

M-PROG

METRIC CARD

TEKNISK MANUAL



M-PROG

METRIC CARD PROG

EPROM Programmerare

©Scandia Metric AB 831101

INNEHÅLL

=====

1.	INLEDNING	3
2.	KORTETS UPPBYGGNAD	4
2.1.	Kontakter och socklar	6
2.2.	I/O adressering	7
2.3.	Interna buffrar	8
2.4.	30 Volts generering och övriga spänningar .	10
2.5.	Reläfunktioner	11
2.6.	VPP reglering, drivsteg och VPP kontrolli .	12
2.7.	Spänningsstyrning 0 / 5 / 12 V	13
2.8.	Spänningsstyrning 5 / 12 V	14
3.	TESTNING AV KORTET	15
4.	PRINCIPSCHEMA	17
5.	KOMPONENTPLACERING	19
6.	KOMPONENTFÖRTECKNING	20
7.	TEKNISKA DATA	23
App 1.	BYGLINGSKONTAKTEN FÖR OLIKA EPROMAR	24

1. INLEDNING

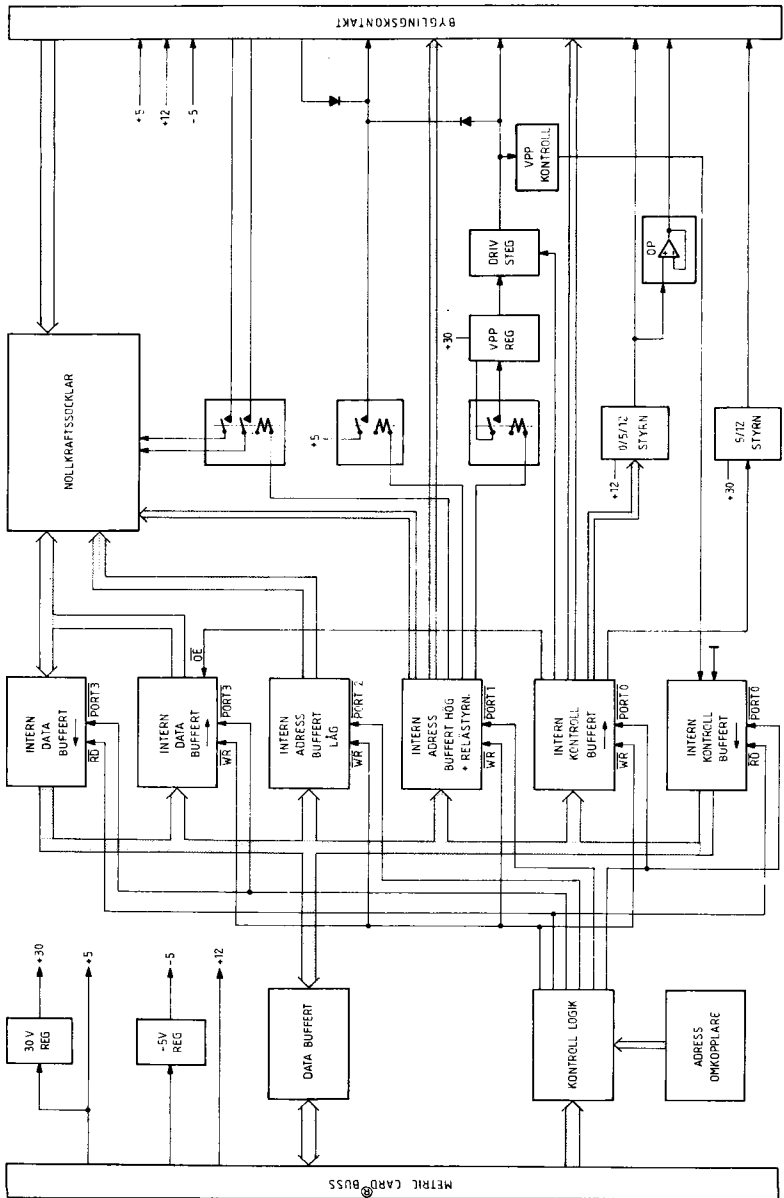
METRIC CARD *) PROG är en EPROM programmerare som ansluts till METRIC CARD bussen. Den klarar EPROM:ar från 2708 till 27128. Till kortet finns ett applikationsprogram som körs under CP/M **) på antingen METRIC 85 eller METRIC 8. Kortet är ett förlängt enkelt Europakort.

På kortet finns en TTL-del mot bussen och delvis ut mot EPROM-socklarna. Dessutom finns logik och komponenter för att kunna generera och styra de spänningar som krävs för programmering v de olika EPROM:arna.

Sätt endast i, och ta ur, METRIC CARD kort med spänningen FRANSLAGEN !!!

*) METRIC CARD is a registered trademark of Scandia Metric AB
**) CP/M is a registered trademark of Digital Research

2. KORTETS UPPBYGGNAD



Figur 1. Blockschema för M-PROG.

Allmänt gäller att samtliga använda bussignaler endast belastas med en TTL LS last.

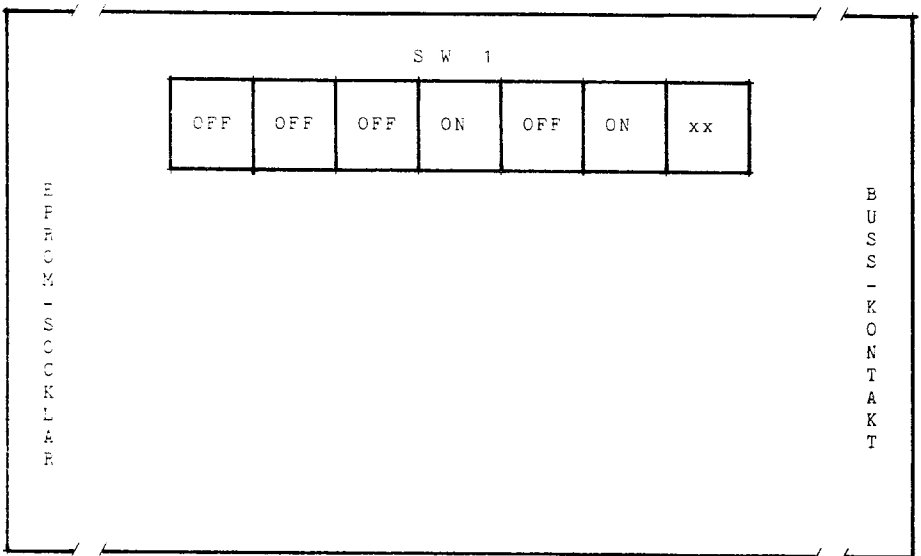
2.1. Kontakter och socklar

På kretskortet finns två nollkraftssocklar, en för 24-pinnars och en för 28-pinnars EPROM:ar. Kortet är inte avsett att kunna programmera två stycken EPROM:ar parallellt !

Dessutom finns en 25-polig DSUB anslutning, till vilken man behöver en EPROM-beroende byglingskontakt. Denna byglingskontakt innehåller inga komponenter.

2.2. I/O adressering

Kortet adresseras via fyra I/O portar i sekvens. Basport-adressen ställs in på en adressomkopplare (DIP-switch SW 1). Omkopplaren kan ställa de sex mest signifikanta bitarna till ett valfritt värde, de övriga två behövs för avkodningen på kortet. Den rekommenderade METRIC CARD standarden är E8H som portadress.



Figur 2. Inställning av kortadress E8H.

Den inställda kortadressen jämförs kontinuerligt med adressbussens A7 - A2 genom två stycken 74LS85 (Z1, Z2). = utgången grändas med IOREQ som vid rätt I/O adress selekterar en 2 till 4 avkodare (74LS139 (Z4)) som avkodar A1 till A0. De fyra utgångarna går till var och en av de fyra interna I/O buffrarna grändade med RD eller WR.

Kortadressen består av ett 6 bitars tal, med de två minst signifikanta bitarna avkodade som en av fyra I/O adresser. Den mest signifikanta biten är den på omkopplaren som sitter närmast nollkraftssocklarna. En bit tolkas som etta om man ställer omkopplaren i läge "OFF" eller "OPEN", en bit tolkas som nolla om man ställer omkopplaren i läge "ON" eller "CLOSED".

2.3. Interna buffrar

Databussen buffras med en bidirektionell driver LS245 (Z5). Dess riktning är alltid från METRIC CARD bussen, utom vid läsning från kortet. Efter LS 245:an ligger en intern buss, som går till fyra latchade 8-bits buffrar.

Alla utgångar från de latchade buffrarna har pull-up motstånd.

Interna dataporten (basport + 3)

Denna är den enda helt bidirektionella porten. Den består av 2 kretsar, en LS374 (Z10) för flöde till EPROM:et och en LS244 (Z11) från EPROM:et. För att dataporten skall vidarebefordra sitt innehåll till EPROMsockeln, måste bit 7 i kontrollporten sättas låg.

Intern adressbuffer låg (basport + 2)

Den låga adressbyten går direkt till EPROM:et. Porten är enkelriktad (LS273 (Z12)).

Intern adressbuffer hög (basport + 1)

Den höga adressbyten signalerna delas på tre sätt. A8 - A9 går direkt till nollkraftssocklarna. Adresserna A10 - A12 går via bygglingskontakten, eftersom deras anslutning till nollkraftssockeln är EPROM-typberoende. A13 A15 styr de tre reläerna. Porten är enkelriktad (LS273 (Z13)).

Interna kontrollporten (Basport + 0)

Kontrollporten har en blandad funktion.

Bit 0 och Bit 1 hanterar 0/5/12 Volts styrningen.

Bit 1	Bit 0	Utspänning
0	0	0 V
0	1	0 V
1	0	12 V
1	1	5 V

Bit 2 styr drivsteget.

Bit 2	VPP
0	0 V
1	21/26 V

Bit 3 - 5 är rena TTL signaler som går till byglingskontakten. Deras funktion är CE, AR och OE. Dessa signaler har till vissa EPROM en annan funktion än vad deras logiska namn anger.

Bit 6 sköter 5/12 Volts styrningen.

Bit 6	Spänning
0	12 V
1	5 V

Bit 7 styr OE på databufferten, se ovan vid den interna dataporten. Signalen är aktivt låg.

Kontrollporten är i princip enkelriktad (LS273 (Z14)). Kontrollporten kan dock lämna två statussignaler vid läsning. Vid läsning gäller bit 0 som VPP-kontroll och bit 7 ska vara 0.

