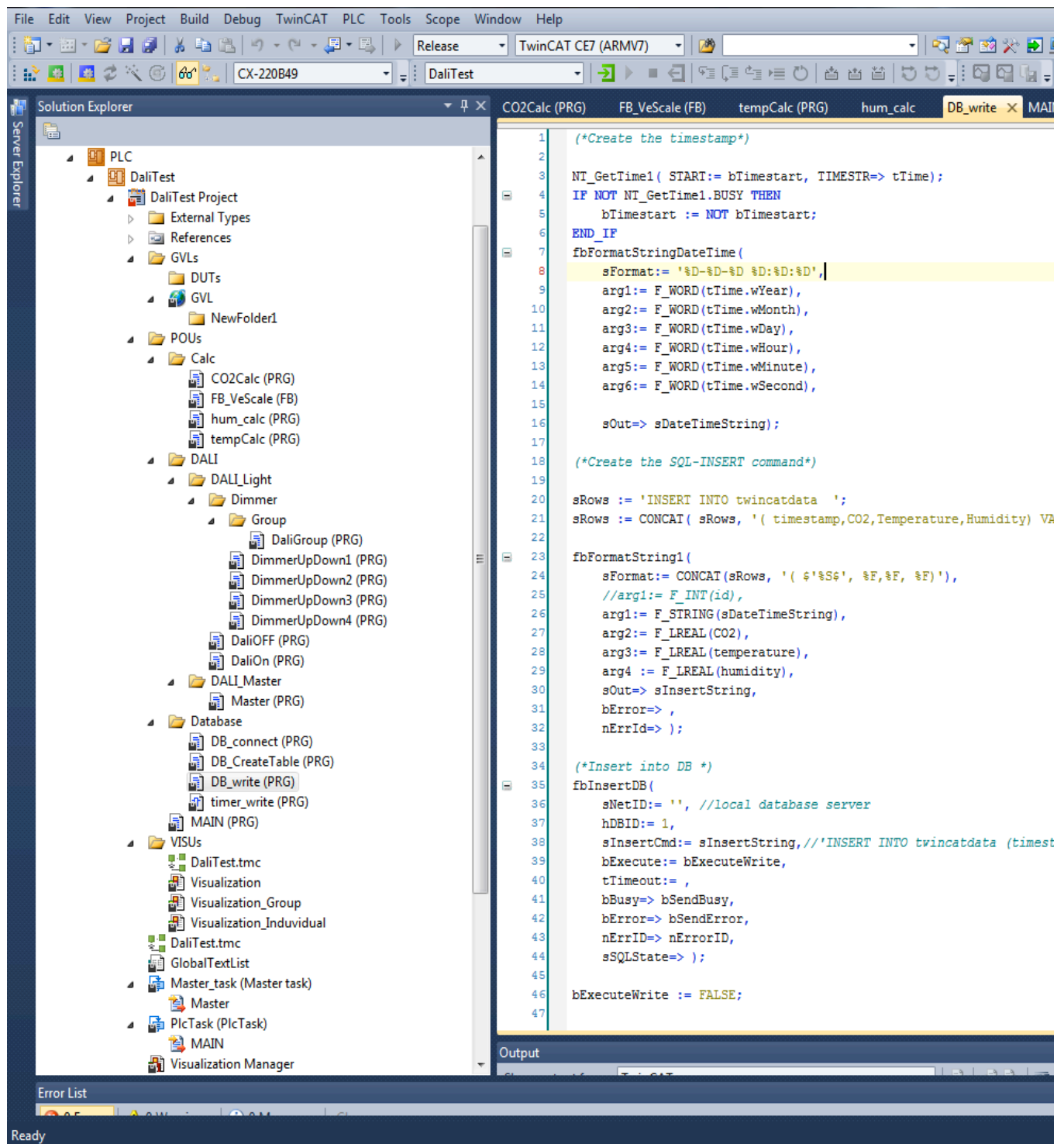


Bild 21: Skärmbild av programmet som skickar in värden till databasen.



