

CODEN: LUTFD2/ (TFRT -5325) /1-070/ (1985)

AUTONOM REGLERCENTRAL

STEFAN NILSSON  
TOR SJÖDIN

INSTITUTIONEN FÖR REGLERTEKNIK  
LUNDS TEKNISKA HÖGSKOLA  
JUNI 1985

<b>LUND INSTITUTE OF TECHNOLOGY</b> DEPARTMENT OF AUTOMATIC CONTROL Box 118 S 221 00 Lund Sweden		Document name Report
Author(s) NILSSON Stefan SJÖDIN Tor		Date of issue June 1985
Supervisor Björn Wittenmark. Michael Lundh Sponsoring organization		Document number CODEN: LUTFD2/(TFRT-5323)/1-070/(1985)
Title and subtitle AUTONOM REGLERCENTRAL • (Stand alone controller)		
Abstract Constructing a PID-controller by using Intel's 8052-AH Basic makes way for a wide range of realtime applications. Implemented in this construction is a foreground process consists of operator communication while the background process contains the regulator algorithm. This makes it possible for the user to change parameters during program execution in real-time. Control over the system can be activated either by using a button-matrix-hand terminal together with an LCD, or by using a usual monitor terminal.		
Key words		
Classification system and/or index terms (if any)		
Supplementary bibliographical information		
ISSN and key title		ISBN
Language Swedish	Number of pages 70	Recipient's notes
Security classification		

RAPPORT

Examensarbete  
AUTONOM REGLERCENTRAL

Examensarbetet är utfört av:

STEFAN NILSSON  
TOR SJÖDIN

Handledare:

MARTIN JERVILL      ElektroSandberg AB  
MICHAEL LUNDH      Inst. för Reglerteknik LTH  
BJÖRN WITTENMARK    Inst. för Reglerteknik LTH

VÅREN 1985

## FÖRORD

Denna rapport innehåller en beskrivning av ett examensarbete utfört av Stefan Nilsson och Tor Sjödin. Vår uppgift gick ut på att konstruera en "Autonom Reglercentral" som företaget ElektroSandberg AB i Malmö har för avsikt att använda som ett utvecklingssystem.

Hårdvaran har vi huvudsakligen arbetat med i Malmö, medan mjukvaran testats och framtagits i rum 1212, M-huset på LTH.

Ett stort tack till vår handledare på ElektroSandberg, Martin Jervill samt den övriga personalen där.

Ett stort tack vill vi också ge våra handledare på Inst. för Regler teknik vid LTH, Michael Lundh och Björn Wittenmark.

LUND JUNI 1985

## INNEHÅLLSFÖRTECKNING

SIDA

1.	SAMMANFATTNING.....	3
2.	ARBETSGANG.....	4-6
3.	HÅRDVARUSPECIFIKATION.....	7-12
3.1	PROCESSORN.....	7
3.2	EXTERNA MINNEN.....	8
3.2.1	RAM.....	8
3.2.2	E2PROM.....	8
3.3	I/O-ENHETER.....	9-11
3.3.1	IMPORT.....	9
3.3.2	UTPORTAR.....	9
3.3.3	D/A-OMVANDLARE.....	10
3.3.4	A/D-OMVANDLARE.....	10
3.3.5	LCD.....	11
3.4	ÖVRIGA ENHETER.....	11-12
3.4.1	KNAPPSATS.....	11
3.4.2	FRONTPANEL.....	12
3.4.3	LARM.....	12
4.	MJUVARUSPECIFIKATION.....	13-16
4.1	REGULATORALGORITMEN.....	14-15
4.2	REALTIDSPROBLEMATIK.....	16
5.	UTVÄRDERING.....	17
5.1.1	KOMMENTAR.....	17
5.1.2	PLANERING.....	17
6.	REFERENSER.....	18

7.	RITNINGAR	
7.1	PROCESSOR OCH EXTERNA MINNEN	
7.2	I/O-ENHETER	
7.3	LARM-ENHET	
7.4	KNAPPSATS OCH LARM-ENHET	
7.5	LCD	
7.6	FRONT PANEL	
7.7	KRETS OCH KOMPONENTPLACERING	
8.	BILAGOR	
8.1	SPECIFIKATION AV EXAMENSARBETE	
8.2	FLÖDESSCHEMA ÖVER PROGRAMVARA	
8.2.1	VERBAL BESKRIVNING AV PROGRAMVARAN	
8.3	LISTA ÖVER VARIABLERS ANVÄNDNING	
8.3.1	LISTA ÖVER KVARVARANDE VARIABLER	
8.4	LISTNING AV PROGRAMVARA	
8.5	KOMPONENT-LISTA	
8.6	GRAF OCH TILLSTÄNDSKODNING	
9.	MANUAL	I-VIII
9.1	KNAPPSATSENS FUNKTION	I-V
9.1.1	KNAPPSATSENS UTSEENDE	I
9.1.2	FUNKTION	I
9.1.3	KNAPPARNAS FUNKTION	II
9.1.4	EXEMPEL	III
9.1.5	PARAMETER KODER	IV
9.1.6	FELMEDDELANDEN	V
9.2	BRUKSANVISNING	VI-VIII
9.2.1	UPPSTART	VI
9.2.2	REGLERING	VII
9.2.3	SKRIVARE	VIII

## 1. SAMMANFATTNING

ElektroSandberg AB i Malmö är ett starkt expansivt företag. Företaget har en utvecklingsavdelning vars huvudsakliga inriktning är övervakning, mätning och styrning av sjukhus, processindustri, fastigheter mm. Utvecklingsavdelningen har relativt nyligen gett sig in på det regler tekniska området. Detta har lett fram till ett examensarbete, nämligen att konstruera en PID-regulator med hjälp av en ny en-kapsel-dator, Intels 8052 AH-BASIC. För att systemet skulle bli komplett, för en PID-regulator, har vi implementerat en 12-bitars analog-digital omvandlare och en 8-bitars digital-analog omvandlare.

En process skall kunna regleras i realtid, samtidigt som operatören skall kunna ändra de i processen ingående parametrarna. Under tiden som programmet exekveras, skall antingen en monitor-terminal eller en hand-terminal, (knappsats), användas. Operatören kan då ändra eller granska de inställda parametrarna. Resultatet presenteras på monitor-skärmen då terminal används och på en LCD då knappsatsen används.

## 2. ARBETSGÅNG

I början av projektet hade vi en sammankomst på institutionen för reglerteknik på LTH tillsammans med Unv. lektor Björn Wittenmark och civ. ing. Martin Jervill. Där diskuterade vi de mål som företaget önskade uppnå med examensarbetet, se bilaga 8.1. På detta möte kom vi fram till att vi troligtvis inte skulle hinna fram till byggandet av LCD och knappats. Vi kom också fram till att det vore lämpligt att lägga punkt 3 (se bilaga 8.1.1) som ett separat examensarbete.

Efter att ha läst igenom de datablad som det fanns att tillgå om processorn 8052AH-BASIC, började vi med att koncentrera oss på hårdvaran. Då det gällde att välja komponenter och kring-kretsar som ingick i konstruktionen var vi till ganska stor del tvungna att hålla oss till företagets interna standard. Efter att ha läst igenom den litteratur som behandlade processorns funktioner, började vi med ritningarna. Då ritningarna var klara valde vi att använda ett europa-kort för att om möjligt få plats med alla komponenterna på ett enda kort.

Därefter skruvade vi ihop ett nätaggregat som med hjälp av ElektroSandbergs standard-kort gav oss rätt spänningsförsörjning till vårt europakort (+/-5v. +/-12v. och jord).

Vi försåg datorn med ett 4\*8kbyte externt minne plus ett 3\*8kbyte E2PROM-minne. Detta gav oss möjlighet till att spara det som vi skrivit i RAM-minnet i ett permanent PROM-minne.

Problem uppstod med spänningsspikar vid av och påsättning av spänningsförsörjningen. Dessa spikar medförde att E2PROM-minnet raderades ofrivilligt. Vi blev därför tvungna att göra en special lösning för att så ej skulle kunna ske (se kap. 3.2.2).



Vi funderade en hel del på hur vi skulle lösa problemet med behandling av in- och utsignaler. Till vårt förfogande hade vi fått en AD574A, en 12-bitars A/D-omvandlare för att ta hand om insignalen. Här stötte vi på problemet med att låta en 8-bitars processor ta emot 12-bitar och sedan lagra detta som ett flyttal. Till problemet hörde också att A/D-omvandlaren behövde en omvandlings-cykel på minst 35µs innan man kunde läsa in det aktuella värdet. Vi hade här två alternativ att lösa detta på. Antingen kunde vi lösa det hela hårdvarumässigt med ett avbrott, som talar om när omvandlings-cykeln är klar eller så skulle vi kunna lösa det helt mjukvarumässigt. Vi valde det senare alternativet (se programvaran bilaga B.4 radnr. 10000,7730,7733 ).

Ytterligare svårigheter med mjukvaran var sådana problemlösningar som kontaktstudsar vid användning av knapp-sats, utskrift av tal på LCD, inläsning av inslaget tal i realtid och lösning av ömsesidig uteslutning. Kontaktstudsarna löste vi genom att använda oss av en sk. "Idiot loop" som läser knappstatusökningen så länge en knapp hålls nedtryckt ( se programraderna 8555-8560 ). Utskrift av tal på LCD löste vi genom att göra om de aktuella parametrarna till ASCII-tecken som sedan lagras i en vektor. När man sedan slår in ett nytt värde till en parameter via knapp-satsen, görs det inslagna talet, som lagts i en vektor, om till ett flyttal och pushas in i regulator-parameters parallellvärde.

Ömsesidig uteslutning har vi löst genom att använda oss av de båda förnämliga kommandona PUSH och POP. Med hjälp av dessa specialinstruktioner, som normalt inte ingår i BASIC, blir problemet betydligt förenklat. Vi använder oss av ett avbrott som heter ONTIME S,10000. Denna sats medför ett realtidsavbrott varje gång som tiden S har passerat. Återhopp sker m hj a kommandot RETI på rad nr10005 i programmet. Eftersom ett realtidsavbrott inte avbryter en påbörjad rad, kan man vara säker på att problem ej uppstår. På raderna 6000 och 6010 sker PUSH resp. POP av de ingående parametrarna.

Vad beträffar D/A-omvandlaren, så hade vi inga problem med 12-bitars bitar, eftersom datorn har en inbyggd flyttalsoperator och vår D/A krets var en 8-bitars krets.

För att lättare kunna överskåda vad som fortgår i en inkopplad process, tillverkade vi en frontpanel (se ritn.nr 7.6) med två analoga visare. En visare för utsignalens storlek, och en visare för insignalens aktuella värde. För insignalen har vi också kopplat in en liten simulator-ratt, som gör det möjligt för användaren att simulera en insignal mellan 0 - 10 volt då spaken längst ned till höger på panelen fälls uppåt. Det är viktigt att denna befinner sig i läge "av" dvs nedåt då man önskar koppla in och reglera en extern process (se ritn.nr 7.6).

Det är väldigt lätt att råka göra ett misstag när man virar ihop olika knytpunkter. Därför arbetade vi enligt dubbel-kontroll metoden, dvs en av oss virade och den andra testringde uppkopplingen efteråt. Genom att koppla en liten bit i taget, så minimerade vi riskerna för att ett fel skulle kunna uppstå.

Efter det att den grundläggande hårdvaran var klar, tog vi med oss hela utrustningen och satte oss i M-huset på LTH för att ha nära kontakt med våra handledare på institutionen för reglerteknik. Efter hand som vi började med att strukturera ett program-skelett, märkte vi att hårdvaran ej var fullständigt komplett. Detta medförde att viss hårdvara tillkom och implementerades allt eftersom nya önskade funktioner uppkom. Som exempel på detta, kan nämnas vårt problem med program-bortfall ur E2PROM vid påslagning av spänningsmatning och installation av blinkande larmfunktion.

Den tid som åtgått att lösa de olika deluppgifterna har varit tämligen jämnt fördelad, så att cirka halva tiden har gått åt till att ta fram den kompletta hårdvaran och den andra halvan har ägnats åt mjukvaran.

Själva programmeringen har gjorts i BASIC eftersom 8052AH har en inbyggd BASIC-interpretator. Det har visat sig att denna processor är väl lämpad till detta användningsområde, men programmet blir lätt svårläst på grund av BASICens ostrukturerade natur. Vi har därför gjort en separat program-beskrivnings-manual, (se bilaga 8.2.1) för att om möjligt förenkla förståelsen av programvaran för den oinvidige.

Arbetet med detta projekt har varit mycket lärorikt inte bara rent tekniskt sett. Vi har också lärt oss att ta kontakt med olika förtag, och att anpassa vår konstruktion till det material som finns tillgängligt. Detta är ett positivt exempel på samarbete mellan högskolan och näringslivet.

### 3. HARDVARUSPECIFIKATION

Matningsspänning till alla kretsar och komponenter på kortet kommer från ElektroSandbergs standardkort. För krets och komponentplacering se ritning 7.7. För komplett komponentlista se bilaga 8.5.

#### 3.1 PROCESSOR

Som processor används Intels 8052AH-Basic (IC1 :Ritn.nr7.1). 8052AH-Basic är en enkapseldator med inbyggd Basic-interpretator. Den kan hantera strängar, logiska operatörer och flyttalsaritmetik. Den kan också acceptera och avge flyttal, heltal och hexadecimala tal. Även EPROM-programmering, memokodsaccess till alla I/O-funktioner och en realtidsklocka finns implicit. Till processorn har vi kopplat sammanlagt 56K-bytes externt minne, 32K-bytes RAM (IC4-IC7 :Ritn.nr7.1) och 24k-bytes E2PR0M (IC12-IC14:Ritn.nr7.1).Styrsignaler för läsning och skrivning i de externa minnena är WR,RD,PSEN,ALE,PROGRAM PULSE/P1.4 och ALE DISABLE/P1.3. RESET-funktionen åstadkommes då brytare S1 slutes (se Ritn.nr7.1). Systemets klockfrekvens erhålles med hjälp av kristall X-TAL till 10,05200 MHz.

Ben 8 på 8052AH-Basic är en seriell utport till en skrivare. Via en buffert (IC10 :Ritn.nr7.1) kopplas ben 8 till AMP-kontakt 2 ben 2. Baudtalet kan sättas på processen m h a kommandot BAUD XXXX. Vi hade det låga baudtalet 300 eftersom ingen handskakning finnes. Via ElektroSanbergs standardbuffertar (IC10 och IC11 :Ritn.nr7.1) kopplas ben 10 och 11 till AMP-kontakt 3 ben 2 och 3 för kommunikation med bildskärm.

### 3.2 EXTERNA MINNEN

#### 3.2.1 RAM

Som RAM används fyra stycken HM 6264 LP-15. Dessa ingår i ElektroSandbergs standardsortiment. De uppfyller de kraven som CPU:n ställer. Kraven är minst 1K-bytes, börja med minnesplats 0 samt vara löpande och fullständigt kodade. Varje RAM har en minneskapacitet på 8K-bytes. RAM adresserna ligger från 0 till 32768. Alla RAM kopplas direkt på databussen ben 11,12,13,15,16,17,18 och 19. Ben 11 LSB och ben 19 MSB (se Ritn.nr7.1). Även addressbussen kopplas direkt till alla RAM. Via chip-select (IC3 Ritn.nr7.1) väljs rätt RAM med avseende på adress. IC2 håller adressens LSB till både RAM och E2PROM.

#### 3.2.2 E2PROM

Som E2PROM används tre stycken 52B33H. Både data och addressbuss kopplas direkt på alla E2PROM (se Ritn.nr7.1). E2PROM ligger på adresserna 08000H till 0DFFFH.

För att sätta ett E2PROM krävs en negativ puls på mellan 1 ms till 20 ms på ben 27 WE. Samtidigt måste signalerna till ben 1 CC och 20 CE varit låga och till ben 22 OE hög i minst 50 ns. Detta har vi löst med två monostabila vippor (IC24 ;Ritn.nr7.1) och med några fördröjningskretsar (IC25 ;Ritn.nr7.1). För att sätta i alla E2PROM krävs att processorn befinner sig i COMMAND MODE vilket innebär CPU:n väntar på kommando från operatören.

Detta i sin tur innebär att både ben 17 RD och 29 PSEN är höga och dessa signaler AND:as och ger en etta till ben 22 OE på alla E2PROM. Genom att S2 och S3 slutas kommer en negativ puls ut från vippa nr.1 på ben 13 Q1 (IC24 Ritn.nr7.1) som ligger hög i normala fall. Denna signal triggar i sin tur vippa Inr.2 via ben 9 A2 (IC24 ;Ritn.nr7.1). Detta gör att en positiv puls kommer ut på ben 5 Q2 som i normala fall ligger låg. Dessa båda pulstider dimensioneras m h a R26 och C14 för vippa nr.1 resp. R23 och C13 för vippa nr.2. Puls längden blir ungefär 15 ms för båda vipporna. För att utsignalen från ben 8 IC9, skall bli hög under vippa nr.2's puls krävs att S4 ej är sluten. S4 måste vara sluten under på och avslag för att förhindra oönskad suddning av alla E2PROM.

FOTNOT: Tidsfördröjning på WE signalen skall vara minst 50 ns vid suddning. Normalvärdet på våra fördröjningskretsar ger en fördröjning på 44 ns. Trots detta fungerar det oklanderligt. Vid tillverkning av fler kort bör detta åtgärdas.

### 3.3 I/O ENHETER

Alla I/O enheter styrs med RD ben 17 IC1 och WR ben 16 IC1 (se Ritn.nr7.2) samt med en adressavkodare IC22 (se Ritn.nr7.2). Då man önskar koppla fler enheter till bussen bör man buffra, annars kan problem uppsåt med systemets FAN-OUT.

#### 3.3.1 INPORT

Som inport har en SN74LS244N (IC19 se Ritn.nr7.2) valts. Kretsen ingår i ElektroSandbergs standardsortiment. Inporten ligger på adress EC00H-EFFFH. Eftersom SN74LS244N har 3-STATE utgångar kopplar vi den direkt på databussen ben 3,5,7,9,12,14,16 och 18. Ben 3 MSB ben 18 LSB. Vi använder bara 5 av de 8 digitala ingångarna. Dessa 5 användes vid avsökning av knappsats via plattkabelkontakt S6.

#### 3.3.2 UTPORTAR

Till de två digitala utportarna, IC15 och IC33, används en C-MOS latch UCN 5801A. IC15, ( se ritning nr7.2 ), ligger på adressen 0E800H-0EBFFH. Utgångarna 17,18,19 och 20 används till knappsatsfunktionen ( se ritn.nr.7.4 ). Utgångarna på ben 13 och 14 används till displayfunktionen.

UCN 5801A har "open-collector" utgång och för att visuellt kunna avgöra om avsökning av knappsatsen sker eller vilken vilken "display mode" som processorn befinner sig i, har vi använt lysdioder (LED1) OCH 220 OHMs motstånd (R15) som kollektor-last. Ben 15 och 16 används ej.

IC19 (se ritning nr7.3), ligger på adress 0F400H-0F7FFFH. Porten används till larmfunktionen.

### 3.3.3 D/A-OMVANDLAREN

För att erhålla en analog utsignal mellan 0-10 volt från processorn, använder vi oss av ElektroSandbergs standardkoppling för D/A -omvandling. Den består av en 6080-PC, IC17, som har en vanlig OP-förstärkare, 741, IC17, som utbuffert. Den analoga utsignalen finns tillgänglig på AMP1 ( se ritn.nr7.2) ben 2. D/A-omvandlaren ligger på adress 0E400H-0E7FFH.

### 3.3.4 A/D-OMVANDLARE

En 12-bitars A/D-omvandlare, AD-574H, IC18, används för att omvandla en analog insignal.AD-omvandlaren ligger på adress 0E00H-0E3FFH. Arbetsområdet för A/D-omvandlaren ligger, i vår koppling mellan 0 och 10 volt hos insignalen. Insignalen går in på ben 13 vid AMP1 ben 3.

Genom att läsa in de 12-bitarna i två omgångar, 8MSB ligger på jämn adress och de 4LSB på udda adress , och sedan mjukvarumässigt ta hand om bitarna, fås en upplösning på 10/(2\*\*12-1) som motsvarar 2.5mvolt per bit. Genom att skriva till AD-omvandlaren med kommandot XBY(0E000H)=02H sätts AD-omvandlarens interna konversions-cykel igång. Konversions-cykeln tar 35us. Detta har vi anpassat i programvaran.

Fördröjningen i AD574-A, från det att CE blir låg till det att datalinjen uppnår "3-state" tillstånd kan en buss konflikt uppstå. Detta kan uppkomma om kretsen kopplas direkt på data-bussen. För att förhindra detta har vi använt oss av en 74LS244N (IC21 ritn.nr7.2) som buffert till AD-OMVANDLAREN.

Trimpotentiometer R5 (ritn.nr7.2), används för att trimma bort offset-fel och trimpot. R4, justerar bit-upplösningen. Alla komponentvärden har valts med hjälp av datablad för AD 574H.

### 3.3.5 LCD

STANLEY's display GMD 24201 har en 2\*24 teckenmatris. Denna innehåller även ASCII-tecken, men även läsning från displayens intern-RAM är möjligt. Man når displayen genom att adressera den på adresserna 0F000H-0F3FFH. Displayen är kopplad direkt på databussen. Vilken mode som displayen skall vara i bestäms av Rs ben 4, och av R/W ben 5. Dessa signaler styrs från en digital utport IC15 ben 13 och ben 14. Dessa båda signaler styrs mjukvarumässigt. Enable E ben 6, styrs från en adressavkodare IC22. Med motståndet R29 regleras ljusstyrkan och avläsningsvinkeln på displayen.

FOTNOT. Enable-signalens pulstid skall vara minst 450ns. Vår enable-signal kommer från IC22 som i sin tur styrs av RD.XOR.WR från IC1. RD och WR signalerna är 400ns, vilket kan innebära fel vid läsningen. Eftersom det fungerade oklanderligt vid försök, valde vi att inte åtgärda detta. Vid tillverkning av fler kort bör detta beaktas.

