

CODEN: LUTFD2/(TFRT-5451)/1-215/(1992)

PC-styrning av fjärrvärmeverk

Bo Andersson
Joachim Grebe

Institutionen för Reglerteknik
Tekniska Högskolan i Lund
Januari 1992

| | | |
|---|--|--------------------------|
| Department of Automatic Control Lund Institute of Technology P.O. Box 118 S-221 00 Lund Sweden | <i>Document name</i> Master Thesis | |
| | <i>Date of issue</i> January 1992 | |
| | <i>Document Number</i> CODEN: LUTFD2/(TFRT-5451)/1-215/(1992) | |
| <i>Author(s)</i> Bo Andersson and Joachim Grebe | <i>Supervisor</i> Rolf Johansson, Nils Arne Sjödahl | |
| | <i>Sponsoring organisation</i> | |
| <i>Title and subtitle</i> PC-styrning av fjärrvärmeverk. (Computer Control of a Heating Plant.) | | |
| <i>Abstract</i> <p>This master thesis describes a graduation work that was carried out on a heating plant owned by Sydbränsle AB. It describes the resulting control system, the way to final resolutions and conclusions.</p> <p>The goal was the usage of a PC as an operator interface to simplify the supervising and controlling of the heating plant. Man/machine interface would be simplified extensively so that the all untrained computer employers would quickly understand and have use of the system. New software based control functions would be written, and directed to an improved dynamic technology, for the emission levels would reach valid limits. The burning process and the fuel management has therefore been shown special interest. Furthermore there would be a possibility for connection by the telephone net-work from an other place than the heating plant, and from there it could supervise the process and make changes in the process control.</p> <p>The result became a totally new computer-controlled system that completely replaced the existing system. An IBM compatible PC (386 SX) is used for the operator interface. The computer-controlled program for the burning process, is placed in a PLC (SattCon15-20). A separate 386 SX-computer is used for the conveyer program and the PLC are directly connected to the operator-computer from were the process can be adjusted. The system is easy to use and has meant a considerable improvement of further fuel management, CO waste, and the burning control. The system has been in operation since January 1991.</p> | | |
| <i>Key words</i> | | |
| <i>Classification system and/or index terms (if any)</i> | | |
| <i>Supplementary bibliographical information</i> | | |
| <i>ISSN and key title</i> | | <i>ISBN</i> |
| <i>Language</i> Swedish | <i>Number of pages</i> 215 | <i>Recipient's notes</i> |
| <i>Security classification</i> | | |

FÖRORD

Examensarbetet har utförts inom ämnesområdet reglerteknik vid Lunds Tekniska Högskola, LTH.

Arbetet har finansierats av Sydbränsle AB och Sydkraft AB

Högskolelektor Rolf Johansson (LTH) har varit handledare och vi vill tacka honom för det intresse han visat.

Ett stort tack riktar vi också till följande personer, som har varit till stor hjälp under projektets gång:

VD Nils-Arne Sjödahl, Sydbränsle AB

Värmeverkschef Tommy Göransson, Sydbränsle AB

Tekn.Lic. Hans Ottosson, Sydkraft AB

Doktorand Per Persson, LTH

Elektriker Bo-Göran Persson, Sydbränsle AB

Service och support, SattControl AB

Lund i december 1991

Joachim Grebe
Bo Andersson

INNEHÅLLSFÖRTECKNING

| | | |
|----------|--|----|
| 1 | INLEDNING | |
| 1.1 | Bakgrund..... | 7 |
| 1.2 | Syfte..... | 7 |
| 1.3 | Mål..... | 7 |
| 1.4 | Genomförande..... | 8 |
| 1.5 | Omfattning..... | 8 |
| | | |
| 2 | INLEDANDE BESKRIVNING AV ANLÄGGNINGEN | |
| 2.1 | Anläggningen..... | 11 |
| 2.2 | Bränslehanteringen..... | 12 |
| 2.3 | Förbränningsprocessen..... | 15 |
| 2.4 | Ursprunglig styr och reglerutrustning..... | 20 |
| | | |
| 3 | BESKRIVNING AV SYSTEMLÖSNING | |
| 3.1 | Uppdraget..... | 23 |
| 3.2 | Val av systemlösning..... | 23 |
| 3.3 | Det slutgiltiga systemet..... | 27 |
| 3.4 | SattCon 15-20..... | 28 |
| 3.5 | Kommunikationen mellan enheterna..... | 29 |
| 3.6 | Den fysiska inkopplingen av systemet..... | 29 |
| | | |
| 4 | BESKRIVNING AV BRÄNSLEHANTERINGEN | |
| 4.1 | Bränslehanteringen..... | 31 |
| 4.2 | Traversprogrammets parametrar..... | 39 |
| 4.3 | Traversprogrammet och dess procedurer..... | 40 |
| | | |
| 5 | BESKRIVNING AV FÖRBRÄNNINGSPROCESSEN | |
| 5.1 | Förbränningsprocessen..... | 53 |
| 5.2 | Pannprogrammets parametrar..... | 58 |
| 5.3 | Pannprogrammet och dess moduler..... | 64 |
| | | |
| 6 | BESKRIVNING AV OPERATÖRSGRÄNSSNITTET | |
| 6.1 | Operatörsgränssnittet..... | 77 |
| 6.2 | Programutveckling i Matrix Layout..... | 78 |
| 6.3 | Operatörsbilderna..... | 80 |
| 6.4 | Uniten Comlipr..... | 82 |
| 6.5 | Larm och utskrifter..... | 82 |
| | | |
| 7 | BESKRIVNING AV FJÄRRSTYRNINGEN | |
| 7.1 | Fjärrstyrningen..... | 87 |
| 7.2 | Modemen..... | 89 |
| | | |
| 8 | SAMMANFATTNING | |
| 8.1 | Sammanfattning..... | 91 |
| | | |
| 9 | LITTERATURFÖRTECKNING | 96 |

10 BILAGOR

- 10.1 Bilaga Traversprogrammet
- 10.2 Bilaga Pannprogrammet
- 10.3 Bilaga Operatörsgränssnittet
- 10.4 Comli-unit för operatörsgränssnitten
- 10.5 Gränssnittet och signalerna ut mot anläggningen
- 10.6 Installering, ominstallering och ändringar i programmen
- 10.7 Den fysiska inkopplingen av systemet

INLEDNING

1.1 BAKGRUND

Sydbränsle AB är ett dotterbolag inom Sydkraftkoncernen och svarar för affärsområdet inhemska fasta bränslen.

Sydbränsle äger och driver ett fjärrvärmeverk i Ljungby, Småland. Värmeverket har varit i drift sedan 1986, men har endast drivits i Sydbränsle AB:s regi sedan 1990.

Verket har en fastbränslepanna på 6 MW samt en oljepanna på 5,8 MW. 1990 levererade man totalt ut 22000 MWh. Fjärrvärmes levereras till närliggande industrier, kontorslokaler och hotell. Totalt 14 kunder är anslutna till fjärrvärmenätet.

Fastbränslepannan eldas med olika biobränslen, huvudsakligen bark, men vintertid även putsdamm, sågspån och flis. Leveranser sker främst från närliggande sågverks- och snickeriindustrier.

Denna rapport är en sammanfattning av ett examensarbete utfört på värmeverket med syfte att förbättra styr- och reglerteknik för produktionssystemet.

1.2 SYFTE

Ersätta befintlig styr- och reglerteknik med operatörsorienterad bildskärmsteknik. I uppgiften ingick även implementering av fjärrstyrning och utvärdering av datoriseringen.

1.3 MÅL

Analysera möjligheten att driva tekniken ett steg längre än vad dagens traditionella motsvarighet innebär och utnyttja ett PC-baserat system med pekverktyg för operatörsövervakningen. Människa/maskin gränssnittet skulle förenklas så långt att helt datortränad personal kan använda och förstå systemet.

1.4 GENOMFÖRANDE

Projektet startade sommaren 1990 med inhämtande av kunskaper om värmeverkets drift och befintliga styr- och reglerteknik

Under höstterminen -90 gjordes programmen för pannstyrningen, bränslehanteringen och operatörsövervakningen. Parallellt med utvecklingen av programmen genomfördes den fysiska inkopplingen av vår hårdvara på det befintliga styrsystemet. Inkopplingen gjordes av behörig elektriker.

Hårdvaran består av en fast PC (386SX) för operatörskommunikation, ett traditionellt PLC system för pannprogrammet och en separat PC (endast dator) för traversstyrningen.

Vårterminen -91 gjordes så driftsättningen och därefter korrigeringar, förbättringar och eliminering av direkta programbuggar.

Under sommarmånaderna genomfördes framtagningen av ett fjärrstyrningssystem för mobil modembaserad PC för 'hemmabruk'.

Under höstterminen -91 gjordes en utvärdering av den nya styr- och reglertekniken vilket finns sammanfattat i denna rapport.

För närvarande används vårt styrsystem vilket varit i drift sedan ett år tillbaka. Det har helt ersatt det tidigare systemet och såväl förbättrat som underlättat driften.

1.5 OMFATTNING

I denna avhandling presenteras det nyutvecklade styr- och regler systemet. En kortfattad beskrivning av avhandlingens omfattning redovisas nedan.

I kapitel 2 ges en inledande beskrivning av värmeverket, bränslehanteringen, förbränningsprocessen och ursprunglig styr- och regler utrustning.

Kapitel 3 beskriver uppdraget och valet av systemlösning.

Kapitel 4 beskriver framtagningen och lösningen av bränslehanterings programmet.

Kapitel 5 beskriver framtagningen och lösningen av pannstyrnings programmet.

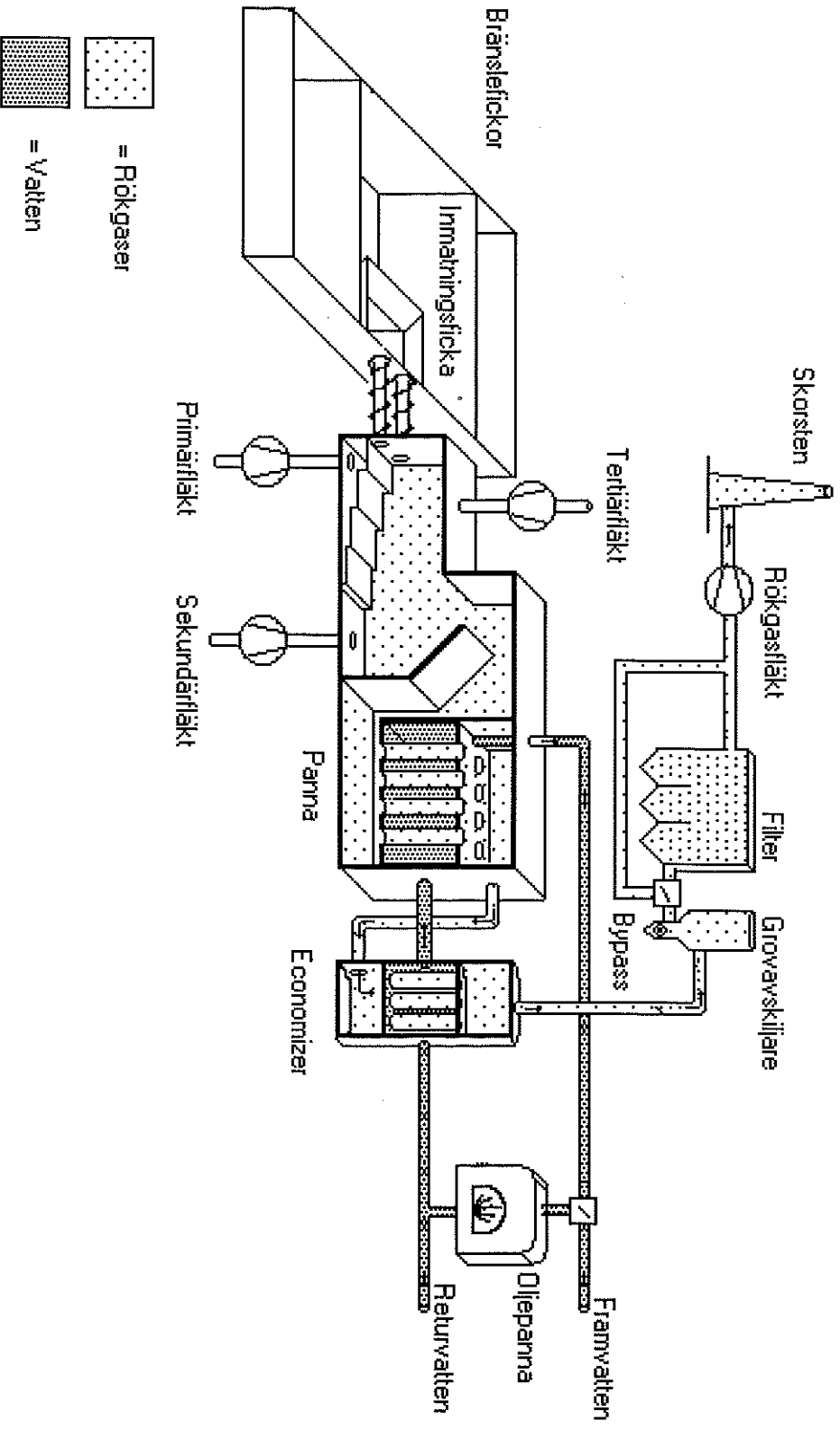
Kapitel 6 beskriver framtagningen och lösningen av operatörsgränssnittet.

Kapitel 7 beskriver lösningen av fjärrstyrningen.

Kapitel 8 innehåller sammanfattning

Kapitel 9 innehåller en lista på den referenslitteratur som använts.

Kapitel 10 innehåller bilagor där bl.a. pannstyrningsprogrammet och bränslehanteringsprogrammet är listade i sin helhet.



Processchema Ljungby Fjärrvärmeverk

INLEDANDE BESKRIVNING AV ANLÄGGNINGEN

2.1 ANLÄGGNINGEN

Verkets hjärta är en panna som eldas med olika avfallsprodukter från närliggande sågverksindustrier. Man eldar framförallt med bark, men även med flis, sågspån och dylikt som p.g.a fukthalter upp till 65 % är billigt. Vattentemperaturen på levererat vatten bör ligga omkring 110 grader C. Vintertid vid stort effektbehov eller vid driftstopp används en oljepanna för att komplettera fastbränslepannan.

På sommaren kan kundernas effektbehov vara mycket lågt, medan kundbehovet kalla dagar klart överstiger pannanläggningens kapacitet. Stora variationer av värmeuttaget på nätet sker dagligen eftersom kundernas värmebehov till mycket stor del är temperaturberoende och sjunker drastiskt vid arbetstidens slut. Då verket ej klarar av att leverera erforderlig effekt, måste kunderna själva gå in och stötta upp med olja. Verkets process kan delas in i fyra olika steg.

1. Bränslehanteringsdel

Denna del har hand om transporten av bränslet från lagerfickorna till en bränslebehållare varifrån det sedan matas in i pannan. Den består av fyra lagerfickor och en travers med en gripskopa på 2 m³.

2. Förbränningsdel

Denna del är den värmeproducerande och uppvärmande delen. Den består av inmatningsskruvar, en eldstad, en panna, diverse spjäll och fläktar och askutmatningsskruvar.

3. Rökssystem

Denna del har till uppgift att avskilja stoft i rökgaserna och därefter leda rökgaserna ut i det fria. Rökssystemet består av rökkanal, stoftavskiljare, stoftutmatningsdon, dragregleringsanordning, rökgasfläkt och skorsten.

4. Fjärrvärmenät

Denna del levererar det uppvärmda vattnet. Två stora cirkulationspumpar pumpar runt vattnet i en kulvert. På kulverten vid varje kund sitter en ventil, samt mätinstrument som registrerar framtemperatur, returtemperatur och flöde. Denna del kommer vi inte att beröra mer i denna rapport, eftersom det ligger utanför examensarbetet.

2.2 BRÄNSLEHANTERINGEN

Med bränslehantering avses transporten av bränslet från lagerfickorna till bränslebehållaren. Därifrån matas i sin tur bränslet med hjälp av två skruvar in i pannan där det förbränns. Påfyllning av lagerfickorna sker antingen direkt via tippning från lastbilscontainrar eller med hjullastare från utomhuslager. Det finns totalt fyra lagerfickor. Dessa benäms A, B, C resp D.

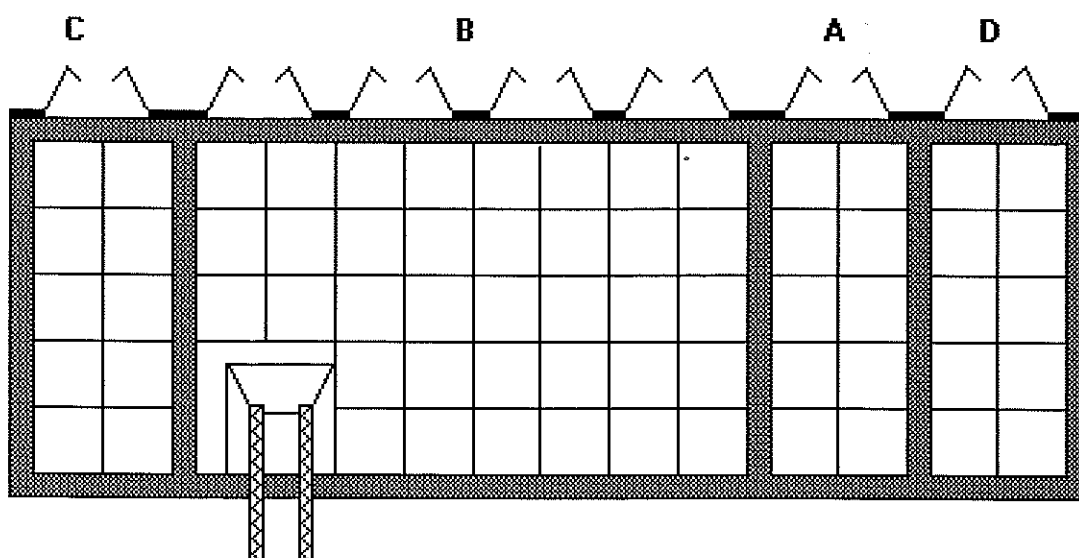


Fig.2 Översiktsbild på lagerfickorna.

Inmatning från fickorna till bränslebehållaren sker m.h.a en travers vars gripskopa har en kapacitet på 2 m³. För att vara mer precis är traversen den del som svarar för rörelsen i 'X-led' medan trallan som löper ovanpå svarar för rörelsen i 'Y-led'.

För att registrera att traversen passerat en viss punkt används induktiva givare. I 'X-led' finns 14 givare och i 'Y-led' finns fem givare. Tillsammans bildar de en matris med inalles $5 \times 14 = 70$ koordinater. B svarar för 40 koordinater medan A, C och D svarar mot 10 koordinater vardera.

För att minska påfrestningar vid uppstart och stopp har drivmotorn på traversen och trallan två hastigheter. Dessa benäms lågfart resp högfart. Om traversens rörelse är längre än över en koordinat, går den upp i högfart. Vid stopp går traversen först ner i lågfart.

Dessa funktioner programmeras varvid man använder sig av de tre läshuvuden som finns på traversens rörliga del. Det mellersta läshuvudet ger signal när traversen är mitt för en koordinat. De båda andra skulle man kunna säga förvarnar för att man snart passerar mittpunkten. Vilket av de båda yttre läshuvuden man 'läser på' beror på i vilken riktning traversen rör sig.

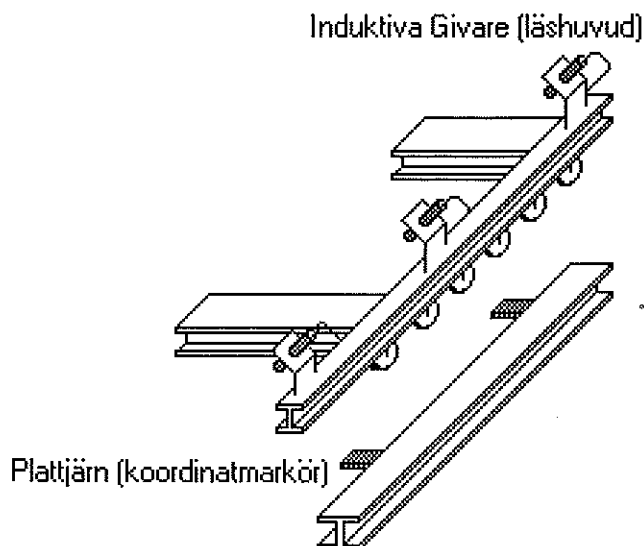


Fig. 3 Traversdel med induktiva givare.

Motsvarande läshuvud finns på trallan. Funktion och användning överensstämmer med traversens.

Bränslebehållaren är sista ledet i inmatningskedjan. Efter att traversen släppt sin last i denna sker därifrån en sluten inmatning av bränslet in i pannan via två skruvar i botten på bränslebehållaren.

Ett problem med inmatningen är valvbildning. Denna uppkommer när bränslet ej 'faller ner' i skruvarna vilket resulterar i att trots skruvarna rör sig matas inget bränsle in i pannan.

En roterande och en hydraulisk omrörare används för att minska valvbildningen. Den hydrauliska omröraren (vi benämner denna bragg) kan sägas vara en rörlig del av bränslebehållarens botten, medan den roterande omröraren är ett rör med mindre skovlar vilka löper parallellt med inmatningsskruvarna (se fig. 4).

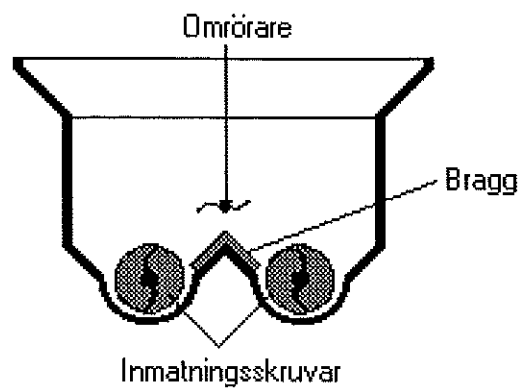


Fig. 4 Bränslebehållaren

En ultraljudsgivare används för att känna av bränslenivån i behållaren. Denna ger signal vid behov av påfyllning. Givaren är inställbar för önskad lägsta bränslenivå.

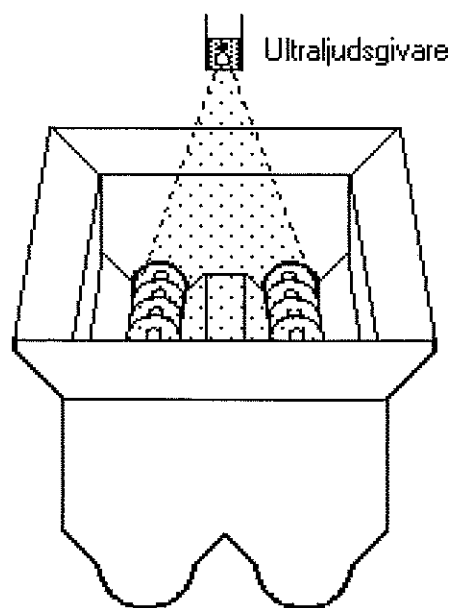


Fig. 5 Ultraljudsgivarens placering

2.3 FÖRBRÄNNINGSPROCESSEN

Denna del förbränner bränslet och hettar upp vattnet till fjärrvärmenätet. En översiktsbild visas i figur 6.

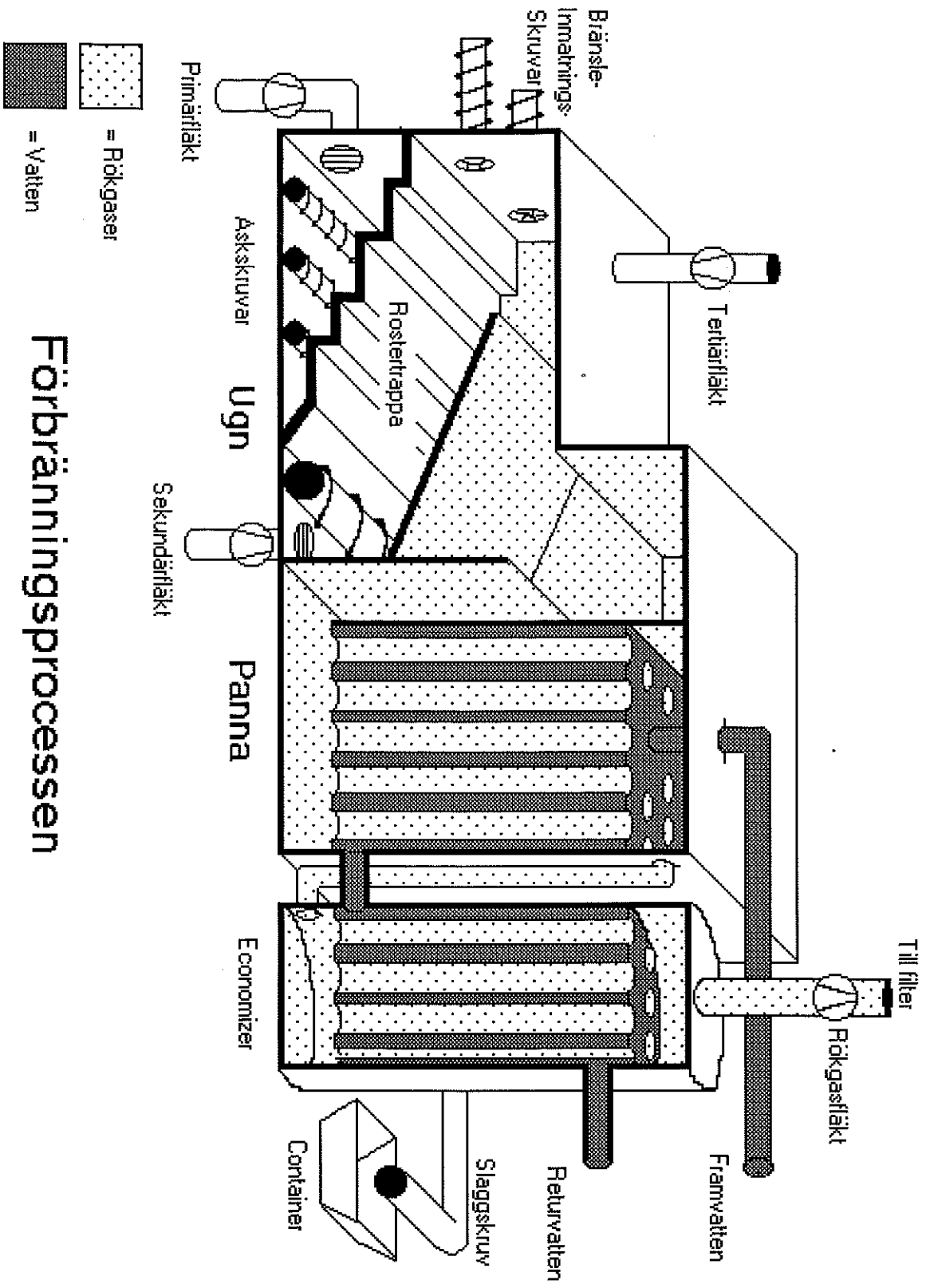
Två begrepp som är viktiga att skilja åt är ugn och panna. En ugn består av en eldstad, medan pannan är den del där värmen i rökgaserna överförs till vattnet. Lite felaktigt brukar man kalla ugn och panna för endast panna.

Bränslet skruvas in i pannan och faller ner på rostertrappan. Under rostern sitter primärluftfläktens utblås som förser förbränningsprocessen med syre. Rostern rör sig fram och tillbaka vilket medför att bränslet allteftersom det torkar och förbränns, förflyttar sig framåt för att slutligen falla ner på askutmatningsskruvarna som transporterar askan till en container. De upphettade rökgaserna får passera i kanaler som omges av det vatten som skall värmas upp. Temperaturen inne i ugnen är omkring 800 grader C. När rökgaserna passerat värmeväxlaren håller rökgaserna en ungefärlig temperatur på 140 grader C. Efter det att rökgaserna har avgett sin värmeenergi passerar de en grovavskiljare och ett slangfilter för att sedan via en rökgasfläkt passera ut genom skorstenen.

Nedan studeras varje funktion och objekt i förbränningsprocessen.

Inmatningsskruvar: Inmatningsskruvarna är två till antalet och matar in bränsle i pannan. Med inmatningsskruvarna regleras mängden bränsle i eldstaden.

Roster: Rostern består av en trappa där varannat trappsteg kan röra sig horisontellt. Med rostern reglerar man hur snabbt bränslet skall passera eldstaden och indirekt tjockleken på bränslebädden. Ju oftare rostern rör på sig, desto snabbare transporteras bränslet vidare. På varje trappsteg finns det smala springor som gör att primärluften kan passera och även en del aska falla ner.



Förbränningsprocessen

Filter: Filtret innan skorstenen är ett så kallat slangfilter. Detta består av av 216, 3,6 meter långa brandslangliknande strumpor. Alla rökgaser passerar genom strumporna varvid de minsta partiklarna i gasen fastnar på strumpornas utsida. Efter en viss tid har så pass många partiklar fastnat att man måste rensa filtret. Rensning sker med luft från en kompressor ungefär var 10:e minut. Luften blåses uppifrån och ner, partiklarna lossnar och trillar ner till en transportskruv som för askan vidare till en container.

Miljökrav: Miljökraven idag sätter gränser på stofthalten och CO-halten i rökgaserna. Kraven lyder enligt följande:

- 1 Utsläppet av stoft från skorstenen får som gränsvärde inte överskrida 100 mg/m³ gas.
- 2 Utsläppet av kolmonoxid från skorstenen får som riktvärde ej överskrida 100 ppm.

Ljudsotning: Vattnet värms upp av rökgaserna, som passerar genom kanaler i pannan. Dock så fastnar det sot på kanalerna, vilket gör att värmeväxlingen blir sämre. Problemet löser man med ljudtyfoner som med några minuters mellanrum ljudligt och hörbart skakar loss soten, varefter den trillar ner för att transporteras bort. Sotning av kanalerna sker c:a 5 ggr/år.

Rökgasfläkt: Denna fläkt skapar ett undertryck i pannan (jämför dammsugare). Fläkten drar ut rökgaserna ur eldstaden genom pannan, via filtret och ut i skorstenen. Framför fläkten sitter ett spjäll som man kan variera luftflödet med. Med detta spjäll styr man eldstadsundertrycket i ugnen. Anledningen till att man använder spjäll i stället för frekvensstyrning (varvtalsstyrning) av motorerna är att frekvensstyrning är dyrt vid stora effekter (c:a 70 kW).

Primärfläkten: Denna fläkt tillför förbränningen huvuddelen av allt syre. Primärfläkten sitter under rostertrappan. Denna fläkt styr effekten i pannan. Framför fläkten sitter det ett spjäll som man reglerar luftflödet med.

Sekundärluftfläkt: Denna fläkt tillför syre till de gaser som ännu inte brunnit upp. Den finjusterar alltså O₂-halten i pannan. Om man t ex eldar med mycket torrt och fint bränsle kan det uppstå en kraftig syrebrist i pannan. Sekundärluften reglerar alltså O₂-halten i eldstaden.

Tertiärluftfläkt: Denna fläkt är den tredje och sista fläkten som tillför O₂. Den skall framför allt ses som en sista utväg att oxidera rökgaserna ytterligare om sekundärfläkten inte räcker till. Framför fläkten sitter ett spjäll. Detta spjäll regleras på CO-halten i pannan om anledningen till för hög CO-halt är brist på syre. Om anledningen till en hög CO-halt är att man använder för fuktigt bränsle kan man inte göra någonting åt den.

Bypasspjäll: Om rökgastemperaturen innan filtret är så pass hög att man riskerar att filterstrumporna börjar glöda och förstöras så måste man för att rädda filtret, leda rökgaserna förbi filtret. Detta gör man med ett bypasspjäll. Om pannan är i offläge d.v.s att pannan är tillfälligt avstängd (under On-Off körning), så leds röken förbi filtret (bypassas) för att få bättre självdrag. Med bättre självdrag får glöden i pannan mera luft och pannan kan stå i off-läge längre. Anledningen till att man inte behöver filtrera rökgaserna när fläktarna inte är på, är att förbränningen är liten och ytterst lite sot bildas.

Askutmatningsskruvar: När bränslet brunnit upp, faller en mindre del aska ner genom springor i rostertrappan. Den större delen av askan transporteras bort av skruven nedanför rostertrappan. Totalt fyra askskruvar matar ut askan på sidan av ugnen. Därefter tar slaggskruvarna vid för att mata ut askan till en container.

Slaggskruvar: Dessa skruvar transporterar askan från sidan av ugnen till en container.

Economizer: Värmeväxlingsprocessen består av två steg. Första steget är i själva pannan. Rökgaserna har här en temperatur på c:a 800 grader C, när de går in i pannan och ungefär 300 grader C när de lämnar pannan. Därefter passerar rökgaserna economizern och får en temperatur omkring 140 grader C. Vattnet passerar de två stegen i omvänd ordning. Först när vattnet är som kallast, passerar det economizern för att sedan något uppvärmt passera pannan. På detta sätt utnyttjar man temperatur-differanserna mellan rökgaserna och vattnet bättre och får bättre verkningsgrad.

Oljepanna: Om man skulle få långvariga avbrott i värmeproduktionen t ex vid revision, reparationer eller ovanligt högt effektuttag, finns det möjlighet att helt eller delvis hjälpa upp framvattentemperaturen med oljeförbränning. Detta är emellertid en förlustaffär varför man vill undvika det så mycket så möjligt. Riktigt kalla vinterdagar måste man dock häpa till med olja för att kunna hålla leveransavtalen.

Några processdata:

| | | | |
|------------------------|-------------------|-------|---|
| CO-halt | ppm | 100 | CO-halten i eldstaden |
| O ₂ -halt | % | 5 | O ₂ -halten efter förbränningen |
| Delta-P | Pa | 40-60 | Tryckskillnaden över bränslebädden |
| Eco-gastemp | C | 140 | Rökgastemperaturen efter economizern |
| Eldstadsundertryck | Pa | 150 | Tryckskillnaden mellan eldstaden och mediat utanför pannan |
| Panntemp | C | 110 | Framtemperaturen på vattnet efter panna och economizer |
| Framtemp | C | 110 | Framtemperaturen efter oljepannan. Denna temperatur är den levererade |
| Returtemp | C | 70 | Temperaturen på returvattnet |
| Rökgastemp | C | 300 | Rökgastemperaturen mellan panna och economizer |
| Ugnstemp | C | 700 | Temperaturen inne i eldstaden |
| LägeRökgasspjäll | % | 100 | Rökgasspjällets läge |
| LägePrimärluftspjäll | % | 100 | Primärluftspjällets läge |
| LägeSekundärluftspjäll | % | 100 | Sekundärluftspjällets läge |
| LägeTertiärluftspjäll | % | 100 | Tertiärluftspjällets läge |
| Effekt | kW | 8000 | Levererad effekt |
| Flöde | m ³ /h | 150 | Vattenflödet på fjärrvärmenätet |

Larm: Erhålls som felindikationer på det som man bedömt vara viktigt ur säkerhets- och driftmässiga synpunkter. En del är naturligtvis mycket allvarliga, medan andra är mindre allvarliga. En del larm stänger av hela anläggningen med omedelbar verkan.

Man skiljer på A- och B-larm. A-larm betyder att fara för anläggningen föreligger, medan B-larm betyder att produktionen är hotad. För jourpersonal gäller att de måste ta sig till anläggningen inom 30 min efter det att ett A-larm har gått. Anläggningen är bemannad dagtid men nattetid är man hänvisad till jourpersonal som är tillgängliga i sin bostad. Larmen är kopplade till en telecentral som ringer upp den som har juren när ett larm har gått. Larmtelecentralen anger inte vilket larm eller vilken typ (A eller B) som har indikerats, utan endast att ett larm har gått. Hittills har detta medfört att jourpersonal har fått ta sig till anläggningen även när helt ofarliga larm har utlösts.

2.4 URSPRUNGLIG STYR OCH REGLERUTRUSTNING

Traversen: Traversen ursprungliga reglering sköttes av en PLC av märket Telemecanique, typbeteckning TSX 47.

Användargränssnittet bestod av tre strömställare på en panel, där man hade möjlighet att välja inmatning från ficka A, B eller C ; eller inmatning med en skopa från B och en från A eller C. Observera att ficka D inte finns med i styrprogrammet. Inmatning och omlastning i D-fickan görs manuellt.

Ytterligare tre strömställare fanns för omlastning, en för vardera ficka.

Panna: Det ursprungliga styr- och reglersystemet levererades av Järnforsen AB. Principen för denna utrustning bygger på reläer samt specialkomponenter för reglerloopar. Reglerutrustningens användargränssnitt består av presentation av processvärde på en panel, där man även har möjlighet att sätta på/stänga av delar av processen. På panelen finns det även indikeringslampor.

För att ändra styrparametrar eller nivåer finns det i en del fall möjlighet att göra det via potentiometrar, i andra fall möjlighet att göra det genom att ställa in tidreläer, termostater etc.

Följande principer användes för styrning av pannan.

| | |
|------------------------|--|
| Inmatningsskruvar: | Tidrelästyrning (4 olika tidreläer m.a.p effekt) |
| Bränsleomrörare: | Kopplat med inmatningsskruvar |
| Roster: | Tidrelästyrning (4 olika tidreläer m.a.p effekt) |
| Filter: | Tidrelästyrning |
| Ljudsotning: | Tidrelästyrning |
| Rökgasfläkt: | Automatik On/Off läge, Y-D start (75kW) |
| Rökgasspjäll: | Pulsregulator för eldstadsundertrycket |
| Primärfläkt: | Automatik On-Off läge, Y-D start |
| Primärluftspjäll: | Kapacitetsregulator (4 läge m.a.p effekt) |
| Sekundärfläkt: | Automatik On/Off läge, Y-D start |
| Sekundärluftspjäll: | Pulsregulator för O ₂ -halten |
| Tertiärfläkt: | Kopplat med sekundärfläkten |
| Tertiärluftspjäll: | Kopplat med sekundärluftspjället |
| Bypasspjäll: | Bypass när pannan är i offläge |
| Askutmatnings-skruvar: | Tidrelästyrning på alla 3 skruvar |
| Slaggskruvar: | Tidrelästyrning på alla 3 skruvar |
| On/Off status: | Termostat med hysteres |

Säkerhetssystemet: Utgörs av ett separat system som för anläggningen till ett säkert tillstånd, oberoende av vad styr och regler systemet gör. Principen för säkerhetssystemet är uppbyggd på brytande signalreläer.

Kapacitetsregulator: Kapacitetsregulatorn är mycket vanlig när man styr pannor i fjärrvärmeverk. Den är egentligen inte reglerande, snarare styrande. Den fungerar som så att man ställer in fyra olika område på framtemperaturen med en potentiometer. Exempelvis :

<95 C, 95-100 C, 100-110 C, > 110 C.

Till dessa olika nivåer kan man sedan sätta ett fast spjälläge på primärspjäll och sekundärspjäll. Sedan kan man även ställa in inmatningsperioder på skruvarna, rostern samt askutmatningen.

Sammanfattningsvis kan man säga att denna styrmetodik kan fungera bra om man alltid använder exakt samma sorts bränsle. Detta gör tyvärr inte Ljungbyverket vilket därför diskvalificerar denna regulator typ i detta sammanhang.

BESKRIVNING AV SYSTEMLÖSNING

3.1 UPPDRAGET

Syftet med examensarbetet var att ersätta värmeverkets befintliga styr- och reglerteknik med operatörsorienterad bildskärmsteknik.

Människa/maskin gränssnittet skulle förenklas så långt att helt datorotränad personal snabbt skulle kunna förstå och använda sig av systemet.

Styr- och reglerfunktionerna skulle inriktas mot förbättrad dynamisk teknik för att utsläppsnivåerna skall nå gällande gränsvärden. Förbränningsprocessen och bränslehanteringen ägnas därmed speciellt intresse.

Det nya styr- och reglersystemet skall arbeta helt parallellt med det ursprungliga som man vid behov skall kunna återgå till.

Alla larmfunktioner skall bibehållas oförändrade liksom nödriftläget.

Operatören skall via mobil eller fast PC kunna koppla upp sig mot värmeverket via telenätet för möjlighet till övervakning och parameterinställningar.

Så länge vi följde ovanstående och dessutom höll oss inom budgeten, hade vi helt fria händer vid val av systemlösning.

3.2 VAL AV SYSTEMLÖSNING

Beskrivningen av anläggningen kan sägas vara resultatet av första delen i vårt arbete. Nämligen det att sätta sig in i värmeverket för att förstå anläggningen, hur den styrdes tidigare, dess svaga sidor såväl som fungerande.

Nästa steg var att dela upp examensarbetet i naturligt avgränsade delprojekt för att därefter fastställa hur dessa delar sinsemellan skulle samverka.

Inledningsvis fanns det tre klart skilda program ur styrregler- och övervakningsteknisk synpunkt.

Nämligen:

- Traversprogrammet
- Pannprogrammet
- Operatörsgränssnittet

Den ursprungliga idén gick ut på att låta all mjukvara ligga i en och samma PC-maskin. Friheten vad gällde val av operativsystem och programspråk var då total. Därutöver skulle en yttre enhet tillkomma för att distribuera de dryga 100-tal in- och utsignaler det var frågan om.

En PC-maskin har ett begränsat antal I/O portar, oftast inte mer än två. Dessa utgörs av det seriella RS232 gränssnittet. För att kunna hantera ett större antal signaler behövs det ett fristående yttre buss-system med egen CPU för att administrera alla signaler. Man kan säga att man utökar antal I/O portar till PC:n. Kommunikationen mellan PC:n och buss-systemet sker seriellt mellan de båda enheternas RS232 portar via ett fördefinierat protokoll.

Till PC:n måste väljas ett operativsystem och närmast till hands är då DOS. Valet av DOS medför tillgång till ett tillnärmelsevis obegränsat sortiment av applikationsprogramvaror.

Största nackdelen med DOS är dess minnesbegränsning vad gäller storleken på de program som körs under operativsystemet. Därutöver saknar DOS möjligheten att köra mer än ett program åt gången.

På det första problemet finns flera lösningar och den vi valde går ut på att man delar upp sitt program i flera mindre delar s.k unitar. Dessa delar kan man sedan kompilera var för sig. Problemet man kommit förbi är att högnivåprogrammet måste laddas ner i det begränsade RAM minnet där det med hjälp av kompilatorn skall översättas till den mycket mindre minneskrävande maskinkoden. Maskinkoden är det program som består av ettor och nollor och som motsvarar det slutliga och körbara programmet.

Andra problemet, det att DOS saknar möjlighet att köra mer än ett program åt gången, är mindre lättlost. Vad man behöver är ett fristående program som klarar av att administrera flera program samtidigt. Det enda program vi hittade heter Omniview eller på svenska 'mångfald'. Funktionen i korthet är den att Omniview kör ett program i taget, men under mycket korta sekvenser och utåt ser det ut som om flera program exekveras samtidigt.

Det programutvecklingsspråk vi ursprungligen tänkte använda för att skriva våra styrprogram resp operatörsgränssnittet i, heter MODULA-2 och kan sägas vara en 'lillebror' till Pascal. Det är ett i PC sammanhang mycket kraftfullt språk som innehåller en egen realtidskärna med möjligheten att köra parallella underprogram samtidigt. Från reglerinstitutionen på LTH skulle vi kunna använda oss av färdigskrivna moduler, såväl regler som bildbyggnadsblock.

Detta var vår ursprungliga idé till systemlösning. Kvar fanns dock ett problem att lösa; vilket system skulle vi använda för att administrera in- och utsignalerna?

På värmeverket fanns redan ett datorsystem för mätvärdesinsamling och övervakning av kundsidan med avseende på framvatten-, returtemperatur och ventillägen. Mätvärdena lagras på hårddisk och kan presenteras grafiskt som trendkurvor via skärm eller plotter. Datoren är en 286:a och för att administrera in- och utsignalerna används ett buss-system som heter SIOX vilket är en förkortning för Serial Input And Output Exchange.

Att använda sig av samma buss-system till vårt system skulle kunna vara fördelaktigt. Två problem framkom emellertid med klar tydlighet. För det första så fanns det inget kommunikationsprotokoll skrivet i MODULA-2, för det andra så skulle det vara svårt att hinna med att distribuera de nästan hundra signalerna någorlunda strikt m.a.p. realtidskraven.

Rolf Johansson, vår handledare på LTH, rekommenderade oss att närmare undersöka SattControls buss-system. Han tyckte det var olämpligt att placera pannstyrningsprogrammet i en PC. Efter lite marknadsundersökningar visade det sig att SattControl var prismässigt konkurrenskraftigt och betydligt flexiblare än SIOUX.

Det av SattControls system som bäst passade våra krav var SattCon 15-20. Grundenheten är ett rack med kraftenhet och centralenhet med programminne. Racketen har plats för 16 processinterface t.ex I\O kort. Systemet programmeras med ett av SattControl utvecklat språk, PBS, vilket är ett maskinvaronära språk specialanpassat för styr- och reglering. Kommunikation med omvärlden sker via två RS232 portar med ett kommunikationsprotokoll kallat Comli.

Problemet med programspråk till PC:n kvarstod i och med valet av SattControl. Comliprotokollet, som är det enda sättet att kommunicera med SattCon från PC-DOS maskiner fanns bara till Basic och Turbopascal. Anledningen till att vi ville undvika just dessa språk var att vi ville ha möjligheter till realtidsprogrammering.

Vi ägnade under några veckor mycket tid och kraft åt att försöka översätta Comliprotokollet från Turbopascal till Modula-2, men problemet med att översätta fasta adresser, Comli kommunicerar endast via coml porten på PC:n, gjorde tidsperspektiven osäkra. Vi gav upp och sökte andra vägar.

Här kommer Omniview in i bilden. Mellan de olika programmen som kördes under Omniview gick det att skicka meddelanden. Traversprogrammet skulle då gå att skriva i Modula-2 och genom att skicka en instruktion till ett annat program skrivit i Turbopascal skulle man kunna kommunicera med SattCon. Dessvärre visade sig Omniview vara beroende av att man körde ett set-up program varje gång man startade datorn och att man inifrån set-up programet startade sitt realtidsprogram. Problem uppstod även med kommunikationen när man skulle börja prioritera de olika processerna.

Problemen med Omniview gjorde att vi valde att skriva traversprogrammet i Turbopascal och därmed kunde man använda sig av Comlirutinerna direkt i traversprogrammet.

Vi hade nu bestämt oss för systemlösningens uppbyggnad och nedan följer en kort sammanfattning med motivering till respektive val.

Traversprogrammet var totalt sett mycket omfattande och vi var osäkra på om hela programmet skulle få plats i SattCon 15-20:s RAM-minne (det slutgiltiga programmet upptog 200 kB i källkodsform). Dessutom hade vi här ett system som helt saknade problem med säkerhetsfrågor och absolut tillförlitlighet. Vidare är inmatningen via traversen inget snabbt system m.a.p styrning. Här fanns alltså möjlighet att skriva programmet i ett högnivåspråk och därmed avsevärt förbättra möjligheterna att strukturera programmet så att det blev lättöverblickat och lätt att ändra i. Att ha traversprogrammet helt separat medförde att man kunde göra ändringar medan pannan var i drift. Detta kom att visa sig vara mycket tidssparande. Idén var alltså att placera traversprogrammet i en separat PC som kommunicerade med SattCon 15-20 via Comli.

Pannprogrammet var den process som ställde störst krav vad gällde driftsäkerhet och tillförlitlighet. Dessutom är det den process som de båda andra programmen är beroende av. Det blev naturligt att här utnyttja SattCon 15-20:s möjlighet till att exekvera egna program skrivna i PBS-språket.

Operatörsgränssnittet kom automatiskt att ligga i en PC. Det är detta program som presenterar all intressant processdata på skärmen. SattControl har själva tagit fram en utvecklingsmiljö för programmering av operatörsgränssnitt, SattGraph1000. Detta var ett mycket dyrt alternativ och dessutom inte helt musbaserat, vilket var

något vi eftersträvade. Att programmera grafik är mycket tidskrävande, varför man är hänvisad till att använda sig av någon form av utvecklingsmiljö där man kan rita bilder. Det utvecklingshjäpmedel vi valde kallas MatrixLayout och är ett Amerikanskt CASE program (Computer Aided System Engineering).

3.3 DET SLUTGILTIGA SYSTEMET

Följande tre grundkomponenter utgör hörnstenarna i det uppbyggda systemet.

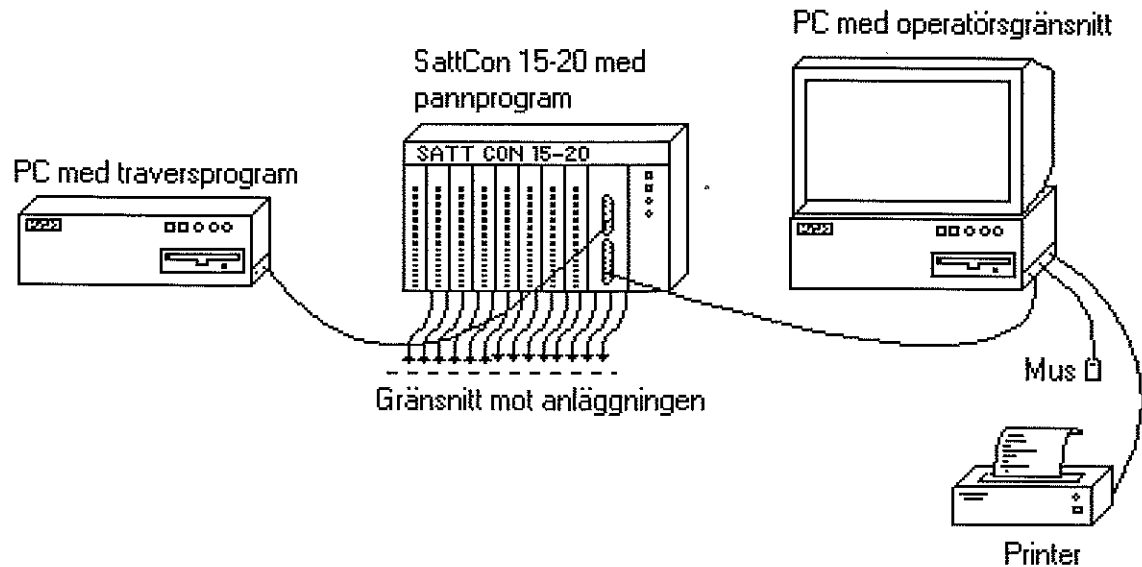


Fig.7 Styr- och reglersystemets uppbyggnad.

PC med traversprogram: Denna dator är en vanlig dos-maskin. Den är utrustad med 386-SX processor med 1 MB RAM-minne, hårddisk samt 3.5 tums floppydisk. Den saknar egen bildskärm. Traversprogrammet är programmerat i Turbo-pascal (ver 5.5). Kommunikationen sker via datorns seriella port COM1.

PC med operatörsgränssnitt: Denna dator sköter om allt utbyte av information mellan människa-maskin. Kommunikationen är helt bildskärmsbaserad och helt musstyrd. På skärmen visas bilder och musstyrda menyer. Tangentbordet behövs aldrig av operatören. Datorn är en vanlig DOS-maskin utrustad med VGA grafik, hårddisk, 3.5 tums floppy, 386-SX processor samt 1MB RAM-minne. Programspråket är Turbopascal och kommunikationen mellan operatördator och SattCon 15-20 sker med kommunikationsprotokollet Comli över port COM1. Det som skickas fram och tillbaka mellan de båda datorerna är t.ex. börvärde, mätvärde, processstatus av skilda slag.

SattCon 15-20 med pannprogram: Denna processdator är s.k raketbaserad och specialgjord för styrning och reglering av processer. I denna dator ligger all styrning av pannan och en liten del av styrningen till traversen. Program-språket är SattControl:s egna PBS-språk som är special-gjort så att sekvenser från relästyrningsschema blir lätta att implementera mjukvarumässigt. Man kan välja vilken typ av gränssnitt man vill ha ut mot processen genom att välja olika kort t ex. digitala in ,digitala ut, 240 VAC eller 24 DC, analoga in eller ut, 0-10 VDC eller 4-20 mA osv. I denna dator ligger en monitor databas, dvs för att från operatören påverka travers-programmet så skall man ändra processdata i databasen. Pannprogrammet och traversprogrammet är helt underställda de data som ligger i databasen som består av register och flaggor.

3.4 SATTCON 15-20

SattCon 15-20 är ett komplett styrsystem avsett för styrning av maskiner och mindre processer. Systemet kan styra både självständigt och i kombination med andra styrsystem. SattCon15-20 hanterar både digitala och analoga signaler. Styrprogrammen (PBS-program) skriver man lämpligen in med programmet DOX5 som medger en slags klartextprogrammering. Med Dox5 kan man skriva in hjälp-kommentarer, använda programbibliotek samt använda macro-insruktioner.

Grundstommen i ett PBS-program består ofta av Booleska ekvationer som utgörs av instruktioner typ OCH, ELLER etc, men innehåller även instruktioner för in- och utläsning av analoga värde, beräkningar, hopp, subrutiner m m. Pannprogrammet som vi senare kommer in på är skrivet i ett s.k PBS-språk.

Teknisk data SattCon 15-20

RAM-minne: 20 Kb

Antal register: 512 st

Antal In/Ut signaler: 1024 st (max)

Antal analoga: 64 st (max)

Konstruktion: Raketbaserat med interfacekort

Tid för en sekvenscykel: < 200 ms

Antal portar: 2 seriella Comli, 1 vdu (för programmering), 1 printerport

3.5 KOMMUNIKATIONEN MELLAN ENHETERNA

Kommunikationen mellan TraversPC:n, SattCon 15-20 och OperatörsPC:n sker seriellt via ett kommunikationsprotokoll kallat COMLI. Comli är företaget SattControl:s egna kommunikationsprotokoll utvecklat för de egna produkterna.

Man kan köpa färdiga procedurer för kommunikation med DOS-PC. Procedurerna finns bl.a i Turbopascal som är programspråket som vi använder. SattCon 15-20 arbetar i vårt fall endast som slave, dvs det är bara på order från PC:n som överföring till/från enheterna sker. PC-maskinerna fungerar alltså i vårt fall som master.

Vi har använt oss av följande procedurer.

- **Initl** som initierar Comlikommunikationen.
- **Init** som sätter överföringshastigheter o.s.v
- **GetReg** som hämtar ett värde från ett visst register i SattCon 15
- **SetReg** som laddar ner ett värde i ett register i SattCon 15
- **GetBit** som hämtar önskad statusbit i SattCon 15
- **SetBit** som sätter en önskad statusbit till ett eller noll i SattCon 15

Inställningar för kommunikationen har vi satt till:

Hastighet = 9600 Baud
Databitar = 8st
Stoppbitar = 1st
Paritet = udda.

För djupare förståelse hänvisar vi till manualen Comli-Pascal.

3.6 DEN FYSISKA INKOPPLINGEN AV SYSTEMET

Det gamla styrsystemet skall arbeta parallellt med det nya. Med detta menas att man när som helst skall kunna koppla ut det nya, och in det gamla. På detta sätt fås en betryggande redundans för styrsystemet i anläggningen. In-/urkoppling av systemet sker med en strömställare på dörren till PLC-skåpet.

Två typer av inkopplingar krävs när man skall ansluta ett nytt system och samtidigt ta bort ett gammalt:

1 Inkoppling av insignaler

2 Inkoppling av utsignaler

Insignaler: Insignaler är lätta att ansluta. Det enda man behöver göra är att tappa av signalen.

Utsignaler: Utsignaler har vi valt att koppla ur med reläer.

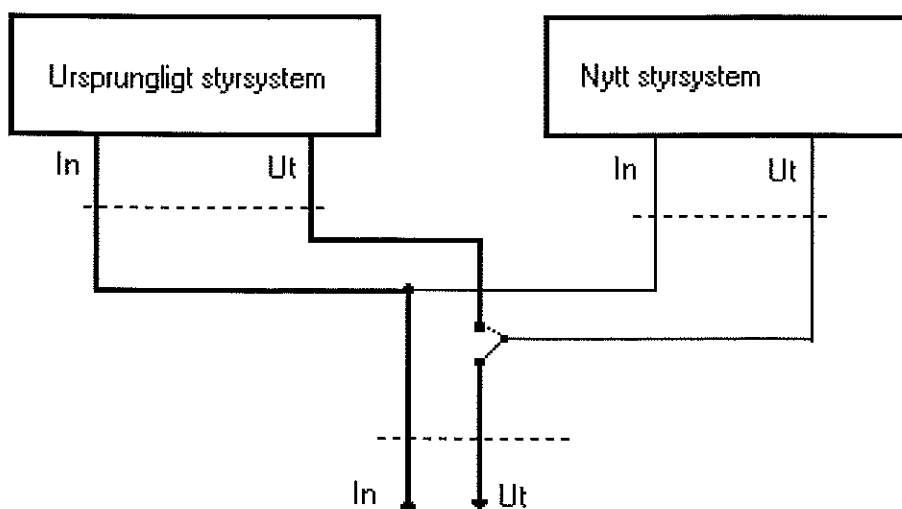


Fig. 7b

BESKRIVNING AV BRÄNSLEHANTERINGEN

4.1 BRÄNSLEHANTERINGEN

Till skillnad från oljepannor som eldas med fastbränsle är det bara i undantagsfall fastbränslepannor eldas med bränsle av jämn kvalitet. Pannan i vårt fall eldas under tillnärmelsevis extrema förhållanden då fukthalten i bränslet kan variera från 4 till 65 %. Medan fastbränslet 'putsdamm' närmast exploderar i pannan så sker motsatsen när man eldar med blöt bark. Skälen till att vilja elda med bränsle av så här ojämn kvalitet är främst ekonomiska; sämre bränsle = lägre pris. Utöver denna vinst är en ytterligare anledning att leverantörer av t.ex. bark ej kan garantera en viss fukthalt då avbarkning och lagring sker utomhus och är helt väderleksberoende.

Det ursprungliga traversprogrammet saknade någon egentlig bränslehantering, med undantag av omlastningsfunktionen. Vår idé var att skriva ett bränslehanteringsprogram snarare än ett rent traversinmatningsprogram. Att använda traversen inte bara till transport av bränslet från lagerfickorna till bränslebehållaren utan till en mer omfattande hantering av bränslet.

Av skäl som nämnts tidigare valde vi att skriva traversprogrammet i Borlands TurboPascal version 5.5. Programmet är skrivet på engelska då Pascal kompilatorn ej klarar svenska tecken, och dessutom är programspråkets grammatik (syntaxen) på engelska.

Indelningen och namnuppdelningen av fickorna i C, B, A, D är helt enligt ursprungliga traversprogrammet, liksom namnet på styrsignalerna.

D-fickan är inte inlagd i operatörsgränssnittet. Anledningen till detta är att den används som reservficka.

Lagerfickornas indelning i koordinater

Grunden till hela traversprogrammet är lagerfickornas indelning i koordinater. Den indelningen går vi först igenom innan vi börjar titta på de olika delarna i traversprogrammet.

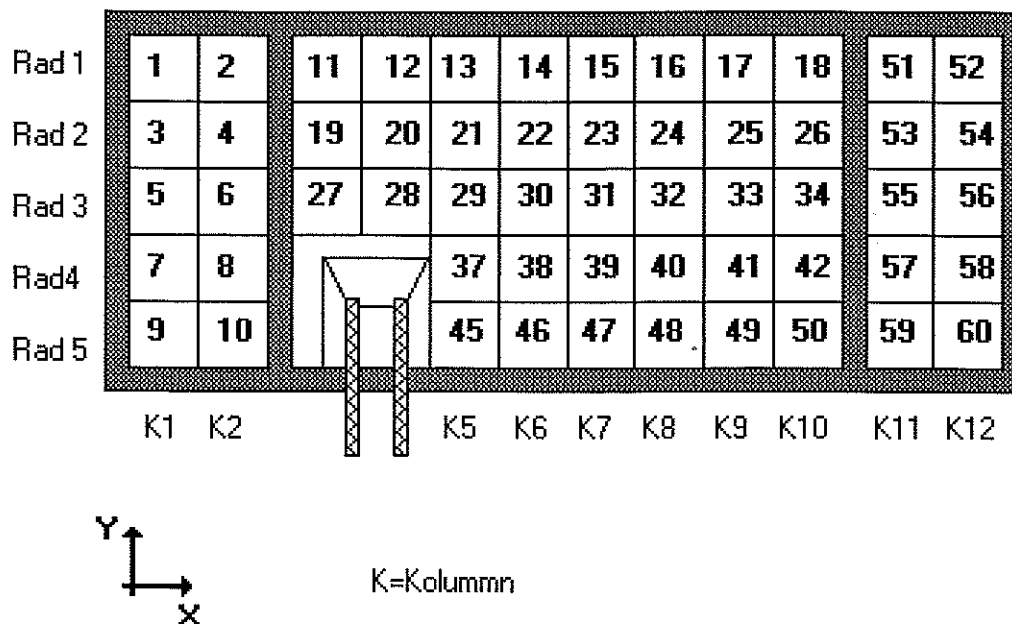


Fig.8 Lagerfickor med koordinatindelning.
OBS. Skilj på rad och y-läge.

Varje ficka har en löpande numrering för att underlätta programmeringen.

Ficka C : 1 till 10
Ficka B : 11 till 50
Ficka A : 51 till 60
Ficka D : 61 till 70

Totalt finns alltså 70 koordinater.

