

Datahantering i mindre fabriker



Adrian Olbers

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



LUNDS
UNIVERSITET
Lunds Tekniska Högskola

Datahantering i mindre fabriker

Adrian Olbers

LTH Ingenjörshögskolan vid Campus Helsingborg

Skriven av: Adrian Olbers

Handledare (LTH): Alma Orucevic-Alagic

Handledare (Företaget): Gustav Jansson

Examinator: Ulf Jeppsson

Mail: adrianolbers@gmail.com

Utförd: Våren 2022

©Copyright Adrian Olbers

LTH Ingenjörshögskolan vid Campus Helsingborg
Lunds universitet
Box 882
251 08 Helsingborg

LTH School of Engineering
Lund University
Box 882
SE-251 08 Helsingborg
Sweden

Tryckt i Sverige
Media-Tryck
Biblioteksdirektionen
Lunds universitet
Lund 2022

Sammanfattning

Arbetet har utförts på Balsgård FoodTech AB i Kristianstad som ägs och drivs av Mårtensson Consulting som också har önskat få detta arbete utfört. Arbetet har syftat till att undersöka om det är möjligt att i mindre fabriker samla in och sammanställa data för att kunna presentera och använda sig av. Detta har utförts genom att studera fabriken implementerade system och variabler samt inspektera en större fabriks redan implementerade databaskommunikation med respektive PLC.

Arbetet kan delas in i tre olika faser som genomgåts för att komma till ett färdigt system, de är **simulering, integrering och implementering**. I simuleringen har de olika programmen programmerats i syfte att fungera var för sig, i integreringen har de olika systemen integreras så att kommunikationen fungerar i ett simulerat tillstånd och implementeringen där ett simulerat fungerande system har implementerats på Balsgård FoodTech AB. Projektet resulterade i att data kunde kommuniceras upp till databasen och data kunde presenteras med hjälp av Power-BI på ett interaktivt och intuitiv sätt.

Nyckelord: SQL-databas, PLC, Power-BI, Optima, VMWare

Abstract

The work has been carried out at Balsgård FoodTech AB in Kristianstad, which is owned and run by Mårtensson Consulting, which has also wished to have this work carried out. The work was aimed to investigate whether it is possible in smaller factories to collect and compile data in order to be able to present and use it. This has been done by assessing the factory's already implemented systems and variables and by inspecting a PLC solution implementation at another factory.

The work can be divided into three different phases, **simulation, integration and implementation**. In the simulation phase the programs were programmed in order to work separately, in the integration the different systems were integrated so that the communication worked in a simulated state and in the implementation phase the simulated integrated system was implemented at Balsgård FoodTech AB. The project resulted in a solution where the data was communicated to the database and the data was presented using PowerBI in an interactive and intuitive way.

Keywords: SQL-database, PLC, Power-BI, Optima, VMWare

Förord

Den här rapporten har avslutat min treåriga utbildning på Lunds Tekniska Högskola där jag har läst programmet Elektroteknik med Automation. Arbetet har tagits fram efter ett projektarbete under hösten 2021 inom kursen Projekt i Automation där jag gjorde projektet hos Mårtensson Consulting. Jag vill med detta förord också passa på att tacka Mårtensson Consulting för möjligheten att få göra examensarbetet hos dem och för deras hjälpsamma inställning under arbetets gång.

Terminologi

- Brandvägg: En uppsättning regler som styr vad och vem som får kommunicera genom eller till routern, servern eller datorn
- Databas: En samlingsplats information och som är strukturerad enligt förkonfigurerade parametrar
- IP: Internet Protocol
- PLC: Programmable Logic Controller
- SQL: Structured Query Language
- Optima: Egenutvecklad applikation av Mårtensson Consulting
- Profinet: En industristandard för att kommunicera data över det industriella nätet
- Pastör: En maskin som desinficerar en vätska och vanlig i livsmedelsindustrin
- Queries: Frågor som kan skickas till databasen moderera databasen
- Simulering: Ett sätt att återskapa verkligheten i en kontrollerad testmiljö
- Server: Ett datorsystem vars syfte är att betjäna andra datorer och system
- Subnät: Samling av ip adresser som ligger på samma lokala instans
- Subnätmask: Definierar vilken del av ip adressen som representerar subnät
- Smart givare: En givare som har någon form av egen beräkningskraft
- VMWare: Applikation för att köra en virtuell dator i datorn
- WWW: World Wide Web, globala internet

Innehållsförteckning

1 Inledning	1
1.1 Bakgrund	2
1.2 Syfte	2
1.3 Avgränsningar	2
1.4 Problemformulering	3
1.5 Mål	3
1.6 Verktyg	3
1.7 Källor	4
2 Teknik	5
2.1 SQL-databas	5
2.2 PLC	6
2.3 Optima	7
2.4 Power BI	8
2.5 Kommunikation	9
2.5.1 Ethernet	9
2.5.2 Industrial Ethernet	10
2.5.3 Profinet	11
2.6 VMWare	11
3 Metod	13
3.1 Simulering	13
3.1.1 PLC	13
3.1.2 Databas	14
3.1.3 Power BI	16
3.2 Integrering	17
3.2.1 PLC till databas	17
3.2.2 Databas till Power BI	19
3.3 Implementering	19
3.3.1 VMWare	19
3.3.2 IP adressering	20
3.3.3 Brandvägg	20
3.3.4 Power BI	21
4 Resultat	25
4.1 PLC	25
4.2 Databas	29
4.3 Power BI	31

5 Slutsats	35
5.1 Framtida utvecklingsmöjligheter	36
6 Referenser	37
6.1 Litteratur	37
6.2 Manualer	39
6.3 Bilder	39
Bilagor	40

1 Inledning

Industrierna blir idag mer och mer digitaliserade, och det installeras fler smarta givare som genererar mer och mer data [7]. Enligt rapporten är ett problem att det blir så mycket information att det blir svårt att strukturera och samla informationen lokalt, vilket tidigare har varit det vanliga. Rapporten påvisar att tillverkningsföretagen har insett att det finns stora vinster i att ta tillvara på och använda informationen, då det kan leda till en större planerbarhet och kontroll över fabriken. Rapporten styrker också att det är av denna anledning som det idag sammanställs mycket data vilket är en stor del av industri 4.0.

Arbetet undersöker en fabrik som är i en fas där man har mycket data att tillgå i form av givarvärden med mera, men där de inte samlas i en databas utan endast används på lokal nivå av en Programmable Logic Controller (PLC) för att reglera processen. Det innebär att det är svårt att få en överblick i hur produktionen går i fabriken då ingen information finns att tillgå i form av centraliserade grafiska presentationer. Därför ämnar detta arbete att skapa en databas som ska samla in data från de olika processerna i fabriken för att sammanställa dessa och presenteras dem för att ge en bättre översikt av fabriken processer.

1.1 Bakgrund

Bakgrunden till att Mårtensson Consulting efterfrågade detta arbete är att de ser att många kunder som driver mindre anläggningar är i ett stadi där de har information att tillgå i sina processer, i form av givardata med mera, men att den inte är samlad i en databas. Då det är något Mårtensson Consulting redan har viss erfarenhet av så vill de utveckla och skapa en visningsplats för hur deras system kan fungera i mindre fabriker. Mårtensson Consulting driver i dagsläget en anläggning som är ett testcenter för företag som vill testa hur deras nya recept på juicer fungerar i serieproduktion, och för företag som vill producera en begränsad mängd juice, utan att behöva investera i nödvändiga maskiner. Det är med denna bakgrund som Mårtensson Consulting vill göra denna plats till ett sorts democenter för lagring och visualisering av data i mindre fabrik då Balsgård är en bra representation av en mindre fabrik med ovan nämnda problem.

1.2 Syfte

Arbetet syftar till att skapa ett system som samlar in data från fabriken och att sammanställa dessa på ett intuitivt och interaktivt sätt så att de fyller en funktion och ger en bättre överblick av status i fabriken. Det är meningen att operatörerna enkelt ska kunna se och interagera med denna data från en dator.

1.3 Avgränsningar

De avgränsningar som finns för arbetet är att systemet endast utvecklas för att fungera i produktionssystem med Siemens PLC. För databasen och presentationen av data från databasen kommer ett Windows operativsystem användas och mjukvaran kommer inte utvecklas för andra operativsystem, dock är det möjligt att visa den visualiserade datan på andra operativsystem så länge respektive operativsystem har åtkomst till en webbläsare. Detta för att fokusera på att utveckla ett system med hänsyn till den tekniska funktionen och inte dess kompatibilitet då den kan anpassas efteråt.

1.4 Problemformulering

Examensarbetet kommer att besvara dessa frågor:

1. Hur kan data kommuniceras upp till en databas från en PLC?
2. Hur lagras data på ett strukturerat effektivt sätt i en databas?
3. Hur plockas data ut för visualisering?

1.5 Mål

Målet med arbetet är att skapa ett databassystem som PLCn i fabriken kan skicka upp information till i form av produktionsdata som temperaturer, flöden, batch-nummer osv. Denna data ska lagras på ett strukturerat sätt med ett lämpligt val av databas. Datan ska sedan presenteras med hjälp av applikationen Power BI för att ge en översikt över den väsentliga produktionsdatan. Projektet ska genomgående ta hänsyn till att systemet ska vara tänkt att fungera i mindre industrier då det är den ursprungliga tanken med arbetet.

1.6 Verktyg

Till arbetet kommer Mårtensson Consulting bistå med samtliga programvaror för att kunna utföra arbetet.

- Siemens Tia Portal användas för att programmera PLCn, mjukvaran används för att programmera Simatic S7 PLCer vilken också är typen som sitter i produktionen.
- Optima, vilket är en egenutvecklad applikation av Mårtensson Consulting. Programmet kommer att användas för att översätta datan från PLCn till databasen.
- Power-BI kommer användas för att presentera datan som ligger i databasen. Datorn som använts för att köra dessa programvaror har varit privat och ägts av studenten som gjort arbetet då det ansågs vara smidigast.
- VMWare kommer att användas för att kunna köra TIA portal och för att kunna kommunicera mellan kontorsnätverket och produktions nätverket.

1.7 Källor

Källor som används under arbetets gång är erfarenheter från anställda på Mårtensson Consulting, egna erfarenheter från tidigare företag, från nuvarande utbildning, från tillverkarna av mjukvarorna och källor på nätet vilket kan hittas i kapitel 7.

Beslut som har tagits under arbetets gång har i vissa fall inte vägts mot konkurrerande alternativ, då företaget redan bestämt sig för att vissa system skulle ingå och vid vissa tillfällen är det dessutom det enda alternativ som fungerar med de redan implementerade systemen. Motivering för arbetssättet var att arbetet inte skulle bli för omfattande. Det positiva med tillvägagångssättet kan vara att implementationen får med sig lösningar från erfarenheter av tidigare implementationer, då valen har gjorts med hjälp av många olika erfarenheter som förhoppningsvis grundlagts utifrån en bred kunskap inom området.

2 Teknik

Kapitlet kommer beskriva tekniken som använts genom arbetet i detalj. De olika delarna i kapitlet kommer ta upp mjukvara och hårdvara som krävs för denna implementation och på vilket sätt respektive instans bidrar till helheten.

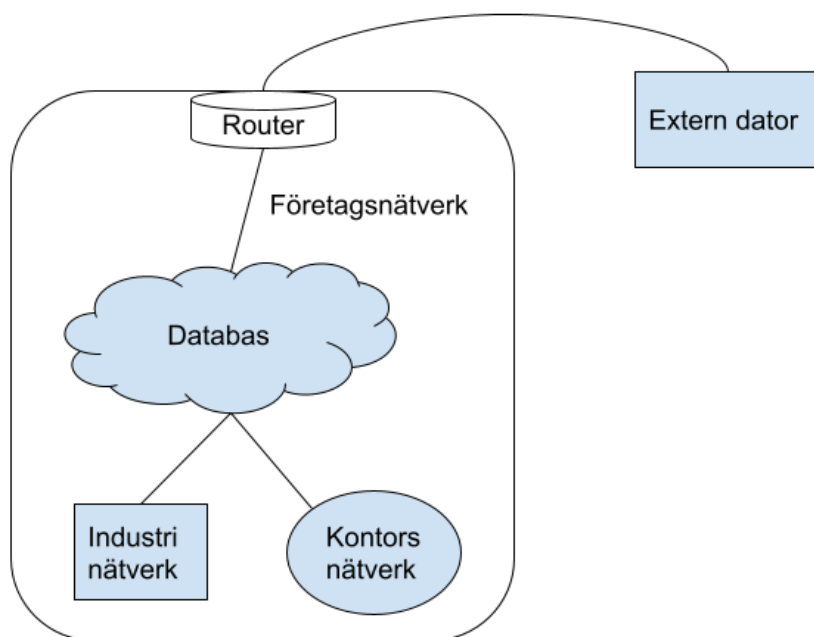
2.1 SQL-databas

Databasen som används i det här arbete är Microsofts SQL databas som heter SQL Server Express. SQL står för Structured Query Language och är ett språk som är brett använt av många olika implementationer av databaser som till exempel SQL Server, MySQL och Oracle [4]. Databasen struktureras med hjälp av relationer vilket ger en struktur som sedan frågor (queries) kan hämta information från enligt [4]. Denna kod kan se ut på följande vis:

```
SELECT *  
FROM Table;
```

Ett exempel som visar hur koden kan skrivas för att hämta all data från databordet (table) med namnet "Table".