## 3.4 ÖVRIGA ENHETER

### 3.4.1 KNAPPSATS

Knappsatsen (se Ritn.nr.7.4) är uppbyggd som en matris med 4 rader och 5 kolonner. Avsökning går till på följande sätt:

En "nolla" sänds ut på en rad, övriga rader ligger höga. Detta sker mjukvaromässigt från en utport IC15 ben 17,18,19 och 20. Efter detta avkännes raden ifrån en digital inport IC19 ben 2,4,6,8 och 11 även detta styrs mjukvaromässigt. Om någon tangent på den aktuella raden är nedtryckt kommer det att kunna maskas ut mjukvaromässigt från inporten. En kontinuerlig avsökning sker som ställer krav på processor ifråga om snabbhet. Här har det visat sig att vår processor 8052AH-Basic tyvärr är ganska långsam.

### 3.4.2 FRONTPANEL

På frontpanelen (se Ritn.nr.7.6) finns två voltmetrar, en för utsignal och en för insignal samt två kontakter för ut respektive insignal. Kontakten för insignalen kan kopplas bort med kontakten S7 och då inkopplas istället vridmotståndet R30 och ljusdioden LED2 in. Inspänningen kan då regleras från 0-10 V med R30. Frontpanelen spänningsmatas via AMP1 enligt följande (se Ritn.nr.7.6): Ben 1 +5 V via R9 220 ohm, Ben 2 utsignal 0-10 V, Ben 3 insignal 0-10 V, Ben 6 +12 V via R8 560 ohm och Ben 8 jord.

### 3.4.3 LARM

Vår larmenhet (se Ritn.nr.7.3) styrs med två digitala signaler ifrån en utport, ben 19 och 20 på IC33. Funktionen för larm är följande:

Vid överstiget RFEL, skall larm ges med en blinkande diod. Larmet kvitteras genom att operatören trycker in knappen "C" antingen på knappatsen eller på terminalen beroende på vilken enhet som används. Om det aktuella reglerfelet fortfarande är större än RFEL markeras detta genom ett fast sken från lysdioden. Problemet har vi löst med ett asynkront sekvensnät som styr en astabil vippra. Anledningen till varför vi valde ett asynkront sekvensnät i stället för att lösa det helt mjukvarumässigt, var för att om möjligt snabba upp processen.

För graf och tillståndskodning se bilaga B.6. I grafen motsvaras RFEL av i0 och kvittenssignalen motsvaras av i1. Grafen avkodas hazardfritt och utsignalen är ekvivalent med tillstånden.

Den astabila vippan består av två monostabila vippor som triggar varandra IC32. Pulstiderna samt släcktiderna kan varieras med motstånden R28 och R29. Pulsfrekvensen är cirka 0.25 sekunder. Parallellt med dioden har en piezo-elektrisk kristall kopplats. Kristallen är placerad inne i knappatsboxen. Ljudsignalen larmar endast då ett nytt larm uppstår dvs endast när dioden blinkar.



## 4. MJUKVARUSPECIFIKATION.

Mjukvaran är uppbyggd kring ett realtidsavbrott med följande konfiguration:

```

YYY   TIME=0:  CLOCK1:  ONTIME S,XXX
SSS   I
      I
      I   FÖRGRUNDSPROCESS: Operatörskontroll
      I
      I
      I
      I
      I
      I   GOTO SSS

XXX   REGLER ALGORITM
      I
      I
      I   BAKGRUNDSPROCESS: Reglering
      I
      I
      I
      I
      I   ONTIME S+TIME,XXX
ZZZ+1 RETI

```

Dessa instruktioner fungerar enligt följande:

Varje gång som tiden S passerat, genererar den inbyggda realtidsklockan ett hopp till raden XXX. Programraderna fram till ONTIME S+TIME,XXX, exekveras och därefter sker ett återhopp till den rad där interruptet aybröt programmet i övrigt. Detta sker med hjälp av instruktionen RETI ( Return from interrupt ).

#### 4.1 REGULATORALGORITMEN

Vår regleralgoritm bygger på följande samband:

$$y(t) = K * (e(t) + TD * de(t) / dt + 1 / TI \int_0^t e(s) ds)$$

där derivatatermen approximeras med:

$$TD * de / dt \approx p * TD / (1 + p * TD / N)$$

Här betecknar  $e$  reglerfelet,  $TD$  derivatattiden, och  $N$  brytffrekvensen i bodediagram.

med hjälp av denna approximation fås överföringsfunktionen:

$$G(s) = K * (1 + 1 / (s * TI) + s * TD / (1 + s * TD / N))$$

För att kunna identifiera de olika termerna i vår program, kan man föra följande resonemang.

Vid varje samplingstillfälle sker en analog/digital omvandling av vår insignal  $XIN$  som räknas om från 12-bitar till ett flyttal (se rad nr. 7730-7733). Därefter bildar vi reglerfelet  $E$  genom att ta  $E = WIN - XIN$  dvs börvärdet-ärvärdet. Detta sker på rad nr. 7735 i vårt program. Därefter beräknas utsignalen  $Y$ .  $Y$  som i vårt fall är ett flyttal, omvandlas till en analog utsignal. Digital/analog omvandlaren håller kvar sitt gamla värde tills dess att en ny omvandling sker.

Då tiden mellan två på varandra följande sampel är tillräckligt kort så erhålles en reglerverkan som är nästan identisk med en analog P-regulator då man endast betraktar produkten  $K * e$ . Denna term finns representerad i vår algoritm på rad nr. 7745.

För att åstadkomma integralverkan så approximeras integralen

$$I(t) = K / TI * \int_0^t e(s) ds$$

med summan

$$I(t) \approx K * h / TI * \sum_{k=1}^{\infty} e(t - k * h).$$

Summan kan beskrivas med rekursionsformeln:

$$I(t) = I(t - h) + e(t - h) * K * h / TI$$

Denna formel finns implementerad i vår regleralgoritm, nämligen faktorn  $A1 = K * h / TI$  på rad nr. 7712.  $K$  motsvaras av  $K1$  och samplingstiden  $h$  motsvaras av  $S$ .

På rad 7765 sker således den slutgiltiga beräkningen av integraldelen. Här finns ytterligare en facilitet inlaggd nämligen Anti integratoruppvridning.

Funktion:

E=WIN-XIN

V=K1\*E+I

Då V överstiger det maximalt tillåtna värdet så skall integraldelen sättas till IN. Detta ger oss:

IN=I+Ymax-V

Samma sak gäller då V underskrider den lägsta tillåtna utsignalen:

IN=Ymin-K1\*E=I+Ymin-V

Detta ger oss koden på rad 7765:

H=H+Y-V+A1\*E

Där H=Integraldelen

V=Den nominella styrsignalen.

För att PID-regulatorn skall bli komplett måste även en derivatadel realiseras. En derivationsoperator kan ej realiseras exakt. Därför sker en approximation enl. föregående sida.

Förenkling sker på följande vis:

$Vd=e^{*(p*TD+N-N)/(1+p*TD*N)}=e^{*(N-N/(1+p*TD/N))}=$

$=N*(1-(TD/N)/(p+TD/N))*e=N*(e-Z)$

Där  $Z=e^{*(TD/N)/(p+TD/N)}$

variabeln z uppfyller differentialekvationen:

$dz/dt+(N/TD)*z=(N/TD)*e$

Då derivatan approximeras med en differens finner man

$(z(t)-z(t-h))/h+N*z(t)/TD=N*e(t)/TD$  dvs.

$z(t)=z(t-h)+(N*h/Td)*(e(t)-z(t))$

Se rad nr. 7740.

Detta ger oss term nummer 3 på programrad 7745:

A4\*(E-Z0).

På raderna 7758-7760 sker en begränsning av utsignalens max och minvärde.

Subroutinen 7950 undersöker om larret skall utlösas samt om kvittenssignal har skickats.

#### 4.2 REALTIDSPROBLEMATIK

Mjukvaran består av en förgrundsprocess och en bakgrundsprocess. Dessa löper parallellt. Förgrundsprocessen består av operatörskontroll via terminal eller knappats beroende på vilket man väljer, se bilaga 8.2. Övergång från knappats till terminal eller omvänt kan ske under körning. I förgrundsprocessen tilldelas regulatorn nya parametervärden, larm kvitteras och regulatorns ingående parametrar visas på bildskärmen respektive LCD. Bakgrundsprocessen består av själva regleralgoritmen som finns beskriven i kapitel 4.1.

Då man skall kunna arbeta i realtid, vilket innebär att ingående parametrar i reglerprocessen skall kunna ändras under själva regleringen, måste ömsesidig uteslutning finnas. Vi har löst detta på följande sätt.

Alla ingående parametervärden finns representerade i två parallella variabler. Det är bara den ena variabeln som ingår i regulatoralgoritmen. Som exempel kan nämnas börvärdet WIN som finns i variabeln W och W2. W ingår i regulatoralgoritmen medan W2 ingår i förgrundsprocessen. Variabler för SCALEMIN, SCALMAX och UTYP har vi inte räknat in bland dessa eftersom de endast beror på givaren i aktuell process. De initialiseras bara och kan ej ändras under reglering. För att ändra dessa tre parametrar krävs en ny uppstart av programmet. Alla andra parametrar går att ändra under programmets gång. Då ett nytt parametervärde skall införas till regulatorn sker följande. Den ena variabeln erhåller det nya parametervärdet. Detta rimlighetstestas vad gäller gränser, till exempel börvärdet W2 måste ligga inom gränserna SCALEMIN och SCALMAX. Alla nya parametrar, det vill säga den ena av de två parallella variablerna, lägges på en stack med hjälp av kommandot PUSH. Detta sker på rad 6000 i programmet. Genom att läsa och tilldela från stacken med kommandot POP tilldelas regulatorvariabeln dess nya parametervärde. Detta sker på rad 6002 i programmet.

Avbrott till bakgrundsprocessen från förgrundsprocessen sker endast mellan två rader i programmet. Detta innebär att POP-instruktionen är odelbar och ömsesidig uteslutning erhålles. Enligt samma princip tilldelas de variabler som skall visas på bildskärmen respektive LCD.

## 5. UTVÄRDERING

### 5.1.1 KOMMENTAR

Som slutsats har vi kunnat konstatera att EN-CHIPS-DATORN B052AH-BASIC är ändamålsenlig för tillämpningsområden där man kräver en enkel lösning av problemen kring realtidsavbrott, men samtidigt inte ställer för höga krav på hastigheten. Enligt specifikationen för examensarbetet skall samplingsintervallet ligga mellan 1-255 sekunder. Regulatorn är alltså i huvudsak avsedd för att reglera relativt långsamma processer till exempel temperaturreglering i byggnader. För samplingsintervall kortare än 1 sekund uppför sig kretsen otillfredsställande. Även detta går att lösa eftersom delar av programmet kan skrivas i assembler i form av subroutiner som man sedan kan låta BASICprogrammet kalla på med hjälp av en CALL-instruktion.

De extra instruktioner som finns implementerade underlättar realtidstillämpningen avsevärt. Exempel på dessa instruktioner är: PUSH och POP, DO-UNTIL och DO-WHILE. En nackdel med BASIC är dess ostrukturerade natur vilket medför att programmet blir svårsläst för den oinvidde.

Det vi saknade hos processorn var möjligheter till ytterligare avbrottsinstruktioner. Processorn har som den ser ut idag endast 2 olika möjligheter till avbrottsstillämningar. Dessa två instruktioner är ONTIME, samt ONEXT. ONTIME styrs helt mjukvarumässigt medan ONEXT styrs hårdvarumässigt via ben 13 på B052 AH-BASIC. Vid avbrott så har ONTIME högre prioritet än vad ONEXT har. För att snabba upp knappavsökningen, borde avsökningen ske hårdvarumässigt. Detta skulle kunna lösas genom att använda sig av ONEXT och en inport.

Det som varit det stora problemet för oss var att snabba upp knappsavsökningen. Eftersom BASIC är ett förhållandvis långsamt programspråk finns det risk för att en siffra som slås in på knappsatsen missas då programmet befinner sig inne i reglerprocessen. En förbättring av detta problem skulle kunna ske genom att man skriver regleralgoritmen i assemblerkod i stället för BASIC eftersom assemblerkod exekveras betydligt snabbare.

Utskriften på LCD skulle kunna förbättras genom att låta den visa båda parallellparametrarna samtidigt. Genom att göra detta så slipper operatören att anteckna det nya parametervärdet.

En seriekanal borde anslutas så att systemet har faciliteten att kunna sammankopplas med ElektroSandbergs eget system, ES2000.

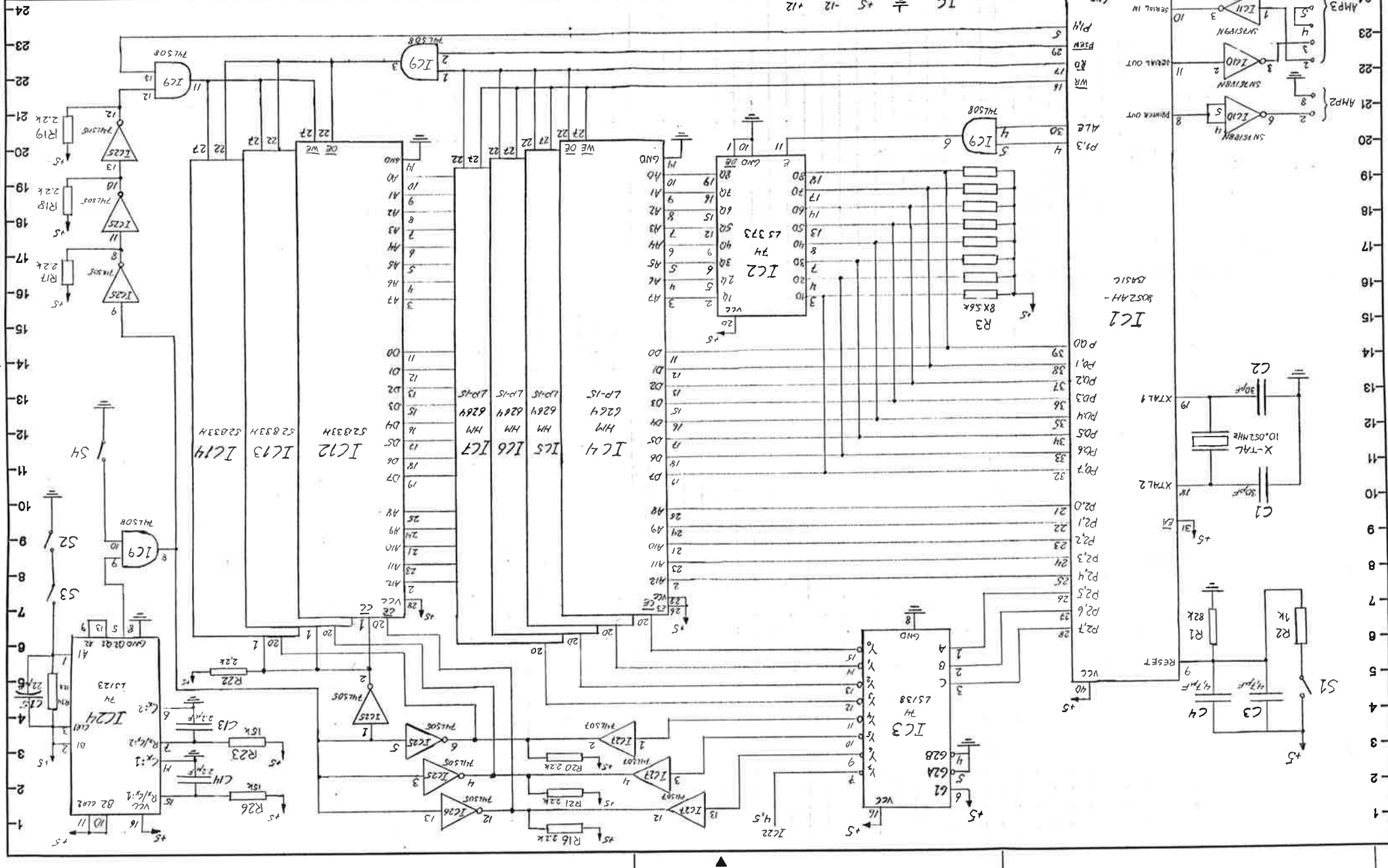
### 5.1.2 PLANERING

Då det gäller uppläggnigen av vårt examensarbete så försökte vi att planera en kontinuerlig dokumentation. Det visade sig emellertid ganska snart att vi trots förberedelser blev ganska stressade för att hålla tidsschemat.

## 6. REFERENSLITTERATUR

1. K-J ASTRÖM: REGLERTEKNIK. CODEN LUTFD2/(TFRT-3166)/1-050/(1982)
2. MICROCONTROLLER HANDBOOK: INTEL CORPORATION LITERATURE DEPARTMENT 1984.
3. ELECTRONIC DESIGN: DEC. 13 1984. VOL 32 nr.25. SCHIPHOL: HAYDEN PUBLISHING COMP.
4. MODERN ELEKTRONIK: NR.2 1985.SUNDBYBERG: NORDPRESS AB.
5. SOLUTIONS: INTEL CORPORATION LITERATURE DEPARTMENT, NOV/DEC 1984.
6. TTL HANDBOOK, TEXAS INSTRUMENTS 1981.
7. ROLF JOHANNESSON: DIGITALTEKNIK. LUND:STUDENT LITTERATUR 1982
8. JOHN KATAUSKY: MC2 BASIC-52 USER MANUAL. INTEL CORP. LITERATURE DEPARTMENT 1984.
9. DATABLAD 52B33H-250: SEEO TECHNOLOGY INC. MARS 1983.
10. DATABLAD UCN-5801A:SPRAGUE ELECTRIC COMP. MASS: NORHT ADAMS 1984.
11. DATABLAD AD574A JD:ANALOG DEVICES.NORWOOD,MASS:ANALOG DEVICES 1984.
12. DATABLAD GMD24201: STANLEY ELECTRIC CO.LTD.NAKAMEGURO:STANLEY ELECTRIC CO.LTD 1984.

