

MILES GORDON TECHNOLOGY plc
představuje tento technický vývojový manuál jako koncept s
prozatímní technickou informací.
Avšak každé úsilí má směřovat k výrobě spolehlivého obsahu
MGT nemůže však zaručit jeho platnost, nebo že nebude nyní
vyroben podle plánovaných změn.

- - SAM Coupé vývoj verze 2.3 - -

Záměrem tohoto manuálu, je představit hardware a software ke vnitřní a vnější práci na počítači SAM Coupé. K tomuto účelu je určen technický dokument, spolehlivý prostředek technických znalostí.

To je záměr MGT pro obstarávání jak množství technických informací, tak možností učinit čtenáře schopným provozovat Coupé.

Jestliže je potřeba nějakého dalšího ujasnění, vysvětlení nebo pomoci, MGT s radostí pomůže.

Úvod k SAM Coupé.

Mikropočítač SAM Coupé je přístroj druhé generace s mikroprocesorem Z 80B s frekvencí 6 MHz. Je vybaven 256 kilobyty paměti s dvěma dalšími objímkami pro dalších 256 kilobytů paměti. Hlavní kontrolou přístroje je VLSI VGT-200 vstupní řady, obvykle k řízení mimo hlavní úkol procesoru/video, listování a řízení všech pamětí, mapování videopaměti, rozvržení barevné palety a všech vstupních a výstupních řídicích portů. Zvuk přístroje je generován Philips SAA 1099 se schopností stereo se šesti kanály při osmi oktávách s 256 tóny v oktávě, dva bílé hlasové generátory, 6 amplitodových kontrolů a dva moderátory vlnění. Jsou zde 4 módy video operací hardwarové kapacity, které ukazují 16 barev z palety 128 na obrazovce.

Je zde 8 integrovaných obvodů na hlavním tiskovém obvodu.

Z 80B ústřední výkonná jednotka / CPU/

32 .. kilobyte čtení pouze pamětí / ROM/

256x4 kilobyte dynamicky náhodně přístupná paměť / DRAM/

256x4 kilobyte dynamicky přístupná náhodná paměť / DRAM/

SAA 1099 čip šesti kanálového směšovače stereo zvuku

Opto-elektronický párovač pro MIDI interface

MC 1377 Video kodér UHF PAL konvertor

MGT-SAM-Coupé 1B ...MGT obvyklé vstupní řady /ASCII/

INTERFACE na stěně Coupé.

Všechna spojení k Coupé, kromě odstranitelných vnitřních disk drivů jsou vyvedena přez zadní panel.

NMI tlačítko

Když stisknete toto tlačítko směruje CPU na adresu 0066H, kde je stanovená adresa v paměti. Je to mnohostranná funkce pro programátory ke stanovení řízení, k oprávnění použití klávesy jako ESC, BREAK nebo CRASH

MIDI IN

Standartní 7 kolíkový konektor DIN. Tento sériový vstup pracující ve 31,25 K baudech je zásobován cestou opto-izolátoru k sériově/paralelnímu konvertoru, který přerušuje CPU, když data mají být shromážděny v MIDI-IN registru /253 dec/. Tento INPUT může také být čten vedle bitem 7 registru video paměti VMPR /252 dec/. Tento konektor je také používán při NETWORK /síti/

PIN	Signál
1	NET-LOOP
2	N.C.
3	NET + LOOP
4	MIDI + IN
5	MIDI - IN
6	NET - LOOP
7	NET + LOOP

MIDI_OUT

Standardní 7 kolíkový konektor DIN. Má za úkol psaní údajů a dat k MIDI OUT registru /253 dec/. MIDI výstup vede proud 7,5 mA ve 31,25 Kbaud. Když odesílá 1 bit z PEN registru /TXFMST z registru 248 dec/ je uložen. Tento OUTPUT může být také řízen bitem 7 z VMPIR / 252 dec/. Vnitřní spojení je provedeno z MIDI IN do MIDI OUT použitím bitu 6 THROM - z MIDI BORDER registru /254 dec/. Tento konektor je také užíván při NETWORK / počítačových sítích/. Přerušování je dáno ukončením odeslání MIDI údajů.

PIN	SIGNÁL
1	NET - LOOP
2	GND
3	NET + LOOP
4	MIDI + OUT
5	MIDI - OUT
6	NET - LOOP
7	NET + LOOP

JOYSTICK

Standardní 9-ti kolíkový typu "D". Joystick má standardní ATARI spoje spolu s 5V. MGT dovoluje pro dvojnásobné ovládání s druhým joystickem použití speciálního konektoru - kolíkové zásuvky. Joystick je čten z portu klávesnice /254 dec/ a pokrývá číselné klávesy 1-5 pro druhý joystick a 6-0 pro první joystick.

PIN	SIGNÁL	
1	UP	nahoru
2	DOWN	dolu
3	LEFT	doleva
4	RIGHT	doprava
5	0 VOLTS	
6	FIRE	střelba
7	+ 5 V	
8	STROBEL 1	
9	STROBEL 2	

Myš

Standartní osmi kolíkový konektor DIN.MGT myš přerušuje CPU na žádost o čtení řádků signálu X a Y. Software je celý v ROM a pokrývá řídicí klávesy. Výstupy myši mohou být čteny z portu klávesnice /254 dec/ s adresou řádků A8 až A15 /RDMSEL /

PIN	SIGNÁL	
1	DOWN	dolů
2	UP	nahoru
3	CTRL	
4	LEFT	doleva
5	RIGHT	doprava
6	MSE INT	
7	RDMSEL	
8	+ 5 voltů	
SCREEN	GROUND	

RESET tlačítko

Když stisknete toto tlačítko přimějete CPU k vyčistění a opakování řádku v počátku paměti. /0000 hex /.

To také obnovuje následující:

- 1.Kontrola flopy disket
- 2.Registry stránky paměti
- 3.Registry přijetí a odeslání MIDI
- 4.Registr BORDER

Neobnovuje barevnou nabídku, registry zvukového čipu, nebo hodnotu LINE přerušovaného registru / 249 dec /

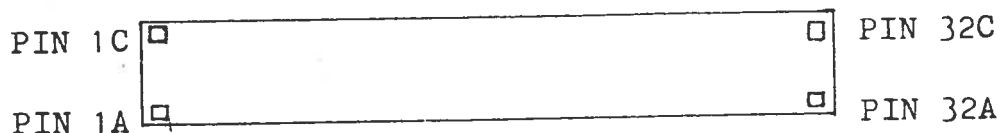
EXPANZNÍ Konektor

Standartní 64 kolíkový Eurokonektor zásuvka s řadou A a C.
Seznam signálů dosažitelný z této zásuvky je ukázán níže.
Znaky řady A jsou v dolní části konektoru a řada C je na-
hoře. Všechny znaky končící na L jsou málo účinné.

PIN	SIGNÁL	PIN	SIGNÁL
1A	DBDIRL	1C	IORQL
2A	RDL	2C	MREQ
3A	WRL	3C	HALTL
4A	BUSAKL	4C	NMIL
5A	WAITL	5C	INTL
6A	BUSREQ	6C	CD1
7A	RESETL	7C	CD0
8A	CMIL	8C	CD7
9A	REFRESHL	9C	CD2
10A	0 voltů	10C	+ 5 voltů
11A	A0	11C	CD6
12A	A1	12C	CD5
13A	A2	13C	CD3
14A	A3	14C	CD4
15A	A4	15C	CPU CLK
16A	A5	16C	A15
17A	A6	17C	A14
18A	A7	18C	A13
19A	A8	19C	A12
20A	A9	20C	A11
21A	A10	21C	DISC 2
22A	MSEINTL	22C	ROMCSL
23A	XMEML	23C	EARMIC
24A	8 MHz	24C	DISC 1L
25A	RED 1	25C	PRINTL
26A	GREEN 1 /zelená/	26C	BLUE 1
27A	C SYNCL	27C	ROMCSRL
28A	SPEN	28C	AUDIO RIGHT OUTPUT /pravý výstup
29A	BLUE 0 /modrá/	29C	AUDIO LEFT OUTPUT /levý výstup/
30A	RED 0 / červená/	30C	COMP VIDEO
31A	BRIGHT	31C	GPEFN 0 / zelená/
32A	+ 5 voltů	32C	0 voltů

Prosím znaky:

Když se podíváte na Eurokonektor v zadní části Coupé, jednotlivé piny jsou číslovány následovně:



Magnetofonový Jack

Standartní 3,5 mm mono Jack. Je obousměrný. Když nahráváme bude spojen se zásuvkou MIC na standartním ~~kazetovém~~ magnetofonu, když přehráváme bude spojen se zásuvkou EAR na magnetofonu. Použití není jiné u Coupé, jestliže EAR a MIC jsou spojeny spolu, ale jenom některé magnetofony to dovolují.

Světelné pero

Standartní 5-ti kolíkový konektor DIN. Konektor má dvojitou funkci:

-vstup světelného pera/pušky tak dobře jako výstup stereo zvuku k řízení HIFI systému na vstup AUX.

PIN	SIGNÁL
1	+ 5 voltů
2	AUDIO LEFT OUTPUT levý výstup
3	0 voltů
4	SPEN INPUT
5	AUDIO RIGHT OUTPUT pravý výstup

Napěťový vstup

Upravený DC vstup z 12 voltů, 100 mA a 5 V, 2 mA. Zcela připravený přístroj má max. spotřebu 15 W.

PIN	SIGNÁL
1	+ 5 voltů
2	0 voltů /signál GROUND/
3	0 voltů / DIGITAL GROUND/
4	COMPOSITE VIDEO
5	+ 12 voltů
6	SOUND OUTPUT MONO / zvukový výstup/

SCART zásuvka

SCART zásuvka se používá k ovládání všech video a zvukových výstupů. Ne všechny spoje jsou standartní.

PIN	SIGNÁL	PIN	SIGNÁL
1	AUDIO OUT RH	2	SPEN
3	AUDIO OUT LH	4	AUDIO EARTH
5	BLUE EARTH/modrý okraj	6	BLUE TTL OUT
7	BLUE LIN.OUT	8	RED TTL OUT
9	GREEN EARTH	10	GREEN TTL OUT
11	GREEN LIN.OUT	12	+5V napětí IN
13	RED EARTH	14	CSYNC EARTH
15	RED LIN.OUT	16	CSYNC
17	C.VID EARTH	18	+ 12V napětí IN
19	C.VID OUT	20	BRIGHT TTL OUT
21	GND		

Úzká štěrbina v DISK DRIVECH

Disk Drivey používané v SAM Coupé jsou typu CITIZEN 3,5",
1 MB / formátován na 780 kB/

Coupé může řídit nejvíce dva disk drivey. Spojení se dvěma
drivey jsou vyrobeny jako dva 32 kolíkové Eurokonektory s
řádky A a B.

Drive 1 je na levé straně přístroje a drive 2 na pravé.
Seznam signálů dosažitelných z těchto pinů jsou ukázány
níže:

PIN	SIGNÁL	PIN	SIGNÁL
1A	0 voltů	1B	WR
2A	0 voltů	2B	A0
3A	0 voltů	3B	A1
4A	0 voltů	4B	D0
5A	0 voltů	5B	D1
6A	0 voltů	6B	D2
7A	0 voltů	7B	D3
8A	0 voltů	8B	D4
9A	0 voltů	9B	D5
10A	0 voltů	10B	D6
11A	5 voltů	11B	D7
12A	5 voltů	12B	8 MHz
13A	5 voltů	13B	RST
14A	5 voltů	14B	
15A	5 voltů	15B	A2
16A	5 voltů	16B	DISC 1 OR DISC 2

UHF_výstup

UHF televizní standartní video a mono zvuk mají výstup na
kanálu 36.

Poznámka: UHF výstup je potom silně poddajný.

Paměť Coupé

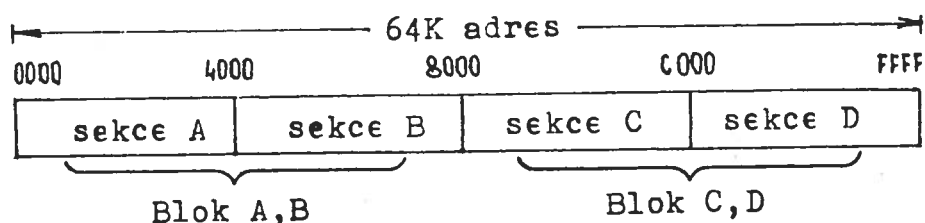
Coupé má kapacitu 512 kB RAM /kromě prostoru přes rozváděcí konektor/ používající signál XMEML. Strojový BASIC je vybaven 256 kB a dvěma zásuvkami pro dalších 256 kB. Používaná paměť je DRAM 256x4 bity - 100ns ú/nanosekund/ časově přístupné 20 kolíkový dvojité plastický balík

Protože Z 80B CPU je limitován 64K adresovaného rozsahu, jedna z funkcí z ASIC /používání určitých celých obvodů / v Coupé je k řízení adresování v této paměti rozdělené na 32 stran po 16 K.

ASIC řídí stránkování přes 8-mi bitový čtecí/psací registry LMPR /nižší paměťová stránka registru/I/O adresa 250 dec\$
HMPR /vyšší paměťová stránka registru/I/O adresa 251 dec\$

Nižších 5 bitů každého registru představuje stránku 0 až 31 /ovšem na basic přístroji s 256K pamětí stránkuje od 0 do 15/

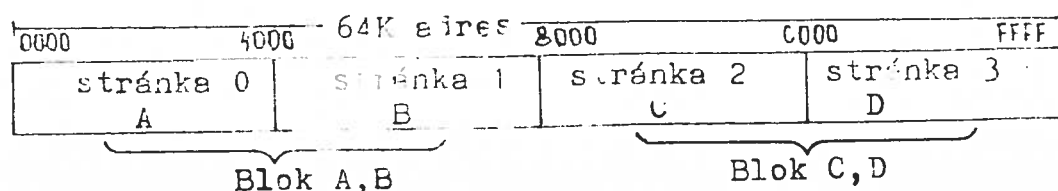
Na ilustraci systému stránkování na Coupé je nejlepší ukázat 64K adresného obvodu Z 80 jako 2 bloky dvou sekcí 16-ti kilových, reprezentujících písmena A, B a C, D:



LMPR řídí blok A,B a HMPR řídí blok C,D

Jestli-že píšeme 00 hex na LMPR, tak stránka 0 paměti je přidělena sekci A adresného obvodu CPU. Sekce B je vždy automaticky přidělena jedné stránce nad sekci A, v tomto případě stránka 1. Jestli-že píšeme 02 hex na HMPR pak stránka 2 paměti je přidělena sekci C adresného obvodu CPU. Sekce D je vždy automaticky přidělena stránce nad sekci C, v tomto případě stránka 3

V našem případě mapa adres CPU bude vypadat takto:



Pro ilustraci operace systému stránkování je zde praktický příklad:

Problém Napiš bloky dat do sekce C paměti k umístění MEMFREE

Řešení Nejprve opatříme registr registrem HL s MEMFREE

LD HL, MEMFREE

Pravidelný postup/subrutina/ DATA je povolán, píše blok údajů z paměti, směřuje k HL, vynese HL a vrátí, když bajty jsou vlevo.

LOOP CALL DATA

Zastaví se, jestliže HL má jít přes hranice stránky

LD A, H

CP C0 hex

JR C, LOOP

HL má jít přes sekci C hranice stránky, tak zanese H u stránky jedna 16-ti K

SBC 40 hex

LD H, A

a vynese HMPR

IN A, /HMPR/

INC A

OUT /HMPR/, A

JR LOOP

Samozřejmě v tomto případě si ověříme, že nemáme více údajů bloků ke vstupu a že nevyneseme plynule dosažitelné stránky paměti.

Musíme pamatovat, že jiné řídící bity jsou používány v těchto registrech a tedy musíme dát pozor při jejich používání.

Obrazové módy Coupé

Jsou zde 4 obrazové módy používané v Coupé, každé použití mění atributy paměti a každý mód má rozdílné atributy pro použití programátorem.

Hardwárový ukazatelé, kteří ukazují obsahy paměti jsou řízeny VMPR / video registr stránky paměti - 252 dec /.

Použitím nižších 5-ti bitů tohoto registru můžeme zpřístupnit až 32 stran obrazové paměti. Je nutno si povšimnout, že módy 3 a 4 používají 24 kB, které jdou přes hranice stránky 16 kB. Kde se to stalo, video adresující hardware sbalí do další stránky mimo tentýž blok.

Např.: Při vstupu strany 12 do VMPR, video sbalí na stranu 13.

To je jednoduchý příklad. Totéž bude použito u kterékoliv stránky

Módy jsou:

MÓD 1

32 buněk x 24 řádků ve dvou barvách ze 16 z palety 128 barev, dostanete 768 znakových buněk / 8x8/ používá 6 kB ze zmapované paměti a 0,75 kB ze znakové paměti. Tento mód napodobuje paměť Spectra.

MÓD 2

32 buněk x 192 řádků ve dvou barvách ze 16 z palety 128 barev, dostanete 6144 znakových buněk / 8x1/ používá 6 kB ze zmapované paměti a 6 kB ze znakové paměti. Tento mód má styčnou paměť adresovanou ve dvou blocích.

MÓD 3

512 bodů x 192 řádků ve 4 barvách z palety 128 barev, dostanete 98304 bodů, užívá 24 kB paměti. Pokud v tomto módu použijete znaky 6 bodů širokých dostaneme 85 znaků na řádek.

MÓD 4

256 bodů x 192 řádků v 16 barvách z palety 128 barev, dostanete 49252 bodů, užívá 24 kB paměti. Tento mód je ideální pro grafické znázornění a když ho použijete v kombinaci s LINE INT registrem můžete znázornit všech 128 barev na obrazovce.

Klávesnice

Coupé má celkem 72 kláves na klávesnici. Je použita jako forma 9x8, užívající dva porty/brány/ KEYBOARD /klávesnici//254 dec/ a STATUS /postavení/ / 249 dec/

Osm vstupních řad je provedeno jako vstupní KEYBORT port nižších 5 bitů je reprezentováno K1, K2, K3, K4 a K5 zatím co vstupní port STATUS vyššími 3 bity reprezentováno K6, K7 a K8. 9 vstupních řádků je provedeno jako CPU řádky adres AD8, AD9, AD10, AD11, AD12, AD13, AD14, AD15 a ASIC řádek RDMSEL

Zde je několik hlavních vstupních bodů ROM:

1. Získat znak z klávesnice buferu
2. Vyčistit bufer a získat další znak
3. Jestliže se chcete podívat, zvolte BREAK

Typický příklad prohlížení klávesnice:

LD	HL, SCAN	; soubor HL s bufer statutem
LD	B, 11111110bin	; soubor horních řad adresy
LOOP LD	C, HIKEY	; port adresy K6-K8
IN	A, /C/	
AND	11100000bin	; pruh nepotřebných bitů
LD	/HL/, A	; podžet to
LD	C, LOKEY	; port adresy K1-K5
IN	A, C	
AND	00011111bin	; pruh nepotřebných bitů
OR	/HL/	; pracuje K1-K8
LD	/HL/, A	; zachovat to
INC	HL	; další umístění v buferu
SCF		; soubor proveden
RLC	B	; otáčí další adresu řádku
JR	C, LOOP	; skok jestliže neudělal
IN	A, /C/	; vstup RDMSEL řádku
AND	00011111	; pruh nepotřebných bitů
LD	/HL/, A	; zachovat to
RET		; návrat k rozboru prohlížení

Zvukové porty /adresa 511 dec/ a /data 255 dec/

Zvukový čip Philips SAA 1099 je řízen přes tyto dva porty adres. Reprodukce používání znaků Philips jsou obsaženy na konci tohoto manuálu spolu s pohotovou zmínkou vedení řídicích registrů.

BORDER port /254 dec/

Tento výstupní port hlavně řídí barvu okolí / border/ obrazovky. Je opatřený 4 bitovou adresou ke CLUT / color look up Table/ k oprávnění barvy k ukázání během času okolí.

bit 0	BCD 1	adresa CLUT pro barvu okolí
bit 1	BCD 2	adresa CLUT pro barvu okolí
bit 2	BCD 4	adresa CLUT pro barvu okolí
bit 3	MIC	výstup řídicího bitu, normálně umístěn vysoko
bit 4	BEEP	výstup řídicího bitu, normálně umístěn nízko
bit 5	BCD 8	adresa CLUT pro barvu okolí
bit 6	THROM	bit umístěn vysoko, přístupný přes operaci MIDI
bit 7	SOFF	bit umístěn vysoko, k oslabení obrazovky, účinný pouze v obrazových módech 3 a 4, také odkládá paměť během sporu do konce období

MIDI_OUT port /253 dec/

Po napsání údaje bajtu tomuto portu hardware automaticky odesílá skrze kanál MIDI OUT v 31,25 k baud, standardní pro MIDI protokol. Bit 4 registru STATUS /249 dec/ je umístěn výše, pokud je tento registr plný.

