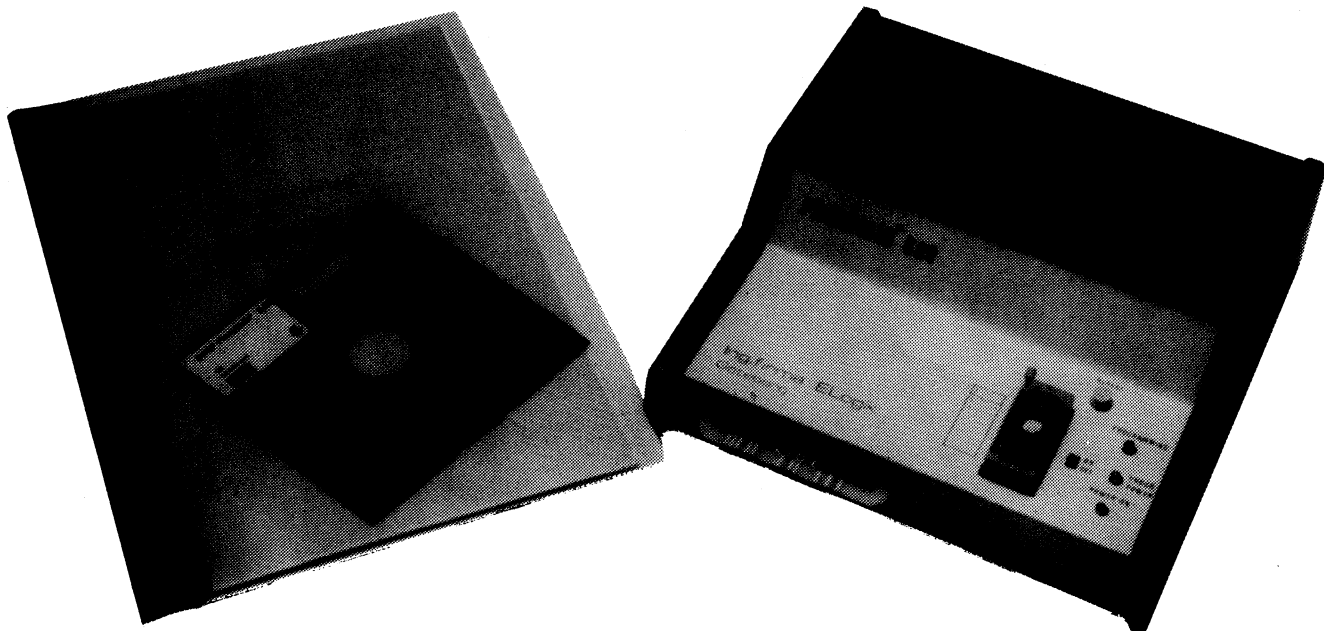


BRUKSANVISNING

PROMMIS 880



PROMMIS[®] 880 är en programmerare för EPROM, som tillsammans med en dator ABC800 eller ABC80 utgör en komplett programmeringsutrustning.

De flesta vanliga typerna av EPROM-minnen kan programmeras av PROMMIS 880. Stor vikt har vid utvecklingen lagts vid att noga följa tillverkarnas specifikationer för spänningsnivåer, pulstider m.m. för de olika typerna.

Med utrustningen levereras programvara på flexskiva för aktuell ABC-dator. Bland programmen ingår rutiner för datalagring, editering, kontroller m.fl. viktiga bifunktioner.

COPYRIGHT.

Informationen i denna bruksanvisning och i de beskrivna programmen är egendom tillhörande Ing.firma ELogik. Den får inte reproduceras, helt eller delvis, utan skriftligt tillstånd av Ing.firma ELogik.

Ing.firma ELogik har copyright på bruksanvisningen och programmen för PROMMIS 880.

FÖRBEHÅLL.

Ing.firma ELogik förbehåller sig rätten att utan varsel göra ändringar i utrustning och programvara.

För fel i denna bruksanvisning, fel i programmerade EPROM-komponenter och följdfel av dessa i andra system ansvaras ej.

Ing.firma ELogik
Thorburnsgatan 3 C
412 64 Göteborg
Tel. 031-83 10 00, 40 96 00

INNEHÅLL.

1. HANDHAVANDE	3
2. ANVÄNDNING AV PROMMIS 880	
2.1. Kontroll av EPROM-radering	4
2.2. Kopiering av EPROM	4
2.3. Kopiering av EPROM, med ändring	5
2.4. Fillagring av PROM-data	5
2.5. Kontroll av EPROM	6
3. TILLÄMPNINGAR MED ABC80	
3.1. Inläggning i EPROM av maskinspråksrutiner	6
3.2. Inläggning i EPROM av BASIC-program	7
3.3. Autostartfunktion för prommade BASIC-program	8
4. EPROM-TYPER	9
5. RADERING AV EPROM-MINNEN	11
6. SPECIFIKATIONER PROMMIS 880	12
7. PROGRAMVARA PROMMIS 880	13
7.1. Filhantering	14
7.2. Editering i minne	15
7.3. Minnestransaktioner	16
7.4. PROM-transaktioner	16
7.5. EPROM-programmering	17
8. ELEKTRONIK	17
9. MINNESKARTA	18
10. TERMFÖRKLARINGAR	19

Bilagor:

1. Beskrivning av hexformat (ITH-format)
2. Listning ROMDRIV
3. Listning ASTART
4. Kretsschema PROMMIS 880
5. Komponentplacering PROMMIS 880

1. HANDHAVANDE PROMMIS 880.

Uppkoppling.

Anslut PROMMIS 880's 9-poliga flatkabelkontakt till datorns seriekontakt. Vid användning med ABC800 kan vilken som helst av seriekanalerna A eller B användas.

Den 5-poliga DIN-kontakten sätts in i datorns kassettkontakt.

Den lilla telefonproppen skall för ABC80 anslutas till datorns motorstyrningskontakt eller för ABC800 till hylsjacken baktill på den 5-poliga DIN-kontakten.

Anslut nätsladden och tryck in nätströmbrytaren, varefter den gröna lysdioden, märkt POWER ON, skall lysa. Det är säkrast att inte ha någon EPROM i sockeln innan programstart skett.

Typval.

Typkortet anger vilken typ av EPROM-minne som för tillfället är vald. Vald typ framgår, förutom av märkningen på den utstickande delen av kortet, bl.a. av texten i huvudmenyn.

Typkortet bör normalt vara instuckat i PROMMIS 880's panel. Vid byte av EPROM-typ förvissar man sig först om att PROM-sockeln är tom, varefter typkortet dras ut, vändes om till nytt läge, och stoppas in på plats igen. Ge RETURN på tangentbordet för att huvudmenyn skall uppfatta och visa den nya typen.

Den (ej nödvändiga) operationen T (typ av EPROM) i huvudmenyn ger viss information om den valda EPROM-typen.

Risker med fel typval.

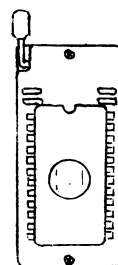
Den via typkortet valda EPROM-typen måste överensstämma med typen på den krets som sedan används, vid risk annars för fel på både kretsen och PROMMIS 880. Speciellt gäller detta om programmering genomförs.

Hållaren.

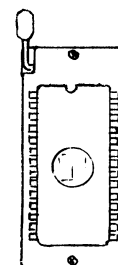
Nollkraftsockelns hävarm skall peka uppåt då kretsen är frigjord från låsningen och kan plockas i eller ur.

Kretsplacering i hållaren.

EPROM-kretsen skall alltid placeras helt "i botten" på sockeln och vänd med ben 1 "uppåt". Se också skissen på PROMMIS 880's panel.



24-bens
EPROM



28-bens
EPROM

Programstart.

Sätt flexskivan med programmen i valfri flexskiveenhet. Ge på tangentbordet kommandot "RUN PROMMIS", varvid programladdning sker, avslutad med presentation på bildskärmen av huvudmenyn.

Versionen för ABC800 kan också startas med tryckning på RESET baktill på tangentbordsdatorn. Svara A eller B på frågan om vilken seriekanal (seriekontakt) som används för PROMMIS 880.

2. ANVÄNDNING AV PROMMIS 880.

2.1. Kontroll av EPROM-radering.

Då man ger kommandot R (raderkontroll) i huvudmenyn går programmet igenom alla adresser i den aktuella EPROM-typen och anger på skärmen adress och data för alla oraderade celler. Tryckning på mellanslagstangenten ger visning av nästa oraderade cell. Knapp RETURN ger återgång till huvudmenyn.

Vid kontroll av flera EPROM i följd behöver man bara trycka på knapp REPEAT efter att ha bytt EPROM i sockeln.

Raderingskontrollen tar ca. 1 sekund att utföra för typ 2708 och i proportion därtill för de andra minnestyperna, beroende på storlek.

En raderingskontroll utförs också, helt automatiskt, omedelbart innan programmering, då dock utan utskrift på skärmen av minnesdata. Om kretsen inte är helt raderad i detta läge meddelas detta på skärmen och man får avgöra om man skall avbryta eller ändå genomföra EPROM-programmeringen.

2.2. Kopiering av EPROM.

En vanlig uppgift för PROMMIS 880 är att skapa en identisk kopia av ett existerande EPROM-minne, vilken förlaga i detta sammanhang brukar kallas för master. Denna slags kopiering kan ske oberoende av innehållet i minnet och av i vilken slags utrustning som det skall användas.

