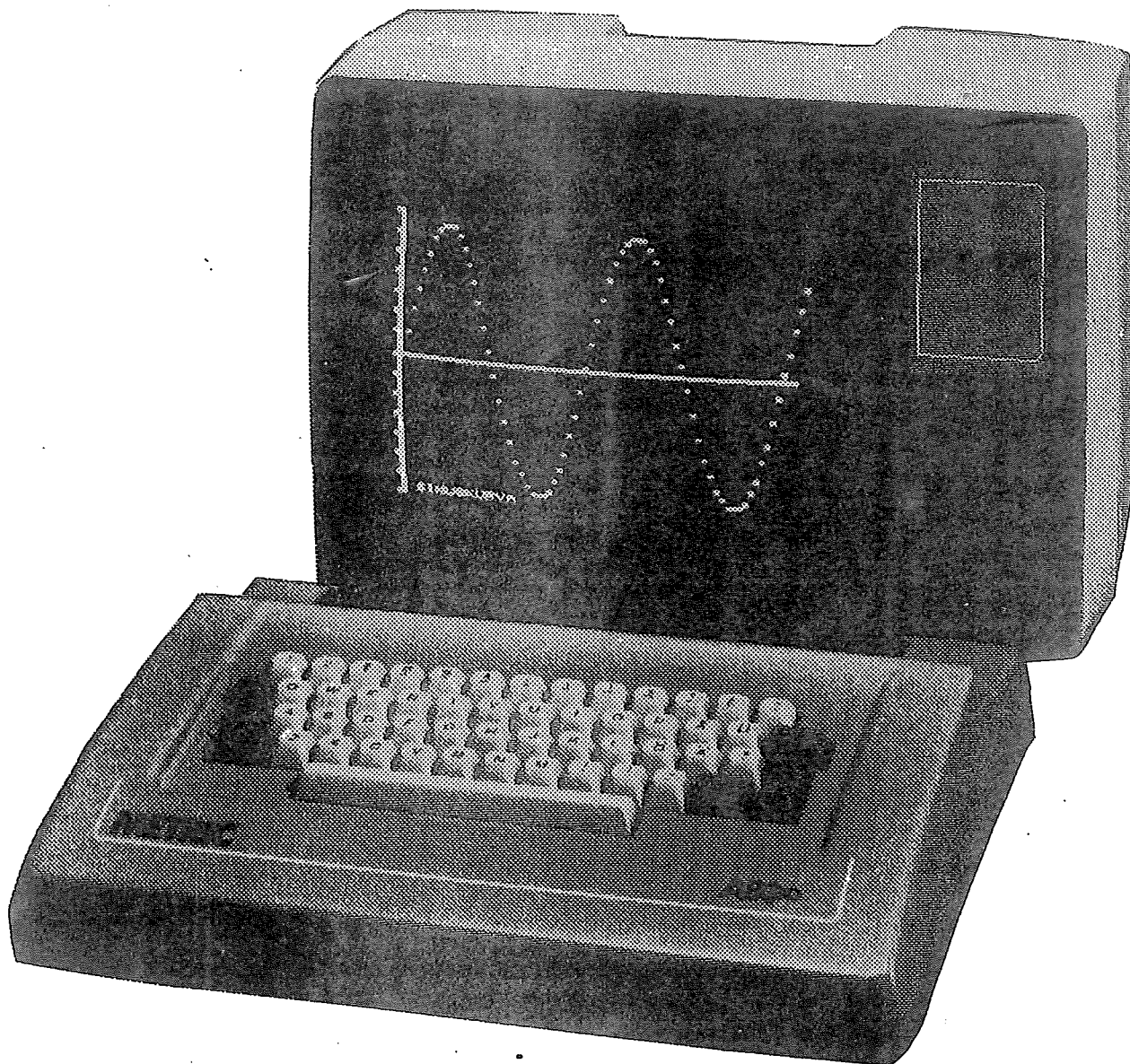


ABC80

KORT BESKRIVNING AV ABC-80 - BASIC



(utdrag ur specifikationer)