Till varje koordinat hör ett X- och ett Y läge . Rad 5 och kolumn 1 svarar mot koordinaten 9. Detta är traversens initieringspunkt och härifrån räknas X och Y lägena.

Koordinat 9 : X läge 1
 Y läge 1
Koordinat 2 : X läge 2
 Y läge 5
Koordinat 31 : X läge 7
 Y läge 3

I början av programmet tilldelas varje koordinat sina respektive X och Y värden. Detta görs i en vektorpost square. Nedan räknas vektorpostens poster upp.

```
feed   : ARRAY[1..70] OF STRING[8]
reload : ARRAY[1..70] OF STRING[8]
X      : ARRAY[1..70] OF INTEGER
Y      : ARRAY[1..70] OF INTEGER
```

Ex. : square.feed[48]='textsträng'

Textsträngen är valfri men får högst innehålla åtta tecken. Denna kan sedan användas i programmet för att kontrollera koordinatstatus vid t.ex. inmatning eller omlastning.

Ett exempel på den information man kan få om koordinat 6.

```
square.feed[6]='dig'
square.X[6]=2
square.Y[6]=3
square.reload[6]='blocked'
```

Genom att referera till vektorn får vi alltså informationen:

- * koordinaten 6 är aktiverad för inmatning
- * X läget är 2
- * Y läget är 3
- * koordinat 6 går ej att omlasta

De textsträngar vi valt att använda oss av är:

```
'dig'      : aktiverad för inmatning eller
            omlastning
'stop'     : blockerad för inmatning och får
            får ej deaktiveras av
            programmet själv.
'empty'    : koordinaten är grävd tom
'dont-exist' : finns ej med i programmet
            (betong fundament till
            bränslebehållaren koord.
            35,36,43,44)
```

Val av inmatningskoordinater

I det gamla traversstyrprogrammet fanns det ingen möjlighet att välja från vilken eller vilka specifika koordinater inom en ficka från vilka man ville att inmatning skulle kunna ske. Inmatningen skedde alltid radvis med början utifrån portarna. Först när en rad i en ficka var tom, började traversen gräva i nästa rad och så vidare, in mot väggen.

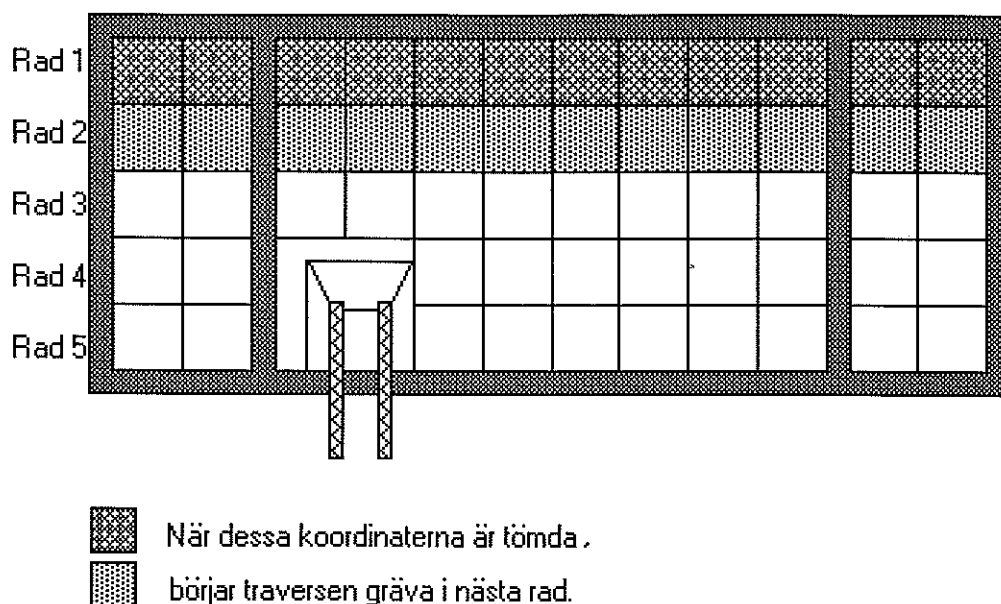


Fig.9 Lagerfickor med markerade inmatningsrader.

Bränslet inom en ficka kan även det variera trots det är av samma typ. Fickorna C och A rymmer vardera 12 lastbilscontainrar och B över 28.

En förbättring vore alltså att även inom fickorna ha möjlighet att välja från vilka enskilda koordinater man vill att grävning skall utföras.

Lösningen blev att utifrån operatörsgränssnittet ha möjlighet att markera önskade inmatningskoordinater. Först när dessa var tomma började inmatning ske enligt den gamla principen med radvis inmatning med start från portarna.

En ytterligare ny finess är möjligheten att önskade koordinater helt kan blockeras för inmatning. Detta har betydelse först när markerade inmatningskoordinater har grävts tomma. Oavsett om hela fickan är tömd, kommer traversen ej att gräva i de blockerade koordinaterna.

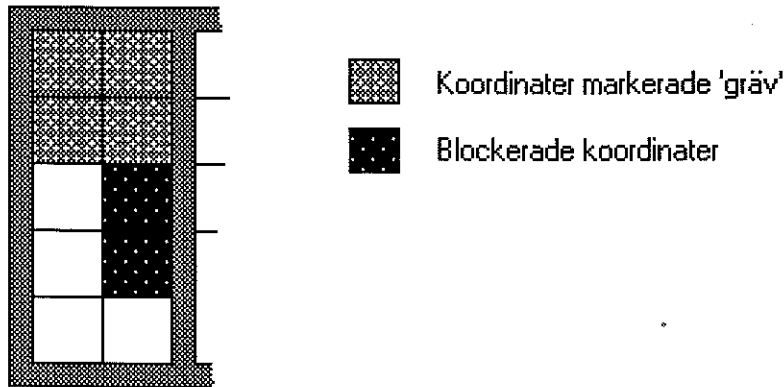


Fig.10 Lagerficka med markerade koordinater.

EX. : I två av koordinaterna har upptäckts ovanligt mycket större pinnar. Man vill undvika att traversen matar in från dessa under natten då anläggningen är obemannad. Detta p.g.a risken för stopp i bränsleinmatningsskruvarna. Genom att blockera dessa rutor riskerar man ej att traversen gräver där trots att markerade inmatningskoordinater grävts tomma.

Inmatningsföljd

Som nämnts tidigare hade det gamla traversprogrammet bara tre olika inmatningsalternativ; en skopa från A och en B, en skopa B och en C eller inmatning med en skopa från någondera av fickorna.

En förbättring vore alltså att öka dessa fasta inmatningsföljder. Detta kunde naturligtvis lösas på ett antal olika sätt och efter en del funderingar på olika alternativ beslöt vi att använda oss av något man skulle kunna kalla 'recept-metoden'. Förutsättningen för denna är att man i de olika fickorna har sorterat bränslet efter fukthalt. Vi låter helt enkelt operatören ha möjlighet att bestämma den blandning han önskar. Ingående 'ingredienser' är fickorna A, B och C. Övre gräns är att receptet inte får innehålla mer än 10 på varandra följande 'ingredienser'.

Ex på tre recept : 1. ACGB
2. BAAA
3. C

Enligt recept ett börjar inmatningen med en skopa från A. Därefter lägger sig gripskopan i ficka C och väntar på nästa påfyllningssignal. Då denna fås tas en skopa från C därefter vid nästa påfyllningssignal en till från C och likaledes sker med B. När serien är genomlöst börjar man om från A igen.

Enligt recept två blir bränsleblandningsförhållandet tre skopor A på en skopa B. Enligt recept tre sker endast inmatning från C.

Matterhorn

Man har nu uppnått ett flexiblare system för bränsleinmatningen, men ett stort problem kvarstår. Grip-skopan har en fast kapacitet på drygt 2 m³ vilket gör att växlingarna mellan inmatat bränsle kan bli stora vilket i sin tur försvårar regleringen av pannan. En enkel men mindre bra lösning på detta är att använda en mycket mindre gripskopa. En betydligt bättre lösning är den som värmeverkschefen Tommy Göransson kom med en morgon:
-Släpp allt bränsle i en fast punkt utanför bränslebehållaren och gräv sedan runt denna.

En mycket enkel lösning på ett mindre enkelt problem. En lösning som kom att visa sig fungera alldeles utmärkt och som kom att kallas Matterhorn.

Principen är enligt följande:

Man följer det recept som man valt men istället för direkt inmatning till bränslebehållaren, går man via en fast koordinat. All inmatning till bränslebehållaren sker sedan runt denna fasta koordinat.

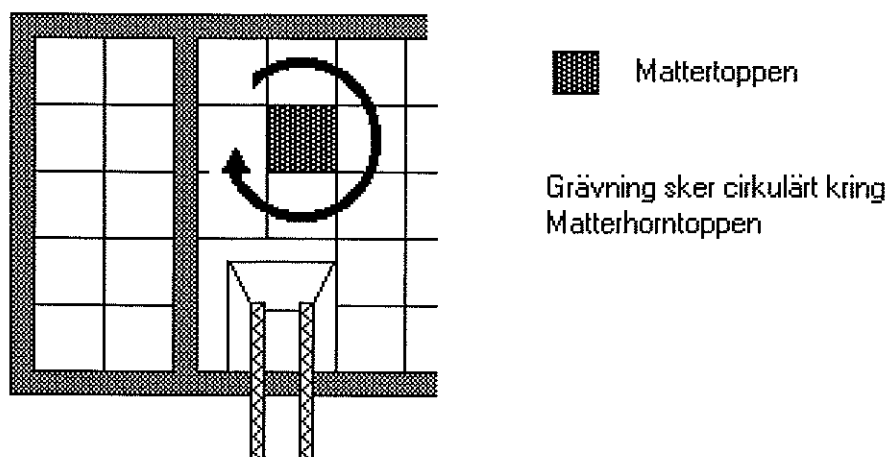


Fig.11 Översiktsbild med Matterhorns placering i ficka B.

Den fasta koordinaten valde vi att lägga i B's ficka, andra koordinaten rakt ut räknat från bränslebehållarens koordinat.

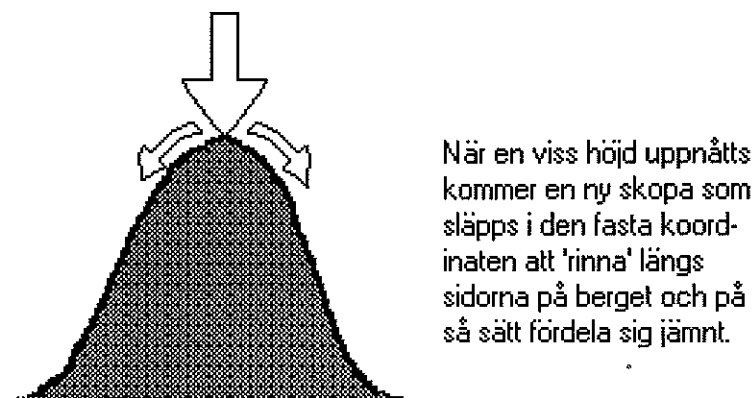


Fig.12 Blandningsprincipen i Matterhorn.

Vid påfyllningssignal hämtar traversen en skopa i aktuell koordinat bredvid den fasta koordinaten och levererar denna till bränslebehållaren. Enligt receptet hämtas därefter en ny skopa, som släpps i den fasta Matterhornkoordinaten. Gripskopen lägger sig sedan i väntläge bredvid koordinaten från vilken inmatning skett i väntan på ny påfyllningssignal.

Som alternativ inmatning finns fortfarande möjligheten med direkt inmatning till bränslebehållaren.

Bränslereglering

Det enda kvarvarande problemet var nu att man körde med mycket bättre bränsleblandning på dagarna, då effekten var högre än på nätterna. En viss tröghet finns i Matterhornprincipen, då det behövs ett antal skopor innan ett nytt recept verkligen slår igenom. Har man hela natten kört på en sämre blandning finns denna naturligtvis kvar som Matterhorns kärna när receptet på morgonen behöver förbättras.

Vi valde att göra två alternativa lösningar.

Den första lösningen på detta var möjligheten att kunna öka Matterhorntoppen, dvs att kunna fylla på i den fasta koordinaten med ett önskat antal skopor utan att gräva omkring toppen.

En andra metod var en form av bränslereglering. Genom att kontrollera vissa pannparametrar kan man se om pannan behöver bättre bränsle för att klara av att hålla inställd framledningstemperatur.

Vi valde variablerna:

Framledningstemperatur

Rökgastemperatur

Kolmonoxid (CO)

Under en viss framledningstemperatur eller under en viss rökgastemperatur eller om CO har legat över ett inställbart värde i mer än fem minuter (vilket tyder på hög fukthalt hos bränslet) sker reglering till bättre bränsle.

Reglering till bättre bränsle sker genom att man matar in en skopa från ficka D direkt till bränslebehållaren. Vid nästa påfyllningssignal görs detta åter om något av reglervillkoren är uppfyllt.

Ficka D finns ej med som alternativ vid val av bränsleinmatningskoordinater eller vid val av recept. I D förvaras alltid bränsle av bästa kvalitet (Ex. Kutterspån).

Omlastning

När nya leveranser av bränsle anländer kan dessa bara tippas i de yttre koordinaterna vid respektive port. Det finns därför naturligtvis ett behov av att kunna omlasta dessa koordinater inåt i fickan.

Vad gällde omlastning valde vi att behålla principen från det gamla programmet, undantaget att omlastningen avbryts vid behov av bränslepåfyllning och därefter fortsätter.

Omlastning sker från fickans två yttre rader, till fickans två inre rader. Detta gäller med undantag av de sex koordinaterna mitt emot bränslebehållaren vilka ej omlastas.

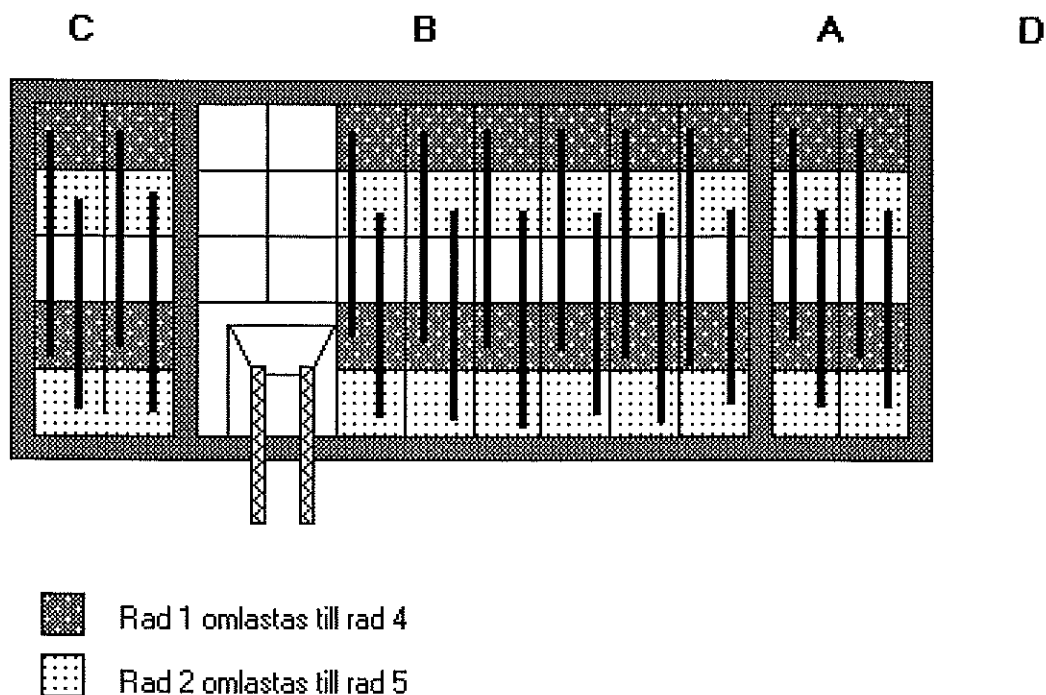


Fig.13 Översiktsbild på omlastningskoordinater.

Det är även möjligt att välja omlastning från D till A. Varje koordinat i D omlastas då till motsvarande i A.

4.2 TRAVERSPROGRAMMETS SIGNALER

Traversprogrammet är uppbyggt av procedurer. Varje procedur kan betraktas som ett byggblock. Flera byggblock kan sättas samman till en ny procedur, vilken i sin tur kan användas som enskilt byggblock. När man använder sig av en färdig procedur behöver man ej tänka på hur den är uppbyggd. Det enda man behöver känna till är funktionen och vilka in- och utparametrar den använder resp ger upphov till.

Nedan följer en genomgång för varje procedur av in- och utparametrar samt funktionen.

Följande signaler används för styrningen av traversen:

Insignaler

| | | |
|--------------------|---|---|
| Låg bränslenivå | : | Signal från en ultraljudsgivare för påfyllning av bränslebehållaren |
| Traversutgångsläge | : | Ändläge för traversen |
| Traversvänster | : | Traversens bakre del passerar givaren |
| Stoppgivaretravers | : | Traversens mittersta del passerar givaren |
| Travershöger | : | Traversens främre del passerar givaren |
| Trallautgångsläge | : | Ändläge för trallan |
| Trallavänster | : | Trallans bakre del passerar givaren |
| Stoppgivaretralla | : | Trallans mittersta del passerar givaren |
| Trallahöger | : | Trallans främre del passerar givaren |
| Skopaövre1 | : | Övre nivå för skopan vid omlastning |
| Skopaövre2 | : | Övre nivå för skopan vid bränsleinmatningen |
| Skopanedre1 | : | Nedre nivå vid omlastning |
| Skopanedre2 | : | Nedre nivå vid bränsleinmatning |
| Nollastskopa | : | Skopan vid bränslenivå |

Utsignaler

| | | |
|----------------|---|-----------------------------------|
| Traversminus | : | Travers mot ändläge (utgångsläge) |
| Traversplus | : | Travers från ändläge (3 m/min) |
| Travershögfart | : | Traversmotor 10 m/min |
| Trallaminus | : | Tralla mot ändläge (3 m/min) |
| Trallaplus | : | Travers från ändläget (3 m/min) |
| Trallahögfart | : | Trallamotor 10 m/min |
| Lyftupp | : | Hissa upp skopan (låg fart) |
| Sänkned | : | Hissa ner skopan (låg fart) |
| Sänksnabbt | : | Motor högfart |
| Lyftsnabbt | : | Motor högfart |
| Öppnaskopan | : | Öppnar gripskopan |
| Stängskopan | : | Stänger gripskopan |

Alla benämningar enligt det ursprungliga systemet. Vi har valt att behålla dessa i nya programmet.

4.3 TRAVERSPROGRAMMET OCH DESS PROCEDURER

Procedure: InitMain

Vid anrop av denna procedur tilldelas koordinaterna i lagerfickorna ingångsvärden vid systemstart.

Funktion: För varje koordinat i inmatningsfickorna finns ett X och ett Y värde. T ex motsvaras koordinat 1 av X=1 och Y=5. Dessa värden tilldelas i InitMain vid systemstart. Vidare tilldelas varje koordinat en ingående status. De som är fria att användas vid inmatning eller omlastning tilldelas textsträngen 'free'. Koordinater som ej existerar i matrisen såsom betongfundament tilldelas 'dontexist'. Dessa värden kan sedan påverkas från operatörsgränssnittet.

Procedure: GetBit With 30sFailureTest

Vid anrop av denna procedur fås felkontroll vid förflyttning av travers eller tralla.

Funktion : När man väntar på att en statusbit skall ettställas (t.ex signal från induktiv givare) fås error=True om detta ej skett inom 30 s. från det att man började läsa på signalen. I annat fall detekteras signalen.

Procedure: Check Level Guard Tank

Vid anrop tilldelar proceduren flaggan FEED värdet True om bränslepåfyllning skall ske, om inte, värdet False.

Funktion: Via en ultraljudsgivare detekteras huruvida bränslepåfyllning skall ske. Givaren läser över ett område på ca 1 m². P.g.a att bränslehöjden varierar slår signalen av och på innan nivån är så låg så att en konstant signal erhålles. I SattCon finns ett program som sätter en flagga då signalen varit 'hög' i 5 sekunder. Detta program måste p.g.a realtidskrav ligga i SattCon.

Procedure: Check Level Guard Store

Vid anrop av proceduren tilldelas vektorposten square.feed[x] textsträngen 'empty' vid tom koordinat eller om omlastning, square.reload[x]:='empty'.

Funktion: Via en induktiv givare som är placerad på trallans kabeltrumma kan man avläsa hur mycket wire som är utrullad. Det finns två inställbara nivåer. Dessa används vid omlastning resp bränsleinmatning.

Procedure: Check Store Empty

Vid anrop av proceduren tilldelas flaggan storeempty värdet True om hela fickan är tom, annars värdet False.

Funktion: Om alla koordinater i en ficka är markerade tomma tilldelas vektorposten fuelstore, med notationen aktuell post, värdet True. Detta sker i den procedur från vilken inmatningen sker. Denna procedur kan sägas översätta aktuell post (A, B eller C) till den generella flaggan storeempty. Översättningen sker genom att man vet vilken ficka som behandlas (exekveras) för tillfället.

Procedure: Open Scoop

Vid anrop av denna procedur öppnas gripskopan om den är stängd.

Funktion: Det finns ingen givare som markerar öppet resp stängt läge för skopan. Öppet läge fås genom att en viss tid lägga ut signalen öppnaskopa. Vid systemstart öppnas alltid skopan, därefter sätts en flagga i SattCon till 1. Man kan sedan i andra procedurer använda sig av denna som en passiv signal för att få veta vilket läge, öppen/stängd, som skopan intar.

Procedure: Close Scoop

Vid anrop av denna procedur stängs gripskopan om den är öppen.

Funktion: Enligt Open_Scoop, med undantaget att flaggan sätts till 0.

Procedure: Wire Down

Vid anrop av denna procedur sänks gripskopan tills nollast erhålles.

Funktion: Gripskopan sänks ner tills givarsignalen nollastskopa erhålles. En flagga, restingMatterhorn, används för att veta om skopan befinner sig i någon av Matterhornkoordinaterna. Om så är fallet hissas ytterligare något lite wire ut för att få skopan att ligga plant mot Matterhorn väggen.

Procedure: Wire Up

Vid anrop av denna procedur hissas gripskopian till sitt övre läge och flaggan resting tilldelas värdet False.

Funktion: Gripskopian hissas upp till sitt övre läge varefter variabeln resting tilldelas värdet False för att markera att traversen 'arbetar'. I denna procedur ligger också stängning av gripskopian vid bränsleinmatning eller omlastning. Detta då skopian först orkar stänga då den hissats upp en bit.

Procedure: Start Traverse

Vid anrop av denna procedur startar traversen i den riktning som svarar mot variabeln directionX.

Funktion: Traversen startar i den riktning som variabeln directionX anger. Variabeln kan anta värdet '+' eller '-' vilket svarar för höger resp vänster i X-led. Start sker i lågfart men om avståndet till nästa stoppkoordinat är större än ett, går traversen upp i högfart. Variabeln range anger detta avstånd.

Procedure: Stop Traverse

Vid anrop av denna procedur stannar traversen över aktuell mittkoordinat.

Funktion: Först stängs höghastighet av och därefter läses givaresignalen för att koordinatens mittpunkt är nådd. När signal erhålles stannas traversen.

Procedure: Start Truck

Vid anrop av denna procedur startar trallan i den riktning som anges av variabeln directionY.

Funktion: Helt i analogi med Start Traverse, med undantag av att variabeln är directionY.

Procedure: Stop Truck

Vid anrop av denna procedur stannar trallan över först nådda mittpunkt. Om variabeln feedtank är True öppnas gripskopan direkt vid anrop.

Funktion: Helt i analogi med Stop Traverse, med undantag av att variabeln är directionY. Proceduren ombesörjer även ett specialfall vid stopp av trallan över bränsleinmatningsfickan. För att inte hela skopans innehåll skall släppas i en punkt i bränslebehållaren låter man trallan vara i rörelse under tömningen. Variabeln feedtank sätts till True då detta önskas.

Procedure: Move Truck

Vid anrop av denna procedur förflyttas trallan till önskat Y läge. Variabeln presentY tilldelas motsvarande koordinatläge.

Funktion: Används vid förflyttning av trallan till ny koordinat. Variabeln i motsvarar den önskade koordinaten och presentY aktuell. Via koordinaten fås önskat Y värde varefter riktning bestäms. Om avståndet är större än en koordinat går trallan upp i högfart. När trallan stannat över önskad koordinat tilldelas presentY det önskade koordinatvärdet.

Procedure: Move Traverse

Vid anrop av denna procedur förflyttas traversen till önskat X läge. Variabeln presentX tilldelas detta läge.

Funktion: Principen är helt i analogi med Move Truck, dock är lösningen annorlunda. P.g.a problem med att hinna läsa koordinaterna vid förflyttning av traversen sköts detta av SattCon.

Procedure: Fill Scoop

Vid anrop av denna procedur sänks skopan till bränslenivån varefter den stängs.

Funktion: Först kontrolleras om skopan är i sitt nedre läge, om inte så hissas den ner. Därefter stängs skopan.

Procedure: Reload Scoop

Vid anrop av denna procedur omlastas en skopa från koordinat k till dennas fasta omlastningspunkt.

Funktion: Varje omlastningsbar koordinat svarar mot en fast koordinat dit omlastningen sker. Genom att ange den koordinat k, vilken man vill omlasta så beräknas omlastningskoordinaten och omlastning sker.

Procedure: Feed Tank

Vid anrop av denna procedur sker påfyllning av bränslebehållaren.

Funktion: Denna procedur förflyttar tralla och travers från inmatningskoordinat till bränslebehållaren där tömning av skopan sker. En kontroll av att traversen befinner sig över bränslebehållaren görs genom att jämföra SattCons registrerade X värde med presentX. Om så ej är fallet omlastnings traversen.

Procedure: Feed Matterhorn

Vid anrop av denna procedur förflyttas trallan och traversen från aktuell koordinat till Matterhorntoppen där skopan töms.

Funktion: Använder sig bara av tidigare procedurer för att förflytta trallan och traversen till Mattertoppen där skopan töms.

Procedure: Check Reloading

Vid anrop av denna procedur kontrolleras om någon ficka skall omlastas, och om så är fallet vilken. Variabeln reload tilldelas True vid omlastning och vektorn reloadstore innehåller aktuell omlastningsficka.

Funktion: Då omlastning skall ske av en ficka så ettställs en flagga i SattCon och ett register används för att ange aktuell omlastningsficka. Proceduren läser dessa och

översätter SattCons siffervärden till en flagga reload och en vektor reloadstore. Reload är True om omlastning skall ske och reloadstore tilldelas 'A', 'B' eller 'C'.

Procedure: Start Reloading

Vid anrop av denna procedur omlastas den ficka som anges av vektorn square.reload. Omlastningen sker till fickans två inre rader.

Funktion: I varje ficka omlastas de två yttre raderna till de två innersta raderna. Första gången en ficka omlastas eller om man byter omlastningsficka tilldelas variabeln firsttimereloading värdet True. Genom vektorn reloadstore sker sedan en initiering av aktuell ficka. Variabeln oldnumber motsvarar sist omlastade koordinaten. Omlastning sker tills variabeln reload sätts till False via operatörsgränssnittet eller variabeln passed blir True. Det senare sker om alla omlastningskoordinater indikerat att de är tomma.

Procedure: Start Reloading Dstore

Vid anrop av denna procedur omlastas D fickan till A fickan

Funktion: D fickan ingår inte i operatörsgränssnittet utan påverkas genom en yttre strömställare. Varje koordinat i D fickan omlastas till motsvarande koordinat i A fickan. Variabeln oldDreload används för att 'komma ihåg' vilken koordinat som sist omlastades. Omlastningen kan endast avbrytas genom att slå ifrån strömställaren.

Procedure: Get Coord Status

Vid anrop av denna procedur läses koordinatstatus från SattCon

Funktion: Via operatörsgränssnittet kan varje koordinat påverkas. Varje koordinat tilldelas något av de tre statusvärdena; gräv, ledig eller blockerad. Ledig svarar mot att koordinaten går att gräva ifrån under förutsättning att de koordinater i fickan som markerats med gräv, är tomma. Blockerad svarar mot att koordinaten aldrig kan tilldelas statusen gräv. Operatörsinställningarna läses ner och

lagras i SattCon, och läses därifrån av Traversprogrammet. Vektorposten fuelstore tilldelas False då hela fickan är tömd, i denna procedur tilldelas den False vid uppdatering.

Procedure: Get Feeding Stores

Vid anrop av denna procedur så hämtas från SattCon den turordning i vilken man önskar gräva i fickorna. Denna lagras i vektorn feedlist.

Funktion: Via operatörsgränssnittet anges önskad inmatningsföljd. Denna följd lagras i SattCon där sedan traversprogrammet kan hämta den. Inmatningsfickorna lagras i en vektor, feedlist. Ex. feedlist[1]='B', feedlist[2]='A', feedlist[3]=' '. Sista elementet i vektorn är alltid ett blanktecken för att markera slut. Högst 11 element kan ingå i inmatningsföljd. En global variabel number används för att ange var i listan man befinner sig. Denna sätts till ett varje gång en ny inmatningsföljd lästs ner.

Procedure: Select ABC

Vid anrop av denna procedur så tilldelas variabeln exe den ficka som står på tur för inmatning.

Funktion: Den globala variabeln exe tilldelas aktuell inmatningsficka genom att tilldelas värdet av feedlist. Om detta är ett blanktecken så tilldelas number 1, dvs man börjar om från början på inmatningsföljden. För aktuellt exe kontrolleras att flaggan storeempty är False dvs att inte hela fickan är tömd. Om så är fallet hoppas denna över, under förutsättning att fler fickor ingår i följd.

Procedure: Goto Next Feeding Coordinate

Vid anrop av denna procedur så förflyttas tralla och travers till den koordinat som står på tur vid nästa inmatningstillfälle.

Funktion: Om inga fler uppgifter skall utföras används denna procedur för att förflytta trallan och traversen till den koordinat som står på tur vid nästa inmatningstillfälle. Proceduren använder sig av oldA, oldB, oldC resp oldM, där sistnämnda motsvarar den koordinat i Matterhorn från vilken senast

inmatades. De övriga på samma sätt för resp
ficka.
Om inmatningsföljden är tom (feedlist[1]=' ')
så används koordinat 15 som viloläge.

Procedure: Start Feeding From A

Vid anrop av denna procedur sker inmatning av en skopa
till bränslebehållaren direkt från ficka A, eller
omlastning till Matteredtoppen om inmatning sker via
Matteredhorn

Funktion: Om det är första gången ficka A används
tilldelas aktuell inmatningskoordinat värdet
51, vilket svarar mot första och yttersta
koordinaten i ficka A. I annat fall tilldelas
den oldA:s värde. Om vektorposten
square.feed='dig' sker inmatning från denna
koordinat. Om så ej är fallet undersöks
koordinat 52 o.s.v tills villkoret uppfylls
eller hela fickan markerats tom ;
fuelstore.A=False. Hela fickan är t.o.m.
först då alla koordinater markerats tomma,
eller om alla är blockerade. Om de
koordinater som markerats för inmatning är
tomma markeras nästa rad i fickan som har
minst en ledig koordinat. När en koordinat
räknats fram sker inmatning direkt till
bränslebehållaren utom då Matteredhornflaggan
är True, varvid omlastning av skopan sker
till Matteredtoppen.

Procedure: Start Feeding From B

Vid anrop av denna procedur sker inmatning av en skopa
till bränslebehållaren direkt från ficka B, eller
omlastning till Matteredtoppen om inmatning sker via
Matteredhorn.

Funktion: Helt i analogi med Start Feeding From A

Procedure: Start Feeding From C

Vid anrop av denna procedur sker inmatning av en skopa
till bränslebehållaren direkt från ficka C, eller
omlastning till Matteredtoppen om inmatning sker via
Matteredhorn.

Funktion: Helt i analogi med Start Feeding From A

Procedure: Start Feeding From Matterhorn

Vid anrop av denna procedur sker inmatning från Matterhorn.

Funktion: Först initieras de koordinater i B fickan som skall användas av Matterhorn. Dessa är 11-13, 19-21, 27-29 där koordinat 20 motsvarar Matterhorntoppen. Ex. på tilldelning: square.feed[11]:='Mdig'. Om det är första gången inmatning sker via Matterhorn tilldelas oldM 11.

Procedure: Start Feeding From ABC

Vid anrop av denna procedur sker inmatning till bränslebehållaren eller omlastning till Matterhorn.

Funktion: Genom att använda sig av tidigare procedurer bestäms vilken ficka som står på tur för inmatning eller om omlastning skall ske till Mattertoppen.

Procedure: Start Feeding

Vid anrop av denna procedur sker inmatning till bränslebehållaren efter beaktande av alla varianter.

Funktion: Det finns tre alternativ till bränsleinmatning.

1. Direkt inmatning från aktuell ficka
2. Inmatning via Matterhorn
3. Vid reglering till bättre bränsle sker inmatning från ficka A.

Alternativ ett och två enligt tidigare beskrivning. Reglering till bättre bränsle sker om SattCon sätter flaggan fuelcontrol till True (1). Detta sker om vissa förutbestämda parametrar underskrids. Dessa kan påverkas från operatörsgränssnittet.

Procedure: Goto Fix Position

Vid anrop av denna procedur förflyttas traversen till koordinaten utanför bränslebehållaren där skopan blir kvar i upphissat läge.

Funktion: Vid t.ex påfyllning av fickorna vill man att gripskopan skall vara upphissad och befinna sig i en fast koordinat. Denna har valts till koordinaten utanför bränslebehållaren. Proceduren ligger sedan i en slinga och går inte ur denna innan flaggan fixposition sätts till False (0)

Procedure: Refill Matterhorn

Vid anrop av denna procedur ökas Matterhorn med en skopa.

Funktion: Fyller på Mattertoppen med en skopa från aktuell ficka varefter variabeln Mattertop ökas med ett.

Procedure: Check Mattertop

Vid anrop av denna procedur ökas vid behov Matterhorn höjden.

Funktion: Vid uppstart av programmet sätts Mattertop till 20. Varje skopa som tas från Matterhorn minskar detta värde med ett. Denna procedur kontrollerar om Mattertop är 20, och om så inte är fallet fylls en skopa på.

Procedure: To Initial Position

Vid anrop av denna procedur oinitieras travers och tralla.

Funktion: Först nollställs alla utgångar på SattCon som berör traversen. Därefter körs traversen till sitt utgångsläge, som i detta fallet är nedre vänstra hörnet av inmatningsficka C. Där tilldelas presentX och presentY värdet 1. Alla flaggor i SattCon nollställs och firsttimereloading tilldelas True.

Main program

Huvudprogrammet använder sig av ovan beskrivna procedurer för att sköta bränslehanteringen. Första delen av programmet är en initieringsdel där alla använda globala variabler tilldelas ett ursprungsvärde. Denna del utförs bara vid uppstart eller omstart av datorn.

Andra delen utförs i en loop som aldrig kan avslutas av operatören. Denna del är uppbyggd så att de viktigaste procedurerna utförs först. Exempelvis har Goto_Fix_Position högre prioritet än Start_Feeding, vilket innebär att inmatning till bränslebehållaren ej sker trots ultraljudsgivaresignal så länge skopan befinner sig i låst läge.

På samma sätt har Start_Feeding högre prioritet än Start_Reloading dvs inmatning till bränslebehållaren sker före omlastning.

Huvudprogrammet kan sägas vara indelat i cykler, där varje cykel motsvarar en rörelse av skopan från ursprungsläge till tömd skopa eller från tömd skopa till viloläge. När cyklen är genomlöst utförs nästa högst prioriterade uppgift. Lägst prioritet har Goto_Next_Feeding_Coordinate, dvs gå till viloläge i nästa inmatningskoordinat.

Alla förändringar av uppgifter sker från SattCon som i sin tur påverkas av insignaler eller från operatörsgränssnittet.

Om inga nya uppgifter skall utföras och traversen redan befinner sig i viloläge så görs ett uppehåll på åtta sekunder innan nästa läsning sker från SattCon. Detta för att minska belastningen på Comlikanalen som använder sig av samma processor som pannprogrammet.

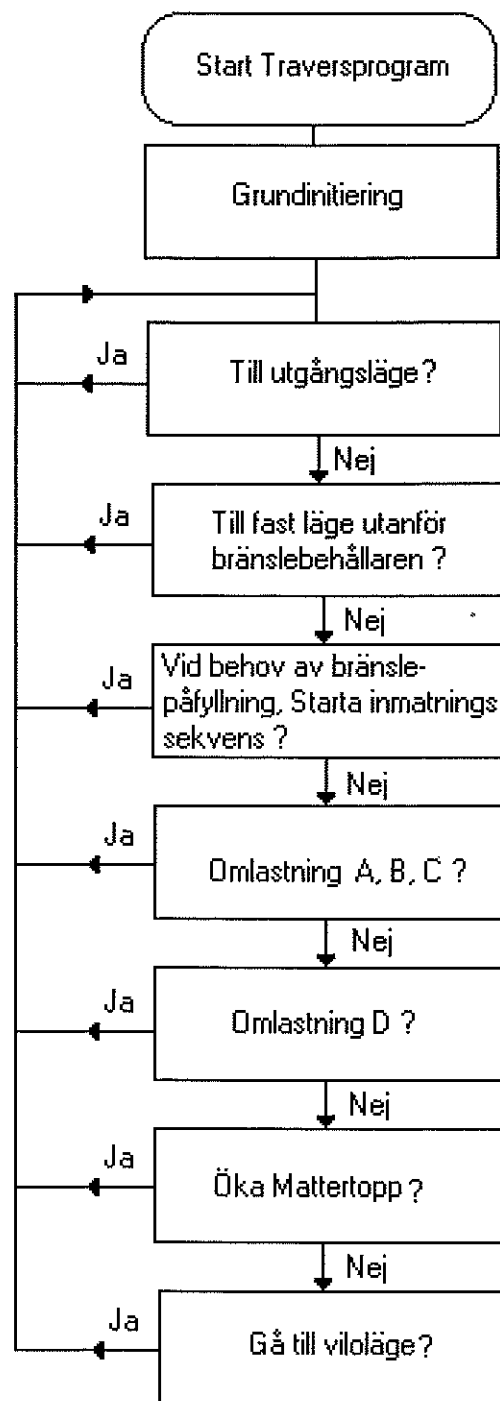


Fig.14 Schema över prioritetsordningen i traversprogrammets huvudloop.

BESKRIVNING AV FÖRBRÄNNINGSPROCESSEN

5.1 FÖRBRÄNNINGSPROCESSEN

I vår uppgift ingick att försöka förbättra förbränningsprocessen som består av fastbränslepannan. En del styrningsprinciper från den gamla regleringen fungerade bra medan andra kunde förbättras. Det man framförallt ville uppnå var att pannan ej skulle överskrida miljökraven samt att pannan skulle gå jämnare (mindre svängigt).

Ett stort problem vid styrningen av pannor i fjärrvärmeverk är att man hela tiden utsätts för varierande belastning. Kunderna sätter på och stänger av ventiler så som de önskar. Exempel på tillfälle när kundernas varierande behov ställer till förtret är tidiga mornar under vinterhalvåret. Klockan 06.00 på morgonen kan effektuttaget på nätet ligga på 2 MW. En timme senare kan effektuttaget ligga på 8 MW. Eldning med fastbränsle är en trög process och att hålla börvärdet på framvattentemperaturen är mycket svårt. Mycket förenklat kan tidsfördröjningen i processen sägas ligga på 15 minuter. Detta är tiden från det att man har fått indikation på att flödet ökar och att temperaturen på nätet minskar tills dess att man försett pannan med bättre bränsle, eldat upp pannan och fått upp framvattentemperaturen. Att ett system som detta blir svängigt är lätt att förstå.

Det är emellertid inte bara framtemperaturen som är viktig att styra. Det gäller att reglera pannan på ett sådant sätt att slitage och underhåll minimeras. Det gäller alltså att låta skruvar gå så lite som möjligt eller t.ex inte starta upp stora fläktar oftare än nödvändigt.

Vi ska nedan gå igenom de funktioner som vi behandlat i examensarbetet.

Pulsregulator

I många pannanläggningar och luftbehandlingssystem arbetar reglerspjällen nära stängt läge, där spjällrörelsen har stor inverkan på luftmängden (man erhåller stor förstärkning i reglerkretsen).

Reglering med konventionella verkningsätt som PI ger stora översvängningar och ofta självsvängning i reglerkretsen.

Pulsregulatorn med inställbara pulszoner och pulslängd har visat sig klara dessa reglerproblem se fig. Genom pulsning av ställdonet i närheten av neutralzonen minskas hastigheten på spjällrörelsen och trycket kan "smyga" in mot börvärdet. Med den inställbara pulslängden kan spjällrörelsen göras mycket liten för varje puls.

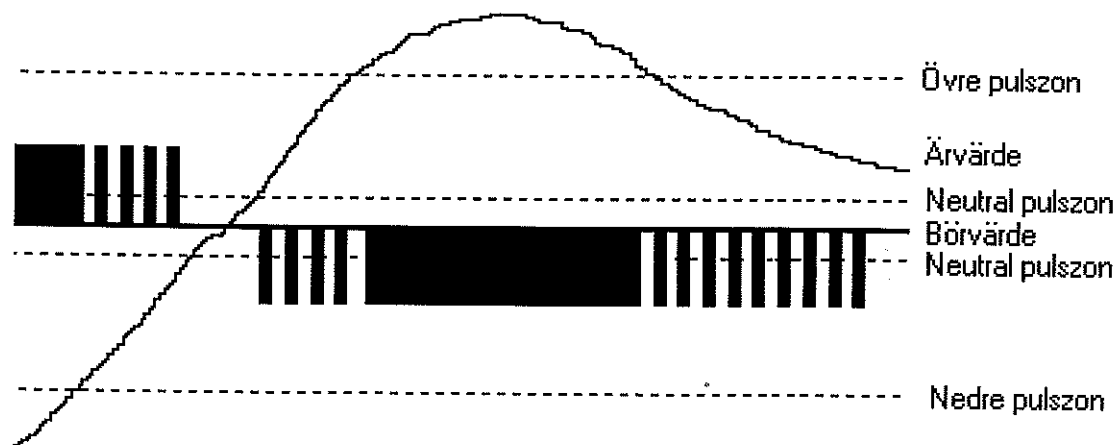


Fig.15 Pulsregulatorn

On-Off är en funktion som stänger av resp sätter på pannan. Att pannan är avstängd betyder inte att den är avställd. Det som är avstängt är fläktarna. Pannans effekt är då i jämvikt med självdraget och glödbädden i eldstaden. Anledningen till att man ibland måste On-Off reglera pannan är att man inte kan reglera ner effekten under 2 MW. Verkningsgraden under denna effekt blir låg.

Inmatning av bränsle in i ugnen sker via 2 st inmatningsskruvar. Tidigare styrdes inmatningen via tidreläer som i sin tur styrdes av kapacitetsregulatorn. Denna reglering fungerade mycket dåligt med varierande storlek på bränslebädden som följd. Anledningen till att regleringen fungerade dåligt var att varierande bränslekvaliteér brinner upp olika snabbt. Vår reglering av hur mycket bränsle som skall matas in i eldstaden styrs av tryckskillnaden över bränslebädden, Delta-p. Primärfläkten som sitter under rostern blåser in luft under rosterbädden och det är när luften tränger sig igenom bränslet som tryckfallet uppstår. Dock kan man inte ha ett fast börvärde på Delta-p eftersom detta varierar med läget på primärspjället. Därför försöker man beräkna hur stort börvärdet skall vara m.a.p primärspjällets läge. Man anger vilket börvärde man har när man kör med helt öppet spjäll samt börvärdet när man kör med helt stängda spjäll. Utifrån dessa två punkter beräknar sedan pannprogrammet det linjära sambandet mellan primärspjällets läge och börvärdet på Delta-p.

Styrningen av inmatningsskruvarna sker sedan enligt On/Off principen. Dock har man möjlighet att On/Off reglera med hysteres.

Ibland kör man med pannan i On-mode men med primärfläkten avstängd. Då faller hela idén med att reglera inmatningen på Delta-p eftersom tryckfallet över bränslebädden blir försumbart. Då går man automatiskt in och reglerar med tidperiodiserade På/Pauscykler.

Det finns fler situationer som påverkar inmatningen. Ibland råkar man få in för mycket torrt bränsle i ugnen. Detta leder till en våldsam förbränning med låg O₂-halt som resultat. I denna situation väljer man att blockera inmatningen tills dess att O₂-halten är normal igen.

Ibland råkar man få in för mycket bränsle i pannan. Diagnosen som konstaterar detta ställer man på att man har hög CO-halt samt låg O₂ halt. Åtgärden är enkel, - sluta med att mata in bränsle. Man får emellertid aldrig helt upphöra med inmatningen alltför lång tid. Då finns det risk för att brand sprider sig från eldstaden, genom inmatningsskruvarna, ut i inmatningsfickan. För att undvika detta låter man inmatningsskruvarna gå en kort stund med jämna mellanrum.

När pannan går i Off-mode går skruvarna på tidstyrning.

Askutmatningen styrs av hur mycket bränsle man har tillfört eldstaden. Tidigare gick askutmatningen på tidsstyrning, vilket ledde till att askutmatningen fick arbeta onödigt mycket. Vi har gjort som så, att vi räknar tiden som inmatningsskruvarna har gått sammanlagt sedan förra askutmatningsproceduren. På detta sätt kan vi räkna ut hur mycket aska som bildas i ugnen. När inmatningsskruvarna har gått en viss tid sätts en flagga att en askutmatningsprocedur skall börja. Vid denna tidpunkt är tiden för askutmatningsproceduren lika med noll. I denna tidpunkt börjar slaggskruvarna vid containern gå. I tidpunkt 1 börjar även "slaggskruv liggande" samt "slaggskruv" att arbeta. I tidpunkt 2 börjar även de fyra askutmatningsskruvarna att mata ut, och i tidpunkt 3 upphör hela askutmatningsproceduren.

Ljudsotning sker av pannan med en tyfon kopplad till en kompressor.

När pannan är i On-mode, så ljudsotar man tidsperiodmässigt.

När pannan är i Off-mode, så ljudsotar man de första sekunderna efter omslag från On.

Rostern styrdes tidigare tidsperiodmässigt av kapacitetsregulatorn. Man lät rostern göra en hel rörelse (fram och tillbaka). En hel rörelse medförde en alltför kraftig påverkan av processen, något som resulterade i att förbränningen tillfälligt saknade luft. Vår styrning av rostern går helt och hållet på hur mycket bränsle som man tillför eldstaden. När inmatningsskruvarna sammanlagt gått ett önskat antal sekunder, så går rostern 6 sekunder. Denna tid motsvarar ungefär 1/6 av en hel rosterrörelse vilket gör att processen påverkas jämnare.

Bragg som är en förkortning för Hydrauliskt bränsle-omröraraggregat sitter i inmatningsfickan. Den styrs precis som rostern. Tidigare gick den med inmatningsskruvarna vilket var onödigt mycket.

Slangfiltret rensas av en kompressor. När pannan är i On-mode så rensas man tidsperiodiserat till/från. Efter omslag till Off-mode så filterrensar man de första sekunderna.

Fläktarna är på när pannan är i On-mode, avstängda när pannan är i Off-mode. När man skall starta upp fläktarna så kan man inte starta alla fläktar samtidigt. Därför kan man ange hur många sekunder som man vill att det skall ta innan fläkten skall starta. Rökgasfläkten måste starta innan primärluftfläkten för att man skall erhålla undertryck i pannan först. Fläktuppstart börjar alltid med att rökgasfläkten får startorder i samma sekund som pannan går över i On-mode. Efter en viss tid startar sedan sekundärfläkten och efter ytterligare en tid startar primärfläkten. Observera att fläktarna måste hinna stänga sina spjäll innan start sker. Rökgasfläkten är t.ex elektriskt förreglad, så att start ej kan ske om spjället inte är stängt. Detta beror på för hög startström vid öppet spjäll.

När pannan är i On-mode kan man ange en panntemperatur-nivå som endast stänger av primärfläkten. Hysteresen på denna On-Off reglering är satt till 2 grader C.

Rökgasspjället reglerar undertrycket i eldstaden. Reglering sker enl. principen pulsreglering.

Man måste ange hur lång tid som skall löpa från det att pannan gått i On-mode, tills dess att regulatorn skall börja reglera.

Primärspjället reglerar effekten i eldstaden. Som insignal används pannvattentemperaturen. Regleringen skedde tidigare via kapacitetsregulatorn. I och med varierande bränslekvaliteér så fungerade inte denna reglering. Vår reglering sker enl. principen för pulsreglering.

Till sin hjälp med att hitta den lägsta effektnivån där förbränningen blir effektivast, så kan man sätta ett s.k minläge. Detta minläge anger det minsta läget som regulatorn får styra spjället till. Dock är tyvärr rökgasfläkten den effektbegränsande faktorn i ugnen, varför man kan sätta ett lägsta tillåtna värde på eldstadsundertrycket. Om detta värde underskrids så stänger primärspjället ändå.

Man måste ange hur lång tid som skall löpa från det att pannen gått i On-mode tills dess att regulatorn skall börja reglera.

Sekundärspjället reglerar O₂-halten i eldstaden. Reglering sker liksom tidigare enligt principen för pulsreglering.

Till sin hjälp med att hitta den lägsta effektnivån där förbränningen blir effektivast, så kan man sätta ett s.k minläge. Detta minläge anger det minsta läget som regulatorn får styra spjället till. Dock är tyvärr rökgasfläkten den effektbegränsande faktorn i ugnen, varför man har möjlighet att sätta ett lägsta tillåtna värde på eldstadsundertrycket. Om detta värde underskrids, så stänger sekundärspjället ändå.

Man måste ange hur lång tid som skall löpa från det att pannen gått i On-mode tills dess att regulatorn skall börja reglera.

Tertiärluftspjället reglerar CO-halten i eldstaden. Reglering sker enl. principen pulsreglering.

Till sin hjälp med att hitta den lägsta effektnivån där förbränningen blir effektivast, så kan man sätta ett s.k minläge. Detta minläge anger det minsta läget som regulatorn får styra spjället till. Dock är tyvärr rökgasfläkten den effektbegränsande faktorn i ugnen, varför man har möjlighet att sätta ett lägsta tillåtna värde på eldstadsundertrycket. Om detta värde underskrids, så stänger tertiärspjället ändå. Tertiärfläkten är elektriskt sammankopplade med sekundärfläkten, varför ingen separat tid anges för hur lång uppstart som erfordras.

Vi har infört en funktion som gör att om det råder akut brist på syre i eldstaden, så öppnar tertiärspjället för att hjälpa upp O2-halten.

Larm: Följande larmindikeringar har vi beaktat i examensarbetet.

- Utlöst motorskydd primärbetydelse
- Bränslebrist i inmatningsficka
- Utlöst sprinkler i bränsleficka
- Låg oljenivå i rosterns hydraulaggregat
- Hög temperatur på hydrauloljan till rostern
- Utlöst nedeldningstermostat
- Utlöst maxtermostat
- Lågt pannvattentryck
- Hög pannvattentryck
- Låg pannvattencirkulation
- Utlöst katastrofskydd
- Utlöst övertrycksvakt eldstad
- Låg rökgastemperatur panna
- Hög rökgastemperatur panna
- Lågt lufttryck kompressor
- Utlöst motorskydd på inmatningsskruvar
- Utlöst motorskydd sekundärbetydelse
- Hög rökgastemperatur innan filtret
- Hög rökgastemperatur efter filtret
- Hög temperatur vid inmatningsskruvar
- Utlöst motorskydd slaggskruvar
- Problem i vattenkulvert
- Problem med vattenreningen
- Problem med fläktkylning på vinden
- Utlöst motorskydd rökgasfläkt
- Kundventil problem (kallas topp)
- Stoftutmatningsproblem
- Hög temperatur i filtret
- Problem i grovavskiljare
- För högt difftryck på varmvattennätet
- Utlöst säkring primärfläkt
- Utlöst säkring tertiärfläkt
- Mintemp
- Hög CO-halt

5.2 PANNPROGRAMMETS PARAMETRAR

För att förstå hur PBS-programmet SattProg fungerar, är det lättast att titta på de funktioner som är kopplat till operatörsgränssnittet. Operatörsgränssnittet är ju kopplat till SattCon 15-20 via en seriell kabel. Två typer av information slussas fram och tillbaka via denna kabel.

* Register

* Bitar

Register är ett 16 bitars ord som kan anta värde mellan 0 och 65536. Det finns 512 olika register. Ett register kan knytas direkt till en variabel, som man namnger speciellt i en s.k identifierarlista. När man väl har döpt ett register behöver man aldrig tänka på att det t.ex är register R068 som är CO-halten. Istället skriver man bara CO-halt, när man menar registret som innehåller aktuell halt. Bitar fungerar precis likadant. En bit är en statusflagga, som kan anta värdet 0 eller 1.

Tyvärr kan ett namn bara innehålla 8 bokstäver, vilket gör att namnen lätt blir lite kryptiska, när man har 10 olika register som är snarlika i sin funktion.

Nedan skall vi gå igenom de register och bitar, samt deras namn och betydelse, som slussas fram och tillbaka mellan operatörsgränssnittet och Sattprogs förbränningsmoduler. Vi går även igenom några av de viktigaste interna statusparametrarna i pannprogrammet som operatören inte tar del av. Kännedom om vissa globala variabler i pannprogrammet kan vara till nytta om man vill sätta sig in i pannprogrammets exakta uppbyggnad.

On-Off

| | | |
|------------|--------|---|
| Ontemp | (R042) | Pannvattentemperaturen som pannan skall starta upp vid. |
| Offtemp | (R043) | Pannvattentemperaturen som pannan skall stängas av vid. |
| On-Off | (526) | Intern statusflagga som anger om pannan är i On-mode eller Off-mode (1=on, 0=off). |
| Off-tid | (R017) | Antal sekunder som pannan varit i Off-mode sedan start. |
| On-tid | (R025) | Antal sekunder som pannan varit i On-mode sedan start. |
| Pflofftemp | (R079) | Man är inte helt bunden till att On/Off köra hela pannan. Denna temperaturnivå stänger av primärfläkten. Fläkten startar igen när pannvattentemperaturen sjunkit till 2 grader C. |

Inmatning

| | | |
|---------------|--------|--|
| Deltböp | (R075) | Delta-P börvärde vid öppet primärspjäll. |
| Deltbst | (R076) | Delta-P börvärde vid stängt primärspjäll. |
| Deltahyst | (R110) | Hysters för On/Off regleringen av inmatningen. |
| Deltap-On | (R037) | Delta-P värde vid vilken inmatningen startat. |
| Deltap-Off | (R038) | Delta-P värde vid vilken inmatningen stoppar. |
| Ainmat-på-p | (R019) | Inmatningsperiod (s) för inmatningsskruvar när pannan är i Off-mode. |
| Ainmat-paus-p | (R021) | Pausperiod (s) för inmatningsskruvar när pannan är i Off-mode |
| Inmatpofft | (R093) | Inmatningsperiod (s) för inmatningsskruvar när endast primärluftfläkten är avstängd. |
| InmatÅofft | (R112) | Pausperiod (s) för inmatningsskruvar när endast primärluftfläkten är avstängd. |
| InmatCOmax1 | (R086) | Dessa två nivåer hör samman. Om båda nivåer överskrids (underskrids) så stoppar inmatningen. |
| InmatO2min1 | (R087) | Om detta O2 värde underskrids så stoppar inmatningen. |
| InmatO2min | (R088) | Om detta O2 värde underskrids så stoppar inmatningen. |
| Inmatsttid | (R095) | Detta är inmatningsstopptiden för skruvarna när inmatningen är stoppad. |
| Inmatpåtid | (R097) | Detta är inmatningsperioden som skruvarna går, trots att inmatningen är stoppad. |

Askutmatning

| | | |
|----------|--------|---|
| Askmängd | (R034) | Detta är tiden (s) som inmatningsskruvarna skall gå för att en askutmatningsprocedur skall börja. |
| Askperl | (R031) | Detta är tidpunktl i askutmatningsproceduren. |

| | | |
|---------|--------|---|
| Askper2 | (R032) | Detta är tidpunkt2 i askutmatningsproceduren. |
| Askper3 | (R033) | Detta är tidpunkt3 i askutmatningsproceduren. |

Ljudsotning

| | | |
|---------------|--------|--|
| Ljudsottid | (R052) | Antal sekunder som ljudsotning sker i Off-läge |
| Pljuds-på-p | (R027) | Ljudsotningsperioden i On-mode |
| Pljuds-paus-p | (R029) | Ljudsotningspaus mellan perioderna i On-mode. |

Roster

| | | |
|-------------|--------|---|
| Rosterstyrt | (R040) | Antal sekunder som inmatningen skall gå för att rostertrappan skall röra sig i 6 sek. |
|-------------|--------|---|

Bragg

| | | |
|------------|--------|--|
| Bransleagx | (R023) | Antal sekunder som inmatningen skall gå för att braggen skall röra sig i 30 sek. |
|------------|--------|--|

Filter

| | | |
|-------------|--------|---|
| Filtertid | (R024) | Antal sekunder som man rensar filtret efter det att pannan slagit om till Off-mode. |
| Filterrenp | (R049) | Rensningsperiod när pannan är i On-mode. |
| Filterpausp | (R051) | Tid mellan filterrensningarna när pannan är i On-mode. |

Fläktar

| | | |
|-------------|--------|--|
| Sekflpåtid | (R035) | Tidsfördröjning för starttidpunkt för sekundärfläkten. |
| Prifläktpåt | (R036) | Tidsfördröjning för starttidpunkt för primärluftfläkten. |
| Pflofftemp | (R079) | Panntemperatur vid vilken primärfläkten stänger av. |

Rökgasspjäll

| | | |
|-------------|--------|--|
| Bör | (R000) | Börvärdet på eldstadsundertrycket. |
| Pausperiod | (R247) | Sekunder mellan pulserna. |
| Pulsperiod | (R246) | Antal sekunder för en puls. |
| Underbör | (R003) | Undre gräns för inpulsningsområde. |
| Övrebör | (R002) | Övre gräns för inpulsningsområde. |
| Hyster | (R001) | Hysteres för börvärdet. |
| Rökflstartt | (R045) | Tidsfördröjning till dess att reglering får börja efter omslag till On-mode. |

Sekundärspjäll

| | | |
|-------------|--------|--|
| Bör | (R011) | Börvärdet på O2-halten |
| Pausperiod | (R251) | Sekunder mellan pulserna. |
| Pulsperiod | (R250) | Antal sekunder för en puls. |
| Underbör | (R014) | Undre gräns för inpulsningsområde. |
| Övrebör | (R013) | Övre gräns för inpulsningsområde. |
| Hyster | (R012) | Hysteres för börvärdet. |
| Sekflstartt | (R046) | Tidsfördröjning till dess att reglering får börja efter omslag till On-mode. |
| Minläge | (R015) | Minsta läget för spjället i On-mode |
| Sekundtrst | (R085) | Minsta tillåtna eldstadsundertryck för att sekundärspjället inte skall tvångsstänga. |

Primärluftspjäll

| | | |
|-------------|--------|--|
| Bör | (R006) | Börvärdet på panntempen |
| Pausperiod | (R241) | Sekunder mellan pulserna. |
| Pulsperiod | (R240) | Antal sekunder för en puls. |
| Underbör | (R009) | Undre gräns för inpulsningsområde. |
| Övrebör | (R008) | Övre gräns för inpulsningsområde. |
| Hyster | (R007) | Hysteres för börvärdet. |
| Priflstartt | (R053) | Tidsfördröjning till dess att reglering får börja efter omslag till On-mode. |

| | | |
|------------|--------|--|
| Minläge | (R010) | Minsta läget för spjället i On-mode. |
| Priundtry1 | (R080) | Minsta tillåtna eldstadsundertryck för att primärspjället inte skall tvångsstänga. |
| Priundtry2 | (R081) | Minsta tillåtna eldstadsundertryck för att primärspjället inte skall öppna. |

Tertiärluftspjäll

| | | |
|-------------|--------|--|
| Bör | (R099) | Börvärdet på CO-halten |
| Pausperiod | (R102) | Sekunder mellan pulserna. |
| Pulsperiod | (R104) | Antal sekunder för en puls. |
| Underbör | (R107) | Undre gräns för inpulsningsområde. |
| Övrebör | (R108) | Övre gräns för inpulsningsområde. |
| Hyster | (R100) | Hysteres för börvärdet. |
| Minläge | (R101) | Minsta läget för spjället i On-mode |
| Tertundtrst | (R083) | Minsta tillåtna eldstadsundertryck för att sekundärspjället inte skall tvångsstänga. |
| Tert02öppna | (R084) | O2-halt som gör att tertiärspjället tvångsöppnar. |

Mätvärde som presenteras på skärmen

Dessa mätvärde är de som presenteras av operatörsgränssnittet på skärmen. Dessa värde används även som dataunderlag till hela processtyrningen.

| | | |
|------------|--------|---|
| Co-halt | (R069) | CO-halten i eldstaden. |
| Delta-p | (R070) | Tryckskillnaden över bränslebädden. |
| Ecogastemp | (R073) | Rökgasttemperaturen efter economizern. |
| Elduttryck | (R071) | Eldstadsundertrycket. |
| Framtemp | (R065) | Levererad vattentemperatur. |
| Panntemp | (R129) | Temperatur efter panna men innan oljepanna. |
| O2-halt | (R068) | O2-halten i eldstaden. |
| Returtemp | (R066) | Returvattentemperaturen till anläggningen. |
| Rökgastemp | (R072) | Rökgasttemperaturen innan economizern. |

| | | |
|-------------|--------|-------------------------------|
| Ugnstemp | (R067) | Temperturen inne i eldstaden. |
| Lägeprimspj | (R074) | Primärspjällsläge. |
| Lägesekspj | (R016) | Sekundärspjällsläge. |
| Lägetertspj | (R106) | Tertiärspjällsläge. |
| Effekt | (R113) | Levererad total effekt. |
| Flöde | (R120) | Vattenflödet på nätet. |

5.3 PANNPROGRAMMET OCH DESS MODULER

Pannprogrammets huvudprogram heter Sattprog och är skrivet i PBS-språk. Programmet är utvecklat i programmet DOX5. SattProg består av ett antal moduler som tillsammans utgör pannstyrningsprogrammet. Varje modul innehåller en eller flera funktioner. För att förstå eller hitta i programmet behöver man inte veta hur varje modul är uppbyggda, utan man kan beakta varje modul som en svart box. Nedan sker en genomgång av varje moduls in- och utparametrar samt hur de påverkar processen.