Ibland finns inte mastern i form av en EPROM utan är i stället en pinkompatibel ROM. Även en sådan kan läsas in av PROMMIS 880. Som exempel på ett sådant kompatibelt ROM-minne kan nämnas de TMS4732 (kompatibla med EPROM 2532), vilka kretsar används i de flesta versionerna av ABC80 för att lagra BASIC-tolken. Kompatibiliteten innebär att man kan byta ut originalkretsarna (4732) i ABC80 mot 2532-or, vilka programmerats med t.ex. en modifierad BASIC-tolk.

Man utför kopieringen genom att, med kommandot L i huvudmenyn, först läsa in mastern till editeringsbuffern (=mellanlagringsarea i datorns minne). Efter insättning av en raderad EPROM i programmeringssockeln ger man så kommandot P för programmering. Då kretsen är programmerad korrekt fås en ljudsignal från datorn och man behöver bara trycka på knapp REPEAT för att efter kretsbyte i hållaren göra ytterligare en kopia. - Om standardantagandena (defaults) gäller, alltså om standard editeringsbuffer används samt om hela den använda EPROM-kretsen skall programmeras, behöver man bara svara med RETURN på programmets alla frågor om parametrar för operationerna L och P.

2.3. Kopiering av EPROM, med ändring.

Ofta vill man kopiera EPROM på sätt som beskrivits ovan men samtidigt införa mindre ändringar av innehållet (data) i minneskretsen. Efter att ha läst in masterkretsen ger man då kommandot E för editering i programmets huvudmeny. Editeringsprogrammet visar minnesinnehållet, en skärmsida i taget, i valfritt hexadecimal, decimal- eller textmod samtidigt som det möjliggör att man gör ändringar. Alla kommandon till denna minneseditor finns förklarade på skärmens nedersta rader.

Efter editeringen gör man programmering på samma sätt som ovan.

2.4. Fillagring av PROM-data.

En EPROM kan lagras för framtiden antingen i form av en fysisk masterkrets, vilket kan bli dyrt och besvärligt i längden, eller i form av datafiler på flexskivor (eller kassetband). Dessa datafiler kan finnas i olika format (sätt att representera data).

ITH-format.

Detta s.k. hexformat är universellt och kan bl.a. användas för att lagra maskinspråksrutiner eftersom alla möjliga bytevärden är tillåtna. En liten nackdel med formatet är att lagringen är ganska skrymmande (ca. 2,4 byte i filen för varje databyte).

Många assemblerer kan fås att ge sitt utresultat (maskinkoden) i form av en ITH-fil.

För definition av ITH-formatet, se bilaga 1.

ISO-format.

För data i form av klartext, som t.ex. BAS-filer från ABC80's eller ABC800's BASIC, liksom andra filer som enbart innehåller skrivbara ISO-tecken (ASCII-tecken), finns möjlighet att använda ISO-formatet. (Angående ISO och ASCII, se termförklaringarna). Filerna i detta format ges ofta namntillägget .BAS eller .TXT. De kan alstras, förutom med BASIC-kommandot LIST, med minneseditorn arbetande i textmod, varefter minnesinnehållet dumpas (skrivs ut) i ISO-format med kommando F i huvudmenyn och sedan kommando S.

POKE-satser.

Ett tredje filformat är det där data representeras i form av POKE-satser. Dessa utgör i sig ett giltigt BASIC-program, vilket då det exekveras ("körs") gör laddning av data till minnet. Fördelen med detta format är att det, till skillnad från ITH-formatet, är direkt hanteringsbart av BASIC-tolken. Observera att något separat program inte behövs för återladdning i minnet av data i detta format; laddning sker ju genom att man exekverar programmet. - Även dessa filer har vanligen namntillägget .BAS.

ABS-format.

Systemprogram (med namntillägget .SYS eller .ABS) är i så kallat ABS-format, vilket är kompaktare än ITH-formatet men inte kan listas direkt på t.ex. en printer. Vissa assemblerorer ger sitt resultat i form av en ABS-fil. PROMMIS 880's program kan ladda (läsa in) en ABS-fil.

Kommandot F (filhantering) i huvudmenyn ger en undermeny för val av filinläsnings- eller fillagringsprogram. Då alla dessa underprogram inte ryms i minnet samtidigt med huvudprogrammet, sker övergång med CHAIN till dem vid behov. Detta märks i form några sekunders väntan på överföringen från flexskiveminnet.

2.5. Kontroll av EPROM.

En ofta förekommande operation är att kontrollera innehållet i en PROM-kapsel gentemot minnet. Angående specialfallet raderingskontroll, se avsnitt 2.1.

Varje programmering åtföljs automatiskt av en kontroll (verifikation), varvid eventuella avvikelser skrivs ut på skärmen. Denna automatiska kontroll är tillräcklig i normala fall.

Vill man så kan man när som helst i huvudmenyn kommandera J för jämförelse. Här kan man jämföra en minnesarea mot EPROM-kapseln eller mot en annan minnesarea. Avvikelser skrivs ut på skärmen, knapptryckning på mellanslag ger nästa utskriftsrad medan knapp RETURN ger återgång till huvudmenyn.

3. TILLÄMPNINGAR MED ABC80.3.1. Inläggning i EPROM av maskinspråksrutiner.

Ibland har man behov av att använda maskinspråksrutiner, alltså sådana skrivna i assemblerkod. Orsaken kan vara att man önskar få större snabbhet i programmet (assemblerkod exekverar ofta 10-30 gånger snabbare än motsvarande BASIC-kod) eller ett behov av att spara minnesutrymme. Om programmet ifråga kommer att behövas ofta eller med snabbast möjliga inladdningstid är det då lämpligt att placera maskinspråskoden i EPROM.

Placering i minnet.

Vid användning sätts EPROM-kapslarna på ett PROM-kort vilket ansluts till datorbussen. PROM-kortet måste adresseras ("jumpras") till lämplig, ledig adress inom den adressrymd som gäller för den använda datorn (ABC80). Se denna bruksanvisnings avsnitt med minneskortor, vilka är översiktliga framställningar över en dators adressrymd.

För ABC80 finns i huvudsak tre alternativ för adressering av ett extra PROM-kort. Om datorn bara har 16 kbyte RAM-minne så finns det en stor ledig area mellan 32768 och 49151 (8000H-BFFFH). Den bör dock inte användas om programmet i fråga också skall kunna användas med en minnesutbyggd ABC80.

Om programmet som skall placeras i PROM är ett drivprogram för en in/utenhet, t.ex. en skrivare, placeras det lämpligen i arean 7000H-7BFFFH som är avsedd för detta ändamål. Skrivarrutiner t. ex. bör i första hand inrymmas mellan 7800H och 7BFFFH. Observera att just detta speciella utrymme på 1 kbyte är mycket väl tilltaget; här kan man ofta utan svårighet få plats med flera drivrutiner förutom den vanliga för enhet PR:.

Det tredje alternativet är adressområdet 4000H-5FFFH vilket alltid finns till hands i ABC80 (där dock t.ex. TKN 80, 80-teckens-tillsats för skärmen, gör anspråk på en del av minnet).

Anrop från BASIC.

En maskinspråksrutin anropas, vare sig den är placerad i RAM eller PROM, från BASIC med satsen eller kommandot CALL. Se närmare härom i datorns BASIC-bruksanvisning, där det också anges hur man överför parametrar. Återgång till BASIC sker med maskinspråksinstruktionen RET. - Detta gäller också för ABC800.

3.2. Inläggning i EPROM av BASIC-program.

Fördelen med att ha ett program placerat i PROM är att man vid t.ex. en maskin- eller processtyrtilämpning inte behöver använda flexskive- eller kassetbandsminnen, vilka kan ha svårt att tåla damm eller liknande i miljön. Laddningen av program går också fortare. Om man dessutom inför en autostartfunktion (se nästa avsnitt) kan datorn och hela processen fås att automatiskt starta om igen efter ett kraftavbrott.

Vår metod att promma ett program bygger på ett enhetsdrivprogram (device driver) för den nyskapade enheten ROM:. Precis som med de verkliga in/utenheterna CAS: eller DRO: (flexskiva 0) kan vi nu använda enhet ROM: i kommandona LOAD eller RUN för att ladda resp. köra programmet. BASIC-programmet placeras i ROM i textform (BAS-fil) och exekveras med kommandot RUN ROM:. Hjälpprogrammet ROMDEVIC utför inläggning av den nödvändiga drivrutinen.

- A. Med det önskade BASIC-programmet i minnes, kommandera först "REN 1,1" för omnumrering vilket kortar ned programmet. Skriv "LIST filnamn" för att skapa en BAS-fil på flexskiva.
- B. Anslut PROMMIS 880 och starta med RUN PROMMIS som vanligt.
- C. Välj filhantering, F, i huvudmenyn och där A, inläsning av ISO-fil. Läs in BAS-filen till editeringsbuffern. Inläsningsprogrammet kan fås att slopa alla REM-satser och onödiga mellanslag vid överföringen, vilket om det väljs kommer att spara minnesutrymme i ROM.

