

Mange små PLC-er inde i PC-en.

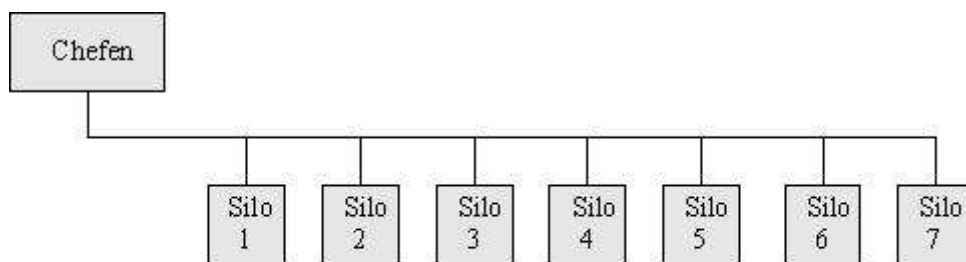
Den typiske applikations-programmør behøver ikke have nogen edb-relateret uddannelse, og ikke have den fjerneste anelse om, hvad ”objektorienteret” betyder. Det koncept brugeren skal forholde sig til, er tæt på at findes i den fysiske virkelighed. Der findes nemlig mange fabrikker, hvor et moderat antal halvstore fysiske PLC-er snakker sammen via kabler.

Det man simulerer med GOPA er - sammenlignet med dette - blot flere og mindre PLC-er. - plus en 100% sikker indbyrdes kommunikation imellem disse (ingen fysiske kabler) og ingen af de fra datakommunikation kendte problematikker som ”handshake”, og hvad der sker, hvis et par af PLC-erne bliver ramt af strømafbrydelse, mens de øvrige arbejder videre o.lign.

I en GOPA-applikation tilstræbes en udveksling af letforståelige meddelelser mellem de små PLC-er, så flest mulige af de involverede interessenter (fra bygherre til operatører) får en fuld forståelse af programmets virkemåde.

Man kan forestille sig de enkelte PLC-er som personer, der har taget opstilling ud for de maskindele, de hver især har ansvaret for. Man opfinder nu nogle spilleregler - herunder mulige ”meddelelser”, disse personer kan sende til hinanden.

I dette eksempel har vi en chef-PLC med ansvaret for en redler med 7 spjæld ned mod 7 siloer. Hvert spjæld har sin egen mini-PLC, som udover at kontrollere spjældet rent fysisk via IO-tabellen (og skærmknapper), adlyder kommandoer fra chef-PLCen.



Stregen der forbinder de 7 mini-PLC-er med chefen, dækker over to interface registre: Kommando og silonummer.

Hver af de 7 silo-spjælds PLC-er har samme kode-indhold. De adskiller sig kun på flg. punkter:

- 1) Silonummeret – hver PLC har sit eget silonummer.
- 2) Linierne i IO-tabellen – hver PLC har sine egne numre i den fælles IO-tabel.
- 3) (option) hver PLC kan have egne linier i den fælles parameter-tabel.
- 4) (option) hver PLC kan have egne skærmknapper m.m. i SCADA-billeder.

Chefen skal kunne åbne en bestemt silo, og efterfølgende lukke for alle andre siloer.

Grundprincippet ved GOPA er, at en chef altid udsender sine kommandoer på generel form til en gruppe hjælpere. Her anvendes overordnede udtryk som ”rigtig”, ”forkert” og ”bedste løsning”. Det er så hjælperne selv, der skal finde ud af om, og hvordan, de skal reagere på disse kommandoer.

Stregerne der forbinder PLC-erne dækker over to registre, der kun er synlige for de PLC-er, der er forbundet. De to registre er kommando (cmd) og silonummer (silo).