Bit 0	VPP test
0	< 16 V
1	>= 18 V

2.4. 30 Volts generering och övriga matningsspänningar

+ 5 och + 12 Volt tas direkt ifrån bussen.

- 5 V behövs till vissa kopplingar och den erhålls från - 12 V via en stabiliserad spänningsomvandlare 7905. Den kan lämna 500 mA.

För att kunna programmera de flesta EPROM behövs en programmeringsspanning på 21, 25 eller 26 volt. En sådan hög spänning finns inte i bussen utan skapas i en DC/DCkonverter från + 5 V till + 15 / - 15 V. Då denna har galvaniskt skild jord, kan man ta ut 120 mA vid + 30 V.

2.5. Reläfunktioner

På kortet finns tre reläer.

Relä 1 ställer max VPP spänning.

REL 1	Max VPP
OFF	26.0 V
ON	21.4 V

Relä 2 släpper fram + 5 V till diodernas katoder.

REL 2	Katodsigan
OFF	-
ON	5 V

Relä 3 kan bryta två signaler från byglingskontakten.

2.6. VPP reglering, drivsteg och VPP kontroll

VPP regleringen sker med en justerbar spänningsregulator LM 317. Med hjälp av ett relä kan två olika spänningsdelare skapas. Dessa två kan ställa utspänningen till 26.0 eller 21.4 V. Spänningsdelarna kan finjusteras med R22 (för 21.4 V) och R23 (för 26.0 V).

Utspänningen från VPP regleringen går in till ett drivsteg som kan styras till 0 V eller VPP. Drivsteget kan lämna 100 mA. Drivsteget har även ett skydd mot att generera VPP samtidigt som RE 2 lämnar 5 V.

VPP kontrollen består av en Zenerdiodkoppling som avläses som en TTL nivå med signalen HVON. Denna kan läsas som kontrollportens databit 0. Om den är hög ligger VPP på minst 18 V, om den är låg på mindre än 16 V.

2.7. Spänningsstyrning 0 / 5 / 12 V

Denna styrning sköts av kontrollportens databitar 0 och 1. Denna koppling är en ren spänningsdelare och kan bara lämna en svag ström, < 50 uA. Om databit 1 är låg lämnar kopplingen 0 V, om den är hög lämnar den antingen 5 (databit 0 hög) eller 12 V (databit 0 låg).

Denna spänning har även fått passera genom en operationsförstärkare 301, med spänningsförstärkning +1, för att erhålla en strömstarkare reglering av 0 / 5 V (15 mA). 12 Voltsläget används inte efter OP:n.

2.8. Spänningsstyrning 5 / 12 V

Denna sker med en spänningsdelare och en emitterföljare, och styrs av kontrollportens databit 6. En låg databit ger spänningsnivån 12 V, och en hög ger 5 V. Emitterföljaren kan lämna 5 mA.

1. TESTNING AV KORTET

Inspektera kortet map lödningar, kretstyper, felvända komponenter och f8rrankring av eventuella tr8dar. St8ll in adress E8 p8 kortet:

S W 1

OFF	OFF	OFF	ON	OFF	ON	xx
-----	-----	-----	----	-----	----	----

Sp8nningss8tt kortet (t ex i en METRIC 8). Det anv8nda kraftaggregatet m8ste t8la kortslutning !

Anslut en voltm8tare till testpunkt 2 eller annan l8mplig jordpunkt p8 kortet.

G8r reset. Kontrollera att 5 Volt finns.

Kontrollera f8ljande sp8nningar:

a)	REG 3	VIN:	29.5 - 30.5 V
b)	REG 2	VOUI:	- 5.25 - - 4.75 V
c)	OP 1	Pin 7	+ 12 V (fr8n bussen)
d)	OP 1	Pin 4	- 12 V (fr8n bussen)
e)	Q 1	Emitter	11.5 - 12.5 V

Kontrollera med ohmm8tare att det inte finns n8gon kortslutning mellan :

a)	RE 3	Pin 7 och Pin 1
b)	RE 3	Pin 8 och Pin 14

 H8r delar sig anvisningarna, har du tillg8ng till ett CP/M system g8r enligt de icke asterisk m8rkte punkterna. Har du enbart tillg8ng till en h8rdvarudebugger, f8lj instruktionerna fr8n punkt 7*.

7

Kör igång programmet PROGTRIM. Detta görs om man skriver PROGTRIM <cr>. Programmet ger en text och frågar om kortadressen, som nu borde vara inställd på E8. Svara E8.

Kontrollera följande spänningar:

a)	J 1	Pin 14	4.75 - 5.25 V
b)	J 1	Pin 3	4.75 - 5.25 V

Justera med R 22 in spänningen i TP3 till 21.4 V. Lås R 22 !

8

Tryck på <cr>. Justera med R 23 in spänningen i TP3 till 26.0 V. Lås R 23 !

7*

Starta hårdvarudebuggern. Skriv data FEH till port E8H och data 80H till port E9H.

Kontrollera följande spänningar:

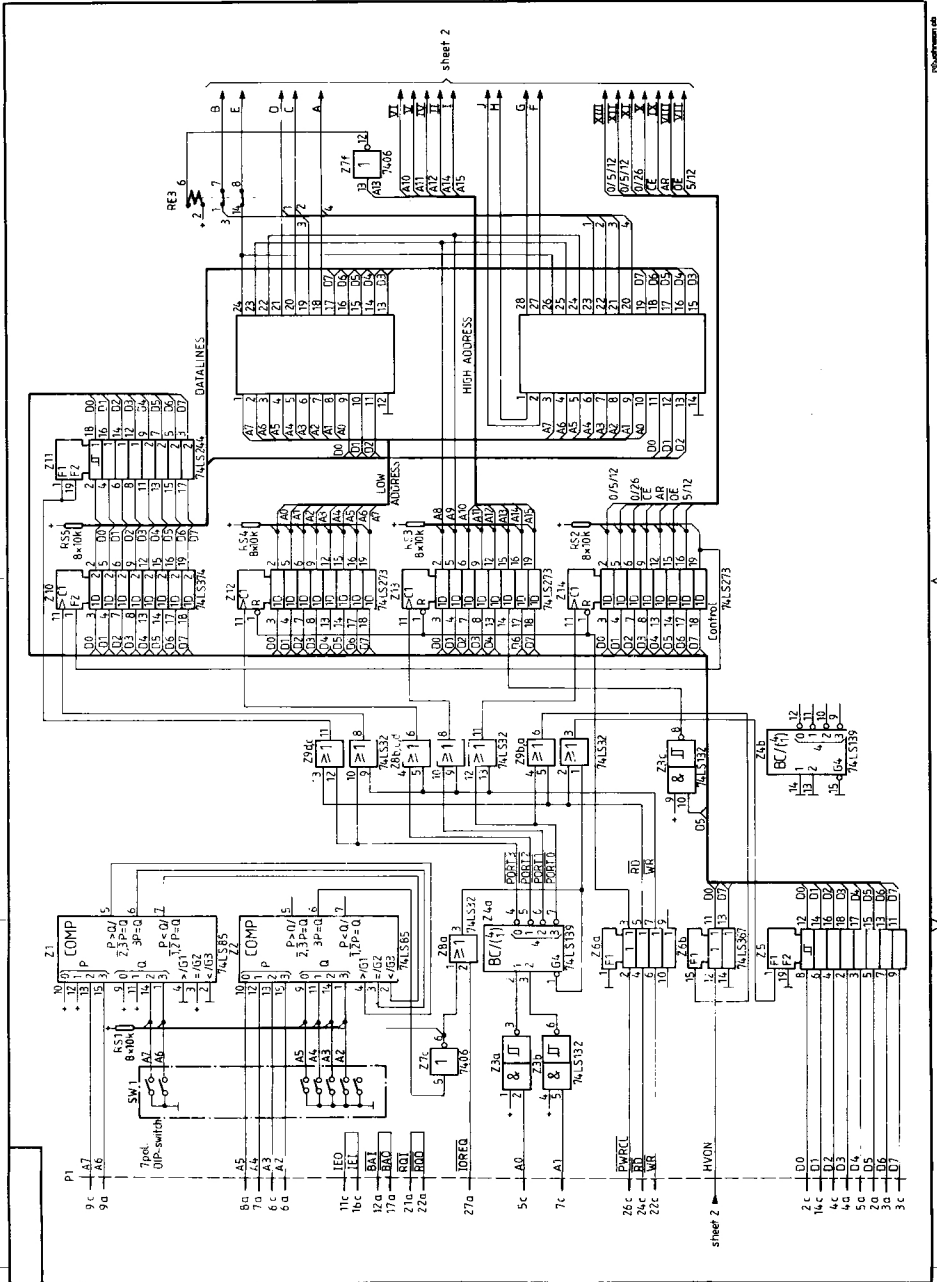
a)	J 1	Pin 14	4.75 - 5.25 V
b)	J 1	Pin 3	4.75 - 5.25 V