Det finns många sätt en databas kan implementeras på men det som alltid behövs är någon typ av serverteknik, till exempel en dator för att köra databasen på. Databasen är normalt implementerad i toppskiktet av det nätverket den är uppkopplad på. Med det menas förenklat att databasen är uppkopplad så att alla enheter i det aktuella nätverket ska komma åt den. Om databasen befinner sig i ett lokalt nätverk så kan det finnas möjlighet att koppla upp sig mot databasen utanför det lokala nätverket, dock med säkerhetsåtgärder för att inte släppa in obehöriga. En visualisering av en typisk implementation av en databas i en fabrik kan se ut enligt figur 2.1.



Figur 2.1 Exempel på hur en databas kan implementeras på ett företag.

Var en databas kopplas upp skapar olika förutsättningar, om den är inom ett lokalt nätverk eller utanför det lokala nätverket. Utifrån förutsättningarna så kan det behöva skapas villkor i brandväggar och andra säkerhetsinstanser, som satts upp i nätverket, för att upprätta kommunikation med databasen. Stor vikt läggs på cybersäkerhet vid implementeringen av en databas då den är ett vanligt mål att vilja erövra vid hackingattacker.

2.2 PLC

PLC (Programmable Logic Controller) är en vida använd dator för industriellt bruk. Den uppfanns på sent 1960-tal för att ersätta den då dominerande relästyrningen inom bilindustrin vilket var en mer simpel elstyrning av komponenter. Det som särskiljer en PLC från andra datorer är robustheten, då en PLC ska vara byggd för att kunna gå utan omstart under en lång tid. Anledningen till detta är att till skillnad från en vanlig server eller dator där det inte är lika kritiskt om den fryser och kräver en omstart så kräver industrin det motsatta. I fabrikers fall är en omstart något som kostar pengar i form av förlorad produktion och måste därför undvikas.

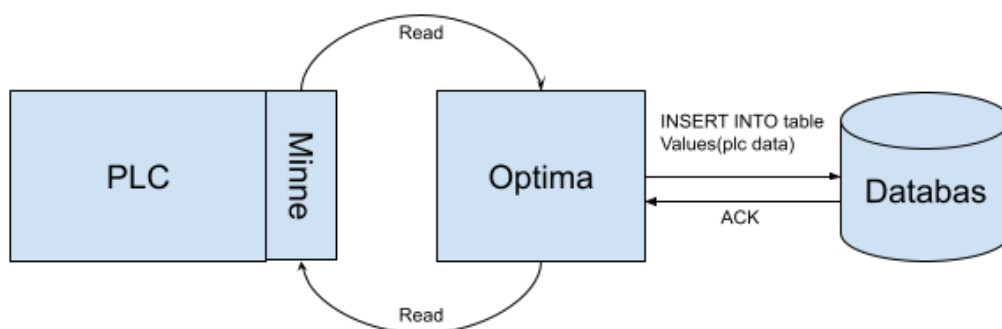
En PLC kan ses som hjärnan lokalt i en produktionsprocess. En typisk implementering av PLC är till exempel där en tank ska fyllas med olika vätskor, då styr PLCn de olika pumparna in till tanken och håller samtidigt koll på blandningen [5].

Exempel på tillverkare av PLCer:

- Siemens
- Mitsubishi
- ABB
- Allen-Bradley

2.3 Optima

Optima är en egenutvecklad mjukvara av Mårtensson Consulting. Ett problem när databaser ska implementeras för att samla data från en PLC så går det inte att skicka direkta queries från PLCn till databasen, då PLCn inte är kompatibel för det. Därför krävs det en mellanhand mellan PLCn och databasen som tolkar PLCns data och lägger in den i databasen. Det är denna uppgift som Optima har. Mer tekniskt så går programmet in i PLCn och läser av en förkonfigurerad minnesarea. När programmet har gjort det så meddelar den PLCn att värdet är läst och då vet PLCn att den datan finns i databasen och PLCn raderar då datan. Optima tar sedan värdet och kan skapa insert i SQL databasen. Det görs med hjälp av programspråket C# integrerat med Siemens bibliotek S7.Net för att läsa av PLCn. Figur 2.2 visar hur Optima opererar.

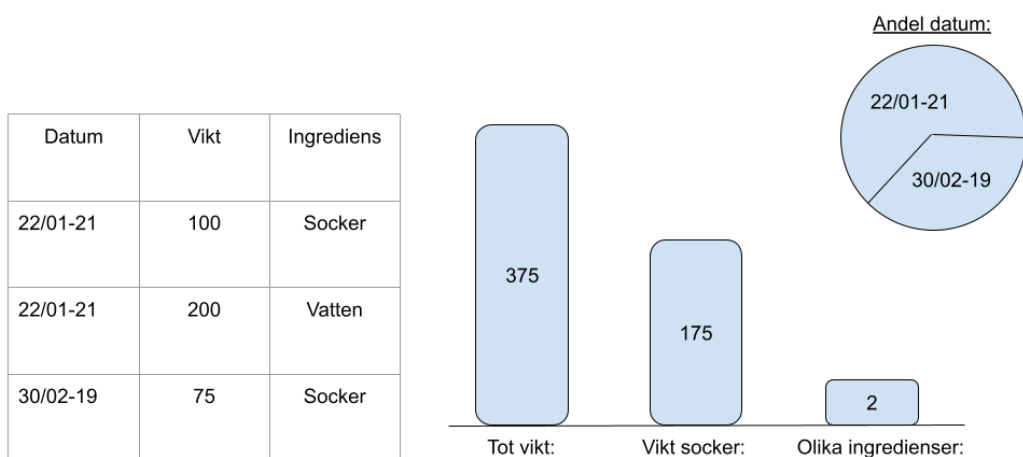


Figur 2.2 Visuellt representation av hur Optima opererar och interagerar med PLCn och databasen.

2.4 Power BI

Power BI är ett datorprogram som skapades av Ron George under namnet Projekt Crescent och som senare lanserades under namnet Power BI i september 2013. När mängder av data lagras i en databas så ska datan också användas på något sätt. Det är här som Power BI kommer in i bilden. Microsoft har skapat verktyget för att till exempel kunna skapa rapporter från det senaste dygnet eller visualisera live data både för produktionspersonal och tjänstemän med mera. Power BI applikationen kopplas till databasen där det väljs ut vilken data som är väsentlig att presentera och på vilket sätt. Power BI används på detta sätt då det är svårt att få en överblick av datan genom att inspektera den råa lagrade datan i databasen [6].

I figur 2.3 nedan exemplifieras hur datan visualiseras.



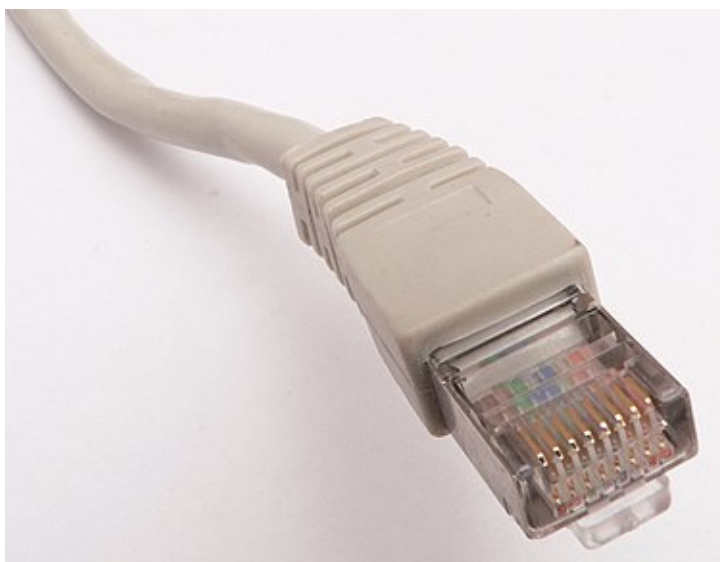
Figur 2.3 Tabellen ska representera databasens råa data och staplarna ska representera hur Power-BI kan representera den datan på olika sätt.

2.5 Kommunikation

Kommunikation i detta arbete syftar på hur olika applikationer och datorer kommunicerar med varandra. Det som definierar hur en viss kommunikation agerar och vilket användningsområde den avses att användas i kallas kommunikationsprotokoll. Kommunikationsprotokollen används och är ofta anpassade för specifika användningsområden då det skiljer sig mellan olika typer av kommunikation och vilken typ av miljö de brukas i [1].

2.5.1 Ethernet

Ethernet är ett samlingsbegrepp för olika trådbundna kommunikationer mellan enheter. Det är en typ av kommunikation som används inom ett lokalt nätverk mellan de stationära enheterna med hjälp av Ethernet kablar, se figur 2.4. En av fördelarna med Ethernet är hastigheten och robustheten till skillnad från till exempel WiFi som lätt kan störas av väggar och annat samt inte uppnå lika höga hastigheter. En nackdel med Ethernet är att den är trådbunden. Det medför att portabiliteten för en enhet som är ansluten till ett nätverk är näst intill obefintlig.



Figur 2.4 visar en rj45 kontakt (ethernet kabel) [14]

2.5.2 Industrial Ethernet

Industrial Ethernet har blivit en undergrupp till Ethernet då kommunikation inom industrin ofta ställer större krav på snabbhet och robusthet. Det finns många olika typer inom familjen industrial Ethernet, bland annat EtherCAT, EtherNet/IP och Profinet.

2.5.3 Profinet

Profinet är en standard inom familjen för Industrial Ethernet vilket är en grupp som är anpassad för industriell användning. Profinet är en av de mer använda kommunikationsprotokollen idag. Den utvecklades och togs fram som en uppdatering av Profibus. Profinet är uppbyggd på mycket liknande sätt som en vanlig Ethernet kabel. Det går också att använda en Profinet kabel som vanlig Ethernet kabel. Det har många fördelar som att koppla in sig direkt i ett nätverk med Profinet moduler med en vanlig dator, vilket underlättar till exempel felsökning. En annan fördel med Profinet gentemot dess föregångare Profibus är att det går att sätta upp en kommunikationsring. Det medför att om en modul går sönder eller en kabel går sönder så drabbar det inte resten av modulerna då det är en ring och alla kan fortfarande prata med alla.

2.6 VMWare

VMWare är ett bolag inom Dell Technologies som skapat och utvecklar mjukvaran för VMWare. Det är ett program som kan skapa en virtuell dator inom ett befintligt operativsystem. Det möjliggör att man på ett specifikt operativsystem kan köra ett annat operativsystem, till exempel kan man köra Windows 7 på en Windows 10 maskin. Det fungerar så att programmet startas, sedan konfigurerar användaren programmet för hur mycket minne och datorkraft det får använda. Sedan emulerar programmet en specificerad fil med ett operativsystem i sig och det öppnar ett fönster vilket visar hur en dator startar upp.

En stor fördel med VMWare är att då vissa äldre program inte ännu är kompatibla med nyare versioner av till exempel Windows så tillhandahåller VMWare en lösning för bakåtkompatibilitet. VMWare är praktisk att använda i Mårtensson Consultings fall då de inte vill knyta en dator till en licens då denna troligen kommer bytas ut inom ett par år. Då kan licensen knytas till den virtuella datorn och på så sätt enkelt flyttas till en annan dator. [\[8\]](#)

3 Metod

I det här kapitel kommer metoden för de olika momenten i examensarbetet beskrivas. Det kommer innefatta hur tillvägagångssättet har sett ut genom examensarbetets gång.

Arbetet startades med att få tillgång till samtliga programvaror som behövdes för att kunna utföra arbetet. Till en början så studerades dessa programvaror med hjälp av användarmanualer och Internet. För att kunna implementera system så beviljades tillgång till en annan fabriks PLC kod, där ett liknande system redan var implementerat och kunde studeras för att få en grund att stå på. Efter det så inspekterades koden i det redan implementerade PLC systemet på Balsgård.