## 4.5 OPC UA och SOA

### 4.5.1 OPC UA

Till en början användes kommunikationsprotokollet ADS för kommunikationen mellan HMI, sensor och CX9020. Beckhoffs ADS protokoll beskriver ett enhets- och fältbussoberoende interface som styr nivån av tillgång till ADS enheter. ADS bidrar dock inte till den tjänsteorienterade arkitekturen som man är ute efter. OPC UA är anpassat för tjänsteorienterad arkitektur med sin täta säkerhet och flexibilitet. Fördelarna med OPC UA är många och nedan nämns några:

- Plattformsoberoende:  
Ger en enkelhet och flexibilitet som ej funnits tidigare. OPC UA tillåter kommunikation mellan olika operativsystem(Microsoft, Mac OS X, Linux) och allt från en liten mikrocontroller till molnbaserad infrastruktur. I projektet användes en pekskärm med ett skapat HMI som kommunicerade med CX9020 med Windows CE.
- Säkert:  
Erbjuder kryptering, autentisering och granskning. OPC-UA är vänlig mot brandväggen men samtidigt väldigt säkert.
  1. Transport: Flera protokoll är definierade och man har alternativ att välja mellan.
  2. Kryptering: Meddelanden sänds säkert på 128 eller 256-bitars krypteringsnivå.
  3. Meddelandesignering: Meddelanden tas emot exakt som de skickades.
  4. Autentisering: Varje UA klient och server identifieras genom OpenSSL.
  5. Användarkontroll: Applikationer kan kräva inloggningsuppgifter för att autentisera användaren.
- Uttänjbar:  
Detta möjliggjorde att man kunde lägga till funktioner utan att de existerande funktionerna påverkades.
- Upptäckande:  
Man kunde hitta alla tillgängliga OPC-servrar i den lokala PC:n och/eller i det lokala nätet.

OPC UA ger väldigt många fördelar och känns i princip överlägset. Dock var installationen och konfigurationen av OPC UA mycket mer komplicerat än t.ex. ADS. Importen av variabler var mer krångligt och krävde att man startade om OPC UA- servern varje gång man lagt till en ny variabel. [8]

## 4.5.2 SOA

Tidigare nämndes att huvudsyftet med examensarbetet var SOA innan det avgränsades till att inte vara det alls. Projektet skulle ändå försöka ses ur ett service-synsätt där syftet med ett valfritt IT-system är att spara tid och pengar. I SOA vill man återanvända funktioner så att man inte behöver skapa nya hela tiden. Man kan skala ner arkitekturen och ha en centraliserad styrning.

I detta projekt styrs flera processer med en central styrenhet(CX9020). För kommunikation använder vi OPC UA som är en SOA.

Vilka fördelar ger SOA till en affärsverksamhet?

- Förbättrat informationsflöde
- Organisatorisk flexibilitet
- Mindre mjukvaruutveckling och lägre kostnader
- Prestandamätning
- Upptäcker säkerhetsattacker
- Konfidentialitet och integritet
- Förbättrad hanterbarhet och säkerhet

[5][6]

## 5. Resultat

Efter många veckors arbete blev projektet fullbordat. Från början såg allting väldigt rörigt ut och man såg inte hur allt skulle bli ett enda system. Det första som påbörjades var programmeringen av DALI-belysningen. Det innebar allt från att koppla nya stickproppar till armaturerna till att adressera dem och markera. Sedan lades många dagar till programmeringen av belysningen där teori och praktik krockade flera gånger. Det underlättade väldigt mycket att ha två armaturer kopplade, tillgängliga för att laborera med och se resultaten fysiskt. Finslipningen av styrningen av belysningen blev färdigt sist fast det påbörjades först.

Under tiden som det laborerades med belysningen påbörjades konfigureringen av databasen och programmeringen för skrivning till databasen. Detta var ett delmoment som var mest bekant. Inom ett par dagar var det färdigt och nu kunde man skriva data till databasen.

Därefter behövdes det värden som skulle skickas in till databasen. Sensorn kopplades till terminalen för digitala ingångar och TwinCAT tog emot råa värden som skalades om. Sensorn, databasen och belysningen var redo för användning och det som saknades var att kunna styra detta visuellt.

Indusoft användes för att skapa HMI:et. Arbetet med HMI tog en stor del av tiden eftersom det gick att finslipa på hela tiden. Resultatet av arbetet med HMI var väldigt lyckat och HMI:et kändes väldigt professionellt och lättmanövrerat. Bilderna 16-19 visar det färdiga HMI:et.

Nu kommunicerade HMI:et med CX9020 som kommunicerade med databasen, belysningen och sensorn. För att knyta ihop allting valde man att använda kommunikationsprotokollet OPC UA.

Sista veckorna användes för att finslipa alla funktioner och utseendet på HMI:et. Koden i TwinCAT kommenterades för att ge lättare förståelse och alla program fick lättförståeliga namn.