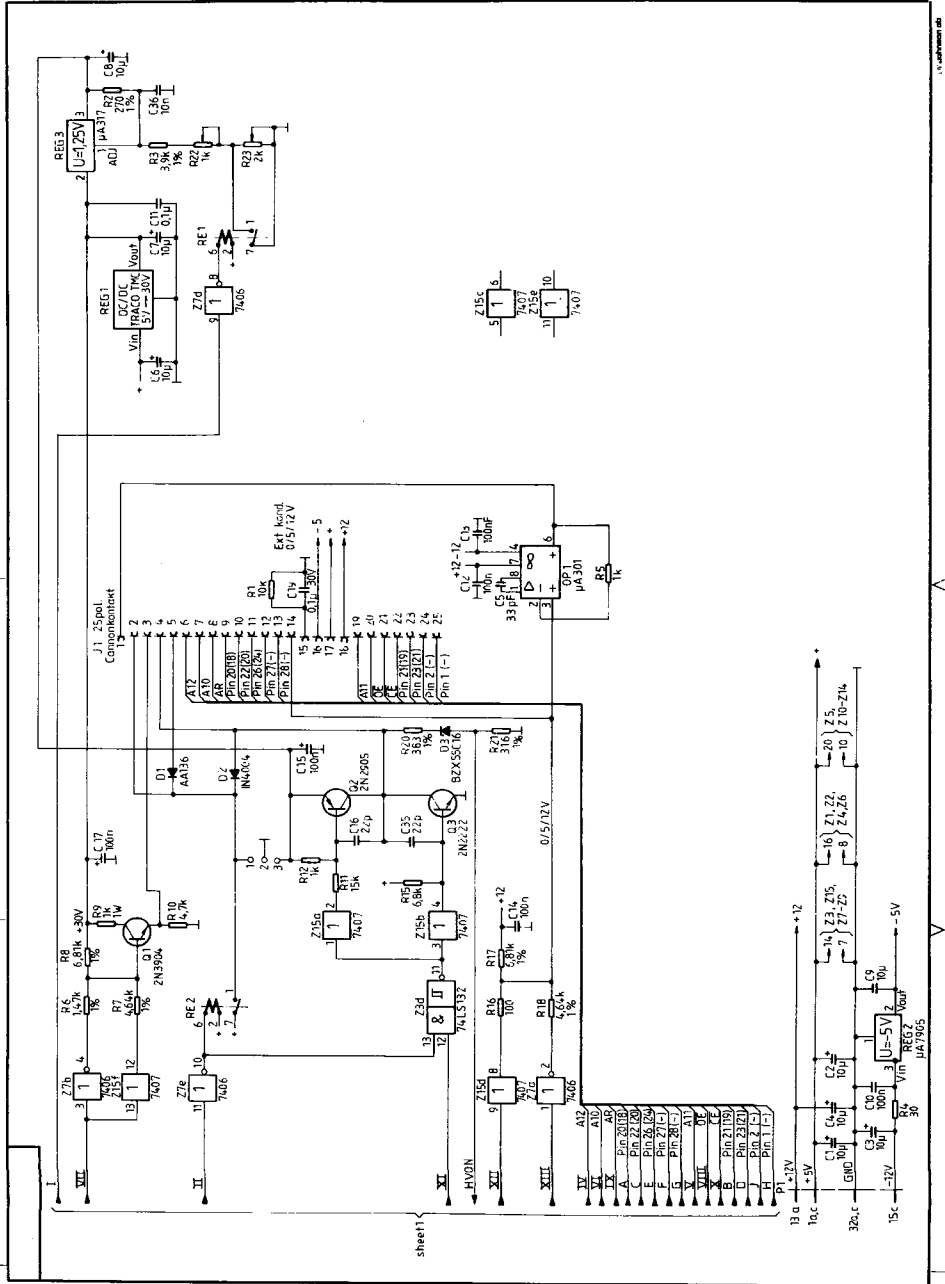
Justera med R 22 in spänningen i TP3 till 21.4 V. Lås R 22 !

8*

Skriv data 00H till port E9H. Justera med R 23 in spänningen i TP3 till 26.0 V. Lås R 23 !

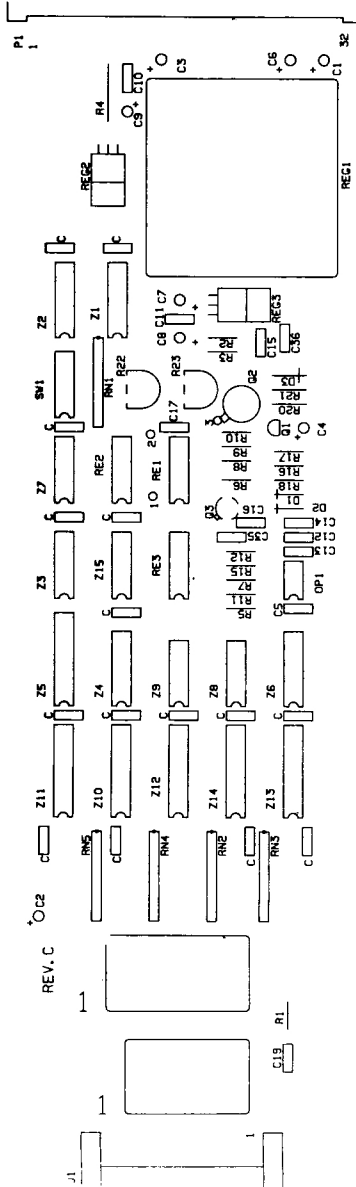
4. PRINCIPSCHEMA





14-079999-02

5. KOMPONENTPLACERING



6. KOMPONENTFÖRTECKNING

Följande komponentförteckning gäller för M-PROG Rev B2 och C.

IC

--

No	Typ	Beskrivning
Z1	74LS85	4 bit magnitude comparator
Z2	74LS85	4 bit magnitude comparator
Z3	74LS132	Quad NAND with Schmit trigger input
Z4	74LS139	Dual 2 to 4 line decoder
Z5	74LS245	Octal bus transceiver
Z6	74LS367	Hex bus drivers
Z7	7406	Hex inverters with OC
Z8	74LS32	Quad OR
Z9	74LS32	Quad OR
Z10	74LS374	Octal D-type Flip-flops 3-state
Z11	74LS244	Octal buffers
Z12	74LS273	Octal D-type Flip-flop
Z13	74LS273	Octal D-type Flip-flop
Z14	74LS273	Octal D-type Flip-flop
Z15	7407	Hex drivers with OC

Spänningsreglering

REG 1 TRACO DC/DC-konverter 5 V till +/- 15 V, 120 mA
 REG 2 uA 7905 (TO-220), -5 V regulator
 REG 3 LM 317 (TO-200), justerbar spänningsregulator

Transistorer

Q1 2N3904
 Q2 2N2905 (TO-5)
 Q3 2N2222

Dioder

D1 AA136
 D2 1N4004
 D3 BZX55C16

Resistorer

(1/8 W där ej annat angivs)

R1	10 k	5 %	
R2	261	1 %	
R3	3.83 k	1 %	
R4	30	5 %	(1/2 W)
R5	1 k	5 %	
R6	1.47 k	1 %	
R7	4.64 k	1 %	
R8	6.81 k	1 %	
R9	1 k	5 %	(1 W)
R10	4.7 k	5 %	
R11	15 k	5 %	
R12	1 k	5 %	
R13	--		
R14	--		
R15	6.8 k	5 %	
R16	100	5 %	
R17	6.81 k	1 %	
R18	4.64 k	1 %	
R19	--		
R20	323	1 %	
R21	316	1 %	
R22	1k		Kapslad trimpotentiometer
R23	2k		Kapslad trimpotentiometer

Kondensatorer

C1	10 uF	10 V	(Tantal)
C2	10 uF	10 V	(Tantal)
C3	10 uF	35 V	(Tantal)
C4	10 uF	35 V	(Tantal)
C5	33 pF	30 V	
C6	10 uF	35 V	(Tantal)
C7	10 uF	35 V	(Tantal)
C8	10 uF	35 V	(Tantal)
C9	10 uF	10 V	(Tantal)
C10	100 nF	35 V	(Wima MKS2)
C11	100 nF	35 V	(Wima MKS2)
C12	100 nF	35 V	(Wima MKS2)
C13	100 nF	35 V	(Wima MKS2)
C14	100 nF	35 V	(Wima MKS2)
C15	100 nF	35 V	(Wima MKS2)
C16	22 pF		Keramisk

C17	100 nF	35 V	(Wima MKS2)	
C18	--			
C19	100 nF	35 V	(Wima MKS2)	
C20 - C34	100 nF	35 V	(Wima MKS2)	Avkopplings-
	kondensatorer, en vid varje IC.			
C35	22 pF		Keramisk	
C36	10 nF	35 V	(WIMA MKS2)	

Motståndsnät

SIL 1	7 *	10 k
SIL 2	8 *	10 k
SIL 3	8 *	10 k
SIL 4	8 *	10 k
SIL 5	8 *	10 k

Reläer

RE 1	Clare PRMA 1A05B
RE 2	Clare PRMA 1A05B
RE 3	Clare PRMA 2A005A

OP amp

OP 1	uA301
------	-------

Socklar, kontakter och diverse

	Textool 24 pin nollkraftssockel
	Textool 28 pin nollkraftssockel
J1	Cannon DS-25P
P1	Kortkontakt Europadon 96/64
SW1	DIP switch 6 pol
	Mönsterkort

7. TEKNISKA DATA

Strömförbrukning + 5 V < 1000 mA
+ 12 V < 50 mA
- 12 V < 50 mA

Storlek Förlängt Europakort, 100 * 320 mm

Buss Metric

Kortadresser Internt används fyra. Man anger den lägsta portadressen av ett sådant fyrstal. De tillåtna kortadresserna är 64 stycken (00H, 04H, ..., FCH.)

APP 1. BYGLINGSKONTAKTEN FÖR OLIKA EPROMAR

METRIC CARD PROG kan hantera de flesta EPROM typerna. Till varje typ behöver man endast iordningställa en 25 polig bygglingskontakt, ("Cannonkontakt").

Siffrorna anger stiftnumreringen i den 25 poliga kontakten. Värden inom parentes anger antalet matningsspänningar.

1 KB 2708 (3) 2758/2508 (1)

4 - 9	2 - 23
10 - 14	5 - 17
11 - 17	8 - 22
16 - 23	9 - 21
18 - 22	10 - 20
	11 - 17

2 KB 2716 (3) 2516/2716 (1)

3 - 11	2 - 23
4 - 9	5 - 17
7 - 10	7 - 22
16 - 23	9 - 21
18 - 22	10 - 20
	11 - 17

4 KB Intel 2732 Texas 2532 Motorola 68732

2 - 10	2 - 23	1 - 5
2 - 15	5 - 17	2 - 10
7 - 22	7 - 22	7 - 22
9 - 21	9 - 19	8 - 23
11 - 17	10 - 21	9 - 19
19 - 23	11 - 17	11 - 17

8 KB Intel 2764 Texas 2564 Mostek 2764 Motorola 68764

2 - 25	2 - 25	2 - 10	1 - 5
5 - 17	5 - 13	6 - 24	2 - 10
6 - 24	6 - 23	7 - 22	6 - 23
7 - 22	7 - 22	9 - 21	7 - 22
8 - 12	8 - 10	13 - 17	9 - 19
9 - 21	9 - 19	19 - 23	11 - 17
10 - 20	12 - 21		
13 - 17	12 - 24		
19 - 23	13 - 17		

16 KB Intel 27128

- 2 - 25
- 5 - 17
- 6 - 24
- 7 - 22
- 8 - 11
- 9 - 21
- 10 - 20
- 12 - 14
- 13 - 17
- 19 - 23

```
*****  
*                                     *  
*           M - P R O G               *  
*                                     *  
*****
```