3.1 Simulering

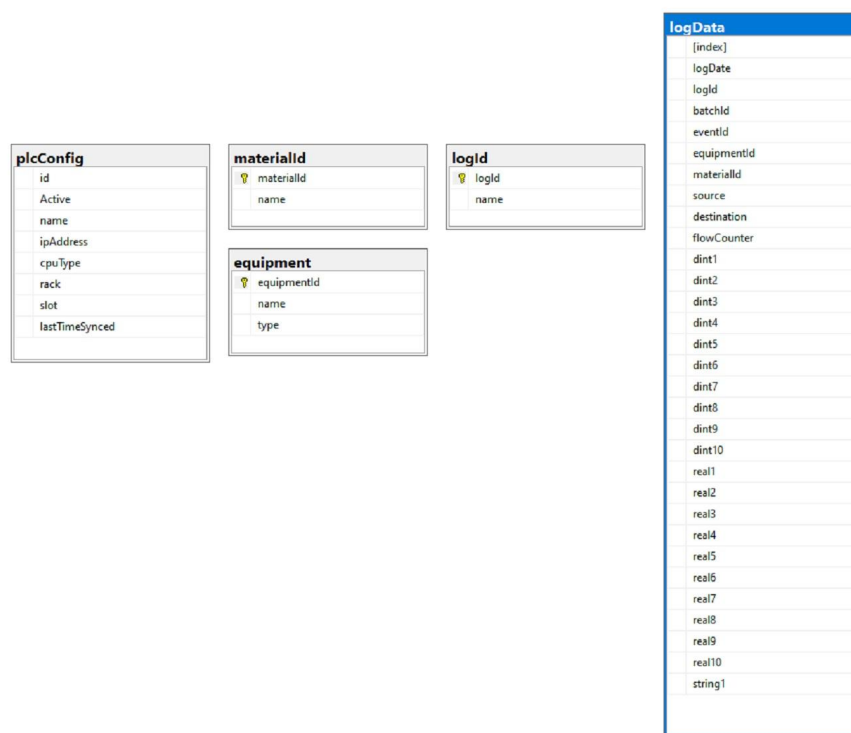
Då företaget Balsgård FoodTech AB ligger relativt långt ifrån studieorten Helsingborg så insågs det tidigt att arbetet var tvunget att kunna bedrivas och tester utföras utanför fabriken. Med detta som bakgrund så ansågs det lämpligt att börja med att hitta sätt att bygga upp systemet virtuellt. För att kunna simulera fabriken så gavs det tillgång till en licens för Siemens S7 PLC-SIM V3.0 [\[9\]](#) vilket är en simuleringsmiljö för PLCer. De andra programvarorna kunde köras på utvecklingsdatorn utan vidare licenser.

3.1.1 PLC

För att kunna simulera PLCn så ansågs det mest lämpade verktyget vara Siemens mjukvara S7 PLC-SIM. Då mjukvara är utvecklad av Siemens så fungerar den mycket väl mot programmeringsprogrammet för PLCn (Siemens TIA portal) som också används för programmering av PLCn i fabriken på Balsgård. PLC-SIM har en funktion som skapar en server som driver en PLC som i teorin fungerar på samma sätt som en verklig PLC(digital tvilling). För att köra mjukvaran så användes VMWare då det var på detta sätt som Mårtensson Consulting kunde tillhandahålla licensen.

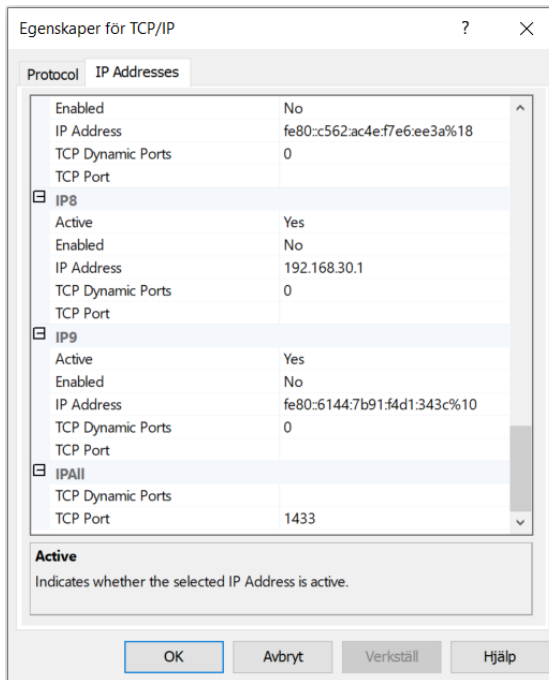
3.1.2 Databas

För att välja databas så gjordes ingen ytterligare undersökning. Eftersom Mårtensson Consulting har använt sig av Microsoft SQL-Express tidigare ansågs det vara bäst att fortsätta med denna även här då företaget redan hade kunskap om mjukvaran. Detta underlättar också för framtida uppdateringar och utveckling. Databasen lämpar sig också bra för utveckling då det är gratis lagring upp till 10GB vilket är mer än tillräckligt för projektet. Databasen kan också enkelt konfigureras på en vanlig dator vilket underlättar utvecklingen, att undvika byta dator för att modifiera databasen och att inte behöva sätta upp kommunikation över Internet utan endast kommunicera lokalt applikationer emellan. För att Optima skulle kunna kommunicera data till databasen så behövdes en viss struktur och inställningar konfigureras. Detta var något som inte kunde påverkas då Optima krävde denna struktur. Databasen behövde tabeller enligt figur 3.1.

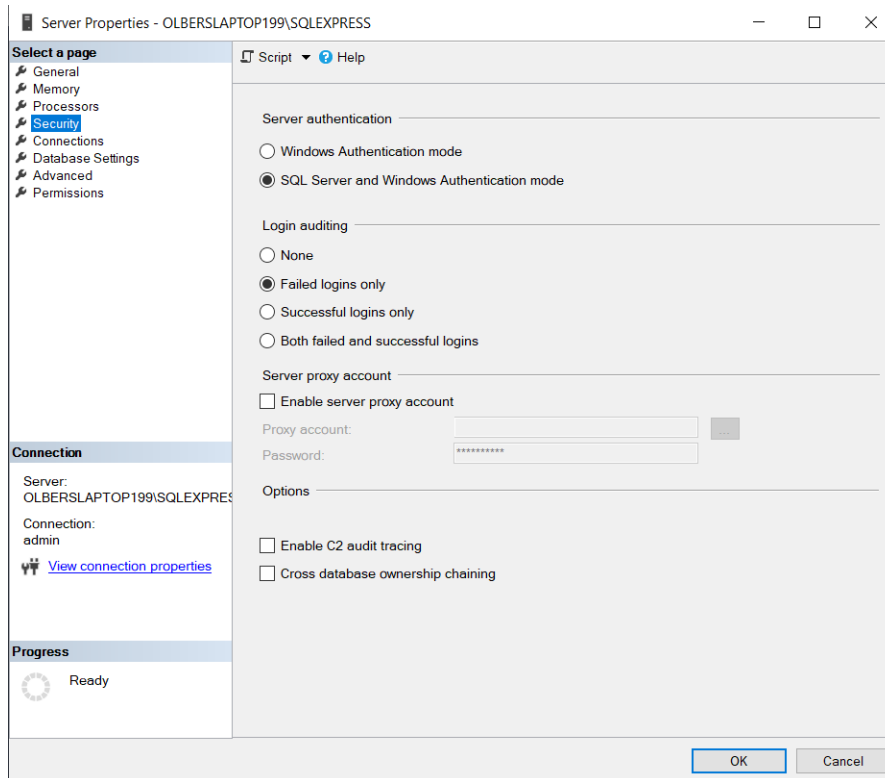


Figur 3.1 Visar tabeller på hur databasen byggdes upp för att kunna mot ta information.

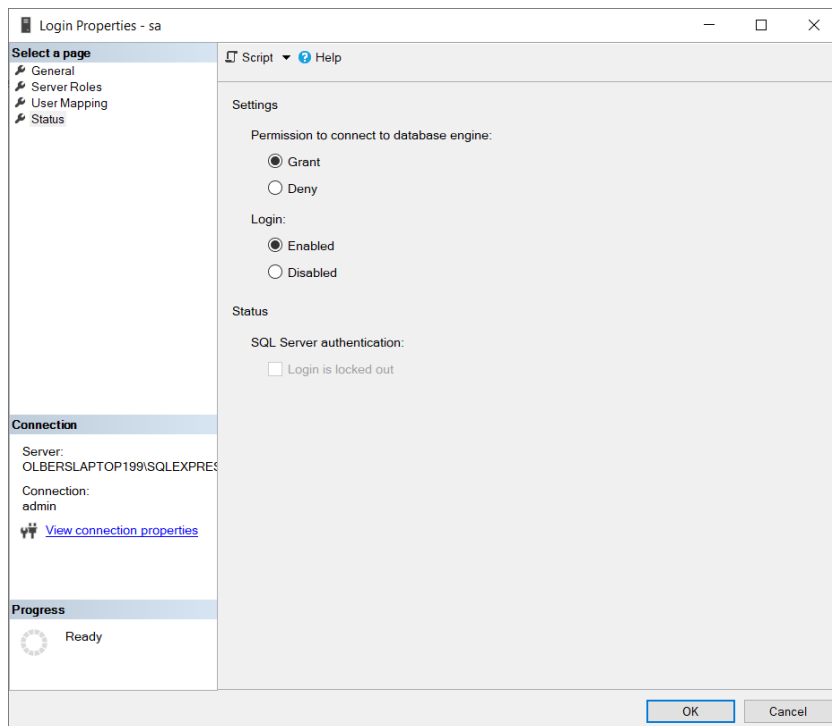
Inställningar gjordes enligt figur 3.(2-4), för att applikationer skulle kunna komma åt databasen, hämta och ändra datan i databasen med mera.



Figur 3.2 Inställningar för att definiera TCP port 1433 vilket är viktigt för att datorn ska leda inkommande förfrågningar rätt.



Figur 3.3 Inställningar för att tillåta SQL Server Authentication och inte bara Windows authentication. Login auditing var ställd på detta vis per default.



Figur 3.4 Inställningar för att tillåta anslutningar mot databasen men också för att aktivera login vid anslutning mot databasen.

3.1.3 Power BI

Då databasen temporärt kördes på datorn där programmeringen skrivs så konfigureras också Power BI på den datorn [11]. Detta gör också att det i simuleringsfasen kunde undvikas att kommunikationen mellan databasen och applikationen Power BI behöver upprättas över nätet och istället kommunicera lokalt på en dator vilket underlättade utvecklingen. En begränsning i utvecklingen av Power BI i det här skedet var att datan som kommer ligga i databasen när den implementeras i fabriken inte ännu finns i databasen. Då Power BI är ett program som direkt kopplar data från databasen till programmet så blir det mer arbete att skapa den visuella presentationen före datan finns i databasen. Power BI är alltså den komponent i arbetet som förväntas vara minst färdig vid implementering.

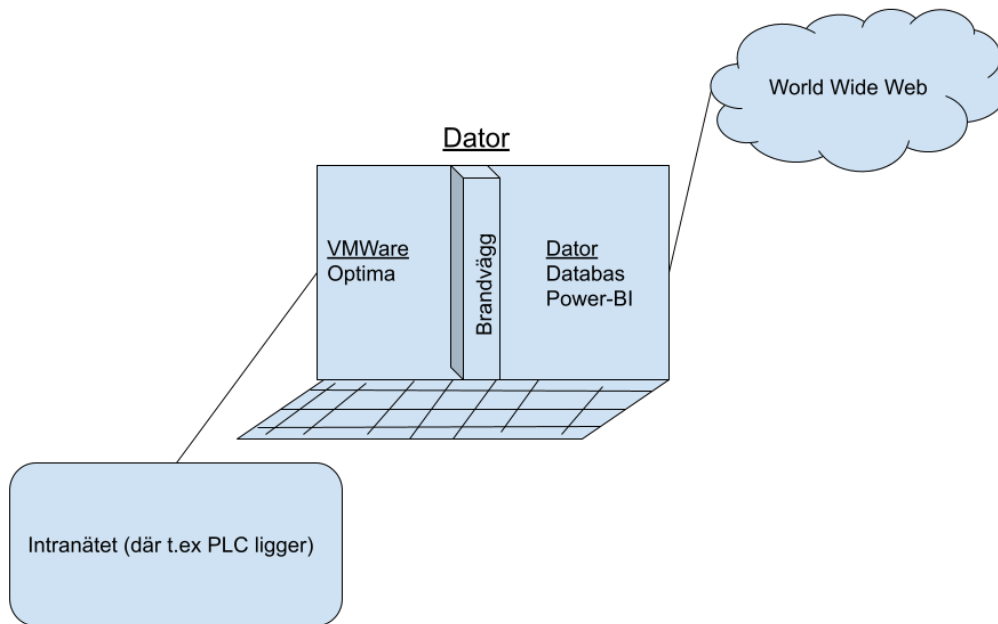
3.2 Integrering

Denna sektion kommer att behandla hur de olika komponenterna ska kommunicera och interagera med varandra. Det finns många olika sätt att sätta upp denna typ av system. Det som kommer vara i fokus för kommunikationen är att centralisera så mycket som möjligt på samma dator då det underlättar arbetet samtidigt som det gör systemet säkrare. En nackdel med att centralisera är att det gör systemet mer sårbart för om servern som det körs på går sönder. För att säkra detta så ska systemet säkerhetskopieras före idrifttagning så att om servern går sönder så ska det kunna snabbt komma på plats igen. Datan som kommer att ligga på databasen är inte lika viktig då det endast är olika mätvärden från processen, alltså kommer arbetet inte ta hänsyn till säkerhetskopiering av datan över tid.