Modulerna som börjar på A är alla avsedda för pannstyrningen när pannan är avstängd (offmode). Modulerna som börjar på P är avsedda för pannstyrningen när pannan är på (On-mode).

Huvudprogram: Sattprog

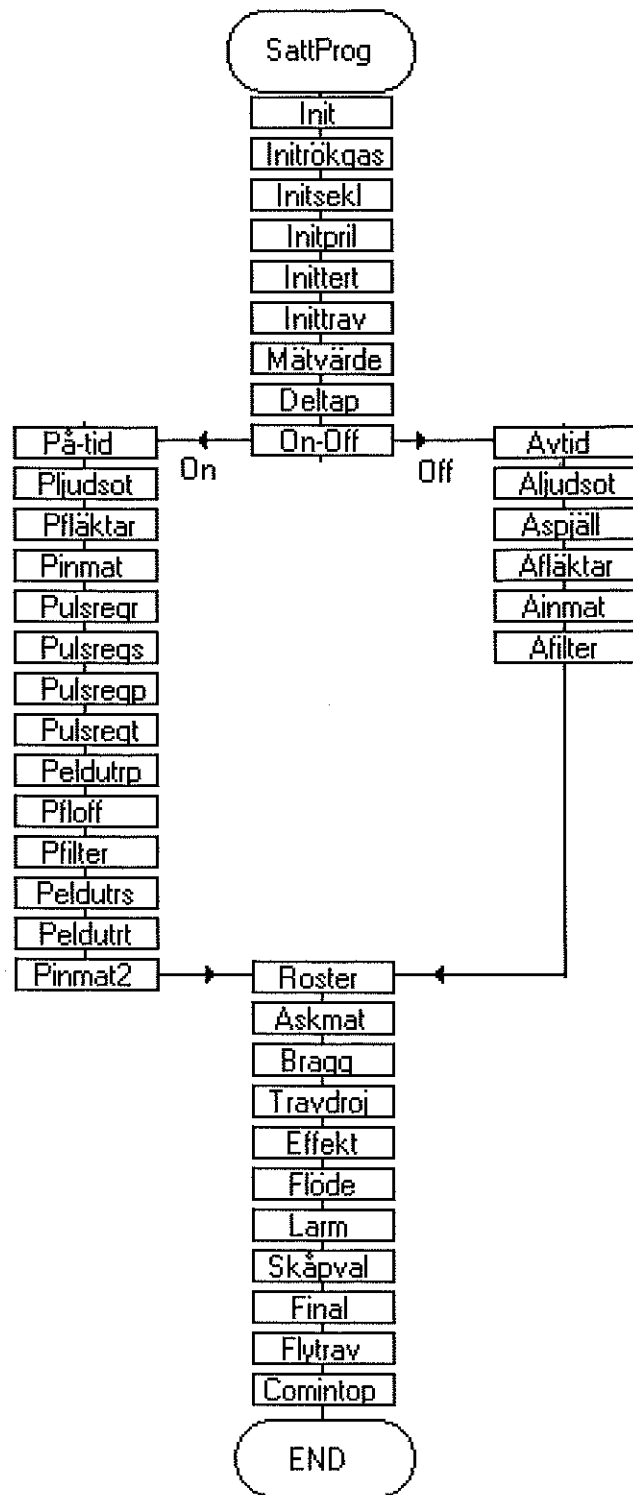


Fig.16 Flödesschema för pannprogrammet Sattprog

Init (Initiera)

Funktion: Initierar parametrar som berör pannan.
Initiering sker första svepet efter påslag av SattCon 15-20

Ut: Ontemp, Offftemp, Askper1, Askper2, Askper3, Askmängd, Ljudsottid, Rosterstyrt, Ainmat, Ainmat-pa-p, Ainmat-paus-p, Afiltertid, Pljuds-pa-p, Pljuds-paus-p, Rökfläktstartt, Sekfläktstartt, Priflstartt, Sekflpåtid, Prifläktpåtid, Co-gräns, Filterrenp, Filterpausp, Deltaböp, Deltbst, Deltahyst, Pfllofftemp, Sinmat-på-p, Sinmat-Paus-P, Priundtry2, Priundtry1, Sekundtrst, Terto2öppna, Tertundtrst, Inmatcomax1, Inmato2min1, Inmato2min, Sinmat2-pas-p, Sinmat2-PÅ-P, Energiper, Flödeper, Bransleagx, Traversmatp

Initrokg, Initsekl, Initpril, Inittert

(Initierarökgasspjällreglering,
InitieraSekundärspjällreglering
InitieraPrimärspjällreglering,
InitieraTertiärspjällreglering)

Funktion: Initierar parametrar som påverkar styrningen av eldstadsundertrycket, O2-halten, effekten samt Co-halten. Initiering sker första svepet som SattCon 15-20 är påslagen. Modulerna är uppbyggda på exakt samma sätt.

Ut: Bör, Pausperiod, Pulsperiod, Minläge, Spjälläge, Underbör, Övrebör, Hyster

Inittrav (InitieraTraversparametrar)

Funktion: Initierar alla parametrar som SattCon 15, TraversPc:n samt operatörsgrenssnittet behöver för att starta. Initiering sker första svepet efter att man slagit på SattCon 15-20.

Ut: K(1-60), Traversmatp, Omlastpåav, Inmatföljd(1-9)

Mätvärde (MätvärdesHantering)

Funktion: Hämtar in värde från AD-omvandlare samt skalar om värdena så att de stämmer efter sina enheter.

Ut: Framtemp, Returtemp, Ugnstemp, O2-Halt, CO-Halt, Delta-p, Elduttryck, Rokgastemp, Ecogastemp, Panntemp, Pspjälläge, Tspjälläge, Sekspjälläge

Deltap (Delta-P Börberäkning)

Funktion: Denna modul beräknar det flytande DeltaPBör värdet. Eftersom DeltaP (tryckskillnaden över bränslebädden) varierar med primärspjälläget, så måste även DeltaPBör räknas om beroende på primärspjälläget. Detta går till som så att man anger vad DeltaPBör skall vara när primärspjället är stängt samt när det är öppet. Modulen beräknar sedan, utifrån de 2 punkter som angivits, det rätlinjiga sambandet och erhåller börvärdet kopplat till aktuellt primärspjälläge. Efter denna lilla beräkning så räknar man ut On/Off nivåerna som sedan skall styra inmatningsskruven. On/Off nivåerna bestäms av vilken hysteres man vill ha.

In: Deltböp, Deltst, Pspjälläge, Deltahyst
Ut: Deltap-on, Deltap-off

On-Off (On-Off mode för pannan)

Funktion: Denna modul beräknar om panna skall gå i on mode eller off-mode. Detta beror helt och hållet på panntemperaturen.

In: Framtemp, Ontemp, Offtemp
Ut: On-off, Ontid, Uppstart, Stängning

Av-Tid (Av-Tid uppräknig)

Funktion: Det enda denna modul gör är att den räknar upp tiden som gått sedan pannan gått i Off mode.

Ut: Offtid

Aljudsot (Avstängdpanna Ljudsotning)

Funktion: Denna modul sköter om ljudsotningen när pannan har gått off.

In: Offtid
Ut: Ljudsotf

Aspjäll (Avstängdpanna Spjällstängning)

Funktion: Denna modul stänger alla spjäll utom rökasspjället.

Ut: Öppnarspjf, Stängrspjf, Öppnarspjf,
Stängpspjf, Öppnasspjf, Stängsspjf,
Öppnatpjf, Stängpspjf

Afläktar (Avståndpanna Fläktavstängning)

Funktion: Denna modul stänger av alla flaggor.
Ut: Rökfläktfl, Prifläktfl, Sekfläktfl

Ainmat (Avstängdpanna Inmatning)

Funktion: Denna modul hanterar inmatningen när pannan är i off-mode. Inmatningen sker tidsperiodmässigt med inmatningsperiod samt pausperiod.

In: Ainmat-på-p, Ainmat-paus-p
Ut: Inmatsfl

Afilter (Avstängdpanna Filter)

Funktion: Denna modul ser till att filterrensning sker de första önskade sekunderna efter omslag till Off-mode av pannan. Dessutom ser modulen till att bypassning sker av filtret när pannan inte är off.

In: Av-Tid
Ut: Filterflag, Bypassf

På-Tid (På-Tid uppräknig)

Funktion: Denna modul räknar upp tiden för hur långtid panna varit i on mode.

Ut: On-tid

Pljudsotning(PannaPå Ljudsotning)

Funktion: Denna modul sköter om ljudsotningen av pannan i On-mode. Ljudsotning sker enkelt med tidsmässiga perioder

In: Pljuds-På-P, Pljuds-Paus-P
Ut: Ljudsotf

Pfläktar (PannaPå Fläktar)

Funktion: Denna modul startar upp fläktarna. För uppstart av fläktarna krävs att alla spjäll är stängda vilket modulen ser till. Uppstart sker efter självbestämda tider efter omslag till on för resektive fläkt. Modulen håller även reda på när resp fläkt har startat så

att reglerspärarna i systemet nollställs efter hand.

In: Uppstart, Stängning, Sekflstartt, Priflstartt, Sekflpåtid, Rökflstartt, On-tid, Prifläktpåt.
Ut: Uppstart, Sreglerspär, Preglerspär, Rreglerspär, Rökfläktfl, Sekfläktfl, Prifläktfl

Pinmat (PannaPå Inmatning)

Funktion: Denna modul är en av tre som har med inmatningen i On-mode att göra (även Pinmat2 och Pfloff). Denna modul sköter det normala sättet att inmata, nämligen att On/Off reglera inmatningen m.a.p deltap (tryckskillnaden över bränslebädden). Börvärdet på Deltap gör att om det hålls, så hålls rätt bränslemängd i eldstaden.

In: Deltap-On, Deltap-Off
Ut: Inmatsfl

Pulsregr (Pulsregulator till Rök-gasspjället)

Funktion: Denna modul styr Rök-gasspjället så att eldstadsundertrycket är det önskade. Regleringen sker med tidigare beskriven pulsregulator.

In: Rpulsper, Rpausper, Rhysteres, Rbörvärde, Runderbör, Röverbör, Eldutryck
Ut: Öppnarspjf, Stängrpsjf

Pulsregs (Pulsregulator till Sekundärluftspjället)

Funktion: Denna modul styr Sekundärluftspjället så att O2-halten i eldstaden är den önskade. Regleringen sker enligt principen med tidigare beskriven pulsregulator.

In: Spulsper, Spausper, Sminläge, Shysteres, Sbörvärde, Sunderbör, Söverbör, O2-Halt
Ut: Öppnasspjf, Stängsspjf

Pulsregp (Pulsregulator till Primärluftspjället)

Funktion: Denna modul styr primärluftspjället så att framvattentemperaturen är den önskade. Regleringen sker enligt principen med tidigare beskriven pulsregulator.

In: Ppulsper, Ppausper, Pminläge, Physteres, Pbörvärde, Punderbör, Pöverbör, Panntemp
Ut: Öppnasspjf, Stängsspjf

Pulsregt (Pulsregulator till Tertiärluftspjället)

Funktion: Denna modul styr Tertiärluftspjället så att CO-halten i eldstaden är den önskade. Regleringen sker enligt principen med tidigare beskriven pulsregulator.

In: Tpulsper, Tpauser, Tminläge, Thysteres, Tbörvärde, Tunderbör, Töverbör, CO-halt

Ut: Öppnatspjf, Stängtspjf

Peldutr (PannaPå Eldstadsundertryckvakt för Primärspjäll)

Funktion: Rökgasfläkten i panna är den svagaste länken i processen. När man kör med full effekt orkar inte rökgasfläkten suga ut alla rökgaser. Man är då tvingad att låta de andra spjällen minska lufttillförseln något. Denna modul har två nivåvakter. 1 nivå där primärspjället inte får öppna, en nivå där primärspjället stängs.

In: Eldstadsundertryck, Priundtry1, Priundtry2, Öppnatspjf, Stängtspjf

Ut: Öppnatspjf, Stängtspjf

Pfloff (PannaPå Primärfläkt separat Off funktion)

Funktion: Som hjälp till att reglera panneffekten kan man stänga av primärfläkten, trots att panna är i On-mode. När man gör detta så kan man inte längre reglera inmatningen på deltap eftersom det inte är något tryckfall över bränslebädden när primärfläkten är avstängd. Denna modul stänger av primärluftfläkten vid en viss panntemperatur samt ändrar inmatningen så att inmatningen blir paus/period styrd.

In: Pflofftemp, Sinmat-på-p, Sinmat-paus-p, Panntemp, Prifläktfl, Inmatsfl

Ut: Inmatsfl, Prifläktfl

Pfilter (PannaPå Filterrensning)

Funktion: Denna modul rensar slangfiltret med tryckluft med paus/på intervall när panna är i ON-mode. Om CO-halten överstiger visst gränsvärde, skjuts filterrensningen som bypassar filtret upp.

In: CO-Gräns, Filterrenp, Filterpausp
Ut: Filterflag, Bypassf

Peldutrs (PannaPå Eldstadsundertryckvakt för Sekundärspjäll)

Funktion: Rökgasfläkten i pannan är den svagaste länken i processen. När man kör med full effekt orkar inte rökgasfläkten suga ut alla rökgaser. Man är då tvingad att låta de andra spjällen minska lufttillförseln något. Denna modul har en nivå där sekundärspjället stängs.

In: Eldstadsundertryck, Sekundtrst, Öppnasspjf, Stängsspjf
Ut: Öppnasspjf, Stängpssjf

Peldutrt (PannaPå Eldstadsundertryckvakt och O2-vakt för Tertiärluftspjäll)

Funktion: Rökgasfläkten i pannan är den svagaste länken i processen. När man kör med full effekt orkar inte rökgasfläkten suga ut alla rökgaser. Man är då tvingad att låta de andra spjällen minska lufttillförseln något. Denna modul har en nivå där tertiärluftspjället stängs. Om O2-halten är extremt låg (under viss nivå), används tertiärspjället även till att blåsa in mer luft.

Denna modul handhar även denna funktion.
In: Eldstadsundertryck, Tertundtrst, Öppnatspjf, Stängtspjf, Tero2öppna
Ut: Öppnatspjf, Stängptsjf

Pinmat2 (PannaPå Inmatning2)

Funktion: Normalt styrs inmatningen på hur mycket bränsle det finns i pannan. Dock kan det ibland inträffa att förbränningen i pannan inte är riktigt bra p.g.a att man använder för fuktigt eller för mycket bränsle. Använder man för torrt bränsle räcker O2-halten inte till med hög CO-halt som följd. Då måste man minska mängden bränsle. Denna modul har två olika fall i vilka inmatningen blockeras. Ett fall är om O2-halten understiger en viss nivå. Då kan man säga att bränslet är för torrt. Det andra fallet är om O2-halten understiger en annan nivå samt CO-halten

överstiger en CO-nivå. I detta fall kan man säga att pannan har för mycket bränsle.

Båda ovan nämnda fall leder till att inmatningen upphör. Dock måste man mata in lite bränsle då och då för att inte elden skall sprida sig bakåt, genom inmatningsskruvarna, in i bränslefickan. Denna modul sköter även om detta.

In: O2-halt, CO-halt, Inmatsfl, InmatO2min
InmatCOmax1, InmatO2min1, Sinmat2-På-p,
Sinmat2-pa(u)s-p
Ut: Inmatsfl

Roster (Roster)

Funktion: När inmatningsskruvarna har gått ett visst antal sekunder så låter man rostern gå 6 sek.
In: Inmatsfl, Rostertid
Ut: Rosterpåfl

Askmat (Askutmatning till slaggskruvar vidare till container)

Funktion: Askutmatning går till enligt följande
Först sätts en flagga till att askutmatning skall börja. Denna flagga sätts efter ett visst antal inmatningssekunder. Först matar man ut sista askan närmast containern, kompletterar med att dessutom mata ut med slaggskruvarna samt för att till slut mata ut med alla skruvar innan askutmatningen är klar.
In: Inmatsfl, Asktid, Askper1, Askper2, Askper3
Ut: Ask1flag, Ask2flag, Ask3flag

Bragg (Hydrauliskt Bränsleomröraraggregat)

Funktion: Denna modul gör att när inmatningsskruvarna har gått 30 sekunder så går braggen vald tid.
In: Inmatsfl, Bransleagx
Ut: Bransleagf

Travdroj (Travers drivrutiner)

Funktion: Denna modul innehåller en del rutiner som är tidsmässigt strikta m.a.p realtidskraven. Denna modul skall ses som en del av traversprogrammet i TraversPC:n.

Den första drivrutinen håller reda på hur länge ultraljudsgivaren har varit aktiv i ett sträck. När tiden överstiger en viss nivå så sätts en flagga som säger att det är dags för traversen att mata in en ny skopa.

Den andra funktionen är att den gör en liten delay på signalen TRAVERS UTGÅNGSLÄGE. Utan denna delay finns det risk för att den seriellt anslutna traversPC:n bommar signalen.

En tredje funktion är att man kontrollerar att traversen verkligen rör på sig när den har fått signal om att flytta på sig. Om man inte fått en signal inom 30 sek. så sättes en felflagga.

En fjärde funktion är att man dubbelkontrollerar traversens läge när den skall mata in i bränslefickan. Genom att beräkna koordinatförflyttningarna på två olika sätt, så minskar man risken för att traversen gå fel. Om det upptäcks så sätter man en felflagga.

En femte funktion är att man detekterar om traversen kopplas in på manuell styrning. Om så är fallet så sättes en flagga om att oinitiering måste ske.

Den sjätte funktionen i denna modul är bränsleregleringsfunktionen. Den kontrollerar att effekten i pannan räcker till. Den kontrollerar även att CO-halten är under tillåtet värde. Om dessa två kriterier inte är uppfyllda, sättes en signal som säger att en skopa bra bränsle skall matas in direkt i bränsleinmatningsfickan.

Den sista och sjunde funktionen förhindrar att man försöker lägga ut motstridiga signaler till traversen. Anledningen till detta kan ske är att givarna är störningskänsliga.

In: Framtempmin, Røkgastempmin, COMax, Newx, Traversmatt

Ut: Braskopa, Matatraversfl

Effekt (Effektpuls omräknare)

Funktion: Denna modul beräknar tiden för fem energipulser. Därefter beräknas pulserna och tiden om. Resultet är en uppdatering av den levererade effekten.

In: Energipuls

Ut: Effekt

Flöde (Flödespuls omräknare)

Funktion: Denna modul beräknar tiden för 6 flödespulser. Därefter beräknas den åtgångna tiden och pulserna om till flödet på nätet varefter uppdatering sker.

In: Flödespuls

Ut: Flöde

Larm

Funktion: Denna moduls uppgift är bland annat att göra samma sak som det ordinarie säkerhets systemet. I stora drag så åtgärdas olika larm med att stänga av olika saker. För exakt inblick så hänvisar vi till kapitlet Pannprogrammets Parametrar

En annan funktion är att förse se operatörsgränssnittet med information, samt till att eventuella larm blir utskrivna på printer. Ett register innehåller det senaste larmet, en flagga säger att larmet fortfarande gäller och flagga säger att larmet skall printas ut. När operatörsgränssnittet skrivit ut larmet, så nollställer detsamma flaggan.

In: Alla larm,

Ut: Senastlarm, Printlarm, larmfl

Skåpval (Val av specialfunktion på Paneldörren till SattCon 15-20)

Funktion: Möjlighet att med strömställare stänga av vissa objekt. Man kan välja att stänga av Pcstyrningen, Roster, Inmatningen, Askutmatningen, Bypassning samt ljudsotning.

In: Pcstyrning, Blockrost, blockinmat, blockask, blockfilt, blockljudsot

Final

Funktion: Denna modul lägger ut alla status flaggor på de fysiska utgångarna.

In: Satusflaggorna

Ut: De fysiska utgångarna, se bilaga Signaler.

Travflyt (Flytta till önskad koordinat för travers, ej tralla)

In: Newx, Traversläge

Comintop (CO-halt till mintopsystemet)

Funktion: Förser Ljungbyverkets mintopsystem med en analog utgång 4-20 mA .

BESKRIVNING AV OPERATÖRSGRÄNSSNITTET

6.1 OPERATÖRSGRÄNSITTET

Operatörsgränssnittet är det datoriserade systemets ansikte utåt och egentligen det enda som driftspersonalen dagligen använder sig av för att påverka och övervaka processen. Det är därför viktigt att operatörsgränssnittet är användarvänligt och inte kräver specialutbildad personal för att utnyttja systemets hela kapacitet. Det är välkänt att det nyttjaren inte kan tillgodogöra sig eller behärska, tycker han heller inte om att använda. Det vi tycker att man skall kunna kräva av ett operatörsgränssnitt kan sammanfattas enligt nedan:

- * God tillgänglighet/användarvänlighet
- * Enkelt att avläsa börvärden, larm, vilka fläktar som är i drift etc.
- * Enkelt att snabbt ändra inställda parametrar såsom börvärden och annan funktionsdata.

Givetvis måste man beakta operatörsgränssnittets användningsområde. Skall det användas av tekniker för grundinställningar i forskningssyfte eller liknande, är kraven andra och mer inriktade mot exakt och enhetlig processbeskrivning. Däremot om operatörsgränssnittet skall användas dagligen av driftspersonal med fler uppgifter än att bara koncentrera sig på processen ser vi ingen anledning att använda sig av symboler av teknisk-ritningstyp. Åtminstone absolut inte i den utsträckningen som man kan se skräckexempel på, bilder nerlusade med information och data och i det närmaste helt obegripliga för någon som ej är väl insatt i processen.

De moderna operatörssystemen såsom t.ex. SattControls SattGraph 1000 eller Ahlströms Alcont system är alla mycket väl genomtänkta och uppfyller de krav man kan ställa vad det gäller god tillgänglighet. Man har i ökad utsträckning lyckats minska informationen på varje bild och istället zoomar man in vid behov av mer information alternativt så öppnas ett nytt mindre fönster ovanpå processbilden. Uppbyggandet av bilder görs i en grafisk editor alternativt med färdiga bilder. Till varje bild knyter man sedan olika in och utgångskort med färdig händelsestyrning beroende på bildtypen. Förnämliga system men allt för dyra för vårt försöksprojekt. I runda tal kostar ett SattCon övervakningssystem drygt 140.000 kr och ett Alcont system avsevärt mer.

Att själva skriva all grafisk bildbehandling var helt uteslutet p.g.a den tid det skulle ta. Vad vi behövde var en grafisk editor med möjlighet att rita våra bilder i, och därefter själva tillfoga program med processpåverkan. Fördelen med att använda ett fristående program för grafisk bilduppbyggnad var att man själv kunde välja datorsystem och dessutom fanns möjligheten att undvika att binda upp sig med licensavtal eller dylikt.

Det program vi kom att använda oss av heter MATRIX LAYOUT och är ett amerikanskt CASE program. Priset för programvaran var otroligt låga 2000 kronor. CASE är en förkortning av Computer Aided System Engineering och är en komplett programutvecklingsmiljö.

Man skriver inte programmet i ett högnivåspråk i traditionell mening, utan programmeringen sker via bildikoner som sedan översätts till maskinkod. Bildikonerna knyts samman i flödesscheman för längre instruktionsrepetoarer. Till en bildikon kan man sedan knyta en grafisk bild ritad i paintbrush eller i MATRIX egna ritprogram paint. Funktioner som filhantering, fönsterhantering, rullgardinsmenyer, matematiska algoritmer och mycket mer finns implementerat som bildikoner. Dessutom kan man skriva egna bildikoner om så önskas.

Vi har valt att inte lista operatörsgränssnittet i denna rapport eftersom den av Matrix Layout genererade källkoden är på 8000 rader (130 sidor). Vad vi har tagit med är en listning av huvudprogrammet samt den textfil från vilken vi klipper in våra importerade procedurer. I huvudprogrammet som är väldigt kort står det vilka procedurer som gör vad.

6.2 PROGRAMUTVECKLING I MATRIX

Vi skall förklara hur programmet fungerar genom att försöka beskriva tillvägagångssättet punkt för punkt, hur man tillverkar en bild som skall presentera mätvärde på skärmen.

Det första man gör är att man inifrån Matrix Layout skaffar sig ett huvudprogram. Vi har döpt vårt till Opcomm. I detta huvudprogram definierar man vilka menyer man vill ha, i vårt fall t.ex pannan.

Därefter skapar man i Matrix Layout erforderlig källkod i valfritt språk t.ex. C eller Basic, i vårt fall Turbo-pascal. Källkoden man erhåller innefattar ett skalprogram som visar upp en rullgardinsmeny med texten panna. Den fil som innehåller den speciella grafiska utvecklingskoden har filnamnet .LAY medan den skapade källkoden i vårt fall får benämningen .PAS.

Därefter gäller det att tillverka proceduren som skall verkställas när man klickar på panna i menyn. Först går man in i det grafiska ritprogrammet paint och ritar sin bild, definerar variabler som skall ritas ut på bilden, bestämmer läge och typsnitt på siffrorna samt definierar knappar som kan leda till att andra värde skall visas på skärmen.

När man är klar med detta så skapar man sin källkod till proceduren i Turbopascal (fil typ .PAS) och man får en unit. Om man går in i källkoden i uniten så kan man se proceduren och dess variabler. Proceduren är lätt att följa. Nu måste man tyvärr gå in i källkoden och lägga till kommunikationsprocedurerna från Comli. T.ex måste man lägga till

```
GetReg (2,65,framtemp);      (se kap 3.6 )
```

Vad detta tillägg gör är att proceduren GetReg hämtar upp värde från register 65 i SattCon 15-20 och tilldelar variabeln framtemp detta värde. Variabeln framtemp har man tidigare definierat inifrån Matrix Layout. Tvåan står för att kommunikationen sker via port två på SattCon 15-20.

När uniten är färdig (dvs att man har lagt in alla sina kommunikationsprocedurer) så är uniten klar för kompilering, varefter man får ut en s.k .TPU fil.

När detta är klart går man tillbaka in i källkoden i huvudprogrammet (som är av typen .PAS), importerar sin procedur panna från TPU filen, lägger in den på det ställe i huvudprogrammet, som man ville att pannan skulle presenteras, och kompilerar och erhåller en .EXE fil. Därefter är allting klart.

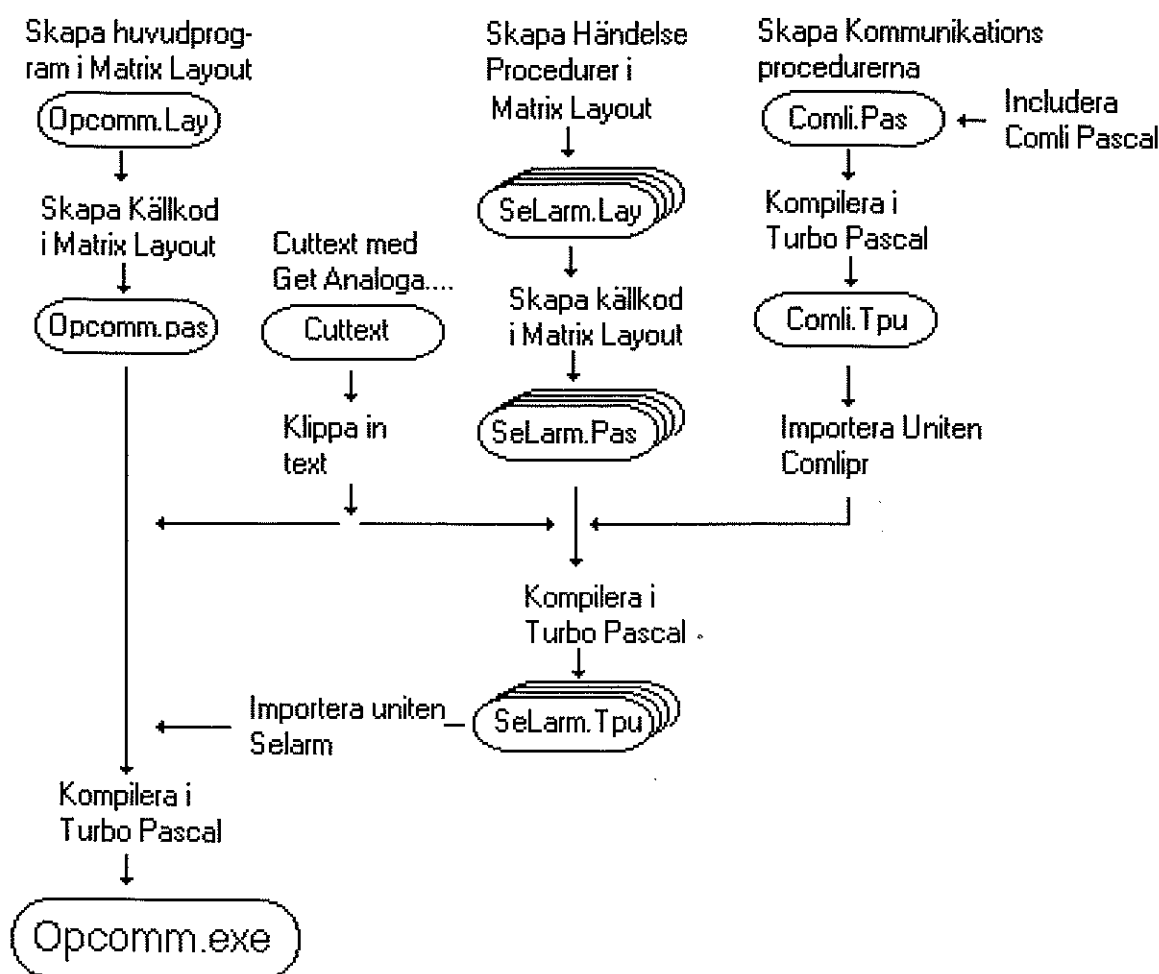


Fig.17 Arbetsgång för utveckling av Opcomm

6.3 OPERATÖRSBILDERNA

Alla förändringar av processparametrar sker från operatörsgränssnittet. Vanligtvis använder man sig av ett tangentbord vid val av nya parametrar eller växling mellan olika processbilder. Vi har valt att gå ett steg längre och helt plockat bort tangentbordet. Via ett pekverktyg, i vårt fall en mus, klickar man via menyer fram nya processbilder eller nya menyer med fler valmöjligheter. Vid ändring av parametrar klickar man på det värde man önskar ändra, varvid en miniräknardisplay ritas upp. Genom att klicka på räknarens tangenter erhålls önskat parametervärde.

På detta sätt undviker man inlärningsmomentet att lära sig vilka bilder som är knutna till speciella tangenter och man minskar även möjligheten till feltryckningar.

När det gäller bilderna i operatörsgränssnittet har vi gjort dem tredimensionella. Det är enligt vår uppfattning mycket enklare att orientera sig på en tredimensionell bild än vad det är på en konventionellt tvådimensionell.

Vi har även provat att frånga de standardiserade rit-symbolerna som används för anläggningsrepresentation. Vi har inte hittat på några nya symboler utan ritar helt enkelt av inmatningsskruven, reglerdonet, pannan eller vad det nu är vi vill avbilda. Vid behov görs en viss stilisering av föremålet för att undvika onödiga detaljer. Man skall givetvis inte vara rädd för att använda vettiga och tydliga standardiserade ritsymboler såsom symbolen för ventil, men många andra symboler är onödigt intetsägande, när man har möjlighet till bättre återgivning.

I så stor utsträckning som möjligt har vi valt att använda symboler, framför text eller färgmarkeringar, vid representation av pågående händelser eller valda funktioner. T.ex. markeras de koordinater som används för inmatning av bränsle till pannan med små spadar och de som ej får användas för inmatning markeras med en polisstopphandske. Om funktionen inmatning via Matterhorn är vald, markeras detta med en symbol för ett berg i processbilden. När pannans roster gör en rörelse visas det på samma sätt. Detsamma gäller för alla inställbara funktioner.

Syftet är främst att man så snabbt och enkelt som möjligt skall få information om pågående aktiviteter. Dessutom undviker man problemet med nersatt färgseende. Även vid inläring av systemet underlättar det att kunna binda en viss funktion till en symbol, förutsatt att symbolen är representativ.

Bilder på pannan och bränslehanteringsens operatörsgränssnitt finns på de två följande sidorna.

6.4 UNITEN COMLIPR

För att underlätta tillägget av ex. getreg i alla procedurer i operatörsgränssnittet så har vi gjort en unit som tillhandahåller samlade procedurer som hämtar upp/laddar ner många värden på en gång.

Ex **GetAnaloga** är en procedur som hämtar upp alla analoga (14 st) värden till operatörsgränssnittet på en gång. I och med detta så behöver man bara anropa **getanaloga** istället för att skriva **getreg(1)**, **Getreg(2)** o.s.v

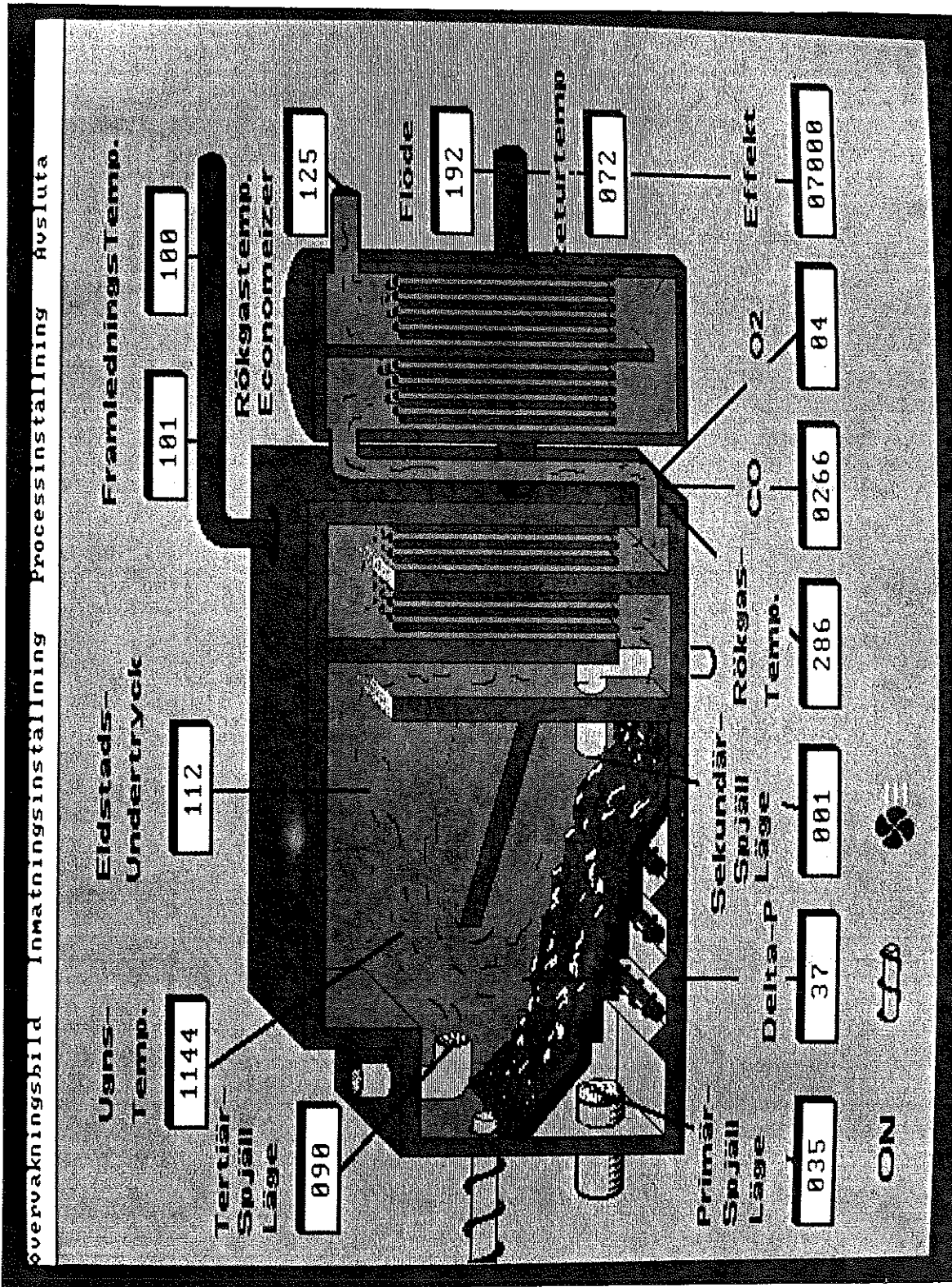
För alla de modifieringar som vi gör med huvudprogrammet samt procedurerna, så har vi samlat allting i en textfil kallad **Cuttext**. Från denna filen klipper vi in våra förändringar.

6.5 LARM OCH UTSKRIFTER

Vi har tidigare noga gått in på hur pannprogrammet och traversprogrammet fungerar. I och med detta så har vi även förklarat vilka parametrar och processdata som är viktiga (kan man processen så kan man även använda operatörsgränssnittet). Vi har dock knappt nämnt larmhanteringen.

I operatörsgränittet finns det olika funktioner som behandlar larm. Den första funktionen fungerar som så att om ett larm aktiveras så kommer en röd ikon märkt "LARM" upp på skärmen. Vi har valt att inte kombinera denna funktion med ljud eftersom hela anläggningen redan dränks i ringklockor när ett larm går. Vad som händer är emellertid, att det aktuella larmet skrivs ut på printern med tidsangivelse. Om man vill titta på vilket larm som aktiverats inifrån operatörsgränsnitett på skärmen, så måste man gå in på meny och klicka på "larm".

Bild panna

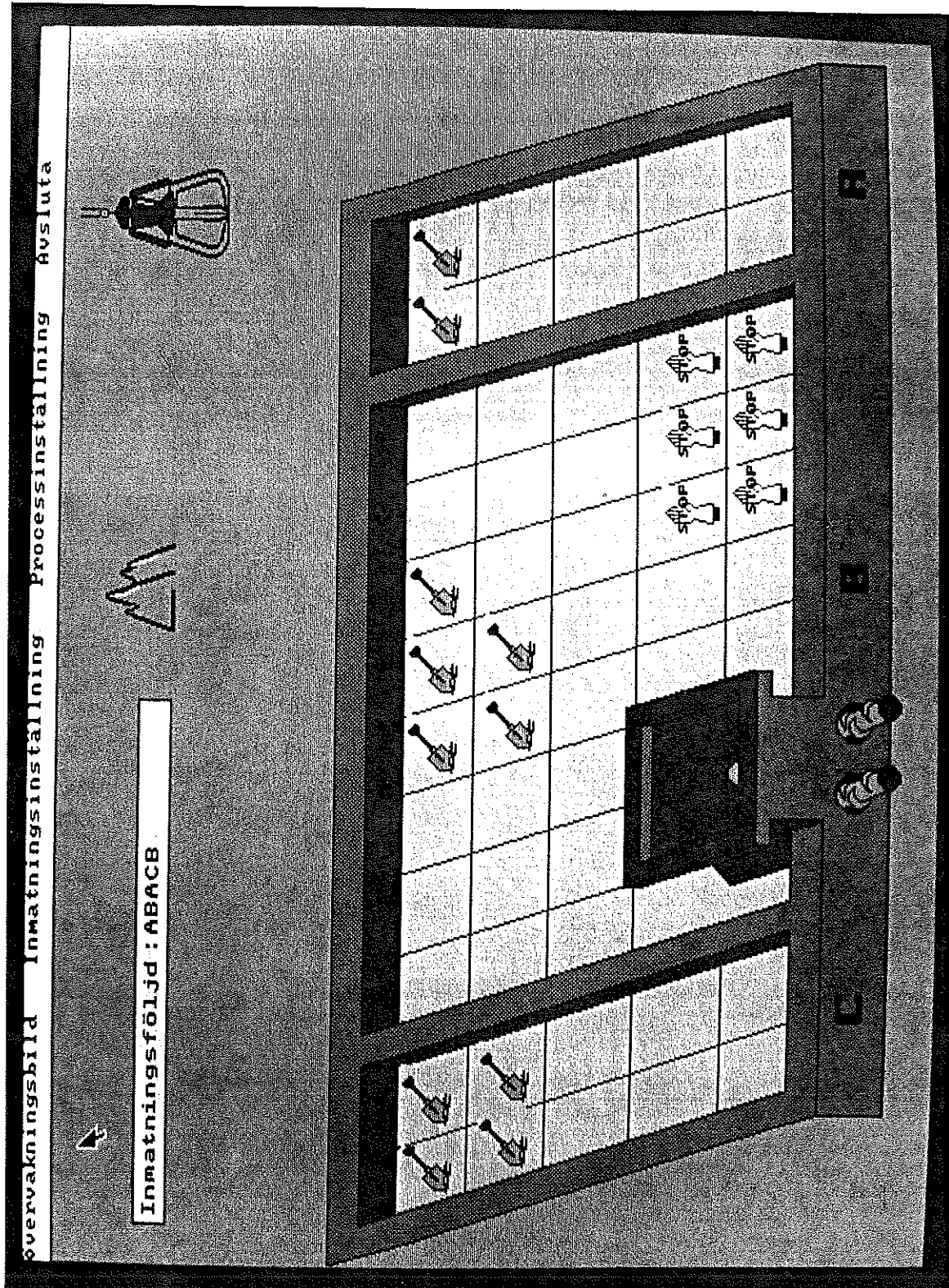


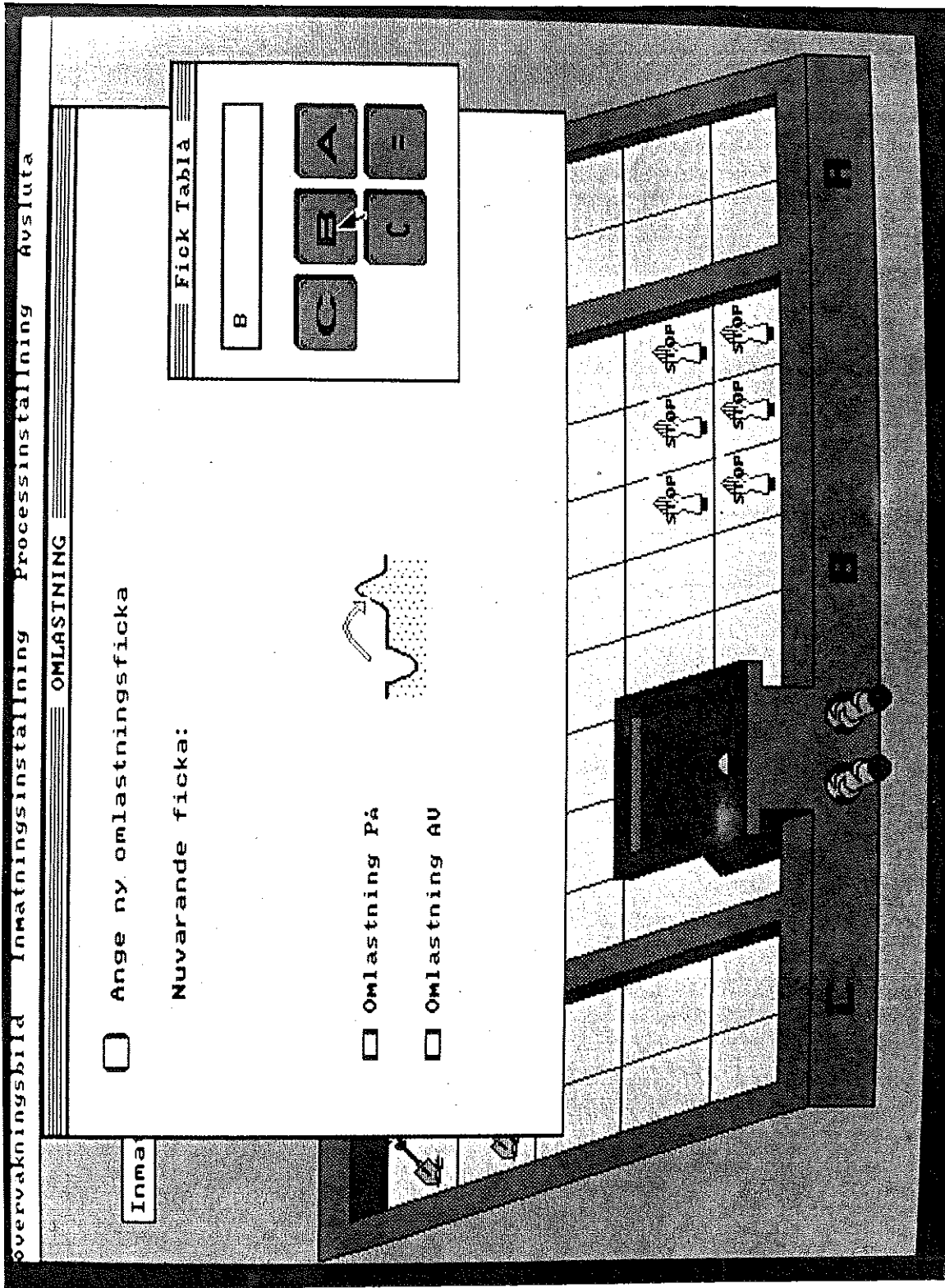
Övervakningsbild Inmatningsinställning Processinginställning Avsluta

Larm

- | | |
|---|---|
| <input type="checkbox"/> Utlöst Motorskydd Primär Bet | <input type="checkbox"/> Utlöst Vakt Inmatningen |
| <input type="checkbox"/> Bränslebrist | <input type="checkbox"/> Utlöst Motorskydd Sekundär Bet. |
| <input type="checkbox"/> Utlöst Sprinkler | <input type="checkbox"/> Hög Røkgastemperatur Economeiser |
| <input type="checkbox"/> Låg Ölje Nivå Hyd Agg Roster | <input type="checkbox"/> Temperatur Inmatning |
| <input type="checkbox"/> Hög Temp. Hyd Agg Roster | <input type="checkbox"/> Utlöst Slaggskruv Skydd |
| <input type="checkbox"/> Utlöst Nedlednings Termostat | <input type="checkbox"/> Kulvert |
| <input type="checkbox"/> Utlöst Maxtermostat | <input type="checkbox"/> VattenRening |
| <input type="checkbox"/> Lågt Pannvattentryck | <input type="checkbox"/> Fläkt Agg. Vinden |
| <input type="checkbox"/> Høgt Pannvattentryck | <input type="checkbox"/> Utlöst Motorskydd Røkgasfläkt |
| <input type="checkbox"/> Låg Pannvattencirkulation | <input type="checkbox"/> TF1, OF1, P6, P7 |
| <input type="checkbox"/> Utlöst Katastrofskydd | <input type="checkbox"/> Stoft Problem |
| <input type="checkbox"/> Utlöst Övertryckvakt Eldstad | <input type="checkbox"/> Filter Fickor |
| <input type="checkbox"/> Låg Røkgastemperatur | <input type="checkbox"/> Grovavskiljare |
| <input type="checkbox"/> Hög Røkgastemperatur | <input type="checkbox"/> Hög Røkgastemperatur Filter |
| <input type="checkbox"/> Lågt Lufttryck kompressor | <input type="checkbox"/> Diff Tryck Vakt |
| <input type="checkbox"/> Min Temp. | |







BESKRIVNING AV FJÄRRSTYRNINGEN

7.1 FJÄRRSTYRNINGEN

Fjärrstyrningens avsikt är att man skall ha möjlighet att hemifrån kunna koppla upp sitt operatörsgränssnitt mot anläggningen. För att göra detta så kan man använda sig av 2 olika principer.

Den första principen för fjärrstyrning innebär att man med en fjärrPC via modem fjärrstyr operatörsPC:n med hjälp av speciell programvara t.ex PC-Anywhere eller Carbon-Copy. Dessa program överför alla signaler som går till/från tangentbord och bildskärm via telenätet. Man exekverar alltså programmet i ordinare dator men överför det som kommer ut på bildskärmen och det som skrivs på tangentbordet. Det är emellertid en väl långsamt att använda en sådan princip i vårt fall eftersom operatörsgränssnittet innehåller mycket grafik som innehåller mycket information. Det skulle ta ungefär 1 min att överföra en färgbild i VGA grafik.

Den andra principen är den som vi använder oss av. Den går ut på att man endast överför de registervärde och statusbitar som man är intresserad av, medan bilderna som operatören vill titta på lagras och exekveras i fjärrdatorn. Genom att välja denna princip kan man använda sig av exakt samma program som programmet till operatörsgränssnittet.

Kommunikationen mellan fjärrPC:n med operatörsgränssnittet går med 9600 Baud, 8 databitar, 1 stoppbit och udda paritet. Genom att använda modem som man kan ställa in terminalen på 9600, 8, 1, Odd så kan man i stället koppla upp sig som figuren nedan visar.

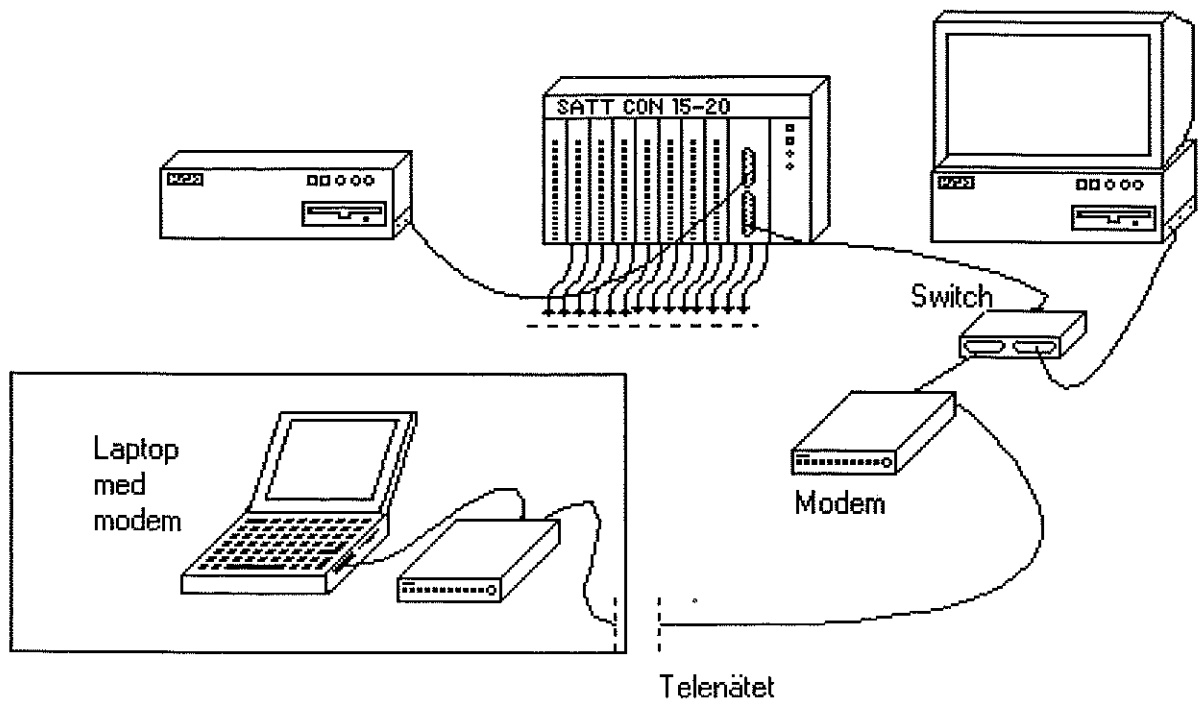


Fig.22 Översiktsbild fjärrstyrningen.

Det fungerar som så att operatören som sitter hemma, först måste koppla upp en fungerande överföringslinje. Detta gör han genom att använda sig av ett kommunikationsprogram som heter Procomm. Modemet, som är Hayes kompatibelt, ringer upp det andra modemet och upprättar kommunikation. Kommunikationen är simplex, dvs informationen går bara i en riktning i taget. Fjärrdatorn som är hemma hos jourhavande operatör är en laptop av det moderna lätta slaget. Därefter startar man sitt operatörsgränssnitt på vanligt vis och har exakt samma övervakningssystem hemma som man har på verket.

För att denna övervakning skall fungera så förutsätter det ju att operatören är uppkopplad hemifrån hela tiden. Detta är emellertid dyrt eftersom det trots allt kostar att använda telenätet. I början av rapporten där vi beskrev att det fanns ett säkerhetssystem, berättade vi om att larmsystemet redan hade en telefunktion. Denna funktion fungerar som så att om ett larm aktiveras så ringer larmsändaren upp jourhavande operatör. Denna uppringning säger inte vilket larm som aktiverats. Detta system använt i kombination med det nya larmsystemet kompletterar alltså säkerhetssystemet så att operatören blir varskodd om, att larm har gått även när fjärrstyrningen inte är uppkopplad.

7.2 MODEMET

Modemet som vi använder är som vi tidigare sagt Hayeskompatibelt vilket numera är standard för PC-modem. Det är dock ganska hårda krav på modemen som skall användas för vår applikation. För det första så måste överföringshastigheten vara 9600 Baud vilket är väldigt högt i modemsammanhang. De flesta modemen har en överföringshastighet på 2400 baud. Det andra stora kravet som kanske är ännu svårare att uppnå är att man skall kunna överföra med terminalinställningen 8, 1, odd vilket motsvarar 11 bitar för modemet att hantera. Det är ytterst få modem på marknaden som klarar av detta.

Det kan nämnas att vi hade väldig otur när det gällde modemen. Den första försäljaren garanterade att modemen skulle klara 8,1,odd vilket även den tekniska specifikationen sade. Efter 4 dagars förtvivlat testande drog vi slutsatsen att modemet måste ha en bugg. Försäljaren blev otrevlig och ifrågasatte vår kompetens. Han kollade emellertid upp det och mycket riktigt, det fanns en bugg. Arga pga förseningen skickade vi tillbaka modemen och beställde nya av en annan leverantör. Vi förhörde oss noga om modemet skulle klara våra krav. Svaret var ett självsäkert ja. Efter 3 dagars ytterligare väntan kom de nya modemen och förhoppningsfullt började vi testa. Men ack, inte fungerade de heller. Vi började tvivla på oss själva men envisades med att påstå att det måste vara en bugg även för den nya leverantören. Han kollade upp det med sin tekniska avdelning, och "såg" man på, det fanns tydligen en bugg. Till sist hittade vi en leverantör som kunde leverera oss modemen som vi önskade.

8 SAMMANFATTNING

De sista större förändringarna i programmen till systemet gjordes i December-90. Systemet var i full drift i slutet av Januari-91. Därefter har mindre buggar korrigerats parallellt med en viss funktionsutvärdering. Dessutom har fjärrstyrningen genomförts och har fungerat sedan Augusti-91.

Sammanfattningsvis fungerar det nya datoriserade styr- och övervakningssystemet mycket bra och har helt ersatt det ursprungliga, som dock körs parallellt med detta för att möjliggöra snabb övergång.

Nedan redogörs för de slutsatser vi kommit fram till rörande de olika delarna i projektet.

Bränslehanteringen

Skillnaderna mellan det ursprungliga traversprogrammet och det nuvarande är flera. Möjligheten till recept-blandning, val av enskilda koordinater, omlastning parallellt med inmatning och Matterhorn har alla bidragit till en avsevärt förbättrad och flexiblare bränslehantering.

Om man skulle peka ut någon del som visionär skulle det tveklöst vara Matterhornprincipen. Att på ett enkelt och precist sätt ha möjlighet att blanda samman olika bränslekvalitéer är ovärderligt.

Valet att placera traversprogrammet i en PC har fördelar såväl som nackdelar. Den stora fördelen har varit möjligheten att göra ändringar i traversprogrammet utan att behöva stänga av pannan. En annan stor fördel är att ett program skrivet i högnivåspråk är mer lättläst och därmed mycket lättare att överblicka och snabbt sätta sig in i.

Att skriva ett program i Turbopascal till en realtids-applikation ser vi dock som mindre lyckat. Vårt val av detta programspråk var dessvärre påtvingat av yttre omständigheter. Att det överhuvudtaget var genomförbart beror på att vi placerat vissa delar i SattCon.

Någon egentlig utvärdering av hållbarheten hos en PC vid kontinuerlig drift har vi inte kunnat göra, men att det skulle utgöra något egentligt problem kan vi inte se. Prisnivån är sådan att den möjliggör reserv PC och

dessutom är tillgången på reservdelar och service heltäckande.

SattControl släppte nyligen en ny programutvecklingsmiljö, Dox 10, för programmering i PBS. Programmeringen och dokumentationen är bildbaserad, uppbyggd av funktionsblock och med möjlighet till GRAFCET-representation av sekvenser. Problemet med PBS tidigare har varit just överskådligheten vid längre sammanhängande sekvenser av den typ som vårt traversprogram är uppbyggt av.

Detta sammantaget gör att vi skulle välja att placera ett framtida traversprogram i en PLC av typ SattCon eller något liknande system med möjlighet till dokumentation och programmering enligt ovan.

Förbränningsprocessen

Förändringarna i styr- och reglermetodiken för pannan skedde på basis av egna idéer samt värmeverkschefen Tommy Göranssons önskemål och rekommendationer.

Vi införde inga omvälvande förändringar i principen för styrningen av pannan, men de förändringar som gjordes har resulterat i att pannan idag går något bättre. Bl.a kan man idag få ut lite mer maxeffekt än vad man kunde tidigare. Även när det gäller CO-utsläppet har man möjlighet att minska denna genom noggrannare inställning av spjällen.

Största fördelen med de nya regulatorerna är att de är lätta att ställa in vilket ökar möjligheten att prova olika inställningar för att utvärdera bästa parametervalet vid olika laster. Noggranna tester kan med enkelhet genomföras, vilket man sedan kan låta vara grunden för utveckling av en bättre regulator baserad på t.ex. oskarp logik.

Bland de genomförda förändringar som gjordes kan det vara värt att nämna följande.

Vi valde att ta bort den gamla kapacitetsregulatorn eftersom denna inte fungerade nöjaktigt. I stället för att reglera alla spjäll, inmatningsskruvar, askutmatningsskruvar samt rostern på framtemperaturen så försökte vi införa lite mer "nyanserad" reglering. Det vi införde som kan betecknas som okonventionellt var att vi styr inmatningen på tryckskillnaden över bränslebädden, Delta-p. Denna idé har visat sig fungera mycket bättre än vad vi från början vågade hoppas på. Även funktionen med att räkna tiden som man har matat in bränsle för att få en hum om hur mycket aska som samlats i eldstaden har visat sig fungera väl. Askskruvarna går idag inte lika mycket

som de gjorde tidigare, vilket minskar kostnader för el och slitage.

Okonventionellt är även att vi låter rostern styras av hur mycket bränsle som har matats in i eldstaden och att vi inte låter rostern göra en hel rörelse (fram och tillbaka). Vi låter rostern göra 1/6 av en hel rörelse. När vi förhörde oss om denna idé bland ugnslieferantörer, så fick vi beskedet att denna funktion skulle vara helt verkningslös. Istället har det visat sig att vi får en jämnare nivå på CO- och O₂-halten. Detta har vi visat genom empiriska försök.

Vad gäller regleringen av framtemperaturen med pulsregulatorn kan sägas, att den inte fungerar riktigt så bra som vi hade hoppats. Regleringen fungerar bra vid höga effekter, men systemet är känsligt för större förändringar av effektuttaget på nätet. Detta beror på att parameterinställningen är ganska snäv och beroende av vilken effekt man producerar. Om man aktivt går in och ändrar på styrparametrarna allteftersom effekten varierar kan man få bra reglering. Det vida effektintervall som pannan rör sig inom (varierar mellan 2 och 8 MW på en timme) gör att man aktivt måste gå in och ändra på styrparametrarna. Gör man inte detta blir systemet snabbt svängigt och börjar automatiskt köras i någonting som kan liknas vid On/Off-styrning.

En starkt bidragande orsaken till att pannan idag går bättre är den förbättrade bränslehanteringen med traversen. Bränslet som matas in i pannan är idag homogent blandat, vilket leder till att pannan går lugnare (mindre svajigt) och därigenom mer till fördel för miljön. Utsläppen av kolmonoxid har minskat väsentligt.

Pannprogrammet var kanske det som var minst problematiskt att sätta i drift, något som vi är väldigt tacksamma för. Driftsättningen skedde under januari månad, då verket körs så hårt det någonsin tillåter. Detta gjorde att vi tyvärr inte hade någon möjlighet att prova funktion för funktion, utan var hänvisade till att låta pannan slås över på vårt system för att i sin helhet låta sig styras och regleras. Beroende på tur eller skicklighet så visade det sig iallafall att programmet fungerade någorlunda väl ifrån första stund. Vi kan berätta att vi var mycket nervösa första gången vi slog över på vårt system.