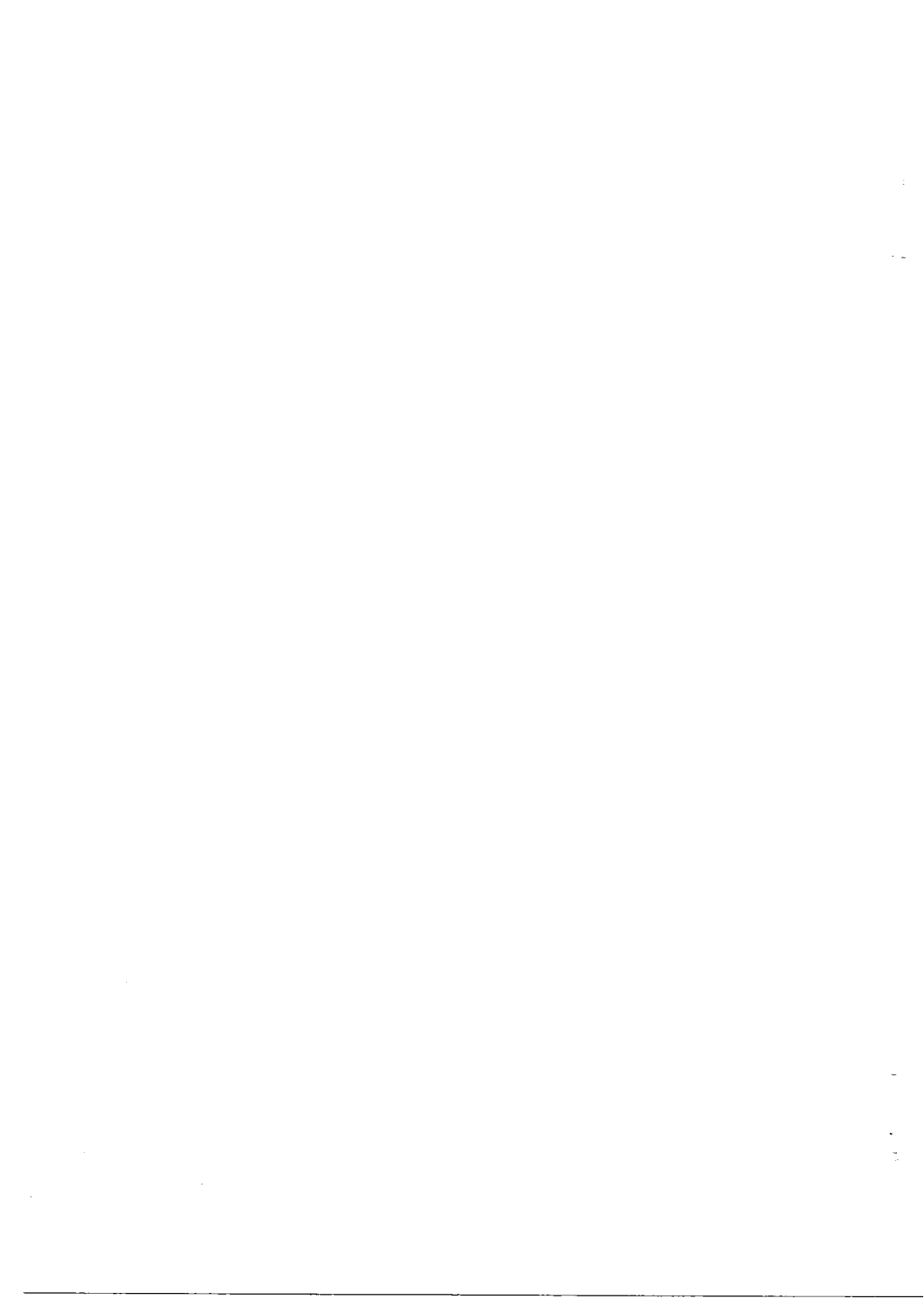
EPROM PROGRAMMERARE UNDER CP/M *)

Programvarumannual

Revision B5

Copyright Scandia Metric AB

*) CP/M is a registered trademark of Digital Research



INNEHÅLL

1.	INLEDNING	3
2.	HANDHAVANDEBESKRIVNING	4
2.1	Allmänt	5
2.2	Detaljerad beskrivning	7
2.2.1.	C kommandot (Check)	7
2.2.2.	F kommandot (CPU frequency)	8
2.2.3.	H kommandot (Help)	8
2.2.4.	L kommandot (List)	8
2.2.5.	P kommandot (Parameter help)	8
2.2.6.	P1 kommandot (System address)	9
2.2.7.	P2 kommandot (Buffer offset)	9
2.2.8.	P3 kommandot (EPROM offset)	9
2.2.9.	P4 kommandot (Length)	10
2.2.10.	P5 kommandot (Card address)	10
2.2.11.	P6 kommandot (Inv address)	10
2.2.12.	P7 kommandot (Inv data)	10
2.2.13.	P8 kommandot (Source)	10
2.2.14.	P9 kommandot (Destination)	11
2.2.15.	Q kommandot (Quit)	12
2.2.16.	R kommandot (Restart)	12
2.2.17.	V kommandot (Verify)	12
2.2.18.	X kommandot (Execute)	12
2.2.19.	/A kommandot (Toggle AR)	15
2.2.20.	/G kommandot (Get parameters)	15
2.2.21.	/L kommandot (Load source)	15
2.2.22.	/S kommandot (Save parameters)	15
2.3	Exempel	16
3.	SYSTEMBESKRIVNING	21
4.	PROGRAMBESKRIVNING	26
App 1.	BYGLINGSKONTAKTEN FÖR OLIKA EPROMAR	27
App 2.	INTRIMNING AV KORTET	29
App 3.	INSTÄLLNING AV KORTADDRESS	30
App 4.	INSTALLATION AV PROGRAMMET	31

1. INLEDNING

M-PROG är ett program som kan läsa och programmera EPROMar på METRIC CARD PROG. Det körs under CP/M på METRIC 8 eller METRIC 85.

Programmet kan hantera de flesta av de vanligaste EPROMarna på marknaden, från 2708 till 27128, när det används tillsammans med METRIC CARD PROG Rev B och högre.

Varje EPROM typ behöver en användartillverkad byglingskontakt, som ansluts på ytterändan av kortet. EPROMarna sätts fast i en av två nollkraftssocklar.

Partiell programmering, offset-programmering, kopiering, dumpning, med eller utan inverterade data och/eller adresser, är exempel på vad M-PROG kan utföra.

M-PROG klarar källkod både från Binär- och Intel-hexfil, förutom möjligheten att läsa EPROM:ar.

2. HANDHAVANDEBESKRIVNING

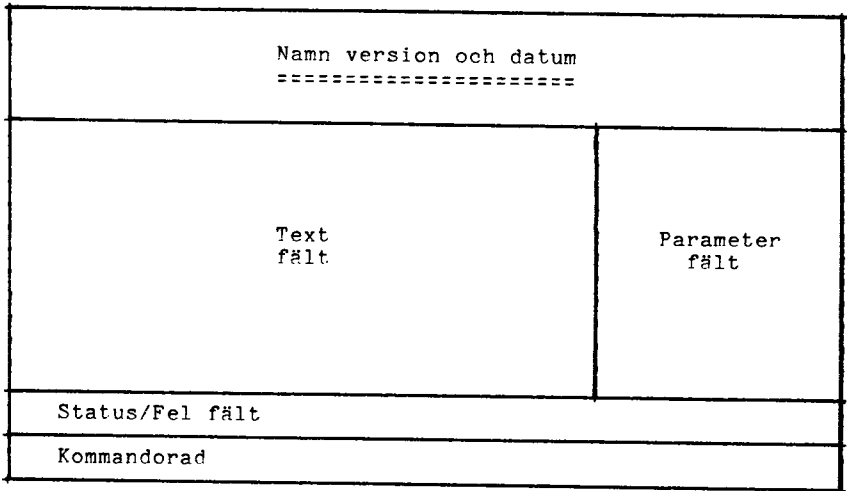
Programmet börjar med en begäran om CPU-frekvens för att de tidsfördröjningar som förekommer i programmet skall få rätt längd. Detta är viktigt eftersom en felaktig inställning i värsta fall kan få till följd att ett EPROM förstörs.

Programmet rör sig hela tiden runt en huvudmeny. I denna specificerar man sina önskemål, vilka lagras i parameterfältet. M-PROG håller alltid en kopia av denna information på bildskärmen. Parameterfältet kan lagras som en fil, och återkallas.

Tvekar man vid någon punkt, så finns det en inbyggd hjälpfunktion till hands.

2.1 Allmänt

M-PROG visar alltid en skärm uppdelad i 5 fält, enligt nedan.



Figur 1. Bildskärmens uppdelning.

Det översta fältet är alltid detsamma, med programnamn, versionsnummer och datum.

Textfältet ger menyer, hjälpinformation, minneslistningar, etc.

Parameterfältet innehåller parametrarna så som de för ögonblicket ser ut. Detta fält innehåller från början globalt definierade defaultvärden. Antalet parametrar är 10 - 12, beroende på vilken uppgift som specificerats och EPROM-typ(er).

I Status/Fel-fältet visas felmeddelanden och andra resultat.

På kommandoraden matas samtliga kommandon in.

Följande EPROM:ar (eller motsvarande) kan hanteras

Typ	Programmeringstid (ungefärligt)	
Texas 2508	50 s	
Texas 2516	100 s	
Texas 2532	200 s	
Texas 2564	400 s	
Intel 2708/3	100 s	
Intel 2758	50 s	
Intel 2716/1	100 s	(1-spänning)
Texas 2716/3	200 s	(3-spänning)
Intel 2732/2732A	200 s	
Intel 2764	400 s	
Intel 27128	800 s	
Motorola 68732	200 s	
Motorola 68764	400 s	

Dessa är de ungefärliga tiderna för att programmera hela minnet. Alla enspännings EPROM:ar kan programmeras partiellt.

2.2 Detaljerad beskrivning

Moder

=====

Programmet befinner sig alltid i en av sex moder: MENU, LIST, VERIFY, CHECK, EXECUTING eller LOAD SOURCE. Detta visar det nuvarande grundtillståndet hos programmet. Under specificeringen av parametrarna befinner man sig i MENU moden. Om man vill lista eller ändra i den interna bufferten, görs det i LIST moden. Om man vill kontrollera att ett EPROM är raderat hamnar man i CHECK moden och en verifiering sker i VERIFY moden. Vid programmering eller dumpning av koden på fil befinner man sig i EXECUTING moden. Slutligen befinner man sig i LOAD SOURCE moden vid inläsning av källkoden till den interna bufferten, vilket dock enbart syns på skärmen om man läser in en binärfil som är större än den interna bufferten.