3.2.1 PLC till databas

För kommunikation till databasen från PLC så användes en applikation som heter Optima utvecklad av Mårtensson Consulting. Optima opererar genom att den skannar av PLCn och skickar upp ny data. Optima körs på samma server/dator som databasen men måste ha tillgång till PLCns nätverk.

Datorn som kör databasen, Optima och senare Power BI kommer vara uppkopplad på både det lokala kontorsnätet samt det produktionsnätverk som PLCn och andra produktionsrelaterade moduler kommunicerar på. Det här genomfördes med hjälp av VMWare. En virtuell dator sattes upp i VMWare och den ska ligga på samma subnät som intranätet. Det blev då bryggan mellan kontorsnätet och produktionsnätverket vilket gör det viktigt att endast Optima får kommunicera mellan den virtuella maskinen och datorn där databasen ligger. Problemet löses enklast genom en brandvägg, figur 3.5.



Figur 3.5 Representerar förhållandet mellan VMWare, datorn, www och intranätet.

Konfigureringen av Optima gick till på följande sätt:

Optima konfigurerades med namn på databasen, användarnamn och lösenord. Därefter kunde det kontrolleras om databasen var uppkopplad. Vidare så konfigurerades applikationen med IP Adress till PLCn. Det gjorde att applikationen kontrollerar om den får kontakt med PLCn. När applikationen fick kontakt med både databasen och PLCn skulle datan som skulle läsas av definieras för att kunna läsa in värden från processen. Det gjordes genom att skriva adresserna till taggarna i PLCn i Optima och koppla den med en beskrivning och namn. Adresser i Siemens PLCer skrivs som till exempel DB10.DBD1.0 vilket beskriver bit 1.0 i datablock 10. I detta skede konfigurerades också på vilket sätt värdet skulle läsas: om den skulle läsas och skrivas in i databasen med till exempel ett specifikt tidsintervall eller om värdet skulle läsas av och vid en viss förändring skrivs in i databasen. Det fick tas hänsyn till att databasen inte var så stor vilket resulterade i att skrivfrekvensen till databasen försöktes hålla nere.

För att logga batcher så behövdes det skrivas en del kod i PLCn för att Optima skulle kunna logga olika batcher. En batch definieras i PLCn genom start av inflöde till tanken och stopp av utflöde ur tanken. Detta ger upphov till tre olika moment i en

batch vilket loggas var för sig. Det är alltså normalt sett en flödesloggning vilket loggas med ett batch id som definieras och skrivs av batch-delen.

3.2.2 Databas till Power BI

Den sista fasen i arbetet var att konfigurera Power BI då det underlättade att ha datan som ska presenteras tillgänglig vid konfigureringen. Power BI skulle köras på samma dator som databasen för att underlätta kommunikationen. Applikationen skulle alltså kunna kommunicera lokalt med databasen och inte behöva sätta upp en TCP-IP förbindelse varje gång den hämtar datan. Det underlättar också trafiken på nätet. Nackdelen är dock att datorn/servern som nu kör både Power BI, Optima och SQL Server express blir relativt hårt belastad och en kritisk punkt i systemet.

3.3 Implementering

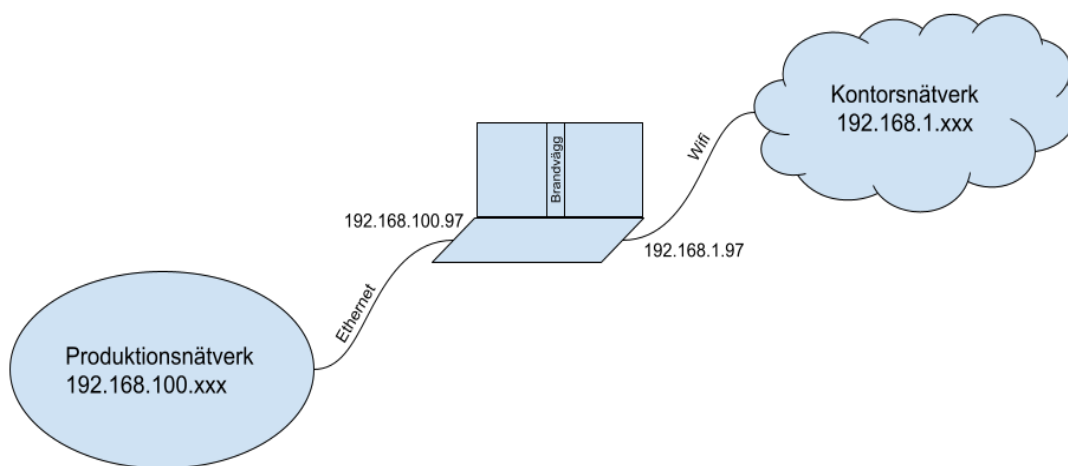
Denna del av metodkapitlet kommer beskriva hur de olika systemen har implementerats efter att de föregående faserna, som beskrivs ovan i kapitel 3.1 och 3.2, var slutförda. Vid integrering så har programmen satts upp att fungera på en dator som var tänkt att sedan användas vid implementering. Den datorn har också använts vid implementering.

3.3.1 VMWare

Ett problem som upptäcktes vid implementering var att fabriken består av två olika nätverk. Den består av ett kontorsnät som är uppkopplad mot WWW och ett produktionsnät som var lokalt kopplat, direkt bortkopplad från kontorsnätet. Detta var ett problem då visualiseringen av databasen skulle vara tillgänglig på samtliga datorer som var uppkopplade på kontorsnätet vilket inte var möjligt med denna konfiguration. För att lösa problemet så kunde VMWare användas. VMWare är, som tidigare beskrivet i kapitel 2, ett program som ger möjligheten att skapa ett operativsystem i ett annat. För att kunna lösa problemet med att nätverken är separerade så behöver en dator kunna kommunicera med båda nätverken samtidigt. Det är möjligt med hjälp av VMWare då den virtuella datorn kan konfigureras att ligga på ett annat subnät än vad datorn läggs på.

3.3.2 IP adressering

VMWare konfigurerades att ha IP adress 192.168.100.97 där subnät masken är 255.255.255.0. Det innebär att den virtuella maskinen på datorn nu ligger på samma nätverk som produktionsnätverket som har subnät 192.168.100.xxx och kommunicerar genom Ethernetkabeln. Datorns ursprungliga operativsystem konfigurerades att vara uppkopplad på kontorsnätet, dock med automatisk dynamisk IP adress då det inte är nödvändigt att ha ett statiskt nummer på den sidan. Datorns interna IP adress som är uppsatt mellan VMWare och huvuddatorn är 192.168.126.1 på datorns sida och 192.168.126.2 på VMWares sida. Konfigurationen representeras i figur 3.6.



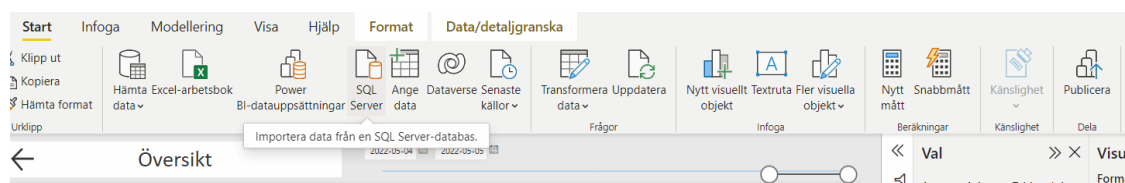
Figur 3.6 IP adresseringen av de olika delarna där molnet till höger och ovalen till vänster representerar subnät.

3.3.3 Brandvägg

En viktig aspekt som behövdes tas hänsyn till var säkerheten. Eftersom då produktionsnätverket fram tills nu inte varit tillgänglig från kontorsnätet så hade det varit avskärmat men nu behövdes det också skyddas från eventuella angrepp. Då den enda kommunikation som var nödvändig mellan den vanliga datorn och VMWare var kommunikation till databasen, så begränsades brandväggen till att endast tillåta kommunikation från den virtuella datorn till databasen. Detta diskuterades med företaget och ansågs ge tillräcklig säkerhet.

3.3.4 Power BI

För implementeringen av Power BI behövdes Power BI laddas ner från Microsofts hemsida. Programmet startades upp på datorn som det senare skulle ligga på. Programmet behövdes konfigureras att kommunicera mot databasen. Detta gjordes med hjälp av menysystemet som ses i figur 3.7.



Figur 3.7 Menysystemet i Power BI.



Figur 3.8 Fönster där databasens uppgifter skrivs in för att kunna initiera en kommunikation till den.

I fönstret, som ses i figur 3.8, för Server så står det localhost\SQLEXPRESS där localhost är en allmän adress som betyder den aktuella datorn. För att datan skulle kunna uppdateras kontinuerligt så behövdes DirectQuery vara markerad. Efter dessa steg så återstod endast att välja ut datan och presentera den på ett visuellt tilltalande vis, detta gjordes med genom att studera manualer och experimentering. Se kapitel 4 för att se hur datan sedan blev visualiserad.

Navigatör

The screenshot shows a navigation interface with a table list on the left and a data table on the right. The table list includes items like 'cipObjectsDc1', 'logData', and 'logid'. The data table has columns: 'index', 'logDate', 'logid', 'batchid', 'eventid', 'equipmentid', 'materialid', 'source', and 'destination'. Below the table are buttons: 'Välj relaterade tabeller', 'Läs in', 'Transformera data', and 'Avbryt'.

index	logDate	logid	batchid	eventid	equipmentid	materialid	source	destination
314641	2022-05-05 06:02:46	1	232501001	1753-01-01 00:00:00	200	0	100	
314634	2022-05-04 06:17:33	1	232401001	1753-01-01 00:00:00	200	0	100	
314635	2022-05-04 06:17:33	3	232401001	2022-05-04 06:17:33	200	0	0	
314640	2022-05-04 11:58:53	2	232401002	1753-01-01 00:00:00	200	0	100	
314642	2022-05-05 06:41:53	2	232501001	1753-01-01 00:00:00	200	0	100	
314643	2022-05-05 06:45:45	1	232501002	1753-01-01 00:00:00	200	0	100	
314638	2022-05-04 11:01:11	1	232401002	1753-01-01 00:00:00	200	0	100	
314639	2022-05-04 11:01:11	3	232401002	2022-05-04 11:01:11	200	0	0	
314644	2022-05-05 07:52:45	2	232501002	1753-01-01 00:00:00	200	0	100	
314646	2022-05-05 07:58:13	2	232501003	1753-01-01 00:00:00	200	0	100	
314636	2022-05-04 07:00:35	2	232401001	1753-01-01 00:00:00	200	0	100	
314637	2022-05-04 07:00:37	4	232401001	2022-05-04 06:17:33	200	0	0	
314645	2022-05-05 07:56:48	1	232501003	1753-01-01 00:00:00	200	0	100	
314520	2022-04-06 06:01:29	1	229601001	1753-01-01 00:00:00	200	0	100	
314521	2022-04-06 06:01:29	3	229601001	2022-04-06 06:01:29	200	0	0	
314522	2022-04-06 06:35:14	2	229601001	1753-01-01 00:00:00	200	0	100	

Figur 3.9 Visar till vänster vilka tabeller som finns i databasen och till höger vilken data som finns i vald tabell.

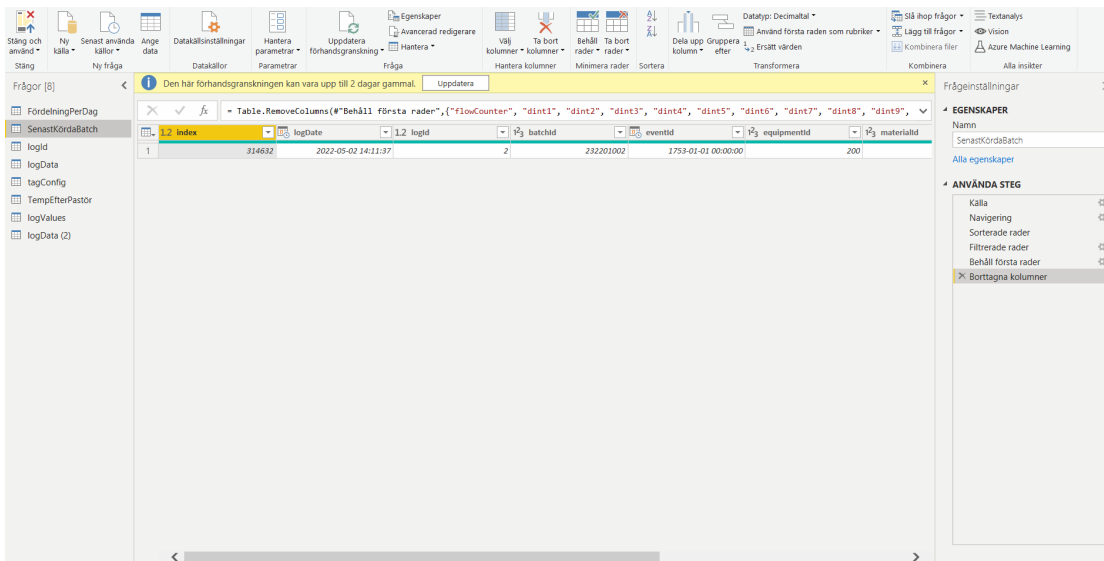
Efter att datan som skulle användas valts ut skulle datan filtreras och grupperas.