Kanske berodde frånvaron av idriftsättningsproblem på att denna del egentligen var den enda som löstes med konventionell teknik och metod. SattControls produkter är väl beprövade och fungerar bra. Den programutvecklingsmiljö vi använde, dox5, fungerar felfritt men kan karaktäriseras som föråldrat och omodernt. SattControls dokumentering är oftast grundlig och rejäl men irri-

terande luckor ställde till förtret. Vi måste dock nämna att SattControl erbjuder en fantastisk fin telefonsupport som vi stundtals utnyttjade flitigt.

Operatörsgränsnittet

Målet med operatörsgränsnittet var att skapa en god ergonomi för användaren och därmed ett system som var anpassat till användarens naturliga förutsättningar och behov i större utsträckning än många av dagens operatörs-system.

Därigenom skulle man kunna minska inlärningstiden och öka förståelsen och användarvänligheten av systemet.

Vi har hittills lärt upp fyra personer ur drifts-personalen på systemet, alla utan tidigare datorträning. Inlärningstiden är minimal och kan sägas stanna vid en genomgång på runt en timme. I det ingår inte utlärnning av regulatorer och hur de lämpligast ställs in, utan bara hur man gör nya parameter inställningar.

Vårt val att helt ta bort tangentbordet har blivit mycket positivt bemött. Det man uppskattar är, att man hela tiden kan hålla ögonen på skärmen, där man har pekverket, och inte måste växla mellan tangentbordet och skärmen.

Systemet var från början tänkt att i så stor utsträckning som möjligt vara självlärande genom möjligheten att klicka på HJÄLP och på så sätt erhålla en närmre beskrivning av olika funktioner. Tyvärr hann vi inte med denna bit under examensarbetet, men funktionen är förberedd och det är vår avsikt att fullfölja även denna bit, efter rapportens inlämnande.

Fjärrstyrningen

Fjärrstyrningen fungerar mycket bra och är en finess som nog kommer att utnyttjas av de flesta som skaffar sig ett datoriserat styr- och övervakningssystem.

Att bara sända processdata över telenätet och låta de grafiska bilderna ligga i den uppringande datorn är nog ett måste. Skall systemet verkligen användas regelbundet av operatören får det inte ta för lång tid att läsa upp bilderna till skärmen.

Ett annat problem man löser med att låta operatörs-programmen ligga i uppringande datorn är säkerhetsbiten. Har man inte tillgång till programmet saknar man möjlighet att ringa upp verkets modem och göra förändringar i parameterinställningarna.

LITTERATURFÖRTECKNING

- 1 SC15 Användarbeskrivning
- 2 Bildskärmssystem, SattControl AB
- 3 SC15 Snabbguide
- 4 SC15 Maskinvara
- 5 SC15 Programmeringsexempel
- 6 Comli Pascal
- 7 Dicom modem användarmanual
- 8 Matrix Layout användarmanual

9 BILAGOR

9.1 TRAVERSPROGRAMMET

PROGRAM TraverseProgram;

USES Crt,Printer;

CONST ON=1; {Aktiverar utg. SattCon}
OFF=0;
comligate=1; {Vald Ingång SattCon}

```
      {***digitala INSIGNALER***}  
lowerfuelstorelevel=141; {ger 1 vid låg bränslenivå}  
traversutg=1106; {SattCon 500} {Ändläge Travers}  
traversright=501; {SattCon 501} {Traversens främre del pass. Koord.mittpkt}  
stoppgivaretravers=502; {SattCon 502} {Passerar Koordinatens mittpunkt}  
trallautg=1107; {SattCon 503} {Ändläge Tralla}  
trallaright=504; {SattCon 504} {Trallans främre del pass. Koord.mittpkt}  
stoppgivaretralla=505; {SattCon 505} {Passerar Koordinatens mittpunkt}  
skopaovrelage2=506; {SattCon 506} {vid bränslemätning}  
skopaovrelage1=507; {SattCon 507} {vid omlastning}  
skopanedrelage1=510; {SattCon 510} {vid omlastning}  
skopanedrelage2=511; {SattCon 511} {vid bränslemätning}  
nollasttravers=512; {SattCon 512} {Skopan vid bränslenivå}  
traversleft=513; {SattCon 513} {Traversens bakre del passerar Koord.mittpunkt}  
trallaleft=514; {SattCon 514} {Trallans bakre del pass. Koord.mittpunkt}
```

```
      {***digitala UTSIGNALER***}  
traversminus=301; {SattCon 301} {Travers mot ändläge (utgångsläge)}  
traversplus=302; {SattCon 302} {Travers från ändläge}  
travershogfart=303; {SattCon 303} {Travers högfart}  
trallaminus=304; {SattCon 304} {Tralla mot ändläge}  
trallaplust=305; {SattCon 305} {Tralla från ändläge}  
trallahogfart=306; {SattCon 306} {Tralla högfart}  
lyftupp=307; {SattCon 307} {Skopa upp}  
sankned=310; {SattCon 310} {Skopa ned}  
sanksnabbt=311; {SattCon 311} {Sänk skopa snabbt}  
lyftsnabbt=311; {SattCon 311} {Lyft skopa snabbt}  
oppnaskopa=312; {SattCon 312} {Öppna skopa}  
stangskopa=313; {SattCon 313} {Stäng Skopa}
```

TYPE CharVektor = 'A'..'C';
CharType = STRING[1];
VektorType =ARRAY[1..13] OF STRING[1];
StringType = STRING[10];
VektorPost=RECORD

```
    feed : ARRAY[1..60] OF STRING[8];  
    reload : ARRAY[1..60] OF STRING[8];  
    X : ARRAY[1..70] OF INTEGER;  
    Y : ARRAY[1..70] OF INTEGER;
```

END;
StorePost=RECORD

```
    A : BOOLEAN;  
    B : BOOLEAN;  
    C : BOOLEAN;
```

END;

PointerType=^Post;
Post=RECORD
 data : CharVektor;
 next : PointerType;
END;

VAR i,k,
 presentX,presentY, {Anger traversens X läge resp trallans Y läge}
 mattertop, {Höjden på Matterhorntoppen}
 number, {Motsvarar vilket element i feedlist som är aktuellt}
 oldA,oldB,oldC,oldM, {Sist grävda koordinaten i fickorna A, B, C o Matterhorn}
 oldnumber, {Sist omlastade koordinat}

Bilagor

```

totalnumber,          {Anger antalet omlastnings koord. i varje ficka}
level,                {Flagga som används av comli vid bitläsning från Sattc.}
matterhornlevel,     {SattCon flagga som är 1 vid inmatning från Matterhorn}
Sattadress,          {Adress i SattCon till In eller Utgång}
fixposition,         {True då skopan hänger utanför bränslebehållaren}
oldDreload,         {Sist omlastade koordinaten i ficka D}
fuelcontrol,         {SattCon flagga som vid 1 ger reglering till bättre}
                    {bränsle}
auto                  : INTEGER; {SattCon flagga som vid 1 ger omlastning}

reload,              {True om omlastning skall ske}
feed,                {True vid bränslepåfyllning}
zeroload,            {SattCon flagga som är 1 när skopan ligger på bränslet}
feedtank,           {True om man vill att skopan skall öppnas strax innan}
                    {trailan stannar}
firsttimeA,firsttimeB,firsttimeC,firsttimeM,
                    {True första gången fickan används}
cancel,              {Alltid False ; Används till huvudprogram loopen}
firsttimereloading, {True första gången man påbörjar omlastning av ficka}
storeempty,         {True om alla koordinater i en ficka är markerade tomma}
matterhorn,         {True om inmatning skall ske via Matterhorn}
restingmatterhorn, {True om skopan befinner sig i väntläge i Matterhorn}
resting,            {True då gripskoppan är i viloläge i någon av fickorna}
error,              {True om något fel uppstår som måste leda till}
                    {omlastering}
firsttimetest,      {True varje gång en förflyttningsprocedur anropas i en}
                    {loop}
firsttimeABC        : BOOLEAN; {True första gången inmatn sker från någon av fickorna}
                    {A, B eller C}

exe                  : CharType;  {Motsvarar den ficka som används för tillfället}
reloadstore,        {Vektor som anger vilken ficka som skall omlastas}
oldreloadstore      : CharVektor; {Vektor som anger sist omlastade ficka}
square              : VektorPost; {Vektorpost innehållande varje koordinats status för}
                    {inmatning och omlastning}
directionX          : StringType; {'+' vid travers höger, '-' vid travers vänster}
directionY          : StringType; {'+' vid tralls utåt, '-' vid tralla inåt}
fuelstore           : StorePost;  {False då en hel ficka är tömd}
p                   : PointerType; {Ej använd pekare till inmatningsföljden}
feedlist            : VektorType; {Innehåller aktuell inmatningsföljd}

{$I comli.sys}      {filer som används vid }
{$I getbit.inc}     {kommunikation med SattCon }
{$I setbit.inc}     {via protokollet Comli }
{$I getreg.inc}
{$I setreg.inc}

```

```

{-----InitMain-----}
{* ger koordinaterna i inmatningsfickorna *}
{* ingångsvärden vid systemstart *}

```

```
PROCEDURE InitSquareCoordinates(VAR square :VektorPost);
```

```
CONST MaxElement=61;
```

```
VAR n,i,k : INTEGER;
```

```
BEGIN
```

```
FOR i:=1 TO MaxElement DO
```

```
BEGIN
```

```
square.feed[i]:='free'; {Fri att markera vid bränslematning}
```

```
square.reload[i]:='free'; {Fri att markera vid omlastning}
```

```
END;
```

```
FOR i:=35 TO 37 DO
```

```
{Motsvarar Bränsle inmatningstanken}
```

```
BEGIN
```

```
square.feed[i]:='dontexist';
```

```
square.reload[i]:='dontexist';
```

```
END;
```

```
FOR i:=43 TO 45 DO
```

```
BEGIN
```

```
square.feed[i]:='dontexist';
```

```
square.reload[i]:='dontexist';
```

Bilagor

```

END;
FOR i:=19 TO 21 DO
    square.reload[i]:='blocked';           {Blockerade vid omlastning}

i:=1;                                     {Nedan följer att varje koordinat }
n:=0;                                     {tilldelas en X- och en Y -koordinat}
WHILE i<10 DO                             {Ändläget (Utgångsläget) har koord. }
    BEGIN                                  {X=1 och Y=1 }
        square.X[i]:=1;
        square.X[i+1]:=2;                 {ficka C}
        square.Y[i]:=5-n;
        square.Y[i+1]:=5-n;
        i:=i+2;
        n:=n+1;
    END;

i:=11;
n:=0;
WHILE i<50 DO                             {ficka B}
    BEGIN
        k:=0;
        WHILE k<8 DO
            BEGIN
                square.X[i]:=k+3;
                square.Y[i]:=5-n;
                k:=k+1;
                i:=i+1;
            END;
            n:=n+1;
        END;
    END;

n:=0;
i:=51;
WHILE i<60 DO                             {ficka A}
    BEGIN
        square.X[i]:=11;
        square.X[i+1]:=12;
        square.Y[i]:=5-n;
        square.Y[i+1]:=5-n;
        i:=i+2;
        n:=n+1;
    END;

n:=0;
i:=61;
WHILE i<70 DO                             {ficka D}
    BEGIN
        square.X[i]:=13;
        square.X[i+1]:=14;
        square.Y[i]:=5-n;
        square.Y[i+1]:=5-n;
        i:=i+2;
        n:=n+1;
    END;

{* 35-37,43+45 dont exist, 44-tankkoord *}

FOR i:=1 TO 4 DO                           {Markering av vilka koordinater som }
    square.reload[i]:='dig';               {skall omlastas vid omlastning }
FOR i:=14 TO 18 DO
    square.reload[i]:='dig';
FOR i:=22 TO 26 DO
    square.reload[i]:='dig';
FOR i:=51 TO 54 DO
    square.reload[i]:='dig';
END;

{-----GetBit With 30sFailureTest-----}
{* När man väntar på att en statusbit *}
{* skall ettställas tilldelas error true*}
{* om inte detta skett inom 30 sek *}

PROCEDURE GetBit_With_30sFailureTest(comligate,Sattadress: INTEGER;
VAR value :INTEGER;
VAR error,firsttimetest : BOOLEAN);

```

Bilagor

```
VAR errorvalue : INTEGER;
BEGIN
  GetBit(1,1105,level);           {Sätts till 1 av SattCon om oinitiering}
  IF level=1 THEN                 {av travers o tralla skall ske}
    error:=TRUE;

  IF error=FALSE THEN
  BEGIN
    IF firsttimetest=TRUE THEN
    BEGIN
      SetBit(comligate,1110,On);   {SattCon tilldelar errorvalue=TRUE }
      firsttimetest:=FALSE;       {om minnesbit 1110 har varit ETT i }
      END;                         {mer än 30 sekunder}

      GetBit(comligate,1111,errorvalue);
      IF errorvalue=0 THEN
      BEGIN
        GetBit(comligate,Sattadress,value);
      END
      ELSE
      BEGIN
        error:=TRUE;
        SetBit(comligate,1111,0);
        SetBit(comligate,1105,On);   {Travers till utgångsläge}

        SetBit(comligate,301,Off);   {Alla utsignaler nollställs}
        SetBit(comligate,302,Off);
        SetBit(comligate,303,Off);
        SetBit(comligate,304,Off);
        SetBit(comligate,305,Off);
        SetBit(comligate,306,Off);
        SetBit(comligate,307,Off);
        SetBit(comligate,310,Off);
        SetBit(comligate,311,Off);

        SetBit(comligate,1110,Off);   {nollställning av minnesbit}
        firsttimetest:=TRUE;
      END;
      IF value=1 THEN
      BEGIN
        SetBit(comligate,1110,Off);   {läsning OK}
        firsttimetest:=TRUE;
      END;
    END;
  END;
END;
```

```
{-----Check Level Guard Tank-----}
{* Ger signal om bränsleinmatning till pannan *}
{* när lågbränslenivåsignalen legat hög ett *}
{* inställbart antal sekunder *}
{* }
```

```
PROCEDURE Check_Level_Guard_Tank(VAR feed,error : BOOLEAN);
```

```
VAR level,rakna : INTEGER;
```

```
BEGIN
  IF error=FALSE THEN
  BEGIN
    GetBit(comligate,626,level);   {SattCon reglerad signal för}
    {låg bränslenivå}

    IF level=1 THEN
    BEGIN
      feed:=TRUE;
    END
    ELSE
      feed:=FALSE;
    END;
  END;
END;
```

```
{-----Check Level Guard Store-----}
{* Detekterar om aktuell koordinat är tom. *}
{* Om så är fallet markeras koordinaten och *}
{* används ej fler gånger tills dess att den*}
```

Bilagor

```

        (* återinitieras via Operatörsgränssnittet *)

PROCEDURE Check_Level_Guard_Store( x :INTEGER; VAR square :VektorPost);

VAR level1,level2 : INTEGER;

BEGIN
  GetBit(1,skopanedrelage2,level2);      {Använder sig av den givare som först}
  IF (level2=1) THEN                      {markerar tom koordinat}
    square.feed[x]:='empty';
  GetBit(1,skopanedrelage1,level1);
  IF ((level1=1) OR (level2=1)) THEN
    square.reload[x]:='empty';
END;

        {-----Check Store Empty-----}
        (* Kontrollerar om hela bränslefickan är tom *)

PROCEDURE Check_Store_Empty(exe :CharType; fuelstore :StorePost;
        VAR storeempty :BOOLEAN);

VAR check : BOOLEAN;

BEGIN
  IF exe='C' THEN
    check:=fuelstore.C;

  IF exe='B' THEN
    check:=fuelstore.B;

  IF exe='A' THEN
    check:=fuelstore.A;

  IF check=TRUE THEN
    storeempty:=FALSE
  ELSE
    storeempty:=TRUE;
END;

        {-----Open Scoop-----}
        (* öppnar skopan *)

PROCEDURE Open_Scoop;

BEGIN
  GetBit(1,1120,level);                  {Passiv signal för öppen\stängd skopa}
  IF level=0 THEN                        {Om skopan är stängd}
    BEGIN
      SetBit(1,oppnaskopa,On);
      delay(4000);
      SetBit(1,oppnaskopa,Off);
      SetBit(1,1120,On);                  {Skopan är i öppet läge}
    END;
  END;

        {-----Close Scoop-----}
        (* Stänger skopan *)

PROCEDURE Close_Scoop;

BEGIN
  GetBit(1,1120,level);
  IF level=1 THEN                        {Om skopan är i öppet läge}
    BEGIN
      SetBit(1,stangskopa,On);
      delay(2500);
      SetBit(1,1120,Off);                {skopan i stängt läge}
    END;
  END;

        {-----Wire Down-----}
        (* Sänker skopan till bränslenivån *)

PROCEDURE Wire_Down(restingmatterhorn,error :BOOLEAN);

```


Bilagor

```
VAR level: INTEGER;

BEGIN
  IF error=FALSE THEN
    BEGIN
      GetBit(1,nollasttravers,level);      {1 om skopan är nerhissad}
      IF (level=0) THEN
        BEGIN
          SetBit(1,sankned,On);
          delay(1000);
          SetBit(1,sanksnabbt,On);
          REPEAT
            GetBit_With_30sFailureTest(comligate,nollasttravers,level,
                                      error,firsttimetest);
          UNTIL ((level=1) OR (error=TRUE));
          IF restingmatterhorn=TRUE THEN   {fördröjning för att slaka wiren ngt}
            delay(200);
          SetBit(1,sanksnabbt,Off);
          delay(600);
          SetBit(1,sankned,Off);
        END;
      END;
    END;
  END;

  {-----Wire_Up-----}
  { * Höjer skopan till sitt övre läge. * }
  { * Två läge finns, ett ngt högre läge * }
  { * som används vid förflyttning mellan* }
  { * fickorna och vid omlastning      * }

PROCEDURE Wire_Up(VAR resting,error :BOOLEAN);
VAR level,skopaovrelage : INTEGER;

BEGIN
  IF error=FALSE THEN
    BEGIN
      level:=0;
      GetBit(1,1121,level);      {level=1: om inmatning från Matterhorn}
      IF level=1 THEN
        skopaovrelage:=skopaovrelage2
      ELSE
        skopaovrelage:=skopaovrelage1;

      GetBit(1,skopaovrelage,level);
      IF (level=0) THEN
        BEGIN
          SetBit(1,lyftupp,On);
          delay(1000);
          SetBit(1,lyftsnabbt,On);

          i:=0;
          REPEAT
            IF i<16 THEN
              BEGIN
                delay(400);
                i:=i+1;
              END;
            IF i=16 THEN
              SetBit(1,stangskopa,Off);

              GetBit_With_30sFailureTest(comligate,skopaovrelage,level,
                                        error,firsttimetest);
            UNTIL ((level=1) OR (error=TRUE));

            SetBit(1,stangskopa,Off);
            SetBit(1,lyftsnabbt,Off);
            delay(100);
            SetBit(1,lyftupp,Off);
            resting:=FALSE;
          END;
        END;
      END;
    END;
  END;
END;
```

Bilagor

```
{-----Start Traverse-----}
{* Startar traversen i viss riktning. *}
{* Om avståndet är större än en koord*}
{* går traversen upp i högfart      *}

PROCEDURE Start_Traverse(range:INTEGER; directionX:StringType;
                        VAR error :BOOLEAN);
BEGIN
  IF error=FALSE THEN
    BEGIN
      IF directionX='+' THEN
        SetBit(1,traversplus,On)
      ELSE
        SetBit(1,traversminus,On);
      delay(1000);
      IF range>1 THEN
        SetBit(1,travershogfart,On);
    END;
  END;
END;

{-----Stop Traverse-----}
{* Stannar traversen när mittpkt. är nådd *}

PROCEDURE Stop_Traverse(directionX:StringType; VAR error :BOOLEAN);
VAR halt : INTEGER;

BEGIN
  SetBit(1,travershogfart,Off);
  IF error=FALSE THEN
    BEGIN
      REPEAT
        GetBit_With_30sFailureTest(comligate,stoppgivaretravers,level,
                                error,firstttimetest);

        UNTIL ((level=1) OR (error=TRUE));
      END;
      {Level:=1 då man nått fram till koordinaten}

      SetBit(1,stoppgivaretravers,Off);
      IF directionX='+' THEN
        SetBit(1,traversplus,Off)
      ELSE
        SetBit(1,traversminus,Off);
    END;
  END;
END;

{-----Start Truck-----}
{* Startar trallan i önskad riktning *}

PROCEDURE Start_Truck(directionY :StringType; VAR error :BOOLEAN);

BEGIN
  IF error=FALSE THEN
    BEGIN
      IF directionY='+' THEN
        SetBit(1,trallaplus,On)
      ELSE
        SetBit(1,trallaminus,On);
      delay(600);
      SetBit(1,trallahogfart,On);
    END;
  END;
END;

{-----Stop Truck-----}
{* Stannar tralla när mittpkt. är nådd *}

PROCEDURE Stop_Truck(directionY : StringType; VAR error,feedtank :BOOLEAN);
VAR halt : INTEGER;

BEGIN
  SetBit(1,trallahogfart,Off);
  IF error=FALSE THEN
    BEGIN
      IF feedtank=TRUE THEN
        BEGIN
```

Bilagor

```

    delay(3000);
    SetBit(1, oppnaskopa, On);           {Öppnar skopan}
    SetBit(1, 1120, On);                 {Talar om att skopan är öppnad}
END;
REPEAT
    GetBit_With_30sFailureTest(comligate, stoppgivaretralla, level,
                               error, firsttimetest);
UNTIL ((level=1) OR (error=TRUE));
END;

IF directionY='+' THEN
    SetBit(1, trallaplus, Off)
ELSE
BEGIN
    SetBit(1, trallaminus, Off);
    IF feedtank=TRUE THEN
        BEGIN
            SetBit(1, oppnaskopa, Off);
        END;
    END;
END;
END;

    {-----Move Truck-----}
    { * Förflyttar trallan till önskad Y-koordinat * }

PROCEDURE Move_Truck(i : INTEGER; square : VektorPost; VAR presentY : INTEGER;
                   VAR error : BOOLEAN);

VAR    newY, range, adres, arrivingsquare, crossing : INTEGER;
       directionY : STRING[1];

BEGIN
    IF error=FALSE THEN
        BEGIN
            newY:=square.Y[i];           {Koordinatens Y-position dit vi önskar}
            IF (presentY <> newY) THEN
                BEGIN
                    IF (presentY > newY) THEN           {val av riktning}
                        BEGIN
                            range:=presentY-newY;     {antalet koord i Y-led mellan nuvarande}
                            directionY:='-';           {koord. och önskad}
                            adres:=trallaleft;
                        END
                    ELSE
                        BEGIN
                            range:=newY-presentY;
                            directionY:='+';
                            adres:=trallaright;
                        END;
                    END;

                    Wire_Up(resting, error);
                    Start_Truck(directionY, error);
                    WHILE (range > 0) DO
                        BEGIN
                            REPEAT
                                GetBit_With_30sFailureTest(comligate, adres, level,
                                                            error, firsttimetest);
                            UNTIL ((level=1) OR (error=TRUE));

                            range:=range-1;
                            IF (range>0) THEN
                                BEGIN
                                    REPEAT
                                        GetBit_With_30sFailureTest(comligate, stoppgivaretralla, level,
                                                                    error, firsttimetest);
                                    UNTIL ((level=1) OR (error=TRUE));
                                END;
                            END;
                            Stop_Truck(directionY, error, feedtank);
                        END;
                    presentY:=newY;
                END;
            END;
        END;
    END;
END;

```

Bilagor

```

    {-----Move Travers-----}
    { * Förflyttning av Travers till önskad      * }
    { * X-koordinat. Förflyttningen utförs av SattCon * }

PROCEDURE Move_Travers(i:INTEGER; square:VektorPost;VAR presentX : INTEGER;
                      VAR error :BOOLEAN );

VAR   newX,range,adres,adres2,arrivingsquare,crossing,compleated: INTEGER;
      directionX                               : STRING[1];
      errorbit                                 : INTEGER;

BEGIN
  IF error=FALSE THEN
    BEGIN
      SetBit(1,1114,0);                               {nollställer initialv. i SattCon}

      newX:=square.X[i];
      GetReg(1,1,300);
      presentX:=TRUNC(RealValue[1]);

      IF (presentX <> newX) THEN
        BEGIN
          Wire_Up(resting,error);
          RealValue[1]:=newX;
          SetReg(1,1,302);
          compleated:=0;
          SetBit(1,1117,0);
          REPEAT                                       {SattCon sätter compleate till ett}
            GetBit(1,1117,compleated);               {när förflyttningen är utförd }
          UNTIL compleated=1;

          GetBit(1,1105,errorbit);
          IF errorbit=1 THEN
            error:=TRUE;
          END;
        END;
      END;
    END;
  END;

    {-----Fill Scoop-----}
    { * Fyller skopan      * }

PROCEDURE Fill_Scoop;

VAR   zeroload : INTEGER;

BEGIN
  GetBit(1,nollasttravers,zeroload);   {zeroload=1 när skopan är i kontakt}
  IF (zeroload=0) THEN                 {med bränslet}
    Wire_Down(matterhorn,error);
  Close_Scoop;
END;

    {-----Reload Scoop-----}
    { * omlastar skopa vid omlastning      * }

PROCEDURE Reload_Scoop(k : INTEGER);

VAR   newk : INTEGER;

BEGIN
  Wire_Up(resting,error);
  IF ((k>=51) AND (k<=60)) OR ((k>=1) AND (k<=10)) THEN
    newk:=k+6                           {Omlastning av A eller C fickan}
  ELSE
    newk:=k+24;                          {Omlastning av ficka B}
  Move_Truck(newk,square,presentY,error);
  Open_Scoop;
END;
```

Bilagor

```
{-----Feed Tank-----}
{* Förflyttar Travers och Tralla från aktuell *}
{* koordinat till bränsleinmatningsfickan där *}
{* skopan töms *}

PROCEDURE Feed_Tank(VAR error,feedtank : BOOLEAN);
CONST tankcoordinate=44;

VAR level,time,traversecoord : INTEGER;

BEGIN
  Wire_Up(resting,error);
  Move_Travers(tankcoordinate,square,presentX,error);

  feedtank:=TRUE;      {Används för att stoppa trallan halv koord tidigare}
                      {vid inmatning till bränslebehållaren}
  Move_Truck(tankcoordinate,square,presentY,error);
  feedtank:=FALSE;

  GetReg(1,1,125);     {SattCons kontrollregister för traverskoordinat}
  traversecoord:=TRUNC(RealValue[1]);
  Open_Scoop;
  SetBit(1,626,0);     {Nollställer flaggan Traversmata i SattCon}
END;

{-----Feed Matterhorn-----}
{* Förflyttar Travers och Tralla från aktuell *}
{* koordinat till matterhorntoppen där skopan töms *}

PROCEDURE Feed_Matterhorn;
CONST matterhorncoordinate=20;

BEGIN
  Wire_Up(resting,error);
  Move_Travers(matterhorncoordinate,square,presentX,error);
  Move_Truck(matterhorncoordinate,square,presentY,error);
  Open_Scoop;
END;

{-----Check Reloading----- }
{* Kontrollerar via SattCon om ngn ficka *}
{* skall omlastas *}

PROCEDURE Check_Reloading(VAR reload : BOOLEAN; VAR reloadstore :CharVektor);
VAR reloadlevel : INTEGER;

BEGIN
  GetBit(1,1103,reloadlevel);      {reloadlevel=0 ; omlastning FRÅN}
  IF reloadlevel=1 THEN
    BEGIN
      GetReg(1,1,220);

      CASE TRUNC(RealValue[1]) OF
        0 : reloadstore:='C';
        1 : reloadstore:='B';
        2 : reloadstore:='A';
      END;
      reload:=TRUE
    END
  ELSE
    reload:=FALSE;
  END;
END;

{-----Start Reloading-----}
{* Omlastar önskad ficka från portarna *}
{* till inner väggen *}

PROCEDURE Start_Reloading(reloadstore : charvektor;
  VAR oldreloadstore :charvektor;
  VAR reload,firsttimereloading : BOOLEAN;
  VAR square : VektorPost; VAR oldnumber :INTEGER;
  VAR totalnumber : INTEGER);

VAR k,jump : INTEGER;
```

Bilagor

passed : BOOLEAN;

BEGIN

IF (oldreloadstore<>reloadstore) THEN

BEGIN

firsttimereloading:=TRUE;

END;

IF firsttimereloading=TRUE THEN

BEGIN

CASE reloadstore OF

'C' : BEGIN

oldnumber:=0;

totalnumber:=4;

END;

'B' : BEGIN

oldnumber:=13;

totalnumber:=10;

END;

'A' : BEGIN

oldnumber:=50;

totalnumber:=4;

END;

END;

firsttimereloading:=FALSE;

END;

jump:=0;

{räknas upp när square.reload<>'dig'}

passed:=FALSE;

{False tills square.reload='dig'}

k:=oldnumber+1;

IF k=5 THEN

{Börja om i ficka C}

k:=1;

IF k=27 THEN

{Börja om i ficka B}

k:=14;

IF k=55 THEN

{Börja om i ficka A}

k:=51;

REPEAT

IF square.reload[k]='dig' THEN

BEGIN

Wire_Up(resting,error);

Move_Truck(k,square,presentY,error);

Move_Travers(k,square,presentX,error);

Fill_Scoop;

Check_Level_Guard_Store(k,square);

Reload_Scoop(k);

passed:=TRUE;

jump:=0;

END

ELSE

BEGIN

jump:=jump+1;

k:=k+1;

IF k=5 THEN

k:=1;

IF k=27 THEN

k:=14;

IF k=55 THEN

k:=51;

IF (jump>totalnumber) THEN

{Om alla koordinater är tömda}

BEGIN

reload:=FALSE;

SetBit(comligate,1103,Off);

{Ställer SattCon flagga}

firsttimereloading:=TRUE;

FOR i:=1 TO 4 DO

square.reload[i]='dig';

FOR i:=14 TO 18 DO

square.reload[i]='dig';

FOR i:=22 TO 26 DO

square.reload[i]='dig';

FOR i:=51 TO 54 DO

square.reload[i]='dig';

END;

END;

UNTIL ((reload=FALSE) OR (passed=TRUE));

Bilagor

```
oldnumber:=k;
oldreloadstore:=reloadstore;
END;
```

```
{-----Start Reloading Dstore-----}
{* Omlastar D fickan till A fickan *}
```

```
PROCEDURE Start_Reloading_Dstore(VAR square :VektorPost;
VAR oldDreload :INTEGER);
VAR k,newk : INTEGER;
```

```
BEGIN
```

```
k:=oldDreload;
```

```
IF k>70 THEN
```

```
k:=61;
```

```
Wire_Up(resting,error);
```

```
Move_Truck(k,square,presentY,error);
```

```
Move_Travers(k,square,presentX,error);
```

```
Fill_Scoop;
```

```
Wire_Up(resting,error);
```

```
newk:=k-10;
```

```
Move_Travers(newk,square,presentX,error);
```

```
Open_Scoop;
```

```
oldDreload:=k+1;
```

```
SetBit(1,1111,0); {Bit 1111 är ett om omlastning skall utföras}
```

```
END;
```

```
{-----Get Coord Status-----}
{* Uppdaterar varje koordinats status. *}
{* Tre olika status : Gräv, Fri(=fri att*)
{* gräva om förmarkerade grävkoord. är *}
{* tomma), Stop(får ej grävas) *}
```

```
PROCEDURE Get_Coord_Status(VAR square : VektorPost; VAR fuelstore :StorePost);
```

```
VAR value,level,Sattaddr : INTEGER;
```

```
BEGIN
```

```
GetBit(1,1100,level);
```

```
{ett om förändring av koord. status}
```

```
IF level=1 THEN
```

```
BEGIN
```

```
Sattaddr:=130;
```

```
{Startregister i SattCon för koord status}
```

```
FOR i:=1 TO 34 DO
```

```
BEGIN
```

```
GetReg(1,1,Sattaddr);
```

```
value:=TRUNC(RealValue[1]);
```

```
IF value=0 THEN
```

```
square.feed[i]='free';
```

```
IF value=1 THEN
```

```
square.feed[i]='dig';
```

```
IF value=2 THEN
```

```
square.feed[i]='stop';
```

```
Sattaddr:=Sattaddr+1;
```

```
END;
```

```
FOR i:=38 TO 42 DO
```

```
BEGIN
```

```
GetReg(1,1,Sattaddr);
```

```
value:=TRUNC(RealValue[1]);
```

```
IF value=0 THEN
```

```
square.feed[i]='free';
```

```
IF value=1 THEN
```

```
square.feed[i]='dig';
```

```
IF value=2 THEN
```

```
square.feed[i]='stop';
```

```
Sattaddr:=Sattaddr+1;
```

```
END;
```

Bilagor

```
FOR i:=46 TO 60 DO
BEGIN
  GetReg(1,1,Sattaddr);
  value:=TRUNC(RealValue[1]);

  IF value=0 THEN
    square.feed[i]:='free';
  IF value=1 THEN
    square.feed[i]:='dig';
  IF value=2 THEN
    square.feed[i]:='stop';

  Sattaddr:=Sattaddr+1;
END;

fuelstore.C:=TRUE;           {Tilldelas False då hela fickan är tömd}
fuelstore.B:=TRUE;
fuelstore.A:=TRUE;

SetBit(1,1100,0);
END;

END;
```

```
{-----Get Feeding Stores-----}
{* Hämtar in i vilken turordning man skall   *}
{* gräva i fickorna Ex : ABBAC, första i A   *}
{* andra och tredje i B, fjärde i A, femte i C *}
```

```
PROCEDURE Get_Feeding_Stores(VAR feedlist : VektorType; VAR number :INTEGER);
```

```
VAR Sattaddr,i,level : INTEGER;
```

```
BEGIN
  GetBit(1,1101,level);           {Anger om nya värden lästs ner}

  IF level=1 THEN
  BEGIN
    i:=0;
    Sattaddr:=200;                 {Startadress SattCon register}
    REPEAT
      i:=i+1;
      GetReg(1,1,Sattaddr);

      CASE TRUNC(RealValue[1]) OF
        0 : feedlist[i]:='C';
        1 : feedlist[i]:='B';
        2 : feedlist[i]:='A';
        9 : feedlist[i]:=' ';
      END;

      Sattaddr:=Sattaddr+1;
    UNTIL ((feedlist[i]=' ') OR (i=12));
    IF i=12 THEN
      feedlist[12]:=' ';           {Max elva fickor kan anges}
      SetBit(1,1101,0);

      number:=1;
    END;
  END;
```

```
{-----Select ABC-----}
{* Ger vilket förråd som står i tur för inmatning*}
```

```
PROCEDURE Select_ABC(VAR exe:CharType; VAR number:INTEGER;
  VAR resting :BOOLEAN; VAR feedlist :VektorType);
```

```
BEGIN
  Get_Feeding_Stores(feedlist,number);
  Get_Coord_Status(square,fuelstore);

  IF resting=FALSE THEN
  BEGIN
    REPEAT
```


Bilagor

```
WRITELN(number);
exe:=feedlist[number];
IF exe=' ' THEN                                {sista elementet i listan är ett blankt.}
BEGIN
  number:=1;
  exe:=feedlist[number];
END;
number:=number+1;
Check_Store_Empty(exe,fuelstore,storeempty);
UNTIL ((storeempty=FALSE) OR (exe=' '));
END;

END;

{----Goto Next Feeding Coordinate----}
{* När inga fler instruktioner skall utföras *}
{* används denna procedur för att lägga skopan *}
{* i den koordinat som står på tur vid ny instr*}
{* om inmatning, alltså ett viloläge för skopan *}

PROCEDURE Goto_Next_Feeding_Coordinate(presentX :INTEGER;
                                       VAR restingmatterhorn,resting :BOOLEAN;
                                       VAR oldA,oldB,oldC,oldM,number :INTEGER );

VAR restingcoord,passed : INTEGER;

BEGIN
  restingmatterhorn:=FALSE;                    {Anger om skopans vilar i Matterhorn}

  GetBit(comligate,1102,matterhornlevel);
  IF matterhornlevel=1 THEN
    matterhorn:=TRUE
  ELSE
    matterhorn:=FALSE;

  IF matterhorn=TRUE THEN
  BEGIN
    restingcoord:=oldM;                        {oldM= den koord som står på tur vid inmatning}
    restingmatterhorn:=TRUE;
  END
  ELSE
  BEGIN
    Select_ABC(exe,number,resting,feedlist);
    number:=number-1;                          {exe koord skall ej räknas med}

    passed:=0;
    IF (exe='A') THEN
    BEGIN
      oldA:=oldA-1;
      REPEAT
        passed:=passed+1;
        oldA:=oldA+1;
        IF oldA=11 THEN
          oldA:=1;
        UNTIL ((square.feed[oldA]='dig') OR (passed=10));
        restingcoord:=oldA;
      END;

    IF (exe='B') THEN
    BEGIN
      oldB:=oldB-1;
      REPEAT
        passed:=passed+1;
        oldB:=oldB+1;
        IF oldB=51 THEN
          oldB:=11;
        UNTIL ((square.feed[oldB]='dig') OR (passed=40));
        restingcoord:=oldB;
      END;

    IF (exe='C') THEN
    BEGIN
      oldC:=oldC-1;
      REPEAT
        passed:=passed+1;
```

Bilagor

```
    oldC:=oldC+1;
    IF oldC=61 THEN
        oldC:=51;
    UNTIL ((square.feed[oldC]='dig') OR (passed=10));
    restingcoord:=oldC;
END;

IF (exe=' ') THEN
BEGIN
    restingcoord:=15;
    exe='B';
END;

END;

Move_Truck(restingcoord,square,presentY,error);
Move_Travers(restingcoord,square,presentX,error);
Wire_Down(restingmatterhorn,error);
resting:=TRUE;

END;

        {-----Start Feeding From A-----}
        { * Används om inmatning sker direkt ur A till * }
        { * bränslebehållaren eller från A till Matter- * }
        { * toppen.Räknar ut ur vilken koordinat som * }
        { * skall grävas. Om alla markerade koord. är * }
        { * tomma räknas nya koordinater fram. * }

PROCEDURE Start_Feeding_From_A( VAR exe :CharType; VAR square :VektorPost;
    VAR fuelstore :StorePost;
    VAR firsttimeA,matterhorn:BOOLEAN;
    VAR oldA,fuelcontrol : INTEGER);

VAR    a,k,jump      : INTEGER;
    passedA,feeded   : BOOLEAN;

BEGIN
    feeded:=FALSE;
    jump:=0;
    IF (firsttimeA=TRUE) THEN
    BEGIN
        a:=51;
        firsttimeA:=FALSE;
    END
    ELSE
        a:=oldA;
    REPEAT
        IF (square.feed[a]='dig') THEN
        BEGIN
            Move_Truck(a,square,presentY,error);
            Move_Travers(a,square,presentX,error);
            Wire_Down(matterhorn,error);
            Fill_Scoop;
            Check_Level_Guard_Store(a,square);
            IF ((matterhorn=TRUE) AND (fuelcontrol=0)) THEN    {fuelcontrol=1 om man behöver }
                Feed_Matterhorn                                {bättre bränsle}
            ELSE
                Feed_Tank(error,feedtank);
            a:=a+1;
            IF ( a>60 ) THEN
                a:=51;
            jump:=0;
            feeded:=TRUE;
        END
        ELSE
        BEGIN
            jump:=jump+1;
            a:=a+1;
            IF (a>60) THEN
            BEGIN
                a:=51;
            END;
            IF (jump=10) THEN                                {om alla koord. är tomma räknas nya fram}

```

Bilagor

```
BEGIN
  jump:=0;
  passedA:=FALSE;
  k:=51;
  REPEAT
    FOR i:=k TO (k+1) DO
      BEGIN
        IF square.feed[i]='free' THEN
          BEGIN
            square.feed[i]='dig';
            passedA:=TRUE;
          END;
        END;
        IF ((k<59) AND (passedA=FALSE)) THEN
          k:=k+2;
        UNTIL ((passedA=TRUE) OR (i=60));
        IF (passedA=FALSE) THEN
          BEGIN
            fuelstore.A:=FALSE;
          END;
          a:=k;
        END;
      END;
    UNTIL ((feeded=TRUE) OR (fuelstore.A=FALSE));

  oldA:=a;
END;

{-----Start Feeding From B-----}
{* enligt förregående procedur, men för fickaB *}

PROCEDURE Start_Feeding_From_B( VAR exe :CharType;VAR square :VektorPost;
                                VAR fuelstore :StorePost;
                                VAR firsttimeB,matterhorn :BOOLEAN;
                                VAR oldB : INTEGER);

VAR  b,k,jump          : INTEGER;
     passedB,feeded   : BOOLEAN;

BEGIN
  feeded:=FALSE;
  jump:=0;
  IF (firsttimeB=TRUE) THEN
    BEGIN
      b:=11;
      firsttimeB:=FALSE;
    END
  ELSE
    b:=oldB;
  REPEAT
    IF (square.feed[b]='dig') THEN
      BEGIN
        Move_Truck(b,square,presentY,error);
        Move_Travers(b,square,presentX,error);
        Wire_Down(matterhorn,error);
        Fill_Scoop;
        Check_Level_Guard_Store(b,square);
        IF matterhorn=TRUE THEN
          Feed_Matterhorn
        ELSE
          Feed_Tank(error,feedtank);
        b:=b+1;
        IF ( b>50 ) THEN
          b:=11;
          jump:=0;
          feeded:=TRUE;
        END
      ELSE
        BEGIN
          jump:=jump+1;
          b:=b+1;
          IF (b>50) THEN
            b:=11;
          END
        END
      END
    END
  END
```

Bilagor

```

IF (jump=40) THEN
  BEGIN
    jump:=0;
    passedB:=FALSE;
    k:=11;
    REPEAT
      FOR i:=k TO (k+7) DO
        BEGIN
          IF square.feed[i]='free' THEN
            BEGIN
              square.feed[i]:='dig';
              passedB:=TRUE;
            END;
          END;
          IF ((k<43) AND (passedB=FALSE)) THEN
            k:=k+8;
          UNTIL ((passedB=TRUE) OR (i=50));
          IF (passedB=FALSE) THEN
            fuelstore.B:=FALSE;
            b:=k;
          END;
        END;
      UNTIL ((feeded=TRUE) OR (fuelstore.B=FALSE));

      oldB:=b;
    END;

    {-----Start Feeding From C-----}
    {* enligt förregående procedur, men för fickaC *}

PROCEDURE Start_Feeding_From_C( VAR exe :CharType; VAR square :VektorPost;
                                VAR fuelstore :StorePost;
                                VAR firsttimeC,matterhorn :BOOLEAN;
                                VAR oldC :INTEGER);

VAR   c,k,jump      : INTEGER;
      passedC,feeded : BOOLEAN;

BEGIN
  feeded:=FALSE;
  jump:=0;
  IF (firsttimeC=TRUE) THEN
    BEGIN
      c:=1;
      firsttimeC:=FALSE;
    END
  ELSE
    c:=oldC;
  REPEAT
    IF (square.feed[c]='dig') THEN
      BEGIN
        Move_Truck(c,square,presentY,error);
        Move_Travers(c,square,presentX,error);
        Wire_Down(matterhorn,error);
        Fill_Scoop;
        Check_Level_Guard_Store(c,square);
        IF matterhorn=TRUE THEN
          Feed_Matterhorn
        ELSE
          Feed_Tank(error,feedtank);
        c:=c+1;
        IF ( c>10 ) THEN
          c:=1;
          jump:=0;
          feeded:=TRUE
        END
      ELSE
        BEGIN
          jump:=jump+1;
          c:=c+1;
          IF (c>10) THEN
            c:=1;
            IF (jump=10) THEN

```

Bilagor

```
BEGIN
  jump:=0;
  passedC:=FALSE;
  k:=1;
  REPEAT
    FOR i:=k TO (k+1) DO
      BEGIN
        IF square.feed[i]='free' THEN
          BEGIN
            square.feed[i]='dig';
            passedC:=TRUE;
          END;
        END;
        IF ((k<9) AND (passedC=FALSE)) THEN
          k:=k+2;
        UNTIL ((passedC=TRUE) OR (i=10));
        IF (passedC=FALSE) THEN
          BEGIN
            fuelstore.C:=FALSE;
          END;
          c:=k;
        END;
      END;
    UNTIL ((feeded=TRUE) OR (fuelstore.C=FALSE));

  oldC:=c;
END;

      {----Start Feeding From Matterhorn----}
      { * Inmatning av bränsle till bränslebehållaren * }
      { * från Matterhorn * }

PROCEDURE Start_Feeding_From_Matterhorn(VAR matterhorn,firsttimeM :BOOLEAN;
                                         VAR square :VektorPost;
                                         VAR oldM,mattertop :INTEGER);

VAR feeded,passedM : BOOLEAN;
    m,jump          : INTEGER;

BEGIN
  FOR i:=11 TO 13 DO {initierar Matterhorns koordinater}
    square.feed[i]:='Mdig';
  FOR i:= 19 TO 21 DO
    square.feed[i]:='Mdig';
  FOR i:= 27 TO 29 DO
    square.feed[i]:='Mdig';
  square.feed[20]:='mattertop';

  feeded:=FALSE;
  jump:=0;
  IF (firsttimeM=TRUE) THEN
    BEGIN
      m:=11;
      firsttimeM:=FALSE;
    END
  ELSE
    m:=oldM;

  REPEAT
    IF (square.feed[m]='Mdig') THEN
      BEGIN
        Move_Truck(m,square,presentY,error);
        Move_Travers(m,square,presentX,error);
        Wire_Down(matterhorn,error);
        Fill_Scoop;
        Check_Level_Guard_Store(m,square);
        SetBit(1,1121,1); {jag skall mata in från Matterhorn}
        Feed_Tank(error,feedtank);
        SetBit(1,1121,0);
        feeded:=TRUE;
        jump:=0;
        mattertop:=mattertop-1;
      END;

    m:=m+1;
    IF (m=14) OR (m=22) THEN {Grävning sker cirkulärt runt Mattertop}
```

Bilagor

```
m:=m+5;
IF (m=20) THEN
  m:=21;
IF (m>=30) THEN
  m:=11;
jump:=jump+1;
IF (jump>10) THEN
  matterhorn:=FALSE;

oldM:=m;
UNTIL ((feeded=TRUE) OR (matterhorn=FALSE));
END;

{-----Start Feeding From ABC-----}
{* Bestämmer vilken ficka som står på tur för *}
{* inmatning. Därefter sker inmatning till bränsle *}
{* behållaren eller påfyllning av Matterhorn *}

PROCEDURE Start_Feeding_From_ABC;
BEGIN
  Get_Coord_Status(square,fuelstore);
  Select_ABC(exe,number,resting,feedlist);

  IF (exe='C') THEN
    Start_Feeding_From_C(exe,square,fuelstore,firsttimeC,matterhorn,oldC);

  IF (exe='B') THEN
    Start_Feeding_From_B(exe,square,fuelstore,firsttimeB,matterhorn,oldB);

  IF (exe='A') THEN
    Start_Feeding_From_A(exe,square,fuelstore,firsttimeA,matterhorn,oldA,fuelcontrol);
END;

{-----Start Feeding-----}
{* Denna procedur bestämmer varifrån inmatning *}
{* skall ske. Tre olika möjligheter: *}
{* (1) Direkt inmatning från resp. ficka *}
{* (2) Inmatning via Matterhorn *}
{* (3) Vid reglering kan inmatning ske direkt *}
{* från C (bäst bränsle) om temp för låg *}

PROCEDURE Start_Feeding(VAR matterhorn,firsttimeM,resting,firsttimeABC :BOOLEAN );
VAR matterlevel,control : INTEGER;
BEGIN
  resting:=FALSE;

  Get_Coord_Status(square,fuelstore);

  GetBit(1,1115,control); {Reglering AV/FÅ}
  IF control=1 THEN
  BEGIN
    GetBit(1,1116,fuelcontrol); {fuelcontrol sätts till ett av }
    WRITE(' BRÄNSLEREGL '); {SattCon vid regl. till bättre bränsle}
  END
  ELSE
    fuelcontrol:=0;

  IF fuelcontrol=1 THEN
  BEGIN
    Start_Feeding_From_A(exe,square,fuelstore,firsttimeC,
                        matterhorn,oldC,fuelcontrol);

    firsttimeABC:=TRUE;
  END
  ELSE
  BEGIN
    GetBit(1,1102,matterlevel); {Ett om inmatning skall ske direkt från M.}
    IF matterlevel=1 THEN
    BEGIN
      Start_Feeding_From_Matterhorn(matterhorn,firsttimeM,
                                   square,oldM,mattertop);

      matterhorn:=TRUE;
```

Bilagor

```

firsttimeABC:=TRUE;
END
ELSE
BEGIN
  IF firsttimeABC=TRUE THEN
    SetBit(1,1100,1);           {göra ny inläsning av inmatngnscoord.}
    firsttimeM:=FALSE;
    matterhorn:=FALSE;
    Start_Feeding_From_ABC;
    firsttimeABC:=FALSE;
  END;
END;
END;

{-----Goto Fix Position-----}
{* Ett fast läge en koordinat framför *}
{* bränslebehållaren dit skopan kan styras *}
{* tex vid påfyllning av fickorna *}

PROCEDURE Goto_Fix_Position;

CONST position=28;

VAR fixposition : INTEGER;

BEGIN
  Wire_Up(resting,error);
  Move_Truck(position,square,presentY,error);
  Move_Travers(position,square,presentX,error);
  REPEAT
    delay(3000);
    GetBit(comligate,1104,fixposition);
  UNTIL (fixposition=0);
END;

{-----Refill Matterhorn-----}
{* Läger en skopa bränsle från aktuell *}
{* ficka på Mattertoppen *}

PROCEDURE Refill_Matterhorn(VAR mattertop :INTEGER; VAR resting :BOOLEAN);

BEGIN
  resting:=FALSE;
  Start_Feeding_From_ABC;
  mattertop:=mattertop+1;
END;

{-----Check Mattertop-----}
{* Kontrollerar om man önskar öka höjden på *}
{* Matterhorn *}

PROCEDURE Check_Mattertop(VAR mattertop : INTEGER);

BEGIN
  GetReg(1,1,221);
  IF TRUNC(RealValue[1])>0 THEN
    BEGIN
      mattertop:=mattertop-TRUNC(RealValue[1]);
      RealValue[1]:=0;
      SetReg(1,1,221);           {genom att minska verklig höjd med önskad}
    END;                       {ökning lurar man programmet som strävar }
  END;                           {efter att höjden skall vara 20 }
END;

{-----To Initial Position-----}
{* Initierar travers och tralla genom att förflytta *}
{* dem till ett fast läge och där tilldela program- *}
{* variablern motsvarande koordinater. Initierar även*}
{* en mängd andra för programmet nödvändiga variabler*}

PROCEDURE To_Initial_Position( VAR presentX,presentY :INTEGER;

```

Bilagor

```
VAR firsttimereloading,resting,error:BOOLEAN);

VAR traverseinitialposition,truckinitialposition : INTEGER;
    range : INTEGER;

BEGIN
  SetBit(1,301,0);           {nollställning av alla utsignaler}
  SetBit(1,302,0);
  SetBit(1,303,0);
  SetBit(1,304,0);
  SetBit(1,305,0);
  SetBit(1,306,0);
  SetBit(1,307,0);
  SetBit(1,310,0);
  SetBit(1,311,0);
  SetBit(1,312,0);
  SetBit(1,313,0);
  RealValue[1]:=0;
  SetReg(1,1,302);          {newX till SattCon}

  REPEAT                    {ligger och väntar på omslag Auto till}
    GetBit(1,516,auto);     {Manuel. Omslag sker via yttre strömmställare }
    delay(1000);
  UNTIL auto=1;

  error:=FALSE;
  SetBit(1,1105,0);         {errorbit}
  SetBit(1,1120,0);         {Skopan är i stängt läge}
  level:=0;
  GetBit(1,skopaovrelagel,level);
  IF (level=0) THEN
  BEGIN
    SetBit(1,lyftupp,On);
    delay(1000);
    SetBit(1,lyftsnabbt,On);
    REPEAT
      GetBit(1,comligate,skopaovrelagel,level);
    UNTIL (level=1) ;
    SetBit(1,lyftsnabbt,Off);
    delay(100);
    SetBit(1,lyftupp,Off);
  END;

  range:=5; {Hjälpsats}
  GetBit(1,traversutg,traverseinitialposition);
  IF (traverseinitialposition=0) THEN
  BEGIN
    Start_Traverse(range,'-',error);

    REPEAT
      GetBit(1,traversutg,traverseinitialposition);
    UNTIL (traverseinitialposition=1);
    error:=FALSE;
    Stop_Traverse('-',error);
  END;
  GetBit(1,trallautg,truckinitialposition);

  IF (truckinitialposition=0) THEN
  BEGIN
    error:=FALSE;
    Start_Truck('-',error);
    REPEAT
      GetBit(1,trallautg,truckinitialposition);
    UNTIL (truckinitialposition=1);
    Stop_Truck('-',error,feedtank);
  END;
  Open_Scoop;

  presentX:=1;
  presentY:=1;
  RealValue[1]:=0;
  SetReg(1,1,125);         {nollställ traverslägeskontrollbiten}
  RealValue[1]:=1;
  SetReg(1,1,300);         {nollställ koordinaträkningen}

  SetBit(1,1100,1);
```


Bilagor

```
SetBit(1,1101,1);
SetBit(1,1110,0); {nollställ minnesbit}
SetBit(1,1111,0); {nollställ errorvärde}
SetBit(1,626,0); {Bränsletidsgivaren}

SetBit(1,1112,0);
SetBit(1,1113,0);
SetBit(1,1114,0);
SetBit(1,1117,0); {Klar signal vid traversförflyttning}
firsttimereloading:=TRUE;
resting:=FALSE;
error:=FALSE;
END;

{-----main program-----}
BEGIN
WRITELN('START AV TRAVERSPROGRAM');
CheckBreak:=TRUE; {aktiverar möjlighet till avbr }
delay(5000);
WRITELN('halt');

InitProgVar;
Init1(12,5,1000,BINcomm,FastPC);
setbit(1,1105,0);
setbit(1,1103,0);
SetBit(1,1106,0); {tidsstyrd traversutg signal}
SetBit(1,1107,0); {tidsstyrd trallautg signal}
SetBit(1,1115,0); {Reglering AV/PÅ}
SetBit(1,1116,0); {SattCon reglering till bättre bränsle}
SetBit(1,1121,0); {Höjdreglering}

number:=1;
InitSquareCoordinates(square);
To_Initial_position(presentX,presentY,firsttimereloading,resting,error);
cancel:=FALSE;

reload:=FALSE;
reloadstore='A';
firsttimereloading:=TRUE;
oldnumber:=0;
oldDreload:=61;

firsttimeA:=TRUE; {Första gången en viss procedur eller }
firsttimeB:=TRUE; {händelse utförs }
firsttimeC:=TRUE;
firsttimeM:=TRUE;
firsttimeABC:=TRUE;

resting:=FALSE;
fixposition:=0;
mattertop:=20; {programet strävar efter att }
{Matterhorn skall vara 20 skopor högt}

storeempty:=FALSE;
feedtank:=FALSE; {Används vid inmatn till bränslebeh.}
{för att stoppa tralla halv koord tidigare}

fuelstore.A:=true;
fuelstore.B:=true;
fuelstore.C:=true;

oldA:=51; {första koordinaten i A}
oldB:=14; {första koordinaten i B}
oldC:=1; {första koordinaten i C}
oldM:=11; {första koordinaten i Matterhorn}

FOR i:=11 TO 13 DO
square.feed[i]:='Mdig';
FOR i:= 19 TO 21 DO
square.feed[i]:='Mdig';
```

Bilagor

```
FOR i:= 27 TO 29 DO
  square.feed[i]:='Mdig';
square.feed[20]:='mattertop';

      {huvudprogrammet är uppbyggt så att de      }
      {viktigaste procedurerna prioriteras högst}

REPEAT
BEGIN
  WRITE('@');
                                {Aktiverar CrtBreak}

  GetBit(1,1105,level);
  IF level=1 THEN
  BEGIN
    To_Initial_Position(presentX,presentY,firsttimereloading,resting,error);
    SetBit(1,1105,0);
  END;

  GetBit(comligate,1104,fixposition);
  IF fixposition=1 THEN
    Goto_Fix_Position
  ELSE
  BEGIN
    Check_Level_Guard_Tank(feed,error);
    IF feed THEN
      Start_Feeding(matterhorn,firsttimeM,resting,firsttimeABC)
    ELSE
    BEGIN
      Check_Reloading(reload,reloadstore);
      IF (reload=TRUE) THEN
        Start_Reloading(reloadstore,oldreloadstore,reload,
          firsttimereloading,square,oldnumber,totainumber)
      ELSE
    BEGIN
      GetBit(1,445,level);
      {Omlastning av D fickan}
      IF level=1 THEN
        Start_Reloading_Dstore(square,oldDreload)
      ELSE
    BEGIN
      Check_Mattertop(mattertop);
      IF ((mattertop<=19) AND (matterhorn=TRUE)) THEN
        Refill_Matterhorn(mattertop,resting)
      ELSE
    BEGIN
      IF resting=FALSE THEN
    BEGIN
      Goto_Next_Feeding_Coordinate(presentX,matterhorn,resting,
        oldA,oldB,oldC,oldM,number);
      END
    ELSE
      delay(8000);
    END;
      END;
    END;
  END;
  END;
  END;
  END;
  UNTIL (cancel=TRUE);
END.
```

Bilagor

9.2 PANNPROGRAMMET

Namn: Sattprog

```
Stmnt Instr Identifier Address Comment
0000 :
      Detta är programmet som ligger i Ljungbyverkets SattCon15-20.
      Programmet styr allt i pannan och liten del av traversen.
      Programmet är uppbyggt i moduler. Varje modul är listad för sig med
      hjälpkomentarer som exakt beskriver sats för sats.
00 Include INIT
      *
0001 :
00 Include INITROKG(ROKGAS)
      *
0002 :
00 Include INITSEKL(SEKLUFT)
      *
0003 :
00 Include INITPRIL(PRIMLUFT)
      *
0004 :
00 Include INITTERT(TERTLUFT)
      *
0005 :
00 Include INITTRAV
      *
0006 :
00 Include MÄTVÄRDE
      *
0007 :
00 Include DELTAP
      *
0008 :
00 Include ON-OFF
      *
0009 :
      HÄR SKER ALL AUTOMATISERAD REGLER
      OCH STYRNING AV PANNAN I OFF(AV)MODE
00 LBL LOOP0 L000
      *
0010 :
00 Include AV-TID
      *
0011 :
00 Include ALJUDSOT
      *
0012 :
00 Include ASPJALL
      *
0013 :
00 Include AFLAKTAR
      *
0014 :
00 Include AINMAT
      *
0015 :
00 Include AFILTER
      *
0016 :
00 JC LOOP2 L002
      *
```

Bilagor

0017 :
HÄR SKER ALL AUTOMATISERAD REGLER OCH
STYRNING AV PANNAN IN ON(PÅ) MODE

00 LBL LOOP1 L001
*

0018 :
00 Include PA-TID
*

0019 :
00 Include PLJUDSOT
*

0020 :
00 Include PFLAKTAR
*

0021 :
00 Include PINMAT
*

0022 :
00 Include PULSREGR(ROKGAS)
*

0023 :
00 Include PULSREGS(SEKLUFT)
*

0024 :
00 Include PULSREGP(PRIMLUFT)
*

0025 :
00 Include PULSREGT(TERTLUFT)
*

0026 :
00 Include PELDUTRP
*

0027 :
00 Include PFLOFF
*

0028 :
00 Include PFILTER
*

0029 :
00 Include PELDUTRS
*

0030 :
00 Include PELDUTRT
*

0031 :
00 Include PINMAT2
*

0032 :
HÄR BÖRJAR DE MODULER SOM ÄR GEMENSAMMA FÖR
PANNAN (BÅDE I ON OCH OFF MODE).
HÄR LIGGER ÄVEN DE MODULER SOM UTGÖR
HJÄLPDRIVRUTINERNA TILL TRAVERSEN.