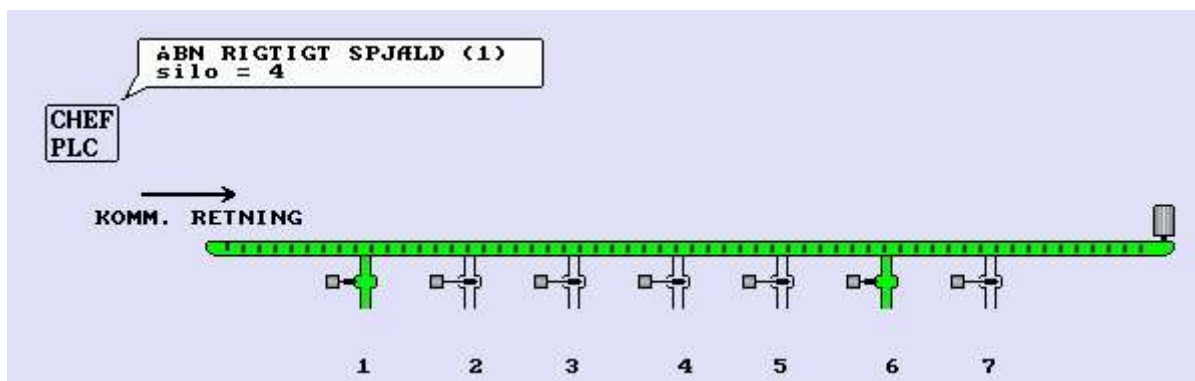
Under tomgang står der nul i kommando-registeret.

Chefen kan udsende kommandoer ved at skrive et tal til dette register "cmd":

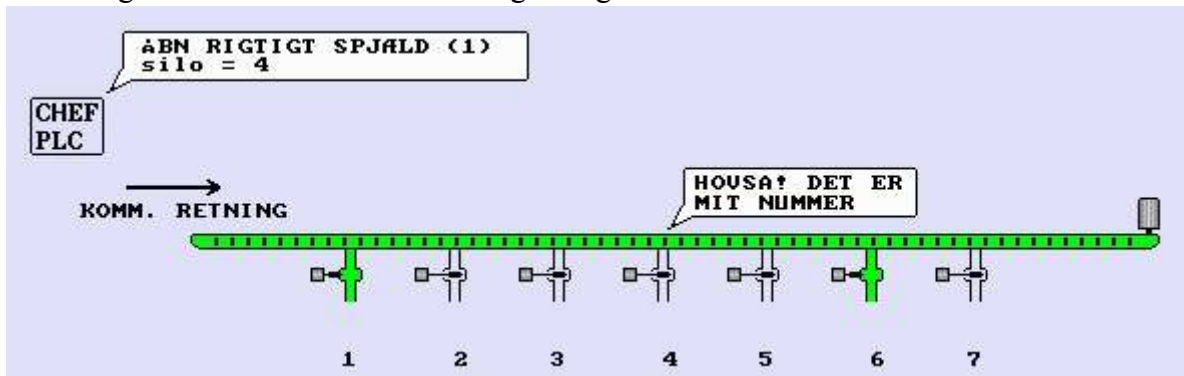
- 1) Åbn rigtigt spjæld (det spjæld hvis nummer står i "silo") og nulstil "cmd" når opgaven er løst.
- 2) Luk forkerte spjæld (dem, der ikke har det nummer, der står i "silo").
- 3) Nulstil "cmd" hvis endnu ikke færdig med at løse opgave 2.

Praktisk eksempel.

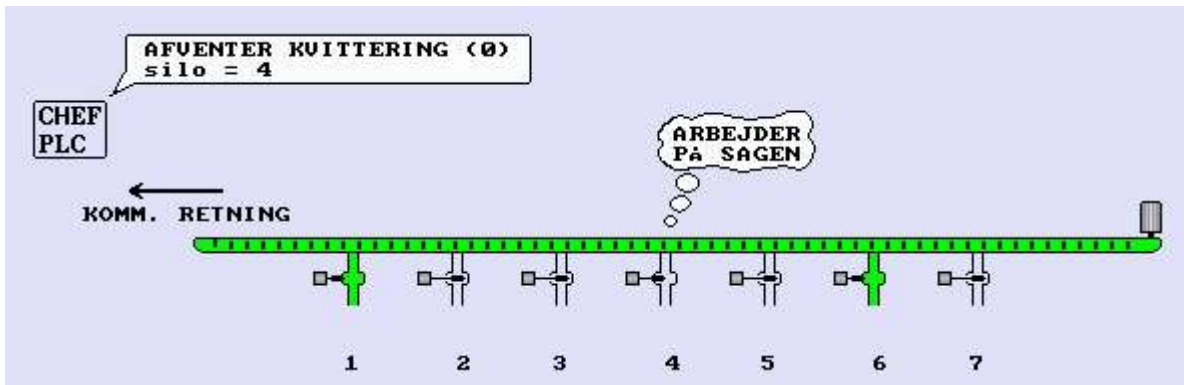
Kommando 1 udsendes sammen med silo nummer 4.