Det som för 15 veckor sen såg omöjligt ut var färdigställt. En fastighetsautomation hade gjorts med ett väldigt lyckat resultat.

Fotot nedan visar projektet där belysningen och sensorn är kopplade till CX9020.

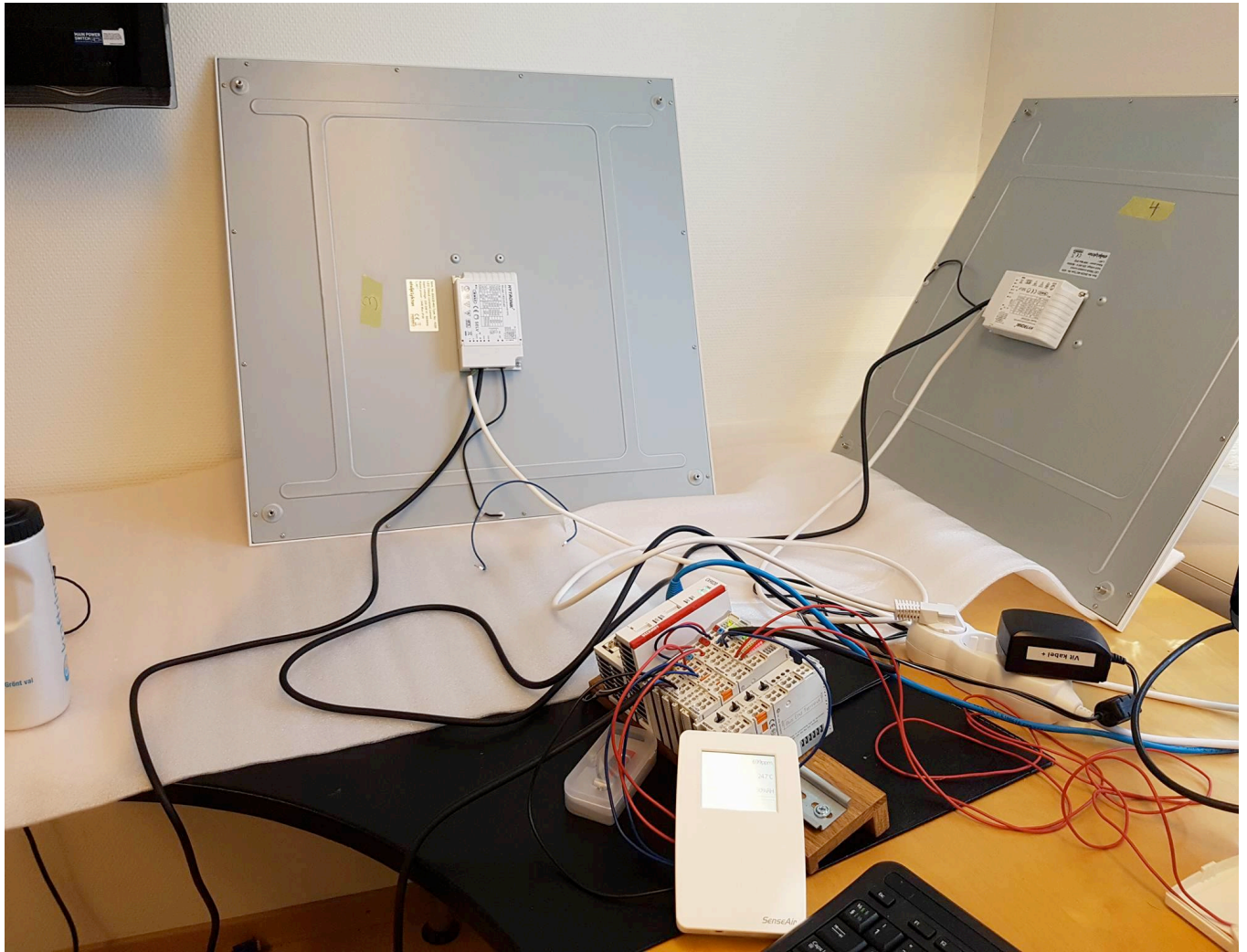


Foto på examensarbetet i slutskedet.