NO	Revision	Andring
101		
Date	Datum	
	App	



IC	+5	-12	+12
9	7	14	7
10	7	14	7
11	7	14	7
25	7	14	7
26	7	14	7
27	7	14	7

Bl. 101

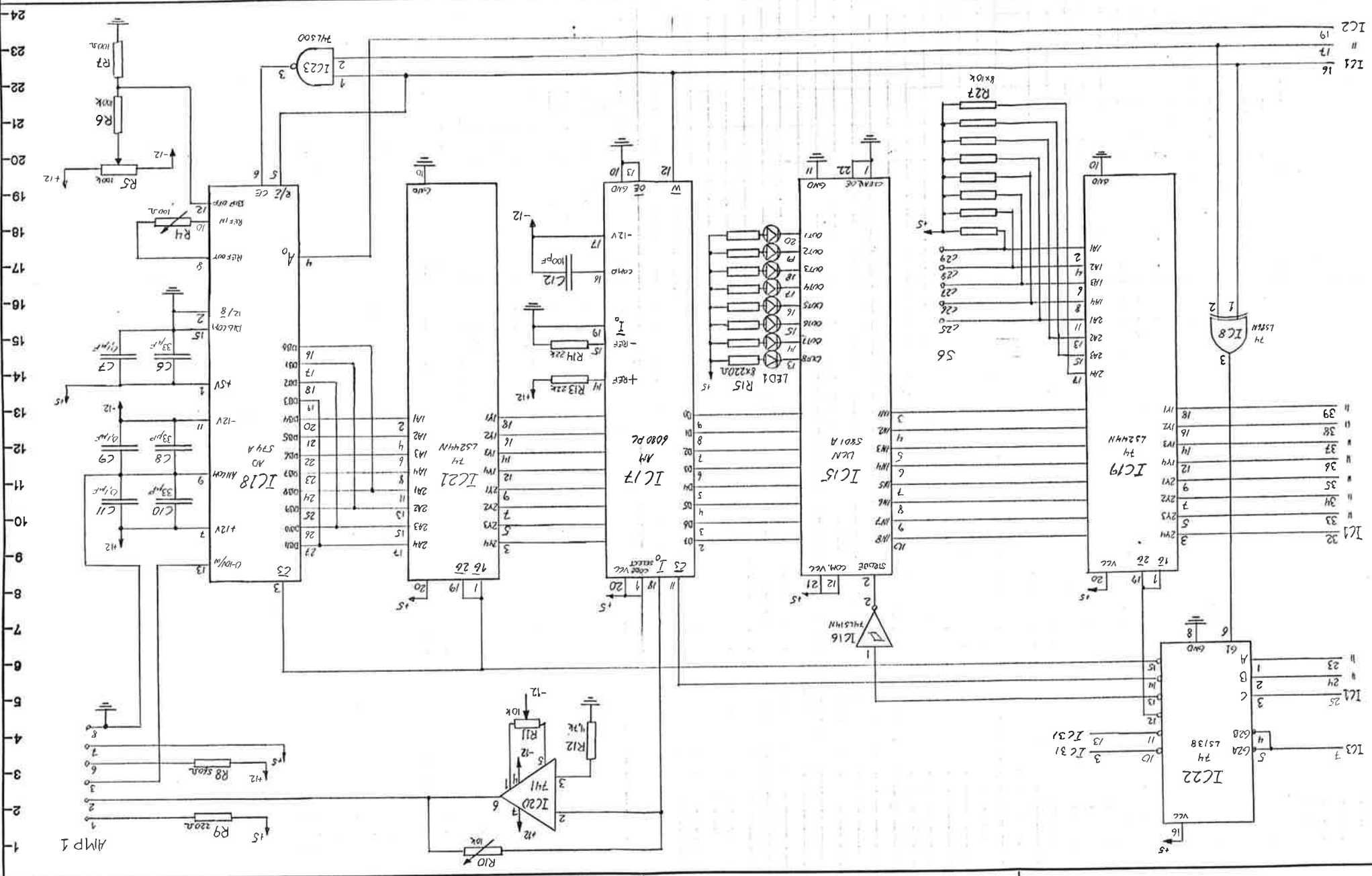
No	Revision	Datum	App

IC  $\neq$  +5  
 8 7 14  
 16 7 14  
 23 7 14

AUTONOM REGLER CENTRAL

Elektro Sandberg ab

Riss	Datum	Rev.	Scale	Scale	Scale	Blad	Sheet
SV	10.06.85	7.2					



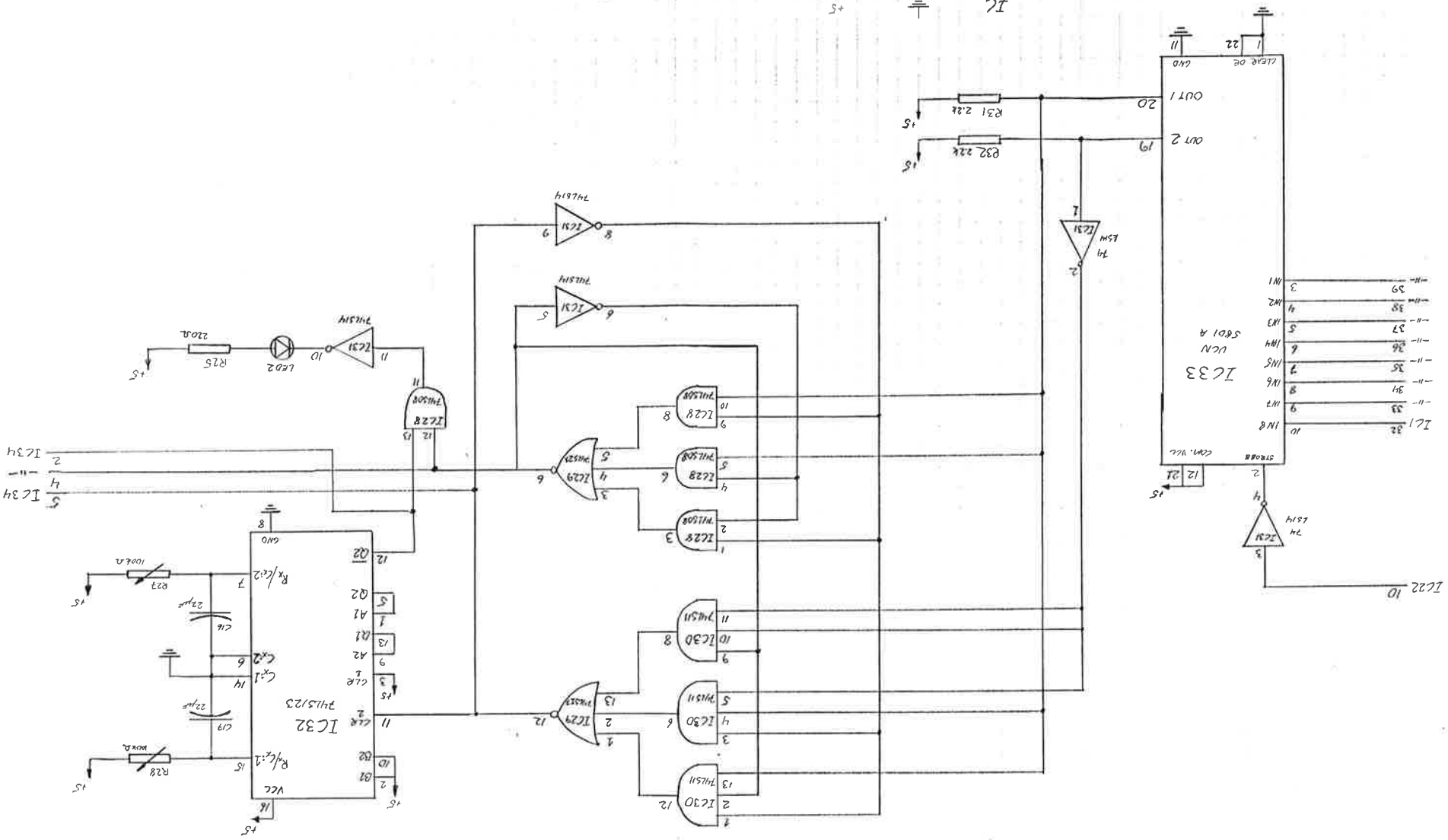


No	Rev	Date	Appr.

Elektro Sandberg ab		LARMENHET	
Autonom Reglercentral		LARMENHET	
Ritad Drawn	Scale	Date	Rev
SA	100685		7.3
Kontroll Design		Fortsbl. Cont. Blad	
Sheet		Sheet	

IC

28	7	14
29	7	14
30	7	14
31	7	14

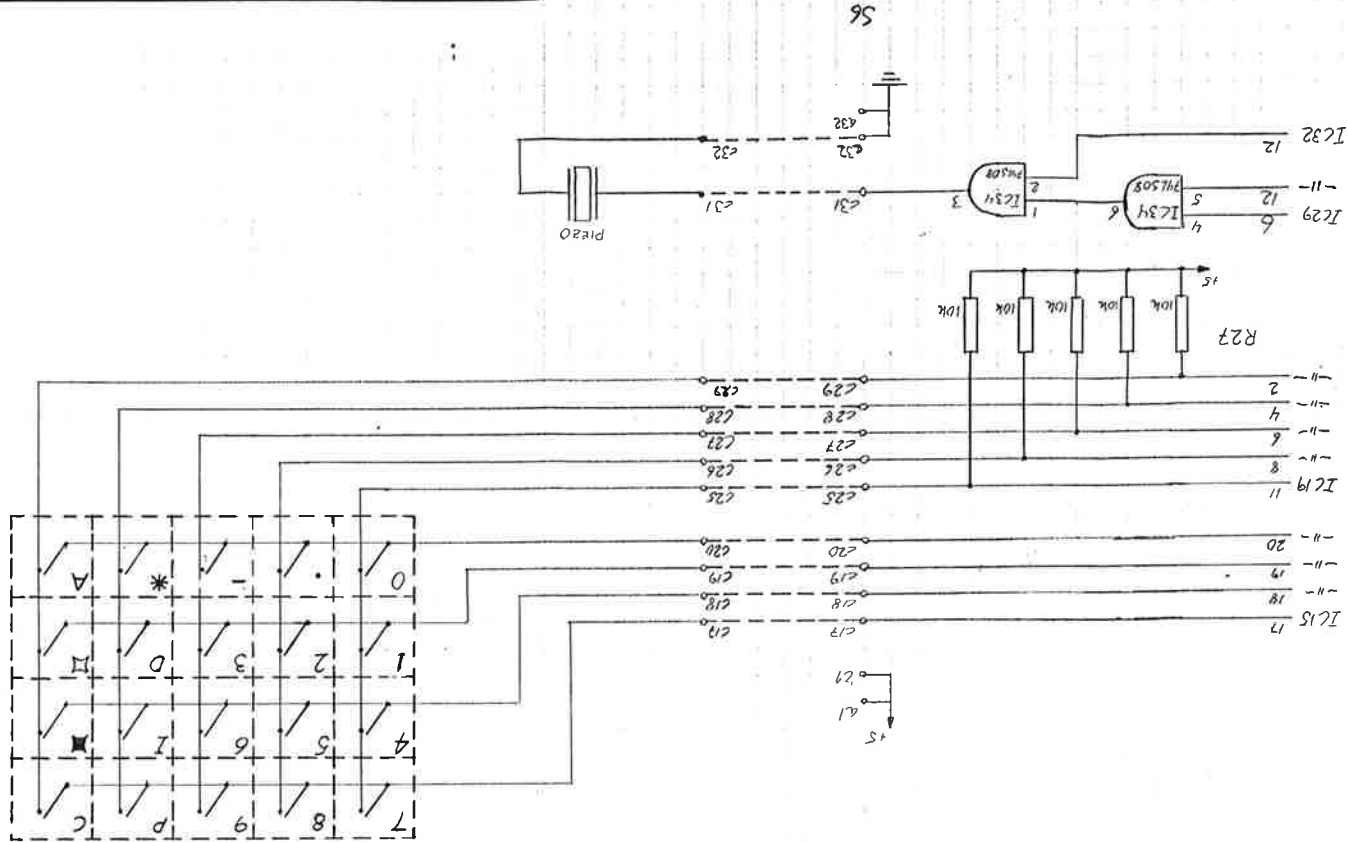


24  
23  
22  
21  
20  
19  
18  
17  
16  
15  
14  
13  
12  
11  
10  
9  
8  
7  
6  
5  
4  
3  
2  
1

B1 101

Noi	Andring	Date	Appd.

KONSTRUKTIVS OCH LAGERINGENHET AUTONOM REGLERCENTRAL		Riktad Design Skala SN 100685	
Riktad Drawn Datum Date Rikn.nr. 7.4		Fortsät. Cont. Bind Sheet	
Drawing.no.		Elektro Sandberg ab	



24  
23  
22  
21  
20  
19  
18  
17  
16  
15  
14  
13  
12  
11  
10  
9  
8  
7  
6  
5  
4  
3  
2  
1

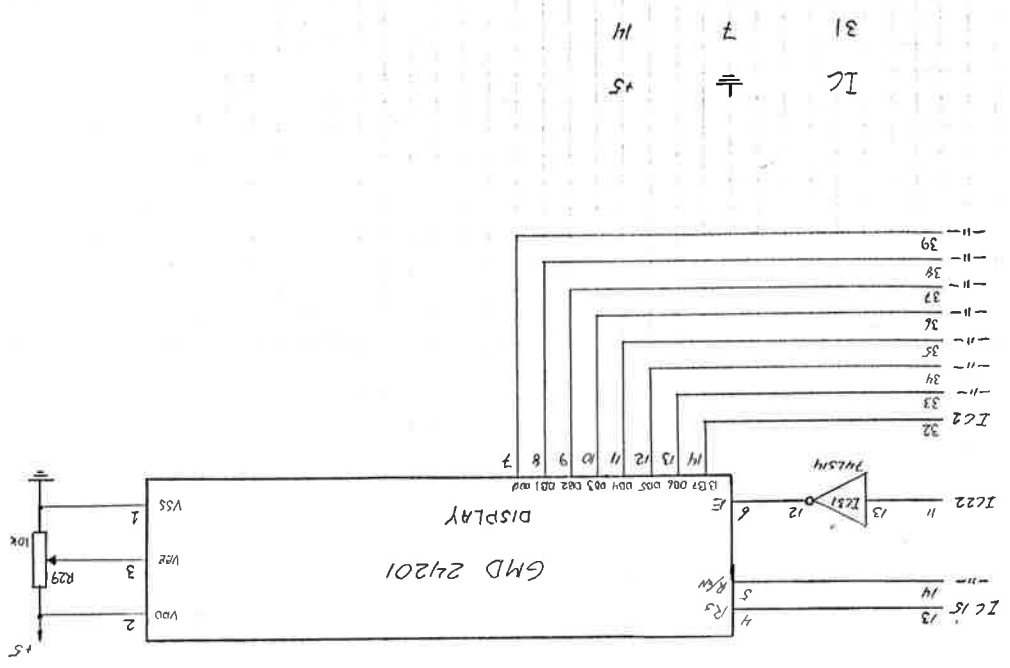
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

Bl. 101

Andring Revision	Datum	Av Appd.

Konst. Design	Skala	Scale	Fortsp. Bl. Cont.	Blad Sheet
SV	100882	75		
Ritad Drawn	Datum Date	Ritn. nr. Drawing no.		

AUTONOM REGLEKCENTRAL  
LCD  
Elektro Sandberg ab



1-24

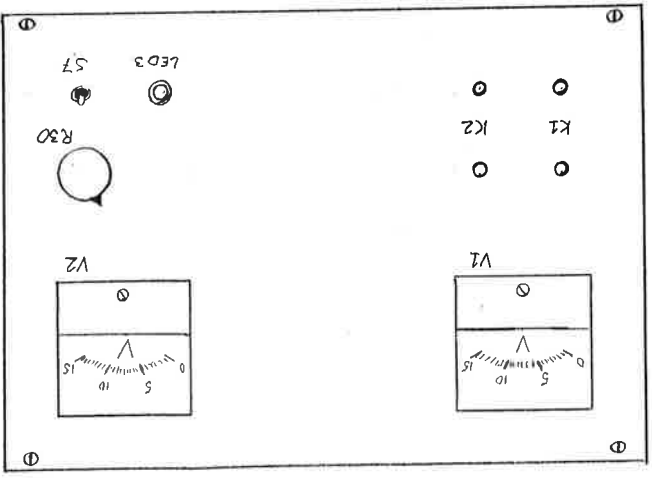
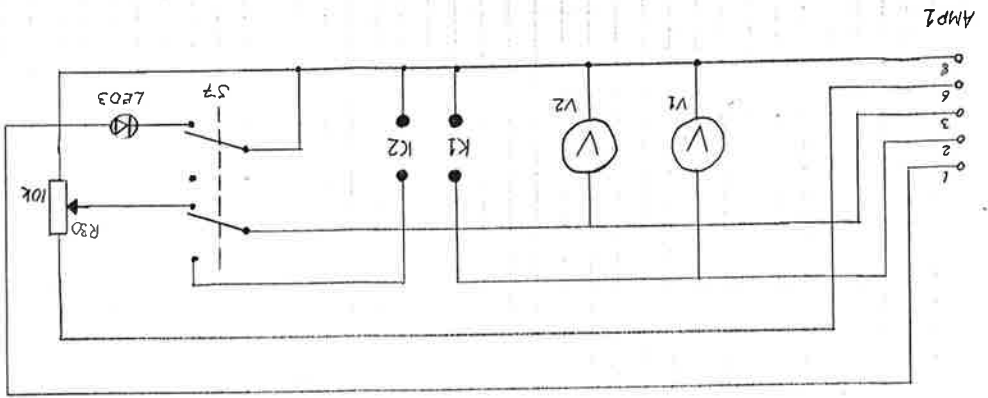
1-24

Av	Datum	Andring	Revisjon	Noi

Sheet	Contl. Bld	Scale	Design	Ritn.no.
		100685	SV	76
Drawing.no.	Date	Date	Date	Date

Elektro Sandberg ab

FRONTPANEL  
AUTONOM REGELCENTRAL



24  
23  
22  
21  
20  
19  
18  
17  
16  
15  
14  
13  
12  
11  
10  
9  
8  
7  
6  
5  
4  
3  
2  
1

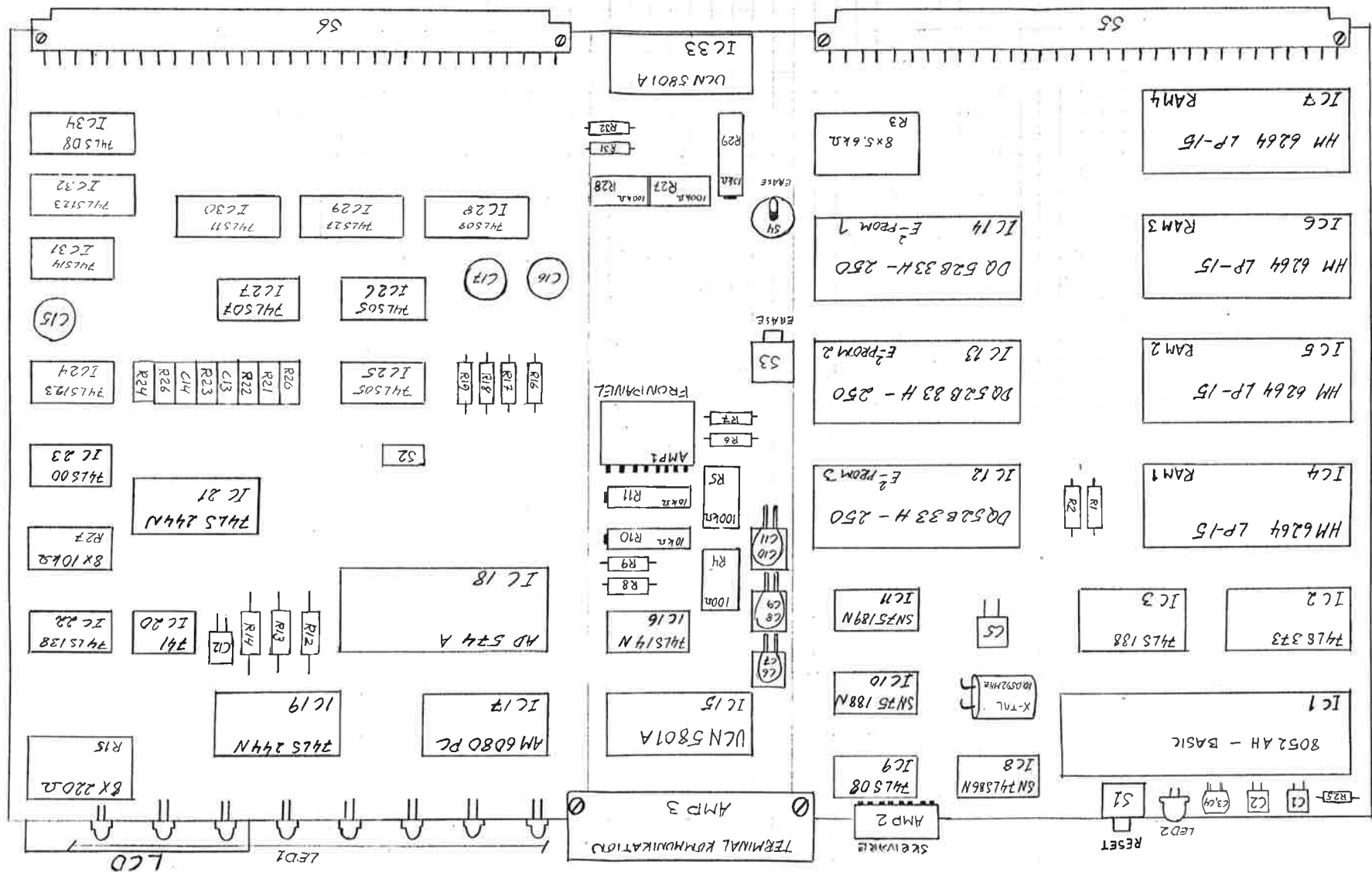
25  
24  
23  
22  
21  
20  
19  
18  
17  
16  
15  
14  
13  
12  
11  
10  
9  
8  
7  
6  
5  
4  
3  
2  
1

No. 101	Revision	Date	Appd.

KRETS OCH KOMPONENTPLACERING		Konstr. Design	Skala Scale	Forstbl. Cont. Bld	Sheet
AUTONOM REGLERCENTRAL		Ritad Drawn	Datum Date	Ritn.nr.	Drawing no.
		SV	1/0685	7.7	

24  
23  
22  
21  
20  
19  
18  
17  
16  
15  
14  
13  
12  
11  
10  
9  
8  
7  
6  
5  
4  
3  
2  
1

25  
24  
23  
22  
21  
20  
19  
18  
17  
16  
15  
14  
13  
12  
11  
10  
9  
8  
7  
6  
5  
4  
3  
2  
1



TERMINAL KÖMPLIKATION      SKEMARE      RESET      LED2      LCD      LED1

850215/exjob1.txt/m.jervill

## EXAMENSARBETE: AUTONOM REGLERCENTRAL

BESKRIVNING: Konstruktion av en fristående reglercentral med Intels 8052AH-Basic microcomputer.

## ETAPP 1: Reglercentral.

Konstruera med nedanstående beskrivna hårdvara en reglercentral och implementera en PID algoritim där operatörskommunikationen sker via en terminal. Konstruktionen bör vara flexibel ifråga om RAM och EEPROM, då systemet s.a.s kommer att användas som utvecklingssystem.

All operatörskommunikation med reglercentralen skall ske i realtid, dvs alla parametrar skall kunna ändras eller visas samtidigt som reglering pågår.

Regulatorns parametrar kan vara:

S    samplingstid    1-255 sek  
 TYP   P/PI/PD/I/ID/D/PID    dir/rev  
 WIN   börvärde flyttal  
 XIN   ärvärde flyttal  
 RFEL: största tillåtna reglerfel  
       Iarm om abs(WIN-XIN)  $\leq$  RFEL  
 KP    regulatorns P-verkan  
 TD    deriveringstiden  
 TI    integreringstiden  
 YMIN  minsta signal regulatorn får styra ut 0-100%  
 YMAX  största signal regulatorn får styra ut 0-100%  
 DYMAX största tillåtna förändring av utsignalen 0-100%  
 UTYP  ställdon typ öka/minska alt. posit.  
 Y     regulatorns utsignal presenteras 0-100%  
 GT    ställdonets gångtid i sek  
 \*\*    \*\*\*\*\*  
 \*\*    \*\*\*\*\*

Andra intressanta faciliteter kan vara framförkoppling, utekompensering, nattsänkning (dvs. två börvärden som väljes med hjälp av en digital ingång).

Flyttal programmeras och presenteras med en decimal.

Integer presenteras utan inledande nollor.

Larm indikeras med en blinkande lysdiod.

Kvittering av ett larm sker med en "etta" på en ingång.

Hårdvara: 8052AH-BASIC

CMOS RAM ( 8k x 8 ) x st  
EEPROM ( 2k x 8 / +5 V ) x st  
A/D -omvandlare 10 alt 12 bits  
för analog ingång 0-20 mA  
D/A -omvandlare 8 bits  
för analog utgång 0 -10 V  
max 8 digitala utgångar  
max 8 digitala ingångar  
seriekanal RS232 för terminal  
seriekanal RS232 för printar

ETAPP 2:

Operatörskommunikation.

Komplettera ovanstående hårdvara med en LCD-display och en knappsats, samt skriv mjukvara för hantering av den.

I övrigt operatörskommunikation som ovan.

ETAPP 3:

"Adaptivitet."

Implementera en enkel adaptiv algoritm.

Alternativt ett koeficient samband typ Ziegler-Nicolls diskreta motsvarighet.

Prova och beskriv.

ALLMANT:

Redovisning enligt standard exjobb-modell med en beskrivande del samt schema.