SCANDIA **METRIC** **AB**

BANVAKTSV. 20, FACK, 171 19 SOLNA, TEL 08/82 04 00

REGIONSKONTOR: ÅBÄCKSGATAN 6, 431 37 MÖLNDAL. TEL 031/81 09 75

DANMARK: TEL 02/80 42 00 NORGE: TEL 02/28 26 24 FINLAND: TEL 90/46 08 44

INNEHÅLL

- . Allmänna BASIC-kommandon
- . Allmänna BASIC-instruktioner
- . Läsning och skrivning
- . Programfilhantering
- . Datafilhantering
- . Stränghantering
- . ASCII-aritmetik
- . Grafik
- . On-satser
- . Matematiska funktioner
- . Slumptal
- . Felsökning
- . Minnesadressering och in/ut-portar
- . Maskinspråksrutiner
- . Specialtecken
- . Variabeltyperna i ABC80
- . V:24-interfacet

appendix

BESKRIVNING AV ABC-80:S BASIC

ALLMÄNNA KOMMANDON

<u>Kommando</u>	<u>Kort beskrivning</u>	<u>Exempel</u>
NEW	Tar bort alla programrader och stänger alla filer	NEW
SCR	Tar bort alla programrader och stänger alla filer	SCR
CLEAR	Nollställer alla variabler och stänger alla filer	CLEAR
LIST	Ger utskrift på bildskärmen av det program som finns i minnet	LIST LIST 100-200
LIST PR:	Ger utskrift på printer av det program som finns i minnet	LIST PR:
RUN	Kör det program som finns i minnet Alla variabler nollställs automatiskt.	RUN
REN	Numrerar om alla rader i programmet och motsvarande GOTO, GOSUB osv	REN (ger 10,20,30,40,... REN 200,20 (ger 200,220,240,260,...))
ED	Ger möjlighet att ändra i en programrad utan att behöva skriva om den	ED 40

ALLMÄNNA INSTRUKTIONER

<u>Instruktion</u>	<u>Kort beskrivning</u>	<u>Exempel</u>
REM	Kommentarer i program	20 REM Här kan det stå vad som helst
DIM	Reserverar utrymme för matris, strängmatris eller vektor	30 DIM A(10,10) 35 DIM A\$(10,10) = 5
FOR	Utför ett programavsnitt ett visst antal gånger	5 FOR I = 1 TO 10
STEP	Ger stegvärde till en FOR-sats	5 FOR I = 1 TO 10 STEP 2
NEXT	Avslutar ett FOR-avsnitt	20 NEXT I
LET	Tilldelar en variabel ett värde (Man kan också skriva uttrycket direkt utan LET)	40 LET A = B + 100 40 A = B + 100
IF-THEN-ELSE	<u>OM</u> <u>XX</u> <u>SÅ</u> <u>YY</u> <u>ANNARS</u> <u>ZZ</u>	10 IF A=B THEN C = A ELSE C = B
GOTO	Får programkörningen att fortsätta på viss rad	70 GOTO 100
GOSUB	Hoppa till subrutin	50 GOSUB 800

ALLMÄNNA INSTRUKTIONER

<u>Instruktion</u>	<u>Kort beskrivning</u>	<u>Exempel</u>
RETURN	Avslutar subrutin och hoppar till raden efter motsvarande GOSUB	800 RETURN
DEF FN_	Definiering av egna funktioner	100 DEF FNZ (X,Y) = X ~ X + Y * Y 120 DEF FNP%(R,K)=R%+K%
STOP	Stannar programkörningen och skriver ut STOP-satsens radnr	50 STOP
END	Stannar programkörningen utan att skriva ut radnummer och stänger alla filer	90 END