- D. Efter återgång till huvudmeny, välj E, editering. Gå in i editeringsbuffern till slutet på BASIC-texten och inför efter sista ODH (CR=vagnretur) slutkoden 03 som allra sista tecken. Notera adressen. Återgå till huvudmenyn.
- E. Sätt en EPROM i hållaren och ge kommandot P, programmering. Inkludera slutkoden i den text som programmeras.
- F. Placera EPROM-kapseln på ett PROM-kort, vilket ansluts till ABC80-bussen.
- G. Gör RESET på ABC80. Skriv "RUN ROMDEVIC". Svara på programmetets fråga om ROM-adressen, varefter det prommade BASIC-programmet laddas in och kommer igång.

Efter att ha kört ROMDEVIC en gång räcker det att skriva "RUN ROM:" för att åter hämta in och köra det prommade programmet.

Man kan ha flera olika BASIC-program i PROM samtidigt. Användaren måste själv hålla reda på adresserna till dem. Vid växling mellan programmen gör "RUN ROMDEVIC".

Bland bilagorna finns en listning över drivprogrammet för enhet ROM:, vilket drivprogram i form av POKE-satser ingår i programmet ROMDEVIC.

3.3. Autostartfunktion för prommade BASIC-program.

Angående fördelarna med PROM-placering av program och autostart, se föregående avsnitt.

Här skall beskrivas hur automatstartfunktionen införs för ett prommat BASIC-program, placerat i den fastlagda ROM-adressen 4100H. Den använda automatstartrutinen börjar i adress 4000H. Flexskiveenhet förutsätts finnas och dess funktion störs ej. Efter automatstarten kan man t.ex. stoppa programmet med Control C och därefter använda flexskiveminnet på vanligt sätt.

ABC80's BASIC-tolk undersöker efter RESET eller kraftpåslag om det finns en hoppinstruktion (JP xxyy = C3 yy xx H) i adress 604BH, tydande på existensen av en flexskiveenhet. För automatstartens skull gör man ett ingrepp här för att lura BASIC-tolken att ta en omväg. Man ersätter xxyy (två byte) med 4000H i adress 604CH. Från adress 604BH skall alltså stå: C3 00 40. Den gamla adressen xxyy inför i automatstartrutinen i adress 4001H-4002H.

Förfarandet i detalj:

- A. Placera BASIC-programtexten i en BAS-fil. Läs in filen till PROMMIS 880's editeringsbuffer, lägg till slutkoden och programmera en EPROM. Allt detta enligt punkterna A-E i föregående avsnitt. Dock skall BASIC-texten placeras i EPROM med början i adress 100H för att startadressen på PROM-kortet skall bli 4100H.

- B. Läs med filhanteringsprogrammet, F i huvudmenyn och därefter I, in filen ASTART.ITH. (Se bilagorna för en listning av ASTART, vilket är autostartrutinen med ingående enhetsdrivprogram för ROM:).
- C. Notera med hjälp av minneseditorn innehållet i adress 604CH-604DH. Låt oss kalla dessa två byte xxyy. Placera xxyy som andra och tredje databyte i editeringsbuffern där innehållet från ASTART.ITH ju nu finns. Den första byten skall vara oförändrat CDH.
- D. Programmera den EPROM-kapsel som redan innehåller början av BASIC-texten med data från ASTART.ITH, totalt 157 byte till PROM-adress 0.
- E. Nu återstår att modifiera ABC80's DOS-ROM i adress 604CH-604DH. Där skall stå adressen 4000H, ingående i en hoppinstruktion. Flytta, med kommando Y i huvudmenyn och sedan L, hela DOS, adress 6000H-6FFFH till editeringsbuffern. Ändra i cellerna motsvarande adress 604CH-604DH till nya data 4000H. Med rätt ordning bytevis skall det alltså stå, med början i ursprunglig adress 604BH: C3 00 40 (hexadecimalt).
- F. Programmera en ny EPROM, av samma eller pinkompatibel typ som den vari DOS är inrymd, med dessa modifierade data. Det är endast två byte som har ändrats, i adress 604CH-604DH, varför det är en enda ROM-kapsel som berörs av DOS-modifieringen.
- G. Byt ut DOS-ROM-kapseln och installera EPROM med autostartrutin och BASIC-text i ett PROM-kort. Gör RESET. På skärmen kommer att stå "Automatstart från ROM:". BASIC-programmet går igång.

Efter ändringar i ASTART-programmet kan BASIC-texten och autostartrutinen placeras på andra adresser än 4100H resp. 4000H. Om flexskiveenhet saknas kan det t.ex. vara lämpligt att placera allt i området där DOS normalt finns, 6000H-6FFFH. Det gäller dock att det alltid måste finnas en hoppinstruktion (börjar med C3H) i adress 604BH.

4. EPROM-TYPER.

EPROM-minnet är i stort sett jämgammalt med mikroprocessorn och har fått sin stora användning tillsammans med denna datorkomponent. Den första EPROM-typen var 1702 som utvecklades av Intel. Efter sin introduktion i början av 1970-talet blev den snabbt mycket populär, trots att den rymde bara 256 byte.

1702 var omständigt att programmera då den krävde flera höga programmeringsspanningar och mycket pulsningar av kontroll- och adresssignalerna. Den var också långsam enligt dagens standard, varianter med accesstider på bortåt 2 mikrosekunder förekom. Den kan därför idag betraktas som en rent historisk typ.

Nästa huvudtyp av EPROM, 2708, var avsevärt förbättrad i programmeringsavseende och dessutom fyra gånger större, den innehåller 1 Kbyte. 2708 är en etablerad typ som finns införd i många utrustningar, varför typen ingår bland dem som PROMMIS 880 kan programmera. Då den kräver tre olika matningsspänningar, nämligen +12 V, +5 V och -5 V och genom att programmeringen fortfarande är lite omständigt genom att man alltid måste programmera hela minnet kan typen nu betraktas som föråldrad. Ett säkert tecken på detta är priset; den kostar nu mer än den bättre och dubbelt så stora 2716-typen!

Dagens industristandard är 2 Kbytetypen 2716. Med den infördes enkelspänningsmatning (bara +5 V) och möjligheten att programmera enstaka celler. Här kommer också standarden med 50 ms programmeringstid per byte, vilken standard gäller också för senare typer. En hel 2716-kapsel programmeras alltså på $2048 * 50 \text{ ms} = 102,4$ sekunder som teoretisk tid. PROMMIS 880 tar här ca. 104 sekunder på sig.

Typ 2758 är en 2716 som vid tillverkningstesterna befunnits defekt i ena minneshalvan. Den innehåller alltså bara 1 Kbyte men är i övrigt helt lik 2716.

Tyvärre har det i fråga om 2716 uppstått ett undantag från regeln att komponenter med samma typnummer är kompatibla trots att de kommer från olika tillverkare. Det gäller att Texas Instruments typ TMS 2716 inte är kompatibel med alla andra tillverkas 2716. Den kräver tre matningsspänningar och programmeras helt annorlunda. PROMMIS 880 kan inte programmera TMS 2716.

För övrigt gäller att kompatibiliteten mellan de olika fabrikanterna inte är så fullständig vad gäller programmering som vid enbart läsning. Vid programmering av en ny leverantörs komponenter är det därför rekommendabelt att kontrollera att dessa är kompatibla med de fabrikat som finns omnämnda i PROMMIS 880 programmet. Se under huvudmenyns rubrik T, Typval.

Vid enbart läsning brukar inte kompatibiliteten vara något större problem. Det finns t.ex. minneskomponenter, gjorda med helt annan tillverkningsteknik, såsom bipolära PROM eller maskprogrammerade ROM, vilka vid läsning är helt pinkompatibla med 2716. Med undantag för en mindre skillnad på något kretsben är också vissa RAM-minnen och EEPROM-typer 2716-kompatibla. - Allt detta bekräftar 2716-typens ställning som dagens industristandard.

Dock finns det större typer, bl.a. 2732 som innehåller 4 Kbyte. Tyvärre gör vissa fabrikanter 2732 och andra 2732A, med skillnaden mellan typerna främst att 2732 använder 25 V programmerings-spänning medan för 2732A gäller 21 V. PROMMIS 880 hanterar båda typerna, dock måste användaren själv hålla reda på vilken typ som gäller. En förväxling här kan ge en förstörd minneskomponent (2732A är känslig för överspänning) eller otillförlitlig programmering.

Texas Instruments motsvarighet till 2732(A) heter 2532. Denna 4 Kbyte-typ programmeras alltid på samma sätt (med 25 V).

Nästa storlek på EPROM, 8 Kbyte, finns i typerna 2764 och 2564. De kan anses väl etablerade i dagens läge (början 1983) med flera existerande leverantörer och stabiliserade priser.

Dagens största EPROM-typ är 27128 från Intel. Man kan vänta sig att ytterligare leverantörer tillkommer.

5. RADERING AV EPROM-MINNET.

Alla idag använda EPROM-typer bygger på samma princip, nämligen att den i detta sammanhang höga programmeringsspänningen (21 - 26 V) åstadkommer ett lokalt överslag i kiselbrickan där man önskar programmera in en logisk "0"-a. Vid överslaget uppstår en liten elektrisk laddning som innesluts i kiseloxiden, vilken är en god isolator. Den inneslutna laddningen stannar kvar för lång tid, mer än 10 år är en vanlig specifikation, om den inte förstörs genom avsiktlig s.k. radering av minnet.