00 LBL LOOP2 L002
*

0033 :
00 Include ROSTER
*

0034 :
00 Include ASKMAT
*

0035 :
00 Include BRAGG
*

0036 :
00 Include TRAVDROJ
*

0037 :
00 Include EFFEKT
*

0038 :
00 Include FLODE
*

Bilagor

0039 :
00 Include LARM
*
0040 :
00 Include SKÁPVAL
*
0041 :
00 Include FINAL
*
0042 :
00 Include FLYTRAV
*
0043 :
00 Include COMINTOP
*
End Of PBS

※

Bilagor
Modul: Init

| Stmnt | Instr | Identifier | Address | Comment |
|--------|-------|-------------------------------------|---------|--|
| 0000 : | | | | |
| 00 | AN | FIRST-SCAN | 1675 | Om första svepet, hoppa inte |
| 01 | JC | <L052> | L052 | |
| | | * | | |
| 0001 : | | | | |
| | | START VÄRDE FÖR REGLER OCH STYRNING | | |
| 00 | GET | K0119 | | STARTVÄRDE PÅ ONTEMP |
| 01 | STO | ONTEMP | Extern | |
| 02 | GET | K0125 | | STARTVÄRDE PÅ OFFTEMP |
| 03 | STO | OFFTEMP | Extern | |
| 04 | GET | K0060 | | STARTVÄRDE PÅ ASKPERIOD1 |
| 05 | STO | ASKPER1 | Extern | |
| 06 | GET | K0090 | | STARTVÄRDE PÅ ASKPERIOD2 |
| 07 | STO | ASKPER2 | Extern | |
| 08 | GET | K0120 | | STARTVÄRDE PÅ ASKPERIOD3 |
| 09 | STO | ASKPER3 | Extern | |
| 10 | GET | K0900 | | ANTAL INMATSEK FÖR ASKTÖMNING |
| 11 | STO | ASKMÄNGD | Extern | |
| 12 | GET | K0003 | | ANTAL SEK FÖR LJUDSOT I OFFLÄG |
| 13 | STO | LJUDSOTTID | Extern | |
| 14 | GET | K0025 | | ANTAL INMATSEK FÖR ROSTERRÖRELSE |
| 15 | STO | ROSTERSTYRT | Extern | |
| | | * | | |
| 0002 : | | | | |
| 00 | GET | K0020 | | INMATNINGSPERIOD(SEK) I OFFLÄGE |
| 01 | STO | AINMAT-PA-P | Extern | |
| 02 | GET | K0200 | | INMATNINGSPAUS(SEK) I OFFLÄGE |
| 03 | STO | AINMAT-PAUS-P | Extern | |
| 04 | GET | K0300 | | FILTERRENINGSTID(SEK) I OFFLÄGE |
| 05 | STO | AFILTERTID | Extern | |
| 06 | GET | K0003 | | LJUDSOTNINGSPERIOD(S) I ONLÄGE |
| 07 | STO | PLJUDS-PA-P | Extern | |
| 08 | GET | K0200 | | LJUDSOTNINGSPAUS I ONLÄGE |
| 09 | STO | PLJUDS-PAUS-P | Extern | |
| 10 | GET | K0070 | | TIDSFÖRDR. INNAN REGLERING RÖKGAS |
| 11 | STO | RÖKFLSTARTT | Extern | |
| 12 | GET | K0110 | | TIDSFÖRDR. INNAN REGLERING AV SEKLUFT |
| 13 | STO | SEKFLSTARTT | Extern | |
| 14 | GET | K0120 | | TIDSFÖRDR. INNAN REGLERING AV PRIMLUFT |
| 15 | STO | PRIFLSTARTT | Extern | |
| | | * | | |
| 0003 : | | | | |
| 00 | GET | K0105 | | TIDSFÖRDR. INNAN START AV SEKUNDÄRFLÄK |
| 01 | STO | SEKFLPÅTID | Extern | |
| 02 | GET | K0115 | | TIDSFÖRDR. INNAN START AV PRIMÄRFLÄKT |
| 03 | STO | PRIFLÄKTPÅT | Extern | |
| 04 | GET | K0001 | | GRÄNS FÖR FILTERERING MED BYPASS |
| 05 | STO | CO-GRÄNS | Extern | |
| 06 | GET | K0001 | | ANTAL FILTERRENINGSSSEK I ONLÄGE |
| 07 | STO | FILTERRENP | Extern | |
| 08 | GET | K9999 | | ANTAL SEKUNDER MELLAN FILTERRENSNINGAR |
| 09 | STO | FILTERPAUSP | Extern | |
| 10 | GET | K0070 | | |
| 11 | STO | DELTBÖP | Extern | BÖRVÄRDE PÅ DELA-P ÖPPET SPJÄLL |
| 12 | GET | K0040 | | |
| 13 | STO | DELTBST | Extern | BÖRVÄRDE PÅ DELTA-P STÅNGT SPJÄLL |
| | | * | | |

Bilagor

0004 :

| | | | | |
|----|-----|---------------|--------|------------------------------------|
| 00 | GET | K0005 | | |
| 01 | STO | DELTAHYST | Extern | HYSTERES PÅ DELTA-P |
| 02 | GET | K0119 | | |
| 03 | STO | PFFLOFFTEMP | Extern | TEMP DÅ PRIMÄRFLÄKTEN SKALL GÅ OFF |
| 04 | GET | K0010 | | |
| 05 | STO | SINMAT-PÅ-P | Extern | INMATNINGSPERIOD DÅ PRIMÄRFLÄKTEN |
| 06 | GET | K0600 | | GÅTT OFF |
| 07 | STO | SINMAT-PAUS-P | Extern | INMATNINGSPAUS DÅ PRIMÄRFLÄKTEN |
| 08 | GET | K0120 | | GÅTT OFF |
| 09 | STO | PRIUNDRY2 | Extern | GRÄNS FÖR ELDSTADSUNDERTRYCKET SÅ |
| 10 | GET | K0105 | | ATT PRIMÄRSPJÄLLET LÅSES |
| 11 | STO | PRIUNDRY1 | Extern | GRÄNS FÖR ELDSTADSUNDERTRYCKET SÅ |
| 12 | GET | K0095 | | PRIMÄRSPJÄLLET LÅSES |
| 13 | STO | SEKUNDRST | Extern | GRÄNS DÅR SEKUNDÄRSPJÄLLET LÅSES |

*

0005 :

| | | | | |
|----|-----|---------------|--------|---------------------------------------|
| 00 | GET | K0004 | | NIVÅ PÅ O2 HALT FÖR ATT TERTIÄRSPJÄLL |
| 01 | STO | TERTO2ÖPPNA | Extern | SKALL ÖPPNA |
| 02 | GET | K0095 | | |
| 03 | STO | TERTUNDRST | Extern | NIVÅ PÅ ELDSTADSUNDERTRYCK FÖR ATT |
| 04 | GET | K3000 | | TERTIÄRSPJÄLLET SKALL STÅNGA |
| 05 | STO | INMATCOMAX1 | Extern | NIVÅ PÅ CO-HALTEN SOM GÖR ATT |
| 06 | GET | K0004 | | TERTIÄRSPJÄLLET ÖPPNAS |
| 07 | STO | INMATO2MIN1 | Extern | NIVÅ PÅ O2-HALTEN SOM GÖR ATT |
| 08 | GET | K0000 | | TERTIÄRSPJÄLLET ÖPPNAS |
| 09 | STO | INMATO2MIN | Extern | O2 HALT SOM ENSAM GÖR ATT |
| 10 | GET | K0120 | | TERTIÄRSPJÄLLET ÖPPNAS |
| 11 | STO | SINMAT2-PAS-P | Extern | INMATNINGSPERIOD OM DET ÄR DÅLIG |
| 12 | GET | K0010 | | FÖRBRÄNNING I PANNAN |
| 13 | STO | SINMAT2-PÅ-P | Extern | INMATNINGSPAUS I FALLET OVAN |

*

0006 :

| | | | | |
|----|-----|-------------|--------|----------------------------------|
| 00 | GET | K0600 | | ANTAL kWh SOM UTGÖR EN MÄTPERIOD |
| 01 | STO | ENERGIPER | Extern | |
| 02 | GET | K0400 | | |
| 03 | STO | FLÖDEPER | Extern | ANTAL m3 SOM UTGÖR EN MÄTPERIOD |
| 04 | GET | K0030 | | |
| 05 | STO | BRANSLEAGX | Extern | ANTAL SEKUNDER SOM BRAGGEN GÅR |
| 06 | GET | K0002 | | |
| 07 | STO | TRAVERSMATP | Extern | ANATAL SEK FÖR IND. BRÄNSLEBRI |
| 08 | GET | K0002 | | |
| 09 | STO | <R220> | R220 | |

*

0007 :

| | | | | |
|----|-----|--------|------|--|
| 00 | LBL | <L052> | L052 | |
|----|-----|--------|------|--|

*

End Of PBS

Bilagor
Module: Initrokg

| Stmnt | Instr | Identifier | Address | Comment |
|--------|-------|------------|---------|---------------------------------------|
| 0000 : | | | | |
| 00 | AN | FIRST-SCAN | 1675 | OM INTE FÖRSTA SVEP SÅ HOPPAFÖRBI |
| 01 | JC | <L053> | L053 | |
| | * | | | |
| 0001 : | | | | |
| 00 | GET | K0170 | | BÖRVÅRDET PÅ ELDSTADSUNDERTRYCKET |
| 01 | STO | BÖR | Extern | |
| 02 | GET | K0020 | | BÖRVÅRDET PÅ ELDSTADSUNDERTRYCKET |
| 03 | STO | PAUSPERIOD | Extern | |
| 04 | GET | K0005 | | PULSLÄNGD (TIONDELSEKUNDER) |
| 05 | STO | PULSPERIOD | Extern | |
| 06 | GET | K0000 | | MINSTA TILLÅTNA LÄGE SPÅ SPJÄLL |
| 07 | STO | MINLÄGE | Extern | |
| 08 | GET | K0100 | | *SÄTT STORT, IDAG FINNS INGEN GIVARE* |
| 09 | STO | SPJÄLLÄGE | Extern | |
| 10 | GET | K0150 | | |
| 11 | STO | UNDERBÖR | Extern | |
| 12 | GET | K0240 | | |
| 13 | STO | ÖVREBÖR | Extern | |
| 14 | GET | K0010 | | *SÄTT STORT, IDAG FINNS INGEN GIVARE* |
| 15 | STO | HYSTER | Extern | |
| | * | | | |
| 0002 : | | | | |
| 00 | LBL | <L053> | L053 | |
| | * | | | |
| | | End Of PBS | | |

Module: Initsekl

| Stmnt | Instr | Identifier | Address | Comment |
|--------|-------|------------|---------|----------------------------------|
| 0000 : | | | | |
| 00 | AN | FIRST-SCAN | 1675 | OM INTE FÖRSTA SVEP, HOPPA FÖRBI |
| 01 | JC | <L054> | L054 | |
| | * | | | |
| 0001 : | | | | |
| 00 | GET | K0008 | | |
| 01 | STO | BÖR | Extern | |
| 02 | GET | K0010 | | |
| 03 | STO | PAUSPERIOD | Extern | |
| 04 | GET | K0003 | | |
| 05 | STO | PULSPERIOD | Extern | |
| 06 | GET | K0020 | | |
| 07 | STO | MINLÄGE | Extern | |
| 08 | GET | K0004 | | |
| 09 | STO | UNDERBÖR | Extern | |
| 10 | GET | K0013 | | |
| 11 | STO | ÖVREBÖR | Extern | |
| 12 | GET | K0001 | | |
| 13 | STO | HYSTER | Extern | |
| | * | | | |
| 0002 : | | | | |
| 00 | LBL | <L054> | L054 | |
| | * | | | |
| | | End Of PBS | | |

Bilagor
Module: Initpril

| Stmnt | Instr | Identifier | Address | Comment |
|--------|-------|------------|---------|-----------------------------------|
| 0000 : | | | | |
| 00 | AN | FIRST-SCAN | 1675 | OM INTE FÖRSTA SVEPSÅ HOPPA FÖRBI |
| 01 | JC | <L055> | L055 | |
| | * | | | |
| 0001 : | | | | |
| 00 | GET | K0115 | | BÖRVÅRDET PÅ VATTENTEMPERATUREN |
| 01 | STO | BÖR | Extern | |
| 02 | GET | K0030 | | |
| 03 | STO | PAUSPERIOD | Extern | |
| 04 | GET | K0010 | | |
| 05 | STO | PULSPERIOD | Extern | |
| 06 | GET | K0025 | | |
| 07 | STO | MINLÄGE | Extern | |
| 08 | GET | K0110 | | |
| 09 | STO | UNDERBÖR | Extern | |
| 10 | GET | K0118 | | |
| 11 | STO | ÖVREBÖR | Extern | |
| 12 | GET | K0001 | | |
| 13 | STO | HYSTER | Extern | |
| | * | | | |
| 0002 : | | | | |
| 00 | LBL | <L055> | L055 | |
| | * | | | |
| | | End Of PBS | | |

Module: Inittert

| Stmnt | Instr | Identifier | Address | Comment |
|--------|-------|------------|---------|-----------------------------------|
| 0000 : | | | | |
| 00 | AN | FIRST-SCAN | 1675 | OM INTE FÖRSTA SVEPSÅ HOPPA FÖRBI |
| 01 | JC | <L061> | L061 | |
| | * | | | |
| 0001 : | | | | |
| 00 | GET | K0050 | | |
| 01 | STO | BÖR | Extern | |
| 02 | GET | K0030 | | |
| 03 | STO | PAUSPERIOD | Extern | |
| 04 | GET | K0005 | | |
| 05 | STO | PULSPERIOD | Extern | |
| 06 | GET | K0010 | | |
| 07 | STO | MINLÄGE | Extern | |
| 08 | GET | K0001 | | |
| 09 | STO | UNDERBÖR | Extern | |
| 10 | GET | K0800 | | |
| 11 | STO | ÖVREBÖR | Extern | |
| 12 | GET | K0030 | | |
| 13 | STO | HYSTER | Extern | |
| | * | | | |
| 0002 : | | | | |
| 00 | LBL | <L061> | L061 | |
| | * | | | |
| | | End Of PBS | | |

Bilagor

Module: Inittrav

| Stmnt | Instr | Identifier | Address | Comment |
|--------|-------|------------|---------|-----------------------|
| 0000 : | | | | |
| 00 | AN | FIRST-SCAN | 1675 | |
| 01 | JC | <L090> | L090 | |
| | * | | | |
| 0001 : | | | | |
| 00 | GET | K0000 | | initiera bränslevalet |
| 01 | STO | <R130> | R130 | i koordinaterna |
| 02 | STO | <R131> | R131 | |
| 03 | STO | <R132> | R132 | |
| 04 | STO | <R133> | R133 | |
| 05 | STO | <R134> | R134 | |
| 06 | STO | <R135> | R135 | |
| 07 | STO | <R136> | R136 | |
| 08 | STO | <R137> | R137 | |
| 09 | STO | <R138> | R138 | |
| 10 | STO | <R139> | R139 | |
| 11 | STO | <R140> | R140 | |
| 12 | STO | <R141> | R141 | |
| 13 | STO | <R142> | R142 | |
| 14 | STO | <R143> | R143 | |
| 15 | STO | <R144> | R144 | |
| 16 | STO | <R145> | R145 | |
| 17 | STO | <R146> | R146 | |
| | * | | | |
| 0002 : | | | | |
| 00 | GET | K0000 | | |
| 01 | STO | <R147> | R147 | |
| 02 | STO | <R148> | R148 | |
| 03 | STO | <R149> | R149 | |
| 04 | STO | <R150> | R150 | |
| 05 | STO | <R151> | R151 | |
| 06 | STO | <R152> | R152 | |
| 07 | STO | <R153> | R153 | |
| 08 | STO | <R154> | R154 | |
| 09 | STO | <R155> | R155 | |
| 10 | STO | <R156> | R156 | |
| 11 | STO | <R157> | R157 | |
| 12 | STO | <R158> | R158 | |
| 13 | STO | <R159> | R159 | |
| 14 | STO | <R160> | R160 | |
| 15 | STO | <R161> | R161 | |
| | * | | | |
| 0003 : | | | | |
| 00 | GET | K0000 | | |
| 01 | STO | <R162> | R162 | |
| 02 | STO | <R163> | R163 | |
| 03 | STO | <R164> | R164 | |
| 04 | STO | <R165> | R165 | |
| 05 | STO | <R166> | R166 | |
| 06 | STO | <R167> | R167 | |
| 07 | STO | <R168> | R168 | |
| 08 | STO | <R169> | R169 | |
| 09 | STO | <R170> | R170 | |
| 10 | STO | <R171> | R171 | |
| 11 | STO | <R172> | R172 | |
| 12 | STO | <R173> | R173 | |
| 13 | STO | <R174> | R174 | |
| 14 | STO | <R175> | R175 | |
| 15 | STO | <R176> | R176 | |
| 16 | STO | <R177> | R177 | |
| 17 | STO | <R178> | R178 | |
| 18 | STO | <R179> | R179 | |
| | * | | | |

Bilagor

0004 :

| | | | |
|----|-----|--------|------|
| 00 | GET | K0000 | |
| 01 | STO | <R180> | R180 |
| 02 | STO | <R181> | R181 |
| 03 | STO | <R182> | R182 |
| 04 | STO | <R183> | R183 |
| 05 | STO | <R184> | R184 |
| 06 | STO | <R185> | R185 |
| 07 | STO | <R186> | R186 |
| 08 | STO | <R187> | R187 |
| 09 | STO | <R188> | R188 |
| 10 | STO | <R189> | R189 |
| 11 | STO | <R190> | R190 |

*

0005 :

| | | | | |
|----|-----|--------|------|-------------------------|
| 00 | GET | K0001 | | Initera inmatningsföljd |
| 01 | STO | <R201> | R201 | |
| 02 | GET | K0009 | | |
| 03 | STO | <R202> | R202 | |
| 04 | STO | <R203> | R203 | |
| 05 | STO | <R203> | R203 | |
| 06 | STO | <R204> | R204 | |
| 07 | STO | <R205> | R205 | |
| 08 | STO | <R206> | R206 | |
| 09 | STO | <R207> | R207 | |
| 10 | STO | <R208> | R208 | |
| 11 | STO | <R209> | R209 | |
| 12 | STO | <R210> | R210 | |
| 13 | STO | <R211> | R211 | |
| 14 | STO | <R212> | R212 | |

*

0006 :

| | | | |
|----|-----|--------|------|
| 00 | GET | K0005 | |
| 01 | STO | <R222> | R222 |

*

0007 :

| | | | | |
|----|----|--------|------|--------------------------|
| 00 | =N | <1103> | 1103 | sätt flagga till travers |
|----|----|--------|------|--------------------------|

*

0008 :

| | | | |
|----|-----|--------|------|
| 00 | LBL | <L090> | L090 |
|----|-----|--------|------|

*

End Of PBS

Bilagor

Module: Mätvärde

| Stmnt | Instr | Identifier | Address | Comment |
|--------|-------|-------------|---------|-----------------------------|
| 0000 : | | | | |
| 00 | GET | <AI640> | AI640 | HÄMTA IN ALLA ANALOGA VÄRDE |
| 01 | STO | FRAMTEMP | Extern | FRÅN PROCESSEN |
| 02 | GET | <AI644> | AI644 | |
| 03 | STO | RETURTEMP | Extern | |
| 04 | GET | <AI650> | AI650 | |
| 05 | STO | UGNSTEMP | Extern | |
| 06 | GET | <AI654> | AI654 | |
| 07 | STO | O2-HALT | Extern | |
| 08 | GET | <AI660> | AI660 | |
| 09 | STO | CO-HALT | Extern | |
| 10 | GET | <AI664> | AI664 | |
| 11 | STO | DELTA-P | Extern | |
| 12 | GET | <AI670> | AI670 | |
| 13 | STO | ELDUTRYCK | Extern | |
| 14 | GET | <AI700> | AI700 | |
| 15 | STO | ROKGASTEMP | Extern | |
| 16 | GET | <AI704> | AI704 | |
| 17 | STO | ECOGASTEMP | Extern | |
| * | | | | |
| 0001 : | | | | |
| 00 | GET | <AI714> | AI714 | gradera primärspjälläge |
| 01 | DIV | K0019 | | |
| 02 | MUL | K0008 | | |
| 03 | DIV | K0100 | | |
| 04 | STO | PSPJÄLLÄGE | Extern | |
| * | | | | |
| 0002 : | | | | |
| 00 | GET | K0180 | | GRADERA PRIMÄRSPJÄLLÄGE |
| 01 | SUB | PSPJÄLLÄGE | Extern | |
| 02 | STO | PSPJÄLLÄGE | Extern | |
| * | | | | |
| 0003 : | | | | |
| 00 | A | PULSE-1HZ | 1677 | MEDELVÄRDEBILDA PSPJÄLL |
| 01 | EDGE | <1053> | 1053 | |
| 02 | INC | ANMÄTPSPJ | Extern | |
| 03 | GET | TOTSPJÄLL | Extern | |
| 04 | ADD | PSPJÄLLÄGE | Extern | |
| 05 | STO | TOTSPJÄLL | Extern | |
| * | | | | |
| 0004 : | | | | |
| 00 | GET | ANMÄTPSPJ | Extern | |
| 01 | CMP | K0005 | | |
| 02 | O | GREATER | 1673 | > |
| 03 | GET | TOTSPJÄLL | Extern | |
| 04 | DIV | K0006 | | |
| 05 | STO | MPSPJÄLLÄGE | Extern | |
| * | | | | |
| 0005 : | | | | |
| 00 | GET | ANMÄTPSPJ | Extern | |
| 01 | CMP | K0005 | | |
| 02 | O | GREATER | 1673 | > |
| 03 | GET | K0000 | | |
| 04 | STO | TOTSPJÄLL | Extern | |
| 05 | STO | ANMÄTPSPJ | Extern | |
| * | | | | |
| 0006 : | | | | |
| 00 | GET | MPSPJÄLLÄGE | Extern | SLUT MEDELVÄRDESBILDNING |
| 01 | STO | PSPJÄLLÄGE | Extern | |
| * | | | | |
| 0007 : | | | | |
| 00 | GET | <AI720> | AI720 | GRADERA TERTIÄRSPJÄLLÄGET |
| 01 | DIV | K0039 | | |
| 02 | MUL | K0010 | | |
| 03 | DIV | K0100 | | |
| 04 | STO | TSPJÄLLÄGE | Extern | |
| * | | | | |
| 0008 : | | | | |
| 00 | GET | K0133 | | |

Bilagor

| | | | | |
|------|------|---------------|--------|--------------------------|
| 01 | SUB | TSPJÄLLÄGE | Extern | |
| 02 | STO | TSPJÄLLÄGE | Extern | |
| | * | | | |
| 0009 | : | | | |
| 00 | GET | <AI724> | AI724 | GRADERA SEKSPJÄLLÄGE |
| 01 | DIV | K0655 | | |
| 02 | STO | SEK-SPJÄLLÄGE | Extern | |
| | * | | | |
| 0010 | : | | | |
| 00 | GET | K0098 | | |
| 01 | SUB | SEK-SPJÄLLÄGE | Extern | |
| 02 | STO | SEK-SPJÄLLÄGE | Extern | |
| | * | | | |
| 0011 | : | | | |
| 00 | GET | FRAMTEMP | Extern | GRADERA FRAMTEMPEN |
| 01 | MUL | K0200 | | |
| 02 | DIV | K65535 | | |
| 03 | STO | FRAMTEMP | Extern | |
| | * | | | |
| 0012 | : | | | |
| 00 | GET | RETURTEMP | Extern | GRADERA RETURTEMPEN |
| 01 | MUL | K0200 | | |
| 02 | DIV | K65535 | | |
| 03 | STO | RETURTEMP | Extern | |
| | * | | | |
| 0013 | : | | | |
| 00 | GET | UGNSTEMP | Extern | GRADERA UGNSTEMPEN |
| 01 | MUL | K1100 | | |
| 02 | DIV | K65535 | | |
| 03 | ADD | K0440 | | |
| 04 | STO | UGNSTEMP | Extern | |
| | * | | | |
| 0014 | : | | | |
| 00 | GET | O2-HALT | Extern | GRADERA O2-HALTEN |
| 01 | MUL | K0022 | | |
| 02 | DIV | K65535 | | |
| 03 | STO | O2-HALT | Extern | |
| | * | | | |
| 0015 | : | | | |
| 00 | GET | CO-HALT | Extern | GRADERA CO-HALTEN |
| 01 | MUL | K5000 | | |
| 02 | DIV | K65535 | | |
| 03 | STO | CO-HALT | Extern | |
| | * | | | |
| 0016 | : | | | |
| 00 | GET | DELTA-P | Extern | GRADERA DELTA-P |
| 01 | MUL | K0100 | | |
| 02 | DIV | K65535 | | |
| 03 | STO | DELTA-P | Extern | |
| | * | | | |
| 0017 | : | | | |
| 00 | A | PULSE-1HZ | 1677 | MEDELVÄRDESBILDA DELTA-P |
| 01 | EDGE | <1055> | 1055 | |
| 02 | INC | ANTMÄTD-P | Extern | |
| 03 | GET | TOTD-P | Extern | |
| 04 | ADD | DELTA-P | Extern | |
| 05 | STO | TOTD-P | Extern | |
| | * | | | |
| 0018 | : | | | |
| 00 | GET | ANTMÄTD-P | Extern | MEDELVÄRDESBILDA DELTA-P |
| 01 | CMP | K0020 | | |
| 02 | O | GREATER | 1673 | > |
| 03 | GET | TOTD-P | Extern | |
| 04 | DIV | K0020 | | |
| 05 | STO | MD-P-LÄGE | Extern | |
| | * | | | |
| 0019 | : | | | |
| 00 | GET | ANTMÄTD-P | Extern | MEDELVÄRDESBILDA DELTA-P |
| 01 | CMP | K0020 | | |
| 02 | O | GREATER | 1673 | > |
| 03 | GET | K0000 | | |
| 04 | STO | TOTD-P | Extern | |
| 05 | STO | ANTMÄTD-P | Extern | |
| | * | | | |
| 0020 | : | | | |
| 00 | GET | MD-P-LÄGE | Extern | MEDELVÄRDESBILDA DELTA-P |

Bilagor
01 STO
*

DELTA-P

Extern

Bilagor

0021 :
00 GET ELDUTRYCK Extern GRADERA ELDSTADSUNDERTRYCKET
01 MUL K0500
02 DIV K65535
03 STO ELDUTRYCK Extern
*
0022 :
00 GET ROKGASTEMP Extern GRADERA RÖKASTEMPERATUREN
01 MUL K0400
02 DIV K65535
03 STO ROKGASTEMP Extern
*
0023 :
00 GET ECOGASTEMP Extern GRADERA RÖKASTEMPERATUREN EFTER
01 MUL K0400 ECONOMEISTERN
02 DIV K65535
03 STO ECOGASTEMP Extern
*
0024 :
00 GET <AI710> AI710 GRADERA PANNTAMPERN FRÅ PT-100 GIVAREN
01 DIV K0100
02 MUL K0016
03 DIV K0065
04 ADD K0005
05 STO PANNTAMP Extern
*
End Of PBS

Module: Deltap

| Stamt | Instr | Identifier | Address | Comment |
|------------|-------|------------|---------|-------------------------------------|
| 0000 : | | | | |
| 00 | GET | DELTBÖP | Extern | BERÄKNA DET FLYTANDE DELTABÖRVÄRDET |
| 01 | SUB | DELTBST | Extern | |
| 02 | RP | | | |
| 03 | MUL | PSPJÄLLÄGE | Extern | |
| 04 | DIV | K0100 | | |
| 05 | ADD | DELTBST | Extern | |
| 06 | STO | DELTABÖR | Extern | |
| * | | | | |
| 0001 : | | | | |
| 00 | GET | K0100 | | BERÄKNA DELTAP-ON |
| 01 | ADD | DELTAHYST | Extern | |
| 02 | STO | DELTAP-ON | Extern | |
| 03 | MUL | DELTABÖR | Extern | |
| 04 | DIV | K0100 | | |
| 05 | STO | DELTAPOFF | Extern | |
| * | | | | |
| 0002 : | | | | |
| 00 | GET | K0100 | | BERÄKNA DELTAP-OFF |
| 01 | SUB | DELTAHYST | Extern | |
| 02 | STO | DELTAP-ON | Extern | |
| 03 | MUL | DELTABÖR | Extern | |
| 04 | DIV | K0100 | | |
| 05 | STO | DELTAP-ON | Extern | |
| * | | | | |
| End Of PBS | | | | |

Bilagor
Module: On-Off

| Stmnt | Instr | Identifier | Address | Comment |
|--------|-------|-------------|---------|---------------------------------------|
| 0000 : | | | | |
| 00 | GET | PANNTEMP | Extern | ÄR FRAMTEMPERATUREN UNDER ONTEMP? |
| 01 | CMP | ONTEMP | Extern | |
| 02 | O | LESS | 1671 | < |
| 03 | = | UNDER-ON-T | Extern | |
| | * | | | |
| 0001 : | | | | |
| 00 | GET | PANNTEMP | Extern | ÄR FRAMTEMPERATUREN UNDER OFFTEMP |
| 01 | CMP | OFFTEMP | Extern | |
| 02 | O | LESS | 1671 | < |
| 03 | = | UNDER-OFF-T | Extern | |
| | * | | | |
| 0002 : | | | | |
| 00 | A | UNDER-OFF-T | Extern | SKA PANNAN GÅ ON ELLER OFF (1=ON) |
| 01 | A | ON-OFF | Extern | |
| 02 | O | UNDER-ON-T | Extern | |
| 03 | = | ON-OFF | Extern | |
| | * | | | |
| 0003 : | | | | |
| 00 | AN | <0446> | 0446 | UTLÖST NEDELNINGSTERMOSTAT |
| 01 | =R | ON-OFF | Extern | |
| | * | | | |
| 0004 : | | | | |
| 00 | A | ON-OFF | Extern | OMSLAG TILL ON GER NOLLSTÄLL ONTID |
| 01 | EDGE | <1000> | 1000 | |
| 02 | GET | K0000 | | |
| 03 | STO | ONTID | Extern | |
| 04 | = | UPPSTART | Extern | |
| 05 | = | STANGNING | Extern | |
| | * | | | |
| 0005 : | | | | |
| 00 | A | ON-OFF | Extern | OMSLAG TILL OFF GER NOLLSTÄLL OFF TID |
| 01 | COM | | | |
| 02 | EDGE | <1001> | 1001 | |
| 03 | GET | K0000 | | |
| 04 | STO | OFFTID | Extern | |
| | * | | | |
| 0006 : | | | | |
| 00 | A | ON-OFF | Extern | OM ON SÅ HOPPA TILL ON MODULER |
| 01 | JC | <L001> | L001 | |
| | * | | | |
| 0007 : | | | | |
| 00 | JC | <L000> | L000 | HOPPA TILL OFF MODULER |
| | * | | | |
| | | End Of PBS | | |

Module: Av-Tid

| Stmnt | Instr | Identifier | Address | Comment |
|--------|-------|------------|---------|----------------------------------|
| 0000 : | | | | |
| 00 | A | PULSE-1HZ | 1677 | ÖKA OFF TID PÅ POSITIV FLANK |
| 01 | EDGE | <1002> | 1002 | |
| 02 | INC | OFFTID | Extern | |
| | * | | | |
| 0001 : | | | | |
| 00 | GET | OFFTID | Extern | OM OFFTID >18 H BLIR OFFTID 18 H |
| 01 | CMP | K60000 | | |
| 02 | O | GREATER | 1673 | > |
| 03 | GET | K60000 | | |
| 04 | STO | OFFTID | Extern | |
| | * | | | |
| | | End Of PBS | | |

Bilagor

Module: Aljudsot

| Stmnt | Instr | Identifier | Address | Comment |
|--------|-------|------------|---------|------------------------------|
| 0000 : | | | | |
| 00 | GET | OFFTID | Extern | LJUDSOTA DE FÖRSTASEKUNDERNA |
| 01 | CMP | LJUDSOTTID | Extern | EFTER OMSLAG TILL OFF |
| 02 | O | LESS | 1671 | < |
| 03 | = | LJUDSOTF | Extern | |
| | * | | | |
| | | | | End Of PBS |

Module: Aspjäll

| Stmnt | Instr | Identifier | Address | Comment |
|--------|-------|------------|---------|----------------------|
| 0000 : | | | | |
| 00 | = | ÖPPNARSPJF | Extern | ÖPNNA RÖKGASSPJÄLL . |
| 01 | =N | STÄNGRSPJF | Extern | STÄNG PRI,SEK,SPJÄLL |
| 02 | = | STÄNGPSPJF | Extern | |
| 03 | =N | ÖPPNAPSPJF | Extern | |
| 04 | = | STÄNGSSPJF | Extern | |
| 05 | =N | ÖPPNASSPJF | Extern | |
| 06 | = | STÄNGTSPJF | Extern | |
| 07 | =N | ÖPPNATSPJF | Extern | |
| | * | | | |
| | | | | End Of PBS |

※

Module: Afläktar

| Stmnt | Instr | Identifier | Address | Comment |
|--------|-------|------------|---------|-----------------------|
| 0000 : | | | | |
| 00 | =N | RÖKFLÄKTFL | Extern | STÄNG AV ALLA FLÄKTAR |
| 01 | =N | PRIFLÄKTFL | Extern | |
| 02 | =N | SEKFLÄKTFL | Extern | |
| | * | | | |
| | | | | End Of PBS |

Bilagor

Module: Ainmat

| Stmnt | Instr | Identifier | Address | Comment |
|--------|------------|---------------|---------|------------------------------------|
| 0000 : | | | | |
| 00 | AN | AINMAT-PÅ-AV | Extern | OM INMATNINGEN AR IPAUS HOPPA |
| 01 | JC | <L013> | L013 | |
| | * | | | |
| 0001 : | | | | |
| 00 | A | PULSE-1HZ | 1677 | OKA PATIDEN FOR INMATNING |
| 01 | EDGE | <1006> | 1006 | |
| 02 | INC | AINMAT-PA-T | Extern | |
| | * | | | |
| 0002 : | | | | |
| 00 | = | INMATSFL | Extern | SATT PÅ INMATNING |
| | * | | | |
| 0003 : | | | | |
| 00 | GET | AINMAT-PA-T | Extern | KOLLA OM PA-TIDEN AR SLUT |
| 01 | CMP | AINMAT-PA-P | Extern | |
| 02 | O | GREATER | 1673 | > |
| 03 | =N | AINMAT-PÅ-AV | Extern | |
| 04 | =N | INMATSFL | Extern | |
| | * | | | |
| 0004 : | | | | |
| 00 | NPW | | | |
| | * | | | |
| 0005 : | | | | |
| 00 | JC | <L014> | L014 | HOPPA TILL SLUTET FOR INMATNINGEN |
| | * | | | |
| 0006 : | | | | |
| 00 | LBL | <L013> | L013 | HAR BORJAR INMATNINGEN I PAUS LAGE |
| | * | | | |
| 0007 : | | | | |
| 00 | A | PULSE-1HZ | 1677 | OKA PAUSTIDEN |
| 01 | EDGE | <1007> | 1007 | |
| 02 | INC | AINMAT-PAUS-T | Extern | |
| | * | | | |
| 0008 : | | | | |
| 00 | =N | INMATSFL | Extern | INGEN INMATNINGS SIGNAL |
| | * | | | |
| 0009 : | | | | |
| 00 | GET | AINMAT-PAUS-T | Extern | KOLLA OM PAUSTIDEN AR SLUT |
| 01 | CMP | AINMAT-PAUS-P | Extern | |
| 02 | O | GREATER | 1673 | > |
| 03 | = | AINMAT-PÅ-AV | Extern | |
| 04 | = | INMATSFL | Extern | |
| 05 | GET | K0000 | | |
| 06 | STO | AINMAT-PA-T | Extern | |
| 07 | STO | AINMAT-PAUS-T | Extern | |
| | * | | | |
| 0010 : | | | | |
| 00 | LBL | <L014> | L014 | SLUTET FOR INMATNINGEN |
| | * | | | |
| | End Of PBS | | | |

Module: Afilter

| Stmnt | Instr | Identifier | Address | Comment |
|--------|-------|------------|---------|----------------------------------|
| 0000 : | | | | |
| 00 | GET | OFFTID | Extern | FILTERRENSA DE FÖRSTA SEKUNDERNA |
| 01 | CMP | AFILTERTID | Extern | EFETR OMSLAG TILL OFF |
| 02 | O | LESS | 1671 | < |
| 03 | = | FILTERFLAG | Extern | |
| | * | | | |
| 0001 : | | | | |
| 00 | = | BYPASSF | Extern | BYPASSA ALLTID I OFFLÅGE |

Bilagor

*
End Of PBS

Module: Pa-Tid

| Stmnt | Instr | Identifier | Address | Comment |
|--------|-------|------------|---------|-------------------------------|
| 0000 : | | | | |
| 00 | A | PULSE-1HZ | 1677 | OKA ONTID PA POSITIV FLANK |
| 01 | EDGE | <1011> | 1011 | |
| 02 | AN | UPPSTART | Extern | |
| 03 | INC | ONTID | Extern | |
| | * | | | |
| 0001 : | | | | |
| 00 | GET | ONTID | Extern | OM ONTID >18 H BLIR ONTID=18H |
| 01 | CMP | K60000 | | |
| 02 | O | GREATER | 1673 | > |
| 03 | GET | K60000 | | |
| 04 | STO | ONTID | Extern | |
| | * | | | |
| | | | | End Of PBS |

Bilagor

Module: Pljudsot

| Stmnt | Instr | Identifier | Address | Comment |
|--------|------------|---------------|---------|--|
| 0000 : | | | | |
| 00 | AN | PLJUDS-PA-AV | Extern | OM LJUDSOTNINGEN AR I PAUS,HOPPA |
| 01 | JC | <L015> | L015 | |
| | * | | | |
| 0001 : | | | | |
| 00 | A | PULSE-1HZ | 1677 | OKA LJUDSOTNINGSTIDEN PA POSITIV FLANK |
| 01 | EDGE | <1012> | 1012 | |
| 02 | INC | PLJUDS-PA-T | Extern | |
| | * | | | |
| 0002 : | | | | |
| 00 | GET | PLJUDS-PA-T | Extern | STANG AV LJUDSOTNINGEN OM |
| 01 | CMP | PLJUDS-PA-P | Extern | OM TIDEN AR SLUT,SAMT NOLLSTALL |
| 02 | O | GREATER | 1673 | TIDEN |
| 03 | =N | PLJUDS-PA-AV | Extern | |
| 04 | =N | LJUDSOTF | Extern | |
| 05 | GET | K0000 | | |
| 06 | STO | PLJUDS-PA-T | Extern | |
| | * | | | |
| 0003 : | | | | |
| 00 | JC | <L016> | L016 | HOPPA FORBI LJUDSOTNING I PAUS |
| | * | | | |
| 0004 : | | | | |
| 00 | LBL | <L015> | L015 | HAR BORJAR LJUDSOTNING I PAUSLAGE |
| | * | | | |
| 0005 : | | | | |
| 00 | A | PULSE-1HZ | 1677 | OKA PAUSTIDEN FOR LJUDSOTNING |
| 01 | EDGE | <1013> | 1013 | |
| 02 | INC | PLJUDS-PAUS-T | Extern | |
| | * | | | |
| 0006 : | | | | |
| 00 | GET | PLJUDS-PAUS-T | Extern | OM LJUDSOTNINGEN AR KLAR |
| 01 | CMP | PLJUDS-PAUS-P | Extern | ,NOLLSTALL ALLA PERIODER OCH |
| 02 | O | GREATER | 1673 | BORJA OM FRAN BORJAN |
| 03 | = | PLJUDS-PA-AV | Extern | |
| 04 | = | LJUDSOTF | Extern | |
| 05 | GET | K0000 | | |
| 06 | STO | PLJUDS-PAUS-T | Extern | |
| | * | | | |
| 0007 : | | | | |
| 00 | LBL | <L016> | L016 | |
| | * | | | |
| | End Of PBS | | | |

Bilagor

Module: Pflaktar

| Stmnt | Instr | Identifier | Address | Comment |
|--------|-------|--------------|---------|--|
| 0000 : | | | | |
| 00 | AN | STANGNING | Extern | OM STANGNIG AV SPJALL KLARA HOPPA |
| 01 | JC | <L022> | L022 | |
| | * | | | |
| 0001 : | | | | |
| 00 | = | STÄNGRSPJF | Extern | STÄNG ALLA SPJALL |
| 01 | = | STÄNGPSPJF | Extern | |
| 02 | = | STÄNGSSPJF | Extern | |
| 03 | =N | ÖPPNARSPJF | Extern | |
| 04 | =N | ÖPPNASSPJF | Extern | |
| 05 | =N | ÖPPNAPSPJF | Extern | |
| | * | | | |
| 0002 : | | | | |
| 00 | = | STANGNING | Extern | |
| 01 | = | UPPSTART | Extern | |
| | * | | | |
| 0003 : | | | | |
| 00 | NPW | | | |
| | * | | | |
| 0004 : | | | | |
| 00 | JC | <L023> | L023 | HOPPA TILL SLUT |
| | * | | | |
| 0005 : | | | | |
| 00 | LBL | <L022> | L022 | |
| | * | | | |
| 0006 : | | | | |
| 00 | = | RÖKFLÄKTFL | Extern | SÄTT I GÅNG RÖKGASFLÄKTEN |
| | * | | | |
| 0007 : | | | | |
| 00 | GET | ONTID | Extern | TILLÅT ATT REGLERA SEKSPJÄLL |
| 01 | CMP | SEKFLSTARTT | Extern | |
| 02 | O | GREATER | 1673 | > |
| 03 | =N | SREGLERSPÄRR | Extern | |
| | * | | | |
| 0008 : | | | | |
| 00 | GET | ONTID | Extern | |
| 01 | CMP | PRIFLSTARTT | Extern | |
| 02 | O | GREATER | 1673 | > |
| 03 | =N | PREGLERSPÄRR | Extern | |
| | * | | | |
| 0009 : | | | | |
| 00 | GET | ONTID | Extern | SÄTT PÅ FLAKTARNA EN TID EFTER SPJALL- |
| 01 | CMP | SEKFLPÅTID | Extern | STÄNGNING |
| 02 | O | GREATER | 1673 | > |
| 03 | = | SEKFLÄKTFL | Extern | |
| | * | | | |
| 0010 : | | | | |
| 00 | GET | ONTID | Extern | TILLÅT ATT BÖRJA REGLERA RÖGGASSPJÄLL |
| 01 | CMP | RÖKFLSTARTT | Extern | |
| 02 | O | GREATER | 1673 | > |
| 03 | =N | RREGLERSPÄRR | Extern | |
| | * | | | |
| 0011 : | | | | |
| 00 | GET | ONTID | Extern | |
| 01 | CMP | PRIFLÄKTPÅT | Extern | |
| 02 | O | GREATER | 1673 | > |
| 03 | = | PRIFLÄKTFL | Extern | |
| 04 | =N | UPPSTART | Extern | |
| | * | | | |
| 0012 : | | | | |
| 00 | LBL | <L023> | L023 | |
| | * | | | |
| | | | | End Of PBS |

Bilagor

Module: Pinmat

| Stmnt | Instr | Identifier | Address | Comment |
|--------|--------|--------------|---------|---------------------------------------|
| 0000 : | | | | |
| | 00 A | PINMAT-PÅ-AV | Extern | OM INMATNING PÅGÅR HOPPA |
| | 01 JC | <L029> | L029 | |
| | * | | | |
| 0001 : | | | | |
| | 00 GET | DELTA-P | Extern | OM DELTA-P UNDER GRÄNS, SÅTT PÅ |
| | 01 CMP | DELTA-P-ON | Extern | |
| | 02 O | LESS | 1671 | < |
| | 03 = | PINMAT-PÅ-AV | Extern | |
| | * | | | |
| 0002 : | | | | |
| | 00 JC | <L030> | L030 | |
| | * | | | |
| 0003 : | | | | |
| | 00 LBL | <L029> | L029 | |
| | * | | | |
| 0004 : | | | | |
| | 00 GET | DELTA-P | Extern | STÄNG AV OM INMATNINGEN ÄR ÖVER GRÄNS |
| | 01 CMP | DELTAPOFF | Extern | |
| | 02 O | GREATER | 1673 | > |
| | 03 =N | PINMAT-PÅ-AV | Extern | |
| | * | | | |
| 0005 : | | | | |
| | 00 LBL | <L030> | L030 | |
| | * | | | |
| 0006 : | | | | |
| | 00 A | PINMAT-PÅ-AV | Extern | OM INMATNING SKALL SKE, |
| | 01 = | INMATSFL | Extern | ETTSTÄLL FLAGGAN FÖR INMATNING; |
| | * | | | |
| | | | | End Of PBS |

Bilagor

Module: Pulsregr

| Stmnt | Instr | Identifier | Address | Comment |
|--------|-------|--------------|---------|---|
| 0000 : | | | | DENNA MODUL HAR SEPARAT IDENTIFIERAR LISTA.DÄRFÖR SKALL DEN SPECIELLA IDF LISTAN STUDERAS FÖR MODULEN |
| 00 | = | MINSKAFLAGGA | Extern | SÄTTER SPJÄLLET PÅ STÄNGNING,DET |
| 01 | =N | ÖKAFLAGGA | Extern | BEHÖVS SENARE |
| | * | | | |
| 0001 : | | | | |
| 00 | A | REGLERSPÄRR | Extern | OM REGLERINGEN ÄNNU INTE FÅR |
| 01 | JC | <L028> | L028 | BÖRJA,SÅ HOPPAR MAN FÖRBI |
| | * | | | |
| 0002 : | | | | |
| 00 | A | PULSFLAGGA | Extern | HOPPAR PÅ REGULATORNS KLOCKPULS |
| 01 | JC | <L024> | L024 | |
| | * | | | |
| 0003 : | | | | |
| 00 | A | PULSE-1HZ | 1677 | ÖKA PAUSTID |
| 01 | EDGE | <1032> | 1032 | |
| 02 | INC | PAUSTID | Extern | |
| | * | | | |
| 0004 : | | | | |
| 00 | GET | PAUSPERIOD | Extern | OM PAUSTID AR SLUT,BORJA PULS |
| 01 | CMP | PAUSTID | Extern | |
| 02 | O | LESS | 1671 | < |
| 03 | = | PULSFLAGGA | Extern | |
| 04 | GET | K0000 | | |
| 05 | STO | PAUSTID | Extern | |
| 06 | STO | PULSTID | Extern | |
| | * | | | |
| 0005 : | | | | |
| 00 | JC | <L025> | L025 | HOPPA FÖRBI PULSNING |
| | * | | | |
| 0006 : | | | | |
| 00 | LBL | <L024> | L024 | PULSNING |
| | * | | | |
| 0007 : | | | | |
| 00 | A | PULSE-1HZ | 1677 | ÖKA PULSTID PÅ FLANK |
| 01 | EDGE | <1031> | 1031 | |
| 02 | INC | PULSTID | Extern | |
| | * | | | |
| 0008 : | | | | |
| 00 | GET | PULSPERIOD | Extern | OM PULSTID SLUT OVERGÅ TILL PAUS |
| 01 | CMP | PULSTID | Extern | |
| 02 | O | LESS | 1671 | < |
| 03 | =N | PULSFLAGGA | Extern | |
| | * | | | |
| 0009 : | | | | |
| 00 | LBL | <L025> | L025 | |
| | * | | | |
| 0010 : | | | | |
| 00 | GET | HALT | Extern | OM HALTEN AR ÖVER ÖVERBÖR SÅ MINSKA |
| 01 | CMP | ÖVREBÖR | Extern | HELA TIDEN OCH HOPPA VIDARE TILL SLUT |
| 02 | O | GREATER | 1673 | > |
| 03 | =N | ÖKAFLAGGA | Extern | |
| 04 | =N | SPJALLOKEY | Extern | |
| 05 | = | MINSKAFLAGGA | Extern | |
| 06 | JC | <L026> | L026 | |
| | * | | | |
| 0011 : | | | | |
| 00 | GET | HALT | Extern | OM HALTEN ÄR UNDER UNDERBÖR SÅ ÖKA |
| 01 | CMP | UNDERBÖR | Extern | HELA TIDEN OCH HOPPA VIDARE |
| 02 | O | LESS | 1671 | < |
| 03 | =N | MINSKAFLAGGA | Extern | |
| 04 | =N | SPJALLOKEY | Extern | |
| 05 | = | ÖKAFLAGGA | Extern | |
| 06 | JC | <L028> | L028 | |

Bilagor
*

Bilagor

```

0012 :
  00 GET    HALT          Extern  ÅR HALTEN STÖRRE ELLER MIDRE AN BÖR
  01 CMP    BÖR           Extern  1= ÖVER 0 =UNDER
  02 O      GREATER     1673   >
  03 =      NIVÅFLAGGA Extern
  *
0013 :
  00 A      NIVÅFLAGGA Extern  HOPPA OM ÖVER BÖR
  01 JC    <L027>      L027
  *
0014 :
  00 GET    BÖR           Extern  OM INNANFÖR INRE HYSTERES OKEY
  01 SUB    HALT          Extern
  02 CMP    HYSTER       Extern
  03 O      LESS         1671   <
  04 =      SPJALLOKEY   Extern
  05 =N     ÖKAFLAGGA    Extern
  06 =N     MINSKAFLAGGA Extern
  07 JC    <L028>      L028
  *
0015 :
  00 LBL    <L027>      L027
  *
0016 :
  00 GET    HALT          Extern  OM INNANFÖR HYSTERES OKEY
  01 SUB    BÖR           Extern
  02 CMP    HYSTER       Extern
  03 O      LESS         1671   <
  04 =      SPJALLOKEY   Extern
  05 =N     ÖKAFLAGGA    Extern
  06 =N     MINSKAFLAGGA Extern
  07 JC    <L028>      L028
  *
0017 :
  00 A      PULSFLAGGA   Extern  OM HALT ÖVER BÖR SÅ MINSKA SPJÄLL
  01 A      NIVÅFLAGGA   Extern
  02 =      MINSKAFLAGGA Extern
  03 =N     ÖKAFLAGGA    Extern
  04 =N     SPJALLOKEY   Extern
  05 JC    <L026>      L026
  *
0018 :
  00 A      PULSFLAGGA   Extern  OM HALT UNDER SÅ ÖKA PÅ PULS
  01 AN     NIVÅFLAGGA   Extern
  02 =      ÖKAFLAGGA    Extern
  03 =N     MINSKAFLAGGA Extern
  04 =N     SPJALLOKEY   Extern
  05 JC    <L028>      L028
  *
0019 :
  00 AN     PULSFLAGGA   Extern  PULSEN AR ICKE AKTIV,MAN GOR INGET
  01 =N     ÖKAFLAGGA    Extern
  02 =N     MINSKAFLAGGA Extern
  *
0020 :
  00 LBL    <L026>      L026   HÄR SKER SPJALLSPÄRRNING
  *
0021 :
  00 GET    SPJÄLLÄGE    Extern  OM SPJÄLLET AR ÖVER MINLÄGE SÅ HOPPA
  01 CMP    MINLÄGE      Extern
  02 O      GREATER     1673   >
  03 JC    <L028>      L028
  *
0022 :
  00 =N     ÖKAFLAGGA    Extern  MAN GÖR INGENTING EFTERSOM
  01 =N     MINSKAFLAGGA Extern  MAN LIGGER PÅ GRÄNSEN
  02 =      SPJALLOKEY   Extern
  *
0023 :
  00 LBL    <L028>      L028
  *
End Of PBS

```

Bilagor

Module: Pulsregs

| Stamt | Instr | Identifier | Address | Comment |
|--------|-------|--------------|---------|--|
| 0000 : | | | | |
| 00 | = | MINSKAFLAGGA | Extern | BÖRJAR ALLTID MED ATT STÄNGA SPJÄLL |
| 01 | =N | ÖKAFLAGGA | Extern | BEHÖVS SENARE OM MAN HOPPAR |
| | * | | | |
| 0001 : | | | | |
| 00 | A | REGLERSPÄRR | Extern | OM DET ÄR FÖR TIDIGT ATT BÖRJA REGLERA |
| 01 | JC | <L021> | L021 | SÅ HOPPAR MAN |
| | * | | | |
| 0002 : | | | | |
| 00 | A | PULSFLAGGA | Extern | HOPPA PÅ REGULATORNS EGEN |
| 01 | JC | <L017> | L017 | KLOCKPULSFLANK |
| | * | | | |
| 0003 : | | | | |
| 00 | A | PULSE-1HZ | 1677 | ÖKA PAUSTID |
| 01 | EDGE | <1021> | 1021 | |
| 02 | INC | PAUSTID | Extern | |
| | * | | | |
| 0004 : | | | | |
| 00 | GET | PAUSPERIOD | Extern | OM PAUSTID AR SLUT, BORJA PULS |
| 01 | CMP | PAUSTID | Extern | |
| 02 | O | LESS | 1671 | < |
| 03 | = | PULSFLAGGA | Extern | |
| 04 | GET | K0000 | | |
| 05 | STO | PAUSTID | Extern | |
| 06 | STO | PULSTID | Extern | |
| | * | | | |
| 0005 : | | | | |
| 00 | JC | <L018> | L018 | HOPPA FÖRBI PULSNING |
| | * | | | |
| 0006 : | | | | |
| 00 | LBL | <L017> | L017 | PULSNING |
| | * | | | |
| 0007 : | | | | |
| 00 | A | PULSE-1HZ | 1677 | ÖKA PULSTID PÅ FLANK |
| 01 | EDGE | <1022> | 1022 | |
| 02 | INC | PULSTID | Extern | |
| | * | | | |
| 0008 : | | | | |
| 00 | GET | PULSPERIOD | Extern | OM PULSTID SLUT OVERGÅ TILL PAUS |
| 01 | CMP | PULSTID | Extern | |
| 02 | O | LESS | 1671 | < |
| 03 | =N | PULSFLAGGA | Extern | |
| | * | | | |
| 0009 : | | | | |
| 00 | LBL | <L018> | L018 | |
| | * | | | |
| 0010 : | | | | |
| 00 | GET | HALT | Extern | OM HALTEN AR ÖVER ÖVERBÖR SÅ MINSKA |
| 01 | CMP | ÖVREBÖR | Extern | HELA TIDEN OCH HOPPA VIDARE TILL SLUT |
| 02 | O | GREATER | 1673 | > |
| 03 | =N | ÖKAFLAGGA | Extern | |
| 04 | =N | SPJALLOKEY | Extern | |
| 05 | = | MINSKAFLAGGA | Extern | |
| 06 | JC | <L019> | L019 | |
| | * | | | |
| 0011 : | | | | |
| 00 | GET | HALT | Extern | OM HALTEN ÄR UNDER UNDERBÖR SÅ ÖKA |
| 01 | CMP | UNDERBÖR | Extern | HELA TIDEN OCH HOPPA VIDARE |
| 02 | O | LESS | 1671 | < |
| 03 | =N | MINSKAFLAGGA | Extern | |
| 04 | =N | SPJALLOKEY | Extern | |
| 05 | = | ÖKAFLAGGA | Extern | |
| 06 | JC | <L021> | L021 | |
| | * | | | |

Bilagor

```

0012 :
  00 GET   HALT           Extern  ÄR HALTEN STÖRRE ELLER MIDRE AN BÖR
  01 CMP   BÖR           Extern  1= ÖVER 0 =UNDER
  02 O     GREATER       1673   >
  03 =     NIVÅFLAGGA    Extern
  *

0013 :
  00 A     NIVÅFLAGGA    Extern  HOPPA OM ÖVER BÖR
  01 JC    <L020>        L020
  *

0014 :
  00 GET   BÖR           Extern  OM INNANFÖR INRE HYSTERES OKEY
  01 SUB   HALT         Extern
  02 CMP   HYSTER       Extern
  03 O     LESS          1671   <
  04 =     SPJALLOKEY    Extern
  05 =N    ÖKAFLAGGA    Extern
  06 =N    MINSKAFLAGGA Extern
  07 JC    <L019>        L019
  *

0015 :
  00 LBL   <L020>        L020
  *

0016 :
  00 GET   HALT           Extern  OM INNANFÖR HYSTERES OKEY
  01 SUB   BÖR           Extern
  02 CMP   HYSTER       Extern
  03 O     LESS          1671   <
  04 =     SPJALLOKEY    Extern
  05 =N    ÖKAFLAGGA    Extern
  06 =N    MINSKAFLAGGA Extern
  07 JC    <L019>        L019
  *

0017 :
  00 A     PULSFLAGGA    Extern  OM HALT ÖVER BÖR SÅ MINSKA SPJÄLL
  01 A     NIVÅFLAGGA    Extern
  02 =     MINSKAFLAGGA Extern
  03 =N    ÖKAFLAGGA    Extern
  04 =N    SPJALLOKEY    Extern
  05 JC    <L019>        L019
  *

0018 :
  00 A     PULSFLAGGA    Extern  OM HALT UNDER SÅ ÖKA PÅ PULS
  01 AN    NIVÅFLAGGA    Extern
  02 =     ÖKAFLAGGA     Extern
  03 =N    MINSKAFLAGGA Extern
  04 =N    SPJALLOKEY    Extern
  05 JC    <L019>        L019
  *

0019 :
  00 AN    PULSFLAGGA    Extern  PULSEN AR ICKE AKTIV,MAN GOR INGET
  01 =N    ÖKAFLAGGA     Extern
  02 =N    MINSKAFLAGGA Extern
  *

0020 :
  00 LBL   <L019>        L019   HÄR SKER SPJALLSPÄRRNING
  *

0021 :
  00 GET   SPJÄLLÄGE     Extern  OM SPJÄLLET AR ÖVER MINLÄGE SÅ HOPPA
  01 CMP   MINLÄGE       Extern
  02 O     GREATER       1673   >
  03 JC    <L021>        L021
  *

0022 :
  00 =R    MINSKAFLAGGA Extern  MAN LIGGER PÅ GRÄNSEN
  *

0023 :
  00 GET   MINLÄGE       Extern  REGULATORN HAR ETT MINLÄGE I ON MODE
  01 SUB   K0002         SOM JUSTERAS IN HÄR
  02 CMP   SPJÄLLÄGE     Extern
  03 O     LESS          1671   <
  04 JC    <L021>        L021
  *

```

Bilagor

0024 :
00 = ÖKAFLAGGA Extern JUSTERING AV MINLÄGE
01 =N MINSKAFLAGGA Extern
*
0025 :
00 LBL <L021> L021
*
End Of PBS

Bilagor

Module: Pulsregp

| Stmnt | Instr | Identifier | Address | Comment |
|--------|-------|--------------|---------|---------------------------------------|
| 0000 : | | | | |
| 00 | = | MINSKAFLAGGA | Extern | SÄTTER SPJÄLLET PÅ STÄNGNING |
| 01 | =N | ÖKAFLAGGA | Extern | BEHÖVS SENARE I MODULEN |
| | * | | | |
| 0001 : | | | | |
| 00 | A | REGLERSPÄRR | Extern | OM REGLERING EJ HAR BÖRJAT |
| 01 | JC | <L007> | L007 | SÅ HOPPA ÖVER |
| | * | | | |
| 0002 : | | | | |
| 00 | A | PULSFLAGGA | Extern | HOPPA PÅ REGULATORNS EGEN KLOCKFLANK |
| 01 | JC | <L003> | L003 | |
| | * | | | |
| 0003 : | | | | |
| 00 | A | PULSE-1HZ | 1677 | ÖKA PAUSTID |
| 01 | EDGE | <1004> | 1004 | |
| 02 | INC | PAUSTID | Extern | |
| | * | | | |
| 0004 : | | | | |
| 00 | GET | PAUSPERIOD | Extern | OM PAUSTID AR SLUT, BORJA PULS |
| 01 | CMP | PAUSTID | Extern | |
| 02 | O | LESS | 1671 | < |
| 03 | = | PULSFLAGGA | Extern | |
| 04 | GET | K0000 | | |
| 05 | STO | PAUSTID | Extern | |
| 06 | STO | PULSTID | Extern | |
| | * | | | |
| 0005 : | | | | |
| 00 | JC | <L004> | L004 | HOPPA FÖRBI PULSNING |
| | * | | | |
| 0006 : | | | | |
| 00 | LBL | <L003> | L003 | PULSNING |
| | * | | | |
| 0007 : | | | | |
| 00 | A | PULSE-1HZ | 1677 | ÖKA PULSTID PÅ FLANK |
| 01 | EDGE | <1005> | 1005 | |
| 02 | INC | PULSTID | Extern | |
| | * | | | |
| 0008 : | | | | |
| 00 | GET | PULSPERIOD | Extern | OM PULSTID SLUT ÖVERGÅ TILL PAUS |
| 01 | CMP | PULSTID | Extern | |
| 02 | O | LESS | 1671 | < |
| 03 | =N | PULSFLAGGA | Extern | |
| | * | | | |
| 0009 : | | | | |
| 00 | LBL | <L004> | L004 | |
| | * | | | |
| 0010 : | | | | |
| 00 | GET | HALT | Extern | OM HALTEN AR ÖVER ÖVERBÖR SÅ MINSKA |
| 01 | CMP | ÖVERBÖR | Extern | HELA TIDEN OCH HOPPA VIDARE TILL SLUT |
| 02 | O | GREATER | 1673 | > |
| 03 | =N | ÖKAFLAGGA | Extern | |
| 04 | =N | SPJALLOKEY | Extern | |
| 05 | = | MINSKAFLAGGA | Extern | |
| 06 | JC | <L005> | L005 | |
| | * | | | |
| 0011 : | | | | |
| 00 | GET | HALT | Extern | OM HALTEN ÄR UNDER UNDERBÖR SÅ ÖKA |
| 01 | CMP | UNDERBÖR | Extern | HELA TIDEN OCH HOPPA VIDARE |
| 02 | O | LESS | 1671 | < |
| 03 | =N | MINSKAFLAGGA | Extern | |
| 04 | =N | SPJALLOKEY | Extern | |
| 05 | = | ÖKAFLAGGA | Extern | |
| 06 | JC | <L007> | L007 | |
| | * | | | |

Bilagor

0012 :
 00 GET HALT Extern ÄR HALTEN STÖRRE ELLER MIDRE AN BÖR
 01 CMP BÖR Extern 1= ÖVER 0 =UNDER
 02 O GREATER 1673 >
 03 = NIVÅFLAGGA Extern
 *

0013 :
 00 A NIVÅFLAGGA Extern HOPPA OM ÖVER BÖR
 01 JC <L006> L006
 *

0014 :
 00 GET BÖR Extern OM INNANFÖR INRE HYSTERES OKEY
 01 SUB HALT Extern
 02 CMP HYSTER Extern
 03 O LESS 1671 <
 04 = SPJALLOKEY Extern
 05 =N ÖKAFLAGGA Extern
 06 =N MINSKAFLAGGA Extern
 07 JC <L005> L005
 *