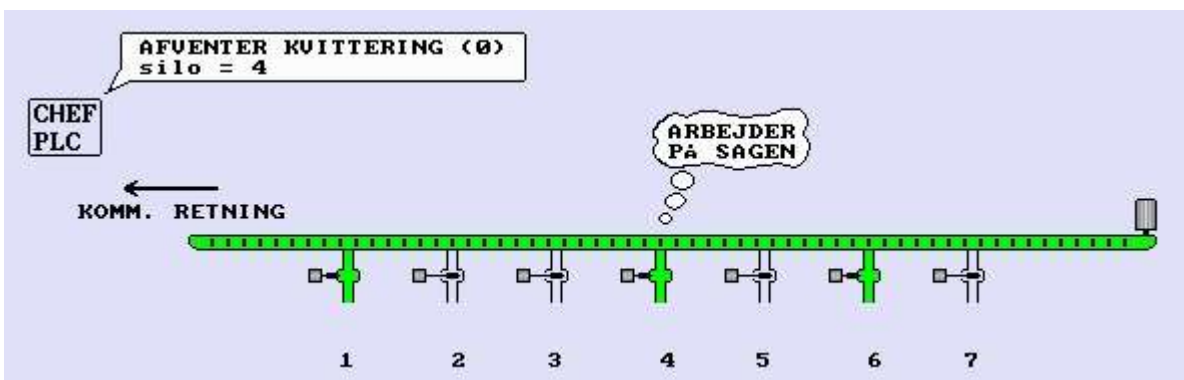
Den silo der genkender nummeret som sit eget reagerer.



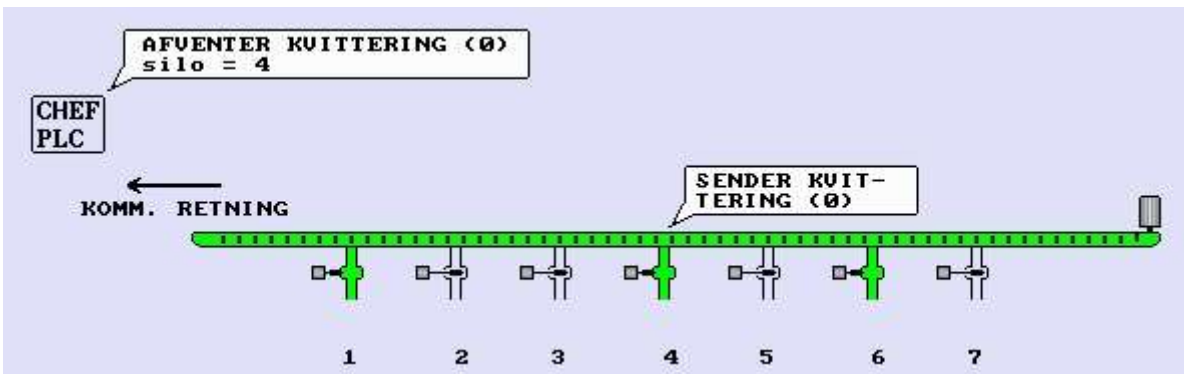
Chefen går i lytte-position, mens spjældet arbejder på sagen.