VMPR - Video stránka paměti registru/252 dec/

Tento čtecí/psací registr hlavně řídí stranu adresovanou paměti zobrazené na obrazovce

bit 0	R/W	BCD 1	řízení stránky video
bit 1	R/W	BCD 2	řízení stránky video
bit 2	R/W	BCD 4	řízení stránky video
bit 3	R/W	BCD 8	řízení stránky video
bit 4	R/W	BCD 16	řízení banky video, používá se k přepnutí mezi bankami 256 kB
bit 5	R/W	MDE0	první bit řízení modu obrazovky
bit 6	R/W	MDE 1	druhý bit řízení modu obrazovky
bit 7	-/W	TXMIDI	výstupy bitu k písmenu řízení kanálu MIDI OUT
bit 7	R/-	RX MIDI	vstup bitu z kanálu MIDI IN

HMPR - Hig memory page register

Tento čtecí/psací registr je používán hlavně k řízení stránkování vyšší paměti adresovaného rozsahu v CPU.

- | | | | |
|-------|-----|--------|---|
| bit 0 | R/W | BCD 1 | řízení vyšší stránky paměti |
| bit 1 | R/W | BCD 2 | řízení vyšší stránky paměti |
| bit 2 | R/W | BCD 4 | řízení vyšší stránky paměti |
| bit 3 | R/W | BCD 8 | řízení vyšší stránky paměti |
| bit 4 | R/W | BCD 16 | řízení vyšší banky paměti |
| bit 5 | R/W | MD3S0 | BCD 4 vyhledá adresu barvy dosažitelné pouze v módu 3 |
| bit 6 | R/W | MD3S1 | BCD 8 vyhledá adresu barvy dosažitelné pouze v módu 3 |
| bit 7 | R/W | MCNTRL | jestliže tento bit je použit, když CPU adresuje vyšší paměť, jak vnější signál XMEM jde dolů a Coupé ukazuje na rozváděcí konektor pro vyšší paměť. |

LMPR - registr stránky nižší paměti / 250 dec/

Tento čtecí/psací registr je používán hlavně k řízení stránkování nižší paměti adresovaného rozsahu CPU.

- | | | | |
|-------|-----|-------|---|
| bit 0 | R/W | BCD1 | řízení nižší stránky paměti |
| bit 1 | R/W | BCD2 | řízení nižší stránky paměti |
| bit 2 | R/W | BCD4 | řízení nižší stránky paměti |
| bit 3 | R/W | BCD8 | řízení nižší stránky paměti |
| bit 4 | R/W | BCD16 | řízení nižší banky paměti |
| bit 5 | R/W | RAM0 | když bit umístí výše, RAM nahradí první polovinu ROM v sekci A mapy adresy CPU |
| bit 6 | R/W | ROM1 | když bit umístí výše, druhá polovina ROM nahradí RAM v sekci D mapy adresy CPU |
| bit 7 | R/W | WPRAM | napsání ochrany RAM v sekci A mapy adresy CPU je umožněno, když tento bit je umístěn výše |

Registr přerušení řádku

Tento pouze psací registr způsobuje přerušení CPU na konci, zkoumaného řádku hodícího se obsahu/na začátku pravého okolí/ Čísla řádků jsou 0 až 191. Tato funkce je vždy schopná, tedy překaží působení chybného čísla řádku od 192 do 255, které může být vloženo. Užitečná funkce tohoto registru může být

použita k přepnutí video módů nebo výměně barevných hodnot.

CLUT - Colour Look Up Table /základní port 248 dec, s AD 8 - AD 11/

Zde je pojednání o 16 pouze 7mi bitových registrech v CLUT

barva	0	registr adres	0	na portu 248
	1		1	504
	2		2	760
	3		3	1016
	4		4	1272
	5		5	1528
	6		6	1784
	7		7	2040
barva	8	registr adres	8	na portu 2296
	9		9	2552
	10		A	2808
	11		B	3064
	12		C	3320
	13		D	3576
	14		E	3832
	15		F	4088

Každý registr má 7 bitů ke znázornění jedné ze 128 možných barev.

bit 0	BLU0	nejmenší významný bit modré
bit 1	RED0	nejmenší významný bit červené
bit 2	GRN0	nejmenší významný bit zelené
bit 3	BRIGHT	polovina síly bitu ve všech barvách
bit 4	BLU1	největší významný bit modré
bit 5	RED1	největší významný bit červené
bit 6	GRN1	největší významný bit zelené

Registry při zapnutí budou naloženy s chybou z ROM, umístění bude odpovídat hodnotám.

registr 0	černá	registr 8	černá
1	modrá	9	světle modrá
2	červená	A	jasně červená
3	oranžová	B	jasně oranžová
4	zelená	C	jasně zelená
5	šedá	D	jasně šedá
6	žlutá	E	jasně žlutá
7	bílá	F	výrazně bílá

Typický postup pro naložení CLUT se seznamu paměti:

LD	HL, TABLE	;základ HL z vrchu seznamu
LD	B, 16	;základ B z objemu seznamu
LD	C, 248	;základ C s adresou portu
OTDR		;vykonat vklad B
		;výstup portu /C/ a
RET		;návrat k povolání pravidelného postupu

Vysvětlení CLUT

Obrazovka Mód 4

Na obrazovce v módu 4, jsou 4 bity užity k adresování CLUT pro barvu bodu. Kromě bajtu, nejdůležitější odkazuje na první bod a nejméně důležitý odkazuje na 2. bod.

Obrazovka Mód 3

Situace je obdobná v módu 3, ačkoliv v této době zde jsou pouze dva bity za bod, který adresuje CLUT. Mimo 8-bitového údaje bajtu, první dva nejdůležitější bity jsou použity jako adresa, pro první bod.

Přirozeně pouze 4 ze 16 možných registrů budou dostupné. To lze překonat použitím mimořádných dvou bitů z HMPR /251 dec/.

Bit 5 z HMPR je užíván ke zpřístupnění BCD 4 vyhledáním adresy barvy a bit 6 z HMPR je užíván ke zpřístupnění BCD8 vyhledáním adresy barvy.

Touto cestou můžeme ještě vyhledat 16 barev uvedených v CLUT avšak vyšší rozhodnutí má grafické znázornění.

Registr atributů / 255 dec/

Tento registr opravňuje programátora ke čtení atributů z běžně ukázaného znaku článku v módu 1 a 2, a třetí bajt ve čtyřech displeích v modech 3 a 4.

Registr klávesnice / 254 dec/

Tento pouze čtecí registr se hlavně používá jako vstupních 5 bitů - nižších - z klávesnice. Je to také vstup pro myš, když adresy řádků AD8 - AD15 jsou vyšší.

bit 0	K1	klávesnice řádek 1	ovládání myši
bit 1	K2		2 myš nahoru
bit 2	K3		3 myš dlu - knoflík 2
bit 3	K4		4 myš doleva - knoflík 1
bit 4	K5		5 myš doprava - knoflík 3
bit 5	SPEN	signál STROBE pro světelné pero/sériový vstupní bit	
bit 6	EAR	sériový vstup z EAR kazetového magnetofonu	
bit 7	SOFF	postavení bitu je ukázáno, jestliže je umístěna vnější paměť	

Registr MIDI_IN /253_dec/

Tento pouze čtecí registr provádí přerušování CPU, když údaje bajtu má číst ze sériového vstupu MIDI. Údaje musí být načteny do CPU před dalším přerušováním /obvykle 320 msec/

Registr postavení /249_dec/

Tento pouze čtecí registr se používá hlavně pro načtení přerušování postavení. Ačkoliv všech pět přerušování jde přes působící CPU pod mode 1 se přerušuje. Zde není známá cesta, která je požadovaná, proto načtení tohoto registru je nezbytné. Přibližný čas přerušování je asi 20 msec.

bit 0	LINE int	...když vydává zvuk, signalizuje řádku přerušování požadovaného registru	
bit 1	MOUSE int	..když vydává zvuk, signalizuje myši, že požaduje přerušování	
bit 2	MIDI IN int.	když vydává zvuk, signalizuje, že MIDI panel má údaj bajtu	
bit 3	FRAME int..	když vydává zvuk, signalizuje, že zkoumaná stavba má být úplná / 50 za sec/	
bit 4	MIDIOUT int.	když vydává zvuk, oznamuje, že registr MIDI OUT má právě úplné údaje vystupu	
bit 5	K6	int..klávesnice řádek 6	
bit 6	K7	int..klávesnice řádek 7	
bit 7	K8	int..klávesnice řádek 8	

Registry pera /LPEN - 248_dec/ / HPEN - 504_dec/

Jestliže světelné pero není zapojeno, tyto pouze čtecí registry jsou nepřetržitě nahoře datovány s běžnou polohou zkoumaného. Registr LPEN je napojen na vodorovný, nebo-li koordinát X funkce zkoumaného a HPEN je napojen svislý, nebo-li koordinát Y funkce zkoumaného.

První dva bity registru LPEN jsou:

bit 0 - vstupy BCD 1 běžné CLUT adresy

bit 1 - TXFMST je postavení bitu pro MIDI OUT. Pokud je vyšší, ukazuje, že bajt je odesílán.

Všeobecný popis MGT SAM Coupé

Pohon Mikroprocesor Z80B, 6MHz
Řízení Obvykle VLSI 10,000 - brána ASIC čipu
ROM 32K x 8 ROM, 150ns, obsahující SAM Basic, disk, BIOS
RAM 256K, maximálně až 512K / 256K x 4 100ns DRAM/
ZVUK Slučovač Philips SAA 1099:6 kanálů, 8 oktáv, stereo s
 kontrolou amplitudy a obsahu, plus volba tvaru vlny.

Grafické

znázornění

nabízející 4 módy

Mód 1 32x24 znakových článků na obrazovce, každý článek je schopen dvou barev, 16 barev vybraných ze 128 slučitelný se Spectrem

Mód 2 - obdobný jako mód 1, ale se 32 x 192 články, každý článek schopen 2 barev, 16 barev vybraných ze 128

Mód 3 - ukazuje 80 sloupců textu - 512 x 192 bodů, každý bod schopen zvolit barvu, 4 barvy na řádku volí ze 128

Mód 4 - 256 x 192 bodů grafického obrazu, každý bod schopen zvolit barvu, 16 barev na řádku volí ze 128

Ve všech módech mohou být barvy předefinovány v řádku přerušení sledující všech 128 barev ukázaných na obrazovce.

INTERFEJSY

UHF / TV kanál 36 / pro kvalitní přenos barev, digitální a lineární RGB přes SCART, standartní ATARI joystick /schopnost ovládat 2 joysticky přes speciální konektor/

Myš - standartní k Coupé.

Světelné pero/světelná puška - standart k Coupé

Domácí kazetový magnetofon

MIDI IN , MIDI OUT / MIDI jdoucí přes přepínač/

NETWORK /sítě/ - stíněný mikrofonový kabel 7 kolíkový konektor
DIN

Výstup zásuvky audio

RS 232 a paralelní tisk jdoucí přes vnější MGT interfejs spojen v rozváděcím portu.

64 kolíkový standartní rozváděcí port pro další periférie

Disk drive 1 nebo 2 a uvnitř namontovaný CITIZEN 3,5",

1MB nenaformátovaný, 780 KB po naformátování

Klávesnice má 72 kláves membránového typu, obsahuje 10 funkčních kláves / nadefinovaných /

Zdroj 4,75 - 5,25 V

Spotřeba 11,2 W

Chvění - působící 5 - 500 Hz/0,5 G

nepůsobící 5- 500 Hz/2 G

Prostředí" okolní teplota - působící 5 - 45 °C

uskladnění 20 - 50 °C

Relativní vlhkost - mokrá žárovka 29,4°C, bez kondenzace

Spolehlivost 10 000 provozních hodin

životnost komponentů větší než 5 let

Hmotnost: 2,26 kg

Napětí primární vstupy 240V, 50Hz

sekundární výstupy 5V, 2A

12V, 200mA

SAM Coupé - paměťová mapa

0FFFFH				GENERAL PURPOSE MACHINE BUFFER SPACE	
	ROM 1				
0C000H				MACHINE STACK	ISPVAL
	START OF CHANNELS		04F00H	FLOATING POINT CALCULATOR STACK	STKEND
05CB6H	SYSTEM VARIABLES	SVARS	04D00H	BASIC STACK FOR GOSUB/DO/PROCED- URES	STKBOT BASSTK
05A00H					BSTKEN
	KEYBOARD TABLE	KTAB		SPARE FOR HEAP OR BASIC STACK	
0580EH	DEF KEY			SYSTEM HEAP.	HEAPEN
	BUFFER	DKBU		3K AVAILABLE DOESN'T PAGE	
05800H	LINE INTERRUPT COLOUR TABLE	LINTCOLS	04000H		HEAPST
05600H	PALETTE TABLE	PALTAB	00000H	ROM 0	
055D8H	UDG PATTERNS CHR\$ (127 - 168)	UDGS			
05490H	CHARACTER PATTERNS CHR\$(32 - 127)	CHARS			
05190H	PAGE ALLOCATION OF 33 BYTES. (1516K PAGE+FFH)	ALLOCT			
05100H					

Další ukazatelé rozsahu paměti Basic

PROG	Začátek programu
NVARS	Začátek číselných proměnných
NVEND	Konec číselných proměnných
	Mezera
SAVARS	Začátek řady/sestavených proměnných
ELINE	Začátek editační řádky
WKSPACE	Začátek pracovního prostoru
WKEND	Konec pracovního prostoru
	Mezera
RAMTOP	Předešlý bajt přidělil program BASICu

Částečný přehled systémových proměnných

jmeno	SVAR číslo	
LNCUR	0	kurzor běžného řádku / obvykle ">" /
LCCUR	1	kurzor znak, když CAPS LOCK je vypnut / obvykle "*" /
UCCUR	2	kurzor znak, když CAPS LOCK je zapnut / obvykle "+" /
BIN1DIG	3	znak užívaný při BIN\$ jako "1" / obvykle "1" /
BIN0DIG	4	znak užívaný při BIN\$ jako "0" / obvykle "0" /
INSTHASH	5	znak užívaný při INSTR jako znak "nějaký boj" / obvykle CHR\$ 23 /
SLDEV	6	běžné písmeno hesla / obvykle "T" na tape /
SLNUM	7	běžný pásek nahrává pomalu, nebo chybně, drive číslo, pokud používáme disk drive /
SPEEDINK	8	pomalé kmitání barev / viz povel PALETTE /
PRRHS	14	text v pravém sloupci pro tisknutí / o jedno menší než maximální řádek podle přání /
AFTERCR	15	znakový kód odeslán k tisku přes CHR\$ 13, pokud je kanál "P" používán / obvykle 10, až dostaneme automaticky požadovaný řádek / . POKE SVAR 15,0 se používá k zastavení ja- kéhokoliv znaku.

jmeno	SVAR čís.	
DMPTL	16	/2 bajty/adresa levého horního rohu na obrazovce pro uložení grafiky
DMPLN	18	délka uložení grafiky /v 8mi bodové jednotce, obvykle 22/
DMPWID	19	šířka uložení grafiky/v jednotce široké 8 bodů, obvykle 32/
DMPWM	20	násobitel šířky uložení grafiky /+=normál, 2 nebo 3 může být použito pro dvoj nebo trojnásobné uložení/
DMPHM	21	násobitel výšky uložení grafiky /1=normál, jiná než 1=zdvounásobení/
GCM1	22	/9 bajtů/počáteční zprávu odeslat k tisk před uložení
GCM2	31	/8 bajtů/ zpráva odeslána před každou uloženou řadou.
GCM3	39	/7 bajtů/ konečná zpráva odeslána k tisku pro uložení/první bajt v každé zprávě je číslo znaku/
TABVAR	47	0, jestliže tiskne čárku bude TAB mít 16 sloupců, jiná než 0 jestliže 8 sl upcůTAB je použito Obvykle 0.
SOFE	50	Povel pro obrazovku obnovená/zrušená. Obrazovka zčerná, pokud nepoužijete klávesnici po dobu 22 minut. POKE SVAR 50,1 zruší tuto vlastnost
TPROMPTS	51	bit 0=1 k zastavení tisku jména řádků během LOAD bit 1=1 k pohotovému zastavení SAVE pásku
BGFLG	52	blok grafických obrazů. 0 pokud je BLOCKS 4, jiné než 0, pokud je BLOCKS 0
FL60R8	53	0 pokud je použit v modu 3, 6bodů široký znak, jinak jiné číslo než 0
CSIZEH	54	výška znaku
CSIZEW	55	šířka znaku
UWRHS	56	vyšší okno pravý sloupec
UWLHS	57	vyšší okno levý sloupec
UWTOP	58	vyšší okno horní řádek
UWBOT	59	vyšší okno spodní řádek

jmeno	SVAR čís.	
LWRHS	60	nižší okno pravý sloupec
LWLHS	61	nižší okno levý sloupec
LWTOP	62	nižší okno horní řádek
LWBOT	63	nižší okno spodní řádek
MODE	64	mód obrazu / minus 1 /
YCOORD	65	běžný grafický koordinát Y, 0 v horní části
XCOORD	66	běžný grafický koordinát / 2 bajty/
LASTH	513	předešlá klávesa stisknuta
KBHEAD	520	klávesa v hlavě klávesnice
REPDEL	521	brání před opakujícími se klávesami
REPSPD	522	rychlost kterou se klávesy opakují
CHARS	566	/2 bajty/adresa 256 bajtů níže hlavního uložení znaků
ERRSOUND	568	délka chybného zvuku
CLICK	569	délka cvaknutí klávesnice
KPFLG	615	funkční klávesy, pokud lichá, čísla běží, pokud sudá
FRAMES	632	/3 bajty/ TV vytváří, dokud byla puštěna
UDG	635	/2 bajty/ adresu znaku obkreslí jako CHR\$ 144
HUDG	637	/2 bajty/ adresu znaku obkreslí jako CHR\$ 169

Přerušování na SAM Coupé

Přerušování je hardwarový signál k mikroprocesoru Z80B. Když je tento signál přijat, mikroprocesor dočasně zastaví běžnou úlohu a provede přerušení prováděného postupu. Jakmile byl tento postup úplný, procesor se vrátí k přerušené úloze. Jsou zde dva typy přerušení, jeden se nazývá NMI a další INT. Obe dvě přerušení jsou spínače na procesoru Z80B.

Přerušení NMI

Tento typ přerušení není obvykle říditelný programátorem, ale SAM Coupé je opatřen tlačítkem NMI. Pokud je toto tlačítko stlačeno, běžná instrukce je kompletována do procesoru a potom adresuje 0066H v ROM, kde může být řízen na adresu k provedení určitých úkolů.

Přerušení INT

Na Z80B jsou 3 přerušovací módy, a až 2. Na další detaily těchto módů Vás odkazují na technickou specifikaci mikroprocesoru Z80B. Přerušovače INT jsou říditelné uživatelem.

Přerušování mód 0 - s tímto přerušováním to je schopnost vnějšího přerušování pochopit IOPQ a M1 /přerušování potvrzeno/ a polohu jediné instrukce v údaji, procesor reaguje na vykonání této instrukce.

Přerušování mód 1 - přerušování způsobí procesor ke kompletaci běžné instrukce a přechod na adresu 0038 Hex, ve které programátor umísťuje přerušování které potřebuje v kódech.

Přerušování mód 2 - to je nejsilnější mód. Dovoluje programátorovi adresovat plochu řízených adres. To jsou ukazatelé k obsluze pravidelného postupu pro rozdílné žádosti. Vyšší bajt řízené adresy je nahrazen registrem I a přerušovaná žádost nahrazuje nižší bajt v údajích.

Používání přerušování u SAM Coupé

Coupé má 6 možných přerušování dostupných programátorovi. Je zde 5 pevných přerušovačů a další je dostupný přes rozváděcí interfejs. Všechna vnitřní přerušování jsou menší než 20 ns.

Těchto 5 přerušovačů slouží v BASICu v modu 1. Pokud bude programátor měnit mód přerušování, potom tyto přerušovače nebudou sloužit, ledaže je CODE takto upraven.

5 přerušování které Coupé poskytuje:

1. Přerušování řádku

Toto přerušování je dosažitelné programátorovi skrze systémový registr 249. Obrazovka počítače se skládá ze 192 řádků /0 až 191/. Programátor může výstup adresovat řádku v rozsahu 0 až 191, počítač potom přerušuje procesor, když má celý řádek, který byl vybrán. K odvrácení přerušování použijte OUTPUT 200 na port adresy 249.

2. Přerušování myši

Toto přerušování nastane pokud je změněno postavení myši.

3. Přerušování MIDI IN

Jestliže kanál MIDI obdrží údaj bajtu, pak je přerušování odesláno k procesoru, tento je zmocněn přijmout údaje k přečtení.

4. Přerušování uspořádání

Když uvažovaná stavba byla úplná procesor je přerušen. Uvažovaná stavba je úplná každých 20 msec.

5. Přerušování MIDI OUT

Když programátor chce výstupy skrz MIDI kanál toto přerušování nastává pro informování programátora že odeslaný údaj je úplný a další údaj může být odeslán.

Postavení registru

Tento registr opravňuje uživatele ke zjištění které z přerušování požaduje obsluhu. Když nastává přerušování původ registru bude ukázán jako 0, nepřerušovaná hesla budou 1. Jestliže všechna hesla ukáží 1, když má nastat přerušování pak přerušování obdrželo z nějakého vnějšího zařízení připojeného k některému konektoru.

Použití vnější RAM na SAM Coupé

Okruh diagramu je obsažen v tomto horním údaji, který dovoluje připojení vnější RAM k Coupé, pak může být přepnutá v ROM SAMa. Poznáte z technického manuálu, že SAM Coupé užívá adresu AD15 místo AD14 ke zpřístupnění horních 16 K ROM. Příčinou je, že ROM je adresováno následovně:

0000H	4000H	8000H	C000H	FFFFH

ú ROM 0	ú	ú	ú ROM 1	ú

Kde ROM 0 je dolních 16 K ROMky, když AD 15 je 0 a ROM 1 je horních 16 K ROMky, když AD 15 je 1.