En mode kan inte väljas direkt, den sätts i och med att man väljer ett utav nedanstående kommandon.

Kommandon

=====

Alla körningar med M-PROG består i att man anger parametrar till parameterfältet eller väljer aktiviteter, genom att man skriver ett eller flera kommandon på kommandoraden. Parameterfältet kan lagras på en datafil och senare återkallas.

Flera kommandon kan ges på en och samma rad, åtskilda av mellanslag. Kommandoraden kan editeras med vanliga CP/M editeringskommandon.

Notera att alla numeriska inmatningar sker med HEXADECIMALA tal.

2.2.1. C kommandot

(Check). Kontrollerar om destinations EPROM:en är raderad. Resultatet visas i Status/Fel fältet. Kommandot kommer dels att fråga om rätt bygglingskontakt är isatt samt om EPROM:en är isatt, därefter kontrolleras innehållet. Resultat blir antingen att EPROM:et är raderat eller inte är det. Om destinationen är en fil ges ett meddelande. Notera att systemet kommer att klassificera ett icke isatt EPROM som raderat. Medans C kommandot är aktivt befinner man sig i CHECK moden.

2.2.2. F kommandot

(select Frequency). Kommandot påverkar längden av en fördröjningsloop. Om du använder en 2.5/2.66 MHz CPU, ange 2, om du använder med en 4 MHz CPU ange 4. METRIC 8 har normalt med en CPU-frekvens på 4 MHz och METRIC 85 2.66 MHz.

Detta kommando har samma verkan som det svar man ger på frågan om CPU-frekvens vid programstarten. Det är alltså möjligt att ändra tidsfördröjningen om man angav fel värde vid programstarten.

2.2.3. H kommandot

(Help). Syntax H(ELP). Ger en övergripande hjälptext, som innehåller övriga tillåtna kommandon. Till varje P kommando kan man erhålla mera information under körning, se P kommandot.

2.2.4. L kommandot

(List). Syntax L=adress. Observera att adress beror på hur parameter 1 är satt. Tar programmet in i LIST moden, samt visar 128 bytes av den interna bufferten i textfältet på skärmen, både i hexadecimal och ASCII representation. När man befinner sig i LIST moden, kan man begära att få se nästa (N) eller föregående (P) 128 bytes av den interna bufferten. Vill man modifiera koden i bufferten anger man C. Om man vill ändra någon byte uppmanas man att ge adress=data, där adressen ska bestå av 2 eller 4 hexadecimala siffror, och data av 2 hexadecimala siffror. Man lämnar LIST moden med Q. Medans man håller på att undersöka eller ändra den interna bufferten befinner man sig i LIST moden.

2.2.5. P kommandot

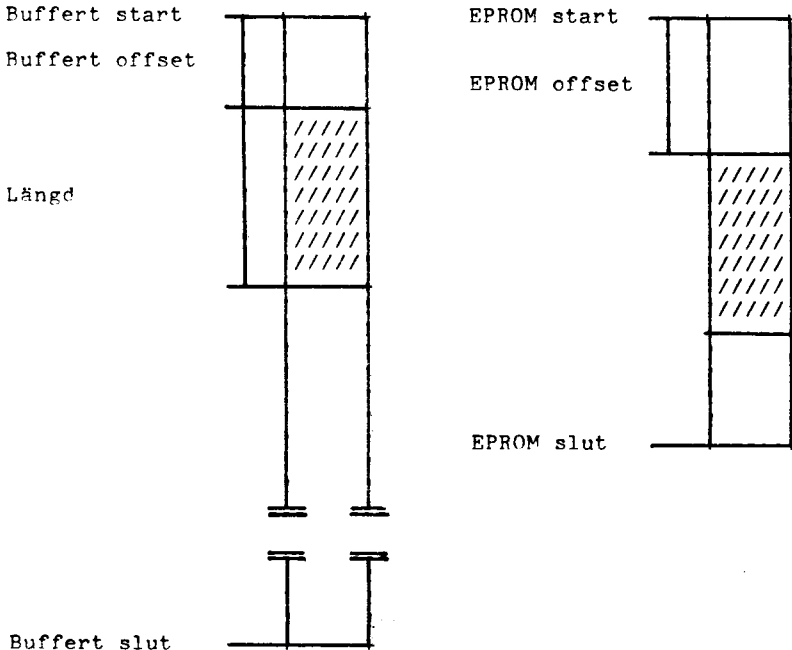
(Parameter). Syntax Pn eller Pn=parameter, där n är en siffra 1..9. Se nedan vid de 9 olika kommandona. Detta är kommandot för att sätta parametrar i parameterfältet, eller för att få mera information om en parameter. Specificerar man enbart siffran får man en hjälptext utskrivna. Anger man dessutom likhetstecknet och någon parameter, kontrolleras parameterns riktighet, och därefter uppdateras parameterfältet.

2.2.6. P1 kommandot

(Parameter 1 help). Parameter 1 är systemadressen, som enbart ändrar den logiska representationen av adresserna på bildskärmen. Om man vill programmera ett EPROM ifrån adress 8000H, kan man sätta system adressen till 8000H för att få riktiga adresser återgivna i menyerna och vid listning.

2.2.7. P2 kommandot

(Parameter 2 help). Detta värde anger offseten från början av den interna bufferten. Den första byte som kommer att sändas ut till EPROM:et blir systemadressen + buffertoffset. Om man ex.vis har 2000H byte binärkod, och EPROM:et rymmer 1000H bytes, behöver man 2 stycken EPROM:ar för att kunna programmera hela binärkoden. Den första EPROM:en kommer då att programmeras med buffertoffset = 0, och den andra med samma buffertinnehåll, men buffertoffseten satt till 1000H. Se figur 2.



Figur 2. Inverkan av offsetvärden \ll 0.

2.2.8. P3 kommandot

(Parameter 3 help). Detta är den andra typen av offset. Med EPROMoffset anger man hur långt in i EPROM:et man vill att första byten skall hamna. Se figur 2.

2.2.9. P4 kommandot

(Parameter 4 help). P4 specificerar hur många bytes man vill sända ut av den interna bufferten, till EPROMet eller till filen. Om destinationen är ett EPROM kan detta antal högst vara lika stort som antalet bytes i EPROMet. Det kan vara mindre om den specificerade delen av källkoden är kortare, eller om man använder sig av en EPROM-offset (P3 kommandot). Anger man orimliga värden, kommer programmet att protestera vid exekveringen. Se figur 2.

2.2.10. P5 kommandot

(Parameter 5 help). Här specificeras METRIC CARD PROG kortets lägsta I/O address. I/O adressen måste vara jämnt delbar med 4. Rekommenderat defaultvärde är E8H. Använder man en annan kortadress, måste man se till att denna inte kolliderar med någon annan adress, se efter i dokumentationen på datorn. Möjliga värden är 00H, 04H, 08H, 0CH, ..., F8H, FCH.

För inställning av kortadressen, se App 3.

2.2.11. P6 kommandot

(Parameter 6 help). Syntax P6=Y/J/N. Specificerar man P6 till ett ord som börjar med Y eller J, kommer P6 att sättas till Yes, anger man P6 till ett ord som börjar med N, sätts P6 till No. Med denna parameter kan man få adressbussen inverterad vid programmeringen. Detta är användbart om målsystemet använder inverterad adressbuss.

2.2.12. P7 kommandot

(Parameter 7 help). Syntax P7=Y/J/N. Specificerar man P7 till ett ord som börjar med Y eller J, kommer P7 att sättas till Yes, anger man P7 till ett ord som börjar med N, sätts P7 till No. Med denna parameter kan man få databussen inverterad vid programmeringen.

2.2.13. P8 kommandot

(Parameter 8 help). Här anger man "källkoden", som kan ligga i ett EPROM eller vara en binärfil eller Intel hexfil. Om man specificerar "källkoden" som en sträng som är en av nedanstående typer (egentligen en delsträng av nedanstående typer), anses det vara ett EPROM, alla övriga strängar klassas som filer.

Tillåtna EPROM typer är (eller motsvarande):

2508	2708/1	2708/3	i2758
2516	2716/1	2716/3	8516
2532	2732	2732A	8532
68732			
2564	i2764	MS2764	68764
i27128			

där i = Intel, MS = Mostek.

En binärfil kan vara av godtycklig storlek. Om filen visar sig vara längre än den interna bufferten (cirka 45 KByte), så får man specificera vilken del av filen som skall hamna i den interna bufferten. Denna fråga kommer först vid inläsningen av filen, dvs vid /L kommandot.

En Intel Hex Fil måste ha suffixet .HEX och får inte spänna över större adressområde än den interna bufferten.

"Källkoden" läses in till den interna bufferten ENBART enligt parameter 8 via kommandot /L. Koden läggs alltid från början av den interna bufferten.

Ett filnamn får inte innehålla dollartecknet.

2.2.14. P9 kommandot

(Parameter 9 help). Med detta kommando specificeras vart man vill sända den interna bufferten. Koden kan antingen sändas till ett EPROM, vanlig programmering, eller till en binärfil, EPROM dump.

För att M-PROG skall sända ut den interna bufferten till ett EPROM måste man ange P9 som en sträng enligt vad som står vid P8 kommandot.

Koden kommer att hamna i en fil, om P9 är satt till en sträng skild ifrån de strängar som angivits vid P8 kommandot.

Bufferten sänds ut under exekveringsfasen, startad med X kommandot, och använder alla tillämpliga parametrar.

Ett filnamn får inte innehålla dollartecknet.

2.2.15. Q kommandot

(Quit). Detta är det kontrollerade sättet att lämna programmet.

2.2.16. R kommandot

(Restart). Kommandot sätter tillbaka parametrarna i parameterfältet till de värden de hade vid programstarten.

2.2.17. V kommandot

(Verify). Detta kommando jämför innehållet i den interna bufferten med innehållet i destinationen. Om destinationen är ett EPROM, kommer programmet att fråga dels om en byglingskontakt är ansluten och om ett EPROM är isatt. Resultatet av jämförelsen visas i Status/Fel fältet och anger totala antalet icke överensstämmande bytes samt hur många av dessa som skulle omöjliggöra en programmering (fatala fel). Bytes som omöjliggör en programmering är en källkodsbyte som innehåller en hög bit och som skall läggas i ett destinations-EPROM som innehåller en låg bit på motsvarande ställe.