LÄSNING OCH SKRIVNING

<u>Instruktion</u>	<u>Kort beskrivning</u>	<u>Exempel</u>
PRINT	Skriver på bildskärmen	30 PRINT "NU HAR DET GÅTT"; A; "SEK"
;	Förkortning för PRINT	35; A,B,C
INPUT	Väntar på inmatning av data från tangentbordet	40 INPUT A,B,C
INPUTLINE	Läser in en rad från tangentbordet komplett med blanktecken och citationstecken (OBS att även tecknen vagnretur och radframmatning kommer att ingå på slutet.)	50 INPUTLINE AØ
GET	Läser ett tecken från tangentbordet till en strängvariabel, utan att "eka" tecknet på skärmen	100 GET AØ
READ	Läser variabelvärden från en DATA-sats	60 READ A,B,CØ,D
DATA	Lagrar variabelvärden som skall läsas av en READ-sats	55 DATA 12,3,45 "I LAGER",2
RESTORE	Gör att nästa READ-sats börjar läsa från den första datasatsen i programmet eller från viss datasats	70 RESTORE 80 RESTORE 55
TAB (K)	Flyttar cursorn till position K på raden	90 PRINT TAB(10);"KOL.10"
CUR (R,K)	Flyttar cursorn till rad R och kolumn K på skärmen. R får vara från 0 t.o.m. 23 och K från 0 t.o.m. 39.	100 PRINT CUR(11,12); "MITT PÅ SKÄRMEN"

PROGRAMFILHANTERING

<u>Instruktion</u>	<u>Kort beskrivning</u>	<u>Exempel</u>
LOAD	Hämtar in ett program från kassett eller floppy som lagrats med SAVE eller LIST Hämta "nästa" program på kassetten	LOAD LUFFAR LOAD CAS:
SAVE	Sparar program på kassett eller floppy komprimerat	SAVE GOLF
UNSAVE	Tar bort fil från floppy	UNSAVE KASSPROG
NAME AS	Byt namn på en fil på floppy	NAME OLDNAMN AS NYTTNAMN
LIST filnamn	Sparar programmet okomprimerat på kassett eller floppy	LIST PROGTEXT
RUN prog	Kör program på kassett	RUN DEMO
MERGE	Laddar in program från kassetten utan att ta bort det program som redan finns i minnet	MERGE SUBRUT
CHAIN	Hämtar in och kör ett nytt program från kassett eller floppy	100 CHAIN PROG2

Filnamn kan vara upp till 8 tecken långt + ev. ".BAS", ".BAC", ".TXT".

DATAFILHANTERING

<u>Instruktion</u>	<u>Kort beskrivning</u>	<u>Exempel</u>
PREPARE AS FILE	Skapar en ny fil med visst logiskt nummer från 1 till 255	40 PREPARE "BREV2.TXT" AS FILE 2
OPEN AS FILE	Tilldelar en fil ett logiskt filnummer från 1 till 255	30 OPEN "BREV.TXT" AS FILE 1
CLOSE	"Stänger" en fil, dvs avslutar läsning eller skrivning mot filen. Vid skrivning skrivs filslut	50 CLOSE 2
PRINT	Skriver till en fil på kassett eller floppy	35 PRINT 1, A,B,C
INPUT	Matar in data från fil på kassett eller floppy	41 INPUT 1,X,Y
INPUTLINE	Läser in en rad från en fil (se vanlig INPUTLINE)	52 INPUTLINE 1,Bo

Filnamn kan vara upp till 8 tecken långt + ev. ".BAS", ".BAC", ".TXT".

STRÄNGHANTERING

<u>Funktion</u>	<u>Kort beskrivning</u>	<u>Exempel</u>
LEFT(A\$,J)	Ger de J första tecken i strängen A\$	10 C\$=LEFT(A\$,J)
MID(A\$,I,J)	Ger J tecken från och med teckenposition I	20 D\$=MID(A\$,I,J)
RIGHT(A\$,I)	Ger alla tecken i A\$ från och med tecken I	30 D2\$=RIGHT(A\$,I)
LEN(A\$)	Ger aktuella längden av strängen A\$	40 L=LEN(A\$)
ASC(A\$)	Ger ASCII-värdet på första tecknet i A\$	50 V=ASC(A\$)
CHR(I)	Ger ett tecken med ASCII-värdet I (Kan också ha upp till 4 argument)	5 PRINT CHR(7) 15 PRINT CHR(65,66,67,33)
INSTR(I,A\$,B\$)	Söker efter strängen B\$ i strängen A\$ med början i pos I. Funktionen ger läget för första förekomsten av B\$ i A\$, eller, om B\$ ej finns i A\$, värdet 0.	60 P=INSTR(I,A\$,B\$)
SPACE(I)	Ger en sträng av I st mellanslag (space)	
STRING(I,C)	Ger en sträng av I st CHR(C)	
NUM(A)	Ger en sträng med de tecken man skulle fått om man skrivit ut A med en PRINT-sats	
VAL(A\$)	Ger värdet av A\$ tolkat som ett tal	
+	Sammanfogar två strängar	10 A\$=B\$+C\$+"!"