Když rozpětí RAM je spojeno s expanzním portem a dvojitá hlava přepínače je v poloze vypnuto /OFF/, potom RAM může být přístupná vybrané vyšší paměti stránky 2 a adresována na adresu 32768

K použití RAM jako ROM obvod musí zpřístupnit RAM jako blok 32 K spíše než 2 bloky 16K. Proto obvod objeví adresu řádků AD 14 a AD 15 a vstupy signálu z RAM na AD 14. To udělá pouze když je přepínač v poloze zapnuto / ON/.

Také, když přepínač je v poloze zapnuto / ON / signál ROM CSL z ASIC je obsažen a proto ASIC očekává ROM skrze expanzní konektor.

Vemte ne vědomí, že všechny adresy a údaje řádků z RAM musí být napojeny na expanzní konektor.

POPIS SAM-BASICu

Názvy číselných proměnných musí začínat písmenem. Čísla, písmena, podtrhávání a mezery by měli následovat za prvním znakem, v počtu max. 32 znaků / mezery nepočítaje /. S cyklickými proměnnými se pracuje jako s běžnými číselnými proměnnými.

Názvy řetězcových proměnných a proměnných polí jsou podobné, ale jsou omezeny deseti znaky / mezery nepočítaje /. Za názvy řetězců a řetězcových polí následuje " \$ ". Řetězce mohou být dlouhé až 65520 znaků. Pole mohou obsahovat celou pohotovostní paměť a i bez indexu může být přes 64K. Programové řádky mohou mít až 127 výrazů a mohou být dlouhé až 16127 bytů.

Klíčová hesla jsou běžně napsány v nezkrácené podobě, ačkoliv klíče mohou být naprogramovány, aby vytvářely úplná slova. Je-li klíčové heslo rozpoznáno, je v listování převedeno do velkých písmen. Klíčové heslo za kterým následuje písmeno nebude rozpoznáno, takže printx je považováno za název procesoru, ale print x a print 1 se stanou PRINT x a PRINT 1.

Stisknete-li klávesu EDIT objeví se v editovací oblasti řádek s " > ". Zadáte-li číslo řádku a pak stisknete EDIT, objeví se pro editování požadovaný řádek. Kurzor může být přemístěn vlevo, vpravo, nahoru nebo dolů v editovaném řádku pomocí kurzorových kláves. Klávesa mazání vymazává doleva, přeřazení maže v pravo / shift + delete /

Kurzor: SAM používá naopak " * " jako kurzor pro nižší případ a naopak "+" jako kurzor pro vyšší případ. Na přání tyto mohou být zaměněny za jakýkoliv jiný znak.

PALETA

Paletu SAMa si můžete představit jako desky podivných tvarů jednoho z malířů, které mají 16 poloh k dispozici pro nádobky s barvou. Zásobníky barev vytvářejí ve skutečnosti 128 barevných odstínů, ale okamžitě můžete použít pouze 16 barev.

Zadáte-li PEN 1, znamená to " na popředí grafiky a znaki použij nádobku v pozici 1 na své paletě ". Obsahuje-li nádobka ve sku-

tečností modrou barvu, poznáme to /podobně jako použití INK 1 na SPECTRU /, ale mohla by obsahovat jakoukoliv barvu vytvořenou výrobcem / jednu ze 128 / a tam není potřeba žádný vztah mezi hodnotou pro PEN a barvou viděnou ve skutečnosti.

Zapnete-li SAMa paleta se nastaví tak, že PEN 0-7 jsou jako INK 0-7 na SPECTRU /černo - bílý / a PEN 8 - 15 jsou jako INK 0 - 7 - s BRIGHT 1. SAM umožňuje například PEN 1; BRIGHT 1 jako alternativu k PEN 8. PEN 2 bude udávat červenou a PEN 10 / nebo BRIGHT 1; PEN 2 / bude udávat světle červenou. PEN 0 a PEN 8 obě dávají černou /BRIGHT black na Spectru vypadá také černě/ protože obě pozice na paletě jsou zavedeny stejnou barvou. V první řadě nastavte do modu 3 / 4 barvy / první čtyři pozice na paletě jsou předdefinovány na černou, modrou, červenou a bílou - jinak by jste pravděpodobně použili černou /modrou/ červenou/purpurovou. Přepnutí do jiného modu by uvedlo do původního stavu předešlé barvy na prvních čtyřech pozicích palety. Přepnutí zpět do modu 3 by použilo barvy, které jste používali v modu 3 jako poslední.

PAPER říká počítači jaký údaj palety použít jakou barvu pozadí pro čištění obrazovky, posun dat vertikálně nebo horizontálně a tisknutí.

Paleta je řízena příkazem PALETTE.

PALETTE používá sám sebe, PALETTE maže počáteční barvy a nastavuje všechna přepínání palety.

PALETTE p,c - nastavuje paletovou pozici p/ é - 15 / na barvu c / 0 - 127 /

PALETTE p,c,b - nastavuje barvy na paletovou pozici p a zaměňuje je pak asi 3 krát za sekundu k dosažení zářících barev.

PALETTE p,c LINE Y - nastavuje dočasně údaje palety, když je dosaženo určené souřadnice Y, až do spodního okraje obrazovky a dovoluje různé palety v různých částech obrazovky / a BORDER/. Lze zobrazit více než 16 barev najednou díky tomuto prudkému přepínání palety. Je možno zobrazit všech 128 barev.

PALETTE p,c,o LINE Y - dočasně nastavuje zářící barvy. Zvolte řádek, několikánásobné změny palety mohou být nastaveny tak, aby se vyskytovali na tomtéž místě / řádku /, ale pouze první dvě se budou nacházet v hraniční oblasti; více jich by mohlo být příčinou "zoubkování" kdyby změněné údaje palety byly provedeny včas. / Kdyby se zaměnilo např. Běžná barva papíru je 0 a údaj palety 0 /

Také je-li příliš mnoho změn palety na jednom řádku - mohlo by se stát, že díky tomu by nějaká změna na následujícím řádku chyběla. V tom případě také všechny pozdější změny palety budou chybovat. V zobrazení může dojít k max. 127 změnám.

PALETTE p LINE Y - smaže jakoukoliv změnu barvy, kterou by způsobila paletová pozice p v řádku y.

Protože vše na zobrazení se může měnit přímo tak, jak se mění paleta, jsou možná některá působivá provedení

PEN - je jako INK na Spectru, ale dovoluje hodnotu 0 - 15. / Může se zobrazit jako INK - v listování se objeví jako PEN /.

Některé příklady:

```
PRINT PEN 5;PAPER 2;"SHRIEK"
```

```
CIRCLE PEN 2; x;y;r
```

BASIC Spectra umožňuje INK 8 nebo PAPER 8 což znamená "průhledný" a 9 s významem "kontrast". SAM používá PEN nebo PAPER 16 a 17 pro totožné případy, ale tyto pracují pouze v módech 1 a 2 / které mají symboly spíše než barvu stupně obrazového prvku / neprojevují se v jiných módech. BRIGHT 8 / nebo 16 / a FLASH 8 a nebo 17 / nastavují průhledný BRIGHT nebo FLASH. FLASH stupně symbolu pracuje pouze v módech 1 a 2.

INVERSE - Po INVERSE 1 jsou znaky spíše tištěny barvou PAPER na pozadí v barvě PEN než naopak. Grafika také používá barvu PAPER. PUT používá komplementární barvy / t.j. barva každého obrazového prvku je tvořena 15-ti běžnými barvami /

OVER - OVER 0 pracuje jako na Spectru. OVER 1 je příčinou toho, že body, řádky a znaky jsou nonekvivalentní s obrazovkou. V módech 1 a 2 vytváří efekty jako Spectrum. Například znak opakovaně natištěný přes sebe pomocí OVER 1 bliká / znak-nezaznamenal / protože znak nonekvivalentní se sebou též má a takto

prázdný PAPER;barva;PAPER je jakákoliv.

Ale v modech 3 a 4 způsobují nonekvivalentnost kroku se sebou /ten je částí zobrazení barvou z paletové pozice 0 / vytváření nul - běžně černá - i když je do každého prvku vestavěna barva.Výsledek je rozdílný od modu 1 a 2,ledaže běžný PAPER je 0

Platné od 27.10.1989

DRAW,CIRCLE,PLOT a PUT reagují také na OVER 2 nebo OVER 3.OVER 2 udává sčítání každého prvku na obrazovce,OVER 3 udává násobení.Jejich činnost je srozumitelná,jestliže berete v úvahu barvu obrazovky BOR / nebo BAND /,používající 4 bitová binární čísla./OR je vhodné pro značkové diagramy a jiné bez ohraňování znaků/.

KEYBOARD

Jakákoliv klávesa klávesnice ve stavu NORMAL,CAPS SHIFT,SYM SHIFT nebo CONTROL může volit jakýkoliv kód / 0 - 255 /.

69 kláves a 4 posuvy udávají 276 klíčů v plánu klávesnice.

Tato mapa má zdánlivě náhodné uspořádání,které odpovídá matici klávesnice.Dodatek ukáže schema klávesnice a polohy v mapě plus počáteční kódy volené každou polohou.Např. mapová poloha 0 má v tomto 98 - tak CHR\$ 98 / "b" / je vytvořeno když je stisknuta klávesa,která ve skutečnosti odpovídá mapové poloze, aniž by bylo použito posunu,nebo převedení.Ale celá mapa může být na přání změněna.

KEY_posn,x - např.KEY 23,134 přikazuje poloze klíče 23 v mapě klíčů vytvořit po stisknutí CHR\$ 134

KEY 0,65 by měli dělat běžné "b" do CHR\$ 65 /"A"/.

Pozice musí být 0 - 275,x musí být 0 - 255.

Jestliže vycházejí z klávesnice určité kódy,vyhledávání se dá příkaz aby sledovalo zda je rozšířená definice pro ně vytvořená.Kódy 192 - 254 mohou být definovány tímto způsobem.Jestliže neexistuje definice budou CHR\$ 192 - 254 použity,jinak se použije rozšířená definice.To umožňuje např.funkčním klíčem být definovány jako RUN / ENTER / nebo LIST /ENTER/.Viz níže.

DEF_KEYCODE

Např. DEF KEYCODE 195:PRINT 123:PRINT "z" vyvolá jakýkoliv následující kód ze 195 vycházejících z klávesnice, aby mohli být rozšířeny do:PRINT 123:PRINT " z " /ENTER /. Toto vytváří 123z se objeví na horním okraji zobrazovací plochy. Celý řádek za DEF KEYCODE 195: je využitý. Je-li poslední dvojtečka přiřazena k řádku, je automatické ENTER potlačeno, např:

```
DEF KEYCODE 193:PRINT "asd":
```

by vytvořilo

```
PRINT " asd " v editovací oblasti
```

DEF KEYCODE může být použito také s řetězcem, spíše než zbytek řádku. Např:

```
DEF KEYCODE 192,"TESTING"
```

vyvolá jakýkoliv následující kód ze 192 vycházejících z klávesnice, aby se rozšířily do "TESTING" /ENTER/.

/Toto by vyvolalo zprávu o chybě/

DEF KEYCODE 192,"TESTING:" by potlačilo ENTER, protože posledním znakem je dvojtečka.

Jednotlivé definice mohou být smazány např:

```
DEF KEYCODE 195: / nic nenásleduje/ , nebo DEF KEYCODE 195," "
```

! DEF KEYCODE ERASE odstraňuje všechny definice !

Jestliže se vyčerpá prostor uvnitř paměti používané DEF KEYCODE je ohlášeno " Too many definitions " - příliš mnoho definic

LPRINT

Podobný k PRINT, ale pracuje přes proud 3, běžně otevřen do kanálu " p " /paralelní tiskárna /. Tam je pružný systém pro převedení znaků s kódy >= 128 do sledu znaků dovolujících např. nastavit alternativní znak tak, aby byl vybrán, zatímco se tiskne.

PRINT

Kontrolní kódy 8 - 12 pracují tak, jak by ste předpokládali, např. PRINT CHR\$ 8 krokuje zpět, 9 přesouvá vpravo, 10 přesouvá dolů, 11 přesouvá nahoru a 12 maže. Když SAM začne pracovat, CHR\$ 128 - 143 tvoří blokovou grafiku ve stylu Spectra.

CHR\$ 144 - 168 jsou cizí UDG. CHR\$ 169 - 254 vypadají stejně jako CHR\$ 32 - 117, ale mohou být také podle vyspělosti uživatele použity jako UDG. Ve skutečnosti je celá sada znaků vložena do RAM / paměť s přímým přístupem /, takže je snadné měnit jakékoliv znaky.

SAM používá znakové kódy nad 128 jako kódy pro oba UDG /blokovou grafiku a znamení pro příkazy v BASICu.

LIST používá znamínka úvozovek k odlišení co by se mělo ve skutečnosti objevit na obrazovce. Např:

```
10 PRINT " foreign set "
```

PRINT a cizí písmeno by mohly obě být uloženy jako tentýž kód v programu, ale skutečnost, že toto je výpis a PRINT je vně znamének úvozovek, rozhoduje o tom, že se zobrazují odlišně. Ale kódovací systém bude patrný jakmile při změně řádku budete znázorňovat cizí znaky - jsou-li vně úvozovek dostáváte přímé klíčové heslo. Toto může být užitečné.

TAB - dovoluje tisknutí v podobě sloupců až do šíře obrazovky, tak např. PRINT TAB 82;"x" je právoplatné používáte-li zrovna mód 3 se znaky širokými 6 obrazových prvků. /TAB 84 je sloupec napravo/.

AT - umožňuje hodnotám řádků a sloupců jít přes zaplněnou plochu obrazovky.

BLOCKS

BLOCKS_0 - vypíná blokovou grafiku a dovoluje CHR\$ 128 - 143 pracovat jako UDG. /Jsou předdefinovány jako cizí znaky/.

BLOCKS_1 - zapíná zpět blokovou grafiku /UDG mohou být ještě přivedeny zpět pomocí BGRAPHICS 0 - jsou ještě tam.

DEF_FN

Názvy funkcí mohou mít jakoukoliv délku, musí začínat písmenem a pokračovat písmeny, čísly, nebo podtrhávkami. "\$" ukončuje název řetězcové funkce. Např:

```
DEF FN LEFT$ /a$,n/= a$ / to n /
```

```
DEF FN LEFT$/b$,3/ by napsal první 3 znaky b$
```

Funkce nesmí mít závorky, jestliže tam nejsou žádné argumenty.

Např:

```
DEF FN charset = DPEEK 23606 + 256
```

PRINT FN charset by udávalo umístění souboru znaků. Umístění výrazu DEF FN nemá vliv na rychlost provádění programu.

Názvy proměnných použitých uvnitř závorek jsou umístěny do FN; mohou být dlouhé max. jedno písmeno.

RECORD__TO__a\$

Příkazy řetězcům zaznamenat grafiku pro rychlé zopakování. Např:

```
RECORD TO a$ : FOR n = 1 TO 100: DRAW TO RND /255/,RND/175/:  
NEXT n
```

Ukládá všech 100 sepsaných příkazů do a\$. /Který je zaveden jako " " čímž se odstartuje/. Také dochází k běžnému zobrazovacímu ději.

Po zapnutí nahrávání budou následující data nahrána v řetězci.

CIRCLE

DRAW a DRAW TO obsahují náčrtek křivky/křivky jsou zaznamenány jako řada přímek, které zabírají prostor, ale mnohem rychleji se zopakují.

PLOT

PEN / např. PEN 2: ne PLOT PEN 2;x,y /

PAPER

OVER

CLS

PAUSE

RECORD STOP - vypíná záznam grafiky. RECORD pracuje ve všech módech

BLITZ_a\$ - Provádí řetěz grafických příkazů. Velice rychle, zvláště ve 4. módu. Zakřivení DRAW je mimořádně urychleno. Pracuje ve všech módech. BLITZ reaguje na proměnné se změněným grafickým měřítkem. /Viz konec spisu/ taky že tvar může být rychle přehrán kamkoliv záměnou XOS a YOS. Velikost přehraného tvaru může být změněna výměnou XRG a YRG. Toto pomalu srazí BLITZ, ale výsledek může být minimalizován změnou rozsahů mocninou 2. /Zdvouje je, nebo půlí./

Toky_a_kanály

CLOSE - uzavřen : OPEN - otevřen

Jako u Spectra /ačkoliv to manuál Spectra nevysvětluje/. Toky 0 a 1 jsou běžně otevřeny do kanálu "K"; PRINT #0; "test"; PAUSE se napíše na spodním okraji obrazovky. PRINT #2 bude napsán v horní části obrazovky, jak je běžně /používaný kanál "S"/ a PRINT #3 účinkuje jako LPRINT /používaný kanál "P"/. Můžete také udělat něco jako OPEN#6;"P":PRINT#6;"testing" pokus odeslat materiál do tiskárny přes tok 6 nebo OPEN#2;"P" takže běžný výstup obrazovky půjde místo toho do tiskárny. To vše je jako u Spectra. Ale INPUT#2 se může - to je nové. Např:

INPUT#2;"hodnota:";V - Vám dovolí znázorňovat v hodnotě V na horním okraji obrazovky, napříště do "hodnota" rachle. Řádek nelze mazat poté, co stisknete odeslání, na rozdíl od nižší části obrazovky.

Příkaz RECORD pracuje přes proud 16 /který není na Spectru/. Proud 16 tiskne do řetězce proměnnou určenou RECORDem a tak to bude ještě dělat po RECORD STOP. To by mohlo být užitečné některým uživatelům. Např: RECORD TO a\$:RECORD STOP:DIR#16;1 by měl umístit katalog disku v a\$. Jiné proudy odděleně od 16 mohou být otevřeny řetězcové proměnné používající např. OPEN#5;"\$"

OPEN#s;"b"/nebo "B"/ dovoluje proudu S poslat cokoliv do tiskárny - to je užitečné pro poslání kontrolních kódů. /Problém s "P" proudem je takový, že kontrolní kódy jsou aktivovány a nesmějí být poslány do tiskárny - např. TAB kontrolní kód pošle do tiskárny počet mezer/. Kanál "P" ve skutečnosti používá kanál "B" pro výstup, pak modifikuje data, co přijímá. Proto všichni pokročilí uživatelé potřebují převádět výstup do řekněme pohonné jednotky sériové tiskárny, která mění výstupní adresu kanálu "b".

DIR,ERASE,FORMAT,MOVE

Nepracuje bez diskové verze SAMa. Vysvětlení najdete v manuálu DISK DRIVU.

SAVE,LOAD,MERGE,VERIFY

Umožňuje manipulaci s programy, řetězci, poli, strojovým kódem a obrázky z obrazovky.

SAVE "název" CODE začátek, délka, prováděcí adresa umožňuje to, že LOAD "název" CODE začne pracovat od dané adresy.

MERGE "název" CODE potlačí samospouštění strojových kódů.

LOAD "název" DATA b\$() pracuje, ale je možné i např. DATA addr\$() nebo DATA abc\$. Všimněte si, že např. SAVE "název" DATA blit\$ je O.K. dokonce i když je blit\$ jednoduchým řetězcem jakým je řetězec BLITZ.

LOAD "název" LINE n zavádí program a jde na řádek n. To převádí jakékoliv číslo auto-startovacího řádku uložení s programem.

SAVE "název" SCREEN\$ ukládá obrázek z obrazovky, běžný mód a paletovou informaci pro pozdější opětovné zavedení. /Podle potřeby LOAD automaticky přepíná módy/

DEVICE

Tento nový příkaz přikazuje SAVE, LOAD, MERGE a VERIFY pracovat s diskem nebo sítí.

DEVICE d - znamená použij diskovou paměťovou jednotku č. 1

DEVICE d1 - totéž co DEVICE d, nebo DEVICE D1

DEVICE N5 - použij síťovou stanici č. 5

DEVICE T - použij pásek, rychlost uložení obdobná jako u Spectra

DEVICE t20 - použij pásek rychlost uložení 20. /Rychlost 105 je rychlost Spectra, 20 je mnohem rychlejší, 255 je pomalá/. Vyšší rychlosti jsou méně spolehlivé, ale 3x rychlost Spectra by mělo zvládnout mnoho magnetofonů. LOAD se automaticky přispisuje rychlosti pásku, který byl použit na uložení souboru.

AUTO

Automatické vytváření čísla řádku. Např.: AUTO 100, 5 - ukazuje čísla řádku od 100 v krocích po 5-ti. AUTO samo o sobě se zapne na běžném řádku plus 10, a bude krokovat po 10-ti

BEEP_d,p

d - je doba trvání v sekundách. Musí být kratší než 16 sec., nebo bude dána zpráva "nota příliš dlouhá". Výška /tónu/ musí být mezi -60 a +71 nebo se ohlásí "neplatná nota"

Frekvence not v cyklech/sec = $440 \times 2^{((P-9)/12)}$

BOOT

Zavádí DOS /diskový operační systém/ z disku.

BORDER

Umožňuje BORDER 0 - 15 tak, že může být zvolen jakýkoliv údaj o paletě.