Antalet fatala fel är bara ett meningsfullt värde om man avser att programmera ett EPROM. OBSERVERA att verifieringen använder sig av offsetinställningar och eventuella inverteringar. Medans man befinner sig i verifieringsrutinen befinner man sig i VERIFY moden.

2.2.18. X kommandot

(eXecute). Utför programmeringen eller dumpningen till fil.

Destinationen = EPROM Före programmering

Om koden sänds till ett EPROM, visas uppgifter om hur programmeringen kommer att ske, och man får kvittera tre saker;

- * Om man är nöjd med parametervärdena
- * Om man har anslutit byglingskontakten.
- * Om man har satt in ett EPROM

OBSERVERA !
 =====

RÄTT BYGLINGSKONTAKT SKA SÄTTAS I INNAN MAN SÄTTER I EPROM:ET, OCH OMVÄNT NÄR MAN VILL TA BORT BÄGGE.

BYGLINGSKONTAKT OCH EPROM SKA SÄTTAS I DÅ PROGRAMMET FRÅGAR EFTER ANSLUTEN BYGLINGSKONTAKT RESPEKTIVE ISATT EPROM.

EPROM:ET SKA VÄNDAS SÅ ATT PINNE 1 ÄR NÄRMEST NOLL-KRAFTSSOCKELNS VIPPARM.

Om de interna testerna går bra, och man svarar tillstyrkande på de tre frågorna, påbörjas programmeringen.

De interna testerna består av en kontroll om de angivna parametrarna är formellt riktiga och passar EPROM typen:

Ett trespänningsmatat EPROM måste programmeras helt och hållet.

Den sista byten man vill programmera måste ligga inom EPROM:ets adressområde.

Den sista byten man vill programmera måste ligga inom den interna buffertens adressområde.

Destinations EPROM:et får inte innehålla en låg bit där man avser att programmera en hög bit. Destinations EPROM:et kan dock innehålla låga bitar på ställen där man avser att programmera låga bitar. Om någon bit i destinations EPROM:et är låg, men programmeringen är möjlig, skrivs en varning ut, men man kan välja att programmera ändå.

Destinationen = EPROM Under programmering

Man kan kontinuerligt följa på skärmen hur stor del av programmeringen som är utförd. En asterisk skrivs för varje 100H bytes som programmerats. För trespännings EPROM:ar visas ett värde som inte direkt ger hur många bytes som är programmerade, men visar "motsvarande" värde eftersom de programmeras på ett annat sätt.

Om programmeringsspanningen inte kommer upp till specificerat värde, kopplas den bort och en felutskrift 'NO PROGRAMMING VOLTAGE' visas. Detta kan orsakas av en upp och ner vänd eller trasig EPROM eller av ett felaktigt PROG kort.

Destinationen = EPROM Efter programmering

Efter programmeringen verifieras alltid innehållet i EPROMet, och totala antalet fel och antalet 'fatala' fel visas.

Ett fel i den totala felsumman anger att en programmerad byte inte blev programmerad korrekt.

Ett sådant fel kan orsakas av:

- * Inget EPROM isatt
- * Felaktig kortadress
- * Trasigt EPROM
- * Inget METRIC CARD PROG kort isatt
- * Ingen spänning påslagen (om man använder expansionslåda).

Ett 'fatal error' är en byte som innehåller åtminstone en programmerad hög bit som kontrolläses som låg bit.

Ett fatalt fel kan orsakas av:

- * Ett trasigt EPROM
- * Felaktig bygglingskontakt
- * Felaktig EPROM-typ

Destination = Fil

Koden kommer att dumpas till en fil. Observera att längden (parameter 4) och buffertoffset (parameter 2) gäller vid fildumpning.

2.2.19. /A kommandot

(toggle Array select). Syntax /A. Detta kommando byter värdet av AR, 'array select', om detta är aktuellt för EPROMet. AR signalen finns endast på EPROM typerna Intel 2758 (vissa) och Motorola 68732.

2.2.20. /G kommandot

(Get parameter file). Syntax /G. Med detta kommando kan man hämta parameteruppställningen från en fil som redan lagrats med /S kommandot, se nedan. Filens namn är underförstått <Parameter9>.PRM. En .PRM fil innehåller enbart parameteruppställningen, inte kod till den interna bufferten.

För att läsa in kod från en fil anger man P8=<Filnamn> /L.

2.2.21. /L kommandot

(Load source). Syntax /L. Detta kommando läser in källkoden till den interna bufferten enligt parameter 8.

2.2.21. /S kommandot

(Save parameter file). Syntax /S. Detta kommando sparar nuvarande parameteruppsättning på en fil, så att de kan återkallas senare. Filens namn blir <Parameter9>.PRM. Denna fil kommer enbart att innehålla parameteruppställningen, inte kod från den interna bufferten.

För att spara koden i den interna bufferten på en fil, ange P9=<Filnamn> X.

2.3 ExempelProgrammering av ett "stort" assemblerprogram

Antag att programmet KINGKONG.COM är ett monitorprogram, länkat till adress 0000H i form av en binärfil och med längden 953H. Detta skall läggas i 2 stycken 2716 EPROM:ar, eftersom ett EPROM inte rymmer hela koden.

Programmering av EPROM nummer 1

Kör igång programmet M-PROG genom att skriva MPROG. Programmet kommer då att fråga efter CPU-frekvens. Normalt anger Du 2.6 för METRIC 85 eller 4 för METRIC 8. Programmet går där-
efter till sin huvudmeny.

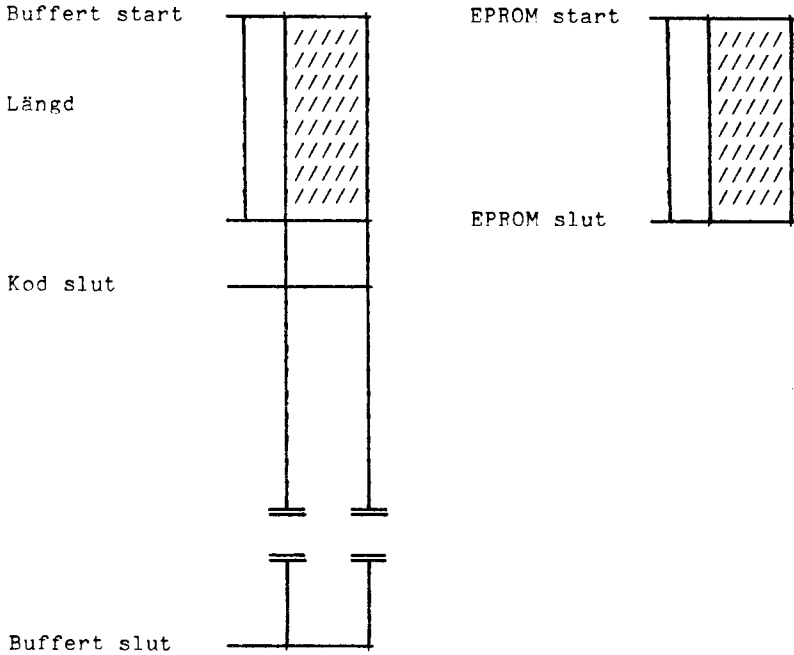
Först specificerar man "source" och hämtar in den. Ange P8=KINGKONG.COM <cr>. Detta specificerar sourcen till en fil som ligger på den inloggade CP/M-driven, med kod i binärformat eftersom suffixet inte är .HEX. P8 är nu ändrat, vilket kan kontrolleras i parameterarean.

Ange sedan /L <cr>. Detta försöker att läsa in kod enligt parameter 8. Om filen finns och läsningen gick bra hamnar koden i den interna bufferten. Kodens storlek kan man läsa på parameterrad 11, som tillkommit i och med /L kommandot.

Om man vill, kan man nu titta på den inlästa koden, genom att ange kommandot L=adress <cr>, där adress är ett värde inom den inlästa koden, dvs 0 <= adress <= 952. Notera att alla numeriska värden är angivna i hexadecimala tal.

Därefter specificerar man vart man vill sända koden. Detta måste i detta fall ske i två omgångar, då ett EPROM inte rymmer hela koden. Ange P9=2716 P4=800 <cr>. Notera att man kan ge flera kommandon på samma rad åtskilda av mellanslag. Detta säger att koden vid exekveringen (se X kommandot) kommer att sändas ut till ett EPROM och ha en längd av 800H bytes (2 KByte). För att specificera en EPROM typ måste man ange en av de strängar som listas på skärmen, om man anger P8 <cr> eller P9 <cr>. Alla övriga strängar klassas som CP/M filer.

Läget är nu följande:



Figur 3. Programmering av EPROM nummer 1, exemplet.

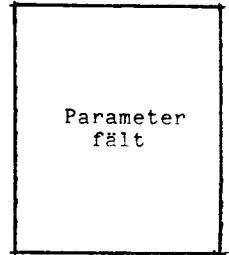
Därefter ger man exekveringskommandot, X <cr>. De angivna parametrarna kontrolleras, och om de är riktiga visas en sammanställning av läget:

==< BLOCK SETTINGS >==

```
The buffer code begins at:      0000

  The buffer code block
    Begins at:                    0000
    Ends at:                      07FF

  The DESTINATION EPROM block
    Begins at:                    0000
    Ends at:                      07FF
```



Figur 4. Parametersammanställning vid programmering.

Denna bild visar hur parametrarna ser ut. Efter denna bild kommer det tre frågor som ska besvaras med "Y" eller "N". Om man svarar "N" avbryts programmeringen.

Först frågas det om värdena är riktiga.

Därefter frågas om rätt byglingskontakt är isatt. Här sätter man i den 25-poliga byglingskontakten, om det inte redan är gjort.

Slutligen frågar programmet om ett EPROM är isatt. FÖRST NU SÄTTER MAN I EPROM:ET. TAG DETTA SOM EN GOD REGEL !

Om EPROM:et inte är raderat och omöjligt att programmera, kommer exekveringen att avbrytas. Om det inte är raderat, men möjligt att programmera, skrivs en varning ut, men man kan välja att fortsätta med att ange Y <cr>.