FÖRETAG:

ElektroSandberg  
Jägershilligatan 15  
213 75 MALMÖ

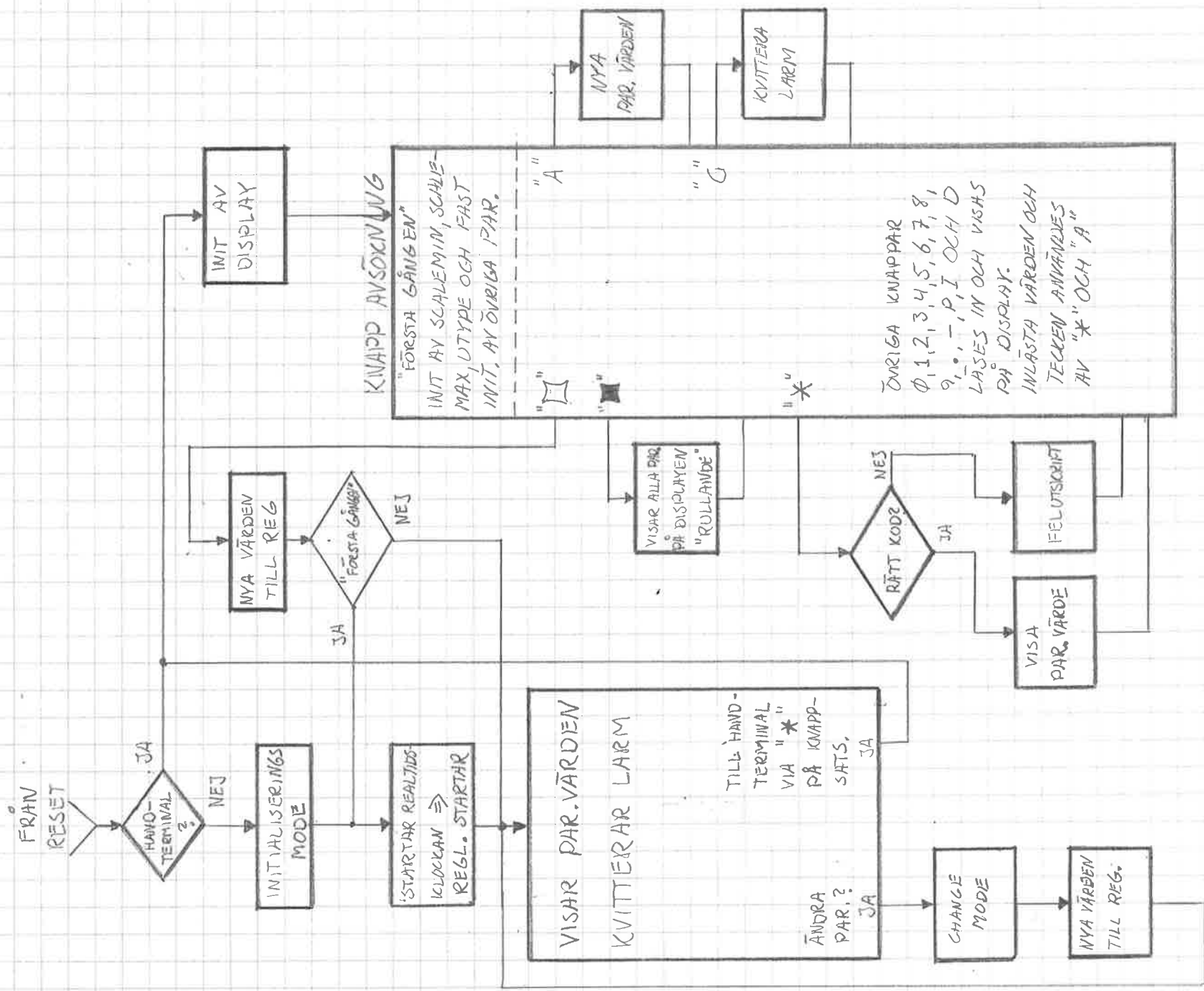
tel: 040-210 110

HANDLEDARE:

Martin Jervill

# Bilaga 8.2

## FLÖDESSKEMA ÖVER PROGRAMVARA





VERBAL BESKRIVNING AV MJKUVRANS FUNKTION

BILAGA 8.2.1

RADNUMMER	
100-120	Initiering av vektorer och teckensträng.
900-910	Init. av larmport.
920	Nollställning av utsignalen.
1000-1020	Init. av LCD.
1025-1075	Utskrift av AUTONOM REGLERCENTRAL PRESS (*) sker på LCD.
1100-1115	Init av begränsningsvärdet.
1200-1270	Slinga som känner av om operatören vill använda knapp-satsen eller terminalen. Programmet snurrar här tills man valt kommunikationssätt.
1300-1350	Hit hoppar programmet då man vill använda knapp-satsen. Här sker init. av begynnelse värden för att processen ej skall kunna startas utan dito. Utskrift av SCALEMIN=? sker Därefter frågas det efter SCALEMAX=?. Sist sker utskrift av DIR=D REV=A ?
1350-1365	Subroutine som skriver ut data-strängarna ovan
1610-1624	0 tilldelas ett tal mellan 0 och 7 beroende på vilken regulator man väljer.
1638-1799	Init. och begränsningar av parametrar. Inslagna värden testas så att de ligger inom tillåtna värden.
1800-1804	Skalning av par.
1810-1811	Tilldelar parallell par. sina värden. Dessa värden skall displayas.
2800	HÄR STARTAS REALTIDSKLOCKAN.
2810	C6=1 då realt. klockan är startad.
2900-3027	Utskrift av fasta värden på terminalen.

3025-3026 Här tilldelas varje parameter sitt parallellvärde. Detta för att en process skall kunna löpa i realtid då ändring av par sker.

3030-3056 De värden som skall uppdateras under drift skrivs ut här. Knappsatsen avläses om \* är nedtryckt.

3058-3070 Utskrift då man hoppat till CHANGE MODE.

3090-4970 Rutin för ändring av aktuella parametrar. De nya värden som man matar in läggs i respektive parameters parallell parameter.

5000-5118 Denna subroutine läser in en siffra i talet och gör om det till flyttal.

5200-5234 Subroutiner för tilldelning av ny regulatortyp.

6000-6014 Här sker mutual exclusion, dvs par.-värdena som man vill ändra till pop'as in i regul.

7000-7666 Här ligger de olika regulatorerna som man hoppar till vid interrupt.

7700-7995 Här ligger huvud-regul.

7751-7760 Kontroll för DYMAX.

7765 Anti-reset-windup.

7950-7990 Kontroll av larm.

8000-8345 Avsökning av knappsats.

8500-8550 Utskrift av inslaget värde på LCD. Kontroll om talet är rimligt och lagring av inslaget tal i vektorn E(I7).

8555-8560 ANTI-KONTAKTSTUDS-LOOP.

8561-8592 Subroutine för utskrift av löpande Y-värde på LCD.

8600-8695 Rutin för utskrift av löpande värden på LCD.

8700-8710 Mellan hopp då användaren har slagit in "\*". Z7=1 då init av MIN o MAXSCALE pågår.

8720-8797 Subroutine för utskrift av XIN o WIN , löpande på LCD.

8900-8908 Mellan-hopp då "A" slagits in.

8910-8915 Hopp då man ändrat färdigt med knapp satsen. LCD'n suddas och de nya värdena pop'as in.

8920-8925 Mellan-hopp då man vill ha löpande utskrift av värden på LCD.

8930-8955 Skriver ut ALARM CLEARED samt nollställer larmet. Parameterflaggorna nollställs.

8995-9030 Hit kommer man om man slagit in en kod och därefter "\*". C eller C0 tilldelas ett värde beroende på vilken kod som man slagit in. Programmet hoppar till en subroutine som mot-svarar rätt parameter.

9050-9758 Här ges utskrift av rätt parametervärde på LCD då rätt kod slagits in. Aktuell par. läggs i A8. Därefter görs talet om till en vektor som sedan skrivs ut m hj a subroutineen på rad 9720

9760-9800 C0 och C ger hopp till respektive parameter då operatören vill ändra en par. Först sker utskrift av par.? Därefter förväntas operatören slå in ett nytt värde och ännu en gång trycka på "A".

9801-9807 Här befinner sig programmet vid init. av SCALE MIN och SCALEMAX. Om de ej ligger inom till-åtna gränser, sker återhopp till rad 1000.

9808-9816 Skallning av de par. som skall visas på LCD.

9818 Hopp till utskrift av DIR=D REV=A ?

9822-9950 Här pushas det nya par.värdet NS, som sedan popas i regulatorns parallell parameter. Här sker också en enkel rimlighetstest av värdena.

9950-9970 Utskrift av TYPE=?.

9975-9987 Utskrift av ILLEGAL VALUE och återhopp till knapp avsökning.

10000 Start av conversions-cykeln till A/D-omv.  
För att värdet skall kunna läsas in måste det  
vara en fördröjning på minst 35us. Vår för-  
dröjning är mellan 90-100us. (se rad 7730-33).

10002-10005 Hopp till aktuell regulator typ. Realtids-  
interrupts-återhopp.

10010-10022 Utskrift av WRONG COMMAND på LCD. Nollställn.  
av flaggor samt återhopp till knappavsökning.

10022-10194 Ger utskrift av parameternamn samt frågotecken  
då man trycker på "A". tex. WIN=?

10200-10208 Subroutine som suddar den undre raden på LCD'n

10220-10222 Initierar LCD'n för utskrift på den undre  
raden.

11000-11124 Subroutine som gör om det inslagna värdet som  
ligger lagrat i vektorn E(I), till ett flyttal  
som läggs i N5. N5 pushas sedan in som den nya  
parametern i regulatorn. Antalet siffror i  
vektorn E(I) finns i variabeln I7.

11130-11146 Hopp till olika tilldelningar av Q1 beroende  
på vilken regulator man vill ändra till.

11150-11174 Init. av DIR=D REV=A ? via knappsats. Om man  
slår fel, så börjar init. om från början.

11200-11330 Tilldelning av Q1 beroende på vilken regulator  
man ändrat till. (Q1 är parallell par. till Q)

11400-11462 Här sker omvandling av flyttal till vektorn  
A(I). Varje ingående siffra placeras som ASCII  
kod i vektorn för att talet skall kunna  
skrivas ut på LCD. Antalet siffror lagras i  
variabeln A9. Talet läses in i A8.

## FÖRTECKNING ÖVER ANVÄNDA VARIABLER

I = I9	=	Är reserverade till räknare.
J = J9	=	" "
K1	=	KP (regulatorns P-verkan, förstärkningen)
K2	=	KPmax
K3	=	KPmin
M1	=	YMIN
M2	=	YMAX
M3	=	DYMAX
Q	=	Används som identifiering i GOSUB (TYPE)
R	=	RFEL (MAX TILLÅTET REGLERFEL).
S	=	SAMPLINGSTIDEN
T1	=	TI ( Integrationstiden ).
T2	=	TD ( Derivata tiden ).
T4	=	TIMAX
T5	=	TIMIN
T6	=	TDMAX
T7	=	TDMIN
T3	=	Flagga som är DIR=> T3=1 REV=> T3=2
W	=	WIN ( Börvärdet )
W1	=	WINMAX=SCALEMAX
W0	=	WINMIN=SCALEMIN

X = XIN ( Ärvärdet )  
Y = Y ( Regulatorns utsignal ).  
Z = Inläst tecken  
N0 = Antalet decimaler i inläst tal. (inläsn. subr)  
N1 = Antalet heltal i inläst tal. " "  
D(10) = Decimaltalsvektor " "  
H(10) = Heltalsvektor " "  
S0 = Nytt samplingsvärde " "  
T8 = Teckentyp i " "  
T9 = TRY AGAIN " "  
N3 = SLASK I " "  
N4 = SLASK I " "  
R0 = Nytt RFEL-VÄRDE ( Parallell parameter till ) R  
Q1 = Nytt Q-VÄRDE ( Parallell parameter till ) Q  
K0 = Nytt KP-VÄRDE ( Parallell parameter till ) K1  
T0 = Nytt TI-VÄRDE ( parallell parameter till ) T1  
T = Nytt TD-VÄRDE ( parallell parameter till ) T2  
M5 = Nytt YMIN-VÄRDE ( Parallell parameter till ) M1  
M6 = Nytt YMAX-VÄRDE ( Parallell parameter till ) M2  
M7 = Nytt DYMAX-VÄRDE ( Parallell parameter till ) M3  
W2 = Nytt WIN-VÄRDE ( Parallell parameter till ) W  
U3 = NY FLAGGA TILL DIR/REV - " " T3  
C = Räkare för nytt TYPE-VÄRDE.  
J(10) = Lagring av ASCII-kod för ny Typ av regulator.

A1 = Variabel i PID-REGULATORN.  
 A2 = " " "  
 A4 = " " "  
 H = INTEGRALDELEN I REGULATORN !  
 N = DERIVATION GAIN I REGULATORN.  
 Y1,Y2 = Begränsning av utsignalen.  
 Z0 = Variabel i regul.  
 E = Aktuellt reglerfel.  
 P = Tilldelas värdet på inporten (ØECØØH) vid avsökning av knappatsen.  
 P0 = Tilldelas ASCII-kod för den knapp som trycks ned på knappatsen.  
 I7 = Räkare för antalet inslagna siffror på knappatsen  
 C = Tilldelas värde för GOSUB-hopp i knappatsen.  
 C0 = " " "  
 C4 = Flagga som talar om för subroutinen ALTER, att ett parameter-värde skall ändras.  
 C5 = Flagga som talar om för ALTER att TYPE skall ändras  
 R3 = Flagga för reglerfel. Om R3=1 (E>RFEL).  
 A5 = Används i subr. ROLL VALUE ON LCD.  
 A6 = " " "  
 X1 = Inläsningsvariabel till A/D-omvandlaren.  
 X2 = Inläsningsvariabel till A/D-omvandlaren.  
 Z7 = Flagga som är 1 då SCALEMIN,SCALEMAX och DIR/REV initieras annars är Z7=0.  
 Z8 = Suddar LCD då antalet siffror som skall skrivas ut är färre än gången innan. Detta förhindrar felaktig, och grötig överskrivning.

K7 = Skalningskonstant för 8-bitar.  
K8 = Skalningskonstant för 12-bitar.  
W7 = Skalat börvärde.  
X7 = Skalat ärvärde.  
C6 = Flagga för realtidsklockan. C6=1 då klockan är startad.  
R7 = Skalat RFEL.  
M9 = Skalat DYMAX.  
Y4 = Intern variabel för YMIN.  
Y5 = Intern variabel för YMAX.  
Y6 = Intern variabel för DYMAX.  
W8 = Intern variabel för INSIGNALEN.  
R4 = Intern variabel för RFEL.  
ø(1) = Teckensträng för regulator-typ.  
ø(2) = Teckensträng för DIR eller REV.

Om man av någon anledning vill utöka programmet, kan det vara lämpligt att utnyttja förteckningen över oanvända variabler ( se bilaga 8.3.1).



FÖRTECKNING ÖVER VARIABLER.

X = UPPTAGEN.

<del>A</del>	A 0	<del>A 1</del>	<del>A 2</del>	<del>A 3</del>	<del>A 4</del>	<del>A 5</del>	<del>A 6</del>	<del>A 7</del>	<del>A 8</del>	<del>A 9</del>
B	B 0	B 1	B 2	B 3	B 4	B 5	B 6	B 7	B 8	B 9
<del>C</del>	<del>C 0</del>	<del>C 1</del>	<del>C 2</del>	<del>C 3</del>	<del>C 4</del>	<del>C 5</del>	<del>C 6</del>	<del>C 7</del>	<del>C 8</del>	<del>C 9</del>
D	D 0	D 1	D 2	D 3	D 4	D 5	D 6	D 7	D 8	D 9
<del>E</del>	<del>E 0</del>	<del>E 1</del>	<del>E 2</del>	<del>E 3</del>	<del>E 4</del>	<del>E 5</del>	<del>E 6</del>	<del>E 7</del>	<del>E 8</del>	<del>E 9</del>
F	F 0	F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8	F 9
G	G 0	G 1	G 2	G 3	G 4	G 5	G 6	G 7	G 8	G 9
<del>H</del>	<del>H 0</del>	<del>H 1</del>	<del>H 2</del>	<del>H 3</del>	<del>H 4</del>	<del>H 5</del>	<del>H 6</del>	<del>H 7</del>	<del>H 8</del>	<del>H 9</del>
<del>I</del>	<del>I 0</del>	<del>I 1</del>	<del>I 2</del>	<del>I 3</del>	<del>I 4</del>	<del>I 5</del>	<del>I 6</del>	<del>I 7</del>	<del>I 8</del>	<del>I 9</del>
<del>J</del>	<del>J 0</del>	<del>J 1</del>	<del>J 2</del>	<del>J 3</del>	<del>J 4</del>	<del>J 5</del>	<del>J 6</del>	<del>J 7</del>	<del>J 8</del>	<del>J 9</del>
<del>K</del>	<del>K 0</del>	<del>K 1</del>	<del>K 2</del>	<del>K 3</del>	<del>K 4</del>	<del>K 5</del>	<del>K 6</del>	<del>K 7</del>	<del>K 8</del>	<del>K 9</del>
L	L 0	L 1	L 2	L 3	L 4	L 5	L 6	L 7	L 8	L 9
M	M 0	<del>M 1</del>	<del>M 2</del>	<del>M 3</del>	<del>M 4</del>	<del>M 5</del>	<del>M 6</del>	<del>M 7</del>	<del>M 8</del>	<del>M 9</del>
<del>N</del>	<del>N 0</del>	<del>N 1</del>	<del>N 2</del>	<del>N 3</del>	<del>N 4</del>	<del>N 5</del>	<del>N 6</del>	<del>N 7</del>	<del>N 8</del>	<del>N 9</del>
O	O 0	O 1	O 2	O 3	<del>O 4</del>	O 5	O 6	O 7	O 8	O 9
<del>P</del>	<del>P 0</del>	<del>P 1</del>	<del>P 2</del>	<del>P 3</del>	<del>P 4</del>	<del>P 5</del>	<del>P 6</del>	<del>P 7</del>	<del>P 8</del>	<del>P 9</del>
<del>Q</del>	<del>Q 0</del>	<del>Q 1</del>	<del>Q 2</del>	<del>Q 3</del>	<del>Q 4</del>	<del>Q 5</del>	<del>Q 6</del>	<del>Q 7</del>	<del>Q 8</del>	<del>Q 9</del>
<del>R</del>	<del>R 0</del>	<del>R 1</del>	<del>R 2</del>	<del>R 3</del>	<del>R 4</del>	<del>R 5</del>	<del>R 6</del>	<del>R 7</del>	<del>R 8</del>	<del>R 9</del>
<del>S</del>	<del>S 0</del>	<del>S 1</del>	<del>S 2</del>	<del>S 3</del>	<del>S 4</del>	<del>S 5</del>	<del>S 6</del>	<del>S 7</del>	<del>S 8</del>	<del>S 9</del>
<del>T</del>	<del>T 0</del>	<del>T 1</del>	<del>T 2</del>	<del>T 3</del>	<del>T 4</del>	<del>T 5</del>	<del>T 6</del>	<del>T 7</del>	<del>T 8</del>	<del>T 9</del>
U	U 0	U 1	U 2	<del>U 3</del>	U 4	U 5	U 6	U 7	U 8	U 9
V	V 0	V 1	V 2	V 3	V 4	V 5	V 6	V 7	V 8	V 9
<del>W</del>	<del>W 0</del>	<del>W 1</del>	<del>W 2</del>	<del>W 3</del>	<del>W 4</del>	<del>W 5</del>	<del>W 6</del>	<del>W 7</del>	<del>W 8</del>	<del>W 9</del>
<del>X</del>	<del>X 0</del>	<del>X 1</del>	<del>X 2</del>	<del>X 3</del>	<del>X 4</del>	<del>X 5</del>	<del>X 6</del>	<del>X 7</del>	<del>X 8</del>	<del>X 9</del>
<del>Y</del>	<del>Y 0</del>	<del>Y 1</del>	<del>Y 2</del>	<del>Y 3</del>	<del>Y 4</del>	<del>Y 5</del>	<del>Y 6</del>	<del>Y 7</del>	<del>Y 8</del>	<del>Y 9</del>
<del>Z</del>	<del>Z 0</del>	<del>Z 1</del>	<del>Z 2</del>	<del>Z 3</del>	<del>Z 4</del>	<del>Z 5</del>	<del>Z 6</del>	<del>Z 7</del>	<del>Z 8</del>	<del>Z 9</del>

## Bilaga 8.4

### LISTNING AV PROGRAMVARA

```

100 STRING 300,21
105 DIM A(15)
110 DIM B(10) : DIM M(15) : DIM L(15)
120 DIM E(15) : DIM J(10) : DIM H(10) : DIM D(10)
900 XBY(OF400H)=2
910 XBY(OF400H)=0
920 XBY(OE400H)=0
1000 XBY(OE800H)=192 : XBY(OF000H)=1 : XBY(OF000H)=6 : XBY(OF000H)=12
1005 XBY(OF000H)=56
1020 XBY(OE800H)=192 : XBY(OF000H)=2 : XBY(OF000H)=128 : XBY(OE800H)=64
1025 FOR I=1 TO 21
1030 READ A(I)
1035 XBY(OF000H)=A(I)
1040 NEXT I
1045 DATA 65,85,84,79,78,79,77,128,82,69,71,76,69,82,67,69
1050 DATA 78,84,82,65,76
1055 GOSUB 10200
1060 XBY(OE800H)=192 : XBY(OF000H)=2 : XBY(OF000H)=6 : XBY(OF000H)=192
1065 XBY(OE800H)=64 : XBY(OF000H)=80 : XBY(OF000H)=82 : XBY(OF000H)=69
1070 XBY(OF000H)=83 : XBY(OF000H)=83 : XBY(OF000H)=128 : XBY(OF000H)=60
1075 XBY(OF000H)=42 : XBY(OF000H)=62
1080 LET K2=10000 : K3=-1
1110 LET T4=100000000 : T5=0 : T6=100000000 : T7=0
1115 LET N=5
1150 PRINT CHR(5)
1155 PRINT : PRINT : PRINT : PRINT
1156 PRINT SPC(19), "*** AUTONOM REGLERCENRAL ***"
1157 PRINT : PRINT : PRINT
1160 PRINT "IF YOU WISH TO USE HAND-TERMINAL PRESS (*)"
1165 PRINT "ON HAND-TERMINAL."
1170 PRINT : PRINT
1175 PRINT "IF YOU WISH TO USE MONITOR TERMINAL PRESS (CR)"
1200 DO
1230 Z=GET
1240 XBY(OE800H)=1
1250 P=XBY(OEC00H)
1260 IF P=253 THEN GOTO 1300
1270 UNTIL Z=13
1280 GOTO 1606
1300 G1=7 : S0=1 : K7=10/255
1301 C5=0
1302 GOSUB 8555
1304 PRINT CHR(5)
1305 Q(1)="PID"
1306 M5=0 : M6=255 : M7=255
1308 Y4=0 : Y5=10 : Y6=10
1310 S=1 : R=50 : M1=0 : M2=10 : K1=10 : T1=7
1311 K0=10 : T0=8 : T=0.5
1315 T2=5 : T3=1 : W=50
1316 M3=50
1320 C6=0
1322 Z7=1 : C4=1
1323 Z9=10
1325 GOSUB 9757
1330 GOSUB 1355
1332 I7=0 : GOTO 8005
1336 DATA 83,67,65,76,69,77,73,78,61,63
1337 GOSUB 10220 : Z7=2
1338 GOSUB 1355
1339 I7=0 : GOTO 8005
1340 DATA 83,67,65,76,69,77,65,88,61,63