CALL

Např: CALL adresa, x, y, fred\$. Viz technická část

CIRCLE

Jako u Spectra, s výjimkou, že kruh může přesahovat přes horní okraj a rychlost je asi sedmdesát krát větší! Také kruhy jsou více symetrické. Jestliže se kruh přesahuje dost daleko od vršku obrazovky bude domalován na spodní část, naopak půjde-li z vrchu.

To může vytvořit pěkné vzory.

CLEAR

Jako u Spectra až na to, že forma CLEAR n je omezena pouze množstvím paměti určené pro BASIC. Např: Je dovoleno CLEAR 90 000.

CLS

CLS nebo CLS 0 čistí celou obrazovku. CLS 1 čistí pouze okénko tohoto textu.

CLS#

CLS se symbolem kontrolní informace dělá: OVER 0: INVERSE 0: FLASH 0: PAPER 0: PEN 7: BORDER 0: PALETTE I dělá CLS

CONTINUE

Pokračuje po chybách

CSIZE w, h

Velikost znaku. Umožňuje změnu horizontální a vertikální rozteče znaků. Např: CSIZE 6, 9 udává znakové buňky o šířce 6 obrazových prvků a výšce 9 obrazových prvků. w musí být 6, 8, nebo 16 /6-ti a 16-ti prvkové rozmístění prvků má efekt pouze v módu 3/. h může být 6-32, ale 6-ti nebo 7-mi prvkové výšky vyžadují předefinování souboru znaků. Dvakrát vysoké znaky se používají při h=16 a více. CSIZE také dělá CLS

DELETE

DELETE n TO m maže část řádků programu. Je-li vynecháno n, použije se 1. Je-li vynecháno m, je použit poslední řádek v programu. Je-li n=m může být vymazán jediný řádek. Neexistuje-li ani n ani m - pak nebude mazán žádný řádek, jestliže tam žádný není.

DIM

Je dovoleno max. 10 znaků dlouhý název /nepočítaje mezery/. Např:
DIM název \$ (10 000, 20) nebo DIM coords (100, 2)
Čísla v závorkách /rozměry/ musí být 1 - 65535. Celková velikost pole je omezena jinak pouze disponibilní /dosažitelnou/ pamětí.

DO, DO WHILE /podmínka/, DO UNTIL /podmínka/

Startuje část programu, která skončí LOOPem. /Ta je nazvána DO-smyčka/. Na začátku smyčky DO umožňuje část programu aby se provedla buď bezpodmínečně / v tom případě je DO jen označením místa, do kterého má smyčka zpět skončit/, nebo podle toho zda je podmínka splněná nebo ne. Je-li část programu neproveditelná, skočí do ní výraz pro LOOP.

LOOP, LOOP WHILE, LOOP UNTIL /podmínka/

Umožňuje podmíněné, nebo nepodmíněné cyklování na konci DO-smyčky. /Podle potřeby mohou být podmínky použity při obou - vchodu i východu ze smyčky /.

EXIT IF

Opouští DO smyčky v prostředku a zároveň jde - je-li podmínka splněná na výzvu za LOOP. Jestliže podmínky nejsou splněné, pak pokračuje.

LOOP IF

Odbíhá od prostředka DO smyčky zpět k DO jestliže je splněna podmínka. Jestli není, pak pokračuje.

DO - smyček může být vloženo /do velikosti 255/.

LIST FORMAT 1 nebo 2 pomáhá vydat jejich strukturu. Např:

```
10 LET tot=0
20 DO WHILE tot < 100
30 DO
40 GET KEYVAL
50 LOOP UNTIL keyval < =9
60LET  tot=tot+keyval
70 PRINT tot
80 LOOP
```

DRAW

RDAW x,y udává relativní směr,DRAW x,y,z dělá to stejné ale s křivkou,DRAW TO x,y udává absolutní směr,DRAW TO x,y,z je to stejné,ale s křivkou

DUMP

Vede kopii nestíněné obrazovky v jakémkoliv módu do tiskárny typu EPSON.V módech 1 a 2 budou obrazové prvky souboru kopírovány černě.V módech 3 a 4,jakákoliv barva,která není barvou běžného obrazovkového pozadí /nastavení PAPERem/ se bude kopírovat černě.Změnou systémových proměnných může být zvolena dvojitá šířka ,nebo výška a kopírována plocha.

POZNÁMKA:

Grafický výpis z paměti na obrazovku bude v módu 3 dvakrát širší než v ostatních módech,protože obrazovka má křížem 512 obrazových prvků namísto 256.

DUMP_CHR\$

Zhotovuje textový výpis z běžného okénka obrazovky.Měl by pracovat s jakoukoliv tiskárnou.Mnohem rychleji zhotoví text než kopii grafiky.

GET -čekej na klíč

Např:GET X.Poté co je klíč určen,x=1 jestliže se našlo 1,2 jestliže se našla 2,A jestliže se našlo a pod.GET A\$ bere jednoprvkový řetězec.Např:

```
10 DO:GET a$:PRINT a$;:LOOP /smyčka cyklus/ píše klíče tak,jak
jste je našli
```


FATPIX ,FATPIX + přikazuje módu 3 použít "fat" /kvadratické/ obrazové prvky a grafiku XRG /vzdálenost x / z 256 ,umožňující vertikálním a horizontálním řádkům mít totožnou výšku /tloušťku/ a kružnicím být kulaté.

FATPIX 0 /počáteční stav/ přikazuje móde 3 použít "úzké" obrazové prvky, které mají poloviční šířku než v jiných módech. XRG je nastaveno na 512 takže PLOT 511,175 nebude používat z obrazové prvky z pravé horní části obrazovky.

FILL

Např: FILL PEN 3,x,y zavádí jakýkoliv tvar jedné barvy určené údajem PEN 3./Jakákoliv jiná barva než ta, která původně slouží v x,y jako hranice/. FILL USING a\$,x,y zavádí jakýkoliv tvar obrazcem určeným a pomocí a\$. Tento obrazec může být nejvhodněji získán z obrazovky použitím např:

GRAB a\$,x,y,16,16 / v módu 4, nebo v módu 3 s "širokými" obrazovými prvky/ nebo GRAB a\$,x,y,32,16 / v módu 3 s "tenkými" obrazovými prvky/. FILL běžně připravuje tzv. pracovní kopii obrazovky aby pomohla zavádět složité tvary / to dovoluje konečné zavedení jakéhokoliv tvaru/. Při plnění malých ploch trvá operace dlouhou dobu. K urychlení procesu se používá varianta FILLu, která takovou tzv. pomocnou kopii neprovádí - ale protože je stále žádán pomocný zápis tak ji musí provést první FILL v sérii. Následující FILLy ji už dělat nemusí za předpokladu, že zavádějí přes stejnou barvu jako první. Např:

CIRCLE 64,88,30

CIRCLE 150,70,40

FILL PEN 2,64,88

FILL PEN 4,70,40,1 /poslední "1" zastavuje vytváření pomocného zápisu/

/také je možno končit údaje za FILL PEN nulou, která dovoluje vytvářet pomocný zápis/

GO TO

Čísla řádků mohou být až 65279. LABEL se může použít. Např:

GO TO newe/viz LABEL/

GO SUB - totéž

GRAB a\$,x,y,w,l

ukládá plochu obrazovky s vrcholem x,y vlevo, šířkou w, délkou l v obrazových prvcích do řetězce. V módech 4 nebo 3 / s širokými obrazovými prvky/ je souřadnice x zaokrouhlována dolů na celou hodnotu a šířka w je zaokrouhlována nahoru na celý počet prvků. V módu 3 s tenkými prvky jsou hodnoty x a w převáděny na násobek čísla 4. GRAB a PUT pracují pouze v módech 3 a 4

PUT x,y,a\$

Umísťuje uloženou plochu na obrazovce. X je zaokrouhleno nahoru na celou hodnotu. Reaguje na INVERSE /např: INVERSE 1: PUT x,y,b\$ nebo PUT INVERSE 1;x,y,c\$/ OVER 0 připsuje vše, co tam už je, OVER 1 udává nonekvivalentnost, OVER 2 udává logický součet, OVER 3 udává logický součin. Nereaguje na INK nabo PAPER - jste spojeni s barvami původně získanými GRABem.

PUT x,y,a\$,m\$

Druhý řetězec, který musí mít stejnou délku jako první, pracuje jako "maska" k určení které prvky budou použity z prvního řetězce. Bity v druhém řetězci by měli být nastaveny na 1 aby se použily data z prvního řetězce - kdyby byly 0, zůstává v tété posici samotné původní zobrazení. To umožňuje zobrazit složitý tvar na pozadí jako by byl vystřižen, tedy bez hranice. Samozřejmě to by mohlo být zdoluhavé kreslit "masku" rukou. Ale ještě lepší je komplexní metoda.

1. vytáhněte původní tvar na obrazovku
2. natáhněte kolem něj "schránku" o něco málo větší, než je plocha, kterou budete GRABovat.
3. Plňte "schránku" PENem 0 abyste udali hranici nul /jestli je potřeba/
4. GRABujete plochu
5. Plňte hraniční plochu PENem 15 abyste udali hranici jedniček
6. Přeložte uloženou plochu přes sebe použitím OVER 1. Hraniční jedničky získáte nonekvivalentností hranic PEN 0 a PEN 15; hraniční nuly získáte nonekvivalentností vnitřního obrázku se sebou. To je pravý opak toho, co potřebujeme na masku.

7.Opět GRABujete plochu

8.Vložte její opět použitím OVER 0, INVERSE 1.Abyste změnili 1 na 0 a naopak

9.Opět GRABujete - dostanete použitelnou masku

Volba maskování nereaguje na INVERSE nebo OVER.

IF - krátké IF je opatřeno nepovinným "ELSE" např:

```
10 IF x=1 THEN PRINT "TRUF":ELSE PRINT"FALSE"
```

Else musí být na začátku nového povalu, na rozdíl od některých BASICů. Situace je složitější, je-li na témže řádku několik IF a ELSE. Dole jsou některé příklady aby Vám ukázali které ELSE bude použito různými IF jestliže určená podmínka je nepravdivá

----- v

```
IF-THEN stat.:IF-THEN stat.:ELSE stat      /stat = příkaz /
```

----- v

```
IF-THEN příkaz.:IF-THEN příkaz.:ELSE příkaz.:ELSE příkaz
```

----- v

```
IF-THEN příkaz.:ELSE příkaz.:IF-THEN příkaz.:ELSE příkaz
```

Dlouhého IF, které pracuje přes mnohonásobné řádky, lze dosáhnout vynecháním THEN. Např:

```
10 IF x=1
20 PRINT "TRALALA"
30 PRINT 123
25 ELSE PRINT "AAA"
30 PRINT "MORE GUFF"
40 END IF
```

Toto by dalo "AAA" a MORE GUFF" jestliže $x \neq 1$. To není potřeba u řádků, které obsahují pouze jeden povel. Vzor ukazuje výsledek použití LIST FORMAT 2. Je-li vynecháno END IF, je dána zpráva "chybějící END IF", v případě že podmínka po dlouhém IF je nepravdivá a počítač se snaží skočit na END IF.

ELSE IF má zvláštní význam jako povel CASE v některých BASICích.

Např:

```
10 IF x=1
20 PRINT "one"
30 ELSE IF x < 5
40 PRINT "less than 5"
45 PRINT "yes really"
50 ELSE IF x=8
60 PRINT "eight"
70 ELSE
80 PRINT "something else"
90 END IF
100 STOP
```

Je-li x=1 je řádek 20 proveden, pak řádek 100

Je-li x=2,3 nebo 4 řádky 40 a 45 jsou provedeny pak řádek 100

Je-li x=8, řádek 60 je proveden, pak řádek 100

Jakékoliv jiné hodnoty provádějí řádek 80, pak řádek 100. Jednou je jakákoli jedna podmínka pravdivá, pak jsou provedeny řádky nebo příkazy a když se jiné ELSE IF nebo ELSE setkají, skáče program přímo na údaj za END IF. Čísla řádků, počet příkazů na řádku atd. jsou nepodstatné. Vkládání dlouhých IF nebo krátkých IF je možné provádět. Vkládat dlouhá IF mezi krátké IF a jeho skupinové ELSE není možné.

INPUT

Běžně je INPUT v činnosti na 2 spodních řádcích obrazovky, ale INPUT # 2 dovoluje použít horní okraj obrazovky.

KEYIN a\$ - znázorňuje řetězec jako by ste to dělali sami. Např:

KEYIN "100 DATA"+ a\$ + ", "+ b\$ by přičítalo nový řádek. Umožňuje programům psát programy, - např. vytvářet povely DATA z části se strojovým kódem.

LABEL - např: 200 LABEL nebo: PRINT "to je krása"

Název pro LABEL musí být názvem legální numerické proměnné. LABEL musí být na začátku řádku i když je dovoleno, aby mu předcházely barevné řídicí kódy, nebo mezery. Když spustíte

jakoukoli část programu /pomocí GOTO,RUN,GOSUB atd / vytvoří se proměnné s názvem udaným po jakých kolí LABELech v programu a jejich hodnota je nastavena na číslo řádku,na kterém se tento LABEL nachází.Jestli že použijete CLEAR nebo upravujete program,jsou proměnné přetvořeny.Ačkoliv jsou proměnné určeny nejvíce pro srozumitelná GOTO a GOSUB,mohou se použít i s jinými,dalšími příkazy např:LIST.

LET - SAM umožňuje dlouhé názvy řetězce a pole a mnoho LET může být děláno najednou.Např:

LET x=1,flag=2,alfa\$="testování",abc/2/=12

DEFAULT

Vytváří proměnnou jestliže tato neexistuje.Např:

DEFAULT x=2,beta\$="ASD".Použijte hlavně v postupech

LIST,LLIST

Listuje celý program nebo listuje počínaje daným řádkem,jestliže za příkladem následuje číslo řádku.

LIST n TO M,LLIST n TO m

Je-li vynecháno n bere se v úvahu první řádek programu.Je-li vynecháno m bere se v úvahu poslední řádek programu.

LIST FORMAT f,LLIST FORMAT f

LIST a LLIST FORMAT mění formu LISTu a LLISTu.

Je-li f=0,pak se listování neprovedou.Je-li f=1 nebo 2 provedou se jistá složení programu v listování s jednou nebo 2 mezerami.Následující hesla mají vliv na provedení:FOR,NEXT,IF,ELSE,END IF,ON,DEF PROC,END PROC,DO,LOOP,EXIT IF,LOOP IF. V seznamu formátu 1 nebo 2 bude každý příkaz vyhotoven na odděleném řádku.Program SAMa smí mít počet řádků do 65279 a listování si vyhrazuje 6 sloupců na levé straně pro číslice čísel řádků a mezeru.

MODE n

n musí být 1-4.Např:MODE 3 maže obrazovku a volí mód 3.

NEW

Maže program a proměnné,čistí obrazovku atd.

ON_ERROR

Např: ON ERROR GOTO 100 - dojde-li k chybě, jdi na řádek 100.

ON ERROR GO SUB 200

ON ERROR - zpracovatel chyby - dojde-li k chybě, proved' proces zpracování chyby.

ON ERROR STOP - vypína chytání chyb. Je-li vychytávání chyb zapnuto ON ERRORem jsou vytvářeny - ERROR proměnných /chybné číslo bude v dodatku / LINO / řádek, kde došlo k chybě / a STAT /povel kde nastala chyba/.

Zatím co se zpracovává chyba je ON ERROR vypnut/takže chyba vytvoří během svého zpracování normální reakci/.

Údaje za "ON ERROR" jsou pak provedeny /takže ON ERROR musí být použit v programu až na ON ERROR STOP/. Podprogramy a procedury se vrátí do příkladu, který následuje za tím, ve kterém došlo k chybě při použití RETURN nebo END PROC. Dočasné vyřazení ON ERROR se zruší použitím RETURN nebo END PROC. CONTINUE se používá k návratu do příkazu, který byl příčinou chyby /kromě případu, kdy je chybou "BREAK"-přerušeni-pak jde CONTINUE na následující příkaz/. CONTINUE neobnovuje opět činnost ON ERRORu. Může se použít POP /čtení dat / aby přečetl RETURN/END PROC adresu když se rozhodnete namísto vrácení pokračovat. To udržuje správný zásobník BASICu.

ON

Může se použít k volbě jednotlivého příkazu ze soupisu příkazů / zbytek programového řádku/. Např:

ON x:PRINT"ONE":PRINT"TWO":GO SUB 100 bude psát "jedna" jestliže x=1, psát "dvě" jestliže x=2 a půjde na řádek 100 podprogramu je-li x=3. Je-li x=0 nebo větší než počet příkazů za ON, bude program pokračovat následujícím řádkem./Dokonce zvolí-li ON příkaz GOSUB nebo jakoukoliv proceduru, žádné další příkazy na řádku s ON nebudou provedeny - RETURN nebo END PROC jdou na následující řádek.

GOTO ON x;100,200,300 a GO SUB ON x;123,345,230

GOTO nebo GO SUB první řádek je-li x=1 atd. Je-li proměnná ≠ 0 nebo větší než počet řádků v soupisu, pokračuje program následujícím příkazem.

OUT_P,x

Dělá výstupní cestu P,X.P musí být 0 - 65535, x musí být 0 - 255

PAUSE

PAUSE čeká 1/50 sekundy nebo dokud není stisknuta klávesa.
PAUSE nebo PAUSE 0 čeká až se stiskne klávesa.

POKE

Dovoluje běžné POKE adresa,x ale také například:
POKE addr,255,129,129,129,129,129,255 až max.32 čísel
lze napsat najednou po sobě jdoucích adres.
POKE addr,a\$ nebo POKE addr,test" bude do paměti zapisovat
řetězec.Viz také MEM\$
Adresa může být 0 - 524287 / 0 až 512K minus 1 /

DPOKE

DPOKE adresa,hodnota - dělá dvojité zápis.Hodnota musí být
0 - 65535

TPOKE

TPOKE adresa,hodnota - dělá trojitý zápis tvaru používaném mno-
ha SAMy.Systémové proměnné v BASICu.Hodnota musí být 0 - 512K.

PLOT

PLOT x,y nastavuje bod do x,y:Lze např:PLOT PEN 12;x,y

POP

Vyřazuje GO SUB,PROC,DO,RETURN adresu.

POP_x - Převádí do proměnné,např:x, vyřazené zpětné číslo řád-
ku.

Procedury

DEF PROC název; DEF PROC název, parametr, parametr ...; DEF PROC název, DA

Začíná definici názvu procesu. DEF PROC musí být prvním klíčovým heslem na řádce / ale je dovoleno, aby mu předcházely mezery a barevné řídící kódy/. Název procedury musí začínat písmenem, za nímž následuje řetěz písmen nebo čísel, nebo "\$". Mezery nelze použít jako součást názvu. Velká a malá písmena jsou si ekvivalentní.

Za názvem procedury může následovat soupis parametrů, nebo klíčových hesel DATA. Parametry dané v definici procedury musí být názvy proměnných. Jsou samočinně umístěny do procesu.

Proměnné, včetně polí, se mění odkazem, jestliže v soupisu DEF PROC před parametrem předchází REF. Při vyvolávání procesu jsou proměnné bez REFu hodnoty převedeny do hodnot ze soupisu. Je možné vynechat některé, nebo všechny hodnoty z vyvolání procesu a doplnit jimi vnitřní procedura za použití DEFAULT - takže je možné abyste měli proceduru s různým počtem parametrů dost podobné příkazu DRAW /tj. DRAW x,y nebo DRAW x,y,z /.

Když je namísto soupisu parametrů použito klíčového hesla DATA nejsou prováděna žádná přiřazení, ale soupis skutečných parametrů po vyvolání procesu může s READem a funkčními položkami reagovat. Umístění procesu v programu nemá žádný vliv na rychlost programu.

Je-li použit název procedury bez toho, že by byla proněj v programu definice, pak se před ohlášením, že chybí DEF PROC, "soupis vnějších příkazů" ptá na možný sobě rovný protějšek. To dovoluje vyspělejšímu uživateli, aby do BASICu přidali nové příkazy.

END PROC

Označuje konec pojmenovaného procesu a tím dovoluje počítači, aby se vyhnul provádění povelů mezi DEF PROC a END PROC, kdyby se proces nevyvolal. Když je proces v chodu, END PROC ukončuje jeho provádění, smazává jakékoli místní použití proměnné a znovu ukládá jejich původní /globální/ hodnoty, jestliže nějaké byly. Byly-li parametry měněny odkazem /před formálním parametrem předcházelo REF/ běžná hodnota formálního parametru

bude převedena do skutečného parametru při vyvolání procesu. Není-li nalezen formální parametr obdržíte zprávu "Ztráta proměnné". Provádění programu se vrací do povelu za vyvoláním procesu.

Použití END PROC bez DEF PROC vyvolá zprávu " Chybí DEF PROC ".

LOCAL

Určuje soupis proměnných / které mohou zahrnovat pole/ které jsou umístěny v procesu.

RANDOMIZE, RANDOMIZE_n

Nastavuje rozložení náhodných čísel na N. Od zapnutí SAMA nastavuje jej RANDOMIZE na počet snímků.

READ

Zcela standart, ale dovoluje číst řádek s řetězcovými proměnnými. Dovoluje vynechat úvozovky v příkazech DATA. Např:

DATA kočka,žába,"pes",123

READ LINE a\$ by vytvořilo a\$="kočka", pak "žába", pak "pes", pak "123" s po sobě jdoucími READy

DATA

DATA příkazy mohou zahrnovat klíčová slova jako "GOTO" aniž by se projevíly. Na rozdíl od tohoto musí být slova bez úvozek legálními číselnými proměnnými, takže frekvence je správně, ale 1x není.