Fönstret för att transformera datan visas i figur 3.10.

The screenshot shows a data transformation interface. The main area displays a data table with columns: 'index', 'logDate', 'logid', 'batchid', 'eventid', 'equipmentid', 'materialid'. On the right, there is a 'Frågenställningar' panel with sections for 'EGENSKAPER' (logData (2)) and 'ANVÄNDA STEG' (Källa, Navigering). The status bar at the bottom indicates '31 KOLUMNER, 127 RADER' and 'Kolumprofileringen baseras på de översta 1000 raderna'.

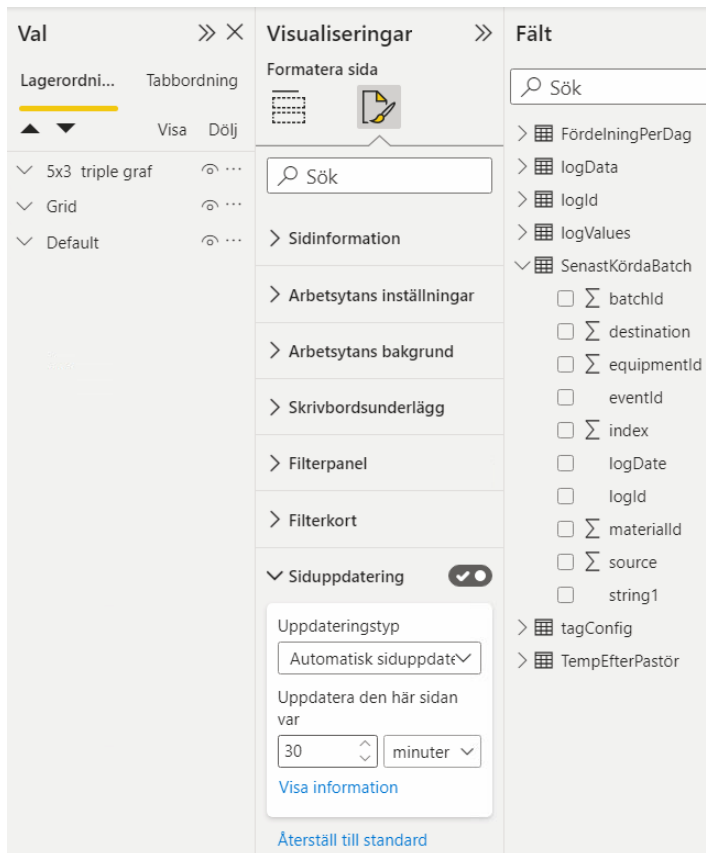
Figur 3.10 Visar hur datan ser ut innan den har blivit transformerad.

I detta steg grupperades och filtrerades datan efter hur den önskades presenteras. Ett exempel på hur datan kunde se ut efter transformering ses i figur 3.11. Datat representerar den senaste avslutade batchen och batch id visas i figur 4.9.



Figur 3.11 Visar hur datan kan se ut efter transformering.

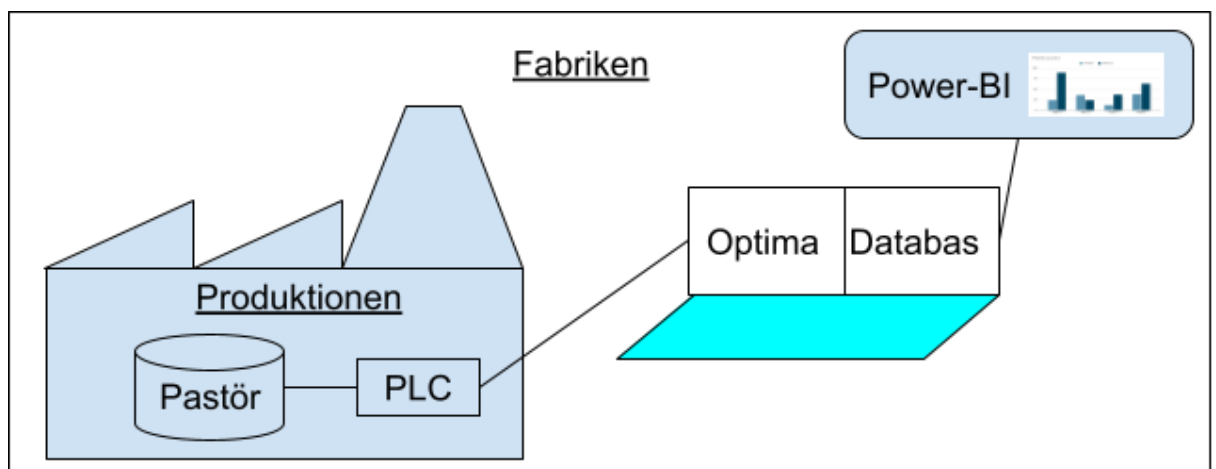
Efter att datan blev transformerad och fått sin struktur för hur den skulle utläsas så återstod endast att konfigurera hur ofta datan skulle uppdateras, alltså hur ofta programmet skulle gå in och läsa av datan i databasen och presentera den. Detta kan ses i figur 3.12.



Figur 3.12 Översiktsbild på inställningarna för programmets visuella inställningar.

4 Resultat

Det allmänna resultatet har blivit som önskat där systemet har kunnat implementeras på Balsgård FoodTech AB samt kunnat testköras och samla in exempeldata så att också grunden för Power BI har kunnat byggas upp. Nedan så beskriver respektive kapitel resultatet för varje del i arbetet med den naturliga uppdelningen PLC, databas och Power BI. Figuren 4.1 beskriver visuellt den aktuella hårdvaru- och mjukvaru strukturen som är implementerad.



Figur 4.1 Övergripande bild över den slutgiltiga lösningen.

4.1 PLC

Resultatet för PLC programmeringen har uppfyllt de önskemål som ställdes vid initieringen av arbetet. Nedan beskrivs det övergripande resultatet av PLC programmeringen då det blir svårt att visualisera för mycket i detalj.

En stor del av arbetet i PLCn har gått till att konfigurera de olika delarna av PLCn för att få med alla de väsentliga delarna. Det har varit till hjälp att studera de redan implementerade system i fabriken där Mårtensson Consulting skapat ett liknande system. Därifrån har också hämtats en del inspiration.

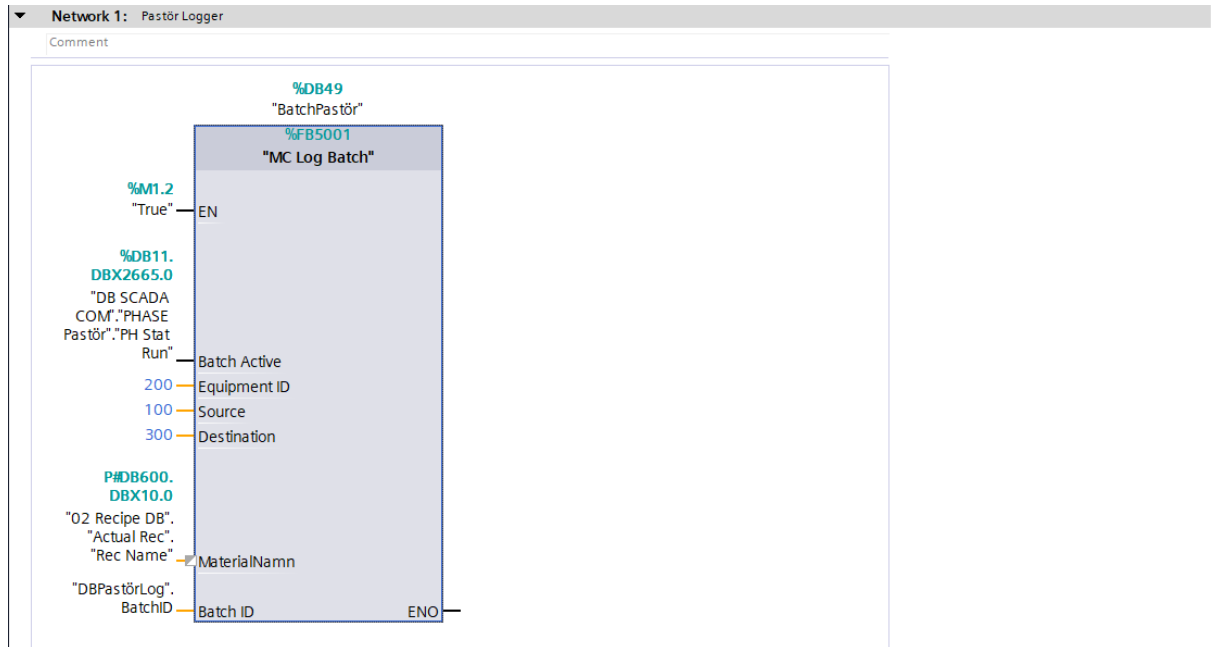
Figur 4.2 visar på hur koden ser ut i den mest väsentliga mappen. PLCn loggar batcher som sedan Optima läser. Figur 4.3 är Network 1 expanderat där det på vänstersidan är ingångar till blocket som hanterar batch loggningen. “Batch Active” ska vara hög (true) när batchen är aktiv, då vet blocket att batchen har startat, ingången som är kopplad till denna kommer från pastören och aktiveras när den kör. “EquipmentID” har värdet 200 för att man ska kunna särskilja på de olika maskinerna senare i databasen. “Source” har värdet 100 vilket indikerar vilken maskin batchen kommer ifrån. “Destination” har samma princip som “Source” men tvärt om, dvs var batchen går efter denna maskin. “MaterialNamn” är ingången där produkten som batchen består av heter. I denna implementation så är det olika recept som definierar batchens innehåll, alltså är ingången på blocket namnet på det aktuella receptet som körs. Sist i blocket finns “BatchID” vilket ska vara ett unikt batch ID som ska vara unikt för den tänka batchen, vilket den också blir med taggen som ligger på ingången.

Figur 4.4 visar hur datan från de olika mätvärdena skrivs över till separata taggar i ett unikt datablock vilket gör att Optima kan konfigureras att läsa in de värdena och lagra dem i databasen. Move blocket är en funktion där värdet som går in på vänstersidan skrivs över till utgången på högersidan.

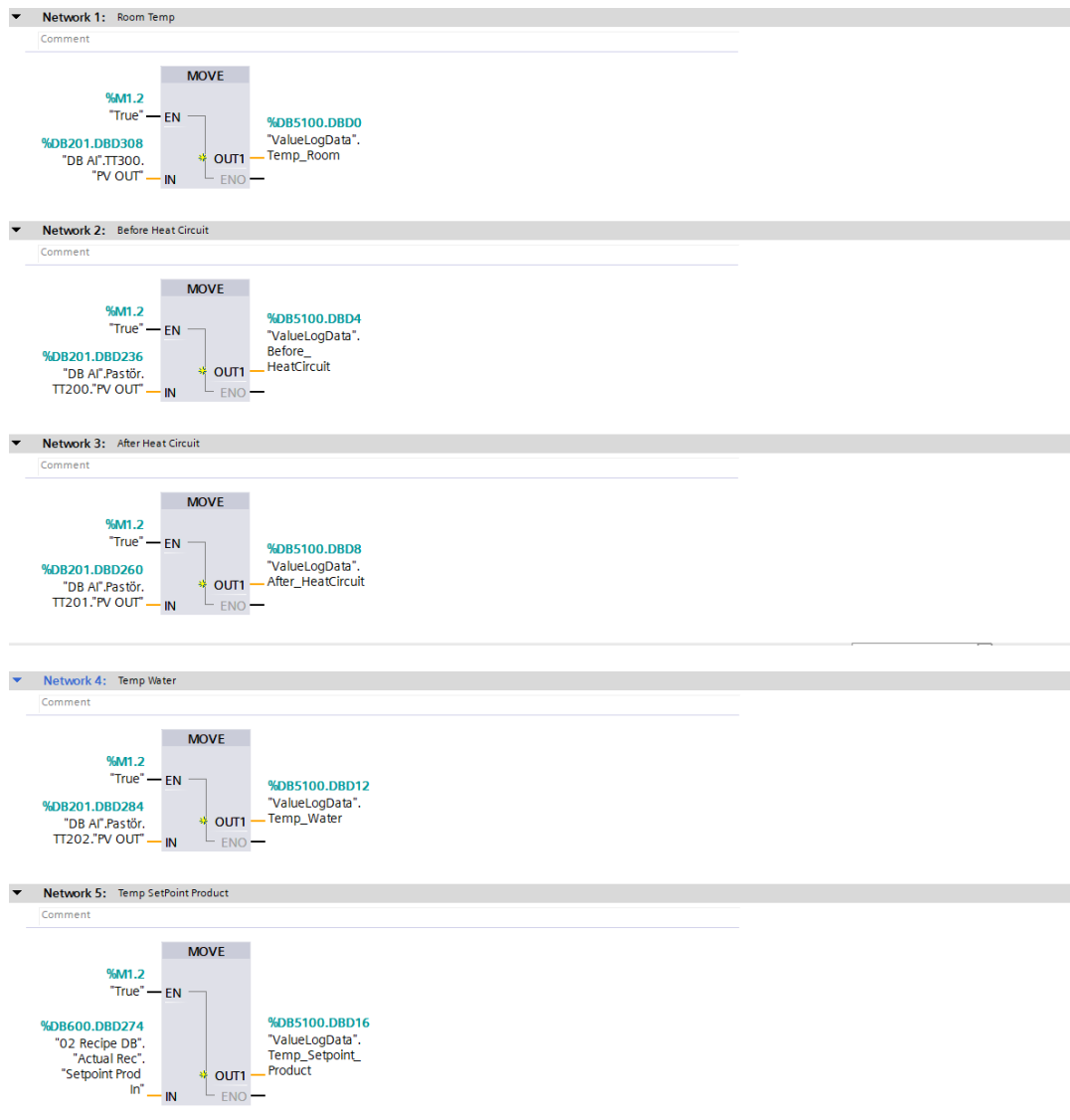
Figur 4.5 visar DB5100 där de olika variablerna som Optima läser av ligger, deras datatyp är definierade till typen real. Figur 4.6 visar den övergripande mappstrukturen med de olika komponenterna som används för att kommunikationen ska fungera. Där kan man bland annat se DB5100 som figur 4.5 visar i expanderad form. FC blocket MC Log Queue Handler är den viktigaste komponenten i denna mapp för att kommunikationen med Optima ska fungera och har behövts konfigurerats för det specifika ändamålet.