```

```

1341 C4=0 : C5=1
1342 Z9=13
1343 GOSUB 9757
1344 GOSUB 1355
1346 DATA 68,73,82,61,68,128,128,82,69,86,61,65,63
1348 GOTO 8005
1350 GOTO 2900
1355 FOR I=1 TO Z9
1357 READ A0
1358 XBY(OFOOOH)=A0
1360 NEXT I
1365 RETURN
1606 PRINT CHR(5)
1607 PRINT TAB (25),"INITIALITION MODE"
1608 LET Q=0
1609 PRINT : PRINT
1610 INPUT "REGULATOR TYPE (P,I,D,PI,PD,ID,PID) ?>",R(1)
1612 IF ASC(R(1),1)=80.AND.ASC(R(1),2)=13 THEN Q=1
1614 IF ASC(R(1),1)=73.AND.ASC(R(1),2)=13 THEN Q=2
1616 IF ASC(R(1),1)=68.AND.ASC(R(1),2)=13 THEN Q=3
1618 IF ASC(R(1),1)=80.AND.ASC(R(1),2)=73.AND.ASC(R(1),3)=13 THEN Q=4
1620 IF ASC(R(1),1)=80.AND.ASC(R(1),2)=68.AND.ASC(R(1),3)=13 THEN Q=5
1622 IF ASC(R(1),1)=73.AND.ASC(R(1),2)=68.AND.ASC(R(1),3)=13 THEN Q=6
1624 IF ASC(R(1),1)=80.AND.ASC(R(1),2)=73.AND.ASC(R(1),3)=68 THEN Q=7
1630 ON Q GOTO 1632,1636,1636,1636,1636,1636,1636
1632 PRINT "WRONG TYPE " : GOTO 1610
1636 DO
1638 INPUT "SAMPLING PERIOD S (1-255 SEC.) ?>",S
1640 IF S>255 THEN PRINT "SMAX=255"
1642 IF S<1 THEN PRINT "SMIN=1"
1644 UNTIL S>=1.AND.S<=255
1650 INPUT "LOW OUTPUT LIMIT YMIN (0-10 VOLT) ?>",M1
1652 IF M1>=0.AND.M1<=10 THEN GOTO 1658
1653 IF M1<0 THEN PRINT "MINIMUM = 0 V"
1654 IF M1>10 THEN PRINT "MAXIMUM = 10 V"
1655 PRINT
1656 PRINT "TRY AGAIN"
1657 GOTO 1650
1658 INPUT "HIGH OUTPUT LIMIT YMAX (0-10 VOLT) ?>",M2
1659 IF M2>=0.AND.M2<=10 THEN GOTO 1665
1660 IF M2<0 THEN PRINT "MINIMUM = 0 V"
1661 IF M2>10 THEN PRINT "MAXIMUM=10 VOLT"
1662 PRINT
1663 PRINT "TRY AGAIN"
1664 GOTO 1658
1665 IF M2>M1 THEN GOTO 1670
1666 PRINT
1667 PRINT "YMAX <= YMIN ! TRY AGAIN"
1668 GOTO 1650
1670 K7=10/255
1672 Y1=M1 : Y2=M2
1674 M1=M1/K7 : M2=M2/K7
1676 INPUT "MAX DER.OUTPUT DYMAX (VOLT) ?>",M3
1677 M3=ABS(M3)
1678 M9=M3
1679 M3=M3/K7
1682 IF Q=2.OR.Q=3.OR.Q=6 THEN GOTO 1694
1685 DO
1686 INPUT "GAIN KP ?>",K1
1688 IF K1>K2 THEN PRINT "KPMAX=",K2
1691 IF K1<K3 THEN PRINT "KPMIN=",K3
1693 UNTIL K1<=K2.AND.K1>=K3

```

```

1694 IF Q=1.OR.Q=3.OR.Q=5 THEN GOTO 1704
1695 DO
1696 INPUT "INTEGRATION TIME TI" ?>,T1
1697 IF T1>T4 THEN PRINT "TIMAX=",T4
1700 IF T1<T5 THEN PRINT "TIMIN=",T5
1702 UNTIL T1<=T4.AND.T1>=T5
1704 IF Q=1.OR.Q=2.OR.Q=4 THEN GOTO 1753
1705 DO
1706 INPUT "DERIVATION TIME TD" ?>,T2
1707 IF T2>T6 THEN PRINT "TDMAX=",T6
1709 IF T2<T7 THEN PRINT "TDMIN=",T7
1711 UNTIL T2<=T6.AND.T2>=T7
1753 INPUT "SCALE MIN" ?>,WO
1755 INPUT "SCALE MAX" ?>,W1
1757 IF W1>WO THEN GOTO 1770
1760 PRINT
1762 PRINT "SCALEMAX <= SCALEMIN ! TRY AGAIN"
1764 GOTO 1753
1769 LET T3=0
1770 INPUT "DIRECT OR REVERSE UTYPE (DIR/REV) ?>",Q(2)
1772 IF ASC(Q(2),1)=68.AND.ASC(Q(2),2)=73.AND.ASC(Q(2),3)=82 THEN T3=1
1773 IF ASC(Q(2),1)=82.AND.ASC(Q(2),2)=69.AND.ASC(Q(2),3)=86 THEN T3=2
1774 IF T3=1.XOR.T3=2 THEN GOTO 1778
1775 PRINT
1776 PRINT "TRY AGAIN"
1777 GOTO 1770
1778 IF T3=1 THEN Q(2)="DIR" ELSE Q(2)="REV"
1780 INPUT "MAX ERROR RFEL (SCALE UNIT) ?>",R7
1782 R7=ABS(R7)
1790 DO
1793 INPUT "REFERENCE WIN (SCALEMIN-SCALEMAX) ?>",W
1795 IF W>W1 THEN PRINT "SCALEMAX=",W1
1797 IF W<WO THEN PRINT "SCALEMIN=",WO
1799 UNTIL W<=W1.AND.W>=WO
1800 K8=(W1-WO)/255
1801 W7=W
1802 W=(W7-WO)/K8
1904 R=R7/K8
1910 PUSH Y1,Y2,M9,W7,R7
1811 POP R4,W8,Y6,Y5,Y4
1900 PRINT
1910 PRINT "MEASUREMENT XIN (SCALE UNIT)"
1911 PRINT "OUTPUT Y (YMIN-YMAX)"
1912 PRINT
1915 PRINT "TO START THE REGULATOR PRESS (CR)!",
1920 DO
1922 Z=BET
1924 UNTIL Z=13
2800 TIME=0 : CLOCK 1 : ONTIME S,10000
2810 C6=1
2900 PRINT CHR(5)
2902 PRINT
2904 SPC (23),"REGULATOR PARAMETERS"
2906 PRINT : PRINT : PRINT
2990 PRINT "DO YOU WISH TO CHANGE PARAMETERS? (Y) "
2991 PRINT "TO CLEAR ALARM (C)"
2992 PRINT : PRINT : PRINT
3000 SPC (3),"TYPE=",Q(1), TAB (30),"S=",S, TAB (57),"RFEL=",R7
3005 PRINT
3010 SPC (3),"KP=",K1, TAB (30),"TD=",T2, TAB (57),"TI=",T1
3015 PRINT
3020 SPC (3),"DYMAX=",M9, TAB (30),"YMIN=",Y1, TAB (57),"YMAX=",Y2
3022 PRINT
3023 SPC (3),"WIN=",W7, TAB (30),"SCALEMIN=",WO, TAB (57),"SCALEMAX=",

```

```

3025 PUSH W,Q,S,R,K1,T1,T2,M1,M2,M3
3026 POP M7,M6,M5,T,TO,KO,RO,SO,Q1,W2
3027 PRINT
3030 DO
3032 XBY(OE800H)=I
3034 P=XBY(OE000H)
3036 IF P=253 THEN GOTO 8000
3052 PRINT SPC (3),"XIN=",X7, TAB (30),"Y=",Y7, TAB (57),"UTYP=",X(2), CR ,
3053 Z=GET
3054 IF Z=67 THEN GOSUB 7985
3056 UNTIL Z=89
3058 PRINT CHR(5)
3059 PRINT TAB (25),"CHANGE MODE" : PRINT : PRINT
3060 PRINT "IF YOU DON'T WANT TO CHANGE THE PARAMETER PRESS (CR)"
3061 PRINT : PRINT
3062 PRINT "IF YOU DON'T WANT TO CHANGE ANY MORE PARAMETERS PRESS Q"
3070 PRINT : PRINT : PRINT : PRINT
3090 PRINT "TYPE=",X(1), TAB (19),"TYPE ?>",
3165 LET C=0
3170 DO
3173 DO
3175 Z=GET
3180 IF Z<>81.AND.Z<>13 THEN PRINT CHR(Z),
3185 UNTIL Z<>0
3190 LET C=C+1
3191 IF C>10 THEN GOTO 3218
3192 LET J(C)=Z
3195 UNTIL Z=13.OR.Z=81
3197 IF J(1)=81 THEN GOTO 6000
3199 IF J(1)=13 THEN GOTO 3400
3200 LET Q1=0
3210 GOSUB 5200
3215 IF Q1>0 THEN GOTO 3230
3217 LET Q1=0
3218 PRINT " WRONG TYPE"
3219 PRINT
3221 PRINT "TRY AGAIN"
3225 GOTO 3090
3230 GOSUB 5220
3235 IF Z=81 THEN GOTO 6000
3400 PRINT
3401 PRINT "WIN=",W7, TAB (19),"WIN ?>",
3402 GOSUB 5000
3404 GOSUB 4900
3410 IF T9=1 THEN GOTO 3401
3416 GOSUB 4950
3433 LET W2=N4
3440 IF NO=0.AND.N1=0 THEN W2=W7
3441 IF W2>=WO.AND.W2<=W1 THEN GOTO 3446
3442 IF W2>W1 THEN PRINT " SCALEMAX=",W1,
3443 IF W2<WO THEN PRINT " SCALEMIN=",WO,
3444 LET T9=1
3445 GOTO 3404
3446 W7=W2
3447 W2=(W2-WO)/KB
3448 IF Z=81 THEN GOTO 6000
3450 PRINT
3451 PRINT "S=",S, TAB (19),"S ?>",
3452 GOSUB 5000
3454 GOSUB 4900
3460 IF T9=1 THEN GOTO 3451

```

```

3466 GOSUB 4950
3484 LET SO=N4
3490 IF NO=0.AND.N1=0 THEN SO=S
3491 IF SO>=1.AND.SO<=255 THEN GOTO 3496
3492 IF SO>255 THEN PRINT " SMAX=255",
3493 IF SO<1 THEN PRINT " SMIN=1",
3494 LET T9=1
3495 GOTO 3454
3496 IF Z=81 THEN GOTO 6000
3500 IF Q1=2.OR.Q1=3.OR.Q1=6 THEN GOTO 3550
3501 PRINT
3502 PRINT "KP=",K1, TAB (19),"KP  ?>",
3503 GOSUB 5000
3504 GOSUB 4900
3510 IF T9=1 THEN GOTO 3502
3516 GOSUB 4950
3534 LET KO=N4
3538 IF NO=0.AND.N1=0 THEN KO=K1
3539 IF KO>=K3.AND.KO<=K2 THEN GOTO 3546
3540 IF KO>K2 THEN PRINT " KPMAX=",K2,
3542 IF KO<K3 THEN PRINT " KPMIN=",K3,
3543 LET T9=1
3544 GOTO 3504
3546 IF Z=81 THEN GOTO 6000
3550 IF Q1=1.OR.Q1=3.OR.Q1=5 THEN GOTO 3600
3551 PRINT
3552 PRINT "TI=",T1, TAB (19),"TI  ?>",
3553 GOSUB 5000
3554 GOSUB 4900
3560 IF T9=1 THEN GOTO 3552
3566 GOSUB 4950
3584 LET T0=N4
3588 IF NO=0.AND.N1=0 THEN T0=T1
3589 IF T0>=T5.AND.T0<=T4 THEN GOTO 3596
3590 IF T0>T4 THEN PRINT " TIMAX=",T4,
3592 IF T0<T5 THEN PRINT " TIMIN=",T5,
3593 LET T9=1
3594 GOTO 3554
3596 IF Z=81 THEN GOTO 6000
3600 IF Q1=1.OR.Q1=2.OR.Q1=4 THEN GOTO 3650
3584 PRINT
3602 PRINT "TD=",T2, TAB (19),"TD  ?>",
3603 GOSUB 5000
3604 GOSUB 4900
3610 IF T9=1 THEN GOTO 3602
3616 GOSUB 4950
3634 LET T=N4
3638 IF NO=0.AND.N1=0 THEN T=T2
3639 IF T>=T7.AND.T<=T6 THEN GOTO 3646
3640 IF T>T6 THEN PRINT " TDMAX=",T6,
3642 IF T<T7 THEN PRINT " TDMIN=",T7,
3643 LET T9=1
3644 GOTO 3604
3646 IF Z=81 THEN GOTO 6000
3650 PRINT
3652 PRINT "YMIN=",Y1, TAB (19),"YMIN ?>",
3653 GOSUB 5000
3654 GOSUB 4900
3660 IF T9=1 THEN GOTO 3652
3666 GOSUB 4950

```

```

3684 LET M5=N4
3687 IF NO=0.AND.N1=0 THEN M5=Y1
3688 IF M5<0.OR.M5>10 THEN GOTO 3900
3689 IF (M5<Y2).AND.(Z=81) THEN GOTO 3904
3690 IF Z=13 THEN GOTO 3908
3694 PRINT " YMAX < YMIN YMAX=",Y2,
3696 LET T9=1
3698 GOTO 3654
3700 PRINT
3702 PRINT "YMAX=",Y2, TAB (19),"YMAX ?>",
3703 GOSUB 5000
3704 GOSUB 4900
3710 IF T9=1 THEN GOTO 3702
3716 GOSUB 4950
3734 LET M6=N4
3737 IF NO=0.AND.N1=0 THEN M6=Y2
3738 IF (M6<0).OR.(M6>10) THEN GOTO 3912
3739 IF M6>=Y1 THEN GOTO 3916
3742 PRINT " YMAX < YMIN !",
3743 LET T9=1
3744 GOTO 3654
3746 IF Z=81 THEN GOTO 6000
3750 PRINT
3752 PRINT "DYMAX=",M9, TAB (19),"DYMAX?>",
3753 GOSUB 5000
3754 GOSUB 4900
3760 IF T9=1 THEN GOTO 3752
3766 GOSUB 4950
3784 LET M7=N4
3787 IF NO=0.AND.N1=0 THEN M7=M9
3788 M9=ABS(M7)
3789 M7=ABS(M7/K7)
3790 IF Z=81 THEN GOTO 6000
3800 PRINT
3802 PRINT "RFEL=",R7, TAB (19),"RFEL ?>",
3803 GOSUB 5000
3804 GOSUB 4900
3810 IF T9=1 THEN GOTO 3802
3816 GOSUB 4950
3834 LET R0=N4
3840 IF NO=0.AND.N1=0 THEN R0=R7
3842 R7=ABS(R0)
3844 R0=ABS(R0/K8)
3846 GOTO 6000
3900 PRINT " MAXIMUM = 10 VOLT MINIMUM = 0 VOLT"
3901 T9=1
3903 GOTO 3654
3904 Y1=M5
3905 M5=M5/K7
3907 GOTO 6000
3908 Y1=M5
3909 M5=M5/K7
3911 GOTO 3700
3912 PRINT " MAXIMUM = 10 VOLT MINIMUM = 0 VOLT"
3913 T9=1
3915 GOTO 3704
3916 Y2=M6
3917 M6=M6/K7
3919 GOTO 3746
4900 IF T9=0 THEN GOTO 4920
4905 PRINT : PRINT

```



```

4915 PRINT "TRY AGAIN"
4920 RETURN
4950 LET N4=0
4955 IF N1>0 THEN GOSUB 5100
4960 IF N0>0 THEN GOSUB 5110
4965 IF T8=1 THEN N4=-N4
4970 RETURN
5000 LET T8=0 : T9=0 : N0=0 : N1=0
5008 GOSUB 5070
5010 IF Z<>45 THEN 5020
5012 LET T8=1
5014 PRINT CHR(Z),
5016 GOSUB 5070
5020 IF Z=81.OR.Z=13 THEN RETURN
5030 IF Z=46 THEN 5050
5032 LET N1=N1+1
5033 IF N1>10 THEN T9=1
5034 PRINT CHR(Z),
5035 IF T9=1 THEN RETURN
5036 GOSUB 5080
5038 LET H(N1)=N3
5040 GOSUB 5070
5042 IF Z=81.OR.Z=13 THEN RETURN ELSE GOTO 5030
5050 PRINT CHR(Z),
5055 GOSUB 5070
5057 IF Z=81.OR.Z=13 THEN RETURN
5060 LET N0=N0+1
5061 IF N0>10 THEN T9=1
5062 PRINT CHR(Z),
5063 IF T9=1 THEN RETURN
5064 GOSUB 5080
5066 LET D(N0)=N3
5068 GOTO 5055
5070 DO
5072 Z=GET
5074 UNTIL Z<>0
5076 RETURN
5080 IF Z=48 THEN N3=0
5081 IF Z=49 THEN N3=1
5082 IF Z=50 THEN N3=2
5083 IF Z=51 THEN N3=3
5084 IF Z=52 THEN N3=4
5085 IF Z=53 THEN N3=5
5086 IF Z=54 THEN N3=6
5087 IF Z=55 THEN N3=7
5088 IF Z=56 THEN N3=8
5089 IF Z=57 THEN N3=9
5090 IF Z<48.OR.Z>57 THEN T9=1
5092 RETURN
5100 LET J=1
5102 FOR I=N1 TO 1 STEP -1
5103 LET N4=N4+H(I)*J
5104 LET J=J*10
5106 NEXT I
5108 RETURN
5110 LET J=10
5112 FOR I=1 TO N0
5113 LET N4=N4+D(I)/J
5114 LET J=J*10
5116 NEXT I
5118 RETURN

```

```

5200 IF J(1)=80.AND.(J(2)=13.OR.J(2)=81) THEN Q1=1
5202 IF J(1)=73.AND.(J(2)=13.OR.J(2)=81) THEN Q1=2
5204 IF J(1)=68.AND.(J(2)=13.OR.J(2)=81) THEN Q1=3
5206 IF J(1)=80.AND.J(2)=73.AND.(J(3)=13.OR.J(3)=81) THEN Q1=4
5208 IF J(1)=80.AND.J(2)=68.AND.(J(3)=13.OR.J(3)=81) THEN Q1=5
5210 IF J(1)=73.AND.J(2)=68.AND.(J(3)=13.OR.J(3)=81) THEN Q1=6
5212 IF J(1)=80.AND.J(2)=73.AND.J(3)=68.AND.(J(4)=13.OR.J(4)=81) THEN Q1=7
5214 RETURN
5220 IF Q1=1 THEN Q(1)="P "
5222 IF Q1=2 THEN Q(1)="I "
5224 IF Q1=3 THEN Q(1)="D "
5226 IF Q1=4 THEN Q(1)="PI "
5228 IF Q1=5 THEN Q(1)="PD "
5230 IF Q1=6 THEN Q(1)="ID "
5232 IF Q1=7 THEN Q(1)="PID"
5234 RETURN
( 70 PUSH S0,Q1,R0,K0,T0,T,M5,M6,M7,W2
6002 POP W,M3,M2,M1,T2,T1,K1,R,Q,S
6003 PUSH Y1,Y2,M9,W7,R7
6004 POP R4,W8,Y6,Y5,Y4
6008 GOTO 2900
( 10 PUSH S0,Q1,R0,K0,T0,T,M5,M6,M7,W2,R4,W8,Y4,Y5,Y6
6012 POP M9,Y2,Y1,W7,R7,W,M3,M2,M1,T2,T1,K1,R,Q,S
6013 IF C6=0 THEN GOTO 2800
6014 GOTO 2900
7000 REM * P-REGULATOR *
7020 A1=0 : A2=0 : A4=0 : H=0
7030 GOSUB 7725
7050 RETURN
7100 REM * I-REGULATOR *
7120 PUSH K1
7130 A2=0 : A4=0 : K1=0
7135 IF T1=0 THEN A1=0 ELSE A1=S/T1
7140 GOSUB 7725
7150 POP K1
7160 RETURN
7200 REM * D-REGULATOR *
7210 A2=S*N/(T2+N*S)
7220 PUSH K1
7230 A1=0 : K1=0 : H=0
7235 A4=N
7240 GOSUB 7725
7250 POP K1
7260 RETURN
7300 RETURN
7400 REM *** PI-REGULATOR ***
7405 LET A1=K1*S/T1
7415 A2=0 : A4=0
7420 GOSUB 7725
7440 RETURN
7500 REM ** PD-REGULATOR **
7505 LET A2=N*S/(T2+N*S)
7506 LET A4=K1*N
7515 A1=0 : H=0
7520 GOSUB 7725
7540 RETURN
7600 REM * ID-REGULATOR *
7620 A2=N*S/(T2+N*S)
7630 PUSH K1
7640 K1=0 : A4=N
7645 IF T1=0 THEN A1=0 ELSE A1=S/T1