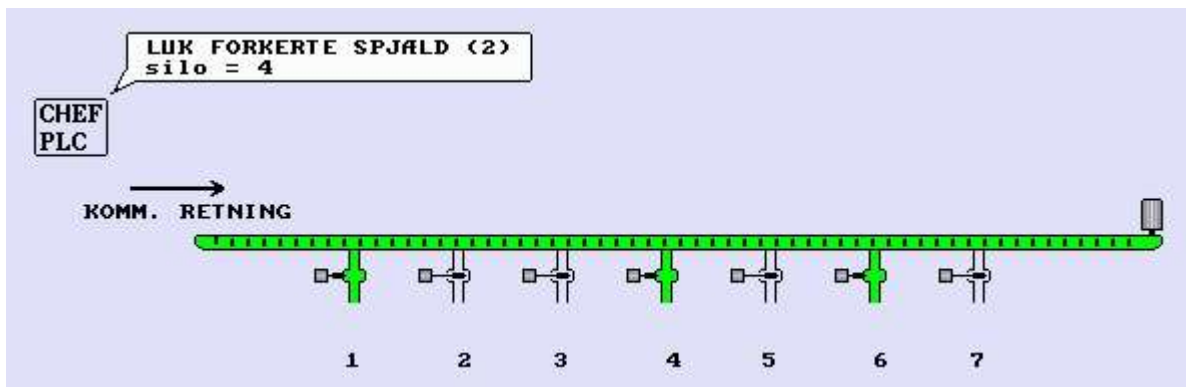
Her kommer den spjæld-åben melding, der fortæller spjældet at opgaven er løst.



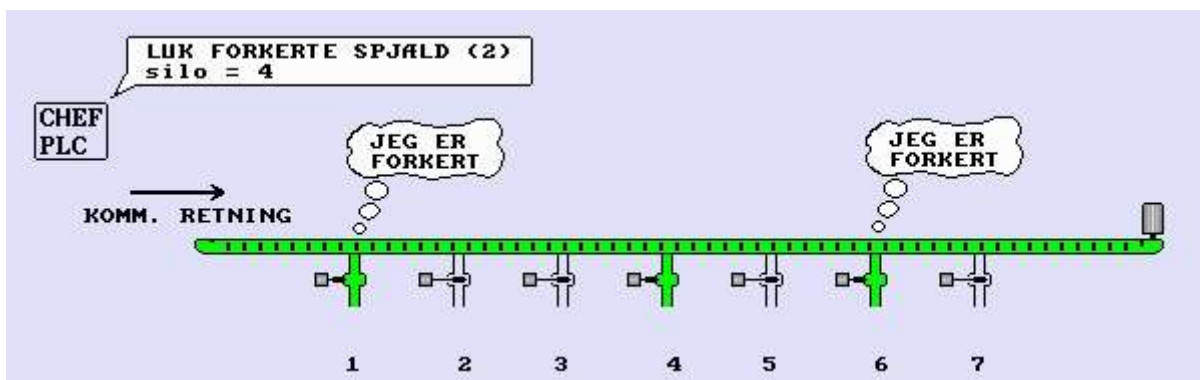
Spjældet fortæller dette til chefen ved at kvittere – d.v.s. nulstille "cmd".



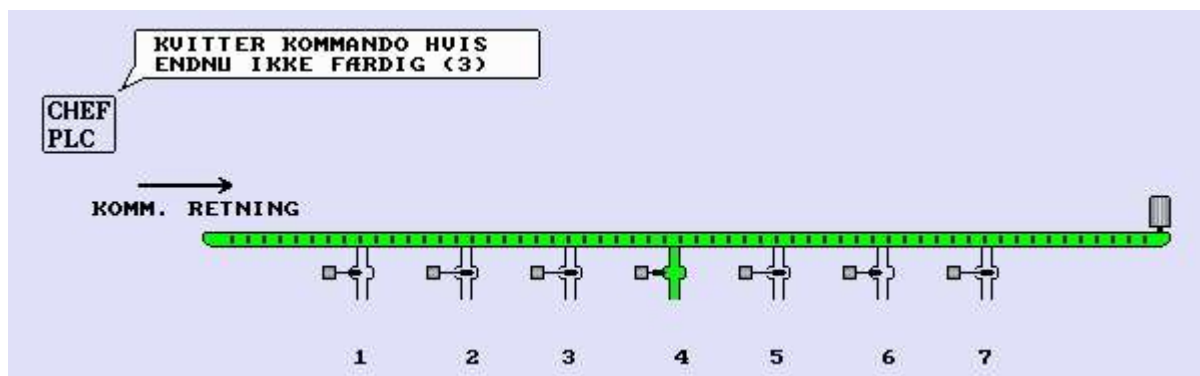
Chefen ved nu at spjældet er åbent, og kommanderer alle øvrige spjæld lukket.