Vid leverans är alltid EPROM-minnet raderat, vilket innebär att alla bitarna är i läge "1". Läser man av en byte får man alltså värdet 255 decimalt (FF hex). Vid programmering inför man laddning i de bitar där man önskar ha "0".

För de modernare typerna (2716 och senare) kan man återkomma och komplettera programmeringen i enstaka celler. Dock gäller att man bara kan programmera en "1"-a till en "0"-a, aldrig tvärtom. För att ändra åt andra hållet måste man radera och då ändras alla bitarna i hela minnet till "1"-or.

Radering av EPROM-minnen sker genom att man utsätter kiselbrickan (chip) för intensiv ultraviolett strålning från en speciell s.k. raderlampa. Ljus från solen och lysrör innehåller också UV-strålning, dock i så låg dos att en radering skulle ta dagar eller veckor. Ändå bör man efter en programmering skydda minnet genom att sätta en klisterlapp på det genomsynliga fönstret.

Med en UV-raderare åstadkommer man fullständig radering på 15-30 minuter typiskt, beroende på lampans intensitet och EPROM-typen. Man bör observera att UV-lampor åldras och minskar i intensitet med tiden. För att få säker radering bör man regelbundet mäta lampans intensitet med ett speciellt optiskt instrument och korrigera raderingstiden efter det erhållna värdet.

Faktum är felaktig radering är en ganska vanlig källa till problem med EPROM-minnen. Man har i dessa fall raderat med för liten UV-dos. Den har visserligen precis räckt till så att en efterföljande avläsningskontroll har visat "1" i alla cellerna, men detta har varit marginellt och så har man efter viss tid, kanske tillsammans med förändrad temperatur eller matningsspänning, fått en gammal men dåligt raderad "0"-bit att krypa fram igen. Dessa fel kan vara intermittenta och därför svåra att finna.

Därför: Varning mot dålig radering av EPROM-minnena!

Om man inte har tillgång till en intensitetsmätare för UV-ljus bör man gardera sig mot raderingproblem genom att alltid ha betydande marginal, t.ex. genom att radera dubbla den tid som normalt behövs för den långsammaste kretsen i en leverans.

6. SPECIFIKATIONER PROMMIS 880.Användning.

Tillsammans med en dator ABC800 eller ABC80, beroende på programvara. Ansluts med en flatkabel till datorns serieports- och kassettbandskontakter. Expansionschassi behövs ej.

Versioner.

PROMMIS 880-800 med programvara för ABC800
 PROMMIS 880-80 " " " ABC80.

Genom komplettering med extra programvara kan samma PROMMIS 880 användas med såväl ABC80 som ABC800.

Programmen levereras på 5" flexskiva, i dubbeldensitet för ABC800 och enkeldensitet för ABC80.

Minneskrav.

Normalt 32 Kbyte RAM i datorn. ABC80-versionen av programvaran kan också arbeta med bara 16 Kbyte RAM, då med flera CHAIN till underprogram och med mindre editeringsbuffer (4 Kbyte i stället för 16 Kbyte).

EPR0M-typer:

<u>16 Kbyte</u>	<u>8 Kbyte</u>	<u>4 Kbyte</u>	<u>2 Kbyte</u>	<u>1 Kbyte</u>
27128	2764	2732	2716	2708 (trespänningstyp)
	2564	2732A	2516	2758
		2532		2508

Val av EPROM-typ.

Sker med ett instickbart typkort, vars information avläses av datorn. Samtliga ovanstående EPROM-typer ingår som standard.

Matningsspänning.

Inbyggt nätaggregat, 220 V 50 Hz, max. 20 W.

Programmeringstid:

2758,2508	52 s
2716,2516	104 s
2708	183 s
2732,2732A,2532	208 s
2764,2564	415 s
27128	830 s

Miljö.

Användningstemp. vid EPROM-läsning : 0 - 50 °C.

Användningstemp. vid EPROM-programmering, i
 enlighet med EPROM-tillverkarnas specifikationer : 20 - 30 °C.

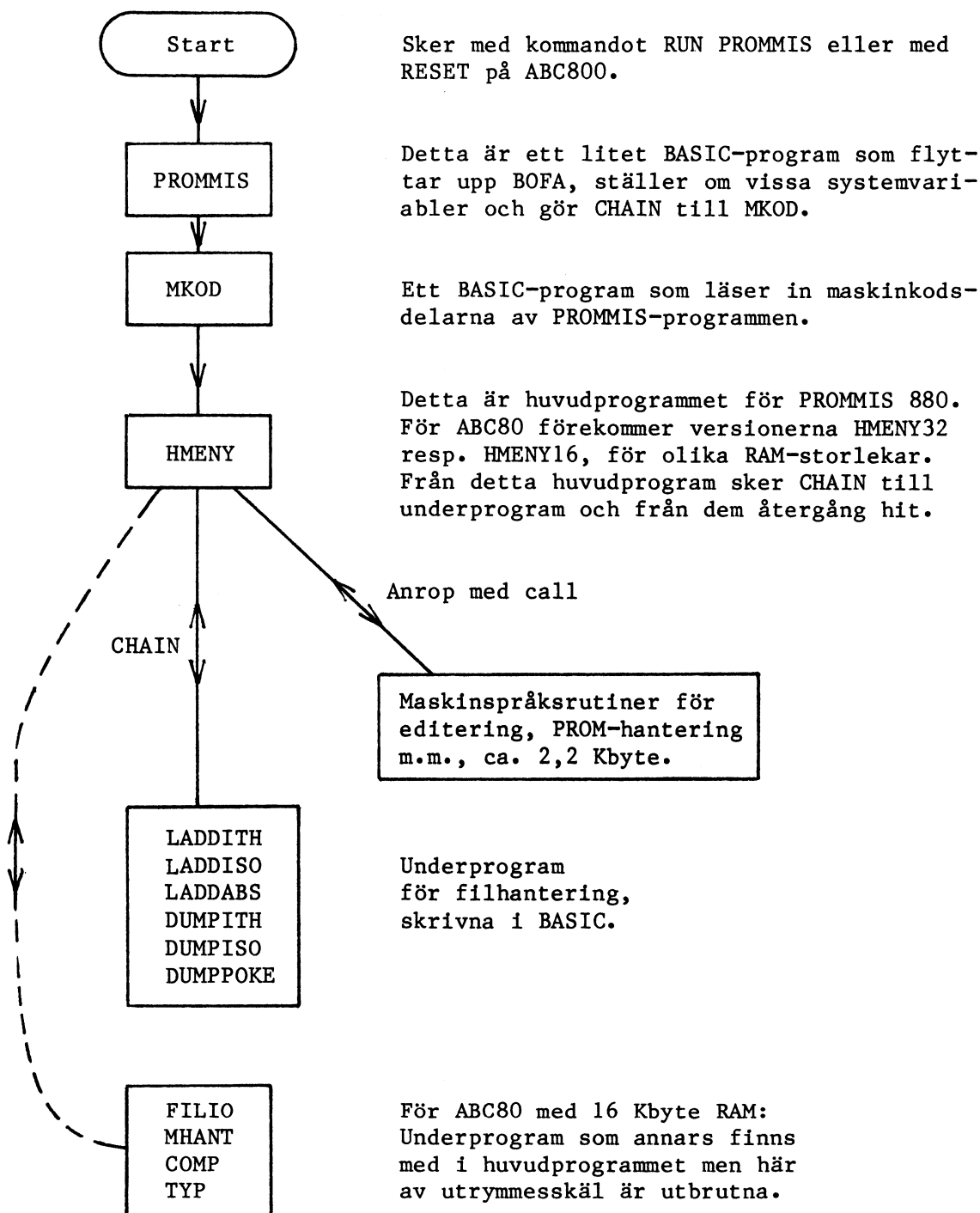
Fuktighet 10 - 90 %, utan daggutfällning.

Storlek, vikt.

Lådans mått 235 * 240 * 80 mm. Vikt 2 kg.

7. PROGRAMVARA.

Efter start är gången genom programmen följande:



Gemensamt för alla program är att svar på programmens frågor om laddningsadress, data och liknande kan ges i antingen hexadecimal eller vanlig decimal form. Se anvisningarna i den skärmbild som visas under programladdningen.

Huvudprogrammet tål att vid behov avbrytas med Control C. Återstarta med RUN eller RUN HMENU om programmet försvunnit.

Vi skall nu gå igenom de olika funktionerna i huvudmenyn, i den ordning som de presenteras på skärmen.

7.1. Filhantering.

Dessa underprogram utför inläsning och lagring av data i olika format. Gemensamt för programmen är att man har valbara möjligheter, t.ex. att ändra laddadresserna. Användandet av defaults gör det enklare att svara på programmens frågor.

LADDITH.

Laddar en ITH-fil, ett format som kan innehålla data med godtyckliga bytevärden (även i form av osynliga ISO-tecken). Detta format är lämpligt för att lagra data i ett EPROM-minne t.ex. som backup.

LADDISO.

Laddar en klartextfil, t.ex. BASIC's egna BAS-filer. För att spara utrymme i PROM vid programplacering där finns de valbara möjligheterna att slopa REM-satser och/eller mellanslag. Detta sker i så fall på ett BASIC-grammatikaliskt riktigt sätt. Även "halva" REM-satser behandlas, som exempel

```
150 GOSUB 710 : REM Kommentartext här
blir 150 GOSUB 710.
```

Mellanslag slopas inte om de ingår i en sträng.