Figur 4.2 Visar strukturen i koden och de olika komponenterna som expanderas i figur 4.3



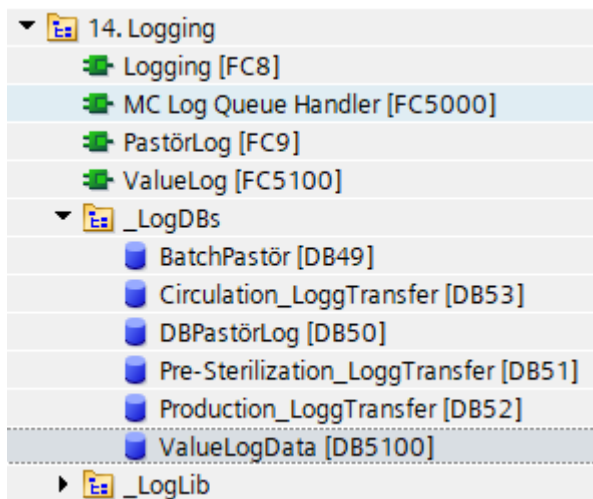
Figur 4.3 Network 1 figur 4.2 expanderat och visar ett funktionsblock som hanterar loggningen pastörens batcher.



Figur 4.4 överföring av data som ska loggas till en ny minnesarea för att samla data för att ge en bättre struktur.

1	Static									
2	Temp_Room	Real	0.0	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	Before_HeatCircuit	Real	4.0	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	After_HeatCircuit	Real	8.0	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5	Temp_Water	Real	12.0	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6	Temp_Setpoint_Produ..	Real	16.0	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figur 4.5 DB där datan som ska loggas samlas för att ge bättre överblick och struktur över datan som ska loggas.

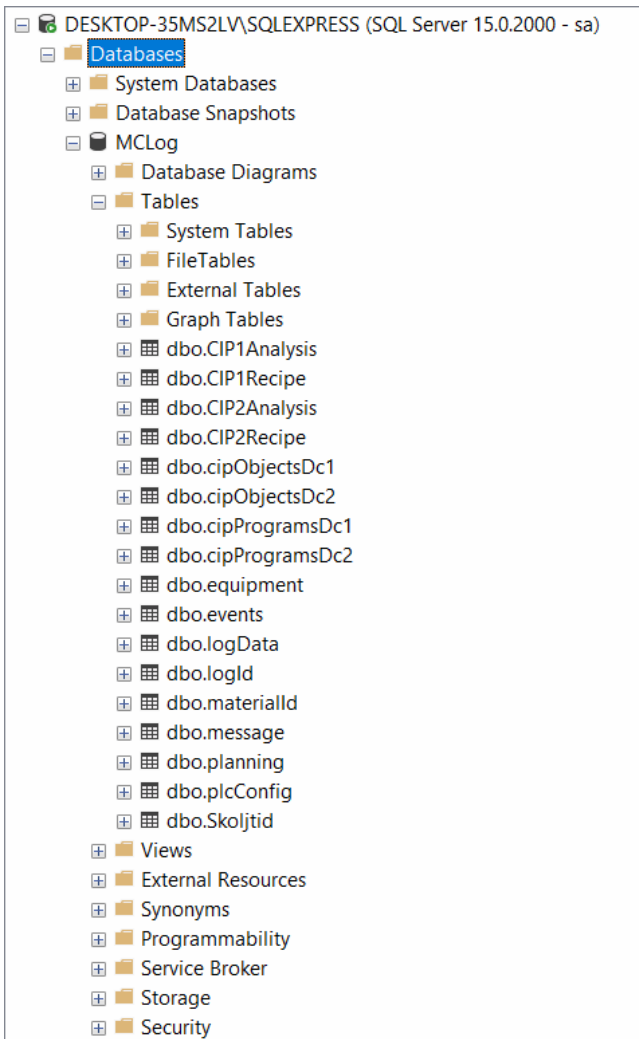


Figur 4.6 Datablock som visas 4.5 i applikations trädets och de olika övergripande delarna där som kodats för att låta Optima läsa datan.

4.2 Databas

Figur 4.7 visar hur databasens uppbyggnad blev. MCLog är namnet på databasen och alla tabeller som denna databas innefattar heter `dbo.*`, det är många tabeller som inte behöver användas men som bara ligger där för framtida implementationer. Den tabell som används av Power BI i detta arbete för att presentera datan heter `dbo.logData`.

I figur 4.8 så visas ett representativt exempel på hur datan kan se ut i `logData`. Vid jämförelse med PLC konfigurerings så kan vi se att variablernas namn går att hitta och likställas med varandra.



Figur 4.7 översiktsbild över alla de olika tablesen i databasen samt en del andra irrelevanta mappar.

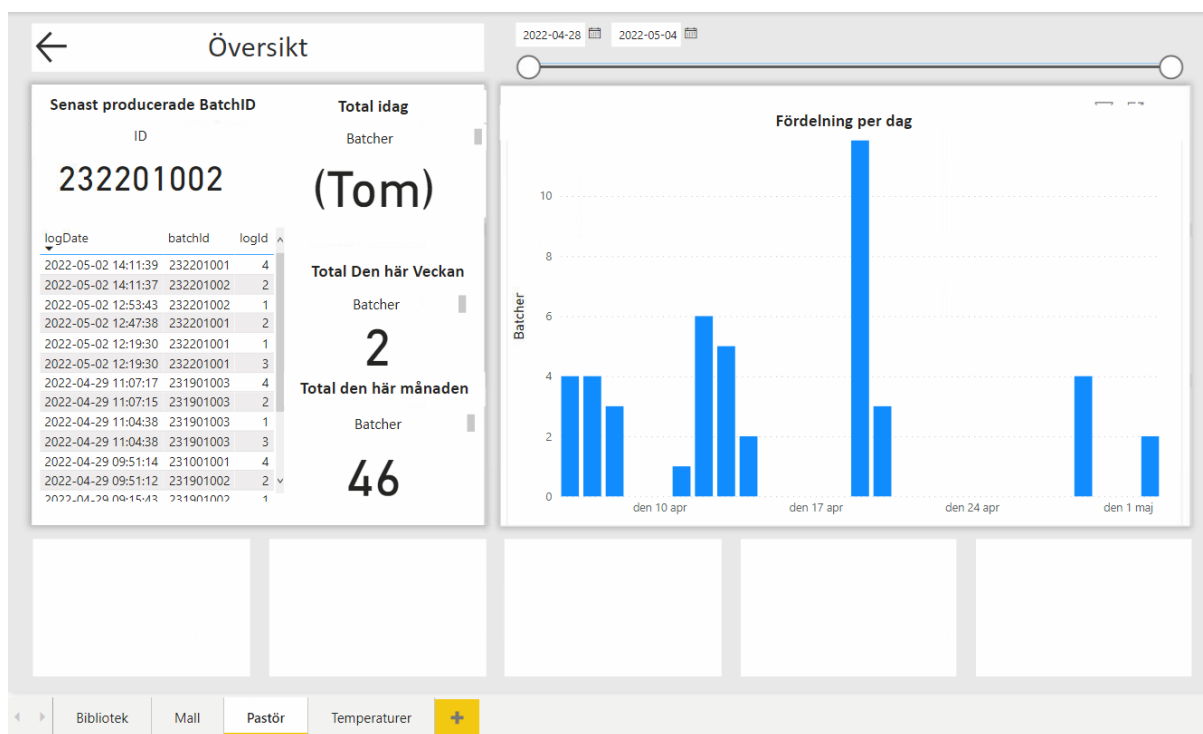
index	logDate	logId	batchId	eventId	equipmentId	materialId	source	destination	flowCounter	dint1	dint2	dint3	dint4	dint5	dint6	dint7	dint8	dint9	dint10
24...	2022-04-19 10:51:06.913	2	230901010	1753-01-01 00:00:00.0000000	200	0	100	300	0	0	0	0	0	0	0	0	0	0	0
24...	2022-04-19 10:51:14.623	4	230901006	2022-04-19 09:39:51.5530070	200	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24...	2022-04-19 10:51:19.480	1	230901011	1753-01-01 00:00:00.0000000	200	0	100	300	0	0	0	0	0	0	0	0	0	0	0
24...	2022-04-19 10:51:19.480	3	230901011	2022-04-19 10:51:19.4800213	200	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24...	2022-04-19 11:25:11.937	2	230901011	1753-01-01 00:00:00.0000000	200	0	100	300	0	0	0	0	0	0	0	0	0	0	0
24...	2022-04-19 11:39:23.647	1	230901012	1753-01-01 00:00:00.0000000	200	0	100	300	0	0	0	0	0	0	0	0	0	0	0
24...	2022-04-19 12:38:09.087	2	230901012	1753-01-01 00:00:00.0000000	200	0	100	300	0	0	0	0	0	0	0	0	0	0	0
24...	2022-04-19 12:38:21.963	4	230901011	2022-04-19 10:51:19.4800213	200	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24...	2022-04-20 08:06:24.930	1	231001001	1753-01-01 00:00:00.0000000	200	0	100	300	0	0	0	0	0	0	0	0	0	0	0
24...	2022-04-20 08:06:24.930	3	231001001	2022-04-20 08:06:24.9312489	200	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24...	2022-04-20 08:20:03.693	2	231001001	1753-01-01 00:00:00.0000000	200	0	100	300	0	0	0	0	0	0	0	0	0	0	0
24...	2022-04-20 08:21:16.503	1	231001002	1753-01-01 00:00:00.0000000	200	0	100	300	0	0	0	0	0	0	0	0	0	0	0
24...	2022-04-20 08:27:40.597	2	231001002	1753-01-01 00:00:00.0000000	200	0	100	300	0	0	0	0	0	0	0	0	0	0	0
24...	2022-04-20 08:28:47.957	1	231001003	1753-01-01 00:00:00.0000000	200	0	100	300	0	0	0	0	0	0	0	0	0	0	0
24...	2022-04-20 09:31:42.670	2	231001003	1753-01-01 00:00:00.0000000	200	0	100	300	0	0	0	0	0	0	0	0	0	0	0

Figur 4.8 Exempel på hur datan ser ut och är strukturerad i ett table när den bara lagras.