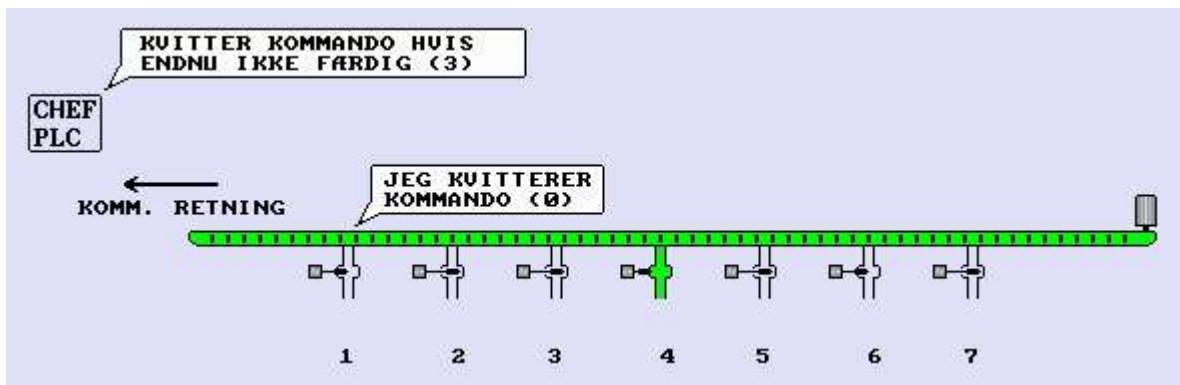
Som sædvanligt finder spjældene selv ud af, hvem der skal reagere på kommandoen. Nemlig når silonummeret IKKE matcher, og der mangler lukket-melding fra spjældet.



Mens de to spjæld 1 og 6 arbejder på sagen, sender chefen kommando-3 – gruppe-forespørgslen, hvor det er meningen, at de spjæld der ”arbejder på sagen”, skal kvittere kommandoen

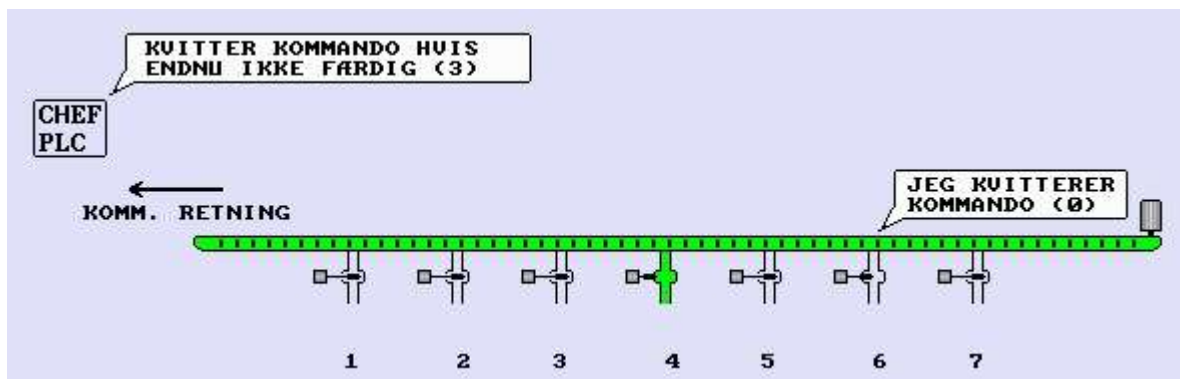


Og her er det så spjæld 1, der kvitterer kommandoen (fordi den tilfældigvis kommer først i scantiden).