0015 :
 00 LBL <L006> L006
 *

0016 :
 00 GET HALT Extern OM INNANFÖR HYSTERES OKEY
 01 SUB BÖR Extern
 02 CMP HYSTER Extern
 03 O LESS 1671 <
 04 = SPJALLOKEY Extern
 05 =N ÖKAFLAGGA Extern
 06 =N MINSKAFLAGGA Extern
 07 JC <L005> L005
 *

0017 :
 00 A PULSFLAGGA Extern OM HALT ÖVER BÖR SÅ MINSKA SPJÄLL
 01 A NIVÅFLAGGA Extern
 02 = MINSKAFLAGGA Extern
 03 =N ÖKAFLAGGA Extern
 04 =N SPJALLOKEY Extern
 05 JC <L005> L005
 *

0018 :
 00 A PULSFLAGGA Extern OM HALT UNDER SÅ ÖKA PÅ PULS
 01 AN NIVÅFLAGGA Extern
 02 = ÖKAFLAGGA Extern
 03 =N MINSKAFLAGGA Extern
 04 =N SPJALLOKEY Extern
 05 JC <L005> L005
 *

0019 :
 00 AN PULSFLAGGA Extern PULSEN AR ICKE AKTIV,MAN GOR INGET
 01 =N ÖKAFLAGGA Extern
 02 =N MINSKAFLAGGA Extern
 *

0020 :
 00 LBL <L005> L005 HÄR SKER SPJALLSPÄRRNING
 *

0021 :
 00 GET SPJÄLLÄGE Extern OM SPJÄLLET AR ÖVER MINLÄGE SÅ HOPPA
 01 CMP MINLÄGE Extern
 02 O GREATER 1673 >
 03 JC <L007> L007
 *

Bilagor

```

0022 :
  00 =R      MINSKAFLAGGA   Extern  MAN LIGGER PÅ GRÄNSEN
    *
0023 :
  00 GET     MINLÄGE        Extern  HÅR JUSTERAS SPJÄLLET IN SÅ ATT
  01 SUB     K0005          Extern  MINLÄGET EFTERFÖLJS
  02 CMP     SPJÄLLÄGE     Extern
  03 O       LESS          1671   <
  04 JC      <L007>        L007
    *
0024 :
  00 =       ÖKAFLAGGA     Extern  JUSTERA MINLÄGET
  01 =N      MINSKAFLAGGA  Extern
    *
0025 :
  00 LBL     <L007>        L007
    *
End Of PBS

```

Module: Pulsregt

| Stmnt | Instr | Identifier | Address | Comment |
|--------|---------|--------------|---------|----------------------------------|
| 0000 : | | | | |
| | 00 = | MINSKAFLAGGA | Extern | SÄTTER SPJÄLLET PÅ STÄNGNING |
| | 01 =N | ÖKAFLAGGA | Extern | BEHÖVS SENARE I MODULEN |
| | | * | | |
| 0001 : | | | | |
| | 00 A | REGLERSPÄRR | Extern | OM REGLERING EJ HAR BÖRJAT |
| | 01 JC | <L038> | L038 | SÅ HOPPA ÖVER |
| | | * | | |
| 0002 : | | | | |
| | 00 A | PULSFLAGGA | Extern | OM PULS AKTIV SÅ HOPPA |
| | 01 JC | <L034> | L034 | |
| | | * | | |
| 0003 : | | | | |
| | 00 A | PULSE-1HZ | 1677 | ÖKA PAUSTID |
| | 01 EDGE | <1023> | 1023 | |
| | 02 INC | PAUSTID | Extern | |
| | | * | | |
| 0004 : | | | | |
| | 00 GET | PAUSPERIOD | Extern | OM PAUSTID AR SLUT, BORJA PULS |
| | 01 CMP | PAUSTID | Extern | |
| | 02 O | LESS | 1671 | < |
| | 03 = | PULSFLAGGA | Extern | |
| | 04 GET | K0000 | | |
| | 05 STO | PAUSTID | Extern | |
| | 06 STO | PULSTID | Extern | |
| | | * | | |
| 0005 : | | | | |
| | 00 JC | <L035> | L035 | HOPPA FÖRBI PULSNING |
| | | * | | |
| 0006 : | | | | |
| | 00 LBL | <L034> | L034 | PULSNING |
| | | * | | |
| 0007 : | | | | |
| | 00 A | PULSE-1HZ | 1677 | ÖKA PULSTID PÅ FLANK |
| | 01 EDGE | <1024> | 1024 | |
| | 02 INC | PULSTID | Extern | |
| | | * | | |
| 0008 : | | | | |
| | 00 GET | PULSPERIOD | Extern | OM PULSTID SLUT ÖVERGÅ TILL PAUS |
| | 01 CMP | PULSTID | Extern | |
| | 02 O | LESS | 1671 | < |
| | 03 =N | PULSFLAGGA | Extern | |
| | | * | | |
| 0009 : | | | | |
| | 00 LBL | <L035> | L035 | |
| | | * | | |

Bilagor

```

0010 :
00 GET    HALT          Extern  OM HALTEN AR ÖVER ÖVERBÖR SÅ MINSKA
01 CMP    ÖVREBÖR        Extern  HELA TIDEN OCH HOPPA VIDARE TILL SLUT
02 O      GREATER    1673   >
03 =      ÖKAFLAGGA  Extern
04 =N     SPJALLOKEY Extern
05 =N     MINSKAFLAGGA Extern
06 JC     <L036>      L036
*

0011 :
00 GET    HALT          Extern  OM HALTEN ÅR UNDER UNDERBÖR SÅ ÖKA
01 CMP    UNDERBÖR   Extern  HELA TIDEN OCH HOPPA VIDARE
02 O      LESS       1671   <
03 =      MINSKAFLAGGA Extern
04 =N     SPJALLOKEY Extern
05 =N     ÖKAFLAGGA  Extern
06 JC     <L036>      L036
*

0012 :
00 GET    HALT          Extern  ÅR HALTEN STÖRRE ELLER MIDRE AN BÖR
01 CMP    BÖR         Extern  I= ÖVER 0 =UNDER
02 O      GREATER    1673   >
03 =      NIVÅFLAGGA Extern
*

0013 :
00 A      NIVÅFLAGGA  Extern  HOPPA OM ÖVER BÖR
01 JC     <L037>      L037
*

0014 :
00 GET    BÖR         Extern  OM INNANFÖR INRE HYSTERES OKEY
01 SUB    HALT        Extern
02 CMP    HYSTER     Extern
03 O      LESS       1671   <
04 =      SPJALLOKEY Extern
05 =N     ÖKAFLAGGA  Extern
06 =N     MINSKAFLAGGA Extern
07 JC     <L036>      L036
*

0015 :
00 LBL    <L037>      L037
*

0016 :
00 GET    HALT          Extern  OM INNANFÖR HYSTERES OKEY
01 SUB    BÖR         Extern
02 CMP    HYSTER     Extern
03 O      LESS       1671   <
04 =      SPJALLOKEY Extern
05 =N     ÖKAFLAGGA  Extern
06 =N     MINSKAFLAGGA Extern
07 JC     <L036>      L036
*

0017 :
00 A      PULSFLAGGA  Extern  OM HALT ÖVER BÖR SÅ MINSKA SPJÄLL
01 A      NIVÅFLAGGA  Extern
02 =N     MINSKAFLAGGA Extern
03 =      ÖKAFLAGGA  Extern
04 =N     SPJALLOKEY Extern
05 JC     <L036>      L036
*

0018 :
00 A      PULSFLAGGA  Extern  OM HALT UNDER SÅ ÖKA PÅ PULS
01 AN     NIVÅFLAGGA  Extern
02 =N     ÖKAFLAGGA  Extern
03 =      MINSKAFLAGGA Extern
04 =N     SPJALLOKEY Extern
05 JC     <L036>      L036
*

0019 :
00 AN     PULSFLAGGA  Extern  PULSEN AR ICKE AKTIV,MAN GOR INGET
01 =N     ÖKAFLAGGA  Extern
02 =N     MINSKAFLAGGA Extern
*

0020 :
00 LBL    <L036>      L036  HÄR SKER SPJALLSPÄRRNING
*

```


Bilagor

```

0021 :
00 GET    SPJÄLLÄGE      Extern  OM SPJÄLLET AR ÖVER MINLÄGE SÅ HOPPA
01 CMP    MINLÄGE        Extern
02 O      GREATER       1673   >
03 JC     <L038>       L038
*
0022 :
00 =R     MINSKAFLAGGA Extern  MAN LIGGER PÅ GRÄNSEN
*
0023 :
00 GET    MINLÄGE      Extern  JUSTERA OM MINLÄGET INTE HÅLLS
01 SUB    K0003
02 CMP    SPJÄLLÄGE    Extern
03 O      LESS         1671   <
04 JC     <L038>       L038
*
0024 :
00 =      ÖKAFLAGGA    Extern  JUSTERA MINLÄGET
01 =N     MINSKAFLAGGA Extern
*
0025 :
00 LBL    <L038>       L038
*
End Of PBS

```

Module: Peldutrp

| Stamt | Instr | Identifier | Address | Comment |
|--------|-------|--------------|---------|--|
| 0000 : | | | | |
| 00 | GET | ELDUTRYCK | Extern | OM ELDSTAD SUNGERTRYCKET UNDERSTIGER |
| 01 | CMP | PRIUNDRY2 | Extern | DENNA NIVÅ SÅ STÄNGER MAN PRIMÄRSPJÄLL |
| 02 | O | GREATER | 1673 | > |
| 03 | JC | <L045> | L045 | |
| | | | | * |
| 0001 : | | | | |
| 00 | GET | ELDUTRYCK | Extern | OM ELDSTAD SUNDERTRUCKET UNDERSTIGER |
| 01 | CMP | PRIUNDRY1 | Extern | DENNA NIVÅ ,SÅ LÅSER MAN PRIMÄRSPJÄLLE |
| 02 | O | GREATER | 1673 | > |
| 03 | JC | <L046> | L046 | |
| | | | | * |
| 0002 : | | | | |
| 00 | =N | ÖPPNAPSPJF | Extern | |
| 01 | = | STÄNGPSPJF | Extern | |
| | | | | * |
| 0003 : | | | | |
| 00 | JC | <L045> | L045 | |
| | | | | * |
| 0004 : | | | | |
| 00 | LBL | <L046> | L046 | |
| | | | | * |
| 0005 : | | | | |
| 00 | =N | ÖPPNAPSPJF | Extern | |
| 01 | =N | STÄNGPSPJF | Extern | |
| | | | | * |
| 0006 : | | | | |
| 00 | LBL | <L045> | L045 | |
| | | | | * |
| 0007 : | | | | |
| 00 | AN | PREGLERSPÄRR | Extern | |
| 01 | JC | <L062> | L062 | |
| | | | | * |
| 0008 : | | | | |
| 00 | = | STÄNGPSPJF | Extern | |
| 01 | =N | ÖPPNAPSPJF | Extern | |
| | | | | * |
| 0009 : | | | | |
| 00 | LBL | <L062> | L062 | |
| | | | | * |
| | | | | End Of PBS |

Bilagor

Module: Pflloff

| Stmnt | Instr | Identifier | Address | Comment |
|--------|-------|--------------|---------|--|
| 0000 : | | | | |
| 00 | GET | K0098 | | BERÄKNA ON/OFF NIVÅERNA FÖR PRIMÄRFLÄK |
| 01 | MUL | PFLOFFTEMP | Extern | MED 2 GRADERS HYSTERES |
| 02 | DIV | K0100 | | |
| 03 | STO | PFLONTEMP | Extern | |
| | * | | | |
| 0001 : | | | | |
| 00 | A | PFLONMODE | Extern | ON-OFF STYRNING AV PRIMÄRFLÄKT |
| 01 | JC | <L040> | L040 | |
| | * | | | |
| 0002 : | | | | |
| 00 | GET | PANNTEMP | Extern | ON-OFF STYRNING AV PRIMÄRFLÄKT |
| 01 | CMP | PFLONTEMP | Extern | |
| 02 | O | LESS | 1671 | < |
| 03 | = | PFLONMODE | Extern | |
| | * | | | |
| 0003 : | | | | |
| 00 | JC | <L041> | L041 | |
| | * | | | |
| 0004 : | | | | |
| 00 | LBL | <L040> | L040 | |
| | * | | | |
| 0005 : | | | | |
| 00 | GET | PANNTEMP | Extern | ON-OFF STYRNING AV PRIMÄRFLÄKT |
| 01 | CMP | PFLOFFTEMP | Extern | |
| 02 | O | GREATER | 1673 | > |
| 03 | =N | PFLONMODE | Extern | |
| | * | | | |
| 0006 : | | | | |
| 00 | LBL | <L041> | L041 | |
| | * | | | |
| 0007 : | | | | |
| 00 | A | PFLONMODE | Extern | |
| 01 | JC | <L042> | L042 | |
| | * | | | |
| 0008 : | | | | |
| 00 | =N | PRIFLÄKTFL | Extern | STÄNG FLÄKT OCH SPJÄLL OM |
| 01 | =N | ÖPPNAPSPJF | Extern | PRIMÄRFLÄKTEN ÄR AVSTÄNGD |
| 02 | = | STÄNGPSPJF | Extern | |
| | * | | | |
| 0009 : | | | | |
| 00 | AN | SINMAT-PÅ-AV | Extern | HÄR BÖRJAR STYRNINGEN AV INMATNINGS- |
| 01 | JC | <L043> | L043 | SKRUVAR OM PRIMÄRFLÄKTEN ÄR AVSTÄNGD |
| | * | | | |
| 0010 : | | | | |
| 00 | A | PULSE-1HZ | 1677 | |
| 01 | EDGE | <1025> | 1025 | |
| 02 | INC | SINMAT-PÅ-T | Extern | |
| | * | | | |
| 0011 : | | | | |
| 00 | = | INMATSFL | Extern | |
| | * | | | |
| 0012 : | | | | |
| 00 | GET | SINMAT-PÅ-T | Extern | |
| 01 | CMP | SINMAT-PÅ-P | Extern | |
| 02 | O | GREATER | 1673 | > |
| 03 | =N | SINMAT-PÅ-AV | Extern | |
| 04 | =N | INMATSFL | Extern | |
| | * | | | |
| 0013 : | | | | |
| 00 | JC | <L044> | L044 | |
| | * | | | |
| 0014 : | | | | |
| 00 | LBL | <L043> | L043 | |
| | * | | | |

Bilagor

0015 :

00 A PULSE-1HZ 1677
01 EDGE <1026> 1026
02 INC SINMAT-PAUS-T Extern

*

0016 :

00 =N INMATSFL Extern

*

0017 :

00 GET SINMAT-PAUS-T Extern
01 CMP SINMAT-PAUS-P Extern
02 O GREATER 1673 >
03 = SINMAT-PÅ-AV Extern
04 = INMATSFL Extern
05 GET K0000
06 STO SINMAT-PÅ-T Extern
07 STO SINMAT-PAUS-T Extern

*

0018 :

00 LBL <L044> L044

*

0019 :

00 LBL <L042> L042 SLUTET

*

0020 :

00 AN PREGLERSPÄRR Extern STÄNG SPJÄLLET OM REGLERSPÄRR
01 JC <L063> L063 RÅDER

*

0021 :

00 = STÄNGPSPJF Extern
01 =N ÖPPNAPSPJF Extern

*

0022 :

00 LBL <L063> L063

*

End Of PBS

Bilagor

Module: Pfilter

| Stamt | Instr | Identifier | Address | Comment |
|--------|------------|--------------|---------|---------------------------------|
| 0000 : | | | | |
| | 00 AN | PFILTREN-ON | Extern | OM FILTERRENING EJ PÅGÅR, HOPPA |
| | 01 JC | <L050> | L050 | |
| | * | | | |
| 0001 : | | | | |
| | 00 GET | CO-HALT | Extern | KOLLA OM CO-HALT ÄR OKAY |
| | 01 CMP | CO-GRÄNS | Extern | |
| | 02 O | GREATER | 1673 | > |
| | 03 =N | CO-OKEY | Extern | |
| | * | | | |
| 0002 : | | | | |
| | 00 A | PULSE-1HZ | 1677 | |
| | 01 EDGE | <1016> | 1016 | |
| | 02 A | CO-OKEY | Extern | |
| | 03 INC | FILTR-PÅ-T | Extern | |
| | * | | | |
| 0003 : | | | | |
| | 00 GET | FILTR-PÅ-T | Extern | |
| | 01 CMP | FILTERRENP | Extern | |
| | 02 O | GREATER | 1673 | > |
| | 03 =N | PFILTREN-ON | Extern | |
| | * | | | |
| 0004 : | | | | |
| | 00 JC | <L051> | L051 | HOPPA FÖRBI FILTERREN I PAUS |
| | * | | | |
| 0005 : | | | | |
| | 00 LBL | <L050> | L050 | |
| | * | | | |
| 0006 : | | | | |
| | 00 A | PULSE-1HZ | 1677 | ÖKA FILTERREN PAUS T |
| | 01 EDGE | <1017> | 1017 | |
| | 02 INC | FILTR-PAUS-T | Extern | |
| | * | | | |
| 0007 : | | | | |
| | 00 GET | FILTR-PAUS-T | Extern | KOLLA OM FILTERPAUS AR SLUT |
| | 01 CMP | FILTERPAUSP | Extern | |
| | 02 O | GREATER | 1673 | > |
| | 03 = | PFILTREN-ON | Extern | |
| | 04 GET | K0000 | | |
| | 05 STO | FILTR-PÅ-T | Extern | |
| | 06 STO | FILTR-PAUS-T | Extern | |
| | * | | | |
| 0008 : | | | | |
| | 00 LBL | <L051> | L051 | |
| | * | | | |
| 0009 : | | | | |
| | 00 A | CO-OKEY | Extern | OM CO-OKEY OCH RENSING, RENSA |
| | 01 A | PFILTREN-ON | Extern | |
| | 02 = | FILTERFLAG | Extern | |
| | 03 = | BYPASSF | Extern | |
| | * | | | |
| | End Of PBS | | | |

Bilagor

Module: Peldutrs

| Stmnt | Instr | Identifier | Address | Comment |
|--------|-------|--------------|---------|--|
| 0000 : | | | | |
| 00 | GET | ELDUTRYCK | Extern | RESTEN AV DENNA MODUL STÄNGER |
| 01 | CMP | SEKUNDRST | Extern | SEKUNDÄRSPJÄLLET OM ELDSTADSUNDERTRYCK |
| 02 | O | GREATER | 1673 | UNDERSTIGER VISS NIVÅ |
| 03 | JC | <L049> | L049 | |
| | * | | | |
| 0001 : | | | | |
| 00 | =N | ÖPPNASSPJF | Extern | |
| 01 | = | STÄNGSSPJF | Extern | |
| | * | | | |
| 0002 : | | | | |
| 00 | LBL | <L049> | L049 | |
| | * | | | |
| 0003 : | | | | |
| 00 | AN | SREGLERSPÄRR | Extern | |
| 01 | JC | <L064> | L064 | |
| | * | | | |
| 0004 : | | | | |
| 00 | = | STÄNGSSPJF | Extern | |
| 01 | =N | ÖPPNASSPJF | Extern | |
| | * | | | |
| 0005 : | | | | |
| 00 | LBL | <L064> | L064 | |
| | * | | | |
| | | End Of PBS | | |

Bilagor

Module: Peldutrt

| Stamt | Instr | Identifier | Address | Comment |
|--------|-------|--------------|---------|-------------------------------------|
| 0000 : | | | | |
| 00 | GET | ELDUTRYCK | Extern | DENNA MODUL STÄNGER TERTIÄRSPJÄLLET |
| 01 | CMP | TERTUNDRST | Extern | OM ELDSTADSUNDERTRYCKET |
| 02 | O | GREATER | 1673 | VISS NIVÅ |
| 03 | JC | <L047> | L047 | |
| | * | | | |
| 0001 : | | | | |
| 00 | =N | ÖPPNATSPJF | Extern | |
| 01 | = | STÄNGTSPJF | Extern | |
| | * | | | |
| 0002 : | | | | |
| 00 | LBL | <L047> | L047 | |
| | * | | | |
| 0003 : | | | | |
| 00 | GET | O2-HALT | Extern | |
| 01 | CMP | TERTOZÖPPNA | Extern | |
| 02 | O | GREATER | 1673 | > |
| 03 | JC | <L048> | L048 | |
| | * | | | |
| 0004 : | | | | |
| 00 | GET | ELDUTRYCK | Extern | |
| 01 | CMP | TERTUNDRST | Extern | |
| 02 | O | LESS | 1671 | < |
| 03 | JC | <L048> | L048 | |
| | * | | | |
| 0005 : | | | | |
| 00 | =N | STÄNGTSPJF | Extern | |
| 01 | = | ÖPPNATSPJF | Extern | |
| | * | | | |
| 0006 : | | | | |
| 00 | LBL | <L048> | L048 | |
| | * | | | |
| 0007 : | | | | |
| 00 | AN | SREGLERSPÄRR | Extern | |
| 01 | JC | <L065> | L065 | |
| | * | | | |
| 0008 : | | | | |
| 00 | = | STÄNGTSPJF | Extern | |
| 01 | =N | ÖPPNATSPJF | Extern | |
| | * | | | |
| 0009 : | | | | |
| 00 | LBL | <L065> | L065 | |
| | * | | | |
| 0010 : | | | | |
| 00 | LBL | <L075> | L075 | |
| | * | | | |

End Of PBS

Bilagor

Module: Pinmat2

| Stmnt | Instr | Identifier | Address | Comment |
|--------|---------|---------------|---------|--------------------------------------|
| 0000 : | 00 =N | INMATBLOCK | Extern | STÄNGER AV INMATNINGEN OM O2 LÅG |
| | * | | | |
| 0001 : | 00 GET | O2-HALT | Extern | STÄNGER AV INMATNING OM O2-HALT |
| | 01 CMP | INMATO2MIN | Extern | ÄR FÖR LÅG |
| | 02 O | GREATER | 1673 | > |
| | 03 JC | <L057> | L057 | |
| | * | | | |
| 0002 : | 00 = | INMATBLOCK | Extern | SE OVAN |
| | * | | | |
| 0003 : | 00 LBL | <L057> | L057 | |
| | * | | | |
| 0004 : | 00 GET | CO-HALT | Extern | STÄNGER AV INMATNINGEN OM O2 FÖR LÅG |
| | 01 CMP | INMATCOMAX1 | Extern | OCH CO FÖR HÖG |
| | 02 O | LESS | 1671 | < |
| | 03 JC | <L058> | L058 | |
| | * | | | |
| 0005 : | 00 GET | O2-HALT | Extern | SE OVAN |
| | 01 CMP | INMATO2MIN1 | Extern | |
| | 02 O | GREATER | 1673 | > |
| | 03 JC | <L058> | L058 | |
| | * | | | |
| 0006 : | 00 = | INMATBLOCK | Extern | SE OVAN |
| | * | | | |
| 0007 : | 00 = | INMATBLOCK | Extern | |
| | * | | | |
| 0008 : | 00 LBL | <L058> | L058 | |
| | * | | | |
| 0009 : | 00 AN | INMATBLOCK | Extern | |
| | 01 JC | <L059> | L059 | |
| | * | | | |
| 0010 : | 00 AN | SINMAT2-PÅ-AV | Extern | OM INMATNINGEN ÄR BLOCKERAD |
| | 01 JC | <L060> | L060 | SÅ MÅSTE MAN ÄNDÅ LÅTA SKRUVARNA GÅ |
| | * | | | |
| 0011 : | 00 A | PULSE-1HZ | 1677 | FÖR ATT UNDVIKA ATT |
| | 01 EDGE | <1027> | 1027 | BRAND SPRIDER SIG BAKÅT |
| | 02 INC | SINMAT2-PÅ-T | Extern | |
| | * | | | |
| 0012 : | 00 = | INMATSFL | Extern | |
| | * | | | |
| 0013 : | 00 GET | SINMAT2-PÅ-T | Extern | DESSA SATSER KÖR SKRUVARNA PERIODVIS |
| | 01 CMP | SINMAT2-PÅ-P | Extern | MELLAN EN VISS PAUSTID |
| | 02 O | GREATER | 1673 | > |
| | 03 =N | SINMAT2-PÅ-AV | Extern | |
| | 04 =N | INMATSFL | Extern | |
| | * | | | |
| 0014 : | 00 JC | <L056> | L056 | |
| | * | | | |
| 0015 : | 00 LBL | <L060> | L060 | |
| | * | | | |

Bilagor

```

0016 :
  00 A      PULSE-1HZ      1677
  01 EDGE  <1030>          1030
  02 INC   SINMAT2-PAS-T  Extern
  *
0017 :
  00 =N     INMATSFL      Extern
  *
0018 :
  00 GET    SINMAT2-PAS-T  Extern  NÄR MAN MATAT IN FÄRDIGT DEN
  01 CMP    SINMAT2-PAS-P  Extern  ANGIVNA TIDEN ,NOLLSTÄLLER MAN
  02 O      GREATER        1673    MAN ALLA PERIODER OCH BÖRJAR RÄKNA OM
  03 =      SINMAT2-PÅ-AV  Extern
  04 =      INMATSFL      Extern
  05 GET    K0000
  06 STO    SINMAT2-PÅ-T  Extern
  07 STO    SINMAT2-PAS-T  Extern
  *
0019 :
  00 LBL    <L056>        L056
  *
0020 :
  00 LBL    <L059>        L059
  *
      End Of PBS
  
```

Module: Roster

| Stmnt | Instr | Identifier | Address | Comment |
|--------|---------|-------------|---------|-------------------------------------|
| 0000 : | | | | |
| | 00 AN | INMATSFL | Extern | OM INMATNING EJ PÅGÅR |
| | 01 JC | <L031> | L031 | HOPPA ÖVER UPPRÄKNING AV TIDEN |
| | | | | * |
| 0001 : | | | | |
| | 00 A | PULSE-1HZ | 1677 | ÖKA TOTAL TID FÖR INMATNINGSSKRUVAR |
| | 01 EDGE | <1014> | 1014 | |
| | 02 INC | RINMATID | Extern | |
| | | | | * |
| 0002 : | | | | |
| | 00 LBL | <L031> | L031 | |
| | | | | * |
| 0003 : | | | | |
| | 00 GET | RINMATID | Extern | OM TOTAL INMATID ÄR ETT VISST VÄRDE |
| | 01 CMP | ROSTERSTYRT | Extern | SÅ SÅTT IGÅNG ROSTERN |
| | 02 O | GREATER | 1673 | > |
| | 03 = | ROSTERPÅFL | Extern | |
| | | | | * |
| 0004 : | | | | |
| | 00 AN | ROSTERPÅFL | Extern | |
| | 01 JC | <L032> | L032 | |
| | | | | * |
| 0005 : | | | | |
| | 00 A | PULSE-1HZ | 1677 | ÖKA GÅNGTID FÖR ROSTER |
| | 01 EDGE | <1015> | 1015 | |
| | 02 INC | ROSTERTID | Extern | |
| | | | | * |
| 0006 : | | | | |
| | 00 GET | ROSTERTID | Extern | SLUTA ROSTERPULS EFTER 2 SEK |
| | 01 CMP | K0006 | | |
| | 02 O | GREATER | 1673 | > |
| | 03 =N | ROSTERPÅFL | Extern | |
| | | | | * |
| 0007 : | | | | |
| | 00 AN | ROSTERPÅFL | Extern | NÄR ROSTER AV NOLLSTÄLL |
| | 01 GET | K0000 | | |
| | 02 STO | RINMATID | Extern | |
| | 03 STO | ROSTERTID | Extern | |
| | | | | * |
| 0008 : | | | | |

Bilagor

00 LBL <L032> L032
 *
 End Of PBS

Module: Askmat

| Stmnt | Instr | Identifier | Address | Comment |
|--------|-------|--------------|---------|--------------------------------------|
| 0000 : | | | | |
| 00 | AN | PÅGÅR-ASKMAT | Extern | OM ASKMATNING INTE PÅGÅR HOPPA |
| 01 | JC | <L010> | L010 | |
| | | * | | |
| 0001 : | | | | |
| 00 | A | PULSE-1HZ | 1677 | ÖKA TIDEN FÖR ASKTÖMNINGS PROCEDUR |
| 01 | EDGE | <1003> | 1003 | |
| 02 | INC | ASKTID | Extern | |
| | | * | | |
| 0002 : | | | | |
| 00 | = | ASK1FLAG | Extern | SÅTT PÅ ASKSKRUV 1 |
| | | * | | |
| 0003 : | | | | |
| 00 | GET | ASKPER1 | Extern | SÅTT PÅ ASKSKRUV2 EFTER TIDSFÖRDRÖJ |
| 01 | CMP | ASKTID | Extern | |
| 02 | O | LESS | 1671 | < |
| 03 | = | ASK2FLAG | Extern | |
| | | * | | |
| 0004 : | | | | |
| 00 | GET | ASKPER2 | Extern | SÅTT PÅ ASKSKRUV3 EFTER TIDSFÖRDRÖJ |
| 01 | CMP | ASKTID | Extern | |
| 02 | O | LESS | 1671 | < |
| 03 | = | ASK3FLAG | Extern | |
| | | * | | |
| 0005 : | | | | |
| 00 | GET | ASKPER3 | Extern | STÄNG AV ALLA ASKSKRUVAR |
| 01 | CMP | ASKTID | Extern | SAMT NOLLSTÄLL ASKTID, TOTAL INMATID |
| 02 | O | LESS | 1671 | < |
| 03 | =N | PÅGÅR-ASKMAT | Extern | |
| 04 | GET | K0000 | | |
| 05 | STO | ASKTID | Extern | |
| 06 | STO | INMATSTID | Extern | |
| | | * | | |
| 0006 : | | | | |
| 00 | LBL | <L010> | L010 | |
| | | * | | |
| 0007 : | | | | |
| 00 | A | ASK1FLAG | Extern | GODKÄN ASKMATNINGS SKRUV1 |
| 01 | A | PÅGÅR-ASKMAT | Extern | |
| 02 | = | ASK1FLAG | Extern | |
| | | * | | |
| 0008 : | | | | |
| 00 | A | ASK2FLAG | Extern | GODKÄN ASKMATNINGS SKRUV2 |
| 01 | A | PÅGÅR-ASKMAT | Extern | |
| 02 | = | ASK2FLAG | Extern | |
| | | * | | |
| 0009 : | | | | |
| 00 | A | ASK3FLAG | Extern | GODKÄN ASKMATNINGS SKRUV3 |
| 01 | A | PÅGÅR-ASKMAT | Extern | |
| 02 | = | ASK3FLAG | Extern | |
| | | * | | |
| 0010 : | | | | |
| 00 | A | PULSE-1HZ | 1677 | |
| 01 | EDGE | <1020> | 1020 | |
| 02 | A | INMATSFL | Extern | ÖKA ÖKA TOTAL INMATNINGS TID 1 SEK |
| 03 | INC | INMATSTID | Extern | |
| | | * | | |
| 0011 : | | | | |
| 00 | GET | INMATSTID | Extern | BÖRJA ASKUTMATNING OM TOTAL INMATID |
| 01 | CMP | ASKMÄNGD | Extern | ÖVERSTIGER ASKMÄNGD MOTSVARANDE |
| 02 | O | GREATER | 1673 | > |

Bilagor

03 =

PÅGÅR-ASKMAT

Extern

*

End Of PBS

Bilagor

Module: Bragg

| Stamt | Instr | Identifier | Address | Comment |
|--------|------------|------------|---------|-------------------------------------|
| 0000 : | | | | |
| | 00 AN | INMATSFL | Extern | RAKNA ANTAL INMATNINGS FLANKER |
| | 01 JC | <L069> | L069 | |
| | * | | | |
| 0001 : | | | | |
| | 00 A | PULSE-1HZ | 1677 | RÅKNA UPP ANTAL SEKUNDER SOM |
| | 01 EDGE | <1010> | 1010 | INMATNINGEN HAR GÅTT |
| | 02 INC | INMATCOUNT | Extern | |
| | * | | | |
| 0002 : | | | | |
| | 00 LBL | <L069> | L069 | |
| | * | | | |
| 0003 : | | | | |
| | 00 GET | INMATCOUNT | Extern | OM BRAGGENTIDEN INTE ÅR SLUT |
| | 01 CMP | BRANSLEAGX | Extern | SÅ FORTSÄTT BRAGGA |
| | 02 O | GREATER | 1673 | > |
| | 03 = | BRANSLEAGF | Extern | |
| | * | | | |
| 0004 : | | | | |
| | 00 AN | BRANSLEAGF | Extern | OM BRAGGNING INTE PÅGÅR HOPPA |
| | 01 JC | <L072> | L072 | |
| | * | | | |
| 0005 : | | | | |
| | 00 A | PULSE-1HZ | 1677 | RÅKNA UPP TIDEN SOM MAN HAR BRAGGAT |
| | 01 EDGE | <1050> | 1050 | |
| | 02 INC | BRAGGTID | Extern | |
| | * | | | |
| 0006 : | | | | |
| | 00 GET | BRAGGTID | Extern | OM BRAGGTIDEN ÅR SLUT, |
| | 01 CMP | K0060 | | SLUTA BRAGGA OCH NOLLSTÅLL |
| | 02 O | GREATER | 1673 | > |
| | 03 =N | BRANSLEAGF | Extern | |
| | * | | | |
| 0007 : | | | | |
| | 00 AN | BRANSLEAGF | Extern | NOLLSTÅLL |
| | 01 GET | K0000 | | |
| | 02 STO | INMATCOUNT | Extern | |
| | 03 STO | BRAGGTID | Extern | |
| | * | | | |
| 0008 : | | | | |
| | 00 LBL | <L072> | L072 | |
| | * | | | |
| | End Of PBS | | | |

Bilagor

Module: Travdroj

| Stmnt | Instr | Identifier | Address | Comment |
|---|-------|---------------|---------|-------------------------|
| 0000 : | | | | |
| Nar ultraljudsgivaren varit tand konstant i Traversmattid sek. ett-stalls matatraversfl | | | | |
| 00 | A | MATATRAVERSFL | | Extern |
| 01 | JC | <L071> | | L071 |
| * | | | | |
| 0001 : | | | | |
| 00 | AN | <0141> | | 0141 |
| 01 | GET | K0000 | | |
| 02 | STO | TRAVERSMATT | | Extern |
| * | | | | |
| 0002 : | | | | |
| 00 | A | PULSE-1HZ | | 1677 |
| 01 | EDGE | <1051> | | 1051 |
| 02 | INC | TRAVERSMATT | | Extern |
| * | | | | |
| 0003 : | | | | |
| 00 | GET | TRAVERSMATT | | Extern |
| 01 | CMP | TRAVERSMATP | | Extern |
| 02 | O | GREATER | | 1673 > |
| 03 | = | MATATRAVERSFL | | Extern |
| 04 | GET | K0000 | | |
| 05 | STO | TRAVERSMATT | | Extern |
| * | | | | |
| 0004 : | | | | |
| 00 | LBL | <L071> | | L071 |
| * | | | | |
| 0005 : | | | | |
| Tidsfordrojning av traversutgangslage signalen | | | | |
| 00 | A | TRAVUTGÅENDE | | Extern TRAVERSCHECKING1 |
| 01 | =S | <1106> | | 1106 |
| * | | | | |
| 0006 : | | | | |
| 00 | A | <1106> | | 1106 |
| 01 | A | <0513> | | 0513 |
| 02 | =R | <1106> | | 1106 |
| * | | | | |
| 0007 : | | | | |
| tidsfordrojning av tralla utgangslage signalen | | | | |
| 00 | A | TRALLAUTG | | Extern |
| 01 | =S | <1107> | | 1107 |
| * | | | | |
| 0008 : | | | | |
| 00 | A | <1107> | | 1107 |
| 01 | A | <0514> | | 0514 |
| 02 | =R | <1107> | | 1107 |
| * | | | | |
| 0009 : | | | | |
| Om 1110 varit ett i 30 sekunder,ett-stalls traverscheck | | | | |
| 00 | A | TRAVERSCHECK | | Extern |
| 01 | JC | <L076> | | L076 |
| * | | | | |
| 0010 : | | | | |
| 00 | AN | <1110> | | 1110 |
| 01 | GET | K0000 | | |
| 02 | STO | TRAVCHECKT | | Extern |
| * | | | | |
| 0011 : | | | | |
| 00 | A | PULSE-1HZ | | 1677 |
| 01 | EDGE | <1037> | | 1037 |
| 02 | INC | TRAVCHECKT | | Extern |
| * | | | | |

Bilagor

0012 :

| | | | | |
|----|-----|--------------|--------|---|
| 00 | GET | TRAVCHECKT | Extern | |
| 01 | CMP | K0050 | | |
| 02 | O | GREATER | 1673 | > |
| 03 | = | TRAVERSCHECK | Extern | |
| 04 | GET | K0000 | | |
| 05 | STO | TRAVCHECKT | Extern | |

*

0013 :

| | | | | |
|----|-----|--------|------|--|
| 00 | LBL | <L076> | L076 | |
|----|-----|--------|------|--|

*

0014 :

Dubbelkontroll av traverslaget vid branslefickan(koor3)

| | | | | |
|----|------|--------|------|--|
| 00 | A | <0501> | 0501 | |
| 01 | EDGE | <1056> | 1056 | |
| 02 | =S | <1113> | 1113 | |

*

0015 :

| | | | | |
|----|------|--------|------|--|
| 00 | A | <0502> | 0502 | |
| 01 | EDGE | <1057> | 1057 | |
| 02 | =S | <1114> | 1114 | |

*

0016 :

| | | | | |
|----|------|----------|--------|--|
| 00 | A | <0501> | 0501 | |
| 01 | EDGE | <1064> | 1064 | |
| 02 | A | <0302> | 0302 | |
| 03 | INC | TRAVLÄGE | Extern | |

*

0017 :

| | | | | |
|----|------|----------|--------|--|
| 00 | A | <0501> | 0501 | |
| 01 | COM | | | |
| 02 | EDGE | <1065> | 1065 | |
| 03 | A | <0301> | 0301 | |
| 04 | DEC | TRAVLÄGE | Extern | |

*

0018 :

| | | | | |
|----|-----|-----------|--------|--|
| 00 | A | <0501> | 0501 | |
| 01 | O | <0513> | 0513 | |
| 02 | RP | | | |
| 03 | AN | <0301> | 0301 | |
| 04 | AN | <0302> | 0302 | |
| 05 | =S | <1105> | 1105 | |
| 06 | =S | <1117> | 1117 | |
| 07 | GET | K0000 | | |
| 08 | STO | NEWX | Extern | |
| 09 | GET | K0003 | | |
| 10 | STO | FLYTTRIKT | Extern | |

*

0019 :

Om man vrider över till manuellt Lage på traversen under traversförflyttning, avslutar man försök till förflyttning och överlämnar kommandot at pc:n

| | | | | |
|----|------|-----------|--------|---------------|
| 00 | A | <0516> | 0516 | trav auto-man |
| 01 | EDGE | <1230> | 1230 | |
| 02 | GET | K0000 | | |
| 03 | STO | NEWX | Extern | |
| 04 | GET | K0003 | | |
| 05 | STO | FLYTTRIKT | Extern | |
| 06 | =S | <1105> | 1105 | |
| 07 | =S | <1117> | 1117 | |
| 08 | =R | <0302> | 0302 | |
| 09 | =R | <0301> | 0301 | |

*

0020 :

Bransleregling. Om framtempen och rokgasttempen understiger gransvardena eller om co.halten overstiger i 300 sekunder ges sattes en bit -braskopa till 1

| | | | | |
|----|-----|----------|--------|---|
| 00 | GET | <R226> | R226 | |
| 01 | CMP | PANNTEMP | Extern | |
| 02 | O | GREATER | 1673 | > |
| 03 | = | MINTEMP | Extern | |

*

Bilagor

```

0021 :
00 GET   ROKGASTEMP   Extern
01 CMP   <R225>      R225
02 O     LESS        1671   <
03 A     MINTEMP    Extern
04 =     FÖRLÅGEFF  Extern
*

0022 :
00 GET   CO-HALT    Extern
01 CMP   <R227>      R227
02 O     GREATER    1673   >
03 =     COÖVER     Extern
*

0023 :
00 AN    COÖVER     Extern
01 =R    KLARAEJCO Extern
*

0024 :
00 A     BRASKOFA   Extern
01 COM
02 EDGE  EDGETRAV2  Extern
03 =R    KLARAEJCO Extern
04 =R    FÖRLÅGEFF Extern
*

0025 :
00 A     KLARAEJCO  Extern
01 JC    <L085>     L085
*

0026 :
00 AN    COÖVER     Extern
01 GET   K0000
02 STO   COCHECKT   Extern
*

0027 :
00 A     PULSE-1HZ  1677
01 EDGE  EDGETRAV1 Extern
02 INC   COCHECKT   Extern
*

0028 :
00 GET   COCHECKT   Extern
01 CMP   K0300
02 O     GREATER    1673   >
03 =     KLARAEJCO Extern
04 GET   K0000
05 STO   COCHECKT   Extern
*

0029 :
00 LBL   <L085>     L085
*

0030 :
00 A     FÖRLÅGEFF  Extern
01 O     KLARAEJCO  Extern
02 =     BRASKOFA   Extern
*

0031 :
Omöjliggör att traversen försöks flyttas i tva riktningar
00 A     <0302>      0302
01 A     <0301>      0301
02 RP
03 =R    <0302>      0302
04 =R    <0301>      0301
*

```

End Of PBS

Bilagor

Module: Effekt

| Stmnt | Instr | Identifier | Address | Comment |
|--------|-------|------------|---------|----------------------------------|
| 0000 : | | | | |
| 00 | A | PULSE-1HZ | 1677 | RÄKNA UPP PERIODTIDEN |
| 01 | EDGE | <1033> | 1033 | |
| 02 | INC | ENERGIPERT | Extern | |
| | * | | | |
| 0001 : | | | | |
| 00 | A | ENERGIPULS | Extern | PÅ ENERGIPULSFLANK |
| 01 | EDGE | <1034> | 1034 | RÄKNA UPP ANTAL PULSER |
| 02 | INC | ANTALEPULS | Extern | |
| | * | | | |
| 0002 : | | | | |
| 00 | GET | ANTALEPULS | Extern | OM ANTAL PULSER ÄR |
| 01 | CMP | K0005 | | 5 SÅ GÖR INGENTING |
| 02 | O | LESS | 1671 | < |
| 03 | JC | <L067> | L067 | |
| | * | | | |
| 0003 : | | | | |
| 00 | GET | ANTALEPULS | Extern | BERÄKNA EFFEKTEN UTIFRÅN |
| 01 | MUL | K3600 | | ANTAL PULSER OCH TIDEN |
| 02 | DIV | ENERGIPERT | Extern | |
| 03 | MUL | K0100 | | |
| 04 | STO | EFFEKT | Extern | |
| | * | | | |
| 0004 : | | | | |
| 00 | GET | K0000 | | NOLLSTÄLL TIDEN OCH ANTAL PULSER |
| 01 | STO | ANTALEPULS | Extern | |
| 02 | STO | ENERGIPERT | Extern | |
| | * | | | |
| 0005 : | | | | |
| 00 | LBL | <L067> | L067 | |
| | * | | | |
| | | | | End Of PBS |

Module: Flöde

| Stmnt | Instr | Identifier | Address | Comment |
|--------|-------|------------|---------|------------------------------------|
| 0000 : | | | | |
| 00 | A | PULSE-1HZ | 1677 | RÄKNA UPP TIDEN FÖR FLÖDESPERIODEN |
| 01 | EDGE | <1035> | 1035 | |
| 02 | INC | FLÖDEPERT | Extern | |
| | * | | | |
| 0001 : | | | | |
| 00 | A | FLÖDESPULS | Extern | RÄKNA UPP ANTAL FLÖDESPULSER |
| 01 | EDGE | <1036> | 1036 | |
| 02 | INC | ANTALEPULS | Extern | |
| | * | | | |
| 0002 : | | | | |
| 00 | GET | ANTALEPULS | Extern | JÄMFÖR ANTAL FLÖDESPULSER |
| 01 | CMP | K0006 | | |
| 02 | O | LESS | 1671 | < |
| 03 | JC | <L068> | L068 | |
| | * | | | |

Bilagor

0003 :

| | | | | |
|----|-----|------------|--------|--|
| 00 | GET | K3600 | | BERÄKNA FLÖDET UTIFRÅN ANATAL PULSER O |
| 01 | DIV | FLÖDEPERT | Extern | TIDEN |
| 02 | MUL | ANTALFPULS | Extern | |
| 03 | STO | FLÖDE | Extern | |

*

0004 :

| | | | | |
|----|-----|------------|--------|--|
| 00 | GET | K0000 | | |
| 01 | STO | ANTALFPULS | Extern | |
| 02 | STO | FLÖDEPERT | Extern | |

*

0005 :

| | | | | |
|----|-----|--------|------|--|
| 00 | LBL | <L068> | L068 | |
|----|-----|--------|------|--|

*

End Of PBS

Bilagor

Module: Larm

| Stmnt | Instr | Identifier | Address | Comment |
|--------|-------|------------|---------|-----------------------------------|
| 0000 : | | | | |
| 00 | AN | <0551> | 0551 | OM CIRKULATIONSPPUMPARNA INTE GÅR |
| 01 | AN | <0550> | 0550 | SÅ STÄNG AV FLÄKTARNA |
| 02 | =R | INMATSFL | Extern | |
| 03 | =R | RÖKFLÄKTFL | Extern | |
| 04 | =R | SEKFLÄKTFL | Extern | |
| 05 | =R | PRIFLÄKTFL | Extern | |
| | | * | | |
| 0001 : | | | | |
| 00 | AN | <0447> | 0447 | UTLÖST MAXTERMOSTAT |
| 01 | =R | RÖKFLÄKTFL | Extern | |
| 02 | =R | PRIFLÄKTFL | Extern | |
| 03 | =R | SEKFLÄKTFL | Extern | |
| 04 | =R | INMATSFL | Extern | |
| | | * | | |
| 0002 : | | | | |
| 00 | AN | <0450> | 0450 | LÅGT PANNVATTENTRYCK |
| 01 | =R | INMATSFL | Extern | |
| 02 | =R | RÖKFLÄKTFL | Extern | |
| 03 | =R | PRIFLÄKTFL | Extern | |
| 04 | =R | SEKFLÄKTFL | Extern | |
| | | * | | |
| 0003 : | | | | |
| 00 | AN | <0451> | 0451 | HÖGT PANNVATTENTRYCK |
| 01 | =R | INMATSFL | Extern | |
| 02 | =R | PRIFLÄKTFL | Extern | |
| 03 | =R | RÖKFLÄKTFL | Extern | |
| 04 | =R | SEKFLÄKTFL | Extern | |
| | | * | | |
| 0004 : | | | | |
| 00 | AN | <0452> | 0452 | LÅG PANNVATTENCIRKULATION |
| 01 | =R | INMATSFL | Extern | |
| 02 | =R | RÖKFLÄKTFL | Extern | |
| 03 | =R | PRIFLÄKTFL | Extern | |
| 04 | =R | SEKFLÄKTFL | Extern | |
| | | * | | |
| 0005 : | | | | |
| 00 | AN | <0453> | 0453 | UTLÖST KATASTROFSKYDD |
| 01 | =R | INMATSFL | Extern | |
| 02 | =R | RÖKFLÄKTFL | Extern | |
| 03 | =R | PRIFLÄKTFL | Extern | |
| 04 | =R | SEKFLÄKTFL | Extern | |
| | | * | | |
| 0006 : | | | | |
| 00 | AN | <0454> | 0454 | UTLÖST ÖVERTRYCKVAKT ELDSTAD |
| 01 | =R | INMATSFL | Extern | |
| 02 | =R | PRIFLÄKTFL | Extern | |
| 03 | =R | SEKFLÄKTFL | Extern | |
| | | * | | |
| 0007 : | | | | |
| 00 | AN | <0455> | 0455 | LÅG RÖKGAASTEMP1 |
| 01 | =R | INMATSFL | Extern | |
| 02 | =R | RÖKFLÄKTFL | Extern | |
| 03 | =R | SEKFLÄKTFL | Extern | |
| 04 | =R | PRIFLÄKTFL | Extern | |
| | | * | | |
| 0008 : | | | | |
| 00 | A | <0540> | 0540 | HÖG RÖKGAASTEMP2 |
| 01 | =R | INMATSFL | Extern | |
| 02 | =R | RÖKFLÄKTFL | Extern | |
| 03 | =R | PRIFLÄKTFL | Extern | |
| 04 | =R | SEKFLÄKTFL | Extern | |
| | | * | | |

Bilagor

```

0009 :
00 A <0150> 0150 HÖGTEMP INMATNING
01 =R INMATSFL Extern
*

0010 :
00 A <0542> 0542 RÖKGASFLÄKT MED TIDSFÖRDRÖJNING
01 = <TI00> TI00
*

0011 :
00 A <T00> T00 RÖKGASFLÄKT MED TIDSFÖRDRÖJNING
01 =R RÖKFLÄKTFL Extern
02 =R SEKFLÄKTFL Extern
03 =R PRIFLÄKTFL Extern
04 =R INMATSFL Extern
*

0012 :
00 A <0543> 0543 STOFT PROBLEM
01 =S BYPASSF Extern
02 =R ASK1FLAG Extern
03 =R ASK2FLAG Extern
04 =R ASK3FLAG Extern
*

0013 :
00 A <0544> 0544 FILTER FICKOR
01 =S BYPASSF Extern
*

0014 :
00 A <0545> 0545 GROVAVSKILJARE
01 =S BYPASSF Extern
*

0015 :
00 A CO-LARM Extern VID HÖG CO SÅ BYPASSA
01 =S BYPASSF Extern
*

0016 :
00 GET ECOGASTEMP Extern
01 CMP K0165
02 O GREATER 1673 >
03 A <0546> 0546 HÖG RÖKGASTEMP3
04 =S BYPASSF Extern
05 =R INMATSFL Extern
06 =R RÖKFLÄKTFL Extern
07 =R SEKFLÄKTFL Extern
08 =R PRIFLÄKTFL Extern
*

0017 :
00 A <0547> 0547 DIFF TRYCK VAKT
01 =S BYPASSF Extern
02 =S FILTERFLAG Extern
*

0018 :
HÖR BÖRJAR DEN DEL SOM FÖRSER OPERTÖRSGRÄNSNITTET
MED REGISTERET SOM TALAR OM DET ÅR ETT LARM, SAMT REGISTERET
SOM TALAR OM VILKET LARM, SAMT ATT OM LARMET INTE ÅR UTSKRIVET
SÅ SKALL DET SKRIVAS UT.
00 A <0440> 0440 REGISTRERA UTLÖST MOTORSKYDD PRIMÄR
01 COM BETYDELSE
02 EDGE <1201> 1201
03 GET K0440
04 STO SENASTLARM Extern
05 =S PRINTLARM Extern
*

0019 :
00 A <0442> 0442 REGISTRERA UTLÖST SPRINKLER
01 COM
02 EDGE <1202> 1202
03 GET K0442
04 STO SENASTLARM Extern
05 =S PRINTLARM Extern
*

```

Bilagor

| | | | | |
|--------|------|------------|--------|--|
| 0020 : | | | | |
| 00 | A | <0443> | 0443 | REGISTERERA LÅG OLVENIVI I HYDRAULAGGR |
| 01 | COM | | | TILL ROSTERN |
| 02 | EDGE | <1203> | 1203 | |
| 03 | GET | K0443 | | |
| 04 | STO | SENASTLARM | Extern | |
| 05 | =S | PRINTLARM | Extern | |
| | * | | | |
| 0021 : | | | | |
| 00 | A | <0444> | 0444 | REGISTERARA HÖG TEMPERATUR ROSTER OLJA |
| 01 | COM | | | |
| 02 | EDGE | <1204> | 1204 | |
| 03 | GET | K0444 | | |
| 04 | STO | SENASTLARM | Extern | |
| 05 | =S | PRINTLARM | Extern | |
| | * | | | |
| 0022 : | | | | |
| 00 | A | <0447> | 0447 | REGISTRERA UTLÖST MAXTERMOSTAT |
| 01 | COM | | | |
| 02 | EDGE | <1205> | 1205 | |
| 03 | GET | K0447 | | |
| 04 | STO | SENASTLARM | Extern | |
| 05 | =S | PRINTLARM | Extern | |
| | * | | | |
| 0023 : | | | | |
| 00 | A | <0450> | 0450 | REGISTRERA LÅGT PANNVATTENTRYCK |
| 01 | COM | | | |
| 02 | EDGE | <1206> | 1206 | |
| 03 | GET | K0450 | | |
| 04 | STO | SENASTLARM | Extern | |
| 05 | =S | PRINTLARM | Extern | |
| | * | | | |
| 0024 : | | | | |
| 00 | A | <0451> | 0451 | REGISTRERA HÖGT PANNVATTENTRYCK |
| 01 | COM | | | |
| 02 | EDGE | <1207> | 1207 | |
| 03 | GET | K0451 | | |
| 04 | STO | SENASTLARM | Extern | |
| 05 | =S | PRINTLARM | Extern | |
| | * | | | |
| 0025 : | | | | |
| 00 | A | <0452> | 0452 | REGISTRERA LÅG PANNVATTENCIRKULATION |
| 01 | COM | | | |
| 02 | EDGE | <1210> | 1210 | |
| 03 | GET | K0452 | | |
| 04 | STO | SENASTLARM | Extern | |
| 05 | =S | PRINTLARM | Extern | |
| | * | | | |
| 0026 : | | | | |
| 00 | A | <0453> | 0453 | REGISTRERA UTLÖST KATASTROFSKYDD |
| 01 | COM | | | |
| 02 | EDGE | <1211> | 1211 | |
| 03 | GET | K0453 | | |
| 04 | STO | SENASTLARM | Extern | |
| 05 | =S | PRINTLARM | Extern | |
| | * | | | |
| 0027 : | | | | |
| 00 | A | <0454> | 0454 | REGISTRERA UTLÖST ÖVERTRYCKVAKT ELDSTA |
| 01 | COM | | | |
| 02 | EDGE | <1212> | 1212 | |
| 03 | GET | K0454 | | |
| 04 | STO | SENASTLARM | Extern | |
| 05 | =S | PRINTLARM | Extern | |
| | * | | | |
| 0028 : | | | | |
| 00 | A | <0455> | 0455 | REGISTRERA LÅG RÖKGASTEMPERATUR 1 |
| 01 | COM | | | |
| 02 | EDGE | <1213> | 1213 | |
| 03 | GET | K0455 | | |
| 04 | STO | SENASTLARM | Extern | |
| 05 | =S | PRINTLARM | Extern | |
| | * | | | |

Bilagor

| | | | | |
|--------|------|------------|--------|--|
| 0029 : | | | | |
| 00 | A | <0456> | 0456 | REGISTRERA HÖG RÖKGASTEMPERATUR 1 |
| 01 | COM | | | |
| 02 | EDGE | <1214> | 1214 | |
| 03 | GET | K0456 | | |
| 04 | STO | SENASTLARM | Extern | |
| 05 | =S | PRINTLARM | Extern | |
| | * | | | |
| 0030 : | | | | |
| 00 | A | <0142> | 0142 | REGISTRERA LÅGT LUFTRYCK KOMPRESSOR |
| 01 | EDGE | <1215> | 1215 | |
| 02 | GET | K0142 | | |
| 03 | STO | SENASTLARM | Extern | |
| 04 | =S | PRINTLARM | Extern | |
| | * | | | |
| 0031 : | | | | |
| 00 | A | <0143> | 0143 | REGISTRERA UTLÖST VAKT INMATNING |
| 01 | EDGE | <1216> | 1216 | |
| 02 | GET | K0143 | | |
| 03 | STO | SENASTLARM | Extern | |
| 04 | =S | PRINTLARM | Extern | |
| | * | | | |
| 0032 : | | | | |
| 00 | A | <0457> | 0457 | REGISTRERA UTLÖST MOTORSKYDD PRIMÄR BE |
| 01 | EDGE | <1217> | 1217 | |
| 02 | GET | K0457 | | |
| 03 | STO | SENASTLARM | Extern | |
| 04 | =S | PRINTLARM | Extern | |
| | * | | | |
| 0033 : | | | | |
| 00 | A | <0540> | 0540 | REGISTRERA HÖG RÖKGASTEMPERATUR 2 |
| 01 | EDGE | <1221> | 1221 | |
| 02 | GET | K0540 | | |
| 03 | STO | SENASTLARM | Extern | |
| 04 | =S | PRINTLARM | Extern | |
| | * | | | |
| 0034 : | | | | |
| 00 | A | <0150> | 0150 | REGISTRERA HÖG TEMPERATUR INMATNING |
| 01 | EDGE | <1223> | 1223 | |
| 02 | GET | K0150 | | |
| 03 | STO | SENASTLARM | Extern | |
| 04 | =S | PRINTLARM | Extern | |
| | * | | | |
| 0035 : | | | | |
| 00 | A | <0541> | 0541 | REGISTRERA UTLÖST SLAGSKRUVSKYDD |
| 01 | EDGE | <1224> | 1224 | |
| 02 | GET | K0541 | | |
| 03 | STO | SENASTLARM | Extern | |
| 04 | =S | PRINTLARM | Extern | |
| | * | | | |
| 0036 : | | | | |
| 00 | A | <0145> | 0145 | REGISTRERA LARM KULVERT |
| 01 | EDGE | <1225> | 1225 | |
| 02 | GET | K0145 | | |
| 03 | STO | SENASTLARM | Extern | |
| 04 | =S | PRINTLARM | Extern | |
| | * | | | |
| 0037 : | | | | |
| 00 | A | <0146> | 0146 | REGISTRERA LARM VATTENRENING |
| 01 | EDGE | <1226> | 1226 | |
| 02 | GET | K0146 | | |
| 03 | STO | SENASTLARM | Extern | |
| 04 | =S | PRINTLARM | Extern | |
| | * | | | |
| 0038 : | | | | |
| 00 | A | <0147> | 0147 | REGISTRERA FLÄKTAGGREGAT VINDEN |
| 01 | EDGE | <1227> | 1227 | |
| 02 | GET | K0147 | | |
| 03 | STO | SENASTLARM | Extern | |
| 04 | =S | PRINTLARM | Extern | |
| | * | | | |

Bilagor

0039 :

| | | | | |
|----|------|------------|--------|-----------------|
| 00 | A | <0144> | 0144 | REGISTRERA TOPP |
| 01 | EDGE | <1231> | 1231 | |
| 02 | GET | K0144 | | |
| 03 | STO | SENASTLARM | Extern | |
| 04 | =S | PRINTLARM | Extern | |

*

0040 :

| | | | | |
|----|------|------------|--------|-------------------------|
| 00 | A | <0543> | 0543 | REGISTRERA STOFTPROBLEM |
| 01 | EDGE | <1232> | 1232 | |
| 02 | GET | K0543 | | |
| 03 | STO | SENASTLARM | Extern | |
| 04 | =S | PRINTLARM | Extern | |

*

0041 :

| | | | | |
|----|------|------------|--------|-------------------------------|
| 00 | A | <0544> | 0544 | REGISTRERA FILTER FICKOR LARM |
| 01 | EDGE | <1233> | 1233 | |
| 02 | GET | K0544 | | |
| 03 | STO | SENASTLARM | Extern | |
| 04 | =S | PRINTLARM | Extern | |

*

0042 :

| | | | | |
|----|------|------------|--------|--------------------------------|
| 00 | A | <0545> | 0545 | REGISTRERA LARM GROVAVSKILJARE |
| 01 | EDGE | <1234> | 1234 | |
| 02 | GET | K0545 | | |
| 03 | STO | SENASTLARM | Extern | |
| 04 | =S | PRINTLARM | Extern | |

*

0043 :

| | | | | |
|----|------|------------|--------|----------------------------------|
| 00 | A | <0546> | 0546 | REGISTRERA HÖGRÖKGASTEMPERATUR 3 |
| 01 | EDGE | <1235> | 1235 | |
| 02 | GET | K0546 | | |
| 03 | STO | SENASTLARM | Extern | |
| 04 | =S | PRINTLARM | Extern | |

*

0044 :

| | | | | |
|----|------|------------|--------|--------------------------|
| 00 | A | <0547> | 0547 | REGISTRERA DIFFTRYCKVAKT |
| 01 | EDGE | <1236> | 1236 | |
| 02 | GET | K0547 | | |
| 03 | STO | SENASTLARM | Extern | |
| 04 | =S | PRINTLARM | Extern | |

*

0045 :

| | | | | |
|----|----|------------|--------|-------------------------------------|
| 00 | AN | PRIFLÄKTFL | Extern | OM PRIMÄRFLÄKT INTE PÅ STÄNG SPJÄLL |
| 01 | =S | STÄNGPSPJF | Extern | |
| 02 | =R | ÖPPNAPSPJF | Extern | |

*

0046 :

| | | | | |
|----|----|------------|--------|------------------------------|
| 00 | AN | RÖKFLÄKTFL | Extern | OM RÖKGASFLÄKTEN INTE GÅR SÅ |
| 01 | =S | STÄNGSSPJF | Extern | FÖRREGLAS FÖLJANDE |
| 02 | =S | STÄNGTSPJF | Extern | |
| 03 | =S | ÖPPNARSPJF | Extern | |
| 04 | =S | STÄNGPSPJF | Extern | |
| 05 | =R | ÖPPNAPSPJF | Extern | |
| 06 | =R | ÖPPNASSPJF | Extern | |
| 07 | =R | ÖPPNATSPJF | Extern | |
| 08 | =R | STÄNGRSPJF | Extern | |