## 6. Slutsats

Ett kontor har automatiserats genom styrning av belysning och analys av luftkvalité. Delarna som har haft störst fokus var styrning av lamporna, mäta och logga luftkvalitén (koldioxidhalten, temperaturen och luftfuktigheten) och avslutningsvis att skapa ett snyggt HMI som binder ihop alla individuella delarna i projektet till en. Möjligheterna inom fastighetsautomation är oändliga. Idag kan man automatisera allt från dörröppningen till rullgardinerna. Avgränsningarna begränsade möjligheterna men man fick en tydlig bild i hur allt fungerar när det binds ihop.

### 6.1 Ljusstyrningen

Projektet gav en inblick på hur stor skillnad den moderna DALI-belysning kan erbjuda i stora implementationer. I projektet gav det ett behagligare ljus och bekvämare styrning via pekskärmen men hur det påverkar en affärsverksamhet energisparmässigt och ekonomiskt kunde inte analyseras. DALI är väldigt smart och möjligheterna är väldigt stora. Sedan DALI introducerades 2001 har användningen växt mycket och det beror på fördelarna som t.ex. lägre kostnader, bekvämare användning, fler funktioner och energisnålheten.

### 6.2 Mätning av luftkvalitén

Att ansluta en sensor av typen som användes i projektet till ett rum/byggnad är ett väldigt bra sätt för att förbättra miljön i rummet samt vara sparsam med energin. Till en början var det tänkt att luftkonditioneringen i rummet skulle styras av sensors värde men det var inte möjligt att utföra. Det avgränsades till att alarm aktiverades när gränserna som har satts upp nåddes i rummet. Under möten är personalen medveten om hur luften i rummet är och kan agera därefter. Man loggade dessutom all data till databasen och ritade upp grafer med det. Graferna kan användas för att titta på tidigare användningen av rummet och jämföra.

## 6.3 HMI

Användandet av gränssnitt och HMI har blivit väldigt populärt i industriella miljöer och det ger en möjlighet att följa processer och kontrollera. Med ett HMI får man en bättre insyn i processen som man ska styra och därför blir det mycket enklare och bekvämare att styra det. HMI:et som skapades var väldigt lyckat och i HMI:et såg man rummets design och kunde styra belysningen i realtid. Via OPC UA har man möjlighet att styra rummet utifrån eller hemifrån via sin telefon eller dator.

## 6.4 Sammanfattning

Examensarbetet hade 4 primära mål, implementera DALI(1.3.1), skapa ett HMI(1.3.2), analysera luftkvaliteten (1.3.3) och analysera och välja protokoll(1.3.4). Examensarbete var mycket praktiskt arbete som krävde mycket energi och långa timmar. Man valde att lägga till så mycket teoretiska moment och analys som möjligt. Till en början var planen att analysera SOA och OPC UA i stor skala men det ansågs vara svårt på grund av att tiden inte var tillräcklig för ett projekt i den storleken. Däremot känns det som att det har varit väldigt givande och man har lärt sig väldigt mycket nytt. Examensarbetet har verkligen knutit ihop det man har lärt sig i skolan, databaser, PLC-programmering etc., med nya kunskaper. Examensarbetet har visat klart och tydligt vad som väntar oss ute i arbetslivet och det har varit väldigt intressant och lärorikt.

## 7. Framtida utvecklingsmöjligheter

Under arbetets gång diskuterades möjligheterna detta examensarbete har. Det går att utveckla och göra hur stort som helst. Det fanns idéer som diskuterades och kan bli verklighet i framtiden. Nedan beskrivs några av idéerna som dök upp under tiden det arbetades med projektet.

- Automatisera projektorduken: Här hade man kunnat installera en motoriserad projektorduk som man kan styra via HMI:et. Det ger en väldigt bekväm och smidig funktion. Att kunna styra allt i ett rum med ett enda program och en pekskärm är väldigt intressant och smidigt.
- Använda LUX-mätare: En installation av en LUX-mätare hade gjort rummet väldigt avancerat och smart. LUX-mätaren hade anpassat DALI-belysningen efter hur mycket ljus som kommer utifrån och gett en behaglig nivå av ljusstyrka i rummet. Man sparar dessutom energi genom att undvika full styrka på ljuset när det inte behövs.
- Styra luftkonditioneringen: Möjligheterna att ansluta sig till luftkonditioneringen i rummet och styra den saknades i nuläget. Byter man ut den och ersätter med något nytt så skapar det nya möjligheter. Med återkoppling från sensorn styrs luftkonditioneringen automatiskt vilket skulle ge en mycket bekväm miljö.

Möjligheterna för framtida utveckling är många och automation i det vardagliga livet blir allt mer vanligt.