Mycket långa textfiler ryms inte på en gång i editeringsbuffern. Därför har man möjlighet att hoppa över ett antal tecken i filens början. Då man också kan specificera inläsningsbufferns storlek har man med detta möjlighet att hantera en lång fil i många små omgångar.

LADDABS.

Laddar filer i ABS-format. Man anger inom vilket adressområde i ABS-filen som data skall hämtas och var de skall laddas i minnet. Om filen är längre än den tillåtna laddlängden får man göra laddningen i flera omgångar.

LADDITH och LADDABS kan fås att ladda till annan adress i minnet än till den laddadress som finns med i filen. Observera då att det är frågan om en ren förskjutning av laddadresser. Om det finns "hål" i infilens dataflöde, t.ex. genererade med assemblerinstruktionerna ORG, DEFS eller DS, kommer därför motsvarande hål att finnas oladdat i minnet.

DUMPITH.

Ett program som skriver ut en minnesarea till en fil i ITH-format. Den skapade filen kan senare läsas in med LADDITH.

Standardlängden i utfilen är 14 databyte per rad, detta för att raderna skall kunna listas (med speciellt program) på ABC80's 40-teckens skärm. Antalet kan modifieras med om tilldelningen av variabeln R% ändras i rad 330, dock maximalt till värdet 53.

DUMPISO.

Motsvarigheten till LADDISO. Man bör i normala fall välja att inte låta osynliga ISO-tecken ingå i filen.

Om det uppstår en utskriftsrad som är längre än 117 tecken kommer ett extra vagnreturtecken (CR=ODH) att inskjutas efter tecken nr. 117. Alltför långa rader kan nämligen inte läsas in senare av BASIC.

DUMPPOKE.

Utfilen är här ett BASIC-program, bestående av POKE-satser för data. Något motsvarande laddprogram behövs inte; det är bara att exekvera BASIC-programmet.

DUMPITH, DUMPISO och DUMPPOKE kan fås att ge sina utskrifter till en skrivare (eller skärm för ABC800) i stället för på fil. Detta genom att ange enhetsnamnet, t.ex. PR: för skrivaren, (ev. CON: för ABC800's skärm) som svar på frågan om filnamn.

7.2. Editering i minne.

Editeringsprogrammet utför visning och möjliggör ändringar av innehållet i editeringsbuffern, vilken är en mellanlagringsarea i datorns minne där data till och från PROMMIS 880 passerar. Editeringsbuffern är 16 Kbyte stor (4 Kbyte för ABC80 med bara 16 Kbyte RAM).

Editorns egenskaper är till stor del självförklarade, se kommando E i huvudmenyn. Alla kommandon finns förklarade på skärmens nedersta rader. För kommandona används på ABC800 funktionsknapparna medan man på ABC80's tangentbord skall ge koder med Control och ett tecken, åtstadkomna genom att knapp Control trycks ned innan och samtidigt med teckenknappen.

Det finns tre arbetsmoder för editorn: med hexadecimal, decimal eller textmässig visning av data.

Hexadecimal mod är den mest användbara, speciellt som här också visas samma area i textform. Dock visas inte så många tecken per rad som i textmod.

Decimal mod är till nytta om man t.ex. vill knappa in data angivna i denna form. Växling mellan hexadecimal och decimal arbetsmod kan sägas vara en form av decimal-hexomvandling för enkelbytevärden. Samma omvandling för 16-bitars tal utför man genom att kommendera editering i en viss adress och sedan avläsa adressen på översta raden.

I textmod, och enbart i denna, finns möjlighet att skjuta in och stryka tecken. Vid inskjutning införs ett mellanslagstecken. Den förskjutning av minnesdata som sker måste vara begränsad för att inte påverkan skall ske av PROMMIS 880's egna program. Begränsningen sker vid nästa "storsida", multipel av 4096 (1000H). Det är alltså maximalt 4096 tecken som flyttas då man stryker eller skjuter in tecken.

I textmod kan man inte införa kontrolltecken i data, utan enbart skrivbara ISO-tecken. Inte ens vagnretur (CR=ODH) kan införas, vilket kan vara till besvär om man skriver in en lång löpande text, t.ex. ett BASIC-program. Gå då ur editorn och fyll (kommando Y+F) buffern med vagnreturtecken. Åter i editeringen kan man nu skriva in sin text och bara hoppa, med högerpilen, över det befintliga vagnreturtecknet vid varje radslut.

7.3. Minnestransaktioner.

I denna grupp ingår huvudmenyns kommandon Y och J, innehållande några hjälprutiner som berör minnet.

Kommando Y används för att fylla, flytta eller checksummeberäkna minnet. Den senare funktionen kan vara användbar för att kontrollera att minnet inte påverkats efter t.ex. provkörning av ett nytt maskinspråksprogram. Checksumman över hela BASIC-ROM'et, adress 0-16383 för ABC80 resp. 0-24575 för ABC800, anger vilken version av BASIC-tolk som används och det erhållna värdet skall stämma överens med det som finns angivet baktill på datorn.

Jämförelse, J i huvudmenyn, sker alltid med en minnesarea som ena parten. Den andra är antingen PROM-data eller en annan minnesarea. Om det finns några avvikelser så skrivs adresser och data ut på skärmen. Mellanslag ger nästa utskrift, medan RETURN-knappen avbryter, alltså ger återgång till huvudmenyn.

7.4. PROM-transaktioner.

L i huvudmenyn medför inläsning av PROM-data till minnet. Användandet av standardantaganden gör att man oftast bara behöver svara med tre RETURN-tryckningar på parameterfrågorna.

Inläsningen föregås av placeringskontroll, omfattande att PROM-kapseln inte saknas, inte är felvänd och (i frågan om 24-bens-kretsar) inte är förskjuten i hållaren. (Närmare bestämt kontrolleras att alla databitar är drivande vid läsning från PROM). Vissa rent elektriska fel i minneskretsen ger också utslag i kontrollen. Kontrollen kan dock inte upptäcka om fel typ av EPROM-minne satts i sockeln.

Raderkontrollen, huvudmenyns R, är en speciell variant av jämförelse där programmet kontrollerar att det står 255 (FFH) i alla PROM-cellerna. Utskrifterna är likartade med dem för jämförelserutinen.

Som första steg i raderkontrollen sker placeringkontroll av EPROM-kapseln.

Raderkontrollen kan upprepas, för nästa EPROM-krets, med knapp REPEAT på PROMMIS-panelen.

7.5. EPROM-programmering.

Initieras med P i huvudmenyn. Efter förfrågan om adresser sker en automatisk raderkontroll. Man kan gå vidare med programmering även om kretsen skulle visa sig vara redan programmerad.

Om användaren inte redan slagit till programmeringsspänningen får han/hon nu en uppmaning att göra så. Programmeringen börjar då den röda lysdioden tänds. Samtidigt börjar nedräkningstiden att visas på nedersta raden.

Programmeringen kan avbrytas med knapp PF2 (ABC800) eller Control Å (ABC80). Minneskretsen är dock redan delvis programmerad, varför radering bör ske om man angivit fel adress el. likn.

Då programmeringen är klar sker automatiskt verifiering, vilket är en jämförelse minne mot PROM-data. Om verifieringen gick utan anmärkning kan man byta EPROM-krets i hållaren och bara trycka på knapp REPEAT för att åstadkomma ytterligare en kopia.

8. ELEKTRONIK.

Data- och kontrollsignaler överförs till och från PROMMIS 880 i ett speciellt serieformat (ej V24 eller RS-232). Fyra utsignaler, varav en utgör klocka, och en insignal används för överföringen. Dessa signaler härrör från PIO-kretsen i ABC80 resp. från SIO- eller DART-kretsen i ABC800, beroende på kanalval.

I enheten används digitala kretsar av CMOS-typ, vilka är relativt känsliga för statisk elektricitet. Panelen på PROMMIS 880 är via en resistor ansluten till jord, varför ett överslag till panelen inte medför någon skada. Man bör dock se upp med att beröra panelen först, inte EPROM-kretsen eller den tomma sockeln. Detta bör framför allt observeras då man har en heltäckningsmatta i rummet.

Kretsschemat för PROMMIS 880 finns bland bilagorna.