Om då programmeringen sätter igång, visas följande rad:

```
_____08.00___10.00___18.00__20.00__28.00__30.00__...
```

Allt eftersom programmeringen fortskrider kommer asterisker att skrivas ut för varje 100H (=256) bytes som är programmerade. Exemplet kommer att ge 8 asterisker.

När så hela EPROM:et är programmerat, verifieras programmeringen automatiskt. Resultatet av denna verifiering visar sig i Resultat/Fel fältet.

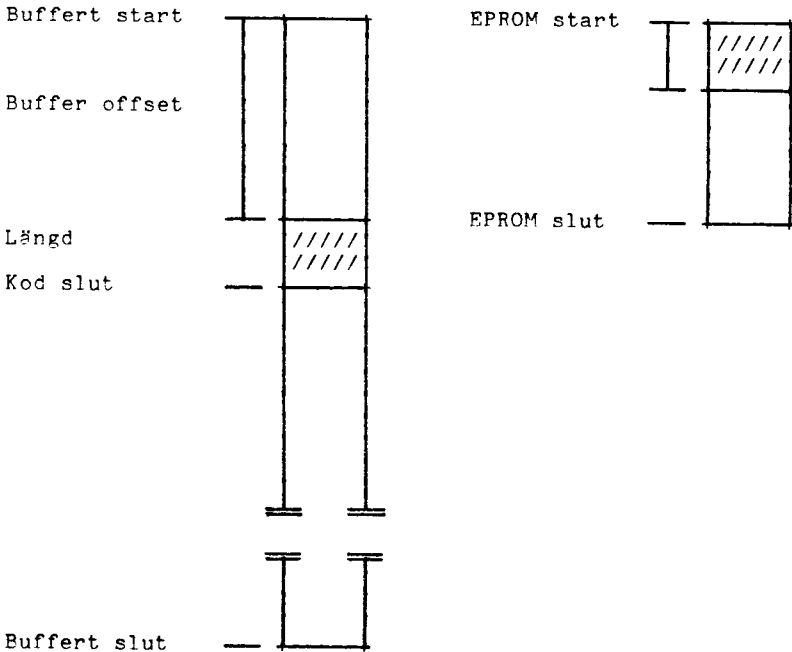
Nu tar man bort EPROM:et ur nollkraftssockeln. Den första programmeringen är klar.

Programmering av EPROM nummer 2

I det angivna exemplet ingick ytterligare en programmering av koden från adress 800H till 952H. För att kunna programmera den delen, får man delvis ange nya värden till parameter-arean.

Koden i den interna bufferten behöver inte läsas in på nytt, den ligger redan där, däremot behöver två parametrar ändras. Ange P2=800 <cr>. Till det andra EPROM:et vill man ju ha koden från den interna buffertens 800H:e byte och framåt. Vidare måste vi korrigera den längd på koden som skall sändas ut till EPROM:et. Ange P4=152 <cr>. Om P4 skulle ha stått kvar som 800H, skulle programmet haprotesterat vid exekveringskommandot.

Programmering nummer 2 kommer att "flytta koden" enligt nedan:



Figur 5. Programmering av EPROM nummer 2, exemplet.

Destinationen är fortfarande ett EPPOM typ 2716 och övriga parametrar oförändrade, ange därför X <cr> och programmera som ovan.

Alla dessa kommandon skulle ha kunnat ges på bara några rader. Först kan man ange:

P8=KINGKONG.COM /L P4=800 P9=2716/1 X <cr>

Sedan tre stycken Y och programmering nummer ett skulle köra igång. På samma sätt kan man ange hela programmering nummer 2 på en rad.

3. SYSTEMBESKRIVNING

För en detaljerad beskrivning av hårdvaran, se M-PROG Teknisk manual.

Anslutningar

EPROM-programmeraren är monterad på ett förlängt Europa-kort, avsett att anslutas till METRIC CARD bussen. Detta gör man i en METRIC 8 genom att ansluta kortet till en utav METRIC 8:s lediga kortplatser. För att komma åt dessa måste bakväggens skyddsplåt tas bort.

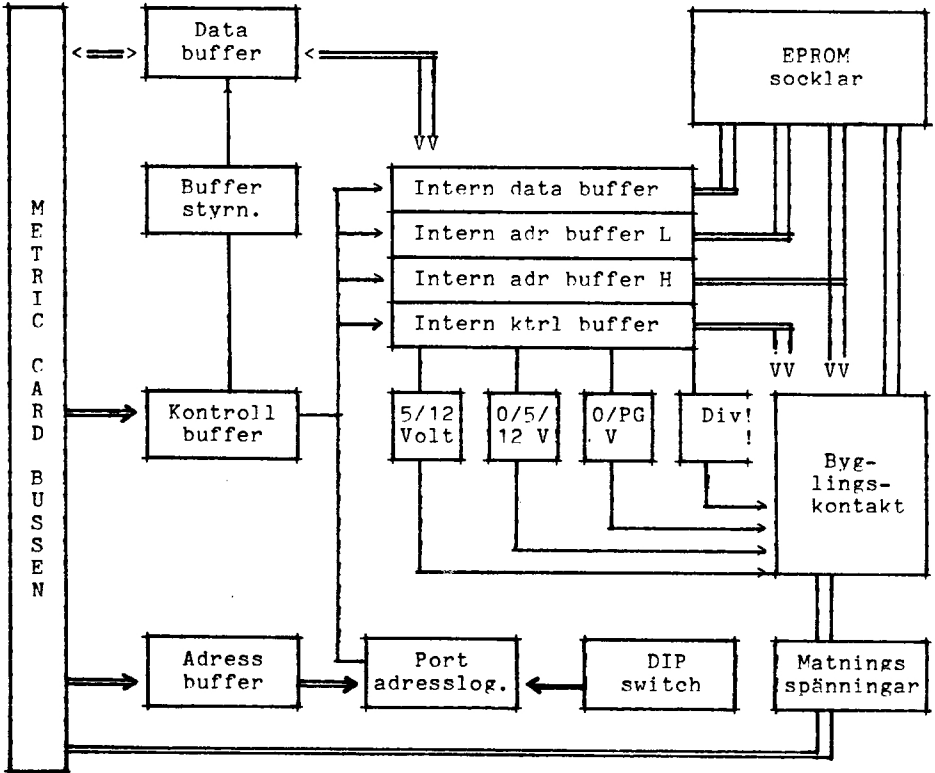
Om man kör programmeraren tillsammans med METRIC 85 så sätts kortet in i en ledig kortplats i expansionslådan EX 85, som i sin tur ska vara kopplad till METRIC 85:an.

Sätt endast i, och ta ur, METRIC CARD kort med spänningen FRÄNSLAGEN !!!

På kretskortet finns två nollkraftssocklar där man pluggar in ett EPROM, samt en 25polig Cannon-kontakt, till vilken man ansluter en EPROM-beroende byglingskontakt. Detta för att underlätta byte mellan olika EPROM-typer.

Blockschema

METRIC CARD PROG är består av följande huvudmoduler:



Figur 6. Blockschema för METRIC CARD PROG (PG står för 21, 25 eller 26 volt). De 4 interna buffrarna är vardera på 8 bitar.

Portar

Kortet använder sig av fyra I/O-portar i följd. Dessa fyra portar kan, genom en DIP-switch, läggas från vilken port-adress som helst som är delbar med 4. Till dessa portar läser man ut information som sedan styr programmeringen. Den första kontrollerar funktionen (programmeringspuls, programmerings-spänning, CS, etc.), den andra håller den höga adressbyten samt reläfunktioner, den tredje den låga adressbyten, och slutligen den fjärde porten håller datat som ska till/från EPROM:et

Basport är ett portnummer jämnt delbart med 4.

Basport + 0	=	Kontroll
Basport + 1	=	Adressport hög
Basport + 2	=	Adressport låg
Basport + 3	=	Dataport

Dataporten är den enda port som är helt bidirektionell. Riktningen på denna port bestäms av bit 8 på kontrollporten.

Kontrollporten kan också läsas av, och innehåller två signifikanta bitar. Om bit 0 är hög, så har programmerings-spänningen kommit upp över 18 Volt. Denna bit testas av programmet vid programmering. Bit 7 är låg på revision P eller högre.

Matningsspänningar

För att kunna programmera de flesta EPROM behövs en programmeringsspanning på 21, 25 eller 26 volt. En sådan hög spänning finns inte i bussen utan "tillverkas" via en DC/DC-konverter från + 5 V till + 15 / - 15 V. Då denna har galvaniskt skild jord, kan man ta ut + 30 V. Denna regleras ner till 26.0 eller 21.4 V, via en justerbar spänningsregulator LM 317. Till de EPROM:ar som behöver 26 V släpps den högre spänningen fram direkt, och till de EPROM:ar som behöver 25 V, får den passera en kiseldiod. Valet mellan 26.0 och 21.4 görs genom att ett relä sluts eller bryts.

- 5 V behövs till vissa kopplingar och den erhålls från - 12 V via en stabiliserad spänningsomvandlare 7905. + 5 och + 12 Volt tas direkt ifrån bussen.

TTL-delen

De bussignaler som utnyttjas belastas bara med en LS-krets.

Med hjälp av två 74LS85, som tillsammans bildar en 6-bits komparator, släpper man bara fram de fyra kortadresser som kortet byglats till. Denna komparator jämför de 6 mest signifikanta adressbitarna. För att sedan få en fullständig avkodning får man leda de 2 minst signifikanta portadressbitarna genom en 2 till 4 avkodare (LS 139). Dessa grindas sedan med IORQ och WR (RD) för att ge ett precist portval.

Databussen buffras med en bidirektionell driver LS245. A0 och A1 har förbättrats genom att de får gå igenom en LS132 Schmitt-grind.

Slutligen buffras RD, WR och PWRCL (Power on clear) med LS367 buffertar.

Ett EPROM har 24 eller 28 pinnar som måste förses med information. Detta görs via 4 stycken latchade 8-bitars I/O-portar. Dataporten (basport + 3) är den enda bidirektionella porten och därför lite annorlunda uppbyggd. Denna del består av 2 kretsar, en LS 374 för flöde till EPROM:et och en LS 244 från EPROM:et. De övriga tre buffrarna är enkelriktade (LS 273). Den höga och låga adressbyten går i princip direkt till EPROM:et. De signaler som logiskt sett skulle varit A13, A14 och A15 styr de tre reläerna. Dock så går adressledningarna A10 och högre via en byglingskontakt eftersom deras anslutning till nollkraftssockeln är EPROM-typberoende. Alla utgångar från de latchade buffrarna har pull-up motstånd.