# Akronymlista

ADS	The Automation Device Specification
CX9020	Beckhoff's Embedded PC
DALI	Digital Addressable Lightning Interface
HMI	Human- Machine Interface
IEC	Internation Engineering Consortium
LUX	SI unit for illuminance
OPC UA	OPC Unified Architecture
PLC	Programmable Logic Controller
PPM	Parts Per Million
RTU	Remote Terminal Unit
RH	Relative Humidity
ST	Structured Text
SOA	Service-Oriented Architecture
SCADA	Supervisory Control And Data Acquisition



# Referenser

- [1] [https://en.wikipedia.org/wiki/Beckhoff\\_Automation](https://en.wikipedia.org/wiki/Beckhoff_Automation) (hämtad 2016-04-28)
- [2] <http://www.beckhoff.se/> (hämtad 2016-04-28)
- [3] <http://www.opengroup.org/soa/source-book/soa/soa.htm> (hämtad 2016-05-06)
- [4] [https://sv.wikipedia.org/wiki/Tj%C3%A4nsteorienterad\\_arkitektur](https://sv.wikipedia.org/wiki/Tj%C3%A4nsteorienterad_arkitektur) (hämtad 2016-05-06)
- [5] [http://www.service-architecture.com/articles/web-services/service-oriented\\_architecture\\_soa\\_definition.html](http://www.service-architecture.com/articles/web-services/service-oriented_architecture_soa_definition.html) (hämtad 2016-05-06)
- [6] [https://www.opengroup.org/soa/source-book/soa/soa\\_features.htm](https://www.opengroup.org/soa/source-book/soa/soa_features.htm) (hämtad 2016-05-06)
- [7] <http://www.logimethods.com/images/diagram-soa.gif> (hämtad 2016-05-06)
- [8] <https://opcfoundation.org/about/opc-technologies/opc-ua/> (hämtad 2016-05-07)
- [9] <https://www.novotek.com/sv/l-sningar/kepware-opc-kommunikationsplattform/opc-och-opc-ua-en-foerklaring> (hämtad 2016-05-07)
- [10] [http://www.ni.com/cms/images/devzone/tut/Figure\\_1\\_20120304162228.jpg](http://www.ni.com/cms/images/devzone/tut/Figure_1_20120304162228.jpg) (hämtad 2016-06-05)
- [11] <http://www.dali-ag.org/discover-dali.html> (hämtad 2016-05-11)
- [12] [http://www1.clipsal.com/dalicontrol/about\\_dali/what\\_is\\_dali](http://www1.clipsal.com/dalicontrol/about_dali/what_is_dali) (hämtad 2016-05-11)
- [14] <http://www.dpstele.com/scada/where-is-used.php> (hämtad 2016-05-15)
- [15] <https://en.wikipedia.org/wiki/SCADA> (hämtad 2016-05-15)
- [16] [https://en.wikipedia.org/wiki/SCADA\\_-\\_/media/File:SCADA\\_schematic\\_overview-s.svg](https://en.wikipedia.org/wiki/SCADA_-_/media/File:SCADA_schematic_overview-s.svg) (hämtad 2016-05-15)
- [17] [http://download.beckhoff.com/download/document/ipc/embedded-pc/embedded-pc-cx/cx9020\\_hwen.pdf](http://download.beckhoff.com/download/document/ipc/embedded-pc/embedded-pc-cx/cx9020_hwen.pdf) (hämtad 2016-05-15)
- [18] <http://www.senseair.com/products/wall-mount/tsense/> (hämtad 2016-05-15)
- [19] <https://sv.wikipedia.org/wiki/Databas> (hämtad 2016-05-15)

[20]

[http://infosys.beckhoff.com/english.php?content=../content/1033/tcplclibdali/html/tcplclibdaliiv2\\_communication.htm&id=](http://infosys.beckhoff.com/english.php?content=../content/1033/tcplclibdali/html/tcplclibdaliiv2_communication.htm&id=) (hämtad 2016-05-17)

[22]

[http://infosys.beckhoff.com/english.php?content=../content/1033/tcplclibdali/html/tcplclibdaliiv2\\_communication.htm&id=](http://infosys.beckhoff.com/english.php?content=../content/1033/tcplclibdali/html/tcplclibdaliiv2_communication.htm&id=) (hämtad 2016-05-17)

[23]

<http://www.artisticlicence.com/WebSiteMaster/User%20Guides/the%20dali%20guide.pdf> (hämtad 2016-05-26)