9. MINNESKARTA.

	ABC800	ABC80 med 32 K RAM	ABC80 med 16 K RAM	
	Systemvariabler	POKE-area		FF80H (65408)
FF00H (65280)				
FE00H (65024)	POKE-area			
	Systemvariabler			FD00H (64768)
	DOSBUF 6,7 = CASBUF 1,2			FB00H (64256)
	PROMMIS 880 maskinspråksrutiner lagrade i DOSBUF 2-5			F700H (63232)
	DOSBUF 0,1			F500H (62720)
	Stack		Stack	
	BASIC-program HMENY m.fl. (växer uppåt)		BASIC-program HMENY16 m.fl. (växer uppåt)	D000H (53248)
			Editerings- buffer (4 K)	C000H (49152)
	Editeringsbuffer (16 Kbyte)		Minne saknas	8000H (32768)
	Video-RAM Printerdriver i ROM			7000H (28672)
	DOS i ROM			6000H (24576)
		ledigt minnesutrymme		4000H (16384)
	BASIC i ROM			0000H (0)
	ABC800	ABC80 med 32 K RAM	ABC80 med 16 K RAM	

10. TERMFÖRKLARINGAR.

ASCII	American Standard Code for Information Interchange. Ett mycket använt standardalfabet, som ange vilka siffervärden som motsvarar bokstäver, siffror och andra tecken. Liknar ISO-alfabetet, saknar dock Å, Ä, och Ö.
bipolär	Den bipolära tillverkningstekniken för integrerade kretsar, inklusive minnen, ger kretsar som är snabba men relativt effektkrävande. Det finns bipolära både PROM och RAM.
CHAIN	Inläsning av delprogram från ett exekverande program sker i BASIC med CHAIN. Se datorns bruksanvisning.
decimal-hex-omvandling	Beräkning av den hexadecimala motsvarigheten till ett decimalt tal. Kan ske med editorn, dess aktuella adress visas överst i båda talrepresentationsformerna.
default	Standardantagande. Ett program kan för bekvämlighets skull ha defaults för sådana frågor till användaren som oftast bara har ett visst svar. Man behöver bara svara med RETURN om standardsvaret gäller.
DOS	Disk Operating System, programmet som hanterar flexskiveminnet.
dynamiska minnen	Denna typ av minneskretsar fordrar regelbunden uppdatering, s.k. refresh, var 2-4 ms för att inte tappa minnet. De har i gengäld större kapacitet än statiska minnen.
editering	Manuell ändring av text eller data. I PROMMIS 880 finns för detta ändamål som hjälpprogram en editor, av typ kallad minneseditor eftersom den enbart arbetar på data i minnet.
editeringsbuffer	Den area i minnet till vilken data inläses innan programmeringen.
EEPROM	Electricallt Erasable Programmable Read Only Memory, ett elektriskt raderbart PROM. En vidareutvecklad EPROM vilken inte behöver UV-ljus för att raderingen. Ofta kan också enstaka byte raderas.
EPR0M	Erasable Programmable Read Only Memory, av användaren raderbart och programmerbart läsminne. En speciell typ av PROM som med hjälp av UV-ljus kan raderas och återanvändas.
hexadecimala tal	Tal angivna i talsystemet med 16 som bas. Kallas ofta för bara hexstal. Som siffror i talsystemet används 0-9,A,B,C,D,E,F.

hex-decimal- omvandling	Beräkning av det decimala värdet av ett hexadecimalt tal. Kan ske med editorn, dess aktuella adress visas överst i båda talsystemen.
ISO-alfabetet	av International Organization for Standardization. Ett standardalfabet för bokstävers och andra teckens siffervärden. Likt ASCII, har dock med Å, Ä, Ö. Omfattar 96 skrivbara tecken och 32 kontrolltecken. Ex.-vis "A"=65 och vagnretur=13.
K	vid användning i digitala sammanhang (ex.vis Kbyte) betecknande 1024. Uttalas "kilo". Bör skiljas från k (ex.vis i kg) som anger 1000.
kompatibel	Överensstämmande. Då två typer av integrerade kretsar är ben- eller pin-kompatibla kan man direkt ersätta dem med varandra i alla tillämpningar.
maskprogrammerad ROM	Vid halvledartillverkning används fotografiska masker vilkas utformning avgör minnets innehåll. Man kan beställa tillverkning efter en egen mask, dock med minimiantal kanske 2000 st. Alternativet i mindre serier är PROM eller EPROM.
PROM	Programmable Read Only Memory, programmerbart läsminne. En apparat- eller systemtillverkare kan själv definiera innehållet i sina kretsar (programmera dem). Användningsmässigt helt likt andra ROM, se också detta ord.
promma	Teknisk slang för att placera data eller program i PROM-minne.
RAM	Random Access Memory, engelsk något oegentlig benämning på läs/skriv-minne. En minneskomponent vars innehåll kan ändras lika ofta och fort som man kan läsa ur den.
ROM	Read Only Memory, läsminne, en minneskomponent med fastlagt innehåll. Tappar ej minnet vid spänningsavbrott. Undergrupper: PROM, EPROM, EEPROM m.fl. Med enbart ROM avses ofta maskprogrammerade ROM-komponenter.
statiskt minne	Nästan alla ROM-minnen är statiska, dvs. fordrar inga klockade insignaler. RAM-minnen finns av både statisk och dynamisk typ.
skrivbar	ISO-koderna 32-127 (20H-7FH) motsvaras av bokstäver och andra utskrivbara tecken. Motsats: kontrolltecken (kod 0-31).

BILAGA 1.

ITH-formatet.

Exempel på en ITH-fil:

```
:10400000CD246DE12A0AFE22F87F21F87F220AFEE4
:104010002E52264F3E4D22FA7F32FC7F21584022FD
:10402000FD7F213740011800CD0B00FD360E012128
:104030004F40E5FBC3F0000D4175746F6D61747303
:10404000746172742066727D6E20524F4D3A0D522B
:10405000554E20524F4D3A0DC37540C37540C37E37
:1040600040C38040C37340C37340C37340C37340B5
:10407000C37340D7B4210041DD750CDD740DA7C9B1
:10408000DD5E0CDD560D1AFE0320039737C977134A
:0D40900023FE0D20F1DD730CDD720DA7C9BC
:00000001FF
```

Varje rad börjar med kolon (:).

De två första tecknen efter kolon anger antalet databyte i raden. Hexadecimalt format används här och överallt i ITH-filen, 10 ovan betyder alltså 10H = 16 byte per rad i t.ex. den första raden.

Nästa fyra tecken anger laddadressen för radens data. Vårt exempel ovan är alltså avsett att laddas med början i adress 4000H (16384 decimalt).

Därefter följer två tecken som anger vilken typ av datarad det är frågan om. Typ 00 anger vanliga datarader, medan typ 01 anger slutrad.

Nu följer data, så många byte som antalsuppgiften avgivit. Varje byte data representeras i filen av två hextecken. Den första databyten i vårt exempel ovan är CDH (205 decimalt).

De sista två tecknen i varje rad är checksumma. Den är beräknad så att summan av alla byte, inkl. de för antal, laddadress, radtyp och checksumma, blir noll (om alla carry över enkelbyte-resultatet ignoreras).

Filen avslutas med raden ":00000001FF".

BILAGA 2.

ROMDRIV för PROMMIS 880

```

;*****
;
; Programmet innehåller devicedrivern
; för ROM: inkl. hopptabell.
;
;*****
000D CR EQU 0DH ;ISO-tecken vagnretur
;
C000 ORG 0C000H
;
; Placering vid 16 k minne. Relokeras av BASIC-
; programmet ROMDEVIC vid annan minnesstorlek.
;
C000 C31DC0 JP OPENPR ;hopptabellen : OPEN
C003 C31DC0 JP OPENPR ; PREPARE
C006 C326C0 JP CLOSE
C009 C328C0 JP INPUT
C00C C31BC0 JP ROMERR
C00F C31BC0 JP ROMERR
C012 C31BC0 JP ROMERR
C015 C31BC0 JP ROMERR
C018 C31BC0 JP ROMERR
;
C01B D7 ROMERR RST 10H ;Ger fel 52 ="EJ TILL
C01C B4 DEFB 52+80H ; DENNA ENHET"
;
C01D 210040 OPENPR LD HL,4000H ;Adress för ROM, ändras
; av BASIC-programmet
C020 DD750C LD (IX+12),L ;Spara i enhetens
C023 DD740D LD (IX+13),H ; parameterfält
;
C026 A7 CLOSE AND A ;Nollad Carry vid
C027 C9 RET ; utgång
;
C028 DD5E0C INPUT LD E,(IX+12) ;Hämtpekare i DE
C02B DD560D LD D,(IX+13)
C02E 1A INPLO LD A,(DE) ;Hämta ett tecken
C02F FE03 CP 3 ;Sluttecken ?
C031 2003 JR NZ,INPA
;
C033 97 SUB A ;Ja, återgå med
C034 37 SCF ; reg A=0 och
C035 C9 RET ; Carry=1
;
C036 77 INPA LD (HL),A ;Lagra tecknet där
;BASIC vill ha det
;öka pekarna
C037 13 INC DE
C038 23 INC HL
C039 FE0D CP CR
C03B 20F1 JR NZ,INPLO ;Radslutstecken?
C03D DD730C LD (IX+12),E ;Ja, återlagra hämtpekaren
C040 DD720D LD (IX+13),D
C043 A7 AND A ;Åter med CY=0
C044 C9 RET
END
```