4.3 Power BI

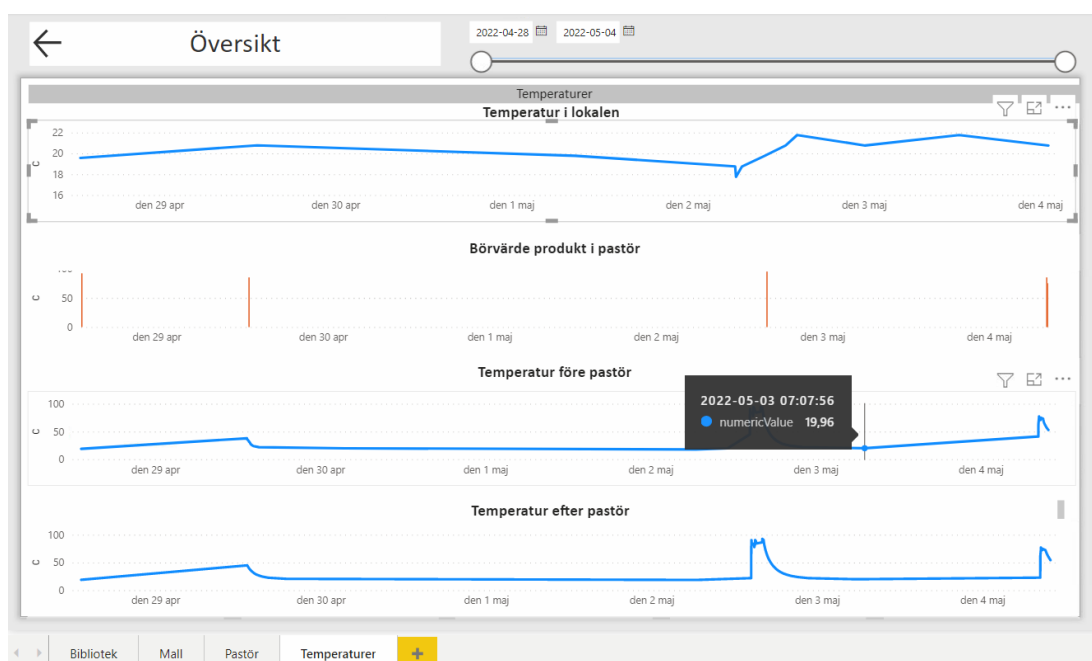
Resultatet på Power BI har uppnått arbetets mening, datan kan presenteras på ett intuitivt och interaktivt sätt. Nedan kommer en beskrivning av figurerna och vilken typ av data de presenterar.

I figur 4.9 visas diagram och olika värden. Som rubrikerna beskriver så representerar de olika siffrorna senast producerade batch id, totalt producerade batcher idag, totalt producerade batcher den här veckan respektive totalt producerade batcher den här månaden. Tabellen till vänster i figur 4.9 är en representation av hur datan ser ut i databasen, dock i ett skalat format. Detta för att ge användaren möjligheten att också inspektera den råa datan. Till höger i figuren så representeras antalet producerade batcher per dag med staplar för att ge en överblick över perioden, där x-axeln är datum och y-axeln är antalet batcher per dag.



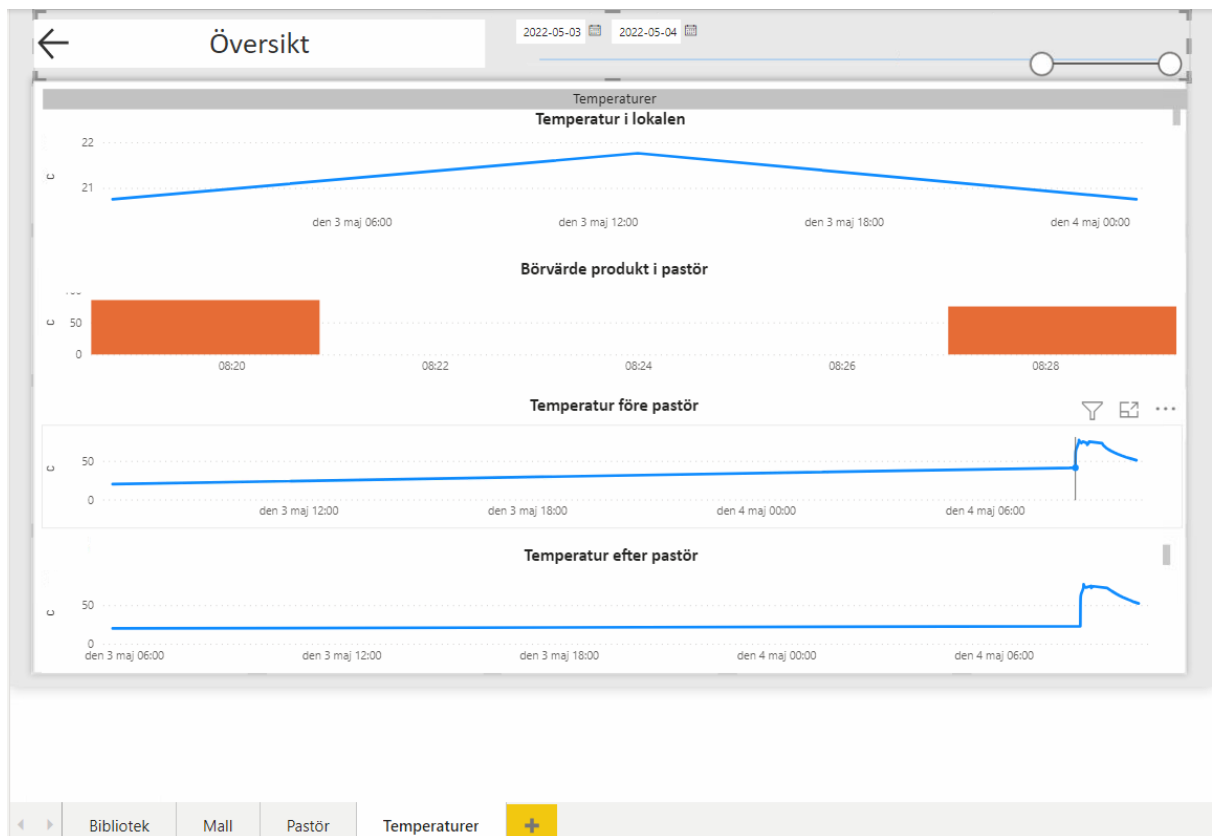
Figur 4.9 Visar hur datan från pastören kan presenteras av Power BI på ett visuellt tilltalande sätt.

I figur 4.10 så visas diagram där datan från olika temperaturgivare presenteras i form av kontinuerliga kurvor över tid. Datan loggas inte hela tiden utan databasen blir uppdaterad vid en viss temperaturändring. Datan som samlas i diagrammen nedan är alltså en representation över tid som inte är lika exakt över tid som givaren kan vara, men tillräckligt noggrann för att ge tillräcklig tillförlitlighet. I diagram 3 uppifrån sett så visas hur det ser ut när musen hovrar över diagrammets linje. Då presenteras punktens värde (i detta fall 19,96C) vilket innebär att det vid 2022-05-03 07:07:56 var det 19,96 grader i slingan före pastören.



Figur 4.10 visar hur alla de olika temperaturerna kan presenteras av Power BI på ett visuellt tilltalande sätt.

I figur 4.11 visas hur det ser ut när intervallet uppe till höger i figuren ändras. Intervallet är kopplat till diagrammens tidsaxlar och när det ändras så skalas också diagrammen efter hur intervallet ställs. Detta är användbart då användaren är intresserad av en viss period, då det är svårt att urskilja små skillnader i ett långt intervall så blir det mer tydligt ju mindre intervallet är. Figur 4.10 och figur 4.11 presenterar samma data men med olika intervall. Det syns då tydligare att temperaturen når sitt toppvärde på dagen kl 12 i figur 4.11 vilket inte syns lika väl i figur 4.10.



Figur 4.11 visar på hur datan går att interagera med för att specificera vilken tid man vill inspektera.

Datan som presenteras i figur 4.10 och figur 4.11 ser ut enligt figur 4.12 när den ligger i databasen vilket också visar på Power BIs funktion som är att göra denna data användbar.

	index	logDate	logId	batchId	eventId	equipmentId	materialId	source	destination
1	314634	2022-05-04 06:17:33.423	1	232401001	1753-01-01 00:00:00.0000000	200	0	100	300
2	314635	2022-05-04 06:17:33.423	3	232401001	2022-05-04 06:17:33.4246133	200	0	0	0
3	314633	2022-05-02 14:11:39.227	4	232201001	2022-05-02 12:19:30.3039424	200	0	0	0
4	314632	2022-05-02 14:11:37.293	2	232201002	1753-01-01 00:00:00.0000000	200	0	100	300
5	314631	2022-05-02 12:53:43.080	1	232201002	1753-01-01 00:00:00.0000000	200	0	100	300
6	314630	2022-05-02 12:47:38.600	2	232201001	1753-01-01 00:00:00.0000000	200	0	100	300
7	314628	2022-05-02 12:19:30.303	1	232201001	1753-01-01 00:00:00.0000000	200	0	100	300
8	314629	2022-05-02 12:19:30.303	3	232201001	2022-05-02 12:19:30.3039424	200	0	0	0
9	314627	2022-04-29 11:07:17.197	4	231901003	2022-04-29 11:04:38.4354552	200	0	0	0
10	314626	2022-04-29 11:07:15.490	2	231901003	1753-01-01 00:00:00.0000000	200	0	100	300
11	314624	2022-04-29 11:04:38.437	1	231901003	1753-01-01 00:00:00.0000000	200	0	100	300
12	314625	2022-04-29 11:04:38.437	3	231901003	2022-04-29 11:04:38.4354552	200	0	0	0
13	314623	2022-04-29 09:51:14.397	4	231001001	2022-04-20 08:06:24.9312489	200	0	0	0

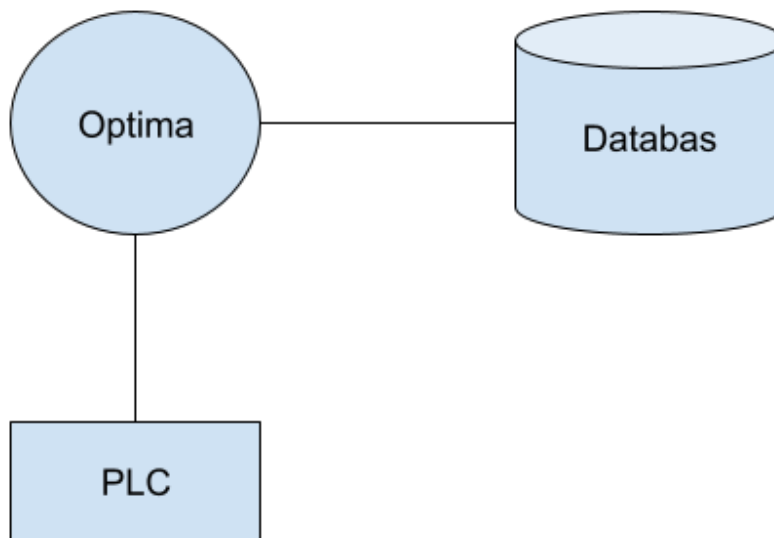
Figur 4.12 data från de olika batcherna och hur de ser ut när den ligger i databasen.

5 Slutsats

Frågeställningarna som ställdes i början av arbetet:

1. Hur kan data kommuniceras upp till en databas från en PLC?
2. Hur lagras data på ett strukturerat effektivt sätt i en databas?
3. Hur plockas data ut för visualisering?

Den första frågan hur data kan kommuniceras upp till en databas från en PLC har besvarats. För att kommunicera upp data till en databas så finns det ett flertal utmaningar som måste lösas innan det är möjligt att implementera. PLCn kan inte direkt kommunicera med databasen, alltså måste det upprättas en instans emellan PLCn och databasen som agerar översättare. Detta löstes i detta arbete genom att använda sig av Mårtensson Consultings egenutvecklade program Optima. Applikationen sätts upp och opererar på en server och kopplas upp mot PLC och databas enligt den visuella representationen nedan.



Figur 5.1 Representation av vart Optima opererar.

Den andra frågan har besvarats genom valet av att använda Optima då den inte ger någon dynamik i hur man lagrar datan i databasen. Optima kräver en viss struktur i databasen för att kunna kommunicera upp datan och lagra den. Det mest effektiva sättet blev alltså att lagra allt i samma tabell och inte i olika tabeller. Dock så behövdes datan fortfarande länkas ihop och struktureras efter lagring. Datan låg alltså efter konfigurering av PLC och databas i en enda lång lista. Då Power BI ger möjligheten att strukturera datan vid inläsning så användes det för att strukturera upp datan istället för att den skulle ligga strukturerad i databasen vid lagring.

Den tredje frågan besvarades genom att använda sig av Power BI. Applikationen var ett intuitivt program som kopplades upp mot databasen. Efter att databasen hade blivit installerad på Balsgård FoodTech AB så samlade den in en del exempeldata som visade på hur strukturen och datan såg ut vid produktion. Då kunde Power BI kopplas upp mot databasen och efter definiering och sammanslagning av olika data så kunde datan också visualiseras på ett intuitivt och visuellt tilltalande sätt.

5.1 Framtida utvecklingsmöjligheter

Framtida utvecklingsmöjligheter för detta arbete är stora då detta arbete lägger grunden för ett system som kan samla in mycket mer data och visualisera den. Mårtensson Consulting har för avsikt att med denna anläggning kunna visa och presentera fördelarna med att lagra produktionsdata. En ambition från Mårtensson Consulting var att också kunna lagra andra typer av manuell data i denna server. Det är möjligt att göra genom att t.ex skapa en webbsida för inmatning av data in i databasen. Det skulle då innebära att datan som databasen innehåller skulle bli ännu mer representativ för hela fabriken. Att kunna leverera en helhetslösning för lagring och presentering av data i mindre fabriker är också Mårtensson Consultings ambition då de ofta kan ha svårt att göra stora investeringar för att utveckla dessa typer av system själva.