För snabbaste körning använd heltalsvariabler och heltal, ex. J%,2%.

ASCII-ARITMETIK

<u>Instruktion</u>	<u>Kort beskrivning</u>	<u>Exempel</u>
ADD(A\$,B\$,I)	Adderar A\$ och B\$ tolkat som tal, svaret <u>avrundas</u> till I decimaler. Strängarna får innehålla siffror, +, - och decimalpunkt, men ej exponent. Max 29 <u>siffror</u> .	
SUB(A\$,B\$,N)	dito subtraktion	

ASCII-ARITMETIK

<u>Instruktion</u>	<u>Kort beskrivning</u>	<u>Exempel</u>
MUL(A α , B α , N)	Multiplicerar A α och B α tolkat som tal, svaret <u>avrundas till</u> N decimaler. Strängarna får innehålla siffror, +, - och decimalpunkt men ej exponent. Max 29 <u>siffror</u> .	
DIV(A α , B α , N)	Dito division	
COMP%(A α , B α)	Jämför A α och B α tolkade som tal -1 om talet i A α är mindre än talet i B α 0 om talet i A α är lika med talet i B α 1 om talet i A α är större än talet i B α	

GRAFIK

<u>Instruktion</u>	<u>Kort beskrivning</u>	<u>Exempel</u>
SETDOT R,K	Tänder en grafisk punkt på skärmen R får vara från 0 till 72 K får vara från 2 till 79	110 SETDOT 18, 19
CLRDOT R,K	Släcker en grafisk punkt på skärmen	120 CLRDOT 18, 19
DOT (R,K)	Ger värdet SANT om grafiska punkten R,K är tänd	130 IF DOT (R,K) THEN SETDOT R+1, K+1

ON-SATSER

<u>Instruktion</u>	<u>Kort beskrivning</u>	<u>Exempel</u>
ON GOTO	Hoppar till olika rader i programmet beroende på värdet av en variabel	10 ON I GOTO 100,200,30
ON GOSUB	Hoppar till olika subrutiner beroende på värdet av en variabel	20 ON I GOSUB 100,200,30
ON RESTORE	Gör RESTORE till olika datasatser beroende på värdet av en variabel	30 ON I RESTORE 50,60,70
ON ERROR GOTO	Ger hopp till viss programrad om fel skulle uppstå under körningen (Funktionen ERRCODE innehåller felnumret)	5 ON ERROR GOTO 999

MATEMATISKA FUNKTIONER

<u>Instruktion</u>	<u>Kort beskrivning</u>	<u>Exempel</u>
SIN (X)	$\sin X$, X i radianer	
COS (X)	$\cos X$, X i radianer	
TAN (X)	$\tan X$, X i radianer	
ATN (X)	$\arctan X$ (radianer)	
LOG (X)	naturliga logaritmen av X, $\ln X$	
LOG10 (X)	10-logaritmen av X, $\log X$	
EXP (X)	e^X	
SQR (X)	Kvadratroten ur X, $\sqrt{X}$	
INT (X)	Största heltal mindre än eller lika med X	
FIX (X)	Heltalsdelen av X	
ABS (X)	Absolutvärde av X, $ X $	
SGN (X)	$\text{SGN}(X) = -1$ om $X < 0$ 0 om $X = 0$ $+1$ om $X > 0$	
PI	3,14159	
SWAP% (I%)	Ger ett heltal som motsvarar I% men där högre och lägre byte bytt plats	

SLUMPTAL

<u>Instruktion</u>	<u>Kort beskrivning</u>	<u>Exempel</u>
RND	Slumptal mellan 0 och 0.999999 inklusive dessa	100 A = RND
RANDOMIZE	Ger RND-funktionen ett slump- mässigt startvärde	110 RANDOMIZE