ASTART för PROMMIS 880

```

;*****
;
; Autostartprogram. Efter RESET eller kraftpå-
; slag kommer programmet i ROM, med start i
; adress 4100H, att exekveras. Innehåller också
; devicedriver för enheten ROM:.
;
;*****
000D CR EQU 0DH ;ISO-tecken vagnretur
;
4000 ORG 4000H
;
4000 CD246D CALL 6D24H ;Om flexskiva används så
; skall här vara CALL
; till den adress som
; från början fanns i
; 604CH. - Om flexskiva
; ej används stryks
; denna instruktion.
;
4003 E1 POP HL ;kasta återhopsadr
;
;Lägg nu in enhet ROM: i devicelistan. Vi kan
; förutsätta att den inte redan finns (RESET).
;
4004 2A0AFE LD HL,(OFEOAH) ;devicepekare
4007 22F87F LD (7FF8H),HL
400A 21F87F LD HL,7FF8H ;ROM: inskjutes
400D 220AFE LD (OFEOAH),HL ;i osynligt
4010 2E52 LD L,'R' ;videominne
4012 264F LD H,'O'
4014 3E4D LD A,'M'
4016 22FA7F LD (7FFAH),HL
4019 32FC7F LD (7FFCH),A
401C 215840 LD HL,HOPPT ;hopptabellpek
401F 22FD7F LD (7FFDH),HL
;
; Skriv meddelande om automatstart.
;
4022 213740 LD HL,MESS ;textbörjan här
4025 011800 LD BC,RKOM-MESS ;antal tecken
4028 CD0B00 CALL OBH ;utskriftsrutin
;
;
402B FD360E01 LD (IY+14),1 ;Sätt kommandomod
402F 214F40 LD HL,RKOM ;Kommandopekare
4032 E5 PUSH HL
4033 FB EI ;Tillåt avbrott
4034 C3F000 JP OFOH ;Åter BASIC
;
4037 0D MESS DEFB CR
4038 4175746F
403C 6D617473
4040 74617274
4044 2066727D
4048 6E20524F
404C 4D3A DEFM 'Automatstart från ROM:'
404E 0D DEFB CR

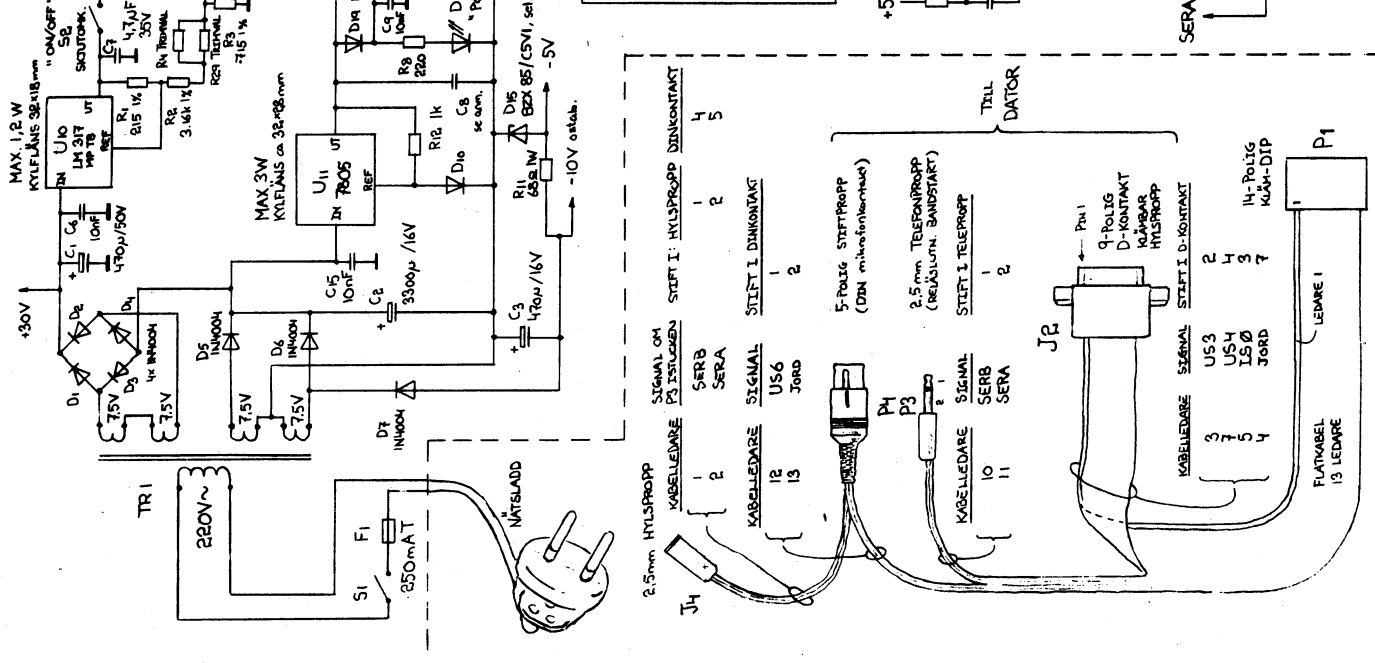
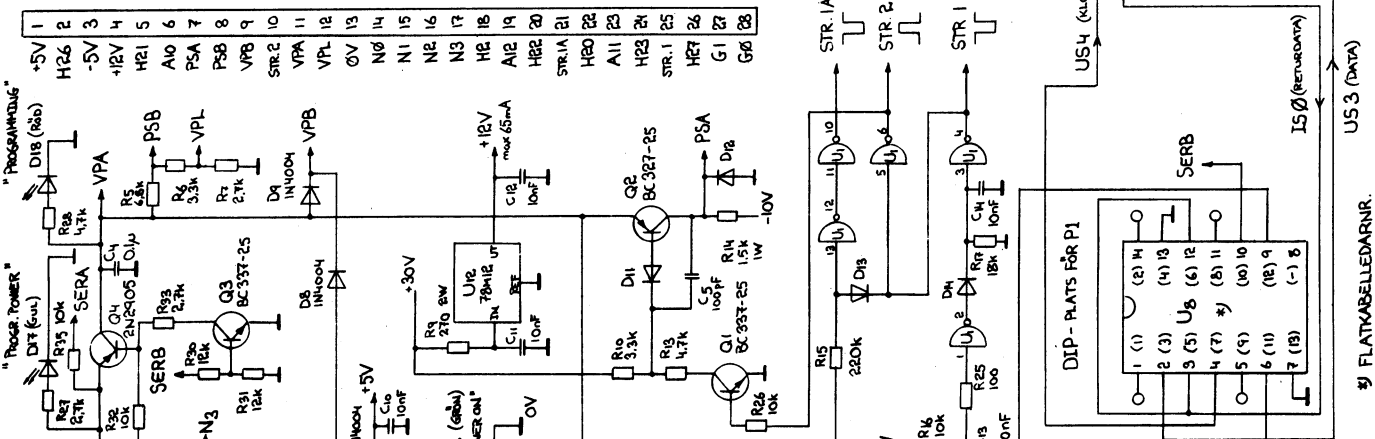
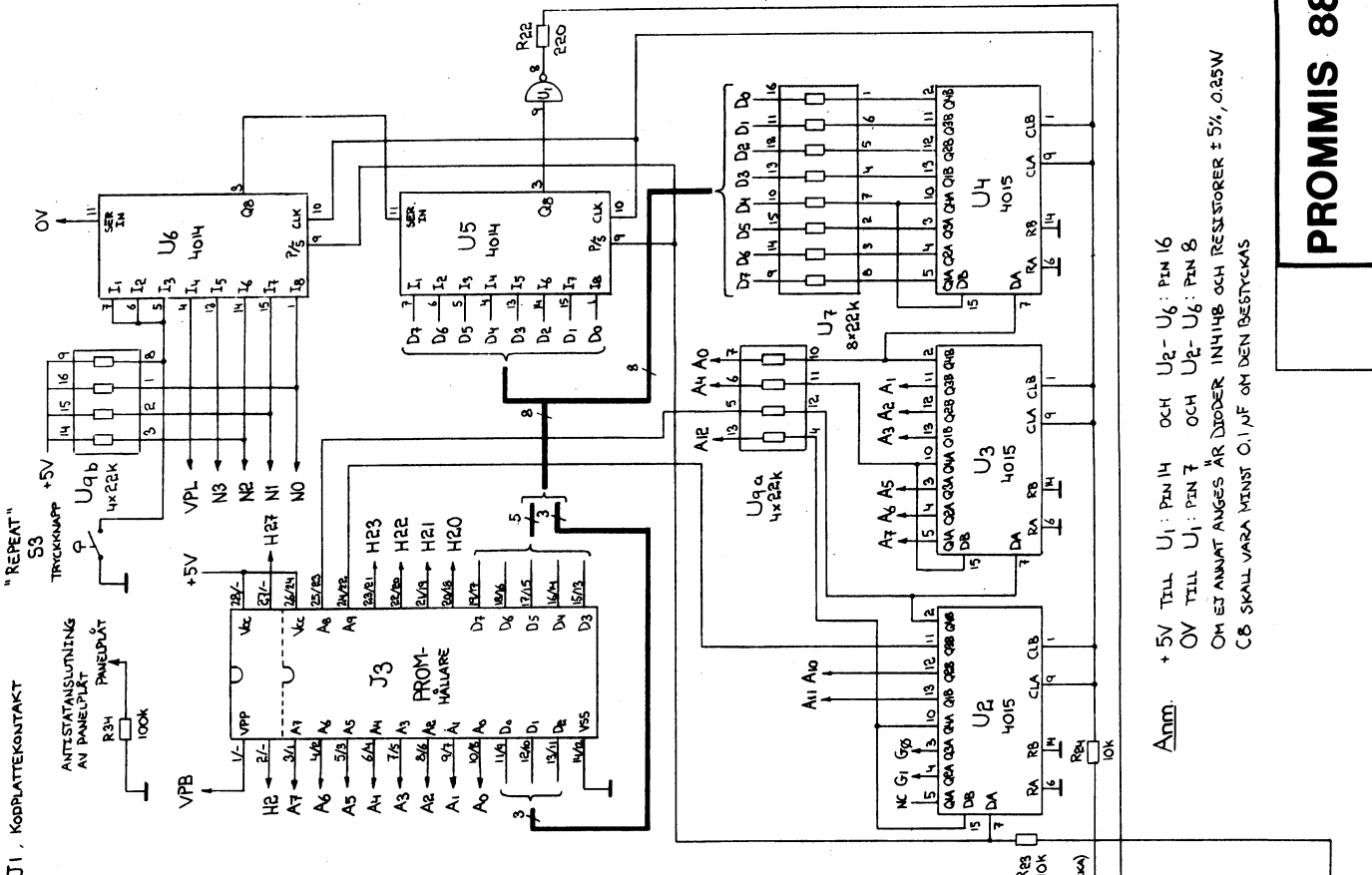
```