RESTORE

Norma RESTORE se také dělá při zavádění BASIC programu

REM

Standartní BASIC. Umožňuje zahrnutí poznámek do programu.

RENUM

Přečíslovává celý program od řádku 10 v krocích po 10.

RENUM n TO m - přečíslovává část programu počínaje řádkem n včetně řádku m. Při vynechání n se bere v úvahu první řádek, při vynechání m řádek poslední.

RENUM LINE L - dělá první nový řádek s číslem L.

RENUM STEP S - krokuje po S

Dělič, řádek a krok může být dle potřeby použit nebo vynechán, ale musí bývat v tomto pořadí:

RENUM n TO m LINE L STEP S

GO TO, GO SUB, RESTORE, RUN, LINE, ON atd. jednají se všemi, ale jakýkoli výraz jako GO TO VAL "100 " je bez odezvy./Obrazovka je použita k zastavení dočasných dat během RENUMu a na konci je setřena/.

RUN

Spouští program. RUN n spouští od řádku N

RETURN - Ukončí podprogram

SCROLL_d,p

Posouvá celou obrazovku /včetně editovací části/ po p prvcích ve směru d. 1=vlevo, 2=nahoru, 3=vpravo, 4=dolů.

Prázdné pozadí je posouváno aby změnilo ztracená data. P může být 1, 2, 4, 6, 8, atd

ROLL_d,p

jako SCROLL, ale obraz se obtáčí/BB 3.0 má obdobné příkazy/.

SCROLL_d,p,x,y,w,l

Posouvá pouze tou částí obrazovky, která má vrchol vlevo x, y, šířku w / v obraz.prvcích/, délku l /obraz.prvků/. x je převedeno na celou souřadnici, šířka na celý počet prvků. Pracuje pouze v módech 3 a 4./Módy 1 a 2 udají zprávu "Neplatný zobrazovací mód"/.

ROLL_d,p,x,y,w,l - jako SCROLL, ale s obtáčením

Tzv. náhradní paměť z konce 24K zobrazovací paměti se používá k uložení dat "obtáčení". Pro horizontální přetáčení je pouze pár bytů/obraz.prvky přesunu : 2, +4/. Pro vertikální přetáčení je použitý prostor z prvků šířky přetáčení oblasti / 2xprvky pohybu, + 3 byty/

SCROLL_CLEAR

Blokuje rychle "scroll", které je normálně dáno, v případě, že obrazovka je plná. Obrazovka se posune bez jakékoli zprávy a bez toho, že by byla stisknuta nějaká klávesa.

SCROLL RESTORE - vrací rychle do původního stavu.

SOUND r,d ; SOUND r,d;r,d;r,d....

Posílá byte dat D registru zvukového čipu R. Může se použít max. 127 párů čísel.

OPEN SCREEN n,m

Např: OPEN SCREEN 2,4 rezervuje paměť pro zobrazení ve tvaru 2, které bude odstartováno v módu 4. Je třeba 32K. N musí být 2-15 nebo se objeví zpráva "Vadné číslo obrazovky". Zobrazení nemusí již existovat, nebo se objeví zpráva "Obrazovka je již otevřená". Při zapnutí SAMa je právě jedna obrazovka s názvem 1.

OPEN SCREEN n,m,c

OPEN SCREEN n,m,0=OPEN SCREEN n,m. OPEN SCREEN n,m,1 otvírá obrazovku, ale to je při běžném zapnutí. Je-li v přístroji několik programů najednou, mohou všechny po svém spuštění využít obrazovku.

CLOSE SCREEN n

Např: CLOSE SCREEN 2 uvolňuje paměť obrazovky pro opětovné využití. Nic se nestane, jestliže obrazovka 2 neexistuje.

SCREEN n

Vybírá obrazovku pro BASIC povely aby použily např: OPEN SCREEN 2: SCREEN 2 by vedlo k tomu, aby PRINT, LIST, PLOT, DRAW atd. použily druhou obrazovku. To by také mohla být automaticky ukázána obrazovka, kdyby se přes ni nepřejelo /viz níže/. Kdyby nebyla obrazovka otevřena, ohlásilo by se vadné číslo obrazovky. N musí být 1 - 16.

Každá obrazovka s tím přetváří detaily své vlastní palety, pozici tisku, barvy, mód atd.

DISPLAY n

Mění vystavenou obrazovku, ale ne tu používanou BASICem. Nemůžete ukázat obrazovku zatím co neviditelně budujete další pomocí PRINTu nebo grafických příkazů. Např:

OPEN SCREEN 2: DISPLAY 1: SCREEN 2: PRINT 123

by ukázalo obrazovku 1 při využívání druhé obrazovky na psaní

Neotevřeli se obrazovka, je ohlášeno vadné číslo obrazovky. N musí být 1 - 16 aby ukázalo žádanou obrazovku, nebo DISPAY 0 nebo samotné DISPLAY ukazuje běžnou obrazovku /nastavenou SCREENem n /

OPEN_n

Je-li možno rezervuje N více 16K stránek paměti pro využití BASICem. Umožňuje CLEAR aby nastavil vnější RAMTOP.

CLOSE_n

Uvolňuje N 16K stránek z užití BASIC programem proměnnými, jestliže toto umožňuje RAMTOP. /BASIC tvrdí že to udržuje všechny stránky v chodu včetně té, ve které je umístěn RAMTOP /. Uvolněné stránky se mohou použít soustavou pro zobrazování na display, diskový operační systém a ostatní programy.

STOP - Standartní BASIC

WINDOW vlevo, vpravo, nejvyšší okraj, spodní okraj

Nastavuje hranice textového okénka. První číslo je sloupec nejvíce vlevo pro použití / 0 a více /, druhé je sloupec vpravo /maximum je 84 v módu 3 a znaky široké 6 obrazových prvků/. Třetí číslo udává nejvyšší řádek pro využití a poslední spodní řádek. Zpráva "Vadné okénko" se objeví v tom případě, jestliže žádané hranice nejsou v tomto módu a CSIZE

F U N K C E

INKEY\$,PI,VAL,VAL\$,CODE,CHR\$,SIN,COS,TAN,ASN,ASC,ATN,LN,SQR,
.....<,a BIN jsou pouze standartem

POINT_(x,y) - udává barvu obrazového prvku / 0 - 15 / v módech
3 a 4 a 0 - 1 v módech 1/2

ATTR_(r,c) udává zprávu o chabě v módech 3 a 4

SCREEN\$(r,c)- pracuje v jakémkoli módu a identifikuje UDG v
rozsahu CHR\$ 128 - 168 i normálních prvků. Grafika bloků nejsou
určeny a nedělá žádné rozdíly zda byl určen BLOCKS 0 nebo 1.

RND - dovoluje RND(x) které volí celé číslo mezi 0 a x. RND ne-
bo RND(0) volí číslo v pohyblivé řádové čárce >=0, ale <1.

USR_n - umožňuje hodnoty z 0 - 524287

USR_ "a" /vrací adresu uživatelem definovaného znaku/byla nale-
zena pomocí: UDG "a"

UDG CHR\$ nebo UDG /s CHR\$ 144 napsaným uvnitř to je uživatelem
definovaný grafický znak/ oba udávají adresu modelu znaku pro
CHR\$ 144.

UDG "A" - udá adresu znakového modelu "A" v souboru hlavních
znaků. Byl-li nastaven horní dosah UDG bude např.:UDG CHR\$170
pracovat.

X_MOUSE - xová souřadnice myši

Y_MOUSE - y-nová souřadnice myši

BUTTON_n- Postavení tlačítka myši. Je-li n=1 - 3 na stisknutí
určeného tlačítka odpoví 1, jinak 0./není-li stisknuto/. Je-li
n=0, odpoví 1 jestliže je stisknuto jakékoli nebo všechna tla-
čítka. Např:

IF BUTTON 2 THEN PRINT "tlačítko 2 stisknuto"

X_PEN - Y_PEN , odpovídá souřadnicím x a y světelného pera.

BIN\$ - Např.:BIN\$ 10 ="00001010";BIN\$32768="1000000....."

HEX\$ - Např.:HEX\$ 10="0A";HEX\$32768="8000";HEX\$100000="0186A0"

& Představuje hexadecimální číslo. Např: &FE=254; &8000=32768
VAL("&" + A\$) pracuje &186a0=100000

DPEEK_addr - dvojité PEEK. Ekvivalent k PEEK adresa +256*PEEK(adresa+)

TPEEK_addr - Trojité PEEK. Užitečné pro vedení systémových proměnných, které obsahují adresy v rozsahu 0 - 512K.

FREE - udává volnou paměť vhodnou pro BASIC program a proměnné

RAMTOP - udává nejvyšší adresu použitelnou BASICem /nastaveno CLEAR em n /. Např: CLEAR 70000:PRINT RAMTOP bude psát 70000

SVAR_n - udává adresu systémové proměnné n. Je užitečné jednou mít soupis systémových proměnných! Např: POKE SVAR 1, "\$" nastává kurzor do podoby značky dolaru; PRINT PEEK SVAR 6 snímá rychlost záření pro zářící barvy.

STRING\$(n,a\$) - udává n opakování a\$. Např: STRING\$(3,"ab") je "ababab".

TRUNC\$a\$ - udává a\$ s jakýmkoli sledem vynechaného prostoru. Vhodné při vybírání dat z řetězcových polí. Např: DIM a\$(10,10): LET a\$(1)="velbloud":PRINT a\$(1) píše "velbloud ", ale PRINT TRUNC\$a\$(1) udává "velbloud"

n MOD m - Např: PRINT 110 MOD 60 udává 50/"vlevo přes"/

n DIV m - Např: PRINT 110 DIV 60 udává 1/ekvivalentní s INT N/M/
PRINT 241 DIV 24 udává 10

n BOR m; n BAND m; n BXOR m

Binární OR - AND - XOR /logický součet, log. součin, nonekvivalence/ Např: 16 BOR 8 je 24. Vysvětluje význam, jestliže znáte binární - použijte pomocí BIN\$. Užitečné pro mnoho "technických" věcí.

INST(n,a\$,b\$)

Např: INSTR(x,A\$,B\$) bude hledat B\$ v A\$ z polohy x a vrátí jeho polohu jestliže jej najde, nebo 0 jestliže jej nenajde.

INSTR(A\$,B\$) začíná z polohy 1. Viz BB 3.0 funkcí INSTRING /nebylo možno vynechat posici/. Kontrolní informace může být použita jako universální znak.

ITEM

Typ následující DATA položky. Je-li ITEM = 0, pak není v těchto příkazech DATA více položek. Je-li ITEM = 1 - následující položka je řetězcového typu. Pro ITEM = 2 následující položka je číselového typu. Mezi jinými je ITEM užitečné v procedurách.

LENGTH /d,název_pole/

Volí velikost polí na 1 nebo 2 rozměrové pole a adresy proměnných; Např.: za DIM a\$/4,6/ LENGTH/1,a\$/=4 a LENGTH /2,a\$/=6. LENGTH /0,a\$/ udává adresu prvního bytu v oblasti dat řetězců, nebo polí. LENGTH /0,xyz/ udává adresu prvního bytu čísla xyz. /To je snadné pro vyvolání, aby bylo ohlášeno rozmístění proměnných, tak aby mohly být pozměněny/. Jeli a\$ jednoduchý řetězec, pak LENGTH /1,a\$/ = LEN a\$ a LENGTH/2,a\$/ = 1

V numerických polích jsou potřeba závorky za názvem, např.: LENGTH /1,xyz/ / /

MEM\$/n TO m/

Upravuje část paměti na řetězec, umožňující to, aby byl LETem určen pro pozdější opětovné umístění kdekoli /použitím POKE/. Také dovoluje INSTRu aby vyhledal paměť. N může být počátek. Oba, n a m, musí být zahrnuty. Délka "porcování" paměti musí být méně než 65536.

RESERVED N

Rezervuje a udává adresu N bytů v "hromadě", skladovacím prostoru, který/na rozdíl od prostoru nad RAMTOPem/ je vždy uvnitř stránkovan. Proto je toto více vhodné pro ukládání určitých druhů strojového kódu a systémových proměnných. Definice zvláštních znaků by měly být uloženy sem. Příklad rezervuje místo více než 20-ti definicím, vytváří v něm HUDG bod, systémové proměnné a definuje první z nových znaků.

10 LET místo určení = RESERVED 160

20 DPOKE SVAR 6377místo určení

30 POKE UDG CHR\$ 169,255,129,129,129,129,129,255,0

Toto by mělo být spuštěno pouze jednou; od té doby co rezervuje místo pokaždé a bude postupně zaplňovat "hromadu".

USR\$ _addr

udává výsledek řetězce programu, tvořeného strojovým kódem, na adrese. Strojový kód by měl končit písmeny BC = délka, DE = začátek, A = počáteční stránka / je-li $DE \geq 32768$ /. / Zdokonalený námět /

Měřítková soustava grafiky

XOS, XRG, YOS, YRG - Řídí grafickou stupnici a počátek. XOS a YOS jsou odchylky přičtené k souřadnicím x a y při používání příkazů grafiky. To umožňuje, aby byl počátek přemístěn z normální polohy do polohy právě nad editovací oblastí, nalevo. RUN nebo CLEAR nastavují XOS a YOS na počáteční hodnoty, LET je mění. YRG a XRG mění měřítko grafické soustavy. XRG je běžně 256, nebo 512, jestli že jsou používány "tenké" obrazové prvky. YRG je běžně 192. Změna XRG např: na 1024 by PLOT 950, 100 převedla na legální příkaz, který by dělal $256/1024 * 950$ před vytvořením grafického zobrazení prvků na obrazovce. / Změna rozsahů hodnotou 2 dá rychlejší grafické zobrazování /.

Normálně při používání módů 2, 4 a módu 3 s "tlustými" prvky je souřadnicový systém grafiky s vrcholem 0,0 právě nad spodním okrajem editovací části vlevo a vrchol 255, 173 nahoře vpravo. Je-li použit mód 3 s "tenkými" prvky, je normální x roztaženo tak že vrchol v pravo nahoře má souřadnice 511, 173. V módu je nastavená výška znaku 8, obraz prvku, raději než 9 a editovací oblastí je využito o 2 méně snímání dat, takže vrchol obrazovky y má souřadnici 175.

Na spodní 2 řádky může být nakresleno k použití např:

PLOT 0, -10

Jestliže je poloha počátku přemístěna LETem XOS = -18, pak

0,0 je velice nízko vlevo na obrazovce a 0, 191 je nahoře vlevo.

;TEXT.SAM - chybová hlášení, klíčová slova, MSGS, tabulky atd.
;Výroba zpráv přes E.G.RST 80H : DB 0 pro OK
;UMVAL je vložen do systému UMSGSGS

UMVAL:

```
DB "      MILES GORDON TECHNOLOGY plc      "
DB "      ",127," 1989  SAM Cou","p"+80H      ;0
DB "scroll","?" +80H                          ;1
DC "      "                                  ;2  nepoužíván
DB "Start tape and then press a ke","y"+80H    ;3
DB "Basic:", " " +80H                          ;4
DB "Numeric",ARRAY+80H                          ;5
DB "String",ARRAY+80H                          ;6
DB "Code:", " " +80H                          ;7
DB "S",CREEN,":"," " +80H                      ;8
```

;ERRMVAL je vložen do systému ERRMSGSGS
- ERRMVAL:

```
DB "O","K"+80H                                ;0
DB "Out",SOFS,"memor","y"+80H                  ;1
DB SNOTS,"foun","d"+80H                        ;2
DB "DATA has all been rea","d"+80H             ;3
DB "Subscript wron","g"+80H                    ;4
DB CNXT,WITHOUT,"FO","R"+80H                   ;5
DB "FOR",WITHOUT,CNXT+80H                      ;6
DB "FN",WITHOUT,"DEF F","N"+80H                ;7
DB "RETURN",WITHOUT,"GOSU","B"+80H             ;8
DB MISSING,"LOO","P"+80H                       ;9
DB "LOOP",WITHOUT,"D","O"+80H                  ;10
DB NO,"POP dat","a"+80H                        ;11
DB MISSING,"DEF PRO","C"+80H                   ;12
DB NO,"END PRO","C"+80H                        ;13
DB BREAK,"- CONTINUE to repea","t"+80H        ;14
DB BREAK,"into progra","m"+80H                 ;15
DB "STOP statemen","t"+80H                     ;16
DB "STOP in INPU","T"+80H                      ;17
DB INVALID,"file",SNAME+80H                    ;18
DB "Loading",ERROR+80H                         ;19
DB INVALID,"devic","e"+80H                     ;20
DB INVALID,"s",TREAM," n",UMBER+80H            ;21
DB "End",SOFS,"fil","e"+80H                    ;22
DB INVALID,"colou","r"+80H                     ;23
DB INVALID,PALET,"colou","r"+80H               ;24
DB "Too many ",PALET,"change","s"+80H          ;25
DB "Parameter",ERROR+80H                       ;26
DB INVALID,"argumen","t"+80H                   ;27
DB "N",UMBER,TOO,"larg","e"+80H                ;28
DB "Not understoo","d"+80H                     ;29
DB "Integer out",SOFS,"rang","e"+80H           ;30
DB "Statement doesn't exis","t"+80H            ;31
DB "Off s",CREEN+80H                           ;32
```

```

DB NO,"room for lin","e"+80H ;33
DB INVALID,"s",CREEN," mod","e"+80H ;34
DB INVALID,"BLITZ cod","e"+80H ;35
DB "Stored area",TOO,"bi","g"+80H ;36
DB INVALID,"PUT bloc","k"+80H ;37
DB "PUT mask mismatc","h"+80H ;38
DB MISSING,"END I","F"+80H ;39
DB INVALID,"variable",SNAME+80H ;40
DB "BASIC stack ful","l"+80H ;41
DB "String",TOO,LONG+80H ;42
DB INVALID,"s",CREEN," n",UMBER+80H ;43
DB "S",CREEN,ISALREDOP+80H ;44
DB "S",TREAM,ISALREDOP+80H ;45
DB "Current s",CREEN+80H ;46 byl neplatný kanál
DB "S",TREAM," is",SNOTS,"ope","n"+80H ;47
DB INVALID,"CLEAR adres","s"+80H ;48
DB INVALID,NOTE+80H ;49
DB NOTE,TOO,LONG+80H ;50
DB "FPC",ERROR+80H ;51
DB "Too many definition","s"+80H ;52
DB "No DO","S"+80H ;53
DB INVALID,"WINDO","W"+80H ;54
DB MISSING,"dis","c"+80H ;55

```

í vkládá nad 200 BYTES

```

COMPLIST: DB "Invalid ",0 ;14 uses

INVALID EQU 0

          DB " without ",0 ;5 uses

WITHOUT EQU 1

          DB "Missing ",0 ;5 uses

MISSING EQU 2

          DB " too ",0 ;3 uses

TOO EQU 3

          DB " is already open",0 ;2 uses

ISALREDOP EQU 4

          DB "palette ",0 ;2 uses

PALET EQU 5

          DB " of ",0 ;3 uses

SOFS EQU 6

          DB " error",0 ;4 uses

```

```

ERROR      EQU 7
            DB "tream",0           ;3 uses

TREAM      EQU 8
            DB "umber",0           ;3 uses

UMBER      EQU 9
            DB "NEXT",0            ;2 uses

CNXT       EQU 10
            DB "No ",0             ;3 uses

NO         EQU 11
            DB "BREAK ",0          ;2 uses

BREAK      EQU 12
            DB "creen",0           ;6 USES

CREEN      EQU 13
            DB " array: ",0        ;2 USES

ARRAY      EQU 14
            DB "Note",0            ;2 USES

NOTE       EQU 15
            DB "long",0            ;2 USES

LONG       EQU 16
            DB " not ",0           ;2 USES

SNOTS      EQU 17
            DB " name",0           ;2 USES

SNAME      EQU 18

```

```

;TOKENS SAM - tabulka projevu SAMa
;Řada funkcí 21 - 69H,předchází FF
;Řada ASCII 20 - 7F
;Blok grafiky/Rozšířená nabídka 80 - 8F
;Rozšířená nabídka 90 - A8
;/ 80H - A8H jedná jako UDGS - programováno jako blok grafiky
. následující za rozšířenou nabídkou
;Zapni zdroj UDGS systému obdobný USR "A".Zabírá 328 bajtů z RAM
Přesahuje přes

```

;Povel projeví použití souvislosti k vyvarování se problémům -
PRINT CHR\$ 160 GIVES UDG,LIST
;Povel udělí klíčové slovo.Opakuje platný UDG S.
;Dovoluje 80 - 90 v PROC jménech.Jestliže úvodní symbol může
být povel?
;použití "." pro vnější povel
;určení 96 - 9F
;povelové řádky A0 - FE
;Funkce mají vypnutou předponu H
;"bezprostřední funkce"