*

0047 :

| | | | | |
|----|----|--------|------|------------|
| 00 | AN | <0440> | 0440 | SUMMA LARM |
| 01 | ON | <0442> | 0442 | |
| 02 | ON | <0443> | 0443 | |
| 03 | ON | <0444> | 0444 | |
| 04 | ON | <0447> | 0447 | |
| 05 | ON | <0450> | 0450 | |
| 06 | ON | <0451> | 0451 | |
| 07 | ON | <0452> | 0452 | |
| 08 | ON | <0453> | 0453 | |
| 09 | ON | <0454> | 0454 | |
| 10 | ON | <0455> | 0455 | |
| 11 | ON | <0456> | 0456 | |
| 12 | O | <0142> | 0142 | |
| 13 | O | <0143> | 0143 | |
| 14 | O | <0457> | 0457 | |
| 15 | = | <0650> | 0650 | |

*

Bilagor

0048 :

| | | | | |
|----|---|--------|------|------------|
| 00 | A | <0540> | 0540 | summa larm |
| 01 | O | <0650> | 0650 | |
| 02 | O | <0150> | 0150 | |
| 03 | O | <0541> | 0541 | |
| 04 | O | <0145> | 0145 | |
| 05 | O | <0146> | 0146 | |
| 06 | O | <0147> | 0147 | |
| 07 | O | <0542> | 0542 | |
| 08 | O | <0144> | 0144 | |
| 09 | O | <0543> | 0543 | |
| 10 | O | <0544> | 0544 | |
| 11 | O | <0545> | 0545 | |
| 12 | O | <0546> | 0546 | |
| 13 | O | <0547> | 0547 | |
| 14 | = | <0650> | 0650 | |

*

0049 :

| | | | | |
|----|----|------------|--------|---------------------------------|
| 00 | A | ON-OFF | Extern | UNDER UPFSTART SÅ FÖRREGLAR MAN |
| 01 | A | UPPSTART | Extern | EFFEKTFÖRBRUKANDE OBJEKT |
| 02 | =R | INMATSFL | Extern | |
| 03 | =R | ROSTERPÅFL | Extern | |
| 04 | =R | BRANSLEAGF | Extern | |
| 05 | =R | ASK1FLAG | Extern | |
| 06 | =R | ASK2FLAG | Extern | |
| 07 | =R | ASK3FLAG | Extern | |

*

End Of FBS

Bilagor

Module: Skåpval

| Stmnt | Instr | Identifier | Address | Comment |
|--------|--------|-------------|---------|--|
| 0000 : | | | | |
| | 00 A | UPPSTART | Extern | UPPSTART SKALL GE BYPASSNING |
| | 01 =S | BYPASSF | Extern | |
| | | * | | |
| 0001 : | | | | |
| | 00 A | PCSTYR-ON | Extern | OM PCSTYRNINGEN ÄR PÅ |
| | 01 JC | <L073> | L073 | HOPPA ÖVER |
| | | * | | |
| 0002 : | | | | |
| | 00 =N | INMATSFL | Extern | OM PCSTYRNINGEN INTE ÄR PÅ |
| | 01 =N | ROSTERPÅFL | Extern | SÅ NOLLSTÄLL ALLA UTGÅNGAR |
| | 02 =N | BRANSLEAGF | Extern | |
| | 03 = | BYPASSF | Extern | |
| | 04 =N | FILTERFLAG | Extern | |
| | 05 =N | LJUDESOTF | Extern | |
| | 06 =N | RÖKFLÄKTFL | Extern | |
| | 07 =N | SEKFLÄKTFL | Extern | |
| | 08 =N | PRIFLÄKTFL | Extern | |
| | 09 =N | ASK1FLAG | Extern | |
| | 10 =N | ASK2FLAG | Extern | |
| | 11 =N | ASK3FLAG | Extern | |
| | 12 =N | ÖPPNARSPJF | Extern | |
| | 13 =N | STÅNGRSPJF | Extern | |
| | 14 =N | ÖPPNASSPJF | Extern | |
| | | * | | |
| 0003 : | | | | |
| | 00 =N | STÅNGSSPJF | Extern | |
| | 01 =N | ÖPPNASSPJF | Extern | |
| | 02 =N | STÅNGRSPJF | Extern | |
| | 03 =N | ÖPPNATSPJF | Extern | |
| | 04 =N | STÅNGTSPJF | Extern | |
| | | * | | |
| 0004 : | | | | |
| | 00 LBL | <L073> | L073 | |
| | | * | | |
| 0005 : | | | | |
| | 00 AN | RÖKFLÄKTFL | Extern | SÄTTER DE GLOBALA PARAMETRARN |
| | 01 GET | K0000 | | UPPSTART OCH STÅNGNING SOM PÅVERKAR HE |
| | 02 STO | ONTID | Extern | PANNPROGRAMMET VID ÖVERGÅNG FRÅN GAMMA |
| | 03 = | UPPSTART | Extern | TILL PC-SYSTEMET MÅSTE MAN ALLTID |
| | 04 = | STANGNING | Extern | STARTA UPP PÅ NYTT |
| | | * | | |
| 0006 : | | | | |
| | 00 AN | BLOCKROST | Extern | STRÖMSTÄLLARE PÅ SKÅPDÖRREN SOM |
| | 01 =R | ROSTERPÅFL | Extern | STÅNGER AV ROSTER |
| | | * | | |
| 0007 : | | | | |
| | 00 AN | BLOCKINMAT | Extern | STRÖMSTÄLLARE PÅ SKÅPET SOM |
| | 01 =R | INMATSFL | Extern | STÅNGE AV INMATNINGEN |
| | 02 =R | BRANSLEAGF | Extern | |
| | | * | | |
| 0008 : | | | | |
| | 00 AN | BLOCKASK | Extern | STRÖMSTÄLLARE SOM STÅNGER |
| | 01 =R | ASK1FLAG | Extern | AV ASKUMATNINGESKRUVARNA |
| | 02 =R | ASK2FLAG | Extern | |
| | 03 =R | ASK3FLAG | Extern | |
| | | * | | |
| 0009 : | | | | |
| | 00 AN | BLOCKFILT | Extern | STRÖMSTÄLLARE PÅ SKÅPET SOM |
| | 01 =S | BYPASSF | Extern | STÅNGER AV FILTERRENING |
| | 02 =R | FILTERFLAG | Extern | |
| | | * | | |
| 0010 : | | | | |
| | 00 AN | BLOCKLJUDES | Extern | STRÖMSTÄLLARE SOM STÅNGER AV |

Bilagor

01 =R

LJUDSOTF

Extern LJUDSOTNINGEN

*

End Of PBS

Bilagor

Module: Final

| Stamt | Instr | Identifier | Address | Comment |
|-------|-------|------------|---------------|---|
| 0000 | : | | | |
| | 00 | A | INMATSFL | Extern DENNA MODUL FÖR ÖVER ALLA STATUS FLAGG |
| | 01 | = | INMATNING-DUT | Extern PÅ DE VERKLIGA UTGÅNGARNA |
| | | * | | |
| 0001 | : | | | |
| | 00 | A | ROSTERPÅFL | Extern |
| | 01 | = | ROSTER-DUT | Extern |
| | | * | | |
| 0002 | : | | | |
| | 00 | A | BRANSLEAGF | Extern |
| | 01 | = | BRAGG-DUT | Extern |
| | | * | | |
| 0003 | : | | | |
| | 00 | A | BYPASSF | Extern |
| | 01 | =N | BYPASS-DUT | Extern |
| | | * | | |
| 0004 | : | | | |
| | 00 | A | FILTERFLAG | Extern |
| | 01 | = | FILTER-DUT | Extern |
| | | * | | |
| 0005 | : | | | |
| | 00 | A | LJUDSOTF | Extern |
| | 01 | = | LJUDSOT-DUT | Extern |
| | | * | | |
| 0006 | : | | | |
| | 00 | A | RÖKFLÄKTFL | Extern |
| | 01 | = | RÖKFLÄKT-DUT | Extern |
| | | * | | |
| 0007 | : | | | |
| | 00 | A | SEKFLÄKTFL | Extern |
| | 01 | = | SEKFLÄKT-DUT | Extern |
| | 02 | = | TERFLÄKT-DUT | Extern |
| | | * | | |
| 0008 | : | | | |
| | 00 | A | PRIFLÄKTFL | Extern |
| | 01 | = | PRIFLÄKT-DUT | Extern |
| | | * | | |
| 0009 | : | | | |
| | 00 | A | ASK1FLAG | Extern |
| | 01 | = | SLAGGCON-DUT | Extern |
| | | * | | |
| 0010 | : | | | |
| | 00 | A | ASK2FLAG | Extern |
| | 01 | = | SLAGGLIGG-DUT | Extern |
| | 02 | = | SLAGGSKR-DUT | Extern |
| | | * | | |
| 0011 | : | | | |
| | 00 | A | ASK3FLAG | Extern |
| | 01 | = | ASKSKRUV-DUT | Extern |
| | | * | | |
| 0012 | : | | | |
| | 00 | A | ÖPPNARSFJF | Extern |
| | 01 | = | ÖPPNARSFJ-DUT | Extern |
| | | * | | |
| 0013 | : | | | |
| | 00 | A | STÄNGRSFJF | Extern |
| | 01 | = | STÄNGRSFJ-DUT | Extern |
| | | * | | |
| 0014 | : | | | |
| | 00 | A | ÖPPNASSFJF | Extern |
| | 01 | = | ÖPPNASSFJ-DUT | Extern |
| | | * | | |
| 0015 | : | | | |
| | 00 | A | STÄNGSSFJF | Extern |
| | 01 | = | STÄNGSSFJ-DUT | Extern |
| | | * | | |

Bilagor

0016 :

00 A ÖPPNAPSPJF Extern

01 = ÖPPNAPSPJ-DUT Extern

*

0017 :

00 A STÄNGPSPJF Extern

01 = STÄNGPSPJ-DUT Extern

*

0018 :

00 A ÖPPNATSPJF Extern

01 = ÖPPNATSPJ-DUT Extern

*

0019 :

00 A STÄNGTSPJF Extern

01 = STÄNGTSPJ-DUT Extern

*

End Of PBS

Bilagor

Module: Flytrav

| Stmnt | Instr | Identifier | Address | Comment |
|--------|-------|-------------|---------|-----------------------------|
| 0000 : | | | | |
| 00 | A | TRAVPLUS | Extern | flyttar traversen till newx |
| 01 | EDGE | <1237> | 1237 | från traversläge |
| 02 | A | <0302> | 0302 | |
| 03 | INC | TRAVERSLÄGE | Extern | |
| | * | | | |
| 0001 : | | | | |
| 00 | A | TRAVMIN | Extern | |
| 01 | EDGE | <1240> | 1240 | |
| 02 | A | <0301> | 0301 | |
| 03 | DEC | TRAVERSLÄGE | Extern | |
| | * | | | |
| 0002 : | | | | |
| 00 | GET | NEWX | Extern | |
| 01 | CMP | K0000 | | |
| 02 | JC | <L088> | L088 | |
| | * | | | |
| 0003 : | | | | |
| 00 | NPW | | | |
| | * | | | |
| 0004 : | | | | |
| 00 | GET | NEWX | Extern | |
| 01 | CMP | TRAVERSLÄGE | Extern | |
| 02 | JC | <L089> | L089 | |
| | * | | | |
| 0005 : | | | | |
| 00 | GET | NEWX | Extern | |
| 01 | CMP | TRAVERSLÄGE | Extern | |
| 02 | O | GREATER | 1673 | > |
| 03 | GET | K0001 | | |
| 04 | STO | FLYTTRIKT | Extern | |
| 05 | JC | <L089> | L089 | |
| | * | | | |
| 0006 : | | | | |
| 00 | GET | K0000 | | |
| 01 | STO | FLYTTRIKT | Extern | |
| | * | | | |
| 0007 : | | | | |
| 00 | LBL | <L089> | L089 | |
| | * | | | |
| 0008 : | | | | |
| 00 | GET | FLYTTRIKT | Extern | |
| 01 | CMP | K0001 | | |
| 02 | COM | | | |
| 03 | JC | <L087> | L087 | |
| | * | | | |
| 0009 : | | | | |
| 00 | =S | <0302> | 0302 | |
| | * | | | |
| 0010 : | | | | |
| 00 | GET | NEWX | Extern | |
| 01 | SUB | TRAVERSLÄGE | Extern | |
| 02 | RP | | | |
| 03 | CMP | K0002 | | |
| 04 | O | GREATER | 1673 | > |
| 05 | = | <TI01> | TI01 | |
| | * | | | |
| 0011 : | | | | |
| 00 | A | <T01> | T01 | |
| 01 | =S | <0303> | 0303 | |
| | * | | | |

Bilagor

```

0012 :
00 GET    NEWX          Extern
01 CMP    TRAVERSLÄGE Extern
02 COM
03 JC     <L087>       L087
*
0013 :
00 =R     <0303>       0303
*
0014 :
00 A      STOPGIVTRAV Extern
01 =R     <0302>       0302
02 =S     BIT          Extern
03 GET    K0000
04 STO    NEWX         Extern
05 GET    K0003
06 STO    FLYTTRIKT   Extern
*
0015 :
00 JC     <L088>       L088
*
0016 :
00 LBL    <L087>       L087
*
0017 :
00 GET    FLYTTRIKT   Extern
01 CMP    K0000
02 COM
03 JC     <L088>       L088
*
0018 :
00 =S     <0301>       0301
*
0019 :
00 GET    TRAVERSLÄGE Extern
01 SUB    NEWX         Extern
02 RP
03 CMP    K0002
04 O      GREATER     1673  >
05 =      <TI02>      TI02
*
0020 :
00 A      <T02>        T02
01 =S     <0303>       0303
*
0021 :
00 GET    TRAVERSLÄGE Extern
01 CMP    NEWX         Extern
02 COM
03 JC     <L088>       L088
*
0022 :
00 =R     <0303>       0303
*
0023 :
00 A      STOPGIVTRAV Extern
01 =R     <0301>       0301
02 =S     BIT          Extern
03 GET    K0000
04 STO    NEWX         Extern
05 GET    K0003
06 STO    FLYTTRIKT   Extern
*
0024 :
00 LBL    <L088>       L088
*
End Of PBS

```

Bilagor

Module: Comintop

| Stmnt | Instr | Identifier | Address | Comment |
|--------|-------|------------|---------|------------------------------------|
| 0000 : | | | | |
| 00 | AN | ON-OFF | Extern | Om Pannan är OFF |
| 01 | =N | <TI03> | TI03 | nollställ timern |
| | * | | | |
| 0001 : | | | | |
| 00 | A | ON-OFF | Extern | Om Pannan är ON |
| 01 | = | <TI03> | TI03 | starta timern |
| | * | | | |
| 0002 : | | | | |
| 00 | AN | <T03> | T03 | Om 5min inte har gått |
| 01 | A | ON-OFF | Extern | Och Pannan är ON |
| 02 | JC | <L099> | L099 | hoppa över uppdateringen |
| | * | | | |
| 0003 : | | | | |
| 00 | GET | <AI660> | AI660 | Uppdatering av CO till Mintopprog. |
| 01 | STO | <AO630> | AO630 | |
| | * | | | |
| 0004 : | | | | |
| 00 | LBL | <L099> | L099 | |
| | * | | | |
| | | | | End Of PBS |

Bilagor

■Addresslista

Denna lista anger alla de I/O, register och tidkanaler som har använts i SattProg

| Address | Identifier | | |
|---------|---------------|------|---------------|
| 0100 | ROSTER-DUT | 0101 | LJUDSOT-DUT |
| 0102 | PRIFLÄKT-DUT | 0103 | SEKFLÄKT-DUT |
| 0140 | BRFICKAFULL | 0141 | BRFICKATOM |
| 0142 | KOMPRESSOR | 0143 | INMATVAKT |
| 0144 | TOPPLARM | 0145 | KULVERT |
| 0146 | VATTENREN | 0147 | FLÄKTVIND |
| 0150 | TEMPINMAT | 0152 | FLÖDESPULS |
| 0153 | ENERGIPULS | 0300 | BYPASS-DUT |
| 0301 | TRAVERS- | 0302 | TRAVERS+ |
| 0303 | TRAVHÖGFART | 0304 | TRALLA- |
| 0305 | TRALLA+ | 0306 | TRALLHÖGFART |
| 0307 | LYFTSKOPA | 0310 | SÄNKSKOPA |
| 0311 | LYFTSNABB | 0312 | ÖPPNASKOPA |
| 0313 | STÄNGSKOPA | 0314 | INMATNING-DUT |
| 0315 | BRAGG-DUT | 0316 | TERFLÄKT-DUT |
| 0317 | RÖKFLÄKT-DUT | 0340 | FILTER-DUT |
| 0341 | ASKSKRUV-DUT | 0344 | ÖPPNAPSPJ-DUT |
| 0345 | STÄNGPSPJ-DUT | 0346 | ÖPPNASSPJ-DUT |
| 0347 | STÄNGSSPJ-DUT | 0350 | ÖPPNATSPJ-DUT |
| 0351 | STÄNGTSPJ-DUT | 0352 | SLAGGCON-DUT |
| 0353 | SLAGGLIGG-DUT | 0354 | SLAGGSKR-DUT |
| 0356 | ÖPPNARSPJ-DUT | 0357 | STÄNGRSPJ-DUT |
| 0420 | RPULSFLAGGA | 0421 | ÖPPNARSPJF |
| 0422 | STÄNGRSPJF | 0423 | RSPJÄLLOKEY |
| 0424 | RNIVÅFLAGGA | 0425 | RREGLERSPÄRR |
| 0426 | PPULSFLAGGA | 0427 | ÖPPNAPSPJF |
| 0430 | STÄNGPSPJF | 0431 | PSPJÄLLOKEY |
| 0432 | PNIVÅFLAGGA | 0433 | PREGLERSPÄRR |
| 0434 | SPULSFLAGGA | 0435 | ÖPPNASSPJF |
| 0436 | STÄNGSSPJF | 0437 | MÅTSKAL |
| 0440 | MOTSKYPRI | 0442 | SPRINKLER |
| 0443 | OLJAROSTER | 0444 | TEMPROSTER |
| 0446 | NEDELTERM | 0447 | MAXTERMOSTAT |
| 0450 | LÅGPANNTRYCK | 0451 | HÖGPANNTRYCK |
| 0452 | LÅGCIRKULAT | 0453 | KATASTROF |
| 0454 | ÖVERTRELD | 0455 | LÅGR1 |
| 0456 | HÖGR1 | 0457 | MOTSKYSEK |
| 0460 | SNIVÅFLAGGA | 0461 | SREGLERSPÄRR |
| 0462 | LJUDSOTF | 0463 | RÖKFLÄKTFL |
| 0464 | SEKFLÄKTFL | 0465 | PRIFLÄKTFL |
| 0466 | AINMAT-PÅ-AV | 0467 | INMATSFL |
| 0470 | BRANSLEAGF | 0471 | FILTERFLAG |
| 0472 | BYPASSF | 0473 | PLJUDS-PA-AV |
| 0474 | PÅGÅR-ASKMAT | 0475 | ASK1FLAG |
| 0476 | ASK2FLAG | 0477 | ASK3FLAG |
| 0500 | TRAVUTGÄENDE | 0501 | TRAVFLUS |
| 0502 | STOPGIVTRAV | 0503 | TRALLAUTG |
| 0504 | TRALLAMIN | 0505 | STOPPGTRALLA |
| 0506 | NIVÅS21 | 0507 | NIVÅS22 |
| 0510 | NIVÅS23 | 0511 | NIVÅS24 |
| 0512 | NOLLASTTRAV | 0513 | TRAVMIN |
| 0514 | TRALLAPLUS | 0515 | PCSTYR-ON |
| 0516 | PRISPSTANGD | 0517 | RÖKSPSTANGD |
| 0520 | STANGNING | 0521 | UPPSTART |
| 0522 | PINMAT-PÅ-AV | 0523 | ROSTERPÅFL |
| 0524 | UNDER-ON-T | 0525 | UNDER-OFF-T |
| 0526 | ON-OFF | 0527 | PFILTREN-ON |
| 0530 | CO-OKEY | 0531 | FÖRLÅGEFF |
| 0532 | COÖVER | 0533 | KLARAEJCO |
| 0534 | EDGE TRAV1 | 0535 | EDGE TRAV2 |
| 0536 | BRAGGON | 0537 | BYPASSAUTO |
| 0540 | HÖGR2 | 0541 | UTLSLAGG |
| 0542 | RÖKGASFL | 0543 | STOPTPROBLEM |
| 0544 | FILTERLARM | 0545 | GROVAVSKILJ |
| 0546 | HÖGR3 | 0547 | DIFFTRYCK |
| 0550 | SCIRKPUMP | 0551 | LCIRKPUMP |
| 0552 | BLOCKROST | 0553 | BLOCKINMAT |
| 0554 | BLOCKASK | 0555 | BLOCKFILT |

Bilagor

| | | | |
|------|---------------|---------------------|---------------|
| 0556 | BLOCKLJUDS | 0557 | CO-LARM |
| 0560 | BYPASSON | 0561 | ADAPTDAGS |
| 0562 | ADAPTFPOS | 0563 | LÅGEFFREGL |
| 0564 | OPERATION-OK | 0565 | ÖKAADPTF |
| 0566 | ÖKAADPTF | 0567 | EDGEAD1 |
| 0570 | EDGEAD2 | 0571 | ADNIVÁFL |
| 0572 | ADPULSFL | 0573 | MINTEMP |
| 0574 | ASK3ON | 0575 | PANNPÅSLAG |
| 0620 | PFLONMODE | 0621 | SINMAT-PÅ-AV |
| 0622 | STÄNGTSPJF | 0623 | ÖPPNATSPIF |
| 0624 | INMATBLOCK | 0625 | SINMAT2-PÅ-AV |
| 0626 | MATATRAVERSFL | 0630 | TNIVÅFLAGGA |
| 0631 | TPULSFLAGGA | 0640 | TRAVUTGFL |
| 0641 | TRAVFLUSEFL | 0642 | STOPGIVTRAVFL |
| 0643 | TRALLAUTGFL | 0644 | TRALLAMINFL |
| 0645 | STOPPTRALAFI | 0646 | TRAVMINFL |
| 0647 | TRALLAPLUSFL | 0650 | LARM-FLAGGA |
| 1000 | FLANK1 | 1001 | FLANK2 |
| 1002 | FLANK3 | 1003 | FLANK4 |
| 1004 | FLANK5 | 1005 | FLANK6 |
| 1006 | FLANK7 | 1007 | FLANK8 |
| 1010 | FLANK9 | 1011 | FLANK10 |
| 1012 | FLANK11 | 1013 | FLANK12 |
| 1014 | FLANK13 | 1015 | FLANK14 |
| 1016 | FLANK15 | 1017 | FLANK16 |
| 1020 | FLANK17 | 1021 | FLANK18 |
| 1022 | FLANK19 | 1023 | FLANK20 |
| 1024 | FLANK21 | 1025 | FLANK22 |
| 1026 | FLANK23 | 1027 | FLANK24 |
| 1030 | FLANK25 | 1031 | FLANK26 |
| 1032 | FLANK27 | 1033 | FLANK28 |
| 1034 | FLANK29 | 1035 | FLANK30 |
| 1036 | FLANK31 | 1037 | FLANK32 |
| 1040 | FLANK33 | 1041 | FLANK34 |
| 1042 | FLANK35 | 1043 | FLANK36 |
| 1044 | FLANK37 | 1045 | FLANK38 |
| 1046 | FLANK39 | 1047 | FLANK40 |
| 1050 | FLANK41 | 1051 | FLANK42 |
| 1052 | FLANK43 | 1053 | FLANK44 |
| 1054 | FLANK45 | 1055 | FLANK46 |
| 1056 | FLANK47 | 1057 | FLANK48 |
| 1060 | FLANK49 | 1061 | FLANK50 |
| 1062 | FLANK51 | 1063 | FLANK52 |
| 1064 | FLANK53 | 1065 | FLANK54 |
| 1066 | TRAVERSINIT | 1100 | KOORNÄST |
| 1101 | BRÄNGLÄST | 1102 | MATTERONOFF |
| 1103 | OMLÄSTONOFF | 1104 | TRAVERSLÄST |
| 1105 | TRAVUTGLÄGE | 1106 | TRAVBIT |
| 1107 | TRAVBIT2 | 1110 | TRAVBIT3 |
| 1111 | TRAVERSHECK | 1113 | TRAVBIT4 |
| 1114 | TRAVNLÄST | 1115 | TRAVERSREGLER |
| 1116 | BRASKOPA | 1117 | BIT |
| 1200 | PRINTLARM | 1201 | FLANK55 |
| 1202 | FLANK56 | 1203 | FLANK57 |
| 1204 | FLANK58 | 1205 | FLANK59 |
| 1206 | FLANK60 | 1207 | FLANK61 |
| 1210 | FLANK62 | 1211 | FLANK63 |
| 1212 | FLANK64 | 1213 | FLANK65 |
| 1214 | FLANK66 | 1215 | FLANK67 |
| 1216 | FLANK68 | 1217 | FLANK69 |
| 1220 | FLANK70 | 1221 | FLANK71 |
| 1222 | FLANK72 | 1223 | FLANK73 |
| 1224 | FLANK74 | 1225 | FLANK75 |
| 1226 | FLANK76 | 1227 | FLANK77 |
| 1230 | FLANK78 | 1231 | FLANK79 |
| 1232 | FLANK80 | 1233 | FLANK81 |
| 1234 | FLANK82 | 1235 | FLANK83 |
| 1236 | FLANK84 | 1237 | FLANK85 |
| 1240 | FLANK86 | | |
| 1241 | FLANK97 | | |
| 1657 | FAST-LOOP | | |
| 1660 | CMD0 | FAST LOOP EXECUTION | |
| 1661 | CMD1 | USER COMMAND 0 | |
| 1662 | CMD2 | USER COMMAND 1 | |
| 1663 | CMD3 | USER COMMAND 2 | |
| 1664 | CMD4 | USER COMMAND 3 | |
| | | USER COMMAND 4 | |

Bilagor

| | | | | |
|-------|---------------|-----------------------|-------|-------------|
| 1665 | CMD5 | USER COMMAND 5 | | |
| 1666 | EXEC-ERROR | PROGRAM EXECUTE ERROR | | |
| 1667 | TEXT-VDU | QUEUE ACTIVE | | |
| 1670 | TEXT-PRINTER | QUEUE ACTIVE | | |
| 1671 | LESS | < | | |
| 1672 | EQUAL | = | | |
| 1673 | GREATER | > | | |
| 1674 | BATTERY-LOW | | 1675 | FIRST-SCAN |
| 1676 | PULSE-10HZ | | 1677 | PULSE-1HZ |
| TI00 | TRAVERTI0 | | TI01 | TRAVERTI1 |
| TI02 | TRAVERTI2 | | TI03 | TRAVERTI3 |
| T00 | TRAVERT0 | | T01 | TRAVERT1 |
| T02 | TRAVERT2 | | T03 | TRAVERT3 |
| L000 | LOOP0 | | L001 | LOOP1 |
| L002 | LOOP2 | | L003 | LOOP3 |
| L004 | LOOP4 | | L005 | LOOP5 |
| L006 | LOOP6 | | L007 | LOOP7 |
| L008 | LOOP8 | | L009 | LOOP9 |
| L010 | LOOP10 | | L011 | LOOP11 |
| L012 | LOOP12 | | L013 | LOOP13 |
| L014 | LOOP14 | | L015 | LOOP15 |
| L016 | LOOP16 | | L017 | LOOP17 |
| L018 | LOOP18 | | L019 | LOOP19 |
| L020 | LOOP20 | | L021 | LOOP21 |
| L022 | LOOP22 | | L023 | LOOP23 |
| L024 | LOOP24 | | L025 | LOOP25 |
| L026 | LOOP26 | | L027 | LOOP27 |
| L028 | LOOP28 | | L029 | LOOP29 |
| L030 | LOOP30 | | L031 | LOOP31 |
| L032 | LOOP32 | | L033 | LOOP33 |
| L034 | LOOP34 | | L035 | LOOP35 |
| L036 | LOOP36 | | L037 | LOOP37 |
| L038 | LOOP38 | | L039 | LOOP39 |
| L040 | LOOP40 | | L041 | LOOP41 |
| L042 | LOOP42 | | L043 | LOOP43 |
| L044 | LOOP44 | | L045 | LOOP45 |
| L046 | LOOP46 | | L047 | LOOP47 |
| L048 | LOOP48 | | L049 | LOOP49 |
| L050 | LOOP50 | | L051 | LOOP51 |
| L052 | LOOP52 | | L053 | LOOP53 |
| L054 | LOOP54 | | L055 | LOOP55 |
| L056 | LOOP56 | | L057 | LOOP57 |
| L058 | LOOP58 | | L059 | LOOP59 |
| L060 | LOOP60 | | L061 | LOOP61 |
| L062 | LOOP62 | | L063 | LOOP63 |
| L064 | LOOP64 | | L065 | LOOP65 |
| L066 | LOOP66 | | L067 | LOOP67 |
| L068 | LOOP68 | | L069 | LOOP69 |
| L070 | LOOP70 | | L071 | LOOP71 |
| L072 | LOOP72 | | L073 | LOOP73 |
| L074 | LOOP74 | | L075 | LOOP75 |
| L076 | LOOP76 | | L077 | LOOP77 |
| L078 | LOOP78 | | L079 | LOOP79 |
| L080 | LOOP80 | | L081 | LOOP81 |
| L082 | LOOP82 | | L083 | LOOP83 |
| L084 | LOOP84 | | L085 | LOOP85 |
| L086 | LOOP86 | | L087 | LOOP87 |
| L088 | LOOP88 | | L089 | LOOP89 |
| L090 | LOOP90 | | L099 | LOOP99 |
| AI640 | ANALOG640 | | AI644 | ANALOG644 |
| AI650 | ANALOG650 | | AI654 | ANALOG654 |
| AI660 | ANALOG660 | | AI664 | ANALOG664 |
| AI670 | ANALOG670 | | AI700 | ANALOG700 |
| AI704 | ANALOG704 | | AI710 | ANALOG710 |
| AI714 | ANALOG714 | | AI720 | ANALOG720 |
| AI724 | ANALOG724 | | AO630 | ANAUT630 |
| R000 | RBÖRVÄRDE | | R001 | RHYSTERES |
| R002 | RÖVREBÖR | | R003 | RUNDERBÖR |
| R004 | RMINLÄGE | | R005 | RSPJÄLLÄGE |
| R006 | PBÖR | | R007 | PHYSTER |
| R008 | PÖVERBÖR | | R009 | FUNDERBÖR |
| R010 | FMINLÄGE | | R011 | O2-BÖR |
| R012 | O2-HYSTER | | R013 | O2-ÖVERBÖR |
| R014 | O2-UNDERBÖR | | R015 | SEK-MINLÄGE |
| R016 | SEK-SPJÄLLÄGE | | R017 | OFFTID |
| R018 | AINMAT-PA-T | | R019 | AINMAT-PA-P |

Bilagor

| | | | |
|------|---------------|------|---------------|
| R020 | AINMAT-PAUS-T | R021 | AINMAT-PAUS-P |
| R022 | INMATCOUNT | R023 | BRANSLEAGX |
| R024 | AFILTERTID | R025 | ONTID |
| R026 | PLJUDS-PA-T | R027 | PLJUDS-PA-P |
| R028 | PLJUDS-PAUS-T | R029 | PLJUDS-PAUS-P |
| R030 | ASKTID | R031 | ASKPER1 |
| R032 | ASKPER2 | R033 | ASKPER3 |
| R034 | ASKMÄNGD | R035 | SEKFLPÅTID |
| R036 | PRIFLÅKTPÅT | R037 | DELTAPOFF |
| R038 | DELTAPOFF | R039 | RINMATTID |
| R040 | ROSTERSTYRT | R041 | ROSTERTID |
| R042 | ONTEMP | R043 | OFFTEMP |
| R044 | INMATSTID | R045 | RÖKFLSTARTT |
| R046 | SEKFLSTARTT | R047 | CO-GRÄNS |
| R048 | FILTR-PÅ-T | R049 | FILTERREN |
| R050 | FILTR-PAUS-T | R051 | FILTERPAUSP |
| R052 | LJUDSOTTID | R053 | PRIFLSTARTT |
| R065 | FRAMTEMP | R066 | RETURTEMP |
| R067 | UGNSTEMP | R068 | O2-HALT |
| R069 | CO-HALT | R070 | DELTA-P |
| R071 | ELDUTRYCK | R072 | ROKASTEMP |
| R073 | ECOGASTEMP | R074 | PSPJÄLLÅGE |
| R075 | DELTBÖP | R076 | DELTBST |
| R077 | INMATPOFFT | R078 | INMATAOFFT |
| R079 | PFFLOFFTEMP | R080 | PRIUNDRY1 |
| R081 | PRIUNDRY2 | R082 | PRIUNDRY3 |
| R083 | TERTUNDRY3 | R084 | TERTOÖPPNA |
| R085 | SEKUNDRY3 | R086 | INMATCOMAX1 |
| R087 | INMATO2MIN1 | R088 | INMATO2MIN |
| R089 | INMATSOFFTID | R090 | INMATPÅTID |
| R091 | PFLONTEMP | R092 | SINMAT-PAUS-T |
| R093 | SINMAT-PÅ-P | R094 | SINMAT-PÅ-T |
| R095 | SINMAT2-PAS-P | R096 | SINMAT2-PAS-T |
| R097 | SINMAT2-PÅ-P | R098 | SINMAT2-PÅ-T |
| R099 | CO-BÖR | R100 | TERTHYST |
| R101 | TMINLÅGE | R102 | TPAUSPER |
| R103 | TPAUSTID | R104 | TPULSPERIOD |
| R105 | TPULSTID | R106 | TSPJÄLLÅGE |
| R107 | TUNDRBÖR | R108 | TÖVREBÖR |
| R109 | DELTABÖR | R110 | DELTAHYST |
| R111 | ÖVERPOFFTE | R112 | SINMAT-PAUS-P |
| R113 | EFFEKT | R114 | ENERGIPER |
| R115 | ENERGIPERT | R116 | ANTALEPULS |
| R117 | ANTALFPULS | R118 | FLÖDEPER |
| R119 | FLÖDEPERT | R120 | FLÖDE |
| R121 | BRAGGTID | R122 | TRAVERSMATT |
| R123 | SENASTLARM | R124 | TRAVCHECKT |
| R125 | TRAVLÅGE | R129 | PANNTEMP |
| R130 | K1 | R131 | K2 |
| R132 | K3 | R133 | K4 |
| R134 | K5 | R135 | K6 |
| R136 | K7 | R137 | K8 |
| R138 | K9 | R139 | K10 |
| R140 | K11 | R141 | K12 |
| R142 | K1 | R143 | K14 |
| R144 | K15 | R145 | K16 |
| R146 | K17 | R147 | K18 |
| R148 | K19 | R149 | K20 |
| R150 | K21 | R151 | K22 |
| R152 | K23 | R153 | K24 |
| R154 | K25 | R155 | K26 |
| R156 | K27 | R157 | K28 |
| R158 | K29 | R159 | K30 |
| R160 | K31 | R161 | K32 |
| R162 | K33 | R163 | K34 |
| R164 | K35 | R165 | K36 |
| R166 | K37 | R167 | K38 |
| R168 | K39 | R169 | K40 |
| R170 | K41 | R171 | K42 |
| R172 | K43 | R173 | K44 |
| R174 | K45 | R175 | K46 |
| R176 | K47 | R177 | K48 |
| R178 | K49 | R179 | K50 |
| R180 | K51 | R181 | K52 |
| R182 | K53 | R183 | K54 |
| R184 | K55 | R185 | K56 |

Bilagor

| | | | |
|------|-------------|------|---------------|
| R186 | K57 | R187 | K58 |
| R188 | K59 | R189 | K60 |
| R190 | KSLUT | R200 | BLANDNING1 |
| R201 | BLANDNING2 | R202 | BLANDNING3 |
| R203 | BLANDNING4 | R204 | BLANDNING5 |
| R205 | BLANDNING6 | R206 | BLANDNING7 |
| R207 | BLANDNING8 | R208 | BLANDNING9 |
| R209 | BLANDNING10 | R210 | TRAVPER1 |
| R211 | TRAVPER2 | R212 | TRAVPER3 |
| R220 | TRAVPER4 | R221 | ÖKAMATTERTOFF |
| R222 | TRAVERSMATP | R225 | RÖKGASTEMPMIN |
| R226 | FRAMTEMPMIN | R227 | CO-MAX |
| R240 | PPULSPER | R241 | PPAUSPER |
| R242 | PPULSTID | R243 | PPAUSTID |
| R244 | RPULSTID | R245 | RPAUSTID |
| R246 | RPULSPER | R247 | RPAUSPER |
| R248 | SPAUSTID | R249 | SPULSTID |
| R250 | SPULSPER | R251 | SPAUSPER |
| R252 | TOTPSPJÄLL | R253 | ANTMÄTPSPJ |
| R254 | ANTMÄTTSPJ | R255 | TOTTSPJÄLL |
| R256 | ANTMÄTD-P | R257 | TOTD-P |
| R258 | MPSPJÄLLÄGE | R259 | MTSPJÄLLÄGE |
| R260 | MD-P-LÄGE | R270 | ADAPTFAKT |
| R271 | ADAPTPER | R272 | ADAPTTID |
| R273 | MAXADAPT | R274 | RÖKGASKONST |
| R275 | RÖKGASM | R276 | RÖKGASTBÖR |
| R277 | VATTENADMAX | R278 | VATTENADMIN |
| R279 | VATTENFAKT | R280 | VATTENFEL |
| R281 | ADHYSTER | R282 | ADPAUSPER |
| R283 | ADPAUST | R284 | ADPULSPER |
| R285 | ADPULST | R286 | LÄGEFFEKT |
| R287 | COCHECKT | R300 | TRAVERSLÄGE |
| R301 | TRALLALÄGE | R302 | NEWX |
| R303 | NEWY | R304 | FLYTTRIKT |

※

Bilagor

9.3 OPCOMM

Huvudprogram: OPCOMM

```
program PC_OPERAT_RSKOMMUNIKATION;

{$V-}
{$F+}
{$N+}
{$A-}
{$M 50000,0,655360}

uses Syngate, Lay_Pas, selarm, comlipr, angekoor, angematf, askrostb,
    filterlj, flaktar, initpcst, omlast, offon, primspj, proginst, reglerbr,
    rokgassp, sekundsp, setraver, tertsp, travinst, sepanna, inmatnin ;

{ ----- Include Files ----- }

{$I C:\FCSTYR\PASCAL\OPCOMM.COD}

{$I C:\FCSTYR\PASCAL\OPCOMM.RSC}

var bakgrundsbild__r_panna:integer;

procedure init_layout_prog_vars;

begin
    menu_flag := 2;
    Menu_Structure := @Menu_Struct_Table;
    data_menu_titles := 4;
    max_menu_width := 34;
    max_menu_height := 11;
end;

begin
    initcomli;
    open_process;
    init_layout_prog_vars;
    initialize_screen (0);
    setup_screen_variables;
    set_desktop ($AA,$55,0);
    fill_desktop (0);
    setup_menu (menu_bar);

    TRAVERSOP_INITIERING(bakgrundsbild__r_panna);

    while (avsluta_flagga <> 2730) do begin
        { håll på }
        get_menu_input (1, MENYVAL);
        { VÄLJ MENY }
        if (MENYVAL.object_type = 2) AND (MENYVAL.object_data1 = $0101) then begin

            overvakpanna (bakgrundsbild__r_panna);

        end
        else if (MENYVAL.object_type = 2) AND (MENYVAL.object_data1 = $0102) then begin

            overvaktravers(bakgrundsbild__r_panna);

        end
        else if (MENYVAL.object_type = 2) AND (MENYVAL.object_data1 = $0103) then begin

            overvaklarm(bakgrundsbild__R_panna);

        end
        else if (MENYVAL.object_type = 2) AND (MENYVAL.object_data1 = $0201) then begin

            ANGE_INMATNINGSKOORDINATER(bakgrundsbild__r_panna);

        end
        else if (MENYVAL.object_type = 2) AND (MENYVAL.object_data1 = $0202) then begin
```

Bilagor

```

                                ange_ny_inmatningsf_ljd;

end
else if (MENYVAL.object_type = 2) AND (MENYVAL.object_data1 = $0203) then begin

                                reglerbr_nsle;

end
else if (MENYVAL.object_type = 2) AND (MENYVAL.object_data1 = $0204) then begin

                                Omlastn_;

end
else if (MENYVAL.object_type = 2) AND (MENYVAL.object_data1 = $0205) then begin

                                TraversInst_llning;

end
else if (MENYVAL.object_type = 2) AND (MENYVAL.object_data1 = $0302) then begin

                                Prim_rspj_ll;

end
else if (MENYVAL.object_type = 2) AND (MENYVAL.object_data1 = $0206) then begin

                                ProgramInst_llning;

end
else if (MENYVAL.object_type = 2) AND (MENYVAL.object_data1 = $0301) then begin

                                Inmatningen;

end
else if (MENYVAL.object_type = 2) AND (MENYVAL.object_data1 = $0303) then begin

                                Sekund_rspj_ll;

end
else if (MENYVAL.object_type = 2) AND (MENYVAL.object_data1 = $0304) then begin

                                tertspj_ll;

end
else if (MENYVAL.object_type = 2) AND (MENYVAL.object_data1 = $0305) then begin

                                R_kgasspj_ll;

end
else if (MENYVAL.object_type = 2) AND (MENYVAL.object_data1 = $0306) then begin

                                Fl_ktar;

end
else if (MENYVAL.object_type = 2) AND (MENYVAL.object_data1 = $0307) then begin

                                FILTERLJUDSOTNING;

end
else if (MENYVAL.object_type = 2) AND (MENYVAL.object_data1 = $0308) then begin

                                Askrosterbragg;

end
else if (MENYVAL.object_type = 2) AND (MENYVAL.object_data1 = $0309) then begin

                                ONOFF;

end;
end;
reset_program;
close_process;
end.
```

Kommentarer:

De markerade områdena utgör de ändringar som man måste göra av den ursprungligt gegerade källkoden. På samma sätt måste man göra ändringar i varje nyskriven tpu som man vill införa.

✱

Bilagor

9.4 COMLIPR

UNIT COMLIPR;

INTERFACE

USES Crt;

Procedure GetAnaloga(var COhalt, Deltap, Ecotemp, Elduttryck,
Framtemp, O2halt, Returtemp, Rokgastemp,
Ugnstemp, Lageprimar, Lagesekund, Lagetert,
Effekt, Flode, Panntemp: INTEGER);

Procedure GetInmatning(var Ainmat_Pa_P, Ainmat_Paus_P,
Deltabop, Deltast, Deltahyst,
Inmatpofft, Inmataofft, InmatCmax1,
InmatO2min1, InmatO2min, Inmatttid,
Inmatspatid: INTEGER);

Procedure SetInmatning(Ainmat_Pa_P, Ainmat_Paus_P,
Deltabop, Deltast, Deltahyst,
Inmatpofft, Inmataofft, InmatCmax1,
InmatO2min1, InmatO2min, Inmatttid,
Inmatspatid: INTEGER);

Procedure GetPriluftspjall(var Pribor, Pausperiod, Pulsperiod, Minlage, Undrebor,
ovrebor, Hysteres, Priundtry1, Priundtry2, Priflaktstartt
: integer);

Procedure SetPriluftspjall(Pribor, Pausperiod, Pulsperiod, Minlage, Undrebor,
ovrebor, Hysteres, Priundtry1, Priundtry2, Priflaktstartt
: integer);

Procedure GetSekluftspjall(var O2Bor, Pausperiod, Pulsperiod, Minlage, Undrebor,
ovrebor, Hysteres, Sekundtrst, Sekfaktstartt: integer);

Procedure SetSekluftspjall(O2Bor, Pausperiod, Pulsperiod, Minlage, Undrebor,
ovrebor, Hysteres, Sekundtrst, Sekfaktstartt: integer);

Procedure GetTerluftspjall(var CObor, Pulsperiod, Pausperiod, Minlage, Undrebor,
ovrebor, Hysteres, Tertundtrst, TertO2oppna: integer);

Procedure SetTerluftspjall(CObor, Pulsperiod, Pausperiod, Minlage, Undrebor,
ovrebor, Hysteres, Tertundtrst, TertO2oppna: integer);

Procedure GetRokgasspjall (var Eldubor, Pausperiod, Pulsperiod, Underbor,
overbor, Hysteres, Rokflstart: integer);

Procedure SetRokgasspjall (Eldubor, Pausperiod, Pulsperiod, Minlage, Underbor,
overbor, Hysteres, Rokflstart: integer);

Procedure GetFlaktar (var Sekflpatid, Prifaktpatid, Pfllofftemp: integer);

Procedure SetFlaktar (Sekflpatid, Prifaktpatid, Pfllofftemp: integer);

Procedure GetRening (var Filtertid, CO_grans, Filterenp, Filterpausp,
Ljudsottid, pljudsottid, Pljuds_paus_p: integer);

Procedure SetRening (var Filtertid, CO_grans, Filterenp, Filterpausp,
Ljudsottid, pljudsottid, Pljuds_paus_p: integer);

Procedure GetAskRostBrag (var Askper1, Askper2, Askper3, Askmamgd,
Rosterstyrt, Bransleagx: integer);

Procedure SetAskRosTBrag (Askper1, Askper2, Askper3, Askmamgd,
Rosterstyrt, Bransleagx: integer);

Procedure GetOnOff (var OnTemp, Offtemp: integer);

Procedure SetOnOff (OnTemp, Offtemp: integer);

Procedure GetLarm (var ump, bb, us, loehar, hthar, unt, um, lpt, hpt, lpc, uks
, uoe, lr1, hr1, llk, uvi, ums, hr2, ti, uss, k, ur, fav, mr, topp

Bilagor

```

,sp,ff,gr,hr3,dtv,mt:Integer);

Procedure GetPannfunktioner(VAR inmatningsflagga,larmflagga,onflagga,rosterflagga,
    primarflaktflagga,larm,prlarm:integer);

PROCEDURE Send_Coord_Status(VAR KS1,KS2,KS3,KS4,KS5,KS6,KS7,KS8,KS9,KS10,
    KS11,KS12,KS13,KS14,KS15,KS16,KS17,KS18,KS19,KS20,
    KS21,KS22,KS23,KS24,KS25,KS26,KS27,KS28,KS29,KS30,
    KS31,KS32,KS33,KS34,KS38,KS39,KS40,KS41,KS42,KS46,
    KS47,KS48,KS49,KS50,KS51,KS52,KS53,KS54,KS55,KS56,
    KS57,KS58,KS59,KS60 : INTEGER);

PROCEDURE Send_Feeding_Stores(feedlist1,feedlist2,feedlist3,feedlist4,
    feedlist5,feedlist6,feedlist7,feedlist8,feedlist9,
    feedlist10,feedlist11,feedlist12,matterhorn :INTEGER);

PROCEDURE Send_Control_Status(rokgastempmin,framledningstempmin,
    C0max,reglerlevel : INTEGER);

PROCEDURE Set_Reeload_Store(reloadstore,omlastning :INTEGER);

PROCEDURE Set_Traverse_Status(TraversFixPosition,TillUtgningslge :INTEGER);

PROCEDURE Set_Program_data(matterhorn_kning,ultraljudstid :INTEGER);

PROCEDURE Get_Coord_Status(VAR KS1,KS2,KS3,KS4,KS5,KS6,KS7,KS8,KS9,KS10,
    KS11,KS12,KS13,KS14,KS15,KS16,KS17,KS18,KS19,KS20,
    KS21,KS22,KS23,KS24,KS25,KS26,KS27,KS28,KS29,KS30,
    KS31,KS32,KS33,KS34,KS38,KS39,KS40,KS41,KS42,KS46,
    KS47,KS48,KS49,KS50,KS51,KS52,KS53,KS54,KS55,KS56,
    KS57,KS58,KS59,KS60 : INTEGER);

PROCEDURE Get_Feeding_Stores(VAR feedlist1,feedlist2,feedlist3,feedlist4,
    feedlist5,feedlist6,feedlist7,feedlist8,feedlist9,
    feedlist10,feedlist11,feedlist12 :INTEGER);

PROCEDURE Get_Reeload_Store(VAR omlastning : INTEGER);

PROCEDURE InitTravers;

PROCEDURE Get_Matterhorn_Status(Var Matterhorn : INTEGER);

Procedure InitComli;

IMPLEMENTATION

{$I comli.sys}
{$I Setbit.inc}
{$I Getbit.inc}
{$I GetReg.inc}
{$I SetReg.inc}

Procedure InitComli;
begin
    InitProgvar;
    Init1(12,5,1000,BINcomm,FastPC);
end;

Procedure GetAnaloga(var COhalt,Deltap,Ecotemp,Elduttryck,
    Framtemp,O2halt,Returtemp,Rokgastemp,
    Ugnstemp,Lageprimar,Lagesekund,Lagetert,
    Effekt,Flode,panntemp:INTEGER);

BEGIN
    GetReg (2,1,69);
    COhalt:=TRUNC(Realvalue[1]);
    Getreg (2,1,70);
    Deltap:=TRUNC(Realvalue[1]);
```

Bilagor

```
GetReg (2,1,73);
ecotemp:=TRUNC(Realvalue[1]);
Getreg (2,1,71);
elduttryck:=TRUNC(Realvalue[1]);
GetReg (2,1,65);
framtemp:=TRUNC(Realvalue[1]);
Getreg (2,1,68);
o2halt:=TRUNC(Realvalue[1]);
GetReg (2,1,66);
returtemp:=TRUNC(Realvalue[1]);
Getreg (2,1,72);
rokgastemp:=TRUNC(Realvalue[1]);
GetReg (2,1,67);
ugnstemp:=TRUNC(Realvalue[1]);
Getreg (2,1,74);
lageprimar:=TRUNC(Realvalue[1]);
GetReg (2,1,16);
lagesekund:=TRUNC(Realvalue[1]);
Getreg (2,1,106);
lagetert:=TRUNC(Realvalue[1]);
GetReg (2,1,113);
effekt:=TRUNC(Realvalue[1]);
Getreg (2,1,120);
flode:=TRUNC(Realvalue[1]);
getreg (2,1,129);
panntemp:=trunc(realvalue[1]);
END;
Procedure GetInmatning(var Ainmat_Pa_P,Ainmat_Paus_P,
Deltabop,Deltast,Deltahyst,
Inmatpofft,Inmataofft,InmatCmax1,
InmatO2min1,InmatO2min,Inmatttid,
Inmatspatid:INTEGER);
```

BEGIN

```
GetReg (2,1,19);
ainmat_pa_p:=TRUNC(Realvalue[1]);
Getreg (2,1,21);
ainmat_paus_p:=TRUNC(Realvalue[1]);
GetReg (2,1,75);
deltabop :=TRUNC(Realvalue[1]);
Getreg (2,1,76);
deltast :=TRUNC(Realvalue[1]);
GetReg (2,1,110);
deltahyst:=TRUNC(Realvalue[1]);
Getreg (2,1,93);
inmatpofft:=TRUNC(Realvalue[1]);
GetReg (2,1,112);
inmataofft:=TRUNC(Realvalue[1]);
Getreg (2,1,86);
inmatcomax1:=TRUNC(Realvalue[1]);
GetReg (2,1,87);
inmato2min1:=TRUNC(Realvalue[1]);
Getreg (2,1,88);
inmato2min:=TRUNC(Realvalue[1]);
GetReg (2,1,95);
inmatttid :=TRUNC(Realvalue[1]);
GetReg (2,1,97);
inmatspatid :=TRUNC(Realvalue[1]);
```

END;

```
Procedure SetInmatning( Ainmat_Pa_P,Ainmat_Paus_P,
Deltabop,Deltast,Deltahyst,
Inmatpofft,Inmataofft,InmatCmax1,
InmatO2min1,InmatO2min,Inmatttid,
Inmatspatid:INTEGER);
```

BEGIN

```
Realvalue[1]:=ainmat_pa_p;
setreg(2,1,19);
Realvalue[1]:=Ainmat_paus_p;
setreg(2,1,21);
Realvalue[1]:=deltabop;
setreg(2,1,75);
Realvalue[1]:=deltast;
setreg(2,1,76);
```

Bilagor

```
Realvalue[1]:=deltahyst;
setreg(2,1,110);
Realvalue[1]:=inmatpofft;
setreg(2,1,93);
Realvalue[1]:=inmataofft;
setreg(2,1,112);
Realvalue[1]:=inmatcomax1;
setreg(2,1,86);
Realvalue[1]:=inmato2min1;
setreg(2,1,87);
Realvalue[1]:=inmato2min;
setreg(2,1,88);
Realvalue[1]:=inmatttid;
setreg(2,1,95);
Realvalue[1]:=inmatspatid;
setreg(2,1,97);
END;
```

```
Procedure GetPriluftspjall(var Pribor,Pausperiod,Pulsperiod,Minlage,Undrebor,
ovrebor,Hysteres,Priundtry1,Priundtry2,Priflaktstartt
:integer);
```

```
BEGIN
  GetReg (2,1,6);
  pribor:=TRUNC(Realvalue[1]);
  Getreg (2,1,241);
  pausperiod:=TRUNC(Realvalue[1]);
  GetReg (2,1,240);
  pulsperiod :=TRUNC(Realvalue[1]);
  Getreg (2,1,10);
  minlage:=TRUNC(Realvalue[1]);
  GetReg (2,1,9);
  undrebor :=TRUNC(Realvalue[1]);
  Getreg (2,1,8);
  ovrebor :=TRUNC(Realvalue[1]);
  GetReg (2,1,7);
  Hysteres :=TRUNC(Realvalue[1]);
  Getreg (2,1,80);
  priundtry1 :=TRUNC(Realvalue[1]);
  GetReg (2,1,81);
  priundtry2 :=TRUNC(Realvalue[1]);
  Getreg (2,1,53);
  priflaktstartt:=TRUNC(Realvalue[1]);
END;
```

```
Procedure SetPriluftspjall( Pribor,Pausperiod,Pulsperiod,Minlage,Undrebor,
ovrebor,Hysteres,Priundtry1,Priundtry2,Priflaktstartt
:integer);
```

```
BEGIN
  Realvalue[1]:=Pribor;
  SetReg(2,1,6);
  Realvalue[1]:=pausperiod;
  SetReg(2,1,241);
  Realvalue[1]:=pulsperiod;
  SetReg(2,1,240);
  Realvalue[1]:=minlage;
  SetReg(2,1,10);
  Realvalue[1]:=undrebor;
  SetReg(2,1,9);
  Realvalue[1]:=ovrebor;
  SetReg(2,1,8);
  Realvalue[1]:=hysteres;
  SetReg(2,1,7);
  Realvalue[1]:=priundtry1;
  SetReg(2,1,80);
  Realvalue[1]:=priundtry2;
  SetReg(2,1,81);
  Realvalue[1]:=priflaktstartt;
  SetReg(2,1,53);
END;
```

```
Procedure GetSekluftspjall(var O2Bor,Pausperiod,Pulsperiod,Minlage,Undrebor,
```


Bilagor

ovrebor, Hysteres, Sekundtrst, Sekfaktstartt: integer);

BEGIN

```
GetReg (2,1,11);
o2bor :=TRUNC(Realvalue[1]);
Getreg (2,1,251);
pausperiod:=TRUNC(Realvalue[1]);
GetReg (2,1,250);
pulsperiod :=TRUNC(Realvalue[1]);
Getreg (2,1,15);
minlage:=TRUNC(Realvalue[1]);
GetReg (2,1,14);
undrebor:=TRUNC(Realvalue[1]);
Getreg (2,1,13);
ovrebor:=TRUNC(Realvalue[1]);
GetReg (2,1,12);
Hysteres :=TRUNC(Realvalue[1]);
Getreg (2,1,85);
sekundtrst :=TRUNC(Realvalue[1]);
GetReg (2,1,46);
sekfaktstartt :=TRUNC(Realvalue[1]);
```

END;

Procedure SetSekluftspjall(O2Bor, Pausperiod, Pulsperiod, Minlage, Undrebor,
ovrebor, Hysteres, Sekundtrst, Sekfaktstartt: integer);

BEGIN

```
RealValue[1]:=o2bor;
SetReg(2,1,11);
RealValue[1]:=pausperiod;
SetReg(2,1,251);
RealValue[1]:=pulsperiod;
SetReg(2,1,250);
RealValue[1]:=minlage;
SetReg(2,1,15);
RealValue[1]:=undrebor;
SetReg(2,1,14);
RealValue[1]:=ovrebor;
SetReg(2,1,13);
RealValue[1]:=hysteres;
SetReg(2,1,12);
RealValue[1]:=sekundtrst;
SetReg(2,1,85);
RealValue[1]:=sekfaktstartt;
setreg(2,1,46);
```

END;

Procedure GetTerluftspjall(var CObor, Pulsperiod, Pausperiod, Minlage, Undrebor,
ovrebor, Hysteres, Tertundtrst, TertO2oppna: integer);

BEGIN

```
GetReg (2,1,99);
cobor :=TRUNC(Realvalue[1]);
Getreg (2,1,104);
pulsperiod:=TRUNC(Realvalue[1]);
GetReg (2,1,102);
pausperiod:=TRUNC(Realvalue[1]);
Getreg (2,1,101);
minlage:=TRUNC(Realvalue[1]);
GetReg (2,1,107);
undrebor:=TRUNC(Realvalue[1]);
Getreg (2,1,108);
ovrebor:=TRUNC(Realvalue[1]);
GetReg (2,1,100);
hysteres:=TRUNC(Realvalue[1]);
Getreg (2,1,83 );
tertundtrst:=TRUNC(Realvalue[1]);
GetReg (2,1,84);
terto2oppna :=TRUNC(Realvalue[1]);
```

END;

Procedure SetTerluftspjall(CObor, Pulsperiod, Pausperiod, Minlage, Undrebor,
ovrebor, Hysteres, Tertundtrst, TertO2oppna: integer);

Bilagor

BEGIN

```
Realvalue[1]:=cobor;  
SetReg(2,1,99);  
Realvalue[1]:=pulsperiod;  
SetReg(2,1,104);  
Realvalue[1]:=pausperiod;  
SetReg(2,1,102);  
Realvalue[1]:=minlage;  
SetReg(2,1,101);  
Realvalue[1]:=undrebtor;  
SetReg(2,1,107);  
Realvalue[1]:=ovrebtor;  
SetReg(2,1,108);  
Realvalue[1]:=hysteres;  
SetReg(2,1,100);  
Realvalue[1]:=tertundtrst;  
SetReg(2,1,83);  
Realvalue[1]:=terto2oppna;  
SetReg(2,1,84);
```

end;

Procedure GetRokgasspjall (var Eldubor,Pausperiod,Pulsperiod,Underbor,
overbor,Hysteres,Rokflstart:integer);

BEGIN

```
GetReg (2,1,0);  
eldubor:=TRUNC(Realvalue[1]);  
Getreg (2,1,247);  
pausperiod:=TRUNC(Realvalue[1]);  
GetReg (2,1,246);  
pulsperiod:=TRUNC(Realvalue[1]);  
GetReg (2,1,3);  
underbor:=TRUNC(Realvalue[1]);  
Getreg (2,1,2);  
overbor :=TRUNC(Realvalue[1]);  
GetReg (2,1,1);  
hysteres :=TRUNC(Realvalue[1]);  
Getreg (2,1,45);  
rokflstart:=TRUNC(Realvalue[1]);
```

END;

Procedure SetRokgasspjall (Eldubor,Pausperiod,Pulsperiod,Minlage,Underbor,
overbor,Hysteres,Rokflstart:integer);

BEGIN

```
RealValue[1]:=eldubor;  
SetReg(2,1,0);  
RealValue[1]:=pausperiod;  
SetReg(2,1,247);  
RealValue[1]:=pulsperiod;  
SetReg(2,1,246);  
RealValue[1]:=minlage;  
SetReg(2,1,4);  
RealValue[1]:=underbor;  
setreg(2,1,3);  
RealValue[1]:=overbor;  
SetReg(2,1,2);  
RealValue[1]:=hysteres;  
SetReg(2,1,1);  
RealValue[1]:=rokflstart;  
SetReg(2,1,45);
```

END;

Procedure GetFlaktar (var Sekflpatid,Prifaktpatid,Pflofftemp:integer);

BEGIN

```
GetReg (2,1,35);  
sekflpatid :=TRUNC(Realvalue[1]);  
Getreg (2,1,36);
```

Bilagor

```
prifaktpatid:=TRUNC(Realvalue[1]);
GetReg (2,1,79);
pflofftemp :=TRUNC(Realvalue[1]);
```

END;

```
Procedure SetFlaktar ( Sekflpatid,Prifaktpatid,Pfloffttemp:integer);
```

BEGIN

```
Realvalue[1]:=sekflpatid;
Setreg(2,1,35);
Realvalue[1]:=prifaktpatid;
setreg(2,1,36);
Realvalue[1]:=pflofftemp;
setreg(2,1,79);
```

END;

```
Procedure GetRening (var Filtertid,CO_grans,Filterenp,Filterpausp,
Ljudsottid,pljudsottid,Pljuds_paus_p:integer);
```