Analoga delen

Eftersom alla EPROM:ar har någon sorts ovanlig(-a) spänning(ar) som ställer om kretsen i programmeringsläge, måste det finnas möjlighet att med TTL-signaler styra denna (dessa) spänningar.

Det finns tre sådana delar på kortet. En som kan styra en spänning mellan 5 och 12 V; en som kan styra en spänning mellan 0 och 25 (21 eller 26) V, samt slutligen en enhet som kan ge 0, 5 eller 12 V, men endast svaga strömmar.

Den första behövs för att klara 3-spännings 2716. Den andra behövs till samtliga EPROM-ar, 21 V till Intel:s HMOS-EPROM-ar, 25 V till övriga enspänningstyper och 26 V till 3-spänningsvarianterna. Den tredje, slutligen, behövs för 2708. Denna tredje signal har även fått passera genom en operationsförstärkare 301, med spänningsförstärkning +1, för att erhålla en strömstarkare reglering av 0 / 5 V som behövs för Motorolas 68732/68764.

För alla olika varianter på EPROM:ar kan man med dessa styrdelar, via dioder och byglingar, åstadkomma de nödvändiga spänningarna.

3. PROGRAMBESKRIVNING

Här nedan följer blandade kommentarer till programmet.

Programmet stänger av avbrotten med instruktionen DI under den tid som programmeringspulsen ligger på, dvs 50 ms för 1-spännings EPROM:ar, och 1 ms för 3-spännings.

MAN FÅR INTE ANVÄNDA SIG AV WAIT-STATES, DA PROGRAMMET FÖRUTSÄTTER ATT CPU:N GÅR MED 2.66 ELLER 4 MHz.

Programmet använder sig av Z-80 instruktioner av typ LDIR, och är därför körbart enbart på Z-80 CP/M system.

Programmet använder inga egna interrupts.

Checksumman i en Intel Hex fil kontrolleras inte av programmet.

APP 1. BYGLINGSKONTAKTEN FÖR OLIKA EPROMAR

M-PROG tillsammans med METRIC CARD PROG kan hantera de flesta EPROM typerna. Till varje typ behöver man endast iordningställa en 25 polig byglingskontakt, ("Cannonkontakt").

Siffrorna anger stiftnumreringen i den 25 poliga kontakten. Värden inom parentes anger antalet matningsspänningar.

1 KB	<u>2708 (3)</u>	<u>2758/2508 (1)</u>
	4 - 9	2 - 23
	10 - 14	5 - 17
	11 - 17	8 - 22
	16 - 23	9 - 21
	18 - 22	10 - 20
		11 - 17

2 KB	<u>2716 (3)</u>	<u>2516/2716 (1)</u>
	3 - 11	2 - 23
	4 - 9	5 - 17
	7 - 10	7 - 22
	16 - 23	9 - 21
	18 - 22	10 - 20
		11 - 17

4 KB	<u>Intel 2732</u>	<u>Texas 2532</u>	<u>Motorola 68732</u>
	2 - 10	2 - 23	1 - 5
	2 - 15	5 - 17	2 - 10
	7 - 22	7 - 22	7 - 22
	9 - 21	9 - 19	8 - 23
	11 - 17	10 - 21	9 - 19
	19 - 23	11 - 17	11 - 17

8 KB	<u>Intel 2764</u>	<u>Texas 2564</u>	<u>Mostek 2764</u>	<u>Motorola 68764</u>
	2 - 25	2 - 25	2 - 10	1 - 5
	5 - 17	5 - 13	6 - 24	2 - 10
	6 - 24	6 - 23	7 - 22	6 - 23
	7 - 22	7 - 22	9 - 21	7 - 22
	8 - 12	8 - 10	13 - 17	9 - 19
	9 - 21	9 - 19	19 - 23	11 - 17
	10 - 20	12 - 21		
	13 - 17	12 - 24		
	19 - 23	13 - 17		

16 KB Intel 27128

- 2 - 25
- 5 - 17
- 6 - 24
- 7 - 22
- 8 - 11
- 9 - 21
- 10 - 20
- 12 - 14
- 13 - 17
- 19 - 23

APP 2. INTRIMNING AV KORTET

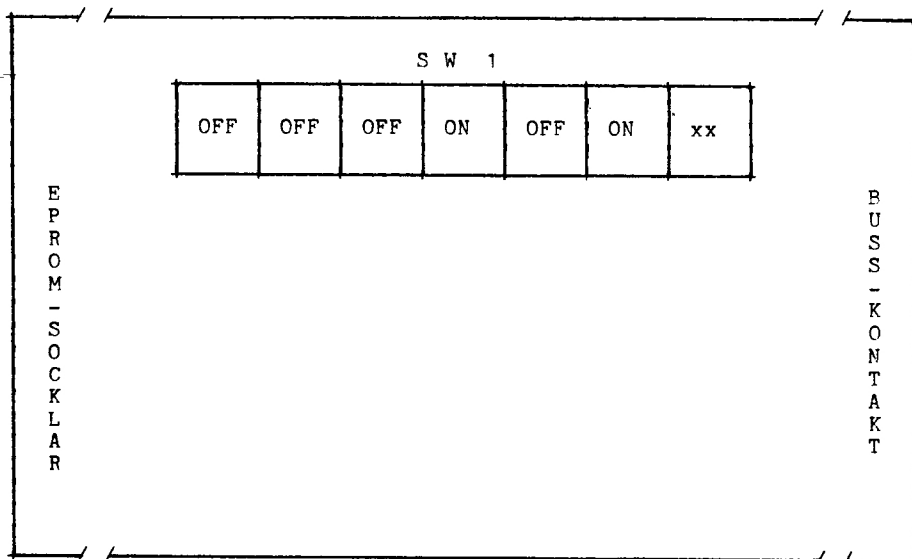
Kortet är vid leverans intrimmat och testat och ska normalt aldrig behöva justeras. Vill man ändå kontrollera detta gör man detta med programmet PROGTRIM om man har tillgång till CP/M, annars med hjälp av en debugger eller liknande (se M-PROG teknisk manual). För att kunna justera programmerings-spänningarna krävs även en voltmeter.

Anslut - polen på voltmeteren till trimpunkt 2, som sitter bredvid potentiometrarna, samt + polen till trimpunkt 3, belägen något längre ner.

Kör igång PROGTRIM. Programmet beskriver när avläsning skall ske, och vilken potentiometer som skall trimmas, etc.

APP 3. INSTÄLLNING AV KORTADRESSEN

Kortadressen är vid leverans inställd på E8H, och ska normalt aldrig behöva ändras. Exempel på kortadress E8H:



Figur 7. Inställning av kortadress E8H.

Kortadressen består av ett 6 bitars tal, med de två minst signifikanta bitarna avkodande en av fyra I/O adresser. Den mest signifikanta biten är den på omkopplaren som sitter närmast nollkraftsocklarna. En bit tolkas som hög om man ställer omkopplaren i läge "OFF" eller "OPEN", en bit tolkas som låg om man ställer omkopplaren i läge "ON" eller "CLOSED".

APP 4. INSTALLATION AV PROGRAMMET

Detta appendix behandlar hur man kan ändra programmets "Form Feed" funktion. Detta behöver dock inte göras då programmet levereras i en form som bör passa alla sorters terminaler.

Vill man ändra detta, krävs en "patchning" i programmet. Denna "patchning" görs med programmet självt. SE TILL ATT DET FINNS EN BACKUP !

Nedan finns en beskrivning i punktform hur denna ändring går till:

- 1) Ta reda på vilket (vilka) tecken terminalen kräver för att göra ett Form Feed, dvs att rensa skärmen och placera cursorn i övre vänstra hörnet. På METRIC 85 görs Form Feed genom att skriva OCH (12D) till bildskärmen.
 - 2) Starta programmet genom att skriva MPROG
 - 3) Läs in programmet till bufferten med kommandot :
P8=MPROG.COM /L <cr>
 - 4) Gå in i LIST-moden :
L=0 <cr>
 - 5) Ta reda på "patchadressen". Denna står som 4 hexadecimala ASCII siffror i programmets byte 4 - 7, där byte 4 är den mest signifikanta. Denna adress kallas nedan ADR.
 - 6) Skriv :
L <ADR>
för att skriva ut bufferinnehållet vid ADR. Detta skall vara:

OD OD OD 20 20 20 20 20 ...
20 20 20 20 20 20 20 20
 - 7) Skriv nu in tecknet (tecknen) för Form Feed med början från ADR (max tre tecken).
För METRIC 85 görs detta med :
C <ADR>=OC <cr>
 - 8) Gå ur List-moden.
 - 9) Skriv det modifierade bufferinnehållet till filen MPROG.COM genom kommandot :
P9=MPROG.COM P4=2600 X <cr>
Y <cr>
- OBS !! SE TILL ATT DET FINNS EN BACK-UP !!
- 10) Gå ur programmet med :
Q <cr>
och testa den nya versionen.

LÄSARENS KOMMENTARER

Dina synpunkter på denna dokumentation är viktig för oss; Det hjälper oss till förbättringar och kommer dig till gagn i framtiden. Var vänlig att besvara frågorna i formuläret och sända det till oss.

Tack.

Ditt namn:	Kvalitet på dokumentationen:		
Företagets namn:		Bra	Varken/eller
Adress:	Tekniskt
Postnr och postadress:	Uppbyggnad
Dokumentets namn:	Fullständighet
Hårdvarukonfiguration:			
.....			
Aktuell programvara:	Vad skulle förbättra materialet ?		
.....			
Uppfyller dokumentationen dina behov ? Ja Nej			
Om Nej, varför inte?			
.....			
.....			
Hur använder du denna dokumentation ?	Övriga kommentarer eller förslag:		
... Som introduktion till ämnet.		
... Som referens-handbok.		
... Som handledning vid utbildning.	Felaktigheter i dokumentationen:		
		

Frånkeras ej.
Mottagaren
betalar
portot.

SCANDIA METRIC AB
Mät och Industrisystem

Svarspost
Kundnummer 290 28016
171 25 SOLNA

100

SCANDIA **METRIC** **AB**
INCENTIVE-GRUPPEN

BANVAKTSVÄGEN 20, BOX 1307, 171 25 SOLNA, TEL 08/82 04 00
REGIONSKONTOR: ÅBACKSGATAN 6, 431 37 MOLNDAL, TEL 031/20 06 50