```

```

7650 GOSUB 7725
7660 POP K1
7666 RETURN
7700 REM *** PID-REGULATOR ***
7710 REM ** PRE-COMPUTATIONS **
7712 IF T1=0 THEN A1=0 ELSE A1=K1*S/T1
7715 A2=N*S/(T2+N*S)
7720 A4=K1*N
7725 REM **** MAIN PROGRAM ****
7730 X1=XBY(OE000H) : X2=XBY(OE001H)
7733 X=((16*X1+X2/16)/16)
7735 E=W-X
7736 IF T3=2 THEN E=X-W
7738 GOSUB 7950
7740 ZO=ZO-A2*(ZO+X)
7744 YO=Y
7745 V=K1*E+H+A4*(E-ZO)
7747 Y=V
7751 IF ABS(Y-YO)<M3 THEN GOTO 7758
7752 IF Y>YO THEN Y=YO+M3
7753 IF Y<YO THEN Y=YO-M3
7755 IF Y<M1 THEN Y=M1
7760 IF Y>M2 THEN Y=M2
7764 XBY(OE400H)=Y
7765 H=H+Y-V+A1*E
7766 X7=K8*X+W0
7768 Y7=K7*Y
7770 RETURN
7950 IF ABS(E)>R THEN R3=1 ELSE R3=0
7955 IF O4=1 THEN GOTO 7970
7960 IF R3=1 THEN XBY(OF400H)=1 ELSE XBY(OF400H)=0
7965 RETURN
7970 IF R3=1 THEN XBY(OF400H)=3 ELSE XBY(OF400H)=2
7975 IF R3=0 THEN O4=0
7980 RETURN
7985 LET O4=1
7990 IF R3=1 THEN XBY(OF400H)=3 ELSE XBY(OF400H)=2
7995 RETURN
8000 PRINT CHR(5)
8001 GOSUB 8555
8003 P=255 : I7=0 : A6=0
8004 GOSUB 8945
8005 XBY(OE800H)=1
8007 P=XBY(OEC00H)
8010 IF P=255 THEN GOTO 8100
8015 IF P=253 THEN GOTO 8700
8020 IF P=239 THEN PO=ASC(0)
8025 IF P=247 THEN PO=ASC(.)
8030 IF P=251 THEN PO=45
8040 IF P=254 THEN GOTO 8900
8050 GOSUB 8500
8100 XBY(OE800H)=2
8105 P=XBY(OEC00H)
8110 IF P=255 THEN GOTO 8200
8115 IF P=254 THEN GOTO 8910
8120 IF P=239 THEN PO=ASC(1)
8130 IF P=247 THEN PO=ASC(2)
8135 IF P=251 THEN PO=ASC(3)
8140 IF P=253 THEN PO=ASC(D)
8150 GOSUB 8500
8200 XBY(OE800H)=4

```

```

8205 P=XBY(OE000H)
8210 IF P=255 THEN GOTO 8300
8220 IF P=254 THEN GOTO 8920
8225 IF P=239 THEN PO=ASC(4)
8230 IF P=247 THEN PO=ASC(5)
8235 IF P=251 THEN PO=ASC(6)
8240 IF P=253 THEN PO=ASC(1)
8245 GOSUB 8500
8300 XBY(OE800H)=8
8305 P=XBY(OE000H)
8310 IF P=255 THEN GOTO 8005
8315 IF P=254 THEN GOTO 8930
8320 IF P=239 THEN PO=ASC(7)
8325 IF P=247 THEN PO=ASC(8)
8330 IF P=251 THEN PO=ASC(9)
8335 IF P=253 THEN PO=ASC(P)
8340 GOSUB 8500
8345 GOTO 8005
8500 DO
8505 REM * IDIOT LOOP *
8510 P=P
8515 UNTIL P<>XBY(OE000H)
8520 I7=I7+1
8522 IF I7<12 THEN GOTO 8530
8524 IF I7=0 THEN GOTO 8710
8526 RESTORE : GOTO 1000
8530 E(I7)=PO
8540 XBY(OE800H)=64
8545 XBY(OF000H)=PO
8550 RETURN
8555 DO
8556 REM * IDIOT LOOP *
8557 P=P
8558 UNTIL P<>XBY(OE000H)
8560 RETURN
8561 GOSUB 9757
8562 GOSUB 10220
8564 GOSUB 9701
8566 XBY(OE800H)=1
8568 P=XBY(OE000H)
8570 IF P=253 THEN RETURN
8572 XBY(OE800H)=4
8574 P=XBY(OE000H)
8576 IF P=254 THEN RETURN
8578 GOSUB 8780
8580 GOSUB 9702
8582 IF I7<Z8 THEN GOSUB 8795
8584 Z8=A9
8585 XBY(OE800H)=8
8586 P=XBY(OE000H)
8587 IF P<>254 THEN GOTO 8566
8588 R4=1
8590 IF R3=1 THEN XBY(OF400H)=3 ELSE XBY(OF400H)=2
8592 GOTO 8566
8600 REM * SUBROUTINE FOR ROLL ON LCD *
8610 A5=1 : A6=A6+1
8612 IF A6>1 THEN GOTO 8620
8616 GOSUB 9757
8620 ON A6 GOSUB 8630,8635,8648,8648,8648,8660,8687,8687,8692,8695
8630 A5=0 : RETURN
8635 GOSUB 9100

```

```

8636 XBY(OE800H)=192 : XBY(OF000H)=2 : XBY(OF000H)=142 : XBY(OE800H)=64
8637 GOSUB 9151
8638 XBY(OE800H)=192 : XBY(OF000H)=7 : XBY(OF000H)=157 : XBY(OE800H)=64
8639 GOSUB 9201
8642 XBY(OE800H)=192 : XBY(OF000H)=6 : XBY(OF000H)=2 : XBY(OF000H)=192
8643 XBY(OE800H)=64 : GOSUB 9301
8645 XBY(OE800H)=192 : XBY(OF000H)=6 : XBY(OF000H)=2 : XBY(OF000H)=203
8646 XBY(OE800H)=64 : GOSUB 9351
8648 XBY(OE800H)=192 : XBY(OF000H)=7 : XBY(OF000H)=214 : XBY(OE800H)=64
8649 GOSUB 9402
8651 RETURN
8660 GOSUB 9757
8662 GOSUB 9451
8664 XBY(OE800H)=192 : XBY(OF000H)=2 : XBY(OF000H)=6 : XBY(OF000H)=142
8666 XBY(OE800H)=64 : GOSUB 9501
8676 XBY(OE800H)=192 : XBY(OF000H)=6 : XBY(OF000H)=192 : XBY(OE800H)=64
8680 XBY(OE800H)=192 : XBY(OF000H)=6 : XBY(OF000H)=203 : XBY(OE800H)=64
( 32 GOSUB 9551
8587 XBY(OE800H)=192 : XBY(OF000H)=7 : XBY(OF000H)=157 : XBY(OE800H)=64
8688 GOSUB 9251
8690 RETURN
8692 GOSUB 8720
( 73 GOSUB 8561
8695 A6=0 : RETURN
8700 GOSUB 8555
8702 IF Z7=0 THEN GOTO 8705
8703 RESTORE : GOTO 1000
8705 IF I7=2 THEN GOTO 8995
8707 C4=0 : C5=0
8710 I7=0 : GOTO 9050
8720 REM *SURROUTINE FOR LOPANDE UTSKR. PA LCD *
8722 GOSUB 9757
8724 GOSUB 9151
8726 GOSUB 10220
8727 GOSUB 9751
8734 XBY(OE800H)=1
8736 P=XBY(OEC00H)
8738 IF P=253 THEN RETURN
8740 XBY(OE800H)=4
8742 P=XBY(OEC00H)
8744 IF P=254 THEN RETURN
8748 GOSUB 8790
8754 GOSUB 9752
8756 IF I7<Z8 THEN GOSUB 8795
8757 Z8=A9
8760 XBY(OE800H)=8
8762 P=XBY(OEC00H)
8764 IF P<>254 THEN GOTO 8734
8766 LET R4=1
8767 IF R3=1 THEN XBY(OF400H)=3 ELSE XBY(OF400H)=2
8777 GOTO 8734
8780 XBY(OE800H)=192 : XBY(OF000H)=2 : XBY(OF000H)=194 : XBY(OE800H)=64
8782 RETURN
8790 XBY(OE800H)=192 : XBY(OF000H)=2 : XBY(OF000H)=196 : XBY(OE800H)=64
8792 RETURN
8794 GOSUB 8790
8795 FOR I=1 TO Z8
8796 XBY(OF000H)=128
8797 NEXT I : RETURN
8900 REM * HOPP TILL ALTER-MODE *
8901 GOSUB 8555

```

```

8902 IF Z7<>0.AND.C5=1 THEN GOTO 11150
8903 IF Z7=0 THEN GOTO 8908
8904 IF Z7<>0.AND.I7<>0 THEN GOTO 8908
8905 RESTORE : GOTO 1000
8908 GOTO 9760
8910 REM * ENTER NEW DATA TO REGUL *
8911 IF Z7=0 THEN GOTO 8913
8912 RESTORE : GOTO 1000
8913 GOSUB 5220
8914 GOSUB 9757
8915 GOTO 6010
8920 REM *ROLL VALUE ON LCD *
8921 IF Z7=0 THEN GOTO 8924
8922 RESTORE : GOTO 1000
8923 GOSUB 8555
8924 GOSUB 8600
8925 GOTO 8005
8930 04=1 : IF Z7<>0 THEN GOTO 8703
8931 GOSUB 8555
8932 IF R3=1 THEN XBY(OF400H)=3 ELSE XBY(OF400H)=2
8934 GOSUB 9757
8937 XBY(OF000H)=65 : XBY(OF000H)=76 : XBY(OF000H)=65 : XBY(OF000H)=82
8938 XBY(OF000H)=77 : XBY(OF000H)=128 : XBY(OF000H)=67 : XBY(OF000H)=76
8939 XBY(OF000H)=69 : XBY(OF000H)=65 : XBY(OF000H)=82
8940 XBY(OF000H)=69 : XBY(OF000H)=68
8941 C0=0 : C=0
8942 I7=0 : C4=0 : C5=0 : GOTO 8004
8945 GOSUB 10200
8950 GOSUB 10220
8952 XBY(OF000H)=67 : XBY(OF000H)=79
8954 XBY(OF000H)=68 : XBY(OF000H)=69 : XBY(OF000H)=63
8955 RETURN
8995 C0=0 : C=0 : A6=0
8996 IF C4<>0.OR.C5<>0 THEN GOTO 8004
8997 C4=0 : C5=0
9000 IF E(1)=49.AND.E(2)=48 THEN C0=1
9002 IF E(1)=50.AND.E(2)=49 THEN C0=2
9004 IF E(1)=51.AND.E(2)=49 THEN C0=3
9006 IF E(1)=52.AND.E(2)=49 THEN C0=4
9008 IF E(1)=53.AND.E(2)=49 THEN C0=5
9010 IF E(1)=54.AND.E(2)=49 THEN C0=6
9012 IF E(1)=55.AND.E(2)=49 THEN C0=7
9014 IF E(1)=56.AND.E(2)=49 THEN C=1
9016 IF E(1)=56.AND.E(2)=50 THEN C=2
9018 IF E(1)=56.AND.E(2)=51 THEN C=3
9020 IF E(1)=56.AND.E(2)=53 THEN C=4
9022 IF E(1)=56.AND.E(2)=54 THEN C=5
9024 IF E(1)=57.AND.E(2)=49 THEN C=6
9026 IF E(1)=57.AND.E(2)=53 THEN C=7
9028 I7=0
9029 ON C0 GOTO 9030,9100,9150,9200,9250,9300,9350,9400
9030 ON C GOTO 9050,9450,9500,9550,9600,9650,9700,9750
9050 GOSUB 9757
9051 XBY(OF000H)=87 : XBY(OF000H)=82 : XBY(OF000H)=79 : XBY(OF000H)=78
9052 XBY(OF000H)=71 : XBY(OF000H)=128 : XBY(OF000H)=67 : XBY(OF000H)=79
9053 XBY(OF000H)=68 : XBY(OF000H)=69 : XBY(OF000H)=33
9055 GOTO 8004
9100 GOSUB 9757
9101 XBY(OF000H)=84 : XBY(OF000H)=89 : XBY(OF000H)=80
9102 XBY(OF000H)=69 : XBY(OF000H)=61
9103 FOR I=1 TO 3

```

```

9105 XBY (OF000H)=ASC(α(1),I)
9110 NEXT I
9111 IF A5=1 THEN RETURN
9120 GOTO 8004
9150 GOSUB 9757
9151 XBY (OF000H)=87 : XBY (OF000H)=73 : XBY (OF000H)=78 : XBY (OF000H)=61
9152 AB=W7 : GOSUB 11400
9153 GOSUB 9720
9154 IF A5=1 THEN RETURN
9155 GOTO 8004
9200 GOSUB 9757
9201 XBY (OF000H)=83 : XBY (OF000H)=61
9202 AB=S
9203 GOSUB 11400
9204 GOSUB 9720
9205 IF A5=1 THEN RETURN
9210 GOTO 8004
9250 GOSUB 9757
9251 XBY (OF000H)=82 : XBY (OF000H)=70 : XBY (OF000H)=69 : XBY (OF000H)=76
9252 XBY (OF000H)=61
9253 AB=R7 : GOSUB 11400
9254 GOSUB 9720
9255 IF A5=1 THEN RETURN
9260 GOTO 8004
9300 GOSUB 9757
9301 XBY (OF000H)=75 : XBY (OF000H)=80 : XBY (OF000H)=61
9303 AB=K1 : GOSUB 11400
9304 GOSUB 9720
9305 IF A5=1 THEN RETURN
9310 GOTO 8004
9350 GOSUB 9757
9351 XBY (OF000H)=84 : XBY (OF000H)=68 : XBY (OF000H)=61
9353 AB=T2 : GOSUB 11400
9355 GOSUB 9720
9358 IF A5=1 THEN RETURN
9360 GOTO 8004
9400 GOSUB 9757
9402 XBY (OF000H)=84 : XBY (OF000H)=73 : XBY (OF000H)=61
9404 AB=T1 : GOSUB 11400
9405 GOSUB 9720
9406 IF A5=1 THEN RETURN
9410 GOTO 8004
9450 GOSUB 9757
9451 XBY (OF000H)=89 : XBY (OF000H)=77 : XBY (OF000H)=73
9452 XBY (OF000H)=78 : XBY (OF000H)=61
9453 AB=Y1 : GOSUB 11400
9454 GOSUB 9720
9455 IF A5=1 THEN RETURN
9457 GOTO 8004
9500 GOSUB 9757
9501 XBY (OF000H)=89 : XBY (OF000H)=77 : XBY (OF000H)=65
9502 XBY (OF000H)=88 : XBY (OF000H)=61
9503 AB=Y2 : GOSUB 11400
9504 GOSUB 9720
9505 IF A5=1 THEN RETURN
9510 GOTO 8004
9550 GOSUB 9757
9551 XBY (OF000H)=68 : XBY (OF000H)=89 : XBY (OF000H)=77 : XBY (OF000H)=65
9552 XBY (OF000H)=88 : XBY (OF000H)=61
9553 AB=M9 : GOSUB 11400
9554 GOSUB 9720

```

```

9555 IF A5=1 THEN RETURN
9557 GOTO 8004
9600 IF A5=1 THEN RETURN
9610 GOTO 8004
9650 GOTO 9050
9700 GOSUB 9757
9701 XBY(OFO00H)=89 : XBY(OFO00H)=61
9702 A8=Y7 : GOSUB 11400
9703 GOSUB 9720
9707 IF A5=1 THEN RETURN
9710 GOTO 8004
9720 FOR I=1 TO A9
9722 XBY(OFO00H)=A(I)
9725 NEXT I
9727 RETURN
9750 GOSUB 9757
C 31 XBY(OFO00H)=88 : XBY(OFO00H)=73 : XBY(OFO00H)=78 : XBY(OFO00H)=61
9752 A8=X7 : GOSUB 11400
9753 GOSUB 9720
9754 IF A5=1 THEN RETURN
9755 GOTO 8004
C 37 XBY(OE800H)=192 : XBY(OFO00H)=1 : XBY(OFO00H)=6 : XBY(OFO00H)=128
9758 XBY(OE800H)=64 : RETURN
9760 REM *SUBROUTINE ALTER *
9762 IF I7<>0.AND.C4=0.AND.C5=0 THEN GOTO 10010
9765 IF C4=1 THEN GOTO 9820
9766 IF C5=1 THEN GOTO 11130
9767 IF C0=0.AND.C=0 THEN GOTO 10010
9768 C4=1
9769 C5=0
9770 I7=0
9771 IF C0=1 THEN GOTO 9760
9772 IF C0=2 THEN GOSUB 10110
9776 IF C0=3 THEN GOSUB 10120
9778 IF C0=4 THEN GOSUB 10130
9780 IF C0=5 THEN GOSUB 10140
9782 IF C0=6 THEN GOSUB 10150
9784 IF C0=7 THEN GOSUB 10160
C 36 IF C=1 THEN GOSUB 10170
9788 IF C=2 THEN GOSUB 10180
9790 IF C=3 THEN GOSUB 10190
9792 IF C=4 THEN GOTO 10010
9794 IF C=5 THEN GOTO 10010
C 36 IF C=6 THEN GOTO 10010
9798 IF C=7 THEN GOTO 10010
9800 GOTO 8005
9801 PUSH N5 : POP W0
9803 GOTO 1337
9805 IF W0>=N5.OR.F9=1 THEN GOTO 9975
9807 PUSH N5 : POP W1
9808 C4=0 : I7=0
9809 K8=(W1-W0)/255
9810 R7=(W1-W0)
9811 W7=W0
9812 R0=R7/K8
9814 W2=(W7-W0)/K8
9815 R4=R7
9816 W8=W7
9818 GOTO 1341
9820 IF I7=0 THEN GOTO 9975
9821 GOSUB 11000
9822 IF F9=1 THEN GOTO 10010

```



```

9825 IF Z7=1 THEN GOTO 9801
9826 IF Z7=2 THEN GOTO 9805
9827 ON C0 GOTO 9830,10010,9850,9860,9870,9890,9900,9910
9830 ON C GOTO 10010,9920,9930,9940,9945,9950
9850 IF N5>W1.OR.W0>N5 THEN GOTO 9975
9851 PUSH N5 : POP W2
9852 W8=W2
9853 W2=(W2-W0)/K8
9854 C0=0 : C4=0
9856 GOTO 8003
9860 IF N5>255.OR.N5<1 THEN GOTO 9975
9861 PUSH N5 : POP S0
9863 C0=0 : C4=0
9864 GOTO 8003
9870 PUSH N5
9871 POP R0
9872 R4=ABS(R0)
9873 R0=ABS(R0/K8)
9874 C0=0 : C4=0
9876 GOTO 8003
9890 IF N5>K2.OR.N5<K3 THEN GOTO 9975
9893 C0=0 : C4=0
9894 GOTO 8003
9900 IF N5>T6.OR.N5<T7 THEN GOTO 9975
9901 PUSH N5 : POP T
9903 C0=0 : C4=0
9904 GOTO 8003
9910 IF N5>T4.OR.N5<T5 THEN GOTO 9975
9911 PUSH N5 : POP T0
9913 C0=0 : C4=0
9914 GOTO 8003
9920 IF (N5>Y5).OR.(N5>10).OR.(N5<0) THEN GOTO 9975
9921 PUSH N5 : POP M5
9923 C0=0 : C4=0
9924 Y4=M5
9925 M5=M5/K7
9929 GOTO 8003
9930 IF (N5<Y4).OR.(N5>10).OR.(N5<0) THEN GOTO 9975
9931 PUSH N5 : POP M6
9933 C0=0 : C4=0
9934 Y5=M6
9935 M6=M6/K7
9939 GOTO 8003
9940 PUSH N5
9941 POP M7
9943 C0=0 : C4=0
9944 Y6=ABS(M7)
9945 M7=ABS(M7/K7)
9949 GOTO 8003
9950 GOTO 10010
9960 GOSUB 10200
9962 GOSUB 10220
9964 XBY(OF000H)=84 : XBY(OF000H)=89 : XBY(OF000H)=80 : XBY(OF000H)=69
9966 XBY(OF000H)=61 : XBY(OF000H)=63
9968 C5=1 : C4=0
9970 GOTO 8005
9975 GOSUB 9757
9977 XBY(OF000H)=73 : XBY(OF000H)=76 : XBY(OF000H)=76 : XBY(OF000H)=69
9979 XBY(OF000H)=71 : XBY(OF000H)=65 : XBY(OF000H)=76
9980 XBY(OF000H)=128 : XBY(OF000H)=86 : XBY(OF000H)=65