En utvecklingsmöjlighet är också att koppla upp fler PLCer då det finns en till PLC i fabriken på Balsgård så skulle den också kunna kopplas upp för lagring av datan från den delen av produktionen. Detta arbete har underlättat och lagt grunden för en enkel utökning av antalet PLCer, vilket kan komma till användning också i framtiden då Mårtensson Consulting har för ambition att utveckla och expandera Balsgård FoodTech ABs produktion.

6 Referenser

6.1 Litteratur

[1] C.E. Spurgeon, Ethernet The Definitive Guide, O'Reilly Media, 2000, Sebastopol, USA

Länk till sida där bok finns tillgänglig:

https://books.google.se/books?hl=sv&lr=&id=TB-cAgAAQBAJ&oi=fnd&pg=PR5&dq=ethernet&ots=F5w4GgK9g7&sig=pR_M8DN3eOqvtxK-fTo52SN-0VY&redir_esc=y#v=onepage&q=ethernet&f=false (Lästes senast: 4/5-22)

[2] VMWare, VMware Workstation 16.2.3 Pro Release Notes, 2022

Länk:

<https://docs.vmware.com/en/VMware-Workstation-Pro/16.2.3/rn/vmware-workstation-pro-1623-release-notes/index.html> (Lästes senast: 27/4-22)

[3] Microsoft, SQL Server 2019: Hardware and software requirements, 2022

länk:

<https://docs.microsoft.com/en-us/sql/sql-server/install/hardware-and-software-requirements-for-installing-sql-server-2019?view=sql-server-ver15> (Lästes senast: 27/4-22)

[4] J. Melton & A. Eisenberg, SQL multimedia and application packages, Sigmod Record, 2001

länk:

<https://dl.acm.org/doi/pdf/10.1145/604264.604280> (Lästes senast: 27/4-22)

[5] E.R. Alphonsusa & M.O. Abdullah, A review on the applications of programmable logic controllers (PLCs), Elsevier, 2016

<https://www.sciencedirect.com/science/article/pii/S1364032116000551>

(Lästes senast: 27/4-22)

[6] Microsoft, Introducing Microsoft Power BI, 2016

Länk till sida där bok finns tillgänglig:

https://books.google.se/books?hl=sv&lr=&id=U1qsDAAAQBAJ&oi=fnd&pg=PT8&dq=power+bi&ots=pAQXU0qVOE&sig=Zd8ZPu2g0Ilu0pg7M6UNp2Z-6Kg&redir_esc=y#v=onepage&q=power%20bi&f=false (Lästes senast: 27/04-22)

[7] J. Björkdahl, M.W.Wallin & C.Kronblad, Digitalisering – mer än teknik, Vinnova, 2018

Länk:

https://www.vinnova.se/contentassets/24d0fc24d08b4d0d900a805384ffc51b/vr_18_06.pdf (Lästes senast: 22/4-22)

[8] VM ware, Why choose VMware, 2016

Länk:

<https://www.vmware.com/cloud-solutions/app-modernization.html>

(Lästes senast: 4/5-22)

[9] Siemens, S7-PLCSIM Advanced V3.0, 2019

Länk:

<https://support.industry.siemens.com/cs/document/109772889/trial-download-simatic-s7-plcsim-advanced-v3-0?dti=0&lc=en-WW>

(Lästes senast: 16/5-22)

6.2 Manualer

[10] VMWare, VMWare workstation manual, 2021

Länk:

<https://docs.vmware.com/en/VMware-Workstation-Pro/16.0/workstation-pro-16-user-guide.pdf>

(Lästes senast: 1/5-22)

[11] Microsoft, Power BI Manual, 2022

Länk:

<https://docs.microsoft.com/sv-se/power-bi/>

(Lästes senast: 3/5-22)

[12] Siemens, Siemens Tia Portal Step 7 Basic V10.5 Manual, 2009

Länk:

https://cache.industry.siemens.com/dl/files/542/40263542/att_829827/v1/GS_STEP7Bas105enUS.pdf

(Lästes senast: 4/5-22)

[13] Microsoft, SQL Server Express Installation Guide, 2016

Länk:

<https://www.bu.edu/csnet/files/2016/09/SQL-Server-Express-Installation-Guide-V01.pdf>

(Lästes senast: 4/5-22)

6.3 Bilder

[14] D. Monniaux, Ethernet RJ45 connector p1160054.jpg, 2007

Länk:

https://commons.wikimedia.org/wiki/File:Ethernet_RJ45_connector_p1160054.jpg

(Hämtades: 15/4-22)

Bilagor

Nedan följer kod från en PLC som hanterar all struktur för kommunikation med Optima för att kunna lagra informationen från PLCn i databasen. All kod nedan är skriven i PLC språket structured text.

Figur A.1 visar koden som tar in variabler såsom Batch ID, Equipment osv för att sedan vid start av batch skicka dessa vidare till blocket “MC Sys Log Data” som markeras i figuren. Figur A.2 visar koden för “MC Sys Log Data” blocket som hittar en ledig plats i arrayen med den loggade datan som ska upp till databasen. Den skriver sedan detta i datablocket som heter Log Queue Master.

Figur A.3 visar datablocket Log Queue Master som Optima är konfigurerad att läsa av. När ett värde är avläst så ska detta värde raderas från loggen då det redan är uppe i databasen. Detta hanteras av blocket MC Log Queue Handler för vilken koden visas i figur A.4. När Optima har hämtat värdet så skickar Optima en signal till PLCn att radera det senaste ur loggen, vilket den gör i detta block.

Figur A.5 visar postern som sammanfattar arbetet kort.

```

1
2 IF "FirstScan" THEN
3     # "Batch Started" := false;
4 END_IF;
5
6 # "MC Batch ID Generator_Instance" ("Trigger Batch ID" := # "Batch Active",
7     "Batch ID" => # "Batch ID");
8
9 REGION Start Batch
10 IF # "Batch Active" THEN
11     #Data."Batch ID" := # "Batch ID";
12     #Data."Equipment ID" := # "Equipment ID";
13     #Data.Source := #Source;
14     #Data.Destination := #Destination;
15     #Data."STRING" := #MaterialNamn;
16 IF NOT # "Batch Started" THEN
17     # "Batch Started" := true;
18     "MC Sys Log Data" ("Log ID" := 1,
19         Data := #Data); // 1 = Start of batch
20 END_IF;
21 END_IF;
22 END_REGION
23
24 REGION Stop Batch
25 IF NOT # "Batch Active" AND # "Batch Started" THEN
26     #Data."Batch ID" := # "Batch ID";
27     #Data."Equipment ID" := # "Equipment ID";
28     #Data.Source := #Source;
29     #Data.Destination := #Destination;
30     #Data."STRING" := #MaterialNamn;
31     # "Batch Started" := FALSE;
32
33     "MC Sys Log Data" ("Log ID" := 2,
34         Data := #Data); // 2 = Stop of batch
35     #Data := "Log Queue Master".Internal."Zero Log";
36 END_IF;

```

Figur A.1 Structured text i blocket "MC Sys Log Data" och är ett redan programmerat block som är tillhandahållet av Mårtensson Consulting.

```

1  "MC Check next Log Number" ("Next Number" => # "Next Free Number",
2  "Next Array" => # "Next Array Number");
3
4  #Data."Log ID" := # "Log ID";
5  #Data.timeStamp := "DB PlcTime".LocalTime;
6
7  IF NOT "Log Queue Master"."Queue OF" THEN
8  IF # "Log ID" = 3 THEN
9      #Data.eventId := #Data.timeStamp;
10  END_IF;
11  IF # "Next Free Number" = # "No Entries" THEN
12      #Data."Queue Sequence" := 1;
13      "Log Queue Master"."SCADA Interface"."Log Values" := #Data;
14  ELSE
15      #Data."Queue Sequence" := # "Next Free Number";
16      "Log Queue Master"."Log Queue"[# "Next Array Number"] := #Data;
17  END_IF;
18  END_IF;
19
20

```

Figur A.2 Structured text i blocket "MC Sys Log Data" och är ett redan programmerat block som är tillhandahållet av Mårtensson Consulting.

Name	Data type	Offset	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervis...	Comment
1	Static									
2	Queue	0.0	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Number of queued events
3	Queue OF	2.0	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Queue Overflow
4	Config	4.0			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
5	SCADA Interface	16.0			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
6	Log Queue	206.0			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
7	Internal	18606.0			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
8	oldHour	18802.0	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			

Figur A.3 DB5000 variabeln Log Queue är en array som Optima läser av för att hämta information om batcher och när dem kördes.

```

1 #Counter := 0;
2 #Temp := 0;
3 IF "Log Queue Master"."SCADA Interface"."Log Values"."Queue Sequence" > 0 THEN
4     "Log Queue Master"."SCADA Interface"."Log Trigger" := true;
5 END_IF;
6
7 IF "Log Queue Master"."SCADA Interface"."Log Trigger" AND "Log Queue Master"."SCADA Interface"."Log Trigger Ack" THEN
8     "Log Queue Master"."SCADA Interface"."Log Trigger" := FALSE;
9     "Log Queue Master"."SCADA Interface"."Log Trigger Ack" := FALSE;
10    "Log Queue Master"."SCADA Interface"."Log Values" := "Log Queue Master"."Log Queue"[1];
11    FOR #Counter := 1 TO 99 DO
12        "Log Queue Master"."Log Queue"[#Counter] := "Log Queue Master"."Log Queue"[#Counter + 1];
13    END_FOR;
14    #Temp := FILL(BVAL := "Log Queue Master".Internal."Zero Log", BLK => "Log Queue Master"."Log Queue"[100]);
15
16 END_IF;
17
18
19 "MC Sys Queue Nr"();
20
21 IF "DB PlcTime".LocalTime.HOUR <> "Log Queue Master".oldHour THEN
22     IF "DB PlcTime".LocalTime.HOUR = 0 THEN
23         "Log Queue Master".Internal."Running Number" := 0;
24     END_IF;
25     "Log Queue Master".oldHour := "DB PlcTime".LocalTime.HOUR;
26 END_IF;
27
28

```

Figur A.4 Structured text i blocket "MC Sys Log Data" och är ett redan programmerat block som är tillhandahållet av Mårtensson Consulting.

Datahantering i mindre fabriker

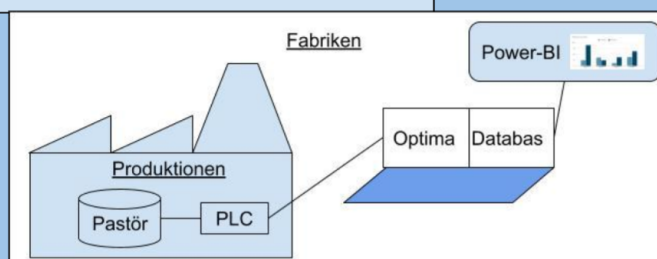


LUNDS
UNIVERSITET
Lunds Tekniska Högskola

Sammanfattning

- Har utförts på Mårtensson Consulting i Kristianstad.
- Arbetet undersöker hur en implementation av insamling och presentation av data i en mindre fabrik kan fungera.
- Detta har utförts genom att studera fabriken implementerade PLC system och variabler samt inspektera en större fabriks redan implementerade databaskommunikation med respektive PLC.
- Arbetet har gått igenom faserna **simulering, integrering och implementering** för att komma till ett resultat.
- Datan kunde, som önskat, kommuniceras upp till databasen och med hjälp av Power BI och presenteras på en interaktivt och intuitivt sätt.

Adrian Olbers, IEA



Problemformulering

1. Hur kan data kommuniceras upp till en databas från en PLC?
2. Hur lagras data på ett strukturerat effektivt sätt i en databas?
3. Hur plockas data ut för visualisering?

Metod

1. Studerade en liknande implementering i större fabrik
2. Simulerade de olika delarna så att de fungerade var för sig.
3. Integrerade de olika systemen att kommunicera och fungera med varandra.
4. Implementerade de olika systemen i fabriken.

Resultat

- Systemet fick de önskade funktionerna
- Power BI kunde presentera data från processerna.
- Det implementerade systemet har möjlighet att utvecklas vidare

Slutsats

- Hur kan data kommuniceras upp till en databas från en PLC?
Genom att använda sig av Mårtenssons egenutvecklade applikation Optima kodat i C#.
- Hur lagras data på ett strukturerat effektivt sätt i en databas?
Genom att lagra datan i en lång lista men med ID för att koppla vilken del av processen datan kommer ifrån och sen använda Power BI för att filtrera och visa datan.
- Hur plockas data ut för visualisering?
Genom att använda sig av Power BI.

Figur A.5 poster för arbete