FELSÖKNING

<u>Instruktion</u>	<u>Kort beskrivning</u>	<u>Exempel</u>
TRACE	Radnummer på exekverade satser skrivs ut under körning (denna sats kan även användas som kommando)	130 TRACE
NOTRACE	Stänger av TRACE	700 NOTRACE

MINNESADRESSERING OCH IN/UT-PORTAR

<u>Instruktion</u>	<u>Kort beskrivning</u>	<u>Exempel</u>
PEEK (X)	Hämtar en byte från minnesadress X	5 A% = PEEK (X)
POKE	Lägger ut data, byte för byte, till en viss minnesadress	10 POKE 65008%, 21,47
INP (I)	Hämtar en byte från port I	15 B% = INP (I)
OUT	Skickar data till en 'utport'	20 OUT 6,D%

MASKINSPRÅKSRUTINER

<u>Instruktion</u>	<u>Kort beskrivning</u>	<u>Exempel</u>
CALL (A)	Anropar maskinspråksrutin på adress A och ger värdet av HL-reg. tolkat som heltal	100 Q% = CALL (65408)
CALL (A ,D%)	Som ovan, men D% läggs i reg-DE vid anropet	110 Q% = CALL (A ,1024%)

OPERATORER

	<u>Kort beskrivning</u>	<u>Exempel</u>
NOT	ICKE.sant om operanden falsk	10 IF NOT A>B THEN 100
OR	ELLER.sant om minst en operand är sann	20 IF A>B OR B>C THEN 100

OPERATORER

	<u>Kort beskrivning</u>	<u>Exempel</u>
AND	OCH, sant om båda sanna	30 IF A > B AND B > C THEN 100
XOR	EXKLUSIVT ELLER, sant om ena sann men inte båda	40 IF A > B XOR B > C THEN 100
IMP	IMPLICERAR (medför) Uttrycket C IMP D är falskt endast om C är sann och D är falsk	50 IF A > B IMP B > C THEN 100
EQV	EKVIVALENT (lika) Sant om båda är sanna eller båda är falska	60 IF A > B EQV B > C THEN 100
**	Upphöjt till heltalsexponent, ex. 2**3=8 Det är endast tillåtet att upphöja till heltalsvariabler eller heltalskonstanter (med %). A**B skall skrivas A**INT(B) för att få heltalsexponent.	70 C = B05% 80 Y=2.23**X%

Operatorerna kan också användas binärt

80 A% = A% AND 7%

SPECIALTECKEN

	<u>Kort beskrivning</u>	<u>Exempel</u>
CHR\$(7)	"BELL", ger ett pip i ABC-80:s inbyggda högtalare	10 PRINT CHR\$(7)
CHR\$(12)	"FORMFEED", tömmer bildskärmen helt	20 PRINT CHR\$(12)
CHR\$(151)	"START GRAFIK" startar grafisk mod på en rad	20 PRINT CHR\$(151);
CHR\$(135)	"SLUT GRAFIK" avslutar grafisk mod på raden	30 PRINT CHR\$(135);

Tangentbordet

CTRL-C	Avbryter pågående programkörning eller listning
CTRL-X	Backstegar hel rad vid inmatning

Variabeltyperna i ABC-80

I ABC-80 finns det flera olika typer av variabler:

1. Flyttalsvariabler

En flyttalsvariabel har namn som består av en bokstav från A-Ö eller en bokstav och en siffra mellan 0 och 9.

Ex. A, Z1, Ö9

En flyttalsvariabel kan ha värden från $\pm 0,1 \text{ E} - 127$ till $\pm 0,999999 \text{ E} + 127$.

(E betecknar 10-exponent, ex. $0,1 \text{ E} - 6$ motsvarar $0,1 \cdot 10^{-6}$).

Resultaten av alla beräkningar avrundas automatiskt till 6 siffror.

2. Heltalsvariabler

En heltalsvariabel har namn som en flyttalsvariabel men åtföljt av %.

Ex. I%, Y1%, A%

En heltalsvariabel kan ha heltalsvärden mellan -32768 och +32767.

Heltalsvärden kan alltså inte ha decimaler, men de går avsevärt snabbare att räkna med. Man brukar därför använda dem som loopräknare och för indexering m.m. Detta finns närmare beskrivet i boken "ABC OM BASIC-PROGRAMMERING".