```

```
9982 XBY(OF000H)=76 : XBY(OF000H)=85 : XBY(OF000H)=69
9984 C=0 : C0=0 : C4=0 : C5=0
9986 IF Z7=0 THEN GOTO 8004
9987 RESTORE : GOTO 1000
10000 XBY(OE000H)=02H
10002 ON 0 GDSUB 7300,7000,7100,7200,7400,7500,7600,7700,
10005 ONTIME TIME+S,10000 : RETI
10010 GOSUB 9757
10012 XBY(OF000H)=87 : XBY(OF000H)=82 : XBY(OF000H)=79 : XBY(OF000H)=78
10014 XBY(OF000H)=71 : XBY(OF000H)=128 : XBY(OF000H)=67
10016 XBY(OF000H)=79 : XBY(OF000H)=77 : XBY(OF000H)=77
10018 XBY(OF000H)=65 : XBY(OF000H)=78 : XBY(OF000H)=68
10020 C4=0
10022 GOTO 8004
10110 GOSUB 10200
10111 GOSUB 10220
10112 XBY(OF000H)=87 : XBY(OF000H)=73 : XBY(OF000H)=78 : XBY(OF000H)=61
10113 XBY(OF000H)=63 : RETURN
10120 GOSUB 10200
10122 GOSUB 10220
10123 XBY(OF000H)=83 : XBY(OF000H)=61 : XBY(OF000H)=63
10124 RETURN
10130 GOSUB 10200
10132 GOSUB 10220
10133 XBY(OF000H)=82 : XBY(OF000H)=70 : XBY(OF000H)=69 : XBY(OF000H)=76
10134 XBY(OF000H)=61 : XBY(OF000H)=63 : RETURN
10140 GOSUB 10200
10141 GOSUB 10220
10142 XBY(OF000H)=75 : XBY(OF000H)=80 : XBY(OF000H)=61 : XBY(OF000H)=63
10143 RETURN
10150 GOSUB 10200
10151 GOSUB 10220
10152 XBY(OF000H)=84 : XBY(OF000H)=68 : XBY(OF000H)=61 : XBY(OF000H)=63
10153 RETURN
10160 GOSUB 10200
10161 GOSUB 10220
10162 XBY(OF000H)=84 : XBY(OF000H)=73 : XBY(OF000H)=61 : XBY(OF000H)=63
10163 RETURN
10170 GOSUB 10200
10172 GOSUB 10220
10174 XBY(OF000H)=89 : XBY(OF000H)=77 : XBY(OF000H)=73 : XBY(OF000H)=78
10175 XBY(OF000H)=61 : XBY(OF000H)=63 : RETURN
10180 GOSUB 10200
10182 GOSUB 10220
10183 XBY(OF000H)=89 : XBY(OF000H)=77 : XBY(OF000H)=65 : XBY(OF000H)=88
10185 XBY(OF000H)=61 : XBY(OF000H)=63 : RETURN
10190 GOSUB 10200
10192 GOSUB 10220
10193 XBY(OF000H)=68 : XBY(OF000H)=89 : XBY(OF000H)=77 : XBY(OF000H)=65
10194 XBY(OF000H)=88 : XBY(OF000H)=61 : XBY(OF000H)=63 : RETURN
10200 GOSUB 10220
10203 FOR I=1 TO 40
10204 XBY(OF000H)=128
10206 NEXT I
10208 RETURN
10220 XBY(OE800H)=192 : XBY(OF000H)=6 : XBY(OF000H)=192 : XBY(OE800H)=64
10222 RETURN
```

```

11000 LET N5=0
11002 LET F8=0
11004 LET F9=0
11006 LET N8=0
11008 LET N9=0
11010 IF I7>0 THEN GOTO 11016
11012 GOSUB 10200
11014 RETURN
11016 IF I7>1 THEN GOTO 11022
11018 IF E(1)=ASC(.).OR.E(1)=45 THEN F9=1
11020 IF F9=1 THEN RETURN
11022 LET I9=1
11024 IF E(1)<>45 THEN GOTO 11032
11026 LET F8=1
11028 LET I9=I9+1
11032 IF E(I9)=ASC(.) THEN GOTO 11044
11034 LET N8=N8+1
11036 GOSUB 11100
11038 LET M(N8)=N6
11040 LET I9=I9+1
11042 IF I9<=I7 THEN GOTO 11032 ELSE GOTO 11056
11044 LET I9=I9+1
11046 LET N9=N9+1
11048 GOSUB 11100
11050 LET L(N9)=N6
11052 LET I9=I9+1
11054 IF I9<=I7 THEN GOTO 11046
11056 IF N8=0 THEN GOTO 11068
11058 LET J=1
11060 FOR I=N8 TO 1 STEP -1
11062 LET N5=N5+M(I)*J
11064 LET J=J*10
11066 NEXT I
11068 IF N9=0 THEN GOTO 11080
11070 LET J=10
11072 FOR I=1 TO N9 STEP 1
11074 N5=N5+L(I)/J
11076 LET J=J*10
11078 NEXT I
11080 IF F8=1 THEN N5=-N5
11082 RETURN
11100 IF E(I9)<48.OR.E(I9)>57 THEN F9=1
11102 IF F9=1 THEN RETURN
11104 IF E(I9)=48 THEN N6=0
11106 IF E(I9)=49 THEN N6=1
11108 IF E(I9)=50 THEN N6=2
11110 IF E(I9)=51 THEN N6=3
11112 IF E(I9)=52 THEN N6=4
11114 IF E(I9)=53 THEN N6=5
11116 IF E(I9)=54 THEN N6=6
11118 IF E(I9)=55 THEN N6=7
11120 IF E(I9)=56 THEN N6=8
11122 IF E(I9)=57 THEN N6=9
11124 RETURN
11130 C0=0 : C=0 : C5=0
11131 IF I7=1.AND.E(1)=80 THEN GOTO 11200
11132 IF I7=1.AND.E(1)=73 THEN GOTO 11220
11134 IF I7=1.AND.E(1)=68 THEN GOTO 11240
11136 IF I7=2.AND.E(1)=80.AND.E(2)=73 THEN GOTO 11260
11138 IF I7=2.AND.E(1)=80.AND.E(2)=68 THEN GOTO 11280
11140 IF I7=2.AND.E(1)=73.AND.E(2)=68 THEN GOTO 11300

```

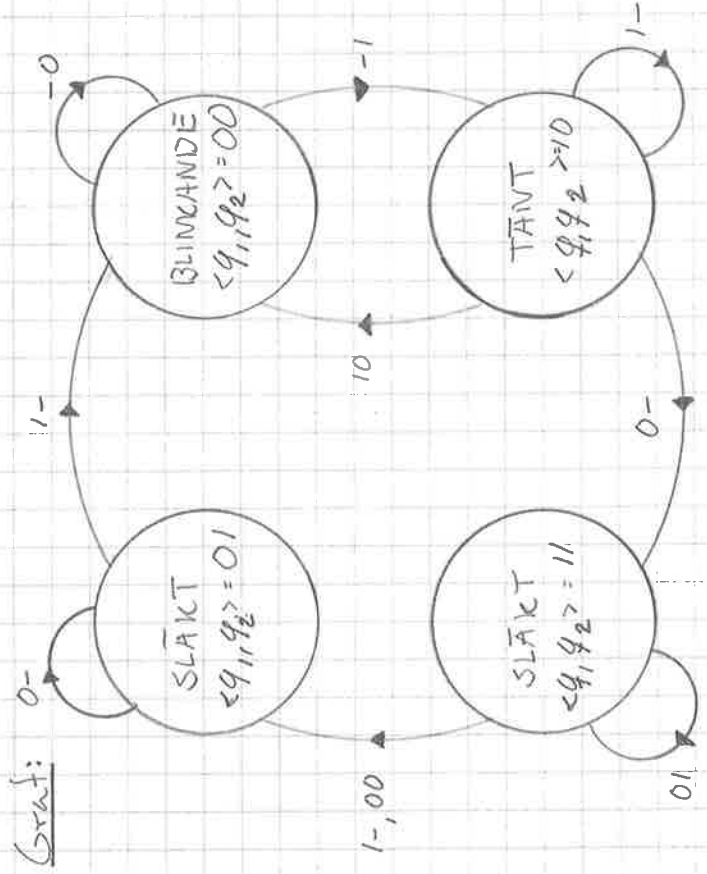
```
11142 IF I7=3.AND.E(1)=80.AND.E(2)=73.AND.E(3)=68 THEN GOTO 11320
11146 GOTO 10010
11150 IF I7=0 THEN GOTO 11160
11152 IF I7=1.AND.E(1)=68 THEN 11170
11155 RESTORE ; GOTO 1000
11160 T3=2 ; Z7=0 ; C5=0
11162 X(2)="REV"
11164 GOTO 8001
11170 T3=1 ; Z7=0 ; C5=0
11172 X(2)="DIR"
11174 GOTO 8001
11200 Q1=1 ; C4=0
11215 GOTO 8003
11220 Q1=2 ; C4=0
11230 GOTO 8003
11240 Q1=3 ; C4=0
11245 PRINT T3
11250 GOTO 8003
11260 Q1=4 ; C4=0
11270 GOTO 8003
11280 Q1=5 ; C4=0
11290 GOTO 8003
11300 Q1=6 ; C4=0
11310 GOTO 8003
11320 Q1=7 ; C4=0
11330 GOTO 8003
11400 LET A9=0
11402 IF A8>=0 THEN GOTO 11410
11404 LET A9=A9+1
11406 LET A(A9)=45
11408 LET A8=-A8
11410 D=INT((A8-INT(A8))*100)
11411 A8=INT(A8)
11412 LET A7=0
11414 LET J=1
11416 IF A8/J<1 THEN GOTO 11420
11417 A7=A7+1
11418 J=J*10
11419 GOTO 11416
11420 IF A7=0 THEN GOTO 11430
11422 FOR I=1 TO A7
11423 J=J/10
11424 A3=INT(A8/J)
11426 GOSUB 11450
11427 A8=A8-A3*J
11428 NEXT I
11429 GOTO 11432
11430 LET A9=A9+1
11431 A(A9)=48
11432 LET A9=A9+1
11433 A(A9)=46
11435 LET A3=INT(D/10)
11436 GOSUB 11450
11440 RETURN
11450 LET A9=A9+1
11452 IF A3=0 THEN A(A9)=48
11453 IF A3=1 THEN A(A9)=49
11454 IF A3=2 THEN A(A9)=50
11455 IF A3=3 THEN A(A9)=51
```

```
11456 IF A3=4 THEN A(A9)=52
11457 IF A3=5 THEN A(A9)=53
11458 IF A3=6 THEN A(A9)=54
11459 IF A3=7 THEN A(A9)=55
11460 IF A3=8 THEN A(A9)=56
11461 IF A3=9 THEN A(A9)=57
11462 RETURN
```



# Bilaga 8.6

## Graf och Tillståndsavkodning för larmenhet.



Insignal:  $\dot{z}_1, \dot{z}_2$   
 $\dot{z}_0$  = överstigit RFEEL.  
 $\dot{z}_1$  = knyttens av Larm.

utsignal  $\langle u_1, u_2 \rangle = \langle q_1, q_2 \rangle$

$\delta$	00	01	11	10
00	00	10	10	00
01	01	01	00	00
11	01	11	01	01
10	11	11	10	00

$q_1, q_2$

$q_1^*$	00	01	11	10
00	0	1	1	0
01	0	0	0	0
11	0	1	0	0
10	1	1	1	0

$\dot{z}_2$

$q_1, q_2$

$q_2^*$	00	01	11	10
00	0	0	0	0
01	1	1	0	0
11	1	1	1	1
10	1	1	1	0

$\dot{z}_2$

$q_1, q_2$

$q_1^* = q_1 \dot{z}_0 \vee q_2 \dot{z}_1 \vee q_2 \dot{z}_2$   
 $q_2^* = q_1 \dot{z}_1 \vee q_2 \dot{z}_0 \vee q_2 \dot{z}_2$

9. MANUAL



## 9.1 KNAPPSATSENS FUNKTION

Knappsatsen och terminalen kan vara anslutna samtidigt. När man väl börjat med att använda en enhet tex knappsatsen, måste man avsluta detta användande innan man kan använda sig av terminalens knappar. för att ytterligare visa detta släcks terminalen ned då knappsatsen används och all visuell kommunikation sker via en 2\*24 raders LCD.

### 9.1.1 KNAPPSATSENS UTSEENDE

7	8	9	P	C
4	5	6	I	☒
1	2	3	D	☒
0	.	-	*	A

### 9.1.2 FUNKTION

Då man önskar använda knappsatsen, skall knappen "\*" tryckas ned. Därefter slå man in koden för den parameter som man önskar se på LC-displayn. När rätt kod har slagits in skall man trycka på "\*" igen. Om man nu har slagit in rätt kod, kommer den sökta parametern att visas längst uppe i det vänstra hörnet på LCD:n.

Då man vill ändra en parameter skall man använda knappen "A" (Alter.) Observera att ändring endast kan ske då man redan har en parameter displayad uppe i vänstra hörnet på LCD'n och det på raden under står "code?". Om man nu trycker på "A", så kommer det en ny utskrift på den undre raden. Denna utskrift består av den aktuella parametern med ett frågotecken efter sig, tex "WIN?". Nu är det fritt fram att slå in ett nytt värde. Slår man in ett otillåtet värde kommer displayen att svara med "illegal value". Har man slagit in ett godtagbart värde, så försvinner utskriften av "WIN=" och "code=" kommer på nytt upp på displayen. Nu skall man inte bli förvånad över att det gamla värdet fortfarande står kvar på den övre raden. Systemet fungerar nämligen som så att det aktuella parametervärdet som ingår i regulatorn är det som visas på display tills dess att man trycker på knappen "☒". Först nu läggs det nya värdet in i regulatorn. Man skall alltså inte trycka på "☒" förrän man ändrat samtliga par. som man vill skall ändras. När "☒" tryckts in övergår datorn till "display mode" på terminalen. Det är nu möjligt att fortsätta par. ändringen via terminalen.

## 9.1.1.3 KNAPPARNAS FUNKTION

- 0 - 9 Olika sifferkombinationer används som koder för olika parametrar. Siffrorna används också vid inslagning av nya parametervärden.
- . - (punkt o minus) Används vid ändring av par.värden
- P - I - D Används vid ändring av regulator typ.
- \* Används för att komma in i "knappsats mode".
- \* Används efter inslagen parameter-kod, för att aktuella parametrar skall visas på LCD'n.
- A (Alter), Används då man vill ändra en parameter.
- ⊞ Används då samtliga par. som man önskar ändra är ändrade. Efter att denna knapp tryckts ned, så återvänder systemet till "display mode" på terminalen och man måste åter igen trycka på "\*" för att komma in i "knappsats mode".
- ✱ Används om man vill "rulla" aktuella par. på LCD.
- C Används för att kvittera larm.


## 9.1.4 EXEMPEL

En knappsets-sekvens kan se ut på följande vis, om man vill gå från en PID-regulator till en PI-regulator samt att man vill ändra börvärdet ( WIN ), från 100 till 150:

```

*      övergång till "knappsets mode"
1
0
*      Nu visas den aktuella parametern överst till
      vänster
A      Du talar om att du vill ändra den visade
      parametern.
P      Regulatortypen ändras till PI.
I
A      Ändringen fullbordas genom att A slås in som
      en bekräftelse på att man ej gjort ett slumpfel.

```

Ändring av övriga parametrar går till på motsvarande sätt.  
Observera att regulatorn reglerar med de "gamla" värdena ända tills operatören trycker på  !!

Ändringen av WIN skulle se ut på följande vis:  
21 \* A 150 A klart.

## 9.1.5 PARAMETER KODER

KOD	FUNKTION	ADRESS
10	TYP	9100
21	WIN	9150
31	S	9200
41	RFEL	9250
51	KP	9300
61	TD	9350
71	TI	9400
81	YMIN	9450
82	YMAX	9500
83	DYMAX	9550
91	Y	9700
95	XIN	9750

Samtliga värden förutom Y och XIN går att ändra m hj a knappats.

#### 9.1.6 FELMEDDELANDEN

1. ILLEGAL VALUE      Skrivs ut då inslaget värde ej är tillåtet.
  2. WRONG COMMAND    Skrivs ut då fel kommando givits.
  3. WRONG CODE        Skrivs ut då fel eller ingen kod slagits in.
- OBS! Om error 1 eller 2 uppkommer skall man trycka på "\*".

## 9.2

## BRUKSANVISNING

## 9.2.1 UPFSTART

1. Anslut nätaggregatet till nätet.
2. Anslut terminal, knappsets eller båda delar. Terminal-kommunikation sker via AMP-uttaget på kortets framsida, ( AMP nr. 3 ), se ritning nr.7.7. Knappsetsen ansluts på rackets baksida till kontakt nr. S6.
3. Se till så att kortet sitter väl itryckt och att anslutnings-sladden till frontpanelen ( AMP nr. 1 ), sitter i.
4. Slå till strömställaren på nätaggregatets högra gavel. Då den lilla LEDen på strömförsörjningskortet lyser kan du vara säker på att spänningen är på. Datorn gör nu en självstart av programmet.
5. Ny uppstart sker då du trycker på den lilla svarta knappen under AMP-kontakt 2 (se ritning nr.7.7). Denna knapp gör reset i processorn och datorn gör en självstart.

## 9.2.2 REGLERING

1. Svaret på de frågor som datorn ställer.
2. Då endast knappsets används, ges det inte lika ingående instruktioner som när man använder terminal. Använd knappsetsen enligt funktionsbeskrivningen kap 9.1.
3. Vid start med endast knappsets, måste först SCALEMIN, SCALEMAX samt DIR/REV initieras. Då felaktiga värden matas in börjar init. om från början. Den enda restriktionen är den att SCALEMIN måste väljas mindre än SCALEMAX. DIR och REV väljes beroende på om processen ger en DIREKT eller REVERSERAD signal.
4. När knappsets används i initieringsskedet, sker en automatisk tilldelning av de olika parametervärdena. Detta skyddar processen från att starta utan begynnelsevärdet. Följande värden initieras automatiskt vid uppstart med hjälp av knappsets:

```

TYP=PID          S=1sek          RFEL=SCALEMAX--SCALEMIN
KP=10           TI=8            TD=0.5
YMIN=0 VOLT     YMAX=10 VOLT   DYMAX=50
WIN=SCALEMIN

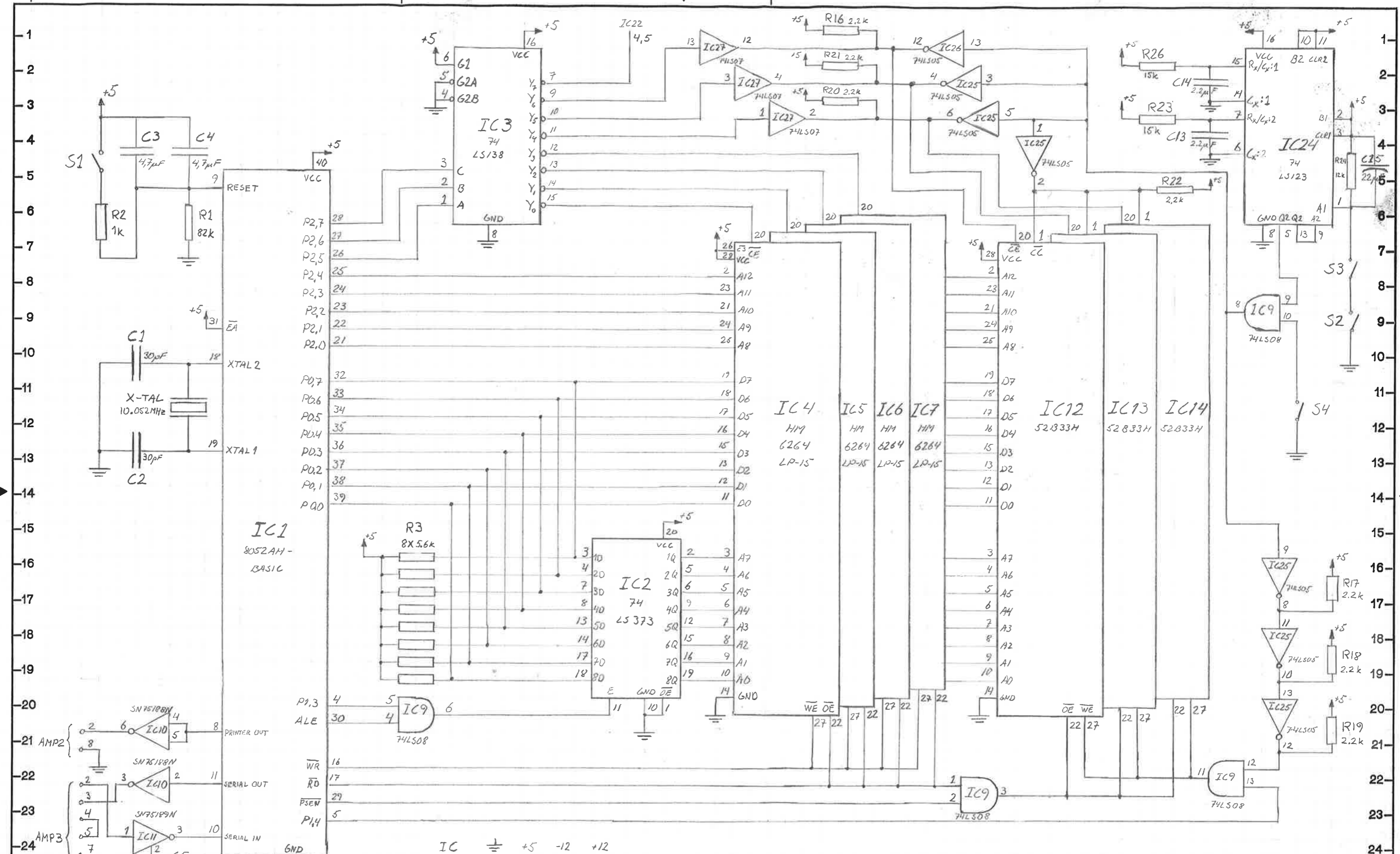
```

Realtidslocken startas då knappen  trycks in och värdena POPas in i regulatorn.

## 9.2.3 SKRIVARE

En skrivare kan anslutas till systemet. Kommunikationen sker via AMP-kontakt nummer 2. Då man vill ta en utskrift av programmet eller om man vill ändra i programmet så att utskrift sker regelbundet på skrivaren, skall man observera att BAUDTALET hos skrivaren och datorn måste stämma överens. Vid korta utskrifter går det utmärkt att använda denna höga hastighet. Självkärlart beror överföringshastigheten på om skrivaren har en tillräckligt stor buffert, men eftersom utskrift sker utan handskakning i vår konstruktion, rekommenderar vi 300 BAUD. Hastigheten ställs helt enkelt genom att man skriver BAUD300 till datorn. Vid inställning av samma mottagningshastighet på skrivaren, se skrivarens instruktionsmanual.