Tabulka klíčových slov

KEYWTAB:	DB 0A0H	
IMFNTL:	DC "PI"	;3BH
	DC "RND"	/
	DC "POINT"	/
	DC "FREE"	/
	DC "LENGTH"	/
	DC "ITEM"	/
	DC "ATTR"	/
	DC "FN"	/
	DC "BIN"	/
	DC "XMOUSE"	/
	DC "YMOUSE"	/
	DC "XPEN"	/
	DC "YPEN"	/
	DC "RAMTOP"	/
	DC "-"	/ nepoužitý v řadě
	DC "INSTR"	;4AH
	DC "INKEY\$"	;4BH
	DC "SCREEN\$"	/
	DC "MEM\$"	/
	DC "-"	/ nepoužitý CHAR\$
	DC "PATH\$"	/
	DC "STRING\$"	/
	DC "-"	/ nepoužitý USING\$
	DC "-"	;52H nepoužitý SHIFT\$

;FPC FUNCTIONS /Funkce/

FPCFNTL:	DC "SIN"	;53H
	DC "COS"	/
	DC "TAN"	/
	DC "ASN"	/
	DC "ACS"	/
	DC "ATN"	/
	DC "LN"	/
	DC "EXP"	/
	DC "ABS"	/
	DC "SGN"	/
	DC "SQR"	/
	DC "INT"	/

```

DC "USR"           ;
DC "IN"            ;
DC "PEEK"          ;
DC "DPEEK"         ;
DC "DVAR"          ;
DC "SVAR"          ;
DC "BUTTON"        ;
DC "EOF"           ;
DC "PTR"           ;   disketa použita?
DC "-"            ;   rezervováno
DC "UDG"           ;
DC "-"            ;   číslo
DC "LEN"           ;
DC "CODE"          ;
DC "VAL$"         ;
DC "VAL"          ;
DC "TRUNC$"       ;
DC "CHR$"         ;
DC "STR$"         ;
DC "BIN$"         ;
DC "HEX$"         ;
DC "USR$"         ;
DC "-"            ;   odpovídá INKEY$ FPC CODE
DC "NOT"          ;
DC "-"            ;
DC "-"            ;
DC "-"            ;79H

```

binární operace

```

BINENTL:  DC "MOD"           ;7AH
          DC "DIV"           ;
          DC "BOR"           ;
          DC "BXOR"          ;
          DC "BAND"          ;
          DC "OR"            ;
          DC "AND"           ;
          DC "<>"             ;
          DC "<="            ;
          DC ">="            ;83H

```

kvalifikování - řádek 85H - 8FH

```

KWDS85:  DC "USING"         ;85H
          DC "WRITE"         ;86
          DC "AT"            ;87
          DC "TAB"           ;88
          DC "OFF"           ;89
          DC "WHILE"         ;8A
          DC "UNTIL"         ;8B
          DC "LINE"          ;8C
          DC "THEN"          ;8D
          DC "TO"            ;8E
          DC "STEP"          ;8F

```

i Pověly. Řádek 90H - FEH

	DC "DIR"	;90	
	DC "FORMAT"	;91	
	DC "ERASE"	;92	
	DC "MOVE"	;93	
	DC "SAVE"	;94	
	DC "LOAD"	;95	
	DC "MERGE"	;96	
	DC "VERIFY"	;97	
	DC "OPEN"	;98	
	DC "CLOSE"	;99	
	DC "CIRCLE"	;9A	
	DC "PLOT"	;9B	
	DC "LET"	;9C	
	DC "BLITZ"	;9D	
	DC "BORDER"	;9E	
	DC "CLS"	;9F	
KWDSA0:	DC "PALETTE"	;A0	
	DC "PEN"	;A1	
	DC "PAPER"	;A2	
	DC "FLASH"	;A3	
	DC "BRIGHT"	;A4	
	DC "INVERSE"	;A5	
	DC "OVER"	;A6	
	DC "FATPIX"	;A7	0=slabý, 1=tlustý
	DC "CSIZE"	;A8	
	DC "BLOCKS"	;A9	
	DC "MODE"	;AA	
	DC "GRAB"	;AB	
	DC "PUT"	;AC	
	DC "BEEP"	;AD	
	DC "SOUND"	;AE	
	DC "NEW"	;AF	
	DC "RUN"	;B0	
	DC "STOP"	;B1	
	DC "CONTINUE"	;B2	
	DC "CLEAR"	;B3	
	DC "GO TO"	;B4	
	DC "GO SUB"	;B5	
	DC "RETURN"	;B6	
	DC "REM"	;B7	
	DC "READ"	;B8	
	DC "DATA"	;B9	
	DC "RESTORE"	;BA	
	DC "PRINT"	;BB	
	DC "INPUT"	;BC	

	DC "LIST"	;BD
	DC "LLIST"	;BE
	DC "DUMP"	;BF
KWDSC0:	DC "FOR"	;C0
	DC "NEXT"	;C1
	DC "PAUSE"	;C2
	DC "DRAW"	;C3
	DC "DEFAULT"	;C4
	DC "DIM"	;C5
	DC "INPUT"	;C6
	DC "RANDOMIZE"	;C7
	DC "DEF FN"	;C8
	DC "DEF KEYCODE"	;C9
	DC "DEF PROC"	;CA
	DC "END PROC"	;CB
	DC "RENUM"	;CC
	DC "DELETE"	;CD
	DC "REF"	;CE
	DC "COPY"	;CF
	DC "-"	;D0 EDIT
	DC "KEYIN"	;D1
	DC "LOCAL"	;D2
	DC "LOOP IF"	;D3
	DC "DO"	;D4
	DC "LOOP"	;D5
	DC "EXIT IF"	;D6
	DC "IF"	;D7
	DC "IF"	;D8
	DC "ELSE"	;D9
	DC "ELSE"	;DA
	DC "END IF"	;DB
	DC "KEY"	;DC
	DC "ON ERROR"	;DD
	DC "ON"	;DE
	DC "GET"	;DF
KWDSE0:	DC "OUT"	;E0
	DC "POKE"	;E1
	DC "DPOKE"	;E2
	DC "RENAME"	;E3
	DC "CALL"	;E4
	DC "ROLL"	;E5
	DC "SCROLL"	;E6
	DC "SCREEN"	;E7
	DC "DISPLAY"	;E8
BTWD:	DC "BOOT"	;E9 použití BOOTu k ověření jméno řádku
	DC "LABEL"	;EA
	DC "FILL"	;EB
	DC "WINDOW"	;EC
	DC "AUTO"	;ED
	DC "POP"	;EE
	DC "RECORD"	;EF
	DC "DEVICE"	;F0

```

DC "PROTECT"          ;F1
DC "HIDE"              ;F2
DC "ZAP"               ;F3
DC "POW"               ;F4
DC "BOOM"              ;F5
DC "ZOOM"              ;F6
DC "-"                 ;F7
DC "-"                 ;F8
DC "-"                 ;F9
DC "-"                 ;FA
DC "-"                 ;FB
DC "-"                 ;FC
DC "-"                 ;FD
DC "-"                 ;FE
DC "INK"               ;FF ( funkční předpona a
                       ;TEMP "INK" kanál

```

/ definice klíče počátku tabulky

```

DKSRC: DB 192,1,0,0BDH          ;F0 LIST
      DB 193,2,0,0CCH,":"      ;F1 RENUM :
      DB 194,2,0,0BBH,":"      ;F2 PRINT :
      DB 195,2,0,0AAH,":"      ;F3 MODE :
      DB 196,1,0,0B0H          ;F4 RUN
      DB 197,1,0,0B2H          ;F5 CONTINUE
      DB 198,2,0,09FH,"#"      ;F6 CLS #
      DB 199,3,0,LOADTOK,22H,22H ;F7 LOAD ""
      DB 200,5,0,LOADTOK,22H,22H,0FFH,CODETOK ;F8 LOAD "" CODE
      DB 201,1,0,0E9H          ;F9 BOOT

      DB 252,2,0,6,":"         ;TAB
      DB 253,3,0,20,1,":"      ;INVERSE 1 CC
      DB 254,3,0,20,0,":"      ;INVERSE 0 CC
DKEN:  DB 0FFH

```

/ počátek tabulky - 18 BYTES

```

CHIT:  DB ">"                ; kurzor běžného řádku
      DB 128,129              ; znaky kurzoru - nižší případ/vyšší případ
      DB "1","0"              ; digitální posluha pro "BIN$ "
      DB "#"                  ; INSTR WILD CARD CHAR
      DB "T",TSPEED           ; DEVICE T, rychlost pásku
      DB 17                   ; rychlost zablesknutí barvy
      DW LINICOLS              ; LINIPTR
      DB 0FFH,0,0             ; XCMDP SHOWS "NO EXTERNAL CMDS"
      DB 79                   ; PRINTER RHS
      DB 10                   ; AFTER CR CHARACTER
      DB 0E9H                 ; port řídicí tiskárny / E8=DATA /
      DB 1                    ; BYTE tiskárny STROBE

MAIT:  DW BSTACK              ; BASSTK
      DW HPEND                ; HEAPEND
      DW HPEND                ; HPST počátek prázdné hromady

```



```
DW FPSB      ;FPSBOT
DW DKBU      ;počátek buferu definované klávesy
DW DKBU+128  ;hranice definované klávesy buferu
DW ENDOUTP   ;schopnost tisku O/P rutina
DW ERRMVAL   ;chybové hlášení
DW UMVAL     ;"UTILITY" hlášení
DW KTAB      ;seznam klávesnice
DW CMDADT    ;počátek seznamu CMD ADDR
```

;První 4 BYTES v CHANTAB vždy použity

```
CHANTAB:  DW PRMAIN
           DW KYIP
           DB "K"

           DW PRMAIN
           DW KYIP
           DB "S"

           DW ADDCHAR
           DW IDERR
           DB "R"

           DW PRMAIN
           DW IDERR
           DB "P"

           DW S16OP
           DW IDERR
           DB "$"

           DW SENDA
           DW IDERR
           DB "B"
```

;mapa 70 klíčů/3 nejsou čteny touto cestou/ v nezměněné poloze,CAPS SHIFT a SYMBOL SHIFT.Každá řada odpovídá bitu na portu /bit 7 nejvyšší/ a každý sloupec portu MSB.CAPS SHIFT /bit 0,port FE/ použit odděleně,což je SYMBOL SHIFT /bit 1,port F7/ a CONTROL /bit 0,port FF/

```
;PORT MSB:      FF  7F  BF  DF  EF  F7  FB  FD  FE
;BIT MAP BYTE   8   7   6   5   4   3   2   1   0
```

```
;254 definován jako INVERSE CC,0
;253 definován jako INVERSE CC,1
;252 definován jako CHR$ 6
```

```
KSRC:  DB      253,07H,0C0H,12,6H,0C9H,198,195 ;BIT 7
        DB "X",".",":","","+",252,200,197,194 ; 6
        DB "X"," ","",";","=","-"," ",199,196,193 ; 5

        DB 09H,"b","h","y","6","5","t","g","v" ; 4
```

```

DB 08H,"n","j","u","7","4","r","f","c" ; 3
DB 0AH,"m","k","i","8","3","e","d","x" ; 2
DB 0BH,"x","l","o","9","2","w","s","z" ; 1
DB "X"," ",0DH,"p","0","1","q","a" ; 0

```

; použití CAPS SHIFT

```

DB "Ç",07H,202,0EH,06H,211,208,205
DB "X",".",":",127,"*",252,210,207,204
DB "X","",",","_","/", " ",209,206,203

```

```

DB 09H,"B","H","Y","&","&","T","G","V"
DB 08H,"N","J","U","'", "$","R","F","C"
DB 0AH,"M","K","I","(", "#","E","D","X"
DB 0BH,"X","L","O",")", "à","W","S","Z"
DB "X"," ",0DH,"P",126,"I","Q","A"

```

; použití SYMBOL SHIFT

```

DB 254,0FH,"0",0EH,06H,"9","6","3"
DB "X",">","*",0CFH,"*",252,"8","5","2"
DB "X","<","+","_","/", " ", "7","4","1"

```

```

DB 09H,9EH,5EH,157,134,133,"$","è",VERIFYTOK
DB 08H,164,"-",129,135,132,"`","é",168
DB 0AH,165,"+",0A5H,128,131,130,9CH,"?"
DB 0BH,"X",60H,148,124,130,">","SAVETOK","?"
DB "X",255,0DH,187,126,129,"<","0B1H

```

; řídicí klávesa seznamu - DISP/VALUE

```

CKTAB: DB 30,137 ;grafika
        DB 31,138
        DB 39,136
        DB 40,139
        DB 48,143
        DB 49,140
        DB 58,141
        DB 67,142

        DB 27,19 ;jas
        DB 47,16 ;inkoust
        DB 65,17 ;papír
        DB 0

```

;počáteční barvy palety - MS GRB,bit jasu,LS GRB/zelená,červená,
;modrá/ bity GRBbGRB

```

počáteční nabídka DB 0000000B ; černá
                   DB 0010000B ; modrá
                   DB 0100000B ; červená
                   DB 0110000B ; oranžová
                   DB 1000000B ; zelená
                   DB 1010000B ; šedá

```

```

DB 1100000B      ; žlutá
DB 1111000B      ; bílá

DB 0000000B      ;BRIGHT BLACK=černá
DB 0010001B      ;světle modrá
DB 0100010B      ;světle červená
DB 0110011B      ;světle růžová
DB 1000100B      ;světle zelená
DB 1010101B      ;světle šedá
DB 1100110B      ;světle žlutá
DB 1111111B      ;      bílá

```

;Barvy pro čtyřbarevný mod

```

DB 0000000B      ;černá
DB 0010001B      ;modrá
DB 0100010B      ;červená
DB 1111111B      ;bílá

```

;Seznam adres povelů.Řádek 90H - FEH

```

CMDADT:  DW DIR          ;DIR      90
          DW NONSENSE     ;FORMAT   91
          DW NONSENSE     ;ERASE   92
          DW NONSENSE     ;MOVE    93
          DW SLMVC        ;SAVE    94
          DW SLMVC        ;LOAD    95
          DW SLMVC        ;MERGE    96
          DW SLMVC        ;VERIFY   97
          DW OPSCRN       ;OPEN    98
          DW CLSCRN       ;CLOSE   99
          DW CIRCLE       ;CIRCLE  9A
          DW PLOT         ;PLOT    9B
          DW LET          ;LET     9C
          DW BLITZ        ;BLITZ   9D
          DW BORDER       ;BORDER  9E
          DW CLS          ;CLS     9F

          DW COLOUR       ;PALETTE A0
          DW PERMS        ;INK     A1
          DW PERMS        ;PAPER   A2
          DW PERMS        ;FLASH   A3
          DW PERMS        ;BRIGHT  A4
          DW PERMS        ;INVERSE  A5
          DW PERMS        ;OVER     A6
          DW FATPIX       ;FATPIX  A7   šířka bodu
          DW WIDTH        ;CSIZE   A8
          DW BGRAPHICS    ;BLOCKS  A9   blok grafiky/UDGS
          DW MODECMD      ;MODE    AA
          DW GRAB         ;GRAB    AB
          DW PUT          ;PUT     AC
          DW BEEP         ;BEEP    AD
          DW SOUND        ;SOUND   AE
          DW NEW          ;NEW     AF

```

DW RUN	;RUN	B0
DW STOP	;STOP	B1
DW CONTINUE	;CONTINUE	B2
DW CLEAR	;CLEAR	B3
DW GOTO	;GO TO	B4
DW GOSUB	;GO SUB	B5
DW RETURN	;RETURN	B6
DW REMARK	;REM	B7
DW READ	;READ	B8
DW DATA	;DATA	B9
DW RESTORE	;RESTORE	BA
DW PRINT	;PRINT	BB
DW LPRINT	;LPRINT	BC
DW LIST	;LIST	BD
DW LLIST	;LLIST	BE
DW COPY	;DUMP	BF
DW FOR	;FOR	C0
DW NEXT	;NEXT	C1
DW PAUSE	;PAUSE	C2
DW DRAW	;DRAW	C3
DW DEFAULT	;DEFAULT	C4
DW DIM	;DIM	C5
DW INPUT	;INPUT	C6
DW RANDOMIZE	;RAND	C7
DW DEFFN	;DEF FN	C8
DW DEFKEY	;DEF KEYCODE	C9
DW DEFPROC	;DEF PROC	CA
DW ENDPROC	;END PROC	CB
DW RENUM	;RENUM	CC
DW DELETE	;DELETE	CD
DW NONSENSE	;REF	CE
DW NONSENSE	;COPY	CF
DW NONSENSE	;EDIT	D0
DW KEYIN	;KEYIN	D1
DW LOCAL	;LOCAL	D2
DW LOOPIF	;LOOP IF	D3
DW DO	;DO	D4
DW LOOP	;LOOP	D5
DW EXITIF	;EXIT IF	D6
DW LIF	;LONG IF	D7
DW SIF	;SHORT IF	D8
DW LELSE	;LELSE	D9
DW ELSE	;ELSE	DA
DW ENDIF	;END IF	DB
DW KEY	;KEY	DC
DW ONERROR	;ON ERROR	DD
DW ON	;ON	DE
DW GET	;GET	DF
DW OUT	;OUT	E0
DW POKE	;POKE	E1
DW DPOKE	;DPOKE	E2
DW NONSENSE	;RENAME	E3

DW CALLER	;CALL	E4
DW ROLL	;ROLL	E5
DW SCROLL	;SCROLL	E6
DW SCREEN	;SCREEN	E7
DW DISPLAY	;DISPLAY	E8
DW BOOT	;BOOT	E9
DW LABEL	;LABEL	EA
DW FILL	;FILL	EB
DW WINDOW	;WINDOW	EC
DW AUTO	;AUTO	ED
DW POP	;POP	EE
DW RECORD	;RECORD	EF
DW SLDEVICE	;DEVICE	F0
DW NONSENSE	;PROTECT	F1
DW NONSENSE	;HIDE	F2
DW ZAP	;ZAP	F3
DW POW	;POW	F4
DW BOOM	;BOOM	F5
DW ZOOM	;ZOOM	F6
DW NONSENSE	;	F7
DW NONSENSE	;	F8
DW NONSENSE	;	F9
DW NONSENSE	;	FA
DW NONSENSE	;	FB
DW NONSENSE	;	FC
DW NONSENSE	;	FD
DW NONSENSE	;	FE
DW NONSENSE	;	FF (FUNCTION PREFIX)

;seznam znaků,užití všech osmi uvažovaných a hodnota užití SCAN 8

```

U8TAB:  DB ",",08H
        DB ";",08H
        DB "_",0FFH

        DB "g",1CH
        DB "j",10H
        DB "p",20H
        DB "q",02H
        DB "y",1CH
        DB 0

```

;Použití znaků Simonem N.GOODWINEM
;Znaky užívají průměrně 6 bodů,mohou být nejvýše 8 bodů,zhuštěná
;forma používá 7,5-ti bitové plátky po znaku,zabírá 600 bajtů VS
1096

CHARSRC:

```

DB 000H,000H,000H,000H,004H
DB 021H,008H,040H,011H,04AH
DB000H,000H,000H,029H,05FH

```

DB 057H,0D4H,0A7H,052H,08EH
DB 029H,05DH,09CH,088H,088H
DB 09CH,0D1H,04AH,022H,0B2H
DB 069H,084H,040H,000H,000H
DB 008H,088H,042H,008H,030H
DB 041H,008H,044H,040H,095H
DB 077H,0DDH,052H,000H,084H
DB 0F9H,008H,000H,000H,000H
DB 031H,084H,000H,002H,0E0H
DB 000H,000H,000H,000H,006H
DB 030H,042H,022H,022H,010H
DB 074H,067H,05CH,0C5H,0C4H
DB 061H,008H,042H,039H,0D1H
DB 008H,088H,08FH,0BAH,021H
DB 030H,062H,0E1H,019H,052H
DB 0F8H,085H,0F8H,078H,021H
DB 08BH,08CH,088H,07AH,031H
DB 077H,0C2H,022H,022H,010H
DB 074H,062H,0E8H,0C5H,0CEH
DB 08CH,05EH,011H,030H,000H
DB 031H,080H,063H,000H,006H
DB 030H,00CH,020H,088H,088H
DB 020H,082H,000H,07CH,01FH
DB 000H,020H,082H,008H,088H
DB 083H,0A2H,011H,010H,004H
DB 074H,067H,05BH,0C1H,0CEH
DB 08CH,07FH,018H,0C7H,0D1H
DB 08FH,0A3H,01FH,03AH,030H
DB 084H,022H,0EFH,046H,031H
DB 08CH,07DH,0F8H,043H,0D0H
DB 087H,0FFH,008H,07AH,010H
DB 083H,0A3H,008H,04EH,02EH
DB 08CH,063H,0F8H,0C6H,02EH
DB 021H,008H,042H,038H,021H
DB 008H,043H,017H,046H,054H
DB 0C5H,025H,018H,042H,010H
DB 084H,03FH,01DH,0D6H,031H
DB 08CH,063H,01CH,0D6H,071H
DB 08BH,0A3H,018H,0C6H,02EH
DB 0F4H,063H,0E8H,042H,00EH
DB 08CH,063H,059H,037H,0D1H
DB 08FH,0A9H,028H,0BAH,030H
DB 070H,062H,0EFH,090H,084H
DB 021H,009H,018H,0C6H,031H
DB 08BH,0A3H,018H,0C6H,02AH
DB 024H,063H,018H,0D6H,0AAH
DB 08CH,054H,045H,046H,031H
DB 08AH,088H,042H,013H,0E1H
DB 011H,011H,00FH,09CH,084H
DB 021H,008H,078H,041H,004H
DB 010H,043H,0C2H,010H,084H
DB 027H,008H,0FAH,090H,084H
DB 020H,000H,000H,000H,000H
DB 0E0H,051H,0C4H,023H,0E0H
DB 003H,082H,0F8H,0BEH,010H