BEGIN

```
GetReg (2,1,24);
filtertid:=TRUNC(Realvalue[1]);
Getreg (2,1,47);
co_grans:=TRUNC(Realvalue[1]);
GetReg (2,1,49);
filterenp:=TRUNC(Realvalue[1]);
Getreg (2,1,51);
filterpausp:=TRUNC(Realvalue[1]);
GetReg (2,1,52);
Ljudsottid:=TRUNC(Realvalue[1]);
Getreg (2,1,27);
pljudsottid:=TRUNC(Realvalue[1]);
getreg(2,1,29);
pljuds_paus_p:=TRUNC(Realvalue[1]);
```

END;

```
Procedure SetRening (var Filtertid,CO_grans,Filterenp,Filterpausp,
Ljudsottid,pljudsottid,pljuds_paus_p:integer);
```

BEGIN

```
Realvalue[1]:=filtertid;
SetReg(2,1,24);
Realvalue[1]:=co_grans;
SetReg(2,1,47);
Realvalue[1]:=filterenp;
SetReg(2,1,49);
Realvalue[1]:=filterpausp;
SetReg(2,1,51);
Realvalue[1]:=Ljudsottid;
SetReg(2,1,52);
Realvalue[1]:=pljuds_paus_p;
setreg(2,1,29);
realvalue[1]:=pljudsottid;
setreg(2,1,27);
end;
```

```
Procedure GetAskRostBrag (var Askper1,Askper2,Askper3,Askmamgd,
Rosterstyrt,Bransleagx:integer);
```

BEGIN

```
GetReg (2,1,31);
askper1:=TRUNC(Realvalue[1]);
Getreg (2,1,32);
askper2:=TRUNC(Realvalue[1]);
GetReg (2,1,33);
askper3:=TRUNC(Realvalue[1]);
Getreg (2,1,34);
askmamgd:=TRUNC(Realvalue[1]);
GetReg (2,1,40);
rosterstyrt:=TRUNC(Realvalue[1]);
Getreg (2,1,23);
```

Bilagor

```
bransleagx:=TRUNC(Realvalue[1]);  
END;
```

```
Procedure SetAskRostBrag ( Askper1,Askper2,Askper3,Askmamgd,  
Rosterstyrt,Bransleagx:integer);
```

```
BEGIN
```

```
Realvalue[1]:=askper1;  
SetReg(2,1,31);  
Realvalue[1]:=askper2;  
SetReg(2,1,32);  
Realvalue[1]:=askper3;  
SetReg(2,1,33);  
Realvalue[1]:=askmamgd;  
SetReg(2,1,34);  
Realvalue[1]:=rosterstyrt;  
SetReg(2,1,40);  
Realvalue[1]:=bransleagx;  
SetReg(2,1,23);
```

```
END;
```

```
Procedure GetOnOff (var OnTemp,Offtemp:integer);
```

```
BEGIN
```

```
GetReg(2,1,42);  
ontemp:=TRUNC(Realvalue[1]);  
getreg(2,1,43);  
offtemp:=TRUNC(realvalue[1]);  
end;
```

```
Procedure SetOnOff ( OnTemp,Offtemp:integer);
```

```
BEGIN
```

```
Realvalue[1]:=ontemp;  
setreg(2,1,42);  
Realvalue[1]:=offtemp;  
setreg(2,1,43);
```

```
END;
```

```
Procedure GetLarm (var ump,bb,us,loehar,hthar,unt,um,lpt,hpt,lpc,uks  
,uoe,lr1,hr1,llk,uvi,ums,hr2,ti,uss,k,ur,fav,mr,topp  
,sp,ff,gr,hr3,dtv,mt:integer);
```

```
Begin
```

```
GetBit(2,440,ump);  
GetBit(2,441,bb);  
GetBit(2,442,us);  
GetBit(2,443,loehar);  
GetBit(2,444,hthar);  
GetBit(2,446,unt);  
GetBit(2,447,um);  
GetBit(2,450,lpt);  
GetBit(2,451,hpt);  
GetBit(2,452,lpc);  
GetBit(2,453,uks);  
GetBit(2,454,uoe);  
GetBit(2,455,lr1);  
GetBit(2,456,hr1);  
GetBit(2,142,llk);  
GetBit(2,143,uvi);  
GetBit(2,457,ums);  
GetBit(2,540,hr2);  
GetBit(2,150,ti);  
GetBit(2,541,uss);  
GetBit(2,145,k);  
GetBit(2,146,ur);  
GetBit(2,147,fav);  
GetBit(2,542,mr);  
GetBit(2,144,topp);  
GetBit(2,543,sp);  
GetBit(2,544,ff);  
GetBit(2,545,gr);  
GetBit(2,546,hr3);  
GetBit(2,547,dtv);  
GetBit(2,516,mt)
```

Bilagor

end;

```
Procedure GetPannfunktioner(var inmatningsflagga,larmflagga,onflagga,rosterflagga,  
    primarflaktflagga,larm,prlarm:integer);
```

```
Begin
```

```
    GetBit(2,467,inmatningsflagga);  
    GetBit(2,650,larmflagga);  
    GetBit(2,526,onflagga);  
    GetBit(2,523,rosterflagga);  
    GetBit(2,465,primarflaktflagga);  
    getbit(2,1200,prlarm);  
    getreg(2,1,123);  
    larm:=trunc(realvalue[1]);  
    SetBit(2,1200,0);
```

```
END;
```

```
{-----Send Coord Status-----}
```

```
PROCEDURE Send_Coord_Status(VAR KS1,KS2,KS3,KS4,KS5,KS6,KS7,KS8,KS9,KS10,  
    KS11,KS12,KS13,KS14,KS15,KS16,KS17,KS18,KS19,KS20,  
    KS21,KS22,KS23,KS24,KS25,KS26,KS27,KS28,KS29,KS30,  
    KS31,KS32,KS33,KS34,KS38,KS39,KS40,KS41,KS42,KS46,  
    KS47,KS48,KS49,KS50,KS51,KS52,KS53,KS54,KS55,KS56,  
    KS57,KS58,KS59,KS60 : INTEGER);
```

```
VAR Sattaddr,i : INTEGER;  
    coordinatstatus : ARRAY[1..60] OF INTEGER;
```

```
BEGIN
```

```
    coordinatstatus[1]:=KS1; coordinatstatus[2]:=KS2;  
    coordinatstatus[3]:=KS3; coordinatstatus[4]:=KS4;  
    coordinatstatus[5]:=KS5; coordinatstatus[6]:=KS6;  
    coordinatstatus[7]:=KS7; coordinatstatus[8]:=KS8;  
    coordinatstatus[9]:=KS9; coordinatstatus[10]:=KS10;  
    coordinatstatus[11]:=KS11; coordinatstatus[12]:=KS12;  
    coordinatstatus[13]:=KS13; coordinatstatus[14]:=KS14;  
    coordinatstatus[15]:=KS15; coordinatstatus[16]:=KS16;  
    coordinatstatus[17]:=KS17; coordinatstatus[18]:=KS18;  
    coordinatstatus[19]:=KS19; coordinatstatus[20]:=KS20;  
    coordinatstatus[21]:=KS21; coordinatstatus[22]:=KS22;  
    coordinatstatus[23]:=KS23; coordinatstatus[24]:=KS24;  
    coordinatstatus[25]:=KS25; coordinatstatus[26]:=KS26;  
    coordinatstatus[27]:=KS27; coordinatstatus[28]:=KS28;  
    coordinatstatus[29]:=KS29; coordinatstatus[30]:=KS30;  
    coordinatstatus[31]:=KS31; coordinatstatus[32]:=KS32;  
    coordinatstatus[33]:=KS33; coordinatstatus[34]:=KS34;  
    coordinatstatus[38]:=KS38; coordinatstatus[39]:=KS39;  
    coordinatstatus[40]:=KS40; coordinatstatus[41]:=KS41;  
    coordinatstatus[42]:=KS42; coordinatstatus[46]:=KS46;  
    coordinatstatus[47]:=KS47; coordinatstatus[48]:=KS48;  
    coordinatstatus[49]:=KS49; coordinatstatus[50]:=KS50;  
    coordinatstatus[51]:=KS51; coordinatstatus[52]:=KS52;  
    coordinatstatus[53]:=KS53; coordinatstatus[54]:=KS54;  
    coordinatstatus[55]:=KS55; coordinatstatus[56]:=KS56;  
    coordinatstatus[57]:=KS57; coordinatstatus[58]:=KS58;  
    coordinatstatus[59]:=KS59; coordinatstatus[60]:=KS60;
```

```
SetBit(2,1100,1); {Flagga sätts att nya värden är nerlästa}
```

```
Sattaddr:=130;
```

```
FOR i:=1 TO 60 DO
```

```
    BEGIN
```

```
        IF ((i<>35) AND (i<>36) AND (i<>37) AND (i<>43) AND (i<>44) AND (i<>45))  
            THEN
```

```
                BEGIN
```

```
                    RealValue[1]:=coordinatstatus[i];
```

```
                    SetReg(2,1,Sattaddr);
```

```
                    Sattaddr:=Sattaddr+1;
```

```
                END;
```

```
    END;
```

```
END;
```

```
{-----Send Feeding Stores-----}
```

```
PROCEDURE Send_Feeding_Stores(feedlist1,feedlist2,feedlist3,feedlist4,
```

Bilagor

```
                feedlist5,feedlist6,feedlist7,feedlist8,feedlist9,
                feedlist10,feedlist11,feedlist12,matterhorn :INTEGER);

VAR Sattaddr,i : INTEGER;
    feedstore : ARRAY[1..13] OF INTEGER;

BEGIN
    feedstore[1]:=feedlist1; feedstore[2]:=feedlist2;
    feedstore[3]:=feedlist3; feedstore[4]:=feedlist4;
    feedstore[5]:=feedlist5; feedstore[6]:=feedlist6;
    feedstore[7]:=feedlist7; feedstore[8]:=feedlist8;
    feedstore[9]:=feedlist9; feedstore[10]:=feedlist10;
    feedstore[11]:=feedlist11; feedstore[12]:=feedlist12;

    i:=0;
    Sattaddr:=200;
    SetBit(2,1101,1);

    REPEAT
        i:=i+1;
        RealValue[1]:=feedstore[i];
        SetReg(2,1,Sattaddr);
        Sattaddr:=Sattaddr+1;
    UNTIL (feedstore[i]=9) ;
    SetBit(2,1102,matterhorn);
END;

                {-----Send Control Status-----}

PROCEDURE Send_Control_Status(rokgestempmin,framledningstempmin,
                             C0max,reglerlevel : INTEGER);
BEGIN
    SetBit(2,1115,reglerlevel);          {reglering AV-PÅ}

    RealValue[1]:=rokgestempmin;
    SetReg(2,1,225);

    RealValue[1]:=framledningstempmin;
    SetReg(2,1,226);

    RealValue[1]:=C0max;
    SetReg(2,1,227);

    SetBit(2,1114,1);                    {Nya värden nerlästa}
END;

                {-----Set Reeload Store-----}

PROCEDURE Set_Reeload_Store(reloadstore,omlastning :INTEGER);
BEGIN
    SetBit(2,1103,omlastning);          {Omlastning AV/PÅ}
    RealValue[1]:=reloadstore;
    SetReg(2,1,220);
END;

                {-----Set Traverse Status-----}

PROCEDURE Set_Traverse_Status(TraversFixPosition,TillUtg_ngsl_ge :INTEGER);
BEGIN
    SetBit(2,1104,TraversFixPosition);
    SetBit(2,1105,TillUtg_ngsl_ge);
END;

                {-----Set Program data-----}

PROCEDURE Set_Program_data(matterhorn_kning,ultraljudstid :INTEGER);
BEGIN
    RealValue[1]:=matterhorn_kning;
    SetReg(2,1,221);
    RealValue[1]:=ultraljudstid;
    SetReg(2,1,222);
```

Bilagor
END;

{-----Get Coord Status-----}

```
PROCEDURE Get_Coord_Status(VAR KS1,KS2,KS3,KS4,KS5,KS6,KS7,KS8,KS9,KS10,  
    KS11,KS12,KS13,KS14,KS15,KS16,KS17,KS18,KS19,KS20,  
    KS21,KS22,KS23,KS24,KS25,KS26,KS27,KS28,KS29,KS30,  
    KS31,KS32,KS33,KS34,KS38,KS39,KS40,KS41,KS42,KS46,  
    KS47,KS48,KS49,KS50,KS51,KS52,KS53,KS54,KS55,KS56,  
    KS57,KS58,KS59,KS60 : INTEGER);
```

```
VAR Sattaddr,i : INTEGER;  
    coordinatstatus : ARRAY[1..60] OF INTEGER;
```

BEGIN

```
Sattaddr:=130;  
FOR i:=1 TO 60 DO  
    BEGIN  
        IF ((i<>35) AND (i<>36) AND (i<>37) AND (i<>43) AND (i<>44) AND (i<>45))  
            THEN  
                BEGIN  
                    GetReg(2,1,Sattaddr);  
                    coordinatstatus[i]:=Trunc(RealValue[1]);  
                    Sattaddr:=Sattaddr+1;  
                END;  
    END;  
END;
```

```
KS1:=coordinatstatus[1]; KS2:=coordinatstatus[2];  
KS3:=coordinatstatus[3]; KS4:=coordinatstatus[4];  
KS5:=coordinatstatus[5]; KS6:=coordinatstatus[6];  
KS7:=coordinatstatus[7]; KS8:=coordinatstatus[8];  
KS9:=coordinatstatus[9]; KS10:=coordinatstatus[10];  
KS11:=coordinatstatus[11]; KS12:=coordinatstatus[12];  
KS13:=coordinatstatus[13]; KS14:=coordinatstatus[14];  
KS15:=coordinatstatus[15]; KS16:=coordinatstatus[16];  
KS17:=coordinatstatus[17]; KS18:=coordinatstatus[18];  
KS19:=coordinatstatus[19]; KS20:=coordinatstatus[20];  
KS21:=coordinatstatus[21]; KS22:=coordinatstatus[22];  
KS23:=coordinatstatus[23]; KS24:=coordinatstatus[24];  
KS25:=coordinatstatus[25]; KS26:=coordinatstatus[26];  
KS27:=coordinatstatus[27]; KS28:=coordinatstatus[28];  
KS29:=coordinatstatus[29]; KS30:=coordinatstatus[30];  
KS31:=coordinatstatus[31]; KS32:=coordinatstatus[32];  
KS33:=coordinatstatus[33]; KS34:=coordinatstatus[34];  
KS38:=coordinatstatus[38]; KS39:=coordinatstatus[39];  
KS40:=coordinatstatus[40]; KS41:=coordinatstatus[41];  
KS42:=coordinatstatus[42]; KS46:=coordinatstatus[46];  
KS47:=coordinatstatus[47]; KS48:=coordinatstatus[48];  
KS49:=coordinatstatus[49]; KS50:=coordinatstatus[50];  
KS51:=coordinatstatus[51]; KS52:=coordinatstatus[52];  
KS53:=coordinatstatus[53]; KS54:=coordinatstatus[54];  
KS55:=coordinatstatus[55]; KS56:=coordinatstatus[56];  
KS57:=coordinatstatus[57]; KS58:=coordinatstatus[58];  
KS59:=coordinatstatus[59]; KS60:=coordinatstatus[60];
```

END;

{-----Get Feeding Stores-----}

```
PROCEDURE Get_Feeding_Stores(VAR feedlist1,feedlist2,feedlist3,feedlist4,  
    feedlist5,feedlist6,feedlist7,feedlist8,feedlist9,  
    feedlist10,feedlist11,feedlist12 : INTEGER);
```

```
VAR Sattaddr,i : INTEGER;  
    feedstore : ARRAY[1..13] OF INTEGER;
```

BEGIN

```
i:=0;  
Sattaddr:=200;
```

```
REPEAT  
    i:=i+1;
```

Bilagor

```
GetReg(2,1,Sattaddr);
feedstore[i]:=TRUNC(RealValue[1]);
Sattaddr:=Sattaddr+1;
UNTIL (feedstore[i]=9) ;
```

```
feedlist1:=feedstore[1]; feedlist2:=feedstore[2];
feedlist3:=feedstore[3]; feedlist4:=feedstore[4];
feedlist5:=feedstore[5]; feedlist6:=feedstore[6];
feedlist7:=feedstore[7]; feedlist8:=feedstore[8];
feedlist9:=feedstore[9]; feedlist10:=feedstore[10];
feedlist11:=feedstore[11]; feedlist12:=feedstore[12];
```

END;

```
{-----Get Reload Store-----}
```

```
PROCEDURE Get_Reload_Store(VAR omlastning : INTEGER);
```

```
BEGIN
```

```
GetBit(2,1103,omlastning);
```

```
END;
```

```
{-----InitTravers-----}
```

```
PROCEDURE InitTravers;
```

```
BEGIN
```

```
RealValue[1]:=2;
```

```
SetReg(2,1,200);
```

```
RealValue[1]:=9;
```

```
SetReg(2,1,201);           {Inmatningsficka:=A}
```

```
RealValue[1]:=4;
```

```
SetReg(2,1,222);           {Ultraljudsgivaren 4s}
```

```
END;
```

```
{-----Get Matterhorn Status-----}
```

```
PROCEDURE Get_Matterhorn_Status(VAR Matterhorn : INTEGER);
```

```
BEGIN
```

```
GetBit(2,1102,matterhorn);
```

```
END;
```

END.

※

9.5 BILAGA SIGNALERNA UT MOT ANLÄGGNINGEN

Analoga in-signaler:

| <u>Benämning</u> | <u>Område</u> | <u>Typ</u> | <u>SC15Address</u> | <u>Ledare</u> | |
|---------------------------------|---------------|------------|--------------------|---------------|---------|
| Framtemperatur | 0-200 | C | 0-20 mA | AI640 | 290-291 |
| Returtemperatur | 0-200 | C | 0-20 mA | AI644 | 292-293 |
| Ugnstemperatur | 400 | | 4-20 mA | AI650 | 294-295 |
| | -1500 | C | | | |
| O2-halt | 0-20 % | | 4-20 mA | AI654 | 276-277 |
| CO-halt | 0 | | 4-20 mA | AI660 | 296-297 |
| | -5000 ppm | | | | |
| Delta-p | 0-100 % | | 4-20 mA | AI664 | 298-299 |
| Eldstads- undertryck | 0-100 Pa | | 0-20 mA | AI670 | 300-301 |
| Rökgas- temperatur1 | 0-400 C | | 0-20 mA | AI700 | 278-279 |
| Rökgastemperatur economeiser | 0-400 C | | 0-20 mA | AI704 | 302-303 |
| Panntemperatur | 0-160 C | | Pt 160 | AI710 | 302-303 |
| Läge | 0-100% | | 0-10 V | AI714 | 304-305 |
| Primärspjäll | | | | | |
| Läge | 0-100 % | | 0-10V | AI720 | 306-307 |
| Tertiärspjäll | | | | | |
| Läge | 0-100 % | | 0-140Ohm | AI724 | 286-287 |
| Sekundärspjäll | | | | | |

Digitala Ut-signaler:

| <u>Benämning</u> | <u>Område</u> | <u>Typ</u> | <u>SC15Addr</u> | <u>Ledare</u> | |
|----------------------|---------------|------------|-----------------|---------------|-----|
| Roster | 24 | VDC | D-Ut | 100 | 427 |
| Ljudsota | 24 | VDC | D-Ut | 101 | 430 |
| Primärfläkt | 24 | VDC | D-Ut | 102 | 428 |
| Tertiärfläkt | 24 | VDC | D-Ut | 103 | 429 |
| Bypassa filtret | 220 | VAC | D-Ut | 300 | 426 |
| Travers - | 220 | VAC | Trav | 301 | 201 |
| Travers + | 220 | VAC | Trav | 302 | 202 |
| Travers högfart | 220 | VAC | Trav | 303 | 203 |
| Tralla - | 220 | VAC | Trav | 304 | 204 |
| Tralla + | 220 | VAC | Trav | 305 | 205 |
| Tralla Högfart | 220 | VAC | Trav | 306 | 206 |
| Lyft upp | 220 | VAC | Trav | 307 | 207 |
| Sänk Ned | 220 | VAC | Trav | 310 | 208 |
| Lyft/Sänk Snabbt | 220 | VAC | Trav | 311 | 209 |
| Öppna Skopa | 220 | VAC | Trav | 312 | 210 |
| Stäng Skopa | 220 | VAC | Trav | 313 | 211 |
| Inmatningsskruvar | 220 | VAC | D-Ut | 314 | 432 |
| Bragg | 220 | VAC | D-Ut | 315 | 416 |
| Rökgasfläkt | 220 | VAC | D-Ut | 317 | 419 |
| Filterrensning | 220 | VAC | D-Ut | 340 | 452 |
| Askskruv 1+2+3 | 220 | VAC | D-Ut | 341 | 420 |
| Primärspjäll Öppna | 220 | VAC | D-Ut | 344 | 412 |
| Primärspjäll Stäng | 220 | VAC | D-Ut | 345 | 413 |
| Sekundärspjäll Öppna | 220 | VAC | D-Ut | 346 | 417 |
| Sekundärspjäll Stäng | 220 | VAC | D-Ut | 347 | 418 |
| Teriärspjäll Öppna | 220 | VAC | D-Ut | 350 | 414 |
| Teriärspjäll Stäng | 220 | VAC | D-Ut | 351 | 415 |
| Slaggskruv Container | 220 | VAC | D-Ut | 352 | 423 |
| Slaggskruv Liggande | 220 | VAC | D-Ut | 353 | 424 |
| Slaggskruv | 220 | VAC | D-Ut | 354 | 425 |
| Rökgasspjäll Öppna | 220 | VAC | D-Ut | 356 | 435 |
| Rökgasspjäll Stäng | 220 | VAC | D-Ut | 357 | 436 |

Bilagor

Digitala In-signalier:

| <u>Benämning</u> | <u>Område</u> | <u>Typ</u> | <u>SC15Addr</u> | <u>Ledare</u> |
|-----------------------------|---------------|------------|-----------------|---------------|
| Bränsleficka Full | 24 VDC | Travers | 140 | 437 |
| Bränsleficka Tom | 24 VDC | Travers | 141 | 438 |
| Lågt Lufttryck Kompressor | 24 VDC | Larm | 142 | 281 |
| Utlöst Vakt Inmatning | 24 VDC | Larm | 143 | 282 |
| Topp (tfl, of1, p6, p7) | 24 VDC | Larm | 144 | 448 |
| Kulvert | 24 VDC | Larm | 145 | 445 |
| Vatten Rening | 24 VDC | Larm | 146 | 446 |
| Fläkt Vinden | 24 VDC | Larm | 147 | 447 |
| Temperatur Inmatning | 24 VDC | Larm | 150 | 285 |
| Mintemp | 24 VDC | Larm | 151 | Omärkt |
| Flödes puls | 24 VDC | D-In | 152 | 256 |
| Energi puls | 24 VDC | D-In | 153 | 257 |
| Utl. Motorskydd Primär | 240 VAC(Inv) | Larm | 440 | 64 |
| Betydelse | | | | |
| Bränslebrist | 240 VAC | Larm | 441 | 65 |
| Utlöst sprinkler | 240 VAC(Inv) | Larm | 442 | 66 |
| Låg Oljenivå Roster | 240 VAC(Inv) | Larm | 443 | 67 |
| Hög Temp Roster | 240 VAC(Inv) | Larm | 444 | 68 |
| Omlastning D-ficka | 240 VAC | D-In | 445 | Omärkt |
| Utl. Nedeldningstermostat | 240 VAC(Inv) | Larm | 446 | 69 |
| Utl Maxtermostat | 240 VAC(Inv) | Larm | 447 | 70 |
| Lågt Pannvattentryck | 240 VAC(Inv) | Larm | 450 | 71 |
| Högt Pannvattentryck | 240 VAC(Inv) | Larm | 451 | 72 |
| Låg Pannvattencirkulation | 240 VAC(Inv) | Larm | 452 | 73 |
| Utl. Katastrofskydd | 240 VAC(Inv) | Larm | 453 | 74 |
| Utl Övertrycks-vakt Eldstad | 240 VAC(Inv) | Larm | 454 | 75 |
| Låg Rökgasttemperatur1 | 240 VAC(Inv) | Larm | 455 | 76 |
| Hög rökgasttemperatur1 | 240 VAC(Inv) | Larm | 456 | 77 |
| Utl. Motorskydd Sekundär | 240 VAC | Larm | 457 | 78 |
| Betydelse | | | | |
| Travers Utgående | 240 VAC | Trav | 500 | 20 |
| Trav + | 240 VAC | Trav | 501 | 21 |
| Stoppgivare Travers | 240 VAC | Trav | 502 | 22 |
| Tralla Utgående | 240 VAC | Trav | 503 | 23 |
| Tralla- | 240 VAC | Trav | 504 | 24 |
| Stoppgivare Tralla | 240 VAC | Trav | 505 | 25 |
| Nivå S21 | 240 VAC | Trav | 506 | 26 |
| Nivå S22 | 240 VAC | Trav | 507 | 27 |
| Nivå S23 | 240 VAC | Trav | 510 | 28 |
| Nivå S24 | 240 VAC | Trav | 511 | 29 |
| Noll Last Travers | 240 VAC | Trav | 512 | 30 |
| Trav - | 240 VAC | Trav | 513 | 31 |
| Tralla + | 240 VAC | Trav | 514 | 32 |
| Datorstyrning på | 240 VAC | D-In | 515 | 515 |
| Travers Auto/Man | 240 VAC | D-In | 516 | 516 |
| Hög Rökgasttemperatur2 | 240 VAC | Larm | 540 | 79 |
| Utl. Slaggskruv skydd | 240 VAC | Larm | 541 | 80 |
| Rökgasfläkt Motorskydd | 240 VAC | Larm | 542 | 200 |
| Stoft Problem | 240 VAC | Larm | 543 | 439 |
| Filter Fickor | 240 VAC | Larm | 544 | 206 |
| Grovavskiljare | 240 VAC | Larm | 545 | 205 |
| Rökgasttemperatur3 | 240 VAC | Larm | 546 | 209 |
| Diff Tryck Vakt Nät | 240 VAC | Larm | 547 | 211 |
| Stora Cirkulationspumpen | 240 VAC | D-In | 550 | 212 |
| Lilla Cirkulationspumpen | 240 VAC | D-In | 551 | 213 |
| Stäng Av Roster | 240 VAC | D-In | 552 | 552 |
| Stäng Av Inmatning | 240 VAC | D-In | 553 | 553 |
| Stäng Av Askutmatning | 240 VAC | D-In | 554 | 554 |
| Stäng Av Filterrensning | 240 VAC | D-In | 555 | 555 |
| Stäng Av Ljudsotning | 240 VAC | D-In | 556 | 556 |
| Hög CO-halt | 240 VAC | Larm | 557 | Omärkt |

Bilagor

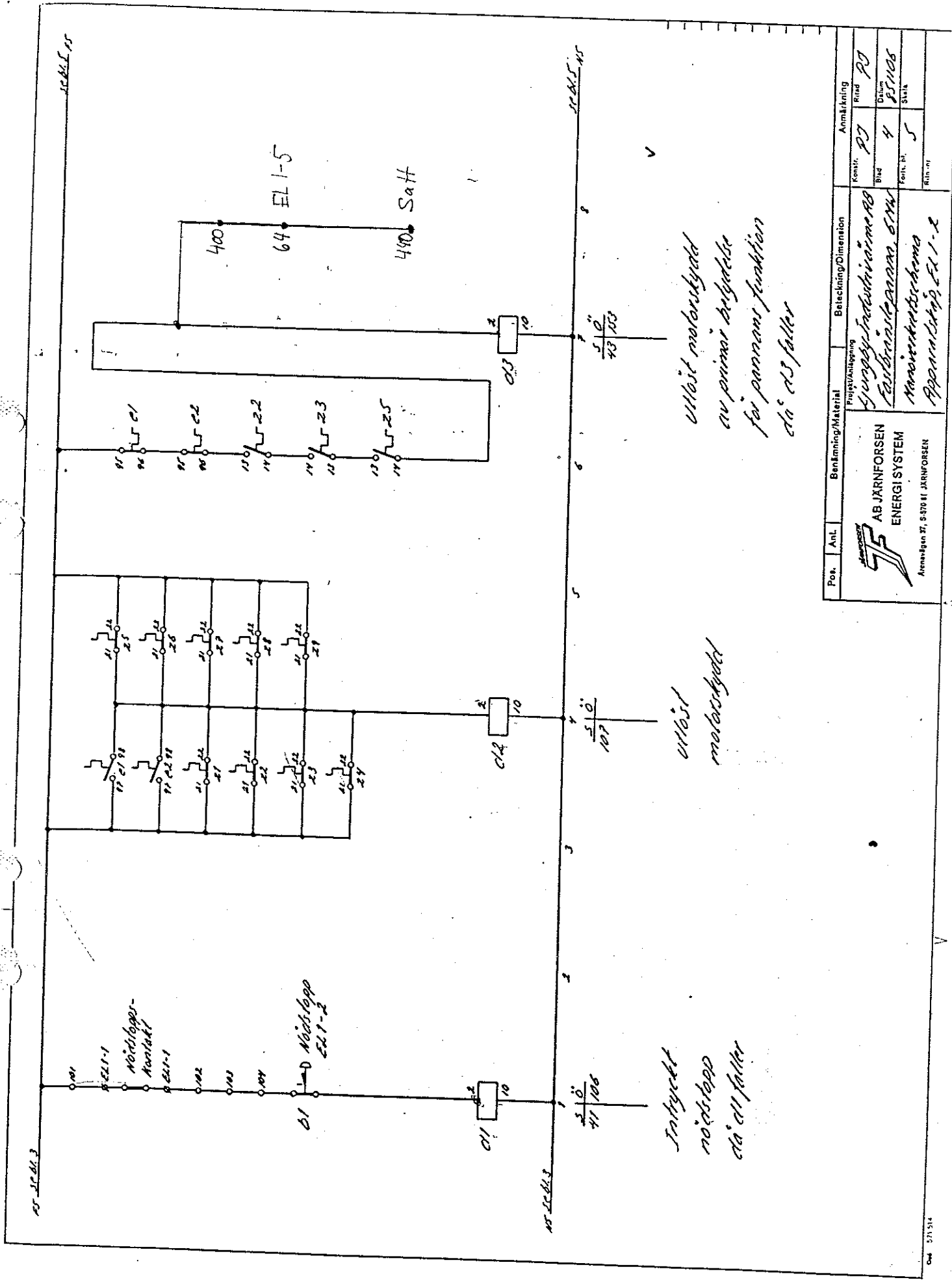
9.6 INSTALLERING, OMINSTALLERING, HUR MAN ÄNDRAR I PROGRAMMEN


Hur man ändrar i PBS programmet

Förr eller senare kommer man att vilja ändra någonting i pannaprogrammet. Om man har programmerat något av SattControls system förut så skall detta inte medföra några problem. Om man är lite grön så kan det vara till hjälp att läsa igenom detta avsnitt.

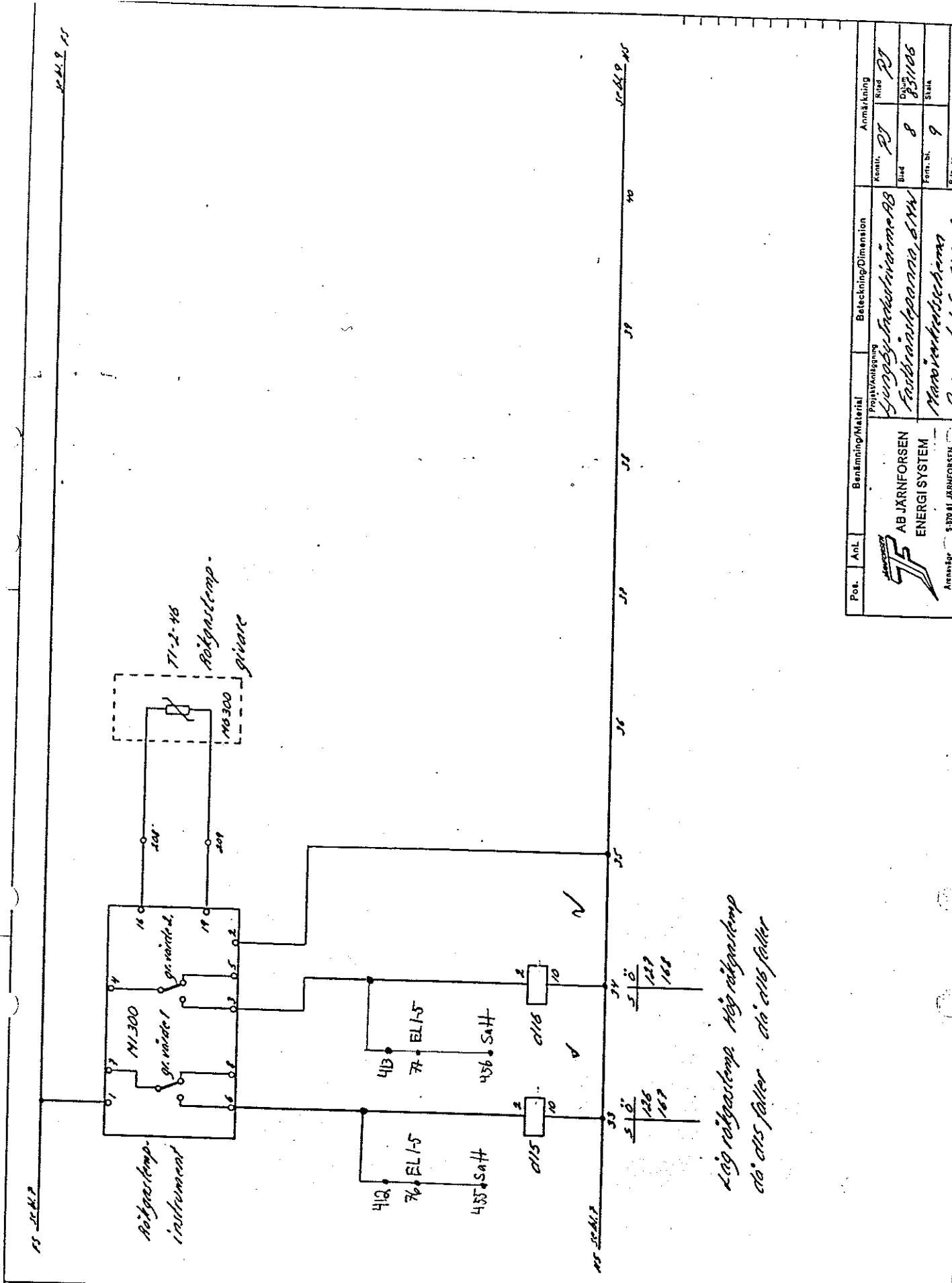
Det första man måste göra är att ladda över Sattprog disketterna (4 st) till en katalog i en hårdisk på en PC-DOS maskin. I och med denna kopiering har man tillgång till programmeringsverktyget DOX5 samt hela pannprog-rammet sattprog. Genom att skriva *X5 sattprog* så öppnar du editorn som ger dig tillgång till huvudprogrammet. Om du vill ändra i programkoden i någon modul så måste du först hoppa till önskad modul. Detta gör du (ex. mätvärde) genom skriva *NAME MÄTVÄRDE*. För att gå in och ändra i PBS-koden skriver du *PBS* och därefter *E*. Du är nu inne i editorn och kan bläddra dig fram i programmet genom att trycka *F9* (framåt) och *F8* (bakåt). Om du vill foga in en sats så gör det genom att trycka *F10*. När du lagt till dina ändringar eller kompletteringar så måste du gå ur PBS-editorn genom att trycka på *ctrl-q*. Efter detta är det dags för kompilering. Då måste du först hoppa till huvudprogrammet och det gör du genom att skriva *NAME SATTPROG*. Sedan kan man kompilera det genom att skriva *MAKE*. Därefter måste du koppla upp en förbindelse från PC:n till SattCon. Detta gör du genom att koppla in en speciell DOX5 kabel i pc:ns com1 och SattCons port märkt *VDU*. Kabeln som är avsedd för detta finns i skåpet till SattCon 15-20. Det sitter redan en kabel i denna port märkt *Com1* men den måste man ta bort så länge. När kabeln är uppkopplad så är det dags för att överföra det nya PBS-programmet. Det gör du genom att skriva *SEND*. Efter ett tag så är överföringen klar och det enda du behöver göra är att avsluta DOX5 genom att skriva *Q*. Därefter tar man bort kabeln mellan SattCon 15 och PC:n och sätter tillbaka den ordinarie Comlikabeln. Efter det att man laddat ner ett program måste man även återstarta systemet genom att vrida nyckeln på skåpet tillbaka och fram igen samt

Bilagor
att trycka på reset knappen till PC:n där
traversprogrammet ligger.



| Pos. | Ant. | Benämning/Material | Projektskissning | Belysning/Dimension | Anmärkning |
|------|------|--|--|---------------------|---|
| | |  AB JÄRNFORSEN ENERGI SYSTEM Annarsvägen 37, S-870 81 JÄRNFORSEN | Snygg-industrierna AB Lärjansvägen 57M Malmö Apparatbyggnad, EL 1-2 | | Rind P3 Datum 4 8/106 Sida 5 Ritn. nr |
| | | | | | |

sk. 9 45



Låg rökgastemp. Höj rökgastemp.
 då 415 faller då 416 faller

| Pos. | Anl. | Benämning/Material | Beteckning/Dimension | Antal | Skid | Anteckning |
|------|------|--------------------------------|-------------------------|-----------|------|--------------|
| | | AB JÄRNFORSEN ENERGI SYSTEM | | | | |
| | | Projekt/Anläggning | Lyngby Industriområde B | | | |
| | | | Facklinjanspänning 6 MV | | | |
| | | | Maskinverkstaden | | | |
| | | | Apparatkap. EL1-2 | | | |
| | | | | Blad | 8 | Datum 83/106 |
| | | | | Form. Sk. | 9 | Skala |
| | | | | Ritn.-nr | | |

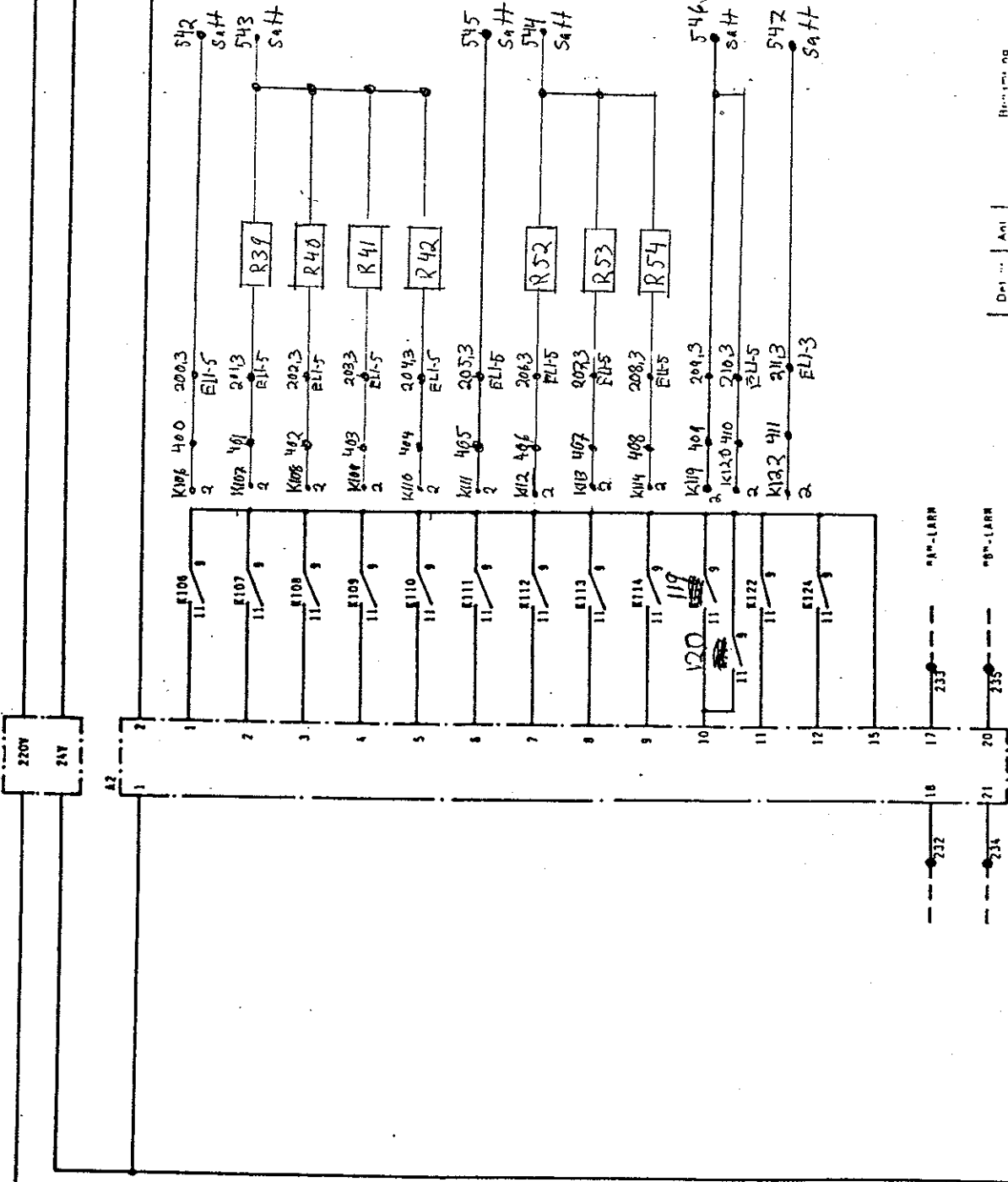
ANKOMD 4 DEC. 1960

B 270V AC
B26

TRANSFORMATOR
220/24V AC

LAMPARELL

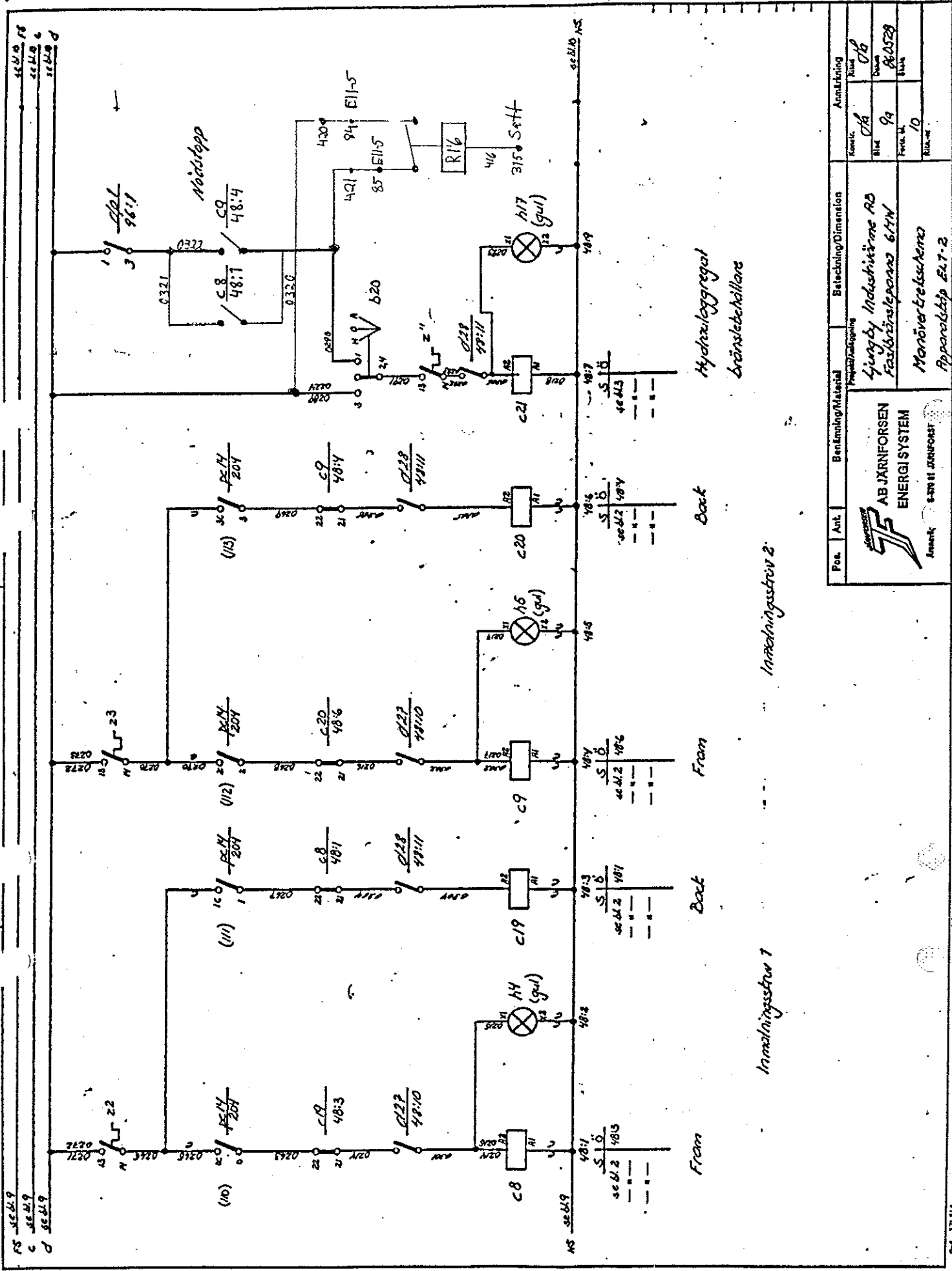
- E106 RÖKGASFLÄKT
- E107 VÄRTALSVAKT
- E108 STÖFTSKRUV 1 Liögårds
- E109 VÄRTALSVAKT
- E110 STÖFTSKRUV 2 Conzainer
- E111 VÄRTALSVAKT
- E112 STÖFTSKRUV 3 Stofskruva 3
- E113 VÄRTALSVAKT
- E114 GROVAVSEILJÄRE
- E115 RIVÄVAKT
- E116 FILTRIFICER 1
- E117 RIVÄVAKT
- E118 FILTRIFICER 2
- E119 RIVÄVAKT
- E120 FILTRIFICER 3
- E121 RIVÄVAKT
- E122 RÖKGASTEMPERATUR
- E123 LÅG/NÖS
- E124 DIFFERENSTRÖMVAKT
- E125 SLANGFILTIER
- E126 RIVÄVAKT



| Del | Anl | Benämning | Material | Mod. Anl. | Ann |
|--|-----|-----------|----------|-----------|-----------|
| FASTBRÄNSLEPANNAN - RÖKGASRENINGSGRÄNSUTRUSTNING | | | | | |
| APPARATSKÅP | | | | | |
| MANGVERKRETSSCHEMA | | | | | |
| FEL OCH LARMINDIKERING | | | | | |
| FTN | | 85-86-01 | | | |
| | | | | | 9317-1116 |

EL 13

Den här ritningen är ett tekniskt dokument och ska användas som sådan. För ytterligare information eller detaljer kontakta ansvarig tekniker.



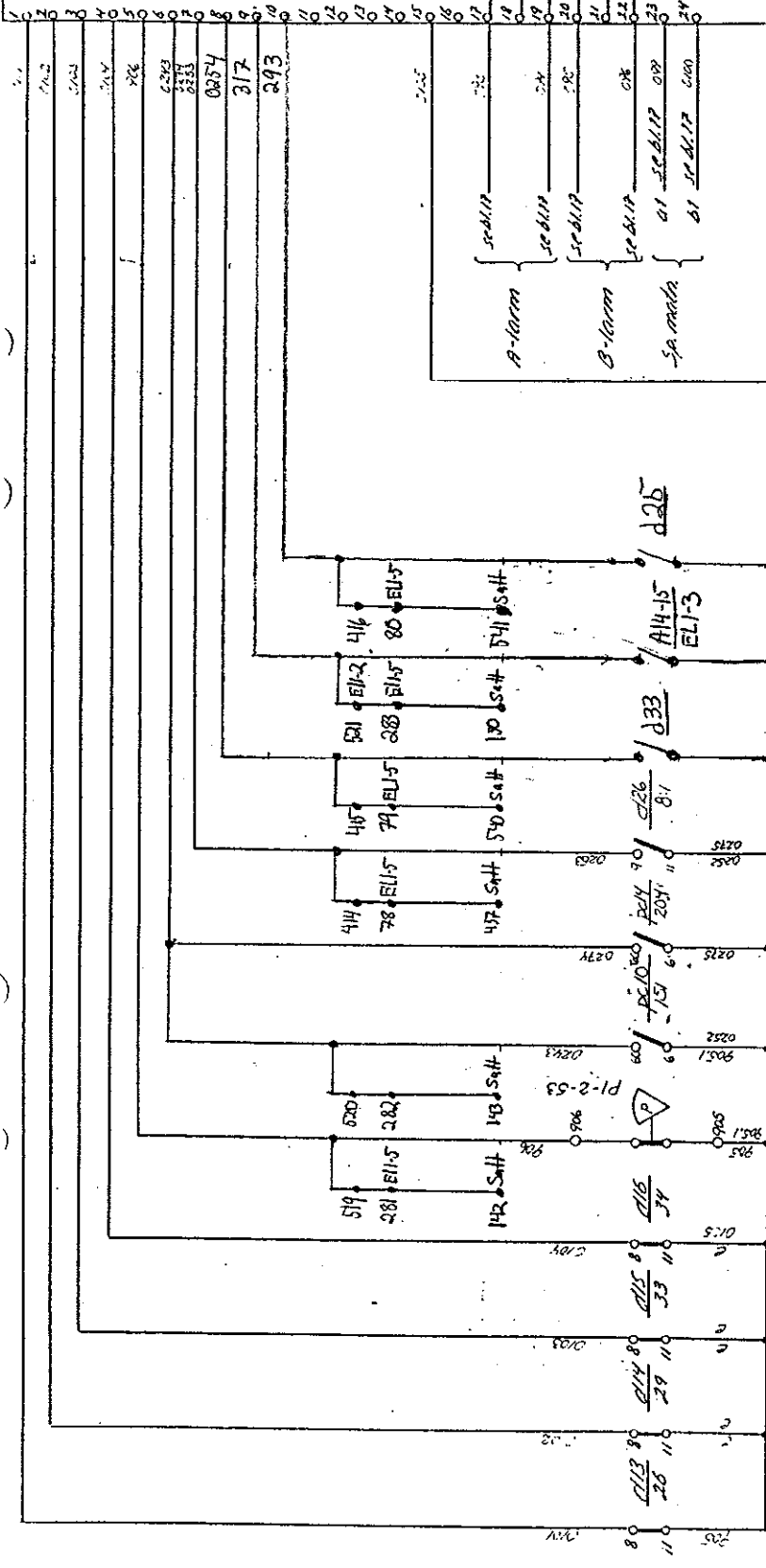
Innredningsströmv 1

Innredningsströmv 2

| Pos. | Ant. | Benämning/Material | Besteckning/Dimension | Årsmärkning |
|------|------|--------------------------------|---|---|
| | | Projektslagning | | |
| | | AB JÄRNFORSEN ENERGI SYSTEM | Lyngby Industrivägs RS Faskbränslepans 61111 | Årsnr. 08 Blad 08 Blad 99 Fas. 10 Ritning |
| | | Årsnr. 8-898 81 JÄRNFORSEN | Manöverbetsvakt Apparatslip EL7-2 | |

PU-3512B

Lamcentral 2.



124 Utöst kontakt
 125 Utöst övertrycks-
 vakt elstod
 126 Låg rökgastemp.
 127 Högrökgastemp.
 128 Löst källtryck
 129 Kompressor
 Utöst överbelastningsvakt
 130 inmarning
 131 Utöst motorskydd
 or sekundär behållare
 132 Högrökgastemp. 2
 133 Högt temp. in marning
 134 Utöst övertryck

| Pos. | Ant. | Benämning/Material | Beteckning/Dimension | Anmärkning |
|------|------|---|----------------------|------------|
| | | AB JÄRNFORSEN ENERGI SYSTEM Arenavägen 1, S-570 81 JÄRNFORSEN | | |
| | | Projekts/Anläggning | | |
| | | Långby Industrivarme AB | | Kontnr. PJ |
| | | Fälthälsöpparna 6 MW | | Blad 10 |
| | | Manöverkretsarna | | Blad 18 |
| | | Apparatshop, H. 1-2 | | Blad 19 |
| | | | | Blad 85107 |
| | | | | Blad 85107 |
| | | | | Blad 19 |
| | | | | Blad 19 |

15-SCB12
C-SCB12
C1-SCB12

15-SCB12
C-SCB12
C1-SCB12

20/2
2/100

100/3
3/100

40/4
4/100

0910 X09
014 024

0910 X09
013 024

0910 X09
014 024

437 101 E115
130 14 29

439 103 E115
14 14 24

438 102 E115
130 14 21

400 34 S4H
C15 R1

424 353 S4H
C10 R1

423 352 S4H
C7 R1

A12 (9W)
X1 X2 R2

A13 (9W)
X1 X2 R2

A14 (9W)
X1 X2 R2

15-SCB12

21

22

23

24

25

26

27

28

29

30

5 10
A12
-11-
-11-
185


5 10
A12
-11-
-11-
185

5 10
A12
-11-
-11-
184

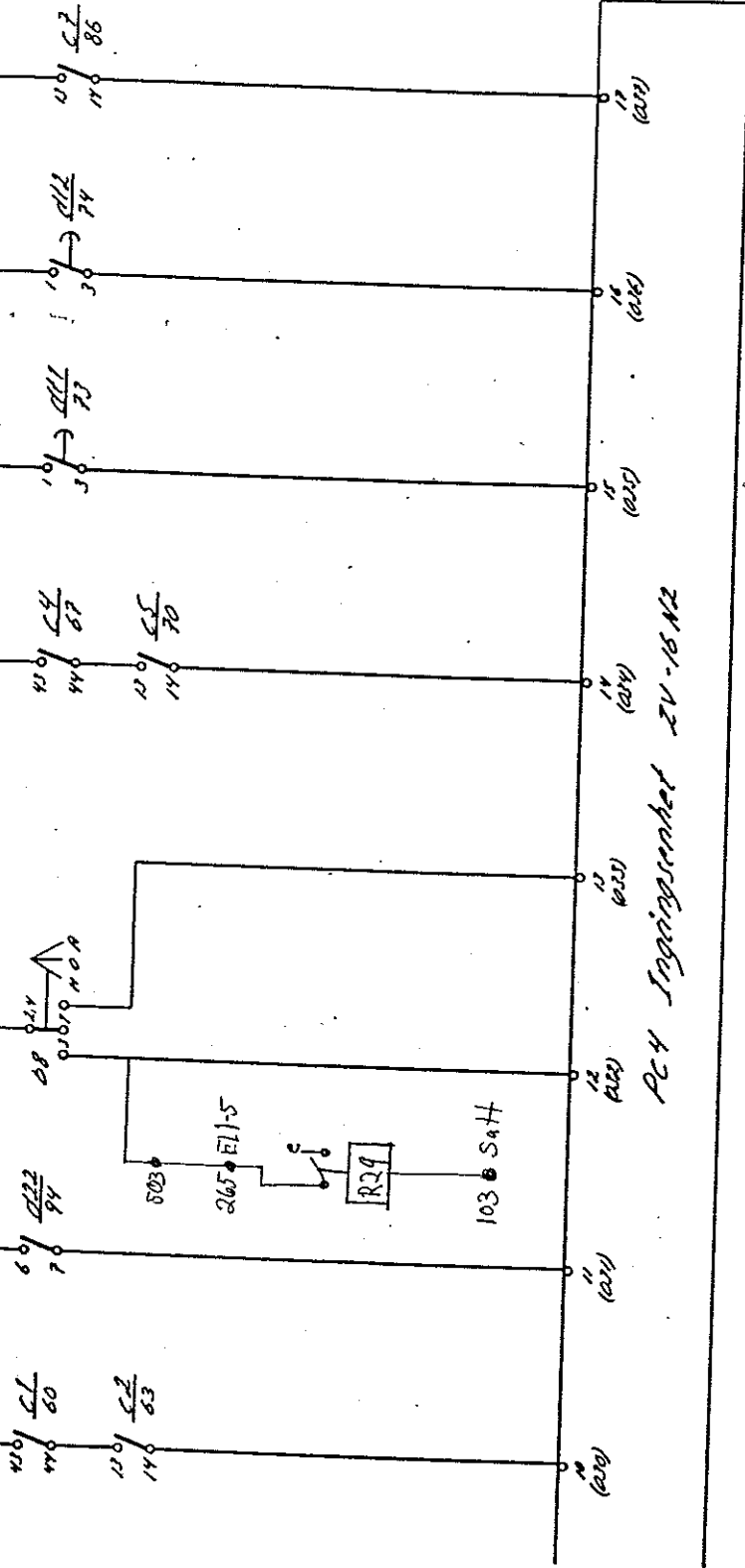
Arkiv 3

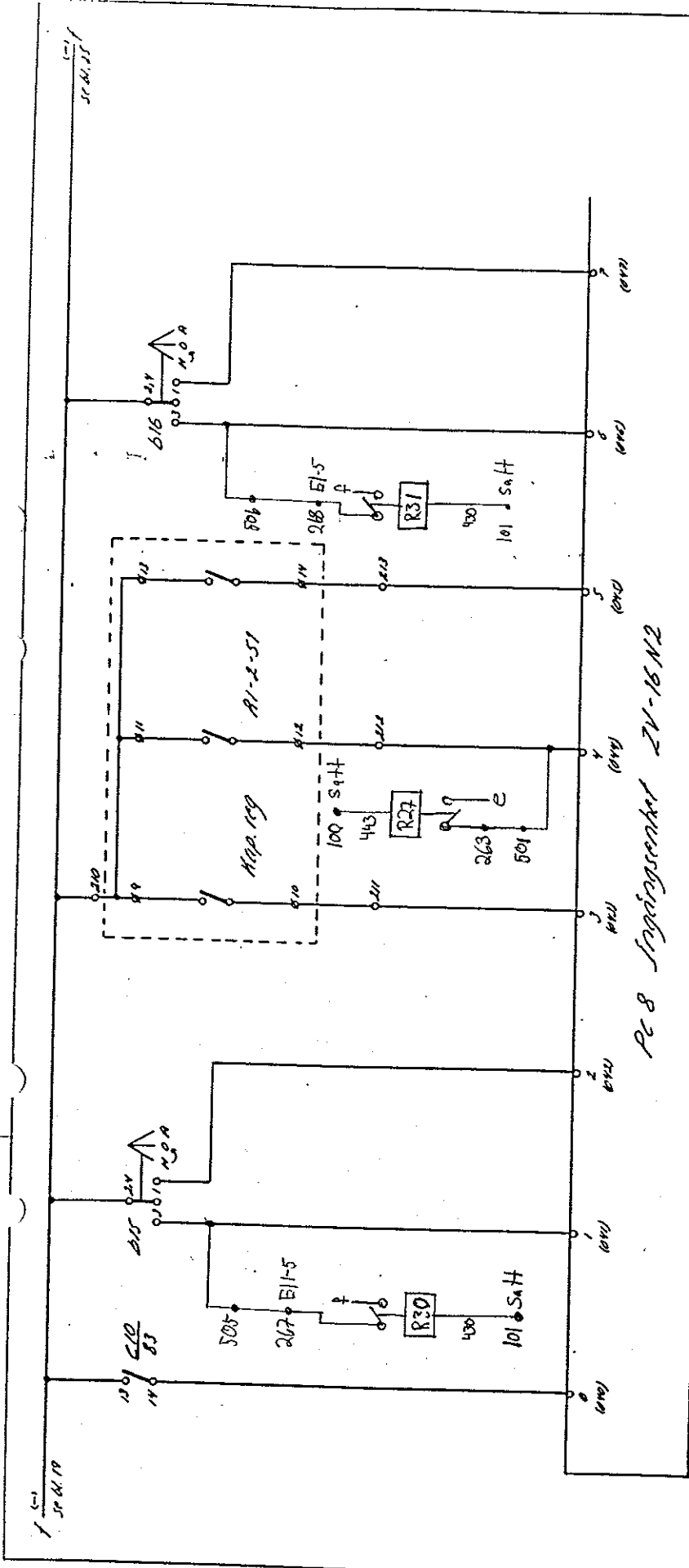
Stoggbruk
liggande

Stoggbruk
container

| Pos. | Ant. | Benämning/Material | Proj/Anläggning | Bezeichnung/Dimension | Anmärkning |
|------|------|--|---|-----------------------|---|
| | |  AB JÄRNFORSEN ENERGI SYSTEM Ansvärd: 081 JÄRNFORSEN | Gungby Industriområde AB Fastbränslepanna 6 MW Manöverutrustnings apparatshop, ELS-2 | | Konst. RJ Bild 14 Fört. bl. 15 Rikt. |
| | | | | | |

5-1
20.01.22





PC 8 Ingenjörskontroll 2V-16.N2

185 186 187 188 189 190 191 192
 Staggskruv Hand Auto Rannstopp. Start Hand Auto
 liggande Tyfon 1 & 2 Panna höger an start Tyfon Tyfon
 i drift Tyfon 1 & 2 drifttemp. ruter inmatning system ekonomiser

| Pos. | Ant. | Benämning/Material | Beledning/Dimension | Anmärkning |
|------|------|-------------------------------|---------------------|---------------|
| | | AB JÄRNFORSN ENERGISYSTEM | | Kont. Rind PJ |
| | | Arbetsgrupp: S-5701 JÄRNFORSN | | Blad 24 |
| | | | | Datum 85/108 |
| | | | | Stata |
| | | | | Förh. Bl. 25 |
| | | | | Ritning |