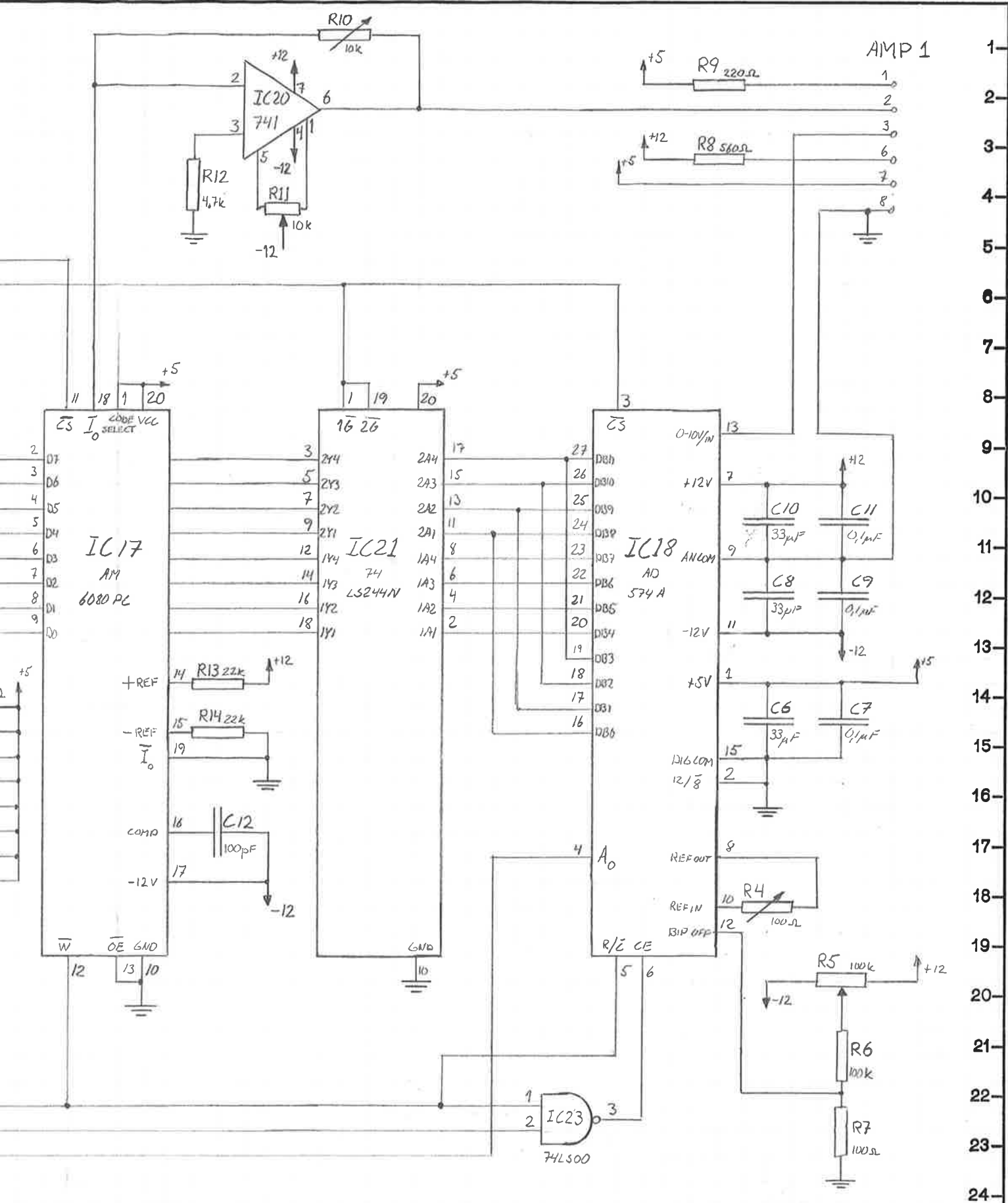
AUTONOM REGLERCENTRAL  
PROCESSOR OCH EXTERNA MINIVÄN

Elektro Sandberg ab

Ritad <i>SV</i>	Datum 100685	Date 7.1	Ritn.nr. 7.1
Konst.Design	Skala	Scale	Fortst.bl. Cont. Blad Sheet

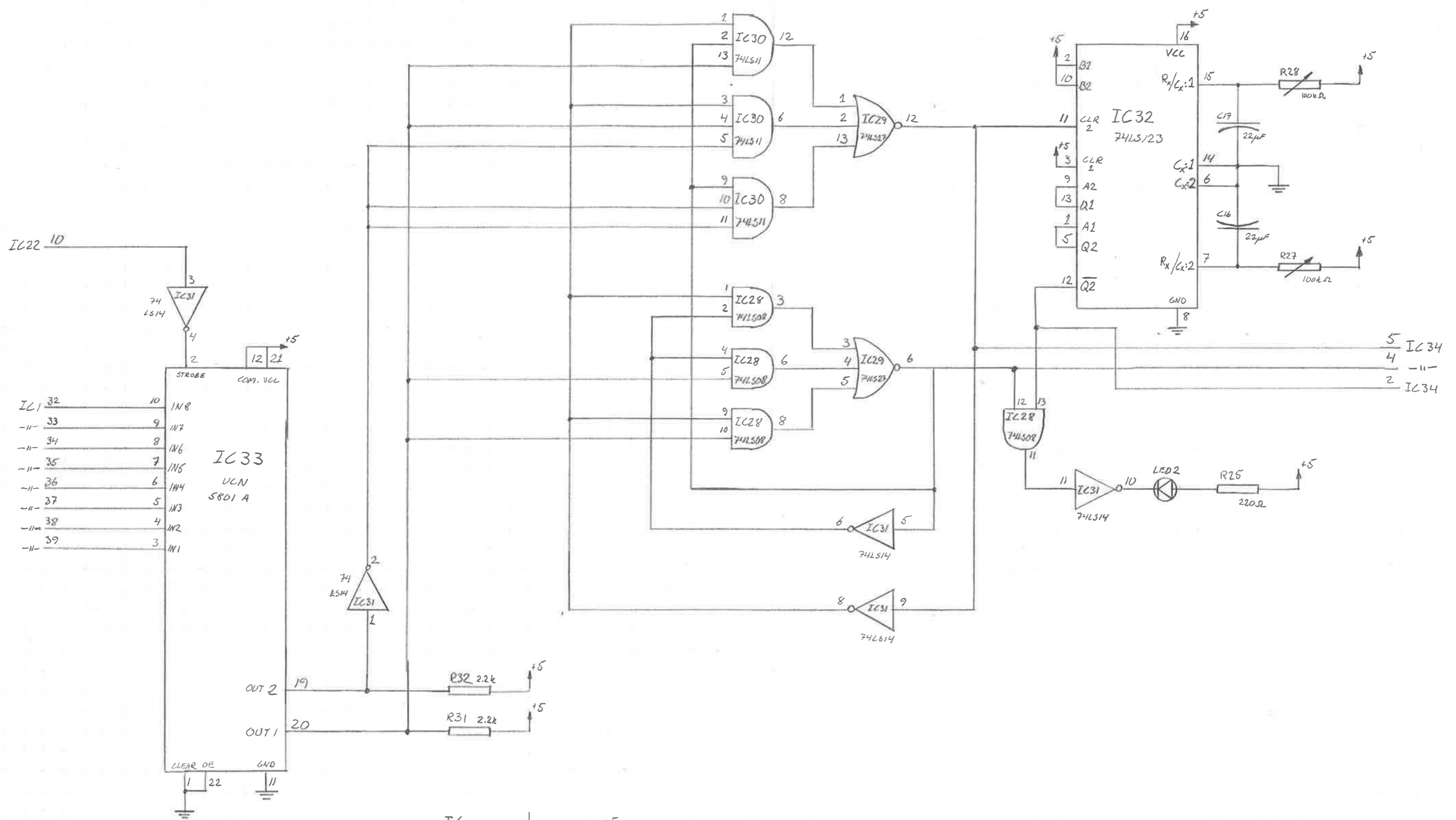
Not No	Andring Revision	Datum Date	Av Appd.

Bl. 101



AUTONOM REGLER CENTRAL  
I/O ENHISTER

<b>Elektro Sandberg ab</b>		Ritid Drawn	Datum Date	Ritn.nr.	Drawing.no.
		SV	100685	7.2	
Konst.Design	Skala Scale	Forts.bl. Cont.	Blad	Sheet	



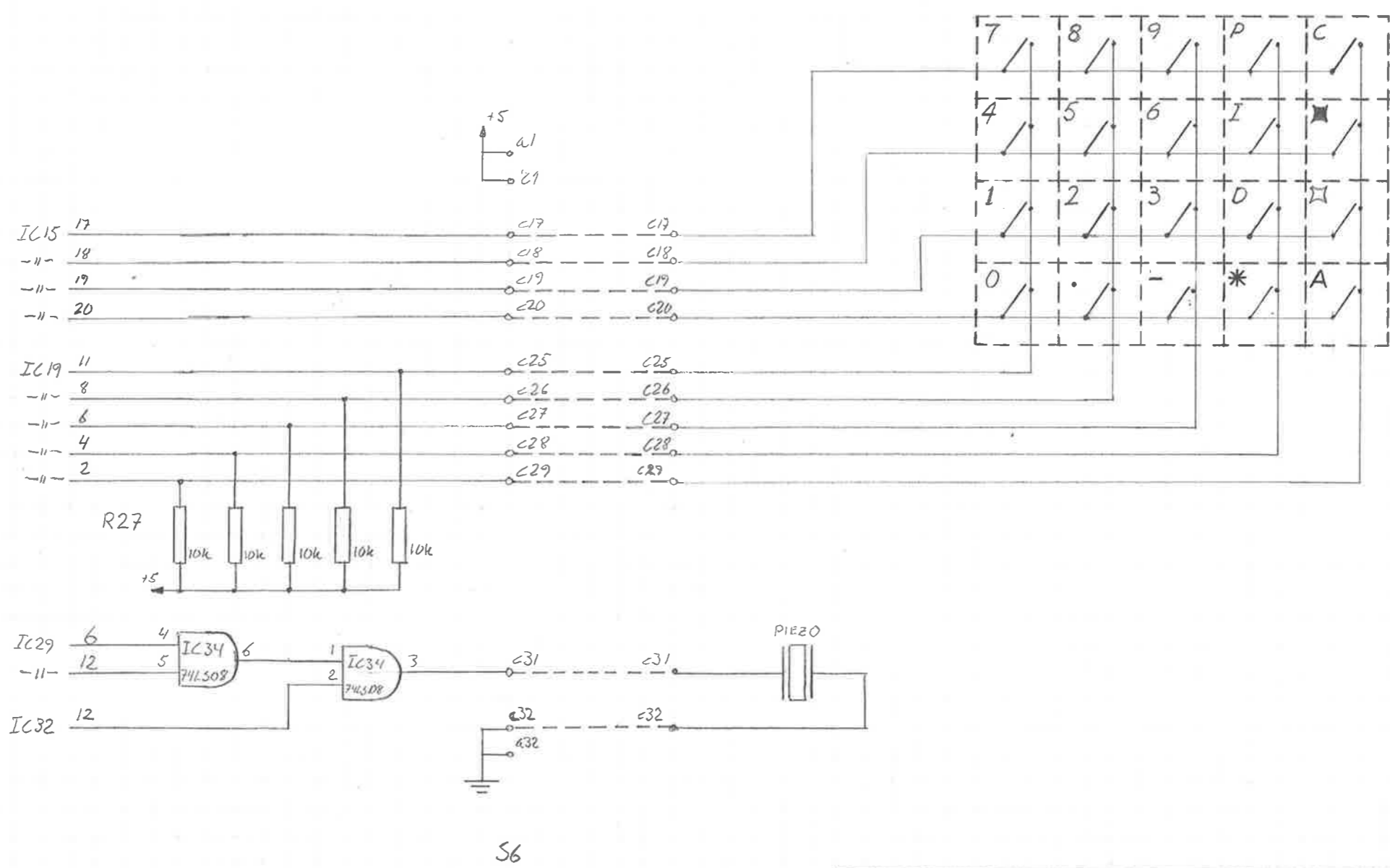
IC	⊥	+5
28	7	14
29	7	14
30	7	14
31	7	14

AUTONOM REGLERCENTRAL LARMENHJET		Elektro Sandberg ab					
		Ritad SV	Datum 100685	Date 7.3	Ritn.nr. Drawing.no.		
Not No	Ändring Revision	Datum Date	Av Appd.	Konst.Design	Skala Scale	Forts.bl. Cont.	Blad Sheet

BL 101

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24

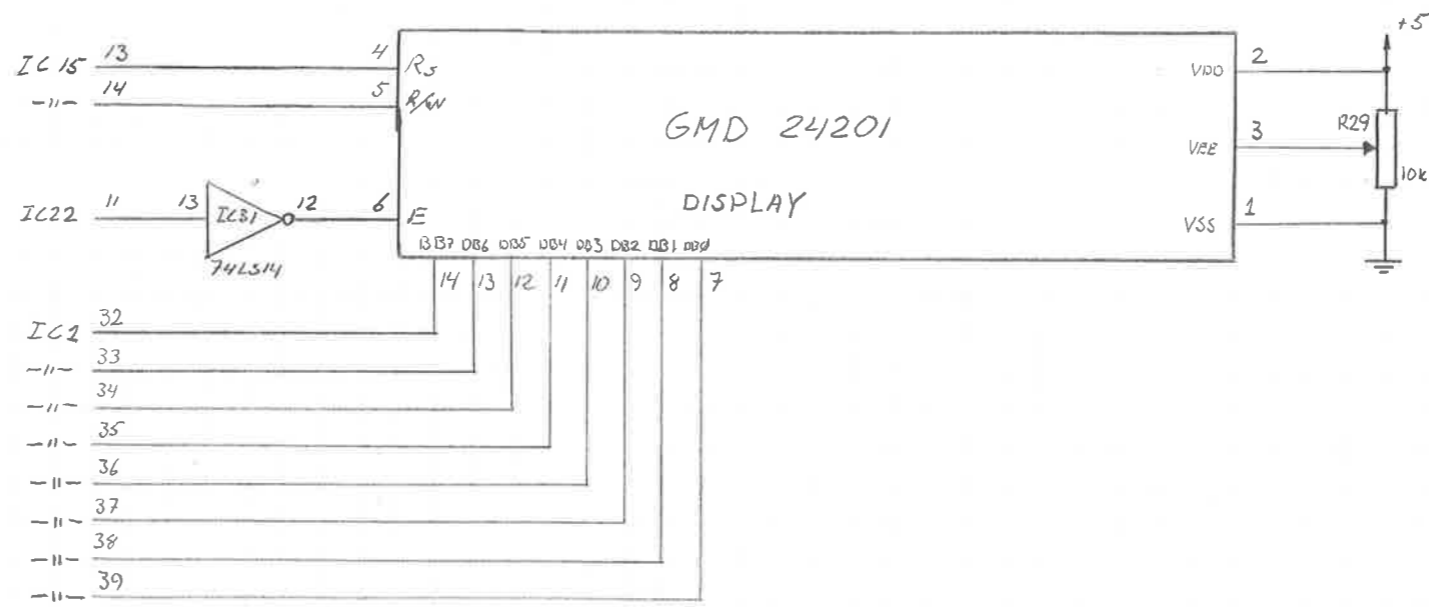


IC    ⊥    +5  
34    7    14

AUTONOM REGLERCENTRAL		<b>Elektro Sandberg ab</b>			
KNAPPSATS OCH LARMENHET					
Ritad SV	Datum 100685	Date 7.4	Ritn.nr.	Drawing.no.	
Konst.Design	Skala	Scale	Fortsb.l. Cont.	Blad	Sheet

Not No	Andring Revision	Datum Date	Av Appd.

Bl. 101

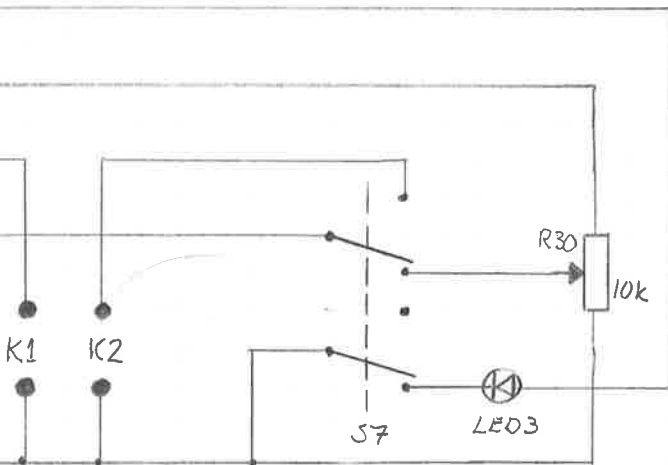
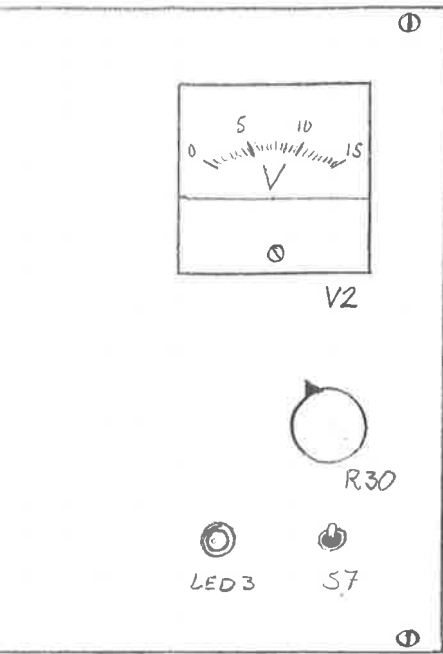


IC  $\perp$  +5  
 31 7 14

Not No	Andring Revision	Datum Date	Av Appd.

AUTONOM REGLERCENTRAL		Elektro Sandberg ab			
LCD		Ritad Drawn	Datum Date	Ritn.nr.	Drawing.no.
		SU	100685	7.5	
Konst.Design	Skala Scale	Forts.bl. Cont.	Blad	Sheet	

Bl. 101



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24

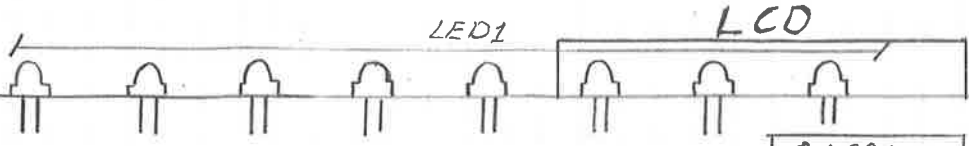
AUTONOM RIEGLER CENTRAL  
FRONT PANEL

**Elektro Sandberg ab**

Ritad	Drawn	Datum	Date	Ritn.nr.	Drawing.no.
	SV		100685	7.6	
Konst.	Design	Skala	Scale	Forts.bl. Cont.	Blad Sheet

Ritad	Drawn	Datum	Date	Ritn.nr.	Drawing.no.
	SV		100685	7.6	
Konst.	Design	Skala	Scale	Forts.bl. Cont.	Blad Sheet

KOMMUNIKATION  
P 3



5801A

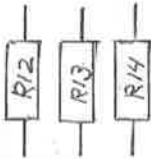
AM 6080 PC  
IC 17

74LS 244 N  
IC 19

8x 220Ω  
R15

74LS14 N  
IC 16

AD 574 A  
IC 18



741  
IC 20

74LS138  
IC 22

R8  
R9

10kΩ R10

10kΩ R11

8x 10kΩ  
R27

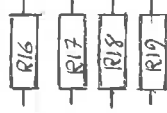
74LS 244 N  
IC 21

74LS00  
IC 23

AMP4

S2

FRONT PANEL



74LS05  
IC 25



74LS123  
IC 24

C16 C17

74LS05  
IC 26

74LS07  
IC 27

C15

74LS14  
IC 31

74LS09  
IC 28

74LS27  
IC 29

74LS11  
IC 30

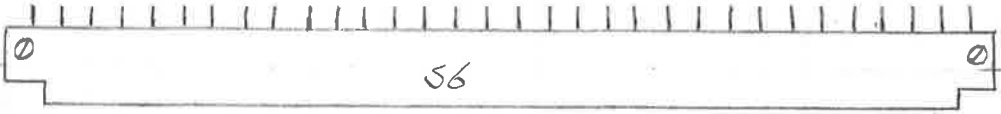
74LS123  
IC 32

100kΩ R27 100kΩ R28

R31  
R32

74LS08  
IC 34

5801A  
33



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24

AUTONOM REGLERCENTRAL		Elektro Sandberg ab			
KRETS OCH KOMPONENTPLACERING					
Ritad	Drawn	Datum	Date	Ritn.nr.	Drawing.no.
	SU		100685	7.7	
Konst.	Design	Skala	Scale	Forts.bl.	Cont. Blad
					Sheet

Betn.	Typ.	Betn.	Typ.	Bctn.	Typ.	Bctn.	Typ.
C1	30pF	R1	82kΩ	R18	2,2kΩ	S1	Switch, reset
C2	30pF	R2	1kΩ	R19	2,2kΩ	S2	Bygling, erase
C3	4,7μF	R3	8x5,6kΩ	R20	2,2kΩ	S3	Switch, erase
C4	4,7μF	R4	100Ω trimpot.	R21	2,2kΩ	S4	Omkopplare, erase
C5	330pF	R5	100kΩ trimpot.	R22	2kΩ	S5	Plattkabelkontakt, spänningsmatning
C6	33μF	R6	100kΩ	R23	15kΩ	S6	Plattkabelkontakt, knappsetsenhet
C7	0,1μF	R7	100Ω	R24	12kΩ	S7	Omkopplare, insignal
C8	33μF	R8	560Ω	R25	220Ω	X-TAL	Kristall, 10.05920 MHz
C9	0,1μF	R9	220Ω	R26	15kΩ	PIEZO	Piezokristall, larm
C10	33μF	R10	10kΩ trimpot.	R27	100kΩ trimpot.	AMP1	Kontakt, Frontpanel
C11	0,1μF	R11	10kΩ trimpot.	R28	100kΩ trimpot.	AMP2	Kontakt, skrivare
C12	100pF	R12	4,7kΩ	R29	10kΩ trimpot.	AMP3	Kontakt, Terminal
C13	2,2μF	R13	22kΩ	R30	10kΩ vridpot.	LED1	8x Ljusdioder, utport IC15
C14	2,2μF	R14	22kΩ	R31	2,2kΩ	LED2	Ljusdiod, larm
C15	22μF	R15	8x 220Ω	R32	2,2kΩ	LED3	Ljusdiod, insignal
C16	22μF	R16	2,2kΩ			K1	Kontakt, utsignal
C17	22μF	R17	2,2kΩ			K2	Kontakt, insignal
						V1	Voltmeter 0-15V, utsignal
						V2	Voltmeter 0-15V, insignal

Not No	Andring Revision	Datum Date	Av Appd.

AUTONOM REGLER CENTRAL		Elektro Sandberg ab			
KOMPONENTLISTA		Ritad Drawn	Datum Date	Ritn.nr.	Drawing.no.
		SV	100685	BILAGA	8.5
Konst.Design	Skala	Scale	Forts.bl. Cont.	Blad	Sheet

Bl. 101