DB 0F4H,063H,01FH,000H,00FH
DB 084H,020H,0F0H,085H,0F1H
DB 08CH,05EH,000H,03AH,03FH
DB 083H,0CCH,094H,079H,008H
DB 040H,000H,0F8H,0C5H,0E1H
DB 084H,03DH,018H,0C6H,024H
DB 001H,008H,042H,008H,080H
DB 021H,008H,042H,042H,011H
DB 097H,025H,012H,010H,084H
DB 021H,00CH,000H,06AH,0B5H
DB 0ADH,040H,00FH,046H,031H
DB 088H,000H,0E8H,0C6H,02EH
DB 000H,03DH,018H,0FAH,000H
DB 003H,0E3H,017H,084H,000H
DB 0B6H,061H,008H,000H,00FH
DB 083H,083H,0E2H,011H,0E4H
DB 021H,006H,000H,046H,031H
DB 08BH,0C0H,008H,0C6H,02AH
DB 020H,001H,018H,0C6H,0AAH
DB 000H,022H,0A2H,02AH,020H
DB 004H,063H,017H,084H,000H
DB 0F8H,088H,08FH,088H,084H
DB 041H,008H,022H,010H,084H
DB 021H,008H,082H,010H,044H
DB 022H,00AH,0A0H,000H,000H
DB 007H,082H,0D4H,0B4H,03EH
DB 000H,000H,00FH,07BH,0C0H
DB 0F7H,0BCH,000H,000H,0C8H
DB 074H,07FH,007H,080H

DB "AW,BG"

SAM Coupe System Variables

;VARS.SAM

VAR2 EQU 5A00H musí začínat na krajní stránce

;dalších 44 bytů začíná společně s "CHIT"

LNCUR	EQU VAR2+00H	; kurzor běžného řádku - obvykle '>')
KURCHAR	EQU VAR2+01H	; (2) znaky kurzoru - nižší případ/ vyšší případ
BIN1DIG	EQU VAR2+03H	; použití BIN\$ - obvykle " 1 "
BINODIG	EQU VAR2+04H	; obvykle " 0 "
INSTHASH	EQU VAR2+05H	; normálně " E "
PSLD	EQU VAR2+06H	; (2) heslo písmeno číslo ; číslo=rychlost pásku/disketa číslo/ prázdná pozice
SPEEDINK	EQU VAR2+08H	; barva textu se zableskne obráceně
LINIPTR	EQU VAR2+09H	; (2) PTR k přerušení řádku, partner CHNG
XCMDP	EQU VAR2+0BH	; (3) strana/adresa z prvního vnějšího CMD nebo FFxxxx
PRRHS	EQU VAR2+0EH	; hranice tisku RHS - 79
AFTERCR	EQU VAR2+0FH	; OA nebo nul oodle toho, který AUTO LF je připraven
LPTPRT1	EQU VAR2+10H	; (2) port řídící tisk/01H STROBE hodnota
TABVAR	EQU VAR2+2FH	; SPARE
M23LSC	EQU VAR2+30H	; 0 jestliže TAB=16 jinak TAB=8
SOFE	EQU VAR2+32H	; (2) M2/3 spodní obrazovka barev ; plocha obrazovky vypnuta obnovena schc nost/zbavena schopnosti 0=zapnuta
TPROMPTS	EQU VAR2+33H	; BIT 0=1zastavuje tisk jmen během zatí ; BIT 1=1zastavuje PROMPTS během vkládár
BGFLG	EQU VAR2+34H	; plocha grafického bloku
FL6OR8	EQU VAR2+35H	; 00=6bitů CHARS v modu 2, NZ=8 bitů
CSIZE	EQU VAR2+36H	; (2) výška/šířka
UWRHS	EQU VAR2+38H	; STARTS AT 31
UWLHS	EQU VAR2+39H	; STARTS AT 0
UWTOP	EQU VAR2+3AH	; STARTS AT 0
UWBOT	EQU VAR2+3BH	; STARTS AT 18 (19 LINES IN UPPER, 2 IN LOWER SCR, 9 PIX)

spodní okno .

LWRHS EQU VAR2+3CH
LWLHS EQU VAR2+3DH
LWTOP EQU VAR2+3EH
LWBOT EQU VAR2+3FH

; začíná na 20

MODE EQU VAR2+40H
YCOORD EQU VAR2+41H
XCOORD EQU VAR2+42H
RLINE EQU XCOORD

; 0-191, 0 až vrchol
; (2) 0-255 (FAT) OR 0-511 (THIN)

; stálá grafika/tisk VARS

THFATP EQU VAR2+44H
ATTRP EQU VAR2+45H
MASKP EQU VAR2+46H
PFLAGP EQU VAR2+47H
M23PAPP EQU VAR2+48H
M23INKP EQU VAR2+49H

; 00= tenké, NZ=široké body
; ATTR užito v modu 0 a 1
; BIT 4=PAPER 9, BIT 6=INK 9
; ubírá nebo zdvojuje hodící se bity
; 00-FF. ubírá hodící se, ledaže si
; přejete tečkování řádku
; MUST BE NEXT TO OVER FOR LD
DE, (MINKT)
; 0-1. NORMAL OR OVER 1 (OR-ING)
; 00/FF FOR NORMAL/INVERSE ;
; 0-3. NORMAL, XOR, OR, AND ; THESE 2
LINKED
; (GRAPHICS OVER - USED BY PUT)

OVERP EQU VAR2+4AH
INVERP EQU VAR2+4BH
GOVERP EQU VAR2+4CH

; Dočasná grafika / tisk VARS

THFAIT EQU VAR2+4DH
ATTRT EQU VAR2+4EH
MASKT EQU VAR2+4FH
PFLAGT EQU VAR2+50H
M23PAPT EQU VAR2+51H
M23INKT EQU VAR2+52H

; opisuje/kopíruje z THFATP když mod=2,
jinak použit NZ/tlusté/

OVERT EQU VAR2+53H
INVERT EQU VAR2+54H
GOVERT EQU VAR2+55H

; BIT 4=PAPER 9, BIT 6=INK 9
; ubírá, nebo zdvojuje hodící se bity
; 00-FF. Ubírá hodící se ledaže si přejete
tečkování řádku
; 0-1. NORMAL OR OVER 1 (OR-ING)
; 00/FF pro normál/obrácený
; 0-3. NORMAL, XOR, OR, AND ; řádkování

; Běžné okno

WINDRHS EQU VAR2+56H
WINDLHS EQU VAR2+57H
WINDTOP EQU VAR2+58H
WINDBOT EQU VAR2+59H

KBQB EQU VAR2+5AH
KBQP EQU VAR2+6AH

SPOSNU EQU VAR2+60H

; (16) tvoření klávesnice
; (2) ukazatelé tvoření klávesnice.
dolů=konec, nahoru=hlavička
; (2) posice na obrazovce, WINDTOP po CLS

SPOSNL	EQU VAR2+6EH	;(2) posice na obrazovce /spodní/ 0,19 CLS
PRFOSN	EQU VAR2+70H	;(2) ukazatel posice/1mimořádný bajt pr vhodný LD/
OPCHAR	EQU VAR2+72H	;použití LPRINTu - běžně 0/P CHAR
DEVICE	EQU VAR2+73H	;0=US, 1=LS, 2=PRINTER, 3=
CLET	EQU VAR2+74H	;CURRENT CHANNEL LETTER K/S/P/B/T/S ETC
IFTYPE	EQU VAR2+75H	; dlouhý/krátký
REFFLG	EQU VAR2+76H	; Z jestliže REF VAR pracuje
CURDISP	EQU VAR2+77H	; běžný display
CUSCRNP	EQU VAR2+78H	; běžná stránka obrazovky
CURP	EQU VAR2+79H	; běžný horní port RAM
CLRP	EQU VAR2+7AH	; běžný dolní port RAM/skládá během stránkování/
CSA	EQU VAR2+7BH	;(2) běžný údaj adresy
FIRST	EQU VAR2+7DH	;(2)
LAST	EQU VAR2+7FH	;(2) čísla řádku např.LIST x TO y.Take se používá k sestavení plátek ;ARRAY SLICER

;ukazují, že jsou připraveny k vytvoření prostoru, opětnému dosažení:

SAVARSP	EQU VAR2+81H	;(2) ;SAVARSP/NUMEND/NVARS musí být v pořadí
SAVARSP	EQU VAR2+82H	
NUMENDP	EQU VAR2+84H	;NUMEND/NVARS/DATADD musí být v pořadí
NUMEND	EQU VAR2+85H	;(2)
NVARSP	EQU VAR2+87H	;(2)
NVARSP	EQU VAR2+88H	
DATADDP	EQU VAR2+8AH	;(2)
DATADD	EQU VAR2+8BH	
WKENDP	EQU VAR2+8DH	;(2)
WKEND	EQU VAR2+8EH	
WORKSPP	EQU VAR2+90H	;(2)
WORKSP	EQU VAR2+91H	
ELINEP	EQU VAR2+93H	;(2)
ELINE	EQU VAR2+94H	
CHADP	EQU VAR2+96H	;(2)
CHAD	EQU VAR2+97H	
KCURP	EQU VAR2+99H	;(2)
KCUR	EQU VAR2+9AH	
NXTLINEP	EQU VAR2+9CH	;(2)
NXTLINE	EQU VAR2+9DH	

```

PRCGP      EQU VAR2+09FH
PRCG       EQU VAR2+0A0H      ; (2)

XPTRP      EQU VAR2+0A2H
XPTR       EQU VAR2+0A3H      ; (2)

DESTP      EQU VAR2+0A5H
DEST       EQU VAR2+0A6H      ; (2)

PRPTRP     EQU VAR2+0A8H
PRPTR      EQU VAR2+0A9H      ; (2) PROC ukazatel

```

; konec upravy PTRS

```

DPPTRP     EQU VAR2+0ABH
DPPTR      EQU VAR2+0ACH      ; (2) DEF PROC ukazatel

```

```

CLAPG      EQU VAR2+0AEH
CLA        EQU VAR2+0AFH      ; (2)

```

```

WINDMAX    EQU VAR2+0B1H      ; (2) okono nejspodnější části/max RHS
LDCO       EQU VAR2+0B3H      ; LD ZX CODE konec postavení stránky
CLOF       EQU VAR2+0B4H      ; postavení úplně čisté stránky
OPSTORE    EQU VAR2+0B5H      ; (2)
DMPFG      EQU VAR2+0B7H      ; jestliže NZ tisk O/P je oddložen
LISTFLG    EQU VAR2+0B8H      ; 0/1/2 pro seznam formátu 0/1/2
LSTFT      EQU VAR2+0B9H      ; dočasná verze LISTFLG použita na
                                panelu "R"
INQUFG     EQU VAR2+0BAH      ; v udané ploše.BIT 0=1 jestliže je v
                                ploše.Obrys nulový.BIT v takovémto počá
                                tečním stavu je "vnější strana" a zname
                                tisk mimo plochu.
                                tisk používá pro 1 takto vždy UDG

```

```

SPROMPT    EQU VAR2+0BBH      ; jestliže NZ, potom neprobíhá "SCROLL?"
;SPARE
INDOPFG    EQU VAR2+0BDH      ; vyhotovuje O/P plachu

```

```

NXTSPCS    EQU VAR2+0BEH
CURSPCS    EQU VAR2+0BFH
NXTTHSPCS  EQU VAR2+0C0H
CURTHSPCS  EQU VAR2+0C1H

```

```

KPOS       EQU VAR2+0C2H      ; (2) kurzor pouze na obrazovce

```

; Další 4 musí zůstat v pořadí

```

SOFFCT     EQU VAR2+0C4H      ; okénko pro vypnutí obrazovky
SOFLG      EQU VAR2+0C5H      ; plocha pro obrazovku má být obrácena
                                mimo NZ nebo na Z
SPEEDIC    EQU VAR2+0C6H      ; okénko pro objevení se INKS
PALFLAG    EQU VAR2+0C7H      ; BIT 0 ukazuje, který druh seznamu je
                                použit
TEMPW1     EQU VAR2+0C8H      ; (2)
TEMPW2     EQU VAR2+0CAH      ; (2)

```


ERRSTAT	EQU VAR2+0143H	
ERRLN	EQU VAR2+0143H	; 2, poukáže na chybný řádek
ONERRFLG	EQU VAR2+0145H	; BIT 7=TEMP ON, BIT 0=PERM ON
ONSTORE	EQU VAR2+0146H	; nedá tprávu na CMD
BCSTORE	EQU VAR2+0147H	; (2) použití RST 30H
M3PAPP	EQU VAR2+0149H	; (2)
M3LSC	EQU VAR2+0143H	; (2)
TEMPW4	EQU VAR2+014DH	; (2) USED BY POINTERS - použití ukazatel
TEMPW5	EQU VAR2+014FH	; (2) USED BY POINTERS - použití ukazatel
		; 1 SPARE - 1 mezera
LPT	EQU VAR2+0152H	; (30)
ORGOFF	EQU VAR2+0170H	; GRAPHICS ORIGIN OFFSET - počátek grafiky
LSOFF	EQU VAR2+0171H	; počáteční bit na spodu obrazovky
RNSTKE	EQU VAR2+0172H	; (2) hranice přepisování
CURCMD	EQU VAR2+0174H	; kód CMD se provádí
LTDFF	EQU VAR2+0175H	; způsobuje/opomíjí plochu
STRM16NM	EQU VAR2+0176H	; (11) TL bajt/jmeno VARu, zapisuje 16 proudů v paměti
GRARF	EQU VAR2+0181H	; grafika uchová plochu - 0 vypnuto
DHADJ	EQU VAR2+0182H	; dvojitá výška.0 jestli že je spodek DH CHAR O/PED
PAGCOUNT	EQU VAR2+0183H	; použití obrácené stránky FARLDIR
MODCOUNT	EQU VAR2+0184H	; (2) mód 16K obrácené použití
BCREG	EQU VAR2+0186H	; (2) BC registr FPC
AUTOFLG	EQU VAR2+0188H	
AUTOSTEP	EQU VAR2+0189H	; (2)
RSTEP	EQU AUTOSTEP	
		; RENUM STEP
LSPTR	EQU VAR2+018BH	; (2) prohlížení řádku PTR
LNPTR	EQU VAR2+018DH	; (1)
MSEDP	EQU VAR2+018EH	; (8) 018E-0195
BUTSTAT	EQU VAR2+018FH	; postavení knoflíku myši
MXCRD	EQU VAR2+0196H	; (2) MOUSE X koordinát
MYCRD	EQU VAR2+0198H	; (2) MOUSE Y koordinát
		; použití tisku FP
FRACLIM	EQU VAR2+019AH	
NPRPOS	EQU VAR2+019BH	; (2)
DIGITS	EQU VAR2+019DH	; musí
EPOWER	EQU VAR2+019EH	; všechno
DECPNTED	EQU VAR2+019FH	; zůstat v tomto pořadí
PRNBUFF	EQU VAR2+01A0H	; (16) ALLOWS -0.0000123456789
		; OR -1.2345678E-35
BCDBUFF	EQU VAR2+01B0H	; (5)
OTHER	EQU VAR2+01B5H	; určení čísla postavení
DCT	EQU VAR2+01B6H	; chybné okénko na disketě
SIPW	EQU VAR2+01B7H	; (2) heslo písmeno/číslice
		; číslo=rychlost pásku/disketa číslo

```

postavení
;1 SPARE 1 mezera
;(2)
;(2) použití LOOKVARS
;(2)
;(2) skok po provedení DOS
;Z jestli že DOS není vložen
;BIT 0 je použit jestli že je DOS řídící
;(2)

;dalších 26 bajtů začíná spolu s MAIT

BASSTK EQU VAR2+01C5H ;(2)
HEAPEND EQU VAR2+01C8H ;(2)
HPST EQU VAR2+01CAH ;(2)
FPSBOT EQU VAR2+01CCH ;(2) začátek FPCS
DKDEF EQU VAR2+01CEH ;(2) klávesa DEF
DKLIM EQU VAR2+01D0H ;(2) hranice klávesy DEF buferu
PATOUT EQU VAR2+01D2H ;(2) adresa "tisku seznamu CHARS" O/P
ERRMSG EQU VAR2+01D4H ;(2)
UMSGS EQU VAR2+01D6H ;(2)
KBTAB EQU VAR2+01D8H ;(2)
CMDADDR EQU VAR2+01DAH ;(2) začátek CMD adresa seznamu v ROMO
MNOP EQU VAR2+01DCH ;(2)
MNIP EQU VAR2+01DEH ;(2)

;mezera

KBUFF EQU VAR2+01EEH ;(18) - 2seznam 72 bitů

; mezera
; uhodit předešlou klávesu

LASTH EQU 5C01H
KDATA EQU 5C02H
LKPB EQU 5C03H ;(2)
REPCT EQU 5C05H
LASTKV EQU 5C06H ;(2)
LASTK EQU 5C08H ;klíč z buferu postavený na vrchol
REPDEL EQU 5C09H
REPPER EQU 5C0AH

; mezera
;(42 - užívá 5C0C - 5C35 pro streams
-5 TO 15. -4=16

STREAMS EQU 5C10H

CHARS EQU 5C36H
RASP EQU 5C38H
PIP EQU 5C39H
ERRNR EQU 5C3AH
FLAGS EQU 5C3BH
TVFLAG EQU 5C3CH
ERRSP EQU 5C3DH
LISTSP EQU 5C3FH

NEWPPC EQU 5C42H ;(2)
NSPPC EQU 5C44H
PPC EQU 5C46H ;(2)

```

SUBP	EQU 5C47H	
BORDCOL	EQU 5C48H	
EP	EQU 5C49H	;(2)
BORDCOL	EQU 5C4BH	
CHANS	EQU 5C4FH	;(2)
CURCHL	EQU 5C51H	;(2)
DEFADDP	EQU 5C53H	
DEFADD	EQU 5C54H	;(2)
STKEND	EQU 5C55H	
KPFLG	EQU 5C57H	; funkční klávesy pokud sudý, číslo bloku pokud lichý
MEM	EQU 5C68H	
FLAGS2	EQU 5C6AH	
SDTOP	EQU 5C6CH	
OLDPPC	EQU 5C6EH	
OSP	EQU 5C70H	
FLAGX	EQU 5C71H	
STRLEN	EQU 5C72H	
SEED	EQU 5C76H	
FRAMES	EQU 5C78H	;(3)
UDG	EQU 5C7BH	;(2)
HUDG	EQU 5C7DH	;(2)
FRAMES34	EQU 5C7FH	;(2)
OLDPOS	EQU 5C82H	
SCRCT	EQU 5C8CH	
SCPTR	EQU 5C9DH	; mezera ;(2) adresa běžné obrazovky ve SCLIST
		; NEW nevyčistí
FISCRNP	EQU 5C9FH	
SCLIST	EQU 5CA0H	;(16) obruba obrazovek
LASTPAGE	EQU 5CB0H	; minulá strana rezervována pro BASIC
RAMTOPP	EQU 5CB1H	
RAMTOP	EQU 5CB2H	;(2)
PRAMTP	EQU 5CB4H	; minulá stránka ukázána ve strojovém kód ; mezera
KEYWNO	EQU 0C4H	
TSPEED	EQU 112	
PITOK	EQU 3BH	
PI	EQU PITOK-1AH	; podívej níže
INSTOK	EQU 4AH	
INSTR	EQU INSTOK-1AH	; použito k oddělení N/\$ kódů
FNTOK	EQU 42H	; použití TPRINT
BINTOK	EQU 43H	; použití TPRINT a CALC5BY
SCRNTOK	EQU 4CH	; použití SAVE/LOAD
SINTOK	EQU 53H	; použití TPRINT
INTOK	EQU 60H	;
CODETOK	EQU 6CH	; použití SAVE/LOAD
CHRTOK	EQU 70H	; použití COPY
MODTOK	EQU 7AH	; použití TPRINT

ANDTOK EQU 80H

; použití TPRINT

USINGTOK EQU 085H
ATTOK EQU 087H
TABTOK EQU 088H
WHILETOK EQU 08AH
UNTILTOK EQU 08BH
LINETOK EQU 08CH
THENTOK EQU 08DH
TOTOK EQU 08EH
STEPTOK EQU 08FH