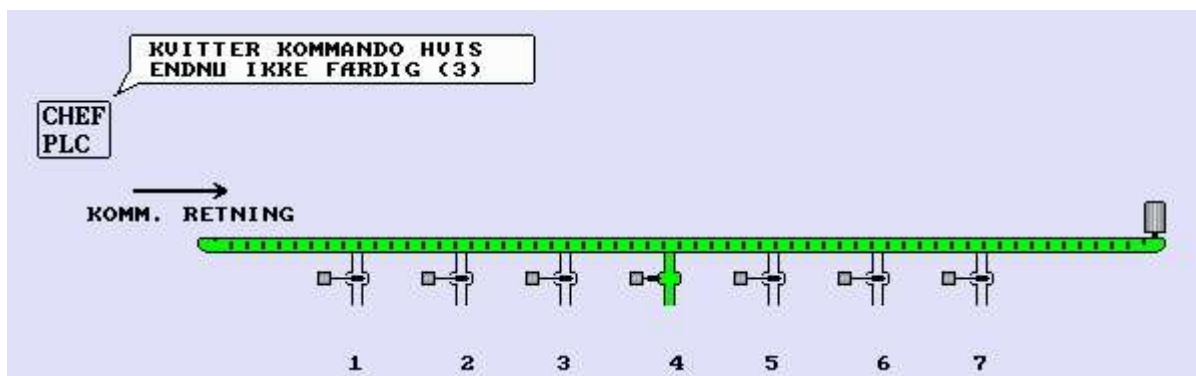


Så længe chefen kan se at de 3-taller, den selv putter i "cmd" registeret i hvert scan, er blevet nulstillet i næste scan, ved chefen, at alle endnu ikke er færdige med at lukke sig.

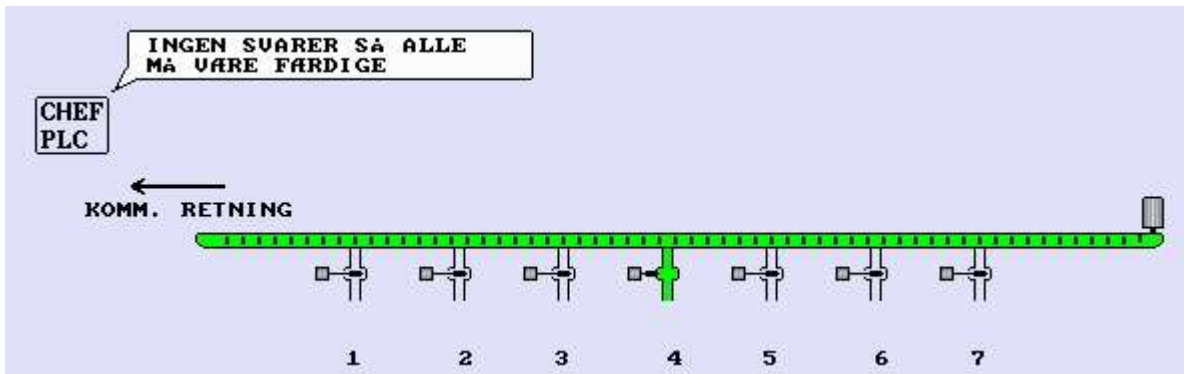
Senere lykkes det for spjæld-1 at blive færdig med opgaven. Da spjæld-1 derfor ikke længere nulstiller 3-tallet, ryger 3-tallet videre til spjæld-6 (der ligger senere i scantiden), som så i stedet nulstiller (kvitterer) kommandoen, fordi spjæld-6 endnu ikke har fået lukket-melding.



Til sidst lykkes det også for spjæld 6 at få sin lukket-melding, så ingen længere kvitterer 3-tallet fra chefen.



Og det opdager chefen allerede i næste scan.



Med denne måde at "snakke sammen" på er det lige meget om der er 7 eller 80 spjæld.