ASTART för PROMMIS 880

```

404F 52554E20
4053 524F4D3A RKOM  DEFM ^RUN ROM:~
4057 OD             DEFB CR
;
4058 C37540 HOPPT  JP   OPENPR ;hopptabellen : OPEN
405B C37540         JP   OPENPR ; PREPARE
405E C37E40         JP   CLOSE
4061 C38040         JP   INPUT
4064 C37340         JP   ROMERR
4067 C37340         JP   ROMERR
406A C37340         JP   ROMERR
406D C37340         JP   ROMERR
4070 C37340         JP   ROMERR
;
4073 D7            ROMERR RST 10H      ;Ger fel 52 ="EJ TILL
4074 B4            DEFB 52+80H      ; DENNA ENHET"
;
;
4075 210041 OPENPR LD  HL,4100H      ;Adress BASIC-text
; i ROM
4078 DD750C         LD  (IX+12),L    ;spara i enhetens
407B DD740D         LD  (IX+13),H    ; parameterfält
;
407E A7            CLOSE AND  A      ;Nollad Carry vid
407F C9            RET              ; utgång
;
;
4080 DD5E0C INPUT  LD  E,(IX+12) ;Hämtpekare i DE
4083 DD560D         LD  D,(IX+13)
4086 1A            INPLO LD  A,(DE)  ;Hämta ett tecken
4087 FE03          CP  3            ;Sluttecken ?
4089 2003          JR  NZ,INPA      ;Ja, återgå med
;Ja, återgå med
408B 97            SUB  A            ; reg A=0 och
408C 37            SCF              ; Carry=1
408D C9            RET
;
408E 77            INPA LD  (HL),A   ;Lagra tecknet där
;BASIC vill ha dem
;
408F 13            INC  DE           ;öka pekarna
4090 23            INC  HL
;
4091 FE0D          CP  CR           ;Radslutstecken?
4093 20F1          JR  NZ,INPLO
;Ja, återlagra
4095 DD730C         LD  (IX+12),E   ;hämtpekaren
4098 DD720D         LD  (IX+13),D
409B A7            AND  A            ;Åter med CY=0
409C C9            RET
;
END

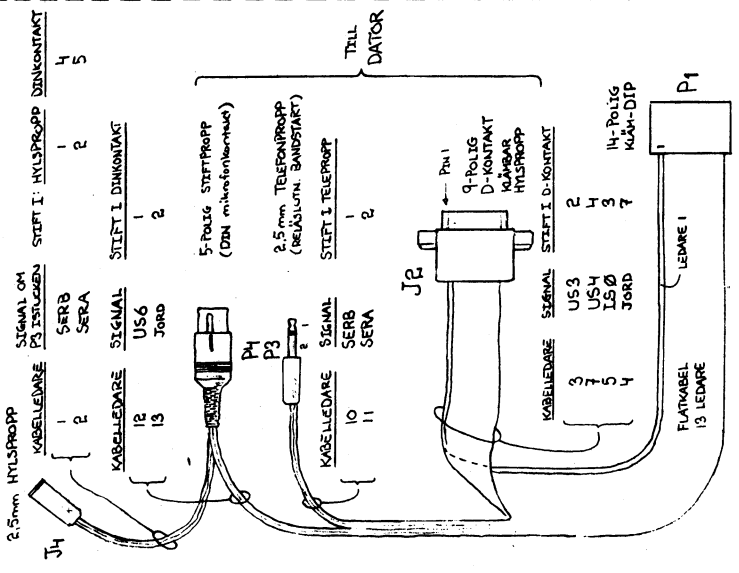
```



Ann.
 +5V TILL U1: PIN 14 OCH U2-U6: PIN 16
 OV TILL U1: PIN 7 OCH U2-U6: PIN 8
 OM ET ANVÄT ANGES ÄR DIODER IN4148 OCH RESISTORER ±5%, 0,25W
 CB SKALL VARA MINST 0,1µF OM DEN BESTYCKAS

PROMMIS 880
 Ing.firma ELogik
 1983-02-22

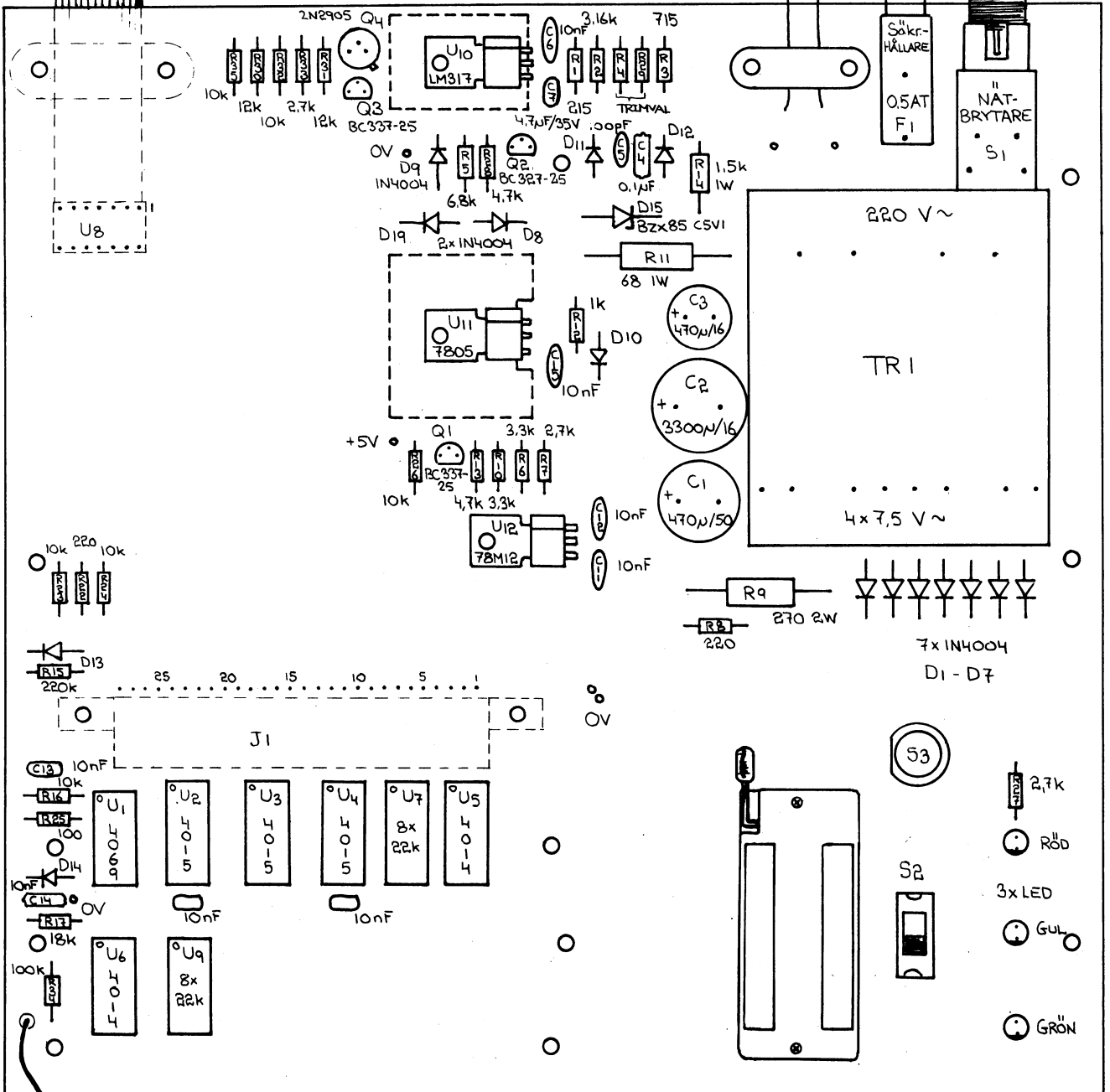
*) FLATKABELLEDARNR. INOM PARANTESEN
 — O Anger NC men med lödd



13 ledare FLATKABEL

LEDARE 1 (RÖD)

NÄTSLADD



TILL FRONTPANEL (ANTISTAT-SKYDD)

STRECKADE DETALJER SITTER PÅ KORTETS UNDERSIDA.

ALLA DIODER DÄR EJ TYP ANGIVITS = 1N4148

PROMMIS 880
KOMponentPLACERING

1983-02-22

Ing.firma ELogik