;FORMATOK EQU 091H
;ERASETOK EQU 092H
SAVETOK EQU 094H
LOADTOK EQU 095H
MERGETOK EQU 096H
VERIFYTOK EQU 097H
;CLOSETOK EQU 099H
;CIRCTOK EQU 09AH
;DRAWTOK EQU 09CH
;BORDERTOK EQU 09EH
;CLSTOK EQU 09FH
;INKTOK EQU 0A1H
;PENTOK EQU 0A1H
;PAPERTOK EQU 0A2H
;INVTOK EQU 0A5H
;OVERTOK EQU 0A6H
;MODETOK EQU 0AAH
;RUNTOK EQU 0B0H
;STOPTOK EQU 0B1H
;CONTTOK EQU 0B2H
;CLEARTOK EQU 0B3H
;GOTOTOK EQU 0B4H
;GOSUBTOK EQU 0B5H
;REMTOK EQU 0B7H
;DATATOK EQU 0B9H
;RESTORETOK EQU 0BAH
;PRINTTOK EQU 0BBH
;LISTTOK EQU 0BDH
;LLISTTOK EQU 0BEH
;FORTOK EQU 0C0H
;NEXTTOK EQU 0C1H
;DEFFNTOK EQU 0C8H
;DEFFPROCTOK EQU 0CAH
;ENDPROCTOK EQU 0CBH
;RENUMTOK EQU 0CCH
;DELETETOK EQU 0CDH
;REFTOK EQU 0CEH
;COPYTOK EQU 0CFH
;LOOPIFTOK EQU 0D3H
;DOTOK EQU 0D4H
;LOOPTOK EQU 0D5H
;EXITIFTOK EQU 0D6H
;LIFTOK EQU 0D7H


```
;SIFTOK      EQU 0D8H
;LELSETOK    EQU 0D9H
;ELSETOK     EQU 0DAH
;ENDIFTOK    EQU 0DBH
;ONERRORTOK  EQU 0DDH
;ONTOK       EQU 0DEH
;OUTTOK      EQU 0E0H
;ROLLTOK     EQU 0E5H
;SCROLLTOK   EQU 0E6H
;SCREENTOK   EQU 0E7H
;BOOTTOK     EQU 0E9H
;LABELTOK    EQU 0EAH
;RECORDTOK   EQU 0EFH
```

```
BTHK      EQU 128
```

```
;DOS vyhozen/DOS může ignorovat nebo
vysvětlit jako ALHK/
```

```
;IX=HDR, IX+50H=HDL
```

```
FOPHK      EQU 129
LDHK       EQU 130
VFYHK      EQU 131
SVHK       EQU 132
```

```
;DOS OPEN (GET HEADER) DOS otevřen
;DOS LOAD                  založen
;DOS VERIFY               ověřen
;DOS SAVE                 nahrán
```

```
OSHK       EQU 134
CSHK       EQU 135
ALHK       EQU 136
```

```
;DOS OPEN STREAM /otevřen STREAM/
;DOS CLOSE STREAM (HDR+1)= písmeno
;DOS LOAD AUTO-LOAD řádek - následuje
vyhození
```

```
DIRHK      EQU 137
```

```
DVHK       EQU 139
EOFHK      EQU 140
PTRHK      EQU 141
PATHHK     EQU 142
```

```
;DOS DVAR
;DOS EOF
;DOS PTR
;DOS PATH$
```

```
COMM       EQU 224
TRCK       EQU 225
SECT       EQU 226
DTRQ       EQU 227
DRES       EQU 9
STPIN      EQU 5BH
STPOUT     EQU 7BH
DRSEC      EQU 80H
```

```
; porty diskety
```

```
;SAVE/LOAD považuje za stejné
```

```
HFG        EQU 15
HDT         EQU 26
HDN         EQU 31
HDRL        EQU 80
NMLEN       EQU 10
```

```
;ukazuje skok na plochu
;ukazuje skok na datum/čas
;ukazuje skok na čísla
;HDR buffer LEN
;max.jmeno řádku LEN
```

YOSDISP	EQU 57	
YRGDISP	EQU 67	
XCSDISP	EQU 77	
XRGDISP	EQU 87	
RSBUFF	EQU 0E003H	
SBO	EQU 8000H	
SBN	EQU 0C000H	
HPEND	EQU 4000H	;konec hromadění
		;prostor pro hranici DO/LOOP/PROC STACK
BSTACK	EQU 4AFFH	;spodek hranice
HDR	EQU 4B00H	;skok LEN=50H. Také použití pro PARPRO přejmenování hranice
HDL	EQU 4B50H	;adresa skoku vloženého buferu
INTSTK	EQU 4C00H	;užívá dolů obvykle 49EEH
BUFF256	EQU 4C00H	;musí být vyrovnaná stránka
FPSB	EQU 4D00H	;FPSBOT
CDBUFF	EQU 4D00H	;CODE BUFFER FOR E.G. MULTI-LDI. MAX LEN=0181H
		;MULTI RRD, RLD, LDD (ROLL, SCROLL, CLS)
ISPVAL	EQU 4F00H	;dolní hranice užití 4E98H ;počáteční hodnota SP
INSTBUF	EQU 4F00H	;BUFFER FOR ROM1 XFER CODE, ETC. 0200H
MSGBUFF	EQU INSTBUF+01C0H	;MAX=MGT MSG
FILBUFF	EQU 5080H	
ALLOCT	EQU 5100H	;musí začít na okraji stránky 32 bajtů - 1 na stránce ; plus mezník
MEMVAL	EQU 5121H	
TLBYTE	EQU 513FH	
NMBUFF	EQU 5140H	
FIRLET	EQU NMBUFF	
NMISTK	EQU 5188H	
SCRNBUF	EQU 5188H	;8 bajtů užito SCREEN\$ pro slučitelný tvar
CHARSVAL	EQU 5190H	
PALTAB	EQU 55D8H	
LINICOLS	EQU 5600H	
DKBU	EQU 5800H	
KTAB	EQU 58E0H	
SNDPORT	EQU 0FFH	
KEYPORT	EQU 0FFH	

MDIPORT	EQU 0F0H
VIDPORT	EQU 0FCH
URPORT	EQU 0FBH
LRPORT	EQU 0FAH
STATPORT	EQU 0F9H
CLUTPORT	EQU 0F8H

;PIADVAL	EQU 80H
;ATADVAL	EQU 98H

;mód 1 obrazovka na 8000H
; přisuzuje mód 1

NOISE GEN 0 (22)
3 2 1 0
EXT 7.8 15.6 31.25

0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

ENABLE MIXER

GENERATOR 0
TONE REGISTER (8)
DCT REGISTER LSB (16)

(21)
ADD BIT 1
(20)

AMPLITUDE REG. (0)
128 64 32 16 8 4 2 1

LEFT
RIGHT

GENERATOR 1
TONE REGISTER (9)
DCT REGISTER MSB (16)

(21)
ADD BIT 2
(20)

AMPLITUDE REG. (1)
128 64 32 16 8 4 2 1

RIGHT
LEFT

GENERATOR 2
TONE REGISTER (10)
DCT REGISTER LSB (17)

(21)
ADD BIT 4
(20)

AMPLITUDE REG. (2)
128 64 32 16 8 4 2 1

RIGHT
LEFT

NOISE GEN 1 (22)
48 32 16 0
EXT 7.8 15.6 31.25

CONTROL

GENERATOR 3
TONE REGISTER (11)
DCT REGISTER MSB (17)

(21)
ADD BIT 8
(20)

AMPLITUDE REG. (3)
128 64 32 16 8 4 2 1

RIGHT
LEFT

GENERATOR 4
TONE REGISTER (12)
DCT REGISTER LSB (18)

(21)
ADD BIT 16
(20)

AMPLITUDE REG. (4)
128 64 32 16 8 4 2 1

RIGHT
LEFT

(21)
ADD BIT 32
(20)

AMPLITUDE REG. (5)
128 64 32 16 8 4 2 1

LEFT
RIGHT

GENERATOR 5
TONE REGISTER (13)
DCT REGISTER MSB (18)
MSB OCTAVE VALUES LSB
0 31 TO 61 Hz 0
16 61 TO 122 Hz 1
32 122 TO 244 Hz 2
48 244 TO 488 Hz 3
64 488 TO 977 Hz 4
80 977 TO 195 kHz 5
96 195 TO 391 kHz 6
112 391 TO 781 kHz 7

144 80 48
176 112
192
208
224
240

CONTROL

ENVELOPE REG. (25)

ENVELOPE REG. (24)
BIT 0 0 STRAIGHT ENV.
BIT 1 1 INVERSE ENV.
BIT 2 2 MAX AMPLITUDE
BIT 3 4 SINGLE DECAY
BIT 4 6 REP DECAY
BIT 5 8 SINGLE TRIANGULAR
BIT 6 10 REP TRIANGULAR
BIT 7 12 SINGLE ATTACK
BIT 8 14 REP ATTACK
BIT 9 16 3 BIT RESOLUTION
BIT 10 0 INTERNAL CLOCK
BIT 11 32 EXTERNAL CLOCK
BIT 12 0 DISABLE ENVELOPE
BIT 13 128 ENABLE ENVELOPE

OUTPUT MIXER REGISTER (28)
ENABLE ALL VALUE 1
RESET ALL VALUE 2

SOUND CHIP SAA1099 HELP SHEET



MILES GORDON TECHNOLOGY LTD.

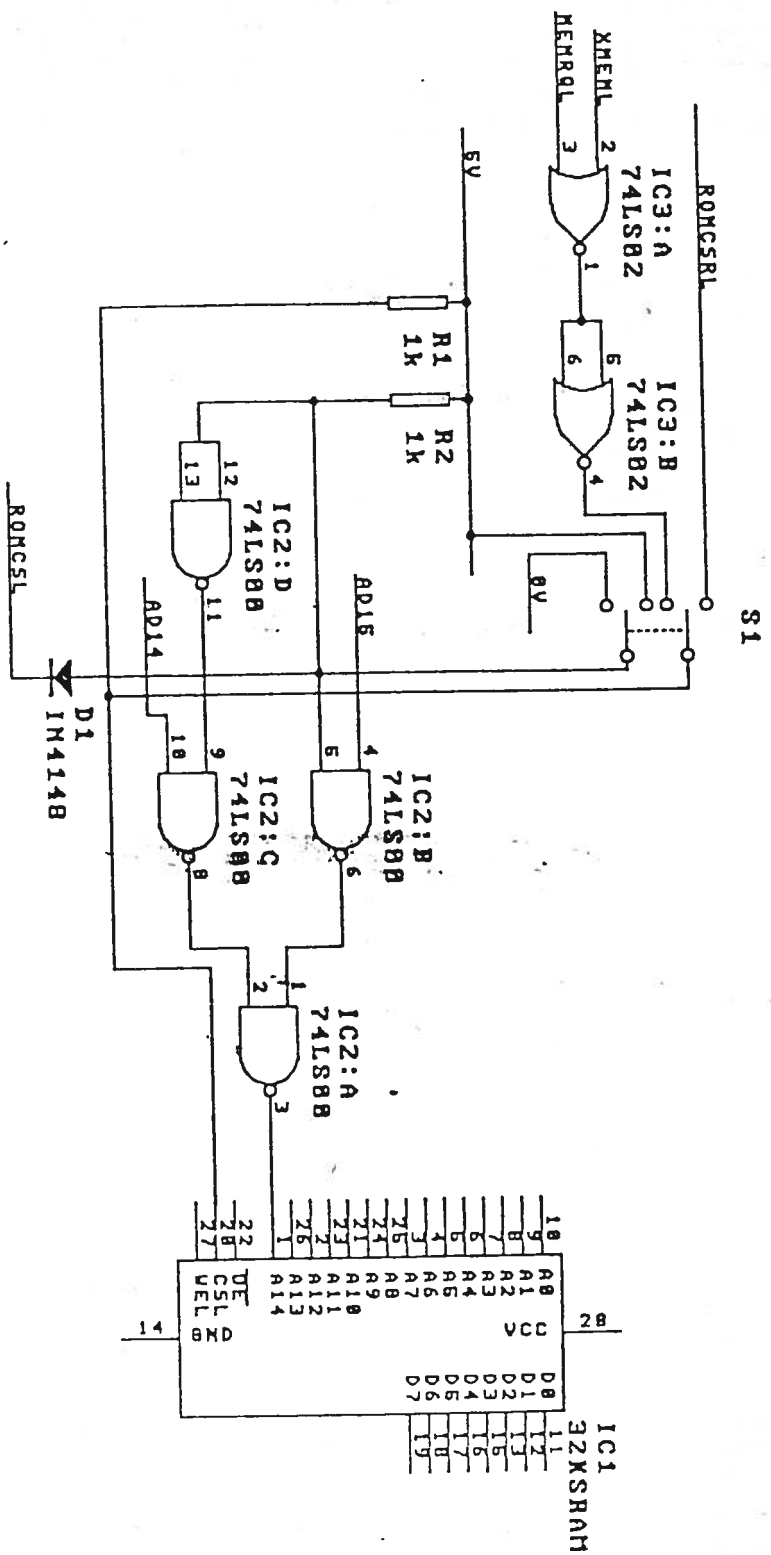


FIG. 1 EXTERNAL RAM ON SAM			
Size	Number	Interface	Revision
A4	1	Interfaces-081	1
Part: 6788(01)	302	Sheet 1 of 1	
		Drawn By: M. J. G.	

1

2

3

4

64K ADDRESSING SPACE OF CPU

0000	3FFF	4000	7FFF	8000	3FFF	C000	FFFF
------	------	------	------	------	------	------	------

ROM 0	SECTION B LMPR +1	SECTION C HMPR	SECTION D HMPR +1
-------	----------------------	-------------------	----------------------

SECTION A LMPR	ROM 1
-------------------	-------

LMPR PORT 250

RAM 0 WRITE	ROM 1	RAM 0	BANK 0 - 1	PAGE 8	PAGE 4	PAGE 2	PAGE 1
----------------	-------	-------	---------------	-----------	-----------	-----------	-----------

HMPR PORT 251

MCNTRL	MD3S1	MD3S0	BANK 0 - 1	PAGE 8	PAGE 4	PAGE 2	PAGE 1
--------	-------	-------	---------------	-----------	-----------	-----------	-----------

VMPR PORT 252

MIDI BIT	MDE 1	MDE 0	BANK 0 - 1	PAGE 8	PAGE 4	PAGE 2	PAGE 1
-------------	-------	-------	---------------	-----------	-----------	-----------	-----------

BANK 0

PAGE 0
PAGE 1
PAGE 2
PAGE 3
PAGE 4
PAGE 5
PAGE 6
PAGE 7
PAGE 8
PAGE 9
PAGE 10
PAGE 11
PAGE 12
PAGE 13
PAGE 14
PAGE 15

BANK 1

PAGE 16
PAGE 17
PAGE 18
PAGE 19
PAGE 20
PAGE 21
PAGE 22
PAGE 23
PAGE 24
PAGE 25
PAGE 26
PAGE 27
PAGE 28
PAGE 29
PAGE 30
PAGE 31

< IF FITTED >

SAM COUPE MEMORY MAP

© MILES GORDON TECHNOLOGY LTD.

SCREEN ATTRIBUTE MAP

1800		TOP GROUP	181F
18E0			18FF
1900			191F
19E0		MIDDLE GROUP	19FF
1A00			1A1F
1AE0		BOTTOM GROUP	1AFF

SCREEN BIT MAP

0000	LINE 1	ROW 1	001F
0100	LINE 2		011F
0200	LINE 3		021F
0300	LINE 4		031F
0400	LINE 5		041F
0500	LINE 6		051F
0600	LINE 7		061F
0700	LINE 8		071F
0020		ROW 2	003F
0720		ROW 3	073F
0040		ROW 4	005F
0740		ROW 5	075F
0060		ROW 6	007F
0760		ROW 7	077F
0080		ROW 8	009F
0780			079F
00A0			00BF
07A0			07BF
00C0			00DF
07C0			07DF
00E0			00FF
07E0			07FF
0800			081F
08E0		MIDDLE GROUP	08FF
1000		BOTTOM GROUP	101F
12E0			12FF

SCREEN MODE 1

22-05-1989

© MILES GORDON TECHNOLOGY LTD.

2000							
2700							
2800							
2900							
3000							
3100							
3200							
3300							
3400							
3500							
3600							
3700							
3800							
3900							
4000							

[illegible]

SCREEN BIT MAP

MILES GORDON TECHNOLOGY LTD.

SCREEN 2 BITS PER PIXEL - MODE 3

	PIXEL 1	PIXEL 2	PIXEL 3	PIXEL 4	PIXEL 5	PIXEL 6	PIXEL 7	PIXEL 8	PIXEL 9	PIXEL 10	PIXEL 11	PIXEL 12	PIXEL 13	PIXEL 14	PIXEL 15	PIXEL 16							PIXEL 509	PIXEL 510	PIXEL 511	PIXEL 512
LINE 1	0000	0000	0001	0002	0003																					
LINE 2	0080	0081	0082	0083																						
LINE 3	0100																									

LINE 128	3F80																									
----------	------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

NEXT PAGE

LINE 129	0000	0001	0002	0003																						
LINE 192	1F80																									

SCREEN 4 BITS PER PIXEL - MODE 4

	PDXEL 1	PDXEL 2	PDXEL 3	PDXEL 4	PDXEL 5	PDXEL 6	PDXEL 7	PDXEL 8					PDXEL 255	PDXEL 256
LINE 1	0000		0001		0002		0003						007C	007E
LINE 2	0080		0081		0082		0083						007C	007D
LINE 3	0100												007E	007F

LINE 128	3F80													3FTF
----------	------	--	--	--	--	--	--	--	--	--	--	--	--	------

NEXT PAGE

LINE 129	0000	0001	0002	0003										007F
LINE 192	1F80													1FTF

SCREEN MODE 4

22-05-1989



MILES GORDON TECHNOLOGY LTD.

INPUT - OUTPUT PORT MAP

PORT	READ	80	40	20	10							WRITE	80	40	20	10							PORT	
		128	64	32	16	8	4	2	1				128	64	32	16	8	4	2	1	0			
		7	6	5	4	3	2	1	0				7	6	5	4	3	2	1	0				
	ATTRIBUTES	FLASH	BRIGHT	GREEN PAPER	RED PAPER	BLUE PAPER	GREEN DPK	RED DPK	BLUE DPK			SQUAND	DATA BUS										255	FF
FE	KEYBOARD	SOFT (00011)	CLK	SPCH	R3	R4	R5	R6	R7			BORDER	SOFT	THROH	VIDEO RCD 0	SCD 0	SCD 1	VIDEO RCD 1	SCD 2	VIDEO RCD 2	SCD 3	VIDEO RCD 3	254	FE
FD	MIDI-IN	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0			MIDI-OUT	D7	D6	D5	D4	D3	D2	D1	D0			253	FD
FC	V-PAGE	ROCKED	HIE 1	NOE 0	0-1	PAGE 0	PAGE 1	PAGE 2	PAGE 3			V-PAGE	THROH	NOE 1	NOE 0	0-1	PAGE 0	PAGE 1	PAGE 2	PAGE 3			252	FC
FB	HM-PAGE	CONTROL	NOE1	NOE2	0-1	PAGE 0	PAGE 1	PAGE 2	PAGE 3			HM-PAGE	CONTROL	NOE1	NOE2	0-1	PAGE 0	PAGE 1	PAGE 2	PAGE 3			251	FB
FA	LM-PAGE	RAM PROT	RAM 1	RAM 0	0-1	PAGE 0	PAGE 1	PAGE 2	PAGE 3			LM-PAGE	RAM PROT	RAM 1	RAM 0	0-1	PAGE 0	PAGE 1	PAGE 2	PAGE 3			250	FA
F9	STATUS	R3	R2	R1	NOI OUT DIT	NOI IN DIT	NOI OUT DIT	NOI IN DIT	NOI OUT DIT			LINE INT	LINE 128	LINE 64	LINE 32	LINE 16	LINE 8	LINE 4	LINE 2	LINE 1			249	F9
F8	PEN	UPON (0001)	LINE 128	LINE 64	LINE 32	LINE 16	LINE 8	LINE 4	LINE 2	LINE 1		CLUT	A3 - 1	A3 - 2	A3 - 3	A3 - 4	A3 - 5	A3 - 6	A3 - 7	A3 - 8			248	F8

E0	224	DISC 1	RANGE 224 TO 231	USING A0, A1, A2		
E8	232	PRINTL	RANGE 232 TO 239	USING A0, A1, A2		
F0	240	DISC 2	RANGE 240 TO 247	USING A0, A1, A2		

PORTS

SEP 1989

© MILES GORDON TECHNOLOGY INC.

Sam Coupe Keycode table (Refer to BASIC Specification)

23	68	59	50	41	32	31	40	49	58	67	22	13	4	24	15	6
14	69	60	51	42	33	30	39	48	57	66	21	12		25	16	7
5	70	61	52	43	34	29	38	47	56	20	11		65	26	17	8
+70	62	53	44	35	28	37	46	19	70	1			+70	3	54	10
+140	+210	64										2	+140	36	45	27

SAM-KEY 260789

PLAN VIEW

ESC	1	2	3	4	5	6	7	8	9	0	-	+	DELETE	F7	F8	F9
TAB	<	>	E	R	T	Y	U	I	O	P	=	.		F4	F5	F6
CAPS	A	S	D	F	G	H	J	K	L	,	'		RETURN	F1	F2	F3
SHIFT	Z	X	C	V	B	M	N	.	.		INV		SHIFT	F0	^	.
SYMBOL	CTRL											EDIT	SYMBOL	<	>	>

LOWER MEMBRANE
CONNECTOR

	9	10	11	12	13	14	15	16	17
PIN 8	49	33	17	34	14	62	68	60	
PIN 7	48	32	16	18	13	30	45	64 59	
PIN 6	47	31	15	1	12	29	44	58	
PIN 5	54	39	23	6	7	24	40	55	72
PIN 4	53	38	22	5	8	25	41	56	70
PIN 3	52	37	21	4	9	26	42	57	71
PIN 2	51	36	20	3	10	27	43	65 69	63
PIN 1	50 61	35	19	2	11	28	46	67	66

UPPER MEMBRANE
CONNECTOR

KEYS = 72
CODES = 69

DUPLICATE
50-61
59-64
65-69

SAM KEYBOARD MATRIX 260789

Back Panel Help Sheet