3. Strängvariabler

Strängvariabler har namn som flyttalsvariabler men åtföljt av strängtecknet X , "sol". (Motsvarar \$ i amerikansk datorlitteratur).

Ex. AX, KX, P1X

I ABC-80 behöver man i allmänhet inte deklarerera storleken hos strängvariabler. En sträng som inte deklarerats med DIM får automatiskt maxlängden 80 tecken. Se också strängmatriser nedan.

4. Fälttabeller

Ett fält kan ha ett eller två index. Ett fält med ett index kallas vektor och ett med två index kallas matris. Index börjar på 0. Elementen kan vara flyttal, heltal eller strängar. Flyttal har 5 bytes per element, heltal 2 bytes. Om man inte deklarerar en matris, reserveras automatiskt plats som om man hade skrivit t.ex. DIM A(10) eller DIM A(10,10).

I ABC-80 kan man behandla strängar på flera olika sätt:

10 DIM A\$ = 80 ger en sträng innehållande upp till 80 tecken

20 DIM B\$(60) ger ett fält med 61 element; [B\$(0) till B\$(60)]
där varje element är 80 tecken långt.

30 DIM C\$(10,10) = 5 ger ett fält med 11 x 11 element, där varje element är 5 tecken långt.

40 DIM D\$(20) = 10 ger ett fält med 21 element, där varje element är 10 tecken långt.

Man brukar säga att C\$ är en strängmatris och D\$ en strängvektor.

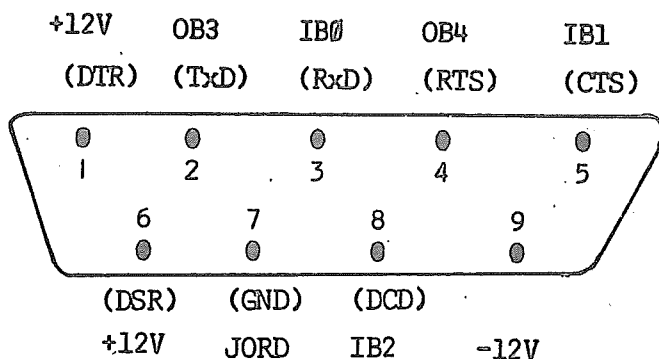
(I rad 20 ovan, DIM B\$(60), ger man inte själv någon längd på sträng-elementen. I detta fall tar varje element ingen plats i minnet förrän det tilldelats ett värde), dock minst 80 tecken.

V:24=interfacet

Detta interface är i första hand avsett för anslutning av ABC-80 till modem och för användning som "intelligent terminal".

Det är naturligtvis inget som hindrar att man här ansluter t.ex. lysdioder, tryckknappar, reläer m.m.

V:24-kontakten



(Förkortningarna inom parentes gäller användning som modemkontrollsignaler)

IB0 betyder In Bit 0, OB3 betyder Out Bit 3

Signalnivån i V:24-kontakten är $\pm 12V$ ut, max 10 mA och ± 5 till $\pm 12V$ in, där minusspänning ger '1'-bit till PIO.

PIO -kretsen i tangentbordet

128	64	32	16	8	4	2	1	Vikt
7	8	5	4	3	2	1	0	Bit

Band in	Band ut	Motor-relä	ut	ut	in	in	in
			(RTS)	(TxD)	(DCD)	(CTS)	(RxD)

V:24-kontakten är kopplad till port 58 (decimalt) som är en Zilog PIO-port. Bit 0,1,2 är inbitar och bit 3 och 4 är utbitar. När man läser från porten får man både in- och utbitarna. Om man önskar maska bort utbitarna kan detta lätt göras med AND:

Exempel

90 Z% = 7% AND INP (58%)

Man kan säga att de olika bitarna är binärt viktade enligt fig.

Exempel

Ex. 2 För att sätta bit 3 på PIO (övriga utbitar nollställs, se Ex. 4)
OUT 58%, 8% :REM 8=2 3

Ex. 3 Vänta på att Inbit 0 går från '0' till '1'

30 IF (1% NAD INP (58%)) = 0% THEN 30